FRAM SPI Reads & Writes and Data Protection During Power Cycles

Applies to All SPI FRAM Devices except FM25640

OVERVIEW

Non-volatile FRAM memory offers tremendous write speed. This feature is most evident in the high speed serial SPI and bytewide parallel memories. Hundreds of bytes can be written in tens of microseconds. EEPROM and Flash memory require tens of milliseconds to do the same. Writing data quickly before losing power is particularly useful in systems that require saving machine state information, parameter settings, or other vital data. With this in mind, care must be taken in controlling signals at power-up and power-down. A FRAM device that is active during a power cycle will potentially overwrite array data very quickly. It is important to understand that they might "remember" things at power supply levels that are mid-level when control signals are active (/CS low).

FRAM SPI devices have no power management circuits other than a simple internal power-on reset circuit. It is the user's responsibility to ensure that V_{DD} is within datasheet tolerances to prevent incorrect operation. It is recommended that the device is powered down with chip enable inactive (high). The system designer should be aware of chip-enable and V_{DD} states during power cycles, especially considering the strange waveforms delivered by switching power supplies, manually controlled power switches, multi-stage turn-on supplies, etc. This application note describes the internal operation of Ramtron's high speed SPI devices (except FM25640) and also offers some system design suggestions to avoid data corruption.

FRAM SPI READ/WRITE CYCLES

All SPI op-codes, addresses, and data are treated as 8-bit data transfers, therefore all internal operations are byte-wide in nature. The serial data is internally buffered by an 8-bit shift register. All operations are initiated on the falling edge of /CS, which is an edge-triggered input. A read or write op-code is then clocked-in, followed by address and data bytes. Memory data is shifted in/out as 8 bits and are transferred internally to/from the memory array as byte-wide transfers. Array accesses are not performed on a clock-by-clock or bit-by-bit basis. For reads, the data is internally prefetched into the 8-bit shift register and subsequently clocked out. Writes to the memory occur after each data byte is clocked-in.

A read cycle starts the array access on the 5th rising clock edge of the last address byte and completes automatically. The data byte is loaded into the 8-bit shift register as a parallel transfer and the data is clocked out. Once the register is loaded, the array is automatically closed. For 64Kb and lower densities, the array read access begins on the 6th clock of the last address byte.

A write cycle starts the memory access on the 5th clock of the data byte but does not complete until the 8th data bit is shifted-in. The write transaction, therefore, completes after the rising edge of the 8th clock. This means the array stays open until the device receives the last data bit, the length of time being dependent on the user. Using the /HOLD pin or stopping the clock will keep the memory array open for an extended period of time.

READ OPERATIONS

For read cycles, the memory array access begins on the 5th rising clock edge of the last address byte and the 5th clock of each subsequent data byte. The array is automatically closed by an internally self-timed circuit. The user has no control over the duration of the internal memory access. Figure 1 shows that the first access loads Data Byte n at marker #3 and the data is shifted out, the second access loads Data Byte n+1 at marker #4 and the data is shifted out, and the third access loads Data Byte n+2 but because /CS goes high, the data is never shifted out. Note that /CS transitions are asynchronous, so deasserting /CS at anytime immediately terminates the operation.

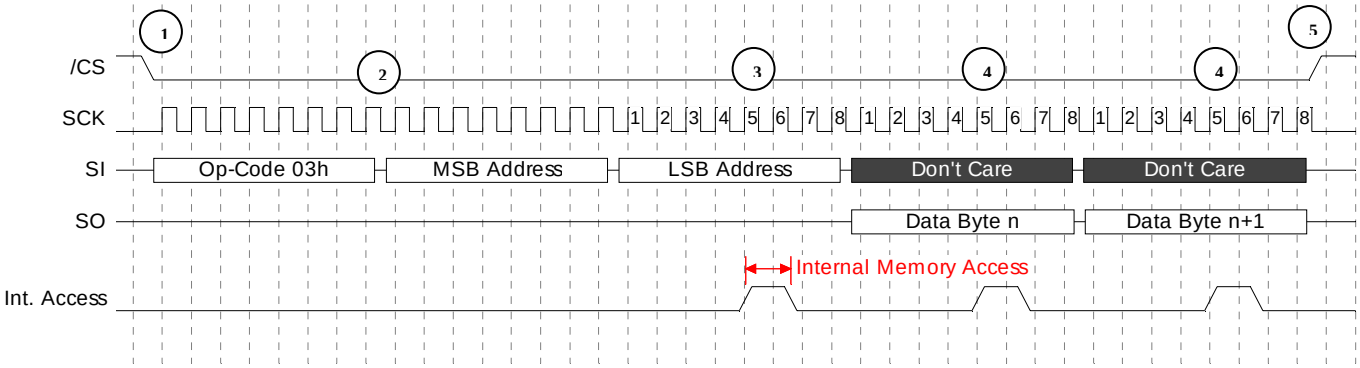

Figure 1. Example of a Read Cycle (FM25256/FM25L256)

