Freescale Semiconductor
Application Note

AN2822
Rev. 0, 9/2004

MSC8103 Bootload Through the HDI16 Port Using Any Host

by Manoj Bapat

In wireless infrastructure and media applications, the Freescale MSC8103 DSP device typically performs speech coding and echo cancellation functions. To implement these functions, communications systems often contain a control processor (host) that distributes the program and data to the MSC8103 DSP devices. The MSC8103 device features a 16-bit parallel port (HDI16) for connecting to a host processor. Additional logic devices are not necessary for connecting an MSC8103 host processor to the HDI16 port of an MSC8103 slave device. However, other host processors may require minimal logic. The HDI16 port contains four address lines (HA[0–3]) for mapping control, data, and reset configuration registers to the memory space of the host processor so that the host has direct control over the MSC8103 slave.

In typical DSP applications, the host processor must write the data and source programs to slave devices during reset (bootloading). Therefore, the option to transmit source programs through the host interface is an important feature of the MSC8103 device. The bootloader program stored in the ROM provides the user with the code necessary to implement this procedure after power-on reset. This application note describes a general programming model for bootloading an MSC8103 device through the HDI16 host interface. The bootloader program stored in MSC8103 ROM loads and executes the source programs distributed by the host processor connected to the HDI16 port. Although the bootloading procedure can be performed using any host processor, an example external host MSC8103 processor is used here for purposes of illustration. This host MSC8103 processor uses its 60x-compatible system bus to send data to the HDI16 port of the slave MSC8103 device.

CONTENTS

This document assumes a working knowledge of the HDI16 port along with the ability to program the MSC8103 DSP in C language. Information on the HDI16 pins, registers, and programming model is available in the *MSC8103 Reference Manual*. [1]

1 Bootloading Overview

During the power-on-reset sequence, an MSC8103 slave device waits for the hard reset configuration word (HRCW) to complete the reset process. The host must transmit the HRCW in order to “wake up” the MSC8103. Then the bootload program stored in the slave MSC8103 ROM begins executing. Next, the host transmits the source program to be loaded and executed on the MSC8103 slave device. The host-side programmer must follow strict guidelines for a successful bootload procedure. These guidelines are based on the sequence of instructions in the bootload program. Bootloading through the host interface occurs in three phases that are essential for proper execution (see the flow chart in **Figure 1**):

- Loading the HRCW to the MSC8103 slave.
- Programming the host to initialize the slave HDI16 port.
- Preparing the data blocks to be transferred.

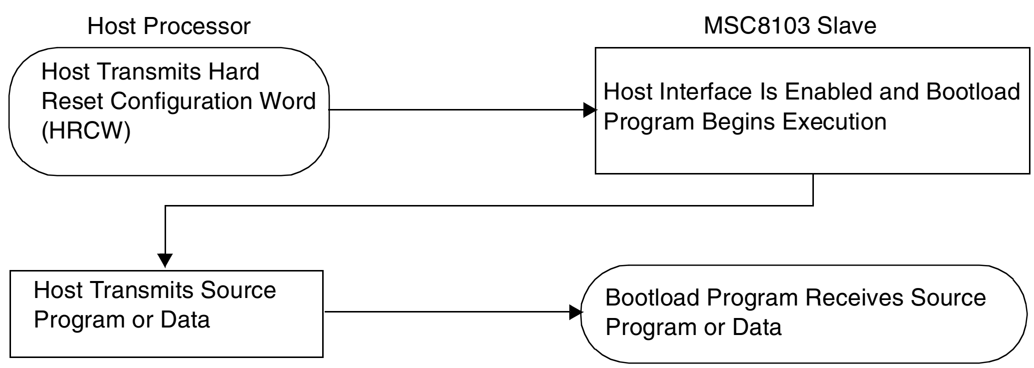

Figure 1. General Bootload Flow Chart

An important feature of the MSC8103 device is its ability to bootload via the HDI16 in 8-bit or 16-bit mode. The host 8-bit (H8BIT) pin must be pulled high during reset if 8-bit functionality is desired. For 16-bit functionality, the H8BIT pin is pulled low. Note that all procedures, flow charts, and examples in this application note are described for 16-bit transfer mode. In 8-bit transfer mode, the wiring is exactly the same except that only the HD[8–15] pins are used. The 8-bit reset configuration sequence is the same as that for the 16-bit mode because the HRCW registers are written in 8-bit segments. In addition, the source code and block structure must be written to bootload the source program in 8-bit segments instead of 16-bit segments. Lastly, the host can read and write only the least significant byte (LSB) of the Interface Control Register (ICR) and Command Vector Register (CVR) if the HDI16 port is operating in 8-bit mode.

1. See also the Freescale application note entitled *Bootstrapping the MSC8101 Device Through the HDI16 Port* (AN2311/D), which describes a software driver to bootstrap a slave MSC8101 device through the slave HDI16 port from an external MSC8101 host.

The host reset configuration sequence, which allows the host to initialize and program the MSC8103 slave device through its host interface, proceeds as follows:

1. Enable the MSC8103 device.

   Begin by sampling the HPE/EE1 pin. HPE must be sampled high on the rising edge of $\overline{\text{PORESET}}$ to enable the MSC8103 host port.

2. Specify the boot mode.

   The BTM[0–1]/EE[4–5] pins are also sampled at the rising edge of $\overline{\text{PORESET}}$. These pins specify the boot mode and must be set to 0 1 for the MSC8103 device to boot via its HDI16 port.

3. Select slave mode.

   $\overline{\text{RSTCONF}}$ must be sampled high on deassertion of $\overline{\text{PORESET}}$ for slave configuration.

4. Start the bootload.

   DBREQ/EE0 must be sampled low on the deassertion of $\overline{\text{PORESET}}$. Otherwise, the device enters Debug mode.

5. Specify either 8-bit or 16-bit mode.

   H8BIT must be sampled high for 8-bit functionality and low for 16-bit functionality.

