AN2567
Rev. 1, 12/2004

MSC8101 HDI16 Throughput Performance Comparisons

By Jason Streeter

The Freescale MSC8101 digital signal processor (DSP) device features a 16 bit wide asynchronous slave peripheral HDI16 port for use by a host processor to transfer data for further processing. This flexible parallel port handles fast asynchronous transfers from a host to the internal HDI16 registers. The MSC8101 device can service the throughput data via the DMA controller, interrupts, or the SC140 core. This application note compares the throughput performance capabilities and limitations of each of these means of servicing the HDI16 port.

CONTENTS

1 HDI16 Basics

The HDI16 port can operate in either 16-bit or 8-bit mode. In addition to the data lines, there are four address signals to access the eleven HDI16 registers available to the host. Two chip selects, are available for use by a host processor in a DSP farm either to broadcast or to individually select a slave HDI16 port. Finally, there are four signals that control the asynchronous transactions. The HDI16 port gluelessly supports a variety of host processors. The host can communicate with the HDI16 port via a dual- or single-strobe access. The host can also use the DMA controller to transfer data to the HDI16 port, taking advantage of the HDI16 DMA request and acknowledge support. Host processor DMA controllers typically have low latency between accesses, allowing for maximum HDI16 performance. This document focuses on the throughput capabilities of the 16-bit HDI16 mode, which is affected by the following parameters and conditions:

- *Host processor memory interface timings*. Relaxing the timings significantly reduces throughput, and optimizing the host timings maximizes the performance.
- *Frequency*. The faster the frequency of the SC140 core, the faster the HDI16 port operates.
- *Method of service*. Throughput may vary if the SC140 core, rather than the DMA controller or an HDI16 interrupt, moves the data from the HDI16 port to the memory.

2 HDI16 Connections and Timings

The throughput calculations discussed here are based on HDI16 timings obtained from a logic analyzer. These calculations take the asynchronous connections and timings into account.

2.1 MSC8101 System Bus to MSC8101 HDI16 Connection

The asynchronous throughput measurements in this application note were obtained using two MSC8101ADS boards. The MSC8101 host processor connects to the MSC8101 HDI16 port as shown in **Figure 1**. The 60x-compatible system bus of the MSC8101 host processor connects to the MSC8101 HDI16 slave port as if it were an external memory.

2.2 MSC8101 HDI16 Timings

The HDI16 port operation differs, depending on whether the host is performing reads or writes. The timings for dual-strobe and single-strobe methods of communication are similar. This application note references only dual-strobe mode since this is the mode used by this application to collect the throughput data for the HDI16 port. In single-strobe mode, a data strobe signal qualifies the access, and an HRW signal specifies whether the access is a read or a write. Dual-strobe mode uses two separate signals to specify whether the access is a read or a write. This section illustrates the differences between reads and writes for dual-strobe accesses.

2.2.1 HDI16 Read Timings

Figure 2 illustrates the timings for an HDI16 read as indicated in the *MSC8101 Technical Data* sheet. When the memory controller on the host processor is programmed, the HDI16 timings must be met to complete the transaction properly. For a read access, the address of the HDI16 register must first be driven on HA[0–3]. Then the host issues a chip select to enable the HDI16 port of the MSC8101 slave. The host continues the transaction by asserting the $\overline{\text{HRD}}$ signal to request the read transaction and then later deasserting $\overline{\text{HRD}}$ after the data has been sampled, thus completing the transaction.

