Programmer's Guide for the Spansion® 65 nm FL-S MirrorBit® Flash Family

By: Gary Swalling

1. Introduction

The S25FL-S NOR, Serial Peripheral Interface flash memory family uses 3.0-volt core voltage, Versatile I/O voltage, and MirrorBit 65 nm technology. This guide discusses new features of the S25FL-S (FL-S) family and software considerations the designer should make when migrating from the S25FL-P family. This document deals with the 128 Mbit, 256 Mbit, and 512 Mbit densities only. Please refer to the S25FL-S family data sheets for detailed specifications.

2. Feature Comparisons Summary

Consider the following items when adapting software with the S25FL-P family, for use with the S25FL-S family. The S25FL-S features include:

- Programming via a 256- or 512-byte aligned and length Page Program Buffer
- CFI and ID information similar to FL129P devices
- Bank address register for accessing locations above 16 Mbyte (128 Mbit)
- An AutoBoot feature that enables code reads immediately after reset
- Larger 1024-byte One Time Program (OTP) area
- Suspend and resume of program and erase operations
- Latency code configuration for reading at fast clock speeds
- Modified timing for various embedded operations
- Command set changes:
  - Removed Deep Power-Down feature
  - Removed the 8-kB Parameter Sector Erase (40h)
  - Removed parallel programming mode
  - New commands with 4-byte address
  - New Double Data Rate (DDR) read commands
  - New Advanced Sector Protection features
  - New Software Reset command
  - New Mode Bit Reset command
  - New Read Serial Flash Discoverable Parameters command (FL512S only)

Publication Number Prog_Guide_65nm_FL-S_Family **Revision** 01 **Issue Date** February 7, 2012

Table 2.1 Feature Comparisons

Key Features	FL128P	FL129P	FL-S	FL512S
Technology	MirrorBit	MirrorBit	MirrorBit Eclipse™	MirrorBit Eclipse
Process Node	90 nm	90 nm	65 nm	65 nm
Densities	128 Mb	128 Mb	128 Mb to 256 Mb	512 Mb
SPI Modes	0 and 3	0 and 3	0 and 3	0 and 3
MIO	No	Yes	Yes	Yes
Parallel (8 bit) mode	Yes	No	No	No
Sector Erase Architecture	Uniform 256 kB or Uniform 64 kB	Uniform 256 kB or Hybrid 64 kB (1)	Uniform 256 kB or Hybrid 64 kB (1)	Uniform 256 kB
Command 20h Erases	64-kB sector	4-kB parameter sector	4-kB parameter sector	none
Page Program Buffer	256 bytes	256 bytes	512 or 256 (2) bytes	512 bytes
Accelerated Programming	Yes	Yes	No	No
CFI	No	Yes	Yes	Yes
SFDP	No	No	No	Yes
Erase and Program Suspend / Resume	No	No	Yes	Yes
Block Protect bits	Yes	Yes	Yes	Yes
Advanced Sector Protection	No	No	Yes	Yes
OTP	No	506 bytes	1024 bytes	1024 bytes
Program-Erase Endurance (typical)	100,000 cycles per sector	100,000 cycles per sector	100,000 cycles per sector	100,000 cycles per sector
Data Retention (typical)	20-year	20-year	20-year	20-year

Notes:

1. *S25FL129P, S25FL128S and S25FL256S devices with 64-kB sectors have thirty-two 4-kB sub-sectors (top or bottom, depending on TBPARM).*
2. *FL-S devices with 256-kB sectors have a 512B Page Program Buffer. FL-S devices with 64-kB sectors have a 256B Page Program Buffer.*

3. Data Alignment and Granularity

Many applications store data in multiples of 512 bytes. The S25FL-S devices have a 512-byte (or 256-byte) Page Program Buffer aligned on buffer-size boundaries. Programming data to the flash is most efficient when writing in buffer-size length and aligned increments. Although smaller writes are allowed, software should be modified, whenever possible, to program data in full, address aligned, buffer increments.

For smaller or misaligned data writes, it is important to note that internally, data is programmed in address aligned groups of 16 bytes. For optimal flash performance and reliability, data should be programmed in multiples of full 16-byte aligned groups, up to the buffer size. While multiple program operations within a group are not recommended on S25FL-S devices, they are allowed for compatibility with legacy SPI devices.

For example, a simple flash file system might write two 512-byte file sectors, each with 12 bytes of metadata. Programming this data sequentially would cause several misalignments.