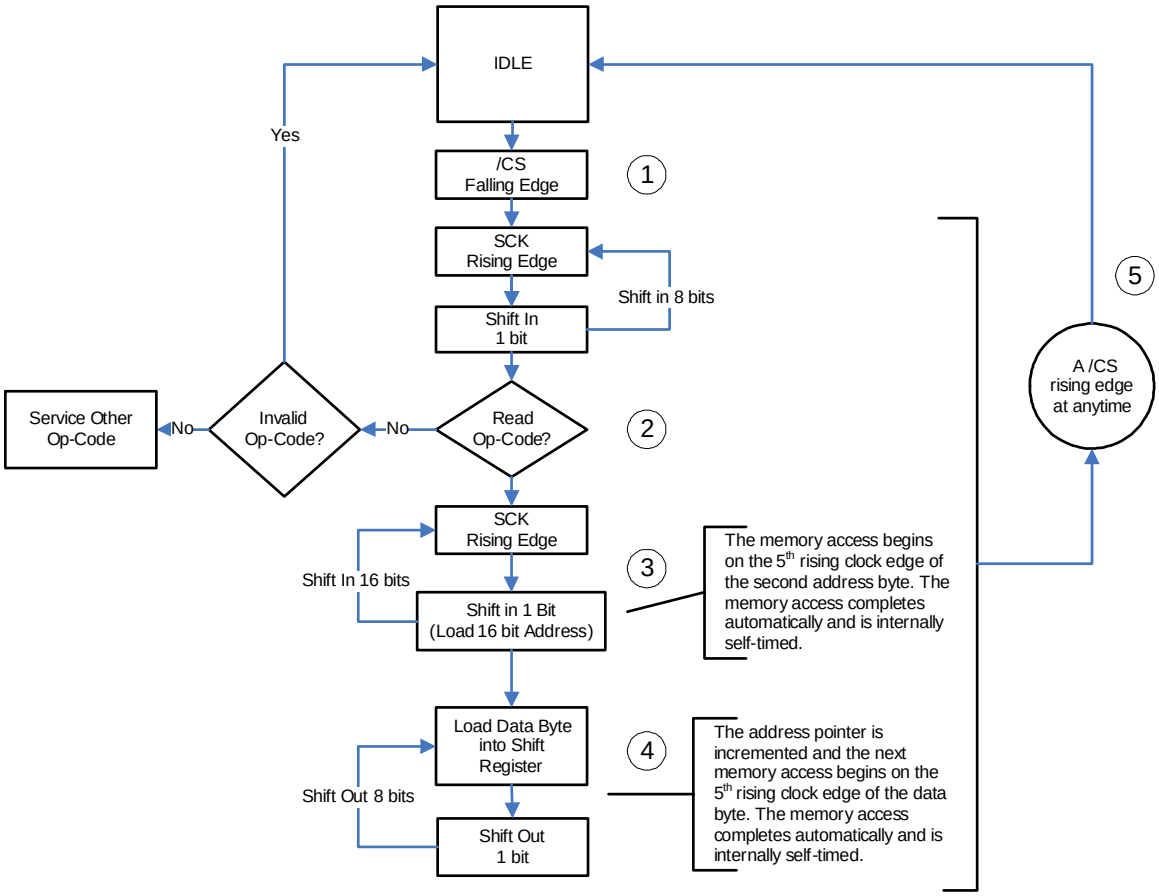

Figure 2. Detailed Flow Diagram of a Read Cycle

WRITE OPERATIONS

A write cycle will not begin unless the WEL bit is set. The WREN op-code enables write cycles by setting the WEL bit. This enable bit is internally reset after a write cycle terminates. The WRITE op-code and two-byte address is then clocked-in, and the memory array is accessed on the 5th clock of the data byte and remains open until the 8th rising clock edge. If the clock is suspended or /HOLD is asserted, the memory access does not complete. It is therefore important that V_{DD} remain within its specified range while the memory access (write) is pending. Do not powerdown the device with a write access pending. Once V_{DD} drops below the minimum specified voltage, the /CS pin must be pulled up to V_{DD} in order to cleanly terminate any ongoing accesses.

If /CS is deasserted before the 8th clock of the data byte, the write operation is aborted and the internal memory cycle is completed. If /CS is deasserted after the 8th clock, the write operation will complete automatically. The write operation is very fast (<200ns). Since V_{DD} does not fall significantly in 200ns, the write cycle is not compromised. Figures 3 and 4 are labeled with numbers that correspond to certain events.

