



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



P8P Parallel Phase Change Memory (PCM)

Features

- High-performance READ
 - 115ns initial READ access
 - 135ns initial READ access
 - 25ns, 8-word asynchronous-page READ
- Architecture
 - Asymmetrically blocked architecture
 - Four 32KB parameter blocks with top or bottom configuration
 - 128KB main blocks
 - Serial peripheral interface (SPI) to enable lower pin count on-board programming
- Phase change memory (PCM)
 - Chalcogenide phase change storage element
 - Bit-alterable WRITE operation
- Voltage and power
 - V_{CC} (core) voltage: 2.7–3.6V
 - V_{CCQ} (I/O) voltage: 1.7–3.6V
 - Standby current: 80 μ A (TYP)
- Quality and reliability
 - More than 1,000,000 WRITE cycles
 - 90nm PCM technology
- Temperature
 - Commercial: 0°C to +70°C (115ns initial READ access)
 - Industrial: –40°C to +85°C (135ns initial READ access)
- Simplified software management
 - No block erase or cleanup required
 - Bit twiddle in either direction (1:0, 0:1)
 - 35 μ s (TYP) PROGRAM SUSPEND
 - 35 μ s (TYP) ERASE SUSPEND
 - Flash data integrator optimized
 - Scalable command set and extended command set compatible
 - Common Flash interface capable
- Density and packaging
 - 128Mb density
 - 56-lead TSOP package
 - 64-ball easy BGA package
- Security
 - One-time programmable registers
 - 64 unique factory device identifier bits
 - 2112 user-programmable OTP bits
 - Selectable OTP space in main array
 - Three adjacent main blocks available for boot code or other secure information
 - Absolute WRITE protection: $V_{PP} = V_{SS}$
 - Power transition ERASE/PROGRAM lockout
 - Individual zero-latency block locking
 - Individual block lock-down

Table of Contents

Features	1
Functional Description	7
Product Features	7
Memory Maps	8
Package Dimensions	11
TSOP Mechanical Specifications	11
64-Ball Easy BGA Package	13
Pinouts and Ballouts	14
Signal Names and Descriptions	16
Bus Operations	17
READ Operations	17
WRITE Operations	17
OUTPUT DISABLE Operations	17
STANDBY Operations	17
RESET Operations	18
Command Set	18
Device Command Codes	18
Device Command Bus Cycles	20
READ Operations	21
READ ARRAY	21
READ IDENTIFIER	22
READ QUERY	22
PROGRAM Operations	23
WORD PROGRAM	23
BIT-ALTERABLE WORD WRITE	24
BUFFERED PROGRAM	24
BIT-ALTERABLE BUFFER WRITE	25
BIT-ALTERABLE BUFFER PROGRAM	25
PROGRAM SUSPEND	25
PROGRAM RESUME	26
PROGRAM PROTECTION	26
ERASE	26
BLOCK ERASE	27
ERASE SUSPEND	27
ERASE RESUME	28
Security Mode	28
Block Locking	28
Zero Latency Block Locking	28
Lock Block	28
Unlock Block	29
Lock Down Block	29
WP# Lock Down Control	29
Block Lock Status	30
Locking Operations During ERASE SUSPEND	30
Permanent OTP Block Locking	31
WP# Lock Down Control for Selectable OTP Lock Blocks	33
Selectable OTP Locking Implementation Details	33
Registers	33
Read Status Register	33
CLEAR STATUS REGISTER Command	34
System Protection Registers	34

Read Protection Register	35
Program Protection Register	35
Lock Protection Register	35
OTP Protection Register Addressing Details	36
Serial Peripheral Interface (SPI)	37
SPI Overview	37
SPI Signal Names	37
SPI Memory Organization	37
SPI Instruction	39
WRITE ENABLE (WREN)	40
WRITE DISABLE (WRDI)	40
READ IDENTIFICATION (RDID)	41
Read Status Register (RDSR)	41
WIP Bit	42
WEL Bit	42
BP3, BP2, BP1, BP0 Bits	42
Top/Bottom Bit	43
SRWD Bit	43
WRITE STATUS REGISTER (WRSR)	43
Read Data Bytes (READ)	44
Read Data Bytes at Higher Speed (FAST_READ)	45
PAGE PROGRAM (PP)	46
SECTOR ERASE (SE)	47
Power and Reset Specification	48
Power-Up and Power-Down	48
Reset Specifications	48
Power Supply Decoupling	49
Maximum Ratings and Operating Conditions	50
Absolute Maximum Ratings	50
Operating Conditions	50
Endurance	51
Electrical Specifications	51
DC Current Characteristics	51
DC Voltage Characteristics	52
AC Characteristics	52
AC Test Conditions	52
Capacitance	53
AC Read Specifications	53
AC Write Specifications	54
SPI AC Specifications	57
Program and Erase Characteristics	59
Ordering Information	60
Supplemental Reference Information	61
Flowcharts	61
Write State Machine	74
Common Flash Interface	77
Query Structure Output	77
Query Structure Overview	78
CFI Query Identification String	78
Extended Query Tables	82

List of Figures

Figure 1:	56-Lead TSOP	11
Figure 2:	64-Ball Easy BGA Package	13
Figure 3:	56-Lead TSOP Pinout (128Mb)	14
Figure 4:	64-Ball Easy BGA Ballout (128Mb)	15
Figure 5:	Example VPP Power Supply Configuration	26
Figure 6:	Block Locking State Diagram	31
Figure 7:	Selectable OTP Locking Illustration (Bottom Parameter Device Example)	33
Figure 8:	Protection Register Memory Map	36
Figure 9:	WRITE ENABLE (WREN) Instruction Sequence	40
Figure 10:	WRITE DISABLE (WRDI) Instruction Sequence	40
Figure 11:	READ IDENTIFICATION (RDID) Instruction Sequence and Data-Out Sequence	41
Figure 12:	READ STATUS REGISTER (RDSR) Instruction Sequence and Data-Out Sequence	43
Figure 13:	WRITE STATUS REGISTER (WRSR) Instruction Sequence	44
Figure 14:	Read Data Bytes (READ) Instruction Sequence and Data-Out Sequence	45
Figure 15:	FAST_READ Instruction Sequence and Data-Out Sequence	46
Figure 16:	PAGE PROGRAM (PP) Instruction Sequence	47
Figure 17:	SECTOR ERASE (SE) Instruction Sequence	48
Figure 18:	Reset Operation Waveforms	49
Figure 19:	AC Input/Output Reference Waveform	52
Figure 20:	Transient Equivalent Testing Load Circuit	52
Figure 21:	Asynchronous Single-Word Read	54
Figure 22:	Asynchronous Page Mode Read Timing	54
Figure 23:	Write-to-Write Timing	55
Figure 24:	Asynchronous Read to Write Timing	56
Figure 25:	Write to Asynchronous Read Timing	56
Figure 26:	Serial Input Timing	58
Figure 27:	Write Protect Setup and Hold Timing during WRSR when SRWD = 1	58
Figure 28:	Hold Timing	58
Figure 29:	Output Timing	59
Figure 30:	WORD PROGRAM or BIT-ALTERABLE WORD WRITE Flowchart	61
Figure 31:	Full WRITE STATUS CHECK Flowchart	62
Figure 32:	WRITE SUSPEND/RESUME Flowchart	63
Figure 33:	BUFFER PROGRAM or Bit-Alterable BUFFER WRITE Flowchart	65
Figure 34:	BLOCK ERASE Flowchart	67
Figure 35:	BLOCK ERASE FULL ERASE STATUS CHECK Flowchart	68
Figure 36:	ERASE SUSPEND/RESUME Flowchart	69
Figure 37:	LOCKING OPERATIONS Flowchart	71
Figure 38:	PROGRAM PROTECTION REGISTER Flowchart	72
Figure 39:	FULL STATUS CHECK Flowchart	73
Figure 40:	Write State Machine — Next State Table	74