Table 3.1 Misaligned Data Storage

Order	1st	2nd	3rd	4th
Size	512 bytes	12 bytes	512 bytes	12 bytes
Byte Offset	0	512	524	1036

<table>
<tr><td>Data Written</td><td colspan="3">Initial 512 bytes</td><td>12 bytes</td><td colspan="4">512 bytes</td><td>12 bytes</td><td>Not Written</td></tr>
<tr><td>Internal Pages</td><td>Group 0</td><td>…</td><td>Group 31</td><td>Group 32</td><td>Group 33</td><td>…</td><td>Group 63</td><td>Group 64</td><td>Group 65</td><td>Group 66</td></tr>
</table>

Instead, the writes should be rearranged to maximize programming performance. In Table 3.2, sector data is written from the bottom of flash, and metadata is written from the top. The S indicates four bytes that are skipped and left unused. Group N is the last page in the device. M is the size of the flash device in bytes.

Table 3.2 Aligned Data Storage

Order	1st	2nd	3rd	4th
Size	512 bytes	12 bytes	512 bytes	12 bytes
Byte Offset	0	M - 16	512	M - 32

Data Written	Initial 512 bytes			512 bytes			Not Written	12 bytes	S	12 bytes	S
Internal Groups	Group 0	…	Group 31	Group 32	…	Group 63	…	Group N - 1		Group N	

Refer to the S25FL-S data sheets for more information on programming data.

4. Device Identification and CFI

The S25FL-S devices have Device Identification (ID) and Common Flash Interface (CFI) information that are backward compatible with the S25FL-P family. The CFI information is accessible through the same command (9Fh) defined for previous devices, but CFI values have changed at some locations, and additional data is available. Refer to the S25FL-S data sheets for a full explanation of the ID codes and the CFI data.

The S25FL-S devices have Device ID codes that are backward compatible with FL-P devices of the same density. Most features are backward compatible, but as described in other sections of this document, there are some differences that systems designers should consider. Software that only reads the ID codes to identify a device will not be aware of the S25FL-S differences and could misuse S25FL-S devices.

Table 4.1 Read_ID or REMS (90h) Manufacturer and Device IDs

Device	Manufacturer ID Byte	Device ID Byte
S25FL128P	01h	17h
S25FL129P	01h	17h
S25FL128S	01h	17h
S25FL256S	01h	18h
S25FL512S	01h	19h

Table 4.2 Read Electronic Signature or RES (ABh) Device IDs

Device	Device ID Byte
S25FL128P	17h
S25FL129P	17h
S25FL128S	17h
S25FL256S	18h
S25FL512S	19h

Table 4.3 RDID (9Fh) Manufacturer and Device IDs

Device	Primary Erase Block Size	Manufacturer Byte 0	Device ID Byte 1	Device ID Byte 2	Extended Byte 3	Extended Byte 4	Reserved Byte 5	Reserved Byte 6:F
S25FL128P	64 kB	01h	20h	18h	03h	01h	X	X
S25FL128P	256 kB	01h	20h	18h	03h	00h	X	X
S25FL129P	64 kB	01h	20h	18h	4Dh	01h	X	X
S25FL129P	256 kB	01h	20h	18h	4Dh	00h	X	X
S25FL128S	64 kB	01h	20h	18h	4Dh	01h	80h	X
S25FL128S	256 kB	01h	20h	18h	4Dh	00h	80h	X
S25FL256S	64 kB	01h	02h	19h	4Dh	01h	80h	X
S25FL256S	256 kB	01h	02h	19h	4Dh	00h	80h	X
S25FL512S	256 kB	01h	02h	20h	4Dh	00h	80h	X

Notes:

1. *X indicates undefined data*
2. *Byte 3 is the length of additional extended data for S25FL-P devices. For devices other than S25FL128P, the extended data includes the CFI information, including the primary algorithm extended query table. The S25FL-S devices also contain additional information in the CFI alternate algorithm extended query table that is referenced by the alternate algorithm extended query table pointer.*

Since the device IDs are the same across different families, software should use the CFI information to detect device features and use the flash devices appropriately. The exception is the S25FL128P, where Byte 3 shows that only three bytes of extended information are present. This indicates that CFI is not supported.

```
If (slld_RDIDCmd (id_data_buffer, 6) == SLLD_OK)
{
    if ((id_data_buffer[3] < 0x4D) && (id_data_buffer[3] != 0))
    {
        // only for the S25FL128P
        cfi_supported = FALSE;
    }
    else
    {
        cfi_supported = TRUE;
    }
}
else
{
    // Handle ID read error
}
```

Below are differences in CFI information between S25FL-S and the previous generation.

