Contents

Contents i

Figures ii

Tables iii

2 Overall Description 2-1

2.1 Chip Structure 2-1

2.1.1 SoC System 2-1

2.1.2 Media Subsystem 2-3

2.1.3 Storage Subsystem 2-3

2.2 Clock 2-4

2.2.1 Function Description 2-4

2.2.2 Clock Input 2-4

2.3 Reset 2-4

2.3.1 Function Description 2-4

2.3.2 Chipset Reset Scheme 2-4

2.3.3 Reset Structure 2-5

2.4 Interrupt 2-7

2.4.1 Function Description 2-7

2.4.2 Interrupt Structure 2-7

2.4.3 Interrupt Mapping 2-8

2.5 Chip Operating Mode and Control 2-14

2.6 Boot Mechanism 2-15

2.6.1 Overall Process 2-15

2.6.2 eMMC Boot 2-16

2.7 Debugging Mode 2-16

2.7.1 JTAG Debugging 2-16

2.7.2 CoreSight Debugging 2-17

2.8 Maintainability and Testability 2-17

2.9 Memory Map 2-17

2.9.1 Address Space Allocation (From the ACPU Perspective) 2-18

Figures

Figure 2-1 SoC system architecture ....**Error! Bookmark not defined.**

Figure 2-2 External reset 2-5

Figure 2-3 GIC architecture..2-8

Tables

Table 2-1 Modules in the SoC system 2-1

Table 2-2 Allocation table of GIC interrupts 2-8

Table 2-3 Register groups and memory address ranges (ACPU)...2-18

2 Overall Description

2.1 Chip Structure

2.1.1 SoC System

Table 2-1 Modules in the SoC system

Module	Description	Module	Description
A73/A53	Application processor cluster, big.LITTLE architecture	IVP	Image and video processor (IVP) module
ASP	Audio signal processor (ASP) subsystem	LPMCU	Low-power processing subsystem
BLPWM	Backlight pulse-width modulation (PWM) module	MMC	Multimedia card (MMC) control module
BOOTROM	On-chip read-only memory (ROM)	NANDC	Flash memory controller (FMC)
CODEC_SSI	Synchronous serial interface (SSI) module used to communicate with the codec	PCTRL	Peripheral controller
CRG	Clock and reset generator (CRG) module	PMCTRL	Power management control (PMC) module such as dynamic frequency scaling (DFS) and dynamic voltage and frequency scaling (DVFS)
CSSYS	Processor joint-debugging module	PMU_I2C	I^2C interface module used to communicate with the power management unit (PMU)

Module	Description	Module	Description
DDRC	Double data rate SDRAM controller (DDRC)	PMU_SSI	SSI module used to communicate with the PMU
DJTAG	JTAG port debugging module	PWM	PWM module
DMAC	Direct memory access controller (DMAC) module	RTC	Real-time clock (RTC) counter
DSS	Display module	SCI	SIM card controller
EFUSEC	eFUSE control module	SCTRL	System controller
Generic interrupt controller (GIC)	Processor interrupt processing module	SEC_P/SEC_S	Security processing module
GNSPWM	Universal PWM module	SPI	Serial peripheral interface (SPI) controller
GPIO	General-purpose input/output (GPIO) interface module	SYS_CNT	Processor-dedicated counter module
GPU	Media service processor	TIMER	Timing and counting module
HKADC_SSI	Bus interface module, used to read the converted digital data in the housekeeping analog-to-digital converter (HKADC) of the external PMU through the SSI	TSENSORC	TSensor controller
I^2C	I^2C controller	TZPC	Security signal allocation module
IOC	I/O control module	UART	Universal serial port module
IOMCU	Sensor-hub processing subsystem	USB3OTG	USB 3.0 controller module
IPC	Inter-core communication module	VENC/VDEC	Video encryption/decryption processing module
ISP	Image signal processing (ISP) module	WD	Watchdog counter
EMMC5.1	eMMC 5.1 controller module	UFS	Unified File system (UFS) controller module
SDIO	SDIO controller module	SD	SD card controller module

Module	Description	Module	Description
PCIe	PCIe controller module	-	-

The system on chip (SoC) system uses the multi-layer bus architecture. Each layer supports separate parallel access.

The bus architecture of the SoC system has the following features:

- Supports hardware coherency through coherency bus interconnection.
- Connects the independent buses from multiple layers to improve the bus bandwidth of the Hi3660 and provide excellent scalability.
- Uses the advanced eXtensible interface (AXI) as the high-speed data bus, supporting 128-bit or 64-bit width.
- Uses the 32-bit AXI as the low-speed data bus and configuration bus.

2.1.2 Media Subsystem

The media subsystem provides superior multimedia processing and acceleration functions:

- Image capturing
- Image display and output
- Acceleration of image encoding and decoding
- 3D graphics acceleration
- Audio processing acceleration

The media subsystem supports the following upper-level applications:

- Digital photographing
- Digital video recording
- Audio recording
- Playing local audio and video
- Browsing local pictures
- Playing audio and video in the stream media format
- Multimedia editor
- Video call
- UI and video hardware acceleration
- Hardware acceleration for gaming

NOTE

For details about the media subsystem, see chapter 6 "Media Processing."

2.1.3 Storage Subsystem

The Hi3660 supports the following external storage interfaces to provide flexible storage solutions for the system and meet different product requirements:

- UFS/eMMC/SD/SDIO static storage card interface
- Low-power double data rate 4 (LPDDR4) dynamic memory interface

- NAND flash interface

NOTE

For details about the storage subsystem, see chapter 7 "Storage Control."

