Freescale Semiconductor
Application Note

AN3670
Rev. 1.0, 10/2008

Implementing a DSI Network Using the MC33781 (Master) and the MC33784 (Slave)

1 Purpose

This application note describes the main features of a Distributed System Interface (DSI) system designed with the MC33781/MC33784. It discusses system fault modes and practical implementation features such as programming and PCB layout.

2 Scope

This note is applicable to the DSI Bus Standard Version 2.02 (March 2005).

3 Introduction

The MC33781 is a DSI Bus Master device, providing four differential DSI 2.02 buses in a single package. It contains the logic to interface the buses to a standard serial peripheral interface (SPI) port and the analog circuitry to drive data and power over the bus as well as receive data from remote DSI slave devices. Figure 1 shows a simplified block diagram of the MC33781.

Contents

freescale™ semiconductor

The MC33784 is a DSI 2.02–compatible Slave device, optimized as a sensor interface. It contains circuits to provide power and A/D conversion for a device such as the Freescale MMA1200EG micro-machined +/-250 G accelerometer. Figure 2 shows a simplified block diagram of the MC33784.

A central module MCU such as one of the Freescale MAC7xxx family of 32-bit automotive microcontrollers provides system control and communicates with the MC33781 via the SPI.

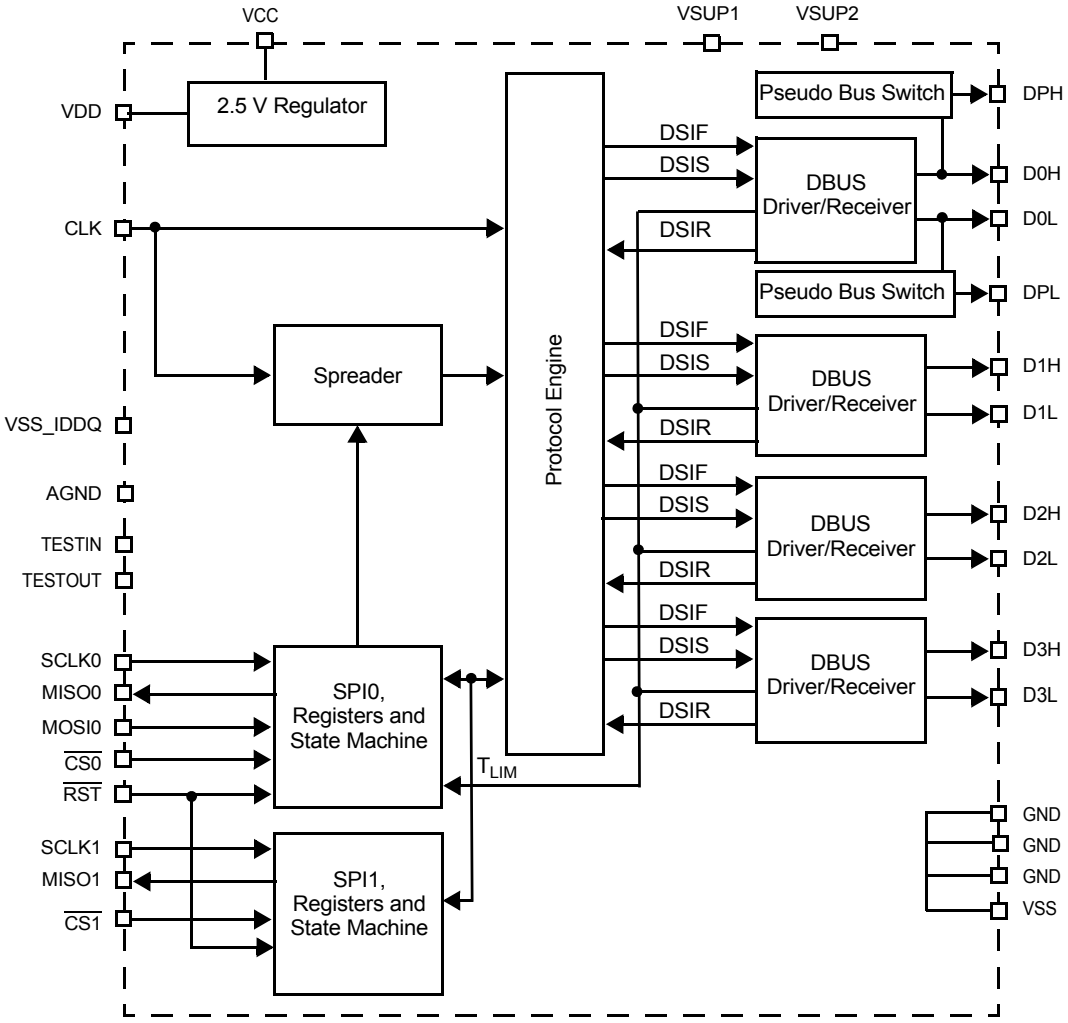


Figure 1. Block Diagram for the 33781, Master

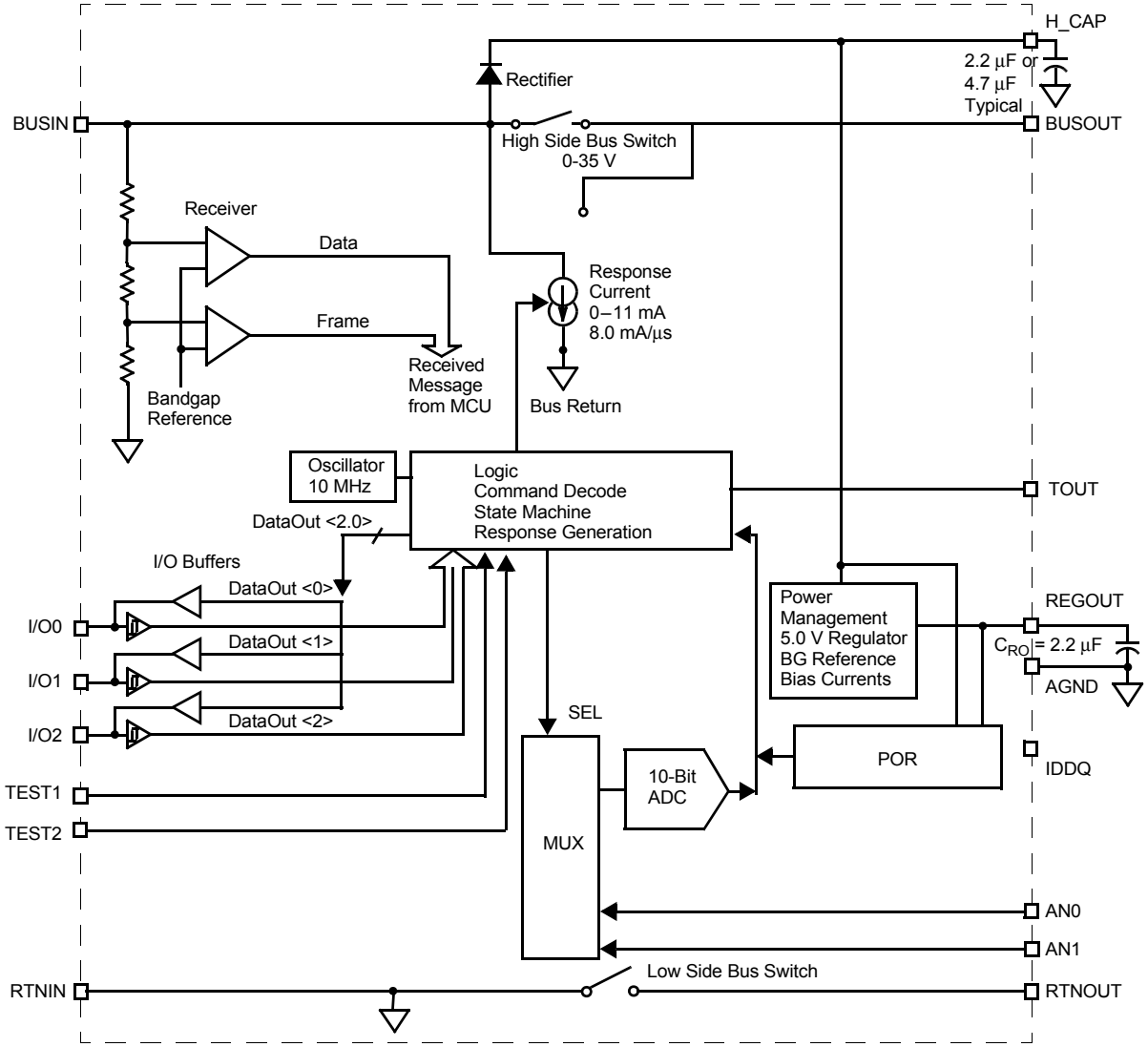


Figure 2. Block Diagram for the 33784, Slave

Table 1. MC33781 (Master) and MC33784 (Slave) Key Features

Key Feature	Comment
Baud rate	Max 200 kbps
# of DSI channels (Master)	4 + 1(Pseudo BUS)
Max # of Slave Devices	60*
A/D resolution (Slave)	10 bit
Receive current (Master)	High, Low, Sum
Number of Bus Switches (Slave)	2 (High side and low side)
SPI interface (Master)	Dual (one is read only)
Package (Master)	SOICW-32
Package (Slave)	SOICN-16

(*) theoretical maximum – practical maximum will be limited by update rate, power consumption, and other factors.

4 Controlling The MC33781 (Master)

The host MCU uses the SPI0 port to access the control and status registers of the MC33781 and to send and retrieve data over the DSI channels. 16 bit messages (command byte plus one data byte) are used to access the MC33781 control registers; 32 bit messages (command byte plus three data bytes) can both access control registers and queue up transfers over the DBUS. Up to four 16 bit slave commands can be queued for transmission over a particular DBUS channel.

