# ESP32-P4

## Datasheet

**High-performance MCU with RISC-V single-core and dual-core microprocessors**

**Large memory**

**Powerful image and voice processing capability**

**Including:**

ESP32-P4R16 (16 MB In-Package PSRAM)

ESP32-P4R32 (32 MB In-Package PSRAM)



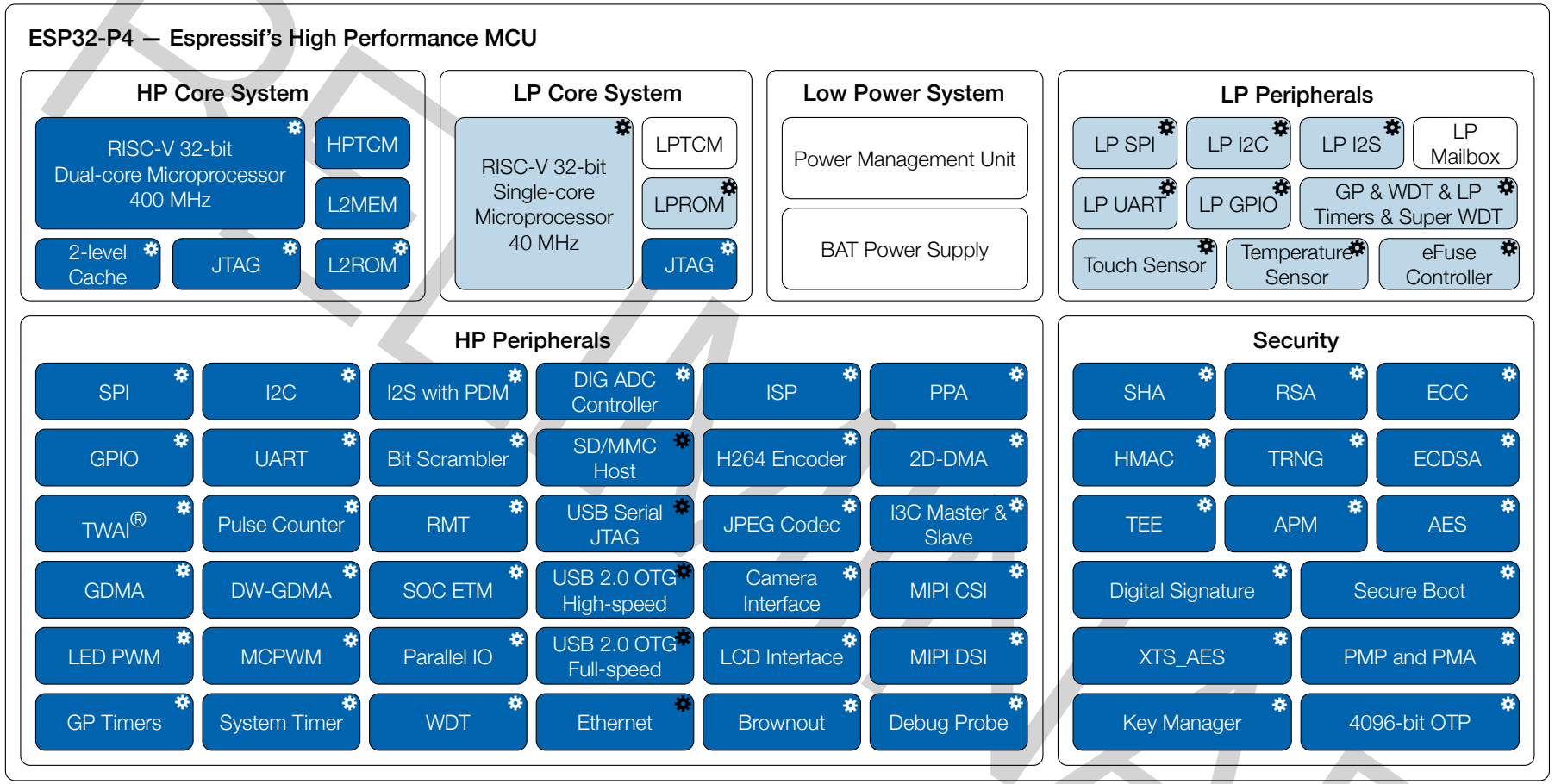
Pre-release v0.1
Espressif Systems

# 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. It has:

- 32-bit RISC-V dual-core processor up to 400 MHz with AI instruction extension and FPU for the HP system
- 32-bit RISC-V single-core processor up to 40 MHz for the LP system
- 128 KB of HP ROM, 16 KB of LP ROM, 768 KB of HP L2MEM, 32 KB of LP SRAM, and 8 KB system tightly-coupled memory (TCM), allowing for zero-wait access
- 96 KB of I1 cache, 128 KB/256 KB/512 KB of I2 cache
- Powerful image and voice processing functionality
  - JPEG Codec
  - Pixel processing accelerator (PPA)
  - Image signal processor (ISP)
  - H264 encoder
- Rich set of peripheral interfaces and 55 GPIOs, ideal for various scenarios and complex applications
- Reliable security features ensured by
  - Cryptographic hardware accelerators that support AES-128/192/256, RSA, HMAC, ECC, digital signature, and secure boot
  - Key manager based on physically unclonable function (PUF)
  - True Random Number Generator (TRNG)
  - PMP and PMA
  - Permission control on accessing internal memory, external memory, and peripherals
  - External memory encryption and decryption