2.2 Clock

2.2.1 Function Description

The Hi3660 accepts external clock inputs, generates required internal operating clocks by using the internal phase-locked loops (PLLs) and clock circuits, and provides multiple clocks for other chips.

The Hi3660 provides 11 internal PLLs for generating operating clocks required by chip modules.

2.2.2 Clock Input

The Hi3660 supports the following external input clocks:

- 32 kHz clock
- 19.2 MHz clock
- External backup clock

2.3 Reset

2.3.1 Function Description

The Hi3660 receives external reset inputs and resets or deasserts reset on internal modules based on the reset deassertion sequence during power-on.

After the AO area is powered on, the internal power-on reset (POR) module outputs low-level signals, resets the Hi3660, and pulls the output level up about 12–48 ms after the power-on. This ensures that the entire chip is in reset state when the I/O (external input reset) is in indefinite state.

Besides POR, the Hi3660 supports the following global reset types:

- Watchdog reset
- Temperature sensor reset
- Chip soft reset

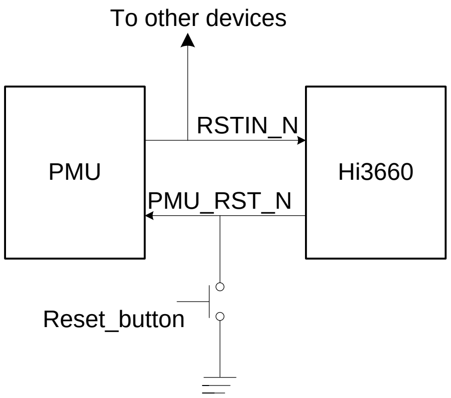
When any of the preceding reset types is valid, the global reset of the Hi3660 is triggered. All reset types have the same priority.

2.3.2 Chipset Reset Scheme

During the power-on process, the PMU provides the reset input signal (RSTIN_N) for the Hi3660. The SoC automatically performs the POR operation on internal modules.

When any global reset mode in the Hi3660 is valid, the Hi3660 outputs the PMU reset signal (PMU_RST_N) to reset the PMU. After that, the PMU pulls RSTIN_N down to implement global reset.

Figure 2-1 External reset



2.3.3 Reset Structure

The Hi3660 supports the following global reset types:

- POR
- Watchdog reset
- Temperature sensor reset
- Chip soft reset

There are two reset modes: dying gasp reset and non-dying gasp reset. Setting SCPEREN1[31] to 1 enables the dying gasp reset mode. Setting SCPERDIS1[31] to 1 disables the dying gasp reset mode.

- Dying gasp reset: When the wd reset request and over-temperature reset request are valid, the request for the DDR SDRAM to enter the self-refresh mode is initiated first. Then there are two options:
  - Wait for the DDR SDRAM to enter the self-refresh mode. (After receiving the interrupt, the software protects the scene, saves the critical system information to the external non-volatile memory or SDRAM, and then sets the SDRAM to the self-refresh mode.)
  - Send the reset request signal, pull the PMU reset signal down, and reset the entire system 1 ms after timeout occurs.
- Non-dying gasp reset: When the wd reset request, over-temperature reset request, and software reset request are valid, directly send the reset request signal, pull the PMU reset signal down, and reset the entire system. The scene is not preserved in the entire process.

POR

The global POR is obtained after two reset signals are ANDed: external reset signal RSTIN_N and the POR output reset signal. After the AO area is powered on, the internal POR signal outputs low level, resets the entire chip, and deasserts reset within 12–48 ms. The PMU generates the external reset signal RSTIN_N. The chip maintains the reset state after the I/O is powered on. The global reset of the chip can be deasserted after the PMU deasserts reset.

Watchdog Reset

The following subsystems provide the watchdogs:

- WD0, WD1, and LPMCU subsystems
- IOMCU subsystem
- Modem subsystem
- ASP subsystem
- IVP subsystem
- ISPA7 subsystem
- UCE
- OCBC
- GPU
- LITTLE core
- Big core

The watchdog module monitors the system running status. In normal cases, the system needs to periodically set the initial count value. If the system does not promptly set the initial count value, the software is running abnormally. In this case, the watchdog performs the following operations:

- The watchdog reports an exception interrupt, loads the initial value of the counter, and re-counts from the initial value.
- If the exception interrupt is not handled, a reset signal is initiated when the watchdog counter is decremented to 0.

Temperature Sensor Reset

The A53, A73, and G3D areas each contains a temperature sensor. When the chip temperature reaches the preset threshold, a reset request signal is initiated to reset the A53 and A73 cores in the chip.

Chip Soft Reset

The software can soft-reset the Hi3660 when necessary. When the software writes to the SCSYSSTAT register, global soft reset of the Hi3660 is triggered.

CAUTION

Soft reset can be performed only in non-dying gasp reset mode. In dying gasp reset mode, writing to SCSYSSTAT does not trigger the global soft reset of the Hi3660.

Module Soft Reset

The software can independently reset the major modules of the Hi3660. These modules include the RTC, timer, GPIO, USB, DMAC, VENC, VDEC, DSS, ISP, ASP, DDRC, MMC, PWM, UART, SPI, and G3D. For details, see the description of each module.

2.4 Interrupt

2.4.1 Function Description

The ACPU uses the GIC to handle and control interrupts. Other microcontrollers, media, and communication processors have their own interrupt handling logic. This section describes the basic interrupt handling functions of the GIC.

The GIC has the following basic features:

- Supports interrupt nesting for the A53, A73, and G3D.
- Manages multi-core interrupt distribution.
- Supports security extension.
- Queries the states of interrupt sources.
- Provides a unique ID for each interrupt.
- Supports configurable interrupt trigger mode: high-level-triggered mode or edge-triggered mode.
- Sets the priority of each interrupt.
- Generates software interrupts.