All SPI0 transactions between the host MCU and the MC33781 are either 16 or 32 bits long. The MC33781 counts the number of clocks received during a frame. A framing error occurs if the count is not 16 or 32; the received message is discarded and no registers are updated. In addition,

$\overline{\text{CS0}}$ going low signifies the start of the frame. A transaction begins with a command byte, followed by either 1 or 3 bytes of data.

Figure 3 and Figure 4 show both 16 bit and 32 bit transfers. In these multi-byte transfers, as long as CS0 is asserted low, each additional byte sent over the SPI0 will be a read/write of data to the sequential next register.

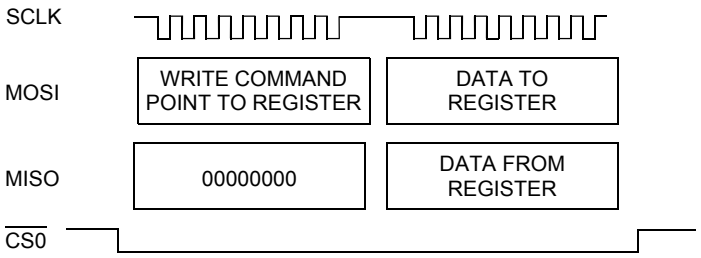


Figure 3. SPI016-Bit Burst Transfer Example

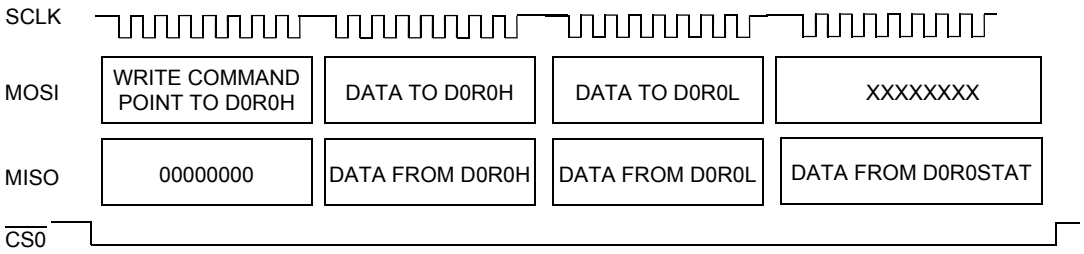


Figure 4. SPI0 32 Bit Burst Transfer Example

The first bit sent (bit 7) of the command byte signals a read or write (write = 1) of data. The last seven bits (bits 6….0) of the command byte constitute the address of the desired register.

For example, the address of the control register for DBUS channel 0 (D0CTRL) is $x0C, so the SPI0 sequence $8C 41 will cause $41 be written to the D0CTRL register.

Similarly, the address of the DBUS 0 high byte data register D0R0H is $x00; a sequence of $80 77 23 will result in $77 and $23 being written into D0R0H and D0R0L respectively. The data $77 23 will subsequently be transmitted over channel 0 as an DSI command.

In a read operation, bit 7 of the command byte is a zero, so for example a command of $06 xx xx xx will result in reading the registers at addresses $06, 07 and 08 (D0R2H, D0R2L and D0R2STAT respectively).

See the MC33781 data sheet for the addresses and bit assignments of the various registers. Note that some register addresses are reserved; writes or reads to these addresses will be ignored.

5 Data Flow Through the DSI System

Data transfer from the SPI0 bus to the DBUS is through the sets of data registers D0R0H/D0R0L through D0R3H/D0R3L. Each set – D0R0H/D0R0L, for example - consists of two physical pairs of registers, one for transmit and one for receive.

Data is written to and read from each register asynchronously; synchronization between the two operations is by means of the TE and RNE status bits.

