Intel® Atom™ Z8000 Processor Series

Datasheet (Volume 1 of 2)

For Volume 2 of 2 refer Document ID: 332066

March 2016

Revision 003

Document Number: 332065-003

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or visit www.intel.com/design/literature.htm.

Intel® Virtualization Technology requires a computer system with an enabled Intel® processor, updated BIOS, and virtual machine monitor (VMM). Functionality, performance or other benefits will vary depending on hardware and software configurations. Software applications may not be compatible with all operating systems. Check with your system manufacturer. Learn more at http://www.intel.com/go/virtualization.

Contents

Figures

Tables

Revision History

Document Number	Revision Number	Description	Revision Date
332065	001	• Initial release	March 2015
332065	002	• Added - Type3 SoC features and specifications included • Updated - Section 2.24, "Hardware Straps" strap pins updated. - Max. Imaging video resolution updated for T4 to 1080p30. - Table 124 VID values for all SKU's to match PRQ values. - Section 12.1, "SoC Storage Overview"	June 2015
332065	003	• Added - Intel® Atom™ processor X5-Z8350, Z8550, and Z8750. Updated Table 3, updated Table 124	March 2016

§ §

1 Introduction

The **Intel® Atom™ Z8000 Processor Series Datasheet** is the Intel Architecture (IA) SoC that integrates the next generation Intel® processor core, Graphics, Memory Controller, and I/O interfaces into a single system-on-chip solution.

The figures below shows the system level block diagram of the SoC. Refer the subsequent chapters for detailed information on the functionality of the different interface blocks.

Note: Throughout this document Intel® Atom™ Z8000 Processor Series is referred as SoC.

Section 1.3 lists the different features supported by the SoC packages.

1.1 References

Refer the following documents, which may be beneficial when reading this document or for additional information:

Document	Document Number
Intel® 64 and IA-32 Architectures Software Developer's Manuals Volume 1: Basic Architecture Volume 2A: Instruction Set Reference, A-M Volume 2B: Instruction Set Reference, N-Z Volume 3A: System Programming Guide Volume 3B: System Programming Guide	http://www.intel.com/products/processor/manuals/index.htm
Intel® Atom™ Z8000 Processor Series Datasheet (Volume 2 of 2)	332066
Intel® Atom™ Z8000 Processor Series Specification Update	332067

1.2 Terminology

Term	Description
ACPI	Advanced Configuration and Power Interface
Cold Reset	Full reset is when PWROK is de-asserted and all system rails except VCCRTC are powered down
DP	Display Port
DTS	Digital Thermal Sensor
EMI	Electro Magnetic Interference
eDP	embedded Display Port

Term	Description
HDMI	High Definition Multimedia Interface. HDMI supports standard, enhanced, or high-definition video, plus multi-channel digital audio on a single cable. HDMI transmits all Advanced Television Systems Committee (ATSC) HDTV standards and supports 8-channel digital audio, with bandwidth to spare for future requirements and enhancements (additional details available at http://www.hdmi.org/).
Intel® TXE	Intel® Trusted Execution Engine
LPDDR	Low Power Dual Data Rate memory technology.
LPE	Low Power Engine
MIPI CSI	MIPI Camera Interface Specification
MIPI DSI	MIPI Display Interface Specification
MP	Mega Pixel
MPEG	Moving Picture Experts Group
MSI	Message Signaled Interrupt. MSI is a transaction initiated outside the host, conveying interrupt information to the receiving agent through the same path that normally carries read and write commands.
MSR	Model Specific Register, as the name implies, is model-specific and may change from processor model number (n) to processor model number (n+1). An MSR is accessed by setting ECX to the register number and executing either the RDMSR or WRMSR instruction. The RDMSR instruction will place the 64 bits of the MSR in the EDX: EAX register pair. The WRMSR writes the contents of the EDX: EAX register pair into the MSR.
PWM	Pulse Width Modulation
POSM	Power on state machine
Rank	A unit of DRAM corresponding to the set of SDRAM devices that are accessed in parallel for a given transaction. For a 64-bit wide data bus using 8-bit (x8) wide SDRAM devices, a rank would be eight devices. Multiple ranks can be added to increase capacity without widening the data bus, at the cost of additional electrical loading.
SCI	System Control Interrupt. SCI is used in the ACPI protocol.
SDRAM	Synchronous Dynamic Random Access Memory
SERR	System Error. SERR is an indication that an unrecoverable error has occurred on an I/O bus.
SMC	System Management Controller or External Controller refers to a separate system management controller that handles reset sequences, sleep state transitions, and other system management tasks.
SMI	System Management Interrupt is used to indicate any of several system conditions (such as thermal sensor events, throttling activated, access to System Management RAM, chassis open, or other system state related activity).
SIO	Serial I/O
TMDS	Transition-Minimized Differential Signaling. TMDS is a serial signaling interface used in DVI and HDMI to send visual data to a display. TMDS is based on low-voltage differential signaling with 8/10b encoding for DC balancing.
Warm Reset	Warm reset is when both PMC_PLTRST# and PMC_CORE_PWROK are asserted.

1.3 SoC Packages

Table 1. SoC Packages (Sheet 1 of 2)

Interface	Category	T4	T3
CPU	No. of Cores	4	4
	Burst Speed	2.4 GHz	1.84 GHz[4]
GPU	Speed	Gen8-LP 12/16EU up to 600 MHz	Gen8-LP 12EU up to 500 MHz
Package Mechanical	Type	17x17mm Type 4	17x17mm Type 3
	IO count	628	378
	Ball count	1380	592
	ball pitch	0.4mm	0.65mm
	Z-height	0.937mm	1.002mm
Memory	Interface, Max transfer data rate	Dual Channel 2x64 bit, LPDDR3 - 1600MT/S	Single Channel 1x32/64 DDR3L-RS - 1600MT/s
	Type	BGA	BGA
	Capacity	2 - 8GB	1 - 2GB
PCIe	Number of ports	2	1
	Port Configuration	1x2, 2x1	x1
Imaging	Number of lanes	6	6
	Lane configuration	4+2, 3+2, 2+2+2	4+2, 3+2, 2+2
	Speed	1.5 GHz	1.5 GHz
	Still & Video	13MP ZSL, 1080p30	8MP, 1080p30
Media	Media decode rate	H.263, MPEG4, H.264, H.265 (HEVC), VP8, VP9, MVC, MPEG2, VC1, JPEG	H.263, MPEG4, H.264, H.265 (HEVC), VP8, VP9, MVC, MPEG2, VC1, JPEG
	Media encode rate	H.264, H.263, VP8, MVC, JPEG	H.264, H.263, VP8, MVC, JPEG
Audio	LPE (Low Power Engine)	3 I2S ports	3 I2S ports
USB devices	USB 3.0	3	Not Supported
	USB 3.0 OTG	1	1
	USB 2.0	-	3
	USB SSIC	2	Not Supported
	USB HSIC	2	2

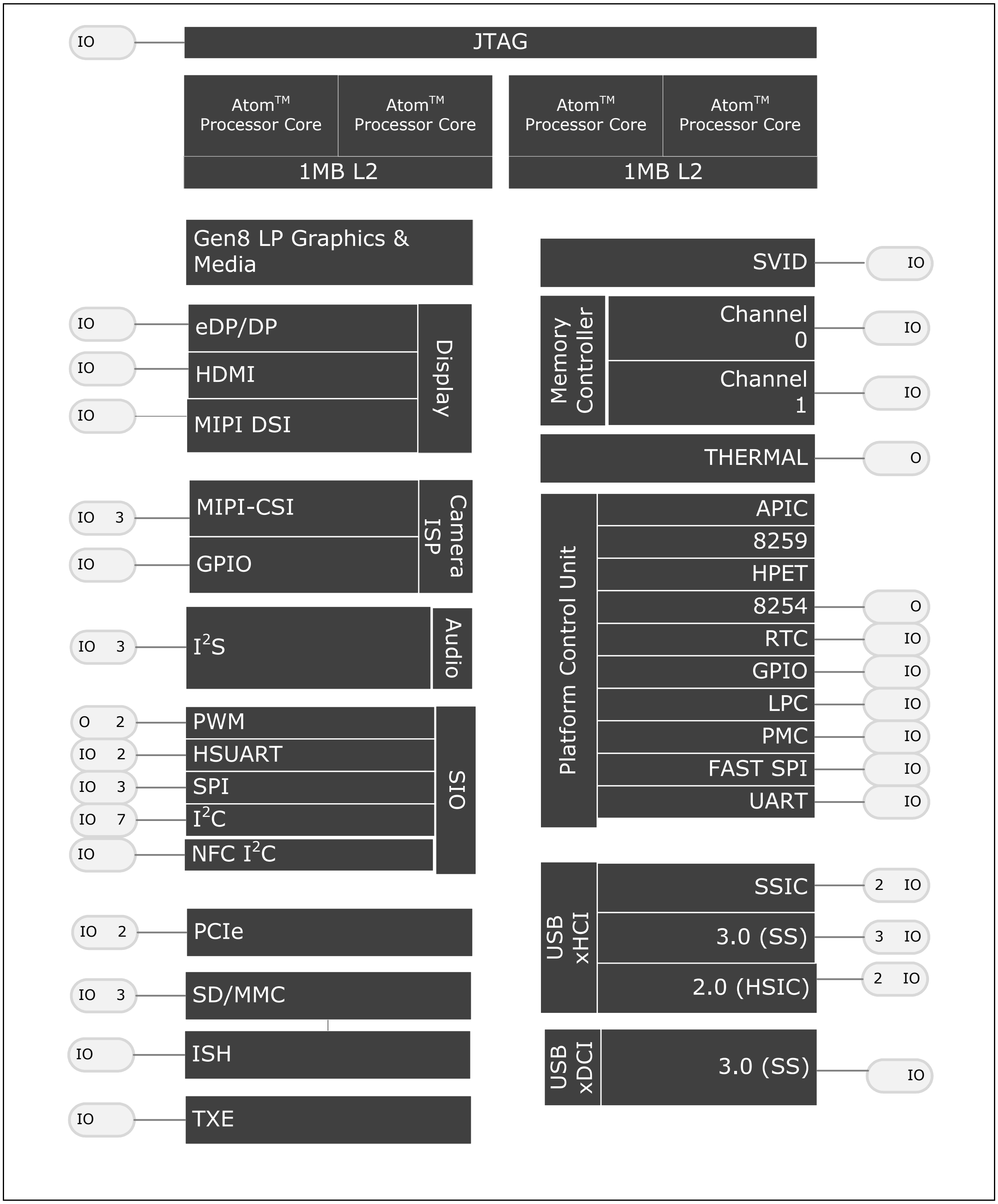
Table 1. SoC Packages (Sheet 2 of 2)

Interface	Category	T4	T3
SIO	LPC	YES	Not supported
	I2C	7	6
	I2C Max Speed	1.7 MHz	1.7 MHz
	I2C NFC	1	1
	I2C ISH	1	1
	SPI	3	Not supported[1]
	SPI Speed	Master Only up to 25 MHz	Master Only up to 25 MHz
	Fast SPI	Quad mode	Dual mode
Storage	SD Card	x1 SDR104	x1 SDR104 [2]
	SDIO	x1 SDR104	x1 SDR104
	eMMC	4.51	4.51
Display	DDI ports	x3	x2
	Max MIPI DSI Resolution	2560x1600 @60fps	1900x1200 @60fps
	MIPI-DSI ports	2x 4 Lanes @ 1Gbps	1x 4 Lanes @ 1Gbps
	Max eDP Resolution	2560x1600 @ 24bbp	1920x1080 @60fps
	eDP ports	2 (2x4 @2.7Gbps)	2 (2x4 @2.7Gbps)
	Max DP 1.1a Resolution	2560x1600 @60fps	2560x1600 @60fps
	Max HDMI 1.4b Resolution	3840x2160 @30fps	1920x1080 @60fps

NOTES:

1. One SPI port is multiplexed with reference clock signal which is GPIO signal, and the usage will be dependent on the GPIO configurations on the platform.
2. Is limited to DDR50 due to PMIC power delivery limitation.
3. MPO available on Display Pipe B only.
4. The Burst Speed mentioned is for 2 Cores.This is PRE-SMT package height.

Figure 1. SoC Block Diagram



1.4 Feature Overview

1.4.1 Processor Core

- Up to four IA-compatible low power Intel® processor cores
  — One thread per core
- Two-wide instruction decode, out of order execution.
- On-die, 32 KB 8-way L1 instruction cache and 24 KB 6-way L1 data cache per core.
- On-die, 1 MB, 16-way L2 cache, shared per two cores.
- 36-bit physical address, 48-bit linear address size support.
- Supported C-states: C0, C1, C6C, C6, C7.
- Supports Intel® Virtualization Technology (Intel® VT-x2).

1.4.2 System Memory Controller

- Memory Controller supports dual-channel DDR3L-RS/LPDDR3.
- Up to two ranks per channel (4 ranks in total).
- 32 Bit or 64 Bit data bus.
- Supports DDR3L-RS/LPDDR3 with 1600 MT/s data rate.
- Supports x32 LPDDR3 DRAM device data widths.
- Supports x16 DDR3L-RS DRAM device data widths.
- Total memory bandwidth supported is 12.8GB/s (for 1600 MT/s single-channel) to 25.6GB/s (for 1600 MT/s dual-channel).
- Supports different physical mappings of bank addresses to optimize performance.
- Supports Dynamic Voltage and Frequency Scaling.
- Out-of-order request processing to increase performance.
- Aggressive power management to reduce power consumption.
- Proactive page closing policies to close unused pages.

1.4.3 Display Controller

- Supports up to 3 Display pipes.
- Supports 2 MIPI DSI ports.
- Supports 3 DDI ports to configure eDP 1.3/DP 1.1a/DVI/HDMI 1.4b.
- Supports 2 panel power sequence for 2 eDP ports.
- Supports Audio on DP/HDMI.
- Supports Intel® Display Power Saving Technology (DPST) 6.0, Panel Self Refresh (PSR) and Display Refresh Rate Switching Technology (DRRS).

1.4.4 Graphics and Media Engine

- Intel's 8th generation (Gen 8) LP graphics and media encode/decode engine.
- Supports 3D rendering, media compositing and video encoding.
- Graphics Burst enabled through energy counters.
- Supports DX*11.1, OpenGL 4.3, OGL ES 3.0, OpenCL 1.2.
- 4x anti-aliasing.
- Full HW acceleration for decode of H.263, MPEG4, H.264, H.265 (HEVC), VP8, VP9, MVC, MPEG2, VC1, JPEG.
- Full HW acceleration for encode of H.264, H.263, VP8, MVC, JPEG.
- Supports Content protection using PAVP2.0, HDCP 1.4 (wired)/2.2 (wireless) and Media Vault DRM.

1.4.5 Image Signal Processor

- Supports up to three MIPI CSI ports.
- Supports up to 13MP sensors.

1.4.6 Power Management

- Supports ACPI 5.0.
- Processor Core states: C0, C1, C1E, C6C, C6 and C7.
- Display and Graphics device states: D0, D3.
- System sleep states: S0, S0ix, S4, S5.
- Support CPU and GFx Burst for selected SKUs.
- Dynamic I/O power reductions (disabling sense amps on input buffers, tristating output buffers).
- Dynamic memory self-refresh.

1.4.7 PCI Express*

- Supports x2 PCIe 2.0 compliant controller.
- Supports both Gen1 and Gen2 data rates.
- The controller provides a max data payload of 128B with the capability of splitting the request at 64B granularity.
- Supports autonomous up-configuration and autonomous down-configuration as target.

1.4.8 USB Controller

1.4.8.1 USB xHCI Controller

USB Host Controller supports:

- Two (2) Super Speed Inter-Chip (SSIC) port.
- Three (3) Super Speed (SS) ports [Backward Compatible of USB 2.0 HS/FS/LS].
- Two (2) High Speed Inter-Chip (HSIC) ports.

Note: SoC can support the 4th SS port when OTG port is in Host mode.

1.4.8.2 USB xDCI Controller

The SoC implements OTG block for device-mode functionality:

- Supports one USB 3.0 Super Speed port with backward compatibility of USB 2.0 High Speed and Low/Full Speed.
- Supports SuperSpeed OTG v3.0 device.
- Supports USB3 Debug Device Class Specification [USB3-debug].

1.4.9 Low Power Engine (LPE) Audio Controller

- Support 3 I2S ports.
- I2S and DDI with dedicated DMA.
- Supports MP3, AAC, AC3/DD+, WMA9, PCM (WAV).
- Provides HW acceleration for common audio and voice functions such as codecs, acoustic echo cancellation, noise cancellation.

1.4.10 Storage

1.4.10.1 Storage Control Cluster (eMMC, SDIO, SD)

- Supports one eMMC 4.51 controller
  - 200 MB/s Data rate
- Supports one SDIO 3.0 interface
  - 800 Mb/s Data rate
- Supports one SDXC controller
  - 800 Mb/s Data rate

1.4.11 Intel® Trusted Execution Engine (Intel® TXE)

Intel TXE is responsible for supporting and handling security related features.

- Supports MediaVault with OMA-DRM and One Time Password.
- Isolated execution environment for crypto operations.
- Supports secure boot - with customer programmable keys to secure code.

1.4.12 Serial I/O (SIO)

- Controller for external devices via SPI, UART, I^{2}C or PWM.
- Each port is multiplexed with general purpose I/O for configurations flexibility.
- Supports up to 7 I^{2}C, NFC I^{2}C, ISH I^{2}C, 2 HSUART, 2 PWM, 3 SPI interface.

1.4.13 Platform Control Unit (PCU)

Platform controller unit is a collection of HW blocks, including UART, debug/boot SPI and Intel legacy block (iLB), that are critical to implement a Windows* compatible platform. Some of its key features are:

- Universal Asynchronous Receiver/Transmitter (UART) with COM1 interface.
- A Fast Serial Peripheral Interface (SPI) for Flash only - stores boot FW and system configuration data.
- Intel Legacy Block (iLB) supports legacy PC platform features
  - — RTC, Interrupts, Timers and Peripheral interface (LPC for TPM) blocks.

1.4.14 Intel®Sensor Hub

Intel® Sensor Hub Supports:

- Acquisition / sampling of sensor data.
- The ability to combine data from individual sensors to create a more complex Virtual sensor that can be directly used by the firmware/OS.
- Low power operation through clock gating and power gating of parts of the ISH together with the ability to turn sensors off.
- The ability to operate independently when the host platform is in low power state.

1.4.15 Package

This SoC is packaged in a Flip-Chip Ball Grid Array (FCBGA) package.

The following table summarizes the package attributes for different SoC SkUs.

Table 2. Package Attributes

	Category	T4	T3
Package	Type	17x17mm Type 4	17x17mm Type 3
	IO count	628	378
	Ball count	1380	592
	ball pitch	0.4mm	0.65mm
	Z-height	0.937mm	1.002mm

1.4.16 SKU List

Table 3. SoC SKU List

Processor Number	Stepping	Package Type	SDP(W)	Core LFM (MHz)/ HFM (GHz)	Core max Burst (GHz)	Tjmax (°C)	TDP/ SDP Tj(°C)	GFx Normal /Burst (MHz)	No. of Graphics EU	Memory Channel	Memory Speed (MT/s)
Z8750	D-1	T4	2.0	480/1.6	2.56	90	70	400/600	16	2x64	LPDDR3-1600
Z8550	D-1	T4	2.0	480/1.44	2.4	90	70	400/600	12	2x64	LPDDR3-1600
Z8350	D-1	T3	2.0	480/1.44	1.92	90	70	400/500	12	1x64	DDR3L/ LPDDR3-1600
Z8700	C-0	T4	2.0	480/1.6	2.4	90	70	400/600	16	2x64	LPDDR3-1600
Z8500	C-0	T4	2.0	480/1.44	2.24	90	70	400/600	12	2x64	LPDDR3-1600
Z8300	C-0	T3	2.0	480/1.44	1.84	90	70	400/500	12	1x32/64	DDR3L-1600

Issue	XHCI controller stuck in reset during warm reboot cycles
Stepping	BXT Ax/Bx, APL Ax
Workaround	BIOS should program USB Host Controller Misc Reg 2 bit2 =1, set Bit5 for A0 stepping and clear for all other steppings
Impact of Workaround	
Notes	USB

Pad Name	GPIO#	Net Name
GP187_DDI0_DDC_SDA	GPIO_187	HDMI_DDC_SDA
GP188_DDI0_DDC_SCL	GPIO_188	HDMI_DDC_SCL
GP193_PNL0_VDDEN	GPIO_193	DISP0_VDDEN

§

2 Physical Interfaces

Many interfaces contain physical pins. These groups of pins make up the physical interfaces. Because of the large number of interfaces and the small size of the package, Some interfaces share their pins with GPIOs, while others use dedicated physical pins. This chapter summarizes the physical interfaces, including the diversity in GPIO multiplexing options.

2.1 Pin States

This section describes the states of each signal before, during and directly after reset. Additionally, Some signals have internal pull-up/pull-down termination resistors, and their values are also provided. All signals with the “” symbol are muxed and may not be available without configuration.

Table 4. Platform Power Well Definitions

Power Type	Voltage Range (V)	Power Well Description
VCC0/1	Refer Table 118	Variable voltage rail for core
VGG	Refer Table 118	Variable voltage rail for Graphics Core
VNN	Refer Table 118	Variable voltage rail for SoC.
V1P15	1.15	Fixed voltage rail for SoC, Graphics, camera
V1P05A	1.05	Fixed voltage rail for P-unit, LPE, TXE,I/O’s, PLL’s and ISH
V1P2A	1.24	Fixed voltage rail for I/O’s and PLL’s.
VDDQ	1.24/1.35	Fixed voltage rail for DDR PHY
VDDQG	1.24/1.35	Fixed voltage rail for DDR PHY
V1P8A	1.8	Fixed voltage rail for I/O’s.
V3P3A	3.3	Fixed voltage rail for I/O’s.
V3P3A_V1P8A	1.8/3.3	Fixed voltage rail for SDIO.
V3P3RTC	3.3	Voltage rai For RTC clock.

Table 5. Buffer Type Definitions (Sheet 1 of 2)

Buffer Type	Buffer Description
MIPI-DPHY	1.24 V tolerant MIPI DPHY buffer type
USB3 PHY	1.0 V tolerant USB3 PHY buffer type
USB2 PHY	1.8 V tolerant USB3 PHY buffer type
SSIC PHY	1.2 V tolerant SSIC PHY buffer type
HSIC PHY	1.2 V tolerant HSIC PHY buffer type
GPIO	GPIO buffer type. This can be of the following types: 1.8/3.3 V.

Table 5. Buffer Type Definitions (Sheet 2 of 2)

Buffer Type	Buffer Description
MODPHY	1.0 V tolerant MODPHY buffer type
DDR3	1.5 V tolerant DDR3 buffer type
Analog	Analog pins that do not have specific digital requirements. Often used for circuit calibration or monitoring.
GPIOMV, HS	GPIO Buffer type, Medium Voltage(1.8V),High Speed (FMAX~208Mhz)
GPIOMV, MS	GPIO Buffer type, Medium Voltage(1.8V),Medium Speed (FMAX~60Mhz)
GPIOMV, MS, CLK	GPIO Buffer type, Medium Voltage(1.8V),Medium Speed (FMAX~60Mhz), Clock
GPIOMV, HS, CLK	GPIO Buffer type, Medium Voltage(1.8V),High Speed (FMAX~208Mhz), Clock
GPIOMV, HS, RCOMP	GPIO Buffer type, Medium Voltage(1.8V),High Speed (FMAX~208Mhz), RCOMP
GPIOMV, MS, I2C	GPIO Buffer type, Medium Voltage(1.8V),Medium Speed (FMAX~60Mhz), I2C
GPIOHV, HS	GPIO Buffer type, High Voltage(1.8V/3.3V),High Speed (FMAX~208Mhz)
GPIOHV, HS, RCOMP	GPIO Buffer type, High Voltage(1.8V/3.3V),High Speed (FMAX~208Mhz), RCOMP

NOTE: GPIO mode, where register controlled will not hit FMAX speeds. It only matters when functionally used.

Table 6. Default Buffer State Definitions (Sheet 1 of 2)

Buffer State	Description
Z	The SoC places this output in a high-impedance state. For inputs, external drivers are not expected.
Do Not Care	The state of the input (driven or tristated) does not affect the processor. For outputs, it is assumed that the output buffer is in a high-impedance state.
V_{OH}	The SoC drives this signal high with a termination of 50 Ω.
V_{OL}	The SoC drives this signal low with a termination of 50 Ω.
Unknown	The processor drives or expects an indeterminate value.
V_{IH}	The SoC expects/requires the signal to be driven high.
V_{IL}	The SoC expects/requires the signal to be driven low.
"P" 1.1V	USB low speed Single ended 1.
Pull-up	This signal is pulled high by a pull-up resistor (internal or external — internal value specified in “Term” column).
Pull-down	This signal is pulled low by a pull-down resistor (internal or external — internal value specified in “Term” column).
Running	The clock is toggling, or the signal is transitioning.
Off	The power plane for this signal is powered down. The processor does not drive outputs, and inputs should not be driven to the processor. (VSS on output)

Table 6. Default Buffer State Definitions (Sheet 2 of 2)

Buffer State	Description
1	Buffer drives V_{OH}
0	Buffer drives V_{OL}
H	Buffer Hi Z, weak PU, default to 20K, unless explicitly specified otherwise
L	Buffer Hi Z, weak PD, default to 20K, unless explicitly specified otherwise
Input H	Input enable, weak PU
Output L	Output enable, weak PU
Pgm	Programmable
Retain	retain configuration/data prior to standby

2.2 System Memory Controller Interface Signals

2.2.1 DDR3L-RS

Table 7. DDR3L-RS System Memory Signals (Sheet 1 of 2)

				Default Buffer State		
Signal Name	Dir	Plat. Power	Type	Pwrgood Assert State	Resetout Deassert State	S0ix
DDR3_M0_MA[15:0]	O	V1P35	DDR	Z	Z	Z
DDR3_M0_CK[1,0]_P	O	V1P35	DDR	Z	Z	Z
DDR3_M0_CK[1,0]_N	O	V1P35	DDR	Z	Z	Z
DDR3_M0_CKE[3:0]	O	V1P35	DDR	Weak 0	0	0
DDR3_M0_CS[1,0]_N	O	V1P35	DDR	Z	Z	Z
DDR3_M0_CAS_N	O	V1P35	DDR	Z	Z	Z
DDR3_M0_RAS_N	O	V1P35	DDR	Z	Z	Z
DDR3_M0_WE_N	O	V1P35	DDR	Z	Z	Z
DDR3_M0_BS[2:0]	O	V1P35	DDR	Z	Z	Z
DDR3_M0_DRAMRST_N	O	V1P35	DDR	Weak 0	0	1
DDR3_M0_ODT[1,0]	O	V1P35	DDR	Z	Z	Z
DDR3_M0_DQ[63:0]	I/O	V1P35	DDR	Z	Z	Z
DDR3_M0_DM[7:0]	O	V1P35	DDR	Z	Z	Z
DDR3_M0_DQSP[7:0]	I/O	V1P35	DDR	Z	Z	Z
DDR3_M0_DQSN[7:0]	I/O	V1P35	DDR	Z	Z	Z
DDR3_M0_OCAVREF	O	V1P35	DDR	Z	Z	Z
DDR3_M0_ODQVREF	O	V1P35	DDR	Z	Z	Z

Table 7. DDR3L-RS System Memory Signals (Sheet 2 of 2)

Signal Name	Dir	Plat. Power	Type	Default Buffer State		
				Pwrgood Assert State	Resetout Deassert State	S0ix
DDR3_M0_RCOMPPD	I	V1P35	DDR	Z	Z	Z
DDR3_M1_MA[15:0]	O	V1P35	DDR	Z	Z	Z
DDR3_M1_CK[1,0]_P	O	V1P35	DDR	Z	Z	Z
DDR3_M1_CK[1,0]_N	O	V1P35	DDR	Z	Z	Z
DDR3_M1_CKE[3:0]	O	V1P35	DDR	Weak 0	0	0
DDR3_M1_CS[1,0]_N	O	V1P35	DDR	Z	Z	Z
DDR3_M1_CAS_N	O	V1P35	DDR	Z	Z	Z
DDR3_M1_RAS_N	O	V1P35	DDR	Z	Z	Z
DDR3_M1_WE_N	O	V1P35	DDR	Z	Z	Z
DDR3_M1_BS[2:0]	O	V1P35	DDR	Z	Z	Z
DDR3_M1_DRAMRST_N	O	V1P35	DDR	Weak 0	0	1
DDR3_M1_ODT[1,0]	O	V1P35	DDR	Z	Z	Z
DDR3_M1_DQ[63:0]	I/O	V1P35	DDR	Z	Z	Z
DDR3_M1_DM[7:0]	O	V1P35	DDR	Z	Z	Z
DDR3_M1_DQS[7:0]_P	I/O	V1P35	DDR	Z	Z	Z
DDR3_M1_DQS[7:0]_N	I/O	V1P35	DDR	Z	Z	Z
DDR3_M1_OCAVREF	O	V1P35	DDR	Z	Z	Z
DDR3_M1_ODQVREF	O	V1P35	DDR	Z	Z	Z
DDR3_M1_RCOMPPD	I	V1P35	DDR	Z	Z	Z
DDR3_DRAM_PWROK	I	V1P35	DDR	Input	Input	Input
DDR3_CORE_PWROK	I	V1P35	DDR	Input	Input	Input

2.2.2 LPDDR3

Table 8. LPDDR3 System Memory Signals (Sheet 1 of 2)

Signal Name	Dir	Plat. Power	Type	Default Buffer State		
				Pwrgood Assert State	Resetout Deassert State	S0ix
LPDDR3_M0_CA[9:0]	O	V1P24	DDR	Z	Z	Z
LPDDR3_M0_CK_P_A/B	O	V1P24	DDR	Z	Z	Z
LPDDR3_M0_CK_N_A/B	O	V1P24	DDR	Z	Z	Z

Table 8. LPDDR3 System Memory Signals (Sheet 2 of 2)

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
LPDDR3_M0_CKE[1:0]_A/B	O	V1P24	DDR	Weak 0	0	0
LPDDR3_M0_CS[1:0]_N	O	V1P24	DDR	Z	Z	Z
LPDDR3_M0_ODT_A/B	O	V1P24	DDR	Z	Z	Z
LPDDR3_M0_DQ[31:0]_A/B	I/O	V1P24	DDR	Z	Z	Z
LPDDR3_M0_DM[3:0]_A/B	O	V1P24	DDR	Z	Z	Z
LPDDR3_M0_DQS[3:0]_P_A/B	I/O	V1P24	DDR	Z	Z	Z
LPDDR3_M0_DQS[3:0]_N_A/B	I/O	V1P24	DDR	Z	Z	Z
LPDDR3_M0_OCAVREF	O	V1P24	DDR	Z	Z	Z
LPDDR3_M0_ODQVREF	O	V1P24	DDR	Z	Z	Z
LPDDR3_M0_RCOMPPD	I	V1P24	DDR	Z	Z	Z
LPDDR3_M1_CA[9:0]	O	V1P24	DDR	Z	Z	Z
LPDDR3_M1_CK_P_A/B	O	V1P24	DDR	Z	Z	Z
LPDDR3_M1_CK_N_A/B	O	V1P24	DDR	Z	Z	Z
LPDDR3_M1_CKE[1:0]_A/B	O	V1P24	DDR	Weak 0	0	0
LPDDR3_M0_CS[1:0]_N	O	V1P24	DDR	Z	Z	Z
LPDDR3_M0_ODT_A/B	O	V1P24	DDR	Z	Z	Z
LPDDR3_M1_DQ[31:0]_A/B	I/O	V1P24	DDR	Z	Z	Z
LPDDR3_M1_DM[3:0]_A/B	O	V1P24	DDR	Z	Z	Z
LPDDR3_M1_DQS[3:0]_P_A/B	I/O	V1P24	DDR	Z	Z	Z
LPDDR3_M1_DQS[3:0]_N_A/B	I/O	V1P24	DDR	Z	Z	Z
LPDDR3_M1_OCAVREF	O	V1P24	DDR	Z	Z	Z
LPDDR3_M1_ODQVREF	O	V1P24	DDR	Z	Z	Z
LPDDR3_M1_RCOMPPD	I	V1P24	DDR	Z	Z	Z
LPDDR3_DRAM_PWROK	I	V1P24	DDR	Input	Input	Input
LPDDR3_CORE_PWROK	I	V1P24	DDR	Input	Input	Input

2.3 USB Controller Interface Signals

2.3.1 USB2.0 Interface Signals

Table 9. USB2.0 Interface Signals

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
USB_DN[3:0]	I/O	V1P8	USB2 PHY	"P" 1.1V	"P" 1.1V	S0i3[1]
USB_DP[3:0]	I/O	V1P8	USB2 PHY	"P" 1.1V	"P" 1.1V	S0i3[1]
USB_OTG_ID	I/O	V1P8	USB2 PHY	Input, weak pull up	Input, weak pull up	Input
USB_VBUSSNS	I/O	V1P8	USB2 PHY	Input	Input	Input
USB_RCOMP	O	V1P8	USB2 PHY	Output	Output	Output
USB_OC[1:0]_N	I/O	V1P8	GPIOMV, MS	Input (20k PU)	Input (20k PU)	Input (20k PU)

NOTES:
1. [1]Depends on USB2 Mode.
2. USB 2.0 Port 0 is the OTG port.

2.3.2 USB HSIC Interface Signals

Table 10. USB 2.0 HSIC Interface Signals

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
USB_HSIC0_DATA	I/O	V1P2	HSIC Buffer	Weak 0	Weak 0	Weak 0
USB_HSIC0_STROBE	I/O	V1P2	HSIC Buffer	Weak 1	Weak 1	Weak 1
USB_HSIC1_DATA	I/O	V1P2	HSIC Buffer	Weak 0	Weak 0	Weak 0
USB_HSIC1_STROBE	I/O	V1P2	HSIC Buffer	Weak 1	Weak 1	Weak 1
USB_HSIC_RCOMP	I	V1P2	HSIC Buffer	Z	Z	Z

NOTE: The HSIC should be reset after SoC.

2.3.3 USB3.0 Interface Signals

2.3.3.1 USB 3.0 Interface Signals

Table 11. USB 3.0 Interface Signals

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
USB3_TXN[3:0]	O	V1P05A	USB3	X	Z	Output
USB3_TXP[3:0]	O	V1P05A	USB3	X	Z	Output
USB3_RXN[3:0]	I	V1P05A	USB3	X	Z	Input
USB3_RXP[3:0]	I	V1P05A	USB3	X	Z	Input
USB3_RCOMP_N	I	V1P05A	USB3	X	Output	Off
USB3_RCOMP_P	I	V1P05A	USB3	X	Output	Off

NOTE: USB3.0 Port 0 is the OTG port.

2.3.3.2 USB SSIC Interface Signals

Table 12. USB SSIC Interface Signals

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
USB_SSIC_RX_N[0,1]	I/O	V1P24	SSIC PHY	Input	Input	Input
USB_SSIC_RX_P[0,1]	I/O	V1P24	SSIC PHY	Input	Input	Input
USB_SSIC_TX_N[0,1]	I/O	V1P24	SSIC PHY	Z	Output	Output
USB_SSIC_TX_P[0,1]	I/O	V1P24	SSIC PHY	Z	Output	Output
USB_SSIC_RCOMP_N	O	V1P24	SSIC PHY	Output	Output	Output
USB_SSIC_RCOMP_P	O	V1P24	SSIC PHY	Output	Output	Output

2.4 Integrated Clock Interface Signals

Table 13. Integrated Clock Interface Signals

Signal Name	Dir	Plat. Power	Type	Default Buffer State Pwrgood Assert State	Resetout Deassert State	S0ix
ICLK_OSCIN	I	V1P0	Crystal Oscillator	Input (Crystal)	Input (Crystal)	Input (Crystal)
ICLK_OSCOUT	O	V1P0	Crystal Oscillator	Output (Crystal)	Output (Crystal)	Output (Crystal)
ICLK_ICOMP	O	Analog	Analog	Input	Input	Input
ICLK_RCOMP	O	Analog	Analog	Input	Input	Input

2.5 Display - Digital Display Interface (DDI) Signals

Table 14. Digital Display Interface Signals (Sheet 1 of 2)

Signal Name	Dir	Plat. Power	Type	Default Buffer State Pwrgood Assert State	Resetout Deassert State	S0ix
DDI0_TXP[3:0]	O	V1P24	DDI	Z	Output	Output
DDI0_TXN[3:0]	O	V1P24	DDI	Z	Output	Output
DDI0_AUXP	I/O	V1P24	DDI	Z	Output	Output
DDI0_AUXN	I/O	V1P24	DDI	Z	Output	Output
DDI0_BKLTCTL	I/O	V1P8	GPIOMV, MS	0	0	0
DDI0_BKLTEN	I/O	V1P8	GPIOMV, MS	0	0	0
DDI0_DDC_CLK	I/O	V1P8	GPIOMV, MS, CLK	Input (20k PU)	Input (20k PU)	Input (20k PU)
DDI0_DDC_DATA	I/O	V1P8	GPIOMV, MS	Input (20k PU)	Input (20k PU)	Input (20k PU)
DDI0_HPD	I/O	V1P8	GPIOMV, MS	Input (20k PD)	Input (20k PD)	Input (20k PD)
DDI0_VDDEN	I/O	V1P8	GPIOMV, MS	0	0	0
DDI0_RCOMP_N	O	V1P24	DDI	Z	Output	Output
DDI0_RCOMP_P	O	V1P24	DDI	Z	Output	Output
DDI1_TXP[3:0]	O	V1P24	DDI	Z	Output	Output
DDI1_TXN[3:0]	O	V1P24	DDI	Z	Output	Output
DDI1_AUXP	I/O	V1P24	DDI	Z	Output	Output

Table 14. Digital Display Interface Signals (Sheet 2 of 2)

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
DDI1_AUXN	I/O	V1P24	DDI	Z	Output	Output
DDI1_BKLTCTL	I/O	V1P8	GPIOMV, MS	0	0	0
DDI1_BKLTEN	I/O	V1P8	GPIOMV, MS	0	0	0
DDI1_DDC_CLK	I/O	V1P8	GPIOMV, MS, CLK	Input (20k PU)	Input (20k PU)	Input (20k PU)
DDI1_DDC_DATA	I/O	V1P8	GPIOMV, MS	Input (20k PU)	Input (20k PU)	Input (20k PU)
DDI1_HPD	I/O	V1P8	GPIOMV, MS	Input (20k PD)	Input (20k PD)	Input (20k PD)
DDI1_VDDEN	I/O	V1P8	GPIOMV, MS	0	0	0
DDI1_RCOMP_N	O	V1P24	DDI	Z	Output	Output
DDI1_RCOMP_P	O	V1P24	DDI	Z	Output	Output
DDI2_DDC_CLK	I/O	V1P8	DDI	Z	Output	Output
DDI2_DDC_DATA	I/O	V1P8	DDI	Z	Output	Output
DDI2_TXP[3:0]	O	V1P24	DDI	Z	Output	Output
DDI2_TXN[3:0]	O	V1P24	DDI	Z	Output	Output
DDI2_AUXP	I/O	V1P24	DDI	Z	Output	Output
DDI2_AUXN	I/O	V1P24	DDI	Z	Output	Output
DDI2_HPD	I/O	V1P8	GPIOMV, MS	Input (20k PD)	Input (20k PD)	Input (20k PD)

2.6 MIPI DSI Interface Signals

Table 15. MIPI DSI Interface Signals (Sheet 1 of 2)

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
MDSI_A_CLKN	O	V1P24	MIPI-DPHY	0	0	0
MDSI_A_CLKP	O	V1P24	MIPI-DPHY	0	0	0
MDSI_A_DN[3:0]	I/O	V1P24	MIPI-DPHY	0	0	0
MDSI_A_DP[3:0]	I/O	V1P24	MIPI-DPHY	0	0	0
MDSI_C_CLKN	O	V1P24	MIPI-DPHY	0	0	0
MDSI_C_CLKP	O	V1P24	MIPI-DPHY	0	0	0

Table 15. MIPI DSI Interface Signals (Sheet 2 of 2)

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
MDSI_C_DN[3:0]	I/O	V1P24	MIPI-DPHY	0	0	0
MDSI_C_DP[3:0]	I/O	V1P24	MIPI-DPHY	0	0	0
MDSI_RCOMP	I/O	V1P24	MIPI-DPHY	0	0	0

2.7 MIPI Camera Serial Interface (CSI) and ISP Interface Signals

Table 16. MIPI CSI Interface Signals

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
MCSI_1_CLKN	I	V1P24	MIPI-DPHY	Input	Input	Input
MCSI_1_CLKP	I	V1P24	MIPI-DPHY	Input	Input	Input
MCSI_1_DN[0:3]	I	V1P24	MIPI-DPHY	Input	Input	Input
MCSI_1_DP[0:3]	I	V1P24	MIPI-DPHY	Input	Input	Input
MCSI_2_CLKN	I	V1P24	MIPI-DPHY	Input	Input	Input
MCSI_2_CLKP	I	V1P24	MIPI-DPHY	Input	Input	Input
MCSI_2_DN[0:1]	I	V1P24	MIPI-DPHY	Input	Input	Input
MCSI_2_DP[0:1]	I	V1P24	MIPI-DPHY	Input	Input	Input
MCSI_3_CLKN	I	V1P24	MIPI-DPHY	Input	Input	Input
MCSI_3_CLKP	I	V1P24	MIPI-DPHY	Input	Input	Input
MCSI_RCOMP	I/O	V1P24	MIPI-DPHY	Input	Input	Input

2.8 PCI Express Signals

Table 17. PCIe Signals and Clocks

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
PCIE_RXN[0:1]	I	V1P05	PCIe PHY	X	Weak Pull Down	Input
PCIE_RXP[0:1]	I	V1P05	PCIe PHY	X	Weak Pull Down	Input
PCIE_TXN[0:1]	O	V1P05	PCIe PHY	X	Z	Output
PCIE_TXP[0:1]	O	V1P05	PCIe PHY	X	Z	Output
P_RCOMP_N	IO	X	PCIe PHY	X		Off
P_RCOMP_P	IO	X	PCIe PHY	X		Off
PCIE_CLKREQ[0:1]_N	IO	V1P8	GPIOMV, MS	X	Input (20k PU)	Prg

2.9 Low Power Engine (LPE) for Audio (I^{2}S) Interface Signals

Table 18. LPE Interface Signals

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
LPE_I2S[2:0]_CLK	I/O	V1P8	GPIOMV, MS	Input (20k PD)	Input (20k PD)	0
LPE_I2S[2:0]_FRM	I/O	V1P8	GPIOMV, MS	Input (20k PD)	Input (20k PD)	1
LPE_I2S[2:0]_DATAOUT	I/O	V1P8	GPIOMV, MS	0 (20k PD)	0 (20k PD)	0
LPE_I2S[2:0]_DATAIN	I/O	V1P8	GPIOMV, MS	Input (20k PD)	Input (20k PD)	Input

2.10 Storage Interface Signals

2.10.1 Storage Controller (eMMC, SDIO, SD)

Table 19. Storage Controller (eMMC, SDIO, SD) Interface Signals

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
MMC1_D[7:0]	I/O	V1P8	GPIOMV, HS	Z (20k PU)	Z (20k PU)	Z (20k PU)
MMC1_CMD	I/O	V1P8	GPIOMV, HS	Z (20k PU)	Z (20k PU)	Z (20k PU)
MMC1_CLK	I/O	V1P8	GPIOMV, HS, CLK	0 (20k PD)	0 (20k PD)	0 (20k PD)
MMC1_RCLK	I/O	V1P8	GPIOMV, HS	Z (20k PD)	Z	Z
MMC1_RESET_N	I/O	V1P8	GPIOMV, HS	Z	Z	Z
MMC1_RCOMP	I/O	V1P8	GPIOMV, HS, RCOMP	Z	Z	Z
SD2_D[2:0]	I/O	V1P8	GPIOMV, HS	Z (20k PU)	Z (20k PU)	Z (20k PU)
SD2_D[3]_CD_N	I/O	V1P8	GPIOMV, HS	Z (20k PU)	Z (20k PU)	Z (20k PU)
SD2_CMD	I/O	V1P8	GPIOMV, HS	Z (20k PU)	Z (20k PU)	Z (20k PU)
SD2_CLK	I/O	V1P8	GPIOMV, HS, CLK	0 (20k PD)	0	0
SD3_D[3:0]	I/O	V1P8/ V3P3	GPIOHV, HS	Z (20k PU)	Z (20k PU)	Z (20k PU)
SD3_CMD	I/O	V1P8/ V3P3	GPIOHV, HS	Z (20k PU)	Z (20k PU)	Z (20k PU)
SD3_PWREN_N	I/O	V1P8	GPIOMV, HS	1 (20k PD)	1	-
SD3_CLK	I/O	V1P8/ V3P3	GPIOHV, HS, CLK	0 (20k PD)	0	0
SD3_RCOMP	I/O	V1P8/ V3P3	GPIOHV, HS, RCOMP	Z	Z	Z
SD3_1P8_EN	I/O	V1P8	GPIOMV, HS	0 (20k PD)	0	-
SD3_CD_N	I/O	V1P8	GPIOMV, HS	Input (20k PU)	Input (20k PU)	Input (20k PU)

2.11 High Speed UART Interface Signals

Table 20. High Speed UART Interface Signals

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
UART1_DATAIN	I/O	V1P8	GPIOMV, MS	Input (20k PU)	Input (20k PU)	Input (20k PU)
UART1_DATAOUT	I/O	V1P8	GPIOMV, MS	1 (20k PU)	1	1
UART1_RTS_N	I/O	V1P8	GPIOMV, MS	1 (20k PU)	1	1
UART1_CTS_N	I/O	V1P8	GPIOMV, MS	Input (20k PU)	Input (20k PU)	Input (20k PU)
UART2_DATAIN	I/O	V1P8	GPIOMV, MS	Input (20k PU)	Input (20k PU)	Input (20k PU)
UART2_DATAOUT	I/O	V1P8	GPIOMV, MS	1 (20k PU)	1	1
UART2_RTS_N	I/O	V1P8	GPIOMV, MS	1 (20k PU)	1	1
UART2_CTS_N	I/O	V1P8	GPIOMV, MS	Input (20k PU)	Input (20k PU)	Input (20k PU)

2.12 I²C Interface Signals

Table 21. I²C Interface Signals

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
I2C0_DATA	I/O	V1P8	GPIOMV, MS, I2C	Z (1k PU, OD)	Z (1k PU, OD)	Z (1k PU, OD)
I2C0_CLK	I/O	V1P8	GPIOMV, MS, I2C	Z (1k PU, OD)	Z (1k PU, OD)	Z (1k PU, OD)
I2C1_DATA	I/O	V1P8	GPIOMV, MS, I2C	Input (20k PU)	Z (20k PU, OD)	Z (20k PU, OD)
I2C1_CLK	I/O	V1P8	GPIOMV, MS, I2C	Input (20k PU)	Z (20k PU, OD)	Z (20k PU, OD)
I2C2_DATA	I/O	V1P8	GPIOMV, MS, I2C	Input (20k PU)	Z (20k PU, OD)	Z (20k PU, OD)
I2C2_CLK	I/O	V1P8	GPIOMV, MS, I2C	Input (20k PU)	Z (20k PU, OD)	Z (20k PU, OD)
I2C3_DATA	I/O	V1P8	GPIOMV, MS, I2C	Input (20k PU)	Z (20k PU, OD)	Z (20k PU, OD)
I2C3_CLK	I/O	V1P8	GPIOMV, MS, I2C	Input (20k PU)	Z (20k PU, OD)	Z (20k PU, OD)
I2C4_DATA	I/O	V1P8	GPIOMV, MS, I2C	Input (20k PU)	Z (20k PU, OD)	Z (20k PU, OD)
I2C4_CLK	I/O	V1P8	GPIOMV, MS, I2C	Input (20k PU)	Z (20k PU, OD)	Z (20k PU, OD)
I2C5_DATA	I/O	V1P8	GPIOMV, MS, I2C	Input (20k PU)	Z (20k PU, OD)	Z (20k PU, OD)
I2C5_CLK	I/O	V1P8	GPIOMV, MS, I2C	Input (20k PU)	Z (20k PU, OD)	Z (20k PU, OD)
I2C6_DATA	I/O	V1P8	GPIOMV, MS, I2C	Input (20k PU)	Z (20k PU, OD)	Z (20k PU, OD)
I2C6_CLK	I/O	V1P8	GPIOMV, MS, I2C	Input (20k PU)	Z (20k PU, OD)	Z (20k PU, OD)

2.13 NFC I^{2}C Interface Signals

Table 22. NFC I^{2}C Interface Signals

Signal Name	Dir	Plat. Power	Type	Default Buffer State Pwrgood Assert State	Resetout Deassert State	S0ix
NFC_I2C_DATA	I/O	V1P8	GPIOMV, MS, I2C	Input (20k PU)	Z (20k PU, OD)	Z (20k PU, OD)
NFC_I2C_CLK	I/O	V1P8	GPIOMV, MS, I2C	Input (20k PU)	Z (20k PU, OD)	Z (20k PU, OD)
GPIO_ALERT	I/O	V1P8	GPIOMV, MS	0 (20k PU)	0	0

2.14 PCU- Fast Serial Peripheral Interface (SPI) Signals

Table 23. PCU- Fast Serial Peripheral Interface (SPI) Signals

Signal Name	Dir	Plat. Power	Type	Default Buffer State Pwrgood Assert State	Resetout Deassert State	S0ix
FST_SPI_CLK	I/O	V1P8	GPIOMV, HS	0 (20k PU)	Output	Output
FST_SPI_CS[0]_N	I/O	V1P8	GPIOMV, HS	1 (20k PU)	Output	Output
FST_SPI_CS[1]_N	I/O	V1P8	GPIOMV, HS	Input (20k PU)	Output	Output
FST_SPI_CS[2]_N	I/O	V1P8	GPIOMV, HS	1 (20k PU)	Output	Output
FST_SPI_D[3:0]	I/O	V1P8	GPIOMV, HS	Input (20k PU)	Input (20k PU)	Input (20k PU)

2.15 PCU - Real Time Clock (RTC) Interface Signals

Table 24. PCU - Real Time Clock (RTC) Interface Signals

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
RTC_X1	I	V3P3	RTC PHY	Input (Crystal)	Input (Crystal)	Input (Crystal)
RTC_X2	O	V3P3	RTC PHY	Output (Crystal)	Output (Crystal)	Output (Crystal)
RTC_RST_N	I	V3P3	RTC PHY	Input	Input	Input
RTC_TEST_N	I	V3P3	RTC PHY	Input	Input	Input
RTC_EXTPAD	O	V3P3	RTC PHY	Input	Input	Input

2.16 PCU - Low Pin Count (LPC) Bridge Interface Signals

Table 25. PCU - LPC Bridge Interface Signals

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
LPC_AD[3:0]	I/O	V3P3/ V1P8	GPIOHV, HS	Input (20k PU)	Input (20k PU)	Input (20k PU)
LPC_FRAME_N	I/O	V3P3/ V1P8	GPIOHV, HS	1 (20k PU)	1	1
LPC_SERIRQ	I/O	V1P8	GPIOHV, HS	Input (20k PU)	Input (20k PU)	Input (20k PU)
LPC_CLKRUN_N	I/O	V3P3/ V1P8	GPIOHV, HS	Input (20k PU)	Input (20k PU)	Input (20k PU)
LPC_CLKOUT[0]	I/O	V3P3/ V1P8	GPIOHV, HS	0 (20k PU)	Clock	0
LPC_CLKOUT[1]	I/O	V3P3/ V1P8	GPIOHV, HS	Input (20k PD)	Input	Input
LPC_RCOMP	I/O	V3P3/ V1P8	GPIOHV, HS	Z	Z	Z

2.17 PCU - Power Management Controller (PMC) Interface Signals

Table 26. PCU - Power Management Controller (PMC) Interface Signals

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
PMC_PLTRST_N	I/O	V1P8	GPIOMV, MS	0 (20k PU)	1	1
PMC_PWRBTN_N	I	V1P8	GPIOMV, MS	Input (20k PU)	Input (20k PU)	Input (20k PU)
PMC_RSTBTN_N	I/O	V1P8	GPIOMV, MS	Input (20k PU)	Input (20k PU)	Input (20k PU)
PMC_SUSPWRDNACK	I/O	V1P8	GPIOMV, MS	0 (20k PD)	0 (20k PD)	0 (20k PD)
PMC_SUS_STAT_N	I/O	V1P8	GPIOMV, MS	0 (20k PU)	1	0
PMC_SUSCLK[0]	I/O	V1P8	GPIOMV, MS	0 (20k PD)	32 KHz Clock	32 KHz Clock
PMC_SLP_S4_N	I/O	V1P8	GPIOMV, MS	0 (20k PU)	1	1
PMC_SLP_S0ix_N	I/O	V1P8	GPIOMV, MS	0 (20k PU)	1	0 at S0ix2
PMC_ACPRESENT	I/O	V1P8	GPIOMV, MS	Input (20k PD)	Input (20k PD)	Input (20k PD)
PMC_BATLOW_N	I/O	V1P8	GPIOMV, MS	Input (20k PU)	Input (20k PU)	Input (20k PU)
PMC_WAKE_N	I/O	V1P8	GPIOMV, MS	Input (20k PU)	Input (20k PU)	Input (20k PU)
PMC_CORE_PWROK	I	V3P3	RTC PHY	Input	Input	Input
PMC_RSMRST_N	I	V3P3	RTC PHY	Input	Input	Input

2.18 Serial Peripheral Interface (SPI) Signals

Table 27. Serial Peripheral Interface (SPI) Signals

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
SPI[1,2,3]_CLK	I/O	V1P8	GPIOMV, HS	0 (20k PU)	0	0
SPI[1,2,3]_CS[0:1]_N	I/O	V1P8	GPIOMV, HS	1 (20k PU)	1	1
SPI[1,2,3]_MOSI	I/O	V1P8	GPIOMV, HS	0 (20k PU)	0	0
SPI[1,2,3]_MISO	I/O	V1P8	GPIOMV, HS	Input (20k PU)	Input (20k PD)	Input

2.19 JTAG Interface Signals

Table 28. JTAG Interface Signals

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
JTAG_TCK	I/O	V1P8	GPIOMV, MS	Input (5k PD)	Input (5k PD)	Input (5k PD)
JTAG_TDI	I/O	V1P8	GPIOMV, MS	Input (5k PU)	Input (5k PU)	Input (5k PU)
JTAG_TDO	I/O	V1P8	GPIOMV, MS	Z	Z	Z
JTAG_TMS	I/O	V1P8	GPIOMV, MS	Input (5k PU)	Input (5k PU)	Input (5k PU)
JTAG_TRST_N	I/O	V1P8	GPIOMV, MS	Input (5k PU)	Input (5k PU)	Input (5k PU)
JTAG_PRDY_N	I/O	V1P8	GPIOMV, MS	Z (5k PU, OD)	Output (5k PU, OD)	Z (5k PU, OD)
JTAG_PREQ_N	I/O	V1P8	GPIOMV, MS	Input (5k PU, OD)	Input (5k PU, OD)	Input (5k PU, OD)

2.20 Integrated Sensor Hub Interface Signals

Table 29. Integrated Sensor Hub Interface Signals (Sheet 1 of 2)

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
ISH_GPIO[7:0]	I/O	V1P8	GPIOMV, MS	Input (20k PD)	Z (20k PU)	Z (20k PU)

Table 29. Integrated Sensor Hub Interface Signals (Sheet 2 of 2)

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
ISH_GPIO[8]	I/O	V1P8	GPIOMV, MS	Input (20k PU)	Z (20k PU, OD)	Z (20k PU, OD)
ISH_GPIO[9]	I/O	V1P8	GPIOMV, MS	Input (20k PU)	Z (20k PU, OD)	Z (20k PU, OD)
ISH_I2C1_SDA	I/O	V1P8	GPIOMV, MS, I2C	Input (20k PU)	Z (20k PU, OD)	Z (20k PU, OD)
ISH_I2C1_CLK	I/O	V1P8	GPIOMV, MS, I2C	Input (20k PU)	Z (20k PU, OD)	Z (20k PU, OD)

2.21 PWM Interface Signals

Table 30. PWM Interface Signal

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
PWM[0]	I/O	V1P8	GPIOMV, MS	0 (20k PD)	0	0
PWM[1]	I/O	V1P8	GPIOMV, MS	0 (20k PU)	0	0

2.22 SVID Interface Signals

Table 31. SVID Interface Signal

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
SVID_DATA	I/O	V1P8	GPIOMV, MS	0	0	1 or Z
SVID_CLK	I/O	V1P8	GPIOMV, MS	0	1 or Z	1 or Z
SVID_ALERT_N	I/O	V1P8	GPIOMV, MS	Input	Input	Input

2.23 Miscellaneous Signals

Table 32. Miscellaneous Signals and Clocks

				Default Buffer State		
Signal Name	**Dir**	**Plat. Power**	**Type**	**Pwrgood Assert State**	**Resetout Deassert State**	**S0ix**
PLT_CLK[0:5]	I/O	V1P8	GPIOMV, MS	0 (20k PD)	Clock (20k PD)	0[1]
PROCHOT_N	I/O	V1P8	GPIOMV, MS	Z	Z	Z

NOTE: ‘0’ in S0i2 or below.

2.24 Hardware Straps

All straps are sampled on the rising edge of **PMC_RSMRST_N**.

While **PMC_RSMRST_N** is low all strap pins are in input mode. Weak pull ups or downs keep straps from floating during this time. Strap values can be changed by driving the strap pins or using stronger pull resistors.

Table 33. Straps (Sheet 1 of 2)

Signal Name	Purpose	Pull up/Pull Down	Strap Description
GPIO_SUS[0][1]	DDI0 Detect	Weak internal pull down of 20K	DDI0 Detect 0 = DDI0 not enabled 1 = DDI0 enabled
GPIO_SUS[1]	DDI1 Detect	Weak internal pull down of 20K	DDI1 Detect 0 = DDI1 not enabled 1 = DDI1 enabled
GPIO_SUS[2]	A16 swap overdrive	Weak internal pull up of 20K	Top Swap (A16 Override) 0 = Change Boot Loader address 1 = Normal Operation
GPIO_SUS[3]	DSI Display Detect	Weak internal pull down of 20K	MIPI DSI Detect 0 = DSI not enabled 1 = DSI enabled
GPIO_SUS[4]	Boot BIOS Strap BBS	Weak internal pull up of 20K	BIOS Boot Selection 0 = Default 1 = SPI
GPIO_SUS[5]	Flash Descriptor Security Override	Weak internal pull up of 20K	Security Flash Descriptors 0 = Override 1 = Normal Operation

Table 33. Straps (Sheet 2 of 2)

Signal Name	Purpose	Pull up/Pull Down	Strap Description
GPIO_SUS[8]	ICLK, USB2, DDI SFR Supply Select	Weak internal pull up of 20K	0 = Supply is 1.25V 1 = Supply is 1.35V This strap also contains PLL LDO 0: supply is 1.25V 1: supply is 1.35V. Selects supply voltage for LDOs used for PLLs, thermal oscillators, USB2, iCLK and DDI
GPIO_SUS[9]	ICLK, USB2, DDI SFR Bypass	Weak internal pull down of 20K	Bypasses LDOs for ICLK 0 = Use LDOs 1 = Bypass LDOs (Supply1.05V on power pins)
GPIO_SUS[10]	POSM Select	Weak internal pull down of 20K	Selects which POSM (power on state machine) will be observed at time 0 0 = Fuse controller 1 = PMC
GPIO_CAMERASB08	ICLK Xtal OSC Bypass	Weak internal pull down of 20K	0 = No Bypass 1 = Bypass
GPIO_CAMERASB09	CCU SUS RO Bypass	Weak internal pull down of 20K	0 = No Bypass 1 = Bypass
GPIO_CAMERASB11	RTC OSC Bypass	Weak internal pull down of 20K	0 = No Bypass 1 = Bypass

NOTE:

1. Ignore this strap and use a software mechanism to detect the relevant DDI port. This signal can be used as a GPIO.

2.25 SoC RCOMP List

Table 34. RCOMP's List (Sheet 1 of 2)

Interface Name	RCOMP Name	Bias	Remarks
DDR3	DDR3_M0_RCOMPPD/ LPDDR3_M0_RCOMPPD	182 Ohm ±1% to Ground	RCOMP pins for DDR3
	DDR3_M1_RCOMPPD LPDDR3_M1_RCOMPPD	182 Ohm ±1% to Ground	
MIPI DSI	MDSI_RCOMP	150 Ohm ±1% to Ground	RCOMP pin for MIPI DSI
MIPI CSI	MCSI_RCOMP	150 Ohm ±1% to Ground	RCOMP pin for MIPI CSI
eMMC	MMC1_RCOMP	100 Ohm ±1% to Ground	eMMC, SDIO, FST_SPI RCOMP
SD Card	SD3_RCOMP	80.6 Ohm ±1% to Ground	SD Card contains its own RCOMP as it can be either 1.8V or 3.3V. Special care is needed to perform an RCOMP any time a card is inserted.
LPC	LPC_RCOMP	100 Ohm ±1% to Ground	LPC has its own RCOMP because it can operate at 1.8V or 3.3V.
iCLK	ICLK_ICOMP	2.5k Ohm ±1% to Ground	The calibration will be handled inside the iCLK.
	ICLK_RCOMP	50 Ohm ±1% to Ground	
USB2	USB_RCOMP	113 Ohm ±1% to Ground	The calibration will be handled inside USB
HSIC	USB_HSIC_RCOMP	45 Ohm ±1% to Ground	The calibration is handled inside the USB HSIC.
SSIC	USB_SSIC_RCOMP_P	90 Ohm ±1% Between SSIC RCOMP pads	The calibration is handled inside the USB SSIC.
	USB_SSIC_RCOMP_N		
USB3	USB3_RCOMP_N	402 Ohm 1% between RCOMP pads	The calibration is handled inside the USB3.
	USB3_RCOMP_P		
GPIO	GPIO0_RCOMP	100 Ohm to Ground	Will be shared across all GPIO buffers on the north side of the chip.
PCIE	PCIE_RCOMP_N	402 Ohm 1% between RCOMP pads	The Calibration is handled in PCIE.
	PCIE_RCOMP_P		

Table 34. RCOMP's List (Sheet 2 of 2)

Interface Name	RCOMP Name	Bias	Remarks
DDI	DDI0_RCOMP_N	402 Ohm 1% between RCOMP pads	The calibration is handled in DDI
	DDI0_RCOMP_P		
	DDI1_RCOMP_N	402 Ohm 1% between RCOMP pads	
	DDI1_RCOMP_P		

2.26 GPIO Muxing

Not all interfaces can be active at the same time. To provide flexibility, these shared interfaces are muxed with GPIOs.

Note: All GPIOs default to function as GPIO name at boot. BIOS is responsible for enabling proper configuration.

GPIO Number= GPIO pin location

GPIO mode= GPIO mode in which the pin operates

Table 35. Multiplexed Functions - T4 SoC (Sheet 1 of 11)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
ISH_GPIO[8]/ ISH_SPI_CS[0]_N/ I2S5_CLK	BL9	E23		ISH_GPIO[8]	ISH_SPI_CS[0]_N	I2S5_CLK			
LPC_AD[2]/ ISH_GPIO[14]/ ISH_I2C0_DATA	BP20	SE45		LPC_AD[2]	ISH_GPIO[14]	ISH_I2C0_DATA			
I2C4_CLK/ DDI0_DDC_CLK/ DDI2_DDC_CLK/ MDSI_DDC_CLK	BP34	SW52		I2C4_CLK	DDI0_DDC_CLK	DDI2_DDC_CLK	MDSI_DDC_CLK		
LPC_AD[3]/ ISH_GPIO[15]/ ISH_I2C0_CLK/ SPI2_MOSI	BR21	SE50		LPC_AD[3]	ISH_GPIO[15]	ISH_I2C0_CLK	SPI2_MOSI		
PMC_PLT_CLK[4]/ ISH_GPIO[14]/ ISH_I2C0_DATA/ SPI2_MISO	BR7	SE3		PMC_PLT_CLK[4]	ISH_GPIO[14]	ISH_I2C0_DATA	SPI2_MISO		
PMC_PLT_CLK[1]/ ISH_GPIO[11]/ ISH_UART_DATAIN /SPI2_CS[1]_N	BR9	SE2		PMC_PLT_CLK[1]	ISH_GPIO[11]	ISH_UART_DATAIN	SPI2_CS[1]_N		

Table 35. Multiplexed Functions - T4 SoC (Sheet 2 of 11)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
I2C4_DATA/DDI2_DDC_DATA/DDI2_DDC_DATA/MDSI_DDC_DATA	BT32	SW46		I2C4_DATA		DDI2_DDC_DATA	MDSI_DDC_DATA		
PMC_PLT_CLK[5]/ISH_GPIO[15]/ISH_I2C0_CLK/SPI2_MOSI	BT6	SE6		PMC_PLT_CLK[5]	ISH_GPIO[15]	ISH_I2C0_CLK	SPI2_MOSI		
MMC1_RCLK/MMC1_RESET_N	BU13	SE69		MMC1_RCLK	MMC1_RESET_N				
PMC_PLT_CLK[2]/ISH_GPIO[12]/ISH_UART_CTS_N/SPI2_CS[0]_N	BU7	SE7		PMC_PLT_CLK[2]	ISH_GPIO[12]	ISH_UART_CTS_N	SPI2_CS[0]_N		
PMC_PLT_CLK[3]/ISH_GPIO[13]/ISH_UART_RTS_N/SPI2_CLK	BU9	SE4		PMC_PLT_CLK[3]	ISH_GPIO[13]	ISH_UART_RTS_N	SPI2_CLK		
DDI2_DDC_CLK/DDI1_DDC_CLK/UART0_DATAOUT/MDSI_DDC_CLK/MDSI_A_TE	E21	N67		DDI2_DDC_CLK	DDI1_DDC_CLK	UART0_DATAOUT	MDSI_DDC_CLK	MDSI_A_TE	
GPIO_N1/C0_BPM3_TX/C1_BPM3_TX	E39	N1						C0_BPM3_TX	C1_BPM3_TX
DDI2_DDC_DATA/DDI1_DDC_DATA/UART0_DATAIN/MDSI_DDC_DATA/MDSI_C_TE	F20	N62		DDI2_DDC_DATA	DDI1_DDC_DATA	UART0_DATAIN	MDSI_DDC_DATA	MDSI_C_TE	
DDI0_DDC_CLK/DDI1_DDC_CLK/MDSI_DDC_CLK	F26	N71		DDI0_DDC_CLK	DDI1_DDC_CLK	MDSI_DDC_CLK			
GPIO_N2/C0_BPM2_TX/C1_BPM2_TX	F38	N2						C0_BPM2_TX	C1_BPM2_TX
GPIO_N4/C0_BPM0_TX/C1_BPM0_TX	G39	N4						C0_BPM0_TX	C1_BPM0_TX
DDI0_DDC_DATA/DDI1_DDC_DATA/MDSI_DDC_DATA	H26	N66		DDI0_DDC_DATA	DDI1_DDC_DATA	MDSI_DDC_DATA			
JTAG2_TMS	J37	N24		JTAG2_TMS					

Table 35. Multiplexed Functions - T4 SoC (Sheet 3 of 11)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
GPIO_N6/C0_BPM3_TX/C1_BPM3_TX	J39	N6						C0_BPM3_TX	C1_BPM3_TX
GPIO_N8/C0_BPM1_TX/C1_BPM1_TX	K40	N8						C0_BPM1_TX	C1_BPM1_TX
GPIO_N3/C0_BPM1_TX/C1_BPM1_TX	B38	N3						C0_BPM1_TX	C1_BPM1_TX
ISH_GPIO[13]/C0_BPM2_TX/C1_BPM2_TX	C39	N7					ISH_GPIO[13]	C0_BPM2_TX	C1_BPM2_TX
GPIO_N0/C0_BPM0_TX/C1_BPM0_TX	D40	N0						C0_BPM0_TX	C1_BPM0_TX
GPIO_CAMERASB03	B28	N51		GPIO_CAMERASB03					
JTAG_TMS	B34	N34		JTAG_TMS					
PMC_PWRBTN_N	BH10	E8		PMC_PWRBTN_N					
SD3_D[2]	BH18	SE33		SD3_D[2]					
PMC_RSTBTN_N	BH24	SE76		PMC_RSTBTN_N					
UART2_RTS_N	BH26	SW19		UART2_RTS_N					
UART2_DATAIN	BH28	SW17		UART2_DATAIN					
LPE_I2S0_CLK	BH32	SW31		LPE_I2S0_CLK					
I2C6_CLK/NMI_N	BH34	SW53		I2C6_CLK	NMI_N				
I2C2_DATA	BH36	SW62		I2C2_DATA					
PMC_BATLOW_N	BH4	E1		PMC_BATLOW_N					
LPE_I2S2_FRM	BH40	SW96		LPE_I2S2_FRM					
PMC_SUS_STAT_N	BH6	E2		PMC_SUS_STAT_N					

Table 35. Multiplexed Functions - T4 SoC (Sheet 4 of 11)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
MMC1_CMD	BJ15	SE23		MMC1_CMD					
LPC_FRAME_N/UART0_DATAIN/SPI2_MISO	BJ19	SE48		LPC_FRAME_N	UART0_DATAIN		SPI2_MISO		
GPIO_ALERT/ISH_GPIO[11]/ISH_UART_DATAIN	BJ21	SE77		GPIO_ALERT	ISH_GPIO[11]	ISH_UART_DATAIN			
FST_SPI_D[2]	BJ25	SW0		FST_SPI_D[2]					
PMC_SLP_S3_N	BJ3	E0		PMC_SLP_S3_N					
LPE_I2S1_DATAIN	BJ30	SW37		LPE_I2S1_DATAIN					
NFC_I2C_CLK	BJ33	SW54		NFC_I2C_CLK					
UART0_DATAIN	BJ37	SW77			UART0_DATAIN				
PMC_PLTRST_N	BJ5	E5		PMC_PLTRST_N					
PMC_WAKE_N	BJ7	E10		PMC_WAKE_N					
PMC_SLP_S4_N	BJ9	E9		PMC_SLP_S4_N					
ISH_GPIO[6]/I2S4_DATAOUT	BK10	E25		ISH_GPIO[6]		I2S4_DATAOUT			
MMC1_D[3]	BK12	SE26		MMC1_D[3]					
MMC1_D[1]	BK14	SE24		MMC1_D[1]					
SD3_D[0]	BK16	SE35		SD3_D[0]					
SPI1_MOSI	BK18	SE64		SPI1_MOSI					
LPC_CLKOUT[0]/ISH_GPIO[10]/ISH_UART_DATAOUT	BK20	SE51		LPC_CLKOUT[0]	ISH_GPIO[10]		ISH_UART_DATAOUT		
PMC_SUSPWRDNACK	BK22	SE83		PMC_SUSPWRDNACK					

Table 35. Multiplexed Functions - T4 SoC (Sheet 5 of 11)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
FST_SPI_D[1]	BK26	SW5		FST_SPI_D[1]					
UART2_DATAOUT	BK28	SW21		UART2_DATAOUT					
LPE_I2S0_FRM	BK32	SW35		LPE_I2S0_FRM					
I2C5_CLK	BK34	SW50		I2C5_CLK					
SPI3_MOSI	BK36	SW82			SPI3_MOSI				
LPE_I2S2_CLK	BK38	SW92		LPE_I2S2_CLK					
PMC_ACPRESENT	BK4	E4		PMC_ACPRESENT					
LPE_I2S2_DATAOUT	BK40	SW97		LPE_I2S2_DATAOUT					
ISH_GPIO[9]/ISH_SPI_MISO/I2S5_FS	BK8	E20		ISH_GPIO[9]	ISH_SPI_MISO	I2S5_FS			
SD2_CLK	BL11	SE19		SD2_CLK					
SD3_D[3]	BL15	SE32		SD3_D[3]					
SD3_CLK	BL17	SE31		SD3_CLK					
SPI1_MISO	BL19	SE60		SPI1_MISO					
LPC_CLKRUN_N/UART0_DATAOUT/SPI2_CLK	BL21	SE46		LPC_CLKRUN_N	UART0_DATAOUT		SPI2_CLK		
FST_SPI_D[3]	BL25	SW3		FST_SPI_D[3]					
UART2_CTS_N	BL27	SW22		UART2_CTS_N					
PMC_SLP_S0IX_N	BL3	E3		PMC_SLP_S0IX_N					
I2C6_DATA/SD3_WP	BL33	SW49		I2C6_DATA	SD3_WP				
I2C2_CLK	BL35	SW66		I2C2_CLK					
UART0_DATAOUT/SPI3_CLK	BL37	SW79			SPI3_CLK	UART0_DATAOUT			

Table 35. Multiplexed Functions - T4 SoC (Sheet 6 of 11)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
LPE_I2S2_DATAIN	BL39	SW94		LPE_I2S2_DATAIN					
MMC1_D[0]	BM12	SE17		MMC1_D[0]					
MMC1_D[2]	BM14	SE20		MMC1_D[2]					
ISH_GPIO[7]/ I2S4_DATAIN	BM2	E16		ISH_GPIO[7]		I2S4_DATAIN			
LPC_CLKOUT[1]/ ISH_GPIO[11]/ ISH_UART_DATAIN	BM20	SE49		LPC_CLKOUT[1]	ISH_GPIO[11]		ISH_UART_DATAIN		
LPC_SERIRQ/ SPI2_CS[0]_N	BM24	SE79		LPC_SERIRQ			SPI2_CS[0]_N		
LPE_I2S0_DATAOUT	BM32	SW30		LPE_I2S0_DATAOUT					
SPI3_CS[0]_N	BM38	SW76				SPI3_CS[0]_N			
ISH_GPIO[3]/ I2S3_DATAIN	BM4	E15		ISH_GPIO[3]		I2S3_DATAIN			
PMC_SUSCLK[0]	BM6	E6		PMC_SUSCLK[0]					
ISH_I2C1_DATA/ ISH_SPI_MOSI/ I2S5_DATAOUT	BM8	E26		ISH_I2C1_DATA	ISH_SPI_MOSI	I2S5_DATAOUT			
SD2_CMD	BN11	SE22		SD2_CMD					
MMC1_CLK	BN15	SE16		MMC1_CLK					
SPI1_CLK	BN19	SE62		SPI1_CLK					
FST_SPI_D[0]	BN25	SW1		FST_SPI_D[0]					
ISH_I2C1_CLK/ ISH_SPI_CLK/ I2S5_DATAIN	BN3	E17		ISH_I2C1_CLK	ISH_SPI_CLK	I2S5_DATAIN			
NFC_I2C_DATA	BN33	SW51		NFC_I2C_DATA					
SPI3_MISO	BN37	SW81			SPI3_MISO				
ISH_GPIO[1]/ I2S3_FS	BN5	E18		ISH_GPIO[1]		I2S3_FS			
SD2_D[3]_CD_N	BP12	SE15		SD2_D[3]_CD_N					
MMC1_D[6]	BP14	SE63		MMC1_D[6]					

Table 35. Multiplexed Functions - T4 SoC (Sheet 7 of 11)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
SD3_D[1]	BP16	SE30		SD3_D[1]					
USB_OC[0]_N	BP22	SE80		USB_OC[0]_N					
FST_SPI_CLK	BP24	SW2		FST_SPI_CLK					
LPE_I2S1_DATAOUT	BP28	SW34		LPE_I2S1_DATAOUT					
I2C1_CLK	BP36	SW63		I2C1_CLK					
GPIO_SW93	BP38	SW93							
ISH_GPIO[5]/I2S4_FS	BP4	E19		ISH_GPIO[5]		I2S4_FS			
PMC_PLT_CLK[0]/ISH_GPIO[10]/ISH_UART_DATAOUT	BP8	SE0		PMC_PLT_CLK[0]	ISH_GPIO[10]	ISH_UART_DATAOUT			
SD2_D[1]	BR11	SE18		SD2_D[1]					
MMC1_D[4]	BR13	SE67		MMC1_D[4]					
SD3_CMD	BR15	SE34		SD3_CMD					
SPI1_CS[1]_N	BR17	SE66		SPI1_CS[1]_N					
LPC_AD[1]/ISH_GPIO[13]/ISH_UART_RTS_N	BR19	SE52		LPC_AD[1]	ISH_GPIO[13]		ISH_UART_RTS_N		
FST_SPI_CS[1]_N	BR23	SW4		FST_SPI_CS[1]_N					
UART1_RTS_N	BR25	SW15		UART1_RTS_N					
UART1_CTS_N	BR27	SW18		UART1_CTS_N					
ISH_GPIO[0]/I2S3_CLK	BR3	E21		ISH_GPIO[0]		I2S3_CLK			
LPE_I2S1_CLK	BR30	SW32		LPE_I2S1_CLK					
I2C5_DATA	BR33	SW45		I2C5_DATA					
I2C1_DATA	BR35	SW60		I2C1_DATA					

Table 35. Multiplexed Functions - T4 SoC (Sheet 8 of 11)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
ISH_GPIO[12]/ ISH_UART_CTS_N	BR37	SW75			ISH_GPIO[12]	ISH_UART_CTS_N			
PCIE_CLKREQ[0]_N	BR39	SW90		PCIE_CLKREQ[0]_N					
SD2_D[0]	BT10	SE25		SD2_D[0]					
MMC1_D[7]	BT14	SE68		MMC1_D[7]					
SPI1_CS[0]_N	BT18	SE61		SPI1_CS[0]_N					
ISH_GPIO[2]/ I2S3_DATAOUT	BT2	E24		ISH_GPIO[2]		I2S3_DATAOUT			
SD3_1P8_EN	BT22	SE85		SD3_1P8_EN					
UART1_DATAIN/ UART0_DATAIN	BT26	SW16		UART1_DATAIN	UART0_DATAIN				
I2C0_CLK	BT36	SW65		I2C0_CLK					
ISH_GPIO[4]/ I2S4_CLK	BT4	E22		ISH_GPIO[4]		I2S4_CLK			
SD3_WP	BT40	SW95			SD3_WP				
SD2_D[2]	BU11	SE21		SD2_D[2]					
MMC1_D[5]	BU15	SE65		MMC1_D[5]					
SD3_CD_N	BU17	SE81		SD3_CD_N					
LPC_AD[0]/ ISH_GPIO[12]/ ISH_UART_CTS_N	BU19	SE47		LPC_AD[0]	ISH_GPIO[12]		ISH_UART_CTS_N		
USB_OC[1]_N	BU21	SE75		USB_OC[1]_N					
SD3_PWREN_N	BU23	SE78		SD3_PWREN_N					
FST_SPI_CS[0]_N	BU25	SW6		FST_SPI_CS[0]_N					
UART1_DATAOUT/ UART0_DATAOUT	BU27	SW20		UART1_DATAOUT	UART0_DATAOUT				
LPE_I2S0_DATAIN	BU30	SW33		LPE_I2S0_DATAIN					
I2C3_DATA	BU33	SW64		I2C3_DATA					

Table 35. Multiplexed Functions - T4 SoC (Sheet 9 of 11)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
I2C0_DATA	BU35	SW61		I2C0_DATA					
GPIO_SW78	BU37	SW78							
PCIE_CLKREQ[1]_N	BU39	SW91		PCIE_CLKREQ[1]_N					
PWM[1]/ ISH_GPIO[10]/ ISH_UART_DATAOUT	BU5	SE1		PWM[1]	ISH_GPIO[10]	ISH_UART_DATAOUT			
FST_SPI_CS[2]_N	BV24	SW7		FST_SPI_CS[2]_N					
LPE_I2S1_FRM	BV28	SW36		LPE_I2S1_FRM					
I2C3_CLK	BV34	SW67		I2C3_CLK					
MMC1_RESET_N/ SPI3_CS[1]_N	BV38	SW80			MMC1_RESET_N	SPI3_CS[1]_N			
PWM[0]	BV4	SE5		PWM[0]					
DDI2_HPD	C21	N68		DDI2_HPD					
GPIO_CAMERASB07	C27	N54		GPIO_CAMERASB07					
GPIO_CAMERASB04	C30	N56		GPIO_CAMERASB04					
SVID_ALERT_N	C33	N38		SVID_ALERT_N					
GPIO_SUS5/ PMC_SUSCLK[1]	C35	N20		PMC_SUSCLK[1]					
GPIO_SUS3/ JTAG2_TDI	C37	N17		JTAG2_TDI					
GPIO_CAMERASB10	D26	N50		GPIO_CAMERASB10					
SVID_DATA	D32	N33	SVID_DATA						
GPIO_SUS6/ PMC_SUSCLK[2]	D36	N25		PMC_SUSCLK[2]					
DDI1_HPD	E25	N64		DDI1_HPD					
GPIO_CAMERASB06	E27	N49		GPIO_CAMERASB06					
GPIO_CAMERASB02	E30	N46		GPIO_CAMERASB02					
JTAG_TDI	E33	N41		JTAG_TDI					

Table 35. Multiplexed Functions - T4 SoC (Sheet 10 of 11)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
JTAG_TRST_N	E35	N30		JTAG_TRST_N					
GPIO_SUS0	E37	N15							
DDI0_VDDEN	F22	N72		DDI0_VDDEN					
GPIO_CAMERASB11	F28	N55		GPIO_CAMERASB11					
SVID_CLK	F32	N40		SVID_CLK					
JTAG_TCK	F34	N31		JTAG_TCK					
GPIO_SUS8	F36	N23		GPIO_SUS8					
GPIO_CAMERASB05	G27	N45		GPIO_CAMERASB05					
JTAG_PRDY_N	G33	N37		JTAG_PRDY_N					
GPIO_SUS4/JTAG2_TDO	G37	N22		JTAG2_TDO					
DDI1_VDDEN/MDSI_DDC_DATA	H22	N69		DDI1_VDDEN	MDSI_DDC_DATA				
PROCHOT_N	H32	N32		PROCHOT_N					
JTAG_PREQ_N	H34	N26		JTAG_PREQ_N					
GPIO_DFX4	H38	N5							
DDI1_BKLTEN/MDSI_DDC_CLK	J21	N70		DDI1_BKLTEN	MDSI_DDC_CLK				
GPIO_CAMERASB09	J27	N52		GPIO_CAMERASB09					
GPIO_CAMERASB00	J30	N48		GPIO_CAMERASB00					
JTAG_TDO	J33	N39		JTAG_TDO					
GPIO_SUS9	J35	N27		GPIO_SUS9					
DDI1_BKLTCTL/MDSI_A_TE/MDSI_C_TE	K20	N63		DDI1_BKLTCTL		MDSI_A_TE	MDSI_C_TE		
DDI0_BKLTCTL	K22	N65		DDI0_BKLTCTL					
DDI0_HPD	K26	N61		DDI0_HPD					
GPIO_CAMERASB08	K28	N47		GPIO_CAMERASB08					

Table 35. Multiplexed Functions - T4 SoC (Sheet 11 of 11)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
GPIO_CAMERASB01	K32	N53		GPIO_CAMERASB01					
GPIO_SUS10	K34	N16		GPIO_SUS10					
GPIO_SUS7/ PMC_SUSCLK[3]	K36	N18		PMC_SUSCLK[3]					
GPIO_SUS1/ JTAG2_TCK	K38	N19		JTAG2_TCK					
DDI0_BKLTEN	L21	N60		DDI0_BKLTEN					

Table 36. Multiplexed Functions - T3 SoC (Sheet 1 of 9)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
DDI0_DDC_CLK/ DDI1_DDC_CLK/ MDSI_DDC_CLK	A10	N71		DDI0_DDC_CLK	DDI1_DDC_CLK	MDSI_DDC_CLK			
GPIO_SE79	AA13	SE79							
I2C4_CLK/ DDI1_DDC_CLK/ DDI2_DDC_CLK/ MDSI_DDC_CLK	AB17	SW52		I2C4_CLK	DDI1_DDC_CLK	DDI2_DDC_CLK	MDSI_DDC_CLK		
I2C4_DATA/ DDI1_DDC_DATA/ DDI2_DDC_DATA/ MDSI_DDC_DATA	AB18	SW46		I2C4_DATA	DDI1_DDC_DATA	DDI2_DDC_DATA	MDSI_DDC_DATA		
PMC_PLT_CLK[3]/ ISH_GPIO[13]/ ISH_UART_RTS_N/ SPI2_CLK	AB4	SE4		PMC_PLT_CLK[3]	ISH_GPIO[13]	ISH_UART_RTS_N	SPI2_CLK		
PMC_PLT_CLK[4]/ ISH_GPIO[14]/ ISH_I2C0_DATA/ SPI2_MISO	AC3	SE3		PMC_PLT_CLK[4]	ISH_GPIO[14]	ISH_I2C0_DATA	SPI2_MISO		
PMC_PLT_CLK[2]/ ISH_GPIO[12]/ ISH_UART_CTS_N/ SPI2_CS[0]_N	AC4	SE7		PMC_PLT_CLK[2]	ISH_GPIO[12]	ISH_UART_CTS_N	SPI2_CS[0]_N		
PMC_PLT_CLK[5]/ ISH_GPIO[15]/ ISH_I2C0_CLK/ SPI2_MOSI	AE3	SE6		PMC_PLT_CLK[5]	ISH_GPIO[15]	ISH_I2C0_CLK	SPI2_MOSI		