The GIC supports the following interrupt types:

- Software-generated interrupt (SGI)

  The GIC supports 16 SGIs (SCI0 to SCI15), which are controlled by writing to registers.
- Private peripheral interrupt (PPI)

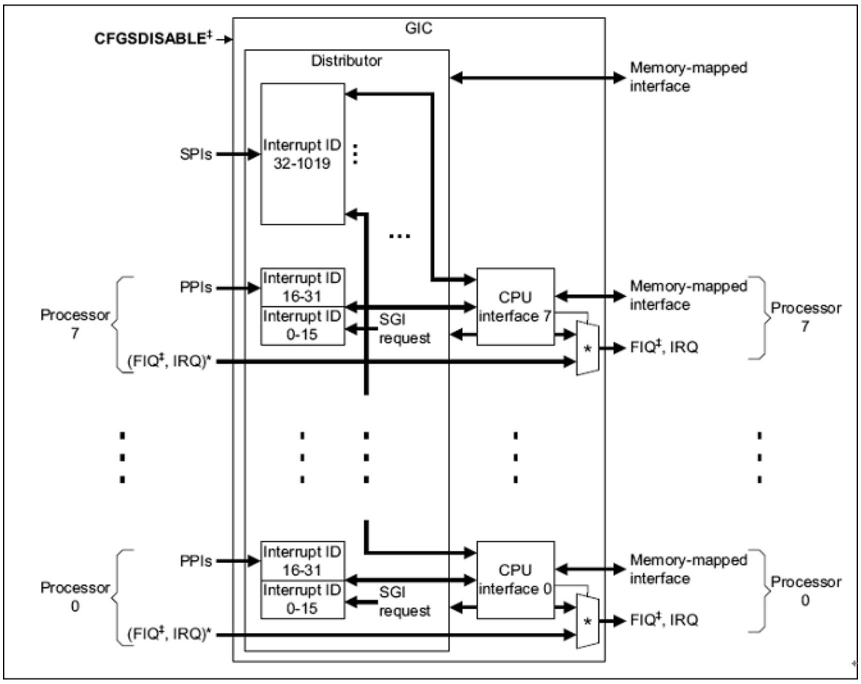
  Each processor corresponds to seven PPIs.
- Shared peripheral interrupt (SPI)

  A total of 352 peripheral interrupts are supported.

2.4.2 Interrupt Structure

As shown in Figure 2-2, the GIC contains one distributor and multiple CPU interfaces. The distributor manages all interrupts in a centralized manner, and the CPU interfaces implement interaction between the interrupts and the CPU. The integrated GIC-400 of the Hi3660 has eight CPU interfaces, which connect to the quad-core A73 and quad-core A53, respectively.

Figure 2-2 GIC architecture



The GIC supports the TrustZone. Each interrupt source can be configured as a secure interrupt source or a non-secure interrupt source.

The GIC configures the priority of each interrupt. A smaller interrupt priority value indicates a higher priority. If two interrupts have the same priority, the interrupt with a smaller interrupt ID takes priority over the other one.

2.4.3 Interrupt Mapping

The Hi3660 GIC supports 384 interrupts, including 352 peripheral interrupts. Table 2-2 lists the interrupt sources and interrupt IDs.

Table 2-2 Allocation table of GIC interrupts

Interrupt Source	GIC Interrupt ID	Interrupt Source	GIC Interrupt ID
A73_interr	32	PMC-AVS-IDLE-G3D	205
A73_exterr	33	M3_LP_wd	206
A73_pmu0	34	~CCI400_err	207
A73_pmu1	35	~&CCI400_overflow[6:0]	208
A73_pmu2	36	~CCI400_overflow[7]	209
A73_pmu3	37	IPC_S_int0	210
A73_cti0	38	IPC_S_int1	211
A73_cti1	39	IPC_S_int4	212

Interrupt Source	GIC Interrupt ID	Interrupt Source	GIC Interrupt ID
A73_cti2	40	IPC_S_mbx0	213
A73_cti3	41	IPC_S_mbx1	214
A73_COMMRX0	42	IPC_S_mbx2	215
A73_COMMRX1	43	IPC_S_mbx3	216
A73_COMMRX2	44	IPC_S_mbx4	217
A73_COMMRX3	45	IPC_S_mbx5	218
A73_COMMTX0	46	IPC_S_mbx6	219
A73_COMMTX1	47	IPC_S_mbx7	220
A73_COMMTX2	48	IPC_S_mbx8	221
A73_COMMTX3	49	IPC_S_mbx9	222
A73_COMMIRQ0	50	IPC_S_mbx18	223
A73_COMMIRQ1	51	IPC_NS_int0	224
A73_COMMIRQ2	52	IPC_NS_int1	225
A73_COMMIRQ3	53	IPC_NS_int4	226
A53_interr	54	IPC_NS_int5	227
A53_exterr	55	IPC_NS_int6	228
A53_pmu0	56	IPC_NS_mbx0	229
A53_pmu1	57	IPC_NS_mbx1	230
A53_pmu2	58	IPC_NS_mbx2	231
A53_pmu3	59	IPC_NS_mbx3	232
A53_cti0	60	IPC_NS_mbx4	233
A53_cti1	61	IPC_NS_mbx5	234
A53_cti2	62	IPC_NS_mbx6	235
A53_cti3	63	IPC_NS_mbx7	236
A53_COMMRX0	64	IPC_NS_mbx8	237
A53_COMMRX1	65	IPC_NS_mbx9	238
A53_COMMRX2	66	IPC_NS_mbx18	239
A53_COMMRX3	67	mdm_aximon_intr	240
A53_COMMTX0	68	MDM_WDOG_intr	241
A53_COMMTX1	69	ASP-IPC-ARM	242