5.1 MC33781 (Master)

The TE (Transmit Register Empty) bit for a register pair is cleared, indicating transmit buffer not empty, on the rising edge of $\overline{CS0}$ after an SPI0 write to that register; it is set (transmit buffer empty) once the data has been sent out over the DBUS.

Conversely, the RNE (Receive Register Not Empty) bit is set when the DBUS writes to the associated register pair and cleared on the rising edge of $\overline{CS0}$ after an SPI0 read of that pair.

The MC33781 can queue up to four sequential bus commands per channel for DBUS transmission.

If the transmit queue is empty, the delay from data being received as an SPI0 message until it starts to appear as a DBUS transmission is given by the parameter TDBUSSTART2. This is the time from the rising edge of $\overline{CS0}$ to the start of the DBUS transition from idle mode to signal mode, as illustrated in Figure 5.

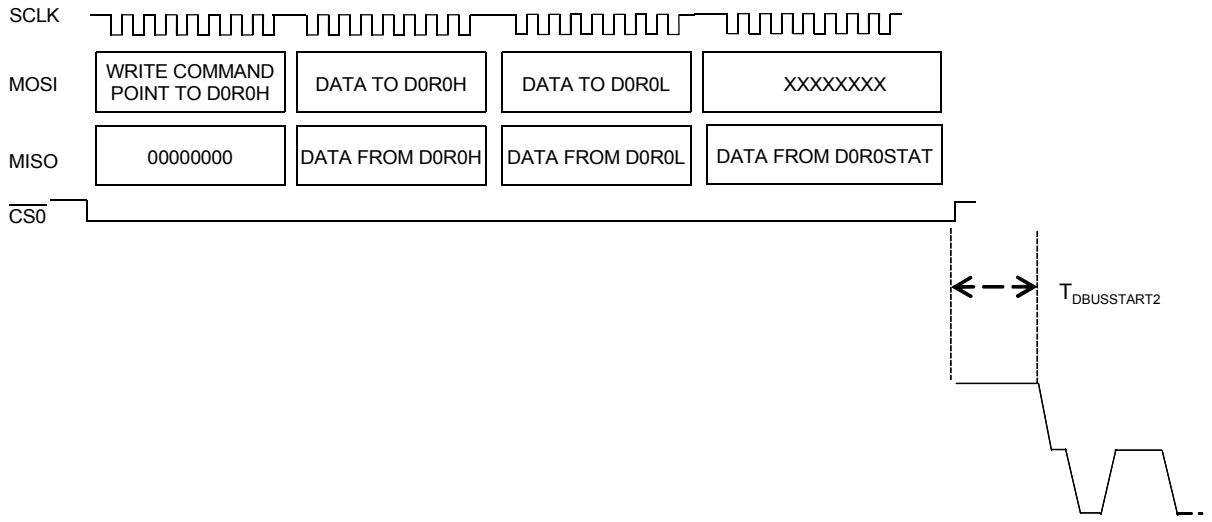


Figure 5. TDBUSSTART2

$T_{DBUSSTART2}$ has a maximum and minimum value of:

$$\frac{2t_{BIT}}{3} + (DLY - 2) \times t_{BIT} < T_{DBUSSTART2} < \frac{5t_{BIT}}{3} + (DLY - 2) \times t_{BIT} \quad \mu s.$$

where t_{BIT} is the bit time and DLY is the delay between DBUS frames, set in the DnCTRL register to be either 4, 5, 6, or 8 bit times.

For example, with t_{BIT} =5.0 µs (200 kbps DBUS speed) and 4 bits interframe delay:

$13.33 < T_{DBUSSTART2} < 18.33$ µs

If the transmit queue is not empty, the transmission delay is dependent on the number and length of messages ahead of the incoming SPI0 message.

5.2 MC33784 (Slave)

After the MC33784 detects that the DBUS message is complete (signified by the bus voltage exceeding the Frame Threshold), there is a delay of approximately 4.6 µs while the CRC is calculated, indicated by t_{CRC_CALC} in Figure 6. Once the CRC is verified, the requested action is carried out.

Figure 6 also shows a ~200 ns turn-on delay in the MC33784 (t_{TOD}) when a logic output is changed from low to high.

See for a Figure 7 representation of a typical data flow.

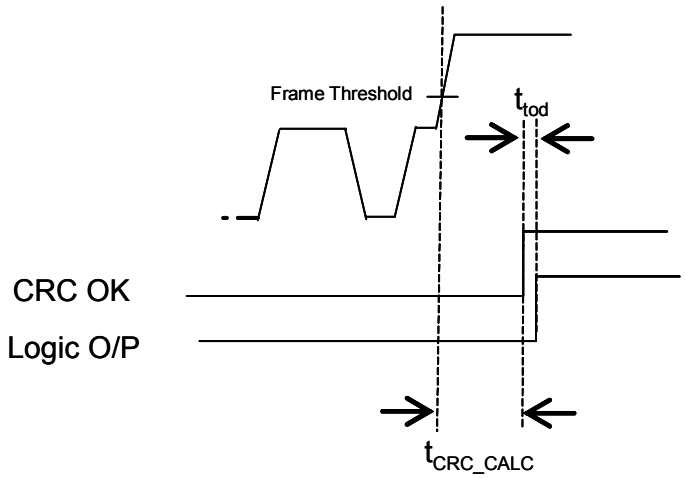


Figure 6. t_{CRC_CALC}

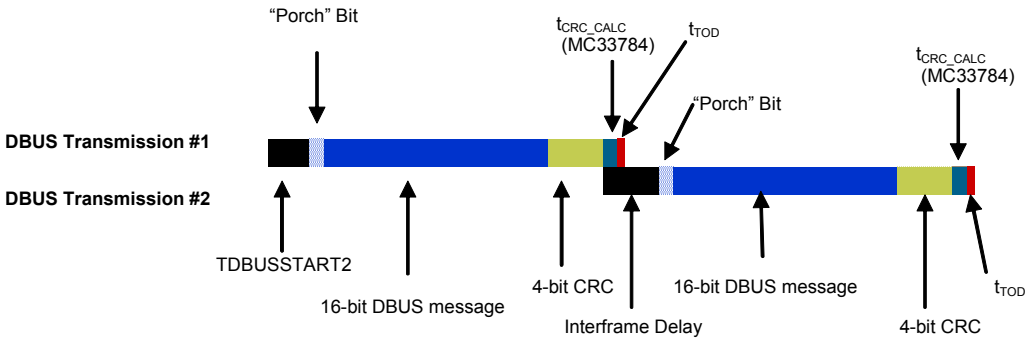


Figure 7. Data Flow for Two Consecutive 16 Bit DBUS Transmissions with a 4 Bit Interframe Delay

5.3 Bus Initialization Sequence

For normal operation, a typical DBUS channel initialization sequence following a reset consists of the following actions. Note that only steps 2 and 5 are required – the others are optional and are used if the default values are not acceptable.

