



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



Dual DBUS Master with Differential Drive and Frequency Spreading

The 33780 is a master device for two differential DBUS buses. 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 the remote slave devices.

The differential mode of the 33780 generates lower electromagnetic interference (EMI) in situations where data rates and wiring make this a problem. Frequency spreading further reduces interference by spreading the energy across many channels, reducing the energy in any single channel.

Features

- Two Independent DBUS I/Os
- Common SPI Interface for All Operations
- Open-Drain Interrupt Output with Pull-up
- Maskable Interrupts for Send and Receive Data Status
- Automatic Message Cyclical Redundancy Checking (CRC) Generation and Checking
- Four-Stage Transmit and Receive Buffers
- 8- to 16-Bit Messages with 0- to 8-Bit CRC
- Independent Frequency Spreading for Each Channel
- Pb-Free Packaging Designated by Suffix Code EG

33780

*
DIFFERENTIAL DBUS MASTER



ORDERING INFORMATION		
Device	Temperature Range (T _A)	Package
MC33780EG/R2	-40°C to 85°C	16 SOICW
MCZ33780EG/R2		

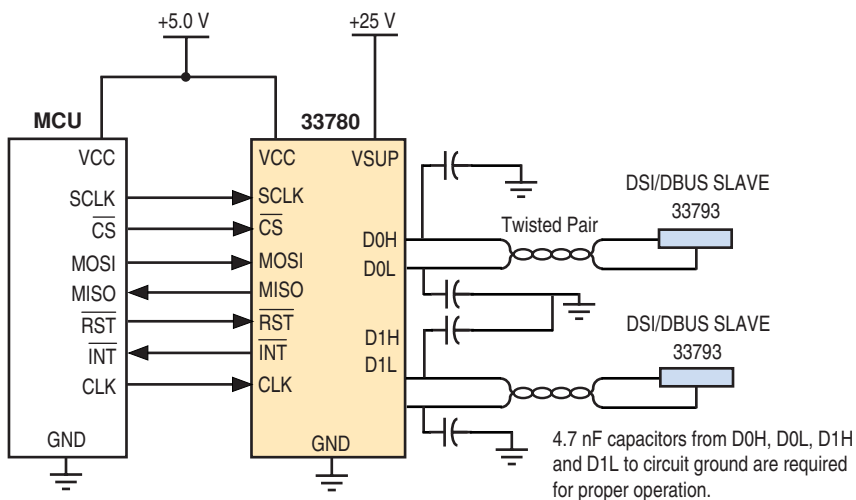


Figure 1. 33780 Simplified Application Diagram

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

INTERNAL BLOCK DIAGRAM

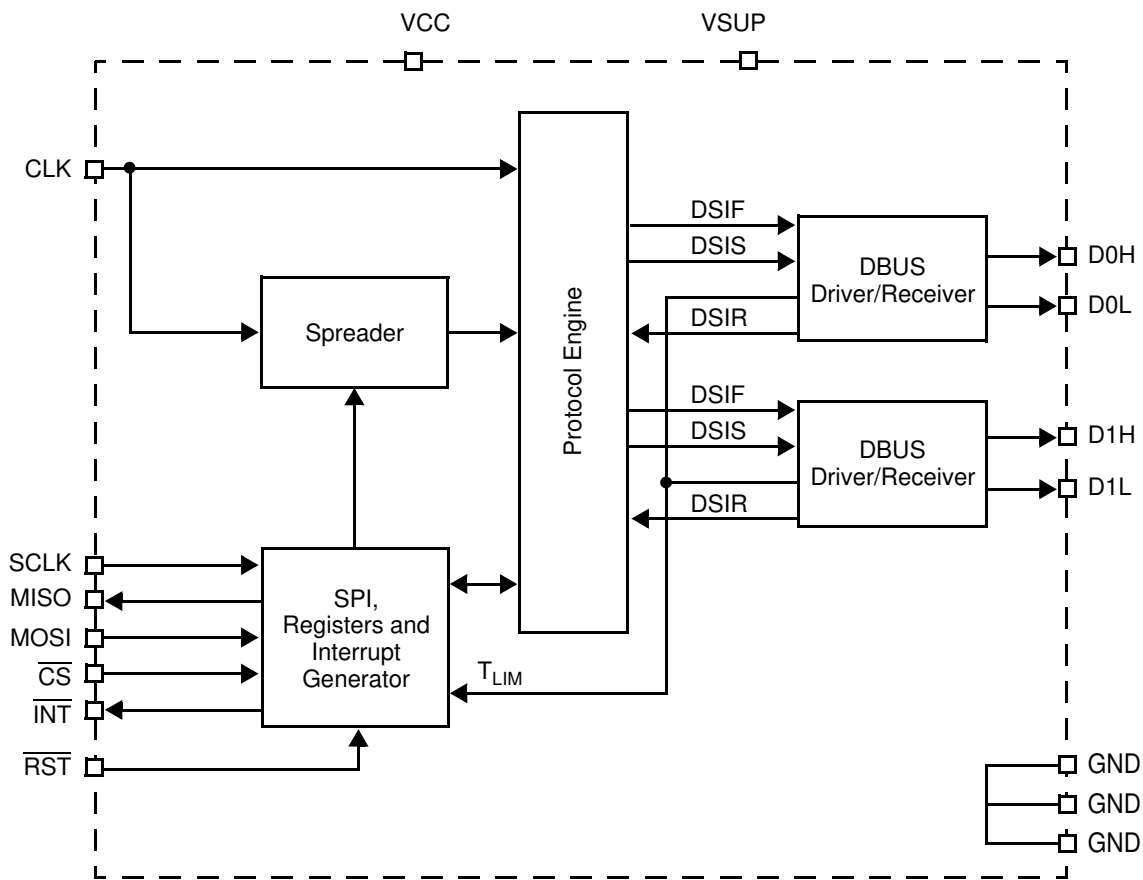


Figure 2. 33780 Internal Block Diagram

PIN CONNECTIONS

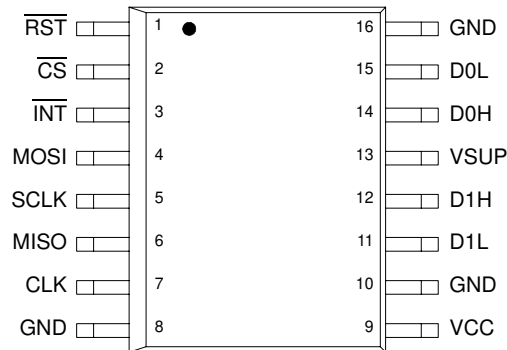


Figure 3. 33780 Pin Connections

Table 1. 33780 Pin Definitions

A functional description of each pin can be found in the Functional Pin Descriptions section beginning on [page 13](#).

Pin	Pin Name	Pin Function	Formal Name	Definition
1	$\overline{\text{RST}}$	Reset	IC Reset	A low level on this pin returns all registers to a known state as indicated in the section entitled Register and Bit Descriptions .
2	$\overline{\text{CS}}$	Input	SPI Chip Select Input	When this signal is high, SPI signals are ignored. Asserting this pin low starts an SPI transaction. The SPI transaction is signaled as completed when this signal returns high.
3	INT	Output	Interrupt Output	This output will be asserted low when an enabled interrupt condition occurs. It contains a pullup current source that will perform a pullup when unasserted.
4	MOSI	Input	Master Out Slave In	SPI data into this IC. This data input is sampled on the positive edge of SCLK.
5	SCLK	Input	Serial Data Clock	Clocks in/out the data to/from the SPI. MISO data changes on the negative transition of the SCLK. MOSI is sampled on the positive edge of the SCLK.
6	MISO	Output	Master In Slave Out	SPI data sent to the MCU by this device. This data output changes on the negative edge of SCLK. When $\overline{\text{CS}}$ is high, this pin is high impedance.
7	CLK	Input	Clock Input	4.0 MHz clock input.
8	GND	Ground	Ground	Ground reference for analog and digital circuits.
9	VCC	Input	Logic Supply	Logic power source input.
10	GND	Ground	Power Ground	Bus 1 power return.
11	D1L	Output Driver	Low-Side Bus 1	Bus 1 low side.
12	D1H	Output Driver	High-Side Bus 1	Bus 1 high side.
13	VSUP	Output	Positive Supply for Bus Output	This supply input is used to provide the positive level output of the bus.
14	D0H	Output Driver	High-Side Bus 0	Bus 0 high side.
15	D0L	Output Driver	Low-Side Bus 0	Bus 0 low side.
16	GND	Ground	Power Ground	Bus 0 power return.

ELECTRICAL CHARACTERISTICS

MAXIMUM RATINGS

Table 2. Maximum Ratings

All voltages are with respect to ground unless otherwise noted. Exceeding these ratings may cause a malfunction or permanent damage to the device.

Ratings	Symbol	Value	Unit
ELECTRICAL RATINGS			
Supply Voltages			V
V_{SUP}	V_{SUP}	-0.3 to 26.5	
Load Dump V_{SUP} (300 ms maximum)	V_{SUPLD}	40	
V_{CC}	V_{CC}	-0.3 to 7.0	
Maximum Voltage on Logic Input/Output Pins	–	-0.3 to $V_{CC} + 0.3$	V
Maximum Voltage on DBUS Pins	V_{DBUS}	-0.3 to $V_{SUP} + 0.3$	V
Maximum DBUS Pin Current	I_{DBUS}	400	mA
Maximum Logic Pin Current	I_{LOGIC}	20	mA
ESD Voltage 1			V
Human Body Model (HBM)	V_{ESD}	±2000	
Machine Model (MM)		±200	
Charge Device Model (CDM)		±750 for corner pins ±500 for others	
THERMAL RATINGS			
Storage Temperature	T_{STG}	-55 to 150	°C
Operating Ambient Temperature	T_A	-40 to 85	°C
Operating Junction Temperature	T_J	-40 to 150	°C
Thermal Shutdown	T_{SD}	155 to 190	°C
Resistance, Junction-to-Ambient	$R_{\theta JA}$	109	°C/W
Resistance, Junction-to-Board	$R_{\theta JB}$	50	°C/W
Peak Package Reflow Temperature During Reflow (2) , (3)	T_{PPRT}	Note 3	°C

Notes

- ESD1 testing is performed in accordance with the Human Body Model (HBM) ($C_{ZAP} = 100$ pF, $R_{ZAP} = 1500$ Ω); ESD2 testing is performed in accordance with the Machine Model (MM) ($C_{ZAP} = 200$ pF, $R_{ZAP} = 0$ Ω); and Charge Body Model (CBM).
- Pin soldering temperature limit is for 10 seconds maximum duration. Not designed for immersion soldering. Exceeding these limits may cause malfunction or permanent damage to the device.
- Freescale's Package Reflow capability meets Pb-free requirements for JEDEC standard J-STD-020C. For Peak Package Reflow Temperature and Moisture Sensitivity Levels (MSL), Go to www.freescale.com, search by part number [e.g. remove prefixes/suffixes and enter the core ID to view all orderable parts. (i.e. MC33xxxD enter 33xxx), and review parametrics.

STATIC ELECTRICAL CHARACTERISTICS

Table 3. Static Electrical Characteristics

Characteristics noted under conditions $4.75\text{ V} \leq V_{CC} \leq 5.25\text{ V}$, $9.0\text{ V} \leq V_{SUP} \leq 25\text{ V}$, $-40^\circ\text{C} \leq T_A \leq 85^\circ\text{C}$ unless otherwise noted. Voltages relative to GND unless otherwise noted. Typical values noted reflect the approximate parameter means at $T_A = 25^\circ\text{C}$ under nominal conditions unless otherwise noted.

