Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

AN99195

Programmer's Guide for the Cypress HyperFlash Family

Author: Zhi Feng

Associated Part Family: HyperFlash

Cypress HyperFlash memories use the low-signal-count, high-performance HyperBus™ interface that is able to transfer up to 333 Mbytes/s. This document describes, for software programmers and system engineers, how to use the HyperFlash family.

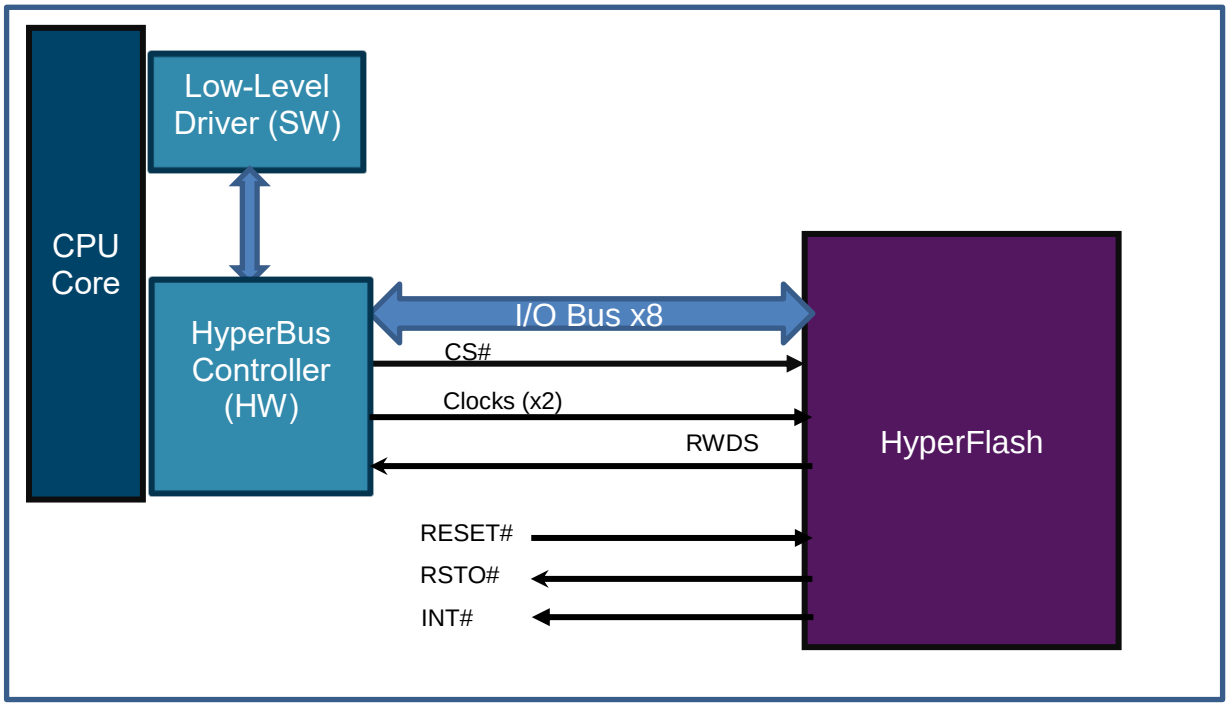
1 Basic HyperFlash Characteristics

The HyperFlash family consists of both 3.0-V KL-S and 1.8-V KS-S devices. Both 3-V and 1.8-V devices use the same software command set. HyperFlash combines the advantages of a low-signal-count interface memory such as serial peripheral interface (SPI) NOR with even higher read performance than Parallel NOR (PNOR) devices. This section compares HyperFlash to traditional PNOR and SPI NOR devices.

1.1 HyperFlash System Diagram

The following diagram shows how a typical embedded system chipset connects to HyperFlash. It requires the HyperBus memory controller that communicates with the HyperFlash through a set of control and I/O pins. The software low-level driver provides basic read, program, and erase functions to the upper-level applications.

