Order Number: MC68HC55
Rev. 2

freescale semiconductor

MC68HC55/D

Technical Data

Two-Channel CMOS ASIC Device

This document contains information on a new product. Specifications and information herein are subject to change without notice.

Section 1. DSI/D (Distributed System Interface – Digital)

The distributed system interface uses a 2-wire bus to interconnect a master controller with sensors and firing units in an airbag system or other automotive body control applications. Remote units are a key feature of this system because they do not require a crystal or other critical timing reference components which might be damaged in a crash.

This document describes a 2-channel CMOS (complmenetary metal-oxide semiconductor) ASIC (application-specific integrated circuit) device, the MC68HC55. It communicates with the (master) serial peripheral interface (SPI) from an MCU and generates interface signals to/from an analog SmartMOS™ device called the DSI/P (where P indicates the physical layer of the DSI system). **1.2 General Description of the DSI System** gives a general overview of the distributed system interface (DSI), providing a context for understanding the details of this specification.

1.1 Features

Features of the MC68HC55 include:

- Compact 16-pin narrow body SOIC (small outline integrated circuit) package
- Compatible with Freescale SPI (serial peripheral interface) communication interface
- Compatible with DSI/P analog ASIC
- Two independent DSI channels
- Pin assignments match DSI/P for easy connection
- Automatic CRC (cyclical redundancy check) generation and checking for robust message verification
- 4-stage transmit and receive FIFOs (first in, first out)
- 8-bit or 16-bit messages, plus 4-bit CRC

SmartMOS is a trademark of Freescale Semiconductor, Inc.

1.2 General Description of the DSI System

The distributed system interface is intended as a robust serial interface system suitable for automotive body applications including airbag safety systems. Specific goals of this system include:

- Reduction of intermodule wiring by using a 2-conductor bus.
- Provide power to remote nodes on the same pair of wires
- Self-organizing system (modules need not be programmed prior to installation)
- No critical timing components such as crystals in remote nodes (subject to damage in a crash)
- Wave shaped signals to minimize electrical noise
- CRC error checking on all messages to assure message integrity

1.3 Overall DSI System Connections

The DSI system consists of:

- Master MCU such as the MC68HC16P3
- One or more DSI/D devices, one DSI/P dual analog SmartMOS device per DSI/D
- Up to 15 remote sensor and firing units per DSI bus

In this DSI system, the master MCU will communicate with the DSI/D via the SPI. The DSI/D will in turn communicate with a DSI/P via three wires using CMOS levels. Finally, the analog DSI/P will interface to the 2-wire DSI bus using a combination of pulse length encoded voltage levels for transmission of data to remote peripheral units and current return signals for received data at the same time. All signals on the 2-wire DSI physical layer bus are wave-shaped to minimize electrical noise. Refer to **Figure 1-1** for the following discussion.

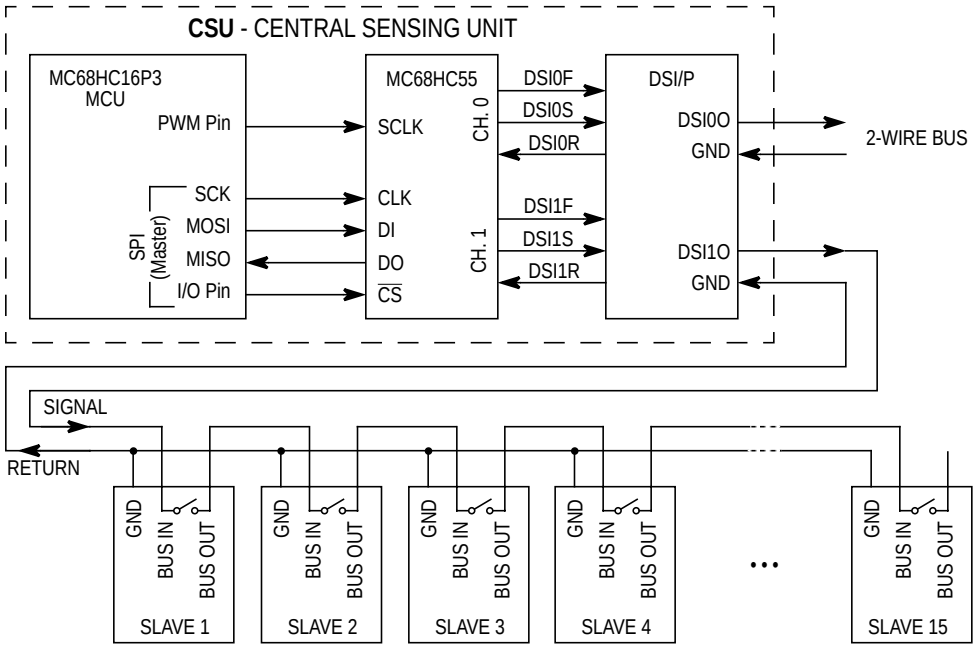


Figure 1-1. DSI System Connections (Other Arrangements Possible)

After reset, the master MCU must first configure the DSI system by assigning a unique 4-bit address to each remote peripheral unit on the 2-wire DSI bus. When the system is initially powered up, the 2-wire bus idles at about 25 volts to provide power to the local power supply capacitor at the first node of the distributed bus. A bus switch in each peripheral node is initially opened until that node has been programmed with a 4-bit address. After power up, the 2-wire bus only goes as far as the first node. The master MCU provides a series of SPI commands to the DSI/D to send a 16-bit (plus 4-bit CRC) message instructing it to assign a 4-bit address to the first node. Address 0000 is a global address which addresses all nodes whether or not they have been initialized with their unique 4-bit address. During this first message, no remote node will respond (provide return data to the master). This first message sets the address of the first node, which causes it to close its bus switch.The bus now goes to the first and second nodes along the 2-wire bus, and the second node power supply capacitor becomes charged by the 25-volt idle level on the bus.

The second message assigns the address to the second node and receives a response message from the first node at the same time. The first node ignores the second address assignment message because it already has an address. During the third address assignment message, only the second node responds (a node only responds to the assignment message once).This sequence continues until all nodes have been assigned addresses and have responded that the assignment was successful.

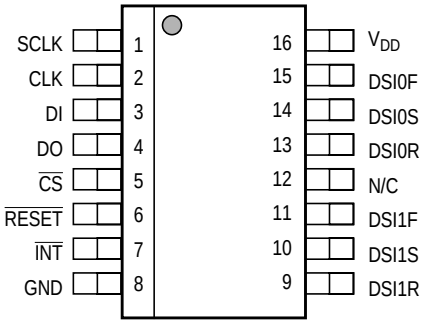
Now that the nodes each have a unique address, messages can be sent and received from individual nodes. Notice that during a message, the node that is receiving the command and the node that is returning a response are not necessarily the same node.

Nodes do not have any permanently programmed address. Addresses are assigned according to the order of the devices on the bus every time power is applied. Multiple nodes may be replaced and/or the system can be reconfigured by adding nodes to the bus, and the system will automatically reconfigure itself at the next power-on.

Section 2. MC68HC55CD Pin Assignments and Descriptions

Refer to **Figure 2-1** for the MC68HC55CD pin assignments. A brief description of the pins is given in this section.

2.1 Pin Assignments



16-lead narrow body SOIC, package #751B-05 issue J

Figure 2-1. MC68HC55CD Pin Assignments

Table 2-1. Pin Information

Pin #	Pin Name	Description	Direction Relative to DSI/D	Source/ Destination
1	SCLK	System clock	Input	From MCU
2	CLK	SPI clock	Input	SCK out from MCU
3	DI	SPI data in to DSI/D	Input	MOSI from MCU
4	DO	SPI data out from DSI/D	Three-state output	MISO to MCU
5	CS	SPI chip select	Active low input	Port output from MCU
6	RESET	DSI/D reset	Active low input	$\overline{\text{RESET}}$ of MCU system
7	INT	Interrupt output	Active low output (open-drain)	To $\overline{\text{IRQ}}$ input of MCU
8	GND	Supply common	—	Supply common
9	DSI1R	DSI channel 1 return	CMOS input	From DSI_R of DSI/P
10	DSI1S	DSI channel 1 signal	CMOS output	To DSI_S of DSI/P
11	DSI1F	DSI channel 1 frame	CMOS output	To DSI_F of DSI/P
12	N/C	No connection	—	—
13	DSI0R	DSI channel 0 return	CMOS input	From DSI_R of DSI/P
14	DSI0S	DSI channel 0 signal	CMOS output	To DSI_S of DSI/P
15	DSI0F	DSI channel 0 frame	CMOS output	To DSI_F of DSI/P
16	V_{DD}	Positive supply	—	—

2.2 Pin Function Descriptions

SCLK — This clock input controls bit and frame timing for messages between the DSI/D and the DSI/P. The length of a bit time may be configured to be 3, 6, 12, or 24 periods of SCLK by the setting of the CDIV0[B:A] and CDIV1[B:A] bit fields (the setting for each DSI/D channel is independently controlled). SCLK is relatively slow (nom. 15 kHz to 450 kHz) compared to the SPI clock (nom. 4 MHz).

CLK — This is the SPI clock signal from the MCU. The DSI/D is a slave SPI device.

