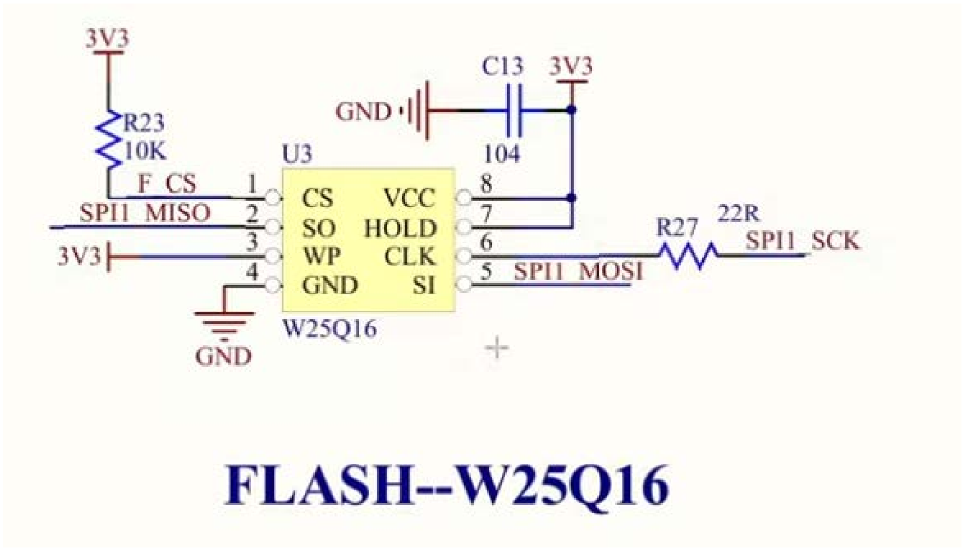
ABOUT THE SPI PROTOCOL AND THE W25Q128 FLASH MEMORY

SOURCE OF INFORMATION: https://www.youtube.com/watch?v=OSfu4ST3dlY

INTRODUCTION:

The following image is an extraction from the datasheet, which is shown in this document as a reference of the W23Q Flash Memory that has 16 Megabytes in order to use it as a reference for how to electrically wire it to our MCU:

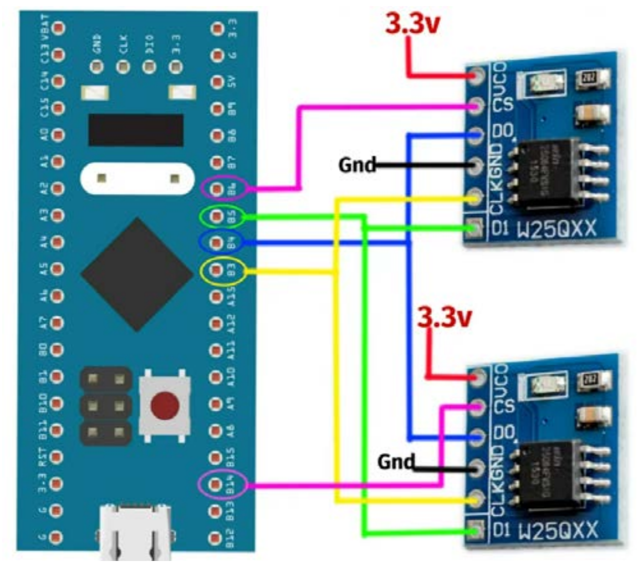


These W23Q Flash Memories use the either the Dual/Quad SPI Protocol or the Standard SPI Protocol for establishing communication between the desired Flash Memory and whatever external device we are using with it (e.g., an MCU). There are several versions of these W23Q Flash Memories, where they differ on the total amount of Flash Memory Size and where they all work similarly at the base level. However, the most practical aspect of these Flash Memories is that their Memory is distributed the same way in all of them at the lower level.

Note that in this project, the Standard SPI Protocol will be used.

HOW TO CONNECT A W25Q FLASH MEMORY TO OUR MCU

The following image illustrates how to connect a W23Q Flash Memory to a MCU:



Where this reference image I picked from the internet that conceptually shows how to connect several devices to the same SPI lines. It is important to note that each device can be independently selected by using their corresponding Chip Select Pin (i.e., the CS Pin). However, please have in mind that at least in my case, with the STM32F103C8T6 Blue Pill MCU, which looks alike to the MCU Blue Pill of that image, I was not able to make the SPI work with those lines which are supposedly enabled with the Alternate Pin function of the STMicroelectronics MCU/MPU devices (which is a functionality in the peripheral configurations of those devices where you can use transfer specific peripherals in your MCU/MPU to other pins). At least with that particular STM32F103C8T6 Blue Pill MCU that I used for this project, the SPI lines that actually worked were the ones that are activated as default and whenever I used the ones with the Alternate Pin function, it did not work for me.

As a reference, know that DI stands for Data In and that DO stands for Data Out.

SOME GENERAL ASPECTS TO KNOW ABOUT THE W25Q128 FLASH MEMORY THAT WILL BE USED IN THIS PROJECT

- The W25Q128 Flash Memory has a Flash Memory array organized into 65’356 programmable pages of 256 bytes each.
- Up to 256 bytes can be programmed at a time.
- The W25Q128 Flash Memory has a Flash Memory has a total size of 128 Mbits (i.e., 16Mbytes).
- Pages can be erased in groups of 16 pages (i.e., 4KB sector erase), groups of 128 pages (i.e., 32KB block erase), groups of 256 pages (i.e., 64KB block erase) or the entire chip (i.e., chip erase).
- The W25Q128 Flash Memory has 4’096 erasable sectors and 256 erasable blocks respectively.
- The small 4KB sectors allow for greater flexibility in applications that require data and parameter storage.
- SPI clock frequencies of up to 104MHz are supported, where it is also allowed to have equivalent clock rates of 208MHz (i.e., 104MHz x 2) for Dual I/O and 416MHz (i.e., 104MHz x 4) for Quad I/O when using the Fast Read Dual/Quad I/O and QPI instructions respectively.
- The Continuous Read Mode allows for accessing memory with as few as 8 clocks of instruction overhead to read a 24-bit address, where this reading mode can be made for 8, 16, 32 or 64 bytes in a Wrap.

The following image illustrates the Block Diagram of a W25Q128 Flash Memory:

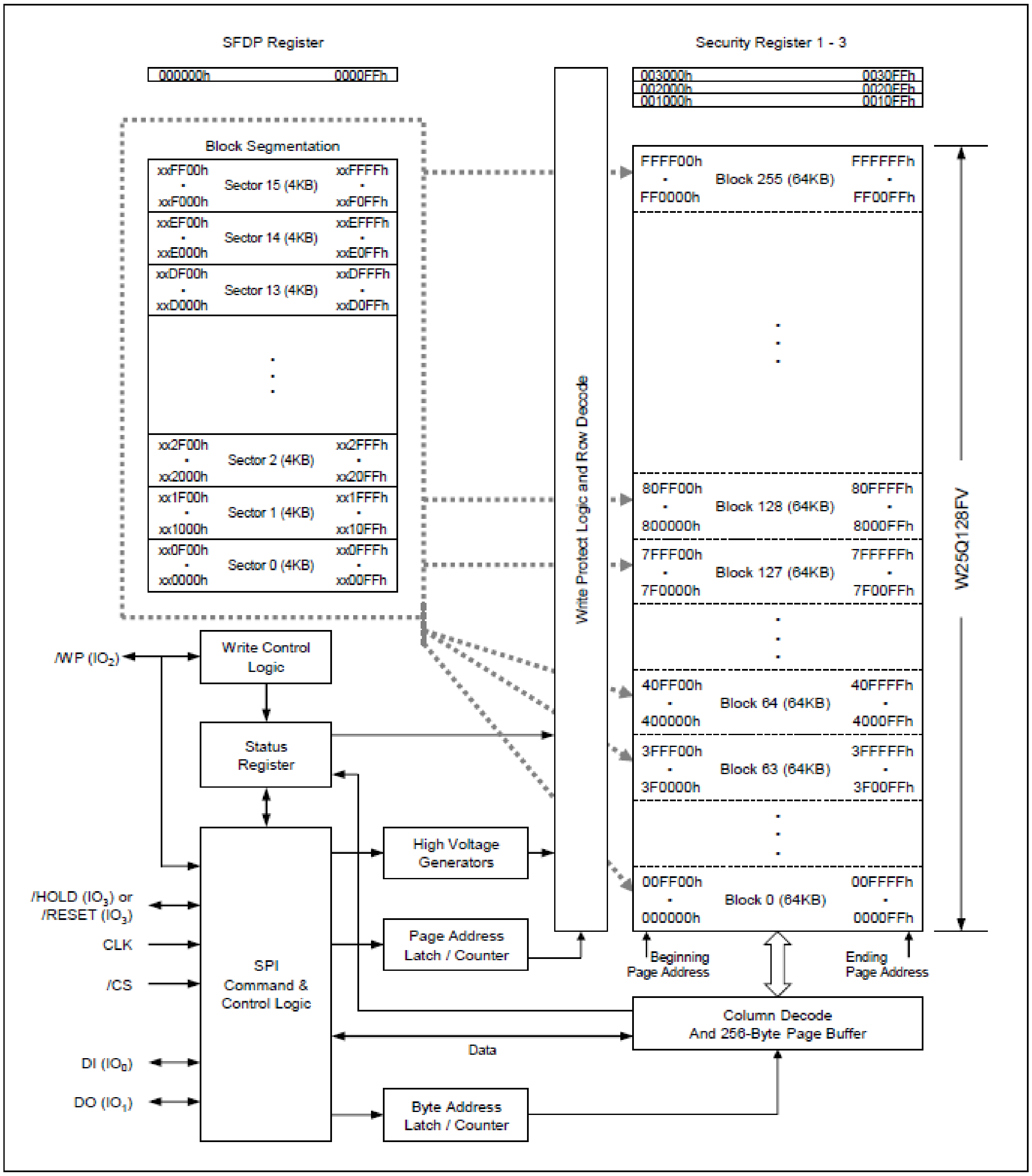
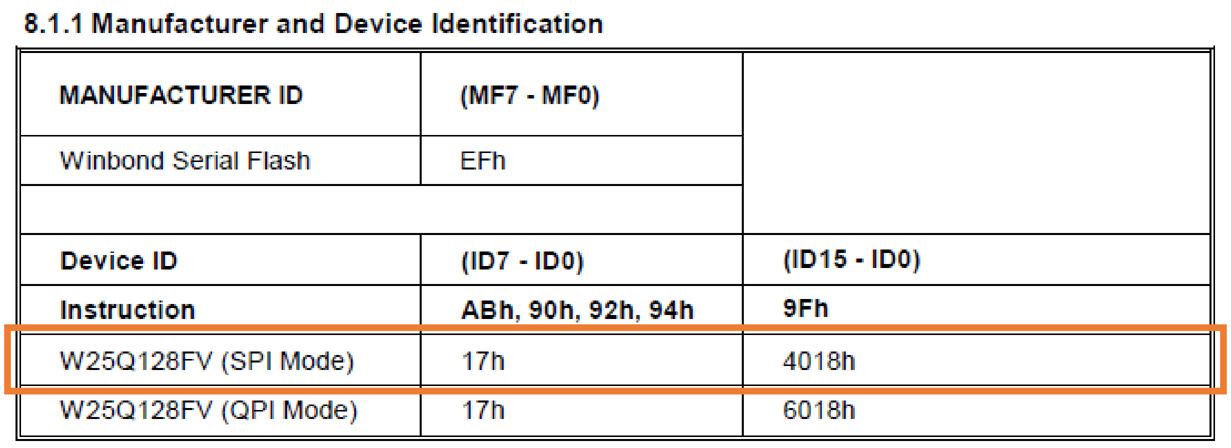


Figure 2. W25Q128FV Serial Flash Memory Block Diagram

INFORMATION TO KNOW BEFORE READING THE ID OF A W25Q FLASH MEMORY FROM OUR MCU

A necessary thing to do whenever communicating with a W25Q Flash Memory is to learn its corresponding ID, which is unique to each product series.

In particular for the W25Q128FV Flash Memory in Standard SPI Protocol, you can check its ID under the following instruction section:



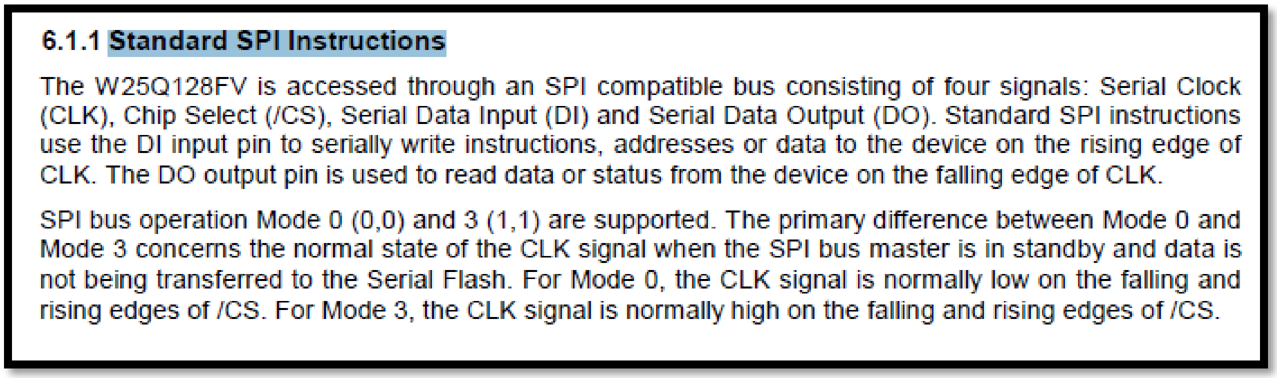
8.1.1 Manufacturer and Device Identification

MANUFACTURER ID	**(MF7 - MF0)**	
Winbond Serial Flash	EFh	
Device ID	**(ID7 - ID0)**	**(ID15 - ID0)**
Instruction	**ABh, 90h, 92h, 94h**	**9Fh**
W25Q128FV (SPI Mode)	17h	4018h
W25Q128FV (QPI Mode)	17h	6018h

CONFIGURING YOUR STM32 MCU FOR ESTABLISHING COMMUNICATION WITH A W25Q FLASH MEMORY

Although the instructor, from the Tutorial from which the information of this document has been based from, has suggested to set a prescaler such that the Baud Rate is around 2.5 Megabits per second, since I am using a different MCU than this person and because I desire to configure it with a Clock Frequency of 2MHz (this is because in many applications where I used the MCU that I used, which is the STM32F103C8T6, I want to set a low Clock Frequency to attempt to lower the energy consumption of the device as much as possible, but that is also something with which most general applications can work with), I will be using 1Mbit/sec Baud Rate instead.

