DSP56303EVM - M68HC12A4EVB
HI08 Interface

Ellery A. Blood
October 1999

Index

Introduction:

The Host Interface (HI08) port on the DSP56303EVM is a byte wide parallel port that can be connected directly to the data bus of a host processor. The HI08 supports a variety of buses and provides glueless connection with a number of industry standard devices, including the M68HC12 Microprocessor. This manual shows two of the ways that the 56303 and the HC12 may be connected through this port.

The HI08 registers on the 56303 for the DSP side of the interface are mapped to 8 sequential addresses in the X memory area. These can be accessed through regular MOVE or MOVEP instructions.

HI08 Registers (DSP Side)

Abbreviation	Address	Name
HDR	X:FFFFC9	Host Port GPIO Data Register
HDDR	X:FFFFC8	Host Port GPIO Direction Register
HTX	X:FFFFC7	Host Transmit Register
RHX	X:FFFFC6	Host Receive Register
HBAR	X:FFFFC5	Host Base Address Register
HPCR	X:FFFFC4	Host Port control Register
HSR	X:FFFFC3	Host Status Register
HCR	X:FFFFC2	Host Control Register

The HC12 side of the interface is accessed as 8 memory mapped registers. The location of these registers in memory is determined by the Chip Select (CS) used for the mapping. The default configuration of the HC12EVB allows for the use of chip selects CS0, CS1, CS2, or CS3 in 8-bit mode without affecting the 16-bit mode that it uses for the onboard ROM and RAM.

HI08 Registers (HC12 Side)

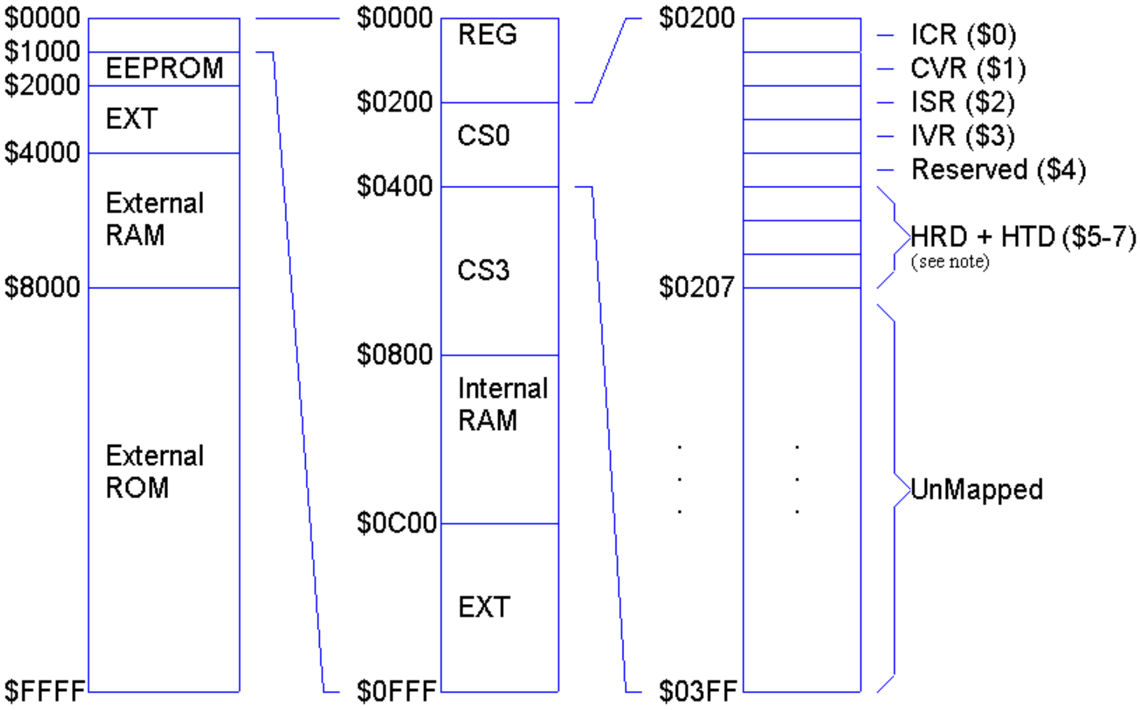
Abbreviation	Address	Name
ICR	0	Interface Control Register
CVR	1	Command Vector Register
ISR	2	Interface Status Register
IVR	3	Interrupt Vector Register
RXH/TXH	5	Receive/Transmit Register High
RXM/TXM	6	Receive/Transmit Register Middle
RXL/TXL	7	Receive/Transmit Register Low