DI — SPI data in to DSI/D (MOSI pin of MCU).

DO — SPI data out from DSI/D (MISO pin of MCU).

$\overline{\text{CS}}$ — SPI chip select input. When this signal is high, the DSI/D is deselected and ignores the other SPI signals. CLK and DI are high impedance inputs, and DO is high-impedance when $\overline{\text{CS}}$ is high.The first SPI transfer after $\overline{\text{CS}}$ goes low is always a command to the DSI/D. If $\overline{\text{CS}}$ is held low for additional SPI transfers, they are considered to be data related to the previous command.

$\overline{\text{RESET}}$ — DSI/D system reset. A low level on this pin forces the DSI/D logic to abort any operation in progress and initialize to a startup condition.

$\overline{\text{INT}}$ — This active-low open-drain output signals that some internal condition needs attention. Separate mask bits are provided for the receive FIFO not empty and transmit FIFO empty interrupt conditions for each DSI/D channel (a total of four mask bits). $\overline{\text{INT}}$ remains low as long as any enabled interrupt condition is still pending.

GND — Power supply ground return

DSI1R — DSI channel 1 return. This is the data input signal from the DSI/P. The DSI/D samples the CMOS level on this pin at the end of a bit time. This level will correspond to the current sensed on the signal line of the DSI physical interface by the DSI/P.

DSI1S — DSI channel 1 signal. This is the data output signal to the DSI/P. Data bits are pulse length encoded voltage levels on this signal line. A logic 0 starts with a falling edge on DSI1S and is low for two-thirds of the bit time and then high for one-third of the bit time. A logic 1 starts with a falling edge on DSI1S and is low for one-third of the bit time and then high for two-thirds of the bit time.

DSI1F — DSI channel 1 frame.This output idles high and is driven low during each transfer frame.

DSI0R — DSI channel 0 return. This is the data input signal from the DSI/P. The DSI/D samples the CMOS level on this pin at the end of a bit time. This level will correspond to the current sensed on the signal line of the DSI physical interface by the DSI/P.

DSI0S — DSI channel 0 signal. This is the data output signal to the DSI/P. Data bits are pulse length encoded voltage levels on this signal

line. A logic 0 starts with a falling edge on DSI0S and is low for two-thirds of the bit time and then high for one-third of the bit time. A logic 1 starts with a falling edge on DSI0S and is low for one-third of the bit time and then high for two-thirds of the bit time.

DSI0F — DSI channel 0 frame. This output idles high and is driven low during each transfer frame.

V_{DD} — This is the positive voltage supply input (nom. 5 V).

Section 3. Registers and Bit Descriptions

The MC68HC55 includes eight 8-bit registers. A master MCU reads from or writes to these registers through an SPI. For more information about the SPI and command protocol, refer to **4.4 SPI Communications**

Table 3-1. Register Summary

Register Address	Register Name	Description
000	DSI0H	DSI channel 0 data register (upper byte)
001	DSI0L	DSI channel 0 data register (lower byte)
010	DSI1H	DSI channel 1 data register (upper byte)
011	DSI1L	DSI channel 1 data register (lower byte)
100	DSISTAT	DSI status register
101	DSI0CTRL	DSI channel 0 control register
110	DSI1CTRL	DSI channel 1 control register
111	DSIENABLE	DSI channel enable bits

3.1 DSI Channel 0 Data Registers

Address: $000

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Blt 15	14	13	12	11	10	9	Bit 8
Reset:	0	0	0	0	0	0	0	0

Figure 3-1. DSI Channel 0 Data Register Upper Byte (DSI0H)

Address: $001

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Blt 7	6	5	4	3	2	1	Bit 0
Reset:	0	0	0	0	0	0	0	0

Figure 3-2. DSI Channel 0 Data Register Lower Byte (DSI0L)

Reads access the receive data FIFO for channel 0. A receive FIFO not empty (RFNE0) status bit in the DSISTAT register indicates when it is appropriate to read the DSI0H and DSI0L register pair. Reading DSI0L, while the receive FIFO is not empty, causes a receive FIFO pop. In the case of reading the 16-bit DSI0H:DSI0L register pair, read DSI0H first so the receiver FIFO pop does not occur prematurely. Writes access the transmit data FIFO for channel 0. A transmit FIFO not full (TFNF0) status bit in the DSISTAT register indicates when it is appropriate (TFNF0 = 1) to write to the DSI0H and DSI0L register pair. Writing DSI0L causes a transmit FIFO push. In the case of writing the 16-bit DSI0H:DSI0L register pair, write DSI0H first so the FIFO push does not occur prematurely. Both receive and transmit are equipped with 4-stage FIFOs.

3.2 DSI Channel 1 Data Registers

Address: $010

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Blt 15	14	13	12	11	10	9	Bit 8
Reset:	0	0	0	0	0	0	0	0

Figure 3-3. DSI Channel 1 Data Register Upper Byte (DSI1H)

Address: $011

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Blt 7	6	5	4	3	2	1	Bit 0
Reset:	0	0	0	0	0	0	0	0

Figure 3-4. DSI Channel 1 Data Register Lower Byte (DSI1L)

For the description of DSI1H and DSI1L, refer to **3.1 DSI Channel 0 Data Registers**.

3.3 DSI Status Register

Address: $100

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ER1	TFE1	TFNF1	RFNE1	ER0	TFE0	TFNF0	RFNE0
Write:								
Reset:	0	1	1	0	0	1	1	0

= Unimplemented

Figure 3-5. DSI Status Register (DSISTAT)

This 8-bit register provides status information for both channels of the DSI/D. A copy of this register is latched at the falling edge of the SPI chip select to avoid asynchronous problems due to bits changing during an SPI transfer. Any changes that occur while chip select remains low (due to SPI data reads or writes or reception of new DSI data, etc.) will not be

visible via the SPI until chip select rises and returns low to start a new SPI transfer. Reads of this register should be considered a snapshot of the status at the last falling edge of chip select.

NOTE: *To guarantee coherence between an SPI read of status and data, the reads must be within the same SPI burst ($\overline{CS}$ must remain continuously low for the data and status reads). One way to assure this is to always read data in a burst, starting with a command referencing DSI0H through DSI1L, leaving the register pointer pointing at the DSISTAT register (see* ***Figure 4-7****). The first SPI transfer which corresponds to the read or write address 000 command will return (read) register 100 (DSISTAT). The values of DSISTAT and DSI0H through DSI1L are latched at the falling edge of $\overline{CS}$, so changes due to DSI transfers are not seen until a future SPI transfer.*

ER1 — CRC Error Bit (Channel 1 Read)

0 = CRC value for the data in the read buffer was correct.
1 = CRC value for the data in the read buffer was not correct (data is not valid).

CRC errors are associated with each data value in the receive FIFO, so each FIFO entry has a bit to indicate whether the data in that stage of the FIFO was received correctly.

Whenever a received value is visible at DSI1H:DSI1L, the associated CRC error status is visible at ER1 in the DSISTAT register. When a new data value becomes visible due to a pop of a previous value, the ER1 status flag reflects the CRC status of the new data value. There is no separate interrupt associated with ER1 because it is always associated with the RFNE1 status flag.

TFE1 — Transmit FIFO Empty Bit (Channel 1)

0 = Transmit FIFO not empty
1 = Transmit FIFO empty

When the transmit FIFO is empty, four consecutive write bursts may be used to fill the FIFO without checking the flags between writes. An interrupt may be generated on the transmit FIFO empty condition.

TFNF1 — Transmit FIFO Not Full Bit (Channel 1)

0 = Transmit FIFO full; no room to write any additional data

1 = FIFO not full; there is room for more data in the transmit FIFO

There is no interrupt associated with the transmit FIFO not full condition. When the conclusion of a DSI transfer frame would cause both TFNF and RFNE to become set, RFNE becomes set but TFNF is not set until one DSI clock cycle later. When the transmit FIFO is full, attempts to write more data into the FIFO are ignored.

RFNE1 — Receive FIFO Not Empty Bit (Channel 1)

0 = Receive FIFO empty; no new data ready to be read

1 = One or more data entries in receive FIFO; data is available to be read

It is not possible to overflow the receive FIFO because it is not possible to get more than four transmit messages into the system at a time. When there is any data in the receive FIFO, a write to the transmit buffer also reads (pops) data from the receive FIFO.

ER0 — CRC Error Bit (Channel 0 Read)

0 = CRC value for the data in the read buffer was correct.

1 = CRC value for the data in the read buffer was not correct (data is not valid).

Refer to the description of ER1.

TFE0 — Transmit FIFO Empty Bit (Channel 0)

0 = Transmit FIFO not empty

1 = Transmit FIFO empty

Refer to the description of TFE1.

TFNF0 — Transmit FIFO Not Full Bit (Channel 0)

0 = Transmit FIFO full; no room to write any additional data

1 = FIFO not full; there is room for more data in the transmit FIFO

Refer to the description of TFNF1.

RFNE0 — Receive FIFO Not Empty Bit (Channel 0)

0 = Receive FIFO empty; no new data ready to be read

1 = One or more data entries in receive FIFO; data is available to be read

Refer to the description of RFNE1.