6. Send the HRCW to the slave HDI16 port to complete the host reset configuration sequence.

   The HRCW is 32 bits wide.[2] The host must send the HRCW in 8-bit segments to the four reset configuration registers (RSCFG [0–3]) associated with the HDI16 port. The HRCW is written in 8-bit segments in both 8-bit mode and 16-bit mode. The RSCFG0 register is the least significant part of the HRCW, and the RSCFG3 register is the most significant part. After four consecutive writes to the mapped addresses 0x8 (RSCFG0), 0x9 (RSCFG1), 0xA (RSCFG2), and 0xB (RSCFG3) on the slave device, the hardware indicates that the values in the reset configuration registers are valid. This completes the reset sequence. Several bits in the HRCW are critical to HDI16 operation. The HRCW[4–5]:BPS bits must have a value of 11 (32-bit port size). The HRCW[I7]:ISPS bit must also have a value of 1. These selections allow the upper 32 bits of the 60x-compatible system bus to be reserved for the HDI16 port.

Figure 2 shows an example host processor connected to the HDI16 port of a slave MSC8103 device. The system data bus lines of the host processor (D[0–15]) directly connect to the data lines of the HDI16 port (HD[0–15]). Since the MSC8103 is a big-endian device, the D0 pin is the most significant. In addition to the data bus lines, the system address bus lines (A[28–31]) directly connect to the host address lines of the HDI16 (HA[0–3]). There are two chip selects for the HDI16, which are active low. In this example, $\overline{\text{HSC1}}$ is connected to the memory controller chip select, and $\overline{\text{HSC2}}$ is pulled high. The read and write strobes are also critical for data transfers, and they must be connected as shown.

Note: Resistors shown are not part of the internal MSC8103 device.

2. For definitions of the bits in the HRCW, consult the Reset chapter of the *MSC8103 Reference Manual*.

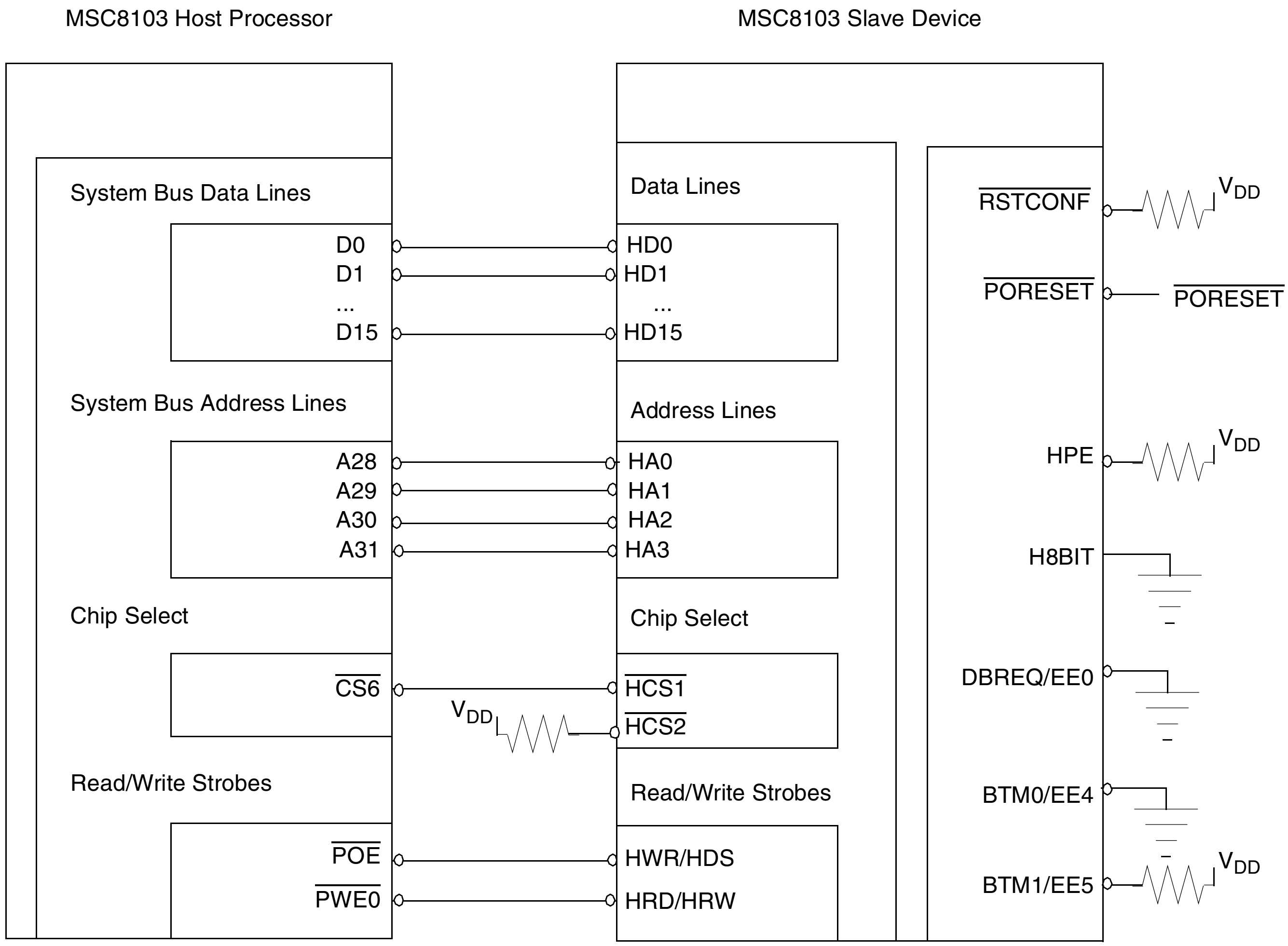

Figure 2. MSC8103 Slave Reset Configuration Set-up For 16-Bit HDI16 Functionality

2 Slave-Side Bootload Program

The bootloading sequence for the HDI16 section of the bootloader program proceeds as follows:[3]

1. Initialize the MSC8103 Vector Base Address (VBA) register and stack pointer to set the base register for the interrupt vector table and initialize a stack pointer for possible exceptions.

   The stack address is 0x68000 in the MSC8103 RAM. When the bootloader program executes, several subroutines are called. When the program jumps to a subroutine, the stack address contains information on the status register and the program counter. If you transfer data to this memory location, the data is corrupted when these subroutines are called in the bootloader program.