List of Tables

Table 1:	Top Parameter Memory Map	8
Table 2:	Bottom Parameter Memory Map	9
Table 3:	TSOP Package Dimensions	11
Table 4:	Easy BGA Package Dimensions	13
Table 5:	Ball/ Pin Descriptions	16
Table 6:	Bus Operations	17
Table 7:	Command Codes and Descriptions	18
Table 8:	Command Sequences in x16 Bus Mode	20
Table 9:	Read Identifier Table	22
Table 10:	Device Codes	22
Table 11:	Buffered Programming and Bit-Alterable Buffer Write Timing Requirements	25
Table 12:	Bit Alterability vs. Flash Bit-Masking	25
Table 13:	Block Locking Truth Table	29
Table 14:	Block Locking State Transitions	30
Table 15:	Selectable OTP Block Locking Feature	32
Table 16:	Selectable OTP Block Locking Programming of PR-LOCK0	32
Table 17:	Status Register Definitions	34
Table 18:	Protection Register Addressing	36
Table 19:	2K OTP Space Addressing	36
Table 20:	Memory Organization	37
Table 21:	Instruction Set	39
Table 22:	Status Register Format	42
Table 23:	Protected Area Sizes	42
Table 24:	Power and Reset	48
Table 25:	Absolute Maximum Ratings	50
Table 26:	Operating Conditions	50
Table 27:	Endurance	51
Table 28:	DC Current Characteristics	51
Table 29:	DC Voltage Characteristics	52
Table 30:	Test Configuration Component Value for Worst-Case Speed Conditions	53
Table 31:	Capacitance: TA = 25°C, f = 1 MHz ¹	53
Table 32:	AC Read Specifications	53
Table 33:	AC Write Characteristics	54
Table 34:	SPI AC Specifications	57
Table 35:	Program and Erase Specifications	59
Table 36:	Active Line Item Ordering Table (0°C to 70°C)	60
Table 37:	Active Line Item Ordering Table (-40°C to 85°C)	60
Table 38:	WORD PROGRAM or BIT-ALTERABLE WORD WRITE Procedure	61
Table 39:	Full WRITE STATUS CHECK Procedure	62
Table 40:	WRITE SUSPEND/ RESUME Procedure	64
Table 41:	BUFFER PROGRAM OR BIT-ALTERABLE BUFFER WRITE Procedure	66
Table 42:	BLOCK ERASE Procedure	67
Table 43:	BLOCK ERASE FULL ERASE STATUS CHECK Procedure	68
Table 44:	ERASE SUSPEND/ RESUME Procedure	70
Table 45:	LOCKING OPERATIONS Procedure	71
Table 46:	PROGRAM PROTECTION REGISTER Procedure	72
Table 47:	FULL STATUS CHECK Procedure	73
Table 48:	Summary of Query Structure Output as a Function of Device and Model	77
Table 49:	Example of Query Structure Output of x16 Devices	78
Table 50:	Query Structure	78
Table 51:	Block Status Register	79



Table 52:	CFI Identification	79
Table 53:	System Interface Information	79
Table 54:	Device Geometry Definition	80
Table 55:	Bit Field Definitions	81
Table 56:	Hex Code and Values for Device Geometry	81
Table 57:	Primary Vendor-Specific Extended Query	82
Table 58:	Protection Register Information	82
Table 59:	Read Information	83
Table 60:	Partition and Erase Block Region Information	83
Table 61:	Hex Code and Values for Partition and Erase Block Regions	85

Functional Description

P8P parallel phase change memory (PCM) is nonvolatile memory that stores information through a reversible structural phase change in a chalcogenide material. The material exhibits a change in material properties, both electrical and optical, when changed from the amorphous (disordered) to the polycrystalline (regularly ordered) state. In the case of PCM, information is stored via the change in resistance that the chalcogenide material experiences when undergoing a phase change. The material also changes optical properties after experiencing a phase change, a characteristic that has been successfully mastered for use in current rewritable optical storage devices, such as rewritable CDs and DVDs.

The P8P parallel PCM storage element consists of a thin film of chalcogenide contacted by a resistive heating element. In PCM, the phase change is induced in the memory cell by highly localized Joule heating caused by an induced current at the material junction. During a WRITE operation, a small volume of the chalcogenide material is made to change phase. The phase change is a reversible process and is modulated by the magnitude of injected current, the applied voltage, and the duration of the heating pulse.

Unlike other proposed alternative memories, P8P parallel PCM technology uses a conventional CMOS process with the addition of a few additional layers to form the memory storage element. Overall, the basic memory manufacturing process used to make PCM is less complex than that of NAND, NOR or DRAM.

P8P parallel PCM combines the benefits of traditional floating gate Flash, both NOR-type and NAND-type, with some of the key attributes of RAM and EEPROM. Like NOR Flash and RAM technology, PCM offers fast random access times. Like NAND flash, PCM has the ability to write moderately fast, and like RAM and EEPROM, PCM supports bit alterable WRITES (overwrite). Unlike Flash, no separate erase step is required to change information from 0 to 1 and 1 to 0. Unlike RAM, however, the technology is nonvolatile with data retention compared with NOR Flash.

Product Features

P8P parallel PCM devices provide the convenience and ease of NOR flash emulation while providing a set of super set features that exploit the inherent capabilities of PCM technology. The device emulates most of the features of Micron embedded memory (P33). This is intended to ease the evaluation and design of P8P parallel PCM into existing hardware and software development platforms. This basic features set is supplemented by the super set features, which are intended to enable the designer to exploit the inherent capabilities of phase change memory technology and to enable the eventual simplification of hardware and software in the design.

The P8P parallel PCM product family supports 128Mb density and are available in 64-ball easy BGA and 56-lead TSOP packages. These are the same pinouts and packages as the existing P33 NOR Flash devices. Designed for low-voltage systems, P8P parallel PCM supports READ, WRITE, and ERASE operations at a core supply of 2.7V V_{CC} . P8P parallel PCM offers additional power savings through standby mode, which is initiated when the system deselects the device by driving CE inactive.

P8P parallel PCM provides a set of commands that are compatible with industry-standard command sequences used by NOR-type Flash. An internal write state machine (WSM) automatically executes the algorithms and timings necessary for BLOCK ERASE and WRITE. Each emulated BLOCK ERASE operation results in the contents of the addressed block being written to all 1s. Data can be programmed in word or buffer increments. Erase suspend enables system software to pause an ERASE command so it can

read or program data in another block. PROGRAM SUSPEND enables system software to pause programming so it can read from other locations within the device. The status register indicates when the WSM's BLOCKERASE or PROGRAM operation is finished.

A 64-byte, 32 word write buffer is also included to enable optimum write performance. Using the write buffer, data is overwritten or programmed in buffer increments. This feature improves system program performance more than 20 times over independent byte writes.

Similar to floating gate Flash, a command user interface (CUI) serves as the interface between the system processor and internal operation of the device. A valid command sequence written to the CUI initiates device automation. In addition to the CUI, a Flash-compatible common Flash interface (CFI) permits software algorithms to be used for entire families of devices. This enables device-independent, JEDEC ID-independent, and forward- and backward-compatible software support for the specified Flash device families.