Also, according to the following extraction of information from the W25Q128FV datasheet:



6.1.1 Standard SPI Instructions

The W25Q128FV is accessed through an SPI compatible bus consisting of four signals: Serial Clock (CLK), Chip Select (/CS), Serial Data Input (DI) and Serial Data Output (DO). Standard SPI instructions use the DI input pin to serially write instructions, addresses or data to the device on the rising edge of CLK. The DO output pin is used to read data or status from the device on the falling edge of CLK.

SPI bus operation Mode 0 (0,0) and 3 (1,1) are supported. The primary difference between Mode 0 and Mode 3 concerns the normal state of the CLK signal when the SPI bus master is in standby and data is not being transferred to the Serial Flash. For Mode 0, the CLK signal is normally low on the falling and rising edges of /CS. For Mode 3, the CLK signal is normally high on the falling and rising edges of /CS.

This type of Flash Memory may work with either the SPI Mode 0 or SPI Mode 3, where basically the only difference between them is the polarity that the Flash Memory will consider with respect to the rising/falling edges of the Clock signal (read the extraction of information shown for more details on this).

Because of this information, it is now known that we can configure the SPI of our MCU to have a Clock Polarity of either Low or High Type. So, in order to follow the same configuration as the instructor from the tutorial from which all the information of this document has been based on, the Low Clock Polarity will be configured in the SPI of our MCU.

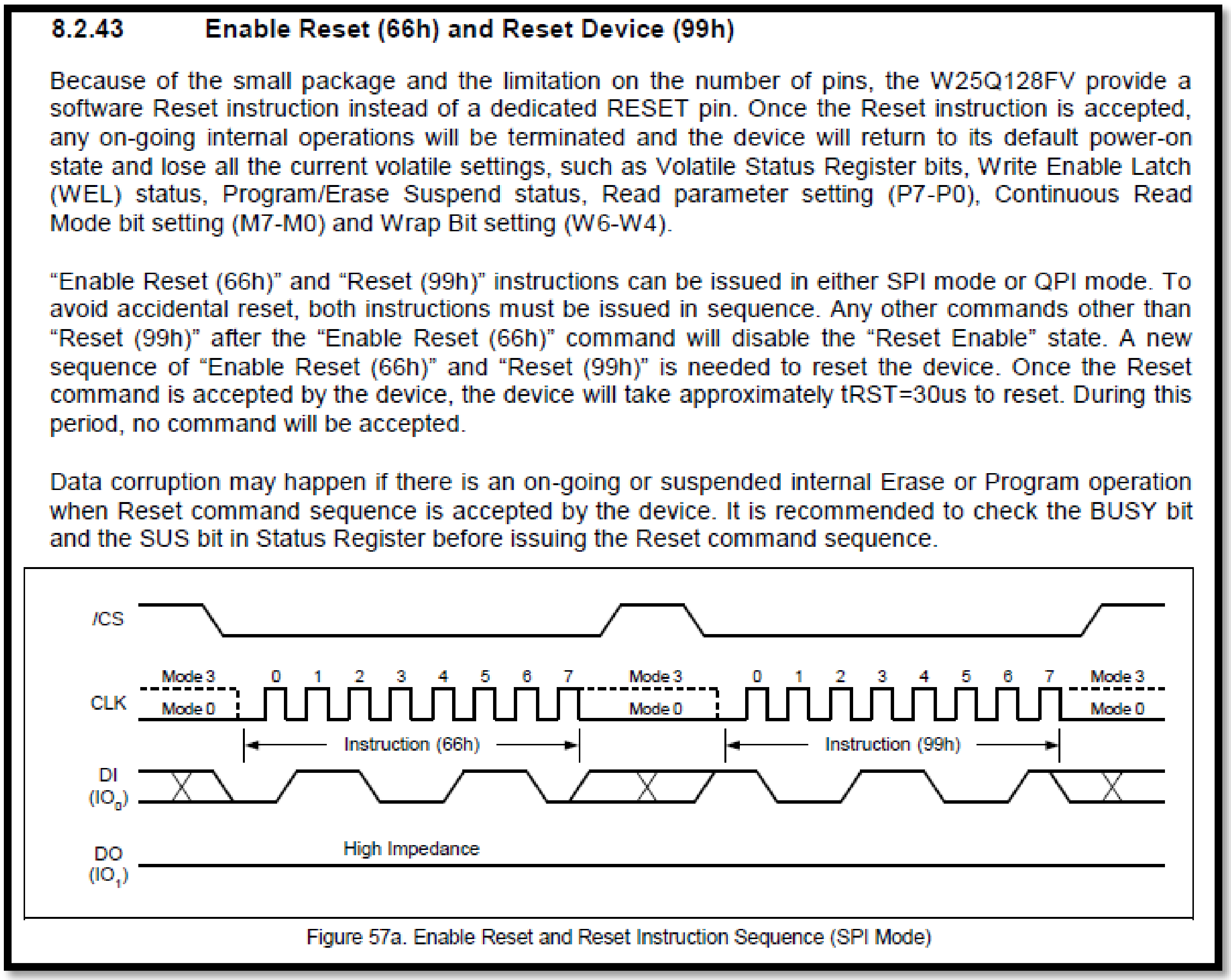
Also, our MCU will be configured with a Clock Phase of 1 Edge, which means that the data will be sampled on the first clock edge, which should correspond to the rising edge. In addition, the Clock Polarity configured will be Low. As a result, since Clock Phase has a value of 1 Edge and since the Clock Polarity has a value of Low, then this means that our MCU will be configured to work with Mode 0 Standard SPI.

Another aspect to take into account is that whatever GPIO Output Pin you configure in your STM32 MCU for controlling the Chip Select Pin of the W25 Flash Memory you are programming, is to configure the initial GPIO Output Level of that MCU's GPIO Output Pin to be High initially so that the W25 Flash Memory is disabled right after our MCU boots. This way, we can enable that Flash Memory whenever we desire by simply changing the State of that GPIO Output Pin to be from High to Low.

One last thing to consider is that the instructor, of the tutorial from which the information of this document was based on, configured the Chip Select GPIO Output Pin, that was just mentioned in the previous paragraph, to work with the highest possible Maximum Output Speed. However, I will try it with the Lowest one since I am more interested in being able to use this Flash Memory in Low Power Applications (the lower the frequency used in the peripherals, the lower the energy consumption in our MCU).

HOW TO RESET THE W25Q128FV FLASH MEMORY PRIOR TO START USING IT

The following image is an extraction from the W25Q128FV Datasheet that contains the information that needs to be known for sending the Reset Command to a W25Q128FV Flash Memory, which can be made by sending the Enable Reset and then the Reset Device commands in that order, as explained there:



8.2.43 Enable Reset (66h) and Reset Device (99h)

Because of the small package and the limitation on the number of pins, the W25Q128FV provide a software Reset instruction instead of a dedicated RESET pin. Once the Reset instruction is accepted, any on-going internal operations will be terminated and the device will return to its default power-on state and lose all the current volatile settings, such as Volatile Status Register bits, Write Enable Latch (WEL) status, Program/Erase Suspend status, Read parameter setting (P7-P0), Continuous Read Mode bit setting (M7-M0) and Wrap Bit setting (W6-W4).