2. Configure the Internal Space Base (ISB) address value of the Internal Memory Map Register (IMMR) to determine the base address of the internal memory space of the system interface unit (SIU) defined by the HRCW and sent by the host processor.
3. Initialize the SRAM to program and set up memory controller banks 10 and 11.
4. Configure the PIC Edge/Level-Triggered Interrupt Priority Register E (ELIRE) to a value of 0x8000 to select IRQ19 edge-triggered mode.

3. The boot sequence for loading a source program via the HDI16 port is determined by the bootloader program shown in an Appendix to the *MSC8103 Reference Manual*.

5. Configure the PIC Edge/Level-Triggered Interrupt Priority Register F (ELIRF) to a value of 0x0008 for $\overline{IRQ20}$ (EOnCE interrupt) edge-triggered mode.
6. Initialize User Programmable Machine C (UPMC) to generate timing patterns for control signals that administer the SRAM.

   The set-up is based on the System Clock Mode Register (SCMR).
7. To determine the boot mode of the slave MSC8103 device, configure the EE4 and EE5 bits in the Exception and Mode Register (EMR).

   If the EE4 and EE5 bits have a value of 01, the bootloader program attempts to load the source program from the HDI16.
8. Disable the software watchdog timer in the System Protection Control Register (SYPCR) to prevent counter time-outs during the bootload procedure: SYPCR[29]:SWE = 0.
9. Set the HEN bit in the HDI16 Host Port Control Register (HPCR) to enable the host interface.
10. Configure the HPCR[9]:H8BIT bit to specify whether the MSC8103 slave device operates in 8-bit or 16-bit transfer mode.

    This bit is set during power-on reset. If the H8BIT pin is low, then the value in the HPCR is cleared to 0 and the device is enabled for 16-bit transfer mode. If the H8BIT pin is high, the value in the HPCR is ignored and the device is enabled for 8-bit transfer mode.
11. Initialize the slave HDI16 port.

    The host processor must set the INIT bit of the Interface Control Register (ICR) in the slave HDI16 port.
12. Transfer data from the host processor to the slave HDI16 port, including calculation of checksums.

    This step transfers all relevant data to specified locations in internal memory for the source program and ensures that data is transferred correctly. The bootloader program sets the following bits in the HCR, which is reflected to the ICR for host-side visibility:
    - HCR[0]:HF4 bit to indicate that the source program has finished loading.
    - The HCR[3]:HF7 bit to indicate whether there is a data transfer error. HF7 is set only if the calculated checksum does not equal the checksum value transferred with the source program. The checksum is optional and is governed by the host processor during initialization of the ICR.
13. Execute the source program transferred by the host processor.

This summary of the HDI16 portion of the bootloader program provides essential information on programming the host. For a detailed flow chart, see **Figure 3**.

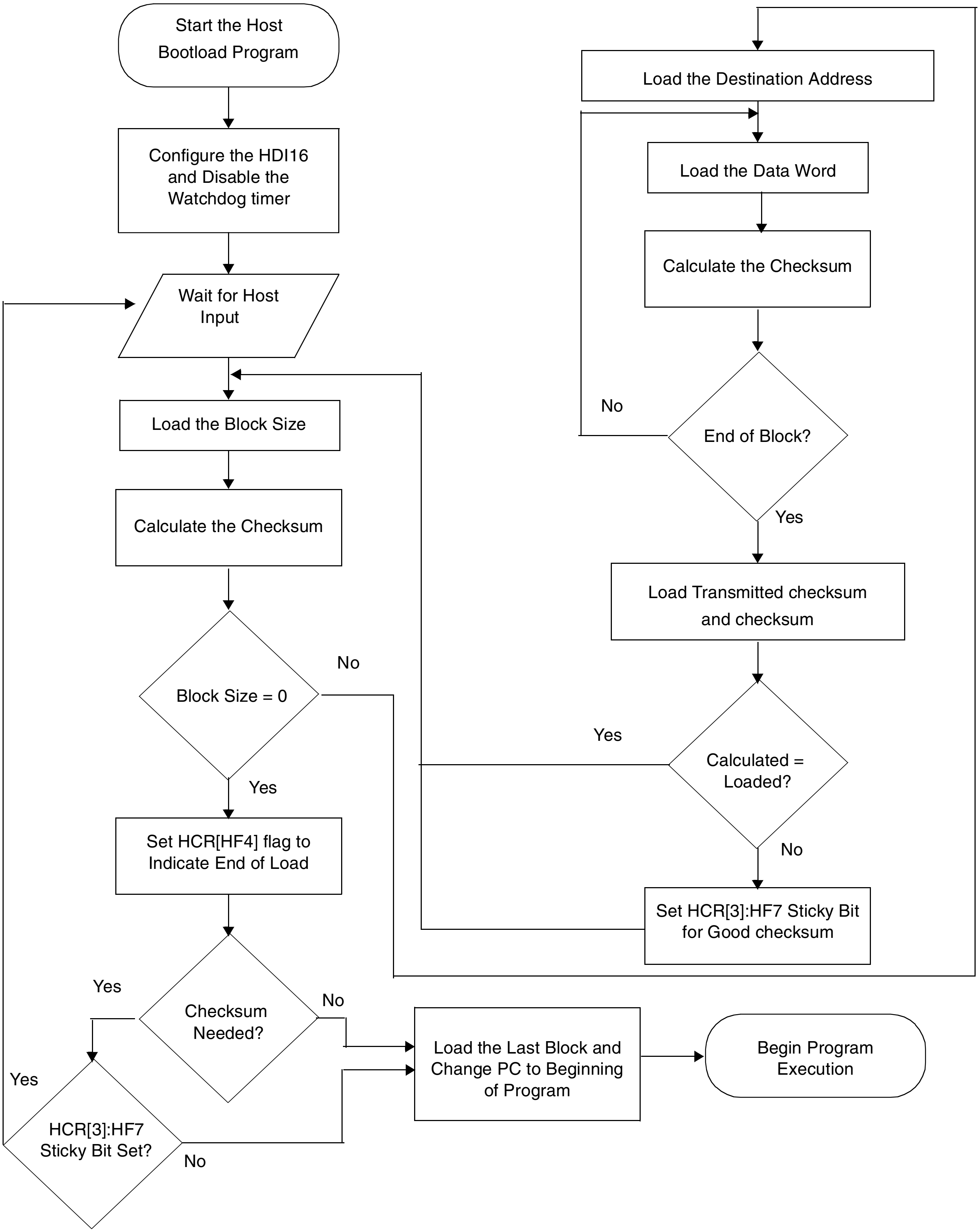

Figure 3. Slave-Side Programming Flow Chart

2.1 Host-Side Bootload Programming

The host-side programming procedure is a direct complement to the slave-side procedure, as follows:

1. Initialize the host-side controllers and registers to configure the port or memory controller that is directly connected to the MSC8103 slave device.
2. Initialize the pointers to the MSC8103 HDI16 Transfer (TX) registers to enable these pointers for data transfers to the TX registers of the HDI16 port.
3. Initialize the pointers to the ICR and ISR of the slave device to enable these pointers for ICR initialization and reading the status of the MSC8103 Host Receive (HORX) data register.
4. Initialize the pointers to the HRCW and source program data bytes to prepare these pointers for data transfers from internal memory, where the source program is stored, to the MSC8103 HDI16 port.
5. Transmit the HRCW to the MSC8103 slave device to reset the MSC8103 device so that the bootloader program is activated.
6. Poll the Transmit Data Empty (TXDE) bit in the ISR to ensure that the TX registers are empty and ready for writes from the host processor.
7. Initialize the ICR[8]:INIT bit so the bootloader program can proceed to transfer the source program.

   In addition to the INIT bit, the ICR[12]:HF3 bit can be set for an optional checksum. If HF3 is set, the bootloader performs a checksum to ensure that data is transferred properly. If HF3 is clear, the bootloader program ignores the checksum routine.
8. Transmit data to the MSC8103 transmit registers.

   The host transmits the source program to the slave MSC8103 device. The MSC8103 HCR is configured by default to have ICR/HCR priority for DMA/Last Address Mode (HICR) bit cleared, which signifies that the last address mode is defined in the HCR. In addition, the Host DMA/Last Address Mode Control (HDM[0–2]) bits are cleared by default in the HCR to indicate that the trigger address in the TX registers is 0x7. The bootloader program assumes that the 64 bits are written to the TX registers before the address is triggered for transfer. That is, 16 bits are written to TX3 (address 0x4), TX2 (address 0x5), TX1 (address 0x6), and TX0 (address 0x7), which then triggers the transfer to the HORX register.
9. Complete the data transfer by checking HCR[0]:HF4 and HCR[3]:HF7 of the slave MSC8103 device to ensure that data loaded and transmitted without errors. HF4 is set if the load completed properly. The checksum is an optional feature in the bootload procedure. If a checksum is required, then HF7 determines whether it is necessary to retransmit the data. Otherwise, the program terminates.

This section provides only a general structure of the host-side programming that complements the MSC8103 slave-side program structure. For a detailed flow chart, see **Figure 4**.