Characteristic	Symbol	Min	Typ	Max	Unit
----------------	--------	-----	-----	-----	------

POWER INPUT REQUIREMENTS (V_{SUP} , V_{CC})

I_{VSUP} Supply Current ($ I_{BUS} \leq 10\ \mu\text{A}$) (Test Mode, CLK = 4.0 MHz) Idle, HiZ Signal High, Signal Low	I_{VSUP}	–	6.5 15	10 23	mA
I_{VCC} Supply Current (Test Mode, CLK = 4.0 MHz) Signal High, Signal Low	I_{VCC}	–	4.5	6.0	mA

MICROCONTROLLER INTERFACE ($\overline{\text{RST}}$, $\overline{\text{CS}}$, MOSI, SCLK, AND CLK)

I/O Logic Levels ($\overline{\text{RST}}$, $\overline{\text{CS}}$, MOSI, SCLK, and CLK) Input High Input Low Input Hysteresis ⁽⁴⁾	V_{IH} V_{IL} V_{HYST}	0.7 – –	– – 500	– 0.3 –	V_{CC} V_{CC} mV
Input Capacitance ⁽⁴⁾ $\overline{\text{RST}}$, $\overline{\text{CS}}$, MOSI, SCLK, and CLK	C_I	–	10	20	pF
Output Low Voltage MISO and $\overline{\text{INT}}$ Pins = 0.3 mA	V_{OL}	0	–	0.8	V
Output High Voltage MISO Pin = -0.3 mA	V_{OH}	$V_{CC} - 0.8$	–	V_{CC}	V
Output Leakage Current MISO Pin = 0 V MISO Pin = V_{CC}	I_{MISO}	-10 -10	– –	10 10	μA
$\overline{\text{INT}}$ Pullup Current $V_{OUT} = V_{CC} - 1.0\text{ V}$	$I_{\overline{\text{INT}}PU}$	-100	-75	-50	μA
SCLK, $\overline{\text{CS}}$ Pullup Current $V_{OUT} = V_{CC} - 1.0\text{ V}$	I_{PU}	-20	-10	-5.0	μA
$\overline{\text{RST}}$ Pulldown Current $V_{OUT} = 1.0\text{ V}$	$I_{\overline{\text{RST}}PD}$	4.0	7.0	10	μA
CLK, MOSI Pulldown Current $V_{OUT} = 1.0\text{ V}$	I_{PD}	5.0	10	20	μA

BUS TRANSMITTER (D0H, D0L, D1H, D1L)

Output Bus Idle Voltage (Differential) $I_{nH} = -200\text{ mA}$, $I_{nL} = 200\text{ mA}$ ⁽⁵⁾	$V_{DnD(IDLE)}$ ⁽⁶⁾	$V_{SUP} - 2.5$	–	–	V
Output Signal High Voltage (Differential) $-12.5\text{ mA} \leq I_{nH} \leq 1.0\text{ mA}$, $-1.0\text{ mA} \leq I_{nL} \leq 12.5\text{ mA}$ ⁽⁵⁾	$V_{DnD(HIGH)}$ ⁽⁶⁾	4.175	4.5	4.825	V

Notes

- 4 Not measured in production.
- 5 I_{nH} = bus current at DnH, I_{nL} = bus current at DnL.
- 6 $V_{DnD} = V_{DnH} - V_{DnL}$.

Table 3. Static Electrical Characteristics (continued)

Characteristics noted under conditions $4.75\text{ V} \leq V_{CC} \leq 5.25\text{ V}$, $9.0\text{ V} \leq V_{SUP} \leq 25\text{ V}$, $-40^\circ\text{C} \leq T_A \leq 85^\circ\text{C}$ unless otherwise noted. Voltages relative to GND unless otherwise noted. Typical values noted reflect the approximate parameter means at $T_A = 25^\circ\text{C}$ under nominal conditions unless otherwise noted.

Characteristic	Symbol	Min	Typ	Max	Unit
Output Signal Low Voltage (Differential) -12.5 mA \leq I _{NH} \leq 1.0 mA, -1.0 mA \leq I _{NL} \leq 12.5 mA ⁽⁵⁾	V _{DnD(LOW)} ⁽⁶⁾	1.175	1.5	1.825	V
V _{mid} , (DnH + DnL)/2 (Voltage Halfway Between Bus High Side and Bus Low Side)	V _{mid}	V _{SUP} /2 - 1.0	V _{SUP} /2	V _{SUP} /2 + 1.0	V
V _{CM} Peak-to-Peak, V _{mid} (Signal High) - V _{mid} (Signal Low) ⁽⁷⁾	V _{CMpp}	–	0.3	–	V
Output High-Side (DnH) Idle Driver Current Limit (DnL open) Source: DnH = 0 V Sink: DnH = V _{SUP}	I _{ICL(HIGH)}	-400 100	– –	-200 –	mA
Output Low-Side (DnL) Idle Driver Current Limit (DnH open) Source: DnL = 0 V Sink: DnL = V _{SUP}	I _{ICL(LOW)}	– 200	– –	-100 400	mA
Output High-Side (DnH) Signal Driver Overcurrent Shutdown Source: Signal High, Signal Low Sink: Signal High, Signal Low	I _{SCL(HIGH)}	-100 30	– –	-30 100	mA
Output Low-Side (DnL) Signal Driver Overcurrent Shutdown Source: Signal High, Signal Low Sink: Signal High, Signal Low	I _{SCL(LOW)}	-100 30	– –	-30 100	mA
Disabled High-Side (DnH) Bus Leakage (DnL open) DnH = 0 V DnH = V _{SUP}	I _{LK(HIGH)}	-1.0 -1.0	-0.18 0.25	1.0 1.0	mA
Disabled Low-Side (DnL) Bus Leakage (DnH open) DnL = 0 V DnL = V _{SUP}	I _{LK(LOW)}	-1.0 -1.0	-0.4 0.08	1.0 1.0	mA

BUS RECEIVER (D0H, D0L, D1H, D1L)

Comparator Trip Point	COMP(TRIP)	5.0	6.0	7.0	mA
-----------------------	------------	-----	-----	-----	----

Notes

7 Not measured in production.

DYNAMIC ELECTRICAL CHARACTERISTICS

Table 4. Dynamic Electrical Characteristics

Characteristics noted under conditions $4.75\text{ V} \leq V_{CC} \leq 5.25\text{ V}$, $9.0\text{ V} \leq V_{SUP} \leq 25\text{ V}$, $-40^\circ\text{C} \leq T_A \leq 85^\circ\text{C}$ unless otherwise noted. Voltages relative to GND unless otherwise noted. Typical values noted reflect the approximate parameter means at $T_A = 25^\circ\text{C}$ under nominal conditions unless otherwise noted.

Characteristic	Symbol	Min	Typ	Max	Unit
CLOCK					
CLK Periods					ns
Time High	tCLKHI	75	–	–	
Time Low	tCLKLO	75	–	–	
Period (System requirement) ⁽⁸⁾	tCLKPER	245	250	255	
CLK Transition (System requirement) ⁽⁸⁾					ns
Time for Low-to-High Transition of the CLK Input Signal	tCLKLH	–	–	50	
Time for High-to-Low Transition of the CLK Input Signal	tCLKHL	–	–	50	
Reset Low Time	tRSTLO	100	–	–	ns
SPI INTERFACE TIMING					
SPI Clock Cycle Time	tCYC	200	–	–	ns
SPI Clock High Time	tHI	80	–	–	ns
SPI Clock Low Time	tLO	80	–	–	ns
SPI $\overline{\text{CS}}$ Lead Time ⁽⁹⁾	tLEAD	100	–	–	ns
SPI $\overline{\text{CS}}$ Lag Time ⁽⁹⁾	tLAG	100	–	–	ns
SPI SCLK Time Between Bytes ⁽⁸⁾	tHI	80	–	–	ns
SPI $\overline{\text{CS}}$ Time Between Bursts ⁽⁸⁾	t $\overline{\text{CS}}$ HI	80	–	–	ns
Data Setup Time	tSU				ns
MOSI Valid Before SCLK Rising Edge ⁽⁹⁾		25	–	–	
Data Hold Time					ns
MOSI Valid After SCLK Rising Edge ⁽⁹⁾	tH	25	–	–	
MISO Valid After SCLK Falling Edge ⁽⁸⁾	tHO	0	–	–	
Data Valid Time	tV				ns
SCLK Falling Edge to MISO Valid, C = 100 pF		–	–	50	
Output Disable Time	tDIS				ns
$\overline{\text{CS}}$ Rise to MISO Hi-Z		–	–	100	
Rise Time (30% V_{CC} to 70% V_{CC}) ⁽⁸⁾	tR				ns
SCLK, MISO		–	–	25	
Fall Time (70% V_{CC} to 30% V_{CC}) ⁽⁸⁾	tF				ns
SCLK, MISO		–	–	25	

Notes

- 8 Not measured in production.
- 9 SPI signal timing from the production test equipment is programmed to ensure compliance.

Table 4. Dynamic Electrical Characteristics (continued)

Characteristics noted under conditions $4.75\text{ V} \leq V_{CC} \leq 5.25\text{ V}$, $9.0\text{ V} \leq V_{SUP} \leq 25\text{ V}$, $-40^\circ\text{C} \leq T_A \leq 85^\circ\text{C}$ unless otherwise noted. Voltages relative to GND unless otherwise noted. Typical values noted reflect the approximate parameter means at $T_A = 25^\circ\text{C}$ under nominal conditions unless otherwise noted.

Characteristic	Symbol	Min	Typ	Max	Unit
BUS TRANSMITTER					
Idle-to-Signal and Signal-to-Idle Slew Rate ($12 \leq V_{SUP} \leq 25\text{ V}$) ⁽¹⁰⁾	$t_{SLEW(IDLE)}$	2.0	4.5	8.0	V/ μs
Signal High-to-Low and Signal Low-to-High Slew Rate ^{(10), (13)} (See Data Valid DSIS to DnD Timing)	$t_{SLEW(SIGNAL)}$	3.0	4.5	8.0	V/ μs
Communication Data Rate Capability ⁽¹³⁾ (Ensured by Transmitter Data Valid and Receiver Delay Measurements)	D_{RATE}	–	–	150	kbps
Signal Bit Time ($1 / D_{RATE}$) ⁽¹³⁾	t_{BIT}	6.67	–	–	μs
\overline{INT} Turn ON Delay, DBUS Transaction End to Receive FIFO INT Low ^{(11), (15)}	t_{INTON}	–	–	$1/3 * t_{BIT} + 0.2$	μs
\overline{INT} Turn ON Delay ($C = 100\text{ pF}$) ⁽¹²⁾ \overline{CS} to \overline{INT} Low	t_{INTON}	–	–	0.2	μs
\overline{INT} Turn OFF Delay, $\overline{CS}/SCLK$ Rising Edge to \overline{INT} High	t_{INTOFF}	–	–	0.2	μs
DBUS Start Delay, $\overline{CS}/SCLK$ Rising Edge to DBUS ^{(11), (13), (15)} Spread Spectrum Mode Disabled	$t_{DBUSSTART1}$	$1/3 * t_{BIT}$	–	$2/3 * t_{BIT}$	μs
Spread Spectrum Mode Enabled	$t_{DBUSSTART2}$	$1/3 * t_{BIT}$	–	$4/3 * t_{BIT}$	μs
Data Valid ^{(10), (12)} DSIF (\overline{CS}) = $0.5 * V_{CC}$ to DnD Fall = 5.5 V	t_{DVLD1}	–	6.0	6.56	μs
DSIS (MOSI) = $0.5 * V_{CC}$ to DnD Fall = $0.2 * \Delta V_{DnD}$ ⁽¹⁴⁾	t_{DVLD2}	0.25	0.8	1.3	μs
DSIS (MOSI) = $0.5 * V_{CC}$ to DnD Rise = $0.8 * \Delta V_{DnD}$ ⁽¹⁴⁾	t_{DVLD3}	0.25	0.8	1.3	μs
DSIF (\overline{CS}) = $0.5 * V_{CC}$ to DnD Rise = 6.5 V	t_{DVLD4}	–	0.8	1.3	μs
Signal Driver Overcurrent Shutdown Delay	t_{OC}	2.0	–	20	μs
Signal Low Time for Logic Zero 33.3% Duty Cycle ⁽¹⁶⁾	t_{0LO}	$2/3 * t_{BIT} - 0.8$	$2/3 * t_{BIT} - 0.6$	$2/3 * t_{BIT} - 0.4$	μs
Signal Low Time for Logic One 66.7% Duty Cycle ⁽¹⁶⁾	t_{1LO}	$1/3 * t_{BIT} - 0.8$	$1/3 * t_{BIT} - 0.6$	$1/3 * t_{BIT} - 0.4$	μs