Table 36. Multiplexed Functions - T3 SoC (Sheet 2 of 9)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
MMC1_RCLK/MMC1_RESET_N	AE8	SE69		MMC1_RCLK	MMC1_RESET_N				
PMC_SUSCLK[3]	B12	N18		PMC_SUSCLK[3]					
JTAG2_TDI	B14	N17		JTAG2_TDI					
DDI0_DDC_DATA/DDI1_DDC_DATA/MDSI_DDC_DATA	C10	N66		DDI0_DDC_DATA	DDI1_DDC_DATA	MDSI_DDC_DATA			
PMC_SUSCLK[2]	C14	N25		PMC_SUSCLK[2]					
JTAG2_TCK	C15	N19		JTAG2_TCK					
GPIO_N3/C0_BPM1_TX/C1_BPM1_TX	C17	N3						C0_BPM1_TX	C1_BPM1_TX
DDI2_DDC_DATA/DDI1_DDC_DATA/UART0_DATAIN/MDSI_DDC_DATA/MDSI_C_TE	C9	N62		DDI2_DDC_DATA	DDI1_DDC_DATA	UART0_DATAIN	MDSI_DDC_DATA	MDSI_C_TE	
JTAG2_TDO	D14	N22		JTAG2_TDO					
JTAG2_TMS	D15	N24		JTAG2_TMS					
DDI1_BKLTCTL/MDSI_A_TE/MDSI_C_TE	D8	N63		DDI1_BKLTCTL		MDSI_A_TE	MDSI_C_TE		
DDI2_DDC_CLK/DDI1_DDC_CLK/UART0_DATAOUT/MDSI_DDC_CLK/MDSI_A_TE	E10	N67		DDI2_DDC_CLK	DDI1_DDC_CLK	UART0_DATAOUT	MDSI_DDC_CLK	MDSI_A_TE	
GPIO_N4/C0_BPM0_TX/C1_BPM0_TX	E16	N4						C0_BPM0_TX	C1_BPM0_TX
GPIO_N0/C0_BPM0_TX/C1_BPM0_TX	E17	N0						C0_BPM0_TX	C1_BPM0_TX
GPIO_N2/C0_BPM2_TX/C1_BPM2_TX	F16	N2						C0_BPM2_TX	C1_BPM2_TX

Table 36. Multiplexed Functions - T3 SoC (Sheet 3 of 9)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
GPIO_N1/C0_BPM3_TX/C1_BPM3_TX	F17	N1						C0_BPM3_TX	C1_BPM3_TX
UART0_DATAIN	V13	SE48			UART0_DATAIN				
PMC_SLP_S0IX_N	W4	E3		PMC_SLP_S0IX_N					
UART0_DATAOUT	Y12	SE46			UART0_DATAOUT				
GPIO_N6/C0_BPM3_TX/C1_BPM3_TX	C16	N6						C0_BPM3_TX	C1_BPM3_TX
GPIO_N8/C0_BPM1_TX/C1_BPM1_TX	D16	N8						C0_BPM1_TX	C1_BPM1_TX
ISH_GPIO[13]/C0_BPM2_TX/C1_BPM2_TX	D17	N7					ISH_GPIO[13]	C0_BPM2_TX	C1_BPM2_TX
GPIO_SUS9	A13	N27		GPIO_SUS9					
GPIO_SUS0	A14	N15							
DDI0_VDDEN	A9	N72		DDI0_VDDEN					
SD3_CMD	AA10	SE34		SD3_CMD					
FST_SPI_CS[0]_N	AA12	SW6		FST_SPI_CS[0]_N					
LPE_I2S1_FRM	AA14	SW36		LPE_I2S1_FRM					
LPE_I2S0_DATAOUT	AA15	SW30		LPE_I2S0_DATAOUT					
LPE_I2S0_CLK	AA16	SW31		LPE_I2S0_CLK					
I2C2_CLK	AA17	SW66		I2C2_CLK					
ISH_I2C1_CLK/ISH_SPI_CLK/I2S5_DATAIN	AA3	E17		ISH_I2C1_CLK	ISH_SPI_CLK	I2S5_DATAIN			

Table 36. Multiplexed Functions - T3 SoC (Sheet 4 of 9)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
ISH_GPIO[9]/ISH_SPI_MISO/I2S5_FS	AA4	E20		ISH_GPIO[9]	ISH_SPI_MISO	I2S5_FS			
ISH_I2C1_DATA/ISH_SPI_MOSI/I2S5_DATAOUT	AA5	E26		ISH_I2C1_DATA	ISH_SPI_MOSI	I2S5_DATAOUT			
MMC1_CMD	AA6	SE23		MMC1_CMD					
MMC1_D[7]	AA7	SE68		MMC1_D[7]					
MMC1_D[0]	AA8	SE17		MMC1_D[0]					
SD3_PWREN_N	AA9	SE78		SD3_PWREN_N					
SD3_D[2]	AB10	SE33		SD3_D[2]					
FST_SPI_D[0]	AB11	SW1		FST_SPI_D[0]					
FST_SPI_CLK	AB12	SW2		FST_SPI_CLK					
UART2_DATAOUT	AB13	SW21		UART2_DATAOUT					
LPE_I2S1_CLK	AB14	SW32		LPE_I2S1_CLK					
LPE_I2S0_DATAIN	AB15	SW33		LPE_I2S0_DATAIN					
I2C5_CLK	AB16	SW50		I2C5_CLK					
LPE_I2S2_CLK	AB19	SW92		LPE_I2S2_CLK					
ISH_GPIO[0]/I2S3_CLK	AB2	E21		ISH_GPIO[0]		I2S3_CLK			
PCIE_CLKREQ[0]_N	AB20	SW90		PCIE_CLKREQ[0]_N					
ISH_GPIO[2]/I2S3_DATAOUT	AB3	E24		ISH_GPIO[2]		I2S3_DATAOUT			
SD2_CMD	AB5	SE22		SD2_CMD					
MMC1_D[2]	AB6	SE20		MMC1_D[2]					

Table 36. Multiplexed Functions - T3 SoC (Sheet 5 of 9)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
MMC1_D[6]	AB7	SE63		MMC1_D[6]					
SD3_1P8_EN	AB8	SE85		SD3_1P8_EN					
SD3_D[1]	AB9	SE30		SD3_D[1]					
ISH_GPIO[7]/ I2S4_DATAIN	AC1	E16		ISH_GPIO[7]		I2S4_DATAIN			
SD3_D[0]	AC10	SE35		SD3_D[0]					
FST_SPI_D[1]	AC11	SW5		FST_SPI_D[1]					
UART2_RTS_N	AC12	SW19		UART2_RTS_N					
UART1_CTS_N	AC13	SW18		UART1_CTS_N					
UART1_DATAOUT/ UART0_DATAOUT	AC14	SW20		UART1_DATAOUT	UART0_DATAOUT				
NFC_I2C_DATA	AC15	SW51		NFC_I2C_DATA					
NFC_I2C_CLK	AC16	SW54		NFC_I2C_CLK					
I2C2_DATA	AC17	SW62		I2C2_DATA					
GPIO_SW78	AC18	SW78							
LPE_I2S2_FRM	AC19	SW96		LPE_I2S2_FRM					
ISH_GPIO[4]/ I2S4_CLK	AC2	E22		ISH_GPIO[4]		I2S4_CLK			
GPIO_SW93	AC20	SW93							
MMC1_D[3]	AC6	SE26		MMC1_D[3]					
MMC1_D[4]	AC7	SE67		MMC1_D[4]					

Table 36. Multiplexed Functions - T3 SoC (Sheet 6 of 9)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
MMC1_D[5]	AC8	SE65		MMC1_D[5]					
SD3_CLK	AC9	SE31		SD3_CLK					
SD3_D[3]	AD10	SE32		SD3_D[3]					
UART2_CTS_N	AD12	SW22		UART2_CTS_N					
UART1_DATAIN/ UART0_DATAIN	AD14	SW16		UART1_DATAIN	UART0_DATAIN				
I2C6_DATA/ SD3_WP	AD16	SW49		I2C6_DATA	SD3_WP				
MMC1_RESET_N	AD18	SW80			MMC1_RESET_N				
PWM[1]/ ISH_GPIO[10]/ ISH_UART_DATAOUT	AD2	SE1		PWM[1]	ISH_GPIO[10]	ISH_UART_DATAOUT			
SD3_WP	AD20	SW95			SD3_WP				
PWM[0]	AD3	SE5		PWM[0]					
SD2_D[0]	AD4	SE25		SD2_D[0]					
SD2_CLK	AD5	SE19		SD2_CLK					
SD2_D[3]_CD_N	AD6	SE15		SD2_D[3]_CD_N					
MMC1_CLK	AD8	SE16		MMC1_CLK					
UART2_DATAIN	AE12	SW17		UART2_DATAIN					
UART1_RTS_N	AE13	SW15		UART1_RTS_N					
I2C6_CLK/NMI_N	AE16	SW53		I2C6_CLK	NMI_N				
I2C5_DATA	AE17	SW45		I2C5_DATA					
UART0_DATAIN	AE18	SW77			UART0_DATAIN				

Table 36. Multiplexed Functions - T3 SoC (Sheet 7 of 9)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
SD2_D[1]	AE4	SE18		SD2_D[1]					
SD2_D[2]	AE5	SE21		SD2_D[2]					
SD3_CD_N	AE9	SE81		SD3_CD_N					
GPIO_CAMERASB10	B10	N50		GPIO_CAMERASB10					
DDI0_BKLTCTL	B8	N65		DDI0_BKLTCTL					
GPIO_CAMERASB11	C11	N55		GPIO_CAMERASB11					
JTAG_TDI	C12	N41		JTAG_TDI					
GPIO_SUS8	C13	N23		GPIO_SUS8					
DDI0_BKLTEN	C8	N60		DDI0_BKLTEN					
GPIO_CAMERASB09	D10	N52		GPIO_CAMERASB09					
GPIO_CAMERASB08	D11	N47		GPIO_CAMERASB08					
DDI0_HPD	D9	N61		DDI0_HPD					
SVID_ALERT_N	E12	N38		SVID_ALERT_N					
JTAG_TRST_N	E13	N30		JTAG_TRST_N					
JTAG_TCK	E14	N31		JTAG_TCK					
DDI2_HPD	E9	N68		DDI2_HPD					
SVID_CLK	F11	N40		SVID_CLK					
SVID_DATA	F12	N33	SVID_DATA						
JTAG_TDO	F13	N39		JTAG_TDO					

Table 36. Multiplexed Functions - T3 SoC (Sheet 8 of 9)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
JTAG_TMS	F14	N34		JTAG_TMS					
GPIO_DFX4	F18	N5							
PROCHOT_N	F9	N32		PROCHOT_N					
LPE_I2S2_DATAIN	U17	SW94		LPE_I2S2_DATAIN					
PMC_SUSPWRDNACK	V12	SE83		PMC_SUSPWRDNACK					
LPE_I2S2_DATAOUT	V18	SW97		LPE_I2S2_DATAOUT					
PMC_SUS_STAT_N	V5	E2		PMC_SUS_STAT_N					
I2C1_DATA	W15	SW60		I2C1_DATA					
I2C1_CLK	W16	SW63		I2C1_CLK					
PMC_PWRBTN_N	W3	E8		PMC_PWRBTN_N					
ISH_GPIO[3]/ I2S3_DATAIN	Y1	E15		ISH_GPIO[3]		I2S3_DATAIN			
LPE_I2S1_DATAIN	Y13	SW37		LPE_I2S1_DATAIN					
LPE_I2S1_DATAOUT	Y14	SW34		LPE_I2S1_DATAOUT					
LPE_I2S0_FRM	Y16	SW35		LPE_I2S0_FRM					
I2C0_CLK	Y17	SW65		I2C0_CLK					

Table 36. Multiplexed Functions - T3 SoC (Sheet 9 of 9)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
I2C0_DATA	Y18	SW61		I2C0_DATA					
ISH_GPIO[1]/ I2S3_FS	Y2	E18		ISH_GPIO[1]		I2S3_FS			
PMC_WAKE_N	Y3	E10		PMC_WAKE_N					
PMC_SUSCLK[0]	Y4	E6		PMC_SUSCLK[0]					
PMC_PLTRST_N	Y5	E5		PMC_PLTRST_N					
MMC1_D[1]	Y8	SE24		MMC1_D[1]					
USB_OC[0]_N	Y9	SE80		USB_OC[1]_N					

Table 37. Multiplexed Functions - T3 SoC (Sheet 1 of 9)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
DDI0_DDC_CLK/ DDI1_DDC_CLK/ MDSI_DDC_CLK	A10	N71		DDI0_DDC_CLK	DDI1_DDC_CLK	MDSI_DDC_CLK			
GPIO_SE79	AA13	SE79							
I2C4_CLK/ DDI1_DDC_CLK/ DDI2_DDC_CLK/ MDSI_DDC_CLK	AB17	SW52		I2C4_CLK	DDI1_DDC_CLK	DDI2_DDC_CLK	MDSI_DDC_CLK		
I2C4_DATA/ DDI1_DDC_DATA/ DDI2_DDC_DATA/ MDSI_DDC_DATA	AB18	SW46		I2C4_DATA	DDI1_DDC_DATA	DDI2_DDC_DATA	MDSI_DDC_DATA		
PMC_PLT_CLK[3]/ ISH_GPIO[13]/ ISH_UART_RTS_N/ SPI2_CLK	AB4	SE4		PMC_PLT_CLK[3]	ISH_GPIO[13]	ISH_UART_RTS_N	SPI2_CLK		

Table 37. Multiplexed Functions - T3 SoC (Sheet 2 of 9)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
PMC_PLT_CLK[4]/ISH_GPIO[14]/ISH_I2C0_DATA/SPI2_MISO	AC3	SE3		PMC_PLT_CLK[4]	ISH_GPIO[14]	ISH_I2C0_DATA	SPI2_MISO		
PMC_PLT_CLK[2]/ISH_GPIO[12]/ISH_UART_CTS_N/SPI2_CS[0]_N	AC4	SE7		PMC_PLT_CLK[2]	ISH_GPIO[12]	ISH_UART_CTS_N	SPI2_CS[0]_N		
PMC_PLT_CLK[5]/ISH_GPIO[15]/ISH_I2C0_CLK/SPI2_MOSI	AE3	SE6		PMC_PLT_CLK[5]	ISH_GPIO[15]	ISH_I2C0_CLK	SPI2_MOSI		
MMC1_RCLK/MMC1_RESET_N	AE8	SE69		MMC1_RCLK	MMC1_RESET_N				
PMC_SUSCLK[3]	B12	N18		PMC_SUSCLK[3]					
JTAG2_TDI	B14	N17		JTAG2_TDI					
DDI0_DDC_DATA/DDI1_DDC_DATA/MDSI_DDC_DATA	C10	N66		DDI0_DDC_DATA	DDI1_DDC_DATA	MDSI_DDC_DATA			
PMC_SUSCLK[2]	C14	N25		PMC_SUSCLK[2]					
JTAG2_TCK	C15	N19		JTAG2_TCK					
GPIO_N3/C0_BPM1_TX/C1_BPM1_TX	C17	N3						C0_BPM1_TX	C1_BPM1_TX
DDI2_DDC_DATA/DDI1_DDC_DATA/UART0_DATAIN/MDSI_DDC_DATA/MDSI_C_TE	C9	N62		DDI2_DDC_DATA	DDI1_DDC_DATA	UART0_DATAIN	MDSI_DDC_DATA	MDSI_C_TE	
JTAG2_TDO	D14	N22		JTAG2_TDO					
JTAG2_TMS	D15	N24		JTAG2_TMS					
DDI1_BKLTCTL/MDSI_A_TE/MDSI_C_TE	D8	N63		DDI1_BKLTCTL		MDSI_A_TE	MDSI_C_TE		

Table 37. Multiplexed Functions - T3 SoC (Sheet 3 of 9)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
DDI2_DDC_CLK/DDI1_DDC_CLK/UART0_DATAOUT/MDSI_DDC_CLK/MDSI_A_TE	E10	N67		DDI2_DDC_CLK	DDI1_DDC_CLK	UART0_DATAOUT	MDSI_DDC_CLK	MDSI_A_TE	
GPIO_N4/C0_BPM0_TX/C1_BPM0_TX	E16	N4						C0_BPM0_TX	C1_BPM0_TX
GPIO_N0/C0_BPM0_TX/C1_BPM0_TX	E17	N0						C0_BPM0_TX	C1_BPM0_TX
GPIO_N2/C0_BPM2_TX/C1_BPM2_TX	F16	N2						C0_BPM2_TX	C1_BPM2_TX
GPIO_N1/C0_BPM3_TX/C1_BPM3_TX	F17	N1						C0_BPM3_TX	C1_BPM3_TX
UART0_DATAIN	V13	SE48			UART0_DATAIN				
PMC_SLP_S0IX_N	W4	E3		PMC_SLP_S0IX_N					
UART0_DATAOUT	Y12	SE46			UART0_DATAOUT				
GPIO_N6/C0_BPM3_TX/C1_BPM3_TX	C16	N6						C0_BPM3_TX	C1_BPM3_TX
GPIO_N8/C0_BPM1_TX/C1_BPM1_TX	D16	N8						C0_BPM1_TX	C1_BPM1_TX
ISH_GPIO[13]/C0_BPM2_TX/C1_BPM2_TX	D17	N7					ISH_GPIO[13]	C0_BPM2_TX	C1_BPM2_TX
GPIO_SUS9	A13	N27		GPIO_SUS9					
GPIO_SUS0	A14	N15							
DDI0_VDDEN	A9	N72		DDI0_VDDEN					
SD3_CMD	AA10	SE34		SD3_CMD					
FST_SPI_CS[0]_N	AA12	SW6		FST_SPI_CS[0]_N					

Table 37. Multiplexed Functions - T3 SoC (Sheet 4 of 9)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
LPE_I2S1_FRM	AA14	SW36		LPE_I2S1_FRM					
LPE_I2S0_DATAOUT	AA15	SW30		LPE_I2S0_DATAOUT					
LPE_I2S0_CLK	AA16	SW31		LPE_I2S0_CLK					
I2C2_CLK	AA17	SW66		I2C2_CLK					
ISH_I2C1_CLK/ISH_SPI_CLK/I2S5_DATAIN	AA3	E17		ISH_I2C1_CLK	ISH_SPI_CLK	I2S5_DATAIN			
ISH_GPIO[9]/ISH_SPI_MISO/I2S5_FS	AA4	E20		ISH_GPIO[9]	ISH_SPI_MISO	I2S5_FS			
ISH_I2C1_DATA/ISH_SPI_MOSI/I2S5_DATAOUT	AA5	E26		ISH_I2C1_DATA	ISH_SPI_MOSI	I2S5_DATAOUT			
MMC1_CMD	AA6	SE23		MMC1_CMD					
MMC1_D[7]	AA7	SE68		MMC1_D[7]					
MMC1_D[0]	AA8	SE17		MMC1_D[0]					
SD3_PWREN_N	AA9	SE78		SD3_PWREN_N					
SD3_D[2]	AB10	SE33		SD3_D[2]					
FST_SPI_D[0]	AB11	SW1		FST_SPI_D[0]					
FST_SPI_CLK	AB12	SW2		FST_SPI_CLK					
UART2_DATAOUT	AB13	SW21		UART2_DATAOUT					
LPE_I2S1_CLK	AB14	SW32		LPE_I2S1_CLK					
LPE_I2S0_DATAIN	AB15	SW33		LPE_I2S0_DATAIN					
I2C5_CLK	AB16	SW50		I2C5_CLK					

Table 37. Multiplexed Functions - T3 SoC (Sheet 5 of 9)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
LPE_I2S2_CLK	AB19	SW92		LPE_I2S2_CLK					
ISH_GPIO[0]/ I2S3_CLK	AB2	E21		ISH_GPIO[0]		I2S3_CLK			
PCIE_CLKREQ[0]_N	AB20	SW90		PCIE_CLKREQ[0]_N					
ISH_GPIO[2]/ I2S3_DATAOUT	AB3	E24		ISH_GPIO[2]		I2S3_DATAOUT			
SD2_CMD	AB5	SE22		SD2_CMD					
MMC1_D[2]	AB6	SE20		MMC1_D[2]					
MMC1_D[6]	AB7	SE63		MMC1_D[6]					
SD3_1P8_EN	AB8	SE85		SD3_1P8_EN					
SD3_D[1]	AB9	SE30		SD3_D[1]					
ISH_GPIO[7]/ I2S4_DATAIN	AC1	E16		ISH_GPIO[7]		I2S4_DATAIN			
SD3_D[0]	AC10	SE35		SD3_D[0]					
FST_SPI_D[1]	AC11	SW5		FST_SPI_D[1]					
UART2_RTS_N	AC12	SW19		UART2_RTS_N					
UART1_CTS_N	AC13	SW18		UART1_CTS_N					
UART1_DATAOUT/ UART0_DATAOUT	AC14	SW20		UART1_DATAOUT	UART0_DATAOUT				
NFC_I2C_DATA	AC15	SW51		NFC_I2C_DATA					
NFC_I2C_CLK	AC16	SW54		NFC_I2C_CLK					
I2C2_DATA	AC17	SW62		I2C2_DATA					

Table 37. Multiplexed Functions - T3 SoC (Sheet 6 of 9)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
GPIO_SW78	AC18	SW78							
LPE_I2S2_FRM	AC19	SW96		LPE_I2S2_FRM					
ISH_GPIO[4]/ I2S4_CLK	AC2	E22		ISH_GPIO[4]		I2S4_CLK			
GPIO_SW93	AC20	SW93							
MMC1_D[3]	AC6	SE26		MMC1_D[3]					
MMC1_D[4]	AC7	SE67		MMC1_D[4]					
MMC1_D[5]	AC8	SE65		MMC1_D[5]					
SD3_CLK	AC9	SE31		SD3_CLK					
SD3_D[3]	AD10	SE32		SD3_D[3]					
UART2_CTS_N	AD12	SW22		UART2_CTS_N					
UART1_DATAIN/ UART0_DATAIN	AD14	SW16		UART1_DATAIN	UART0_DATAIN				
I2C6_DATA/ SD3_WP	AD16	SW49		I2C6_DATA	SD3_WP				
MMC1_RESET_N	AD18	SW80			MMC1_RESET_N				
PWM[1]/ ISH_GPIO[10]/ ISH_UART_DATAOUT	AD2	SE1		PWM[1]	ISH_GPIO[10]	ISH_UART_DATAOUT			
SD3_WP	AD20	SW95			SD3_WP				
PWM[0]	AD3	SE5		PWM[0]					
SD2_D[0]	AD4	SE25		SD2_D[0]					
SD2_CLK	AD5	SE19		SD2_CLK					
SD2_D[3]_CD_N	AD6	SE15		SD2_D[3]_CD_N					

Table 37. Multiplexed Functions - T3 SoC (Sheet 7 of 9)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
MMC1_CLK	AD8	SE16		MMC1_CLK					
UART2_DATAIN	AE12	SW17		UART2_DATAIN					
UART1_RTS_N	AE13	SW15		UART1_RTS_N					
I2C6_CLK/NMI_N	AE16	SW53		I2C6_CLK	NMI_N				
I2C5_DATA	AE17	SW45		I2C5_DATA					
UART0_DATAIN	AE18	SW77			UART0_DATAIN				
SD2_D[1]	AE4	SE18		SD2_D[1]					
SD2_D[2]	AE5	SE21		SD2_D[2]					
SD3_CD_N	AE9	SE81		SD3_CD_N					
GPIO_CAMERASB10	B10	N50		GPIO_CAMERASB10					
DDI0_BKLTCTL	B8	N65		DDI0_BKLTCTL					
GPIO_CAMERASB11	C11	N55		GPIO_CAMERASB11					
GPIO_SUS8	C13	N23		GPIO_SUS8					
DDI0_BKLTEN	C8	N60		DDI0_BKLTEN					
GPIO_CAMERASB09	D10	N52		GPIO_CAMERASB09					
GPIO_CAMERASB08	D11	N47		GPIO_CAMERASB08					
DDI0_HPD	D9	N61		DDI0_HPD					

Table 37. Multiplexed Functions - T3 SoC (Sheet 8 of 9)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
SVID_ALERT_N	E12	N38		SVID_ALERT_N					
DDI2_HPD	E9	N68		DDI2_HPD					
SVID_CLK	F11	N40		SVID_CLK					
SVID_DATA	F12	N33	SVID_DATA						
GPIO_DFX4	F18	N5							
PROCHOT_N	F9	N32		PROCHOT_N					
LPE_I2S2_DATAIN	U17	SW94		LPE_I2S2_DATAIN					
PMC_SUSPWRDNACK	V12	SE83		PMC_SUSPWRDNACK					
LPE_I2S2_DATAOUT	V18	SW97		LPE_I2S2_DATAOUT					
PMC_SUS_STAT_N	V5	E2		PMC_SUS_STAT_N					
I2C1_DATA	W15	SW60		I2C1_DATA					
I2C1_CLK	W16	SW63		I2C1_CLK					
PMC_PWRBTN_N	W3	E8		PMC_PWRBTN_N					
ISH_GPIO[3]/ I2S3_DATAIN	Y1	E15		ISH_GPIO[3]		I2S3_DATAIN			
LPE_I2S1_DATAIN	Y13	SW37		LPE_I2S1_DATAIN					
LPE_I2S1_DATAOUT	Y14	SW34		LPE_I2S1_DATAOUT					
LPE_I2S0_FRM	Y16	SW35		LPE_I2S0_FRM					

Table 37. Multiplexed Functions - T3 SoC (Sheet 9 of 9)

GPIO Pin Name	Package Ball #	GPIO #	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
I2C0_CLK	Y17	SW65		I2C0_CLK					
I2C0_DATA	Y18	SW61		I2C0_DATA					
ISH_GPIO[1]/ I2S3_FS	Y2	E18		ISH_GPIO[1]		I2S3_FS			
PMC_WAKE_N	Y3	E10		PMC_WAKE_N					
PMC_SUSCLK[0]	Y4	E6		PMC_SUSCLK[0]					
PMC_PLTRST_N	Y5	E5		PMC_PLTRST_N					
MMC1_D[1]	Y8	SE24		MMC1_D[1]					
USB_OC[0]_N	Y9	SE80		USB_OC[1]_N					

§

3 Processor Core

Up to four out-of-order execution processor cores are supported, each dual core module supports up to 1 MB of L2 cache.

3.1 Features

- 14nm Process technology.
- Quad Out-of-Order Execution (OOE) processor cores.
- Primary 32 KB, 8-way L1 instruction cache and 24 KiB, 6-way L1 write-back data cache.
- Cores are grouped into dual-core modules: modules share a 1 MB, 16-way L2 cache (2 MB total for Quad Core)Intel® Streaming SIMD Extensions 4.1 and 4.2 (SSE4.1 and SSE4.2), which include new instructions for media and for fast XML parsing.
- Intel® 64 Bit architecture.
- Supports IA 32-bit.
- Supports Intel® VT-x2.
- Supports Intel® Advanced Encryption Standard (AES) New instructions (AES-NI).
- Supports Intel® Carry-Less Multiplication Instruction (PCLMULQDQ).
- Supports Digital Random Number Generator (DRNG).
- Supports C0, C1, C1E, C6C, C6 and C7 states.
- Thermal management support via Intel® Thermal Monitor (TM1 & TM2).
- Uses Power Aware Interrupt Routing (PAIR).

Note: Intel® Hyper-Threading Technology is not supported.

3.1.1 Intel® Virtualization Technology (Intel® VT)

Intel® Virtualization Technology (Intel® VT) makes a single system appear as multiple independent systems to software. This allows multiple, independent operating systems to run simultaneously on a single system. Intel® VT comprises technology components to support virtualization of platforms based on Intel architecture microprocessors and chipsets. Intel® Virtualization Technology for IA-32, Intel® 64 and Intel® Architecture (Intel® VT-x2) added hardware support in the processor to improve the virtualization performance and robustness.

Intel® VT-x2 specifications and functional descriptions are included in the *Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3B* and is available at: http://www.intel.com/products/processor/manuals/index.htm.

Other Intel® VT-x2 documents can be referenced at: http://www.intel.com/technology/virtualization/index.htm

3.1.1.1 Intel® VT-x2 Objectives

- Robust: VMMs no longer need to use paravirtualization or binary translation. This means that they will be able to run off-the-shelf OSs and applications without any special steps.
- Enhanced: Intel® VT enables VMMs to run 64-bit guest operating systems on IA x86 processors.
- More reliable: Due to the hardware support, VMMs can now be smaller, less complex, and more efficient. This improves reliability and availability and reduces the potential for software conflicts.
- More secure: The use of hardware transitions in the VMM strengthens the isolation of VMs and further prevents corruption of one VM from affecting others on the same system. Intel® VT-x2 provides hardware acceleration for virtualization of IA platforms. Virtual Machine Monitor (VMM) can use Intel® VT-x2 features to provide improved reliable virtualized platform.

3.1.1.1.1 Intel® VT-x2 Features

- Extended Page Tables (EPT)
  - EPT is hardware assisted page table physical memory virtualization.
  - Support guest VM execution in unpaged protected mode or in real-address mode.
  - It eliminates VM exits from guest OS to the VMM for shadow page-table maintenance.
- Virtual Processor IDs (VPID)
  - A VM Virtual Processor ID is used to tag processor core hardware structures (such as TLBs) to allow a logic processor to cache information (such as TLBs) for multiple linear address spaces.
  - This avoids flushes on VM transitions to give a lower-cost VM transition time and an overall reduction in virtualization overhead.
- Guest Preemption Timer
  - Mechanism for a VMM to preempt the execution of a guest OS VM after an amount of time specified by the VMM. The VMM sets a timer value before entering a guest.
  - The feature aids VMM developers in flexibility and Quality of Service (QoS) guarantees flexibility in guest VM scheduling and building Quality of Service (QoS) schemes.
- Descriptor-Table Exiting
  - Descriptor-table exiting allows a VMM to protect a guest OS from internal (malicious software based) attack by preventing relocation of key system data

structures like IDT (interrupt descriptor table), GDT (global descriptor table), LDT (local descriptor table), and TSS (task segment selector).

  - A VMM using this feature can intercept (by a VM exit) attempts to relocate these data structures and prevent them from being tampered by malicious software.

- VM Functions
  - VM function is an operation provided by the processor that can be invoked using the VMFUNC instruction from guest VM without a VM exit.
  - VM function to perform EPTP switching is supported and allows guest VM to load a new value for the EPT pointer, thereby establishing a different EPT paging structure hierarchy.

3.1.2 Security and Cryptography Technologies

3.1.2.1 Advanced Encryption Standard New Instructions (AES-NI)

The processor supports Advanced Encryption Standard New Instructions (AES-NI) that are a set of Single Instruction Multiple Data (SIMD) instructions that enable fast and secure data encryption and decryption based on the Advanced Encryption Standard (AES). AES-NI are valuable for a wide range of cryptographic applications, for example: applications that perform bulk encryption/decryption, authentication, random number generation, and authenticated encryption. AES is broadly accepted as the standard for both government and industry applications, and is widely deployed in various protocols.

AES-NI consists of six Intel® SSE instructions. Four instructions, namely AESENC, AESENCLAST, AESDEC, and AESDELAST facilitate high performance AES encryption and decryption. The other two, AESIMC and AESKEYGENASSIST, support the AES key expansion procedure. Together, these instructions provide a full hardware for support AES, offering security, high performance, and a great deal of flexibility.

3.1.2.2 PCLMULQDQ Instruction

The processor supports the carry-less multiplication instruction, PCLMULQDQ. PCLMULQDQ is a Single Instruction Multiple Data (SIMD) instruction that computes the 128-bit carry-less multiplication of two, 64-bit operands without generating and propagating carries. Carry-less multiplication is an essential processing component of several cryptographic systems and standards. Hence, accelerating carry-less multiplication can significantly contribute to achieving high speed secure computing and communication.

3.1.2.3 Digital Random Number Generator

The processor introduces a software visible digital random number generation mechanism supported by a high quality entropy source. This capability is available to programmers through the new RDRAND instruction. The resultant random number generation capability is designed to comply with existing industry standards (ANSI X9.82 and NIST SP 800-90).

Some possible uses of the new RDRAND instruction include cryptographic key generation as used in a variety of applications including communication, digital signatures, secure storage, and so on.

3.1.3 Power Aware Interrupt Routing

PAIR is an improvement in H/W routing of "redirectable" interrupts. Each core power-state is considered in the routing selection to reduce the power or performance impact of interrupts. System BIOS configures the routing algorithm, e.g. fixed-priority, rotating, hash, or PAIR, during setup via non-architectural register. The PAIR algorithm can be biased to optimize for power or performance and the largest gains will be seen in systems with high interrupt rates.

3.2 Platform Identification and CPUID

In addition to verifying the processor signature, the intended processor platform type must be determined to properly target the microcode update. The intended processor platform type is determined by reading bits [52:50] of the IA32_PLATFORM_ID register, (MSR 17h) within the processor. This is a 64-bit register that must be read using the RDMSR instruction. The 3 Platform Id bits, when read as a binary coded decimal (BCD) number, indicate the bit position in the microcode update header's Processor Flags field that is asSoCiated with the installed processor.

Executing the CPUID instruction with EAX=1 will provide the following information.

EAX	Field Description
[31:28]	Reserved
[27:20]	Extended Family value
[19:16]	Extended Model value
[15:13]	Reserved
[12]	Processor Type Bit
[11:8]	Family value
[7:4]	Model value
[3:0]	Stepping ID Value

3.3 References

For further details on Intel® 64 and IA-32 architectures refer Intel® 64 and IA-32 Architectures Software Developer's Manual Combined Volumes:1, 2A, 2B, 2C, 3A, 3B, and 3C:

- http://www.intel.com/content/www/µs/en/processors/architectures-software-developer-manuals.html

For more details on AES-NI refer:

- Intel ® Performance Primitives (IPP) web page - http://software.intel.com/en-us/intel-ipp/
- White Paper on AES-NI - http://software.intel.com/en-us/articles/intel-advanced-encryption-standard-aes-instructions-set/

For more details on using the RDRAND instruction refer Intel® Advanced Vector Extensions Programming Reference.

§

4 Integrated Clock

Clocks are integrated, consisting of multiple variable frequency clock domains, across different voltage domains. This architecture achieves a low power clocking solution that supports the various clocking requirements of the SoC's many interfaces.Platform clocking is provided internally by the iClock block and does not require external devices for clocking. All the required platform clocks are provided by only two inputs: a 19.2 MHz primary reference for the integrated clock block and a 32.768 kHz reference for the Real Time Clock (RTC) block. Both of these would likely be implemented as crystal references.

The different inputs and outputs are listed below.

Table 38. SoC Clock Inputs

Clock Domain	Signal Name	Frequency	Usage/Description
Main	ICLK_OSCIN ICLK_OSCOUT	19.2 MHz	Reference crystal for the iCLK PLL
RTC	RTC_X1 RTC_X2	32.768 kHz	RTC crystal I/O for RTC block
LPC	LPC_CLKOUT	19.2 MHz	Can be configured as an input to compensate for board routing delays through Soft Strap.

Table 39. SoC Clock Outputs (Sheet 1 of 2)

Clock Domain	Signal Name	Frequency	Usage/Description
DDR	LPDDR3_M0_CK_P_A/B LPDDR3_M0_CK_N_A/B LPDDR3_M1_CK_P_A/B LPDDR3_M1_CK_N_A/B	800 MHz	Drives the Memory ranks 0-1. Data rate (MT/s) is 2x the clock rate.
SDXC	MMC1_CLK SD2_CLK SD3_CLK	200 MHz	Clock for Storage Devices
SPI	SPI1_CLK FST_SPI_CLK	20 MHz, 33 MHz, 50 MHz	Clock for SPI flash
PMIC/COMMS	PMC_SUSCLK[0]	32.768 kHz	Pass through clock from RTC oscillator
LPC	LPC_CLKOUT[0:1]	19.2 MHz	Provided to devices requiring LPC clock
Display Port	DDI[0]_TXP[3] DDI[0]_TXN[3]	162 or 270 MHz	Differential clock for DP devices
HDMI	DDI[2]_TXP[3] DDI[2]_TXN[3]	25-297 MHz	Differential clock for HDMI devices

Table 39. SoC Clock Outputs (Sheet 2 of 2)

Clock Domain	Signal Name	Frequency	Usage/Description
HDMI DDC	DDI[2:0]_DDCCLK	100 kHz	Clock for HDMI DDC devices
MIPI DSI	MDSI_A_CLKP MDSI_A_CLKN MDSI_C_CLKP MDSI_C_CLKN	1000 MHz	Differential clock for MIPI DSI Devices
MIPI CSI	MCSI1_CLKP MCSI1_CLKN MCSI2_CLKP MCSI2_CLKN MCSI3_CLKP MCSI3_CLKN	200-400 MHz	Clocks for front and rear cameras
SVID	SVID_CLK	20 MHz	Clock used by voltage regulator
I^2S	LPE_I2S[2:0]_CLK	9.6 MHz	Continuous serial clock for I^2S interfaces
Platform Clocks	PLT_CLK [5:0]	19.2MHz	Platform clocks.
SIO SPI	SPI_CLK	15 MHz	SPI clock output
I^2C	I2C[6:0]_CLK	1.7MHz	I^2C clocks
NFC	NFC_I2C_CLK	100 kHz	Clock for NFC device

§

5 Power Up and Reset Sequence

This chapter provides information on the following topics:

- "Power Up Sequences"
- "Power Down Sequences"
- "Reset Behavior"

5.1 SoC System States

5.1.1 System Sleeping States Control (S-states)

The SoC supports the S0, S0i1, S0i2, S0i3, S4, and S5 sleep states. S4 and S5 states are identical from a hardware and power perspective. The differentiation is software determined (S4 = Suspend to Disk).

The SoC platform architecture assumes the usage of an external power management controller e.g., CPLD or PMIC. Some flows in this section refer the power management controller for support of the S-states transitions.

The SoC sleep states are described in Chapter 7, "Power Management".

5.2 Power Up Sequences

5.2.1 RTC Power Well Transition (G5 to G3 States Transition)

When RTC_VCC (Real Time Clock power) is applied via RTC battery, the following occurs (refer Figure 2 for timing):

1. RTC_VCC ramps. RTC_RST_N should be low.
2. The system starts the real time clock oscillator.
3. A minimum of t1 units after RTC_VCC ramps, the external RTC RC circuit de-asserts RTC_RST_N. The system is now in the G3 state. RTC oscillator is unlikely to be stable at this point.

Figure 2. RTC Power Well Timing Diagrams

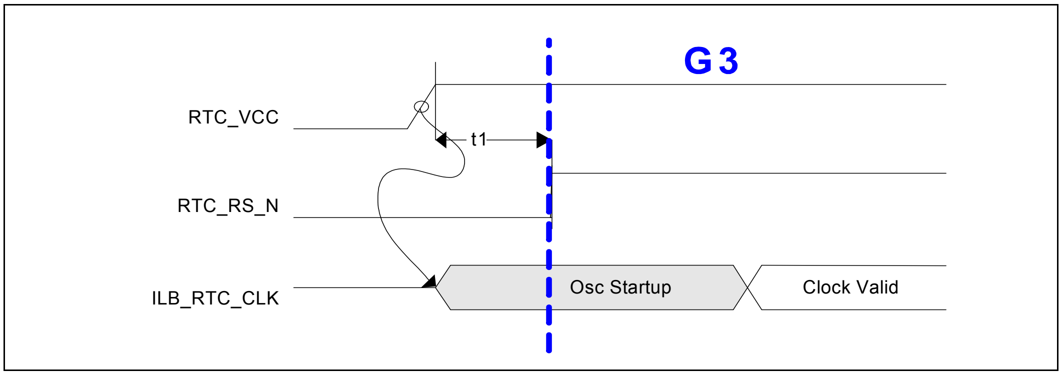


Table 40. RTC Power Well Timing Parameters

Parameter	Description	Min	Max	Units
t1	RTC_VCC to RTC_RST_N de-assertion	9	-	ms

NOTES:
1. This delay is typically created from an RC circuit.
2. The oscillator startup times are component and design specific. A crystal oscillator can take several second to reach a large enough voltage swing. A silicon oscillator can have startups times <10 ms.
3. All VCC measurements points are at 90% nominal VCC voltage.

5.2.2 G3 to S4/S5

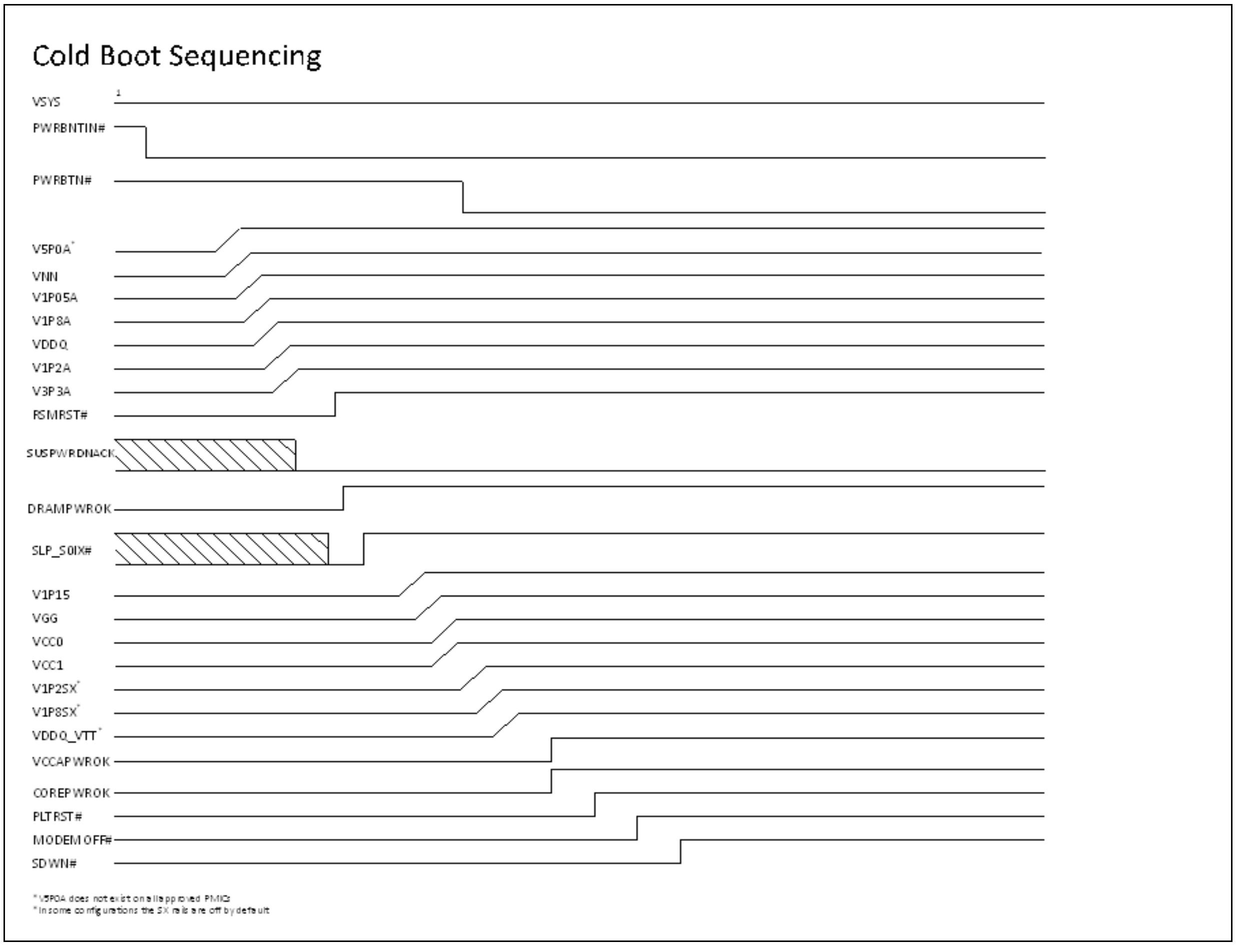
The timings shown in Figure 3 occurs when a board event such as AC power or power button is pressed. The following occurs:

1. Suspend well ramp in the order given.
2. The external power management controller de-asserts PMC_RSMRST_N after the suspend rails become stable.
3. PMC_SUSCLK will begin toggling after the de-assertion of PMC_RSMRST_N.
4. The system is now in S4/S5 state. Depending on policy bits, the system either waits for a wake event, or continues to S0 states.

5.2.3 S4/S5 to S0

1. The external power management controller detects an event (i.e., power button) to initiate transition from S4/S5 to S0.
2. VCC, VNN and other S0 core voltage power rails may be enabled after the initiation of the S4/S5 to S0 event. The VCC and VNN voltage rails must be driven to the default values.
3. After the DRAM power rail ramp, the external power management controller drives DRAM_PWROK high.
4. After all of the S0 core voltage power rails are stable, external power management controller drives PMC_CORE_PWROK and VCCA_PWROK to HIGH.
5. The processor de-asserts PMC_PLTRST_N after PMC_CORE_PWROK is stable. The PMC_PLTRST_N is the main platform reset to other components.
6. The processor will begin fetching code from either the PCU-located SPI interface or the LPC interface.

Figure 3. S4/S5 to S0 (Power Up) Sequence



NOTES:

1. RTC and SUS power rails may come up at the same time if no RTC battery is used.
2. RTC clock should be oscillating, but may not be at 32.768 KHz yet.
3. Wake events show in figure are optional and depending on platform configuration.

5.3 Power Down Sequences

5.3.1 S0 to S4/S5 Sequence

Entry to Sleep states (S4, S5) is initiated by any of the following methods:

- Setting the desired sleep type in PM1_CNT.SLP_TYP and setting PM1_CNT.SLP_EN.
- Detection of an external catastrophic temperature event may cause a transition to G3, if the system is designed to do so.

The following sequence applies to S0-S4/S5 transitions.

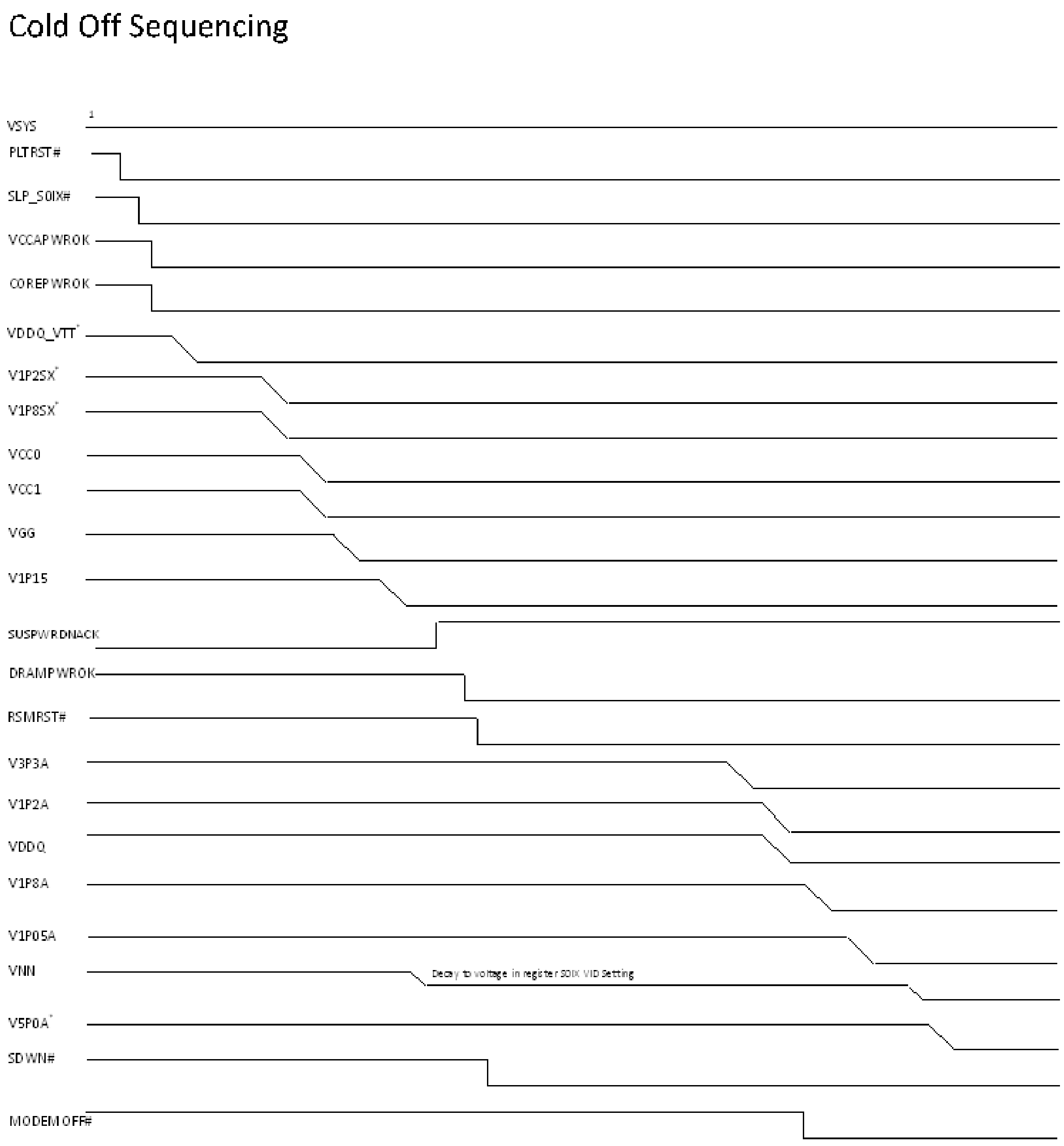
1. The Operating System Power Management (OSPM) will handle the enabling or disabling of interrupt generation after S4 resume. The Operating System Power Management (OSPM) will need to read and clear Wake status information and the processing of the clearing wake status which will include enabling interrupts (both at the core level and platform level).
2. All interrupts in the processor need to be disabled before the S4 sequence is started (and re-enabled on exit). The CPU APIC must be disabled.
3. When the desired sleep state is set in the PM1_CNT.TYP and PM1_CNT.SLP_EN registers, a sleep state request is sent to the PMC.
4. The PMC flushes all the internal buffers to main memory.

The Power Down Sequence is shown in Figure 4 below.

Other Assumptions:

- Entry to a Cx state is mutually exclusive with software-initiated entry to a Sleep state. This is because the processor(s) can only perform one register access at a time. This requirement is enforced by the CPU as well as the OS. The system may hang if it attempts to do a C-state and S-state at the same time.
- The G3 system state cannot be entered via any software mechanism. The G3 state indicates a complete loss of power. In this state, the RTC well may or may not be powered by an external coin cell battery.
- An external Power Management Controller (PMIC/EC) can be used to put the processor in G3 when the S4/S5 state is requested by the SoC. This is done to save power in S4/S5 state. This G3 like state is enabled by removing SUS rails via the SUSPWRDNACK pin. Doing so prevents the use of any of SUS wake events including USB, RTC, and GPIOs including the power button. The external Power Management Controller (or re-application of power) is required to return to S0.

Figure 4. S0 to S4/S5 (Power Down) Sequence



*V5P0A does not exist on all approved PMICs
*In some configurations the SX rails are off by default

5.3.2 S4/S5 to S0 (Exit Sleep States)

Sleep states (S5) are exited based on Wake events. The Wake events will force the system to a full on state (S0), although some non-critical subsystems might still be powered down and have to be brought back manually. For example, the hard disk may be powered down during a sleep state, and have to be enabled via an I/O pin before it can be used. Upon exit from software-entered Sleep states (i.e., those initiated via the PM1_CNT.SLP_EN bit), the PM1_STS_EN.WAK_STS bit will be set.

To enable Wake Events, the possible causes of wake events (and their restrictions) are shown in Table 41.

Table 41. S4/S5 to S0 Cause of Wake Events

Cause	Type	How Enabled
RTC Alarm	Internal	Set PM1_STS_EN.RTC_EN register bit
PMC_PWRBTN_N (Power Button)	External	Default enabled as Wake event
GPIO_NORTH And GPIO_SOUTHWEST	External	GPE0a_EN register (after having gone to S5 via PM1_CNT.SLP_EN, but not after a power failure.) Note: GPIOs that are in the core well are not capable of waking the system from sleep states where the core well is not powered.
GPIO_SOUTHEAST	External	Southeast GPIO can (optionally) be used as Wake sources based on GPIO register programming.
Primary PME_N	Internal	GPEOa_EN.PME_B0_EN register bit. This wake status bit includes multiple internal agents: EHCI (USB2)
PMC - Initiated	Internal	No enable bits. The PMC can wake the host independent of other wake events listed, if desired. A bit is provided in PRSTS for reporting this wake event to BIOS. Note that this wake event may be used as a wake trigger on behalf of some other wake source.

5.3.3 Enter S0ix

The S0IX state is entered when the SoC is in a shallow sleep state. This state is entered when the SoC asserts the PMC_SLP_S0IX_N (LOW) pin to the PMIC. VDDQ_VTT and SX rails are turned off. The VCC rail is turned off by SVID commands (not by PMC_SLP_S0IX_N signal).The VNN rail is set to a voltage set in SVID address 39h.The rest of the VRs remain on but enters into PFM/power save mode.

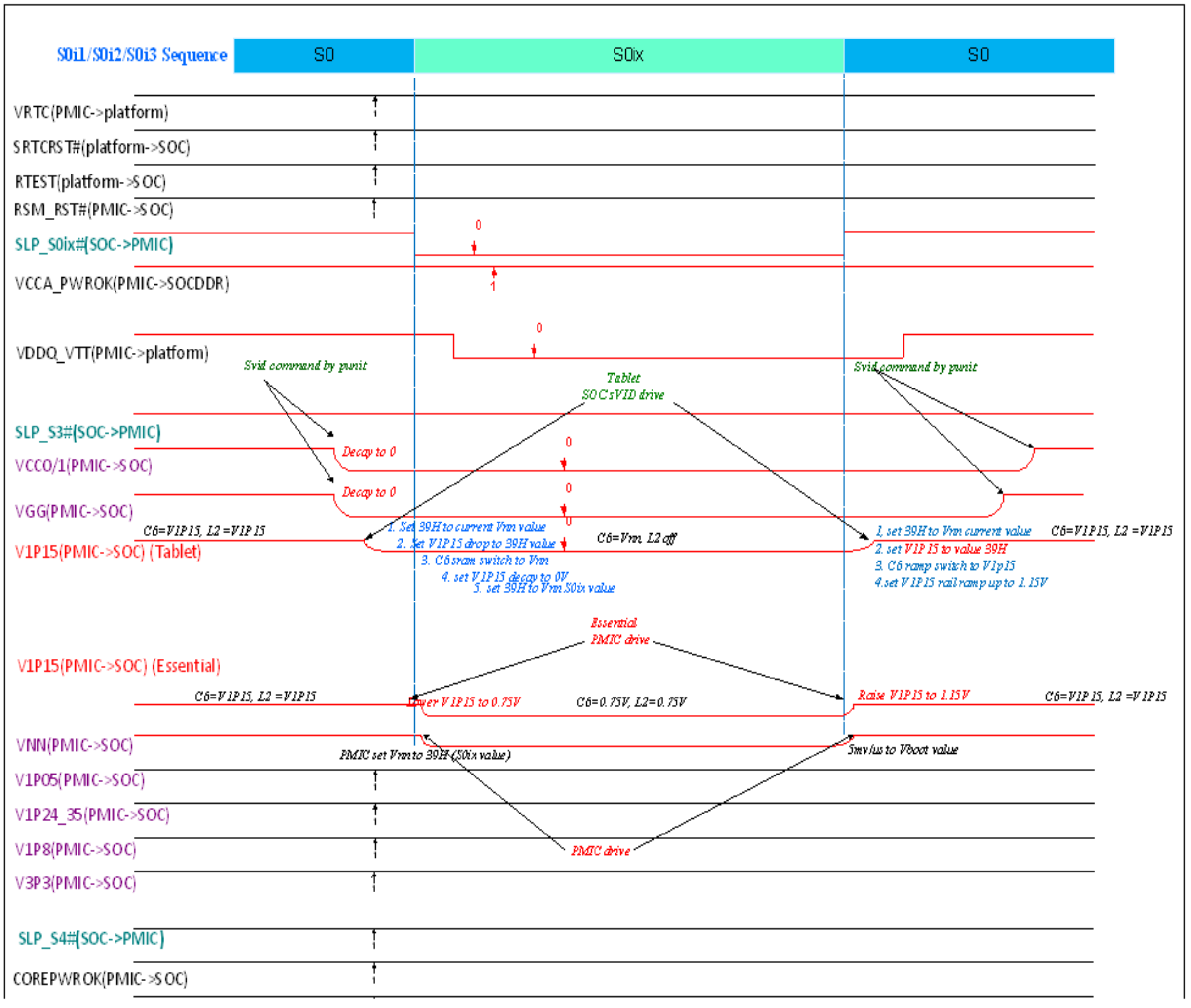
5.3.4 Exit S0ix

The S0IX state is exited when the SoC de-asserts the PMC_SLP_S0IX_Npin (HIGH). VDDQ_VTT and SX rails are turned on. The VCC rail will be turned on by SVID commands (not by PMC_SLP_S0IX_N). The rest of the rails will come out of PFM/power save mode. All SMI/SCI events wake SoC from the S0ix states. The following table lists the addition events that wake the SoC from S0ix states.

Table 42. S0ix Cause of Wake Events

Cause	Type	How Enabled
Any GPIO	External	IO-APIC forwards the interrupt, resulting in S0 (as configured by BIOS). Alternatively use S0ix Wake Register (S0IX_WAKE_EN and S0IX_WAKE_STS) in PMC
LPC CLKRUN	External	Wake from S0i2/3 only when the signal is asserted, moves the SoC to S0i1.
ISH	External	From External Sensors
USB	External	USB Port connected device / host

Figure 5. S0 to S0ix Entry and Exit Sequence



5.3.5 Handling Power Failures

The power failures can occur if the AC power or battery is removed. In this case, when the system was originally in a S0 state, power failure bit (GEN_PMCON1.PWR_FLR) is set after a power failure. Software can clear the bit.

5.4 Reset Behavior

There are several ways to reset the processor.

Table 43. Types of Resets (Sheet 1 of 2)

Trigger	Description	Note
Write of 0Eh to CF9 Register	Write of 0Eh to the CF9 register	TYPE 2: Host Reset with Power Cycle: Cold reset. PMC will lose all the information. All the functionality in SoC gets reset. The host system automatically is powered back up and brought out of reset to S0 state. SoC must not drop this type of reset request if received while the system is in a software-entered S4/5 state.However, SoC is allowed to perform the reset without executing the RESET_WARN protocol in these states. If the system is in S5 due to a reset type #8 event, SoC is allowed to drop this type of reset request.
PMC_RSTBTN_N & CF9h bit 3= 1	User presses the reset button, causing the PMC_RSTBTN_N signal to go active (after the debounce logic)	
PMC_RSTBTN_N & CF9h bit 3= 0	User presses the reset button, causing the PMC_RSTBTN_N signal to go active (after the debounce logic)	TYPE 1:Host Reset with Power Cycle: Warm Reset 1. Host-Only functionality in SoC gets reset 2. Any functionality that needs to remain operational during a host reset must not get reset. 3. PMC does not get reset. 4. RTC remain information. 5. Suspend well remain information 6. S4/S5 drop the warm reset request.
Write of 06h to CF9 Register	Write of 06h to the CF9 register	
TCO watchdog timer	TCO timer reaches zero two times	
S4/S5	The processor is reset when going to S4 or S5 state	TYPE 4: Sx Entry (host stays there) 1. All the Vnn reset by external power Good. Except: 1. PMC remain information. 2. RTC remain information. 3. Suspend well remain information

Table 43. Types of Resets (Sheet 2 of 2)

Trigger	Description	Note
Power Failure	PMC_CORE_PWROK signal goes inactive in S0/S1	TYPE 7: Global, Power Cycle Reset: S0->S4/S5->S0 1. All the Vnn reset by external power Good. 2. All power wells that are controlled by the PMC_SLP_S0iX_N pins are turned off. 3. PMC get reset. 4. External Dram-unchanged Except: 1. RTC retain information. 2. Suspend well retain information.
Write of 06h or 0Eh to CF9 Register	CF9h global Reset bit = 1b	
Host Partition Reset Entry Timeout	Host partition reset entry sequence took longer than the allowed timeout value (presumably due to a failure to receive one of the internal or external handshakes)	
Processor Thermal Trip	The internal thermal sensor signals a catastrophic temperature condition – transition to S5 and reset asserts	SOC_G3: Straight-to-S5 (thermal trip->SOC_G3) SOC power cycle: S0->SOC_G3 SOC lost all the info Except: RTC retain info
PMC_PWRBTN_N	10-second press causes transition to S5 (and reset asserts)	TYPE 8: Straight-to-S5 (Host stays there) SOC power cycle: S0->S4->S5 1. All the Vnn reset by external power Good. 2. All power wells that are controlled by the PMC_SLP_S0iX_N pins are turned off. 3. External Dram-unchanged Except: 1. PMC retain information. 2. RTC retain information. 3. Suspend well retain information
PMC_PWRBTN_N Power Button Override		
S4/S5 Entry Timeout	S4, or S5 entry sequence took longer than the allowed timeout value (presumably due to a failure to receive one of the internal or external handshakes)	
PMC Watchdog Timer	Firmware hang and Watchdog Timeout detected in the PMC platform	
CPU Shutdown with Policy to assert PMC_PLTRST_N	Shutdown special cycle from CPU can cause either INIT or Reset Control-style PMC_PLTRST_N	Type 7:Global, Power Cycle Reset (if CF9h Global Reset bit = 1b) Type 2:Host Reset with Power Cycle (if CF9h Register bit 3 = 1b) Type 1:Host Reset without Power Cycle (others setting)

§

6 Thermal Management

The SoC's thermal management system helps in managing the overall thermal profile of the system to prevent overheating and system breakdown. The architecture implements various proven methods of maintaining maximum performance while remaining within the thermal spec. Throttling mechanisms are used to reduce power consumption when thermal limits of the device are exceeded and the system is notified of critical conditions via interrupts or thermal signalling pins. SoC thermal management differs from legacy implementations primarily by replacing dedicated thermal management hardware with firmware.

The thermal management features are:

- Eight digital thermal sensors (DTS).
- Supports hardware trip point and four programmable trip points based on the temperature indicated by thermal sensors.
- Supports different thermal throttling mechanisms.

6.1 Thermal Sensors

SoC Sensors are based on DTS (Digital Thermal Sensor) to provide more accurate measure of system thermals.

The SoC has 8 DTS's. DTS provides as wires the current temperature around the real estate it occupies on SoC. These are driven to PM unit, which in turn monitor the temperature from DTS on the SoC.

DTS output are adjusted for silicon variations. For a given temperature the output from DTS is always the same irrespective of silicon.

Table 44. Temperature Reading Based on DTS (Sheet 1 of 2)

DTS Counter Value [8:0]	Temperature Reading (If $T_{J\text{-MAX}}$ =90°C)	Temperature Reading (If $T_{J\text{-MAX}}$ =100°C)	Temperature Reading (If $T_{J\text{-MAX}}$ =110°C)	Temperature Reading (If $T_{J\text{-MAX}}$ =100°C) Thermal Read Register [7:0]
127	90°C	100°C	110°C	100°C
137	80°C	90°C	100°C	90°C
147	70°C	80°C	90°C	80°C
157	60°C	70°C	80°C	70°C
167	50°C	60°C	70°C	60°C
177	40°C	50°C	60°C	50°C
187	30°C	40°C	50°C	40°C

Table 44. Temperature Reading Based on DTS (Sheet 2 of 2)

DTS Counter Value [8:0]	Temperature Reading (If $T_{J\text{-}MAX}$ =90°C)	Temperature Reading (If $T_{J\text{-}MAX}$ =100°C)	Temperature Reading (If $T_{J\text{-}MAX}$ =110°C)	Temperature Reading (If $T_{J\text{-}MAX}$ =100°C) Thermal Read Register [7:0]
197	20°C	30°C	40°C	30°C
207	10°C	20°C	30°C	20°C
217	0°C	10°C	20°C	10°C
227	-10°C	0°C	10°C	0°C
237	-20°C	-10°C	0°C	-10°C
247	-30°C	-20°C	-10°C	-20°C
257	-40°C	-30°C	-20°C	-28°C [255]
247	-50°C	-40°C	-30°C	-28°C [255]

Note: DTS encoding of 127 always represents Tjmax. If Tjmax is at 100°C instead of 90°C then the encoding 127 from DTS indicates 100°C, 137 indicates 90°C and so forth.

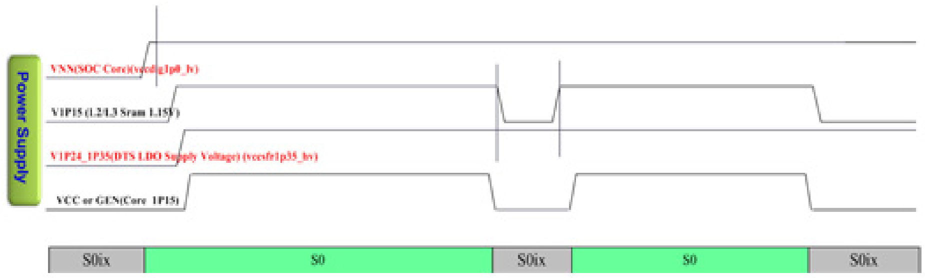
Thermal trip points are of two types:

- **Hard Trip**: The Catastrophic trip points generated by DTS's based on predefined temperature setting defined in fuses.
- **Programmable Trips**: SoC provides four programmable trip settings (Hot, Aux2, Aux1, Aux0) that can be set by firmware/software. Default value for Hot Trip is from Fuses.

6.1.1 DTS Timing

DTS should be enabled only after setting up SoC and system to prevent spurious counts from DTS to trigger thermal events. P-Unit determines when DTS is enabled. The figure below shows the various control signals needed for DTS operations.

Figure 6. DTS Operation Mode


Power Supply
VNN(SOC Core)(vccdig1p0_lv)
V1P15 (1.2/1.3 Sram 1.15V)
V1P24_1P35(DTS LDO Supply Voltage) (vccsfr1p35_lv)
VCC or GEN(Core 1P15)
S0ix
S0
S0ix
S0
S0ix

6.2 Hardware Trips

6.2.1 Catastrophic Trip (THERMTRIP)

Catastrophic trip is generated by DTS whenever the ambient temperature around it reaches (or extends) beyond the max value (indicated by a fuse). Catastrophic trip will not trip unless enabled (DTS are enabled only after HFPLL is locked). Within each DTS Catastrophic trips are flopped to prevent any glitches on Catastrophic signals from affecting the SoC behavior. Catastrophic trips are reset, once set, during power cycles.

Catastrophic trip signals from all DTS in the SoC are combined to generate THERMTRIP function which will in turn shut off all the PLL's and power rails to prevent SoC breakdown. To prevent glitches from triggering shutdown events, Catastrophic trip's from DTS's are registered before being sent out.

6.3 SoC Programmable Trips

Programmable trips can be programmed to cause different actions when triggered to reduce temperature of the die.

6.3.1 Aux3 Trip

By default, the Aux 3 (Hot Trip) point is set by software/firmware has an option to set these to a different value.

This trip point is enabled by firmware to monitor and control the system temperature while the rest of the system is being set up.

6.3.2 Aux2, Aux1, Aux0 Trip

These are fully programmable trip points for general hardware protection mechanisms. The programmable trips are only active after software/firmware enables the trip.

Note: Unlike Aux3, the Aux[2:0] trip registers are defaulted to zero. To prevent spurious results, software/firmware should program the trip values prior to enabling the trip point.

6.4 Platform Trips

6.4.1 PROCHOT#

The platform components use the signal PROCHOT# to indicate thermal events to SoC. Assertion of the PROCHOT# input will trigger Thermal Monitor 1 or Thermal Monitor 2 throttling mechanisms if they are enabled.

6.4.2 EXTTS

SoC does not support external thermal sensors and the corresponding bits in the P-Unit registers will be reserved for future use if needed.

For SoC, PROCHOT is the only mechanism for a platform component to indicate Thermal events to P-Unit.

6.4.3 sVID

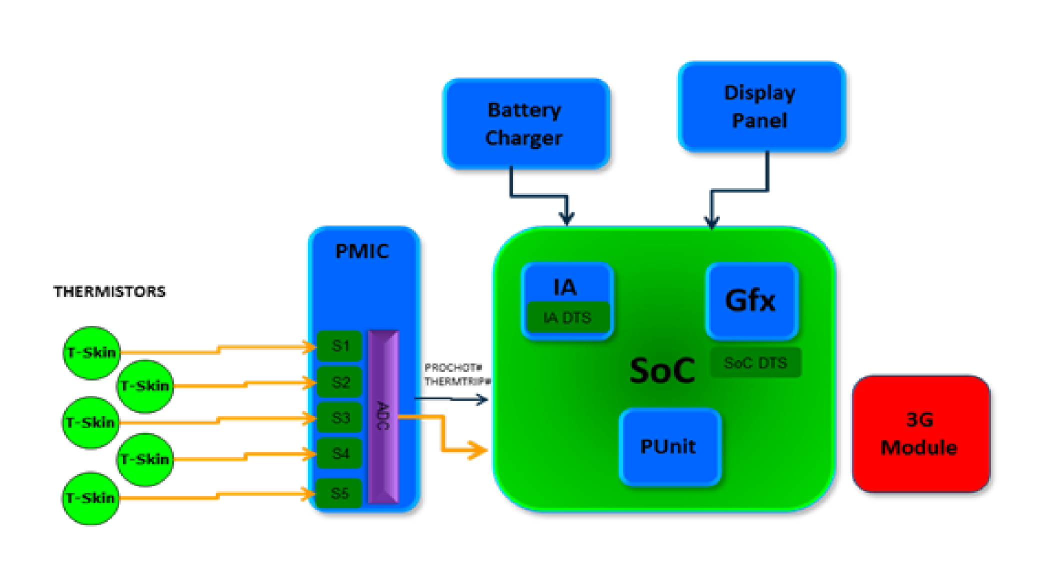
When the Voltage Regulator (VR) reaches it's threshold (VR_Icc_Max, VR_Hot), status bits in sVID are set. sVID sends SVID_Status message to PUnit.

6.5 Dynamic Platform Thermal Framework (DPTF)

SoC is required to support interface for OS level thermal drivers and Intel's DPTF (Dynamic Platform and Thermal Framework) drivers to control thermal management. This interface provides high-level system drivers a mechanism to manage thermal events within the SoC with respect to events outside SoC. These events could potentially be triggered before PM Unit firmware performs active management as DPTF/OS level drivers respond to events on platform outside of SoC.In addition, these interfaces also respond to interrupts from within the SoC.

Platform level thermal management layout is shown in the figure below.

Figure 7. Platform Level Thermal Management HW Layout



The thermal events happen outside of SoC on platform level are reported as interrupts from PMIC. PMIC monitors a number of catastrophic and critical thermal events, such as PMIC over-temperature, system over-temperature (reported by skin sensors), and battery over-temperature.

6.6 Thermal Status

The firmware captures Thermal Trip events (other than THERMTRIP) in status registers to trigger thermal actions. Associated with each event is a set of programmable actions.

§

7 Power Management

This chapter provides information on the following power management topics:

- ACPI States
- Processor Core
- Integrated Graphics Controller

7.1 Features

- ACPI System States support (S0, S0i1, S0i2, S0i3, S4, S5).
- Processor Core/Package States support (C0 – C7).
- SoC Graphics Adapter States support D0 – D3.
- Supports CPU and GFx Burst.
- Dynamic I/O power reductions (disabling sense amps on input buffers, tri-stating output buffers).
- Active power down of display links.

7.2 States Supported

The Power Management states supported by the processor are described in this section.

7.2.1 System States

Table 45. General Power States for System (Sheet 1 of 2)

States/Sub-states	Legacy Name / Description
G0/S0/C0	**FULL ON:** CPU operating. Individual devices may be shut down to save power. The different CPU operating levels are defined by Cx states.
G0/S0/Cx	**Cx State:** CPU manages C-state itself.
G0/S0i1	**S0i1 State:** Low power platform active state. All DRAM and IOSF traffic are halted. PLL are configured to be off. This state allows MP3 playing using ISH/LPE engine
G0/S0i2	**S0i2 State:** The SoC clocks and oscillators are parked
G0/S0i3	**S0i3 State:** All SoC clocks and oscillators are turned off

Table 45. General Power States for System (Sheet 2 of 2)

States/Sub-states	Legacy Name / Description
G1/S4	**Suspend-To-Disk (STD):** The context of the system is maintained on the disk. All of the power is shut down except power for the logic to resume. The S4 and S5 states are treated the same.
G2/S5	**Soft-Off:** System context is not maintained. All of the power is shut down except power for the logic to restart. A full boot is required to restart. A full boot is required when waking. The S4 and S5 states are treated the same.
G3	**Mechanical OFF.** System content is not maintained. All power shutdown except for the RTC. No "Wake" events are possible, because the system does not have any power. This state occurs if the user removes the batteries, turns off a mechanical switch, or if the system power supply is at a level that is insufficient to power the "waking" logic. When system power returns, transition will depend on the state just prior to the entry to G3.

Table 48 shows the transitions rules among the various states.

Note: Transitions among the various states may appear to temporarily transition through intermediate states. These intermediate transitions and states are not listed in the table.

Table 46. Cause of Sx Wake Events

Cause	How Enabled
RTC Alarm	Set RTC_EN bit in PM1_EN Register
Power Button	Always enabled as Wake event from Sx
PMC_SLP_S4_N	None
PMC_BATLOW_N	None
PMC_SUS_STAT_N	None
PMC_SLP_S0IX_N	None
PMC_ACPRESENT	None
PMC_PLTRST_N	None
PMC_SUSCLK[0]	None

The following shows the differences in the sleep states with regards to the processor's output signals.

Table 47. SoC Sx-States to SLP_S*# (Sheet 1 of 2)

State	S0	S4	S5	Reset w/o Power Cycle	Reset w/ Power Cycle
CPU Executing	In C0	OFF	OFF	No	OFF
PMC_SLP_S4_N	HIGH	LOW	LOW	HIGH	LOW

Table 47. SoC Sx-States to SLP_S*# (Sheet 2 of 2)

State	S0	S4	S5	Reset w/o Power Cycle	Reset w/ Power Cycle
S0 Power Rails	ON	OFF	OFF	ON	OFF
PMC_PLTRST_N	0	1	1	1	1
PMC_SUS_STAT_N	HIGH	LOW	LOW	HIGH	LOW

NOTE: The processor treats S4 and S5 requests the same. The processor does not have PMC_SLP_S4_N. PMC_SUS_STAT_N is required to drive low (asserted) even if core well is left on because PMC_SUS_STAT_N also warns of upcoming reset.

Table 48. ACPI PM State Transition Rules

Present State	Transition Trigger	Next State
G0/S0/C0	IA Code MWAIT or LVL Rd	C0/S0/Cx
	PM1_CNT.SLP_EN bit set	G1/Sx or G2/S5 state (specified by PM1_CNT.SLP_TYP)
	Power Button Override	G2/S5
	Mechanical Off/Power Failure	G3
G0/S0/Cx	Cx break events which include: CPU snoop, MSI, Legacy Interrupt, AONT timer	G0/S0/C0
	Power Button Override	G2/S5
	Resume Well Power Failure	G3
G1/S4	Any Enabled Wake Event	G0/S0/C0
	Power button Override	G2/S5
	Resume Well Power Failure	G3
G2/S5	Any Enabled Wake Event	G0/S0/C0
	Resume Well Power Failure	G3
G3	Power Returns	Option to go to S0/C0 (reboot) or G2/S5 (stay off until power button pressed or other enabled wake event) or G1/S4 (if system state was S4 prior to the power failure). Some wake events are preserved through a power failure.

7.2.2 Interface State Combinations

Table 49. G, S and C State Combinations (Sheet 1 of 2)

Global (G) State	Sleep (S) State	Processor Core (C) State	Processor State	System Clocks	Description
G0	S0	C0	Full On	On	Full On

Table 49. G, S and C State Combinations (Sheet 2 of 2)

Global (G) State	Sleep (S) State	Processor Core (C) State	Processor State	System Clocks	Description
G0	S0	C1/C1E	Auto-Halt	On	Auto-Halt
G0	S0	C6	Deep Power Down	On	Deep Power Down
G0	S0ix	C7	Deep Power Down	On	Deep Power Down
G1	S4	Power off		Off except RTC & internal ring OSC	Suspend to Disk
G2	S5	Power off		Off except RTC & internal ring OSC	Soft Off
G3	NA	Power Off		Power off	Hard Off

7.2.3 Integrated Graphics Display States

Table 50. SoC Graphics Adapter State Control

State	Description
D0	Full on, Display active
D3	Power off display

7.2.4 Integrated Memory Controller States

Table 51. Main Memory States

States	Description
Powerup	CKE asserted. Active mode.
Precharge Powerdown	CKE de-asserted (not self-refresh) with all banks closed.
Active Powerdown	CKE de-asserted (not self-refresh) with at least one bank active.
Self-Refresh	CKE de-asserted using device self-refresh

Table 52. D, S and C State Combinations (Sheet 1 of 2)

Graphics Adapter (D) State	Sleep (S) State	(C) State	Description
D0	S0	C0	Full On, Displaying
D0	S0	C1	Auto-Halt, Displaying
D0	S0	C6	Deep Sleep, Display Off

Table 52. D, S and C State Combinations (Sheet 2 of 2)

Graphics Adapter (D) State	Sleep (S) State	(C) State	Description
D0	S0ix	C7	Deep Sleep, Display Off
D3	S0/S0ix	Any	Not Displaying
D3	S4		Not Displaying Suspend to disk Core power off

7.3 Processor Core Power Management

While executing code, Enhanced Intel SpeedStep® Technology optimizes the processor's frequency and core voltage based on workload. Each frequency and voltage operating point is defined by ACPI as a P-state. When the processor is not executing code, it is idle. A low-power idle state is defined by ACPI as a C-state. In general, lower power C-states have longer entry and exit latencies.

7.3.1 Enhanced Intel SpeedStep® Technology

The following are the key features of Enhanced Intel SpeedStep® Technology:

- Applicable to Processor Core Voltage and Graphic Core Voltage
- Multiple frequency and voltage points for optimal performance and power efficiency. These operating points are known as P-states.
- Frequency selection is software controlled by writing to processor MSRs. The voltage is optimized based on the selected frequency:
  - If the target frequency is higher than the current frequency, Core_VCC is ramped up slowly to an optimized voltage. This voltage is signaled by the SVID signals to the voltage regulator. Once the voltage is established, the PLL locks on to the target frequency.
  - If the target frequency is lower than the current frequency, the PLL locks to the target frequency, then transitions to a lower voltage by signaling the target voltage on the SVID signals.
- The processor controls voltage ramp rates by requesting appropriate ramp rates from an external SVID controller.
- Because there is low transition latency between P-states, a significant number of transitions per second are possible.
- Thermal Monitor mode.
  - Refer Chapter 6, "Thermal Management"

7.3.2 Dynamic Cache Sizing*

Dynamic Cache Sizing allows the processor to flush and disable a programmable number of L2 cache ways upon each Deeper Sleep entry under the following condition:

- The C0 timer that tracks continuous residency in the Normal state, has not expired. This timer is cleared during the first entry into Deeper Sleep to allow consecutive Deeper Sleep entries to shrink the L2 cache as needed.
- The predefined L2 shrink threshold is triggered.

7.3.3 Low-Power Idle States

When the processor core is idle, low-power idle states (C-states) are used to save power. More power savings actions are taken for numerically higher C-state. However, higher C-states have longer exit and entry latencies. Resolution of C-state occur at the thread, processor core, and processor core level.

7.3.3.1 Clock Control and Low-Power States*

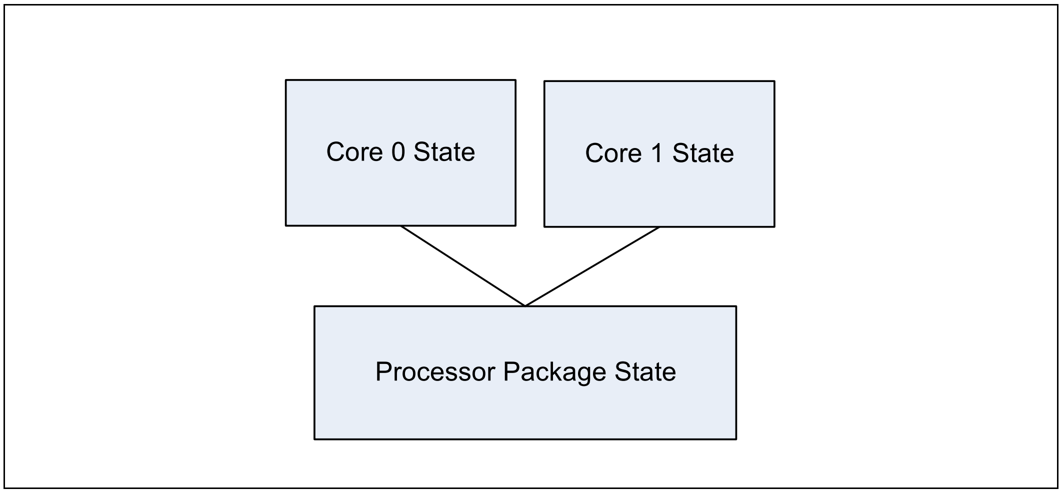
The processor core supports low power states at core level. The central power management logic ensures the entire processor core enters the new common processor core power state. For processor core power states higher than C1, this would be done by initiating a P_LVLx (P_LVL4 & P_LVL6) I/O read to all of the cores. States that require external intervention and typically map back to processor core power states. States for processor core include Normal (C0, C1), and Stop Grant.

The processor core implements two software interfaces for requesting low power states: MWAIT instruction extensions with sub-state specifies and P_LVLx reads to the ACPI P_BLK register block mapped in the processor core's I/O address space. The P_LVLx I/O reads are converted to equivalent MWAIT C-state requests inside the processor core and do not directly result in I/O reads on the processor core bus. The monitor address does not need to be setup before using the P_LVLx I/O read interface. The sub-state specifications used for each P_LVLx read can be configured in a software programmable MSR by BIOS.

The Cx state ends due to a break event. Based on the break event, the processor returns the system to C0. The following are examples of such break events:

- Any unmasked interrupt goes active
- Any internal event that will cause an NMI or SMI_B
- CPU Pending Break Event (PBE_B)
- MSI

Figure 8. Idle Power Management Breakdown of Processor Cores



7.3.4 Processor Core C-States Description

Table 53. Processor Core/ States Support

State	Description
C0	Active mode, processor executing code
C1	AutoHALT state
C1E	AutoHALT State with lowest frequency and voltage operating point.
C6	Deep Power Down. Prior to entering the Deep Power Down Technology (code named C6) State, The core process will flush its cache and save its core context to a special on die SRAM on a different power plane. Once Deep Power Down Technology (code named C6) sequence has completed. The core processor’s voltage is completely shut off.
C7	Execution cores in this state behave similarly to the C6 state. Voltage is removed from the system agent domain

The following state descriptions assume that both threads are in common low power state.

7.3.4.1 Core C0 State

The normal operating state of a core where code is being executed.

7.3.4.2 Core C1/C1E State

C1/C1E is a low power state entered when a core execute a HLT or MWAIT(C1/C1E) instruction.

A System Management Interrupt (SMI) handler returns execution to either Normal state or the C1/C1E state. Refer *Intel® 64 and IA-32 Architecture Software Developer's Manual, Volume 3A/3B: System Programmer's Guide* for more information.

While a core is in C1/C1E state, it processes bus snoops and snoops from other threads. For more information on C1E, refer Section 7.3.8.2, "Package C1/C1E".

7.3.4.3 Core C6 State

Individual core can enter the C6 state by initiating a P_LVL3 I/O read or an MWAIT(C6) instruction. Before entering core C6, the core will save its architectural state to a dedicated SRAM.Once complete, a core will have its voltage reduced. During exit, the core is powered on and its architectural state is restored.

7.3.4.4 Core C7 State

Individual core can enter the C7 state by initiating a P_LVL7 I/O read or an MWAIT(C7) instruction. The core C7 state exhibits the same behavior as core C6 state, but in addition gives permission to the internal Power Management logic to enter a package S0ix state if possible.

7.3.4.5 C-State Auto-Demotion

In general, deeper C-states, such as C6 or C7, have long latencies and higher energy entry/exit costs. The resulting performance and energy penalties become significant when the entry/exit frequency of a deeper C-state is high. Therefore incorrect or inefficient usage of deeper C-states has a negative impact on battery life. In order to increase residency and improve battery life in deeper C-states, the processor supports C-state auto-demotion.

This is the C-State auto-demotion option:

- C7/C6 to C1

The decision to demote a core from C7/C6 to C1 is based on each core's immediate residency history. Upon each core C7/C6 request, the core C-state is demoted to C1 until a sufficient amount of residency has been established. At that point, a core is allowed to go into C6 or C7.

This feature is disabled by default. BIOS must enable it in the PMG_CST_CONFIG_CONTROL register. The auto-demotion policy is also configured by this register.

7.3.5 Module C-states

Table 54. Module C-states

C-states	Core Status	Cache status
C0	At least one core in C0	Normal Operation
C1	Both cores HALTed. Most clocks OFF	No Cache flushed; Snoops wake up cores
C6	Both cores in C6 (powered off) CPLL bypassed (powered off) CPU Refclk OFF BIU domain powered off	Core DL1s flushed L2 flushed L2 domain powered off C2 popup NOT required

7.3.6 Module C6

There are two module C-states the Punit can put a CPU module into depending on the type of C-state entry sub-state hint and remaining size of L2. In this module C-state, both cores are power gated and all ways of L2 cache can be flushed. In this state, the Punit can power gate the BIU/L2 Vcc domain as well as the VCCSRAM_GT domain.

7.3.7 S0i1

Once the core has entered package C6 or C7, the SoC can transition to S0i1. S0i1 transitions from a PC6 means that L2 state will be preserved through S0i1. Transitions from C7 no longer have state retention. These two paths are quite different due to the requirements on the L2 power rails and the need to snoop the core.