Table 4.4 CFI Data Differences (Sheet 1 of 2)

CFI Register	Byte Address	FL129P	FL-S
Alternate OEM Command Set	17h	00h	53h
	18h	00h	46h
Address for Alternate OEM Extended Table	19h	00h	51h
	1Ah	00h	00h
Typical timeout per single byte program = 2^N µs	1Fh	0Bh	06h
Typical timeout for minimum size page program = 2^N µs (00h = Not Supported)	20h	0Bh	08h (256B page) 09h (512B page)
Typical timeout per individual sector erase = 2^N ms	21h	09h	08h (4 kB or 64 kB) 09h (256 kB)
Typical timeout for full chip erase = 2^N ms (00h = Not Supported)	22h	11h	0Fh (128 Mb) 10h (256 Mb) 11h (512 Mb)
Maximum timeout for byte program 2^N times typical	23h	01h	02h
Maximum timeout for page program 2^N times typical	24h	01h	02h
Maximum timeout per individual sector erase 2^N times typical	25h	02h	03h
Maximum timeout for full chip erase 2^N times typical (00h = not supported)	26h	01h	03h
Flash Device Interface Description 0000h = x8-only 0001h = x16-only 0002h = x8/x16 capable 0003h = x32 only 0004h = Single I/O SPI, 3-byte address 0005h = Multi I/O SPI, 3-byte address 0102h = Multi I/O SPI, 3 or 4 byte address	28h	05h	02h
	29h	05h	01h
Maximum number of bytes in multi-byte write = 2^N (0000 = not supported)	2Ah	08h	08h (256B page) 09h (512B page)
	2Bh	00h	00h
Address Sensitive Unlock (Bits 1-0) 00b = Required, 01b = Not Required Process Technology (Bits 5-2) 0000b = 0.23 µm Floating Gate 0001b = 0.17 µm Floating Gate 0010b = 0.23 µm MirrorBit 0011b = 0.11 µm Floating Gate 0100b = 0.11 µm MirrorBit 0101b = 0.09 µm MirrorBit 1000b = 0.065 µm MirrorBit	45h	15h	21h
Erase Suspend 00h = Not Supported 01h = Read Only 02h = Read and Program	46h	00h	02h
Sector Protect 00 = Not Supported X = Number of sectors in group	47h	04h	01h
Sector Protect/Unprotect Scheme 04h = High Voltage Method 05h = Software Command Locking Method 08h = Advanced Sector Protection Method 09h = Secure	49h	05h	08h
Page Mode Type 00h = Not Supported 01h = 4 Word Page 02h = 8 Word Page 03h = 256 Byte Program Page 04h = 512 Byte Program Page	4Ch	03h	03h (256B page) 04h (512B page)

Table 4.4 CFI Data Differences (Sheet 2 of 2)

CFI Register	Byte Address	FL129P	FL-S
ACC (Acceleration) Supply Minimum 00h = Not Supported D7-D4: Volt, D3-D0: 100 mV	4Dh	85h	00h
ACC (Acceleration) Supply Maximum 00h = Not Supported D7-D4: Volt, D3-D0: 100mV	4Eh	95h	00h
Program Suspend 00h = Not Supported 01h = Supported	50h	00h	01h
Query-unique ASCII string “ALT”	51h	—	41h
	52h	—	4Ch
	53h	—	54h
Major version number = 2, ASCII	54h	—	32h
Minor version number = 0, ASCII	55h	—	30h

There are no CFI entries for offsets above 50h in FL-P devices. In FL-S, this is the start of the Alternate Vendor-Specific Extended Query. In order to support FL-S devices, software should read the entire 512-byte ID-CFI space.

```
if (cfi_supported)
{
    cfi_length = 512;

    If (slld_RDIDCmd (id_data_buffer, cfi_length) != SLLD_OK)
    {
        // Handle CFI read error
    }
}
```

Software can check the Address for Alternate OEM Extended Table (CFI offsets 19h and 1Ah) for the start of CFI Alternate Vendor-Specific Extended Query. After the header for this table, the subsequent sections are organized into parameters of varying lengths. Each parameter starts with an identifier and size. Software can search the table for a certain parameter ID by checking each parameter and skipping to the next one, as needed. In the following example, the code is checking for four byte address instruction support.

```
int four_byte_addr_cmds = FALSE;

alt_vendor_query = (WORD)cfi_data_buffer[0x19];
alt_vendor_query |= (WORD)cfi_data_buffer[0x1A] << 8;

if (alt_vendor_query)// if offset 19h or 1Ah is non-zero
{
    if ((cfi_data_buffer[alt_vendor_query] != 'A')
        || (cfi_data_buffer[alt_vendor_query + 1] != 'L')
        || (cfi_data_buffer[alt_vendor_query + 2] != 'T'))
    {
        // handle error for invalid Alternate Vendor-Specific Extended Query header
    }

    index = alt_vendor_query + 5;// first parameter

    while (index < 512)
    {
        if (cfi_data_buffer[index] == 0x80)
        {
            // offset 2, bit 2 indicates 4 byte address support
            if (cfi_data_buffer[index + 2] & 0x4 == 0)
                four_byte_addr_cmds = TRUE;

            break;// we found the desired parameter
        }
        else
        {
            index++;// move from the parameter ID to the length byte

            // skip to the next parameter
            index += (cfi_data_buffer[index] + 1);
        }
    }
}
```

Refer to the S25FL-S data sheets for more information on ID and CFI data.