# Block Diagram




**Block Diagram of ESP32-P4**

# 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
- Image and voice processing interfaces
  - JPEG Codec
  - Pixel processing accelerator (PPA)
  - Image signal processor (ISP)
  - H264 encoder
- Digital interfaces:
  - 4 × SPI
  - 1 × LP SPI
  - 5 × UART
  - 1 × LP UART
  - 1 × I3C
  - 2 × I2C
  - 1 × LP I2C
  - 3 × I2S
  - 1 × LP I2S
  - 1 × RMT
  - 1 × LED PWM, up to 8 channels
  - 2 × Motor Control PWM (MCPWM), up to 6 channels
  - 1 × Pulse Count Controller (PCNT), up to 4 channels
  - 3 × TWAI® controller, compatible with ISO 11898-1 (CAN Specification 2.0)
  - 1 × USB 2.0 OTG High-Speed
  - 1 × USB 2.0 OTG Full-Speed
  - 1 × USB 2.0 Full-Speed Serial/JTAG controller
  - 1 × 100 Mbit Ethernet
  - 1 × SDMMC 3.0 host
  - 1 × MIPI CSI-2, 2-lane x 1.5 Gbps
  - 1 × MIPI DSI, 2-lane x 1.5 Gbps
  - 1 × 24-bit LCD parallel port
  - 1 × 16-bit CAM parallel port
  - 2 x GDMA controller
  - 1 x DW-GDMA controller
  - 1 x 2D-DMA controller
  - 1 × Bit scrambler
  - 1 × Event Task Matrix (ETM)
  - 1 × Parallel IO interface (PARLIO)
  - 1 × LP Mailbox
- Analog interfaces:
  - 2 × 12-bit multi-channel ADC
  - 1 × Temperature sensor
  - 1 × Touch sensor, up to 14 channels
  - 1 × Analog PAD voltage comparator
  - 1 × Brown-out detector
- Timers:
  - 4 × 54-bit HP general-purpose timers
  - 2 × 52-bit HP system timers
  - 2 × 32-bit HP watchdog timers

  - 1 × 48-bit LP general-purpose timer
  - 1 × 32-bit LP watchdog timer
  - 1 × 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 (A Non-exhaustive List)

With large memory and powerful image and voice processing capability, ESP32-P4 is an ideal choice for IoT devices in the following areas:

- Image and Voice Processing
  - Image recognition
  - Speech recognition
  - IoT Edge
  - IP Camera (IPC)
- Smart Home
  - Light control
  - Smart button
  - Smart plug
  - Indoor positioning
  - Meter reading systems
  - Security systems
  - HVAC systems
- Industrial Automation
  - Industrial robot
  - Human machine interface (HMI)
  - Industrial field bus
  - Asset management
  - Personnel tracking
- Health Care
  - Health monitor
  - Baby monitor
- Consumer Electronics
  - Smartwatch and bracelet
  - Over-the-top (OTT) devices
  - Logger toys and proximity sensing toys
  - Gaming consoles
  - Wireless remote controls
- Smart Agriculture
  - Smart greenhouse
  - Smart irrigation
  - Agricultural robot
  - Livestock tracking
- Retail and Catering
  - POS machines
  - Service robot
- Generic Low-power IoT Sensor Hubs
- Generic Low-power IoT Data Loggers

# Contents

# List of Tables

# List of Figures

# 1 ESP32-P4 Series Comparison

## 1.1 Nomenclature

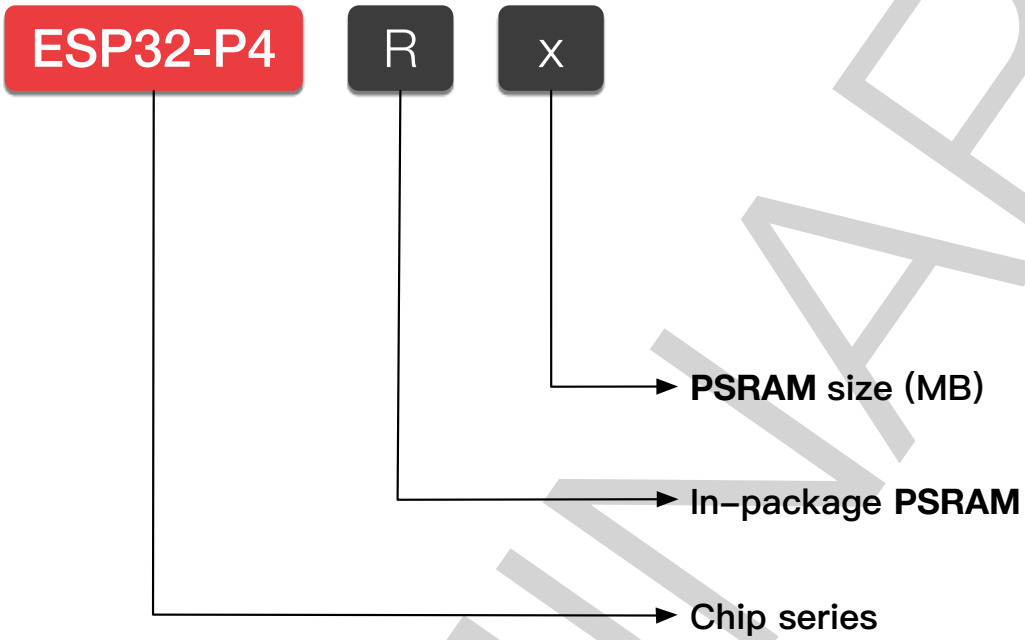



Figure 1-1. ESP32-P4 Series Nomenclature

## 1.2 Comparison

Table 1-1. ESP32-P4 Series Comparison

| Ordering Code | In-package PSRAM | Ambient Temperature (°C) | SPI Voltage | Package |
|---|---|---|---|---|
| ESP32-P4R16 | 16 MB (Octal SPI) | –40 ~ 105 | 1.8 V | QFN104 |
| ESP32-P4R32 | 32 MB (Octal SPI) | –40 ~ 105 | 1.8 V | QFN104 |

