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

USB5537B

7-Port SS/HS USB Hub Controller

PRODUCT FEATURES

Datasheet

General Description

The SMSC USB5537B hub is a 7-port SuperSpeed/Hi-Speed, low-power, configurable hub controller family fully compliant with the *USB 3.0 Specification*. The USB5537B supports 5 Gbps SuperSpeed (SS), 480 Mbps Hi-Speed (HS), 12 Mbps Full-Speed (FS) and 1.5 Mbps Low-Speed (LS) USB signalling for complete coverage of all defined USB operating speeds.

The USB5537B supports legacy USB speeds through its USB 2.0 hub controller. The new SuperSpeed hub controller (available on 4 of the 7 ports) operates in parallel with the USB 2.0 controller, so the 5 Gbps SuperSpeed data transfers are not affected by the slower USB 2.0 traffic.

The USB5537B supports battery charging on a per port basis. On battery charging enabled ports, the devices provide automatic USB data line handshaking. The handshaking supports USB 1.2 Charging Downstream Port (CDP), Dedicated Charging Port (DCP) and legacy devices.

The USB5537B is configured for operation through internal default settings, where custom configurations are supported through an on-chip OTP ROM, an external SPI ROM, or SMBus.

All LED and port control signal pins are under firmware control in order to allow for maximum operational flexibility; those pins can also be configured as GPIOs.

Features

- USB 3.0 compliant 5 Gbps, 480 Mbps, 12 Mbps and 1.5 Mbps operation, USB pins are 5 V tolerant
  - Integrated termination and pull-up/pull-down resistors
- Four downstream USB 3.0 ports
- Three additional USB 2.0 ports for use cases where SS is not required
- Supports battery charging of most popular battery powered devices
  - USB-IF Battery Charging rev. 1.2 support (DCP & CDP)
  - Apple Portable product charger emulation
  - Blackberry charger emulation
  - Chinese YD/T 1591-2006 charger emulation
  - Chinese YD/T 1591-2009 charger emulation
  - Supports additional portable devices
- Emulates portable/handheld native wall chargers
  - Charging profiles emulate a handheld device's wall charger to enable fast charging (minutes vs. hours)
- Enables charging from a mobile platform that is off
- Support tablets' high current requirements
- Optimized for low-power operation and low thermal dissipation
- Vendor Specific Messaging (VSM) support for firmware upload over USB
- Configuration via OTP ROM, SPI ROM, or SMBus
- On-chip 8051 µC manages GPIOs, VBUS, and other hub signals
- 8 K RAM, 32 K ROM
- One Time programmable (OTP) ROM: 8 kbit
  - Includes on-chip charge pump
- Single 25 MHz XTAL or clock input for all on-chip PLL and clocking requirements
- Supports JTAG boundary scan
- PHYBoost (USB 2.0)
  - Selectable drive strength for improved signal integrity
- VariSense (USB 2.0)
  - controls the receiver sensitivity enabling four programmable levels of USB signal receive sensitivity
- IETF RFC 4122 compliant 128-bit UUID

Software Features

- Compatible with Microsoft Windows 7, Vista, XP, Mac OSX10.4+, and Linux Hub Drivers
- LEDs (configuration dependent)
  - Multi color LED scheme vividly shows port capability and operating speed
  - Embedded 8051 Micro controller for Hub configuration
- Standard hub I/O (port power, over-current sense, LEDs) are GPIOs controlled by a µC instead of hard coded functions to allow for flexibility for OEM differentiation

Order Numbers:

ORDER NUMBERS*	DESCRIPTION	LEAD-FREE ROHS COMPLIANT PACKAGE	TEMPERATURE RANGE
USB5537B-4100AKZE	Hybrid 4/7-Port Hub with VSM & Apple/BC 1.2 Charging	72QFN 10 x 10 mm 6.0 mm exposed pad	0ºC to 70ºC
USB5537Bi-4100AKZE		72QFN 10 x 10 mm 7.9 mm exposed pad	-40ºC to 85ºC

* Add “TR” to the end of any order number to order tape and reel. Reel size is 3000 pieces.

This product meets the halogen maximum concentration values per IEC61249-2-21

For RoHS compliance and environmental information, please visit www.smsc.com/rohs

Please contact your SMSC sales representative for additional documentation related to this product such as application notes, anomaly sheets, and design guidelines.

Conventions

Within this manual, the following abbreviations and symbols are used to improve readability.

Example	Description
BIT	Name of a single bit within a field
FIELD.BIT	Name of a single bit (BIT) in FIELD
x…y	Range from x to y, inclusive
BITS[m:n]	Groups of bits from m to n, inclusive
PIN	Pin Name
zzzzb	Binary number (value zzzz)
0xzzz	Hexadecimal number (value zzz)
zzh	Hexadecimal number (value zz)
rsvd	Reserved memory location. Must write 0, read value indeterminate
`code`	Instruction code, or API function or parameter
Section Name	Section or Document name
x	Don't care
<Parameter>	<> indicate a Parameter is optional or is only used under some conditions
{,Parameter}	Braces indicate Parameter(s) that repeat one or more times
[Parameter]	Brackets indicate a nested Parameter. This Parameter is not real and actually decodes into one or more real parameters.

