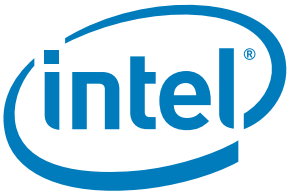
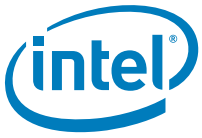


N-series Intel® Pentium® Processors and Intel® Celeron® Processors

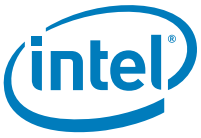
Datasheet – Volume 1 of 3

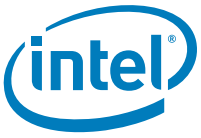
February 2016

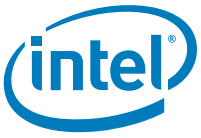
Document Number: 332092-002

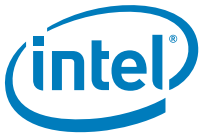


Contents



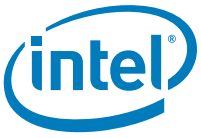






Figures

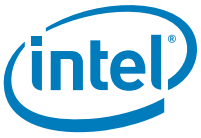
Tables

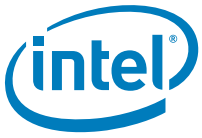


Revision History

Revision Number	Description	Revision Date
001	Initial release	April 2015
002	• Minor updates throughout for clarity **Chapter 1: Introduction** • Added information on J-series Intel® Pentium® Processors and Intel® Celeron® Processors • Updated Section 1.3. Removed support for HS400 in the Storage Interface section of the table **Chapter 5: Power Sequence** • Added information on SUSPWRDNACK usage **Chapter 6, Thermal Management** • Updated Section 5.3. Added note. **Chapter 21: Electrical Specifications** • Table 21-3. Updated table title to reflect stepping (C-step). Updated S3, S4 and S5 Imax values for VNN • Table 21-3. Updated Notes section • Table 21-4. Added SoC power rail and max current specifications for D-step	February 2016

§ §





1 Introduction

The N-series Intel® Pentium® processor and Intel® Celeron® processor families are 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.

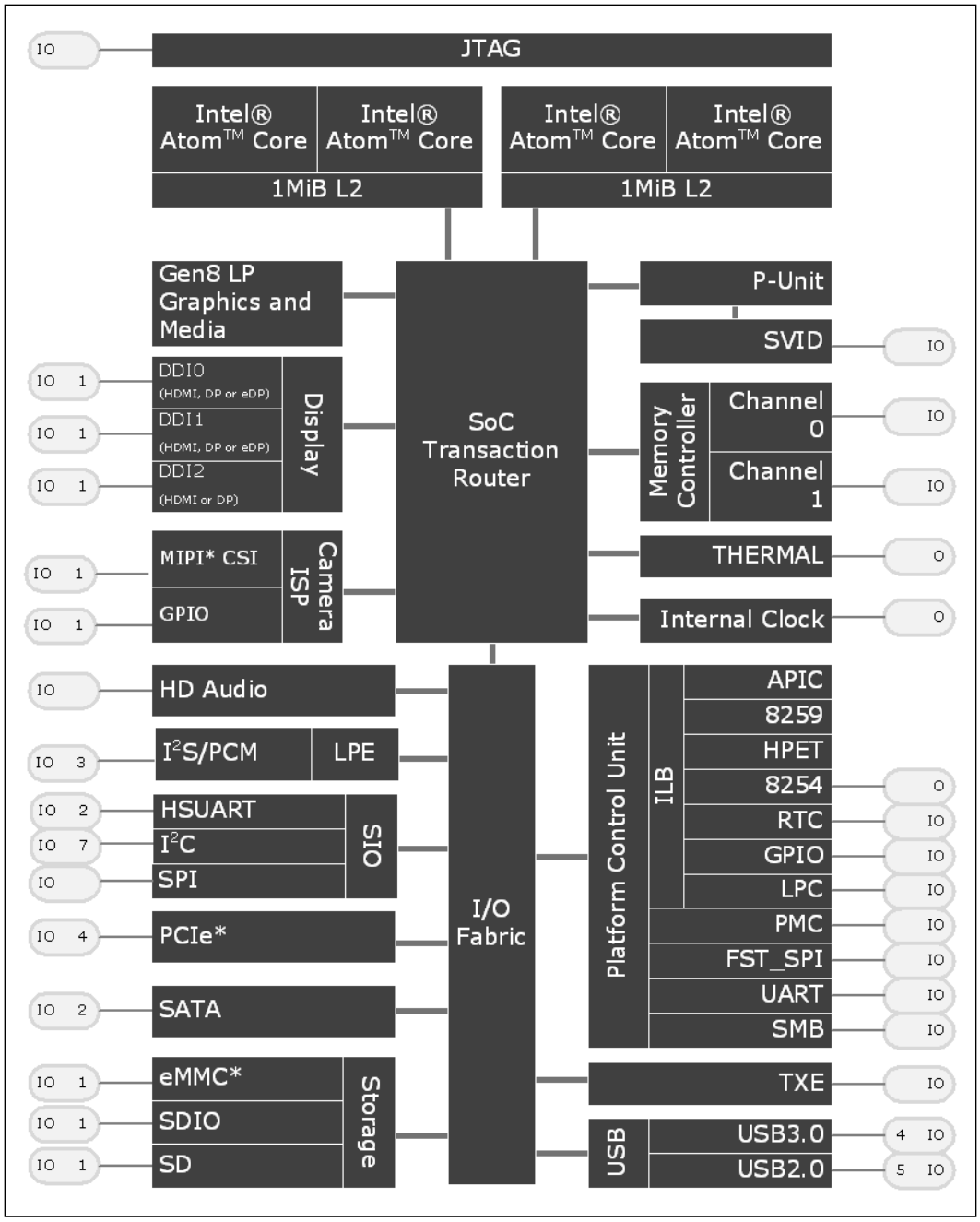
The following figure shows the system-level block diagram of the SoC. Refer to the subsequent chapters for detailed information on the functionality of the different interface blocks.

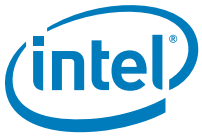
This document is distributed as a part of the complete Datasheet document consisting of three volumes. Refer to Section 1.1, "Document Structure and Scope" on page 19 for high-level content listings of each volume.

Throughout this document, the N-series Intel® Pentium® processor and Intel® Celeron® processor families may be referred to simply as "processor".

Throughout this document, the N-series Intel® Pentium® processor and Intel® Celeron® processor families refers to the Intel® Pentium® processor N3710, N3700 and Intel® Celeron® processors N3160, N3150, N3060, N3050, N3010, and N3000. The J-series Intel® Pentium® processor and Intel® Celeron® processor families refers to the Intel® Pentium® processor J3710 and Intel® Celeron® processors J3160 and J3060.

Figure 1-1. SoC Block Diagram (Netbook)





1.1 Document Structure and Scope

The following table summarizes the structure and scope of each volume of the processor Datasheet. Refer to the "Related Documents" section for order information.

Table 1-1. Structure of the Processor Datasheet (Sheet 1 of 2)

Description
Volume 1: Architecture, Ballout, Package, and Electrical Specifications
• Introduction
• Physical Interfaces
• Processor Core
• Integrated Clock
• Thermal Management
• Power Management
• System Memory Controller
• Graphics, Video, and Display
• MIPI*-CSI (Camera Serial Interface) and ISP
• SoC Storage
• USB Controller Interfaces
• Low Power Engine (LPE) for Audio (I^2S)
• Intel® Trusted Execution Engine (Intel® TXE)
• Intel® High Definition Audio (Intel® HD Audio)
• Serial I/O (SIO) Overview
• Platform Controller Unit (PCU) Overview
• Serial ATA (SATA)
• PCI Express* 2.0
• Ball Map, Ball Out, and SoC Pin Locations
• Package Information
• Electrical Specifications
Volume 2: Registers
• Register Access Method Registers
• Mapping Address Space Registers
• System Memory Controller Registers
• SoC Transaction Router Registers
• Graphics, Video, and Display Registers
• MIPI*-CSI (Camera Serial Interface) and ISP Registers
• SoC Storage Registers
• USB xHCI PCI Configuration Registers
Volume 3: Registers
• Low Power Engine (LPE) for Audio (I2S) Registers
• Intel® High Definition Audio (Intel® HD Audio) Registers
• Serial I/O (SIO) Registers

Table 1-1. Structure of the Processor Datasheet (Sheet 2 of 2)

Description
• Platform Controller Unit (PCU) Registers
• Serial ATA (SATA) Registers
• PCI Express* 2.0 Registers

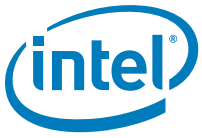
1.2 Terminology

Term	Description
AHCI	Advanced Host Controller Interface
ACPI	Advanced Configuration and Power Interface
CCm	Closely Coupled Memory
CCI	Camera Control Interface
Cold Reset	Full reset is when PWROK is de-asserted and all system rails except VCCRTC are powered down
CRU	Clock Reset Unit
CSI	Camera Serial Interface
DP	DisplayPort*
DTS	Digital Thermal Sensor
EIOB	Electronic In/Out Board
EMI	Electro Magnetic Interference
eDP	embedded DisplayPort*
GPIO	General Purpose IO
HDCP	High-Bandwidth Digital Content Protection
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/).
HPET	High Precision Event Timer
IGD	Internal Graphics Unit
Intel® TXE	Intel® Trusted Execution Engine (Intel® TXE)
IPC	Inter-Processor Communication
ISH	Integrated Sensor Hub
ISP	Image Signal Processor
LCD	Liquid Crystal Display
LFM	Low Frequency Mode
LPC	Low Pin Count
LPDDR	Low Power Dual Data Rate memory technology.
LPE	Low Power Engine
MFM	Minimum Frequency Mode
MIPI*-CSI	MIPI*-Camera Interface Specification
MIPI*-DSI	MIPI*-Display Interface Specification

Term	Description
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.
PCU	Platform Controller Unit
PEG/PCIe	PCI Express Graphics /PCI Express
PMU	Power Management Unit
PWM	Pulse Width Modulation
PSP	Programmable Serial Protocol
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.
RTC	Real Time Clock
SATA	Serial ATA
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.
SMBus	System Management 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
SPI	Serial Peripheral Interface
SSP	Synchronous Serial Protocol
TDP	Thermal Design Power
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.
UART	Universal Asynchronous Receiver/Transmitter
VCO	Voltage Controlled Oscillator
Warm Reset	Warm reset is when both PMC_PLTRST# and PMC_CORE_PWROK are asserted.

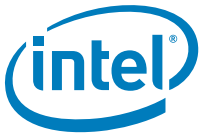
1.3 Feature Overview

Interface	Category	SoC Features
CPU	No. Cores	Up to 4 IA low-power Intel® processor cores Quad Out-of-Order Execution (OOE) processor cores Based on 14nm processor technology
	Modules / Cashes	Cores are grouped into Dual-Core modules On-die, 32KB 8-way L1 instruction cache and 24KB 6-way L1 data cache per core. On-die, 1MB, 16-way L2 cache, shared per two cores (module)
	Threads	One thread per core ***Note:*** Intel® Hyper-Threading Technology is not supported
	Address size	Support 36-bit physical address, 48-bit linear address size
	Core State	C0, C1, C1E, C6C, C6, and C7 states
Package	Type	Type 3 BGA (FCBGA15)
	Processor Core Process	14 nm
	X-Y Dimension	25 mm x 27 mm
	Post-SMT Height	1 mm
	Ball Pitch	0.593 mm
	Pin/Ball Count	1170
Memory	Interface	DDR3L (1.35V DRAM interface I/Os) Dual Channel Up to two ranks per channel (4 ranks in total)
	Transfer Data rate	Up to 1600MT/s
	Device Data Width	x8, x16
	Memory Bandwidth	12.8GB/s for 1600MT/s single-channel 25.6GB/s for 1600MT/s dual-channel
	Data bus	64-bit only per channel
	DRAM Device Technologies	Standard 1Gb, 2Gb, 4Gb, and 8Gb Read latency 5, 6, 7, 8, 9, 10, 11, 12, 13 Write latency 3, 4, 5, 6, 7, 8
	Other	Support Truck Clock Gating Support early SR exit Support slow power-down Support command signal tri-state not driving a valid command Support different physical mappings of bank address to optimize performance Support Dynamic Voltage and Frequency Scaling Aggressive power management to reduce power consumption Proactive page closing policies to close unused pages
Graphics	Generation	Gen 8-LP Intel® graphics core
	Units	16 Execution Units (EUs)
	HW Accelerators	3-D: DirectX 11.1, OpenGL, 4.2, OpenGL ES 3.0, OpenCL 1.2 2-D: HEVC, H.264, MPEG2, VC-1 WMV9.
	Other	Support content protection using PAVP2.0, HDCP (1.4 wired/2.2 wireless) and Media Vault DRM Support 4x anti-aliasing Graphics Burst enabled through energy counters



Interface	Category	SoC Features
Display	Interfaces	3 Digital Display Interfaces (DDIs) Max of 3 simultaneously displays
	Configurations	eDP*: support on 2 port only, DDI[0:1] DP*: Support on all 3 ports, DDI[0:2] HDMI: Support on all 3 ports, DDI[0:2]
	Transfer Data Rate	eDP: 2.7Gb/s DP: 2.7Gb/s HDMI: 2.97Gb/s
	Max Resolution	eDP: 2560 x 1440 @ 60Hz DP: 3840 x 2160 @ 30Hz HDMI: 3840 x2 160 @ 30Hz
	Other	Support Audio on DP and HDMI only Support Intel® Display Power Saving Technology (Intel® DPST) Support Display Refresh Rate Switching Technology (DRRS)
Intel® High Definition Audio	No. Ports	3 LPE (SSP) I2S ports ***Note:*** LPE is supported for non Windows* Operating System platforms only.
	Other	Decode:MP3, AAC-LC, HE-AAC v1/2, WMA9, 10, PRO, Lossless, Voice, MPEG layer 2, Real Audio, OggVorbis, FLAC, DD/DD+ Encode: MP3, AAC-LC, WMA, DD-2channel Supports MSI and legacy interrupt delivery Support for ACPI D3 and D0 Device States Supports up to: • 6 streams (three input, three output) • 16 channels per stream • 32 bits/sample • 192 KHz sample rate 24 MHz HDA_CLK supports • SDO double pumped at 48Mb/s • SDI single pumped at 24Mb/s Supports 1.5V and 1.8V mode Supports optional Immediate Command/Response mechanism
Imaging	Interface	MIPI*-CSI 2.0
	No. Ports	Up to 3 ports
	No. Lanes	Up to 6 Lanes
	Data Rate	Up to 1.5Gbps (Resulting in roughly 1.2Gbps/s of actual pixels)
	Resolution	2-D Image: Up to 5 MP 2-D Video: Full HD 1080p30 Audio: Full HD 1080p30
	Other	Support Image Signal Processor (ISP) with DMA and local SRAM (Image data received by MIPI*-CSI interface is relayed to the ISP for processing) Support lossless compressed image streams to increase the effective bandwidth without losing data

Interface	Category	SoC Features
PCI Express*	Interface	PCIe* 2.0
	Signaling Rate	5.0 or 2.5 GT/s operation per root port
	No. Lanes	4 Lanes and up to 4 PCIe* root ports
	Flexible Root port configurations	Support (4)x1 - (1)x2,(2)x1 - (1)x4 - (2)x2 Default option: (4)x1
	Interrupts and Events	Legacy (INTx) and MSI Interrupts General Purpose Events Express Card Hot-plug Event System error Events
	Power Management	Link State support for L0s, L1, L2 Power down in ACPI S3 state - L3
	Other	Support Virtual Channel for VC0 only
Serial ATA	Interface	SATA Gen3 (600MB/sec.), SATA Gen2 (300MB/sec), SATA Gen1 (250MB/sec)
	No. Ports	2 SATA ports
	Signaling Rate	SATA Gen3 (6Gbps), SATA Gen2 (3Gbps), SATA Gen1 (1.5Gbps)
	Other	Support Hot-plug Support AHCI operations (Application layer is configurable for AHCI) Support clock gating and dynamic trunk gating
Serial I/O	I^2C Ports	7
	I^2C Speed	Standard mode (bit rate up to 100Kb/s) Fast mode (bit rate up to 400Kb/s) Fast Mode Plus (bit rate up to 1Mb/s) High-Speed mode (bit rate up to 1.7Mb/s)
	HSUART Ports	2
	HSUART Baud Rate	Between 300 and 3686400
	SPI	***Note:*** SIO SPI is supported for non Windows* Operating System platforms only.
	SPI Speed	Up to 20Mb/s
	SPI Other	Single interrupt line could be assigned to interrupt PCI INT [A] or ACPISIO INT[1] Configurable frame format, clock polarity and clock phase supporting three SPI peripherals only Two Chip selects are supported for each of the 2 SPI controllers (SPI1 and SPI3). Supports master mode only Receive and transit buffers are both 256 x 32 bits The receive buffer has only 1 water mark The transmit buffer has 2 water marks



Interface	Category	SoC Features
Storage	SD Card Interface	v3.0 (1 port)
	SD Card Speed	Host Clock rate variable between 0 and 200 MHz SDR104 mode (up to 800Mb/s data rate using 4 parallel data lines)
	SD Card data transfer rate	Transfer the date in 1 bit and 4-bit SD modes Transfers the data in following UHS-I modes (SDR12/25/50/104 and DDR50).
	SD Card Other	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 Interface can not be used as a wake event
	SDIO Interface	v3.0 (1 port)
	SDIO Speed	Host Clock rate variable between 0 and 200 MHz SDR104 mode (up to 800Mb/s data rate using 4 parallel data lines)
	SDIO data transfer rate	Transfer the date in 1 bit and 4-bit SD modes Transfers the data in following UHS-I modes (SDR12/25/50/104 and DDR50).
	SDIO Other	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 Interface can not be used as a wake event
	eMMC Interface	v4.5.1 (1 port)
	eMMC Speed	Host Clock rate variable between 0 and 200MHz HS200 mode (Up to 1600 Mb/s data rate using 8-bit parallel lines) High Speed DDR mode (Up to 800 Mb/s data rate using 8-bit parallel lines)
	eMMC data transfer rate	Transfer the date in 1 bit and 4-bit SD modes
	eMMC Other	Cyclic Redundancy Check CRC7 for command and CRC16 for data integrity
USB	USB 3.0	4 Super Speed (SS) Ports in total (All multiplexed with HS ports)
	USB 3.0 Max Speed	5Gb/s
	USB 2.0	1 High Speed (HS) Ports + 4 multiplexed with SS Ports (5 Ports in total)
	USB 2.0 Max Speed	5Gb/s
	USB HSIC	2 High Speed Inter-Chip Ports
	USB HSIC Max Speed	480Mb/s Only

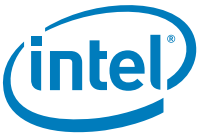
Interface	Category	SoC Features
Platform Controller Unit (PCU)	UART	Max Baud Rate: 115,200 bps (thus recommended for debug only) 16550 controller compliant Reduced Signal Count: TX and RX only COM1 interface
	FAST SPI	For SPI Flash only, of up to 16MB size each. No other SPI peripherals are supported. Stores boot firmware and system configuration data Supports frequencies of 20 MHz, 33 MHz, and 50 MHz **Note:** Fast_SPI signals do not get tri-stated during RSMRST# assertion. **Note:** Flash Sharing is not supported for the processor Platforms
	PMC	Controls many of the power management features present in the SoC.
	iLB	Supports legacy PC platform features Sub-blocks include LPC, GPIO, 8259 PIC, I/O-APIC, 8254 timers, HPET timers and the RTC.

1.4 Related Documents

Table 1-2. Related Documents

Title	Document Number / Location
N-series Intel® Pentium® Processors and Intel® Celeron® Processors Datasheet Volume 2 of 3	332093
N-series Intel® Pentium® Processors and Intel® Celeron® Processors Datasheet Volume 3 of 3	332094
N-series Intel® Pentium® Processors and Intel® Celeron® Processors Specification Update	332095

§



2 Physical Interfaces

2.1 Platform Power Rails

Table 2-1. Platform Power Well Definitions

Power Type	Pin Name	Voltage Range (V)	Power Well Description	Tolerance	Power System States
VCC	CORE_VCC1	0.5–1.3	Variable voltage rail for core. (VCC0 & VCC1 rails merged into one single voltage rail VCC)	See Table 21-5 on page 220	S0
VGG	DDI_VGG	0.5–1.2	Variable voltage rail for Graphics Core.	See Table 21-5 on page 220	S0
VNN	UNCORE_VNN_S4	0.5–1.05	Config-1: Variable voltage rail for SoC.	See Table 21-5 on page 220	S0 - S5
	UNCORE_VNN_S4	1.05	Config-2: Fixed VID rail for SoC.		
V1P05A	VCCSRAMSOCIUN_1P05, FUSE_V1P05A_G3, VCCSRAMSOCIUN_1P05, USB3_V1P05A_G3, SATA_V1P05A_G3, PCIE_V1P05A_G3, DDR_V1P05A_G3	1.05	Fixed voltage rail for P-unit, LPE, TXE, I/Os and PLLs.	±5%	S0 - S5
V1P15S	VCCSRAMGEN_1P15, FUSE_V1P15, CORE_V1P15	1.15	Fixed voltage rail for SoC L2, SoC RAM, Graphics, camera.	±5%	S0
V1P24A	CORE_VSFR_G3, USB_VDDQ_G3 USBHSIC_V1P2A_G3, ICLK_VSFR_G3 MIPI_V1P2A_G3	1.24	Fixed voltage rail for I/Os and PLLs.	±5%	S0 - S5
V1P8A	GPIO_V1P8A_G3, SDIO_V3P3A_V1P8A_G3[1] ,FUSE_V1P8A_G3	1.8	Fixed voltage rail for I/Os.	±5%	S0 - S5
VDDQ	DDR_VDDQ_G_S4 DDRSFR_VDDQ_G_S4	1.35	Fixed voltage rail for DDR3L PHY.	±5%	S0 - S3
V3P3A	RTC_V3P3A_G5, SDIO_V3P3A_V1P8A_G3[1]	3.3	Fixed voltage rail for I/Os.	±5%	S0 - S5
V3P3_RTC	RTC_V3P3RTC_G5	3.3	Fixed Voltage rail for RTC (Real Time Clock)	2–3VDC @ Battery or else 3.3 volts (pre-diode drop)	S0 - G3
HD Audio Rail	VCCCFIOAZA_1P80	1.5/1.8	1.5V or 1.8V fixed voltage rail for HD Audio.	±5%	S0 - S5

Notes:
1. The voltage supply for SDIO can be 1.8V or 3.3V.

Table 2-2. Buffer Type Definitions

Buffer Type	Buffer Description
MIPI-DPHY	1.24V tolerant MIPI DPHY buffer type
USB3 PHY	1.0V tolerant USB3 PHY buffer type
USB2 PHY	1.8V tolerant USB3 PHY buffer type
HSIC PHY	1.2V tolerant HSIC PHY buffer type
SATA PHY	1.0V tolerant SATA PHY buffer type
PCIe PHY	1.0V tolerant PCIe* PHY buffer type.
RTC PHY	3.3V tolerant RTC PHY buffer type.
GPIO	GPIO buffer type. This can be of the following types: 1.8/3.3V.
MODPHY	1.0V tolerant MODPHY buffer type.
DDR3	1.5V 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~208 MHz).
GPIOMV, MS	GPIO Buffer type, Medium Voltage (1.8V), Medium Speed (FMAX~60 MHz).
GPIOMV, MS, CLK	GPIO Buffer type, Medium Voltage(1.8V),Medium Speed (FMAX~60 MHz), Clock.
GPIOMV, HS, CLK	GPIO Buffer type, Medium Voltage (1.8V), High Speed (FMAX~208 MHz), Clock.
GPIOMV, HS, RCOMP	GPIO Buffer type, Medium Voltage (1.8V), High Speed (FMAX~208 MHz), RCOMP.
GPIOMV, MS, I2C	GPIO Buffer type, Medium Voltage (1.8V), Medium Speed (FMAX~60 MHz), I2C.
GPIOHV, HS	GPIO Buffer type, High Voltage (1.8V/3.3V), High Speed (FMAX~208 MHz).
GPIOHV, HS, RCOMP	GPIO Buffer type, High Voltage (1.8V/3.3V), High Speed (FMAX~208 MHz), RCOMP.

Note: GPIO mode, where register controlled will not hit MAX speeds they only matter when functionally used.

Table 2-3. Default Memory Controller Interface Signals (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.

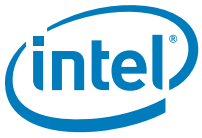


Table 2-3. Default Memory Controller Interface Signals (Sheet 2 of 2)

Buffer State	Description
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)
1	Buffer drives V_{OH}
0	Buffer driver 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.
Retrain	Retrain configuration/data prior to standby.

2.2 SoC Physical Signal Per Interface

This section lists signals groups of each interface and describes the states of each signal during supported buffer states.

2.2.1 System Memory Controller Interface Signals (DDR3L)

Table 2-4. DDR3L System Memory Signals (Sheet 1 of 2)

				Default Buffer State	
Signal Name	Dir	Platform Power	Type	Pwrgood Assert State	Resetout De-assert State
DDR3_M0_MA[15:0]	O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M0_CK[1,0]_P	O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M0_CK[1,0]_N	O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M0_CKE[1:0]	O	VDDQ (V1P35)	DDR3	Weak 0	0
DDR3_M0_CS[1,0]_N	O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M0_CAS_N	O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M0_RAS_N	O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M0_WE_N	O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M0_BS[2:0]	O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M0_DRAMRST_N	O	VDDQ (V1P35)	DDR3	Weak 0	0
DDR3_M0_ODT[1,0]	O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M0_DQ[63:0]	I/O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M0_DM[7:0]	O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M0_DQSP[7:0]	I/O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M0_DQSN[7:0]	I/O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M0_OCAVREF	I	0.5*VDDQ	DDR3	Z	Z
DDR3_M0_ODQVREF	I	0.5*VDDQ	DDR3	Z	Z
DDR3_M0_RCOMP	I	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M1_MA[15:0]	O	VDDQ (V1P35)	DDR3	Z	Z

Table 2-4. DDR3L System Memory Signals (Sheet 2 of 2)

Signal Name	Dir	Platform Power	Type	Default Buffer State	
				Pwrgood Assert State	Resetout De-assert State
DDR3_M1_CK[1,0]_P	O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M1_CK[1,0]_N	O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M1_CKE[1:0]	O	VDDQ (V1P35)	DDR3	Weak 0	0
DDR3_M1_CS[1,0]_N	O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M1_CAS_N	O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M1_RAS_N	O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M1_WE_N	O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M1_BS[2:0]	O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M1_DRAMRST_N	O	VDDQ (V1P35)	DDR3	Weak 0	0
DDR3_M1_ODT[1,0]	O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M1_DQ[63:0]	I/O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M1_DM[7:0]	O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M1_DQS[7:0]_P	I/O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M1_DQS[7:0]_N	I/O	VDDQ (V1P35)	DDR3	Z	Z
DDR3_M1_OCAVREF	I	0.5*VDDQ	DDR3	Z	Z
DDR3_M1_ODQVREF	I	0.5*VDDQ	DDR3	Z	Z
DDR3_M1_RCOMP	I	VDDQ (V1P35)	DDR3	Z	Z
DDR3_DRAM_PWROK	I	VDDQ (V1P35)	DDR3	Input	Input
DDR3_VCCA_PWROK	I	VDDQ (V1P35)	DDR3	Input	Input

2.2.2 USB 2.0 Controller Interface Signals

Table 2-5. USB 2.0 Interface Signals

Signal Name	Dir	Platform Power	Type	Default Buffer State	
				Pwrgood Assert State	Resetout De-assert State
USB_DN[4:0]	I/O	V3P3A	USB2 PHY	"P" 1.1V	"P" 1.1V
USB_DP[4:0]	I/O	V3P3A	USB2 PHY	"P" 1.1V	"P" 1.1V
USB_RCOMP	I/O	V1P8A	USB2 PHY	Output	Output
USB_OC[1:0]_N	I/O	V1P8A	GPIOMV, MS	Input (20K PU)	Input (20K PU)

Note:
1. Depends on USB 2.0 Mode.

Table 2-6. USB 2.0 HSIC Interface Signals

Signal Name	Dir	Platform Power	Type	Default Buffer State	
				Pwrgood Assert State	Resetout De-assert State
USB_HSIC_[0:1]_DATA	I/O	V1P24A	HSIC Buffer	Weak 0	Weak 0
USB_HSIC_[0:1]_STROBE	I/O	V1P24A	HSIC Buffer	Weak 1	Weak 1
USB_HSIC_RCOMP	I/O	V1P24A	HSIC Buffer	Z	Z

Notes:
1. HSIC should only be used with USB Hubs. HSIC is not supported for individual USB devices. The HSIC should be reset after SoC.
2. Only 1x HSIC port should be used for external hub.

2.2.3 USB 3.0 Interface Signals

Table 2-7. USB 3.0 Interface Signals

Signal Name	Dir	Platform Power	Type	Default Buffer State	
				Pwrgood Assert State	Resetout De-assert State
USB3_TXN[3:0]	O	V1P05A	USB3 PHY	X	Z
USB3_TXP[3:0]	O	V1P05A	USB3 PHY	X	Z
USB3_RXN[3:0]	I	V1P05A	USB3 PHY	X	Z
USB3_RXP[3:0]	I	V1P05A	USB3 PHY	X	Z
USB3_RCOMP_N	I/O	V1P05A	USB3 PHY	X	Output
USB3_RCOMP_P	I/O	V1P05A	USB3 PHY	X	Output

2.2.4 Integrated Clock Interface Signals

Table 2-8. Integrated Clock Interface Signals

Signal Name	Dir	Platform Power	Type	Default Buffer State	
				Pwrgood Assert State	Resetout De-assert State
CLK_DIFF_N[0:3]	O	V1P05A	Analog	1	1
CLK_DIFF_P[0:3]	O	V1P05A	Analog	0	0
ICLK_OSCIN	I	V1P05A	Crystal Oscillator	Input (Crystal)	Input (Crystal)
ICLK_OSCOUT	O	V1P05A	Crystal Oscillator	Output (Crystal)	Output (Crystal)
ICLK_ICOMP	I/O	GND	Analog	Input	Input
ICLK_RCOMP	I/O	GND	Analog	Input	Input

2.2.5 Display—Digital Display Interface (DDI) Signals

Table 2-9. Digital Display Interface Signals (Sheet 1 of 2)

Signal Name	Dir	Platform Power	Type	Default Buffer State	
				Pwrgood Assert State	Resetout De-assert State
DDI0_TXP[3:0]	O	V1P35	MODPHY	Z	Output
DDI0_TXN[3:0]	O	V1P35	MODPHY	Z	Output
DDI0_AUXP	I/O	V1P35	MODPHY	Z	Output
DDI0_AUXN	I/O	V1P35	MODPHY	Z	Output
DDI0_RCOMP_N	I/O	V1P35	MODPHY	Z	Output
DDI0_RCOMP_P	I/O	V1P35	MODPHY	Z	Output
DDI0_DDC_CLK (multiplexed with DDI1_DDC_CLK)	I/O	V1P8A	GPIOMV, MS, CLK	Input (20k PU)	Input (20k PU)
DDI0_DDC_DATA (multiplexed with DDI1_DDC_DATA)	I/O	V1P8A	GPIOMV, MS	Input (20k PU)	Input (20k PU)
DDI0_HPD	I/O	V1P8A	GPIOMV, MS	Input (20k PD)	Input (20k PD)
DDI0_VDDEN	I/O	V1P8A	GPIOMV, MS	0	0
DDI0_BKLTCTL	I/O	V1P8A	GPIOMV, MS	Z	Output
DDI0_BKLTEN	I/O	V1P8A	GPIOMV, MS	Z	Output
DDI1_TXP[3:0]	O	V1P35	MODPHY	Z	Output
DDI1_TXN[3:0]	O	V1P35	MODPHY	Z	Output
DDI1_AUXP	I/O	V1P35	MODPHY	Z	Output
DDI1_AUXN	I/O	V1P35	MODPHY	Z	Output
DDI1_RCOMP_N	I/O	V1P35	MODPHY	Z	Output
DDI1_RCOMP_P	I/O	V1P35	MODPHY	Z	Output
DDI1_BKLTCTL	I/O	V1P8A	GPIOMV, MS	Z	Output
DDI1_BKLTEN	I/O	V1P8A	GPIOMV, MS	Z	Output
DDI1_DDC_CLK	I/O	V1P8A	GPIOMV, MS, CLK	Input (20k PU)	Input (20k PU)
DDI1_DDC_DATA	I/O	V1P8A	GPIOMV, MS	Input (20k PU)	Input (20k PU)
DDI1_HPD	I/O	V1P8A	GPIOMV, MS	Input (20k PD)	Input (20k PD)
DDI1_VDDEN	I/O	V1P8A	GPIOMV, MS	0	0
DDI2_TXP[3:0]	O	V1P35	MODPHY	Z	Output
DDI2_TXN[3:0]	O	V1P35	MODPHY	Z	Output
DDI2_AUXP	I/O	V1P35	MODPHY	Z	Output
DDI2_AUXN	I/O	V1P35	MODPHY	Z	Output

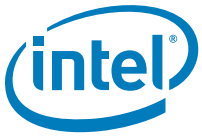


Table 2-9. Digital Display Interface Signals (Sheet 2 of 2)

Signal Name	Dir	Platform Power	Type	Default Buffer State	
				Pwrgood Assert State	Resetout De-assert State
DDI2_DDC_CLK	I/O	V1P8A	GPIOMV, MS, CLK	Input (20k PU)	Input (20k PU)
DDI2_DDC_DATA	I/O	V1P8A	GPIOMV, MS	Input (20k PU)	Input (20k PU)
DDI2_HPD	I/O	V1P8A	GPIOMV, MS	Input (20k PD)	Input (20k PD)

2.2.6 MIPI*-CSI (Camera Serial Interface) and ISP Interface Signals

Table 2-10. MIPI*-CSI Interface Signals

Signal Name	Dir	Platform Power	Type	Default Buffer State	
				Pwrgood Assert State	Resetout De-assert State
MCSI_1_CLKN	I	V1P24A	MIPI*-DPHY	Input	Input
MCSI_1_CLKP	I	V1P24A	MIPI*-DPHY	Input	Input
MCSI_1_DN[0:3]	I	V1P24A	MIPI*-DPHY	Input	Input
MCSI_1_DP[0:3]	I	V1P24A	MIPI*-DPHY	Input	Input
MCSI_2_CLKN	I	V1P24A	MIPI*-DPHY	Input	Input
MCSI_2_CLKP	I	V1P24A	MIPI*-DPHY	Input	Input
MCSI_2_DN[0:1]	I	V1P24A	MIPI*-DPHY	Input	Input
MCSI_2_DP[0:1]	I	V1P24A	MIPI*-DPHY	Input	Input
MCSI_3_CLKN	I	V1P24A	MIPI*-DPHY	Input	Input
MCSI_3_CLKP	I	V1P24A	MIPI*-DPHY	Input	Input
MCSI_RCOMP	I/O	V1P24A	MIPI*-DPHY	Z	Z
GP_CAMERASB[00:11]	I/O	V1P8A	GPIOMV, HS	Input (20k PD)	Input (20k PD)

2.2.7 Storage Controller Interface Signals

Table 2-11. Storage Controller (e-MMC*, SDIO, SD) Interface Signals (Sheet 1 of 2)

Signal Name	Dir	Platform Power	Type	Default Buffer State	
				Pwrgood Assert State	Resetout De-assert State
SDMMC1_D[7:0]	I/O	V1P8A	GPIOMV, HS	Z (20K PU)	Z (20K PU)
SDMMC1_CMD	I/O	V1P8A	GPIOMV, HS	Z (20K PU)	Z (20K PU)
SDMMC1_CLK	I/O	V1P8A	GPIOMV, HS, CLK	0 (20K PD)	0 (20K PD)
SDMMC1_RCLK	I/O	V1P8A	GPIOMV, HS	Z (20K PD)	Z

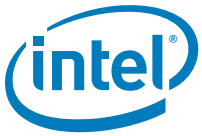
Table 2-11. Storage Controller (e-MMC*, SDIO, SD) Interface Signals (Sheet 2 of 2)

Signal Name	Dir	Platform Power	Type	Default Buffer State	
				Pwrgood Assert State	Resetout De-assert State
SDMMC1_RCOMP	I/O	V1P8A	GPIOMV, HS, RCOMP	Z	Z
SDMMC2_D[2:0]	I/O	V1P8A	GPIOMV, HS	Z (20K PU)	Z (20K PU)
SDMMC2_D[3]_CD_N	I/O	V1P8A	GPIOMV, HS	Z (20K PU)	Z (20K PU)
SDMMC2_CMD	I/O	V1P8A	GPIOMV, HS	Z (20K PU)	Z (20K PU)
SDMMC2_CLK	I/O	V1P8A	GPIOMV, HS, CLK	0 (20K PD)	0
SDMMC3_D[3:0]	I/O	V1P8A/ V3P3A	GPIOHV, HS	Z (20K PU)	Z (20K PU)
SDMMC3_CMD	I/O	V1P8A/ V3P3A	GPIOHV, HS	Z (20K PU)	Z (20K PU)
SDMMC3_PWREN_N	I/O	V1P8A	GPIOMV, MS	1 (20K PD)	1
SDMMC3_CLK	I/O	V1P8A/ V3P3A	GPIOHV, HS, CLK	0 (20K PD)	0
SDMMC3_RCOMP	I/O	V1P8A/ V3P3A	GPIOHV, HS, RCOMP	Z	Z
SDMMC3_1P8_EN	I/O	V1P8A	GPIOMV, MS	0 (20K PD)	0
SDMMC3_CD_N	I/O	V1P8A	GPIOMV, MS	Input (20K PU)	Input (20K PU)

2.2.8 High Speed UART Interface Signals

Table 2-12. High Speed UART Interface Signals

Signal Name	Dir	Platform Power	Type	Default Buffer State	
				Pwrgood Assert State	Resetout De-assert State
UART1_RXD	I/O	V1P8A	GPIOMV, MS	Input (20K PU)	Input (20K PU)
UART1_TXD	I/O	V1P8A	GPIOMV, MS	1 (20K PU)	1
UART1_RTS_N	I/O	V1P8A	GPIOMV, MS	1 (20K PU)	1
UART1_CTS_N	I/O	V1P8A	GPIOMV, MS	Input (20K PU)	Input (20K PU)
UART2_RXD	I/O	V1P8A	GPIOMV, MS	Input (20K PU)	Input (20K PU)
UART2_TXD	I/O	V1P8A	GPIOMV, MS	1 (20K PU)	1
UART2_RTS_N	I/O	V1P8A	GPIOMV, MS	1 (20K PU)	1
UART2_CTS_N	I/O	V1P8A	GPIOMV, MS	Input (20K PU)	Input (20K PU)



2.2.9 I²C Interface Signals

Table 2-13. I²C Interface Signals

Signal Name	Dir	Platform Power	Type	Default Buffer State	
				Pwrgood Assert State	Resetout De-assert State
I2C0_DATA	I/O	V1P8A	GPIOMV, MS, I2C	Z (1K PU, OD)	Z (1K PU, OD)
I2C0_CLK	I/O	V1P8A	GPIOMV, MS, I2C	Z (1K PU, OD)	Z (1K PU, OD)
I2C1_DATA	I/O	V1P8A	GPIOMV, MS, I2C	Input (20K PU)	Z (20K PU, OD)
I2C1_CLK	I/O	V1P8A	GPIOMV, MS, I2C	Input (20K PU)	Z (20K PU, OD)
I2C2_DATA	I/O	V1P8A	GPIOMV, MS, I2C	Input (20K PU)	Z (20K PU, OD)
I2C2_CLK	I/O	V1P8A	GPIOMV, MS, I2C	Input (20K PU)	Z (20K PU, OD)
I2C3_DATA	I/O	V1P8A	GPIOMV, MS, I2C	Input (20K PU)	Z (20K PU, OD)
I2C3_CLK	I/O	V1P8A	GPIOMV, MS, I2C	Input (20K PU)	Z (20K PU, OD)
I2C4_DATA	I/O	V1P8A	GPIOMV, MS, I2C	Input (20K PU)	Z (20K PU, OD)
I2C4_CLK	I/O	V1P8A	GPIOMV, MS, I2C	Input (20K PU)	Z (20K PU, OD)
I2C5_DATA	I/O	V1P8A	GPIOMV, MS, I2C	Input (20K PU)	Z (20K PU, OD)
I2C5_CLK	I/O	V1P8A	GPIOMV, MS, I2C	Input (20K PU)	Z (20K PU, OD)
I2C6_DATA	I/O	V1P8A	GPIOMV, MS, I2C	Input (20K PU)	Z (20K PU, OD)
I2C6_CLK	I/O	V1P8A	GPIOMV, MS, I2C	Input (20K PU)	Z (20K PU, OD)

2.3 SIO—Serial Peripheral Interface (SPI) Signals

Table 2-14. SIO—Serial Peripheral Interface (SPI) Signals

Signal Name	Dir	Platform Power	Type	Default Buffer State	
				Pwrgood Assert State	Resetout De-assert State
SPI1_CLK	O	V1P8A	GPIO	0 (20K PU)	0
SPI1_CS[0:1]_N	O	V1P8A	GPIO	1 (20K PU)	1
SPI1_MOSI	I	V1P8A	GPIO	0 (20K PU)	0
SPI1_MISO	O	V1P8A	GPIO	Input (20K PU)	Input (20K PD)
Note: SIO SPI is supported for non-Windows based platform only.					

2.3.1 PCU—Fast Serial Peripheral Interface (SPI) Signals

Table 2-15. PCU—Fast Serial Peripheral Interface (SPI) Signals

Signal Name	Dir	Plat. Power	Type	Default Buffer State	
				Pwrgood Assert State	Resetout De-assert State
FST_SPI_CLK	I/O	V1P8A	GPIOMV, HS	0 (20K PU)	Output
FST_SPI_CS[0]_N	I/O	V1P8A	GPIOMV, HS	1 (20K PU)	Output
FST_SPI_CS[1]_N	I/O	V1P8A	GPIOMV, HS	Input (20K PU)	Output
FST_SPI_CS[2]_N	I/O	V1P8A	GPIOMV, HS	1 (20K PU)	Output
FST_SPI_D[0:3]	I/O	V1P8A	GPIOMV, HS	Input (20K PU)	Input (20K PU)

Note: Flash Sharing is not supported for the processor Platform.

Note: The FST_SPI_CS0, FST_SPI_CS2, and FST_SPI_CLK signals do not get Tri-Stated during RSMRST_N assertion.

2.3.2 PCU—Real Time Clock (RTC) Interface Signals

Table 2-16. PCU—Real Time Clock (RTC) Interface Signals

Signal Name	Dir	Platform Power	Type	Default Buffer State	
				Pwrgood Assert State	Resetout De-assert State
RTC_X1	I	V3P3_RTC	RTC PHY	Input (Crystal)	Input (Crystal)
RTC_X2	O	V3P3_RTC	RTC PHY	Output (Crystal)	Output (Crystal)
RTC_RST_N	I	V3P3_RTC	RTC PHY	Input	Input
RTC_TEST_N	I	V3P3_RTC	RTC PHY	Input	Input
RTC_EXTPAD	O	V3P3_RTC	RTC PHY	Output	Output
CORE_PWROK	I	V3P3_RTC	RTC PHY	Input	Input
PMU_RSMRST_N	I	V3P3_RTC	RTC PHY	Input	Input

2.3.3 PCU—Low Pin Count (LPC) Bridge Interface Signals

Table 2-17. PCU—LPC Bridge Interface Signals (Sheet 1 of 2)

Signal Name	Dir	Platform Power	Type	Default Buffer State	
				Pwrgood Assert State	Resetout De-assert State
LPC_AD[0:3]	I/O	V3P3A/ V1P8A	GPIOHV, HS	Input (20K PU)	Input (20K PU)
LPC_FRAME_N	I/O	V3P3A/ V1P8A	GPIOHV, HS	1 (20K PU)	1
LPC_SERIRQ	I/O	V1P8A	GPIOMV, MS	Input (20K PU)	Input (20K PU)

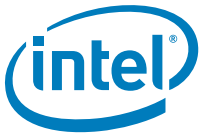


Table 2-17. PCU—LPC Bridge Interface Signals (Sheet 2 of 2)

				Default Buffer State	
Signal Name	Dir	Platform Power	Type	Pwrgood Assert State	Resetout De-assert State
LPC_CLKRUN_N	I/O	V3P3A/ V1P8A	GPIOHV, HS	Input (20K PU)	Input (20K PU)
LPC_CLK[0]	I/O	V3P3A/ V1P8A	GPIOHV, HS	0 (20K PU)	Clock
LPC_CLK[1]	I/O	V3P3A/ V1P8A	GPIOHV, HS	Input (20K PD)	Input
LPC_RCOMP	I/O	V3P3A/ V1P8A	GPIOHV, HS, RCOMP	Z	Z

2.3.4 JTAG Interface Signals

Table 2-18. JTAG Interface Signals

				Default Buffer State	
Signal Name	Dir	Platform Power	Type	Pwrgood Assert State	Resetout De-assert State
JTAG_TCK	I/O	V1P8A	GPIOMV, MS	Input (5K PD)	Input (5K PD)
JTAG_TDI	I/O	V1P8A	GPIOMV, MS	Input (5K PU)	Input (5K PU)
JTAG_TDO	I/O	V1P8A	GPIOMV, MS	Z	Z
JTAG_TMS	I/O	V1P8A	GPIOMV, MS	Input (5K PU)	Input (5K PU)
JTAG_TRST_N	I/O	V1P8A	GPIOMV, MS	Input (5K PU)	Input (5K PU)
JTAG_PRDY_N	I/O	V1P8A	GPIOMV, MS	Z (5K PU, OD)	Output (5K PU, OD)
JTAG_PREQ_N	I/O	V1P8A	GPIOMV, MS	Input (5K PU, OD)	Input (5K PU, OD)

2.3.5 PCI Express* (PCIe*) Signals

Table 2-19. PCI Express* (PCIe*) Signals and Clocks

				Default Buffer State	
Signal Name	Dir	Platform Power	Type	Pwrgood Assert State	Resetout De-assert State
PCIE_RXN[0:3]	I	V1P05A	PCIe* PHY	X	Weak pull-down
PCIE_RXP[0:3]	I	V1P05A	PCIe* PHY	X	Weak pull-down
PCIE_TXN[0:3]	O	V1P05A	PCIe* PHY	X	Z
PCIE_TXP[0:3]	O	V1P05A	PCIe* PHY	X	Z
PCIE_RCOMP_N	I/O	V1P05A	PCIe* PHY	X	Output
PCIE_RCOMP_P	I/O	V1P05A	PCIe* PHY	X	Output
PCIE_CLKREQ[0:3]_N	I/O	V1P8A	GPIOMV, MS	Input (20K PU)	Input (20K PU)

2.3.6 SATA Signals

Table 2-20. SATA Signals and Clocks

Signal Name	Dir	Platform Power	Type	Default Buffer State	
				Pwrgood Assert State	Resetout De-assert State
SATA_GP[0:1]	I/O	V1P8A	GPIOMV, MS	Input (20K PD)	Input (20K PD)
SATA_GP[2:3]	I/O	V1P8A	GPIOMV, MS	0 (20K PD)	0 (20K PD)
SATA_LED_N	I/O	V1P8A	GPIOMV, MS	1 (20K PD)	Prg
SATA_RXN[0:1]	I/O	V1P05A	SATA PHY	X	Weak pull-down
SATA_RXP[0:1]	I/O	V1P05A	SATA PHY	X	Weak pull-down
SATA_TXN[0:1]	I/O	V1P05A	SATA PHY	X	Z
SATA_TXP[0:1]	I/O	V1P05A	SATA PHY	X	Z
SATA_RCOMP_N	I/O	V1P05A	SATA PHY	X	Output
SATA_RCOMP_P	I/O	V1P05A	SATA PHY	X	Output

2.3.7 SMBus Signals

Table 2-21. SMBus Signals and Clocks

Signal Name	Dir	Platform Power	Type	Default Buffer State	
				Pwrgood Assert State	Resetout De-assert State
MF_SMB_ALERT_N	I/O	V1P8A	GPIOMV, MS	Input (20K PU)	Input (20K PU)
MF_SMB_CLK	I/O	V1P8A	GPIOMV, MS	Input (20K PU)	Input (20K PU)
MF_SMB_DATA	I/O	V1P8A	GPIOMV, MS	Input (20K PU)	Input (20K PU)

2.3.8 Intel® High Definition Audio (Intel® HD Audio) Signals

Table 2-22. Intel® High Definition Audio (Intel® HD Audio) Signals and Clocks (Sheet 1 of 2)

Signal Name	Dir	Platform Power	Type	Default Buffer State	
				Pwrgood Assert State	Resetout De-assert State
MF_HDA_CLK	I/O	V1P5A or V1P8A	GPIOMV, MS	0 (20K PD)	0 (20K PD)
MF_HDA_DOCKEN_N	I/O	V1P5A or V1P8A	GPIOMV, MS	Input (20K PD)	Input (20K PD)
MF_HDA_DOCKRST_N	I/O	V1P5A or V1P8A	GPIOMV, MS	0 (20K PD)	0 (20K PD)
MF_HDA_RST_N	I/O	V1P5A or V1P8A	GPIOMV, MS	Input (20K PD)	Input (20K PD)

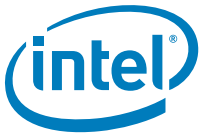


Table 2-22. Intel® High Definition Audio (Intel® HD Audio) Signals and Clocks (Sheet 2 of 2)

				Default Buffer State	
Signal Name	Dir	Platform Power	Type	Pwrgood Assert State	Resetout De-assert State
MF_HDA_SDI[0:1]	I/O	V1P5A or V1P8A	GPIOMV, MS	Input (20K PD)	Input (20K PD)
MF_HDA_SDO	I/O	V1P5A or V1P8A	GPIOMV, MS	Input (20K PD)	Input (20K PD)
MF_HDA_SYNC	I/O	V1P5A or V1P8A	GPIOMV, MS	Input (20K PD)	Input (20K PD)

Notes: The processor HD Audio logic buffers can support 1.8V. However, functionality with 1.8V has not been validated.