**Note:**
Octal SPI of PSRAM supports eight-line and sixteen-line SPI

# 2 Pin Definition

## 2.1 Pin Layout

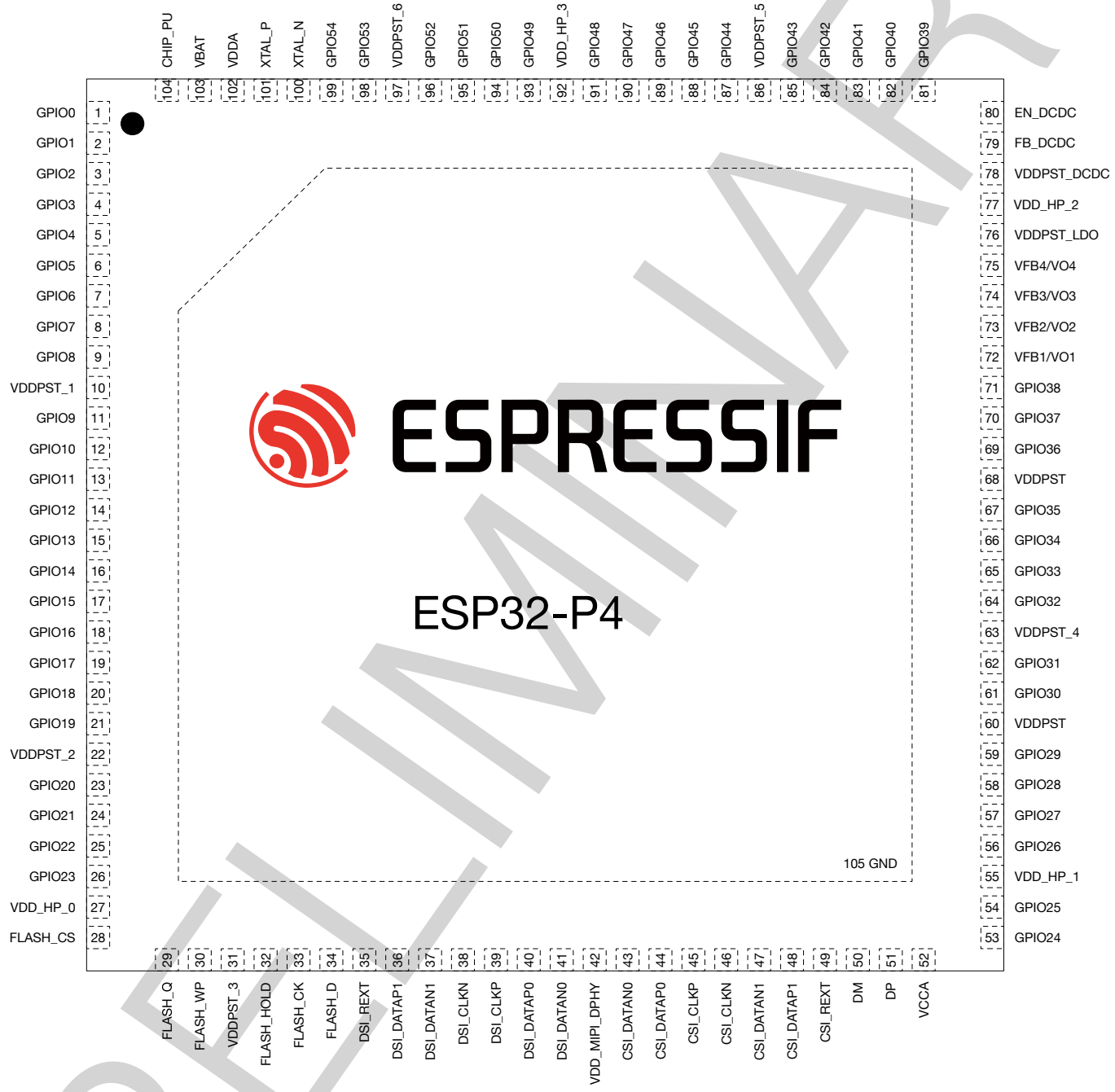



**Figure 2-1. ESP32-P4 Pin Layout (Top View)**

## 2.2 Pin Description

**Table 2-1. Pin Description**

| Name | No. | Type[1] | Power Supply Pin | Function |
|---|---|---|---|---|
| GPIO0 | 1 | I/O/T | VDDPST_1 | **GPIO0**, XTAL_32K_N |
| GPIO1 | 2 | I/O/T | VDDPST_1 | **GPIO1**, XTAL_32K_P |
| GPIO2 | 3 | I/O/T | VDDPST_1 | GPIO2, TOUCH_0, **MTCK** |
| GPIO3 | 4 | I/O/T | VDDPST_1 | GPIO3, TOUCH_1, **MTDI** |
| GPIO4 | 5 | I/O/T | VDDPST_1 | GPIO4, TOUCH_2, **MTMS** |
| GPIO5 | 6 | I/O/T | VDDPST_1 | GPIO5, TOUCH_3, **MTDO** |
| GPIO6 | 7 | I/O/T | VDDPST_1 | **GPIO6**, TOUCH_4, GPSPI SPI2 HOLD |
| GPIO7 | 8 | I/O/T | VDDPST_1 | **GPIO7**, TOUCH_5, GPSPI SPI2 CS |
| GPIO8 | 9 | I/O/T | VDDPST_1 | **GPIO8**, TOUCH_6, GPSPI SPI2 D, UART0_RTS |
| VDDPST_1 | 10 | $P_{IO}$ | — | 3.3 V LP IO power supply |
| GPIO9 | 11 | I/O/T | VDDPST_1 | **GPIO9**, TOUCH_7, GPSPI SPI2 CK, UART0_CTS |
| GPIO10 | 12 | I/O/T | VDDPST_1 | **GPIO10**, TOUCH_8, GPSPI SPI2 Q, UART1_TXD |
| GPIO11 | 13 | I/O/T | VDDPST_1 | **GPIO11**, TOUCH_9, GPSPI SPI2 WP, UART1_RTD |
| GPIO12 | 14 | I/O/T | VDDPST_1 | **GPIO12**, TOUCH_10, UART1_RTS |
| GPIO13 | 15 | I/O/T | VDDPST_1 | **GPIO13**, TOUCH_11, UART0_CTS |
| GPIO14 | 16 | I/O/T | VDDPST_1 | **GPIO14**, TOUCH_12, LP_UART_TXD, LP_UART_TXD |
| GPIO15 | 17 | I/O/T | VDDPST_1 | **GPIO15**, TOUCH_13, LP_UART_RXD |
| GPIO16 | 18 | I/O/T | VDDPST_1 | **GPIO16**, ADC0 |
| GPIO17 | 19 | I/O/T | VDDPST_1 | **GPIO17**, ADC1 |
| GPIO18 | 20 | I/O/T | VDDPST_1 | **GPIO18**, ADC2 |
| GPIO19 | 21 | I/O/T | VDDPST_1 | **GPIO19**, ADC3 |
| VDDPST_2 | 22 | $P_{IO}$ | — | 1.8/3.3 V HP IO power supply |
| GPIO20 | 23 | I/O/T | VDDPST_2 | **GPIO20**, ADC4 |
| GPIO21 | 24 | I/O/T | VDDPST_2 | **GPIO21**, ADC5 |
| GPIO22 | 25 | I/O/T | VDDPST_2 | **GPIO22**, ADC6, DBG_PSRAM_CK |
| GPIO23 | 26 | I/O/T | VDDPST_2 | **GPIO23**, ADC7, 50 MHz reference clock output, DBG_PSRAM_CS |
| VDD_HP_0 | 27 | $P_D$ | — | Digital power supply for HP system (1.1 ~ 1.3 V) |
| FLASH_CS | 28 | O | VDDPST_3 | FLASH_CS |
| FLASH_Q | 29 | I/O/T | VDDPST_3 | FLASH_Q |
| FLASH_WP | 30 | I/O/T | VDDPST_3 | FLASH_WP |
| VDDPST_3 | 31 | $P_{IO}$ | — | Flash IO power supply |
| FLASH_HOLD | 32 | I/O/T | VDDPST_3 | FLASH_HOLD |
| FLASH_CK | 33 | O | VDDPST_3 | FLASH_CK |
| FLASH_D | 34 | I/O/T | VDDPST_3 | FLASH_D |
| DSI_REXT | 35 | I/O/T | VDD_MIPI_DPHY | MIPI DSI PHY 4.02 kΩ external resistor |
| DSI_DATAP1 | 36 | I/O/T | VDD_MIPI_DPHY | MIPI DSI PHY DATAP1 |
| DSI_DATAN1 | 37 | I/O/T | VDD_MIPI_DPHY | MIPI DSI PHY DATAN1 |