3.4 DSI Channel Control Registers

Address: $101

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	CDIV0B	CDIV0A	DLY0B	DLY0A	RIE0	TIE0	0	MS0
Reset:	0	0	0	0	0	0	0	0

Figure 3-6. DSI Channel 0 Control Register (DSI0CTRL)

This register should be written before data is sent over the DSI bus. Any write to this register causes any DSI transfer in progress on this DSI channel to be aborted (see **4.2 Abort Function**). The bits in this register are updated as they are received over the SPI, but no new values take effect until the next DSI clock cycle after the conclusion of the SPI write to this register.

CDIV0[B:A] — Clock Divider for Channel 0 Bits

The CDIV0[B:A] bits specify an additional divider between the SCLK input and the bit timing circuitry.

When CDIV0[B:A] are set for 0:0, each bit time on the DSI/D to DSI/P interface is three SCLK periods long.

Table 3-2. CDIV0 Divider Information

CDIV0[B:A]	Divisor	SCLK Periods per Bit Time
0:0	÷1	3
0:1	÷2	6
1:0	÷4	12
1:1	÷8	24

DLY0[B:A] — Inter-Frame Delay for Channel 0 Bits

These bits specify the minimum delay between transfer frames on the DSI/D to DSI/P interface. When DLY0[B:A] are set for 0:0, there will be a minimum of four bit times of idle line from the end of a transfer

frame to the beginning of the next frame. The length of a bit time depends upon the SCLK input frequency and the current setting in the CDIV0[B:A] bits.

Table 3-3. DLY0 Frame Spacing Information

DLY0[B:A]	Minimum Delay Between Frames (Bit Times)
0:0	4
0:1	8
1:0	16
1:1	32

RIE0 — Receive Interrupt Enable (Channel 0) Bit

0 = Receive interrupt disabled; RFNE0 status does not affect $\overline{INT}$ pin.

1 = Receive interrupt enabled; whenever the RFNE0 status flag is 1, the $\overline{INT}$ pin will be low to request an interrupt.

TIE0 — Transmit Interrupt Enable (Channel 0) Bit

0 = Transmit interrupt disabled; TFE0 status does not affect $\overline{INT}$ pin.

1 = Transmit interrupt enabled, whenever the TFE0 status flag is 1, the $\overline{INT}$ pin will be low to request an interrupt.

MS0 — Message Size (Channel 0) Bit

0 = 16 data bits plus 4 CRC bits data bits 15 through 0 then 4 CRC bits

1 = 8 data bits plus 4 CRC bits data bits 7 through 0 then 4 CRC bits

If the DSI0CTRL register is written while a transfer is in progress, the transfer is aborted without being completed.

Address: $110

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	CDIV1B	CDIV1A	DLY1B	DLY1A	RIE1	TIE1	0	MS1
Reset:	0	0	0	0	0	0	0	0

Figure 3-7. DSI Channel 1 Control Register (DSI1CTRL)

This register should be written before data is sent over the DSI bus. Any write to this register causes any DSI transfer in progress on this DSI channel to be aborted (see **4.2 Abort Function**). The bits in this register are updated as they are received over the SPI interface, but no new values take effect until the next DSI clock cycle after the conclusion of the SPI write to this register.

CDIV1[B:A] — Clock Divider for Channel 1 Bits

The CDIV1[B:A] bits specify an additional divider between the SCLK input and the bit timing circuitry. When CDIV1[B:A] are set for 0:0, each bit time on the DSI/D to DSI/P interface is three SCLK periods long.

Table 3-4. CDIV1 Divider Information

CDIV1[B:A]	Divisor	SCLK Periods per Bit Time
0:0	÷1	3
0:1	÷2	6
1:0	÷4	12
1:1	÷8	24

DLY1[B:A] — Inter-Frame Delay for Channel 1 Bits

These bits specify the minimum delay between transfer frames on the DSI/D to DSI/P interface. When DLY1[B:A] are set for 0:0, there will be a minimum of four bit times of idle line from the end of a transfer

frame to the beginning of the next frame. The length of a bit time depends upon the SCLK input frequency and the current setting in the CDIV1[B:A] bits.

Table 3-5. DLY1 Frame Spacing Information

DLY1[B:A]	Minimum Delay between Frames (Bit Times)
0:0	4
0:1	8
1:0	16
1:1	32

RIE1 — Receive Interrupt Enable (Channel 1) Bit

0 = Receive interrupt disabled; RFNE1 status does not affect $\overline{INT}$ pin.

1 = Receive interrupt enabled; whenever the RFNE1 status flag is 1, the $\overline{INT}$ pin will be low to request an interrupt

TIE1 — Transmit Interrupt Enable (Channel 1) Bit

0 = Transmit interrupt disabled; TFE1 status does not affect $\overline{INT}$ pin.

1 = Transmit interrupt enabled; whenever the TFE1 status flag is 1, the $\overline{INT}$ pin will be low to request an interrupt

MS1 — Message Size (Channel 1) Bit

0 = 16 data bits plus 4 CRC bits data bits 15 through 0 then 4 CRC bits

1 = 8 data bits plus 4 CRC bits data bits 7 through 0 then 4 CRC bits

If the DSI1CTRL register is written while a transfer is in progress, the transfer is aborted without being completed.

3.5 DSI Channel Enable Bits

Address: $111

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	0	0	0	0	0	0	EN1	EN0
Reset:							0	0

Figure 3-8. DSI Channel Enable Bits (DSIENABLE)

This read/write register is used to enable or disable each DSI channel. When a DSI channel is disabled, its DSIxF pin is high and its DSIxS pin is low which forces the DSI/P device to three-state its bus outputs. Disabling a DSI channel clears the transmit and receive FIFOs for that channel. See **4.3 Enable (Disable) Function**.

NOTE: *Bits [7:2] are reserved and read as 0s. These bits could be used in future versions of the MC68HC55.*

EN1 — Enable for DSI Channel 1 Bit
- 0 = DSI channel 1 disabled;
  DSI1F pin = high (1), DSI1S pin = low (0)
- 1 = DSI channel 1 enabled; operates normally

EN0 — Enable for DSI Channel 0 Bit
- 0 = DSI channel 0 disabled;
  DSI0F pin = high (1), DSI0S pin = low (0)
- 1 = DSI channel 0 enabled; operates normally

Section 4. Functional Description

The MC68HC55 is controlled by a master MCU through an SPI, and handles the digital portions of a distributed system interface (DSI) system. This device includes two separate DSI channels, each capable of interfacing to up to 15 DSI bus slave devices (nodes). The DSI physical layer uses a 2-wire bus with analog wave-shaped voltage and current signals so an analog SmartMOS device called a DSI/P is needed to interface the CMOS logic levels of the DSI/D to the analog physical layer of the DSI bus. Refer to **Figure 4-1** for the following discussions.

Major subsystems within the MC68HC55 include:

- Serial peripheral interface (SPI) to the master MCU
- A register pointer block
- Two channels of DSI data registers buffers:
  - Transmit data register (high and low bytes)
  - Receive data register (high and low bytes)
- CRC block for each channel
- Control and status registers
- Serial clock (SCLK) input block
- 4-level FIFOs on each transmit and receive buffer.