Figure 1. HyperFlash System Diagram



1.2 Memory Architecture

HyperFlash has uniform sector architecture with a sector size of 256 KB. A sector is the smallest erasable block in a flash device. To give users the flexibility of having smaller sectors, there is a user configuration option to partially overlay either the first or the last sector of the device. Those smaller sectors are called parameter sectors in which users typically store system parameters.

HyperFlash has an internal RAM area called Write Buffer, which is aligned to a 512-byte boundary. When using the Write Buffer to program the memory, HyperFlash allows up to 512 bytes to be programmed in one operation.

Typically, operations internal to the device that take a period of time to complete, such as erasing a sector and programming, are called embedded operations (EO). During an EO, the device is busy and most commands are forbidden. Users can use the Read Status Register command to determine the completion of the EO.

Read commands are associated with pages. A page is a 16-word (32-byte) area of the array, on a 16-word boundary. Additional latency may be inserted when reading across the page boundaries. Refer to Section 6 for more details. A Half-Page is defined as a 16-byte of memory, half the size of a page. Half-Pages will be mentioned in Section 7 when the programming topic is discussed.

1.3 Comparison to Traditional Parallel NOR Devices

Software operations in HyperFlash are similar to traditional PNOR devices such as the Cypress GL-P and GL-S families. Although the physical pin connection of HyperFlash is different from PNOR devices, it adopts the PNOR command set as its baseline command set.

For example, to erase a sector of the device, users would issue a sequence of write commands identical to the erase command sequence used by the Cypress GL-P and GL-S families. The electrical signaling differences between HyperFlash and PNOR are transparent to the software. The HyperFlash memory controller in the host system handles the translation of software write and read accesses into the HyperBus signaling protocol. Therefore, users can make use of the same Low-Level Driver (LLD) software for earlier generations of PNOR with HyperFlash devices. HyperFlash devices have optional commands in addition to the baseline command set that enable additional features of the devices.

1.4 Comparison to SPI NOR Devices

The similarity between HyperFlash and SPI NOR devices is only in their signal pin physical placement on the package. The signaling on the pins for HyperFlash conforms to the HyperBus specification whereas SPI devices communicate using an SPI protocol and the command sets are different.

2 Configuring HyperFlash Devices

HyperFlash devices provide a Non-Volatile Configuration Register (NVCR), and its counterpart, the Volatile Configuration Register (VCR). The VCR is for users to temporarily change configuration settings for testing. The VCR value will be reset to the one held in the NVCR at the next power cycle. If a nonvolatile configuration is desired, the NVCR should be updated to the desired value. The NVCR and VCR are collectively referred to as xVCR registers. Table 1 shows the bit assignments of the xVCRs.

Table 1. Volatile and Non-Volatile Configuration Registers

xVCR Bit	Function	Settings (Binary)
xVCR.15	Reserved	1 - Reserved (default)
xVCR14 – xVCR12	Drive Strength	000 - (default) (Refer to datasheet for actual device-dependent impedance)
xVCR.11	xVCR Freeze	0 - VCR or NVCR Locked (No Programming or Erasing of NVCR, no changes to VCR) 1 - VCR and NVCR Unlocked (factory default)
xVCR.10	SSR Freeze	0 - Secure Silicon Region Locked (Programming not allowed) 1 - Secure Silicon Region Unlocked (factory default)
xVCR.9 – xVCR.8	Parameter-Sector Mapping	00 - Parameter-Sectors and Read Password Sectors mapped into lowest addresses 01 - Parameter-Sectors and Read Password Sectors mapped into highest addresses 10 - Uniform Sectors with Read Password Sector mapped into lowest addresses. (factory default) 11 - Uniform Sectors with Read Password Sector mapped into highest addresses

xVCR Bit	Function	Settings (Binary)
xVCR.7 – xVCR.4	Read Latency	1011 - 16 Clock Latency (factory default)
xVCR.3	Reserved	1 - Reserved (default)
xVCR.2	Reserved	0 - Reserved (default)
xVCR.1 – xVCR.0	Burst Length	00 – Reserved 01 - 64 bytes 10 - 16 bytes 11 - 32 bytes (factory default)

2.1 Before First Memory Array Program or Erase

The HyperFlash family has the default uniform sector architecture with a 256-KB sector size. A user configuration option is available to partially overlay either the first sector or the last sector with eight 4-KB parameter sectors. This configuration is controlled by the Non-Volatile Configuration Register Bits 9-8. See Table 1 for the actual setting values.

