JESD 21-C Section Title: Annex M: Serial Presence Detect (SPD) for LPDDR3 and LPDDR4 SDRAM Modules

Committee Document Reference Title: LPDDR3 and LPDDR4 SPD Document Release 2

Device Type Identifier: LPDDR3-DIMM Revision 1.1
LPDDR4-DIMM Revision 1.1

1 Scope

This Annex describes the serial presence detect (SPD) values for all LPDDR modules. Differences between module types are encapsulated in subsections of this annex. These presence detect values are those referenced in the SPD standard document for 'Specific Features'. The following SPD fields will be documented in the order presented in Section 2, with the exception of bytes 128~255 which are documented in separate sections, one for each family of module types. Further description of Byte 2 is found in Annex A of the SPD standard.

All unused entries will be coded as 0x00. All unused bits in defined bytes will be coded as 0 except where noted.

Timing parameters in the SPD represent the operation of the module including all Drams and support devices at the lowest supported supply voltage (see SPD byte 11), and are valid from t_{CKAVG}min to t_{CKAVG}max (see SPD bytes 18 and 19).

To allow for maximum flexibility as devices evolve, SPD fields described in this document may support device configuration and timing options that are not included in the JEDEC LPDDR SDRAM data sheet (JESD79-4). Please refer to DRAM supplier data sheets or JESD79-4 to determine the compatibility of components.

2 Address map

The following is the SPD address map for all LPDDR modules. It describes where the individual lookup table entries will be held in the serial EEPROM. Consistent with the definition of LPDDR generation SPD devices which have four individual write protection blocks of 128 bytes in length each, the SPD contents are aligned with these blocks as shown in Table 1.

Table 1 — Address Map

Block	Range		Description
0	0~127	0x000~0x07F	Base Configuration and DRAM Parameters
1	128~255	0x080~0x0FF	Module Specific Parameters -- See annexes for details
2	256~319	0x100~0x13F	Hybrid Memory Parameters
	320~383	0x140~0x17F	Manufacturing Information
3	384~511	0x180~0x1FF	End User Programmable

2.1 Block Descriptions

2.1.1 Block 0: Base Configuration and DRAM Parameters

Table 2 details the location of each byte in this block.

Table 2 — Block 0: Base Configuration and DRAM Parameters

Byte Number		Function Described	Notes
0	0x000	Number of Serial PD Bytes Written / SPD Device Size	1, 2
1	0x001	SPD Revision	
2	0x002	Key Byte / DRAM Device Type	
3	0x003	Key Byte / Module Type	
4	0x004	SDRAM Density and Banks	3
5	0x005	SDRAM Addressing	3

NOTE 1 Number of SPD bytes written will typically be programmed as 384 bytes.
NOTE 2 Size of SPD device will typically be programmed as 512 bytes.
NOTE 3 From LPDDR SDRAM data sheet.
NOTE 4 These are optional, in accordance with the JEDEC specification.

2.1.1 Block 0: Base Configuration and DRAM Parameters (Cont'd)

Table 2 — Block 0: Base Configuration and DRAM Parameters (Cont'd)

Byte Number		Function Described	Notes
6	0x006	SDRAM Package Type	3
7	0x007	SDRAM Optional Features	3
8	0x008	SDRAM Thermal and Refresh Options	3
9	0x009	Other SDRAM Optional Features	3
10	0x00A	Reserved -- must be coded as 0x00	
11	0x00B	Module Nominal Voltage, VDD	3
12	0x00C	Module Organization	
13	0x00D	Bus Width	
14	0x00E	Module Thermal Sensor	
15	0x00F	Extended module type	
16	0x010	Signal Loading	3
17	0x011	Timebases	
18	0x012	SDRAM Minimum Cycle Time (t_{CKAVG}min)	3
19	0x013	SDRAM Maximum Cycle Time (t_{CKAVG}max)	3
20	0x014	CAS Latencies Supported, First Byte	3
21	0x015	CAS Latencies Supported, Second Byte	3
22	0x016	CAS Latencies Supported, Third Byte	3
23	0x017	CAS Latencies Supported, Fourth Byte	3
24	0x018	Minimum CAS Latency Time (t_{AA}min)	3
25	0x019	Read & Write Latency Set Options	
26	0x01A	Minimum RAS to CAS Delay Time (t_{RCD}min)	3
27	0x01B	Minimum Row Precharge Delay Time (t_{RPab}min)	3
28	0x01C	Minimum Row Precharge Delay Time (t_{RPpb}min)	
29	0x01D	Minimum Refresh Recovery Delay Time (t_{RFCab}min), LSB	3
30	0x01E	Minimum Refresh Recovery Delay Time (t_{RFCab}min), MSB	3
31	0x01F	Minimum Refresh Recovery Delay Time (t_{RFCpb}min), LSB	3
32	0x020	Minimum Refresh Recovery Delay Time (t_{RFCpb}min), MSB	3
33~59	0x029~0x03B	Reserved -- must be coded as 0x00	
60~77	0x03C~0x04D	Connector to SDRAM Bit Mapping	
78~119	0x04E~0x074	Reserved -- must be coded as 0x00	
120	0x078	Fine Offset for Minimum Row Precharge Delay Time (t_{RPpb}min)	3
121	0x079	Fine Offset for Minimum Row Precharge Delay Time (t_{RPab}min)	3
122	0x07A	Fine Offset for Minimum RAS to CAS Delay Time (t_{RCD}min)	3
123	0x07B	Fine Offset for Minimum CAS Latency Time (t_{AA}min)	3
124	0x07C	Fine Offset for SDRAM Maximum Cycle Time (t_{CKAVG}max)	3
125	0x07D	Fine Offset for SDRAM Minimum Cycle Time (t_{CKAVG}min)	3
126	0x07E	CRC for Base Configuration Section, Least Significant Byte	
127	0x07F	CRC for Base Configuration Section, Most Significant Byte	

NOTE 1 Number of SPD bytes written will typically be programmed as 384 bytes.
NOTE 2 Size of SPD device will typically be programmed as 512 bytes.
NOTE 3 From LPDDR SDRAM data sheet.
NOTE 4 These are optional, in accordance with the JEDEC specification.

2.1.2 Block 1: Module Specific Parameters

Bytes 128~255 (0x080~0x0FF). Parameters in this block are specific to the module type as selected by the contents of SPD Key Byte 3 bits 3~0. Refer to the appropriate annex for detailed byte descriptions.

2.1.3 Block 2, lower half: Hybrid Memory Parameters

Bytes 256~319 (0x100~0x13F). Parameters in this block are specific to the hybrid memory type as selected by the contents of SPD Key Byte 3 bits 7~4. Refer to the appropriate annex for detailed byte descriptions.

2.1.4 Block 2, upper half: Manufacturing Information

Bytes 320~383 (0x140~0x17F). Table 3 details the location of each byte in this block.

Table 3 — Block 2, upper half: Manufacturing Information

Byte Number		Function Described	Notes
320	0x140	Module Manufacturer's ID Code, Least Significant Byte	
321	0x141	Module Manufacturer's ID Code, Most Significant Byte	
322	0x142	Module Manufacturing Location	
323~324	0x143~0x144	Module Manufacturing Date	
325~328	0x145~0x148	Module Serial Number	
329~348	0x149~0x15C	Module Part Number	
349	0x15D	Module Revision Code	
350	0x15E	DRAM Manufacturer's ID Code, Least Significant Byte	
351	0x15F	DRAM Manufacturer's ID Code, Most Significant Byte	
352	0x160	DRAM Stepping	
353~381	0x161~0x17D	Module Manufacturer's Specific Data	
382~383	0x17E~0x17F	Reserved; must be coded as 0x00	

2.1.5 Block 3: End User Programmable

Bytes 384~511 (0x180~0x1FF) Bytes in this block are reserved for use by end users.

3 Details of Each Byte

3.1 General Configuration Section: Bytes 0~127 (0x000~0x07F)

This section contains defines parameters that are common to all LPDDR module types.

3.1.1 Byte 0 (0x000): Number of Bytes Used / Number of Bytes in SPD Device

Byte 0 (0x000) is shown in Table 4. The least significant nibble of this byte describes the total number of bytes used by the module manufacturer for the SPD data and any (optional) specific supplier information. The byte count includes the fields for all required and optional data. Bits 6~4 describe the total size of the serial memory used to hold the Serial Presence Detect data.

Table 4 — Byte 0 (0x000): Number of Bytes Used / Number of Bytes in SPD Device

Bit 7	Bits 6~4	Bits 3~0
Reserved	**SPD Bytes Total**	**SPD Bytes Used**
Reserved; must be coded as 0	Bit [6, 5, 4] : 000 = Undefined 001 = 256 010 = 512 All others reserved	Bit [3, 2, 1, 0] : 0000 = Undefined 0001 = 128 0010 = 256 0011 = 384 0100 = 512 All others reserved

NOTE Typical programming of bits 3~0 will be 0011 (384 bytes).

3.1.2 .Byte 1 (0x001): SPD Revision

This byte, shown in Table 5, describes the compatibility level of the encoding of the bytes contained in the SPD EEPROM, and the current collection of valid defined bytes. Software should examine the upper nibble (Encoding Level) to determine if it can correctly interpret the contents of the module SPD. The lower nibble (Additions Level) can optionally be used to determine which additional bytes or attribute bits have been defined; however, since any undefined additional byte must be encoded as 0x00 or undefined attribute bit must be defined as 0, software can safely detect additional bytes and use safe defaults if a zero encoding is read for these bytes.

Table 5 — Byte 1 (0x001): SPD Revision

Production Status	SPD Revision	Encoding Level				Additions Level				Hex
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
Pre-production	Revision 0.0	0	0	0	0	0	0	0	0	00
	Revision 0.1	0	0	0	0	0	0	0	1	01
	...	.	.	.	.	.	.	.	.	.
	Revision 0.9	0	0	0	0	1	0	0	1	09
Production	Revision 1.0	0	0	0	1	0	0	0	0	10
	Revision 1.1	0	0	0	1	0	0	0	1	11
	...	.	.	.	.	.	.	.	.	...
Undefined	Undefined	1	1	1	1	1	1	1	1	FF

The Additions Level is never reduced even after an increment of the Encoding Level. For example, if the current SPD revision level were 1.2 and a change in Encoding Level were approved, the next revision level would be 2.2. If additions to revision 2.2 were approved, the next revision would be 2.3. Changes in the Encoding Level are extremely rare, however, since they can create incompatibilities with older systems.

The exceptions to the above rule are the SPD revision levels used during development prior to the Revision 1.0 release. Revisions 0.0 through 0.9 are used to indicate sequential pre-production SPD revision levels, however the first production release will be Revision 1.0.