Table of Contents

List of Tables

List of Figures

Chapter 1 Block Diagram

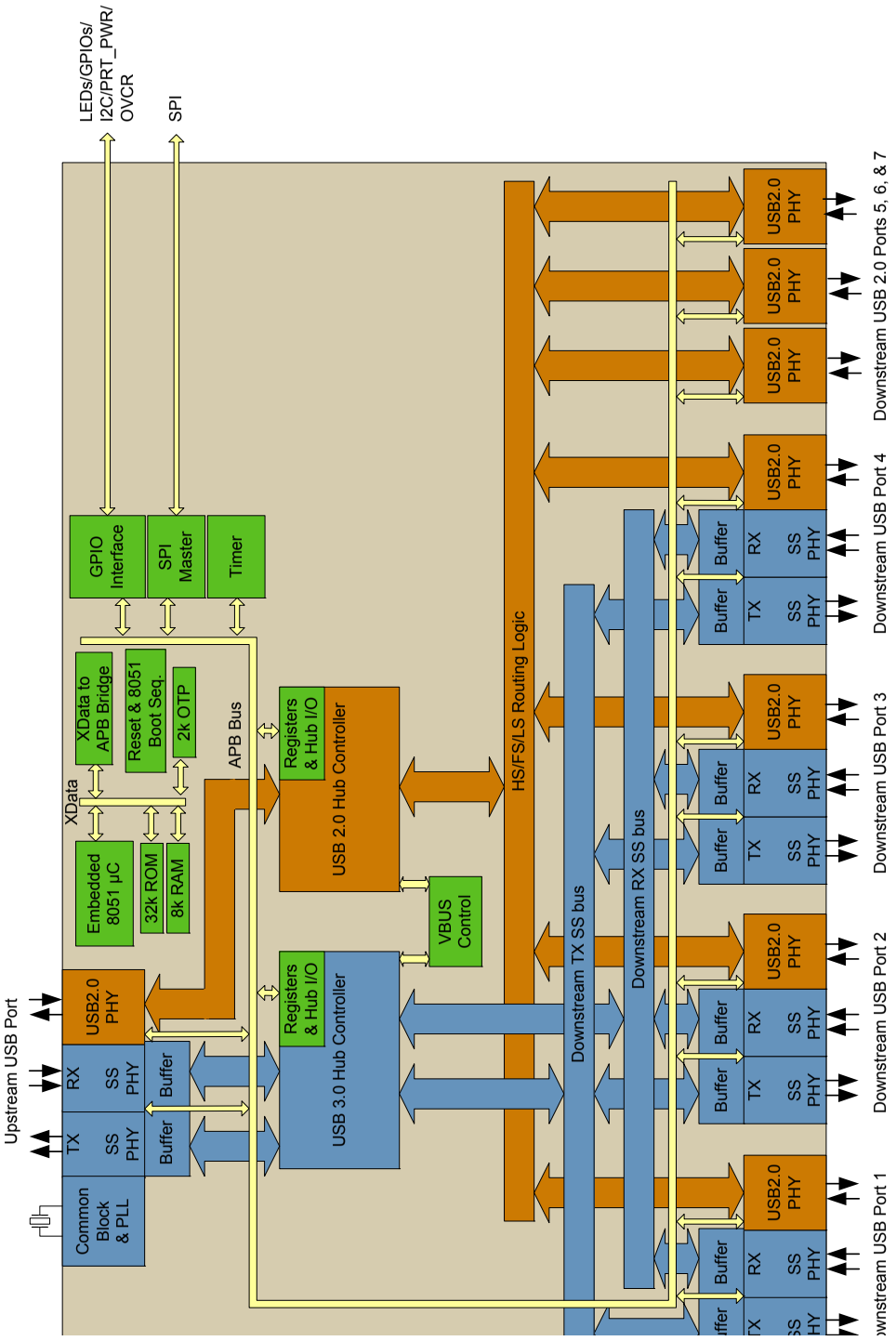

Figure 1.1 USB5537B Block Diagram

Chapter 2 Overview

The SMSC USB5537B hub is a 7-port, low-power, configurable Hub Controller fully compliant with the *USB 3.0 Specification* [2]. The USB5537B supports 5 Gbps SuperSpeed (SS), 480 Mbps Hi-Speed (HS), 12 Mbps Full-Speed (FS) and 1.5 Mbps Low-Speed (LS) USB signalling for complete coverage of all defined USB operating speeds.

All required resistors on the USB ports are integrated into the hub. This includes all series termination resistors and all required pull-down and pull-up resistors on D+ and D- pins. The over-current sense inputs for the downstream facing ports have internal pull-up resistors.

The USB5537B hub includes programmable features such as:

- **MultiTRAK™ Technology**: implements a dedicated Transaction Translator (TT) for each port. Dedicated TTs help maintain consistent full-speed data throughput regardless of the number of active downstream connections.
- **PortSwap**: allows direct alignment of USB signals (D+/D-) to connectors to avoid uneven trace length or crossing of the USB differential signals on the PCB.
- **PHYBoost**: enables 4 programmable levels of USB signal drive strength in downstream port transceivers (HS, FS, LS USB only). PHYBoost will also attempt to restore USB signal integrity.

Note: At the time of this writing, products with a USB Host (or products such as docking stations with USB hubs that must be submitted with an attached host enabled platform) are eligible for USB logo with a mixture of USB2.0 and USB3.0 ports. An example of such a product is a dedicated docking station that can only be used with a specific model or family of laptops. In this example, the docking station is submitted for USB logo testing with a laptop and it is tested as a system. USB devices containing a hub with an exposed upstream port that can be plugged into any host platform through USB cables and connectors (examples are USB enabled monitors, printers, hard drives, etc.) are not eligible for logo with a mixture of USB 3.0 and USB 2.0 ports. These devices will need to consume the USB 2.0-only ports internally in order to be eligible for a USB logo.

2.1 Configurable Features

The SMSC USB5537B hub controller provides a default configuration that is sufficient for most applications. When the hub is initialized in the default configuration, the following features may be configured:

- Downstream non-removable ports, where the hub will automatically report as a compound device
- Downstream disabled ports
- Downstream port power control and over-current detection on a ganged or individual basis
- USB signal drive strength
- USB differential pair pin location

The USB5537B hub controllers can alternatively be configured by OTP or as an SMBus slave device. When the hub is configured by an OTP or over SMBus, the following configurable features are provided:

- Support for compound devices on a port-by-port basis
- Selectable over-current sensing and port power control on an individual or ganged basis to match the circuit board component selection
- Customizable vendor ID, product ID, and device ID
- Configurable delay time for filtering the over-current sense inputs
- Indication of the maximum current that the hub consumes from the USB upstream port
- Indication of the maximum current required for the hub controller
- Custom string descriptors (up to 30 characters):
  Product, manufacturer, and serial number

Chapter 3 Pin Information

This chapter outlines the pinning configurations for each chip. The detailed pin descriptions are listed by function in Section 3.2: *Pin Descriptions (Grouped by Function)* on page 11.