5. Bank Address Register

Standard SPI instructions transfer three bytes of address data, which is sufficient for densities up to 128 Mb (16 MB). Since the S25FL-S device family includes devices larger than 128 Mb, three bytes cannot address the entire memory space. The Bank Address Register (BAR) allows the new generation of Spansion SPI devices to support standard SPI commands and a wide range of SPI controllers. The BAR must be initialized, including the upper address bits, before issuing the standard SPI instructions.

The BAR can also be configured for the S25FL-S device to expect four bytes of address with standard SPI instructions. The Extended Address Enable field must be initialized before issuing the modified SPI instructions. This approach may not be compatible with certain SPI controllers. Consider the implementation of the SPI interface for your platform to determine the best solution to address the entire flash device.

The default value of the BAR sets upper address bits to 0 and disables the extended address feature. This allows standard operation on lower density S25FL-S devices. It also allows legacy software, firmware, and controllers the ability to access the first 128 Mb (16 MB) of memory on higher density devices.

Table 5.1 Bank Address Register

Bits	Field Name	Function	Default State	Description
7	EXTADD	Extended Address Enable	0b	1 = 4 byte (32-bits) addressing required from command 0 = 3 byte (24-bits) addressing from command
6 to 2	RFU	Reserved	00000b	Reserved for Future Use
1	BA25	Bank Address	0b	A25 for 512-Mbit device, RFU for lower density
0	BA24	Bank Address	0b	A24 for 256- and 512-Mbit device, RFU for lower density

The BAR is volatile and must be configured after each power on or reset. It can be read and written with new commands or legacy commands. In order to support legacy SPI controllers with minimal software changes, the BAR can be read using instruction B9h. This enables Deep Power Down mode for many SPI devices. Since the S25FL-S does not support the Deep Power Down feature, this instruction is used for accessing the BAR instead. When a Write Register command immediately follows a Bank Register Access command, the data will be written to the BAR instead of Status Register-1.

Table 5.2 Bank Address Register Commands

Instruction	Command Name	New Description	Legacy Description
B9h	BRAC	Bank Register Access	Deep Power-Down
01h	WRR	Write Register	Write Register
16h	BRRD	Bank Register Read	—
17h	BRWR	Bank Register Write	—

A third solution is a new set of commands that include four bytes of address and are not related to the BAR. This approach is discussed in Section 11.4, *New Commands with 4-byte Address* on page 12. The Bank Address Register is not planned to be supported in future device families, therefore the user is recommended to use the Extended Address method or the new four byte address instructions to minimize future migration issues. Refer to the S25FL-S data sheets for more detailed information on the Bank Address Register.

6. AutoBoot Register

The AutoBoot feature enables S25FL-S devices to output data immediately after power on reset, hardware reset, or software reset. This allows the host to access boot code without issuing a read instruction. The AutoBoot Register must be configured before using this feature. The register is non-volatile, and the feature is disabled by default from the factory. This allows backwards-compatibility with legacy software and firmware.

Table 6.1 AutoBoot Register

Bits	Field Name	Function	Default State	Description
31 to 9	ABSA	AutoBoot Start Address	0	512-byte boundary address for the start of boot code access
8 to 1	ABSD	AutoBoot Start Delay	00h	Number of initial delay cycles between CS# going low and the first bit of boot code being transferred
0	ABE	AutoBoot Enable	0b	1 = AutoBoot is enabled 0 = AutoBoot is not enabled

The AutoBoot Start Delay defines the number of clock cycles to wait after the first falling edge of CS# before driving data to the host. Data must be read from a 512-byte aligned memory location. The AutoBoot Register can be read and written with the new commands AutoBoot Read (ABRD 14h) and AutoBoot Write (ABWR 15h). Refer to the S25FL-S data sheets for more information on the AutoBoot Register.

7. One Time Program (OTP) Address Space

The S25FL-S family of devices offer a larger OTP area than FL-P devices. The FL-P OTP is organization into two 8-byte ESN regions, thirty 16-byte regions, and one 10-byte region. The FL-S OTP is organized into thirty-two regions of 32 bytes each. Each region is individually lockable for both device families.

Table 7.1 OTP Address Space

Bytes	FL129P (1)	FL-S
0 - 7	ESN1	ESN
8 - 15	ESN2 or customer use	
16 - 19	Customer use	OTP Lock Bytes
20 - 31		RFU (2)
32 - 47	Customer use	Customer use
48 - 63	Customer use	
64 - 79	Customer use	Customer use
80 - 95	Customer use	
96 - 495	…	…
496 - 506	Customer use	Customer use
506 - 527	N/A (3)	
528 - 559	N/A (3)	Customer use
560 - 591	N/A (3)	Customer use
592 - 991	N/A (3)	…
992 - 1023	N/A (3)	Customer use

Notes:

1. *S25FL128P does not include the OTP feature.*
2. *RFU = Reserved for Future Use.*
3. *N/A = Not Available*

Refer to the S25FL-S data sheets for more information on the OTP.