This document allows for SPD contents for multiple families of LPDDR memory modules, with a separate annex for each family that defines the bytes in SPD locations 128~255 (0x080~0x0FF). These module families and their respective annexes are:

- Annex M-1: LP-DIMM Revision 1.0
- Annex M-2: LPDDR3-DIMM Revision 1.1
  LPDDR4-DIMM Revision 1.1

The SPD revision level for each module family type is independent. This allows changes to be made to a new module family annex, for example, without necessarily changing the revision of LP-DIMMs. In this context, the SPD revision value corresponds to all SPD bytes *for that DIMM type*. It also means that over time, the revisions for each module type may vary. Note that changes to a DIMM specific annex does not affect the revisions of other module types, but changes in the General Section of the SPD affect all DIMM types. The example in Table 6 suggests a possible historical progression:

Table 6 — Hypothetical Historic Progression of SPD Revisions by DIMM Type

Event	LP-DIMM
Pre-standard SPD release	0.5
Initial SPD release	1.0
SPD additions made	1.1
SPD encoding change made	2.1

3.1.3 Byte 2 (0x002): Key Byte / DRAM Device Type

This byte, shown in Table 7, is the key byte used by the system BIOS to determine how to interpret all other bytes in the SPD EEPROM. The BIOS must check this byte first to ensure that the EEPROM data is interpreted correctly. Any DRAM or Module type that requires significant changes to the SPD format (beyond defining previously undefined bytes or bits) also requires a new entry in the key byte table below.

Table 7 — Byte 2 (0x002): Key Byte / DRAM Device Type

Line #	Hex	SDRAM / Module Type Corresponding to Key Byte
0	00	Reserved
1	01	Fast Page Mode
2	02	EDO
3	03	Pipelined Nibble
4	04	SDRAM
5	05	ROM
6	06	DDR SGRAM
7	07	DDR SDRAM
8	08	DDR2 SDRAM
9	09	DDR2 SDRAM FB-DIMM
10	0A	DDR2 SDRAM FB-DIMM PROBE
11	0B	DDR3 SDRAM
12	0C	DDR4 SDRAM
13	0D	Reserved
14	0E	DDR4E SDRAM
15	0F	LPDDR3 SDRAM
16	10	LPDDR4 SDRAM
-	-	-
253	FD	Reserved
254	FE	Reserved
255	FF	Reserved

The DRAM Device Type byte defines an interface compatibility family more than it identifies a specific memory device. For example, where an LP memory module may expose the devices directly to the edge connector contacts, a load reduced memory module presents the interfaces of registering clock drivers and data buffers to the edge connector contacts. There may be memory modules with completely different devices, such as non-volatile memories, however these may define themselves as a base memory type such as LPDDR SDRAM for interface compatibility with the memory controller. The memory controller must be aware of its capabilities when parsing the SPD.

3.1.4 Byte 3 (0x003): Key Byte / Module Type

This byte, shown in Table 8, is a Key Byte used to index the module specific section of the SPD from bytes 128~255 and bytes 256~319. Byte 3 bits 3~0 identifies the SDRAM memory module type.

Bits 7~4 identifies the architecture of the secondary memory type, if any, present on the module in addition to the base DRAM. These secondary hybrid architectures may apply to any Base Module Type. The index in bits 7~4 is used to index the hybrid memory architecture specific section of the SPD from bytes 256~319.

3.1.4 Byte 3 (0x003): Key Byte / Module Type (Cont'd)

System software supporting hybrid memory modules should parse the DRAM attributes per the Base Module Type and combine the indexed information per the Hybrid Memory Type to determine the combined memory module capabilities.

Where base memory parameters apply to this class of hybrid module, these will be documented with those bytes in the base section. Extended parameters are encoded in SPD bytes 256~319. When a module has no base memory present, the special coding 1111 shall be used in bits 3~0 of byte 2.

Table 8 — Byte 3 (0x003): Key Byte / Module Type

Bit 7	Bits 6~4	Bits 3~0
Hybrid	**Hybrid Media**	**Base Module Type**
0 = Not hybrid (Module is DRAM only) 1 = Hybrid module (See bits 6~4 for hybrid type)	Bits [6, 5, 4]: 000 = Not hybrid 001 = Reserved All other codes reserved	Bits [3, 2, 1, 0]: 0000 = Extended DIMM type, see byte 15 (0x00F) 0111 = LP-DIMM 1110 = Non-DIMM Solution 1111 = Reserved
Base Module Type Definitions: LP-DIMM: Low Power Unbuffered Small Outline Dual In-Line Memory Module, 64-bit data bus Non-DIMM Solution: Used to define when the SDRAM Devices may be placed on the main board rather than on a module. NOTE No hybrid LPDDR compatible modules are defined at this time.		

Examples:

0x07 = LP-DIMM, no hybrid memory present

0x0E = soldered down memory, no hybrid memory present

NOTE LP-DIMMs incorporate two independent 32 bit channels per module. Non-DIMM solutions such as solder down applications may incorporate any number of channels, yet use SPD codes to simplify software configuration for the application.

3.1.5 Byte 4 (0x004): SDRAM Density and Banks

This byte, shown in Table 9, defines the total density of each LPDDR SDRAM, in bits, and the number of internal banks and bank groups into which the memory array is divided. These values come from the LPDDR SDRAM data sheet.

Table 9 — Byte 4 (0x004): SDRAM Density and Banks

Bits 7~6	Bits 5~4	Bits 3~0
Bank Group Bits	**Bank Address Bits[1]**	**Total SDRAM capacity, per die, in Gigabits**
Bits [7, 6]: 00 = 0 (no bank groups) 01 = 1 (2 bank groups) 10 = 2 (4 bank groups) 11 = reserved	Bit [5, 4] : 00 = 2 (4 banks) 01 = 3 (8 banks) All others reserved	Bit [3, 2, 1, 0] : 0000 = Reserved 0001 = Reserved 0010 = 1 Gb 0011 = 2 Gb 0100 = 4 Gb 0101 = 8 Gb 0110 = 16 Gb 0111 = 32 Gb 1000 = 12 Gb 1001 = 24 Gb 1010 = 3 Gb 1011 = 6 Gb 1100 = 18 Gb All others reserved

3.1.6 Byte 5 (0x005): SDRAM Addressing

This byte, shown in Table 10, describes the row addressing and the column addressing in each SDRAM device. Bits 2~0 encode the number of column address bits, and bits 5~3 encode the number of row address bits. These values come from the LPDDR SDRAM data sheet.

Table 10 — Byte 5 (0x005): SDRAM Addressing

Bits 7~6	Bits 5~3	Bits 2~0
Reserved	**Row Address Bits**	**Column Address Bits**
Reserved; must be coded as 00	Bit [5, 4, 3] : 000 = 12 001 = 13 010 = 14 011 = 15 100 = 16 101 = 17 110 = 18 All others reserved	Bit [2, 1, 0] : 000 = 9 001 = 10 010 = 11 011 = 12 All others reserved

3.1.7 Byte 6 (0x006): SDRAM Package Type

This byte, shown in Table 11, describes the type of SDRAM Device on the module.

Table 11 — Byte 6 (0x006): SDRAM Package Type

Bit 7	Bits 6~4	Bits 3~2	Bit 1~0
SDRAM Package Type	**Die per SDRAM Package**	**Channels per SDRAM Package**	**Signal Loading Index[2]**
0 = Monolithic DRAM Device 1 = Non-Monolithic Device[1]	000 = 1 die 001 = 2 die 010 = 3 die 011 = 4 die 100 = 5 die 101 = 6 die 110 = 7 die 111 = 8 die All others reserved.	00 = 1 Channel 01 = 2 Channels 10 = 4 Channels 11 = Reserved	00 = Not specified 01 = Byte 16 Signal Loading Matrix 1 10 = Reserved 11 = Byte 16 Signal Loading Matrix 2

NOTE 1 This includes Dual Die, Quad Die, Multi-Die, or physically stacked devices - anything that is outside the standard monolithic device
NOTE 2 Index into SPD byte 16, Signal Loading. Selects one matrix of signal loads.
NOTE 3 Monolithic devices (bit 7 = 0) should code bits 1~0 as 00.

3.1.8 Byte 7 (0x007): SDRAM Optional Features

This byte, shown in Table 12, defines support for certain SDRAM features. This value comes from the LPDDR SDRAM data sheet.

Table 12 — Byte 7 (0x007): SDRAM Optional Features

Bits 7~6	Bits 5~4	Bits 3~0
Reserved	**Maximum Activate Window (tMAW)**	**Maximum Activate Count (MAC)**
Reserved; must be coded as 00	Bits [5, 4]: 00 = 8192 * tREFI 01 = 4096 * tREFI 10 = 2048 * tREFI 11 = Reserved	Bits [3, 2, 1, 0] : 0000 = Untested MAC[1] 0001 = 700 K 0010 = 600 K 0011 = 500 K 0100 = 400 K 0101 = 300 K 0110 = 200 K 0111 = Reserved 1000 = Unlimited MAC[2] All others reserved

NOTE 1 Untested MAC means the device is not tested for tMAW and/or MAC; no particular value should be assumed.
NOTE 2 Unlimited MAC means there is no restriction to the number of activates to a given row in a refresh period providing DRAM timing requirements such as tRCmin and refresh requirements are not violated.

3.1.9 Byte 8 (0x008): SDRAM Thermal and Refresh Options

This byte, shown in Table 13, describes the module's supported operating temperature ranges and refresh options. These values come from the LPDDR SDRAM data sheet. Please refer to the LPDDR SDRAM data sheet (JESD79-4 or supplier data sheet) for a complete description of these options.

Table 13 — Byte 8 (0x008): SDRAM Thermal and Refresh Options

Bits 7~0
Reserved
Reserved; must be coded as 0x00

3.1.10 Byte 9 (0x009): Other SDRAM Optional Features

This byte, shown in Table 14, defines support for certain SDRAM features. This value comes from the LPDDR SDRAM data sheet.

Table 14 — Byte 9 (0x009): Other SDRAM Optional Features

Bits 7~6	Bit 5	Bits 4~0
Post Package Repair (PPR)	**Soft PPR**	**Reserved**
00: PPR not supported 01: Post package repair supported, one row per bank group 10: Reserved 11: Reserved	0 = Soft PPR not supported 1 = Soft PPR supported	Reserved; must be coded as 00000

3.1.11 Byte 10 (0x00A):

Reserved, must be coded as 0x00

3.1.12 Byte 11 (0x00B): Module Nominal Voltage, VDD

Reserved, must be coded as 0x00

3.1.13 Byte 12 (0x00C): Module Organization

This byte, shown in Table 15, describes the organization of the SDRAM module. Bits 2~0 encode the device width of the SDRAM devices. Bits 5~3 encode the number of package ranks on the module

Table 15 — Byte 12 (0x00C): Module Organization

Bit 7	Bit 6	Bits 5~3	Bits 2~0
Reserved	**Byte Mode Identification**	**Package Ranks per Channel**	**Device Data Width Per Channel**
Reserved; must be coded as 0	Bit [6]: 0 = Standard Device 1 = Byte Mode Device	Bit [5, 4, 3] : 000 = 1 Package Rank 001 = 2 Package Ranks 010 = 3 Package Ranks 011 = 4 Package Ranks All others reserved	Bit [2, 1, 0] : 000 = 4 bits 001 = 8 bits 010 = 16 bits 011 = 32 bits All others reserved

“Package ranks per channel” refers to the number of common chip select signals across the data width of the channel such that assertion of each chip select enables a number of DRAM devices equivalent to the width of the channel.
LP-DIMMs incorporate two independent channels per module, each with a unique set of chip selects; bits 5~3 represent the number of chip selects on each channel.