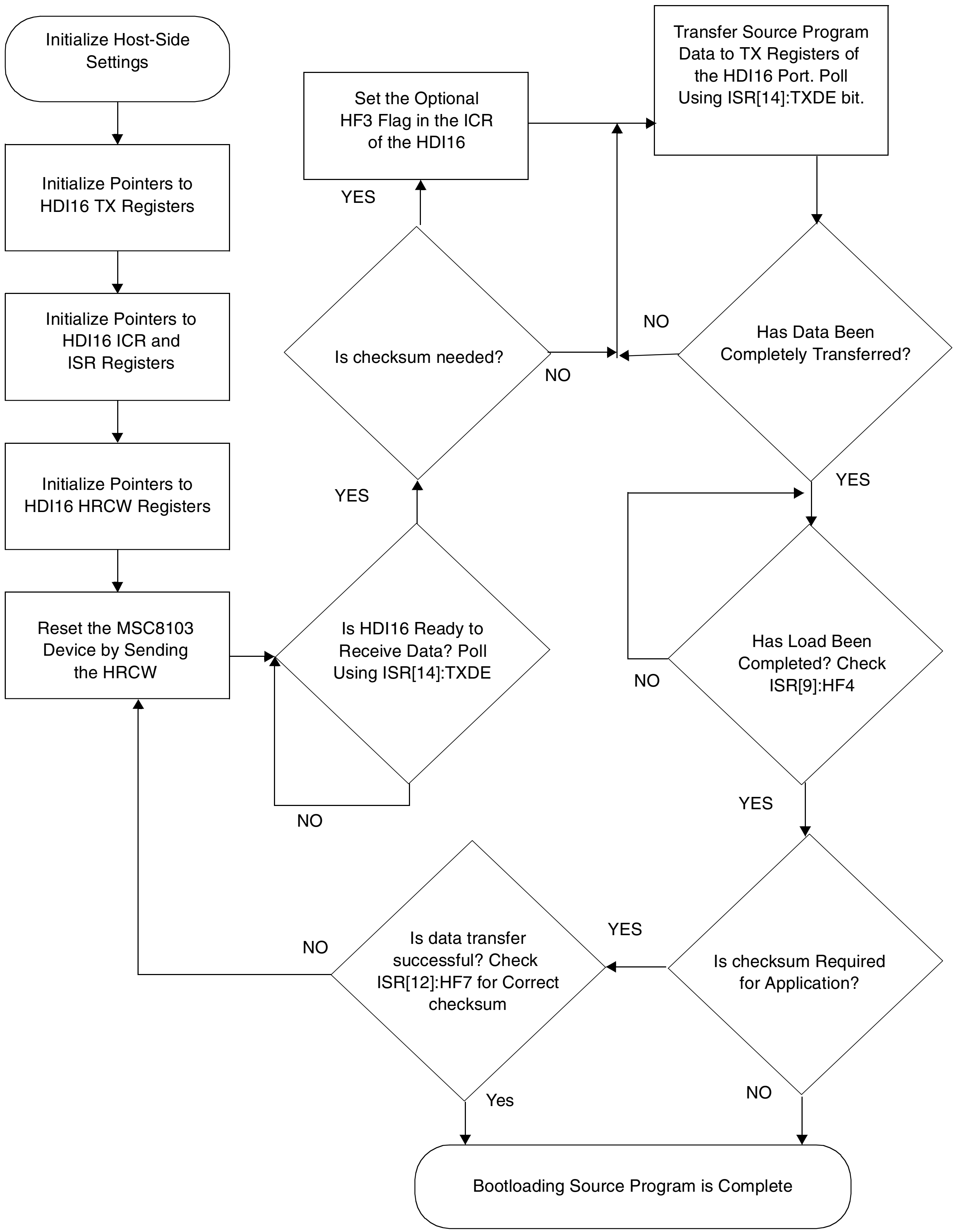

Figure 4. Host-Side Bootload Program Flow Chart

2.2 Block Structures

The structure of the transmitted source program is critical to the execution of the bootload procedure. The source program is transferred in 16-bit opcodes or data words that can be obtained from the list file that is created after the program is designed and assembled. The source opcodes are transferred in blocks. The bootload program expects the opcodes or data to be transferred according to the following rules and formatted as shown in **Table 1** and **Table 2**, which illustrate the block structure for the MSC8103 device. The shading in the tables represent the items that differ between the early mask set revisions of the MSC8101 device and the MSC8103 device.

- The source data must be aligned on a 16-byte boundary in MSC8103 memory. This directly corresponds to the address where the blocks are to be stored in the MSC8103.
- The block size is the number of source program words (16-bits) in a particular block and must be a multiple of 8 bytes. If the block size is 0, the bootloader program assumes that it is the last block.
- The source data must be ordered in big-endian format so that the lower address contains the most significant data.
- There must be at least two blocks transferred (first block and end block), but there is no limit to the number of blocks that can be transferred. The only restriction is the size of the memory in the MSC8103 device.
- The CHECKSUM and $\overline{\text{CHECKSUM}}$ are calculated using the exclusive OR (XOR) function. Each transmitted opcode or data word is XOR'd with the previous XOR result. The first block to the n – 1 block checksum calculation includes the size and the address where the block is stored. The checksum in the last block should also include the address where the bootload program starts. The following example illustrates how the checksum should be calculated.

  Find the checksum given the following 4-bit inputs.

  —Input 1 = 1010

  —Input 2 = 0111

  —Input 3 = 1100

  Solution Part 1:

  XOR —1010 = Input 1

  XOR —0111 = Input 2

  —1101 = Result 1

  Solution Part 2:

  XOR —1101 = Result 1

  XOR —1100 = Input 3

  —0001 = *Final Checksum Result*

Note: Because of the BOOT1 erratum, the checksum does not apply to the MSC8101 bootload procedure in any mask et of the MSC8101 device that precedes mask 1K87M (RevA). That is, it does not apply to 0K40A, 1K42A, or 2K42A.

Table 1. Structure of First and Successive Blocks For MSC8103 and MSC8101, Mask Set 2K87M

16-bit Word Order	Description
1	Most significant part of the size of the first program block to be loaded
2	Least significant part of the size of the first program block to be loaded
3	Most significant part of the address where the first program block is to be loaded into the MSC8103 slave device
4	Least significant part of the address where the first program block is to be loaded into the MSC8103 slave device
5	First word of the source program
n	Last word of the source program
n+1	$\overline{\text{Checksum}}$ - $\overline{\text{XOR}}$ for first block, including address and size
n+2	Checksum - XOR for first block, including address and size

Table 2. Structure of Last Block for MSC8103 and MSC8101, Mask Set 2K87M

16-bit Word Order	Description
1	Must contain the following data: 0x0000
2	Must contain the following data: 0x0000
3	Most significant part of the start address of the source program
4	Least significant part of the start address of the source program
5	Must contain the following data: 0x0000
6	Must contain the following data: 0x0000
7	Checksum - XOR for first block, including address and size
8	$\overline{\text{Checksum}}$ - $\overline{\text{XOR}}$ for first block, including address and size

The block structure of early revisions of the MSC8101 device (see **Table 3** and **Table 4**) differs slightly from that for the 2K87M mask set. The shading in the tables represent the items that differ between the early mask set revisions and 2K87M.

Table 3. Structure of Blocks for 1K42A and 2K42A MSC8101 Mask Set Revisions

16-bit Word Order	Description
1	Most significant part of the size of the first program block to be loaded
2	Least significant part of the size of the first program block to be loaded
3	Most significant part of the address where the first program block is to be loaded into the MSC8103 slave device
4	Least significant part of the address where the first program block is to be loaded into the MSC8103 slave device
5	First word of the source program
n	Last word of the source program
n+1	$\overline{\text{Checksum}}$ - Not applicable
n+2	Checksum - Not applicable

Table 4. Last Block for 1K42A and 2K42A MSC8101 Mask Set Revisions

16-bit Word Order	Description
1	Must contain the following data: 0x0000
2	Must contain the following data: 0x0000
3	Most significant part of the start address of the source program
4	Least significant part of the start address of the source program
5	Checksum - Not applicable
6	Checksum - Not applicable
7	Must contain the following data: 0x0000
8	Must contain the following data: 0x0000

3 16-Bit Bootload Example

This section describes how to implement a 16-bit bootload through the HDI16 port of an MSC8103 slave device using an MSC8103 host processor. The example shows how to reset the MSC8103 slave device through the HDI16 port and demonstrates the bootload procedure described in the previous sections. Source code is available in the `.zip` file that accompanies this application note (`AN2822SW.zip`). Comments in the source code help you to follow the example for easy implementation. To implement the example, you must have two MSC8103 application development system (MSC8103ADS) boards, wiring capabilities, and the ability to create and assemble programs for the MSC8103 device.

The source code loaded into the MSC8103 slave device is a program that flashes LEDs on the MSC8103ADS board to indicate whether the bootload succeeded. The source code for the host processor does not use the checksum option. If the checksum option is necessary, the block structure should be as indicated in **Section 2.2**.

3.1 MSC8103ADS Board Settings

To implement the 16-bit bootload example, it is necessary to set up the MSC8103ADS boards with the switch settings detailed in this section for both slave-side and host-side switches.

3.1.1 Slave-Side Switch Settings

All switch settings for the 16-bit bootload example using the slave-side MSC8103ADS board are detailed as follows:

- *Host Interface Setting*. Switch 1 is defined in **Figure 5**.