Notes

- 10 $C = 7.5\text{ nF}$ from DnH to DnL and 4.7 nF from DnH and DnL to GND, capacitor tolerance = $\pm 10\%$.
- 11 In the case where the SPI write to DnL (initiating a DBUS transaction start or causing an interrupt) is the last byte in the burst sequence, timing is from rising edge of \overline{CS} . Otherwise, timing is from the first SCLK rising edge of the next SPI burst byte.
- 12 Delays are measured in test mode to determine the delay for analog signal paths.
- 13 Not measured in production.
- 14 $\Delta V_{DnD} = V_{DnD(HIGH)} - V_{DnD(LOW)}$.
- 15 Internal digital delay only.
- 16 Guaranteed by design.

Table 4. Dynamic Electrical Characteristics (continued)

Characteristics noted under conditions $4.75\text{ V} \leq V_{CC} \leq 5.25\text{ V}$, $9.0\text{ V} \leq V_{SUP} \leq 25\text{ V}$, $-40^\circ\text{C} \leq T_A \leq 85^\circ\text{C}$ unless otherwise noted. Voltages relative to GND unless otherwise noted. Typical values noted reflect the approximate parameter means at $T_A = 25^\circ\text{C}$ under nominal conditions unless otherwise noted.

Characteristic	Symbol	Min	Typ	Max	Unit
BUS RECEIVER					
Receiver Delay Time ($I_{RSP} = 0\text{ mA} / 11\text{ mA step}$) ⁽¹⁷⁾					ns
$I_{RSP} = -6.0\text{ mA}$ to DSIR ($\overline{\text{INT}} = 0.5 * V_{CC}$)	t_{DRH}	250	–	750	
$I_{RSP} = -6.0\text{ mA}$ to DSIR ($\overline{\text{INT}} = 0.5 * V_{CC}$)	t_{DRL}	250	–	750	
Receiver Delay Time ($I_{RSP} = \text{COMP}(\text{TRIP}) - 2.0\text{ mA}/\text{COMP}(\text{TRIP}) + 2.0\text{ mA step}$) ⁽¹⁷⁾					ns
$I_{RSP} = \text{COMP}(\text{TRIP})$ to DSIR ($\overline{\text{INT}} = 0.5 * V_{CC}$)	t_{DRH}	500	–	1500	
$I_{RSP} = \text{COMP}(\text{TRIP})$ to DSIR ($\overline{\text{INT}} = 0.5 * V_{CC}$)	t_{DRL}	500	–	1500	
SPREAD SPECTRUM					
Central Frequency Range	f_{CEN}	132	–	148	kHz
Bit Period Deviation (+/-) ($f_{CENMin} \leq f_{CEN} \leq f_{CENMax}$)					ns
DEV[1:0] = 10	t_{DEV10}	400	–	600	
DEV[1:0] = 01 @ $f_{CEN}=138.5\text{KHz}$ (Typ)	t_{DEV01}	800	–	1100	
Frame Jitter (Max) ($f_{CENMin} \leq f_{CEN} \leq f_{CENMax}$) (PLL On)					μs
DEV On	$J_{FRDEVON}$	–	–	± 1.0	
DEV Off	$J_{FRDEVOFF}$	–	–	± 1.0	

Notes

- 17 Delays are measured in test mode to determine the delay for analog signal path.