With the advent of LPDDR4, a new feature call byte mode was added to the device that allows half of the I/O’s to be shared with another die, forming a single x16 channel. This function allowed for doubling the capacity of the package without adding more loading on the DQ lines. Since there are multiple configuration differences such as ZQ calibration when in Byte mode, it is important for systems to know about this bit.

3.1.14 Byte 13 (0x00D): Bus Width

This byte, shown in Table 16, describes the width of the SDRAM memory bus for each channel. Bits 2~0 encode the primary bus width. Bits 4~3 encode the bus width extension such as parity or ECC (not currently supported for LPDDR solutions). Bits 7~5 describe the number of LPDDR channels represented by this SPD.

Table 16 — Byte 13 (0x00D): Bus Width

Bits 7~5	Bits 4~3	Bits 2~0
Number of System Channels[2]	**Bus Width Extension[1], in bits**	**System Channel Bus Width[2], in bits**
Bits [7, 6, 5]: 000 = 1 channel 001 = 2 channels 010 = 3 channels 011 = 4 channels 100 = 8 channels All others reserved	Bit [4, 3] : 00 = 0 bits (no extension) 01 = Reserved All others reserved	Bit [2, 1, 0] : 000 = 8 bits 001 = 16 bits 010 = 32 bits 011 = 64 bits All others reserved

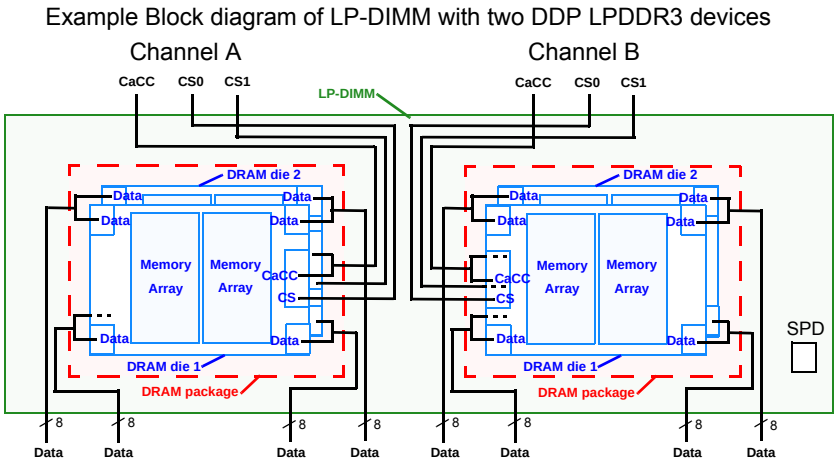
NOTE 1 LPDDR modules have no ECC option defined.
NOTE 2 Standard LP-DIMMs define two 32 bit wide channels per module.
NOTE 3 Solution width (e.g., DIMM width) = (Channel Bus Width + Bus Width Extension) * Number of Channels

Examples:

- One 32 bit channel, no parity or ECC (32 bits total width): 000 000 010
- Two 32 bit channels, no parity or ECC (64 bits total width): 001 000 010
- One 64 bit channel, no parity or ECC (64 bits total width): 000 000 011

3.1.14.1 Calculating Total Solution Capacity

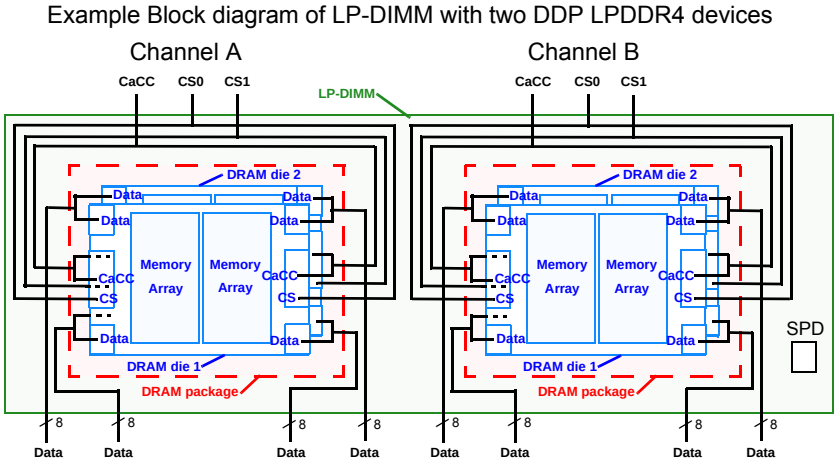
The LPDDR SPD supports solder down or module solutions, so calculating total solution capacity combines calculating the per-channel capacity multiplied by the number of channels. The SPD assumes that all channels have equivalent capacity and configuration. JEDEC standard LPDDR memory modules have two 32-bit channels, so the configuration of such modules is used to describe the terminology used to calculate module capacity, as shown in Figures 1 and 2.



Number of die per package = 2
Number of chip selects per channel = 2
Number of package ranks per channel = 2
SDRAM device width = 32
Channel bus width = 32
Number of channels = 2

CaCC = Command, address, control, clocks
CS = chip select

Figure 1 — Example Block Diagram of LP-DIMM with two DDP LPDDR3 Devices



Number of die per package = 2
Number of chip selects per channel = 2
Number of package ranks per channel = 2
SDRAM device width = 32
Channel bus width = 32
Number of channels = 2

CaCC = Command, address, control, clocks
CS = chip select

Figure 2 — Example Block Diagram of LP-DIMM with two DDP LPDDR4 Devices

The total memory capacity of the module may be calculated from SPD values.

For example, to calculate the total capacity, in megabytes or gigabytes, of a typical module:

Per Channel Capacity = (SDRAM Capacity Per Die ÷ 8) * Number of Die per Package * Channel Bus Width ÷ (SDRAM Device Data Width Per Channel * Channels Per SDRAM Package)

Total Solution Capacity = Per Channel Capacity * Number of Channels

where:

- SDRAM Capacity Per Die = SPD byte 4 bits 3~0
- Number of Die per SDRAM Package = SPD byte 6 bits 6~4
- System Channel Bus Width = SPD byte 13 bits 2~0
- SDRAM Device Data Width Per Channel = SPD byte 12 bits 2~0
- Number of System Channels = SPD byte 13 bits 7~5
- Channels Per SDRAM Package = SPD byte 6 bits 3~2

Examples:

LP-DIMM system with two 32 bit channels, each channel has one DDP SDRAM (2 die per package), each SDRAM die having 4 Gb of data, with one 32 bit wide data interface per SDRAM package:

- Per channel capacity = (4 Gb ÷ 8) * 2 * 32 ÷ (32 * 1) = 1 GB
- Total solution capacity = 1 GB * 2 = 2 GB

Solder-down solution with three 32 bit channels, each channel has two SDP SDRAMs (1 die per package), each SDRAM die having 8 Gb of data, with 16 bit wide data interface per SDRAM package:

- Per channel capacity = (8 Gb ÷ 8 * 1 * 32 ÷ (16 * 1) = 2 GB
- Total solution capacity = 2 GB * 3 = 6 GB

3.1.15 Byte 14 (0x00E): Module Thermal Sensor

This byte, shown in Table 17, describes the module's supported thermal options.

Table 17 — Byte 14 (0x00E): Module Thermal Sensor

Bit 7	Bits 6~0
Thermal Sensor[1]	**Reserved**
0 = Thermal sensor not incorporated onto this assembly 1 = Thermal sensor incorporated onto this assembly	0 = Undefined All others reserved

NOTE 1 Thermal sensor compliant with TSE2004av specifications.

3.1.16 Byte 15 (0x00F): Extended Module Type

This byte, shown in Table 18, extends the module type field of byte 3. Used when byte 3 bits 3~0 = 0000.

Table 18 — Byte 15 (0x00F): Extended Module Type

Bit 7~4	Bits 3~0
Reserved	**Extended Base Module Type**
Reserved; must be coded as 0000	Bits [3, 2, 1, 0]: 0000 = Reserved; must be coded as 0000 ... 1111 = Reserved; must be coded as 0000

3.1.17 Byte 16 (0x010): Signal Loading

This byte, shown in Table 19, defines the number of loads placed on significant groups of signals in the memory solution. The signal groups are data/strobe/mask, command/address/control/clock, and chip select. This byte is indexed from SPD byte 6 bits 1~0 to allow for a variety of loading matrices. LPDDR4 devices with two channels per device that are driven simultaneously, as in JEDEC standard LP-DIMMs, count each device load separately.

Table 19 — Byte 16 (0x010): Signal Loading

Signal Loading Matrix 1 (SPD byte 6 bits 1~0 = 01)		
Bits 7~6	**Bits 5~3**	**Bits 2~0**
Data/Strobe/Mask Loading	**Command/Address/Control/Clock Loading**	**Chip Select Loading**
Bits [7, 6]: 00 = 1 load 01 = 2 loads 10 = 4 loads 11 = Reserved	Bits [5, 4, 3]: 000 = 1 load 001 = 2 loads 010 = 4 loads 011 = 8 loads All other codes reserved	Bits [2, 1, 0]: 000 = 1 load 001 = 2 loads 010 = 4 loads 011 = 8 loads All other codes reserved
Signal Loading Matrix 2 (SPD byte 6 bits 1~0 = 11)		
Bits 7~6	**Bits 5~3**	**Bits 2~0**
Data/Strobe/Mask Loading	**Command/Address/Control/Clock Loading**	**Chip Select Loading**
Reserved	Reserved	Reserved

3.1.18 Byte 17 (0x011): Timebases

This byte, shown in Table 20, defines a value in picoseconds that represents the fundamental timebase for fine grain and medium grain timing calculations. These values are used as a multiplier for formulating subsequent timing parameters.

