AN304

SPI Guide for F-RAM™

Author: Harsha Medu
Associated Part Family: FM25xxx
Associated Code Examples: CE204087
Related Application Notes: click here

AN304 provides the functional description, timing, and example code for SPI F-RAMs.

Contents

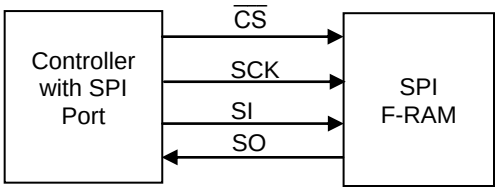
1 Introduction

The FM25xxx F-RAM product family employs an industry-standard 4-wire SPI interface. They are high-speed (up to 40 MHz), low-power, nonvolatile memory devices. SPI F-RAM densities start from 4 Kbit and extend up to 4 Mbit. This application note reviews the functional and timing aspects of these devices.

2 Why Use SPI?

The Serial Peripheral Interface (SPI) is a serial bus created by Motorola (now Freescale) and is provided as a dedicated interface on their MCUs and those from other semiconductor suppliers, such as Cypress, TI, Atmel, Microchip, Analog Devices, and more. SPI ports are also offered in DSPs, network processors, FPGAs, etc. For microcontroller-based systems that require high serial data rates, the SPI interface is an ideal choice. Serial data throughput correlates to the serial clock speed (SCK signal in Figure 1); most Cypress serial F-RAMs can be clocked at up to 40 MHz. Some microcontrollers do not have a dedicated SPI port, so the use of bit-banging provides a means to use GPIO pins for SPI operation. This method involves the software which controls or “bangs away” at the I/O port. Figure 1 shows the basic SPI interface.

Figure 1. Basic SPI Interface

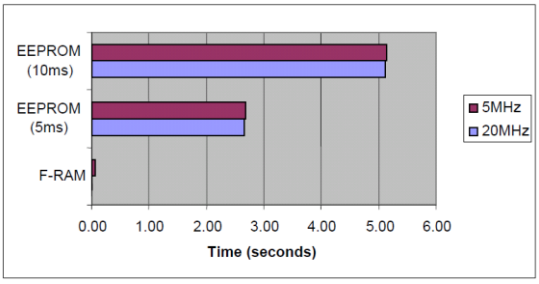


3 Speed Advantages

F-RAM memory technology enables large data blocks to be written much faster than EEPROM or flash equivalents. Unlike EEPROM or flash, F-RAM devices do not use a page buffer. F-RAM writes each data byte immediately following the eighth bit in each byte received. The combination of no-write delays and high clock speed makes the F-RAM a compelling choice for any application that needs to write a lot of data quickly. Designers have complete freedom over how many bytes to write to the SPI F-RAM. When a byte or two is written in random locations in an F-RAM, the write cycle time is approximately 1 µs, whereas an EEPROM or flash imposes its 5 ms to 10 ms write cycle. In addition, designers do not have to worry about page buffer sizes that change when the system grows to the next memory density.

Figure 2 provides a chart that compares the time required to write a 256-Kbit array in F-RAM and EEPROM. Even for an EEPROM with a 64-byte page buffer, the F-RAM device is orders of magnitude faster at the same clock rate. This is especially significant on a production line where there is a limited time to program the system settings.

Figure 2. Write Time to Fill a 256-Kbit SPI Memory Array



Note that the time taken to write a 256-Kbit EEPROM memory is not significantly improved by increasing the clock frequency from 5 MHz to 20 MHz. The long write delay needed for each page-write dominates. For a 20-MHz F-RAM memory, the entire 32-Kbyte array can be written in just 13 ms, which is a small value and does not appear in the chart provided in Figure 2.

4 The SPI Bus

The SPI interface consists of four pins as shown in Figure 1. All transactions occur with $\overline{CS}$ LOW while address, control, and data are serially clocked into the device in byte-size blocks. Address, control, and data-in are clocked in on the SI pin, and data-out is clocked out on the SO pin.

Opcodes provide control over the device. Read and write transactions follow the sequence: opcode, address, and data. Two other transactions that do not involve a data transfer are used to set/clear the Write Enable Latch (WEL) bit in the Status Register (The Status Register is shown in Table 4 and covered in detail later in this document) For EEPROM and flash-based SPI memories, the Status Register also holds an important bit called $\overline{RDY}$ It is the ready flag that tells the SPI controller if a write cycle has completed or not. EEPROMs and flash memories typically require 5 ms to 10 ms of delay before the device can be accessed after a write. With F-RAM there are no delays, no waiting for the internal write to complete, and therefore the $\overline{RDY}$ bit is always logic ‘0’ (The F-RAM Status Register includes this $\overline{RDY}$ = 0 bit so that controllers having firmware that works with the slower EEPROM and Flash memories can quickly adapt to the faster F-RAM product.). In other words, controllers can read and write to F-RAM memory at true RAM speeds.