8. Suspend and Resume of Program and Erase

The S25FL-S devices support suspend and resume of program operations, as well as suspend and resume of erase operations. When the device is actively erasing, the host can issue the Erase Suspend (ERSP 75h) command to pause the operation. The device can accept read and/or program commands for sectors other than the erase-suspended sector. Once the erase can continue, the host should issue the Read Status Register 2 (RDSR2 07h) command to determine if the erase operation was suspended or completed. If the erase was suspended, the host must issue the Erase Resume (ERRS 7Ah) command to continue the operation.

When the device is actively programming, the host can issue the Program Suspend (PGSP 85h) command to pause the operation. The device can accept read commands for sectors other than the program-suspended sector. Once the program can continue, the host should issue the Read Status Register 2 (RDSR2 07h) command to determine if the program operation was suspended or completed. If the program was suspended, the host must issue the Program Resume (PGRS 8Ah) command to continue the operation.

Table 8.1 Status Register-2 (SR2)

Bits	Field Name	Function	Default State	Description
7 to 2	RFU	Reserved	000000b	Reserved for Future Use
1	ES	Erase Suspend	0b	1 = In erase suspend mode 0 = Not in erase suspend mode
0	PS	Program Suspend	0b	1 = In program suspend mode 0 = Not in program suspend mode

Status Register-2 cannot be modified directly. It is a read-only register. Refer to the S25FL-S data sheets for more information on Status Register-2.

9. Latency Code Configuration

In order to sustain read operations at high speeds, latency cycles are needed between the address and data transfer. Each read instruction for the previous generation of SPI devices required a certain number of mode and dummy cycles. S25FL-S devices provide a Latency Code feature that allows Fast, Dual, and Quad read commands to operate with different latency cycles at different clock speeds. Depending on the Latency Code chosen and the read command, the number of mode and dummy cycles can be determined. These settings also differ for High Performance versus Enhanced High Performance devices, which have separate ordering part numbers. Below is an example of the mode and dummy cycles for the Quad I/O Read (EBh) with an Enhanced High Performance device.

Table 9.1 Latency Cycles for Quad I/O Read (EBh)

Device	Frequency (MHz)	Latency Code	Mode Cycles	Dummy Cycles
S25FL-S Enhanced High Performance	≤ 50	11b	2	1
	≤ 80	00b	2	4
	≤ 90	01b	2	4
	≤ 104	10b	2	5
S25FL129P	≤ 80	—	2	4

Latency Code tables describing the mode and dummy cycles needed for each read command can be found in the S25FL-S data sheet and the CFI Vendor-Specific Extended Query. The Latency Code is a non-volatile field in Configuration Register 1. The default value is 00b.

During dummy cycles, the flash will ignore any data provided by the host. But when issuing Double Data Rate (DDR) read commands, the host must not drive data during the dummy cycles.   For DDR read commands, S25FL-S devices may optionally drive an 8 edge Data Learning Pattern (DLP) during the dummy cycles immediately before the start of data. The host controller can use the DLP to determine the phase shift from SCK to data edges in order to capture data at the center of the data eye. An LC that provides 5 or more dummy cycles should be selected to allow 1 cycle of additional time for the host to stop driving before the memory starts driving the 4 cycle DLP. For example, the Enhanced High Performance DLP for DDR is shown in Table 9.2.

Table 9.2 Data Learning Pattern for DDR Dual I/O Read (BDh)

Device	Frequency (MHz)	Latency Code	Mode Cycles	Dummy Cycles	Enough Dummy Cycles for DLP?
S25FL-S Enhanced High Performance	≤ 50	11b	2	2	No
	≤ 80	00b	2	4	No
	≤ 90	01b	2	5	Yes
	≤ 104	10b	2	6	Yes

Refer to the S25FL-S data sheets for more information on Latency Codes, Mode Bits, dummy cycles, and Data Learning Pattern.

10. Embedded Operation Timing

There are timing differences between S25FL-S devices and previous generations. Table 10.1 highlights timings that may impact software design. Many of these values are also accessible to software via the Common Flash Interface (CFI) as described in Section 4., *Device Identification and CFI* on page 3. Refer to the respective data sheets for full timing and CFI specifications.