2.3.9 Power Management Unit (PMU) Signals

Table 2-23. Power Management Unit (PMU) Signals and Clocks

				Default Buffer State	
Signal Name	Dir	Platform Power	Type	Pwrgood Assert State	Resetout De-assert State
PMU_AC_PRESENT	I/O	V1P8A	GPIOMV, MS	Input (20K PD)	Input (20K PD)
PMU_BATLOW_N	I/O	V1P8A	GPIOMV, MS	Input (20K PU)	Input (20K PU)
PMU_PLTRST_N	I/O	V1P8A	GPIOMV, MS	0 (20K PU)	1
PMU_PWRBTN_N	I	V1P8A	GPIOMV, MS	Input (20K PU)	Input (20K PU)
PMU_RESETBUTTON_N	I/O	V1P8A	GPIOMV, MS	Input (20K PU)	Input (20K PU)
PMU_SLP_LAN_N	I/O	V1P8A	GPIOMV, MS	0 (20K PU)	0
PMU_WAKE_LAN_N	I	V1P8A	GPIOMV, MS	Input (20K PU)	Input (20K PU)
PMU_SLP_S3_N	I/O	V1P8A	GPIOMV, MS	0 (20K PU)	1
PMU_SLP_S4_N	I/O	V1P8A	GPIOMV, MS	0 (20K PU)	1
PMU_WAKE_N	I/O	V1P8A	GPIOMV, MS	Input (20K PU)	Input (20K PU)
PMU_SUSCLK	I/O	V1P8A	GPIOMV, MS	0 (20K PU)	32 KHz ClocK
SUS_STAT_N	I/O	V1P8A	GPIOMV, MS	0 (20K PU)	1
SUSPWRDNACK	I/O	V1P8A	GPIOMV, MS	0 (20K PD)	0 (20K PD)

2.3.10 Speaker Signals

Table 2-24. SPEAKER Signals and Clocks

				Default Buffer State	
Signal Name	Dir	Platform Power	Type	Pwrgood Assert State	Resetout De-assert State
SPKR	I/O	V1P8A	GPIO	0 (20K PU)	Prg

2.3.11 Miscellaneous Signals

Table 2-25. Miscellaneous Signals and Clocks

Signal Name	Dir.	Platform Power	Type	Default Buffer State	
				Pwrgood Assert State	Resetout De-assert State
SVID0_DATA	I/O	V1P05A	GPIOMV, MS	0	0
SVID0_CLK	O	V1P05A	GPIOMV, MS	0	1 or Z
SVID0_ALERT_N	I/O	V1P05A	GPIOMV, MS	Input	Input
PROCHOT_N	I/O	V1P8A	GPIOMV, MS	Z	Z
PLT_CLK[0:5]	O	V1P8A	GPIOMV, MS	0 (20k PD)	Clock (20K PD)

2.4 Hardware Straps

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

Table 2-26. Hard Strap Description and Functionality (Sheet 1 of 2)

Signal Name	Purpose	Pull-Up/Pull-Down	Strap Description
GPIO_SUS[0]	DDI0 Detect	Weak internal (20k PD)	0 = DDI0 not detected 1 = DDI0 detected
GPIO_SUS[1]	DDI1 Detect	Weak internal (20k PD)	0 = DDI1 not detected 1 = DDI1 detected
GPIO_SUS[2]	Top Swap (A16) override	Weak internal (20k PU)	0 = Change Boot Loader address 1 = Normal Operation
GPIO_SUS[3]	MIPI-DSI Display Detect	Weak internal (20k PD)	0 = DSI Port not detected 1= DSI Port detected ***Note:*** DSI is not supported for the processor. This strap will not enable DSI on the processor. Leave the pin floating if GPIO functionality is not used.
GPIO_SUS[4]	Boot BIOS Strap (BBS)	Weak internal (20k PU)	0 = No SPI (Default) 1 = SPI
GPIO_SUS[5]	Flash Descriptor Security Override	Weak internal (20k PU)	0 = Override 1 = Normal Operation
GPIO_SUS[6]	Halt Boot Strap	Weak internal (20k PU)	1= Normal Operation ***Note:*** This strap **MUST** be High at RSMRST_N de-assert to ensure proper platform operation and use of GPIO_DFX[8:0]
GPIO_SUS[8]	PLLs, ICLK, USB2, DDI SFR Supply Select	Weak internal (20k PU)	0 = Supply is 1.25V 1 = Supply is 1.35V
GPIO_SUS[9]	ICLK, USB2, DDI SFR Bypass	Weak internal (20k PD)	0 = No bypass 1 = Bypass with 1.05V
GPIO_CAMERASB08	ICLK Xtal OSC Bypass	Weak internal (20k PD)	0 = No Bypass (Default) 1 = Bypass

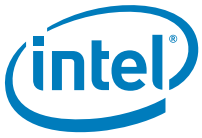


Table 2-26. Hard Strap Description and Functionality (Sheet 2 of 2)

Signal Name	Purpose	Pull-Up/Pull-Down	Strap Description
GPIO_CAMERASB09	CCU SUS RO Bypass	Weak internal (20k PD)	0 = No Bypass (Default) 1 = Bypass
GPIO_CAMERASB11	RTC OSC Bypass	Weak internal (20k PD)	0 = No Bypass (Default) 1 = Bypass

Notes:
1. All straps are sampled on the rising edge (de-assertion) of PMU_RSMRST_N
2. For proper operation of GPIO functionality, ensure that there is no contention or conflict between the Hard Strap selection and GPIO direction.
3. It is imperative that GPIO_SUS[6] is sampled High (Logic 1) at PMC_RSMRST_N to ensure proper system and GPIO functionality.
4. There is no Hard Strap for DDI2 Detection. Enabling DDI2 does not require a Hard Strap, it is always enabled by default whether it is used for HDMI, AVI or DP. (eDP* can only be used on DDI0 and DDI1)

2.5 GPIO Multiplexing

GPIO (General Purpose IO) are provided for added design flexibility. There are 192 GPIOs on the processor and all these signals can be used as GPIO. Each of these signals has a "default" mode and function based on SoC design. These pins have multiple functionality (modes) depending on the configuration done through the BIOS. Configuration of these pins as GPIOs is also done through BIOS.The list in the following table provides the details on the specifications of the GPIO signals. For more details on the GPIO Configuration Registers, refer to the processor Datasheet Volume 3, Sections 35.5–35.9.

Note: All the GPIOs listed here are powered by the 1.8VA (always on) rail. Depending on the design implementation there may be some amount of power leakage observed in Sx state due to (a) Pull-up on un-powered devices and (b) Driving "High" into an un-powered device. Adding BIOS patch for those GPIOs before entering Sx state is applicable to reduce power leakage scenarios.

Note: All GPIO signals have weak internal terminations and unused pins do not need to be terminated on the platform. If they are terminated, then BIOS needs to disable the internal terminations to avoid any issues related to leakage.

Note: Default Function for GPIO_DFX[8:0] is listed as RSVD, but they can be used for normal GPIO functionality.

Note: The following features are not supported for the processor but the related signals have been listed in this table since they can be used as normal GPIOs.

ISH (Integrated Sensor Hub), PWM (Pulse Width Modulator), SIO (Serial IO) SPI (may be supported for some non-Windows operating systems), Connected Standby (S0ix related signals)

Note: Ensure that PMU_RESETBUTTON_N is used as native functionality. If used improperly, can cause the SoC to reset.

Table 2-27. GPIO Multiplexing and Modes (Sheet 1 of 6)

Count	SoC Pin No.	CFIO Name	Default Mode	Default Function	GPIO SoC Power Rail	Pwrgood Assert State	Resetout De-assert State	Optional Modes/ Direction
1	AB41	GP_CAMERASB00	1	GP_CAMERASB00	V1P8A	Input (20K PD)	Input (20k PD)	
2	AB45	GP_CAMERASB01	1	GP_CAMERASB01	V1P8A	Input (20k PD)	Input (20k PD)	
3	Y41	GP_CAMERASB10	3	RSVD Input	V1P8A	Input (20k PD)	Input (20k PD)	
4	V40	GP_CAMERASB11	1	GP_CAMERASB11	V1P8A	Input (20k PD)	Input (20k PD)	
5	AB44	GP_CAMERASB02	1	GP_CAMERASB02	V1P8A	Input (20k PD)	Input (20k PD)	
6	AC53	GP_CAMERASB03	1	GP_CAMERASB03	V1P8A	Input (20k PD)	Input (20k PD)	
7	AB51	GP_CAMERASB04	1	GP_CAMERASB04	V1P8A	Input (20k PD)	Input (20k PD)	
8	AB52	GP_CAMERASB05	1	GP_CAMERASB05	V1P8A	Input (20k PD)	Input (20k PD)	
9	AA51	GP_CAMERASB06	1	GP_CAMERASB06	V1P8A	Input (20k PD)	Input (20k PD)	
10	AB40	GP_CAMERASB07	1	GP_CAMERASB07	V1P8A	Input (20k PD)	Input (20k PD)	
11	Y44	GP_CAMERASB08	1	GP_CAMERASB08	V1P8A	Input (20k PD)	Input (20k PD)	
12	Y42	GP_CAMERASB09	1	GP_CAMERASB09	V1P8A	Input (20k PD)	Input (20k PD)	
13	W51	HV_DDI0_HPD	1	HV_DDI0_HPD	V1P8A	Input (20k PD)	Input (20k PD)	
14	Y51	HV_DDI0_DDC_SCL	1	HV_DDI0_DDC_SCL	V1P8A	Input (20k PU)	Input (20k PU)	Mode2/ HV_DDI1_DDC_SCL/IO
15	Y52	HV_DDI0_DDC_SDA	1	HV_DDI0_DDC_SDA	V1P8A	Input (20k PU)	Input (20k PU)	Mode 2/ HV_DDI1_DDC_SDA/IO

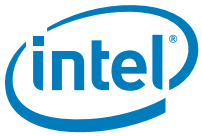


Table 2-27. GPIO Multiplexing and Modes (Sheet 2 of 6)

Count	SoC Pin No.	CFIO Name	Default Mode	Default Function	GPIO SoC Power Rail	Pwrgood Assert State	Resetout De-assert State	Optional Modes/ Direction
16	V51	PANEL0_BKLTCTL	1	PANEL0_BKLTCTL	V1P8A	0	0	
17	V52	PANEL0_BKLTEN	1	PANEL0_BKLTEN	V1P8A	0	0	
18	W53	PANEL0_VDDEN	1	PANEL0_VDDEN	V1P8A	0	0	
19	R51	HV_DDI1_HPD	1	HV_DDI1_HPD	V1P8A	Input (20k PD)	Input (20k PD)	
20	P52	PANEL1_BKLTCTL	1	PANEL1_BKLTCTL	V1P8A	0	0	
21	P51	PANEL1_BKLTEN	1	PANEL1_BKLTEN	V1P8A	0	0	
22	R53	PANEL1_VDDEN	1	PANEL1_VDDEN	V1P8A	0	0	
23	U51	HV_DDI2_HPD	1	HV_DDI2_HPD	V1P8A	Input (20k PD)	Input (20k PD)	
24	T51	HV_DDI2_DDC_SCL	1	HV_DDI2_DDC_SCL	V1P8A	Input (20k PU)	Input (20k PU)	Mode2/ HV_DDI1_DDC_SCL/IO Mode3/UART0_TXD/O
25	T52	HV_DDI2_DDC_SDA	1	HV_DDI2_DDC_SDA	V1P8A	Input (20k PU)	Input (20k PU)	Mode2/ HV_DDI1_DDC_SDA/IO Mode3/UART0_RXD/I
26	M7	SDMMC1_CLK	1	SDMMC1_CLK	V1P8A	0 (20k PD)	0 (20k PD)	
27	P6	SDMMC1_CMD	1	SDMMC1_CMD	V1P8A	Z (20K PU)	Z (20K PU)	
28	M6	SDMMC1_D0	1	SDMMC1_D0	V1P8A	Z (20k PU)	Z (20k PU)	
29	M4	SDMMC1_D1	1	SDMMC1_D1	V1P8A	Z (20k PU)	Z (20k PU)	
30	P9	SDMMC1_D2	1	SDMMC1_D2	V1P8A	Z (20k PU)	Z (20k PU)	
31	P7	SDMMC1_D3_CD_N	1	SDMMC1_D3_CD_N	V1P8A	Z (20k PU)	Z (20k PU)	
32	T6	MMC1_D4_SD_WE	1	MMC1_D4_SD_WE	V1P8A	Z (20k PU)	Z (20k PU)	
33	T7	SDMMC1_D5	1	SDMMC1_D5	V1P8A	Z (20k PU)	Z (20k PU)	
34	T10	SDMMC1_D6	1	SDMMC1_D6	V1P8A	Z (20k PU)	Z (20k PU)	
35	T12	SDMMC1_D7	1	SDMMC1_D7	V1P8A	Z (20k PU)	Z (20k PU)	
36	T13	SDMMC1_RCLK	1	SDMMC1_RCLK	V1P8A	Z (20k PD)	Z	
37	W3	FST_SPI_CLK	1	FST_SPI_CLK	V1P8A	0 (20k PU)	Output	
38	V4	FST_SPI_CS0_N	1	FST_SPI_CS0_N	V1P8A	1 (20k PU)	Output	
39	V6	FST_SPI_CS1_N	1	FST_SPI_CS1_N	V1P8A	Input (20k PU)	Output	
40	V7	FST_SPI_CS2_N	1	FST_SPI_CS2_N	V1P8A	1 (20k PU)	Output	
41	V2	FST_SPI_D0	1	FST_SPI_D0	V1P8A	Input (20k PU)	Input (20k PU)	
42	V3	FST_SPI_D1	1	FST_SPI_D1	V1P8A	Input (20k PU)	Input (20k PU)	
43	U1	FST_SPI_D2	1	FST_SPI_D2	V1P8A	Input (20k PU)	Input (20k PU)	
44	U3	FST_SPI_D3	1	FST_SPI_D3	V1P8A	Input (20k PU)	Input (20k PU)	
45	AM40	GPIO_DFX0	1	RSVD Inputs	V1P8A	Input (20k PD)	Input (20k PD)	Mode5/C0_BPM0_TX/ DFX IO Mode6/C1_BPM0_TX/ DFX IO
46	AM41	GPIO_DFX1	1	RSVD Inputs	V1P8A	Input (20k PD)	Input (20k PD)	Mode5/C0_BPM1_TX/ DFX O Mode6/C1_BPM1_TX/ DFX O
47	AM44	GPIO_DFX2	1	RSVD Inputs	V1P8A	Input (20k PD)	Input (20k PD)	Mode5/C0_BPM2_TX/ DFX O Mode6/C1_BPM2_TX/ DFX O
48	AM45	GPIO_DFX3	1	RSVD Inputs	V1P8A	Input (20k PD)	Input (20k PD)	Mode5/C0_BPM3_TX/ DFX IO Mode6/C1_BPM3_TX/ DFX IO
49	AM47	GPIO_DFX4	1	RSVD Inputs	V1P8A	Input (20k PD)	Input (20k PD)	
50	AK48	GPIO_DFX5	1	RSVD Inputs	V1P8A	Input (20k PU)	Input (20k PU)	Mode5/C0_BPM0_TX/ DFX IO Mode6/C1_BPM0_TX/ DFX IO

Table 2-27. GPIO Multiplexing and Modes (Sheet 3 of 6)

Count	SoC Pin No.	CFIO Name	Default Mode	Default Function	GPIO SoC Power Rail	Pwrgood Assert State	Resetout De-assert State	Optional Modes/ Direction
51	AM48	GPIO_DFX6	1	RSVD Inputs	V1P8A	Input (20k PU)	Input (20k PU)	Mode5/C0_BPM1_TX/ DFX O Mode6/C1_BPM1_TX/ DFX O Mode8/IERR/O
52	AK41	GPIO_DFX7	1	RSVD Inputs	V1P8A	Input (20k PU)	Input (20k PU)	Mode5/C0_BPM2_TX/ DFX O Mode6/C1_BPM2_TX/ DFX O
53	AK42	GPIO_DFX8	1	RSVD Inputs	V1P8A	Input (20k PU)	Input (20k PU)	Mode5/C0_BPM3_TX/ DFX IO Mode6/C1_BPM3_TX/ DFX IO
54	AD51	GPIO_SUS0	GPI	GPIO_SUS0	V1P8A	Input (20k PD)	Input (20k PD)	
55	AD52	GPIO_SUS1	1	GPIO_SUS1	V1P8A	Input (20k PD)	Input (20k PD)	Mode6/PCI_WAKE1_N/I
56	AH50	GPIO_SUS2	1	GPIO_SUS2	V1P8A	Input (20k PU)	Input (20k PU)	Mode6/PCI_WAKE2_N/I
57	AH48	GPIO_SUS3	1	GPIO_SUS3	V1P8A	Input (20k PD)	Input (20k PD)	Mode6/PCI_WAKE3_N/I
58	AH51	GPIO_SUS4	1	GPIO_SUS4	V1P8A	Input (20k PU)	Input (20k PU)	Mode6/PCI_WAKE4_N/I
59	AH52	GPIO_SUS5	GPI	GPIO_SUS5	V1P8A	Input (20k PU)	Input (20k PU)	
60	AG51	GPIO_SUS6	GPI	GPIO_SUS6	V1P8A	Input (20k PU)	Input (20k PU)	
61	AG53	GPIO_SUS7	GPI	GPIO_SUS7	V1P8A	Input (20k PU)	Input (20k PU)	
62	AF51	SEC_GPIO_SUS8	1	SEC_GPIO_SUS8	V1P8A	Input (20k PU)	Input (20k PU)	
63	AF52	SEC_GPIO_SUS9	GPI	SEC_GPIO_SUS9	V1P8A	Input (20k PD)	Input (20k PD)	
64	AE51	SEC_GPIO_SUS10	GPI	CSE_GPIO_SUS10	V1P8A	Input (20k PD)	Input (20k PD)	
65	AC51	SEC_GPIO_SUS11	1	SEC_GPIO_SUS11	V1P8A	0 (20k PD)	0	
66	AD9	MF_HDA_CLK	1	GP_SSP_0_I2S_TXD	V1P8A/ V1P5A	0 (20k PD)	0 (20k PD)	Mode2/HDA_CLK/O
67	AB9	MF_HDA_DOCKEN_N	1	GP_SSP_1_I2S_RXD	V1P8A/ V1P5A	Input (20k PD)	Input (20k PD)	
68	AB7	MF_HDA_DOCKRST_N	1	GP_SSP_1_I2S_TXD	V1P8A/ V1P5A	0 (20k PD)	0 (20k PD)	
69	AF13	MF_HDA_RST_N	1	GP_SSP_0_I2S_CLK	V1P8A/ V1P5A	Input (20k PD)	Input (20k PD)	Mode2/HDA_RSTB/I
70	AD7	MF_HDA_SDI0	1	GP_SSP_1_I2S_CLK	V1P8A/ V1P5A	Input (20k PD)	Input (20k PD)	Mode2/HDA_SDI0/O
71	AD6	MF_HDA_SDI1	1	GP_SSP_1_I2S_FS	V1P8A/ V1P5A	Input (20k PD)	Input (20k PD)	Mode2/HDA_SDI1/O
72	AF14	MF_HDA_SDO	1	GP_SSP_0_I2S_RXD	V1P8A/ V1P5A	Input (20k PD)	Input (20k PD)	Mode2/HDA_SDO/O
73	AF12	MF_HDA_SYNC	1	GP_SSP_0_I2S_FS	V1P8A/ V1P5A	Input (20k PD)	Input (20k PD)	Mode2/HDA_SYNC/I
74	AD13	UART1_CTS_N	1	UART1_CTS_N	V1P8A	Input (20k PU)	Input (20k PU)	
75	AD14	UART1_RTS_N	1	UART1_RTS_N	V1P8A	1 (20k PU)	1	
76	AD12	UART1_RXD	1	UART1_RXD	V1P8A	Input (20k PU)	Input (20k PU)	Mode2/UART0_RXD/I
77	AD10	UART1_TXD	1	UART1_TXD	V1P8A	1 (20k PU)	1	Mode2/UART0_TXD/O
78	V9	UART2_CTS_N	1	UART2_CTS_N	V1P8A	Input (20k PU)	Input (20k PU)	
79	V10	UART2_RTS_N	1	UART2_RTS_N	V1P8A	1 (20k PU)	1	
80	Y7	UART2_RXD	1	UART2_RXD	V1P8A	Input (20k PU)	Input (20k PU)	
81	Y6	UART2_TXD	1	UART2_TXD	V1P8A	1 (20k PU)	1	
82	Y3	GPIO_ALERT	1	GPIO_ALERT	V1P8A	0 (20k PU)	0	
83	AK6	I2C0_SCL	1	I2C0_SCL	V1P8A	Z (1k PU, OD) input	Z (1k PU, OD)	
84	AH7	I2C0_SDA	1	I2C0_SDA	V1P8A	Z (1k PU, OD) input	Z (1k PU, OD)	
85	J14	I2C1_SCL	1	I2C1_SCL	V1P8A	Input (20k PU)	Z (20k PU, OD)	

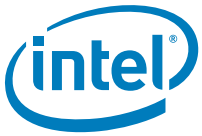


Table 2-27. GPIO Multiplexing and Modes (Sheet 4 of 6)

Count	SoC Pin No.	CFIO Name	Default Mode	Default Function	GPIO SoC Power Rail	Pwrgood Assert State	Resetout De-assert State	Optional Modes/ Direction
86	AH6	I2C1_SDA	1	I2C1_SDA	V1P8A	Input (20k PU)	Z (20k PU, OD)	
87	AF9	I2C2_SCL	1	I2C2_SCL	V1P8A	Input (20k PU)	Z (20k PU, OD)	
88	AF7	I2C2_SDA	1	I2C2_SDA	V1P8A	Input (20k PU)	Z (20k PU, OD)	
89	AE4	I2C3_SCL	1	I2C3_SCL	V1P8A	Input (20k PU)	Z (20k PU, OD)	
90	AD2	I2C3_SDA	1	I2C3_SDA	V1P8A	Input (20k PU)	Z (20k PU, OD)	
91	AC1	I2C4_SCL	1	I2C4_SCL	V1P8A	Input (20k PU)	Z (20k PU, OD)	Mode2/ HV_DDI0_DDC_SCL/O Mode3/ HV_DDI2_DDC_SCL/O
92	AD3	I2C4_SDA	1	I2C4_SDA	V1P8A	Input (20k PU)	Z (20k PU, OD)	Mode2/ HV_DDI0_DDC_SDA/IO Mode3/ HV_DDI2_DDC_SDA/IO
93	AB2	I2C5_SCL	1	I2C5_SCL	V1P8A	Input (20k PU)	Z (20k PU, OD)	
94	AC3	I2C5_SDA	1	I2C5_SDA	V1P8A	Input (20k PU)	Z (20k PU, OD)	
95	AA1	I2C6_SCL	1	I2C6_SCL	V1P8A	Input (20k PU)	Z (20k PU, OD)	Mode2/NMI_N
96	AB3	I2C6_SDA	1	I2C6_SDA	V1P8A	Input (20k PU)	Z (20k PU, OD)	Mode2/SDMMC3_WP/I
97	C11	MF_ISH_GPIO_0	1	ISH_GPIO_0	V1P8A	Input (20k PD)	Z (20k PU)	
98	B10	MF_ISH_GPIO_1	1	ISH_GPIO_1	V1P8A	Input (20k PD)	Z (20k PU)	
99	F12	MF_ISH_GPIO_2	1	ISH_GPIO_2	V1P8A	Input (20k PD)	Z (20k PU)	
100	F10	MF_ISH_GPIO_3	1	ISH_GPIO_3	V1P8A	Input (20k PD)	Z (20k PU)	
101	D12	MF_ISH_GPIO_4	1	ISH_GPIO_4	V1P8A	Input (20k PD)	Z (20k PU)	
102	E8	MF_ISH_GPIO_5	1	ISH_GPIO_5	V1P8A	Input (20k PD)	Z (20k PU)	
103	C7	MF_ISH_GPIO_6	1	ISH_GPIO_6	V1P8A	Input (20k PD)	Z (20k PU)	
104	D6	MF_ISH_GPIO_7	1	ISH_GPIO_7	V1P8A	Input (20k PD)	Z (20k PU)	
105	J12	MF_ISH_GPIO_8	1	ISH_GPIO_8	V1P8A	Input (20k PU)	Z (20k PU, OD)	
106	F7	MF_ISH_GPIO_9	1	ISH_GPIO_9	V1P8A	Input (20k PU)	Z (20k PU, OD)	
107	L13	MF_ISH_I2C1_SDA	1	ISH_I2C1_SDA	V1P8A	Input (20k PU)	Z (20k PU, OD)	
108	J14	MF_ISH_I2C1_SCL	1	ISH_I2C1_SCL	V1P8A	Input (20k PU)	Z (20k PU, OD)	
109	AD45	CX_PRDY_N	1	PRDY_N	V1P8A	Z (5k PU, OD)	Output (5k PU, OD)	
110	AF41	CX_PREQ_N	1	PREQ_N	V1P8A	Input (5k PU, OD)	Input (5k PU, OD)	
111	AF42	TCK	1	TCK	V1P8A	Input (5k PD)	Input (5k PD)	
112	AD47	TDI	1	TDI	V1P8A	Input (5k PU)	Input (5k PU)	
113	AF40	TDO	1	TDO	V1P8A	Z	Z	
114	AD48	TMS	1	TMS	V1P8A	Input (5k PU)	Input (5k PU)	
115	AB48	TRST_N	1	TRST_N	V1P8A	Input (5k PU)	Input (5k PU)	
116	T2	ILB_SERIRQ	Z	ILB_SERIRQ	V1P8A	Input (20k PU)	Input (20k PU)	
117	P2	MF_LPC_CLKOUT0	1	LPC_CLKOUT0	V3P3A/ V1P8A	0 (20k PD)	SLP	
118	R3	MF_LPC_CLKOUT1	1	LPC_CLKOUT1	V3P3A/ V1P8A	Input (20k PD)	Input	
119	M3	MF_LPC_AD0	1	LPC_AD0	V3P3A/ V1P8A	Input (20k PU)	Input (20k PU)	

Table 2-27. GPIO Multiplexing and Modes (Sheet 5 of 6)

Count	SoC Pin No.	CFIO Name	Default Mode	Default Function	GPIO SoC Power Rail	Pwrgood Assert State	Resetout De-assert State	Optional Modes/ Direction
120	M2	MF_LPC_AD1	1	LPC_AD1	V3P3A/ V1P8A	Input (20k PU)	Input (20k PU)	
121	N3	MF_LPC_AD2	1	LPC_AD2	V3P3A/ V1P8A	Input (20k PU)	Input (20k PU)	
122	N1	MF_LPC_AD3	1	LPC_AD3	V3P3A/ V1P8A	Input (20k PU)	Input (20k PU)	
123	T3	LPC_CLKRUN_N	1	LPC_CLKRUN_N	V3P3A/ V1P8A	Input (20k PU)	Input (20k PU)	Mode2/UART0_TXD/O
124	P3	LPC_FRAME_N	1	LPC_FRAME_N	V3P3A/ V1P8A	1 (20k PU)	1	Mode2/UART0_RXD/I
125	AK9	GP_SSP_2_CLK	1	GP_SSP_2_I2S_CLK	V1P8A	Input (20k PD)	Input (20k PD)	
126	AK10	GP_SSP_2_FS	1	GP_SSP_2_I2S_FS	V1P8A	Input (20k PD)	Input (20k PD)	
127	AK13	GP_SSP_2_RXD	1	GP_SSP_2_I2S_RXD	V1P8A	Input (20k PD)	Input (20k PD)	
128	AK12	GP_SSP_2_TXD	1	GP_SSP_2_I2S_TXD	V1P8A	0 (20k PD)	0 (20k PD)	
129	AM10	PCIE_CLKREQ0_N	1	PCIE_CLKREQ0_N	V1P8A	Input (20k PU)	Input (20k PU)	
130	AM12	PCIE_CLKREQ1_N	1	PCIE_CLKREQ1_N	V1P8A	Input (20k PU)	Input (20k PU)	
131	AK14	PCIE_CLKREQ2_N	1	PCIE_CLKREQ2_N	V1P8A	Input (20k PU)	Input (20k PU)	
132	AM14	PCIE_CLKREQ3_N	1	PCIE_CLKREQ3_N	V1P8A	Input (20k PU)	Input (20k PU)	Mode2/SDMMC3_WP/I
133	A9	MF_PLT_CLK0	1	PLT_CLK0	V1P8A	0 (20k PD)	Clock (20k PD)	
134	C9	MF_PLT_CLK1	1	PLT_CLK1	V1P8A	0 (20k PD)	Clock (20k PD)	
135	B8	MF_PLT_CLK2	1	PLT_CLK2	V1P8A	0 (20k PD)	Clock (20k PD)	
136	B7	MF_PLT_CLK3	1	PLT_CLK3	V1P8A	0 (20k PD)	Clock (20k PD)	
137	B5	MF_PLT_CLK4	1	PLT_CLK4	V1P8A	0 (20k PD)	Clock (20k PD)	
138	B4	MF_PLT_CLK5	1	PLT_CLK5	V1P8A	0 (20k PD)	Clock (20k PD)	
139	C13	PMU_AC_PRESENT	1	PMU_AC_PRESENT	V1P8A	Input (20k PD)	Input (20k PD)	
140	C14	PMU_BATLOW_N	1	PMU_BATLOW_N	V1P8A	Input (20k PU)	Input (20k PU)	
141	F14	PMU_PLTRST_N	1	PMU_PLTRST_N	V1P8A	0 (20k PU)	1	
142	M16	PMU_PWRBTN_N	1	PMU_PWRBTN_N	V1P8A	Input (20k PU)	Input (20k PU)	
143	AF2	PMU_RESETBUTTON_N	1	PMU_RESETBUTTON_N	V1P8A	Input (20k PU)	Input (20k PU)	
144	B12	PMU_SLP_LAN_N	1	PMU_SLP_LAN_N	V1P8A	0 (20k PU)	0	
145	A13	PMU_SLP_S0IX_N	1	PMU_SLP_S0IX_N	V1P8A	0 (20k PU)	1	
146	B14	PMU_SLP_S3_N	1	PMU_SLP_S3_N	V1P8A	0 (20k PU)	1	
147	C12	PMU_SLP_S4_N	1	PMU_SLP_S4_N	V1P8A	0 (20k PU)	1	
148	C15	PMU_SUSCLK	1	PMU_SUSCLK	V1P8A	0 (20k PD)	32KHz Clock	
149	N16	PMU_WAKE_N	1	PMU_WAKE_N	V1P8A	Input (20k PU)	Input (20k PU)	
150	P18	PMU_WAKE_LAN_N	1	PMU_WAKE_LAN_N	V1P8A	Input (20k PU)	Input (20k PU)	
151	D14	SUS_STAT_N	1	SUS_STAT_N	V1P8A	0 (20k PU)	1	
152	AE3	SUSPWRDNACK	1	SUSPWRDNACK	V1P8A	0 (20k PD)	0 (20k PD)	
153	H5	PWM0	1	PWM0	V1P8A	0 (20k PD)	0	
154	H7	PWM1	1	PWM1	V1P8A	0 (20k PD)	0	
155	AH2	SATA_GP0	1	SATA_GP0	V1P8A	Input (20k PD)	Input (20k PD)	
156	AG3	SATA_GP1	1	SATA_GP1	V1P8A	Input (20k PD)	Input (20k PD)	
157	AG1	SATA_GP2	1	SATA_DEVSLP0	V1P8A	0 (20k PD)	0 (20k PD)	
158	AF3	SATA_GP3	1	SATA_DEVSLP1	V1P8A	0 (20k PD)	0 (20k PD)	Mode2/ MMC1_RESET_B/O
159	AH3	SATA_LEDN	1	SATA_LEDN	V1P8A	1 (20k PD)	PRG	Mode2/UART0_RXD/I
160	K2	SDMMC3_1P8_EN	1	SDMMC3_1P8_EN	V1P8A	0 (20k PD)	0	
161	K3	SDMMC3_CD_N	1	SDMMC3_CD_N	V1P8A	Input (20k PU)	Input (20k PU)	

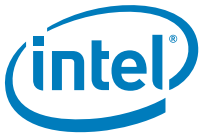
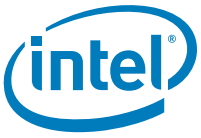
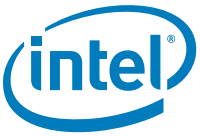


Table 2-27. GPIO Multiplexing and Modes (Sheet 6 of 6)

Count	SoC Pin No.	CFIO Name	Default Mode	Default Function	GPIO SoC Power Rail	Pwrgood Assert State	Resetout De-assert State	Optional Modes/ Direction
162	F2	SDMMC3_CLK	1	SDMMC3_CLK	V3P3A/ V1P8A	0 (20k PD)	0	
163	D2	SDMMC3_CMD	1	SDMMC3_CMD	V3P3A/ V1P8A	Z (20k PU)	Z (20k PU)	
164	J1	SDMMC3_D0	1	SDMMC3_D0	V3P3A/ V1P8A	Z (20k PU)	Z (20k PU)	
165	J3	SDMMC3_D1	1	SDMMC3_D1	V3P3A/ V1P8A	Z (20k PU)	Z (20k PU)	
166	H3	SDMMC3_D2	1	SDMMC3_D2	V3P3A/ V1P8A	Z (20k PU)	Z (20k PU)	
167	G2	SDMMC3_D3	1	SDMMC3_D3	V3P3A/ V1P8A	Z (20k PU)	Z (20k PU)	
168	L3	SDMMC3_PWR_EN_N	1	SDMMC3_PWR_EN_N	V1P8A	1 (20k PD)	1	
169	K10	SDMMC2_CLK	1	SDMMC2_CLK	V1P8A	0 (20k PD)	0	
170	K9	SDMMC2_CMD	1	SDMMC2_CMD	V1P8A	Z (20k PU)	Z (20k PU)	
171	M12	SDMMC2_D0	1	SDMMC2_D0	V1P8A	Z (20k PU)	Z (20k PU)	
172	M10	SDMMC2_D1	1	SDMMC2_D1	V1P8A	Z (20k PU)	Z (20k PU)	
173	K7	SDMMC2_D2	1	SDMMC2_D2	V1P8A	Z (20k PU)	Z (20k PU)	
174	K6	SDMMC2_D3_CD_N	1	SDMMC2_D3_CD_N	V1P8A	Z (20k PU)	Z (20k PU)	
175	AM9	MF_SMB_ALERT_N	1	SMB_ALERT_N	V1P8A	Input (20k PU)	Input (20k PU)	Mode3/UART0_TXD/O
176	AM6	MF_SMB_CLK	1	SMB_CLK	V1P8A	Input (20k PU)	Input (20k PU)	
177	AM7	MF_SMB_DATA	1	SMB_DATA	V1P8A	Input (20k PU)	Input (20k PU)	
178	H4	SPKR	1	SPKR	V1P8A	0 (20k PU)	PRG	
179	V14	SPI1_CLK	1	SPI1_CLK	V1P8A	0 (20k PU)	0	
180	Y13	SPI1_CS0_N	1	SPI1_CS0_N	V1P8A	1 (20k PU)	1	
181	Y12	SPI1_CS1_N	1	SPI1_CS1_N	V1P8A	1 (20k PU)	1	
182	V13	SPI1_MISO	1	SPI1_MISO	V1P8A	Input (20k PU)	Input (20k PD)	
183	V12	SPI1_MOSI	1	SPI1_MOSI	V1P8A	0 (20k PU)	0	
184	AD40	SVID0_ALERT_N	1	SVID0_ALERT_N	V1P8A	Input	Input	
185	AD42	SVID0_CLK	1	SVID0_CLK	V1P8A	0	1 or Z	
186	AD41	SVID0_DATA	0	SVID0_DATA	V1P8A	0	0	
187	AD50	PROCHOT_N	1	PROCHOT_N	V1P8A	Z	Z	
188	P14	USB_OC0_N	1	USB_OC0_N	V1P8A	Input (20k PU)	Input (20k PU)	
189	P16	USB_OC1_N	1	USB_OC1_N	V1P8A	Input (20k PU)	Input (20k PU)	

§ §





3 Processor Core

3.1 SoC Transaction Router

The SoC Transaction Router is a central hub that routes transactions between the CPU cores, graphics controller, I/O, and the memory controller.

In general, it handles:

CPU Core Interface: Requests for CPU Core-initiated memory and I/O read and write operations and processor-initiated message-signaled interrupt transactions:

- Device MMIO and PCI configuration routing.
- Buffering and memory arbitration.
- PCI Configuration and MMIO accesses to host device (0/0/0).

For more information on SoC Transaction Router Registers, refer to *the SoC Datasheet* Volume 2 and Volume 3 (See Related Documents section).

3.2 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-x) added hardware support in the processor to improve the virtualization performance and robustness.

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

3.2.1 Intel® VT-x Objectives

- **Robust**: VMMs no longer need to use paravirtualization or binary translation. This means that they will be able to run off-the-shelf operating systems 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-x provides hardware acceleration for virtualization of IA platforms. Virtual Machine Monitor (VMM) can use Intel® VT-x features to provide improved reliable virtualized platform.

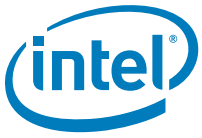
3.2.2 Intel® VT-x Features

- 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
  - A VM function is an operation provided by the processor that can be invoked using the VMFUNC instruction from guest VM without a VM exit
  - A 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

3.3 Security and Cryptography Technologies

3.3.1 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.



3.3.2 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 forth.

3.3.3 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, for example, fixed-priority, rotating, hash, or PAIR, during setup by means of 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.

3.4 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

3.5 References

For further details of Intel® 64 and IA-32 architectures refer to 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/us/en/processors/architectures-software-developer-manuals.html

For more details on Intel® Performance Primitives refer to:

- Intel® Performance Primitives web page—http://software.intel.com/en-us/intel-ipp/

For more details on using the RDRAND instruction, refer to Intel® Advanced Vector Extensions (Intel® AVX) 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 many SoC 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 in the following tables.

Table 4-1. 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_CLK	19.2 MHz or 25 MHz	Can be configured as an input to compensate for board routing delays through Soft Strap.

Table 4-2. SoC Clock Outputs (Sheet 1 of 2)

Clock Domain	Signal Name	Frequency	Usage/Description
DDR	DDR3_M0_CKP[1,0] DDR3_M0_CKN[1,0] DDR3_M1_CKP[1,0] DDR3_M1_CKN[1,0]	800 MHz	Drives the Memory ranks 0-1. Data rate (MT/s) is 2x the clock rate.
SDHC	MMC1_CLK SD2_CLK SD3_CLK	200 MHz	Clock for Storage Devices.
SPI	FST_SPI_CLK	20 MHz, 33 MHz, 50 MHz	Clock for SPI flash, 20 MHz by default.
COMMS	PMC_SUSCLK[0]	32.768 KHz	Pass through clock from RTC oscillator.
LPC	LPC_CLK[0:1]	19.2 MHz or 25 MHz	Provided to devices requiring LPC clock.
DisplayPort*	DDI0_TXP[3:0] DDI0_TXN[3:0] DDI1_TXP[3:0] DDI1_TXN[3:0] DDI2_TXP[3:0] DDI2_TXN[3:0]	162 or 270 MHz	Differential clock for DP devices.

Table 4-2. SoC Clock Outputs (Sheet 2 of 2)

Clock Domain	Signal Name	Frequency	Usage/Description
HDMI	DDI0_TXP[3:0] DDI0_TXN[3:0] DDI1_TXP[3:0] DDI1_TXN[3:0] DDI2_TXP[3:0] DDI2_TXN[3:0]	25–297 MHz	Differential clock for HDMI devices
HDMI DDC	DDI[2:0]_DDC_CLK	100 KHz	Clock for HDMI DDC 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	SVID0_CLK	20 MHz	Clock used by voltage regulator
Platform Clocks	PLT_CLK [5:0]	19.2 MHz	Platform clocks
I^2C	I2C[6:0]_CLK	1.7 MHz	I^2C clocks

§ §

5 Thermal Management

5.1 Overview

The thermal management system for the SoC 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 specification. Throttling mechanisms are used to reduce power consumption when thermal limits of the device are exceeded and the system is notified of critical conditions by means of interrupts or thermal signalling pins. SoC thermal management differs from legacy implementations primarily by replacing dedicated thermal management hardware with firmware.

The thermal management system:

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

5.2 Digital Thermal Sensors

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

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

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

Table 5-1. Temperature Reading Based on DTS (Sheet 1 of 2)

DTS Counter Value[8:0]	Temperature Reading (Tj_{MAX} = 90 °C)
127	90 °C
137	80 °C
147	70 °C
157	60 °C
167	50 °C
177	40 °C
187	30 °C
197	20 °C
207	10 °C
217	0 °C
227	-10 °C

Table 5-1. Temperature Reading Based on DTS (Sheet 2 of 2)

DTS Counter Value[8:0]	Temperature Reading (Tj_{MAX} = 90 °C)
237	-20 °C
247	-30 °C
255	-38 °C

Note: DTS encoding of 127 always represents Tj_{MAX} at 90 °C, the encoding 137 from DST indicates 80 °C and so forth.

Note: The DTS value 255 represent the minimum temperature and thus -38 °C is the lowest temperature will be reported by the SoC.

Thermal trip points are of two types:

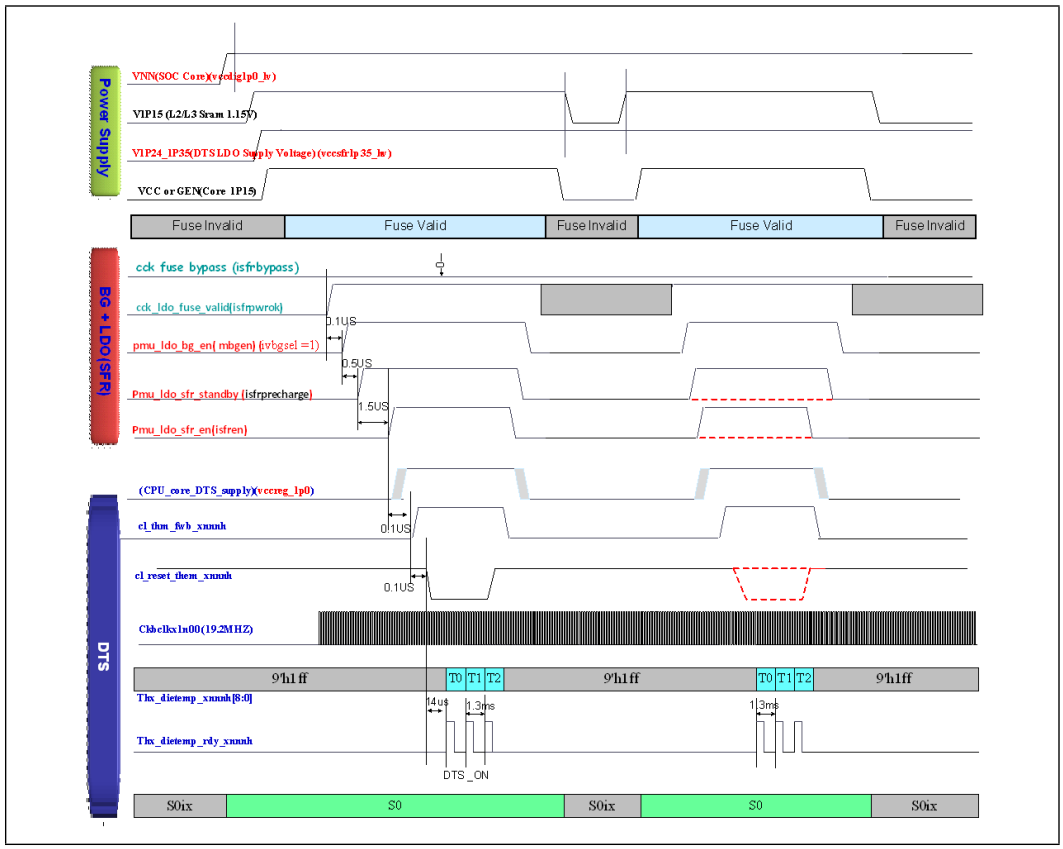
- **Hardware Trip**: The Catastrophic trip points generated by DTSs based on predefined temperature setting defined in fuses.
- **Programmable Trips**: Four programmable trip settings that can be set by firmware/software.

Note: DTS accuracy is ±8 °C under 60°C and ±5 °C above 60 °C.

5.2.1 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. Figure 5-1 shows the various control signals needed for DTS operations.

Figure 5-1. DTS Mode of Operation



5.3 Hardware Trips

5.3.1 Catastrophic Trip (THERMTRIP)

Catastrophic trip is generated by DTS whenever the ambient temperature around it reaches (or extends) beyond the maximum 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 PLLs and power rails to prevent SoC breakdown. To prevent glitches from triggering shutdown events, catastrophic trips from DTSs are registered before being sent out.

5.4 SoC Programmable Trips

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

5.4.1 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.

5.4.2 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 default to zero. To prevent spurious results, software/firmware should program the trip values prior to enabling the trip point.

5.5 Platform Trips

5.5.1 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.

5.5.2 EXTTS

The 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 the P-unit.

5.5.3 SVID

When the Voltage Regulator (VR) reaches its threshold (VR_Icc_Max, VR_Hot), status bits in SVID are set. SVID sends SVID_Status message to P-unit.

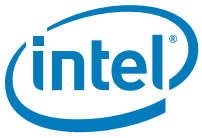
5.6 Dynamic Platform Thermal Framework (DPTF)

The 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.

5.7 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.

§ §



6 Power Management

6.1 Power Management Features

- ACPI System States support (S0, S3, S4, and S5)
- Processor Core/Package States support (C0–C7)
- SoC Graphics Adapter States support D0–D3.
- Support Processor and GFx Burst
- Dynamic I/O power reductions (disabling sense amps on input buffers, tri-stating output buffers)
- Active power-down of Display links

6.2 Power Management States Supported

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

6.2.1 System States

Table 6-1. General Power States for System (Sheet 1 of 2)

States/ Sub-states	Legacy Name/Description	CPU State	Graphics Adapter State
G0/S0/C0	**FULL ON:** Processor operating. Individual devices may be shut down to save power. The different processor operating levels are defined by Cx states.	Full on	D0
G0/S0/Cx	**Cx State:** Processor manages C-State itself.	C1/C1E: Auto Halt	D0
		C6: Deep Power Down	D3/Display Off
		C7: Deep Power Down	D3/Display Off
G1/S3	**Suspend-To-RAM (STR):** The system context is maintained in system DRAM, but power is shut to non-critical circuits. Memory is retained, and refreshes continue. All external clocks are shut off; RTC clock and internal ring oscillator clocks are still toggling.	Off	Display Off

Table 6-1. General Power States for System (Sheet 2 of 2)

States/ Sub-states	Legacy Name/Description	CPU State	Graphics Adapter State
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.	Off	Display Off
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.	Off	Display Off
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.	Off	Display Off

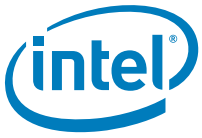


Table 6-3 shows the transitions rules among the various states.

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.

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

Table 6-2. Platform Voltage Rails and Power Modes

Power Type	Voltage Range (V)	S0	S3	S4/S5	G3
VCC (0 and 1)	0.5–1.3	On	Off	Off	Off
VGG	0.5–1.2	On	Off	Off	Off
VNN	0.5–1.05 (or 1.05 Fixed)	On	On	On	Off
V1P15S	1.15	On	Off	Off	Off
V1P05A	1.05	On	On	On	Off
V1P24A	1.24	On	On	On	Off
VDDQ	1.35	On	On	Off	Off
V1P8A	1.8	On	On	On	Off
V3P3A	3.3	On	On	On	Off
VSDIO	1.8/3.3	On	Off	Off	Off
V3P3_RTC	3.3	On	On	On	On
VCC_HDA	1.5/1.8	On	On	On	Off

Table 6-3. 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: Processor 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.

6.2.2 Integrated Memory Controller States

Table 6-4. Main Memory States

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

6.3 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.

6.3.1 Enhanced Intel® SpeedStep® Technology

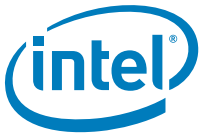
The following are the key features of Enhanced Intel® SpeedStep® Technology:

- 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 to Chapter 6, Thermal Management.

6.3.2 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.



6.3.3 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.