There are mainly six opcodes that control SPI F-RAM devices. A few of the SPI F-RAM devices add extra opcodes for the Fast Read, Sleep entry, Device ID, and Serial number read functions. Each opcode is an 8-bit command that instructs the memory or Status Register to perform some operation. There can be only one opcode transmitted for each active $\overline{CS}$ cycle. Table 1 describes all the opcodes:

Table 1. Description of Opcodes

Name	Op-Code	Address	Dummy Byte	Data	Action
WREN	0000_0110b	-	-	-	Sets WEL
WRITE	0000_0010b[2]	3-byte[1]	-	Memory Data in	Writes data to F-RAM array if WEL=1. When $\overline{CS}$ goes HIGH, WEL is cleared.
READ	0000_0011b[2]	3-byte[1]	-	Memory Data out	Reads data from F-RAM array
WRDI	0000_0100b	-	-	-	Clears WEL
RDSR	0000_0101b	-	-	Status Register data out	Read WPEN, BP(1:0), WEL bits
WRSR	0000_0001b	-	-	Status Register data in	Write WPEN and BP(1:0) bits
SLEEP[3]	1011_1001b	-	-	-	Enter Sleep mode
FSTRD[3]	0000_1011b	3-byte[1]	1-byte	Memory Data out	Reads data from F-RAM array at 40 MHz
RDID	1001_1111b	-	-	9-byte Device ID data out	Reads 9-byte device id
SNR[3]	1100_0011b	-	-	8-byte Serial Number data out	Reads 8-byte serial number

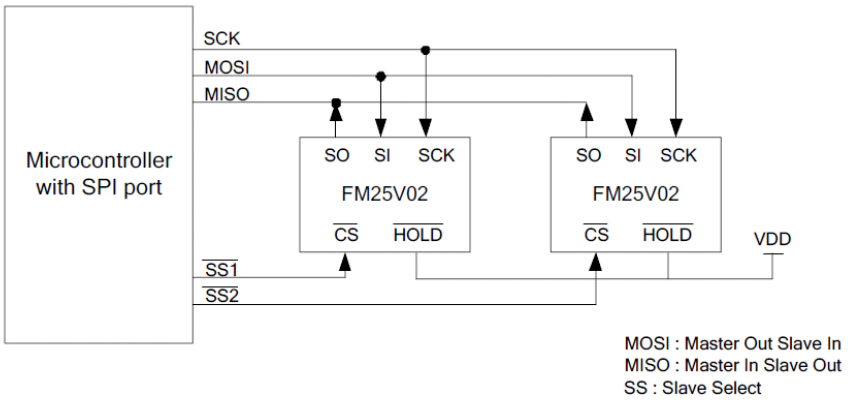
Notes

1. Some SPI devices may use 1-byte or 2-byte addressing depending on the density. Refer to Table 2.
2. For 4-Kbit devices, bit 3 of the Write and Read op-codes correspond to upper address bit (A8).
3. All SPI devices may not support this command.

5 System Hookup

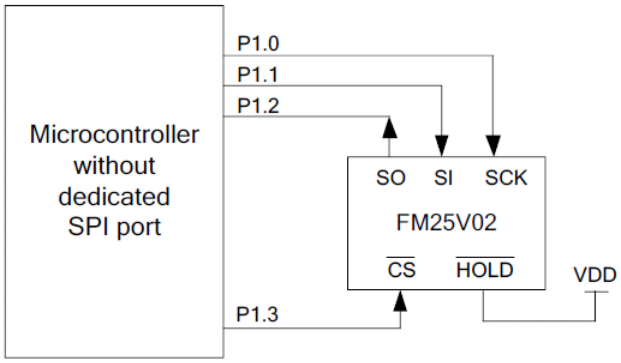
Multiple devices may be used as long as the controller has extra pins to drive a chip-select to each F-RAM device. Figure 3 shows the system configuration for two F-RAM devices interfaced to the standard SPI port of a microcontroller.

Figure 3. System Configuration for Two F-RAM Devices



For a microcontroller that has no dedicated SPI bus, a general purpose port may be used as shown in Figure 4. A bit-banging code drives this interface.

Figure 4. System Configuration for Microcontroller using GPIO – Single F-RAM



6 Standalone SPI F-RAM Products

The following table summarizes the basic characteristics of the standalone SPI F-RAM products.

