ESP32-P4 Series

Datasheet

High-performance MCU with one RISC-V 32-bit dual-core microprocessor and one single-core 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

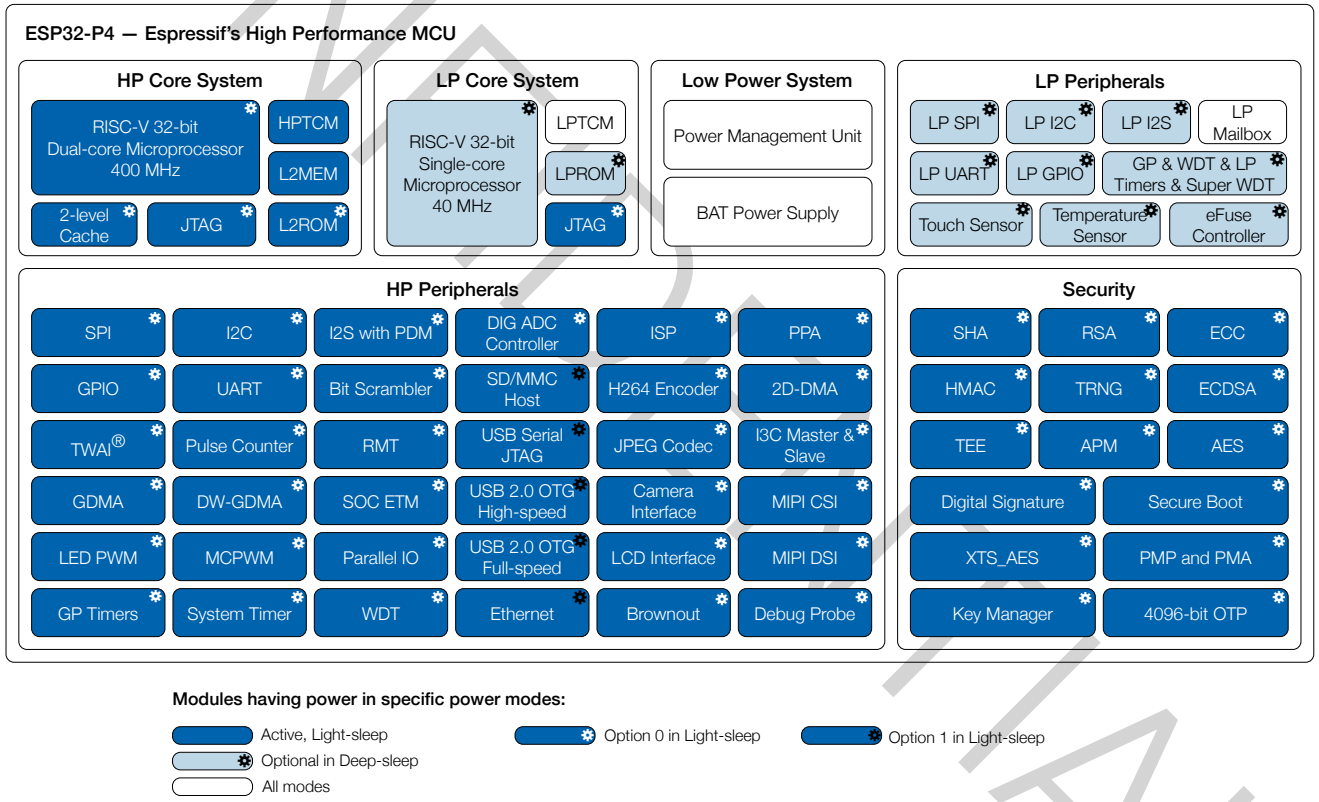
ESP32-P4NRW32

Pre-release v0.4
Espressif Systems

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

Note: The functionality of USB Serial JTAG is not available, which will be supported in the future chip revision.

Features

CPU and Memory

- 32-bit RISC-V dual-core processor up to 400 MHz for HP system
- 32-bit RISC-V single-core processor up to 40 MHz for LP system
- 128 KB HP ROM
- 16 KB LP ROM
- 768 KB HP L2MEM
- 32 KB LP SRAM
- 8 KB system TCM
- Multiple high-speed external memory interfaces
- Two-level high-speed cache

Advanced Peripheral Interfaces

- 55 programmable GPIOs
  - Five strapping GPIOs
- Image and voice processing interfaces:
  - JPEG Codec
  - Pixel processing accelerator (PPA)
  - Image signal processor (ISP)
  - H264 encoder
- Digital interfaces:
  - Four SPIs
  - LP SPI
  - Five UARTs
  - LP UART
  - I3C
  - Two I2Cs
  - LP I2C
  - Three I2Ss
  - LP I2S
  - RMT
  - LED PWM, up to 8 channels
  - Two Motor Control PWMs (MCPWMs), up to 6 channels

  - Pulse Count Controller (PCNT), up to 4 channels
  - Three TWAI® controllers, compatible with ISO 11898-1
  - USB 2.0 OTG High-Speed
  - USB 2.0 OTG Full-Speed
  - USB 2.0 Full-Speed Serial/JTAG controller
  - 100 Mbit Ethernet
  - SDMMC 3.0 host
  - MIPI CSI-2, 2-lane x 1.5 Gbps
  - MIPI DSI, 2-lane x 1.5 Gbps
  - 24-bit LCD parallel port
  - 16-bit CAM parallel port
  - Two GDMA controllers
  - DW-GDMA controller
  - 2D-DMA controller
  - Bit scrambler
  - Event Task Matrix (ETM)
  - Parallel IO interface (PARLIO)
  - LP Mailbox
- Analog interfaces:
  - Two 12-bit multi-channel ADCs
  - Temperature sensor
  - Touch sensor, up to 14 channels
  - Analog voltage comparator
  - Brown-out detector
- Timers:
  - Four 54-bit HP general-purpose timers
  - Two 52-bit HP system timers
  - Two 32-bit HP watchdog timers
  - 48-bit LP general-purpose timer
  - 32-bit LP watchdog timer
  - Analog super watchdog timer

Security

- Secure boot
- One-time writing security ensured by eFuse OTP
- Cryptographic hardware acceleration:
  - AES-128/256 (FIPS PUB 197, against DPA attack)
  - SHA Accelerator (FIPS PUB 180-4)
  - RSA Accelerator
  - ECC Accelerator
  - Elliptic Curve Digital Signature Algorithm (ECDSA)
  - Digital signature
  - HMAC
- Key Manager based on physically unclonable functions (PUF)
- Access permission management (APM)
- True Random Number Generator (TRNG)
- PMP and PMA

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
- POS Machines
- Service Robot
- Audio Devices
- Generic Low-power IoT Sensor Hubs
- Generic Low-power IoT Data Loggers
- Cameras for Video Streaming
- USB Devices
- Speech Recognition
- Image Recognition
- Touch and Proximity Sensing

Contents

List of Tables

List of Figures

1 ESP32-P4 Series Comparison

1.1 Nomenclature

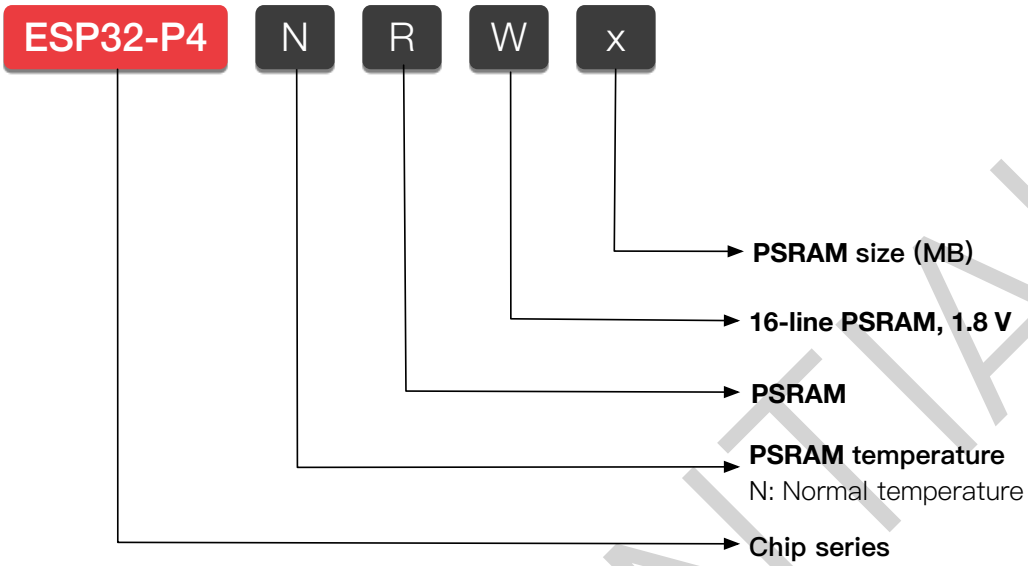


Figure 1-1. ESP32-P4 Series Nomenclature

1.2 Comparison

Table 1-1. ESP32-P4 Series Comparison

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

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

[2] For more information on VDDPST, see Section *2.6 Power Supply*.