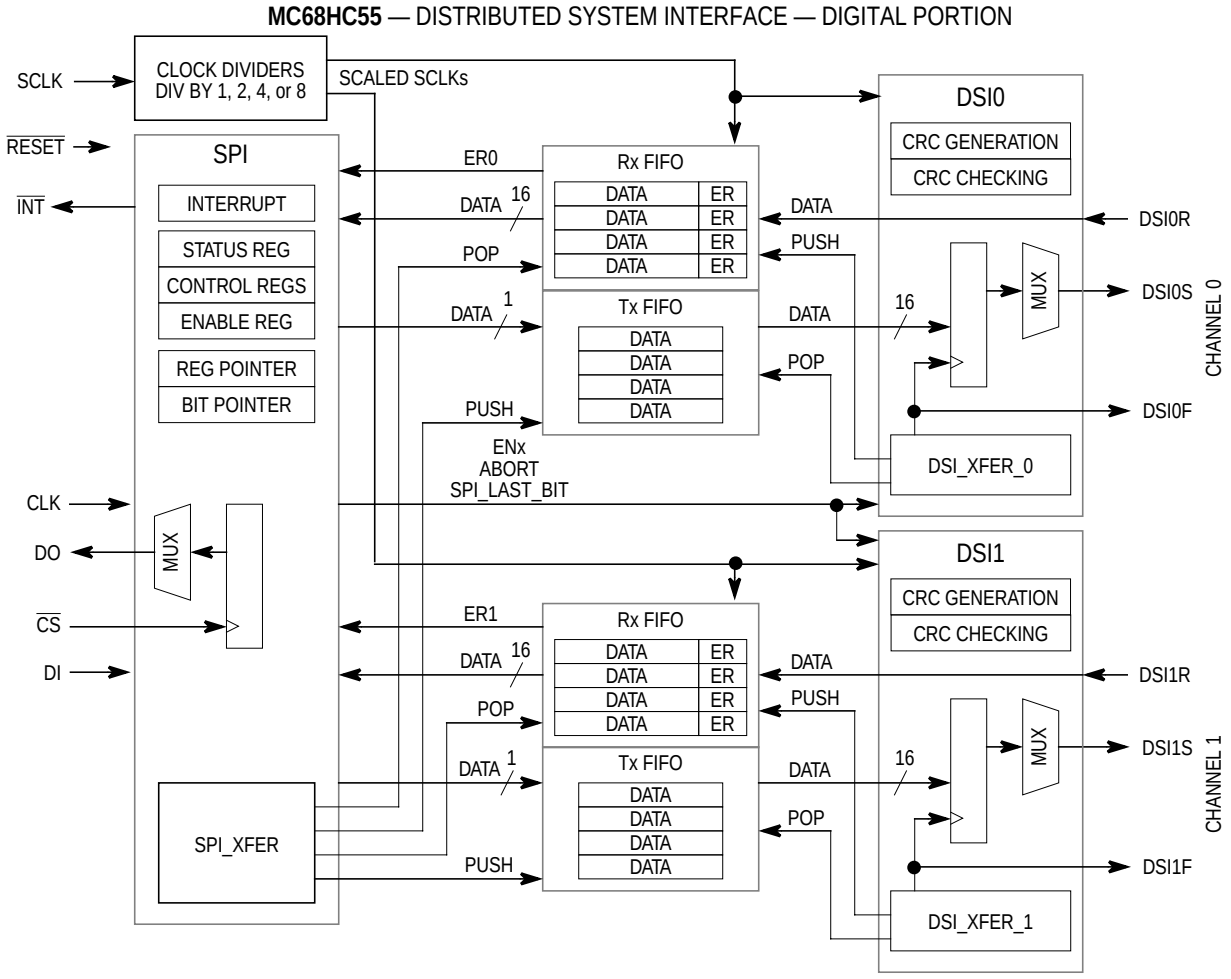


Figure 4-1. MC68HC55 Block Diagram

4.1 Reset Function

A low level on RESET forces all FIFO bits to be cleared. The receive and transmit FIFO pointers are cleared, which effectively forces the FIFOs to the empty condition. Since the DSI channels are disabled (ENx = 0), the DSIxF pins are high and the DSIxS pins are low, which forces the DSI/P devices to three-state their bus outputs.

Reset also forces all MC68HC55 state machines to their idle state.

4.2 Abort Function

Any DSI transfer that was in progress is stopped as soon as the SPI write that caused the abort begins. The DSI/D to DSI/P interface lines return to their idle states. The abort condition is true throughout the SPI write to the DSI control register.

After the last bit of the DSI control register is written, the transmit and receive FIFO pointers are reset, which effectively clears these FIFOs and forces the FIFO locations to 0. A minimum inter-frame delay is then timed (using the new values of clock scaling and delay control bits) to allow reserve capacitors in remote nodes to charge. (Any partial inter-frame delay based on old control settings is forgotten.)

4.3 Enable (Disable) Function

When a DSI channel is disabled, its DSIxF pin is high and its DSIxS pin is low which forces the DSI/P device to three-state its bus output. The transmit and receive FIFO pointers are reset, which effectively clears these FIFOs and forces the FIFO locations to 0. Any DSI transfer that was in progress is stopped.

Although the SPI clock and the DSI input clock both typically come from the same MCU system clock in an MCU plus DSI/D system, there is no guaranteed relationship between these clocks, so the system was designed as if these clocks were asynchronous. The FIFO architecture eliminated most of the cases where these clocks need to interact, and the remaining cases were designed with extra care to prevent asynchronous problems.

Figure 4-2 explains the notation used in the subsequent state diagrams. Entry to the IDLE state is asynchronous and all other state transitions are synchronous. The note in the upper right corner of the figure identifies which edge of which clock or signal is used to synchronize state transitions. Each arrow or arc has a condition which must be true before the transition can take place. This condition can be the value of a single signal or a more complex logic function. A slash (/) indicates the

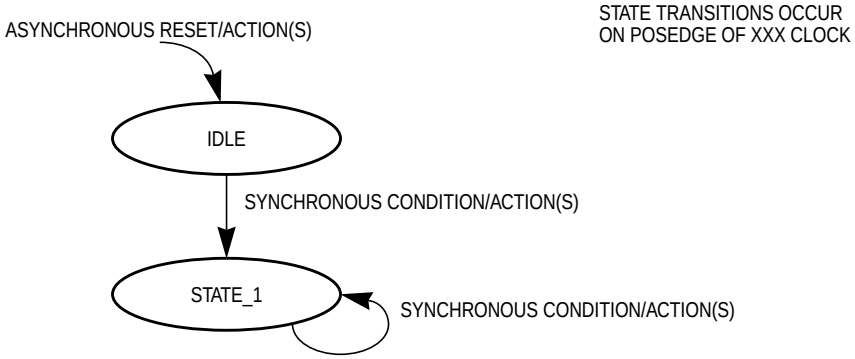


Figure 4-2. State Diagram Notation

end of the condition or equation which must be true for a transition to occur. The statement or statements after the slash are executed during the transition to the next state. These state diagrams are not a complete description of the entire MC68HC55. They are intended to include just enough relevant data to understand the operation of the state machines and basic DSI/D functions.

Figure 4-3 describes how SPI transfers lead to transmit FIFO push operations or transfer abort actions. State transitions in this state machine are synchronous with rising edges of the SPI clock. The initial state, SP_IDLE, is entered asynchronously whenever internal reset becomes active or the SPI chip select input goes high. Upon entry to the idle state, the SPI_WRITE signal is deactivated and the SPI bit counter is set to 7 (it will count down as bits are received).

When the SPI chip select goes low (active), the first SPI transfer will be a command byte and the first bit indicates a write or read command. The SPI_WRITE signal takes on the value of this first bit, and the state machine enters the COMMAND_TRANSFER state, where the remaining bits of the command byte are received. The last three bits of the command set the initial value of the register pointer (update occurs on the next SPI clock falling edge). After the command byte is complete, the state machine advances to the SPI_BURST state, which remains active until the SPI chip select goes high or the MC68HC55 is reset.

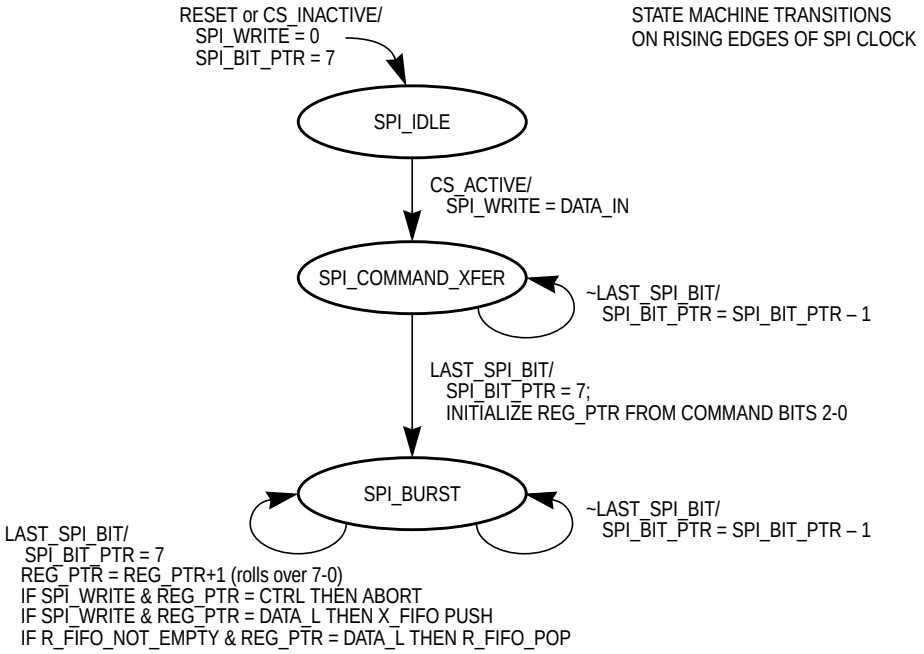


Figure 4-3. State Diagram — SPI Transfer

In the SPI_BURST state, new SPI characters are read-from or written-to-and-read-from DSI/D registers. If the control register is written, an ABORT request is generated which will immediately stop any DSI transfer that was in progress (refer to **Figure 4-4**). If the DATA register low byte is written, a transmit FIFO push operation is generated (see **Figure 4-5**). If the DATA register low byte is accessed (read or written) and there is at least one entry in the receive FIFO, a receive FIFO pop operation is generated.

When a DSI transfer results in both an R_FIFO_PUSH and an X_FIFO_POP, the R_FIFO_PUSH is performed first to avoid the possibility of the transmit FIFO from getting ahead of the receive FIFO.

Figure 4-4 describes what happens during DSI serial transfers. State transfers in this state machine are synchronous with positive edges on the scaled SCLK and the initial state is WAIT_FRAME_DELAY. Initial entry into this state is caused by a reset, abort, or by enable becoming inactive. These conditions cause an asynchronous entry into this state. The exit to the next state, TRANSFER_DSI_BITS, needs to be synchronous.