Table 2. SPI F-RAM Product Lineup

		3V									5V			
	FM25L04B	**FM25L16B**	**FM25CL64B**	**FM25V01**	**FM25V02**	**FM25V05**	**FM25V10**	**FM25V20 FM25V20A**	**FM25H20**	**FM25V40**	**FM25040B**	**FM25C160B**	**FM25640B**	**FM25W256**
Density	4 Kbit	16 Kbit	64 Kbit	128 Kbit	256 Kbit	512 Kbit	1 Mbit	2 Mbit	2 Mbit	4 Mbit	4 Kbit	16 Kbit	64 Kbit	256 Kbit
Organized Internally	512 x 8	2K x 8	8K x 8	16K x 8	32K x 8	64K x 8	128K x 8	256K x 8	256K x 8	512K x 8	512 x 8	2K x 8	8K x 8	32K x 8
Number of address bits	9	11	13	14	15	16	17	18	18	19	9	11	13	15
Number of address bytes	1	2	2	2	2	2	3	3	3	3	1	2	2	2
Operating Voltage	2.7-3.6 V	2.7-3.6 V	2.7-3.65 V	2.0-3.6 V	2.0-3.6 V	2.0-3.6 V	2.0-3.6 V	2.0-3.6 V	2.7-3.6 V	2.0-3.6 V	4.5-5.5 V	4.5-5.5 V	4.5-5.5 V	2.7-5.5 V
Max. Clock Freq.	20 MHz	20 MHz	20 MHz	40 MHz	40 MHz	40 MHz	40 MHz	40 MHz	40 MHz	40 MHz	20 MHz	20 MHz	20 MHz	20 MHz
Supported Clock Modes	0, 3	0, 3	0, 3	0, 3	0, 3	0, 3	0, 3	0, 3	0, 3	0, 3	0, 3	0, 3	0, 3	0, 3
Sleep Mode				✓	✓	✓	✓	✓	✓	✓				
Unique S/N							✓							
Device ID				✓	✓	✓	✓	✓		✓				
Package	SOIC8 DFN8 (4x4.5)	SOIC8 DFN8 (4x4.5)	SOIC8 DFN8 (4x4.5)	SOIC8	SOIC8 DFN8 (4x4.5)	SOIC8	SOIC8	Wide SOIC8 DFN8[1] (5x6)	Wide SOIC8 DFN8[1] (5x6)	Wide SOIC8 DFN8[1] (5x6)	SOIC8	SOIC8	SOIC8	SOIC8

Note:
1. 5 x 6 mm DFN8 conform to SOIC8 footprint.

7 READ/WRITE Transactions

The SPI interface is synchronous to a clock that is driven by the controller. All the F-RAM SPI devices will register data input on the rising edge of SCK and drive the data back to the controller on the falling edge of SCK. To comply with this timing, controllers generally drive signals to the memory on the falling edge of SCK so that the signals have time to propagate and satisfy the setup timing specifications of the memory device.

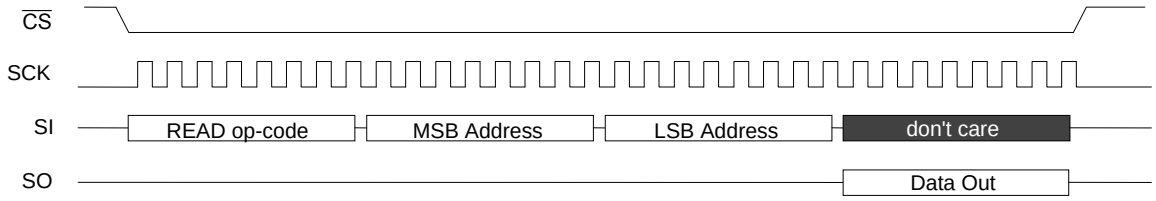
7.1 Memory Reads

Format: READ opcode, MSB Address, LSB Address, Data-out, (Data-out, Data-out ...)

During a Read cycle, the controller issues a READ opcode and address. Data comes out on the SO pin: for example, data-out(0), data-out(1), data-out(2), and so on. The $\overline{CS}$ pin must remain LOW throughout the cycle. When $\overline{CS}$ is deasserted HIGH, data output stops and SO goes to a HI-Z state. The clocked-in address is the starting address of the first data byte. Subsequent data bytes may be accessed simply by keeping $\overline{CS}$ LOW while clocking-out data byte after data byte, each byte being read from an address incremented by the SPI F-RAM device.

Figure 5 shows a two-byte address, which is used for 16 Kbit through 512 Kbit densities.

Figure 5. Read SPI Timing



7.2 Memory Writes

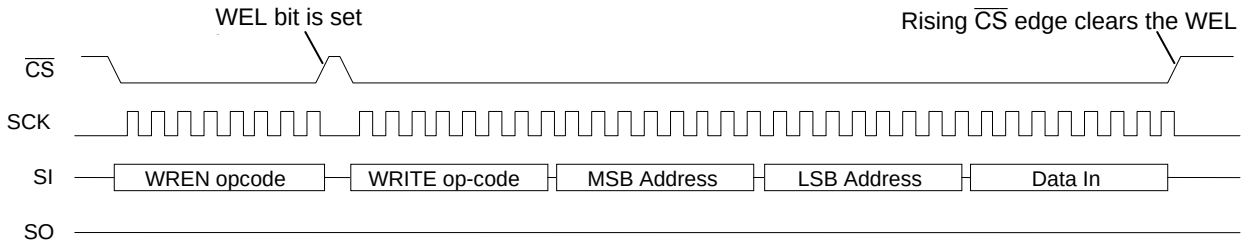
Format: WREN opcode, WRITE opcode, MSB Address, LSB Address, Data-in, (Data-in, Data-in, ...)

A write cycle requires the controller to issue two opcodes, WREN and WRITE, in the following sequence. Each opcode must be bounded by $\overline{CS}$ LOW. The WREN opcode is followed by the WRITE opcode, address, and data; for example, Data-in(0), Data-in(1), Data-in(2), and so on. The clocked-in address is the starting address of the first data byte. Subsequent data bytes may be written by keeping $\overline{CS}$ LOW while clocking-in data byte after data byte, each byte being written to an address incremented by the SPI F-RAM device. Each data byte is written to the F-RAM array on the eighth clock edge of that byte. There are no page buffers or write delays.