7.3.8 Package C-States*

The processor supports C0, C1/C1E,C6 and C7 power states. The following is a summary of the general rules for package C-state entry. These apply to all package C-states unless specified otherwise:

- Package C-state request is determined by the lowest numerical core C-state amongst all cores.
- A package C-state is automatically resolved by the processor depending on the core idle power states and the status of the platform components.
- Each core can be at a lower idle power state than the package if the platform does not grant the processor permission to enter a requested package C-state.
- The platform may allow additional power savings to be realized in the processor.
- For package C-states, the processor is not required to enter C0 before entering any other C-state.

The processor exits a package C-state when a break event is detected.

7.3.8.1 Package C0

The normal operating state for the processor. The processor remains in the normal state when at least one of its cores is in the C0 state or when the platform has not granted permission to the processor to go into a low power state. Individual cores may be in lower power idle states while the package is in C0.

7.3.8.2 Package C1/C1E

No additional power reduction actions are taken in the package C1 state. However, if the C1E sub-state is enabled, the processor automatically transitions to the lowest supported core clock frequency, followed by a reduction in voltage.

The package enters the C1 low power state when:

- At least one core is in the C1 state.
- The other cores are in a C1 or lower power state.

The package enters the C1E state when:

- All cores have directly requested C1E via MWAIT(C1) with a C1E sub-state hint.
- All cores are in a power state lower that C1/C1E but the package low power state is limited to C1/C1E via the PMG_CST_CONFIG_CONTROL MSR.
- All cores have requested C1 using HLT or MWAIT(C1) and C1E auto-promotion is enabled in IA32_MISC_ENABLES.

No notification to the system occurs upon entry to C1/C1E.

7.3.8.3 Package C6 State

A processor enters the package C6 low power state when:

- At least one core is in the C6 state.
- The other cores are in a C6 or lower power state, and the processor has been granted permission by the platform.
- The platform has not granted a request to a package C7 state but has allowed a package C6 state.

In package C6 state, all cores have saved their architectural state and have had their core voltages reduced to zero volts.

7.3.8.4 Package C7 State

A processor enters the package C7 low power state when all cores are in the C7 state. In package C7, the processor will take action to remove power from portions of the system agent.

Core break events are handled the same way as in package C6.

7.3.9 Graphics, Video and Display Power Management

7.3.9.1 Graphics and video decoder C-State

GFX C-State (GC6) are designed to optimize the average power to the graphics and video decoder engines during times of idleness. GFX C-state is entered when the graphics engine, has no workload being currently worked on and no outstanding graphics memory transactions. When the idleness condition is met, the processor will power gate the Graphics and video decoder engines.

7.3.9.2 Intel® Display Power Saving Technology (Intel® DPST)

The Intel DPST technique achieves backlight power savings while maintaining visual experience. This is accomplished by adaptively enhancing the displayed image while decreasing the backlight brightness simultaneously. The goal of this technique is to provide equivalent end-user image quality at a decreased backlight power level.

1. The original (input) image produced by the operating system or application is analyzed by the Intel DPST subsystem. An interrupt to Intel® DPST software is generated whenever a meaningful change in the image attributes is detected. (A meaningful change is when the Intel DPST software algorithm determines that enough brightness, contrast, or color change has occurred to the displaying images that the image enhancement and backlight control needs to be altered.)
2. Intel DPST subsystem applies an image-specific enhancement to increase image contrast, brightness, and other attributes.
3. A corresponding decrease to the backlight brightness is applied simultaneously to produce an image with similar user-perceived quality (such as brightness) as the original image. Intel DPST 6.0 has improved the software algorithms and has minor hardware changes to better handle backlight phase-in and ensures the documented and validated method to interrupt hardware phase-in.

7.3.9.3 Intel® Automatic Display Brightness

The Intel Automatic Display Brightness feature dynamically adjusts the backlight brightness based upon the current ambient light environment. This feature requires an additional sensor to be on the panel front. The sensor receives the changing ambient light conditions and sends the interrupts to the Intel Graphics driver. As per the change in Lux, (current ambient light illuminance), the new backlight setting can be adjusted through BLC. The converse applies for a brightly lit environment. Intel Automatic Display Brightness increases the back light setting.

7.3.9.4 Intel® Seamless Display Refresh Rate Switching Technology (Intel® SDRRS Technology)

When a Local Flat Panel (LFP) supports multiple refresh rates, the Intel® Display Refresh Rate Switching power conservation feature can be enabled. The higher refresh rate will be used when on plugged in power or when the end user has not selected/

enabled this feature. The graphics software will automatically switch to a lower refresh rate for maximum battery life when the design application is on battery power and when the user has selected/enabled this feature.

There are two distinct implementations of Intel SDRRS—static and seamless. The static Intel SDRRS method uses a mode change to assign the new refresh rate. The seamless Intel SDRRS method is able to accomplish the refresh rate assignment without a mode change and therefore does not experience some of the visual artifacts associated with the mode change (SetMode) method.

7.4 Memory Power Management

The main memory is power managed during normal operation and in low-power states.

7.4.1 Disabling Unused System Memory Outputs

Any system memory (SM) interface signal that goes to a memory module connector in which it is not connected to any actual memory devices (such as unpopulated, or single-sided) is tri-stated. The benefits of disabling unused SM signals are:

- Reduced power consumption.
- Reduced possible overshoot/undershoot signal quality issues seen by the processor I/O buffer receivers caused by reflections from potentially un-terminated transmission lines.

When a given rank is not populated, the corresponding chip select and CKE signals are not driven.

SCKE tri-state should be enabled by BIOS where appropriate, since at reset all rows must be assumed to be populated.

7.4.2 DRAM Power Management and Initialization

The processor implements extensive support for power management on the SDRAM interface. There are four SDRAM operations associated with the Clock Enable (CKE) signals, which the SDRAM controller supports. The processor drives four CKE pins to perform these operations.

7.4.2.1 Initialization Role of CKE*

During power-up, CKE is the only input to the SDRAM that is recognized (other than the DDR3 reset pin) once power is applied. It must be driven LOW by the DDR controller to make sure the SDRAM components float DQ and DQS during power- up.

CKE signals remain LOW (while any reset is active) until the BIOS writes to a configuration register. Using this method, CKE is guaranteed to remain inactive for much longer than the specified 200 micro-seconds after power and clocks to SDRAM devices are stable.

7.4.2.2 Conditional Self-Refresh

Intel Rapid Memory Power Management (Intel RMPM) conditionally places memory into self-refresh in the package low-power states. RMPM functionality depends on graphics/ display state (relevant only when internal graphics is being used), as well as memory traffic patterns generated by other connected I/O devices.

When entering the Suspend-to-RAM (STR) state, the processor core flushes pending cycles and then places all SDRAM ranks into self refresh. In STR, the CKE signals remain LOW so the SDRAM devices perform self-refresh.

The target behavior is to enter self-refresh for the package low-power states as long as there are no memory requests to service.

7.4.2.3 Dynamic Power Down Operation

Dynamic power-down of memory is employed during normal operation. Based on idle conditions, a given memory rank may be powered down. The IMC implements aggressive CKE control to dynamically put the DRAM devices in a power down state.The processor core controller can be configured to put the devices in active power down (CKE deassertion with open pages) or precharge power down (CKE deassertion with all pages closed). Precharge power down provides greater power savings but has a bigger performance impact, since all pages will first be closed before putting the devices in power down mode.

If dynamic power-down is enabled, all ranks are powered up before doing a refresh cycle and all ranks are powered down at the end of refresh.

7.4.2.4 DRAM I/O Power Management

Unused signals should be disabled to save power and reduce electromagnetic interference. This includes all signals associated with an unused memory channel.

The I/O buffer for an unused signal should be tri-stated (output driver disabled), the input receiver (differential sense-amp) should be disabled, and any DLL circuitry related ONLY to unused signals should be disabled. The input path must be gated to prevent spurious results due to noise on the unused signals (typically handled automatically when input receiver is disabled).

§

8 System Memory Controller

The system memory controller supports DDR3L-RS/LPDDR3 protocol with up to two 64-bit wide dual rank channels at data rates up to 1600 MT/s with ECC is also available on a single DDR3L-RS channel.

8.1 Signal Descriptions

Refer Chapter 2, "Physical Interfaces" for additional details.

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin.
- **Direction**: The buffer direction can be either input, output, or I/O (bidirectional).
- **Type:** The buffer type found in Chapter 19, "Electrical Specifications".
- **Description**: A brief explanation of the signal's function.

8.1.1 DDR3L-RS Interface Signals

Table 55. Memory Channel 0 DDR3L-RS Signals (Sheet 1 of 2)

Signal Name	Direction Type	Description
DDR3_M0_CK[1,0]_P **DDR3_M0_CK[1,0]_N**	O DDR3	**Clock PAD:** (1 pair per Rank) Driven by PHY to DRAM.
DDR3_M0_CS[1,0]_N	O DDR3	**Chip Select:** (1 per Rank). Driven by PHY to DRAM.
DDR3_M0_CKE[3,0]	O DDR3	**Clock Enable:** (power management) Driven by PHY to DRAM.
DDR3_M0_MA[15:0]	O DDR3	**Memory Address:** Driven by PHY to DRAM.
DDR3_M0_BS[2:0]	O DDR3	**Bank Select:** Driven by PHY to DRAM.
DDR3_M0_RAS_N	O DDR3	**Row Address Select:** Used with DDR3_M0_CAS# and DDR3_M0_WE# (along with DDR3_M0_CS#) to define the DRAM Commands
DDR3_M0_CAS_N	O DDR3	**Column Address Select:** Used with DDR3_M0_RAS# and DDR3_M0_WE# (along with DDR3_M0_CS#) to define the SRAM Commands
DDR3_M0_WE_N	O DDR3	**Write Enable Control Signal:** Used with DDR3_M0_WE# and DDR3_M0_CAS# (along with control signal, DDR3_M0_CS#) to define the DRAM Commands.

Table 55. Memory Channel 0 DDR3L-RS Signals (Sheet 2 of 2)

Signal Name	Direction Type	Description
DDR3_M0_DQ[63:0]	I/O DDR3	**Data Lines:** Bidirectional signals between DRAM/ PHY
DDR3_M0_DM[7:0]	O DDR3	**Data Mask:** DM is an output mask signal for write data. Output data is masked when DM is sampled HIGH coincident with that output data during a Write access. DM is sampled on both edges of DQS.
DDR3_M0_DQS[7:0]_P **DDR3_M0_DQS[7:0]_N**	I/O DDR3	**Data Strobes:** The data is captured at the crossing point of each 'P' and its compliment 'N' during read and write transactions. For reads, the strobe crossover and data are edge aligned, whereas in the Write command, the strobe crossing is in the centre of the data window.
DDR3_M0_ODT[1,0]	O DDR3	**On Die Termination:** ODT signal going to DRAM in order to turn ON the DRAM ODT during Write.
DDR3_M0_RCOMPD	I DDR	**Resistor Compensation:** This signal needs to be terminated to VSS on board.This signal is driven from external clock source.
DDR3_M0_OCAVREF	O DDR	**Reference Voltage:** DDR3 CA interface Reference Voltage
DDR3_M0_ODQVREF	O DDR	**Reference Voltage:** DDR3 DQ interface Reference Voltage
DDR3_CORE_PWROK	I DDR	**Core Power OK:** This signal indicates the status of the DRAM Core power supply (power on in S0).Active high signal indicates that DDR PHY voltage(1.5v) is good.
DDR3_VDD_S4_PWROK	I DDR	**VDD Power OK:** Asserted once the VRM is settled.
DDR3_M0_DRAMRST_N	O	**DRAM Reset:** This signal is used to reset DRAM devices.

Table 56. Memory Channel 1 DDR3L-RS Signals (Sheet 1 of 2)

Signal Name	Direction Type	Description
DDR3_M1_CK[1,0]_P **DDR3_M1_CK[1,0]_N**	O DDR3	**Clock PAD:** (1 pair per Rank) Driven by PHY to DRAM.
DDR3_M1_CS[1,0]_N	O DDR3	**Chip Select:** (1 per Rank). Driven by PHY to DRAM.
DDR3_M1_CKE[3,0]	O DDR3	**Clock Enable:** (power management) Driven by PHY to DRAM.
DDR3_M1_MA[15:0]	O DDR3	**Memory Address:** Driven by PHY to DRAM.
DDR3_M1_BS[2:0]	O DDR3	**Bank Select:** Driven by PHY to DRAM.

Table 56. Memory Channel 1 DDR3L-RS Signals (Sheet 2 of 2)

Signal Name	Direction Type	Description
DDR3_M1_RAS_N	O DDR3	**Row Address Select:** Used with DDR3_M0_CAS# and DDR3_M0_WE# (along with DDR3_M0_CS#) to define the DRAM Commands
DDR3_M1_CAS_N	O DDR3	**Column Address Select:** Used with DDR3_M0_RAS# and DDR3_M0_WE# (along with DDR3_M0_CS#) to define the SRAM Commands
DDR3_M1_WE_N	O DDR3	**Write Enable Control Signal:** Used with DDR3_M0_WE# and DDR3_M0_CAS# (along with control signal, DDR3_M0_CS#) to define the DRAM Commands.
DDR3_M1_DQ[63:0]	I/O DDR3	**Data Lines:** Bidirectional signals between DRAM/ PHY
DDR3_M1_DM[7:0]	O DDR3	**Data Mask:** DM is an output mask signal for write data. Output data is masked when DM is sampled HIGH coincident with that output data during a Write access. DM is sampled on both edges of DQS.
DDR3_M1_DQS[7:0]_P **DDR3_M1_DQS[7:0]_N**	I/O DDR3	**Data Strobes:** The data is captured at the crossing point of DDR3_M1_DQSP[7:0] and its compliment 'N' during read and write transactions. For reads, the strobe crossover and data are edge aligned, whereas in the Write command, the strobe crossing is in the centre of the data window.
DDR3_M1_ODT[1,0]	O DDR3	**On Die Termination:** ODT signal going to DRAM in order to turn ON the DRAM ODT during Write.
DDR3_M1_DRAMRST_N	O	**Reset DRAM:** This signal can be used to reset DRAM devices.

8.1.2 DDR3L-RS Interface Signals

Table 57. Memory Channel 0 DDR3L-RS Signals (Sheet 1 of 2)

Signal Name	Direction Type	Description
DDR3_M0_CK[1,0]_P **DDR3_M0_CK[1,0]_N**	O DDR3	**Clock PAD:** (1 pair per Rank) Driven by PHY to DRAM.
DDR3_M0_CS[1,0]_N	O DDR3	**Chip Select:** (1 per Rank). Driven by PHY to DRAM.
DDR3_M0_CKE[3,0]	O DDR3	**Clock Enable:** (power management) Driven by PHY to DRAM.
DDR3_M0_MA[15:0]	O DDR3	**Memory Address:** Driven by PHY to DRAM.
DDR3_M0_BS[2:0]	O DDR3	**Bank Select:** Driven by PHY to DRAM.
DDR3_M0_RAS_N	O DDR3	**Row Address Select:** Used with DDR3_M0_CAS# and DDR3_M0_WE# (along with DDR3_M0_CS#) to define the DRAM Commands

Table 57. Memory Channel 0 DDR3L-RS Signals (Sheet 2 of 2)

Signal Name	Direction Type	Description
DDR3_M0_CAS_N	O DDR3	**Column Address Select:** Used with DDR3_M0_RAS# and DDR3_M0_WE# (along with DDR3_M0_CS#) to define the SRAM Commands
DDR3_M0_WE_N	O DDR3	**Write Enable Control Signal:** Used with DDR3_M0_WE# and DDR3_M0_CAS# (along with control signal, DDR3_M0_CS#) to define the DRAM Commands.
DDR3_M0_DQ[63:0]	I/O DDR3	**Data Lines:** Bidirectional signals between DRAM/ PHY
DDR3_M0_DM[7:0]	O DDR3	**Data Mask:** DM is an output mask signal for write data. Output data is masked when DM is sampled HIGH coincident with that output data during a Write access. DM is sampled on both edges of DQS.
DDR3_M0_DQS[7:0]_P **DDR3_M0_DQS[7:0]_N**	I/O DDR3	**Data Strobes:** The data is captured at the crossing point of each 'P' and its compliment 'N' during read and write transactions. For reads, the strobe crossover and data are edge aligned, whereas in the Write command, the strobe crossing is in the centre of the data window.
DDR3_M0_ODT[1,0]	O DDR3	**On Die Termination:** ODT signal going to DRAM in order to turn ON the DRAM ODT during Write.
DDR3_M0_RCOMPD	I DDR	**Resistor Compensation:** This signal needs to be terminated to VSS on board.This signal is driven from external clock source.
DDR3_M0_OCAVREF	O DDR	**Reference Voltage:** DDR3 CA interface Reference Voltage
DDR3_M0_ODQVREF	O DDR	**Reference Voltage:** DDR3 DQ interface Reference Voltage
DDR3_CORE_PWROK	I DDR	**Core Power OK:** This signal indicates the status of the DRAM Core power supply (power on in S0).Active high signal indicates that DDR PHY voltage(1.5v) is good.
DDR3_VDD_S4_PWROK	I DDR	**VDD Power OK:** Asserted once the VRM is settled.
DDR3_M0_DRAMRST_N	O	**DRAM Reset:** This signal is used to reset DRAM devices.

Table 58. Memory Channel 1 DDR3L-RS Signals

Signal Name	Direction Type	Description
DDR3_M1_CK[1,0]_P **DDR3_M1_CK[1,0]_N**	O DDR3	**Clock PAD:** (1 pair per Rank) Driven by PHY to DRAM.
DDR3_M1_CS[1,0]_N	O DDR3	**Chip Select:** (1 per Rank). Driven by PHY to DRAM.
DDR3_M1_CKE[3,0]	O DDR3	**Clock Enable:** (power management) Driven by PHY to DRAM.
DDR3_M1_MA[15:0]	O DDR3	**Memory Address:** Driven by PHY to DRAM.
DDR3_M1_BS[2:0]	O DDR3	**Bank Select:** Driven by PHY to DRAM.
DDR3_M1_RAS_N	O DDR3	**Row Address Select:** Used with DDR3_M0_CAS# and DDR3_M0_WE# (along with DDR3_M0_CS#) to define the DRAM Commands
DDR3_M1_CAS_N	O DDR3	**Column Address Select:** Used with DDR3_M0_RAS# and DDR3_M0_WE# (along with DDR3_M0_CS#) to define the SRAM Commands
DDR3_M1_WE_N	O DDR3	**Write Enable Control Signal:** Used with DDR3_M0_WE# and DDR3_M0_CAS# (along with control signal, DDR3_M0_CS#) to define the DRAM Commands.
DDR3_M1_DQ[63:0]	I/O DDR3	**Data Lines:** Bidirectional signals between DRAM/ PHY
DDR3_M1_DM[7:0]	O DDR3	**Data Mask:** DM is an output mask signal for write data. Output data is masked when DM is sampled HIGH coincident with that output data during a Write access. DM is sampled on both edges of DQS.
DDR3_M1_DQS[7:0]_P **DDR3_M1_DQS[7:0]_N**	I/O DDR3	**Data Strobes:** The data is captured at the crossing point of DDR3_M1_DQSP[7:0] and its compliment 'N' during read and write transactions. For reads, the strobe crossover and data are edge aligned, whereas in the Write command, the strobe crossing is in the centre of the data window.
DDR3_M1_ODT[1,0]	O DDR3	**On Die Termination:** ODT signal going to DRAM in order to turn ON the DRAM ODT during Write.
DDR3_M1_DRAMRST_N	O	**Reset DRAM:** This signal can be used to reset DRAM devices.

8.1.3 LPDDR3 Interface Signals

Table 59. Memory Channel 0 LPDDR3 Signals

Signal Name	Direction Type	Description
LPDDR3_M0_CK_P_A/B **LPDDR3_M0_CK_N_A/B**	O DDR3	**SDRAM and inverted Differential Clock:** (1 pair per Rank) The differential clock pair is used to latch the command into DRAM. Each pair corresponds to one rank on DRAM side.
LPDDR3_M0_CS[1,0]_N	O DDR3	**Chip Select:** (1 per Rank). Used to qualify the command on the command bus for a particular rank.
LPDDR3_M0_CKE[1,0]_A/B	O DDR3	**Clock Enable:** (power management) It is used during DRAM power up/power down and Self refresh. **NOTE:** LPDDR3 uses only LPDDR3_M0_CKE[2,0]. LPDDR3_M0_CKE[1,3] are not being used for LPDDR3.
LPDDR3_M0_CA[9:0]	O DDR3	**Memory Address:** Memory address bus for writing data to memory and reading data from memory. These signals follow common clock protocol w.r.t. LPDDR3_M0_CKN, LPDDR3_M0_CKP pairs
LPDDR3_M0_DQ[31:0]_A/B	I/O DDR3	**Data Lines:** Data signal interface to the DRAM data bus
LPDDR3_M0_DM[3:0]_A/B	O DDR3	**Data Mask:** DM is an output mask signal for write data. Output data is masked when DM is sampled HIGH coincident with that output data during a Write access. DM is sampled on both edges of DQS.
LPDDR3_M0_DQS[3:0]_P_A/B **LPDDR3_M0_DQS[3:0]_N_A/B**	I/O DDR3	**Data Strobes:** The data is captured at the crossing point of each 'P' and its compliment 'N' during read and write transactions. For reads, the strobe crossover and data are edge aligned, whereas in the Write command, the strobe crossing is in the centre of the data window.
LPDDR3_M0_ODT_A/B	O DDR3	**On Die Termination:** ODT signal going to DRAM in order to turn ON the DRAM ODT during Write.
LPDDR3_M0_RCOMPD	I DDR	**Resistor Compensation:** This signal needs to be terminated to VSS on board. This signal is driven from external clock source.
LPDDR3_M0_OCAVREF	O DDR	**Reference Voltage:** LPLPDDR3 CA interface Reference Voltage
LPDDR3_M0_ODQVREF	O DDR	**Reference Voltage:** LPLPDDR3 DQ interface Reference Voltage

Table 60. Memory Channel 1 LPDDR3 Signals

Signal Name	Direction Type	Description
LPDDR3_M1_CK_P_A/B **LPDDR3_M1_CK_N_A/B**	O DDR3	**SDRAM and inverted Differential Clock:** (1 pair per Rank) The differential clock pair is used to latch the command into DRAM. Each pair corresponds to one rank on DRAM side.
LPDDR3_M1_CS[1,0]_N	O DDR3	**Chip Select:** (1 per Rank). Used to qualify the command on the command bus for a particular rank.
LPDDR3_M1_CKE[1,0]_A/B	O DDR3	**Clock Enable:** (power management) It is used during DRAM power up/power down and Self refresh. **NOTE:** LPDDR3 uses only LPDDR3_M1_CKE[0,2]. LPDDR3_M1_CKE[1,3] are not being used for LPDDR3.
LPDDR3_M1_CA[9:0]	O DDR3	**Memory Address:** Memory address bus for writing data to memory and reading data from memory. These signals follow common clock protocol relative to LPDDR3_M1_CKN, LPDDR3_M1_CKP pairs
LPDDR3_M1_DQ[31:0]_A/B	I/O DDR3	**Data Lines:** Data signal interface to the DRAM data bus.
LPDDR3_M1_DM[3:0]_A/B	O DDR3	**Data Mask:** DM is an output mask signal for write data. Output data is masked when DM is sampled HIGH coincident with that output data during a Write access. DM is sampled on both edges of DQS.
LPDDR3_M1_DQS[3:0]_P_A/B **LPDDR3_M1_DQS[3:0]_N_A/B**	I/O DDR3	**Data Strobes:** The data is captured at the crossing point of LPDDR3_M1_DQSP[7:0] and its compliment 'N' during read and write transactions. For reads, the strobe crossover and data are edge aligned, whereas in the Write command, the strobe crossing is in the centre of the data window.
LPDDR3_M1_ODT_A/B	O DDR3	**On Die Termination:** ODT signal going to DRAM in order to turn ON the DRAM ODT during Write.
LPDDR3_M1_OCAVREF	O DDR	**Reference Voltage:** LPDDR3 CA interface Reference Voltage
LPDDR3_M1_ODQVREF	O DDR	**Reference Voltage:** LPDDR3 DQ interface Reference Voltage

8.1.4 ECC Support

The system memory controller supports ECC. When ECC is enabled, only Memory Channel 0 will be active. Memory Channel 1 will be disabled and used for the ECC data pins. The table below shows the details on the muxing relationship between the ECC Signals and the Memory Channel 1 signals.

Note: ECC SO-DIMMs are not backwards compatible with non-ECC SO-DIMMs.

Table 62. ECC Signals

Signal Name	Direction Type	Description
DDR3_M0_ECC_DQ[7:0]	I/O DDR3	ECC Check Data Bits These are muxed with channel 1.
DDR3_M0_ECC_DM	O DDR3	**ECC Data Mask:** DM is an optional output mask signal for write data. Output data is masked when DM is sampled HIGH coincident with that output data during a Write access. DM is sampled on both edges of ECC_DQS. This signal is muxed with channel 1 and may not be needed.
DDR3_M0_ECC_DQSP **DDR3_M0_ECC_DQSN**	I/O DDR3	**ECC Data Strobes:** The data is captured at the crossing point the 'P' and its compliment 'N' during read and write transactions. For reads, the strobe crossover and data are edge aligned, whereas in the Write command, the strobe crossing is in the centre of the data window. These are muxed with channel 1.

8.2 Features

8.2.1 System Memory Technology Supported

The system memory controller supports the following DDR3L-RS/LPDDR3 Data Transfer Rates, DRAM Device Technologies:

- DDR3L-RS/LPDDR3 Data Transfer Rates: 1600MT/s (12.8 GB/s per channel).
- LPDDR3 (1.2V DRAM VDDQ)
- DDR3L-RS (1.35V DRAM interface I/Os)
- DDR3L-RS DRAM Device Technology
  - Standard 2 Gb technologies and addressing
  - Read latency 5, 6, 7, 8, 9, 10, 11, 12, 13
  - Write latency 3, 4, 5, 6, 7, 8
- LPDDR3 DRAM Device Technology
  - x64, 253 ball LPDDR3 DRAM package
  - 8 GB (2 rank per channel) package density
  - Standard 2 Gb, 4 Gb and 8 Gb DRAM technologies and addressing
  - Read latency 5, 6, 7, 8, 9, 10, 11, 12, 13
  - Write latency 3, 4, 5, 6, 7, 8
- Supports Trunk Clock Gating
- ECC supports 64-bit data bus on DDR3L-RS single channel
- Supports early SR exit
- Supports slow power down
- Supports CA tri-state when not driving a valid command

Table 63. Supported LPDDR3 DRAM Devices

DRAM Density	Data Width	Banks	Bank Address	Row Address	Column Address	Page Size
2Gb	x32	8	BA[2:0]	A[13:0]	A[8:0]	2KB
4Gb	x32	8	BA[2:0]	A[13:0]	A[9:0]	4KB
8Gb	x32	8	BA[2:0]	A[14:0]	A[9:0]	4KB

Table 64. Supported DDR3L-RS DRAM Devices

DRAM Density	Data Width	Banks	Bank Address	Row Address	Column Address	Page Size
2Gb	x16	8	BA[2:0]	A[13:0]	A[9:0]	2KB

Table 65. Supported DDR3L-RS DRAM Devices

DRAM Density	Data Width	Banks	Bank Address	Row Address	Column Address	Page Size
2Gb	x16	8	BA[2:0]	A[13:0]	A[9:0]	2KB

Table 66. Supported LPDDR3 Memory Size Per Rank

Memory Size/ Rank	DRAM Chips/ Rank	DRAM Chip Density	DRAM Chip Data Width	Page Size @ 64-bit Data Bus
512MB	2	2Gb	x32	4KB = 2KB * 2 chips
1GB	2	4Gb	x32	8KB = 4KB * 2 chips
2GB	2	8Gb	x32	8KB = 4KB * 2 chips

Table 67. Supported DDR3L-RS Memory Size Per Rank

Memory Size/ Rank	DRAM Chips/ Rank	DRAM Chip Density	DRAM Chip Data Width	Page Size @ 64-bit Data Bus
1GB	4	2Gb	x16	8KB = 2KB * 4 chips

8.3 Register Map

For more information on System Memory Controller registers, refer Intel® Atom™ Z8000 Processor Series Datasheet (Volume 2 of 2), Doc ID:332066.

§

9 Graphics, Video and Display

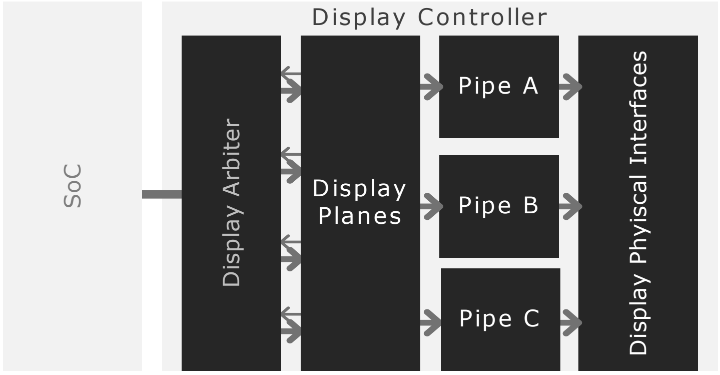
This chapter provides an overview of Graphics, Video and Display features of the SoC.

9.1 Features

The key features of the individual blocks are as follows:

- Refreshed eight generation Intel graphics core with sixteen Execution Units (EUs)
  - 3D graphics hardware acceleration including support for DirectX*11.1, OpenGL 4.3, OGL ES 3.0, OpenCL 1.2.
  - Video decode hardware acceleration including support for H.263, MPEG4, H.264, H.265 (HEVC), VP8, VP9, MVC, MPEG2, VC1, JPEG.
  - Video encode hardware acceleration including support for H.264, H.263, VP8, MVC, JPEG.
  - Display controller, incorporating the display planes, pipes and physical interfaces.
  - Four planes available per pipe - 1x Primary, 2x Video Sprite & 1x Cursor.
  - Three multi-purpose Digital Display Interface (DDI) PHYs implementing HDMI, DVI, DisplayPort (DP) or Embedded DisplayPort (eDP) support.
  - Two dedicated digital Display Serial Interface PHYs implementing MIPI-DSI support.

9.2 SoC Graphics Display



The Processor Graphics controller display pipe can be broken down into three components:

- Display Planes
- Display Pipes
- Display Physical Interfaces

A display plane is a single displayed surface in memory and contains one image (desktop, cursor, overlay). It is the portion of the display hardware logic that defines the format and location of a rectangular region of memory that can be displayed on a display output device and delivers that data to a display pipe. This is clocked by the Core Display Clock.

9.2.1 Primary Planes A, B and C

Planes A, B and C are the main display planes and are associated with Pipes A, B and C respectively. Each plane supports per-pixel alpha blending.

9.2.2 Video Sprite Planes A, B, C, D, E and F

Video Sprite Planes A, B, C, D, E and F are planes optimized for video decode.

- Pipe A – Primary planeA, VSpriteA, VSpriteB, CusrorA
- Pipe B – Primary planeB, VSpriteC, VSpriteD, CursorB
- Pipe C – Primary planeC, VSpriteE, VSpriteF, CursorC

9.2.3 Cursors A, B and C

Cursors A, B and C are small, fixed-sized planes dedicated for mouse cursor acceleration, and are associated with Planes A, B and C respectively.

9.3 Display Pipes

The display pipe blends and synchronizes pixel data received from one or more display planes and adds the timing of the display output device upon which the image is displayed.

The display pipes A, B and C operate independently of each other at the rate of one pixel per clock. They can attach to any of the display interfaces.

9.4 Display Physical Interfaces

The display physical interfaces consist of output logic and pins that transmit the display data to the associated encoding logic and send the data to the display device. These interfaces are digital (MIPI-DSI, DisplayPort*, Embedded DisplayPort*, DVI and HDMI*) interfaces.

Figure 9. Display Pipe to Port Mapping

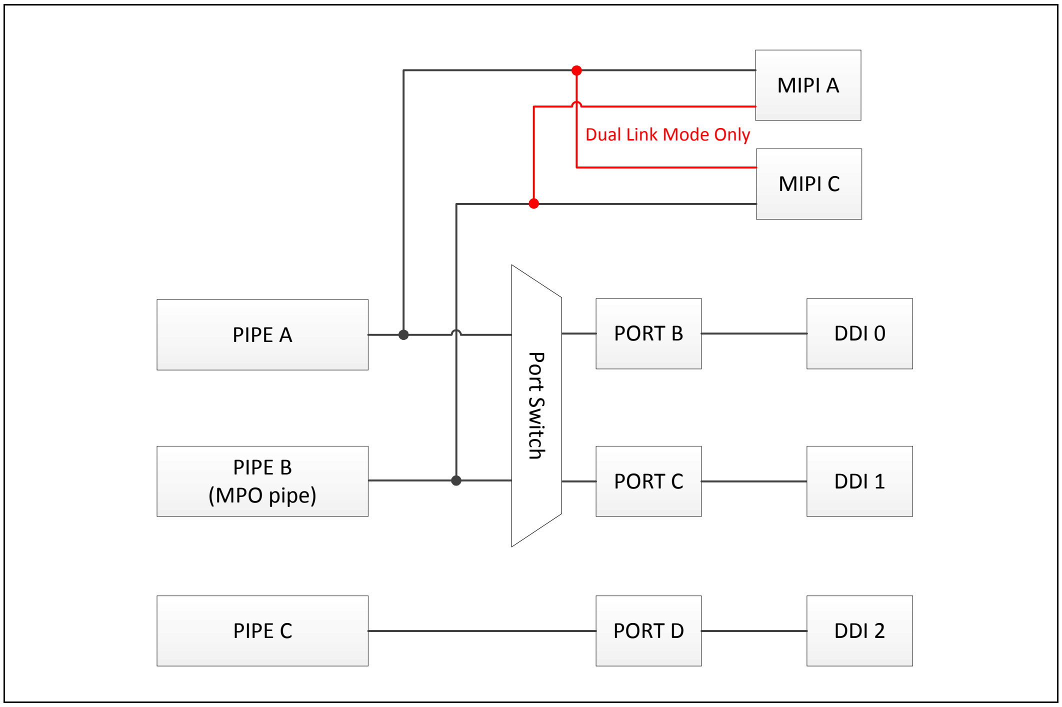


Figure 10. Display Pipe to Port Mapping [T3]

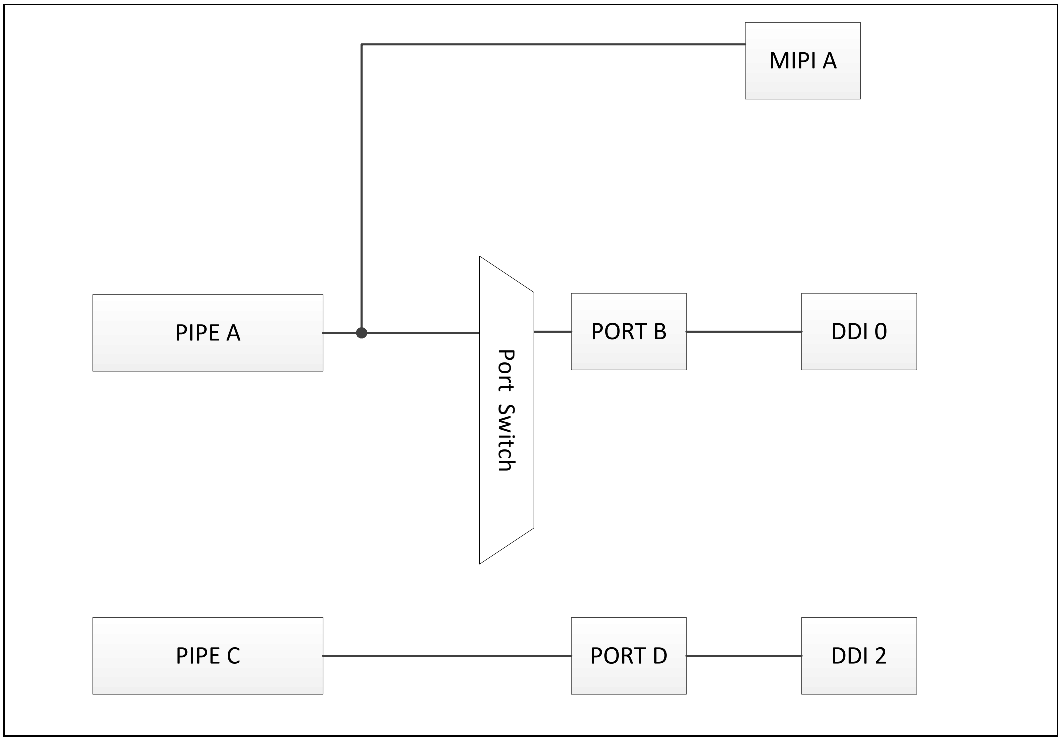
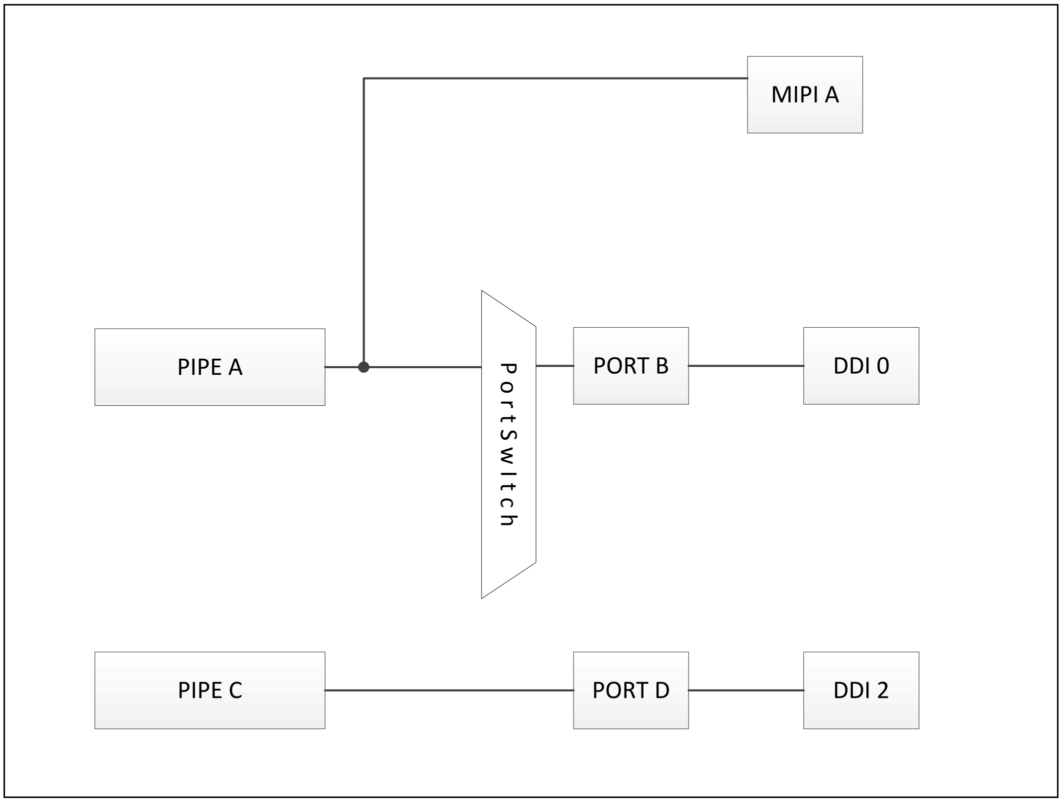


Figure 11. Display Pipe to Port Mapping [T3]



9.4.1 Digital Display Interfaces

Table 68. SoC Display Configuration (Sheet 1 of 2)

Feature	MIPI DSI	eDP	DP	HDMI/DVI
Number of Ports	2 (x4 @ 1Gbps)	2 (x4 @2.7Gbps)	2 (x4 @2.7GHz)	2 (x4 @2.97GHz)
Max Resolution	2560x1600 24bpp @60Hz	2560x1600 24bpp @60Hz	2560x1600 24bpp @60Hz	1920x1080 24bpp @120Hz/ 3840x2160 24bpp @ 30Hz
Standard	DSI1.01/ DPHY1.00	eDP1.3	DP1.1a	HDMI1.4b
Power gated during S0ix w/display off	Yes	Yes	Yes	Yes
DRRS (Refresh reduction)	Yes (M/N pair)	Yes (Panel command)	N/A	N/A
Self-Refresh with Frame buffer in Panel	Yes (Command Mode)	Yes (PSR)	No	No

Table 68. SoC Display Configuration (Sheet 2 of 2)

Feature	MIPI DSI	eDP	DP	HDMI/DVI
Content-Based backlight control	DPST6.0/CABC	DPST6/CABC	N/A	N/A
HDCP wired display	N/A	N/A(ASSR support)	1.4	1.4
HDCP wireless display	N/A	N/A(ASSR support)	2.2	2.2
PAVP	AES-encrypted buffer, plane control, panic attack			
SEC	All display registers can be accessed by CEC			
LPE Audio	N/A	N/A	Yes	Yes
Compressed Audio	N/A	N/A	Yes	Yes

Table 69. SoC Display supported Resolutions

	1 Display		2 Displays		3 Displays
	1 Internal	1 External	1 Internal +1 External	2 External	1 Internal +2 External
Internal #1	**eDP*** 2560x1600 @ 60Hz or **MIPI DSI*** 2560x1600 @60Hz	N/A	**eDP*** 2560x1600 @ 60Hz or **MIPI DSI*** 2560x1600 @60Hz	N/A	**eDP*** 2560x1600 @ 60Hz or **MIPI DSI*** 2560x1600 @60Hz
External #1	N/A	**HDMI/DP*** 3840x2160 @ 30Hz 2560x1600 @ 60Hz	**HDMI/DP*** 3840x2160 @ 30Hz 2560x1600 @ 60Hz	N/A	**HDMI/DP*** 3840x2160 @ 30Hz 2560x1600 @ 60Hz
External #2	N/A	N/A	N/A	**HDMI/DP*** 3840x2160 @ 30Hz 2560x1600 @ 60Hz	**HDMI/DP*** 3840x2160 @ 30Hz 2560x1600 @ 60Hz

NOTES:

1. SoC is supported maximum of 3 simultaneous displays. External display in both clone and extended modes.
2. Experience may differ based on configuration, resolution, and work loads.

9.4.1.1 Signal Descriptions

Refer Chapter 2, “Physical Interfaces” for additional details.

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin.
- **Direction**: The buffer direction can be either input, output, or I/O (bidirectional).
- **Type:** The buffer type found in Chapter 19, “Electrical Specifications”.
- **Description**: A brief explanation of the signal’s function.

Table 71. Display Physical Interfaces Signal Names (2 of 2)

Signal names	Direction Type	Description
MDSI_A_CLKP	O	MIPI Clock output for port A
MDSI_A_CLKN	O	MIPI Clock complement output for port A
MDSI_A_DP[3:0]	I/O	MIPI Data Lane 3:0 for port A
MDSI_A_DN[3:0]	I/O	MIPI Data Lane 3:0 complement for port A
MDSI_C_CLKP	O	MIPI Clock output for port C
MDSI_C_CLKN	O	MIPI Clock complement output for port C
MDSI_C_DP[3:0]	I/O	MIPI Data Lane 3:0 for port C
MDSI_C_DN[3:0]	I/O	MIPI Data Lane 3:0 complement for port C
MDSI_A_TE	I/O	Tearing Effect Signal from x4 port A display
MDSI_C_TE	I	Tearing Effect Signal from x4 port C display
MDSI_DDC_DATA	I/O	DDC Data
MDSI_DDC_CLK	I/O	DDC Clock
MDSI_RCOMP	I/O	MDSI_RCOMP: This is for pre-driver slew rate compensation for the MIPI DSI Interface. An external precision resistor of 150 Ω ±1% should be connected from this pin to ground.

9.4.1.2 Features

9.4.1.2.1 MIPI-DSI

Dual Link interface supports display resolution up to 2560 x 1600p @ 60 Hz with 24b per pixel.

Interface supports maximum of 1Gbps per lane.

Full Frame Buffer Panel

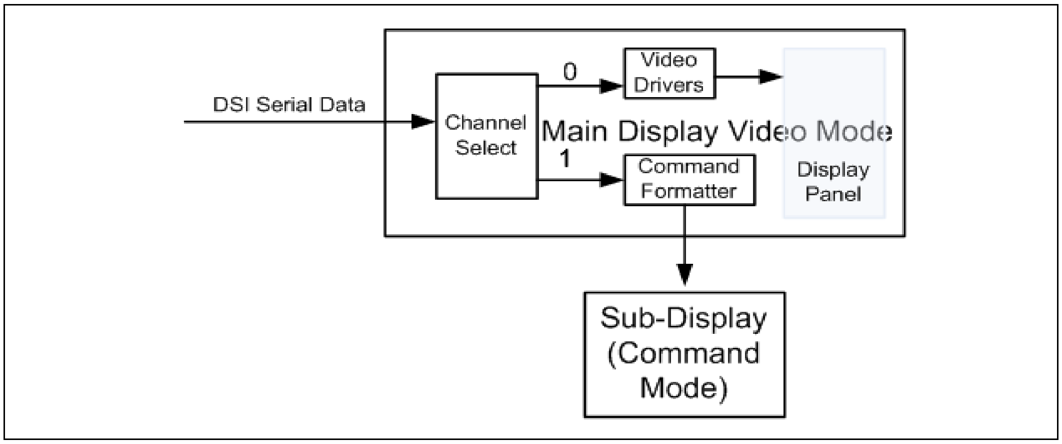
The display controller supports full frame buffer display (also called command-mode display) with optimizations for both SoC power consumption and system power consumption. Full frame buffer panel does not need to be refreshed regularly by a frame buffer in system memory so the path between panel and system memory can be power-managed as much as possible until a new request occurs to update one or more planes that are active in the display pipe.

Sub-Display Support

The display controller supports a sub-display panel that uses a different virtual channel and shares the same interface with the main panel. The pixel data for this sub-display can come from a direct system memory read or it can come from the output at the pipe as described. Sub-display allows, for example, the pixel stream to be updated more frequently or presented in a format and/or resolution that would require software to convert or scale the panel resolution.

One example usage of sub-display is as a view finder for camera. The camera interface unit may output images in a format and resolution that are not read by the sub-display itself or must be blended with camera application graphics.

Figure 12. Sub-Display Connection



Partial Display Mode Support

The display controller supports a partial display mode that utilizes the MIPI command set to transition the panel from normal mode to partial display mode, so a small part of the display panel can be kept active for pixel data. The same panel can switch from full screen mode to a sub-display mode with a handful of scan lines to show time, date, signal strength indicator, etc., to save power for the host processor and display panel.

There are two scenarios:

- Type 1 display panel—both full display and partial display operates in command mode.

- Type 2 display panel—full display (normal mode) operate in video mode; partial display operates in command mode. This requires the host processor and display panel to be in sync in transition from normal mode to partial mode after 2 frames from the enter_partial_mode command.

The software driver must implement most of the protocols of transition and send the correct commands to the display panel to start the transition. The software driver must program the display controller to select the buffer for partial display (display pipe output or system memory) and follow the protocol to be in sync with the display panel.

When the display transitions from partial mode to normal mode, it is recommended to turn the display off to avoid tearing effect as in a flow chart in DCS specification.

MIPI DSI Dual-link Mode

The SoC supports MIPI DSI dual-link mode, so that a single display can transmit a single stream of video data across two independent MIPI DSI interfaces. The packetization and timing of each link follows MIPI DSI 1.00 and DPHY 1.00 precisely, but the receiving device, which is a panel or a bridge, can combine the streaming data from two interfaces and display it in a single panel.

There are two types of dual-link panels that the SoC can support:

- Front-back type of panel, the first half of columns of pixels is always transmitted by port A and the second half of columns of pixels is always transmitted by port B.
- Pixel alternative type of panel, odd columns of pixels are always transmitted by port A and even columns of pixels are always transmitted by port B. So the 1st, 3rd, 5th, 7th, etc., pixels are separated at the source and sent in the first interface; the 2nd, 4th, 6th, 8th, etc., pixels are sent in the second interface.When the platform requires a dual-link interface for a large MIPI DSI panel or bridge (usually with resolution larger than 1920x1080 in which a 4-lane interface does not have enough bandwidth), the driver treats dual-link a special port configuration, with special handling of DSI controller but the operation of dual-link mode is consistent with single-link mode for planes and pipe operations. The system interface with upper level of SW does not need to change, like flip mechanism, interrupt, and so on.

LVDS Panel Support

An external MIPI DSI-to-LVDS bridge device is required to connect the display controller to a LVDS panel. A bridge device is used for larger panels.

9.4.1.3 High Definition Multimedia Interface

The High-Definition Multimedia Interface (HDMI) is provided for transmitting digital audio and video signals from DVD players, set-top boxes and other audiovisual sources to television sets, projectors and other video displays. It can carry high quality multi-channel audio data and all standard and high-definition consumer electronics video formats. HDMI display interface connecting the SoC and display devices utilizes transition minimized differential signaling (TMDS) to carry audiovisual information through the same HDMI cable.

HDMI includes three separate communications channels: TMDS, DDC, and the optional CEC (consumer electronics control) (not supported by the SoC). As shown in Figure 13, the HDMI cable carries four differential pairs that make up the TMDS data and clock channels. These channels are used to carry video, audio, and auxiliary data. In addition, HDMI carries a VESA DDC. The DDC is used by an HDMI Source to determine the capabilities and characteristics of the sink.

Audio, video and auxiliary (control/status) data is transmitted across the three TMDS data channels. The video pixel clock is transmitted on the TMDS clock channel and is used by the receiver for data recovery on the three data channels. The digital display data signals driven natively through the SoC are AC coupled and needs level shifting to convert the AC coupled signals to the HDMI compliant digital signals.

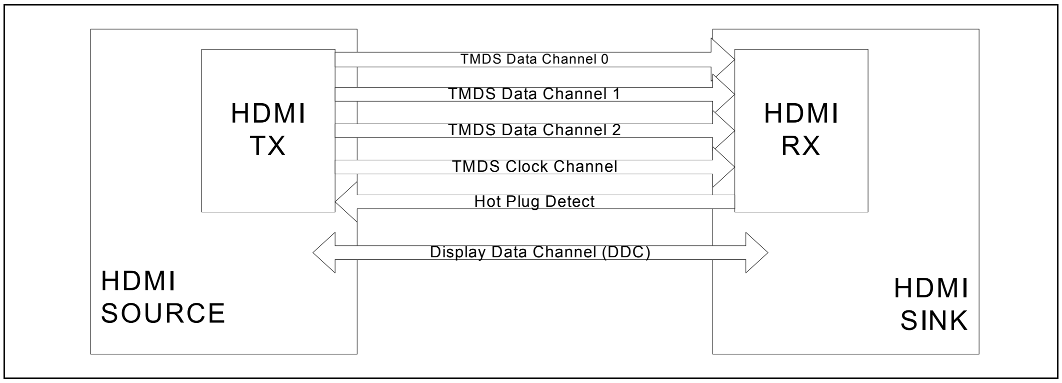
The SoC HDMI interface is designed as per the High-Definition Multimedia Interface Specification 1.4. The SoC supports High-Definition Multimedia Interface Compliance Test Specification 1.4.

9.4.1.3.1 Stereoscopic Support on HDMI

SoC display supports HDMI 1.4 3D video formats. If the HDMI panel is detected to support 3D video format then the SW driver will program Pipe2dB for the correct pipe timing parameters.

The left and right frames can be loaded from independent frame buffers in the main memory. Depending on the input S3D format, the display controller can be enabled do perform frame repositioning, image scaling, line interleaving.

Figure 13. HDMI Overview



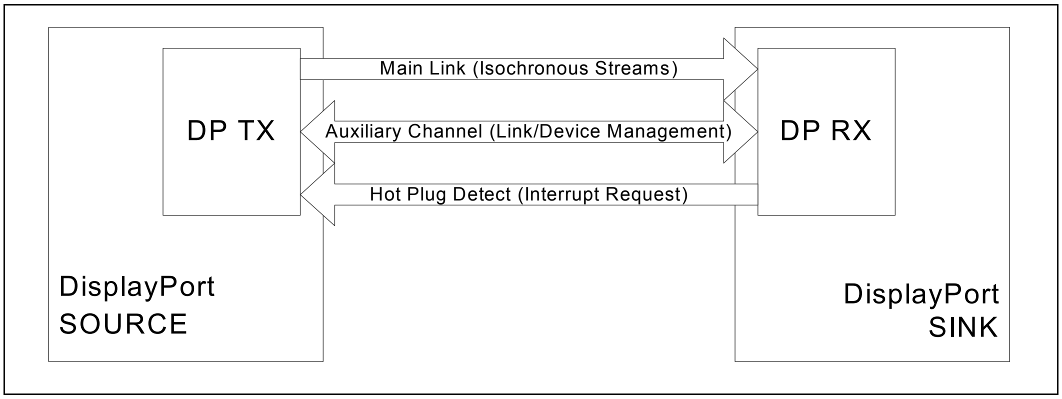
9.4.1.4 Display Port

Display Port is a digital communication interface that utilizes differential signalling to achieve a high bandwidth bus interface designed to support connections between PCs and monitors, projectors, and TV displays. Display Port is also suitable for display connections between consumer electronics devices such as high definition optical disc players, set top boxes, and TV displays.

A Display Port consists of a Main Link, Auxiliary channel, and a Hot Plug Detect signal. The Main Link is a uni-directional, high-bandwidth, and low latency channel used for transport of isochronous data streams such as uncompressed video and audio. The Auxiliary Channel (AUX CH) is a half-duplex bidirectional channel used for link management and device control. The Hot Plug Detect (HPD) signal serves as an interrupt request for the sink device.

The SoC supports DisplayPort Standard Version 1.2.

Figure 14. DisplayPort* Overview


DP TX
Main Link (Isochronous Streams)
Auxiliary Channel (Link/Device Management)
Hot Plug Detect (Interrupt Request)
DP RX
DisplayPort SOURCE
DisplayPort SINK

9.4.1.5 Embedded DisplayPort (eDP)

Embedded DisplayPort (eDP) is a embedded version of the DisplayPort standard oriented towards applications such as notebook and All-In-One PCs. eDP is supported only on Digital Display Interfaces 0 and/or 1. Like DisplayPort, Embedded DisplayPort also consists of a Main Link, Auxiliary channel, and a optional Hot Plug Detect signal.

Each eDP port can be configured for up-to 4 lanes.

The SoC supports Embedded DisplayPort Standard Version 1.3.

9.4.1.5.1 DisplayPort Auxiliary Channel

A bidirectional AC coupled AUX channel interface replaces the I^2C for EDID read, link management and device control. I^2C-to-Aux bridges are required to connect legacy display devices.

9.4.1.5.2 Hot-Plug Detect (HPD)

SoC supports HPD for Hot-Plug sink events on the HDMI and DisplayPort interfaces.

9.4.1.5.3 Integrated Audio over HDMI and DisplayPort

SoC can support two audio streams on DP/HDMI ports. Each stream can be programmable to either DDI port. HDMI/DP audio streams can be sent with video streams as follows.

LPE mode: In this mode the uncompressed or compressed audio sample buffers are generated either by OS the audio stack or by audio Lower Power Engine (LPE) and stored in system memory.The display controller fetches audio samples from these buffers, forms an SPDIF frame with VUCP and preamble (if needed), then sends out with video packets.

9.4.1.5.4 High-Bandwidth Digital Content Protection (HDCP)

HDCP is the technology for protecting high definition content against unauthorized copy or unreceptive between a source (computer, digital set top boxes, etc.) and the sink (panels, monitor, and TV). The SoC supports HDCP 1.4(wired)/2.2(wireless) for content protection over wired displays (HDMI, DisplayPort and Embedded DisplayPort).

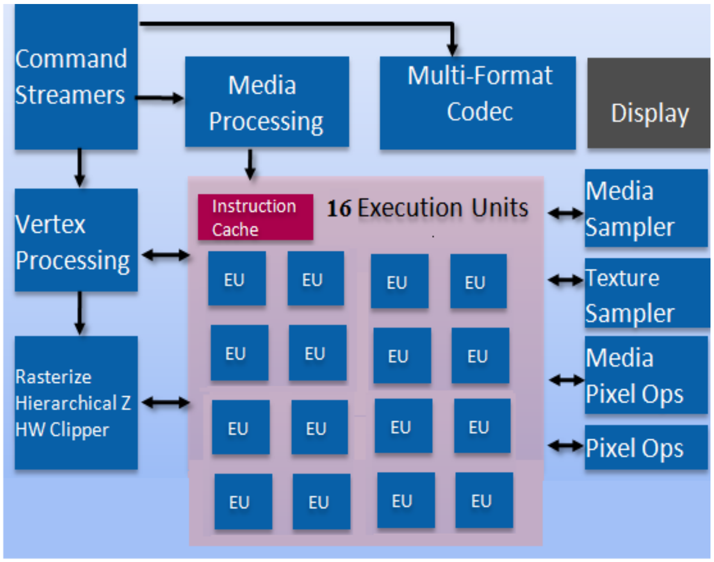
9.5 References

- High-Definition Multimedia Interface Specification, Version 1.4b
- High-bandwidth Digital Content Protection System, Revision 1.4
- VESA DisplayPort Standard, Version 1.2
- VESA Embedded DisplayPort Standard, Version 1.3

9.6 3D Graphics and Video

The SoC implements a derivative of the Generation 8 LP graphics engine which consists of rendering engine and bit stream encoder/decoder engine. The rendering engine is used for 3D rendering, media compositing and video encoding. The Graphics engine is built around sixteen execution units (EUs).

Figure 15. 3D Graphics Block Diagram



9.7 Features

The 3D graphics pipeline architecture simultaneously operates on different primitives or on different portions of the same primitive. All the cores are fully programmable, increasing the versatility of the 3D Engine. The Gen 8.0 LP 3D engine provides the following performance and power-management enhancements:

- Hierarchal-Z
- Video quality enhancements

9.7.1 3D Engine Execution Units

- The EUs perform 128-bit wide execution per clock
- Support SIMD8 instructions for vertex processing and SIMD16 instructions for pixel processing

9.7.2 3D Pipeline

9.7.2.1 Vertex Fetch (VF) Stage

The VF stage executes 3DPRIMITIVE commands. Some enhancements have been included to better support legacy D3D APIs as well as SGI OpenGL*.

9.7.2.2 Vertex Shader (VS) Stage

The VS stage performs shading of vertices output by the VF function. The VS unit produces an output vertex reference for every input vertex reference received from the VF unit, in the order received.

9.7.2.3 Geometry Shader (GS) Stage

The GS stage receives inputs from the VS stage. Compiled application-provided GS programs, specifying an algorithm to convert the vertices of an input object into some output primitives. For example, a GS shader may convert lines of a line strip into polygons representing a corresponding segment of a blade of grass centered on the line. Or it could use adjacency information to detect silhouette edges of triangles and output polygons extruding out from the edges.

9.7.2.4 Clip Stage

The Clip stage performs general processing on incoming 3D objects. However, it also includes specialized logic to perform a Clip Test function on incoming objects. The Clip Test optimizes generalized 3D Clipping. The Clip unit examines the position of incoming vertices, and accepts/rejects 3D objects based on its Clip algorithm.

9.7.2.5 Strips and Fans (SF) Stage

The SF stage performs setup operations required to rasterize 3D objects. The outputs from the SF stage to the Windower stage contain implementation-specific information required for the rasterization of objects and also supports clipping of primitives to some extent.

9.7.2.6 Windower/IZ (WIZ) Stage

The WIZ unit performs an early depth test, which removes failing pixels and eliminates unnecessary processing overhead.

The Windower uses the parameters provided by the SF unit in the object-specific rasterization algorithms. The WIZ unit rasterizes objects into the corresponding set of pixels. The Windower is also capable of performing dithering, whereby the illusion of a higher resolution when using low-bpp channels in color buffers is possible. Color dithering diffuses the sharp color bands seen on smooth-shaded objects.

9.7.3 Video Engine

The video engine is part of the Intel Processor Graphics for image processing, play-back and transcode of Video applications. Processor Graphics video engine has a dedicated fixed hardware pipe-line for high quality decode and encode of media content. This engine supports Full HW acceleration for decode of AVC/H.264, VC-1 and MPEG -2 contents along with encode of MPEG-2 and AVC/H.264 apart from various video processing features. The new Processor Graphics Video engine adds support for processing features such as frame rate conversion, image stabilization and gamut conversion.

9.8 VED (Video Encode/Decode)

The video engine is part of the Intel Processor Graphics for image processing, play-back and transcode of Video applications. Processor Graphics video engine has a dedicated fixed hardware pipe-line for high quality decode and encode of media content.

9.8.1 Features

The features for the Video decode hardware accelerator in SoC are:

- VED core can be configured on a time division multiplex basis to handle single, dual and multi-stream HD decoding/encoding.
- VED provides full hardware acceleration Decode/Encode support below Media formats.

Table 72. Hardware Accelerated Video Decode/Encode Codec Support

Encode Format	Profile	Level	Resolution	Bitrate (Mbps)	Frame Rate
H.263			480p		30
H.264	HP/BP/CBP	L5.1	4kx2k 1080p	100-130	30 120
VP8			4kx2k		30
MVC	HP/BP/CBP	L4.2	1080p		60
JPEG	1067Mpps (420), 800Mpps (422) @400Mhz				

Decode Format	Profile	Level	Resolution	Bitrate (Mbps)	Frame Rate
H.263			480p		30
MPEG4	SP		480p		30
H.264	HP,MP,CBP	L5.2	4Kx2K 1080P	200-250	60 240
H.265(HEVC)	MP	L5	4Kx2K		30
VP8			4kx2k		30
VP9			1080p		30
MVC			4Kx2K		30
MPEG2	MP	HL	1080p		60
VC1	AP	L4	1080P		60
JPEG	1067Mpps (420), 800Mpps (422) @400Mhz				

9.9 Register Map

For more information on Graphics, Video and Display registers refer Intel® Atom™ Z8000 Processor Series Datasheet (Volume 2 of 2), Doc ID:332066.

§

10 PCI Express 2.0

There are up to two PCI Express root ports, each supporting the *PCI Express* Base Specification*, Rev. 2.0 at a maximum 5 GT/s signaling rate. The root ports can be configured to support a diverse set of lane assignments.

10.1 Signal Descriptions

Refer Chapter 2, “Physical Interfaces” for additional details.

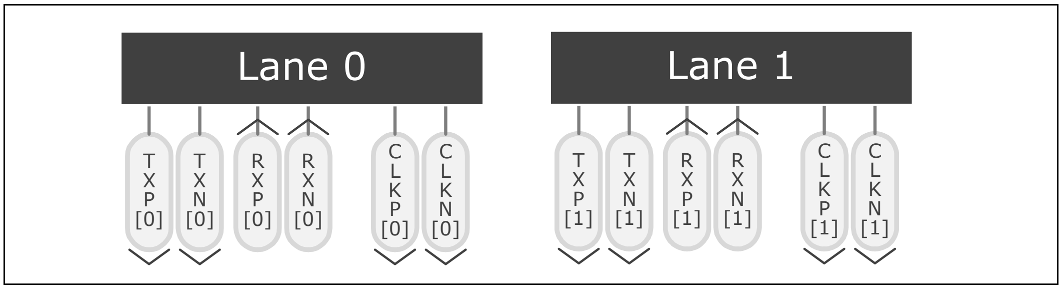
The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin.
- **Direction**: The buffer direction can be either input, output, or I/O (bidirectional).
- **Type:** The buffer type found in Chapter 19, “Electrical Specifications”.
- **Description**: A brief explanation of the signal’s function.

Table 73. Signals

Signal Name	Direction /Type	Description
PCIE_TXP[1:0] **PCIE_TXN[1:0]**	O PCIe	**PCI Express* Transmit** PCI Express* Ports 1:0 transmit pair (P and N) signals. Each pair makes up the transmit half of a lane.
PCIE_RXP[1:0] **PCIE_RXN[1:0]**	I PCIe	**PCI Express* Receive:** PCI Express* Ports 1:0 receive pair (P and N) signals. Each pair makes up the receive half of lane.
PCIE_CLKREQ[1:0]_N	IO	**PCI Express* Clock Request** Used for devices that need to request one of the output clocks. Each clock request maps to the matching PCIe Root Port (e.g. PCIE_CLKREQ#[0] maps to PCIE Root Port [0] and so on) **NOTE:** These signals are muxed and may be used by other functions.
P_RCOMP_P **P_RCOMP_N**	I/O	These pins connected with 402 Ohm 1% between RCOMP pads.

Figure 16. PCIe* 2.0 Lane 0 Signal Example



10.2 Features

- Conforms to *PCI Express* Base Specification*, Rev. 2.0.
- 5.0 or 2.5 GT/s operation per root port.
- Virtual Channel support for VC0 only.
- x1, x2 link widths (auto negotiated).
- Spread Spectrum Clocking (SSC) is supported for PCIe Gen1 components.
- Flexible Root Port configuration options
  - (1) x2's
  - (2) x1
- Interrupts and Events
  - Legacy (INT*x*) and MSI Interrupts
  - General Purpose Events
  - Express Card Hot Plug Events
  - System Error Events
- Power Management
  - Link State support for ASPM(L0s, L1), L1 sub states (L1.SNOOZ,L1.OFF), L23_RDY,L2 and L3.
  - Powered down in ACPI S3 state - L3.

Note: Intel recommends disabling Spread Spectrum Clocking (SSC), if PCIe Gen2 based component is used.

10.2.1 Root Port Configurations

Depending on SKU, there are up to two possible lane assignments for root ports 1-2.

Root port configurations are set by SoftStraps stored in SPI flash, and the default option is "(2) x1". Links for each root port will train automatically to the maximum possible for each port.

Note: x2 link widths are not common. Most devices will only train to x1.

Note: PCI functions in PCI configuration space are disabled for root ports not available.

10.2.2 Interrupts and Events

A root port is capable of handling interrupts and events from an end point device. A root port can also generate its own interrupts for some events, including power management and hot plug events, but also including error events.

There are two interrupt types a root port will receive from an end point device: INTx (legacy), and MSI. MSI's are automatically passed upstream by the root port, just as other memory writes would be. INTx messages are delivered to the Legacy block's interrupt router/controller by the root port.

Events and interrupts that are handled by the root port are shown with the possible interrupts they can deliver to the interrupt decoder/router.

Table 74. Possible Interrupts Generated From Events/Packets

Packet/Event	Type	INTx	MSI	SERR	SCI	SMI	GPE
INTx	Packet	X	X				
PM_PME	Packet	X	X				
Power Management (PM)	Event	X	X		X	X	
Hot Plug (HP)	Event	X	X		X	X	
ERR_CORR	Packet			X			
ERR_NONFATAL	Packet			X			
ERR_FATAL	Packet			X			
Internal Error	Event			X			
VDM	Packet						X

NOTE: Above table lists the possible interrupts and events generated based on Packets received, or events generated in the root port. Configuration needed by software to enable the different interrupts as applicable.

When INTx interrupts are received by an end point, they are mapped to the following interrupts and sent to the interrupt decoder/router in the iLB.

Table 75. Interrupt Generated for INT[A-D] Interrupts

	INTA	INTB	INTC	INTD
Root Port 1	INTA#	INTB#	INTC#	INTD#
Root Port 2	INTD#	INTA#	INTB#	INTC#

NOTE: Interrupts generated from events within the root port are not swizzled.

10.2.2.1 Express Card Hot Plug Events

Express Card Hot Plug is available based on Presence Detection for each root port.

Note: A full Hot Plug Controller is not implemented.

Presence detection occurs when a PCI Express* device is plugged in and power is supplied. The physical layer will detect the presence of the device, and the root port will set the SLSTS.PDS and SLSTS.PDC bits.

When a device is removed and detected by the physical layer, the root port will clear the SLSTS.PDS bit, and set the SLSTS.PDC bit.

Interrupts can be generated by the root port when a hot plug event occurs. A hot plug event is defined as the transition of the SLSTS.PDC bit from 0 to 1. Software can set the SLCTL.PDE and SLTCTL.HPE bits to allow hot plug events to generate an interrupt.

If SLCTL.PDE and SLTCTL.HPE are both set, and STSTS.PDC transitions from 0 to 1, an interrupt will be generated.

10.2.2.2 System Error (SERR)

System Error events are support by both internal and external sources. Refer the PCI Express* Base Specification, Rev. 2.0 for details.

10.2.3 Power Management

Each root port's link supports L0s, L1, and L2/3 link states per PCI Express* Base Specification, Rev. 2.0. L2/3 is entered on entry to S3.

10.3 References

PCI Express Base Specification*, Rev. 2.0

10.4 Register Map

For more information on PCI Express* 2.0 registers refer Intel® Atom™ Z8000 Processor Series Datasheet (Volume 2 of 2), Doc ID:332066.

§

11 MIPI-Camera Serial Interface (CSI) and ISP

MIPI CSI and controller front end interfaces with three sensors and is capable of simultaneously acquiring three streams, one from each sensor. These three streams are presented to the ISP.

11.1 Signal Descriptions

Refer Chapter 2, "Physical Interfaces" for additional details.

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin.
- **Direction**: The buffer direction can be either input, output, or I/O (bidirectional).
- **Type:** The buffer type found in Chapter 19, "Electrical Specifications".
- **Description**: A brief explanation of the signal's function.

Table 76. CSI Signals

Signal Name	Direction	Description
MCSI1_CLKP/N	I	**Clock Lane:** MIPI CSI input clock lane 0 for port 1.
MCSI1_DP/N[3:0]	I	**Data Lanes:** Four MIPI CSI Data Lanes (0-3) for port 1. Lanes 2 and 3 can optionally used as data lanes for port 3.
MCSI2_CLKP/N	I	**Clock Lane:** MIPI CSI input clock lane 0 for port 2.
MCSI2_DP/N[1:0]	I	**Data Lane:** Two MIPI CSI Data Lanes for port 2.
MCSI3_CLKP/N	I	**Clock Lane:** MIPI CSI input clock lane 0 for port 3.
MCSI_RCOMP	I/O	**Resistor Compensation:** This is for pre-driver slew rate compensation for the MIPI CSI Interface.

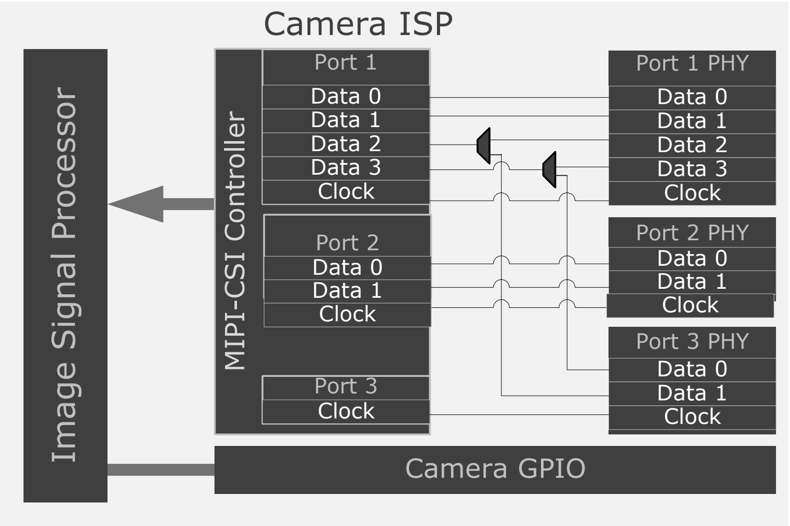
Table 77. GPIO Signals (Sheet 1 of 2)

Signal Name	Direction /Type	Description
MCSI_GPIO[00]	I/O	Output from shutter switch when its pressed halfway. This switch state is used to trigger the Auto focus LED for Xenon Flash or Torch mode for LED Flash
MCSI_GPIO[01]	I/O	Output from shutter switch when its pressed full way. This switch state is used to trigger Xenon flash or LED Flash
MCSI_GPIO[02]	I/O	Active high control signal to Xenon Flash to start charging the Capacitor
MCSI_GPIO[03]	I/O	Active low output from Xenon Flash to indicate that the capacitor is fully charged and is ready to be triggered

Table 77. GPIO Signals (Sheet 2 of 2)

Signal Name	Direction /Type	Description
MCSI_GPIO[04]	I/O	Active high Xenon Flash trigger / Enables Torch Mode on LED Flash IC
MCSI_GPIO[05]	I/O	Enables Red Eye Reduction LED for Xenon / Triggers STROBE on LED Flash IC /
MCSI_GPIO[06]	I/O	Camera Sensor 0 Strobe Output to SoC to indicate beginning of capture / Active high signal to still camera to power down the device.
MCSI_GPIO[07]	I/O	Camera Sensor 1 Strobe Output to SoC to indicate beginning of capture / Active high signal to still camera to power down the device.
MCSI_GPIO[08]	I/O	Active high signal to video camera to power down the device.
MCSI_GPIO[09]	I/O	Active low output signal to reset digital still camera #0.
MCSI_GPIO[10]	I/O	Active low output signal to reset digital still camera #1
MCSI_GPIO[11]	I/O	Active low output signal to reset digital video camera

Figure 17. Camera Connectivity



11.2 Features

- Integrated MIPI-CSI 2.0 interface.
- Image Signal Processor (ISP) with DMA and local SRAM.
- Imaging data is received by the MIPI-CSI interface and is relayed to the ISP for processing.
- Up to six MIPI-CSI 2.0 data lanes.
  - Each lane can operate at up to 1.5Gbp/s. resulting in roughly 1.2 Gbp/s of actual pixels.
- The MIPI-CSI interface supports lossless compressed image streams to increases the effective bandwidth without losing data.
- Up to 13MP sensors supported, and full HD 1080p30.
  - Can also support Stereo HD 1080p30.

11.2.1 Imaging Capabilities

The following table summarizes imaging capabilities.

Table 78. Imaging Capabilities

Feature	Capabilities
Sensor interface	Configurable MIPI-CSI2 interfaces. 3 sensors: x2, x2, x2 or x1 x2, x3 2 sensors: x4, x2
Simultaneous sensors	Up to 3 simultaneous sensors
2D Image capture	13MP ZSL @ 18fps
2D video capture	Up to 1080p30
Input formats	RAW 8, 10, 12, 14, RGB444, 565, 888, YUV420, 422, JPEG.
Output formats)	YUV422, YUV420, RAW
Special Features	Image and video stabilization Low light noise reduction Burst mode capture Memory to memory processing 3A (Auto Exposure (AE), Auto White Balance (AWB) and Auto Focus (AF)) High Dynamic Range (HDR) Multi-focus Zero shutter lag

11.2.2 Simultaneous Acquisition

SoC will support on-the-fly processing for only one image at a time. While this image is being processed on-the-fly, images from the other two cameras are saved to DRAM for later processing.

11.2.3 Primary Camera Still Image Resolution

Maximum still image resolution for the primary camera in post-processing mode is limited by the resolution of the sensors. Currently 13Mpixel sensors are supported.

Higher resolution, or higher frame rates are supported as long as the product of resolution and frame rate does not exceed 235 Mpixels/s (= 13 Mpixels * 18 fps).

Maximum primary camera on-the-fly stereoscopic still image resolution for primary camera is 8 Mpixel for each of the left and right images at 18 fps. The number of Mpixels can be increased by decreasing the frame rate.

11.2.4 Burst Mode Support

The SoC supports capturing multiple images back to back at maximum sensor resolution. At least 5 images must be captured in burst mode. The maximum number of images that can be so captured is limited only by available system memory. These images need not be processed on-the-fly.

11.2.5 Continuous Mode Capture

SoC supports capturing images and saving them to DRAM in a ring of frame buffers continuously at maximum sensor resolution. This adds a round trip to memory for every frame and increases the bandwidth requirements.

11.2.6 Secondary Camera Still Image Resolution

Maximum secondary camera still image resolution is 4 Mpixel at 15 fps.

11.2.7 Primary Camera Video Resolution

Maximum primary camera video resolution is 1080p60.

Maximum primary camera dual video resolution is 1080p30.

11.2.8 Secondary Camera Video Resolution

Maximum secondary camera video resolution is 1080p30.

11.2.9 Bit Depth

Capable of processing 14-bit images at the stated performance levels.

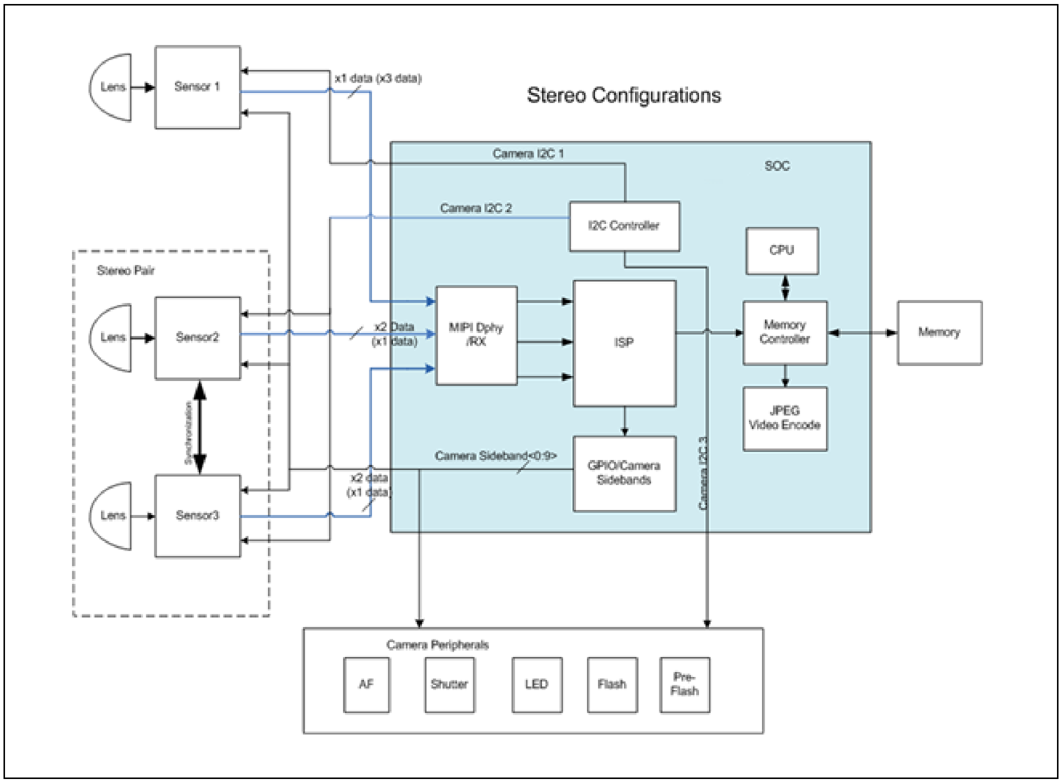
Capable of processing 18-bit images at half the performance levels, i.e. process on-the-fly 13 Mpixel 18-bit images at 7 fps instead of 15 fps.

Capable of processing up to 18-bit precision.

The higher precision processing will be employed mainly for high dynamic range imaging (HDR).

11.3 Imaging Subsystem Integration

Figure 18. Image Processing Components



11.3.1 CPU Core

The CPU core augments the signal processing capabilities of the hardware to perform post-processing on images such as auto focus, auto white balance, and auto exposure. The CPU also runs the drivers that control the GPIOs and I^{2}C for sensor control.

11.3.2 Imaging Signal Processor (ISP)

The ISP (Imaging Signal Processor) includes a 64-way vector processor enabling high quality camera functionality. Key features include support of three camera sensors.

11.3.2.1 MIPI-CSI-2 Ports

The SoC has three MIPI clock lanes and six MIPI data lanes. The Analog Front End (AFE) and Digital Physical Layer (DPHY) take these lanes and connects them to three virtual ports. Two data lanes are dedicated to each of the rear facing cameras and the remaining data lane is connected to the front facing camera. The MIPI interfaces follow the MIPI-CSI-2 specifications as defined by the MIPI Alliance. They support YUV420, YUV422, RGB444, RGB555, RGB565, and RAW 8b/10b/12b. Both MIPI ports support compression settings specified in MIPI-CSI-2 draft specification 1.01.00 Annex E. The compression is implemented in Hardware with support for Predictor 1 and Predictor 2. Supported compression schemes:

- 12-8-12
- 12-7-12
- 12-6-12
- 10-8-10
- 10-7-10
- 10-6-10

The data compression schemes above use an X-Y-Z naming convention where X is the number of bits per pixel in the original image, Y is the encoded (compressed) bits per pixel and Z is the decoded (uncompressed) bits per pixel.

11.3.2.2 I^{2}C for Camera Interface

The platform supports three (3) I^{2}C ports for the camera interface. These ports are used to control the camera sensors and the camera peripherals such as flash LED and lens motor.

11.3.2.3 Camera Sideband for Camera Interface

Twelve (12) GPIO signals are allocated for camera functions, refer Table 77 for signal names. These GPIOs are multiplexed and are available for other usages without powering on the ISP. The ISP provides a timing control block through which the GPIOs can be controlled to support assertion, de-assertion, pulse widths and delay. The configuration below of camera GPIOs is just an example of how the GPIOs can be used. Several of these functions could be implemented using I^{2}C, depending on the sensor implementation for the platform.

- Sensor Reset signals
  - —Force hardware reset on one or more of the sensors.
- Sensor Single Shot Trigger signal
  - —Indicate that the target sensor needs to send a full frame in a single shot mode, or to capture the full frame for flash synchronization.
- PreLight Trigger signal
  - —Light up a pilot lamp prior to firing the flash for preventing red-eye.
- Flash Trigger signal
  - —Indicate that a full frame is about to be captured. The Flash fires when it detects an assertion of the signal.
- Sensor Strobe Trigger signal
  - —Asserted by the target sensor to indicate the start of a full frame, when it is configured in the single shot mode, or to indicate a flash exposed frame for flash synchronization.

11.4 Functional Description

At a high level, the Camera Subsystem supports the following modes:

- Preview
- Image capture
- Video capture

11.4.1 Preview

Once the ISP and the camera subsystem is enabled, the ISP goes into the preview mode where very low resolution frames, such as VGA/480p (programmable), are being processed.

11.4.2 Image Capture

During the image capture mode, the camera subsystem can acquire at a peak throughput of 13 Mpixels @ 18fps. While doing this, it continues to output preview frames simultaneously.

- The ISP can output RAW, RGB or YUV formats. The ISP can capture one full frame at a time or perform burst mode capture, where up to five full back-to-back frames are recorded.
- The ISP will not limit the number of back-to-back full frames captured, but the number is programmable and determined on how much memory can be allocated dynamically.
- The ISP can process all the frames on the fly and writes to memory only after fully processing the frames, without requiring download of any part of the frame for further processing.
    - —The exceptions to this approach are image stabilization and some other advanced functions requiring temporal information over multiple frames.

The ISP can support image stabilization in image capture model.

- The ISP initially outputs preview frames.
- When the user decides to capture the picture, image stabilization is enabled. The ISP checks the previous frame for motion and compensates for it appropriately.

Auto Exposure (AE), Auto Focus (AF), and Auto White Balance (AWB), together known as 3A, are implemented in the CPU to provide flexibility.

11.4.3 Video Capture

During video recording, the ISP can capture video up to 1080p @ 60 fps and output preview frames concurrently. The ISP output video frames to memory in YUV420 or YUV422 format.

11.4.4 ISP

The Camera subsystem consists of 2 parts, the hardware subsystem and a software stack that implements the ISP functionality on top of this hardware.

The core of the ISP is a vector processor. The vector processor is supported by the following components:

- Interfaces for data and control
- A small input formatter that parallelizes the data
- A scalar (RISC) processor, for system control and low-rate processing
- An accelerator for scaling, digital zoom, and lens distortion correction
- A DMA engine transfers large amounts of data such as input and output image data or large parameter sets between LPDDR2 and the ISP block.

11.4.5 Memory Management Unit (MMU)

The camera subsystem has capabilities to deal with a virtual address space, since a contiguous memory range in the order of 16–32MB cannot be guaranteed by the OS.

11.4.5.1 Interface

The MMU performs the lookup required for address translation from a 32-bit virtual address to 36-bit physical address. The lookup tables are stored external to the system. The MMU performs the lookup through a master interface without burst support that is connected to the Open Core Protocol (OCP) master of the subsystem. The MMU configuration registers can be accessed through a 32-bit Core I/O (CIO) slave interface. Additionally there is a 32-bit CIO slave interface connected to the address translator.

