

CC3100 SPI Host Interface

Contents

Introduction

Abbreviations

SPI Modes

SPI Configurations

Shared SPI

3-Wires SPI

Summarization

Host Interface protocol - SPI perspective

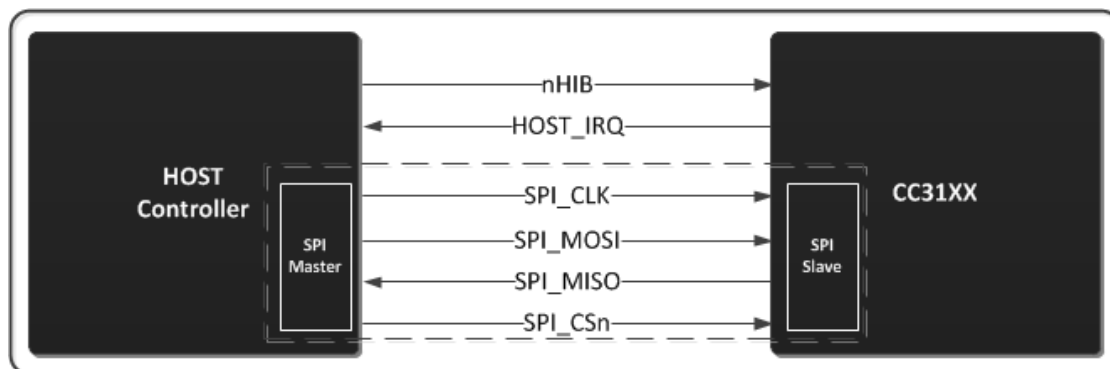
Initialization flow

Errata

Introduction

The purpose of this page is to describe the details of SimpleLink™ CC3100™ SPI (Serial Peripheral Interface) host interface. This page provides complementary information to the DS. It covers the main properties of the host interface protocol including supported modes, structure of different commands and communication flow. The page also provides guidelines for SPI configuration at the host side. It should be used by programmers during early integration stages.

The SPI bus is typically comprised of four lines, plus one interrupt line from the device to the host controller:



Basic SPI Configuration

Notice that the CC3100 has two SPI interfaces. One for the auxiliary serial flash device (that bus is labeled FLASH_SPI), and one for the host interface (labeled HOST_SPI). This document only refers to the latter.

Abbreviations

- **MISO** - Master In Slave Out SPI line
- **MOSI** - Master Out Slave In SPI line
- **SYNC** - Synchronization word

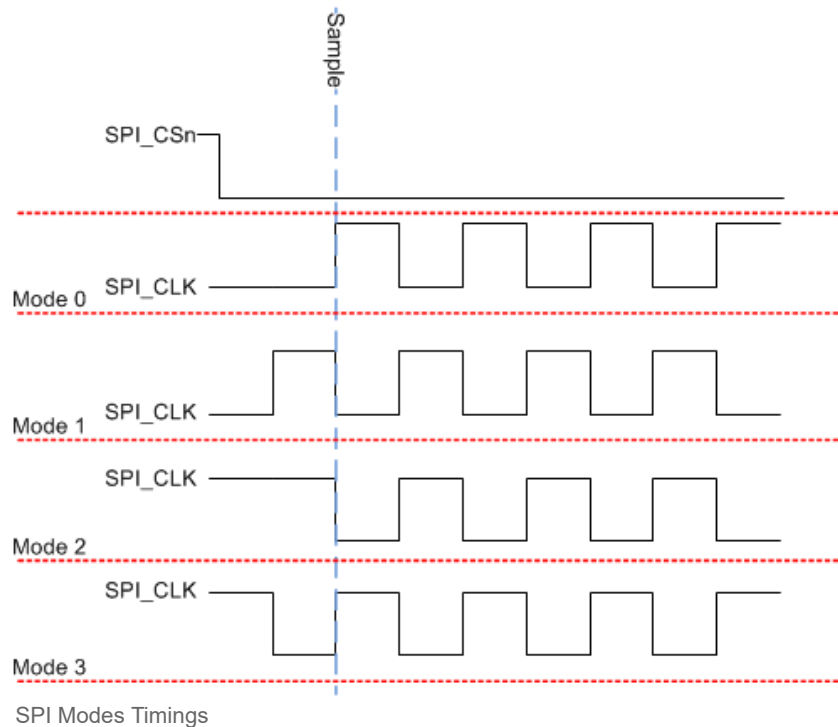
SPI Modes

There are four modes of operation defined in the SPI standard. For communication to be successful, the master and slave devices must be configured in the same way.

