ESP32-P4 Series

Datasheet Pre-release v0.5

MCU with one RISC-V 32-bit dual-core high-performance microprocessor and one single-core low-power microprocessor
Powerful image and voice processing capability
16 MB or 32 MB PSRAM in the chip's package
55 GPIOs, rich set of peripherals
QFN104 (10×10 mm) Package

Including:

ESP32-P4NRW16

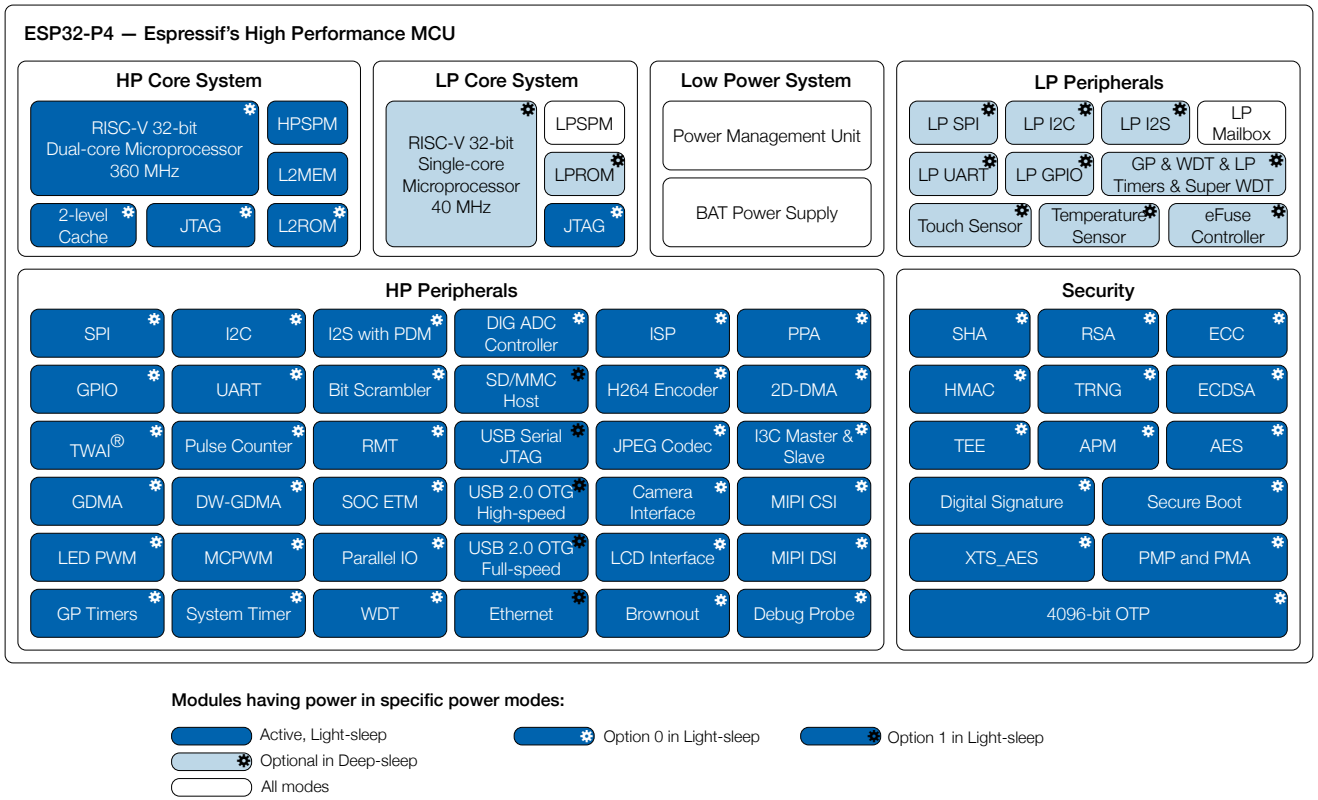
ESP32-P4NRW32

ESPRESSIF

Product Overview

ESP32-P4 is a high-performance MCU that supports large internal memory and has powerful image and voice processing capabilities. The MCU consists of a High Performance (HP) system and a Low Power (LP) system. The HP system contains a RISC-V dual-core CPU and rich peripherals, while the LP system contains a low-power RISC-V single-core CPU and various peripherals optimized for low-power applications.

The functional block diagram of the SoC is shown below.



ESP32-P4 Functional Block Diagram

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

Features

CPU and Memory

- 32-bit RISC-V dual-core processor up to 360 MHz for HP system (**The default clock frequency is configured to 360 MHz. If you require a higher clock frequency of 400 MHz, please contact us.**)
- 32-bit RISC-V single-core processor up to 40 MHz for LP system
- CoreMark® Score (dual-core):
  - at 360 MHz: 2489.62 CoreMark; 6.92 CoreMark/MHz
- 128 KB HP ROM
- 16 KB LP ROM
- 768 KB HP L2MEM
- 32 KB LP SRAM
- 8 KB system SPM (Scratchpad Memory)
- Multiple high-speed external memory interfaces
- Two-level high-speed cache

System DMA

- GDMA Controller
- VDMA Controller
- 2D-DMA Controller

Advanced Peripheral Interfaces and Sensors

- 55 programmable GPIOs
  - Five strapping GPIOs
- Image processing subsystem:
  - JPEG Codec
  - Image Signal Processor (ISP)
  - Pixel-Processing Accelerator (PPA)
  - LCD and Camera controller
  - H264 encoder
  - MIPI CSI
  - MIPI DSI
- Digital interfaces and peripherals:
  - Five UARTs

  - LP UART
  - Four SPIs
  - LP SPI
  - Two I2Cs
  - LP I2C
  - Analog I2C
  - I3C
  - Three I2Ss
  - LP I2S
  - Pulse Count Controller (PCNT)
  - USB 2.0 High-Speed OTG
  - USB 2.0 Full-Speed OTG
  - USB Serial/JTAG Controller
  - Ethernet Media Access Controller (EMAC)
  - Two-Wire Automotive Interface (TWAI)
  - SD/MMC Host Controller (SDHOST)
  - LED PWM Controller (LEDC)
  - Motor Control PWM (MCPWM)
  - Remote Control Peripheral (RMT)
  - Parallel IO Controller (PARLIO)
  - BitScrambler
  - Voice Activity Detection (VAD)
- Analog peripherals and sensors:
  - Touch sensor
  - Temperature sensor
  - Two ADC Controllers
  - Analog voltage comparator
- Timers:
  - Two 52-bit HP system timers
  - Four 54-bit HP general-purpose timers
  - Two 32-bit HP watchdog timers (MWDT)
  - 32-bit LP watchdog timer (RWDT)
  - Analog super watchdog timer (SWD)

  - 48-bit LP general-purpose timer (RTC Timer)

Security

- Secure boot
- One-time writing security ensured by eFuse OTP
- Cryptography/Security Components:
  - AES Accelerator
  - ECC Accelerator
  - HMAC Accelerator
  - RSA Accelerator
  - SHA Accelerator
  - Digital Signature Algorithm
  - Elliptic Curve Digital Signature Algorithm (ECDSA)
  - External Memory Encryption and Decryption (XTS_AES)
  - True Random Number Generator (TRNG)
- Permission Control (PMS)

Applications

With low power consumption, ESP32-P4 is an ideal choice for IoT devices in the following areas:

- Smart Home
- Industrial Automation
- Health Care
- Consumer Electronics
- Smart Agriculture
- Retail Self-Service Terminals (POS, Vending Machines)
- Service Robot
- Multimedia Player
- Cameras for Video Streaming
- High-Speed USB Host and Device
- Smart Voice Interaction Terminal
- Edge Vision AI Processor
- HMI Control Panel

Contents

List of Tables

List of Figures

1 ESP32-P4 Series Comparison

1.1 Nomenclature

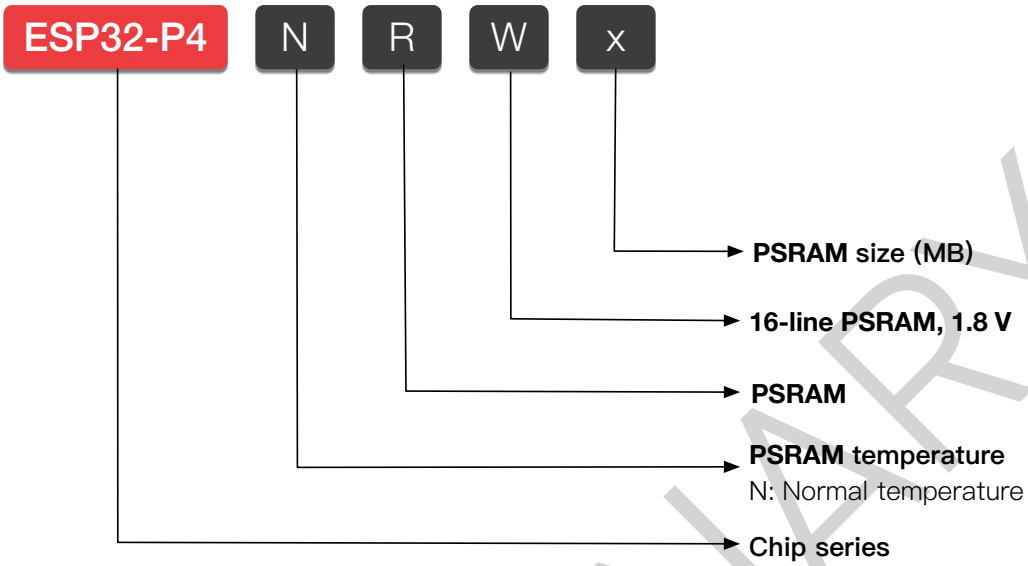


Figure 1-1. ESP32-P4 Series Nomenclature

1.2 Comparison

Table 1-1. ESP32-P4 Series Comparison

Ordering Code[1]	In-Package PSRAM	Ambient Temp.[2] (°C)	VDD_PSRAM_0/1 Voltage [3]
ESP32-P4NRW16	16 MB (OPI/HPI)[4]	–40 ~ 85	1.8 V
ESP32-P4NRW32	32 MB (OPI/HPI)[4]	–40 ~ 85	1.8 V

[1] For details on chip marking and packing, see Section 6 *Packaging*.

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

[3] For more information on VDD_PSRAM_0/1, see Section 2.6 *Power Supply*.

[4] OPI of PSRAM supports transferring eight-bit commands, addresses, and data; HPI supports transferring eight-bit commands and addresses as well as 16-bit data. For details about SPI modes, see Section 2.7 *Pin Mapping Between Chip and Flash*.

2 Pins

2.1 Pin Layout

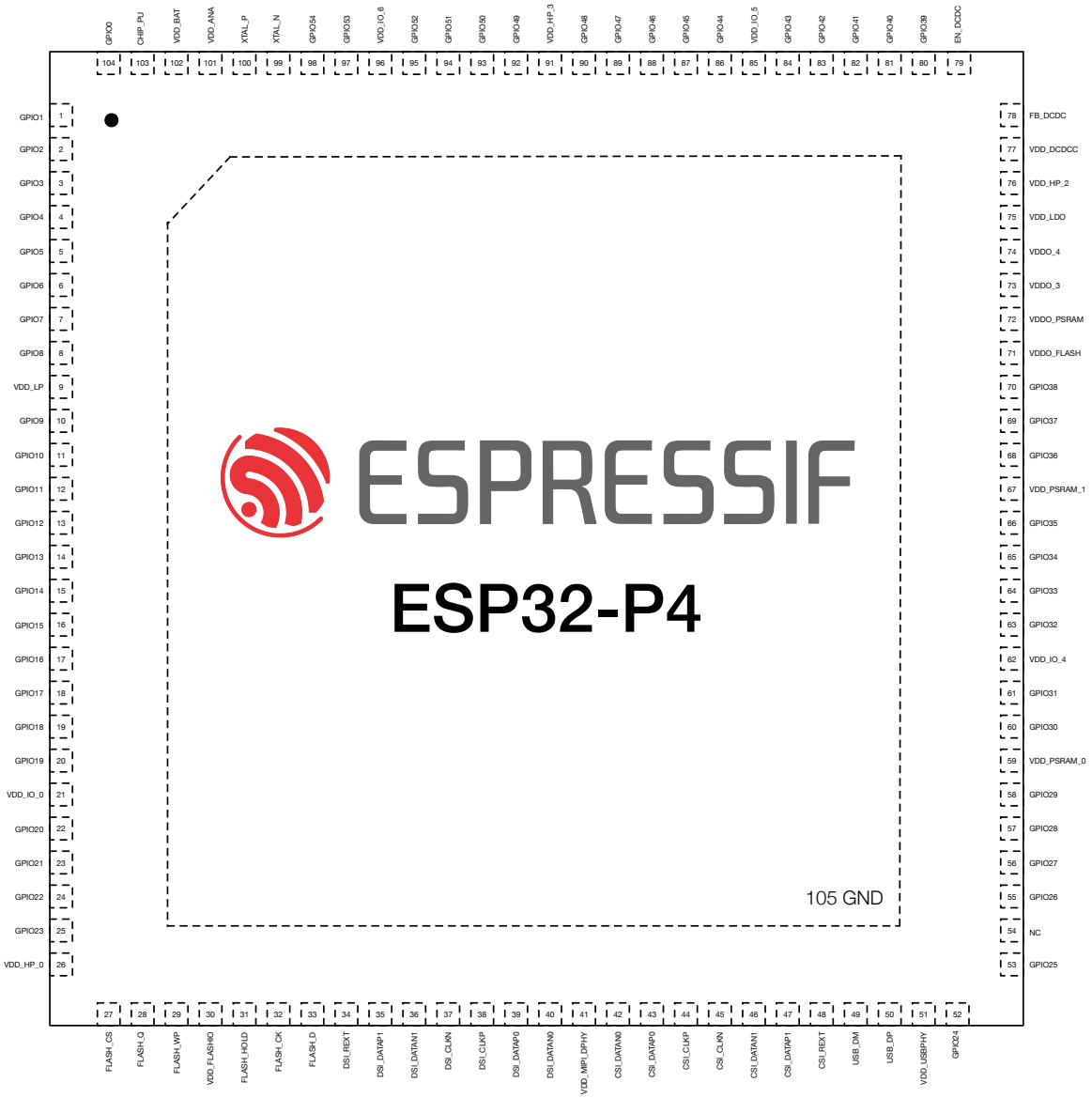


Figure 2-1. ESP32-P4 Pin Layout (Top View)

2.2 Pin Overview

The ESP32-P4 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. Pin muxing is controlled via software programmable registers (see *ESP32-P4 Technical Reference Manual* > Chapter *GPIO Matrix and IO MUX*). In addition, ESP32-P4 has a number of pins that are dedicated to certain peripherals, such as MIPI DSI and CSI, and cannot be used for general-purpose IO.

All in all, the ESP32-P4 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 2-5 *LP IO MUX Functions*
  - **Some** IO pins have predefined **analog functions** – see Table 2-7 *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.
- **Dedicated interface pins** can only be used by certain peripherals, such as flash, MIPI DSI, and MIPI CSI – see Table 2-9 *Dedicated Interface Pins*
- **Analog pins** that have exclusively-dedicated **analog functions** – see Table 2-10 *Analog Pins*
- **Power pins** that supply power to the chip components and non-power pins – see Table 2-11 *Power Pins*

Table 2-1 *Pin Overview* gives an overview of all the pins. For more information, see the respective sections for each pin type below, or Appendix A – ESP32-P4 Consolidated Pin Overview.

