ESP32-S31 Series

Datasheet Pre-release v0.2

RISC-V 32-bit dual-core microprocessor
2.4 GHz Wi-Fi 6, Bluetooth® 5.4 (LE), Bluetooth® Classic, Zigbee, and Thread (802.15.4)
Concurrent access to flash and PSRAM
60 GPIOs
QFN80 (8 × 8 mm) Package

Including:

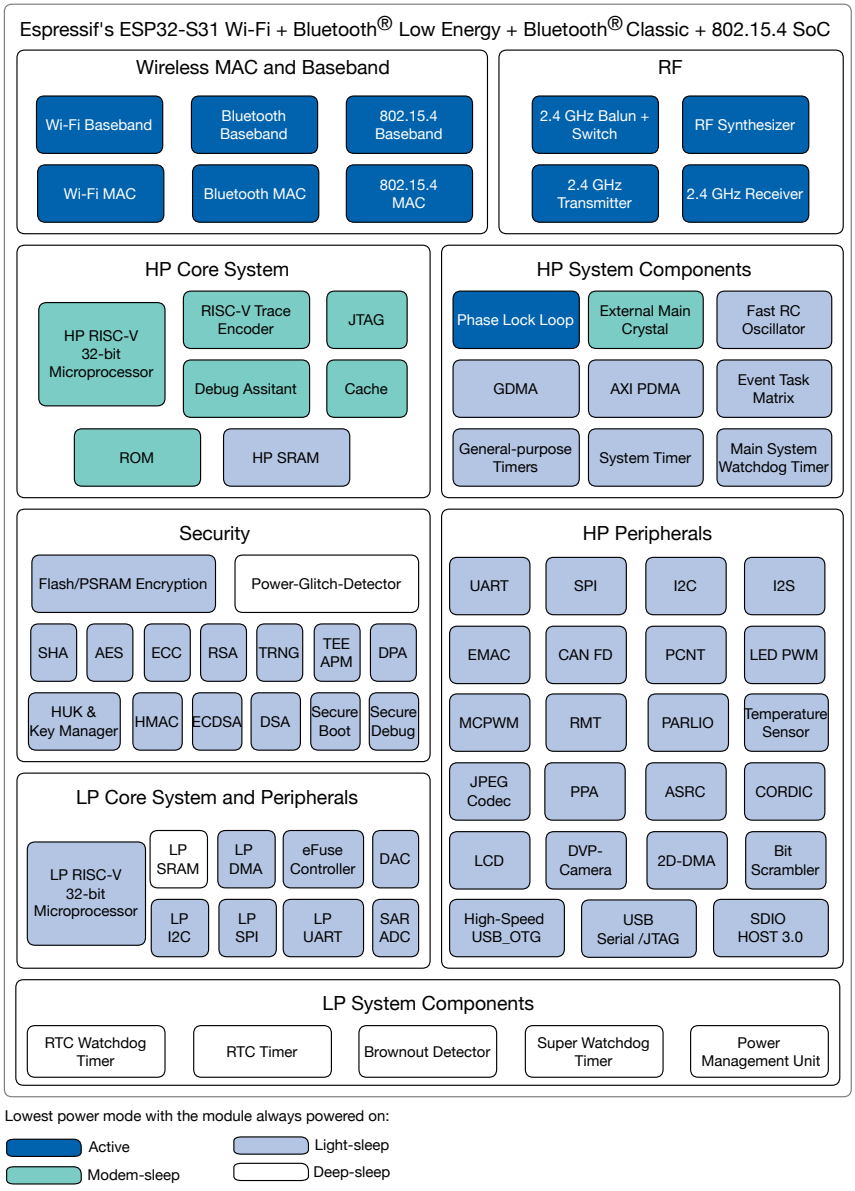
ESP32-S31NRV16

ESP32-S31NRV32

Product Overview

The ESP32-S31 SoC (System on Chip) supports 2.4 GHz Wi-Fi 6, Bluetooth® 5.4 (LE), Bluetooth® Classic, Zigbee 3.0, and Thread 1.4. It consists of a high-performance (HP) 32-bit RISC-V processor, a low-power (LP) 32-bit RISC-V processor, wireless baseband and MAC (Wi-Fi, Bluetooth, and 802.15.4), RF module, and numerous peripherals. The chip is designed for applications requiring high-throughput wireless connectivity and higher processing performance.

The functional block diagram of the SoC is shown below.



ESP32-S31 Functional Block Diagram

For more information on power consumption, see Section 4.1.4.6 *Power Management Unit*.

Features

Wi-Fi

- 1T1R in 2.4 GHz band
- Operating frequency: 2412 ~ 2484 MHz
- IEEE 802.11ax-compliant:
  - 20 MHz-only non-AP mode
  - Uplink and downlink OFDMA to enhance connectivity and performance in congested environments for IoT applications
  - Downlink MU-MIMO (multi-user, multiple input, multiple output) to increase network capacity
  - Beamformee that improves signal quality
  - Spatial reuse to maximize parallel transmissions
  - Target wake time (TWT) that optimizes power saving mechanisms
- Fully compatible with IEEE 802.11b/g/n protocol:
  - 20 MHz and 40 MHz bandwidth
  - Data rate up to 150 Mbps
  - Wi-Fi multimedia (WMM)
  - TX/RX A-MPDU, TX/RX A-MSDU
  - Immediate block ACK
  - Fragmentation and defragmentation
  - Transmission opportunity (TXOP)
  - Automatic beacon monitoring (hardware TSF)
  - Four virtual Wi-Fi interfaces
  - Simultaneous support for Infrastructure BSS in Station mode, SoftAP mode, Station + SoftAP mode, and promiscuous mode
    Note that when ESP32-S31 scans in Station mode, the SoftAP channel will change along with the Station channel
  - Antenna diversity
  - 802.11mc FTM

Bluetooth

Bluetooth LE

- Full support for the Bluetooth 5.4 (LE) core specification
- Bluetooth Mesh 1.1
- LE Audio (Isochronous Channels, BIS and CIS)

- Direction Finding (AoA/AoD)
- Periodic Advertising with Responses (PAwR)
- LE Connection Subrating
- LE Power Control
- Speed: 125 Kbps, 500 Kbps, 1 Mbps, 2 Mbps
- LE Advertising Extensions and Multiple Advertising Sets
- Simultaneous Operation of Broadcaster, Observer, Central, and Peripheral Devices

Bluetooth Classic

- Data rates: Basic rate 1 Mbps, Enhanced data rate 2 Mbps, 3 Mbps
- Asynchronous connection-oriented (ACL) links, synchronous connection-oriented (SCO) and enhanced synchronous connection-oriented (eSCO) links
- Voice coding formats: A-law, μ-law, CVSD and transparent data
- Channel Classification and adaptive frequency hopping (AFH)
- Traditional power control and enhanced power control
- Secure simple pairing (SSP)
- Data encryption: E0 and AES-CCM
- Secure Connections
- Sniff mode and Sniff Subrating mode
- Role switching
- Active Peripheral Broadcast
- Multiple connections and scatternet
  - Up to 2 synchronous links in the same piconet
  - As a central device, supports up to 7 ACL links in the same piconet
  - Up to 3 piconets simultaneously (1 central device, 2 peripheral devices)
- Maximum output power meets Power Class 1 device requirements
- Hardware-implemented clear channel assessment (CCA)

IEEE 802.15.4

- Compliant with IEEE 802.15.4-2015 protocol
- OQPSK PHY in 2.4 GHz band
- Data rate: 250 Kbps
- Thread 1.4
- Zigbee 3.0

- Matter
- Application-layer protocols (HomeKit, MQTT, etc.)

CPU and Memory

- Dual-core 32-bit RISC-V processor, clock speed up to 320 MHz
- ULP-RISC-V coprocessor
- Single-precision floating-point unit (FPU) per core
- 128-bit data bus with SIMD commands (supported by only one core)
- Sv32 two-level page-table address translation
- Private instruction cache (I-cache) for each core and shared data cache (D-cache)
- 320 KB ROM (accessed through dedicated ROM cache)
- 512 KB shared SRAM
- 32 KB low-power SRAM
- External PSRAM Interface: 250 MHz 8-bit DDR PSRAM (in-package only)
- Concurrent access to flash and PSRAM
- Supports multiple SPI interfaces for external flash/RAM:
  - Dedicated interfaces: SPI, Dual SPI, Quad SPI, Octal SPI, QPI, OPI
  - General-purpose interfaces: SPI, Dual SPI, Quad SPI, QPI (supports multiple flash or RAM devices)
- Supports Flash In-Circuit Programming (ICP)

System DMA

- General DMA controller (GDMA):
  - AHB-DMA (AHB_PDMA): 5 RX channels and 5 TX channels
  - AXI-DMA (AXI_PDMA): 3 RX channels and 3 TX channels
- 2D-DMA controller

Peripherals

- 60 Programmable GPIOs
  - 4 Strapping pins
  - 6 Pins for off-package flash connection
- Image and Audio Processing:
  - JPEG Codec
  - Pixel-Processing Accelerator (PPA)
  - Audio Sample Rate Converter (ASRC)

  - CORDIC Accelerator
  - LCD and Camera Controller
- Digital Interfaces and Peripherals:
  - 4 UART + 1 low-power UART (LP UART)
  - 6 SPI + 1 low-power SPI (LP SPI)
    * 2 SPI dedicated to in-package PSRAM
    * 2 SPI dedicated to off-package flash
    * 2 general-purpose SPI
    * 1 low-power SPI
  - 2 I2C + 1 low-power I2C (LP I2C)
  - 2 I2S interfaces with hardware-level Bluetooth Audio
  - 2 pulse counters
  - USB 2.0 high-speed OTG
  - USB Serial/JTAG controller
  - 1000 Mbps Ethernet MAC
  - CAN FD controller, compatible with ISO 11898-1:2015
  - SDIO host controller with 2 slots
  - 2 LED PWM controllers, up to 8 channels
  - 4 Motor Control PWM (MCPWM)
  - Infrared Remote Control (TX/RX)
  - Parallel IO Controller (PARLIO)
  - BitScrambler
- Analog Signal Processing:
  - Touch Sensor
  - Temperature Sensor
  - 2 x 12-bit SAR ADC, up to 16 channels
  - 2 x 10-bit DAC and 2 x 12-bit DAC
  - Analog Voltage Comparator (1 Reference Voltage Input + 3 Comparison Voltage Inputs)
- Timers:
  - 4 x 54-bit General-Purpose Timers
  - 1 x 52-bit System Timer
  - 3 Watchdog Timers

Power Management

- Power Management Unit (PMU)
- Brown-out Detector

Security

- Cryptographic Hardware Acceleration:
  - AES accelerator
  - ECC accelerator
  - HMAC accelerator
  - RSA accelerator
  - SHA accelerator
  - RSA digital signature peripheral (RSA_DS)
  - ECDSA digital signature peripheral (ECDSA_DS)
- Secure Debug Controller (SDC)
- External Memory Encryption and Decryption (XTS_AES)
- True Random Number Generator (TRNG)
- Key Manager
- Power Glitch Detector
- Secure Boot
- Side-Channel Attack Resistance (DPA)
- Access Permission Management (APM) and Trusted Execution Environment (TEE) Controller

RF Module

- Antenna switches, RF balun, power amplifier, low-noise receive amplifier
- Up to +20.5 dBm of power for an 802.11b transmission
- Up to +19.5 dBm of power for an 802.11ax transmission
- Up to -105 dBm of sensitivity for Bluetooth LE receiver (125 Kbps)

Applications

With low power consumption, ESP32-S31 is an ideal choice for IoT devices in the following areas:

- Smart Home
- Industrial Automation
- Health Care
- Consumer Electronics
- Smart Agriculture
- POS Machines
- Service Robot
- Audio Devices
- Generic Low-power IoT Sensor Hubs
- Generic Low-power IoT Data Loggers
- Cameras for Video Streaming
- USB Devices
- Speech Recognition
- Image Recognition
- Wi-Fi + Bluetooth Networking Card
- Touch and Proximity Sensing

Note:

Check the link or the QR code to make sure that you use the latest version of this document:
https://www.espressif.com/documentation/esp32-s31_datasheet_en.pdf

Contents

List of Tables

List of Figures

1 ESP32-S31 Series Information

1.1 Nomenclature

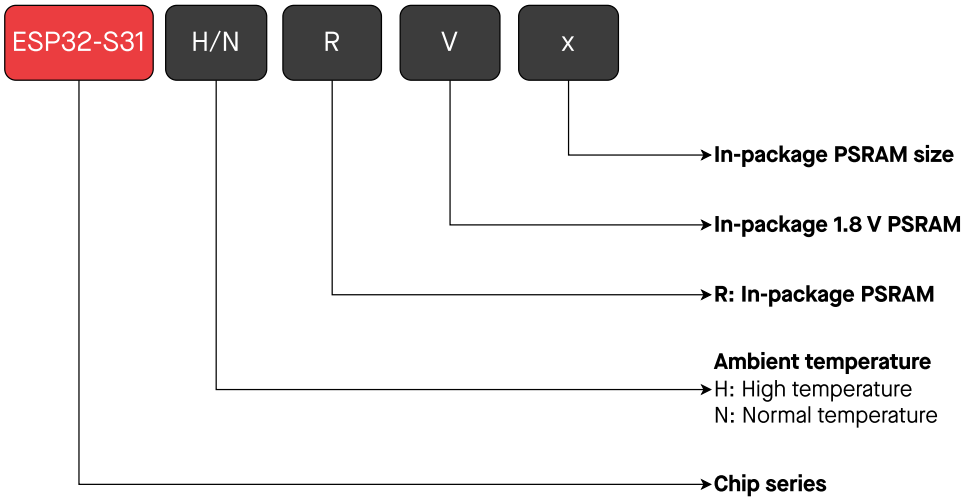


Figure 1-1. ESP32-S31 Series Nomenclature

1.2 Series Information

Table 1-1. ESP32-S31 Series Information

Part Number [1]	Ambient Temp. [2] (°C)	In-Package PSRAM	Package
ESP32-S31NRV16	–40∼85	16 MB (Octal SPI)[3]	QFN80 (8×8 mm)
ESP32-S31NRV32	–40∼85	32 MB (Octal SPI)	QFN80 (8×8 mm)

[1] For details on chip packaging, see Section 7 *Packaging*.

[2] Ambient temperature specifies the recommended temperature range of the environment outside an Espressif chip.

[3] For details about SPI modes, see Section 2.6 *Pin Mapping Between Chip and Flash*.

2 Pins

2.1 Pin Layout

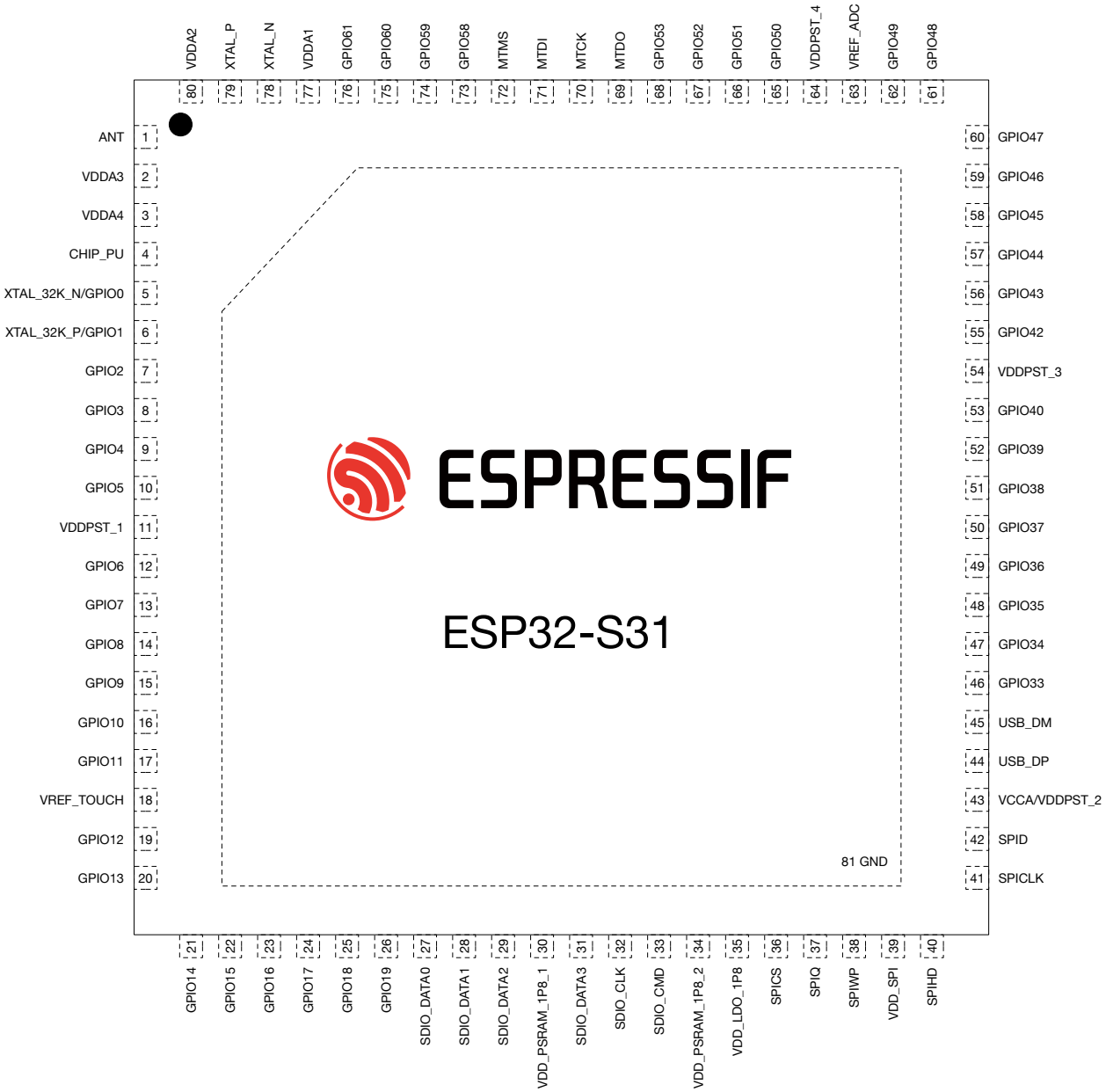


Figure 2-1. ESP32-S31 Pin Layout (Top View)

2.2 Pin Overview

The ESP32-S31 chip integrates multiple peripherals that require communication with the outside world. To keep the chip package size reasonably small, the number of available pins has to be limited. So the only way to route all the incoming and outgoing signals is through pin multiplexing.

The ESP32-S31 chip has the following types of pins:

- **IO pins** with the following predefined sets of functions to choose from:
  - **Each** IO pin has predefined **IO MUX functions** – see Table 2-3 *IO MUX Functions*
  - **Some** IO pins have predefined **LP IO MUX functions** – see Table 2-4 *LP IO MUX Functions*
  - **Some** IO pins have predefined **analog functions** – see Table 2-6 *Analog Functions*

  Predefined functions means that each IO pin has a set of direct connections to certain signals from on-chip components. During run-time, the user can configure which component signal from a predefined set to connect to a certain pin at a certain time via memory mapped registers.
- **Analog pins** that have exclusively-dedicated **analog functions** – see Table 2-7 *Analog Pins*
- **Power pins** supply power to the chip components and non-power pins – see Table 2-8 *Power Pins*

Table 2-1 *Pin Overview* gives an overview of all the pins. For more information, see respective sections below. Alternatively, see *Appendix A – ESP32-S31 Consolidated Pin Overview*.

Table 2-1. Pin Overview

Pin No.	Pin Name	Pin Type	Pin Providing Power [2]	Pin Settings [3]		Pin Function Sets [1]		
				At Reset	After Reset	IO MUX	LP IO MUX	Analog
1	ANT	Analog						
2	VDDA3	Power						
3	VDDA4	Power						
4	CHIP_PU	Analog						
5	XTAL_32K_N/GPIO0	IO	VDDPST_1					
6	XTAL_32K_P/GPIO1	IO	VDDPST_1					
7	GPIO2	IO	VDDPST_1			**IO MUX**	LP IO MUX	
8	GPIO3	IO	VDDPST_1			**IO MUX**	LP IO MUX	
9	GPIO4	IO	VDDPST_1			**IO MUX**	LP IO MUX	
10	GPIO5	IO	VDDPST_1			**IO MUX**	LP IO MUX	
11	VDDPST_1	Power						
12	GPIO6	IO	VDDPST_1			**IO MUX**	LP IO MUX	Analog
13	GPIO7	IO	VDDPST_1			**IO MUX**	LP IO MUX	Analog
14	GPIO8	IO	VDDPST_1		IE	**IO MUX**		Analog
15	GPIO9	IO	VDDPST_1		IE	**IO MUX**		Analog
16	GPIO10	IO	VDDPST_1		IE	**IO MUX**		Analog
17	GPIO11	IO	VDDPST_1		IE	**IO MUX**		Analog
18	VREF_TOUCH	Analog						
19	GPIO12	IO	VDDPST_1		IE	**IO MUX**		Analog
20	GPIO13	IO	VDDPST_1		IE	**IO MUX**		Analog
21	GPIO14	IO	VDDPST_1		IE	**IO MUX**		Analog

Cont'd on next page

Cont'd from previous page

Pin No.	Pin Name	Pin Type	Pin Providing Power [2]	Pin Settings [3]		Pin Function Sets [1]		
				At Reset	After Reset	IO MUX	LP IO MUX	Analog
22	GPIO15	IO	VDDPST_1		IE	**IO MUX**		Analog
23	GPIO16	IO	VDDPST_1		IE	**IO MUX**		Analog
24	GPIO17	IO	VDDPST_1		IE	**IO MUX**		Analog
25	GPIO18	IO	VDDPST_1		IE	**IO MUX**		Analog
26	GPIO19	IO	VDDPST_1		IE	**IO MUX**		Analog
27	SDIO_DATA0	IO	VDDPST_SD		IE	**IO MUX**		
28	SDIO_DATA1	IO	VDDPST_SD		IE	**IO MUX**		
29	SDIO_DATA2	IO	VDDPST_SD		IE	**IO MUX**		
30	VDD_PSRAM_1P8_1	Power						
31	SDIO_DATA3	IO	VDDPST_SD		IE	**IO MUX**		
32	SDIO_CLK	IO	VDDPST_SD		IE	**IO MUX**		
33	SDIO_CMD	IO	VDDPST_SD		IE	**IO MUX**		
34	VDD_PSRAM_1P8_2	Power						
35	VDD_LDO_1P8	Power						
36	SPICS	IO	VDD_SPI	WPU	WPU, IE	**IO MUX**		
37	SPIQ	IO	VDD_SPI	WPU	WPU, IE	**IO MUX**		
38	SPIWP	IO	VDD_SPI	WPU	WPU, IE	IO MUX		
39	VDD_SPI	Power						
40	SPIHD	IO	VDD_SPI	WPU	WPU, IE	**IO MUX**		
41	SPICLK	IO	VDD_SPI	WPU	WPU, IE	**IO MUX**		
42	SPID	IO	VDD_SPI	WPU	WPU, IE	**IO MUX**		
43	VCCA/VDDPST_2	Power						
44	USB_DP	Analog	VDDPST_2					
45	USB_DM	Analog	VDDPST_2					
46	GPIO33	IO	VDDPST_3	drv=3	USB_PU, IE, drv=3	IO MUX		Analog
47	GPIO34	IO	VDDPST_3	drv=3	USB_PU, IE, drv=3	IO MUX		Analog
48	GPIO35	IO	VDDPST_3		IE	**IO MUX**		
49	GPIO36	IO	VDDPST_3	IE	IE	**IO MUX**		
50	GPIO37	IO	VDDPST_3	IE	IE	**IO MUX**		Analog
51	GPIO38	IO	VDDPST_3	IE	IE	**IO MUX**		Analog
52	GPIO39	IO	VDDPST_3	IE	IE	**IO MUX**		Analog
53	GPIO40	IO	VDDPST_3	IE	IE	**IO MUX**		Analog
54	VDDPST_3	Power						
55	GPIO42	IO	VDDPST_3		IE	**IO MUX**		Analog
56	GPIO43	IO	VDDPST_3		IE	**IO MUX**		Analog
57	GPIO44	IO	VDDPST_3		IE	**IO MUX**		Analog
58	GPIO45	IO	VDDPST_3		IE	**IO MUX**		Analog
59	GPIO46	IO	VDDPST_3		IE	**IO MUX**		Analog
60	GPIO47	IO	VDDPST_3		IE	**IO MUX**		Analog
61	GPIO48	IO	VDDPST_3			**IO MUX**		Analog
62	GPIO49	IO	VDDPST_3			**IO MUX**		Analog
63	VREF_ADC	Analog						
64	VDDPST_4	Power						
65	GPIO50	IO	VDDPST_4			**IO MUX**		Analog
66	GPIO51	IO	VDDPST_4			**IO MUX**		Analog