“Enable Reset (66h)” and “Reset (99h)” instructions can be issued in either SPI mode or QPI mode. To avoid accidental reset, both instructions must be issued in sequence. Any other commands other than “Reset (99h)” after the “Enable Reset (66h)” command will disable the “Reset Enable” state. A new sequence of “Enable Reset (66h)” and “Reset (99h)” is needed to reset the device. Once the Reset command is accepted by the device, the device will take approximately tRST=30us to reset. During this period, no command will be accepted.

Data corruption may happen if there is an on-going or suspended internal Erase or Program operation when Reset command sequence is accepted by the device. It is recommended to check the BUSY bit and the SUS bit in Status Register before issuing the Reset command sequence.

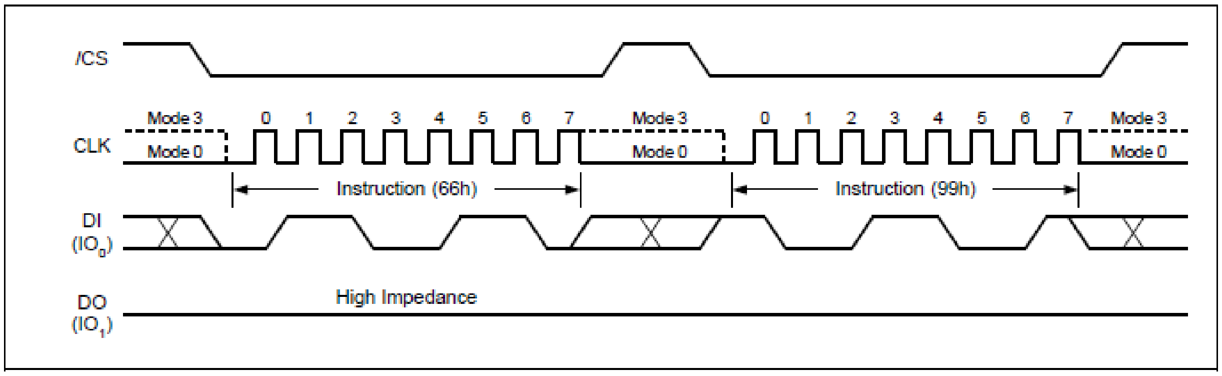


Figure 57a. Enable Reset and Reset Instruction Sequence (SPI Mode)

HOW TO READ THE MANUFACTURER DEVICE ID OF THE W25Q128FV FLASH MEMORY

There are several IDs that can be read from the W25Q128FV Flash Memory Device according to its datasheet, like the Manufacturer ID, the Device ID and the JEDEC ID. However, from among these, the JEDEC ID is more particular to a specific type of W25 Flash Memory Device (It will give the same ID for the same type of Flash Memories with the same Memory Size, but with that ID, you could tell the type of Flash Memory and its size as it was just mentioned). Therefore, the one that we will be reading for this project is the JEDED ID:

8.2.29 Read JEDEC ID (9Fh)

For compatibility reasons, the W25Q128FV provides several instructions to electronically determine the identity of the device. The Read JEDEC ID instruction is compatible with the JEDEC standard for SPI compatible serial memories that was adopted in 2003. The instruction is initiated by driving the /CS pin low and shifting the instruction code "9Fh". The JEDEC assigned Manufacturer ID byte for Winbond (EFh) and two Device ID bytes, Memory Type (ID15-ID8) and Capacity (ID7-ID0) are then shifted out on the falling edge of CLK with most significant bit (MSB) first as shown in Figure 43a & 43b. For memory type and capacity values refer to Manufacturer and Device Identification table.

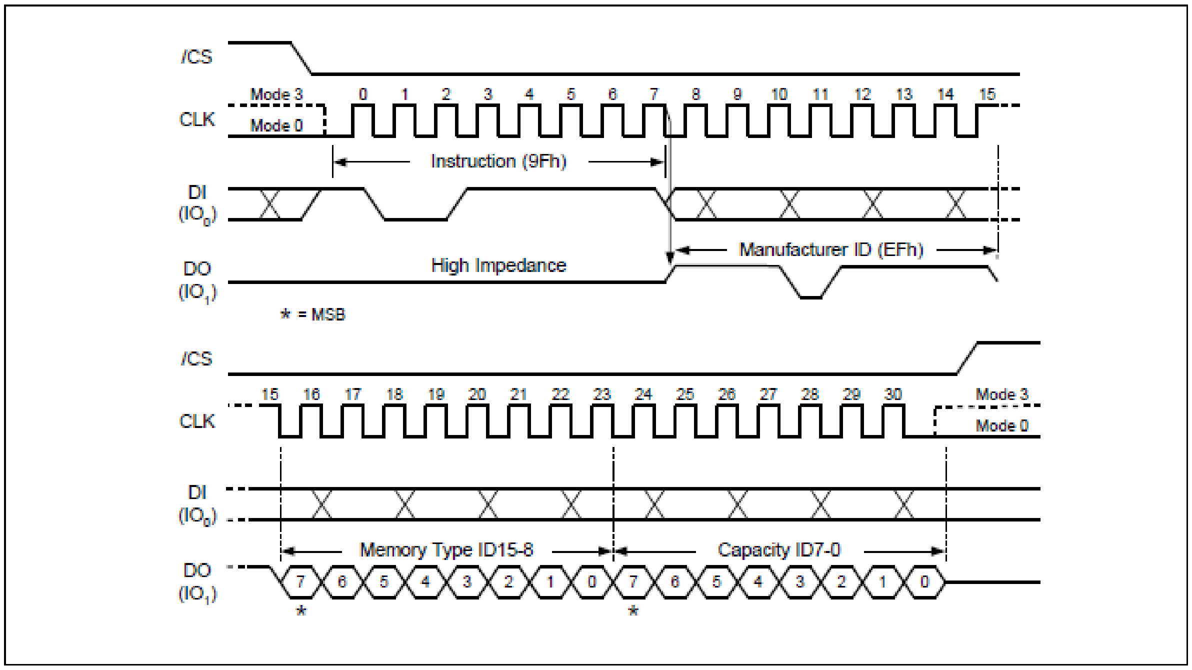


Figure 43a. Read JEDEC ID Instruction (SPI Mode)