Table 2-1. Pin Overview

Pin No.	Pin Name	Pin Type	Pin Providing Power [2, 3]	Pin Settings [4]		Pin Function Sets [1]		
				At Reset	After Reset	IO MUX	LP IO MUX	Analog
1	GPIO1	IO	VDD_LP / VDD_BAT	–	–	**IO MUX**	LP IO MUX	Analog
2	GPIO2	IO	VDD_LP / VDD_BAT	–	IE, WPU[5]	**IO MUX**	LP IO MUX	Analog
3	GPIO3	IO	VDD_LP / VDD_BAT	–	IE	**IO MUX**	LP IO MUX	Analog
4	GPIO4	IO	VDD_LP	–	IE	**IO MUX**	LP IO MUX	Analog
5	GPIO5	IO	VDD_LP	–	–	**IO MUX**	LP IO MUX	Analog
6	GPIO6	IO	VDD_LP	–	–	**IO MUX**	LP IO MUX	Analog
7	GPIO7	IO	VDD_LP	–	–	**IO MUX**	LP IO MUX	Analog
8	GPIO8	IO	VDD_LP	–	–	**IO MUX**	LP IO MUX	Analog
9	VDD_LP	Power	–	–	–	–	–	–
10	GPIO9	IO	VDD_LP	–	–	**IO MUX**	LP IO MUX	Analog
11	GPIO10	IO	VDD_LP	–	–	**IO MUX**	LP IO MUX	Analog
12	GPIO11	IO	VDD_LP	–	–	**IO MUX**	LP IO MUX	Analog
13	GPIO12	IO	VDD_LP	–	–	**IO MUX**	LP IO MUX	Analog
14	GPIO13	IO	VDD_LP	–	–	**IO MUX**	LP IO MUX	Analog
15	GPIO14	IO	VDD_LP	–	–	**IO MUX**	LP IO MUX	Analog

Cont'd on next page

Table 2-1 – cont'd from previous page

Pin No.	Pin Name	Pin Type	Pin Providing Power [2, 3]	Pin Settings [4]		Pin Function Sets [1]		
				At Reset	After Reset	IO MUX	LP IO MUX	Analog
16	GPIO15	IO	VDD_LP	–	–	**IO MUX**	LP IO MUX	Analog
17	GPIO16	IO	VDD_IO_0	–	–	**IO MUX**	–	Analog
18	GPIO17	IO	VDD_IO_0	–	–	**IO MUX**	–	Analog
19	GPIO18	IO	VDD_IO_0	–	–	**IO MUX**	–	Analog
20	GPIO19	IO	VDD_IO_0	–	–	**IO MUX**	–	Analog
21	VDD_IO_0	Power	–	–	–	–	–	–
22	GPIO20	IO	VDD_IO_0	–	–	**IO MUX**	–	Analog
23	GPIO21	IO	VDD_IO_0	–	–	**IO MUX**	–	Analog
24	GPIO22	IO	VDD_IO_0	–	–	**IO MUX**	–	Analog
25	GPIO23	IO	VDD_IO_0	–	–	**IO MUX**	–	Analog
26	VDD_HP_0	Power	–	–	–	–	–	–
27	FLASH_CS	Dedicated	VDD_FLASHIO	–	–	–	–	–
28	FLASH_Q	Dedicated	VDD_FLASHIO	–	–	–	–	–
29	FLASH_WP	Dedicated	VDD_FLASHIO	–	–	–	–	–
30	VDD_FLASHIO	Power	–	–	–	–	–	–
31	FLASH_HOLD	Dedicated	VDD_FLASHIO	–	–	–	–	–
32	FLASH_CK	Dedicated	VDD_FLASHIO	–	–	–	–	–
33	FLASH_D	Dedicated	VDD_FLASHIO	–	–	–	–	–
34	DSI_REXT	Dedicated	VDD_MIPI_DPHY	–	–	–	–	–
35	DSI_DATAP1	Dedicated	VDD_MIPI_DPHY	–	–	–	–	–
36	DSI_DATAN1	Dedicated	VDD_MIPI_DPHY	–	–	–	–	–
37	DSI_CLKN	Dedicated	VDD_MIPI_DPHY	–	–	–	–	–
38	DSI_CLKP	Dedicated	VDD_MIPI_DPHY	–	–	–	–	–
39	DSI_DATAP0	Dedicated	VDD_MIPI_DPHY	–	–	–	–	–
40	DSI_DATAN0	Dedicated	VDD_MIPI_DPHY	–	–	–	–	–
41	VDD_MIPI_DPHY	Power	–	–	–	–	–	–
42	CSI_DATAN0	Dedicated	VDD_MIPI_DPHY	–	–	–	–	–
43	CSI_DATAP0	Dedicated	VDD_MIPI_DPHY	–	–	–	–	–
44	CSI_CLKP	Dedicated	VDD_MIPI_DPHY	–	–	–	–	–
45	CSI_CLKN	Dedicated	VDD_MIPI_DPHY	–	–	–	–	–
46	CSI_DATAN1	Dedicated	VDD_MIPI_DPHY	–	–	–	–	–
47	CSI_DATAP1	Dedicated	VDD_MIPI_DPHY	–	–	–	–	–
48	CSI_REXT	Dedicated	VDD_MIPI_DPHY	–	–	–	–	–
49	USB_DM	Dedicated	VDD_USBPHY	–	–	–	–	–
50	USB_DP	Dedicated	VDD_USBPHY	–	–	–	–	–
51	VDD_USBPHY	Power	–	–	–	–	–	–
52	GPIO24	IO	VDD_IO_4	–	–	IO MUX	–	**Analog**
53	GPIO25	IO	VDD_IO_4	–	USB_PU	IO MUX	–	**Analog**
54	NC	–	–	–	–	–	–	–
55	GPIO26	IO	VDD_IO_4	–	–	**IO MUX**	–	Analog
56	GPIO27	IO	VDD_IO_4	–	–	**IO MUX**	–	Analog
57	GPIO28	IO	VDD_IO_4	–	–	**IO MUX**	–	–
58	GPIO29	IO	VDD_IO_4	–	–	**IO MUX**	–	–
59	VDD_PSRAM_0	Power	–	–	–	–	–	–
60	GPIO30	IO	VDD_IO_4	–	–	**IO MUX**	–	–

Cont'd on next page

Table 2-1 – cont'd from previous page

Pin No.	Pin Name	Pin Type	Pin Providing Power [2, 3]	Pin Settings [4]		Pin Function Sets [1]		
				At Reset	After Reset	IO MUX	LP IO MUX	Analog
61	GPIO31	IO	VDD_IO_4	–	–	**IO MUX**	–	–
62	VDD_IO_4	Power	–	–	–	–	–	–
63	GPIO32	IO	VDD_IO_4	IE	–	**IO MUX**	–	–
64	GPIO33	IO	VDD_IO_4	IE	–	**IO MUX**	–	–
65	GPIO34	IO	VDD_IO_4	IE	–	**IO MUX**	–	–
66	GPIO35	IO	VDD_IO_4	IE, WPU	–	**IO MUX**	–	–
67	VDD_PSRAM_1	Power	–	–	–	–	–	-
68	GPIO36	IO	VDD_IO_4	IE	–	**IO MUX**	–	–
69	GPIO37	IO	VDD_IO_4	IE	–	**IO MUX**	–	–
70	GPIO38	IO	VDD_IO_4	IE	–	**IO MUX**	–	–
71	VDDO_FLASH	Power	–	–	–	–	–	-
72	VDDO_PSRAM	Power	–	–	–	–	–	-
73	VDDO_3	Power	–	–	–	–	–	-
74	VDDO_4	Power	–	–	–	–	–	-
75	VDD_LDO	Power	–	–	–	–	–	–
76	VDD_HP_2	Power	–	–	–	–	–	–
77	VDD_DCDCC	Power	–	–	–	–	–	–
78	FB_DCDC	Analog	–	–	–	–	–	–
79	EN_DCDC	Analog	–	–	–	–	–	–
80	GPIO39	IO	VDD_IO_5	–	–	**IO MUX**	–	–
81	GPIO40	IO	VDD_IO_5	–	–	**IO MUX**	–	–
82	GPIO41	IO	VDD_IO_5	–	–	**IO MUX**	–	–
83	GPIO42	IO	VDD_IO_5	–	–	**IO MUX**	–	–
84	GPIO43	IO	VDD_IO_5	–	–	**IO MUX**	–	–
85	VDD_IO_5	Power	–	–	–	–	–	–
86	GPIO44	IO	VDD_IO_5	–	–	**IO MUX**	–	–
87	GPIO45	IO	VDD_IO_5	–	–	**IO MUX**	–	–
88	GPIO46	IO	VDD_IO_5	–	–	**IO MUX**	–	–
89	GPIO47	IO	VDD_IO_5	–	–	**IO MUX**	–	–
90	GPIO48	IO	VDD_IO_5	–	–	**IO MUX**	–	–
91	VDD_HP_3	Power	–	–	–	–	–	-
92	GPIO49	IO	VDD_IO_6	–	–	**IO MUX**	–	Analog
93	GPIO50	IO	VDD_IO_6	–	–	**IO MUX**	–	Analog
94	GPIO51	IO	VDD_IO_6	–	–	**IO MUX**	–	Analog
95	GPIO52	IO	VDD_IO_6	–	–	**IO MUX**	–	Analog
96	VDD_IO_6	Power	–	–	–	–	–	-
97	GPIO53	IO	VDD_IO_6	–	–	**IO MUX**	–	Analog
98	GPIO54	IO	VDD_IO_6	–	–	**IO MUX**	–	Analog
99	XTAL_N	Analog	–	–	–	–	–	-
100	XTAL_P	Analog	–	–	–	–	–	-
101	VDD_ANA	Power	–	–	–	–	–	-
102	VDD_BAT	Power	–	–	–	–	–	-
103	CHIP_PU	Analog	–	–	–	–	–	–
104	GPIO0	IO	VDD_LP / VDD_BAT	–	–	**IO MUX**	LP IO MUX	Analog
105	GND	Power	–	–	–	–	–	–

1. **Bold** marks the pin function set in which a pin has its default function in the default boot mode. See Section 3.1 *Chip Boot Mode*

Control.

2. In column **Pin Providing Power**, regarding pins powered by VDD_LP / VDD_BAT:
   - Pin Providing Power (either VDD_LP or VDD_BAT) can be configured via a register .
3. Default drive strength for IO pins is 20 mA except for GPIO24 and GPIO25 which have default drive strength of 40 mA.
4. 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
   - USB_PU – USB pull-up resistor enabled
     - By default, the USB function is enabled for USB pins (i.e., GPIO24/26 and GPIO25/27), 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_GPIOx_FUN_WPU/WPD).
5. Depends on the value of EFUSE_DIS_PAD_JTAG
   - `0` (default), input enabled, pull-up resistor enabled (IE = 1, WPU = 1)
   - `1`, input disabled, in high impedance state (IE = 0)

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-P4 can be connected to one of the four signals (IO MUX functions, i.e., F0–F3), as listed in Table 2-3 *IO MUX Functions*.

Among the four 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 *IO MUX Functions*.

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
REF_50M_CLK_PAD	50 MHz reference clock output	Provides 50 MHz clock for internal and external modules
GMAC_PHY_RXDV_PAD 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

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

Table 2-3. IO MUX Pin Functions

Pin No.	IO MUX / GPIO Name [2]	IO MUX Function [1, 2, 3]							
		F0	Type [3]	F1	Type	F2	Type	F3	Type
1	GPIO1	**GPIO1**	I/O/T	GPIO1	I/O/T	–	–	–	–
2	GPIO2	**MTCK**	I1	GPIO2	I/O/T	–	–	–	–
3	GPIO3	**MTDI**	I1	GPIO3	I/O/T	–	–	–	–
4	GPIO4	**MTMS**	I0	GPIO4	I/O/T	–	–	–	–
5	GPIO5	**MTDO**	O/T	GPIO5	I/O/T	–	–	–	–
6	GPIO6	**GPIO6**	I/O/T	GPIO6	I/O/T	–	–	SPI2_HOLD_PAD	I1/O/T
7	GPIO7	**GPIO7**	I/O/T	GPIO7	I/O/T	–	–	SPI2_CS_PAD	I1/O/T
8	GPIO8	**GPIO8**	I/O/T	GPIO8	I/O/T	–	–	SPI2_D_PAD	I1/O/T
10	GPIO9	**GPIO9**	I/O/T	GPIO9	I/O/T	–	–	SPI2_CK_PAD	I1/O/T
11	GPIO10	**GPIO10**	I/O/T	GPIO10	I/O/T	–	–	SPI2_Q_PAD	I1/O/T
12	GPIO11	**GPIO11**	I/O/T	GPIO11	I/O/T	–	–	SPI2_WP_PAD	I1/O/T
13	GPIO12	**GPIO12**	I/O/T	GPIO12	I/O/T	–	–	–	–
14	GPIO13	**GPIO13**	I/O/T	GPIO13	I/O/T	–	–	–	–
15	GPIO14	**GPIO14**	I/O/T	GPIO14	I/O/T	–	–	–	–
16	GPIO15	**GPIO15**	I/O/T	GPIO15	I/O/T	–	–	–	–
17	GPIO16	**GPIO16**	I/O/T	GPIO16	I/O/T	–	–	–	–
18	GPIO17	**GPIO17**	I/O/T	GPIO17	I/O/T	–	–	–	–
19	GPIO18	**GPIO18**	I/O/T	GPIO18	I/O/T	–	–	–	–
20	GPIO19	**GPIO19**	I/O/T	GPIO19	I/O/T	–	–	–	–
22	GPIO20	**GPIO20**	I/O/T	GPIO20	I/O/T	–	–	–	–
23	GPIO21	**GPIO21**	I/O/T	GPIO21	I/O/T	–	–	–	–
24	GPIO22	**GPIO22**	I/O/T	GPIO22	I/O/T	–	–	–	–
25	GPIO23	**GPIO23**	I/O/T	GPIO23	I/O/T	–	–	REF_50M_CLK_PAD	O
52	GPIO24	GPIO24	I/O/T	GPIO24	I/O/T	–	–	–	–

Cont'd on next page

Table 2-3 – cont'd from previous page

Pin No.	IO MUX / GPIO Name [2]	IO MUX Function [1, 2, 3]							
		F0	Type [3]	F1	Type	F2	Type	F3	Type
53	GPIO25	GPIO25	I/O/T	GPIO25	I/O/T	–	–	–	–
55	GPIO26	**GPIO26**	I/O/T	GPIO26	I/O/T	–	–	–	–
56	GPIO27	**GPIO27**	I/O/T	GPIO27	I/O/T	–	–	–	–
57	GPIO28	**GPIO28**	I/O/T	GPIO28	I/O/T	SPI2_CS_PAD	I1/O/T	GMAC_PHY_RXDV_PAD	I0
58	GPIO29	**GPIO29**	I/O/T	GPIO29	I/O/T	SPI2_D_PAD	I1/O/T	GMAC_PHY_RXD0_PAD	I0
60	GPIO30	**GPIO30**	I/O/T	GPIO30	I/O/T	SPI2_CK_PAD	I1/O/T	GMAC_PHY_RXD1_PAD	I0
61	GPIO31	**GPIO31**	I/O/T	GPIO31	I/O/T	SPI2_Q_PAD	I1/O/T	GMAC_PHY_RXER_PAD	I0
63	GPIO32	**GPIO32**	I/O/T	GPIO32	I/O/T	SPI2_HOLD_PAD	I1/O/T	GMAC_RMII_CLK_PAD	I0
64	GPIO33	**GPIO33**	I/O/T	GPIO33	I/O/T	SPI2_WP_PAD	I1/O/T	GMAC_PHY_TXEN_PAD	O
65	GPIO34	**GPIO34**	I/O/T	GPIO34	I/O/T	SPI2_IO4_PAD	I1/O/T	GMAC_PHY_TXD0_PAD	O
66	GPIO35	**GPIO35**	I/O/T	GPIO35	I/O/T	SPI2_IO5_PAD	I1/O/T	GMAC_PHY_TXD1_PAD	O
68	GPIO36	**GPIO36**	I/O/T	GPIO36	I/O/T	SPI2_IO6_PAD	I1/O/T	GMAC_PHY_TXER_PAD	O
69	GPIO37	**UART0_TXD_PAD**	O	GPIO37	I/O/T	SPI2_IO7_PAD	I1/O/T	–	–
70	GPIO38	**UART0_RXD_PAD**	I1	GPIO38	I/O/T	SPI2_DQS_PAD	O/T	–	–
80	GPIO39	SD1_CDATA0_PAD	I1/O/T	**GPIO39**	I/O/T	–	–	REF_50M_CLK_PAD	O
81	GPIO40	SD1_CDATA1_PAD	I1/O/T	**GPIO40**	I/O/T	–	–	GMAC_PHY_TXEN_PAD	O
82	GPIO41	SD1_CDATA2_PAD	I1/O/T	**GPIO41**	I/O/T	–	–	GMAC_PHY_TXD0_PAD	O
83	GPIO42	SD1_CDATA3_PAD	I1/O/T	**GPIO42**	I/O/T	–	–	GMAC_PHY_TXD1_PAD	O
84	GPIO43	SD1_CCLK_PAD	O	**GPIO43**	I/O/T	–	–	GMAC_PHY_TXER_PAD	O
86	GPIO44	SD1_CCMD_PAD	I1/O/T	**GPIO44**	I/O/T	–	–	GMAC_RMII_CLK_PAD	I0
87	GPIO45	SD1_CDATA4_PAD	I1/O/T	**GPIO45**	I/O/T	–	–	GMAC_PHY_RXDV_PAD	I0
88	GPIO46	SD1_CDATA5_PAD	I1/O/T	**GPIO46**	I/O/T	–	–	GMAC_PHY_RXD0_PAD	I0
89	GPIO47	SD1_CDATA6_PAD	I1/O/T	**GPIO47**	I/O/T	–	–	GMAC_PHY_RXD1_PAD	I0
90	GPIO48	SD1_CDATA7_PAD	I1/O/T	**GPIO48**	I/O/T	–	–	GMAC_PHY_RXER_PAD	I0