11.5 MIPI-CSI-2 Receiver

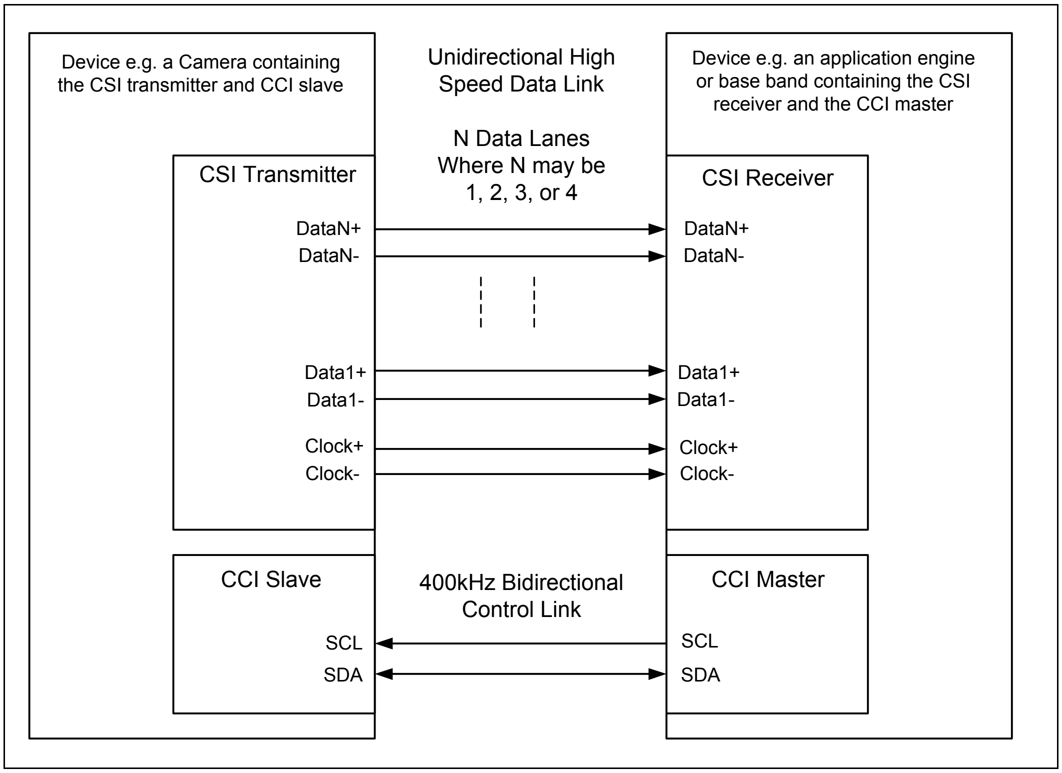
MIPI-CSI-2 devices are camera serial interface devices. They are categorized into two types, a CSI transmitter device with Camera Control Interface (CCI) slave and CSI receiver device with CCI master.

Data transfer by means of MIPI-CSI is unidirectional that is, from transmitter to receiver. CCI data transfer is bidirectional between the CCI slave and master.

Camera Serial Interface Bus (CSI) is a type of serial bus that enables transfer of data between a Transmitter device and a receiver device. The CSI device has a point-to-point connections with another CSI device by means of D-PHYs and as shown in Figure 19.

Similarly, CCI (Camera Control Interface bus) is a type of serial bus that enables transfer back and forth between the master CCI and a Slave CCI Unit.

Figure 19. MIPI-CSI Bus Block Diagram


Device e.g. a Camera containing the CSI transmitter and CCI slave
Unidirectional High Speed Data Link
Device e.g. an application engine or base band containing the CSI receiver and the CCI master
N Data Lanes Where N may be 1, 2, 3, or 4
CSI Transmitter
CSI Receiver
DataN+
DataN-
DataN+
DataN-
Data1+
Data1-
Data1+
Data1-
Clock+
Clock-
Clock+
Clock-
CCI Slave
400kHz Bidirectional Control Link
CCI Master
SCL
SDA
SCL
SDA

D-PHY data lane signals are transferred point-to-point differentially using two signal lines and a clock lane. There are two signaling modes, a high speed mode that operates up-to 1500Mbs and a low power mode that works at 10Mbs. The mode is set to low power mode and a stop state at start up/power up. Depending on the desired data transfer type, the lanes switch between high and low power modes.

The CCI interface consists of an I^2C bus which has a clock line and a bidirectional data line.

The MIPI-CSI-2 devices operate in a layered fashion. There are 5 layers identified at the receiver and transmitter ends.

MIPI-CSI-2 Functional Layers:

- **PHY Layer**
  - An embedded electrical layer sends and detects start of packet signalling and end of packet signalling on the data lanes. It contains a serializer and deserializer unit to interface with the PPI / lane management unit. There is also a clock divider unit to source and receive the clock during different modes of operation.
- **PPI/Lane Management Unit**
  - This layer does the lane buffering and distributes the data in the lanes as programmed in a round robin manner and also merges them for the PLI/Low Level Protocol unit.

- **PLI/Low Level Protocol Unit**
  - This layer packetizes as well as de-packetizes the data with respect to channels, frames, colors and line formats. There is also a CRC checker or CRC generator unit to pack the payload data with CRC checksum bits for payload data protection.
- **Pixel/Byte to Byte/Pixel Packing Formats**
  - Conversion of pixel formats to data bytes in the payload data is done depending on the type of image data supported by the application. It also re-converts the raw data bytes to pixel format understandable to the application layer.
- **Application**
  - Depending on the type of formats, camera types, capability of the camera used by the transmitter, the application layer recovers the image formats and reproduces the image in the display unit. It also works on de-framing the data into pixel-to-packing formats. High level encoding and decoding of image data is handled in the application unit.

11.5.1 MIPI-CSI-2 Receiver Features

CSI Features:

- Compliant to CSI-2 MIPI specification for Camera Serial Interface (Version 1.00).
- Supports standard D-PHY receiver compliant to the MIPI Specification.
- Supports PHY data programmability up to four lanes.
- Supports PHY data time-out programming.
- Has controls to start and re-start the CSI-2 data transmission for synchronization failures and to support recovery.
- The ISP may not support all the data formats that the CSI-2 receiver can handle.
- Refer Table 78 for formats supported by the ISP
- Supports all generic short packet data types
- Single Image Signal Processor interface for pixel transfers to support multiple image streams for all virtual channel numbers

D-PHY Features:

- Supports synchronous transfer in high speed mode with a bit rate of 80-1500Mb/s.
- Supports asynchronous transfer in low power mode with a bit rate of 10Mb/s.
- Differential signalling for HS data.
- Spaced one-hot encoding for Low Power [LP] data.
- Data lanes support transfer of data in high speed as well as low power modes.
- Supports ultra low power mode, escape mode, and high speed mode.
- Hasa clock divider unit to generate clock for parallel data reception and transmission from and to the PPI unit.
- Activates and disconnects high speed terminators for reception and control mode.
- Activates and disconnects low power terminators for reception and transmission.

11.6 Register Map

For more information on MIPI- Camera Serial Interface (CSI) and ISP registers refer Intel® Atom™ Z8000 Processor Series Datasheet (Volume 2 of 2), Doc ID:332066.

§

12 SoC Storage

12.1 SoC Storage Overview

12.1.1 Storage Control Cluster (eMMC, SDIO, SD)

The SCC consists of SDIO, SD and eMMC controllers to support mass storage and IO devices.

- One eMMC 4.51 interface
- One SD 3.0 interface
- One SDIO 3.0 interface

12.2 Signal Descriptions

Refer Chapter 2, “Physical Interfaces” for additional details.

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin.
- **Direction**: The buffer direction can be either input, output, or I/O (bidirectional).
- **Type:** The buffer type found in Chapter 19, “Electrical Specifications”.
- **Description**: A brief explanation of the signal’s function.

Table 79. eMMC Signals

Signal Name	Direction /Type	Description
MMC1_CLK	I/O/GPIO	**eMMC Clock** The frequency may vary between 25 and 200MHz.
MMC1_D[7:0]	I/O/GPIO	**eMMC Port Data bits 0 to 7** Bidirectional port used to transfer data to and from eMMC device. By default, after power up or reset, only D[0] is used for data transfer. A wider data bus can be configured for data transfer, using either D[0]-D[3] or D[0]-D[7], by the MultiMedia Card controller. The MultiMedia Card includes internal pull-ups for data lines D[1]-D[7]. Immediately after entering the 4-bit mode, the card disconnects the internal pull ups of lines D[1], D[2], and D[3]. Correspondingly, immediately after entering to the 8-bit mode the card disconnects the internal pull-ups of lines D[1]-D[7].
MMC1_CMD	I/O/GPIO	**eMMC Port Command** This signal is used for card initialization and transfer of commands. It has two modes—open-drain for initialization, and push-pull for fast command transfer.
MMC1_RCOMP	I/O/GPIO	**eMMC RCOMP** This signal is used for pre-driver slew rate compensation.
MMC1_RST_N	I/O/GPIO	**eMMC Reset Signals** Active low to reset.
MMC1_RCLK	I/GPIO	**eMMC Return Clock Signals**

Table 80. SDIO Signals

Signal Name	Direction /Type	Description
SD2_CLK	I/O/GPIO	**SDIO Clock** The frequency may vary between 25 and 200MHz.
SD2_D[2:0]	I/O/GPIO	**SDIO Port Data bits 0 to 2** Bidirectional port used to transfer data to and from SDIO device. By default, after power up or reset, only D[0] is used for data transfer. A wider data bus can be configured for data transfer, using D[0]-D[3].
SD2_D[3]_CD_N	I/O/GPIO	**SDIO Port Data bit 3** Bidirectional port used to transfer data to and from the SDIO device. Also, **Card Detect**. Active low when device is present.
SD2_CMD	I/O/GPIO	**SDIO Port Command** This signal is used for card initialization and transfer of commands. It has two modes—open-drain for initialization, and push-pull for fast command transfer.

Table 81. SD Signals (Sheet 1 of 2)

Signal Name	Direction /Type	Description
SD3_CLK	I/O/GPIO	**SD Card Clock** The frequency may vary between 25 and 200 MHz.
SD3_D[3:0]	I/O/GPIO	**SD Card Data bits 0 to 3** Bidirectional port used to transfer data to and from SD/MMC card. By default, after power up or reset, only D[0] is used for data transfer. A wider data bus can be configured for data transfer, using D[0]-D[3].
SD3_CD_N	I/O/GPIO	**SD Card Detect** Active low when a card is present. Floating (pulled high with internal PU) when a card is not present.
SD3_CMD	I/O/GPIO	**SD Card Command** This signal is used for card initialization and transfer of commands. It has two modes—open-drain for initialization, and push-pull for fast command transfer.
SD3_1P8EN	I/O/GPIO	**SD Card 1.8V Enable** Controls the voltage of the SD Card, the default is low (3.3V). The voltage is 1.8V when this signal is high.

Table 81. SD Signals (Sheet 2 of 2)

Signal Name	Direction /Type	Description
SD3_RCOMP	I/O/GPIO	**SD Card RCOMP** This signal is used for pre-driver slew rate compensation.
SD3_PWREN_N	I/O/GPIO	**SD Card Power Enable** This signal is used to enable power on a SD device.
SD3_WP	I/O/GPIO	**SD Card Write Protect** Active high to protect from write.

12.3 Features

12.3.1 Memory Capacity

- Standard Capacity SD Memory Card (SDSC): Up to and including 2 GB.
- High Capacity SD Memory Card (SDHC): More than 2GB and up to and including 32GB.
- Extended Capacity SD Memory Card (SDXC): More than 32GB and up to and including 2TB.

12.3.2 SDIO/SD Interface Features

- Host clock rate variable between 0 and 200 MHz.
- Up to 800 Mbits per second data rate using 4 parallel data lines (SDR104 mode).
- Transfers the data in 1 bit and 4 bit SD modes.
- Transfers the data in following UHS-I modes (SDR12/25/50/104 and DDR50).
- Cyclic Redundancy Check CRC7 for command and CRC16 for data integrity.
- Designed to work with I/O cards, Read-only cards and Read/Write cards.
- Supports Read wait Control, Suspend/Resume operation.

12.3.3 eMMC Interface Features

- Supports eMMC v4.51.
- Host clock rate variable between 0 and 200 MHz.
- Supports HS400 mode.
- Up to 1600 Mbits per second data rate using 8 bit parallel data lines (High Speed DDR mode).
- Up to 3200 Mbits per second data rate using 8 bit parallel data lines (HS400 mode).
- Transfers the data in 1 bit, 4 bit and 8 bit modes.
- Cyclic Redundancy Check CRC7 for command and CRC16 for data integrity.
- Supports MMC Plus and MMC mobile.

12.3.4 Storage Interfaces

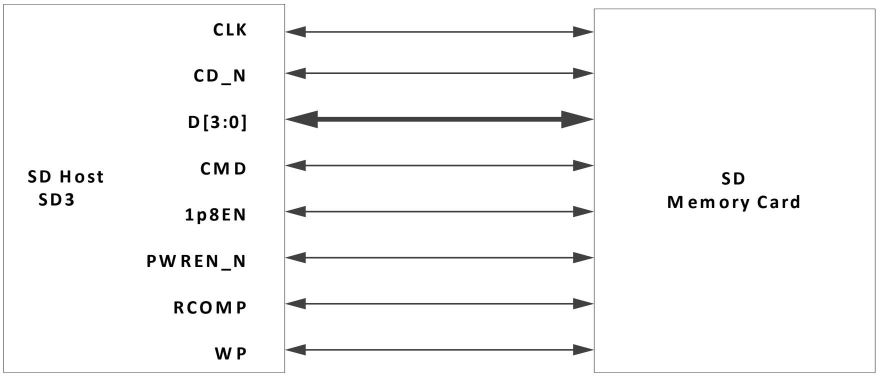
This section provides a very high level overview of the SD, SDIO, eMMC 4.51 specification.

12.3.4.1 SD 3.0 Bus Interface

The SD Card bus has a single master, single slaves (card), synchronous topology (refer Figure 20). During initialization process commands are sent to the card, allowing the application to detect the card and assign logical addresses to the physical slot. All data communication in the Card Identification Mode uses the command line (CMD) only.

SD bus allows dynamic configuration of the number of data lines. After power up, by default, the SD Card will use only SD3_D[0] for data transfer. After initialization the host can change the bus width (number of active data lines). This feature allows easy trade off between hardware cost and system performance. Note that while DAT1-SD3_D[3:1] are not in use, the SoC will tri-state those signals.

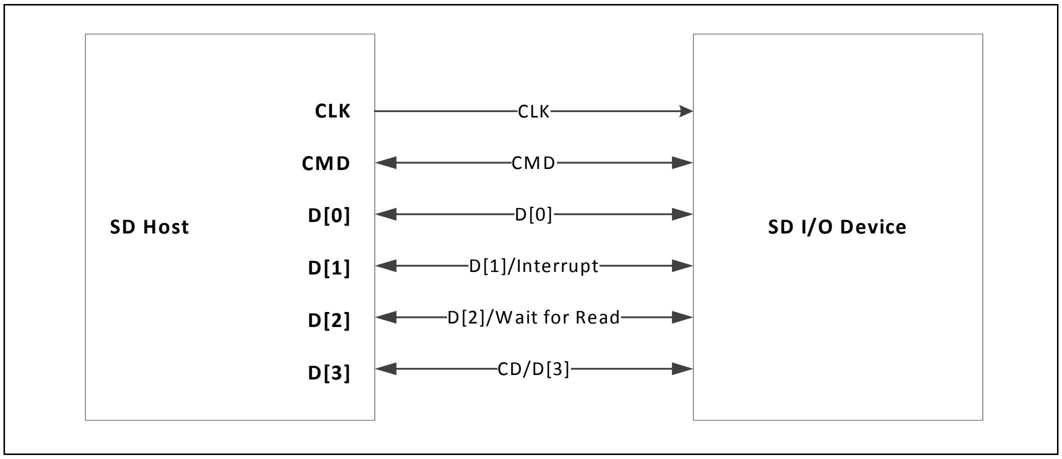
Figure 20. SD Memory Card Bus Topology



12.3.4.2 SDIO 3.0 Interface

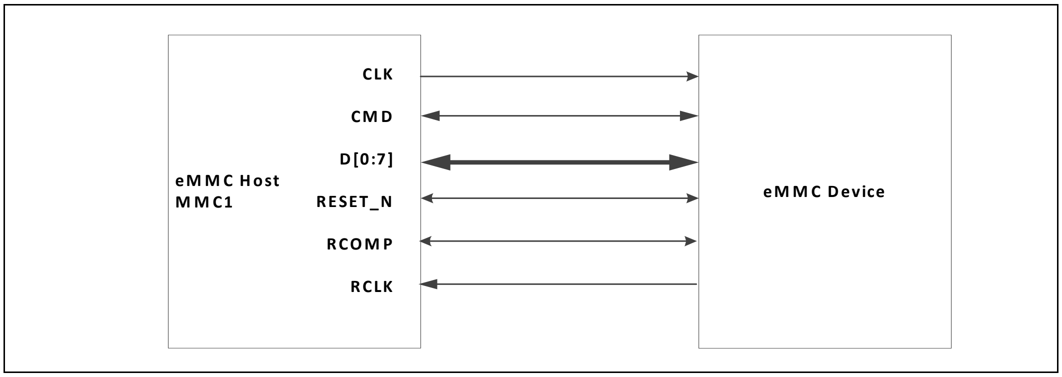
The SDIO interface is the very much like the SD card interface. The SoC supports one SDIO device.

Figure 21. SDIO Device Bus Topology



12.3.4.3 eMMC 4.51 Interface

Figure 22. eMMC Interface



The standard offers performance enhancement features, including HS400 support and has an interface bandwidth of 400 MByte/sec.

The command protocol is significantly improved with Packed Commands (the ability to group a series of commands in a single data transaction), Context ID (grouping different memory transactions under a single ID so the device can understand that they are related), and Data Tag (tagging specific write transactions so they can be prioritized and targeted to a memory region with higher performance and better reliability).

The v4.51 standard also adds provision for volatile data cache, which can greatly reduce the latency between data transactions to improve performance.

12.4 References

The controller is configured to comply with:

- SD Specification Part 01 Physical Layer Specification version 3.00, April 16, 2009.
- SD Specification Part E1 SDIO Specification version 3.00, December 16, 2010.
- SD Specification Part A2 SD Host Controller Standard Specification version 3.00, February 18, 2010.
- SD Specification Part 03 security Specification version 1.01, April 15, 2001.
- Embedded MultiMedia Card (eMMC) Product Standard v4.51, JESD84-A5.

12.5 Register Map

For more information on SoC Storage registers refer Intel® Atom™ Z8000 Processor Series Datasheet (Volume 2 of 2), Doc ID:332066.

§

13 USB Controller Interfaces

USB Controller contains xHCI host controller that supports xHCI framework and USB1/2/3 specifications. And it has xDCI controller block for device only mode functionality. These 2 controllers will use an integrated mux to select between the 2 modes. All of this functionality is located in xDCI Controller.

13.1 SoC Supports

- Two (2) Super Speed Inter-Chip (SSIC) ports
- Three (3) Super Speed (SS) ports [Backward Compatible of USB 2.0 HS/FS/LS]
- One (1) Super Speed (SS) OTG port
- Two (2) High Speed Inter-Chip (HSIC) ports

Note: SoC can support the 4th SS port when OTG port is in Host mode.

13.2 Signal Descriptions

Refer Chapter 2, "Physical Interfaces" for additional details.

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin.
- **Direction**: The buffer direction can be either input, output, or I/O (bidirectional).
- **Type:** The buffer type found in Chapter 19, "Electrical Specifications".
- **Description**: A brief explanation of the signal's function.

Table 82. USB SSIC Signals

Signal Name	Direction /Type	Description
USB_SSIC_RX_P/ N[0,1]	I/O/ SSIC PHY	**Receiver serial data inputs:** High-speed serialized data inputs.
USB_SSIC_TX_P/ N[0,1]	I/O/ SSIC PHY	**Transmitter serial data outputs:** High-Speed Serialized data outputs.
USB_SSIC_RCOMP _P/N	I / SSIC PHY	**Resistor Compensation:** An external resistor of 90 Ohm ±1% must be connected between the RCOMP pads.

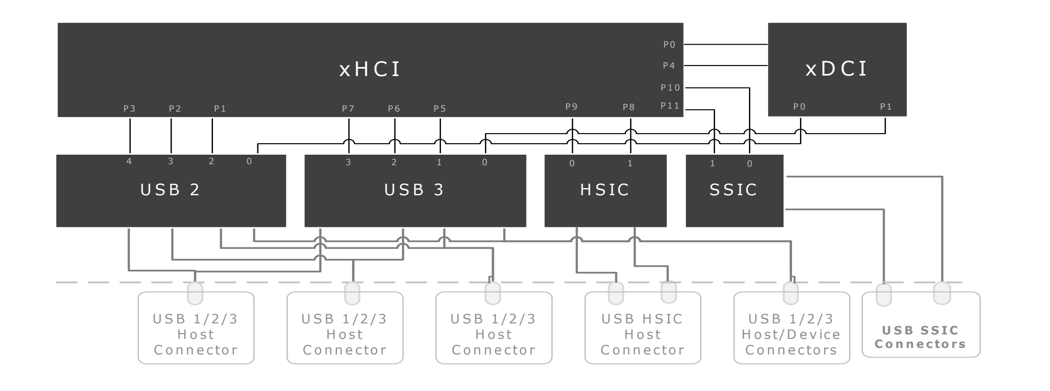
Table 83. USB Signals

Signal Name	Direction /Type	Description
USB3_TXP/ N[0:3]	O USB3 PHY	**Transmitter serial data outputs:** High-Speed Serialized data outputs.
USB3_RXP/ N[0:3]	I USB3 PHY	**Receiver serial data inputs:** High-speed serialized data inputs.
USB3_RCOMP_P /N	I USB3 PHY	**Resistor Compensation:** An external resistor of 402 Ohm ±1% must be connected between the RCOMP pads.
USB_DP/N[0:3]	I/O USB2 PHY	**USB2 Data:** High speed serialized data I/O.
USB_RCOMP	O USB2 PHY	**Resistor Compensation:** An external resistor of 113 Ohm ±1% must be connected between pin and GND.
USB_OTG_ID	I/O USB2 PHY	**OTG ID:** Pin out to detect the OTG ID.
USB_PLL_MON	O USB2 PHY	USB High Speed Observation
USB_VBUSSNS	I/O USB2 PHY	OTG Interface: VBUS_Sense

Table 84. HSIC Signals

Signal Name	Direction /Type	Description
USB_HSIC[0:1]_DATA	I/O HSIC Buffer	**HSIC Data**.
USB_HSIC[0:1]_STROBE	I/O HSIC Buffer	**HSIC Strobe**
USB_HSIC_RCOMP	I/O HSIC Buffer	**Resistor Compensation:** RCOMP for HSIC buffer. Resistor: 45Ohm +/-1% connected between USB_HSIC_RCOMP and ground.

Figure 23. xHCI Port Mapping



13.3 USB 3.0 xHCI (Extensible Host Controller Interface)

The xHCI compliant host controller can control up to 2 SSIC, 3 USB3.0 ports. USB3.0 being backward compatible to support USB2.0. It supports devices conforming to USB 1.x to 3.0 at bit rates up to 5 Gbps.

13.3.1 USB 3.0 Host Features

The USB 3.0 Super Speed data interface is a four wire differential (TX and RX pairs) interface that supports simultaneous bi-directional data transmission. The interface supports a bit rate of 5 Gbps with a maximum theoretical data throughput over 3.2 Gbps due to 8b/10b symbol encoding scheme and protocol overhead (link flow control, packet framing and protocol overhead).

Low Frequency Periodic signaling (LFPS) is used to communicate initialization, training and power management information across a link that is in low power link state without using Super Speed signaling. This reduces power consumption.

13.3.1.1 USB SSIC

- Supports the SuperSpeed protocol only as defined in [USB 3.0].
- Optimized for Power, Area, Cost and EMI robustness.
- Supports 2 ports of 1 lane each.

13.3.1.2 USB 3.0

- Supported by xHCI software host controller interface.
- USB3 port disable.
- Supports local dynamic clock gating and trunk clock gating.
- Supports USB 3.0 LPM (U0, U1, U2, and U3) and also a SS Disabled low power state.

- Supports USB3 Debug Device.
- Supports IVCAM(USB PC Camera).

13.3.2 USB HSIC Features

HSIC is a 2-signal (strobe and data) source synchronous serial interface for on board inter-chip USB communication. The interface uses 240 MHz DDR signaling to provide High-Speed 480 Mb/s USB transfers which are 100% host driver compatible with traditional USB cable connected topologies. Full Speed (FS) and Low Speed (LS) USB transfers are not directly supported by the HSIC interface.

Major feature and performance highlights are as follows:

- Supported by xHCI software host controller interface
- High-Speed 480 Mb/s data rate only.
- Source-synchronous serial interface.
- Power is only consumed when a transfer is in progress.
- No Plug and Play support.
- No hot plug removal/attach.
- Signals driven at 1.2V standard LVCMOS levels.
- Designed for low power applications.
- Support for two host ports compliant to High Speed Inter-Chip Supplement (HSIC) to the USB 2.0 Specification. (USB 2.0).
- Clock request/ack mechanism.

13.4 USB 3.0 xDCI (Extensible Device Controller Interface)

The xDCI compliant Device controller can control up to 1 USB3.0 OTG port. USB3 being backward compatible to support USB2.0. It supports devices conforming to USB 1.x to 3.0 at bit rates up to 5 Gbps.

13.5 References

USB 3.0 Specification

USB 2.0 Specification (Includes High-Speed Inter-Chip USB Electrical Specification)

13.5.1 Host Controller Specifications

Extensible Host Controller Interface (xHCI) Specification for USB 3.0 version 1.0.

13.6 Register Map

For more information on USB Controller Interfaces registers refer Intel® Atom™ Z8000 Processor Series Datasheet (Volume 2 of 2), Doc ID:332066.

§

14 Low Power Engine (LPE) for Audio (I^2S)

Low Power Engine for Audio provides acceleration for common audio and voice functions. The voice and audio engine provides a mechanism for rendering audio and voice streams and tones from the operating system, applications to an audio or voice codec, and ultimately to the speaker, headphones, or Bluetooth* headsets.

Audio streams in the SoC can be encoded and decoded by the Low Power Engine (LPE) in the Audio subsystem.

LPE Audio provides three external I^2S audio interfaces.

14.1 Signal Descriptions

Refer Chapter 2, "Physical Interfaces" for additional details.

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin.
- **Direction**: The buffer direction can be either input, output, or I/O (bidirectional).
- **Type:** The buffer type found in Chapter 19, "Electrical Specifications".
- **Description**: A brief explanation of the signal's function.

Table 85. LPE Signals

Signal Name	Direction /Type	Description
LPE_I2S[2:0]_CLK	I/O	Clock signal for I^2S
LPE_I2S[2:0]_FRM	I/O	Frame select signal for I^2S
LPE_I2S[2:0]_DATAIN	I/O	RX data for I^2S
LPE_I2S[2:0]_DATAOUT	I/O	TX data for I^2S

NOTE: All LPE signals are muxed and may be used by other functions.

14.2 Features

The LPE Audio Subsystem consists of the following:

- Integrated, power-efficient 32-bit architecture core with 24-bit audio processing instructions.
- Core processing speeds up to 343 MHz.
- Closely Coupled Memories (CCMs)
  - — 80 KB Instruction RAM

- — 160 KB Data RAM
- — 48 KB Instruction Cache
- — 96 KB Data Cache

- Very low-power consumption coupled with high-fidelity 24-bit audio.
- Dual-issue, static, super-scalar VLIW processing engine.
- Mode-less switching between 16-, 24-, and 64-bit dual-issue instructions.
- Dual MACs which can operate with 32 x 16-bit and/or 24 x 24-bit operands.
- Inter-Process Communication (IPC) mechanism to communicate with the SoC Processor Core including 4 KB mailbox memory.
- Flexible audio interfaces include three SSPs with I^{2}S port functionality for BI-directional audio transfers.
  - — I^{2}S mode supports PCM payloads
  - — Frame counters for all I^{2}S ports
- High Performance DMA
  - — DMA IP to support multiple outstanding transactions
  - — Interleaved scatter-gather support for Audio DMA transfers
- Clock switching logic including new frequency increments.
- External timer function with an always running clock.

The LPE core runs at a peak clock frequency of 343 MHz and has dedicated on-chip program and data memories and caches. The LPE core can access shared SRAM blocks, and external DRAM through OCP fabric. It communicates with audio peripherals using the audio sub-fabric, and employs Inter-Processor Communication (IPC) mechanism to communicate with the SoC Processor Core.

The Audio subsystem includes two OCP-based DMA engines. These DMA engines support single and multi-block transfers. They can be configured to transfer data between DRAM and audio CCMs or transfer data between CCMs and the audio peripheral interfaces.

All these interfaces are peripherals in the Audio subsystem. LPE, LPE DMA, or the SoC processor core may access the peripherals during normal operation. The PMC may access all peripherals during specific tasks such as at boot time or during power state changes.A complete audio solution based on an internal audio processing engine which includes several I^{2}S-based output ports.

The audio core used is a dedicated audio DSP core designed specifically for audio processing (decoding, post-processing, mixing, etc.)

Note: LPE requires systems with more than 512 MB memory. This is required since the LPE firmware must reside at a stolen memory location on 512 MB boundaries below 3 GiB. The LPE firmware itself is ~1 MB, and is reserved by BIOS for LPE use.

14.2.1 Audio Capabilities

14.2.1.1 Audio Decode

Audio core supports decoding of the following formats:

- MP3
- AAC-LC
- HE-AAC v1/2
- WMA9,10, PRO, Lossless, Voice
- MPEG layer 2
- RealAudio
- OggVorbis
- FLAC
- DD/DD+

14.2.1.2 Audio Encode

Audio core supports encoding of the following formats:

- MP3
- AAC-LC
- WMA
- DD-2channel

14.3 Clocks

14.3.1 Clock Frequencies

Table 86 shows the clock frequency options for the Audio functional blocks.

Table 86. Clock Frequencies

Clock	Frequency	Notes
Audio core	343/250/200 MHz/100/ 50 MHz/2x Osc/Osc 50(RO)/100(RO)	Audio input clock trunk. CCU drives one of several frequencies as noted.
DMA 0	50/OSC	DMA clock
DMA1	50/OSC	DMA clock
Audio fabric clock	50/OSC	Fabric clock derived from audio core clock

Table 86. Clock Frequencies

Clock	Frequency	Notes
SSP0 Clock	Fabric side: 50/OSC Link side: Up to 19.2 MHz	SSP0 clock domains
SSP1 Clock	Fabric side: 50/OSC Link side: Up to 19.2 MHz	SSP1 clock domains
SSP2 Clock	Fabric side: 50/OSC Link side: Up to 19.2 MHz	SSP2 clock domains

14.3.2 38.4 MHz Clock for LPE

38.4 MHz, the 2X OSC clock, is added to increase MIPS for low power MP3 mode. This frequency will be supplied by the clock doubler internal to the SoC’s Clock Control Unit.

14.3.3 Calibrated Ring Osc (50/100 MHz) Clock for LPE

A calibrated Ring Oscillator in the CCU_SUS provides a 50Mhz or an 100Mhz clock as another option for higher MIPS for low power MP3 mode. It is expected that this will be required to support decode of HE-AAC streams in the low power mode.

14.3.4 Cache and CCM Clocking

Data CCM, Data cache, Instruction CCM, and Instruction Cache run off of the LPE clock. These memories are in a single clock domain.

Note: All Data CCM and Instruction CCM run in the same clock domain.

14.4 SSP (I^{2}S)

The SoC audio subsystem consists of the LPE Audio Engine and three Synchronous Serial Protocol (SSP) ports. These ports are used in PCM mode and enable simultaneous support of voice and audio streams over I^{2}S. The SoC audio subsystem also includes two DMA controllers dedicated to the LPE. The LPE DMA controllers are used for transferring data between external memory and CCMs, between CCMs and the SSP ports, and between CCMs. All peripheral ports can operate simultaneously.

The Enhanced SSP Serial Ports are full-duplex synchronous serial interfaces. They can connect to a variety of external analog-to-digital (A/D) converters, audio, and telecommunication codecs, and many other devices which use serial protocols for transferring data. Formats supported include National* Microwire, Texas Instruments* Synchronous Serial Protocol (SSP), Motorola* Serial Peripheral Interface (SPI) protocol and a flexible Programmable Serial Port protocol (PSP).

The Enhanced SSPs operate in master mode (the attached peripheral functions as a slave) or slave mode (the attached peripheral functions as a master), and support serial bit rates from 0 to 6.5 Mbps, dependent on the input clock. Serial data formats range from 4 to 32-bits in length. Two on-chip register blocks function as independent FIFOs for transmit and receive data.

FIFOs may be loaded or emptied by the system processor using single transfers or DMA burst transfers of up to the FIFO depth. Each 32-bit word from the bus fills one entry in a FIFO using the lower significant bits of a 32-bit word.

14.4.1 Features

The SSP port features are:

- Inter-IC Sound (I^2S) protocols, are supported by programming the Programmable Serial Protocol (PSP).
- One FIFO for transmit data (TXFIFO) and a second, independent, FIFO for receive data (RXFIFO), where each FIFO is 16 samples deep x 32 bits wide.
- Data sample sizes from 8, 16, 18, or 32 bits.
- 6.5 Mbps maximum serial bit-rate in both modes: master and slave.
- Clock master or slave mode operations.
- Receive-without-transmit operation.
- Network mode with up to eight time slots for PSP formats, and independent transmit/receive in any/all/none of the time slots.
- After updating SSP configuration, for example active slot count, the SSP will need to be disabled and enabled again. In other words, a SSP will not function correctly if a user changes the configuration setting on the fly.

14.5 Register Map

For more information on Low Power Engine (LPE) for Audio (I2S) registers refer Intel® Atom™ Z8000 Processor Series Datasheet (Volume 2 of 2), Doc ID:332066.

§

15 Intel® Trusted Execution Engine (Intel® TXE)

This chapter describes the security components and capabilities. The security system contains an Intel® TXE and additional hardware security feature that enable a secure and robust platform.

15.1 Features

15.1.1 Security Features

Intel® TXE in the SoC is responsible for supporting and handling security related features.

- 32-bit RISC processor.
- 256KB Data/Code RAM accessible only to the Intel® TXE.
- 128KB On Chip Mask ROM for storage of Intel® TXE code.
- Inter-Processor Communication for message passing between the Host CPU and Intel® TXE.
- 64 byte input and output command buffers.
- 256 byte shared payload (enables 2048-bit keys to be exchanged as part of the command).
- Multiple context DMA engine to transfer data between Host CPU address domain (System memory) and the Intel® TXE; programmable by the Intel® TXE CPU only.
- Secure I^{2}C interface to NFC using master I^{2}C block integrated into the Intel TXE -IP. Secure GPIOs to support input alert and two GP Outputs.

15.1.1.1 HW Accelerators

- DES/3DES (ECB, CBC) – 128b ABA key for 3DES Key Ladder Operations.
- Three AES engines - Two fast -128 and one slow- 128/256.
- Exponentiation Acceleration Unit (EAU) for modular exponentiation, modular reduction, large number addition, subtraction, and multiplication.
- SHA1, SHA256/384/512, MD5.

15.1.1.2 FW Utilities and Ciphers

- RSA (with EAU acceleration).
- Flash Write Enable/Disable.
- Comprehensive IPC Command Set.
- Chip Unique Key encryption key wrapping of other platform keys (Flash).

15.1.1.3 Downloadable FW Utilities and Ciphers

- Integrated Theft Deterrence Technology - Intel® Anti-Theft Technology (Intel® AT).
- One Time Programmable (OTP).
- Firmware TPM (fTPM) measured boot.

15.1.2 TXE Interaction with NFC

- The NFC device requests attention from the TXE from GPIO_ALERT pin to a SoC input interrupt pin (GPIO_SUS[8] pin).
- The GPIO block sends the pin value to TXE over a dedicated wire.
- The wire is connected to the TXE clock request mechanism in order to get a clock for sampling the wire. The TXE bridge includes a configuration register which includes an enable bit to qualify the clock request (which allows masking the clock request, in case the GPIO_SUS[8] is not used by NFC), and a polarity bit (which allows selecting whether the a clock request would be set on a high or low value in the wire).)
- The same qualified & polarity configured clock request input is also sent to PMU. In S0ix PMU uses it as a wake request.
- When a clock is available, the wire value is updated to an ICR (SICR31) in TXE bridge.
- TXE Bridge configuration register also includes two bits that allow detection of falling and/or rising edge on the alert pin. They cause an ISR (SISR[31]) to be set. When both ISR and IER bits for the alert are set an interrupt is generated.
- When the TXE is interrupted it parses the interrupt status registers in the TXE Bridge and figures the cause is the NFC device.
- TXE clears the Bridge ISR and sets configuration to detect the next edge on the alert pin.
- In order to use the I2C master, the TXE sets an I2C clock request register in the Bridge.
- The firmware then uses the I2C master to communicate with the NFC device. The firmware configures the I2C master to read up to 33Bytes of data (up to 36 bytes are supported by HW for read/write).
- When the I2C read is completed, the firmware is interrupted. The TXE may then read the data/status and clear the interrupt.
- The firmware repeats read/write sequence’s as many times as it needs.
- When firmware is done with the I2C master, it must poll the controller to check that the I2C bus is idle before writing to the register to remove the I2C clock request, and before any reset of the I2C controller or power gating sequence. Shutting off clock or I2C master before the completion all activity on the bus will hang the I2C device.

§

16 Intel® Sensor Hub

This chapter describes Intel® Sensor Hub.

16.1 Signal Descriptions

Refer Chapter 2, “Physical Interfaces” for additional details.

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin.
- **Direction**: The buffer direction can be either input, output, or I/O (bidirectional).
- **Type:** The buffer type found in Chapter 19, “Electrical Specifications”.
- **Description**: A brief explanation of the signal’s function.

Table 87. ISH Signals

Signal Name	Direction	Description
ISH_I2C1_CLK	I/O	**Clock Lane:** ISH input clock
ISH_I2C1_SDA	I/O	**Data Lane:** ISH Data Lane
ISH_GPIO	I/O	ISH GPIO

16.2 Features

- Acquisition / sampling of sensor data.
- The ability to combine data from individual sensors to create a more complex Virtual sensor that can be directly used by the firmware/OS.
- Low power operation through clock gating of the ISH together with the ability to turn sensors off under control of host SW.
- The ability to operate independently when the host platform is in a low power state(S0-S0i3).
- Power saving features.
- Clock gating and power gating of functional blocks depending on current workloads.

16.2.1 Hardware

- Minute IA microprocessor.
- 384KB on chip Data/Code SRAM accessible only to the ISH.
- 8KB on chip ROM for ISH boot code.

- Inter-Processor Communication for message passing between the Host CPU and Intel® ISH.
  - — Single Command/Doorbell DWORD register and 32 DWORD data registers each direction.
- Inter-Processor Communication for message passing between the Intel® ISH and Intel® TXE for ISH FW load.
  - — Single Command/Doorbell DWORD register and 32 DWORD data registers each direction.
- Inter-Processor Communication for message passing between the Intel® ISH and PMC for ISH power management and ISH TXE communication assistance by PMC.
  - — Single Command/Doorbell DWORD register each direction.
- DMA engine to transfer data between Host CPU address domain (System memory) and the Intel® ISH; programmable by the Intel® ISH CPU only.
- Two I2C interfaces and up to 15 GPIO lines for connecting sensors to ISH.

§

17 Serial IO (SIO) Overview

The Serial I/O (SIO) is a collection of hardware blocks that implement simple but key serial I/O interfaces for platform usage. These hardware blocks include:

- “SIO - I2C Interface”
- “SIO - High Speed UART”
- “SIO - Serial Peripheral Interface (SPI)”
- “SIO - Pulse Width Modulation (PWM)”

17.1 SIO - Serial Peripheral Interface (SPI)

The Serial I/O implements three SPI controllers that supports master mode.

17.1.1 Signal Descriptions

Refer Chapter 2, "Physical Interfaces" for additional details.

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin.
- **Direction**: The buffer direction can be either input, output, or I/O (bidirectional).
- **Type:** The buffer type found in Chapter 19, "Electrical Specifications".
- **Description**: A brief explanation of the signal's function.

Table 88. SPI Interface Signals

Signal Name	Direction /Type	Description
SPI[1,2,3]_CLK	O GPIO	**SPI Clock:** When the bus is idle, the owner will drive the clock signal low.
SPI[1,2,3]_CS[0]_N	O GPIO	**SPI Chip Select 0:** Used as the SPI Chip select 0.
SPI[1,2,3]_CS[1]_N	O GPIO	**SPI Chip Select 1:** Used as the SPI Chip select 1.
SPI[1,2,3]_MISO	I GPIO	**SPI Master IN Slave OUT:** Data input pin for the SoC.
SPI[1,2,3]_MOSI	O GPIO	**SPI Master OUT Slave IN:** Data output pin for the SoC. Operates as a second data input pin for the SoC when in Single Input, Dual Output Fast Read mode.

17.1.2 Features

The following is list of SPI features:

- Single interrupt line.
  - Could be assigned to interrupt PCI INT [A] or ACPISIO INT[1].
- Configurable frame format, clock polarity and clock phase.
- Supports three SPI peripherals only.
- Two Chip selects are supported for each of the 3 SPI controllers.
- Supports master mode only.
- Receive and transit buffers are both 256x32 Bits.
  - The receive buffer has only 1 water mark.
  - The transmit buffer has 2 water marks.

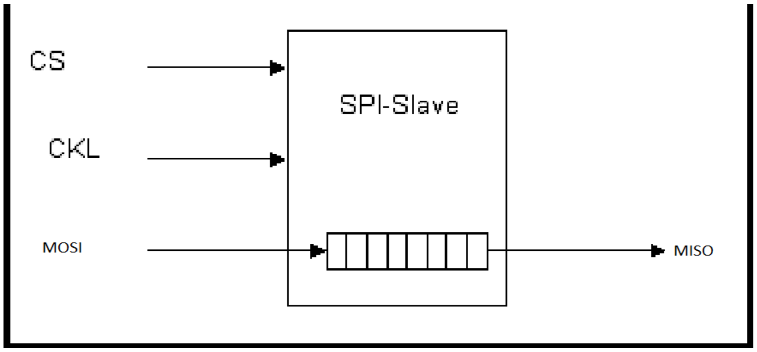
- Supports up to 20 Mbps.

17.1.2.1 General

The Serial Peripheral Interface is used primarily for a synchronous serial communication of host processor and peripherals.

In the standard configuration for a slave device, two control and two data lines are used. The data output MISO serves on the one hand the reading back of data, offers however also the possibility to cascade several devices. The data output of the preceding device then forms the data input for the next IC.

Figure 24. SPI Slave



There is a MASTER and a SLAVE mode. The MASTER device provides the clock signal and determines the state of the chip select lines, i.e. it activates the SLAVE it wants to communicate with. CS and CKL are therefore outputs. The SLAVE device receives the clock and chip select from the MASTER, CS and CKL are therefore inputs. This means there is one master, while the number of slaves is only limited by the number of chip selects.

A SPI device can be a simple shift register up to an independent subsystem. The basic principle of a shift register is always present. Command codes as well as data values are serially transferred, pumped into a shift register and are then internally available for parallel processing.

The SPI requires two control lines (CS and CLK) and two data lines MOSI (Master-Out-Slave-In) and MISO (Master-In-Slave-Out).

17.1.2.2 Data and Control lines for SPI

With CS (Chip-Select) the corresponding peripheral device is selected. This pin is mostly active-low. In the un-selected state the MISO lines are hi-Z and therefore inactive. The master decides with which peripheral device it wants to communicate. The clock line CLK is brought to the device whether it is selected or not. The clock serves as synchronization of the data communication. The majority of SPI devices provide these four lines. Sometimes it happens that MOSI and MISO are multiplexed.

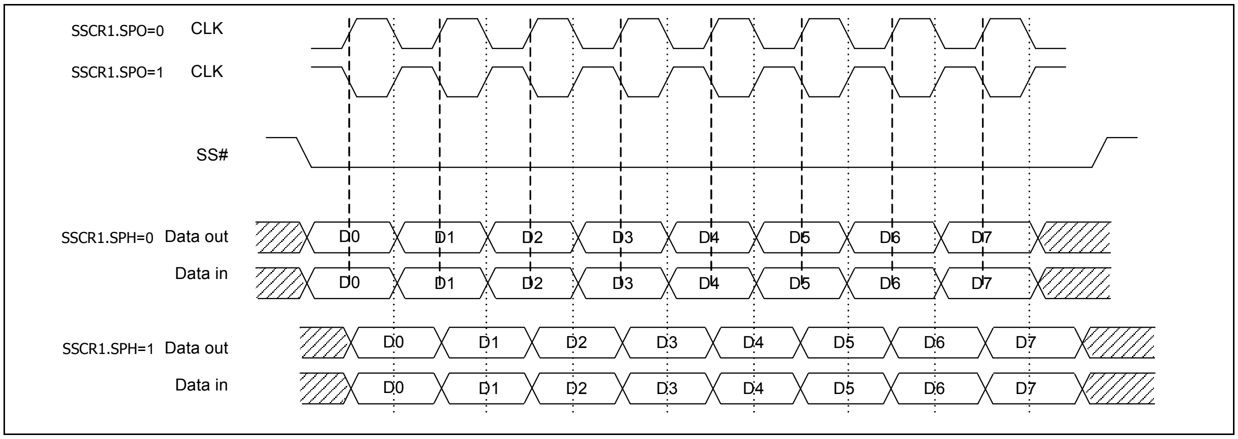
17.1.2.3 SPI Configuration: Clock Phase and Polarity

SPI clock phase and clock polarity overview.

- The SSCR1.SPO polarity setting bit determines whether the serial transfer occurs on the rising edge of the clock or the falling edge of the clock.
  - When SSCR1.SPO = 0, the inactive or idle state of SPI1_CLK is low.
  - When SSCR1.SPO = 1, the inactive or idle state of SPI1_CLK is high.
- The SSCR1.SPH phase setting bit selects the relationship of the serial clock with the slave select signal.
  - When SSCR1.SPH = 0, SPI1_CLK is inactive until one cycle after the start of a frame and active until 1/2 cycle after the end of a frame.
  - When SSCR1.SPH = 1, SPI1_CLK is inactive until 1/2 cycle after the start of a frame and active until one cycle after the end of a frame.

Below figure shows an 8-bit data transfer with different phase and polarity settings.

Figure 25. Clock Phase and Polarity



In a single frame transfer, the SPI controller supports all four possible combinations for the serial clock phase and polarity.

The combinations of polarity and phases are referred to as modes which are commonly numbered according to the following convention, with SSCR1.SPO as the high order bit and SSCR1.SPH as the low order bit.

Table 89. SPI Modes

Mode	SSCR1.SPO	SSCR1.SPH
0	0	0
1	0	1
2	1	0
3	1	1

17.2 SIO - I^{2}C Interface

The SoC supports 7 instances of I^{2}C controller. Both 7-bit and 10-bit addressing modes are supported. These controllers operate in master mode only.

17.2.1 Signal Descriptions

I^{2}C is a two-wire bus for inter-IC communication. Data and clock signals carry information between the connected devices. The following is the I^{2}C Interface. The SoC supports 7 I^{2}C interfaces for general purpose to control external devices. The I^{2}C signals are muxed over GPIOs.

Refer Chapter 2, "Physical Interfaces" for additional details.

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin.
- **Direction**: The buffer direction can be either input, output, or I/O (bidirectional).
- **Type:** The buffer type found in Chapter 19, "Electrical Specifications".
- **Description**: A brief explanation of the signal's function.

Table 90. I^{2}C[6:0] Signals

Signal Name	Direction /Type	Description
I2C[6:0]_DATA	I/O/ GPIOMV, MS, I2C	**I^{2}C Serial Data** *These signals are muxed and may be used by other functions.*
I2C[6:0]_CLK	I/O/ GPIOMV, MS, I2C	**I^{2}C Serial Clock** *These signals are muxed and may be used by other functions.*

17.2.2 NFC I^{2}C Interface Signals

Table 91. NFC I^{2}C Interface Signals

Signal Name	Direction/ Type	Description
NFC_I2C_DATA	I/O/ GPIOMV, MS, I2C	**NFC I^{2}C Serial Data** *These signals are muxed and may be used by other functions.*
NFC_I2C_CLK	I/O/ GPIOMV, MS, I2C	**NFC I^{2}C Serial Clock** *These signals are muxed and may be used by other functions.*
GPIO_ALERT	I/O/ GPIOMV, MS	**ALERT pin for NFC** *These signals are muxed and may be used by other functions.*

17.2.3 Features

17.2.3.1 I^{2}C Protocol

The I^{2}C bus is a two-wire serial interface, consisting of a serial data line and a serial clock. These wires carry information between the devices connected to the bus. Each device is recognized by a unique address and can operate as either a “transmitter” or “receiver,” depending on the function of the device. Devices are considered slaves when performing data transfers, as the SoC will always be a Master. A master is a device which initiates a data transfer on the bus and generates the clock signals to permit that transfer. At that time, any device addressed is considered a slave.

- The SoC is always the I^{2}C master; and it supports multi-master mode.
- The SoC can support clock stretching by slave devices.
- The I2Cx_DATA line is a bidirectional signal and changes only while the I2Cx_CLK line is low, except for STOP, START, and RESTART conditions.
- The output drivers are open-drain or open-collector to perform wire-AND functions on the bus.
- The maximum number of devices on the bus is limited by the maximum capacitance specification of 400 pF. Refer Chapter 19, “Electrical Specifications” for more details.
- Data is transmitted in byte packages.

17.2.3.2 I^{2}C Modes of Operation

The I^{2}C module can operate in the following modes:

- Standard mode (with a bit rate up to 100 Kb/s).
- Fast mode (with a bit rate up to 400 Kb/s).
- Fast Mode plus mode (with a bit rate up to 1 Mb/s).
- High-speed mode (with a bit rate up to 1.7 Mb/s).

The I^{2}C can communicate with devices only using these modes as long as they are attached to the bus. Additionally, high speed mode, fast mode plus and fast mode devices are downward compatible.

- High-speed mode devices can communicate with fast mode and standard mode devices in a mixed speed bus system.
- Fast mode devices can communicate with standard mode devices in a 0–100 Kb/s I^{2}C bus system.

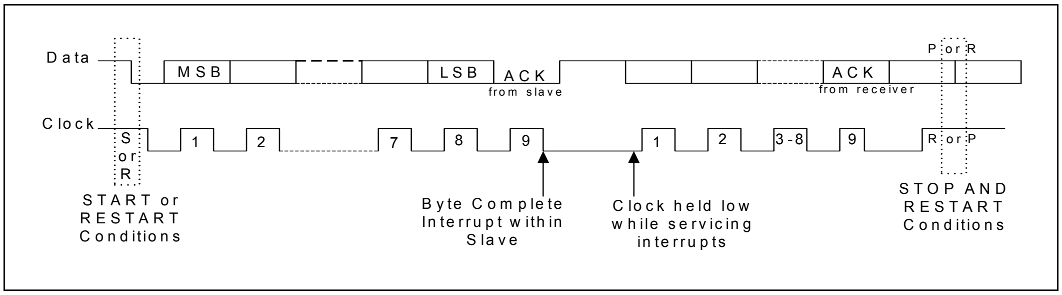
However, according to the I^{2}C specification, standard mode devices are not upward compatible and should not be incorporated in a fast-mode I^{2}C bus system since they cannot follow the higher transfer rate and unpredictable states would occur.

Refer Table 1 for more information on the I2C interface speed for different Sku’s.

17.2.3.3 Functional Description

- The I^2C master is responsible for generating the clock and controlling the transfer of data.
- The slave is responsible for either transmitting or receiving data to/from the master.
- The acknowledgement of data is sent by the device that is receiving data, which can be either a master or a slave.
- Each slave has a unique address that is determined by the system designer:
  - When a master wants to communicate with a slave, the master transmits a START/RESTART condition that is then followed by the slave's address and a control bit (R/W), to determine if the master wants to transmit data or receive data from the slave.
  - The slave then sends an acknowledge (ACK) pulse after the address.
- If the master (master-transmitter) is writing to the slave (slave-receiver)
  - The receiver gets one byte of data.
  - This transaction continues until the master terminates the transmission with a STOP condition.
- If the master is reading from a slave (master-receiver)
  - The slave transmits (slave-transmitter) a byte of data to the master, and the master then acknowledges the transaction with the ACK pulse.
  - This transaction continues until the master terminates the transmission by not acknowledging (NACK) the transaction after the last byte is received, and then the master issues a STOP condition or addresses another slave after issuing a RESTART condition. This behavior is illustrated in below figure.

Figure 26. Data Transfer on the I^2C Bus


Data
Clock
MSB
LSB
ACK
from slave
ACK
from receiver
P or R
S
or
R
1
2
7
8
9
1
2
3-8
9
R or P
START or RESTART Conditions
Byte Complete Interrupt within Slave
Clock held low while servicing interrupts
STOP AND RESTART Conditions

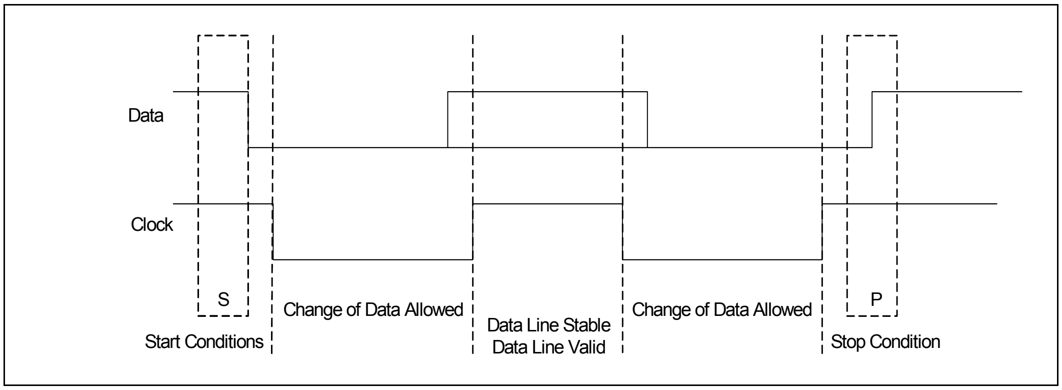
17.2.3.3.1 START and STOP Conditions

When the bus is idle, both the clock and data signals are pulled high through external pull-up resistors on the bus.

When the master wants to start a transmission on the bus, the master issues a START condition.

- This is defined to be a high-to-low transition of the data signal while the clock is high.
- When the master wants to terminate the transmission, the master issues a STOP condition. This is defined to be a low-to-high transition of the data line while the clock is high. Figure 27 shows the timing of the START and STOP conditions.
- When data is being transmitted on the bus, the data line must be stable when the clock is high.

Figure 27. START and STOP Conditions



The signal transitions for the START/STOP conditions, as depicted above, reflect those observed at the output of the master driving the I^{2}C bus. Care should be taken when observing the data/clock signals at the input of the slave(s), because unequal line delays may result in an incorrect data/clock timing relationship.

17.3 NFC I^{2}C

NFC device requires 1.8V I/Os.

For more information refer "TXE Interaction with NFC"

17.3.1 References

I^{2}C-Bus Specification and User Manual, Revision 03: http://ics.nxp.com/support/documents/interface/pdf/i2c.bus.specification.pdf

17.4 SIO - High Speed UART

The SoC implements two instances of high speed UART controller that support baud rates between 300 and 3686400. Hardware flow control is also supported.

17.4.1 Signal Descriptions

Refer Chapter 2, “Physical Interfaces” for additional details.

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin.
- **Direction**: The buffer direction can be either input, output, or I/O (bidirectional).
- **Type:** The buffer type found in Chapter 19, “Electrical Specifications”.
- **Description**: A brief explanation of the signal’s function.

Table 92. UART 1 Interface Signals

Signal Name	Direction/ Type	Description
UART1_DATAIN	I/O/ GPIOMV, MS	**High-speed UART receive data input**: *This signal is muxed and may be used by other functions.*
UART1_DATAOUT	I/O/ GPIOMV, MS	**High-speed UART transmit data:** *This signal is muxed and may be used by other functions.*
UART1_RTS_N	I/O/ GPIOMV, MS	**High-speed UART request to send:** *This signal is muxed and may be used by other functions.*
UART1_CTS_N	I/O/ GPIOMV, MS	**High-speed UART clear to send:** *This signal is muxed and may be used by other functions.*

Table 93. UART 2 Interface Signals

Signal Name	Direction/ Type	Description
UART2_DATAIN	I/O/ GPIOMV, MS	**High-speed UART receive data input:** *This signal is muxed and may be used by other functions.*
UART2_DATAOUT	I/O/ GPIOMV, MS	**High-speed UART transmit data:** *This signal is muxed and may be used by other functions.*
UART2_RTS_N	I/O/ GPIOMV, MS	**High-speed UART request to send:** *This signal is muxed and may be used by other functions.*
UART2_CTS_N	I/O/ GPIOMV, MS	**High-speed UART clear to send:** *This signal is muxed and may be used by other functions.*

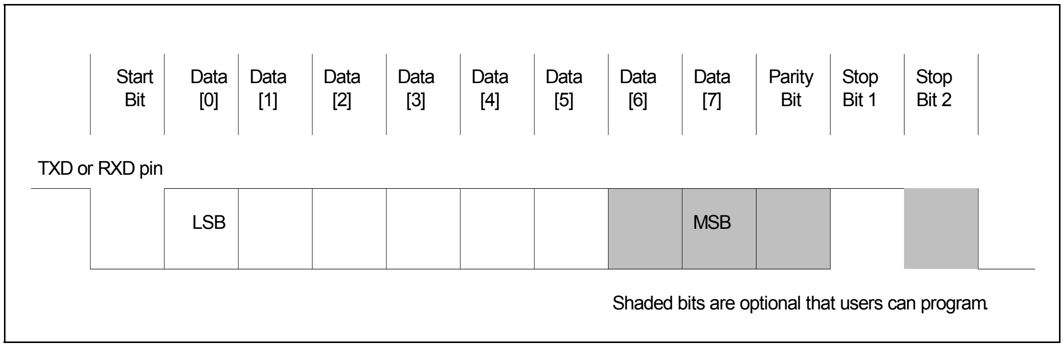
17.4.2 Features

17.4.2.1 UART Function

The UART transmits and receives data in bit frames as shown in Figure 29.

- Each data frame is between 7 and 12 bits long, depending on the size of data programmed and if parity and stop bits are enabled.
- The frame begins with a start bit that is represented by a high-to-low transition.
- Next, 5 to 8 bits of data are transmitted, beginning with the least significant bit. An optional parity bit follows, which is set if even parity is enabled and an odd number of ones exist within the data byte; or, if odd parity is enabled and the data byte contains an even number of ones.
- The data frame ends with one, one-and-one-half, or two stop bits (as programmed by users), which is represented by one or two successive bit periods of a logic one.

Figure 29. UART Data Transfer Flow



Each UART has a Transmit FIFO and a Receive FIFO and each holds 64 characters of data. There are two separate methods for moving data into/out of the FIFOs—Interrupts and Polling.

17.4.2.2 Clock and Reset

The BAUD rate generates from base frequency of 50 MHz.

17.4.2.3 Baud Rate Generator

The baud rates for the UARTs are generated with from the base frequency (Fbase) indicated in Table 94 by programming the DLH and DLL registers as divisor. The hexadecimal value of the divisor is (IER_DLH[7:0]<<8) | RBR_THR_DLL[7:0].

Fbase 44236800 Hz can be achieved by programming the DDS Multiplier as 44,236,800 (in decimal), and DDS Divisor as the system clock frequency in Hz. (50,000,000 in decimal when the system clock frequency is 50 MHz.)

The output baud rate 3686400 is equal to the base frequency divided by thirteen times the value of the divisor, as follows: baud rate = (Fbase) / (13 * divisor). The output baud rate for all other baud rates is equal to the base frequency divided by sixteen times the value of the divisor, as follows: baud rate = (Fbase) / (16 * divisor).

Table 94. Baud Rates Achievable with Different DLAB Settings

DLH,DLL Divisor	DLH,DLL Divisor Hexadecimal	Baud Rate
Fbase 1: 47923200 Hz		
1	0001	3686400
Fbase 2: 44236800 Hz		
1	0001	2764800
3	0003	921600
6	0006	460800
9	0009	307200
12	000C	230400
15	000F	184320
18	0012	153600
24	0018	115200
48	0030	57600
72	0048	38400
144	0090	19200
288	0120	9600
384	0180	7200
576	0240	4800
768	0300	3600
1152	0480	2400
1536	0600	1800
2304	0900	1200
4608	1200	600
9216	2400	300

17.4.3 Use

Each UART has a transmit FIFO and a receive FIFO, each FIFO holding 64 characters of data. Three separate methods move data into and out of the FIFOs: interrupts, DMA, and polled.

17.4.3.1 DMA Mode Operation

17.4.3.1.1 Receiver DMA

The data transfer from the HSUART to host memory is controlled by the DMA write channel. To configure the channel in write mode, channel direction in the channel control register needs to be programmed to “1”. The software need to program the descriptor start address register, descriptor transfer size register, and descriptor control register before starting the channel using the channel active bit in the channel control register.

17.4.3.1.2 Transmit DMA

The data transfer from host memory to HSUART is controlled by DMA read channel. To configure the channel in read mode, channel direction in the channel control register needs to be programmed to “0”. The software need to program the descriptor start address register, descriptor transfer size register, and descriptor control register before starting the channel using the channel active bit in the channel control register.

17.4.3.1.3 Removing Trailing Bytes in DMA Mode

When the number of entries in the Receive FIFO is less than its trigger level, and no additional data is received, the remaining bytes are called Trailing bytes. These are DMAed out by the DMA as it has visibility into the FIFO Occupancy register.

17.4.3.2 FIFO Polled-Mode Operation

With the FIFOs enabled (IIR_FCR.IID0_FIFOE bit set to 1), clearing IER_DLH[7] and IER_DLH[4:0] puts the serial port in the FIFO Polled Operation mode. Because the receiver and the transmitter are controlled separately, either one or both can be in Polled Operation mode. In this mode, software checks Receiver and Transmitter status using the Line Status Register (LSR). The processor polls the following bits for Receive and Transmit Data Service.

17.4.3.2.1 Receive Data Service

The processor checks data ready (LSR.DR) bit which is set when 1 or more bytes remains in the Receive FIFO or Receive Buffer Register (RBR_THR_DLL).

17.4.3.2.2 Transmit Data Service

The processor checks transmit data request LSR.THRE bit, which is set when the transmitter needs data.

The processor can also check transmitter empty LSR.TEMT, which is set when the Transmit FIFO or Holding register is empty.

17.4.3.2.3 Autoflow Control

Autoflow Control uses Clear-to-Send (nCTS) and Request-to-Send (nRTS) signals to automatically control the flow of data between the UART and external modem. When autoflow is enabled, the remote device is not allowed to send data unless the UART asserts nRTS low. If the UART de-asserts nRTS while the remote device is sending data, the remote device is allowed to send one additional byte after nRTS is deasserted. An overflow could occur if the remote device violates this rule. Likewise, the UART is not allowed to transmit data unless the remote device asserts nCTS low. This feature increases system efficiency and eliminates the possibility of a Receive FIFO Overflow error due to long interrupt latency.

Autoflow mode can be used in two ways: Full autoflow, automating both nCTS and nRTS, and half autoflow, automating only nCTS. Full Autoflow is enabled by writing a 1 to bits 1 and 5 of the Modem Control Register (MCR). Auto-nCTS-Only mode is enabled by writing a 1 to bit 5 and a 0 to bit 1 of the MCR register.

17.4.3.2.4 RTS (UART Output)

When in full autoflow mode, nRTS is asserted when the UART FIFO is ready to receive data from the remote transmitter. This occurs when the amount of data in the Receive FIFO is below the programmable threshold value. When the amount of data in the Receive FIFO reaches the programmable threshold, nRTS is de-asserted. It will be asserted once again when enough bytes are removed from the FIFO to lower the data level below the threshold.

17.4.3.2.5 CTS (UART Input)

When in Full or Half-Autoflow mode, nCTS is asserted by the remote receiver when the receiver is ready to receive data from the UART. The UART checks nCTS before sending the next byte of data and will not transmit the byte until nCTS is low. If nCTS goes high while the transfer of a byte is in progress, the transmitter will complete this byte.

17.5 SIO - Pulse Width Modulation (PWM)

The Pulse Width Modulation block allows control the frequency and duty cycle of an output signal. The SoC has 2 instances of the PWM interface.

17.5.1 Signal Descriptions

Refer Chapter 2, "Physical Interfaces" for additional details.

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin.
- **Direction**: The buffer direction can be either input, output, or I/O (bidirectional).
- **Type:** The buffer type found in Chapter 19, "Electrical Specifications".
- **Description**: A brief explanation of the signal's function.

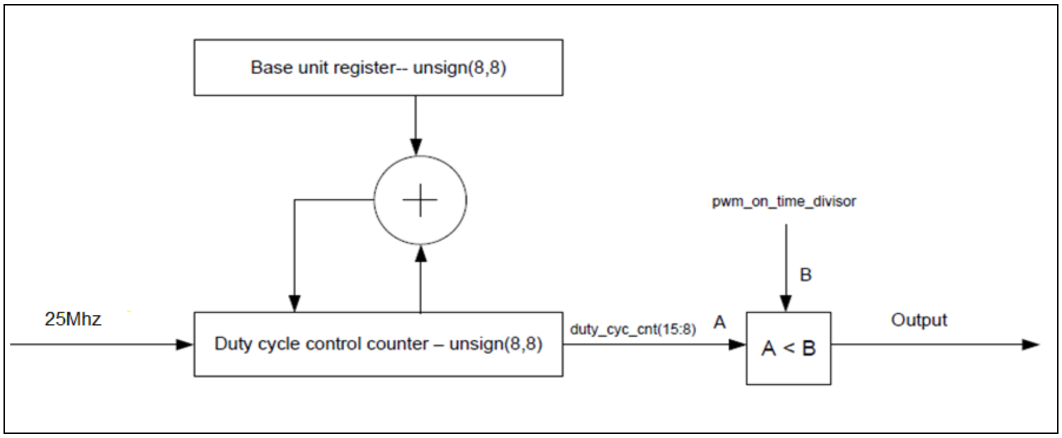
Figure 30. PWM Signals

Signal Name	Direction/ Type	Description
PWM[0]	I/O/ GPIOMV, MS	Pulse Width Modulation output 0.
PWM[1]	I/O/ GPIOMV, MS	Pulse Width Modulation output 1.

17.5.2 Features

The software controls the PWM block by updating the PWMCTRL register and setting the PWMCTRL.PWM_SW_UPDATE bit whenever a change in frequency or duty cycle of the PWM output signal is required. The PWM block applies the new settings at the start of the next output cycle and resets the PWMCTRL.PWM_SW_UPDATE bit. The SoC uses 25 MHz for the counter. Refer Figure 31 for PWM block diagram.

Figure 31. PWM Block Diagram


Base unit register-- unsign(8,8)
+
pwm_on_time_divisor
B
25Mhz
Duty cycle control counter – unsign(8,8)
duty_cyc_cnt(15:8)
A
A < B
Output

There are two controls of the PWM output:

- Frequency is controlled by the PWMCTRL.PWM_BASE_UNIT bits. The PWMCTRL.PWM_BASE_UNIT value is added to a 16 bit counter every clock cycle and the counter roll-over marks the start of a new cycle.
- **Duty cycle** is controlled by the PWMCTRL.PWM_ON_TIME_DIVISOR setting (0 to 255). When the counter rolls-over it is reset and a new cycle starts with the output signal being 0, once the counter reaches the PWMCTRL.PWM_ON_TIME_DIVISOR value the output toggles to 1 and stays high until the counter rolls over.

The PWM block is clocked by the 25 MHz oscillator clock. The output frequency can be estimated with the equation:

- Target frequency = 25MHz * base_unit value/256.

NOTE: The larger the value of base_unit, the larger the error that the PWM output frequency will have with respect to the equation above. For example any Base_unit_value > 128 will result in 12.5 MHz max frequency. Any value between 86 and 128 will result in 8.33 MHz output frequency. And accordingly the larger the base_unit value the smaller duty cycle resolution. Maximum duty cycle resolution is 8 bits.

Table 95 illustrates the output frequency and duty-cycle resolution for different settings of the base_unit_value (when using 25 MHz oscillator clock).

Table 95. Example PWM Output Frequency and Resolution

Target Frequency	Base Unit Value	CLK Cycle Count	Duty Cycle Resolution
12.5 MHz	>=128	1	no resolution
1.07 MHz	11	23	<8 bit resolution
488 kHz	5	51	<8 bit resolution
97.6 kHz	1	256	8 resolution
48.8 kHz	0.5	Theoretically 512 but only 255 available since On Time Divisor is only 8b	>8bit
0	0	0	Flat 0 output

17.6 Register Map

For more information on Serial IO registers refer Intel® Atom™ Z8000 Processor Series Datasheet (Volume 2 of 2), Doc ID:332066.

§

18 Platform Controller Unit (PCU) Overview

Platform Controller Unit (PCU) is a collection of HW blocks that are critical for implementing a Windows* compatible platform. These HW blocks include:

- "PCU - Power Management Controller (PMC)"
- "PCU - Fast Serial Peripheral Interface (SPI)"
  - For boot FW and system configuration data Flash storage
- "PCU - Universal Asynchronous Receiver/Transmitter (UART)"
- "PCU - Intel Legacy Block (iLB) Overview"

The PCU also implements some high level configuration features for BIOS/EFI boot.

18.1 Features

The key features of the individual blocks are as follows:

- Universal Asynchronous Receiver/Transmitter (UART)
  - 16550 controller compliant.
  - Reduced Signal Count: TX and RX only.
  - COM1 interface.
- Fast Serial Peripheral Interface (FST_SPI)
  - For SPI Flash, of up to 16MB size per chip select is supported. No other SPI peripherals are supported.
  - Stores boot FW and system configuration data.
  - Supports frequencies of 20 MHz, 33 MHz and 50 MHz.
- Power Management Controller (PMC)
  - Controls many of the power management features present in the SoC.
- Intel Legacy Block (iLB)
  - Supports legacy PC platform features.
  - Sub-blocks include LPC, GPIO, 8259 PIC, IO-APIC, 8254 timers, HPET timers and the RTC.

18.2 PCU - Power Management Controller (PMC)

Power Management Controller (PMC) controls many of the power management features present in the SoC.

18.2.1 Signal Descriptions

Refer Chapter 2, "Physical Interfaces" for additional details.

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin.
- **Direction**: The buffer direction can be either input, output, or I/O (bidirectional).
- **Type:** The buffer type found in Chapter 19, "Electrical Specifications".
- **Description**: A brief explanation of the signal's function.

Table 96. PMC Signals (Sheet 1 of 2)

Signal Name	Direction /Type	Description
PMC_ACPRESENT	I/O/ GPIOMV, MS	**AC Present:** This input pin indicates when the platform is plugged into AC power.
PMC_BATLOW_N	I/O/ GPIOMV, MS	**Battery Low:** An input from the battery to indicate that there is insufficient power to boot the system. Assertion will prevent wake from the S4/S5 state. This signal can also be enabled to cause an SMI_N when asserted. In desktop configurations without a battery, this signal should be tied high to V1P8_S5.
PMC_CORE_PWROK	I/GPIOMV, MS	**Core Power OK:** When asserted, this signal is an indication to the SoC that all of its core power rails have been stable for 10 ms. It can be driven asynchronously. When it is negated, the SoC asserts PMC_PLTRST_N. **NOTE:** It is required that the power rails associated with PCI Express (typically the 3.3V, 5V, and 12V core well rails) have been valid for 99 ms prior to PMC_CORE_PWROK assertion in order to comply with the 100 ms T_{PVPERL} PCI Express 2.0 specification on PMC_PLTRST_N deassertion. **NOTE:** PMC_CORE_PWROK must not glitch, even if PMC_RSMRST_N is low.
PMC_PLTRST_N	I/O/ GPIOMV, MS	**Platform Reset:** The SoC asserts this signal to reset devices on the platform. The SoC asserts the signal during power-up and when software initiates a hard reset sequence through the Reset Control (RST_CNT) register.

Table 96. PMC Signals (Sheet 2 of 2)

Signal Name	Direction /Type	Description
PMC_PWRBTN_N	I/O/ GPIOMV, MS	**Power Button:** The signal will cause SMI_N or SCI to indicate a system request to go to a sleep state. If the system is already in a sleep state, this signal will cause a wake event. If the signal is pressed for more than 4 seconds, this will cause an unconditional transition (power button override) to the S5 state. Override will occur even if the system is in the S4 states. This signal has an internal pull-up resistor and has an internal 16 ms de-bounce on the input.
PMC_RSMRST_N	I/GPIOMV, MS	**Resume Well Reset:** Used for resetting the resume well. An external RC circuit is required to guarantee that the resume well power is valid prior to this signal going high.
PMC_RSTBTN_N	I/O/ GPIOMV, MS	**System Reset:** This signal forces an internal reset after being debounced. *This signal is muxed and may be used by other functions.*
PMC_SLP_S0IX_N	I/O/ GPIOMV, MS	**S0ix Sleep Control:** This signal is for power plane control. It can be used to control system power when it is in a S0ix state.
PMC_SLP_S4_N	I/O/ GPIOMV, MS	**S4 Sleep Control:** This signal is for power plane control. It can be used to control system power when it is in a S4 (Suspend to Disk) or S5 (Soft Off) state.
PMC_SUS_STAT_N	I/O/ GPIOMV, MS	**Suspend Status:** This signal is asserted by the SoC to indicate that the system will be entering a low power state soon. This can be monitored by devices with memory that need to switch from normal refresh to suspend refresh mode. It can also be used by other peripherals as an indication that they should isolate their outputs that may be going to powered-off planes. *This signal is muxed and may be used by other functions.*
PMC_SUSCLK	I/O/ GPIOMV, MS	**Suspend Clock:** This 32 kHz clock is an output of the RTC generator circuit for use by other chips for refresh clock. *This signal is muxed and may be used by other functions.*
PMC_SUSPWRDNACK	I/O/ GPIOMV, MS	**Suspend Power Down Acknowledge:** Asserted by the SoC when it does not require its Suspend well to be powered. This pin requires a pull-up to UNCORE_V1P8_G3. *This signal is muxed and may be used by other functions.*

18.2.2 Features

18.2.2.1 Sx-G3-Sx, Handling Power Failures

Depending on when the power failure occurs and how the system is designed, different transitions could occur due to a power failure.

The GEN_PMCON1.AG3E bit provides the ability to program whether or not the system should boot once power returns after a power loss event. If the policy is to not boot, the system remains in an S5 state (unless previously in S4). There are only two possible events that will wake the system after a power failure.

- **PMC_PWRBTN_N:** PMC_PWRBTN_N is always enabled as a wake event. When RSMRST_N is low (G3 state), the PM1_STS_EN.PWRBTN_STS bit is reset. When the SoC exits G3 after power returns (PMC_RSMRST_N goes high), the PMC_PWRBTN_N signal is already high (because the suspend plane goes high before PMC_RSMRST_N goes high) and the PM1_STS_EN.PWRBTN_STS bit is 0b.
- **RTC Alarm:** The PM1_STS_EN.RTC_EN bit is in the RTC well and is preserved after a power loss. Like PM1_STS_EN.PWRBTN_STS the PM1_STS_EN.RTC_STS bit is cleared when PMC_RSMRST_N goes low.

The SoC monitors both PMC_CORE_PWROK and PMC_RSMRST_N to detect for power failures. If PMC_CORE_PWROK goes low, the GEN_PMCON1.PWR_FLR bit is set. If PMC_RSMRST_N goes low, GEN_PMCON1.SUS_PWR_FLR is set.

Table 97. Transitions Due to Power Failure

State at Power Failure	GEN_PMCON1.AG3E bit	Transition When Power Returns
S0	1 0	S5 S0
S4	1 0	S4 S0
S5	1 0	S5 S0

18.2.2.2 Event Input Signals and Usage

The SoC has various input signals that trigger specific events. This section describes those signals and how they should be used.

18.2.2.2.1 PMC_PWRBTN_N (Power Button)

The PMC_PWRBTN_N signal operates as a “Fixed Power Button” as described in the Advanced Configuration and Power Interface specification. The signal has a 16 ms debounce on the input. The state transition descriptions are included in Table 98.

Note: The transitions start as soon as the PMC_PWRBTN_N is pressed (but after the debounce logic), and does not depend on when the power button is released.

Note: During the time that the PMC_SLP_S4_N signal is stretched for the minimum assertion width (if enabled), the power button is not a wake event. Refer note below for more details.

Table 98. Transitions Due to Power Button

Present State	Event	Transition/Action	Comment
S0/Cx	PMC_PWRBTN_N goes low	SMI_N or SCI generated (depending on PM1_CNT.SCI_EN, PM1_STS_EN.PWRBTN_EN and SMI_EN.GBL_SMI_EN)	Software typically initiates a Sleep state
S4/S5	PMC_PWRBTN_N goes low	Wake Event. Transitions to S0 state	Standard wakeup
G3	PMC_PWRBTN_N pressed	None	No effect since no power Not latched nor detected
S0, S4	PMC_PWRBTN_N held low for at least 4 consecutive seconds	Unconditional transition to S5 state	No dependence on processor or any other subsystem
S0ix	PMC_PWRBTN_N goes low	Wake Event. Transitions to S0 state	PM1_STS_EN.PWRBTN_EN should be set since a SMI/SCI event is required.

Power Button Override Function

If PMC_PWRBTN_N is observed active for at least four consecutive seconds, the state machine should unconditionally transition to the S5 state, regardless of present state (S0–S4), even if the PMC_CORE_PWROK is not active. In this case, the transition to the G2/S5 state should not depend on any particular response from the processor nor any similar dependency from any other subsystem.

The PMC_ PWRBTN_N status is readable to check if the button is currently being pressed or has been released. The status is taken after the de-bounce, and is readable using the GEN_PMCON2.PWRBTN_LVL bit.

Note: The 4 seconds PMC_PWRBTN_N assertion should only be used if a system lock-up has occurred. The 4-second timer starts counting when the SoC is in a S0 state. If the PMC_PWRBTN_N signal is asserted and held active when the system is in a suspend state (S4), the assertion causes a wake event. Once the system has resumed to the S0 state, the 4-second timer starts.

Note: During the time that the SLP_S4_N signal is stretched for the minimum assertion width (if enabled by GEN_PMCON1.S4ASE), the power button is not a wake event. As a result, it is conceivable that the user will press and continue to hold the power button waiting for the system to awake. Since a 4 seconds press of the power button is already defined as an unconditional power down, the power button timer will be forced to inactive while the power-cycle timer is in progress. Once the power-cycle timer has expired, the power button awakes the system. Once the minimum PMC_SLP_S4_N power cycle expires, the power button must be pressed for another 4 to 5 seconds to create the override condition to S5.

18.2.2.2.2 Sleep Button

The Advanced Configuration and Power Interface specification defines an optional sleep button. It differs from the power button in that it only is a request to go from S0 to S4 (not S5). Also, in an S5 state, the power button can wake the system, but the sleep button cannot.

Although the SoC does not include a specific signal designated as a sleep button, one of the GPIO signals can be used to create a "Control Method" sleep button. Refer Advanced Configuration and Power Interface specification for implementation details.

18.2.2.2.3 PME_B0 (PCI Power Management Event Bus 0)

The GPE0a_STS.PME_B0_STS bit exists to implement PME_N-like functionality for any internal device on Bus 0 with PCI power management capabilities.

18.2.2.2.4 PMC_RSTBTN_N Signal

When the PMC_RSTBTN_N pin is detected as active after the 16 ms debounce logic, the SoC attempts to perform a "graceful" reset, by waiting for the relevant internal devices to signal their idleness. If all devices are idle when the pin is detected active, the reset occurs immediately; otherwise, a counter starts. If at any point during the count all devices go idle the reset occurs. If the counter expires and any device is still active, a reset is forced upon the system even though activity is still occurring.

Once the reset is asserted, it remains asserted for 5 to 6 ms regardless of whether the PMC_RSTBTN_N input remains asserted or not. It cannot occur again until PMC_RSTBTN_N has been detected inactive after the debounce logic, and the system is back to a full S0 state with PMC_PLTRST_N inactive. Note that if RST_CNT.FULL_RST is set then PMC_RSTBTN_N will result in a full power cycle reset.

18.2.2.3 System Power Planes

The system has several independent power planes, as described in Table 99.

Note: When a particular power plane is shut off, it should go to a 0 V level.

Table 99. System Power Planes

Plane	Controlled By	Description
Devices and Memory	PMC_SLP_S4_N	When PMC_SLP_S4_N goes active, power can be shut off to any circuit not required to wake the system from the S4/S5. Since the memory context does not need to be preserved in the S4/S5 state, the power to the memory can also be shut down. S4 and S5 requests are treated the same so no PMC_SLP_S5_N signal is implemented.
Devices	Implementation Specific	Individual subsystems may have their own power plane. For example, GPIO signals may be used to control the power to disk drives, audio amplifiers, or the display screen.
Suspend	PMC_SUSPWRDNACK	The suspend power planes are generally left on whenever the system has a charged main battery or is plugged in to AC power. In some cases, it may be preferable to disable the suspend power planes in S4/S5 states to save additional power. This requires some external logic (such as an embedded controller) to ensure that a wake event is still possible (such as the power button). When the SeC is enabled it is advised that the suspend power planes not be removed. Doing so may result in extremely long Sx exit times since the SeC if forced to consider it a cold boot which may, in turn, cause exit latency violations for software using the TXE.

18.2.2.3.1 Power Plane Control with PMC_SLP_S0IX_N and PMC_SLP_S4_N

The PMC_SLP_S0IX_N output signal can be used to cut power to any systems supplies that are not required during a S0ix system state.

Cutting power to the core may be done using the power supply, or by external FETs on the motherboard.

The PMC_SLP_S4_N output signal can be used to cut power to the system core supply, as well as power to the system memory, since the context of the system is saved on the disk. Cutting power to the memory may be done using the power supply, or by external FETs on the motherboard.

18.2.2.3.2 PMC_SLP_S4_N and Suspend-To-RAM Sequencing

The system memory suspend voltage regulator is controlled by the Glue logic. The PMC_SLP_S4_N signal should be used to remove power to system memory. The PMC_SLP_S4_N logic in the SoC provides a mechanism to fully cycle the power to the DRAM and/or detect if the power is not cycled for a minimum time.

Note: To use the minimum DRAM power-down feature that is enabled by the GEN_PMCON1.S4ASE bit, the DRAM power must be controlled by the PMC_SLP_S4_N signal.

18.2.2.3.3 PMC_CORE_PWROK Signal

When asserted, PMC_CORE_PWROK is an indication to the SoC that its core well power rails are powered and stable. PMC_CORE_PWROK can be driven asynchronously. When PMC_CORE_PWROK is low, the SoC asynchronously asserts PMC_PLTRST_N. PMC_CORE_PWROK must not glitch, even if PMC_RSMRST_N is low.

It is required that the power rails associated with PCI Express have been valid for 99 ms prior to PWROK assertion in order to comply with the 100 ms T_{PVPERL} PCI Express 2.0 specification on PMC_PLTRST_N deassertion.

Note: PMC_RSTBTN_N is recommended for implementing the system reset button. This saves external logic that is needed if the PMC_CORE_PWROK input is used. Additionally, it allows for better handling of the processor resets and avoids improperly reporting power failures.

18.2.2.3.4 PMC_BATLOW_N (Battery Low)

The PMC_BATLOW_N input can inhibit waking from S4, and S5 states if there is not sufficient power. It also causes an SMI if the system is already in an S0 state.

18.2.2.4 SMI_N/SCI Generation

Upon any enabled SMI event taking place while the SMI_EN.EOS bit is set, the SoC will clear the EOS bit and assert SMI to the CPU core, which will cause it to enter SMM space. SMI assertion is performed using a Virtual Legacy Wire (VLW) message. Prior system generations (those based upon legacy processors) used an actual SMI_N pin.

Once the SMI message has been delivered, the SoC takes no action on behalf of active SMI events until Host software sets the End of SMI (EOS) bit. At that point, if any SMI events are still active, the SoC will send another SMI message.

The SCI is a level-mode interrupt that is typically handled by an ACPI-aware operating system. In non-APIC systems (which is the default), the SCI IRQ is routed to one of the 8259 interrupts (IRQ 9, 10, or 11). The 8259 interrupt controller must be programmed to level mode for that interrupt.

In systems using the APIC, the SCI can be routed to interrupts IRQs[11:9] or IRQs[23:20]. The interrupt polarity changes depending on whether it is on an interrupt shareable with a PIRQ or not. The interrupt remains asserted until all SCI sources are removed.

Table 100 shows which events can cause an SMI and SCI. Note that some events can be programmed to cause either an SMI or SCI. The usage of the event for SCI (instead of SMI) is typically associated with an ACPI-based system. Each SMI or SCI source has a corresponding enable and status bit.