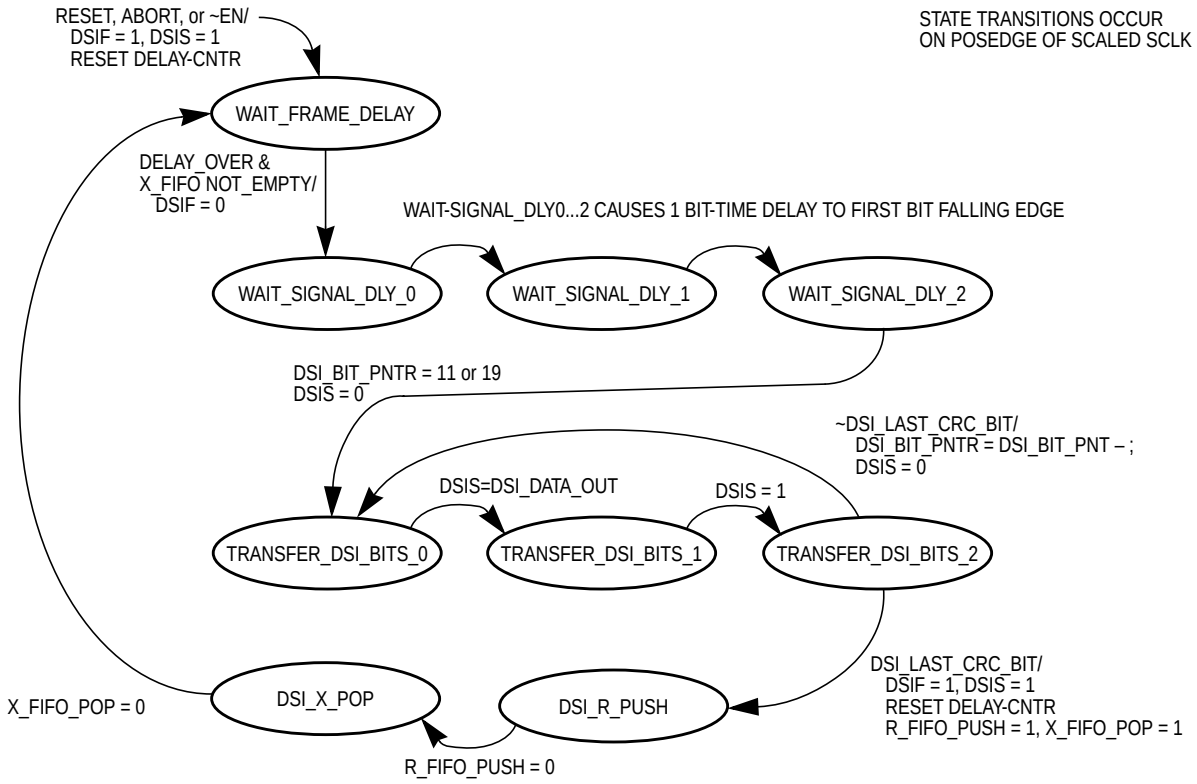


Figure 4-4. State Diagram — DSI Transfer

When enable is true and there is at least one valid entry in the transmit FIFO, the DSI frame signal is driven low to start a frame. States WAIT_SIGNAL_DLY_0 through WAIT_SIGNAL_DLY_2 create a one DSI bit-time delay before the start of the first data bit. After WAIT_SIGNAL_DLY_2, the DSI_BIT_PTR gets initialized to 11 or 19 (depending upon the value in the MSx control bit), and the TRANSFER_DSI_BITS_0 state is entered.

TRANSFER_DSI_BITS_0 through TRANSFER_DSI_BITS_2 form a loop where each pass corresponds to one DSI bit time. During the first third of the bit, the DSIxS pin is low; during the second third, DSIxS is low for a 0 or high for a 1; during the last third of the bit time, DSIxS is high. Provided this is not the end of the last CRC bit, the bit pointer is decremented and the loop is repeated.

After the last CRC bit, the DSI_R_PUSH state is entered. This state ensures that the CRC flag is stable prior to adjusting the receive (and transmit) FIFO pointers. The DSI_X_POP state prevents an X_FIFO_POP from occurring at the same time as an R_FIFO_PUSH.

After DSI_X_POP, the state transitions back to the WAIT_FRAME_DELAY state. This state ensures proper frame spacing is allowed to charge up the storage capacitors in remote nodes. Notice that the delay counter was reset at the end of the last CRC bit so the delay period can start to time out even while the DSI_R_PUSH and DSI_X_POP states are being processed.

Figure 4-5 describes the operation of the transmit FIFO. This FIFO is four levels deep, including the stage which is written into by the SPI and the stage which provides the data for the current DSI serial transfer. State transitions in this state machine occur at the trailing edges of X_FIFO_PUSH and X_FIFO_POP.

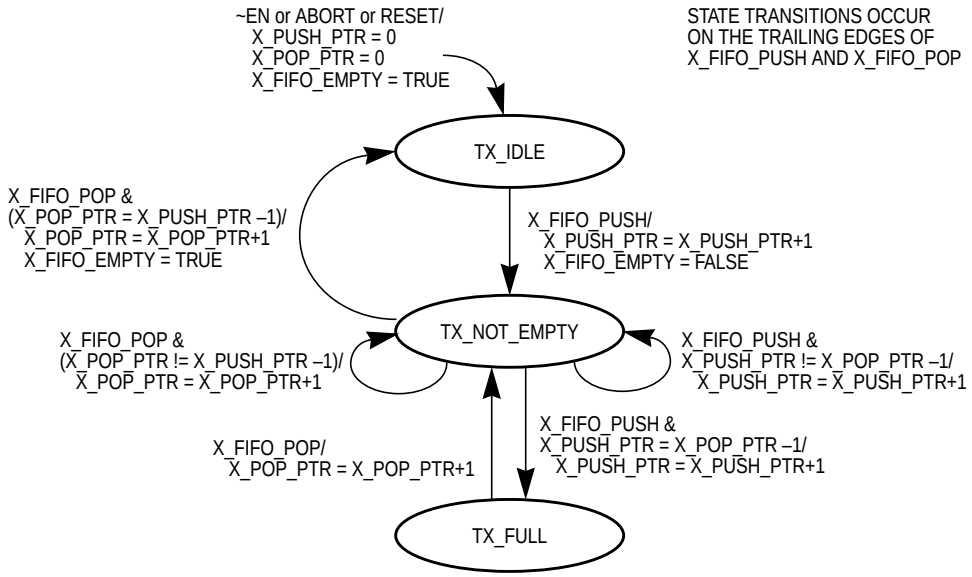


Figure 4-5. State Diagram — Transmit FIFO

When this FIFO is completely empty, the SPI can write four new values to fill the FIFO without waiting for any action on the DSI side of the FIFO. Values are “pushed” into the FIFO from the SPI and values are “popped” after they have been serially sent out the DSI. When the FIFO is full, additional attempts to write new data from the SPI side are ignored. The host MCU should be sure the TFNFx status bit is set before writing more data to the FIFO.

Reset, abort, or enable going to 0 cause asynchronous entry to the TX_IDLE state which corresponds to the FIFO empty condition. The push and pop pointers are cleared and X_FIFO_EMPTY is set to true. X_FIFO_PUSH causes the push pointer to be incremented, X_FIFO_EMPTY to be set to false, and the state to transition to TX_NOT_EMPTY. The push request comes from the SPI transfer state machine after a new value has been written into the FIFO.

From TX_NOT_EMPTY, several things can happen. Additional values can be pushed into the FIFO, if the push pointer is the same as the pop pointer minus one. This push fills the FIFO so the state advances to TX_FULL. Each time a new data value is pushed into the FIFO, the push pointer is incremented. From TX_NOT_EMPTY, values may also be popped from the FIFO, freeing a stage for additional data. If the pop pointer is the same as the push pointer minus one, the pop removes the last value in the FIFO, so X_FIFO_EMPTY is set to true and the state changes back to TX_IDLE. Each time a value is popped, the pop pointer is incremented.

When the transmit FIFO is full, no additional data can be written into the FIFO, so no new push requests will be generated. From TX_FULL, the only valid change is caused by a pop which causes the pop pointer to increment and the state goes back to TX_NOT_EMPTY. (Of course reset, abort, or disable could cause the state to asynchronously change to the TX_IDLE state).

Figure 4-6 describes the operation of the receive FIFO. State transitions in this state machine occur at the trailing edges of R_FIFO_PUSH and R_FIFO_POP. The receive FIFO is four levels deep, including the stage which receives serial data from the current DSI transfer and the stage that is accessible for SPI reads. To assure coherence of data and status, each FIFO stage includes an extra bit for the CRC error status for each received data word. Also for coherency, the DSI transfer state machine imposes a delay at the end of a DSI transfer to assure that the CRC status is stable before issuing the R_FIFO_PUSH request. The RX_IDLE state is asynchronously entered at system reset, when the enable bit goes low, or when there is an abort.