If users want to configure parameter sectors, these two bits should be programmed to the desired values before any programming or erasing of the flash array; otherwise, the data in the overlay portion of the flash array may be lost.

2.2 After All Configurations Are Set

After the power-up default configuration has been determined and fully debugged, users may use the NVCR Bit 11 to permanently lock the xVCR registers. After the bit is programmed to '0', no changes can be made to xVCRs. The device configuration is permanently locked.

3 Status Register

HyperFlash has a single 16-bit Status Register (SR) that can be used to check the current state of the device, embedded operation status, or previous program or erase status. Only the lower eight bits of the SR are defined.

Table 2. Status Register

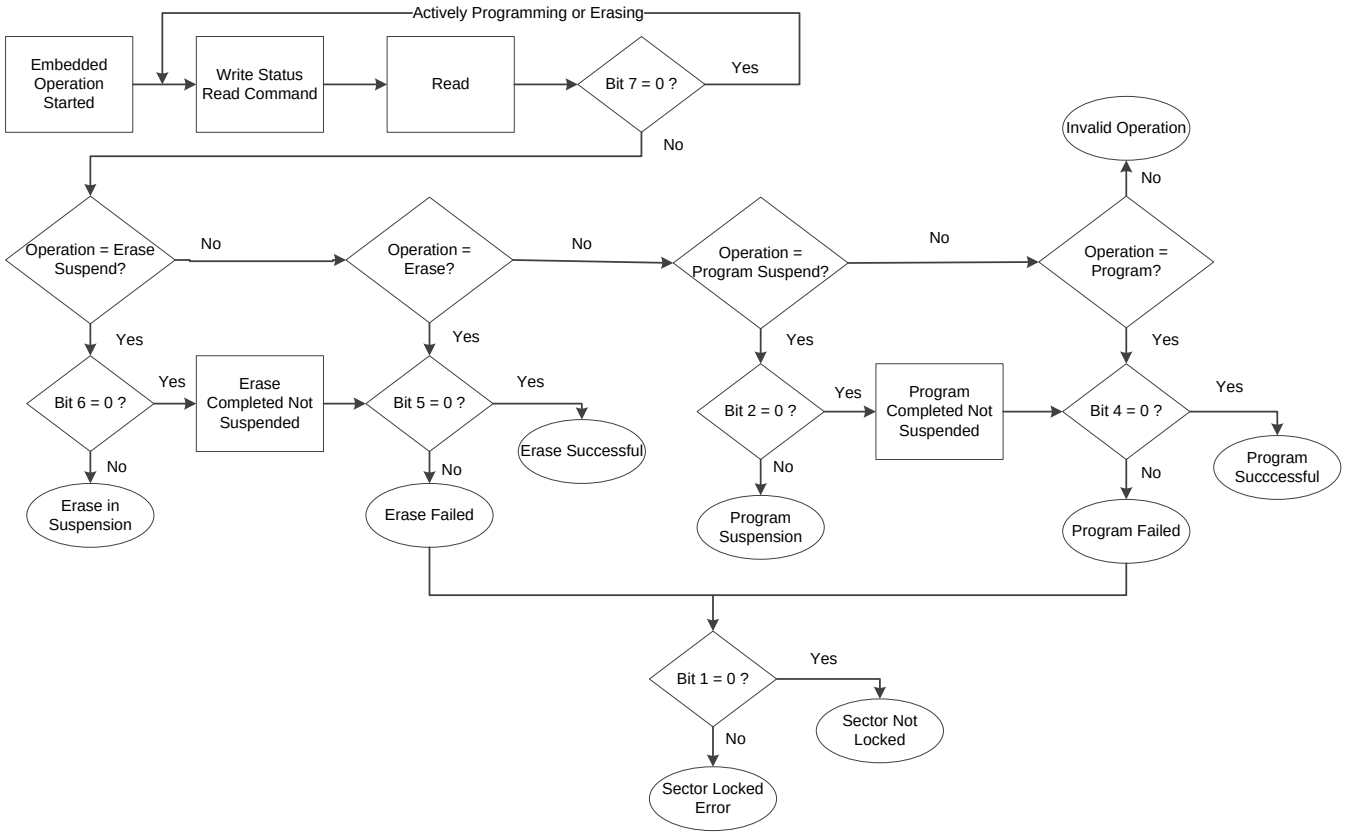
Bit #	15:9	8	7	6	5	4	3	2	1	0
Bit Description	Reserved	Reserved	Device Ready Bit	Erase Suspend Status Bit	Erase Status Bit	Program Status Bit	Write Buffer Abort Status Bit	Program Suspend Status Bit	Sector Lock Status Bit	Sector Erase Status Bit
Bit Name			DRB	ESSB	ESB	PSB	WBASB	PSSB	SLSB	ESTAT
Reset Status	X	0	1	0	0	0	0	0	0	0
Busy Status	Invalid	Invalid	0	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid
Ready Status	X	X	1	0 = No Erase in Suspension 1 = Erase in Suspension	0 = Erase Successful 1 = Erase Fail	0 = Program Successful 1 = Program Fail	0 = Program Not Aborted 1 = Program Aborted during Write to Buffer Command	0 = No Program in Suspension 1 = Program in Suspension	0 = Sector Not Locked during Operation 1 = Sector Locked Error	0 = Sector Erase Status Command Result = previous erase did not complete successfully 1 = Sector Erase Status Command Result = previous erase completed successfully