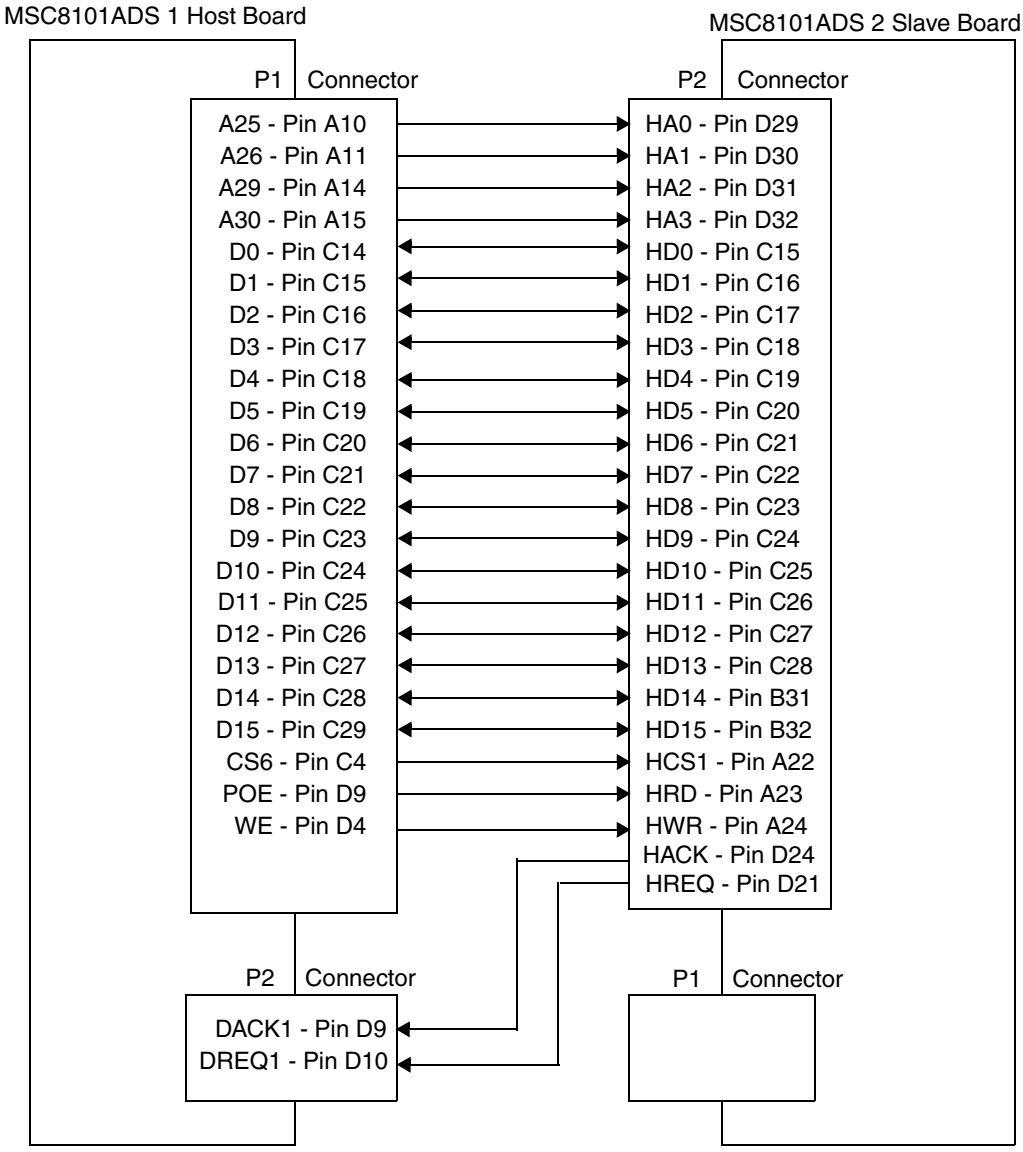

Figure 1. MSC8101ADS(1) System Bus to MSC8101ADS(2) HDI16 Connection

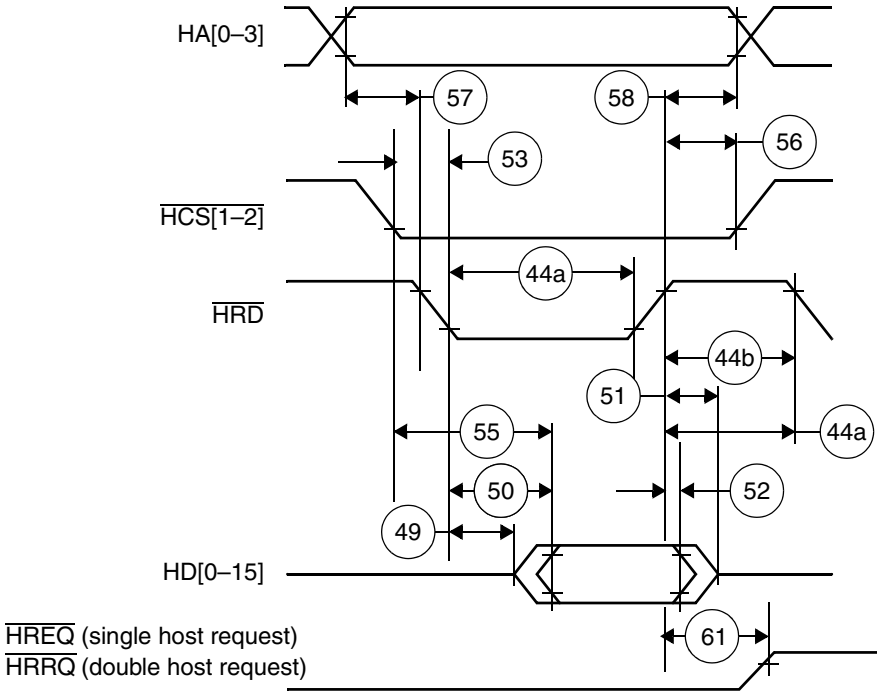

Figure 2. HDI16 Read Timings (Dual-Strobe Mode)

2.2.2 HDI16 Write Timings

Figure 3 shows the timings for the HDI16 write transaction as indicated in the *MSC8101 Technical Data* sheet. When the host memory controller is programmed, the MSC8101 HDI16 write timing requirements must be met for proper execution. A write transaction is almost exactly the same as a read transaction. For a write transaction, the address of the HDI16 register must be driven first. As with the read transaction, the host processor drives the chip select to enable the HDI16 of the slave