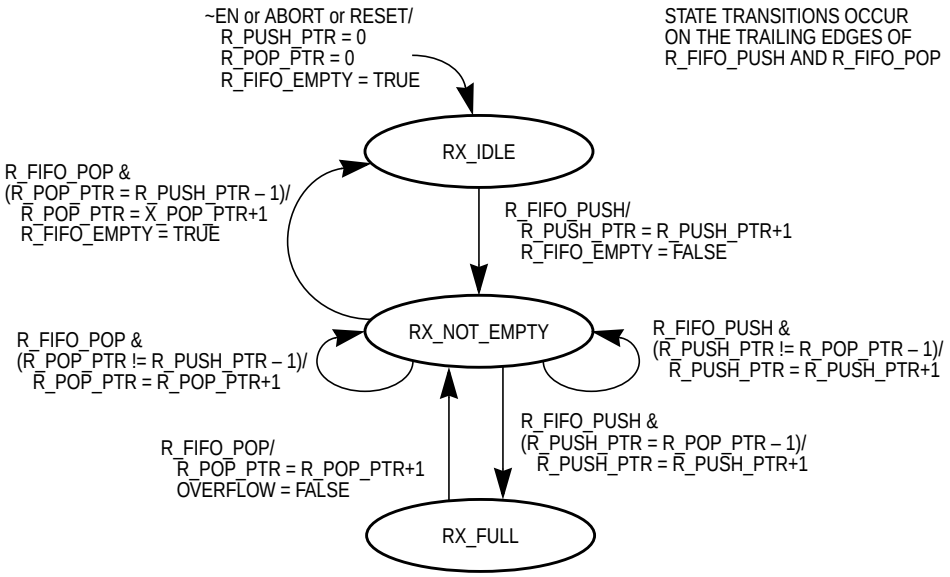


Figure 4-6. State Diagram — Receive FIFO

During normal operation of the receive FIFO, values are pushed into the FIFO from the DSI serial interface, causing the push pointer to increment. After the SPI has read a data word, the receive FIFO is popped, which makes the location available for additional data from the DSI interface (it is the user's responsibility to read status and data within the same burst to assure coherence). The RX_NOT_EMPTY state is active as long as there is some data in the FIFO.

The RX_FULL state is entered when enough data has been pushed into the FIFO from the DSI to cause the push pointer to catch up to the pop pointer. Since it is not possible to introduce another DSI serial character without reading (pop) the receive FIFO, it is not possible to overflow the receive FIFO.

4.4 SPI Communications

The DSI/D is a slave peripheral device which is designed to interface to a Freescake SPI configured as a master with CPHA = CPOL = 0. When the MC68HC55 is deselected ($\overline{CS}$ pin at logic 1), $\overline{CS}$, DI, and CLK are all high-impedance inputs, and DO is disabled (forced to a high-impedance state).

The first SPI transfer, after $\overline{CS}$ is driven from high to low (to select the MC68HC55), is considered a read or write command to the DSI/D (called a command transfer). Bit 7 of the command specifies a write (1) or read (0) command while bits 2, 1, and 0 of the command specify the address of one of the eight registers in the DSI/D. This command establishes an internal register pointer. Data sent back to the master MCU from the DSI/D during a command transfer is the read data from the address previously pointed to (0s for the first transfer after reset). Any additional SPI transfers that occur while $\overline{CS}$ remains low are called data transfers. These additional data transfers write and/or read successive DSI/D registers. The internal register pointer is incremented after each data transfer and automatically rolls over from 7 (111) to 0 (000).

During read data transfers, the data sent from the DSI/D (DO) to the MCU (MISO) is data that was read from the selected DSI/D register at the end of the previous SPI transfer. Data sent from the master MCU (MOSI) to the DSI/D (DI) is ignored. The internal DSI/D register pointer is incremented at the end of the transfer (rolls over to 000 if it was 111 during this transfer).

During write data transfers, the data sent from the DSI/D (DO) to the MCU (MISO) is data that was read from the selected DSI/D register at the end of the previous SPI transfer (or at the falling edge of $\overline{CS}$). Data

sent from the master MCU (MOSI) to the DSI/D (DI) is captured on rising edges of SCK and written, bit by bit, to the selected register as data bits are received (X_FIFO is not pushed unless all bits of DATA_L written). At the end of the transfer, the internal DSI/D register pointer is incremented (rolls over to 000 if it was 111 during this transfer).

The current data from the receive FIFOs (pointed to by the receive pop pointers) is latched along with current status information at any falling edge on the SPI chip select line. This assures that the information is coherent in case further DSI activity causes any changes.

If the user violates normal procedure and writes the transmit data buffer when TFNFx = 0 (indicating the FIFO is full), the new transfer is ignored.

Figure 4-7 Shows a typical burst transmission between the SPI of the host MCU and the SPI in the MC68HC55. This example assumes the DSI/D register pointer was pointing at the STATUS register (100) before the start of the burst. The first SPI transfer is the command from the host MCU (write address 000 in this case). During this first transfer, the data returned to the MCU is the data that was latched from the status register at the falling edge on $\overline{CS}$. During the next SPI transfer, the data from DSI0H (receive FIFO at pop pointer) is sent back to the MCU while new data is written to DSI0H (transmit FIFO at push pointer). Notice $\overline{CS}$ remains low throughout the whole burst sequence. As each SPI transfer completes, the DSI/D register pointer is incremented to point at the next register. At the end of this burst, the register pointer is left pointing to the status register so it is ready for the next SPI burst.

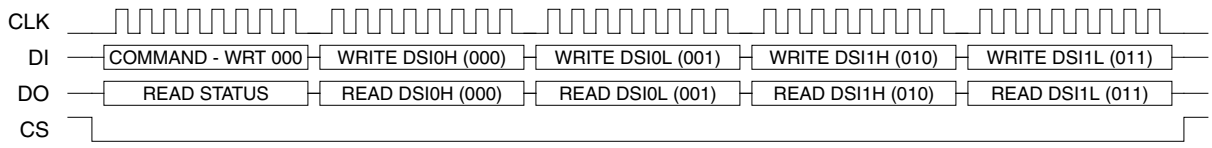


Figure 4-7. SPI Burst Transfer Example

Although it looks like the read and write for an address are occurring at the same time, the changes caused by the write are not reflected in the read that is taking place during the same SPI transfer. Also, if the burst were extended such that it read status again, the status changes caused earlier during the same burst would not be reflected (status is latched at chip select fall).

4.5 DSI/D to DSI/P Communications

DSI/D to DSI/P communications involve a frame (DSIxF), a data signal (DSIxS), and a data return (DSIxR) signal. This ASIC device supports two independent channels of DSI communications and is used with a matching 2-channel analog SmartMOS device called the DSI/P. All DSI/D to DSI/P communications are initiated by the DSI/D in response to register writes within the DSI/D which in turn are controlled by commands received through the SPI.

Messages are eight or 16 bits of data (as controlled by the MSx bit in the DSIxCTRL register) plus four bits of CRC, so messages contain 12 or 20 bits of information. CRC generation and checking are done in the DSI/D. A message starts with a falling edge on the DSIxF signal which marks the start of a frame. There is a one bit-time delay before the MSB (most significant bit) of data appears on the DSIxS pin. Data bits start with a falling edge on DSIxS. The low time is one-third of the bit time for a 1 and two-thirds of a bit time for 0. Data is transmitted on DSIxS and received on DSIxR pins simultaneously. Receive data is the captured level on the DSIxR pin at the end of each bit time. At the end of the bit time for the last CRC bit, the DSIxF pin returns to a logic high (idle level). The DSI/D imposes a minimum delay between successive frames of 4, 8, 16, or 32 bit times.

The user initiates a message by writing (via the SPI interface from the MCU) to the low byte of the data register (DSIxL). When 16-bit messages are to be sent, this allows the user to write the DSIxH register and then the DSIxL register before the combined 16-bit data value (DSIxH:DSIxL) is sent. The user should first check the TFNFx status flag to be sure the transmit FIFO is not full before writing a new data value to DSIxH and/or DSIxL. When the minimum inter-frame delay has been

satisfied, the DSIxF pin will go low, indicating the start of a new transfer frame.

Data is shifted out of DSIxS (MSB first) and shifted in to DSIxR at the same time. As a message is received it is stored bit-by-bit into the next available receive FIFO location (if the FIFO is already full, an internal overflow flag is set and the serial receive data is lost). For each data value in the receive FIFO, there is a 1-bit status flag to indicate if there was a CRC error while receiving the data. At the end of a DSI transfer (and after the CRC error status is stable), the RFNEx flag is set (if it wasn't already) to indicate there is data in the receive FIFO to be read.

4.6 CRC Generation/Checking

Whenever a message is sent from the DSI/D to the DSI/P, a 4-bit CRC value is computed and serially sent as the next four bits after the LSB (least significant bit) of the data. The DSI/P passes the message, including the CRC bits, along to a remote peripheral which computes a separate CRC value as the message data is received. If this computed CRC does not agree with the CRC value received in the message, the peripheral device considers the message invalid.

Messages received by the DSI/D include a 4-bit CRC value which was computed in the peripheral device that is responding to the DSI/D. As the DSI/D receives a message, it computes a separate 4-bit CRC value and compares this with the CRC value in the received message. If these values do not agree, the message is considered invalid and the ERx status flag is set as the receive data is transferred into the receive data buffer.

When no remote peripheral responds to a DSI/D message, the data pattern received by the DSI/D will be all 0s with a CRC value of 0000 which is detected as a CRC error. The correct CRC value for a $00 or $0000 message would be 1010. The same 1010 CRC value applies for both 8- and 16-bit messages of all 0s.