**Cont'd on next page**

Table 2-1 – cont'd from previous page

| Name | No. | Type | Power Supply Pin | Function |
|---|---|---|---|---|
| DSI_CLKN | 38 | I/O/T | VDD_MIPI_DPHY | MIPI DSI PHY CLKN |
| DSI_CLKP | 39 | I/O/T | VDD_MIPI_DPHY | MIPI DSI PHY CLKP |
| DSI_DATAP0 | 40 | I/O/T | VDD_MIPI_DPHY | MIPI DSI PHY DATAP0 |
| DSI_DATAN0 | 41 | I/O/T | VDD_MIPI_DPHY | MIPI DSI PHY DATAN0 |
| VDD_MIPI_DPHY | 42 | $P_{IO}$ | — | MIPI DPHY 2.5 V power supply |
| CSI_DATAN0 | 43 | I/O/T | VDD_MIPI_DPHY | MIPI CSI PHY DATAN0 |
| CSI_DATAP0 | 44 | I/O/T | VDD_MIPI_DPHY | MIPI CSI PHY DATAP0 |
| CSI_CLKP | 45 | I/O/T | VDD_MIPI_DPHY | MIPI CSI PHY CLKP |
| CSI_CLKN | 46 | I/O/T | VDD_MIPI_DPHY | MIPI CSI PHY CLKN |
| CSI_DATAN1 | 47 | I/O/T | VDD_MIPI_DPHY | MIPI CSI PHY DATAN1 |
| CSI_DATAP1 | 48 | I/O/T | VDD_MIPI_DPHY | MIPI CSI PHY DATAP1 |
| CSI_REXT | 49 | I/O/T | VDD_MIPI_DPHY | MIPI CSI PHY 4.02 KΩ external resistor |
| DM | 50 | I/O/T | VCCA | USB2 OTG PHY DM |
| DP | 51 | I/O/T | VCCA | USB2 OTG PHY DP |
| VCCA | 52 | $P_{IO}$ | — | USB2 PHY PHY 3.3 V power supply |
| GPIO24 | 53 | I/O/T | VDDPST_4 | **GPIO24**, USB1P1_N0 |
| GPIO25 | 54 | I/O/T | VDDPST_4 | **GPIO25**, USB1P1_P0 |
| VDD_HP_1 | 55 | $P_D$ | — | Digital power supply for HP system (1.1 ~ 1.3 V) |
| GPIO26 | 56 | I/O/T | VDDPST_4 | **GPIO26**, USB1P1_N1 |
| GPIO27 | 57 | I/O/T | VDDPST_4 | **GPIO27**, USB1P1_P1 |
| GPIO28 | 58 | I/O/T | VDDPST_4 | **GPIO28**, GPSPI SPI2 CS, EMAC PHY RXDV, DBG_PSRAM_D |
| GPIO29 | 59 | I/O/T | VDDPST_4 | **GPIO29**, GPSPI SPI2 D, EMAC PHY RXD0, DBG_PSRAM_Q |
| VDDPST | 60 | $P_{IO}$ | — | PSRAM 1.8 V power supply |
| GPIO30 | 61 | I/O/T | VDDPST_4 | **GPIO30**, GPSPI SPI2 CK, EMAC PHY RXD1, DBG_PSRAM_WP |
| GPIO31 | 62 | I/O/T | VDDPST_4 | **GPIO31**, GPSPI SPI2 Q, EMAC PHY RXER, DBG_PSRAM_HOLD |
| VDDPST_4 | 63 | $P_{IO}$ | — | 1.8/3.3 V HP IO power supply |
| GPIO32 | 64 | I/O/T | VDDPST_4 | **GPIO32**, I3CMST_SCL, GPSPI SPI2 HOLD, EMAC RMII CLK, DBG_PSRAM_DQ4 |
| GPIO33 | 65 | I/O/T | VDDPST_4 | **GPIO33**, I3CMST_SDA, GPSPI SPI2 WP, EMAC PHY TXEN, DBG_PSRAM_DQ5 |
| GPIO34 | 66 | I/O/T | VDDPST_4 | **GPIO34**, GPSPI SPI2 IO4, EMAC PHY TXD0, DBG_PSRAM_DQ6 |
| GPIO35 | 67 | I/O/T | VDDPST_4 | **GPIO35**, GPSPI SPI2 IO5, EMAC PHY TXD1, DBG_PSRAM_DQ7 |
| VDDPST | 68 | $P_{IO}$ | — | PSRAM 1.8 V power supply |
| GPIO36 | 69 | I/O/T | VDDPST_4 | **GPIO36**, GPSPI SPI2 IO6, EMAC PHY TXER, DBG_PSRAM_DQS0 |
| GPIO37 | 70 | I/O/T | VDDPST_4 | GPIO37, **UART0_TXD**, GPSPI SPI2 IO7 |