Cont'd on next page

Cont'd from previous page

Pin No.	Pin Name	Pin Type	Pin Providing Power [2]	Pin Settings [3]		Pin Function Sets [1]		
				At Reset	After Reset	IO MUX	LP IO MUX	Analog
67	GPIO52	IO	VDDPST_4		IE	**IO MUX**		Analog
68	GPIO53	IO	VDDPST_4		IE	**IO MUX**		Analog
69	MTDO	IO	VDDPST_4		IE	**IO MUX**		Analog
70	MTCK	IO	VDDPST_4		IE	**IO MUX**		Analog
71	MTDI	IO	VDDPST_4		IE	**IO MUX**		Analog
72	MTMS	IO	VDDPST_4		IE	**IO MUX**		Analog
73	GPIO58	IO	VDDPST_4		IE	**IO MUX**		
74	GPIO59	IO	VDDPST_4		IE	**IO MUX**		
75	GPIO60	IO	VDDPST_4	WPU, IE	WPU, IE	**IO MUX**		
76	GPIO61	IO	VDDPST_4	WPU, IE	WPU, IE	**IO MUX**		
77	VDDA1	Power						
78	XTAL_N	Analog						
79	XTAL_P	Analog						
80	VDDA2	Power						

1. **Bold** marks the pin function set in which a pin has its default function in the default boot mode. For more information about the default boot mode, see Section 3.1 *Chip Boot Mode Control*.
2. In column **Pin Providing Power**: Pins powered by VDD_SPI:
   - The actual power source is the internal power rail that supplies VDD_SPI. For details, see Section 2.5.2 *Power Scheme*.
3. Column **Pin Settings** shows predefined settings at reset and after reset with the following abbreviations:
   - IE – input enabled
   - WPU – internal weak pull-up resistor enabled
   - WPD – internal weak pull-down resistor enabled
   - USB_PU – USB pull-up resistor enabled
     - By default, the USB function is enabled for USB pins (i.e., GPIO33 and GPIO34), and the pin pull-up is decided by the USB pull-up. The USB pull-up is controlled by USB_SERIAL_JTAG_DP/DM_PULLUP and the pull-up resistor value is controlled by USB_SERIAL_JTAG_PULLUP_VALUE.
     - When the USB function is disabled, USB pins are used as regular GPIOs and the pin's internal weak pull-up and pull-down resistors are disabled by default (configurable by IO_MUX_FUN_WPU/WPD).
4. Default drive strength for all IO pins is 20 mA.
5. Depends on the value of EFUSE_DIS_PAD_JTAG
   - `0` - WPU is enabled
   - `1` - the pin is floating

2.3 IO Pins

2.3.1 IO MUX Functions

The IO MUX allows multiple input/output signals to be connected to a single input/output pin. Each IO pin of ESP32-S31 can be connected to one of the five signals (IO MUX functions, i.e., F0–F4), as listed in Table 2-3 *IO MUX Functions*.

Among the five sets of signals:

- Some are routed via the GPIO Matrix (**GPIO0, GPIO1, etc.**), which incorporates internal signal routing circuitry for mapping signals programmatically. It gives the pin access to almost any peripheral signals. However, the flexibility of programmatic mapping comes at a cost as it might affect the latency of routed signals.
- Some are directly routed from certain peripherals (**U0TXD, MTCK, etc.**), including UART0, JTAG, and SPI2 - see Table 2-2 *Peripheral Signals Routed via IO MUX*.

Table 2-2. Peripheral Signals Routed via IO MUX

Pin Function	Signal	Description
MTCK MTDO MTDI MTMS	Test clock Test data out Test data in Test mode select	JTAG interface for debugging
SPI2_HOLD_PAD SPI2_CS_PAD SPI2_D_PAD SPI2_CK_PAD SPI2_Q_PAD SPI2_WP_PAD	Hold Chip select Data in Clock Data out Write protect	3.3 V SPI2 interface which can operate in master and slave modes. The interface supports 1-line, 2-line, 4-line, and 8-line modes (the 8-line mode is supported only in the master mode).
SPI2_IO..._PAD SPI2_DQS_PAD	Data Data strobe/data mask	The high 4-bit data line interface and the DQS interface for 3.3 V SPI2 interface in 8-line SPI mode
UART0_TXD_PAD UART0_RXD_PAD	Transmit data Receive data	UART0 Interface
GMAC_PHY_RXDV_PAD [1] GMAC_PHY_RXD..._PAD GMAC_PHY_RXER_PAD GMAC_PHY_TXDV_PAD GMAC_PHY_TXD..._PAD GMAC_PHY_TXER_PAD GMAC_PHY_TXEN_PAD GMAC_RMII_CLK_PAD	Receive data valid Receive data line 0/1 Receive error Transmit data valid Transmit data line 0/1 Transmit error Transmit enable RMII clock	RMII Ethernet PHY interface
SD1_CDATA..._PAD SD1_CCLK_PAD SD1_CCMD_PAD	Card data line 0–7 of SD1 Card clock of SD1 Card command of SD1	SDIO3.0 interface
CAM_DATA..._PAD CAM_PCLK_PAD CAM_XCLK_PAD CAM_V_SYNC_PAD	Data in Pixel clock in Clock out Vertical sync	Camera interface

Cont'd on next page

Table 2-2 – cont'd from previous page

Pin Function	Signal	Description
CAM_H_SYNC_PAD	Horizontal sync	
LCD_DATA..._OUT_PAD	Data out	LCD display interface
LCD_PCLK_PAD	Pixel clock out	
LCD_H_ENABLE_PAD	Horizontal data enable	
LCD_V_SYNC_PAD	Vertical sync	
LCD_H_SYNC_PAD	Horizontal sync	

[1] The PAD layer does not distinguish between MII and RMII interfaces. This signal is used as RX_DV in MII mode and as CRS_DV in RMII mode.

Table 2-3 *IO MUX Functions* shows the IO MUX functions of IO pins.

Table 2-3. IO MUX Functions

		IO MUX Functions									
Pin No.	GPIO	F0	Type	F1	Type	F2	Type	F3	Type	F4	Type
5	XTAL_32K_N	**GPIO0**	I/O/T	GPIO0	I/O/T						
6	XTAL_32K_P	**GPIO1**	I/O/T	GPIO1	I/O/T						
7	GPIO2	**GPIO2**	I/O/T	GPIO2	I/O/T			lcd_data19_out_pad	O		
8	GPIO3	**GPIO3**	I/O/T	GPIO3	I/O/T			lcd_data20_out_pad	O		
9	GPIO4	**GPIO4**	I/O/T	GPIO4	I/O/T			lcd_data21_out_pad	O		
10	GPIO5	**GPIO5**	I/O/T	GPIO5	I/O/T			lcd_data22_out_pad	O		
12	GPIO6	**GPIO6**	I/O/T	GPIO6	I/O/T						
13	GPIO7	**GPIO7**	I/O/T	GPIO7	I/O/T			lcd_data23_out_pad	O		
14	GPIO8	**GPIO8**	I/O/T	GPIO8	I/O/T	gmac_phy_txd0_pad	O	lcd_data0_out_pad	O		
15	GPIO9	**spi2_hold_pad**	I1/O/T	GPIO9	I/O/T	gmac_phy_txd1_pad	O	lcd_data1_out_pad	O	dbg_psram_ck_pad	O
16	GPIO10	**spi2_cs_pad**	I1/O/T	GPIO10	I/O/T	gmac_phy_txd2_pad	O	lcd_data2_out_pad	O	dbg_psram_cs_pad	O
17	GPIO11	**spi2_d_pad**	I1/O/T	GPIO11	I/O/T	gmac_phy_txd3_pad	O	lcd_data3_out_pad	O	dbg_psram_d_pad	O
19	GPIO12	**spi2_ck_pad**	I1/O/T	GPIO12	I/O/T	gmac_phy_txen_pad	O	lcd_data4_out_pad	O	dbg_psram_q_pad	O
20	GPIO13	**spi2_q_pad**	I1/O/T	GPIO13	I/O/T	gmac_rmii_clk_pad	I0/O/T	lcd_data5_out_pad	O	dbg_psram_wp_pad	O
21	GPIO14	**spi2_wp_pad**	I1/O/T	GPIO14	I/O/T	gmac_rx_clk_pad	I0	lcd_data6_out_pad	O	dbg_psram_hold_pad	O
22	GPIO15	**spi2_io4_pad**	I1/O/T	GPIO15	I/O/T	gmac_phy_rxdv_pad	I0	lcd_data7_out_pad	O	dbg_psram_dq4_pad	O
23	GPIO16	**spi2_io5_pad**	I1/O/T	GPIO16	I/O/T	gmac_phy_rxd3_pad	I0	lcd_data8_out_pad	O	dbg_psram_dq5_pad	O
24	GPIO17	**spi2_io6_pad**	I1/O/T	GPIO17	I/O/T	gmac_phy_rxd2_pad	I0	lcd_data9_out_pad	O	dbg_psram_dq6_pad	O
25	GPIO18	**spi2_io7_pad**	I1/O/T	GPIO18	I/O/T	gmac_phy_rxd1_pad	I0	lcd_data10_out_pad	O	dbg_psram_dq7_pad	O
26	GPIO19	**spi2_dqs_pad**	O/T	GPIO19	I/O/T	gmac_phy_rxd0_pad	I0	lcd_data11_out_pad	O	dbg_psram_dqs_0_pad	O
27	SDIO_DATA0	**GPIO20**	I/O/T	GPIO20	I/O/T	spi2_ck_pad	I1/O/T			dbg_flash_ck_pad	O
28	SDIO_DATA1	**GPIO21**	I/O/T	GPIO21	I/O/T	spi2_d_pad	I1/O/T			dbg_flash_d_pad	O
29	SDIO_DATA2	**GPIO22**	I/O/T	GPIO22	I/O/T	spi2_q_pad	I1/O/T			dbg_flash_cs_pad	O
31	SDIO_DATA3	**GPIO23**	I/O/T	GPIO23	I/O/T	spi2_cs_pad	I1/O/T			dbg_flash_q_pad	O
32	SDIO_CLK	**GPIO24**	I/O/T	GPIO24	I/O/T	spi2_hold_pad	I1/O/T			dbg_flash_wp_pad	O
33	SDIO_CMD	**GPIO25**	I/O/T	GPIO25	I/O/T	spi2_wp_pad	I1/O/T			dbg_flash_hold_pad	O
36	SPICS	**flash_cs_pad**	O/T	GPIO26	I/O/T						
37	SPIQ	**flash_q_pad**	I1/O/T	GPIO27	I/O/T						

Cont'd on next page

Cont'd from previous page

Pin No.	GPIO	IO MUX Functions									
		F0	Type	F1	Type	F2	Type	F3	Type	F4	Type
38	SPIWP	flash_wp_pad	I1/O/T	GPIO28	I/O/T						
40	SPIHD	**flash_hold_pad**	I1/O/T	GPIO30	I/O/T						
41	SPICLK	**flash_ck_pad**	O/T	GPIO31	I/O/T						
42	SPID	**flash_d_pad**	I1/O/T	GPIO32	I/O/T						
46	GPIO33	GPIO33	I/O/T	GPIO33	I/O/T			lcd_data12_out_pad	O		
47	GPIO34	GPIO34	I/O/T	GPIO34	I/O/T			lcd_data13_out_pad	O		
48	GPIO35	**GPIO35**	I/O/T	GPIO35	I/O/T	ref_gmac_clk_pad	O	lcd_data14_out_pad	O	sd2_cdata0_pad	I1/O/T
49	GPIO36	**GPIO36**	I/O/T	GPIO36	I/O/T	gmac_phy_rxdv_pad	I0	lcd_data15_out_pad	O	sd2_cdata1_pad	I1/O/T
50	GPIO37	**GPIO37**	I/O/T	GPIO37	I/O/T	gmac_phy_txen_pad	O	lcd_data16_out_pad	O	sd2_cdata2_pad	I1/O/T
51	GPIO38	**GPIO38**	I/O/T	GPIO38	I/O/T	gmac_phy_rxd3_pad	I0	lcd_data17_out_pad	O	sd2_cdata3_pad	I1/O/T
52	GPIO39	**GPIO39**	I/O/T	GPIO39	I/O/T	gmac_phy_rxd2_pad	I0	lcd_data18_out_pad	O	sd2_cclk_pad	O
53	GPIO40	**GPIO40**	I/O/T	GPIO40	I/O/T	gmac_phy_rxd1_pad	I0	lcd_pclk_pad	O	sd2_ccmd_pad	I1/O/T
55	GPIO42	**GPIO42**	I/O/T	GPIO42	I/O/T	gmac_rx_clk_pad	I0				
56	GPIO43	**GPIO43**	I/O/T	GPIO43	I/O/T	gmac_rmii_clk_pad	I0/O/T	lcd_h_enable_pad	O		
57	GPIO44	**GPIO44**	I/O/T	GPIO44	I/O/T	gmac_phy_txd0_pad	O	lcd_h_sync_pad	O		
58	GPIO45	**GPIO45**	I/O/T	GPIO45	I/O/T	gmac_phy_txd1_pad	O	lcd_v_sync_pad	O		
59	GPIO46	**GPIO46**	I/O/T	GPIO46	I/O/T	gmac_phy_txd2_pad	O	cam_data0_in_pad	I0		
60	GPIO47	**GPIO47**	I/O/T	GPIO47	I/O/T	gmac_phy_txd3_pad	O	cam_data1_in_pad	I0		
61	GPIO48	**GPIO48**	I/O/T	GPIO48	I/O/T			cam_data2_in_pad	I0		
62	GPIO49	**GPIO49**	I/O/T	GPIO49	I/O/T			cam_data3_in_pad	I0		
65	GPIO50	**GPIO50**	I/O/T	GPIO50	I/O/T			cam_data4_in_pad	I0		
66	GPIO51	**GPIO51**	I/O/T	GPIO51	I/O/T			cam_data5_in_pad	I0		
67	GPIO52	**GPIO52**	I/O/T	GPIO52	I/O/T	spi2_cs_pad	I1/O/T	cam_data6_in_pad	I0		
68	GPIO53	**GPIO53**	I/O/T	GPIO53	I/O/T	spi2_ck_pad	I1/O/T	cam_data7_in_pad	I0		
69	MTDO	**MTDO**	O/T	GPIO54	I/O/T	spi2_d_pad	I1/O/T	cam_pclk_pad	I0		
70	MTCK	**MTCK**	I1	GPIO55	I/O/T	spi2_q_pad	I1/O/T	cam_xclk_pad	O		
71	MTDI	**MTDI**	I1	GPIO56	I/O/T	spi2_hold_pad	I1/O/T	cam_v_sync_pad	I0		
72	MTMS	**MTMS**	I1	GPIO57	I/O/T	spi2_wp_pad	I1/O/T	cam_h_sync_pad	I0		
73	GPIO58	**uart0_txd_pad**	O	GPIO58	I/O/T						

Cont'd on next page

Cont'd from previous page

Pin No.	GPIO	IO MUX Functions									
		F0	Type	F1	Type	F2	Type	F3	Type	F4	Type
74	GPIO59	**uart0_rxd_pad**	I1	GPIO59	I/O/T						
75	GPIO60	**GPIO60**	I/O/T	GPIO60	I/O/T						
76	GPIO61	**GPIO61**	I/O/T	GPIO61	I/O/T						

[1] **Bold** marks the default pin functions in the default boot mode. See Section 3.1 *Chip Boot Mode Control*.

[2] Regarding highlighted cells, see Section 2.3.4 *Restrictions for GPIOs and LP GPIOs*.

[3] Each IO MUX function (F*n*, *n* = 0–4) is associated with a *type*. The description of *type* is as follows:

- I – input. O – output. T – high impedance.
- I1 – input; if the pin is assigned a function other than F*n*, the input signal of F*n* is always `1`.
- I0 – input; if the pin is assigned a function other than F*n*, the input signal of F*n* is always `0`.

2.3.2 LP IO MUX Functions

When the chip is in Deep-sleep mode, the IO MUX described in Section 2.3.1 *IO MUX Functions* is not available. This is where the LP IO MUX is used. As LP IO pins are connected to the LP system, the LP IO MUX allows one LP input/output pin to be connected to multiple input/output signals in Deep-sleep mode.

LP IO pins can be assigned to **LP functions**. They can work as LP GPIOs (**LP_GPIO0, LP_GPIO1, etc.**).

Table 2-4 *LP IO MUX Functions* lists the LP IO MUX functions of LP IO pins.

Table 2-4. LP IO MUX Functions

Pin No.	LP IO Name[1]	LP IO MUX Function							
		F0	Type	F1	Type	F2	Type	F3	Type
5	LP_GPIO0	LP_GPIO0	I/O/T	LP_GPIO0	I/O/T			LP_PROBE_TOP_OUT0	O
6	LP_GPIO1	LP_GPIO1	I/O/T	LP_GPIO1	I/O/T			LP_PROBE_TOP_OUT1	O
7	LP_GPIO2	LP_UART_DTRN_PAD	O	LP_GPIO2	I/O/T	LP_SPI_CK_PAD	I1/O/T	LP_PROBE_TOP_OUT2	O
8	LP_GPIO3	LP_UART_DSRN_PAD	I1	LP_GPIO3	I/O/T	LP_SPI_CS_PAD	I1/O/T	LP_PROBE_TOP_OUT3	O
9	LP_GPIO4	LP_UART_RTSN_PAD	O	LP_GPIO4	I/O/T	LP_SPI_D_PAD	I1/O/T	LP_PROBE_TOP_OUT4	O
10	LP_GPIO5	LP_UART_CTSN_PAD	I1	LP_GPIO5	I/O/T	LP_SPI_Q_PAD	I1/O/T	LP_PROBE_TOP_OUT5	O
12	LP_GPIO6	LP_UART_TXD_PAD	O	LP_GPIO6	I/O/T	LP_I2C_SCL_PAD	I1/O/T	LP_PROBE_TOP_OUT6	O
13	LP_GPIO7	LP_UART_RXD_PAD	I1	LP_GPIO7	I/O/T	LP_I2C_SDA_PAD	I1/O/T	LP_PROBE_TOP_OUT7	O

[1] This column lists LP GPIO names because LP IO MUX functions are configured via LP GPIO registers that use LP GPIO numbering.

2.3.3 Analog Functions

Some IO pins also have **analog functions**, for analog peripherals (such as ADC) in any power mode. Internal analog signals are routed to these analog functions, see Table 2-5 *Analog Signals Routed to Analog Functions*.

Table 2-5. Analog Signals Routed to Analog Functions

Pin Function	Signal	Description
TOUCH...	Touch sensor channel ... signal	Touch sensor interface
ADC..._CH...	ADC1/2 channel ... signal	ADC1/2 interface
XTAL_32K_N XTAL_32K_P	Negative clock signal Positive clock signal	32 kHz external clock input/output connected to ESP32-S31's crystal or oscillator
USB_D- USB_D+	Data - Data +	USB OTG and USB Serial/JTAG function
PAD_COMP...	Analog voltage comparator channel... signal	Analog voltage comparator interface

Table 2-6 *Analog Functions* shows the analog functions of IO pins.

Table 2-6. Analog Functions

Pin No.	GPIO	Analog Functions F0
12	GPIO6	TOUCH_CHANNEL0
13	GPIO7	TOUCH_CHANNEL1
14	GPIO8	TOUCH_CHANNEL2
15	GPIO9	TOUCH_CHANNEL3
16	GPIO10	TOUCH_CHANNEL4
17	GPIO11	TOUCH_CHANNEL5
19	GPIO12	TOUCH_CHANNEL6
20	GPIO13	TOUCH_CHANNEL7
21	GPIO14	TOUCH_CHANNEL8
22	GPIO15	TOUCH_CHANNEL9
23	GPIO16	TOUCH_CHANNEL10
24	GPIO17	TOUCH_CHANNEL11
25	GPIO18	TOUCH_CHANNEL12
26	GPIO19	TOUCH_CHANNEL13
39	GPIO29	VDD_SPI
46	GPIO33	USB1P1_N0
47	GPIO34	USB1P1_P0
50	GPIO37	PAD COMP0 (MUX4)
51	GPIO38	PAD COMP1 (MUX4)
52	GPIO39	PAD COMP2 (MUX4)
53	GPIO40	PAD COMP3 (MUX4)
55	GPIO42	ADC1_CHANNEL0_N
56	GPIO43	ADC1_CHANNEL0_P
57	GPIO44	ADC1_CHANNEL1_N

Cont'd on next page

Cont'd from previous page

Pin No.	GPIO	Analog Functions F0
58	GPIO45	ADC1_CHANNEL1_P
59	GPIO46	ADC1_CHANNEL2_N
60	GPIO47	ADC1_CHANNEL2_P
61	GPIO48	ADC1_CHANNEL3_N
62	GPIO49	ADC1_CHANNEL3_P
65	GPIO50	ADC2_CHANNEL0_N
66	GPIO51	ADC2_CHANNEL0_P
67	GPIO52	ADC2_CHANNEL1_N
68	GPIO53	ADC2_CHANNEL1_P
69	GPIO54	ADC2_CHANNEL2_N
70	GPIO55	ADC2_CHANNEL2_P
71	GPIO56	ADC2_CHANNEL3_N
72	GPIO57	ADC2_CHANNEL3_P

[1] **Bold** marks the default pin functions in the default boot mode. For more information about the default boot mode, see Section 3.1 *Chip Boot Mode Control*.

[2] This column lists the GPIO names, since analog functions are configured with GPIO registers that use GPIO numbering.

[3] Regarding highlighted cells, see Section 2.3.4 *Restrictions for GPIOs and LP GPIOs*.

2.3.4 Restrictions for GPIOs and LP GPIOs

All IO pins of the ESP32-S31 have GPIO and some have LP GPIO pin functions. However, the IO pins are multiplexed and can be configured for different purposes based on the requirements. Some IOs have restrictions for usage. It is essential to consider the multiplexed nature and the limitations when using these IO pins.

In tables of this chapter, some pin functions are highlighted. The non-highlighted GPIO or LP GPIO pins are recommended for use first. If more pins are needed, the highlighted GPIOs or LP GPIOs should be chosen carefully to avoid conflicts with important pin functions.

The highlighted IO pins have the following important pin functions:

- GPIO – allocated for communication with flash/PSRAM and NOT recommended for other uses. For details, see Section 2.6 Pin Mapping Between Chip and Flash.
- GPIO – have one of the following important functions:
  - **Strapping pins** – need to be at certain logic levels at startup. See Section 3 *Boot Configurations*.

