



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China





S26KL512S/S26KS512S
S26KL256S/S26KS256S
S26KL128S/S26KS128S

512 Mbit (64 Mbyte)/256 Mbit (32 Mbyte)/ 128 Mbit (16 Mbyte), 1.8V/3.0V HyperFlash™ Family

Features

- 3.0V I/O, 11 bus signals
 - Single ended clock
- 1.8V I/O, 12 bus signals
 - Differential clock (CK, CK#)
- Chip Select (CS#)
- 8-bit data bus (DQ[7:0])
- Read-Write Data Strobe (RWDS)
 - HyperFlash™ memories use RWDS only as a Read Data Strobe
- Up to 333 MB/s sustained read throughput
- Double-Data Rate (DDR) – two data transfers per clock
- 166-MHz clock rate (333 MB/s) at 1.8V V_{CC}
- 100-MHz clock rate (200 MB/s) at 3.0V V_{CC}
- 96-ns initial random read access time
 - Initial random access read latency: 5 to 16 clock cycles
- Sequential burst transactions
- Configurable Burst Characteristics
 - Wrapped burst lengths:
 - 16 bytes (8 clocks)
 - 32 bytes (16 clocks)
 - 64 bytes (32 clocks)
 - Linear burst
 - Hybrid option — one wrapped burst followed by linear burst
 - Wrapped or linear burst type selected in each transaction
 - Configurable output drive strength
- Low Power Modes
 - Active Clock Stop During Read: 12 mA, no wake-up required
 - Standby: 25 μ A (typical), no wake-up required
 - Deep Power-Down: 8 μ A (typical)
 - 300 μ s wake-up required
- INT# output to generate external interrupt
 - Busy to Ready Transition
 - ECC detection
- RSTO# output to generate system level power-on reset
 - User configurable RSTO# Low period
- 512-byte Program Buffer
- Sector Erase
 - Uniform 256-kB sectors
 - Optional Eight 4-kB Parameter Sectors (32 kB total)
- Advanced Sector Protection
 - Volatile and non-volatile protection methods for each sector
- Separate 1024-byte one-time program array
- Operating Temperature
 - Industrial (–40°C to +85°C)
 - Industrial Plus (–40°C to +105°C)
 - Extended (–40°C to +125°C)
 - Automotive, AEC-Q100 Grade 3 (–40°C to +85°C)
 - Automotive, AEC-Q100 Grade 2 (–40°C to +105°C)
 - Automotive, AEC-Q100 Grade 1 (–40°C to +125°C)
- ISO/TS16949 and AEC Q100 Certified
- Endurance
 - 100,000 program/erase cycles
- Retention
 - 20 year data retention
- Erase and Program Current
 - Max Peak \leq 100 mA
- Packaging Options
 - 24-Ball FBGA
- Additional Features
 - ECC 1-bit correction, 2-bit detection
 - CRC (Check-value Calculation)

Performance Summary

Read Access Timings	
Maximum Clock Rate at 1.8V V_{CC}/V_{CCQ}	166 MHz
Maximum Clock Rate at 3.0V V_{CC}/V_{CCQ}	100 MHz
Maximum Access Time, (t_{ACC})	96 ns
Maximum CS# Access Time to First Word @ 166 MHz	118 ns

Typical Program / Erase Times	
Single Word Programming (2B = 16b)	500 μ s (~4 kB/s)
Write Buffer Programming (512B = 4096b)	475 μ s (~1 MB/s)
Sector Erase Time (256 kB = 2 Mb)	930 ms (~282 kB/s)

Typical Current Consumption	
Burst Read (Continuous Read at 166 MHz)	80 mA
Power-On Reset	80 mA
Sector Erase Current	60 mA
Write Buffer Programming Current	60 mA
Standby (CS# = High)	25 μ A
Deep Power-Down (CS# = High, 85°C)	30 μ A (512 Mb)
	4 μ A (all other densities)

Contents

1. General Description	4	9.1 Endurance	70
1.1 DDR Center Aligned Read Strobe Functionality (DCARS).....	6	9.2 Data Retention.....	70
1.2 Error Detection and Correction Functionality	6	Hardware Interface	
2. Connection Diagram	9	10. Electrical Specifications	71
2.1 FBGA 24-Ball 5 x 5 Array Footprint	9	10.1 Absolute Maximum Ratings	71
3. Signal Description	10	10.2 Latchup Characteristics	72
4. HyperBus Protocol	11	10.3 Operating Ranges.....	72
4.1 Command / Address Bit Assignments	12	10.4 DC Characteristics (CMOS Compatible).....	73
4.2 Read Operations	12	10.5 Power-Up and Power-Down	75
4.3 HyperFlash Read with DCARS Timing	16	10.6 Power-Off with Hardware Data Protection.....	79
4.4 Write Operations	17	10.7 Power Conservation Modes.....	79
5. Address Space Maps	19	11. Timing Specifications	81
5.1 Flash Memory Array.....	20	11.1 AC Test Conditions.....	81
5.2 Device ID and CFI (ID-CFI) ASO	22	11.2 AC Characteristics	82
6. Embedded Operations	24	12. Embedded Algorithm Performance	88
6.1 Embedded Algorithm Controller (EAC)	24	13. Ordering Information	89
6.2 Program and Erase Summary	25	13.1 Ordering Part Numbers.....	89
6.3 Data Protection	49	13.2 Valid Combinations — Standard.....	90
7. Device ID and Common Flash Interface (ID-CFI) ASO Map	59	13.3 Valid Combinations — Automotive Grade / AEC-Q100	92
7.1 Device ID and Common Flash Interface (ID-CFI) ASO Map — Standard	59	14. Physical Interface	94
7.2 Device ID and Common Flash Interface (ID-CFI) ASO Map — Automotive Grade / AEC-Q100	64	14.1 Physical Diagram	94
8. Software Interface Reference	65	15. Document History Page	95
8.1 Command Summary	65	Sales, Solutions, and Legal Information	99
9. Data Integrity	70	Worldwide Sales and Design Support	99
		Products	99
		PSoC® Solutions	99
		Cypress Developer Community	99
		Technical Support	99

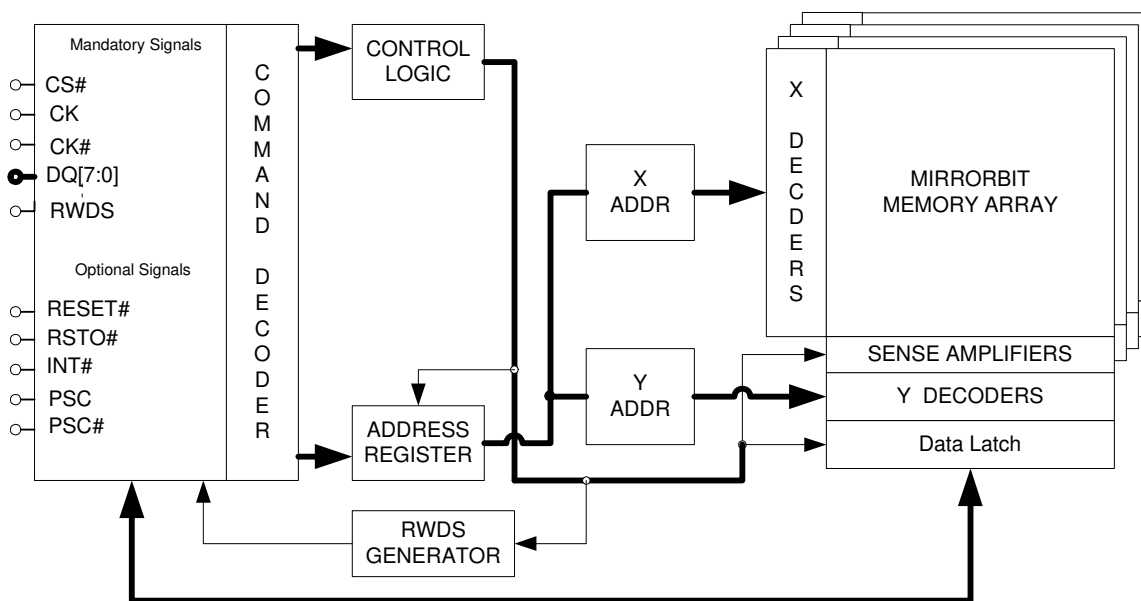
1. General Description

The Cypress HyperFlash family of products are high-speed CMOS, MirrorBit® NOR flash devices with the HyperBus low signal count DDR (Double Data Rate) interface, that achieves high speed read throughput. The DDR protocol transfers two data bytes per clock cycle on the data (DQ) signals. A read or write access for the HyperFlash consists of a series of 16-bit wide, one clock cycle data transfers at the internal HyperFlash core and two corresponding 8-bit wide, one-half-clock-cycle data transfers on the DQ signals.