Table 20 — Byte 17 (0x011): Timebases

Bits 7~4	Bits 3~2	Bits 1~0
Reserved	**Medium Timebase (MTB)**	**Fine Timebase (FTB)**
Reserved; must be coded as 0000	Bits [3, 2]: 00 = 125 ps All others reserved	Bits [0, 1]: 00 = 1 ps All others reserved

3.1.18.1 Relating the MTB and FTB

When a timing value tXX cannot be expressed by an integer number of MTB units, the SPD must be encoded using both the MTB and FTB. The Fine Offsets are encoded using a two's complement value which, when multiplied by the FTB yields a positive or negative correction factor. Typically, for safety and for legacy compatibility, the MTB portion is rounded UP and the FTB correction is a negative value. The general algorithm for programming SPD values is:

```
Temp_val = tXX / MTB                          // Calculate as real number
Remainder = Temp_val modulo 1                 // Determine if integer # MTBs
Fine_Correction = 1 - Remainder               // If needed, what correction
if (Remainder == 0) then                      // Integer # MTBs?
        tXX(MTB) = Temp_val                   // Convert to integer
        tXX(FTB) = 0                          // No correction needed
else                                          // Needs correction
        tXX(MTB) = ceiling (Temp_val)         // Round up for safety in legacy systems
        tXX(FTB) = Fine_Correction * MTB / FTB // Correction is negative offset
endif
```

To recalculate the value of tXX from the SPD values, a general formula BIOSes may use is:

```
tXX = tXX(MTB) * MTB + tXX(FTB) * FTB
```

Example (see Table 21):

Table 21 — Example: Recalculation of MTB and FTB

t_{CKAVG}min SPD Calculations Using MTB and FTB			
Speed Bin	**t_{CKAVG}min Value Decimal**	**SPD byte 18 Decimal (Hexadecimal)**	**SPD byte 125 Decimal (Hexadecimal)**
LPDDR3-1866	1.071 ns	9 (0x09)	-54 (0xCA)
	=	(9 * 0.125) + (-54 * 0.001)	
NOTE Examples assume MTB of 0.125 ns and FTB of 0.001 ns			

Timing parameters using both MTB and FTB are shown in Table 22:

Table 22 — Timing Parameters using both MTB and FTB

Parameter	MTB Byte(s)	FTB Byte
t_{CKAVG}min	18 (0x012)	125 (0x07D)
t_{CKAVG}max	19 (0x013)	124 (0x07C)
t_{AA}min	24 (0x018)	123 (0x07B)
t_{RCD}min	25 (0x019)	122 (0x07A)
t_{RPab}min	26 (0x01A)	121 (0x079)
t_{RPpb}min	27 (0x01B)	120 (0x078)

The encoding of two's complement fine timebase offsets shown in Table 23:

Table 23 — Encoding of Two's Complement Fine Timebase Offsets

Coding		Value (Dec)	Value (Hex)	FTB Timebase
Bit 7	Bits 6~0			1 ps
0	1111111	+127	7F	+127 ps
0	1111110	+126	7E	+126 ps
...	...	...	...	
0	0000001	+1	01	+1 ps
0	0000000	0	00	0
1	1111111	-1	FF	-1 ps
1	1111110	-2	FE	-2 ps
...	...	...	...	
1	0000000	-128	80	-128 ps

3.1.18.2 Rounding Algorithms

Software algorithms for calculation of timing parameters are subject to rounding errors from many sources. For example, a system may use a memory clock with a nominal frequency of 933.33... MHz, or a clock period of 1.0714... ns. Similarly, a system with a memory clock frequency of 1066.66... MHz yields mathematically a clock period of 0.9375... ns. In most cases, it is impossible to express all digits after the decimal point exactly, and rounding must be done because the SPD establishes a minimum granularity for timing parameters of 1 ps.

Rules for rounding must be defined to allow optimization of memory module performance without violating device parameters. In some cases, rounding errors cause an unnecessary clock of delay, affecting system performance. These rounding algorithms rely on results that are within correction factors on device testing and specification to avoid losing performance due to rounding errors.

These rules are:

1. Clock periods such as t_{CKAVG}min are defined to 1 ps of accuracy; for example, 0.9375... ns is defined as 937 ps and 1.0714... ns is defined as 1071 ps.

3.1.18.2 Rounding Algorithms (Cont'd)

2. Using real math, parameters like t_{AA}min, t_{RCD}min, etc. which are programmed in systems in numbers of clocks (nCK) but expressed in the SPD in units of time (in ns) are divided by the clock period (in ns) yielding a ratio of clock units (nCK), a correction factor of 2.5% is subtracted, then the result is set to the next higher integer number of clocks:

$$\text{nCK} = \text{ceiling}\left[\left(\frac{\text{parameter\_in\_ns}}{\text{application\_}t_{CK}\text{\_in\_ns}}\right) - 0.025\right]$$

3. Alternatively, programmers may prefer to use integer math instead of real math by expressing timing in ps, scaling the desired parameter value by 1000, dividing by the application clock period, adding an inverse correction factor of 97.4%, dividing the result by 1000, then truncating down to the next lower integer value:

$$\text{nCK} = \text{truncate}\left[\frac{\left(\frac{\text{parameter\_in\_ps x 1000}}{\text{application\_}t_{CK}\text{\_in\_ps}}\right) + 974}{1000}\right]$$

4. Either algorithm should yield identical results. In case of any conflict between the two methodologies, the integer method shall prevail.

Examples

Example 1, using REAL math to convert t_{RCD}min from ns to nCK (see Table 24):

```
// This algorithm subtracts 2.5% correction factor and rounds up to next integer value

real MTB, FTB, TrcdMin, Correction, ClockPeriod, TempNck;
int TrcdInNck;

MTB = 0.125;                                    // From SPD[17], in ns
FTB = 0.001;                                    // From SPD[17], in ns
TrcdMin = (SPD[26] * MTB) + (SPD[122] * FTB);   // Calculate tRCDmin in ns (FTB is negative offset)
Correction = 0.025;                             // 2.5%, per rounding algorithm
ClockPeriod = ApplicationTck;                   // Frequency (clock period) is application dependent
TempNck = TrcdMin / ClockPeriod;                // Initial calculation of nCK
TempNck = TempNck - Correction;                 // Subtract correction factor from nCK
TrcdInNck = (int)ceiling(TempNck);              // Ceiling to next higher integer value
```

Table 24 — Example 1, using REAL math to convert t_{RCD}min from ns to nCK

LPDDR4 Devices Operating at Various Frequencies (Standard and Downbin) Timing Parameter: t_{RCD}min = 18.0 ns at all frequencies						
Device Speed Grade	Device t_{RCD}	Application t_{CK}	Device t_{RCD} ÷ Application t_{CK}	2.5% Correction	(t_{RCD} ÷ t_{CK}) — Correction	Ceiling Result
	ns	ns	ratio (real)	(real)	ratio (real)	nCK (integer)
1600	18.0	1.25	14.400	0.025	14.375	15
1600	18.0	1.500	12.000	0.025	11.975	12
1866	18.0	1.071	16.807	0.025	16.782	17
2133	18.0	0.937	19.210	0.025	19.185	20

Example 2, using INT math to convert t_{RCD}min from ns to nCK (see Table 25):

```
// This algorithm uses adds 97.4% of a clock and truncates down to the next lower integer value

int MTB, FTB, TrcdMin, ClockPeriod, TempNck, TrcdInNck;

MTB = 125;                                      // From SPD[17], in ps
FTB = 1;                                        // From SPD[17], in ps
TrcdMin = (SPD[26] * MTB) + (SPD[122] * FTB);   // Calculate tRCDmin in ps (FTB is negative offset)
ClockPeriod = ApplicationTckInPs;               // Clock period is application specific
TempNck = (TrcdMin * 1000) / ApplicationTckInPs;// Preliminary nCK calculation, scaled by 1000
TempNck = TempNck + 974;                        // Apply inverse of 2.5% correction factor
TrcdInNck = (int)(TempNck / 1000);              // Truncate to next lower integer
```

Table 25 — Example 2, using INT math to convert t_{RCD}min from ns to nCK

LPDDR4 Devices Operating at Various Frequencies (Standard and Downbin) Timing Parameter: t_{RCD}min = 18.0 ns at all frequencies (18000 ps)					
Device Speed Grade	Device t_{RCD}	Application t_{CK}	(Device t_{RCD} * 1000) ÷ Application t_{CK}	Add Inverse Correction	Truncate Corrected nCK / 1000
	ps	ps	Scaled nCK	Scaled nCK	nCK (integer)
1600	18000	1250	14400	15374	15
1600	18000	1500	12000	12974	12
1866	18000	1071	16806	17780	17
2133	18000	937	19210	20184	20

3.1.19 Byte 18 (0x012): SDRAM Minimum Cycle Time (t_{CKAVG}min)

This byte, shown in Table 26, defines the minimum cycle time for the SDRAM module, in medium timebase (MTB) units. This number applies to all applicable components on the module. This byte applies to SDRAM and support components as well as the overall capability of the DIMM. This value comes from the LPDDRLPDDR SDRAM and support component data sheets.

Table 26 — Byte 18 (0x012): SDRAM Minimum Cycle Time (t_{CKAVG}min)

Bits 7~0
Minimum SDRAM Cycle Time (t_{CKAVG}min) MTB Units
Values defined from 1 to 255

If t_{CKAVG}min cannot be divided evenly by the MTB, this byte must be rounded up to the next larger integer and the Fine Offset for t_{CKAVG}min (SPD byte 125) used for correction to get the actual value.

Examples (shown in Table 27):

Table 27 — Examples of SDRAM Minimum Cycle Time

t_{CKAVG}min (MTB units)		MTB (ns)	t_{CKAVG}min Offset (FTB units)[1]		FTB (ns)	t_{CKAVG}min Result (ns)	Use
10	0x0A	0.125	0	0	0.001	1.25	LPDDR3-1600 (800 MHz clock)
9	0x09	0.125	-54	0xCA	0.001	1.071	LPDDR3-1866 (933 MHz clock)
8	0x08	0.125	-63	0xC1	0.001	0.938	LPDDR3-2133 (1067 MHz clock)
5	0x05	0.125	0	0	0.001	0.625	LPDDR4-3200 (1600 MHz clock)

NOTE 1 See SPD byte 125.

3.1.20 Byte 19 (0x013): SDRAM Maximum Cycle Time (t_{CKAVG}max)

This byte, shown in Table 28,defines the maximum cycle time for the SDRAM module, in medium timebase (MTB) units. This number applies to all applicable components on the module. This byte applies to SDRAM and support components as well as the overall capability of the DIMM. This value comes from the LPDDRLPDDR SDRAM and support component data sheets.

Table 28 — Byte 19 (0x013): SDRAM Maximum Cycle Time (t_{CKAVG}max)

Bits 7~0
Minimum SDRAM Cycle Time (t_{CKAVG}max) MTB Units
Values defined from 1 to 255

3.1.20 Byte 19 (0x013): SDRAM Maximum Cycle Time (t_{CKAVG}max) (Cont'd)

If t_{CKAVG}max cannot be divided evenly by the MTB, this byte must be rounded up to the next larger integer and the Fine Offset for t_{CKAVG}max (SPD byte 124) used for correction to get the actual value.