1. Set the transmission parameters (optional).
   a) Minimum delay between DBUS transfer frames -- DnCTRL register bits 4 & 5, default value 4 bit times (00)
   b) Select message size – DnCTRL register bit 0, default value long word (0)
   c) Select loop mode control if required -- DnCTRL register bits 1 & 2, default value disabled (00)
2. Enable the channel (required) -- DnEN register bit 0, default value disabled (0).
3. Set channel CRC polynomial (optional) – DnPOLY register, default value $11.
4. Set channel CRC seed (optional) – DnSEED register, default value $0A.
5. Initialize the slave devices (required) – DnRnH and DnRnL registers.
6. Repeat Steps 1–5 for the other channels as needed.

5.4 Slave Response To Initialization Sequence

After a reset, all slave bus switches will be open. The initialization command sets the slave address, configures the digital I/O, and other operations. Any commands received prior to the initialization command are ignored by the slave device. After receipt of a valid initialization command, a slave will close its bus switches and ignore subsequent initialization commands.

In the daisy chain configuration the sequence for a channel is as follows:

Table 2. Daisy Chain Configuration

Command →	POR	Init (Address 1)	Init (Address 2)	Init (Address 3)	Init (Address 4)
Slave 1	Bus switch open	Initialized to address 1, bus switch closed	Initialization response		
Slave 2	Bus switch open	Bus switch open	Initialized to address 2, bus switch closed	Initialization response	
Slave 3	Bus switch open	Bus switch open	Bus switch open	Initialized to address 3, bus switch closed	Initialization response
Slave 4	Bus switch open	Bus switch open	Bus switch open	Bus switch open	Initialized to address 4, bus switch closed

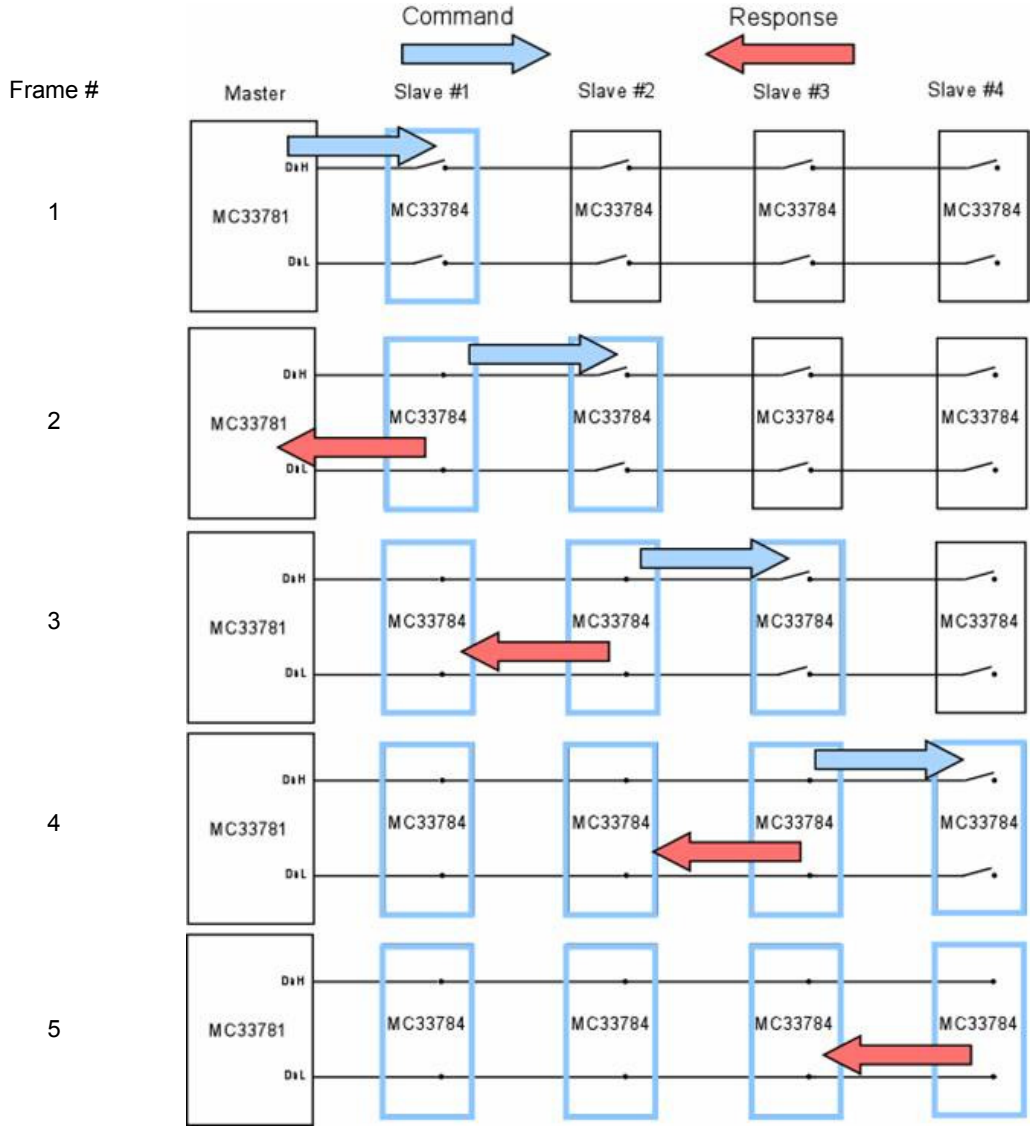


Figure 8. Graphical Representation of the Command/Respond Sequence

The message formats for the initialization commands can be found in the MC33784 data sheet in the “Logic Commands And Registers” section.

As an example, here is the initialization sequence for a MC33781 communicating with 6 MC33784 slaves – four on DBUS 0 and two on the Pseudo Bus.

Table 3. 33781 Initialization Sequence

Hex Command	Action
8D 01 11 0A	Initialize DBUS Channel 0
80 61 00 00	Assign Bus 0 Address 1
80 62 00 00	Assign Bus 0 Address 2
80 63 00 00	Assign Bus 0 Address 3
80 64 00 00	Assign Bus 0 Address 4
BD 07 11 0A	Initialize Pseudo Bus
80 65 00 00	Assign Bus 0 Address 5
80 66 00 00	Assign Bus 0 Address 6

The pseudo bus requires a slightly different initialization sequence. See “Using The Pseudo Bus” below for a description.

6 Fault Detection And Correction

6.1 MC33781 (Master) Power-On Check

There are 128 fuse bits in the MC33781 that are used to trim various voltage and current parameters during manufacturing. After a reset the MC33781 reads all of the fuse bits and transfers the information to internal flip-flops. At the same time, the fuse parity bit is checked and verified. If the check fails, a fuse parity error bit is set (MASKID register bit 7).