TIMING DIAGRAMS

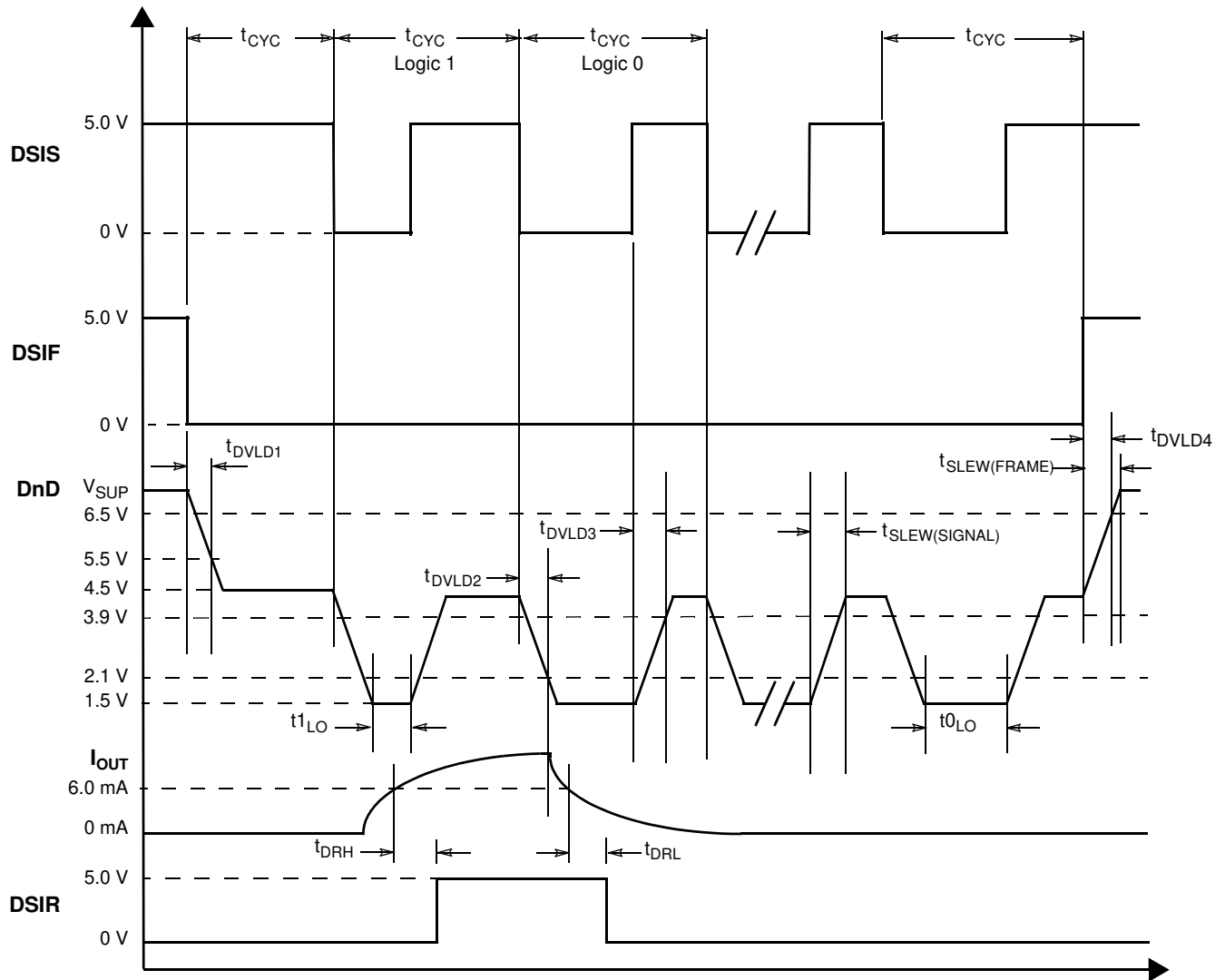


Figure 4. DBUS Timing Characteristics

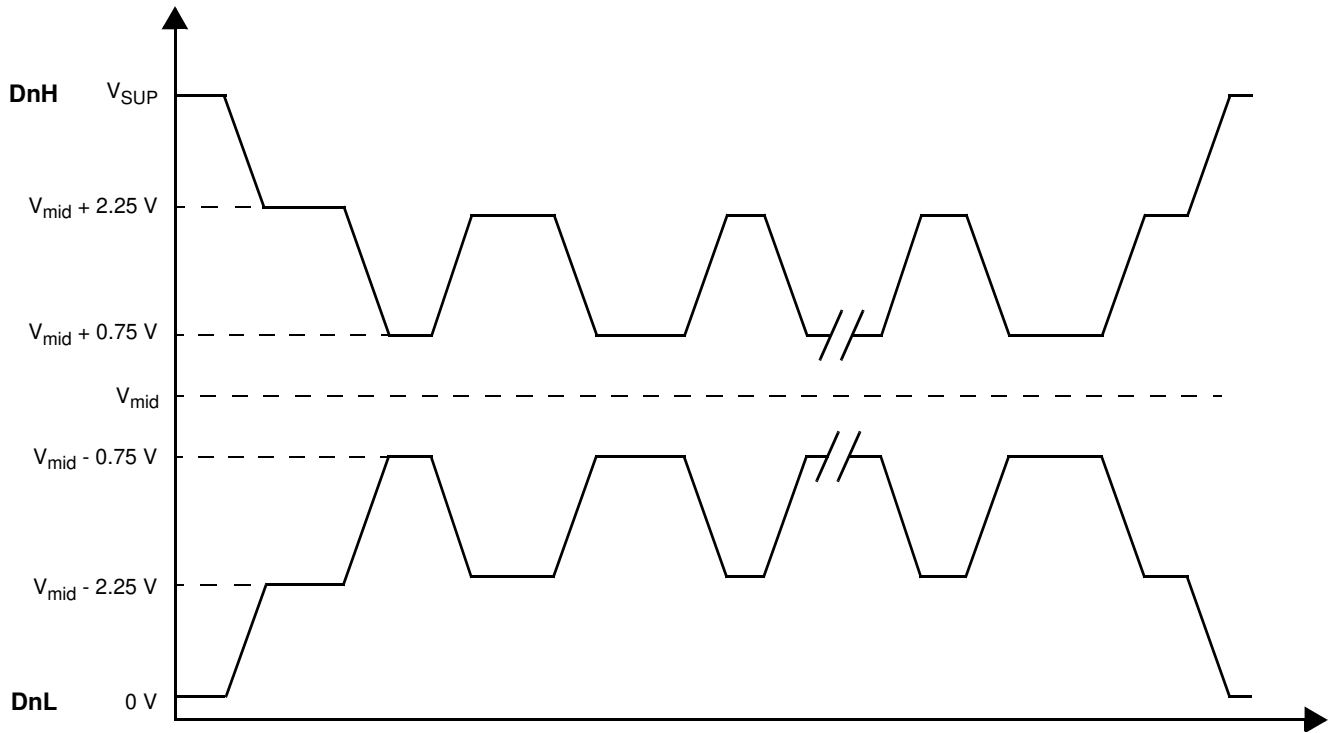


Figure 5. DBUS Normal Bus Waveforms

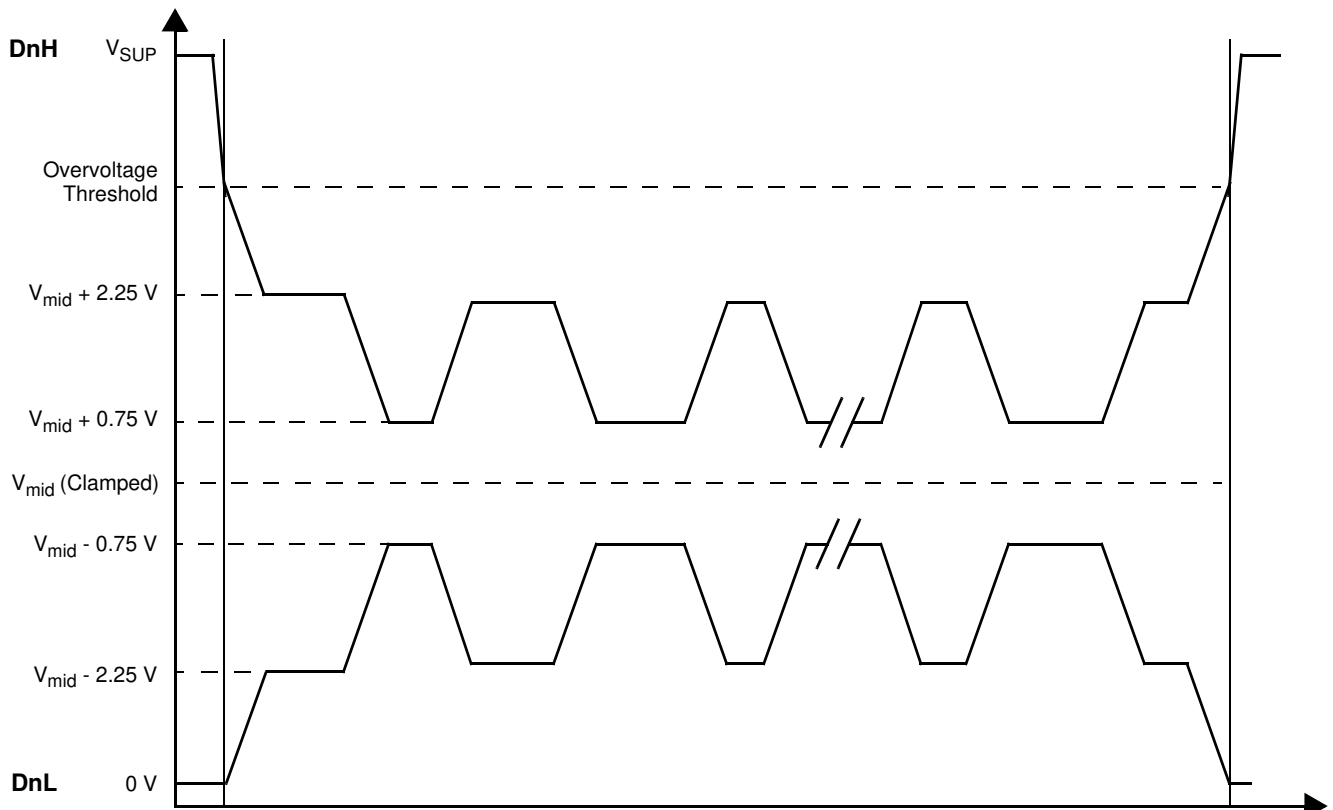
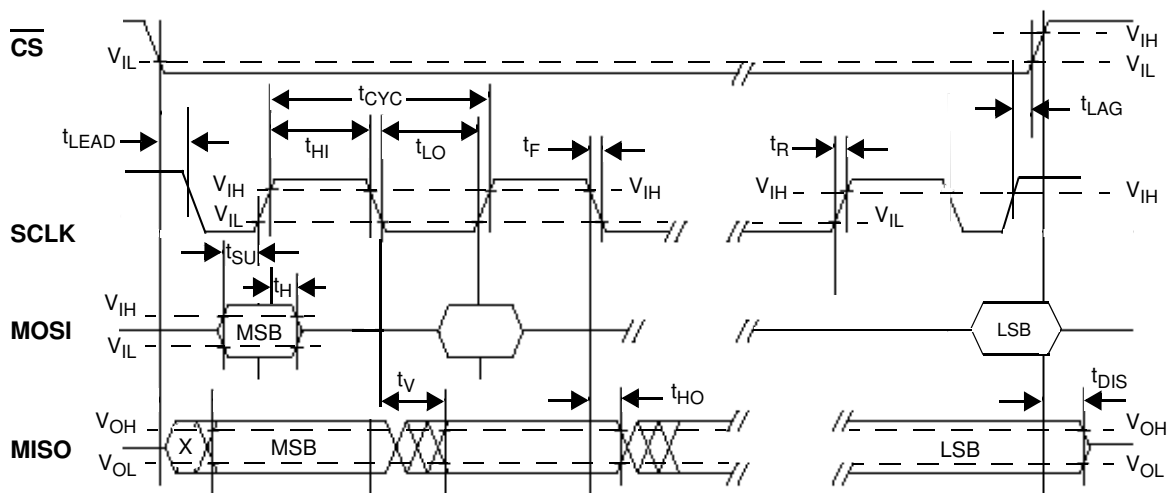


Figure 6. DBUS Overvoltage Bus Waveforms



X = Don't care

$V_{IH} = 70\% V_{CC}$, $V_{OH} = 70\% V_{CC}$

$V_{IL} = 30\% V_{CC}$, $V_{OL} = 30\% V_{CC}$

Figure 7. SPI Interface Timing

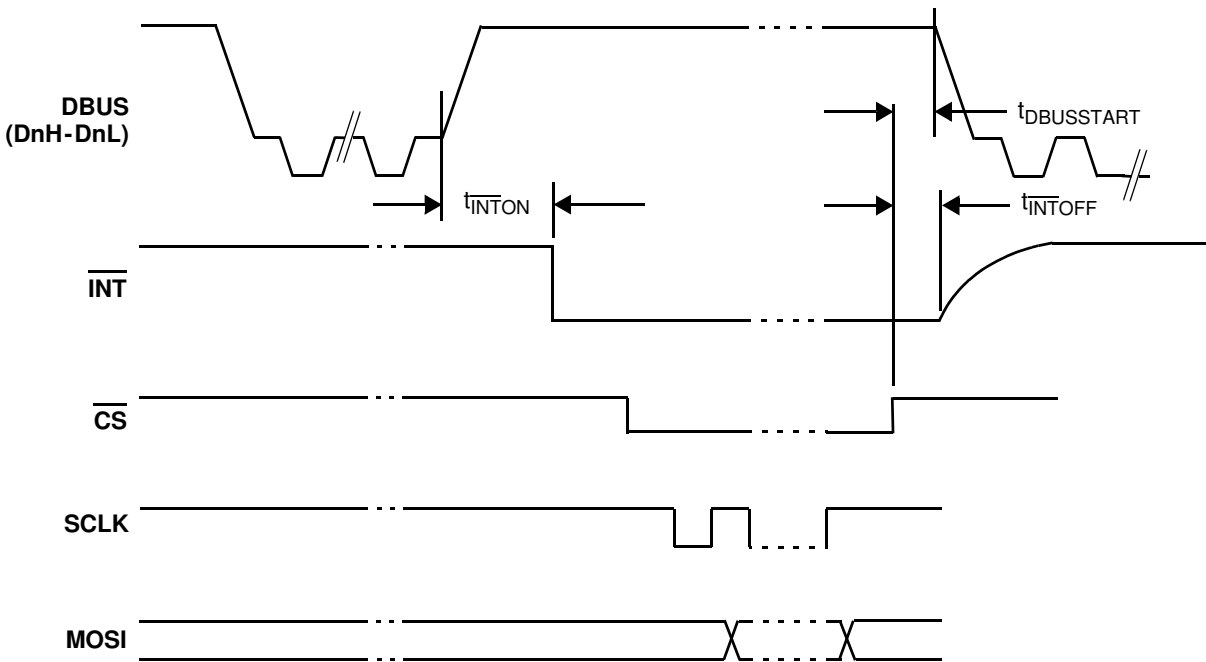


Figure 8. INT and Bus Start Timing

FUNCTIONAL DESCRIPTIONS

INTRODUCTION

The 33780 is intended to be used as a master device in a distributed system. It contains both protocol generators and physical interfaces to allow an MCU to communicate with devices on the bus using only a simple SPI interface. Two differential busses are provided.

Using a loop-back wire allows operation of the bus in the presence of an open circuit. This is immediate and no interruption is caused by the open circuit. The differential outputs have reduced electromagnetic radiation and susceptibility.

The equivalent bus capacitance consists of capacitors connected between the two bus wires and capacitors between the bus wires and ground. Because the voltage change on either of the bus wires to ground is only 1/2 the

amount of change between the two bus wires, the capacitance to ground only conducts half as much current as it would if connected directly across the bus. The equivalent bus capacitance of a capacitor to ground from the bus wires is one half of the actual amount of the capacitor. The amount of capacitance from either bus wire to ground should be kept the same in order to achieve the lowest radiated EMI energy. The 4.7 nF capacitors required between the bus wires and ground result in an equivalent of 2.35 nF of capacitance across the bus as seen by either bus wire.

[Table 5](#) shows the voltages used for operation. Low side (LS) is the bus wire that is the most negative and high side (HS) is the bus wire that is the most positive. [Figure 5](#) shows the bus waveforms in normal operation.

Table 5. High-Side and Low-Side Typical Voltages (Voltage Relative to Ground)

Low Side			High Side		
IDLE	HIGH	LOW	IDLE	HIGH	LOW
0	Vmid-2.25 ⁽¹⁸⁾	Vmid-0.75 ⁽¹⁸⁾	V _{SUP}	Vmid+2.25 ⁽¹⁸⁾	Vmid+0.75 ⁽¹⁸⁾

Notes

18 Vmid = V_{SUP}/2.

FUNCTIONAL PIN DESCRIPTIONS

RESET ($\overline{\text{RST}}$)

When pulled low, this will reset all internal registers to a known state as indicated in the section entitled [Register and Bit Descriptions](#).

CHIP SELECT ($\overline{\text{CS}}$)

This input is used to select the SPI port when pulled to ground. When high, the SPI signals are ignored. The SPI transaction is signaled as completed when this signal returns high.

INTERRUPT ($\overline{\text{INT}}$)

This output will be asserted low when an enabled interrupt condition occurs. It contains an internal current pull-up source so that it will remain high when not active. The output is open-drain so that it can be ORed together with other open-drain outputs so that this IC or any of the others can assert an interrupt.

MASTER OUT/SLAVE IN (MOSI)

This is the SPI data input to the device. This data is sampled on the positive (rising) edge of SCLK.

SERIAL CLOCK (SCLK)

This is the clock signal from the SPI master device. It controls the clocking of data to the device and data read from the device.