Note that the WEL bit in the Status Register is internally set and cleared by the SPI F-RAM device. It is set after clocking-in the WREN opcode and is cleared on the rising edge of $\overline{CS}$ at the end of a write operation. Reading the Status Register (RDSR opcode) between the WREN and WRITE opcodes will not clear the WEL bit. Some users read the Status Register immediately following the WREN to check that the WEL bit is set. However, reading the WEL bit is not necessary to complete the write operation.

Figure 6 shows a complete single-byte write transaction. This shows a two-byte address, which is used for 16 Kbit through 512 Kbit densities.

Figure 6. Write SPI Timing



7.3 Status Register Write

Format: WREN opcode, WRSR opcode, Data-in

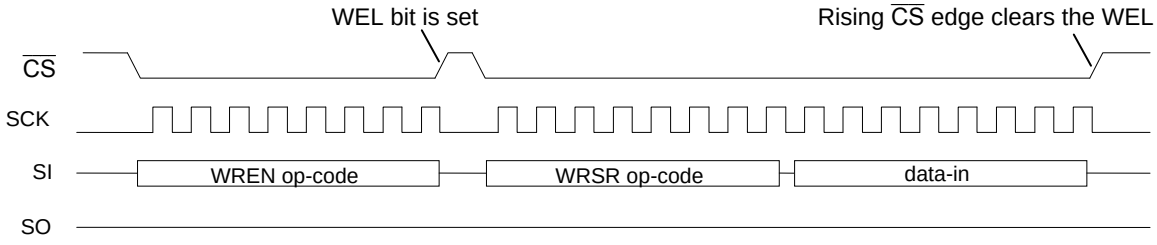
Status register contains the $\overline{WP}$ enable (WPEN) bit and the block protect (BP1, BP0) bits as shown here:

Status Register							
7	6	5	4	3	2	1	0
WPEN	0	0	0	BP1	BP0	WEL	0

Writing the Status Register allows the user to write-protect memory blocks and enable the $\overline{WP}$ pin. There are two block-protect bits, BP1 and BP0. They provide upper quarter, upper half, or entire array protection against writes. The BP0, BP1, and WPEN bits are highlighted yellow to indicate that they are nonvolatile, retaining their written values through power cycling events. WPEN enables or disables the external $\overline{WP}$ pin. Software may be used to override the $\overline{WP}$ pin from system tampering. WEL is a read-only bit that simply tells the user if the Write Enable Latch has been set, which allows writes to either the Status Register or the memory.

A complete status register write transaction is shown in Figure 7.

Figure 7. Write Status Register Timing

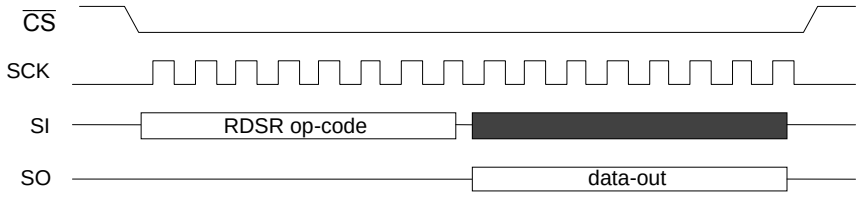


7.4 Status Register Read

Format: RDSR opcode, Data-out

Reading the Status Register allows the user to view the state of the WPEN bit, the BP (1:0) write-protect bits, and the WEL bit. A complete status register read transaction is shown in Figure 8.