Interrupt Source	GIC Interrupt ID	Interrupt Source	GIC Interrupt ID
A53_COMMTX2	70	ASP-IPC-MCPU	243
A53_COMMTX3	71	ASP-IPC-BBE16	244
A53_COMMIRQ0	72	ASP_WD	245
A53_COMMIRQ1	73	ASP_AXI_DLOCK	246
A53_COMMIRQ2	74	ASP_DMA_SECURE	247
A53_COMMIRQ3	75	ASP_DMA_SECURE_N	248
WatchDog0	76	SCI0	249
WatchDog1	77	SCI1	250
RTC0	78	SOCP0	251
RTC1	79	SOCP1	252
TIME00	80	MDM_IPF_intr0	253
TIME01	81	MDM_IPF_intr1	254
TIME10	82	|ddrc_fatal_int[3:0]	255
TIME11	83	mdm_axi_dlock_int	256
TIME20	84	mdm_wdt1_intr (CDSP)	257
TIME21	85	~GIC_IRQ_OUT[0]	258
TIME30	86	~GIC_IRQ_OUT[1]	259
TIME31	87	~GIC_IRQ_OUT[2]	260
TIME40	88	~GIC_IRQ_OUT[3]	261
TIME41	89	~GIC_IRQ_OUT[4]	262
TIME50	90	~GIC_IRQ_OUT[5]	263
TIME51	91	~GIC_IRQ_OUT[6]	264
TIME60	92	~GIC_IRQ_OUT[7]	265
TIME61	93	~GIC_FIQ_OUT[0]	266
TIME70	94	~GIC_FIQ_OUT[1]	267
TIME71	95	~GIC_FIQ_OUT[2]	268
TIME80	96	~GIC_FIQ_OUT[3]	269
TIME81	97	~GIC_FIQ_OUT[4]	270
TIME90	98	~GIC_FIQ_OUT[5]	271
TIME91	99	~GIC_FIQ_OUT[6]	272

Interrupt Source	GIC Interrupt ID	Interrupt Source	GIC Interrupt ID
TIME100	100	~GIC_FIQ_OUT[7]	273
TIME101	101	NANDC	274
TIME110	102	CoreSight_ETR_Full	275
TIME111	103	CoreSight_ETF_Full	276
TIME120	104	DSS-pdp	277
TIME121	105	DSS-sdp	278
UART0	106	DSS-offline	279
UART1	107	DSS_mcu_pdp	280
UART2	108	DSS_mcu_sdp	281
UART4	109	DSS_mcu_offline	282
UART5	110	DSS_dsi0	283
UART6	111	DSS_dsi1	284
SPI1	112	IVP32_SMMU_irpt_s	285
I^2C3	113	IVP32_SMMU_irpt_ns	286
I^2C4	114	IVP32_WATCH_DOG	287
I^2C5 (PMU_I2C)	115	ATGC	288
GPIO0_INTR1	116	G3D_IRQEVENT	289
GPIO1_INTR1	117	G3D_JOB	290
GPIO2_INTR1	118	G3D_MMU	291
GPIO3_INTR1	119	G3D_GPU	292
GPIO4_INTR1	120	isp_irq[0]	293
GPIO5_INTR1	121	isp_irq[1]	294
GPIO6_INTR1	122	isp_irq[2]	295
GPIO7_INTR1	123	isp_irq[3]	296
GPIO8_INTR1	124	isp_irq[4]	297
GPIO9_INTR1	125	isp_irq[5]	298
GPIO10_INTR1	126	isp_irq[6]	299
GPIO11_INTR1	127	isp_irq[7]	300
GPIO12_INTR1	128	isp_a7_to_gic_mbx_int[0]	301
GPIO13_INTR1	129	isp_a7_to_gic_mbx_int[1]	302

Interrupt Source	**GIC Interrupt ID**	**Interrupt Source**	**GIC Interrupt ID**
GPIO14_INTR1	130	isp_a7_to_gic_ipc_int	303
GPIO15_INTR1	131	isp_a7_watchdog_int	304
GPIO16_INTR1	132	isp_axi_dlcok	305
GPIO17_INTR1	133	isp_a7_irq_out	306
GPIO18_INTR1	134	ivp32_dwaxi_dlock_irq	307
GPIO19_INTR1	135	mmbuf_asc0	308
GPIO20_INTR1	136	mmbuf_asc1	309
GPIO21_INTR1	137	UFS	310
GPIO22_INTR1	138	pcie_link_down_int	311
GPIO23_INTR1	139	pcie_edma_int	312
GPIO24_INTR1	140	pcie_pm_int	313
GPIO25_INTR1	141	pcie_radm_inta	314
GPIO26_INTR1	142	pcie_radm_intb	315
GPIO27_INTR1	143	pcie_radm_intc	316
IOMCU_WD	144	pcie_radm_intd	317
IOMCU_SPI	145	psam_intr[0]	318
IOMCU_UART3	146	psam_intr[1]	319
IOMCU_UART8	147	ocbc_pe_npint[0]	320
IOMCU_SPI2	148	intr_wdog_ocbc	321
IOMCU_I2C3	149	intr_vdec_mfde_norm	322
IOMCU_I2C0	150	intr_vdec_scd_norm	323
IOMCU_I2C1	151	intr_vdec_bpd_norm	324
IOMCU_I2C2	152	intr_vdec_mmu_norm	325
IOMCU_GPIO0_INT1	153	intr_vdec_mfde_safe	326
IOMCU_GPIO1_INT1	154	intr_vdec_scd_safe	327
IOMCU_GPIO2_INT1	155	intr_vdec_bpd_safe	328
IOMCU_GPIO3_INT1	156	intr_vdec_mmu_safe	329
IOMCU_DMAC_INT0	157	intr_venc_vedu_norm	330
IOMCU_DMAC_NS_INT0	158	intr_venc_mmu_norm	331
PERF_STAT	159	intr_venc_vedu_safe	332