The four SPI modes are:

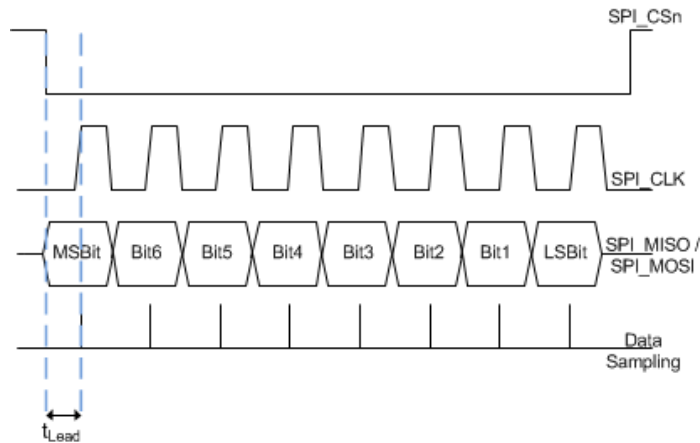
Mode	Polarity	Phase	Description
0	0	0	SPI_CLK is active high and sample commences on the rising edge
1	0	1	SPI_CLK is active high and sample commences on the falling edge
2	1	0	SPI_CLK is active low and sample commences on the rising edge
3	1	1	SPI_CLK is active low and sample commences on the falling edge

The following diagram shows the four different timing configurations:



CC3100 is working in **mode 0**. This means that data is sampled on the rising edge of the clock and changed on the falling edge of the clock. It is important to note that the first bit of each word has to be output by the master at least half a clock cycle prior to the first clock edge.

A single 1 byte transaction in mode 0 is described in the following drawing:



SPI Mode 0 Detailed Timings

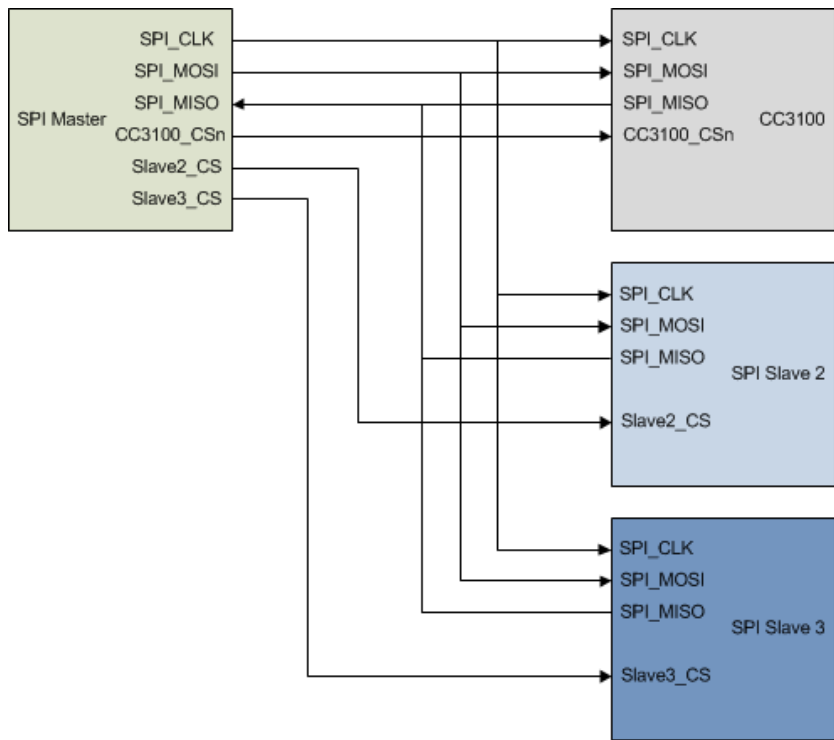
t_{Lead} should be at least half a clock cycle long, and so its value depends on the host clock frequency.

SPI Configurations

Shared SPI

The CC3100 may share the SPI bus with other slaves, all connected to a single master. In this case, the CLK, MOSI and MISO lines will be shared with the other slaves, and the CC3100 will have its own CS_n to signal which messages are directed to the CC3100. Please note that the MISO line goes into HiZ state in between words the CC3100 is transmitting to avoid possible contention with the other slaves.