Both data and command/address information are transferred in DDR fashion over the 8-bit data bus. The clock input signals are used for signal capture by the HyperFlash device when receiving command/address/data information on the DQ signals. The Read Data Strobe (RWDS) is an output from the HyperFlash device that indicates when data is being transferred from the memory to the host. RWDS is referenced to the rising and falling edges of CK during the data transfer portion of read operations. Command/address/write-data values are center aligned with the clock edges and read-data values are edge aligned with the transitions of RWDS.

Read and write operations to the HyperFlash device are burst oriented. Read transactions can be specified to use either a wrapped or linear burst. During wrapped operation, accesses start at a selected location and continue for a configured number of locations in a group wrap sequence. During linear operation accesses start at a selected location and continue in a sequential manner until the read operation is terminated, when CS# returns High. Write transactions transfer one or more 16-bit values.

Figure 1. Logic Block Diagram



The HyperFlash family consists of multiple densities, 1.8V/3.0V core and I/O, non-volatile, synchronous flash memory devices. These devices have an 8-bit (1-byte) wide DDR data bus and use only word-wide (16-bit data) address boundaries. Read operations provide 16 bits of data during each clock cycle (8 bits on each clock edge). Write operations take 16 bits of data from each clock cycle (8 bits on each clock edge).

Each random read accesses a 32-byte length and aligned set of data called a page. Each page consists of a pair of 16-byte aligned groups of array data called half-pages. Half-pages are aligned on 16-byte address boundaries. A read access requires two clock cycles to define the target half-page address and the burst type, then an additional initial latency. During the initial latency period the third clock cycle will specify the starting address within the target half-page. After the initial data value has been output, additional data can be read from the Page on subsequent clock cycles in either a wrapped or linear manner. When configured in linear burst mode, while a page is being burst out, the device will automatically fetch the next sequential page from the MirrorBit flash memory array. This simultaneous burst output while fetching from the array allows for a linear sequential burst operation that can provide a sustained output of 333 MB/s data rate (1-byte (8-bit data bus) * 2 (Data on both clock edges) * 166 MHz = 333 MB/s).

Table 1. S26KS Address Map

Type	Count	Addresses	Notes
Word Address within a half-page (16 byte)	8 (word addresses)	A2 – A0	16 bytes
Word Address within Write Buffer Line (512 byte)	256 (word addresses)	A7 – A0	512 bytes
Half-pages (16 bytes) within Erase Sector (256 kB)	8192 (half-pages)	A16 – A3	
Write Buffer Lines (512 bytes) within Erase Sector (256 kB)	512 (lines)	A16 – A8	
Total Number of Erase Sectors (256 kB)	256 (512 Mb) 128 (256 Mb) 64 (128 Mb)	Amax – A17	

The device control logic is subdivided into two parallel operating sections: the Host Interface Controller (HIC) and the Embedded Algorithm Controller (EAC). The HIC monitors signal levels on the device inputs and drives outputs as needed to complete read and write data transfers with the host system (HyperFlash master). The HIC delivers data from the currently entered address map on read transfers; places write transfer address and data information into the EAC command memory; notifies the EAC of power transition, and write transfers. The EAC looks in the command memory, after a write transfer, for legal command sequences and performs the related Embedded Algorithms.

Changing the non-volatile data in the memory array requires a complex sequence of operations that are called Embedded Algorithms (EA). The algorithms are managed entirely by the internal EAC. The main algorithms perform programming and erase of the main flash array data. The host system writes command codes to the flash device address space. The EAC receives the command, performs all the necessary steps to complete the command, and provides status information during the progress of an EA.

The erased state of each memory bit is a logic 1. Programming changes a logic 1 (High) to a logic 0 (Low). Only an erase operation is able to change a 0 to a 1. An erase operation must be performed on an entire 256-kbyte (or 4-kbyte for parameter sectors) aligned group of data called a Sector. When shipped from Cypress all Sectors are erased.

Programming is done via a 512-byte Write Buffer. It is possible to write from one to 256 words, anywhere within the Write Buffer before starting a programming operation. Within the flash memory array, each 512-byte aligned group of data is called a Line. A programming operation transfers data from the volatile Write Buffer to a non-volatile memory array Line. The operation is called Write Buffer Programming.

The Write Buffer is filled with 1s after reset or the completion of any operation using the Write Buffer. Any locations not written to a 0 by a Write to Buffer command are by default still filled with 1s. Any 1s in the Write Buffer do not affect data in the memory array during a programming operation.

In addition to the mandatory signals (CS#, CK, CK#, DQ [7:0], RWDS) dedicated to the HyperBus, the device also includes optional signals (RESET#, INT#, RSTO#, and Phase Shifted clocks PSC/PSC#).

When RESET# transitions from Low to High the device returns to the default state that occurs after an internal Power-On Reset (POR).

The INT# output can provide an interrupt to the HyperFlash master to indicate when the HyperFlash transitions from busy to ready at the end of a program or erase operation.

The RSTO# is an open-drain output used to indicate when a POR is occurring within the device and can be used as a system level reset signal. Upon completion of the internal POR the RSTO# signal will transition from Low to high impedance after a user defined timeout period has expired. Upon transition to the high impedance state the external pull-up resistance will pull RSTO# High and the device immediately is placed into the Standby state.

PSC/PSC# are differential Phase Shifted Clock inputs used as a reference for RWDS edges instead of CK/CK#. Refer to [Section 1.1, DDR Center Aligned Read Strobe Functionality \(DCARS\) on page 6](#) for more details.

1.1 DDR Center Aligned Read Strobe Functionality (DCARS)

The HyperFlash memories offer a configurable feature that enables independent skewing (phase shifting) of the RWDS signal with respect to the read data outputs.

When the DDR Center Aligned Read Strobe (DCARS) feature is enabled, a second differential Phase Shifted Clock input PSC/PSC# is used as the reference for RWDS edges instead of CK/CK#. The second clock is generally a copy of CK/CK# that is phase shifted 90 degrees to place the RWDS edges centered within the DQ signals valid data window. However, other degrees of phase shift between CK/CK# and PSC/PSC# may be used to optimize the position of RWDS edges within the DQ signals valid data window so that RWDS provides the desired amount of data setup and hold time in relation to RWDS edges.

PSC/PSC# is not used during a write transaction. PSC and PSC# may be driven Low and High respectively or, both may be driven Low during write transactions.

1.2 Error Detection and Correction Functionality

1.2.1 Error Correction Code

HyperFlash memories provide embedded Hamming Error Correction Code (ECC) generation during Flash memory array programming, with error detection and correction during read.

As each 16-byte aligned half-page of data, loaded into the Write Buffer, is transferred to the 512-byte flash memory array Line, an Error Correction Code (ECC) for each Half-page ECC unit is also programmed in to a portion of the memory array not visible to the host system software.

The ECC information is checked during each Half-page Flash array read operation. Any one bit error within the Half-page will be corrected by the ECC logic during the access of each Half-page.

The ECC information for each Half-page can be written once after each erase of the sector containing each Half-page. Programming within the same Half-page more than once will disable error detection and correction within that Half-page.

Word Programming and Write Buffer Programming, more than once within a Half-page, is supported for legacy software compatibility. However, for the best data integrity, it is recommended to not use Word Programming or Write Buffer Programming to program within a Half-page, more than once. Multiple writes to the same half page without an erase will disable the ECC functionality since the ECC syndrome becomes invalid. For applications requiring multiple programming operations within the same Half-page, it is recommended to add system software Error Detection and Correction, to enhance the data integrity of Half-pages that are programmed more than once.