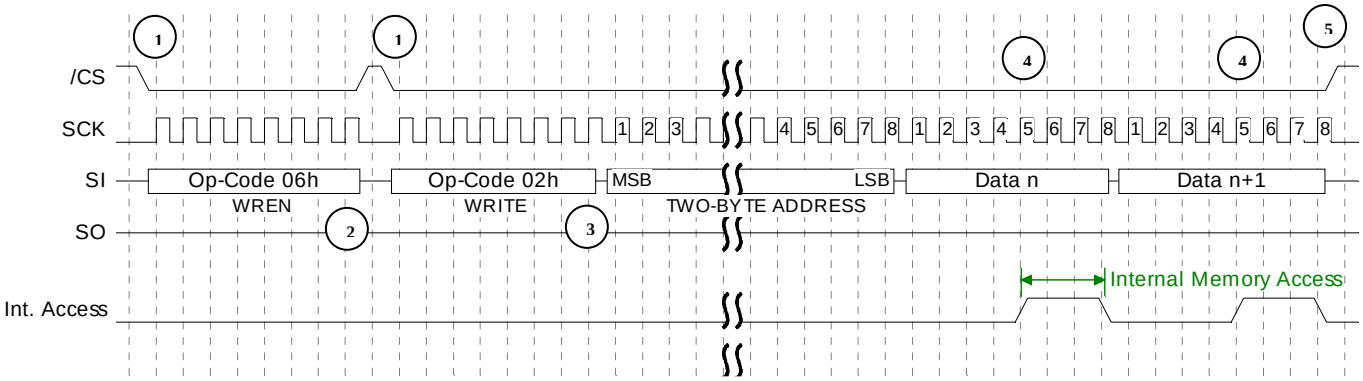

Figure 3. Example of a Write Cycle (FM25256/FM25L256)