[3] 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

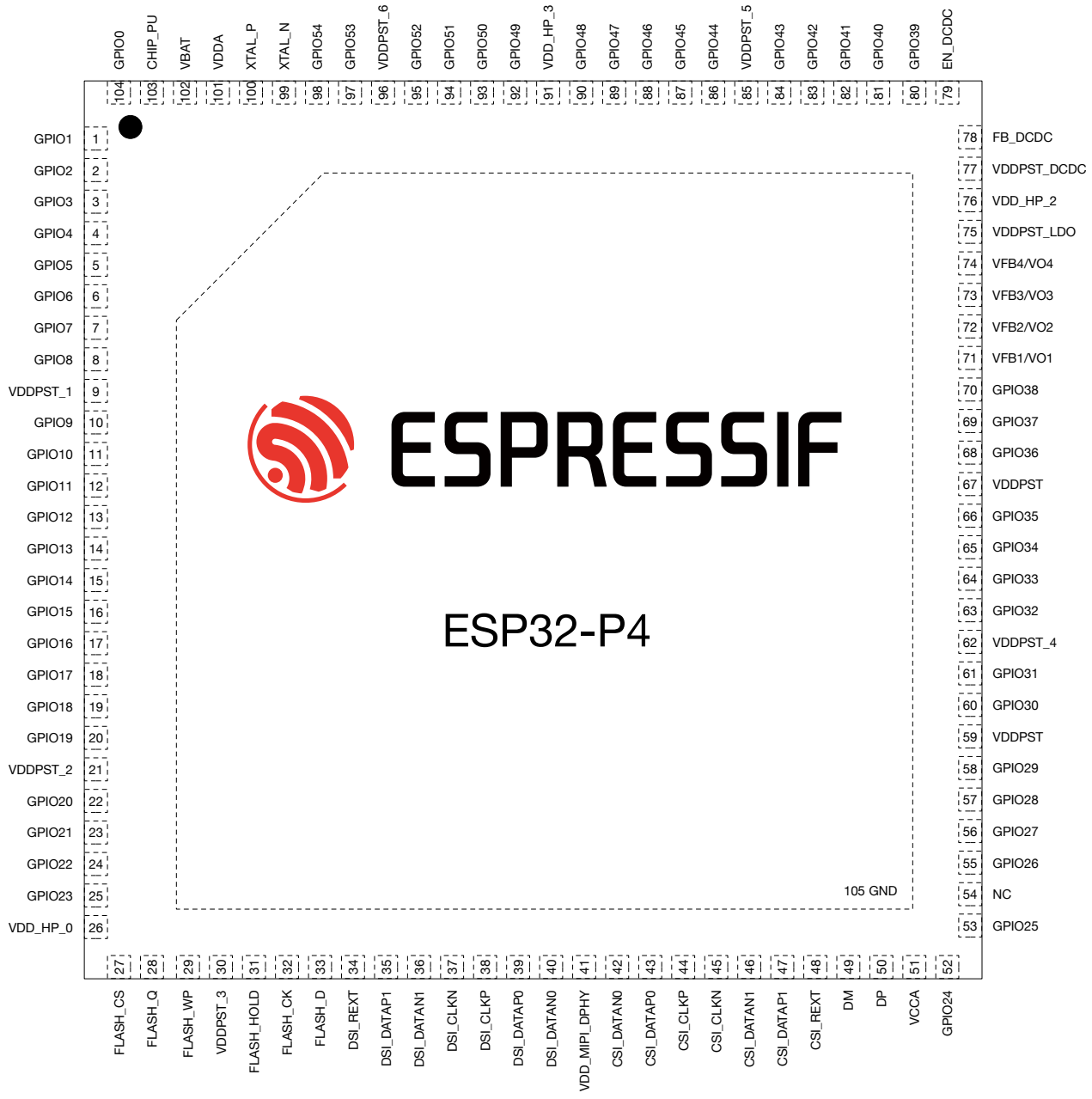


Figure 2-1. ESP32-P4 Pin Layout (Top View)

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 . 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:
  - **All** IO pins have 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 digital pins** can only be used by certain peripherals, such as flash, MIPI DSI, and MIPI CSI – see Table 2-9 *Dedicated Digital 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	VDDPST_1 / VBAT	–	–	**IO MUX**	LP IO MUX	Analog
2	GPIO2	IO	VDDPST_1 / VBAT	–	IE, WPU[5]	**IO MUX**	LP IO MUX	Analog
3	GPIO3	IO	VDDPST_1 / VBAT	–	IE	**IO MUX**	LP IO MUX	Analog
4	GPIO4	IO	VDDPST_1	–	IE	**IO MUX**	LP IO MUX	Analog
5	GPIO5	IO	VDDPST_1	–	–	**IO MUX**	LP IO MUX	Analog
6	GPIO6	IO	VDDPST_1	–	–	**IO MUX**	LP IO MUX	Analog
7	GPIO7	IO	VDDPST_1	–	–	**IO MUX**	LP IO MUX	Analog
8	GPIO8	IO	VDDPST_1	–	–	**IO MUX**	LP IO MUX	Analog
9	VDDPST_1	Power	–	–	–	–	–	–
10	GPIO9	IO	VDDPST_1	–	–	**IO MUX**	LP IO MUX	Analog
11	GPIO10	IO	VDDPST_1	–	–	**IO MUX**	LP IO MUX	Analog
12	GPIO11	IO	VDDPST_1	–	–	**IO MUX**	LP IO MUX	Analog
13	GPIO12	IO	VDDPST_1	–	–	**IO MUX**	LP IO MUX	Analog
14	GPIO13	IO	VDDPST_1	–	–	**IO MUX**	LP IO MUX	Analog
15	GPIO14	IO	VDDPST_1	–	–	**IO MUX**	LP IO MUX	Analog
16	GPIO15	IO	VDDPST_1	–	–	**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
17	GPIO16	IO	VDDPST_2	–	–	**IO MUX**	–	Analog
18	GPIO17	IO	VDDPST_2	–	–	**IO MUX**	–	Analog
19	GPIO18	IO	VDDPST_2	–	–	**IO MUX**	–	Analog
20	GPIO19	IO	VDDPST_2	–	–	**IO MUX**	–	Analog
21	VDDPST_2	Power	–	–	–	–	–	–
22	GPIO20	IO	VDDPST_2	–	–	**IO MUX**	–	Analog
23	GPIO21	IO	VDDPST_2	–	–	**IO MUX**	–	Analog
24	GPIO22	IO	VDDPST_2	–	–	**IO MUX**	–	Analog
25	GPIO23	IO	VDDPST_2	–	–	**IO MUX**	–	Analog
26	VDD_HP_0	Power	–	–	–	–	–	–
27	FLASH_CS	Dedicated	VDDPST_3	–	–	–	–	–
28	FLASH_Q	Dedicated	VDDPST_3	–	–	–	–	–
29	FLASH_WP	Dedicated	VDDPST_3	–	–	–	–	–
30	VDDPST_3	Power	–	–	–	–	–	–
31	FLASH_HOLD	Dedicated	VDDPST_3	–	–	–	–	–
32	FLASH_CK	Dedicated	VDDPST_3	–	–	–	–	–
33	FLASH_D	Dedicated	VDDPST_3	–	–	–	–	–
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	DM	Dedicated	VCCA	–	–	–	–	–
50	DP	Dedicated	VCCA	–	–	–	–	–
51	VCCA	Power	–	–	–	–	–	–
52	GPIO24	IO	VDDPST_4	–	–	IO MUX	–	**Analog**
53	GPIO25	IO	VDDPST_4	–	USB_PU	IO MUX	–	**Analog**
54	NC	–	–	–	–	–	–	–
55	GPIO26	IO	VDDPST_4	–	–	**IO MUX**	–	Analog
56	GPIO27	IO	VDDPST_4	–	–	**IO MUX**	–	Analog
57	GPIO28	IO	VDDPST_4	–	–	**IO MUX**	–	–
58	GPIO29	IO	VDDPST_4	–	–	**IO MUX**	–	–
59	VDDPST	Power	–	–	–	–	–	–
60	GPIO30	IO	VDDPST_4	–	–	**IO MUX**	–	–
61	GPIO31	IO	VDDPST_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
62	VDDPST_4	Power	–	–	–	–	–	–
63	GPIO32	IO	VDDPST_4	IE	–	**IO MUX**	–	–
64	GPIO33	IO	VDDPST_4	IE	–	**IO MUX**	–	–
65	GPIO34	IO	VDDPST_4	IE	–	**IO MUX**	–	–
66	GPIO35	IO	VDDPST_4	IE, WPU	–	**IO MUX**	–	–
67	VDDPST	Power	–	–	–	–	–	-
68	GPIO36	IO	VDDPST_4	IE	–	**IO MUX**	–	–
69	GPIO37	IO	VDDPST_4	IE	–	**IO MUX**	–	–
70	GPIO38	IO	VDDPST_4	IE	–	**IO MUX**	–	–
71	VFB/VO1	Power	–	–	–	–	–	-
72	VFB/VO2	Power	–	–	–	–	–	-
73	VFB/VO3	Power	–	–	–	–	–	-
74	VFB/VO4	Power	–	–	–	–	–	-
75	VDDPST_LDO	Power	–	–	–	–	–	–
76	VDD_HP_2	Power	–	–	–	–	–	–
77	VDDPST_DCDC	Power	–	–	–	–	–	–
78	FB_DCDC	Analog	–	–	–	–	–	–
79	EN_DCDC	Analog	–	–	–	–	–	–
80	GPIO39	IO	VDDPST_5	–	–	**IO MUX**	–	–
81	GPIO40	IO	VDDPST_5	–	–	**IO MUX**	–	–
82	GPIO41	IO	VDDPST_5	–	–	**IO MUX**	–	–
83	GPIO42	IO	VDDPST_5	–	–	**IO MUX**	–	–
84	GPIO43	IO	VDDPST_5	–	–	**IO MUX**	–	–
85	VDDPST_5	Power	–	–	–	–	–	–
86	GPIO44	IO	VDDPST_5	–	–	**IO MUX**	–	–
87	GPIO45	IO	VDDPST_5	–	–	**IO MUX**	–	–
88	GPIO46	IO	VDDPST_5	–	–	**IO MUX**	–	–
89	GPIO47	IO	VDDPST_5	–	–	**IO MUX**	–	–
90	GPIO48	IO	VDDPST_5	–	–	**IO MUX**	–	–
91	VDD_HP_3	Power	–	–	–	–	–	-
92	GPIO49	IO	VDDPST_6	–	–	**IO MUX**	–	Analog
93	GPIO50	IO	VDDPST_6	–	–	**IO MUX**	–	Analog
94	GPIO51	IO	VDDPST_6	–	–	**IO MUX**	–	Analog
95	GPIO52	IO	VDDPST_6	–	–	**IO MUX**	–	Analog
96	VDDPST_6	Power	–	–	–	–	–	-
97	GPIO53	IO	VDDPST_6	–	–	**IO MUX**	–	Analog
98	GPIO54	IO	VDDPST_6	–	–	**IO MUX**	–	Analog
99	XTAL_N	Analog	–	–	–	–	–	-
100	XTAL_P	Analog	–	–	–	–	–	-
101	VDDA	Power	–	–	–	–	–	-
102	VBAT	Power	–	–	–	–	–	-
103	CHIP_PU	Analog	–	–	–	–	–	–
104	GPIO0	IO	VDDPST_1 / VBAT	–	–	**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 VDDPST_1 / VBAT:
    - Pin Providing Power (either VDDPST_1 or VBAT) 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 UART0/1, 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 higher 4 bits data line interface and DQS interface for 3.3 V SPI2 interface in 8-line SPI mode
UART...RTS_PAD UART...CTS_PAD UART...TXD_PAD UART...RXD_PAD	Request to send Clear to send Transmit data Receive data	UART0/1 Interface
REF_50M_CLK_PAD	50 MHz reference clock output	Provide 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	GMAC PHY interface and RMII interface
SD1_CDATA..._PAD SD1_CCLK_PAD SD1_CCMD_PAD	Card data line 0 ~ 7 Card clock Card command	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	UART0_RTS_PAD	O	SPI2_D_PAD	I1/O/T
10	GPIO9	**GPIO9**	I/O/T	GPIO9	I/O/T	UART0_CTS_PAD	I1	SPI2_CK_PAD	I1/O/T
11	GPIO10	**GPIO10**	I/O/T	GPIO10	I/O/T	UART1_TXD_PAD	O	SPI2_Q_PAD	I1/O/T
12	GPIO11	**GPIO11**	I/O/T	GPIO11	I/O/T	UART1_RXD_PAD	I1	SPI2_WP_PAD	I1/O/T
13	GPIO12	**GPIO12**	I/O/T	GPIO12	I/O/T	UART1_RTS_PAD	O	–	–
14	GPIO13	**GPIO13**	I/O/T	GPIO13	I/O/T	UART1_CTS_PAD	I1	–	–
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	IO
58	GPIO29	**GPIO29**	I/O/T	GPIO29	I/O/T	SPI2_D_PAD	I1/O/T	GMAC_PHY_RXD0_PAD	IO
60	GPIO30	**GPIO30**	I/O/T	GPIO30	I/O/T	SPI2_CK_PAD	I1/O/T	GMAC_PHY_RXD1_PAD	IO
61	GPIO31	**GPIO31**	I/O/T	GPIO31	I/O/T	SPI2_Q_PAD	I1/O/T	GMAC_PHY_RXER_PAD	IO
63	GPIO32	**GPIO32**	I/O/T	GPIO32	I/O/T	SPI2_HOLD_PAD	I1/O/T	GMAC_RMII_CLK_PAD	IO
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	IO
87	GPIO45	SD1_CDATA4_PAD	I1/O/T	**GPIO45**	I/O/T	–	–	GMAC_PHY_RXDV_PAD	IO
88	GPIO46	SD1_CDATA5_PAD	I1/O/T	**GPIO46**	I/O/T	–	–	GMAC_PHY_RXD0_PAD	IO
89	GPIO47	SD1_CDATA6_PAD	I1/O/T	**GPIO47**	I/O/T	–	–	GMAC_PHY_RXD1_PAD	IO
90	GPIO48	SD1_CDATA7_PAD	I1/O/T	**GPIO48**	I/O/T	–	–	GMAC_PHY_RXER_PAD	IO