Table 100. Causes of SMI and SCI (Sheet 1 of 2)

Event	Status Indication[1]	Enable Condition	Interrupt Result			
			SMI_EN. GBL_SMI_EN=1b		SMI_EN. GBL_SMI_EN=0b	
			PM1_CNT.SCI_EN=1b	PM1_CNT.SCI_EN=0b	PM1_CNT.SCI_EN=1b	PM1_CNT.SCI_EN=0b
Power Button Override[3]	PM1_STS_EN. PWRBTNOR_STS	None	SCI	None	SCI	None
RTC Alarm	PM1_STS_EN. RTC_STS	PM1_STS_EN_EN. RTC_EN=1b	SCI	SMI	SCI	None
Power Button Press	PM1_STS_EN. PWRBTN_STS	PM1_STS_EN_EN. PWRBTN_EN=1b	SCI	SMI	SCI	None
SMI_EN.BIOS_RLS bit written to 1b[4]	PM1_STS_EN. GBL_STS	PM1_STS_EN_EN. GBL_EN=1b	SCI			
ACPI Timer overflow (2.34 seconds)	PM1_STS_EN. TMROF_STS	PM1_STS_EN_EN. TMROF_EN =1b	SCI	SMI	SCI	None
GPI[n][9]	GPE0a_STS. CORE_GPIO_STS[n][2] or GPE0a_STS. SUS_GPIO_STS[n][2]	GPIO_ROUT[n] = 10b & GPE0a_EN. CORE_GPIO_EN[n][2]=1b or GPE0a_EN. SUS_GPIO_EN[n][2]=1b	SCI	None	SCI	None
Internal, Bus 0, PME-Capable Agents (PME_B0)	GPE0a_STS. PME_B0_STS	GPE0_EN. PME_B0_EN=1b	SCI	SMI	SCI	None
BATLOW_N pin goes low	GPE0a_STS. BATLOW_STS_N	GPE0_EN. BATLOW_EN=1b	SCI	SMI	SCI	None
Software Generated GPE	GPE0a_STS. SWGPE_STS	GPE0_EN. SWGPE_EN=1b	SCI	SMI	SCI	None
DOSCI message from GUNIT[5]	GPE0a_STS. GUNIT_STS	None (enabled by G-Unit[8])	SCI	None	SCI	None
ASSERT_SMI message from SPI[5]	SMI_STS. SPI_SMI_STS	None (enabled by SPI controller)	SMI		None	
ASSERT_IS_SMI message from USB	SMI_STS. USB_IS_STS	SMI_EN. USB_IS_SMI_EN=1b	SMI		None	
ASSERT_SMI message from USB	SMI_STS.USB_STS	SMI_EN. USB_SMI_EN=1b	SMI		None	
ASSERT_SMI message from iLB[5]	SMI_STS. ILB_SMI_STS	None (enabled by iLB)	SMI		None	
Periodic timer expires	SMI_STS. PERIODIC_STS	SMI_EN. PERIODIC_EN=1b	SMI		None	

Table 100. Causes of SMI and SCI (Sheet 2 of 2)

Event	Status Indication[1]	Enable Condition	Interrupt Result			
			SMI_EN.GBL_SMI_EN=1b		SMI_EN.GBL_SMI_EN=0b	
			PM1_CNT.SCI_EN=1b	PM1_CNT.SCI_EN=0b	PM1_CNT.SCI_EN=1b	PM1_CNT.SCI_EN=0b
WDT first expiration	SMI_STS.TCO_STS	SMI_EN.TCO_EN=1b	SMI		None	
64 ms timer expires	SMI_STS.SWSMI_TMR_STS	SMI_EN.SWSMI_TMR_EN=1b	SMI		None	
PM1_CNT.SLP_EN bit written to 1b	SMI_STS.SMI_ON_SLP_EN_STS	SMI_EN.SMI_ON_SLP_EN=1b	Sync SMI[6]		None	
PM1_CNT.GBL_RLS written to 1b	SMI_STS.BIOS_STS	SMI_EN.BIOS_EN=1b	Sync SMI[6]		None	
DOSMI message from GUNIT[5]	SMI_STS.GUNIT_SMI_STS	None (enabled by G-Unit[8])	SMI		None	
ASSERT_IS_SMI message from iLB[5]	SMI_STS.ILB_SMI_STS	None (enabled by iLB)	Sync SMI[7]		None	
GPI[n][10]	ALT_GPIO_SMI.CORE_GPIO_SMI_STS[n][2] or ALT_GPIO_SMI.SUS_GPIO_SMI_STS[n][2]	GPIO_ROUT[n]=01b & ALT_GPIO_SMI.CORE_GPIO_SMI_EN[n][2]=1b or ALT_GPIO_SMI.SUS_GPIO_SMI_EN[n][2]=1b	SMI		None	
USB Per-Port Registers Write Enable bit is changed from 0b to 1b	UPRWC.WE_STS & SMI_STS.USB_IS_STS	UPRWC.WE_SMI_E=1b & SMI_EN.USB_IS_SMI_EN=1b	Sync SMI[6]		None	

NOTES:

1. Most of the status bits (except otherwise is noted) are set according to event occurrence regardless to the enable bit.
2. GPIO status bits are set only if enable criteria is true. GPIO_ROUT[n]=10b & GPE0a_EN.x_GPIO_EN[n] for GPE0a_STS.x_GPIO_STS[n] (SCI). GPIO_ROUT[n]=01b & ALT_GPIO_SMI. x_GPIO_SMI_EN[n]=1b for ALT_GPIO_SMI.x_GPIO_SMI_STS[n] (SMI).
3. When power button override occurs, the system will transition immediately to S5. The SCI will only occur after the next wake to S0 if the residual status bit (PM1_STS_EN.PWRBTNOR_STS) is not cleared prior to setting PM1_CNT.SCI_EN.
4. PM1_STS_EN.GBL_STS being set will cause an SCI, even if the PM1_CNT.SCI_EN bit is not set. Software must take great care not to set the SMI_ENBIOS_RLS bit (which causes PM1_STS_EN.GBL_STS to be set) if the SCI handler is not in place.
5. No enable bits for these SCI/SMI messages in the PMC. Enable capability should be implemented in the source unit.
6. Sync SMI has the same message opcode toward T-Unit. Special treatment regarding this Sync SMI is holding completion to host till SYNC_SMI_ACK message is received from T-Unit.

7. Sync SMI has the same message opcode toward T-Unit. Special treatment regarding this Sync SMI is holding the SSMI_ACK message to iLB till SYNC_SMI_ACK message is received from T-Unit.
8. The G-Unit is an internal functional sub-block which forms part of the graphics functional block.
9. The GPE0a_STS.CORE_GPIO_STS[31:24] & GPE0a_EN.CORE_GPIO_EN[31:24] register bits correspond to GPIO_S0_SC[7:0]. GPE0a_STS.SUS_GPIO_STS[23:16] & GPE0a_EN.SUS_GPIO_EN[23:16] correspond to GPIO_S5[7:0].
10. The ALT_GPIO_SMI.CORE_GPIO_SMI_STS[31:24] & ALT_GPIO_SMI.CORE_GPIO_SMI_EN[15:8] register bits correspond to GPIO_S0_SC[7:0]. ALT_GPIO_SMI.SUS_GPIO_SMI_STS[23:16] & ALT_GPIO_SMI.SUS_GPIO_SMI_EN[7:0] correspond to GPIO_S5[7:0].

18.2.2.5 Platform Clock Support

The SoC supports up to 6 clocks (PMC_PLT_CLK[5:0])with a frequency of 19.2 MHz. These clocks are available for general system use, where appropriate and each have Control and Frequency register fields associated with them.

18.2.2.6 INIT_N (Initialization) Generation

The INIT_N functionality is implemented as a ‘virtual wire’ internal to the SoC rather than a discrete signal. This virtual wire is asserted based on any one of the events described in below table. When any of these events occur, INIT_N is asserted for 16 PCI clocks and then driven high.

INIT_N, when asserted, resets integer registers inside the CPU cores without affecting its internal caches or floating-point registers. The cores then begin execution at the power on Reset vector configured during power on configuration.

Table 101. INIT_N Assertion Causes

Cause
PORT92.INIT_NOW transitions from 0b to1b.
RST_CNT.SYS_RST = 0b and RST_CNT.RST_CPU transitions from 0b to 1b

18.2.3 References

Advanced Configuration and Power Interface Specification, Revision 3.0: http://www.acpi.info/

18.3 PCU - Fast Serial Peripheral Interface (SPI)

The SoC implements a SPI controller as the interface for BIOS Flash storage. This SPI Flash device is also required to support configuration storage for the firmware for the Trusted Execution Engine. The controller supports a maximum of two SPI Flash devices, using two chip select signals, with speeds of 14.28 MHz, 20 MHz, 25 MHz, 40 MHz or 50 MHz and both have to be Fast SPI. SoC Supports FAST SPI mode.

Note: The default interface speed is 20 MHz.

SPI 'Fast mode' is quad mode.

18.3.1 Signal Descriptions

Refer Chapter 2, "Physical Interfaces" for additional details.

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin.
- **Direction**: The buffer direction can be either input, output, or I/O (bidirectional).
- **Type:** The buffer type found in Chapter 19, "Electrical Specifications".
- **Description**: A brief explanation of the signal's function.

Table 102. SPI Signals

Signal Name	Direction /Type	Description
FST_SPI_CLK	I/O GPIO	**Fast SPI Clock:** When the bus is idle, the owner will drive the clock signal low.
FST_SPI_CS[0]_N	I/O GPIO	**Fast SPI Chip Select 0:** Used as the SPI bus request signal for the first SPI Flash device.
FST_SPI_CS[1]_N	I/O GPIO	**Fast SPI Chip Select 1:** Used as the SPI bus request signal for the second SPI Flash devices.
FST_SPI_CS[2]_N	I/O GPIO	**Fast SPI Chip Select 2:** Used as the SPI bus request signal for the second SPI Flash devices.
FST_SPI_D[3:0]	I/O GPIO	**Fast SPI Data Pad:** Data Input/output pin for the SoC.

Note: All SPI signals are tri-stated when PMC_RSMRST_N and PMC_CORE_PWROK are asserted. FST_SPI_CS[0:2] and FST_SPI_CLK are not tri-stated.

18.3.2 Features

1)Descriptor Mode Capabilities

a)Two modes of operation

i)Descriptor mode with security access restrictions ii)Non-Descriptor mode, no access security restrictions (ICH7 style)

(1)BIOS Only

(2)If the SPI Flash Signature is invalid, the SPI flash operates in non-descriptor mode

a. Supports Flash that is divided into 5 regions and accessible by 3 masters

i)Regions (5)

(1) Flash Descriptor and Chipset Soft Straps

(2) BIOS

(3) TXE

(4) Platform Data

ii)Masters (3)

(1) Host CPU (for BIOS)

(2) TXE

iii)Regions are allowed to extend across multiple Flash components

iv)Regions are aligned to 4K blocks/sectors

b. Chipset Soft Strap region provides the ability to use Flash NVM as an alternative to hardware pullup/pulldown resistors for both SoC and the processor Complex

i)Each Unit that pulls Soft straps from SPI should have a default value that is used if the Flash Signature is invalid.

c. The top of the Flash Descriptor contains the Flash Upper Map

ii)This is used by software to define Flash vendor specific capabilities

d. The top 256B of the flash descriptor is reserved for use by the OEM

2)Security Capabilities

a. Descriptor based Region Restriction: Hardware enforced security restricting master accesses to different regions

i)Flash Descriptor region settings define separate read/write access to each region per master.

ii)Uses SAI for master accesses security checking

(1)Soft Strap+fuse to disable sourceID and SAI checks

iii)Flash Security Override Pin Strap

(1)Removes all descriptor based security

(2)Disables the write protection to the BIOS Protected Range 4 (PR4).

iv)Each master can grant other masters read/write access to its region

b. Protected Range Registers.

i)3 sets (one for each master) of Lockable Protected Range registers that can restrict program register accesses from the same master.

ii)Can span multiple regions

iii)Separate read and write protection

iv)Special case: BIOS PR4 write protect values are received from Soft Strap and affect all masters.

c. SMI Write Protection for BIOS

i)If enabled, will cause an SMI if a program register access occurs. The primary purpose of this requirement is to support SMI based BIOS update utilities.

d. Illegal Instruction protection for instructions such as Chip Erase

e. Lockable software sequencing opcodes

3)SPI Flash Access

a. Direct Read Access

b. Program Register Access

i)Hardware Sequencing

(1)Software Sequencing uses HW to provide the basic instructions of read, write, and erase.

ii)Software Sequencing

(1)Allows SW to use any legal Opcode

c. Support for Boot BIOS on SPI.

i)Non-boot BIOS that is accessible through program register only can be used on SPI when boot BIOS is located on some other interface.

d. Pre-fetching/Caching to improve performance

i)Separate 64B pre-fetch/cache each for HOST and SEC direct read accesses

4)SFDP Parameter Discoverability[1]

5)Flash Component Capabilities

a. In Descriptor mode, supports two SPI Flash components using two separate chip select pins, CS0# and CS1#. Only one component supported in non-descriptor mode.

i)Components must have the same erasable block/sector size

ii)Each component can be up to 16MB (32MB total addressable) using 24-bit addressing.

b. 1.8V SPI I/O buffer VCC

c. Supports the SPI Fast Read/Write instruction and frequencies of 20MHz, 33MHz and 50 MHz. Supports the SPI Dual Output Fast Read/Write instruction with frequencies of 20 MHz, 33 MHz and 50 MHz

d. Supports the SPI Quad Output Fast Read/Write instruction with frequencies of 20 MHz, 33 MHz and 50 MHz

e. Uses standardized Flash Instruction Set.

f. Supports non-power of 2 flash sizes, with the following restrictions:

i)Only supported in Descriptor Mode.

ii)BIOS accesses in non-descriptor mode to a non-binary flash size will not function properly.

iii)The Flash Regions must be programmed to the actual size of the Flash Component(s).

iv)If using two flash components, the 1st flash component (the one with the Flash Descriptor) must be of binary size. The 2nd flash component can be a non-binary size. If using only one flash component, it can be of non-binary size.

v)The value programmed in the Flash Descriptor Component Density must be set to the next power of 2 value larger than the non-binary size.

8)Reset Capabilities

a. RSMRST#

i)When RSMRST# is asserted, SoC will tri-state with a weak pull-up all SPI pins

i)The SPI Controller will implement a sideband handshake((handshake is reset warn message)) with PMC when a host reset is requested to allow the SPI Flash controller to complete any outstanding atomic sequences and quiescence the SPI Bus

Note: There is no N*parameter headers support on SoC, DTR and 32-bit addressing is not supported.

18.4 PCU - Universal Asynchronous Receiver/ Transmitter (UART)

This section describes the Universal Asynchronous Receiver/Transmitter (UART) serial port integrated into the PCU. The UART may be controlled through programmed IO.

Note: Only a minimal ball-count, comprising receive & transmit signals, UART port is implemented. Further, a maximum baud rate of only 115,200 bps is supported. For this reason, it is recommended that the UART port be used for debug purposes only.

18.4.1 Signal Descriptions

Refer Chapter 2, "Physical Interfaces" for additional details.

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin.
- **Direction**: The buffer direction can be either input, output, or I/O (bidirectional).
- **Type:** The buffer type found in Chapter 19, "Electrical Specifications".
- **Description**: A brief explanation of the signal's function.

Table 103. UART Signals

Signal Name	Direction /Type	Description
UART0_DATAIN	I/GPIOHV, HS	**COM1 Receive:** Serial data input from device pin to the receive port. *This signal is muxed and may be used by other functions.*
UART0_DATAOUT	O/GPIOHV, HS	**COM1 Transmit:** Serial data output from transmit port to the device pin. *This signal is muxed and may be used by other functions.*

18.4.2 Features

The serial port consists of a UART which supports a subset of the functions of the 16550 industry standard.

The UART performs serial-to-parallel conversion on data characters received from a peripheral device and parallel-to-serial conversion on data characters received from the processor. The processor may read the complete status of the UART at any time during the functional operation. Available status information includes the type and condition of the transfer operations being performed by the UART and any error conditions.

The serial port may operate in either FIFO or non-FIFO mode. In FIFO mode, a 16-byte transmit FIFO holds data from the processor to be transmitted on the serial link and a 16-byte Receive FIFO buffers data from the serial link until read by the processor.

The UART includes a programmable baud rate generator which is capable of generating a baud rate of between 50 bps and 115,200 bps from a fixed baud clock input of 1.8432 MHz. The baud rate is calculated as follows:

Baud Rate Calculation:

$$\mathrm{BaudRate} = \frac{1.8432 \times 10^6}{16 \times \mathrm{Divisor}}$$

The divisor is defined by the Divisor Latch LSB and Divisor Latch MSB registers. Some common values are shown in Table 104.

Table 104. Baud Rate Examples

Desired Baud Rate	Divisor	Divisor Latch LSB Register	Divisor Latch MSB Register
115,200	1	1h	0h
57,600	2	2h	0h
38,400	3	3h	0h
19,200	6	6h	0h
9,600	12	Ch	0h
4,800	24	18h	0h
2,400	48	30h	0h
1,200	96	60h	0h
300	384	80h	1h
50	2,304	0h	9h

The UART has interrupt support and those interrupts may be programmed to the user's requirements, minimizing the computing required to handle the communications link. Each UART may operate in a polled or an interrupt driven environment as configured by software.

18.4.2.1 FIFO Operation

18.4.2.1.1 FIFO Interrupt Mode Operation

Receiver Interrupt

When the Receive FIFO and receiver interrupts are enabled (FIFO Control Register, bit 0 = 1b and Interrupt Enable Register (IIR), bit 0 = 1b), receiver interrupts occur as follows:

- The receive data available interrupt is invoked when the FIFO has reached its programmed trigger level. The interrupt is cleared when the FIFO drops below the programmed trigger level.

- The IIR receive data available indication also occurs when the FIFO trigger level is reached, and like the interrupt, the bits are cleared when the FIFO drops below the trigger level.
- The receiver line status interrupt (IIR = C6h), as before, has the highest priority. The receiver data available interrupt (IIR = C4h) is lower. The line status interrupt occurs only when the character at the top of the FIFO has errors.
- The COM1_LSR.DR bit is set to 1b as soon as a character is transferred from the shift register to the Receive FIFO. This bit is reset to 0b when the FIFO is empty.

Character Time Out Interrupt

When the receiver FIFO and receiver time out interrupt are enabled, a character time out interrupt occurs when all of the following conditions exist:

- At least one character is in the FIFO.
- The last received character was longer than four continuous character times ago (if 2 stop bits are programmed the second one is included in this time delay).
- The most recent processor read of the FIFO was longer than four continuous character times ago.
- The receiver FIFO trigger level is greater than one.

The maximum time between a received character and a timeout interrupt is 160 ms at 300 baud with a 12-bit receive character (i.e., 1 start, 8 data, 1 parity, and 2 stop bits).

When a time out interrupt occurs, it is cleared and the timer is reset when the processor reads one character from the receiver FIFO. If a time out interrupt has not occurred, the time out timer is reset after a new character is received or after the processor reads the receiver FIFO.

Transmit Interrupt

When the transmitter FIFO and transmitter interrupt are enabled (FIFO Control Register, bit 0 = 1b and Interrupt Enable Register, bit 0 = 1b), transmit interrupts occur as follows:

The Transmit Data Request interrupt occurs when the transmit FIFO is half empty or more than half empty. The interrupt is cleared as soon as the Transmit Holding Register is written (1 to 16 characters may be written to the transmit FIFO while servicing the interrupt) or the Interrupt Identification Register is read.

18.4.2.1.2 FIFO Polled Mode Operation

With the FIFOs enabled (FIFO Control register, bit 0 = 1b), setting Interrupt Enable register (IER), bits 3:0 = 000b puts the serial port in the FIFO polled mode of operation. Since the receiver and the transmitter are controlled separately, either one or both may be in the polled mode of operation. In this mode, software checks receiver and transmitter status through the Line Status Register (LSR). As stated in the register description:

- LSR[0] is set as long as there is one byte in the receiver FIFO.
- LSR[1] through LSR[4] specify which error(s) has occurred for the character at the top of the FIFO. Character error status is handled the same way as interrupt mode. The Interrupt Identification Register is not affected since IER[2] = 0b.
- LSR[5] indicates when the transmitter FIFO needs data.
- LSR[6] indicates that both the transmitter FIFO and shift register are empty.
- LSR[7] indicates whether there are any errors in the receiver FIFO.

18.4.3 Use

18.4.3.1 Base I/O Address

COM1

The base I/O address for the COM1 UART is fixed to 3F8h.

18.4.3.2 Legacy Interrupt

COM1

The legacy interrupt assigned to the COM1 UART is fixed to IRQ4.

18.4.4 UART Enable/Disable

The COM1 UART may be enabled or disabled using the UART_CONT.COM1EN register bit. By default, the UART is disabled.

Note: It is recommended that the UART be disabled during normal platform operation. An enabled UART can interfere with platform power management.

18.4.5 IO Mapped Registers

There are 12 registers associated with the UART. These registers share eight address locations in the IO address space. Table 105 shows the registers and their addresses as offsets of a base address. Note that the state of the COM1_LCR.DLAB register bit, which is the most significant bit (MSB) of the Serial Line Control register, affects the selection of certain of the UART registers. The COM1_LCR.DLAB register bit must be set high by the system software to access the Baud Rate Generator Divisor Latches.

18.5 Register Map

Table 105. Register Access List

Register Address (Offset to Base IO Address)	COM1_LCR.DLAB Value	Register Access Type	Register Accessed
0h	0b	RO	Receiver Buffer[1]
0h	0b	WO	Transmitter Holding[1]
0h	1b	RW	Divisor Latch LSB (Lowest Significant Bit)[1]
1h	0b	RW	Interrupt Enable[2]
1h	1b	RW	Divisor Latch MSB (Most Significant Bit)[2]
2h	xb	RO	Interrupt Identification[3]
2h	xb	WO	FIFO Control[3]
3h	xb	RW	Line Control
4h	xb	RW	Modem Control[4]
5h	xb	RO	Line Status
6h	xb	RO	Modem Status[4]
7h	xb	RW	Scratchpad

NOTES:

1. These registers are consolidated in the Receiver Buffer / Transmitter Holding Register (COM1_RX_TX_BUFFER).
2. These registers are consolidated in the Interrupt Enable Register (COM1_IER)
3. These registers are consolidated in the Interrupt Identification / FIFO Control Register (COM1_IIR).
4. These registers are implemented but unused since the UART signals related to modem interaction are not implemented.

18.6 PCU - Intel Legacy Block (iLB) Overview

The Intel Legacy Block (iLB) is a collection of disparate functional blocks that are critical for implementing the legacy PC platform features. These blocks include:

- "PCU - iLB - Low Pin Count (LPC) Bridge"
- "PCU - iLB - Real Time Clock (RTC)"
- "PCU - iLB - 8254 Timers"
- "PCU - iLB - High Precision Event Timer (HPET)"
- "PCU - iLB - GPIO"
- "PCU - iLB - IO APIC"
- "PCU - iLB - 8259 Programmable Interrupt Controllers (PIC)"

The iLB also implements a register range for configuration of some of those blocks along with support for Non-Maskable Interrupts (NMI).

18.6.1 Signal Descriptions

Refer Chapter 2, "Physical Interfaces" for additional details as well as the subsequent sections.

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin.
- **Direction**: The buffer direction can be either input, output, or I/O (bidirectional).
- **Type:** The buffer type found in Chapter 19, "Electrical Specifications".
- **Description**: A brief explanation of the signal's function.

Table 106. iLB Signals

Signal Name	Direction /Type	Description
NMI_N	I/GPIOMV, MS	Non-Maskable Interrupt: This is an NMI event indication into the SoC. *This signal is muxed and may be used by other functions.*

18.6.2 Features

18.6.2.1 Key Features

The key features of various blocks are as follows:

- LPC Interface
  - — Supports Low Pin Count (LPC) 1.1 Specification
  - — No support for DMA or bus mastering
  - — Supports Trusted Platform Module (TPM) 1.2

- General Purpose Input Output
  - Legacy control interface for SoC GPIOs
  - I/O mapped registers
- 8259 Programmable Interrupt Controller
  - Supports Legacy interrupt
  - 15 total interrupts through two cascaded controllers
  - I/O mapped registers
- I/O Advanced Programmable Interrupt Controller
  - Supports Legacy-free interrupt
  - 115 total interrupts
  - Memory mapped registers
- 8254
  - Legacy timer support
  - Three timers with fixed uses: System Timer, Refresh Request Signal and Speaker Tone
  - I/O mapped registers
- HPET - High Performance Event Timers
  - Supports Legacy-free timer
  - Three timers and one counter
  - Memory mapped registers
- Real-Time Clock (RTC)
  - 242 byte RAM backed by battery (aka CMOS RAM)
  - Can generate wake/interrupt when time matches programmed value
  - I/O and indexed registers

18.6.2.2 Non-Maskable Interrupt

NMI support is enabled by setting the NMI Enable (NMI_EN) bit, at IO Port 70h, Bit 7, to 1b.

Non-Maskable Interrupts (NMIs) can be generated by several sources, as described in Table 107.

Table 107. NMI Sources

NMI Source	NMI Source Enabler/ Disabler	NMI Source Status	Alternate Configuration
SERR# goes active **NOTE:** A SERR# is only generated internally in the SoC)	NSC.SNE	NSC.SNS	All NMI sources may, alternatively, generate a SMI by setting GNMI.NMI2SMIEN=1b The SoC uses GNMI.NMI2SMIST for observing SMI status
IOCHK# goes active **NOTE:** A IOCHK# is only generated as a SERIRQ# frame	NSC.INE	NSC.INS	
NMI goes active **NOTE:** Active can be defined as being on the positive or negative edge of the signal using the GNMI.GNMIED register bit.	GNMI.GNMIED	GNMI.GNMIS	
Software sets the GNMI.NMIN register bit	GNMI.NMIN	GNMI.NMINS	

18.6.2.3 S0ix Support

There is no requirement to set “HPET_GCFG.EN” to 0b. Basically turn off HPET during S0i2/3. RTD3hot status is not a key requirement for OS anymore.

The S1 state described in the HPET spec is a "CPU Stop Grant" condition. This condition is met during the S0i2/3 states, (although entry into S0i2/3 is performed in a different way).

18.6.3 Use

18.6.3.1 S0ix Support

Prior to entry into S0i2 or S0i3 state, the driver/OS must set HPET_GCFG.EN to 0b to indicate RTD3hot status.

18.7 PCU - iLB - Low Pin Count (LPC) Bridge

The SoC implements an LPC Interface as described in the LPC 1.1 Specification. The Low Pin Count (LPC) bridge function of the SoC resides in PCI Device 31, Function 0.

Note: In addition to the LPC bridge interface function, D31:F0 contains other functional units including interrupt controllers, timers, power management, system management, GPIO, and RTC.

18.7.1 Signal Descriptions

Refer Chapter 2, “Physical Interfaces” for additional details.

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin.
- **Direction**: The buffer direction can be either input, output, or I/O (bidirectional).
- **Type:** The buffer type found in Chapter 19, “Electrical Specifications”.
- **Description**: A brief explanation of the signal’s function.

Table 108. LPC Signals

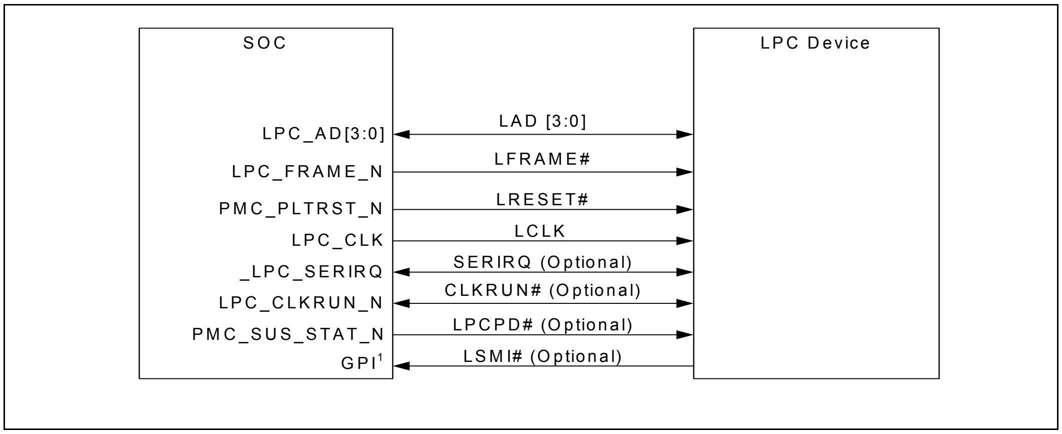
Signal Name	Direction/ Type	Description
LPC_AD[3:0]	I/O/ GPIOHV, HS	**LPC Multiplexed Command, Address, Data:** Internal pull-ups are provided for these signals. *These signals are muxed and may be used by other functions.*
LPC_CLKOUT[0]	I/O/ GPIOHV, HS	**LPC Clock [0] Out:** 19MHz PCI-like clock driven to LPC peripherals. *These signals are muxed and may be used by other functions.*
LPC_CLKOUT[1]	I/O/ GPIOHV, HS	**LPC Clock [1] Out:** 19MHz PCI-like clock driven to LPC peripherals. Can be configured as an input to compensate for board routing delays through Soft Strap. *These signals are muxed and may be used by other functions.*
LPC_CLKRUN_N	I/O/ GPIOHV, HS	**LPC Clock Run:** Input to determine the status of LPC_CLK and an open drain output used to request starting or speeding up LPC_CLK. This is a sustained tri-state signal used by the central resource to request permission to stop or slow LPC_CLK. The central resource is responsible for maintaining the signal in the asserted state when LPC_CLK is running and deasserts the signal to request permission to stop or slow LPC_CLK. An internal pull-up is provided for this signal. *This signal is muxed and may be used by other functions.*
LPC_FRAME_N	I/O/ GPIOHV, HS	**LPC Frame:** This signal indicates the start of an LPC cycle, or an abort. *This signal is muxed and may be used by other functions.*
LPC_SERIRQ	I/O/ GPIOHV, HS	**Serial Interrupt Request:** This signal implements the serial interrupt protocol. *This signal is muxed and may be used by other functions.* **NOTE:** A level shifter needs to be implemented on this signal.

18.7.2 Features

The LPC interface to the SoC is shown in Figure 32. Note that the SoC implements all of the signals that are shown as optional, but peripherals are not required to do so.

Note: The LPC controller does not implement bus mastering cycles or DMA.

Figure 32. LPC Interface Diagram



NOTE: The General Purpose Input (GPI) must use a SMI capable GPIO: GPIO_S0_SC[7:0].

18.7.2.1 Memory Cycle Notes

For cycles below 16M, the LPC Controller will perform standard LPC memory cycles. For cycles targeting firmware (BIOS/EFI code only), firmware memory cycles are used. Only 8-bit transfers are performed. If a larger transfer appears, the LPC controller will break it into multiple 8-bit transfers until the request is satisfied.

If the cycle is not claimed by any peripheral (and subsequently aborted), the LPC Controller will return a value of all 1's to the CPU.

18.7.2.2 Trusted Platform Module (TPM) 1.2 Support

The LPC interface supports accessing Trusted Platform Module (TPM) 1.2 devices via the LPC TPM START encoding. Memory addresses within the range FED00000h to FED40FFFh will be accepted by the LPC Bridge and sent on LPC as TPM special cycles. No additional checking of the memory cycle is performed.

Note: This is different to the FED00000h to FED4BFFFh range implemented on some other Intel components since no Intel® Trusted Execution Technology (Intel® TXT) transactions are supported.

18.7.2.3 FWH Cycle Notes

If the LPC controller receives any SYNC returned from the device other than short (0101), long wait (0110), or ready (0000) when running a FWH cycle, indeterminate results may occur. A FWH device is not allowed to assert an Error SYNC.

BIOS/EFI boot from LPC is not supported when Secure Boot is enabled.

18.7.2.4 Subtractive Decode

All cycles that are not decoded internally, and are not targeted for LPC (i.e., configuration cycles, IO cycles above 64KB and memory cycles above 16MB), will be sent to LPC with LPC_FRAME_N not asserted.

18.7.2.5 POST Code Redirection

Writes to addresses 80h - 8Fh in IO register space will also be passed to the LPC bus.

Note: Reads of these addresses do not result in any LPC transactions.

18.7.2.6 Power Management

18.7.2.6.1 LPCPD_N Protocol

Same timings as for PMC_SUS_STAT_N. After driving PMC_SUS_STAT_N active, the SoC drives LPC_FRAME_N low, and tri-states (or drives low) LPC_AD[3:0].

Note: The Low Pin Count Interface Specification, Revision 1.1 defines the LPCPD_N protocol where there is at least 30 µs from LPCPD_N assertion to LRST_N assertion. This specification explicitly states that this protocol only applies to entry/exit of low power states which does not include asynchronous reset events. The SoC asserts both PMC_SUS_STAT_N (connects to LPCPD_N) and PLTRST_N (connects to LRST_N) at the same time during a global reset. This is not inconsistent with the LPC LPCPD_N protocol.

18.7.2.6.2 Clock Run (CLKRUN)

When there are no pending LPC cycles, and SERIRQ is in quiet mode, the SoC can shut down the LPC clock. The SoC indicates that the LPC clock is going to shut down by de-asserting the LPC_CLKRUN_N signal. LPC devices that require the clock to stay running should drive LPC_CLKRUN_N_N low within 4 clocks of its de-assertion. If no device drives the signal low within 4 clocks, the LPC clock will stop. If a device asserts LPC_CLKRUN_N, the SoC will start the LPC clock and assert LPC_CLKRUN_N.

Note: The CLKRUN protocol is disabled by default. Refer Section 18.7.3.2.2, "Clock Run Enable" for further details.

18.7.2.7 Serialized IRQ (SERIRQ)

The interrupt controller supports a serial IRQ scheme. The signal used to transmit this information is shared between the interrupt controller and all peripherals that support serial interrupts. The signal line, LPC_SERIRQ, is synchronous to LPC clock, and follows the sustained tri-state protocol that is used by LPC signals. The serial IRQ protocol defines this sustained tri-state signaling in the following fashion:

- **S - Sample Phase:** Signal driven low.

- **R - Recovery Phase:** Signal driven high.
- **T - Turn-around Phase:** Signal released.

The interrupt controller supports 21 serial interrupts. These represent the 15 ISA interrupts (IRQ0- 1, 3-15), the four PCI interrupts, and the control signals SMI_N and IOCHK_N. Serial interrupt information is transferred using three types of frames:

- Start Frame: LPC_SERIRQ line driven low by the interrupt controller to indicate the start of IRQ transmission.
- Data Frames: IRQ information transmitted by peripherals. The interrupt controller supports 21 data frames.
- Stop Frame: LPC_SERIRQ line driven low by the interrupt controller to indicate end of transmission and next mode of operation.

18.7.2.7.1 Start Frame

The serial IRQ protocol has two modes of operation which affect the start frame:

- Continuous Mode: The interrupt controller is solely responsible for generating the start frame.
- Quiet Mode: Peripheral initiates the start frame, and the interrupt controller completes it.

These modes are entered via the length of the stop frame.

Continuous mode must be entered first, to start the first frame. This start frame width is 8 LPC clocks. This is a polling mode.

In Quiet mode, the LPC_SERIRQ line remains inactive and pulled up between the Stop and Start Frame until a peripheral drives LPC_SERIRQ low. The interrupt controller senses the line low and drives it low for the remainder of the Start Frame. Since the first LPC clock of the start frame was driven by the peripheral, the interrupt controller drives LPC_SERIRQ low for 1 LPC clock less than in continuous mode. This mode of operation allows for lower power operation.

18.7.2.7.2 Data Frames

Once the Start frame has been initiated, the LPC_SERIRQ peripherals start counting frames based on the rising edge of LPC_SERIRQ. Each of the IRQ/DATA frames has exactly 3 phases of 1 clock each:

- **Sample Phase:** During this phase, a device drives LPC_SERIRQ low if its corresponding interrupt signal is low. If its corresponding interrupt is high, then the LPC_SERIRQ devices tri-state LPC_SERIRQ. LPC_SERIRQ remains high due to pull-up resistors.
- **Recovery Phase:** During this phase, a device drives LPC_SERIRQ high if it was driven low during the Sample Phase. If it was not driven during the sample phase, it remains tri-stated in this phase.
- **Turn-around Phase:** The device tri-states LPC_SERIRQ.

18.7.2.7.3 Stop Frame

After the data frames, a Stop Frame will be driven by the interrupt controller. LPC_SERIRQ will be driven low for two or three LPC clocks. The number of clocks is determined by the SCNT.MD register bit. The number of clocks determines the next mode, as indicated in Table 109.

Table 109. SERIRQ, Stop Frame Width to Operation Mode Mapping

Stop Frame Width	Next Mode
Two LPC clocks	**Quiet Mode:** Any SERIRQ device initiates a Start Frame
Three LPC clocks	**Continuous Mode:** Only the interrupt controller initiates a Start Frame

18.7.2.7.4 Serial Interrupts Not Supported

There are four interrupts on the serial stream which are not supported by the interrupt controller. These interrupts are:

- IRQ0: Heartbeat interrupt generated off of the internal 8254 counter 0.
- IRQ8: RTC interrupt can only be generated internally.
- IRQ13: This interrupt (floating point error) is not supported.

The interrupt controller will ignore the state of these interrupts in the stream.

18.7.2.7.5 Data Frame Format and Issues

Table below shows the format of the data frames. The decoded INT[A:D]_N values are ANDed with the corresponding PCI-express input signals (PIRQ[A:D]_N). This way, the interrupt can be shared.

The other interrupts decoded via SERIRQ are also ANDed with the corresponding internal interrupts. For example, if IRQ10 is set to be used as the SCI, then it is ANDed with the decoded value for IRQ10 from the SERIRQ stream.

Table 110. SERIRQ Interrupt Mapping

Data Frame #	Interrupt	Clocks Past Start Frame	Comment
1	IRQ0	2	Ignored. Can only be generated via the internal 8524
2	IRQ1	5	Before port 60h latch
3	SMI_N	8	Causes SMI_N if low. Sets SMI_STS.ILB_SMI_STS register bit.
4	IRQ3	11	
5	IRQ4	14	
6	IRQ5	17	

Table 110. SERIRQ Interrupt Mapping

Data Frame #	Interrupt	Clocks Past Start Frame	Comment
7	IRQ6	20	
8	IRQ7	23	
9	IRQ8	26	Ignored. IRQ8_N can only be generated internally
10	IRQ9	29	
11	IRQ10	32	
12	IRQ11	35	
13	IRQ12	38	Before port 60h latch
14	IRQ13	41	Ignored.
15	IRQ14	44	Ignored
16	IRQ15	47	
17	IOCHCK_N	50	Same as ISA IOCHCK_N going active.
18	PCI INTA_N	53	
19	PCI INTB_N	56	
20	PCI INTC_N	59	
21	PCI INTD_N	62	

18.7.2.7.6 S0ix Support

During S0i2 and S0i3, the LPC and SERIRQ interfaces are disabled.

18.7.3 Usage

18.7.3.1 LPC Clock Delay Compensation

In order to meet LPC interface AC timing requirements, a LPC clock loop back is required. The operation of this loop back can be configured in two ways:

1. On the SOC: In this configuration, LPC_CLK[0] is looped back on itself on the SOC pad.

   a. Benefit:

   LPC_CLK[0] and LPC_CLK[1] are both available for system clocking

   b. Drawback:

   Clock delay compensation is less effective at compensating for mainboard delay

   c. Soft Strap & Register Requirements:

   Soft Strap LPCCLK_SLC = 0b

Configuration is reflected by register bit LPCC.LPCCLK_SLC=0b

Soft Strap LPCCLK1_ENB = 0b (LPC_CLK[1] disabled) or 1b (LPC_CLK[1] enabled)

2. Configuration is reflected by register bit LPCC.LPCCLK1EN=0b (LPC_CLK[1] disabled) or 1b (LPC_CLK[1] enabled)
3. On the main board: In this configuration, LPC_CLK[0] is looped back to LPC_CLK[1] on the main board.

   a. Benefit:

   Clock delay compensating in more effective at compensating for main board delay

   b. Drawback:

   Only LPC_CLK[0] is available for system clocking. LPC_CLK[1] must be disabled.

   c. Soft Strap & Register Requirements:

   Soft Strap LPCCLK_SLC = 1b

   Configuration is reflected by register bit LPCC.LPCCLK_SLC=1b

   Soft Strap LPCCLK1_ENB = 0b (LPC_CLK[1] disabled)

   Configuration is reflected by register bit LPCC.LPCCLK1EN=0b

18.7.3.2 LPC Power Management

18.7.3.2.1 Clock Enabling

The LPC clocks can be enabled or disabled by setting or clearing, respectively, the LPCC.LPCCLK[1:0]EN bits.

18.7.3.2.2 Clock Run Enable

The Clock Run protocol is disabled by default and should only be enabled during operating system run-time, once all LPC devices have been initialized. The Clock Run protocol is enabled by setting the LPCC.CLKRUN_EN register bit.

18.7.3.3 SERIRQ Disable

Serialized IRQ support may be disabled by setting the OIC.SIRQEN bit to 0b.

18.7.4 References

- Low Pin Count Interface Specification, Revision 1.1 (LPC): http://www.intel.com/design/chipsets/industry/lpc.htm.
- Serialized IRQ Support for PCI Systems, Revision 6.0: http://www.smsc.com/media/Downloads_Public/papers/serirq60.doc.
- Implementing Industry Standard Architecture (ISA) with Intel® Express Chipsets (318244): http://www.intel.com/assets/pdf/whitepaper/318244.pdf.

18.8 PCU - iLB - Real Time Clock (RTC)

The SoC contains a real-time clock with 242 bytes of battery-backed RAM. The real-time clock performs two key functions—keeping track of the time of day and storing system data, even when the system is powered down. The RTC operates on a 32.768 kHz crystal and a 3.3 V battery.

The RTC supports two lockable memory ranges. By setting bits in the configuration space, two 8-byte ranges can be locked to read and write accesses. This prevents unauthorized reading of passwords or other system security information.

The RTC supports a date alarm that allows for scheduling a wake up event up to 30 days in advance.

18.8.1 Signal Descriptions

Refer Chapter 2, “Physical Interfaces” for additional details.

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin.
- **Direction**: The buffer direction can be either input, output, or I/O (bidirectional).
- **Type:** The buffer type found in Chapter 19, “Electrical Specifications”.
- **Description**: A brief explanation of the signal’s function.

Table 111. RTC Signals

Signal Name	Direction /Type	Description
RTC_X1	I Analog	**Crystal Input 1:** This signal is connected to the 32.768 kHz crystal. If no external crystal is used, the signal can be driven with the desired clock rate.
RTC_X2	I Analog	**Crystal Input 2:** This signal is connected to the 32.768 kHz crystal. If no external crystal is used, the signal should be left floating.
RTC_RST_N	I	**RTC Reset:** When asserted, this signal resets register bits in the RTC well. **NOTE:** Unless CMOS is being cleared (only to be done in the G3 power state), the signal input must always be high when all other RTC power planes are on. **NOTE:** In the case where the RTC battery is dead or missing on the platform, the signal should be deasserted before the PMC_RSMRST_N signal is deasserted.
RTC_TEST_N	I	**RTC Battery Test:** An external RC circuit creates a time delay for the signal such that it will go high (to ILB_RTC_3P3_G3) sometime after the battery voltage is valid. The RC time delay should be in the 10-20 ms range. This signal will be asserted just after suspend power is up if the coin cell battery is weak. **NOTE:** This signal may also be used for debug purposes, as part of a XDP port.

18.8.2 Features

The Real Time Clock (RTC) module provides a battery backed-up date and time keeping device. Three interrupt features are available: time of day alarm with once a second to once a month range, periodic rates of 122 ms to 500 ms, and end of update cycle notification. Seconds, minutes, hours, days, day of week, month, and year are counted. The hour is represented in twelve or twenty-four hour format, and data can be represented in BCD or binary format. The design is meant to be functionally compatible with the Motorola* MS146818B. The time keeping comes from a 32.768 kHz oscillating source, which is divided to achieve an update every second. The lower 14 bytes on the lower RAM block have very specific functions. The first ten are for time and date information. The next four (0Ah to 0Dh) are registers, which configure and report RTC functions. A host-initiated write takes precedence over a hardware update in the event of a collision.

18.8.2.1 Update Cycles

An update cycle occurs once a second, if the B.SET bit is not asserted and the divide chain is properly configured. During this procedure, the stored time and date are incremented, overflow checked, a matching alarm condition is checked, and the time and date are rewritten to the RAM locations. The update cycle starts at least 488 ms after A.UIP is asserted, and the entire cycle does not take more than 1984 ms to complete. The time and date RAM locations (00h to 09h) are disconnected from the external bus during this time.

18.8.3 Interrupts

The real-time clock interrupt is internally routed within the SoC both to the I/O APIC and the 8259. It is mapped to interrupt vector 8. This interrupt does not leave the SoC, nor is it shared with any other interrupt. IRQ8# from the ILB_LPC_SERIRQ stream is ignored. However, the High Performance Event Timers can also be mapped to IRQ8#; in this case, the RTC interrupt is blocked.

18.8.3.1 Lockable RAM Ranges

The RTC battery-backed RAM supports two 8-byte ranges that can be locked: the RC.UL and RC.LL register bits. When the locking bits are set, the corresponding range in the RAM is not readable or writable. A write cycle to those locations will have no effect. A read cycle to those locations will not return the location's actual value (resultant value is undefined).

Once a range is locked, the range can be unlocked only by a hard reset, which will invoke the BIOS and allow it to re-lock the RAM range.

18.8.3.2 Clearing Battery-Backed RTC RAM

Clearing CMOS RAM in an SoC-based platform can be done by using a jumper on RTC_RST_N or a GPI. Implementations should not attempt to clear CMOS by using a jumper to pull RTC_VCC low.

18.8.3.2.1 Using RTC_RST_N to Clear CMOS

A jumper on RTC_RST_N can be used to clear CMOS values, as well as reset to default, the state of those configuration bits that reside in the RTC power well. When the RTC_RST_N is strapped to ground, the GEN_PMCON1.RPS register bit will be set and those configuration bits in the RTC power well will be set to their default state. BIOS can monitor the state of this bit, and manually clear the RTC CMOS array once the system is booted. The normal position would cause RTC_RST_N to be pulled up through a weak pull-up resistor. Table 112 shows which bits are set to their default state when RTC_RST_N is asserted. This RTC_RST_N jumper technique allows the jumper to be moved and then replaced—all while the system is powered off. Then, once booted, the GEN_PMCON1.RPS bit can be detected in the set state.

Table 112. Register Bits Reset by RTC_RST_N Assertion

Register Bit	Bit(s)	Default State
RCRB_GENERAL_CONTROL.TS	1	xb
GEN_PMCON1.PME_B0_S5_DIS	15	0b
GEN_PMCON1.WOL_EN_OVRD	13	0b
GEN_PMCON1.DIS_SLP_X_STRCH_SUS_UP	12	0b
GEN_PMCON1.RTC Reserved	8	0b
GEN_PMCON1.SWSMI_RATESEL	7:6	00b
GEN_PMCON1.S4MAW	5:4	00b
GEN_PMCON1.S4ASE	3	0b
GEN_PMCON1.RPS	2	1b
GEN_PMCON1.AG3E	0	0b
PM1_STS_EN.RTC_EN	26	0b
PM1_STS_EN.PWRBTNOR_STS	11	0b
PM1_CNT.SLP_TYP	12:10	0b
GPE0a_EN.PME_B0_EN	13	0b
GPE0a_EN.BATLOW_EN	10	0b

18.8.3.3 Using GPI to Clear CMOS

A jumper on a GPI can also be used to clear CMOS values. BIOS should detect the setting of this GPI on system boot-up, and manually clear the CMOS array.

Note: The GPI strap technique to clear CMOS requires multiple steps to implement. The system is booted with the jumper in new position, then powered back down. The jumper is replaced back to the normal position, then the system is rebooted again.

Warning: Do not implement a jumper on RTC_VCC to clear CMOS.

18.8.3.4 S0ix Support

During S0i3, the RTC interface is active.

18.8.4 References

Accessing the Real Time Clock Registers and the NMI Enable Bit: http://download.intel.com/design/intarch/PAPERS/321088.pdf.

18.8.5 IO Mapped Registers

The RTC internal registers and RAM is organized as two banks of 128 bytes each, called the standard and extended banks.

Note: It is not possible to disable the extended bank.

The first 14 bytes of the standard bank contain the RTC time and date information along with four registers, A - D, that are used for configuration of the RTC. The extended bank contains a full 128 bytes of battery backed SRAM. All data movement between the host CPU and the RTC is done through registers mapped to the standard I/O space.

Note: Registers reg_RTC_IR_type and reg_RTC_TR_type are used for data movement to and from the standard bank. Registers reg_RTC_RIR_type and reg_RTC_RTR_type are used for data movement to and from the extended bank. All of these registers have alias I/O locations, as indicated in Table 113.

Table 113. I/O Registers Alias Locations

Register	Original I/O Location	Alias I/O Location
reg_RTC_IR_type	70h	74h
reg_RTC_TR_type	71h	75h
reg_RTC_RIR_type	72h	76h
reg_RTC_RTR_type	73h	77h

18.8.6 Indexed Registers

The RTC contains indexed registers that are accessed via the reg_RTC_IR_type and reg_RTC_TR_type registers.

Table 114. RTC Indexed Registers (Sheet 1 of 2)

Start	End	Name
00h	00h	Seconds
01h	01h	Seconds Alarm
02h	02h	Minutes
03h	03h	Minutes Alarm
04h	04h	Hours
05h	05h	Hours Alarm
06h	06h	Day of Week

Table 114. RTC Indexed Registers (Sheet 2 of 2)

Start	End	Name
07h	07h	Day of Month
08h	08h	Month
09h	09h	Year
0Ah	0Ah	Register A
0Bh	0Bh	Register B
0Ch	0Ch	Register C
0Dh	0Dh	Register D
0Eh	7Fh	114 Bytes of User RAM

18.9 PCU - iLB - 8254 Timers

The 8254 contains three counters which have fixed uses including system timer and speaker tone. All registers are clocked by a 14.31818 MHz clock.

18.9.1 Signal Descriptions

Refer Chapter 2, “Physical Interfaces” for additional details.

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin.
- **Direction**: The buffer direction can be either input, output, or I/O (bidirectional).
- **Type:** The buffer type found in Chapter 19, “Electrical Specifications”.
- **Description**: A brief explanation of the signal’s function.

18.9.2 Features

18.9.2.1 Counter 0, System Timer

This counter functions as the system timer by controlling the state of IRQ0 and is programmed for Mode 3 operation. The counter produces a square wave with a period equal to the product of the counter period (838 ns) and the initial count value. The counter loads the initial count value one counter period after software writes the count value to the counter I/O address. The counter initially asserts IRQ0 and decrements the count value by two each counter period. The counter negates IRQ0 when the count value reaches 0. It then reloads the initial count value and again decrements the initial count value by two each counter period. The counter then asserts IRQ0 when the count value reaches 0, reloads the initial count value, and repeats the cycle, alternately asserting and negating IRQ0.

18.9.2.2 Counter 1, Refresh Request Signal

This counter is programmed for Mode 2 operation and impacts the period of the NSC.RTS register bit. Programming the counter to anything other than Mode 2 results in undefined behavior.

18.9.2.3 Counter 2, Speaker Tone

This counter provides the speaker tone and is typically programmed for Mode 3 operation. The counter provides a speaker frequency equal to the counter clock frequency (1.193 MHz) divided by the initial count value. The speaker must be enabled by a write to the NSC.SDE register bit.

18.9.2.4 S0ix Support

During S0i2 and S0i3, the 8254 timer is halted. A platform that requires the 8254 timer to be always active, should disable S0i2/3 using the S0ix_Enable register.

18.9.3 Usage

18.9.3.1 Timer Programming

The counter/timers are programmed in the following fashion:

1. Write a control word to select a counter.
2. Write an initial count for that counter.
3. Load the least and/or most significant bytes (as required by Control Word Bits 5, 4) of the 16-bit counter.
4. Repeat with other counters.

Only two conventions need to be observed when programming the counters. First, for each counter, the control word must be written before the initial count is written. Second, the initial count must follow the count format specified in the control word (least significant byte only, most significant byte only, or least significant byte and then most significant byte).

A new initial count may be written to a counter at any time without affecting the counter's programmed mode. Counting is affected as described in the mode definitions. The new count must follow the programmed count format.

If a counter is programmed to read/write two-byte counts, the following precaution applies: A program must not transfer control between writing the first and second byte to another routine which also writes into that same counter. Otherwise, the counter will be loaded with an incorrect count.

The Control Word Register at port 43h controls the operation of all three counters. Several commands are available:

- **Control Word Command.** Specifies which counter to read or write, the operating mode, and the count format (binary or BCD).
- **Counter Latch Command.** Latches the current count so that it can be read by the system. The countdown process continues.
- **Read Back Command.** Reads the count value, programmed mode, the current state of the OUT pins, and the state of the Null Count Flag of the selected counter.

Table 115 lists the six operating modes for the interval counters.

Table 115. Counter Operating Modes

Mode	Function	Description
0	Out signal on end of count (=0)	Output is 0. When count goes to 0, output goes to 1 and stays at 1 until counter is reprogrammed.
1	Hardware re-triggerable one-shot	Output is 0. When count goes to 0, output goes to 1 for one clock time.
2	Rate generator (divide by n counter)	Output is 1. Output goes to 0 for one clock time, then back to 1 and counter is reloaded.

Table 115. Counter Operating Modes

Mode	Function	Description
3	Square wave output	Output is 1. Output goes to 0 when counter rolls over, and counter is reloaded. Output goes to 1 when counter rolls over, and counter is reloaded, etc.
4	Software triggered strobe	Output is 1. Output goes to 0 when count expires for one clock time.
5	Hardware triggered strobe	Output is 1. Output goes to 0 when count expires for one clock time.

18.9.3.2 Reading from Interval Timer

It is often desirable to read the value of a counter without disturbing the count in progress. There are three methods for reading the counters: a simple read operation, counter Latch command, and the Read-Back command. Each is explained below.

With the simple read and counter latch command methods, the count must be read according to the programmed format; specifically, if the counter is programmed for two byte counts, two bytes must be read. The two bytes do not have to be read one right after the other. Read, write, or programming operations for other counters may be inserted between them.

18.9.3.2.1 Simple Read

The first method is to perform a simple read operation. The counter is selected through Port 40h (Counter 0), 41h (Counter 1), or 42h (Counter 2).

Note: Performing a direct read from the counter does not return a determinate value, because the counting process is asynchronous to read operations. However, in the case of Counter 2, the count can be stopped by writing 0b to the NSC.TC2E register bit.

18.9.3.2.2 Counter Latch Command

The Counter Latch command, written to Port 43h, latches the count of a specific counter at the time the command is received. This command is used to ensure that the count read from the counter is accurate, particularly when reading a two-byte count. The count value is then read from each counter’s Count register as was programmed by the Control register.

The count is held in the latch until it is read or the counter is reprogrammed. The count is then unlatched. This allows reading the contents of the counters on the fly without affecting counting in progress. Multiple Counter Latch Commands may be used to latch more than one counter. Counter Latch commands do not affect the programmed mode of the counter in any way.

If a counter is latched and then, some time later, latched again before the count is read, the second Counter Latch command is ignored. The count read is the count at the time the first Counter Latch command was issued.

18.9.3.2.3 Read Back Command

The Read Back command, written to Port 43h, latches the count value, programmed mode, and current states of the OUT pin and Null Count flag of the selected counter or counters. The value of the counter and its status may then be read by I/O access to the counter address.

The Read Back command may be used to latch multiple counter outputs at one time. This single command is functionally equivalent to several counter latch commands, one for each counter latched. Each counter's latched count is held until it is read or reprogrammed. Once read, a counter is unlatched. The other counters remain latched until they are read. If multiple count Read Back commands are issued to the same counter without reading the count, all but the first are ignored.

The Read Back command may additionally be used to latch status information of selected counters. The status of a counter is accessed by a read from that counter's I/O port address. If multiple counter status latch operations are performed without reading the status, all but the first are ignored.

Both count and status of the selected counters may be latched simultaneously. This is functionally the same as issuing two consecutive, separate Read Back commands. If multiple count and/or status Read Back commands are issued to the same counters without any intervening reads, all but the first are ignored.

If both count and status of a counter are latched, the first read operation from that counter returns the latched status, regardless of which was latched first. The next one or two reads, depending on whether the counter is programmed for one or two type counts, returns the latched count. Subsequent reads return unlatched count.

18.10 PCU - iLB - High Precision Event Timer (HPET)

This function provides a set of timers that to be used by the operating system for timing events. One timer block is implemented, containing one counter and three timers.

18.10.1 Features

18.10.1.1 Non-Periodic Mode - All Timers

This mode can be thought of as creating a one-shot. When a timer is set up for non-periodic mode, it generates an interrupt when the value in the main counter matches the value in the timer's comparator register. As timers 1 and 2 are 32-bit, they will generate another interrupt when the main counter wraps.

T0CV cannot be programmed reliably by a single 64-bit write in a 32-bit environment unless only the periodic rate is being changed. If T0CV needs to be re-initialized, the following algorithm is performed:

1. Set T0C.TVS
2. Set T0CV[31:0]
3. Set T0C.TVS
4. Set T0CV[63:32]

Every timer is required to support the non-periodic mode of operation.

18.10.1.2 Periodic Mode - Timer 0 Only

When set up for periodic mode, when the main counter value matches the value in T0CV, an interrupt is generated (if enabled). Hardware then increases T0CV by the last value written to T0CV. During run-time, T0CV can be read to find out when the next periodic interrupt will be generated. Software is expected to remember the last value written to T0CV.

Example: if the value written to T0CV is 00000123h, then

- An interrupt will be generated when the main counter reaches 00000123h.
- T0CV will then be adjusted to 00000246h.
- Another interrupt will be generated when the main counter reaches 00000246h.
- T0CV will then be adjusted to 00000369h.

When the incremented value is greater than the maximum value possible for T0CV, the value will wrap around through 0. For example, if the current value in a 32-bit timer is FFFF0000h and the last value written to this register is 20000, then after the next interrupt the value will change to 00010000h.

If software wants to change the periodic rate, it writes a new value to T0CV. When the timer's comparator matches, the new value is added to derive the next matching point. If software resets the main counter, the value in the comparator's value register must also be reset by setting T0C.TVS. To avoid race conditions, this should be done with the main counter halted. The following usage model is expected:

1. Software clears GCFG.EN to prevent any interrupts.
2. Software clears the main counter by writing a value of 00h to it.
3. Software sets T0C.TVS.
4. Software writes the new value in T0CV.
5. Software sets GCFG.EN to enable interrupts.

18.10.1.2.1 Interrupts

If each timer has a unique interrupt and the timer has been configured for edge-triggered mode, then there are no specific steps required. If configured to level-triggered mode, then its interrupt must be cleared by software by writing a '1' back to the bit position for the interrupt to be cleared.

Interrupts associated with the various timers have several interrupt mapping options. Software should mask GCFG.LRE when reprogramming HPET interrupt routing to avoid spurious interrupts.

18.10.1.2.2 Mapping Option #1: Legacy Option (GCFG.LRE set)

This forces the following mapping:

Table 116. 8254 Interrupt Mapping

Timer	8259 Mapping	APIC Mapping	Comment
0	IRQ0	IRQ2	The 8254 timer will not cause any interrupts
1	IRQ8	IRQ8	RTC will not cause any interrupts.
2	T2C.IR	T2C.IRC	

18.10.1.2.3 Mapping Option #2: Standard Option (GCFG.LRE cleared)

Each timer has its own routing control. The interrupts can be routed to various interrupts in the I/O APIC. T[2:0]C.IRC indicates which interrupts are valid options for routing. If a timer is set for edge-triggered mode, the timers should not be shared with any other interrupts.

18.10.1.3 S0ix Support

During S0i1, the HPET is kept running. During S0i2 & S0i3, the HPET is halted.

18.10.1.4 S0ix Support

Prior to entry into S0i2 or S0i3 state, the driver/OS must set HPET_GCFG.EN to 0b to indicate RTD3$_{hot}$ status.

18.10.2 References

IA-PC HPET (High Precision Event Timers) Specification, Revision 1.0a: http://www.intel.com/hardware design/hpetspec_1.pdf.

18.10.3 Memory Mapped Registers

The register space is memory mapped to a 1K block at address FED00000h. All registers are in the core well. Accesses that cross register boundaries result in undefined behavior.

18.11 PCU - iLB - GPIO

187 GPIOs are available for use. Most of these GPIOs can be used as legacy GPIOs. This chapter describes their use as legacy GPIOs.

18.11.1 Signal Descriptions

Refer Chapter 2, "Physical Interfaces" for additional details.

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin.
- **Direction**: The buffer direction can be either input, output, or I/O (bidirectional).
- **Type:** The buffer type found in Chapter 19, "Electrical Specifications".
- **Description**: A brief explanation of the signal's function.

18.11.2 Features

GPIOs can generate general purpose events (GPEs) on rising and/or falling edges.

18.11.2.1 GPIO Controller

The GPIO controllers handle all GPIO interface to SoC,

- — GPIO NORTH - used for Camera sensors, DFX, SVID, and Display Pins.
- — GPIO SOUTHEAST - Defines the pads/Pins for MMC/SD host controller, LPC pins, FAST SPI pins and Platform Clock.
- — GPIO SOUTHWEST - Defines the Pads/Pins for HS UART,I2S HS, LPE, PCIe and SPI pins.
- — GPIO EAST - Defines the Pads/Pins for SoC power state related signals of PMU and ISH pins.

18.11.3 Usage

Each GPIO has six registers that control how it is used, or report its status:

- Use Select
- I/O Select
- GPIO Level
- Trigger Positive Edge
- Trigger Negative Edge
- Trigger Status

The Use Select register selects a GPIO pin as a GPIO, or leaves it as its programmed function. This register must be set for all other registers to affect the GPIO.

The I/O Select register determines the direction of the GPIO.

The Trigger Positive Edge and Trigger Negative Edge registers enable general purpose events on a rising and falling edge respectively. This only applies to GPIOs set as input.

The Trigger Status register is used by software to determine if the GPIO triggered a GPE. This only applies to GPIOs set as input and with one or both of the Trigger modes enabled.

Additionally, there is one additional register for each S5 GPIO:

- Wake Enable

This register allows S5 GPIOs to trigger a wake event based on the Trigger registers' settings.

18.11.4 GPIO Registers

18.11.4.1 SD Card and LPC Pins (3.3V versus 1.8V Modes)

The CFIO cells for both the SD Card Pins (SDMMC3_*) and LPC (LPC_*) are 3.3V capable.

To use as 1.8V IOs:

- Set power supply to 1.8V for the pads.
- Set v1p8mode in family configuration register.
- Trigger a RCOMP cycle using Family RCOMP register
- Copy RCOMP value to Family p and n strength values.

Note: All GPIO registers must be accessed as double words. Unpredictable results will occur otherwise.

Note: All MMIO GPIO *_PAD_VAL's must set Ienenb = 0 in order to read the pad_val of the GPIO. This applies to RO GPIO's as well.

18.12 PCU - iLB - Interrupt Decoding and Routing

The interrupt decoder is responsible for receiving interrupt messages from other devices in the SoC and decoding them for consumption by the interrupt router, the "PCU - iLB - 8259 Programmable Interrupt Controllers (PIC)" and/or the "PCU - iLB - IO APIC".

The interrupt router is responsible for mapping each incoming interrupt to the appropriate PIRQx, for consumption by the "PCU - iLB - 8259 Programmable Interrupt Controllers (PIC)" and/or the "PCU - iLB - IO APIC".

18.12.1 Features

18.12.1.1 Interrupt Decoder

The interrupt decoder receives interrupt messages from devices in the SoC. These interrupts can be split into two primary groups:

- For consumption by the interrupt router
- For consumption by the 8259 PIC

For Consumption by Interrupt Router

When a PCI-mapped device in the SoC asserts or de-asserts an INT[A:D] interrupt, an interrupt message is sent to the decoder. This message is decoded to indicate to the interrupt router which specific interrupt is asserted or de-asserted and which device the INT[A:D] interrupt originated from.

For Consumption by the 8259 PIC

When a device in the SoC asserts or de-asserts a legacy interrupt (IRQ), an interrupt message is sent to the decoder. This message is decoded to indicate to the 8259 PIC, which specific interrupt (IRQ[3, 4, 14 or 15]) was asserted or de-asserted.

18.12.1.2 Interrupt Router

The interrupt router aggregates the INT[A:D] interrupts for each PCI-mapped device in the SoC, received from the interrupt decoder, and the INT[A:D] interrupts direct from the Serialized IRQ controller. It then maps these aggregated interrupts to 8 PCI based interrupts: PIRQ[A:H]. This mapping is configured using the IR[31:0] registers.

PCI based interrupts PIRQ[A:H] are then available for consumption by either the 8259 PICs or the IO-APIC, depending on the configuration of the 8 PIRQx Routing Control Registers: PIRQA, PIQRB, PIRQC, PIRQD, PIRQE, PIRQF, PIRQG, PIRQH.

Routing PCI Based Interrupts to 8259 PIC

The interrupt router can be programmed to allow PIRQA-PIRQH to be routed internally to the 8259 as ISA compatible interrupts IRQ 3–7, 9–12 & 14–15. The assignment is programmable through the 8 PIRQx Routing Control Registers: PIRQA, PIQRB, PIRQC, PIRQD, PIRQE, PIRQF, PIRQG, PIRQH. One or more PIRQs can be routed to the same IRQ input. If ISA Compatible Interrupts are not required, the Route registers can be programmed to disable steering.

The PIRQx# lines are defined as active low, level sensitive. When a PIRQx# is routed to specified IRQ line, software must change the IRQ's corresponding ELCR bit to level sensitive mode. The SoC internally inverts the PIRQx# line to send an active high level to the PIC. When a PCI interrupt is routed onto the PIC, the selected IRQ can no longer be used by an active high device (through SERIRQ). However, active low interrupts can share their interrupt with PCI interrupts.

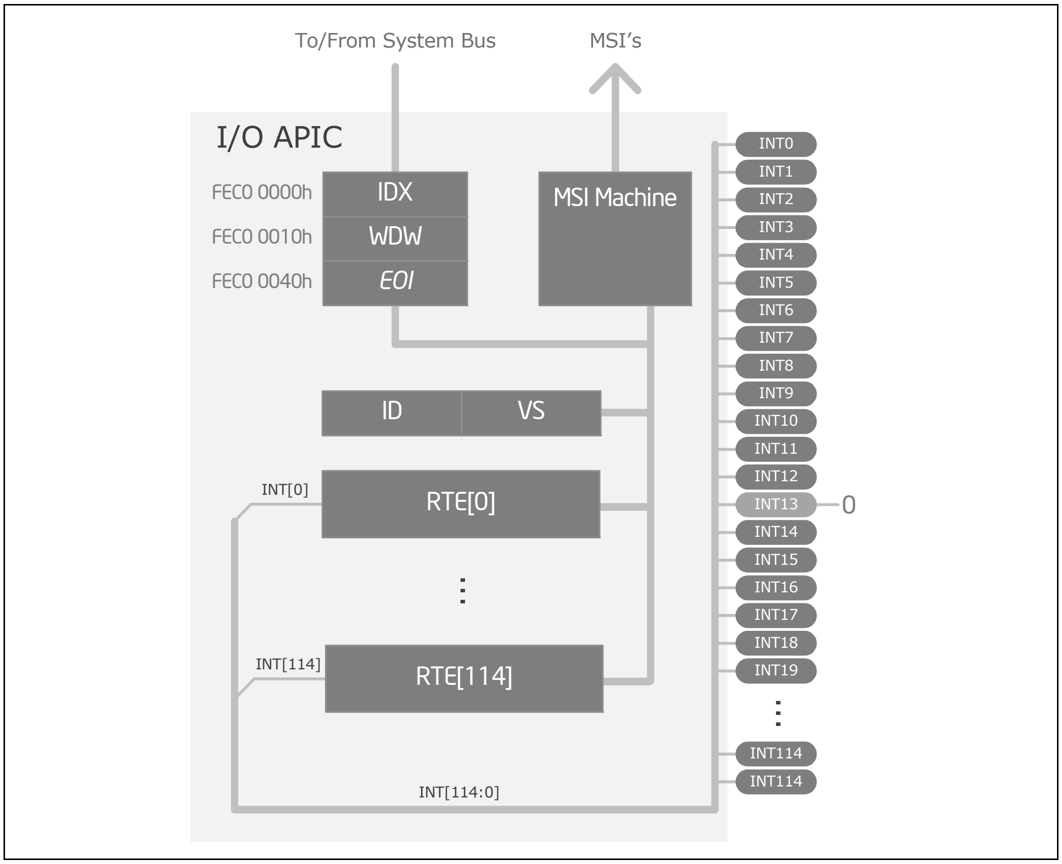
18.13 PCU - iLB - IO APIC

The IO Advanced Programmable Interrupt Controller (APIC) is used to support line interrupts more flexibly than the 8259 PIC. Line interrupts are routed to it from multiple sources, including legacy devices, via the interrupt decoder and serial IRQs, or they are routed to it from the interrupt router in the iLB. These line based interrupts are then used to generate interrupt messages targeting the local APIC in the processor.

18.13.1 Features

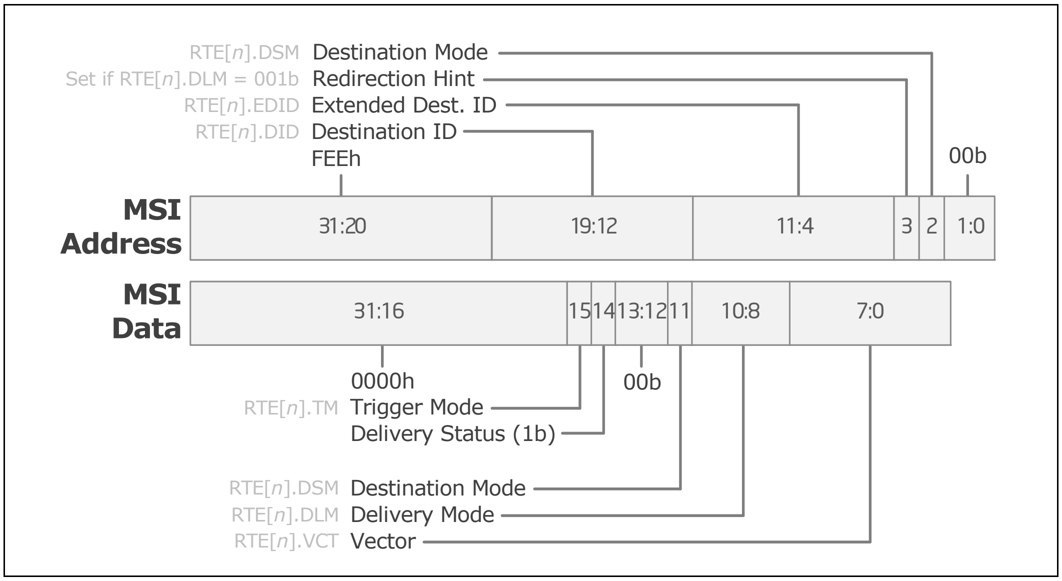
- 115 interrupt lines
  - IRQ0-114
- Edge or level trigger mode per interrupt
- Active low or high polarity per interrupt
- Works with local APIC in processor via MSIs
- MSIs can target specific processor core
- Established APIC programming model

Figure 33. Detailed Block Diagram



MSIs generated by the I/O APIC are sent as 32-bit memory writes to the Local APIC. The address and data of the write transaction are used as follows.

Figure 34. MSI Address and Data



Destination ID (DID) and Extended Destination ID (EDID) are used to target a specific processor core’s local APIC.

18.13.2 Usage

The I/O APIC contains indirectly accessed I/O APIC registers and normal memory mapped registers. There are three memory mapped registers:

- Index Register (IDX)
- Window Register (WDW)
- End Of Interrupt Register (EOI)

The Index register selects an indirect I/O APIC register (ID/VS/RTE[*n*]) to appear in the Window register.

The Window register is used to read or write the indirect register selected by the Index register.

The EOI register is written to by the Local APIC in the processor. The I/O APIC compares the lower eight bits written to the EOI register to the Vector set for each interrupt (RTE.VCT). All interrupts that match this vector will have their RTE.RIRR register cleared. All other EOI register bits are ignored.

18.13.3 Indirect I/O APIC Registers

These registers are selected with the IDX register, and read/written through the WDW register. Accessing these registers must be done as DW requests, otherwise unspecified behavior will result. Software should not attempt to write to reserved registers. Reserved registers may return non-zero values when read.

Note: There is one pair of redirection (RTE) registers per interrupt line. Each pair forms a 64-bit RTE register.

Note: Specified offsets should be placed in IDX, not added to IDX.

18.14 PCU - iLB - 8259 Programmable Interrupt Controllers (PIC)

SoC provides an ISA-compatible programmable interrupt controller (PIC) that incorporates the functionality of two, cascaded 8259 interrupt controllers.

18.14.1 Features

In addition to providing support for ISA compatible interrupts, this interrupt controller can also support PCI based interrupts (PIRQs) by mapping the PCI interrupt onto a compatible ISA interrupt line. Each 8259 controller supports eight interrupts, numbered 0–7. Table 117 shows how the controllers are connected.

Note: SoC does not implement any external PIRQ# signals. The PIRQs referred to in this chapter originate from the interrupt routing unit.

Table 117. Interrupt Controller Connections

8259	8259 Input	Connected Pin / Function
Master	0	Internal Timer / Counter 0 output or HPET #0; determined by GCFG.LRE register bit
	1	IRQ1 using SERIRQ, Keyboard Emulation
	2	Slave controller INTR output
	3	IRQ3 via SERIRQ, PIRQx or PCU UART 1
	4	IRQ4 via SERIRQ or PIRQx
	5	IRQ5 via SERIRQ or PIRQx
	6	IRQ6 via SERIRQ or PIRQx
	7	IRQ7 via SERIRQ or PIRQx
Slave	0	Inverted IRQ8# from internal RTC or HPET
	1	IRQ9 via SERIRQ, SCI or PIRQx
	2	IRQ10 via SERIRQ, SCI or PIRQx
	3	IRQ11 via SERIRQ, SCI, HPET or PIRQx
	4	IRQ12 via SERIRQ, PIRQx or mouse emulation
	5	None
	6	PIRQx
	7	IRQ15 via SERIRQ or PIRQx o

The SoC cascades the slave controller onto the master controller through master controller interrupt input 2. This means there are only 15 possible interrupts for the SoC PIC.

Interrupts can be programmed individually to be edge or level, except for IRQ0, IRQ2 and IRQ8#.

Note: Active-low interrupt sources (such as a PIRQ#) are inverted inside the SoC. In the following descriptions of the 8259s, the interrupt levels are in reference to the signals at the internal interface of the 8259s, after the required inversions have occurred. Therefore, the term "high" indicates "active," which means "low" on an originating PIRQ#.

18.14.1.1 Interrupt Handling

Generating Interrupts

The PIC interrupt sequence involves three bits, from the IRR, ISR, and IMR, for each interrupt level. These bits are used to determine the interrupt vector returned, and status of any other pending interrupts. Table 118 defines the IRR, ISR, and IMR.

Table 118. Interrupt Status Registers

Bit	Description
IRR	**Interrupt Request Register.** This bit is set on a low to high transition of the interrupt line in edge mode, and by an active high level in level mode.
ISR	**Interrupt Service Register.** This bit is set, and the corresponding IRR bit cleared, when an interrupt acknowledge cycle is seen, and the vector returned is for that interrupt.
IMR	**Interrupt Mask Register.** This bit determines whether an interrupt is masked. Masked interrupts will not generate INTR.

Acknowledging Interrupts

The processor generates an interrupt acknowledge cycle that is translated into a Interrupt Acknowledge Cycle to the SoC. The PIC translates this command into two internal INTA# pulses expected by the 8259 controllers. The PIC uses the first internal INTA# pulse to freeze the state of the interrupts for priority resolution. On the second INTA# pulse, the master or slave sends the interrupt vector to the processor with the acknowledged interrupt code. This code is based upon the ICW2.IVBA bits, combined with the ICW2.IRL bits representing the interrupt within that controller.

Note: References to ICWx and OCWx registers are relevant to both the master and slave 8259 controllers.

Table 119. Content of Interrupt Vector Byte

Master, Slave Interrupt	Bits [7:3]	Bits [2:0]
IRQ7,15	ICW2.IVBA	111
IRQ6,14		110
IRQ5,13		101
IRQ4,12		100
IRQ3,11		011
IRQ2,10		010
IRQ1,9		001
IRQ0,8		000

Hardware/Software Interrupt Sequence

1. One or more of the Interrupt Request lines (IRQ) are raised high in edge mode, or seen high in level mode, setting the corresponding IRR bit.
2. The PIC sends INTR active to the processor if an asserted interrupt is not masked.
3. The processor acknowledges the INTR and responds with an interrupt acknowledge cycle.
4. Upon observing the special cycle, the SoC converts it into the two cycles that the internal 8259 pair can respond to. Each cycle appears as an interrupt acknowledge pulse on the internal INTA# pin of the cascaded interrupt controllers.
5. Upon receiving the first internally generated INTA# pulse, the highest priority ISR bit is set and the corresponding IRR bit is reset. On the trailing edge of the first pulse, a slave identification code is broadcast by the master to the slave on a private, internal three bit wide bus. The slave controller uses these bits to determine if it must respond with an interrupt vector during the second INTA# pulse.
6. Upon receiving the second internally generated INTA# pulse, the PIC returns the interrupt vector. If no interrupt request is present because the request was too short in duration, the PIC returns vector 7 from the master controller.
7. This completes the interrupt cycle. In AEOI mode the ISR bit is reset at the end of the second INTA# pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

18.14.1.2 Initialization Command Words (ICWx)

Before operation can begin, each 8259 must be initialized. In the SoC, this is a four byte sequence. The four initialization command words are referred to by their acronyms: ICW1, ICW2, ICW3, and ICW4.

The base address for each 8259 initialization command word is a fixed location in the I/O memory space: 20h for the master controller, and A0h for the slave controller.

ICW1

A write to the master or slave controller base address with data bit 4 equal to 1 is interpreted as a write to ICW1. Upon sensing this write, the PIC expects three more byte writes to 21h for the master controller, or A1h for the slave controller, to complete the ICW sequence.

A write to ICW1 starts the initialization sequence during which the following automatically occur:

1. Following initialization, an interrupt request (IRQ) input must make a low-to-high transition to generate an interrupt.
2. The Interrupt Mask Register is cleared.
3. IRQ7 input is assigned priority 7.
4. The slave mode address is set to 7.
5. Special mask mode is cleared and Status Read is set to IRR.

ICW2

The second write in the sequence (ICW2) is programmed to provide bits [7:3] of the interrupt vector that will be released during an interrupt acknowledge. A different base is selected for each interrupt controller.

ICW3

The third write in the sequence (ICW3) has a different meaning for each controller.

- For the master controller, ICW3 is used to indicate which IRQ input line is used to cascade the slave controller. Within the SoC, IRQ2 is used. Therefore, MICW3.CCC is set to a 1, and the other bits are set to 0s.
- For the slave controller, ICW3 is the slave identification code used during an interrupt acknowledge cycle. On interrupt acknowledge cycles, the master controller broadcasts a code to the slave controller if the cascaded interrupt won arbitration on the master controller. The slave controller compares this identification code to the value stored in its ICW3, and if it matches, the slave controller assumes responsibility for broadcasting the interrupt vector.

ICW4

The final write in the sequence (ICW4) must be programmed for both controllers. At the very least, ICW4.MM must be set to a 1 to indicate that the controllers are operating in an Intel Architecture-based system.

18.14.1.3 Operation Command Words (OCW)

These command words reprogram the Interrupt controller to operate in various interrupt modes.

- OCW1 masks and unmasks interrupt lines.

- OCW2 controls the rotation of interrupt priorities when in rotating priority mode, and controls the EOI function.
- OCW3 sets up ISR/IRR reads, enables/disables the special mask mode (SMM), and enables/disables polled interrupt mode.

18.14.1.4 Modes of Operation

Fully Nested Mode

In this mode, interrupt requests are ordered in priority from 0 through 7, with 0 being the highest. When an interrupt is acknowledged, the highest priority request is determined and its vector placed on the bus. Additionally, the ISR for the interrupt is set. This ISR bit remains set until: the processor issues an EOI command immediately before returning from the service routine; or if in AEOI mode, on the trailing edge of the second INTA#. While the ISR bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels generate another interrupt.

Interrupt priorities can be changed in the rotating priority mode.

Special Fully-Nested Mode

This mode is used in the case of a system where cascading is used, and the priority has to be conserved within each slave. In this case, the special fully-nested mode is programmed to the master controller. This mode is similar to the fully-nested mode with the following exceptions:

- When an interrupt request from a certain slave is in service, this slave is not locked out from the master's priority logic and further interrupt requests from higher priority interrupts within the slave are recognized by the master and initiate interrupts to the processor. In the normal-nested mode, a slave is masked out when its request is in service.
- When exiting the Interrupt Service routine, software has to check whether the interrupt serviced was the only one from that slave. This is done by sending a Non-Specific EOI command to the slave and then reading its ISR. If it is 0, a non-specific EOI can also be sent to the master.

Automatic Rotation Mode (Equal Priority Devices)

In some applications, there are a number of interrupting devices of equal priority. Automatic rotation mode provides for a sequential 8-way rotation. In this mode, a device receives the lowest priority after being serviced. In the worst case, a device requesting an interrupt has to wait until each of seven other devices are serviced at most once.

There are two ways to accomplish automatic rotation using OCW2.REOI; the Rotation on Non-Specific EOI Command (OCW2.REOI=101b) and the rotate in automatic EOI mode which is set by (OCW2.REOI=100b).

Specific Rotation Mode (Specific Priority)

Software can change interrupt priorities by programming the bottom priority. For example, if IRQ5 is programmed as the bottom priority device, then IRQ6 is the highest priority device. The Set Priority Command is issued in OCW2 to accomplish this, where: OCW2.REOI=11xb, and OCW2.ILS is the binary priority level code of the bottom priority device.

In this mode, internal status is updated by software control during OCW2. However, it is independent of the EOI command. Priority changes can be executed during an EOI command by using the Rotate on Specific EOI Command in OCW2 (OCW2.REOI=111b) and OCW2.ILS=IRQ level to receive bottom priority.

Poll Mode

Poll mode can be used to conserve space in the interrupt vector table. Multiple interrupts that can be serviced by one interrupt service routine do not need separate vectors if the service routine uses the poll command. Poll mode can also be used to expand the number of interrupts. The polling interrupt service routine can call the appropriate service routine, instead of providing the interrupt vectors in the vector table. In this mode, the INTR output is not used and the microprocessor internal Interrupt Enable flip-flop is reset, disabling its interrupt input. Service to devices is achieved by software using a Poll command.

The Poll command is issued by setting OCW3.PMC. The PIC treats its next I/O read as an interrupt acknowledge, sets the appropriate ISR bit if there is a request, and reads the priority level. Interrupts are frozen from the OCW3 write to the I/O read. The byte returned during the I/O read contains a 1 in Bit 7 if there is an interrupt, and the binary code of the highest priority level in Bits 2:0.

Edge and Level Triggered Mode

In ISA systems this mode is programmed using ICW1.LTIM, which sets level or edge for the entire controller. In the SoC, this bit is disabled and a register for edge and level triggered mode selection, per interrupt input, is included. This is the Edge/Level control Registers ELCR1 and ELCR2.

If an ELCR bit is 0, an interrupt request will be recognized by a low-to-high transition on the corresponding IRQ input. The IRQ input can remain high without generating another interrupt. If an ELCR bit is 1, an interrupt request will be recognized by a high level on the corresponding IRQ input and there is no need for an edge detection. The interrupt request must be removed before the EOI command is issued to prevent a second interrupt from occurring.

In both the edge and level triggered modes, the IRQ inputs must remain active until after the falling edge of the first internal INTA#. If the IRQ input goes inactive before this time, a default IRQ7 vector is returned.

18.14.1.5 End of Interrupt (EOI) Operations

An EOI can occur in one of two fashions: by a command word write issued to the PIC before returning from a service routine, the EOI command; or automatically when the ICW4.AEOI bit is set to 1.

Normal End of Interrupt

In normal EOI, software writes an EOI command before leaving the interrupt service routine to mark the interrupt as completed. There are two forms of EOI commands: Specific and Non-Specific. When a Non-Specific EOI command is issued, the PIC clears the highest ISR bit of those that are set to 1. Non-Specific EOI is the normal mode of operation of the PIC within the SoC, as the interrupt being serviced currently is the interrupt entered with the interrupt acknowledge. When the PIC is operated in modes that preserve the fully nested structure, software can determine which ISR bit to clear by issuing a Specific EOI.

An ISR bit that is masked is not cleared by a Non-Specific EOI if the PIC is in the special mask mode. An EOI command must be issued for both the master and slave controller.

Automatic End of Interrupt Mode

In this mode, the PIC automatically performs a Non-Specific EOI operation at the trailing edge of the last interrupt acknowledge pulse. From a system standpoint, this mode should be used only when a nested multi-level interrupt structure is not required within a single PIC. The AEOI mode can only be used in the master controller and not the slave controller.

Note: Both the master and slave PICs have an AEOI bit: MICW4.AEOI and SICW4.AEOI respectively. Only the MICW4.AEOI bit should be set by software. The SICW4.AEOI bit should not be set by software.

18.14.1.6 Masking Interrupts

Masking on an Individual Interrupt Request

Each interrupt request can be masked individually by the Interrupt Mask Register (IMR). This register is programmed through OCW1. Each bit in the IMR masks one interrupt channel. Masking IRQ2 on the master controller masks all requests for service from the slave controller.

Special Mask Mode

Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control. For example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion.

The special mask mode enables all interrupts not masked by a bit set in the Mask register. Normally, when an interrupt service routine acknowledges an interrupt without issuing an EOI to clear the ISR bit, the interrupt controller inhibits all lower priority requests. In the special mask mode, any interrupts may be selectively enabled by loading the Mask Register with the appropriate pattern.

The special mask mode is set by OCW3.ESMM=1b & OCW3.SMM=1b, and cleared where OCW3.ESMM=1b & OCW3.SMM=0b.

18.14.1.7 S0ix Support

During S0i2 & S0i3, the 8259 PICs are disabled. A platform that requires the 8259 PICs to be always active, should disable S0i2/3 using the S0ix_Enable register.

18.14.2 IO Mapped Registers

The interrupt controller registers are located at 20h and 21h for the master controller (IRQ0 - 7), and at A0h and A1h for the slave controller (IRQ8 - 13). These registers have multiple functions, depending upon the data written to them. Table 120 is a description of the different register possibilities for each address.

Note: The register descriptions after Table 120 represent one register possibility.