    Note:
    Strapping pins are highlighted by pin name, instead of pin functions.

  - **USB1P1_P/N** – by default, connected to the USB Serial/JTAG controller or USB OTG. To function as GPIOs, these pins need to be reconfigured.
  - **JTAG interface** – often used for debugging. See Table 2-2 *Peripheral Signals Routed via IO MUX*. To free these pins up, the pin functions USB_D+/- of the USB Serial/JTAG controller can be used instead. See also Section 3.4 *JTAG Signal Source Control*.
  - **UART interface** – often used for debugging. See Table 2-2 *Peripheral Signals Routed via IO MUX*.

See also *Appendix A – ESP32-S31 Consolidated Pin Overview*.

2.4 Analog Pins

Table 2-7. Analog Pins

Pin No.	Pin Name	Pin Type	Pin Function
1	ANT	I/O	RF input and output
4	CHIP_PU	I	High level: Chip enabled (powered on); Low level: Chip disabled (powered off); Do not leave CHIP_PU floating
18	VREF_TOUCH	I/O	TOUCH reference voltage
44	USB_DP	I/O	USB D+
45	USB_DM	I/O	USB D-
63	VREF_ADC	I/O	ADC reference voltage
78	XTAL_N	—	External clock input/output for connecting active or passive crystal oscillators of ESP32-S31. For an active crystal oscillator, connect to P and leave N floating
79	XTAL_P		

2.5 Power Supply

2.5.1 Power Pins

The chip is powered via the power pins described in Table 2-8 *Power Pins*.

Table 2-8. Power Pins

Pin No.	Pin Name	Direction	Power Supply[1]	
			Power Domain / Other[2]	IO Pins
2	VDDA3	Input	Analog power domain	
3	VDDA4	Input	Analog power domain	
11	VDDPST_1	Input	Digital LP power domain	LP IO [3]
30	VDD_PSRAM_1P8_1	Input	PSRAM	PSRAM IO
34	VDD_PSRAM_1P8_2	Input	PSRAM	PSRAM IO
35	VDD_LDO_1P8	Output	PSRAM/FLASH/SD	PSRAM/FLASH/SD IO
39	VDD_SPI [2]	Input	FLASH	FLASH IO
		Output	FLASH	FLASH IO
43	VCCA/VDDPST_2	Input	USB_PHY/Digital HP power domain	USB IO
54	VDDPST_3	Input	Digital HP power domain	HP IO
64	VDDPST_4	Input	Digital HP power domain	HP IO
77	VDDA1	Input	Analog power domain	
80	VDDA2	Input	Analog power domain	

[1] See in conjunction with Section 2.5.2 *Power Scheme*.

[2] LP IO pins are those powered by VDDPST_1, as shown in Figure 2-2 *ESP32-S31 Power Scheme*. See also Table 2-1 *Pin Overview* > Column *Pin Providing Power*.

2.5.2 Power Scheme

The power scheme is shown in Figure 2-2 *ESP32-S31 Power Scheme*.

The components on the chip are powered via voltage regulators.

Table 2-9. Voltage Regulators

Voltage Regulator	Output	Power Supply
HP_LDO+MEM_LDO+MEM_LDO_SLV	1.1 V	Digital power domain
LP_LDO	1.1 V	LP power domain
PSRAM	1.8 V	Configurable to supply in-package PSRAM or off-package memory
FLASH	3.3 V	Configurable to supply in-package flash
SDIO	3.3V/1.8V	Configurable as the SDIO power supply

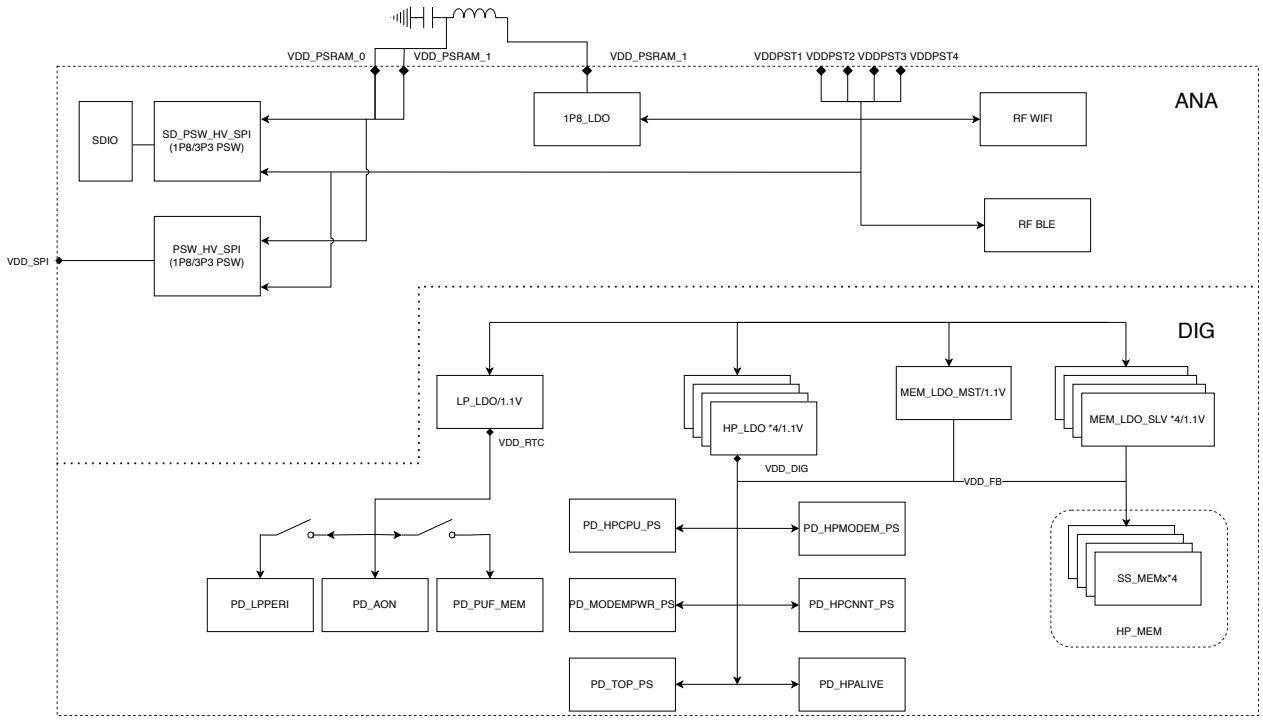


Figure 2-2. ESP32-S31 Power Scheme

2.5.3 Chip Power-up and Reset

Once the power is supplied to the chip, its power rails need a short time to stabilize. After that, CHIP_PU – the pin used for power-up and reset – is pulled high to activate the chip. For information on CHIP_PU as well as power-up and reset timing, see Figure 2-3 and Table 2-10.

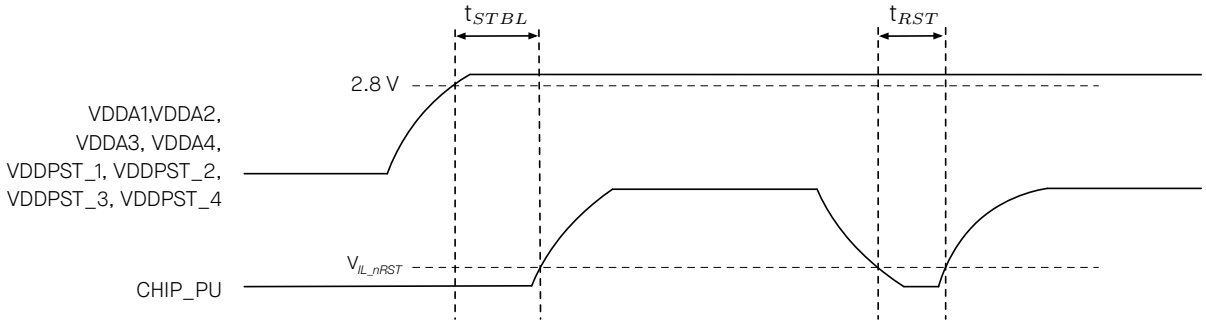


Figure 2-3. Visualization of Timing Parameters for Power-up and Reset

Table 2-10. Description of Timing Parameters for Power-up and Reset

Parameter	Description	Min (ms)
t_{STBL}	Time reserved for the power rails of VDDA1, VDDA2, VDDA3, VDDA4, VDDPST_1, VDDPST_2, VDDPST_3, VDDPST_4 to stabilize before the CHIP_PU pin is pulled high to activate the chip	1
t_{RST}	Time reserved for CHIP_PU to stay below V_{IL_nRST} to reset the chip (see Table 5-3)	1

2.6 Pin Mapping Between Chip and Flash

ESP32-S31 requires off-package flash to store application firmware and data. ESP32-S31 supports up to 256 MB flash, which can be connected through SPI, Dual SPI, and Quad SPI/QPI.

ESP32-S31 includes eight-line PSRAM with the operation voltage of 1.8 V. Please note that PSRAM is not pinned out.

Table 2-11 lists the pin mapping between the chip and flash for all SPI modes.

For more information on SPI controllers, see also Section 4.2.2.2 *SPI Controller (SPI)*.

Table 2-11. Pin Mapping Between Chip and Off-Package Flash

Pin No.	Pin Name	Single SPI	Dual SPI	Quad SPI/QPI
27	SPICS	CS#	CS#	CS#
28	SPIQ	DO	DO	DO
29	SPIWP	WP#	WP#	WP#
31	SPIHD	HOLD#	HOLD#	HOLD#
32	SPICLK	CLK	CLK	CLK
33	SPID	DI	DI	DI

3 Boot Configurations

The chip allows for configuring the following boot parameters through strapping pins and eFuse parameters at power-up or a hardware reset, without microcontroller interaction.

- **Chip boot mode**
  - Strapping pin: GPIO60 and GPIO61
- **ROM message printing**
  - Strapping pin: GPIO60
  - eFuse parameter: EFUSE_UART_PRINT_CONTROL and EFUSE_DIS_USB_SERIAL_JTAG_ROM_PRINT
- **JTAG signal source**
  - Strapping pin: GPIO37
  - eFuse parameter: EFUSE_DIS_PAD_JTAG, EFUSE_DIS_USB_JTAG, and EFUSE_JTAG_SEL_ENABLE

The default values of all the above eFuse parameters are 0, which means that they are not burnt. Given that eFuse is one-time programmable, once programmed to 1, it can never be reverted to 0.

The default values of the strapping pins, namely the logic levels, are determined by pins' internal weak pull-up/pull-down resistors at reset if the pins are not connected to any circuit, or connected to an external high-impedance circuit.

Table 3-1. Default Configuration of Strapping Pins

Strapping Pin	Default Configuration	Bit Value
GPIO61	Weak pull-up	1
GPIO60	Weak pull-up	1

To change the bit values, the strapping pins should be connected to external pull-down/pull-up resistances. If the ESP32-S31 is used as a device by a host MCU, the strapping pin voltage levels can also be controlled by the host MCU.

All strapping pins have latches. At Chip Reset, the latches sample the bit values of their respective strapping pins and store them until the chip is powered down or shut down. The states of latches cannot be changed in any other way. It makes the strapping pin values available during the entire chip operation, and the pins are freed up to be used as regular IO pins after reset.

The timing of signals connected to the strapping pins should adhere to the *setup time* and *hold time* specifications in Table 3-2 and Figure 3-1.

Table 3-2. Description of Timing Parameters for the Strapping Pins

Parameter	Description	Min (ms)
t_{SU}	*Setup time* is the time reserved for the power rails to stabilize before the CHIP_PU pin is pulled high to activate the chip.	0
t_H	*Hold time* is the time reserved for the chip to read the strapping pin values after CHIP_PU is already high and before these pins start operating as regular IO pins.	3

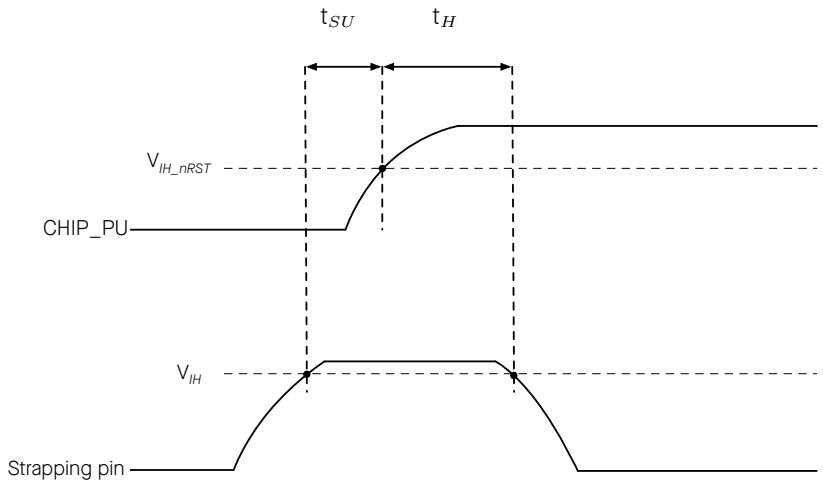


Figure 3-1. Visualization of Timing Parameters for the Strapping Pins

3.1 Chip Boot Mode Control

GPIO60 and GPIO61 control the boot mode after the reset is released. See Table 3-3 *Chip Boot Mode Control*.

Table 3-3. Chip Boot Mode Control

Boot Mode	GPIO61	GPIO60
SPI boot mode[1]	**1**	Any value
Joint download boot mode [2]	0	0

[1] **Bold** marks the default value and configuration.

[2] Joint Download Boot mode supports the following download methods:

- USB-Serial-JTAG Download Boot
- USB-OTG Download Boot
- UART Download Boot
- GPSPI Download Boot

In addition to SPI Boot and Joint Download Boot modes, ESP32-S31 also supports SPI Download Boot mode.

3.2 Secure Debug Controller (SDC)

ESP32-S31 is the first chip to support the Secure Debug Controller (SDC).

When all of the following conditions are met, the chip supports SDC-related commands in Joint Download Boot mode:

- The chip has entered Joint download boot mode
- eFuse bit EFUSE_RMA_ENA has been burned

Note: SDC-related capabilities are not affected by eFuse bits EFUSE_DIS_DOWNLOAD_MODE and EFUSE_ENABLE_SECURE_DOWNLOAD.

Before using SDC, complete eFuse and certificate configuration as follows:

1. Burn the SDC public-key hash into eFuse;
2. In Joint Download Boot mode, transfer the SDC certificate to the chip via esptool; the ROM verifies the certificate in hardware.

After successful certificate verification, the ROM temporarily enables the following debugging capabilities at the hardware level (even if the corresponding eFuse bits have been burned to disable them):

- Software JTAG, even if EFUSE_SOFT_DIS_JTAG has been burned;
- Download mode, even if EFUSE_DIS_DOWNLOAD_MODE has been burned;
- Forced entry into SPI Boot mode for application software debugging.

The SDC authorization state is valid only for the current run. After power-on reset, the verification result is cleared; to debug again, repeat certificate transfer and verification.

3.3 ROM Messages Printing Control

During the boot process, the messages by the ROM code can be printed to:

- **(Default) UART0 and USB Serial/JTAG controller**
- USB Serial/JTAG controller
- UART0

EFUSE_UART_PRINT_CONTROL and GPIO60 control ROM messages printing to **UART0** as shown in Table 3-4 *UART0 ROM Message Printing Control*.

Table 3-4. UART0 ROM Message Printing Control

UART0 ROM Code Printing	EFUSE_UART_PRINT_CONTROL	GPIO60
Enabled	**0**	Ignored
	1	0
	2	1
Disabled	1	1
	2	0
	3	Ignored

[1] **Bold** marks the default value and configuration.

EFUSE_DIS_USB_SERIAL_JTAG_ROM_PRINT controls the printing to **USB Serial/JTAG controller** as shown in Table 3-5 *USB Serial/JTAG ROM Message Printing Control*.

Table 3-5. USB Serial/JTAG ROM Message Printing Control

USB Serial/JTAG ROM Code Printing	EFUSE_DIS_USB_SERIAL_JTAG_ROM_PRINT
Enabled	**0**
Disabled	1

[1] **Bold** marks the default value and configuration.

3.4 JTAG Signal Source Control

The strapping pin GPIO37 can be used to control the source of JTAG signals during the early boot process. This pin does not have any internal pull resistors and the strapping value must be controlled by the external circuit that cannot be in a high impedance state.

As Table 3-6 *JTAG Signal Source Control* shows, GPIO37 is used in combination with EFUSE_DIS_PAD_JTAG, EFUSE_DIS_USB_JTAG, and EFUSE_JTAG_SEL_ENABLE .

Table 3-6. JTAG Signal Source Control

JTAG Signal Source	EFUSE_DIS_PAD_JTAG	EFUSE_DIS_USB_JTAG	EFUSE_JTAG_SEL_ENABLE	GPIO37
USB Serieal/JTAG Controller	**0**	**0**	**0**	Ignored
	0	0	1	1
	1	0	Ignored	Ignored
JTAG pins [2]	0	0	1	0
	0	1	Ignored	Ignored
JTAG is disabled	1	1	Ignored	Ignored

[1] **Bold** marks the default value and configuration.

[2] JTAG pins refer to MTDI, MTCK, MTMS, and MTDO.

4 Functional Description

4.1 System

This section describes the core of the chip's operation, covering its microprocessor, memory organization, system components, and security features.

4.1.1 Microprocessor and Master

This subsection describes the core processing units within the chip and their capabilities.

4.1.1.1 High-Performance CPU

ESP32-S31 has a high-performance 32-bit RISC-V dual-core processor with the following features:

Feature List

- Five-stage pipeline that supports clock frequency of up to 320 MHz
- RV32IMAFC ISA (instruction set architecture)
- Zc extensions (Zcb, Zcmp, and Zcmt)
- Zb extensions
- Custom AI and DSP extension (XespV)
- Custom hardware loop instructions (XespLoop)
- Compliant with RISC-V Sv32 virtual memory scheme
- Compliant with RISC-V Core Local Interrupt (CLINT)
- Compliant with RISC-V Core-Local Interrupt Controller (CLIC)
- Branch predictor BHT, BTB, and RAS
- Up to four hardware breakpoints/watchpoints
- Up to 32 PMP regions and 16 PMA regions
- Machine, Supervisor and User privilege modes
- USB/JTAG for debugging
- Compliant with RISC-V debug specification v0.13
- Offline trace debug that is compliant with RISC-V Trace Specification v2.0

4.1.1.2 RISC-V Trace Encoder (TRACE)

The RISC-V Trace Encoder in the ESP32-S31 chip provides a way to capture detailed trace information from the High-Performance CPU's execution, enabling deeper analysis and optimization of the system. It connects to the HP CPU's instruction trace interface and compresses the information into smaller packets, which are then stored in internal SRAM.

Feature List

- Compatible with Efficient Trace for RISC-V v2.0
- Delta address mode and full address mode
- A filter unit
- Notifying an instruction address via debug trigger or filter unit
- Support for the following sideband signals to control trace data flow:
  - Debugging trigger to start or end encoder
  - When the hart is halted, the encoder can report the last packet and then stop
  - When the hart is reset, the encoder can report the last packet and then stop
  - Stalling the hart when FIFO is almost full
- Arbitrary address range of the trace memory size
- Configurable synchronization modes:
  - Synchronization counter counts by packet
  - Synchronization counter counts by cycle
  - Synchronization counter can be disabled
- Trace lost status to indicate packet loss
- Automatic restart after packet loss
- Memory writing in the loop or non-loop mode
- Two interrupts:
  - Triggered when the packet size exceeds the configured memory space
  - Triggered when a packet is lost
- FIFO (128 × 8 bits) to buffer packets
- AHB burst transmission with configurable burst length

4.1.1.3 Low-Power CPU

ESP32-S31 integrates an LP 32-bit RISC-V single-core processor. This LP CPU is designed as a simplified, low-power replacement of HP CPU in sleep modes. It can be also used to supplement the functions of the HP CPU in normal working mode. The LP CPU and LP memory remain powered on in Deep-sleep mode. Hence, the developer can store a program for the LP CPU in the LP memory to access LP IO, LP peripherals, and real-time timers in Deep-sleep mode.

Feature List

- Two-stage pipeline that supports a clock frequency of up to 40 MHz
- RV32IMAC ISA (instruction set architecture)
- 18 vector interrupts

- Debug module compliant with RISC-V External Debug Support Version 0.13 with external debugger support over an industry-standard JTAG/USB port
- Hardware trigger compliant with RISC-V External Debug Support Version 0.13 with up to 2 breakpoints/watchpoints
- Core performance metric events
- Can wake up the HP CPU or trigger an interrupt to the HP CPU
- Access to HP memory and LP memory
- Access to the entire peripheral address space

4.1.2 System DMA

This subsection describes the system DMA.

4.1.2.1 GDMA Controller (GDMA-AHB, GDMA-AXI)

General Direct Memory Access (GDMA) is a feature that allows peripheral-to-memory, memory-to-peripheral, and memory-to-memory data transfer at high speed. The CPU is not involved in the GDMA transfer and therefore is more efficient with less workload.

ESP32-S31 has two types of general-purpose DMA controllers, namely GDMA-AHB and GDMA-AXI, to directly access the AHB bus or the AXI bus respectively. One GDMA-AHB is placed on both the HP and LP sides.

Feature List

- Architecture:
  - GDMA-AHB: AHB bus architecture
  - GDMA-AXI: AXI bus architecture, which gives the possibility to complete up to eight transactions out of order and up to eight outstanding transactions
- Programmable length of data to be transferred in bytes
- Access via any address and size
- Alignment:
  - GDMA-AHB:
    * Descriptor address: 1-word aligned
    * Data address and length:
      · Internal memory and non-encrypted external memory address space: no requirements
      · Encrypted external memory address space: 16-byte aligned
  - GDMA-AXI:
    * Descriptor address: 2-word aligned
    * Data address and length:
      · Internal memory and non-encrypted external memory address space: no requirements

  - Encrypted external memory address space: 16-byte aligned

- Linked list of descriptors
- INCR4/INCR8/INCR16 burst transfers when accessing memory via GDMA-AHB
- The GDMA-AHB on the HP side has 5 transmit channels and 5 receive channels; the GDMA-AHB on the LP side has 2 transmit channels and 2 receive channels
- Software-configurable selection of peripheral requesting its service
- Configurable channel priority and weight arbitration
- Support for memory transfer
- Linked list switch interrupt mechanism (only supported by GDMA-AXI)

4.1.2.2 2D-DMA Controller (2D-DMA)

The 2D-DMA controller is a DMA (Direct Memory Access) dedicated to two-dimensional image processing. In addition to all the features of GDMA-AXI, it includes support for macroblock reordering and color space conversion (CSC) to better meet the data transfer requirements from JPEG and PPA. Notably, the 2D-DMA facilitates memory-to-memory transfers, enabling the movement of macroblocks between different segments of memory address space while concurrently performing color space conversion.

Feature List

- One AXI master interface
- Data transfer with unaligned starting addresses
- Memory-to-memory, peripheral-to-memory (RX), and memory-to-peripheral (TX) data transfer
- Four memory-to-peripheral channels, and three peripheral-to-memory channels
- Support for PPA and JPEG Codec
- Macroblock reordering
- Color space conversion
- Configurable channel priority and weight

4.1.3 Memory Organization

This subsection describes the memory arrangement to explain how data is stored, accessed, and managed for efficient operation.

Figure 4-1 illustrates the address mapping structure of ESP32-S31.