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	IO
94	GPIO51	**GPIO51**	I/O/T	GPIO51	I/O/T	–	–	GMAC_PHY_RXDV_PAD	IO
95	GPIO52	**GPIO52**	I/O/T	GPIO52	I/O/T	–	–	GMAC_PHY_RXD0_PAD	IO
97	GPIO53	**GPIO53**	I/O/T	GPIO53	I/O/T	–	–	GMAC_PHY_RXD1_PAD	IO
98	GPIO54	**GPIO54**	I/O/T	GPIO54	I/O/T	–	–	GMAC_PHY_RXER_PAD	IO
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 VDDPST_1 or VBAT.

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

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 ESP32-P4's oscillator
TOUCH_CHANNEL...	Touch sensor channel 0 ~ 13 signal	Touch sensor interface
ADC..._CHANNEL...	ADC1/2 channel 0 ~ 7 signal	ADC1/2 interface
USB1P1_N... USB1P1_P...	Negative pole of differential signal in full-speed USB OTG 2.0 Positive pole of differential signal in full-speed USB OTG 2.0	Full-speed USB OTG 2.0 interface
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_CHANNEL0	–
3	GPIO3	TOUCH_CHANNEL1	–
4	GPIO4	TOUCH_CHANNEL2	–
5	GPIO5	TOUCH_CHANNEL3	–
6	GPIO6	TOUCH_CHANNEL4	–
7	GPIO7	TOUCH_CHANNEL5	–
8	GPIO8	TOUCH_CHANNEL6	–
10	GPIO9	TOUCH_CHANNEL7	–
11	GPIO10	TOUCH_CHANNEL8	–
12	GPIO11	TOUCH_CHANNEL9	–
13	GPIO12	TOUCH_CHANNEL10	–
14	GPIO13	TOUCH_CHANNEL11	–
15	GPIO14	TOUCH_CHANNEL12	–
16	GPIO15	TOUCH_CHANNEL13	–
17	GPIO16	ADC1_CHANNEL0	–
18	GPIO17	ADC1_CHANNEL1	–
19	GPIO18	ADC1_CHANNEL2	–
20	GPIO19	ADC1_CHANNEL3	–
22	GPIO20	ADC1_CHANNEL4	–

Cont'd on next page

Table 2-7 – cont'd from previous page

Pin No.	Analog IO Name	Analog Function [1] F0	F1
23	GPIO21	ADC1_CHANNEL5	–
24	GPIO22	ADC1_CHANNEL6	–
25	GPIO23	ADC1_CHANNEL7	–
52	GPIO24	**USB1P1_N0**	–
53	GPIO25	**USB1P1_P0**	–
55	GPIO26	USB1P1_N1	–
56	GPIO27	USB1P1_P1	–
92	GPIO49	ADC2_CHANNEL2	–
93	GPIO50	ADC2_CHANNEL3	–
94	GPIO51	ADC2_CHANNEL4	ANA_COMP0
95	GPIO52	ADC2_CHANNEL5	ANA_COMP0
97	GPIO53	ADC2_CHANNEL6	ANA_COMP1
98	GPIO54	ADC2_CHANNEL7	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*.

2.3.4 Restrictions for GPIOs and LP GPIOs

All IO pins of ESP32-P4 have GPIO and some have LP GPIO pin functions. However, the IO pins are multiplexed and can be configured for different purposes based on the requirements. Some IOs have restrictions for usage. It is essential to consider the multiplexed nature and the limitations when using these IO pins.

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

The highlighted IO pins have one of the following important functions:

- **Strapping pins** – need to be at certain logic levels at startup. See Section 3 *Boot Configurations*.
- **USB1P1_N/P** – 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 3.3 *JTAG Signal Source Control*.
- **UART interface** – often used for debugging. See Table 2-2 *IO MUX Functions*.
- **LP UART interface** – often used for debugging in low-power mode. See Table 2-4 *LP IO MUX Functions*.

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

2.4 Dedicated Digital 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	Chip select	Flash connection
FLASH_Q	Data output	
FLASH_WP	Write protect	
FLASH_HOLD	Hold	
FLASH_CK	Clock	
FLASH_D	Data in	
MIPI DSI PHY 4.02 KΩ EXTERNAL RESISTOR	External resistor 4.02 KΩ	MIPI DSI connection
MIPI DSI PHY DATAP...	Data positive channel 0/1	
MIPI DSI PHY DATAN...	Data negative channel 0/1	
MIPI DSI PHY CLKN	Clock negative channel	
MIPI DSI PHY CLKP	Clock positive channel	
MIPI CSI PHY 4.02 KΩ EXTERNAL RESISTOR	External resistor 4.02 KΩ	MIPI CSI connection
MIPI CSI PHY DATAP...	Data positive channel 0/1	
MIPI CSI PHY DATAN...	Data negative channel 0/1	
MIPI CSI PHY CLKN	Clock negative channel	
MIPI CSI PHY CLKP	Clock positive channel	
USB2 OTG PHY DM	USB 2.0 OTG PHY Data minus channel	USB 2.0 OTG PHY connection
USB2 OTG PHY DP	USB 2.0 OTG PHY Data plus channel	

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

Table 2-9. Peripheral-Dedicated Functions

Pin No.	Dedicated Digital Pin	Function	
		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 Digital Pin	Function 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	DM	USB2 OTG PHY DM	I/O/T
50	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_0/1/2/3 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	IO Pins
9	VDDPST_1	Input	LP power domain	LP IO
21	VDDPST_2	Input	Digital power domain	HP IO
26	VDD_HP_0	Input	Digital power domain	
30	VDDPST_3	Input	Flash	flash IO
41	VDD_MIPI_DPHY	Input	Digital power domain	MIPI IO
51	VCCA	Input	Digital power domain	High-speed USB IO
59	VDDPST	Input	PSRAM	PSRAM IO
62	VDDPST_4	Input	Digital power domain	HP IO
67	VDDPST	Input	PSRAM	PSRAM IO
71	VFB/VO1	Output	Output 0.1 A current at the maximum	
72	VFB/VO2	Output	Output 0.1 A current at the maximum	
73	VFB/VO3	Output	Output 0.1 A current at the maximum	
74	VFB/VO4	Output	Output 0.2 A current at the maximum	
75	VDDPST_LDO	Input	Analog power domain, providing power for LDO	
76	VDD_HP_2	Input	Digital power domain	
77	VDDPST_DCDC	Input	Analog power domain, providing power for DC/DC	
85	VDDPST_5	Input	Digital power domain	HP IO
91	VDD_HP_3	Input	Digital power domain	
96	VDDPST_6	Input	Digital power domain	HP IO
101	VDDA	Input	Analog power domain	
102	VBAT	Input	Analog power domain, connecting to external batteries optionally	
105	GND	—	External ground connection	

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

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 128 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 2-12 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*.

Table 2-12. 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
- **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 system 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.

3.1 Chip Boot Mode Control

GPIO35 ~ GPIO38 control the boot mode after the reset is released. See Table 3-2 *Chip Boot Mode Control*.

Table 3-2. Boot Mode Control

Boot Mode	GPIO35	GPIO36	GPIO37	GPIO38
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

3.2 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 3-3 *UART0 ROM Message Printing Control*.

Table 3-3. 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 3-4 *USB Serial/JTAG ROM Message Printing Control*.

Table 3-4. 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.

3.3 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 3-5 shows, GPIO34 is used in combination with EFUSE_DIS_PAD_JTAG, EFUSE_DIS_USB_JTAG, and EFUSE_JTAG_SEL_ENABLE.

Table 3-5. 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:

- five-stage pipeline that supports clock frequency of up to 400 MHz
- RV32IMAFC ISA (instruction set architecture)
- Zc extensions (Zcb, Zcmp, and Zcmt)
- custom AI and DSP extension (Xai)
- custom hardware loop instructions (Xhwlp)
- 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 3 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 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.

The Xai extension has the following features:

- eight 128-bit new general-purpose registers
- 128-bit vector operations, e.g., complex multiplication, addition, subtraction, multiplication, shifting, comparison, etc
- data handling instructions and load/store operation instructions combined

- aligned and unaligned 128-bit vector data load/store
- configurable rounding and saturation modes

4.1.1.3 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.