6.2 MC33781 (Slave) Loopback Mode

The MC33781 has a loopback mode that enables testing of the transmit and receive circuits without sending data out over the bus. Loopback is selected for a channel by setting the two control register (DnCTRL) Loop Mode bits Loop0 and Loop1 to 11. Any other combination will disable loopback.

During loop back a multiplexer selects the DSIS (transmit) signal instead of the DSIR (receive) signal for loading into the receive buffers. See Figure 9 for the signal path.

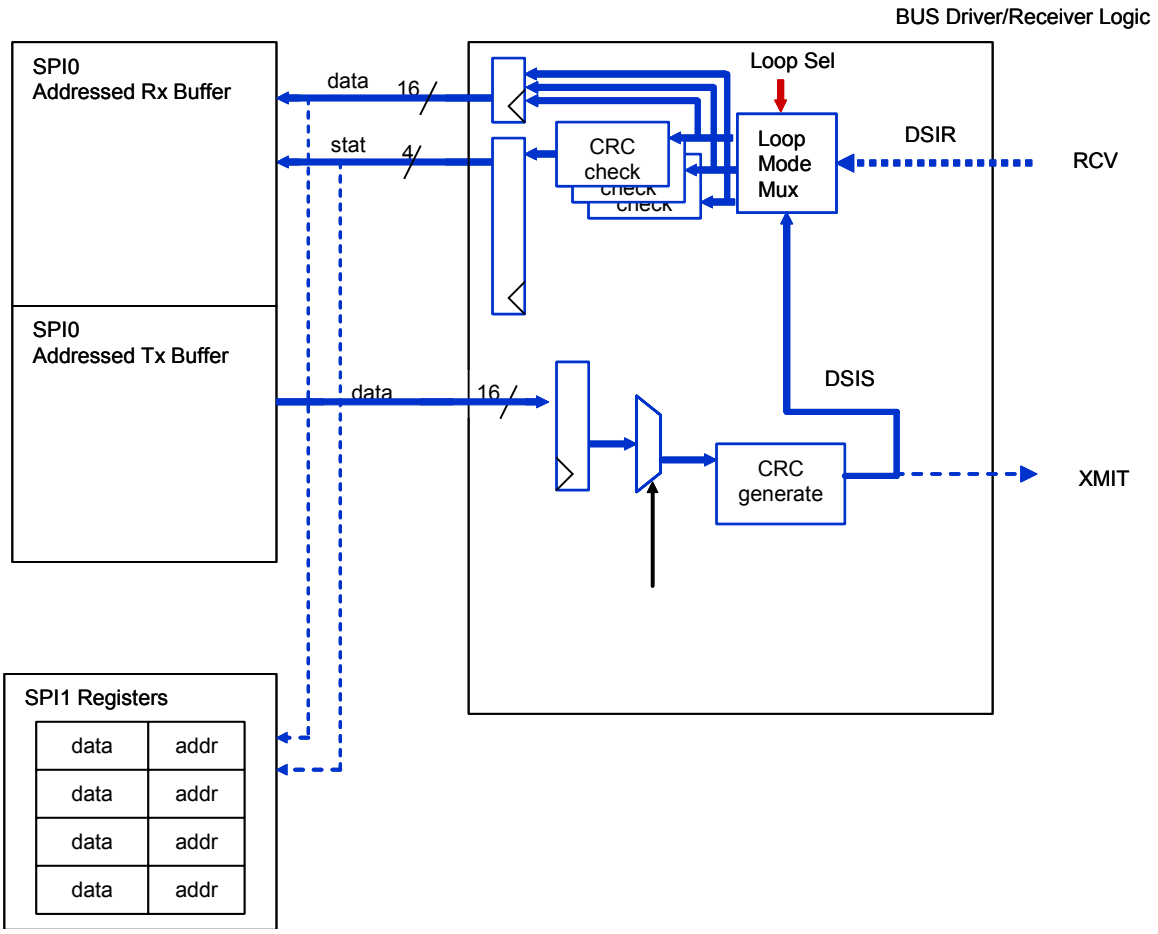


Figure 9. Loopback Mode Signal Path

DBUS transmitter or receiver fault conditions will not affect the test when in loopback. Loopback mode clears the EN bit (DnEN register bit 0) for the channel, so it must be re-enabled before transmitting data.

Both SPI0 and SPI1 operate during loopback mode. However, the write to the SPI1 buffer only occurs if the data is all 1's or all 0's and the address is the complement of the data.

6.3 CRC Generation & Checking

Both the MC33781 and MC33784 add a variable length CRC to all messages before transmission. An incorrect CRC indicates that the received data is not valid.

The default CRC is 4 bits long and uses a polynomial of $X^4 + 1$ (00010001) and a seed value of 1010. The CRC can be changed to between 0 and 8 bits in length and the polynomial and seed value are selectable, allowing the system designer to tailor the CRC coverage level.

The polynomial $X^4 + 1$ has a length of 5 bits, ignoring leading zeros. It can be shown that such a polynomial can detect all burst errors of length ≤ 5. For a burst error of > 5 bits, the probability of

the frame being accepted as valid is 1 in 2^4 (1 in 16) , assuming that all bit patterns are equally likely.

6.4 MC33781 (Master) Receiver Decision Block

In the presence of faults or common-mode noise, it is possible that the DnH and DnL current inputs will give conflicting inputs. Consequently, the MC33781 uses a receiver decision block to determine whether it has received a valid message.

The receiver decision logic block has three inputs. Two are derived from the Receiver High input and the Receiver Low input; the third is the sum of the Receiver High and Receiver Low input, called Receiver Sum.

Table 4 shows the MC33781 Receiver Decision Block and its response to various bus pin conditions. The MC33784 draws a nominal 6.0 mA to indicate a logic high; a logic high on both Receiver High and Receiver Low inputs will result in 12 mA on Receiver Sum.