3.1 Pin Configurations

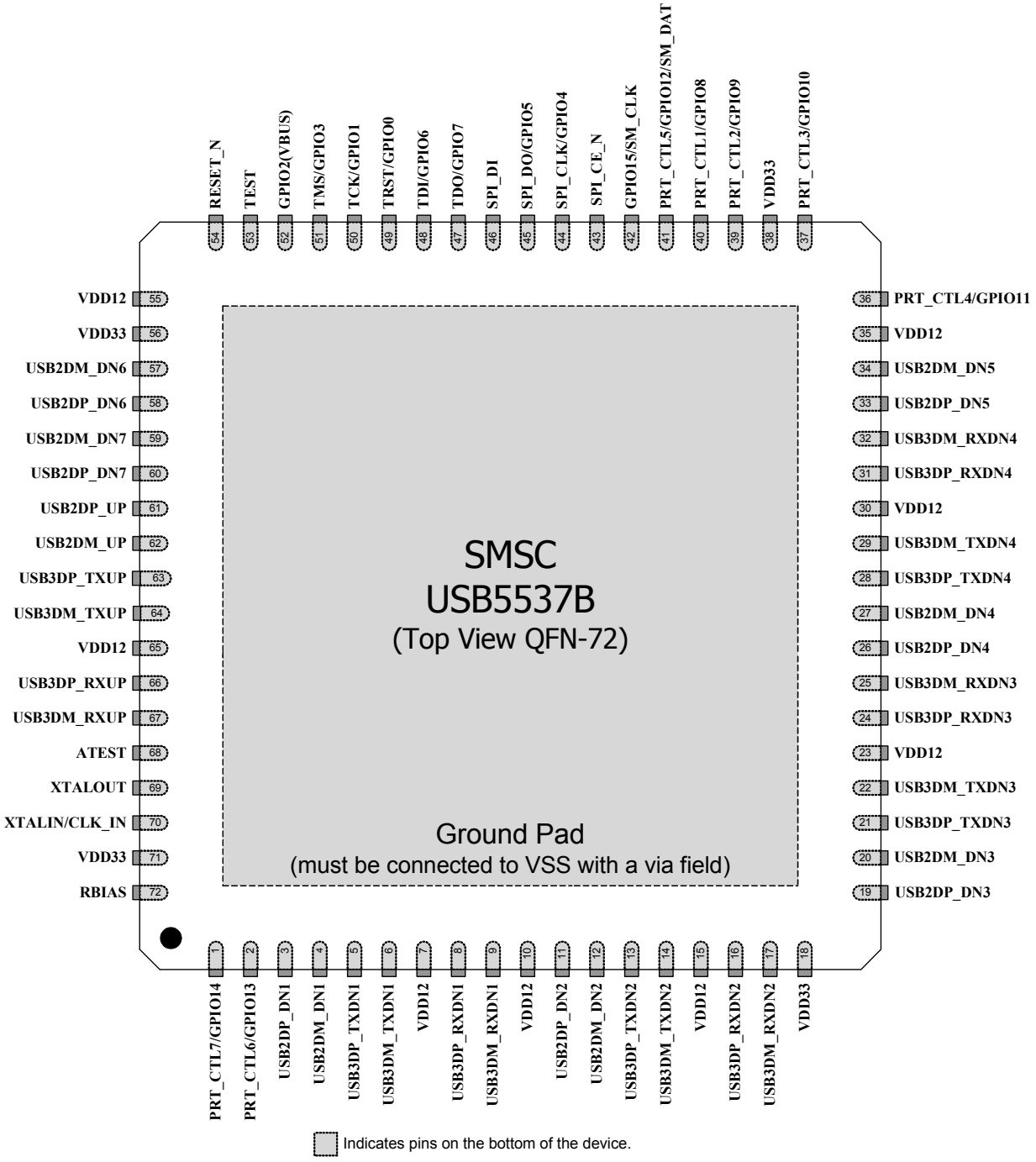

Figure 3.1 USB5537B 72-pin QFN

3.2 Pin Descriptions (Grouped by Function)

An *N* at the end of a signal name indicates that the active (asserted) state occurs when the signal is at a low voltage level. When the *N* is not present, the signal is asserted when it is at a high voltage level. The terms assertion and negation are used exclusively in order to avoid confusion when working with a mixture of active low and active high signals. The term assert, or assertion, indicates that a signal is active, independent of whether that level is represented by a high or low voltage. The term negate, or negation, indicates that a signal is inactive.

SYMBOL	BUFFER TYPE	DESCRIPTION
USB 3.0 INTERFACE		
USB3DP_TXUP	IO-U	USB 3 Upstream Upstream SuperSpeed transmit data plus
USB3DM_TXUP	IO-U	USB 3 Upstream Upstream SuperSpeed transmit data minus
USB3DP_RXUP	IO-U	USB 3 Upstream Upstream SuperSpeed receive data plus
USB3DM_RXUP	IO-U	USB 3 Upstream Upstream SuperSpeed receive data minus
USB3DP_TXDN[4:1]	IO-U	USB 3 Downstream Downstream SuperSpeed transmit data plus for ports 1 through 4.
USB3DM_TXDN[4:1]	IO-U	USB 3 Downstream Downstream SuperSpeed transmit data minus for ports 1 through 4.
USB3DP_RXDN[4:1]	IO-U	USB 3 Downstream Downstream SuperSpeed receive data plus for ports 1 through 4.
USB3DM_RXDN[4:1]	IO-U	USB 3 Downstream Downstream SuperSpeed receive data minus for ports 1 through 4.
USB 2.0 INTERFACE		
USB2DP_UP	IO-U	USB Bus Data These pins connect to the upstream USB bus data signals.
USB2DM_UP	IO-U	USB Bus Data These pins connect to the upstream USB bus data signals.
USB2DP_DN[7:1]	IO-U	Hi-Speed USB Data Downstream Hi-Speed data plus for ports 1 through 7.
USB2DM_DN[7:1]	IO-U	Hi-Speed USB Data Downstream Hi-Speed data minus for ports 1 through 7.

Table 3.1 USB5537B Pin Descriptions

SYMBOL	BUFFER TYPE	DESCRIPTION
USB PORT CONTROL		
PRT_PWR[7:1]/ PRT_CTL[7:1]/ GPIO[14:8]	O12	USB Power Enable Enables power to USB peripheral devices downstream.
GPIO2 (VBUS)	I/O12	Upstream VBUS Power Detect This pin can be used to detect the state of the upstream bus power.
SPI INTERFACE		
SPI_CE_N	I/O12	SPI Enable
SPI_CLK	I/O12	SPI Clock
GPIO4		This pin may be used either as an input; edge sensitive interrupt input; or output. Custom firmware is required to activate this function.
SPI_DO	I/O12PD	SPI Serial Data Out The output for the SPI port.
GPIO5		General Purpose I/O Pin 5 This pin may be used either as an input; edge sensitive interrupt input; or output. Custom firmware is required to activate this function.
SPI_DI	I/O12	SPI Serial Data In The SPI data in to the controller from the ROM. This pin has a weak internal pull-down applied at all times to prevent floating.
JTAG/LED/OCS INTERFACE		
TRST	I/O12PD	JTAG Asynchronous Reset
LED0/ GPIO0		Customizable LED Output 0
TCK	I/O12	JTAG Clock This input is used for JTAG boundary scan and has a weak pull-down. It can be left floating or grounded when not used. If the JTAG is connected, then this signal will be detected high, and the software disables the pull up after reset.
LED1		Customizable LED Output 1
GPIO1 (OCS1)		Over-Current Sense 1 Input from external current monitor indicating an over-current condition.
TMS	I/O12	JTAG TMS Used for JTAG boundary scan.
GPIO3 (OCS2)		Over-Current Sense 2 Input from external current monitor indicating an over-current condition.

Table 3.1 USB5537B Pin Descriptions (continued)

SYMBOL	BUFFER TYPE	DESCRIPTION
TDI	I/O12	JTAG TDI Used for JTAG boundary scan.
GPIO6 (OCS3)		Over-Current Sense 3 Input from external current monitor indicating an over-current condition.
TDO	I/O12	JTAG TDO Used for JTAG boundary scan.
GPIO7 (OCS4)		Over-Current Sense 4 Input from external current monitor indicating an over-current condition.
MISC		
RESET_N	IS	Reset Input The system uses this active low signal to reset the chip. The active low pulse should be at least 1 µs wide.
XTALIN	ICLKx	Crystal Input: 25 MHz crystal. This pin connects to either one terminal of the crystal or to an external 25 MHz clock when a crystal is not used.
CLK_IN		External Clock Input This pin connects to either one terminal of the crystal or to an external 25 MHz clock when a crystal is not used.
XTALOUT	OCLKx	Crystal Output The clock output, providing a crystal 25 MHz. When an external clock source is used to drive **XTALIN/CLKIN**, this pin becomes a no connect.
TEST	IPD	Test Pin Treat as a no connect pin or connect to ground. No trace or signal should be routed or attached to this pin.
RBIAS	I-R	USB Transceiver Bias A12.0 kΩ (+/- 1%) resistor is attached from ground to this pin to set the transceiver's internal bias settings.
ATEST	A	Analog Test Pin This signal is used for testing the chip and must always be connected to ground.
GPIO15	I/O12	General Purpose I/O Pin 15
SM_CLK		SMBus Clock
SM_DAT	I/O12	SMBus Data Pin **Note:** This pin is MUXed with **PRT_CTRL5/GPIO12**
DIGITAL AND POWER (12 PINS AND 1 GROUND PAD)		
(4) **VDD33**		3.3 V Power

Table 3.1 USB5537B Pin Descriptions (continued)

SYMBOL	BUFFER TYPE	DESCRIPTION
(8) **VDD12**		1.25 V Power
VSS		Ground Pad This exposed pad is the device's only connection to VSS and the primary thermal conduction path. Connect to an appropriate via field.