LP CPU has the following features:

- two-stage pipeline that supports a clock frequency of up to 40 MHz
- RV32IMAC ISA (instruction set architecture)
- 32 32-bit general-purpose registers
- 32-bit multiplier and divider
- support for interrupts
- up to 2 hardware breakpoints/watchpoints
- JTAG for debugging
- compliant with RISC-V debug specification v0.13
- boot by the CPU, its dedicated timer, or LP IO

4.1.2 System DMA

This subsection describes the system DMA.

4.1.2.1 GDMA Controller

ESP32-P4 is equipped with two types of general DMAs (GDMAs) with different direct access buses, AHB and AXI. They are referred to as GDMA-AHB and GDMA-AXI. They have the following features:

- GDMA-AHB and GDMA-AXI both have 6 channels, with 3 transmit channels and 3 receive channels
- a peripheral can be mapped and bounded to any channel of GDMA
- arbitration way among channels is configurable, supporting both fixed priority arbiter and weight arbiter
- GDMA-AHB and GDMA-AXI employ linked lists to control data transfer. GDMA-AHB is limited to accessing in-package memory (SRAM and LP_MEM), while GDMA-AXI can access both in-package and out-package memory (PSRAM). They both support peripheral-to-memory and memory-to-memory data transfer at a high speed
- GDMA-AHB and GDMA-AXI both support unaligned address access
- GDMA-AHB supports I3C, UHCI, I2S, ADC, and RMT
- GDMA-AXI supports LCD, CAM, GP-SPI, PARLIO, AES, and SHA

4.1.2.2 DW-GDMA Controller

DW-GDMA controller on ESP32-P4 can realize memory-to-memory, peripheral-to-memory, and memory-to-peripheral data transfer, supporting the following features:

- two AXI master interfaces
- four channels
- hardware handshake with CSI, DSI, and ISP
- flow control using DMA or peripherals
- data transfer with unaligned starting addresses
- suspend, resume, and abort of channel transfer
- arbitration priority among channels being configured by registers
- single-block transfer
- multi-block with continuous addressing, automatic reloading register configuration, shadow registers, and linked lists

4.1.2.3 2D-DMA Controller

2D-DMA controller on ESP32-P4 is dedicated for processing 2D images, supporting the following features:

- one AXI master interface
- data transfer with unaligned starting addresses
- memory-to-memory, peripheral-to-memory (TX), and memory-to-peripheral (RX) data transfer
- three memory-to-peripheral channels, two peripheral-to-memory channels
- communication with PPA and JPEG Codec
- configurable channel priority and weight
- flexible macroblock ordering
- color format conversion

4.1.3 Memory Organization

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

Figure 4-1 illustrates the address mapping structure of ESP32-P4.

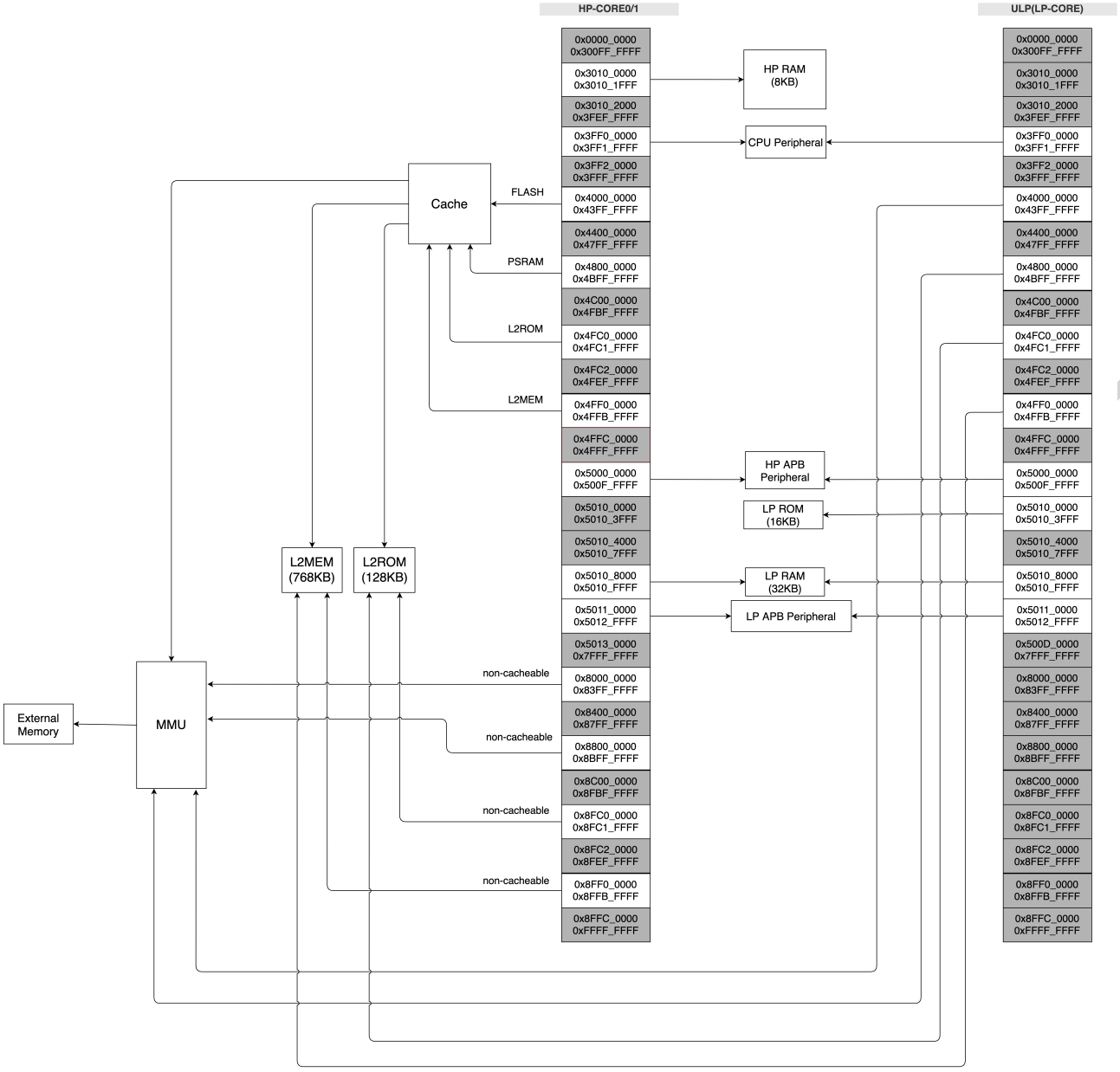


Figure 4-1. Address Mapping Structure

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 TCM**: 400 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 128 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 4-Kbit eFuse to store parameters, which are burned and read by an eFuse controller. The eFuse controller has the following features:

- 4 Kbits in total, with 1792 bits reserved for users, e.g., encryption key and device ID
- one-time programmable storage
- configurable write protection
- configurable read protection
- various hardware encoding schemes to protect against data corruption

4.1.3.3 Cache

ESP32-P4 employs the two-level cache structure, which has the following features:

- 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 IO MUX and GPIO Matrix

ESP32-P4 has 55 GPIO pins which can be assigned various functions by configuring corresponding registers. Besides digital signals, some GPIOs can be also used for analog functions, such as ADC.

All GPIOs have selectable internal pull-up or pull-down, or can be set to high impedance. When these GPIOs are configured as an input, the input value can be read by software through the register. Input GPIOs can also be set to generate edge-triggered or level-triggered CPU interrupts. All digital IO pins are bi-directional, non-inverting and tristate, including input and output buffers with tristate control. These pins can be multiplexed with other functions, such as the UART, SPI, etc. For low-power operations, the GPIOs can be set to holding state.

The IO MUX and the GPIO matrix are used to route signals from peripherals to GPIO pins. Together they provide highly configurable I/O. Using GPIO Matrix, peripheral input signals can be configured from any IO pins while peripheral output signals can be configured to any IO pins.

ESP32-P4 includes HP and LP systems, and thus it has IO MUXs and the GPIO matrices for the HP system and the LP system. GPIO0 ~ GPIO15 can be configured to be controlled by the HP system or the LP system. By default, these pins are controlled by the HP system.

4.1.4.2 Reset and Clock

CPU Clock

The CPU clock has three possible sources:

- external main crystal clock
- internal fast RC oscillator (typically about 20 MHz, and adjustable)
- 400 MHz PLL clock

The application can select the clock source from the three sources above. The selected clock source drives the CPU clock directly, or after division, depending on the application. Once the CPU is reset, the default clock source would be the external main crystal clock.

> **Note:**
> ESP32-P4 is unable to operate without an external main crystal clock.

RTC Clock

The RTC slow clock is used for RTC counter, RTC watchdog and low-power controller. It has four possible sources:

- XTAL32K_CLK (32 kHz): external crystal clock
- RC_SLOW_CLK (150 kHz by default): internal slow RC oscillator

- OSC_SLOW_CLK (32 kHz by default): external slow clock input through XTAL_32K_N
- RC32K_CLK (32 kHz): internal slow RC oscillator

The RTC fast clock is used for RTC peripherals and sensor controllers. It has three possible sources:

- XTAL_CLK (40 MHz): external main crystal clock
- RC_FAST_CLK (20 MHz by default): internal fast RC oscillator with adjustable frequency
- PLL_LP_CLK (8 MHz): internal PLL clock, with reference clock of XTAL32K_CLK

Audio PLL Clock

Audio PLL clock is a highly configurable, low-jitter and accurate clock source for audio applications, supporting frequency adjustment in the range of 6 ~ 125 MHz.

4.1.4.3 Event Task Matrix

ESP32-P4 integrates an SOC ETM with multiple channels. Each input event on channels is mapped to an output task. Events are generated by peripherals, while tasks are received by peripherals. The SOC ETM has the following features:

- up to 50 mapping channels, each connected to an event and a task and controlled independently
- an event or a task can be mapped to any tasks or events in the matrix. That is to say, one event can be mapped to different tasks via multiple channels, or different events can be mapped to the same task via their individual channels
- peripherals supporting ETM include GPIO, LED PWM, general-purpose timers, RTC Watchdog Timer, system timer, MCPWM, temperature sensor, ADC, I2S, GDMA, 2D-DMA, and PMU

4.1.4.4 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.

4.1.4.5 System Timer

ESP32-P4 integrates a 52-bit system timer, which has two 52-bit counters and three comparators. The system timer has the following features:

- counters with a fixed clock frequency of 16 MHz
- three types of independent interrupts can be generated according to the alarm value

- two alarm modes: target mode and period mode
- 52-bit target alarm value and 26-bit periodic alarm value
- automatic reload of the counter value
- counters can be stalled if CPU is stalled or in OCD mode
- events that output real-time alarms

4.1.4.6 Timer Group

ESP32-P4 is embedded with four 54-bit general-purpose timers, which are based on 16-bit prescalers and 54-bit auto-reload-capable up/down-timers.