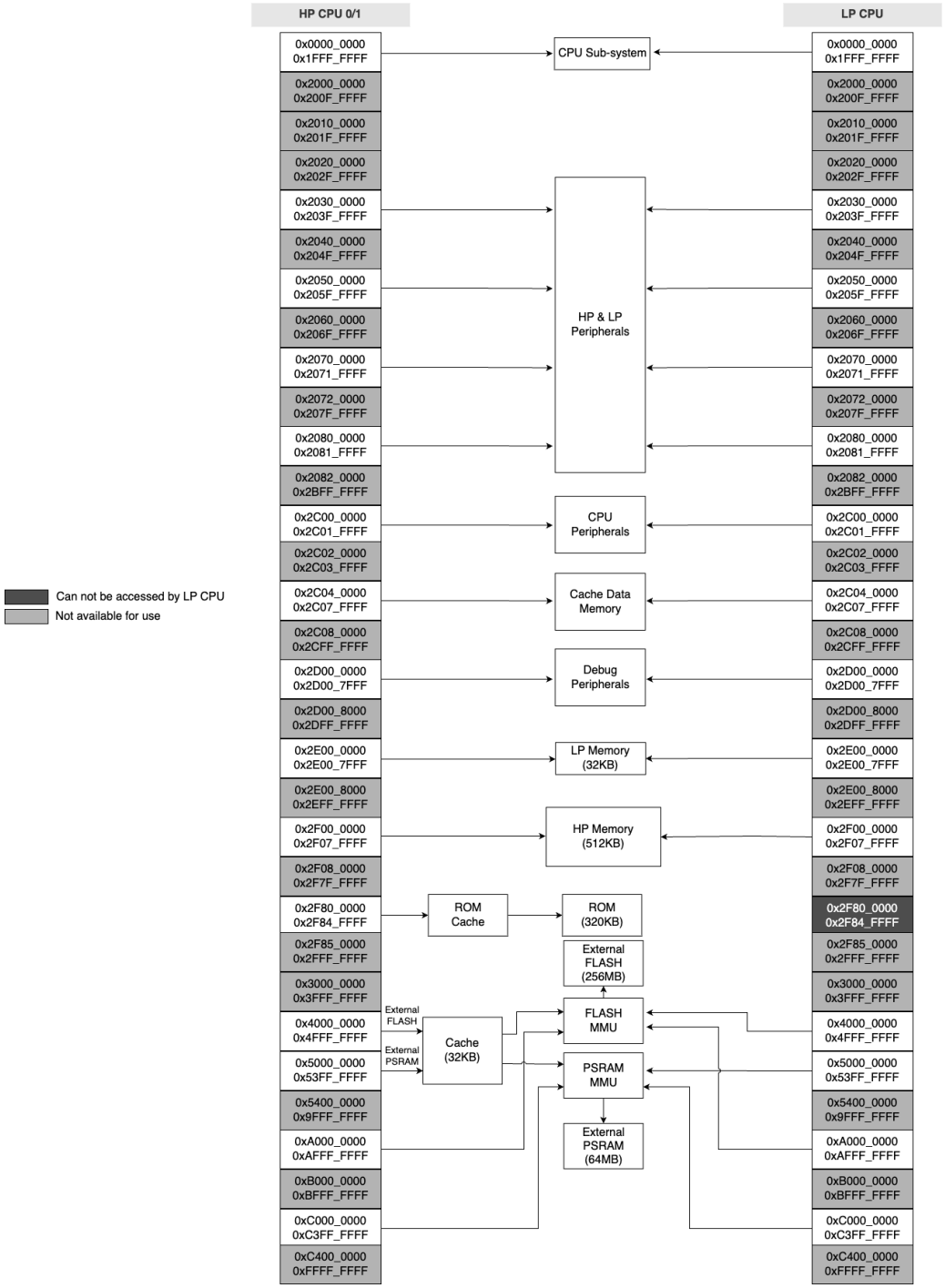


Figure 4-1. Address Mapping Structure

4.1.3.1 Internal Memory

The internal memory of ESP32-S31 refers to the memory integrated on the chip die or in the chip package, including ROM, SRAM, eFuse, and flash.

Feature List

- 320 KB of ROM for booting and core functions
- 512 KB of SRAM for data and instructions

- 32 KB of low-power SRAM (LP SRAM) that can be accessed by HP CPU or LP CPU. It can retain data in Deep-sleep mode
- 4 Kbit of eFuse memory, with 1536 bits available for users.

4.1.3.2 External Memory

ESP32-S31 supports SPI, Dual SPI, Quad SPI, and QPI interfaces that allow connection to the external flash and PSRAM.

CPU's instruction memory space and read-only data memory space can map into the external flash and PSRAM of ESP32-S31. External flash supports up to 256 MB, while PSRAM supports up to 64 MB. ESP32-S31 supports hardware encryption/decryption based on XTS-AES to protect developers' programs and data in the external flash and PSRAM.

Feature List

- The 256 MB flash instruction space is mapped to external flash in 256 KB blocks, while the 64 MB PSRAM instruction space is mapped to external PSRAM in 64 KB blocks. Both support 32-bit instruction fetch.
- The 256 MB flash data space is mapped to external flash in 256 KB blocks, while the 64 MB PSRAM data space is mapped to external PSRAM in 64 KB blocks. External flash supports 8-bit, 16-bit, and 32-bit reads, while PSRAM supports 8-bit, 16-bit, and 32-bit read and write operations.

Note:
After ESP32-S31 is initialized, software can customize the mapping of external flash and PSRAM into the CPU address space.

4.1.3.3 eFuse Controller (eFuse)

The eFuse memory is a one-time programmable memory that stores parameters and user data, and the eFuse controller of ESP32-S31 is used to program and read this eFuse memory.

Feature List

- 4 Kbits in total, with 1536 bits reserved for users, e.g., encryption key and device ID
- One-time programmable storage
- Configurable write protection
- Configurable read protection
- Various hardware encoding schemes to protect against data corruption

4.1.3.4 Cache

The ESP32-S31 adopts a cache architecture in which each CPU (Core0 and Core1) has a private instruction cache, while both CPUs share a unified data cache.

Feature List

- 32 KB L1 instruction cache, 64 B block size, two-way set associative
- 64 KB L1 data cache, 64 B block size, two-way set associative, supporting write-through and write-back policies
- Cacheable and non-cacheable access
- Pre-load function
- Lock function
- Critical word first and early restart

4.1.4 System Components

This subsection describes the essential components that contribute to the overall functionality and control of the system.

4.1.4.1 GPIO Matrix and IO MUX

The ESP32-S31 chip features 60 GPIO pins, including 8 low-power (LP) GPIO pins and 52 high-performance (HP) GPIO pins. Each pin can be used as a general-purpose I/O, or be connected to an internal peripheral signal.

- Through HP GPIO matrix and HP IO MUX, HP peripheral input signals can be from any GPIO pins, and HP peripheral output signals can be routed to any GPIO pins.
- Through LP GPIO matrix and LP IO MUX, LP peripheral input signals can be from any LP GPIO pins, and LP peripheral output signals can be routed to any LP GPIO pins.

Together these modules provide highly configurable I/O. The 60 GPIO pins are numbered as follows: GPIO0 GPIO28, GPIO30 GPIO40, GPIO42 GPIO61.

- LP GPIO pins (GPIO0–GPIO7) can be used by either HP or LP peripherals.
- HP GPIO pins (GPIO8–GPIO28, GPIO30–GPIO40, GPIO42–GPIO61) can be used only by HP peripherals.

Feature List

HP GPIO matrix has the following features:

- A full-switching matrix between HP peripheral input/output signals and the GPIO pins
- 243 HP peripheral input signals sourced from the input of any GPIO pins
- 247 HP peripheral output signals routed to the output of any GPIO pins
- Signal synchronization for HP peripheral inputs based on HP IO MUX operating clock
- GPIO Filter hardware for input signal filtering
- Glitch Filter hardware for second-time filtering on input signal
- Sigma delta modulated (SDM) output
- GPIO simple input and output

- HP GPIO Wakeup

HP IO MUX has the following features:

- Control of 60 GPIOs (GPIO0–GPIO28, GPIO30–GPIO40, GPIO42–GPIO61) for HP peripherals.
- A configuration register provided for each GPIO pin, to control the pin's input/output, pull-up/pull-down, drive strength, and function selection.
- Better high-frequency digital performance achieved by routing some digital signals (SPI, EMAC) directly from HP IO MUX to peripherals.

LP GPIO matrix has the following features:

- A full-switching matrix between the LP peripheral input/output signals and the LP GPIO pins
- 9 LP peripheral input signals sourced from the input of any LP GPIO pins
- 26 LP peripheral output signals routed to the output of any LP GPIO pins
- GPIO Filter hardware for input signal filtering
- GPIO simple input and output
- LP GPIO Wakeup

LP IO MUX has the following features:

- Control of 8 LP GPIO pins (GPIO0–GPIO7) for LP peripherals.
- A configuration register provided for each LP GPIO pin, to control the pin's input/output, pull-up/pull-down, drive strength, function selection, and IO MUX selection.

4.1.4.2 Reset

ESP32-S31 provides four types of reset that occur at different levels, namely CPU Reset, Core Reset, System Reset, and Chip Reset. All reset types mentioned above (except Chip Reset) preserve the data stored in internal memory.

- Four reset types:
  - CPU Reset: resets CPU core. HP CPU0, HP CPU1, and LP CPU can be reset independently:
    - HP CPU0 will be automatically released from reset after chip power-up.
    - HP CPU1 is at reset by default after chip power-up, and needs to be manually released from reset.
    - LP CPU is at reset after chip power-up, and needs to be manually released from reset by configuring the power management unit (PMU).
  - Core Reset: resets the whole digital system except for LP AON. HP core and LP core can be reset independently: HP Core Reset resets HP CPU0, HP CPU1, HP peripherals, HP GPIO, etc., and LP Core Reset resets LP CPU and LP peripherals.
  - System Reset: resets the whole digital system, including the LP system.
  - Chip Reset: resets the whole chip.
- Software reset and hardware reset:

  - Software Reset: triggered via software by configuring the corresponding registers of CPU.
  - Hardware Reset: triggered directly by the hardware.

4.1.4.3 Clock

ESP32-S31 clocks are mainly sourced from oscillator (OSC, including Resistor-Capacitor circuit), crystal (XTAL), and PLL circuit, and then processed by the dividers or selectors, which allows most functional modules to select their working clock according to their power consumption and performance requirements.

ESP32-S31 clock sources can be categorized based on frequency as follows:

- High-performance clocks, primarily used to provide operating clocks for HP CPU0/1 and HP digital peripherals:
  - BBPLL_CLK: 480 MHz internal PLL clock (reference clock: XTAL_CLK)
  - CPLL_CLK: 320 MHz internal PLL clock (reference clock: XTAL_CLK)
  - MPLL_CLK: 500 MHz internal PLL clock (reference clock: XTAL_CLK)
  - APLL_CLK: 120 MHz internal PLL clock (reference clock: XTAL_CLK)
  - XTALX2_CLK: 80 MHz clock (reference clock: XTAL_CLK); used to provide a relatively high-speed clock for the HP domain when PLLs are not enabled
- Low-power clocks, primarily used for the low-power system and peripherals operating in low-power modes:
  - XTAL32K_CLK: 32 kHz external crystal oscillator clock
  - RC_SLOW_CLK: internal slow RC oscillator with adjustable frequency (typically 150 kHz)
  - EXT32K_CLK: external clock from a PAD, typically 32 kHz with limited accuracy
  - XTAL_CLK: 40 MHz external crystal oscillator clock
  - RC_FAST_CLK: internal fast RC oscillator with adjustable frequency (default 17.48 MHz)

4.1.4.4 Interrupt Matrix

ESP32-S31 interrupt matrix is used to route interrupt requests generated by peripherals and events to CPU interrupts.

Feature List

- Accepts 155 external interrupt sources as inputs
- Generates 32 external interrupts for the CPU as outputs
- Supports querying the current status of external interrupt sources
- Supports mapping multiple interrupt sources to a single CPU interrupt (i.e., shared interrupts)
- Supports interrupt delegation (lower-level interrupts can be remapped to higher privilege modes for CPU handling):
  - Supports User Mode interrupts being remapped to Supervisor Mode

- Supports User Mode interrupts being remapped to Machine Mode
- Supports Supervisor Mode interrupts being remapped to Machine Mode
- Supports error detection for interrupt delegation configuration

4.1.4.5 Event Task Matrix

The Event Task Matrix (ETM) peripheral contains 50 configurable channels. Each channel can map an event of any specified peripheral to a task of any specified peripheral. In this way, peripherals can be triggered to execute specified tasks without CPU intervention.

Feature List

- Receive various events from multiple peripherals
- Generate various tasks for multiple peripherals
- 50 independently configurable ETM channels
- An ETM channel can be set up to receive any event, and map it to any task
- Each ETM channel can be enabled independently. If not enabled, the channel will not respond to the configured event and generate the task mapped to that event
- Support for checking event and task status
- Peripherals supporting ETM include GPIO, LED PWM, general-purpose timers, RTC Timer, system timer, MCPWM, temperature sensor, ADC, I2S, LP CPU, GDMA-AHB, GDMA-AXI, 2D DMA, and PMU

4.1.4.6 Power Management Unit

The ESP32-S31 has an advanced Power Management Unit (PMU). It can be flexibly configured to power up different power domains of the chip to achieve the best balance between chip performance, power consumption, and wakeup latency.

The integrated LP CPU allows the ESP32-S31 to operate in Deep-sleep mode with most of the power domains turned off, thus achieving extremely low-power consumption.

Configuring the PMU is a complex procedure. To simplify power management for typical scenarios, there are the following **predefined power modes** that power up different combinations of power domains:

- **Active mode** – The HP CPU, RF circuits, and all peripherals are on. The chip can process data, receive, transmit, and listen.
- **Modem-sleep mode** – The HP CPU is on, but the clock frequency can be reduced. The wireless connections can be configured to remain active as RF circuits are periodically switched on when required.
- **Light-sleep mode** – The HP CPU stops running, and can be optionally powered on. The LP peripherals, as well as the LP CPU can be woken up periodically by the timer. The chip can be woken up via all wake up mechanisms: MAC, RTC timer, or external interrupts. Wireless connections can remain active. Some groups of digital peripherals can be optionally powered off.
- **Deep-sleep mode** – Only the LP system is powered on. Wireless connection data is stored in LP memory.

4.1.4.7 System Timer

The System Timer (SYSTIMER) in the ESP32-S31 chip is a 52-bit timer that can be used to generate tick interrupts for the operating system or as a general timer to generate periodic or one-time interrupts.

Feature List

- Two 52-bit counters and three 52-bit comparators
- Counters with an average clock frequency of 16 MHz
- Three types of independent interrupts generated according to alarm value
- Two alarm modes: target mode and period mode
- 52-bit alarm values and 26-bit alarm periods
- Automatic reload of counter value
- Counters can be stalled if the CPU is stalled or in on-chip-debugging mode
- Real-time alarm events

4.1.4.8 Timer Group (TIMG)

The Timer Group (TIMG) in the ESP32-S31 chip can be used to precisely time an interval, trigger an interrupt after a particular interval (periodically and aperiodically), or act as a hardware clock. ESP32-S31 has two timer groups, TIMG0 and TIMG1, each consisting of one general-purpose timer and one Main System Watchdog Timer.

Feature List

- 16-bit prescaler
- 54-bit time-base counter programmable to incrementing or decrementing
- Able to read real-time value of the time-base counter
- Halt and resume the time-base counter
- Programmable alarm generation
- Timer value reload (auto-reload at an alarm or a software-controlled instant reload)
- Frequency calculation of slow clock for TIMG0
- Level interrupt generation
- Real-time alarm events
- Support several ETM tasks and events

4.1.4.9 Watchdog Timers (WDT)

ESP32-S31 contains three digital watchdog timers: one in each of the two timer groups (called Main System Watchdog Timers, or MWDT) and one in the LP system (called the RTC Watchdog Timer, or RWDT).

In SPI Boot mode, RWDT and the MWDT in timer group 0 are enabled automatically in order to detect errors that may occur during the flash boot process and facilitate recovery.

ESP32-S31 also has one analog watchdog timer: Super watchdog (SWD). It is an ultra-low-power circuit in analog domain that helps to prevent the system from operating in a sub-optimal state and resets the system if required.

Feature List

- Four stages, each with a separately programmable timeout value and timeout action
- Timeout actions:
  - MWDT: interrupt, HP CPU reset, HP core reset
  - RWDT: interrupt, HP CPU reset, HP core reset, system reset
- Flash boot protection under SPI Boot mode at stage 0:
  - MWDT0: HP core reset upon timeout
  - RWDT: system reset upon timeout
- Write protection that makes WDT register read only unless unlocked
- 32-bit timeout counter
- Clock source:
  - MWDT: PLL_F80M_CLK, RC_FAST_CLK or XTAL_CLK
  - RWDT: LP_DYN_SLOW_CLK

4.1.4.10 RTC Timer

RTC Timer is an important module for implementing low power management of ESP32-S31. Based on a 48-bit readable counter, RTC Timer is mainly used as a system timer in low power mode when the timer peripheral in the HP system is unavailable. It also allows for configuring timer interrupts and logging the time when specific events happen in the system.

Feature List

- 48-bit counter
- Time logging when one of the following events happens:
  - HP system reset
  - CPU enters stall state
  - CPU exits stall state
  - Crystal powers up
  - Crystal powers down
- Time logging through register configuration
- Occurrence time cached of the most recent two specific events

- Generation of interrupts at target times, which are configurable. It is also possible to configure two target times simultaneously.
- Uninterrupted operation during any reset or sleep mode, except for power-on reset of LP system.
- Supports counter ticks that trigger the system timer to lock time.

4.1.4.11 Permission Control (PMS)

The Permission Control module in ESP32-S31 is responsible for managing access permissions to memory and peripheral registers. It consists of two parts: PMP (Physical Memory Protection) and APM (Access Permission Management).

Feature List

- access permission management for ROM, HP memory, HP peripheral, and LP peripheral address spaces
- APM supports each master (such as DMA) to select one of the four security modes
- access permission configuration for up to 32 address ranges
- individual permission configuration for each register
- Interrupt function and exception information record

4.1.4.12 System Registers

The System Registers in the ESP32-S31 chip are used to configure various auxiliary chip features.

Feature List

- Controls external memory encryption
- Controls HP/LP core debugging
- Controls bus timeout protection
- Software interrupts
- Memory power management
- Clock control
- PAD BIST control

4.1.4.13 Debug Assistant

The Debug Assistant provides a set of functions to help locate bugs and issues during software debugging. It offers various monitoring capabilities and logging features to assist in identifying and resolving software errors efficiently.

Feature List

- **Region read/write monitoring**: Monitors whether the High-Performance dual-core CPU (HP CPU0 and HP CPU1) bus reads from or writes to a specified memory address space. A detected read or write in the monitored address space triggers an interrupt.

- **Stack pointer (SP) monitoring**: Monitors whether the SP exceeds the specified address space. A bounds violation triggers an interrupt.
- **Program counter (PC) logging**: Records the PC value. The last PC value at the most recent reset of the HP CPU*n* (Below, *n* = 0, 1) can be retrieved.
- **Lockup monitoring**: Monitors whether HP CPU*n* encounters a lockup, and records the cause of the first exception, the trap value (tval), instruction PC (iaddr), and current privilege level (priv) when the first abnormal event occurs.
- **Bus access logging**: Records information about bus access. When the HP CPU*n* or the Direct Memory Access controller (DMA) writes a specified value, the Debug Assistant module records the data type, address of the write operation, and the PC value when the write is performed by the HP CPU*n*, then stores this information in the HP SRAM.

4.1.4.14 LP Mailbox

ESP32-S31 integrates an LP Mailbox module that provides an efficient inter-core communication mechanism between the LP CPU and HP CPU0/1. The LP Mailbox module comprises sixteen 32-bit message registers that the LP CPU and HP CPU0/1 can use to store and exchange messages. Inter-core communication between LP CPU and HP CPU0/1 is achieved through an interrupt mechanism implemented in the LP Mailbox module.

Feature List

- Sixteen 32-bit message registers for inter-core communication
- LP CPU external interrupt signal
- HP CPU0/1 external interrupt signal

4.1.4.15 Brown-out Detector

With the Brown-out detector, ESP32-S31 monitors the voltage levels of pin VDDPST. If the voltage on these pins drops below the predefined threshold (defaulting to 2.4 V), the detector triggers signals to shut down certain power-consuming blocks (e.g., flash), ensuring that the digital module has sufficient time to save and transfer important data.

Feature List

- Monitors the voltage level of pin VDDPST
- Two configurable monitoring modes
  - Mode 0: The brown-out detector triggers interrupts when the brown-out counter reaches the predefined threshold and selects the reset mode according to the configuration.
  - Mode 1: The brown-out detector triggers a system reset when the voltage falls below the threshold.
- Configurable voltage-monitoring thresholds and noise tolerance

4.1.5 Cryptography and Security Component

This subsection describes the security features incorporated into the chip, which safeguard data and operations.

4.1.5.1 AES Accelerator (AES)

ESP32-S31 integrates a hardware AES (Advanced Encryption Standard) accelerator that performs data encryption and decryption using the AES algorithm. It supports two working modes: typical AES and DMA-AES. Overall, compared with software-based AES computation, the hardware AES accelerator significantly improves processing speed. In addition, the ESP32-S31 AES accelerator includes a configurable anti–side-channel attack (anti-DPA) feature, providing enhanced security.

Feature List

- Typical AES working mode
  - AES-128/AES-256 encryption and decryption operations compliant with NIST FIPS 197
- DMA-AES working mode
  - AES-128/AES-256 encryption and decryption operations compliant with NIST FIPS 197
  - Block cipher mode compliant with NIST SP 800-38A
    - ECB (Electronic Codebook)
    - CBC (Cipher Block Chaining)
    - OFB (Output Feedback)
    - CTR (Counter)
    - CFB8 (8-bit Cipher Feedback)
    - CFB128 (128-bit Cipher Feedback)
  - GCM (Galois/Counter Mode)
  - Interrupt on completion of computation
- Configurable anti–side-channel attack (anti-DPA) capability

4.1.5.2 ECC Accelerator (ECC)

Elliptic Curve Cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves. ECC allows smaller keys compared to RSA cryptography while providing equivalent security.

ESP32-S31's ECC Accelerator can complete various calculations based on different elliptic curves, thus accelerating the ECC algorithm and ECC-derived algorithms (such as ECDSA).

Feature List

- three elliptic curves, namely P-192, P-256 and P-384 defined in FIPS 186-3

- Supports the SM2 cryptographic algorithm (as defined in SM2 Elliptic Curve Public Key Cryptography Algorithm)
- two coordinate systems, namely Affine Coordinates and Jacobian Coordinates
- different point operations, including point addition, point multiplication, and point verification
- different modular operations based on the order or mod base of the curve, including mod addition, mod subtraction, mod multiplication, and mod division
- interrupt upon completion of calculation
- secure operating mode for Base Point Multiplication within a specified time frame

4.1.5.3 HMAC Accelerator (HMAC)

The HMAC Accelerator (HMAC) module is designed to compute Message Authentication Codes (MACs) using the SHA-256 Hash algorithm and keys as described in RFC 2104. It provides hardware support for HMAC computations, significantly reducing software complexity and improving performance.

Feature List

- standard HMAC-SHA-256 algorithm
- hash result only accessible by configurable hardware peripheral (in downstream mode)
- compatibility with challenge-response authentication algorithm
- required keys for the Digital Signature (DS) peripheral (in downstream mode)
- re-enabled soft-disabled JTAG (in downstream mode)

4.1.5.4 RSA Accelerator (RSA)

The RSA accelerator provides hardware support for high-precision computation used in various RSA asymmetric cipher algorithms, significantly reducing the operation time and software complexity. Compared with RSA algorithms implemented solely in software, this hardware accelerator speeds up RSA algorithms significantly. The RSA accelerator also supports operands of different lengths, which provides more flexibility during the computation.