Table 3.1 USB5537B Pin Descriptions (continued)

3.3 Buffer Type Descriptions

Table 3.2 Buffer Type Descriptions

BUFFER TYPE	DESCRIPTION
I	Input
I/O	Input/output
IPD	Input with internal weak pull-down resistor
IPU	Input with internal weak pull-up resistor
IS	Input with Schmitt trigger
O12	Output 12 mA
I/O12	Input/output buffer with 12 mA sink and 12 mA source
I/OSD12	Open drain with Schmitt trigger and 12 mA sink.
ICLKx	XTAL clock input
OCLKx	XTAL clock output
I-R	RBIAS
I/O-U	Analog input/output defined in USB specification

Chapter 4 Configuration Options

The USB5537B must be configured in order to correctly function when attached to a USB host controller. The hub can be configured either internally or externally depending on the implemented interface (see Chapter 5: *Interfacing to the USB5537B* on page 16 for details).

4.1 SPI ROM

When the SPI interface is configured, the USB5537B is will perform code execution from an external SPI ROM.

4.2 SMBus

Two SMBus modes (based on the used slave address) are available: Legacy and Advanced.

4.2.1 SMBus Legacy Mode

The SMBus Legacy Mode provides access to all internal USB 2.0 registers, and is enabled based on the 7-bit slave address of 0101100b. The hub will not respond to the general call address of 0000000b.

4.2.2 SMBus Advanced Mode

The SMBus Advanced Mode provides access to all USB 2.0 and USB 3.0 registers, and is enabled based on the 7-bit slave address of 0101101b. The hub will not respond to the general call address of 0000000b. The protocol is based on the SMBus block read/write, except the register offset is extended to 16 bits (high byte, low byte).

Chapter 5 Interfacing to the USB5537B

The hub will interface to external memory depending on configuration of the USB5537B pins associated with each interface type. The USB5537B will first check to see whether an external SPI Flash is present. If present, the chip will operate entirely from the external ROM. When an external SPI Flash is not present, the USB5537B will look to see whether SMBus is configured. When SMBus is enabled, the SMBus can operate in either legacy (USB 2.0 only) or advanced mode (access to both USB 2.0 and 3.0 registers). If no external options are detected, the USB5537B will operate from the internal OTP memory.

5.1 SPI Interface

The USB5537B is capable of code execution from an external SPI ROM. On power up, the firmware looks for an external SPI flash device that contains a valid signature of `2DFU` (device firmware upgrade) beginning at address 0xFFFA. If a valid signature is found, then the external ROM is enabled and the code execution begins at address 0x0000 in the external SPI device. If a valid signature is not found, then execution continues from internal ROM. The following sections describe the interface options to the external SPI ROM.

5.1.1 Operation of the Hi-Speed Read Sequence

The SPI controller will automatically handle code reads going out to the SPI ROM Address. When the controller detects a read, the controller drops the **SPI_CE**, and puts out a 0x0B, followed by the 24-bit address. The SPI controller then puts out a DUMMY byte. The next eight clocks clock in the first byte. When the first byte is clocked in a ready signal is sent back to the processor, and the processor gets one byte.

After the processor gets the first byte, its address will change. If the address is one more than the last address, the SPI controller will clock out one more byte. If the address in anything other than one more than the last address, the SPI controller will terminate the transaction by taking **SPI_CE** high. As long as the addresses are sequential, the SPI Controller will keep clocking in data.