Cont'd on next page

Table 2-3 – cont'd from previous page

Pin No.	IO MUX / GPIO Name [2]	IO MUX Function [1, 2, 3]							
		F0	Type [3]	F1	Type	F2	Type	F3	Type
92	GPIO49	**GPIO49**	I/O/T	GPIO49	I/O/T	–	–	GMAC_PHY_TXEN_PAD	O
93	GPIO50	**GPIO50**	I/O/T	GPIO50	I/O/T	–	–	GMAC_RMII_CLK_PAD	I0
94	GPIO51	**GPIO51**	I/O/T	GPIO51	I/O/T	–	–	GMAC_PHY_RXDV_PAD	I0
95	GPIO52	**GPIO52**	I/O/T	GPIO52	I/O/T	–	–	GMAC_PHY_RXD0_PAD	I0
97	GPIO53	**GPIO53**	I/O/T	GPIO53	I/O/T	–	–	GMAC_PHY_RXD1_PAD	I0
98	GPIO54	**GPIO54**	I/O/T	GPIO54	I/O/T	–	–	GMAC_PHY_RXER_PAD	I0
104	GPIO0	**GPIO0**	I/O/T	GPIO0	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* = 0–3) 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* will not work. That is where the LP IO MUX comes in. It allows multiple input/output signals to be a single input/output pin in Deep-sleep mode, as the pin is connected to the LP system and powered by VDD_LP or VDD_BAT.

LP IO pins can be assigned to **LP IO MUX functions**. They can

- Either work as LP GPIOs (**LP_GPIO0, LP_GPIO1, etc.**), connected to the LP CPU
- Or connect to LP peripheral signals (**LP_UART_TXD_PAD, LP_UART_RXD_PAD**) - see Table 2-4 *LP IO MUX Functions*

Table 2-4. LP Peripheral Signals Routed via LP IO MUX

Pin Function	Signal	Description
LP_UART_TXD_PAD	Transmit data	LP UART interface
LP_UART_RXD_PAD	Receive data	

Table 2-5 *LP IO MUX Functions* shows the LP functions of LP IO pins.

Table 2-5. LP IO MUX Functions

Pin No.	LP IO Name [1, 2]	LP IO MUX Function			
		F0	Type	F1	Type
1	LP_GPIO1	LP_GPIO1	I/O/T	LP_GPIO1	I/O/T
2	LP_GPIO2	LP_GPIO2	I/O/T	LP_GPIO2	I/O/T
3	LP_GPIO3	LP_GPIO3	I/O/T	LP_GPIO3	I/O/T
4	LP_GPIO4	LP_GPIO4	I/O/T	LP_GPIO4	I/O/T
5	LP_GPIO5	LP_GPIO5	I/O/T	LP_GPIO5	I/O/T
6	LP_GPIO6	LP_GPIO6	I/O/T	LP_GPIO6	I/O/T
7	LP_GPIO7	LP_GPIO7	I/O/T	LP_GPIO7	I/O/T
8	LP_GPIO8	LP_GPIO8	I/O/T	LP_GPIO8	I/O/T
10	LP_GPIO9	LP_GPIO9	I/O/T	LP_GPIO9	I/O/T
11	LP_GPIO10	LP_GPIO10	I/O/T	LP_GPIO10	I/O/T
12	LP_GPIO11	LP_GPIO11	I/O/T	LP_GPIO11	I/O/T
13	LP_GPIO12	LP_GPIO12	I/O/T	LP_GPIO12	I/O/T
14	LP_GPIO13	LP_GPIO13	I/O/T	LP_GPIO13	I/O/T
15	LP_UART_TXD_PAD	LP_UART_TXD_PAD	O	LP_GPIO14	I/O/T
16	LP_UART_RXD_PAD	LP_UART_RXD_PAD	I1	LP_GPIO15	I/O/T
104	LP_GPIO0	LP_GPIO0	I/O/T	LP_GPIO0	I/O/T

[1] This column lists the LP GPIO names, since LP functions are configured with 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 touch sensor and ADC) in any power mode. Internal analog signals are routed to these analog functions, see Table 2-6 *Analog Functions*.

Table 2-6. Analog Signals Routed to Analog Functions

Pin Function	Signal	Description
XTAL_32K_N XTAL_32K_P	Negative clock signal Positive clock signal	32 kHz external clock input/output connected to the oscillator
TOUCH_CHANNEL...	Touch sensor channel signal	Touch sensor interface
ADC..._CHANNEL...	ADC1/2 channel signal	ADC1/2 interface
USB1P1_N... USB1P1_P...	USB D- USB D+	USB 2.0 full-speed OTG interface and USB Serial/JTAG function
ANA_COMP...	Voltage of P0/P1	Analog voltage comparator 0/1 interface

Table 2-7 *Analog Functions* shows the analog functions of IO pins.

Table 2-7. Analog Functions

Pin No.	Analog IO Name	Analog Function [1]	
		F0	F1
1	GPIO1	XTAL_32K_P	–
2	GPIO2	TOUCH_CHANNEL1	–
3	GPIO3	TOUCH_CHANNEL2	–
4	GPIO4	TOUCH_CHANNEL3	–
5	GPIO5	TOUCH_CHANNEL4	–
6	GPIO6	TOUCH_CHANNEL5	–
7	GPIO7	TOUCH_CHANNEL6	–
8	GPIO8	TOUCH_CHANNEL7	–
10	GPIO9	TOUCH_CHANNEL8	–
11	GPIO10	TOUCH_CHANNEL9	–
12	GPIO11	TOUCH_CHANNEL10	–
13	GPIO12	TOUCH_CHANNEL11	–
14	GPIO13	TOUCH_CHANNEL12	–
15	GPIO14	TOUCH_CHANNEL13	–
16	GPIO15	TOUCH_CHANNEL14	–
17	GPIO16	ADC1_CHANNEL0	–
18	GPIO17	ADC1_CHANNEL1	–
19	GPIO18	ADC1_CHANNEL2	–
20	GPIO19	ADC1_CHANNEL3	–
22	GPIO20	ADC1_CHANNEL4	–
23	GPIO21	ADC1_CHANNEL5	–
24	GPIO22	ADC1_CHANNEL6	–

Cont'd on next page

Table 2-7 – cont'd from previous page

Pin No.	Analog IO Name	Analog Function [1]	
		F0	F1
25	GPIO23	ADC1_CHANNEL7	–
52	GPIO24	**USB1P1_N0**	–
53	GPIO25	**USB1P1_P0**	–
55	GPIO26	USB1P1_N1	–
56	GPIO27	USB1P1_P1	–
92	GPIO49	ADC2_CHANNEL0	–
93	GPIO50	ADC2_CHANNEL1	–
94	GPIO51	ADC2_CHANNEL2	ANA_COMP0
95	GPIO52	ADC2_CHANNEL3	ANA_COMP0
97	GPIO53	ADC2_CHANNEL4	ANA_COMP1
98	GPIO54	ADC2_CHANNEL5	ANA_COMP1
104	GPIO0	XTAL_32K_N	–

[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*.

2.3.4 Restrictions for GPIOs and LP GPIOs

All IO pins of ESP32-P4 have GPIO pin functions, 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 one of the following important functions:

- **Strapping pins** – need to be at certain logic levels at startup. See Section 3 *Boot Configurations*.
- **USB1P1_N0/P0** – by default, connected to the USB Serial/JTAG Controller. To function as GPIOs, these pins need to be reconfigured.
- **JTAG interface** – often used for debugging. See Table 2-2 *IO MUX Functions*. To free these pins up, the pin functions USB1P1_N/P 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 *IO MUX Functions*.

See also Appendix A – ESP32-P4 Consolidated Pin Overview.

2.4 Dedicated Interface Pins

Some pins are dedicated to a few important peripherals, such as MIPI DSI and MIPI CSI.

Table 2-8. Peripheral-Dedicated Signals

Pin Function	Signal	Description
FLASH_CS FLASH_Q FLASH_WP FLASH_HOLD FLASH_CK FLASH_D	Chip select Data output Write protect Hold Clock Data in	Flash connection
MIPI DSI PHY 4.02 kΩ EXTERNAL RESISTOR MIPI DSI PHY DATAP... MIPI DSI PHY DATAN... MIPI DSI PHY CLKN MIPI DSI PHY CLKP	External resistor 4.02 kΩ Data positive channel 0/1 Data negative channel 0/1 Clock negative channel Clock positive channel	MIPI DSI connection
MIPI CSI PHY 4.02 kΩ EXTERNAL RESISTOR MIPI CSI PHY DATAP... MIPI CSI PHY DATAN... MIPI CSI PHY CLKN MIPI CSI PHY CLKP	External resistor 4.02 kΩ Data positive channel 0/1 Data negative channel 0/1 Clock negative channel Clock positive channel	MIPI CSI connection
USB2 OTG PHY DM USB2 OTG PHY DP	USB D- USB D+	USB 2.0 high-speed OTG connection

Table 2-9 *Dedicated Interface Pins* lists the peripheral-dedicated functions of pins.

Table 2-9. Dedicated Interface Pins

Pin No.	Dedicated Interface Pin	Function [1]	
		F0	Type
27	FLASH_CS	FLASH_CS	O
28	FLASH_Q	FLASH_Q	I/O/T
29	FLASH_WP	FLASH_WP	I/O/T
31	FLASH_HOLD	FLASH_HOLD	I/O/T
32	FLASH_CK	FLASH_CK	O
33	FLASH_D	FLASH_D	I/O/T
34	DSI_REXT	MIPI DSI PHY 4.02 KΩ EXTERNAL RESISTOR	I/O/T
35	DSI_DATAP1	MIPI DSI PHY DATAP1	I/O/T
36	DSI_DATAN1	MIPI DSI PHY DATAN1	I/O/T
37	DSI_CLKN	MIPI DSI PHY CLKN	I/O/T
38	DSI_CLKP	MIPI DSI PHY CLKP	I/O/T
39	DSI_DATAP0	MIPI DSI PHY DATAP0	I/O/T

Cont'd on next page

Table 2-9 – cont'd from previous page

Pin No.	Dedicated Interface Pin	Function [1]	
		F0	Type
40	DSI_DATAN0	MIPI DSI PHY DATAN0	I/O/T
42	CSI_DATAN0	MIPI CSI PHY DATAN0	I/O/T
43	CSI_DATAP0	MIPI CSI PHY DATAP0	I/O/T
44	CSI_CLKP	MIPI CSI PHY CLKP	I/O/T
45	CSI_CLKN	MIPI CSI PHY CLKN	I/O/T
46	CSI_DATAN1	MIPI CSI PHY DATAN1	I/O/T
47	CSI_DATAP1	MIPI CSI PHY DATAP1	I/O/T
48	CSI_REXT	MIPI CSI PHY 4.02 kΩ EXTERNAL RESISTOR	I/O/T
49	USB_DM	USB2 OTG PHY DM	I/O/T
50	USB_DP	USB2 OTG PHY DP	I/O/T

2.5 Analog Pins

Table 2-10. Analog Pins

Pin No.	Pin Name	Pin Type	Pin Function
78	FB_DCDC	—	Feedback pin of power supply for external DC/DC. It regulates the voltage of VDD_HP_0/2/3 together with feedback resistors of external DC/DC
79	EN_DCDC	O	Enable pin of external DC/DC
99	XTAL_N	—	External clock input/output connected to chip's crystal or oscillator. P/N means differential clock positive/negative.
100	XTAL_P	—	
103	CHIP_PU	I	High: on, enables the chip (powered up). Low: off, disables the chip (powered down). Note: Do not leave the CHIP_PU pin floating.

2.6 Power Supply

2.6.1 Power Pins

The chip is powered via the power pins described in Table 2-11 *Power Pins*.

Table 2-11. Power Pins

Pin No.	Pin Name	Direction	Power Supply [1]	
			Power Domain / Other [3]	IO Pins
9	VDD_LP	Input	LP power domain	LP IO [4]
21	VDD_IO_0	Input	Digital power domain	HP IO
26	VDD_HP_0	Input	Digital power domain	
30	VDD_FLASHIO[2]	Input	Flash	flash IO
41	VDD_MIPI_DPHY	Input	MIPI PHY	MIPI IO
51	VDD_USBPHY	Input	USB PHY	High-speed USB IO
59	VDD_PSRAM_0	Input	PSRAM	PSRAM IO
62	VDD_IO_4	Input	Digital power domain	HP IO
67	VDD_PSRAM_1	Input	PSRAM	PSRAM IO
71	VDDO_FLASH	Output	Off-package flash, output 50 mA current at the maximum	
72	VDDO_PSRAM	Output	In-package and off-package PSRAM, output 50 mA current at the maximum	
73	VDDO_3	Output	Output 50 mA current at the maximum	
74	VDDO_4	Output	Output 50 mA current at the maximum	
75	VDD_LDO	Input	Analog power domain, providing power for LDOs	
76	VDD_HP_2	Input	Digital power domain	
77	VDD_DCDCC	Input	Analog power domain, providing power for DC/DC	
85	VDD_IO_5	Input	Digital power domain	HP IO
91	VDD_HP_3	Input	Digital power domain	
96	VDD_IO_6	Input	Digital power domain	HP IO
101	VDD_ANA	Input	Analog power domain	
102	VDD_BAT	Input	Analog power domain, connecting to external batteries optionally	
105	GND	—	External ground connection	

[1] See in conjunction with Section 2.6.2 *Power Scheme*.

[2] VDD_FLASHIO provides power for flash IO, and the voltage should be adjusted according to the specific flash model. In this document, all related descriptions are based on a 3.3 V flash as an example.

[3] For recommended and maximum voltage and current, see Section 5.1 *Absolute Maximum Ratings* and Section 5.2 *Recommended Operating Conditions*.

[4] LP IO pins are those powered by VDD_LP or VDD_BAT, as shown in Figure 2-2 *ESP32-P4 Power Scheme*. See also Table 2-1 *Pin Overview* > Column *Pin Providing Power*.