6.3.3.1 Clock Control and Low-Power States

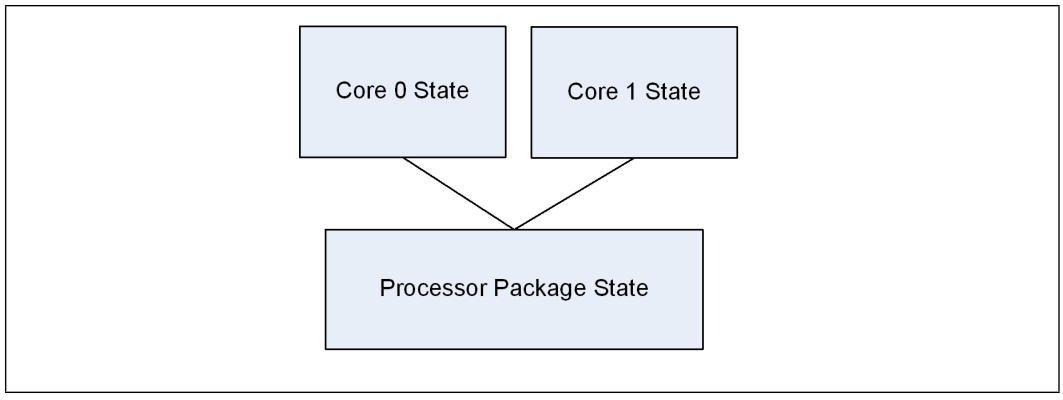
The processor core supports low power states at core level. The central power management logic ensures the entire processor core enters the new common processor core power state. For processor core power states higher than C1, this would be done by initiating a P_LVLx (P_LVL4 and P_LVL6) I/O read to 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 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
- Processor Pending Break Event (PBE_B)
- MSI

Figure 6-1. Idle Power Management Breakdown of the Processor Cores



6.3.4 Processor Core C-States Description

Table 6-5. 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 are based on the assumption that both threads are in the common low power state.

6.3.4.1 Core C0 State

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

6.3.4.2 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. See the *Intel® 64 and IA-32 Architecture Software Developer's Manual, Volume 3A/3B: System Programmer's Guide* for more information.

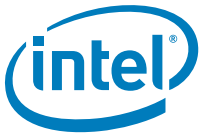
While a core is in C1/C1E state, it processes bus snoops and snoops from other threads. For more information on C1E, see "Section 6.3.5.2, "Package C1/C1E State" on page 66".

6.3.4.3 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.

6.3.4.4 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 S0 state if possible.



6.3.4.5 C-State Auto-Demotion

In general, deeper C-States, such as C6, 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.

6.3.5 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. Depending on the type of break event, the processor does the following:

- If a core break event is received, the target core is activated and the break event message is forwarded to the target core.
  - If the break event is not masked, the target core enters the core C0 state and the processor enters package C0.
  - If the break event is masked, the processor attempts to re-enter its previous package state.
- If the break event was due to a memory access or snoop request.
  - But the platform did not request to keep the processor in a higher package C-State, the package returns to its previous C-State.
  - And the platform requests a higher power C-State, the memory access or snoop request is serviced and the package remains in the higher power C-State.

Table 6-6. Coordination of Core/Module Power States at the Package Level

Package C-State		Core/Module 1				
		C0	C1	C6NS	C6FS	C7
Core/Module 0	C0	C0	C0[1]	C0	C0	C0
	C1	C0	C1[1]	C1[1]	C1[1]	C1[1]
	C6NS	C0	C1[1]	C6C	C6	C6
	C6FS	C0	C1[1]	C6C	C6	C6
	C7	C0	C1[1]	C6C	C6	C7

Notes:
1. If enabled, the package C-State will be C1E if all actives cores have resolved a core C1 state or higher.
2. C6NS implies only the core should be powergated, but the L2 cache contents should be retained.
3. C6FS implies the core should be powergated, and the L2 cache can be fully flushed to get even more power savings.
4. C6C is C6-Conditional where the L2 cache is still powered.
5. Two cores of the SoC will make up one module.

6.3.5.1 Package C0 State

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 State.

6.3.5.2 Package C1/C1E State

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:

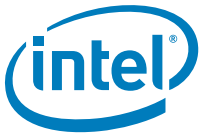
- All cores have directly requested C1E by means of MWAIT(C1) with a C1E sub-state hint.
- All cores are in a power state lower that C1/C1E but the package low power state is limited to C1/C1E by means of the PMG_CST_CONFIG_CONTROL MSR.
- All cores have requested C1 using HLT or MWAIT(C1) and C1E auto-promotion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

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

6.3.5.3 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.

6.3.5.4 Package C7 State

A processor enters the package C7 low power state when all cores are in the C7 State. In package C7 State, 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 State.

6.3.6 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.

6.3.7 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 5.0 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.

6.3.8 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.

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

When a Local Flat Panel (LFP) supports multiple refresh rates, the Intel® Seamless Display Refresh Rate Switching Technology (Intel® SDRRS Technology) 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 Technology—static and seamless. The static Intel® SDRRS Technology method uses a mode change to assign the new refresh rate. The seamless Intel® SDRRS Technology 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.

6.4 Memory Power Management

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

6.4.1 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 DIMM connector is unpopulated, or is 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.

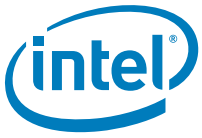
SCKE tri-state should be enabled by BIOS where appropriate, since at reset all rows must be assumed to be populated.

6.4.2 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.

6.4.2.1 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.

6.4.2.2 Conditional Self-Refresh

Intel® Rapid Memory Power Management (Intel® RMPM) conditionally places memory into self-refresh in the package low-power states. Intel® 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.

6.4.2.3 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 de-assertion with open pages) or precharge power-down (CKE de-assertion 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.

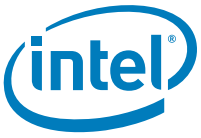
If dynamic power-down is enabled, all ranks are powered up before doing a refresh cycle and all ranks are powered down at the end of refresh.

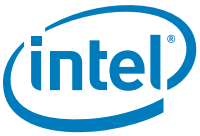
6.4.2.4 DRAM I/O Power Management

Unused signals should be disabled to save power and reduce electromagnetic interference. This includes all signals associated with an unused memory channel. Clocks can be controlled on a per SO-DIMM basis. Exceptions are made for per SO-DIMM control signals such as CS#, CKE, and ODT for unpopulated SO-DIMM slots.

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).

§ §





7 System Memory Controller

7.1 DDR3L Interface Signals

Table 7-1. Memory Channel 0 DDR3L 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[1,0]	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.
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 DDR3	**Resistor Compensation:** This signal needs to be terminated to VSS on board. This signal is driven from external clock source.
DDR3_M0_OCAVREF	I DDR3	**Reference Voltage:** DDR3 CA interface Reference Voltage.
DDR3_M0_ODQVREF	I DDR3	**Reference Voltage:** DDR3 DQ interface Reference Voltage.

Table 7-1. Memory Channel 0 DDR3L Signals (Sheet 2 of 2)

Signal Name	Direction Type	Description
Notes: 1. For Channel 1 signals, refer to this table 8-1, where Channel 1 signals have the same functions and descriptions as of correspondent signals of channel 0. The only exception would be the signal name. For Channel 1 signals they will be referred to as **DDR3_M1_**Function where in Channel 0 they were referred to as **DDR3_M0_**Function. 2. Vref signals are not connected on the SoC side: leave as NC. 3. There is no support for memory modules with different technologies or capacities on opposite sides of the same memory module. If one side of a memory module is populated, the other side is either identical or empty.		

Table 7-2. Other Memory DDR3L Signals

Signal Name	Direction Type	Description
DDR3_CORE_PWROK	I DDR3	**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_DRAM_PWROK	I DDR3	**VDD Power OK:** Asserted once the VRM is settled.
DDR3_M0_DRAMRST_N	O	**DRAM Reset:** This signal is used to reset DRAM devices.

7.2 System Memory Technology Supported

Table 7-3. Supported DDR3L DRAM Devices

DRAM Density	Data Width	Banks	Bank Address	Row Address	Column Address	Page Size
1Gb	x8	8	BA[2:0]	A[13:0]	A[9:0]	1KB
2Gb	x8	8	BA[2:0]	A[14:0]	A[9:0]	1KB
4Gb	x8	8	BA[2:0]	A[15:0]	A[9:0]	1KB
8Gb	x8	8	BA[2:0]	A[15:0]	A[11], A[9:0]	2KB
1Gb	x16	8	BA[2:0]	A[12:0]	A[9:0]	2KB
2Gb	x16	8	BA[2:0]	A[13:0]	A[9:0]	2KB
4Gb	x16	8	BA[2:0]	A[14:0]	A[9:0]	2KB
8Gb	x16	8	BA[2:0]	A[15:0]	A[9:0]	2KB

Table 7-4. Supported DDR3L Memory Size Per Rank

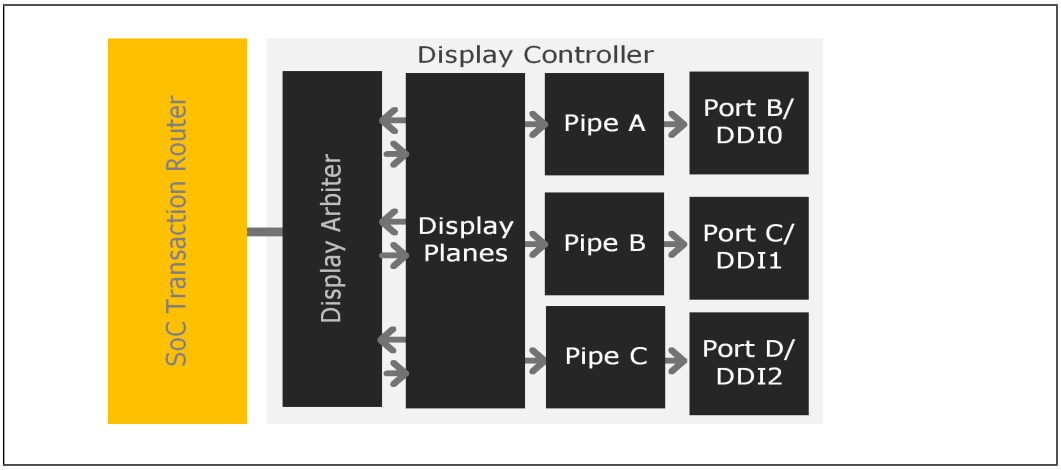
Memory Size/Rank	DRAM Chips/Rank	DRAM Chip Density	DRAM Chip Data Width	Page Size @ 64-bit Data Bus
1GB	8	1Gb	x8	8KB = 1KB * 8 chips
2GB	8	2Gb	x8	8KB = 1KB * 8chips
4GB	8	4Gb	x8	8KB = 1KB * 8 chips
8GB	8	8Gb	x8	16KB = 2KB * 8 chips
512MB	4	1Gb	x16	8KB = 2KB * 4 chips
1GB	4	2Gb	x16	8KB = 2KB * 4 chips
2GB	4	4Gb	x16	8KB = 2KB * 4 chips
4GB	4	8Gb	x16	8KB = 2KB * 4 chips

§ §

8 Graphics, Video, and Display

8.1 SoC Graphics Display

Figure 8-1. SoC Graphics Display Diagram



The Processor Graphics controller display pipe can be divided into four components that are all incorporated into the Display Controller:

- Display Planes
- Display Pipes
- Display Physical Interfaces
- Four planes available per pipe - 1x Primary, 2x Video Sprite and 1x Cursor

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.

8.1.1 Primary Display 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.

8.1.1.1 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

8.1.1.2 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.

8.1.2 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 be attached to any of the display interfaces.

8.1.3 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 (DisplayPort*, embedded DisplayPort*, DVI and HDMI*) interfaces.

8.2 Digital Display Interfaces

Table 8-1. Display Technologies Support

Technology	Standard
eDP* 1.4	VESA embedded DisplayPort* Standard, Version 1.4
DP* 1.1a	VESA DisplayPort* Standard, Version 1.1a
HDMI* 1.4b	High-Definition Multi-media Interface Specification, Version 1.4b.
HDCP* 1.4 wired/ 2.2 wireless	High-bandwidth Digital Content Protection System (HDCP), Revision 1.4

Notes:
1. All SoC display interfaces are designed per specifications provided in industry standard listed above. For specifications of each technology, refer to the correspondent standard and follow guidance provided.
2. The SoC supports High Definition Content Protection Technology (HDCP) on all supported wired displays (HDMI, DP* and eDP*).

Table 8-2. SoC Display Configuration (Sheet 1 of 2)

Feature	eDP*	DP	HDMI/DVI
Number of Ports	2 (DDI[0:1]) (2x4 @2.7 Gb/s)	3 (DDI[0:2]) (2x4 @2.7 Gb/s)	3 (DDI[0:2]) (2x4 @2.97 Gb/s)
Maximum Resolution	2560x1440 @ 60Hz 24bpp	3840x2160 @ 30Hz 2560x1600 @ 60Hz 24bpp	3840x2160 @ 30Hz 2560x1600 @ 60Hz 24bpp
Minimum Resolution	none	none	480i/576i
Data Rate	10.8Gb/s	10.8Gb/s	6.6 Gb/s
Standard	eDP1.4	DP1.1a	HDMI1.4b
Power gated during display off	Yes	Yes	Yes
DRRS (Refresh reduction)	Yes (Panel command)	N/A	N/A
Self-Refresh with Frame buffer in Panel	No	No	No

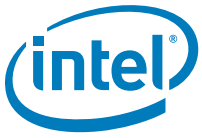


Table 8-2. SoC Display Configuration (Sheet 2 of 2)

Feature	eDP*	DP	HDMI/DVI
Content-Based backlight control	DPST6/CABC	N/A	N/A
HDCP wired display	N/A(ASSR support)	1.4	1.4
PAVP	AES-encrypted buffer, plane control, panic attack		
SEC	All display registers can be accessed by CEC		
HD-Audio	N/A	Yes	Yes
LPE Audio	N/A	No	No
Compressed Audio	N/A	Yes	Yes

Table 8-3. SoC Display supported Resolutions

	1 Display only 2 Displays		2 Displays			3 Displays
	1 Internal	1 External	1 Internal + 1 External	1 Internal + 1 WIDI	2 Externals	1 Internal + 2 Externals
Internal #1	eDP* 2560x1440 @ 60Hz	N/A	eDP* 2560x1440 @ 60Hz	eDP* 2560x1440 @ 60Hz	N/A	eDP* 2560x1440 @ 60Hz
External #1	N/A	HDMI/DP* 3840x2160 @ 30Hz 2560x1600 @ 60Hz	HDMI/DP* 3840x2160 @ 30Hz 2560x1600 @ 60Hz	WIDI 1920x1080 @ 30Hz	HDMI/DP* 3840x2160 @ 30Hz 2560x1600 @ 60Hz	HDMI/DP* 3840x2160 @ 30Hz 2560x1600 @ 60Hz
External #2	N/A	N/A	N/A	N/A	HDMI/DP* 3840x2160 @ 30Hz 2560x1600 @ 60Hz	HDMI/DP* 3840x2160 @ 30Hz 2560x1600 @ 60Hz

Notes:
1. SoC supported maximum of 3 simultaneous displays. External display in both clone and extended modes.
2. WiDi resolution dependent on antenna configuration. 1080p assumes 2x2 and expect 720p for 1x1.
3. Experience may differ based on configuration, resolution, and work loads.

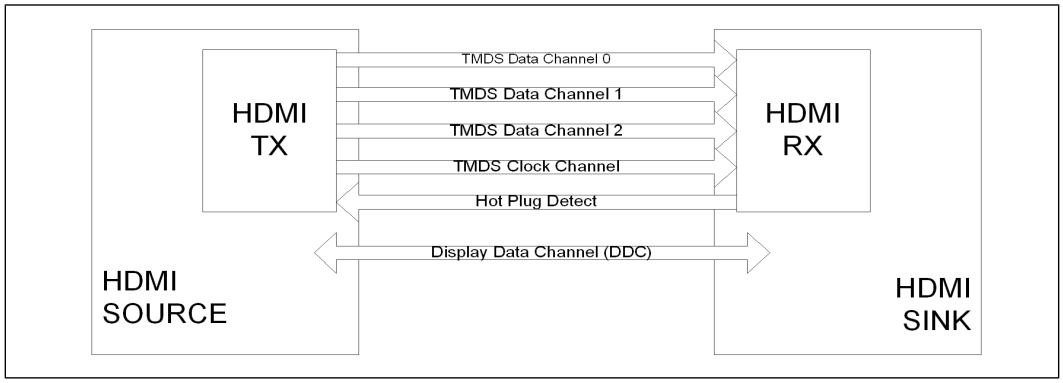
8.2.1 High Definition Multi-media Interface (HDMI)

The High-Definition Multi-media 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 8-2, 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.

Figure 8-2. HDMI Overview

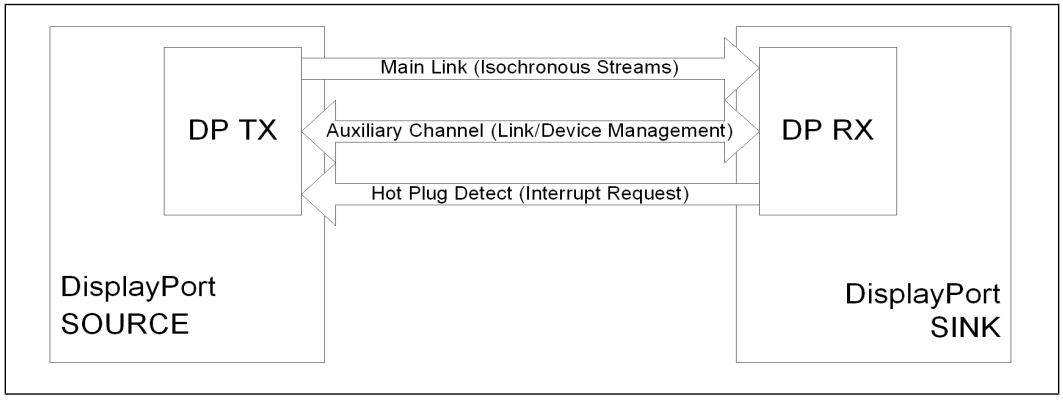


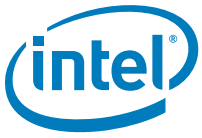
8.2.1.1 DisplayPort*

DisplayPort* 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. DisplayPort* is also suitable for display connections between consumer electronics devices such as high definition optical disc players, set top boxes, and TV displays.

A DisplayPort* 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 bi-directional channel used for link management and device control. The Hot-Plug Detect (HPD) signal serves as an interrupt request for the sink device.

Figure 8-3. DisplayPort* Overview





8.2.1.2 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.

8.2.1.3 DisplayPort* Auxiliary Channel

A bi-directional 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.

8.2.1.4 Hot-Plug Detect (HPD)

The SoC supports HPD for hot-plug sink events on the HDMI and DisplayPort* interfaces.

8.2.1.5 Integrated Audio Over HDMI and DisplayPort*

The SoC can support each audio streams on DP/HDMI ports. Each stream can be programmed to each DDI port.

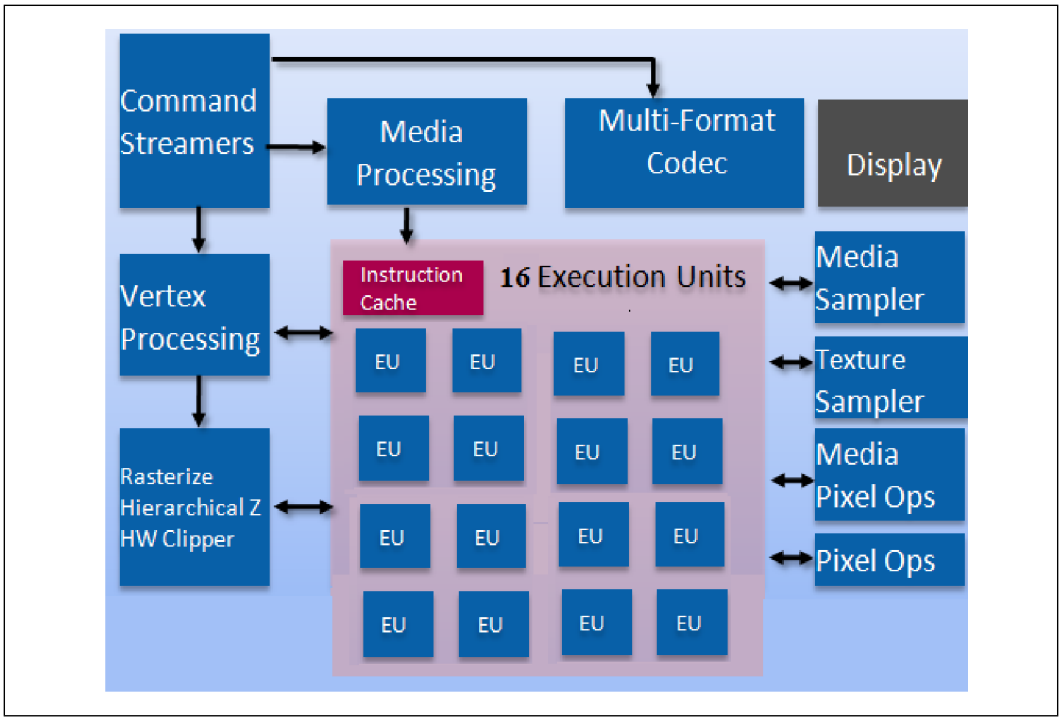
8.2.1.6 High-Bandwidth Digital Content Protection (HDCP)

HDCP is the technology for protecting high definition content against unauthorized copy or unreceptive between a source (computer, digital set top boxes, and so forth.) 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*).

8.3 3-D 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 3-D rendering, media compositing and video encoding. The Graphics engine is built around sixteen execution units (EUs).

Figure 8-4. 3-D Graphics Block Diagram



8.3.1 Features

The 3-D 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 3-D Engine. The Gen 8.0 LP 3-D engine provides the following performance and power-management enhancements:

- Hierarchal-Z
- Video quality enhancements

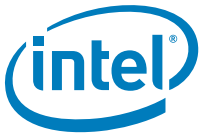
8.3.2 3-D Engine Execution Units

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

8.3.3 3-D Pipeline

8.3.3.1 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*.



8.3.3.2 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.

8.3.3.3 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.

8.3.3.4 Clip Stage

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

8.3.3.5 Strips and Fans (SF) Stage

The SF stage performs setup operations required to rasterize 3-D 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.

8.3.3.6 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.

8.4 VED (Video Encode/Decode)

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

This engine supports Full Hardware acceleration for decode of AVC/H.264, VC-1 and MPEG2 contents along with encode of MPEG2 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.

8.4.1 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 and below Media formats is supported as follow.

Table 8-4. Hardware Accelerated Video Decode/Encode Codec Support

Codec Format	Win8.1		Win7		Open Source Linux*	
	Decode Level	Encode Level	Decode Level	Encode Level	Decode Level	Encode Level
HEVC (H.265)	Supported (Hybrid solution)	Not Supported	Supported (Hybrid solution)	Not Supported	Not Supported	Not Supported
H.264	Supported	Supported	Supported	Supported	Supported	Supported
MPEG2	Supported	Not Supported (SW only)	Supported	Not Supported (SW only)	Supported	Supported
MVC	Supported (As two separate streams via ACV)	Supported (As two separate streams via ACV)	Supported (As two separate streams via ACV)	Supported (As two separate streams via ACV)	Supported	Supported, No KPI available
VC-1	Supported	Not Supported	Supported	Not Supported	Supported, No KPI available	Not Supported
WMV9	Supported	Not Supported	Supported	Not Supported	Supported, No KPI available	Not Supported
JPEG/MJPEG	Supported	Supported	Supported	Supported	Supported	Supported
VP8	Supported	Not Supported	Supported	Not Supported	Supported	Supported
VP9	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported
SVC	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported
MPEG4P2	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported
H.263	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported
Sorenson	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported
Xvid, DivX	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported
AVS	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported

Note: VP9 media codec GPU Accelerator to be supported post TTM, for non-Windows operating systems only.

Table 8-5. Resolution Details on Supported HW Accelerated Video Decode/Encode Codec (Sheet 1 of 2)

Codec Format	Decode Features	Encode Features
HEVC (H.265)	Profiles: MP. L5 up to 1080p120, 4kx2kp30	Not Supported
H.264	Profiles: CBP, MP, HP. L5.2 up to 1080p240, 4kx2kp60	Profiles: CBP, MP, HP. L5.1 up to 1080p120, 4kx2kp30
MPEG2	Profiles: HD, MP, HL 1080p60	Profiles: HD MP HL 1080p30
MVC	Profiles: CBP, MP HP. L5.2 up to 4kx3kp60 (4kx2kp30 per eye)	Profiles: CBP, MP HP L5.1 up to 4kx2kp30 (1080p60 per eye)

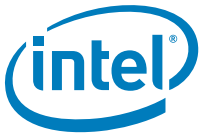
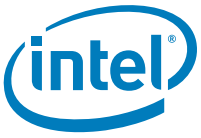


Table 8-5. Resolution Details on Supported HW Accelerated Video Decode/Encode Codec (Sheet 2 of 2)

Codec Format	Decode Features	Encode Features
VC-1	Profiles: AP. L4 1080p60	Not Supported
WMV9	Profiles: MP HL 1080p30	Not Supported
JPEG/MJPEG	850 Mpps (420), 640 Mpps (422), 420 Mpps (44)	850 Mpps (420), 640 Mpps (422), 420 Mpps (44)
VP8	Up to 4kx2kp60	Up to 4kx2kp30
VP9	Up to 1080p30	Up to 720p30

Notes:
1. VP9 media codec GPU Accelerator to be supported post TTM, for non-Windows operating systems only.
2. Resolution details for media codec on open source Linux OS depends on platform features and drivers used. Decode/ Encode features may not align to Table 9-4 that is specific to Win8.1 and Win7 operating systems.

§ §



9 MIPI*-CSI (Camera Serial Interface) and ISP

9.1 Signal Descriptions

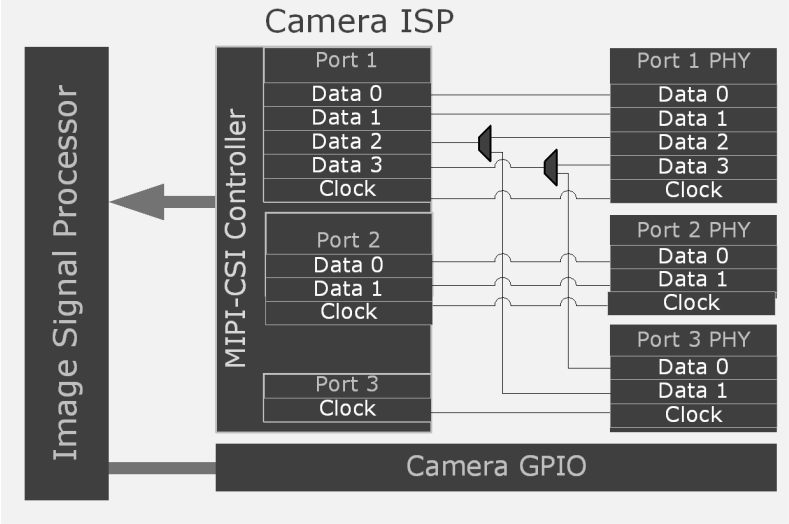
Table 9-1. CSI Signals

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:** 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 9-2. GPIO Signals

Signal Name	Direction/ Type	Description
GP_CAMERASB00	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.
GP_CAMERASB01	I/O	Output from shutter switch when its pressed full way. This switch state is used to trigger Xenon flash or LED Flash.
GP_CAMERASB02	I/O	Active high control signal to Xenon Flash to start charging the Capacitor.
GP_CAMERASB03	I/O	Active low output from Xenon Flash to indicate that the capacitor is fully charged and is ready to be triggered.
GP_CAMERASB04	I/O	Active high Xenon Flash trigger/Enables Torch Mode on LED Flash IC.
GP_CAMERASB05	I/O	Enables Red Eye Reduction LED for Xenon/Triggers STROBE on LED Flash IC.
GP_CAMERASB06	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.
GP_CAMERASB07	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.
GP_CAMERASB08	I/O	Active high signal to video camera to power-down the device.
GP_CAMERASB09	I/O	Active low output signal to reset digital still camera #0.
GP_CAMERASB10	I/O	Active low output signal to reset digital still camera #1.
GP_CAMERASB11	I/O	Active low output signal to reset digital video camera.

Figure 9-1. Camera Connectivity



9.1.1 Imaging Capabilities

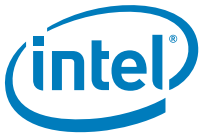
The following table summarizes imaging capabilities.

Table 9-3. 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
2-D Image capture	5MP, frame rate (30 fps)
2-D 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

9.1.2 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.



9.1.3 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 5 megapixel sensors are supported.

9.1.4 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.

9.1.5 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.

9.1.6 Secondary Camera Still Image Resolution

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

9.1.7 Primary Camera Video Resolution

Maximum primary camera video resolution is 1080p30.

Maximum primary camera dual video resolution is 1080p30.

9.1.8 Secondary Camera Video Resolution

Maximum secondary camera video resolution is 1080p30.

9.1.9 Bit Depth

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

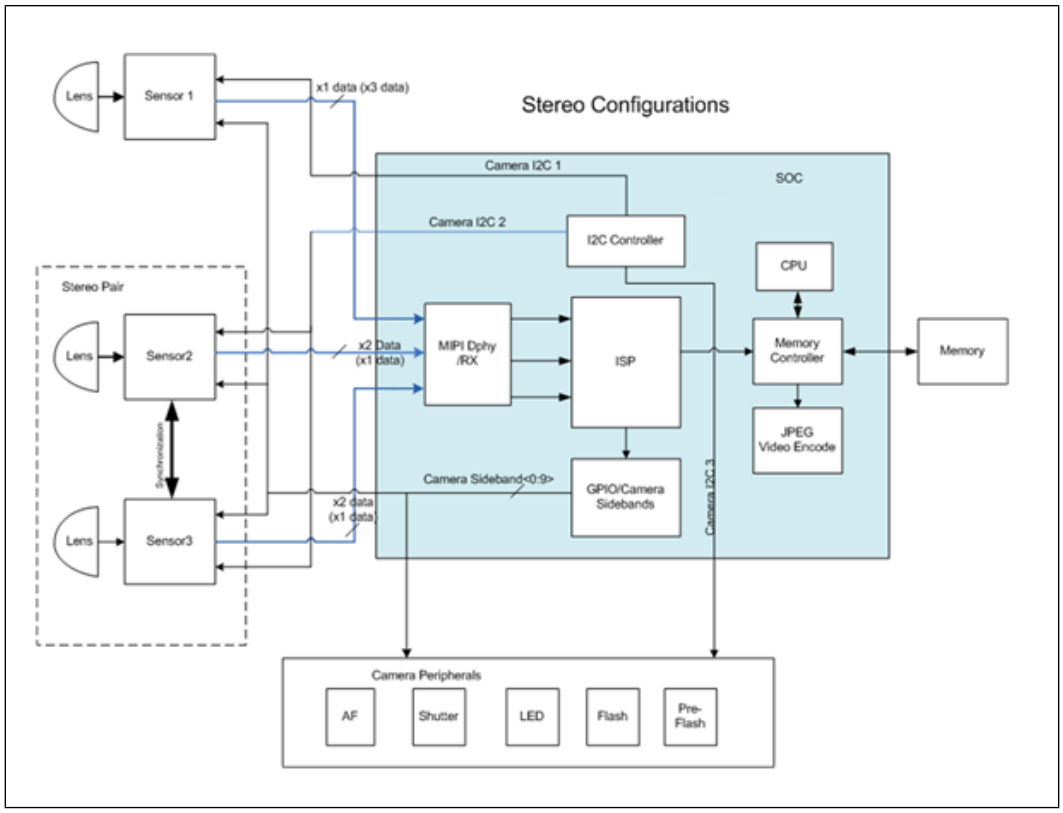
Capable of processing 18-bit images at half the performance levels; that is, process on-the-fly 16 megapixel 18-bit images at 7 fps instead of 15 fps.

Capable of processing up to 18-bit precision.

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

9.2 Imaging Subsystem Integration

Figure 9-2. Image Processing Components



9.2.1 Processor Core

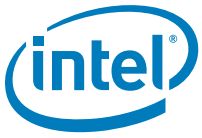
The processor core augments the signal processing capabilities of the hardware to perform post-processing on images such as auto focus, auto white balance, and auto exposure. The processor also runs the drivers that control the GPIOs and I^2C for sensor control.

9.2.2 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.

9.2.2.1 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.

9.2.2.2 I^2C for Camera Interface

The platform supports three (3) I^2C 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.

9.2.2.3 Camera Sideband for Camera Interface

Twelve (12) GPIO signals (GP_CAMERASB[11:00]) are allocated for camera functions; refer to Table 9-2 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 of camera GPIOs listed below is just an example of how the GPIOs can be used. Several of these functions could be implemented using I^2C, 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.

9.3 Functional Description

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

- Preview
- Image capture
- Video capture

9.3.1 Preview Mode

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.

9.3.2 Image Capture

During the image capture mode, the camera subsystem can acquire at a peak throughput of 5megapixels. 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 processor to provide flexibility.

9.3.3 Video Capture

During video recording, the ISP can capture video up to 1080p @ 30 fps and output preview frames concurrently. The ISP outputs video frames to memory in YUV420 or YUV422 format.

9.3.4 ISP Overview

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.

9.4 MIPI*-CSI-2 Receiver

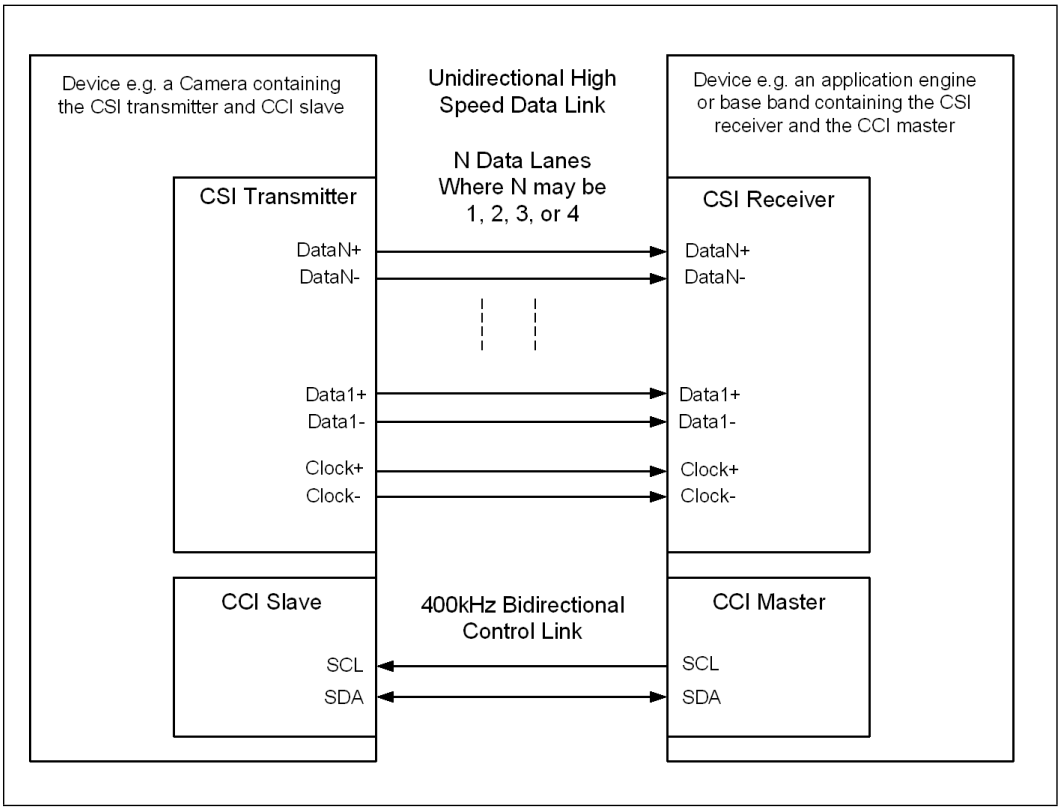
MIPI*-CSI-2 devices are camera serial interface devices. They are categorized into two types – a CSI transmitter device with Camera Control Interface (CCI) slave and CSI receiver device with CCI master.

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 9-3.

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 9-3. MIPI*-CSI Bus Block Diagram



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 that 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.

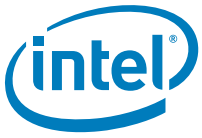
MIPI*-CSI-2 Functional Layers:

- **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 are ECC generator and corrector units to recover the data free from errors in the packet headers. 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.

9.4.1 MIPI*-CSI-2 Receiver Features

CSI Features:

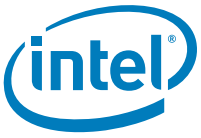
- Compliant to CSI-2 MIPI* specification for Camera Serial Interface (Version 1.00)
- Supports standard D-PHY receiver compliant to the MIPI* Specification
- Supports PHY data programmability up to four lanes.
- Supports PHY data time-out programming.
- Has controls to start and re-start the CSI-2 data 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 to Table 9-3 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.

§ §



10 SoC Storage

10.1 SoC Storage Overview

10.1.1 Storage Control Cluster (e-MMC*, SDIO, SD)

The SCC consists of SDIO, SD and e-MMC* controllers to support mass storage and I/O devices.

- Supports e-MMC v4.5.1
- One SD 3.0 interface
- One SDIO 3.0 interface

10.2 Signal Descriptions

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

Table 10-1. e-MMC* Signals

Signal Name	Direction/ Type	Description
SDMMC1_CLK	I/O/GPIO	**e-MMC* Clock** The frequency may vary between 25 and 200 MHz.
SDMMC1_D[2:0] **SDMMC1_D3_CD_N** **SDMMC1_D4_SD_WE** **SDMMC1_D[7:5]**	I/O/GPIO	**e-MMC* Port Data bits 0 to 7** Bidirectional port used to transfer data to and from e-MMC* 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 Multi-media Card controller. The Multi-media 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]. Some data signals have optional functionality: SDMMC1_D1: Data Line Bit 1 or Interrupt SDMMC1_D2: Data Line Bit2 or Read Wait SDMMC1_D3_CD#: Data Line Bit 3/SD card Detect
SDMMC1_CMD	I/O/GPIO	**e-MMC* 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.
SDMMC1_RCOMP	I/O/GPIO	**e-MMC* RCOMP** This signal is used for pre-driver slew rate compensation.

Table 10-2. SDIO Signals

Signal Name	Direction/ Type	Description
SDMMC2_CLK	I/O/GPIO	**SDIO Clock** The frequency may vary between 25 and 200 MHz.
SDMMC2_D[2:0]	I/O/GPIO	**SD Card Data bits [2:0]** 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 [2:0]. ***Note:*** Unused data lines will be tri-stated by the SoC logic.
SDMMC2_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.
SDMMC2_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 10-3. SD Signals

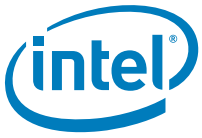
Signal Name	Direction/ Type	Description
SDMMC3_CLK	I/O/GPIO	**SD Card Clock** The frequency may vary between 25 and 200 MHz.
SDMMC3_D[3:0]	I/O/GPIO	**SD Card Data bits 0 to 3** Bidirectional port used to transfer data to and from SD/MMC card. By default, after power up or reset, only D[0] is used for data transfer. A wider data bus can be configured for data transfer, using D[3:0]. ***Note:*** Unused data lines will be tri-stated by the SoC logic.
SDMMC3_CD#	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. ***Note:*** The processor does not support plug-in eMMC device. Only soldered down eMMC device is supported.
SDMMC3_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.
SDMMC3_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.
SDMMC3_RCOMP	I/O/GPIO	**SD Card RCOMP** This signal is used for pre-driver slew rate compensation.
SDMMC3_PWR_EN#	I/O/GPIO	**SD Card Power Enable** This signal is used to enable power on a SD device.

10.3 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 Multi-Media Card (e-MMC*) Product Standard v4.5, JESD84-A5

§ §



11 USB Controller Interfaces

11.1 SoC Supports

- Four (4) Super Speed (SS) ports
- Five (5) High Speed (HS) ports
- Two (2) High Speed Inter-Chip (HSIC) ports

Note: There is one dedicated HS port (USB 2.0), while the other 4 HS ports are multiplexed with SS ports and can be used either by USB 2.0 or USB 3.0.

Note: It is recommended to disable USB OTG through BIOS. The Disable_OTG Soft Strap is not functional.

Note: Global Valid Bit is used to enable/disable debug features. It is recommended that users set the Global Valid Bit =1 on production ready systems for lower SoC power.

11.2 Signal Descriptions

Table 11-1. USB Signals

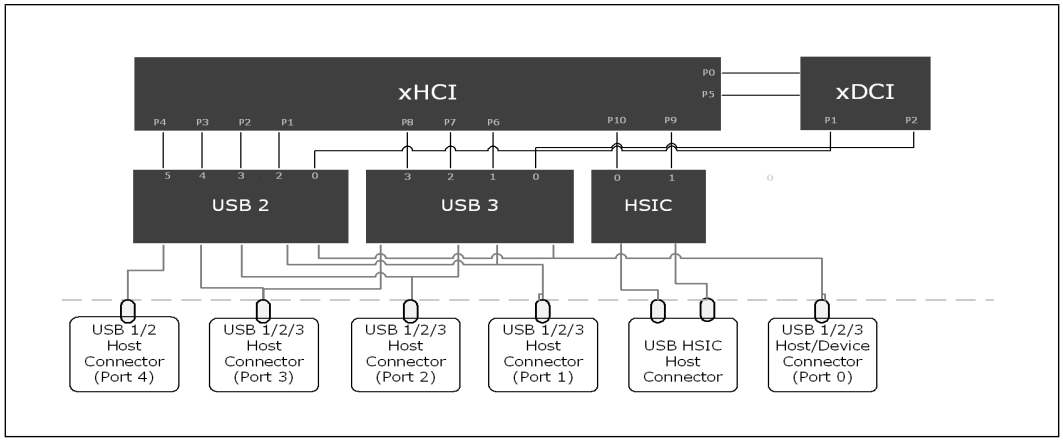
Signal Name	Direction/ Type	Description
USB3_TXP/N[0:3]	O USB 3.0 PHY	**Transmitter serial data outputs:** High-Speed Serialized data outputs.
USB3_RXP/N[0:3]	I USB 3.0 PHY	**Receiver serial data inputs:** High-speed serialized data inputs.
USB3_RCOMP_P/N	I USB 3.0 PHY	**Resistor Compensation:** An external resistor must be connected.
USB_DP/N[0:4]	I/O USB 2.0 PHY	**USB2 Data:** High speed serialized data I/O.
USB_RCOMP	O USB 2.0 PHY	**Resistor Compensation:** An external resistor must be connected.
USB_PLL_MON	O USB 2.0 PHY	USB High Speed Observation
USB_OTG_ID	I USB 2.0 PHY	**On-The-Go ID:** The signal is to identify if a Host or Device is connected to its port. ***Note:*** This is applicable only for Chrome OS based systems
USB_OC[1:0]_N	I USB 2.0 PHY	**Over Current detection:** This pin is used to indicate an over-current condition to the controller.

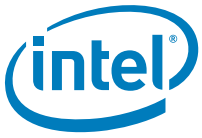
Table 11-2. HSIC Signals

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 HSIC Buffer	**Resistor Compensation:** RCOMP for HSIC buffer.

Note: See Chapter 2, "Physical Interfaces" for additional details.

Figure 11-1. xHCI Port Mapping





11.3 USB 3.0 xHCI (Extensible Host Controller Interface)

The xHCI compliant host controller can control up to 4 (four) USB 3.0 and 1 (one) USB 2.0 host. It supports devices conforming to USB 1.x to 3.0 at bit rates up to 5Gbps. All USB 3.0 ports support xHCI debug port functionality.

11.3.1 Features of USB 3.0 Host

1. SuperSpeed data interface is a four wire differential (TX and RX pairs).
2. Interface supports a bit rate of 5Gbps with a maximum theoretical data throughput over 3.2Gbps.

11.3.1.1 USB 3.0 Features

- Supported by xHCI software host controller interface
- USB 3.0 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
- Support for USB 3.0 Debug Device
- Supports IVCAM (USB PC Camera)

11.3.2 Features of USB HSIC

1. Two (2)-signal (strobe and data) source synchronous serial interface for on board inter-chip USB communication.
2. Uses 240 MHz DDR signaling to provide High-Speed 480Mb/s USB transfers.
3. Full Speed (FS) and Low Speed (LS) USB transfers are not directly supported by the HSIC interface.

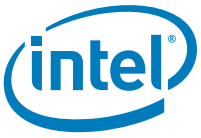
Major feature and performance highlights are as follows:

- Supported by xHCI software host controller interface
- High-Speed 480Mb/s data rate only
- Source-synchronous serial interface

11.4 References

- USB 3.0 Specification
- USB 2.0 Specification (Includes High-Speed Inter-Chip USB Electrical Specification)
- Extensible Host Controller Interface (xHCI) Specification v1.1

§ §



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

The 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.

Note: LPE is supported for non-Windows based platforms. It is not supported for Windows based platforms.

12.1 Signal Descriptions

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

Table 12-1. LPE Signals

Signal Name	Direction/Type	Description
GP_SSP_[2:0]_I2S_CLK	I/O	Clock signal for I^2S
GP_SSP_[2:0]_I2S_FRM	I/O	Frame select signal for I^2S
GP_SSP_[2:0]_I2S_DATAIN	I/O	RX data for I^2S
GP_SSP_[2:0]_I2S_DATAOUT	I/O	TX data for I^2S
Note: All LPE signals are multiplexed and may be used by other functions.		

12.2 Features

The LPE Audio Subsystem consists of the following:

- Integrated, power-efficient 32-bit architecture core with 24-bit audio processing instructions
- LPE Core processing speeds up to 343 MHz
- Closely Coupled Memories (CCMs)
  - 80KB Instruction RAM
  - 160KB Data RAM
  - 48KB Instruction Cache
  - 96KB 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 4KB mailbox memory

- Flexible audio interfaces include three SSPs with I^2S port functionality for bi-directional audio transfers
  - I^2S mode supports PCM payloads
  - Frame counters for all I^2S 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.
- Communicates to SRAM and external RAM through OCP fabric.
- Communicates with Audio peripherals using audio sub-fabric and Inter-Processor Communication (IPC) mechanism to communicate with the SoC Processor Core.

Note: Since LPE firmware must reside at a stolen memory location on 512MB boundaries below 3GB, it requires more than 512MB system memory. The LPE firmware itself is ~1MB, and is reserved by BIOS for LPE use.

12.2.1 Audio Capabilities

12.2.1.1 Audio Decode

The 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+

12.2.1.2 Audio Encode

The Audio core supports encoding of the following formats:

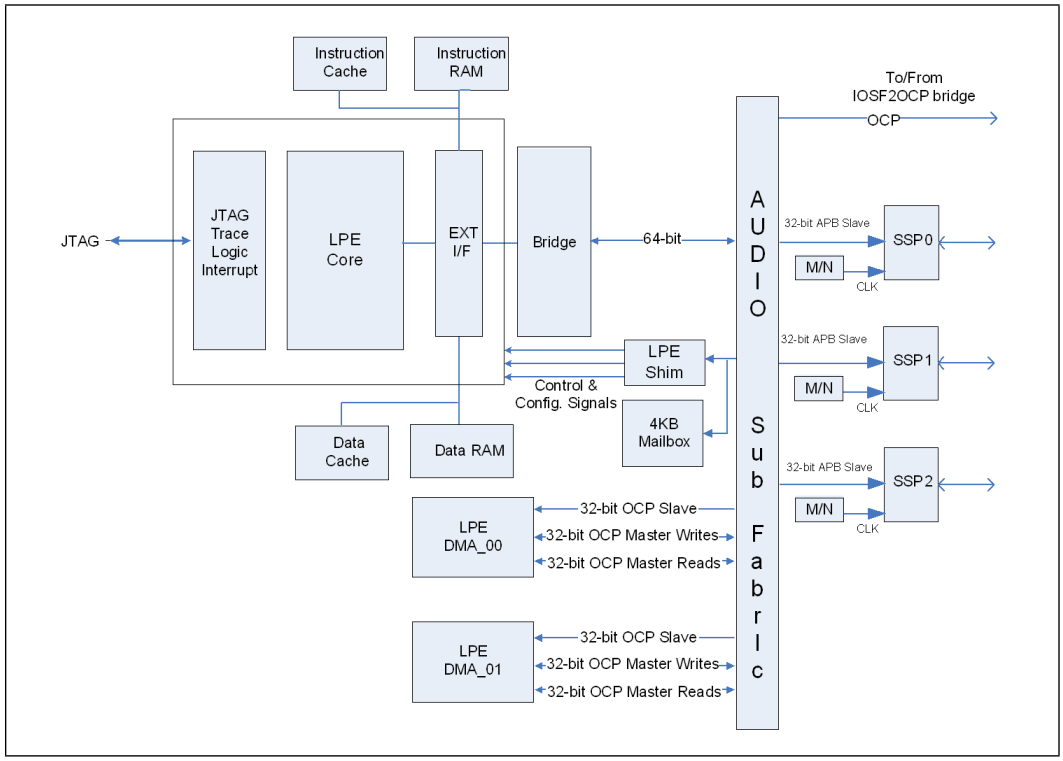
- MP3
- AAC-LC
- WMA
- DD-2channel

12.3 Detailed Block Level Description

12.3.1 LPE Core

The LPE core in the SoC runs at maximum frequency of 343 MHz and interfaces with the rest of the SoC system through the OCP bus. It is one of the masters on the Audio Sub-Fabric The IA-32 Processor and LPE DMA engines are the other masters on the fabric. The following figure shows the LPE core and its interfaces.