Feature List

- Large-number modular exponentiation with two optional acceleration options
- Large-number modular multiplication, up to 4096 bits
- Large-number multiplication, with operands up to 2048 bits
- Operands of different lengths
- Interrupt on completion of computation

4.1.5.5 SHA Accelerator (SHA)

ESP32-S31 integrates an SHA accelerator, which is a hardware device that speeds up the SHA algorithm significantly, compared with an SHA algorithm implemented solely in software. The SHA accelerator integrated

in ESP32-S31 has two working modes, Typical SHA and DMA-SHA.

Feature List

- The following hash algorithms introduced in FIPS PUB 180-4 Spec.
  - SHA-1
  - SHA-224
  - SHA-256
  - SHA-384
  - SHA-512
  - SHA-512/224
  - SHA-512/256
  - SHA-512/t
- Supports SM3 cryptographic hash algorithm
- Two working modes
  - Typical SHA
  - DMA-SHA
- Interleaved function when working in Typical SHA working mode
- Interrupt function when working in DMA-SHA working mode

4.1.5.6 RSA Digital Signature Peripheral (RSA_DS)

Digital signature technology uses cryptographic algorithms to verify the authenticity and integrity of messages. It can also be used to authenticate a device to a server or to verify whether a message has been tampered with.

ESP32-S31 includes an RSA Digital Signature Peripheral (RSA_DS) that provides hardware acceleration for efficiently generating RSA-based digital signatures. The RSA_DS peripheral uses the RSA_DS_KEY (generated by HMAC or provisioned by the key manager) to decrypt pre-encrypted parameters and compute the signature. All these operations occur entirely in hardware. During the process, sensitive data such as the key for decrypting RSA parameters, the input/output keys of the HMAC key derivation function, or keys provisioned by the key manager remain inaccessible to the user.

Feature List

- RSA digital signatures with key length up to 4096 bits
- Encrypted private key data, only decryptable by the RSA_DS peripheral
- SHA-256 digest to protect private key data against tampering by an attacker

4.1.5.7 ECDSA Digital Signature Peripheral (ECDSA_DS)

In cryptography, the Elliptic Curve Digital Signature Algorithm (ECDSA) is an analog of the Digital Signature Algorithm (DSA) which uses elliptic-curve cryptography.

ESP32-S31's ECDSA accelerator efficiently computes signatures while ensuring the confidentiality of the signing process to prevent information leakage. It provides strong security guarantees without impacting performance, making it suitable for high-speed cryptographic operations and protecting user data.

Feature List

- Digital signature generation and signature verification
- Three NIST elliptic curves, namely P-192, P-256, and P-384 (as defined in the FIPS 186-5 Specification) and Chinese SM2 algorithm (for detailed definitions, see Public Key Cryptographic Algorithm SM2 Based on Elliptic Curves)
- Multiple hash algorithms, including SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, and SHA-512/256 (as defined in the FIPS PUB 180-4 Specification) and the Chinese SM3 algorithm (for detailed definitions, see SM3 Cryptographic Hash Algorithm)
- Provides high-security features:
  - Implements dynamic access control under different operating states to prevent key leakage caused by any intermediate data exposure
  - Signature generation and verification are fixed-time operations, resistant to side-channel attacks

4.1.5.8 Secure Debug Controller (SDC)

The Secure Debug Controller (SDC) module implements a hardware-enforced secure debug unlocking protocol for deployed chips. SDC provides the only authorized path to selectively re-enable debug and download interfaces, ensuring chips can enter debug mode only after being returned to authorized service centers. This capability is enabled by eFuse configuration, preventing software-only bypass.

SDC unlocking uses the on-chip SHA and ECDSA_DS peripherals for cryptographic acceleration. The unlock process follows a two-phase challenge-response protocol: in the request phase, SDC calculates a chip-unique CHIP_INFO to generate the certificate; in the verification phase, the certificate is loaded into SDC for hash and signature verification. Any verification failure puts SDC into a terminal lock state, and further attempts require a power cycle.

Feature List

- Three independently selectable re-enable outputs: JTAG re-enable, Download mode, and Force SPI boot
- Two operating modes: Certificate Request and Certificate Verification
- Optional nonce integration for per-session challenge uniqueness
- Hash mode: SHA-256
- Public-key signature mode: ECDSA P-256

4.1.5.9 External Memory Encryption and Decryption (XTS_AES)

The ESP32-S31 integrates an External Memory Encryption and Decryption module that complies with the XTS-AES standard algorithm specified in IEEE Std 1619-2007, providing security for users' application code and data stored in the external memory (flash and RAM). Users can store proprietary firmware and sensitive data (e.g., credentials for gaining access to a private network) in the external flash, or store general data in the external RAM.

Feature List

- General XTS-AES algorithm, compliant with IEEE Std 1619-2007
- Software-based manual encryption
- High-speed auto encryption without software's participation
- High-speed auto decryption without software's participation
- Encryption and decryption functions jointly enabled/disabled by register configuration, eFuse parameters, and boot mode
- Configurable anti–side-channel attack (anti-DPA) capability
- Independent keys for flash and PSRAM respectively

4.1.5.10 Random Number Generator (RNG)

The ESP32-S31 contains a true random number generator (TRNG), which generates 32-bit random numbers that can be used for cryptographical operations, among other things.

The TRNG in ESP32-S31 generates true random numbers, which means random numbers generated from a physical process, rather than by means of an algorithm. No number generated within the specified range is more or less likely to appear than any other number.

Feature List

- RNG entropy source
  - thermal noise from SAR ADC
  - an asynchronous clock mismatch
  - BUF_CHAIN

4.1.5.11 Key Manager

ESP32-S31 stores and deploys keys with the Key Manager as the security core. Key Manager uses the unique physically unclonable function (PUF) of each chip to generate the hardware unique key (HUK) which is unique to the chip and serves as the root of trust (RoT) for the chip. HUK is automatically generated each time the chip is powered on and disappears when the chip is powered off. In this way, Key Manager secures key storage and deployment.

Key Manager of ESP32-S31 stores key information (non-plaintext information for recovering the key) in external memory, realizing flexible key management functionalities such as unlimited key storage and dynamic key switching.

Feature List

The HUK Generator has the following features:

- HUK Generation Mode:
  - Generates a new HUK and its recovery information
- HUK Recovery Mode:
  - Recovers a deployed HUK with its recovery information
- Prompt for HUK recovery error
- Prompt for HUK risk level

The Key Manager has the following features:

- Unlimited number of keys
- Specified private key deployment (AES Deploy Mode):
  - Users specify the value of the key
- Negotiated private key deployment (ECDH0 Deploy Mode):
  - Highest security mode: there is no need to worry about the leaks of data in external channels
  - Requires chip initiation to obtain the private key
  - Negotiates the key value between each chip and the user
- Negotiated private key deployment (ECDH1 Deploy Mode):
  - Provides auxiliary key software/scripts to users
  - No need to boot the chip to obtain the private key
- Random key deployment (Random Deploy Mode):
  - Deploys a hardware-generated random key with nobody knowing the exact value
- Private key recovery deployment (Private Key Recovery Mode):
  - Recovers exactly the same key by entering the key information generated during deployment
- Key information export (key_info Export Mode):
  - Generates unique key information for the same key each time

4.1.5.12 Power Glitch Detector

ESP32-S31 can monitor the voltage of the power supply in real time. When a voltage glitch occurs, the chip will reset immediately to prevent power glitch attacks.

Feature List

- Detect four supply voltages, including VDDPST and VDDA
- Configurable threshold for power glitch (around 2.4 V by default)
- Can be enabled through eFuse at power-up

4.1.5.13 Secure Boot

The Secure boot feature in the ESP32-S31 chip ensures that only the signed firmware can be booted.

Feature List

- Supported signature types
  - ECDSA P-192 signature
  - ECDSA P-256 signature
  - ECDSA P-384 signature

For details, see ESP-IDF Programming Guide > Secure Boot v2.

4.2 Peripherals

This section describes the chip's peripheral capabilities, covering connectivity interfaces and on-chip sensors that extend its functionality.

4.2.1 Image and Voice Processing

This subsection describes the peripherals for image and voice processing.

4.2.1.1 JPEG Codec

ESP32-S31's JPEG codec is an image codec, which is based on the JPEG baseline standard, for compressing (encoding) and decompressing (decoding) images to reduce the bandwidth required to transmit images or the space required to store images, making it possible to process large-resolution images.

Feature List

When used as an encoder, the JPEG codec has the following features:

- Integrated discrete cosine transform algorithm
- Integrated canonical Huffman coding
- RGB888, RGB565, YUV444, YUV422, YUV420 and GRAY as original input image formats
- Supports converting (if needed) and compressing RGB888, RGB565, or YUV444 images into YUV444, YUV422, or YUV420 formats, and supports converting (if needed) and compressing YUV422 images into YUV422 or YUV420 formats. Compression is only available for YUV444, YUV422, and YUV420 formats
- Four configurable quantization coefficient tables with 8-bit or 16-bit precision
- Performance:
  - Still image compression: up to 4K resolution
  - Dynamic image compression: up to 720P@30fps (excluding header encoding time)
- Automatically added stuffed zero byte
- Automatically added EOI marker

When used as a decoder, the JPEG codec has the following features:

- Integrated inverse discrete cosine transform algorithm
- Integrated Huffman decoding
- Supported image formats for compressed bitstream decoding: YUV444, YUV422, YUV420, and GRAY.
- Four configurable quantization coefficient tables with 8-bit or 16-bit precision
- Two DC and two AC Huffman tables
- Supports image decoding of any resolution. However, the resolution of the output decoded image differs from the format of the input image:
  - YUV444, GRAY: both the horizontal and vertical resolutions of the output decoded image are multiples of 8, i.e., 150 × 150 images with an output resolution of 152 × 152

  - YUV422: the horizontal resolution of the output decoded image is the multiples of 16 and the vertical resolution is multiples of 8, i.e., 150 × 150 images with an output resolution of 160 × 152
  - YUV420: both the horizontal and vertical resolutions of the output decoded image are multiples of 16, i.e., 150 × 150 images with an output resolution of 160 × 160
- Performance:
  - Still image decoding: up to 4K resolution
  - Dynamic image decoding: up to 720P@30fps (excluding header parsing time)

Pin Assignment

The JPEG Codec does not interact directly with IOs, so it has no pins assigned.

4.2.1.2 Pixel-Processing Accelerator (PPA)

ESP32-S31 includes a pixel-processing accelerator (PPA) with scaling-rotation-mirror (SRM) and image blending (BLEND) functionalities.

Feature List

- Image rotation, scaling, and mirroring by SRM:
  - Input formats: ARGB8888, RGB888, RGB565, YUV422, YUV420, GRAY
  - Output formats: ARGB8888, RGB888, RGB565, YUV422, YUV420, GRAY
  - Counterclockwise rotation angles: 0°, 90°, 180°, 270°
  - Horizontal and vertical scaling with scaling factors of 4-bit integer part and 8-bit fractional part
  - Horizontal and vertical mirroring
- Blending two layers of the same size and filling images with specific pixels by BLEND:
  - Foreground input formats: ARGB8888, RGB888, RGB565, L4, L8, A4, A8
  - Background input formats: ARGB8888, RGB888, RGB565, YUV422, YUV420, GRAY, L4, L8
  - Output formats: ARGB8888, RGB888, RGB565, YUV422, YUV420, GRAY
  - Layer blending based on the Alpha channel. If layers lack an Alpha channel, it can be provided through register configuration.
  - Special color filtering by setting color-key ranges of foreground and background layers

Pin Assignment

The pixel-processing accelerator does not directly interact with IOs, so it has no pins assigned.

4.2.1.3 Audio Sample Rate Converter (ASRC)

Audio Sample Rate Converter (ASRC) is an accelerator-type audio processing module used for high-quality conversion of audio signals between different sample rates. It enables collaboration between devices with

varying sample rate standards. This module typically interacts with on-chip or off-chip memory through a DMA interface to achieve efficient transmission and processing of audio data.

Feature List

- Supports conversion between 8 kHz, 16 kHz, 32 kHz, 44.1 kHz, and 48 kHz
- Input and output signal width is 16 bits, using Q1.15 format
- Sample rate conversion uses two integer resamplers and one fractional sample rate converter (FRC) in a cascade, with the cascade configurable via software:
  - The integer resampler's conversion factor can be set to 2 or $\frac{1}{2}$
  - The FRC supports conversion factors configurable between 0 and 2 (excluding 0 and 2)
- Supports multiple channel modes, including:
  - Mono receive (Rx) / Mono transmit (Tx)
  - Stereo receive (Rx) / Stereo transmit (Tx), processed in parallel
  - Mono receive / Stereo transmit
  - Stereo receive / Mono transmit
- Supports input data of any length, and generates EOF (End of Frame) flags based on output length
- When two channels are used independently, each channel supports the following modes:
  - Mono input, mono output
  - Mono input, stereo output (both channels use the same data)
  - Stereo input, mono output (choose one channel for processing)
- When both channels are used together, each channel can be configured as stereo input or output

4.2.1.4 CORDIC Accelerator (CORDIC)

ESP32-S31 includes a hardware CORDIC accelerator. Its core principle is to iteratively approach a target angle by executing a sequence of fixed-angle rotations associated with computation cycles, enabling mathematical operations such as trigonometric function calculations.

Feature List

- Supports q1.15 and q1.31 fixed-point formats
- Supports circular, linear, and hyperbolic systems
- Supports rotation mode and vectoring mode
- Supported functions: sin, cos, sinh, cosh, atan(x), atan(y/x), atanh, modulus, square root, and natural logarithm
- Configurable computation precision (number of operation cycles)
- Supports polling-based result readout
- Supports interrupt-based result readout

- Supports direct DMA connection mode

Pin Assignment

The CORDIC accelerator does not need direct interaction with IO and therefore requires no dedicated pin assignment.

4.2.1.5 LCD and Camera Controller (LCD_CAM)

The LCD and Camera controller (LCD_CAM) on the ESP32-S31, consisting of an independent LCD control module and a camera control module, is a versatile component designed to facilitate interfacing with both LCDs and cameras.

Feature List

- Operation modes:
  - LCD master TX mode
  - Camera slave RX mode
  - Camera master RX mode
- Simultaneous connection to an external LCD and a camera
- External LCD interface:
  - 8/16/24-bit parallel output modes
  - RGB, MOTO6800, and I8080 LCD formats
  - LCD data retrieved from internal memory or external memory via GDMA
- External camera (DVP image sensor) interface:
  - 8/16-bit parallel input modes
  - Camera data stored in internal or external memory via GDMA
- Interrupt support

Pin Assignment

For CAM and LCD interfaces of the Camera-LCD controller, the pins used can be chosen from any GPIOs via the GPIO Matrix.

4.2.2 Connectivity Interface

This subsection describes the connectivity interfaces on the chip that enable communication and interaction with external devices and networks.

4.2.2.1 UART Controller (UART)

The UART controllers in ESP32-S31 are used for asynchronous serial data transmission and reception between the chip and external UART devices. ESP32-S31 includes four UART controllers in the main system and one low-power LP UART.

Feature List

- Programmable baud rates up to 5 MBaud
- RAM shared by TX FIFOs and RX FIFOs
- Support for various lengths of data bits and stop bits
- Parity bit support
- Special character AT_CMD detection
- RS485 protocol support (not applicable to LP UART)
- IrDA protocol support (not applicable to LP UART)
- High-speed data communication using GDMA (not applicable to LP UART)
- Receive timeout feature
- UART as the wake-up source
- Software and hardware flow control

Pin Assignment

UART0 and LP UART each have fixed direct-connected pins via the HP IO MUX and LP IO MUX. They also support mapping to other pins through the GPIO Matrix. UART1, UART2, and UART3 are routed to any HP GPIO pins via the GPIO Matrix.

4.2.2.2 SPI Controller (SPI)

The Serial Peripheral Interface (SPI) is a synchronous serial interface commonly used for communicating with external peripherals. The ESP32-S31 chip integrates four SPI controllers:

- MSPI controller, including two sub-controllers
  - FLASH MSPI controller
    - FLASH MSPI SPI0
    - FLASH MSPI SPI1
  - PSRAM MSPI controller
    - PSRAM MSPI SPI0
    - PSRAM MSPI SPI1
- General Purpose SPI2 (GP-SPI2)
- General Purpose SPI3 (GP-SPI3)
- Low-Power SPI (LP-SPI)

Feature List

GP-SPI has the following features:

- Works as master or as slave

- Half- and full-duplex communications
- CPU- and DMA-controlled transfers
- Various data modes
  - **GP-SPI2**
    * 1-bit SPI mode
    * 2-bit Dual SPI mode
    * 4-bit Quad SPI mode
    * QPI mode
    * 8-bit Octal SPI mode (available only when GP-SPI2 works as a master)
    * OPI mode (available only when GP-SPI2 works as a master)
  - **GP-SPI3**
    * 1-bit SPI mode
    * 2-bit Dual SPI mode
    * 4-bit Quad SPI mode
    * QPI mode
- Configurable module clock frequency
  - Master: up to 80 MHz
  - Slave: up to 60 MHz
- Configurable data length
  - CPU-controlled transfer as master or as slave: 1–64 bytes
  - DMA-controlled single transfer as master: 1–32 KB
  - DMA-controlled configurable segmented transfer as master: data length is unlimited
  - DMA-controlled single transfer or segmented transfer as slave: data length is unlimited
- Configurable bit read/write order
- Independent interrupts for CPU-controlled transfer and DMA-controlled transfer
- Configurable clock polarity and phase
- Four SPI clock modes: mode 0–mode 3
- Multiple CS lines as master
  - **GP-SPI2**: CS0–CS5
  - **GP-SPI3**: CS0–CS2
- Able to communicate with SPI devices, such as a sensor, a screen controller, as well as a flash or RAM chip

LP-SPI is a simplified version of GP-SPI and has a subset of GP-SPI's features:

- Works as a master or as a slave

- Half- and full-duplex communications
- CPU-controlled transfer
- 1-bit SPI data mode
- Configurable module clock frequency:
  - Master: up to 40 MHz
  - Slave: up to 40 MHz
- Configurable data length:
  - CPU-controlled transfer as master or as slave: 1–64 bytes
- Configurable bit read/write order
- Interrupts for CPU-controlled transfer
- Configurable clock polarity and phase
- Four SPI clock modes: mode 0–mode 3
- One CS line as master: CS0
- Wake-up feature as slave (the only new feature compared with GP-SPI)

Pin Assignment

The Flash SPI interface uses the dedicated digital pins 27–33.

The GP-SPI2 controller includes one four-line interface and one eight-line interface. The pins connected to the four-line interface are multiplexed via the IO MUX with GPIO20 to GPIO25, or GPIO50 to GPIO52, as well as the JTAG interface. The pins connected to the eight-line interface are multiplexed with GPIO9 to GPIO19, which also serves as the first set of RMII interface pins for the EMAC. If high-speed performance is not critical for the GP-SPI2 interface, you can select pins from any GPIOs via the GPIO Matrix.

For GP-SPI3, the pins used can be chosen from any GPIOs via the GPIO Matrix.

The pins for the LP-SPI interface can be chosen from any pins via the LP GPIO Matrix.

4.2.2.3 I2C Controller (I2C)

ESP32-S31 has two HP_I2C bus interfaces and one LP_I2C bus interface. The HP_I2C interfaces can operate in either I2C master or slave mode, while the LP_I2C interface supports master mode only.

Feature List

- Standard mode (100 Kbit/s)
- Fast mode (400 Kbit/s)
- Up to 800 Kbit/s (constrained by SCL and SDA pull-up strength)
- 7-bit and 10-bit addressing mode
- Dual address mode
- 7-bit broadcast address

Pin Assignment

For I2C, the pins used can be chosen from any GPIOs via the GPIO Matrix.

4.2.2.4 I2S Controller (I2S)

The I2S Controller in the ESP32-S31 chip provides a flexible communication interface for streaming digital data in multimedia applications, particularly digital audio applications.

Feature List

- Master mode and slave mode
- Full-duplex and half-duplex communications
- Separate TX and RX units that can work independently or simultaneously
- A variety of audio standards supported:
  - TDM Philips standard
  - TDM MSB alignment standard
  - TDM PCM standard
  - PDM standard
- Various TX/RX modes
  - TDM TX mode, up to 16 channels supported
  - TDM RX mode, up to 16 channels supported
  - PDM TX mode
    - Raw PDM data transmission
    - PCM-to-PDM data format conversion, up to 2 channels supported
  - PDM RX mode
    - Raw PDM data reception
- Configurable clock source with frequency up to 96 MHz
- Configurable high-precision sample clock with a variety of sampling frequencies supported
- 8/16/24/32-bit data width
- Synchronous counter in TX mode
- ETM feature
- Direct memory access
- Standard I2S interface interrupts

Pin Assignment

The pins for the I2S controller can be chosen from any GPIOs via the GPIO Matrix.

4.2.2.5 Pulse Count Controller (PCNT)

The Pulse Count controller (PCNT) in ESP32-S31 captures pulses and counts pulse edges in seven modes.

Feature List

- Four independent pulse counters (units) that count from 1 to 65535
- Each unit consists of two independent channels sharing one pulse counter
- All channels have input pulse signals (e.g., sig_ch0_u*n*) with their corresponding control signals (e.g., ctrl_ch0_u*n*)
- Independently filter glitches of input pulse signals (sig_ch0_u*n* and sig_ch1_u*n*) and control signals (ctrl_ch0_u*n* and ctrl_ch1_u*n*) on each unit
- Each channel has the following parameters:
  1. Selection between counting on positive or negative edges of the input pulse signal
  2. Configuration to Increment, Decrement, or Disable counter mode for control of signal's high and low states
  3. Step count alert triggered by setting the upcount/downcount step threshold
  4. Clearing of the pulse count controller value by setting the clear register or sending a clear signal through GPIO input
  5. Generation and recording of a corresponding event signal for each counter mode, with the ability to report it to the interrupt task
- Maximum frequency of pulses: $\frac{f_{APB\_CLK}}{2}$

Pin Assignment

The pins for the Pulse Count controller can be chosen from any GPIOs via the GPIO Matrix.

4.2.2.6 USB 2.0 High-Speed OTG

The ESP32-S31 chip features a USB 2.0 High-Speed On-The-Go peripheral (OTG_HS) with an integrated transceiver. This OTG_HS complies with the USB 2.0 specification, OTG Revision 1.3, and OTG Revision 2.0 specifications. The interface supports USB 2.0 High-Speed mode (480 Mbit/s), Full-Speed mode (12 Mbit/s), and Low-Speed mode (1.5 Mbit/s).

- When OTG_HS operates in High-Speed or Full-Speed modes, it can be configured as either a Host or a Device.
- When OTG_HS operates in Low-Speed mode, it can only be configured as a Host.

Feature List

General Features

- USB 2.0 specification, OTG Revision 1.3 and OTG Revision 2.0 specifications

- High-Speed, Full-Speed, and Low-Speed data rates
- As a host and a device in High-Speed mode and Full-Speed mode
- Dynamic FIFO (DFIFO) sizing, each device EP/host channel can dynamically allocate a maximum of 4 KB FIFO.
- Up to 8 non-periodic transactions and 16 periodic transactions per microframe
- Multiple modes of memory access
  - Scatter/Gather DMA mode
  - Buffer DMA mode
  - Slave Mode
- Integrated UTMI High-Speed transceiver

Device Mode Features

- Endpoint 0 always present, bi-directional, consisting of EP0 IN and EP0 OUT
- 15 additional endpoints 1–15, configurable as IN or OUT
- Maximum of eight IN endpoints concurrently active at any time, including EP0 IN
- All OUT endpoints share a single RX FIFO
- Each IN endpoint has a dedicated TX FIFO

Host Mode Features

- 16 host channels
- RX FIFO: shared by all periodic and non-periodic transactions
- Two TX FIFO:
  - One shared by all non-periodic transactions
  - One shared by all periodic transactions
- All of the above FIFOs share a 4 KB RAM.
- The size of each FIFO is configurable, with a maximum of 4 KB.