The serial peripheral interface (SPI) enables in-system programming through minimal pin count interface. This interface is provided in addition to a traditional parallel system interface. This feature has been added to facilitate the on-board, in-system programming of code into the P8P parallel PCM device after it has been soldered to a circuit board. Preprogramming code prior to high temperature board attach is not recommended with a P8P parallel PCM device. Although device reliability across the operating temperature range is typically superior to that of floating gate Flash, the P8P parallel PCM device may be subject to thermally-activated disturbs at higher temperatures; however, no permanent device damage occurs either during leaded or lead-free board attach.

P8P parallel PCM block locking enables zero-latency block locking/unlocking and permanent locking. Permanent block locking provides enhanced security for boot code. The combination of these two locking features provides complete locking solution for code and data.

PCM technology also supports the ability to change each memory bit independently from 0 to 1 or 1 to 0 without an intervening BLOCKERASE operation. Bit alterability enables software to write to the nonvolatile memory in a similar manner as writing to RAM or EEPROM without the overhead of erasing blocks prior to write. Bit Alterable writes use similar command sequences as word programming and Buffer Programming.

Memory Maps

Table 1: Top Parameter Memory Map

Programming Region Number	Size (KW)	Block	128M b
7	16	130	7FC000-7FFFFF
	16	129	7F8000-7FBFFF
	16	128	7F4000-7F7FFF
	16	127	7F0000-7F3FFF
	64	126	7E0000-7EFFFF
	⋮	⋮	⋮
6	64	112	700000-70FFFF
	64	111	6F0000-6FFFFF
	⋮	⋮	⋮
	64	96	600000-60FFFF

Table 1: Top Parameter Memory Map (Continued)

Programming Region Number	Size (KW)	Block	128M b
5	64	95	5F0000-5FFFFFF
	⋮	⋮	⋮
4	64	80	500000-50FFFF
	⋮	⋮	⋮
3	64	79	4F0000-4FFFFFF
	⋮	⋮	⋮
2	64	64	400000-40FFFF
	⋮	⋮	⋮
1	64	63	3F0000-3FFFFFF
	⋮	⋮	⋮
0	64	48	300000-30FFFF
	⋮	⋮	⋮
5	64	47	2F0000-2FFFFFF
	⋮	⋮	⋮
4	64	32	200000-20FFFF
	⋮	⋮	⋮
3	64	31	1F0000-1FFFFFF
	⋮	⋮	⋮
2	64	16	100000-10FFFF
	⋮	⋮	⋮
1	64	15	0F0000-0FFFFFF
	⋮	⋮	⋮
0	64	0	000000-00FFFF
	⋮	⋮	⋮

Table 2: Bottom Parameter Memory Map

Programming Region Number	Size (KW)	Block	128M b
7	64	130	7F0000-7FFFFFF
	⋮	⋮	⋮
6	64	115	700000-70FFFF
	⋮	⋮	⋮
5	64	114	6F0000-6FFFFFF
	⋮	⋮	⋮
4	64	99	600000-60FFFF
	⋮	⋮	⋮
3	64	98	5F0000-5FFFFFF
	⋮	⋮	⋮
2	64	83	500000-50FFFF
	⋮	⋮	⋮
1	64	82	4F0000-4FFFFFF
	⋮	⋮	⋮
0	64	67	400000-40FFFF
	⋮	⋮	⋮
5	64	66	3F0000-3FFFFFF
	⋮	⋮	⋮
4	64	51	300000-30FFFF
	⋮	⋮	⋮
3	64	50	2F0000-2FFFFFF
	⋮	⋮	⋮
2	64	35	200000-20FFFF
	⋮	⋮	⋮

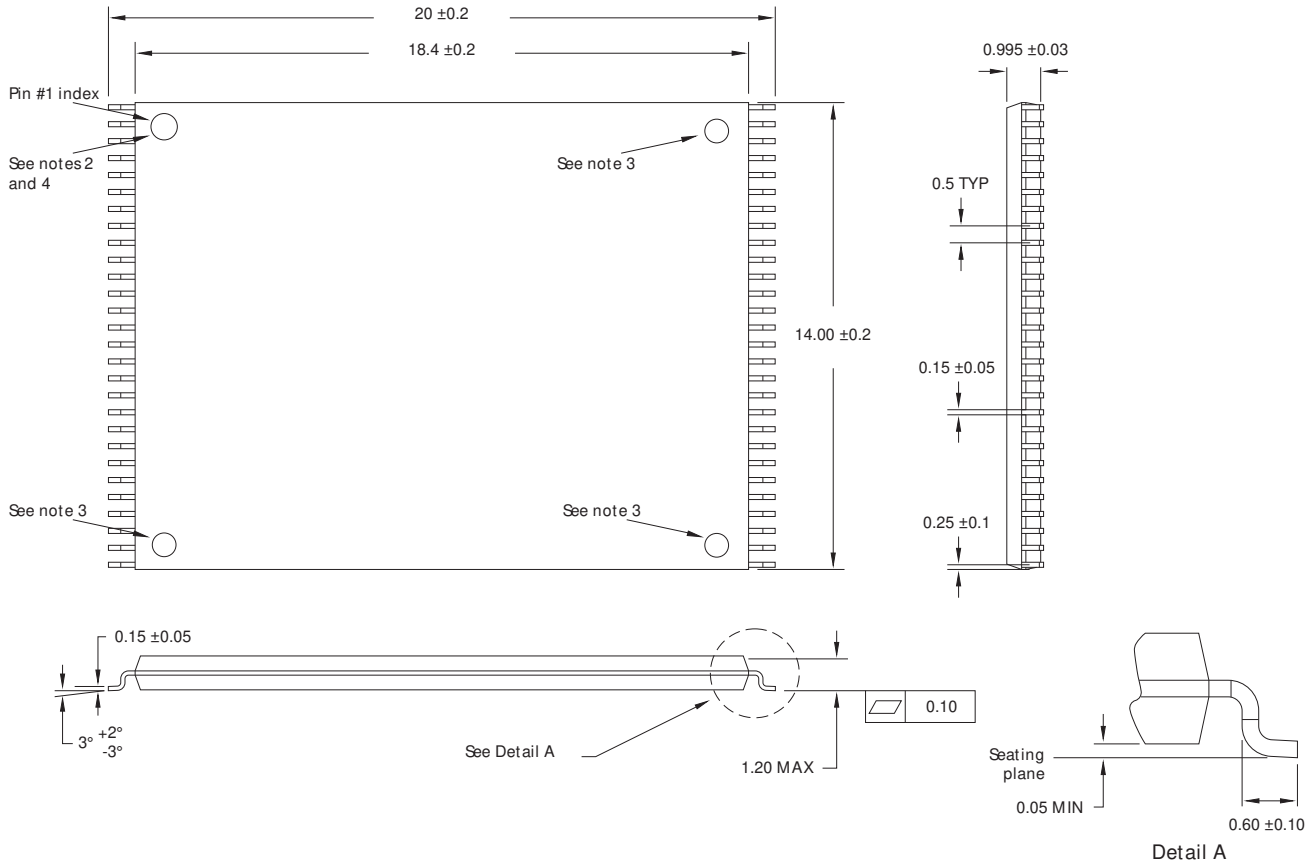
Table 2: Bottom Parameter Memory Map (Continued)

Programming Region Number	Size (KW)	Block	128Mb
1	64	34	1F0000-1FFFFFF
	⋮	⋮	⋮
0	64	19	100000-10FFFF
	64	18	0F0000-0FFFFFF
	⋮	⋮	⋮
	64	4	010000-01FFFF
	16	3	00C000-00FFFF
	16	2	008000-00BFFF
	16	1	004000-007FFF
	16	0	000000-003FFF

Package Dimensions

TSOP Mechanical Specifications

Figure 1: 56-Lead TSOP



- Notes:
1. One dimple on package denotes pin 1.
 2. If two dimples exist, then the larger dimple denotes pin 1.
 3. Pin 1 will always be in the upper left corner of the package, in reference to the product mark.