Table 4. MC33781 (Master) Receiver Decision Logic

Bus Pin Conditions	Receiver High 6 ± 1 mA	Receiver Low 6 ± 1 mA	Receiver Sum 12 ± 6mA	High and Low XOR (bit/bit)	High and Sum XOR (bit/bit)	Low and Sum XOR (bit/bit)	ER Bit	SPI0 DnRnxData	SPI1 DnRnxData
Normal	CRC Ok	CRC Ok	CRC Ok	H*L Ok	N/A	N/A	0	Receiver High	Receiver Low
				H*L Not OK			1	Receiver High	Receiver Low
Out of Spec	CRC Ok	CRC Ok	Bad CRC	H*L Ok	N/A	N/A	0	Receiver High	Receiver Low
				H*L Not OK			1	Receiver High	Receiver Low
Fault	CRC Ok	Bad CRC	CRC Ok	N/A	H*S Ok	N/A	0	Receiver High	Receiver Sum1
					H*S Not OK		1	Receiver High	Receiver Low
Fault L	CRC Ok	Bad CRC	Bad CRC	N/A	N/A	N/A	1	Receiver High	Receiver Low
Fault	Bad CRC	CRC OK	CRC OK	N/A	N/A	L*S Ok	0	Receiver Sum0	Receiver Low
						L*S Not OK	1	Receiver High	Receiver Low
Fault H	Bad CRC	CRC Ok	Bad CRC	N/A	N/A	N/A	1	Receiver High	Receiver Low
Common Mode Noise	Bad CRC	Bad CRC	CRC Ok	N/A	N/A	N/A	0	Receiver Sum0	Receiver Sum1
Fault	Bad CRC	Bad CRC	Bad CRC	N/A	N/A	N/A	1	Receiver High	Receiver Low

Note that SPI0 and SPI1 derive their output data from different sources. SPI0 uses either Receiver High or Receiver Sum0; SPI1 uses either Receiver Low or Receiver Sum1.

In order to provide the maximum protection against a single-point failure causing a disruption in communication, the decision paths for the two SPI channels are independent internally, even when the signal names are the same. For example, the Receiver Sum path is divided into Receiver Sum0 and Receiver Sum1, which use different holding registers in the Receiver logic. These registers are duplicates, although they will always hold the same data unless there is a fault in one of the data paths. See Figure 10.

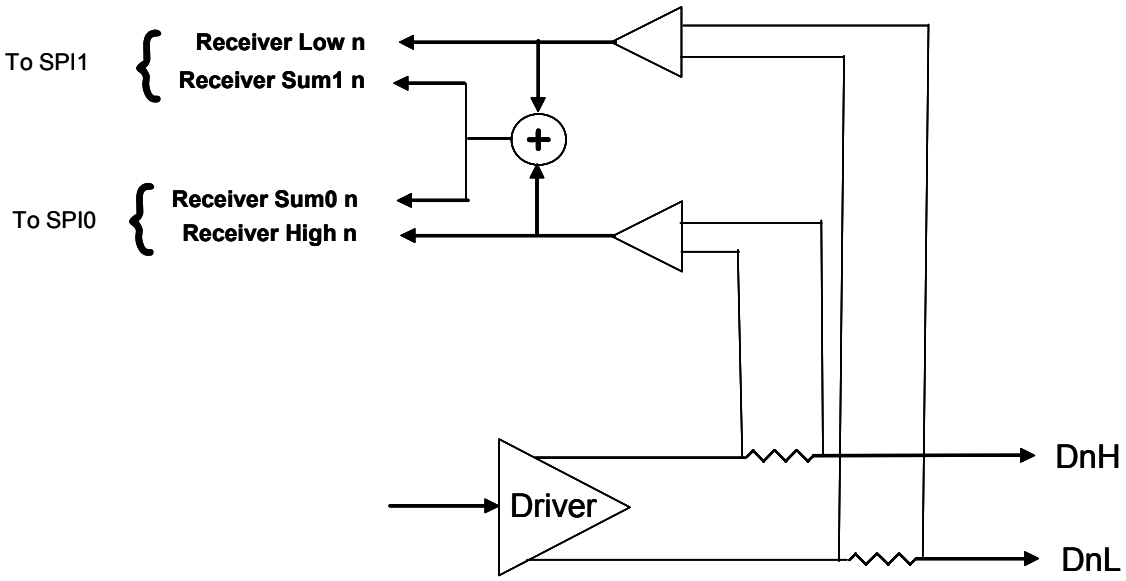


Figure 10. Single Point Failure Protection

7 System Response To Fault Conditions

The MC33781 register set contains several registers that indicate channel status and fault conditions. Here are the applicable registers for DBUS channel 0:

Table 5. MC33781 (Master) Register Set for DBUS Channel 0

Register Address	Register Name	Register Definition	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0000010	D0R0STAT	DBUS 0 Reg 0 Status	ER	TE	SDS	RNE	ICL	0	FIX0	FIX1
0001101	D0EN	DBUS 0 Enable Status	TS	ISDD	-	-	-	BSWH	BSWL	EN

The system response to some possible fault conditions is describe below.

7.1 DnH or DnL Short to GND/VBATT

7.1.1 Hard Short

A hard short of DnH, DPH, BUSIN (MC33784) or BUSOUT (MC33784) to GND, or a hard short of DnL, DPL, RTNIN (MC33784) or RTNOUT (MC33784) to VBATT during idle time will activate the current limit circuit; both drivers will be disabled after a short delay. The ICL bit in the DnRnSTAT register is set for the affected channel.

A hard short of DnH, DPH, BUSIN (MC33784), BUSOUT (MC33784), DnL, DPL, RTNIN (MC33784) or RTNOUT (MC33784) to either GND or VBATT during signal time will result in both drivers being shut down for the rest of the transaction after a short delay. The SDS bit in the DnRnSTAT register is also set. If the over-current limit is reached during two consecutive frames, the bus drivers are disabled and the ISDD bit in the DnEN register is set. In the case of channel 0, the Pseudo Bus switches are also opened and BSWH & BSWL bits are cleared.

Each channel is protected by a thermal shutdown circuit. If the channel bus thermal limit is reached, the bus drivers are disabled and the TS bit in the DnEN register is set.

In the case of channel 0, the Pseudo Bus switches are also opened and BSWH & BSWL bits are cleared. For a thermal shutdown on the Pseudo Bus, however, the Pseudo Bus switches are opened and the BSWH & BSWL bits are cleared, but no other register bits are set and channel 0 operation is unaffected.

7.1.2 Resistive Short

A resistive short during signal mode can be classified according to how much current draw it causes.

- Resistive Short, current draw > 40 mA (typ)

This will cause an over-current shutdown with the same results as a hard short.

- Resistive Short, current draw < 40 mA (typ)

If the resistive short is from DnH to ground then the following actions will occur:

a) Receiver High will report a CRC error (i.e., response current always > 6mA threshold)
b) Receiver Sum will report a CRC error
c) Receiver Low will see response current and report a CRC OK.

This case is described in the “Fault H” row in the Receiver Decision Logic table.

The reverse case for short on DnL is described in the “Fault L” row in Table 4, MC33781 (Master) Receiver Decision Logic.

7.2 Common-Mode Noise

The MC33781 can withstand up to 20 mA of common-mode noise (40 mA pp max) without causing a CRC error in Receiver Sum, although there will be CRC errors in both Receiver High and Receiver Low, since their respective response currents are above the 6mA threshold under all conditions.

This condition is described in the “Common Mode Noise” row in Table 4. No ER bit is set and Receiver Sum Data is used.