The timers' features are summarized as follows:

- a 16-bit clock prescaler, from 2 to 65536
- a 54-bit time-base counter programmable to be incrementing or decrementing
- able to read real-time values of the time-base counter
- halting and resuming the time-base counter
- programmable alarm generation
- level interrupt generation
- events that output real-time alarms
- tasks that respond to ETM inputs, including starting/stopping timers, starting alarms, reading real-time values of timers, and reloading values of timers

4.1.4.7 Watchdog Timers

ESP32-P4 contains three watchdog timers: one in each of the two timer groups (called Main System Watchdog Timers, or MWDT) and one in the RTC module (called the RTC Watchdog Timer, or RWDT).

During the flash boot process, RWDT and the MWDT in timer group 0 (TIMG0) are enabled automatically in order to detect and recover from booting errors.

Watchdog timers have the following features:

- four stages, each with a programmable timeout value. Each stage can be configured, enabled and disabled separately
- interrupt, CPU reset, or core reset for MWDT upon expiry of each stage; interrupt, CPU reset, core reset, or system reset for RWDT upon expiry of each stage
- 32-bit expiry counter
- write protection, to prevent RWDT and MWDT configuration from being altered inadvertently
- flash boot protection
  If the boot process from an SPI flash does not complete within a predetermined period of time, the watchdog will reboot the entire main system.

4.1.4.8 RTC Timer

ESP32-P4 includes a 48-bit LP timer, which consists of one 48-bit counter and two alarm comparators. It has the following features:

- frequency of the counter is always same as that of RC_DYN_SLOW_CLK
- two alarm comparators can generate two independent interrupts based on different alarm values
- 48-bit one-time specific alarm value
- reading the counter value when CPU stalls, XTAL40M_CLK switches or system resets

4.1.4.9 Access Permission Management

ESP32-P4 integrates an APM module to manage access permissions. The module compares information transmitted over the bus with predefined configurations and decides if to grant access. APM has the following features:

- DMA APM supporting 32 regions with configurable addresses
- APB APM supporting 2 regions with configurable addresses
- APB access assigning independent APM for each peripheral address range based on region control
- managing APB access permissions for HP CORE0, HP CORE1 and LP CORE independently
- managing APB access permissions for User Mode and Machine Mode independently
- exception records

4.1.4.10 LP Mailbox

ESP32-P4 includes an LP Mailbox that provides 16 groups of 32-bit messages. A write operation to any of the 16 groups of messages can trigger LP and HP interrupts.

4.1.4.11 Brown-out Detector

With the Brown-out detector, ESP32-P4 can monitor voltages of power supply pins and trigger an interrupt or reset when voltages are abnormal.

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

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.

The following functionality is supported:

- typical AES working mode
  - AES-128/AES-256 encryption and decryption are compatible with NIST FIPS 197
- DMA-AES working mode
  - AES-128/AES-256 encryption and decryption are compatible with NIST FIPS 197
  - Block cipher mode, compliant with NIST SP 800-38A
    * ECB (Electronic Codebook)
    * CBC (Cipher Block Chaining)
    * OFB (Output Feedback)
    * CTR (Counter)
    * CFB8 (8-bit Cipher Feedback)
    * CFB128 (128-bit Cipher Feedback)
  - GCM (Galois/Counter Mode), compliant with NIST SP 800-38D
  - interrupt on completion of computation

4.1.5.2 ECC Accelerator

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).

ESP32-P4's ECC accelerator has the following features:

- two different elliptic curves, namely P-192 and P-256 defined in FIPS 186-3
- two optional coordinate systems, Affine coordinate systems and Jacobian coordinate systems
- multiple optional point operations, including point addition, point multiplication, and point verification
- multiple optional modular operations based on the order or modulus of the curve, including modular addition, modular subtraction, modular multiplication, and modular division
- interrupt upon completion of calculation

4.1.5.3 HMAC Accelerator

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. Main features are as follows:

- standard HMAC-SHA-256 algorithm
- Hash result only accessible by configurable hardware peripheral (in downstream mode)
- compatible with challenge-response authentication algorithm

- required keys for the Digital Signature (DS) peripheral (in downstream mode)
- re-enabled soft-disabled JTAG (in downstream mode)

4.1.5.4 RSA Accelerator

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. The following functionality is supported:

- 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

ESP32-P4 integrates an SHA accelerator, which is a hardware device that speeds up the SHA algorithm significantly, compared with a SHA algorithm implemented solely in software. The SHA accelerator integrated in ESP32-P4 has two working modes, Typical SHA and DMA-SHA.

The following functionality is supported:

- the following hash algorithms introduced in FIPS PUB 180-4 Spec
  - SHA-1
  - SHA-224
  - SHA-256
- 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

A Digital Signature (DS) 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 DS module providing hardware acceleration of messages' signatures based on RSA. HMAC is used as the key derivation function to output the DS_KEY key using eFuse as the input key. Subsequently, the DS module uses DS_KEY to decrypt the pre-encrypted parameters and calculate the

signature. The whole process happens in hardware so that neither the decryption key for the RSA parameters nor the input key for the HMAC key derivation function can be seen by users while calculating the signature.

The following functionality is supported:

- RSA digital signatures with key length up to 4096 bits
- encrypted private key data, only decryptable by DS module
- SHA-256 digest to protect private key data against tampering by an attacker

4.1.5.7 Elliptic Curve Digital Signature Algorithm

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 offers fast computations while ensuring the confidentiality of the signing process to prevent information leakage. This makes it a valuable tool for applications that require high-speed cryptographic operations with strong security guarantees. By using the ECDSA accelerator, users can be confident that their data is being protected without sacrificing performance.

ESP32-P4's ECDSA accelerator

- supports signature generation and verification
- supports two types of elliptic curves, P-192 and P-256 defined in FIPS 186-3
- supports two Hash algorithms for hashing operations of information, SHA-224 and SHA-256 supported by FIPS PUB 180-4 Spec
- provides high-level security with dynamic access control under different operating conditions, preventing key leakage due to any intermediate data leakage

4.1.5.8 External Memory Encryption and Decryption

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). Users can store proprietary firmware and sensitive data (e.g., credentials for gaining access to a private network) to the off-package flash. The following functionality is supported:

- general XTS-AES algorithm, compliant with IEEE Std 1619-2007
- software-based manual encryption
- high-speed auto decryption without software's participation
- encryption and decryption functions jointly enabled/disabled by registers configuration, eFuse parameters, and boot mode
- configurable Anti-DPA

4.1.5.9 True Random Number Generator

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.

4.1.5.10 Key Manager

ESP32-P4 stores and deploys keys with the Key Manager as the security core. Key Manager uses the unique physically unclonable function (PUF) of each chip to generate the hardware unique key (HUK) which is unique to the chip and serves as the root of trust (RoT) for the chip. HUK is automatically generated each time the chip is powered on and disappears when the chip is powered off. In this way, Key Manager secures key storage and deployment.

Key Manager of ESP32-P4 stores key information (non-plaintext information for recovering the key) in external memory, realizing flexible key management functionalities such as unlimited key storage and dynamic key switching.

4.2 Peripherals

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

4.2.1 Image and Voice Processing

This subsection describes the peripherals for image and voice processing.

4.2.1.1 JPEG Codec

ESP32-P4 contains a baseline JPEG codec that can be configured as a JPEG encoder or a JPEG decoder.

When configured as a JPEG encoder, it supports the following features:

- 8-bit color sampling
- original image formats: RGB888, RGB565, YUV422, and GRAY
- to-be compressed image formats: YUV444, YUV422 and YUV420, i.e. color space conversion from RGB to YUV for original images
- configurable quantization parameters
- still image encoding with a resolution up to 4K
- maximum MJPEG encoding performance of 720p@88fps or 1080p@34fps (excluding header transfer time)

When configured as a JPEG decoder, it has the following features:

- 8-bit color sampling
- compressed image formats: YUV444, YUV422, and YUV420
- four 8-bit or 16-bit precision quantization tables (determined by the quantization table transmitted in the header)
- two DC and two AC Huffman encoding tables (determined by the Huffman encoding table transmitted by the header)
- compressed images with resolutions that are multiples of 8
- still image decoding with a resolution of up to 4K
- maximum MJPEG decoding performance of 720p@88fps or 1080p@30fps (excluding header decoding time)

Pin Assignment

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

4.2.1.2 Image Signal Processor

ESP32-P4 includes an image signal processor (ISP), which supports the following features:

- maximum resolution: 1920 x 1080

- three input channels: MIPI CSI, DVP, and DW-GDMA
- input formats: RAW8, RAW10, and RAW12
- output formats: RAW8, RGB888, RGB565, YUV422, and YUV420
- pipeline structures: Bayer Filter (BF), Demosaic, Color Correction Matrix (CCM), Gamma Correction, Edge, Contrast/Hue/Saturation/Brightness, Automatic Exposure (AE), Automatic Focus (AF), Automatic White Balance (AWB), and 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

ESP32-P4 includes a pixel-processing accelerator (PPA) with scaling-rotation-mirror (SRM) and image blending (BLEND) functionalities.

- SRM enables image rotation, scaling, and mirroring, supporting
  - input formats: ARGB8888, RGB888, RGB565, and YUV420
  - output formats: ARGB8888, RGB888, RGB565, and YUV420
  - counterclockwise rotation: 90°, 180°, 270°
  - horizontal and vertical scaling with scaling factors of 8-bit integer part and 4-bit fractional part
  - horizontal and vertical mirroring
- BLEND enables blending of two layers of the same size, supporting
  - input formats: ARGB8888, RGB888, RGB565, L4, L8, A4, and A8
  - output formats: ARGB8888, RGB888, and RGB565
  - layer blending based on the Alpha channel. The Alpha channel can be provided by register configuration if layers do not contain such information
  - 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 Camera-LCD Controller

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.

LCD_CAM has the following features:

- Supports the following operation modes:
  - LCD master TX mode

  - Camera slave RX mode
  - Camera master RX mode
- Supports simultaneous connection to an external LCD and a camera
- When interfacing with an external LCD, the following is supported:
  - 8/16/24-bit parallel output modes
  - RGB, MOTO6800, and I8080 LCD formats
  - LCD data retrieved from internal memory or external memory via GDMA
- When interfacing with an external camera (i.e., DVP image sensor), the following is supported:
  - 8/16-bit parallel input modes
  - Camera data stored in internal or external memory via GDMA
- Supports interrupts

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 has the following features:

- 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 9 prediction modes for 4 x 4 partitioning and all 4 prediction modes for 16 x 16 partitioning of intra luma macroblock
- all 4 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 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 8 rectangular ROI areas at any position. These ROI areas have fixed priorities and can be overlapped with each other. Each ROI area can be assigned with 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. MIPI CSI interface supports the following features:

- 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. The MIPI DSI interface has the following features:

- 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

ESP32-P4 has five UART interfaces, UART0 ~ 4, which provide hardware flow control (CTS and RTS signals) and software flow control (XON and XOFF).