Table 120. I/O Registers Alias Locations

Registers	Original I/O Location	Alias I/O Locations
MICW1 MOCW2 MOCW3	20h	24h
		28h
		2Ch
		30h
		34h
		38h
		3Ch
MICW2 MICW3 MICW4 MOCW1	21h	25h
		29h
		2Dh
		31h
		35h
		39h
		3Dh

Table 120. I/O Registers Alias Locations

Registers	Original I/O Location	Alias I/O Locations
SICW1 SOCW2 SOCW3	A0h	A4h
		A8h
		ACh
		B0h
		B4h
		B8h
		BCh
SICW2 SICW3 SICW4 SOCW1	A1h	A5h
		A9h
		ADh
		B1h
		B5h
		B9h
		BDh
ELCR1	4D0h	N/A
ELCR2	4D1h	N/A

18.15 Register Map

For more information on Platform Controller Unit (PCU) registers refer Intel® Atom™ Z8000 Processor Series Datasheet (Volume 2 of 2), Doc ID:332066.

§

19 Electrical Specifications

This chapter is categorized into the following sections:

- "Absolute Maximum and Minimum Specifications"
- "Thermal Specifications"
- "Storage Conditions"
- "Voltage and Current Specifications"
- "Crystal Specifications"
- "DC Specifications"

19.1 Absolute Maximum and Minimum Specifications

The absolute maximum and minimum specifications are used to specify conditions allowable outside of the functional limits of the SoC, but with possible reduced life expectancy once returned to function limits.

At conditions exceeding absolute specifications, neither functionality nor long term reliability can be expected. Parts may not function at all once returned to functional limits.

Although the processor contains protective circuitry to resist damage from Electrostatic discharge (ESD), precautions should always be taken to avoid high static voltages or electric fields.

19.2 Thermal Specifications

These specifications define the operating thermal limits of the SoC. Thermal solutions not designed to provide the following level of thermal capability may affect the long-term reliability of the processor and system, but more likely result in performance throttling to ensure silicon junction temperatures within specification.

This section specifies the thermal specifications for all SKUs. Some definitions are needed, however. "Tj Max" defines the maximum operating silicon junction temperature. Unless otherwise specified, all specifications in this document assume Tj Max as the worse case junction temperature. This is the temperature needed to ensure TDP specifications when running at guaranteed CPU and graphics frequencies. "TDP" defines the thermal dissipated power for a worse case estimated real world thermal scenario. "SDP", or scenario dissipated power, defines the thermal dissipated power under a lighter workload specific to a user scenario and at a lower thermal junction temperature than Tj Max. Note that turbo frequencies are opportunistically selected when thermal headroom exists. Automatic throttling along with a proper thermal solution ensure Tj Max will not be exceeded.

Table 121. Thermal Specifications

	T4	T3
T_j Max	90 °C	90 °C
T_j Min	0 °C	0 °C
T_j @ Max. Steady State Power (SDP)	70 °C	70 °C
SDP	2W	2.2 W

19.3 Storage Conditions

This section specifies absolute maximum and minimum storage temperature and humidity limits for given time durations. Failure to adhere to the specified limits could result in physical damage to the component. If this is suspected, Intel recommends a visual inspection to determine possible physical damage to the silicon or surface components.

Table 122. Storage Conditions Prior to Board Attach

Symbol	Parameter	Min	Max
Tabsolute storage	Device storage temperature when exceeded for any length of time.	-25 °C	125 °C
Tshort term storage	The ambient storage temperature and time for up to 72 hours.	-25 °C	85 °C
Tsustained storage	The ambient storage temperature and time for up to 30 months.	5 °C	40 °C
RHsustained storage	The maximum device storage relative humidity for up to 30 months.		60% @ 24 °C

NOTES:

- Specified temperatures are not to exceed values based on data collected. Exceptions for surface mount re-flow are specified by the applicable JEDEC standard. Non-adherence may affect processor reliability.
- Component product device storage temperature qualification methods may follow JESD22-A119 (low temperature) and JESD22-A103 (high temperature) standards when applicable for volatile memory.
- Component stress testing is conducted in conformance with JESD22-A104.
- The JEDEC J-JSTD-020 moisture level rating and associated handling practices apply to all moisture sensitive devices removed from the moisture barrier bag.

19.3.1 Post Board-Attach

The storage condition limits for the component once attached to the application board are not specified. Intel does not conduct component-level certification assessments post board-attach given the multitude of attach methods, socket types, and board types used by customers.

Provided as general guidance only, board-level Intel-branded products are specified and certified to meet the following temperature and humidity limits:

- Non-Operating Temperature Limit: -40 °C to 70 °C
- Humidity: 50% to 90%, non-condensing with a maximum wet-bulb of 28 °C

19.4 Voltage and Current Specifications

The I/O buffer supply voltages are specified at the SoC package balls. The tolerances shown in Table 138 are inclusive of all noise from DC up to 20 MHz. The voltage rails should be measured with a bandwidth limited oscilloscope with a roll-off of 3 dB/decade above 20 MHz under all operating conditions. Table 124 indicates which supplies are connected directly to a voltage regulator or to a filtered voltage rail. For voltage rails that are connected to a filter, they should be measured at the input of the filter. If the recommended platform decoupling guidelines cannot be met, the system designer will have to make trade-offs between the voltage regulator out DC tolerance and the decoupling performances of the capacitor network to stay within the voltage tolerances listed below.

Note: The SoC is a pre-launch product. Voltage and current specifications are subject to change.

Table 123. SoC Power Rail DC Specs and Max Current

Platform Rail		Voltage Tolerances	Max Icc
V1P05A	UNCORE1_V1P05A_G3	1.05 V DC: ±2% AC: ±2%	1700 mA
	UNCORE2_V1P05A_G3		
	DDR_V1P05A_G3		
	USB3_V1P05A_G3		
	USBSSIC_V1P05A_G3		
	F_V1P05A_G3		
	PCIECLK_V1P05A_G3		
V1P15	CORE_V1P15_S0iX	1.15 V DC: ±2% AC: ±3%	2100 mA
	DDI_V1P15_S0iX		
	UNCORE_V1P15_S0iX		
	F_V1P15_S0iX		

Table 123. SoC Power Rail DC Specs and Max Current

Platform Rail		Voltage Tolerances	Max Icc
V1P2A	USBSSIC_V1P2A_G3	1.24 V DC: ±2% AC: ±2%	67 mA
	MIPI_V1P2A_G3		
	USBHSIC_V1P2A_G3		
V1P8A	USB_V1P8A_G3	1.8 V DC: ±2% AC: ±2%	971 mA
	UNCORE_V1P8A_G3		
	GPIOSE_V1P8A_G3		
	GPION_V1P8A_G3		
	F_V1P8A_G3		
V3P3A	USB_V3P3A_G3	3.3 V DC: ±2% AC: ±2%	196 mA
	F_V3P3A_G3		
	RTC_V3P3A_G5		
V3P3A_V1P8A	SDIO_V3P3A_V1P8A_G3	1.8 V/3.3 V DC: ±2% AC: ±2%	-
	LPC_V3P3A_V1P8A_S4		
VSFR	ICLK_VSFR_G3	1.05 V/1.24 V/ 1.35 V DC: ±2% AC: ±3%	-
	CORE0_VSFR_G3		
	CORE1_VSFR_G3		
	UNCORE_VSFR_G3		
VCC0	CORE_VCC0_S0iX	Refer Table 124	3200 mA
	CORE_VCC0_SENSE		
VCC1	CORE_VCC1_S0iX	Refer Table 124	3200 mA
	CORE_VCC1_SENSE		
VNN	UNCORE_VNN_S4	Refer Table 124	2500 mA
	UNCORE_VNN_SENSE		
VGG	DDI_VGG_S0iX	Refer Table 124	8000 mA
	DDI_VGG_SENSE		

Table 123. SoC Power Rail DC Specs and Max Current

Platform Rail		Voltage Tolerances	Max Icc
VDDQ	DDI1_VDDQ_G3	1.24 V/1.35 V DC: ±2% AC: ±2%	1900 - 2500mA
	DDI2_VDDQ_G3		
	USB_VDDQ_G3		
VDDQG	DDR_VDDQG_S4	1.24 V/1.35 V DC: ±2% AC: ±2%	
	DDRCH0_VDDQG_S4		
	DDRCH1_VDDQG_S4		
	DDRSFRCH0_VDDQG_S4		
	DDRSFRCH1_VDDQG_S4		
V3P3RTC	RTC_V3P3RTC_G5	G5: 2-3 V at battery Otherwise V3P3A (pre diode drop)	-

NOTE:

1. RTC_VCC average current draw (G5) is specified at 27°C under battery conditions
2. This value is applicable only for Z8350 and Z8300 SKUs.

19.4.1 VCC and VNN Voltage Specifications

Table 124 and Table 138 list the DC specifications for the SoC power rails. They are valid only while meeting specifications for junction temperature, clock frequency, and input voltages. Care should be taken to read all notes associated with each parameter.

Table 124. VCC and VNN DC Voltage Specifications

Symbol	Parameter	Min	Typ	Max	Unit	Note
CORE_VCC VID [Z8700, Z8750]	Core VID Target Range	0.6		1.30	V	
CORE_VCC VID [Z8500, Z8550]	Core VID Target Range	0.6		1.28	V	
CORE_VCC VID [Z8300, Z8350]	Core VID Target Range	0.6		1.13	V	
CORE_VCC0_S0iX	V_{CC0} for SoC Core 0	Refer VCC VID			V	2
CORE_VCC1_S0iX	V_{CC1} for SoC Core 1	Refer VCC VID			V	2
UNCORE_VNN VID [Z8700]	Uncore VID Target Range	0.4		1.28	V	
UNCORE_VNN VID [Z8500]	Uncore VID Target Range	0.4		1.28	V	
UNCORE_VNN VID [Z8300, Z8350]	Uncore VID Target Range	0.4		1.1	V	
UNCORE_VNN_S4	V_{NN} for SoC Uncore	Refer VNN VID			V	2
DDI_VGG_S0iX [Z8700]	V_{GG} for SoC Display	0.4		0.9	V	
DDI_VGG_S0iX [Z8500]	V_{GG} for SoC Display	0.4		0.9	V	
DDI_VGG_S0iX [Z8300, Z8350]	V_{GG} for SoC Display	0.4		1.1	V	

Table 124. VCC and VNN DC Voltage Specifications

Symbol	Parameter	Min	Typ	Max	Unit	Note
CORE_VCC/UNCORE_VNN V_{BOOT}	Default target V_{CC}/V_{NN} voltage for initial power up.		1.0 or 1.1		V	3
VCC0/1 Tolerance	Tolerance of VCC0/1 voltage at VID target.	DC: ±2% AC: ±3%			%	1
VNN Tolerance	Tolerance of VNN voltage at VID target.	DC: ±2% AC: ±2%			%	1
VGG Tolerance	Tolerance of VGG voltage at VID target.	DC: ±2% AC: ±3%			%	1

NOTES:

1. Contact local Intel representative for load line and tolerance details.
2. Each SoC is programmed with voltage identification value (VID), which is set at manufacturing and cannot be altered. Individual VID values are calibrated during manufacturing such that two SoCs at the same frequency may have different settings within the VID range. Note this differs from the VID employed by the SoC during a power management event.
3. Refer VR12/IMVP7 Pulse Width Modulation specification for additional details. Either value is ok.

19.4.2 CPU ESD LEVEL

Table 125. CPU ESD level details

CPU Type	CDM	HBM
T4/T3	±250v	±1000v

19.5 Crystal Specifications

There are two crystal oscillators. One for RTC which maintains time and provides initial timing reference for power sequencing. The other is for the Integrated Clock, which covers clocking for the entire SoC.

Table 126. ILB RTC Crystal Specification

Symbol	Parameter	Min	Typ	Max	Units	Notes
F_{RTC}	Frequency	-	32.768	-	kHz	1
T_{PPM}	Crystal frequency tolerance (refer notes)	-	-	+/-20	ppm	1
P_{DRIVE}	Crystal drive load	-	0.1	0.5	uW	1
C_{LOAD}	Crystal load capacitance		12.5		pF	
C_{SHUNT}	Crystal shunt capacitance	-	1.3	-	pF	1
C_1/C_2	Load Capacitance tolerance			+/-10	%	

NOTES:

1. These are the specifications needed to select a crystal oscillator for the RTC circuit.

2. Crystal tolerance impacts RTC time. A 10 ppm crystal is recommended for 1.7 s tolerance per day, RTC circuit itself contributes addition 10 ppm for a total of 20 ppm in this example.

Table 127. Integrated Clock Crystal Specification

Symbol	Parameter	Min	Typ	Max	Units	Notes
F_{ICLK}	Frequency	-	19.2	-	MHz	1
T_{PPM}	Crystal frequency tolerance & stability	-	-	+/-30	ppm	1
P_{DRIVE}	Crystal drive load	-	-	100	uW	1
R_{ESR}	ESR	-	-	80	Ohm	1
C_{LOAD}	Crystal load capacitance		12		pF	
C_{SHUNT}	Crystal shunt capacitance	-	0.75	-	pF	1
C_1/C_2	Load Capacitance tolerance			+/-10	%	

NOTE:

1. These are the specifications required to select a crystal oscillator for the Integrated Clock circuit. Crystal must be AT cut, fundamental, parallel resonance.

19.6 DC Specifications

Platform reference voltages are specified at DC only. V_{REF} measurements should be made with respect to the supply voltages specified in "Voltage and Current Specifications".

Note: $V_{IH/OH}$ Max and $V_{IL/OL}$ Min values are bounded by reference voltages.

The following DC Specifications are explained in this section:

- "Display DC Specification"
- "MIPI-Camera Serial Interface (CSI) DC Specification"
- "SDIO DC Specification"
- "SD Card DC Specification"
- "eMMC 4.51 DC Specification"
- "JTAG DC Specification"
- "LPDDR3 Memory Controller DC Specification"
- "USB 2.0 Host DC Specification"
- "USB 3.0 DC Specification"
- "SSIC DC Specification"
- "SPI and FST_SPI DC Specification"
- "Power Management/Thermal (PMC) and RTC DC Specification"
- "SVID DC Specification"
- "GPIO DC Specification"

- "SIO - I2C DC Specification"
- "SIO - UART DC Specification"
- "I2S (Audio) DC Specification"
- "PCI Express DC Specification"

Note: Care should be taken to read all notes associated with each parameter.

19.6.1 Display DC Specification

DC specifications for display interfaces:

- "Display Port DC Specification"
- "HDMI DC Specification"
- "Embedded Display Port DC Specification"
- "Display Port AUX Channel DC Specification"
- "Embedded Display Port AUX Channel DC Specification"
- "DDC Signal DC Specification"
- "MIPI DSI DC Specification"

19.6.1.1 Display Port DC Specification

Table 128. Display Port DC specification

Symbol	Parameter	Min	Typ	Max	Units	Notes
$V_{TX-DIFFp-p-Level0}$	Differential Peak-to-peak Output Voltage Level 0	0.34	0.4	0.46	V	
$V_{TX-DIFFp-p-Level1}$	Differential Peak-to-peak Output Voltage Level 1	0.51	0.6	0.68	V	
$V_{TX-DIFFp-p-Level2}$	Differential Peak-to-peak Output Voltage Level 2	0.69	0.8	0.92	V	
$VT_{X-DIFFp-p-Level3}$	Differential Peak-to-peak Output Voltage Level 3	0.85	1.2	1.38	V	
$V_{TX-PREEMP-RATIO}$	No Pre-emphasis	0.0	0.0	0.0	dB	
	3.5 dB Pre-emphasis	2.8	3.5	4.2	dB	
	6.0 dB Pre-emphasis	4.8	6.0	7.2	dB	
	9.5 dB Pre-emphasis	7.5	9.5	11.4	dB	
$V_{TX-DC-CM}$	Tx DC Common Mode Voltage	0		2.0	V	

Table 128. Display Port DC specification

Symbol	Parameter	Min	Typ	Max	Units	Notes
$RL_{TX-DIFF}$	Differential Return Loss at 0.675GHz at Tx Package pins	12			dB	
	Differential Return Loss at 1.35 GHz at Tx Package pins	9			dB	1
C_{TX}	TX Output Capacitance			1.5	pF	2

NOTES:

1. Straight loss line between 0.675 GHz and 1.35 GHz.
2. Represents only the effective lump capacitance seen at the SoC interface that shunts the TX termination.

19.6.1.2 HDMI DC Specification

Table 129. HDMI DC specification

Symbol	Parameter	Min	Typ	Max	Units	Notes
Voff	Single Ended Standby (off), output voltage	-10		10	mV	1 @ AVcc
Vswing	Single Ended output swing voltage	400		600	mV	
V_{OH} (<=165 MHz)	Single Ended high level, output voltage	-10		10	mv	1 @ AVcc
V_{OH}(>165 MHz)	Single Ended high level, output voltage	-200		10	mV	1 @ AVcc
V_{OL}(<=165 MHz)	Single Ended low level, output voltage	-600		-400	mV	1 @ AVcc
V_{OL}(>165MHz)	Single Ended low level, output voltage	-700		-400	mV	1 @ AVcc

NOTE: 1. The min/max values are with reference to AVcc (Analog Voltage level) = 3.3V ±5%

19.6.1.3 Embedded Display Port DC Specification

Table 130. Embedded Display Port DC Specification

Symbol	Parameter	Min	Typ	Max	Units	Notes
$V_{TX-DIFFp-p-Level0}$	Differential Peak-to-peak Output Voltage Level 0	0.18	0.2	0.22	V	1,2
$V_{TX-DIFFp-p-Level1}$	Differential Peak-to-peak Output Voltage Level 1	0.2	0.25	0.275	V	1,2
$V_{TX-DIFFp-p-Level2}$	Differential Peak-to-peak Output Voltage Level 2	0.27	0.3	0.33	V	1,2

Table 130. Embedded Display Port DC Specification

Symbol	Parameter	Min	Typ	Max	Units	Notes
$VT_{X\text{-}DIFFp\text{-}p\text{-}Level3}$	Differential Peak-to-peak Output Voltage Level 3	0.315	0.35	0.385	V	1,2
$VT_{X\text{-}DIFFp\text{-}p\text{-}Level4}$	Differential Peak-to-peak Output Voltage Level 4	0.36	0.4	0.44	V	1,2
$VT_{X\text{-}DIFFp\text{-}p\text{-}Level5}$	Differential Peak-to-peak Output Voltage Level 5	0.405	0.45	0.495	V	1,2
$VT_{X\text{-}DIFFp\text{-}p\text{-}MAX}$	Maximum Allowed Differential Peak-to-peak Output Voltage			1.380	V	3
$V_{TX\text{-}DC\text{-}CM}$	Tx DC Common Mode Voltage	0		2.0	V	1,2,3
$V_{TX\text{-}PREEMP\text{-}RATIO}$	No Pre-emphasis	0.0	0.0	0.0	dB	1,2,3
	3.5 dB Pre-emphasis	2.8	3.5	4.2	dB	1,2,3
	6.0 dB Pre-emphasis	4.8	6.0	7.2	dB	1,2,3
	9.5 dB Pre-emphasis	7.5	9.5	11.4	dB	1,2,3
$RL_{TX\text{-}DIFF}$	Differential Return Loss at 0.675GHz at Tx Package pins	12			dB	4
	Differential Return Loss at 1.35 GHz at Tx Package pins	9			dB	4
C_{TX}	TX Output Capacitance			1.5	pF	5

NOTES:

1. Steps between VTX-DIFFP-P voltages must be monotonic. The actual VTX-DIFFP-P-1 voltage must be equal to or greater than the actual VTX-DIFFP-P-0 voltage; the actual VTX-DIFFP-P-2 voltage must be greater than the actual VTX-DIFFP-P-1 voltage; etc.
2. The recommended minimum VTX-DIFFP-P delta between adjacent voltages is mV.
3. Allows eDP Source devices to support differential signal voltages compatible with eDP v1.3 (and lower) devices and designs.
4. Straight loss line between 0.675 GHz and 1.35 GHz.
5. Represents only the effective lump capacitance seen at the SoC interface that shunts the TX termination.

19.6.1.4 Display Port AUX Channel DC Specification

Table 131. DDI AUX Channel DC Specification

Symbol	Parameter	Min	Typ	Max	Units	Notes
$V_{AUX\text{-}DIFFp\text{-}p}$	AUX Peak-to-peak Voltage at a transmitting Device	0.29		1.38	V	1
$V_{AUX\text{-}\_TERM\_R}$	AUX CH termination DC resistance		100		Ω	
$V_{AUX\text{-}DC\text{-}CM}$	AUX DC Common Mode Voltage	0		2.0	V	2
$V_{AUX\text{-}TURN\text{-}CM}$	AUX turn around common mode voltage			0.3	V	3
I_{AUX_SHORT}	AUX Short Circuit Current Limit			90	mA	4
C_{AUX}	AC Coupling Capacitor	75		200	nF	5

NOTES:
1. $V_{AUX\text{-}DIFFp\text{-}p}$= 2*|$V_{AUXP}$ – V_{AUXM}|
2. Common mode voltage is equal to V_{bias_Tx} (or V_{bias_Rx}) voltage.
3. Steady state common mode voltage shift between transmit and receive modes of operation.
4. Total drive current of the transmitter when it is shorted to its ground.
5. All DisplayPort Main Link lanes as well as AUX CH must be AC coupled. AC coupling capacitors must be placed on the transmitter side. Placement of AC coupling capacitors on the receiver side is optional.

19.6.1.5 Embedded Display Port AUX Channel DC Specification

Table 132. Embedded Display Port AUX Channel DC Specification

Symbol	Parameter	Min	Typ	Max	Units	Notes
$V_{AUX\text{-}DIFFp\text{-}p}$	AUX Peak-to-peak Voltage at a transmitting Device	0.29		1.38	V	1
$V_{AUX\text{-}\_TERM\_R}$	AUX CH termination DC resistance		100		Ω	
$V_{AUX\text{-}DC\text{-}CM}$	AUX DC Common Mode Voltage	0		1.2	V	2
$V_{AUX\text{-}TURN\text{-}CM}$	AUX turn around common mode voltage			0.3	V	3
I_{AUX_SHORT}	AUX Short Circuit Current Limit			90	mA	4
C_{AUX}	AC Coupling Capacitor	75		200	nF	5

NOTES:
1. $V_{AUX\text{-}DIFFp\text{-}p}$= 2*|$V_{AUXP}$ – V_{AUXM}|
2. Common mode voltage is equal to V_{bias_Tx} (or V_{bias_Rx}) voltage.
3. Steady state common mode voltage shift between transmit and receive modes of operation.

4. Total drive current of the transmitter when it is shorted to its ground.
5. All DisplayPort Main Link lanes as well as AUX CH must be AC coupled. AC coupling capacitors must be placed on the transmitter side. Placement of AC coupling capacitors on the receiver side is optional.

19.6.1.6 DDC Signal DC Specification

Table 133. DDC Signal DC Specification (DCC_DATA, DDC_CLK)

Symbol	Parameter	Min	Typ	Max	Units	Notes
V_{REF}	I/O Voltage	GPION_V1P8A_G3			V	
V_{IH}	Input High Voltage	$0.75*V_{REF}$			V	1
V_{IL}	Input Low Voltage			$0.35*V_{REF}$	V	2
V_{OL}	Output Low Voltage			0.45	V	3
I_i	Input Pin Leakage	-10		10	µA	4

NOTES:

1. V_{IH} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical high value
2. V_{IL} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical low value.
3. This buffer reaches VOH/VOL with 3mA load.
4. For VIN between 0V and CORE_VCC_S0iX. Measured when driver is tri-stated.

Table 134. DDC Misc Signal DC Specification (HPD, BKLTCTL, VDDEN, BKLTEN)

Symbol	Parameter	Min	Typ	Max	Units	Notes
V_{REF}	I/O Voltage	GPION_V1P8A_G3			V	
V_{IH}	Input High Voltage	$0.75*V_{REF}$			V	1
V_{IL}	Input Low Voltage			$0.35*V_{REF}$	V	2
Z_{pu}	Pull up Impedance	40	50	60	Ω	3
Z_{pd}	Pull down Impedance	40	50	60	Ω	3
I_i	Input Pin Leakage	-10		10	µA	4

NOTES:

1. V_{IH} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical high value.
2. V_{IL} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical low value.
3. Measured at CORE_VCC0_S0iX and CORE_VCC1_S0iX.
4. For VIN between 0V and CORE_VCC0_S0iX and CORE_VCC1_S0iX. Measured when driver is tri-stated.
5. This buffer reaches VOH/VOL with 3mA load.

Figure 35. Definition of Differential Voltage and Differential Voltage Peak-to-Peak

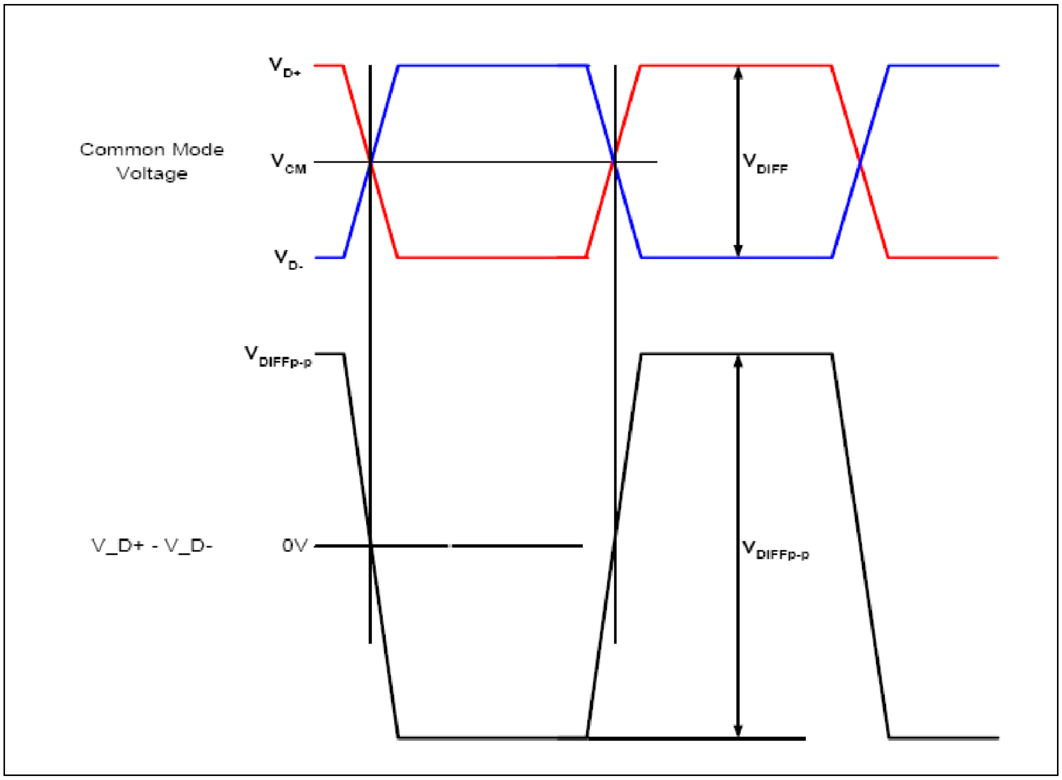
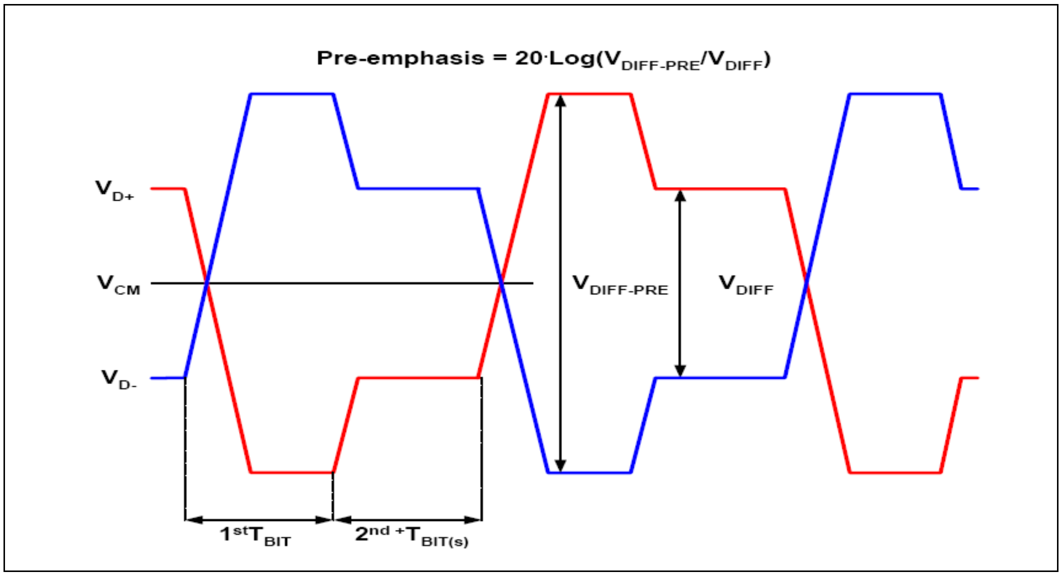


Figure 36. Definition of Pre-emphasis



19.6.1.7 MIPI DSI DC Specification

Table 135. MIPI DSI DC Specification

Symbol	Parameter	Min.	Nom.	Max.	Unit	Notes
ILEAK	Pin Leakage current	-10	–	10	µA	
MIPI DSI HS-TX Mode						
V_{CMTX}	HS transmit static common-mode voltage	150	200	250	mV	
$\|V_{CMTX(1,0)}\|$	V_{CMTX} mismatch when output is differential-1 or differential-0	–	–	5	mV	
$\|V_{OD}\|$	HS transmit differential voltage	140	200	270	mV	
$\|\Delta V_{OD}\|$	V_{OD} mismatch when output is Differential-1 or Differential-0	–	–	14	mV	
V_{OHHS}	HS output high voltage	–	–	360	mV	
Z_{OS}	Single-ended output impedance	40	50	62.5	Ω	
ΔZ_{OS}	Single-ended output impedance mismatch	–	–	10	%	
MIPI DSI LP-TX Mode						
V_{OH}	Thevenin output high level	1.1	1.2	1.3	V	
V_{OL}	Thevenin output low level	-50	–	50	mV	
Z_{OLP}	Output impedance of LP transmitter	50	–	–	Ω	1
MIPI DSI LP-RX Mode						
V_{IH}	Logic 1 input voltage	880	–	–	mV	
V_{IL}	Logic 0 input voltage, not in ULP state	–	–	550	mV	
V_{HYST}	Input hysteresis	25	–	–	mV	
V_{IHCD}	Logic 1 Contention threshold	450	–	–	mV	
V_{ILCD}	Logic 0 Contention threshold	–	–	200	mV	

NOTE: Deviates from MIPI D-PHY specification Rev 1.0, which has minimum ZOLP of 110 Ω.

19.6.2 MIPI-Camera Serial Interface (CSI) DC Specification

Table 136. MIPI HS-RX/MIPI LP-RX Minimum, Nominal, and Maximum Voltage Parameters

Symbol	Parameter	Min.	Typ.	Max.	Unit	Notes
I_{LEAK}	Pin Leakage current	-10	–	10	µA	
MIPI-CSI HS-RX Mode						
$V_{CMRX(DC)}$	Common-mode voltage HS receive mode	70	–	330	mV	1
V_{IDTH}	Differential input high threshold	–	–	70	mV	
V_{IDTL}	Differential input low threshold	-70	–	–	mV	
V_{IHHS}	Single-ended input high voltage	–	–	460	mV	
V_{ILHS}	Single-ended input low voltage	-40	–	–	mV	
$V_{TERM-EN}$	Single-ended threshold for HS termination enable	–	–	450	mV	
Z_{ID}	Differential input impedance	80	100	125	Ω	
MIPI-CSI LP-RX Mode						
V_{IH}	Logic 1 input voltage	880	–	–	mV	
V_{IL}	Logic 0 input voltage, not in ULP state	–	–	550	mV	
$V_{IL-ULPS}$	Logic 0 input voltage, ULP state	–	–	300	mV	
V_{HYST}	Input hysteresis	25	–	–	mV	

NOTE: 1. Setup/hold violation will be seen for a VCM higher than 250mv.

19.6.3 SDIO DC Specification

Table 137 provides the SDIO DC Specification, for all other DC Specifications not listed in this table, refer to Table 160.

Table 137. SDIO DC Specification

Symbol	Parameter	Min.	Typ.	Max.	Unit	Notes
V_{OH}	Output High Voltage	1.4	–	–	V	Measured at I_{OH} maximum.
I_{OH}/I_{OL}	Current at VoL/Voh	-2	–	–	mA	

19.6.4 SD Card DC Specification

Table 138 provides the SD Card DC Specification, for all other DC Specifications not listed in in this table, refer to Table 160.

Table 138. SD Card DC Specification

Symbol	Parameter	Min.	Max.	Unit
V_{REF}	I/O Voltage	SDIO_V3P3A_V1P8A_G3		
$V_{OH(3.3)}$	Output High Voltage	0.75*V_{REF}	–	V
$V_{OL(3.3)}$	Output Low Voltage	-	0.125*V_{REF}	V
$V_{IH\ (3.3)}$	Input High Voltage (3.3 V)	0.625*V_{REF}	V_{REF}+0.3	V
$V_{IL\ (3.3)}$	Input Low Voltage (3.3 V)	VSS-0.3	0.25*V_{REF}	V
$V_{OH(1.8)}$	Output High Voltage	1.40	–	V
$V_{OL(1.8)}$	Output Low Voltage	-	0.45	V
$V_{IH\ (1.8)}$	Input High Voltage (1.8 V)	1.27	2.00	V
$V_{IL\ (1.8)}$	Input Low Voltage (1.8 V)	VSS-0.3	0.58	V
I_{OH}/I_{OL}	Current at VoL/Voh	-2	2	mA
C_{LOAD}	total Load Capacitance	-	40	pF

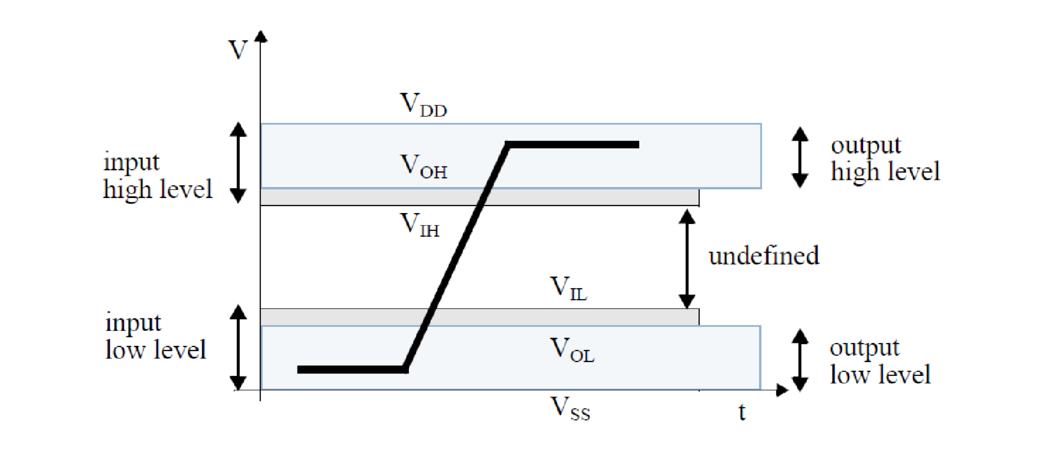
19.6.5 eMMC 4.51 DC Specification

Table 139. eMMC 4.51 DC Electrical Specifications

Symbol	Parameter	Min	Max	Units
V_{REF}	I/O Voltage	GPIOSE_V1P8A_G3		
V_{OH}	Output HIGH voltage	V_{REF} - 0.45	-	V
V_{OI}	Output LOW voltage	-	0.45	V
V_{IH}	Input HIGH voltage	0.65 * V_{REF}	V_{REF} + 0.3	V
V_{IL}	Input LOW voltage	-0.3	0.35 * V_{REF}	V
C_L	Bus Signal Line capacitance	-	30	pF
I_{IL}	Input Leakage Current	-10	10	μA
I_{0L}	Output Leakage Current	-10	10	μA

NOTE: This buffer reaches VOH/VOL with 3mA load.

Figure 37. eMMC 4.51 DC Bus Signal Level



19.6.6 JTAG DC Specification

Table 140. JTAG Signal Group DC Specification (JTAG_TCK, JTAG_TMS, JTAG_TDI, JTAG_TRST_N) (Sheet 1 of 2)

Symbol	Parameter	Min	Typ	Max	Units	Notes
V_{REF}	I/O Voltage	GPION_V1P8A_G3				
V_{IH}	Input High Voltage	0.75*V_{REF}			V	1
V_{IL}	Input Low Voltage			0.35*V_{REF}	V	2
R_{wpu}	Weak Pull Up Impedance	2.5	5	7.5	kΩ	3

Table 140. JTAG Signal Group DC Specification (JTAG_TCK, JTAG_TMS, JTAG_TDI, JTAG_TRST_N) (Sheet 2 of 2)

Symbol	Parameter	Min	Typ	Max	Units	Notes
R_{wpd}	Weak Pull Down Impedance	2.5	5	7.5	kΩ	3
$R_{wpu-20K}$	Weak Pull Up Impedance 20K	12		28	kΩ	4
$R_{wpd-40K}$	Weak Pull Down Impedance 40K	20		70	kΩ	4

NOTES:

1. V_{IH} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical high value.
2. V_{IL} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical low value.
3. Measured at GPIO_V1P8A_G3.
4. Rwpu_40k and Rwpd_40k are only used for JTAG_TRST#.
5. This buffer reaches VOH/VOL with 3mA load.

Table 141. JTAG Signal Group DC Specification (JTAG_TDO)

Symbol	Parameter	Min	Typ	Max	Units	Notes
V_{REF}	I/O Voltage	GPION_V1P8A_G3				
V_{IH}	Input High Voltage	0.75*V_{REF}			V	1
V_{IL}	Input Low Voltage			0.45*V_{REF}	V	2
Z_{pd}	Pull down Impedance	17.5		35	Ω	3
R_{wpu}	Weak Pull Up Impedance	12		28	kΩ	3
R_{wpd}	Weak Pull Down Impedance	20		70	kΩ	3

NOTES:

1. V_{IH} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical high value.
2. V_{IL} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical low value.
3. Measured at GPIO_V1P8A_G3.
4. This buffer reaches VOH/VOL with 3mA load.

Table 142. JTAG Signal Group DC Specification (JTAG_PRDY#, JTAG_PREQ#)

Symbol	Parameter	Min	Typ	Max	Units	Notes
V_{REF}	I/O Voltage	GPION_V1P8A_G3				
V_{IH}	Input High Voltage	0.75*V_{REF}			V	1
V_{IL}	Input Low Voltage			0.45*V_{REF}	V	2
Z_{pd}	Pull down Impedance	17.5		35	Ω	3
R_{wpu}	Weak Pull Up Impedance	2.5	5	7.5	kΩ	3

NOTES:

1. V_{IH} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical high value.
2. V_{IL} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical low value.
3. Measured at GPIO_V1P8A_G3.
4. This buffer reaches VOH/VOL with 3mA load.

19.6.7 DDR3L-RS Memory Controller DC Specification

Table 143. DDR3L-RS Signal Group DC Specifications

Symbol	Parameter	Min	Type	Max	Units	Notes
V_{IL}	Input Low Voltage			DDR_VREF - 200mV	V	1
V_{IH}	Input High Voltage	DDR_VREF + 200mV			V	2, 3
V_{OL}	Output Low Voltage		(DDR_VDDQG_S4 / 2)* (RON / (RON+RVTT_TERM))			3,4
V_{OH}	Output High Voltage		DDR_VDDQG_S4 - ((DDR_VDDQG_S4 / 2)* (RON/ (RON+RVTT_TERM))		V	3,4
I_{IL}	Input Leakage Current			5	μA	For all DDR Signals
R_{ON}	DDR3L-RS Clock Buffer strength	26		40	Ω	5
C_{IO}	DQ/DQS/DQS# DDR3L-RS IO Pin Capacitance		3.0		pF	

NOTES:

1. V_{IL} is defined as the maximum voltage level at the receiving agent that will be received as a logical low value. DDR_VREF is normally DDR_VDDQG_S4
2. V_{IH} is defined as the minimum voltage level at the receiving agent that will be received as a logical high value. DDR_VREF is normally DDR_VDDQG_S4
3. V_{IH} and V_{OH} may experience excursions above DDR_VDDQG_S4. However, input signal drivers must comply with the signal quality specifications.
4. RON is DDR driver resistance whereas RTT_TERM is DDR ODT resistance which is controlled by DDR.
5. DDR3L-1333 CLK buffer Ron is 26Ohm and SR target is 4V/ns; DQ-DQS buffer Ron is 30Ohms and SR target is 4V/ns; CMD/CTL buffer Ron is 20Ohms and SR target is 1.8V/ns.

19.6.8 LPDDR3 Memory Controller DC Specification

Table 144. LPDDR3 Signal Group DC Specifications

Symbol	Parameter	Min	Typ	Max	Units	Notes
DDR_VDDQG_S4	I/O Supply Voltage	1.14	1.24	1.26	V	
V_{IL}	Input Low Voltage			DDR_VREF - 200 mV	V	
V_{IH}	Input High Voltage	DDR_VREF + 200 mV			V	
V_{OL}	Output Low Voltage	-	0.260	-	V	1,2
V_{OH}	Output High Voltage	-	0.960	-	V	1,2
I_{IL}	Input Leakage Current	-	5	-	µA	3,4
R_{ON}	Clock Buffer strength	26		40	Ω	
C_{IO}	IO Pin Capacitance		3.0		pF	

NOTES:

1. V_{OL} and V_{OH} is determined with 40Ohm buffer strength setting into a 60Ohm to 0.5x V1p5_ddr test load.
2. LPDDR3-1066 CLK buffer Ron is 35Ohm and SR target is 2.5V/ns; DQ-DQS buffer Ron is 40Ohms and SR target is 2V/ns; CMD/CTL buffer Ron is 30Ohms and SR target is 1.5V/ns.
3. Applies to the pin to VCC or VSS leakage current.
4. Applies to the pin to pin leakage current.

19.6.9 USB 2.0 Host DC Specification

Table 145. USB 2.0 Host DC Specification (Sheet 1 of 3)

Symbol	Parameter	Min	Type	Max	Units	Notes
Supply Voltage						
VBUS	High-power Port	4.75		5.25	V	2
VBUS	Low-power Port	4.20		5.25	V	
Supply Current						
ICCPRT	High-power Hub Port (out)	500			mA	
ICCUPT	Low-power Hub Port (out)	100			mA	
ICCHPF	High-power Function (in)			500	mA	
ICCLPF	Low-power Function (in)			100	mA	
ICCINIT	Unconfigured Function/Hub (in)			100	mA	
ICCSH	Suspended High-power Device			2.5	mA	15
ICCSL	Suspended Low-power Device			500	µA	

Table 145. USB 2.0 Host DC Specification (Sheet 2 of 3)

Symbol	Parameter	Min	Type	Max	Units	Notes
Input Levels for Low-/full-speed						
VIH	High (driven)	2.0			V	4
VIHZ	High (floating)	2.7		3.6	V	4
VIL	Low			0.8	V	4
VDI	Differential Input Sensitivity	0.2			V	|(D+)-(D-)|;Figure ; Note 4
VCM	Differential Common Mode Range	0.8		2.5	V	Includes VDI range; Figure; Note 4
Input Levels for High-speed						
VHSSQ	High-speed squelch detection threshold (differential signal amplitude)	100		150	mV	
VHSDSC	High speed disconnect detection threshold (differential signal amplitude)	525		625	mV	
	High-speed differential input signaling levels					16
VHSCM	High-speed data signaling common mode voltage range (guideline for receiver)	-50		500	mV	
Output Levels for Low-/full-speed						
VOL	Low	0.0		0.8	V	4,5
VOH	High (Driven)	2.8		3.6	V	4,6
VOSE1	SE1	0.8			V	
VCRS	Output Signal Crossover Voltage	1.3		2.0	V	10
Output Levels for High-speed:						
VHSOI	High-speed idle level	-10		10	mV	
VHSOH	High-speed data signaling high	360		440	mV	
VHSOL	High-speed data signaling low	-10		10	mV	
VCHIRPJ	Chirp J level (differential voltage)	700		1100	mV	

Table 145. USB 2.0 Host DC Specification (Sheet 3 of 3)

Symbol	Parameter	Min	Type	Max	Units	Notes
VCHIRPK	Chirp K level (differential voltage)	-900		-500	mV	
Decoupling Capacitance:						
CHPB	Downstream Facing Port Bypass Capacitance (per hub)	120			µF	
CRPB	Upstream Facing Port Bypass Capacitance	1.0		10.0	µF	9
Input Capacitance for Low-/full-speed:						
CIND	Downstream Facing Port			150	pF	2
CINUB	Upstream Facing Port (w/o cable)			100	pF	3
CEDGE	Transceiver edge rate control capacitance			75	pF	
Input Impedance for High-speed:						
	TDR spec for high-speed termination					
Terminations:						
RPU	Bus Pull-up Resistor on Upstream Facing Port	1.425		1.575	kΩ	1.5 kΩ ±5%
RPD	Bus Pull-down Resistor on Downstream Facing Port	14.25		15.75	kΩ	1.5 kΩ ±5%
ZINP	Input impedance exclusive of pull-up/pull-down (for low-/full speed)	300			kΩ	
VTERM	Termination voltage for upstream facing port pull-up (RPU)	3.0		3.6	V	
Terminations in High-speed:						
VHSTERM	Termination voltage in high speed	-10		10	mV	
RTERM	High Speed Termination	40	45	50	Ω	
VBUSD	VBUS Voltage drop for detachable cables	-	-	1	mV	

NOTES:

1. Measured at A plug.
2. Measured at A receptacle.
3. Measured at B receptacle.
4. Measured at A or B connector.
5. Measured with RL of 1.425 kΩ to 3.6 V.

6. Measured with RL of 14.25 kΩ to GND.
7. Timing difference between the differential data signals.
8. Measured at crossover point of differential data signals.
9. The maximum load specification is the maximum effective capacitive load allowed that meets the target VBUS drop of 330 mV.
10. Excluding the first transition from the Idle state.
11. The two transitions should be a (nominal) bit time apart.
12. For both transitions of differential signaling.
13. Must accept as valid EOP.
14. Single-ended capacitance of D+ or D- is the capacitance of D+/D- to all other conductors and, if present, shield in the cable. That is, to measure the single-ended capacitance of D+, short D-, VBUS, GND, and the shield line together and measure the capacitance of D+ to the other conductors .
15. For high power devices (non-hubs) when enabled for remote wakeup.
16. Specified by eye pattern templates.

19.6.10 USB HSIC DC Specification

Table 146. USB HSIC DC Electrical Specifications

Symbol	Parameter	Min	Max	Units
V_{REF}	I/O Voltage	USBHSIC_V1P2A_G3		
V_{OH}	Output HIGH voltage	0.75 * V_{REF}	-	V
V_{Ol}	Output LOW voltage	-	0.25 * V_{REF}	V
V_{IH}	Input HIGH voltage	0.65 * V_{REF}	V_{REF} + 0.3	V
V_{IL}	Input LOW voltage	-0.3	0.35 * V_{REF}	V
O_D	I/O Pad Drive Strength	40	60	Ω
C_L	Load capacitance	1	5	pF
Z_I	I/O input impedance	240	-	kΩ
T_I	Characteristic Trace Impedance	45	55	Ω

19.6.11 USB 3.0 DC Specification

Table 147. USB 3.0 DC transmitter specifications

Symbol	Parameter	Min	Max	Units	Notes
UI	Unit Interval	199.94	200.06	ps	4
$V_{TX-DIFF-PP}$	Differential peak-peak Tx voltage swing	0.8	1.2	V	
$V_{TX-DIFF-PP-LOW}$	Low-Power Differential peak-peak Tx voltage swing	0.4	1.2	V	1
$R_{TX-DIFF-DC}$	DC differential impedance	72	92	Ω	
$V_{TX-RCV-DETECT}$	The amount of voltage change allowed during Receiver Detection		0.6	V	2

Symbol	Parameter	Min	Max	Units	Notes
$C_{AC\text{-}COUPLING}$	AC Coupling Capacitor	75	200	nF	3
$T_{CDR\text{-}SLEW\text{-}MAX}$	Maximum slew rate		10	ms/s	
$C_{TX\text{-}PARASITIC}$	Tx input capacitance for return loss	-	1.25	pf	5
	Eye Height	100	1200	mV	7,9
Dj	Deterministic Jitter	-	0.43	UI	7,8,9
Rj	Random Jitter	-	0.23	UI	6,7,8,10
Tj	Total Jitter	-	0.66	UI	7,8,9

NOTES:

1. There is no de-emphasis requirement in this mode. De-emphasis is implementation specific for this mode.
2. Detect voltage transition should be an increase in voltage on the pin looking at the detect signal to avoid a high impedance requirement when an “off” receiver's input goes below output.
3. All transmitters shall be AC coupled. The AC coupling is required either within the media or within the transmitting component itself.
4. The specified UI is equivalent to a tolerance of +-300 ppm for each device. period does not account for SSC induced variations.
5. parasitic capacitance to ground.
6. Measured over 10^6 consecutive UI and extrapolated to 10^{-12} BER.
7. Measured after receiver equalization function.
8. Measured at the end of reference channel and cables at TP1.
9. The eye height is measured at the maximum opening.
10. The Rj spec is calculated at 14.069 times the RMS random jitter for 10^{-12} BER.

Table 148. USB 3.0 DC LFPS specifications

Symbol	Parameter	Min	Max	Units	Notes
T_{PERIOD}		20	100	ns	
$V_{CM\text{-}AC\text{-}LFPS}$		-	10	mV	
$V_{CM\text{-}AC\text{-}LFPS\text{-}ACTIVE}$		-	10	mV	
$V_{TX\text{-}DIFF\text{-}PP\text{-}LFPS}$	peak-peak Differential amplitude	0.8	1.2	V	
$V_{TX\text{-}DIFF\text{-}PP\text{-}LFPS\text{-}LP}$	Low power peak-peak Differential amplitude	0.4	0.6	V	
$T_{RISEFALL2080}$		-	4	ns	
Duty cycle		40	60	%	
$C_{TX\text{-}PARASITIC}$	Tx input capacitance for return loss	-	1.25	pf	

Table 149. USB 3.0 DC Receiver specifications

Symbol	Parameter	Min	Max	Units	Notes
UI	Unit Interval	199.94	200.06	ps	1
$R_{RX\text{-}DC}$	Receiver DC common mode impedance	18	30	Ω	2
$R_{RX\text{-}DIFF\text{-}DC}$	DC differential impedance	72	120	Ω	3
$Z_{RX\text{-}HIGH\text{-}IMP\text{-}DCPOS}$	DC input CM input for V>0 during reset or power down	25	-	kΩ	4
$V_{RX\text{-}LFPS\text{-}DETDIFFp\text{-}p}$	LFPS detect threshold	100	300	mV	
f1	tolerance corner	-	4.9	MHz	
J_{RJ}	Random Jitter	-	0.0121	UI rms	1
$J_{RJP\text{-}P}$	Random Jitter peak-peak at 10^{-12}	-	0.17	UI p-p	1,4
Sj @0.5MHz	Sinusoidal Jitter	-	2	UI p-p	1,2,3
Sj @1MHz	Sinusoidal Jitter	-	1	UI p-p	1,2,3
Sj @2MHz	Sinusoidal Jitter	-	0.5	UI p-p	1,2,3
Sj @f1MHz	Sinusoidal Jitter	-	0.2	UI p-p	1,2,3
Sj @50MHz	Sinusoidal Jitter	-	0.2	UI p-p	1,2,3
V_full_swing	transition bit differential voltage swing	-	0.75	V p-p	1
V_EQ_level	Non transition bit voltage (equalization)	-	-3	db	1

NOTES:

1. All parameters are measured at TP1.
2. Due to time limitations at compliance testing, only a subset of frequencies can be tested. however, the Rx is required to tolerate Pj at all frequencies between the compliance test points.
3. During the Rx tolerance test, SSC is generated by test equipment and present all the time.
4. Random jitter is also present during the Rx tolerance test.

19.6.12 SSIC DC Specification

Table 150. SSIC DC Specification

Symbol	Parameter	Min.	Nom.	Max.	Unit
R_{REF_RT}	Reference load for when the Transmitter is terminated.		100		Ω
R_{REF_NT}	Reference load for when the Transmitter is not terminated.	10			kΩ
Z_R	Reference impedance.		100		Ω
$V_{DIFF\_DC\_LA\_RT\_TX}$	Large Amplitude differential TX DC voltage when the Transmitter is terminated. Defined for R_{REF_RT}[1] and test pattern[2]	160		240	mV
$V_{DIFF\_DC\_LA\_NT\_TX}$	Large Amplitude differential TX DC voltage when the Transmitter is not terminated. Defined for R_{REF_NT}[3] and test pattern[2]	320		480	mV
$V_{DIFF\_DC\_SA\_RT\_TX}$	Small Amplitude differential TX DC voltage when the Transmitter is terminated. Defined for R_{REF_RT}[1] and test pattern[2]	100		130	mV
$V_{DIFF\_DC\_SA\_NT\_TX}$	Small Amplitude differential TX DC voltage when the Transmitter is not terminated. Defined for R_{REF_NT}[3] and test pattern[2]	200		260	mV

Table 150. SSIC DC Specification

Symbol	Parameter	Min.	Nom.	Max.	Unit
$V_{CM\_LA\_TX}$	Large Amplitude common-mode TX voltage. Defined R_{REF_RT}[1] and test pattern[2]	160		260	mV
$V_{CM\_SA\_TX}$	Small Amplitude common-mode TX voltage. Defined R_{REF_RT}[1] and test pattern[2]	80		190	mV
C_{PIN_RX}	PIN Capacitance	-		1.5	pF

NOTES:

1. External reference load R_{REF_RT} and a reference impedance Z_{REF_RT} that conform to SRL_{REF_RT} (return loss of Z_{REF_RT}).
2. Defined when driving both a DIF-N and a DIF-P LINE state.
3. External reference load R_{REF_NT} and capacitances at TXDP and at TXDN within the limit of C_{PIN_RX}.

19.6.13 LPC DC Specification

Table 151. LPC 1.8V Signal Group DC Specification

Symbol	Parameter	Min	Typ	Max	Units	Notes
V_{IH}	Input High Voltage	1.5	1.8	1.8 +0.5	V	
V_{IL}	Input Low Voltage	-0.5	0	0.8	V	
V_{OH}	Output High Voltage	0.9 x 1.8			V	
V_{OL}	Output Low Voltage			0.1 x 1.8	V	
I_{OH}	Output High Current		0.5		mA	
I_{OL}	Output Low Current		-1.5		mA	
I_{LEAK}	Input Leakage Current	-10		10	µA	
C_{IN}	Input Capacitance			10	pF	

Table 152. LPC 3.3V Signal Group DC Specification (Sheet 1 of 2)

Symbol	Parameter	Min	Typ	Max	Units	Notes
V_{IH}	Input High Voltage	2.0	3.3	3.3 +0.5	V	1
V_{IL}	Input Low Voltage	-0.5	0	0.8	V	2
V_{OH}	Output High Voltage	2.5			V	3

Table 152. LPC 3.3V Signal Group DC Specification (Sheet 2 of 2)

Symbol	Parameter	Min	Typ	Max	Units	Notes
V_{OL}	Output Low Voltage			0.4	V	3
I_{OH}	Output High Current		0.5		mA	3
I_{OL}	Output Low Current		-1.5		mA	3
I_{LEAK}	Input Leakage Current	-10		10	μA	
C_{IN}	Input Capacitance			10	pF	

NOTES:

1. V_{IH} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical high value, Applies to LPC_AD[3:0], LPC_CLKRUN_N.
2. V_{IL} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical low value. Applies to LPC_AD[3:0], ILB_LPC_CLKRUN_N.
3. V_{OH} is tested with Iout=500uA, V_{OL} is tested with Iout=1500uA.
4. Applies to LPC_AD[3:0],LPC_CLKRUN_N and LPC_FRAME_N.
5. LPC_SERIRQ is always a 1.8V I/O irrespective of the value of LPC_V1P8V3P3_S4.

19.6.14 SPI and FST_SPI DC Specification

Table 153. SPI and FST_SPI Signal Group DC Specification

Symbol	Parameter	Min	Typ	Max	Units	Notes
V_{REF}	I/O Voltage	GPIOSE_1P8A_G3			V	3
V_{IH}	Input High Voltage	0.65 * V_{REF}			V	2
V_{IL}	Input Low Voltage	-0.5		0.35 * V_{REF}	V	2
V_{OH}	Output High Voltage	V_{REF} - 0.45		1.8V	V	1
V_{OL}	Output Low Voltage			0.45	V	1
I_{OH}	Output High Current			2	mA	1
I_{OL}	Output Low Current	-2			mA	1

NOTES:

1. Applies to SPI1_CS[1:0], SPI1_CLK, SPI1_MOSI.
2. Applies to SPI1_MISO and SPI1_MOSI.
3. The I/O buffer supply voltage is measured at the SoC package pins. The tolerances shown are inclusive of all noise from DC up to 20 MHz. In testing, the voltage rails should be measured with a bandwidth limited oscilloscope that has a rolloff of 3 dB/decade above 20 MHz.
4. This buffer reaches VOH/VOL with 3mA load.

19.6.15 Power Management/Thermal (PMC) and RTC DC Specification

Table 154. Power Management Signal Group DC Specification

Symbol	Parameter	Min	Typ	Max	Units	Notes
V_{REF}	I/O Voltage	GPIOSE_1P8A_G3			V	
V_{IH}	Input High Voltage	0.65 * V_{REF}			V	2
V_{IL}	Input Low Voltage	-0.5		0.35 * V_{REF}	V	2,3
V_{OH}	Output High Voltage	V_{REF} - 0.45		1.8V	V	1
V_{OL}	Output Low Voltage			0.45	V	1
I_{OH}	Output High Current			2	mA	1
I_{OL}	Output Low Current	-2			mA	1

NOTES:

1. The data in this table apply to signals - PMC_ACPRESENT, PMC_BATLOW_N, PMC_PLTRST_N, PMC_PWRBTN_N,PMC_SLP_S4_N, PMC_SUS_STAT_N, PMC_SUSCLK[3:0], PMC_SUSPWRDNACK
2. V_{IH} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical high value
3. V_{IL} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical low value.
4. This buffer reaches VOH/VOL with 3mA load.

Table 155. PMC_RSTBTN# 1.8V Core Well Signal Group DC Specification

Symbol	Parameter	Min	Typ	Max	Units	Notes
VREF	I/O Voltage	UNCORE_V1P8_G3			V	
V_{IH}	Input High Voltage	0.65* VREF			V	1
V_{IL}	Input Low Voltage			0.35* VREF	V	2

NOTES:

1. V_{IH} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical high value.
2. V_{IL} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical low value.

Table 156. Power Management and RTC Well Signal Group DC Specification

Symbol	Parameter	Min	Typ	Max	Units	Notes
VREF	I/O Voltage	RTC_V3P3RTC_G5				
V_{IH}	Input High Voltage	2.0	-	-	V	1
V_{IL}	Input Low Voltage	-	-	0.78	V	2

NOTES:

1. V_{IH} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical high value.
2. V_{IL} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical low value.

Table 157. RTC Well DC Specification

Symbol	Parameter	Min	Typ	Max	Units	Notes
V_{IH}	Input High Voltage	2.3	-	-	V	1
V_{IL}	Input Low Voltage	-	-	0.78	V	2

NOTES:

1. V_{IH} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical high value.
2. V_{IL} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical low value.

Table 158. PROCHOT# Signal Group DC Specification

Symbol	Parameter	Min	Type	Max	Units	Notes
V_{REF}	I/O Voltage	GPION_V1P8A_G3				
V_{IH}	Input High Voltage	0.75*V_{REF}		V_{REF}	V	1
V_{IL}	Input Low Voltage			0.45*V_{REF}	V	2
V_{OL}	Output Low Voltage			0.35 * V_{REF}	V	
I_{OL}	Output Low Current			-5	mA	

NOTES:

1. V_{IH} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical high value
2. V_{IL} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical low value.
3. This buffer reaches VOH/VOL with 3mA load.

19.6.16 SVID DC Specification

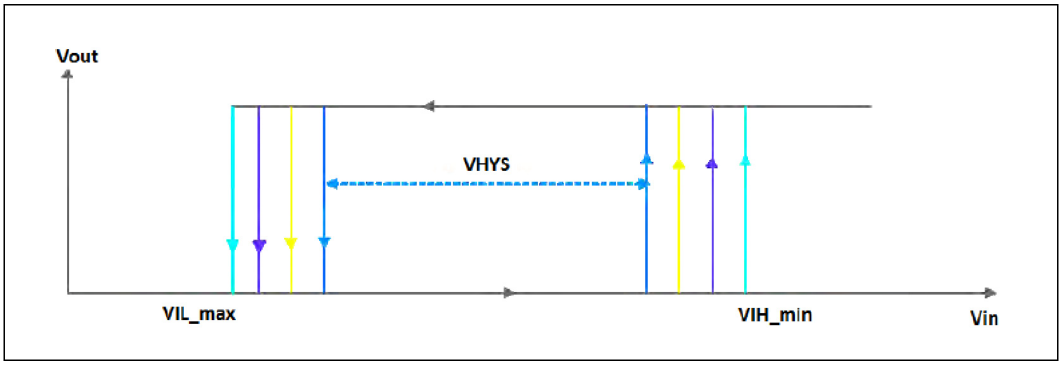
Table 159. SVID Signal Group DC Specification (SVID_DATA, SVID_CLK, SVID_ALERT_N)

Symbol	Parameter	Min	Typ	Max	Units	Notes
V_{REF}	I/O Voltage	GPION_V1P8A_G3				
V_{IH}	Input High Voltage	0.65*V_{REF}			V	1
V_{IL}	Input Low Voltage			0.35*V_{REF}	V	1
V_{OH}	Output High Voltage	V_{REF} - 0.45		V_{REF}	V	1
V_{OL}	Output Low Voltage			0.45	V	4
V_{HYS}	Hysteresis Voltage	0.1			V	
R_{ON}	BUffer on Resistance	40		60	Ω	2
I_{L}	Leakage Current	-10		10	µA	3
C_{PAD}	Pad Capacitance			9	pF	4
V_{PIN}	Pin Capacitance			10	pF	
Z_{pd}	Pull down Impedance	35	50	70	Ω	

NOTES:

1. GPIO_V1P8A_G3 refers to instantaneous voltage VSS_SENSE.
2. Measured at 0.31 * GPIO_V1P8A_G3.
3. V_{IN} between 0V and GPIO_V1P8A_G3.
4. CPAD includes die capacitance only. No package parasitic included.
5. This buffer reaches VOH/VOL with 3mA load.

Figure 38. Definition of VHYS



19.6.17 GPIO DC Specification

GPIO Buffer DC specifications.

Table 160. GPIO 1.8V Core Well Signal Group DC Specification

Symbol	Parameter	Min	Typ	Max	Units	Notes
V_{REF}	I/O Voltage	GPION_V1P8A_G3 GPIOSE_V1P8A_G3				
V_{IH}	Input High Voltage	0.65*V_{REF}			V	
V_{IL}	Input Low Voltage			0.35 * V_{REF}	V	
V_{OH}	Output High Voltage	V_{REF} - 0.45		V_{REF}	V	
V_{OL}	Output Low Voltage			0.45	V	
V_{Hys}	Input Hysteresis	0.1			V	
I_L	Leakage Current	-10		10	mA	
C_{LOAD}	Load Capacitance	2		75	pF	

NOTE: This buffer reaches VOH/VOL with 3mA load

19.6.18 SIO - I^2C DC Specification

Table 161. I^2C Signal Electrical Specifications

Symbol	Parameter	Min	Typ	Max	Units	Notes
V_{REF}	I/O Voltage	GPIOSE_V1P8A_G3			V	
V_{IH}	Input High Voltage	0.7 * V_{REF}			V	
V_{IL}	Input Low Voltage			0.3 * V_{REF}	V	
V_{OL}	Output Low Voltage			0.2 * V_{REF}	V	
V_{Hys}	Input Hysteresis	0.1			V	
C_{PIN}	Pin Capacitance	2		5	pF	

NOTE: This buffer reaches VOH/VOL with 3mA load.

19.6.19 SIO - UART DC Specification

Refer to GPIO Buffer (1.8V) DC Specification, mentioned Section 19.6.17, "GPIO DC Specification"

19.6.20 I^2S (Audio) DC Specification

Refer to the GPIO Buffer (1.8V) DC Specification, mentioned Section 19.6.17, "GPIO DC Specification"

19.6.21 PCI Express DC Specification

Table 162. PCI Express DC Receive Signal Characteristics

Symbol	Parameter	Min	Type	Max	Unit	Notes
$V_{RXDIFF\ Gen1}$	Differential RX Peak to Peak	175		1200	mV	1
$V_{RXDIFF\ Gen2}$	Differential RX Peak to Peak	100		1200	mV	1

NOTE:

1. PCI Express differential peak to peak = 2*|RXp[x] – RXn[x]|

Table 163. PCI Express DC Transmit Characteristics (Sheet 1 of 3)

Symbol	Parameter	Gen 1		Gen 2		Unit	Notes
		Min	Max	Min	Max		
UI	Unit Interval	399.88	400.12	199.94	200.06	ps	1
$V_{TX-DIFF-PP}$	Differential p-p Tx voltage swing	800	1200	800	1200	mV	
$V_{TX-DIFF-LP}$	Differential TX Peak to Peak (low power mode)	400	1200	400	1200	mV	

Table 163. PCI Express DC Transmit Characteristics (Sheet 2 of 3)

Symbol	Parameter	Gen 1		Gen 2		Unit	Notes
		Min	Max	Min	Max		
$V_{TX-DE-RATIO-3.5DB}$	Tx de-emphasis level	3	4	3	4	db	
$V_{TX-DE-RATIO-6DB}$	Tx de-emphasis level	-	-	5.5	6.5	db	
$T_{MIN-PULSE}$	Instantaneous pulse width	-	-	0.9	-	UI	
T_{TX-EYE}	Transmitter Eye including all jitter sources	0.75	-	0.75	-	UI	
$T_{TX-EYE-MEDIAN-to MAX-JITTER}$	Maximum time between the jitter median and max deviation from the median	-	0.125	-	-	UI	2
$T_{TX-HF-DJ-DD}$	Tx deterministic jitter > 1.5 MHz	-	-	-	0.15	UI	3
$T_{RF-MISMATCH}$	Tx rise/fall mismatch	-	-	-	0.1	UI	
$T_{TX-RISE-FALL}$	Transmitter rise and fall time	-	-	0.15	-	UI	
$V_{TX-CM-AC-PP}$	Tx AC peak-peak common mode voltage	-	-	-	150	mVpp	
$V_{TX-DC-CM}$	Transmitter DC common-mode voltage	0	3.6	0	3.6	V	4
$V_{TX-CM-DC-LINEDELTA}$	Absolute Delta of DC Common Mode Voltage between D+ and D-	0	25	0	25	mV	
$Z_{TX-DIFF-DC}$	DC differential Tx impedance	80	120	-	120	Ω	
$I_{TX-SHORT}$	Transmitter short-circuit current limit	-	90	-	90	mA	5

Table 163. PCI Express DC Transmit Characteristics (Sheet 3 of 3)

Symbol	Parameter	Gen 1		Gen 2		Unit	Notes
		Min	Max	Min	Max		
$V_{TX-CM-DC-ACTIVEIDLE-DELTA}$	Absolute Delta of DC Common Mode Voltage during L0 and Electrical Idle.	0	100	0	100	mV	
$V_{TX-IDLE-DIFF-AC-p}$	Electrical Idle Differential Peak Output Voltage	0	20	0	20	mV	
$V_{TX-IDLE-DIFF-D}C$	DC Electrical Idle Differential Output Voltage	-	-	0	5	mV	
T_{TX-DJ}	Tx deterministic jitter	-	-	-	57	ps	
T_{TX-RJ}	Tx Random jitter	-	-	-	3.41	ps	
$T_{TX-MEDIAN-to MAX-JITTER}$	Maximum Transmitter Medium-to-max jitter	-	77	-	-	ps	
T_{TX-TJ}	Total Jitter @ BER 1E-12	-	-	-	105	ps	

NOTES:

1. The specified UI is equivalent to a tolerance of +-300 ppm for each RefClk source. period does not account for SSC induced variations. SSC permits a +0, - 5000 ppm modulation of the clock frequency at a modulation rate not to exceed 33 kHz.
2. Measured differentially at zero crossing points after applying the 2.5 GT/s clock recovery function
3. Deterministic jitter only
4. The allowed DC common-mode voltage at a transmitter pin under any conditions.
5. The total single-ended current a transmitter can supply when shorted to ground.

Table 164. PCI Express DC Clock Request Input Signal Characteristics

Symbol	Parameter	Min	Type	Max	Unit	Notes
V_{REF}	I/O Voltage	UNCORE_V1P8_S4				
V_{IL}	Input Low Voltage			0.3*V_{REF}	V	1
V_{IH}	Input High Voltage	0.65*V_{REF}			V	1

NOTE:

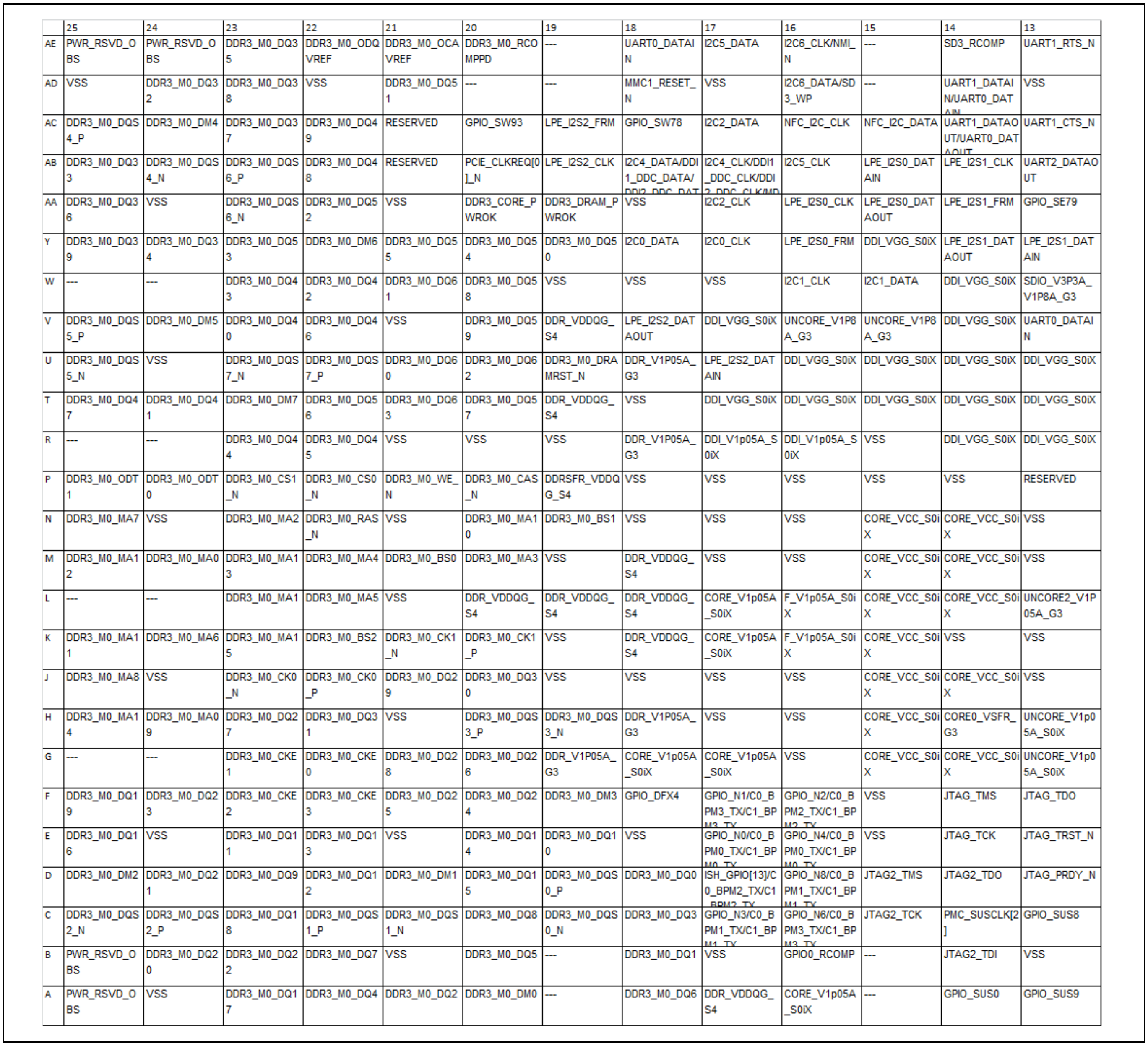
1. 3.3 V refers to UNCORE_3P3_S0 for signals in the core well. Refer Chapter 2, “Physical Interfaces” for signal and power well association.

§

20 Ballout and Ball Map

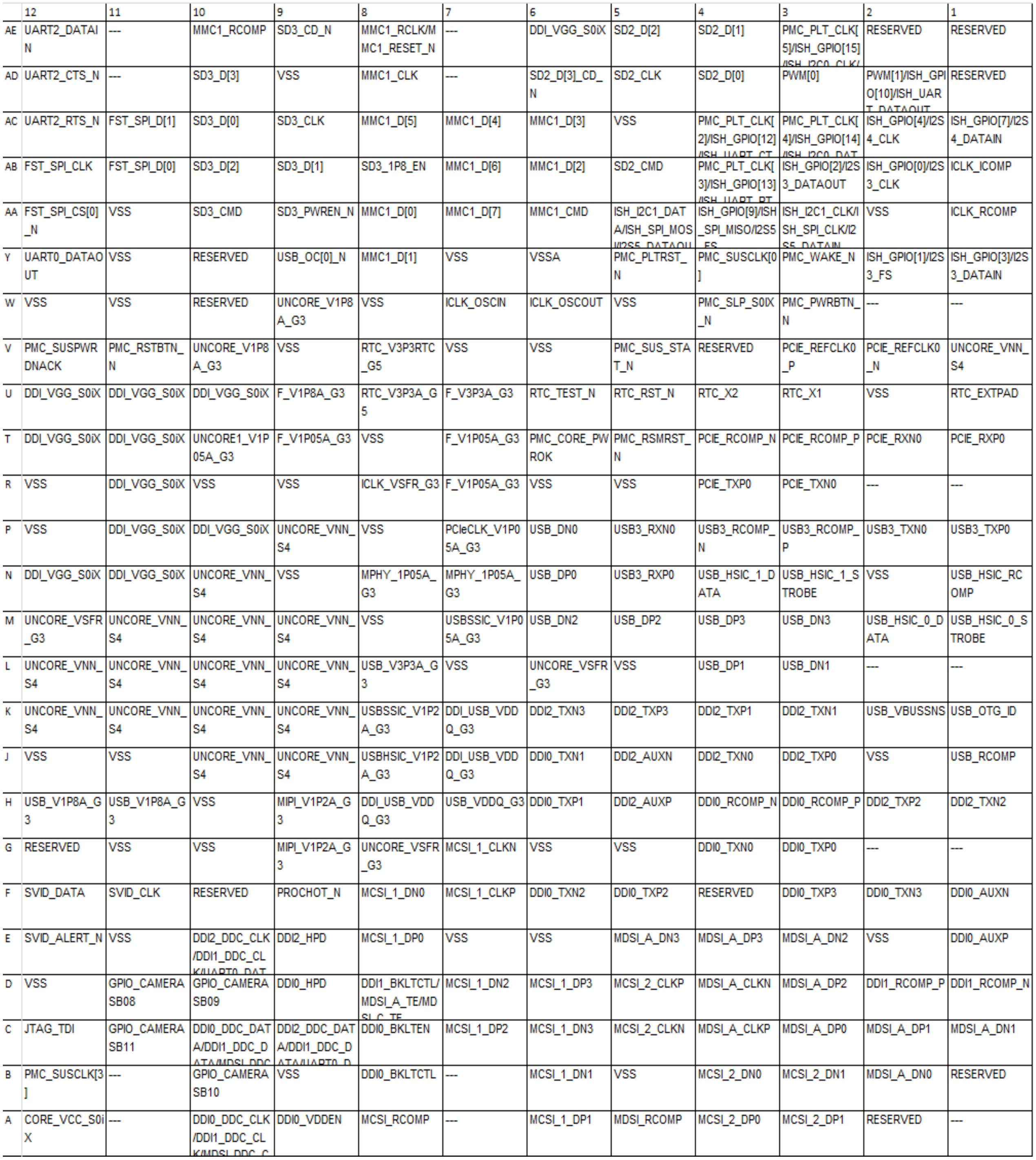
20.1 Ballout

Figure 39. Ballout - DDR3L-RS (T3) Top View Part A



	25	24	23	22	21	20	19	18	17	16	15	14	13
AE	PWR_RSVD_OBS	PWR_RSVD_OBS	DDR3_M0_DQ35	DDR3_M0_ODQVREF	DDR3_M0_OCAVREF	DDR3_M0_RCOMPPD	---	UART0_DATAIN	I2C5_DATA	I2C6_CLK/NMI_N	---	SD3_RCOMP	UART1_RTS_N
AD	VSS	DDR3_M0_DQ32	DDR3_M0_DQ38	VSS	DDR3_M0_DQ51	---	---	MMC1_RESET_N	VSS	I2C6_DATA/SD3_WP	---	UART1_DATAIN/UART0_DATAIN	VSS
AC	DDR3_M0_DQS4_P	DDR3_M0_DM4	DDR3_M0_DQ37	DDR3_M0_DQ49	RESERVED	GPIO_SW93	LPE_I2S2_FRM	GPIO_SW78	I2C2_DATA	NFC_I2C_CLK	NFC_I2C_DATA	UART1_DATAOUT/UART0_DATAOUT	UART1_CTS_N
AB	DDR3_M0_DQ33	DDR3_M0_DQS4_N	DDR3_M0_DQS6_P	DDR3_M0_DQ48	RESERVED	PCIE_CLKREQ[0]_N	LPE_I2S2_CLK	I2C4_DATA/DDI1_DDC_DATA/DDI2_DDC_DAT	I2C4_CLK/DDI1_DDC_CLK/DDI2_DDC_CLK/MD	I2C5_CLK	LPE_I2S0_DATAIN	LPE_I2S1_CLK	UART2_DATAOUT
AA	DDR3_M0_DQ36	VSS	DDR3_M0_DQS6_N	DDR3_M0_DQ52	VSS	DDR3_CORE_PWROK	DDR3_DRAM_PWROK	VSS	I2C2_CLK	LPE_I2S0_CLK	LPE_I2S0_DATAOUT	LPE_I2S1_FRM	GPIO_SE79
Y	DDR3_M0_DQ39	DDR3_M0_DQ34	DDR3_M0_DQ53	DDR3_M0_DM6	DDR3_M0_DQ55	DDR3_M0_DQ54	DDR3_M0_DQ50	I2C0_DATA	I2C0_CLK	LPE_I2S0_FRM	DDI_VGG_S0iX	LPE_I2S1_DATAOUT	LPE_I2S1_DATAIN
W	---	---	DDR3_M0_DQ43	DDR3_M0_DQ42	DDR3_M0_DQ61	DDR3_M0_DQ58	VSS	VSS	VSS	I2C1_CLK	I2C1_DATA	DDI_VGG_S0iX	SDIO_V3P3A_V1P8A_G3
V	DDR3_M0_DQS5_P	DDR3_M0_DM5	DDR3_M0_DQ40	DDR3_M0_DQ46	VSS	DDR3_M0_DQ59	DDR_VDDQG_S4	LPE_I2S2_DATAOUT	DDI_VGG_S0iX	UNCORE_V1P8A_G3	UNCORE_V1P8A_G3	DDI_VGG_S0iX	UART0_DATAIN
U	DDR3_M0_DQS5_N	VSS	DDR3_M0_DQS7_N	DDR3_M0_DQS7_P	DDR3_M0_DQ60	DDR3_M0_DQ62	DDR3_M0_DRAMRST_N	DDR_V1P05A_G3	LPE_I2S2_DATAIN	DDI_VGG_S0iX	DDI_VGG_S0iX	DDI_VGG_S0iX	DDI_VGG_S0iX
T	DDR3_M0_DQ47	DDR3_M0_DQ41	DDR3_M0_DM7	DDR3_M0_DQ56	DDR3_M0_DQ63	DDR3_M0_DQ57	DDR_VDDQG_S4	VSS	DDI_VGG_S0iX	DDI_VGG_S0iX	DDI_VGG_S0iX	DDI_VGG_S0iX	DDI_VGG_S0iX
R	---	---	DDR3_M0_DQ44	DDR3_M0_DQ45	VSS	VSS	VSS	DDR_V1P05A_G3	DDI_V1p05A_S0iX	DDI_V1p05A_S0iX	VSS	DDI_VGG_S0iX	DDI_VGG_S0iX
P	DDR3_M0_ODT1	DDR3_M0_ODT0	DDR3_M0_CS1_N	DDR3_M0_CS0_N	DDR3_M0_WE_N	DDR3_M0_CAS_N	DDRSFR_VDDQG_S4	VSS	VSS	VSS	VSS	VSS	RESERVED
N	DDR3_M0_MA7	VSS	DDR3_M0_MA2	DDR3_M0_RAS_N	VSS	DDR3_M0_MA10	DDR3_M0_BS1	VSS	VSS	VSS	CORE_VCC_S0iX	CORE_VCC_S0iX	VSS
M	DDR3_M0_MA12	DDR3_M0_MA0	DDR3_M0_MA13	DDR3_M0_MA4	DDR3_M0_BS0	DDR3_M0_MA3	VSS	DDR_VDDQG_S4	VSS	VSS	CORE_VCC_S0iX	CORE_VCC_S0iX	VSS
L	---	---	DDR3_M0_MA1	DDR3_M0_MA5	VSS	DDR_VDDQG_S4	DDR_VDDQG_S4	DDR_VDDQG_S4	CORE_V1p05A_S0iX	F_V1p05A_S0iX	CORE_VCC_S0iX	CORE_VCC_S0iX	UNCORE2_V1P05A_G3
K	DDR3_M0_MA11	DDR3_M0_MA6	DDR3_M0_MA15	DDR3_M0_BS2	DDR3_M0_CK1_N	DDR3_M0_CK1_P	VSS	DDR_VDDQG_S4	CORE_V1p05A_S0iX	F_V1p05A_S0iX	CORE_VCC_S0iX	VSS	VSS
J	DDR3_M0_MA8	VSS	DDR3_M0_CK0_N	DDR3_M0_CK0_P	DDR3_M0_DQ29	DDR3_M0_DQ30	VSS	VSS	VSS	VSS	CORE_VCC_S0iX	CORE_VCC_S0iX	VSS
H	DDR3_M0_MA14	DDR3_M0_MA09	DDR3_M0_DQ27	DDR3_M0_DQ31	VSS	DDR3_M0_DQS3_P	DDR3_M0_DQS3_N	DDR_V1P05A_G3	VSS	VSS	CORE_VCC_S0iX	CORE0_VSFR_G3	UNCORE_V1p05A_S0iX
G	---	---	DDR3_M0_CKE1	DDR3_M0_CKE0	DDR3_M0_DQ28	DDR3_M0_DQ26	DDR_V1P05A_G3	CORE_V1p05A_S0iX	CORE_V1p05A_S0iX	VSS	CORE_VCC_S0iX	CORE_VCC_S0iX	UNCORE_V1p05A_S0iX
F	DDR3_M0_DQ19	DDR3_M0_DQ23	DDR3_M0_CKE2	DDR3_M0_CKE3	DDR3_M0_DQ25	DDR3_M0_DQ24	DDR3_M0_DM3	GPIO_DFX4	GPIO_N1/C0_BPM3_TX/C1_BPM3_TX	GPIO_N2/C0_BPM2_TX/C1_BPM2_TX	VSS	JTAG_TMS	JTAG_TDO
E	DDR3_M0_DQ16	VSS	DDR3_M0_DQ11	DDR3_M0_DQ13	VSS	DDR3_M0_DQ14	DDR3_M0_DQ10	VSS	GPIO_N0/C0_BPM0_TX/C1_BPM0_TX	GPIO_N4/C0_BPM0_TX/C1_BPM0_TX	VSS	JTAG_TCK	JTAG_TRST_N
D	DDR3_M0_DM2	DDR3_M0_DQ21	DDR3_M0_DQ9	DDR3_M0_DQ12	DDR3_M0_DM1	DDR3_M0_DQ15	DDR3_M0_DQS0_P	DDR3_M0_DQ0	ISH_GPIO[13]/C0_BPM2_TX/C1_BPM2_TX	GPIO_N8/C0_BPM1_TX/C1_BPM1_TX	JTAG2_TMS	JTAG2_TDO	JTAG_PRDY_N
C	DDR3_M0_DQS2_N	DDR3_M0_DQS2_P	DDR3_M0_DQ18	DDR3_M0_DQS1_P	DDR3_M0_DQS1_N	DDR3_M0_DQ8	DDR3_M0_DQS0_N	DDR3_M0_DQ3	GPIO_N3/C0_BPM1_TX/C1_BPM1_TX	GPIO_N6/C0_BPM3_TX/C1_BPM3_TX	JTAG2_TCK	PMC_SUSCLK[2]	GPIO_SUS8
B	PWR_RSVD_OBS	DDR3_M0_DQ20	DDR3_M0_DQ22	DDR3_M0_DQ7	VSS	DDR3_M0_DQ5	---	DDR3_M0_DQ1	VSS	GPIO0_RCOMP	---	JTAG2_TDI	VSS
A	PWR_RSVD_OBS	VSS	DDR3_M0_DQ17	DDR3_M0_DQ4	DDR3_M0_DQ2	DDR3_M0_DM0	---	DDR3_M0_DQ6	DDR_VDDQG_S4	CORE_V1p05A_S0iX	---	GPIO_SUS0	GPIO_SUS9