The three 8-bit receive registers RXH, RXM, and RXL make up one 24-bit receive register, HRD. The three 8-bit transmit registers TXH, TXM, and TXL make up one 24-bit transmit register, HTD. HRD is read-only and HTD is write only. This allows both HRD and HTD to use the same 3 addresses. The order of the High, Medium and

Low registers can be reversed by setting or clearing the HLEND (HI08 Little Endian) bit (ICR:bit 5). When HLEND=0, the High registers are at $5 and the Low registers are at $7. When HLEND=1, the High registers are at $7 and the Low registers are at $5. The Middle registers are at $6 regardless of the state of HLEND.

HC12A4 Memory Space:

The memory space on this evaluation board is more than 80% filled with External RAM and ROM and internal Registers, RAM and EEPROM. This leaves 4 areas free for external memory/peripherals without affecting the default configuration. These areas are $0200-$03FF (512 bytes), $0400-$07FF (1Kb), $0C00-$0FFF (1Kb), and $2000-$3FFF (8Kb). The first area can be configured to be addressed in narrow (8-bit) data bus mode while the other areas are addressed in wide (16-bit) mode.



(note: HRD and HTD are each made of 3 registers (High, Medium, and Low). The 3 HRD registers are read-only, the 3 HTD registers are write only. This allows HRD and HTD to share the same 3 addresses.)

(figure 1)

The narrow data bus mode is made possible through using the register-following chip selects with the 512-byte external memory block. This block is specifically designed to allow 8-bit accesses when in 16-bit expanded mode and used in conjunction with the chip selects.

For the use of the HI08 interface, we will be using the 512-byte memory block with CS0 to map the HI08 registers to memory. A graphical representation of the memory map is shown in figure 1.

Basic (glueless) Configuration:

The most basic way of connecting these two processors is to take advantage of the highly configurable characteristics of the HI08 and the HC12 bus. The HI08 can support single/dual data strobe buses, multiplexed/non multiplexed buses, and can configure the polarity of various bus signals. Because of these abilities, it is possible to connect the HI08 directly to the HC12 bus without any additional logic or signal drivers.

In order to implement this interface, the port must be properly configured to match the characteristics of the HC12 Bus in expanded mode. The HC12 bus for this application uses a negative polarity chip select, positive polarity data strobe and non-multiplexed address/data lines. This is done through settings in the Host Port Control Register (HPCR), one of the 8 16 bit HI08 registers[1]. Writing the value $020E to the HPCR configures the port as follows:

HPCR Bit Representation

```
HPCR[0] : 0 GPIO Pins disabled
HPCR[1] : X Host Address Line 8 enabled
HPCR[2] : X Host Address Line 9 enabled
HPCR[3] : 1 Host Chip Select (Address Line 10) enabled
HPCR[4] : 0 Host Request disabled
HPCR[5] : 0 Host Acknowledge disabled
HPCR[6] : 0 Host Port disabled (will be set to 1 later)
HPCR[7] : 0 Reserved[2]
HPCR[8] : X Not Applicable[3]
HPCR[9] : 1 Host Data Strobe Polarity active high
HPCR[10]: X Not Applicable
HPCR[11]: 0 Host Multiplexed Bus disabled
HPCR[12]: 0 Host Dual Data Strobe disabled
HPCR[13]: 0 Host Chip Select Polarity active low
HPCR[14]: X Not Applicable
```