Interrupt Source	GIC Interrupt ID	Interrupt Source	GIC Interrupt ID
IOMCU_COMB	160	intr_venc_mmu_safe	333
IOMCU_BLPWM	161	intr_qosbuf0	334
NOC-comb	162	intr_qosbuf1	335
intr_dmss	163	intr_ddrc2_err	336
intr_ddrc0_err	164	intr_ddrc3_err	337
intr_ddrc1_err	165	intr_ddrphy[0]	338
PMCTRL	166	intr_ddrphy[1]	339
SECENG_P	167	intr_ddrphy[2]	340
SECENG_S	168	intr_ddrphy[3]	341
EMMC51	169	intr0_mdm_ipc_gic_s	342
ASP_IPC_MODEM_CBBE	170	intr1_mdm_ipc_gic_s	343
SD3	171	SPI3	344
SDIO	172	SPI4 (Finger/Ink screen)	345
GPIO28_INTR1	173	I2C7	346
PERI_DMAC_int0	174	intr_uce0_wdog	347
PERI_DMAC_NS_int0	175	intr_uce1_wdog	348
CLK_MONITOR (in SCTRL)	176	intr_uce2_wdog	349
TSENSOR_A73	177	intr_uce3_wdog	350
TSENSOR_A53	178	intr_exmbist	351
TSENSOR_G3D	179	intr_hisee_wdog	352
TSENSOR_Modem	180	intr_hisee_ipc_mbx_gic[0]	353
ASP_ARM_SECURE (asp_hmdi secure interrupt and src_up secure interrupt)	181	intr_hisee_ipc_mbx_gic[1]	354
ASP_ARM (asp_hmdi non-secure interrupt, src_up non-secure interrupt, and slimbus combined interrupt)	182	intr_hisee_ipc_mbx_gic[2]	355
VDM_INT2	183	intr_hisee_ipc_mbx_gic[3]	356

Interrupt Source	GIC Interrupt ID	Interrupt Source	GIC Interrupt ID
VDM_INT0	184	intr_hisee_ipc_mbx_gic[4]	357
VDM_INT1	185	intr_hisee_ipc_mbx_gic[5]	358
|{MODEM_IPC0[0], MDM_IPC_APPCPU_intr0}	186	intr_hisee_ipc_mbx_gic[6]	359
|{MODEM_IPC1[0], MDM_IPC_APPCPU_intr1}	187	intr_hisee_ipc_mbx_gic[7]	360
MDM_bus_err	188	intr_hisee_alarm[0]	361
Reserved	189	intr_hisee_alarm[1]	362
MDM_EDMAC0_INTR_NS[0]	190	Reserved	363
USB3	191	Reserved	364
Reserved	192	intr_hisee_eh2h_slv	365
USB3_OTG	193	intr_hisee_as2ap_irq	366
USB3_BC	194	intr_hisee_ds2ap_irq	367
GPIO1_SE_INTR1	195	intr_hisee_senc2ap_irq	368
GPIO0_SE_INTR1	196	GPIO0_EMMC	369
PMC-DVFS-A73	197	GPIO1_EMMC	370
PMC-DVFS-A53	198	AONOC_TIMEOUT	371
PMC-DVFS-G3D	199	intr_hisee_tsensor[0]	372
PMC-AVS-A73	200	intr_hisee_tsensor[1]	373
PMC-AVS-A53	201	intr_hisee_lockup	374
PMC-AVS-G3D	202	intr_hisee_dma	375
PMC-AVS-IDLE-A73	203	Reserved	376~383
PMC-AVS-IDLE-A53	204		

2.5 Chip Operating Mode and Control

The Hi3660 system supports four operating modes, which are controlled by configuring SCCTRL[modectrl] (0xFFF0_A000 for LPMCU access and 0x4020_A000 for CPU access):

- 000: The system mode is switched to sleep mode.

- 001: The system mode is switched to doze mode.
- 01*X*: The system mode is switched to slow mode.
- 1*XX*: The system mode is switched to normal mode.

After POR, the state machine is in slow mode by default. The software controls the system state transition by configuring SCCTRL[modectrl] (0xFFF0_A000 for LPMCU access and 0x4020_A000 for CPU access).

The system state machine is restored to slow mode after a global reset such as the global soft reset, watchdog reset, or Tsensor over-temperature reset.

2.6 Boot Mechanism

2.6.1 Overall Process

The Hi3660 supports two boot modes: USB loading mode and memory boot mode. The memory boot modes include eMMC boot mode and UFS boot mode. All these modes are booted by the BOOTROM in the Hi3660. Then the corresponding boot process is started.

Apart from the preceding common boot modes, the Hi3660 also supports the NAND boot and UFS boot in test mode.

Pin Settings

For the UFS boot mode booted by BOOTROM, the pin settings are as follows:

- TEST_MODE: 0
- BOOT_MODE: 1
- BOOT_UFS: 1

Basic Process

Step 1 Power on the system to start POR.