Pin Assignment

The pins connected to USB2 OTG PHY DM (USB_D-) and USB2 OTG PHY DP (USB_D+) signals of USB 2.0 High-Speed OTG are dedicated pin 44 and pin 45. Other signals can be routed to any GPIOs via the GPIO matrix.

4.2.2.7 USB Serial/JTAG Controller (USB_SERIAL_JTAG)

ESP32-S31 contains a USB Serial/JTAG controller. This unit can be used to program the SoC's flash, read program output, as well as attach a debugger to the running program. All of these are possible for any computer with a USB host without any active external components.

Feature List

- USB 2.0 full speed compliant, capable of up to 12 Mbit/s transfer speed (note that this controller does not support the faster 480 Mbit/s high-speed transfer mode)
- CDC-ACM virtual serial port and JTAG adapter functionality
- programming the chip's flash
- CPU debugging with compact JTAG instructions
- a full-speed USB PHY integrated in the chip

Pin Assignment

The pins for the USB Serial/JTAG controller are multiplexed with GPIO33 and GPIO34 via IO MUX.

4.2.2.8 Ethernet Media Access Controller (EMAC)

By using the external Ethernet PHY (physical layer), ESP32-S31 can send and receive data via Ethernet MAC (Media Access Controller) according to the IEEE 802.3 standard.

ESP32-S31 Ethernet MAC complies with the following standards:

- IEEE 802.3-2002 for Ethernet MAC
- IEEE 1588-2008 standard for precise networked clock synchronization
- IEEE 802.3 standard Media Independent Interface (MII), Reduced Media Independent Interface (RMII), and Reduced Gigabit Media Independent Interface (RGMII)
- IEEE 802.3az-2010 for Energy Efficient Ethernet
- IEEE 802.1Q for VLAN frame format

Feature List

- Data rates of 10/100/1000 Mbit/s through an external PHY interface
- Communication with an external Ethernet PHY through IEEE 802.3-compliant MII, RMII, or RGMII interface (only one can be used at a time)
- Full-duplex and half-duplex modes
  - Carrier Sense Multiple Access or Collision Detection (CSMA/CD) protocol in half-duplex mode
  - IEEE 802.3x flow control in full-duplex mode
  - Optional forwarding of received pause control frame to the user application in full-duplex mode
  - Back-pressure flow control in half-duplex mode
  - Automatic transmission of zero-quanta pause frame on deassertion of flow control input in full-duplex mode
- Preamble and start-of-frame data (SFD) insertion in Transmit, and deletion in Receive paths
- Automatic CRC and padding (all 0) generation controllable on a per-frame basis
- Options for automatic padding generation for data below the minimum frame length

- Programmable frame length supporting jumbo frames of up to 16 KB
- Programmable inter-frame gap (IFG) from 40 to 96 bit times in steps of 8
- Flexible address filtering modes:
  - Up to nine 48-bit perfect address filters with per-byte masking
  - Up to nine 48-bit source address (SA) comparisons with per-byte masking
  - Option to pass all multicast addressed frames
  - Promiscuous mode to pass all frames without filtering for network monitoring
  - Passes all incoming packets (as per filter) with a status report
- Separate 32-bit status returned for transmission and reception packets
- IEEE 802.1Q VLAN tag detection for reception frames
- Separate transmission, reception, and control interfaces for the application
- Management Data Input/Output (MDIO) interface for PHY device configuration and management
- Checksum offload for received IPv4 and TCP packets encapsulated by the Ethernet frame
- Checking IPv4 header checksum and TCP, UDP, or ICMP checksum encapsulated in IPv4 or IPv6 datagrams
- 64-bit timestamp for each transmitted and received frame (see IEEE 1588-2008)
- Energy Efficient Ethernet support (see IEEE 802.3az-2010)
- CRC replacement, SA insertion/replacement, and VLAN insertion/replacement/deletion in transmit frames
- Two FIFOs: 1024-byte TX FIFO and 256-byte RX FIFO
- Receive status vectors inserted into RX FIFO after the EOF (end of frame) transfer, allowing multiple-frame storage without requiring an additional FIFO for status
- Option to forward good runt frames
- Statistics generation with pulse signaling for dropped or corrupted frames due to RX FIFO overflow
- Automatic re-transmission of collision frames
- Frame discarding in cases of late collisions, excessive collisions, excessive deferrals, or underflow conditions
- Software control for TX FIFO flushing

Pin Assignment

EMAC RGMII interface corresponds to the following designated pins in IO MUX:

- The pins are multiplexed with GPIO8–GPIO19

EMAC RMII interface corresponds to the following designated pins in IO MUX:

- The pins are multiplexed with GPIO8, GPIO9, GPIO12, GPIO13, GPIO15, GPIO18, GPIO19

EMAC MII interface uses the available RGMII pins and additionally requires three arbitrary GPIOs for signals such as rxderr/csr/col.

MDIO and other signals can be routed to any GPIOs via the GPIO Matrix.

4.2.2.9 CAN FD Controller

The CAN FD is a multi-master multi-cast communication protocol. The CAN FD controller facilitates the communication based on this protocol. The CAN FD controllers integrated in ESP32-S31 are protocol-compatible with the CAN FD specification but have not been formally certified.

Feature List

- compliant with ISO11898-1:2015
- RX buffer FIFO with 128 words (6 CAN FD frames with 64-byte data payloads, 21 CAN/CAN FD frames with 8-byte data payloads)
- 4 TX buffers (1 CAN FD frame in each TX buffer)
- 32-bit APB interface
- support of ISO and non-ISO CAN FD protocol
- timestamping and time triggered transmission
- three single-bit filters and one range filter
- support interrupts
- loopback, bus monitoring, ACK forbidden, self test, and restricted operation modes

Pin Assignment

The pins for the CAN FD Controller can be chosen from any GPIOs via the GPIO Matrix.

4.2.2.10 SD/MMC Host Controller (SDHOST)

ESP32-S31 has an SD/MMC Host Controller.

Feature List

- Supports two external cards
- SD Memory Card Specifications v3.0 and v3.01
- Secure Digital I/O (SDIO) v3.0
- CE-ATA v1.1
- MMC v4.41 and eMMC v4.5, v4.51
- 1-bit and 4-bit bus width modes (8-bit mode not supported)

ESP32-S31 SD/SDIO/MMC Host Controller can simultaneously support two SD/SDIO/MMC 4.41 cards, and supports one SD card operating at 1.8 V.

Pin Assignment

For the SD/SDIO/MMC host controller, Card 1 can use GPIO20–GPIO25 via IO MUX, and Card 2 can use GPIO35–GPIO40 via IO MUX.

4.2.2.11 LED PWM Controller (LEDC)

The LED PWM Controller is a peripheral designed to generate PWM signals for LED control. It has specialized features such as automatic duty cycle fading. However, the LED PWM Controller can also be used to generate PWM signals for other purposes.

Feature List

- Two independent LED PWM controllers, each with eight independent PWM generator channels (16 channels in total)
- Maximum PWM duty cycle resolution: 20 bits
- Four independent timers per controller with 20-bit counters, fractional clock divider, and configurable overflow value
- Adjustable phase of PWM signal output
- PWM duty cycle dithering
- Automatic duty cycle fading —gradual increase/decrease of a PWM's duty cycle without interference from the processor. An interrupt will be generated upon fade completion
- Up to 16 duty cycle ranges for each PWM generator to generate gamma curve signals; each range can be independently configured for fade direction (increase or decrease), fade amount (per-step duty change), fade count (number of steps per range), and fade frequency
- PWM signal output in low-power mode (Light-sleep mode)
- Event generation and task response related to the Event Task Matrix (ETM) peripheral

Pin Assignment

The pins for the LED PWM controller can be chosen from any GPIOs via the GPIO Matrix.

4.2.2.12 Motor Control PWM (MCPWM)

ESP32-S31 integrates an MCPWM that can be used to drive digital motors and smart light.

Feature List

- A clock divider (prescaler), three PWM timers, three PWM operators, an Event Task Matrix (ETM) module, a Fault Detection module, and a dedicated capture module. PWM timers are used to generate timing references. PWM operators generate desired waveform based on the timing references
- A PWM operator can use the timing reference of any PWM timer
- A PWM operator can use the same timing reference with other PWM operators
- PWM operators can use different PWM timers' values to produce independent PWM signals

- PWM timers can be synchronized

Pin Assignment

The pins for the Motor Control PWM can be chosen from any GPIOs via the GPIO Matrix.

4.2.2.13 Remote Control Peripheral (RMT)

The Remote Control Peripheral (RMT) supports four channels of infrared remote transmission and four channels of infrared remote reception. By controlling pulse waveform through software, it supports various infrared and other single wire protocols.

Feature List

- Eight channels:
  - TX channels 0–3
  - RX channels 4–7
  - Eight channels share a 384 x 32-bit RAM
- The transmitter supports:
  - Normal TX mode
  - Wrap TX mode
  - Continuous TX mode
  - Modulation on TX pulses
  - Multiple channels transmitting data simultaneously (programmable)
  - GDMA access supported by TX channel 3
- The receiver supports:
  - Normal RX mode
  - Wrap RX mode
  - RX filtering
  - Demodulation on RX pulses
  - GDMA access supported by RX channel 7

Pin Assignment

The pins for the remote control peripheral can be chosen from any GPIOs via the GPIO Matrix.

4.2.2.14 Parallel IO Controller (PARLIO)

ESP32-S31 contains a Parallel IO controller (PARLIO) capable of transferring data between external devices and internal memory on a parallel bus through General Direct Memory Access (GDMA).

Feature List

- Various clock sources:
  - Including external IO clock PAD_CLK_TX/RX and internal system clock XTAL_CLK, PLL_F160M_CLK, and RC_FAST_CLK
  - Maximum IO clock frequency of 40 MHz
  - Integer and fractional clock frequency division
- 1/2/4/8/16-bit configurable data bus width
- Full-duplex communication with 16-bit data bus width
- Bit reversal when data bus width is 1/2/4-bit
- RX unit for receiving IO parallel data, which supports:
  - Output clock gating
  - RX unit input and output clock inverse
  - Various receive modes
  - Configurable GDMA SUC EOF generation
  - Configurable IO pin of external enable signal
- TX unit for sending IO parallel data, which supports:
  - Output clock gating
  - TX unit input and output clock inverse
  - Configurable TX EOF generation
  - Valid signal output
  - Configurable bus idle value

Pin Assignment

The pins for the parallel IO controller can be chosen from any GPIOs via the GPIO Matrix.

4.2.2.15 BitScrambler

The ESP32-S31 has an extensive amount of DMA-capable peripherals. These can move data from memory to an external device, and vice versa, without any interference from the CPU. This only works if the external device needs or emits the data in question in the same format as the software expects it: if not, the CPU needs to rewrite the format of the data. Examples include a need to swap bytes, reverse bytes, and shift the data left or right.

Since bitwise operations can be relatively CPU-intensive and DMA is designed specifically to offload such work from the CPU, ESP32-S31 integrates two dedicated peripherals called BitScramblers. These modules are designed to transform data formats during transfers between memory and peripherals. One BitScrambler handles memory-to-peripheral (or memory-to-memory) transfers, while the other is dedicated to peripheral-to-memory transfers. While BitScramblers can handle the bitwise operations mentioned earlier, they

are in fact flexible, programmable state machines capable of performing more advanced transformations as well.

Feature List

- Two BitScramblers, one for RX (peripheral-to-memory), one for TX (memory-to-peripheral)
- Support for memory-to-memory transfers
- Processing up to 32 bits per DMA clock period
- Data path controlled by a BitScrambler program stored in instruction memory
- Input registers able to read 0, 8, 16, or 32 bits per clock cycle
- Output registers:
  - Able to write 0, 8, 16, or 32 bits per clock cycle
  - Data sources for output register bits: 64 bits of input data, two counters, LUT RAM data, data output of last cycle, comparators
  - With some restrictions, each of the 32 output register bits can come from any bit on the data sources
- An 8 x 257-bit instruction memory for storing eight instructions, controlling control flow, and the data path
- 2048 bytes of lookup table (LUT) memory, configurable as various word widths

Pin Assignment

The BitScrambler does not directly interact with IOs, so it has no pins assigned.

4.2.3 Analog Signal Processing

This subsection describes components on the chip that sense and process real-world data.

4.2.3.1 Touch Sensor (TOUCH)

ESP32-S31 has 14 capacitive-sensing GPIOs, which detect variations induced by touching or approaching the GPIOs with a finger or other objects. The low-noise nature of the design and the high sensitivity of the circuit allow relatively small pads to be used. Arrays of pads can also be used, so that a larger area or more points can be detected. The touch sensing performance can be further enhanced by the waterproof design, detection of frequency hopping, and digital filtering feature.

Feature List

- Detection of 14 capacitive touch pins
- Sampling triggered by software or dedicated hardware timer
- Two sampling methods:
  - Pulses from the touch pins used as clock signals to count the sampling period
  - Pulses from the touch pins used as digital signals; sample the rising edge of the digital signal with the system clock to count the sampling period

- Scan mode, supporting sequential sampling of multiple touch pins by configuring the Touch FSM.
- Timeout mechanism to monitor channel abnormality
- Frequency hopping to increase the anti-interference of detection
- Proximity sensing mode with up to three configurable channels
- Configuration of individual touch sensors to operate normally in sleep mode
- Wake-up by touch sensor
- Moisture resistance
- Waterproof design

Pin Assignment

The touch sensor interface is multiplexed with GPIO6~GPIO19 pins. When the pins are configured for the analog function, the multiplexed digital functions are disabled.

4.2.3.2 Temperature Sensor (TSENS)

ESP32-S31 provides a temperature sensor for real-time monitoring of temperature changes within the chip. The sensor converts analog voltage to digital values and provides compensation for temperature offsets.

Feature List

- Software-triggered temperature measurement, which once triggered, the sensor continuously measures temperature. Software can read the data at any time.
- Hardware-triggered automatic temperature monitoring
- Two automatic monitoring modes with interrupt generation
- Configurable temperature offset based on the application scenario for improved accuracy
- Configurable temperature measurement range
- Support for Event Task Matrix (ETM)-related events and tasks

4.2.3.3 ADC Controller (ADC)

ESP32-S31 integrates two 12-bit successive approximation ADCs (SAR ADCs), each capable of measuring analog signals from up to eight pins.

Feature List

- 12-bit resolution
- Analog inputs sampling from up to eight pins per ADC (16 pins in total)
- One-shot sampling mode and multi-channel sampling mode
- In multi-channel sampling mode:
  - Configurable channel sampling sequence

  - Two filters with configurable filter coefficients
  - Two threshold monitors that can trigger an interrupt when the filtered value is below a low threshold or above a high threshold
  - Continuous transfer of converted data to memory via GDMA interface
- Support for several Event Task Matrix (ETM) related events and tasks

Pin Assignment

SAR ADC1 pins are multiplexed with GPIO42 ~ GPIO49; SAR ADC2 pins are multiplexed with GPIO50 ~ GPIO57.

4.2.3.4 DAC Controller (DAC)

ESP32-S31 provides a digital-to-analog converter that converts digital signals into analog voltages and outputs them on two dedicated chip pads. The two channels connect to two pads respectively and can output independently without affecting each other.

Feature List

- Supports output value configuration by software via PDMA, or sine-wave output via an internal lookup table
- Output voltage range: 0 V to 3.3 V
- Supports maintaining output level in ultra-low-power mode

Pin Assignment

DAC uses dedicated pads and is supported only on GPIO4 (DAC channel 0) and GPIO5 (DAC channel 1).

4.2.3.5 Analog Voltage Comparator

ESP32-S31 includes one analog voltage comparator with four dedicated pads. Three of these pads can be selected to compare their voltages against the remaining pad. Alternatively, any of the three pads can be compared against an internally adjustable stable reference voltage.

Feature List

- Reference voltage selectable between internal reference and external reference
- Internal reference voltage range: 0 ~ 0.7 * VDDPST
- Internal reference voltage supports hysteresis
- ETM support
- Interrupt output when the measured voltage crosses the reference voltage

Pin Assignment

The analog voltagecomparator uses dedicated pads and is supported only on GPIO37, GPIO38, GPIO39, and GPIO40. Any of these pads can be configured as the measured-input pad or the reference-input pad.

4.3 Wireless Communication

This section describes the chip's wireless communication capabilities, spanning radio technology, Wi-Fi, Bluetooth LE, Bluetooth Classic and 802.15.4.

4.3.1 Radio

This subsection describes the fundamental radio technology embedded in the chip that facilitates wireless communication and data exchange.

4.3.1.1 2.4 GHz Receiver

The 2.4 GHz receiver demodulates the 2.4 GHz RF signal to quadrature baseband signals and converts them to the digital domain with two high-resolution, high-speed ADCs. To adapt to varying signal channel conditions, ESP32-S31 integrates RF filters, Automatic Gain Control (AGC), DC offset cancelation circuits, and baseband filters.

4.3.1.2 2.4 GHz Transmitter

The 2.4 GHz transmitter modulates the quadrature baseband signals to the 2.4 GHz RF signal, and drives the antenna with a high-powered CMOS power amplifier. The use of digital calibration further improves the linearity of the power amplifier.

Additional calibrations are integrated to cancel any radio imperfections, such as:

- Carrier leakage
- I/Q amplitude/phase matching
- Baseband nonlinearities
- RF nonlinearities
- Antenna matching

These built-in calibration routines reduce the cost, time, and specialized equipment required for product testing.

4.3.1.3 Clock Generator

The clock generator produces quadrature clock signals of 2.4 GHz for both the receiver and the transmitter. All components of the clock generator are integrated into the chip, including inductors, varactors, filters, regulators and dividers.

The clock generator has built-in calibration and self-test circuits. Quadrature clock phases and phase noise are optimized on chip with patented calibration algorithms which ensure the best performance of the receiver and the transmitter.

4.3.2 Wi-Fi

This subsection describes the chip's Wi-Fi capabilities, which facilitate wireless communication at a high data rate.

4.3.2.1 Wi-Fi Radio and Baseband

The ESP32-S31 Wi-Fi radio and baseband support the following features:

- 1T1R in 2.4 GHz band
- 802.11ax
  - 20 MHz-only non-AP mode
  - MCS0 ~MCS9
  - Uplink and downlink OFDMA
  - Downlink MU-MIMO (multi-user, multiple input, multiple output)
  - Longer OFDM symbol, with 0.8, 1.6, 3.2 μs guard interval
  - DCM (dual carrier modulation), up to 16-QAM
  - Single-user/multi-user beamformee
  - Channel quality indication (CQI)
  - RX STBC (single spatial stream)
- 802.11b/g/n
  - MCS0 ~MCS7 that supports 20 MHz and 40 MHz bandwidth
  - MCS32
  - Data rate up to 150 Mbps
  - 0.4 μs guard interval
- Adjustable transmitting power
- Antenna diversity
  ESP32-S31 supports antenna diversity with an external RF switch. This switch is controlled by one or more GPIOs, and used to select the best antenna to minimize the effects of channel imperfections.

4.3.2.2 Wi-Fi MAC

ESP32-S31 implements the full IEEE 802.11 b/g/n/ax Wi-Fi MAC protocol. ESP32-S31 supports the Basic Service Set (BSS) STA and SoftAP operations under the Distributed Coordination Function (DCF). Power management is handled automatically with minimal host interaction to minimize the active duty period.

The ESP32-S31 Wi-Fi MAC applies the following low-level protocol functions automatically:

- Four virtual Wi-Fi interfaces
- Infrastructure BSS in Station mode, SoftAP mode, Station + SoftAP mode, and promiscuous mode
- RTS protection, CTS-to-Self protection, Immediate Block ACK
- Fragmentation and defragmentation
- TX/RX A-MPDU, TX/RX A-MSDU
- Transmit opportunity (TXOP)

- Wi-Fi multimedia (WMM)
- GCMP, CCMP, TKIP, WAPI, WEP, BIP, WPA2-PSK/WPA2-Enterprise, and WPA3-PSK/WPA3-Enterprise
- Automatic beacon monitoring (hardware TSF)
- 802.11mc FTM
- 802.11ax supports:
  - Target wake time (TWT) requester
  - Multiple BSSIDs
  - Triggered response scheduling
  - Multi-user Request-to-Send (MU-RTS), Multi-user Block ACK Request (MU-BAR), and Multi-STA Block ACK (M-BA) frame
  - Intra-PPDU power saving mechanism
  - Two network allocation vectors (NAV)
  - BSS coloring
  - Spatial reuse
  - Uplink power headroom
  - Operating mode control
  - Buffer status report
  - TXOP duration RTS threshold
  - UL-OFDMA random access (UORA)

4.3.2.3 Networking Features

Espressif provides libraries for TCP/IP networking, ESP-WIFI-MESH networking, and other networking protocols over Wi-Fi. TLS 1.0, 1.1 and 1.2 is also supported.

4.3.3 Bluetooth LE

This subsection describes the chip's Bluetooth LE capabilities for low-power data communication, supporting a wide range of use cases such as efficient data transfer, Bluetooth LE Audio, positioning, and electronic shelf labels.

4.3.3.1 Bluetooth LE PHY

Bluetooth Low Energy PHY in ESP32-S31 supports:

- 1 Mbps PHY
- 2 Mbps PHY for Higher Data Rates
- Coded PHY for Longer Range (125 Kbps and 500 Kbps)
- HW Listen Before Talk (LBT)

4.3.3.2 Bluetooth LE Link Controller

Bluetooth Low Energy Link Controller and Host in ESP32-S31 support:

- LE Audio (Isochronous Channels, BIS and CIS)
- Direction Finding (AoA/AoD)
- Periodic Advertising with Responses (PAwR)
- LE Connection Subrating
- LE Advertising Extensions and Multiple Advertising Sets
- Simultaneous Operation of Broadcaster, Observer, Central, and Peripheral Devices
- Adaptive Frequency Hopping and Channel Assessment
- LE Channel Selection Algorithm #2
- LE Power Control
- Advertising Coding Selection
- Encrypted Advertising Data
- LE GATT Security Levels Characteristic
- AdvDataInfo in Periodic Advertising
- LE Channel Classification
- Enhanced Attribute Protocol
- Advertising Channel Index
- GATT Caching
- Periodic Advertising Sync Transfer
- High Duty Cycle Non-Connectable Advertising
- LE Data Packet Length Extension
- LE Secure Connections
- LE Privacy 1.2
- Link Layer Extended Scanner Filter Policies
- Low Duty Cycle Directed Advertising
- Link Layer Encryption
- LE Ping

4.3.4 Bluetooth Classic

This subsection describes the chip's Bluetooth Classic capabilities for sustained data throughput and traditional Bluetooth audio (e.g., A2DP, HFP), suitable for established audio ecosystems and continuous data links.