Figure 12-1. Audio Cluster Block Diagram

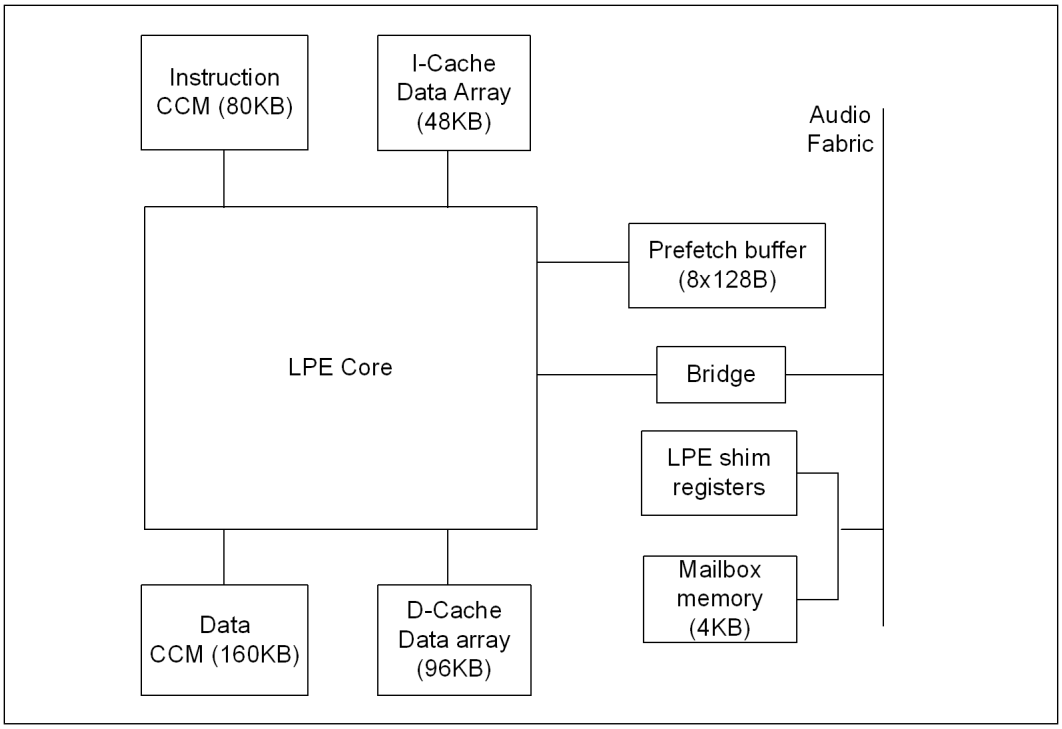


The main DSP hardware is a two-multiplier, multiply/accumulate unit, a register file LPE_PR to hold pairs of 24-bit data items, a register file LPE-OR to hold 56-bit accumulator values, an arithmetic/logic unit to operate on the LPE_PR and LPE_OR values, and a shift unit to operate on the LPE_PR and LPE_OR values. The multiply/accumulate unit also supports multiplication of 32-bit values from LPE_OR registers by 16-bit values from LPE_PR registers, with the 48-bit result written or accumulated in the LPE_OR register. The instructions for the DSP subsystems are built from operations that are divided into two sets: the slot 0 set and the slot 1 set. In each execution cycle, zero or one operations from each set can be executed independently according to the static bundling expressed in the machine code.

12.3.2 Memory Architecture

The LPE core is configured to use local memory and local caches. It has 80KB of Instruction Closely Coupled Memory (CCM), 160KB of Data CCM, 48KB of Instruction Cache and 96KB of Data Cache. The LPE core also has access to 4KB of mailbox memory and external DRAM.

Figure 12-2. Memory Connections for LPE



12.3.3 Instruction Closely Coupled Memory (CCM)

Instruction CCM for the core is used for loading commonly used routines as well as time-critical processing. Examples of time critical processing are acoustic echo cancellation and noise cancellation during voice calls.

Instruction CCM is initialized after reset by an external DMA controller. Runtime update of instruction CCM can be done either using explicit instructions or using an external DMA controller with inbound access.

12.3.4 Data Closely Coupled Memory (CCM)

Data CCM can be initialized after reset by an external DMA controller using inbound access. Runtime update of data CCM can be done either using stores to Data CCM or using an external DMA controller with inbound access.

12.3.5 Mailbox Memory and Data Exchange

The mailbox memory is a shared memory region in LPE address space that is accessible by the SoC Processor Core, PMC, and LPE. It is used when Doorbell registers cannot hold all the information that one processor wishes to communicate to the other. A typical example of such data blocks are audio stream related parameters when starting a new stream. The structures of data communicated through the mailbox are not defined in hardware so that software may partition the mailbox memory in any desired way and create any meaningful structures required.

12.4 Software Implementation Considerations

12.4.1 SoC Processor Core Cache Coherence

Traffic generated by the LPE core is considered non-cacheable and non-coherent with respect to the SoC Processor Core cache. DMA traffic is considered cacheable and checked for coherency with the SoC Processor Core cache.

Implications of this implementation are as follows:

- All code and tables for the LPE core need to be explicitly flushed from the SoC Processor Core cache if they are ever accessed.
- If the LPE core directly accesses data buffers in system DDR, the driver must explicitly flush the buffer from the SoC Processor Core cache
- If DMA accesses data buffers from system DRAM, the driver need not flush the data buffer from the SoC Processor Core cache.

12.4.2 Interrupts

12.4.2.1 LPE Peripheral Interrupts

Each of the LPE peripherals generates its own interrupts. SSP0, SSP1, and SSP2 have one interrupt each. Each of the DMA channels have individual interrupt lines. These interrupts are connected to the LPE core through the PISR register. The same interrupts are routed to IOAPIC through the ISRX register. The LPE core and SoC Processor Core have individual masks to enable these interrupts.

12.4.2.2 Interrupts Between SoC Processor Core and the LPE

The interrupts between the SoC Processor Core and the LPE are handled through the inter-processor communication registers. Whenever the SoC Processor Core writes to the IPCX communication register, an interrupt is generated to the LPE. The LPE firmware sees there is a message waiting from the SoC Processor Core, and reads the IPCX register for the data. This data is a pre-configured message, where the message structure has been decided beforehand between the SoC Processor Core and the LPE. Similarly we have the IPCD register for the communication between the LPE and SoC Processor Core. Once the LPE writes to the IPCD register, an interrupt should be generated for the SoC Processor Core and the SoC Processor Core should read the message from the IPCD register and act accordingly. From a software viewpoint, the mechanism remains the same as before. From a hardware view point, the interrupt to IA-32 gets routed by means of the IOAPIC block. The IPC from Audio to IA-32 gets a dedicated interrupt line to the IOAPIC.

12.4.2.3 Interrupts Between PMC and LPE

The interrupts between PMC and LPE are also handled using Inter Process Communication registers.

12.4.3 Power Management Options for the LPE Core

- WAITI
  - Allows the LPE core to suspend operation until an interrupt occurs by executing the optional WAITI instruction.
- External Run/Stall Control Signal
  - This processor input allows external logic to stall large portions of the LPE pipeline by shutting off the clock to much of the processor's logic to reduce operating power when the LPE computational capabilities are not immediately needed by the system.

Note: Using the WAITI instruction to power-down the processor will save more power than use of the external run/stall signal because the WAITI instruction disables more of the LPE's internal clocks.

12.4.4 External Timer

This timer always runs from SSP clock (before M/N divider) at 24/25 MHz. The timer starts running once the run bit (refer to the External timer register definition for details) is set and the clear bit is cleared.

The timer generates an Interrupt pulse when the counter value matches the "match" value. The interrupt does not get generated if the match value is set to "0". The timer runs in free running mode and rolls over after all 32 bits have become all 1s.

The timer continues to run as long as the run bit is set. Once the run bit is cleared, the timer holds the current value. The clear bit needs to be set to restart the timer from "0".

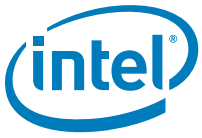
12.5 Clocks

12.5.1 Clock Frequencies

Table 12-2 shows the clock frequency options for the audio functional blocks.

Table 12-2. 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
SSP0 Clock	Fabric side: 50/OSC Link side: Up to 24 MHz	SSP0 clock domains
SSP1 Clock	Fabric side: 50/OSC Link side: Up to 24 MHz	SSP1 clock domains
SSP2 Clock	Fabric side: 50/OSC Link side: Up to 24 MHz	SSP2 clock domains



12.5.2 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 Clock Control Unit.

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

A calibrated Ring Oscillator in the CCU_SUS provides a 50 MHz or an 100 MHz 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.

12.5.4 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.

12.5.5 SSP Clocking

SSP could be used as either clock masters or clock slaves. Consequently, these IPs have dual clock domains.

The first clock domain is clocked from an internal clock (for example, fabric clock) and is used for generic logic like interrupt generation and register access.

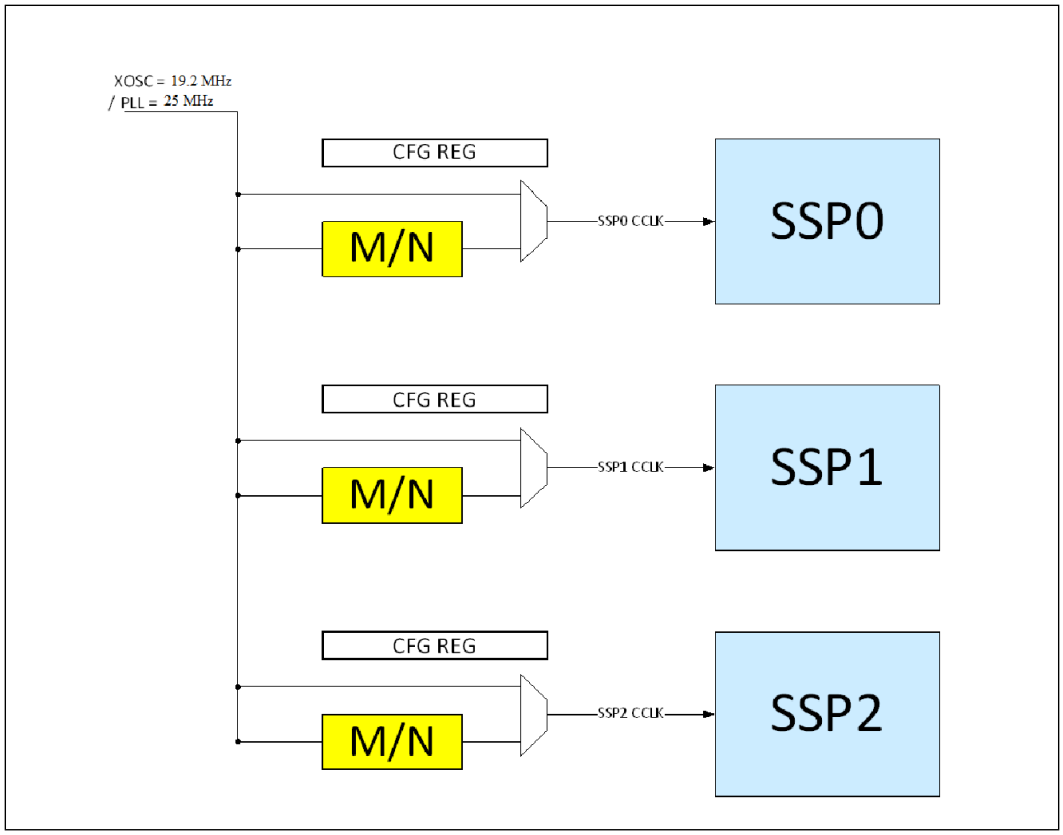
The second clock domain drives the serial shift register (either driven internally or externally). When driven internally, this clock can be sourced from XTAL clock 24 MHz or PLL 25 MHz. These clocks are then divided down within the serial interface IP to generate the final bit clock for the interface.

After power on, if the SSP input I/O clock is in high state, first transition of the clock from high to low may be missing due to the SoC clock gating logic.

12.5.6 M/N Divider

LPE SSP in master mode uses the SSP CCLK to drive the serial clock. It has very limited option to divide CCLK. An M/N divider is added between the 24 MHz clock (XOSC) from CCU to each SSP CCLK input as shown in following diagram.

Figure 12-3. SSP CCLK Structure



Note: The M/N divider has a bypass option so VLV could be configured to act same as TNG.

The LPE M/N divider is designed to produce a clock signal for the SSP block used in master mode. The divider is based on a generic NOM/DENOM divider. The supplied Master clock is 24 MHz (XTAL) or 25 MHz (LPPLL), but usually be used by the 25 MHz clock.

This mechanism is good for a wide spectrum of generated clocks. Two registers must be configured to get the target SSP clock. The values for the Nominator and Denominator registers are the smallest divider of:

$$\frac{Nominator}{Denominator} = \frac{Target\_clock}{Source\_clock}$$

12.5.6.1 Example

To generate a 17.64 MHz (= 400 x 44.1 KHz) output clock out of 25 MHz clock, you need to program "NOM = 441" and "DENOM = 625":

17.64 MHz = (441/625) x 25 MHz

In general the M over N can generate fractional devisor that could be used for generating the required clocks for audio codec. Table 12-3 describes some configuration options of this generic divider.

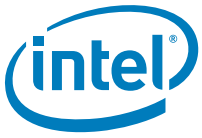


Table 12-3. M/N Values—Examples

Source Clock Frequency	Requested Clock	M/N Value
25 MHz	48 KHz	6/3125
	48K x 24 = 1.152 MHz	1152/25000
	48K x 32 = 1.536 MHz	1536/25000
	48K x 64 = 3.072 MHz	3072/25000
	44.1 KHz	441/250000
	48K x 400 = 19.2 MHz	96/125
	44.1K x 400 = 17.64 MHz	441/625

12.5.6.2 Accuracy and Jitter

The output of the M/N is equal to the desired clock in average with Jitter of 20nTXE for 25 MHz input clock.

12.5.6.3 Configuration

The following configurable fields per M/N divider/SSP are in LPE shim registers.

Table 12-4. M/N Configurable Fields

Field	Width	Description
Bypass	1 bit	When set M/N divider is bypass. Clock from CCU is connected directly to SSP CCLK
EN	1 bit	Enable the divider
Update	1 bit	Update divider parameters
M Value	20 bits	Nominator value
N Value	20 bits	Denominator value

12.6 SSP (I^2S)

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^2S. 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.

12.6.1 Introduction

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 25Mbps, 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.

12.6.2 SSP 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
- 12.5Mbps 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.

12.6.3 Operation

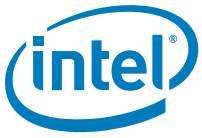
Serial data is transferred between the LPE core or the SoC Processor Core and an external peripheral through FIFOs in one of the SSP ports. Data transfers between an SSP port and memory are initiated by either the LPE core or the SoC Processor Core using programmed I/O, or by DMA bursts. Although it is possible to initiate transfers directly from the SoC Processor Core, current driver design uses LPE for all PCM operations. Separate transmit and receive FIFOs and serial data paths permit simultaneous transfers in both directions to and from the external peripheral, depending on the protocols chosen.

Programmed I/O can transfer data between:

- The LPE core and the FIFO Data register for the TXFIFO
- The SoC Processor Core and the FIFO Data register for the TXFIFO
- The LPE core and the FIFO Data register for the RXFIFO
- The SoC Processor Core and the FIFO Data register for the RXFIFO
- The SoC Processor Core and the control or status registers
- The LPE core and the control or status registers

DMA bursts can transfer data between:

- Universal memory and the FIFO Data register for the TXFIFO
- Universal memory and the FIFO Data register for the RXFIFO
- Universal memory and the sequentially addressed control or status registers



12.6.4 LPE and DMA FIFO Access

The LPE or DMA access data through the Enhanced SSP Port's Transmit and Receive FIFOs. An LPE access takes the form of programmed I/O, transferring one FIFO entry per access. LPE accesses would normally be triggered off of an SSSR Interrupt and must always be 32-bits wide. LPE Writes to the FIFOs are 32-bits wide, but the serializing logic will ignore all bits beyond the programmed FIFO data size (EDSS/DSS value). LPE Reads to the FIFOs are also 32-bits wide, but the Receive data written into the RX FIFO (from the RXD line) is stored with zeroes in the MSBs down to the programmed data size. The FIFOs can also be accessed by DMA bursts, which must be in multiples of 1, 2 or 4 bytes, depending upon the EDSS value, and must also transfer one FIFO entry per access. When the SSCR0.EDSS bit is set, DMA bursts must be in multiples of 4 bytes (the DMA must have the Enhanced SSP configured as a 32-bit peripheral).The DMA's width register must be at least the SSP data size programmed into the SSP control registers EDSS and DSS. The FIFO is seen as one 32-bit location by the processor. For Writes, the Enhanced SSP port takes the data from the Transmit FIFO, serializes it, and sends it over the serial wire (I2S[2:0]_DATAOUT) to the external peripheral. Receive data from the external peripheral (on I2S[2:0]_DATAIN) is converted to parallel words and stored in the Receive FIFO.

A programmable FIFO trigger threshold, when exceeded, generates an Interrupt or DMA service request that, if enabled, signals the processor or DMA respectively to empty the Receive FIFO or to refill the Transmit FIFO.

The Transmit and Receive FIFOs are differentiated by whether the access is a Read or a Write transfer. Reads automatically target the Receive FIFO, while Writes will write data to the Transmit FIFO. From a memory-map perspective, they are at the same address. FIFOs are 16 samples deep by 32-bits wide. Each read or write is to a 1 SSP sample.

12.6.5 Supported Formats

The SSP consists of four pins that are used to transfer data between the SoC and external audio codecs, modems, or other peripherals. Although four serial-data formats exist, each has the same basic structure, and in all cases the following pins are used in the following manner:

- I2Sx_CLK—Defines the bit rate at which serial data is driven onto and sampled from the port
- I2Sx_FRM—Defines the boundaries of a basic data "unit," comprised of multiple serial bits
- I2Sx_DATAIN—The serial data path for received data, from system to peripheral
- I2Sx_DATAOUT—The serial data path for transmitted data, from peripheral to system

A data frame can contain from 4- to 32-bits, depending on the selected format. Serial data is transmitted most significant bit first. The Programmable Serial Protocol (PSP) format is used to implement I^2S.

Master and Slave modes are supported. When driven by the Enhanced SSP, the I2Sx_CLK only toggles during active transfers (not continuously) unless ECRA/ECRB functions are used. When the I2Sx_CLK is driven by another device, it is allowed to be either continuous or only driven during transfers, but certain restrictions on PSP parameters apply.

Normally, the serial clock (I2Sx_CLK), if driven by the Enhanced SSP Port, only toggles while an active data transfer is underway. There are several conditions, however, that may cause the clock to run continuously. If the Receive With Out Transmit mode is enabled by setting the SSCR1.RWOT bit to 1, the I2Sx_CLK will toggle regardless of whether Transmit data exists within the Transmit FIFO. The I2Sx_CLK will also toggle continuously if the Enhanced SSP is in Network mode, or if ECRA, or ECRB is enabled. At other times, I2Sx_CLK will be held in an inactive I2Sx_FRM or idle state, as defined by the specified protocol under which it operates.

12.6.5.1 Programmable Serial Protocol (PSP)

There are many variations of the frame behavior for different codecs and protocol formats. To allow flexibility the PSP format allows I2Sx_FRM to be programmable in direction, delay, polarity, and width. Master and Slave modes are supported. PSP can be programmed to be either full or half duplex.

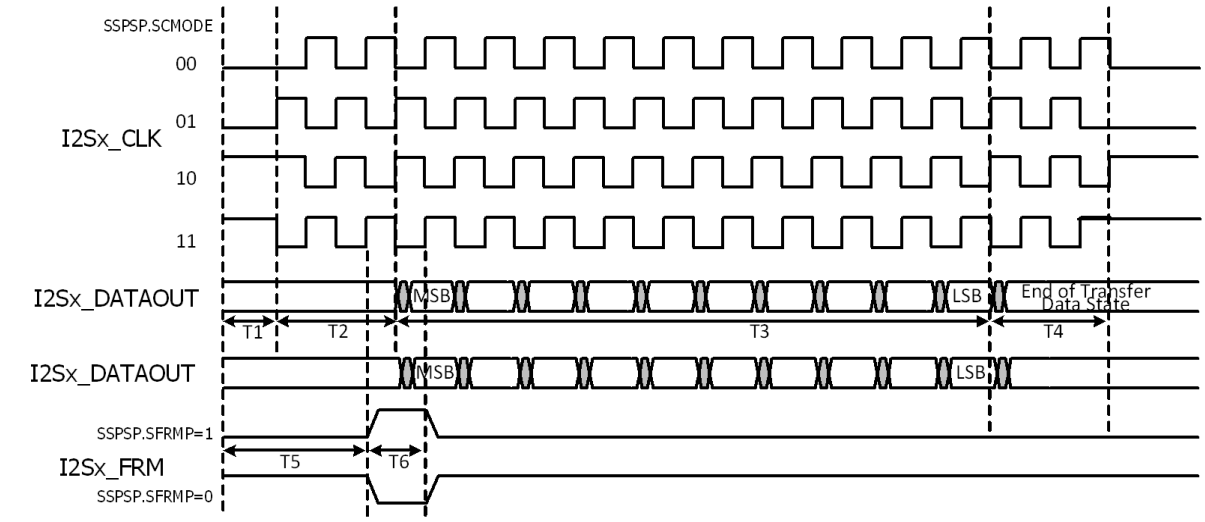
The I2Sx_CLK function behavior varies between each format. PSP lets programmers choose which edge of I2Sx_CLK to use for switching Transmit data, and for sampling Receive data. In addition, programmers can control the idle state for I2Sx_CLK and the number of active clocks that precede and follow the data transmission.

The PSP format provides programmability for several parameters that determine the transfer timings between data samples. There are four possible serial clock sub-modes, depending on the I2Sx_CLK edges selected for driving data and sampling received data, and the selection of idle state of the clock.

For the PSP format, the Idle and Disable modes of the I2Sx_DATAOUT, I2Sx_CLK, and I2Sx_FRM are programmable by means of the SSPSP.ETDS, SSPSP.SCMODE and SSPSP.SFRMP bits. When Transmit data is ready, the I2Sx_CLK will remain in its Idle state for the number of serial clock (I2Sx_CLK) clock periods programmed within the Start Delay (SSPSP.STRTDLY) field. I2Sx_CLK will then start toggling, I2Sx_DATAOUT will remain in the idle state for the number of cycles programmed within the Dummy Start (SSPSP.DMYSTRT) field. The I2Sx_FRM signal will be asserted after the number of half-clocks programmed in the SSPSP.SFRDLY field. The I2Sx_FRM signal will remain asserted for the number of clocks programmed within the SSPSP.SFRMWDTH then de-assert. Four to 32 bits can be transferred per frame. Once the last bit (LSB) is transferred, the I2Sx_CLK will continue toggling based off the Dummy Stop (SSPSP.DMYSTOP) field. I2Sx_DATAOUT either retains the last value transmitted or is forced to zero, depending on the value programmed within the End of Transfer Data State (SSPSP.ETDS) field, when the controller goes into Idle mode, unless the Enhanced SSP port is disabled or reset (which forces I2Sx_DATAOUT to zero).

With the assertion of I2Sx_FRM, Receive data is simultaneously driven from the peripheral on I2Sx_DATAIN, most significant bit first. Data transitions on I2Sx_CLK based on the Serial Clock Mode selected and is sampled by the controller on the opposite edge. When the Enhanced SSP is a master to the frame synch (I2Sx_FRM) and a slave to the clock (I2Sx_CLK), then at least three extra clocks (I2Sx_CLKs) will be needed at the beginning and end of each block of transfers to synchronize control signals from the APB clock domain into the SSP clock domain (a block of transfers is a group of back-to-back continuous transfers).

Figure 12-4. Programmable Serial Protocol Format



Note: When in PSP format, if the SSP is the master of the clock (I2Sx_CLK is an output) and the SSPSP.ETDS bit is cleared, the End of Transfer Data State for the I2Sx_DATAOUT line is 0. If the SSP is the master of the clock, and the SSPSP.ETDS bit is set, the I2Sx_DATAOUT line remains at the last bit transmitted (LSB). If the SSP is a slave to the clock (I2Sx_CLK is an input), and modes 1 or 3 are used, the ETDS can only change from the LSB if more clocks (I2Sx_CLK) are sent to the SSP (that is, dummy stop clocks or slave clock is free running.

Figure 12-5. Programmable Serial Protocol Format (Consecutive Transfers)

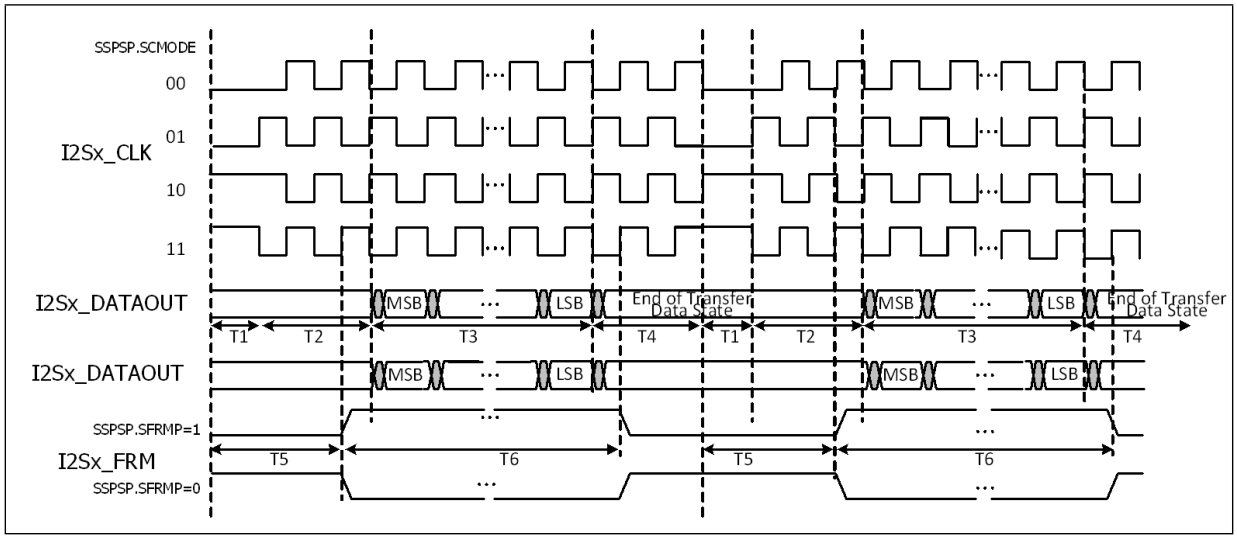


Table 12-5. Programmable Protocol Parameters

Symbol	Definition (Register.Bit Field)	Range	Units
	Serial Clock Mode (SSPSP.SCMODE)	(Drive, Sample, I2Sx_CLK Idle) 0 = Fall, Rise, Low 1 = Rise, Fall, Low 2 = Rise, Fall, High 3 = Fall, Rise, High	
	Serial Frame Polarity (SSPSP.SFRMP)	High or Low	
T1	Start Delay (SSPSP.STRTDLY)	0–7	Clock Period
T2	Dummy Start (SSPSP.DMYSTRT)	0–3	Clock Period
T3	Data Size (SSCRO.EDSS AND SSCRO.DSS)	4–32	Clock Period
T4	Dummy Stop (SSPSP.DMYSTOP)	0–3	Clock Period
T5	I2Sx_FRM Delay (SPSP.SFRMDLY)	0–88	Half Clock Period
T6	I2Sx_FRM Width (SSPSP.SFRMWDTH)	1–44	Clock Period
	End of Transfer Data State (SSPSP.ETDS)	Low or [bit 0]	

The I2Sx_FRM Delay must not extend beyond the end of T4. I2Sx_FRM Width must be asserted for at least 1 I2Sx_CLK, and should be de-asserted before the end of the T4 cycle (for example, in terms of time, not bit values, $(T5 + T6) \leq (T1 + T2 + T3 + T4)$, $1 \leq T6 < (T2 + T3 + T4)$, and $(T5 + T6) \geq (T1 + 1)$ to ensure that I2Sx_FRM is asserted for at least 2 edges of the I2Sx_CLK). The T1 Start Delay value should be programmed to 0 when the I2Sx_CLK is enabled by either of the SSCR1.ECRA or SSCR1.ECRB bits. While the PSP can be programmed to generate the assertion of I2Sx_FRM during the middle of the data transfer (after the MSB was sent), the Enhanced SSP will not be able to receive data in Frame slave mode (SSCR1SFRMDIR = 1) if the assertion of Frame is not before the MSB is sent (that is $T5 \leq T2$ if SSCR1.SFRMDIR =1). Transmit Data will transition from the "End of Transfer Data State" to the next MSB value upon the assertion of Frame. The Start Delay field should be programmed to 0 whenever I2Sx_CLK or I2Sx_FRM is configured as an input. Clock state is not defined between two active frame periods. Clock can be active or inactive between two active frame periods.

§ §

13 Intel® Trusted Execution Engine (Intel® TXE)

This section describes the security components and capabilities of the Intel Trusted Execution Engine (TXE) security co-processor.

Note: TXE firmware is required on the processor platform as part of the PCU SPI flash image. The PCU SPI interface must be operating in descriptor mode in order for the TXE to be able to access its firmware.

13.1 Features

13.1.1 Security Feature

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

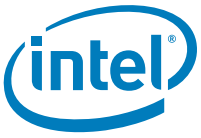
Intel® TXE features:

- 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 Processor 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 Processor address domain (System memory) and the Intel® TXE; programmable by the Intel® TXE processor only.

13.1.1.1 Hardware 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

§ §



14 Intel® High Definition Audio (Intel® HD Audio)

The Intel® High Definition Audio (Intel® HD Audio) is an architecture and infrastructure to support high-quality audio implementations for PCs.

The Intel® High Definition Audio (Intel® HD Audio) controller consists of a set of DMA engines that are used to move samples of digitally encoded data between system memory and internal/external codecs. The controller communicates with the internal/external codecs over the Intel® HD Audio serial link.The output DMA engines move digital data from system memory to a D-A converter in a codec. The SoC implements a single Serial Data Output (SDO) signal that is connected to the external codecs. The input DMA engines move digital data from the A-D converter in the codec to system memory. The platform supports up to two external codecs by implementing two Serial Data Input (SDI) signals, each being dedicated to a single codec.

Audio software renders outbound, and processes inbound data to/from buffers in memory. The location of the individual buffers is described by a Buffer Descriptor List that is fetched and processed by the audio controller. The data in the buffers is arranged in a pre-defined format. The output DMA engines fetch the digital data from memory and reformat it based on the programmed sample rate, bits/sample and number of channels. The data from the output DMA engines is then combined and serially sent to the codec(s) over the Intel® HD Audio link. The input DMA engines receive data from the codec(s) over the Intel® HD Audio link and format the data based on the programmable attributes for that stream. The data is then written to memory in the predefined format for software to process. Each DMA engine moves one “stream” of data. A single codec can accept or generate multiple “streams” of data, one for each A-D or D-A converter in the codec. Multiple codecs can accept the same output “stream” processed by a single DMA engine.

Codec commands and responses are also transported to and from the codec by means of DMA engines. The DMA engine dedicated to transporting commands from the Command Output Ring Buffer (CORB) in memory to the codec(s) is called the CORB engine. The DMA engine dedicated to transporting responses from the codec(s) to the Response Input Ring Buffer in memory is called the RIRB engine. Every command sent to a codec yields a response from that codec. Some commands are “broadcast” type commands in which case a response will be generated from each codec. A codec may also be programmed to generate unsolicited responses, which the RIRB engine also processes. The platform also supports Programmed I/O-based Immediate Command/Response transport mechanism that can be used by BIOS prior to memory initialization.

14.1 Signal Descriptions

Table 14-1. Signals Description

Signal Name	Direction/ Type	Description
HDA_RST#	O	**Intel HD Audio Reset:** Master hardware reset to external codecs
HDA_SYNC	O	**Intel HD Audio Sync:** 48 KHz fixed rate
HDA_CLK	O	**Intel HD Audio Bit Clock (Output):** 24 MHz serial data clock generated by the Intel® HD Audio controller
HDA_SDO	O	**Intel HD Audio Data Out:** Serial TDM data output to the codec(s). The serial output is double-pumped for a bit rate of 48Mb/s
HDA_SDI[1:0]	I	**Intel HD Audio Serial Data In[1:0]:** Serial TDM data input from the codec(s). The serial input is single-pumped for a bit rate of 24Mb/second.
HDA_DOCKEN#	O	**Intel HD Audio Docking Enable:** Enable audio docking isolation logic.
HDA_DOCKRST#	O	**Intel HD Audio Docking Reset:** Audio docking station reset

The signals in the table above are all multiplexed and maybe used by other functions.

14.2 Features

The Intel® HD Audio Controller supports the following features:

- Supports MSI and legacy interrupt delivery
- Support for ACPI D3 and D0 Device States
- Supports up to:
  - 6 streams (three input, three output)
  - 16 channels per stream
  - 32 bits/sample
  - 192 KHz sample rate
  - 24 MHz HDA_CLK supports
  - SDO double pumped at 48Mb/s
  - SDI single pumped at 24Mb/s
- Supports 1.5V and 1.8V mode
- Supports optional Immediate Command/Response mechanism

14.3 References

High Definition Audio Specification, Revision 1.0a

- http://www.intel.com/content/dam/www/public/us/en/documents/product-specifications/high-definition-audio-specification.pdf

§ §

15 Serial I/O (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—Serial Peripheral SPI
- SIO—I^2C Interface
- SIO—High Speed UART

Note: SIO Serial Peripheral Interface (SPI) is supported for a platform supporting non-Windows operating systems only.

15.1 Register Map

For more information on SIO registers, refer to the Processor Datasheet Volume 2 and Volume 3 (See Related Documents section).

15.2 SIO—Serial Peripheral Interface (SPI)

The Serial I/O implements one SPI controller that supports master mode.

Note: SIO SPI can operate up to 20 MHz

15.2.1 Signal Descriptions

Table 15-1. SPI Interface Signals

Signal Name	Direction/Type	Description
SPI1_CLK	O GPIO	**SPI Serial Clock**
SPI1_CS[1:0]#	O GPIO	**SPI Chip Select** SPI Chip Select is active low.
SPI1_MOSI	O GPIO	**SPI Master Output Slave Input** Serial Data In
SPI1_MISO	I GPIO	**SPI Slave Output Master Input** Serial Data out

15.2.1.1 Clock Phase and Polarity

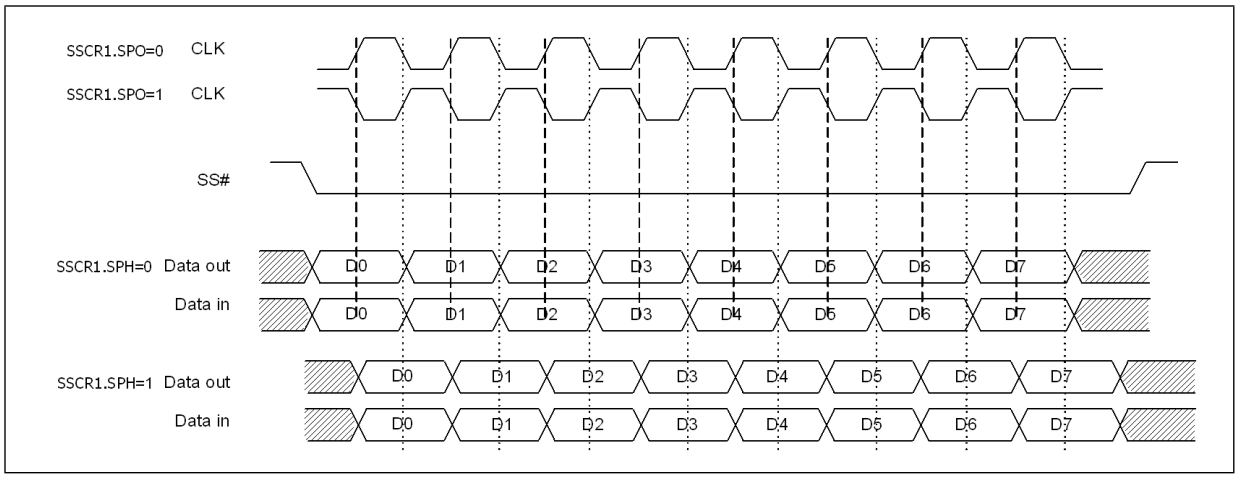
SPI clock phase and clock polarity overview:

- 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 SIO_SPI_CLK is low.
  - — When SSCR1.SPO = 1, the inactive or idle state of SIO_SPI_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, SIO_SPI_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 = 0, SIO_SPI_CLK is inactive until 1/2 cycle after the start of a frame and active until one cycle after the end of a frame.

Figure 15-1 shows an 8-bit data transfer with different phase and polarity settings.

Figure 15-1. 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.

15.2.2 SIO—I^{2}C Interface

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

15.2.3 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 seven I^{2}C interfaces for general purpose to control external devices. The I^{2}C signals are multiplexed over GPIOs.

Table 15-2. I^{2}C [6:0] Signals

Signal Name	Direction/ Type	Description
I2C[6:0]_DATA	I/O CMOS1.8	**I^{2}C Serial Data** These SIO I^{2}C signals are multiplexed and may be used by other functions.
I2C[6:0]_CLK	I/O CMOS1.8	**I^{2}C Serial Clock** These SIO I^{2}C signals are multiplexed and may be used by other functions.

15.2.4 Features

15.2.4.1 I²C Protocol

The I²C 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 I²C 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 to Chapter 21, "Electrical Specifications" for details.
- Data is transmitted in byte packages.

15.2.4.2 I²C Modes of Operation

The I²C module can operate in the following modes:

- Standard mode (bit rate up to 100Kb/s)
- Fast mode (bit rate up to 400Kb/s)
- Fast Mode Plus (bit rate up to 1Mb/s)
- High-Speed mode (bit rate up to 1.7Mb/s)

The I²C 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.

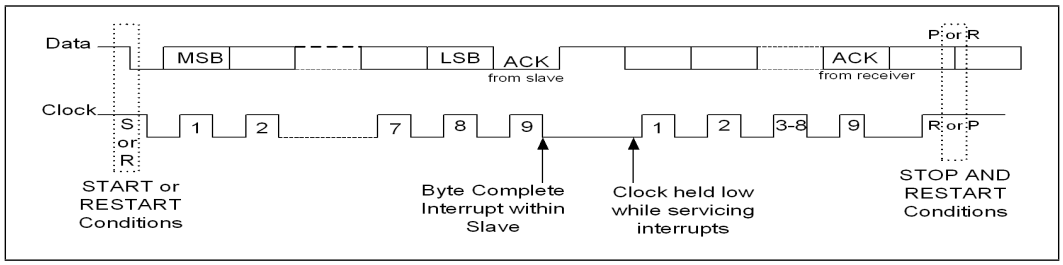
- High-Speed mode devices can communicate with fast mode and standard mode devices in a mixed speed bus system.
- Fast mode devices can communicate with standard mode devices in a 0–100Kb/s I²C bus system.

However, according to the I²C specification, standard mode devices are not upward compatible and should not be incorporated in a fast-mode I²C bus system since they cannot follow the higher transfer rate and unpredictable states would occur.

15.2.4.3 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 the following figure.

Figure 15-2. Data Transfer on the I^2C Bus



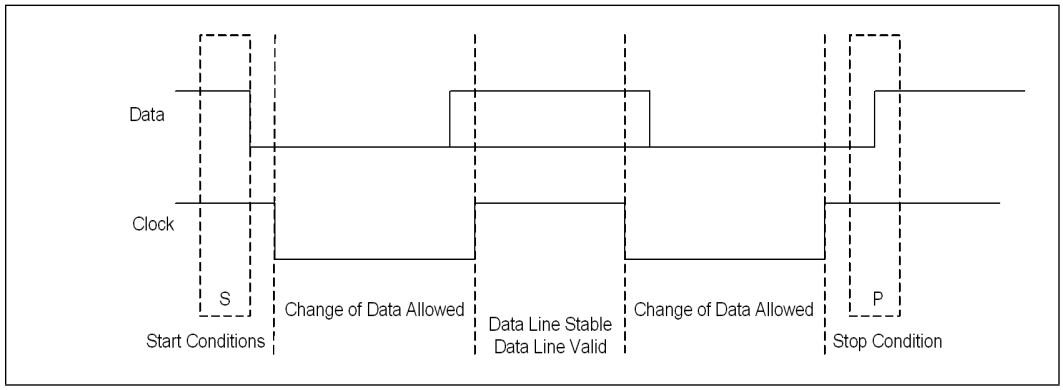
15.2.4.3.1 START and STOP Conditions

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 15-3 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 15-3. 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^2C 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.

15.2.5 References

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

15.2.6 Register Map

Refer to Chapter 23, Register Access Method Registers and Chapter 24, Mapping Address Space Registers in Volume 2 for additional information.

15.3 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.

15.3.1 Signal Descriptions

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

Table 15-3. UART 1 Interface Signals

Signal Name	Direction/ Type	Description
UART1_RXD	I GPIOMV, MS	**High Speed UART receive data input** This signal is multiplexed and may be used by other functions.
UART1_TXD	O GPIOMV, MS	**High Speed UART transmit data** This signal is multiplexed and may be used by other functions.
UART1_RTS#	O GPIOMV, MS	**High Speed UART request to send** This signal is multiplexed and may be used by other functions.
UART1_CTS#	I GPIOMV, MS	**High Speed UART clear to send** This signal is multiplexed and may be used by other functions.

Table 15-4. UART 2 Interface Signals

Signal Name	Direction/ Type	Description
UART2_RXD	I GPIOMV, MS	**High Speed UART receive data input** This signal is multiplexed and may be used by other functions.
UART2_TXD	O GPIOMV, MS	**High Speed UART transmit data** This signal is multiplexed and may be used by other functions.
UART2_RTS#	O GPIOMV, MS	**High Speed UART request to send** This signal is multiplexed and may be used by other functions.
UART2_CTS#	I GPIOMV, MS	**High Speed UART clear to send** This signal is multiplexed and may be used by other functions.

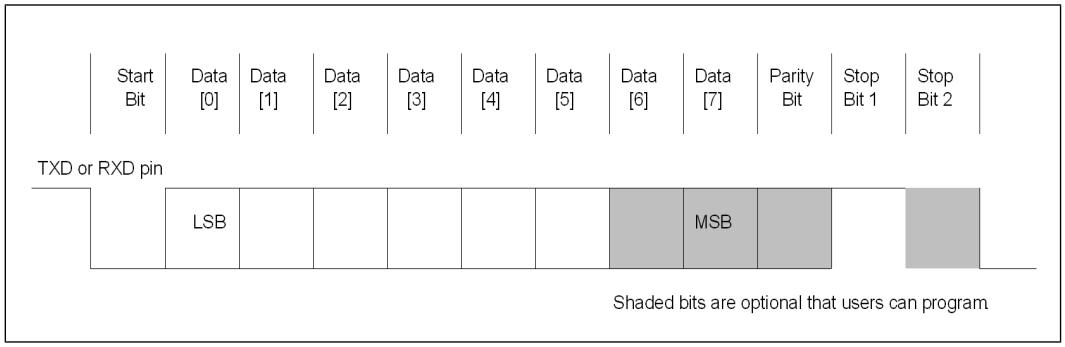
15.3.2 Features

15.3.2.1 UART Function

The UART transmits and receives data in bit frames as shown in Figure 15-4.

- 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. 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), that is represented by one or two successive bit periods of a logic one.

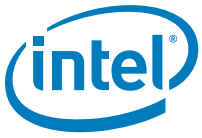
Figure 15-4. UART Data Transfer



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.

15.3.2.2 Clock and Reset

The BAUD rate generates from base frequency of 50 MHz.



15.3.2.3 Baud Rate Generator

The baud rates for the UARTs are generated with from the base frequency (Fbase) indicated in Table 15-5 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 15-5. 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

15.3.3 Use

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

15.3.3.1 DMA Mode Operation

15.3.3.1.1 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”. Software needs 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.

15.3.3.1.2 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”. Software needs 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.

15.3.3.1.3 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.

15.3.3.2 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.

15.3.3.2.1 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).

15.3.3.2.2 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.

15.3.3.2.3 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 de-asserted. 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.

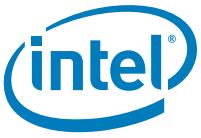
15.3.3.2.4 RTS (UART Output)

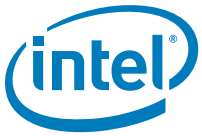
When in full autoflow mode, nRTS is asserted when the UART FIFO is ready to receive data from the remote transmitter. This occurs when the amount of data in the Receive FIFO is below the programmable threshold value. When the amount of data in the Receive FIFO reaches the programmable threshold, nRTS 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.

15.3.3.2.5 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.

§ §





16 Platform Controller Unit (PCU) Overview

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

- "PMU—Power Management Controller (PMC)"
- "PCU—Serial Peripheral Interface (SPI)"

  — For boot FW and system configuration data Flash storage

  Note: Flash Sharing is not supported for the processor Platforms

  Note: Fast_SPI signals do not get tri-stated during RSMRST# assertion.
- "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.

16.1 PCU Configuration Features for BIOS/EFI Boot Overview

16.1.1 BIOS/EFI Top Swap

While updating the BIOS/EFI boot sector in flash, unexpected system power loss can cause an incomplete write resulting in a corrupt boot sector. For this reason, two boot sectors are stored in the flash.

The location of the secondary boot sector is defined by inverting one of the bits of the address (A16, A17, or A18) that the processor core will attempt to fetch code from. This address bit will vary depending on the size of the boot block. BBSize register bit definition for further details.

Prior to starting writes to the primary BIOS/EFI boot sector, the Top-Swap indicator is set. From this point onwards, the secondary boot sector will be used. Only after successful completion of the primary boot sector write should the Top-Swap indicator be cleared and the primary boot sector be used again.

There are two methods that can be used to implement the Top-Swap indicator.

16.1.1.1 BIOS/EFI Controlled

BIOS/EFI can use the GCS.TS register bit to set the Top-Swap indicator. The GCS.TS bit is stored in the RTC well and, therefore, keeps its value even when the system is powered down.

Note: Writes to GCS.TS will be unsuccessful if the GCS.BILD bit has been set.

16.1.1.2 Hardware Controlled

System hardware, external to the SoC, can be used to assert or de-assert the Top-Swap strapping input signal. If the signal is sampled as being asserted during power-up, then Top-Swap is active.

Note: The Top-Swap strap is an active low signal and is multiplexed with the GPIO_SUS2 signal.

Note: The Top-Swap strap, when asserted at power-up, forces Top-Swap to be active even if GCS.TS bit is cleared but does not change the GCS.TS bit itself. The GCS.TS bit can not be changed if Top-Swap pin strap was sampled as being asserted until the next power-up when Top-Swap is sampled as being de-asserted.

16.1.2 BIOS/EFI Boot Strap

BIOS/EFI may be booted from the PCU SPI interface or the iLB LPC interface. The choice of SPI or LPC is configured by the BIOS/EFI Boot Strap (BBS). The possible configurations of the BBS are indicated in Table 16-1.

Note: 1) The BBS is multiplexed with the GPIO_SUS4 signal.

2) BIOS/EFI boot from the LPC interface is not available when Secure Boot is enabled.

Table 16-1. BBS Configurations

BBS Level	Description
Low (0b)	Boot from LPC
High (1b)	Boot from SPI

16.2 PMU—Power Management Controller (PMC)

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

16.2.1 Signal Descriptions

Note: These signals are part of CFIO (GPIO) and may be used by other functions.

Table 16-2. PMC Signals (Sheet 1 of 2)

Signal Name	Direction/ Type	Description
PMU_AC_PRESENT	I CMOS V1P8	**AC Present:** This input pin indicates when the platform is plugged into AC power.
PMU_BATLOW#	I CMOS V1P8	**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# when asserted. In desktop configurations without a battery, this signal should be tied high to V1P8_S5.
PMU_PLTRST#	O CMOS V1P8	**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.

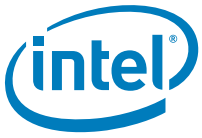


Table 16-2. PMC Signals (Sheet 2 of 2)

Signal Name	Direction/ Type	Description
PMU_PWRBTN#	I CMOS V1P8	**Power Button:** The signal will cause SMI# 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 debounce on the input.
PMU_RESETBUTTON#	I CMOS V1P8	**System Reset:** This signal forces an internal reset after being debounced.
PMU_SLP_S4#	O CMOS V1P8	**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.
PMU_SLP_S3#	CMOS V1P8	**S3 Sleep Control:** This signal is for power plane control. It can be used to control system power when it is in a S3 (Suspend To RAM), S4 (Suspend to Disk), or S5 (Soft Off) states.
PMU_SUS_STAT#	O CMOS V1P8	**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.
PMU_SUSCLK	O CMOS V1P8	**Suspend Clock:** This 32 KHz clock is an output of the RTC generator circuit for use by other chips for refresh clock.
PMU_SUSPWRDNACK	O CMOS V1P8	**Suspend Power Down Acknowledge:** Asserted by the SoC when it does not require its Suspend well to be powered.

16.2.2 Features

16.2.2.1 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.

- **PMU_PWRBTN#:** PMU_PWRBTN# is always enabled as a wake event. When RSMRST# is low (G3 state), the PM1_STS_EN.PWRBTN_STS bit is reset. When the SoC exits G3 after power returns (RSM_RST# goes high), the PMU_PWRBTN# signal is already high (because the suspend plane goes high before RSM_RST# 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 RSM_RST# goes low.

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

Table 16-3. 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

16.2.2.2 Event Input Signals and Their Usage

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

16.2.2.2.1 PMU_PWRBTN# (Power Button)

The PMU_PWRBTN# 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 16-4.

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

Note: During the time that the PMU_SLP_S4# signal is stretched for the minimum assertion width (if enabled), the power button is not a wake event.

Note: See below for more details.