After issuing a command that triggers an embedded operation, such as programming or erasing, users should always check the SR to make sure that the embedded operation has completed before proceeding to the next command. During an embedded operation, only the program/erase suspend[1] command or the Status Register Read command will be accepted. All other commands are ignored.

If an error happens during an embedded operation, users need to clear the error bit by using the Clear Status Register command before proceeding to the next command.

The following diagram shows a polling function that uses the Status Register to determine the device status after starting an embedded operation such as programming or erasing a sector. The algorithm here has the assumption that the type of the most recent operation, shown in the four large diamonds in the middle of the diagram, is passed into this software polling function. This implementation simplifies the software flow. It is feasible to design an algorithm that does not need to know the type of operation it is polling for; however, it will be more complicated.

Figure 2. Polling Function Using Status Register to Determine Device Status After Starting an Embedded Operation



4 Reset

There are three kinds of resets in a HyperFlash device: Power on Reset (POR), also called cold reset; Hardware Reset, also called warm reset triggered by the RESET# signal; and the Software Reset triggered by the Software Reset Command (F0h).

Upon cold and warm resets, all VCR bits are loaded from the corresponding NVCR bit values.

The RSTO# pin can be used on the system to indicate the completion of a POR. The pin will transition from LOW to HIGH state after the device completes the power up stage, plus a user-defined timeout period.

[1] During a program or erase operation, users may issue a suspend command to suspend the EO in order to quickly go back to the read mode. The operation will be suspended until the resume command is entered.

Software Reset is typically used to clear the Status Register or get the device back to Read Array Mode from an Address Space Overlay (ASO) state or an unknown state. Software reset will not affect ongoing embedded operations or any Configuration Register values.

5 CFI / Device ID

Similar to earlier PNOR devices, the HyperFlash family follows the JEDEC Common Flash Interface (CFI) specification to provide a standardized data structure that may be read by a command. The data structure contains information 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.

Both ID (Autoselect) and CFI commands in HyperFlash allow users to access the combined ID/CFI data set. All data contained within the ID/CFI data set is available after using either the ID (Autoselect) or CFI Entry sequence.