Table 3: TSOP Package Dimensions

Parameter	Symbol	Millimeters			Inches		
		Min	Nom	Max	Min	Nom	Max
Package height	A	–	–	1.200	–	–	0.047
Standoff	A ₁	0.050	–	–	0.002	–	–
Package body thickness	A ₂	0.965	0.995	1.025	0.038	0.039	0.040
Lead width	b	0.100	0.150	0.200	0.004	0.006	0.008
Lead thickness	c	0.100	0.150	0.200	0.004	0.006	0.008
Package body length	D ₁	18.200	18.400	18.600	0.717	0.724	0.732
Package body width	E	13.800	14.000	14.200	0.543	0.551	0.559
Lead pitch	e	–	0.500	–	–	0.0197	–
Terminal dimension	D	19.800	20.00	20.200	0.780	0.787	0.795

Table 3: TSOP Package Dimensions (Continued)

Parameter	Symbol	Millimeters			Inches		
		Min	Nom	Max	Min	Nom	Max
Lead tip length	L	0.500	0.600	0.700	0.020	0.024	0.028
Lead count	N	–	56	–	–	56	–
Lead tip angle	q	0°	3°	5°	0°	3°	5°
Seating plane coplanarity	Y	–	–	0.100	–	–	0.004
Lead to package offset	Z	0.150	0.250	0.350	0.006	0.010	0.014

64-Ball Easy BGA Package

Figure 2: 64-Ball Easy BGA Package

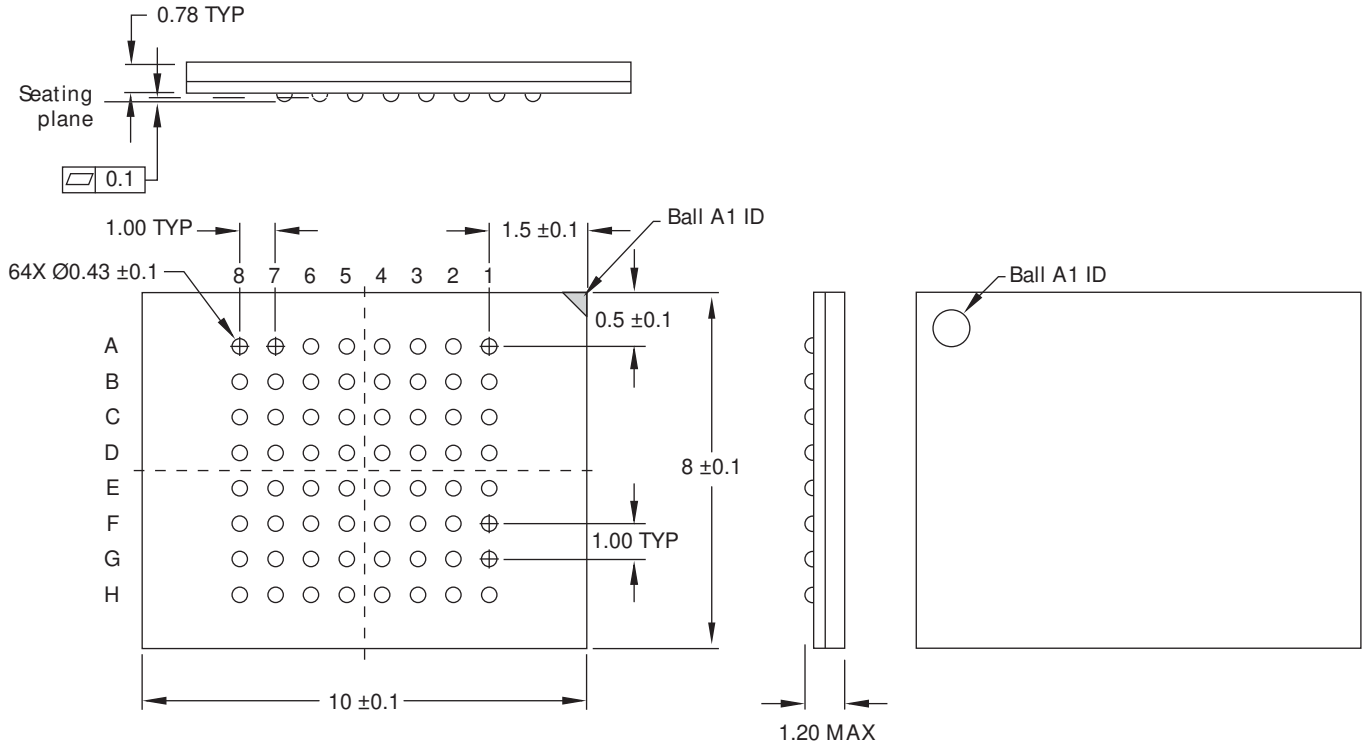


Table 4: Easy BGA Package Dimensions

Parameter	Symbol	Millimeters		
		Min	Nom	Max
Package height (128Mb)	A	–	–	1.20
Ball height	A1	0.25	–	–
Package body thickness (128Mb)	A2	–	0.78	–
Ball (Lead) width	b	0.33	0.43	0.53
Package body width	D	9.90	10.00	10.10
Package body length	E	7.90	8.00	8.10
Pitch	e	–	1.00	–
Ball (lead) count	N	–	64	–
Seating plane coplanarity	Y	–	–	0.10
Corner to ball A1 distance along D	S1	1.40	1.50	1.60
Corner to ball A1 distance along E	S2	0.49	0.50	0.51