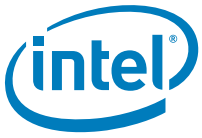
Table 16-4. Transitions Due to Power Button

Present State	Event	Transition/Action	Comment
S0/Cx	PMU_PWRBTN# goes low	SMI# 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	PMU_PWRBTN# goes low	Wake Event. Transitions to S0 state	Standard wakeup
G3	PMU_PWRBTN# pressed	None	No effect since no power Not latched nor detected
S0, S4	PMU_PWRBTN# held low for at least 4 consecutive seconds	Unconditional transition to S5 state	No dependence on processor or any other subsystem

Power Button Override Function

If PMU_PWRBTN# 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 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 PMU_ PWRBTN# status is readable to check if the button is currently being pressed or has been released. The status is taken after the debounce, and is readable using the GEN_PMCON2.PWRBTN_LVL bit.



Note: The 4-second PMU_PWRBTN# 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 PMU_PWRBTN# 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# 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-second 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 PMU_SLP_S4# power-cycle expires, the power button must be pressed for another 4 to 5 seconds to create the override condition to S5.

16.2.2.2.2 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. See the Advanced Configuration and Power Interface specification for implementation details.

16.2.2.2.3 PME_B0 (PCI Power Management Event Bus 0)

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

16.2.2.2.4 PMU_RSTBTN# Signal

When the PMU_RSTBTN# 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 5–6 ms regardless of whether the PMU_RSTBTN# input remains asserted or not. It cannot occur again until PMU_RSTBTN# has been detected inactive after the debounce logic, and the system is back to a full S0 state with PMU_PLTRST# inactive.

Note: If RST_CNT.FULL_RST is set, then PMU_RSTBTN# will result in a full Power-cycle Reset.

16.2.2.3 System Power Planes

The system has several independent power planes, as described in Table 16-5.

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

Table 16-5. System Power Planes

Plane	Controlled By	Description
Devices and Memory	PMU_SLP_S4#	When PMU_SLP_S4# 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 PMU_SLP_S5# 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	PMU_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.

16.2.2.3.1 Power Plane Control with PMU_SLP_S4#

The PMU_SLP_S4# 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.

16.2.2.3.2 PMU_SLP_S4# and Suspend-To-RAM Sequencing

The system memory suspend voltage regulator is controlled by Glue logic. The PMU_SLP_S4# signal should be used to remove power to system memory. The PMU_SLP_S4# 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 PMU_SLP_S4# signal.

16.2.2.3.3 CORE_PWROK Signal

When asserted, CORE_PWROK is an indication to the SoC that its core well power rails are powered and stable. CORE_PWROK can be driven asynchronously. When CORE_PWROK is low, the SoC asynchronously asserts PMU_PLTRST#. CORE_PWROK must not glitch, even if PMU_RSMRST# 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 PMU_PLTRST# de-assertion.

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

16.2.2.3.4 PMU_BATLOW# (Battery Low)

The PMU_BATLOW# 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.

16.2.2.4 SMI#/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 Processor 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# 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 16-6 shows which events can cause an SMI and SCI.

Note: 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 16-6. Causes of SMI and SCI (Sheet 1 of 3)

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

Table 16-6. Causes of SMI and SCI (Sheet 2 of 3)

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
GPI[n][9]	GPE0a_STS. CORE_GPIO_STS[n][2] or GPE0a_STS. SUS_GPIO_STS[n][2]	GPIO_ROUT[n] = 10b and 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# pin goes low	GPE0a_STS. BATLOW_STS#	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 G-unit[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	
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 G-unit[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 and ALT_GPIO_SMI. CORE_GPIO_SMI_EN[n][2]=1b or ALT_GPIO_SMI. SUS_GPIO_SMI_EN[n][2]=1b	SMI		None	

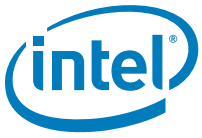


Table 16-6. Causes of SMI and SCI (Sheet 3 of 3)

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
USB Per-Port Registers Write Enable bit is changed from 0b to 1b	UPRWC.WE_STS and SMI_STS. USB_IS_STS	UPRWC. WE_SMI_E=1b and SMI_EN. USB_IS_SMI_EN=1b	SYNC SMI[6]		None	

Notes:
1. Most of the status bits (except otherwise noted) are set according to event occurrence regardless to the enable bit.
2. GPIO status bits are set only if enable criteria is true. Refer to the processor Datasheet Volume 3 Section 35.2.17 and 35.4.4 for more details (see Related Documents section).
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 is not cleared prior to setting PM1_CNT.SCI_EN. Refer to processor Datasheet Volume 3 Section 35.4.1 for more details (see Related Documents section).
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. Refer to the processor Datasheet Volume 3 Section 35.4.1 for more details (see Related Documents section).
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. Refer to the processor Datasheet Volume 3 of 3 Section 35.4.4 for more details (see Related Documents section).
10. Refer to the processor Datasheet Volume 3 of 3 Section 35.4.8 for more details (see Related Documents section).

16.2.2.5 Platform Clock Support

The SoC supports up to 6 clocks (PMU_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.

16.2.2.6 INIT# (Initialization) Generation

The INIT# 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# is asserted for 16 PCI clocks and then driven high.

INIT#, when asserted, resets integer registers inside the Processor 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 16-7. INIT# 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

16.2.3 References

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

16.3 PCU—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 20 MHz, 33 MHz, or 50 MHz. Dual and Quad I/O Flash devices are supported in addition to standard flash devices.

Note: The default interface speed is 20 MHz. Frequency capability of the Flash Component should be higher than the maximum set frequency of SPI bus.

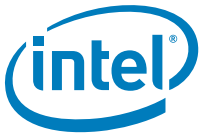
Note: Flash Sharing is not supported for the processor platforms.

Note: Fast_SPI signals do not get tri-stated during RSMRST# assertion.

16.3.1 Signal Descriptions

Table 16-8. SPI Signals

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



16.3.2 Features

1)Descriptor has 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

b)Supports Flash that is divided into 4 regions and accessible by two masters

i)Regions (4)

(1) Flash Descriptor and Chipset Soft Straps

(2) BIOS

(3) TXE

(4) Platform Data

ii)Masters (3)

(1) Host Processor (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 pull-up/pull-down 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.

b)The top of the Flash Descriptor contains the Flash Upper Map

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

b)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.

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)2sets (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 Hardware to provide the basic instructions of read, write, and erase.

ii)Software Sequencing

(1)Allows software to use any legal Opcode

c)Support for Boot BIOS on SPI or LPC FWH.

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.

Note: No validation has been done with BIOS on LPC. Reference design uses SPI

d)Prefetching/Caching to improve performance

i)Separate 64B prefetch/cache each for HOST and TXE direct read accesses

4)SFDP Parameter Discoverability[1]

5)Flash Component Capabilities

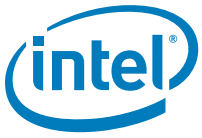
a)In Descriptor mode, supports two SPI Flash components using two separate chip select pins, CS0# and CS1#. Only one component supported in non-descriptor mode.

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 20 MHz, 33 MHz 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 Dual and 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 two 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 first flash component (the one with the Flash Descriptor) must be of binary size. The second 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 two value larger than the non-binary size.

8)Reset Capabilities

a)RSMRST#

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

16.4 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 I/O.

Note: Only a minimal ball-count, comprising receive and 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.

16.4.1 Signal Descriptions

Table 16-9. UART Signals

Signal Name	Direction/ Type	Description
UART0_RXD	I GPIO UART	**COM1 Receive:** Serial data input from device pin to the receive port. This signal is multiplexed and may be used by other functions. ***Note:*** Refer to Section 2.4, "Hardware Straps" on page 40 to get more details
UART0_TXD	O GPIO UART	**COM1 Transmit:** Serial data output from transmit port to the device pin. This signal is multiplexed and may be used by other functions. ***Note:*** Refer to Section 2.4, "Hardware Straps" on page 40 to get more details
Notes: 1. These signals are part of PCU Logic. 2. Among others, these signals are multiplexed with LPC_CLKRUN_N and LPC_FRAME_N (Enabled in Mode2).		

16.4.2 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:

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

The divisor is defined by the Divisor Latch LSB and Divisor Latch MSB registers. Some common values are shown in Table 16-10.

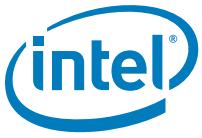


Table 16-10. Baud Rate Examples

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 user 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.

16.4.2.1 FIFO Operation

16.4.2.1.1 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 Timeout 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 two 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 (that is, 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.

16.4.2.1.2 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.

16.4.3 Use

16.4.3.1 Base I/O Address

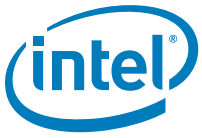
16.4.3.1.1 COM1

The base I/O address for the COM1 UART is fixed to 3F8h.

16.4.3.2 Legacy Interrupt

16.4.3.2.1 COM1

The legacy interrupt assigned to the COM1 UART is fixed to IRQ3.



16.4.4 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.

16.4.5 I/O Mapped Registers

There are 12 registers associated with the UART. These registers share eight address locations in the I/O address space. Table 16-11 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.

16.5 Register Map

Table 16-11. 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.

16.6 PCU—System Management Bus (SMBus)

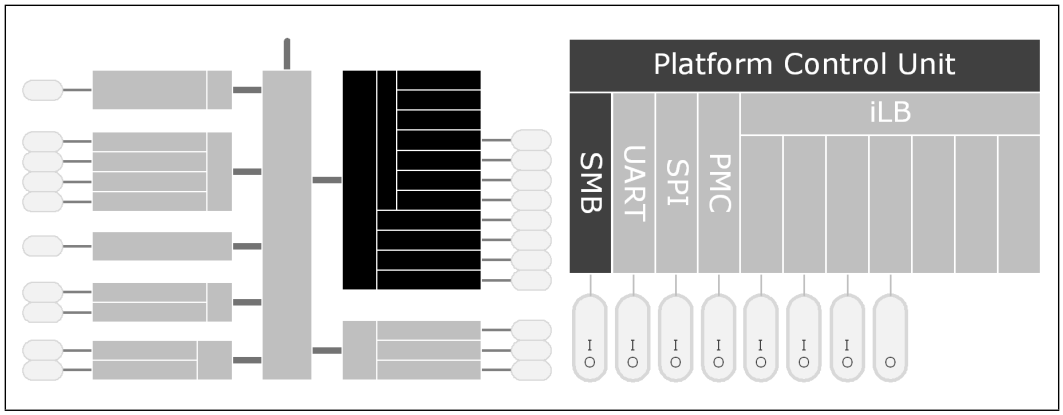
The SoC provides a System Management Bus (SMBus) 2.0 host controller. The Host controller provides a mechanism for the processor to initiate communications with SMBus peripherals (slaves). The SoC is also capable of operating in a mode in which it can communicate with I^2C compatible devices.

The SoC can perform SMBus messages with packet error checking (PEC) enabled or disabled. The actual PEC calculation and checking can be performed in either hardware or software.

The SMBus Address Resolution Protocol (ARP) is supported by using the existing host controller commands through software, except for the Host Notify command (which is actually a received message).

The programming model of the host controller is combined into two portions: a PCI configuration portion, and a system I/O mapped portion. All static configurations, such as the I/O base address, is done using the PCI configuration space. Real-time programming of the Host interface is done in system I/O space.

Figure 16-1. Platform Control Unit—System Management Bus

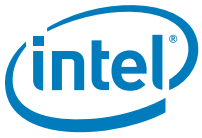


16.6.1 Signal Descriptions

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

Table 16-12. SMBus Signal Names

Signal Name	Direction Type	Description
PCU_SMB_ALERT#	I/OD CMOS1.8	**SMBus Alert** This signal is used to generate internal SMI#. This signal is multiplexed and may be used by other functions.
PCU_SMB_CLK	I/OD CMOS1.8	**SMBus Clock** External pull-up resistor is required. This signal is multiplexed and may be used by other functions.
PCU_SMB_DATA	I/OD CMOS1.8	**SMBus Data** External pull-up resistor is required. This signal is multiplexed and may be used by other functions.



16.6.2 Features

16.6.2.1 Host Controller

The SMBus host controller is used to send commands to other SMBus slave devices. Software sets up the host controller with an address, command and for writes, data and optional PEC—and then tells the controller to start. When the controller has finished transmitting data on writes, or receiving data on reads, it generates an SMI# or interrupt, if enabled.

The host controller supports 8 command protocols of the SMBus interface (see *System Management Bus (SMBus) Specification, Version 2.0*): Quick Command, Send Byte, Receive Byte, Write Byte/Word, Read Byte/Word, Process Call, Block Read/Write and Block Write–Block Read Process Call. Additionally, it supports 1 command protocol for I^2C devices—I^2C Read.

The SMBus host controller requires that the various data and command fields be setup for the type of command to be sent. When software sets the START bit, the SMBus Host controller performs the requested transaction, and interrupts the processor (or generates an SMI#) when the transaction is completed. Once a START command has been issued, the values of the "active registers" (Host Control (SMB_Mem_HCTL), Host Command (SMB_Mem_HCMD), Transmit Slave Address (SMB_Mem_TSA), Data 0 (SMB_Mem_HD0), Data 1 (SMB_Mem_HD1)) should not be changed or read until the interrupt status message (SMB_Mem_HSTS.INTR) has been set (indicating the completion of the command). Any register values needed for computation purposes should be saved prior to issuing of a new command, as the SMBus host controller updates all registers while completing the new command.

16.6.2.1.1 Command Protocols

In all of the following commands, the Host Status (SMB_Mem_HSTS) register is used to determine the progress of the command. While the command is in operation, the SMB_Mem_HSTS.HBSY bit is set. If the command completes successfully, the SMB_Mem_HSTS.INTR bit will be set. If the device does not respond with an acknowledge, and the transaction times out, the SMB_Mem_HSTS.DEVERR bit is set. If software sets the SMB_Mem_HCTL.KILL bit while the command is running, the transaction will stop and the SMB_Mem_HSTS.FAILED bit will be set.

Quick Command

When programmed for a Quick Command, the Transmit Slave Address (SMB_Mem_TSA) register is sent. The PEC byte is never appended to the Quick Protocol. Software should force the SMB_Config_HCTL.PECEN bit to 0b when performing the Quick Command. Software must force the SMB_Config_HCFG.I2C_EN bit to 0b when running this command. See section 5.5.1 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

Send Byte/Receive Byte

For the Send Byte command, the Transmit Slave Address (SMB_Mem_TSA) and Host Command (SMB_Mem_HCMD) registers are sent. For the Receive Byte command, the Transmit Slave Address (SMB_Mem_TSA) register is sent. The data received is stored in the Data 0 (SMB_Mem_HD0) register. Software must force the SMB_Config_HCFG.I2C_EN bit to 0b when running this command.

The Receive Byte is similar to a Send Byte, the only difference is the direction of data transfer. See sections 5.5.2 and 5.5.3 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

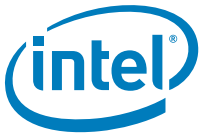
Write Byte/Word

The first byte of a Write Byte/Word access is the command code. The next 1 or 2 bytes are the data to be written. When programmed for a Write Byte/Word command, the Transmit Slave Address (SMB_Mem_TSA), Host Command (SMB_Mem_HCMD), and Data 0 (SMB_Mem_HD0) registers are sent. In addition, the Data 1 (SMB_Mem_HD1) register is sent on a Write Word command. Software must force the SMB_Config_HCFG.I2C_EN bit to 0 when running this command. See section 5.5.4 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

Read Byte/Word

Reading data is slightly more complicated than writing data. First the SoC must write a command to the slave device. Then it must follow that command with a repeated start condition to denote a read from that device's address. The slave then returns 1 or 2 bytes of data. Software must force the SMB_Config_HCFG.I2C_EN bit to 0b when running this command.

When programmed for the read byte/word command, the Transmit Slave Address (SMB_Mem_TSA) and Host Command (SMB_Mem_HCMD) registers are sent. Data is received into the Data 0 (SMB_Mem_HD0) on the read byte, and the Data 0 (SMB_Mem_HD0) and Data 1 (SMB_Mem_HD1) registers on the read word. See section 5.5.5 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.



Process Call

The process call is so named because a command sends data and waits for the slave to return a value dependent on that data. The protocol is simply a Write Word followed by a Read Word, but without a second command or stop condition.

When programmed for the Process Call command, the SoC transmits the Transmit Slave Address (SMB_Mem_TSA), Host Command (SMB_Mem_HCMD), Data 0 (SMB_Mem_HD0) and Data 1 (SMB_Mem_HD1) registers. Data received from the device is stored in the Data 0 (SMB_Mem_HD0) and Data 1 (SMB_Mem_HD1) registers. The Process Call command with SMB_Config_HCFG.I2C_EN set and the SMB_Config_HCTL.PECEN bit set produces undefined results. Software must force either SMB_Config_HCFG.I2C_EN or SMB_Config_HCTL.PECEN and SMB_Mem_AUXC.AAC to 0b when running this command. See section 5.5.6 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

Note: For process call command, the value written into SMB_Mem_TSA.RW needs to be 0b.

Note: If the SMB_Config_HCFG.I2C_EN bit is set, the protocol sequence changes slightly: the Command Code (Bits 18:11 in the bit sequence) are not sent - as a result, the slave will not acknowledge (Bit 19 in the sequence).

Block Read/Write

The SoC contains a 32-byte buffer for read and write data which can be enabled by setting SMB_Mem_AUXC.E32B, as opposed to a single byte of buffering. This 32-byte buffer is filled with write data before transmission, and filled with read data on reception. In the SoC, the interrupt is generated only after a transmission or reception of 32 bytes, or when the entire byte count has been transmitted/received.

Note: When operating in I^2C mode (SMB_Config_HCFG.I2C_EN bit is set), the SoC will never use the 32-byte buffer for any block commands.

The byte count field is transmitted but ignored by the SoC as software will end the transfer after all bytes it cares about have been sent or received.

For a Block Write, software must either force the SMB_Config_HCFG.I2C_EN bit or both the SMB_Config_HCTL.PECEN and SMB_Mem_AUXC.AAC bits to 0b when running this command.

The block write begins with a slave address and a write condition. After the command code the SoC issues a byte count describing how many more bytes will follow in the message. If a slave had 20 bytes to send, the first byte would be the number 20 (14h), followed by 20 bytes of data. The byte count may not be 0. A Block Read or Write is allowed to transfer a maximum of 32 data bytes.

When programmed for a block write command, the Transmit Slave Address (SMB_Mem_TSA), Host Command (SMB_Mem_HCMD) and Data 0 (SMB_Mem_HD0) registers are sent. Data is then sent from the Host Block Data (SMB_Mem_HBD) register; the total data sent being the value stored in the Data 0 (SMB_Mem_HD0) register. On block read commands, the first byte received is stored in the Data 0 (SMB_Mem_HD0) register, and the remaining bytes are stored in the Host Block Data (SMB_Mem_HBD) register. See section 5.5.7 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

Note: For Block Write, if the SMB_Config_HCFG.I2C_EN bit is set, the format of the command changes slightly. The SoC will still send the number of bytes (on writes) or receive the number of bytes (on reads) indicated in the Data 0 (SMB_Mem_HD0) register. However, it will not send the contents of the Data 0 (SMB_Mem_HD0) register as part of the message. Also, the Block Write protocol sequence changes slightly: the Byte Count (bits 27:20 in the bit sequence) are not sent—as a result, the slave will not acknowledge (bit 28 in the sequence).

Block Write–Block Read Process Call

The block write-block read process call is a two-part message. The call begins with a slave address and a write condition. After the command code the host issues a write byte count (M) that describes how many more bytes will be written in the first part of the message. If a master has 6 bytes to send, the byte count field will have the value 6 (0000 0110b), followed by the 6 bytes of data. The write byte count (M) cannot be 0.

The second part of the message is a block of read data beginning with a repeated start condition followed by the slave address and a Read bit. The next byte is the read byte count (N), which may differ from the write byte count (M). The read byte count (N) cannot be 0.

The combined data payload must not exceed 32 bytes. The byte length restrictions of this process call are summarized as follows:

- $M \geq 1$ byte
- $N \geq 1$ byte
- $M + N \leq 32$ bytes

The read byte count does not include the PEC byte. The PEC is computed on the total message beginning with the first slave address and using the normal PEC computational rules. It is highly recommended that a PEC byte be used with the Block Write-Block Read Process Call. Software must do a read to the Host Command (SMB_Mem_HCMD) register to reset the 32 byte buffer pointer prior to reading the Host Block Data (SMB_Mem_HBD) register.

Note: There is no STOP condition before the repeated START condition, and that a NACK signifies the end of the read transfer.

Note: The SMB_Mem_AUXC.E32B bit in the Auxiliary Control register must be set when using this protocol.

See section 5.5.8 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

I²C Read

This command allows the SoC to perform block reads to certain I²C devices, such as serial EEPROMs. The SMBus Block Read supports the 7-bit addressing mode only.

However, this does not allow access to devices using the I²C "Combined Format" that has data bytes after the address. Typically these data bytes correspond to an offset (address) within the serial memory chips.

Note: This command is supported independent of the setting of the SMB_Config_HCFG.I2C_EN bit. The I²C Read command with the SMB_Config_HCTL.PECEN bit set produces undefined results. Software must force both the SMB_Config_HCTL.PECEN and SMB_Mem_AUXC.AAC bit to 0b when running this command.

For I²C Read command, the value written into SMB_Mem_TSA.RW needs to be 1b. The format that is used for the command is shown in the following table.

Table 16-13. I²C Block Read

Bit	Description
1	Start
8:2	Slave Address – 7 bits
9	Write
10	Acknowledge from slave
18:11	Send Data 1 (SMB_Mem_HD1) register
19	Acknowledge from slave
20	Repeated Start
27:21	Slave Address – 7 bits
28	Read
29	Acknowledge from slave
37:30	Data byte 1 from slave – 8 bits
38	Acknowledge
46:39	Data byte 2 from slave – 8 bits
47	Acknowledge
–	Data bytes from slave/Acknowledge
–	Data byte N from slave – 8 bits
–	NOT Acknowledge
–	Stop

The SoC will continue reading data from the peripheral until the NAK is received.

16.6.2.2 Bus Arbitration

Several masters may attempt to get on the bus at the same time by driving the PCU_SMB_DATA line low to signal a start condition. The SoC continuously monitors the PCU_SMB_DATA line. When the SoC is attempting to drive the bus to a 1 by letting go of the PCU_SMB_DATA line, and it samples PCU_SMB_DATA low, then some other master is driving the bus and the SoC will stop transferring data.

If the SoC sees that it has lost arbitration, the condition is called a collision. The SoC will set SMB_Mem_HSTS.BERR, and if enabled, generate an interrupt or SMI#. The processor is responsible for restarting the transaction.

The SoC, as a SMBus master, drives the clock. When the SoC is sending address or command or data bytes on writes, it drives data relative to the clock it is also driving. It will not start toggling the clock until the start or stop condition meets proper setup and hold time. The SoC will also ensure minimum time between SMBus transactions as a master.

16.6.2.3 Bus Timing

16.6.2.3.1 Clock Stretching

Some devices may not be able to handle their clock toggling at the rate that the SoC as an SMBus master would like. They have the capability of stretching the low time of the clock. When the SoC attempts to release the clock (allowing the clock to go high), the clock will remain low for an extended period of time.

The SoC monitors the SMBus clock line after it releases the bus to determine whether to enable the counter for the high time of the clock. While the bus is still low, the high time counter must not be enabled. Similarly, the low period of the clock can be stretched by an SMBus master if it is not ready to send or receive data.

16.6.2.3.2 Bus Timeout (SoC as SMBus Master)

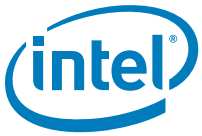
If there is an error in the transaction, such that an SMBus device does not signal an acknowledge, or holds the clock lower than the allowed time-out time, the transaction will time out. The SoC will discard the cycle and set the SMB_Mem_HSTS.DEVERR bit. The time out minimum is 25 ms (800 RTC clocks). The time-out counter inside the SoC will start after the last bit of data is transferred by the SoC and it is waiting for a response.

The 25-ms timeout counter will not count under the following conditions:

1. The SMB_Mem_HSTS.BYTE_DONE_STS bit is set
2. The TCO_STS.SECOND_TO_STS bit is not set (this indicates that the system has not locked up).

16.6.2.4 Interrupts/SMI#

The SoC SMBus controller uses INTB as its virtual interrupt wire. However, the system can alternatively be set up to generate SMI# instead of an interrupt, by setting the SMB_Config_HCFG.SMI_EN bit.



The following tables specify how the various enable bits in the SMBus function control the generation of the interrupt and Host SMI internal signals. The rows in the tables are additive, which means that if more than one row is true for a particular scenario, then the Results for all of the activated rows will occur.

Table 16-14. Enable for PCU_SMB_ALERT#

Event	SMB_Mem_ HCTL.INTREN	SMB_Config_ HCFG.SMI_EN	SMB_Mem_ SCMD.SMBALT DIS	Result
PCU_SMB_ALERT# asserted low (always reported in SMB_Mem_ HSTS.SMBALERT)	X	1	0	Slave SMI# generated (SMBUS_SMI_STS)
	1	0	0	Interrupt generated

Table 16-15. Enables for SMBus Host Events

Event	SMB_Mem_ HCTL.INTREN	SMB_Config_ HCFG.SMI_EN	Event
Any combination of SMB_Mem_HSTS. FAILED, SMB_Mem_HSTS. BERR, SMB_Mem_HSTS. DEVERR, SMB_Mem_HSTS. INTR asserted	0	X	None
	1	0	Interrupt generated
	1	1	Host SMI# generated

Table 16-16. Enables for the Host Notify Command

SMB_Mem_ SCMD.HNINTREN	SMB_Config_ HCFG.SMI_EN	SMB_Mem_ SCMD.HNWAKEEN	Result
0	X	0	None
1	0	X	Interrupt generated
1	1	X	Slave SMI# generated (SMBUS_SMI_STS)

16.6.2.5 PCU_SMB_ALERT#

PCU_SMB_ALERT# is multiplexed with UART0_TXD. When enabled and the signal is asserted, the SoC can generate an interrupt or an SMI#.

Note: Using this signal as a wake event from S4/S5 is not supported.

16.6.2.6 SMBus CRC Generation and Checking

If the SMB_Mem_AUXC.AAC is set, the SoC automatically calculates and drives CRC at the end of the transmitted packet for write cycles, and will check the CRC for read cycles. It will not transmit the contents of the Packet Error Check Data Register (SMB_Mem_PEC) PEC register for CRC. The SMB_Mem_HCTL.PECEN bit must not be set if this bit is set, or unspecified behavior will result.

If the read cycle results in a CRC error, the SMB_Mem_HSTS.DEVERR bit and the SMB_Mem_AUXS.CRCE bit will be set.

16.6.2.7 SMBus Slave Interface

The SoC does not implement a complete SMBus slave interface. Only the Host Notify Command is implemented to maintain specification compatibility.

16.6.2.7.1 Format of Host Notify Command

The SoC tracks and responds to the standard Host Notify command as specified in the System Management Bus (SMBus) Specification, Version 2.0. The host address for this command is fixed to 0001000b. If the SoC already has data for a previously-received host notify command that has not been serviced yet by the host software (as indicated by the SMB_Mem_SSTS.HNST bit), then it will NACK following the host address byte of the protocol. This allows the host to communicate non-acceptance to the master and retain the host notify address and data values for the previous cycle until host software completely services the interrupt.

Note: Host software must always clear the SMB_Mem_SSTS.HNST bit after completing any necessary reads of the address and data registers.

The following table shows the Host Notify format.

Table 16-17. Host Notify Format (Sheet 1 of 2)

Bit	Description	Driven By	Comment
1	Start	External Master	
2:8	SMB Host Address – 7 bits	External Master	Always 0001_000
9	Write	External Master	Always 0
10	ACK (or NACK)	SoC	SoC NACKs if SMB_Mem_SSTS.HNST is 1
11:17	Device Address – 7 bits	External Master	Indicates the address of the master; loaded into the Notify Device Address Register (SMB_Mem_NDA)
18	Unused – Always 0	External Master	7-bit-only address; this bit is inserted to complete the byte
19	ACK	SoC	
22:27	Data Byte Low – 8 bits	External Master	Loaded into the Notify Data Low Byte Register (SMB_Mem_NDLB)

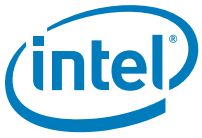


Table 16-17. Host Notify Format (Sheet 2 of 2)

Bit	Description	Driven By	Comment
28	ACK	SoC	
29:36	Data Byte High – 8 bits	External Master	Loaded into the Notify Data High Byte Register (SMB_Mem_NDHB)
37	ACK	SoC	
38	Stop	External Master	

16.6.2.8 Function Disable

The SMBus interface may be disable by setting FUNC_DIS_2.SMB_DIS to 1b.

16.6.3 References

System Management Bus (SMBus) Specification Version 2.0: http://www.smbus.org/specs/

16.7 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 Interrupt Decoding and Routing"
- "PCU—iLB I/O 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).

16.7.1 Signal Descriptions

Table 16-18. iLB Signals

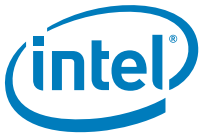
Signal Name	Direction/ Type	Description
NMI#	I/ GPIO	Non-Maskable Interrupt: This is an NMI event indication into the SoC. This signal is multiplexed and may be used by other functions.

16.7.2 Features

16.7.2.1 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
  - Legacy interrupt support
  - 15 total interrupts through two cascaded controllers
  - I/O mapped registers
- I/O Advanced Programmable Interrupt Controller
  - Legacy-free interrupt support
  - 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)
  - Legacy-free timer support
  - Three timers and one counter
  - Memory mapped registers
- Real-Time Clock (RTC)
  - 242 byte RAM backed by battery (Also Known As, CMOS RAM)
  - Can generate wake/interrupt when time matches programmed value
  - I/O and indexed registers

16.7.2.2 Non-Maskable Interrupt

NMI support is enabled by setting the NMI Enable (NMI_EN) bit, at I/O Port 70h, Bit 7, to 1b.

Non-Maskable Interrupts (NMIs) can be generated by several sources, as described in Table 16-19.

Table 16-19. 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	
ILB_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	

Note: The NSC register is documented in the PCU_ILB 8254 Timers Memory Mapped I/O Registers Section.

16.8 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.

16.8.1 Signal Descriptions

Table 16-20. LPC Signals

Signal Name	Direction/ Type	Description
LPC_AD[3:0]	I/O HSHV 3.3/1.8	**LPC Multiplexed Command, Address, Data:** Internal pull-ups are provided for these signals. These signals are multiplexed and may be used by other functions. ***Note:*** To set LPC_AD[3:0] power supply to 1.8V, set V1P8 mode in family configuration, trigger a RCOMP cycle using family RCOMP registers and lastly, copy RCOMP value to family p and n strength values.
LPC_CLK[0]	O HSHV 3.3/1.8	**LPC Clock [0] Out:** 25 MHz PCI-like clock driven to LPC peripherals. These signals are multiplexed and may be used by other functions.
LPC_CLK[1]	O HSHV 3.3/1.8	**LPC Clock [1] Out:** 25 MHz PCI-like clock driven to LPC peripherals. Can be configured as an input to compensate for board routing delays through SoftStrap. These signals are multiplexed and may be used by other functions.
LPC_CLKRUN_N	I/O HSHV 3.3/1.8	**LPC Clock Run:** Input to determine the status of ILB_LPC_CLK and an open drain output used to request starting or speeding up ILB_LPC_CLK. This is a sustained tri-state signal used by the central resource to request permission to stop or slow ILB_LPC_CLK. The central resource is responsible for maintaining the signal in the asserted state when ILB_LPC_CLK is running and de-asserts the signal to request permission to stop or slow ILB_LPC_CLK. An internal pull-up is provided for this signal. This signal is multiplexed and may be used by other functions.
LPC_FRAME_N	I/O HSHV 3.3/1.8	**LPC Frame:** This signal indicates the start of an LPC cycle, or an abort. This signal is multiplexed and may be used by other functions.
LPC_SERIRQ	I/O MSMV 1.8	**Serial Interrupt Request:** This signal implements the serial interrupt protocol. This signal is multiplexed and may be used by other functions.
LPC_RCOMP	I CMOS3.3/1.8	Compensation Resistor.

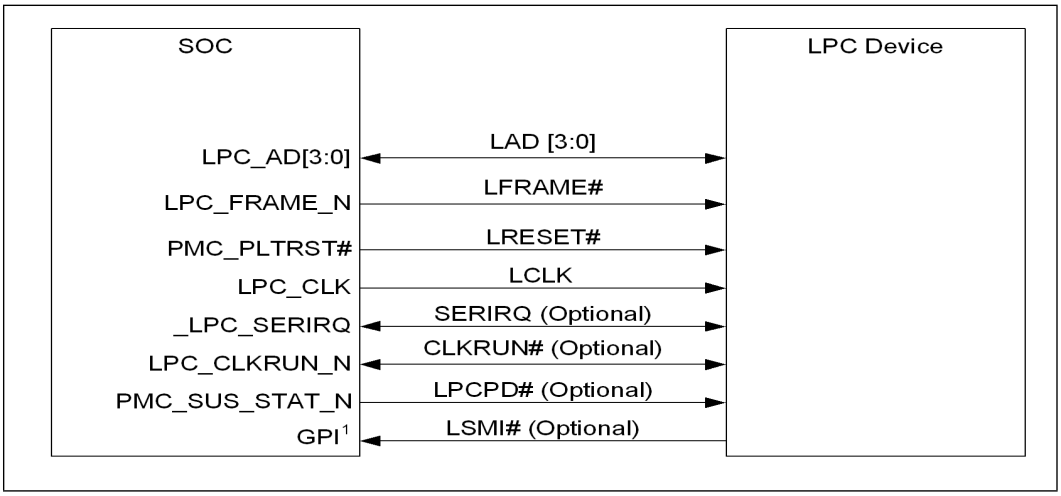
16.8.2 Features

The LPC interface to the SoC is shown in Figure 16-2.

Note: 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 16-2. LPC Interface Diagram



Note: The General Purpose Input (GPI) must use a SMI capable.

16.8.2.1 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 1s to the processor.

16.8.2.2 Trusted Platform Module (TPM) 1.2 Support

The LPC interface supports accessing Trusted Platform Module (TPM) 1.2 devices by means of 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.

16.8.2.3 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.

Note: No validation has been done with BIOS on LPC. Reference design uses SPI.

16.8.2.4 Subtractive Decode

All cycles that are not decoded internally, and are not targeted for LPC (that is, configuration cycles, I/O cycles above 64KB and memory cycles above 16MB), will be sent to LPC with ILB_LPC_FRAME# not asserted.

16.8.2.5 POST Code Redirection

Writes to addresses 80h - 8Fh in I/O register space will also be passed to the LPC bus.

Note: Reads of these addresses do not result in any LPC transactions.

16.8.2.6 Power Management

16.8.2.6.1 LPCPD# Protocol

Same timings as for PMC_SUS_STAT#. After driving PMC_SUS_STAT# active, the SoC drives ILB_LPC_FRAME# low, and tri-states (or drives low) ILB_LPC_AD[3:0].

Note: The Low Pin Count Interface Specification, Revision 1.1 defines the LPCPD# protocol where there is at least 30 µs from LPCPD# assertion to LRST# 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# (connects to LPCPD#) and ILB_PLTRST# (connects to LRST#) at the same time during a global reset. This is not inconsistent with the LPC LPCPD# protocol.

16.8.2.6.2 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 ILB_LPC_CLKRUN# signal. LPC devices that require the clock to stay running should drive ILB_LPC_CLKRUN# 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 ILB_LPC_CLKRUN#, the SoC will start the LPC clock and assert ILB_LPC_CLKRUN#.

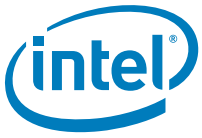
Note: The CLKRUN protocol is disabled by default. See Section 16.8.3.2.2, "Clock Run Enable" on page 162 for further details.

16.8.2.7 Serialized IRQ (SERIRQ)

16.8.2.7.1 Overview

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, ILB_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 - Turnaround Phase:** Signal released



The interrupt controller supports 21 serial interrupts. These represent the 15 ISA interrupts (IRQ 0-1, 3-15), the four PCI interrupts, and the control signals SMI# and IOCHK#. Serial interrupt information is transferred using three types of frames:

- Start Frame: ILB_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: ILB_LPC_SERIRQ line driven low by the interrupt controller to indicate end of transmission and next mode of operation.

16.8.2.7.2 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 by means of 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 ILB_LPC_SERIRQ line remains inactive and pulled up between the Stop and Start Frame until a peripheral drives ILB_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 ILB_LPC_SERIRQ low for 1 LPC clock less than in continuous mode. This mode of operation allows for lower power operation.

16.8.2.7.3 Data Frames

Once the Start frame has been initiated, the ILB_LPC_SERIRQ peripherals start counting frames based on the rising edge of ILB_LPC_SERIRQ. Each of the IRQ/DATA frames has exactly 3 phases of 1 clock each:

- **Sample Phase—**During this phase, a device drives ILB_LPC_SERIRQ low if its corresponding interrupt signal is low. If its corresponding interrupt is high, then the ILB_LPC_SERIRQ devices tri-state ILB_LPC_SERIRQ. ILB_LPC_SERIRQ remains high due to pull-up resistors.
- **Recovery Phase—**During this phase, a device drives ILB_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 ILB_LPC_SERIRQ.

16.8.2.7.4 Stop Frame

After the data frames, a Stop Frame will be driven by the interrupt controller. ILB_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 16-21.

Table 16-21. 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

16.8.2.7.5 Serial Interrupts Not Supported

There are three (3) 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.

16.8.2.7.6 Data Frame Format and Issues

Table 16-22 shows the format of the data frames. The decoded INT[A:D]# values are ANDed with the corresponding PCI-express input signals (PIRQ[A:D]#). This way, the interrupt can be shared.

The other interrupts decoded by means of 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 16-22. SERIRQ Interrupt Mapping (Sheet 1 of 2)

Data Frame Number	Interrupt	Clocks Past Start Frame	Comment
1	IRQ0	2	Ignored—Can only be generated by means of the internal 8524
2	IRQ1	5	Before port 60h latch
3	SMI#	8	Causes SMI# if low. Sets SMI_STS.ILB_SMI_STS register bit.
4	IRQ3	11	
5	IRQ4	14	
6	IRQ5	17	
7	IRQ6	20	
8	IRQ7	23	
9	IRQ8	26	Ignored—IRQ8# 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#	50	Same as ISA IOCHCK# going active.

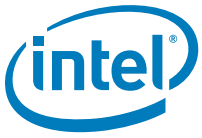


Table 16-22. SERIRQ Interrupt Mapping (Sheet 2 of 2)

Data Frame Number	Interrupt	Clocks Past Start Frame	Comment
18	PCI INTA#	53	
19	PCI INTB#	56	
20	PCI INTC#	59	
21	PCI INTD#	62	

16.8.3 Use

16.8.3.1 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, ILB_LPC_CLK[0] is looped back on itself on the SoC pad.
   a. Benefit:

   ILB_LPC_CLK[0] and ILB_LPC_CLK[1] are both available for system clocking

   b. Drawback:

   Clock delay compensation is less effective at compensating for main board delay

   c. Soft Strap and Register Requirements:

   Soft Strap LPCCLK_SLC = 0b

   Configuration is reflected by register bit LPCC.LPCCLK_SLC=0b

   Soft Strap LPCCLK1_ENB = 0b (ILB_LPC_CLK[1] disabled) or 1b (ILB_LPC_CLK[1] enabled)
2. Configuration is reflected by register bit LPCC.LPCCLK1EN=0b (ILB_LPC_CLK[1] disabled) or 1b (ILB_LPC_CLK[1] enabled)
3. On the main board: In this configuration, ILB_LPC_CLK[0] is looped back to ILB_LPC_CLK[1] on the main board.
   a. Benefit:

   Clock delay compensating in more effective at compensating for main board delay

   b. Drawback:

   Only ILB_LPC_CLK[0] is available for system clocking. ILB_LPC_CLK[1] must be disabled.

   c. Soft Strap and Register Requirements:

   Soft Strap LPCCLK_SLC = 1b

   Configuration is reflected by register bit LPCC.LPCCLK_SLC=1b

   Soft Strap LPCCLK1_ENB = 0b (ILB_LPC_CLK[1] disabled)

   Configuration is reflected by register bit LPCC.LPCCLK1EN=0b

16.8.3.2 LPC Power Management

16.8.3.2.1 Clock Enabling

The LPC clocks can be enabled or disabled by setting or clearing, respectively, the LPCC.LPCCLK[1:0]EN bits.

16.8.3.2.2 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.

16.8.3.3 SERIRQ Disable

Serialized IRQ support may be disabled by setting the OIC.SIRQEN bit to 0b.

16.8.4 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

16.9 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 3.3V 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.

16.9.1 Signal Descriptions

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

Table 16-23. RTC Signals (Sheet 1 of 2)

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.

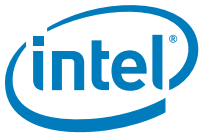


Table 16-23. RTC Signals (Sheet 2 of 2)

Signal Name	Direction/ Type	Description
RTC_RST#	I CMOS V3P3	**RTC Reset** RTC Reset: An external RC circuit creates a time delay for the signal such that it will go high (de-assert) sometime after the battery voltage is valid. The RC time delay should be in the 10-20 ms range. Contact your Intel representative for details. When asserted, this signal resets all register bits in the RTC well except GEN_PMCON1.RPS. ***Note:*** GEN_PMCON1.RPS will only be set when RTEST# is asserted (low). ***Notes:*** Unless registers are 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. ***Notes:*** In the case where the RTC battery is dead or missing on the platform, the signal should be deasserted before the PMC_RSMRST# signal is deasserted.
RTEST#	I CMOS V3P3	**RTC Battery Test** RTC Battery Test: An external RC circuit creates a time delay for the signal such that it will go high (de-assert) sometime after the battery voltage is valid. The RC time delay should be in the 10-20 ms range. Contact your Intel representative for details. If the battery is missing/ weak, this signal appears low (asserted) at boot just after the suspend power rail (V3P3A) is up since it will not have time to meet Vih when V3P3A is high. The weak/ missing battery condition is reported in the GEN_PMCON1.RPS (RTC Power Status) register. When asserted, BIOS may clear the RTC CMOS RAM. ***Note:*** RSM_RST# signal needs to toggle in order for bit status to propagate and reflect in the GEN_PMCON registers ***Notes:*** Unless CMOS is being cleared (only to be done in the G3 power state) or the battery is low, the signal input must always be high when all other RTC power planes are on. ***Notes:*** This signal may also be used for debug purposes, as part of a XDP port. Contact your Intel representative for details.
COREPOWER	I CMOS V3P3	**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#. ***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 TPVPERL PCI Express* 2.0 specification on PMC_PLTRST# de-assertion. ***Note:*** PMC_CORE_PWROK must not glitch, even if PMC_RSMRST# is low.
RSM_RST#	I CMOS V3P3	**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.

16.9.2 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 122–500 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.

16.9.2.1 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.

16.9.3 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.

16.9.3.1 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.

16.9.3.2 Clearing Battery-Backed RTC RAM

Clearing CMOS RAM in an SoC-based platform can be done by using a jumper on ILB_RTC_RST# or a GPI. Implementations should not attempt to clear CMOS by using a jumper to pull RTC_VCC low.

16.9.3.2.1 Using RTC_RST# to Clear CMOS

A jumper on RTC_RST# 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# is strapped to ground, all the bits of GEN_PMCON1 (except GEN_PMCON1.RPS - which is only set by assertion of R_TEST# assertion) 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# to be pulled up through a weak pull-up resistor. Table 16-24 shows which bits are set to their default state when RTC_RST# is asserted. This RTC_RST# jumper technique allows the jumper to be moved and then replaced while the system is powered off.

Note: RSM_RST# signal needs to toggle in order for bit status to propagate and reflect in the GEN_PMCON registers.

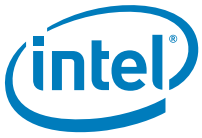


Table 16-24. Register Bits Reset by RTC_RST# Assertion

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.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

Note: RTC_RST# will not set GEN_PMCON1.RPS. This bit is only set when R_TEST# is asserted low. When the system is rebooted, GEN_PMCON1.RPS bit can be detected in the set state.

16.9.3.3 Using a 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.

16.9.4 References

Accessing the Real Time Clock Registers and the NMI Enable Bit: http://download.intel.com/design/intarch/PAPERS/321088.pdf

16.9.5 I/O 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 processor 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 16-25.

Table 16-25. 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

16.9.6 Indexed Registers

The RTC contains indexed registers that are accessed by means of the reg_RTC_IR_type and reg_RTC_TR_type registers.

Table 16-26. RTC Indexed Registers

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
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

16.10 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.

16.10.1 Signal Descriptions

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

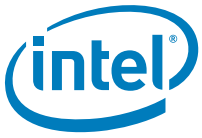


Table 16-27. 8254 Signals

Signal Name	Direction/ Type	Description
ILB_8254_SPKR	O	**Speaker** The signal drives an external speaker driver device, which in turn drives the system speaker. Upon PMC_PLTRST#, its output state is 0. This signal is multiplexed and may be used by other functions.

16.10.2 Features

16.10.2.1 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.

16.10.2.2 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.

16.10.2.3 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.

16.10.3 Use

16.10.3.1 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 16-28 lists the six, operating modes for the interval counters.

Table 16-28. 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.
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, and so forth.
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.

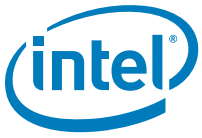
16.10.3.2 Reading From the Interval Timer

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.

16.10.3.2.1 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.

16.10.3.2.2 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.

16.10.3.2.3 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.

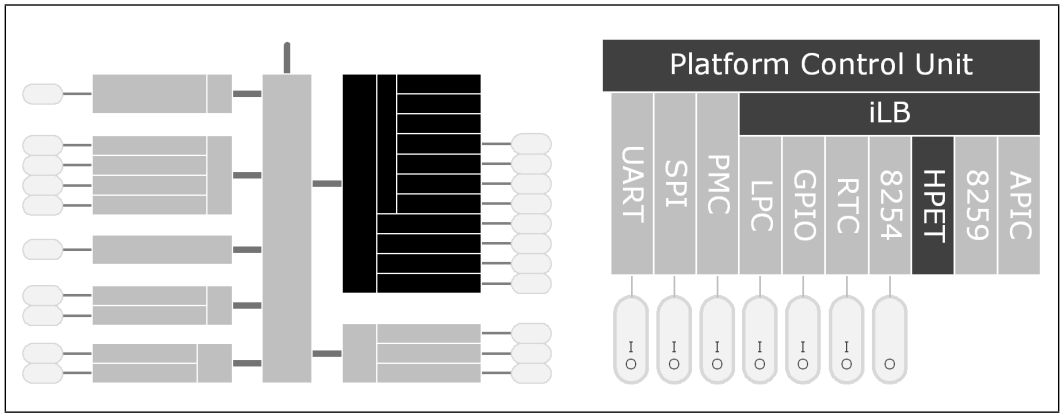
If both count and status of a counter are latched, the first read operation from that counter returns the latched status, regardless of which was latched first. The next one or two reads, depending on whether the counter is programmed for one or two type counts, returns the latched count. Subsequent reads return unlatched count.

16.11 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.

Note: HPET operation is not guaranteed when LPC clock is running at 25MHz. Designs needing LPC to be at 25 MHz should use an alternate timer like PM_TIMER, RTC, or LAPIC timer for their application.

Figure 16-3. Platform Control Unit—High Precision Event Timer (HPET)



16.11.1 Features

16.11.1.1 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.

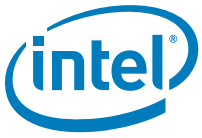
T0CV cannot be programmed reliably by a single 64-bit write in a 32-bit environment unless only the periodic rate is being changed. If T0CV needs to be re-initialized, the following algorithm is performed:

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.

16.11.1.2 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.

16.11.1.2.1 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.