2.6.2 Power Scheme

The power scheme is shown in Figure 2-2 *ESP32-P4 Power Scheme*.

The components on the chip are powered via voltage regulators.

Table 2-12. Voltage Regulators

Voltage Regulator	Output	Power Supply
HP LDO	1.1 V	HP power domain
LP LDO	1.1 V	LP power domain
Flash LDO	1.8 V/3.3 V	Can be configured to power off-package flash
VDD_PSRAM LDO	1.9 V	Can be configured to power in-package PSRAM
VO3 LDO	0.5 ~ 2.7 V/3.3 V	Can be configured to power external devices
VO4 LDO	0.5 ~ 2.7 V/3.3 V	Can be configured to power external devices

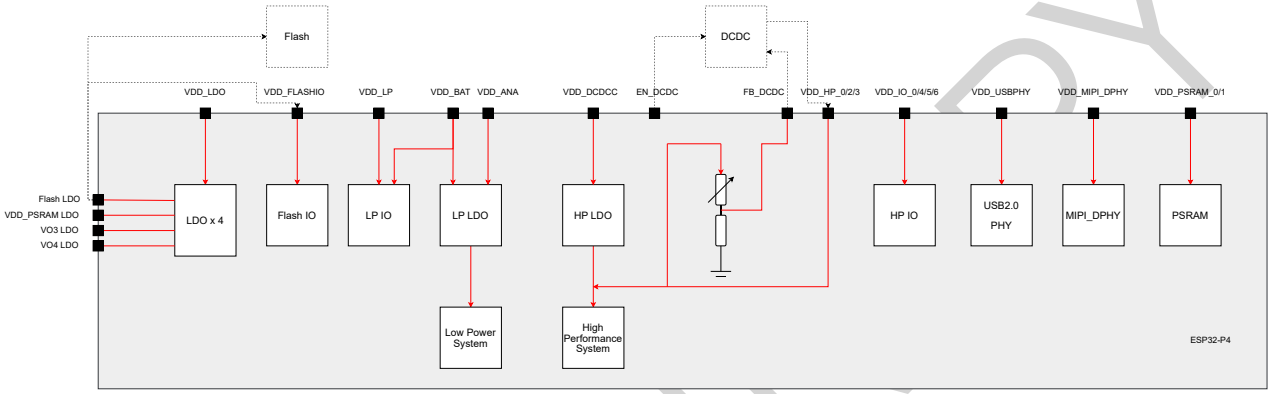


Figure 2-2. ESP32-P4 Power Scheme

2.6.3 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 2-13.

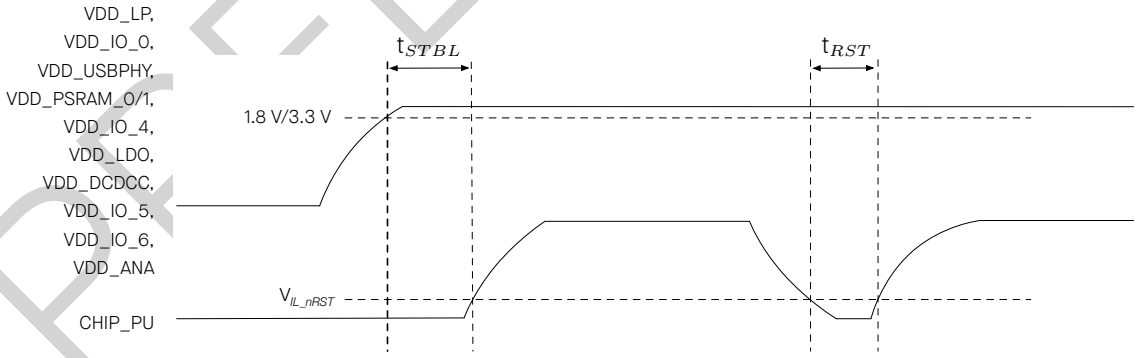


Figure 2-3. Visualization of Timing Parameters for Power-up and Reset

Table 2-13. Description of Timing Parameters for Power-up and Reset

Parameter	Description	Min (μs)
t_{STBL}	Time reserved for the power rails of VDD_LP, VDD_IO_0, VDD_USBPHY, VDD_PSRAM_0/1, VDD_IO_4, VDD_LDO, VDD_DCDCC, VDD_IO_5, VDD_IO_6 and VDD_ANA to stabilize before the CHIP_PU pin is pulled high to activate the chip	50
t_{RST}	Time reserved for CHIP_PU to stay below V_{IL_nRST} to reset the chip (see Table 5-4)	1000

2.7 Pin Mapping Between Chip and Flash

ESP32-P4 requires off-package flash to store application firmware and data. ESP32-P4 supports up to 64 MB flash, which can be connected through SPI, Dual SPI, and Quad SPI/QPI.

ESP32-P4 includes sixteen-line PSRAM with the operation voltage of 1.8 V. Please note that PSRAM is not pinned out.

Table 2-14 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 2-14. Pin Mapping Between Chip and off-package Flash

Pin No.	Pin Name	Single SPI	Dual SPI	Quad SPI/QPI
27	FLASH_CS	CS#	CS#	CS#
28	FLASH_Q	DO	DO	DO
29	FLASH_WP	WP#	WP#	WP#
31	FLASH_HOLD	HOLD#	HOLD#	HOLD#
32	FLSH_CK	CLK	CLK	CLK
33	FLSHA_D	DI	DI	DI

3 Boot Configurations

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

- **Chip boot mode**
  - Strapping pin: GPIO35, GPIO36, GPIO37 and GPIO38
- **VDDO_FLASH Voltage**
  - eFuse bit: EFUSE_0PXA_TIEH_SEL_0
- **ROM message printing**
  - Strapping pin: GPIO36
  - eFuse bit: EFUSE_UART_PRINT_CONTROL
- **JTAG signal source**
  - Strapping pin: GPIO34
  - eFuse bit: EFUSE_DIS_PAD_JTAG, EFUSE_DIS_USB_JTAG, and EFUSE_JTAG_SEL_ENABLE

The default values of all the above eFuse bits are 0, which means that they are not burnt. Given that eFuse is one-time programmable, once an eFuse bit is 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
GPIO34	Floating	–
GPIO35	Weak pull-up	1
GPIO36	Floating	–
GPIO37	Floating	–
GPIO38	Floating	–

To change the bit values, the strapping pins should be connected to external pull-down/pull-up resistors. If the ESP32-P4 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

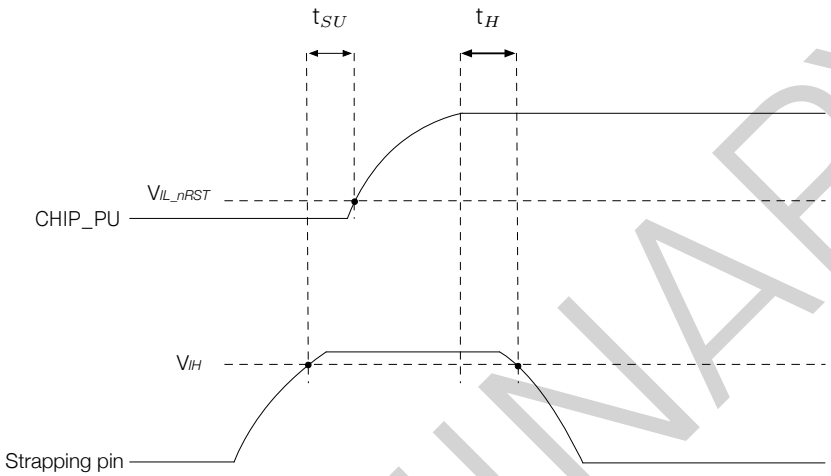


Figure 3-1. Visualization of Timing Parameters for the Strapping Pins

3.1 Chip Boot Mode Control

GPIO35–GPIO38 control the boot mode after the reset is released. See Table 3-3 *Chip Boot Mode Control*.

Table 3-3. Boot Mode Control

Boot Mode	GPIO35	GPIO36	GPIO37[3]	GPIO38[3]
SPI Boot	**1**	Any value	Any value	Any value
Joint Download Boot[2]	0	1	Any value	Any value

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

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

- USB Download Boot:
  - USB-Serial-JTAG Download Boot
  - USB 2.0 OTG Download Boot
- UART Download Boot
- SPI Slave Download Boot

[3] For details about the functionalities of GPIO37 and GPIO38, see *ESP32-P4 Technical Reference Manual* > Chapter *Chip Boot Control*.

In SPI Boot mode, the ROM bootloader loads and executes the program from SPI flash to boot the system.

In Joint Download Boot mode, users can download binary files into flash using USB, UART0, or SPI slave interface. It is also possible to download binary files into L2MEM and execute them from L2MEM.

In addition to SPI Boot and Joint Download Boot modes, ESP32-P4 also supports SPI Download Boot mode. For details, please see *ESP32-P4 Technical Reference Manual* > Chapter *Chip Boot Control*.

3.2 VDDO_FLASH Voltage Control

ESP32-P4 supplies power to flash via VDDO_FLASH, which outputs 3.3 V by default. After burning EFUSE_0PXA_TIEH_SEL_0, the output changes to 1.8 V.

Table 3-4. VDDO_FLASH Voltage Control

VDDO_FLASH power source [2]	EFUSE_0PXA_TIEH_SEL_0	Voltage
Flash LDO	**0**	**3.3 V**
	2	1.8 V

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

[2] See Section 2.6.2 *Power Scheme*.

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 GPIO36 control ROM messages printing to **UART0** as shown in Table 3-5 *UART0 ROM Message Printing Control*.

Table 3-5. UART0 ROM Message Printing Control

UART0 ROM Code Printing	EFUSE_UART_PRINT_CONTROL	GPIO36
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 3-6 *USB Serial/JTAG ROM Message Printing Control*.

Table 3-6. 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 GPIO34 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 3-7 *JTAG Signal Source Control* shows, GPIO34 is used in combination with EFUSE_DIS_PAD_JTAG, EFUSE_DIS_USB_JTAG, and EFUSE_JTAG_SEL_ENABLE.

Table 3-7. JTAG Signal Source Control

JTAG Signal Source	EFUSE_DIS_PAD_JTAG	EFUSE_DIS_USB_JTAG	EFUSE_JTAG_SEL_ENABLE	GPIO34
USB Serial/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, DMA controllers, 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-P4 has an HP 32-bit RISC-V dual-core processor with the following features:

Feature List

- Five-stage pipeline that supports clock frequency of up to 360 MHz
- RV32IMAFC ISA (instruction set architecture)
- Zc extensions (Zcb, Zcmp, and Zcmt)
- Custom AI and DSP extension (XespV)
- Custom hardware loop instructions (XespLoop)
- 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 three hardware breakpoints/watchpoints
- Up to 16 PMP/PMA regions
- Machine 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-P4 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 Processor Instruction Extensions

The ESP32-P4 HP 32-bit RISC-V dual-core processor supports standard RV32IMAFCZc extensions, and it also contains a custom extended instruction set Xhwlp which reduces the number of instructions in the loop body to improve performance, and a custom AI and DSP extension Xai to improve operation efficiency of specific AI and DSP algorithms.

Feature List

- Eight new 128-bit general-purpose registers
- 128-bit vector operations, including complex multiplication, addition, subtraction, multiplication, shifting, and comparison
- Combined data handling instructions and load/store operation instructions
- Aligned and unaligned 128-bit vector data load/store
- Configurable rounding and saturation modes

4.1.1.4 Low-Power CPU

ESP32-P4 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
- Wake-up interrupt for 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-P4 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.

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 (only internal memory is accessible): no requirements
  - 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
- INCR burst transfer when accessing memory
- Three transmit channels and three receive channels for each controller
- Software-configurable selection of peripheral requesting its service
- Configurable channel priority and weight arbitration
- Support for memory transfer
- CRC calculation of data

4.1.2.2 VDMA Controller (VDMA)

DMA (Direct Memory Access) enables direct access to system memory or peripherals without CPU involvement. The VDMA controller on ESP32-P4 is a general-purpose DMA that performs high-speed data transfer from memory to memory, from memory to peripheral, and from peripheral to memory. The VDMA complies with the AXI3 protocol and includes two AXI master interfaces. This design allows users to select between the two interfaces for data transfer dynamically.

Feature List

- Four channels for unidirectional data transfer from source to destination
- Two AXI master interfaces
- Handshake with MIPI DSI (Display Serial Interface) and ISP (Image Signal Processor)
- Memory-to-memory, ISP-to-memory, and MIPI DSI-to-memory transfer types
- Multiple levels of DMA transfer hierarchy
- Configurable transfer type, transfer length, and transfer size for each channel
- Single-block transfer
- Multi-block transfer based on contiguous address, automatic reloading register configuration, shadow registers, and linked lists
- Independent configuration of multi-block transfer type for source transfer and destination transfer

- Channel disabling without data loss
- Channel suspension, resume, and abortion
- Configurable priorities among arbitration channels
- Flow control using VDMA or peripherals
- Programmable mapping between peripherals and channels

4.1.2.3 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
- Three memory-to-peripheral channels, and two 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 *Address Mapping Structure* illustrates the address mapping structure of ESP32-P4.

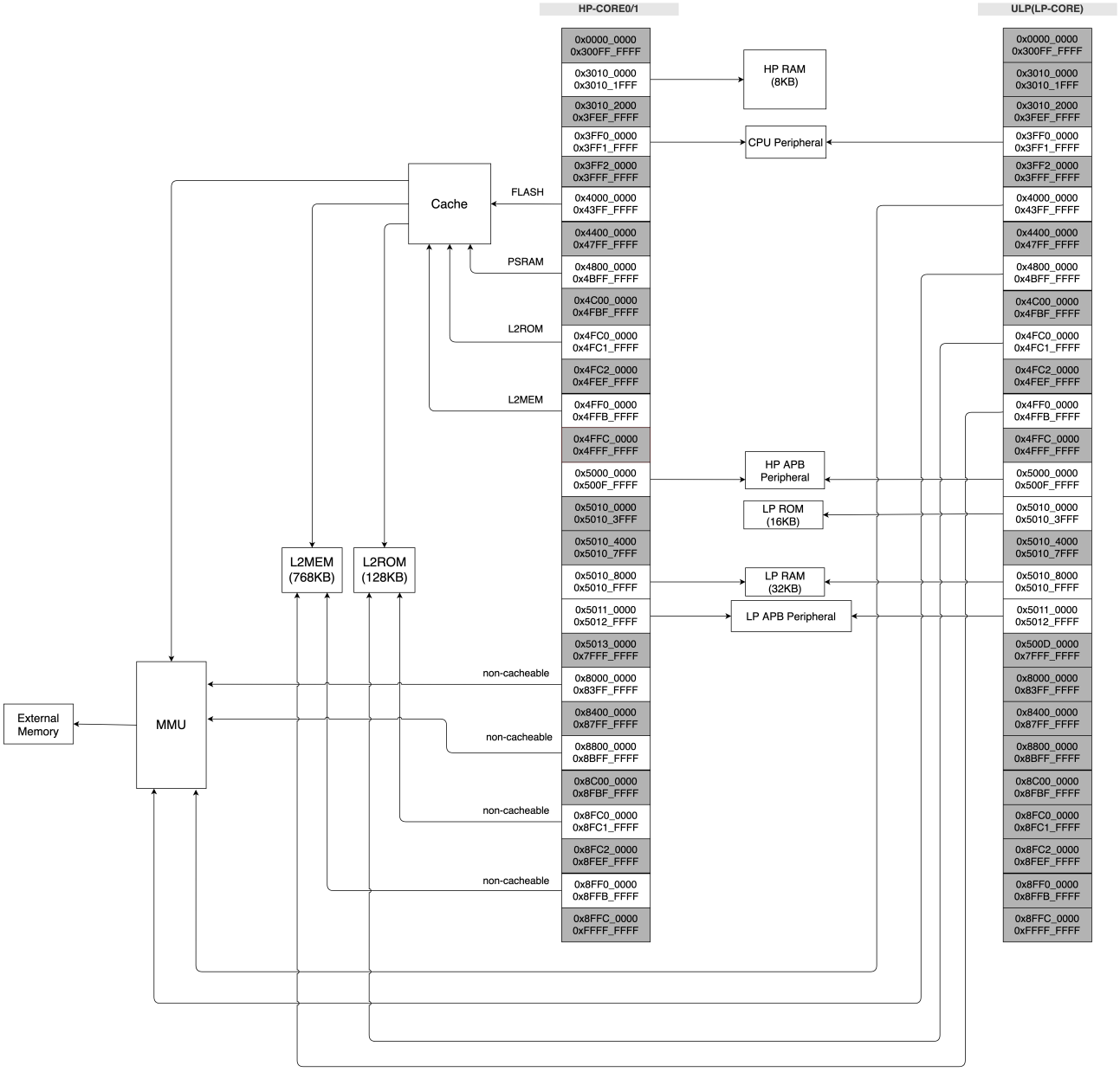


Figure 4-1. Address Mapping Structure

4.1.3.1 System and Memory

Internal Memory

ESP32-P4's internal memory includes:

- **128 KB of HP ROM**: 200 MHz, for HP CPU booting and core functions
- **768 KB of HP L2MEM**: 200 MHz, for HP CPU data and instructions
- **16 KB of LP ROM**: 40 MHz, for LP CPU booting and core functions
- **32 KB of LP SRAM**: 40 MHz, for LP CPU data and instructions
- **4 Kbit of eFuse**: 1792 bits are reserved for user data, such as encryption key and device ID
- **8 KB of SPM (Scratchpad Memory)**: 360 MHz, for HP CPU fast access

External Flash and RAM

ESP32-P4 supports SPI, Dual SPI, Quad SPI, QPI interfaces that allow connection to external flash; supports OPI and HPI interfaces that allow connection to external RAM.