Figure 40. Ballout - DDR3L-RS (T3) Top View Part B



	12	11	10	9	8	7	6	5	4	3	2	1
AE	UART2_DATAIN	---	MMC1_RCOMP	SD3_CD_N	MMC1_RCLK/MMC1_RESET_N	---	DDI_VGG_S0iX	SD2_D[2]	SD2_D[1]	PMC_PLT_CLK[5]/ISH_GPIO[15]/ISH_I2C0_CLK/	RESERVED	RESERVED
AD	UART2_CTS_N	---	SD3_D[3]	VSS	MMC1_CLK	---	SD2_D[3]_CD_N	SD2_CLK	SD2_D[0]	PWM[0]	PWM[1]/ISH_GPIO[10]/ISH_UART_DATAOUT	RESERVED
AC	UART2_RTS_N	FST_SPI_D[1]	SD3_D[0]	SD3_CLK	MMC1_D[5]	MMC1_D[4]	MMC1_D[3]	VSS	PMC_PLT_CLK[2]/ISH_GPIO[12]/ISH_UART_CT	PMC_PLT_CLK[4]/ISH_GPIO[14]/ISH_I2C0_DAT	ISH_GPIO[4]/I2S4_CLK	ISH_GPIO[7]/I2S4_DATAIN
AB	FST_SPI_CLK	FST_SPI_D[0]	SD3_D[2]	SD3_D[1]	SD3_1P8_EN	MMC1_D[6]	MMC1_D[2]	SD2_CMD	PMC_PLT_CLK[3]/ISH_GPIO[13]/ISH_UART_RT	ISH_GPIO[2]/I2S3_DATAOUT	ISH_GPIO[0]/I2S3_CLK	ICLK_ICOMP
AA	FST_SPI_CS[0]_N	VSS	SD3_CMD	SD3_PWREN_N	MMC1_D[0]	MMC1_D[7]	MMC1_CMD	ISH_I2C1_DATA/ISH_SPI_MOSI/I2S5_DATAOU	ISH_GPIO[9]/ISH_SPI_MISO/I2S5_FS	ISH_I2C1_CLK/ISH_SPI_CLK/I2S5_DATAIN	VSS	ICLK_RCOMP
Y	UART0_DATAOUT	VSS	RESERVED	USB_OC[0]_N	MMC1_D[1]	VSS	VSSA	PMC_PLTRST_N	PMC_SUSCLK[0]	PMC_WAKE_N	ISH_GPIO[1]/I2S3_FS	ISH_GPIO[3]/I2S3_DATAIN
W	VSS	VSS	RESERVED	UNCORE_V1P8A_G3	VSS	ICLK_OSCIN	ICLK_OSCOUT	VSS	PMC_SLP_S0IX_N	PMC_PWRBTN_N	---	---
V	PMC_SUSPWRDNACK	PMC_RSTBTN_N	UNCORE_V1P8A_G3	VSS	RTC_V3P3RTC_G5	VSS	VSS	PMC_SUS_STAT_N	RESERVED	PCIE_REFCLK0_P	PCIE_REFCLK0_N	UNCORE_VNN_S4
U	DDI_VGG_S0iX	DDI_VGG_S0iX	DDI_VGG_S0iX	F_V1P8A_G3	RTC_V3P3A_G5	F_V3P3A_G3	RTC_TEST_N	RTC_RST_N	RTC_X2	RTC_X1	VSS	RTC_EXTPAD
T	DDI_VGG_S0iX	DDI_VGG_S0iX	UNCORE1_V1P05A_G3	F_V1P05A_G3	VSS	F_V1P05A_G3	PMC_CORE_PWROK	PMC_RSMRST_N	PCIE_RCOMP_N	PCIE_RCOMP_P	PCIE_RXN0	PCIE_RXP0
R	VSS	DDI_VGG_S0iX	VSS	VSS	ICLK_VSFR_G3	F_V1P05A_G3	VSS	VSS	PCIE_TXP0	PCIE_TXN0	---	---
P	VSS	DDI_VGG_S0iX	DDI_VGG_S0iX	UNCORE_VNN_S4	VSS	PCIeCLK_V1P05A_G3	USB_DN0	USB3_RXN0	USB3_RCOMP_N	USB3_RCOMP_P	USB3_TXN0	USB3_TXP0
N	DDI_VGG_S0iX	DDI_VGG_S0iX	UNCORE_VNN_S4	VSS	MPHY_1P05A_G3	MPHY_1P05A_G3	USB_DP0	USB3_RXP0	USB_HSIC_1_DATA	USB_HSIC_1_STROBE	VSS	USB_HSIC_RCOMP
M	UNCORE_VSFR_G3	UNCORE_VNN_S4	UNCORE_VNN_S4	UNCORE_VNN_S4	VSS	USBSSIC_V1P05A_G3	USB_DN2	USB_DP2	USB_DP3	USB_DN3	USB_HSIC_0_DATA	USB_HSIC_0_STROBE
L	UNCORE_VNN_S4	UNCORE_VNN_S4	UNCORE_VNN_S4	UNCORE_VNN_S4	USB_V3P3A_G3	VSS	UNCORE_VSFR_G3	VSS	USB_DP1	USB_DN1	---	---
K	UNCORE_VNN_S4	UNCORE_VNN_S4	UNCORE_VNN_S4	UNCORE_VNN_S4	USBSSIC_V1P2A_G3	DDI_USB_VDDQ_G3	DDI2_TXN3	DDI2_TXP3	DDI2_TXP1	DDI2_TXN1	USB_VBUSSNS	USB_OTG_ID
J	VSS	VSS	UNCORE_VNN_S4	UNCORE_VNN_S4	USBHSIC_V1P2A_G3	DDI_USB_VDDQ_G3	DDI0_TXN1	DDI2_AUXN	DDI2_TXN0	DDI2_TXP0	VSS	USB_RCOMP
H	USB_V1P8A_G3	USB_V1P8A_G3	VSS	MIPI_V1P2A_G3	DDI_USB_VDDQ_G3	USB_VDDQ_G3	DDI0_TXP1	DDI2_AUXP	DDI0_RCOMP_N	DDI0_RCOMP_P	DDI2_TXP2	DDI2_TXN2
G	RESERVED	VSS	VSS	MIPI_V1P2A_G3	UNCORE_VSFR_G3	MCSI_1_CLKN	VSS	VSS	DDI0_TXN0	DDI0_TXP0	---	---
F	SVID_DATA	SVID_CLK	RESERVED	PROCHOT_N	MCSI_1_DN0	MCSI_1_CLKP	DDI0_TXN2	DDI0_TXP2	RESERVED	DDI0_TXP3	DDI0_TXN3	DDI0_AUXN
E	SVID_ALERT_N	VSS	DDI2_DDC_CLK/DDI1_DDC_CLK/UART0_DAT	DDI2_HPD	MCSI_1_DP0	VSS	VSS	MDSI_A_DN3	MDSI_A_DP3	MDSI_A_DN2	VSS	DDI0_AUXP
D	VSS	GPIO_CAMERASB08	GPIO_CAMERASB09	DDI0_HPD	DDI1_BKLTCTL/MDSI_A_TE/MDSI_C_TE	MCSI_1_DN2	MCSI_1_DP3	MCSI_2_CLKP	MDSI_A_CLKN	MDSI_A_DP2	DDI1_RCOMP_P	DDI1_RCOMP_N
C	JTAG_TDI	GPIO_CAMERASB11	DDI0_DDC_DATA/DDI1_DDC_DATA/MDSI_DDC	DDI2_DDC_DATA/DDI1_DDC_DATA/UART0_D	DDI0_BKLTEN	MCSI_1_DP2	MCSI_1_DN3	MCSI_2_CLKN	MDSI_A_CLKP	MDSI_A_DP0	MDSI_A_DP1	MDSI_A_DN1
B	PMC_SUSCLK[3]	---	GPIO_CAMERASB10	VSS	DDI0_BKLTCTL	---	MCSI_1_DN1	VSS	MCSI_2_DN0	MCSI_2_DN1	MDSI_A_DN0	RESERVED
A	CORE_VCC_S0iX	---	DDI0_DDC_CLK/DDI1_DDC_CLK/MDSI_DDC_C	DDI0_VDDEN	MCSI_RCOMP	---	MCSI_1_DP1	MDSI_RCOMP	MCSI_2_DP0	MCSI_2_DP1	RESERVED	---

Figure 41. Ballout LPPDR3 (T4) Top View Part A

	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41
BW			VSS		VSS		VSS		VSS		VSS		VSS		VSS		DDR_V1P05A_G3		DDR_V1P05A_G3
BV		VSS		LPDDR3_M1_CS[1]_N		LPDDR3_M1_ODT_B		VSS		LPDDR3_M1_DQS[2]_P_B		VSS		LPDDR3_M1_DQS[3]_P_B		VSS		LPDDR3_M1_DQ[30]_B	
BU	VSS		LPDDR3_M1_CKE[1]_B		LPDDR3_M1_ODT_A		LPDDR3_M1_DQ[17]_B		LPDDR3_M1_DQS[2]_N_B		LPDDR3_M1_DQ[22]_B		LPDDR3_M1_DQS[3]_N_B		LPDDR3_M1_DQ[24]_B		LPDDR3_M1_DQ[25]_B		VSS
BT		LPDDR3_M1_CKE[1]_A		VSS		VSS		LPDDR3_M1_DQ[20]_B		VSS		LPDDR3_M1_DQ[21]_B		VSS		LPDDR3_M1_DQ[26]_B		VSS	
BR	VSS		RESERVED		RESERVED		VSS		LPDDR3_M1_DQ[18]_B		LPDDR3_M1_DM[2]_B		LPDDR3_M1_DQ[23]_B		LPDDR3_M1_DQ[27]_B		LPDDR3_M1_DQ[29]_B		VSS
BP		VSS		RESERVED		RESERVED		LPDDR3_M1_DQ[16]_B		LPDDR3_M1_DQ[19]_B		VSS		LPDDR3_M1_DM[3]_B		LPDDR3_M1_DQ[31]_B		LPDDR3_M1_DQ[28]_B	
BN	DDR_VDDQG_S4		RESERVED		RESERVED		VSS		LPDDR3_M1_DQS[0]_N_B		LPDDR3_M1_DQ[05]_B		LPDDR3_M1_DM[0]_B		VSS		LPDDR3_M1_DQ[12]_B		VSS
BM		LPDDR3_M1_CKE[0]_B		VSS		RESERVED		LPDDR3_M1_DQ[04]_B		LPDDR3_M1_DQS[0]_P_B		LPDDR3_M1_DQ[06]_B		VSS		VSS		LPDDR3_M1_DQ[13]_B	
BL	DDR_VDDQG_S4		LPDDR3_M1_CS[0]_N		RESERVED		LPDDR3_M1_DQ[02]_B		LPDDR3_M1_DQ[00]_B		LPDDR3_M1_DQ[01]_B		LPDDR3_M1_DQS[1]_N_B		LPDDR3_M1_DQ[08]_B		LPDDR3_M1_DQ[11]_B		VSS
BK		VSS		LPDDR3_M1_CKE[0]_A		VSS		LPDDR3_M1_DQ[03]_B		VSS		VSS		LPDDR3_M1_DQS[1]_P_B		LPDDR3_M1_DQ[10]_B		LPDDR3_M1_DQ[15]_B	
BJ	DDR_VDDQG_S4		LPDDR3_M1_CA[07]		LPDDR3_M1_CA[05]		RESERVED		LPDDR3_M1_CA[03]		LPDDR3_M1_CA[00]		LPDDR3_M1_DQ[07]_B		VSS		LPDDR3_M1_DQ[14]_B		VSS
BH		VSS		VSS		LPDDR3_M1_CA[04]		VSS		LPDDR3_M1_CA[01]		RESERVED		LPDDR3_M1_DM[1]_B		LPDDR3_M1_DQ[09]_B		VSS	
BG	DDR_VDDQG_S4		LPDDR3_M1_CK_P_B		LPDDR3_M1_CK_N_B		VSS		LPDDR3_M1_CA[06]		VSS								
BF		VSS		LPDDR3_M1_CK_P_A		LPDDR3_M1_CK_N_A		VSS		RESERVED		LPDDR3_M1_CA[02]		VSS		VSS		VSS	
BE	VSS		VSS		LPDDR3_M1_DQ[13]_A		LPDDR3_M1_CA[09]		RESERVED		RESERVED								
BD		LPDDR3_M1_DQ[12]		VSS		LPDDR3_M1_DQ[15]_A		LPDDR3_M1_CA[08]		LPDDR3_M1_ODQVREF		LPDDR3_M1_OCAVREF			VSS		VSS		VSS
BC	VSS		LPDDR3_M1_DQ[14]_A		LPDDR3_M1_DM[1]_A		VSS		VSS		VSS								
BB		VSS		LPDDR3_M1_DQS[1]_N_A		VSS		VSS		LPDDR3_M1_DQ[28]_A		LPDDR3_M1_DQ[30]_A		VSS		VSS		VSS	
BA	DDR_VDDQG_S4		LPDDR3_M1_DQ[10]_A		LPDDR3_M1_DQS[1]_P_A		VSS		LPDDR3_M1_DQS[3]_N_A		LPDDR3_M1_DQ[31]_A				VSS		VSS		VSS
AY		LPDDR3_M1_DQ[09]_A		VSS		LPDDR3_M1_DQ[11]_A		LPDDR3_M1_DQ[25]_A		LPDDR3_M1_DQS[3]_P_A		LPDDR3_M1_DQ[27]_A							
AW	DDR_VDDQG_S4		LPDDR3_M1_DQ[07]_A		LPDDR3_M1_DQ[08]_A		LPDDR3_M1_DQ[26]_A		LPDDR3_M1_DQ[29]_A		VSS			VSS		VSS		DDR_V1P05A_G3	
AV		VSS		LPDDR3_M1_DQ[06]_A		LPDDR3_M1_DQ[05]_A		LPDDR3_M1_DQ[24]_A		LPDDR3_M1_DM[3]_A		LPDDR3_M1_DM[2]_A							
AU	DDR_VDDQG_S4		LPDDR3_M1_DQ[02]_A		LPDDR3_M1_DQ[01]_A		VSS		VSS		VSS				VSS		VSS		DDRSFRCH1_VDDQG_S4
AT		LPDDR3_M1_DQS[0]_N_A		VSS		VSS		VSS		LPDDR3_M1_DQ[21]_A		LPDDR3_M1_DQ[22]_A		VSS		VSS		DDRCH1_VDDQG_S4	
AR	DDR_VDDQG_S4		LPDDR3_M1_DQS[0]_P_A		LPDDR3_M1_DQ[03]_A		LPDDR3_M1_DQ[20]_A		LPDDR3_M1_DQS[2]_N_A		LPDDR3_M1_DQ[17]_A								
AP		VSS		LPDDR3_M1_DM[0]_A		VSS		LPDDR3_M1_DQ[16]_A		LPDDR3_M1_DQS[2]_P_A		LPDDR3_M1_DQ[18]_A			RESERVED		LPDDR3_M1_RCOMPPD		VSS
AN	VSS		LPDDR3_M1_DQ[00]_A		LPDDR3_M1_DQ[04]_A		VSS		VSS		VSS								
AM		VSS		VSS		VSS		VSS		LPDDR3_M1_DQ[23]_A		LPDDR3_M1_DQ[19]_A		RESERVED		VSS		VSS	
AL																			
AK	VSS		VSS		VSS		VSS		LPDDR3_CORE_PWROK		LPDDR3_DRAM_PWROK				VSS		VSS		RESERVED
AJ																			
AH		VSS		VSS		VSS		VSS		LPDDR3_M0_DQ[24]_B		LPDDR3_M0_DQ[28]_B		RESERVED		VSS		VSS	
AG	VSS		LPDDR3_M0_DQ[15]_B		LPDDR3_M0_DQ[11]_B		VSS		VSS		VSS								
AF		VSS		LPDDR3_M0_DM[1]_B		VSS		LPDDR3_M0_DQ[31]_B		LPDDR3_M0_DQS[3]_P_B		LPDDR3_M0_DQ[29]_B			RESERVED		VSS		CORE_VSS1_SENSE
AE	DDR_VDDQG_S4		LPDDR3_M0_DQS[1]_P_B		LPDDR3_M0_DQ[12]_B		LPDDR3_M0_DQ[27]_B		LPDDR3_M0_DQS[3]_N_B		LPDDR3_M0_DQ[30]_B								
AD		LPDDR3_M0_DQS[1]_N_B		VSS		VSS		VSS		LPDDR3_M0_DQ[26]_B		LPDDR3_M0_DQ[25]_B		VSS		LPDDR3_M0_RCOMPPD		DDRCH0_VDDQG_S4	
AC	DDR_VDDQG_S4		LPDDR3_M0_DQ[13]_B		LPDDR3_M0_DQ[14]_B		VSS		VSS		VSS				VSS		VSS		DDRSFRCH0_VDDQG_S4
AB		VSS		LPDDR3_M0_DQ[09]_B		LPDDR3_M0_DQ[10]_B		LPDDR3_M0_DQ[23]_B		LPDDR3_M0_DM[02]_B		LPDDR3_M0_DM[03]_B							
AA	DDR_VDDQG_S4		LPDDR3_M0_DQ[08]_B		LPDDR3_M0_DQ[07]_B		LPDDR3_M0_DQ[21]_B		LPDDR3_M0_DQ[18]_B		VSS			VSS		VSS		DDR_V1P05A_G3	
Y		LPDDR3_M0_DQ[06]_B		VSS		LPDDR3_M0_DQ[04]_B		LPDDR3_M0_DQ[22]_B		LPDDR3_M0_DQS[2]_P_B		LPDDR3_M0_DQ[20]_B							
W	DDR_VDDQG_S4		LPDDR3_M0_DQ[05]_B		LPDDR3_M0_DQS[0]_P_B		VSS		LPDDR3_M0_DQS[2]_N_B		LPDDR3_M0_DQ[16]_B				VSS		VSS		VSS
V		VSS		LPDDR3_M0_DQS[0]_N_B		VSS		VSS		LPDDR3_M0_DQ[19]_B		LPDDR3_M0_DQ[17]_B		VSS		VSS		VSS	
U	VSS		LPDDR3_M0_DQ[01]_B		LPDDR3_M0_DM[0]_B		VSS		VSS		VSS								
T		LPDDR3_M0_DQ[03]_B		VSS		LPDDR3_M0_DQ[00]_B		LPDDR3_M0_CA[8]		LPDDR3_M0_ODQVREF		LPDDR3_M0_OCAVREF			VSS		VSS		VSS
R	VSS		VSS		LPDDR3_M0_DQ[02]_B		LPDDR3_M0_CA[9]		RESERVED		RESERVED								
P		VSS		LPDDR3_M0_CK_P_B		LPDDR3_M0_CK_N_B		VSS		RESERVED		LPDDR3_M0_CA[2]		VSS		VSS		VSS	
N	DDR_VDDQG_S4		LPDDR3_M0_CK_P_A		LPDDR3_M0_CK_N_A		VSS		LPDDR3_M0_CA[6]		VSS								
M		VSS		VSS		LPDDR3_M0_CA[4]		VSS		LPDDR3_M0_CA[1]		RESERVED		LPDDR3_M0_DM[0]_A		LPDDR3_M0_DQ[06]_A		VSS	
L	DDR_VDDQG_S4		LPDDR3_M0_CA[7]		LPDDR3_M0_CA[5]		RESERVED		LPDDR3_M0_CA[3]		LPDDR3_M0_CA[0]		LPDDR3_M0_DQ[08]_A		VSS		LPDDR3_M0_DQ[01]_A		VSS
K		VSS		LPDDR3_M0_CKE[0]_B		VSS		LPDDR3_M0_DQ[12]_A		VSS		VSS		LPDDR3_M0_DQS[0]_P_A		LPDDR3_M0_DQ[05]_A		LPDDR3_M0_DQ[00]_A	
J	DDR_VDDQG_S4		LPDDR3_M0_CS[0]_N		RESERVED		LPDDR3_M0_DQ[13]_A		LPDDR3_M0_DQ[15]_A		LPDDR3_M0_DQ[14]_A		LPDDR3_M0_DQS[0]_N_A		LPDDR3_M0_DQ[07]_A		LPDDR3_M0_DQ[04]_A		VSS
H		LPDDR3_M0_CKE[0]_A		VSS		RESERVED		LPDDR3_M0_DQ[11]_A		LPDDR3_M0_DQS[1]_P_A		LPDDR3_M0_DQ[09]_A		VSS		VSS		LPDDR3_M0_DQ[02]_A	
G	DDR_VDDQG_S4		RESERVED		RESERVED		VSS		LPDDR3_M0_DQS[1]_N_A		LPDDR3_M0_DQ[10]_A		LPDDR3_M0_DM[1]_A		VSS		LPDDR3_M0_DQ[03]_A		VSS
F		VSS		RESERVED		RESERVED		LPDDR3_M0_DQ[31]_A		LPDDR3_M0_DQ[28]_A		VSS		LPDDR3_M0_DM[2]_A		LPDDR3_M0_DQ[16]_A		LPDDR3_M0_DQ[19]_A	
E	VSS		RESERVED		RESERVED		VSS		LPDDR3_M0_DQ[29]_A		LPDDR3_M0_DM[03]_A		LPDDR3_M0_DQ[24]_A		LPDDR3_M0_DQ[20]_A		LPDDR3_M0_DQ[18]_A		VSS
D		LPDDR3_M0_CKE[1]_B		VSS		VSS		LPDDR3_M0_DQ[27]_A		VSS		LPDDR3_M0_DQ[26]_A		VSS		LPDDR3_M0_DQ[21]_A		VSS	
C	VSS		LPDDR3_M0_CKE[1]_A		LPDDR3_M0_ODT_B		LPDDR3_M0_DQ[30]_A		LPDDR3_M0_DQS[3]_N_A		LPDDR3_M0_DQ[25]_A		LPDDR3_M0_DQS[2]_N_A		LPDDR3_M0_DQ[23]_A		LPDDR3_M0_DQ[22]_A		VSS
B		VSS		LPDDR3_M0_CS[1]_N		LPDDR3_M0_ODT_A		VSS		LPDDR3_M0_DQS[3]_P_A		VSS		LPDDR3_M0_DQS[2]_P_A		VSS		LPDDR3_M0_DQ[17]_A	
A			VSS		VSS		VSS		VSS		VSS		VSS		VSS		DDR_V1P05A_G3		DDR_V1P05A_G3

Figure 42. Ballout LPPDR3 (T4) Top View Part B

	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21
BW		VSS		DDI_V1P15_S0iX		DDI_V1P15_S0iX		UNCORE_V1P8A_G3			VSS			GPIOSE_V1P8A_G3		GPIOSE_V1P8A_G3		GPIOSE_V1P8A_G3		GPIOSE_V1P8A_G3
BV	DDR_V1P05A_G3		MMC1_RESET_N/SPI3_CS[1]		DDI_V1P15_S0iX		I2C3_CLK		UNCORE_V1P8A_G3				LPE_I2S1_FRM		GPIOSE_V1P8A_G3		FST_SPI_CS[2]_N		GPIOSE_V1P8A_G3	
BU		PCIE_CLKREQ[1]_N		GPIO_SW78		I2C0_DATA		I2C3_DATA			LPE_I2S0_DATAIN			UART1_DATAOUT/UART0_DATAOUT		FST_SPI_CS[0]_N		SD3_PWREN_N		USB_OC[1]_N
BT	SD3_WP		VSS		I2C0_CLK		VSS		I2C4_DATA/DDI2_DDC_DATA/DDI2_DD				VSS		UART1_DATAIN/UART0_DATAIN		VSS		SD3_1P8_EN	
BR		PCIE_CLKREQ[0]_N		ISH_GPIO[12]/ISH_UART_CTS_N		I2C1_DATA		I2C5_DATA			LPE_I2S1_CLK			UART1_CTS_N		UART1_RTS_N		FST_SPI_CS[1]_N		LPC_AD[3]/ISH_GPIO[15]/ISH_I2C0_CLK/
BP	VSS		GPIO_SW93		I2C1_CLK		I2C4_CLK/DDI0_DDC_CLK/DDI2_DDC_CLK		VSS				LPE_I2S1_DATAOUT		VSS		FST_SPI_CLK		USB_OC[0]_N	
BN		VSS		SPI3_MISO		VSS		NFC_I2C_DATA			VSS			VSS		FST_SPI_D[0]		VSS		VSS
BM	VSS		SPI3_CS[0]_N		VSS		VSS		LPE_I2S0_DATAOUT				VSS		VSS		LPC_SERIRQ/SPI2_CS[0]_N		VSS	
BL		LPE_I2S2_DATAIN		UART0_DATAOUT/SPI3_CLK		I2C2_CLK		I2C6_DATA/SD3_WP			VSS			UART2_CTS_N		FST_SPI_D[3]		VSS		LPC_CLKRUN_N/UART0_DATAOUT/SPI2
BK	LPE_I2S2_DATAOUT		LPE_I2S2_CLK		SPI3_MOSI		I2C5_CLK		LPE_I2S0_FRM				UART2_DATAOUT		FST_SPI_D[1]		VSS		PMC_SUSPWRDNACK	
BJ		VSS		UART0_DATAIN		VSS		NFC_I2C_CLK			LPE_I2S1_DATAIN			VSS		FST_SPI_D[2]		VSS		GPIO_ALERT/ISH_GPIO[11]/ISH_UART
BH	LPE_I2S2_FRM		VSS		I2C2_DATA		I2C6_CLK/NMI_N		LPE_I2S0_CLK				UART2_DATAIN		UART2_RTS_N		PMC_RSTBTN_N		LPC_RCOMP	
BG																				
BF	VSS		PWR_RSVD_OBS		VSS		VSS		VSS			GPIOSE_V1P8A_G3		VSS		LPC_V3P3A_V1P8A_S4		SDIO_V3P3A_V1P8A_G3		VSS
BE																				
BD		DDI_VGG_S0iX		DDI_VGG_S0iX		DDI_VGG_S0iX		DDI_VGG_S0iX		GPIOSE_V1P8A_G3			GPIOSE_V1P8A_G3		DDI_VGG_S0iX		VSS		DDI_VGG_S0iX	
BC																				
BB	DDI_VGG_S0iX		DDI_VGG_S0iX		DDI_VGG_S0iX		DDI_VGG_S0iX		DDI_VGG_S0iX			DDI_VGG_S0iX		DDI_VGG_S0iX		DDI_VGG_S0iX		DDI_VGG_S0iX		UNCORE_VNN_S4
BA		DDI_VGG_S0iX		VSS		VSS		DDI_VGG_S0iX		DDI_VGG_S0iX			VSS		VSS		DDI_VSS_SENSE		DDI_VGG_S0iX	
AY																				
AW	DDI_VGG_S0iX		DDI_VGG_S0iX		DDI_VGG_S0iX		DDI_VGG_S0iX		DDI_VGG_S0iX			DDI_VGG_S0iX		DDI_VGG_S0iX		DDI_VGG_SENSE		DDI_VGG_S0iX		UNCORE_VNN_S4
AV																				
AU		DDI_VGG_S0iX		DDI_VGG_S0iX		DDI_VGG_S0iX		DDI_VGG_S0iX		DDI_VGG_S0iX			DDI_VGG_S0iX		DDI_VGG_S0iX		DDI_VGG_S0iX		UNCORE_VNN_S4	
AT	DDI_VGG_S0iX		DDI_VGG_S0iX		VSS		VSS		PWR_RSVD_OBS			DDI_VGG_S0iX		VSS		VSS		DDI_VGG_S0iX		UNCORE_VNN_S4
AR																				
AP		DDI_VGG_S0iX		DDI_VGG_S0iX		DDI_VGG_S0iX		DDI_VGG_S0iX		DDI_VGG_S0iX			DDI_VGG_S0iX		DDI_VGG_S0iX		DDI_VGG_S0iX		DDI_VGG_S0iX	
AN																				
AM	DDI_VGG_S0iX		DDI_VGG_S0iX		DDI_VGG_S0iX		PWR_RSVD_OBS		VSS			VSS		DDI_VGG_S0iX		DDI_VGG_S0iX		VSS		DDI_VGG_S0iX
AL																				
AK		RESERVED		CORE_VCC1_S0iX		CORE_VCC1_S0iX		CORE_VCC1_S0iX		CORE1_VSFR_G3			CORE1_VSFR_G3		VSS		DDI_VGG_S0iX		DDI_VGG_S0iX	
AJ																				
AH	VSS		CORE_VCC1_S0iX		CORE_VCC1_S0iX		CORE_VCC1_S0iX		CORE_VCC1_S0iX			VSS		RESERVED		UNCORE_VNN_S4		UNCORE_VNN_S4		UNCORE_VNN_S4
AG																				
AF		CORE_VCC1_SENSE		CORE_VCC1_S0iX		CORE_VCC1_S0iX		CORE_VCC1_S0iX		VSS			RESERVED		UNCORE2_V1P05A_G3		UNCORE2_V1P05A_G3		UNCORE_VNN_S4	
AE																				
AD	F_V1P15_S0iX		CORE_VCC1_S0iX		CORE_VCC1_S0iX		CORE_VCC1_S0iX		CORE_VCC1_S0iX			UNCORE_VSFR_G3		UNCORE_VNN_S4		UNCORE_VNN_S4		VSS		UNCORE_VNN_S4
AC		VSS		VSS		VSS		UNCORE_VNN_S4		UNCORE_VNN_S4			RESERVED		UNCORE_VNN_SENSE		UNCORE_VSS_SENSE		VSS	
AB																				
AA	VSS		VSS		UNCORE_VNN_S4		UNCORE_VNN_S4		VSS			UNCORE_VNN_S4		UNCORE_VNN_S4		UNCORE_V1P15_S0iX		UNCORE_VNN_S4		UNCORE_VNN_S4
Y																				
W		F_V1P15_S0iX		CORE_VCC0_S0iX		CORE_VCC0_S0iX		CORE_VCC0_S0iX		CORE0_VSFR_G3			UNCORE_VNN_S4		UNCORE_VNN_S4		UNCORE_V1P15_S0iX		UNCORE_VNN_S4	
V	CORE_VCC0_SENSE		CORE_VSS0_SENSE		CORE_VCC0_S0iX		CORE_VCC0_S0iX		CORE_VCC0_S0iX			VSS		PWR_RSVD_OBS		UNCORE_VNN_S4		UNCORE_VNN_S4		UNCORE_VNN_S4
U																				
T		VSS		CORE_VCC0_S0iX		CORE_VCC0_S0iX		CORE_VCC0_S0iX		CORE_VCC0_S0iX			VSS		VSS		VSS		VSS	
R																				
P	VSS		CORE_VCC0_S0iX		CORE_VCC0_S0iX		CORE_VCC0_S0iX		CORE_VCC0_S0iX			VSS		CORE_VCC1_S0iX		CORE_VCC1_S0iX		VSS		VSS
N																				
M	GPIO0_RCOMP		VSS		VSS		VSS		VSS				CORE_VCC1_S0iX		CORE_VCC1_S0iX		CORE_VCC1_S0iX		VSS	
L		VSS		PWR_RSVD_OBS		VSS		VSS			VSS			VSS		CORE_VCC1_S0iX		CORE_VCC1_S0iX		DDI0_BKLTEN
K	GPIO_N8/C0_BPM1_TX/C1_BPM1_TX		GPIO_SUS1/JTAG2_TCK		GPIO_SUS7/PMC_SUSCLK[3]		GPIO_SUS10		GPIO_CAMERASB01				GPIO_CAMERASB08		DDI0_HPD		CORE_VCC1_S0iX		DDI0_BKLTCTL	
J		GPIO_N6/C0_BPM3_TX/C1_BPM3_TX		JTAG2_TMS		GPIO_SUS9		JTAG_TDO			GPIO_CAMERASB00			GPIO_CAMERASB09		CORE_VCC1_S0iX		CORE_VCC1_S0iX		DDI1_BKLTEN/MDSI_DDC_CLK
H	VSS		GPIO_N5		VSS		JTAG_PREQ_N		PROCHOT_N				VSS		DDI0_DDC_DATA/DDI1_DDC_DATA/M		CORE_VCC1_S0iX		DDI1_VDDEN/MDSI_DDC_DATA	
G		GPIO_N4/C0_BPM0_TX/C1_BPM0_TX		GPIO_SUS4/JTAG2_TDO		VSS		JTAG_PRDY_N			VSS			GPIO_CAMERASB05		VSS		CORE_VCC1_S0iX		VSS
F	VSS		GPIO_N2/C0_BPM2_TX/C1_BPM2_TX		GPIO_SUS8		JTAG_TCK		SVID_CLK				GPIO_CAMERASB11		DDI0_DDC_CLK/DDI1_DDC_CLK/MDSI_D		CORE_VCC1_S0iX		DDI0_VDDEN	
E		GPIO_N1/C0_BPM3_TX/C1_BPM3_TX		GPIO_SUS0		JTAG_TRST_N		JTAG_TDI			GPIO_CAMERASB02			GPIO_CAMERASB06		DDI1_HPD		CORE_VCC1_S0iX		DDI2_DDC_CLK/DDI1_DDC_CLK/UART0
D	GPIO_N0/C0_BPM0_TX/C1_BPM0_TX		VSS		GPIO_SUS6/PMC_SUSCLK[2]		VSS		SVID_DATA				VSS		GPIO_CAMERASB10		CORE_VCC1_S0iX		VSS	
C		ISH_GPIO[13]/C0_BPM2_TX/C1_BPM2_TX		GPIO_SUS3/JTAG2_TDI		GPIO_SUS5/PMC_SUSCLK[1]		SVID_ALERT_N			GPIO_CAMERASB04			GPIO_CAMERASB07		VSS		CORE_VCC1_S0iX		DDI2_HPD
B	DDR_V1P05A_G3		GPIO_N3/C0_BPM1_TX/C1_BPM1_TX		CORE_V1P15_S0iX		JTAG_TMS		VSS				GPIO_CAMERASB03		CORE_VCC1_S0iX		CORE_VCC1_S0iX		CORE_VCC0_S0iX	
A		VSS		CORE_V1P15_S0iX		CORE_V1P15_S0iX		CORE_V1P15_S0iX			CORE_V1P15_S0iX			VSS		CORE_VCC1_S0iX		CORE_VCC1_S0iX		CORE_VCC0_S0iX

Figure 43. Ballout LPPDR3 (T4) Top View Part C

	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
BW		VSS		LPC_V3P3A_V1P8A_S4		SDIO_V3P3A_V1P8A_G3		DDI_VGG_S0iX		DDI_VGG_S0iX		DDI_VGG_S0iX		VSS		VSS		VSS		
BV	VSS		LPC_V3P3A_V1P8A_S4		VSS		DDI_VGG_S0iX		VSS		DDI_VGG_S0iX		VSS		VSS		PWM[0]		VSS	
BU		LPC_AD[0]/ISH_GPIO[12]/ISH_UART_CT		SD3_CD_N		MMC1_D[5]		MMC1_RCLK/MMC1_RESET_N		SD2_D[2]		PMC_PLT_CLK[3]/ISH_GPIO[13]/ISH_UA		PMC_PLT_CLK[2]/ISH_GPIO[12]/ISH_UA		PWM[1]/ISH_GPIO[10]/ISH_UART_DATA		VSS		VSS
BT	VSS		SPI1_CS[0]_N		VSS		MMC1_D[7]		VSS		SD2_D[0]		VSS		PMC_PLT_CLK[5]/ISH_GPIO[15]/ISH_I2C		ISH_GPIO[4]/I2S4_CLK		ISH_GPIO[2]/I2S3_DATAOUT	
BR		LPC_AD[1]/ISH_GPIO[13]/ISH_UART_RT		SPI1_CS[1]_N		SD3_CMD		MMC1_D[4]		SD2_D[1]		PMC_PLT_CLK[1]/ISH_GPIO[11]/ISH_UA		PMC_PLT_CLK[4]/ISH_GPIO[14]/ISH_I2C		VSS		ISH_GPIO[0]/I2S3_CLK		VSS
BP	LPC_AD[2]/ISH_GPIO[14]/ISH_I2C0_DAT		VSS		SD3_D[1]		MMC1_D[6]		SD2_D[3]_CD_N		VSS		PMC_PLT_CLK[0]/ISH_GPIO[10]/ISH_UA		VSS		ISH_GPIO[5]/I2S4_FS		UNCORE_VNN_S4	
BN		SPI1_CLK		VSS		MMC1_CLK		VSS		SD2_CMD		RESERVED		VSS		ISH_GPIO[1]/I2S3_FS		ISH_I2C1_CLK/ISH_SPI_CLK/I2S5_DATAI		UNCORE_VNN_S4
BM	LPC_CLKOUT[1]/ISH_GPIO[11]/ISH_UAR		VSS		VSS		MMC1_D[2]		MMC1_D[0]		VSS		ISH_I2C1_DATA/ISH_SPI_MOSI/I2S5_D		PMC_SUSCLK[0]		ISH_GPIO[3]/I2S3_DATAIN		ISH_GPIO[7]/I2S4_DATAIN	
BL		SPI1_MISO		SD3_CLK		SD3_D[3]		VSS		SD2_CLK		ISH_GPIO[8]/ISH_SPI_CS[0]_N/I2S5_CLK		VSS		VSS		PMC_SLP_S0IX_N		UNCORE_VNN_S4
BK	LPC_CLKOUT[0]/ISH_GPIO[10]/ISH_UAR		SPI1_MOSI		SD3_D[0]		MMC1_D[1]		MMC1_D[3]		ISH_GPIO[6]/I2S4_DATAOUT		ISH_GPIO[9]/ISH_SPI_MISO/I2S5_FS		VSS		PMC_ACPRESENT		VSS	
BJ		LPC_FRAME_N/UART0_DATAIN/SPI2_M		VSS		MMC1_CMD		VSS		VSS		PMC_SLP_S4_N		PMC_WAKE_N		PMC_PLTRST_N		PMC_SLP_S3_N		ICLK_VSFR_G3
BH	VSS		SD3_D[2]		SD3_RCOMP		MMC1_RCOMP		VSS		PMC_PWRBTN_N		VSS		PMC_SUS_STAT_N		PMC_BATLOW_N		VSS	
BG										RTC_EXTPAD		RTC_X1		VSS		VSS		PMC_CORE_PWROK		PCIeCLK_V1P05A_G3
BF		VSS		VSS		VSS			VSS		RTC_X2		RESERVED		VSS		RTC_RST_N		PCIeCLK_V1P05A_G3	
BE										VSS		RESERVED		VSS		RESERVED		PMC_RSMRST_N		VSS
BD	UNCORE_VNN_S4		VSS		VSS		RTC_V3P3A_G5		VSS		RESERVED		RESERVED		VSS		RTC_TEST_N		PCIE_REFCLK[1]_N	
BC										VSS		RESERVED		VSS		VSS		PCIE_REFCLK[1]_P		ICLK_VSFR_G3
BB		VSS		F_V1P8A_G3		RTC_V3P3RTC_G5			ICLK_RCOMP		VSS		VSSA		VSS		PCIE_REFCLK[0]_N		ICLK_VSFR_G3	
BA	UNCORE_VNN_S4		F_V1P05A_G3		F_V1P05A_G3		VSS			VSS		ICLK_OSCOUT		VSS		PCIE_REFCLK[0]_P		PCIE_RCOMP_N		VSS
AY									RESERVED		VSS		ICLK_OSCIN		VSS		PCIE_RXP[0]		PCIE_RCOMP_P	
AW		UNCORE1_V1P05A_G3		UNCORE1_V1P05A_G3		VSS				VSS		VSS		VSS		VSS		PCIE_RXN[0]		PCIE_V1P05_G3
AV									ICLK_ICOMP		PCIE_TXN[0]		VSS		VSS		PCIE_RXN[1]		VSS	
AU	VSS		F_V1P05A_G3		VSS		F_V3P3A_G3			PCIE_TXP[0]		PCIE_TXN[1]		VSS		PCIE_RXP[1]		S_RCOMP_N		PCIE_V1P05_G3
AT		UNCORE_VNN_S4		VSS		VSS			VSS		PCIE_TXP[1]		RESERVED		VSS		S_RCOMP_P		RESERVED	
AR										VSS		RESERVED		VSS		VSS		RESERVED		PWR_RSVD_V1P05
AP	UNCORE_VNN_S4		VSS		VSS		VSS		VSS		USB3_TXN[0]		VSS		USB3_TXN[1]		USB3_RXN[0]		VSS	
AN										USB3_TXP[0]		USB3_TXN[3]		USB3_TXP[1]		USB3_RXP[0]		VSS		PWR_RSVD_V1P05
AM		UNCORE_VNN_S4		VSS		VSS			VSS		USB3_TXN[2]		USB3_TXP[3]		VSS		USB3_RXP[2]		VSS	
AL																				
AK	UNCORE_VNN_S4		RESERVED		RESERVED		VSS			VSS		USB3_TXP[2]		VSS		USB3_RXN[2]		USB3_RCOMP_P		USB3_V1P05A_G3
AJ																				
AH		UNCORE_VNN_S4		VSS		VSS			VSS		VSS		VSS		VSS		USB3_RCOMP_N		USB3_RXP[1]	
AG										VSS		USB_SSIC_RCOMP_N		VSS		VSS		USB3_RXN[1]		USB3_V1P05A_G3
AF	UNCORE_VNN_S4		USBSSIC_V1P2A_G3		VSS		UNCORE_VSFR_G3		VSS		USB_SSIC0_TX_N		USB_SSIC_RCOMP_P		VSS		USB3_RXN[3]		VSS	
AE										USB_SSIC1_TX_P		USB_SSIC0_TX_P		VSS		USB3_RXP[3]		RESERVED		USBSSIC_V1P05A_G3
AD		UNCORE_VNN_S4		VSS		VSS			USB_HSIC_RCOMP		USB_SSIC1_TX_N		VSS		VSS		USB_SSIC0_RX_N		RESERVED	
AC	UNCORE_VNN_S4		USB_V1P8A_G3		VSS		VSS			VSS		VSS		VSS		USB_SSIC1_RX_N		USB_SSIC0_RX_P		VSS
AB									USB_RCOMP		USB_HSIC_1_STROBE		USB_HSIC_1_DATA		VSS		USB_SSIC1_RX_P		USBHSIC_V1P2A_G3	
AA		UNCORE_VNN_S4		VSS		VSS				VSS		USB_HSIC_0_DATA		USB_HSIC_0_STROBE		USB_DN[3]		VSS		USBHSIC_V1P2A_G3
Y									VSS		VSS		VSS		USB_DP3		USB_DP[2]		VSS	
W	UNCORE_VNN_S4		VSS		VSS		VSS			USB_DN[0]		USB_DP[1]		VSS		DDI2_TXN[3]		USB_DN[2]		USB_VDDQ_G3
V		VSS		VSS		VSS			USB_DP[0]		USB_DN[1]		VSS		VSS		DDI2_TXP[3]		USB_VDDQ_G3	
U										DDI2_TXP[2]		RESERVED		USB_OTG_ID		VSS		DDI2_TXN[1]		USB_VDDQ_G3
T	VSS		VSS		VSS		UNCORE_VSFR_G3		VSS		DDI2_TXN[2]		USB_VBUSSNS		VSS		DDI2_AUXP		DDI2_TXP[1]	
R										VSS		VSS		VSS		DDI0_RCOMP_P		DDI2_AUXN		VSS
P		VSS		VSS		VSS			VSS		DDI2_TXP[0]		VSS		VSS		DDI0_RCOMP_N		USB_V3P3A_G3	
N										DDI2_TXN[0]		DDI0_TXP[3]		VSS		DDI0_TXN[2]		VSS		USB_V3P3A_G3
M	VSS		VSS		VSS		VSS		VSS		DDI0_TXN[3]		DDI0_AUXN		DDI0_TXP[2]		DDI0_TXP[1]		DDI1_VDDQ_G3	
L		VSS		VSS		VSS		VSS		VSS		DDI0_AUXP		VSS		DDI0_TXN[1]		VSS		DDI1_VDDQ_G3
K	DDI1_BKLTCTL/MDSI_A_TE/MDSI_C_TE		MCSI_RCOMP		MCSI_3_CLKN		MDSI_RCOMP		MDSI_A_DN[0]		VSS		DDI1_RCOMP_P		VSS		DDI0_TXN[0]		VSS	
J		MCSI_1_DN[0]		MCSI_3_CLKP		MCSI_2_CLKN		VSS		MDSI_A_DP[0]		MDSI_C_DP[3]		DDI1_RCOMP_N		DDI0_TXP[0]		DDI1_AUXN		DDI2_VDDQ_G3
H	MCSI_1_DP[0]		MCSI_1_DN[3]		MCSI_2_CLKP		MCSI_2_DN[1]		VSS		MDSI_A_DN[3]		MDSI_C_DN[3]		MDSI_C_DN[1]		DDI1_TXP[0]		DDI1_AUXP	
G		MCSI_1_DP[3]		VSS		MCSI_2_DP[1]		VSS		MDSI_A_DN[2]		MDSI_A_DP[3]		MDSI_C_DP[1]		VSS		DDI1_TXN[0]		DDI2_VDDQ_G3
F	DDI2_DDC_DATA/DDI1_DDC_DATA/UA		VSS		VSS		VSS		MDSI_A_DP[2]		VSS		VSS		VSS		DDI1_TXP[1]		VSS	
E		VSS		MCSI_1_DP[2]		VSS		RESERVED		VSS		MDSI_A_DP[1]		VSS		DDI1_TXN[1]		DDI1_TXN[2]		VSS
D	VSS		MCSI_1_DN[2]		MCSI_1_CLKN		RESERVED		RESERVED		MDSI_A_DN[1]		MDSI_C_DN[0]		MDSI_C_DN[2]		DDI1_TXN[3]		DDI1_TXP[2]	
C		MCSI_1_DP[1]		MCSI_1_CLKP		MCSI_2_DN[0]		RESERVED		MDSI_A_CLKN		MDSI_C_DP[0]		MDSI_C_DP[2]		MDSI_C_CLKN		DDI1_TXP[3]		VSS
B	MCSI_1_DN[1]		CORE_VCC0_S0iX		MCSI_2_DP[0]		GPION_V1P8A_G3		MDSI_A_CLKP		MIPI_V1P2A_G3		VSS		MDSI_C_CLKP		VSS		VSS	
A		CORE_VCC0_S0iX		VSS		GPION_V1P8A_G3		VSS		MIPI_V1P2A_G3	---	MIPI_V1P2A_G3		VSS		VSS		VSS		---

20.2 SoC T3 Pin List Location

Ball #	DDR3L-RS Customer Pin List
A10	DDI0_DDC_CLK/DDI1_DDC_CLK/MDSI_DDC_CLK
A12	CORE_VCC_S0iX
A13	GPIO_SUS9
A14	GPIO_SUS0
A16	CORE_V1p05A_S0iX
A17	DDR_VDDQG_S4
A18	DDR3_M0_DQ6
A2	RESERVED
A20	DDR3_M0_DM0
A21	DDR3_M0_DQ2
A22	DDR3_M0_DQ4
A23	DDR3_M0_DQ17
A24	VSS
A25	PWR_RSVD_OBS
A3	MCSI_2_DP1
A4	MCSI_2_DP0
A5	MDSI_RCOMP
A6	MCSI_1_DP1
A8	MCSI_RCOMP
A9	DDI0_VDDEN
AA1	ICLK_RCOMP
AA10	SD3_CMD
AA11	VSS
AA12	FST_SPI_CS0_N
AA13	GPIO_SE79
AA14	LPE_I2S1_FRM
AA15	LPE_I2S0_DATAOUT
AA16	LPE_I2S0_CLK
AA17	I2C2_CLK
AA18	VSS
AA19	DDR3_DRAM_PWROK
AA2	VSS
AA20	DDR3_CORE_PWROK
AA21	VSS
AA22	DDR3_M0_DQ52
AA23	DDR3_M0_DQS6_N
AA24	VSS
AA25	DDR3_M0_DQ36
AA3	ISH_I2C1_CLK/ISH_SPI_CLK/I2S5_DATAIN
AA4	ISH_GPIO9/ISH_SPI_MISO/I2S5_FS
AA5	ISH_I2C1_DATA/ISH_SPI_MOSI/I2S5_DATAOUT
AA6	MMC1_CMD
AA7	MMC1_D7
AA8	MMC1_D0
AA9	SD3_PWREN_N
AB1	ICLK_ICOMP
AB10	SD3_D2
AB11	FST_SPI_D0
AB12	FST_SPI_CLK
AB13	UART2_DATAOUT
AB14	LPE_I2S1_CLK
AB15	LPE_I2S0_DATAIN
AB16	I2C5_CLK
AB17	I2C4_CLK/DDI1_DDC_CLK/DDI2_DDC_CLK/MDSI_DDC_CLK
AB18	I2C4_DATA/DDI1_DDC_DATA/DDI2_DDC_DATA/MDSI_DDC_DATA
AB19	LPE_I2S2_CLK
AB2	ISH_GPIO0/I2S3_CLK
AB20	PCIE_CLKREQ0_N
AB21	RESERVED
AB22	DDR3_M0_DQ48
AB23	DDR3_M0_DQS6_P
AB24	DDR3_M0_DQS4_N

Ball #	DDR3L-RS Customer Pin List
AB25	DDR3_M0_DQ33
AB3	ISH_GPIO2/I2S3_DATAOUT
AB4	PMC_PLT_CLK3/ISH_GPIO13/ ISH_UART_RTS_N/SPI2_CLK
AB5	SD2_CMD
AB6	MMC1_D2
AB7	MMC1_D6
AB8	SD3_1P8_EN
AB9	SD3_D1
AC1	ISH_GPIO7/I2S4_DATAIN
AC10	SD3_D0
AC11	FST_SPI_D1
AC12	UART2_RTS_N
AC13	UART1_CTS_N
AC14	UART1_DATAOUT/ UART0_DATAOUT
AC15	NFC_I2C_DATA
AC16	NFC_I2C_CLK
AC17	I2C2_DATA
AC18	GPIO_SW78
AC19	LPE_I2S2_FRM
AC2	ISH_GPIO4/I2S4_CLK
AC20	GPIO_SW93
AC21	RESERVED
AC22	DDR3_M0_DQ49
AC23	DDR3_M0_DQ37
AC24	DDR3_M0_DM4
AC25	DDR3_M0_DQS4_P
AC3	PMC_PLT_CLK4/ISH_GPIO14/ ISH_I2C0_DATA/SPI2_MISO
AC4	PMC_PLT_CLK2/ISH_GPIO12/ ISH_UART_CTS_N/ SPI2_CS0_N
AC5	VSS
AC6	MMC1_D3
AC7	MMC1_D4
AC8	MMC1_D5
AC9	SD3_CLK

Ball #	DDR3L-RS Customer Pin List
AD1	RESERVED
AD10	SD3_D3
AD12	UART2_CTS_N
AD13	VSS
AD14	UART1_DATAIN/ UART0_DATAIN
AD16	I2C6_DATA/SD3_WP
AD17	VSS
AD18	MMC1_RESET_N
AD2	PWM1/ISH_GPIO10/ ISH_UART_DATAOUT
AD20	SD3_WP
AD21	DDR3_M0_DQ51
AD22	VSS
AD23	DDR3_M0_DQ38
AD24	DDR3_M0_DQ32
AD25	VSS
AD3	PWM0
AD4	SD2_D0
AD5	SD2_CLK
AD6	SD2_D3_CD_N
AD8	MMC1_CLK
AD9	VSS
AE1	RESERVED
AE10	MMC1_RCOMP
AE12	UART2_DATAIN
AE13	UART1_RTS_N
AE14	SD3_RCOMP
AE16	I2C6_CLK/NMI_N
AE17	I2C5_DATA
AE18	UART0_DATAIN
AE2	RESERVED
AE20	DDR3_M0_RCOMPPD
AE21	DDR3_M0_OCAVREF
AE22	DDR3_M0_ODQVREF
AE23	DDR3_M0_DQ35
AE24	PWR_RSVD_OBS

Ball #	DDR3L-RS Customer Pin List
AE25	PWR_RSVD_OBS
AE3	PMC_PLT_CLK5/ISH_GPIO15/ ISH_I2C0_CLK/SPI2_MOSI
AE4	SD2_D1
AE5	SD2_D2
AE6	DDI_VGG_S0iX
AE8	MMC1_RCLK/MMC1_RESET_N
AE9	SD3_CD_N
B1	RESERVED
B10	GPIO_CAMERASB10
B12	PMC_SUSCLK3
B13	VSS
B14	JTAG2_TDI
B16	GPIO0_RCOMP
B17	VSS
B18	DDR3_M0_DQ1
B2	MDSI_A_DN0
B20	DDR3_M0_DQ5
B21	VSS
B22	DDR3_M0_DQ7
B23	DDR3_M0_DQ22
B24	DDR3_M0_DQ20
B25	PWR_RSVD_OBS
B3	MCSI_2_DN1
B4	MCSI_2_DN0
B5	VSS
B6	MCSI_1_DN1
B8	DDI0_BKLTCTL
B9	VSS
C1	MDSI_A_DN1
C10	DDI0_DDC_DATA/ DDI1_DDC_DATA/ MDSI_DDC_DATA
C11	GPIO_CAMERASB11
C12	JTAG_TDI
C13	GPIO_SUS8
C14	PMC_SUSCLK2
C15	JTAG2_TCK
C16	GPIO_N6/C0_BPM3_TX/ C1_BPM3_TX
C17	GPIO_N3/C0_BPM1_TX/ C1_BPM1_TX
C18	DDR3_M0_DQ3
C19	DDR3_M0_DQS0_N
C2	MDSI_A_DP1
C20	DDR3_M0_DQ8
C21	DDR3_M0_DQS1_N
C22	DDR3_M0_DQS1_P
C23	DDR3_M0_DQ18
C24	DDR3_M0_DQS2_P
C25	DDR3_M0_DQS2_N
C3	MDSI_A_DP0
C4	MDSI_A_CLKP
C5	MCSI_2_CLKN
C6	MCSI_1_DN3
C7	MCSI_1_DP2
C8	DDI0_BKLTEN
C9	DDI2_DDC_DATA/ DDI1_DDC_DATA/ UART0_DATAIN/ MDSI_DDC_DATA/MDSI_C_TE
D1	DDI1_RCOMP_N
D10	GPIO_CAMERASB09
D11	GPIO_CAMERASB08
D12	VSS
D13	JTAG_PRDY_N
D14	JTAG2_TDO
D15	JTAG2_TMS
D16	GPIO_N8/C0_BPM1_TX/ C1_BPM1_TX
D17	ISH_GPIO13/C0_BPM2_TX/ C1_BPM2_TX
D18	DDR3_M0_DQ0
D19	DDR3_M0_DQS0_P
D2	DDI1_RCOMP_P
D20	DDR3_M0_DQ15
D21	DDR3_M0_DM1

Ball #	DDR3L-RS Customer Pin List
D22	DDR3_M0_DQ12
D23	DDR3_M0_DQ9
D24	DDR3_M0_DQ21
D25	DDR3_M0_DM2
D3	MDSI_A_DP2
D4	MDSI_A_CLKN
D5	MCSI_2_CLKP
D6	MCSI_1_DP3
D7	MCSI_1_DN2
D8	DDI1_BKLTCTL/MDSI_A_TE/MDSI_C_TE
D9	DDI0_HPD
E1	DDI0_AUXP
E10	DDI2_DDC_CLK/DDI1_DDC_CLK/UART0_DATAOUT/MDSI_DDC_CLK/MDSI_A_TE
E11	VSS
E12	SVID_ALERT_N
E13	JTAG_TRST_N
E14	JTAG_TCK
E15	VSS
E16	GPIO_N4/C0_BPM0_TX/C1_BPM0_TX
E17	GPIO_N0/C0_BPM0_TX/C1_BPM0_TX
E18	VSS
E19	DDR3_M0_DQ10
E2	VSS
E20	DDR3_M0_DQ14
E21	VSS
E22	DDR3_M0_DQ13
E23	DDR3_M0_DQ11
E24	VSS
E25	DDR3_M0_DQ16
E3	MDSI_A_DN2
E4	MDSI_A_DP3
E5	MDSI_A_DN3
E6	VSS
E7	VSS
E8	MCSI_1_DP0
E9	DDI2_HPD
F1	DDI0_AUXN
F10	RESERVED
F11	SVID_CLK
F12	SVID_DATA
F13	JTAG_TDO
F14	JTAG_TMS
F15	VSS
F16	GPIO_N2/C0_BPM2_TX/C1_BPM2_TX
F17	GPIO_N1/C0_BPM3_TX/C1_BPM3_TX
F18	GPIO_DFX4
F19	DDR3_M0_DM3
F2	DDI0_TXN3
F20	DDR3_M0_DQ24
F21	DDR3_M0_DQ25
F22	DDR3_M0_CKE3
F23	DDR3_M0_CKE2
F24	DDR3_M0_DQ23
F25	DDR3_M0_DQ19
F3	DDI0_TXP3
F4	RESERVED
F5	DDI0_TXP2
F6	DDI0_TXN2
F7	MCSI_1_CLKP
F8	MCSI_1_DN0
F9	PROCHOT_N
G10	VSS
G11	VSS
G12	RESERVED
G13	UNCORE_V1p05A_S0iX
G14	CORE_VCC_S0iX
G15	CORE_VCC_S0iX
G16	VSS

Ball #	DDR3L-RS Customer Pin List
G17	CORE_V1p05A_S0iX
G18	CORE_V1p05A_S0iX
G19	DDR_V1P05A_G3
G20	DDR3_M0_DQ26
G21	DDR3_M0_DQ28
G22	DDR3_M0_CKE0
G23	DDR3_M0_CKE1
G3	DDI0_TXP0
G4	DDI0_TXN0
G5	VSS
G6	VSS
G7	MCSI_1_CLKN
G8	UNCORE_VSFR_G3
G9	MIPI_V1P2A_G3
H1	DDI2_TXN2
H10	VSS
H11	USB_V1P8A_G3
H12	USB_V1P8A_G3
H13	UNCORE_V1p05A_S0iX
H14	CORE0_VSFR_G3
H15	CORE_VCC_S0iX
H16	VSS
H17	VSS
H18	DDR_V1P05A_G3
H19	DDR3_M0_DQS3_N
H2	DDI2_TXP2
H20	DDR3_M0_DQS3_P
H21	VSS
H22	DDR3_M0_DQ31
H23	DDR3_M0_DQ27
H24	DDR3_M0_MA09
H25	DDR3_M0_MA14
H3	DDI0_RCOMP_P
H4	DDI0_RCOMP_N
H5	DDI2_AUXP
H6	DDI0_TXP1
H7	USB_VDDQ_G3

Ball #	DDR3L-RS Customer Pin List
H8	DDI_USB_VDDQ_G3
H9	MIPI_V1P2A_G3
J1	USB_RCOMP
J10	UNCORE_VNN_S4
J11	VSS
J12	VSS
J13	VSS
J14	CORE_VCC_S0iX
J15	CORE_VCC_S0iX
J16	VSS
J17	VSS
J18	VSS
J19	VSS
J2	VSS
J20	DDR3_M0_DQ30
J21	DDR3_M0_DQ29
J22	DDR3_M0_CK0_P
J23	DDR3_M0_CK0_N
J24	VSS
J25	DDR3_M0_MA8
J3	DDI2_TXP0
J4	DDI2_TXN0
J5	DDI2_AUXN
J6	DDI0_TXN1
J7	DDI_USB_VDDQ_G3
J8	USBHSIC_V1P2A_G3
J9	UNCORE_VNN_S4
K1	USB_OTG_ID
K10	UNCORE_VNN_S4
K11	UNCORE_VNN_S4
K12	UNCORE_VNN_S4
K13	VSS
K14	VSS
K15	CORE_VCC_S0iX
K16	F_V1p05A_S0iX
K17	CORE_V1p05A_S0iX
K18	DDR_VDDQG_S4

Ball #	DDR3L-RS Customer Pin List
K19	VSS
K2	USB_VBUSSNS
K20	DDR3_M0_CK1_P
K21	DDR3_M0_CK1_N
K22	DDR3_M0_BS2
K23	DDR3_M0_MA15
K24	DDR3_M0_MA6
K25	DDR3_M0_MA11
K3	DDI2_TXN1
K4	DDI2_TXP1
K5	DDI2_TXP3
K6	DDI2_TXN3
K7	DDI_USB_VDDQ_G3
K8	USBSSIC_V1P2A_G3
K9	UNCORE_VNN_S4
L10	UNCORE_VNN_S4
L11	UNCORE_VNN_S4
L12	UNCORE_VNN_S4
L13	UNCORE2_V1P05A_G3
L14	CORE_VCC_S0iX
L15	CORE_VCC_S0iX
L16	F_V1p05A_S0iX
L17	CORE_V1p05A_S0iX
L18	DDR_VDDQG_S4
L19	DDR_VDDQG_S4
L20	DDR_VDDQG_S4
L21	VSS
L22	DDR3_M0_MA5
L23	DDR3_M0_MA1
L3	USB_DN1
L4	USB_DP1
L5	VSS
L6	UNCORE_VSFR_G3
L7	VSS
L8	USB_V3P3A_G3
L9	UNCORE_VNN_S4
M1	USB_HSIC_0_STROBE

Ball #	DDR3L-RS Customer Pin List
M10	UNCORE_VNN_S4
M11	UNCORE_VNN_S4
M12	UNCORE_VSFR_G3
M13	VSS
M14	CORE_VCC_S0iX
M15	CORE_VCC_S0iX
M16	VSS
M17	VSS
M18	DDR_VDDQG_S4
M19	VSS
M2	USB_HSIC_0_DATA
M20	DDR3_M0_MA3
M21	DDR3_M0_BS0
M22	DDR3_M0_MA4
M23	DDR3_M0_MA13
M24	DDR3_M0_MA0
M25	DDR3_M0_MA12
M3	USB_DN3
M4	USB_DP3
M5	USB_DP2
M6	USB_DN2
M7	USBSSIC_V1P05A_G3
M8	VSS
M9	UNCORE_VNN_S4
N1	USB_HSIC_RCOMP
N10	UNCORE_VNN_S4
N11	DDI_VGG_S0iX
N12	DDI_VGG_S0iX
N13	VSS
N14	CORE_VCC_S0iX
N15	CORE_VCC_S0iX
N16	VSS
N17	VSS
N18	VSS
N19	DDR3_M0_BS1
N2	VSS
N20	DDR3_M0_MA10

Ball #	DDR3L-RS Customer Pin List
N21	VSS
N22	DDR3_M0_RAS_N
N23	DDR3_M0_MA2
N24	VSS
N25	DDR3_M0_MA7
N3	USB_HSIC_1_STROBE
N4	USB_HSIC_1_DATA
N5	USB3_RXP0
N6	USB_DP0
N7	MPHY_1P05A_G3
N8	MPHY_1P05A_G3
N9	VSS
P1	USB3_TXP0
P10	DDI_VGG_S0iX
P11	DDI_VGG_S0iX
P12	VSS
P13	RESERVED
P14	VSS
P15	VSS
P16	VSS
P17	VSS
P18	VSS
P19	DDRSFR_VDDQG_S4
P2	USB3_TXN0
P20	DDR3_M0_CAS_N
P21	DDR3_M0_WE_N
P22	DDR3_M0_CS0_N
P23	DDR3_M0_CS1_N
P24	DDR3_M0_ODT0
P25	DDR3_M0_ODT1
P3	USB3_RCOMP_P
P4	USB3_RCOMP_N
P5	USB3_RXN0
P6	USB_DN0
P7	PCIeCLK_V1P05A_G3
P8	VSS
P9	UNCORE_VNN_S4

Ball #	DDR3L-RS Customer Pin List
R10	VSS
R11	DDI_VGG_S0iX
R12	VSS
R13	DDI_VGG_S0iX
R14	DDI_VGG_S0iX
R15	VSS
R16	DDI_V1p05A_S0iX
R17	DDI_V1p05A_S0iX
R18	DDR_V1P05A_G3
R19	VSS
R20	VSS
R21	VSS
R22	DDR3_M0_DQ45
R23	DDR3_M0_DQ44
R3	PCIE_TXN0
R4	PCIE_TXP0
R5	VSS
R6	VSS
R7	F_V1P05A_G3
R8	ICLK_VSFR_G3
R9	VSS
T1	PCIE_RXP0
T10	UNCORE1_V1P05A_G3
T11	DDI_VGG_S0iX
T12	DDI_VGG_S0iX
T13	DDI_VGG_S0iX
T14	DDI_VGG_S0iX
T15	DDI_VGG_S0iX
T16	DDI_VGG_S0iX
T17	DDI_VGG_S0iX
T18	VSS
T19	DDR_VDDQG_S4
T2	PCIE_RXN0
T20	DDR3_M0_DQ57
T21	DDR3_M0_DQ63
T22	DDR3_M0_DQ56
T23	DDR3_M0_DM7

Ball #	DDR3L-RS Customer Pin List
T24	DDR3_M0_DQ41
T25	DDR3_M0_DQ47
T3	PCIE_RCOMP_P
T4	PCIE_RCOMP_N
T5	PMC_RSMRST_N
T6	PMC_CORE_PWROK
T7	F_V1P05A_G3
T8	VSS
T9	F_V1P05A_G3
U1	RTC_EXTPAD
U10	DDI_VGG_S0iX
U11	DDI_VGG_S0iX
U12	DDI_VGG_S0iX
U13	DDI_VGG_S0iX
U14	DDI_VGG_S0iX
U15	DDI_VGG_S0iX
U16	DDI_VGG_S0iX
U17	LPE_I2S2_DATAIN
U18	DDR_V1P05A_G3
U19	DDR3_M0_DRAMRST_N
U2	VSS
U20	DDR3_M0_DQ62
U21	DDR3_M0_DQ60
U22	DDR3_M0_DQS7_P
U23	DDR3_M0_DQS7_N
U24	VSS
U25	DDR3_M0_DQS5_N
U3	RTC_X1
U4	RTC_X2
U5	RTC_RST_N
U6	RTC_TEST_N
U7	F_V3P3A_G3
U8	RTC_V3P3A_G5
U9	F_V1P8A_G3
V1	UNCORE_VNN_S4
V10	UNCORE_V1P8A_G3
V11	PMC_RSTBTN_N

Ball #	DDR3L-RS Customer Pin List
V12	PMC_SUSPWRDNACK
V13	UART0_DATAIN
V14	DDI_VGG_S0iX
V15	UNCORE_V1P8A_G3
V16	UNCORE_V1P8A_G3
V17	DDI_VGG_S0iX
V18	LPE_I2S2_DATAOUT
V19	DDR_VDDQG_S4
V2	PCIE_REFCLK0_N
V20	DDR3_M0_DQ59
V21	VSS
V22	DDR3_M0_DQ46
V23	DDR3_M0_DQ40
V24	DDR3_M0_DM5
V25	DDR3_M0_DQS5_P
V3	PCIE_REFCLK0_P
V4	RESERVED
V5	PMC_SUS_STAT_N
V6	VSS
V7	VSS
V8	RTC_V3P3RTC_G5
V9	VSS
W10	RESERVED
W11	VSS
W12	VSS
W13	SDIO_V3P3A_V1P8A_G3
W14	DDI_VGG_S0iX
W15	I2C1_DATA
W16	I2C1_CLK
W17	VSS
W18	VSS
W19	VSS
W20	DDR3_M0_DQ58
W21	DDR3_M0_DQ61
W22	DDR3_M0_DQ42
W23	DDR3_M0_DQ43
W3	PMC_PWRBTN_N

Ball #	DDR3L-RS Customer Pin List
W4	PMC_SLP_S0IX_N
W5	VSS
W6	ICLK_OSCOUT
W7	ICLK_OSCIN
W8	VSS
W9	UNCORE_V1P8A_G3
Y1	ISH_GPIO3/I2S3_DATAIN
Y10	RESERVED
Y11	VSS
Y12	UART0_DATAOUT
Y13	LPE_I2S1_DATAIN
Y14	LPE_I2S1_DATAOUT
Y15	DDI_VGG_S0iX
Y16	LPE_I2S0_FRM
Y17	I2C0_CLK
Y18	I2C0_DATA
Y19	DDR3_M0_DQ50

Ball #	DDR3L-RS Customer Pin List
Y2	ISH_GPIO1/I2S3_FS
Y20	DDR3_M0_DQ54
Y21	DDR3_M0_DQ55
Y22	DDR3_M0_DM6
Y23	DDR3_M0_DQ53
Y24	DDR3_M0_DQ34
Y25	DDR3_M0_DQ39
Y3	PMC_WAKE_N
Y4	PMC_SUSCLK0
Y5	PMC_PLTRST_N
Y6	VSSA
Y7	VSS
Y8	MMC1_D1
Y9	USB_OC0_N

20.3 SoC T4 Pin List Location

Ball #	Customer Name - LPDDR3
A1	--
A10	--
A11	MIPI_V1P2A_G3
A12	--
A13	VSS
A14	--
A15	GPION_V1P8A_G3
A16	--
A17	VSS
A18	--
A19	CORE_VCC0_S0iX
A2	--
A20	--
A21	CORE_VCC0_S0iX

Ball #	Customer Name - LPDDR3
A22	--
A23	CORE_VCC1_S0iX
A24	--
A25	CORE_VCC1_S0iX
A26	--
A27	VSS
A28	--
A29	--
A3	VSS
A30	CORE_V1P15_S0iX
A31	--
A32	--
A33	CORE_V1P15_S0iX
A34	--

Ball #	Customer Name - LPDDR3
A35	CORE_V1P15_S0iX
A36	--
A37	CORE_V1P15_S0iX
A38	--
A39	VSS
A4	--
A40	--
A41	DDR_V1P05A_G3
A42	--
A43	DDR_V1P05A_G3
A44	--
A45	VSS
A46	--
A47	VSS
A48	--
A49	VSS
A5	VSS
A50	--
A51	VSS
A52	--
A53	VSS
A54	--
A55	VSS
A56	--
A57	VSS
A58	--
A59	--
A6	--
A7	VSS
A8	--
A9	MIPI_V1P2A_G3
AA1	USBHSIC_V1P2A_G3
AA10	--
AA11	VSS
AA12	--
AA13	--
AA14	--

Ball #	Customer Name - LPDDR3
AA15	VSS
AA16	--
AA17	VSS
AA18	--
AA19	UNCORE_VNN_S4
AA2	--
AA20	--
AA21	UNCORE_VNN_S4
AA22	--
AA23	UNCORE_VNN_S4
AA24	--
AA25	UNCORE_V1P15_S0iX
AA26	--
AA27	UNCORE_VNN_S4
AA28	--
AA29	UNCORE_VNN_S4
AA3	VSS
AA30	--
AA31	--
AA32	VSS
AA33	--
AA34	UNCORE_VNN_S4
AA35	--
AA36	UNCORE_VNN_S4
AA37	--
AA38	VSS
AA39	--
AA4	--
AA40	VSS
AA41	--
AA42	DDR_V1P05A_G3
AA43	--
AA44	VSS
AA45	--
AA46	VSS
AA47	--
AA48	--

Ball #	Customer Name - LPDDR3
AA49	VSS
AA5	USB_DN[3]
AA50	--
AA51	LPDDR3_M0_DQ[18]_B
AA52	--
AA53	LPDDR3_M0_DQ[21]_B
AA54	--
AA55	LPDDR3_M0_DQ[07]_B
AA56	--
AA57	LPDDR3_M0_DQ[08]_B
AA58	--
AA59	DDR_VDDQG_S4
AA6	--
AA7	USB_HSIC_0_STROBE
AA8	--
AA9	USB_HSIC_0_DATA
AB1	--
AB10	USB_HSIC_1_STROBE
AB11	--
AB12	USB_RCOMP
AB13	--
AB14	--
AB15	--
AB16	--
AB17	--
AB18	--
AB19	--
AB2	USBHSIC_V1P2A_G3
AB20	--
AB21	--
AB22	--
AB23	--
AB24	--
AB25	--
AB26	--
AB27	--
AB28	--
AB29	--
AB3	--
AB30	--
AB31	--
AB32	--
AB33	--
AB34	--
AB35	--
AB36	--
AB37	--
AB38	--
AB39	--
AB4	USB_SSIC1_RX_P
AB40	--
AB41	--
AB42	--
AB43	--
AB44	--
AB45	--
AB46	--
AB47	--
AB48	LPDDR3_M0_DM[03]_B
AB49	--
AB5	--
AB50	LPDDR3_M0_DM[02]_B
AB51	--
AB52	LPDDR3_M0_DQ[23]_B
AB53	--
AB54	LPDDR3_M0_DQ[10]_B
AB55	--
AB56	LPDDR3_M0_DQ[09]_B
AB57	--
AB58	VSS
AB59	--
AB6	VSS
AB7	--
AB8	USB_HSIC_1_DATA

Ball #	Customer Name - LPDDR3
AB9	--
AC1	VSS
AC10	--
AC11	VSS
AC12	--
AC13	--
AC14	VSS
AC15	--
AC16	VSS
AC17	--
AC18	USB_V1P8A_G3
AC19	--
AC2	--
AC20	UNCORE_VNN_S4
AC21	--
AC22	VSS
AC23	--
AC24	UNCORE_VSS_SENSE
AC25	--
AC26	UNCORE_VNN_SENSE
AC27	--
AC28	RESERVED
AC29	--
AC3	USB_SSIC0_RX_P
AC30	--
AC31	UNCORE_VNN_S4
AC32	--
AC33	UNCORE_VNN_S4
AC34	--
AC35	VSS
AC36	--
AC37	VSS
AC38	--
AC39	VSS
AC4	--
AC40	--
AC41	DDRSFRCH0_VDDQG_S4

Ball #	Customer Name - LPDDR3
AC42	--
AC43	VSS
AC44	--
AC45	VSS
AC46	--
AC47	--
AC48	--
AC49	VSS
AC5	USB_SSIC1_RX_N
AC50	--
AC51	VSS
AC52	--
AC53	VSS
AC54	--
AC55	LPDDR3_M0_DQ[14]_B
AC56	--
AC57	LPDDR3_M0_DQ[13]_B
AC58	--
AC59	DDR_VDDQG_S4
AC6	--
AC7	VSS
AC8	--
AC9	VSS
AD1	--
AD10	USB_SSIC1_TX_N
AD11	--
AD12	USB_HSIC_RCOMP
AD13	--
AD14	--
AD15	VSS
AD16	--
AD17	VSS
AD18	--
AD19	UNCORE_VNN_S4
AD2	RESERVED
AD20	--
AD21	UNCORE_VNN_S4

Ball #	Customer Name - LPDDR3
AD22	--
AD23	VSS
AD24	--
AD25	UNCORE_VNN_S4
AD26	--
AD27	UNCORE_VNN_S4
AD28	--
AD29	UNCORE_VSFR_G3
AD3	--
AD30	--
AD31	--
AD32	CORE_VCC1_S0iX
AD33	--
AD34	CORE_VCC1_S0iX
AD35	--
AD36	CORE_VCC1_S0iX
AD37	--
AD38	CORE_VCC1_S0iX
AD39	--
AD4	USB_SSIC0_RX_N
AD40	F_V1P15_S0iX
AD41	--
AD42	DDRCH0_VDDQG_S4
AD43	--
AD44	LPDDR3_M0_RCOMPPD
AD45	--
AD46	VSS
AD47	--
AD48	LPDDR3_M0_DQ[25]_B
AD49	--
AD5	--
AD50	LPDDR3_M0_DQ[26]_B
AD51	--
AD52	VSS
AD53	--
AD54	VSS
AD55	--

Ball #	Customer Name - LPDDR3
AD56	VSS
AD57	--
AD58	LPDDR3_M0_DQS[1]_N_B
AD59	--
AD6	VSS
AD7	--
AD8	VSS
AD9	--
AE1	USBSSIC_V1P05A_G3
AE10	--
AE11	USB_SSIC1_TX_P
AE12	--
AE13	--
AE14	--
AE15	--
AE16	--
AE17	--
AE18	--
AE19	--
AE2	--
AE20	--
AE21	--
AE22	--
AE23	--
AE24	--
AE25	--
AE26	--
AE27	--
AE28	--
AE29	--
AE3	RESERVED
AE30	--
AE31	--
AE32	--
AE33	--
AE34	--
AE35	--

Ball #	Customer Name - LPDDR3
AE36	--
AE37	--
AE38	--
AE39	--
AE4	--
AE40	--
AE41	--
AE42	--
AE43	--
AE44	--
AE45	--
AE46	--
AE47	--
AE48	--
AE49	LPDDR3_M0_DQ[30]_B
AE5	USB3_RXP[3]
AE50	--
AE51	LPDDR3_M0_DQS[3]_N_B
AE52	--
AE53	LPDDR3_M0_DQ[27]_B
AE54	--
AE55	LPDDR3_M0_DQ[12]_B
AE56	--
AE57	LPDDR3_M0_DQS[1]_P_B
AE58	--
AE59	DDR_VDDQG_S4
AE6	--
AE7	VSS
AE8	--
AE9	USB_SSIC0_TX_P
AF1	--
AF10	USB_SSIC0_TX_N
AF11	--
AF12	VSS
AF13	--
AF14	UNCORE_VSFR_G3
AF15	--

Ball #	Customer Name - LPDDR3
AF16	VSS
AF17	--
AF18	USBSSIC_V1P2A_G3
AF19	--
AF2	VSS
AF20	UNCORE_VNN_S4
AF21	--
AF22	UNCORE_VNN_S4
AF23	--
AF24	UNCORE2_V1P05A_G3
AF25	--
AF26	UNCORE2_V1P05A_G3
AF27	--
AF28	RESERVED
AF29	--
AF3	--
AF30	--
AF31	VSS
AF32	--
AF33	CORE_VCC1_S0iX
AF34	--
AF35	CORE_VCC1_S0iX
AF36	--
AF37	CORE_VCC1_S0iX
AF38	--
AF39	CORE_VCC1_SENSE
AF4	USB3_RXN[3]
AF40	--
AF41	CORE_VSS1_SENSE
AF42	--
AF43	VSS
AF44	--
AF45	RESERVED
AF46	--
AF47	--
AF48	LPDDR3_M0_DQ[29]_B
AF49	--

Ball #	Customer Name - LPDDR3
AF5	--
AF50	LPDDR3_M0_DQS[3]_P_B
AF51	--
AF52	LPDDR3_M0_DQ[31]_B
AF53	--
AF54	VSS
AF55	--
AF56	LPDDR3_M0_DM[1]_B
AF57	--
AF58	VSS
AF59	--
AF6	VSS
AF7	--
AF8	USB_SSIC_RCOMP_P
AF9	--
AG1	USB3_V1P05A_G3
AG10	--
AG11	VSS
AG12	--
AG13	--
AG14	--
AG15	--
AG16	--
AG17	--
AG18	--
AG19	--
AG2	--
AG20	--
AG21	--
AG22	--
AG23	--
AG24	--
AG25	--
AG26	--
AG27	--
AG28	--
AG29	--

Ball #	Customer Name - LPDDR3
AG3	USB3_RXN[1]
AG30	--
AG31	--
AG32	--
AG33	--
AG34	--
AG35	--
AG36	--
AG37	--
AG38	--
AG39	--
AG4	--
AG40	--
AG41	--
AG42	--
AG43	--
AG44	--
AG45	--
AG46	--
AG47	--
AG48	--
AG49	VSS
AG5	VSS
AG50	--
AG51	VSS
AG52	--
AG53	VSS
AG54	--
AG55	LPDDR3_M0_DQ[11]_B
AG56	--
AG57	LPDDR3_M0_DQ[15]_B
AG58	--
AG59	VSS
AG6	--
AG7	VSS
AG8	--
AG9	USB_SSIC_RCOMP_N

Ball #	Customer Name - LPDDR3
AH1	--
AH10	VSS
AH11	--
AH12	VSS
AH13	--
AH14	--
AH15	VSS
AH16	--
AH17	VSS
AH18	--
AH19	UNCORE_VNN_S4
AH2	USB3_RXP[1]
AH20	--
AH21	UNCORE_VNN_S4
AH22	--
AH23	UNCORE_VNN_S4
AH24	--
AH25	UNCORE_VNN_S4
AH26	--
AH27	RESERVED
AH28	--
AH29	VSS
AH3	--
AH30	--
AH31	--
AH32	CORE_VCC1_S0iX
AH33	--
AH34	CORE_VCC1_S0iX
AH35	--
AH36	CORE_VCC1_S0iX
AH37	--
AH38	CORE_VCC1_S0iX
AH39	--
AH4	USB3_RCOMP_N
AH40	VSS
AH41	--
AH42	VSS

Ball #	Customer Name - LPDDR3
AH43	--
AH44	VSS
AH45	--
AH46	RESERVED
AH47	--
AH48	LPDDR3_M0_DQ[28]_B
AH49	--
AH5	--
AH50	LPDDR3_M0_DQ[24]_B
AH51	--
AH52	VSS
AH53	--
AH54	VSS
AH55	--
AH56	VSS
AH57	--
AH58	VSS
AH59	--
AH6	VSS
AH7	--
AH8	VSS
AH9	--
AJ1	--
AJ10	--
AJ11	--
AJ12	--
AJ13	--
AJ14	--
AJ15	--
AJ16	--
AJ17	--
AJ18	--
AJ19	--
AJ2	--
AJ20	--
AJ21	--
AJ22	--

Ball #	Customer Name - LPDDR3
AJ23	--
AJ24	--
AJ25	--
AJ26	--
AJ27	--
AJ28	--
AJ29	--
AJ3	--
AJ30	--
AJ31	--
AJ32	--
AJ33	--
AJ34	--
AJ35	--
AJ36	--
AJ37	--
AJ38	--
AJ39	--
AJ4	--
AJ40	--
AJ41	--
AJ42	--
AJ43	--
AJ44	--
AJ45	--
AJ46	--
AJ47	--
AJ48	--
AJ49	--
AJ5	--
AJ50	--
AJ51	--
AJ52	--
AJ53	--
AJ54	--
AJ55	--
AJ56	--

Ball #	Customer Name - LPDDR3
AJ57	--
AJ58	--
AJ59	--
AJ6	--
AJ7	--
AJ8	--
AJ9	--
AK1	USB3_V1P05A_G3
AK10	--
AK11	VSS
AK12	--
AK13	--
AK14	VSS
AK15	--
AK16	RESERVED
AK17	--
AK18	RESERVED
AK19	--
AK2	--
AK20	UNCORE_VNN_S4
AK21	--
AK22	DDI_VGG_S0iX
AK23	--
AK24	DDI_VGG_S0iX
AK25	--
AK26	VSS
AK27	--
AK28	CORE1_VSFR_G3
AK29	--
AK3	USB3_RCOMP_P
AK30	--
AK31	CORE1_VSFR_G3
AK32	--
AK33	CORE_VCC1_S0iX
AK34	--
AK35	CORE_VCC1_S0iX
AK36	--

Ball #	Customer Name - LPDDR3
AK37	CORE_VCC1_S0iX
AK38	--
AK39	RESERVED
AK4	--
AK40	--
AK41	RESERVED
AK42	--
AK43	VSS
AK44	--
AK45	VSS
AK46	--
AK47	--
AK48	--
AK49	LPDDR3_DRAM_PWROK
AK5	USB3_RXN[2]
AK50	--
AK51	LPDDR3_CORE_PWROK
AK52	--
AK53	VSS
AK54	--
AK55	VSS
AK56	--
AK57	VSS
AK58	--
AK59	VSS
AK6	--
AK7	VSS
AK8	--
AK9	USB3_TXP[2]
AL1	--
AL10	--
AL11	--
AL12	--
AL13	--
AL14	--
AL15	--
AL16	--

Ball #	Customer Name - LPDDR3
AL17	--
AL18	--
AL19	--
AL2	--
AL20	--
AL21	--
AL22	--
AL23	--
AL24	--
AL25	--
AL26	--
AL27	--
AL28	--
AL29	--
AL3	--
AL30	--
AL31	--
AL32	--
AL33	--
AL34	--
AL35	--
AL36	--
AL37	--
AL38	--
AL39	--
AL4	--
AL40	--
AL41	--
AL42	--
AL43	--
AL44	--
AL45	--
AL46	--
AL47	--
AL48	--
AL49	--
AL5	--

Ball #	Customer Name - LPDDR3
AL50	--
AL51	--
AL52	--
AL53	--
AL54	--
AL55	--
AL56	--
AL57	--
AL58	--
AL59	--
AL6	--
AL7	--
AL8	--
AL9	--
AM1	--
AM10	USB3_TXN[2]
AM11	--
AM12	VSS
AM13	--
AM14	--
AM15	VSS
AM16	--
AM17	VSS
AM18	--
AM19	UNCORE_VNN_S4
AM2	VSS
AM20	--
AM21	DDI_VGG_S0iX
AM22	--
AM23	VSS
AM24	--
AM25	DDI_VGG_S0iX
AM26	--
AM27	DDI_VGG_S0iX
AM28	--
AM29	VSS
AM3	--

Ball #	Customer Name - LPDDR3
AM30	--
AM31	--
AM32	VSS
AM33	--
AM34	PWR_RSVD_OBS
AM35	--
AM36	DDI_VGG_S0iX
AM37	--
AM38	DDI_VGG_S0iX
AM39	--
AM4	USB3_RXP[2]
AM40	DDI_VGG_S0iX
AM41	--
AM42	VSS
AM43	--
AM44	VSS
AM45	--
AM46	RESERVED
AM47	--
AM48	LPDDR3_M1_DQ[19]_A
AM49	--
AM5	--
AM50	LPDDR3_M1_DQ[23]_A
AM51	--
AM52	VSS
AM53	--
AM54	VSS
AM55	--
AM56	VSS
AM57	--
AM58	VSS
AM59	--
AM6	VSS
AM7	--
AM8	USB3_TXP[3]
AM9	--
AN1	PWR_RSVD_V1P05