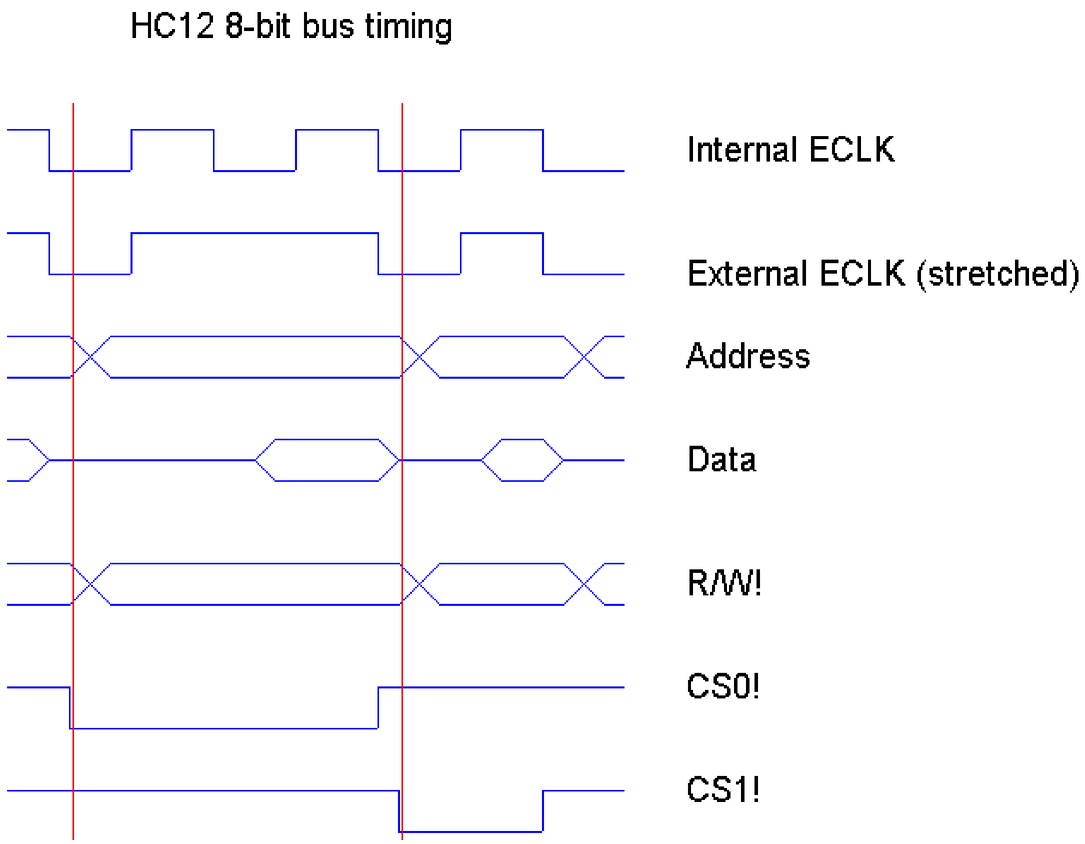
[1] Because the registers are 16 bits long in a 24-bit architecture, the most significant 8 bits should be set to 0. For example, to set a register to $1234, the value $001234 would be written to the location of the register.

[2] Reserved bits should be set to 0 for future compatibility.

[3] Bits marked with an X are not applicable to the current configuration and are set to 0.

```
HPCR[15]: X Not Applicable
```

The HC12 bus configuration has several options specifically designed to accommodate slow pheriperals/memory. Figure 2 shows the bus timing for two read/write operations using two separate chip selects. The first operation utilizes CS0 and has a 1 ECLK cycle stretch. The second operation utilizes CS1 and has no cycle stretching on the enable (External ECLK). Each chip select can use a cycle stretch of 0,1,2 or 3 ECLK cycles. The length of the cycle stretch is determined by the CSSTR0 and CSSTR1 registers. These registers have two bits (A and B) for each chip select, which represents a 2-bit binary number specifying the number of cycles to stretch.



(figure 2)

The HC12 bus can be configured to enable CS0 and the 8-bit data bus mode. CS0 is enabled by setting the CS0E bit (CSCTL0:bit 0) by setting memory location $003C to $31. The memory that CS0 selects is set to use the 8-bit data bus instead of 16-bit by setting the NDRC (MISC:bit 5) bit by setting memory location $0013 to $40.

With the HI08/HC12 interface configured this way, the HC12 bus may be connected to the HI08 as shown in Appendix: A. The 8 registers are now mapped to the bottom 256 bytes of the 512-byte memory block. The 8 registers are repeated throughout the block because the extra address lines (lines A3-A8) are not decoded.

This configuration acts only as a basic data transfer medium. The HI08 is capable of many additional functions including DMA, interrupt triggering, more advanced handshaking, and other features. The more advanced features will not be discussed in this document.

Full Address Decoding Configuration:

By adding some external logic, the basic configuration can be enhanced to only take up only the 8 bytes of memory instead of the full 256. This is accomplished by using the CS0 signal and A3 - A8 address lines from the HC12 to activate the Chip Select on the DSP56303. One way to do this is by using a 74LS688 eight-bit magnitude comparator. In order for the DSP registers to be mapped to only 0200 - 0207 on the HC12, the CS should receive a logic low only when lines A3-A8 are low and CS0 is low. This is accomplished by connecting CS0 to P0, and A3-A8 to P1-P6 and P7 to ground. Connect Q0-Q7 to ground. When the signals match, the P=Q! output on the comparator will be asserted low. This is passed to the CS input. The wiring for this is shown in Appendix: B.

