DSP56303EVM - M68HC11EVBU
HI08 Interface

Ellery A. Blood
September 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 M68HC11 Microprocessor. This manual shows a few of the ways which the 56303 and the HC11 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

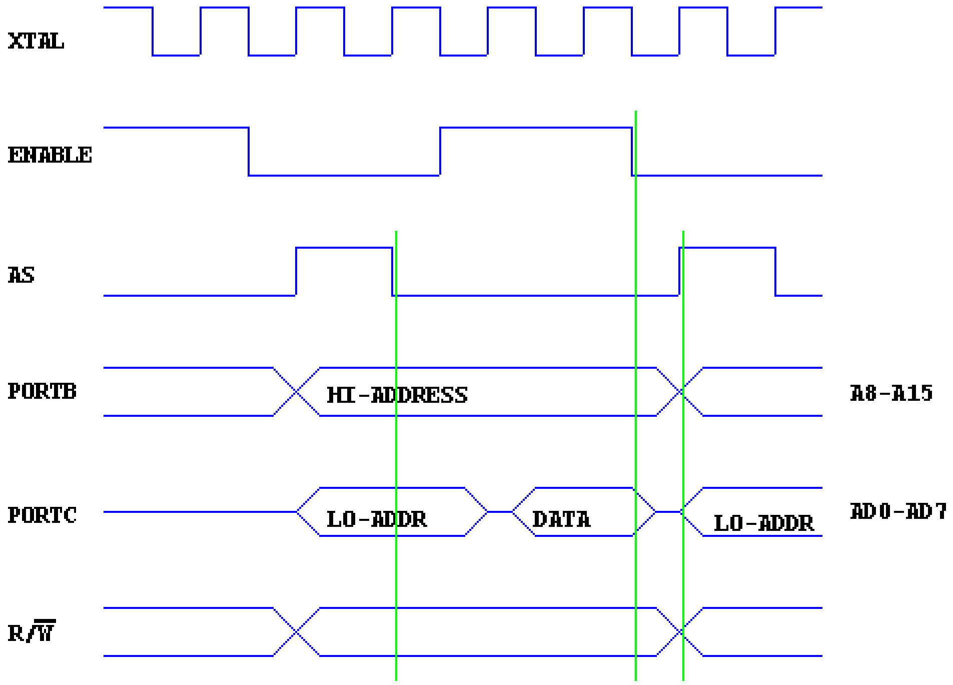
The HC11 side of the interface is accessed as 8 memory mapped registers. The location of these registers in memory is determined by the value set in the Base Address Register (HBAR). When the 8 most significant bits of the address are equal to the value stored in HBAR, the HI08 activates an internal chip select which allows the 3 least significant bits to specify which of the 8 registers are to be written/read. These registers can be accessed with the MOVE command.

HI08 Registers (HC11 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

Basic (glueless) Configuration:

The most basic way of connecting these two processors is to take advantage of the highly configurable characteristics of the HI08. 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 HC11 bus (figure 1) without any additional logic or signal drivers.



(figure 1)

In order to implement this interface, the port must be properly configured to match the characteristics of the HC11 Bus in expanded mode. This is done through settings in the Host Port Control Register (HPCR), one of the 8 16 bit HI08 registers[1]. Writing the value $2E0E to the HPCR configures the port as follows:

HPCR Bit Representation

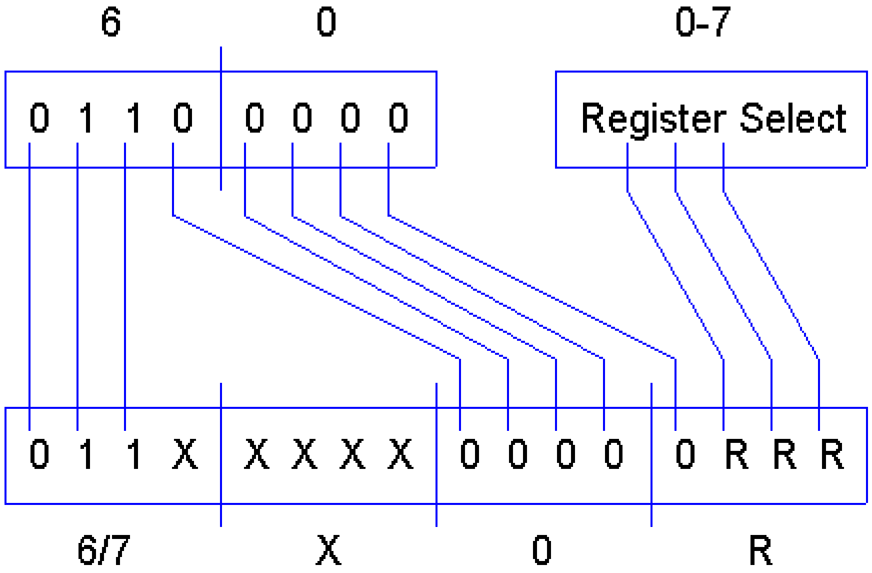
```
HPCR[0] : 0 GPIO Pins disabled
HPCR[1] : 1 Host Address Line 8 enabled
HPCR[2] : 1 Host Address Line 9 enabled
HPCR[3] : 1 Host Chip Select (Address Line 10) enabled
HPCR[4] : 0 Host Request disabled
```

[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.

```
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]: 1 Host Address Strobe Polarity active high
HPCR[11]: 1 Host Multiplexed Bus enabled
HPCR[12]: 0 Host Dual Data Strobe disabled
HPCR[13]: 1 Host Chip Select Polarity active high
HPCR[14]: X Not Applicable
HPCR[15]: X Not Applicable
```

With the HI08 configured this way, the HC11 bus may be connected to the HI08 as shown in Appendix: A. 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.

The operation of the HI08 is accomplished as follows. Initially a base address is written to the HBAR register. We will use $60 for this example. As shown in figure 2, several address lines are left unconnected (denoted by X's). This leads to the effect that the registers may be accessed from multiple base addressed, $6000, $6100, $6200,



(figure 2)

[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.

$6300... $7E00, $7F00. In each of these locations, the registers are addressed by their number relative to the base address. ICR is at $6000, IVR is at $6003, etc. This takes up 8 KB of address space for only 8 registers, but saves us from having to build any external hardware logic.

Full Address Decoding Configuration:

By adding some external logic, the basic configuration can be enhanced to only take up only 8 bytes of memory instead of the 2KB. This is accomplished by using the E(enable) signal and A8 - A12 signals from the HC11 to activate the CS (chip select) on the DSP56303. One way to do this is by using a 74LS688 eight-bit magnitude comparator and an 74LS04 inverter. In order for the DSP registers to be mapped to only 6000 - 6008 on the HC11, the CS should receive a logic high only when lines A8-A12 are low and E is high. This is accomplished by connecting E to P0, and A8-A12 to P1-P5 and P6 - P7 to ground. Connect Q0 to power and Q1-Q7 to ground. When the signals are equal, the P=Q! output on the comparator will be asserted low. This is passed through the inverter and sent to the CS input. This is shown in Appendix: B.

Full address decoding can also be accomplished without the inverter by changing the configuration on the DSP so that the Chip Enable is triggered low instead of high. This setting is on bit 13 of the HPCR (0=active low, 1-active high) in the DSP configuration.

Transferring Data:

In order to transfer data through the HI08, both sides of the interface have a read-only register 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 HC11 has not yet read the information, when HTDE is 1, the HC11 has read the information and it is safe to send more data.

On the HC11 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 HC11 may read from the data registers. When the RXDF is 0, the HC11 should wait before reading data. The RXDF is set whenever the HTX is written to, and cleared when the HC11 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 HC11 should wait for the DSP to read the data. This flag is set whenever the HC11 writes to the least significant byte of the data registers, and cleared when the DSP reads from the HTX register.