There is a mode that may be enabled for two bit error detection. When this mode is enabled, any one bit error in a Half-page is corrected and any two bit error is detected and reported. In this mode, the ability to write to the same half-page more than once, after an erase, is disabled. In this mode, attempting to program more than once in the same Half-page will result in programming operation failure status.

ECC errors may be detected by reading an ECC status register, enabling an interrupt, or enabling the RWDS to stop when an uncorrectable error is encountered - to create a bus error before data is transferred to the HyperBus master.

A register is provided to capture the address location of the ECC error.

A counter is provided to count ECC corrections or uncorrectable errors.

1.2.2 Cyclic Redundancy Check

A group of commands are provided to perform a hardware accelerated Cyclic Redundancy Check (CRC) calculation over a user defined address range. The calculation is another type of embedded operation similar to programming or erase, in which the device is busy while the calculation is in progress. The CRC operation uses a 32-bit polynomial able to detect up to a 32-bit long group of error bits.

A command is used to enter the CRC Address Space Overlay (ASO) where the desired address range is loaded to start the CRC calculation. While entered in the CRC ASO the status of the CRC operation may be checked, suspended to read from the memory array, resumed, and the resulting check-value read. See [Address Space Maps on page 19](#).

1.2.2.1 CRC Check-Value Calculation

The Check-value Calculation command sequence causes the device to perform a Cyclic Redundancy Check (CRC) calculation over a user defined address range. The CRC calculation is achieved with the polynomial described in [Figure 2](#).

The Check-value generation sequence is started by entering the CRC ASO. The next step is to load the beginning address into the CRC Start Address Register identifying the beginning of the address range that will be covered by the CRC calculation. Next the ending address is loaded into the CRC End Address Register, this step starts the CRC calculation. The CRC process calculates the Check-value on the data contained at the starting address through the ending address.

During the calculation period the device goes into the Busy state (SR[7] = 0). Once the Check-value Calculation has completed the device returns to the Ready state (SR[7] = 1) and the calculated Check-value is available in the Check-value Low Result Register and the Check-value High Result Register. The Check-value Low Result Register contains Check-value bits 0-15 and can be read from address 0 while the device is in the CRC ASO. The Check-value High Result Register contains bits 16-31 and can be read from address 1 while the device is in the CRC ASO. The Check-value Low Result Register and the Check-value High Result Register are loaded with 0s once the CRC calculation process is initiated.

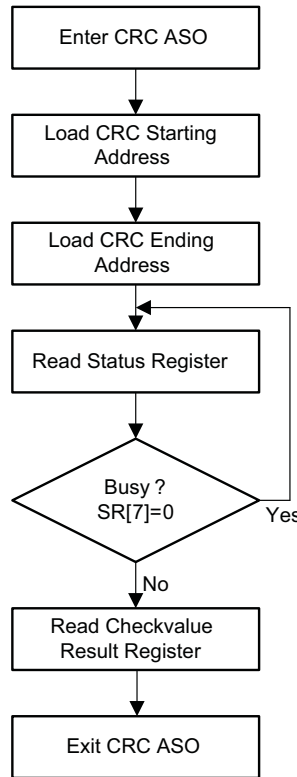
The Check-value Calculation can only be initiated when the device is in Standby State and once started can be suspended with the CRC Suspend sequence to read data from the array. During the Suspended state the CRC Suspend Status Bit (CRCSSB) in the Status Register will be set (SR[8] = 1). Once suspended, the host can read the Status Register, read data from the array and can resume the CRC calculation by using the CRC Resume command sequence. Once initiated the CRC ASO can be terminated with the ASO Exit Command or a Hardware Reset to return the device to Read Array Mode. The Check-value Calculation can not be performed while another ASO is active. A Hardware Reset will clear the value in the CRC Start Address Register, CRC End Address Register, Check-value High Result Register and the Check-value Low Result Register.

The Ending Address (EA) should be at least two addresses higher than the Starting Address (SA). If $EA < SA + 2$ the Check-value Calculation will abort and the device will return to the Ready state (SR[7] = 1). SR[3] will be set (1) to indicate the aborted condition. If $EA < SA + 2$ the Check-value High Result Register and the Check-value Low Result Register will hold indeterminate data.

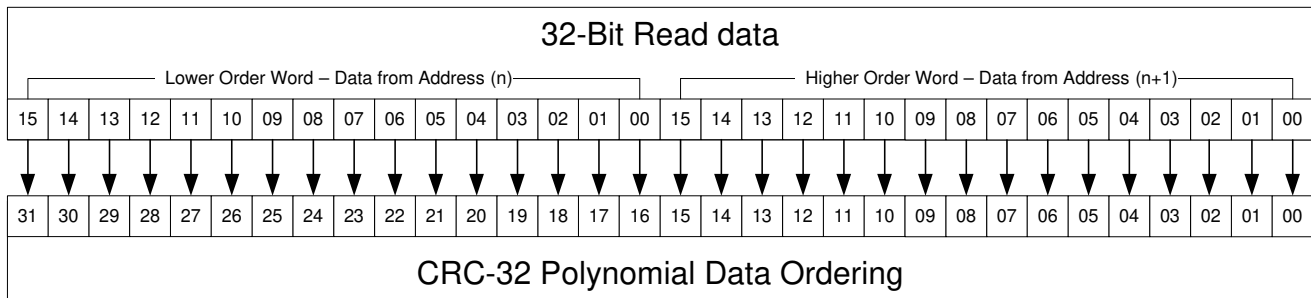
Figure 2. CRC-32 Polynomial

$$\text{CRC-32C Polynomial} = X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$$

Figure 3. Check-Value Calculation Sequence



The read data ordering used in calculating the check-value from the CRC-32 polynomial is shown as follows:

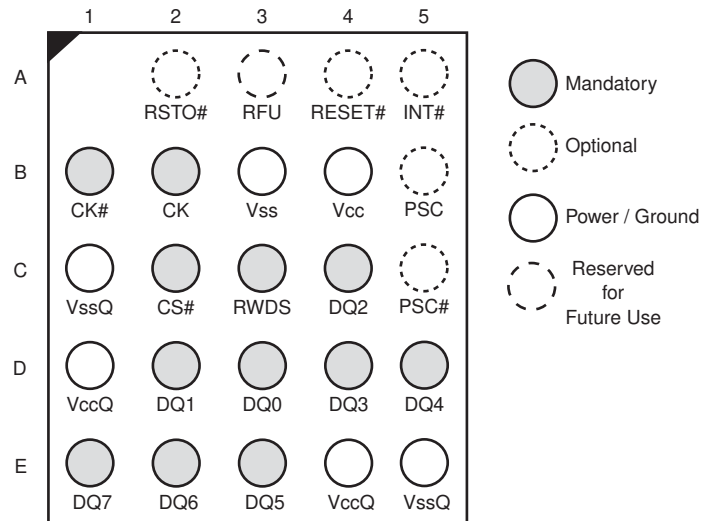


2. Connection Diagram

2.1 FBGA 24-Ball 5 x 5 Array Footprint

HyperFlash devices are provided in Fortified Ball Grid Array (FBGA), 1 mm pitch, 24-ball, 5 x 5 ball array footprint, with 6mm x 8mm body. The package height is device dependent and may be either 1 mm or 1.2 mm. See [Ordering Information on page 89](#). Consult the device data sheet Ordering Part Number valid combinations section for the package in use.

Figure 4. 24-Ball FBGA, 6 x 8 mm, 5x5 Ball Footprint, Top View



Notes:

1. B1 (CK#) is RFU on the 3.0V device (model 02).
2. B5 (PSC) and C5 (PSC#) are RFU on standard 3.0V and 1.8V devices (model 02). C5 (PSC#) is RFU on 3V DCARS device (model 03).
3. RFU — Reserved for Future Use. Not currently connected internally but the pin/ball location should be left unconnected and unused by PCB routing channel for future compatibility. The pin/ball may be used by a signal in the future.