Examples (shown in Table 29):

Table 29 — Examples of Maximum Cycle Time

t_{CKAVG}max (MTB units)		MTB (ns)	t_{CKAVG}max Offset (FTB units)[1]		FTB (ns)	t_{CKAVG}max Result (ns)	Use
12	0x0C	0.125	0	0	0.001	1.500	LPDDR3-1600 (800 MHz clock)
12	0x0C	0.125	0	0	0.001	1.500	LPDDR3-1866 (933 MHz clock)
12	0x0C	0.125	0	0	0.001	1.500	LPDDR3-2133 (1067 MHz clock)
		0.125			0.001	tbd	LPDDR4-3200 (1600 MHz clock)

NOTE 1 See SPD byte 124.

3.1.21 Byte 20 (0x014): CAS Latencies Supported, First Byte
Byte 21 (0x015): CAS Latencies Supported, Second Byte
Byte 22 (0x016): CAS Latencies Supported, Third Byte
Byte 23 (0x017): CAS Latencies Supported, Fourth Byte

These bytes, shown in Table 30, define which CAS Latency (CL) values are supported. The range is from CL = 3 through CL = 44 with one bit per possible CAS Latency. A 1 in a bit position means that CL is supported, a 0 in that bit position means it is not supported. These values come from the LPDDR SDRAM data sheets, JESD209-x.

Table 30 — Bytes 20~23: CAS Latencies Supported

Byte 20: CAS Latencies Supported, First Byte							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
CL = 14	**CL = 12**	**CL = 11**	**CL = 10**	**CL = 9**	**CL = 8**	**CL = 6**	**CL = 3**
0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1
Byte 21: CAS Latencies Supported, Second Byte							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
CL = 30	**CL = 28**	**CL = 26**	**CL = 24**	**CL = 22**	**CL = 20**	**CL = 18**	**CL = 16**
0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1
Byte 22: CAS Latencies Supported, Third Byte							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reserved	**CL = 44**	**Reserved**	**CL = 40**	**Reserved**	**CL = 36**	**Reserved**	**CL = 32**
0	0 or 1	0	0 or 1	0	0 or 1	0	0 or 1
Byte 23: CAS Latencies Supported, Fourth Byte							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reserved	**Reserved**	**Reserved**	**Reserved**	**Reserved**	**Reserved**	**Reserved**	**Reserved**
0	0	0	0	0	0	0	0

NOTE 1 Byte 23 bit 7 - Currently reserved for selection of CAS Latency range
NOTE 2 Byte 23 bit 6 - Currently reserved for ability to identify the DRAM Technology if a different table is required.
NOTE 3 For each bit position, 0 means this CAS Latency is not supported, 1 means this CAS Latency is supported.

Example (shown in Table 31):

Byte 20 = 0xB4 (= 1011 0100) – first byte

Byte 21 = 0x05 (= 0000 0101) – second byte
Byte 22 = 0x00 (= 0000 0000) – third byte
Byte 23 = 0x40 (= 0100 0000) – fourth byte

Table 31 — Example of CAS Latencies

CAS Latencies	30	28	26	24	22	20	18	16	14	12	11	10	9	8	6	3
CL Mask	0	0	0	0	0	1	0	1	1	0	1	1	0	1	0	0
CAS Latencies	R	R	R	R	R	R	R	R	R	44	R	40	R	36	R	32
CL Mask	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Results: Actual CAS Latencies supported for LPDDRx = 8, 10, 11, 14, 16, 20,

3.1.22 Byte 24 (0x018): Minimum CAS Latency Time (t_{AA}min)

This word, shown in Table 32, defines the minimum CAS Latency in medium timebase (MTB) units. This value comes from the LPDDR SDRAM data sheet.

Table 32 — Byte 24 (0x018): Minimum CAS Latency Time (t_{AA}min)

Bits 7~0
Minimum SDRAM CAS Latency Time (t_{AA}min) MTB Units
Values defined from 1 to 255

If t_{AA}min cannot be divided evenly by the MTB, this byte must be rounded up to the next larger integer and the Fine Offset for t_{AA}min (SPD byte 123) used for correction to get the actual value.

Examples (shown in Table 33):

Table 33 — Examples of Minimum CAS Latency Time

t_{AA}min (MTB units)		MTB (ns)	t_{AA}min Offset (FTB units)[1]		FTB (ns)	t_{AA}min Result (ns)	Use
100	0x64	0.125	0	0	0.001	12.50	LPDDR4-1600 (800 MHz clock)
103	0x67	0.125	-26	0xE6	0.001	12.85	LPDDR3-1866 (933 MHz clock)
106	0x6A	0.125	-120	0x88	0.001	13.13	LPDDR3-2133 (1066 MHz clock)
		0.125			0.001	tbd	LPDDR4-3200 (1600 MHz clock)

NOTE 1 See SPD byte 123
NOTE 2 Refer to device data sheet for downbin support details.

3.1.23 Byte 25 (0x019): Read & Write Latency Set options

This byte, shown in Table 34, defines the support options for both Read Latency with DBI-Disabled or Enabled and Write Latency Set A and Set B.

Table 34 — Byte 25 (0x019): Read & Write Latency Set options

Bits 7~4	Bits 3~2	Bits 1~0
Reserved	Write Latency Set	Read Latency Mode
Reserved; must be coded as 0000	Bits [3, 2]: 00 = Write Latency Set A 01 = Write Latency Set B All others reserved	Bits [0, 1]: 00 = RL & nRTP for DBI-RD Disabled 01 = RL & nRTP for DBI-RD Enabled All others reserved

3.1.24 Byte 26 (0x01A): Minimum RAS to CAS Delay Time (t_{RCD}min)

This word, shown in Table 35, defines the minimum SDRAM RAS to CAS Delay Time in fine timebase (FTB) units. This value comes from the LPDDR SDRAM data sheet.

Table 35 — Byte 26 (0x01A): Minimum RAS to CAS Delay Time (t_{RCD}min)

Bits 7~0
Byte 25: Minimum SDRAM RAS to CAS Delay Time (t_{RCD}min) FTB Units
Values defined from 1 to 255

If t_{RCD}min cannot be divided evenly by the MTB, this byte must be rounded up to the next larger integer and the Fine Offset for t_{RCD}min (SPD byte 122) used for correction to get the actual value.

Examples (shown in Table 36):

Table 36 — Examples of Minimum RAS to CAS Delay Time

t_{RCD}min (MTB units)		MTB (ns)	t_{RCD}min Offset (FTB units)[1]		FTB (ns)	t_{RCD}min Result (ns)	Use
120	0x78	0.125	0	0	0.001	15ns	LPDDR3-1600 (Fast)
144	0x90	0.125	0	0	0.001	18ns	LPDDR3-1600 (Typ)
192	0xC0	0.125	0	0	0.001	24ns	LPDDR3-2133 (Slow)
144	0x90	0.125	0	0	0.001	18ns	LPDDR4-3200 (1600 MHz clock)

NOTE 1 See SPD byte 122
NOTE 2 Refer to device data sheet for downbin support details.

3.1.25 Byte 27 (0x01B): All Banks Minimum Row Precharge Delay Time (t_{RPab}min)

This word, shown in Table 37, defines the minimum All Banks SDRAM Row Precharge Delay Time in medium timebase (MTB) units. This value comes from the LPDDR SDRAM data sheet.

Table 37 — Byte 27 (0x01B): All Banks Minimum Row Precharge Delay Time (t_{RPab}min)

Bits 7~0
Minimum Row Precharge Time (t_{RPab}min) MTB Units
Values defined from 1 to 255

If t_{RPab}min cannot be divided evenly by the MTB, this byte must be rounded up to the next larger integer and the Fine Offset for t_{RPab}min (SPD byte 120) used for correction to get the actual value.

Examples (shown in Table 38):

Table 38 — Minimum Row Precharge Delay Time

t_{RPabp}min (MTB units)		MTB (ns)	t_{RPab}min Offset (FTB units)[1]		FTB (ns)	t_{RPab}min Result (ns)	Use
144	0x90	0.125	0	0	0.001	18ns	LPDDR3-1600 (Fast)
168	0xA8	0.125	0	0	0.001	21ns	LPDDR3-1600 (Typ)
216	0xD8	0.125	0	0	0.001	27ns	LPDDR3-2133 (Slow)
168	0xA8	0.125	0	0	0.001	21ns	LPDDR4-3200 (1600 MHz clock)

NOTE 1 See SPD byte 121
NOTE 2 Device supports downbinning in lower frequency applications; see supplier data sheet

3.1.26 Byte 28 (0x01C): Per Bank Minimum Row Precharge Delay Time (t_{RPpb}min)

This word, shown in Table 39, defines the minimum Per Bank SDRAM Row Precharge Delay Time in medium timebase (MTB) units. This value comes from the LPDDR SDRAM data sheet.

Table 39 — Byte 28 (0x01C): Per Bank Minimum Row Precharge Delay Time (t_{RPpb}min)

Bits 7~0
Minimum Row Precharge Time (t_{RPpb}min) MTB Units
Values defined from 1 to 255

If t_{RPpb}min cannot be divided evenly by the MTB, this byte must be rounded up to the next larger integer and the Fine Offset for t_{RPpb}min (SPD byte 121) used for correction to get the actual value.

Examples (shown in Table 40):

Table 40 — Examples of Per Bank Minimum Row Precharge Delay Time

t_{RPpb}min (MTB units)		MTB (ns)	t_{RPpb}min Offset (FTB units)[1]		FTB (ns)	t_{RPpb}min Result (ns)	Use
120	0x78	0.125	0	0	0.001	15ns	LPDDR3-1600 (Fast)
144	0x90	0.125	0	0	0.001	18ns	LPDDR3-1600 (Typ)
192	0xC0	0.125	0	0	0.001	24ns	LPDDR3-2133 (Slow)
144	0x90	0.125	0	0	0.001	18ns	LPDDR4-3200 (1600 MHz clock)

NOTE 1 See SPD byte 121
NOTE 2 Device supports downbinning in lower frequency applications; see supplier data sheet

3.1.27 Byte 29 (0x01D): All Banks Minimum Refresh Recovery Delay Time (t_{RFCab}min), LSB
Byte 30 (0x01E): All Banks Minimum Refresh Recovery Delay Time (t_{RFCab}min), MSB

This word, shown in Table 41, defines the minimum All Banks SDRAM Refresh Recovery Time Delay in medium timebase (MTB) units. These values come from the LPDDR3 or LPDDR4 SDRAM data sheet.