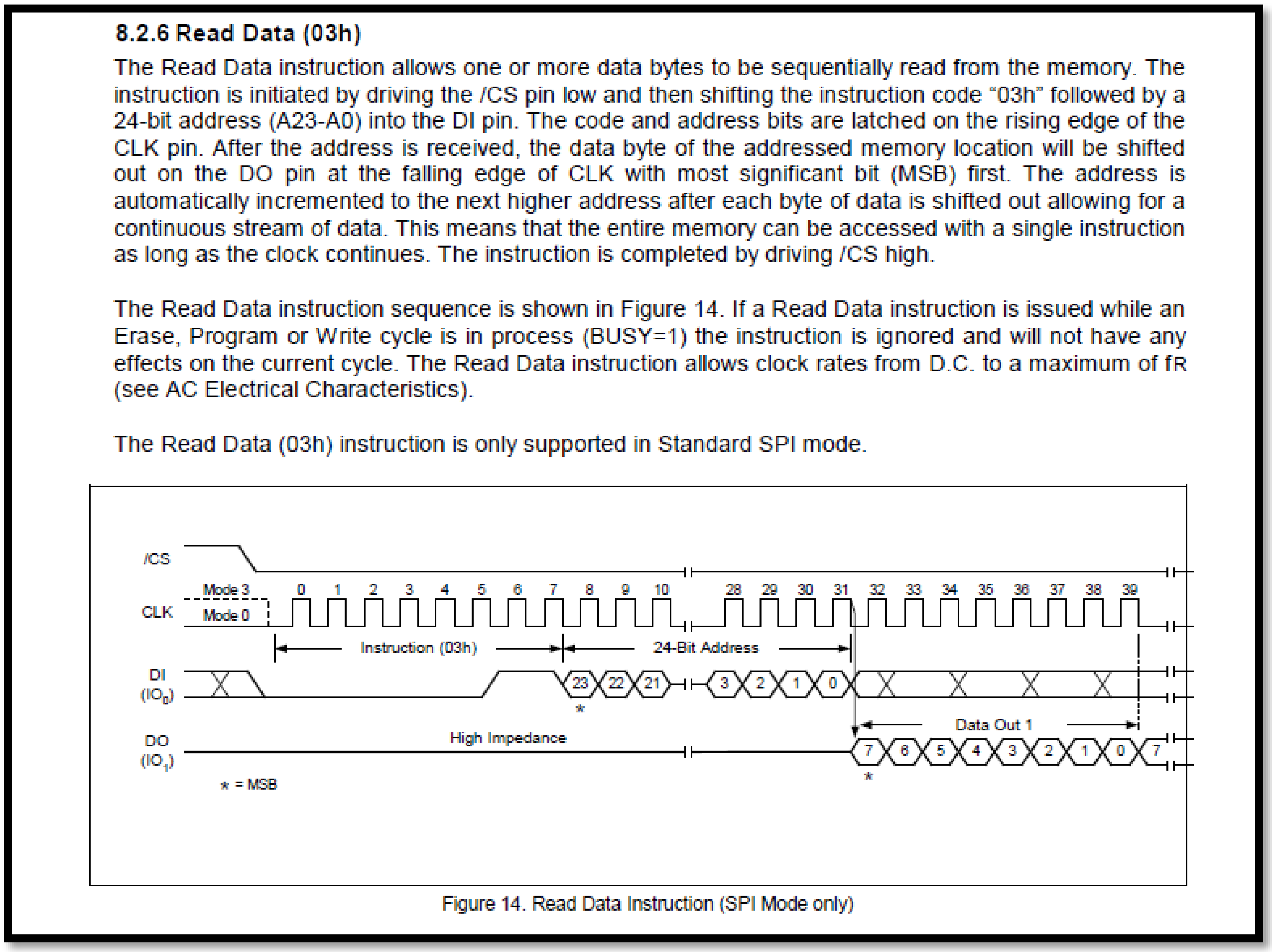
NOTE: *The expected response from a W25Q128FV Device after sending to it the Read JEDEC ID Instruction in Standard SPI Protocol, is to receive the value of 0x40 for the Memory Type and the value of 0x18 for the Memory Capacity, according to the following table extracted from its datasheet:*

8.1.1 Manufacturer and Device Identification

MANUFACTURER ID	**(MF7 - MF0)**	
Winbond Serial Flash	EFh	
Device ID	**(ID7 - ID0)**	**(ID15 - ID0)**
Instruction	**ABh, 90h, 92h, 94h**	**9Fh**
W25Q128FV (SPI Mode)	17h	4018h
W25Q128FV (QPI Mode)	17h	6018h

HOW TO READ DATA FROM THE W25Q128FV FLASH MEMORY

The following image is an extraction from the datasheet of the W25Q128FV Flash Memory about how to use the Read Data instruction:



8.2.6 Read Data (03h)

The Read Data instruction allows one or more data bytes to be sequentially read from the memory. The instruction is initiated by driving the /CS pin low and then shifting the instruction code “03h” followed by a 24-bit address (A23-A0) into the DI pin. The code and address bits are latched on the rising edge of the CLK pin. After the address is received, the data byte of the addressed memory location will be shifted out on the DO pin at the falling edge of CLK with most significant bit (MSB) first. The address is automatically incremented to the next higher address after each byte of data is shifted out allowing for a continuous stream of data. This means that the entire memory can be accessed with a single instruction as long as the clock continues. The instruction is completed by driving /CS high.

The Read Data instruction sequence is shown in Figure 14. If a Read Data instruction is issued while an Erase, Program or Write cycle is in process (BUSY=1) the instruction is ignored and will not have any effects on the current cycle. The Read Data instruction allows clock rates from D.C. to a maximum of fR (see AC Electrical Characteristics).

The Read Data (03h) instruction is only supported in Standard SPI mode.

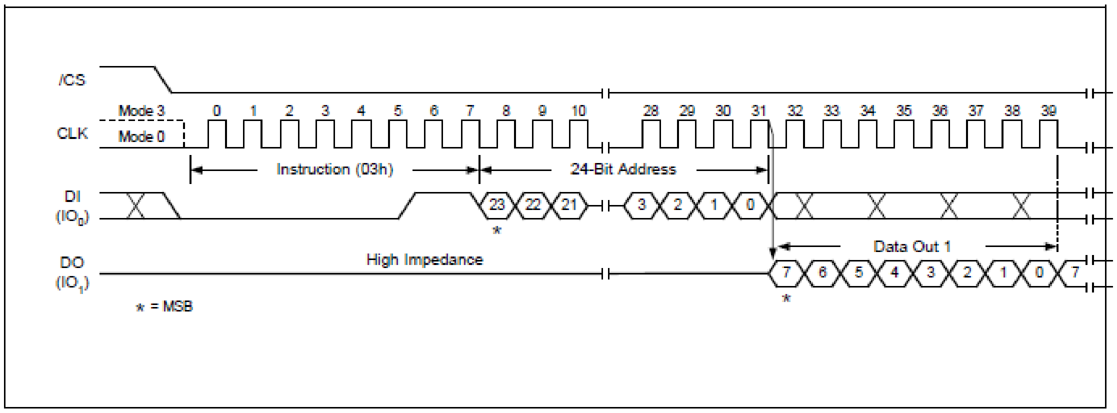


Figure 14. Read Data Instruction (SPI Mode only)

HOW TO FAST READ DATA FROM THE W25Q128FV FLASH MEMORY

The following image is an extraction from the datasheet of the W25Q128FV Flash Memory about how to use the Fast Read Data instruction:

8.2.7 Fast Read (0Bh)

The Fast Read instruction is similar to the Read Data instruction except that it can operate at the highest possible frequency of FR (see AC Electrical Characteristics). This is accomplished by adding eight "dummy" clocks after the 24-bit address as shown in Figure 16. The dummy clocks allow the devices internal circuits additional time for setting up the initial address. During the dummy clocks the data value on the DO pin is a "don't care".