MSC8101. Then the host asserts the HWR signal to begin the write transaction. After the data is latched into the HDI16, the HWR signal is deasserted to terminate the transaction.

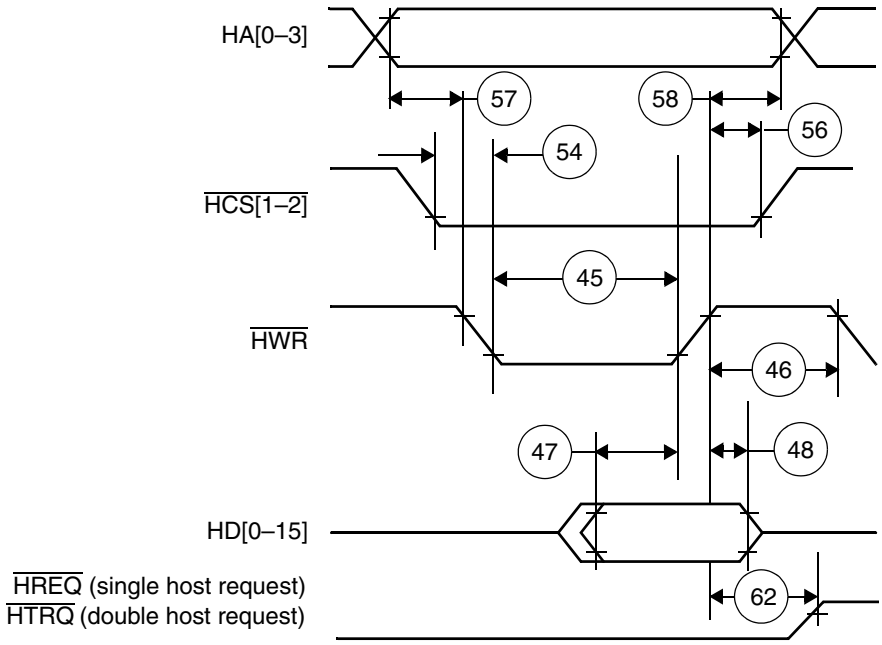

Figure 3. HDI16 Write Timings (Dual-Strobe Mode)

2.2.3 HDI16 Timing Values

The timing values associated with the preceding diagrams are listed in **Table 1**. These values are taken directly from the *MSC8101 Technical Data* sheet and are based on T_c, which is the inverse of the core frequency. In this application note, the T_c value is 3.33 ns, which corresponds to 300 MHz. These timings are used to obtain the theoretical throughput values in **Section 3**, *Theoretical Throughput,* on page 6.