Figure 8. Read Status Register Timing



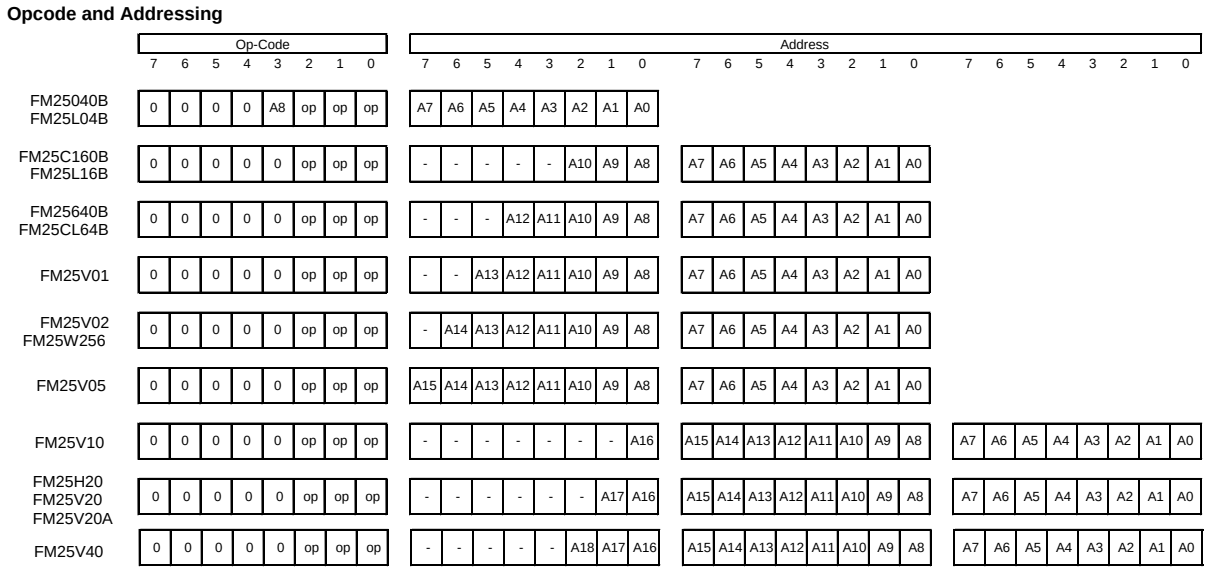
8 SPI F-RAM Addressing

Standalone SPI F-RAM devices require a 1-byte, 2-byte, or 3-byte address, depending on the density. Following the opcode, a starting address is shifted in MSB first. Immediately following the least significant address byte (LSB), data-in is expected from the master for writes and data-out is driven by the memory for reads. As long as SCK continues to toggle, the internal address is automatically incremented and data-in/data-out continues until $\overline{CS}$ is deasserted.

Note: 4-Kbit devices require only one byte of address.

Figure 9 shows the addressing for the different densities of SPI F-RAM devices.

Figure 9. Addressing Differences Between Densities



9 Placing a Higher-Density F-RAM Device in a Low-Density Socket

If a particular device is not available and the board or system is designed for a lower density F-RAM device, it is possible to substitute with a higher density device. For example, a system that is designed to use a 16-Kbit device can also use a 64 Kbit, 128 Kbit, 256 Kbit, or 512 Kbit device. Figure 9 shows that these devices use a common 2-byte address for read and write operations. The devices within these densities are identical in terms of pinout, package (SOIC), and read/write functionality, assuming that the specified operating voltage and timing requirements are met. There are two potential issues that may be a problem: the system uses the address wrap feature in the device, or the system uses the block protect feature. A 16-Kbit device wraps at 0x800, a 64-Kbit device wraps at 0x2000, a 128-Kbit device address wraps at 0x4000, and so on. The block-protect boundaries are spaced at twice the address when comparing device densities that are 2x from each other.

For example, Figure 10 and Figure 11 show the differences in the serial address streams between 16-Kbit and 128-Kbit devices.

When the 16-Kbit and 128-Kbit device address requirements shown in Figure 9 are compared, it is clear that the three additional address bit locations (A13, A12, and A11 circled in RED as shown in Figure 11) are used on a 128-Kbit device. As long as the controller drives these three address bits consistently for both reads and writes, a higher-density device will work in a system designed for a lower-density part. A 1-Mbit density (or higher) device uses a 3-byte address and cannot be used as a replacement part in a system that is designed for lower densities. For example, a system that issues a 2-byte address will not work properly if a 3-byte address memory is used. Refer to Table 2 for information about the number of address bytes by density.

Figure 10. FM25L16B Write Cycle (WREN not shown)

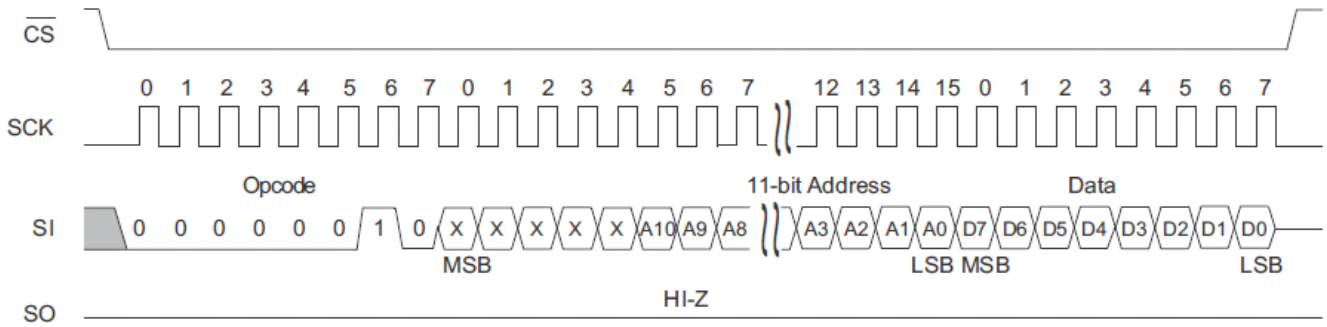
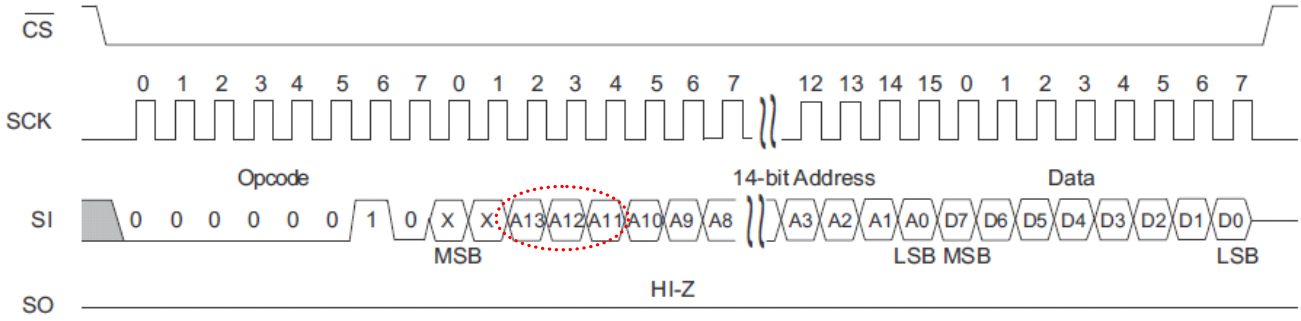