CRC errors are indicated by the ER0 and ER1 status bits. ERx status bits are cleared when new data is transferred from the shifter to the read data buffer and the CRC value was correct.

4.7 CRC Computation

The 4-bit CRC uses an initialization value of 1010 and a polynomial of x^4+1. The following is a VHDL description of the CRC algorithm.

```
-------------------------------------------------------
-- Calculates the 4-bit CRC (x^4 + 1) serially for
-- 8 or 16 bits of data.
-------------------------------------------------------

constant CRCPoly: std_logic_vector := “0001”; -- x^4 +1
constant InitCrc: std_logic_vector := “1010”;

procedure SerialCalculateCRC4(CRC: input std_logic_vector;Data: in std_logic) is
variable Xor1: std_logic;
begin
Xor1 := CRC(3) xor Data;
CRC := CRC(2 downto 0) & ‘0’; -- Shift left 1 bit
if Xor1 = ‘1’ then
   CRC := CRC xor CRCPoly
end if;
end SerialCalculateCRC4;
```

4.8 Message Size Special Cases

The response to any 8-bit message is expected to be another 8-bit message and the response to any 16-bit message is expected to be another 16-bit message. All DSI messages are initiated by the master MCU through the DSI/D. This cuases to some special cases when there is a transition from one message size to a different message size. The first DSI messages that set up the addresses of the DSI system peripherals are 16-bit messages. The firing commands are also 16-bit messages. All other messages are eight bits.

When the previous message was eight bits and the current message is 16 bits, the response message (which is also eight bits) finishes before the current message frame and the CRC bits look like data bits 7 through 4 in the new 16-bit message format. Since the CRC validation of this 8-bit message response is not reliable, this 8-bit response should not be used.

When the previous message was 16 bits and the current message is eight bits, the response message (which is also 16 bits) cannot finish before the current message frame. The last four data bits and the four CRC bits are lost. Bits 7 through 4 of the 16-bit response message look like the CRC bits of an 8-bit response and almost certainly would not be correct. Since the response is incomplete and the CRC check is not valid, this response is not useful.

The 16- to 8-bit message size transition normally only occurs after setting up the addresses of the DSI bus peripherals. During address setup, a message with address 0000 is sent to attempt to set the address of the next peripheral on the daisy-chained bus. Before any peripherals have been assigned an address, their bus switches are opened so the addressing message only goes to the first peripheral in line. As each peripheral gets an address, it closes its bus switch so the next address assignment command can reach the next peripheral in line on the bus. Each peripheral responds to an address assignment only once (during the next message after the command that set its address). When the master MCU fails to receive a response, it knows it has passed the last peripheral, and that the (16-bit) address assignment command that received no response will have no response either. At this point, the master will begin sending 8-bit messages and the first such message frame will have no meaningful response associated with it.

The first message after reset is also a special case because there was no previous message and therefore there will be no meaningful response during the first message transfer.

Section 5. Timing and Electrical Specifications

5.1 Maximum Ratings

Maximum ratings summarized here describe the worst case conditions that the device can be subjected to without risk of erroneous operation or permanent damage.

Parameter	Symbol	Min	Max	Units
Maximum voltage on input pins V_{DD}, SCLK, CLK, DI, DO, $\overline{CS}$, $\overline{RESET}$, DSI1R, DSI0R	V_{In}	–0.3	7.0	V
Operating temperature	T_A	–40	85	°C
Storage temperature	T_{stg}	–55	150	°C
Operating junction temperature	T_J	—	150	°C
Lead temperature (soldering, 10 seconds)	T_{Lead}	—	230	°C
Thermal resistance	θ_{JA}	—	125	°C/W
Continuous current per pin	I_D	–5.00	5.00	mA

5.2 DC Electrical Characteristics

Parameter[1]	Symbol	Min	Max	Units
Supply current SCLK = 450 kHz	I_{DD}	—	3	mA
Input high voltage SCLK, CLK, DI, $\overline{CS}$, $\overline{RESET}$, DSI1R, DSI0R	V_{IH}	$0.7*V_{DD}$	$V_{DD}+0.3$	V
Input low voltage SCLK, CLK, DI, $\overline{CS}$, $\overline{RESET}$, DSI1R, DSI0R	V_{IL}	–0.3	$0.3*V_{DD}$	V
Output high voltage (@I_{OH} = –800 µA) DO, $\overline{INT}$, DSI1F, DSI0F, DSI1S, DSI0S	V_{OH}	$0.8*V_{DD}$	V_{DD}	V
Output low voltage (@I_{OL} = 800 µA) DO, $\overline{INT}$, DSI1F, DSI0F, DSI1S, DSI0S	V_{OL}	0.0	$0.2*V_{DD}$	V
Input leakage SCLK, CLK, DI, $\overline{CS}$, $\overline{RESET}$, DSI1R, DSI0R	I_{In}	—	±10	µA
I/O port three-state leakage DO	I_{OZ}	—	±10	µA

1. 4.5 volts ≤ V_{DD} ≤ 5.5 volts; –40°C ≤ T_{AMB} ≤ 85°C

5.3 Timing Characteristics for DSI/D to DSI/P Interface

This section describes the AC timing characteristics of the DSI/D to DSI/P interface pins (DSIxF, DSIxS, DSIxR). The signal bit time is derived by dividing the master clock input (SCLK) by 3, 6, 12, or 24 (see **Figure 3-7** description of CDIVx[B:A] control bits). The bit time (t_{Bit}) is then used as the basis for other timing specifications.

Parameter[1]	Symbol	Min	Typ	Max	Units
Communication rate (MC68HC55 to DSI/P)	—	5	—	150	kbits/s
Signal bit time ($t_{SCLKCYC}$ * 3, 6, 12, or 24)	t_{Bit}	6.67	—	200	µs
Master clock cycle time	$t_{SCLKCYC}$	2.22	$t_{Bit}/3$	66.7	µs
Master clock duty cycle	—	40	50	60	%
Frame start to signal delay time	t_{DLY1}	t_{Bit}–0.2	t_{Bit}	t_{Bit}+0.2	µs
Signal end to frame end delay time	t_{DLY2}	–0.2	0	0.2	µs
Signal low time for logic 0 (33.3% duty cycle guaranteed by design)	$t0_{LO}$	$^2/_3t_{Bit}$–0.2	$^2/_3t_{Bit}$	$^2/_3t_{Bit}$+0.2	µs
Signal low time for logic 1 (66.7% duty cycle guaranteed by design)	$t1_{LO}$	$^1/_3t_{Bit}$–0.2	$^1/_3t_{Bit}$	$^1/_3t_{Bit}$+0.2	µs
Receive data setup — DSI1R, DSI0R	t_{RSU}	20	—	—	ns
Receive data hold — DSI1R, DSI0R	t_{RH}	20	—	—	ns
Rise time (20% V_{DD} to 70% V_{DD}) DSI1F, DSI1S, DSI0F, DSI0S	t_{Rise}	—	—	100	ns
Fall time (70% V_{DD} to 20% V_{DD}) DSI1F, DSI1S, DSI0F, DSI0S	t_{Fall}	—	—	100	ns

1. 4.5 volts ≤ V_{DD} ≤ 5.5 volts; –40°C ≤ T_{AMB} ≤ 85°C; C ≤ 100 pF load on all DSI/D to DSI/P pins

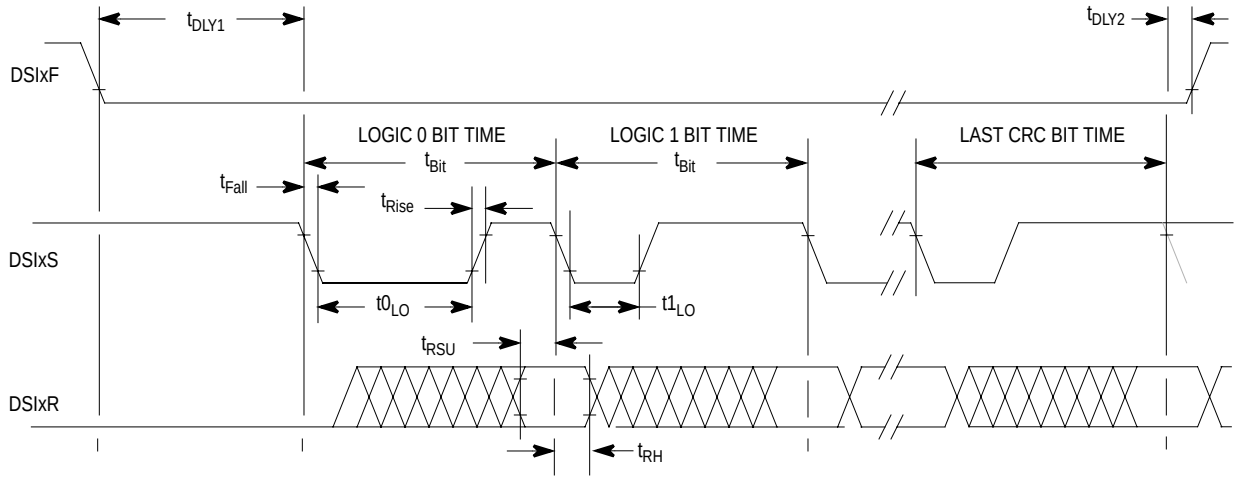


Figure 5-1. DSI/D to DSI/P Interface Timing

5.4 Timing Characteristics for SPI Interface

The table and **Figure 5-2** describe AC timing characteristics of the SPI pins. This timing is compatible with a Freescale SPI system that is set up as a master with CPHA = CPOL = 0. **Figure 5-3** is a general timing diagram for a single SPI transfer.