The external flash and RAM can be mapped into the CPU instruction memory space and read-only data memory space. The external RAM can also be mapped into the CPU data memory space. ESP32-P4 supports up to 64 MB of external flash and 64 MB of external RAM, and hardware encryption/decryption based on XTS-AES to protect users' programs and data in flash and external RAM.

Through high-speed caches, ESP32-P4 can support at a time up to:

- External flash or RAM mapped into 64 MB instruction space as individual blocks of 64 KB
- External RAM mapped into 64 MB data space as individual blocks of 64 KB. 8-bit, 16-bit, 32-bit, and 128-bit reads and writes are supported. External flash can also be mapped into 64 MB data space as individual blocks of 64 KB, supporting 8-bit, 16-bit, 32-bit, and 128-bit reads.

Note:
After ESP32-P4 is initialized, firmware can customize the mapping of external RAM or flash into the CPU address space.

4.1.3.2 eFuse Controller

ESP32-P4 contains a 4096-bit eFuse memory to store parameters and user data. The parameters include control parameters for some hardware modules, system data parameters and keys used for the encryption/decryption module. Once an eFuse bit is programmed to 1, it can never be reverted to 0.

Feature List

- 4096-bit one-time programmable memory (including up to 1792 bits reserved for custom use)
- Configurable write protection
- Configurable read protection
- Various hardware encoding schemes against data corruption

4.1.3.3 Cache

ESP32-P4 employs the two-level cache structure.

Feature List

- 16 KB of L1 instruction cache, 64 B of block size, four-way set associative
- 64 KB of L1 data cache, 64 B of block size, two-way set associative, supporting two writing strategies write-through and write-back
- 128 KB/256 KB/512 KB of L2 cache, 64 B/128 B of block size, eight-way set associative
- 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-P4 chip features 55 GPIO pins, including 16 low-power (LP) GPIO pins and 39 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 55 GPIO pins are numbered from GPIO0 to GPIO54.

- LP GPIO pins (GPIO0–GPIO15) can be used by either HP or LP peripherals.
- HP GPIO pins (GPIO16–GPIO54) 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
- 222 HP peripheral input signals sourced from the input of any GPIO pins
- 232 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 55 GPIOs (GPIO0–GPIO54) 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
- 14 LP peripheral input signals sourced from the input of any LP GPIO pins
- 14 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 feature:

- Control of 16 LP GPIO pins (GPIO0–GPIO15) 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-P4 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 CPI1, 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-P4 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-P4 clocks can be classified into two types depending on their frequencies:

- High speed clocks for devices working at a higher frequency, such as HP CPU0/1 and digital peripherals
  - CPLL_CLK: internal 360 MHz PLL clock. Its reference clock is XTAL_CLK
  - MPLL_CLK: internal 500 MHz PLL clock. Its reference clock is XTAL_CLK
  - SPLL_CLK: internal 480 MHz PLL clock. Its reference clock is XTAL_CLK
- Slow speed clocks for LP system and some peripherals working in low-power mode
  - XTAL32K_CLK: external 32 kHz crystal clock
  - RC_SLOW_CLK: internal slow RC oscillator with adjustable frequency (150 kHz by default)
  - OSC_SLOW_CLK: external slow clock input through XTAL_32K_N, with a frequency of 32 kHz by default. After configuring this GPIO, also configure the Hold function
  - XTAL_CLK: 40 MHz external crystal clock
  - RC_FAST_CLK: internal fast RC oscillator with adjustable frequency (20 MHz by default)
  - PLL_LP_CLK: internal PLL clock with a frequency of 8 MHz by default. Its reference clock can be XTAL32K_CLK

4.1.4.4 Interrupt Matrix

The Interrupt Matrix in the ESP32-P4 chip routes interrupt requests generated by various peripherals to CPU interrupts.

Feature List

- 126 peripheral interrupt sources accepted as input
- 32 HP CPU0 peripheral interrupts and 32 HP CPU1 peripheral interrupts generated to HP CPU as output
- Current interrupt status query of peripheral interrupt sources
- Multiple interrupt sources mapping to a single HP CPU0 or HP CPU1 interrupt (i.e., shared interrupts)

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 Low-Power Management

With advanced power-management technologies, ESP32-P4 can switch between different power modes.

- Active mode: CPU and all peripherals are powered on.
- Light-sleep mode: CPU is paused. Any wake-up events (host, RTC timer, or external interrupts) will wake up the chip. CPU (excluding L2MEM) and most peripherals (See ESP32-P4 Block Diagram) can also be powered down based on requirements to further reduce power consumption.
- Deep-sleep mode: CPU (including L2MEM) and most peripherals (See ESP32-P4 Block Diagram) are powered down. Only the LP memory is powered on, and some peripherals of the LP system can be powered down based on requirements.

4.1.4.7 System Timer

ESP32-P4 provides a 52-bit system timer, which can be used to generate tick interrupts for the operating system, or be used as a general timer to generate periodic interrupts or one-time interrupts.

Feature List

- Two 52-bit counters and three 52-bit comparators
- Software accessing registers clocked by APB_CLK
- CNT_CLK used for counting, with an average frequency of 16 MHz in two counting cycles
- 40 MHz XTAL_CLK as the clock source of CNT_CLK
- 52-bit alarm values (t) and 26-bit alarm periods (δt)
- Two modes to generate alarms:
  - Target mode: only a one-time alarm is generated based on the alarm value (t)
  - Period mode: periodic alarms are generated based on the alarm period (δt)
- Three comparators generating three independent interrupts based on configured alarm value (t) or alarm period (δt)
- Software configuring the reference count value. For example, the system timer is able to load back the sleep time recorded by RTC timer via software after Light-sleep
- Able to stall or continue running when CPU stalls or enters the on-chip-debugging mode
- Alarm for Event Task Matrix (ETM) event

4.1.4.8 Timer Group (TIMG)

ESP32-P4 chip contains two timer groups. Each timer group consists of two general-purpose timers and one Main System Watchdog Timer (MWDT). The general-purpose timer is based on a 16-bit prescaler and a 54-bit auto-reload-capable up-down counter.

Feature List

- A 54-bit time-base counter programmable to incrementing or decrementing
- Three clock sources: PLL_F80M_CLK or XTAL_CLK or RC_FAST_CLK
- A 16-bit clock prescaler, from 2 to 65536
- Able to read real-time value of the time-base counter
- Able to halt and resume the time-base counter
- Programmable alarm generation
- Timer value reload —Auto-reload at alarm or software-controlled instant reload
- Calculate clock frequency —Calculate the measured frequency of the clock based on the crystal clock
- Level interrupt generation
- Support several ETM tasks and events

4.1.4.9 Watchdog Timers (WDT)

ESP32-P4 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-P4 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-P4. 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.

4.1.4.11 Permission Control (PMS)

ESP32-P4 integrates an APM module to manage access permissions.

Feature List

- Up to 32 configurable address ranges for each DMA master
- Access permission management for each CPU core to access internal memory, external memory, and peripheral registers
- Support for interrupts
- Support for exception information record

4.1.4.12 System Registers

The System Registers in the ESP32-P4 chip are used to configure various auxiliary chip features.

Feature List

- Control External memory encryption and decryption
- Control HP core/LP core debugging
- Control Bus timeout protection

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

- **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 will trigger an interrupt.
- **Stack pointer (SP) monitoring**: Monitors whether the SP exceeds the specified address space. A bounds violation will trigger an interrupt.
- **Program counter (PC) logging**: Records the PC value. The developer can get the last PC value at the most recent reset of HP CPU0 or HP CPU1.
- **Bus access logging**: Records the information about bus access. When the HP CPU0, HP CPU1, or the Direct Memory Access controller (DMA) writes a specified value, the Debug Assistant module will record the data type, address of this write operation, and additionally the PC value when the write is performed by HP CPU0 or HP CPU1, and push such information to the HP L2MEM.

4.1.4.14 LP Mailbox

ESP32-P4 integrates an LP Mailbox module which provides an efficient inter-core communication mechanism between the LP CPU and HP CPU0/1. The LP Mailbox module comprises of sixteen 32-bit message registers that the LP CPU and HP CPU0/1 can use to store and exchange message. Inter-core communication between LP CPU and HP CPU0/1 is achieved through an interrupt mechanism implemented within 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-P4 monitors the voltage levels of pins VDD_ANA and VDD_BAT. If the voltage on these pins drops below the predefined threshold (defaulting to 2.7 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 pins VDD_ANA and VDD_BAT
- 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
- Configurable handling modes for under-voltage events

4.1.5 Cryptography/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-P4 integrates an Advanced Encryption Standard (AES) accelerator, which is a hardware device that speeds up computation using AES algorithm significantly, compared to AES algorithms implemented solely in software. The AES accelerator integrated in ESP32-P4 has two working modes, which are Typical AES and DMA-AES.

Feature List

- Typical AES working mode
  - AES-128/AES-256 encryption and decryption
- DMA-AES working mode
  - AES-128/AES-256 encryption and decryption
  - Block cipher mode
    - 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

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-P4'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

- 2 different elliptic curves, namely P-192 and P-256 defined in FIPS 186-3
- 11 working modes
- Interrupt upon completion of calculation

4.1.5.3 HMAC Accelerator (HMAC)

The Hash-based Message Authentication Code (HMAC) module computes Message Authentication Codes (MACs) using hash algorithm SHA-256 and keys as described in RFC 2104. The 256-bit HMAC key is stored in an eFuse key block and can be set as read-protected, i.e., the key is not accessible from outside the HMAC accelerator.

Feature List

- Standard HMAC-SHA-256 algorithm
- HMAC-SHA-256 calculation based on key in eFuse,
  - whose result cannot be accessed by software in downstream mode for high security
  - whose result can be accessed by software in upstream mode
- Generates required keys for the Digital Signature Algorithm (DSA) peripheral in downstream mode
- Re-enables 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-P4 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-P4 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*
- 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 Digital Signature Algorithm (DSA)

The Digital Signature Algorithm (DSA) is used to verify the authenticity and integrity of a message using a cryptographic algorithm. This can be used to validate a device's identity to a server or to check the integrity of a message.

ESP32-P4 includes a Digital Signature Algorithm (DSA) module providing hardware acceleration of messages' signatures based on RSA. HMAC is used as the key derivation function (KDF) to output the DSA_KEY key using a key stored in eFuse as the input key. Subsequently, the DSA module uses DSA_KEY to decrypt the pre-encrypted parameters and calculate the signature. The whole process happens in hardware so that all the keys involved during the calculating process cannot be seen by users, guaranteeing the security of the operation.

Feature List

- RSA digital signatures with key length up to 4096 bits
- Encrypted private key data, only decryptable by the DSA module

- SHA-256 digest to protect private key data against tampering by an attacker

4.1.5.7 Elliptic Curve Digital Signature Algorithm (ECDSA)

In cryptography, the Elliptic Curve Digital Signature Algorithm (ECDSA) offers a variant of the Digital Signature Algorithm (DSA) which uses elliptic-curve cryptography.

ESP32-P4's ECDSA accelerator provides a secure and efficient environment for computing ECDSA signatures. It enables high-speed cryptographic operations while preserving the confidentiality of the signing process, effectively minimizing the risk of information leakage. This makes it particularly valuable for applications that demand both strong security and fast performance. With the ECDSA accelerator, users can trust that their data is well protected—without compromising on speed.

Feature List

- Digital signature verification
- Two different elliptic curves, namely P-192 and P-256, defined in FIPS 186-3 Spec
- Two hash algorithms for message hash in the ECDSA operation, namely SHA-224 and SHA-256, defined in FIPS PUB 180-4 Spec
- Dynamic access permission in different operation statuses to ensure information security

4.1.5.8 External Memory Encryption and Decryption (XTS_AES)

The ESP32-P4 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 and decryption without software's participation
- Encryption and decryption functions jointly enabled/disabled by register configuration, eFuse parameters, and boot mode
- Configurable Anti-DPA

4.1.5.9 Random Number Generator (RNG)

The ESP32-P4 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-P4 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.

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 Processing

This subsection describes the peripherals for image and voice processing.

4.2.1.1 JPEG Codec

ESP32-P4'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, YUV422 and GRAY as original input image formats
- Conversion of RGB888 and RGB565 into YUV444, YUV422 or YUV420 (the only formats supported by impression) for image compression
- 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 1080P@40fps,720P@70fps (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 1080P@40fps, 720P@70fps (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 Image Signal Processor (ISP)

ESP32-P4 includes an image signal processor (ISP), which is a pipeline composed of various image processing algorithms.

Feature List

- Maximum resolution: 1920 x 1080
- Three input channels: MIPI-CSI, DVP, and AXI-DMAC
- Input formats: RAW8, RAW10, and RAW12
- Output formats: RAW8, RGB888, RGB565, YUV422, and YUV420
- Pipeline features:
  - Bayer filter (BF)
  - Demosaic
  - Color correction matrix (CCM)
  - Gamma correction
  - RGB2YUV
  - Sharpen
  - Contrast/hue/saturation/luminance adjustment (COLOR)
  - YUV_limit
  - YUV2RGB
  - Automatic exposure statistics (AE)
  - Automatic focus statistics (AF)
  - Automatic white balance statistics (AWB)
  - Histogram statistics (HIST)

Pin Assignment

For the CAM interface of the image signal processor, the pins used can be chosen from any GPIOs via the GPIO Matrix.

4.2.1.3 Pixel-Processing Accelerator (PPA)

ESP32-P4 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, YUV420
  - Output formats: ARGB8888, RGB888, RGB565, YUV420
  - 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:
  - Input formats: ARGB8888, RGB888, RGB565, L4, L8, A4, A8
  - Output formats: ARGB8888, RGB888, RGB565
  - 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.

4.2.1.4 LCD and Camera Controller (LCD_CAM)

The LCD and Camera controller (LCD_CAM) on the ESP32-P4, 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.