3. Signal Description

Figure 5. HyperFlash Interface

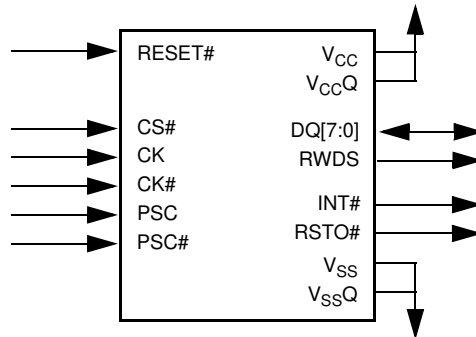


Table 2. Signal Descriptions

Symbol	Type	M / O	Description
CS#	Input	M	Chip Select. HyperFlash bus transactions are initiated with a High to Low transition. HyperFlash bus transactions are terminated with a Low to High transition.
CK, CK#	Input	M	Differential Clock. Command / Address / Data information is input or output with respect to the crossing of the CK and CK# signals. CK# is only used on the 1.8V devices and may be left open or connected to CK on 3V devices.
RWDS	Output	M	Read Write Data Strobe. Output data during read transactions are edge aligned with RWDS.
DQ[7..0]	Input / Output	M	Data Input / Output. Command / Address / Data information is transferred on these DQs during read and write transactions.
PSC, PSC#	Input	O	Phase Shifted Clock. PSC/PSC# allows independent skewing of the RWDS signal with respect to the CK/CK# inputs. PSC# is only used on the 1.8V device. PSC and PSC# may be driven High and Low respectively or both may be driven Low during write transactions.
INT#	Output (open drain)	O	INT Output. When Low, the device is indicating that an internal event has occurred. This signal is intended to be used as a system level interrupt for the device to indicate that an on-chip event has occurred. INT# is an open-drain output.
RESET#	Input	O	Hardware Reset. When Low, the device will self initialize and return to the array read state. RWDS and DQ[7:0] are placed into the High-Z state when RESET# is Low. RESET# includes a weak pull-up, if RESET# is left unconnected it will be pulled up to the High state.
RSTO#	Output (open drain)	O	RSTO# Output. RSTO# is an open-drain output used to indicate when a POR is occurring within the device and can be used as a system level reset signal. Upon completion of the internal POR the RSTO# signal will transition from Low to high impedance after a user defined timeout period has elapsed. Upon transition to the high impedance state the external pull-up resistance will pull RSTO# High and the device immediately is placed into the Standby state.
V _{CC}	Power Supply	P/G	Power.
V _{CCQ}	Power Supply	P/G	Input / Output Power.
V _{SS}	Power Supply	P/G	Ground.
V _{SSQ}	Power Supply	P/G	Input / Output Ground.

Note:

1. M = Mandatory; O = Optional; P/G = Power / Ground

4. HyperBus Protocol

All bus transactions can be classified as either read or write. A bus transaction is started with CS# going Low with CK = Low and CK# = High. The transaction to be performed is presented to the HyperFlash device during the first three clock cycles in a DDR manner using all six clock edges. These first three clocks transfer three words of Command / Address (CA0, CA1, CA2) information to define the transaction characteristics:

- Read or write transaction.
- Whether the transaction will be to the memory array or to register space.
 - Although the HyperBus protocol provides for slave devices that have both memory and register address spaces, HyperFlash memories described in this specification do not differentiate between memory and registers as separate address spaces. There is a single address space selected by any transaction, independent of whether the transaction indicates the target location is in memory space or register space. Write transactions always place the transaction address and data into a command register set (buffer). Read transactions return data from the memory array or from a register address space window that has been temporarily overlaid within the single address space by the execution of commands. The single address space with register space overlays methodology is backward compatible with legacy parallel NOR Flash memory program and erase software drivers.
- Whether a transaction will use a linear or wrapped burst sequence.
 - HyperFlash write transactions do not support burst sequence and ignore the burst type indication. Write command transactions transfer a single word per write. Only the Word Program command write data transfer may be done with a linear burst at up to 50 MHz.
- The target half-page address (row and upper order column address).
- The target Word (within half-page) address (lower order column address).

Once the transaction has been defined, a number of idle clock cycles are used to satisfy any read latency requirements before data is transferred. Once the target data has been transferred the HyperBus master host completes the transaction by driving CS# High with CK = Low and CK# = High. Data is transferred as 16-bit values with the first eight bits (15-8) transferred on a High going CK (write data or CA bits) or RWDS edge (read data) and the second eight bits (7-0) being transferred on the Low going CK or RWDS edge. Data transfers during read or write operations can be ended at any time by bringing CS# High when CK = Low and CK# = High. Read data is edge aligned with RWDS transitions and Write data is center aligned with clock edges.

4.1 Command / Address Bit Assignments

Table 3. Command / Address Bit Assignments

CA Bit#	Bit Name	Bit Function
47	R/W#	Identifies the transaction as a Read or Write. 1 = Read operation 0 = Write operation Target space is defined in CA46.
46	Target	Indicates whether the Read or Write operation accesses the memory or register spaces. 0 = memory space 1 = register space The register space is intended to be used by volatile memory and peripheral devices. The HyperFlash devices will not take advantage of this feature and this bit should be set to 0 during Read or Write transactions.
45	Burst Type	Indicates whether the burst will be linear or wrapped. 0 = Wrapped Burst 1 = Linear Burst
44-39 (1 Gb) 44-38 (512 Mb)	Reserved	Reserved for future address expansion. Reserved bits should be set to 0 by the host controller.
38-16 (1 Gb) 37-16 (512 Mb)	Row and Upper Column Address	Half page component of target address.
15-3	Reserved	Reserved for future column address expansion. Reserved bits should be set to 0 by the host controller.
2-0	Lower Column Address	Lower Column component of the target address: System word address bits A2-0 selecting the starting word within a half-page.

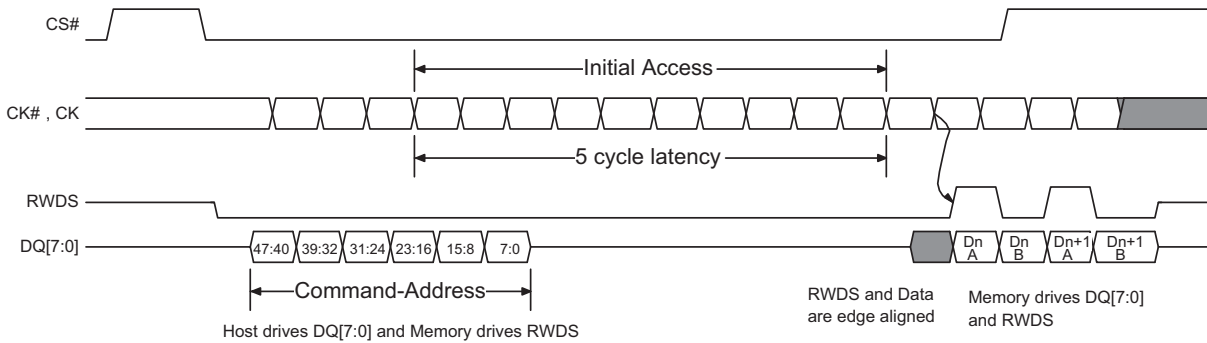
4.2 Read Operations

CA0 indicates that a read operation is to be performed and also indicates the burst type (wrapped or linear). Read operations begin the internal array access as soon as the half-page address has been presented in CA0 and CA1. CA2 identifies the target word address within the chosen half-page. The host then continues clocking for a number of cycles defined by the latency count setting in the Configuration Register. Once these latency clocks have been completed the memory starts to simultaneously transition the Read Write Data Strobe (RWDS) and begins outputting the target data. New data is output in an edge aligned fashion upon every transition of RWDS. Data will continue to be output as long as the host continues to transition the clock (CK and CK#). Wrapped bursts will continue to wrap within the burst length and linear burst will output data in a sequential manner across page boundaries. A Hybrid Burst provides one initial wrapped burst followed by linear burst, as described in [Section 6.3.11, Hybrid Burst on page 56](#). Wrapped reads can be performed from the main array, the CFI Tables and the Secure Silicon Region. Read transfers can be ended at any time by bringing CS# High when CK = Low and CK# = High.

When a linear burst reaches the last address in the array, if the burst continues, the address counter will wrap around and roll back to address 000000h, allowing the read sequence to be continued indefinitely. The entire memory can therefore be read out with one single read instruction.