MASTER IN/SLAVE OUT (MISO)

This is the SPI data from the device to the SPI master (the MCU). Data changes on the negative (falling) transition of the SCLK.

CLOCK (CLK)

This is the main clock source for the internal logic. It must be 4.0 MHz.

GROUND (GND)

Ground source for both logic and DBUS return.

POWER SOURCE (VCC)

Logic power source. Nominal value is +5.0 V. This should be bypassed with a small capacitor to ground (0.01-0.1 μF)

LOW-SIDE BUS (DnL)

There are two independent LOW-SIDE outputs, D0L and D1L. They comprise the low-side differential output signal of

the DBUS physical layer as shown in [Figure 5](#). They also provide power to the slave modules during the DBUS idle time. The output of DnL should have a bypass capacitor of 4.7 nF to ground.

HIGH-SIDE BUS (DnH)

There are two independent HIGH-SIDE outputs, D0H and D1H. They comprise the high-side differential output signal of the DBUS physical layer. They also provide power to the slave modules during the DBUS idle time. See [Figure 5](#). The

output of DnL should have a bypass capacitor of 4.7 nF to ground.

POSITIVE SUPPLY FOR BUS OUTPUT (VSUP)

This 9.0 V to 25 V power supply is used to provide power to the slave devices attached to the DBUS. During the bus idle time, the storage capacitors in the slave modules are charged up to maintain a regulated supply to the slave device.

FUNCTIONAL INTERNAL BLOCK DESCRIPTION

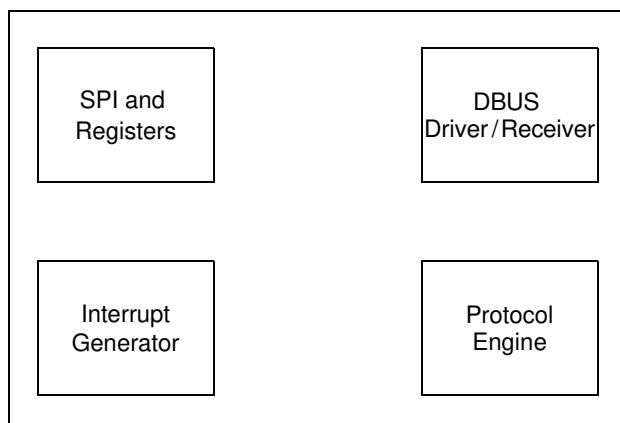


Figure 9. Block Illustration

The 33780 is controlled by an MCU through an SPI interface. It handles the digital and physical layer portions of a DBUS master node. Two separate DBUS channels are included, each capable of interfacing to up to 15 DBUS slave devices (nodes). The physical layer uses a two-wire bus with analog wave-shaped voltage and current signals. Refer to [Figure 1](#).

Major subsystems include the following:

- SPI interface to an MCU
- A register pointer block
- Two channels of DBUS protocol state logic
- CRC block for each channel
- Control and status registers
- 4-level FIFOs on each transmit and receive buffer

SPI AND REGISTERS

This block contains the SPI interface logic and the control and response registers that are written to and read from the SPI interface.

The IC is an SPI slave-type device, so MOSI (Master-Out-Slave-In) is an input and MISO (Master-In-Slave-Out) is an output. \overline{CS} and SCLK are also inputs.

The SPI port will handle byte and multi-byte transfers. It addresses 22 registers. The 33780 combines the functions of both the 68HC55 (DSID) and the 33790 (DSIP). The 33780 uses the eight control registers defined in the DSID, and the remaining registers are needed for the additional modes of operation in the 33780. The organization of the registers is described in the section entitled [Register and Bit Descriptions](#).

INTERRUPT GENERATOR

This circuit accepts unmasked interrupt inputs for data flow. It drives an open-drain output that allows the output to be OR connected with other open-drain outputs so that this IC or any of the others can assert an interrupt. An unmasked interrupt will cause the \overline{INT} to pull down the output. Interrupts can be generated by two circumstances: (1) a Transmit FIFO register becoming empty, or (2) the Receive FIFO becoming not empty. Both of these events occur at the end of a DBUS transaction. Either of these two events will generate an interrupt when enabled by setting bits in the DnCTRL registers.

Similarly, the interrupt signal can be cleared in two ways: (1) the Transmit FIFO becomes not empty, or (2) the Receive FIFO becomes empty. Both of these events are checked at the end of an SPI word (either with \overline{CS} rising or with the rising edge of SCLK of a new data byte in an SPI burst).

PROTOCOL ENGINE

This block converts the data to be transmitted from the registers into the DBUS sequences, and converts DBUS response sequences to data in the registers. It generates the proper DBUS timing.

The DBUS transmit protocol uses a *return to 1* type data with a duty cycle determined by the logic state. The protocol requires Cyclical Redundancy Check (CRC) generation and validation.

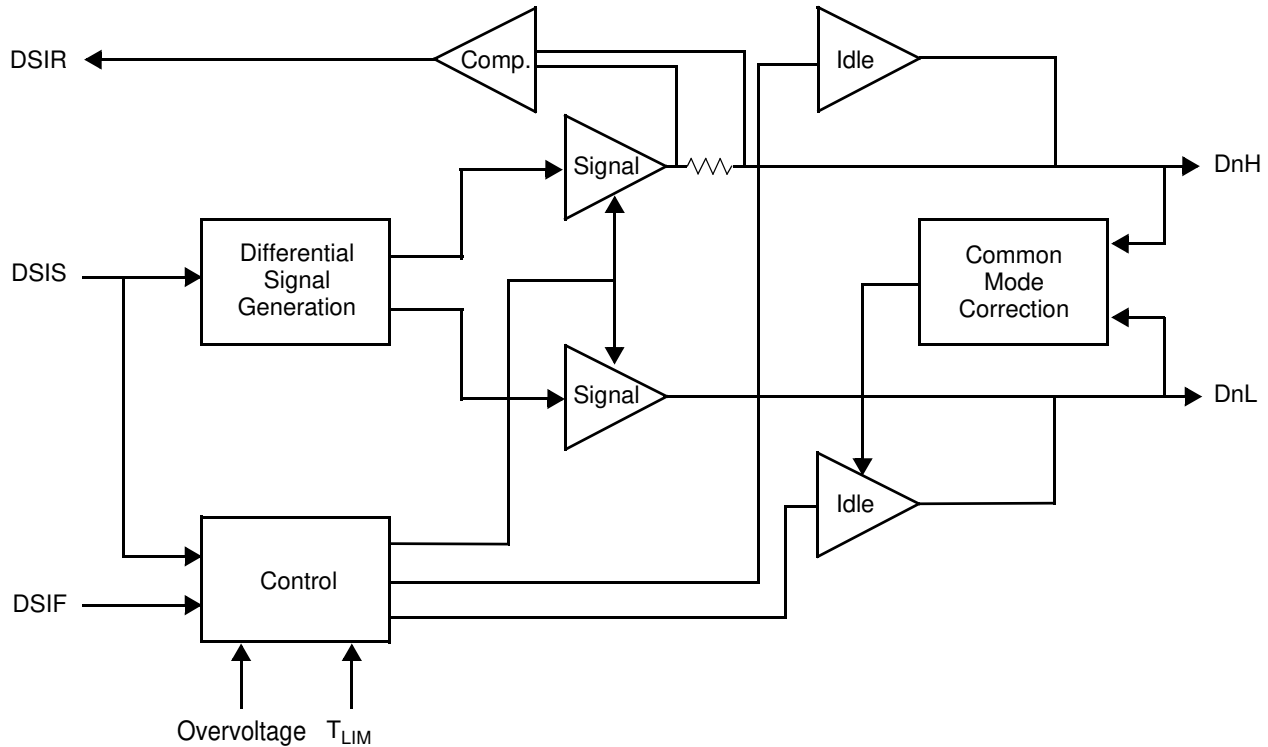


Figure 10. Driver/Receiver Block Diagram

DBUS DRIVER / RECEIVER (PHYSICAL LAYER)

This block translates the transmit data to the voltage and current needed to drive the DBUS. It also detects the response current from the slave devices and translates that current into digital levels. These circuits can drive their outputs to the levels listed in [Table 5](#).

The internal signal DSIF controls the Idle to Signalling state change, and internal signal DSIS controls the signal level, high or low. DSIR is the slave device response signal to the logic. This is shown in [Table 6](#).

Table 6. Internal Signal Truth Table

DSIF	DSIS	T_{LIM}	DSIR	DnD
0	0	0	Return Data	Signal Low
0	1	0	Return Data	Signal High
1	0	0	0	High Impedance
1	1	0	0	Idle
X	X	1	0	High Impedance

The DBUS driver block diagram is shown in [Figure 10](#). The circuit uses independent drivers for the Idle and Signal states. This allows each driver to be optimized for its function. The *Idle* driver is active in Idle and during the transitions from Idle to Signal high and Signal high to Idle. The *Signal* driver is only active during actual signaling. Both drivers are disabled in HiZ.

The Idle driver is required to supply a high current to recharge the Slave device storage capacitors. It is also required to drive the DBUS load capacitances and control the slew rate over a wide supply voltage range. The DnH and DnL Idle drivers are each optimized for their specific drive requirements.

The Signal driver is optimized for driving the DBUS load, and has the requirement of good slew rate control and stability over a wide range of load conditions. The DnH and DnL outputs use identical Signal driver circuits.

To ensure stability of the Signal driver, capacitors must be connected between each output and ground. These are the DBUS common mode capacitors. In addition, a bypass capacitor is required at V_{SUP} . These capacitors must be located close to the IC pins and provide a low impedance path to ground.

The DSIF signal controls the state of the drivers, enabling the Idle drivers or Signal drivers as is appropriate. A comparator in the *Control* block compares the DnL output voltage with the internal Signal high voltage to determine the transition from Idle driver to Signal driver. The overvoltage signal modifies the driver characteristics. This is described in more detail in the [Load Dump Operation](#) section.

The overtemperature signal is also applied to this block.

The *Differential Signal Generation* block converts the DSIS signal to the DBUS differential signal voltage levels. This differential signal is buffered and slew rate controlled by the Signal drivers. This block is active in all driver modes.

A special requirement of the differential bus is to maintain a low common mode voltage. This is especially important during the Idle to Signal transition in order to produce a smooth changeover to the Signal driver. This is accomplished by monitoring the common mode voltage and modifying the Idle driver slew rates. This is the function of the *Common Mode Correction* block. An additional feature to make a smooth changeover and minimize undershoot is to reduce the slew rate as the changeover point is approached. This block is not illustrated in [Figure 10](#).

A sense resistor between the Signal driver and the DnH output detects the Slave device response current. A comparator (*Comp.*) generates the signal DSIR that is supplied to the logic.

The comparator consists of a sense amplifier with offset (V_{OS}), a filter capacitor and logic gate with buffers to produce the logic signal (DSIR). The sense amplifier is a 'gm' stage that amplifies the voltage across the sense resistor (R_S) to produce an output current that charges and discharges a filter capacitor. The voltage across the filter capacitor is compared

with the threshold voltage of the logic gate to produce the output signal. The voltage across the filter capacitor is clamped between VCC and ground. See [Figure 11](#).