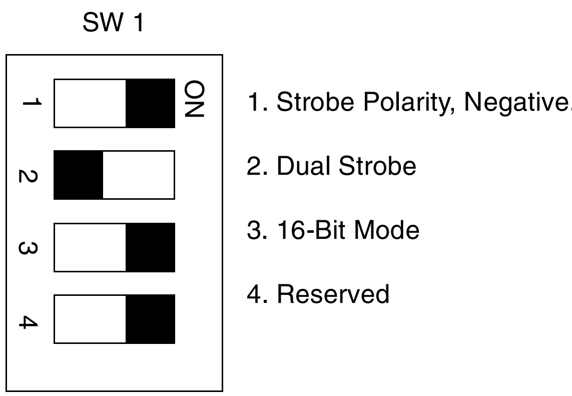

Figure 5. Slave-Side MSC8103ADS Switch 1Settings

- *Emulator Enable*. Switch 2 settings are defined in **Figure 6**.

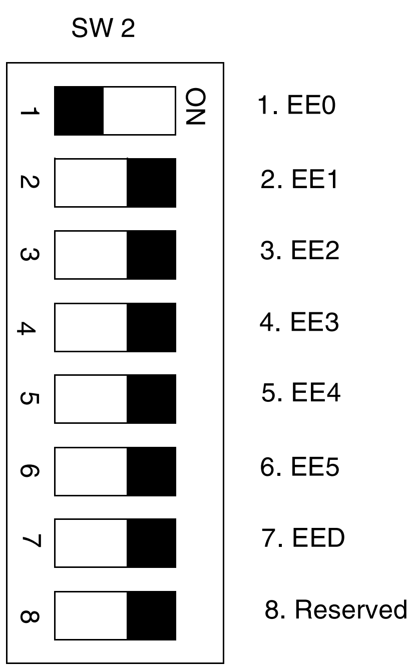

Figure 6. Slave-Side MSC8103ADS Switch 2 Settings

- *Data Bus Width Setting*. Switch 5 and Switch 6 are defined in **Figure 7**.

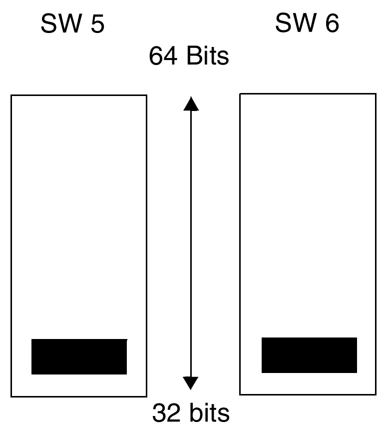

Figure 7. Slave-Side MSC8103ADS Switch 5 and Switch 6 Settings

- *Configuration Switch*. Switch 9 is defined in **Figure 8**. The slave MSC8103 uses clock mode 40 in this example. With a 20 MHz input oscillator, this mode yields a core speed of 80 MHz.

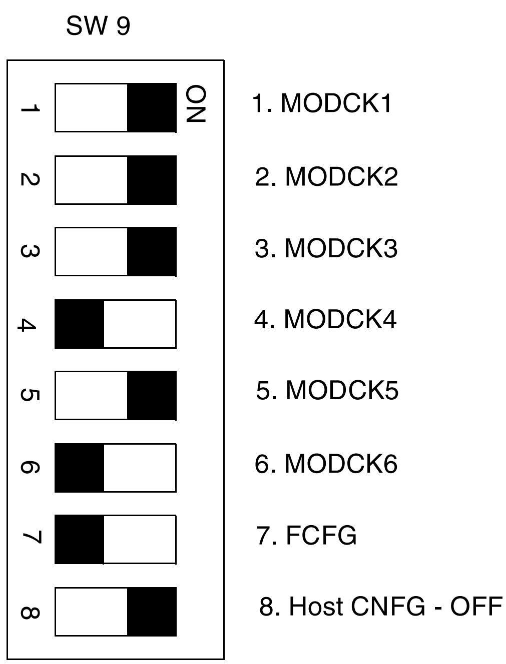

Figure 8. Slave-Side MSC8103ADS Switch 9 Settings

- *Boot Mode Select*. Switch 10 is defined in **Figure 9**.

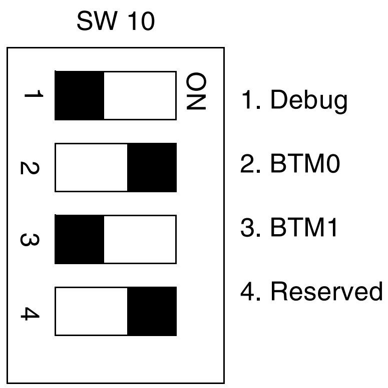

Figure 9. Slave-Side MSC8103ADS Switch 10 Settings

- *Software Options*. Switch 11 is defined in **Figure 10**.

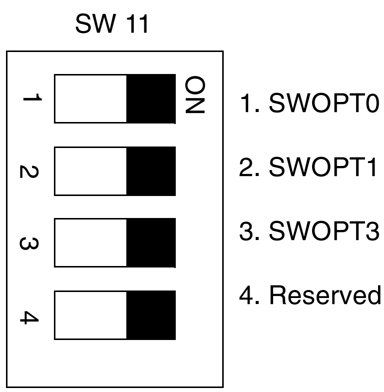

Figure 10. Slave-Side MSC8103ADS Switch 11 Settings

3.1.2 Host-Side Switch Settings

This section defines all switch settings for the 16-bit bootload example using the host-side MSC8103ADS board, as follows:

- *Host Interface Setting*. Switch 1 is defined in **Figure 11**.

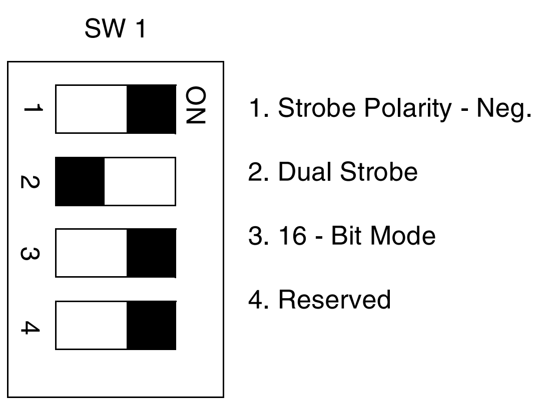

Figure 11. Host-Side MSC8103ADS Switch 1 Settings

- *Emulator Enable*. Switch 2 settings are defined in **Figure 12**.