See the HyperFlash family datasheet (referenced at the end of the App Note) for a complete explanation of the ID/CFI data structure.

6 Maximizing Read Performance

HyperFlash devices are burst-mode flash memories. To get the maximum read performance, users will need to optimize two parameters. First is the initial latency, the time between when the read operation is initiated and when the HyperFlash device starts outputting the data on the I/O lines. The second is the type and length of burst.

The HyperFlash memory array initial access time is defined as t_{ACC} in the datasheet. This time is required to move the data from the flash array to the data I/O; t_{ACC} is independent of the clock frequency the user has chosen to run at. The higher the clock frequency, the more clock cycles are required for the data to reach the device outputs. This time duration in terms of clock cycles is called latency clocks.

In HyperFlash, the number of latency clocks is controlled by four Latency Code Configuration Register bits: xVCR Bits 7-4. The following table shows the relationship between the latency code, latency clocks, and the maximum clock frequency that will satisfy the t_{ACC} requirement for initial read latency.

Table 3. Latency Settings

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.*

HyperFlash devices being burst read devices are most efficient when more than one word of data is read at a time. Reading single words will incur the latency cycles for each word, whereas reading multiple words in a single read command will only incur the latency cycles once. So users need to verify that all functions used to access the HyperFlash are using the burst mode; do not assume that library functions are written efficiently.

HyperFlash devices support two types of bursts: wrapped and linear. Wrapped bursts can be 16, 32, or 64 bytes long depending on the value of the xVCR bits 0 and 1. Linear bursts allow sequential data to be read for an arbitrary length.

If the read is configured to be linear and the read address does not start on a 16-byte alignment boundary (0h or 8h word address multiple offset), in addition to the initial latency, the user may need to insert a page-crossing latency when crossing the first 32-byte alignment boundary (to the second page). The initial and first page-crossing latency cycles depend on the clock frequency and the starting address offset. See tables in the datasheet for the actual counts.

Note that all subsequent page crossings within the same linear access will not require any additional latency clocks. Wrapped read transactions of 16-byte and 32-byte bursts do not cross page boundaries and do not incur inter-page-boundary-crossing latencies. For a 64-byte wrapped burst read, additional latency may need to be inserted during the page boundary crossing, depending on the starting address.

7 Programming

HyperFlash family devices have a 512-byte write buffer, which is aligned to 512-byte boundaries. Programming data to the flash device is most efficient when writing in a 512-byte length and aligned increments. Although smaller writes are allowed, for best performance, software should, whenever possible, program data in full, address-aligned, write buffer increments.

In HyperFlash, the flash memory array is structured in 16-byte length and aligned Half-Pages. While multiple program operations within a Half-Page are not recommended, they are allowed for backward compatibility with traditional Parallel NOR products. Programming more than once within the same Half-Page, per erase of the sector in which the Half-Page resides, reduces the data integrity of the Half-Page. If it is required to program in less than Half-Page multiples such that Half-Pages are programmed more than once per erase, it is recommended to add software error correction information for the data in such Half-Pages.

For example, a simple flash file system may write 512-byte file records, each with 30-byte metadata. If the user programs the next file sector immediately after the 30-byte metadata, it would cause misalignments to all subsequent sectors, as shown in Table 4. In this case, it would be highly recommended that the user insert 2-byte padding data at the end of the 30-byte metadata structure so that all files sectors and metadata are aligned with the internal 16-byte Half-Pages, and 512-byte Write Buffer pages.