Step 2 Judge the boot mode.

Execute the BOOTROM code. Read the boot_mode register to judge the boot mode. If the value of the BOOT_MODE pin is 1, enter the memory boot branch. Then read the boot_ufs register. If the value of the BOOT_UFS pin is 1, enter the UFS boot process.

Step 3 Initialize the UFS clock and IP.

Step 4 Start the UFS link startup process.

Step 5 Initialize the UFS parameters and components.

Step 6 Transfer the bootloader image in the UFS device to the RAM.

Step 7 Verify the security.

Step 8 Execute the bootloader.

Step 9 Initialize the DDR, and copy the fastboot images stored in the UFS device to the DDR. Initialize the ACPU and deassert reset. Then the ACPU side starts executing fastboot and performs the subsequent startup process.

----End

2.6.2 eMMC Boot

Pin Settings

For the eMMC boot mode booted by BOOTROM, the pin settings are as follows:

- TEST_MODE: 0
- BOOT_MODE: 1
- BOOT_UFS: 0

Basic Process

Step 1 Power on the system to start POR.

Step 2 Judge the boot mode.

Execute the BOOTROM code. Read the boot_mode register to judge the boot mode. If the value of the BOOT_MODE pin is 1, enter the memory boot branch. Then read the boot_ufs register. If the value of the BOOT_UFS pin is 0, enter the eMMC boot process.

Step 3 Initialize the eMMC clock and IP.

Step 4 Transfer the images in the eMMC device to the buffer in the eMMC and then to the RAM.

Step 5 Verify the security.

Step 6 Execute the bootloader.

Step 7 Initialize the DDR, and copy the fastboot images stored in the eMMC device to the DDR. Initialize the ACPU and deassert reset. Then the ACPU side starts executing fastboot and performs the subsequent startup process.

----End

2.7 Debugging Mode

2.7.1 JTAG Debugging

The Hi3660 provides the JTAG interface that complies with the IEEE 1149.1 standard:

- The DSP simulator can debug the four internal DSPs.
- The PC can connect to the JTAG simulator to separately debug the ARM processor.

The JTAG MUX connects external JTAG pin signals to the cores of the Hi3660.

The debugging steps are as follows:

Step 1 Power on and reset the Hi3660.

Step 2 Set JTAG_SEL1 and JTAG_SEL0 to 2'b01 to multiplex the CPU JTAG function on the JTAG pin.

Step 3 Set JTAG_SEL1 and JTAG_SEL0 to 2'b00 to enter the register selection mode. Configure the system control register JTAGSYS_SW_SEL [7:0] to switch to the selected debugging interface for debugging.

Step 4 Connect the corresponding simulator and open the corresponding debugging software to start debugging.

----End

2.7.2 CoreSight Debugging

The Hi3660 has a powerful debug system that integrates an ARM CoreSight system. The CoreSight system supports the following features:

- Top-level CoreSight and local CoreSight in each cluster. The local CoreSight contains the A73 CoreSight and A53 CoreSight.
- Intrusive debugging (debug) and non-intrusive debugging (trace)

  A73 and A53 support both debug and trace.
- Software debugging and traditional JTAG debugging

2.8 Maintainability and Testability

The Hi3660 provides the following **maintainability and testability** means:

- JTAG debugging

2.9 Memory Map

The Hi3660 supports the 8-/6-/4-GB DDR storage solution. The system address space varies according to the capacity of the connected DDR and the processor perspective. The general principles are as follows:

- When the 4-GB component is connected, in the unified addressing space of the entire chip system viewed from the perspective of the ACPU, IVP, GPU, VENC, VDEC, DSS, and ISP:
  - The 0–3.5 GB and 4–4.5 GB address space is specified as the accessible 4-GB DRAM space.
  - The 3.5–4 GB address space viewed from the perspective of these masters is the register space.

  Only the 0–3.5 GB DRAM space and the 3.5–4 GB peripheral space are accessible.
- When the 8-GB component is connected, in the unified addressing space of the entire chip system viewed from the perspective of the A53, A73, IVP, GPU, VENC, VDEC, DSS, and ISP:
  - The 0–3.5 GB and 4–8.5 GB address space is specified as the accessible 8-GB DRAM space.
  - The 3.5–4 GB address space viewed from the perspective of these masters is the register space.

The modem and peripheral subsystems (including the IOMCU, LPMCU, ASP, DMAC, USB3OTG, SECENG, and MMC) can access only the 0–3.5 GB DRAM space and the 3.5–4 GB peripheral space.

- When the 6-GB DDR or DDR with other capacity is connected, the address mapping solution is similar to that when the 8-GB DDR is connected. The 3.5–4 GB space is used as the peripheral space.

2.9.1 Address Space Allocation (From the ACPU Perspective)

CAUTION

To prevent unpredictable results, do not access the address space marked with "Reserved".

Table 2-3 lists all the register groups and memory address ranges visible to the Hi3660 ACPU.