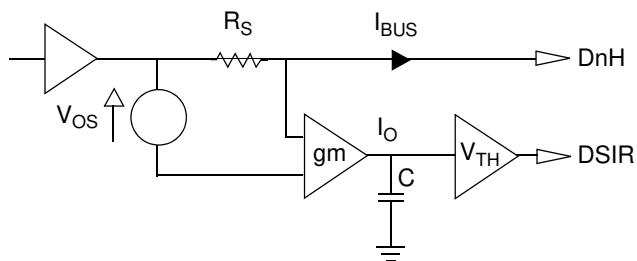


Figure 11. Receive Filter

Definitions

- C = value of filter capacitor = 2.0 pF
- V_{TH} = threshold of logic gate = $V_{CC}/2 = 2.5$ V
- A = current gain from sense resistor to filter capacitor = $I_O/I_{BUS} = 3.0 \mu A/mA$ (the amplifier saturates with an output current of $\pm 40 \mu A$)
- $I_{BUS}[mA]$ = bus response current.
- $I_{TH}[mA]$ = response current threshold = $V_{OS}/R_S = 6$

The filter delay time is given by:

$$t[\mu s] = (C * V_{TH})/A(I_{BUS} - I_{TH}) = 1.7/(I_{BUS} - I_{TH})$$

The filter characteristic can also be expressed as the product of the overdrive current ($I_{BUS} - I_{TH}$) and the duration of the interference pulse, which must be less than $1.7 \mu s * mA$ for the interference to be filtered.

SPREAD SPECTRUM

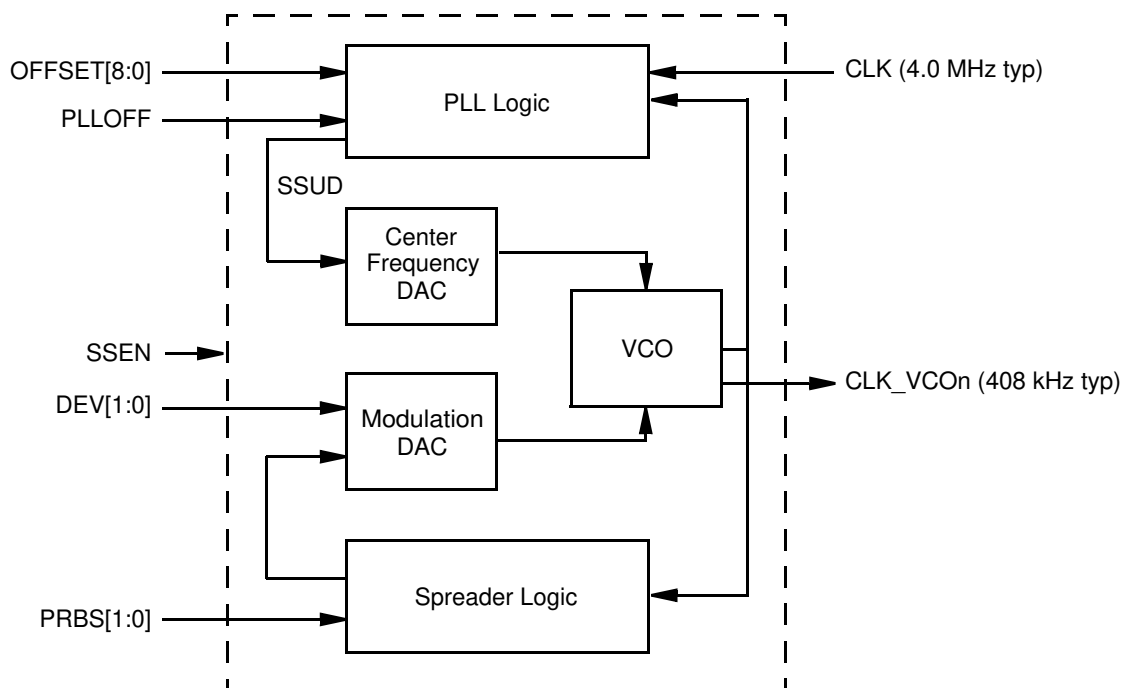


Figure 12. Spread Spectrum Block Diagram

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. This results in undesired signals appearing at multiples of the frequency that can be strong enough to interfere with a desired signal.

A significant decrease of radiated EMI can be achieved by randomly changing the duration of each bit. This can significantly reduce the 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. For instance, if the bit rate is 136 kbps, there will be a harmonic at 680 kHz. If it is changed in frequency so that only 1/10 of the bits are at the 136 kbps rate, the signal energy at 680 kHz will be reduced by 90%.

A circuit to do this is included in this IC and can perform the *spreading* of the signal independently for each channel. This is done in the Spread Spectrum (SS) Block Diagram shown in [Figure 12](#).

Spreading can be enabled by setting the SSENn bits in the DnSSCTRL registers. There are 64 possible bit durations that are equally spaced between the shortest and longest bit times. Because they are evenly spaced by a time difference and not by a frequency difference (the reciprocal of time), all frequency domain parameters of the SS block are expressed in units of time.

VCO

The output of the voltage-controlled oscillator (VCO) is used as the bit rate clock. Three cycles of this clock are used to create each bit of data on the DBUS.

There are two voltages that control the period (1/frequency) of the signal coming from the VCO. The voltage coming from the Center Frequency DAC (Digital-to-Analog Converter) in [Figure 12](#) is used to keep the average period constant. The voltage coming from the Spreader DAC changes the period in random steps to spread the signal. The Phase Locked Loop (PLL)-derived changes are much slower to update the period than the ones derived from the Spreader Logic. This prevents the two “loops” from interacting with each other.

PLL

The PLL *loop* compensates for temperature drift and the variations in processing of the IC that would otherwise change the average data rate (center frequency). It does this by comparing a time reference derived from the clock signal (4.0 MHz) to the period of the VCO output. If the ratio is not correct, it will change the frequency of the VCO by changing the digital value it sends to the Center Frequency DAC.

The PLL has fast and slow update rates for making these changes. It enters a fast update mode automatically anytime the OFFSET register is written to using the SPI, or following a reset. This fast acquisition mode consists of 64 VCO update cycles (1.4 ms per update cycle) that last about 90 ms. This is done to quickly adjust the center frequency after changes have been made. After the fast acquisition, the PLL switches

automatically to a slow acquisition mode (180.224 ms per update cycle, based on 4.0 MHz clock).

SPREADER LOGIC

The Spreader Logic contains a pseudo-random binary sequence (PRBS) generator and time compensation circuitry. The PRBS can generate maximal length sequences of 6, 7, 11, and 15 bits. Maximal length means there is no repeat of the sequence until 2^n counts have been reached, where n is the selected length.

A special feature of the Spreader Logic is that the bit periods are chosen in a way to keep the length of the frame constant, provided that the total number of bits is even. This is useful if the time between samples made by the slaves must be kept relatively constant. Without this feature, the time from sample-to-sample would vary randomly.

The DEV1 and DEV0 bits in the DnSSCTRL register control whether the deviation is enabled or disabled.

The Spreader Logic is synchronized to only change the value of the digital word to the Spreader DAC at the beginning of a DBUS bit. When spreading is enabled, these changes will occur once per DBUS bit-time.

FUNCTIONAL DEVICE OPERATION

LOGIC COMMANDS AND REGISTERS

SPI COMMUNICATIONS

All SPI transactions start with a command byte and can be followed by 1 or more bytes of data. The start of an SPI transaction is signaled by \overline{CS} being asserted low. The first bit sent (bit 7) of the first byte signals a read or write (write = 1) of data. The last five bits (bits 4–0) of the command set a

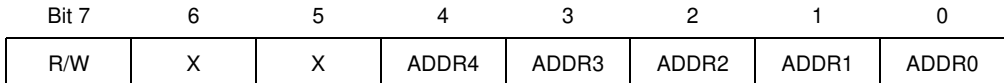


Figure 13. SPI Communications, First Byte of Burst Transfer

The receive FIFO is popped only when the SPI reads or writes the low data register (DnL). The Control and Status registers can be read without affecting the receive FIFO. The transmit FIFO is popped at the end of the DBUS transaction.

Figure 14 shows an example of a write operation. This example assumes the last SPI transaction read or wrote the data from register 00011 and is now pointing at 00100 (D01STAT). During the first byte of the SPI transaction, the first MOSI bit is 1 (write) and the last five are 00000. During this command byte, MISO returns the data from register 00100 (D01STAT). During the next SPI transactions, MOSI

pointer to the desired register. Bits 5 and 6 are unused. See Figure 13. As long as \overline{CS} is asserted low, each additional byte sent over the SPI will be a read/write of data to the sequential next register. After address 10101 is written to, the next write will wrap around to address 00000.

updates the data in register 00000 with new data while reading back the old data via MISO.

Although it looks like the read and write for an address are occurring at the same time, the changes caused earlier during the same burst would not be reflected by the data returned, because the D01STAT is latched at \overline{CS} going low.

When a short word is selected for Bus 0 (MS0 in D0CTRL is set), the D0H register is skipped in the sequence. The same is true for the D1H register when MS1 is set and SWLEN1 = 1000.

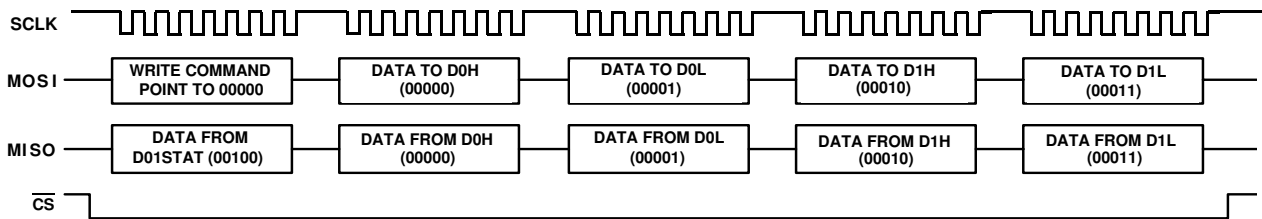


Figure 14. SPI Burst Transfer Example

DBUS COMMUNICATIONS

The DBUS messages contain data from the DnH and DnL registers. A CRC pattern is automatically appended to each

message. The data and CRC lengths are programmed by the DnLENGTH register. Figure 15 shows the structure of the DBUS message.

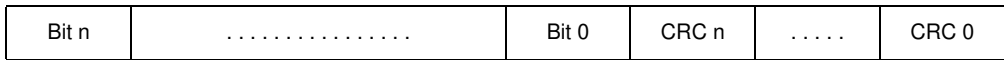


Figure 15. DBUS Communications Message

DBUS Driver/Receiver communications involve a frame (DSIF), a data signal (DSIS), and a data return (DSIR) signal. These are signals internal to the IC associated with the protocol engine.

A message starts with a falling edge on the DSIF signal, which marks the start of a frame. There is a one bit-time delay before the MSB of data appears on the DSIS pin. Data bits start with a falling edge on DSIS. The low time is 1/3 of the bit time for a 1, and 2/3 of a bit time for a 0. Data is transmitted on DSIS and received on DSIR pins simultaneously. Receive data is the captured level on the DSIR pin at the end of each

bit time. At the end of the bit time for the last CRC bit, the DSIF pin returns to a logic high (Idle level). A minimum delay is imposed between successive frames as determined by the DnCTRL register.

Users initiate a message by writing (via the SPI interface from the MCU) to the low byte of the data register (DnL). When 9- to 16-bit messages are to be sent, the user writes to the DnH register first and then the DnL register before the combined 9 to 16-bit data value DnH:DnL is sent on the DBUS. The user should first check the TFNFn status flag to be sure the transmit FIFO is not full before writing a new data

value to DnH and/or DnL. When the minimum inter-frame delay has been satisfied, the DSIF pin will go low, indicating the start of a new transfer frame.