The 16-byte and 32-byte wrapped bursts do not cross page boundaries and do not incur inter-page boundary crossing latencies. For a 64-byte wrapped burst read, a latency may occur during the target address to next page boundary crossing, depending on the starting address ([See 64-byte Wrapped Burst Address Sequence \(Latency Code = 16\) on page 40.](#))

Figure 6. Read Operation



Notes:

1. Transactions must be initiated with CK = Low and CK# = High. CS# must return High before a new transaction is initiated.
2. Read access from the flash array starts once CA[23:16] is captured.
3. The read latency is defined by the Read Latency value in the Volatile Configuration Register (or the Non-Volatile Configuration Register).
4. In this example of a read operation the Latency Count was set to five clocks.

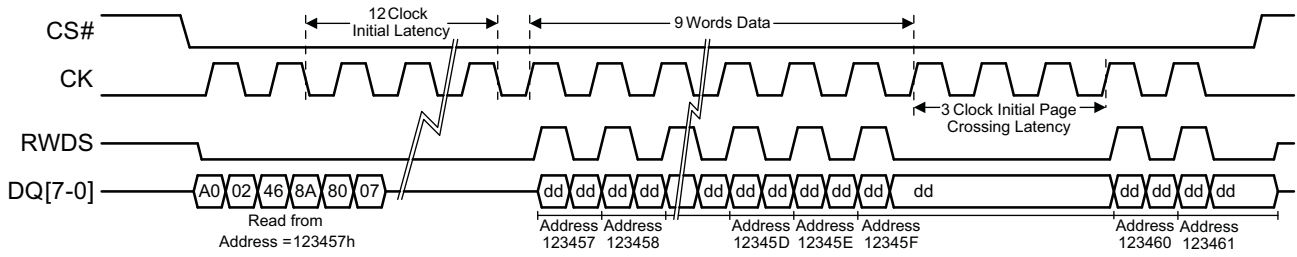
Table 4. Maximum Operating Frequency for Latency Code Options

Latency Code	Latency Clocks	Maximum Operating Frequency (MHz)
0000	5	52
0001	6	62
0010	7	72
0011	8	83
0100	9	93
0101	10	104
0110	11	114
0111	12	125
1000	13	135
1001	14	145
1010	15	156
1011	16	166
1100	Reserved	NA
1101	Reserved	NA
1110	Reserved	NA
1111	Reserved	NA

Notes:

1. Default NVCR latency setting when the device is shipped from the factory is 16 clocks.
2. The Latency Code is the value loaded into (Non) Volatile Configuration Register bits xVCR[7:4].
3. Maximum Operating Frequency assumed to be using a device with $t_{ACC} = 96$ ns.

Figure 7. Read Transaction Crossing a Page Boundary



Notes:

1. Read operation starting at device address 123457h.
2. Latency code loaded into the Configuration Register is 0111b which results in 12 latency clocks.
3. Page boundary crossing requires three clocks in this case. 12 clock initial latency minus 9 clocks (words) of initial data.
4. CK# is not shown but is the complement of the CK signal.
5. CA45 = 1 for a linear Read burst.

Table 5. First Page Boundary Crossing During Linear Read (Latency Count = 11 Clocks)

Target Address	Clock Cycle																													
	0	1	2	3	...	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30						
0	CA0	CA1	CA2	Bus Turnaround + Initial Latency		D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17							
1					D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18								
2					D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19								
3					D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20								
4					D4	D6	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21								
5					D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D22								
6					D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	X	D16	D17	D18	D19	D20	D21	D22								
7					D7	D8	D9	D10	D11	D12	D13	D14	D15	X	X	D16	D17	D18	D19	D20	D21	D22								
8					D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	D24	D25								
9					D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	D24	D25	D26								
10					D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	D24	D25	D26	D27								
11					D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	D24	D25	D26	D27	D28								
12					D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	D24	D25	D26	D27	D28	D29								
13					D13	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	D24	D25	D26	D27	D28	D29	D30								
14					D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	X	D24	D25	D26	D27	D28	D29	D30								
15					D15	D16	D17	D18	D19	D20	D21	D22	D23	X	X	D24	D25	D26	D27	D28	D29	D30								
16					D16	D17	D18	D19	D20	D21	D22	D23	D24	D25	D26	D27	D28	D29	D30	D31	D32	D33								
	—	—	1	2	...	11	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—						
Latency Count																														

Table 6. First Page Boundary Crossing During Linear Read (Latency Count = 16 Clocks)

Target Address	Clock Cycle After CS# Goes Low																																		
	0	1	2	3	...	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35											
0	CA0	CA1	CA2	Bus Turnaround + Initial Latency		D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17												
1					D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	X	D16	D17													
2					D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	X	X	D16	D17													
3					D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	X	X	X	D16	D17													
4					D4	D6	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	X	X	X	X	D16	D17													
5					D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	X	X	X	X	X	D16	D17													
6					D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	X	X	X	X	X	X	D16	D17													
7					D7	D8	D9	D10	D11	D12	D13	D14	D15	X	X	X	X	X	X	X	D16	D17													
8					D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	D24	D25													
9					D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	X	D24	D25													
10					D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	X	X	D24	D25													
11					D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	X	X	X	D24	D25													
12					D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	X	X	X	X	D24	D25													
13					D13	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	X	X	X	X	X	D24	D25													
14					D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	X	X	X	X	X	X	D24	D25													
15					D15	D16	D17	D18	D19	D20	D21	D22	D23	X	X	X	X	X	X	X	D24	D25													
16					D16	D17	D18	D19	D20	D21	D22	D23	D24	D25	D26	D27	D28	D29	D30	D31	D32	D33													
	—	—	1	2	...	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—											
Latency Count																																			

To calculate latency when crossing a page boundary, use the following formula:

```

if ((PS - LTCY) < ADDR & (SP - 1))
{
  ((ADDR & (SP - 1)) - PS + LTCY)
}
else
{0}

```

where:

PS = page size = 16 words

SP = sub-page size = 8 words

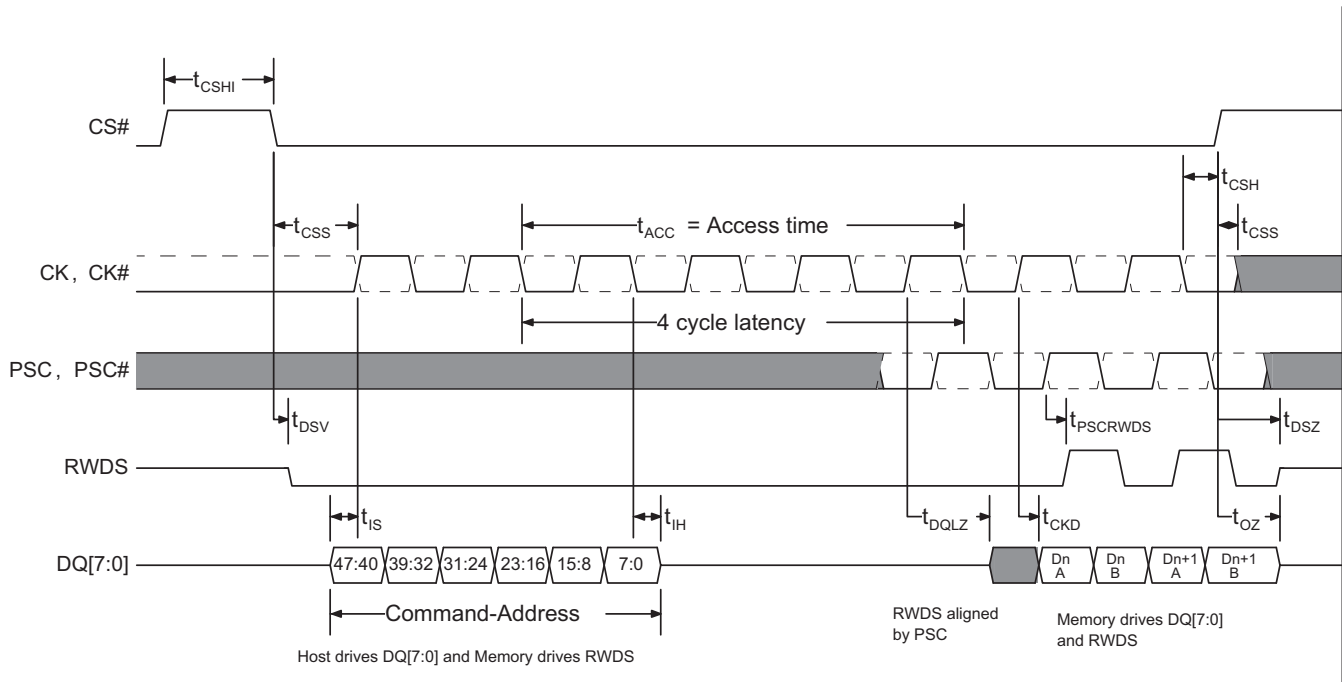
LTCY = latency

ADDR = target address

4.3 HyperFlash Read with DCARS Timing

The illustrations and parameters in this section are only those needed to define the DCARS feature and show the relationship between the Phase Shifted Clock, RWDS, and data.