Figure 11. FM25V01 Write Cycle (WREN not shown)



10 Clocking Modes

The FM25xxx device families support two of the four SPI standard clocking modes: Mode 0 and Mode 3. Note that independent of the mode, all F-RAM parts clock data into the device on the rising SCK edge and clock data out on the falling edge of SCK. The difference between Modes 0 and 3 is simply whether SCK starts LOW or HIGH when $\overline{CS}$ is asserted LOW. The different SPI Modes are listed in Table 3.

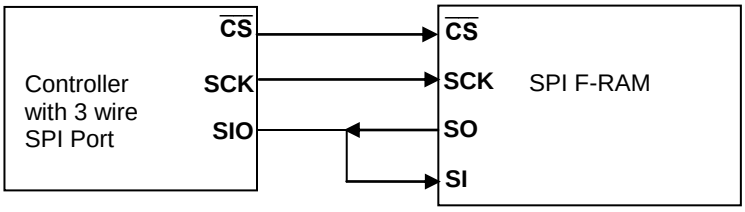
Table 3. SPI Modes

	Mode 0	Mode 1	Mode 2	Mode 3
SCK Starts …	**LOW**	**LOW**	**HIGH**	**HIGH**
SI Data-In Latched on …	↑	↓	↓	↑
SO Data-Out Driven from …	↓	↑	↑	↓

11 Half-Duplex Operation

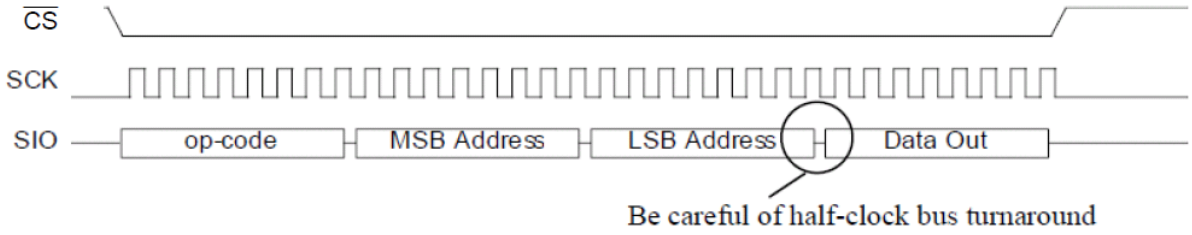
To reduce the pin count on an SPI interface, the data lines can be tied together to create a common data I/O line. This 3-wire interface is SPI's minimum pin count configuration as shown in Figure 12. The controller must now ensure that the SIO line is hi-Z during read cycles. Otherwise, bus contention will occur. A secondary issue is that because the data bus is now half-duplex, this potentially reduces the data bandwidth.

Figure 12. True 3-Wire SPI Interface



Regardless of the timing mode (0 or 3), all SPI F-RAMs latch data-in on the rising edge of SCK and drive data-out on the falling edge of the SCK. An SPI read transaction is shown in Figure 13.

Figure 13. Common Data I/O Line and Bus Turnaround



12 Write Protection

SPI F-RAM devices can be write-protected using the $\overline{WP}$ hardware pin or by programming the Status Register bits. The Status Register itself may be protected in addition to the F-RAM memory array. The Status Register contains nonvolatile block-protect bits BP (1:0) to disable writes to portions of the memory array. BP0, BP1, and WPEN bits are highlighted yellow in Table 5 to indicate that they are nonvolatile and the settings will survive power cycling. WPEN enables the $\overline{WP}$ hardware pin. Bit location 0 is reserved as a $\overline{RDY}$ bit for compatibility with EEPROM and serial Flash. This bit is used in these devices so that the user can determine whether or not the memory is ready for another command by reading the Status Register. The $\overline{RDY}$ bit is internally hardwired LOW in all SPI F-RAM devices because the chip is always ready (zero delay) after a write cycle.

Table 4: Status Register and Block Protect Settings

Status Register							
7	6	5	4	3	2	1	0
WPEN	0	0	0	BP1	BP0	WEL	0

A write-protection table is provided in Table 5, which covers all the cases for write-protecting the Status Register and the F-RAM array. When WEL = 0, all the writes to the F-RAM array and status register are blocked.

Table 5. Write Protection

WEL	WPEN	$\overline{WP}$	Protected Blocks	Unprotected Blocks	Status Register
0	X	X	Protected	Protected	Protected
1	0	X	Protected	Unprotected	Unprotected
1	1	0	Protected	Unprotected	Protected
1	1	1	Protected	Unprotected	Unprotected

- F-RAM array is protected by BP bit(s) HIGH, even if WPEN = 0 and $\overline{WP}$ pin = 1
- F-RAM array is not protected when BP bit(s) LOW, even if WPEN = 1 and $\overline{WP}$ pin = 0
- Status Register is protected only when WPEN = 1 and $\overline{WP}$ pin = 0

Note: FM25040B and FM25L04B do not have a WPEN bit. All writes (memory array and status register) are blocked when $\overline{WP}$ pin = 0.