Table 10.1 Embedded Operation Time Differences

Flash Operation	Typ, Min, or Max	FL128P	FL129P	FL-S	Units
WRR Write Time	Max	100	50	500	ms
Page Programming (256 bytes)	Typ	1200	1500	250	µs
	Max	3000	3000	750	µs
Page Programming (512 bytes)	Typ	—	—	340	µs
	Max	—	—	750	µs
Sector Erase (4 kB or 8 kB)	Typ	-	200	130	ms
	Max	-	800	650	ms
Sector Erase (64 kB physical)	Typ	500	500	130	ms
	Max	3000	2000	650	ms
Sector Erase (64 kB logical, 16 x 4 kB)	Typ	—	—	2080	ms
	Max	—	—	10,400	ms
Sector Erase (256 kB)	Typ	2000	2000	520	ms
	Max	12,000	8000	2600	ms
Erase Suspend	Max	—	—	45	µs
Program Suspend	Max	—	—	40	µs
Erase Resume to Next Erase Suspend	Min	—	—	100	µs
Program Resume to Next Program Suspend	Min	—	—	100	µs
Software or Hardware Reset t_{RPH}	Min	—	—	35	µs

Another timing difference between S25FL-S and S25FL-P devices is the maximum clock frequency, which varies for certain commands. Table 10.2 shows notable differences in the speeds.

Table 10.2 Maximum Clock Frequency (MHz) for Various Instructions

Instruction	FL128P	FL129P	FL-S
READ (03h)	40	40	50
RDID (9Fh)	40	50	133
RES (ABh)	104	104	50
Other Serial Instructions	104	104	133
Dual or Quad Read Instructions	—	80	104
Quad Page Program	—	80	80
DDR Instructions	—	—	66

Refer to the respective data sheets for full timing and CFI specifications.

11. Command Set Changes

11.1 Removed Deep Power-Down

S25FL-S devices do not support the deep power-down feature. The Deep Power-Down instruction (B9h) that is supported in S25FL-P devices is used for a different purpose (see Table 5.2, *Bank Address Register Commands* on page 8). The Release from Deep Power-Down and Read Electronic Signature instruction (ABh) is available for backwards-compatibility as the Read Electronic Signature instruction. The Read Identification command (9Fh) is recommended for new software. Refer to the S25FL-S data sheets for more information on supported commands.

11.2 Removed 8 kB Parameter Sector Erase (40h)

S25FL-S devices do not support the 8-kB Parameter Sector Erase (P8E) instruction (40h). This command erases two sequential 4-kB parameter sectors with a single command on S25FL129P devices. Similarly, the S25FL-S family supports the 4-kB Parameter Sector Erase (P4E) instruction (20h). This command erases a single parameter sector. Software and firmware using the P8E command needs to be modified to use the P4E command instead. Refer to the S25FL-S data sheets for more information on the Parameter Sector Erase command.

11.3 Removed Parallel Programming Mode

S25FL128P devices in the 16-pin SO package support Parallel Mode for programming 8 bits at once. Parallel Mode is not available with FL-S family devices, however Quad Page Program is supported.

11.4 New Commands with 4-byte Address

Standard SPI instructions transfer three bytes of address data, which is sufficient for densities up to 128 Mb (16 MB). Since the S25FL-S device family includes devices larger than 128 Mb, three bytes cannot address the entire memory space. Several new instructions require four cycles of address, which can address these larger SPI devices. Table 11.1 lists these new 4-byte address instructions.

Table 11.1 Four Address Cycle Instructions

Instruction Name	Description	Instruction Code
4FAST_READ	Read Fast (4-byte Address)	0Ch
4READ	Read (4-byte Address)	13h
4DOR	Read Dual Out (4-byte Address)	3Ch
4QOR	Read Quad Out (4-byte Address)	6Ch
4DIOR	Dual I/O Read (4-byte Address)	BCh
4QIOR	Quad I/O Read (4-byte Address)	ECh
4DDRFR	DDR Fast Read (4-byte Address)	0Eh
4DDRDIOR	DDR Dual I/O Read (4-byte Address)	BEh
4DDRQIOR	DDR Quad I/O Read (4-byte Address)	EEh
4PP	Page Program (4-byte Address)	12h
4QPP	Quad Page Program (4-byte Address)	34h
4P4E	Parameter 4 kB-sector Erase (4-byte Address)	21h
4SE	Erase 64 kB or 256 kB (4-byte Address)	DCh

Migrating to the new instructions will require software changes. Hardware changes may also be needed, since some SPI controllers may not support the new instructions. Alternate solutions for addressing upper memory are discussed in Section 5., *Bank Address Register* on page 7. Consider the implementation of the SPI interface for your platform to determine the best solution. Refer to the S25FL-S data sheets for more information on the 4-byte address commands.

11.5 New Double Data Rate (DDR) Read Commands

Spansion SPI flash memory support SPI clock modes 0 and 3. These devices latch inputs on the rising clock (SCK) edge and drive outputs to the host on the falling edge of the clock signal. The S25FL-S devices support new Double Data Rate (DDR) read commands. After latching the instruction on the rising clock edges, all subsequent command cycles use every clock edge. Address and mode bits are latched on the rising and falling clock edges. When enabled, the Data Learning Pattern is driven on both rising and falling clock edges during the dummy cycles. Data outputs are also driven on each clock edge.