DBUS Driver/Receiver communications involve a frame (DSIF), a data signal (DSIS), and a data return (DSIR) signal. These are signals internal to the IC associated with the prData is shifted out of DSIS (MSB first) and shifted into DSIR at the same time. As a message is received, it is stored bit-by-bit into the next available receive FIFO location. For each data value in the receive FIFO, there is a one-bit status flag to indicate whether or not there was a CRC error while receiving the data. At the end of a DBUS transfer (and after the CRC error status is stable), the RFNE flag is set (if it was not already) to indicate there is data in the receive FIFO to be read.

DATA RATE

In non-spread spectrum mode, the data rate is determined by the system clock (CLK) and the programmable clock divider. (The Clock Divider ratio n is defined in [Table 10](#).)

$$\text{Data Rate} = f_{\text{CLK}} / (27 * n)$$

In spread spectrum mode, the data rate is determined by the system clock (CLK) and the DnOFFSETL/H register programming. Note the programmable clock divider does not control the data rate in Spread Spectrum mode. Refer to [Register and Bit Descriptions](#) section for details.

The following table gives the correspondence between the offset and the data rate for $f_{\text{CLK}} = 4.0 \text{ MHz}$.

Table 7. Data Rate versus OFFSET (Spread Spectrum)

OFFSET		Data Rate
HEX	DEC	kHz
00	0	121.2
3F	63	136.1
7A	122	150.1
9F	159	158.9

For other clock frequencies, the data rate can be computed using the following formula:

$$\text{Data Rate} = (f_{\text{CLK}} / 33) * (1 + \text{OFFSET} / 512)$$

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

CRC GENERATION / CHECKING

Whenever a message is sent on the DBUS, a 0- to 8-bit CRC value is computed and serially sent as the next n bits after the LSB of the data. The CRC length, polynomial, and initial seed are determined by the CRCLen[3:0], CRCPOLY[7:0], and CRCSEED[7:0] control register fields. The message, including the CRC bits, is passed 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 include a 0- to 8-bit CRC value, which was computed in the peripheral device that is responding. As the message is received, a separate 0- to 8-bit CRC value is computed and is compared with the CRC value in the received message. If these values do not agree, the message is considered invalid and the ERn status bits in the D01STAT register are set as the receive data is transferred into the receive data buffer.

When no remote peripheral responds to a message, the data pattern received will be all zeros with a CRC value of 0, which may be detected as a CRC error depending on the values of CRCLen[3:0], CRCPOLY[7:0], and CRCSEED[7:0]. On the other hand, if a remote peripheral is attached and responds with all zeros with a CRC value of 1010, this may be detected as a non-error condition.

CRC COMPUTATION

The CRC algorithm uses a programmable initialization value, or seed, of CRCSEED[7:0] and a programmable polynomial of CRCPOLY[7:0]. [Figure 16](#) is a VHDL description of the CRC algorithm for the DBUS standard 4-bit CRC with its initial value of 1010. A seed value is chosen so that a zero data value will generate a CRC value of 1010. A block diagram of the default CRC calculation is shown in [Figure 17](#).

```

-----
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;
    
```

Figure 16. CRC Algorithm

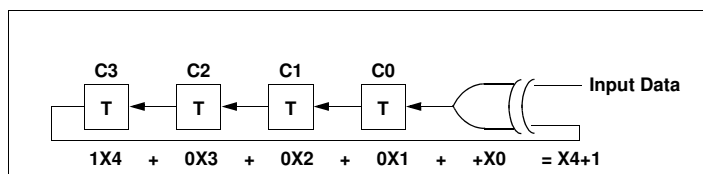


Figure 17. Default CRC Block Diagram

MESSAGE SIZE SPECIAL CASES

The response to any 8- to 15-bit message is expected to be another 8- to 15-bit message and the response to any 16-bit message is expected to be another 16-bit message. This gives rise to some special cases when there is a transition from one message size to a different message size. Some messages must be long words (16 bits of data), others can be short words (8 to 15 bits of data).

The following are examples where the word is a standard DSI formatted short word (8 bits of data and 4 bits of CRC).

Example 1: If the previous message was a short word and the current message is a long word, the response message (which is also a short word) finishes before the current message frame and the CRC bits look like data bits in the long word format. Since the CRC validation of this short word message response is not reliable, this short word response should not be used.

Example 2: If the previous message was a long word and the current message is a short word, the response message (which is also a long word) cannot finish before the current message frame. Bits three to zero of the data and the CRC bits are lost. Data bits seven to four of the 16-bit response message look like the CRC bits of an 8-bit response and almost certainly would not be correct. Because the response

is incomplete and the CRC check is probably not valid, this response is not useful.

The long word to short word message size transition normally only occurs after setting up the DBUS 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). After the last peripheral has been assigned an address, any subsequent address assignments will receive no response. When the master MCU fails to receive a response, it knows it has passed the last peripheral. At this point, short word messages may be sent. The first such message will have no meaningful response associated with it.

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

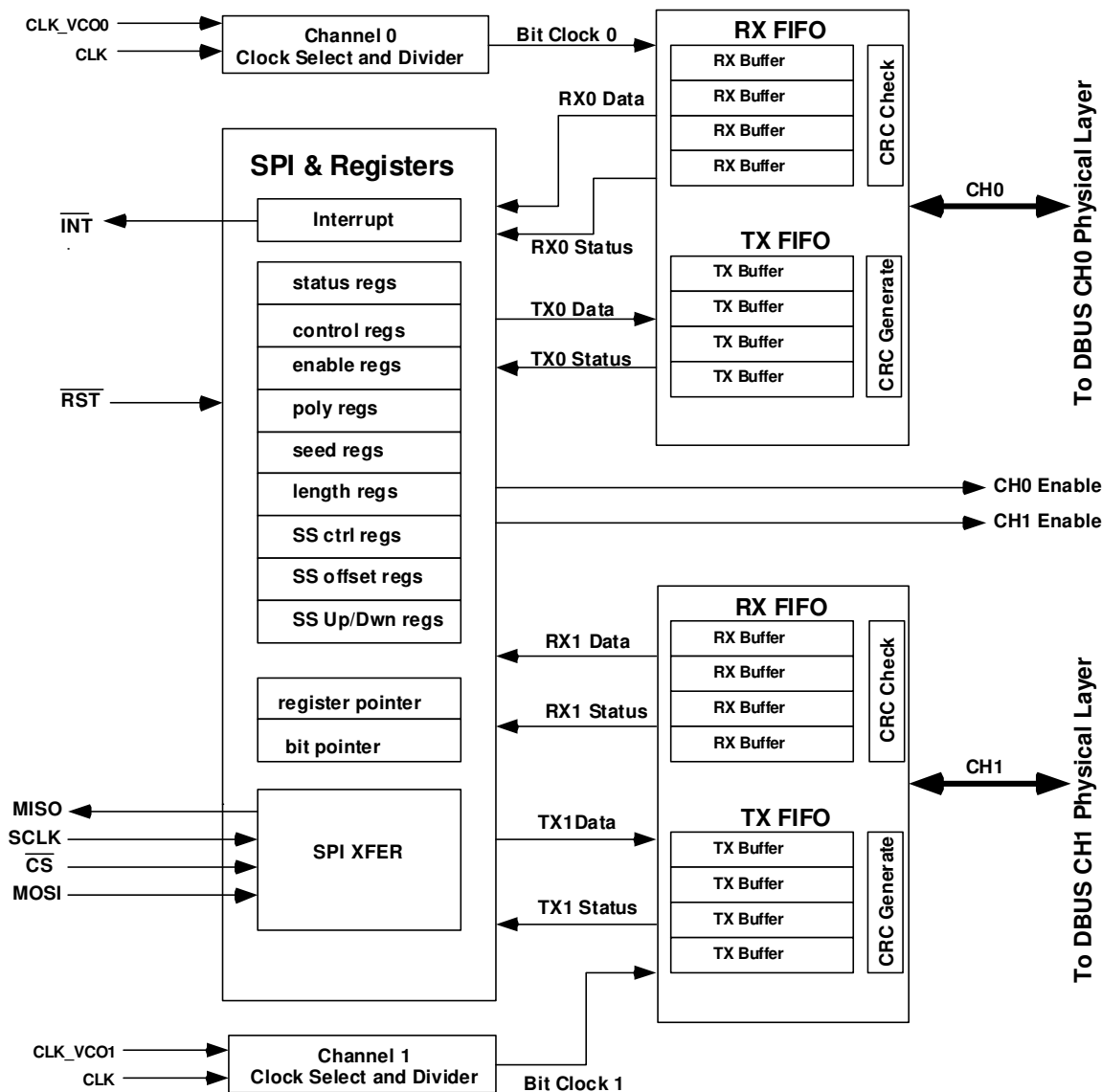


Figure 18. Logic Block Diagram

LOGIC BLOCK DIAGRAM DESCRIPTION

Figure 18, Logic Block Diagram, shows a block diagram of the major logic blocks in the IC.

SPI

The SPI is a standard serial peripheral interface. This interface provides two-way communications between the IC and an MCU. The MCU can write to registers that control the operation of the IC and read back the conditions in the IC using the SPI. It can also write data to be sent out on the DBUS and read data that was returned on the DBUS. The register pointer and bit pointer are used to control which registers and bits are being written to and read from using the SPI. Its operation is described in detail in the section entitled [SPI Communications on page 19](#).

REGISTERS

The register set consists of control, status, transmit, and receive types. They are written and read using the SPI interface and are affected by events in the IC. Detailed descriptions of their operation and use can be found throughout later sections of this data sheet.

INTERRUPT

The Interrupt block controls the $\overline{\text{INT}}$ output pin. The main purpose of the Interrupt is to quickly inform the MCU when data has been received via the DBUS or when the DBUS transmit buffer is empty. The $\overline{\text{INT}}$ output can only drive the level on the pin low. The internal pull-up current or an external resistor to V_{CC} is used to pull this pin high. This is done so that other ICs can be connected to the interrupt pin on the

MCU. If the DBUS IC or any of the other ICs want to assert an interrupt to the MCU, they can do so by pulling the pin low. This is similar to a logical OR of the outputs because this IC or any of the others can assert the interrupt to the MCU. The operation of the Interrupt is described in detail in the section titled [Interrupt Generator on page 14](#).

$\overline{\text{RST}}$

Asserting this pin low will cause the part to reset, forcing registers to a known state. The description for these registers shows the bit values that will occur due to a reset. All bus activity will be halted and not allowed to restart, and no SPI activity will be recognized until the $\overline{\text{RST}}$ goes to a logic high level.

CLOCK SELECT AND DIVIDER

There is an independent Clock Select and Divider for each channel. These circuits are controlled by register writes to the SPI and can select whether the Spread Spectrum Clock (CLK_VCO_n) is used for the bit clock or the unspread clock is used. They also contain dividers that can be selected to reduce the bit rate by integer ratios in the unspread mode only.

RXFIFO

The RXFIFO is an automatic register set that allows up to four responses to be stored without being transferred to the MCU via the SPI. This is done so that data will not be lost even if the MCU takes time to read the response data. When the MCU reads a response from one of the DBUS registers, the earliest response to be received is the one read. In other words, the first in response will be the first out (FIFO). When the RXFIFO becomes not empty and interrupts are enabled, the MCU receives an interrupt via $\overline{\text{INT}}$.