13 Power Cycling

An F-RAM device is a high-speed nonvolatile memory and power glitches occurring during either a read or write sequence may incorrectly overwrite (corrupt) array data. For example, the device can inadvertently write data at mid-level power supply levels when chip-select is active (LOW). The SPI F-RAM datasheets specify (recommend) that the device is powered down with chip-select inactive (HIGH).

SPI F-RAM devices have no power management circuits other than a simple internal power-on reset circuit. Ensure that V_{DD} is within the datasheet tolerances to prevent incorrect operation. It is recommended that the V_{DD} power supply voltage ramp up and ramp down in a well-controlled manner. Switch-mode power supplies are notorious for uncontrolled outputs as they power-up or power-down.

The system designer should be aware of chip-enable and V_{DD} states during power cycles. For more details on data protection, refer to "*AN302 - F-RAM SPI Read & Write Internal Operation and Data Protection*".

14 Summary

The application note covers the functional features, timing, and example code for different F-RAM SPI parts.

15 Related Application Notes

You can refer to the following application notes for better understanding of the SPI F-RAM devices.

- AN302 - F-RAM SPI Read & Write Internal Operation and Data Protection
- AN408 - A Design Guide to SPI F-RAM Processor Companion - FM33256B

A Pseudo Code Examples (1-Byte Address, 4-Kbit Devices)

For code example project, refer to CE204087 - Interfacing SPI nvRAM with PSoC® 3/5

```
#define WREN  0x06
#define WRITE 0x02 // Write opcode to access lower half of memory
#define WRITE 0x0A // Write opcode to access upper half of memory
#define READ  0x03  // Read opcode to access lower half of memory
#define READ  0x0B  // Read opcode to access upper half of memory
#define RDSR  0x05
#define WRSR  0x01
#define WRDI  0x04
```

In every case below, the parentheses designate the $\overline{CS}$ pin going LOW "(" and HIGH ")".

```
/****** Memory Write (single byte to location 0130h) *******/
WREN (0x06)         // Sets WEL bit. WREN must precede WRITE opcode.
WRITE (0x0A,        // 0x02 is WRITE opcode and A8 bit set
0x30,               // starting address
0x55)               // 0x55 is data written to location 0130h

/****** Memory Write (multiple bytes to starting location 01FCh) *******/
WREN (0x06)         // Sets WEL bit. WREN must precede WRITE opcode.
WRITE (0x0A,        // 0x02 is WRITE opcode and A8 bit set
0xFC,               // starting address
0x55,               // 0x55 is data written to location 01FCh
0xAA,               // 0xAA is data written to location 01FDh
0x55,               // 0x55 is data written to location 01FEh
0xAA)               // 0xAA is data written to location 01FFh

/****** Memory Read (single byte from location 01D3h) *******/
READ (0x0B,         // 0x03 is READ opcode
0xD3,               // starting address
0xAA)               // 0xAA is data read from location 01D3h

/****** Memory Read (multiple bytes from starting location 01FCh) *******/
READ (0x0B,         // 0x03 is READ opcode
0xFC,               // starting address
0x55,               // 0x55 is data read from location 01FCh
0xAA,               // 0xAA is data read from location 01FDh
0x55,               // 0x55 is data read from location 01FEh
0xAA)               // 0xAA is data read from location 01FFh

/****** Write Status Register (write protect upper half of memory) *******/
WREN (0x06)         // Sets WEL bit. WREN must precede WRSR opcode.
WRSR (0x01,         // 0x01 is WRSR opcode
0xF8)               // 0xF8 sets the BP1 bit which protects the upper
                    // half of the memory array. The upper nibble
                    // set to "F" attempts to write the upper bits
                    // to 1.

/****** Read Status Register *******/
RDSR (0x05,         // 0x05 is RDSR opcode
0x08)               // 0x08 tells us that the BP1 bit is set and that
                    // the upper half of the memory array is protected.
                    // The upper nibble returns "0" since they are
                    // hardwired low.
```

NOTE: Text in **BLUE** indicates data being sent by the controller. Text in **RED** indicates data being received by the controller.

B Pseudo Code Examples (2-Byte Address, 16-Kbit Through 512-Kbit Devices)

For code example project, refer to CE204087 - Interfacing SPI nvRAM with PSoC® 3/5

```
#define WREN 0x06
#define WRITE 0x02
#define READ 0x03
#define RDSR 0x05
#define WRSR 0x01
#define WRDI 0x04
```