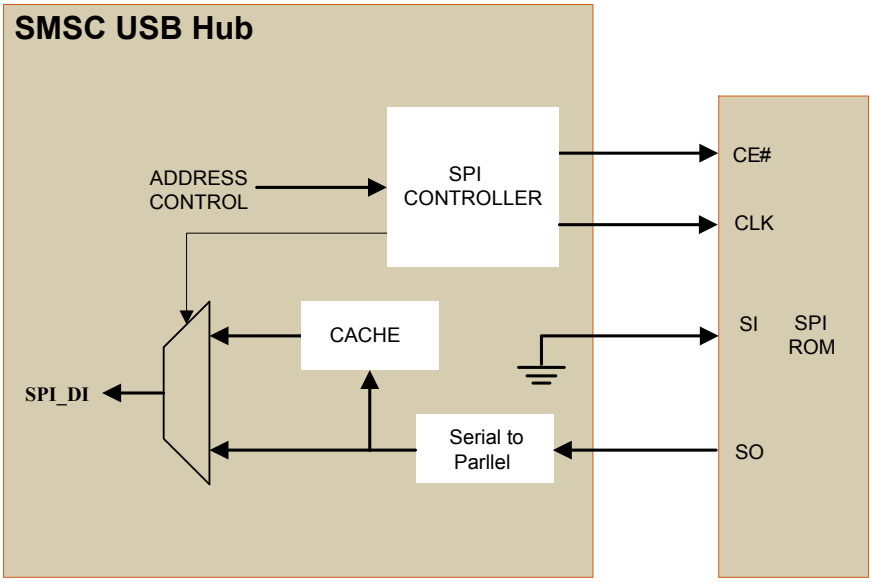

Figure 5.1 SPI Hi-Speed Read Operation

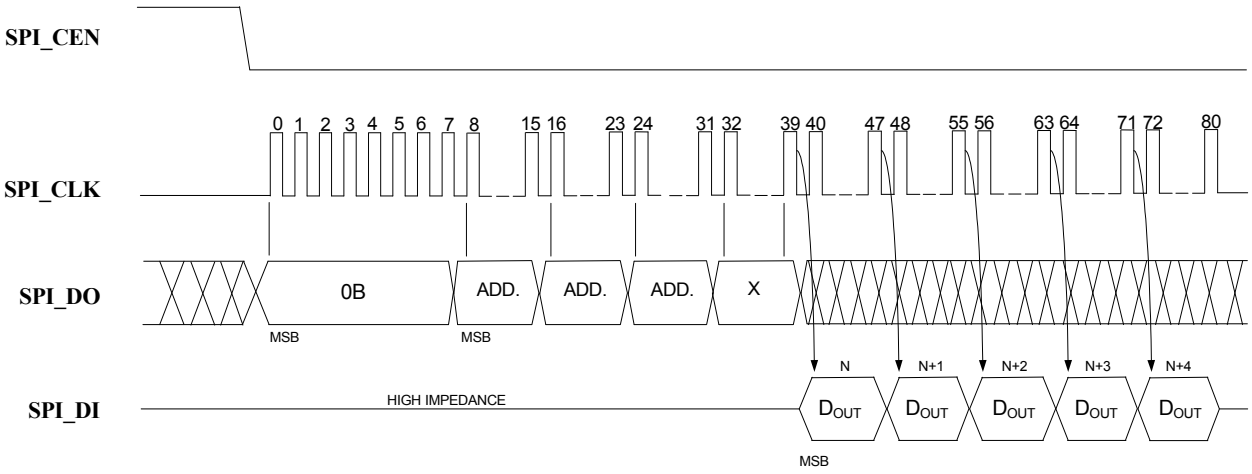

Figure 5.2 SPI Hi-Speed Read Sequence

5.1.2 Operation of the Dual Hi-Speed Read Sequence

The SPI controller will also support dual data mode. When configured in dual mode, the SPI controller will automatically handle reads going out to the SPI ROM. When the controller detects a read, the controller drops the **SPI_CE_N**, and puts out a 0x3B, followed by the 24-bit address. The SPI controller then puts out a DUMMY byte. The next four clocks clock in the first byte. The data appears two bits at a time on data out and data in. When the first byte is clocked in a ready signal is sent back to the processor, and the processor gets one byte.

After the processor gets the first byte, the address will change. If the address is one more than the last address, the SPI controller will clock out one more byte. If the address in anything other than one more than the last address, the SPI controller will terminate the transaction by taking **SPI_CE_N** high. As long as the addresses are sequential, the SPI Controller will keep clocking in data.

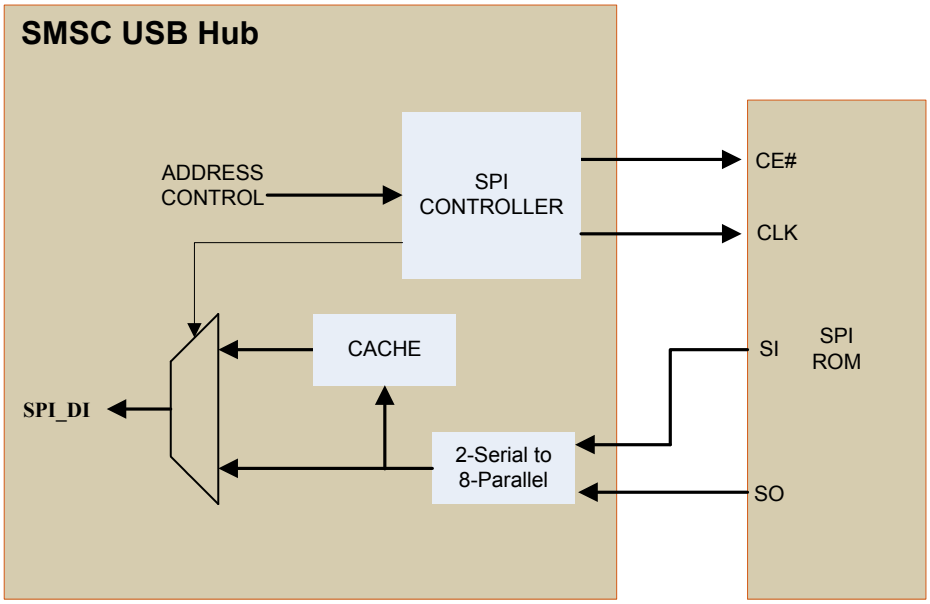

Figure 5.3 SPI Dual Hi-Speed Read Operation

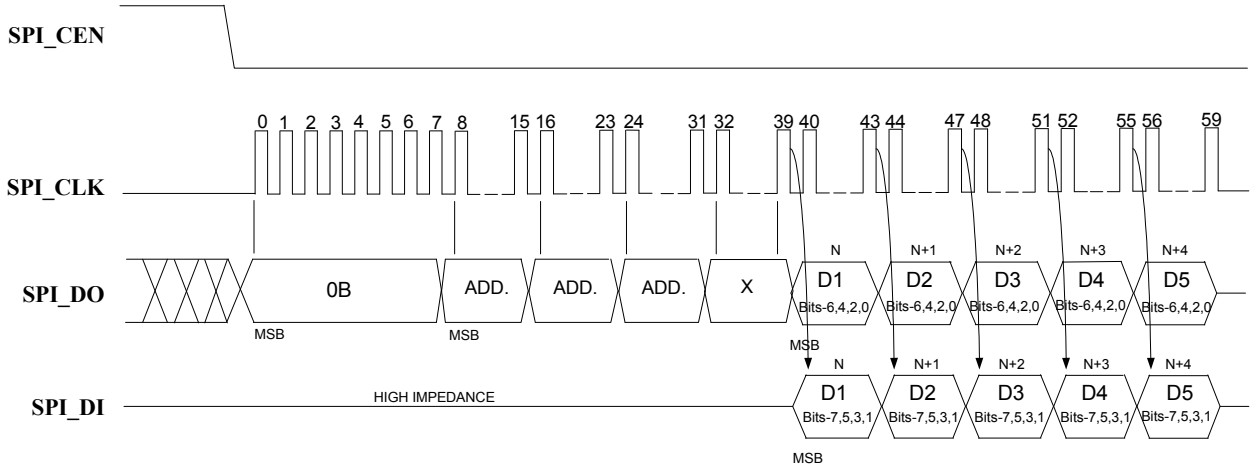

Figure 5.4 SPI Dual Hi-Speed Read Sequence

5.1.3 32-Byte Cache

There is a 32-byte pipeline cache, and associated with the cache is a base address pointer and a length pointer. Once the SPI controller detects a jump, the base address pointer is initialized to that address. As each new sequential data byte is fetched, the data is written into the cache, and the length is incremented. If the sequential run exceeds 32 bytes, the base address pointer is incremented to indicate the last 32 bytes fetched. If the USB5537B does a jump, and the jump is in the cache address range, the fetch is done in 1 clock from the internal cache instead of an external access.

5.1.4 Interface Operation to SPI Port When Not Doing Fast Reads

There is an 8-byte command buffer: SPI_CMD_BUF[7:0]; an 8-byte response buffer: SPI_RESP_BUF[7:0]; and a length register that counts out the number of bytes: SPI_CMD_LEN. Additionally, there is a self-clearing **GO** bit in the SPI_CTL Register. Once the **GO** bit is set, the device drops **SPI_CE_N**, and starts clocking. It will put out SPI_CMD_LEN X 8 number of clocks. After the first byte, the COMMAND, has been sent out, and the **SPI_DI** is stored in the SPI_RESP buffer. If the SPI_CMD_LEN is longer than the SPI_CMD_BUF, don't cares are sent out on the **SPI_DO** line. This mode is used for program execution out of internal RAM or ROM.