The following drawing shows a typical multi-slave configuration:



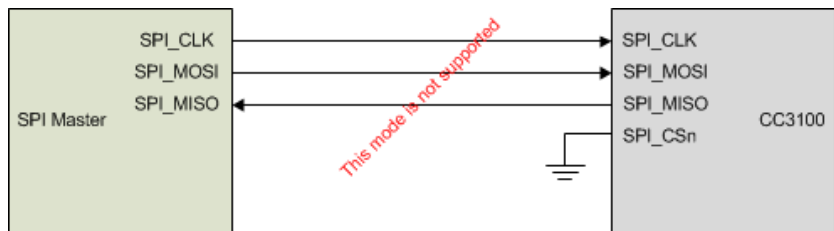
Shared SPI Block Diagram

Note that the other slaves must also have their MISO lines tri-stated in between data writing cycles to prevent the risk of line contention or data corruption.

3-Wires SPI

When a single SPI slave configuration is used (not shared SPI mode), and in the case of a pin limited platform, the user may want to eliminate the CS line and tie it to GND. This option is **NOT** supported by the current CC3100 revision.

The following drawing shows a typical 3-wires SPI connection:



3-Wires Configuration Block Diagram

Summarization

The following table summarizes the different supported configurations:

Property	Supported CC3100 Configuration
Clock polarity	Data is output on the clock's falling edge, sampled on the rising edge
Clock phase	Clock idles at logical 0
Word size	32/16/8 bits
Host Endianity	Little Endian / Big Endian
Bit order	MSBit first
Chip select polarity	Active low
Host Interrupt polarity	Active high
Host Interrupt mode	Rising edge or level '1'
Clock Frequency	Up to 20MHz
Chip select assertion between words	Optional (CSn can be kept asserted for entire message)
3-Wires mode	Not supported
Shared SPI	Supported

Note that the Host Endianity and word size are automatically detected by the SimpleLink™ device with no special configuration or handling in the Host application.

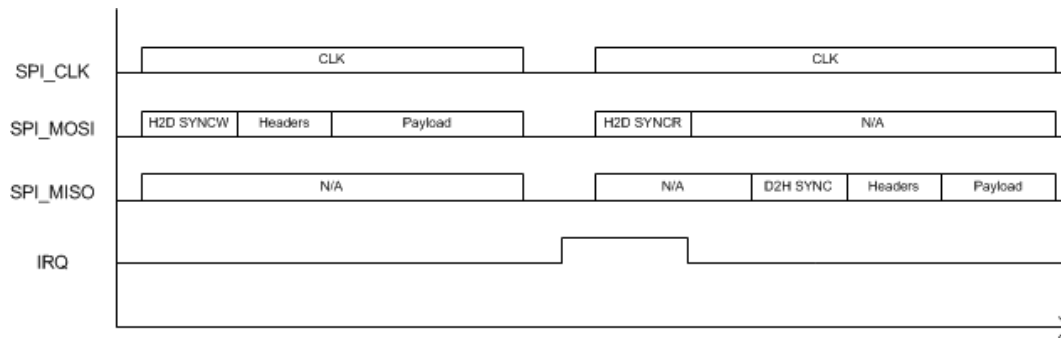
For exact timing requirements, please refer to the CC3100 datasheet.

Host Interface protocol - SPI perspective

As described on [Message Types in CC31xx Host Interface](#), the communication between the host and the CC3100 device is comprised of several types of messages:

1. Command
2. Command complete
3. Data
4. Asynchronous events

The following diagram describes the flow of a command from the host to the device along with the command complete indication from the device to the host (covering 1 & 2 in the list above):



CC3100 SPI Host Command Flow

As seen in the drawing, when the host writes to the device, the data from the device on the MISO line should be disregarded by the host, and vice versa – when the host reads from the device, the data on the MOSI line is disregarded by the CC3100 device. Having said that, it is still a good practice to keep the data to the device as all 0xFFs when reading from it.

The communication starts with the host sending the write SYNC word (for information regarding the SYNC words, see [Synchronization words in CC31xx Host Interface](#)), followed by header information, and then payload, when applicable. Once the command has been analyzed by the device, it asserts the IRQ interrupt line. The host then writes the read SYNC word. The device will clear the interrupt line and prepare the response. The host then reads continuously until the D2H SYNC pattern is detected. All data until that point is discarded. The SYNC word is then followed by headers and then payload, when applicable.