Above 20 mA common mode noise, a CRC error will occur and the ER bit will be set, as described in the last row of Table 4.

7.3 DnH to DnL Short Circuit

Shorting the high and low side of a DBUS channel together will result in all 1’s on both Receiver High and Receiver Low, including all of the CRC bits, giving a CRC error.

Shorting high and low side of a DBUS channel together will result in reporting a fault condition with a CRC error and the ER bit will be set.

7.4 DH to DL Resistive Short

Depending on the resistive short value, either Receiver High or Low will report all 1's and the ER bit will be set.

7.5 DnH or DnL Open Circuit

An open circuit on DnH or DnL will result in a loss of bus communication. All Receivers High, Low and Sum will report a bad CRC. The ER bit is set and Receiver High wrong data is reported. This is also described by the last row in Table 4.

8 Bus Speed

The allowable data rate between the MC33781 and the MC33784 is variable within limits. The minimum data rate is defined by the DSI specification to be 5 kbits/sec. The maximum data rate is not defined by the specification. In practice, the upper limits will be set by EMC, total bus capacitance and other considerations.

Table 6 shows suggested maximum data rates with common values of slave module input (BUSIN to RTNIN) and output (BUSOUT to RTNOUT) capacitors. The values are based on simulations with the test circuit shown in Figure 11 and are for reference only.

Table 6. Maximum Data Rates

Slave Module Capacitance (C_{in}/C_{out})	Suggested Max Data Rate
470 pF / 470 pF	200 kbps
680 pF / 470 pF	185 kbps
1 nF / 470 pF	160 kbps

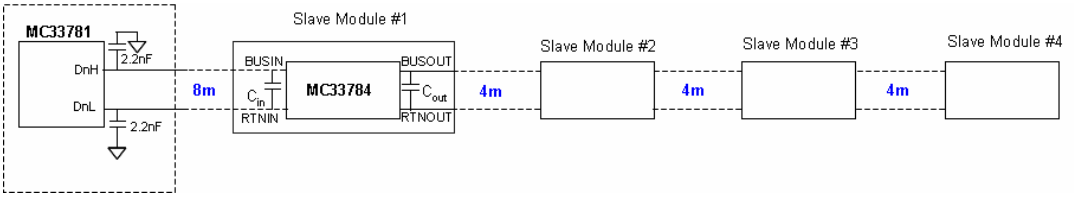


Figure 11. Test Circuit

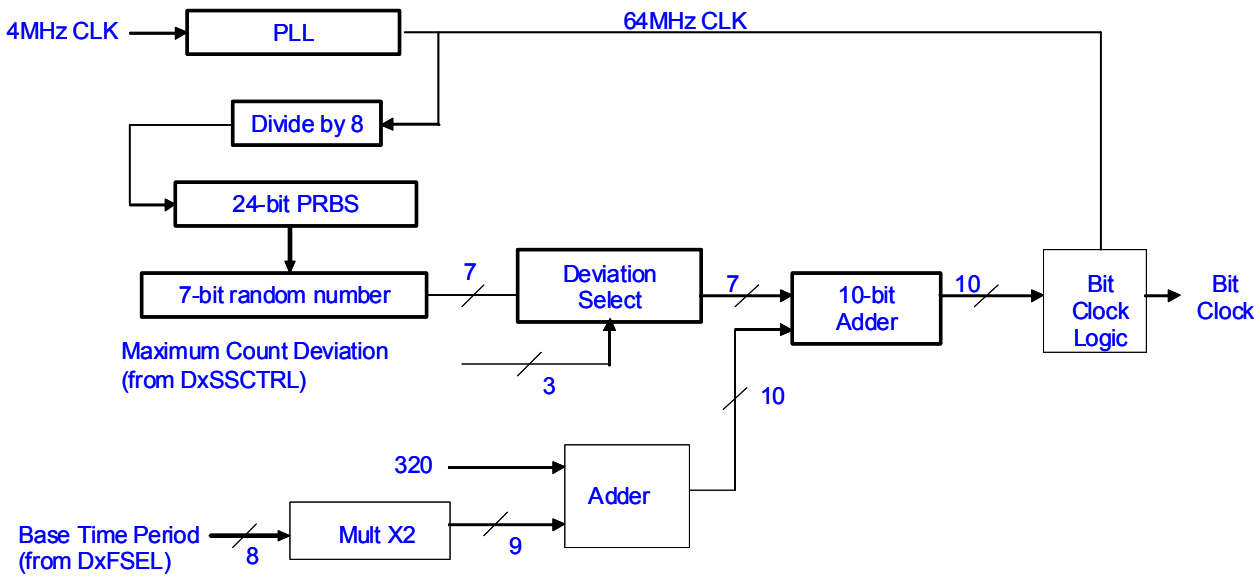
There is a minimum frame delay between MC33781 commands as required by the DSI specification to allow recharging of the energy storage capacitors in the MC33784 or other slave devices. The minimum frame delay required is dependent upon several factors including the bus speed, the current consumption of the slaves and the amount of energy storage in the network.

9 Spread Spectrum Control

The dominant source of radiated electromagnetic interference (EMI) from the DBUS bus is due to the regular periodic frequency of the data bits. At a steady bit rate, the time period for each bit is the same, which results in a steady fundamental frequency plus harmonics. Consequently, unwanted signals appear at multiples of the fundamental frequency; these can be strong enough to interfere with the desired signal.

A significant decrease in radiated EMI can be achieved by randomly changing the duration of each bit. This reduces the energy amplitude by having the signal spend a much smaller percentage of time at any specific frequency. The signal strength of the fundamental and harmonics are reduced directly by the percentage of time it spends on a specific frequency.

A circuit to accomplish this is included in the MC33781, and can perform the spreading of the signal independently for each channel, while generating the bit clock timing for the channel. The spread spectrum circuitry is discussed in detail in the MC33781 data sheet.



9.1 Choosing The Maximum Deviation Value

It is important to select a maximum deviation value that is appropriate for the system. A larger maximum deviation results in spreading the bit energy to more frequencies. However, this number also establishes the maximum period for any random bit on that channel. If the system requires that a minimum number of bits be transferred within a fixed time period, then the user must select a minimum base bit time and maximum deviation time that will meet the criteria.

Only certain deviations value can be chosen: the allowable deviation values (expressed as the # of 64 MHz clock periods) are selected by the DEV[2:0] bits in the spread spectrum control register (DnSSCTRL) as follows:

DEV[0:2]	# Periods
000	0
001	16
010	32
011	64
100	78

Here are two examples using 10-bit Enhanced Short Words (ESW).

The ESW message contains a total of 14 bits (2 data bits, 4 address bits, 4 command bits, plus 4 CRC bits). The calculations are similar for other message lengths.