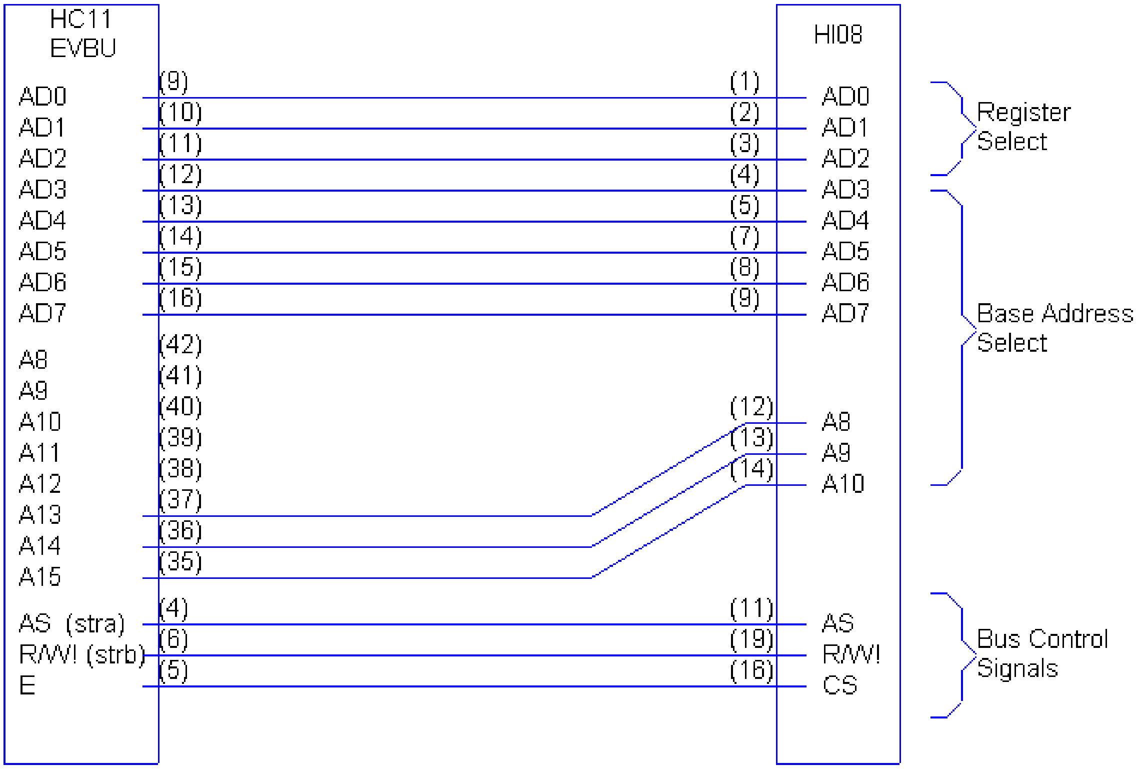
The DSP side of the data transfer is accomplished through 2 24-bit registers, HTX (write only) and HRX (read only). To transmit data, a 24-bit word is written to the HTX after checking the status bits to make sure it is safe. To receive data, a 24-bit word is read from the HRX after checking the status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