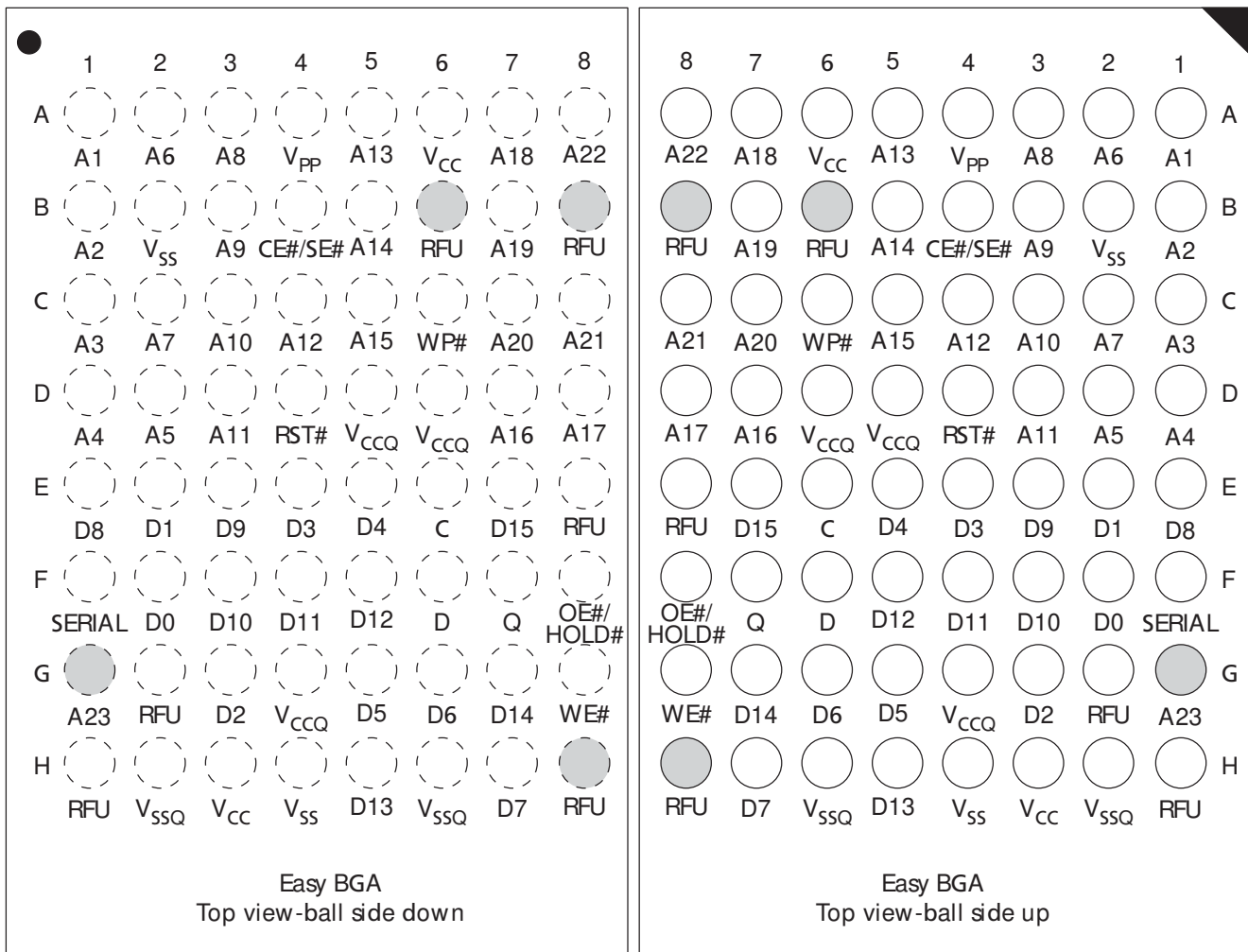
Pinouts and Ballouts

Figure 3: 56-Lead TSOP Pinout (128Mb)



- Notes: 1. A1 is the least significant address bit to be compatible with x8 addressing systems even though P8P parallel PCM is a 16-bit data bus.

Figure 4: 64-Ball Easy BGA Ballout (128Mb)



Notes: 1. A1 is the least significant address bit to be compatible with x8 addressing systems even though P8P parallel PCM is a 16-bit data bus.

Signal Names and Descriptions

Table 5: Ball/Pin Descriptions

Symbol	Type	Description
A[1:MAX:1]	Input	Address inputs: Device address inputs. 128Mb: A[23:1]. The address bus for TSOP and easy BGA starts at A1. P8P parallel PCM uses x16 addressing. The P8P parallel PCM package is x8 addressing and is compatible with J3 or P30 products.
DQ[15:0]	Input/ Output	Data input/outputs: Inputs data and commands during WRITES (internally latched). Outputs data during READ operations. Data signals float when CE# or OE# are V _{IH} or RST# is V _{IL} .
CE# or S#	Input	Chip enable: CE# LOW activates internal control logic, I/O buffers, decoders, and sense amps. CE# HIGH deselected the device, places it in standby state, and places data outputs at High-Z.
	SPI	SPI select: S# LOW activates WRITE commands to the SPI interface. Raising S# to V _{IH} completes (or terminates) the SPI command cycle; it also sets Q to High-Z.
OE# or HOLD#	Input	Output enable: Active LOW OE# enables the device's output data buffers during a READ cycle. With OE# at V _{IH} , device data outputs are placed in High-Z state.
	SPI	SPI HOLD#: When asserted, suspends the current cycle and sets Q to High-Z until de-asserted.
RST#	Input	Reset chip: When LOW, RST# resets internal automation and inhibits WRITE operations. This provides data protection during power transitions. RST# HIGH enables normal operation. The device is in 8-word page mode array read after reset exits.
WE#	Input	Write enable: controls command user interface (CUI) and array WRITES. Its rising edge latches addresses and data.
WP#	Input	Write protect: Disables/enables the lock-down function. When WP# is V _{IL} , the lock-down mechanism is enabled and software cannot unlock blocks marked lock-down. When WP# is V _{IH} , the lock-down mechanism is disabled and blocks previously locked-down are now locked; software can unlock and lock them. After WP# goes LOW, blocks previously marked lock-down revert to that state.
C	SPI	SPI clock: Synchronization clock for input and output data
D	SPI	SPI data input: Serial data input for op codes, address, and program data bytes. Input data is clocked in on the rising edge of C, starting with the MSB.
Q	SPI	SPI data output: Serial data output for read data. Output data is clocked out, triggered by the falling edge of C, starting with the MSB.
SERIAL	SPI	SPI enable: SERIAL is a port select switching between the normal parallel or serial interface. When V _{SS} the normal (non-SPI) P8P parallel PCM interface, is enabled, all other SPI inputs are "Don't Care," and Q is at High-Z. When V _{CC} SPI mode is enabled, all non-SPI inputs are "Don't Care," and all outputs are at High-Z. This pin has an internal weak pull-down resistor to select the normal parallel interface when users leave the pin floating. A CAM can be used to permanently disable this feature.
V _{PP}	Pwr	Erase and write power: A valid V _{PP} voltage enables erase or programming. Memory contents can't be altered when V _{PP} ≤ V _{PPLK} . Set V _{PP} = V _{CC} for in-system PROGRAM and ERASE operations. To accommodate resistor or diode drops from the system supply, the V _{IH} level of V _{PP} can be as low as V _{PPL,min} . Program/erase voltage is normally 1.7–3.6V.
V _{CC}	Pwr	Device power supply: WRITES are inhibited at V _{CC} ≤ V _{LKO} . Device operations at invalid V _{CC} voltages should not be attempted.
V _{CCQ}	Pwr	Output power supply: Enables all outputs to be driven at V _{CCQ} . This input may be tied directly to V _{CC} if V _{CCQ} is to function within the V _{CC} range.
V _{SS}	Pwr	Ground: Connects device circuitry to system ground.
V _{SSQ}	Pwr	I/O ground: Tie to GND.
NC		No connect: No internal connection; can be driven or floated.
DU		Don't use: Don't connect to power supply or other signals.
RFU		Reserved for future use: Don't connect to other signals.

Bus Operations

CE# at V_{IL} and RST# at V_{IH} enables device READ operations. Assume addresses are always valid. OE# LOW activates the outputs and gates selected data onto the I/O bus. WE# LOW enables device WRITE operations. When the V_{PP} voltage $\leq V_{PPLK}$ (lock-out voltage), only READ operations are enabled.