Table 11.2 Double Data Rate Read Instructions

Instruction Name	Description	Instruction Code
DDRFR	DDR Fast Read (3- or 4-byte Address)	0Dh
4DDRFR	DDR Fast Read (4-byte Address)	0Eh
DDRDIOR	DDR Dual I/O Read (3- or 4-byte Address)	BDh
4DDRDIOR	DDR Dual I/O Read (4-byte Address)	BEh
DDRQIOR	DDR Quad I/O Read (3- or 4-byte Address)	EDh
4DDRQIOR	DDR Quad I/O Read (4-byte Address)	EEh

Refer to the S25FL-S data sheets for more information on DDR read commands.

11.6 New Advanced Sector Protection Features

S25FL-S devices introduce several sector protection features similar to Spansion's parallel NOR families. This group of features, called Advanced Sector Protection (ASP), allows software to prevent program and erase operations on the individual sectors. Each flash sector can be protected by a Persistent Protection Bit (PPB) or a Dynamic Protection Bit (DYB). DYBs can be cleared and set separately by the host any time the device is ready to receive commands. DYBs are volatile and default to unprotect at power-on reset. PPBs are non-volatile. They can be cleared (protect) individually, but they can only be set (unprotect) all together. The host can modify PPBs by issuing commands as long as they are not locked by the PPB Lock Bit.

All PPBs are locked in the current state when the host issues a PPB Lock Bit Write command, clearing the PPB Lock Bit to 0. The PPB Lock Bit cannot be modified again until power-on reset. There are two possible modes of behavior for the PPB Lock Bit. The Persistent Method automatically sets the PPB Lock Bit to 1 after power-on reset. The Password Method only sets the PPB Lock Bit if the correct password is written to the device. Since the password must be configured before the Password Method is enabled, the devices default to the Persistent Method. Flash devices ship with all sectors unprotected.

Table 11.3 Advanced Sector Protection Instructions

Instruction Name	Description	Instruction Code
ASPRD	ASP Read	2Bh
ASPP	ASP Program	2Fh
DYBRD	DYB Read	E0h
DYBWR	DYB Write	E1h
PPBRD	PPB Read	E2h
PPBP	PPB Program	E3h
PPBE	PPB Erase	E4h
PLBRD	PPB Lock Bit Read	A7h
PLBWR	PPB Lock Bit Write	A6h
PASSRD	Password Read	E7h
PASSP	Password Program	E8h
PASSU	Password Unlock	E9h

Refer to the S25FL-S data sheets for a full explanation of Advanced Sector Protection.

11.7 New Software Reset Command

S25FL-S devices introduce a new command called Software Reset, which was not supported in the S25FL-P family of devices. Software Reset restores the device to its original power-up state. However, the volatile register bits FREEZE and PPB Lock are not modified. Software Reset clears Erase Error and Programming Error bits in Status Register 1. The Block Protection Bits in Status Register 1 are reset if they are configured to be volatile in the BPNV bit of Configuration Register 1. Erase Suspend and Program Suspend bits in Status Register 2 and the Write Enable Latch and Write in Progress bits in Status Register 1 are cleared to 0 since any active or suspended operations are interrupted. The Bank Address Register is reset to 0, the Volatile Data Learning Register is reset, and all of the DYBs are reset to unprotect. The host must wait 35 µs between issuing the Software Reset Command and issuing further instructions to the flash.

Refer to the S25FL-S data sheets for more details on the Software Reset Command.

11.8 New Mode Bit Reset Command

S25FL-S devices introduce a new command called Mode Bit Reset (MBR), which was not supported in the S25FL-P family of devices. Mode bits are sent to the device as part of the command sequence for certain read commands (see Table 11.4).

Table 11.4 Read Commands with Mode Bits

Ordering Option	Instruction Description	Instruction Code
High Performance	Quad I/O Read	EBh, ECh
	Read DDR Quad I/O	EDh, EEh
Enhanced High Performance	Dual I/O Read	BBh, BCh
	Quad I/O Read	EBh, ECh
	DDR Fast Read	0Dh, 0Eh
	DDR Dual I/O Read	BDh, BEh
	Read DDR Quad I/O	EDh, EEh

Specifying a particular data pattern in the Mode bits causes the flash to expect address initially in the next read command sequence, skipping the instruction. Only address and Mode bits are received by the flash until a subsequent Mode bit pattern indicates the end of continuous reading. If a system supports multiple threads, interrupt handlers, or other modular software approaches where the host may access the flash without knowledge of the current state of the device, instructions could be misinterpreted. In these situations, software should issue the MBR Command before issuing further instructions.

Not all of the S25FL-S packages include a RESET# pin. For systems where the RESET# signal is not connected to the flash, the MBR command should be issued after system reset to ensure the flash is ready to respond to instructions from the host. Refer to the S25FL-S data sheets for more details on the Mode Bit Reset Command.