A data write sequence (#3 in the list above) looks like a subset of a command sequence, consisting of the host write alone (data write does not get acknowledged by the device):

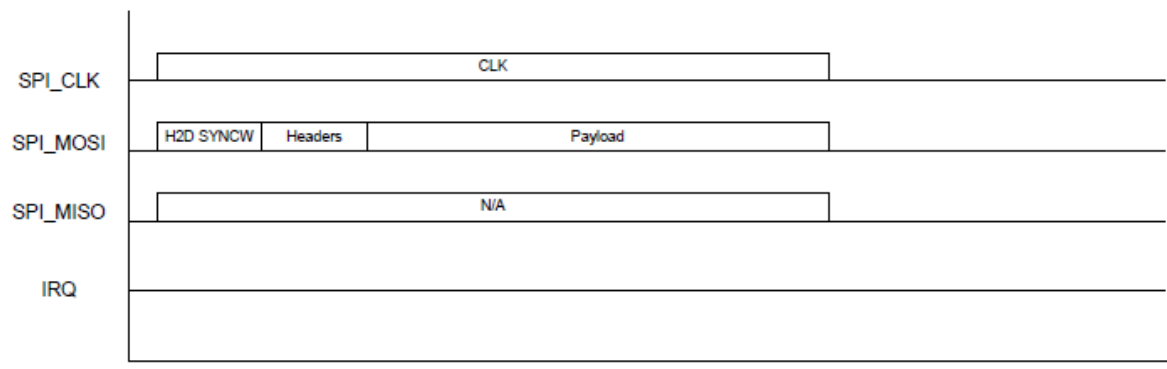


Figure 7 - CC3100 SPI Host Data TX Flow

CC3100 SPI Host Data TX Flow

An asynchronous event from the device to the host (item 4 in the list above) is also a subset of the command sequence, starting from an asynchronous interrupt from the device to the host:

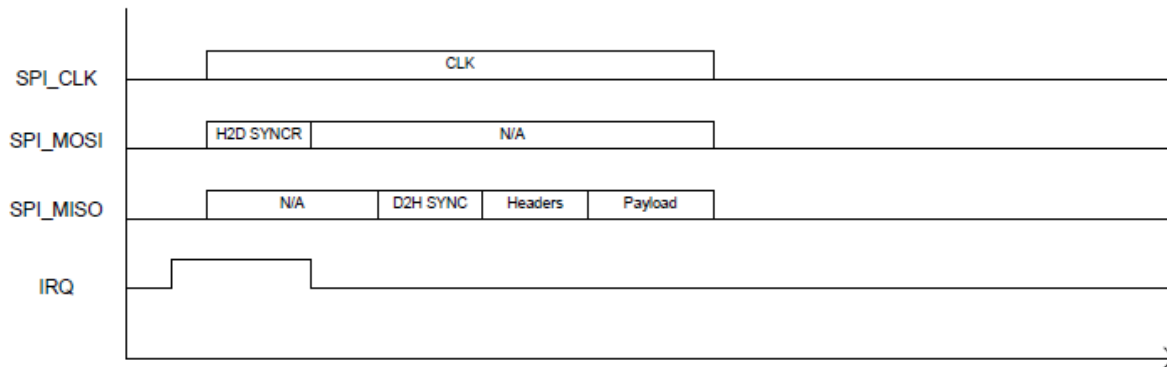
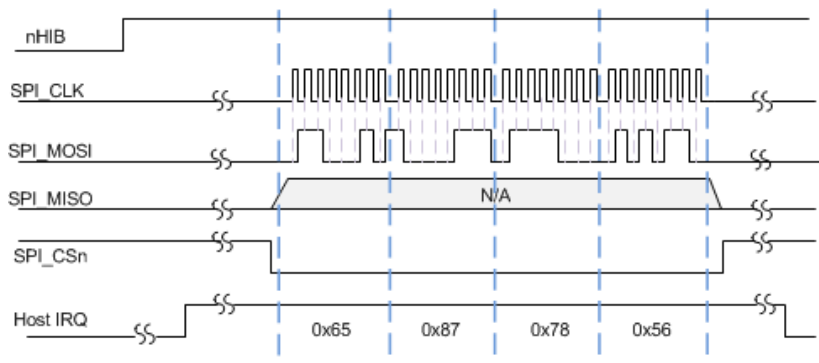