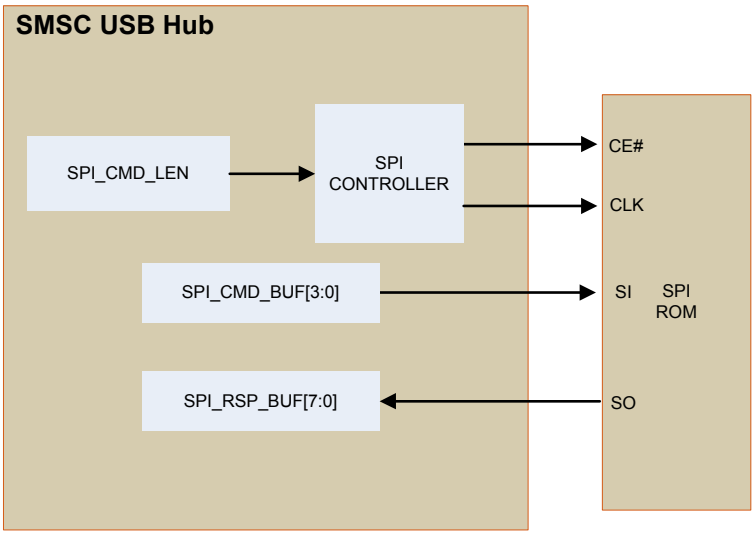

Figure 5.5 SPI Internally-Controlled Operation

5.1.4.1 ERASE EXAMPLE

To perform a SCTR_ERASE, 32BLK_ERASE, or 64BLK_ERASE, the device writes 0x20, 0x52, or 0xD8, respectively to the first byte of the command buffer, followed by a 3-byte address. The length of the transfer is set to 4 bytes. To do this, the device first drops **SPI_CE_N**, then counts out 8 clocks. It then puts out the 8 bits of command, followed by 24 bits of address of the location to be erased on the **SPI_DO** pin. When the transfer is complete, the **SPI_CE_N** goes high, while the **SPI_DI** line is ignored in this example.

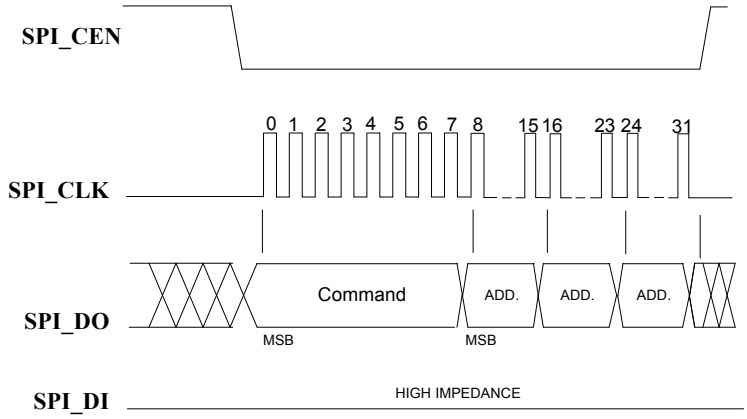

Figure 5.6 SPI Erase Sequence

5.1.4.2 BYTE PROGRAM EXAMPLE

To perform a Byte Program, the device writes 0x02 to the first byte of the command buffer, followed by a 3-byte address of the location that will be written to, and one data byte. The length of the transfer is set to 5 bytes. The device first drops **SPI_CE_N**, 8 bits of command are clocked out, followed by 24 bits of address, and one byte of data on the **SPI_DO** pin. The **SPI_DI** line is not used in this example.

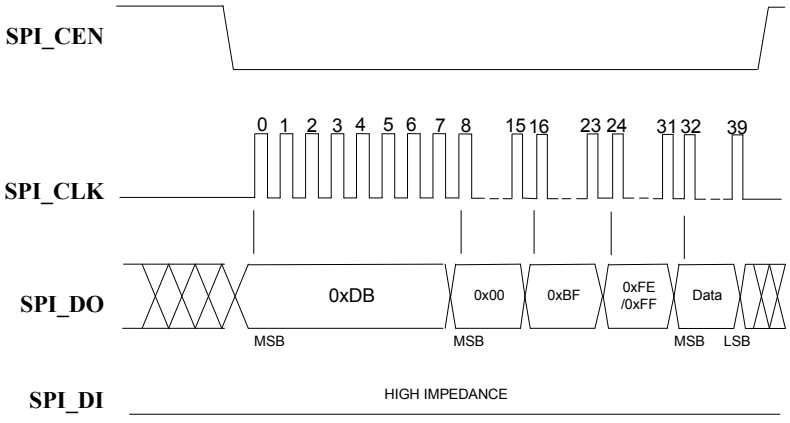

Figure 5.7 SPI Byte Program

5.1.4.3 COMMAND ONLY PROGRAM EXAMPLE

To perform a single byte command such as the following:

- WRDI
- WREN
- EWSR
- CHIP_ERASE
- EBSY
- DBSY

The device writes the opcode into the first byte of the SPI_CMD_BUF and the SPI_CMD_LEN is set to one. The device first drops **SPI_CE**, then 8 bits of the command are clocked out on the **SPI_DO** pin. The **SPI_DI** is not used in this example.

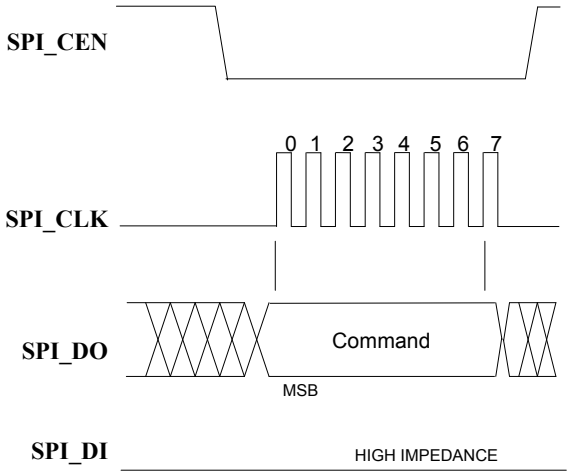

Figure 5.8 SPI Command Only Sequence

5.1.4.4 JEDEC-ID READ EXAMPLE

To perform a JEDEC-ID command, the device writes 0x9F into the first byte of the SPI_CMD_BUF and the length of the transfer is 4 bytes. The device first drops **SPI_CE_N**, then 8 bits of the command are clocked out, followed by the 24 bits of dummy bytes (due to the length being set to 4) on the **SPI_DO** pin. When the transfer is complete, the **SPI_CE_N** goes high. After the first byte, the data on **SPI_DI** is clocked into the SPI_RSP_BUF. At the end of the command, there are three valid bytes in the SPI_RSP_BUF. In this example, 0xBF, 0x25, 0x8E.