Table 6: Bus Operations

State	RST#	CE#	OE#	WE#	DQ[15:0]	Notes
READ (main array)	V_{IH}	V_{IL}	V_{IL}	V_{IH}	D_{OUT}	
READ (status, query, identifier)	V_{IH}	V_{IL}	V_{IL}	V_{IH}	D_{OUT}	
OUTPUT DISABLE	V_{IH}	V_{IL}	V_{IH}	V_{IH}	High-Z	
STANDBY	V_{IH}	V_{IH}	X	X	High-Z	2
RESET	V_{IL}	X	X	X	High-Z	2
WRITE	V_{IH}	V_{IL}	V_{IH}	V_{IL}	D_{IN}	1

- Notes:
1. See Table 8 on page 16 for valid D_{IN} during a WRITE operation.
 2. X = "Don't Care" (L or H).
 3. OE# and WE# should never be asserted simultaneously. If this occurs, OE# overrides WE#.

READ Operations

To perform a READ operation, RST# and WE# must be de-asserted while CE# and OE# are asserted. CE# is the device select control. When asserted, it enables the Flash memory device. OE# is the data output control. When asserted, the addressed Flash memory data is driven onto the I/O bus.

WRITE Operations

To perform a WRITE operation, both CE# and WE# are asserted while RST# and OE# are de-asserted. During a WRITE operation, address and data are latched on the rising edge of WE# or CE#, whichever occurs first. Table 7 on page 14 describes the bus cycle sequence for each of the supported device commands, and Table 8 on page 16 describes each command. See "AC Characteristics" on page 48 for signal timing details.

- Notes:
1. WRITE operations with invalid V_{CC} and/or V_{PP} voltages can produce spurious results and should not be attempted.

OUTPUT DISABLE Operations

When OE# is de-asserted, device outputs DQ[15:0] are disabled and placed in High-Z; WAIT is also placed in High-Z.

STANDBY Operations

When CE# is de-asserted, the device is deselected and placed in standby, substantially reducing power consumption. In standby, the data outputs are placed in High-Z, independent of the level placed on OE#. Standby current, I_{CCS} , is the average current measured over any 5ms time interval, 5 μ s after CE# is de-asserted. During standby, average current is measured over the same time interval 5 μ s after CE# is de-asserted.

When the device is deselected (while CE# is de-asserted) during a PROGRAM or ERASE operation, it continues to consume active power until the PROGRAM or ERASE operation is completed.

RESET Operations

As with any automated device, it is important to assert RST# when the system is reset. When the system comes out of reset, the system processor attempts to read from the Flash memory if it is the system boot device. If a CPU reset occurs with no Flash memory reset, improper CPU initialization may occur because the Flash memory may be providing status information rather than array data. Micron Flash memory devices enable proper CPU initialization following a system reset using the RST# input. RST# should be controlled by the same low true RESET signal that resets the system CPU.

After initial power-up or reset, the device defaults to asynchronous read array mode, and the status register is set to 0x80. Asserting RST# de-energizes all internal circuits and places the output drivers in High-Z. When RST# is asserted, the device shuts down the operation in progress, a process that takes a minimum amount of time to complete. When RST# has been de-asserted, the device is reset to asynchronous read array state.

Note: If RST# is asserted during a PROGRAM or ERASE operation, the operation is terminated, and the memory contents at the aborted location (for a PROGRAM) or block (for an ERASE) are no longer valid because the data may have been only partially written or erased.

When returning from a reset (RST# de-asserted), a minimum wait is required before the initial read access outputs valid data. Also, a minimum delay is required after a reset before a WRITE cycle can be initiated. After this wake-up interval passes, normal operation is restored. See “AC Characteristics” on page 48 for details about signal timing.

Command Set

Device Command Codes

The system CPU provides control of all in-system READ, WRITE, and ERASE operations of the device via the system bus. The on-chip write state machine (WSM) manages all block erase and word program algorithms.

Device commands are written to the command user interface (CUI) to control all Flash memory device operations. The CUI does not occupy an addressable memory location; it is the mechanism through which the Flash device is controlled.

Table 7: Command Codes and Descriptions

Mode	Code	Command	Description
Read	FFh	READ ARRAY	Places device in read array mode so that data signals output array data on DQ[15:0].
	70h	READ STATUS REGISTER	Places the device in status register read mode. Status data is output on DQ[7:0]. The device automatically enters this mode after a PROGRAM or ERASE command is issued to it.
	90h	READ ID CODE	Puts the device in read identifier mode. Device reads from the addresses output manufacturer/device codes, block lock status, or protection register data on DQ[15:0].
	98h	READ QUERY	Puts the device in read query mode. Device reads from the address given outputting the common Flash interface information on DQ[7:0].
	50h	CLEAR STATUS REGISTER	The WSM can set the status register's block lock (SR1), V _{PP} (SR3), program (SR4), and erase (SR5) status bits to 1, but cannot clear them. Device reset or the CLEAR STATUS REGISTER command at any device address clears those bits to 0.

Table 7: Command Codes and Descriptions (Continued)

Mode	Code	Command	Description
Program	40h	PROGRAM SET-UP	This preferred program command's first cycle prepares the CUI for a PROGRAM operation. The second cycle latches address and data and executes the WSM program algorithm at this location. Status register updates occur when CE# or OE# is toggled. A READ ARRAY command is required to read array data after programming.
	10h	ALT SET-UP	Equivalent to a PROGRAM SET-UP command (40h).
	42h	BIT-ALTERABLE WRITE	The command sequence is the same as WORD PROGRAM (40h). The difference is that the state of the PCM memory cell can change from a 0 to 1 or 1 to 0, unlike a Flash memory cell, which can only change from 1 to 0 during programming.
	E8h	BUFFERED PROGRAM	This command loads a variable number of bytes up to the buffer size 32 words onto the program buffer.
	EAh	BIT-ALTERABLE BUFFERED WRITE	This command sequence is similar to BUFFERED PROGRAM, but the BUFFER WRITE command is bit alterable or overwrite operation. The command sequence is the same as E8h.
	DEh	BUFFER PROGRAM ON ALL 1s	This command is the same as BUFFERED PROGRAM, but the user indicates that the page is already set to all 1s. The command sequence is the same as E8h.
	D0h	BUFFERED WRITE CONFIRM	The confirm command is issued after the data streaming for writing into the buffer is done. This initiates the WSM to carry out the buffered programming algorithm.
Erase	20h	BLOCK ERASE SET-UP	Prepares the CUI for block erase. The device emulates erasure of the block addressed by the ERASE CONFIRM command by writing all 1s. If the next command is not ERASE CONFIRM: The CUI sets status register bits SP4 and SP5 to 1. The CUI places the device in the read status register mode. The CUI waits for another command.
	D0h	ERASE CONFIRM	If the first command was ERASE SET-UP (20h), the CUI latches address and data, and then emulates erasure of the block indicated by the ERASE CONFIRM cycle address.
Suspend	B0h	WRITE SUSPEND or ERASE SUSPEND	This command issued at any device address initiates suspension of the currently executing PROGRAM/ERASE operation. The status register, invoked by a READ STATUS REGISTER command, indicates successful SUSPEND operation by setting status bits SP2 (write suspend) or SP6 (erase suspend) and SR7. The WSM remains in suspend mode regardless of the control signal states, except RST# = V _{IL} .
	D0h	SUSPEND RESUME	This command issued at any device address resumes suspended PROGRAM or ERASE operation.
Block Locking	60h	LOCK SET-UP	Prepares the CUI for lock configuration. If the next command is not BLOCK LOCK, UNLOCK, or LOCK-DOWN the CUI sets SP4 and SP5 to indicate command sequence error.
	01h	LOCK BLOCK	If the previous command was LOCK SET-UP (60h), the CUI locks the addressed block.
	D0h	UNLOCK BLOCK	After a LOCK SET-UP (60h) command, the CUI latches the address and unlocks the addressed block.
	2Fh	LOCK-DOWN	After a LOCK SET-UP (60h) command, the CUI latches the address and locks down the addressed block.
Protection	C0h	PROTECTION PROGRAM SET-UP	Prepares the CUI for a protection register program operation. The second cycle latches address and data and starts the WSM's protection register program or lock algorithm. Toggling CE# or OE# updates the PCM status register data. To read array data after programming, issue a READ ARRAY command.