Table 41 — Bytes 29 and 30: All Banks Minimum Refresh Recovery Delay Time (t_{RFCab}min)

Minimum SDRAM Refresh Recovery Delay Time (t_{RFCab}min) MTB Units	
Byte 31	Byte 30
Bits 15~8	Bits 7~0
Values defined from 1 to 65535	

Examples (shown in Table 42):

Table 42 — Examples of All Banks Minimum Refresh Recovery Delay Time

t_{RFCab}min (MTB units)		MTB (ns)	t_{RFCab}min Result (ns)	Use
1040	0x0410	0.125	130ns	4 Gb LPDDR3 SDRAM
1680	0x0690	0.125	210ns	6 Gb LPDDR3 SDRAM
1680	0x0690	0.125	210ns	8 Gb LPDDR3 SDRAM
1040	0x0410	0.125	130ns	4 Gb LPDDR4 SDRAM
1440	0x05A0	0.125	180ns	6 Gb LPDDR4 SDRAM
1440	0x05A0	0.125	180ns	8 Gb LPDDR4 SDRAM

Table 42 — Examples of All Banks Minimum Refresh Recovery Delay Time

t_{RFCab}min (MTB units)		MTB (ns)	t_{RFCab}min Result (ns)	Use
1680	0x0690	0.125	210ns	12 Gb LPDDR4 SDRAM
1680	0x0690	0.125	210ns	16 Gb LPDDR4 SDRAM

3.1.28 Byte 31 (0x01F): Per Bank Minimum Refresh Recovery Delay Time (t_{RFCpb}min), LSB Byte 32 (0x020): Per Bank Minimum Refresh Recovery Delay Time ($t_{RFC[b}$min), MSB

This word, shown in Table 43, defines the minimum Per Bank SDRAM Refresh Recovery Time Delay in medium timebase (MTB) units. These values come from the LPDDR3 or LPDDR4 SDRAM data sheet.

Table 43 — Bytes 31 and 32: Per Bank Minimum Refresh Recovery Delay Time (t_{RFCpb}min)

Per Bank Minimum SDRAM Refresh Recovery Delay Time (t_{RFCpb}min) MTB Units	
Byte 31	Byte 30
Bits 15~8	Bits 7~0
Values defined from 1 to 65535	

Examples (shown in Table 44):

Table 44 — Examples of Per Bank Minimum Refresh Recovery Delay Time

t_{RFCpb}min (MTB units)		MTB (ns)	t_{RFCpb}min Result (ns)	Use
480	0x01E0	0.125	60ns	4 Gb LPDDR3 SDRAM
720	0x02D0	0.125	90ns	6 Gb LPDDR3 SDRAM
720	0x02D0	0.125	90ns	8 Gb LPDDR3 SDRAM
		0.125		
480	0x01E0	0.125	60ns	4 Gb LPDDR4 SDRAM
720	0x02D0	0.125	90ns	6 Gb LPDDR4 SDRAM
720	0x02D0	0.125	90ns	8 Gb LPDDR4 SDRAM
720	0x02D0	0.125	90ns	12 Gb LPDDR4 SDRAM
720	0x02D0	0.125	90ns	16 Gb LPDDR4 SDRAM

3.1.29 Byte 33~59 (0x021~0x03B): Reserved, Base Configuration Section

Must be coded as 0x00

3.1.30 Bytes 60~77 (0x03C~0x04D): Connector to SDRAM Bit Mapping

These bytes, shown in Table 45, document the connection between data signals at the edge connector of a module to the LPDDR3 or LPDDR4 SDRAM inputs pins for package rank 0 of the module. This information is used by the controller to route data onto the correct bit lines for CRC transmission as described in the LPDDR3 or LPDDR4 SDRAM data sheet JESD79-4. Each byte describes the mapping for one nibble (four bits) of data. In addition, each SPD byte describes the mapping between package rank 0 bits and equivalent bits in other ranks.

Table 45 — Bytes 60~77 (0x03C~0x04D): Connector to SDRAM Bit Mapping

SPD Byte		Connector Bits	SPD Byte		Connector Bits	SPD Byte		Connector Bits
60	0x03C	DQ0-3	66	0x042	DQ24-27	72	0x048	DQ40-43
61	0x03D	DQ4-7	67	0x043	DQ28-31	73	0x049	DQ44-47

Table 45 — Bytes 60~77 (0x03C~0x04D): Connector to SDRAM Bit Mapping

SPD Byte		Connector Bits	SPD Byte		Connector Bits	SPD Byte		Connector Bits
62	0x03E	DQ8-11	68	0x044	CB0-3	74	0x04A	DQ48-51
63	0x03F	DQ12-15	69	0x045	CB4-7	75	0x04B	DQ52-55
64	0x040	DQ16-19	70	0x046	DQ32-35	76	0x04C	DQ56-59
65	0x041	DQ20-23	71	0x047	DQ36-39	77	0x04D	DQ60-63

The mapping rules are as follows:

- All bits within a nibble at the edge connector must be wired to the same SDRAM.
- All bits within a byte at the edge connector must be wired to the same SDRAM for x8 and wider SDRAMs.
- Bits within a nibble may be swapped in any order.
- Nibbles may be swapped within a byte.
- Bytes may be wired in any order within the SDRAM width for x16 and wider SDRAMs.

Each SPD byte in the Bit Mapping array is encoded as follows (see Table 46). If the nibble at the edge connector is wired to the lower nibble of a byte at the SDRAM (x8 and wider), then bit 5 is coded as 0. If the nibble at the edge connector is wired to the upper nibble of a byte at the SDRAM, then bit 5 is coded as 1. Bit 5 = 0 for all x4 based modules. Bits 6~7 define the connectivity between bits in different package ranks.

Table 46 — Bit Mapping Array

Bits 7 ~ 6	Bit 5	Bits 4 ~ 0
Package Rank Map	**Wired to Upper/Lower Nibble**	**Bit Order at SDRAM**
See Package Rank Map table	0 = lower nibble at SDRAM 1 = upper nibble at SDRAM	See Nibble Map table

Package Rank Map: Bits 7~6 in each SPD byte define the mapping between bits in Package Rank 0 and other package ranks on the module. The mapping rules are defined in Table 47:

Table 47 — Mapping Rules

Package Rank Map				
Bits 7 ~ 6	**Bit Order at SDRAM**			
00	Even package ranks (0, 2, etc.) have the same mapping Odd package ranks (1, 3, etc) map SDRAM data bits relative to Package Rank 0 as follows:			
	DQ0 → DQ1	DQ8 → tbd	DQ16 → tbd	DQ24 → tbd
	DQ1 → DQ0	DQ9 → tbd	DQ17 → tbd	DQ25 → tbd
	DQ2 → DQ3	DQ10 → tbd	DQ18 → tbd	DQ26 → tbd
	DQ3 → DQ2	DQ11 → tbd	DQ19 → tbd	DQ27 → tbd
	DQ4 → DQ5	DQ12 → tbd	DQ20 → tbd	DQ28 → tbd
	DQ5 → DQ4	DQ13 → tbd	DQ21 → tbd	DQ29 → tbd
	DQ6 → DQ7	DQ14 → tbd	DQ22 → tbd	DQ30 → tbd
	DQ7 → DQ6	DQ15 → tbd	DQ23 → tbd	DQ31 → tbd
01 10 11	Reserved			

3.1.30 Bytes 60~77 (0x03C~0x04D): Connector to SDRAM Bit Mapping (Cont'd)

The Nibble Map is coded as shown in Table 48:

Table 48 — Nibble Map Encoding

Nibble Map					
Nibble Bit Order at Connector	**Bit Map Index Bits 4 ~ 0**	**Bit 5 = 0: Wired to Lower Nibble Bit Order within SDRAM Byte**		**Bit 5 = 1: Wired to Upper Nibble Bit Order within SDRAM Byte**	
	00000	0x00	Bit Map not specified	0x20	Bit Map not specified
0, 1, 2, 3 (4, 5, 6, 7) ...	00001	0x01	0, 1, 2, 3	0x21	4, 5, 6, 7
	00010	0x02	0, 1, 3, 2	0x22	4, 5, 7, 6
	00011	0x03	0, 2, 1, 3	0x23	4, 6, 5, 7
	00100	0x04	0, 2, 3, 1	0x24	4, 6, 7, 5
	00101	0x05	0, 3, 1, 2	0x25	4, 7, 5, 6
	00110	0x06	0, 3, 2, 1	0x26	4, 7, 6, 5
	00111	0x07	1, 0, 2, 3	0x27	5, 4, 6, 7
0, 1, 2, 3 (4, 5, 6, 7) ...	01000	0x08	1, 0, 3, 2	0x28	5, 4, 7, 6
	01001	0x09	1, 2, 0, 3	0x29	5, 6, 4, 7
	01010	0x0A	1, 2, 3, 0	0x2A	5, 6, 7, 4
	01011	0x0B	1, 3, 0, 2	0x2B	5, 7, 4, 6
	01100	0x0C	1, 3, 2, 0	0x2C	5, 7, 6, 4
	01101	0x0D	2, 0, 1, 3	0x2D	6, 4, 5, 7
	01110	0x0E	2, 0, 3, 1	0x2E	6, 4, 7, 5
	01111	0x0F	2, 1, 0, 3	0x2F	6, 5, 4, 7
0, 1, 2, 3 (4, 5, 6, 7) ...	10000	0x10	2, 1, 3, 0	0x30	6, 5, 7, 4
	10001	0x11	2, 3, 0, 1	0x31	6, 7, 4, 5
	10010	0x12	2, 3, 1, 0	0x32	6, 7, 5, 4
	10011	0x13	3, 0, 1, 2	0x33	7, 4, 5, 6
	10100	0x14	3, 0, 2, 1	0x34	7, 4, 6, 5
	10101	0x15	3, 1, 0, 2	0x35	7, 5, 4, 6
	10110	0x16	3, 1, 2, 0	0x36	7, 5, 6, 4
	10111	0x17	3, 2, 0, 1	0x37	7, 6, 4, 5
	11000	0x18	3, 2, 1, 0	0x38	7, 6, 5, 4
	All other codes		Reserved		Reserved

NOTE Hex codes shown in this table are for bits 5~0 only and must be added to bits 7~6 (Package Rank Map bits) for the SPD byte entry

Example (shown in Figure 3 and Table 49):