4.3.4.1 Bluetooth Classic PHY

ESP32-S31 series chips Bluetooth Classic physical layer supports the following features:

- Basic rate (GFSK): 1 Mbps
- Enhanced data rate (π/4 DQPSK): 2 Mbps
- Enhanced data rate (8DPSK): 3 Mbps
- Hardware-implemented clear channel assessment (CCA)
- Maximum output power meets Power Class 1 device requirements

4.3.4.2 Bluetooth Classic Link Controller

ESP32-S31 series chips Bluetooth Classic link controller can perform device inquiry and paging operations, can form piconets and scatternets, and supports active mode and sniff mode in connection state, as well as role switching, data encryption, power control, adaptive frequency hopping and other operations. The main features of the Bluetooth Classic link controller are as follows:

- Device discovery (inquiry and inquiry scan)
- Connection establishment (paging and paging scan)
- Asynchronous connection-oriented (ACL) and synchronous connection-oriented (SCO/eSCO)
- Voice coding supports A-law, μ-law, CVSD and transparent data
- Voice data channel supports HCI or PCM/I2S interface
- Secure simple pairing (SSP)
- E0 encryption and AES-CCM encryption
- Secure Connections
- Channel classification and adaptive frequency hopping (AFH)
- Sniff mode and Sniff Subrating
- Role switching
- Traditional power control and enhanced power control
- Ping
- Piconet and scatternet management
- Active Peripheral Broadcast

4.3.5 802.15.4

This subsection describes the chip's compatibility with the 802.15.4 standard, which facilitates wireless communication for low-power, short-range applications.

4.3.5.1 802.15.4 PHY

ESP32-S31's 802.15.4 PHY supports:

- O-QPSK PHY in 2.4 GHz
- 250 Kbps data rate
- RSSI and LQI

4.3.5.2 802.15.4 MAC

ESP32-S31 supports most key features defined in IEEE Standard 802.15.4-2015, including:

- CSMA/CA
- Active scan and energy detect
- HW frame filter
- HW auto acknowledge
- HW auto frame pending
- Coordinated sampled listening (CSL)

5 Electrical Characteristics

Note:
The values presented in this section are **preliminary** and may change with the final release of this datasheet.

5.1 Absolute Maximum Ratings

Stresses above those listed in Table 5-1 *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only and normal operation of the device at these or any other conditions beyond those indicated in Section 5.2 *Recommended Power Supply Characteristics* is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

Table 5-1. Absolute Maximum Ratings

Parameter	Description	Min	Max	Unit
Input power pins [1]	Allowed input voltage	–0.3	3.6	V
I_{output} [2]	Cumulative IO output current	—	1500	mA
T_{STORE}	Storage temperature	–40	150	°C

[1] For more information on input power pins, see Section 2.5.1 *Power Pins*.

[2] The product proved to be fully functional after all its IO pins were pulled high while being connected to ground for 24 consecutive hours at ambient temperature of 25 °C.

5.2 Recommended Power Supply Characteristics

For recommended ambient temperature, see Section 1 *ESP32-S31 Series Information*.

Table 5-2. Recommended Power Characteristics

Parameter [1]	Description	Min	Typ	Max	Unit
VDDA1, VDDA2, VDDA3, VDDA4	Recommended input voltage	3.0	3.3	3.6	V
VDDPST_1, VDDPST_2, VDDPST_3, VDDPST_4 [2]	Recommended input voltage	3.0	3.3	3.6	V
I_{VDD}	Cumulative input current	0.6	—	—	A

[1] See in conjunction with Section 2.5 *Power Supply*.

[2] When writing to eFuses, the voltage should not exceed 3.3 V because eFuse programming circuitry is sensitive to higher voltages.

5.3 DC Characteristics (3.3 V, 25 °C)

Table 5-3. DC Characteristics (3.3 V, 25 °C)

Parameter	Description	Min	Typ	Max	Unit
C_{IN}	Pin capacitance	—	2	—	pF
V_{IH}	High-level input voltage	0.75 × VDD [1]	—	VDD [1] + 0.3	V
V_{IL}	Low-level input voltage	–0.3	—	0.25 × VDD [1]	V
I_{IH}	High-level input current	—	—	50	nA
I_{IL}	Low-level input current	—	—	50	nA
V_{OH} [2]	High-level output voltage	0.8 × VDD [1]	—	—	V
V_{OL} [2]	Low-level output voltage	—	—	0.1 × VDD [1]	V
I_{OH}	High-level source current (VDD [1] = 3.3 V, V_{OH} >= 2.64 V, PAD_DRIVER = 3)	—	40	—	mA
I_{OL}	Low-level sink current (VDD [1] = 3.3 V, V_{OL} = 0.495 V, PAD_DRIVER = 3)	—	28	—	mA
R_{PU}	Internal weak pull-up resistor	—	45	—	kΩ
R_{PD}	Internal weak pull-down resistor	—	45	—	kΩ
V_{IH_nRST}	Chip reset release voltage (CHIP_PU voltage is within the specified range)	0.75 × VDD [1]	—	VDD [1] + 0.3	V
V_{IL_nRST}	Chip reset voltage (CHIP_PU voltage is within the specified range)	–0.3	—	0.15 × VDD [1]	V

[1] VDD – voltage from a power pin of a respective power domain.

[2] V_{OH} and V_{OL} are measured using high-impedance load.

5.4 Current Consumption

5.4.1 Current Consumption in Active Mode

The current consumption measurements are taken with a 3.3 V supply at 25 °C ambient temperature.

TX current consumption is rated at a 100% duty cycle.

RX current consumption is rated when the peripherals are disabled and the CPU idle.

Table 5-4. Current Consumption for Wi-Fi (2.4 GHz) in Active Mode

Work Mode	RF Condition	Description	Peak (mA)
Active (RF working)	TX	802.11b, 1 Mbps, DSSS @ 20.5 dBm	348
		802.11g, 54 Mbps, OFDM @ 18 dBm	285
		802.11n, HT20, MCS7 @ 17 dBm	271
		802.11n, HT40, MCS7 @ 17 dBm	286
		802.11ax, MCS9, @ 14 dBm	249
	RX	802.11b/g/n, HT20	110
		802.11n, HT40	117
		802.11ax, HE20	111

Table 5-5. Current Consumption for Bluetooth LE in Active Mode

Work Mode	RF Condition	Description	Peak (mA)
Active (RF working)	TX	Bluetooth LE @ 20.5 dBm	347
		Bluetooth LE @ 11.5 dBm	219
		Bluetooth LE @ 0 dBm	142
		Bluetooth LE @ –15.5 dBm	117
	RX	Bluetooth LE	102

Table 5-6. Current Consumption for Bluetooth Classic in Active Mode

Work Mode	RF Condition	Description	Peak (mA)
Active (RF working)	TX	Bluetooth Classic @ 12 dBm	220
		Bluetooth Classic @ 6 dBm	179
		Bluetooth Classic @ 0 dBm	143
		Bluetooth Classic @ –15 dBm	116
	RX	Bluetooth Classic	96

Table 5-7. Current Consumption for 802.15.4 in Active Mode

Work Mode	RF Condition	Description	Peak (mA)
Active (RF working)	TX	802.15.4 @ 20.5 dBm	342
		802.15.4 @ 12 dBm	220
		802.15.4 @ 0 dBm	151
		802.15.4 @ –15 dBm	115
	RX	802.15.4	101

6 RF Characteristics

This section contains tables with RF characteristics of the Espressif product.

The RF data is measured at the antenna port, where RF cable is connected, including the front-end loss. The front-end circuit is a 0 Ω resistor.

Devices should operate in the center frequency range allocated by regional regulatory authorities. The target center frequency range and the target transmit power are configurable by software. See ESP RF Test Tool and Test Guide for instructions.

Unless otherwise stated, the RF tests are conducted with a 3.3 V (±5%) supply at 25 °C ambient temperature.

6.1 Wi-Fi Radio

Table 6-1. 2.4 GHz Wi-Fi RF Characteristics

Name	Description
Center frequency range of operating channel	2412 ~ 2484 MHz
Wi-Fi wireless standard	IEEE 802.11b/g/n/ax

6.1.1 Wi-Fi RF Transmitter (TX) Characteristics

Table 6-2. 2.4 GHz TX Power with Spectral Mask and EVM Meeting 802.11 Standards

Rate	Min (dBm)	Typ (dBm)	Max (dBm)
802.11b, 1 Mbps, DSSS	—	20.5	—
802.11b, 11 Mbps, CCK	—	20.5	—
802.11g, 6 Mbps, OFDM	—	19.5	—
802.11g, 54 Mbps, OFDM	—	17.5	—
802.11n, HT20, MCS0	—	19.5	—
802.11n, HT20, MCS7	—	16.5	—
802.11n, HT40, MCS0	—	18.0	—
802.11n, HT40, MCS7	—	16.0	—
802.11ax, HE20, MCS0	—	19.5	—
802.11ax, HE20, MCS9	—	14.5	—

Table 6-3. 2.4 GHz TX EVM Test[1]

Rate	Min (dB)	Typ (dB)	Limit (dB)
802.11b, 1 Mbps, DSSS	—	–25.0	–10.0
802.11b, 11 Mbps, CCK	—	–25.0	–10.0

Cont'd on next page

Table 6-3 – cont'd from previous page

Rate	Min (dB)	Typ (dB)	Limit (dB)
802.11g, 6 Mbps, OFDM	—	–24.0	–5.0
802.11g, 54 Mbps, OFDM	—	–31.0	–25.0
802.11n, HT20, MCS0	—	–24.0	–5.0
802.11n, HT20, MCS7	—	–33.0	–27.0
802.11n, HT40, MCS0	—	–25.0	–5.0
802.11n, HT40, MCS7	—	–32.0	–27.0
802.11ax, HE20, MCS0	—	–25.0	–5.0
802.11ax, HE20, MCS9	—	–34.0	–32.0

[1] EVM is measured at the corresponding typical TX power provided in Table 6-2 *2.4 GHz TX Power with Spectral Mask and EVM Meeting 802.11 Standards* above.

6.1.2 Wi-Fi RF Receiver (RX) Characteristics

For RX tests, the PER (packet error rate) limit is 8% for 802.11b, and 10% for 802.11g/n/ax.

Table 6-4. 2.4 GHz RX Sensitivity

Rate	Min (dBm)	Typ (dBm)	Max (dBm)
802.11b, 1 Mbps, DSSS	—	–99.5	—
802.11b, 2 Mbps, DSSS	—	–96.5	—
802.11b, 5.5 Mbps, CCK	—	–93.5	—
802.11b, 11 Mbps, CCK	—	–89.5	—
802.11g, 6 Mbps, OFDM	—	–94.5	—
802.11g, 9 Mbps, OFDM	—	–92.5	—
802.11g, 12 Mbps, OFDM	—	–92.5	—
802.11g, 18 Mbps, OFDM	—	–89.5	—
802.11g, 24 Mbps, OFDM	—	–87.0	—
802.11g, 36 Mbps, OFDM	—	–83.0	—
802.11g, 48 Mbps, OFDM	—	–79.0	—
802.11g, 54 Mbps, OFDM	—	–77.5	—
802.11n, HT20, MCS0	—	–94.5	—
802.11n, HT20, MCS1	—	–92.0	—
802.11n, HT20, MCS2	—	–89.5	—
802.11n, HT20, MCS3	—	–86.0	—
802.11n, HT20, MCS4	—	–83.0	—
802.11n, HT20, MCS5	—	–78.5	—
802.11n, HT20, MCS6	—	–77.0	—
802.11n, HT20, MCS7	—	–75.0	—
802.11n, HT40, MCS0	—	–92.0	—
802.11n, HT40, MCS1	—	–89.5	—

Cont'd on next page

Table 6-4 – cont'd from previous page

Rate	Min (dBm)	Typ (dBm)	Max (dBm)
802.11n, HT40, MCS2	—	–87.0	—
802.11n, HT40, MCS3	—	–83.5	—
802.11n, HT40, MCS4	—	–80.5	—
802.11n, HT40, MCS5	—	–76.0	—
802.11n, HT40, MCS6	—	–74.5	—
802.11n, HT40, MCS7	—	–73.5	—
802.11ax, HE20, MCS0	—	–94.0	—
802.11ax, HE20, MCS1	—	–91.0	—
802.11ax, HE20, MCS2	—	–88.0	—
802.11ax, HE20, MCS3	—	–85.5	—
802.11ax, HE20, MCS4	—	–82.0	—
802.11ax, HE20, MCS5	—	–78.0	—
802.11ax, HE20, MCS6	—	–76.5	—
802.11ax, HE20, MCS7	—	–74.5	—
802.11ax, HE20, MCS8	—	–71.0	—
802.11ax, HE20, MCS9	—	–68.5	—

Table 6-5. 2.4 GHz Maximum RX Level

Rate	Min (dBm)	Typ (dBm)	Max (dBm)
802.11b, 1 Mbps, DSSS	—	5	—
802.11b, 11 Mbps, CCK	—	5	—
802.11g, 6 Mbps, OFDM	—	5	—
802.11g, 54 Mbps, OFDM	—	0	—
802.11n, HT20, MCS0	—	5	—
802.11n, HT20, MCS7	—	0	—
802.11n, HT40, MCS0	—	5	—
802.11n, HT40, MCS7	—	0	—
802.11ax, HE20, MCS0	—	5	—
802.11ax, HE20, MCS9	—	0	—

6.2 Bluetooth LE Radio

Table 6-6. Bluetooth LE RF Characteristics

Name	Description
Center frequency range of operating channel	2402 ~ 2480 MHz
RF transmit power range	–14.5 ~ 20.5 dBm

6.2.1 Bluetooth LE RF Transmitter (TX) Characteristics

Table 6-7. Bluetooth LE - Transmitter Characteristics - 1 Mbps

Parameter	Description	Min	Typ	Max	Unit
Carrier frequency offset and drift	Max. $\lvert f_n \rvert_{n=0,\,1,\,2,\,3,\,\ldots k}$	—	0.5	—	kHz
	Max. $\lvert f_0 - f_n \rvert_{n=2,\,3,\,4,\,\ldots k}$	—	1.0	—	kHz
	Max. $\lvert f_n - f_{n-5} \rvert_{n=6,\,7,\,8,\,\ldots k}$	—	1.0	—	kHz
	$\lvert f_1 - f_0 \rvert$	—	1.4	—	kHz
Modulation characteristics	$\Delta\, F1_{\text{avg}}$	—	256.3	—	kHz
	Min. $\Delta\, F2_{\text{max}}$ (for at least 99.9% of all $\Delta\, F2_{\text{max}}$)	—	252.6	—	kHz
	$\Delta\, F2_{\text{avg}} / \Delta\, F1_{\text{avg}}$	—	0.95	—	—
In-band emissions	± 2 MHz offset	—	–29	—	dBm
	± 3 MHz offset	—	–38	—	dBm
	> ± 3 MHz offset	—	–44	—	dBm

Table 6-8. Bluetooth LE - Transmitter Characteristics - 2 Mbps

Parameter	Description	Min	Typ	Max	Unit
Carrier frequency offset and drift	Max. $\lvert f_n \rvert_{n=0,\,1,\,2,\,3,\,\ldots k}$	—	0.5	—	kHz
	Max. $\lvert f_0 - f_n \rvert_{n=2,\,3,\,4,\,\ldots k}$	—	1.3	—	kHz
	Max. $\lvert f_n - f_{n-5} \rvert_{n=6,\,7,\,8,\,\ldots k}$	—	0.9	—	kHz
	$\lvert f_1 - f_0 \rvert$	—	0.5	—	kHz
Modulation characteristics	$\Delta\, F1_{\text{avg}}$	—	508.2	—	kHz
	Min. $\Delta\, F2_{\text{max}}$ (for at least 99.9% of all $\Delta\, F2_{\text{max}}$)	—	516.9	—	kHz
	$\Delta\, F2_{\text{avg}} / \Delta\, F1_{\text{avg}}$	—	0.97	—	—
In-band emissions	± 4 MHz offset	—	–43	—	dBm
	± 5 MHz offset	—	–45	—	dBm
	> ± 5 MHz offset	—		—	dBm

Table 6-9. Bluetooth LE - Transmitter Characteristics - 125 Kbps

Parameter	Description	Min	Typ	Max	Unit
Carrier frequency offset and drift	Max. $\lvert f_n \rvert_{n=0,\,1,\,2,\,3,\,\ldots k}$	—	1.3	—	kHz
	Max. $\lvert f_0 - f_n \rvert_{n=1,\,2,\,3,\,\ldots k}$	—	0.5	—	kHz
	$\lvert f_0 - f_3 \rvert$	—	0.1	—	kHz
	Max. $\lvert f_n - f_{n-3} \rvert_{n=7,\,8,\,9,\,\ldots k}$	—	0.6	—	kHz
Modulation characteristics	$\Delta\, F1_{\text{avg}}$	—	255.7	—	kHz
	Min. $\Delta\, F1_{\text{max}}$ (for at least 99.9% of all $\Delta\, F1_{\text{max}}$)	—	261.9	—	kHz
In-band emissions	± 2 MHz offset	—	–26	—	dBm
	± 3 MHz offset	—	–36	—	dBm
	> ± 3 MHz offset	—	–40	—	dBm

Table 6-10. Bluetooth LE - Transmitter Characteristics - 500 Kbps

Parameter	Description	Min	Typ	Max	Unit
Carrier frequency offset and drift	Max. $\lvert f_n \rvert_{n=0,\,1,\,2,\,3,\,\ldots k}$	—	0.7	—	kHz
	Max. $\lvert f_0 - f_n \rvert_{n=1,\,2,\,3,\,\ldots k}$	—	0.5	—	kHz
	$\lvert f_0 - f_3 \rvert$	—	0.1	—	kHz
	Max. $\lvert f_n - f_{n-3} \rvert_{n=7,\,8,\,9,\,\ldots k}$	—	0.6	—	kHz
Modulation characteristics	$\Delta\, F2_{\text{avg}}$	—	245.7	—	kHz
	Min. $\Delta\, F2_{\text{max}}$ (for at least 99.9% of all $\Delta\, F2_{\text{max}}$)	—	252.9	—	kHz
In-band emissions	± 2 MHz offset	—	–30	—	dBm
	± 3 MHz offset	—	–40	—	dBm
	> ± 3 MHz offset	—	–45	—	dBm

6.2.2 Bluetooth LE RF Receiver (RX) Characteristics

Table 6-11. Bluetooth LE - Receiver Characteristics - 1 Mbps

Parameter	Description	Min	Typ	Max	Unit
Sensitivity @30.8% PER	—	—	–98.0	—	dBm
Maximum received signal @30.8% PER	—	—	8	—	dBm

Table 6-12. Bluetooth LE - Receiver Characteristics - 2 Mbps

Parameter	Description	Min	Typ	Max	Unit
Sensitivity @30.8% PER	—	—	–94.5	—	dBm
Maximum received signal @30.8% PER	—	—	8	—	dBm

Table 6-13. Bluetooth LE - Receiver Characteristics - 125 Kbps

Parameter	Description	Min	Typ	Max	Unit
Sensitivity @30.8% PER	—	—	–105.0	—	dBm
Maximum received signal @30.8% PER	—	—	8	—	dBm

Table 6-14. Bluetooth LE - Receiver Characteristics - 500 Kbps

Parameter	Description	Min	Typ	Max	Unit
Sensitivity @30.8% PER	—	—	–101.0	—	dBm
Maximum received signal @30.8% PER	—	—	8	—	dBm

6.3 Bluetooth Classic Radio

Table 6-15. Bluetooth Classic RF Characteristics

Name	Description
Center frequency range of operating channel	2402 ~ 2480 MHz
RF transmit power range	–14.5 ~ 12.5 dBm

6.3.1 Bluetooth Classic RF Transmitter (TX) Characteristics

Table 6-16. Bluetooth Classic - Transmitter Characteristics - Basic Rate (BR)

Parameter	Description	Min	Typ	Max	Unit
20 dB bandwidth	—	—	0.8	—	MHz
Adjacent channel transmit power	F = F0 ± 2 MHz	—	−31.0	—	dBm
	F = F0 ± 3 MHz	—	−44.0	—	dBm
	F = F0 ± > 3 MHz	—	−48.0	—	dBm
Modulation characteristics	$\Delta\ F1_{avg}$	—	155.0	—	kHz
	Min. $\Delta\ F2_{max}$ (for at least 99.9% of all $\Delta\ F2_{max}$)	—	160.0	—	kHz
	$\Delta\ F2_{avg}/\Delta\ F1_{avg}$	—	0.96	—	—
ICFT	—	—	1.0	—	kHz
Drift rate	—	—	0.5	—	kHz/50 μs
Drift (DH1)	—	—	0.5	—	kHz
Drift (DH5)	—	—	0.5	—	kHz

Table 6-17. Bluetooth Classic - Transmitter Characteristics - Enhanced Data Rate (EDR)

Parameter	Description	Min	Typ	Max	Unit
$\pi/4$ DQPSK max w_0	—	—	−2.0	—	kHz
$\pi/4$ DQPSK max w_i	—	—	−15.0	—	kHz
$\pi/4$ DQPSK max $\|w_i + w_0\|$	—	—	−17.0	—	kHz
8DPSK max w_0	—	—	−4.0	—	kHz
8DPSK max w_i	—	—	−17.0	—	kHz
8DPSK max $\|w_i + w_0\|$	—	—	−21.0	—	kHz
$\pi/4$ DQPSK modulation accuracy	RMS DEVM	—	7.0	—	%
	99% DEVM	—	0.1	—	%
	Peak DEVM	—	11.0	—	%
8DPSK modulation accuracy	RMS DEVM	—	7.0	—	%
	99% DEVM	—	0.1	—	%
	Peak DEVM	—	12.0	—	%
In-band emissions	F = F0 ± 1 MHz	—	−35	—	dBm
	F = F0 ± 2 MHz	—	−25	—	dBm
	F = F0 ± 3 MHz	—	−39	—	dBm
	F = F0 ± > 3 MHz	—	−43	—	dBm

6.3.2 Bluetooth Classic RF Receiver (RX) Characteristics

Table 6-18. Bluetooth Classic - Receiver Characteristics - Basic Rate (BR)

Parameter	Description	Min	Typ	Max	Unit
Sensitivity with dirty transmit off @0.1% BER	—	—	−95.0	—	dBm
Maximum received signal @0.1% BER	—	—	8	—	dBm

Table 6-19. Bluetooth Classic - Receiver Characteristics - Enhanced Data Rate (EDR)

Parameter	Description	Min	Typ	Max	Unit
$\pi/4$ DQPSK					
Sensitivity with dirty transmit off @0.01% BER	—	—	−94.0	—	dBm
Maximum received signal @0.01% BER	—	—	8	—	dBm
8DPSK					
Sensitivity with dirty transmit off @0.01% BER	—	—	−87.0	—	dBm
Maximum received signal @0.01% BER	—	—	2	—	dBm

6.4 802.15.4 Radio

Table 6-20. 802.15.4 RF Characteristics

Name	Description
Center frequency range of operating channel	2405 ~ 2480 MHz

[1] Zigbee in the 2.4 GHz range supports 16 channels at 5 MHz spacing from channel 11 to channel 26.