Ball #	Customer Name - LPDDR3
AN10	--
AN11	USB3_TXP[0]
AN12	--
AN13	--
AN14	--
AN15	--
AN16	--
AN17	--
AN18	--
AN19	--
AN2	--
AN20	--
AN21	--
AN22	--
AN23	--
AN24	--
AN25	--
AN26	--
AN27	--
AN28	--
AN29	--
AN3	VSS
AN30	--
AN31	--
AN32	--
AN33	--
AN34	--
AN35	--
AN36	--
AN37	--
AN38	--
AN39	--
AN4	--
AN40	--
AN41	--
AN42	--
AN43	--
AN44	--
AN45	--
AN46	--
AN47	--
AN48	--
AN49	VSS
AN5	USB3_RXP[0]
AN50	--
AN51	VSS
AN52	--
AN53	VSS
AN54	--
AN55	LPDDR3_M1_DQ[04]_A
AN56	--
AN57	LPDDR3_M1_DQ[00]_A
AN58	--
AN59	VSS
AN6	--
AN7	USB3_TXP[1]
AN8	--
AN9	USB3_TXN[3]
AP1	--
AP10	USB3_TXN[0]
AP11	--
AP12	VSS
AP13	--
AP14	VSS
AP15	--
AP16	VSS
AP17	--
AP18	VSS
AP19	--
AP2	VSS
AP20	UNCORE_VNN_S4
AP21	--
AP22	DDI_VGG_S0iX
AP23	--

Ball #	Customer Name - LPDDR3
AP24	DDI_VGG_S0iX
AP25	--
AP26	DDI_VGG_S0iX
AP27	--
AP28	DDI_VGG_S0iX
AP29	--
AP3	--
AP30	--
AP31	DDI_VGG_S0iX
AP32	--
AP33	DDI_VGG_S0iX
AP34	--
AP35	DDI_VGG_S0iX
AP36	--
AP37	DDI_VGG_S0iX
AP38	--
AP39	DDI_VGG_S0iX
AP4	USB3_RXN[0]
AP40	--
AP41	VSS
AP42	--
AP43	LPDDR3_M1_RCOMPPD
AP44	--
AP45	RESERVED
AP46	--
AP47	--
AP48	LPDDR3_M1_DQ[18]_A
AP49	--
AP5	--
AP50	LPDDR3_M1_DQS[2]_P_A
AP51	--
AP52	LPDDR3_M1_DQ[16]_A
AP53	--
AP54	VSS
AP55	--
AP56	LPDDR3_M1_DM[0]_A
AP57	--

Ball #	Customer Name - LPDDR3
AP58	VSS
AP59	--
AP6	USB3_TXN[1]
AP7	--
AP8	VSS
AP9	--
AR1	PWR_RSVD_V1P05
AR10	--
AR11	VSS
AR12	--
AR13	--
AR14	--
AR15	--
AR16	--
AR17	--
AR18	--
AR19	--
AR2	--
AR20	--
AR21	--
AR22	--
AR23	--
AR24	--
AR25	--
AR26	--
AR27	--
AR28	--
AR29	--
AR3	RESERVED
AR30	--
AR31	--
AR32	--
AR33	--
AR34	--
AR35	--
AR36	--
AR37	--

Ball #	Customer Name - LPDDR3
AR38	--
AR39	--
AR4	--
AR40	--
AR41	--
AR42	--
AR43	--
AR44	--
AR45	--
AR46	--
AR47	--
AR48	--
AR49	LPDDR3_M1_DQ[17]_A
AR5	VSS
AR50	--
AR51	LPDDR3_M1_DQS[2]_N_A
AR52	--
AR53	LPDDR3_M1_DQ[20]_A
AR54	--
AR55	LPDDR3_M1_DQ[03]_A
AR56	--
AR57	LPDDR3_M1_DQS[0]_P_A
AR58	--
AR59	DDR_VDDQG_S4
AR6	--
AR7	VSS
AR8	--
AR9	RESERVED
AT1	--
AT10	PCIE_TXP[1]
AT11	--
AT12	VSS
AT13	--
AT14	--
AT15	VSS
AT16	--
AT17	VSS

Ball #	Customer Name - LPDDR3
AT18	--
AT19	UNCORE_VNN_S4
AT2	RESERVED
AT20	--
AT21	UNCORE_VNN_S4
AT22	--
AT23	DDI_VGG_S0iX
AT24	--
AT25	VSS
AT26	--
AT27	VSS
AT28	--
AT29	DDI_VGG_S0iX
AT3	--
AT30	--
AT31	--
AT32	PWR_RSVD_OBS
AT33	--
AT34	VSS
AT35	--
AT36	VSS
AT37	--
AT38	DDI_VGG_S0iX
AT39	--
AT4	S_RCOMP_P
AT40	DDI_VGG_S0iX
AT41	--
AT42	DDRCH1_VDDQG_S4
AT43	--
AT44	VSS
AT45	--
AT46	VSS
AT47	--
AT48	LPDDR3_M1_DQ[22]_A
AT49	--
AT5	--
AT50	LPDDR3_M1_DQ[21]_A

Ball #	Customer Name - LPDDR3
AT51	--
AT52	VSS
AT53	--
AT54	VSS
AT55	--
AT56	VSS
AT57	--
AT58	LPDDR3_M1_DQS[0]_N_A
AT59	--
AT6	VSS
AT7	--
AT8	RESERVED
AT9	--
AU1	PCIE_V1P05_G3
AU10	--
AU11	PCIE_TXP[0]
AU12	--
AU13	--
AU14	F_V3P3A_G3
AU15	--
AU16	VSS
AU17	--
AU18	F_V1P05A_G3
AU19	--
AU2	--
AU20	VSS
AU21	--
AU22	UNCORE_VNN_S4
AU23	--
AU24	DDI_VGG_S0iX
AU25	--
AU26	DDI_VGG_S0iX
AU27	--
AU28	DDI_VGG_S0iX
AU29	--
AU3	S_RCOMP_N
AU30	--

Ball #	Customer Name - LPDDR3
AU31	DDI_VGG_S0iX
AU32	--
AU33	DDI_VGG_S0iX
AU34	--
AU35	DDI_VGG_S0iX
AU36	--
AU37	DDI_VGG_S0iX
AU38	--
AU39	DDI_VGG_S0iX
AU4	--
AU40	--
AU41	DDRSFRCH1_VDDQG_S4
AU42	--
AU43	VSS
AU44	--
AU45	VSS
AU46	--
AU47	--
AU48	--
AU49	VSS
AU5	PCIE_RXP[1]
AU50	--
AU51	VSS
AU52	--
AU53	VSS
AU54	--
AU55	LPDDR3_M1_DQ[01]_A
AU56	--
AU57	LPDDR3_M1_DQ[02]_A
AU58	--
AU59	DDR_VDDQG_S4
AU6	--
AU7	VSS
AU8	--
AU9	PCIE_TXN[1]
AV1	--
AV10	PCIE_TXN[0]

Ball #	Customer Name - LPDDR3
AV11	--
AV12	ICLK_ICOMP
AV13	--
AV14	--
AV15	--
AV16	--
AV17	--
AV18	--
AV19	--
AV2	VSS
AV20	--
AV21	--
AV22	--
AV23	--
AV24	--
AV25	--
AV26	--
AV27	--
AV28	--
AV29	--
AV3	--
AV30	--
AV31	--
AV32	--
AV33	--
AV34	--
AV35	--
AV36	--
AV37	--
AV38	--
AV39	--
AV4	PCIE_RXN[1]
AV40	--
AV41	--
AV42	--
AV43	--
AV44	--

Ball #	Customer Name - LPDDR3
AV45	--
AV46	--
AV47	--
AV48	LPDDR3_M1_DM[2]_A
AV49	--
AV5	--
AV50	LPDDR3_M1_DM[3]_A
AV51	--
AV52	LPDDR3_M1_DQ[24]_A
AV53	--
AV54	LPDDR3_M1_DQ[05]_A
AV55	--
AV56	LPDDR3_M1_DQ[06]_A
AV57	--
AV58	VSS
AV59	--
AV6	VSS
AV7	--
AV8	VSS
AV9	--
AW1	PCIE_V1P05_G3
AW10	--
AW11	VSS
AW12	--
AW13	--
AW14	--
AW15	VSS
AW16	--
AW17	UNCORE1_V1P05A_G3
AW18	--
AW19	UNCORE1_V1P05A_G3
AW2	--
AW20	--
AW21	UNCORE_VNN_S4
AW22	--
AW23	DDI_VGG_S0iX
AW24	--

Ball #	Customer Name - LPDDR3
AW25	DDI_VGG_SENSE
AW26	--
AW27	DDI_VGG_S0iX
AW28	--
AW29	DDI_VGG_S0iX
AW3	PCIE_RXN[0]
AW30	--
AW31	--
AW32	DDI_VGG_S0iX
AW33	--
AW34	DDI_VGG_S0iX
AW35	--
AW36	DDI_VGG_S0iX
AW37	--
AW38	DDI_VGG_S0iX
AW39	--
AW4	--
AW40	DDI_VGG_S0iX
AW41	--
AW42	DDR_V1P05A_G3
AW43	--
AW44	VSS
AW45	--
AW46	VSS
AW47	--
AW48	--
AW49	VSS
AW5	VSS
AW50	--
AW51	LPDDR3_M1_DQ[29]_A
AW52	--
AW53	LPDDR3_M1_DQ[26]_A
AW54	--
AW55	LPDDR3_M1_DQ[08]_A
AW56	--
AW57	LPDDR3_M1_DQ[07]_A
AW58	--

Ball #	Customer Name - LPDDR3
AW59	DDR_VDDQG_S4
AW6	--
AW7	VSS
AW8	--
AW9	VSS
AY1	--
AY10	VSS
AY11	--
AY12	RESERVED
AY13	--
AY14	--
AY15	--
AY16	--
AY17	--
AY18	--
AY19	--
AY2	PCIE_RCOMP_P
AY20	--
AY21	--
AY22	--
AY23	--
AY24	--
AY25	--
AY26	--
AY27	--
AY28	--
AY29	--
AY3	--
AY30	--
AY31	--
AY32	--
AY33	--
AY34	--
AY35	--
AY36	--
AY37	--
AY38	--

Ball #	Customer Name - LPDDR3
AY39	--
AY4	PCIE_RXP[0]
AY40	--
AY41	--
AY42	--
AY43	--
AY44	--
AY45	--
AY46	--
AY47	--
AY48	LPDDR3_M1_DQ[27]_A
AY49	--
AY5	--
AY50	LPDDR3_M1_DQS[3]_P_A
AY51	--
AY52	LPDDR3_M1_DQ[25]_A
AY53	--
AY54	LPDDR3_M1_DQ[11]_A
AY55	--
AY56	VSS
AY57	--
AY58	LPDDR3_M1_DQ[09]_A
AY59	--
AY6	VSS
AY7	--
AY8	ICLK_OSCIN
AY9	--
B1	--
B10	MIPI_V1P2A_G3
B11	--
B12	MDSI_A_CLKP
B13	--
B14	GPION_V1P8A_G3
B15	--
B16	MCSI_2_DP[0]
B17	--
B18	CORE_VCC0_S0iX

Ball #	Customer Name - LPDDR3
B19	--
B2	VSS
B20	MCSI_1_DN[1]
B21	--
B22	CORE_VCC0_S0iX
B23	--
B24	CORE_VCC1_S0iX
B25	--
B26	CORE_VCC1_S0iX
B27	--
B28	GPIO_CAMERASB03
B29	--
B3	--
B30	--
B31	--
B32	VSS
B33	--
B34	JTAG_TMS
B35	--
B36	CORE_V1P15_S0iX
B37	--
B38	GPIO_N3/C0_BPM1_TX/ C1_BPM1_TX
B39	--
B4	VSS
B40	DDR_V1P05A_G3
B41	--
B42	LPDDR3_M0_DQ[17]_A
B43	--
B44	VSS
B45	--
B46	LPDDR3_M0_DQS[2]_P_A
B47	--
B48	VSS
B49	--
B5	--
B50	LPDDR3_M0_DQS[3]_P_A
B51	--

Ball #	Customer Name - LPDDR3
B52	VSS
B53	--
B54	LPDDR3_M0_ODT_A
B55	--
B56	LPDDR3_M0_CS[1]_N
B57	--
B58	VSS
B59	--
B6	MDSI_C_CLKP
B7	--
B8	VSS
B9	--
BA1	VSS
BA10	--
BA11	VSS
BA12	--
BA13	--
BA14	VSS
BA15	--
BA16	F_V1P05A_G3
BA17	--
BA18	F_V1P05A_G3
BA19	--
BA2	--
BA20	UNCORE_VNN_S4
BA21	--
BA22	DDI_VGG_S0iX
BA23	--
BA24	DDI_VSS_SENSE
BA25	--
BA26	VSS
BA27	--
BA28	VSS
BA29	--
BA3	PCIE_RCOMP_N
BA30	--
BA31	DDI_VGG_S0iX

Ball #	Customer Name - LPDDR3
BA32	--
BA33	DDI_VGG_S0iX
BA34	--
BA35	VSS
BA36	--
BA37	VSS
BA38	--
BA39	DDI_VGG_S0iX
BA4	--
BA40	--
BA41	VSS
BA42	--
BA43	VSS
BA44	--
BA45	VSS
BA46	--
BA47	--
BA48	--
BA49	LPDDR3_M1_DQ[31]_A
BA5	PCIE_REFCLK[0]_P
BA50	--
BA51	LPDDR3_M1_DQS[3]_N_A
BA52	--
BA53	VSS
BA54	--
BA55	LPDDR3_M1_DQS[1]_P_A
BA56	--
BA57	LPDDR3_M1_DQ[10]_A
BA58	--
BA59	DDR_VDDQG_S4
BA6	--
BA7	VSS
BA8	--
BA9	ICLK_OSCOUT
BB1	--
BB10	VSS
BB11	--

Ball #	Customer Name - LPDDR3
BB12	ICLK_RCOMP
BB13	--
BB14	--
BB15	RTC_V3P3RTC_G5
BB16	--
BB17	F_V1P8A_G3
BB18	--
BB19	VSS
BB2	ICLK_VSFR_G3
BB20	--
BB21	UNCORE_VNN_S4
BB22	--
BB23	DDI_VGG_S0iX
BB24	--
BB25	DDI_VGG_S0iX
BB26	--
BB27	DDI_VGG_S0iX
BB28	--
BB29	DDI_VGG_S0iX
BB3	--
BB30	--
BB31	--
BB32	DDI_VGG_S0iX
BB33	--
BB34	DDI_VGG_S0iX
BB35	--
BB36	DDI_VGG_S0iX
BB37	--
BB38	DDI_VGG_S0iX
BB39	--
BB4	PCIE_REFCLK[0]_N
BB40	DDI_VGG_S0iX
BB41	--
BB42	VSS
BB43	--
BB44	VSS
BB45	--

Ball #	Customer Name - LPDDR3
BB46	VSS
BB47	--
BB48	LPDDR3_M1_DQ[30]_A
BB49	--
BB5	--
BB50	LPDDR3_M1_DQ[28]_A
BB51	--
BB52	VSS
BB53	--
BB54	VSS
BB55	--
BB56	LPDDR3_M1_DQS[1]_N_A
BB57	--
BB58	VSS
BB59	--
BB6	VSS
BB7	--
BB8	VSSA
BB9	--
BC1	ICLK_VSFR_G3
BC10	--
BC11	VSS
BC12	--
BC13	--
BC14	--
BC15	--
BC16	--
BC17	--
BC18	--
BC19	--
BC2	--
BC20	--
BC21	--
BC22	--
BC23	--
BC24	--
BC25	--

Ball #	Customer Name - LPDDR3
BC26	--
BC27	--
BC28	--
BC29	--
BC3	PCIE_REFCLK[1]_P
BC30	--
BC31	--
BC32	--
BC33	--
BC34	--
BC35	--
BC36	--
BC37	--
BC38	--
BC39	--
BC4	--
BC40	--
BC41	--
BC42	--
BC43	--
BC44	--
BC45	--
BC46	--
BC47	--
BC48	--
BC49	VSS
BC5	VSS
BC50	--
BC51	VSS
BC52	--
BC53	VSS
BC54	--
BC55	LPDDR3_M1_DM[1]_A
BC56	--
BC57	LPDDR3_M1_DQ[14]_A
BC58	--
BC59	VSS

Ball #	Customer Name - LPDDR3
BC6	--
BC7	VSS
BC8	--
BC9	RESERVED
BD1	--
BD10	RESERVED
BD11	--
BD12	VSS
BD13	--
BD14	RTC_V3P3A_G5
BD15	--
BD16	VSS
BD17	--
BD18	VSS
BD19	--
BD2	PCIE_REFCLK[1]_N
BD20	UNCORE_VNN_S4
BD21	--
BD22	DDI_VGG_S0iX
BD23	--
BD24	VSS
BD25	--
BD26	DDI_VGG_S0iX
BD27	--
BD28	GPIOSE_V1P8A_G3
BD29	--
BD3	--
BD30	--
BD31	GPIOSE_V1P8A_G3
BD32	--
BD33	DDI_VGG_S0iX
BD34	--
BD35	DDI_VGG_S0iX
BD36	--
BD37	DDI_VGG_S0iX
BD38	--
BD39	DDI_VGG_S0iX

Ball #	Customer Name - LPDDR3
BD4	RTC_TEST_N
BD40	--
BD41	VSS
BD42	--
BD43	VSS
BD44	--
BD45	VSS
BD46	--
BD47	--
BD48	LPDDR3_M1_OCAVREF
BD49	--
BD5	--
BD50	LPDDR3_M1_ODQVREF
BD51	--
BD52	LPDDR3_M1_CA[08]
BD53	--
BD54	LPDDR3_M1_DQ[15]_A
BD55	--
BD56	VSS
BD57	--
BD58	LPDDR3_M1_DQ[12]
BD59	--
BD6	VSS
BD7	--
BD8	RESERVED
BD9	--
BE1	VSS
BE10	--
BE11	VSS
BE12	--
BE13	--
BE14	--
BE15	--
BE16	--
BE17	--
BE18	--
BE19	--

Ball #	Customer Name - LPDDR3
BE2	--
BE20	--
BE21	--
BE22	--
BE23	--
BE24	--
BE25	--
BE26	--
BE27	--
BE28	--
BE29	--
BE3	PMC_RSMRST_N
BE30	--
BE31	--
BE32	--
BE33	--
BE34	--
BE35	--
BE36	--
BE37	--
BE38	--
BE39	--
BE4	--
BE40	--
BE41	--
BE42	--
BE43	--
BE44	--
BE45	--
BE46	--
BE47	--
BE48	--
BE49	RESERVED
BE5	RESERVED
BE50	--
BE51	RESERVED
BE52	--

Ball #	Customer Name - LPDDR3
BE53	LPDDR3_M1_CA[09]
BE54	--
BE55	LPDDR3_M1_DQ[13]_A
BE56	--
BE57	VSS
BE58	--
BE59	VSS
BE6	--
BE7	VSS
BE8	--
BE9	RESERVED
BF1	--
BF10	RTC_X2
BF11	--
BF12	VSS
BF13	--
BF14	--
BF15	VSS
BF16	--
BF17	VSS
BF18	--
BF19	VSS
BF2	PCIeCLK_V1P05A_G3
BF20	--
BF21	VSS
BF22	--
BF23	SDIO_V3P3A_V1P8A_G3
BF24	--
BF25	LPC_V3P3A_V1P8A_S4
BF26	--
BF27	VSS
BF28	--
BF29	GPIOSE_V1P8A_G3
BF3	--
BF30	--
BF31	--
BF32	VSS

Ball #	Customer Name - LPDDR3
BF33	--
BF34	VSS
BF35	--
BF36	VSS
BF37	--
BF38	PWR_RSVD_OBS
BF39	--
BF4	RTC_RST_N
BF40	VSS
BF41	--
BF42	VSS
BF43	--
BF44	VSS
BF45	--
BF46	VSS
BF47	--
BF48	LPDDR3_M1_CA[02]
BF49	--
BF5	--
BF50	RESERVED
BF51	--
BF52	VSS
BF53	--
BF54	LPDDR3_M1_CK_N_A
BF55	--
BF56	LPDDR3_M1_CK_P_A
BF57	--
BF58	VSS
BF59	--
BF6	VSS
BF7	--
BF8	RESERVED
BF9	--
BG1	PCIeCLK_V1P05A_G3
BG10	--
BG11	RTC_EXTPAD
BG12	--

Ball #	Customer Name - LPDDR3
BG13	--
BG14	--
BG15	--
BG16	--
BG17	--
BG18	--
BG19	--
BG2	--
BG20	--
BG21	--
BG22	--
BG23	--
BG24	--
BG25	--
BG26	--
BG27	--
BG28	--
BG29	--
BG3	PMC_CORE_PWROK
BG30	--
BG31	--
BG32	--
BG33	--
BG34	--
BG35	--
BG36	--
BG37	--
BG38	--
BG39	--
BG4	--
BG40	--
BG41	--
BG42	--
BG43	--
BG44	--
BG45	--
BG46	--

Ball #	Customer Name - LPDDR3
BG47	--
BG48	--
BG49	VSS
BG5	VSS
BG50	--
BG51	LPDDR3_M1_CA[06]
BG52	--
BG53	VSS
BG54	--
BG55	LPDDR3_M1_CK_N_B
BG56	--
BG57	LPDDR3_M1_CK_P_B
BG58	--
BG59	DDR_VDDQG_S4
BG6	--
BG7	VSS
BG8	--
BG9	RTC_X1
BH1	--
BH10	PMC_PWRBTN_N
BH11	--
BH12	VSS
BH13	--
BH14	MMC1_RCOMP
BH15	--
BH16	SD3_RCOMP
BH17	--
BH18	SD3_D[2]
BH19	--
BH2	VSS
BH20	VSS
BH21	--
BH22	LPC_RCOMP
BH23	--
BH24	PMC_RSTBTN_N
BH25	--
BH26	UART2_RTS_N

Ball #	Customer Name - LPDDR3
BH27	--
BH28	UART2_DATAIN
BH29	--
BH3	--
BH30	--
BH31	--
BH32	LPE_I2S0_CLK
BH33	--
BH34	I2C6_CLK/NMI_N
BH35	--
BH36	I2C2_DATA
BH37	--
BH38	VSS
BH39	--
BH4	PMC_BATLOW_N
BH40	LPE_I2S2_FRM
BH41	--
BH42	VSS
BH43	--
BH44	LPDDR3_M1_DQ[09]_B
BH45	--
BH46	LPDDR3_M1_DM[1]_B
BH47	--
BH48	RESERVED
BH49	--
BH5	--
BH50	LPDDR3_M1_CA[01]
BH51	--
BH52	VSS
BH53	--
BH54	LPDDR3_M1_CA[04]
BH55	--
BH56	VSS
BH57	--
BH58	VSS
BH59	--
BH6	PMC_SUS_STAT_N

Ball #	Customer Name - LPDDR3
BH7	--
BH8	VSS
BH9	--
BJ1	ICLK_VSFR_G3
BJ10	--
BJ11	VSS
BJ12	--
BJ13	VSS
BJ14	--
BJ15	MMC1_CMD
BJ16	--
BJ17	VSS
BJ18	--
BJ19	LPC_FRAME_N/UART0_DATAIN/ SPI2_MISO
BJ2	--
BJ20	--
BJ21	GPIO_ALERT/ISH_GPIO[11]/ ISH_UART_DATAIN
BJ22	--
BJ23	VSS
BJ24	--
BJ25	FST_SPI_D[2]
BJ26	--
BJ27	VSS
BJ28	--
BJ29	--
BJ3	PMC_SLP_S3_N
BJ30	LPE_I2S1_DATAIN
BJ31	--
BJ32	--
BJ33	NFC_I2C_CLK
BJ34	--
BJ35	VSS
BJ36	--
BJ37	UART0_DATAIN
BJ38	--
BJ39	VSS

Ball #	Customer Name - LPDDR3
BJ4	--
BJ40	--
BJ41	VSS
BJ42	--
BJ43	LPDDR3_M1_DQ[14]_B
BJ44	--
BJ45	VSS
BJ46	--
BJ47	LPDDR3_M1_DQ[07]_B
BJ48	--
BJ49	LPDDR3_M1_CA[00]
BJ5	PMC_PLTRST_N
BJ50	--
BJ51	LPDDR3_M1_CA[03]
BJ52	--
BJ53	RESERVED
BJ54	--
BJ55	LPDDR3_M1_CA[05]
BJ56	--
BJ57	LPDDR3_M1_CA[07]
BJ58	--
BJ59	DDR_VDDQG_S4
BJ6	--
BJ7	PMC_WAKE_N
BJ8	--
BJ9	PMC_SLP_S4_N
BK1	--
BK10	ISH_GPIO[6]/I2S4_DATAOUT
BK11	--
BK12	MMC1_D[3]
BK13	--
BK14	MMC1_D[1]
BK15	--
BK16	SD3_D[0]
BK17	--
BK18	SPI1_MOSI
BK19	--

Ball #	Customer Name - LPDDR3
BK2	VSS
BK20	LPC_CLKOUT[0]/ISH_GPIO[10]/ ISH_UART_DATAOUT
BK21	--
BK22	PMC_SUSPWRDNACK
BK23	--
BK24	VSS
BK25	--
BK26	FST_SPI_D[1]
BK27	--
BK28	UART2_DATAOUT
BK29	--
BK3	--
BK30	--
BK31	--
BK32	LPE_I2S0_FRM
BK33	--
BK34	I2C5_CLK
BK35	--
BK36	SPI3_MOSI
BK37	--
BK38	LPE_I2S2_CLK
BK39	--
BK4	PMC_ACPRESENT
BK40	LPE_I2S2_DATAOUT
BK41	--
BK42	LPDDR3_M1_DQ[15]_B
BK43	--
BK44	LPDDR3_M1_DQ[10]_B
BK45	--
BK46	LPDDR3_M1_DQS[1]_P_B
BK47	--
BK48	VSS
BK49	--
BK5	--
BK50	VSS
BK51	--
BK52	LPDDR3_M1_DQ[03]_B

Ball #	Customer Name - LPDDR3
BK53	--
BK54	VSS
BK55	--
BK56	LPDDR3_M1_CKE[0]_A
BK57	--
BK58	VSS
BK59	--
BK6	VSS
BK7	--
BK8	ISH_GPIO[9]/ISH_SPI_MISO/ I2S5_FS
BK9	--
BL1	UNCORE_VNN_S4
BL10	--
BL11	SD2_CLK
BL12	--
BL13	VSS
BL14	--
BL15	SD3_D[3]
BL16	--
BL17	SD3_CLK
BL18	--
BL19	SPI1_MISO
BL2	--
BL20	--
BL21	LPC_CLKRUN_N/ UART0_DATAOUT/SPI2_CLK
BL22	--
BL23	VSS
BL24	--
BL25	FST_SPI_D[3]
BL26	--
BL27	UART2_CTS_N
BL28	--
BL29	--
BL3	PMC_SLP_S0IX_N
BL30	VSS
BL31	--

Ball #	Customer Name - LPDDR3
BL32	--
BL33	I2C6_DATA/SD3_WP
BL34	--
BL35	I2C2_CLK
BL36	--
BL37	UART0_DATAOUT/SPI3_CLK
BL38	--
BL39	LPE_I2S2_DATAIN
BL4	--
BL40	--
BL41	VSS
BL42	--
BL43	LPDDR3_M1_DQ[11]_B
BL44	--
BL45	LPDDR3_M1_DQ[08]_B
BL46	--
BL47	LPDDR3_M1_DQS[1]_N_B
BL48	--
BL49	LPDDR3_M1_DQ[01]_B
BL5	VSS
BL50	--
BL51	LPDDR3_M1_DQ[00]_B
BL52	--
BL53	LPDDR3_M1_DQ[02]_B
BL54	--
BL55	RESERVED
BL56	--
BL57	LPDDR3_M1_CS[0]_N
BL58	--
BL59	DDR_VDDQG_S4
BL6	--
BL7	VSS
BL8	--
BL9	ISH_GPIO[8]/ISH_SPI_CS[0]_N/ I2S5_CLK
BM1	--
BM10	VSS
BM11	--

Ball #	Customer Name - LPDDR3
BM12	MMC1_D[0]
BM13	--
BM14	MMC1_D[2]
BM15	--
BM16	VSS
BM17	--
BM18	VSS
BM19	--
BM2	ISH_GPIO[7]/I2S4_DATAIN
BM20	LPC_CLKOUT[1]/ISH_GPIO[11]/ ISH_UART_DATAIN
BM21	--
BM22	VSS
BM23	--
BM24	LPC_SERIRQ/SPI2_CS[0]_N
BM25	--
BM26	VSS
BM27	--
BM28	VSS
BM29	--
BM3	--
BM30	--
BM31	--
BM32	LPE_I2S0_DATAOUT
BM33	--
BM34	VSS
BM35	--
BM36	VSS
BM37	--
BM38	SPI3_CS[0]_N
BM39	--
BM4	ISH_GPIO[3]/I2S3_DATAIN
BM40	VSS
BM41	--
BM42	LPDDR3_M1_DQ[13]_B
BM43	--
BM44	VSS
BM45	--

Ball #	Customer Name - LPDDR3
BM46	VSS
BM47	--
BM48	LPDDR3_M1_DQ[06]_B
BM49	--
BM5	--
BM50	LPDDR3_M1_DQS[0]_P_B
BM51	--
BM52	LPDDR3_M1_DQ[04]_B
BM53	--
BM54	RESERVED
BM55	--
BM56	VSS
BM57	--
BM58	LPDDR3_M1_CKE[0]_B
BM59	--
BM6	PMC_SUSCLK[0]
BM7	--
BM8	ISH_I2C1_DATA/ISH_SPI_MOSI/ I2S5_DATAOUT
BM9	--
BN1	UNCORE_VNN_S4
BN10	--
BN11	SD2_CMD
BN12	--
BN13	VSS
BN14	--
BN15	MMC1_CLK
BN16	--
BN17	VSS
BN18	--
BN19	SPI1_CLK
BN2	--
BN20	--
BN21	VSS
BN22	--
BN23	VSS
BN24	--
BN25	FST_SPI_D[0]

Ball #	Customer Name - LPDDR3
BN26	--
BN27	VSS
BN28	--
BN29	--
BN3	ISH_I2C1_CLK/ISH_SPI_CLK/ I2S5_DATAIN
BN30	VSS
BN31	--
BN32	--
BN33	NFC_I2C_DATA
BN34	--
BN35	VSS
BN36	--
BN37	SPI3_MISO
BN38	--
BN39	VSS
BN4	--
BN40	--
BN41	VSS
BN42	--
BN43	LPDDR3_M1_DQ[12]_B
BN44	--
BN45	VSS
BN46	--
BN47	LPDDR3_M1_DM[0]_B
BN48	--
BN49	LPDDR3_M1_DQ[05]_B
BN5	ISH_GPIO[1]/I2S3_FS
BN50	--
BN51	LPDDR3_M1_DQS[0]_N_B
BN52	--
BN53	VSS
BN54	--
BN55	RESERVED
BN56	--
BN57	RESERVED
BN58	--
BN59	DDR_VDDQG_S4

Ball #	Customer Name - LPDDR3
BN6	--
BN7	VSS
BN8	--
BN9	RESERVED
BP1	--
BP10	VSS
BP11	--
BP12	SD2_D[3]_CD_N
BP13	--
BP14	MMC1_D[6]
BP15	--
BP16	SD3_D[1]
BP17	--
BP18	VSS
BP19	--
BP2	UNCORE_VNN_S4
BP20	LPC_AD[2]/ISH_GPIO[14]/ ISH_I2C0_DATA
BP21	--
BP22	USB_OC[0]_N
BP23	--
BP24	FST_SPI_CLK
BP25	--
BP26	VSS
BP27	--
BP28	LPE_I2S1_DATAOUT
BP29	--
BP3	--
BP30	--
BP31	--
BP32	VSS
BP33	--
BP34	I2C4_CLK/DDI0_DDC_CLK/ DDI2_DDC_CLK/MDSI_DDC_CLK
BP35	--
BP36	I2C1_CLK
BP37	--
BP38	GPIO_SW93

Ball #	Customer Name - LPDDR3
BP39	--
BP4	ISH_GPIO[5]/I2S4_FS
BP40	VSS
BP41	--
BP42	LPDDR3_M1_DQ[28]_B
BP43	--
BP44	LPDDR3_M1_DQ[31]_B
BP45	--
BP46	LPDDR3_M1_DM[3]_B
BP47	--
BP48	VSS
BP49	--
BP5	--
BP50	LPDDR3_M1_DQ[19]_B
BP51	--
BP52	LPDDR3_M1_DQ[16]_B
BP53	--
BP54	RESERVED
BP55	--
BP56	RESERVED
BP57	--
BP58	VSS
BP59	--
BP6	VSS
BP7	--
BP8	PMC_PLT_CLK[0]/ISH_GPIO[10]/ ISH_UART_DATAOUT
BP9	--
BR1	VSS
BR10	--
BR11	SD2_D[1]
BR12	--
BR13	MMC1_D[4]
BR14	--
BR15	SD3_CMD
BR16	--
BR17	SPI1_CS[1]_N
BR18	--

Ball #	Customer Name - LPDDR3
BR19	LPC_AD[1]/ISH_GPIO[13]/ ISH_UART_RTS_N
BR2	--
BR20	--
BR21	LPC_AD[3]/ISH_GPIO[15]/ ISH_I2C0_CLK/SPI2_MOSI
BR22	--
BR23	FST_SPI_CS[1]_N
BR24	--
BR25	UART1_RTS_N
BR26	--
BR27	UART1_CTS_N
BR28	--
BR29	--
BR3	ISH_GPIO[0]/I2S3_CLK
BR30	LPE_I2S1_CLK
BR31	--
BR32	--
BR33	I2C5_DATA
BR34	--
BR35	I2C1_DATA
BR36	--
BR37	ISH_GPIO[12]/ISH_UART_CTS_N
BR38	--
BR39	PCIE_CLKREQ[0]_N
BR4	--
BR40	--
BR41	VSS
BR42	--
BR43	LPDDR3_M1_DQ[29]_B
BR44	--
BR45	LPDDR3_M1_DQ[27]_B
BR46	--
BR47	LPDDR3_M1_DQ[23]_B
BR48	--
BR49	LPDDR3_M1_DM[2]_B
BR5	VSS
BR50	--

Ball #	Customer Name - LPDDR3
BR51	LPDDR3_M1_DQ[18]_B
BR52	--
BR53	VSS
BR54	--
BR55	RESERVED
BR56	--
BR57	RESERVED
BR58	--
BR59	VSS
BR6	--
BR7	PMC_PLT_CLK[4]/ISH_GPIO[14]/ ISH_I2C0_DATA/SPI2_MISO
BR8	--
BR9	PMC_PLT_CLK[1]/ISH_GPIO[11]/ ISH_UART_DATAIN/SPI2_CS[1]_N
BT1	--
BT10	SD2_D[0]
BT11	--
BT12	VSS
BT13	--
BT14	MMC1_D[7]
BT15	--
BT16	VSS
BT17	--
BT18	SPI1_CS[0]_N
BT19	--
BT2	ISH_GPIO[2]/I2S3_DATAOUT
BT20	VSS
BT21	--
BT22	SD3_1P8_EN
BT23	--
BT24	VSS
BT25	--
BT26	UART1_DATAIN/UART0_DATAIN
BT27	--
BT28	VSS
BT29	--
BT3	--

Ball #	Customer Name - LPDDR3
BT30	--
BT31	--
BT32	I2C4_DATA/DDI2_DDC_DATA/ DDI2_DDC_DATA/ MDSI_DDC_DATA
BT33	--
BT34	VSS
BT35	--
BT36	I2C0_CLK
BT37	--
BT38	VSS
BT39	--
BT4	ISH_GPIO[4]/I2S4_CLK
BT40	SD3_WP
BT41	--
BT42	VSS
BT43	--
BT44	LPDDR3_M1_DQ[26]_B
BT45	--
BT46	VSS
BT47	--
BT48	LPDDR3_M1_DQ[21]_B
BT49	--
BT5	--
BT50	VSS
BT51	--
BT52	LPDDR3_M1_DQ[20]_B
BT53	--
BT54	VSS
BT55	--
BT56	VSS
BT57	--
BT58	LPDDR3_M1_CKE[1]_A
BT59	--
BT6	PMC_PLT_CLK[5]/ISH_GPIO[15]/ ISH_I2C0_CLK/SPI2_MOSI
BT7	--
BT8	VSS

Ball #	Customer Name - LPDDR3
BT9	--
BU1	VSS
BU10	--
BU11	SD2_D[2]
BU12	--
BU13	MMC1_RCLK/MMC1_RESET_N
BU14	--
BU15	MMC1_D[5]
BU16	--
BU17	SD3_CD_N
BU18	--
BU19	LPC_AD[0]/ISH_GPIO[12]/ ISH_UART_CTS_N
BU2	--
BU20	--
BU21	USB_OC[1]_N
BU22	--
BU23	SD3_PWREN_N
BU24	--
BU25	FST_SPI_CS[0]_N
BU26	--
BU27	UART1_DATAOUT/ UART0_DATAOUT
BU28	--
BU29	--
BU3	VSS
BU30	LPE_I2S0_DATAIN
BU31	--
BU32	--
BU33	I2C3_DATA
BU34	--
BU35	I2C0_DATA
BU36	--
BU37	GPIO_SW78
BU38	--
BU39	PCIE_CLKREQ[1]_N
BU4	--
BU40	--

Ball #	Customer Name - LPDDR3
BU41	VSS
BU42	--
BU43	LPDDR3_M1_DQ[25]_B
BU44	--
BU45	LPDDR3_M1_DQ[24]_B
BU46	--
BU47	LPDDR3_M1_DQS[3]_N_B
BU48	--
BU49	LPDDR3_M1_DQ[22]_B
BU5	PWM[1]/ISH_GPIO[10]/ ISH_UART_DATAOUT
BU50	--
BU51	LPDDR3_M1_DQS[2]_N_B
BU52	--
BU53	LPDDR3_M1_DQ[17]_B
BU54	--
BU55	LPDDR3_M1_ODT_A
BU56	--
BU57	LPDDR3_M1_CKE[1]_B
BU58	--
BU59	VSS
BU6	--
BU7	PMC_PLT_CLK[2]/ISH_GPIO[12]/ ISH_UART_CTS_N/SPI2_CS[0]_N
BU8	--
BU9	PMC_PLT_CLK[3]/ISH_GPIO[13]/ ISH_UART_RTS_N/SPI2_CLK
BV1	--
BV10	DDI_VGG_S0iX
BV11	--
BV12	VSS
BV13	--
BV14	DDI_VGG_S0iX
BV15	--
BV16	VSS
BV17	--
BV18	LPC_V3P3A_V1P8A_S4
BV19	--

Ball #	Customer Name - LPDDR3
BV2	VSS
BV20	VSS
BV21	--
BV22	GPIOSE_V1P8A_G3
BV23	--
BV24	FST_SPI_CS[2]_N
BV25	--
BV26	GPIOSE_V1P8A_G3
BV27	--
BV28	LPE_I2S1_FRM
BV29	--
BV3	--
BV30	--
BV31	--
BV32	UNCORE_V1P8A_G3
BV33	--
BV34	I2C3_CLK
BV35	--
BV36	DDI_V1P15_S0iX
BV37	--
BV38	MMC1_RESET_N /SPI3_CS[1]_N
BV39	--
BV4	PWM[0]
BV40	DDR_V1P05A_G3
BV41	--
BV42	LPDDR3_M1_DQ[30]_B
BV43	--
BV44	VSS
BV45	--
BV46	LPDDR3_M1_DQS[3]_P_B
BV47	--
BV48	VSS
BV49	--
BV5	--
BV50	LPDDR3_M1_DQS[2]_P_B
BV51	--
BV52	VSS

Ball #	Customer Name - LPDDR3
BV53	--
BV54	LPDDR3_M1_ODT_B
BV55	--
BV56	LPDDR3_M1_CS[1]_N
BV57	--
BV58	VSS
BV59	--
BV6	VSS
BV7	--
BV8	VSS
BV9	--
BW1	--
BW10	--
BW11	DDI_VGG_S0iX
BW12	--
BW13	DDI_VGG_S0iX
BW14	--
BW15	SDIO_V3P3A_V1P8A_G3
BW16	--
BW17	LPC_V3P3A_V1P8A_S4
BW18	--
BW19	VSS
BW2	--
BW20	--
BW21	GPIOSE_V1P8A_G3
BW22	--
BW23	GPIOSE_V1P8A_G3
BW24	--
BW25	GPIOSE_V1P8A_G3
BW26	--
BW27	GPIOSE_V1P8A_G3
BW28	--
BW29	--
BW3	VSS
BW30	VSS
BW31	--
BW32	--

Ball #	Customer Name - LPDDR3
BW33	UNCORE_V1P8A_G3
BW34	--
BW35	DDI_V1P15_S0iX
BW36	--
BW37	DDI_V1P15_S0iX
BW38	--
BW39	VSS
BW4	--
BW40	--
BW41	DDR_V1P05A_G3
BW42	--
BW43	DDR_V1P05A_G3
BW44	--
BW45	VSS
BW46	--
BW47	VSS
BW48	--
BW49	VSS
BW5	VSS
BW50	--
BW51	VSS
BW52	--
BW53	VSS
BW54	--
BW55	VSS
BW56	--
BW57	VSS
BW58	--
BW59	--
BW6	--
BW7	VSS
BW8	--
BW9	DDI_VGG_S0iX
C1	VSS
C10	--
C11	MDSI_A_CLKN
C12	--
C13	RESERVED
C14	--
C15	MCSI_2_DN[0]
C16	--
C17	MCSI_1_CLKP
C18	--
C19	MCSI_1_DP[1]
C2	--
C20	--
C21	DDI2_HPD
C22	--
C23	CORE_VCC1_S0iX
C24	--
C25	VSS
C26	--
C27	GPIO_CAMERASB07
C28	--
C29	--
C3	DDI1_TXP[3]
C30	GPIO_CAMERASB04
C31	--
C32	--
C33	SVID_ALERT_N
C34	--
C35	GPIO_SUS5/PMC_SUSCLK[1]
C36	--
C37	GPIO_SUS3/JTAG2_TDI
C38	--
C39	ISH_GPIO[13]/C0_BPM2_TX/ C1_BPM2_TX
C4	--
C40	--
C41	VSS
C42	--
C43	LPDDR3_M0_DQ[22]_A
C44	--
C45	LPDDR3_M0_DQ[23]_A
C46	--

Ball #	Customer Name - LPDDR3
C47	LPDDR3_M0_DQS[2]_N_A
C48	--
C49	LPDDR3_M0_DQ[25]_A
C5	MDSI_C_CLKN
C50	--
C51	LPDDR3_M0_DQS[3]_N_A
C52	--
C53	LPDDR3_M0_DQ[30]_A
C54	--
C55	LPDDR3_M0_ODT_B
C56	--
C57	LPDDR3_M0_CKE[1]_A
C58	--
C59	VSS
C6	--
C7	MDSI_C_DP[2]
C8	--
C9	MDSI_C_DP[0]
D1	--
D10	MDSI_A_DN[1]
D11	--
D12	RESERVED
D13	--
D14	RESERVED
D15	--
D16	MCSI_1_CLKN
D17	--
D18	MCSI_1_DN[2]
D19	--
D2	DDI1_TXP[2]
D20	VSS
D21	--
D22	VSS
D23	--
D24	CORE_VCC1_S0iX
D25	--
D26	GPIO_CAMERASB10
D27	--
D28	VSS
D29	--
D3	--
D30	--
D31	--
D32	SVID_DATA
D33	--
D34	VSS
D35	--
D36	GPIO_SUS6/PMC_SUSCLK[2]
D37	--
D38	VSS
D39	--
D4	DDI1_TXN[3]
D40	GPIO_N0/C0_BPM0_TX/ C1_BPM0_TX
D41	--
D42	VSS
D43	--
D44	LPDDR3_M0_DQ[21]_A
D45	--
D46	VSS
D47	--
D48	LPDDR3_M0_DQ[26]_A
D49	--
D5	--
D50	VSS
D51	--
D52	LPDDR3_M0_DQ[27]_A
D53	--
D54	VSS
D55	--
D56	VSS
D57	--
D58	LPDDR3_M0_CKE[1]_B
D59	--
D6	MDSI_C_DN[2]

Ball #	Customer Name - LPDDR3
D7	--
D8	MDSI_C_DN[0]
D9	--
E1	VSS
E10	--
E11	VSS
E12	--
E13	RESERVED
E14	--
E15	VSS
E16	--
E17	MCSI_1_DP[2]
E18	--
E19	VSS
E2	--
E20	--
E21	DDI2_DDC_CLK/DDI1_DDC_CLK/ UART0_DATAOUT/ MDSI_DDC_CLK/MDSI_A_TE
E22	--
E23	CORE_VCC1_S0iX
E24	--
E25	DDI1_HPD
E26	--
E27	GPIO_CAMERASB06
E28	--
E29	--
E3	DDI1_TXN[2]
E30	GPIO_CAMERASB02
E31	--
E32	--
E33	JTAG_TDI
E34	--
E35	JTAG_TRST_N
E36	--
E37	GPIO_SUS0
E38	--

Ball #	Customer Name - LPDDR3
E39	GPIO_N1/C0_BPM3_TX/ C1_BPM3_TX
E4	--
E40	--
E41	VSS
E42	--
E43	LPDDR3_M0_DQ[18]_A
E44	--
E45	LPDDR3_M0_DQ[20]_A
E46	--
E47	LPDDR3_M0_DQ[24]_A
E48	--
E49	LPDDR3_M0_DM[03]_A
E5	DDI1_TXN[1]
E50	--
E51	LPDDR3_M0_DQ[29]_A
E52	--
E53	VSS
E54	--
E55	RESERVED
E56	--
E57	RESERVED
E58	--
E59	VSS
E6	--
E7	VSS
E8	--
E9	MDSI_A_DP[1]
F1	--
F10	VSS
F11	--
F12	MDSI_A_DP[2]
F13	--
F14	VSS
F15	--
F16	VSS
F17	--
F18	VSS

Ball #	Customer Name - LPDDR3
F19	--
F2	VSS
F20	DDI2_DDC_DATA/ DDI1_DDC_DATA/UART0_DATAIN/ MDSI_DDC_DATA/MDSI_C_TE
F21	--
F22	DDI0_VDDEN
F23	--
F24	CORE_VCC1_S0iX
F25	--
F26	DDI0_DDC_CLK/DDI1_DDC_CLK/ MDSI_DDC_CLK
F27	--
F28	GPIO_CAMERASB11
F29	--
F3	--
F30	--
F31	--
F32	SVID_CLK
F33	--
F34	JTAG_TCK
F35	--
F36	GPIO_SUS8
F37	--
F38	GPIO_N2/C0_BPM2_TX/ C1_BPM2_TX
F39	--
F4	DDI1_TXP[1]
F40	VSS
F41	--
F42	LPDDR3_M0_DQ[19]_A
F43	--
F44	LPDDR3_M0_DQ[16]_A
F45	--
F46	LPDDR3_M0_DM[2]_A
F47	--
F48	VSS
F49	--
F5	--

Ball #	Customer Name - LPDDR3
F50	LPDDR3_M0_DQ[28]_A
F51	--
F52	LPDDR3_M0_DQ[31]_A
F53	--
F54	RESERVED
F55	--
F56	RESERVED
F57	--
F58	VSS
F59	--
F6	VSS
F7	--
F8	VSS
F9	--
G1	DDI2_VDDQ_G3
G10	--
G11	MDSI_A_DN[2]
G12	--
G13	VSS
G14	--
G15	MCSI_2_DP[1]
G16	--
G17	VSS
G18	--
G19	MCSI_1_DP[3]
G2	--
G20	--
G21	VSS
G22	--
G23	CORE_VCC1_S0iX
G24	--
G25	VSS
G26	--
G27	GPIO_CAMERASB05
G28	--
G29	--
G3	DDI1_TXN[0]

Ball #	Customer Name - LPDDR3
G30	VSS
G31	--
G32	--
G33	JTAG_PRDY_N
G34	--
G35	VSS
G36	--
G37	GPIO_SUS4/JTAG2_TDO
G38	--
G39	GPIO_N4/C0_BPM0_TX/ C1_BPM0_TX
G4	--
G40	--
G41	VSS
G42	--
G43	LPDDR3_M0_DQ[03]_A
G44	--
G45	VSS
G46	--
G47	LPDDR3_M0_DM[1]_A
G48	--
G49	LPDDR3_M0_DQ[10]_A
G5	VSS
G50	--
G51	LPDDR3_M0_DQS[1]_N_A
G52	--
G53	VSS
G54	--
G55	RESERVED
G56	--
G57	RESERVED
G58	--
G59	DDR_VDDQG_S4
G6	--
G7	MDSI_C_DP[1]
G8	--
G9	MDSI_A_DP[3]
H1	--

Ball #	Customer Name - LPDDR3
H10	MDSI_A_DN[3]
H11	--
H12	VSS
H13	--
H14	MCSI_2_DN[1]
H15	--
H16	MCSI_2_CLKP
H17	--
H18	MCSI_1_DN[3]
H19	--
H2	DDI1_AUXP
H20	MCSI_1_DP[0]
H21	--
H22	DDI1_VDDEN/MDSI_DDC_DATA
H23	--
H24	CORE_VCC1_S0iX
H25	--
H26	DDI0_DDC_DATA/ DDI1_DDC_DATA/ MDSI_DDC_DATA
H27	--
H28	VSS
H29	--
H3	--
H30	--
H31	--
H32	PROCHOT_N
H33	--
H34	JTAG_PREQ_N
H35	--
H36	VSS
H37	--
H38	GPIO_N5
H39	--
H4	DDI1_TXP[0]
H40	VSS
H41	--
H42	LPDDR3_M0_DQ[02]_A

Ball #	Customer Name - LPDDR3
H43	--
H44	VSS
H45	--
H46	VSS
H47	--
H48	LPDDR3_M0_DQ[09]_A
H49	--
H5	--
H50	LPDDR3_M0_DQS[1]_P_A
H51	--
H52	LPDDR3_M0_DQ[11]_A
H53	--
H54	RESERVED
H55	--
H56	VSS
H57	--
H58	LPDDR3_M0_CKE[0]_A
H59	--
H6	MDSI_C_DN[1]
H7	--
H8	MDSI_C_DN[3]
H9	--
J1	DDI2_VDDQ_G3
J10	--
J11	MDSI_A_DP[0]
J12	--
J13	VSS
J14	--
J15	MCSI_2_CLKN
J16	--
J17	MCSI_3_CLKP
J18	--
J19	MCSI_1_DN[0]
J2	--
J20	--
J21	DDI1_BKLTEN/MDSI_DDC_CLK
J22	--

Ball #	Customer Name - LPDDR3
J23	CORE_VCC1_S0iX
J24	--
J25	CORE_VCC1_S0iX
J26	--
J27	GPIO_CAMERASB09
J28	--
J29	--
J3	DDI1_AUXN
J30	GPIO_CAMERASB00
J31	--
J32	--
J33	JTAG_TDO
J34	--
J35	GPIO_SUS9
J36	--
J37	JTAG2_TMS
J38	--
J39	GPIO_N6/C0_BPM3_TX/ C1_BPM3_TX
J4	--
J40	--
J41	VSS
J42	--
J43	LPDDR3_M0_DQ[04]_A
J44	--
J45	LPDDR3_M0_DQ[07]_A
J46	--
J47	LPDDR3_M0_DQS[0]_N_A
J48	--
J49	LPDDR3_M0_DQ[14]_A
J5	DDI0_TXP[0]
J50	--
J51	LPDDR3_M0_DQ[15]_A
J52	--
J53	LPDDR3_M0_DQ[13]_A
J54	--
J55	RESERVED
J56	--

Ball #	Customer Name - LPDDR3
J57	LPDDR3_M0_CS[0]_N
J58	--
J59	DDR_VDDQG_S4
J6	--
J7	DDI1_RCOMP_N
J8	--
J9	MDSI_C_DP[3]
K1	--
K10	VSS
K11	--
K12	MDSI_A_DN[0]
K13	--
K14	MDSI_RCOMP
K15	--
K16	MCSI_3_CLKN
K17	--
K18	MCSI_RCOMP
K19	--
K2	VSS
K20	DDI1_BKLTCTL/MDSI_A_TE/ MDSI_C_TE
K21	--
K22	DDI0_BKLTCTL
K23	--
K24	CORE_VCC1_S0iX
K25	--
K26	DDI0_HPD
K27	--
K28	GPIO_CAMERASB08
K29	--
K3	--
K30	--
K31	--
K32	GPIO_CAMERASB01
K33	--
K34	GPIO_SUS10
K35	--
K36	GPIO_SUS7/PMC_SUSCLK[3]

Ball #	Customer Name - LPDDR3
K37	--
K38	GPIO_SUS1/JTAG2_TCK
K39	--
K4	DDI0_TXN[0]
K40	GPIO_N8/C0_BPM1_TX/ C1_BPM1_TX
K41	--
K42	LPDDR3_M0_DQ[00]_A
K43	--
K44	LPDDR3_M0_DQ[05]_A
K45	--
K46	LPDDR3_M0_DQS[0]_P_A
K47	--
K48	VSS
K49	--
K5	--
K50	VSS
K51	--
K52	LPDDR3_M0_DQ[12]_A
K53	--
K54	VSS
K55	--
K56	LPDDR3_M0_CKE[0]_B
K57	--
K58	VSS
K59	--
K6	VSS
K7	--
K8	DDI1_RCOMP_P
K9	--
L1	DDI1_VDDQ_G3
L10	--
L11	VSS
L12	--
L13	VSS
L14	--
L15	VSS
L16	--

Ball #	Customer Name - LPDDR3
L17	VSS
L18	--
L19	VSS
L2	--
L20	--
L21	DDI0_BKLTEN
L22	--
L23	CORE_VCC1_S0iX
L24	--
L25	CORE_VCC1_S0iX
L26	--
L27	VSS
L28	--
L29	--
L3	VSS
L30	VSS
L31	--
L32	--
L33	VSS
L34	--
L35	VSS
L36	--
L37	PWR_RSVD_OBS
L38	--
L39	VSS
L4	--
L40	--
L41	VSS
L42	--
L43	LPDDR3_M0_DQ[01]_A
L44	--
L45	VSS
L46	--
L47	LPDDR3_M0_DQ[08]_A
L48	--
L49	LPDDR3_M0_CA[0]
L5	DDI0_TXN[1]

Ball #	Customer Name - LPDDR3
L50	--
L51	LPDDR3_M0_CA[3]
L52	--
L53	RESERVED
L54	--
L55	LPDDR3_M0_CA[5]
L56	--
L57	LPDDR3_M0_CA[7]
L58	--
L59	DDR_VDDQG_S4
L6	--
L7	VSS
L8	--
L9	DDI0_AUXP
M1	--
M10	DDI0_TXN[3]
M11	--
M12	VSS
M13	--
M14	VSS
M15	--
M16	VSS
M17	--
M18	VSS
M19	--
M2	DDI1_VDDQ_G3
M20	VSS
M21	--
M22	VSS
M23	--
M24	CORE_VCC1_S0iX
M25	--
M26	CORE_VCC1_S0iX
M27	--
M28	CORE_VCC1_S0iX
M29	--
M3	--

Ball #	Customer Name - LPDDR3
M30	--
M31	--
M32	VSS
M33	--
M34	VSS
M35	--
M36	VSS
M37	--
M38	VSS
M39	--
M4	DDI0_TXP[1]
M40	GPIO0_RCOMP
M41	--
M42	VSS
M43	--
M44	LPDDR3_M0_DQ[06]_A
M45	--
M46	LPDDR3_M0_DM[0]_A
M47	--
M48	RESERVED
M49	--
M5	--
M50	LPDDR3_M0_CA[1]
M51	--
M52	VSS
M53	--
M54	LPDDR3_M0_CA[4]
M55	--
M56	VSS
M57	--
M58	VSS
M59	--
M6	DDI0_TXP[2]
M7	--
M8	DDI0_AUXN
M9	--
N1	USB_V3P3A_G3

Ball #	Customer Name - LPDDR3
N10	--
N11	DDI2_TXN[0]
N12	--
N13	--
N14	--
N15	--
N16	--
N17	--
N18	--
N19	--
N2	--
N20	--
N21	--
N22	--
N23	--
N24	--
N25	--
N26	--
N27	--
N28	--
N29	--
N3	VSS
N30	--
N31	--
N32	--
N33	--
N34	--
N35	--
N36	--
N37	--
N38	--
N39	--
N4	--
N40	--
N41	--
N42	--
N43	--

Ball #	Customer Name - LPDDR3
N44	--
N45	--
N46	--
N47	--
N48	--
N49	VSS
N5	DDI0_TXN[2]
N50	--
N51	LPDDR3_M0_CA[6]
N52	--
N53	VSS
N54	--
N55	LPDDR3_M0_CK_N_A
N56	--
N57	LPDDR3_M0_CK_P_A
N58	--
N59	DDR_VDDQG_S4
N6	--
N7	VSS
N8	--
N9	DDI0_TXP[3]
P1	--
P10	DDI2_TXP[0]
P11	--
P12	VSS
P13	--
P14	--
P15	VSS
P16	--
P17	VSS
P18	--
P19	VSS
P2	USB_V3P3A_G3
P20	--
P21	VSS
P22	--
P23	VSS

Ball #	Customer Name - LPDDR3
P24	--
P25	CORE_VCC1_S0iX
P26	--
P27	CORE_VCC1_S0iX
P28	--
P29	VSS
P3	--
P30	--
P31	--
P32	CORE_VCC0_S0iX
P33	--
P34	CORE_VCC0_S0iX
P35	--
P36	CORE_VCC0_S0iX
P37	--
P38	CORE_VCC0_S0iX
P39	--
P4	DDI0_RCOMP_N
P40	VSS
P41	--
P42	VSS
P43	--
P44	VSS
P45	--
P46	VSS
P47	--
P48	LPDDR3_M0_CA[2]
P49	--
P5	--
P50	RESERVED
P51	--
P52	VSS
P53	--
P54	LPDDR3_M0_CK_N_B
P55	--
P56	LPDDR3_M0_CK_P_B
P57	--

Ball #	Customer Name - LPDDR3
P58	VSS
P59	--
P6	VSS
P7	--
P8	VSS
P9	--
R1	VSS
R10	--
R11	VSS
R12	--
R13	--
R14	--
R15	--
R16	--
R17	--
R18	--
R19	--
R2	--
R20	--
R21	--
R22	--
R23	--
R24	--
R25	--
R26	--
R27	--
R28	--
R29	--
R3	DDI2_AUXN
R30	--
R31	--
R32	--
R33	--
R34	--
R35	--
R36	--
R37	--

Ball #	Customer Name - LPDDR3
R38	--
R39	--
R4	--
R40	--
R41	--
R42	--
R43	--
R44	--
R45	--
R46	--
R47	--
R48	--
R49	RESERVED
R5	DDI0_RCOMP_P
R50	--
R51	RESERVED
R52	--
R53	LPDDR3_M0_CA[9]
R54	--
R55	LPDDR3_M0_DQ[02]_B
R56	--
R57	VSS
R58	--
R59	VSS
R6	--
R7	VSS
R8	--
R9	VSS
T1	--
T10	DDI2_TXN[2]
T11	--
T12	VSS
T13	--
T14	UNCORE_VSFR_G3
T15	--
T16	VSS
T17	--

Ball #	Customer Name - LPDDR3
T18	VSS
T19	--
T2	DDI2_TXP[1]
T20	VSS
T21	--
T22	VSS
T23	--
T24	VSS
T25	--
T26	VSS
T27	--
T28	VSS
T29	--
T3	--
T30	--
T31	CORE_VCC0_S0iX
T32	--
T33	CORE_VCC0_S0iX
T34	--
T35	CORE_VCC0_S0iX
T36	--
T37	CORE_VCC0_S0iX
T38	--
T39	VSS
T4	DDI2_AUXP
T40	--
T41	VSS
T42	--
T43	VSS
T44	--
T45	VSS
T46	--
T47	--
T48	LPDDR3_M0_OCAVREF
T49	--
T5	--
T50	LPDDR3_M0_ODQVREF

Ball #	Customer Name - LPDDR3
T51	--
T52	LPDDR3_M0_CA[8]
T53	--
T54	LPDDR3_M0_DQ[00]_B
T55	--
T56	VSS
T57	--
T58	LPDDR3_M0_DQ[03]_B
T59	--
T6	VSS
T7	--
T8	USB_VBUSSNS
T9	--
U1	USB_VDDQ_G3
U10	--
U11	DDI2_TXP[2]
U12	--
U13	--
U14	--
U15	--
U16	--
U17	--
U18	--
U19	--
U2	--
U20	--
U21	--
U22	--
U23	--
U24	--
U25	--
U26	--
U27	--
U28	--
U29	--
U3	DDI2_TXN[1]
U30	--

Ball #	Customer Name - LPDDR3
U31	--
U32	--
U33	--
U34	--
U35	--
U36	--
U37	--
U38	--
U39	--
U4	--
U40	--
U41	--
U42	--
U43	--
U44	--
U45	--
U46	--
U47	--
U48	--
U49	VSS
U5	VSS
U50	--
U51	VSS
U52	--
U53	VSS
U54	--
U55	LPDDR3_M0_DM[0]_B
U56	--
U57	LPDDR3_M0_DQ[01]_B
U58	--
U59	VSS
U6	--
U7	USB_OTG_ID
U8	--
U9	RESERVED
V1	--
V10	USB_DN[1]

Ball #	Customer Name - LPDDR3
V11	--
V12	USB_DP[0]
V13	--
V14	--
V15	VSS
V16	--
V17	VSS
V18	--
V19	VSS
V2	USB_VDDQ_G3
V20	--
V21	UNCORE_VNN_S4
V22	--
V23	UNCORE_VNN_S4
V24	--
V25	UNCORE_VNN_S4
V26	--
V27	PWR_RSVD_OBS
V28	--
V29	VSS
V3	--
V30	--
V31	--
V32	CORE_VCC0_S0iX
V33	--
V34	CORE_VCC0_S0iX
V35	--
V36	CORE_VCC0_S0iX
V37	--
V38	CORE_VSS0_SENSE
V39	--
V4	DDI2_TXP[3]
V40	CORE_VCC0_SENSE
V41	--
V42	VSS
V43	--
V44	VSS

Ball #	Customer Name - LPDDR3
V45	--
V46	VSS
V47	--
V48	LPDDR3_M0_DQ[17]_B
V49	--
V5	--
V50	LPDDR3_M0_DQ[19]_B
V51	--
V52	VSS
V53	--
V54	VSS
V55	--
V56	LPDDR3_M0_DQS[0]_N_B
V57	--
V58	VSS
V59	--
V6	VSS
V7	--
V8	VSS
V9	--
W1	USB_VDDQ_G3
W10	--
W11	USB_DN[0]
W12	--
W13	--
W14	VSS
W15	--
W16	VSS
W17	--
W18	VSS
W19	--
W2	--
W20	UNCORE_VNN_S4
W21	--
W22	UNCORE_VNN_S4
W23	--
W24	UNCORE_V1P15_S0iX

Ball #	Customer Name - LPDDR3
W25	--
W26	UNCORE_VNN_S4
W27	--
W28	UNCORE_VNN_S4
W29	--
W3	USB_DN[2]
W30	--
W31	CORE0_VSFR_G3
W32	--
W33	CORE_VCC0_S0iX
W34	--
W35	CORE_VCC0_S0iX
W36	--
W37	CORE_VCC0_S0iX
W38	--
W39	F_V1P15_S0iX
W4	--
W40	--
W41	VSS
W42	--
W43	VSS
W44	--
W45	VSS
W46	--
W47	--
W48	--
W49	LPDDR3_M0_DQ[16]_B
W5	DDI2_TXN[3]
W50	--
W51	LPDDR3_M0_DQS[2]_N_B
W52	--
W53	VSS
W54	--
W55	LPDDR3_M0_DQS[0]_P_B
W56	--
W57	LPDDR3_M0_DQ[05]_B
W58	--

Ball #	Customer Name - LPDDR3
W59	DDR_VDDQG_S4
W6	--
W7	VSS
W8	--
W9	USB_DP[1]
Y1	--
Y10	VSS
Y11	--
Y12	VSS
Y13	--
Y14	--
Y15	--
Y16	--
Y17	--
Y18	--
Y19	--
Y2	VSS
Y20	--
Y21	--
Y22	--
Y23	--
Y24	--
Y25	--
Y26	--
Y27	--
Y28	--
Y29	--
Y3	--
Y30	--
Y31	--
Y32	--
Y33	--
Y34	--
Y35	--
Y36	--
Y37	--
Y38	--

Ball #	Customer Name - LPDDR3
Y39	--
Y4	USB_DP[2]
Y40	--
Y41	--
Y42	--
Y43	--
Y44	--
Y45	--
Y46	--
Y47	--
Y48	LPDDR3_M0_DQ[20]_B
Y49	--
Y5	--
Y50	LPDDR3_M0_DQS[2]_P_B
Y51	--
Y52	LPDDR3_M0_DQ[22]_B
Y53	--
Y54	LPDDR3_M0_DQ[04]_B
Y55	--
Y56	VSS
Y57	--
Y58	LPDDR3_M0_DQ[06]_B
Y59	--
Y6	USB_DP[3]
Y7	--
Y8	VSS
Y9	--

§

21 Package Information

The SoC comes in Flip-Chip Ball Grid Array (FCBGA) package and consists of a silicon die mounted face down on an organic substrate populated with solder balls on the bottom side. Care should be taken to avoid contact with the package inside this area.

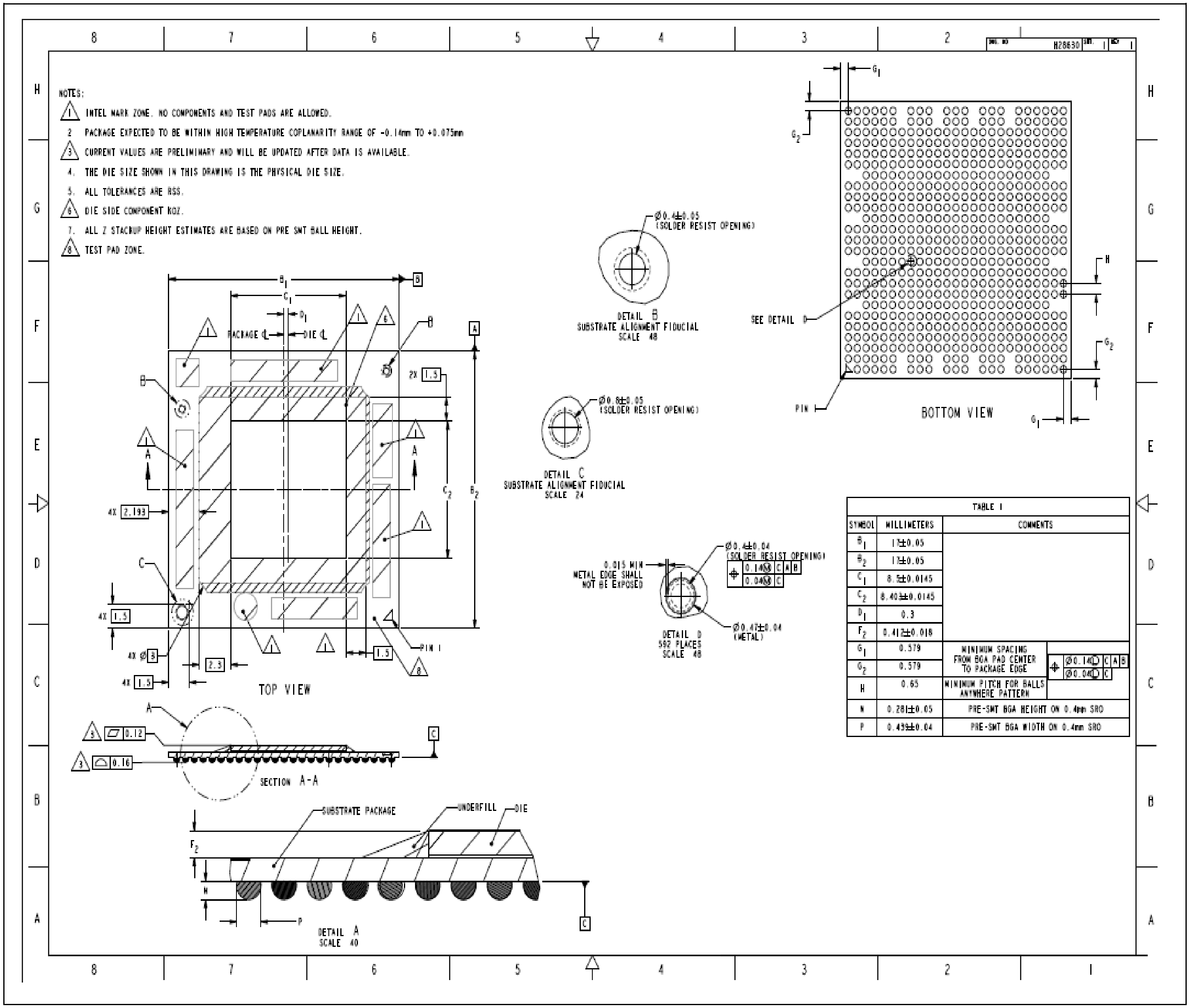
21.1 SoC Attributes

Table 165. SoC Attributes

	Category	T4	T3
Package	Type	17x17mm Type 4	17x17mm Type 3
	IO count	628	378
	Core Process (nm)	14	14
	Ball count	1380	592
	Ball pitch	0.4mm	0.65mm
	Z-height	0.937mm	1.002mm

21.2 Package Diagrams

Figure 44. Package Mechanical Drawing for T3 (Part 1)



NOTES:

1 INTEL MARK ZONE. NO COMPONENTS AND TEST PADS ARE ALLOWED.

2 PACKAGE EXPECTED TO BE WITHIN HIGH TEMPERATURE COPLANARITY RANGE OF -0.14mm TO +0.075mm

3 CURRENT VALUES ARE PRELIMINARY AND WILL BE UPDATED AFTER DATA IS AVAILABLE.

4. THE DIE SIZE SHOWN IN THIS DRAWING IS THE PHYSICAL DIE SIZE.

5. ALL TOLERANCES ARE RSS.

6 DIE SIDE COMPONENT KOZ.

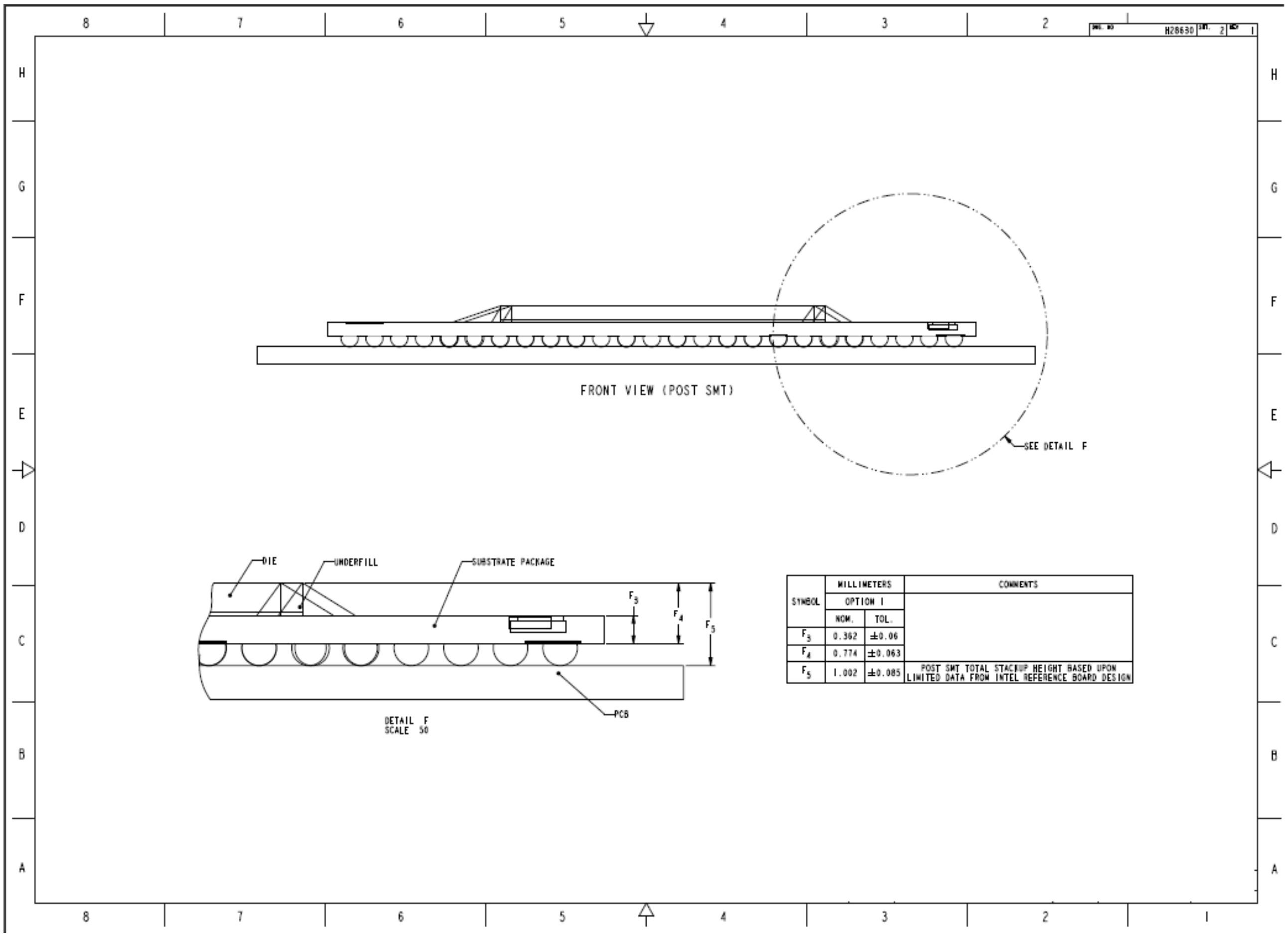
7. ALL Z STACKUP HEIGHT ESTIMATES ARE BASED ON PRE SMT BALL HEIGHT.

8 TEST PAD ZONE.

TABLE 1

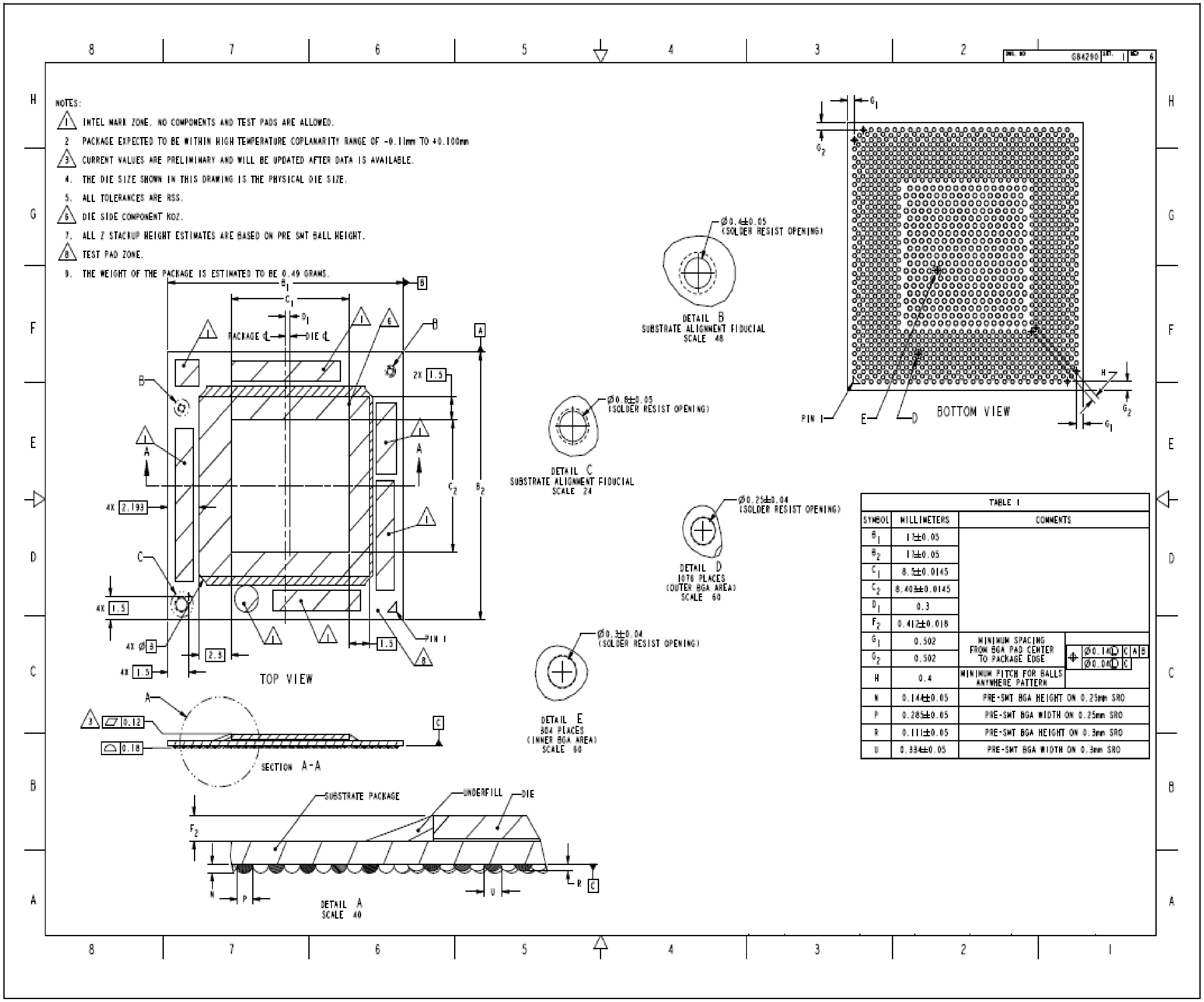
SYMBOL	MILLIMETERS	COMMENTS	
B_1	17±0.05		
B_2	17±0.05		
C_1	8.5±0.0145		
C_2	8.403±0.0145		
D_1	0.3		
F_2	0.412±0.018		
G_1	0.579	MINIMUM SPACING FROM BGA PAD CENTER TO PACKAGE EDGE	⌖ Ø0.14Ⓛ C A B Ø0.04Ⓛ C
G_2	0.579		
H	0.65	MINIMUM PITCH FOR BALLS ANYWHERE PATTERN	
N	0.281±0.05	PRE-SMT BGA HEIGHT ON 0.4mm SRO	
P	0.439±0.04	PRE-SMT BGA WIDTH ON 0.4mm SRO	

Figure 45. Package Mechanical Drawing for T3 (Part 2)



SYMBOL	MILLIMETERS		COMMENTS
	OPTION I		
	NOM.	TOL.	
F_3	0.362	±0.06	
F_4	0.774	±0.063	
F_5	1.002	±0.085	POST SMT TOTAL STACKUP HEIGHT BASED UPON LIMITED DATA FROM INTEL REFERENCE BOARD DESIGN

Figure 46. Package Mechanical Drawing for T4 (Part 1)



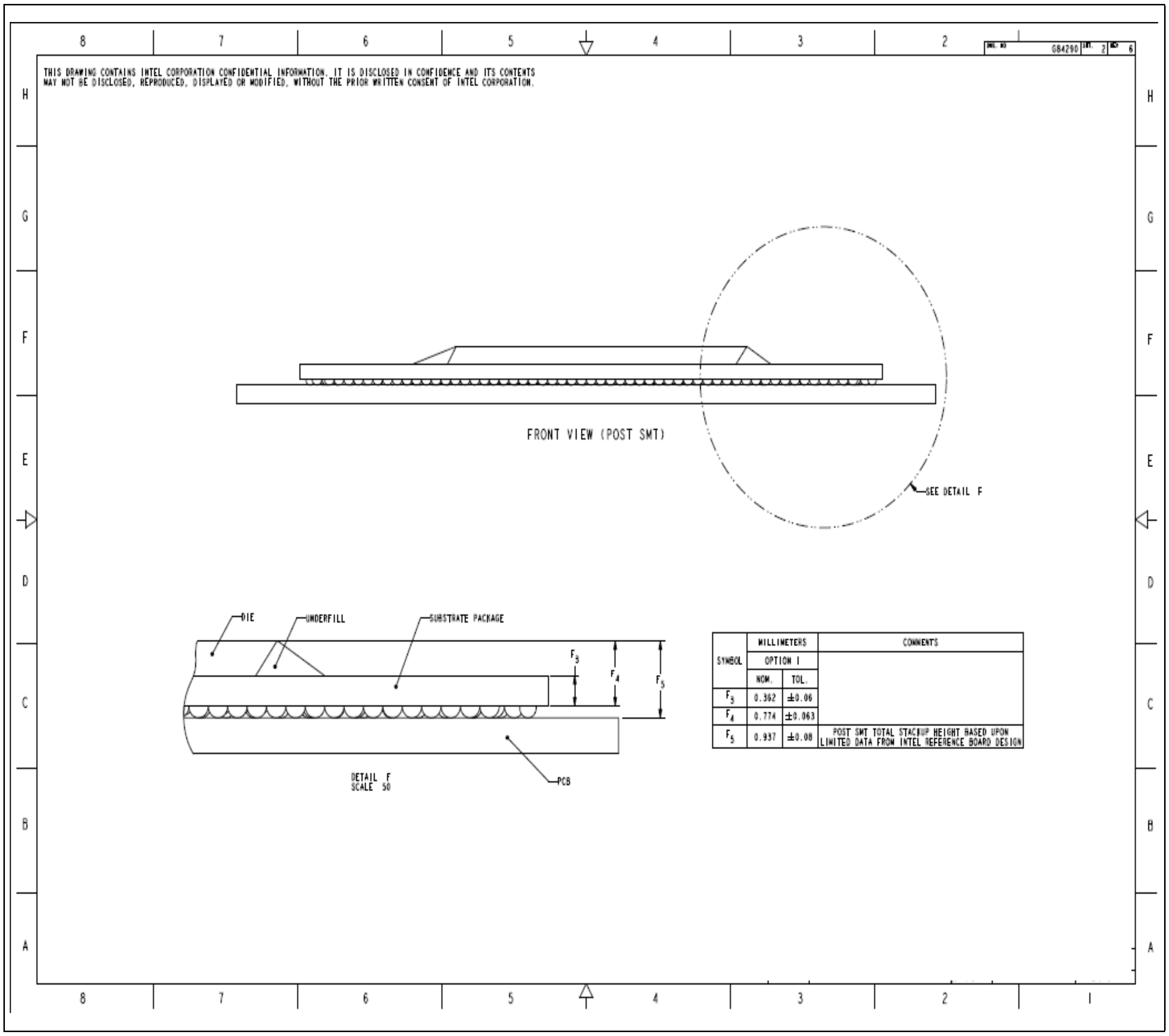
NOTES:

1. INTEL MARK ZONE. NO COMPONENTS AND TEST PADS ARE ALLOWED.
2. PACKAGE EXPECTED TO BE WITHIN HIGH TEMPERATURE COPLANARITY RANGE OF -0.11mm TO +0.100mm
3. CURRENT VALUES ARE PRELIMINARY AND WILL BE UPDATED AFTER DATA IS AVAILABLE.
4. THE DIE SIZE SHOWN IN THIS DRAWING IS THE PHYSICAL DIE SIZE.
5. ALL TOLERANCES ARE RSS.
6. DIE SIDE COMPONENT KOZ.
7. ALL Z STACKUP HEIGHT ESTIMATES ARE BASED ON PRE SMT BALL HEIGHT.
8. TEST PAD ZONE.
9. THE WEIGHT OF THE PACKAGE IS ESTIMATED TO BE 0.49 GRAMS.

TABLE 1

SYMBOL	MILLIMETERS	COMMENTS	
B_1	13±0.05		
B_2	13±0.05		
C_1	8.5±0.0145		
C_2	8.403±0.0145		
D_1	0.3		
F_2	0.412±0.018		
G_1	0.502	MINIMUM SPACING FROM BGA PAD CENTER TO PACKAGE EDGE	⌖ ⌀0.14Ⓛ C A B ⌀0.04Ⓛ C
G_2	0.502		
H	0.4	MINIMUM PITCH FOR BALLS ANYWHERE PATTERN	
N	0.144±0.05	PRE-SMT BGA HEIGHT ON 0.25mm SRO	
P	0.285±0.05	PRE-SMT BGA WIDTH ON 0.25mm SRO	
R	0.111±0.05	PRE-SMT BGA HEIGHT ON 0.3mm SRO	
U	0.334±0.05	PRE-SMT BGA WIDTH ON 0.3mm SRO	

Figure 47. Package Mechanical Drawing for T4 (Part 2)



THIS DRAWING CONTAINS INTEL CORPORATION CONFIDENTIAL INFORMATION. IT IS DISCLOSED IN CONFIDENCE AND ITS CONTENTS MAY NOT BE DISCLOSED, REPRODUCED, DISPLAYED OR MODIFIED, WITHOUT THE PRIOR WRITTEN CONSENT OF INTEL CORPORATION.

SYMBOL	MILLIMETERS		COMMENTS
	OPTION 1		
	NOM.	TOL.	
F_3	0.362	±0.06	
F_4	0.774	±0.063	
F_5	0.937	±0.08	POST SMT TOTAL STACKUP HEIGHT BASED UPON LIMITED DATA FROM INTEL REFERENCE BOARD DESIGN

§