**Cont'd on next page**

Table 2-1 – cont'd from previous page

| Name | No. | Type | Power Supply Pin | Function |
|---|---|---|---|---|
| GPIO38 | 71 | I/O/T | VDDPST_4 | GPIO38, **UART0_RXD**, GPSPI SPI2 DQS |
| VFB/VO1 | 72 | $P_O$ | — | External power supply pin, outputting 0.1 A current |
| VFB/VO2 | 73 | $P_O$ | — | External power supply pin, outputting 0.1 A current |
| VFB/VO3 | 74 | $P_O$ | — | External power supply pin, outputting 0.1 A current |
| VFB/VO4 | 75 | $P_O$ | — | External power supply pin, outputting 0.2 A current |
| VDDPST_LDO | 76 | $P_O$ | — | 3.3 V external LDO power supply pin |
| VDD_HP_2 | 77 | $P_D$ | — | Digital power supply for HP system (1.1 ~ 1.3 V) |
| VDDPST_DCDC | 78 | $P_{DCDC}$ | — | 3.3 V external DCDC power supply pin |
| FB_DCDC | 79 | $P_{DCDC}$ | — | DC/DC feedback power supply pin |
| EN_DCDC | 80 | O | VDDPST_DCDC | DC/DC enable |
| GPIO39 | 81 | I/O/T | VDDPST_5 | **GPIO39**, SD1_CDATA0, reference clock of 50 MHz output, DBG_PSRAM_DQ8 |
| GPIO40 | 82 | I/O/T | VDDPST_5 | **GPIO40**, SD1_CDATA1, EMAC PHY TXEN, DBG_PSRAM_DQ8 |
| GPIO41 | 83 | I/O/T | VDDPST_5 | **GPIO41**, SD1_CDATA2, EMAC PHY TXD0, DBG_PSRAM_DQ9 |
| GPIO42 | 84 | I/O/T | VDDPST_5 | **GPIO42**, SD1_CDATA3, EMAC PHY TXD1, DBG_PSRAM_DQ10 |
| GPIO43 | 85 | I/O/T | VDDPST_5 | **GPIO43**, SD1_CCLK, EMAC PHY TXER, DBG_PSRAM_DQ11 |
| VDDPST_5 | 86 | $P_{IO}$ | — | 1.8/3.3 V HP IO power supply |
| GPIO44 | 87 | I/O/T | VDDPST_5 | **GPIO44**, SD1_CCMD, EMAC RMII CLK, DBG_PSRAM_DQ13 |
| GPIO45 | 88 | I/O/T | VDDPST_5 | **GPIO45**, SD1_CDATA4, EMAC PHY RXDV, DBG_PSRAM_DQ14 |
| GPIO46 | 89 | I/O/T | VDDPST_5 | **GPIO46**, SD1_CDATA5, EMAC PHY RXD0, DBG_PSRAM_DQ15 |
| GPIO47 | 90 | I/O/T | VDDPST_5 | **GPIO47**, SD1_CDATA6, EMAC PHY RXD1, DBG_PSRAM_DQS1 |
| GPIO48 | 91 | I/O/T | VDDPST_5 | **GPIO48**, SD1_CDATA7, EMAC PHY RXER |
| VDD_HP_3 | 92 | $P_D$ | — | Digital power supply for HP system (1.1 ~ 1.3 V) |
| GPIO49 | 93 | I/O/T | VDDPST_6 | **GPIO49**, ADC8, EMAC PHY TXEN, DBG_FLASH_CS |
| GPIO50 | 94 | I/O/T | VDDPST_6 | **GPIO50**, ADC9, EMAC RMII CLK, DBG_FLASH_Q |
| GPIO51 | 95 | I/O/T | VDDPST_6 | **GPIO51**, ADC10, EMAC PHY RXDV, DBG_FLASH_WP |
| GPIO52 | 96 | I/O/T | VDDPST_6 | **GPIO52**, ADC11, EMAC PHY RXD0, DBG_FLASH_HOLD |
| VDDPST_6 | 97 | $P_{IO}$ | — | 1.8/3.3 V HP IO power supply |
| GPIO53 | 98 | I/O/T | VDDPST_6 | **GPIO53**, ADC12, EMAC PHY RXD1, DBG_FLASH_CK |

**Cont'd on next page**

Table 2-1 – cont'd from previous page

| Name | No. | Type | Power Supply Pin | Function |
|---|---|---|---|---|
| GPIO54 | 99 | I/O/T | VDDPST_6 | **GPIO54**, ADC13, EMAC PHY RXER, DBG_FLASH_D |
| XTAL_N | 100 | — | — | External crystal output |
| XTAL_P | 101 | — | — | External crystal input |
| VDDA | 102 | $P_A$ | — | Analog power supply (3.3 V) |
| VBAT | 103 | $P_A$ | — | Analog power supply or battery power supply (3.3 V) |
| CHIP_PU | 104 | I | — | High: on, enables the chip (powered up).<br>Low: off, the chip powers off (powered down).<br>Note: Do not leave the CHIP_PU pin floating. |
| GND | 105 | — | — | Ground |

[1] $P_A$: analog power supply; $P_D$: digital power supply; $P_{IO}$: IO pin power supply; $P_O$: output power supply; I: input; O: output; T: high impedance.