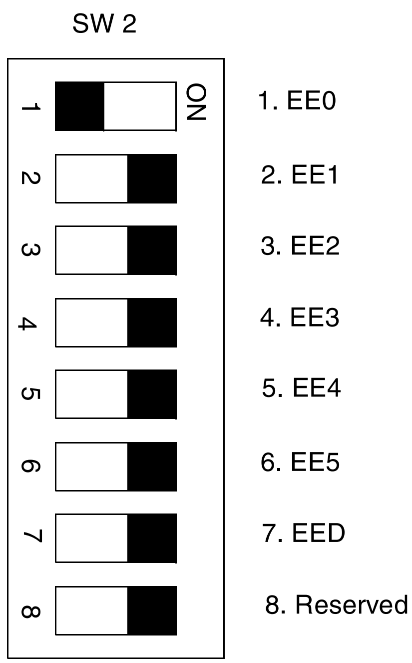

Figure 12. Host-Side MSC8103ADS Switch 2 Settings

- *Data Bus Width Setting*. Switch 5 and Switch 6 are defined in **Figure 13**.

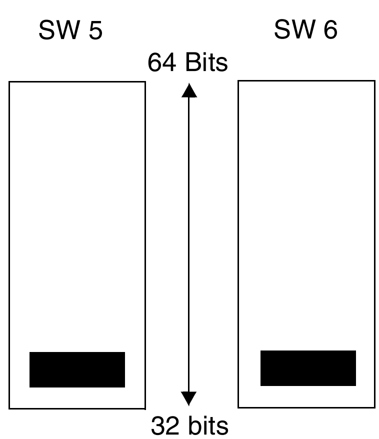

Figure 13. Host-Side MSC8103ADS Switch 5 and Switch 6 Settings

- *Configuration Switch*. Switch 9 is defined in **Figure 14**. The host MSC8103 uses clock mode 40 in this example. With a 20 MHz input oscillator, the core speed is 80 MHz.

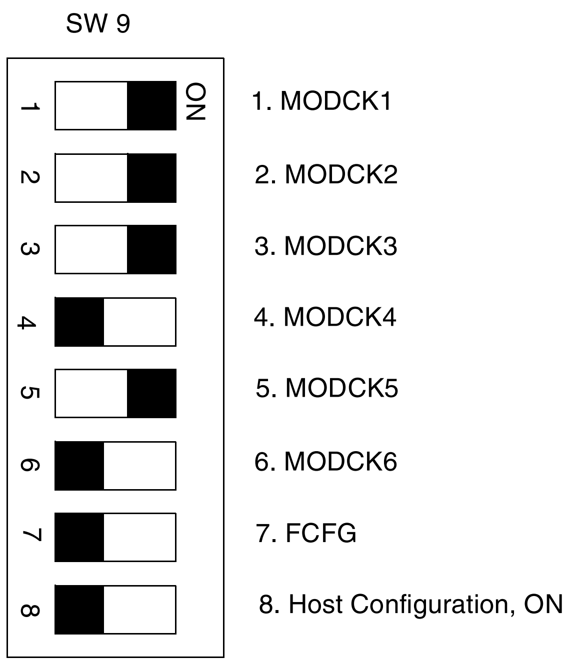

Figure 14. Host-Side MSC8103ADS Switch 9 Settings

- *Boot Mode Select*. Switch 10 is defined in **Figure 15**.

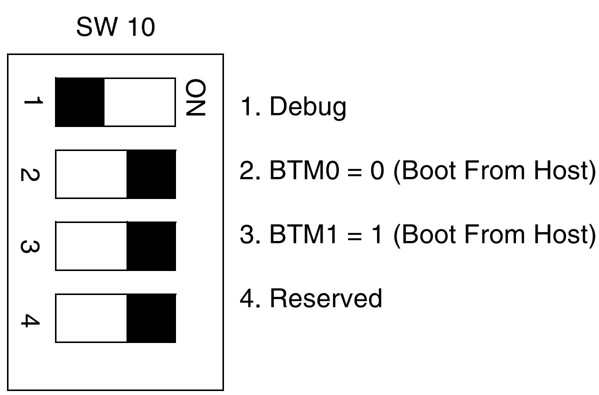

Figure 15. Host-Side MSC8103ADS Switch 10 Settings

- *Software Options*. Switch 11 is defined in **Figure 16**.

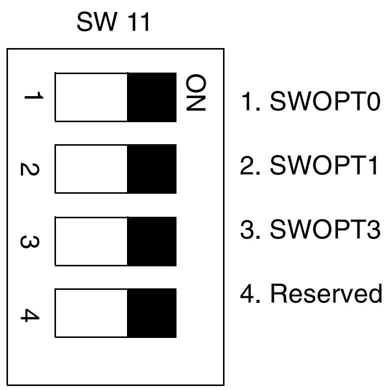

Figure 16. Host-Side MSC8103ADS Switch 11 Settings

3.2 Wiring Diagram

For this application example, two MSC8103ADS boards are interfaced. One MSC8103 device acts as the host and the other acts as the slave. **Figure 17** demonstrates how to wire the two MSC8103ADS boards.