Figure 8. HyperFlash Read DCARS Timing Diagram



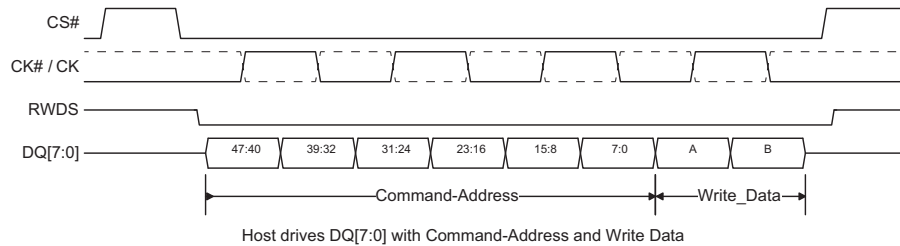
Notes:

1. Transactions must be initiated with CK = Low and CK# = High. CS# must return High before a new transaction is initiated.
2. CK# and PSC# are optional and shown as dashed line waveforms.
3. The memory drives RWDS during read transactions.
4. This example demonstrates a latency code setting of four clocks and no additional initial latency required.

4.4 Write Operations

A write operation starts with the first three clock cycles providing the CAx (Command / Address) information indicating the transaction characteristics. The Burst Type bit CA[45] is 'don't care' because the HyperFlash device only supports a single write transaction of 16b or a continuous linear write burst that is only supported when loading data during a Word Program command. Immediately following the CA information the host is able to transfer the write data on the DQ bus. The first byte (A) of data is presented on the rising edge of CK and the second byte (B) is presented on the falling edge of CK. Write data is center aligned with the CK/CK# inputs. Write transfers can be ended at any time by bringing CS# High when CK = Low and CK# = High.

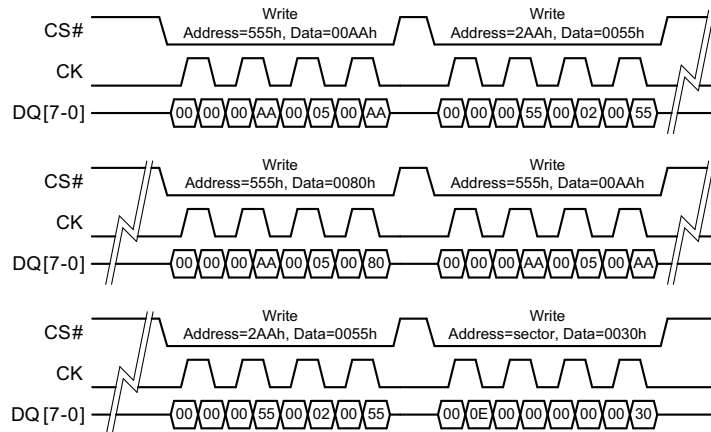
Figure 9. Write Operation



Notes:

1. Transactions must be initiated with CK = Low and CK# = High. CS# must return High before a new transaction is initiated.
2. RWDS will be driven Low as long as CS# is Low.
3. Write operations are limited to a transaction of a single word (16b) or a linear write burst supported only when loading data during a Word Program command.

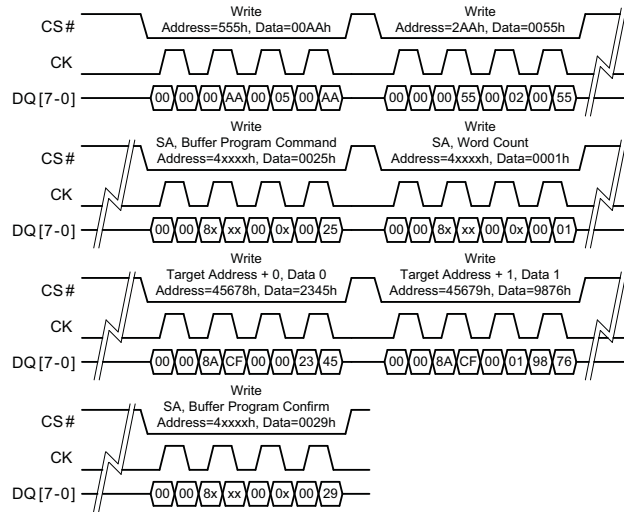
Figure 10. Write Transaction Usage Example: Erase Operation Command Sequence



Notes:

1. See Figure 16, Sector Erase Operation on page 33 for the Erase Operation Command Sequence flowchart.
2. Erase operation to the sector starting at 0700000h.
3. CK# is not shown but is the complement of the CK signal.
4. RWDS is not shown and is not used during Write transactions.

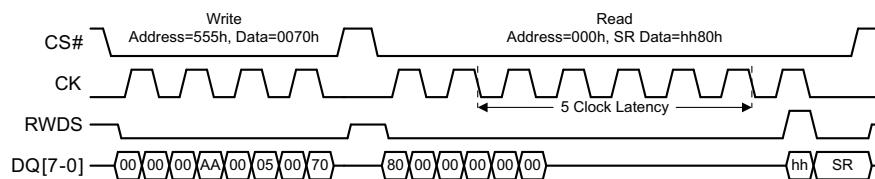
Figure 11. Write Transaction Usage Example: Write Buffer Program Command Sequence



Notes:

1. See [Figure 14, Write Buffer Programming Operation with Status Register](#) on page 29 for the Write Buffer Program Operation Command Sequence flowchart.
2. Program 2345h into address 45678h, and 9876h into address 45679h.
3. CK# is not shown but is the complement of the CK signal.
4. RWDS is not shown and is not used during Write transactions.

Figure 12. Status Read Transaction Example



Note:

1. CK# is not shown but is the complement of the CK signal.

5. Address Space Maps

Although the HyperBus protocol provides for slave devices that have both memory and register address spaces, HyperFlash memories described in this specification do not differentiate between memory and registers as separate address spaces. There is a single address space selected by any transaction, independent of whether the HyperBus transaction indicates the target location is in memory space or register space of the selected device.

Write transactions always place the transaction address and data into a a command register set (buffer).

Read transactions return data from the memory array or from a register address space window that has been temporarily overlaid within the single address space by the execution of commands. The single address range with register space overlays methodology is backward compatible with legacy parallel NOR Flash memory program and erase software drivers.

There are several separate address spaces that may appear within the address range of the flash memory device. One address space is visible (entered) at any given time.

- **Flash Memory Array:** the main non-volatile memory array used for storage of data that may be randomly accessed by read operations.
- **ID/CFI:** a flash memory array used for Cypress factory programmed device characteristics information. This area contains the Device Identification (ID) and Common Flash Interface (CFI) information tables.
- **Secure Silicon Region (SSR):** a 1024-byte one-time programmable non-volatile memory array used for Cypress factory programmed permanent data, and customer programmable permanent data.
- **Persistent Protection Bits (PPB):** a non-volatile memory array with one bit for each Sector. When programmed, each bit protects the related Sector from erasure and programming.
- **PPB Lock Bit:** a volatile register bit used to enable or disable programming and erase of the PPB bits.
- **Password:** an OTP non-volatile array used to store a 64-bit password used to enable changing the state of the PPB Lock Bit when using Password Mode Sector Protection.
- **Dynamic Protection Bits (DYB):** a volatile array with one bit for each Sector. When set, each bit protects the related Sector from erasure and programming.
- **ECC Status:** read the address of ECC corrected data and total ECC error count.
- **CRC:** read the CRC Check-value.
- **Status or Peripheral Registers:** register access used to display Embedded Algorithm status and read or write other registers.

The Flash Memory Array is the primary and default address space but, it may be overlaid by one other address space, at any one time. Each alternate address space is called an Address Space Overlay (ASO).