Notes: 1. Do not use unassigned (reserved) commands.

Device Command Bus Cycles

Device operations are initiated by writing specific device commands to the CUI. Several commands, including WORD PROGRAM and BLOCK ERASE, are used to modify array data commands. Writing either command to the CUI initiates a sequence of internally timed functions that culminate in the completion of the requested task. However, the operation can be aborted either by asserting RST# or by issuing an appropriate SUSPEND command.

Table 8: Command Sequences in x16 Bus Mode

Mode	Command	Bus Cycles	First Bus Cycle			Second Bus Cycle		
			Oper	Addr ¹	Data ²	Oper	Addr ¹	Data ²
Read	READ ARRAY/RESET	1	WRITE	DnA	FFh	–	–	–
	READ DEVICE IDENTIFIERS	≥ 2	WRITE	DnA	90h	READ	DBA+IA	ID
	READ QUERY	≥ 2	WRITE	DnA	98h	READ	DBA+QA	QD
	READ STATUS REGISTER	2	WRITE	BA	70h	READ	BA	SRD
	CLEAR STATUS REGISTER	1	WRITE	X	50h	–	–	–
Program	PROGRAM	2	WRITE	WA	40h or 10h	WRITE	WA	WD
	BIT-ALTERABLE PROGRAM	2	WRITE	WA	42h	WRITE	PA	PD
	BUFFERED PROGRAM ³	> 2	WRITE	WA	E8h	WRITE	WA	N-1
	BIT-ALTERABLE BUFFERED PROGRAM ³	> 2	WRITE	WA	EAh	WRITE	WA	N-1
	BUFFERED PROGRAM ON ALL1s	> 2	WRITE	WA	DEh	WRITE	WA	N-1
Erase	BLOCK ERASE	2	WRITE	BA	20h	WRITE	BA	D0h
Suspend	PROGRAM/ERASE SUSPEND	1	WRITE	X	B0h	–	–	–
	PROGRAM/ERASE RESUME	1	WRITE	X	D0h	–	–	–
Block Lock	LOCK BLOCK	2	WRITE	BA	60h	WRITE	BA	01h
	UNLOCK BLOCK	2	WRITE	BA	60h	WRITE	BA	D0h
	LOCK-DOWN BLOCK	2	WRITE	BA	60h	WRITE	BA	2Fh
Protection	PROTECTION PROGRAM	2	WRITE	PA	C0h	WRITE	PA	PD
	LOCK PROTECTION PROGRAM	2	WRITE	LPA	C0h	WRITE	LPA	FFFDh

- Notes:
1. First command cycle address should be the same as the operation's target address.
 X = Any valid address within the device
 IA = Identification code address
 BA = Address within the block
 LPA = Lock protection address (from the CFI); P8P parallel PCM LPA is at 0080h
 PA = 4-word protection address in the user-programmable area of device identification plane
 DnA = Address within the device
 DBA = Device base address: (A[*MAX*:1] = 0h)
 PRA = Program region
 QA = Query code address
 WA = Word address of memory location to be written
 2. SRD = Data read from the status register
 WD = Data to be written at location WA
 ID = Identifier code data
 PD = User-programmable protection data
 QD = Query code data on DQ[7:0]
 N = Data count to be loaded into the device to indicate how many words would be written

into the buffer; because the internal registers count from 0, the user writes $N - 1$ to load N words.

3. The second cycle of the BUFFERED PROGRAM command, which is the count being loaded into the buffer, is followed by data streaming up to 32 words, and then a CONFIRM command is issued that triggers the programming operation. Refer to “Figure 33 on page 61.”

READ Operations

P8P parallel PCM has several read modes:

- **Read array mode** returns PCM array data from the addressed locations.
- **Read identifier mode** returns manufacturer device identifier data, block lock status, and protection register data.
- **Read query mode** returns device CFI (or query) data.
- **Read Status Register mode** returns the device status register data. A system processor can check the status register to determine the device’s state or to monitor program or erase progress.

READ ARRAY

The READ ARRAY command places (or resets) the device to read array mode. Upon initial device power-up or after reset (RST# transitions from V_{IL} to V_{IH}), the device defaults to read array mode. If an ERASE or PROGRAM SUSPEND command suspends the WSM, a subsequent READ ARRAY command will place the device in read array mode. The READ ARRAY command functions independently of V_{PP} voltage.

READ IDENTIFIER

The read identifier mode is used to access the manufacturer/device identifier, block lock status, and protection register codes. The identifier space occupies the address range supplied by the READ IDENTIFIER command (90h) address.

Table 9: Read Identifier Table

Parameter	Address ^{1, 2}	Data
Manufacturer code	DBA + 000000h	0089h
Device code	DBA + 000001h	ID (see Table 10 on page 18)
Block lock configuration	BBA + 000002h	Lock
Block Is unlocked		DQ0 = 0
Block Is locked		DQ0 = 1
Block Is not locked down		DQ1 = 0
Block Is locked down		DQ1 = 1
Reserved for future use		DQ[7:2]
Lock protection register 0		DBA + 000080h
64-bit factory-programmable protection register	DBA + 000081h–000084h	Protection register data
64-bit user-programmable protection register	DBA + 000085h–000088h	Protection register data
Lock protection register 1	DBA + 000089h	PR-LK1
16 x 128-bit user-programmable protection registers	DBA + 00008Ah–0000109h	Protection register data

Notes: 1. DBA = Device base address: (A[MAX:18] = DBA). Micron reserves other configuration address locations.

2. BBA = Block base address.

Table 10: Device Codes

Device	Device Code (Byte/Word)			Mode
	Hex	Binary		
		High Byte	Low Byte	
128Mb	881E	10001000	00011110	Top boot
128Mb	8821	10001000	00100001	Bottom boot

READ QUERY

The query space comes to the foreground and occupies the device address range supplied by the READ QUERY command address. The mode outputs CFI data when the device addresses are read. “Common Flash Interface” on page 73 describes the query mode information and addresses. Write the READ ARRAY command to return to read array mode. The read performance of this CFI data follows the same timings as the main array.

In addition to other ID mode data, the protection registers (such as block locking information and the device JEDEC ID) may be accessed as long as there are no ongoing WRITE or ERASE operations.

Query (CFI) data is read by sending the READ QUERY command to the device. Reading the query data is subject to the same restrictions as reading the protection registers.

PROGRAM Operations

Five WRITE operations are available in P8P parallel PCM.

- WORD PROGRAM (40h, or 10h)
- BIT-ALTERABLE WORD WRITE (42h)
- BUFFERED PROGRAM (E8h)
- BIT-ALTERABLE BUFFERED WRITE (EAh)
- BUFFERED PROGRAM ON ALL 1s (DEh)

Writing a PROGRAM command to the device initiates internally timed sequences that write the requested word. The WSM executes a sequence of internally timed events to write desired bits at the addressed location and to verify that the bits are sufficiently written. For word programming, the memory changes specifically addressed bits to 0; 1 bits do not change the memory cell contents. This enables individual data bits to be programmed (0) while 1 bits serve as data masks. For BIT-ALTERABLE WORD WRITE, the memory cell can change from 0 to 1 or 1 to a 0.