11.9 Read Serial Flash Discoverable Parameters Command

The S25FL512S device introduces a new command called Read Serial Flash Discoverable Parameters (RSFDP), which was not supported in the S25FL-P family, nor in lower-density S25FL-S devices. Before reading SFDP data, the host must issue the 5Ah instruction, followed by a 24-bit address of 000000h, and then eight dummy cycles.

SFDP is implemented according to the JEDEC JESD216 standard. It provides the host with device information similar to the Common Flash Interface. Software can determine if Read SFDP is supported by issuing the command and checking for the expected header signature.

```
sfdp_header_length = 0x18;

If ((slld_RSFDPCmd (sfdp_data_buffer, sfdp_header_length) == SLLD_OK)
        && (sfdp_data_buffer[0] == 'S')
        && (sfdp_data_buffer[1] == 'F')
        && (sfdp_data_buffer[2] == 'D')
        && (sfdp_data_buffer[3] == 'P'))
{
    sfdp_supported = TRUE;
}
else
{
    sfdp_supported = FALSE;
}
```

Software can check the SFDP header (offsets 14h, 15h, and 16h) for the start of the Spansion parameter table. After the header for this table, the subsequent sections are organized into parameters of varying lengths. Each parameter starts with an identifier and size. Software can search the table for a certain parameter ID by checking each parameter and skipping to the next one, as needed. In the following example, the code is checking for four byte address instruction support.

```
int four_byte_addr_cmds = FALSE;

// Read the Spansion parameter table length and address from the SFDP header
sfdp_parameter_length = sfdp_data_buffer[0x13];// in double words
sfdp_parameter_address = (DWORD)sfdp_data_buffer[0x14];
sfdp_parameter_address |= (DWORD)sfdp_data_buffer[0x15] << 8;
sfdp_parameter_address |= (DWORD)sfdp_data_buffer[0x16] << 16;

if (sfdp_parameter_address)// if offsets 13h, 14h, or 15h are non-zero
{
    // Convert double word offset to byte offset
    sfdp_parameter_address *= 4;

    // Convert double word length to byte size
    sfdp_parameter_table_end = sfdp_parameter_address + sfdp_parameter_length * 4;

    If (slld_RSFDPCmd (sfdp_data_buffer, sfdp_parameter_table_end) != SLLD_OK)
    {
        // handle Read SFDP command error
    }

    alt_vendor_query = (WORD)sfdp_data_buffer [sfdp_parameter_address + 0x19];
    alt_vendor_query |= (WORD)sfdp_data_buffer [sfdp_parameter_address + 0x1A] << 8;
    alt_vendor_query += sfdp_parameter_address;

    if ((sfdp_data_buffer[alt_vendor_query] != 'A')
        || (sfdp_data_buffer [alt_vendor_query + 1] != 'L')
        || (sfdp_data_buffer [alt_vendor_query + 2] != 'T'))
    {
        // handle error for invalid Alternate Vendor-Specific Extended Query header
    }

    index = alt_vendor_query + 5;// first parameter

    while (index < sfdp_parameter_table_end)
    {
        if (sfdp_data_buffer[index] == 0x80)
        {
            // offset 2, bit 2 indicates 4 byte address support
            if (sfdp_data_buffer[index + 2] & 0x4 == 0)
                four_byte_addr_cmds = TRUE;

            break;// we found the desired parameter
        }
        else
        {
            index++;// move from the parameter ID to the length byte

            // skip to the next parameter
            index += (sfdp_data_buffer[index] + 1);
        }
    }
}
```

Refer to the S25FL-S data sheets for more details on the Read SFDP Command.

12. Conclusion

The S25FL-S family offers an easy transition from FL-P devices. Some software changes may be needed to store data efficiently. New features give system designers more options and flexibility.

13. References

- S25FL128S and S25FL256S MirrorBit Flash Non-Volatile Memory Data Sheet
- S25FL512S MirrorBit Flash Non-Volatile Memory Data Sheet
- Spansion Application Note, Migration from FL-P to FL-S Family SPI

14. Revision History

Section	Description
Revision 01 (February 7, 2012)	
	Initial release

Colophon

The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for any use that includes fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for any use where chance of failure is intolerable (i.e., submersible repeater and artificial satellite). Please note that Spansion will not be liable to you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products. Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions. If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the US Export Administration Regulations or the applicable laws of any other country, the prior authorization by the respective government entity will be required for export of those products.

Trademarks and Notice

The contents of this document are subject to change without notice. This document may contain information on a Spansion product under development by Spansion. Spansion reserves the right to change or discontinue work on any product without notice. The information in this document is provided as is without warranty or guarantee of any kind as to its accuracy, completeness, operability, fitness for particular purpose, merchantability, non-infringement of third-party rights, or any other warranty, express, implied, or statutory. Spansion assumes no liability for any damages of any kind arising out of the use of the information in this document.