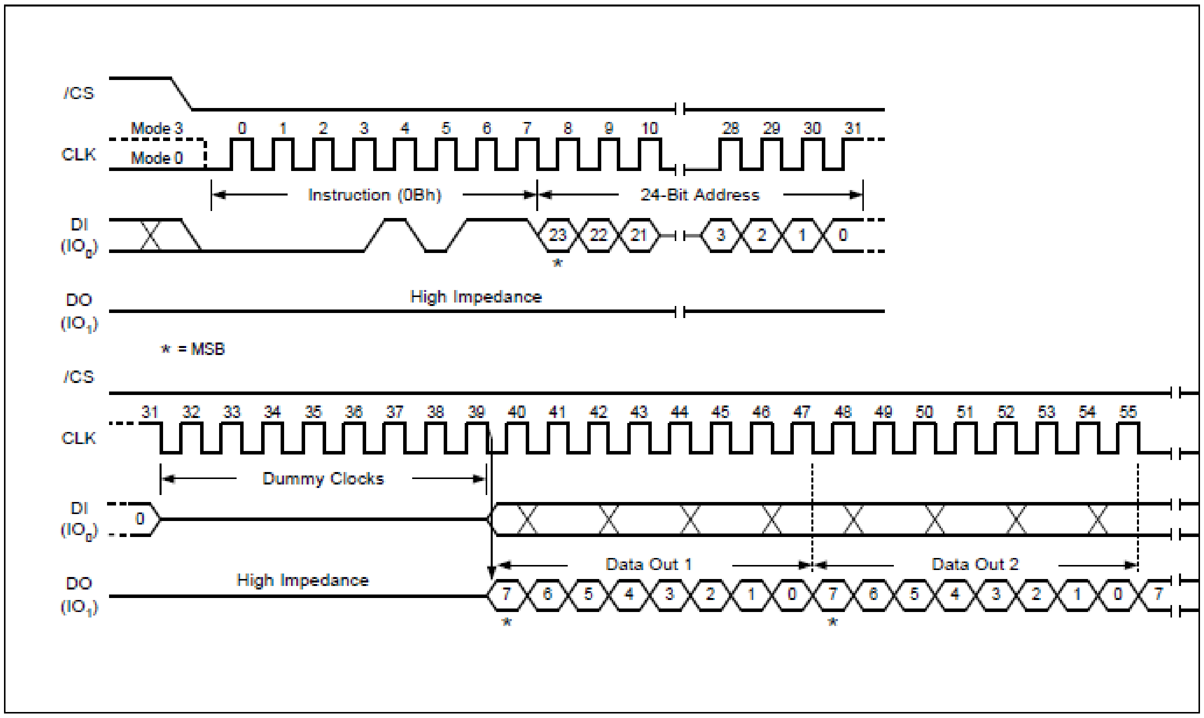


Figure 16a. Fast Read Instruction (SPI Mode)

HOW TO SECTOR ERASE (THE MINIMUM ERASABLE SIZE) IN THE W25Q128FV FLASH MEMORY

The following image is an extraction from the datasheet of the W25Q128FV Flash Memory about how to use the Sector Erase instruction:

8.2.17 Sector Erase (20h)

The Sector Erase instruction sets all memory within a specified sector (4K-bytes) to the erased state of all 1s (FFh). A Write Enable instruction must be executed before the device will accept the Sector Erase Instruction (Status Register bit WEL must equal 1). The instruction is initiated by driving the /CS pin low and shifting the instruction code "20h" followed a 24-bit sector address (A23-A0). The Sector Erase instruction sequence is shown in Figure 31a & 31b.

The /CS pin must be driven high after the eighth bit of the last byte has been latched. If this is not done the Sector Erase instruction will not be executed. After /CS is driven high, the self-timed Sector Erase instruction will commence for a time duration of tSE (See AC Characteristics). While the Sector Erase cycle is in progress, the Read Status Register instruction may still be accessed for checking the status of the BUSY bit. The BUSY bit is a 1 during the Sector Erase cycle and becomes a 0 when the cycle is finished and the device is ready to accept other instructions again. After the Sector Erase cycle has finished the Write Enable Latch (WEL) bit in the Status Register is cleared to 0. The Sector Erase instruction will not be executed if the addressed page is protected by the Block Protect (CMP, SEC, TB, BP2, BP1, and BP0) bits or the Individual Block/Sector Locks.

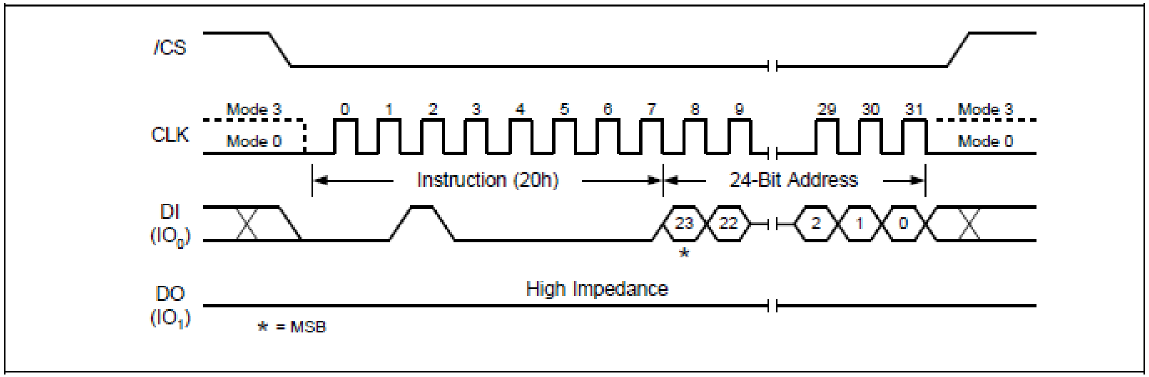


Figure 31a. Sector Erase Instruction (SPI Mode)

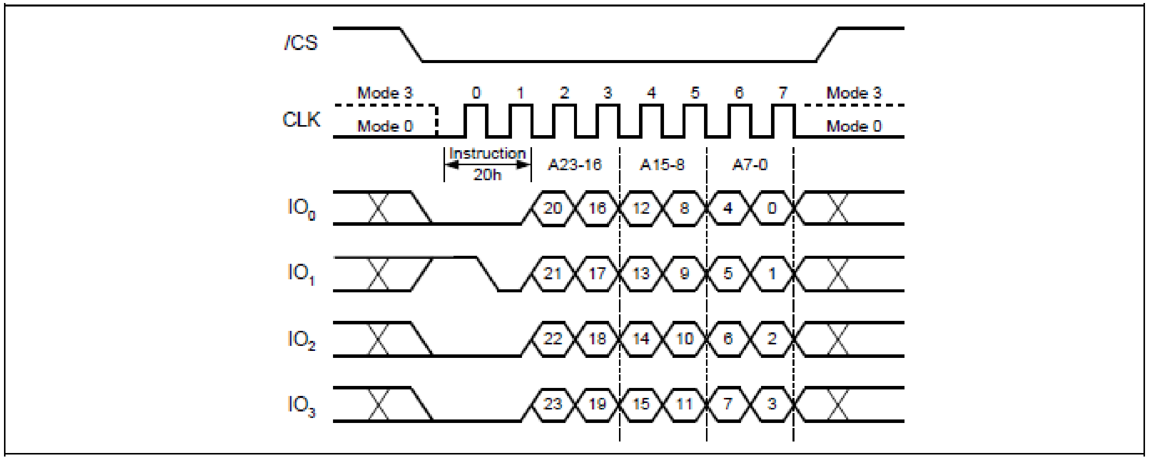
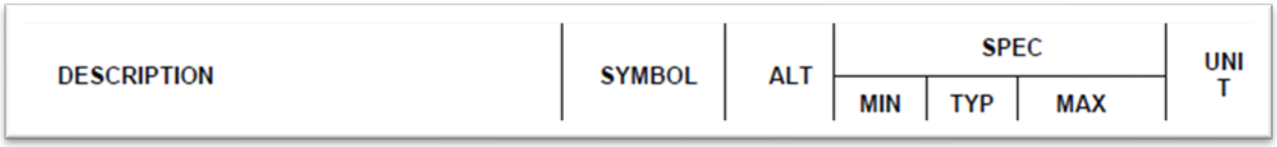
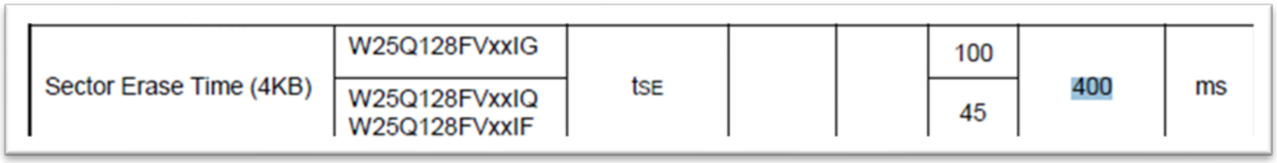