Transferring Data:

In order to transfer data through the HI08, both sides of the interface have read-only registers that tell the status of information passing through. On the DSP side, the Host Status Register (HSR) has 2 read-only bits (HRDF and HTDE) that are important to this configuration.

HSR Bit 0, Host Receive Data Full (HRDF) tells the status of received information. When the HRDF is 0, the data is not yet ready to be read. When the HRDF is 1, the data is ready to be read. The HRDF is set whenever the host writes to the least significant byte of the transmit registers. It is cleared when the HRX register is read by the DSP.

HSR Bit 1, Host Transmit Data Empty (HTDE) tells the status of sent information. When HTDE is 0, the HC12 has not yet read the information, when HTDE is 1, the HC12 has read the information and it is safe to send more data.

On the HC12 side of the interface, the Interrupt Status Register (ISR) contains 2 read-only bits (RXDF and TXDE) that are important to this configuration. ISR Bit 0 is the Receive Data Register Full (RXDF) flag. When the RXDF is 1, the HC12 may read from the data registers. When the RXDF is 0, the HC12 should wait before reading data. The RXDF is set whenever the HTX is written to, and cleared when the HC12 reads from the data registers.

ISR Bit 1 is the Transmit Data Register Empty (TXDE) flag. When this flag is 1, it is safe to write to the data registers. When RXDE is 0, the HC12 should wait for the DSP to read the data. This flag is set whenever the HC12 writes to the least significant byte of the data registers, and cleared when the DSP reads from the HTX register.

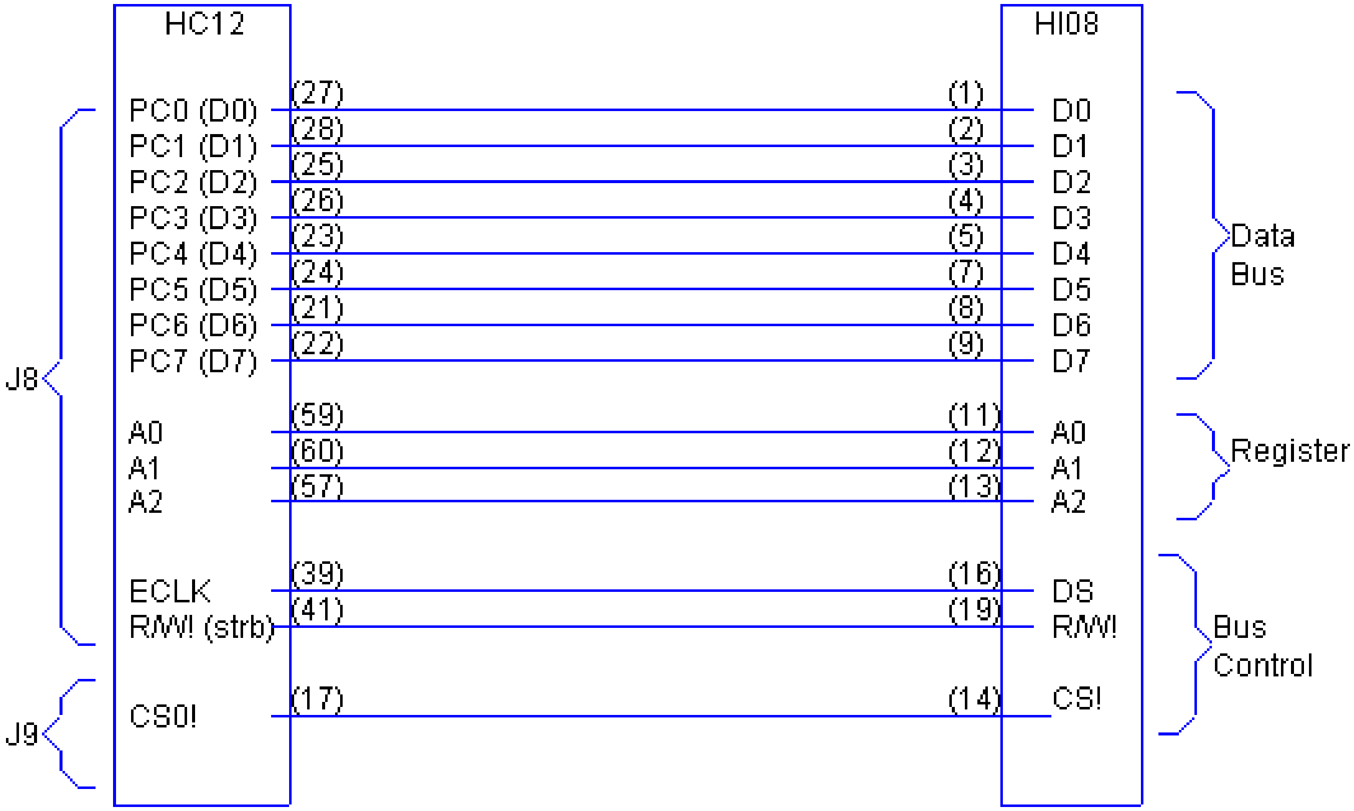
The DSP side of the data transfer is accomplished through 2 24-bit registers, HTX (write only) and HRX (read only). To transmit data, a 24-bit word is written to the HTX after checking the status bits to make sure it is safe. To receive data, a 24-bit word is read from the HRX after checking the status bits to make sure it is valid.