16.11.1.2.2 Mapping Option Number 1—Legacy Option (GCFG.LRE Set)

This forces the following mapping:

Table 16-29. 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	

16.11.1.2.3 Mapping Option Number 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.

16.11.2 References

IA-PC HPET (High Precision Event Timers) Specification, Revision 1.0a: http://www.intel.com/hardware design/hpetspec_1.pd

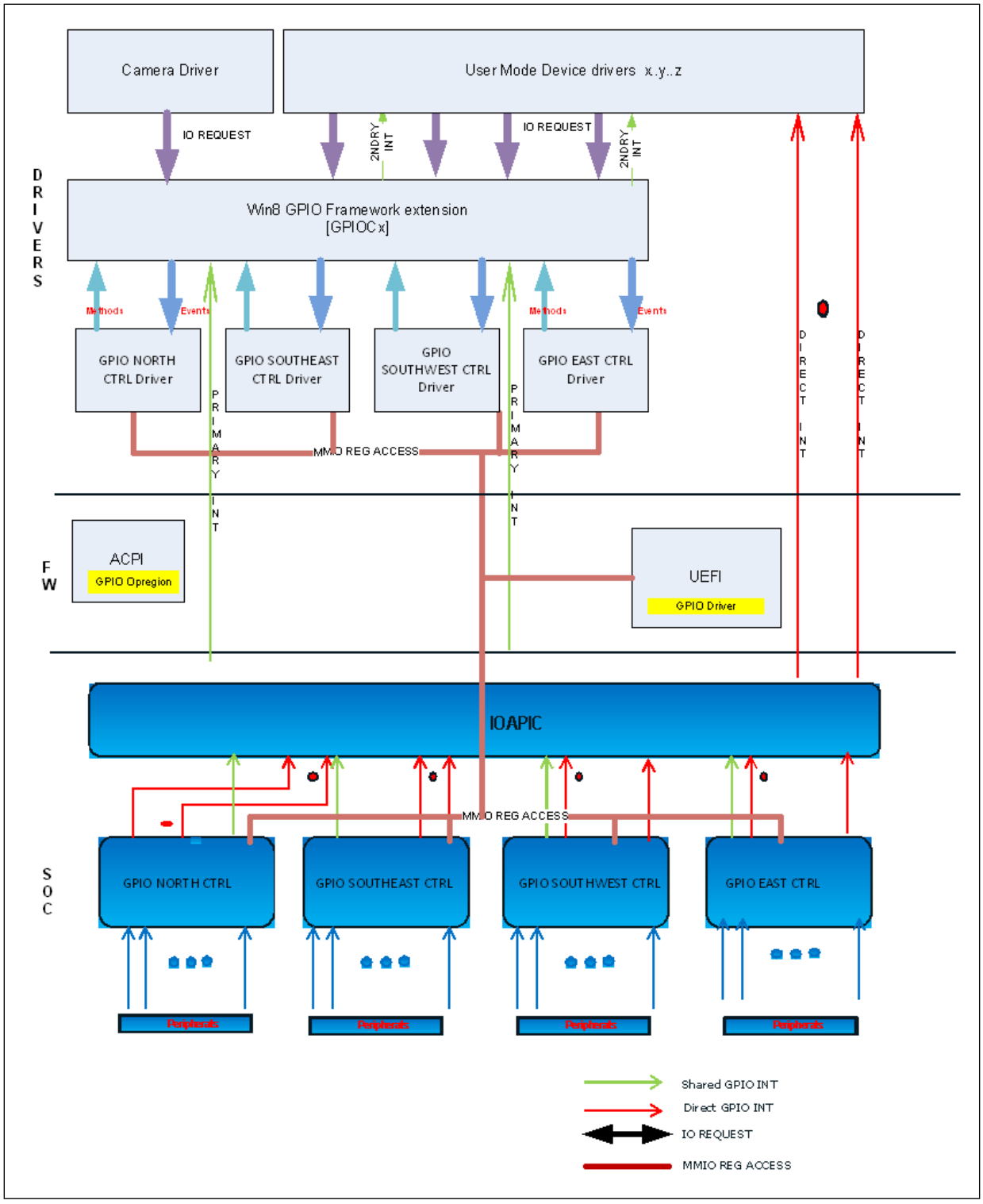
16.11.3 Memory Mapped Registers

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.

16.12 PCU—iLB GPIO

The processor SoC contains 4 GPIO controllers that interact with the operating system by means of the BIOS/ACPI Firmware. A GPIO Controller driver and MSFT GPIO framework provides the GPIO I/O services, interrupt services and event handling mechanism.

Figure 16-4. GPIO Stack Block Diagram



16.12.1 Signal Description

See 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 is in Chapter 21, "Electrical Specifications"
- Description: A brief explanation of the signal function

The processor SoC has a maximum of 198 GPIOs and these GPIOs are divided into 4 communities. GPIOs are multiplexed with other alternate modes.

16.12.2 GPIO Controller

The GPIO controllers handle all GPIO/CFIO interface to the processor SoC.

- GPIO NORTH—used for Camera sensors, DFX, SVID, and as Display Pins.
- GPIO SOUTHEAST—Defines the Pads/Pins for MMC Storage/SD host controllers for storage and COMMS, LPC Pins, Fast SPI Pins, and Platform clock
- GPIO SOUTHWEST—Defines the Pads/Pins for HS UART, I2C HS, PCIe*, SMBus, and SPI Pins
- GPIO EAST—Defines the Pads SoC power state related signals of PMU and ISH Pins

Note: ISH (Integrated Sensor Hub) is not supported for the processor; however, the GPIOs can be used for other functionality.

16.12.3 Use

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.

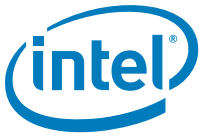
The I/O Select register determines the direction of the GPIO.

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



16.12.4 GPIO Registers

16.12.4.1 Memory Space Address Mapping

All addresses in Table 16-30 are offsets from the CFIO memory space base address, also known as, IOBASE. Base address for CFIO memory space registers (IOBASE) are located in D31:F0:0x5B. Each GPIO Community has 16-bit addressing with a possible address space of 64kb.

Table 16-30. Generic Community Address Ranges

Offset	Name	Access Control Group
0x0000	Access Control Policy Registers	Group0
0x0200	GPIO Controller Wake Logic Registers	Group1
0x0300	GPIO Controller IMG Registers	Group2
0x0400	Community Registers	Group3
0x0500	PWM Registers	Group4
0x600-0x62F	DLL Control Registers (Southeast Only)	
0x0600 - 0x0FFF	Reserved area for expansion	None
0x1000	Family Broadcast Write Registers	Group5
0x1080	Family0 Registers	Group6
0x1100	Family1 Registers	
0x4000	Pad Broadcast Write Registers	Group7
0x4400	Family0 Pad Registers	Group8
0x4800	Family1 Pad Registers	Group9

16.12.5 Register Address Mapping

The following table describes registers description and address offset according to pad number and its CFIO controller.

Table 16-31. Register Address Mapping (Sheet 1 of 2)

Address	Default	Name	Description
0x0000	0xFFFF_FFFF	gpio_read_access_policy_access_reg	GPIO Read Access Control
0x0100	0xFFFF_FFFF	gpio_write_access_policy_access_reg	GPIO Write Access Control
0x0200	0x0000_0000	gpio_wake_status_reg_0	GPIO_WAKE_STATUS_REG #0
0x0280	0x0000_0000	gpio_wake_mask_reg_0	GPIO_WAKE_MASK_REG #0
0x0300	0x0000_0000	gpio_interrupt_status	GPIO Interrupt Status
0x0380	0x0000_0000	gpio_interrupt_mask	GPIO Interrupt Status
0x1000	0x0000_0000	gpio_family_bw_mask_31_0	GPIO Broadcast Data Mask bit Register
0x1004	0x0000_0000	gpio_family_bw_data_31_0	GPIO Broadcast Data Register
0x1008	0x0000_0000	gpio_family_broadcast_reg_mask_0	gpio_family_broadcast_reg_mask #0
0x1080 +0x20*family#	Varies	<family_name>_family_rcomp_control_reg	RCOMP Control Register

Table 16-31. Register Address Mapping (Sheet 2 of 2)

Address	Default	Name	Description
0x1084 +0x20*family#	Varies	<family_name>_family_rcomp_offset_reg	RCOMP Offset Register
0x1088 +0x20*family#	Varies	<family_name>_family_rcomp_override_reg	RCOMP Override Register
0x108C +0x20*family#	Varies	<family_name>_family_rcomp_value_reg	RCOMP Value Register
0x1090 +0x20*family#	Varies	<family_name>_family_config_rcomp_reg	family_config_rcomp_reg
0x1094 +0x20*family#	Varies	<family_name>_family_config_reg	gpio_family_configuration_register
0x4000	0x0000_0000	gpio_pad_bw_mask_31_0	GPIO Broadcast Data Mask bit Register
0x4004	0x0000_0000	gpio_pad_bw_data_31_0	GPIO Broadcast Data Register
0x4008	0x0000_0000	gpio_pad_broadcast_reg_mask_0	gpio_pad_broadcast_reg_mask #0
0x4400 +0x400*family# +0x8*pad#		<pad_name>_PAD_CFG0	Pad Control Register 0
0x4404 +0x400*family# +0x8*pad#		<pad_name>_PAD_CFG1	Pad Control Register 1

16.12.6 Hard Strap Logic

Hard straps are used to change settings during boot prior to any on die-firmware or BIOS execution. Hard straps also change settings prior to any flash reads, unlike the soft straps which reside in the flash data.

While 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.

All Straps Sampled at Posedge of RSMRST_N. After the straps are sampled the pins can be used functionally as pin changes will no longer affect the strapped values.

Refer to Chapter 2, "Physical Interfaces" for GPIO strap pin list.

16.13 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 I/O 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 I/O APIC".

16.13.1 Features

16.13.1.1 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

16.13.1.1.1 For Consumption by the 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.

16.13.1.1.2 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.

16.13.1.2 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 I/O-APIC, depending on the configuration of the 8 PIRQx Routing Control Registers: PIRQA, PIQRB, PIRQC, PIRQD, PIRQE, PIRQF, PIRQG, PIRQH.

16.13.1.2.1 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, and 14–15. The assignment is programmable through the 8 PIRQx Routing Control Registers: PIRQA, PIQRB, PIRQC, PIRQD, PIRQE, PIRQF, PIRQG, and 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.

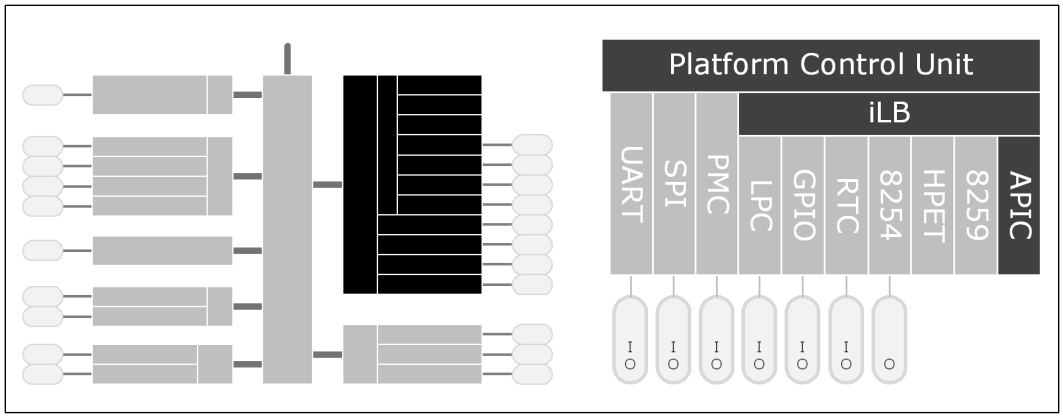
The PIRQx# lines are defined as active low, level sensitive. When a PIRQx# is routed to specified IRQ line, software must change the IRQs corresponding ELCR bit to level sensitive mode. The SoC internally inverts the PIRQx# line to send an active high level to the PIC. When a PCI interrupt is routed 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.

16.14 PCU—iLB I/O APIC

The I/O 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, by means of 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.

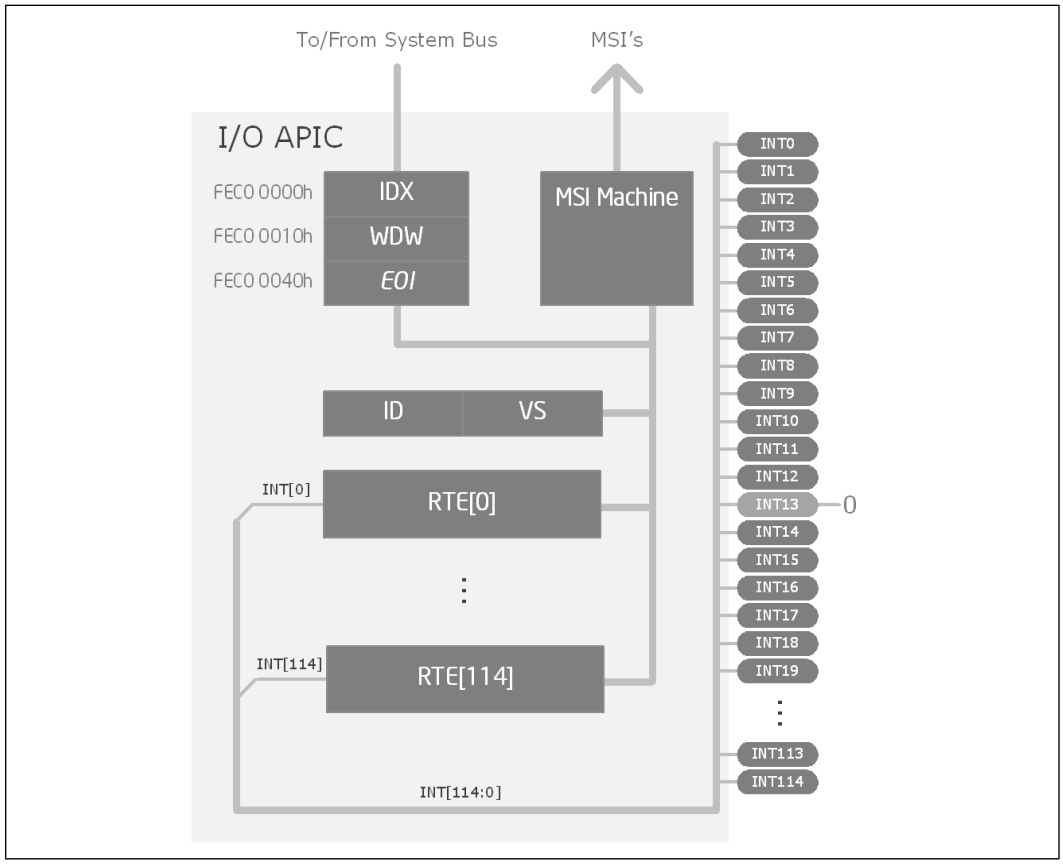
Figure 16-5. Platform Control Unit—APIC



16.14.1 Features

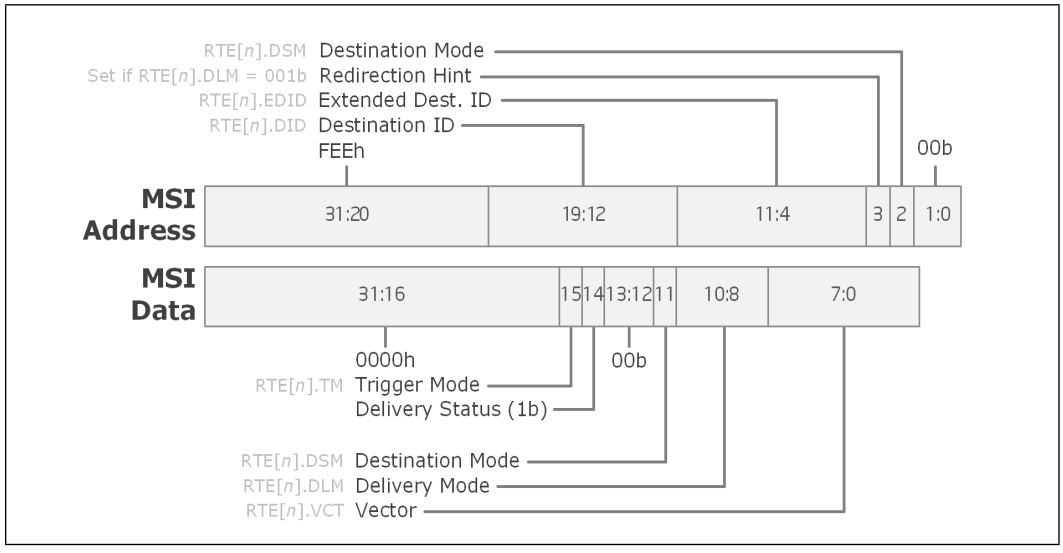
- 115 interrupt lines
  - IRQ0-114
- Edge or level trigger mode per interrupt
- Active low or high polarity per interrupt
- Works with local APIC in processor by means of MSIs
- MSIs can target specific processor core
- Established APIC programming model

Figure 16-6. 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 shown in the following figure.

Figure 16-7. MSI Address and Data



Destination ID (DID) and Extended Destination ID (EDID) are used to target a specific processor core's local APIC.

16.14.2 Use

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.

16.14.3 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.

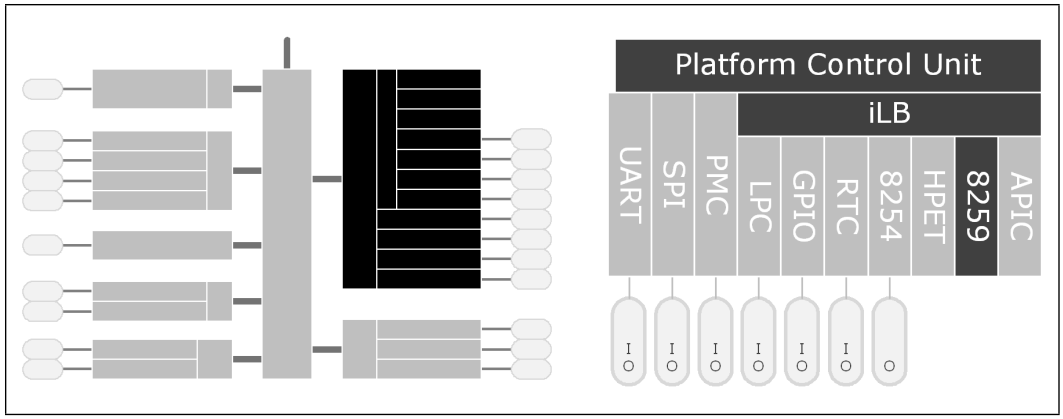
Note: There is one pair of redirection (RTE) registers per interrupt line. Each pair forms a 64-bit RTE register.

Note: Specified offsets should be placed in IDX, not added to IDX.

16.15 PCU—iLB 8259 Programmable Interrupt Controllers (PIC)

The SoC provides an ISA-compatible programmable interrupt controller (PIC) that incorporates the functionality of two, cascaded 8259 interrupt controllers.

Figure 16-8. Platform Control Unit—8259 Programmable Interrupt Controllers



16.15.1 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 16-32 shows how the controllers are connected.

Note: The SoC does not implement any external PIRQ# signals. The PIRQs referred to in this chapter originate from the interrupt routing unit.

Table 16-32. Interrupt Controller Connections (Sheet 1 of 2)

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 by means of SERIRQ or PIRQx
	4	IRQ4 by means of SERIRQ or PIRQx or PCU UART1
	5	IRQ5 by means of SERIRQ or PIRQx
	6	IRQ6 by means of SERIRQ or PIRQx
	7	IRQ7 by means of SERIRQ or PIRQx

Table 16-32. Interrupt Controller Connections (Sheet 2 of 2)

8259	8259 Input	Connected Pin/Function
Slave	0	Inverted IRQ8# from internal RTC or HPET
	1	IRQ9 by means of SERIRQ, SCI or PIRQx
	2	IRQ10 by means of SERIRQ, SCI or PIRQx
	3	IRQ11 by means of SERIRQ, SCI, HPET or PIRQx
	4	IRQ12 by means of SERIRQ, PIRQx or mouse emulation
	5	None
	6	PIRQx or IRQ14 from SATA Controller
	7	IRQ15 by means of SERIRQ or PIRQx or IRQ15 from SATA Controller

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#.

16.15.1.1 Interrupt Handling

16.15.1.1.1 Generating Interrupts

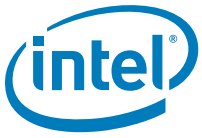
The PIC interrupt sequence involves three bits, from the IRR, ISR, and IMR, for each interrupt level. These bits are used to determine the interrupt vector returned, and status of any other pending interrupts. Table 16-33 defines the IRR, ISR, and IMR.

Table 16-33. 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.

16.15.1.1.2 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 16-34. 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

16.15.1.1.3 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.

16.15.1.2 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.

16.15.1.2.1 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.

16.15.1.2.2 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.

16.15.1.2.3 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.

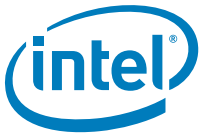
16.15.1.2.4 ICW4

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.

16.15.1.3 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.



16.15.1.4 Modes of Operation

16.15.1.4.1 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.

16.15.1.4.2 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.

16.15.1.4.3 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).

16.15.1.4.4 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.

16.15.1.4.5 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.

16.15.1.4.6 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.

16.15.1.5 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.

16.15.1.5.1 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.

16.15.1.5.2 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.

16.15.1.6 Masking Interrupts

16.15.1.6.1 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.

16.15.1.6.2 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 and OCW3.SMM=1b, and cleared where OCW3.ESMM=1b and OCW3.SMM=0b.

16.15.2 I/O Mapped Registers

The interrupt controller registers are located at 20h and 21h for the master controller (IRQ 0 - 7), and at A0h and A1h for the slave controller (IRQ 8 - 13). These registers have multiple functions, depending upon the data written to them. Table 16-35 is a description of the different register possibilities for each address.

Note: The register descriptions after Table 16-35 represent one register possibility.

Table 16-35. 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
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

§ §

17 Serial ATA (SATA)

17.1 Functional Feature Descriptions

Feature	Description
Native Command Queuing (NCQ)	Allows the device to reorder commands for more efficient data transfers
Auto Activate for DMA	Collapses a DMA Setup then DMA Activate sequence into a DMA Setup only
Hot-plug Support	Allows for device detection without power being applied and ability to connect and disconnect devices without prior notification to the system
Asynchronous Signal Recovery	Provides a recovery from a loss of signal or establishing communication after hot-plug
6Gb/s Transfer Rate	Capable of data transfers up to 6Gb/s with Gen 3 SATA ***Note:*** Gen 1 and Gen 2 support different transfer rates
ATAPI Asynchronous Notification	A mechanism for a device to send a notification to the host that the device requires attention
Host and Link Initiated Power Management	Capability for the host controller or device to request Partial and Slumber interface power states
Staggered Spin-Up	Enables the host to spin up hard drives sequentially to prevent power load problems on boot
DEVSLP	Device Sleep (DEVSLP) is a host-controlled hardware signal which enables a SATA host and device to enter an ultra-low interface power state.

17.2 Signal Descriptions

Table 17-1. Signals Description

Signal Name	Direction/Reference	Description
SATA_GP[3:0]	I/O V1P8S	**Serial ATA Port [3:0] General Purpose:** This is an input pin which can be configured as an interlock switch or as a general purpose I/O, depending on the platform. When used as an interlock switch status indication, this signal should be driven to '0' to indicate that the switch is closed, and to '1' to indicate that the switch is open. **Note:** • SATA_GP[0] is multiplexed with ISH_GPIO_12. • SATA_GP[1] is multiplexed with SPI3_CS0# • SATA_GP[2] is multiplexed with SATA_DEVSLP[0]. • SATA_GP[3] is multiplexed with SATA_DEVSLP[1]
SATA_LED#	O V1P8S	**Serial ATA LED:** This is an open-collector output pin driven during SATA command activity. It is to be connected to external circuitry that can provide the current to drive a platform LED. When active, the LED is on. When tristated, the LED is off.
SATA_TXP[1:0] **SATA_TXN[1:0]**	O V1P05A	**Serial ATA Port 1 and 0:** These are outbound high-speed differential signals to Port 1 and 0.
SATA_RXP[1:0] **SATA_RXN[1:0]**	I V1P05A	**Serial ATA Port 1 and 0:** These are inbound high-speed differential signals to Port 1 and 0.
SATA_RCOMP_P and SATA_RCOMP_N	O V1P8S	**Serial ATA Impedance Compensation**: These pins are used to connect the external resistors used for RCOMP.

17.3 Features

17.3.1 Supported Features

Table 17-2. SATA/AHCI Feature Matrix

Feature	AHCI Enabled
Native Command Queuing (NCQ)	Supported
Auto Activate for DMA	Supported
Hot-plug Support	Supported
Asynchronous Signal Recovery	Supported
3Gb/s Transfer Rate	Supported
ATAPI Asynchronous Notification	Supported
Host and Link Initiated Power Management	Supported
Staggered Spin-Up	Supported
Command Completion Coalescing	N/A

17.3.2 Features Not Supported

- Port Multiplier
- FIS Based Switching
- Command Based Switching
- IDE Mode
- Cold Presence Detect
- Function Level Reset (FLR)
- Command Completion Coalescing
- Enclosure Management

17.4 References

- Serial ATA Specification rev 3.1
- Serial ATA Advanced Host Controller Interface (AHCI) Specification rev 1.3.1
- Serial ATA II: Extensions to Serial ATA 1.0 Specification, Revision 1.0

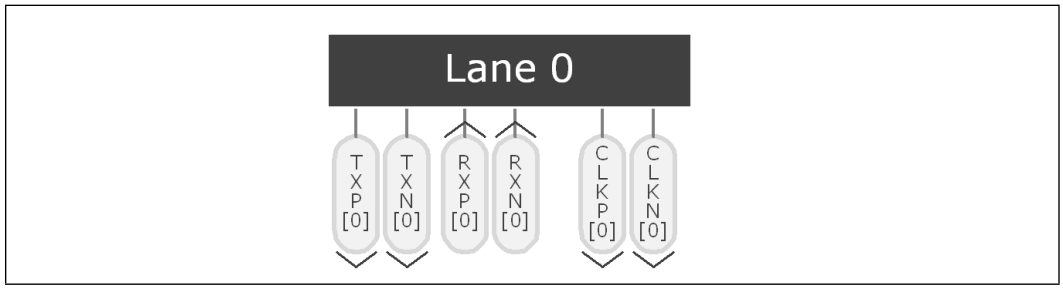
§ §

18 PCI Express* 2.0

There are four lanes and up to four 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.

18.1 Signal Descriptions

Figure 18-1. PCI Express* 2.0 Lane 0 Signal Example



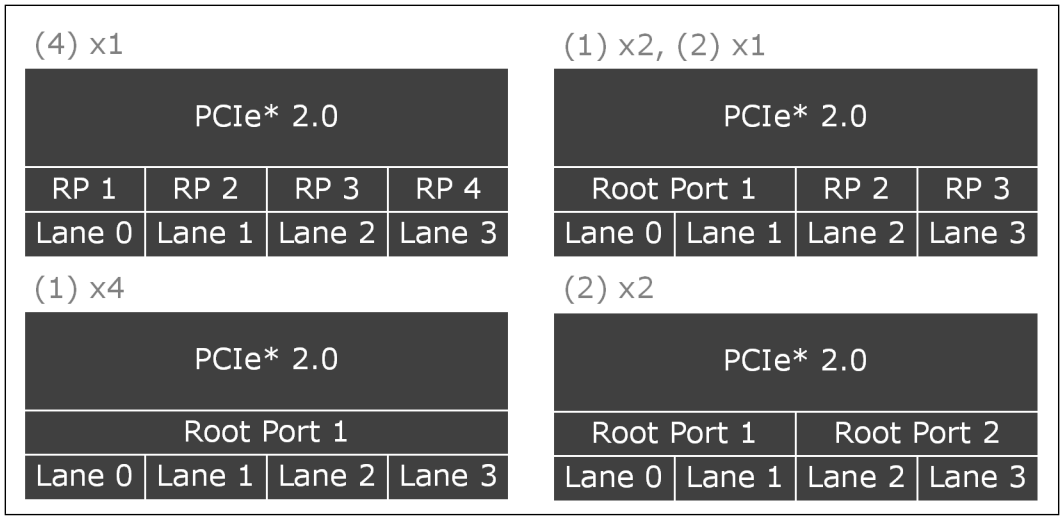
18.2 Features

- 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, and x4 link widths (auto negotiated)
- Flexible Root Port (1-24) configuration options
  - (4) x1s
  - (2) x2's
  - (1) x2 plus (2) x1s
  - (1) x4
- 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 L0s, L1 and L2
  - Powered down in ACPI S3 state - L3

18.2.1 Root Port Configurations

Depending on SKU, there are up to four possible lane assignments for root ports 1-4.

Figure 18-2. Root Port Configuration Options



Root port configurations are set by SoftStraps stored in SPI flash, and the default option is "(4) 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 or x4.

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

18.2.2 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. MSIs are automatically passed upstream by the root port, just as other memory writes would be. INTx messages are delivered to the Legacy Block interrupt router/controller by the root port.

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

Table 18-1. Supported Interrupts Generated From Events/Packets (Sheet 1 of 2)

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			

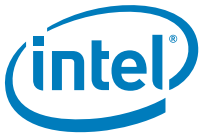


Table 18-1. Supported Interrupts Generated From Events/Packets (Sheet 2 of 2)

Packet/Event	Type	INTx	MSI	SERR	SCI	SMI	GPE
ERR_NONFATAL	Packet			X			
ERR_FATAL	Packet			X			
Internal Error	Event			X			
VDM	Packet						X

Note: Table 18-1 lists the supported 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 18-2. 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#
Root Port 3	INTC#	INTD#	INTA#	INTB#
Root Port 4	INTB#	INTC#	INTD#	INTA#

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

18.2.2.1 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.

18.2.2.2 System Error (SERR)

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

18.2.3 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.

18.3 References

PCI Express Base Specification*, Rev. 2.0

§ §

19 Ball Map, Ball Out, and SoC Pin Locations

Figure 19-1. Ball Map (Top Left View—Columns 53–29)

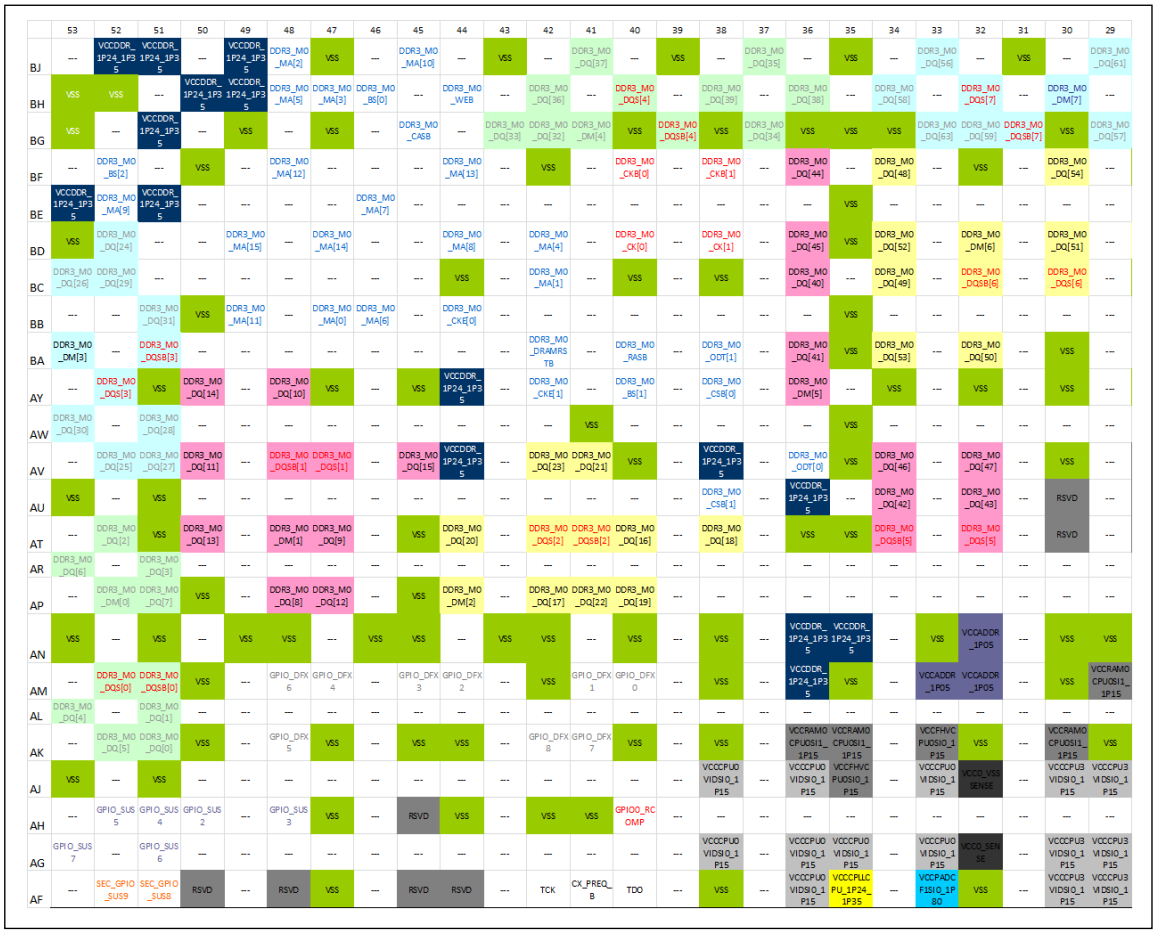


Figure 19-2. Ball Map—DDR3L (Top Right View—Columns 28 – 4)

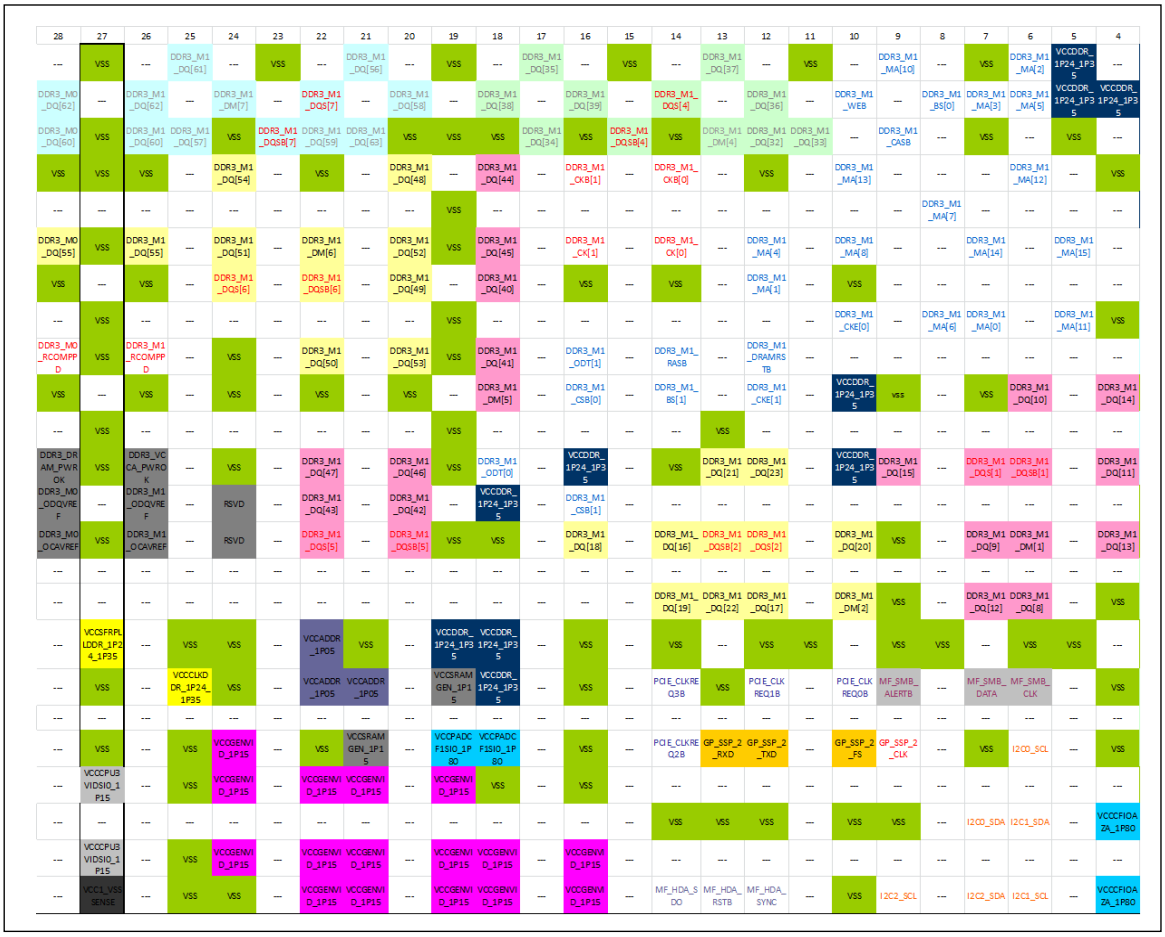
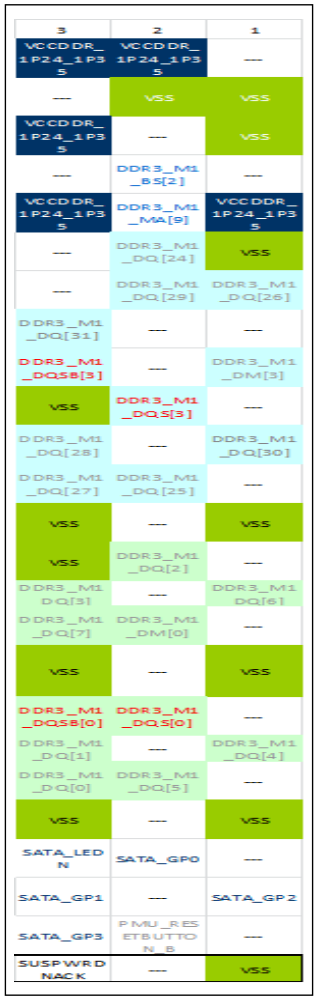


Figure 19-3. Ball Map—DDR3L (Top Right View Columns 3–1)

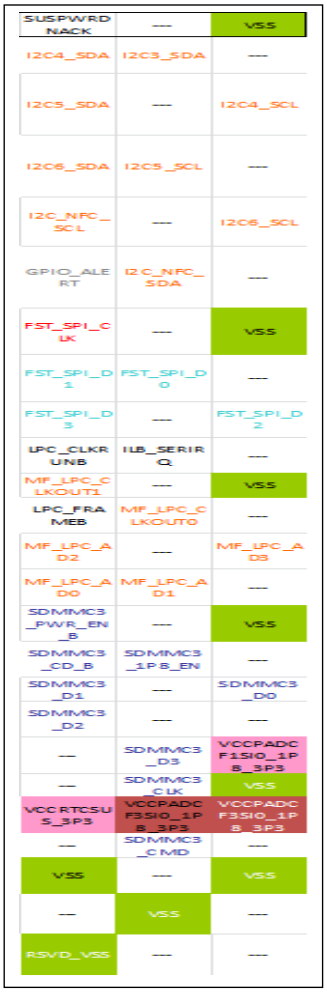


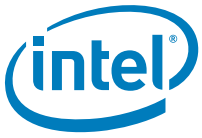
3	2	1
VCCDDR_1P24_1P35	VCCDDR_1P24_1P35	---
---	VSS	VSS
VCCDDR_1P24_1P35	---	VSS
---	DDR3_M1_B5[2]	---
VCCDDR_1P24_1P35	DDR3_M1_MA[9]	VCCDDR_1P24_1P35
---	DDR3_M1_DQ[24]	VSS
---	DDR3_M1_DQ[29]	DDR3_M1_DQ[26]
DDR3_M1_DQ[31]	---	---
DDR3_M1_DQSB[3]	---	DDR3_M1_DM[3]
VSS	DDR3_M1_DQS[3]	---
DDR3_M1_DQ[28]	---	DDR3_M1_DQ[30]
DDR3_M1_DQ[27]	DDR3_M1_DQ[25]	---
VSS	---	VSS
VSS	DDR3_M1_DQ[2]	---
DDR3_M1_DQ[3]	---	DDR3_M1_DQ[6]
DDR3_M1_DQ[7]	DDR3_M1_DM[0]	---
VSS	---	VSS
DDR3_M1_DQSB[0]	DDR3_M1_DQS[0]	---
DDR3_M1_DQ[1]	---	DDR3_M1_DQ[4]
DDR3_M1_DQ[0]	DDR3_M1_DQ[5]	---
VSS	---	VSS
SATA_LEDN	SATA_GP0	---
SATA_GP1	---	SATA_GP2
SATA_GP3	PMU_RESETBUTTON_B	---
SUSPWRDNACK	---	VSS

Figure 19-4. Ball Map—DDR3L (Bottom Left View—Columns 53–29)

Figure 19-5. Ball Map—DDR3L (Bottom Right View—Columns 28 – 4)

Figure 19-6. Ball Map—DDR3L (Bottom Right View Columns 3–1)





19.1 SoC Pin List Locations

Table 19-1. SoC Pin List Locations (Sheet 1 of 11)

Pin #	Pin Name	Pin #	Pin Name	Pin #	Pin Name
M24	VSSA	Y30	VCCSRAMSOCIUN_1P05	T44	MCSI_1_CLKP
A5	VSS_NCTF	Y32	VCCSRAMSOCIUN_1P05	T45	MCSI_1_CLKN
A6	VSS_NCTF	Y33	VCCSRAMSOCIUN_1P05	AF48	RSVD
B2	VSS_NCTF	Y35	VCCSRAMSOCIUN_1P05	AF50	RSVD
B52	VSS_NCTF	AK21	VCCSRAMGEN_1P15	AF45	RSVD
BG1	VSS_NCTF	AM19	VCCSRAMGEN_1P15	AF44	RSVD
BG53	VSS_NCTF	AA18	UNCORE_VNN_S4	T4	LPC_HVT_RCOMP
BH1	VSS_NCTF	AA19	UNCORE_VNN_S4	P3	LPC_FRAME_N
BH2	VSS_NCTF	AA21	UNCORE_VNN_S4	T3	LPC_CLKRUN_N
BH52	VSS_NCTF	AA22	UNCORE_VNN_S4	G18	RSVD_VSS
BH53	VSS_NCTF	AA24	UNCORE_VNN_S4	T2	ILB_SERIRQ
C1	VSS_NCTF	AA25	UNCORE_VNN_S4	N20	ICLKRCOMP
F1	VSS_NCTF	AC18	UNCORE_VNN_S4	P20	ICLKICOMP
A11	VSS	AC19	UNCORE_VNN_S4	AB3	I2C6_SDA
A15	VSS	AC21	UNCORE_VNN_S4	AA1	I2C6_SCL
A19	VSS	AC22	UNCORE_VNN_S4	AC3	I2C5_SDA
A23	VSS	AC24	UNCORE_VNN_S4	AB2	I2C5_SCL
A31	VSS	AC25	UNCORE_VNN_S4	AD3	I2C4_SDA
A39	VSS	AD25	UNCORE_VNN_S4	AC1	I2C4_SCL
A43	VSS	AD27	UNCORE_VNN_S4	AD2	I2C3_SDA
A47	VSS	H44	USB_VDDQ_G3	AE4	I2C3_SCL
A7	VSS	AN27	DDRSFR_VDDQ_G_S4	AF7	I2C2_SDA
AA16	VSS	Y25	ICLK_VSFR_G3	AF9	I2C2_SCL
AA27	VSS	Y27	ICLK_VSFR_G3	AH6	I2C1_SDA
AA38	VSS	D4	RTC_V3P3A_G5	AF6	I2C1_SCL
AA53	VSS	E3	RTC_V3P3A_G5	AH7	I2C0_SDA
AB10	VSS	B6	RTC_V3P3RTC_G5	AK6	I2C0_SCL
AB12	VSS	C5	RTC_V3P3RTC_G5	Y2	RSVD
AB13	VSS	AK30	CORE_V1P15	AA3	RSVD
AB14	VSS	AK35	CORE_V1P15	U51	HV_DDI2_HPD
AB4	VSS	AK36	CORE_V1P15	T52	HV_DDI2_DDC_SDA
AB42	VSS	AM29	CORE_V1P15	T51	HV_DDI2_DDC_SCL
AB47	VSS	AC30	CORE_VSFR_G3	R51	HV_DDI1_HPD
AB50	VSS	P38	CORE_VSFR_G3	W51	HV_DDI0_HPD

Table 19-1. SoC Pin List Locations (Sheet 2 of 11)

Pin #	Pin Name	Pin #	Pin Name	Pin #	Pin Name
AB6	VSS	V30	CORE_VSFR_G3	Y52	HV_DDI0_DDC_SDA
AC16	VSS	P41	USBSSIC_V1P2A_G3	Y51	HV_DDI0_DDC_SCL
AC29	VSS	V29	USBSSIC_V1P05A_G3	AH45	RSVD
AC33	VSS	M41	USBHSIC_V1P2A_G3	P26	RSVD
AC35	VSS	E1	SDIO_V3P3A_V1P8A_G3	AH40	GPIO0_RCOMP
AC38	VSS	E2	SDIO_V3P3A_V1P8A_G3	AG53	GPIO_SUS7
AD21	VSS	Y18	GPIO_V1P8A_G3	AG51	GPIO_SUS6
AD30	VSS	AD33	GPIO_V1P8A_G3	AH52	GPIO_SUS5
AD32	VSS	AF33	GPIO_V1P8A_G3	AH51	GPIO_SUS4
AD36	VSS	AK18	GPIO_V1P8A_G3	AH48	GPIO_SUS3
AD4	VSS	AK19	GPIO_V1P8A_G3	AH50	GPIO_SUS2
AD44	VSS	G1	SDIO_V3P3A_V1P8A_G3	AD52	GPIO_SUS1
AE1	VSS	AD16	DDI_VGG	AD51	GPIO_SUS0
AE11	VSS	AD18	DDI_VGG	AK42	GPIO_DFX8
AE12	VSS	AD19	DDI_VGG	AK41	GPIO_DFX7
AE14	VSS	AF16	DDI_VGG	AM48	GPIO_DFX6
AE40	VSS	AF18	DDI_VGG	AK48	GPIO_DFX5
AE42	VSS	AF19	DDI_VGG	AM47	GPIO_DFX4
AE43	VSS	AF21	DDI_VGG	AM45	GPIO_DFX3
AE45	VSS	AF22	DDI_VGG	AM44	GPIO_DFX2
AE46	VSS	AG16	DDI_VGG	AM41	GPIO_DFX1
AE48	VSS	AG18	DDI_VGG	AM40	GPIO_DFX0
AE50	VSS	AG19	DDI_VGG	Y3	GPIO_ALERT
AE53	VSS	AG21	DDI_VGG	AK12	GP_SSP_2_TXD
AE6	VSS	AG22	DDI_VGG	AK13	GP_SSP_2_RXD
AE8	VSS	AG24	DDI_VGG	AK10	GP_SSP_2_FS
AE9	VSS	AJ19	DDI_VGG	AK9	GP_SSP_2_CLK
AF10	VSS	AJ21	DDI_VGG	V40	GP_CAMERASB11
AF24	VSS	AJ22	DDI_VGG	Y41	GP_CAMERASB10
AF25	VSS	AJ24	DDI_VGG	Y42	GP_CAMERASB09
AF32	VSS	AK24	DDI_VGG	Y44	GP_CAMERASB08
AF38	VSS	AJ35	FUSE_V1P15	AB40	GP_CAMERASB07
AF47	VSS	AK33	FUSE_V1P15	AA51	GP_CAMERASB06
AG25	VSS	G10	FUSE0_V1P05A_G3	AB52	GP_CAMERASB05
AH10	VSS	H10	FUSE1_V1P05A_G4	AB51	GP_CAMERASB04
AH12	VSS	N18	FUSE3_V1P05A_G5	AC53	GP_CAMERASB03

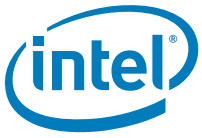


Table 19-1. SoC Pin List Locations (Sheet 3 of 11)

Pin #	Pin Name	Pin #	Pin Name	Pin #	Pin Name
AH13	VSS	P40	MIPI_V1P2A_G3	AB44	GP_CAMERASB02
AH14	VSS	T40	MIPI_V1P2A_G3	AB45	GP_CAMERASB01
AH41	VSS	V18	ICLK_GND_OFF	AB41	GP_CAMERASB00
AH42	VSS	V19	ICLK_GND_OFF	K20	RSVD
AH44	VSS	BE1	DDR_VDDQ_G_S4	M20	RSVD
AH47	VSS	BE53	DDR_VDDQ_G_S4	A3	RSVD
AH9	VSS	BJ2	DDR_VDDQ_G_S4	U3	FST_SPI_D3
AJ1	VSS	BJ3	DDR_VDDQ_G_S4	U1	FST_SPI_D2
AJ16	VSS	BJ49	DDR_VDDQ_G_S4	V3	FST_SPI_D1
AJ18	VSS	BJ5	DDR_VDDQ_G_S4	V2	FST_SPI_D0
AJ25	VSS	BJ51	DDR_VDDQ_G_S4	V7	FST_SPI_CS2_N
AJ3	VSS	BJ52	DDR_VDDQ_G_S4	V6	FST_SPI_CS1_N
AJ51	VSS	AM18	DDR_VDDQ_G_S4	V4	FST_SPI_CS0_N
AJ53	VSS	AM36	DDR_VDDQ_G_S4	W3	FST_SPI_CLK
AK16	VSS	AN18	DDR_VDDQ_G_S4	M26	RSVD
AK22	VSS	AN19	DDR_VDDQ_G_S4	M13	RSVD
AK25	VSS	AN35	DDR_VDDQ_G_S4	AV26	DDR3_VCCA_PWROK
AK27	VSS	AN36	DDR_VDDQ_G_S4	BH10	DDR3_M1_WE_N
AK29	VSS	AU18	DDR_VDDQ_G_S4	BA26	DDR3_M1_RCOMP
AK32	VSS	AU36	DDR_VDDQ_G_S4	BA14	DDR3_M1_RAS_N
AK38	VSS	AV10	DDR_VDDQ_G_S4	BA16	DDR3_M1_ODT[1]
AK4	VSS	AV16	DDR_VDDQ_G_S4	AV18	DDR3_M1_ODT[0]
AK40	VSS	AV38	DDR_VDDQ_G_S4	AU26	DDR3_M1_ODQVREF
AK44	VSS	AV44	DDR_VDDQ_G_S4	AT26	DDR3_M1_OCAVREF
AK45	VSS	AY10	DDR_VDDQ_G_S4	AT24	RSVD
AK47	VSS	AY44	DDR_VDDQ_G_S4	AU24	RSVD
AK50	VSS	BE3	DDR_VDDQ_G_S4	BE2	DDR3_M1_MA[9]
AK7	VSS	BE51	DDR_VDDQ_G_S4	BD10	DDR3_M1_MA[8]
AM13	VSS	BG3	DDR_VDDQ_G_S4	BE8	DDR3_M1_MA[7]
AM16	VSS	BG51	DDR_VDDQ_G_S4	BB8	DDR3_M1_MA[6]
AM24	VSS	BH4	DDR_VDDQ_G_S4	BH6	DDR3_M1_MA[5]
AM27	VSS	BH49	DDR_VDDQ_G_S4	BD12	DDR3_M1_MA[4]
AM30	VSS	BH5	DDR_VDDQ_G_S4	BH7	DDR3_M1_MA[3]
AM35	VSS	BH50	DDR_VDDQ_G_S4	BJ6	DDR3_M1_MA[2]
AM38	VSS	AF29	CORE_VCC1	BD5	DDR3_M1_MA[15]
AM4	VSS	AF30	CORE_VCC1	BD7	DDR3_M1_MA[14]
AM42	VSS	AG27	CORE_VCC1	BF10	DDR3_M1_MA[13]
AM50	VSS	AG29	CORE_VCC1	BF6	DDR3_M1_MA[12]
AN1	VSS	AG30	CORE_VCC1	BB5	DDR3_M1_MA[11]