Table 1. Host Interface (HDI16) Timing[1]

Number	Characteristics[3]	Expression	Value	Unit
44a	Read data strobe minimum assertion width[4] $\overline{\text{HACK}}$ read minimum assertion width	$(1.5 \times T_C) + 5.0$	9.95	ns
44b	Read data strobe minimum deassertion width[4] $\overline{\text{HACK}}$ read minimum deassertion width	$T_C + 5.0$	8.3	ns
44c	Read data strobe minimum deassertion width[4] after "Last Data Register" reads[5,6], or between two consecutive CVR, ICR, or ISR reads[7] $\overline{\text{HACK}}$ minimum deassertion width after "Last Data Register" reads[5,6]	$(2.5 \times T_C) + 5.0$	13.25	ns
45	Write data strobe minimum assertion width[8] $\overline{\text{HACK}}$ write minimum assertion width	$(1.5 \times T_C) + 5.0$	9.95	ns
46	Write data strobe minimum deassertion width[8] $\overline{\text{HACK}}$ write minimum deassertion width after ICR, CVR and Data Register writes[5]	$(2.5 \times T_C) + 5.0$	13.25	ns
47	Host data input minimum setup time before write data strobe deassertion[8] Host data input minimum setup time before $\overline{\text{HACK}}$ write deassertion	—	5.0	ns
48	Host data input minimum hold time after write data strobe deassertion[8] Host data input minimum hold time after $\overline{\text{HACK}}$ write deassertion	—	5.0	ns
49	Read data strobe minimum assertion to output data active from high impedance[4] $\overline{\text{HACK}}$ read minimum assertion to output data active from high impedance	—	5.0	ns
50	Read data strobe maximum assertion to output data valid[4] $\overline{\text{HACK}}$ read maximum assertion to output data valid	$(2.0 \times T_C) + 5.0$	11.6	ns
51	Read data strobe maximum deassertion to output data high impedance[4] $\overline{\text{HACK}}$ read maximum deassertion to output data high impedance	—	5.0	ns
52	Output data minimum hold time after read data strobe deassertion[4] Output data minimum hold time after $\overline{\text{HACK}}$ read deassertion	—	5.0	ns
53	$\overline{\text{HCS}}$[1–2] minimum assertion to read data strobe assertion[4]	—	5.0	ns
54	$\overline{\text{HCS}}$[1–2] minimum assertion to write data strobe assertion[8]	—	5.0	ns
55	$\overline{\text{HCS}}$[1–2] maximum assertion to output data valid	$T_C + 5.0$	8.3	ns
56	$\overline{\text{HCS}}$[1–2] minimum hold time after data strobe deassertion[9]	—	0.0	ns
57	HA[0–3], HRW minimum setup time before data strobe assertion[9] • Read • Write	—	 0 5.0	 ns ns
58	HA[0–3], HRW minimum hold time after data strobe deassertion[9]	—	5.0	ns
61	Maximum delay from read data strobe deassertion to host request deassertion for "Last Data Register" read[4, 5, 10]	$(3.5 \times T_C) + 5.0$	16.55	ns
62	Maximum delay from write data strobe deassertion to host request deassertion for "Last Data Register" write[5,8,10]	$(2.5 \times T_C) + 5$	13.25	ns

Table 1. Host Interface (HDI16) Timing[1] (Continued)

Number	Characteristics[3]	Expression	Value	Unit
63	Minimum delay from DMA $\overline{\text{HACK}}$ (OAD=0) or Read/Write data strobe(OAD=1) deassertion to $\overline{\text{HREQ}}$ assertion.	$(4.0 \times T_C) + 5.0$	18.2	ns
64	Maximum delay from DMA $\overline{\text{HACK}}$ (OAD=0) or Read/Write data strobe(OAD=1) assertion to $\overline{\text{HREQ}}$ deassertion	$(3.5 \times T_C) + 5.0$	16.55	ns