## 2.3 Power Scheme

Table 2-2. Description of ESP32-P4 Power Supply Pins

| Type | Pin Name | Power Supply to |
|---|---|---|
| $P_{IO}$ | VDDPST_1 | Group0 IO[1] |
| | VDDPST_2 | Group1 IO[1] |
| | VDDPST_3 | Group2 IO[1] |
| | VDDPST_4 | Group3 IO[1] |
| | VDDPST_5 | Group4 IO[1] |
| | VDDPST_6 | Group5 IO[1] |
| | VDD_MIPI_DPHY | MIPI IO and PHY |
| | VCCA | USB IO and PHY |
| $P_A$ | VBAT | Analog System |
| | VDDPST_LDO | |
| | VDDPST_DCDC | |
| | VDDPST | |
| | VDDA | |
| $P_D$ | VDD_HP_0/1/2/3/4 | Digital System |

[1] For a complete list of IO pins powered by VDDPST_1 ~ VDDPST_6, see Table 2-1.

## 2.4 Strapping Pins

At each startup or reset, a chip requires some initial configuration parameters, such as in which boot mode to load the chip, etc. These parameters are passed over via strapping pins. After reset, the strapping pins work as regular pins.

ESP32-P4 has the following parameters controlled by the given strapping pins at chip reset:

- **Chip boot mode** – GPIO35, GPIO36, GPIO37, and GPIO38

- **JTAG signal source** – GPIO34

GPIO35 is connected to the chip's internal weak pull-up resistor at chip reset. If GPIO35 has no external connection or the connected external line is in the high impedance state, this internal weak pull-up resistor determines the default bit value of GPIO35.

**Table 2-3. Default Configuration of Strapping Pins**

| Strapping Pin | Default Config | Bit Value |
|---|---|---|
| **GPIO34** | Floating | – |
| **GPIO35** | Pull-up | 1 |
| **GPIO36** | Floating | – |
| **GPIO37** | Floating | – |
| **GPIO38** | Floating | – |

To change the bit values, the strapping pins should be connected to external pull-up/pull-down resistors. If ESP32-P4 is used as a device by a host MCU, the voltages of strapping pins 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 IOs after reset.

### 2.4.1 Chip Boot Mode Control

After the reset is released, the combination of GPIO35 ~ GPIO38 controls the boot mode. See Table 2-4 Boot Mode Control.

**Table 2-4. Boot Mode Control**

| Boot Mode | GPIO35 | GPIO36 | GPIO37 | GPIO38 |
|---|---|---|---|---|
| Default Config | 1 (Pull-up) | – (Floating) | – (Floating) | – (Floating) |
| **SPI Boot** | 1 | Any value | Any value | Any value |
| **Joint Download Boot**[1] | 0 | 1 | Any value | Any value |
| **SPI Download Boot**[1] | 0 | 0 | 0 | 1 |

[1] Joint Download mode supports the following download methods:
- USB Serial/JTAG Download Boot
- USB OTG 2.0 Download Boot
- UART Download Boot

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

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

### 2.4.2 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-up/pull-down resistors and thus its value must be controlled by the external circuit that cannot be in a high impedance state.

As Table 2-5 shows, GPIO34 is used in combination with EFUSE_DIS_PAD_JTAG, EFUSE_DIS_USB_JTAG, and EFUSE_JTAG_SEL_ENABLE.

**Table 2-5. JTAG Signal Source Control**

| eFuse 1[a] | eFuse 2[b] | eFuse 3[c] | GPIO34 | JTAG Signal Source |
|---|---|---|---|---|
| 0 | 0 | 0 | Ignored | USB Serial/JTAG Controller |
| | | 1 | 0 | JTAG pins MTDI, MTCK, MTMS, and MTDO |
| | | | 1 | USB Serial/JTAG Controller |
| 0 | 1 | Ignored | Ignored | JTAG pins MTDI, MTCK, MTMS, and MTDO |
| 1 | 0 | Ignored | Ignored | USB Serial/JTAG Controller |
| 1 | 1 | Ignored | Ignored | JTAG is disabled |

[a] eFuse 1: EFUSE_DIS_PAD_JTAG
[b] eFuse 2: EFUSE_DIS_USB_JTAG
[c] eFuse 3: EFUSE_JTAG_SEL_ENABLE

# 3 Functional Description

This chapter describes the functions of ESP32-P4.

## 3.1 CPU and Memory

### 3.1.1 HP 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

### 3.1.2 LP 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

### 3.1.3 Processor Instruction Extensions (PIE)

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:

- 8 128-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

### 3.1.4 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

### 3.1.5 External Flash and SRAM

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

The external flash and RAM can be mapped into the CPU instruction memory space and read-only data memory space. The external RAM can also be mapped into the CPU data memory space. ESP32-P4 supports up to 64 MB of external flash and 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, but only 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.