Table 19-1. SoC Pin List Locations (Sheet 4 of 11)

Pin #	Pin Name	Pin #	Pin Name	Pin #	Pin Name
AN11	VSS	AJ27	CORE_VCC1	BJ9	DDR3_M1_MA[10]
AN12	VSS	AJ29	CORE_VCC1	BC12	DDR3_M1_MA[1]
AN14	VSS	AJ30	CORE_VCC1	BB7	DDR3_M1_MA[0]
AN16	VSS	AF36	CORE_VCC1	BA12	DDR3_M1_DRAMRST_N
AN21	VSS	AG33	CORE_VCC1	BG23	DDR3_M1_DQSB[7]
AN24	VSS	AG35	CORE_VCC1	BC22	DDR3_M1_DQSB[6]
AN25	VSS	AG36	CORE_VCC1	AT20	DDR3_M1_DQSB[5]
AN29	VSS	AG38	CORE_VCC1	BG15	DDR3_M1_DQSB[4]
AN3	VSS	AJ33	CORE_VCC1	BA3	DDR3_M1_DQSB[3]
AN30	VSS	AJ36	CORE_VCC1	AT13	DDR3_M1_DQSB[2]
AN33	VSS	AJ38	CORE_VCC1	AV6	DDR3_M1_DQSB[1]
AN38	VSS	AC36	CORE_VSFR_G3	AM3	DDR3_M1_DQSB[0]
AN40	VSS	AD35	CORE_VSFR_G3	BH22	DDR3_M1_DQS[7]
AN42	VSS	AD38	CORE_VSFR_G3	BC24	DDR3_M1_DQS[6]
AN43	VSS	AF35	CORE_VSFR_G3	AT22	DDR3_M1_DQS[5]
AN45	VSS	AM25	DDR_VDDQ_G_S4	BH14	DDR3_M1_DQS[4]
AN46	VSS	AF4	VCCCFIOAZA_1P80	AY2	DDR3_M1_DQS[3]
AN48	VSS	AH4	VCCCFIOAZA_1P80	AT12	DDR3_M1_DQS[2]
AN49	VSS	U27	USB3_V1P05A_G3	AV7	DDR3_M1_DQS[1]
AN5	VSS	V27	USB3_V1P05A_G3	AM2	DDR3_M1_DQS[0]
AN51	VSS	U22	SATA_V1P05A_G3	AT7	DDR3_M1_DQ[9]
AN53	VSS	U24	SATA_V1P05A_G3	AP6	DDR3_M1_DQ[8]
AN6	VSS	V22	PCIE_V1P05A_G3	AP3	DDR3_M1_DQ[7]
AN8	VSS	V24	PCIE_V1P05A_G3	BG21	DDR3_M1_DQ[63]
AN9	VSS	AM21	DDR_V1P05A_G3	BH26	DDR3_M1_DQ[62]
AP4	VSS	AM22	DDR_V1P05A_G3	BJ25	DDR3_M1_DQ[61]
AP45	VSS	AM32	DDR_V1P05A_G3	BG26	DDR3_M1_DQ[60]
AP50	VSS	AM33	DDR_V1P05A_G3	AR1	DDR3_M1_DQ[6]
AP9	VSS	AN22	DDR_V1P05A_G3	BG22	DDR3_M1_DQ[59]
AT18	VSS	AN32	DDR_V1P05A_G3	BH20	DDR3_M1_DQ[58]
AT19	VSS	V36	DDI_VDDQ_G3	BG25	DDR3_M1_DQ[57]
AT27	VSS	Y36	DDI_VDDQ_G3	BJ21	DDR3_M1_DQ[56]
AT3	VSS	U16	FUSE_V1P8A_G3	BD26	DDR3_M1_DQ[55]
AT35	VSS	U19	FUSE_V1P05A_G3	BF24	DDR3_M1_DQ[54]
AT36	VSS	AF27	CORE_VSS1_SENSE	BA20	DDR3_M1_DQ[53]

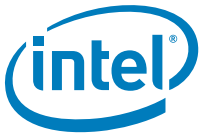


Table 19-1. SoC Pin List Locations (Sheet 5 of 11)

Pin #	Pin Name	Pin #	Pin Name	Pin #	Pin Name
AT45	VSS	AD29	CORE_VCC1_SENSE	BD20	DDR3_M1_DQ[52]
AT51	VSS	AJ32	CORE_VSS0_SENSE	BD24	DDR3_M1_DQ[51]
AT9	VSS	AG32	CORE_VCC0_SENSE	BA22	DDR3_M1_DQ[50]
AU1	VSS	C35	USB3_TXP3	AK2	DDR3_M1_DQ[5]
AU3	VSS	C34	USB3_TXP2	BC20	DDR3_M1_DQ[49]
AU51	VSS	A33	USB3_TXP1	BF20	DDR3_M1_DQ[48]
AU53	VSS	B32	USB3_TXP0	AV22	DDR3_M1_DQ[47]
AV14	VSS	A35	USB3_TXN3	AV20	DDR3_M1_DQ[46]
AV19	VSS	B34	USB3_TXN2	BD18	DDR3_M1_DQ[45]
AV24	VSS	C33	USB3_TXN1	BF18	DDR3_M1_DQ[44]
AV27	VSS	C32	USB3_TXN0	AU22	DDR3_M1_DQ[43]
AV30	VSS	G34	USB3_RXP3	AU20	DDR3_M1_DQ[42]
AV35	VSS	G32	USB3_RXP2	BA18	DDR3_M1_DQ[41]
AV40	VSS	F30	USB3_RXP1	BC18	DDR3_M1_DQ[40]
AW13	VSS	F28	USB3_RXP0	AL1	DDR3_M1_DQ[4]
AW19	VSS	J34	USB3_RXN3	BH16	DDR3_M1_DQ[39]
AW27	VSS	J32	USB3_RXN2	BH18	DDR3_M1_DQ[38]
AW35	VSS	D30	USB3_RXN1	BJ13	DDR3_M1_DQ[37]
AW41	VSS	D28	USB3_RXN0	BH12	DDR3_M1_DQ[36]
AY20	VSS	D34	USB3_RCOMP_P	BJ17	DDR3_M1_DQ[35]
AY22	VSS	F34	USB3_RCOMP_N	BG17	DDR3_M1_DQ[34]
AY24	VSS	B47	USB_VBUSSNS	BG11	DDR3_M1_DQ[33]
AY26	VSS	C37	RSVD	BG12	DDR3_M1_DQ[32]
AY28	VSS	A37	RSVD	BB3	DDR3_M1_DQ[31]
AY3	VSS	F36	RSVD	AW1	DDR3_M1_DQ[30]
AY30	VSS	D36	RSVD	AR3	DDR3_M1_DQ[3]
AY32	VSS	M34	RSVD	BC2	DDR3_M1_DQ[29]
AY34	VSS	M32	RSVD	AW3	DDR3_M1_DQ[28]
AY45	VSS	N34	RSVD	AV3	DDR3_M1_DQ[27]
AY47	VSS	P34	RSVD	BC1	DDR3_M1_DQ[26]
AY51	VSS	A48	USB_RCOMP	AV2	DDR3_M1_DQ[25]
AY7	VSS	B48	USB_OTG_ID	BD2	DDR3_M1_DQ[24]
AY9	vss	P16	USB_OC1_N	AV12	DDR3_M1_DQ[23]
B28	VSS	P14	USB_OC0_N	AP13	DDR3_M1_DQ[22]
B36	VSS	B46	RSVD	AV13	DDR3_M1_DQ[21]
BA19	VSS	N38	USB_HSIC_RCOMP	AT10	DDR3_M1_DQ[20]
BA24	VSS	K38	USB_HSIC_1_STROBE	AT2	DDR3_M1_DQ[2]
BA27	VSS	M38	USB_HSIC_1_DATA	AP14	DDR3_M1_DQ[19]

Table 19-1. SoC Pin List Locations (Sheet 6 of 11)

Pin #	Pin Name	Pin #	Pin Name	Pin #	Pin Name
BA30	VSS	M36	USB_HSIC_0_STROBE	AT16	DDR3_M1_DQ[18]
BA35	VSS	N36	USB_HSIC_0_DATA	AP12	DDR3_M1_DQ[17]
BB19	VSS	B40	USB_DP4	AT14	DDR3_M1_DQ[16]
BB27	VSS	C45	USB_DP3	AV9	DDR3_M1_DQ[15]
BB35	VSS	C41	USB_DP2	AY4	DDR3_M1_DQ[14]
BB4	VSS	C43	USB_DP1	AT4	DDR3_M1_DQ[13]
BB50	VSS	C42	USB_DP0	AP7	DDR3_M1_DQ[12]
BC10	VSS	C40	USB_DN4	AV4	DDR3_M1_DQ[11]
BC14	VSS	A45	USB_DN3	AY6	DDR3_M1_DQ[10]
BC16	VSS	A41	USB_DN2	AL3	DDR3_M1_DQ[1]
BC26	VSS	B44	USB_DN1	AK3	DDR3_M1_DQ[0]
BC28	VSS	B42	USB_DN0	BH24	DDR3_M1_DM[7]
BC38	VSS	C38	RSVD	BD22	DDR3_M1_DM[6]
BC40	VSS	B38	RSVD	AY18	DDR3_M1_DM[5]
BC44	VSS	G36	RSVD	BG13	DDR3_M1_DM[4]
BD1	VSS	J36	RSVD	BA1	DDR3_M1_DM[3]
BD19	VSS	Y6	UART2_TXD	AP10	DDR3_M1_DM[2]
BD27	VSS	Y7	UART2_RXD	AT6	DDR3_M1_DM[1]
BD35	VSS	V10	UART2_RTS_N	AP2	DDR3_M1_DM[0]
BD53	VSS	V9	UART2_CTS_N	AU16	DDR3_M1_CSB[1]
BE19	VSS	AD10	UART1_TXD	AY16	DDR3_M1_CSB[0]
BE35	VSS	AD12	UART1_RXD	AY12	DDR3_M1_CKE[1]
BF12	VSS	AD14	UART1_RTS_N	BB10	DDR3_M1_CKE[0]
BF22	VSS	AD13	UART1_CTS_N	BF16	DDR3_M1_CKB[1]
BF26	VSS	AB48	TRST_N	BF14	DDR3_M1_CKB[0]
BF27	VSS	AD48	TMS	BD16	DDR3_M1_CK[1]
BF28	VSS	P30	RSVD	BD14	DDR3_M1_CK[0]
BF32	VSS	P28	RSVD	BG9	DDR3_M1_CAS_N
BF4	VSS	AF40	TDO	BF2	DDR3_M1_BS[2]
BF42	VSS	AD47	TDI	AY14	DDR3_M1_BS[1]
BF50	VSS	AF42	TCK	BH8	DDR3_M1_BS[0]
BG14	VSS	AD41	SVID0_DATA	BH44	DDR3_M0_WE_N
BG16	VSS	AD42	SVID0_CLK	BA28	DDR3_M0_RCOMP
BG18	VSS	AD40	SVID0_ALERT_N	BA40	DDR3_M0_RAS_N
BG19	VSS	AE3	SUSPWRDNACK	BA38	DDR3_M0_ODT[1]
BG20	VSS	D14	SUS_STAT_N	AV36	DDR3_M0_ODT[0]
BG24	VSS	D18	SRTCRST_N	AU28	DDR3_M0_ODQVREF
BG27	VSS	H4	SPKR	AT28	DDR3_M0_OCAVREF
BG30	VSS	V12	SPI1_MOSI	AT30	RSVD

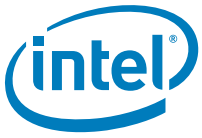


Table 19-1. SoC Pin List Locations (Sheet 7 of 11)

Pin #	Pin Name	Pin #	Pin Name	Pin #	Pin Name
BG34	VSS	V13	SPI1_MISO	AU30	RSVD
BG35	VSS	Y12	SPI1_CS1_N	BE52	DDR3_M0_MA[9]
BG36	VSS	Y13	SPI1_CS0_N	BD44	DDR3_M0_MA[8]
BG38	VSS	V14	SPI1_CLK	BE46	DDR3_M0_MA[7]
BG40	VSS	J26	RSVD	BB46	DDR3_M0_MA[6]
BG47	VSS	K26	RSVD	BH48	DDR3_M0_MA[5]
BG49	VSS	N26	RSVD	BD42	DDR3_M0_MA[4]
BG5	VSS	AF52	SEC_GPIO_SUS9	BH47	DDR3_M0_MA[3]
BG7	VSS	AF51	SEC_GPIO_SUS8	BJ48	DDR3_M0_MA[2]
BJ11	VSS	AC51	SEC_GPIO_SUS11	BD49	DDR3_M0_MA[15]
BJ15	VSS	AE51	SEC_GPIO_SUS10	BD47	DDR3_M0_MA[14]
BJ19	VSS	P12	SDMMC3_RCOMP	BF44	DDR3_M0_MA[13]
BJ23	VSS	L3	SDMMC3_PWR_EN_N	BF48	DDR3_M0_MA[12]
BJ27	VSS	G2	SDMMC3_D3	BB49	DDR3_M0_MA[11]
BJ31	VSS	H3	SDMMC3_D2	BJ45	DDR3_M0_MA[10]
BJ35	VSS	J3	SDMMC3_D1	BC42	DDR3_M0_MA[1]
BJ39	VSS	J1	SDMMC3_D0	BB47	DDR3_M0_MA[0]
BJ43	VSS	D2	SDMMC3_CMD	BA42	DDR3_M0_DRAMRST_N
BJ47	VSS	F2	SDMMC3_CLK	BG31	DDR3_M0_DQSB[7]
BJ7	VSS	K3	SDMMC3_CD_N	BC32	DDR3_M0_DQSB[6]
C20	VSS	K2	SDMMC3_1P8_EN	AT34	DDR3_M0_DQSB[5]
C22	VSS	K6	SDMMC2_D3_CD_N	BG39	DDR3_M0_DQSB[4]
C28	VSS	K7	SDMMC2_D2	BA51	DDR3_M0_DQSB[3]
C3	VSS	M10	SDMMC2_D1	AT41	DDR3_M0_DQSB[2]
C30	VSS	M12	SDMMC2_D0	AV48	DDR3_M0_DQSB[1]
C36	VSS	K9	SDMMC2_CMD	AM51	DDR3_M0_DQSB[0]
C39	VSS	K10	SDMMC2_CLK	BH32	DDR3_M0_DQS[7]
C47	VSS	P13	SDMMC1_RCOMP	BC30	DDR3_M0_DQS[6]
D10	VSS	P7	SDMMC1_D3_CD_N	AT32	DDR3_M0_DQS[5]
D16	VSS	P9	SDMMC1_D2	BH40	DDR3_M0_DQS[4]
D24	VSS	M4	SDMMC1_D1	AY52	DDR3_M0_DQS[3]
D27	VSS	M6	SDMMC1_D0	AT42	DDR3_M0_DQS[2]
D32	VSS	P6	SDMMC1_CMD	AV47	DDR3_M0_DQS[1]
D38	VSS	M7	SDMMC1_CLK	AM52	DDR3_M0_DQS[0]
D40	VSS	C29	SATA_TXP1	AT47	DDR3_M0_DQ[9]
D42	VSS	C31	SATA_TXP0	AP48	DDR3_M0_DQ[8]

Table 19-1. SoC Pin List Locations (Sheet 8 of 11)

Pin #	Pin Name
E19	VSS
E35	VSS
E46	VSS
E51	VSS
F19	VSS
F24	VSS
F27	VSS
F32	VSS
F35	VSS
F5	VSS
G12	VSS
G14	VSS
G22	VSS
G26	VSS
G28	VSS
G30	VSS
H19	VSS
H27	VSS
H35	VSS
H8	VSS
J18	VSS
J19	VSS
J22	VSS
J27	VSS
J30	VSS
J35	VSS
J38	VSS
J42	VSS
J53	VSS
K12	VSS
K14	VSS
K16	VSS
K22	VSS
K24	VSS
K30	VSS
K32	VSS
K34	VSS
K36	VSS

Pin #	Pin Name
A29	SATA_TXN1
B30	SATA_TXN0
J28	SATA_RXP1
N28	SATA_RXP0
K28	SATA_RXN1
M28	SATA_RXN0
N30	SATA_RCOMP_P
M30	SATA_RCOMP_N
AH3	SATA_LED_N
AF3	SATA_GP3/ SATA_DEVSLP1
AG1	SATA_GP2/ SATA_DEVSLP0
AG3	SATA_GP1
AH2	SATA_GP0
J16	RTC_TEST_N
M44	RSVD
K44	RSVD
K48	RSVD
K47	RSVD
F18	RSMRST_N
H7	RSVD
H5	RSVD
AD50	PROCHOT_N
P18	PMU_WAKE_LAN_N
N16	PMU_WAKE_N
C15	PMU_SUSCLK
C12	PMU_SLP_S4_N
B14	PMU_SLP_S3_N
A13	PMU_SLP_S0ix_N
B12	PMU_SLP_LAN_N
AF2	PMU_RESETBUTTON_N
M16	PMU_PWRBTN_N
F14	PMU_PLTRST_N
C14	PMU_BATLOW_N
C13	PMU_AC_PRESENT
A27	PCIE_TXP3
B26	PCIE_TXP2
A25	PCIE_TXP1
C24	PCIE_TXP0

Pin #	Pin Name
AP51	DDR3_M0_DQ[7]
BG33	DDR3_M0_DQ[63]
BH28	DDR3_M0_DQ[62]
BJ29	DDR3_M0_DQ[61]
BG28	DDR3_M0_DQ[60]
AR53	DDR3_M0_DQ[6]
BG32	DDR3_M0_DQ[59]
BH34	DDR3_M0_DQ[58]
BG29	DDR3_M0_DQ[57]
BJ33	DDR3_M0_DQ[56]
BD28	DDR3_M0_DQ[55]
BF30	DDR3_M0_DQ[54]
BA34	DDR3_M0_DQ[53]
BD34	DDR3_M0_DQ[52]
BD30	DDR3_M0_DQ[51]
BA32	DDR3_M0_DQ[50]
AK52	DDR3_M0_DQ[5]
BC34	DDR3_M0_DQ[49]
BF34	DDR3_M0_DQ[48]
AV32	DDR3_M0_DQ[47]
AV34	DDR3_M0_DQ[46]
BD36	DDR3_M0_DQ[45]
BF36	DDR3_M0_DQ[44]
AU32	DDR3_M0_DQ[43]
AU34	DDR3_M0_DQ[42]
BA36	DDR3_M0_DQ[41]
BC36	DDR3_M0_DQ[40]
AL53	DDR3_M0_DQ[4]
BH38	DDR3_M0_DQ[39]
BH36	DDR3_M0_DQ[38]
BJ41	DDR3_M0_DQ[37]
BH42	DDR3_M0_DQ[36]
BJ37	DDR3_M0_DQ[35]
BG37	DDR3_M0_DQ[34]
BG43	DDR3_M0_DQ[33]
BG42	DDR3_M0_DQ[32]
BB51	DDR3_M0_DQ[31]
AW53	DDR3_M0_DQ[30]

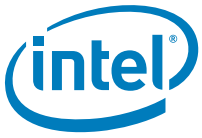


Table 19-1. SoC Pin List Locations (Sheet 9 of 11)

Pin #	Pin Name	Pin #	Pin Name	Pin #	Pin Name
K4	VSS	C27	PCIE_TXN3	AR51	DDR3_M0_DQ[3]
K45	VSS	C26	PCIE_TXN2	BC52	DDR3_M0_DQ[29]
K50	VSS	C25	PCIE_TXN1	AW51	DDR3_M0_DQ[28]
L1	VSS	B24	PCIE_TXN0	AV51	DDR3_M0_DQ[27]
L19	VSS	G24	PCIE_RXP3	BC53	DDR3_M0_DQ[26]
L27	VSS	D22	PCIE_RXP2	AV52	DDR3_M0_DQ[25]
L35	VSS	D20	PCIE_RXP1	BD52	DDR3_M0_DQ[24]
L41	VSS	G20	PCIE_RXP0	AV42	DDR3_M0_DQ[23]
M14	VSS	J24	PCIE_RXN3	AP41	DDR3_M0_DQ[22]
M19	VSS	F22	PCIE_RXN2	AV41	DDR3_M0_DQ[21]
M27	VSS	F20	PCIE_RXN1	AT44	DDR3_M0_DQ[20]
M35	VSS	J20	PCIE_RXN0	AT52	DDR3_M0_DQ[2]
M40	VSS	D26	PCIE_RCOMP_P	AP40	DDR3_M0_DQ[19]
M45	VSS	F26	PCIE_RCOMP_N	AT38	DDR3_M0_DQ[18]
M50	VSS	AM14	PCIE_CLKREQ3_N	AP42	DDR3_M0_DQ[17]
M9	VSS	AK14	PCIE_CLKREQ2_N	AT40	DDR3_M0_DQ[16]
N22	VSS	AM12	PCIE_CLKREQ1_N	AV45	DDR3_M0_DQ[15]
N24	VSS	AM10	PCIE_CLKREQ0_N	AY50	DDR3_M0_DQ[14]
N32	VSS	R53	PANEL1_VDDEN	AT50	DDR3_M0_DQ[13]
N51	VSS	P51	PANEL1_BKLTEN	AP47	DDR3_M0_DQ[12]
N53	VSS	P52	PANEL1_BKLTCTL	AV50	DDR3_M0_DQ[11]
P10	VSS	W53	PANEL0_VDDEN	AY48	DDR3_M0_DQ[10]
P19	VSS	V52	PANEL0_BKLTEN	AL51	DDR3_M0_DQ[1]
P22	VSS	V51	PANEL0_BKLTCTL	AK51	DDR3_M0_DQ[0]
P27	VSS	M22	OSCOUT	BH30	DDR3_M0_DM[7]
P32	VSS	P24	OSCIN	BD32	DDR3_M0_DM[6]
P35	VSS	T13	SDMMC1_RCLK	AY36	DDR3_M0_DM[5]
P36	VSS	T12	SDMMC1_D7	BG41	DDR3_M0_DM[4]
P4	VSS	T10	SDMMC1_D6	BA53	DDR3_M0_DM[3]
P42	VSS	T7	SDMMC1_D5	AP44	DDR3_M0_DM[2]
R1	VSS	T6	SDMMC1_D4_SD_WE	AT48	DDR3_M0_DM[1]
T14	VSS	AM7	MF_SMB_DATA	AP52	DDR3_M0_DM[0]
T47	VSS	AM6	MF_SMB_CLK	AU38	DDR3_M0_CSB[1]
T9	VSS	AM9	MF_SMB_ALERT_N	AY38	DDR3_M0_CSB[0]
U11	VSS	B4	MF_PLT_CLK5	AY42	DDR3_M0_CKE[1]
U12	VSS	B5	MF_PLT_CLK4	BB44	DDR3_M0_CKE[0]
U14	VSS	B7	MF_PLT_CLK3	BF38	DDR3_M0_CKB[1]
U18	VSS	B8	MF_PLT_CLK2	BF40	DDR3_M0_CKB[0]
U21	VSS	C9	MF_PLT_CLK1	BD38	DDR3_M0_CK[1]
U25	VSS	A9	MF_PLT_CLK0	BD40	DDR3_M0_CK[0]

Table 19-1. SoC Pin List Locations (Sheet 10 of 11)

Pin #	Pin Name	Pin #	Pin Name	Pin #	Pin Name
U29	VSS	R3	MF_LPC_CLKOUT1	BG45	DDR3_M0_CAS_N
U30	VSS	P2	MF_LPC_CLKOUT0	BF52	DDR3_M0_BS[2]
U32	VSS	N1	MF_LPC_AD3	AY40	DDR3_M0_BS[1]
U33	VSS	N3	MF_LPC_AD2	BH46	DDR3_M0_BS[0]
U36	VSS	M2	MF_LPC_AD1	AV28	DDR3_DRAM_PWROK
U38	VSS	M3	MF_LPC_AD0	D44	DDI2_TXP[3]
U40	VSS	L13	RSVD	F42	DDI2_TXP[2]
U42	VSS	J14	RSVD	J40	DDI2_TXP[1]
U43	VSS	F7	RSVD	F40	DDI2_TXP[0]
U45	VSS	J12	RSVD	F44	DDI2_TXN[3]
U46	VSS	D6	RSVD	G42	DDI2_TXN[2]
U48	VSS	C7	RSVD	K40	DDI2_TXN[1]
U49	VSS	E8	RSVD	G40	DDI2_TXN[0]
U5	VSS	D12	RSVD	D48	DDI2_AUXP
U53	VSS	F10	RSVD	C49	DDI2_AUXN
U6	VSS	F12	RSVD	M52	DDI1_TXP[3]
U8	VSS	B10	RSVD	L53	DDI1_TXP[2]
U9	VSS	C11	RSVD	K51	DDI1_TXP[1]
V16	VSS	AF12	MF_HDA_SYNC	J51	DDI1_TXP[0]
V21	VSS	AF14	MF_HDA_SDO	M51	DDI1_TXN[3]
V25	VSS	AD6	MF_HDA_SDI1	L51	DDI1_TXN[2]
V32	VSS	AD7	MF_HDA_SDI0	K52	DDI1_TXN[1]
V38	VSS	AF13	MF_HDA_RST_N	H51	DDI1_TXN[0]
V41	VSS	AB7	MF_HDA_DOCKRST_N	F47	DDI1_PLLOBS_P
V42	VSS	AB9	MF_HDA_DOCKEN_N	F49	DDI1_PLLOBS_N
V44	VSS	AD9	MF_HDA_CLK	M42	DDI1_AUXP
W1	VSS	G44	RSVD	K42	DDI1_AUXN
Y10	VSS	A49	RSVD	G53	DDI0_TXP[3]
Y14	VSS	A51	RSVD	F53	DDI0_TXP[2]
Y16	VSS	C53	RSVD	H49	DDI0_TXP[1]
Y19	VSS	E53	RSVD	D50	DDI0_TXP[0]
Y21	VSS	B49	RSVD	G52	DDI0_TXN[3]
Y22	VSS	B50	RSVD	F52	DDI0_TXN[2]
Y24	VSS	D52	RSVD	H50	DDI0_TXN[1]
Y29	VSS	E52	RSVD	C51	DDI0_TXN[0]
Y38	VSS	B53	RSVD	F38	DDI0_PLLOBS_P
Y4	VSS	A52	RSVD	G38	DDI0_PLLOBS_N
Y40	VSS	P44	MCSI_COMP	H47	DDI0_AUXP
Y45	VSS	T50	MCSI_3_CLKP	H46	DDI0_AUXN

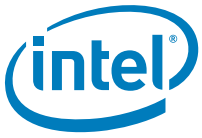
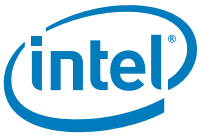


Table 19-1. SoC Pin List Locations (Sheet 11 of 11)

Pin #	Pin Name	Pin #	Pin Name	Pin #	Pin Name
Y50	VSS	T48	MCSI_3_CLKN	AF41	CX_PREQ_N
Y9	VSS	M48	MCSI_2_DP[1]	AD45	CX_PRDY_N
AC27	UNCORE_VSS_SENSE	P47	MCSI_2_DP[0]	G16	COREPWROK
AD22	UNCORE_VSS_SENSE	M47	MCSI_2_DN[1]	C16	RSVD
AD24	DDI_VGG_SENSE	P45	MCSI_2_DN[0]	C17	CLK_DIFF_P[3]
U35	USB_VDDQ_G3	P50	MCSI_2_CLKP	C18	CLK_DIFF_P[2]
V35	USB_VDDQ_G3	P48	MCSI_2_CLKN	C19	CLK_DIFF_P[1]
B22	USB_V3P3A_G3	T41	MCSI_1_DP[3]	A21	CLK_DIFF_P[0]
C23	USB_V3P3A_G3	V50	MCSI_1_DP[2]	B16	RSVD
AA29	USB_V1P8A_G3	V45	MCSI_1_DP[1]	A17	CLK_DIFF_N[3]
AA30	RSVD	Y47	MCSI_1_DP[0]	B18	CLK_DIFF_N[2]
AA32	VCCSRAMSOCIUN_1P05	T42	MCSI_1_DN[3]	B20	CLK_DIFF_N[1]
AA33	VCCSRAMSOCIUN_1P05	V48	MCSI_1_DN[2]	C21	CLK_DIFF_N[0]
AA35	VCCSRAMSOCIUN_1P05	V47	MCSI_1_DN[1]	F16	RTC_EXTPAD
AA36	VCCSRAMSOCIUN_1P05	Y48	MCSI_1_DN[0]	K18	RTC_X2
AC32	VCCSRAMSOCIUN_1P05			M18	RTC_X1
V33	VCCSRAMSOCIUN_1P05				

§ §



20 Package Information

The SoC comes in a 27 mm x 25 mm Flip-Chip Ball Grid Array (FCBGA) package and consists of a silicon die mounted face down on an organic substrate populated with 1170 solder balls on the bottom side. Capacitors may be placed in the area surrounding the die. Because the die-side capacitors are electrically conductive, and only slightly shorter than the die height, care should be taken to avoid contacting the capacitors with electrically conductive materials. Doing so may short the capacitors and possibly damage the device or render it inactive.

The use of an insulating material between the capacitors and any thermal solution should be considered to prevent capacitor shorting. An exclusion, or keep out zone, surrounds the die and capacitors, and identifies the contact area for the package. Care should be taken to avoid contact with the package inside this area.

20.1 SoC Attributes

Attribute	SoC
X-Y dimensions (mm)	27 mm x 25 mm
Processor Core Process (nm)	14
Post - SMT Height (mm)	1
Minimum BGA Ball Pitch (mm)	0.593
Die Thickness (µm)	370
Total Pin Count	1170
Package Type	FCBGA15

20.2 Package Diagrams

Figure 20-1. Package Mechanical Drawing—Part 1 of 3

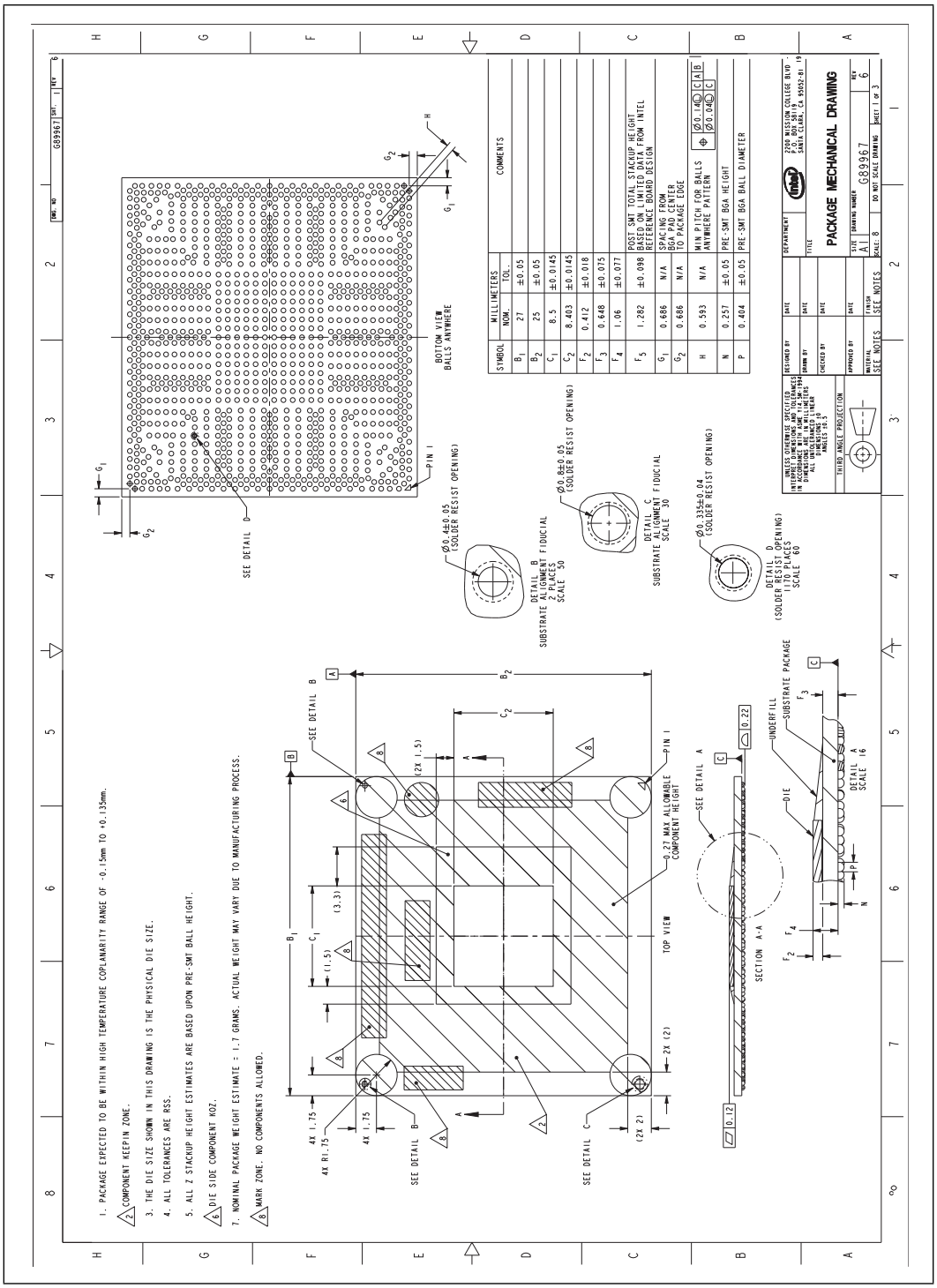
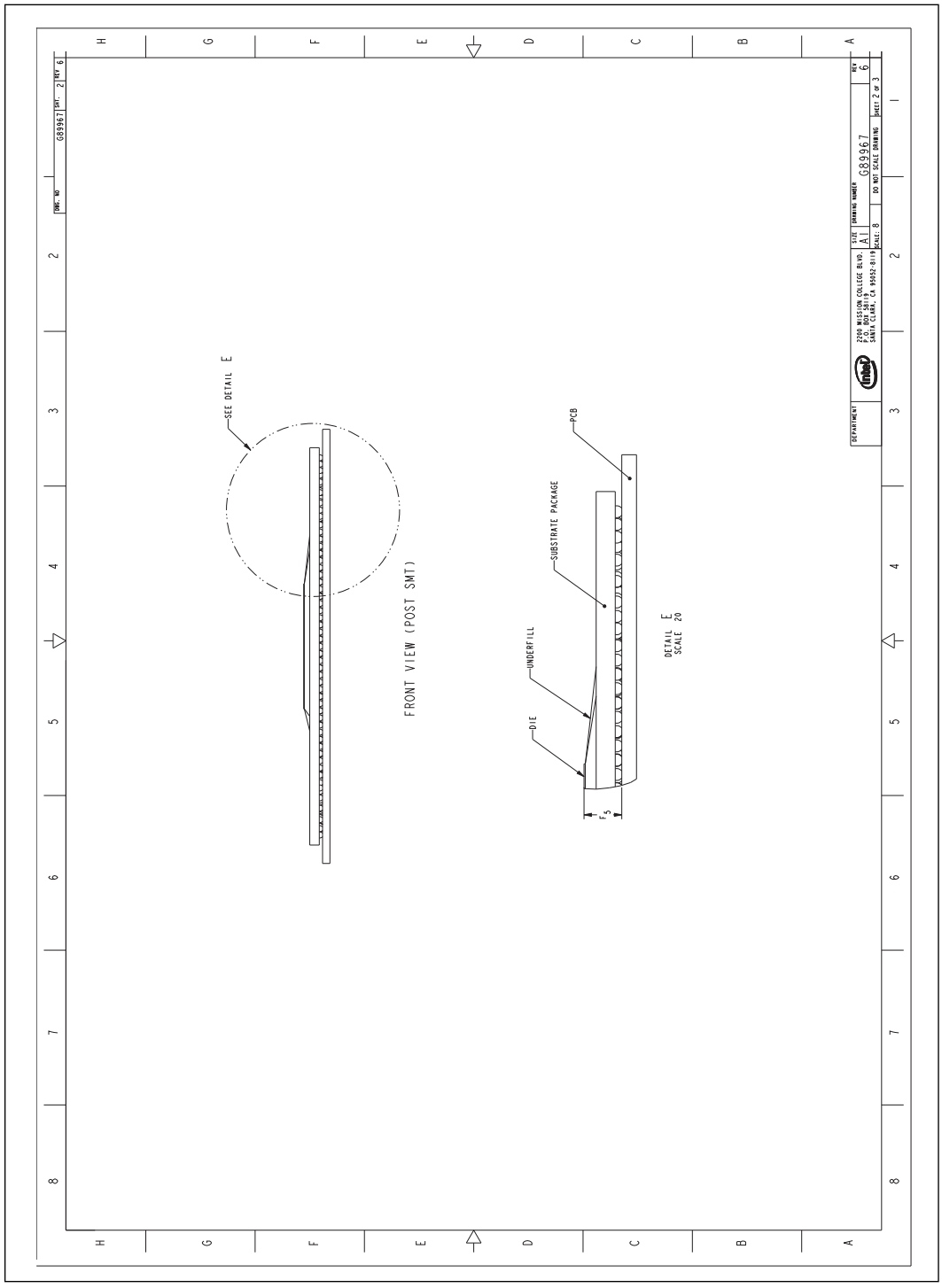
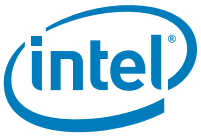


Figure 20-2. Package Mechanical Drawing—Part 2 of 3



§ §



21 Electrical Specifications

21.1 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.

21.2 Thermal Specifications

The thermal solution provides both component-level and system-level thermal management. To allow optimal operation and long-term reliability of Intel processor-based systems, the system/processor thermal solution should be designed so that the processor:

- Remains below the maximum junction temperature (Tj_{MAX}) specification at the maximum thermal design power (TDP).
- Conforms to system constraints, such as system acoustics, system skin-temperatures, and exhaust-temperature requirements.

Caution: Thermal specifications given in this section are on the component and package level and apply specifically to the processor. Operating the processor outside the specified limits may result in permanent damage to the processor and potentially other components in the system.

The processor TDP is the maximum sustained power that should be used for design of the processor thermal solution. TDP is a power dissipation and junction temperature operating condition limit, specified in this document, that is validated during manufacturing for the base configuration when executing a near worst case commercially available workload as specified by Intel for the SKU segment. TDP may be exceeded for short periods of time or if running a very high power workload.

The following table specifies the thermal limits for the processor based on the definitions above.

Note: 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 21-1. SoC Base Frequencies and Thermal Specifications

SKU Segment	Configur-ation	Processor Frequency LFM/HFM/Burst (Hz)	Graphics Frequency LFM/HFM/Dyn (Hz)	Thermal Design Power TDP (W)	Scenario Design Power SDP (W)	Tj_{MAX} (°C)
N3700, QC	Base	480M/1.6G/2.4G	200M/400M/700M	6	4	90
N3150,QC	Base	480M/1.6G/2.08G	200M/320M/640M	6	4	90
N3050,DC	Base	480M/1.6G/2.16G	200M/320M/600M	6	4	90
N3000,DC	Base	480M/1.04G/2.08G	200M/320M/600M	4	3	90

21.3 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 21-2. 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.	–20 °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.	N/A	60%RH @ 24 °C

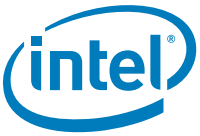
Notes:
1. Specified temperatures are not to exceed values based on data collected. Exceptions for surface mount re-flow are specified by the applicable JEDEC* standard. Non-adherence may affect processor reliability.
2. Component product device storage temperature qualification methods may follow JESD22-A119 (low temperature) and JESD22-A103 (high temperature) standards when applicable for volatile memory.
3. Component stress testing is conducted in conformance with JESD22-A104.
4. The JEDEC* J-JSTD-020 moisture level rating and associated handling practices apply to all moisture sensitive devices removed from the moisture barrier bag.

21.3.1 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



21.4 Voltage and Current Specifications

The I/O buffer supply voltages are specified at the SoC package balls. The tolerances shown in Table 21-3 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 21-5 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.

Note: RTC_VCC average current draw (G5) is specified at 27 °C under battery conditions

Table 21-3. C-Step SoC Power Rail DC Specifications and Maximum Current

Power Rail (SoC)	N3000 (TDP-4W) (SDP-3W) (DC) (S0-Imax) (mA)	N3050 (TDP-6W) (SDP-4W) (DC) (S0 Imax) (mA)	N3700 (TDP-6W)(SDP-4W) (QC) (S0 Imax), (mA)	N3150 (TDP-6W) (SDP-4W) (QC) (Imax, mA)	S3 Imax (mA)	S4 Imax (mA)	S5 Imax (mA)
VCC0+VCC1 (merged)	3600	3600	7700	7700	0	0	0
VGG	11000	11000	11000	11000	0	0	0
VNN	3500	3500	3500	3500	175	175	175
V1P05A	1900	2000	2000	2000	15	15	15
V1P15S	500	500	500	500	0	0	0
V1P24A	500	500	500	500	5	5	5
V1P5S or V1P8S	20	20	20	20	1	0	0
V1P8A	550	550	550	550	5	5	5
V3P3A_PRIME	200	200	200	200	1	1	1
LPC IO (3.3V)	148	148	148	148	1	1	1
VSDIO (3.3V)	141	141	141	141	1	0	0
VSDIO (1.8V)	93	93	93	93	1	0	0
VDDQ (1.35V)	2400	2400	2400	2400	15	0	0
VCC_RTC	1	1	1	1	1	1	1

Notes:

1. VCC_RTC Iccmax in G3 state is 6uA. This current specification is valid at an ambient temperature of 25^{o}C with 3V coin cell battery
2. The data in this table only represent peak or worst case conditions and does NOT represent sustained or average current requirements.
3. The data in this table should ONLY be used for power delivery or voltage regulator (VR) design. These numbers should only be used as guidance to enable appropriate power delivery or voltage regulator part selection and should not be used for Battery Life analysis

Table 21-4. D-Step SoC Power Rail DC Specifications and Maximum Current

Power Rail (SoC)	J3060 (TDP-6W) (S0-Imax) (mA)	J3160 (TDP-6W) (S0-Imax) (mA)	J3710 (TDP-6.5W) (S0-Imax) (mA)	S3 Imax (mA)	S4 Imax (mA)	S5 Imax (mA)
VCC0+VCC1 (merged)	10000	10000	10000	0	0	0
VGG	12000	12000	12000	0	0	0
VNN	3500	3500	3500	175	175	175
V1P05A	1900	2000	2000	15	15	15
V1P15S	500	500	500	0	0	0
V1P24A	500	500	500	5	5	5
VIP5S or V1P8S	20	20	20	1	0	0
V1P8A	550	550	550	5	5	5
V3P3A_PRIME	200	200	200	1	1	1
LPC IO (3.3V)	148	148	148	1	1	1
VSDIO (3.3V)	141	141	141	1	0	0
VSDIO (1.8V)	93	93	93	1	0	0
VDDQ (1.35V)	2400	2400	2400	15	0	0
VCC_RTC	1	1	1	1	1	1

Notes:
1. VCC_RTC Iccmax in G3 state is 6uA. This current specification is valid at an ambient temperature of 25 ^{o}C with 3V coin cell battery
2. The data in this table only represent peak or worst case conditions and does NOT represent sustained or average current requirements.
3. The data in this table should ONLY be used for power delivery or voltage regulator (VR) design. These numbers should only be used as guidance to enable appropriate power delivery or voltage regulator part selection and should not be used for Battery Life analysis

21.4.1 VCC, VGG, and VNN Voltage Specifications

Table 21-5 and Table 21-19 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 21-5. VCC, VGG, and VNN DC Voltage Specifications (Sheet 1 of 2)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Notes
CORE_VCC VID	Core VID Target Range	0.5		1.3	V	
CORE_VCC0	V_{CC0} for SoC Core 0	See VCC VID			V	2, 5
CORE_VCC1	V_{CC1} for SoC Core 1	See VCC VID			V	2, 5
UNCORE_VNN VID	Uncore VID Target Range	0.5		1.05	V	
UNCORE_VNN_G3	V_{NN} for SoC Uncore	See VNN VID			V	2, 5
DDI_VGG	V_{GG} for SoC Display	0.5		1.2	V	
CORE_VCC/ UNCORE_VNN V_{BOOT}	Default target V_{CC}/V_{NN} voltage for initial power up.	–	1.0	–	V	4
VCC0/1 Tolerance	Tolerance of VCC0/1 voltage at VID target.	-50	–	+50	mV	

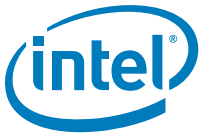


Table 21-5. VCC, VGG, and VNN DC Voltage Specifications (Sheet 2 of 2)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Notes
VNN Tolerance	Tolerance of VNN voltage at VID target.	-50	–	+50	mV	
VGG Tolerance	Tolerance of VGG voltage at VID target	–60	–	60	mV	6

Notes:
1. 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. This differs from the VID employed by the SoC during a power management event.
2. N/A
3. CC0 and VCC1 are merged into single SVID rail.
4. Depending on configuration chosen, VNN SVID can be either variable VID or fixed VID to a voltage such as 1.05V.
5. The processor SVID is based on VR12.1 requirement. VID DAC is similar to IMPV7 VID DAC.
6. Maximum threshold allowed can be 90mV as long as it returns to 60mV within 20 µs.

Table 21-6.VSDIO Voltage Setting

SDMMC3_PWR_EN_B	SDMMC3_1P8_EN	VSDIO(V)
1	0	0
1	1	0
0	0	3.3
0	1	1.8

21.4.2 Voltage Identification (VID)

Table 21-7 specifies the voltage level corresponding to the eight-bit VID value transmitted over serial VID (SVID) interface per IMVP7 specification. A '1' in this table refers to a high voltage level and a '0' refers to a low voltage level. If the voltage regulation circuit cannot supply the voltage that is requested, the voltage regulator must disable itself. The SVID signals are CMOS push/pull drivers. Refer to Table 21-36 for the DC specifications for these signals. The VID codes will change due to performance, temperature and/or current load changes in order to minimize the power of the part. A voltage range is provided in Table 21-5. The specifications are set so that one voltage regulator can operate with all supported frequencies.

Individual SoC VID values may be set during manufacturing so that two devices at the same core frequency may have different default VID settings. This is shown in the VID range values in Table 21-5. The SoC provides the ability to operate while transitioning to an adjacent VID and its associated voltage. This will represent a DC shift in the loadline.

Note: The following table lists all voltages possible per IMVP7 specification. Not all voltages are valid on actual SKUs.