Figure 31b. Sector Erase Instruction (QPI Mode)

Where the datasheet states the following value for t_{SE}:

DESCRIPTION		SYMBOL	ALT	SPEC			UNIT
				MIN	TYP	MAX	
Sector Erase Time (4KB)	W25Q128FVxxIG	t_{SE}			100	400	ms
	W25Q128FVxxIQ W25Q128FVxxIF				45		

THE WRITE ENABLE INSTRUCTION AND THE WRITE DISABLE INSTRUCTION OF THE W25Q128FV FLASH MEMORY

The following image is an extraction from the datasheet of the W25Q128FV Flash Memory about how to use the Write Enable instruction:

8.2.1 Write Enable (06h)

The Write Enable instruction (Figure 5) sets the Write Enable Latch (WEL) bit in the Status Register to a 1. The WEL bit must be set prior to every Page Program, Quad Page Program, Sector Erase, Block Erase, Chip Erase, Write Status Register and Erase/Program Security Registers instruction. The Write Enable instruction is entered by driving /CS low, shifting the instruction code "06h" into the Data Input (DI) pin on the rising edge of CLK, and then driving /CS high.

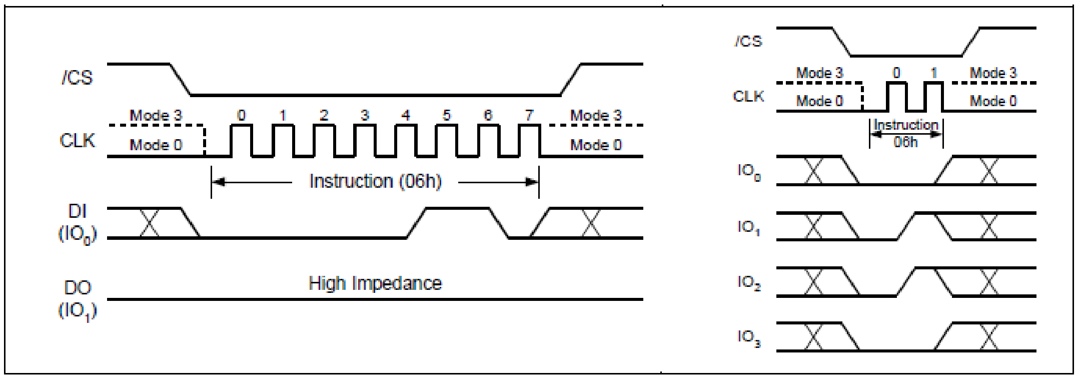


Figure 5. Write Enable Instruction for SPI Mode (left) or QPI Mode (right)

The following image is an extraction from the datasheet of the W25Q128FV Flash Memory about how to use the Write Disable instruction:

8.2.3 Write Disable (04h)

The Write Disable instruction (Figure 7) resets the Write Enable Latch (WEL) bit in the Status Register to a 0. The Write Disable instruction is entered by driving /CS low, shifting the instruction code "04h" into the DI pin and then driving /CS high. Note that the WEL bit is automatically reset after Power-up and upon completion of the Write Status Register, Erase/Program Security Registers, Page Program, Quad Page Program, Sector Erase, Block Erase, Chip Erase and Reset instructions.