### 3.1.6 Address Mapping Structure

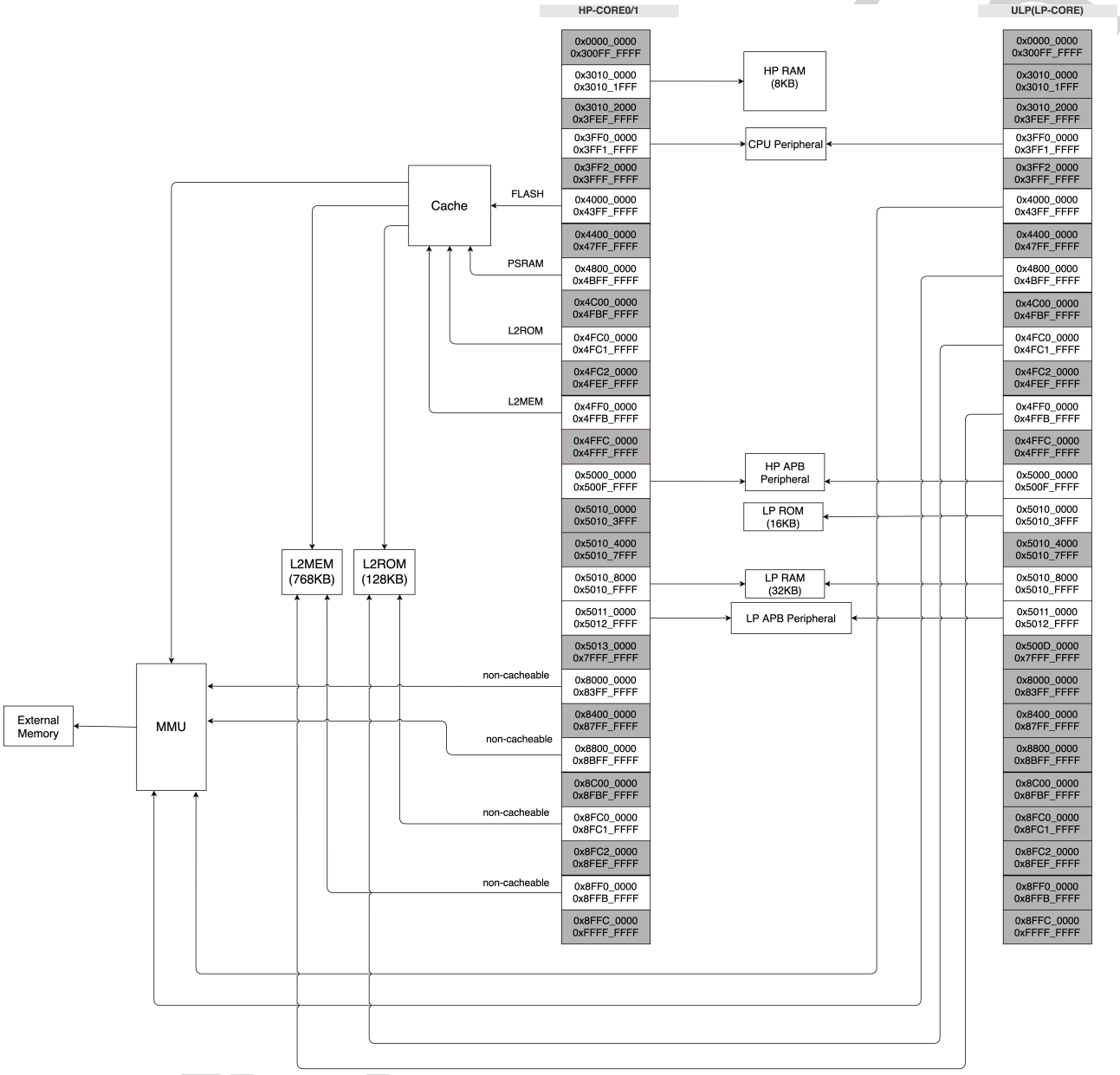



**Figure 3-1. Address Mapping Structure**

**Note:**
The memory space with gray background is not available for use.

### 3.1.7 Cache

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

- 16 KB of I1 instruction cache, 64 B of block size, four-way set associative

- 64 KB of I1 data cache, 64 B of block size, four-way set associative, supporting two writing strategies write-through and write-back
- 128 KB/256 KB/512 KB of I2 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

### 3.1.8 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

## 3.2 System Clocks

### 3.2.1 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.

### 3.2.2 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

### 3.2.3 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.

## 3.3 Analog Peripherals

### 3.3.1 Analog-to-Digital Converter (ADC)

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

For GPIOs assigned to ADC, please refer to Table 3-3.

### 3.3.2 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.

### 3.3.3 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.

For GPIOs assigned to the touch sensor, please refer to Table 3-3.

### 3.3.4 Analog PAD Voltage Comparator

ESP32-P4 provides two groups of analog PAD 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.

### 3.3.5 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.

## 3.4 Digital Peripherals

### 3.4.1 General Purpose Input/Output Interface (GPIO)

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.

Table 3-1 shows the LP IO MUX functions of each pin, and Table 3-2 shows the HP IO MUX functions of each pin.

**Table 3-1. LP IO MUX Pin Functions**

| Name | No. | Function 0 | Function 1 |
|---|---|---|---|
| GPIO0 | 1 | LPGPIO0 | LPGPIO0 |
| GPIO1 | 2 | LPGPIO1 | LPGPIO1 |
| GPIO2 | 3 | LPGPIO2 | LPGPIO2 |
| GPIO3 | 4 | LPGPIO3 | LPGPIO3 |
| GPIO4 | 5 | LPGPIO4 | LPGPIO4 |
| GPIO5 | 6 | LPGPIO5 | LPGPIO5 |
| GPIO6 | 7 | LPGPIO6 | LPGPIO6 |
| GPIO7 | 8 | LPGPIO7 | LPGPIO7 |
| GPIO8 | 9 | LPGPIO8 | LPGPIO8 |
| GPIO9 | 11 | LPGPIO9 | LPGPIO9 |
| GPIO10 | 12 | LPGPIO10 | LPGPIO10 |
| GPIO11 | 13 | LPGPIO11 | LPGPIO11 |
| GPIO12 | 14 | LPGPIO12 | LPGPIO12 |
| GPIO13 | 15 | LPGPIO13 | LPGPIO13 |
| GPIO14 | 16 | LPUTXD | LPGPIO14 |
| GPIO15 | 17 | LPURTD | LPGPIO15 |