UART0 ~ 4 support asynchronous communication (RS232 and RS485) and IrDA at a speed of up to 5 Mbps. UART0 ~ 4 interfaces are connected to GDMA via the common UHCI0 (the common master control interface), and can be accessed by the GDMA controller or directly by the CPU.

Pin Assignment

For **UART0**, the pins connected to transmit and receive signals (U0TXD and U0RXD) are multiplexed with GPIO37 ~ GPIO38 and the eight-line interface of SPI2 controller via IO MUX. The pins connected to hardware flow control signals (U0RTS and U0CTS) are multiplexed with GPIO8 ~ GPIO9 and the four-line interface of SPI2 controller via IO MUX.

For **UART1**, the pins connected to transmit and receive signals (U1TXD and U1RXD) are multiplexed with GPIO10 ~ GPIO11 and the four-line interface of SPI2 controller via IO MUX. The pins connected to hardware flow control signals (U1RTS and U1CTS) are multiplexed with GPIO12 ~ GPIO13 via IO MUX.

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

4.2.2.2 SPI Controller

ESP32-P4 has the following SPI interfaces:

- flash SPI used to access off-package flash
- PSRAM SPI used to access in-package or off-package PSRAM
- SPI2 is a general-purpose SPI controller
- SPI3 is a general-purpose SPI controller
- Low-power SPI (LP-SPI)

Features of flash SPI and PSRAM SPI

- Can be configured to operate in SPI memory mode
- Data is transferred in bytes
- Flash SPI supports up to four-line SDR reads and writes, while PSRAM SPI supports up to sixteen-line DDR reads and writes
- The clock frequency is configurable. Flash SPI supports up to a maximum of 120 MHz in SDR mode, while PSRAM SPI supports up to a maximum of 250 MHz in DDR mode

Features of SPI2 and SPI3

- Works as master or as slave
- Half- and full-duplex communication
- CPU- and DMA-controlled transfer

- SPI2 supports 1-bit SPI, 2-bit Dual SPI, 4-bit Quad SPI, QPI, 8-bit Octal SPI, and OPI modes; SPI3 supports 1-bit SPI, 2-bit Dual SPI, 4-bit Quad SPI, and QPI modes
- Configurable module clock frequency: the frequency up to 80 MHz in the master mode; the frequency up to 60 MHz in the slave mode
- Configurable data length
- Configurable bit read/write order
- Configurable clock polarity and phase

Features of LP-SPI

- Works as master or as slave
- Half- and full-duplex communication
- CPU-controlled transfer
- 1-bit SPI data mode
- Configurable data length

Pin Assignment

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

The 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 and UART0/1 interfaces 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.

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

4.2.2.3 I2C Controller

ESP32-P4 has two I2C bus interfaces which are used for I2C master mode or slave mode, depending on the user's configuration. The I2C interfaces support:

- standard mode (100 Kbit/s)
- fast mode (400 Kbit/s)
- up to 800 Kbit/s (constrained by SCL and SDA pull-up strength)
- 7-bit and 10-bit addressing mode
- double addressing mode
- 7-bit broadcast address

Pin Assignment

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

4.2.2.4 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.

Pin Assignment

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

4.2.2.5 I3C Controller

ESP32-P4 includes one I3C master interface and one I3C slave interface. 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

The I3C slave interface supports the following features:

- limited compatible with I2C mode
- SDR mode
- programmable static addresses
- dynamic address allocation
- multiple Common Command Codes (CCC)
- In-Band interrupts

Pin Assignment

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

The pins for the I3C slave interface can be chosen from any GPIOs via the GPIO Matrix.

4.2.2.6 I2S Controller

ESP32-P4 includes three standard I2S interfaces. These interfaces can operate as a master or a slave in full-duplex or half-duplex mode, and can be configured for 8-bit, 16-bit, 24-bit, or 32-bit serial communication. BCK clock frequency, from 10 kHz up to 40 MHz, is supported.

The I2S interfaces support TDM PCM, TDM MSB alignment, TDM standard, and PDM interface. They connect to the GDMA controller. Among these three I2S interfaces, I2S0 supports PDM-PCM input and PCM-PDM output.

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 includes an LP I2S RX interface, which connects with internal memory and can be configured for voice activity detection (VAD). LP I2S RX interface has the following features:

- slave mode only
- I2S serial 16-bit data collection mode
- BCK clock in the range from 10 kHz to 5 MHz
- TDM PCM and TDM MSB alignment, TDM standard, and PDM RX interface

Pin Assignment

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

4.2.2.8 Pulse Count Controller

The pulse count controller (PCNT) in ESP32-P4 captures pulses and counts pulse edges in seven modes. It has the following features:

- four independent pulse counters (units) that count from 1 to 65535
- each unit consists of two independent channels sharing one pulse counter
- all channels have input pulse signals (e.g. sig_ch0_u*n*) with their corresponding control signals (e.g. ctrl_ch0_u*n*)
- independently filter glitches of input pulse signals (sig_ch0_u*n* and sig_ch1_u*n*) and control signals (ctrl_ch0_u*n* and ctrl_ch1_u*n*) on each unit
- each channel has the following parameters:
  1. selection between counting on positive or negative edges of the input pulse signal
  2. configuration to Increment, Decrement, or Disable counter mode for control signal's high and low states
- can be reset by inputting signals from pins
- pulse frequency up to 40 MHz

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 OTG High-Speed

ESP32-P4 features a high-speed USB OTG interface along with an integrated transceiver. The USB OTG interface complies with the USB 2.0 specification. It has the following features:

General Features

- compatible with USB 2.0, OTG 1.3, and OTG 2.0
- high-speed and full-speed data rates
- support Dual-role-devices (DRD), i.e. it can be used both as host and device
- dynamic FIFO (DFIFO) sizing up to 4 KB
- multiple memory access modes
  - Scatter/Gather DMA mode
  - Buffer DMA mode
- integrated UTMI high-speed transceiver
- remote wake-up

Device Mode Features

- Endpoint 0 always existing (bi-directional, consisting of EP0 IN and EP0 OUT)
- 15 additional endpoints (Endpoint 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 channels
  - a control channel consists of two channels (IN and OUT), as IN and OUT transactions must be handled separately. Only the Control transfer type is supported.
  - each of the other 15 channels is dynamically configurable to be IN or OUT, and supports Bulk, Isochronous, and Interrupt transfer types.
- all channels share an RX FIFO, non-periodic TX FIFO, and periodic TX FIFO. The size of each FIFO is configurable.

Pin Assignment

For USB 2.0 OTG High-Speed interface, the pins for USB_D- and USB_D+ are dedicated pins 49 ~ 50. Other signals can be routed to any GPIOs via the GPIO matrix.

4.2.2.10 USB 2.0 OTG Full-Speed

ESP32-P4 features a full-speed USB OTG interface along with an integrated transceiver. The USB OTG interface complies with the USB 2.0 specification. It has the following features:

General Features

- full-speed and low-speed data rates
- HNP and SRP as A-device or B-device
- dynamic FIFO (DFIFO) sizing up to 1 KB

- multiple memory access modes
  - Scatter/Gather DMA mode
  - Buffer DMA mode
- two integrated full-speed transceivers
- choosing from two 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

Device Mode Features

- Endpoint 0 always existing (bi-directional, consisting of EP0 IN and EP0 OUT)
- six additional endpoints (Endpoint 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

- 8 channels
  - a control channel consists of two channels (IN and OUT), as IN and OUT transactions must be handled separately. Only the Control transfer type is supported.
  - each of the other seven channels is dynamically configurable to be IN or OUT, and supports Bulk, Isochronous, and Interrupt transfer types.
- All channels share an RX FIFO, non-periodic TX FIFO, and periodic TX FIFO. The size of each FIFO is configurable.

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 OTG Full-Speed 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

ESP32-P4 integrates a USB Serial/JTAG controller. This controller has the following features:

- 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 for the USB Serial/JTAG Controller are multiplexed with GPIO24 ~ GPIO25. Alternatively, these pins can also be multiplexed with GPIO26 ~ GPIO27.

4.2.2.12 Ethernet Media Access Controller

Ethernet MAC controller on ESP32-P4 transfers data in compliance with the standard IEEE802.3-2008. It supports the following features:

- data transfer through MII or RMII interface
- reading and writing PHY register through MDIO interface
- IEEE1588-2002 and IEEE1588-2008
- Energy-Efficient Ethernet (EEE)
- Magic Packet Detection
- Remote Wake-up Frame Detection
- full-duplex and half-duplex communication
- built-in DMA transferring data with linked lists

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 and the SDI3.0 interface 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.

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

ESP32-P4 has one TWAI® controller with the following features:

- compatible 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, and Self-Test (no acknowledgment required)

- 64-byte receive FIFO
- acceptance filter (single and dual filter modes)
- error detection and handling: error counters, configurable error interrupt threshold, error code capture, arbitration lost capture, and the automatic stand-by mode of the transceiver
- receiving timestamp of data frames

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

ESP32-P4 has an SD/MMC Host Controller with the following features:

- Secure Digital (SD) memory version 3.0 and version 3.01
- Secure Digital I/O (SDIO) version 3.0
- Consumer Electronics Advanced Transport Architecture (CE-ATA) version 1.1
- Multimedia Cards (MMC version 4.41, eMMC version 4.5 and version 4.51)
- up to 80 MHz clock output
- three data bus modes:
  - 1-bit
  - 4-bit (supports two SD/SDIO/MMC 4.41 cards, and one SD card operating at 1.8 V in 4-bit mode)
  - 8-bit

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

The LED PWM controller can generate independent digital waveform on eight channels. The LED PWM controller supports:

- generating digital waveform with configurable periods and duty cycle. The resolution of duty cycle can be up to 20 bits
- multiple clock sources, including 80 MHz PLL clock, external main crystal clock, and internal fast RC oscillator
- operation when the CPU is in Light-sleep mode
- gradual increase or decrease of duty cycle, which is useful for the LED RGB color-gradient generator

- up to 16 duty cycle ranges for gamma curve generation, each can be independently configured in terms of duty cycle direction (increase or decrease), step size, the number of steps, and step frequency

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

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, and a dedicated capture submodule.