Parameter[1]	Symbol	Min	Typ	Max	Units
SPI clock frequency	f_{SPI}	0	—	4	MHz
SPI clock cycle time	t_{CYC}	227	—	—	ns
SPI clock high time	t_{HI}	91	125	—	ns
SPI clock low time	t_{LO}	91	125	—	ns
SPI $\overline{CS}$ lead time	t_{Lead}	227	—	—	ns
SPI $\overline{CS}$ lag time	t_{Lag}	227	—	—	ns
Data setup time DI valid before CLK rising edge	t_{SU}	25	—	—	ns
Data hold time DI valid after CLK rising edge DO valid after CLK falling edge	 t_H t_{HO}	 25 0	 — —	 — —	ns
Access time $\overline{CS}$ fall to DO valid	t_A	—	—	100	ns
Data valid time CLK falling edge to DO valid	t_V	—	—	100	ns
Output disable time $\overline{CS}$ rise to DO Hi-Z	t_{DIS}	—	—	100	ns
Rise time (20% V_{DD} to 70% V_{DD}) CLK, DO	t_R	—	—	25	ns
Fall time (70% V_{DD} to 20% V_{DD}) CLK, DO	t_F	—	—	25	ns

1. 4.5 volts ≤ V_{DD} ≤ 5.5 volts; –40°C ≤ T_{AMB} ≤ 85°C; C ≤ 200 pF load on all SPI pins

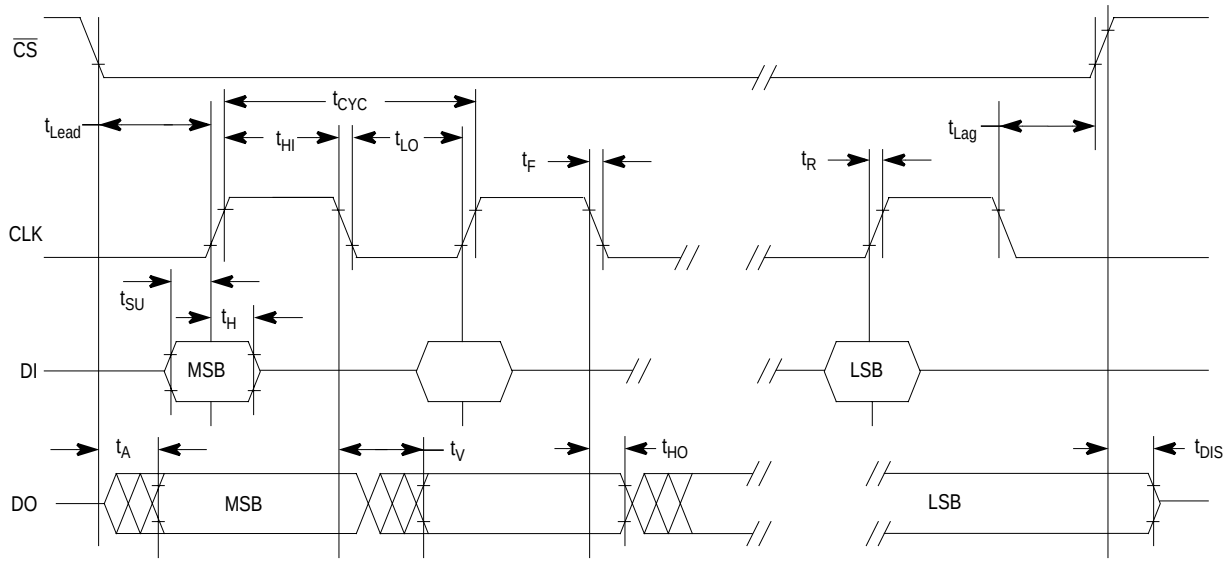


Figure 5-2. SPI Interface Timing

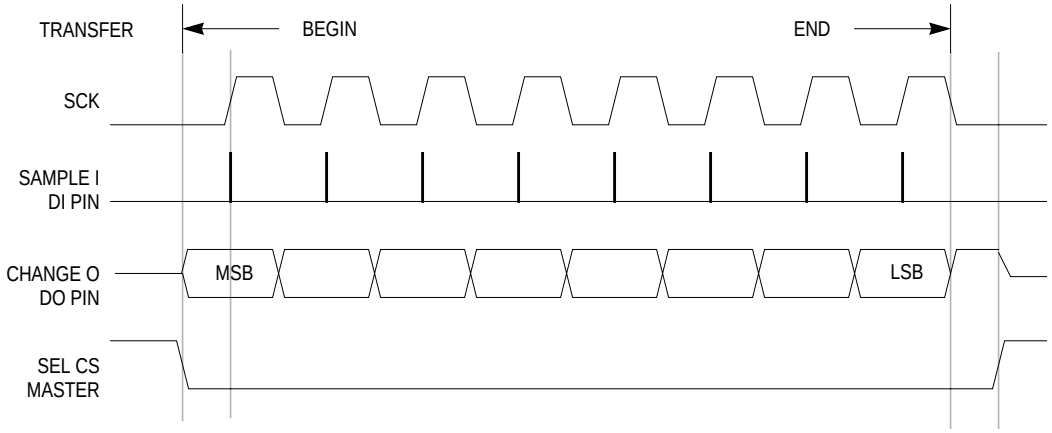
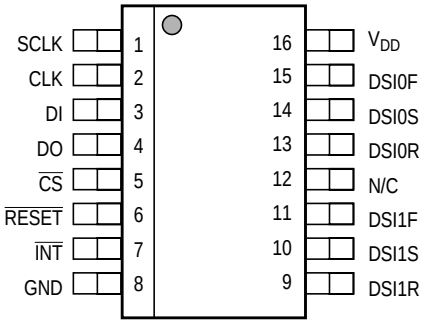


Figure 5-3. SPI Timing Diagram for a Single Transfer

Section 6. Mechanical Data and Ordering Information

6.1 Pin Assignments

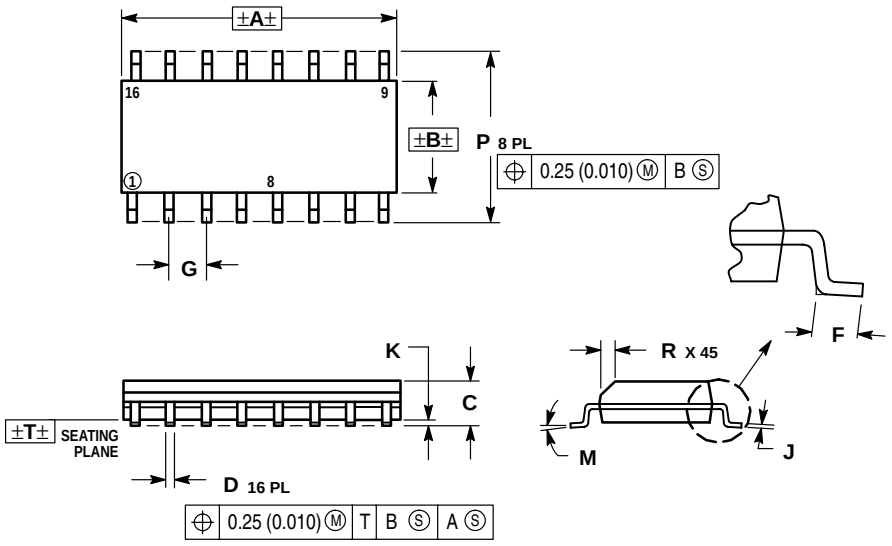
The MC68HC55 is available in the 16-pin small outline integrated circuit (SOIC) package.



16-lead narrow body SOIC, package #751B-05 issue J

Figure 6-1. MC68HC55CD Pin Assignments

6.2 Mechanical Data



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION.
4. MAXIMUM MOLD PROTRUSION 0.15 (0.006) PER SIDE.
5. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.127 (0.005) TOTAL IN EXCESS OF THE D DIMENSION AT MAXIMUM MATERIAL CONDITION.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	9.80	10.00	0.386	0.393
B	3.80	4.00	0.150	0.157
C	1.35	1.75	0.054	0.068
D	0.35	0.49	0.014	0.019
F	0.40	1.25	0.016	0.049
G	1.27 BSC		0.050 BSC	
J	0.19	0.25	0.008	0.009
K	0.10	0.25	0.004	0.009
M	0	7	0	7
P	5.80	6.20	0.229	0.244
R	0.25	0.50	0.010	0.019

Figure 6-2. 16-pin SOIC Package Dimensions

6.3 Ordering Information

Package	Temperature Range	MC Order Number
16-pin SOIC	–40°C to 85°C	MC68HC55CD

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 Muenchen, 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
support.asia@freescale.com

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

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics of their non-RoHS-compliant and/or non-Pb-free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to http://www.freescale.com/epp.

MC68HC55
Rev. 2, 10/2006

freescale™
semiconductor