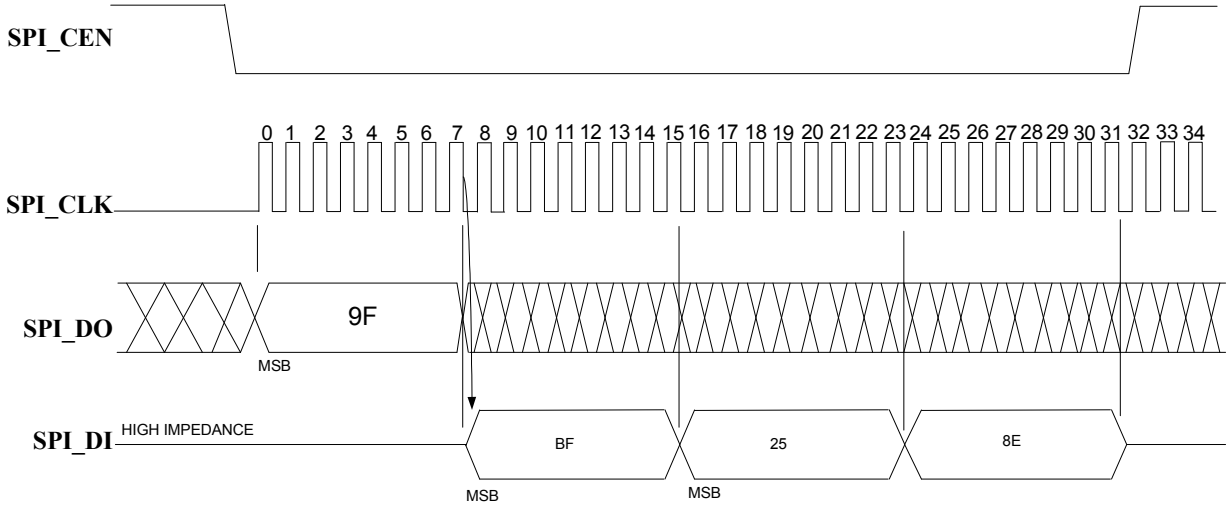

Figure 5.9 SPI JEDEC-ID Sequence

5.1.5 SPI Timing

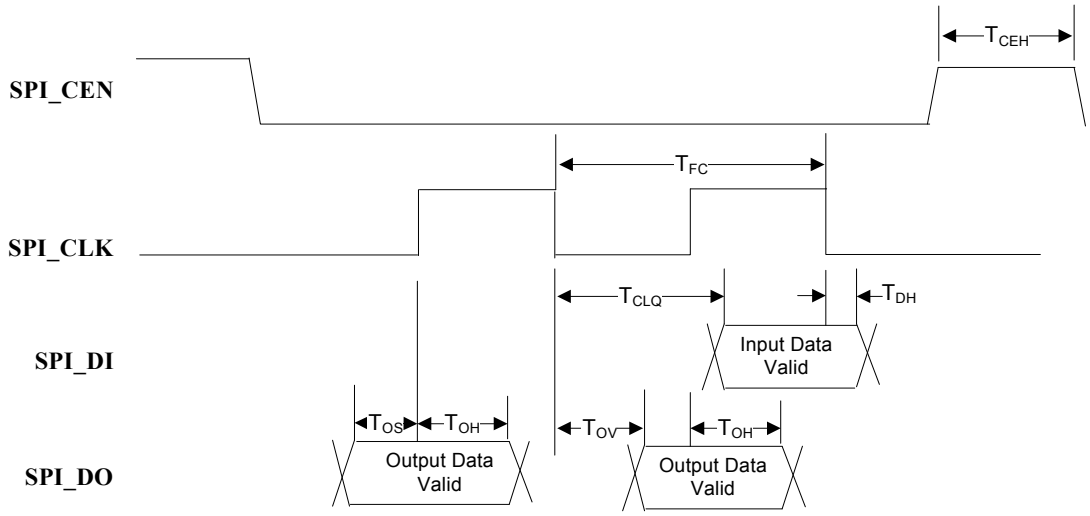

Figure 5.10 SPI Timing

Name	Parameter	Min	Max	Unit
T_{FC}	Clock Frequency		60	MHz
T_{CEH}	Chip Enable High Time	50		ns
T_{CLQ}	Clock to Input Data		9	ns
T_{DH}	Input Data Hold Time	0		ns
T_{OS}	Output Set up Time	5		ns
T_{OH}	Output Hold Time	5		ns
T_{OV}	Clock to Output Valid	4		ns

Table 5.1 SPI Timing Operation

5.2 SMBus Slave Interface

The SMBus slave interface is enabled when pull-up resistors are detected on both **SM_DAT** and **SM_CLK** for the first millisecond after reset. If the SMBus interface is enabled, then the USB5537B will wait indefinitely for the SMBus host to configure the device. Once SMBus configuration is complete, device initialization will proceed. To disable the SMBus, a pull-down resistor of 10 KΩ must be applied to **SM_DAT**. If SMBus is disabled, the device proceeds directly to device initialization using either an external I^2C (if present) and the internal OTP ROM.

5.2.1 Pull-Up Resistor for SMBus

External pull-up resistors (10 kΩ recommended) are required on the **SM_DAT** and **SM_CLK** pins when implementing either SMBus mode.

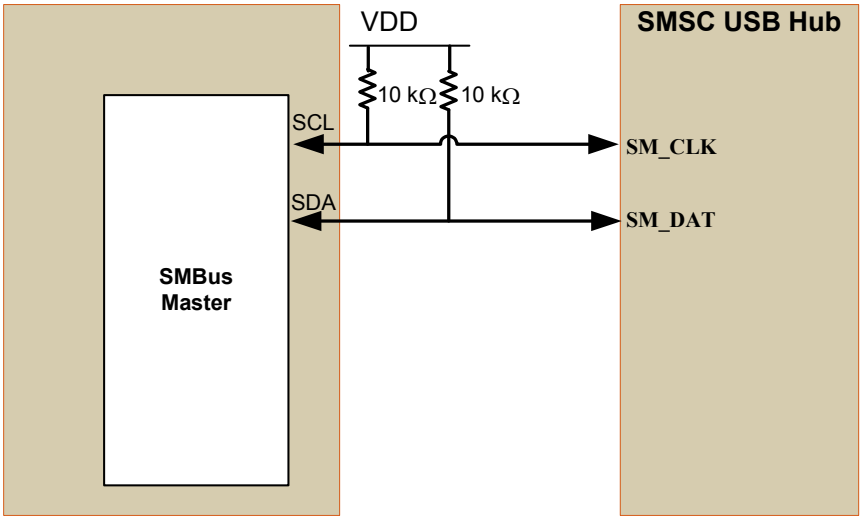

Figure 5.11 SMBus Slave Connection

5.2.2 Protocol Implementation

Typical block write and block read protocols are shown in Figure 5.12 and Figure 5.13. Register accesses are performed using 7-bit slave addressing, an 8- or 16-bit register address field (for legacy and advanced modes, respectively), and an 8-bit data field. The shading shown in the figures during a read or write indicates the hub is driving data on the **SM_DAT** line; otherwise, host data is on the **SM_DAT** line.

The SMBus slave address assigned to the hub (0101100b or 0101101b) allows it to be identified on the SMBus. The register address field is the internal address of the register to be accessed. The register data field is the data that the host is attempting to write to the register or the contents of the register that the host is attempting to read.

Note: Data bytes are transferred MSB first.

5.2.2.1 Block Write/Read

The block write begins with a slave address and a write condition. After the command code, the host issues a byte count which describes how many more bytes will follow in the message. If a slave had 20 bytes to send, the first byte would be the number 20 (14h), followed by the 20 bytes of data. The byte count may not be zero. A block write or read allows a transfer maximum of 32 data bytes.