The HC11 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

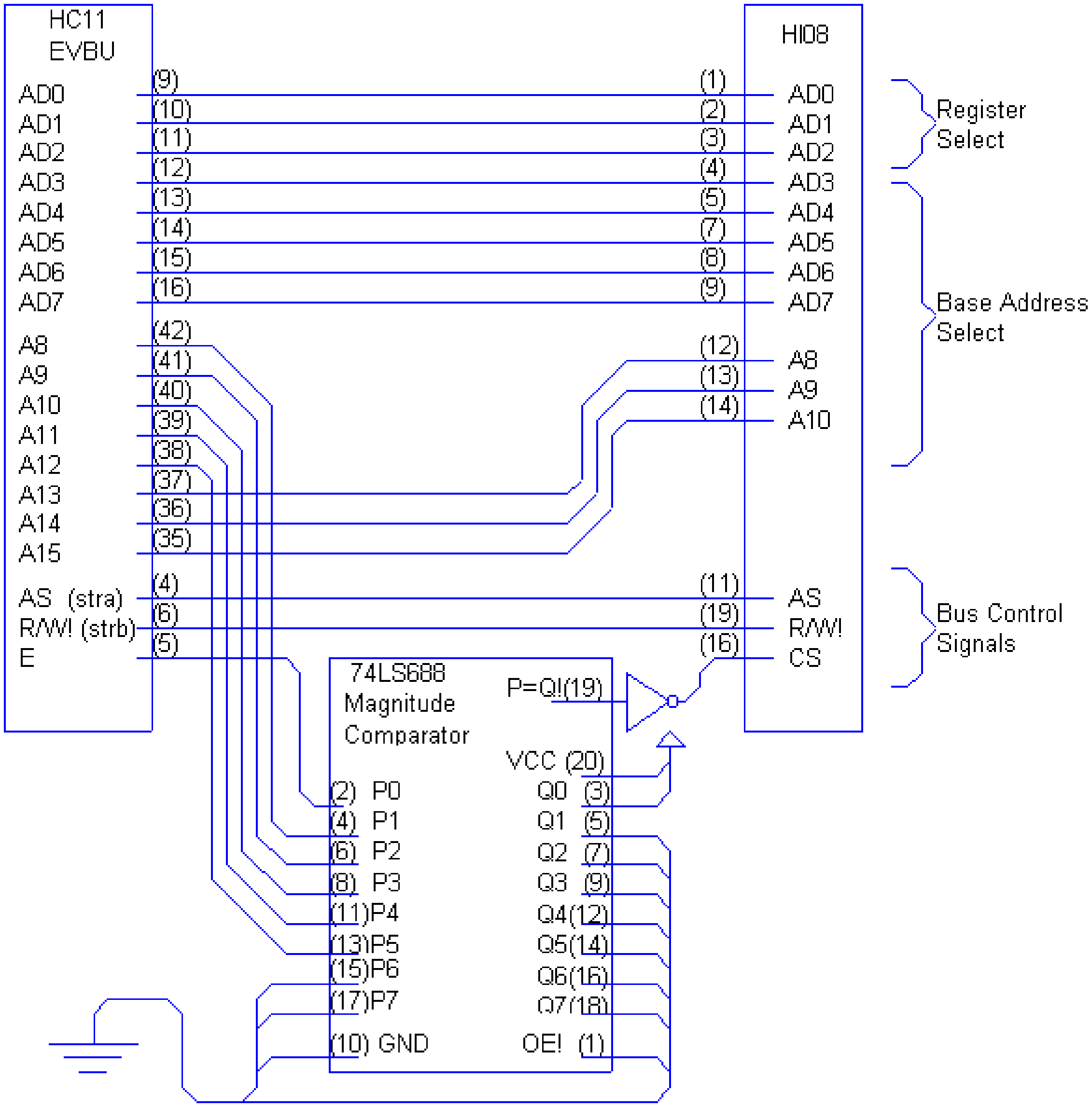
Basic (glueless) Configuration

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 BUFFALO Monitor on the HC11 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 HC11 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 HC11 interfacing by writing $2E0E to the HPCR.

```
CHANGE x:ffffc4 002e0e
      or (for Full Address w/o the inverter)
CHANGE x:ffffc4 000e0e
```

9. Set the base address for the HC11 bus by writing $60 to the HBAR.

```
CHANGE x:ffffc5 000060
```

10. Connect the HC11 to COM2 port with an RS232 serial cable.

11. Power on the HC11

12. Open a terminal window to the HC11 (Hyper terminal with a 9600-baud connection set for "direct to COM2" works well).

The interface is now configured, and the HI08 registers are mapped to $6000-$6007 on the HC11.

13. Set the HI08 Enable bit in the HPCR.

```
CHANGE x:ffffc4 002e4e
      or (for Full Address w/o the inverter)
CHANGE x:ffffc4 000e4e
```

Transferring Data to the HC11

1. Display the memory area on the HC11.

```
MD 6000
```

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 HC11

```
MD 6000
```

You should notice that the memory in locations 6005, 6006, and 6007 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 6005
ab
MM 6006
cd
MM 6007
ef
```

You will receive the warning message :

```
rom-
```

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.

H68HC11EVBU Universal Evaluation Board User's Manual
(M68HC11EVBU/AD2)