NOTES:
1. T_C = 1/ DSPCLK. At 300 MHz, T_C = 3.3 ns
2. V_{CC} = 3.3 V ± 0.3 V; T_J = –40°C to +100 °C, C_L = 50 pF
3. The read data strobe is $\overline{\text{HRD}}$/HRD in the dual data strobe mode and $\overline{\text{HDS}}$/HDS in the single data strobe mode.
4. In 64-bit mode, The “last data register” is the register at address $7, which is the last location to be read or written in data transfers. This is RX0/TX0 in the little endian mode (HBE = 0), or RX3/TX3 in the big endian mode (HBE = 1).
5. This timing is applicable only if a read from the “last data register” is followed by a read from the RXL, RXM, or RXH registers without first polling RXDF or HREQ bits, or waiting for the assertion of the $\overline{\text{HREQ}}$/HREQ signal.
6. This timing is applicable only if two consecutive reads from one of these registers are executed.
7. The write data strobe is HWR in the dual data strobe mode and HDS in the single data strobe mode.
8. The data strobe is host read ($\overline{\text{HRD}}$/HRD) or host write ($\overline{\text{HWR}}$/HWR) in the dual data strobe mode and host data strobe ($\overline{\text{HDS}}$/HDS) in the single data strobe mode.
9. The host request is $\overline{\text{HREQ}}$/HREQ in the single host request mode and $\overline{\text{HRRQ}}$/HRRQ and $\overline{\text{HTRQ}}$/HTRQ in the double host request mode. $\overline{\text{HRRQ}}$/HRRQ is deasserted only when HOTX fifo is empty, $\overline{\text{HTRQ}}$/HTRQ is deasserted only if HORX fifo is full (treat as level Host Request).

3 Theoretical Throughput

When actual throughput measurements are obtained, it is good practice to have a theoretical value for comparison. The HDI16 read transaction, which is programmed into the host MSC8101 UPM memory controller, requires 5 cycles with a system bus clock at 100 MHz. The throughput is calculated by adding the cycle time of the UPM transaction and applying the result to Equation 1. The transaction starts when the UPM asserts the address lines and ends when the UPM deasserts the read or write strobe. For a read transaction, the throughput is 38.37 MB/s. The HDI16 write transaction, which is also programmed into the host MSC8101 UPM memory controller, requires 4 cycles with a system bus clock at 100 MHz. The throughput is calculated the same way as the read transaction, by adding the cycle time of the UPM transaction and applying the result to **Equation 1**. For a write transaction, the throughput is 47.96 MB/s.

Equation 1

$$T = (N) / (t \times M)$$

where:

- T = Throughput
- N = Number of bytes transferred
- t = Time of transaction
- $M = 2^{20} = 1{,}048{,}576$

4 Actual Throughput

The actual throughput measurements for the HDI16 port depend on such conditions as the HDI16 data bus width, mode of operation, register definitions, and single or host DMA accesses.

4.1 Throughput Conditions

The conditions, modes, and register settings used to obtain the HDI16 throughput performance measurements discussed in this section are as follows:

- Host MSC8101ADS board set-up includes an MSC8101 processor, mask set 1K42M.
- Slave MSC8101ADS board set-up includes an MSC8101 processor, mask set 1K42M.
- The HDI16 operates in 64-bit mode and dual-strobe mode with a data bus that is 16 bits wide.
- The host and slave MSC8101 devices operate in mode 8 with a 25 MHz crystal. This yields a system bus frequency of 100 MHz and a core clock speed of 300 MHz.
- Additional loads on the system bus include SDRAM, Flash memory, and buffers.
- Host DMA and single read and write accesses are back-to-back. Host DMA means that the MSC8101 host uses its DMA controller to transfer data with better performance for back-to-back transfers.
- The pipeline maximum depth bit has a value of 0 in the host MSC8101 Bus Control Register (BCR) for a pipeline depth of 1.
- Throughput measurements for host DMA accesses are taken over an average of 32 bytes (16 transactions).
- Throughput measurements for single accesses are taken over an average of 32 bytes (16 transactions).
- Only one MSC8101 is used to gather the data.

4.2 HDI16 Read Throughput

This section discusses the three different ways the slave MSC8101 can transmit data to its HDI16:

- *SC140 core*. The SC140 core uses move instructions to transfer data from memory to the HDI16 TX registers. The SC140 core must poll the HDI16 flags to determine when the data can be transmitted to the HDI16. This type of service is called the slave poll method.
- *DMA controller*. The DMA can be initialized so that when the HDI16 is empty, the DMA controller automatically sends data from internal memory to the HDI16. This type of service is called the slave DMA method.
- *HDI16 interrupts*. When the HDI16 TX registers are empty, an interrupt occurs and the SC140 core can move the data from the internal memory or general-purpose registers to the HDI16. This type of service is called the slave HDI16 interrupt.