Table 21-7. IMVP7.0 Voltage Identification Reference (Sheet 1 of 8)

VID7	VID6	VID5	VID4	VID3	VID2	VID1	VID0	Hex Bit 1	Hex Bit 0	V_{CC} (V)
0	0	0	0	0	0	0	0	0	0	0.00000
0	0	0	0	0	0	0	1	0	1	0.25000
0	0	0	0	0	0	1	0	0	2	0.25500

Table 21-7. IMVP7.0 Voltage Identification Reference (Sheet 2 of 8)

VID7	VID6	VID5	VID4	VID3	VID2	VID1	VID0	Hex Bit 1	Hex Bit 0	V_{CC} (V)
0	0	0	0	0	0	1	1	0	3	0.26000
0	0	0	0	0	1	0	0	0	4	0.26500
0	0	0	0	0	1	0	1	0	5	0.27000
0	0	0	0	0	1	1	0	0	6	0.27500
0	0	0	0	0	1	1	1	0	7	0.28000
0	0	0	0	1	0	0	0	0	8	0.28500
0	0	0	0	1	0	0	1	0	9	0.29000
0	0	0	0	1	0	1	0	0	A	0.29500
0	0	0	0	1	0	1	1	0	B	0.30000
0	0	0	0	1	1	0	0	0	C	0.30500
0	0	0	0	1	1	0	1	0	D	0.31000
0	0	0	0	1	1	1	0	0	E	0.31500
0	0	0	0	1	1	1	1	0	F	0.32000
0	0	0	1	0	0	0	0	1	0	0.32500
0	0	0	1	0	0	0	1	1	1	0.33000
0	0	0	1	0	0	1	0	1	2	0.33500
0	0	0	1	0	0	1	1	1	3	0.34000
0	0	0	1	0	1	0	0	1	4	0.34500
0	0	0	1	0	1	0	1	1	5	0.35000
0	0	0	1	0	1	1	0	1	6	0.35500
0	0	0	1	0	1	1	1	1	7	0.36000
0	0	0	1	1	0	0	0	1	8	0.36500
0	0	0	1	1	0	0	1	1	9	0.37000
0	0	0	1	1	0	1	0	1	A	0.37500
0	0	0	1	1	0	1	1	1	B	0.38000
0	0	0	1	1	1	0	0	1	C	0.38500
0	0	0	1	1	1	0	1	1	D	0.39000
0	0	0	1	1	1	1	0	1	E	0.39500
0	0	0	1	1	1	1	1	1	F	0.40000
0	0	1	0	0	0	0	0	2	0	0.40500
0	0	1	0	0	0	0	1	2	1	0.41000
0	0	1	0	0	0	1	0	2	2	0.41500
0	0	1	0	0	0	1	1	2	3	0.42000
0	0	1	0	0	1	0	0	2	4	0.42500
0	0	1	0	0	1	0	1	2	5	0.43000
0	0	1	0	0	1	1	0	2	6	0.43500
0	0	1	0	0	1	1	1	2	7	0.44000
0	0	1	0	1	0	0	0	2	8	0.44500
0	0	1	0	1	0	0	1	2	9	0.45000
0	0	1	0	1	0	1	0	2	A	0.45500

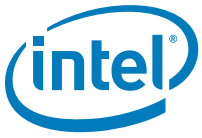


Table 21-7. IMVP7.0 Voltage Identification Reference (Sheet 3 of 8)

VID7	VID6	VID5	VID4	VID3	VID2	VID1	VID0	Hex Bit 1	Hex Bit 0	V_{CC} (V)
0	0	1	0	1	0	1	1	2	B	0.46000
0	0	1	0	1	1	0	0	2	C	0.46500
0	0	1	0	1	1	0	1	2	D	0.47000
0	0	1	0	1	1	1	0	2	E	0.47500
0	0	1	0	1	1	1	1	2	F	0.48000
0	0	1	1	0	0	0	0	3	0	0.48500
0	0	1	1	0	0	0	1	3	1	0.49000
0	0	1	1	0	0	1	0	3	2	0.49500
0	0	1	1	0	0	1	1	3	3	0.50000
0	0	1	1	0	1	0	0	3	4	0.50500
0	0	1	1	0	1	0	1	3	5	0.51000
0	0	1	1	0	1	1	0	3	6	0.51500
0	0	1	1	0	1	1	1	3	7	0.52000
0	0	1	1	1	0	0	0	3	8	0.52500
0	0	1	1	1	0	0	1	3	9	0.53000
0	0	1	1	1	0	1	0	3	A	0.53500
0	0	1	1	1	0	1	1	3	B	0.54000
0	0	1	1	1	1	0	0	3	C	0.54500
0	0	1	1	1	1	0	1	3	D	0.55000
0	0	1	1	1	1	1	0	3	E	0.55500
0	0	1	1	1	1	1	1	3	F	0.56000
0	1	0	0	0	0	0	0	4	0	0.56500
0	1	0	0	0	0	0	1	4	1	0.57000
0	1	0	0	0	0	1	0	4	2	0.57500
0	1	0	0	0	0	1	1	4	3	0.58000
0	1	0	0	0	1	0	0	4	4	0.58500
0	1	0	0	0	1	0	1	4	5	0.59000
0	1	0	0	0	1	1	0	4	6	0.59500
0	1	0	0	0	1	1	1	4	7	0.60000
0	1	0	0	1	0	0	0	4	8	0.60500
0	1	0	0	1	0	0	1	4	9	0.61000
0	1	0	0	1	0	1	0	4	A	0.61500
0	1	0	0	1	0	1	1	4	B	0.62000
0	1	0	0	1	1	0	0	4	C	0.62500
0	1	0	0	1	1	0	1	4	D	0.63000
0	1	0	0	1	1	1	0	4	E	0.63500
0	1	0	0	1	1	1	1	4	F	0.64000
0	1	0	1	0	0	0	0	5	0	0.64500
0	1	0	1	0	0	0	1	5	1	0.65000
0	1	0	1	0	0	1	0	5	2	0.65500

Table 21-7. IMVP7.0 Voltage Identification Reference (Sheet 4 of 8)

VID7	VID6	VID5	VID4	VID3	VID2	VID1	VID0	Hex Bit 1	Hex Bit 0	V_{CC} (V)
0	1	0	1	0	0	1	1	5	3	0.66000
0	1	0	1	0	1	0	0	5	4	0.66500
0	1	0	1	0	1	0	1	5	5	0.67000
0	1	0	1	0	1	1	0	5	6	0.67500
0	1	0	1	0	1	1	1	5	7	0.68000
0	1	0	1	1	0	0	0	5	8	0.68500
0	1	0	1	1	0	0	1	5	9	0.69000
0	1	0	1	1	0	1	0	5	A	0.69500
0	1	0	1	1	0	1	1	5	B	0.70000
0	1	0	1	1	1	0	0	5	C	0.70500
0	1	0	1	1	1	0	1	5	D	0.71000
0	1	0	1	1	1	1	0	5	E	0.71500
0	1	0	1	1	1	1	1	5	F	0.72000
0	1	1	0	0	0	0	0	6	0	0.72500
0	1	1	0	0	0	0	1	6	1	0.73000
0	1	1	0	0	0	1	0	6	2	0.73500
0	1	1	0	0	0	1	1	6	3	0.74000
0	1	1	0	0	1	0	0	6	4	0.74500
0	1	1	0	0	1	0	1	6	5	0.75000
0	1	1	0	0	1	1	0	6	6	0.75500
0	1	1	0	0	1	1	1	6	7	0.76000
0	1	1	0	1	0	0	0	6	8	0.76500
0	1	1	0	1	0	0	1	6	9	0.77000
0	1	1	0	1	0	1	0	6	A	0.77500
0	1	1	0	1	0	1	1	6	B	0.78000
0	1	1	0	1	1	0	0	6	C	0.78500
0	1	1	0	1	1	0	1	6	D	0.79000
0	1	1	0	1	1	1	0	6	E	0.79500
0	1	1	0	1	1	1	1	6	F	0.80000
0	1	1	1	0	0	0	0	7	0	0.80500
0	1	1	1	0	0	0	1	7	1	0.81000
0	1	1	1	0	0	1	0	7	2	0.81500
0	1	1	1	0	0	1	1	7	3	0.82000
0	1	1	1	0	1	0	0	7	4	0.82500
0	1	1	1	0	1	0	1	7	5	0.83000
0	1	1	1	0	1	1	0	7	6	0.83500
0	1	1	1	0	1	1	1	7	7	0.84000
0	1	1	1	1	0	0	0	7	8	0.84500
0	1	1	1	1	0	0	1	7	9	0.85000
0	1	1	1	1	0	1	0	7	A	0.85500

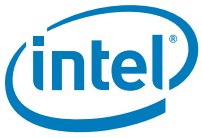


Table 21-7. IMVP7.0 Voltage Identification Reference (Sheet 5 of 8)

VID7	VID6	VID5	VID4	VID3	VID2	VID1	VID0	Hex Bit 1	Hex Bit 0	V_{CC} (V)
0	1	1	1	1	0	1	1	7	B	0.86000
0	1	1	1	1	1	0	0	7	C	0.86500
0	1	1	1	1	1	0	1	7	D	0.87000
0	1	1	1	1	1	1	0	7	E	0.87500
0	1	1	1	1	1	1	1	7	F	0.88000
1	0	0	1	0	0	0	0	8	0	0.88500
1	0	0	1	0	0	0	1	8	1	0.89000
1	0	0	1	0	0	1	0	8	2	0.89500
1	0	0	0	0	0	1	1	8	3	0.90000
1	0	0	0	0	1	0	0	8	4	0.90500
1	0	0	0	0	1	0	1	8	5	0.91000
1	0	0	0	0	1	1	0	8	6	0.91500
1	0	0	0	0	1	1	1	8	7	0.92000
1	0	0	0	1	0	0	0	8	8	0.92500
1	0	0	0	1	0	0	1	8	9	0.93000
1	0	0	0	1	0	1	0	8	A	0.93500
1	0	0	0	1	0	1	1	8	B	0.94000
1	0	0	0	1	1	0	0	8	C	0.94500
1	0	0	0	1	1	0	1	8	D	0.95000
1	0	0	0	1	1	1	0	8	E	0.95500
1	0	0	0	1	1	1	1	8	F	0.96000
1	0	0	0	0	0	0	0	9	0	0.96500
1	0	0	0	0	0	0	1	9	1	0.97000
1	0	0	0	0	0	1	0	9	2	0.97500
1	0	0	1	0	0	1	1	9	3	0.98000
1	0	0	1	0	1	0	0	9	4	0.98500
1	0	0	1	0	1	0	1	9	5	0.99000
1	0	0	1	0	1	1	0	9	6	0.99500
1	0	0	1	0	1	1	1	9	7	1.00000
1	0	0	1	1	0	0	0	9	8	1.00500
1	0	0	1	1	0	0	1	9	9	1.01000
1	0	0	1	1	0	1	0	9	A	1.01500
1	0	0	1	1	0	1	1	9	B	1.02000
1	0	0	1	1	1	0	0	9	C	1.02500
1	0	0	1	1	1	0	1	9	D	1.03000
1	0	0	1	1	1	1	0	9	E	1.03500
1	0	0	1	1	1	1	1	9	F	1.04000
1	0	1	1	0	0	0	0	A	0	1.04500
1	0	1	1	0	0	0	1	A	1	1.05000
1	0	1	1	0	0	1	0	A	2	1.05500

Table 21-7. IMVP7.0 Voltage Identification Reference (Sheet 6 of 8)

VID7	VID6	VID5	VID4	VID3	VID2	VID1	VID0	Hex Bit 1	Hex Bit 0	V_{CC} (V)
1	0	1	0	0	0	1	1	A	3	1.06000
1	0	1	0	0	1	0	0	A	4	1.06500
1	0	1	0	0	1	0	1	A	5	1.07000
1	0	1	0	0	1	1	0	A	6	1.07500
1	0	1	0	0	1	1	1	A	7	1.08000
1	0	1	0	1	0	0	0	A	8	1.08500
1	0	1	0	1	0	0	1	A	9	1.09000
1	0	1	0	1	0	1	0	A	A	1.09500
1	0	1	0	1	0	1	1	A	B	1.10000
1	0	1	0	1	1	0	0	A	C	1.10500
1	0	1	0	1	1	0	1	A	D	1.11000
1	0	1	0	1	1	1	0	A	E	1.11500
1	0	1	0	1	1	1	1	A	F	1.12000
1	0	1	0	0	0	0	0	B	0	1.12500
1	0	1	0	0	0	0	1	B	1	1.13000
1	0	1	0	0	0	1	0	B	2	1.13500
1	0	1	1	0	0	1	1	B	3	1.14000
1	0	1	1	0	1	0	0	B	4	1.14500
1	0	1	1	0	1	0	1	B	5	1.15000
1	0	1	1	0	1	1	0	B	6	1.15500
1	0	1	1	0	1	1	1	B	7	1.16000
1	0	1	1	1	0	0	0	B	8	1.16500
1	0	1	1	1	0	0	1	B	9	1.17000
1	0	1	1	1	0	1	0	B	A	1.17500
1	0	1	1	1	0	1	1	B	B	1.18000
1	0	1	1	1	1	0	0	B	C	1.18500
1	0	1	1	1	1	0	1	B	D	1.19000
1	0	1	1	1	1	1	0	B	E	1.19500
1	0	1	1	1	1	1	1	B	F	1.20000
1	1	0	0	0	0	0	0	C	0	1.20500
1	1	0	0	0	0	0	1	C	1	1.21000
1	1	0	0	0	0	1	0	C	2	1.21500
1	1	0	0	0	0	1	1	C	3	1.22000
1	1	0	0	0	1	0	0	C	4	1.22500
1	1	0	0	0	1	0	1	C	5	1.23000
1	1	0	0	0	1	1	0	C	6	1.23500
1	1	0	0	0	1	1	1	C	7	1.24000
1	1	0	0	1	1	0	0	C	8	1.24500
1	1	0	0	1	0	0	1	C	9	1.25000
1	1	0	0	1	0	1	0	C	A	1.25500

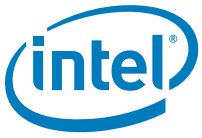


Table 21-7. IMVP7.0 Voltage Identification Reference (Sheet 7 of 8)

VID7	VID6	VID5	VID4	VID3	VID2	VID1	VID0	Hex Bit 1	Hex Bit 0	V_{CC} (V)
1	1	0	0	1	0	1	1	C	B	1.26000
1	1	0	0	1	0	0	0	C	C	1.26500
1	1	0	0	1	1	0	1	C	D	1.27000
1	1	0	0	1	1	1	0	C	E	1.27500
1	1	0	0	1	1	1	1	C	F	1.28000
1	1	0	1	0	1	0	0	D	0	1.28500
1	1	0	1	0	1	0	1	D	1	1.29000
1	1	0	1	0	0	1	0	D	2	1.29500
1	1	0	1	0	0	1	1	D	3	1.30000
1	1	0	1	0	1	0	0	D	4	1.30500
1	1	0	1	0	1	0	1	D	5	1.31000
1	1	0	1	0	1	1	0	D	6	1.31500
1	1	0	1	0	1	1	1	D	7	1.32000
1	1	0	1	1	0	0	0	D	8	1.32500
1	1	0	1	1	0	0	1	D	9	1.33000
1	1	0	1	1	0	1	0	D	A	1.33500
1	1	0	1	1	0	1	1	D	B	1.34000
1	1	0	1	1	1	0	0	D	C	1.34500
1	1	0	1	1	1	0	1	D	D	1.35000
1	1	0	1	1	1	1	0	D	E	1.35500
1	1	0	1	1	1	1	1	D	F	1.36000
1	1	1	0	0	0	0	0	E	0	1.36500
1	1	1	0	0	0	0	1	E	1	1.37000
1	1	1	0	0	0	1	0	E	2	1.37500
1	1	1	0	0	0	1	1	E	3	1.38000
1	1	1	0	0	1	0	0	E	4	1.38500
1	1	1	0	0	1	0	1	E	5	1.39000
1	1	1	0	0	1	1	0	E	6	1.39500
1	1	1	0	0	1	1	1	E	7	1.40000
1	1	1	0	1	0	0	0	E	8	1.40500
1	1	1	0	1	0	0	1	E	9	1.41000
1	1	1	0	1	0	1	0	E	A	1.41500
1	1	1	0	1	0	1	1	E	B	1.42000
1	1	1	0	1	1	0	0	E	C	1.42500
1	1	1	0	1	1	0	1	E	D	1.43000
1	1	1	0	1	1	1	0	E	E	1.43500
1	1	1	0	1	1	1	1	E	F	1.44000
1	1	1	1	0	0	0	0	F	0	1.44500
1	1	1	1	0	0	0	1	F	1	1.45000
1	1	1	1	0	0	1	0	F	2	1.45500

Table 21-7. IMVP7.0 Voltage Identification Reference (Sheet 8 of 8)

VID7	VID6	VID5	VID4	VID3	VID2	VID1	VID0	Hex Bit 1	Hex Bit 0	V_{CC} (V)
1	1	1	1	0	0	1	1	F	3	1.46000
1	1	1	1	0	1	0	0	F	4	1.46500
1	1	1	1	0	1	0	1	F	5	1.47000
1	1	1	1	0	1	1	0	F	6	1.47500
1	1	1	1	0	1	1	1	F	7	1.48000
1	1	1	1	1	0	0	0	F	8	1.48500
1	1	1	1	1	0	0	1	F	9	1.49000
1	1	1	1	1	0	1	0	F	A	1.49500
1	1	1	1	1	0	1	1	F	B	1.50000
1	1	1	1	1	1	0	0	F	C	1.49500
1	1	1	1	1	1	0	1	F	D	1.50000
1	1	1	1	1	1	1	0	F	E	1.49500
1	1	1	1	1	1	1	1	F	F	1.50000

21.5 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 processor.

Table 21-8. ILB RTC Crystal Specification

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
F_{RTC}	Frequency	–	32.768	–	KHz	1
T_{PPM}	Crystal frequency tolerance (see notes)	–	–	±20	ppm	1
R_{ESR}	ESR	–	–	50	KOhm	1
$C_{X1,2}$	Capacitance of X1, X2 pins	–	–	15	pF	1

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 second tolerance per day, RTC circuit itself contributes addition 10 ppm for a total of 20 ppm in this example.

Table 21-9. Integrated Clock Crystal Specification

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
F_{ICLK}	Frequency	–	19.2	–	MHz	1
T_{PPM}	Crystal frequency tolerance and stability	-30	–	+30	ppm	1
P_{DRIVE}	Crystal drive load	–	–	100	µW	1
R_{ESR}	ESR	–	–	80	Ohm	1
C_{LOAD}	Crystal load capacitance	–	12	–	pF	

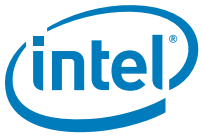


Table 21-9. Integrated Clock Crystal Specification

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
C_{SHUNT}	Crystal shunt capacitance	–	–	2	pF	1
C1, C2	Load Capacitance tolerance	-10	–	10	%	1

Note:
1. These are the specifications needed to select a crystal oscillator for the Integrated Clock circuit. Crystal must be AT cut, fundamental, parallel resonance.

21.6 DC Specifications

Platform reference voltages are specified at DC only. V_{REF} measurements should be made with respect to the supply voltages specified in Section 21.4, "Voltage and Current Specifications" on page 219.

Note: $V_{IH/OH}$ Maximum and $V_{IL/OL}$ Minimum values are bounded by reference voltages.

See the following DC Specifications in this section:

- Section 21.6.1, "Display DC Specification" on page 230
- Section 21.6.2, "MIPI*-Camera Serial Interface (CSI) DC Specification" on page 235
- Section 21.6.3, "SCC—SDIO DC Specification" on page 235
- Section 21.6.4, "SCC—SD Card DC Specification" on page 235
- Section 21.6.5, "eMMC* 4.51 DC Electrical Specification" on page 236
- Section 21.6.6, "JTAG DC Specification" on page 237
- Section 21.6.7, "DDR3L Memory Controller DC Specification" on page 238
- Section 21.6.8, "USB 2.0 Host DC Specification" on page 238
- Section , "" on page 240
- Section 21.6.11, "LPC DC Specification" on page 241
- Section 21.6.12, "PCU SPI DC Specification" on page 242
- Section 21.6.13, "Power Management/Thermal (PMC) and RTC DC Specification" on page 242
- Section 21.6.14, "SVID DC Specification" on page 244
- Section 21.6.15, "GPIO DC Specification" on page 245
- Section 21.6.16, "SIO-SPI DC Specifications" on page 245
- Section 21.6.17, "SIO—I2C DC Specification" on page 245
- Section 21.6.18, "SIO—UART DC Specification" on page 246
- Section 21.6.19, "I2S Audio DC Specification" on page 246
- Section 21-40, "HD Audio DC Specifications for 1.5V" on page 246
- Section 21.6.21, "SMBus (System Management) DC Specification" on page 247
- Section 21.6.22, "PCI Express* DC Specification" on page 247
- Section 21.6.23, "Serial ATA (SATA) DC Specification" on page 247

Note: Care should be taken to read all notes associated with each parameter.

21.6.1 Display DC Specification

DC specifications for display interfaces:

21.6.1.1 DisplayPort* DC Specification

Table 21-10. DisplayPort* 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	
$RL_{TX-DIFF}$	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.

21.6.1.2 HDMI DC Specification

Table 21-11. HDMI DC Specification

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
Voff	Single Ended Standby (off), output voltage	-10	–	10	mV	6 @ AVcc
Vswing	Single Ended output swing voltage	400	–	600	mV	
V_{OH} (≤165 MHz)	Single Ended high level, output voltage	-10	–	10	mv	6 @ AVcc
V_{OH}(>165 MHz)	Single Ended high level, output voltage	-200	–	10	mV	6 @ AVcc
V_{OL}(≤165 MHz)	Single Ended low level, output voltage	-600	–	-400	mV	6 @ AVcc
V_{OL}(>165 MHz)	Single Ended low level, output voltage	-700	–	-400	mV	6 @ AVcc
Note: 1. AVcc =Analog Voltage level						

21.6.1.3 embedded DisplayPort* DC Specification

Table 21-12. embedded Display Port* DC Specification

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
$V_{TX\text{-}DIFFp\text{-}p\text{-}Level0}$	Differential Peak-to-peak Output Voltage Level 0	0.18	0.2	0.22	V	1,2
$V_{TX\text{-}DIFFp\text{-}p\text{-}Level1}$	Differential Peak-to-peak Output Voltage Level 1	0.2	0.	0.275	V	1,2
$V_{TX\text{-}DIFFp\text{-}p\text{-}Level2}$	Differential Peak-to-peak Output Voltage Level 2	0.27	0.3	0.33	V	1,2
$VT_{X\text{-}DIFFp\text{-}p\text{-}Level3}$	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
$V_{TX\text{-}PREEMP\text{-}RATIO}$	No Pre-emphasis	0.0	0.0	0.0	dB	1
	3.5 dB Pre-emphasis	2.8	3.5	4.2	dB	1
	6.0 dB Pre-emphasis	4.8	6.0	7.2	dB	1
	9.5 dB Pre-emphasis	7.5	9.5	11.4	dB	1

Table 21-12. embedded 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	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; and so forth.
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.

21.6.1.4 DisplayPort* AUX Channel DC Specification

Table 21-13. DDI AUX Channel DC Specification

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
$V_{AUX-DIFFp-p}$	AUX Peak-to-peak Voltage at a transmitting Device	0.29	–	1.38	V	1
$V_{AUX-TERM\_R}$	AUX CH termination DC resistance	–	100	–	Ω	
$V_{AUX-DC-CM}$	AUX DC Common Mode Voltage	0	–	2.0	V	2
$V_{AUX-TURN-CM}$	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. $V_{AUX-DIFFp-p} = 2 * |V_{AUXP} - V_{AUXM}|$
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.

21.6.1.5 embedded Display Port* AUX Channel DC Specification

Table 21-14. embedded Display Port* AUX Channel DC Specification (Sheet 1 of 2)

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
$V_{AUX-DIFFp-p}$	AUX Peak-to-peak Voltage at a transmitting Device	0.29	–	1.38	V	1
$V_{AUX-\_TERM\_R}$	AUX CH termination DC resistance	–	100	–	Ω	
$V_{AUX-DC-CM}$	AUX DC Common Mode Voltage	0	–	1.2	V	2
$V_{AUX-TURN-CM}$	AUX turn around common mode voltage	–	–	0.3	V	3

Table 21-14. embedded Display Port* AUX Channel DC Specification (Sheet 2 of 2)

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
I_{AUX_SHORT}	AUX Short Circuit Current Limit	–	–	90	mA	4
C_{AUX}	AC Coupling Capacitor	75	–	200	nF	5

Notes:
1. $V_{AUX-DIFFp-p} = 2 * |V_{AUXP} - V_{AUXM}|$
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.

21.6.1.6 DDC Signal DC Specification

Table 21-15. DDC Signal DC Specification (DCC_DATA, DDC_CLK)

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
V_{REF}	I/O Voltage	MIPI_V1P8_S4			V	
V_{IH}	Input High Voltage	0.65 * V_{REF}	–	–	V	1
V_{IL}	Input Low Voltage	–	–	0.35 * V_{REF}	V	2
V_{OL}	Output Low Voltage	–	–	0.4	V	3
I_i	Input Pin Leakage	-30	–	30	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. 3mA sink current.
4. For VIN between 0V and CORE_VCC. Measured when driver is tri-stated.

Table 21-16. DDC Miscellaneous Signal DC Specification (HPD, BKLTCTL, VDDEN, BKLTEN)

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
V_{REF}	I/O Voltage	MIPI_V1P8_S4			V	
V_{IH}	Input High Voltage	0.65 * V_{REF}	–	–	V	1
V_{IL}	Input Low Voltage	0	–	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	-20	–	20	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 and CORE_VCC1.
4. For VIN between 0V and CORE_VCC0 and CORE_VCC1. Measured when driver is tri-stated.

Figure 21-1. Definition of Differential Voltage and Differential Voltage Peak-to-Peak

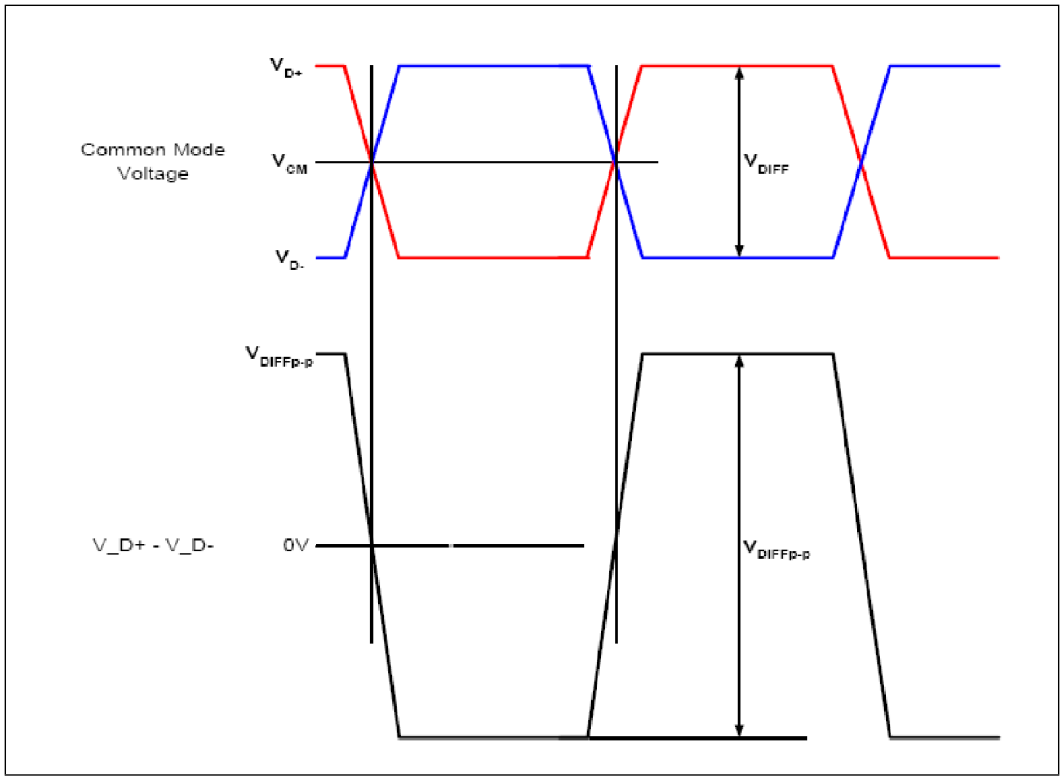
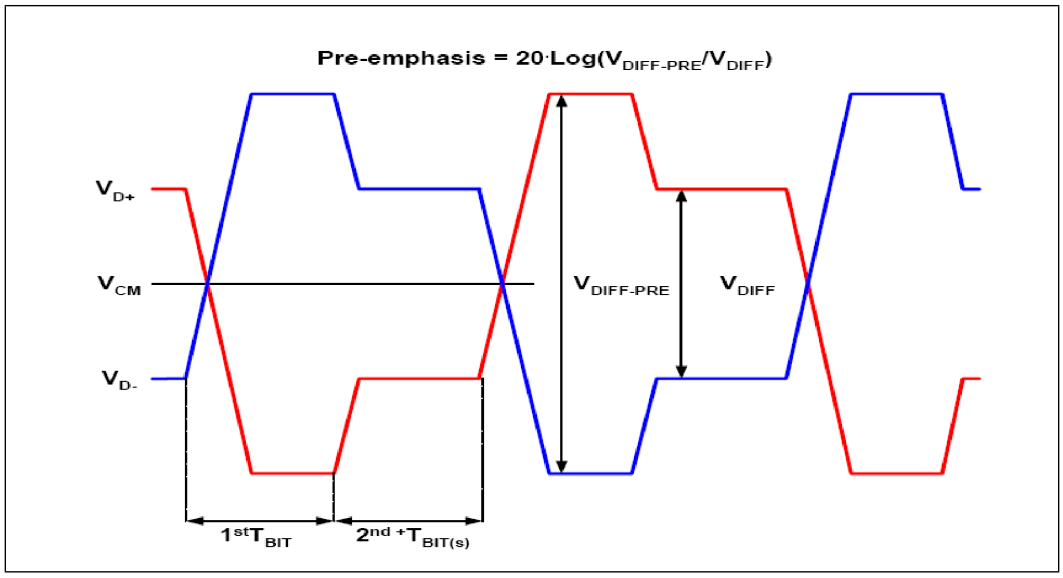


Figure 21-2. Definition of Pre-Emphasis



21.6.2 MIPI*-Camera Serial Interface (CSI) DC Specification

Table 21-17. 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	
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	1	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		–	–	mV	

21.6.3 SCC—SDIO DC Specification

Table 21-18 provides the SDIO DC Specification. For all other DC Specifications not listed in Table 21-18, refer to Table 21-37, "GPIO 1.8V Core Well Signal Group DC Specification.

Table 21-18. 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	

21.6.4 SCC—SD Card DC Specification

Table 21-19 provides the SD Card DC Specification. For all other DC Specifications not listed in Table 21-19, refer to Table 21-37, "GPIO 1.8V Core Well Signal Group DC Specification.

Table 21-19. SD Card DC Specification (Sheet 1 of 2)

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	–	0.1 * V_{REF}	V

Table 21-19. SD Card DC Specification (Sheet 2 of 2)

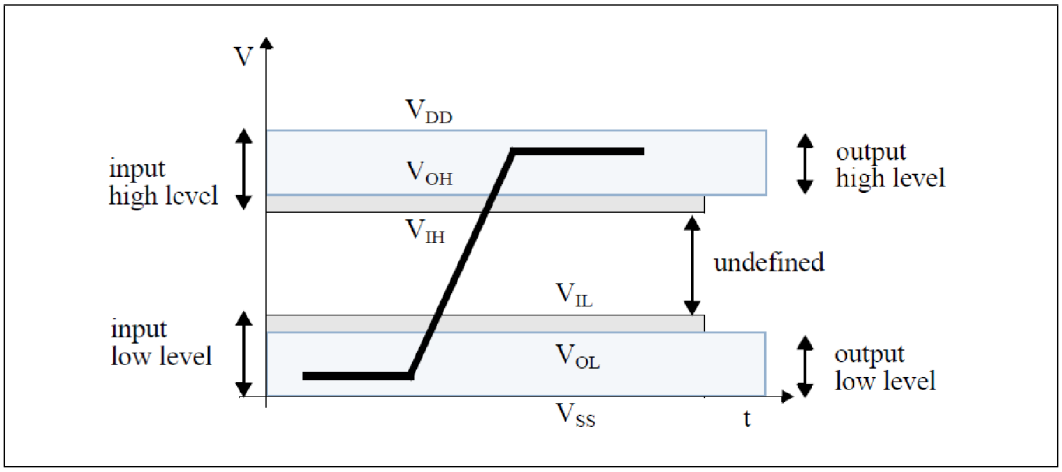
Symbol	Parameter	Min.	Max.	Unit
$V_{IH\ (3.3)}$	Input High Voltage (3.3V)	0.6 * V_{REF}	V_{REF}+0.3	V
$V_{IL\ (3.3)}$	Input Low Voltage (3.3V)	VSS-0.3	0.25 * V_{REF}	V
$V_{OH(1.8)}$	Output High Voltage	1.40	–	V
$V_{OL(1.8)}$	Output Low Voltage	–	0.45	V
$V_{IH\ (1.8)}$	Input High Voltage (1.8V)	1.27	2.00	V
$V_{IL\ (1.8)}$	Input Low Voltage (1.8V)	VSS-0.3	0.58	V
I_{OH}/I_{OL}	Current at VoL/Voh	-2	2	mA
C_{LOAD}	total Load Capacitance	–	40	pF

21.6.5 eMMC* 4.51 DC Electrical Specification

Table 21-20. eMMC* 4.51 DC Electrical Specifications

Symbol	Parameter	Min.	Max.	Units
V_{REF}	I/O Voltage	GPIO_V1P8A_G3		
V_{OH}	Output HIGH voltage	V_{REF} - 0.45	–	V
V_{Ol}	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	-2	2	µA
I_{0L}	Output Leakage Current	-2	2	µA

Figure 21-3. 4.51 DC Bus Signal Level



21.6.6 JTAG DC Specification

Table 21-21. JTAG Signal Group DC Specification (JTAG_TCK, JTAG_TMS, JTAG_TDI, JTAG_TRST_N)

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
V_{REF}	I/O Voltage	GPIO_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
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_20k and Rwpd_40k are only used for JTAG_TRST#.

Table 21-22. JTAG Signal Group DC Specification (JTAG_TDO)

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
V_{REF}	I/O Voltage	GPIO_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

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.

Table 21-23. JTAG Signal Group DC Specification (JTAG_PRDY#, JTAG_PREQ#)

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
V_{REF}	I/O Voltage	GPIO_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

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.

21.6.7 DDR3L Memory Controller DC Specification

Table 21-24. DDR3L Signal Group DC Specifications

Symbol	Parameter	Min.	Typ.	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 Clock Buffer strength	26	–	40	Ω	5
C_{IO}	DQ/DQS/ DQS# DDR3L I/O 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.
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_S4/2.
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-1333/1600: CLK buffer Ron is 26 ohm and SR target is 4V/ns; DQ-DQS buffer Ron is 30 ohms and SR target is 4V/ns; CMD/CTL buffer Ron is 20 Ohms and SR target is 1.8V/ns.

21.6.8 USB 2.0 Host DC Specification

Table 21-25. USB 2.0 Host DC Specification (Sheet 1 of 3)

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
Supply Voltage						
V_{BUS}	High-power Port	4.75	–	5.	V	2
V_{BUS}	Low-power Port	4.20	–	5.	V	
Supply Current						
I_{CCPRT}	High-power Hub Port (out)	500	–	–	mA	
I_{CCUPT}	Low-power Hub Port (out)	100	–	–	mA	
I_{CCHPF}	High-power Function (in)	–	–	500	mA	
I_{CCLPF}	Low-power Function (in)	–	–	100	mA	
I_{CCINIT}	Unconfigured Function/Hub (in)	–	–	100	mA	
I_{CCSH}	Suspended High-power Device	–	–	2.5	mA	15
I_{CCSL}	Suspended Low-power Device	–	–	500	µA	

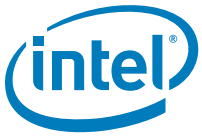


Table 21-25. USB 2.0 Host DC Specification (Sheet 2 of 3)

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
Input Levels for Low/Full-speed						
V_{IH}	High (driven)	2.0	–	–	V	4
V_{IHZ}	High (floating)	2.7	–	3.6	V	4
V_{IL}	Low		–	0.8	V	4
V_{DI}	Differential Input Sensitivity	0.2	–	–	V	|(D+)–(D-)|;Figure; **See Note** 4
V_{CM}	Differential Common Mode Range	0.8	–	2.5	V	Includes VDI range; Figure; **See Note** 4
Input Levels for High-speed						
V_{HSSQ}	High-speed squelch detection threshold (differential signal amplitude)	100	–	150	mV	
V_{HSDSC}	High speed disconnect detection threshold (differential signal amplitude)	525	–	625	mV	
	High-speed differential input signaling levels	–	–	–		16
V_{HSCM}	High-speed data signaling common mode voltage range (guideline for receiver)	-50	–	500	mV	
Output Levels for Low-/full-speed						
V_{OL}	Low	0.0	–	0.8	V	4,5
V_{OH}	High (Driven)	2.8	–	3.6	V	4,6
V_{OSE1}	SE1	0.8	–	–	V	
V_{CRS}	Output Signal Crossover Voltage	1.3	–	2.0	V	10
Output Levels for High-speed						
V_{HSOI}	High-speed idle level	-10	–	10	mV	
V_{HSOH}	High-speed data signaling high	360	–	440	mV	
V_{HSOL}	High-speed data signaling low	-10	–	10	mV	
V_{CHIRPJ}	Chirp J level (differential voltage)	700	–	1100	mV	
V_{CHIRPK}	Chirp K level (differential voltage)	-900	–	-500	mV	
Decoupling Capacitance						
C_{HPB}	Downstream Facing Port Bypass Capacitance (per hub)	120	–	–	µF	
C_{RPB}	Upstream Facing Port Bypass Capacitance	1.0	–	10.0	µF	9
Input Capacitance for Low-/Full-speed						
C_{IND}	Downstream Facing Port	–	–	150	pF	2
C_{INUB}	Upstream Facing Port (w/o cable)	–	–	100	pF	3
C_{EDGE}	Transceiver edge rate control capacitance	–	–	75	pF	

Table 21-25. USB 2.0 Host DC Specification (Sheet 3 of 3)

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
Input Impedance for High-speed						
	TDR specification for high-speed termination	–	–	–		
Terminations						
R_{PU}	Bus Pull-up Resistor on Upstream Facing Port	1.425	–	1.575	KΩ	1.5 KΩ ±5%
R_{PD}	Bus Pull-down Resistor on Downstream Facing Port	14.25	–	15.75	KΩ	1.5 KΩ ±5%
Z_{INP}	Input impedance exclusive of pull-up/ pull-down (for low-/full speed)	300	–	–	KΩ	
V_{TERM}	Termination voltage for upstream facing port pull-up (RPU)	3.0	–	3.6	V	
Terminations in High-speed						
V_{HSTERM}	Termination voltage in high speed	-10	–	10	mV	
R_{TERM}	High Speed Termination	40	45	50	Ω	
V_{BUSD}	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 1.4 KΩ to 3.6V.
6. Measured with RL of 14. 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 330mV.
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.

21.6.9 USB HSIC DC Specification

Table 21-26. USB HSIC DC Electrical Specification (Sheet 1 of 2)

Symbol	Parameter	Min.	Max.	Units
V_{REF}	I/O Voltage	USB_HSIC_V1.2V	–	
V_{OH}	Output HIGH voltage	0.75*Vref	–	V
V_{OL}	Output LOW voltage	–	$0.25*V_{REF}$	V
V_{IH}	Input HIGH voltage	0.65*Vref	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	Ohms

Table 21-26. USB HSIC DC Electrical Specification (Sheet 2 of 2)

Symbol	Parameter	Min.	Max.	Units
C_L	Load Capacitance	1	5	pF
Z_I	I/O Input Impedance	240	–	K Ohms
T_I	Characteristic Trace Impedance	45	55	Ohms

21.6.10 USB 3.0 DC Specification

Table 21-27. USB 3.0 DC Specification

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
UI	Unit Interval	199.94	–	200.06	ps	1
$V_{TX-DIFF-PP}$	Differential peak-peak Tx voltage swing	0.9	1	1.05	V	
$V_{TX-DIFF-PP-LOW}$	Low-Power Differential peak-peak Tx voltage swing	0.4	–	1.2	V	2
$V_{TX-DE-RATIO}$	Tx De-Emphasis	3.45	3.5	3.65	dB	
$R_{TX-DIFF-DC}$	DC differential impedance	88	–	92	Ω	
$V_{TX-RCV-DETECT}$	The amount of voltage change allowed during Receiver Detection	–	–	0.6	V	3
$C_{AC-COUPLING}$	AC Coupling Capacitor	75	–	200	nF	4
$t_{CDR\_SLEW\_MAX}$	Maximum slew rate	–	–	10	ms/s	

Notes:
1. The specified UI is equivalent to a tolerance of 300 ppm for each device. Period does not account for SSC induced variations.
2. There is no de-emphasis requirement in this mode. De-emphasis is implementation specific for this mode.
3. 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.
4. All transmitters shall be AC coupled. The AC coupling is required either within the media or within the transmitting component itself.

21.6.11 LPC DC Specification

Table 21-28. 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 21-29. LPC 3.3V Signal Group DC Specification

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
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.

21.6.12 PCU SPI DC Specification

Table 21-30. PCU SPI DC Specification

Symbol	Parameter	Min.	Max.	Units	Notes
V_{IH}	Input High Voltage	1.25	–	V	
V_{IL}	Input Low Voltage	–	0.693	V	
V_{OH}	Output High Voltage	1.17	–	V	
V_{OL}	Output Low Voltage	–	0.45	V	
I_{OH}	Output High Current	-3	3	mA	
I_{OL}	Output Low Current	-3	3	mA	
I_{LEAK}	Input Leakage Current	-2	2	µA	
C_{IN}	Input Capacitance	2	5	pF	

21.6.13 Power Management/Thermal (PMC) and RTC DC Specification

Table 21-31. Power Management 1.8V Suspend Well Signal Group DC Specification (Sheet 1 of 2)

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
V_{REF}	I/O Voltage	V1P8A			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
V_{OH}	Output High Voltage	V_{REF} – 0.45	–	1.8V	V	1
V_{OL}	Output Low Voltage	–	–	0.45	V	1

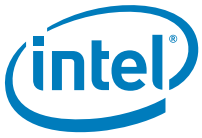


Table 21-31. Power Management 1.8V Suspend Well Signal Group DC Specification (Sheet 2 of 2)

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
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.

Table 21-32. PMC_RSTBTN# 1.8V Core Well Signal Group DC Specification

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
V_{REF}	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 21-33. 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 21-34. 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 21-35. PROCHOT# Signal Group DC Specification

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
V_{REF}	I/O Voltage		1p8V			
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.

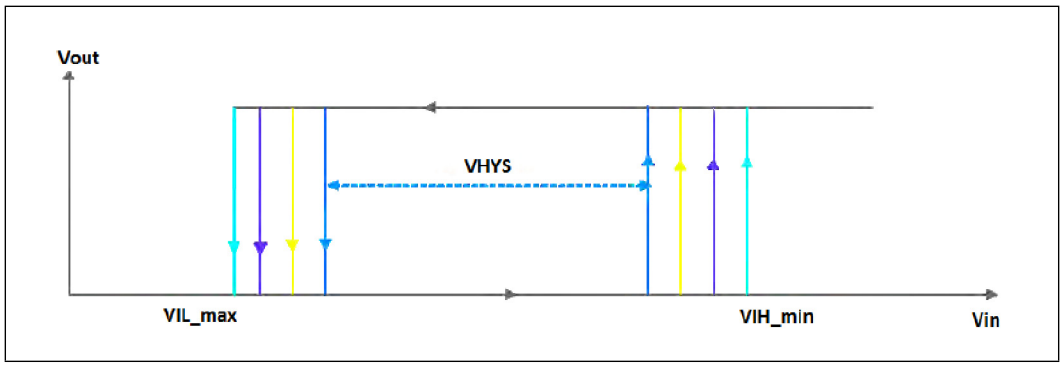
21.6.14 SVID DC Specification

Table 21-36. SVID Signal Group DC Specification (SVID_DATA, SVID_CLK, SVID_ALERT_N)

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
V_{REF}	I/O Voltage		V1P05A			
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.

Figure 21-4. Definition of VHYS in the DDR#L Interface Timing Specification



21.6.15 GPIO DC Specification

Table 21-37. GPIO 1.8V Core Well Signal Group DC Specification

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
V_{REF}	I/O Voltage	GPIO_V1P8A_G3			V	
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	-2	–	2	mA	
C_{LOAD}	Load Capacitance	2	–	75	pF	

21.6.16 SIO-SPI DC Specifications

Table 21-38. SIO SPI DC Specifications

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
V_{REF}	I/O Voltage	GPIO_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.

21.6.17 SIO—I^{2}C DC Specification

Table 21-39. I^{2}C Signal Electrical Specifications

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
V_{REF}	I/O Voltage	V1P8_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	

21.6.18 SIO—UART DC Specification

Refer to the GPIO Buffer (1.8V) DC Specification that is mentioned in Section 21.6.15, "GPIO DC Specification" on page 245.

21.6.19 I²S Audio DC Specification

Refer to the GPIO Buffer (1.8V) DC Specification that is mentioned in Section 21.6.15, "GPIO DC Specification" on page 245.

21.6.20 High Definition Audio DC Specifications

Table 21-40. HD Audio DC Specifications for 1.5V

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
V_{cc}	Supply voltage	1.418	–	1.583	V	
V_{IH}	Input High Voltage	0.6*Vcc	–	–	V	
V_{IL}	Input Low Voltage	–	–	0.4*Vcc	V	
V_{OH}	Output High Voltage	0.9*Vcc	–	–	V	1
V_{OL}	Output Low Voltage	–	–	0.10*Vcc	V	2
I_{IL}	Input Leakage Current	–	–	±10	uA	3, 4
C_{IN}	Input Pin Capacitance	–	–	7.5	pF	
L_{PIN}	Pin Inductance	–	–	20	nH	5

Notes:
1. At Iout = -500 µA
2. At Iout = 1500 µA
3. At 0<Vin<Vcc
4. For SDI (Serial Data In) buffers (or in general any bidirectional buffer with tri-state output), input leakage current also include hi-Z output leakage.
5. This is a recommendation, not an absolute requirement. The actual value should be provided with the component data sheet.

Table 21-41. HD Audio DC Specification for 1.8V

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
V_{cc}	Supply voltage	1.71	–	1.89	V	
V_{IH}	Input High Voltage	0.6 * Vcc	–	–	V	
V_{IL}	Input Low Voltage	–	–	0.35 * Vcc	V	
V_{OH}	Output High Voltage	0.9 * Vcc	–	–	V	1
V_{OL}	Output Low Voltage	–	–	0.10 * Vcc	V	2
I_{IL}	Input Leakage Current	–	–	±10	µA	3, 4
C_{IN}	Input Pin Capacitance	–	–	7.5	pF	
L_{PIN}	Pin Inductance	–	–	20	nH	5

Notes:
1. At Iout = -500 µA
2. At Iout = 1500 µA
3. At 0<Vin<Vcc
4. For **SDI (Serial Data In)** buffers (or in general any bidirectional buffer with tri-state output), input leakage current also include hi-Z output leakage.
5. This is a recommendation, not an absolute requirement. The actual value should be provided with the component data sheet.

21.6.21 SMBus (System Management) DC Specification

Table 21-42. SMBus DC Specification

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
V_{cc}	Supply voltage	–	–	1.89	V	
V_{IH}	Input High Voltage	2.1	–	5.5	V	
V_{IL}	Input Low Voltage	–	–	0.8	V	
V_{OH}	Output High Voltage	–	–	–	V	
V_{OL}	Output Low Voltage	–	–	0.4	V	1
I_{IL}	Input Leakage Current	-5	–	5	µA	
I_{PullUp}	Pull Up current	100	–	350	µA	
Note: 1. At Iout = 350 µA						

21.6.22 PCI Express* DC Specification

Table 21-43. PCI Express DC Receiver Signal Characteristics

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
V_{RXIDFF} (Gen1)	Differential RX Peak to Peak	175	–	1200	mV	1
V_{RXIDFF} (Gen1)	Differential RX Peak to Peak	100	–	1200	mV	1
Note: 1. PCI Express peak to peak = 2*[RXp[x] – RXn[x]]						

Table 21-44. PCI Express DC Transmit Signal Characteristics

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
V_{TXDIFF}	Differential TX Peak to Peak	800	–	1200	mV	1
$V_{TXDIFF-LP}$	Differential TX Peak to Peak (Low power mode)	400	–	1200	mV	1
Note: 1. PCI Express peak to peak = 2*[RXp[x] – RXn[x]]						

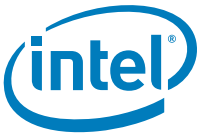
Table 21-45. PCI Express DC Clock Request Input Signal Characteristics

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
V_{REF}	I/O Voltage	UNCORE_V1P8_S4			V	
V_{IL}	Input Low Voltage	–	–	$0.3*V_{Ref}$	V	1
V_{IH}	Input High Voltage	$0.65*V_{Ref}$	–	–	V	

21.6.23 Serial ATA (SATA) DC Specification

Refer to the "SATA Revision 3.2" for the latest specification.

§ §



Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Intel:
FH8066501715924 SR2A8 FH8066501715923 SR2A7 FH8066501715925 SR2A9 FH8066501715915 SR29J FH8066501715913 SR29F FH8066501715914 SR29H

OCEAN CHIPS
Океан Электроники
Поставка электронных компонентов

Компания «Океан Электроники» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Поставка оригинальных импортных электронных компонентов напрямую с производств Америки, Европы и Азии, а так же с крупнейших складов мира;
- Широкая линейка поставок активных и пассивных импортных электронных компонентов (более 30 млн. наименований);
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Помощь Конструкторского Отдела и консультации квалифицированных инженеров;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Поставка электронных компонентов под контролем ВП;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- При необходимости вся продукция военного и аэрокосмического назначения проходит испытания и сертификацию в лаборатории (по согласованию с заказчиком);
- Поставка специализированных компонентов военного и аэрокосмического уровня качества (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Actel, Aeroflex, Peregrine, VPT, Syfer, Eurofarad, Texas Instruments, MS Kennedy, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits, General Dynamics и др.);

Компания «Океан Электроники» является официальным дистрибьютором и эксклюзивным представителем в России одного из крупнейших производителей разъемов военного и аэрокосмического назначения «**JONHON**», а так же официальным дистрибьютором и эксклюзивным представителем в России производителя высокотехнологичных и надежных решений для передачи СВЧ сигналов «**FORSTAR**».

AVIC JONHON

«JONHON» (основан в 1970 г.)

Разъемы специального, военного и аэрокосмического назначения:

(Применяются в военной, авиационной, аэрокосмической, морской, железнодорожной, горно- и нефтедобывающей отраслях промышленности)

«FORSTAR» (основан в 1998 г.)

ВЧ соединители, коаксиальные кабели, кабельные сборки и микроволновые компоненты:

(Применяются в телекоммуникациях гражданского и специального назначения, в средствах связи, РЛС, а так же военной, авиационной и аэрокосмической отраслях промышленности).

FORSTAR®

Телефон: 8 (812) 309-75-97 (многоканальный)
Факс: 8 (812) 320-03-32
Электронная почта: ocean@oceanchips.ru
Web: http://oceanchips.ru/
Адрес: 198099, г. Санкт-Петербург, ул. Калинина, д. 2, корп. 4, лит. А