Design Criteria: Frame Time = 125 µsec

With a 64 MHz PLL clock, 1 clock period (1 count) = (1/64) x 10^{-6} s = 15.625 ns.

1. Bit rate = 185 kHz with 4 interframe bits

Total # of bits in frame = 19 (1 porch bit + 14 bits enhanced short word message+ 4 interframe bits)

125 μsec frame time → 8000 periods

185 kHz bit rate → 346 periods

19 bits per frame @ 185 kHz bit rate → 19 x 346 = 6574 periods

Maximum counts per frame available for spreading = 8000 – 6574 = 1426 periods

Maximum available counts per bit = 1426/19 = 75 periods

From the table, it can be seen that 64 periods is the appropriate choice.

2. Bit rate = 160 kHz.

125 μsec frame time → 8000 periods

160 kHz bit rate → 400 periods

19 bits per frame @ 160 kHz bit rate → 19 x 400 = 7600 periods

8000 – 7600 = 400 periods total allowable deviation.

Maximum available counts per bit = 400/19 = 21 periods

In this case, a deviation of 16 periods should be chosen.

10 Using The Pseudo Bus

The bus drivers on channel 0 can drive two external bus wire sets (D0H/D0L and DPH/DPL). The pseudo bus allows the use of two redundant sensors (for example two front crash sensors) on a single channel without a wiring fault on one leg affecting the other. Both the high and low sides of channel 0 include a pseudo bus switch as shown in Figure 12.

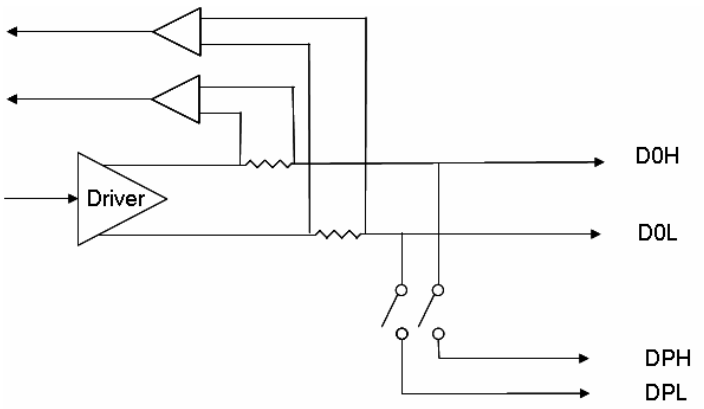


Figure 12. Pseudo bus connection

10.1 Initializing The Pseudo Bus

After a device reset the pseudo bus switches are open. During initialization, the master initializes the slave devices on channel 0 using the sequence described earlier in this application note. The channel 0 slaves should be initialized before the pseudo bus switches are closed, after which the devices on the pseudo bus can be initialized.

Note that when the pseudo bus switches are closed, the channel 0 drivers are driving both DBUS 0 and the pseudo bus in parallel, so there must be no duplication between the DBUS 0 and DBUS P slave addresses.

The pseudo bus switches are controlled by BSWH and BSWL (bits 2 & 1) in the D0EN register. Clearing these bits will open the pseudo bus switches, although they are typically closed during normal operation.

The pseudo bus switches have independent thermal shutdown protection. Once the thermal shutdown point is reached, the bus switch is opened and the BSWH and/or BSWL bit is cleared in the channel 0 DEN register.

11 Using the SPI1 Port

In a typical airbag system, the primary microcontroller transfers commands and data via SPI0. It is important that a failure in the primary microcontroller does not cause an inadvertent airbag deployment, so a secondary microcontroller or ASIC is often used to monitor the incoming sensor data. Both controllers must agree in order for a deployment to occur.

The MC33781 has a second SPI port, SPI1, specifically for this purpose. SPI1 operation is limited to transmitting analog data (AN0 only) from the slave devices on DBUS channels 2 and 3.

SPI0 is bidirectional and communicates data from all channels as well as MC33781 control and status information.

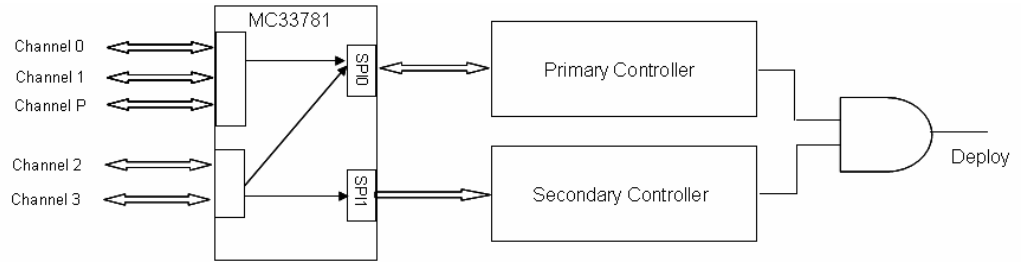


Figure 13. SPI0/SPI1 Communications Flow

The main differences between SPI0 and SPI1 are summarized in Table 7.

Table 7. SPI0 and SPI1

Feature	SPI0	SPI1	SPI1 Comments
SCLK	√	√	
CS	√	√	
MOSI	√	X	
MISO	√	√	Output only
DBUS Channel Data	0,1,2,3,P	2,3	Analog channel 0 only
Transfer # of Bytes	2 or 4	2 only	SPI0 & SPI1 different formats

11.1 SPI1 Operation

The SPI1 control circuitry monitors bus traffic on channels 2 & 3. When a Request AN0 command is detected, it saves the requested slave address together with the response from the next frame. SPI1 then concatenates the requested address, the response data, and two status bits indicating the channel used, and queues this up for a SPI1 read.

Note that if any status bit indicates a bus error, the address and channel bits are still stored in the SPI1 buffer, but the data bits are set to all zeros for this message.

The SPI1 bit definitions are detailed in the MC33781 data sheet.

12 Layout Considerations

12.1 MC33781 (Master)

To ensure stability of the bus drivers, capacitors must be connected between each output and ground. These are the DBUS common mode capacitors. In addition, bypass capacitors are required at the VSUP1 & VSUP2 pins. These capacitors must be located close to the IC pins and provide a low-impedance path to ground.

12.2 MC33784 (Slave)

The layout of the slave module can affect the system performance especially under conditions of high induced bus current – for example, as simulated during BCI (Bulk Current Injection) testing.

BUSIN/RTIN and BUSOUT/RTNOUT capacitors should be used. Typical values are from 400 pF to 1.0 nF.

Separating the RTNIN and AGND traces as much as practical has found to be of value in reducing ADC variation during BCI testing.

13 References

- MC33781 Data sheet
- MC33784 Data sheet

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

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)
www.freescale.com/support

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 China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
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.

AN3670
Rev. 1.0
10/2008

freescale™
semiconductor