Each ASO replaces (overlays) either the sector selected by the command that enters the ASO or the entire flash device address range, depending on the ASO entry command. If only one sector is overlaid by an ASO the remaining sectors of the memory array remain readable. Any address range not defined by a particular ASO address map, is reserved for future use. Unless otherwise stated all read accesses outside of an ASO address map returns non-valid (undefined) data. The locations will display actively driven data but their meaning is not defined.

There are multiple address map modes that determine what appears in the flash device address space at any given time:

- **Read Mode**
- **Status Register (SR) Mode**
- **Address Space Overlay (ASO) Mode**
- **Peripheral Register Mode**

In Read Mode the entire Flash Memory Array may be directly read by the host system memory controller. The memory device Embedded Algorithm Controller (EAC), puts the device in Read Mode during Power-On, after a Hardware Reset, after a Command Reset, or after an Embedded Algorithm (EA) is suspended. Read accesses and commands are accepted in Read Mode. A subset of commands is accepted in Read Mode when an EA is suspended.

While in any mode, the Status Register read command may be issued to cause the Status Register ASO to appear at every word address in the device address space. In this Status Register ASO Mode, the device interface waits for a read access and, any write access is ignored. The next read access to the device accesses the content of the Status Register, exits the Status Register ASO, and returns to the previous (calling) mode in which the Status Register read command was received.

Similarly, commands that read and write other registers use Peripheral Register Mode, in which the register appears in a temporary ASO that is automatically exited after the read or write of the command selected register. The read or write occurs in the last cycle of the register access command sequence.

In EA Mode the EAC is performing an Embedded Algorithm, such as programming or erasing a non-volatile memory array. While in EA Mode, none of the Flash Memory Array is readable. While in EA Mode, only the Program / Erase Suspend command or the Status Register Read command will be accepted. All other commands are ignored. Thus, no other ASO may be entered from the EA Mode.

In ASO Mode, one of the remaining overlay address spaces is entered (overlaid on the Flash Memory Array address map). Only one ASO may be entered at any one time. Commands to the device affect the currently entered ASO. Only certain commands are valid for each ASO. These are listed in each ASO related section of [Table 39, Command Definitions on page 65](#).

The following ASOs have non-volatile data that may be programmed to change 1s to 0s:

- Secure Silicon Region
- ASP Configuration Register (ASPR)
- Persistent Protection Bits (PPB)
- Password
- Only the PPB ASO has non-volatile data that may be erased to change 0s to 1s.

When a program or erase command is issued while one of the non-volatile ASOs is entered, the EA operates on the ASO. The ASO is not readable while the EA is active. When the EA is completed the ASO remains entered and is again readable. Suspend and Resume commands are ignored during an EA operating on any of these ASOs.

The Peripheral Register Mode is used to manage the Power-On Reset Timer, Interrupt Configuration Register, Interrupt Status Register, Volatile Configuration Register, and the Non-Volatile Configuration Register.

5.1 Flash Memory Array

The S26KL/S26KS family has a uniform sector architecture with a sector size of 256 kB. The following tables show the sector architecture of the devices.

A user configuration option is available to overlay either the first sector (SA00) or last sector (SAmax) with eight 4-kB Parameter-Sectors. The Parameter-Sector address map showing how the lowest or highest sector is partitioned is shown in the following memory address map tables. The Parameter-Sectors can be erased and programmed in the normal manner using the standard erase and program command sequences targeting the appropriate Parameter-Sector addresses. Note that the smaller Parameter-Sectors need to include A[16:11] as part of the address identifying the target Parameter-Sector during erase and program command sequences. Configuring the first or last uniform sector to include the parameter sectors is accomplished with the Non-Volatile Configuration Register.

Note: The following tables have been condensed to show sector related information for an entire device on a single page. Sectors and their address ranges that are not explicitly listed (such as SA001 – SA510) have sector starting and ending addresses that form the same pattern as all other sectors of that size. For example, all 256-kB sectors have the pattern XX00000h – XX1FFFFh.

Table 7. S26KL512S and S26KS512S Sector and Memory Address Map

Sector Size (kbyte)	Sector Count	Sector Range	Address Range (16-bit)	Notes
256	256	SA00	0000000h – 001FFFFh	Sector Starting Address
		:	:	–
		SA255	1FE0000h – 1FFFFFFh	Sector Ending Address

Table 8. S26KL256S and S26KS256S Sector and Memory Address Map

Sector Size (kbyte)	Sector Count	Sector Range	Address Range (16-bit)	Notes
256	128	SA00	0000000h – 001FFFFh	Sector Starting Address
		:	:	–
		SA127	0FE0000h – 0FFFFFFh	Sector Ending Address

Table 9. S26KL128S and S26KS128S Sector and Memory Address Map

Sector Size (kbyte)	Sector Count	Sector Range	Address Range (16-bit)	Notes
256	64	SA00	0000000h – 001FFFFh	Sector Starting Address
		:	:	–
		SA63	07E0000h – 07FFFFFFh	Sector Ending Address

Table 10. Main Array Sector 0 Overlaid with Eight 4-kB Parameter-Sectors

Main Array Sector Size	Parameter-Sector Number	Address Size	Address Range (16-bit)	Notes
256 kB	0	4 kB	0000000h – 00007FFh	Start of Parameter-Sector 0
	1	4 kB	0000800h – 0000FFFh	Parameter-Sector 1
	2	4 kB	0001000h – 00017FFh	Parameter-Sector 2
	3	4 kB	0001800h – 0001FFFh	Parameter-Sector 3
	4	4 kB	0002000h – 00027FFh	Parameter-Sector 4
	5	4 kB	0002800h – 0002FFFh	Parameter-Sector 5
	6	4 kB	0003000h – 00037FFh	Parameter-Sector 6
	7	4 kB	0003800h – 0003FFFh	End of Parameter-Sector 7
	Exposed Portion of Main Array Sector 0	224 kB	0004000h – 001FFFFh	Mapped to exposed portion of Main Array Sector 0

Table 11. Last Sector Overlaid with Eight 4-kB Parameter-Sectors

Main Array Sector Size	Parameter-Sector Number	Address Size	Address Range (16-bit)	Notes
256 kB	Exposed portion of last sector in main array	224 kB	xx00000h – xx1BFFFh	Mapped to exposed portion of Main Array Sector (last)
	0	4 kB	xx1C000h – xx1C7FFh	Start of Parameter-Sector 0
	1	4 kB	xx1C800h – xx1CFFFh	Parameter-Sector 1
	2	4 kB	xx1D000h – xx1D7FFh	Parameter-Sector 2
	3	4 kB	xx1D800h – xx1DFFFh	Parameter-Sector 3
	4	4 kB	xx1E000h – xx1E7FFh	Parameter-Sector 4
	5	4 kB	xx1E800h – xx1EFFFh	Parameter-Sector 5
	6	4 kB	xx1F000h – xx1F7FFh	Parameter-Sector 6
	7	4 kB	xx1F800h – xx1FFFFh	End of Parameter-Sector 7

5.2 Device ID and CFI (ID-CFI) ASO

There are two traditional methods for systems to identify the type of flash memory installed in the system. One is Device Identification (ID). The other method is called Common Flash Interface (CFI).

For ID, a command is used to enable an address space overlay where up to 16 word locations can be read to get JEDEC manufacturer identification (ID), device ID, and some configuration and protection status information from the flash memory. The system can use the manufacturer and device IDs to select the appropriate driver software to use with the flash device.

CFI also uses a command to enable an Address Space Overlay where an extendable table of standard information about how the flash memory is organized and operates can be read. With this method the driver software does not have to be written with the specifics of each possible memory device in mind. Instead the driver software is written in a more general way to handle many different devices but adjusts the driver behavior based on the information in the CFI table.

Traditionally these two address spaces have used separate commands and were separate overlays. However, the mapping of these two address spaces are non-overlapping and so can be combined in to a single address space and appear together in a single overlay. Either of the traditional commands used to access (enter) the Autoselect (ID) or CFI overlay will cause the now combined ID-CFI address map to appear.

The ID-CFI address map appears within, and overlays the Flash Memory Array data of the sector selected by the address used in the ID-CFI enter command. While the ID-CFI ASO is entered the content of all other sectors is undefined.