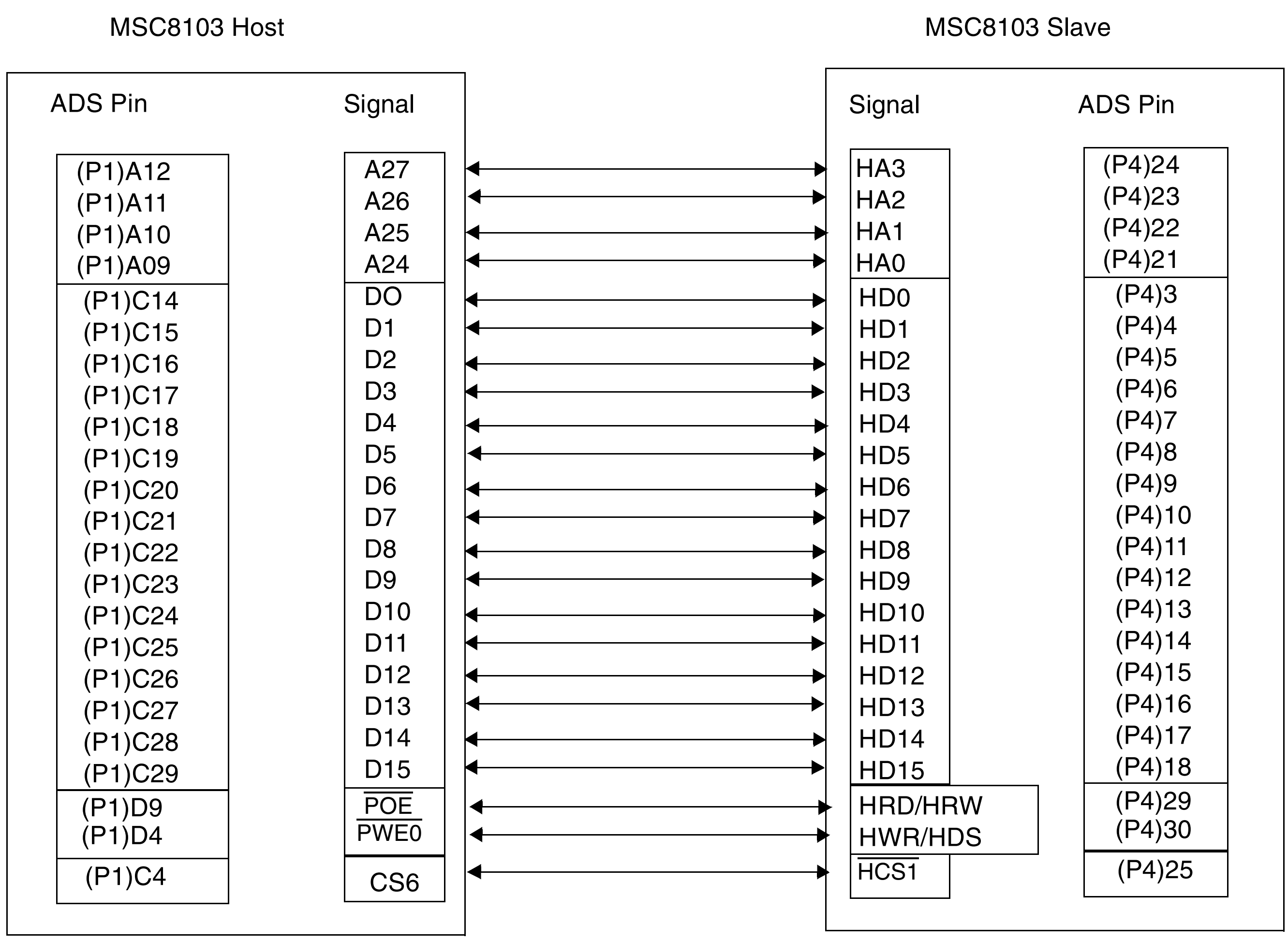

Figure 17. MSC8103 Host to MSC8103 Slave Wiring Diagram

3.3 Source Code For Resetting and Bootloading the Slave

The C code for resetting and bootloading the slave is provided in the software package that accompanies this application note. The code is an example for the 16-bit mode for a slave MSC8103. The host processor used for this application note is the MSC8103. For details, refer to the documentation included in the package.

4 8-Bit Bootload Example

This section explains how to initiate an 8-bit bootload procedure using two MSC8103ADS boards. The 8-bit procedure is similar to the 16-bit bootload procedure, and this section focuses on the differences.

4.1 8-Bit MSC8103ADS Settings

The H8BIT pin must be pulled high to configure the HDI16 for an 8-bit bootload. The only difference between the 16-bit and 8-bit settings is the host interface setting (switch 1). Switch 1 (bit 3) on the slave side must be switched to off for 8-bit mode. This allows the H8BIT pin on the MSC8103 slave side to be pulled high, thus initializing the device for 8-bit mode.

4.2 8-Bit Wiring

For 8-bit operation using the two MSC8103ADS boards, only eight system data bus pins (D[8–15]) are needed to connect to the host interface data pins (HD[8–15]). The wiring diagram for 8-bit implementation is the same as shown in **Figure 17**, except that the host MSC8103 D[0–7] pins connected to the slave MSC8103 HD[0–7] pins are not needed. The HD8 pin is the most significant, and the HD15 pin is the least significant, as is indicated in the *MSC8103 Reference Manual*. The D[0–7] system data bus pins are not used because of the host processor memory controller configuration. Typically, the D[0–7] pins would be used for 8-bit implementation along with an 8-bit port size in the memory controller configuration. In the example source code, however, the memory controller is configured for a 16-bit port size. Instead of rewiring the board for 8-bit bootload operation, the D[8–15] system data bus pins are used with the GPCM configured as a 16-bit port.

4.3 8-Bit Reset Source Code

The reset configuration sequence is exactly the same for 8-bit and 16-bit operation because the hard reset configuration registers are 8 bits each. The HRCW must be written in 8-bit segments to reset the MSC8103 slave using the HDI16. Therefore, the reset code discussed in **Section 3.3** applies to the 8-bit bootload procedure as well as the 16-bit bootload procedure.

4.4 8-Bit Bootloading Source Code

The source code for an 8-bit bootload procedure is similar to the code in the 16-bit bootload example in **Section 3**. The block structure is exactly the same as indicated in **Section 2.2**, *Block Structures*, but the data must be sent 8 bits at a time. Therefore, the 8-bit operation requires twice as many writes or transmits as the 16-bit bootload example.

5 References

[1] *MSC8101 Bootload through the HDI6 port using any host, AN2325D.*

[2] *MSC8103 Reference Manual*, Freescale (MSC8103RM/D).

[3] *MSC8101 Application Development System User's Manual*, Freescale (MSC8101ADSUM/D).

[4] *MSC8101 User's Guide*, Freescale (MSC8101UG/D).

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations not listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GMBH
Technical Information Center
Schatzbogen 7
81829 München, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Technical Information Center
3-20-1, Minami-Azabu. Minato-ku
Tokyo 106-8573, Japan
0120 191014 or +81-3-3440-3569
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T. Hong Kong
+800 2666 8080

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

AN2822
Rev. 0
9/2004