In every case below, the parentheses designate the $\overline{CS}$ pin going LOW "(" and HIGH ")".

```
/****** Memory Write (single byte to location 0F30h) *******/
WREN (0x06)  // Sets WEL bit. WREN must precede WRITE opcode.
WRITE (0x02, // 0x02 is WRITE opcode
0x0F,        // starting address MSB
0x30,        // starting address LSB
0x55)        // 0x55 is data written to location 0F30h

/****** Memory Write (multiple bytes to starting location 07FC) *******/
WREN (0x06)  // Sets WEL bit. WREN must precede WRITE opcode.
WRITE (0x02, // 0x02 is WRITE opcode
0x07,        // starting address MSB
0xFC,        // starting address LSB
0x55,        // 0x55 is data written to location 07FCh
0xAA,        // 0xAA is data written to location 07FDh
0x55,        // 0x55 is data written to location 07FEh
0xAA)        // 0xAA is data written to location 07FFh

/****** Memory Read (single byte from location 0F31h) *******/
READ (0x03,  // 0x03 is READ opcode
0x0F,        // starting address MSB
0x31,        // starting address LSB
0xAA)        // 0xAA is data read from location 0F31h

/****** Memory Read (multiple bytes from starting location 07FCh) *******/
READ (0x03,  // 0x03 is READ opcode
0x07,        // starting address MSB
0xFC,        // starting address LSB
0x55,        // 0x55 is data read from location 07FCh
0xAA,        // 0xAA is data read from location 07FDh
0x55,        // 0x55 is data read from location 07FEh
0xAA)        // 0xAA is data read from location 07FFh

/****** Write Status Register (write protect upper half of memory) *******/
WREN (0x06) // Sets WEL bit. WREN must precede WRSR opcode.
WRSR (0x01, // 0x01 is WRSR opcode
0x08)       // 0x08 sets the BP1 bit which protects the upper
            // half of the memory array.
/****** Read Status Register *******/
RDSR (0x05, // 0x05 is RDSR opcode
0x88)       // 0x88 tells us that the BP1 bit is set and that
            // the upper half of the memory array is protected.
            // The WPEN bit is also set which works with
            // the WP pin to protect the Status Register.
```

NOTE: Text in **BLUE** indicates data being sent by the controller. Text in **RED** indicates data being received by the controller.

C Pseudo Code Examples (3-Byte Address, 1-Mbit Through 4-Mbit Devices)

For code example project, refer to CE204087 - Interfacing SPI nvRAM with PSoC® 3/5

```
#define WREN 0x06
#define WRITE 0x02
#define READ 0x03
#define RDSR 0x05
#define WRSR 0x01
#define WRDI 0x04
```

In every case below, the parentheses designate the $\overline{CS}$ pin going LOW"(" and HIGH")".

```
/****** Memory Write (single byte to location 1BF30h) *******/
WREN (0x06)  // Sets WEL bit. WREN must precede WRITE opcode.
WRITE (0x02, // 0x02 is WRITE opcode
0x01,        // starting address MSB
0xBF,        // starting address 2nd byte
0x30,        // starting address LSB
0x55)        // 0x55 is data written to location 1BF30h

/****** Memory Write (multiple bytes to starting location 1B7FC) *******/
WREN (0x06)  // Sets WEL bit. WREN must precede WRITE opcode.
WRITE (0x02, // 0x02 is WRITE opcode
0x01,        // starting address MSB
0xB7,        // starting address 2nd byte
0xFC,        // starting address LSB
0x55,        // 0x55 is data written to location 1B7FCh
0xAA,        // 0xAA is data written to location 1B7FDh
0x55,        // 0x55 is data written to location 1B7FEh
0xAA)        // 0xAA is data written to location 1B7FFh

/****** Memory Read (single byte from location 1BF31h) *******/
READ (0x03,  // 0x03 is READ opcode
0x01,        // starting address MSB
0xBF,        // starting address 2nd byte
0x31,        // starting address LSB
0xAA)        // 0xAA is data read from location 1BF31h

/****** Memory Read (multiple bytes from starting location 1B7FCh) *******/
READ (0x03,  // 0x03 is READ opcode
0x01,        // starting address MSB
0xB7,        // starting address 2nd byte
0xFC,        // starting address LSB
0x55,        // 0x55 is data read from location 1B7FCh
0xAA,        // 0xAA is data read from location 1B7FDh
0x55,        // 0x55 is data read from location 1B7FEh
0xAA)        // 0xAA is data read from location 1B7FFh

/****** Write Status Register (write protect upper half of memory) *******/
WREN (0x06)  // Sets WEL bit. WREN must precede WRSR opcode.
WRSR (0x01,  // 0x01 is WRSR opcode
0x08)        // 0x08 sets the BP1 bit which protects the upper
             // half of the memory array.

/****** Read Status Register *******/
RDSR (0x05,  // 0x05 is RDSR opcode
0x88)        // 0x88 tells us that the BP1 bit is set and that
             // the upper half of the memory array is protected.
             // The WPEN bit is also set which works with
             // the WP pin to protect the Status Register.
```

NOTE: Text in **BLUE** indicates data being sent by the controller. Text in **RED** indicates data being received by the controller.

Document History

Document Title: AN304 - SPI Guide for F-RAM™

Document Number: 001-87196

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	4039506	MEDU	06/25/2013	New Spec.
*A	4557729	MEDU	10/31/2014	Added PSoC 3-based User Module project Updated Table 1 with all opcodes
*B	4573028	MEDU	11/18/2014	Added the associated project.
*C	4581444	MEDU	11/27/2014	Updated for 4-Mbit device
*D	5293268	MEDU	06/02/2016	Added a reference to code example CE204087 Updated template
*E	5702254	AESATMP8	04/26/2017	Updated logo and Copyright.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709