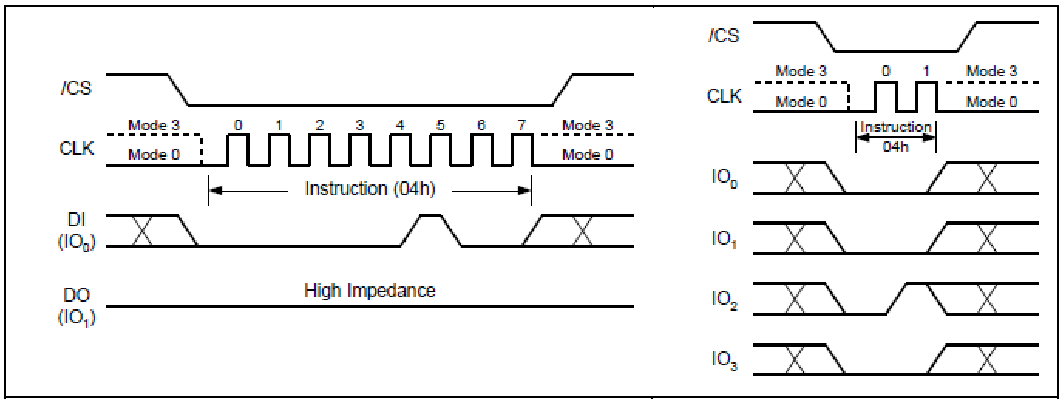
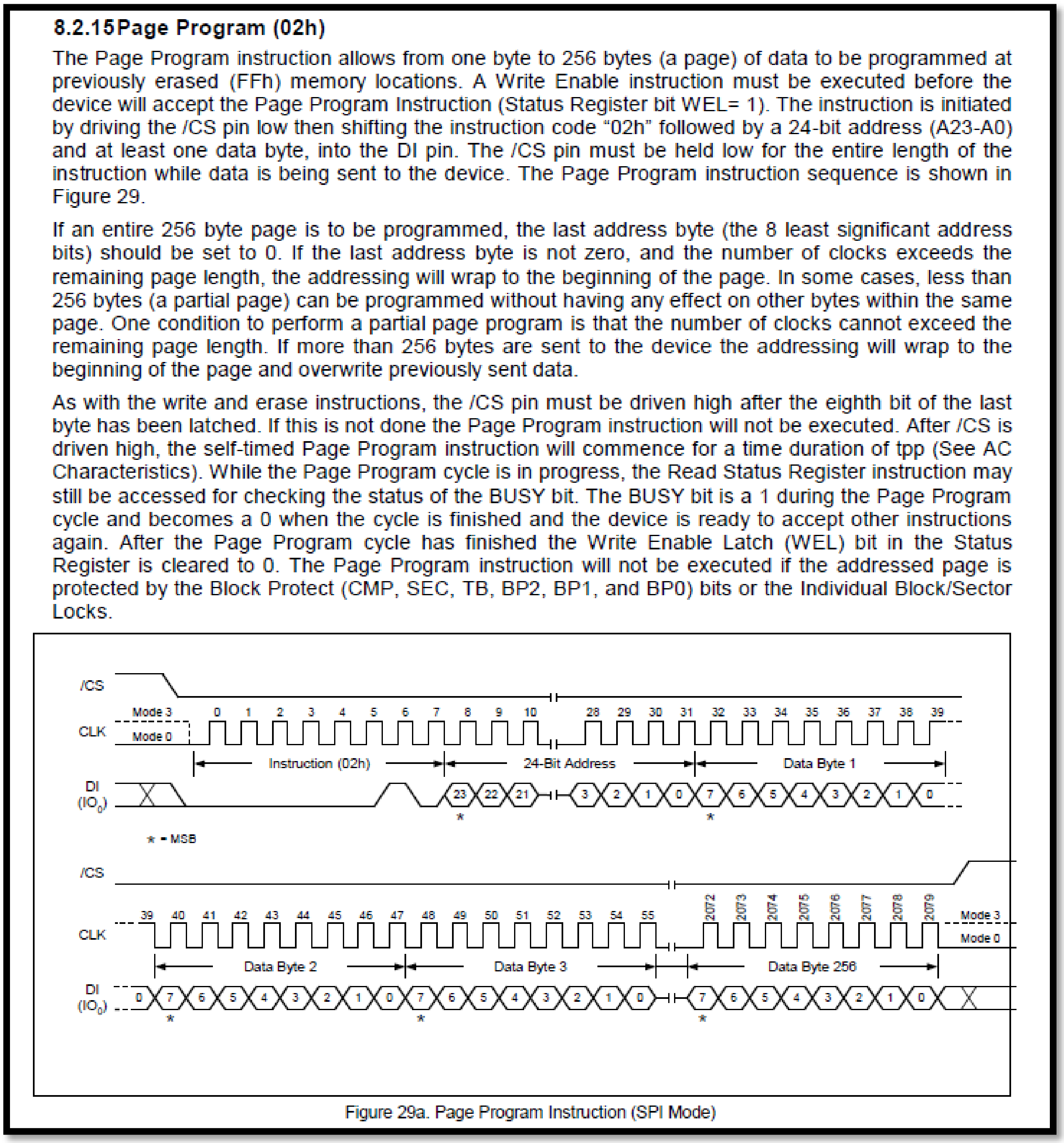


Figure 7. Write Disable Instruction for SPI Mode (left) or QPI Mode (right)

HOW TO WRITE FROM ONE TO 256 BYTES (I.E., A PAGE) IN THE W25Q128FV FLASH MEMORY

The following image is an extraction from the datasheet of the W25Q128FV Flash Memory about how to use the Page Program instruction:



8.2.15 Page Program (02h)

The Page Program instruction allows from one byte to 256 bytes (a page) of data to be programmed at previously erased (FFh) memory locations. A Write Enable instruction must be executed before the device will accept the Page Program Instruction (Status Register bit WEL= 1). The instruction is initiated by driving the /CS pin low then shifting the instruction code "02h" followed by a 24-bit address (A23-A0) and at least one data byte, into the DI pin. The /CS pin must be held low for the entire length of the instruction while data is being sent to the device. The Page Program instruction sequence is shown in Figure 29.

If an entire 256 byte page is to be programmed, the last address byte (the 8 least significant address bits) should be set to 0. If the last address byte is not zero, and the number of clocks exceeds the remaining page length, the addressing will wrap to the beginning of the page. In some cases, less than 256 bytes (a partial page) can be programmed without having any effect on other bytes within the same page. One condition to perform a partial page program is that the number of clocks cannot exceed the remaining page length. If more than 256 bytes are sent to the device the addressing will wrap to the beginning of the page and overwrite previously sent data.

As with the write and erase instructions, the /CS pin must be driven high after the eighth bit of the last byte has been latched. If this is not done the Page Program instruction will not be executed. After /CS is driven high, the self-timed Page Program instruction will commence for a time duration of tpp (See AC Characteristics). While the Page Program cycle is in progress, the Read Status Register instruction may still be accessed for checking the status of the BUSY bit. The BUSY bit is a 1 during the Page Program cycle and becomes a 0 when the cycle is finished and the device is ready to accept other instructions again. After the Page Program cycle has finished the Write Enable Latch (WEL) bit in the Status Register is cleared to 0. The Page Program instruction will not be executed if the addressed page is protected by the Block Protect (CMP, SEC, TB, BP2, BP1, and BP0) bits or the Individual Block/Sector Locks.

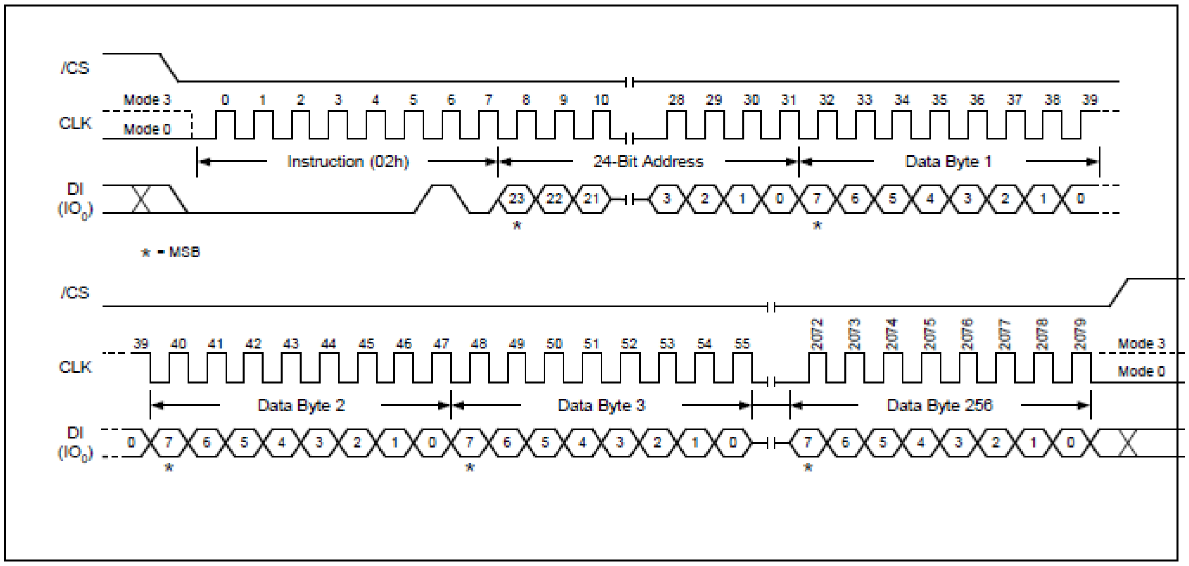


Figure 29a. Page Program Instruction (SPI Mode)

Where:

DESCRIPTION	SYMBOL	ALT	SPEC			UNI T
			MIN	TYP	MAX	
Page Program Time	t_{PP}			0.7	3	ms