The HC12 side of the data transfer is accomplished through 3 8-bit registers. These 3 registers are actually 6 registers mapped to the same memory location. The 3 receive registers (RXH, RXM, RXL) are all read-only. The 3 transmit registers (TXH, TXM, TXL) are all write-only. To send data, the most significant byte should be written to TXH, the middle should be written to TXM, and the least should be written to TXL after first checking the status bits to ensure that there is not already data waiting to be read. TXL should be written to last to ensure all the data is present before the DSP receives it. To receive data, the top, middle, and bottom bytes are read from TRH, TRM, and TRL respectively after checking the status bits for valid data.

Appendix: A - Wiring Diagrams

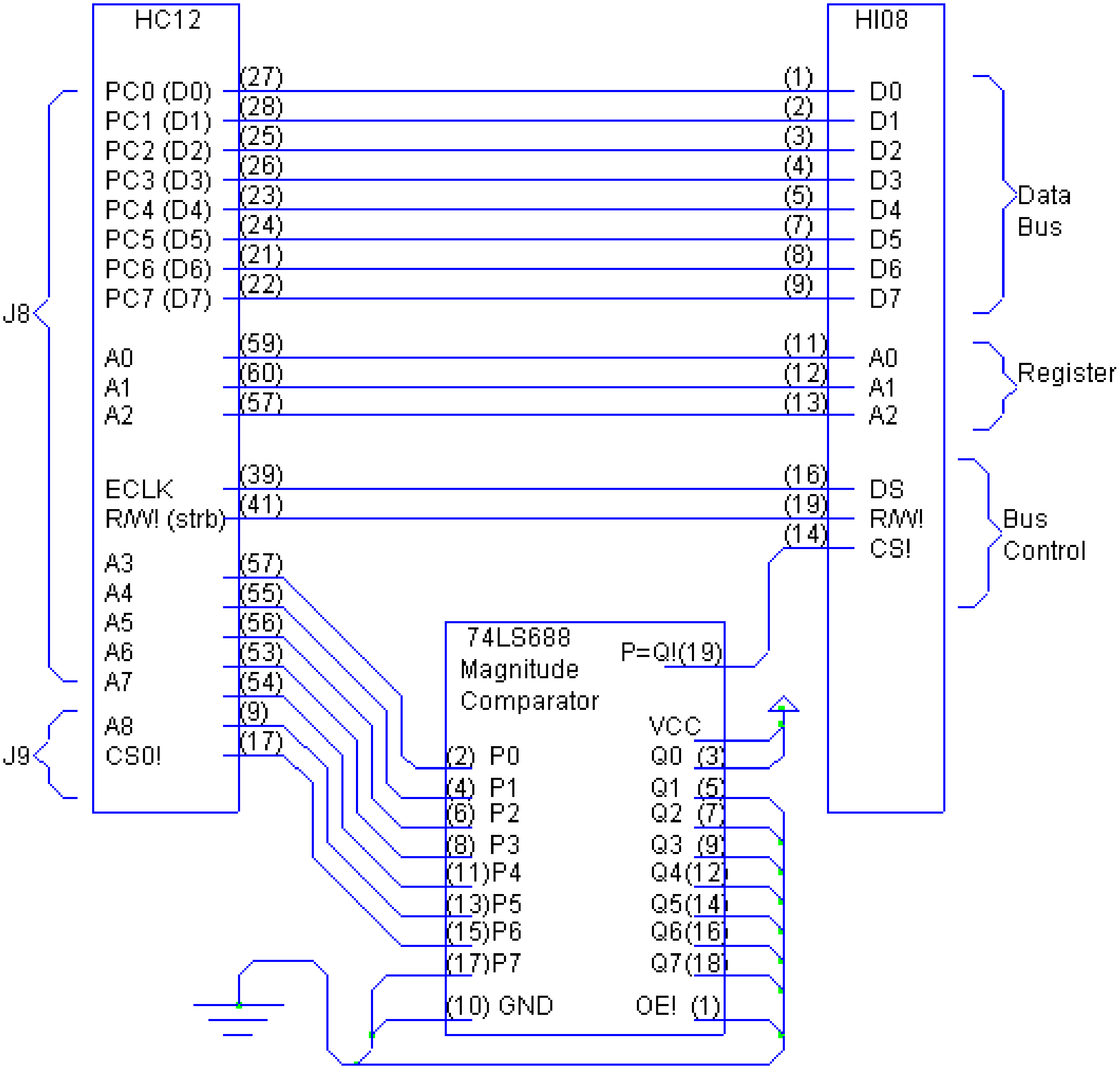
Basic (glueless) Configuration

Wiring Diagram



Full Address Decoding Configuration

Wiring Diagram



Appendix: B - HI08 Walkthrough

The following is a step by step example of how the HI08 functions by using the D-Bug12 Monitor on the HC12 and the Debugger on the DSP.

Setup

1. If not already installed, install the EVM563xx debugger from the Suite56 CD.

2. Connect the DSP and HC12 as shown in Appendix A.

3. Connect the left 9-pin serial port (DEBUG) on the DSP to the COM1 port on the PC with an RS232 serial cable.

4. Power on the DSP.

5. Open the debugging software for the DSP (EVM30xw.exe) that comes with the EVM. If the **Data Display** and **Command** windows are not already open, open them by selecting **Data** and/or **CMD** respectively from the View menu.

6. Click the Stop button in the debugger.

The following commands are to be typed into the Command window on the debugger.

7. Set the input radix to hexadecimal
   ```
   RADIX hex
   ```

8. Configure the HI08 for HC12 interfacing by writing $0208 to the HPCR.
   ```
   CHANGE x:ffffc4 000208
   ```

9. Connect the HC12 to COM2 port with an RS232 serial cable.

10. Power on the HC12