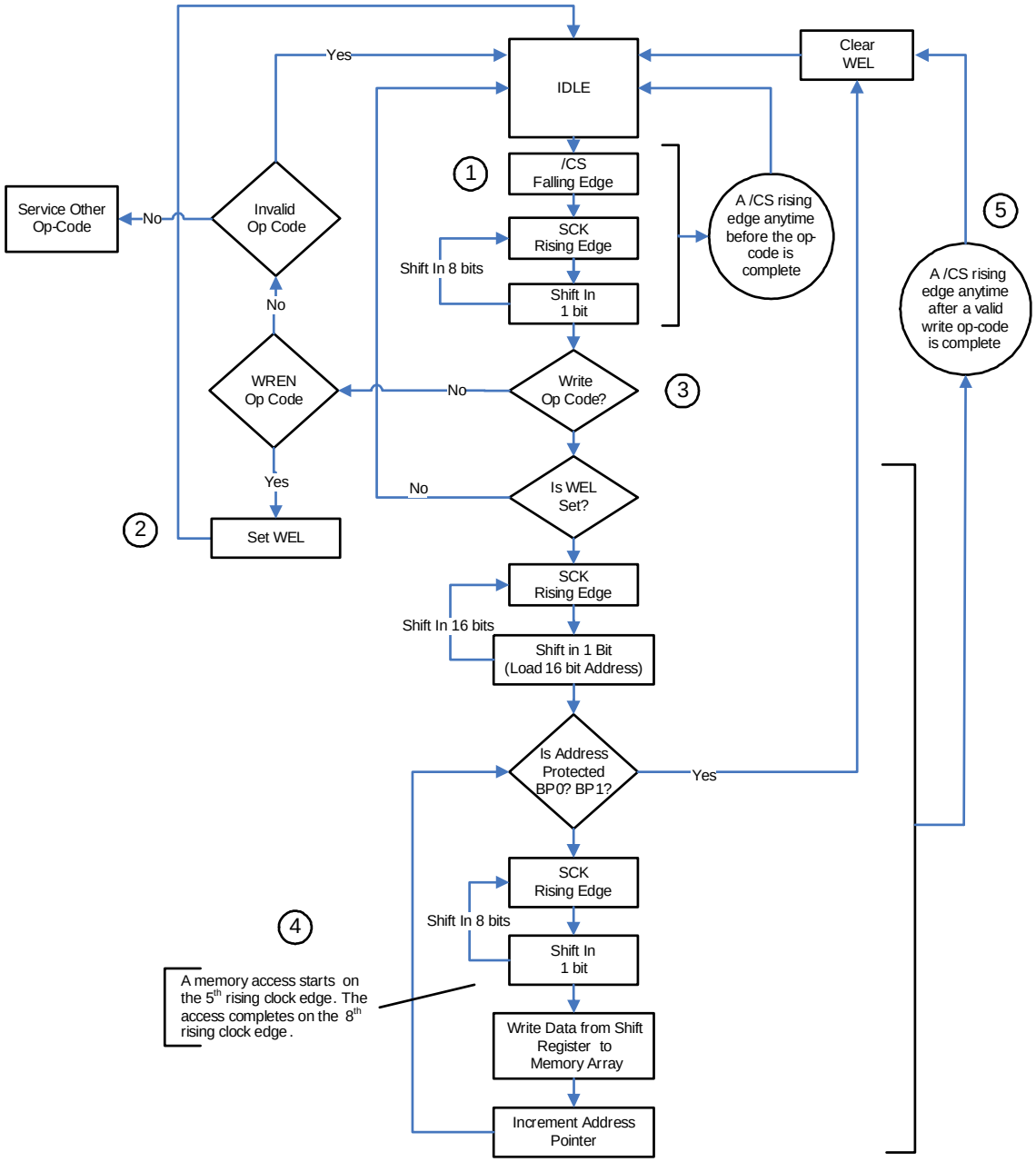

Figure 4. Detailed Flow Diagram of a Write Cycle

/CS NEEDS ATTENTION

To protect the FRAM from corrupting data during power cycles, we recommend that the primary control pin, /CS, is held inactive as V_{DD} powers up and powers down. In many cases, this may be as simple as a pullup resistor on the MCU's output pin that drives /CS. A value in the 5K-10K ohm is recommended. As the system microcontroller powers up, its outputs will tri-state before the power supply reaches sufficient voltage to turn various internal circuits on, thereby allowing the pullup resistor to keep the signal at V_{DD}. Likewise, at powerdown there is a V_{DD} voltage reached that causes the outputs to "let go", again allowing the pullup resistor to do its job.

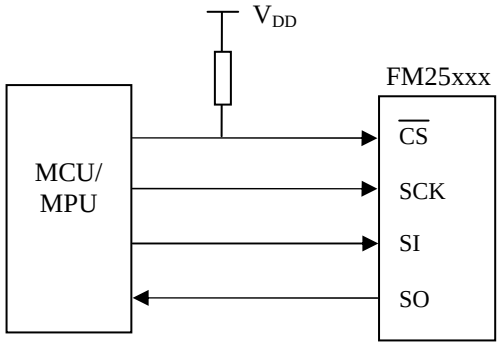

Figure 5. Pullup Resistor Tied to Microcontroller's Tri-Stateable Buffer

However, not all MCUs are so well behaved. The use of a system reset chip and tri-stateable buffer between the system microcontroller and FRAM may be required. Not all microcontrollers will be high-Z at voltages below their normal operating range. A tri-stateable buffer has an active-low enable input that places the outputs in a hi-Z state whenever the pin is high. These buffers typically are well behaved down to 1V. The active-high POR output releases when it reaches its trip voltage. Shown below is a '1G125 single gate buffer. These are packaged in a very small SOT-23 package, requiring very little board space.

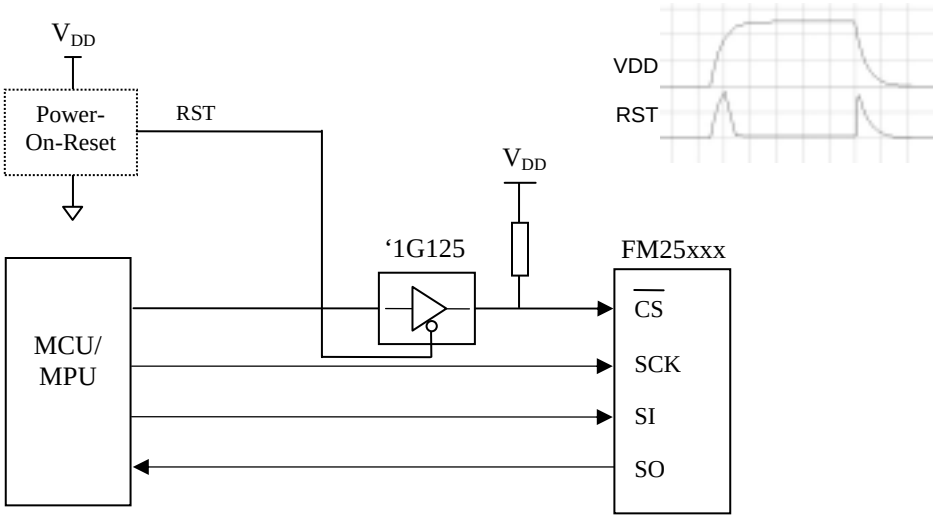

Figure 6. Tri-Stateable Buffer Used on /CS pin of SPI FRAM Device