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

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. All eight channels share a 384 × 32-bit memory block to store transmit or receive waveform. There are one transmit channel and one receive channel that support DMA access.

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

ESP32-P4 contains a Parallel IO controller (PARLIO) for transmitting and receiving parallel data up to 16-bit. It connects to the GDMA controller and includes two modules, one transmit module and one receive module. It has the following features:

- receive/transmit module supporting multiple clock sources of up to 40 MHz and clock division
- receiving/transmitting 1/2/4/8/16-bit data
- receive module supporting multiple data collection modes

Pin Assignment

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

4.2.2.19 Bit-scrambler

The BitScrambler controller includes transmit and receive channels, located in the transmit and receive paths of GDMA, respectively. The BitScrambler controller can be used for endianness conversion, adding/deleting data, formatting data stream, etc. In addition, lookup tables can be used to implement functionalities such as generating complex waveforms, ADC curve correction, etc.

Its transmit and receive channels can work independently. Each channel supports six instructions and contains the instruction cache with a depth of eight and a 2048-byte lookup table. The six instructions can be programmable by users, and the controller processes the data stream in bits according to a user-defined instruction sequence.

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.

Pin Assignment

The pins of the touch sensor are multiplexed with GPIO2 ~ GPIO15, LP_GPIO2 ~ LP_GPIO15, UART0/1 interfaces, 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

The temperature sensor generates a voltage that varies with temperature. The voltage is internally converted via an ADC into a digital value.

The temperature sensor has a range of –40 °C to 125 °C. It is designed primarily to sense the temperature changes inside the chip. The temperature value depends on factors like microcontroller clock frequency or I/O load. Generally, the chip's internal temperature is higher than the ambient temperature.

Pin Assignment

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

4.2.3.3 ADC Controller

ESP32-P4 integrates two 12-bit SAR ADCs which support measurements on 14 channels (analog-enabled pins).

Pin Assignment

The pins of the ADC controller are multiplexed with GPIO16 ~ GPIO23, GPIO49 ~ GPIO54, and the third RMII interface of EMAC.

4.2.3.4 Analog Voltage Comparator

ESP32-P4 provides two groups of analog voltage comparators, and each group contains two PADs. This peripheral can be used to compare the voltages of the two PADs or compare the voltage of one PAD with a stable internal voltage that can be adjustable.

Pin Assignment

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

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
VDDPST_LDO, VDDPST_DCDC, VDDA, VBAT	Voltage applied to power supply pins per power domain	–0.3	3.6	V
VDDPST_1, VDDPST_2, VDDPST_3, VDDPST_4, VDDPST_5, VDDPST_6, VDDPST	Voltage applied to I/O power supply pins per power domain	–0.3	3.6	V
VDD_HP_0, VDD_HP_2, VDD_HP_3	Voltage applied to core power supply pins per power domain (from DCDC)	0	1.3	V
VDD_MIPI_DPHY	Voltage applied to MIPI PHY power supply pins per power domain	0	2.75	V
VCCA	Voltage applied to USB_PHY power supply pins per power domain	–0.66	3.96	V
T_{STORE}	Storage temperature	–40	150	°C

5.2 Recommended Operating Conditions

Table 5-2. Recommended Operating Conditions

Parameter	Description	Min	Typ	Max	Unit
VDDPST_LDO, VDDPST_DCDC, VDDA, VBAT	Voltage applied to power supply pins per power domain	3.0	3.3	3.6	V
VDDPST_1, VDDPST_2, VDDPST_3, VDDPST_4, VDDPST_5, VDDPST_6, VDDPST	Voltage applied to I/O power supply pins per power domain	1.65/3.0	1.8/3.3	1.95/3.6	V
VDD_HP_0, VDD_HP_2, VDD_HP_3	Voltage applied to core power supply pins per power domain (from DCDC)	0.99	1.1	1.21	V

Cont'd on next page

Table 5-2 – cont'd from previous page

Parameter	Description	Min	Typ	Max	Unit
VDD_MIPI_DPHY	Voltage applied to MIPI PHY power supply pins per power domain	2.25	2.5	2.75	V
VCCA	Voltage applied to USB_PHY power supply pins per power domain	2.97	3.3	3.63	V
I_{VDD}[1]	Current supplied to core	0.5	—	—	A
T_A	Ambient temperature	–40	—	85	°C

[1] The output current of VDD_HP_x should be 500 mA or more.

5.3 VFB_VO1 Output Characteristics

Table 5-3. VFB_VO1 Internal and Output Characteristics

Parameter	Description	Typ	Unit
R_{VFB}	VFB_VO1 powered by VDDPST_LDO via R_{VFB} for 3.3 V flash [1]	7.5	Ω
I_{VFB}	Output current when VFB_VO1 is powered by Flash Voltage Regulator for 1.8 V flash	100	mA

[1] VDDPST_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_VBAT[1]	—	VDD_VBAT[1]+ 0.3	V
V_{IL_nRST}	Chip reset voltage (CHIP_PU should satisfy the required voltage)	–0.3	—	0.25 × VDD_VBAT[1]	V

[1] VDD_VBAT is the voltage for VBAT, which provides power for the analog circuit.

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

5.5 Current Consumption in Active and Low-power Modes

Table 5-5. Current Consumption in Active Mode

Mode	CPU Frequency (MHz)	Description	Typ (mA)	
			All Peripherals Clocks Disabled	All Peripherals Clocks Enabled[1]
Active[2]	360	CPU is running	NA	77
		CPU is idle	41	NA
	180	CPU is running	NA	64
		CPU is idle	36	NA
	90	CPU is running	NA	48
		CPU is idle	30	NA
	40	CPU is running	NA	37
		CPU is idle	26	NA

[1] In practice, the current consumption might be different depending on which peripherals are enabled.

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

Table 5-6. Current Consumption in Low-Power Modes

Mode	Description	Typ (mA)
Light-sleep	All GPIOs are high-impedance, and all power supplies are enabled	3.4
	All GPIOs are high-impedance, most of peripherals are disabled, and chip is connected through USB	0.2
	All peripherals are disabled, and data is stored in HP memory	0.15
Deep-sleep	LP timer and LP memory are powered on	0.013
Power off	CHIP_PU is set to low level, the chip is powered off	NA