4.2.1 Single Read Throughput, Slave Poll Method

The maximum HDI16 single read throughput with the slave polling the HDI16 flags is based on the accesses to the internal HDI16 registers. **Figure 4** shows the timings obtained from the logic analyzer. The host reference clock is the MSC8101 system bus clock by which the UPM controls the HDI16 accesses. The figure illustrates the address lines, the chip select, and the read enable signal as it pertains to the read transaction, all in accordance with the timing diagram illustrated in **Section 2.2.1**, *HDI16 Read Timings,* on page 2. The throughput is calculated to be

11.21 MB/s. This value translates into approximately 16.9 SC140 host bus clock cycles and is averaged over 16 transactions. Since the MSC8101 host issues the transactions with a **move** instruction, the SC140 core stalls during the transaction. If the SC140 core operates three times faster than the system bus clock, the SC140 core stalls for 16.9 × 3 = 50.7 core clock cycles. The system bus clock is not connected to the MSC8101 HDI16 but is illustrated in **Figure 4** because the memory controller signals are based on this clock.

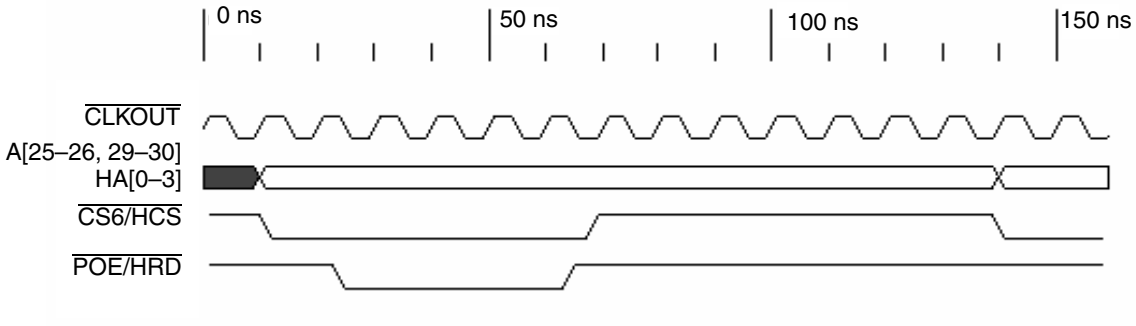

Figure 4. Single Read, Slave Poll Method

4.2.2 Single Read Throughput, Slave HDI16 Interrupt Method

The maximum HDI16 single read throughput with the slave HDI16 interrupt method is based on the accesses to the internal HDI16 registers. **Figure 4** shows the timings obtained from the logic analyzer, which were the timings indicated in **Section 4.2.1**. The throughput is calculated to be 11.21 MB/s, which is the same as the slave poll method of transmitting the data. Everything stated in **Section 4.2.1** applies to the slave HDI16 interrupt method.

Note: The slave DMA method of retrieving data is not applicable unless large amounts of data are transferred, so this application note does not discuss this method, except for host DMA single read accesses.

4.2.3 Host DMA Read Throughput, Slave Poll Method

The maximum HDI16 host DMA read throughput with the slave polling the HDI16 flags is based on the host DMA accesses to the internal HDI16 registers. **Figure 5** shows the timings obtained form the logic analyzer.The figure illustrates the host clock, the address lines, chip select and the read enable signal. The throughput is calculated to be 34.38 MB/s. The DMA on the host side has a reduced latency when receiving data from consecutive HDI16 address locations, thus the host DMA transaction has a much larger throughput than a single read transaction.