6.4.1 802.15.4 RF Transmitter (TX) Characteristics

Table 6-21. 802.15.4 Transmitter Characteristics - 250 Kbps

Parameter	Min	Typ	Max	Unit
RF transmit power range	–14.5	—	20.5	dBm
EVM	—	24.0%	—	—

6.4.2 802.15.4 RF Receiver (RX) Characteristics

Table 6-22. 802.15.4 Receiver Characteristics - 250 Kbps

Parameter	Description	Min	Typ	Max	Unit
Sensitivity @1% PER	—	—	–102.0	—	dBm

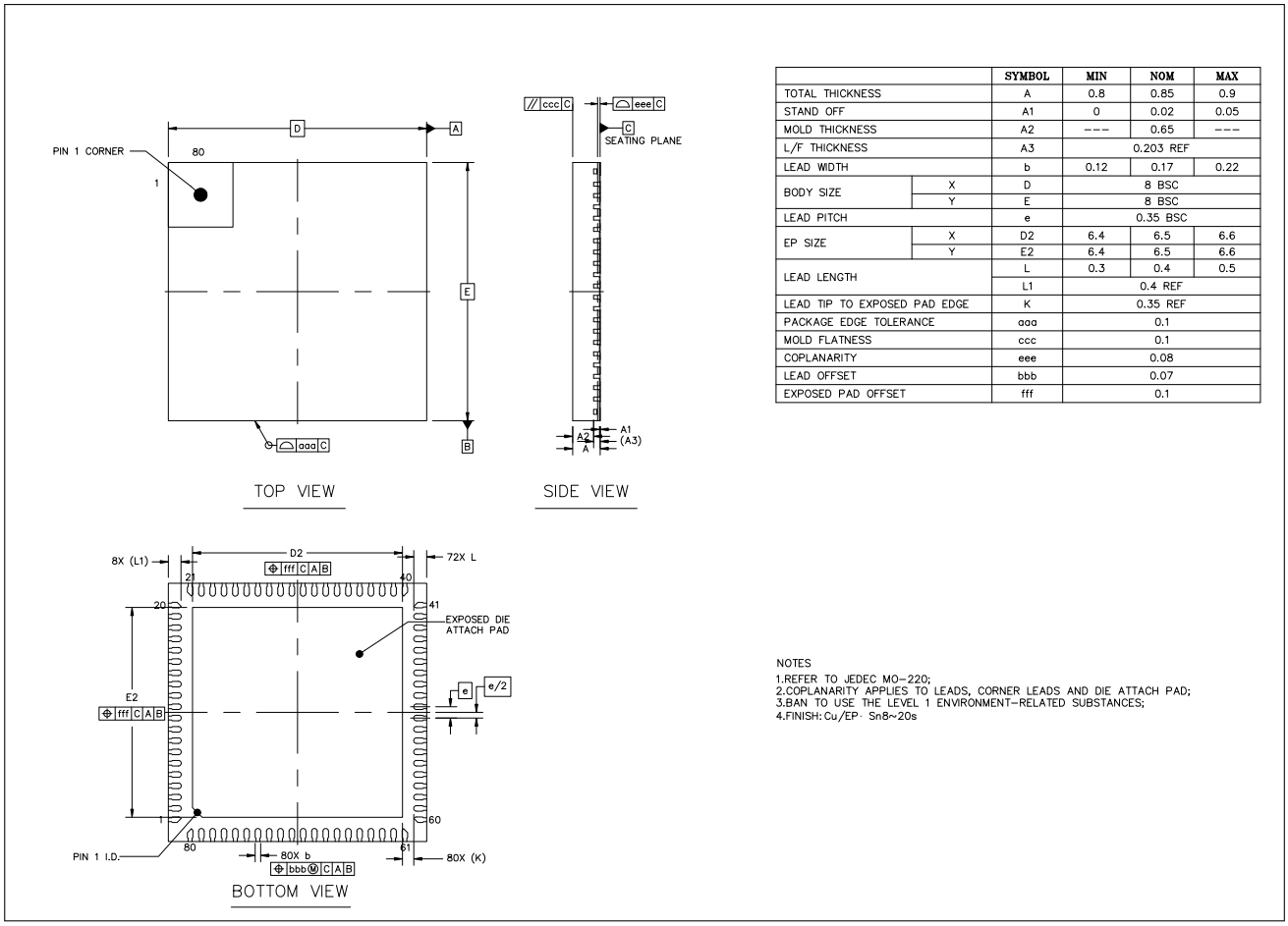
Cont'd on next page

Table 6-22 – cont'd from previous page

Parameter	Description	Min	Typ	Max	Unit
Maximum received signal @1% PER	—	—	8	—	dBm

7 Packaging

- The pins of the chip are numbered in anti-clockwise order starting from Pin 1 in the top view. For pin numbers and pin names, see also Figure 2-1 *ESP32-S31 Pin Layout (Top View)*.



		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	0.8	0.85	0.9
STAND OFF		A1	0	0.02	0.05
MOLD THICKNESS		A2	---	0.65	---
L/F THICKNESS		A3		0.203 REF	
LEAD WIDTH		b	0.12	0.17	0.22
BODY SIZE	X	D		8 BSC	
	Y	E		8 BSC	
LEAD PITCH		e		0.35 BSC	
EP SIZE	X	D2	6.4	6.5	6.6
	Y	E2	6.4	6.5	6.6
LEAD LENGTH		L	0.3	0.4	0.5
		L1		0.4 REF	
LEAD TIP TO EXPOSED PAD EDGE		K		0.35 REF	
PACKAGE EDGE TOLERANCE		aaa		0.1	
MOLD FLATNESS		ccc		0.1	
COPLANARITY		eee		0.08	
LEAD OFFSET		bbb		0.07	
EXPOSED PAD OFFSET		fff		0.1	

NOTES
1.REFER TO JEDEC MO-220;
2.COPLANARITY APPLIES TO LEADS, CORNER LEADS AND DIE ATTACH PAD;
3.BAN TO USE THE LEVEL 1 ENVIRONMENT-RELATED SUBSTANCES;
4.FINISH: Cu/EP· Sn8~20s

Figure 7-1. QFN80 (8×8 mm) Package

ESP32-S31 Consolidated Pin Overview

Table 7-1. Consolidated Pin Overview

Pin No.	Pin Name	Pin Type	Pin Providing Power	Pin Settings		Analog Functions	LP IO MUX Functions								IO MUX Functions									
				At Reset	After Reset	F0	F0	Type	F1	Type	F2	Type	F3	Type	F0	Type	F1	Type	F2	Type	F3	Type	F4	Type
1	ANT	Analog																						
2	VDDA3	Power																						
3	VDDA4	Power																						
4	CHIP_PU	Analog																						
5	XTAL_32K_N/GPIO0	IO	VDDPST_1				LP_GPIO0	I/O/T	LP_GPIO0	I/O/T			LP_PROBE_TOP_OUT0	O	**GPIO0**	I/O/T	GPIO0	I/O/T						
6	XTAL_32K_P/GPIO1	IO	VDDPST_1				LP_GPIO1	I/O/T	LP_GPIO1	I/O/T			LP_PROBE_TOP_OUT1	O	**GPIO1**	I/O/T	GPIO1	I/O/T						
7	GPIO2	IO	VDDPST_1				LP_UART_DTRN_PAD	O	LP_GPIO2	I/O/T	LP_SPI_CK_PAD	I1/O/T	LP_PROBE_TOP_OUT2	O	**GPIO2**	I/O/T	GPIO2	I/O/T			lcd_data19_out_pad	O		
8	GPIO3	IO	VDDPST_1				LP_UART_DSRN_PAD	I1	LP_GPIO3	I/O/T	LP_SPI_CS_PAD	I1/O/T	LP_PROBE_TOP_OUT3	O	**GPIO3**	I/O/T	GPIO3	I/O/T			lcd_data20_out_pad	O		
9	GPIO4	IO	VDDPST_1				LP_UART_RTSN_PAD	O	LP_GPIO4	I/O/T	LP_SPI_D_PAD	I1/O/T	LP_PROBE_TOP_OUT4	O	**GPIO4**	I/O/T	GPIO4	I/O/T			lcd_data21_out_pad	O		
10	GPIO5	IO	VDDPST_1				LP_UART_CTSN_PAD	I1	LP_GPIO5	I/O/T	LP_SPI_Q_PAD	I1/O/T	LP_PROBE_TOP_OUT5	O	**GPIO5**	I/O/T	GPIO5	I/O/T			lcd_data22_out_pad	O		
11	VDDPST_1	Power																						
12	GPIO6	IO	VDDPST_1			TOUCH_CHANNEL0	LP_UART_TXD_PAD	O	LP_GPIO6	I/O/T	LP_I2C_SCL_PAD	I1/O/T	LP_PROBE_TOP_OUT6	O	**GPIO6**	I/O/T	GPIO6	I/O/T						
13	GPIO7	IO	VDDPST_1			TOUCH_CHANNEL1	LP_UART_RXD_PAD	I1	LP_GPIO7	I/O/T	LP_I2C_SDA_PAD	I1/O/T	LP_PROBE_TOP_OUT7	O	**GPIO7**	I/O/T	GPIO7	I/O/T			lcd_data23_out_pad	O		
14	GPIO8	IO	VDDPST_1		IE	TOUCH_CHANNEL2									**GPIO8**	I/O/T	GPIO8	I/O/T	gmac_phy_txd0_pad	O	lcd_data0_out_pad	O		
15	GPIO9	IO	VDDPST_1		IE	TOUCH_CHANNEL3									**spi2_hold_pad**	I1/O/T	GPIO9	I/O/T	gmac_phy_txd1_pad	O	lcd_data1_out_pad	O	dbg_psram_ck_pad	O
16	GPIO10	IO	VDDPST_1		IE	TOUCH_CHANNEL4									**spi2_cs_pad**	I1/O/T	GPIO10	I/O/T	gmac_phy_txd2_pad	O	lcd_data2_out_pad	O	dbg_psram_cs_pad	O
17	GPIO11	IO	VDDPST_1		IE	TOUCH_CHANNEL5									**spi2_d_pad**	I1/O/T	GPIO11	I/O/T	gmac_phy_txd3_pad	O	lcd_data3_out_pad	O	dbg_psram_d_pad	O
18	VREF_TOUCH	Analog																						
19	GPIO12	IO	VDDPST_1		IE	TOUCH_CHANNEL6									**spi2_ck_pad**	I1/O/T	GPIO12	I/O/T	gmac_phy_txen_pad	O	lcd_data4_out_pad	O	dbg_psram_q_pad	O
20	GPIO13	IO	VDDPST_1		IE	TOUCH_CHANNEL7									**spi2_q_pad**	I1/O/T	GPIO13	I/O/T	gmac_rmii_clk_pad	IO/O/T	lcd_data5_out_pad	O	dbg_psram_wp_pad	O
21	GPIO14	IO	VDDPST_1		IE	TOUCH_CHANNEL8									**spi2_wp_pad**	I1/O/T	GPIO14	I/O/T	gmac_rx_clk_pad	IO	lcd_data6_out_pad	O	dbg_psram_hold_pad	O
22	GPIO15	IO	VDDPST_1		IE	TOUCH_CHANNEL9									**spi2_io4_pad**	I1/O/T	GPIO15	I/O/T	gmac_phy_rxdv_pad	IO	lcd_data7_out_pad	O	dbg_psram_dq4_pad	O
23	GPIO16	IO	VDDPST_1		IE	TOUCH_CHANNEL10									**spi2_io5_pad**	I1/O/T	GPIO16	I/O/T	gmac_phy_rxd3_pad	IO	lcd_data8_out_pad	O	dbg_psram_dq5_pad	O
24	GPIO17	IO	VDDPST_1		IE	TOUCH_CHANNEL11									**spi2_io6_pad**	I1/O/T	GPIO17	I/O/T	gmac_phy_rxd2_pad	IO	lcd_data9_out_pad	O	dbg_psram_dq6_pad	O
25	GPIO18	IO	VDDPST_1		IE	TOUCH_CHANNEL12									**spi2_io7_pad**	I1/O/T	GPIO18	I/O/T	gmac_phy_rxd1_pad	IO	lcd_data10_out_pad	O	dbg_psram_dq7_pad	O
26	GPIO19	IO	VDDPST_1		IE	TOUCH_CHANNEL13									**spi2_dqs_pad**	O/T	GPIO19	I/O/T	gmac_phy_rxd0_pad	IO	lcd_data11_out_pad	O	dbg_psram_dqs_0_pad	O
27	SDIO_DATA0	IO	VDDPST_SD		IE										**GPIO20**	I/O/T	GPIO20	I/O/T	spi2_ck_pad	I1/O/T			dbg_flash_ck_pad	O
28	SDIO_DATA1	IO	VDDPST_SD		IE										**GPIO21**	I/O/T	GPIO21	I/O/T	spi2_d_pad	I1/O/T			dbg_flash_d_pad	O
29	SDIO_DATA2	IO	VDDPST_SD		IE										**GPIO22**	I/O/T	GPIO22	I/O/T	spi2_q_pad	I1/O/T			dbg_flash_cs_pad	O
30	VDD_PSRAM_1P8_1	Power																						
31	SDIO_DATA3	IO	VDDPST_SD		IE										**GPIO23**	I/O/T	GPIO23	I/O/T	spi2_cs_pad	I1/O/T			dbg_flash_q_pad	O
32	SDIO_CLK	IO	VDDPST_SD		IE										**GPIO24**	I/O/T	GPIO24	I/O/T	spi2_hold_pad	I1/O/T			dbg_flash_wp_pad	O
33	SDIO_CMD	IO	VDDPST_SD		IE										**GPIO25**	I/O/T	GPIO25	I/O/T	spi2_wp_pad	I1/O/T			dbg_flash_hold_pad	O
34	VDD_PSRAM_1P8_2	Power																						
35	VDD_LDO_1P8	Power																						
36	SPICS	IO	VDD_SPI	WPU	WPU, IE										**flash_cs_pad**	O/T	GPIO26	I/O/T						
37	SPIQ	IO	VDD_SPI	WPU	WPU, IE										**flash_q_pad**	I1/O/T	GPIO27	I/O/T						
38	SPIWP	IO	VDD_SPI	WPU	WPU, IE										flash_wp_pad	I1/O/T	GPIO28	I/O/T						
39	VDD_SPI	Power				VDD_SPI																		
40	SPIHD	IO	VDD_SPI	WPU	WPU, IE										**flash_hold_pad**	I1/O/T	GPIO30	I/O/T						
41	SPICLK	IO	VDD_SPI	WPU	WPU, IE										**flash_ck_pad**	O/T	GPIO31	I/O/T						
42	SPID	IO	VDD_SPI	WPU	WPU, IE										**flash_d_pad**	I1/O/T	GPIO32	I/O/T						
43	VCCA/VDDPST_2	Power																						
44	USB_DP	Analog	VDDPST_2																					
45	USB_DM	Analog	VDDPST_2																					
46	GPIO33	IO	VDDPST_3	drv=3	USB_PU, IE, drv=3	USB1P1_N0									GPIO33	I/O/T	GPIO33	I/O/T			lcd_data12_out_pad	O		
47	GPIO34	IO	VDDPST_3	drv=3	USB_PU, IE, drv=3	USB1P1_P0									GPIO34	I/O/T	GPIO34	I/O/T			lcd_data13_out_pad	O		
48	GPIO35	IO	VDDPST_3		IE										**GPIO35**	I/O/T	GPIO35	I/O/T	ref_gmac_clk_pad	O	lcd_data14_out_pad	O	sd2_cdata0_pad	I1/O/T
49	GPIO36	IO	VDDPST_3	IE	IE										**GPIO36**	I/O/T	GPIO36	I/O/T	gmac_phy_rxdv_pad	IO	lcd_data15_out_pad	O	sd2_cdata1_pad	I1/O/T
50	GPIO37	IO	VDDPST_3	IE	IE	PAD COMP0 (MUX4)									**GPIO37**	I/O/T	GPIO37	I/O/T	gmac_phy_txen_pad	O	lcd_data16_out_pad	O	sd2_cdata2_pad	I1/O/T
51	GPIO38	IO	VDDPST_3	IE	IE	PAD COMP1 (MUX4)									**GPIO38**	I/O/T	GPIO38	I/O/T	gmac_phy_rxd3_pad	IO	lcd_data17_out_pad	O	sd2_cdata3_pad	I1/O/T
52	GPIO39	IO	VDDPST_3	IE	IE	PAD COMP2 (MUX4)									**GPIO39**	I/O/T	GPIO39	I/O/T	gmac_phy_rxd2_pad	IO	lcd_data18_out_pad	O	sd2_cclk_pad	O
53	GPIO40	IO	VDDPST_3	IE	IE	PAD COMP3 (MUX4)									**GPIO40**	I/O/T	GPIO40	I/O/T	gmac_phy_rxd1_pad	IO	lcd_pclk_pad	O	sd2_ccmd_pad	I1/O/T
54	VDDPST_3	Power																						
55	GPIO42	IO	VDDPST_3		IE	ADC1_CHANNEL0_N									**GPIO42**	I/O/T	GPIO42	I/O/T	gmac_rx_clk_pad	IO				
56	GPIO43	IO	VDDPST_3		IE	ADC1_CHANNEL0_P									**GPIO43**	I/O/T	GPIO43	I/O/T	gmac_rmii_clk_pad	IO/O/T	lcd_h_enable_pad	O		
57	GPIO44	IO	VDDPST_3		IE	ADC1_CHANNEL1_N									**GPIO44**	I/O/T	GPIO44	I/O/T	gmac_phy_txd0_pad	O	lcd_h_sync_pad	O		
58	GPIO45	IO	VDDPST_3		IE	ADC1_CHANNEL1_P									**GPIO45**	I/O/T	GPIO45	I/O/T	gmac_phy_txd1_pad	O	lcd_v_sync_pad	O		
59	GPIO46	IO	VDDPST_3		IE	ADC1_CHANNEL2_N									**GPIO46**	I/O/T	GPIO46	I/O/T	gmac_phy_txd2_pad	O	cam_data0_in_pad	IO		
60	GPIO47	IO	VDDPST_3		IE	ADC1_CHANNEL2_P									**GPIO47**	I/O/T	GPIO47	I/O/T	gmac_phy_txd3_pad	O	cam_data1_in_pad	IO		
61	GPIO48	IO	VDDPST_3			ADC1_CHANNEL3_N									**GPIO48**	I/O/T	GPIO48	I/O/T			cam_data2_in_pad	IO		
62	GPIO49	IO	VDDPST_3			ADC1_CHANNEL3_P									**GPIO49**	I/O/T	GPIO49	I/O/T			cam_data3_in_pad	IO		
63	VREF_ADC	Analog																						
64	VDDPST_4	Power																						
65	GPIO50	IO	VDDPST_4			ADC2_CHANNEL0_N									**GPIO50**	I/O/T	GPIO50	I/O/T			cam_data4_in_pad	IO		
66	GPIO51	IO	VDDPST_4			ADC2_CHANNEL0_P									**GPIO51**	I/O/T	GPIO51	I/O/T			cam_data5_in_pad	IO		
67	GPIO52	IO	VDDPST_4		IE	ADC2_CHANNEL1_N									**GPIO52**	I/O/T	GPIO52	I/O/T	spi2_cs_pad	I1/O/T	cam_data6_in_pad	IO		
68	GPIO53	IO	VDDPST_4		IE	ADC2_CHANNEL1_P									**GPIO53**	I/O/T	GPIO53	I/O/T	spi2_ck_pad	I1/O/T	cam_data7_in_pad	IO		
69	MTDO	IO	VDDPST_4		IE	ADC2_CHANNEL2_N									**MTDO**	O/T	GPIO54	I/O/T	spi2_d_pad	I1/O/T	cam_pclk_pad	IO		
70	MTCK	IO	VDDPST_4		IE	ADC2_CHANNEL2_P									**MTCK**	I1	GPIO55	I/O/T	spi2_q_pad	I1/O/T	cam_xclk_pad	O		
71	MTDI	IO	VDDPST_4		IE	ADC2_CHANNEL3_N									**MTDI**	I1	GPIO56	I/O/T	spi2_hold_pad	I1/O/T	cam_v_sync_pad	IO		
72	MTMS	IO	VDDPST_4		IE	ADC2_CHANNEL3_P									**MTMS**	I1	GPIO57	I/O/T	spi2_wp_pad	I1/O/T	cam_h_sync_pad	IO		
73	GPIO58	IO	VDDPST_4		IE										**uart0_txd_pad**	O	GPIO58	I/O/T						

Cont'd on next page

– Cont'd from previous page

Pin No.	Pin Name	Pin Type	Pin Providing Power	Pin Settings		Analog Functions	LP IO MUX Functions								IO MUX Functions									
				At Reset	After Reset	F0	F0	Type	F1	Type	F2	Type	F3	Type	F0	Type	F1	Type	F2	Type	F3	Type	F4	Type
74	GPIO59	IO	VDDPST_4		IE										uart0_rxd_pad	I1	GPIO59	I/O/T						
75	GPIO60	IO	VDDPST_4	WPU, IE	WPU, IE										GPIO60	I/O/T	GPIO60	I/O/T						
76	GPIO61	IO	VDDPST_4	WPU, IE	WPU, IE										GPIO61	I/O/T	GPIO61	I/O/T						
77	VDDA1	Power																						
78	XTAL_N	Analog																						
79	XTAL_P	Analog																						
80	VDDA2	Power																						

* For details, see Section 2 *Pins*. Regarding highlighted cells, see Section 2.3.4 *Restrictions for GPIOs and LP GPIOs*.

Glossary

strapping pin

A type of GPIO pin used to configure certain operational settings during the chip's power-up, and can be reconfigured as normal GPIO after the chip's reset 34

eFuse parameter

A parameter stored in an electrically programmable fuse (eFuse) memory within a chip. The parameter can be set by programming EFUSE_PGM_DATA*n*_REG registers, and read by reading a register field named after the parameter 34

SPI boot mode

A boot mode in which users load and execute the existing code from SPI flash 35

joint download boot mode

A boot mode in which users can download code into flash via the UART or other interfaces (see Table 3-3 *Chip Boot Mode Control* > Note), and load and execute the downloaded code from the flash or SRAM 35, 36

Related Documentation and Resources

Related Documentation

- *Certificates*
  https://espressif.com/en/support/documents/certificates
- *Documentation Updates and Update Notification Subscription*
  https://espressif.com/en/support/download/documents

Developer Zone

- *ESP-IDF* and other development frameworks on GitHub.
  https://github.com/espressif
- *ESP32 BBS Forum* – Engineer-to-Engineer (E2E) Community for Espressif products where you can post questions, share knowledge, explore ideas, and help solve problems with fellow engineers.
  https://esp32.com/
- *ESP-FAQ* – A summary document of frequently asked questions released by Espressif.
  https://espressif.com/projects/esp-faq/en/latest/index.html
- *The ESP Journal* – Best Practices, Articles, and Notes from Espressif folks.
  https://blog.espressif.com/
- See the tabs *SDKs and Demos*, *Apps*, *Tools*, *AT Firmware*.
  https://espressif.com/en/support/download/sdks-demos

Products

- *ESP32-S31 Series SoCs* – Browse through all ESP32-S31 SoCs.
  https://espressif.com/en/products/socs?id=ESP32-S31
- *ESP32-S31 Series Modules* – Browse through all ESP32-S31-based modules.
  https://espressif.com/en/products/modules?id=ESP32-S31
- *ESP32-S31 Series DevKits* – Browse through all ESP32-S31-based devkits.
  https://espressif.com/en/products/devkits?id=ESP32-S31
- *ESP Product Selector* – Find an Espressif hardware product suitable for your needs by comparing or applying filters.
  https://products.espressif.com/#/product-selector?language=en

Contact Us

- See the tabs *Sales Questions*, *Technical Enquiries*, *Circuit Schematic & PCB Design Review*, *Get Samples* (Online stores), *Become Our Supplier*, *Comments & Suggestions*.
  https://espressif.com/en/contact-us/sales-questions

Revision History

Date	Version	Release notes
2026-05-21	v0.2	• Removed "Confidential" classification for the ESP32-S31 chip • 3 *Boot Configurations*: Added 3.2 *Secure Debug Controller (SDC)* • 4.2.2.9 *CAN FD Controller*: Updated from TWAI to CAN FD controller description • 5.4.1 *Current Consumption in Active Mode*: Added current consumption data in Active mode • 7 *Packaging*: Added QFN80 package diagram
2026-04-03	v0.1	Draft

Disclaimer and Copyright Notice

www.espressif.com