4.2.1.5 H264 Encoder

ESP32-P4 contains a baseline H264 encoder, which is used for real-time video sequence compression, significantly reducing the total amount of data while minimizing video quality loss.

Feature List

- YUV420 progressive video with the maximum encoding performance of 1080p@30fps
- I-frame and P-frame
- GOP mode and dual-stream mode (in dual-stream mode, the total bandwidth of the two video image sequences to be encoded should not exceed 1080p@30fps)
- Intra luma macroblock of 4 x 4 and 16 x 16 partitioning
- All nine prediction modes for 4 x 4 partitioning and all four prediction modes for 16 x 16 partitioning of intra luma macroblock
- All four prediction modes for intra chroma macroblock
- All partition modes of inter prediction macroblock: 4 x 4, 4 x 8, 8 x 4, 8 x 8, 8 x 16, 16 x 8, and 16 x 16
- Motion estimation with the precision of 1/2 and 1/4 pixel
- Search range of inter prediction horizontal motion being [–29.75, +16.75], vertical search range being [–13.75, +13.75]
- Enabling and disabling the deblocking filter
- Context adaptive variable length coding (CAVLC)
- P-skip macroblock
- P slice supporting I macroblock
- Decimate operation of luma and chroma component quantization results
- Fixed QP and rate control at the macroblock level

- MV merge for outputting the MV of each macroblock to memory
- Region of interest (ROI). It can configure up to eight rectangular ROI areas at any position. These ROI areas have fixed priorities and can overlap with each other. Each ROI area can be assigned a fixed QP or QP offset, and a non-ROI area can be specified with a QP offset.

Pin Assignment

The H264 encoder does not directly interact with IOs, so it has no pins assigned.

4.2.1.6 MIPI CSI

ESP32-P4 includes one MIPI CSI interface for connecting cameras of the MIPI interface.

Feature List

- Compliant with MIPI CSI-2
- Compliant with DPHY v1.1
- 2-lane x 1.5 Gbps
- Input formats: RGB888, RGB666, RGB565, YUV422, YUV420, RAW8, RAW10, and RAW12

Pin Assignment

The MIPI CSI interface uses the dedicated digital pins 42–48.

4.2.1.7 MIPI DSI

ESP32-P4 features a MIPI DSI interface for connecting displays of the MIPI interface.

Feature List

- Compliant with MIPI DSI
- Compliant with DPHY v1.1
- 2-lane x 1.5 Gbps
- Input formats: RGB888, RGB666, RGB565, and YUV422
- Output formats: RGB888, RGB666, and RGB565
- Using the video mode to output video stream
- Outputting image patterns

Pin Assignment

The MIPI DSI interface uses the dedicated digital pins 34–40.

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)

ESP32-P4 has six UART controllers, including five UARTs in the HP system and one low-power (LP) UART.

Feature List

Table 4-1. UART and LP UART Feautre Comparison

UART Feature	LP UART Feature
Programmable baud rate up to 5 MBaud	
260 x 8-bit RAM, shared by TX FIFOs and RX FIFOs of the UART controllers	20 x 8-bit RAM, shared by the TX FIFO and RX FIFO of LP UART
Full-duplex asynchronous communication	
Data bits (5 to 8 bits)	
Stop bits (1, 1.5, or 2 bits)	
Parity bit	
Special character AT_CMD detection	
RS485 protocol	—
IrDA protocol	—
High-speed data communication using GDMA	—
Receive timeout	
UART as wakeup source	
Software and hardware flow control	
Three prescalable clock sources: 1. XTAL_CLK 2. RC_FAST_CLK 3. PLL_F80M_CLK	Three prescalable clock sources 1. RC_FAST_CLK 2. XTAL_DIV_CLK 3. PLL_F8M_CLK

Pin Assignment

For UART0–UART4 interfaces, the pins used can be chosen from any GPIOs via the GPIO Matrix. By default, the pins connected to transmit and receive signals (UART0_TXD_PAD and UART0_RXD_PAD) of UART0 are multiplexed with GPIO37–GPIO38 and the eight-line interface of SPI2 controller via IO MUX.

For LP UART, the pins used can be chosen from any LP GPIOs via the LP GPIO Matrix. By default, the pins connected to transmit and receive signals (LP_UART_TXD_PAD and LP_UART_RXD_PAD) are multiplexed with LP_GPIO14–LP_GPIO15 via LP IO MUX.

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-P4 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 with GPIO6–GPIO11 via the IO MUX. The pins connected to the eight-line interface are multiplexed with GPIO28–GPIO38, UART0 interface, and the first RMII interface of EMAC controller via the IO MUX. 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-P4 has three I2C controllers: two in the main system and one in the low-power system. The two I2C controllers in the main system can act as a master or a slave (referred to as I2C below), while the one in the low-power system can only act as a master (referred to as LP_I2C below), which can still work when the main system sleeps.

Feature List

The I2C controller of ESP32-P4 has the following features:

- Master mode and slave mode
- Communication between multiple masters and slaves
- Standard mode (100 Kbit/s)
- Fast mode (400 Kbit/s)
- 7-bit addressing and 10-bit addressing
- Continuous data transfer achieved by pulling SCL low in slave mode
- Programmable digital noise filtering
- Dual address mode, which uses slave address and slave memory or register address

Pin Assignment

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

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

4.2.2.4 Analog I2C Controller

This module is a dedicated I2C host that communicates with some analog modules to configure parameters of these modules. Each configurable module has an I2C slave with its own address.

Feature List

- Master mode only
- 7-bit addressing
- Adjustable transmission rate
- Communication in the sleep modes supported by the Low-Power CPU
- Dual master operation mode

Pin Assignment

The analog I2C interface connects internal analog components without requiring allocating IO pins.

4.2.2.5 I3C Controller

ESP32-P4 includes one I3C master interface.

Feature List

The I3C master interface supports the following features:

- Compliant with I3C protocol
- Compatible with I2C mode (FM, FM+)
- SDR mode
- Dynamic address allocation
- In-Band interrupts
- DMA transfer

Pin Assignment

For I3C master interface, the pins for clock and data signals are multiplexed with GPIO32–GPIO33 via the GPIO matrix. Other signals can be routed to any GPIOs via the GPIO matrix.

4.2.2.6 I2S Controller (I2S)

ESP32-P4 has three built-in I2S interfaces, which provide flexible communication interfaces for streaming digital data in multimedia applications, especially 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 supported:
  - 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 (for I2S0 only), up to two channels supported

  - PDM RX mode
    * Raw PDM data reception
    * PDM-to-PCM data format conversion (for I2S0 only), up to eight channels supported
- Configurable APLL clock with frequencies up to 240 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 (GDMA-AHB only)
- Standard I2S interface interrupts

Pin Assignment

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

4.2.2.7 LP I2S Controller

ESP32-P4 has a built-in LP I2S interface, which provides a data reception communication interface for Voice Activity Detection (VAD) and some digital audio applications in low power mode.

Feature List

- RX master mode and slave mode
- A variety of audio standards supported:
  - TDM Philips standard
  - TDM MSB alignment standard
  - TDM PCM standard
  - PDM standard
- Various RX modes supported:
  - TDM RX mode, up to two channels supported
  - PDM RX mode
    * Raw PDM data reception
    * PDM-to-PCM data format conversion, up to two channels supported
- Configurable sample clock with a variety of sampling frequencies supported
- 16-bit data communication
- Standard LP I2S interface interrupts

Pin Assignment

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

4.2.2.8 Pulse Count Controller (PCNT)

The pulse count controller (PCNT) is designed to count input pulses.

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 with their corresponding control signals
- Independently filter glitches of input pulse signals and control signals on each unit
- Each channel has the following parameters:
  1. Selection between counting on rising or falling edges of the input pulse signal
  2. Configuration to Increment, Decrement, or Disable counter mode for control signal's high and low states
- 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.

4.2.2.9 USB 2.0 High-Speed OTG

The ESP32-P4 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.
- 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 pin49 and pin50. Other signals can be routed to any GPIOs via the GPIO matrix.

4.2.2.10 USB 2.0 Full-Speed OTG

The ESP32-P4 features a USB 2.0 Full-Speed On-The-Go peripheral (henceforth referred to as OTG_FS) along with integrated transceivers. This OTG_FS conforms to USB 2.0 specification, OTG Revision 1.3, and OTG Revision 2.0 specifications, OTG_FS can operate as either a USB Host or Device and supports 12 Mbit/s full-speed (FS) and 1.5 Mbit/s low-speed (LS) data rates of the USB 2.0 specification. The Host Negotiation Protocol (HNP) and the Session Request Protocol (SRP) are also supported.

Feature List

General Features

- USB 2.0 specification, OTG Revision 1.3 and OTG Revision 2.0 specifications
- USB 2.0 full-speed and low-speed data rates
- HNP and SRP as A-device or B-device

- Dynamic FIFO (DFIFO) sizing, maximum to 1 KB
- Multiple modes of memory access
  - Scatter/Gather DMA mode
  - Buffer DMA mode
  - Slave mode
- Two integrated transceivers

Device Mode Features

- Endpoint 0 always present, bi-directional, consisting of EP0 IN and EP0 OUT
- Six additional endpoints 1–6, configurable as IN or OUT
- Maximum of five 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

- Eight 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 1 KB RAM.
- The size of each FIFO is configurable, with a maximum of 1 KB.

Pin Assignment

The pins connected to D+ and D- signals for two pairs of USB PHY are multiplexed with GPIO24–GPIO25 and GPIO26–GPIO27. The USB 2.0 Full-Speed OTG interface can use each of them. By default, the pins are multiplexed with GPIO26–GPIO27. In addition, the functionalities of USB_D- and USB_D+ can be exchanged.

Other signals can be routed to any GPIOs via the GPIO matrix.

4.2.2.11 USB Serial/JTAG Controller (USB_SERIAL_JTAG)

ESP32-P4 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.

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
- Two integrated full-speed transceivers
- Choosing from two full-speed integrated transceivers GPIO24/GPIO25 and GPIO26/GPIO27
- Supporting USB 2.0 OTG using one of the integrated transceivers while USB Serial/JTAG using the other one

Pin Assignment

The pins connected to D+ and D- signals for two pairs of USB PHY are multiplexed with GPIO24–GPIO25 and GPIO26–GPIO27. The USB Serial/JTAG Controller interface can use each of them. By default, the pins are multiplexed with GPIO24–GPIO25.

4.2.2.12 Ethernet Media Access Controller (EMAC)

By using the external Ethernet PHY (physical layer), ESP32-P4 can send and receive data via Ethernet MAC (Media Access Controller) according to the IEEE 802.3 standard.

ESP32-P4 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) and Reduced Media Independent Interface (RMII)
- IEEE 802.3az-2010 for Energy Efficient Ethernet
- IEEE 802.1Q for VLAN frame format

Feature List

- Data rates of 10/100 Mbit/s through an external PHY interface
- Communication with an external Fast Ethernet PHY through IEEE 802.3-compliant MII and RMII interfaces
- 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 eight 48-bit perfect address filters with per-byte masking
  - Up to eight 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: 256-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

The Ethernet media access controller includes three RMII interfaces. The pins connected to the first RMII interface are multiplexed with GPIO28–GPIO36 and the SPI2 interface via IO MUX. The pins connected to the second RMII interface are multiplexed with GPIO40–GPIO48 via IO MUX. The third RMII interface does not include transmit signals except for the transmit enable signal (RMII_TXEN). The pins connected to the third

interface are multiplexed with GPIO49–GPIO54 via IO MUX. Three sets of signals can be used in any combination.

The pins for the MII interface, MDIO interface, and other interfaces can be chosen from any GPIOs via the GPIO Matrix.

4.2.2.13 Two-Wire Automotive Interface (TWAI)

ESP32-P4 contains three TWAI controllers. Each controller can individually be connected to a TWAI bus via an external transceiver.

Feature List

- Compatibility with ISO 11898-1 protocol (CAN Specification 2.0)
- Standard Frame Format (11-bit ID) and Extended Frame Format (29-bit ID)
- Bit rates from 1 Kbit/s to 1 Mbit/s
- Multiple modes of operation:
  - Normal
  - Listen-only (no influence on bus)
  - Self-test (no acknowledgment required during data transmission)
- 64-byte Receive FIFO
- Special transmissions:
  - Single-shot transmissions (does not automatically re-transmit upon error)
  - Self-reception (the TWAI controller transmits and receives messages simultaneously)
- Acceptance Filter (supports Single and Dual-filter modes)
- Error detection and handling:
  - Error counters
  - Configurable error warning limit
  - Error code capture
  - Arbitration lost capture
  - Automatic transceiver standby

Pin Assignment

The pins for the two-wire automotive interface can be chosen from any GPIOs via the GPIO Matrix.

4.2.2.14 SD/MMC Host Controller (SDHOST)

ESP32-P4 has an SD/MMC Host Controller.

Feature List

- Two external cards
- SD memory Card specification v3.0 and v3.01
- MMC: v4.41, v4.5, and v4.51
- CE-ATA: v1.1
- 1-bit, 4-bit, and 8-bit modes

Pin Assignment

For the SD/MMC host controller, the pins connected to clock, command, and data signals of the SDIO3.0 interface are multiplexed with GPIO39–GPIO48, the second RMII interface of EMAC, and the output signal of 50 MHz clock via IO MUX. Other signals can be routed to any GPIOs via the GPIO matrix.

For the SDIO2.0 interface, the pins can be chosen from any GPIOs via the GPIO Matrix.

4.2.2.15 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

- Eight independent PWM generators (i.e., eight channels)
- Maximum PWM duty cycle resolution: 20 bits
- Four independent timers that support fractional division
- 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 in terms of fading direction (increase or decrease), fading amount (the amount by which the duty cycle increases or decreases each time), the number of fades (how many times the duty cycle fades in one range), and fading 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.

4.2.2.16 Motor Control PWM (MCPWM)

ESP32-P4 integrates two MCPWMs that can be used to drive digital motors and smart light. Every MCPWM has a clock divider (prescaler), three PWM timers, three PWM operators, a dedicated capture submodule, an Event Task Matrix (ETM) module, and an fault detection module.

Feature List

PWM timers are used to generate timing references. The PWM operators generate desired waveform based on the timing references. By configuration, a PWM operator can use the timing reference of any PWM timer, and use the same timing reference with other PWM operators. PWM operators can also 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.

4.2.2.17 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.

4.2.2.18 Parallel IO Controller (PARLIO)

ESP32-P4 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.

4.2.2.19 BitScrambler

The ESP32-P4 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-P4 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

ESP32-P4 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 pins of the touch sensor are multiplexed with GPIO2–GPIO15, LP_GPIO2–LP_GPIO15, LP_UART interface, and one four-line interface of SPI2. When the pins are configured for the analog function, the multiplexed digital functions are disabled.

4.2.3.2 Temperature Sensor (TSENS)

ESP32-P4 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, supporting two wake-up modes
- 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

Pin Assignment

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

4.2.3.3 ADC Controller (ADC)

ESP32-P4 integrates two 12-bit successive approximation ADCs (SAR ADCs) for measuring analog signals from up to 14 pins.

Feature List

- HP ADC and LP ADC controllers can control the SAR ADC via software
- 12-bit resolution
- Analog input sampling from up to 14 pins
- HP ADC controllers:
  - Multi-channel sampling control module with configurable channel sampling sequence
  - Mode control module supporting dual HP ADC sampling
  - Two filters with configurable filter coefficients
  - Two threshold monitors that trigger an interrupt when filtered data exceeds a high threshold or falls below a low threshold
  - Continuous transfer of conversion results to memory via the GDMA interface
- LP ADC controllers:
  - One-shot sampling mode
  - Sampling in sleep mode (e.g., Deep-sleep)
- Event Task Matrix (ETM) support for various events and tasks

Pin Assignment

The pins of the ADC controller are multiplexed with GPIO16–GPIO23, GPIO49–GPIO54, the interfaces of two analog voltage comparators, and the third RMII interface of EMAC.