11. Open a terminal window to the HC12 (Hyper terminal with a 9600-baud connection set for "direct to COM2" works well).

12. Set the ECLK cycle stretch for CS0 to stretch 2 cycles.
    ```
    MM 003F
    02
    ```

13. Set the register following chip selects to use narrow (8-bit) bus data mode.
    ```
    MM 0013
    40
    ```

14. Enable CS0

```
MM 003C
31
```

The interface is now configured, and the HI08 registers are mapped to $0200-$0207 on the HC12.

15. Set the HI08 Enable bit in the HPCR.

```
CHANGE x:ffffc4 000248
```

Transferring Data to the HC12

1. Display the memory area on the HC12.

```
MD 0200
```

2. Write 24-bit data to HTX (i.e. $123456)

```
CHANGE x:ffffc7 123456
```

You will receive the warning message :

```
Write Error at X:FFFFC7: Wr-00123456 Rd-00000000
```

This is normal since the register is write only and the debugger reads the value and compares it directly following a write to confirm correct operation.

3. Display the new data on the HC12

```
MD 0200
```

You should notice that the memory in locations 0205, 0206, and 0207 are now $12, $34, and $56 respectively.

Transferring Data to the DSP

1. Display the memory area on the DSP.

```
Display x:ffffc2
```

2. Write 24-bit data (i.e. $abcdef) to the 3 8-bit registers

```
MM 0205
ab
MM 0206
cd
MM 0207
ef
```

You will receive the warning message :

```
Can't write target memory
```

This is normal because the same addresses are used for receiving as for transmitting. So, you will be unable to read what was just written.

3. Display the new data on the DSP

```
REFRESH
```

You should notice that the memory in location x:ffffc6 (HRX) is now $ABCDEF.

Appendix: C - Resources

DSP56303 Users Manual (DSP56303UM/AD), Motorola inc. 1996

DSP56300 24-bit Digital Signal Processor Family Manual, Motorola inc.

68HC12 Evaluation Board User's Manual, Motorola inc. 1996

68Hc12 CPU12 Reference Manual, Motorola inc. 1996,1997

MC68HC12A4 16-Bit Microcontroller Technical Summary, Motorola inc. 1997