Table 4. Misaligned Data Storage

File sector	1st record 512 Bytes			1st metadata 30 Bytes		2nd record 512 Bytes			2nd metadata #0 Bytes	
Flash Page	HP 0	… …	HP 31	HP 32	HP 33	HP 34	… …	HP 64	HP 65	HP 66

Note: HP: Half Page (16-byte)

Table 5. Aligned Data Storage

File sector	1st sector 512 Bytes			1st metadata 30 + 2 pad Bytes		2nd sector 512 Bytes			2nd metadata 30 + 2 pad Bytes	
Flash Page	HP 0	… …	HP 31	HP 32	HP 33	HP 34	… …	HP 65	HP 66	HP 67

8 Erasing

HyperFlash family devices have uniform sector size of 256 KB. The first or the last sector can be partially overlaid by eight parameter sectors each with a 4-KB size. Users need to pay attention to the fact that in such overlay configuration, there is a sector above or below the parameter sectors (224 KB in size) between the parameter sectors and the remaining uniform 256-KB sectors. When issuing an erase command, the address used must be within the intended sector.

9 Secure Silicon Region (SSR)

HyperFlash family devices provide 1024 bytes of Secure Silicon Region (SSR) that is a One-Time Programmable (OTP) area separated from the main flash array. The area is divided into 32 individually lockable, 32-byte aligned regions (32 × 32 bytes = 1024 bytes). Users can access the SSR by entering the SSR Address Space Overlay (ASO).

When reading from the SSR, enter the SSR ASO and perform a read to an SSR address offset. If the address entered is outside of the 1024-byte SSR address range, main array data will be retrieved.

When programming the SSR, enter the SSR ASO and perform programming in the same way as in the normal array.

The SSR is protected by the FREEZE bit in xVCR Bit 10. If FREEZE is set, the program command will be ignored. No error is reported.

The Region 0 of the SSR (first 32 bytes) is a special region. The first 16 bytes of Region 0 are reserved for Cypress to program in a Random Number that can be used as a unique device identification such as a serial number. The next four bytes are the Lock Bits. Each lock bit controls the corresponding 32 SSR regions, from Region 0 to Region 31. Users can program these bits to individually lock any SSR regions. Once the region is locked, it is permanently locked.

Any attempt to program the Random Number area will result in a program error. Similarly, if an SSR region is locked by its lock bit, any attempt to program into the region will result in a program error.

When programming the SSR, the program page size is the same as the normal flash array page program; that is, 512 bytes. That means the user can program multiple SSR regions in the same program command.

If the program data entered is more than the page size, the data will be wrapped to the beginning of the page, just like a normal page programming command. In this case, the wrapped data may coincide with Region 0, which is a special region as mentioned above. In this case, the program command may fail if it contains some data intended for the first 16 bytes of Region 0. Loading data beyond the end of the page programming buffer is not recommended.

10 INT# Output Pin

HyperFlash family devices offer an INT# output pin that users can configure to generate an interrupt to the system when transitioning from BUSY to READY state, for example, completion of an embedded operation.

Users can read the Interrupt Status Register (ISR) to verify whether a Busy-to-Ready event has occurred, or that the last POR was completed successfully, and then reset the ISR by writing the corresponding bits to '1'.

11 Conclusion

Cypress HyperFlash family offers the same physical footprint as SPI devices, and the same software command interface as the Parallel NOR devices. It also combines the advantages of the low pin count of the SPI devices and the high throughput of NOR devices. In fact, its performance is even higher than Parallel NOR devices.

The HyperFlash family provides an easy transition for users who have used SPI NOR or Parallel NOR devices in their systems, reduces the system cost, and helps achieve better system performance.

Contact Cypress Customer Support for additional help when using HyperFlash family devices.

12 References

- S26KL/KSxxxS datasheet
- S26KL/KSxxxS Low-Level Driver

Document History

Document Title: AN99195 - Programmer's Guide for the Cypress HyperFlash Family

Document Number: 001-99195

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	4852549	ZHFE	08/17/2015	New Application Note
*A	5461225	AHCL	10/04/2016	Updated Figure 2 Updated the Programming section Updated template
*B	5705882	AESATP12	04/21/2017	Updated logo and copyright
*C	6307809	YOQI	09/12/2018	Updated broken hyperlinks

Worldwide Sales and Design Support

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

Products

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

PSoC® Solutions

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

Cypress Developer Community

Community | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

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