TXFIFO

The TXFIFO is an automatic register set that allows up to four transmit data packets to be stored for future transmission on the DBUS. This is done to prevent the overwrite of transmit data if the transmission of the previous data has not been completed. The oldest data in the registers is the first to be sent when the DBUS is ready to send. In other words, the first data put into the registers to be sent will be the first out when the DBUS is available (FIFO). When the TXFIFO becomes empty and interrupts are enabled, the MCU receives an interrupt via $\overline{\text{INT}}$.

CH0/CH1 ENABLE

The output of these signals control whether the DBUS can drive power and signalling onto the bus. These are directly controlled by bits written to the control registers.

CH0/CH1 OUTPUTS

These signals control the physical layer drivers and receive data from the physical layer receivers. The physical

layer will convert the 0 V to 5.0 V low power logic signals to the higher voltage (up to 26.5 V) and drive (150 mA nominal) levels necessary for the DBUS to be used. It also converts the low current (0 mA to 11 mA typical) loading of the response signal from the slave to logic voltage levels to allow the response from the slaves to be received. These internal signals are named DSIF, DSIS, and DSIR.

CRC GENERATORS

Each channel contains a CRC generator that adds a series of bits to each of the transmitted data words sent out on the DBUS. The CRC bits are created from the data pattern and are used by the slave devices to determine if one or more of the data bits sent was in error. The detailed operation and control of this function is covered in the section entitled [CRC Generation/Checking on page 20](#).

CRC CHECK

This circuit checks the CRC bits that have been added to the end of the response by the slave device. For a given pattern of received data a new CRC is generated and compared to the CRC bits received. If they do not match, a bit is set in the status register indicating a CRC error for the response. This bit is read back using the SPI during the same SPI transaction that reads the response in order to keep them associated with each other. The CRC bits are removed by the IC and not seen by the MCU when reading the data registers. Operation of the CRC Check is covered in the section entitled [CRC Generation/Checking on page 20](#).

SPI AND PROTOCOL ENGINE STATE MACHINES

Although the SPI clock and the DBUS input clock both typically come from the same MCU system clock in an MCU plus 33780 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 19](#) 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 that 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 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 MC33780, they are intended to include just enough relevant data to understand the operation of the state machines and basic functions.

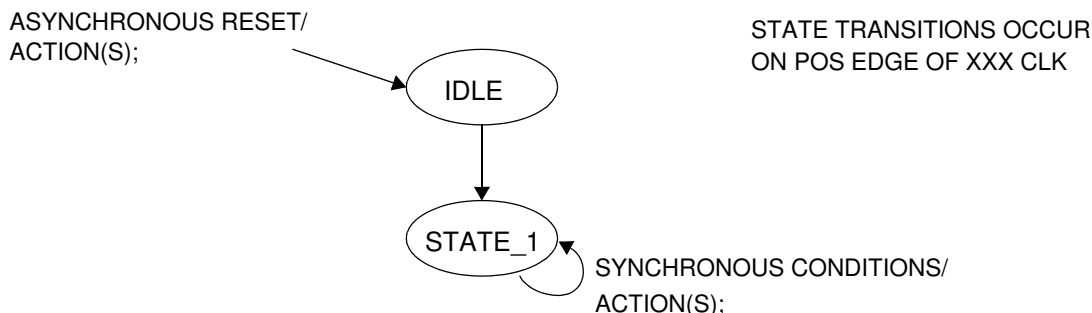


Figure 19. State Diagram Notation

Figure 20, 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 (SCLK). The initial state, SPI_IDLE, is entered asynchronously whenever internal reset becomes active or the SPI chip select (\overline{CS}) input is de-asserted. 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 \overline{CS} 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 SPI_CMD_XFER state, where the remaining bits of the command byte are received. The last five bits of the command set the initial value of the register pointer. After the command byte is complete, the state machine advances to the SPI_BURST

state, which remains active until \overline{CS} goes high (or the MC33780 is reset).

In the SPI_BURST state, new SPI characters are read-from, or written-to-and-read-from, MC33780 registers. If the control register (or CRC polynomial, CRC seed, CRC length, or spread spectrum control) is written, an ABORT request is generated that will immediately stop any DBUS transfer that was in progress (refer to the DBUS transfer state diagram). If the DATA register low byte is written, a transmit FIFO push operation is generated (see transmit FIFO state diagram). 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 DBUS 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.

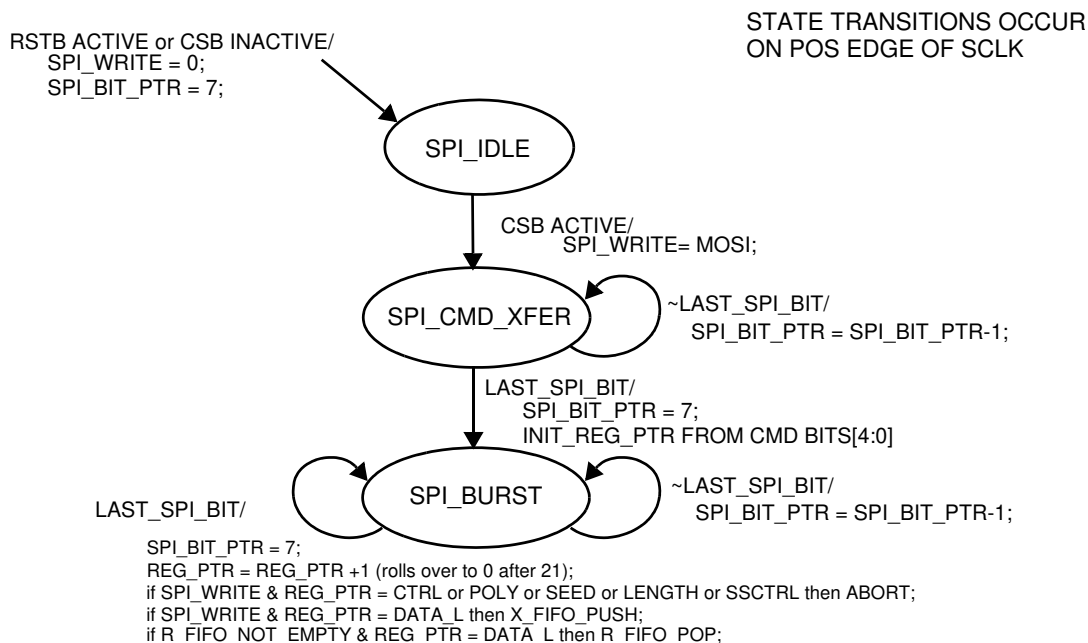


Figure 20. State Diagram of SPI Transfer

Figure 21 describes what happens during DBUS serial transfers. State transfers in this state machine are synchronous with positive edges on the scaled DBUS 1/3rd bit clock and the initial state is WAIT_FRAME_DLY. 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, WAIT_SIG_DLY_0, needs to be synchronous.

When enable is true and there is at least one valid entry in the transmit FIFO, the DBUS frame signal is driven low to start a frame. States WAIT_SIG_DLY_0 through WAIT_SIG_DLY_2 create a one DBUS bit-time delay before the start of the first data bit. After WAIT_SIG_DLY_2, the DBUS_BIT_PTR gets initialized to the total word length, as determined by the MSx, SWLENx, and CRCLenx bits. The XFER_DBUS_BIT_0 state is then entered.

XFER_DBUS_BIT_0 through XFER_DBUS_BIT_2 form a loop where each pass corresponds to one DBUS bit time. During the first third of the bit the DSIS signal is low, during the second third DSIS is low for a zero or high for a one, during the last third of the bit time DSIS 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 DBUS_R_PUSH state is entered. This state ensures that the CRC flag is stable prior to adjusting the receive (and transmit) FIFO pointers. The DBUS_X_POP state prevents an X_FIFO_POP from occurring at the same time as an R_FIFO_PUSH.

After DBUS_X_POP, the state transitions back to the WAIT_FRAME_DLY 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 DBUS_R_PUSH and DBUS_X_POP states are being processed.

Figure 22 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 DBUS serial transfer. State transitions in this state machine occur at the trailing edges of X_FIFO_PUSH and X_FIFO_POP.

When this FIFO is completely empty, the SPI can write four new values to fill the FIFO without waiting for any action on the DBUS side of the FIFO. Values are pushed into the FIFO from the SPI interface and values are popped after they have been serially sent out of the DBUS interface. 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 zero causes 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 incremental, 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.

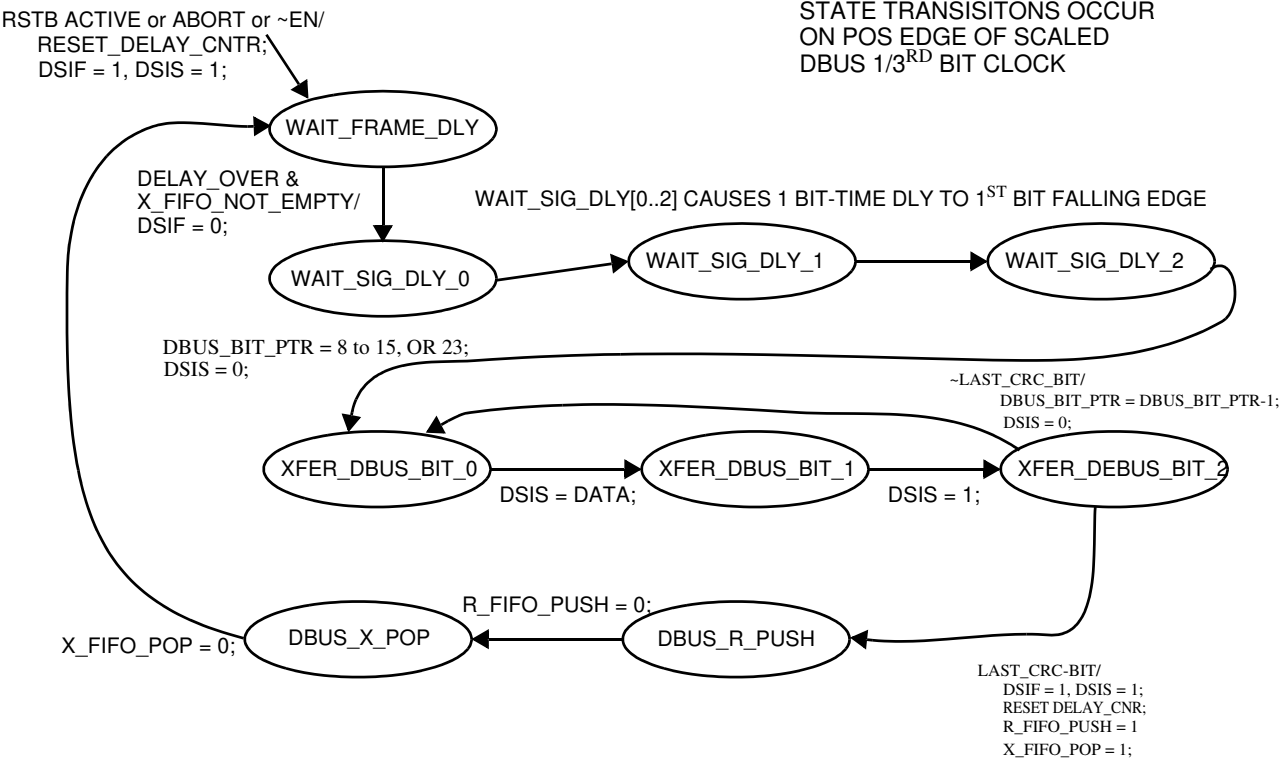


Figure 21. State Diagram of DBUS Transfer