Table 2-3 Register groups and memory address ranges (ACPU)

Start Address	End Address	Size (Byte)	Module
0xFFF38000	0xFFF38FFF	4K	PMU_SSI2
0xFFF36000	0xFFF36FFF	4K	PMU_SSI1
0xFFF35000	0xFFF35FFF	4K	PERI_CRG
0xFFF34000	0xFFF34FFF	4K	PMU_SSI0
0xFFF33000	0xFFF33FFF	4K	PMU_I2C
0xFFF32000	0xFFF32FFF	4K	UART6
0xFFF31000	0xFFF31FFF	4K	PMCTRL
0xFFF30000	0xFFF30FFF	4K	TSENSORC
0xFFF20000	0xFFF2FFFF	64K	Reserved
0xFFF1F000	0xFFF1FFFF	4K	Reserved
0xFFF1D000	0xFFF1DFFF	4K	GPIO28
0xFFF1C000	0xFFF1CFFF	4K	TIMER8
0xFFF1B000	0xFFF1BFFF	4K	TIMER7
0xFFF1A000	0xFFF1AFFF	4K	TIMER6
0xFFF19000	0xFFF19FFF	4K	TIMER5
0xFFF18000	0xFFF18FFF	4K	TIMER4
0xFFF17000	0xFFF17FFF	4K	TIMER3
0xFFF16000	0xFFF16FFF	4K	TIMER2
0xFFF15000	0xFFF15FFF	4K	TIMER1

Start Address	End Address	Size (Byte)	Module
0xFFF14000	0xFFF14FFF	4K	TIMER0
0xFFF11000	0xFFF11FFF	4K	AO_IOC
0xFFF10000	0xFFF10FFF	4K	GPIO27
0xFFF0F000	0xFFF0FFFF	4K	GPIO26
0xFFF0E000	0xFFF0EFFF	4K	GPIO25
0xFFF0D000	0xFFF0DFFF	4K	GPIO24
0xFFF0C000	0xFFF0CFFF	4K	GPIO23
0xFFF0B000	0xFFF0BFFF	4K	GPIO22
0xFFF0A000	0xFFF0AFFF	4K	SCTRL
0xFFF08000	0xFFF09FFF	8K	SYS_CNT
0xFFF05000	0xFFF05FFF	4K	RTC1
0xFFF04000	0xFFF04FFF	4K	RTC0
0xFFD00000	0xFFD7FFFF	512K	IOMCU
0xFF400000	0xFFCFFFFF	9M	Reserved
0xFF3FF000	0xFF3FFFFF	4K	SDIO0
0xFF3FE000	0xFF3FEFFF	4K	PCIE_APB_CFG
0xFF3FD000	0xFF3FDFFF	4K	IOC_MMC1
0xFF3FC000	0xFF3FCFFF	4K	Reserved
0xFF3FB000	0xFF3FBFFF	4K	EMMC
0xFF3E2000	0xFF3FAFFF	100K	Reserved
0xFF3E1000	0xFF3E1FFF	4K	GPIO1_MMC1
0xFF3E0000	0xFF3E0FFF	4K	GPIO0_MMC1
0xFF3B8000	0xFF3DFFFF	160K	Reserved
0xFF3B7000	0xFF3B7FFF	4K	Reserved
0xFF3B6000	0xFF3B6FFF	4K	IOC_FIX
0xFF3B5000	0xFF3B5FFF	4K	GPIO19
0xFF3B4000	0xFF3B4FFF	4K	GPIO18
0xFF3B3000	0xFF3B3FFF	4K	SPI3
0xFF3B2000	0xFF3B2FFF	4K	Reserved
0xFF3B1000	0xFF3B1FFF	4K	UFS_SYS_CTRL
0xFF3B0000	0xFF3B0FFF	4K	UFS_CFG

Start Address	End Address	Size (Byte)	Module
0xFF3A0000	0xFF3AFFFF	64K	Reserved
0xFF390000	0xFF39FFFF	64K	Reserved
0xFF380000	0xFF38FFFF	64K	Reserved
0xFF37F000	0xFF37FFFF	4K	SD3
0xFF37E000	0xFF37EFFF	4K	IOC_MMC0
0xFF37D000	0xFF37DFFF	4K	Reserved
0xFF300000	0xFF37CFFF	500K	Reserved
0xFF201000	0xFF2FFFFF	1020K	Reserved
0xFF200000	0xFF200FFF	4K	USB3OTG_BC
0xFF100000	0xFF1FFFFF	1M	USB3OTG
0xFF050000	0xFF0FFFFF	704K	Reserved
0xFF013000	0xFF02FFFF	116K	Reserved
0xFF012000	0xFF012FFF	4K	Reserved
0xFF011000	0xFF011FFF	4K	IPC_MDM_NS
0xFF010000	0xFF010FFF	4K	IPC_MDM_S
0xFF00F000	0xFF00FFFF	4K	Reserved
0xFF000000	0xFF00EFFF	60K	Reserved
0xFDF31000	0xFDFFFFFF	828K	Reserved
0xFDF30000	0xFDF30FFF	4K	PERI_DMAC
0xFDF20000	0xFDF2FFFF	64K	Reserved
0xFDF16000	0xFDF1FFFF	40K	Reserved
0xFDF15000	0xFDF15FFF	4K	Reserved
0xFDF14000	0xFDF14FFF	4K	Reserved
0xFDF13000	0xFDF13FFF	4K	Reserved
0xFDF12000	0xFDF12FFF	4K	Reserved
0xFDF11000	0xFDF11FFF	4K	Reserved
0xFDF10000	0xFDF10FFF	4K	PERF_STAT
0xFDF0D000	0xFDF0DFFF	4K	I2C4
0xFDF0C000	0xFDF0CFFF	4K	I2C3
0xFDF0B000	0xFDF0BFFF	4K	I2C7
0xFDF09000	0xFDF0AFFF	8K	Reserved