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	VDDPST_1/VBAT	–	–	**GPIO1**	I/O/T	GPIO1	I/O/T	–	–	–	–	LP_GPIO1	I/O/T	LP_GPIO1	I/O/T	XTAL_32K_P	–
2	GPIO2	IO	VDDPST_1/VBAT	–	IE, WPU	**MTCK**	I1	GPIO2	I/O/T	–	–	–	–	LP_GPIO2	I/O/T	LP_GPIO2	I/O/T	TOUCH_CHANNEL0	–
3	GPIO3	IO	VDDPST_1/VBAT	–	IE	**MTDI**	I1	GPIO3	I/O/T	–	–	–	–	LP_GPIO3	I/O/T	LP_GPIO3	I/O/T	TOUCH_CHANNEL1	–
4	GPIO4	IO	VDDPST_1	–	IE	**MTMS**	I0	GPIO4	I/O/T	–	–	–	–	LP_GPIO4	I/O/T	LP_GPIO4	I/O/T	TOUCH_CHANNEL2	–
5	GPIO5	IO	VDDPST_1	–	–	**MTDO**	O/T	GPIO5	I/O/T	–	–	–	–	LP_GPIO5	I/O/T	LP_GPIO5	I/O/T	TOUCH_CHANNEL3	–
6	GPIO6	IO	VDDPST_1	–	–	**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_CHANNEL4	–
7	GPIO7	IO	VDDPST_1	–	–	**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_CHANNEL5	–
8	GPIO8	IO	VDDPST_1	–	–	**GPIO8**	I/O/T	GPIO8	I/O/T	UART0_RTS_PAD	O	SPI2_D_PAD	I1/O/T	LP_GPIO8	I/O/T	LP_GPIO8	I/O/T	TOUCH_CHANNEL6	–
9	VDDPST_1	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
10	GPIO9	IO	VDDPST_1	–	–	**GPIO9**	I/O/T	GPIO9	I/O/T	UART0_CTS_PAD	I1	SPI2_CK_PAD	I1/O/T	LP_GPIO9	I/O/T	LP_GPIO9	I/O/T	TOUCH_CHANNEL7	–
11	GPIO10	IO	VDDPST_1	–	–	**GPIO10**	I/O/T	GPIO10	I/O/T	UART1_TXD_PAD	O	SPI2_Q_PAD	I1/O/T	LP_GPIO10	I/O/T	LP_GPIO10	I/O/T	TOUCH_CHANNEL8	–
12	GPIO11	IO	VDDPST_1	–	–	**GPIO11**	I/O/T	GPIO11	I/O/T	UART1_RXD_PAD	I1	SPI2_WP_PAD	I1/O/T	LP_GPIO11	I/O/T	LP_GPIO11	I/O/T	TOUCH_CHANNEL9	–
13	GPIO12	IO	VDDPST_1	–	–	**GPIO12**	I/O/T	GPIO12	I/O/T	UART1_RTS_PAD	O	–	–	LP_GPIO12	I/O/T	LP_GPIO12	I/O/T	TOUCH_CHANNEL10	–
14	GPIO13	IO	VDDPST_1	–	–	**GPIO13**	I/O/T	GPIO13	I/O/T	UART1_CTS_PAD	I1	–	–	LP_GPIO13	I/O/T	LP_GPIO13	I/O/T	TOUCH_CHANNEL11	–
15	GPIO14	IO	VDDPST_1	–	–	**GPIO14**	I/O/T	GPIO14	I/O/T	–	–	–	–	LP_UART_TXD_PAD	O	LP_GPIO14	I/O/T	TOUCH_CHANNEL12	–
16	GPIO15	IO	VDDPST_1	–	–	**GPIO15**	I/O/T	GPIO15	I/O/T	–	–	–	–	LP_UART_RXD_PAD	I1	LP_GPIO15	I/O/T	TOUCH_CHANNEL13	–
17	GPIO16	IO	VDDPST_2	–	–	**GPIO16**	I/O/T	GPIO16	I/O/T	–	–	–	–	–	–	–	–	ADC1_CHANNEL0	–
18	GPIO17	IO	VDDPST_2	–	–	**GPIO17**	I/O/T	GPIO17	I/O/T	–	–	–	–	–	–	–	–	ADC1_CHANNEL1	–
19	GPIO18	IO	VDDPST_2	–	–	**GPIO18**	I/O/T	GPIO18	I/O/T	–	–	–	–	–	–	–	–	ADC1_CHANNEL2	–
20	GPIO19	IO	VDDPST_2	–	–	**GPIO19**	I/O/T	GPIO19	I/O/T	–	–	–	–	–	–	–	–	ADC1_CHANNEL3	–
21	VDDPST_2	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
22	GPIO20	IO	VDDPST_2	–	–	**GPIO20**	I/O/T	GPIO20	I/O/T	–	–	–	–	–	–	–	–	ADC1_CHANNEL4	–
23	GPIO21	IO	VDDPST_2	–	–	**GPIO21**	I/O/T	GPIO21	I/O/T	–	–	–	–	–	–	–	–	ADC1_CHANNEL5	–
24	GPIO22	IO	VDDPST_2	–	–	**GPIO22**	I/O/T	GPIO22	I/O/T	–	–	–	–	–	–	–	–	ADC1_CHANNEL6	–
25	GPIO23	IO	VDDPST_2	–	–	**GPIO23**	I/O/T	GPIO23	I/O/T	–	–	REF_50M_CLK_PAD	O	–	–	–	–	ADC1_CHANNEL7	–
26	VDD_HP_0	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
27	FLASH_CS	Dedicated	VDDPST_3	–	–	FLASH_CS	O	–	–	–	–	–	–	–	–	–	–	–	–
28	FLASH_Q	Dedicated	VDDPST_3	–	–	FLASH_Q	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
29	FLASH_WP	Dedicated	VDDPST_3	–	–	FLASH_WP	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
30	VDDPST_3	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
31	FLASH_HOLD	Dedicated	VDDPST_3	–	–	FLASH_HOLD	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
32	FLASH_CK	Dedicated	VDDPST_3	–	–	FLASH_CK	O	–	–	–	–	–	–	–	–	–	–	–	–
33	FLASH_D	Dedicated	VDDPST_3	–	–	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 5-7 – 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	VCCA	–	–	USB2 OTG PHY DM	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
50	DP	Dedicated	VCCA	–	–	USB2 OTG PHY DP	I/O/T	–	–	–	–	–	–	–	–	–	–	–	–
51	VCCA	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
52	GPIO24	IO	VDDPST_4	–	–	GPIO24	I/O/T	GPIO24	I/O/T	–	–	–	–	–	–	–	–	USB1P1_N0	–
53	GPIO25	IO	VDDPST_4	–	IE, USB_WPU	GPIO25	I/O/T	GPIO25	I/O/T	–	–	–	–	–	–	–	–	USB1P1_P0	–
54	NC	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
55	GPIO26	IO	VDDPST_4	–	–	**GPIO26**	I/O/T	GPIO26	I/O/T	–	–	–	–	–	–	–	–	USB1P1_N1	–
56	GPIO27	IO	VDDPST_4	–	–	**GPIO27**	I/O/T	GPIO27	I/O/T	–	–	–	–	–	–	–	–	USB1P1_P1	–
57	GPIO28	IO	VDDPST_4	–	–	**GPIO28**	I/O/T	GPIO28	I/O/T	SPI2_CS_PAD	I1/O/T	GMAC_PHY_RXDV_PAD	IO	–	–	–	–	–	–
58	GPIO29	IO	VDDPST_4	–	–	**GPIO29**	I/O/T	GPIO29	I/O/T	SPI2_D_PAD	I1/O/T	GMAC_PHY_RXD0_PAD	IO	–	–	–	–	–	–
59	VDDPST	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
60	GPIO30	IO	VDDPST_4	–	–	**GPIO30**	I/O/T	GPIO30	I/O/T	SPI2_CK_PAD	I1/O/T	GMAC_PHY_RXD1_PAD	IO	–	–	–	–	–	–
61	GPIO31	IO	VDDPST_4	–	–	**GPIO31**	I/O/T	GPIO31	I/O/T	SPI2_Q_PAD	I1/O/T	GMAC_PHY_RXER_PAD	IO	–	–	–	–	–	–
62	VDDPST_4	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
63	GPIO32	IO	VDDPST_4	IE	–	**GPIO32**	I/O/T	GPIO32	I/O/T	SPI2_HOLD_PAD	I1/O/T	GMAC_RMII_CLK_PAD	IO	–	–	–	–	–	–
64	GPIO33	IO	VDDPST_4	IE	–	**GPIO33**	I/O/T	GPIO33	I/O/T	SPI2_WP_PAD	I1/O/T	GMAC_PHY_TXEN_PAD	O	–	–	–	–	–	–
65	GPIO34	IO	VDDPST_4	IE	–	**GPIO34**	I/O/T	GPIO34	I/O/T	SPI2_IO4_PAD	I1/O/T	GMAC_PHY_TXD0_PAD	O	–	–	–	–	–	–
66	GPIO35	IO	VDDPST_4	IE, WPU	–	**GPIO35**	I/O/T	GPIO35	I/O/T	SPI2_IO5_PAD	I1/O/T	GMAC_PHY_TXD1_PAD	O	–	–	–	–	–	–
67	VDDPST	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
68	GPIO36	IO	VDDPST_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	VDDPST_4	IE	IE	**UART0_TXD_PAD**	O	GPIO37	I/O/T	SPI2_IO7_PAD	I1/O/T	–	–	–	–	–	–	–	–
70	GPIO38	IO	VDDPST_4	IE	–	**UART0_RXD_PAD**	I1	GPIO38	I/O/T	SPI2_DQS_PAD	O/T	–	–	–	–	–	–	–	–
71	VFB/VO1	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
72	VFB/VO2	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
73	VFB/VO3	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
74	VFB/VO4	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
75	VDDPST_LDO	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
76	VDD_HP_2	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
77	VDDPST_DCDC	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
78	FB_DCDC	Analog	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
79	EN_DCDC	Analog	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
80	GPIO39	IO	VDDPST_5	–	–	SD1_CDATA0_PAD	I1/O/T	**GPIO39**	I/O/T	–	–	REF_50M_CLK_PAD	O	–	–	–	–	–	–
81	GPIO40	IO	VDDPST_5	–	–	SD1_CDATA1_PAD	I1/O/T	**GPIO40**	I/O/T	–	–	GMAC_PHY_TXEN_PAD	O	–	–	–	–	–	–
82	GPIO41	IO	VDDPST_5	–	–	SD1_CDATA2_PAD	I1/O/T	**GPIO41**	I/O/T	–	–	GMAC_PHY_TXD0_PAD	O	–	–	–	–	–	–
83	GPIO42	IO	VDDPST_5	–	–	SD1_CDATA3_PAD	I1/O/T	**GPIO42**	I/O/T	–	–	GMAC_PHY_TXD1_PAD	O	–	–	–	–	–	–
84	GPIO43	IO	VDDPST_5	–	–	SD1_CCLK_PAD	O	**GPIO43**	I/O/T	–	–	GMAC_PHY_TXER_PAD	O	–	–	–	–	–	–
85	VDDPST_5	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
86	GPIO44	IO	VDDPST_5	–	–	SD1_CCMD_PAD	I1/O/T	**GPIO44**	I/O/T	–	–	GMAC_RMII_CLK_PAD	IO	–	–	–	–	–	–
87	GPIO45	IO	VDDPST_5	–	–	SD1_CDATA4_PAD	I1/O/T	**GPIO45**	I/O/T	–	–	GMAC_PHY_RXDV_PAD	IO	–	–	–	–	–	–
88	GPIO46	IO	VDDPST_5	–	–	SD1_CDATA5_PAD	I1/O/T	**GPIO46**	I/O/T	–	–	GMAC_PHY_RXD0_PAD	IO	–	–	–	–	–	–
89	GPIO47	IO	VDDPST_5	–	–	SD1_CDATA6_PAD	I1/O/T	**GPIO47**	I/O/T	–	–	GMAC_PHY_RXD1_PAD	IO	–	–	–	–	–	–
90	GPIO48	IO	VDDPST_5	–	–	SD1_CDATA7_PAD	I1/O/T	**GPIO48**	I/O/T	–	–	GMAC_PHY_RXER_PAD	IO	–	–	–	–	–	–
91	VDD_HP_3	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
92	GPIO49	IO	VDDPST_6	–	–	**GPIO49**	I/O/T	GPIO49	I/O/T	–	–	GMAC_PHY_TXEN_PAD	O	–	–	–	–	ADC2_CHANNEL2	–
93	GPIO50	IO	VDDPST_6	–	–	**GPIO50**	I/O/T	GPIO50	I/O/T	–	–	GMAC_RMII_CLK_PAD	IO	–	–	–	–	ADC2_CHANNEL3	–
94	GPIO51	IO	VDDPST_6	–	–	**GPIO51**	I/O/T	GPIO51	I/O/T	–	–	GMAC_PHY_RXDV_PAD	IO	–	–	–	–	ADC2_CHANNEL4	ANA_COMPO
95	GPIO52	IO	VDDPST_6	–	–	**GPIO52**	I/O/T	GPIO52	I/O/T	–	–	GMAC_PHY_RXD0_PAD	IO	–	–	–	–	ADC2_CHANNEL5	ANA_COMPO
96	VDDPST_6	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
97	GPIO53	IO	VDDPST_6	–	–	**GPIO53**	I/O/T	GPIO53	I/O/T	–	–	GMAC_PHY_RXD1_PAD	IO	–	–	–	–	ADC2_CHANNEL6	ANA_COMP1
98	GPIO54	IO	VDDPST_6	–	–	**GPIO54**	I/O/T	GPIO54	I/O/T	–	–	GMAC_PHY_RXER_PAD	IO	–	–	–	–	ADC2_CHANNEL7	ANA_COMP1
99	XTAL_N	Analog	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
100	XTAL_P	Analog	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Cont'd on next page

Table 5-7 – 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	VDDA	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
102	VBAT	Power	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
103	CHIP_PU	Analog	VDDA	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
104	GPIO0	IO	VDDPST_1/VBAT	–	–	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
2024-06-04	v0.4	Preliminary release
2024-05-24	v0.3	Preliminary release
2024-01-09	v0.2	Preliminary release
2023-07-26	v0.1	Preliminary release

www.espressif.com

Disclaimer and Copyright Notice

Information in this document, including URL references, is subject to change without notice.

ALL THIRD PARTY' S INFORMATION IN THIS DOCUMENT IS PROVIDED AS IS WITH NO WARRANTIES TO ITS AUTHENTICITY AND ACCURACY.

NO WARRANTY IS PROVIDED TO THIS DOCUMENT FOR ITS MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, NOR DOES ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.

All liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No licenses express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

The Wi-Fi Alliance Member logo is a trademark of the Wi-Fi Alliance. The Bluetooth logo is a registered trademark of Bluetooth SIG.

All trade names, trademarks and registered trademarks mentioned in this document are property of their respective owners, and are hereby acknowledged.