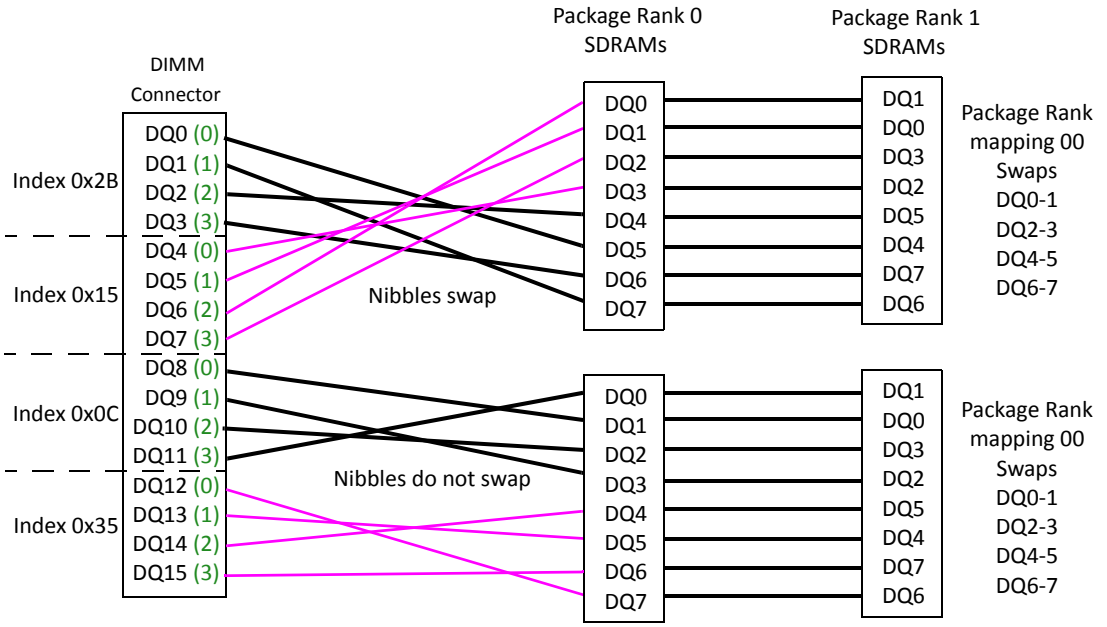


Figure 3 — Two Package Rank x8 Module (example only; may not represent a specific design)

Table 49 — Two Package Rank x8 Module (example only; may not represent a specific design)

DQ bit at DIMM Connector																			
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
First SDRAM																			
5	7	4	6	Nibble wired to upper nibble of SDRAM byte (bit 5 = 1) using Nibble Map 01011 (bits 4~0)															
4	6	5	7	Code 0x2B stored in the SPD table for the first nibble															
				3	1	0	2	Nibble wired to lower nibble of SDRAM byte (bit 5 = 0) using Nibble										
				2	0	1	3	Map 1010 (bits 4~0). Code 0x15 stored for the second nibble										
								Second SDRAM											
								1	3	2	0	Code 0x0C for the 3rd nibble							
								0	2	3	1								
												7	5	4	6	Code 0x35 for the fourth nibble			
												6	4	5	7				

Package Rank Map 00
Even Package Ranks
Odd Package Ranks

3.1.31 Bytes 78~119 (0x04E~0x077): Reserved, Base Configuration Section

Must be coded as 0x00

3.1.32 Byte 120 (0x078): Fine Offset for Minimum Per Bank Row Precharge Delay Time (t_{RPpb}min)

This byte modifies the calculation of SPD Byte 28 (MTB units) with a fine correction using FTB units. The value of t_{RPpb}min comes from the SDRAM data sheet. This value is a two's complement multiplier for FTB units, ranging from +127 to -128.

Examples: See SPD byte 28, in Section 3.1.26. For Two's Complement encoding, see Section 3.1.18.1, Relating the MTB and FTB.

3.1.33 Byte 121 (0x079): Fine Offset for Minimum All Banks Row Precharge Delay Time (t_{RPab}min)

This byte modifies the calculation of SPD Byte 27 (MTB units) with a fine correction using FTB units. The value of t_{RP}min comes from the SDRAM data sheet. This value is a two's complement multiplier for FTB units, ranging from +127 to -128.

Examples: See SPD byte 27, in Section 3.1.25. For Two's Complement encoding, see Section 3.1.18.1, Relating the MTB and FTB.

3.1.34 Byte 122 (0x07A): Fine Offset for Minimum RAS to CAS Delay Time (t_{RCD}min)

This byte modifies the calculation of SPD Byte 26 (MTB units) with a fine correction using FTB units. The value of t_{RCD}min comes from the SDRAM data sheet. This value is a two's complement multiplier for FTB units, ranging from +127 to -128.

Examples: See SPD byte 26, in Section 3.1.24. For Two's Complement encoding, see Section 3.1.18.1, Relating the MTB and FTB.

3.1.35 Byte 123 (0x07B): Fine Offset for Minimum CAS Latency Time (t_{AA}min)

This byte modifies the calculation of SPD Byte 24 (MTB units) with a fine correction using FTB units. The value of t_{AA}min comes from the SDRAM data sheet. This value is a two's complement multiplier for FTB units, ranging from +127 to -128.

Examples: See SPD Byte 24, in Section 3.1.22. For Two's Complement encoding, see Section 3.1.18.1, Relating the MTB and FTB.

3.1.36 Byte 124 (0x07C): Fine Offset for SDRAM Maximum Cycle Time (t_{CKAVG}max)

This byte modifies the calculation of SPD Byte 19 (MTB units) with a fine correction using FTB units. The value of t_{CKAVG}max comes from the SDRAM data sheet. This value is a two's complement multiplier for FTB units, ranging from +127 to -128.

Examples: See SPD byte 19, in Section 3.1.22. For Two's Complement encoding, see Section 3.1.18.1, Relating the MTB and FTB.

3.1.37 Byte 125 (0x07D): Fine Offset for SDRAM Minimum Cycle Time (t_{CKAVG}min)

This byte modifies the calculation of SPD Byte 18 (MTB units) with a fine correction using FTB units. The value of t_{CKAVG}min comes from the SDRAM data sheet. This value is a two's complement multiplier for FTB units, ranging from +127 to -128.

Examples: See SPD byte 18, in Section 3.1.19. For Two's Complement encoding, see Section 3.1.18.1, Relating the MTB and FTB.

3.1.38 Byte 126 (0x07E): Cyclical Redundancy Code (CRC) for Base Configuration Section, LSB
Byte 127 (0x07F): Cyclical Redundancy Code (CRC) for Base Configuration Section, MSB

This two-byte field contains the calculated CRC for bytes 0~125 (0x000~0x07D) in the SPD. The following algorithm and data structures (shown in C) are to be followed in calculating and checking the code.

```
int Crc16 (char *ptr, int count)
{
   int crc, i;

   crc = 0;
   while (--count >= 0) {
      crc = crc ^ (int)*ptr++ << 8;
      for (i = 0; i < 8; ++i)
         if (crc & 0x8000)
            crc = crc << 1 ^ 0x1021;
         else
            crc = crc << 1;
   }
   return (crc & 0xFFFF);
}

char spdBytes[] = { SPD_byte_0, SPD_byte_1, ..., SPD_byte_N-1 };
int data16;

data16 = Crc16 (spdBytes, sizeof(spdBytes));
SPD_byte_126 = (char) (data16 & 0xFF);
SPD_byte_127 = (char) (data16 >> 8);
```

3.2 Module-Specific Section: Bytes 128~255 (0x080~0x0FF)

This section contains SPD bytes which are specific to families LPDDR3 or LPDDR4 module families. Module Type Key Byte 3 is used as an index for the encoding of bytes 128~255. The content of bytes 128~255 are described in multiple annexes, one for each memory module family.

3.3 Hybrid Memory Architecture Specific Parameters: Bytes 256~319 (0x100~0x13F)

This section contains SPD bytes which are specific to hybrid memory module families. Module Type Key Byte 3 bits 7~4 are used as an index for the encoding of bytes 256~319. The content of bytes 256~319 are described in multiple annexes, one for each hybrid module family.

3.4 Module Supplier's Data: Bytes 320~383 (0x140~0x17F)

3.4.1 Byte 320 (0x140): Module Manufacturer ID Code, LSB
Byte 321 (0x141): Module Manufacturer ID Code, MSB

This two-byte field, shown in Table 50, indicates the manufacturer of the module, encoded as follows: the first byte is the number of continuation bytes indicated in JEP-106; the second byte is the last non-zero byte of the manufacturer's ID code, again as indicated in JEP-106.

Table 50 — Bytes 320 and 321: Module Manufacturer ID Code

Byte 321, Bits 7~0	Byte 320, Bit 7	Byte 320, Bits 6~0
Last non-zero byte, Module Manufacturer	**Odd Parity for Byte 320, bits 6~0**	**Number of continuation codes, Module Manufacturer**
See JEP-106		See JEP-106

Examples (shown in Table 51):

Table 51 — Examples of Module Manufacturer ID Code

Company	JEP-106		# continuation codes	SPD	
	Bank	Code		Byte 320	Byte 321
Fujitsu	1	04	0	0x80	0x04
US Modular	5	A8	4	0x04	0xA8

3.4.2 Byte 322 (0x142): Module Manufacturing Location

The module manufacturer includes an identifier that uniquely defines the manufacturing location of the memory module. While the SPD specification will not attempt to present a decode table for manufacturing sites, the individual manufacturer may keep track of manufacturing location and its appropriate decode represented in this byte.

3.4.3 Bytes 323~324 (0x143~0x144): Module Manufacturing Date

The module manufacturer includes a date code for the module. The JEDEC definitions for bytes 323 and 324 are year and week respectively. These bytes must be represented in Binary Coded Decimal (BCD). For example, week 47 in year 2014 would be coded as 0x14 (0001 0100) in byte 323 and 0x47 (0100 0111) in byte 324.

3.4.4 Bytes 325~328 (0x145~0x148): Module Serial Number

The supplier must include a unique serial number for the module. The supplier may use whatever decode method desired to maintain a unique serial number for each module.

One method of achieving this is by assigning a byte in the field from 325~328 as a tester ID byte and using the remaining bytes as a sequential serial number. Bytes 320~328 will then result in a nine-byte unique module identifier. Note that part number is not included in this identifier: the supplier may not give the same value for Bytes 320~328 to more than one DIMM even if the DIMMs have different part numbers.

3.4.5 Bytes 329~348 (0x149~15C): Module Part Number

The manufacturer's part number is written in ASCII format within these bytes. Unused digits are coded as ASCII blanks (0x20).

3.4.6 Bytes 349 (0x15D): Module Revision Code

This refers to the module revision code. While the SPD specification will not attempt to define the format for this information, the individual manufacturer may keep track of the revision code and its appropriate decode represented in this byte. This revision code refers to the manufacturer's assembly revision level and may be different than the raw card revision in SPD bytes 128 and 130.

3.4.7 Byte 350 (0x15E): DRAM Manufacturer ID Code, LSB
Byte 351 (0x15F): DRAM Manufacturer ID Code, MSB

This two-byte field, shown in Table 52, indicates the manufacturer of the DRAM on the module, encoded as follows: the first byte is the number of continuation bytes indicated in JEP-106; the second byte is the last non-zero byte of the manufacturer's ID code, again as indicated in JEP-106.