4.2.3.4 Analog Voltage Comparator

ESP32-P4 integrates two analog voltage comparators. These comparators rely on special pads that support voltage comparison functionality to monitor voltage changes on these pads.

Feature List

- Voltage comparison
  - Configurable voltage comparison mode
  - Configurable reference voltage
- Interrupt upon changes of voltage comparison result
- ETM event generation

Pin Assignment

The pins of the analog voltage comparator are multiplexed with GPIO51–GPIO52, GPIO53–GPIO54, the interface of one ADC controller, and the third RMII interface of EMAC.

4.2.3.5 Voice Activity Detection (VAD)

ESP32-P4 integrates a Voice Activity Detection (VAD) module. This module facilitates the hardware implementation of the first-stage algorithm for voice wake-up and other multimedia functions. Additionally, it provides hardware support for low-power voice wake-up solutions.

Feature List

- VAD algorithm processes voice data frame by frame, with each frame containing 256 data points. The data sampling rate is 8 kHz, and the bit width is 16 bits
- 2 KB buffer that stores up to four frames of data
- Independent system wake-up source
- Configurable interrupt sources
- Flexible configuration of algorithm parameters

Pin Assignment

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

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 Operating Conditions 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
VDD_LDO, VDD_DCDCC, VDD_ANA, VDD_BAT, VDD_LP	Allowed input voltage	–0.3	3.6	V
VDD_IO_0, VDD_FLASHIO[3], VDD_IO_4, VDD_IO_5, VDD_IO_6	Allowed input voltage	1.62/–0.3	1.98/3.6	V
VDD_PSRAM_0, VDD_PSRAM_1	Allowed input voltage	1.62	1.98	V
VDD_HP_0, VDD_HP_2, VDD_HP_3	Allowed input voltage	0	1.3	V
VDD_MIPI_DPHY	Allowed input voltage	0	2.75	V
VDD_USBPHY	Allowed input voltage	–0.66	3.96	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 2.6.1 *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.

[3] VDD_FLASHIO provides power for flash IO, and the voltage should be adjusted according to the specific flash model.

5.2 Recommended Operating Conditions

Table 5-2. Recommended Operating Conditions

Parameter	Description	Min	Typ	Max	Unit
VDD_LDO, VDD_DCDCC, VDD_ANA, VDD_BAT, VDD_LP	Recommended input voltage	3.0	3.3	3.6	V
VDD_IO_0, VDD_FLASHIO, VDD_IO_4, VDD_IO_5, VDD_IO_6	Recommended input voltage	1.65/3.0	1.8/3.3	1.95/3.6	V
VDD_PSRAM_0, VDD_PSRAM_1	Recommended input voltage	1.65	1.8	1.95	V
VDD_HP_0, VDD_HP_2, VDD_HP_3	Recommended input voltage	0.99	1.1	1.3	V
VDD_MIPI_DPHY	Recommended input voltage	2.25	2.5	2.75	V

Cont'd on next page

Table 5-2 – cont'd from previous page

Parameter	Description	Min	Typ	Max	Unit
VDD_USBPHY	Recommended input voltage	2.97	3.3	3.63	V
I_{VDD}	Current supplied to core	0.5	—	—	A
T_A	Ambient temperature	–40	—	85	°C

[1] The chip can automatically adjust the input voltage of VDD_HP_x based on the situation.

5.3 VDDO_FLASH Output Characteristics

Table 5-3. VDDO_FLASH Internal and Output Characteristics

Parameter	Description	Typ	Unit
R_{VFB}	VDDO_FLASH powered by VDD_LDO via R_{VFB} for 3.3 V flash [1]	3	Ω
I_{VFB}	Output current when VDDO_FLASH is powered by Flash Voltage Regulator for 1.8 V flash	50	mA

[1] See in conjunction with Section 2.6.2 *Power Scheme*.

[1] VDD_LDO must be more than *VDD_flash_min* + *I_flash_max* × R_{VFB};
where

- *VDD_flash_min* – minimum operating voltage of flash
- *I_flash_max* – maximum operating current of flash

5.4 DC Characteristics (3.3 V, 25 °C)

Table 5-4. 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}	Pull-up resistor	—	45	—	kΩ
R_{PD}	Pull-down resistor	—	45	—	kΩ
V_{IH_nRST}	Chip reset release voltage (CHIP_PU should satisfy the required voltage)	0.75 × VDD_BAT	—	VDD_BAT + 0.3	V

V_{IL_nRST}	Chip reset voltage (CHIP_PU should satisfy the required voltage)	–0.3	—	0.25 × VDD_BAT	V

[1] VDD is the voltage for power pins VDD_IO_0/4/5/6.

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

5.5 ADC Characteristics

The measurements in this section are taken with an external 100 nF capacitor connected to the ADC, using DC signals as input, and at an ambient temperature of 25 °C.

Table 5-5. ADC Characteristics

Symbol	Min	Max	Unit
DNL (Differential nonlinearity) [1]	–1	3	LSB
INL (Integral nonlinearity)	–5	3	LSB
Sampling rate	—	100	kSPS [2]

[1] To get better DNL results, you can sample multiple times and apply a filter, or calculate the average value.

[2] kSPS means kilo samples-per-second.

The calibrated ADC results after hardware calibration and software calibration are shown in Table 5-6 *ADC Characteristics*. For higher accuracy, you may implement your own calibration methods.

Table 5-6. ADC Calibration Results

Parameter	Description	Min	Max	Unit
Total error	ATTEN0, effective measurement range of 0–1000	–12	12	mV
	ATTEN1, effective measurement range of 0–1300	–12	12	mV
	ATTEN2, effective measurement range of 0–1900	–12	12	mV
	ATTEN3, effective measurement range of 0–3300	–15	15	mV

5.6 Current Consumption in Active and Low-power Modes

Table 5-7. Current Consumption in Active Mode

Work mode	Frequency (MHz)	Description	Typ[1] (mA)	Typ[2] (mA)
Active[3]	360	WAITI (Dual core in idle state)	35	65
		Dual-core while(1) loop operation	80	103
		Single core running CoreMark instructions, the other core in idle state	70	92
		Dual core running 32-bit data access instructions	92	123
	180	WAITI (Dual core in idle state)	32	59
		Dual-core while(1) loop operation	56	77

Cont'd on next page

Table 5-7 – cont'd from previous page

Work mode	Frequency (MHz)	Description	Typ[1] (mA)	Typ[2] (mA)
		Single core running CoreMark instructions, the other core in idle state	51	72
		Dual core running 32-bit data access instructions	65	87
	90	WAITI (Dual core in idle state)	28	44
		Dual-core while(1) loop operation	40	83
		Single core running CoreMark instructions, the other core in idle state	37	51
		Dual core running 32-bit data access instructions	45	61
	40	WAITI (Dual core in idle state)	26	35
		Dual-core while(1) loop operation	31	39
		Single core running CoreMark instructions, the other core in idle state	30	38
		Dual core running 32-bit data access instructions	33	41

[1] Current consumption when all peripheral clocks are **disabled**.

[2] Current consumption when all peripheral clocks are **enabled**. In practice, the current consumption might be different depending on which peripherals are enabled.

[3] In Active mode, the current consumption might be higher when accessing flash/PSRAM.

Table 5-8. Current Consumption in Low-Power Modes

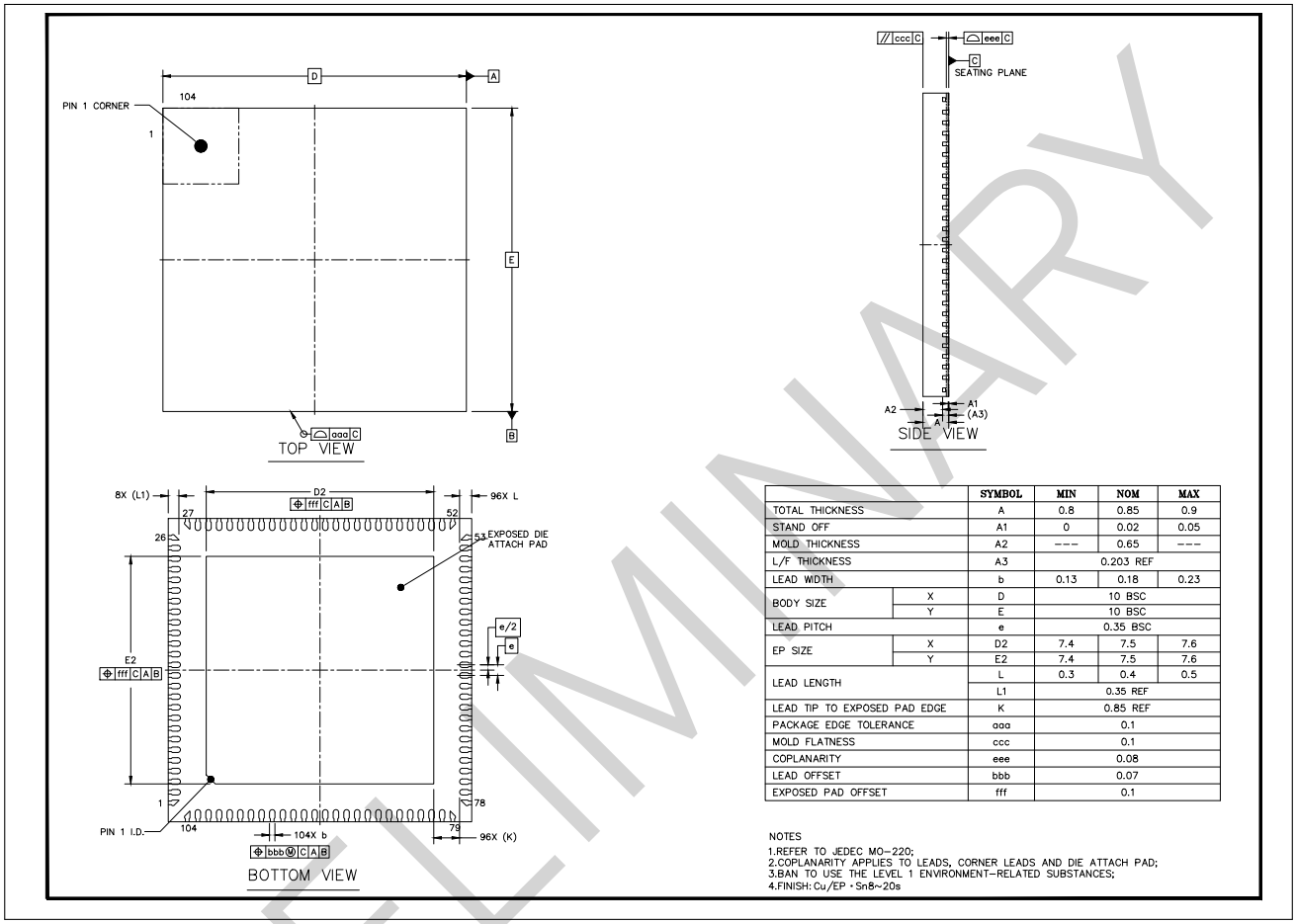
Mode	Description	Typ (mA)[1]
Light-sleep[2]	All GPIOs are high-impedance, and all power supplies are enabled	3.5
	All GPIOs are high-impedance, most of peripherals are disabled, and chip is connected through USB	0.25
	All peripherals are disabled, and data is stored in HP memory	0.2
Deep-sleep	LP timer and LP memory are powered on	0.025
Power off	CHIP_PU is set to low level, the chip is powered off	0.002

[1] The power consumption data was measured with USB 2.0 not working.

[2] The current in Light-sleep mode refers to the current measured when the PSRAM is not powered. In Light-sleep mode, if the PSRAM is powered on, the chip's internal current increases by about 0.2 mA, in addition to the current required for the PSRAM's operating mode.

6 Packaging

- For information about tape, reel, and chip marking, please refer to *ESP32-P4 Chip Packaging Information*.
- 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-P4 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	0.13	0.18	0.23
BODY SIZE	X	D		10 BSC	
	Y	E		10 BSC	
LEAD PITCH		e		0.35 BSC	
EP SIZE	X	D2	7.4	7.5	7.6
	Y	E2	7.4	7.5	7.6
LEAD LENGTH		L	0.3	0.4	0.5
		L1		0.35 REF	
LEAD TIP TO EXPOSED PAD EDGE		K		0.85 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 6-1. QFN104 (10×10 mm) Package

Related Documentation and Resources

Related Documentation

- ESP32-P4 Technical Reference Manual – Detailed information on how to use the ESP32-P4 memory and peripherals.
- ESP32-P4 Hardware Design Guidelines – Guidelines on how to integrate the ESP32-P4 into your hardware product.
- *Certificates*
  https://espressif.com/en/support/documents/certificates
- *ESP32-P4 Product/Process Change Notifications (PCN)*
  https://espressif.com/en/support/documents/pcns?keys=ESP32-P4
- *ESP32-P4 Advisories* – Information on security, bugs, compatibility, component reliability.
  https://espressif.com/en/support/documents/advisories?keys=ESP32-P4
- *Documentation Updates and Update Notification Subscription*
  https://espressif.com/en/support/download/documents

Developer Zone

- ESP-IDF Programming Guide for ESP32-P4 – Extensive documentation for the ESP-IDF development framework.
- *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/
- *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-P4 Series SoCs* – Browse through all ESP32-P4 SoCs.
  https://espressif.com/en/products/socs?id=ESP32-P4
- *ESP32-P4 Series DevKits* – Browse through all ESP32-P4-based devkits.
  https://espressif.com/en/products/devkits?id=ESP32-P4
- *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