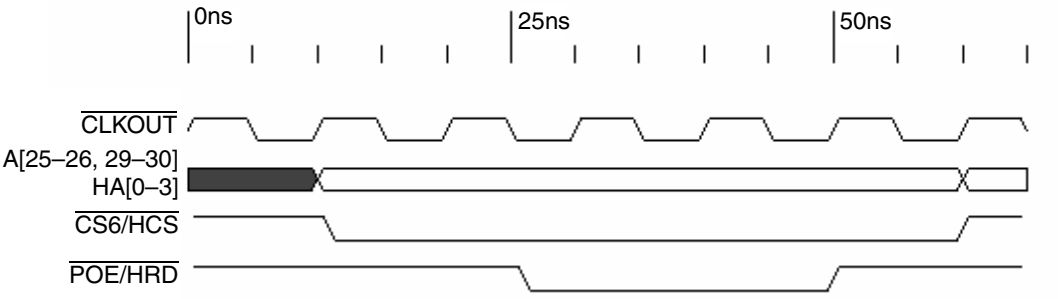

Figure 5. Host DMA Read, Slave Poll Method

4.2.4 Host DMA Read Throughput, Slave DMA Method

The maximum HDI16 host DMA read throughput with the slave DMA method is 34.38 MB/s. **Figure 5** shows the timings obtained form the logic analyzer, which are the timings indicated in **Section 4.2.3**. This method obtains the same throughput as is obtained with the slave poll method. The SC140 core does not have to be involved with the transactions once the DMA controller and the HDI16 are initialized.

4.2.5 Host DMA Read Throughput, Slave HDI16 Interrupt Method

The maximum HDI16 host DMA read throughput with the slave HDI16 interrupt method is 34.38 MB/s. **Figure 5** shows the timings obtained form the logic analyzer, which are the same timings indicated in **Section 4.2.3**. This method obtains the same throughput as was obtained with the slave poll method and the slave DMA method.

4.3 HDI16 Write Throughput

As with the read transactions, there are three different ways that the slave MSC8101 can receive data from the HDI16: the slave poll method, the slave DMA method, and the slave HDI16 interrupt method.

4.3.1 Single Write Throughput, Slave Poll Method

The maximum HDI16 single write throughput with the slave polling the HDI16 flags is based on the accesses to the internal HDI16 registers. **Figure 6** shows the timings obtained form the logic analyzer. The figure illustrates the address lines, the chip select, and the write enable signal as it pertains to the write transaction, all in accordance with the timing diagram illustrated in **Section 2.2.2**, *HDI16 Write Timings,* on page 4. The throughput is calculated to be 15.25 MB/s. This value translates into approximately 12.5 MSC8101 host bus clock cycles and is averaged over 16 transactions. Since the MSC8101 host issues the transactions with a **move** instruction, the SC140 core stalls during the transaction. If the SC140 core operates three times faster than the system bus clock, the SC140 core stalls for 12.5 × 3 = 37.5 core clock cycles.

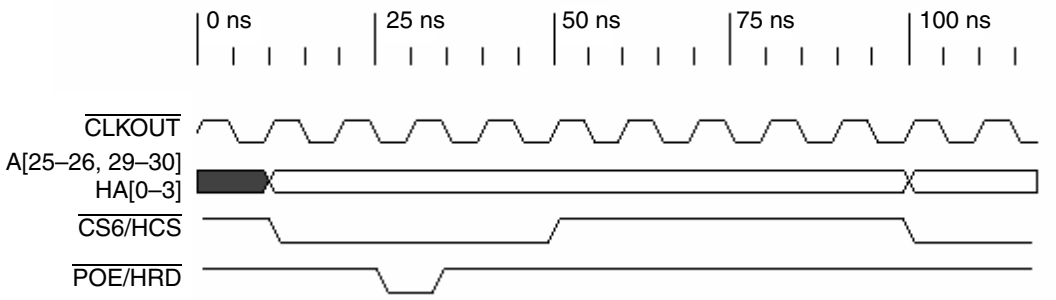

Figure 6. Single Write, Slave Poll Method

4.3.2 Single Write Throughput, Slave HDI16 Interrupt Method

The maximum HDI16 single write throughput with the slave HDI16 interrupt method is also 15.25 MB/s. **Figure 6** shows the timings obtained form the logic analyzer, which are the same as those indicated in **Section 4.3.1**. The throughput is the same as for the slave poll method of receiving the data. Everything stated in **Section 4.3.1** applies to this section. The slave DMA method of retrieving data is not applicable unless large amounts of data are transferred, so this application note does not discuss this method, except for host DMA single read accesses.

4.3.3 Host DMA Write Throughput Slave Poll Method

The maximum HDI16 host DMA write throughput with the slave polling the HDI16 flags is based on the accesses to the internal HDI16 registers. **Figure 7** shows the timings obtained form the logic analyzer. The figure illustrates the host clock, address lines, chip select, and write enable signal. The throughput is calculated to be 46.95 MBs/s. The DMA controller on the host side has a reduced latency when it transmits data to consecutive HDI16 registers. Therefore, the host DMA transaction has a much larger throughput than a single write transaction.