The ID-CFI address map starts at location zero of the selected sector. Locations above the maximum defined address of the ID-CFI ASO to the maximum address of the selected sector have undefined data. The ID-CFI enter commands use the same address and data values used on previous generation memories to access the JEDEC Manufacturer ID (Autoselect) and Common Flash Interface (CFI) information, respectively.

Table 12. ID-CFI Address Map Overview

Word Address	Description	Read / Write
(SA) + 0000h to 000Fh	Device ID (traditional Autoselect values)	Read Only
(SA) + 0010h to 0079h	CFI data structure	Read Only
(SA) + 007Ah to 00FFh	Undefined	Read Only

For the complete address map see [Table 33, ID \(Autoselect\) Address Map on page 59](#).

5.2.1 Device ID

The Joint Electron Device Engineering Council (JEDEC) standard JEP106T defines the manufacturer ID for a compliant memory. Common industry usage defined a method and format for reading the manufacturer ID and a device specific ID from a memory device. The manufacturer and device ID information is primarily intended for programming equipment to automatically match a device with the corresponding programming algorithm. Cypress has added additional fields within this 32-byte address space.

The original industry format was structured to work with any memory data bus width e.g. x8, x16, x32. The ID code values are traditionally byte wide but are located at bus width address boundaries such that incrementing the device address inputs will read successive byte, word, or double word locations with the ID codes always located in the least significant byte location of the data bus. Because the device data bus is word wide, each code byte is located in the lower half of each word location. The original industry format made the high order byte always zero. Cypress has modified the format to use both bytes in some words of the address space. For the detail description of the Device ID address map see [Table 33, ID \(Autoselect\) Address Map on page 59](#).

5.2.2 Common Flash Memory Interface

The JEDEC Common Flash Interface (CFI) specification (JESD68.01) defines a standardized data structure that may be read from a flash memory device, which allows vendor-specified software algorithms to be used for entire families of devices. The data structure contains information for system configuration such as various electrical and timing parameters, and special functions supported by the device. Software support can then be device-independent, Device ID-independent, and forward-and-backward-compatible for entire flash device families.

The system can read CFI information at the addresses within the selected sector as shown in [7., Device ID and Common Flash Interface \(ID-CFI\) ASO Map on page 59](#).

Like the Device ID information, CFI information is structured to work with any memory data bus width e.g. x8, x16, x32. The code values are always byte wide but are located at data bus width address boundaries such that incrementing the device address reads successive byte, word, or double word locations with the codes always located in the least significant byte location of the data bus. Because the data bus is word wide, each code byte is located in the lower half of each word location and the high order byte is always zero.

For further information, refer to the *CFI Specification, Version 1.5* (or later), and the *JEDEC publications JEP137-A and JESD68.01*.

6. Embedded Operations

6.1 Embedded Algorithm Controller (EAC)

The EAC takes commands from the host system for programming and erasing the flash memory arrays and performs all the complex operations needed to change the non-volatile memory state. This frees the host system from any need to manage the program and erase processes.

There are five EAC operation categories:

- Deep Power-Down
- Standby (Read Mode)
- Address Space Switching
- Embedded Algorithms (EA)
- Advanced Sector Protection (ASP) Management

6.1.1 Deep Power-Down

In the Deep Power-Down (DPD) Mode current consumption is driven to the lowest level. The DPD Mode must be entered while the device is in the Standby state while not in an ASO.

6.1.2 EAC Standby

In the Standby State current consumption is greatly reduced. The EAC enters its Standby State when no command is being processed and no Embedded Algorithm is in progress. If the device is deselected ($CS\# = \text{High}$) during an Embedded Algorithm, the device still draws active current until the operation is completed (I_{CC3}). I_{CC4} in [10.4, DC Characteristics \(CMOS Compatible\)](#) on [page 73](#) represents the standby current specification when both the Host Interface and EAC are in their Standby state.

6.1.3 Address Space Switching

Writing specific address and data sequences (command sequences) switch the memory device address space from the Flash Memory Array to one of the Address Space Overlays (ASO).

Embedded Algorithms operate on the information visible in the currently active (entered) ASO. The system continues to have access to the ASO until the system issues an ASO Exit command, performs a Hardware Reset, or until power is removed from the device. An ASO Exit Command switches from an ASO back to the Flash Memory Array address space. The commands accepted when a particular ASO is entered are listed between the ASO Enter and Exit commands in the command definitions table. See [8.1, Command Summary on page 65](#) for address and data requirements for all command sequences.

6.1.4 Embedded Algorithms (EA)

Changing the non-volatile data in the memory array requires a complex sequence of operations that are called Embedded Algorithms (EA). The algorithms are managed entirely by the device's internal Embedded Algorithm Controller (EAC). The main algorithms perform programming and erase of the Main Array Data and the ASOs. The host system writes command codes to the flash device address space. The EAC receives the commands, performs all the necessary steps to complete the command, and provides status information during the progress of an EA.

6.2 Program and Erase Summary

Flash data bits are erased in parallel in a large group called a sector. The erase operation places each data bit in the sector in the logical 1 state (High). Flash data bits may be individually programmed from the erased 1 state to the programmed logical 0 (low) state. A data bit of 0 cannot be programmed back to a 1. A succeeding read shows that the data is still 0. Only erase operations can convert a 0 to a 1. Programming the same word location more than once with different 0 bits will result in the logical AND of the previous data and the new data being programmed.

The duration of program and erase operations is shown in [Section 12., Embedded Algorithm Performance on page 88](#).

Program and erase operations may be suspended.

- An erase operation may be suspended to allow either programming or reading of another sector (not in the erase sector) in the erase operation. No other erase operation can be started during an erase suspend.
- A program operation may be suspended to allow reading of another location (not in the Line being programmed).
- No other program or erase operation may be started during a suspended program operation; program or erase commands will be ignored during a suspended program operation.
- After an intervening program operation or read access is complete the suspended erase or program operation may be resumed.
- Program and Erase operations may be interrupted as often as necessary but in order for a program or erase operation to progress to completion there must be some periods of time between resume and the next suspend commands greater than or equal to t_{PRS} or t_{ERS} in [Embedded Algorithm Performance on page 88](#).
- When an Embedded Algorithm (EA) is complete, the EAC returns to the operation state and address space from which the EA was started (Erase Suspend or EAC Standby).

The system can determine the status of a program or erase operation by reading the Status Register ([Error Types and Clearing Procedures on page 45](#)).

Any commands written to the device during the Embedded Program Algorithm are ignored except the Program Suspend, and Status Read command.

Any commands written to the device during the Embedded Erase Algorithm are ignored except Erase Suspend and Status Read command.

A Hardware Reset immediately terminates any in progress program / erase operation and returns to Read Mode after t_{RPH} time. The terminated operation should be reinitiated once the device has returned to the Standby state, to ensure data integrity.

For performance and reliability reasons programming is internally done on 16-byte half-pages, using an aligned 16-byte address range.

I_{CC3} in [DC Characteristics \(CMOS Compatible\) on page 73](#) represents the active current specification for a write (Embedded Algorithm) operation.

6.2.1 Program Granularity

The S26KL/S26KS supports two methods of programming, Word or Write Buffer Programming.

Word programming examines the data word supplied by the command and programs 0's in the addressed memory array word to match the 0's in the command data word.

Write Buffer Programming examines the write buffer and programs 0's in the addressed memory array Line to match the 0's in the write buffer. The write buffer does not need to be completely filled with data. It is allowed to program as little as a single bit, several bits, a single word, a few words, a half-page, multiple half-pages, or the entire buffer as one programming operation. Use of the write buffer method reduces host system overhead in writing program commands and reduces memory device internal overhead in programming operations to make Write Buffer Programming more efficient and thus faster than programming individual words with the Word Programming command.

Each half-page can be programmed by either method. Half-pages programmed by different methods may be mixed within a Line.

Word Programming and Write Buffer Programming, more than once within a half-page, is supported for legacy software compatibility. However, using Word Programming or Write Buffer Programming more than once within a half-page without an erase will disable the device's ECC functionality for that half-page. For applications requiring multiple programming operations within the same half-page, it is recommended to add system software Error Detection and Correction, to enhance the data integrity of half-pages.