Figure 8 - CC3100 SPI Device Asynchronous Event Flow

CC3100 SPI Device Asynchronous Event Flow

Initialization flow

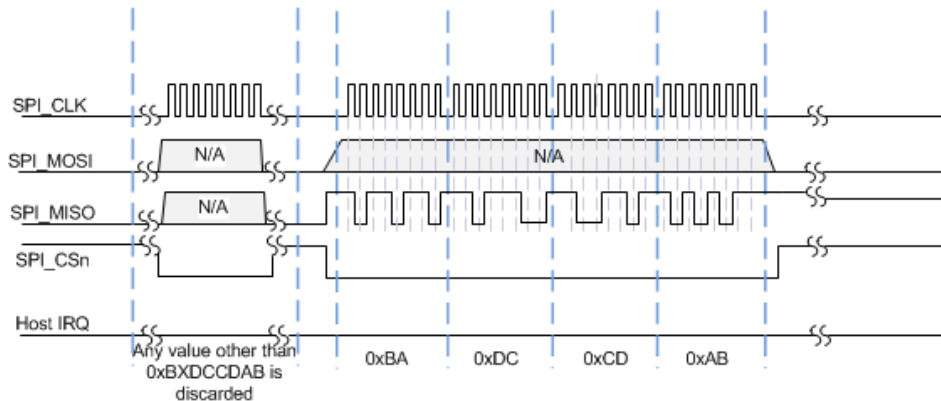
After the Host IRQ has been asserted for the first time, the host will write a sync word to the device. This will cause the host interrupt to prepare the message to the Host and clear the IRQ line. The process is described in the following drawing:



CC3100 Host to Device Synchronization word

Once the sync word has been written by the host, the device knows the host will now read the message that caused the IRQ to rise (in this case, it was initialization complete).

Since the host may precede the device's data readiness, there is another synchronization word that the host is looking for before parsing the response. All data before the data ready synchronization word is discarded by the driver. Please note that in some or even most cases, the first word read from the device will be the synchronization word, and no superfluous reads will be made.



CC3100 Device to Host Scynrhonization Word

If the host will send sync word for read while there is no message to send from the SimpleLink device to the Host, the SimpleLink device will send dummy message to avoid synchronization loss.

Errata

<p>1. switchcategory:MultiCore= <ul style="list-style-type: none"> For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum <p>Please post only comments related to the article CC3100 SPI Host Interface here.</p> </p>	<p>Keystone= <ul style="list-style-type: none"> For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum <p>Please post only comments related to the article CC3100 SPI Host Interface here.</p> </p>	<p>C2000= <i>For technical support on the C2000 please post your questions on The C2000 Forum. Please post only comments about the article CC3100 SPI Host Interface here.</i> </p>	<p>DaVinci= <i>For technical support on DaVinciplease post your questions on The DaVinci Forum. Please post only comments about the article CC3100 SPI Host Interface here.</i> </p>	<p>MSP430= <i>For technical support on MSP430 please post your questions on The MSP430 Forum. Please post only comments about the article CC3100 SPI Host Interface here.</i> </p>	<p>OMAP35x= <i>For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article CC3100 SPI Host Interface here.</i> </p>	<p>OMAPL1= <i>For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article CC3100 SPI Host Interface here.</i> </p>	<p>MAVRK= <i>For technical support on MAVRK please post your questions on The MAVRK Toolbox Forum. Please post only comments about the article CC3100 SPI Host Interface here.</i> </p> <p><i>For technical support please post your questions at http://e2e.ti.com. Please post only comments about the article CC3100 SPI Host Interface here.</i></p>
---	--	--	---	---	--	---	---

Links

[Processors](#)



[Amplifiers & Linear](#)

[Audio](#)

[Broadband RF/IF & Digital Radio](#)

[Clocks & Timers](#)

[Data Converters](#)

[DLP & MEMS](#)

[High-Reliability](#)

[Interface](#)

[Logic](#)

[Power Management](#)

▪ [ARM Processors](#)

▪ [Digital Signal Processors \(DSP\)](#)

▪ [Microcontrollers \(MCU\)](#)

▪ [OMAP Applications Processors](#)

[Switches & Multiplexers](#)

[Temperature Sensors & Control ICs](#)

[Wireless Connectivity](#)

Retrieved from "https://processors.wiki.ti.com/index.php?title=CC3100_SPI_Host_Interface&oldid=227430"

This page was last edited on 9 May 2017, at 15:16.

Content is available under [Creative Commons Attribution-ShareAlike](#) unless otherwise noted.