Table 52 — Bytes 350 and 351: DRAM Manufacturer ID Code

Byte 351, Bits 7~0	Byte 350, Bit 7	Byte 350, Bits 6~0
Last non-zero byte, DRAM Manufacturer	**Odd Parity for Byte 350, bits 6~0**	**Number of continuation codes, DRAM Manufacturer**
See JEP-106		See JEP-106

Example: See bytes 320~321, in Section 3.4.1, for example manufacturer codes.

3.4.8 Byte 352 (0x160): DRAM Stepping

This byte, shown in Table 53, defines the vendor die revision level (often called the "stepping") of the DRAMs on the module. This byte is optional. For modules without DRAM stepping information, this byte should be programmed to 0xFF.

Table 53 — Byte 352 (0x160): DRAM Stepping

Bits 7~0
DRAM Stepping
Programmed in straight Hex format - no conversion needed. 00 - Valid 01 - Valid .. FE - Valid FF - Undefined (No Stepping Number Provided)

Examples (shown in Table 54):

Table 54 — Examples of DRAM Stepping

Code	Meaning
0x00	Stepping 0
0x01	Stepping 1
0x31	Stepping 3.1
0xA3	Stepping A3
0xB1	Stepping B1
0xFF	Stepping information not provided

3.4.9 Bytes 353~381 (0x161~0x17D): Manufacturer's Specific Data

The module manufacturer may include any additional information desired into the module within these locations.

3.4.10 Byte 382~383 (0x17E~0x17F): Reserved

Must be coded as 0x00.

3.5 ASCII Decode Matrix for SPDs

Table 55 is a subset of the full ASCII standard which is used for coding bytes in the Serial Presence Detect EEPROM that require ASCII characters:

Table 55 — ASCII Decode Matrix for SPDs

	Second Hex Digit in Pair															
First Hex Digit in Pair	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
2	Blank Space								(	)				- Dash	. Period	
3	0	1	2	3	4	5	6	7	8	9						
4		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z					
6		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z					

Examples (shown in Table 56):

0x20 = Blank Space
0x34 = ‘4’
0x41 = ‘A’

Table 56 — SPD Bytes 329~348

Manufacturer’s PN	Coded in ASCII
13M32734BCD-260Y	31334D33323733344243442D323630592020

4 Module Specific Bytes for LP Memory Module Types (Bytes 128~255, 0x080~0x0FF)

This section defines the encoding of SPD bytes 128~255 when Memory Technology Key Byte 2 contains the value 0x0B and Module Type Key Byte 3 contains any of the following:

- 0xH7, LP-DIMM
- where H refers to the hybrid memory architecture, if any present on the module

Table 57 provides the SPD address map for the module specific section, bytes 128~255, of the SPD for LP-DIMM Module Types.

Table 57 — Module Specific SPD Bytes for LP-DIMM Module Types

Byte Number		Function Described	Notes
128	0x080	Raw Card Extension, Module Nominal Height	
129	0x081	Module Maximum Thickness	
130	0x082	Reference Raw Card Used	
132~253	0x084~0x0FD	Reserved -- Must be coded as 0x00	
254	0x0FE	CRC for Module Specific Section, Least Significant Byte	
255	0x0FF	CRC for Module Specific Section, Most Significant Byte	

4.1 Specific Byte Descriptions

4.1.1 Byte 128 (0x080) (LP-DIMM): Raw Card Extension, Module Nominal Height

The upper 3 bits of this byte, shown in Table 58, define extensions to the Raw Card Revision in Byte 130. The lower 5 bits of this byte define the nominal height (A dimension) in millimeters of the fully assembled module including heat spreaders or other added components. Refer to the relevant JEDEC JC-11 module outline (MO) documents for dimension definitions.

Table 58 — Byte 128 (0x080) (LP-DIMM): Raw Card Extension, Module Nominal Height

Bits 7~5	Bits 4~0
Raw Card Extension	**Module Nominal Height max, in mm (baseline height = 15 mm)**
000 = raw card revisions 0 to 3; see byte 130 001 = raw card revision 4 010 = raw card revision 5 011 = raw card revision 6 100 = raw card revision 7 101 = raw card revision 8 110 = raw card revision 9 111 = raw card revision 10	00000 = height ≤ 15 mm 00001 = 15 < height ≤ 16 mm 00010 = 16 < height ≤ 17 mm 00011 = 17 < height ≤ 18 mm 00100 = 18 < height ≤ 19 mm ... 01010 = 24 < height ≤ 25 mm 01011 = 25 < height ≤ 26 mm ... 01111 = 29 < height ≤ 30 mm 10000 = 30 < height ≤ 31 mm 10001 = 31 < height ≤ 32 mm ... 11111 = 45 mm < height

Examples (shown in Table 59):

Table 59 — Examples of Byte 128 (0x080

Nominal Module Height	Coding, bits 4~0	Meaning
mm	Binary	mm
18.75	00100	18 < height ≤ 19 mm
25.40	01011	25 < height ≤ 26 mm
30.00	01111	29 < height ≤ 30 mm
30.25	10000	30 < height ≤ 31 mm
31.25	10001	30 < height ≤ 31 mm

4.1.2 Byte 129 (0x081) (LP-DIMM): Module Maximum Thickness

This byte, shown in Table 60, defines the maximum thickness (E dimension) in millimeters of the fully assembled module including heat spreaders or other added components above the module circuit board surface. Thickness of the front of the module is calculated as the E1 dimension minus the PCB thickness. Thickness of the back of the module is calculated as the E dimension minus the E1 dimension, rounding up to the next integer. Refer to the relevant JEDEC JC-11 module outline (MO) documents for dimension definitions

Table 60 — Byte 129 (0x081) (LP-DIMM): Module Maximum Thickness

Bits 7~4	Bits 3~0
Module Maximum Thickness max, Back, in mm (baseline thickness = 1 mm)	**Module Maximum Thickness max, Front, in mm (baseline thickness = 1 mm)**
0000 = thickness ≤ 1 mm 0001 = 1 < thickness ≤ 2 mm 0010 = 2 < thickness ≤ 3 mm 0011 = 3 < thickness ≤ 4 mm ... 1110 = 14 < thickness ≤ 15 mm 1111 = 15 < thickness	0000 = thickness ≤ 1 mm 0001 = 1 < thickness ≤ 2 mm 0010 = 2 < thickness ≤ 3 mm 0011 = 3 < thickness ≤ 4 mm ... 1110 = 14 < thickness ≤ 15 mm 1111 = 15 < thickness
NOTE Thickness = E - E1	NOTE Thickness = E1 - PCB

4.1.3 Byte 130 (0x082) (LP-DIMM): Reference Raw Card Used

This byte, shown in Tables 61 and 62, indicates which JEDEC reference design raw card was used as the basis for the module assembly, if any. Bits 4~0 describe the raw card and bits 6~5 describe the revision level of that raw card. Special reference raw card indicator, ZZ, is used when no JEDEC standard raw card reference design was used as the basis for the module design. Pre-production modules should be encoded as revision 0 in bits 6~5.

Table 61 — Byte 130 (0x082) (LP-DIMM): Reference Raw Cards A through AL

Bit 7	Bits 6~5	Bits 4~0
Reference Raw Card Extension	**Reference Raw Card Revision**	**Reference Raw Card**
0 = Reference raw cards A through AL	00 = revision 0 01 = revision 1 10 = revision 2 11 = revision 3 See byte 128 for extensions beyond revision 3; these bits must be coded as 11 for all revisions greater than 3	When bit 7 = 0, 00000 = Reference raw card A 00001 = Reference raw card B 00010 = Reference raw card C 00011 = Reference raw card D 00100 = Reference raw card E 00101 = Reference raw card F 00110 = Reference raw card G 00111 = Reference raw card H 01000 = Reference raw card J 01001 = Reference raw card K 01010 = Reference raw card L 01011 = Reference raw card M 01100 = Reference raw card N 01101 = Reference raw card P 01110 = Reference raw card R 01111 = Reference raw card T 10000 = Reference raw card U 10001 = Reference raw card V 10010 = Reference raw card W 10011 = Reference raw card Y 10100 = Reference raw card AA 10101 = Reference raw card AB 10110 = Reference raw card AC 10111 = Reference raw card AD 11000 = Reference raw card AE 11001 = Reference raw card AF 11010 = Reference raw card AG 11011 = Reference raw card AH 11100 = Reference raw card AJ 11101 = Reference raw card AK 11110 = Reference raw card AL 11111 = ZZ (no JEDEC reference raw card design used)

Table 62 — Byte 130 (0x082) (LP-DIMM): Reference Raw Cards AM through CB

Bit 7	Bits 6~5	Bits 4~0
Reference Raw Card Extension	**Reference Raw Card Revision**	**Reference Raw Card**
1 = Reference raw cards AM through CB	00 = revision 0 01 = revision 1 10 = revision 2 11 = revision 3 See byte 128 for extensions beyond revision 3; these bits must be coded as 11 for all revisions greater than 3	When bit 7 = 1, 00000 = Reference raw card AM 00001 = Reference raw card AN 00010 = Reference raw card AP 00011 = Reference raw card AR 00100 = Reference raw card AT 00101 = Reference raw card AU 00110 = Reference raw card AV 00111 = Reference raw card AW 01000 = Reference raw card AY 01001 = Reference raw card BA 01010 = Reference raw card BB 01011 = Reference raw card BC 01100 = Reference raw card BD 01101 = Reference raw card BE 01110 = Reference raw card BF 01111 = Reference raw card BG 10000 = Reference raw card BH 10001 = Reference raw card BJ 10010 = Reference raw card BK 10011 = Reference raw card BL 10100 = Reference raw card BM 10101 = Reference raw card BN 10110 = Reference raw card BP 10111 = Reference raw card BR 11000 = Reference raw card BT 11001 = Reference raw card BU 11010 = Reference raw card BV 11011 = Reference raw card BW 11100 = Reference raw card BY 11101 = Reference raw card CA 11110 = Reference raw card CB 11111 = ZZ (no JEDEC reference raw card design used)

4.1.4 Bytes 131~253 (0x083~0x0FD) (LP-DIMM):

Reserved — must be coded as 0x00

4.1.5 Byte 254 (0x0FE) (LP-DIMM): Cyclical Redundancy Code (CRC) for Module Specific Section, LSB Byte 255 (0x0FF) (LP-DIMM): Cyclical Redundancy Code (CRC) for Module Specific Section, MSB

This two-byte field contains the calculated CRC for bytes 128~253 (0x080~0x0FD) in the SPD. LPDDR3 or LPDDR4 See bytes 126~127, in Section 3.1.38, for a coding example.