Appendix A – ESP32-P4 Consolidated Pin Overview

Pin No.	Pin Name	Pin Type	Pin Providing Power	Pin Settings		HP IO MUX Function								LP IO MUX				Analog Function	
				At Reset	After Reset	F0	Type	F1	Type	F2	Type	F3	Type	F0	Type	F1	Type	F0	F1
1	GPIO1	IO	VDD_LP/VDD_BAT	–	–	GPIO1	I/O/T	GPIO1	I/O/T	–	–	–	–	LP_GPIO1	I/O/T	LP_GPIO1	I/O/T	XTAL_32K_P	–
2	GPIO2	IO	VDD_LP/VDD_BAT	–	IE, WPU	MTCK	I1	GPIO2	I/O/T	–	–	–	–	LP_GPIO2	I/O/T	LP_GPIO2	I/O/T	TOUCH_CHANNEL1	–
3	GPIO3	IO	VDD_LP/VDD_BAT	–	IE	MTDI	I1	GPIO3	I/O/T	–	–	–	–	LP_GPIO3	I/O/T	LP_GPIO3	I/O/T	TOUCH_CHANNEL2	–
4	GPIO4	IO	VDD_LP	–	IE	MTMS	I0	GPIO4	I/O/T	–	–	–	–	LP_GPIO4	I/O/T	LP_GPIO4	I/O/T	TOUCH_CHANNEL3	–
5	GPIO5	IO	VDD_LP	–	–	MTDO	O/T	GPIO5	I/O/T	–	–	–	–	LP_GPIO5	I/O/T	LP_GPIO5	I/O/T	TOUCH_CHANNEL4	–
6	GPIO6	IO	VDD_LP	–	–	GPIO6	I/O/T	GPIO6	I/O/T	–	–	SPI2_HOLD_PAD	I1/O/T	LP_GPIO6	I/O/T	LP_GPIO6	I/O/T	TOUCH_CHANNEL5	–
7	GPIO7	IO	VDD_LP	–	–	GPIO7	I/O/T	GPIO7	I/O/T	–	–	SPI2_CS_PAD	I1/O/T	LP_GPIO7	I/O/T	LP_GPIO7	I/O/T	TOUCH_CHANNEL6	–
8	GPIO8	IO	VDD_LP	–	–	GPIO8	I/O/T	GPIO8	I/O/T	–	–	SPI2_D_PAD	I1/O/T	LP_GPIO8	I/O/T	LP_GPIO8	I/O/T	TOUCH_CHANNEL7	–
9	VDD_LP	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
10	GPIO9	IO	VDD_LP	–	–	GPIO9	I/O/T	GPIO9	I/O/T	–	–	SPI2_CK_PAD	I1/O/T	LP_GPIO9	I/O/T	LP_GPIO9	I/O/T	TOUCH_CHANNEL8	–
11	GPIO10	IO	VDD_LP	–	–	GPIO10	I/O/T	GPIO10	I/O/T	–	–	SPI2_Q_PAD	I1/O/T	LP_GPIO10	I/O/T	LP_GPIO10	I/O/T	TOUCH_CHANNEL9	–
12	GPIO11	IO	VDD_LP	–	–	GPIO11	I/O/T	GPIO11	I/O/T	–	–	SPI2_WP_PAD	I1/O/T	LP_GPIO11	I/O/T	LP_GPIO11	I/O/T	TOUCH_CHANNEL10	–
13	GPIO12	IO	VDD_LP	–	–	GPIO12	I/O/T	GPIO12	I/O/T	–	–	–	–	LP_GPIO12	I/O/T	LP_GPIO12	I/O/T	TOUCH_CHANNEL11	–
14	GPIO13	IO	VDD_LP	–	–	GPIO13	I/O/T	GPIO13	I/O/T	–	–	–	–	LP_GPIO13	I/O/T	LP_GPIO13	I/O/T	TOUCH_CHANNEL12	–
15	GPIO14	IO	VDD_LP	–	–	GPIO14	I/O/T	GPIO14	I/O/T	–	–	–	–	LP_UART_TXD_PAD	O	LP_GPIO14	I/O/T	TOUCH_CHANNEL13	–
16	GPIO15	IO	VDD_LP	–	–	GPIO15	I/O/T	GPIO15	I/O/T	–	–	–	–	LP_UART_RXD_PAD	I1	LP_GPIO15	I/O/T	TOUCH_CHANNEL14	–
17	GPIO16	IO	VDD_IO_0	–	–	GPIO16	I/O/T	GPIO16	I/O/T	–	–	–	–	–	–	–	–	ADC1_CHANNEL0	–
18	GPIO17	IO	VDD_IO_0	–	–	GPIO17	I/O/T	GPIO17	I/O/T	–	–	–	–	–	–	–	–	ADC1_CHANNEL1	–
19	GPIO18	IO	VDD_IO_0	–	–	GPIO18	I/O/T	GPIO18	I/O/T	–	–	–	–	–	–	–	–	ADC1_CHANNEL2	–
20	GPIO19	IO	VDD_IO_0	–	–	GPIO19	I/O/T	GPIO19	I/O/T	–	–	–	–	–	–	–	–	ADC1_CHANNEL3	–
21	VDD_IO_0	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
22	GPIO20	IO	VDD_IO_0	–	–	GPIO20	I/O/T	GPIO20	I/O/T	–	–	–	–	–	–	–	–	ADC1_CHANNEL4	–
23	GPIO21	IO	VDD_IO_0	–	–	GPIO21	I/O/T	GPIO21	I/O/T	–	–	–	–	–	–	–	–	ADC1_CHANNEL5	–
24	GPIO22	IO	VDD_IO_0	–	–	GPIO22	I/O/T	GPIO22	I/O/T	–	–	–	–	–	–	–	–	ADC1_CHANNEL6	–
25	GPIO23	IO	VDD_IO_0	–	–	GPIO23	I/O/T	GPIO23	I/O/T	–	–	REF_50M_CLK_PAD	O	–	–	–	–	ADC1_CHANNEL7	–
26	VDD_HP_0	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
27	FLASH_CS	Dedicated	VDD_FLASHIO	–	–	FLASH_CS	O	–	–	–	–	–	–	–	–	–	–	–	–
28	FLASH_Q	Dedicated	VDD_FLASHIO	–	–	FLASH_Q	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
29	FLASH_WP	Dedicated	VDD_FLASHIO	–	–	FLASH_WP	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
30	VDD_FLASHIO	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
31	FLASH_HOLD	Dedicated	VDD_FLASHIO	–	–	FLASH_HOLD	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
32	FLASH_CK	Dedicated	VDD_FLASHIO	–	–	FLASH_CK	O	–	–	–	–	–	–	–	–	–	–	–	–
33	FLASH_D	Dedicated	VDD_FLASHIO	–	–	FLASH_D	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
34	DSI_REXT	Dedicated	VDD_MIPI_DPHY	–	–	MIPI DSI PHY 4.02 KΩ EXTERNAL RESISTOR	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
35	DSI_DATAP1	Dedicated	VDD_MIPI_DPHY	–	–	MIPI DSI PHY DATAP1	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
36	DSI_DATAN1	Dedicated	VDD_MIPI_DPHY	–	–	MIPI DSI PHY DATAN1	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
37	DSI_CLKN	Dedicated	VDD_MIPI_DPHY	–	–	MIPI DSI PHY CLKN	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
38	DSI_CLKP	Dedicated	VDD_MIPI_DPHY	–	–	MIPI DSI PHY CLKP	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
39	DSI_DATAP0	Dedicated	VDD_MIPI_DPHY	–	–	MIPI DSI PHY DATAP0	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
40	DSI_DATAN0	Dedicated	VDD_MIPI_DPHY	–	–	MIPI DSI PHY DATAN0	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
41	VDD_MIPI_DPHY	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
42	CSI_DATAN0	Dedicated	VDD_MIPI_DPHY	–	–	MIPI CSI PHY DATAN0	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
43	CSI_DATAP0	Dedicated	VDD_MIPI_DPHY	–	–	MIPI CSI PHY DATAP0	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
44	CSI_CLKP	Dedicated	VDD_MIPI_DPHY	–	–	MIPI CSI PHY CLKP	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
45	CSI_CLKN	Dedicated	VDD_MIPI_DPHY	–	–	MIPI CSI PHY CLKN	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–

Cont'd on next page

Table 6-1 – Cont'd from previous page

Pin No.	Pin Name	Pin Type	Pin Providing Power	Pin Settings		HP IO MUX Function								LP IO MUX				Analog Function	
				At Reset	After Reset	F0	Type	F1	Type	F2	Type	F3	Type	F0	Type	F1	Type	F0	F1
46	CSI_DATAN1	Dedicated	VDD_MIPI_DPHY	–	–	MIPI CSI PHY DATAN1	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
47	CSI_DATAP1	Dedicated	VDD_MIPI_DPHY	–	–	MIPI CSI PHY DATAP1	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
48	CSI_REXT	Dedicated	VDD_MIPI_DPHY	–	–	MIPI CSI PHY 4.02 KΩ EXTERNAL RESISTOR	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
49	DM	Dedicated	VDD_USBPHY	–	–	USB2 OTG PHY DM	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
50	DP	Dedicated	VDD_USBPHY	–	–	USB2 OTG PHY DP	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
51	VDD_USBPHY	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
52	GPIO24	IO	VDD_IO_4	–	–	GPIO24	I/O/T	GPIO24	I/O/T	–	–	–	–	–	–	–	–	USB1P1_N0	–
53	GPIO25	IO	VDD_IO_4	–	IE, USB_WPU	GPIO25	I/O/T	GPIO25	I/O/T	–	–	–	–	–	–	–	–	USB1P1_P0	–
54	NC	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
55	GPIO26	IO	VDD_IO_4	–	–	**GPIO26**	I/O/T	GPIO26	I/O/T	–	–	–	–	–	–	–	–	USB1P1_N1	–
56	GPIO27	IO	VDD_IO_4	–	–	**GPIO27**	I/O/T	GPIO27	I/O/T	–	–	–	–	–	–	–	–	USB1P1_P1	–
57	GPIO28	IO	VDD_IO_4	–	–	**GPIO28**	I/O/T	GPIO28	I/O/T	SPI2_CS_PAD	I1/O/T	GMAC_PHY_RXDV_PAD	IO	–	–	–	–	–	–
58	GPIO29	IO	VDD_IO_4	–	–	**GPIO29**	I/O/T	GPIO29	I/O/T	SPI2_D_PAD	I1/O/T	GMAC_PHY_RXD0_PAD	IO	–	–	–	–	–	–
59	VDD_PSRAM_0	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
60	GPIO30	IO	VDD_IO_4	–	–	**GPIO30**	I/O/T	GPIO30	I/O/T	SPI2_CK_PAD	I1/O/T	GMAC_PHY_RXD1_PAD	IO	–	–	–	–	–	–
61	GPIO31	IO	VDD_IO_4	–	–	**GPIO31**	I/O/T	GPIO31	I/O/T	SPI2_Q_PAD	I1/O/T	GMAC_PHY_RXER_PAD	IO	–	–	–	–	–	–
62	VDD_IO_4	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
63	GPIO32	IO	VDD_IO_4	IE	–	**GPIO32**	I/O/T	GPIO32	I/O/T	SPI2_HOLD_PAD	I1/O/T	GMAC_RMII_CLK_PAD	IO	–	–	–	–	–	–
64	GPIO33	IO	VDD_IO_4	IE	–	**GPIO33**	I/O/T	GPIO33	I/O/T	SPI2_WP_PAD	I1/O/T	GMAC_PHY_TXEN_PAD	O	–	–	–	–	–	–
65	GPIO34	IO	VDD_IO_4	IE	–	**GPIO34**	I/O/T	GPIO34	I/O/T	SPI2_IO4_PAD	I1/O/T	GMAC_PHY_TXD0_PAD	O	–	–	–	–	–	–
66	GPIO35	IO	VDD_IO_4	IE, WPU	–	**GPIO35**	I/O/T	GPIO35	I/O/T	SPI2_IO5_PAD	I1/O/T	GMAC_PHY_TXD1_PAD	O	–	–	–	–	–	–
67	VDD_PSRAM_1	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
68	GPIO36	IO	VDD_IO_4	IE, WPU	–	**GPIO36**	I/O/T	GPIO36	I/O/T	SPI2_IO6_PAD	I1/O/T	GMAC_PHY_TXER_PAD	O	–	–	–	–	–	–
69	GPIO37	IO	VDD_IO_4	IE	IE	**UART0_TXD_PAD**	O	GPIO37	I/O/T	SPI2_IO7_PAD	I1/O/T	–	–	–	–	–	–	–	–
70	GPIO38	IO	VDD_IO_4	IE	–	**UART0_RXD_PAD**	I1	GPIO38	I/O/T	SPI2_DQS_PAD	O/T	–	–	–	–	–	–	–	–
71	VDDO_FLASH	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
72	VDDO_PSRAM	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
73	VDDO_3	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
74	VDDO_4	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
75	VDD_LDO	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
76	VDD_HP_2	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
77	VDD_DCDCC	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
78	FB_DCDC	Analog	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
79	EN_DCDC	Analog	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
80	GPIO39	IO	VDD_IO_5	–	–	SD1_CDATA0_PAD	I1/O/T	**GPIO39**	I/O/T	–	–	REF_50M_CLK_PAD	O	–	–	–	–	–	–
81	GPIO40	IO	VDD_IO_5	–	–	SD1_CDATA1_PAD	I1/O/T	**GPIO40**	I/O/T	–	–	GMAC_PHY_TXEN_PAD	O	–	–	–	–	–	–
82	GPIO41	IO	VDD_IO_5	–	–	SD1_CDATA2_PAD	I1/O/T	**GPIO41**	I/O/T	–	–	GMAC_PHY_TXD0_PAD	O	–	–	–	–	–	–
83	GPIO42	IO	VDD_IO_5	–	–	SD1_CDATA3_PAD	I1/O/T	**GPIO42**	I/O/T	–	–	GMAC_PHY_TXD1_PAD	O	–	–	–	–	–	–
84	GPIO43	IO	VDD_IO_5	–	–	SD1_CCLK_PAD	O	**GPIO43**	I/O/T	–	–	GMAC_PHY_TXER_PAD	O	–	–	–	–	–	–
85	VDD_IO_5	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
86	GPIO44	IO	VDD_IO_5	–	–	SD1_CCMD_PAD	I1/O/T	**GPIO44**	I/O/T	–	–	GMAC_RMII_CLK_PAD	IO	–	–	–	–	–	–
87	GPIO45	IO	VDD_IO_5	–	–	SD1_CDATA4_PAD	I1/O/T	**GPIO45**	I/O/T	–	–	GMAC_PHY_RXDV_PAD	IO	–	–	–	–	–	–
88	GPIO46	IO	VDD_IO_5	–	–	SD1_CDATA5_PAD	I1/O/T	**GPIO46**	I/O/T	–	–	GMAC_PHY_RXD0_PAD	IO	–	–	–	–	–	–
89	GPIO47	IO	VDD_IO_5	–	–	SD1_CDATA6_PAD	I1/O/T	**GPIO47**	I/O/T	–	–	GMAC_PHY_RXD1_PAD	IO	–	–	–	–	–	–
90	GPIO48	IO	VDD_IO_5	–	–	SD1_CDATA7_PAD	I1/O/T	**GPIO48**	I/O/T	–	–	GMAC_PHY_RXER_PAD	IO	–	–	–	–	–	–
91	VDD_HP_3	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
92	GPIO49	IO	VDD_IO_6	–	–	**GPIO49**	I/O/T	GPIO49	I/O/T	–	–	GMAC_PHY_TXEN_PAD	O	–	–	–	–	ADC2_CHANNEL0	–
93	GPIO50	IO	VDD_IO_6	–	–	**GPIO50**	I/O/T	GPIO50	I/O/T	–	–	GMAC_RMII_CLK_PAD	IO	–	–	–	–	ADC2_CHANNEL1	–
94	GPIO51	IO	VDD_IO_6	–	–	**GPIO51**	I/O/T	GPIO51	I/O/T	–	–	GMAC_PHY_RXDV_PAD	IO	–	–	–	–	ADC2_CHANNEL2	ANA_COMPO
95	GPIO52	IO	VDD_IO_6	–	–	**GPIO52**	I/O/T	GPIO52	I/O/T	–	–	GMAC_PHY_RXD0_PAD	IO	–	–	–	–	ADC2_CHANNEL3	ANA_COMPO
96	VDD_IO_6	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
97	GPIO53	IO	VDD_IO_6	–	–	**GPIO53**	I/O/T	GPIO53	I/O/T	–	–	GMAC_PHY_RXD1_PAD	IO	–	–	–	–	ADC2_CHANNEL4	ANA_COMP1
98	GPIO54	IO	VDD_IO_6	–	–	**GPIO54**	I/O/T	GPIO54	I/O/T	–	–	GMAC_PHY_RXER_PAD	IO	–	–	–	–	ADC2_CHANNEL5	ANA_COMP1
99	XTAL_N	Analog	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
100	XTAL_P	Analog	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Cont'd on next page

Table 6-1 – Cont'd from previous page

Pin No.	Pin Name	Pin Type	Pin Providing Power	Pin Settings		HP IO MUX Function								LP IO MUX				Analog Function	
				At Reset	After Reset	F0	Type	F1	Type	F2	Type	F3	Type	F0	Type	F1	Type	F0	F1
101	VDD_ANA	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
102	VDD_BAT	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
103	CHIP_PU	Analog	VDD_ANA	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
104	GPIO0	IO	VDD_LP/VDD_BAT	–	–	GPIO0	I/O/T	GPIO0	I/O/T	–	–	–	–	LP_GPIO0	I/O/T	LP_GPIO0	I/O/T	XTAL_32K_N	–
105	GND	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

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

Revision History

Date	Version	Release notes
2025-06-03	v0.5	Preliminary release

Disclaimer and Copyright Notice

www.espressif.com