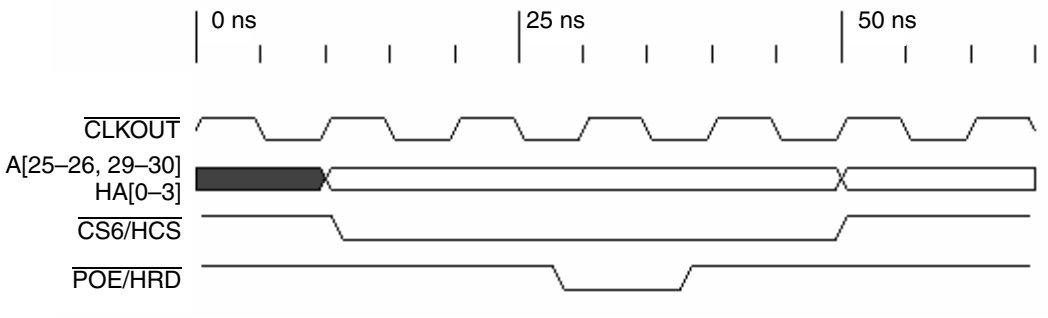

Figure 7. Host DMA Write, Slave Poll Method

4.3.4 Host DMA Write Throughput, Slave DMA Method

The maximum HDI16 host DMA write throughput with the slave DMA method is 46.95 MB/s. **Figure 7** shows the timings obtained form the logic analyzer, which are the same timings as those indicated in **Section 4.3.3**. This is the same throughput that is obtained with the slave poll method.

4.3.5 Host DMA Write Throughput, Slave HDI16 Interrupt Method

The maximum HDI16 host DMA write throughput with the slave HDI16 interrupt method is the same as the throughput obtained with the slave poll method and the slave DMA method. **Figure 7** shows the timings obtained form the logic analyzer, which are the same timings indicated in **Section 4.3.3**. The throughput is calculated to be 46.95 MB/s.

5 Summary

The MSC8101 HDI16 throughput measurements discussed in this application note lead to the following conclusions:

- *Single reads and writes*. The performance numbers for single reads and writes are much lower than those for the host DMA throughput results. When the host uses the DMA controller to transfer the data to consecutive memory locations on the slave MSC8101 HDI16, the latency between accesses is less than consecutive core accesses. The throughput numbers suggest that the MSC8101 host should use single reads/writes only when accessing HDI16 registers or transferring very few bytes of data. When large amounts of data are transferred, the host DMA method is preferred because of the larger throughput capabilities of the HDI16.
- *Single reads* have the same throughput no matter how the data is transferred to the HDI16 registers. The slave DMA is not applicable unless large amounts of data are transferred. In fact, this application note did not explore this type of transaction for a single read. Therefore, for a single read, the slave HDI16 interrupt should be used to save the SC140 core the effort of polling the HDI16 flags. This

frees the SC140 core for other processing tasks until the HDI16 registers are ready for the data transfer.

- *Single writes* have the same throughput no matter how the MSC8101 slave receives the data from the HDI16 registers. As with single reads, the slave DMA method is not applicable for single writes unless large amounts of data are transferred. Therefore, for a single write, the slave HDI16 interrupt should be used to save the SC140 core the effort of polling the HDI16 flags.
- *Host DMA reads* have the same throughput no matter how the data is transmitted to the HDI16 registers. Therefore, when a lot of data is transferred, use the slave DMA method to free the SC140 core for other processing tasks.
- *Host DMA writes* have the same throughput no matter how the data is received from the HDI16 registers. Therefore, when a lot of data is transferred, use the slave DMA method to free the SC140 core for other processing tasks.

6 References

All of the following documents are available online at the Freescale web site listed on the back page of this document:

- MSC8101RM/D, *MSC8101 Reference Manual*
- MSC8101/D, *MSC8101 Technical Data* sheet
- MSC8101ADSUM/D, *MSC8101 Application Development System User's Manual*
- AN2505/D, *MSC8102 Asynchronous DSI Throughput*

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.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014 or +81 3 5437 9125
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

AN2567
Rev. 1
12/2004

freescale™
semiconductor