The status register can be examined for write progress and errors by reading any address within the device during a WRITE operation. Issuing a READ STATUS REGISTER command brings the status register to the foreground enabling write progress to be monitored or detected at other device addresses. Status register bit SR7 indicates device write status while the write sequence executes. CE# or OE# toggle (during polling) updates the status register. Valid commands that can be issued to the writing device during write include READ STATUS REGISTER, WRITE SUSPEND, READ IDENTIFIER, READ QUERY, and READ ARRAY; however, READ ARRAY will return unknown data while the device is busy.

When writing completes, status register bit SR4 indicates write success if zero (0) or failure if set (1). If SR3 is set (1), the WSM couldn't execute the WRITE command because V_{PP} was outside acceptable limits. If SR1 is set (1), the WRITE operation targeted a locked block and was aborted. Attempting to write in an erase suspended block will result in failure, and SR4 will be set (1).

After examining the status register, clear it by issuing the CLEAR STATUS REGISTER command before issuing a new command. The device remains in status register mode until another command is written to that device. Any command can follow after writing completes.

WORD PROGRAM

The system processor writes the WORD PROGRAM SETUP command (40h/10h) to the device followed by a second WRITE that specifies the address and data to be programmed. The device accessed during both of the command cycles automatically outputs status register data when the device address is read. The device accessed during the second cycle (the data cycle) of the program command sequence will be where the data is programmed. See Figure 33 on page 61.

When V_{PP} is greater than V_{PPLK} , program and erase currents are drawn through the V_{CC} input. If V_{PP} is driven by a logic signal, V_{PP} must remain above $V_{PP,min}$ to perform in-system PCM modifications. Figure 5 on page 22 shows PCM power supply usage in various configurations.

BIT-ALTERABLE WORD WRITE

The BIT-ALTERABLE WORD WRITE command executes just like the WORD PROGRAM command (40h/10h), using a two-write command sequence. The BIT-ALTERABLE WRITE SETUP command (42h) is written to the CUI, followed by the specific address and data to be written. The WSM will start executing the programming algorithm, but the data written to the CUI will be directly overwritten into the PCM memory. This is unlike Flash memory, which can only be written from 1 to 0 without a prior erase of the entire block. See Table 12 on page 21. This overwrite function eliminates Flash bit masking, which means that the software cannot use a 1 in a data mask to produce no change of the memory cell, as might occur with floating gate Flash.

BUFFERED PROGRAM

A BUFFERED PROGRAM command sequence initiates the loading of a variable number of words, up to the buffer size (32 words), into the program buffer and then into the PCM device. First, the BUFFERED PROGRAM SETUP command is issued along with the BLOCK ADDRESS (Figure 33 on page 61). When status register bit SR7 is set to 1, the buffer is ready for loading. Now a word count is given to the part with the block address.

On the next write, a device starting address is given along with the program buffer data. Subsequent writes provide additional device addresses and data, depending on the count. All subsequent addresses must lie within the starting address plus the buffer size. Maximum programming performance and lower power are obtained by aligning the starting address at the beginning of a 32-word boundary. A misaligned starting address is not allowed and results in invalid data. After the final buffer data is given, a PROGRAM BUFFER CONFIRM command is issued. This initiates the WSM to begin copying the buffer data to the PCM array.

If a command other than PROGRAM BUFFER CONFIRM command (D0h) is written to the device, an invalid command/sequence error will be generated, and status register bits SR5 and SR4 will be set to a 1. For additional buffer writes, issue another PROGRAM BUFFER SETUP command and check SR7. If an error occurs while writing, the device will stop writing, and status register bit SR4 will be set to a 1 to indicate a program failure. The internal WSM verify only detects errors for 1s that do not successfully program to 0s.

If a program error is detected, the status register should be cleared by the user before issuing the next PROGRAM command. Additionally, if the user attempts to program past the block boundary with a PROGRAM BUFFER command, the device will abort the PROGRAM BUFFER operation. This will generate an invalid command/sequence error and status register bits SR5 and SR4 will be set to a 1. All bus cycles in the buffered programming sequence should be addressed to the same block. If a buffered programming is attempted while the $V_{PP} \leq V_{PPLK}$, status register bits SR4 and SR3 will be set to 1.

Buffered write attempts with invalid V_{CC} and V_{PP} voltages produce spurious results and should not be attempted. Buffered program operations with $V_{IH} < RST\# < V_{HH}$ may produce spurious results and should not be attempted.

Successful programming requires that the addressed block's locking status to be cleared. If the block is locked down, then the WP# pin must be raised HIGH, and then the block could be unlocked to execute a PROGRAM operation. An attempt to program a locked block results in setting of SR4 and SR1 to a 1 (for example, error in programming).

BIT-ALTERABLE BUFFER WRITE

The BIT-ALTERABLE BUFFER WRITE command sequence is the same as for BUFFER PROGRAM. For command sequence, see “BUFFERED PROGRAM” on page 20. The primary difference between the two buffer commands is when the WSM starts executing, the data written to the buffer will be directly overwritten into the PCM memory, unlike Flash Memory, which can only go from 1 to 0 before an erase of the entire block. See Table 12 on page 21. This overwrite function eliminates Flash bit masking, which means software cannot use a 1 in a data mask for no change of the memory cell, as might occur with floating gate Flash.

The advantage of bit alterability is that no block erase is needed prior to writing a block, which minimizes system overhead for software management of data and ultimately improves latency and determinism and reduces power consumption because of reduction of system overhead. Storing counter variables can easily be handled using PCM memory because a 0 can change to a 1 or a 1 can change to a 0.

Table 11: Buffered Programming and Bit-Alterable Buffer Write Timing Requirements

Alignment	Programming Time	Example
32-word/64-byte aligned	^t PROG/PB	Start address = 1FFF10h; end address = 1FFF2Fh

Table 12: Bit Alterability vs. Flash Bit-Masking

Programming Function	Command Issued	Memory Cell Current State	Data From User	Memory Cell After Programming
Flash bit masking	40h or E8h	0	0	0
	40h or E8h	0	1	0
	40h or E8h	1	0	0
	40h or E8h	1	1	1
Bit alterability	42h or EAh	0	0	0
	42h or EAh	0	1	1
	42h or EAh	1	0	0
	42h or EAh	1	1	1

BIT-ALTERABLE BUFFER PROGRAM

This mode is sometimes referred to as PRESET BUFFERED PROGRAM.

PROGRAM ON ALL 1s is similar to program mode (1s treated as masks; 0s written to cells) with the assumption that all the locations in the addressed page have previously been set (1s). Performance of BUFFER PROGRAM ON ALL 1s expected to be better than buffered program mode because the pre-read step before programming is eliminated. The command sequence for BUFFERED PROGRAM ON ALL 1s is the same as BUFFERED PROGRAM command (E8h).

PROGRAM SUSPEND

Issuing the PROGRAM SUSPEND command while programming suspends the programming operation. This enables data to be accessed from the device other than the one being programmed. The PROGRAM SUSPEND command can be issued to any device address. A PROGRAM operation can be suspended to perform reads only. Additionally, a PROGRAM operation that is running during an ERASE SUSPEND can be suspended to perform a READ operation.