Note: For the following SMBus tables:

Denotes Master-to-Slave Denotes Slave-to-Master

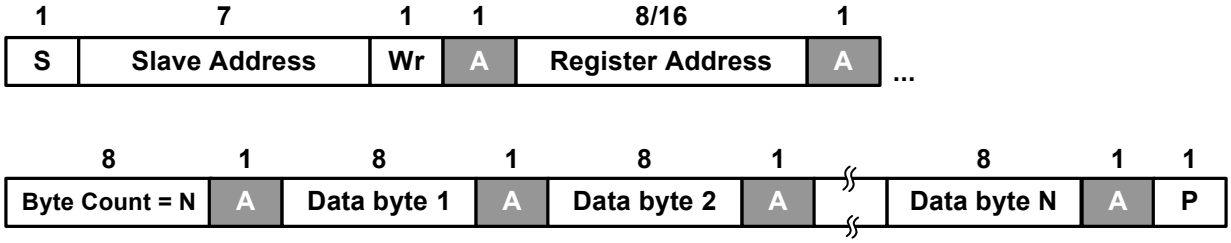

Figure 5.12 Block Write

5.2.2.2 Block Read

A block read differs from a block write in that the repeated start condition exists to satisfy the I^2C specification's requirement for a change in the transfer direction.

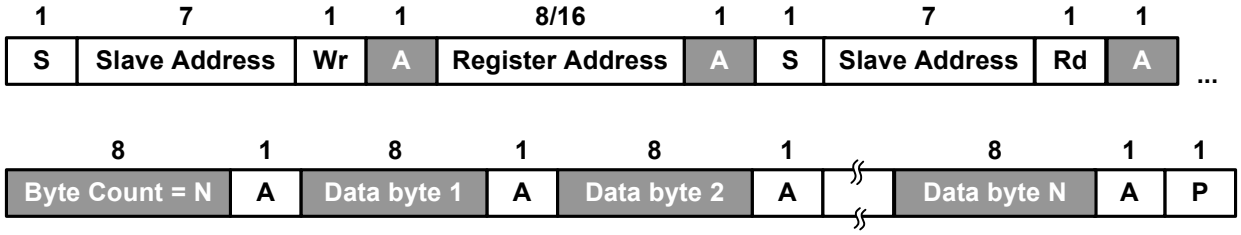

Figure 5.13 Block Read

5.2.2.3 Invalid Protocol Response Behavior

Note that any attempt to update registers with an invalid protocol will not be updated. The only valid protocols are write block and read block (described above), where the hub only responds to the 7-bit hardware selected slave addresses (0101100b or 0101101b). Additionally, the only valid registers for the hub are outlined in the *USB5537B Configuration Release Notes* documentation.

5.2.3 Slave Device Timeout

Devices in a transfer can abort the transfer in progress and release the bus when any single clock low interval exceeds 25 ms ($T_{TIMEOUT, MIN}$). The master must detect this condition and generate a stop condition within or after the transfer of the interrupted data byte. Slave devices must reset their communication and be able to receive a new START condition no later than 35 ms ($T_{TIMEOUT, MAX}$).

Note: Some simple devices do not contain a clock low drive circuit; this simple kind of device typically resets its communications port after a start or stop condition. The slave device timeout must be implemented.

5.2.4 Stretching the SCLK Signal

The hub supports stretching of the SCLK by other devices on the SMBus. The hub will stretch the clock as needed.

5.2.5 Bus Reset Sequence

The SMBus slave interface resets and returns to the idle state upon a START condition followed immediately by a STOP condition.

5.2.6 SMBus Alert Response Address

The SMBALERT# signal is not supported by the USB5537B.

5.2.7 SMBus Timing

The SMBus slave interface complies with the *SMBus Specification Revision 1.0*. See Section 2.1, *AC Specifications* on page 3 for more information.

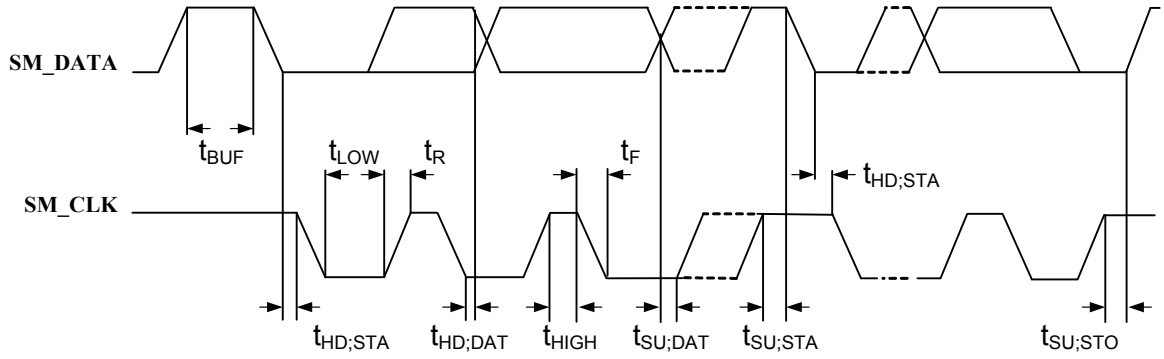

Figure 5.14 SMBus Slave Timing Diagram

SYMBOL	PARAMETER	MIN	MAX	UNIT
f_{SCL}	SM_CLK clock frequency	0	100	KHz
$t_{HD;STA}$	Hold time START condition	4	-	µs
t_{LOW}	LOW period of the SM_CLK clock	4.7	-	µs
t_{HIGH}	HIGH period of the SM_CLK clock	4	-	µs
$t_{SU;STA}$	Set-up time for a repeated START condition	4.7	-	µs
$t_{HD;DAT}$	DATA hold time\	0	-	ns
$t_{SU;DAT}$	DATA set-up time	250	-	ns
t_R	Rise time of both SM_DATA and SM_CLK signals	-	1000	ns
t_F	Fall time of both SM_CLK and SM_DATA lines	-	300	ns
$t_{SU;STO}$	Set-up time for a STOP condition	4	-	µs
t_{BUF}	Bus free time between a STOP and START condition	4.7	-	µs

Table 5.2 SMBus Slave Timing Modes

5.3 Reset

There are two different resets that the device experiences. One is a hardware reset (either from the internal POR reset circuit or via the **RESET_N** pin) and the second is a USB Bus Reset.

5.3.1 Internal POR

All reset timing parameters are guaranteed by design.

5.3.2 External Hardware Reset

A valid hardware reset is defined as assertion of **RESET_N** for a minimum of 1 μs after all power supplies are within operating range. While reset is asserted, the device (and its associated external circuitry) consumes less than 500 μA of current from the upstream USB power source.

Assertion of **RESET_N** (external pin) causes the following:

1. The PHY is disabled, and the differential pairs will be in a high-impedance state.
2. All transactions immediately terminate; no states are saved.
3. All internal registers return to the default state.
4. The external crystal oscillator is halted.
5. The PLL is halted.