Start Address	End Address	Size (Byte)	Module
0xFDF08000	0xFDF08FFF	4K	SPI1
0xFDF07000	0xFDF07FFF	4K	Reserved
0xFDF06000	0xFDF06FFF	4K	SPI4
0xFDF05000	0xFDF05FFF	4K	UART5
0xFDF04000	0xFDF04FFF	4K	Reserved
0xFDF03000	0xFDF03FFF	4K	UART2
0xFDF02000	0xFDF02FFF	4K	UART0
0xFDF01000	0xFDF01FFF	4K	UART4
0xFDF00000	0xFDF00FFF	4K	UART1
0xFC000000	0xFDEFFFFF	31M	Reserved
0xF4000000	0xFBFFFFFF	128M	PCIECtrl
0xF3F40000	0xF3FFFFFF	768K	Reserved
0xF3F00000	0xF3F3FFFF	256K	PCIEPHY
0xF1300000	0xF3EFFFFF	44M	Reserved
0xF12F0000	0xF12FFFFF	64K	Reserved
0xF1110000	0xF12EFFFF	1920K	Reserved
0xF0E00000	0xF0E1FFFF	128K	Reserved
0xF0C00000	0xF0DFFFFF	2M	Reserved
0xF0000000	0xF0BFFFFF	12M	IOMCU_TCM
0xED800000	0xEFFFFFFF	40M	Reserved
0xEC000000	0xED7FFFFF	24M	CSSYS_APB
0xE9890000	0xE989FFFF	64K	MMC0_NOC_Service_Target
0xE9880000	0xE988FFFF	64K	MMC1_NOC_Service_Target
0xE9870000	0xE987FFFF	64K	AOBUS_Service_Target
0xE9860000	0xE986FFFF	64K	DMA_NOC_Service_Target
0xE9810000	0xE981FFFF	64K	UFSBUS_Service_Target
0xE9800000	0xE980FFFF	64K	CFGBUS_Service_Target
0xE8E00000	0xE97FFFFF	10M	Reserved
0xE8DD0000	0xE8DFFFFF	192K	Reserved
0xE8A20000	0xE8A20FFF	4K	GPIO21
0xE8A1F000	0xE8A1FFFF	4K	GPIO20

Start Address	End Address	Size (Byte)	Module
0xE8A1E000	0xE8A1EFFF	4K	Reserved
0xE8A1D000	0xE8A1DFFF	4K	Reserved
0xE8A1C000	0xE8A1CFFF	4K	GPIO17
0xE8A1B000	0xE8A1BFFF	4K	GPIO16
0xE8A1A000	0xE8A1AFFF	4K	GPIO15
0xE8A19000	0xE8A19FFF	4K	GPIO14
0xE8A18000	0xE8A18FFF	4K	GPIO13
0xE8A17000	0xE8A17FFF	4K	GPIO12
0xE8A16000	0xE8A16FFF	4K	GPIO11
0xE8A15000	0xE8A15FFF	4K	GPIO10
0xE8A14000	0xE8A14FFF	4K	GPIO9
0xE8A13000	0xE8A13FFF	4K	GPIO8
0xE8A12000	0xE8A12FFF	4K	GPIO7
0xE8A11000	0xE8A11FFF	4K	GPIO6
0xE8A10000	0xE8A10FFF	4K	GPIO5
0xE8A0F000	0xE8A0FFFF	4K	GPIO4
0xE8A0E000	0xE8A0EFFF	4K	GPIO3
0xE8A0D000	0xE8A0DFFF	4K	GPIO2
0xE8A0C000	0xE8A0CFFF	4K	GPIO1
0xE8A0B000	0xE8A0BFFF	4K	GPIO0
0xE8A0A000	0xE8A0AFFF	4K	GPIO0_SE
0xE8A09000	0xE8A09FFF	4K	PCTRL
0xE8A07000	0xE8A07FFF	4K	WD1
0xE8A06000	0xE8A06FFF	4K	WD0
0xE8A04000	0xE8A04FFF	4K	PWM
0xE8A03000	0xE8A03FFF	4K	TIMER12
0xE8A02000	0xE8A02FFF	4K	TIMER11
0xE8A01000	0xE8A01FFF	4K	TIMER10
0xE8A00000	0xE8A00FFF	4K	TIMER9
0xE8971000	0xE89FFFFF	572K	Reserved
0xE896E000	0xE8970FFF	12K	Reserved

Start Address	End Address	Size (Byte)	Module
0xE896D000	0xE896DFFF	4K	Reserved
0xE896C000	0xE896CFFF	4K	IOC
0xE896B000	0xE896BFFF	4K	IPC_NS
0xE896A000	0xE896AFFF	4K	IPC
0xE8969800	0xE8969FFF	2K	Reserved
0xE8961800	0xE89697FF	32K	Reserved
0xE8961400	0xE89617FF	1K	Reserved
0xE8961000	0xE89613FF	1K	Reserved
0xE8960000	0xE8960FFF	4K	Reserved
0xE8950000	0xE895FFFF	64K	Reserved
0xE8300000	0xE83FFFFF	1M	Reserved
0xE82C4000	0xE82FFFFF	240K	Reserved
0xE82C0000	0xE82C3FFF	16K	G3D
0xE82BA000	0xE82BFFFF	24K	Reserved
0xE82B9000	0xE82B9FFF	4K	CODEC_SSI
0xE82B8000	0xE82B8FFF	4K	HKADC_SSI
0xE82B0000	0xE82B7FFF	32K	GIC400
0xE82A0000	0xE82AFFFF	64K	Reserved
0xE8200000	0xE829FFFF	640K	Reserved
0xE8100000	0xE81FFFFF	1M	CCI_CFG
0x00000000	0xDFFFFFFF	3584M	DRAM

NOTE

Table 2-3 lists the device address allocation in the 4 GB space. The DRAM space is 0–3.5 GB. When the 8-/6-/4-GB DDR is connected, the DRAM occupies the addresses that are beyond the 4 GB space.