Table 3-2. HP IO MUX Pin Functions

| Name | No. | Function 0 | Function 1 | Function 2 | Function 3 | Reset | Notes |
|---|---|---|---|---|---|---|---|
| GPIO0 | 1 | HPGPIO0 | HPGPIO0 | - | - | 0 | R |
| GPIO1 | 2 | HPGPIO1 | HPGPIO1 | - | - | 0 | R |
| GPIO2 | 3 | MTCK | HPGPIO2 | - | - | 1* | R |
| GPIO3 | 4 | MTDI | HPGPIO3 | - | - | 1 | R |
| GPIO4 | 5 | MTMS | HPGPIO4 | - | - | 1 | R |
| GPIO5 | 6 | MTDO | HPGPIO5 | - | - | 1 | R |
| GPIO6 | 7 | HPGPIO6 | HPGPIO6 | - | SPI2HD | 0 | R |
| GPIO7 | 8 | HPGPIO7 | HPGPIO7 | - | SPI2CS | 0 | R |
| GPIO8 | 9 | HPGPIO8 | HPGPIO8 | U0RTS | SPI2D | 0 | R |
| GPIO9 | 11 | HPGPIO9 | HPGPIO9 | U0CTS | SPI2CK | 0 | R |
| GPIO10 | 12 | HPGPIO10 | HPGPIO10 | U1TXD | SPI2Q | 0 | R |
| GPIO11 | 13 | HPGPIO11 | HPGPIO11 | U1RXD | SPI2WP | 0 | R |
| GPIO12 | 14 | HPGPIO12 | HPGPIO12 | U1RTS | - | 0 | R |
| GPIO13 | 15 | HPGPIO13 | HPGPIO13 | U1CTS | - | 0 | R |
| GPIO14 | 16 | HPGPIO14 | HPGPIO14 | — | - | 0 | R |
| GPIO15 | 17 | HPGPIO15 | HPGPIO15 | — | - | 0 | R |
| GPIO16 | 18 | HPGPIO16 | HPGPIO16 | — | - | 0 | R |
| GPIO17 | 19 | HPGPIO17 | HPGPIO17 | — | - | 0 | R |
| GPIO18 | 20 | HPGPIO18 | HPGPIO18 | — | - | 0 | R |
| GPIO19 | 21 | HPGPIO19 | HPGPIO19 | — | - | 0 | R |
| GPIO20 | 23 | HPGPIO20 | HPGPIO20 | — | - | 0 | R |
| GPIO21 | 24 | HPGPIO21 | HPGPIO21 | — | - | 0 | R |
| GPIO22 | 25 | HPGPIO22 | HPGPIO22 | — | - | 0 | R |
| GPIO23 | 26 | HPGPIO23 | HPGPIO23 | — | REF50M | 0 | R |
| GPIO24 | 53 | HPGPIO24 | HPGPIO24 | — | - | 0 | R, USB |
| GPIO25 | 54 | HPGPIO25 | HPGPIO25 | — | - | 3* | R, USB |
| GPIO26 | 56 | HPGPIO26 | HPGPIO26 | — | - | 0 | R |
| GPIO27 | 57 | HPGPIO27 | HPGPIO27 | — | - | 0 | R |
| GPIO28 | 58 | HPGPIO28 | HPGPIO28 | SPI2CS | RMII_RXDV | 0 | — |
| GPIO29 | 59 | HPGPIO29 | HPGPIO29 | SPI2D | RMII_RXD0 | 0 | — |
| GPIO30 | 61 | HPGPIO30 | HPGPIO30 | SPI2CK | RMII_RXD1 | 0 | — |
| GPIO31 | 62 | HPGPIO31 | HPGPIO31 | SPI2Q | RMII_RXER | 0 | — |
| GPIO32 | 64 | HPGPIO32 | HPGPIO32 | SPI2HD | RMII_CLK | 0 | — |
| GPIO33 | 65 | HPGPIO33 | HPGPIO33 | SPI2WP | RMII_TXEN | 0 | — |
| GPIO34 | 66 | HPGPIO34 | HPGPIO34 | SPI2D4 | RMII_TXD0 | 0 | — |
| GPIO35 | 67 | HPGPIO35 | HPGPIO35 | SPI2D5 | RMII_TXD1 | 0 | — |
| GPIO36 | 69 | HPGPIO36 | HPGPIO36 | SPI2D6 | RMII_TXER | 0 | — |
| GPIO37 | 70 | U0TXD | HPGPIO37 | SPI2D7 | - | 5 | — |
| GPIO38 | 71 | U0RXD | HPGPIO38 | SPI2DQS | - | 0 | — |
| GPIO39 | 81 | SD1_D0 | HPGPIO39 | — | REF50M | 0 | — |
| GPIO40 | 82 | SD1_D1 | HPGPIO40 | — | RMII_TXEN | 0 | — |

**Cont'd on next page**

Table 3-2 – cont'd from previous page

| Name | No. | Function 0 | Function 1 | Function 2 | Function 3 | Reset | Notes |
|---|---|---|---|---|---|---|---|
| GPIO41 | 83 | SD1_D2 | HPGPIO41 | — | RMII_TXD0 | 0 | — |
| GPIO42 | 84 | SD1_D3 | HPGPIO42 | — | RMII_TXD1 | 0 | — |
| GPIO43 | 85 | SD1_CK | HPGPIO43 | — | RMII_TXER | 4 | — |
| GPIO44 | 87 | SD1_CMD | HPGPIO44 | — | RMII_CLK | 0 | — |
| GPIO45 | 88 | SD1_D4 | HPGPIO45 | — | RMII_RXDV | 0 | — |
| GPIO46 | 89 | SD1_D5 | HPGPIO46 | — | RMII_RXD0 | 0 | — |
| GPIO47 | 90 | SD1_D6 | HPGPIO47 | — | RMII_RXD1 | 0 | — |
| GPIO48 | 91 | SD1_D7 | HPGPIO48 | — | RMII_RXER | 0 | — |
| GPIO49 | 93 | HPGPIO49 | HPGPIO49 | — | RMII_TXEN | 0 | R |
| GPIO50 | 94 | HPGPIO50 | HPGPIO50 | — | RMII_CLK | 0 | R |
| GPIO51 | 95 | HPGPIO51 | HPGPIO51 | — | RMII_RXDV | 0 | R |
| GPIO52 | 96 | HPGPIO52 | HPGPIO52 | — | RMII_RXD0 | 0 | R |
| GPIO53 | 98 | HPGPIO53 | HPGPIO53 | — | RMII_RXD1 | 0 | R |
| GPIO54 | 99 | HPGPIO54 | HPGPIO54 | — | RMII_RXER | 0 | R |

### Reset

The default configuration of each pin after reset:

- **0** - input disabled, in high impedance state (IE = 0)
- **1** - input disabled, in high impedance state (IE = 1)
- **1*** - When the value of eFuse bit EFUSE_DIS_PAD_JTAG is
  0 (default), input enabled, pull-up resistor enabled (IE = 1, WPU = 1)
  1, input enabled, in high impedance state (IE = 1)
- **3*** - input enabled, pull-up resistor enabled (IE = 1, WPU = 0, USB_WPU = 1). See details in Notes.
- **4** - input disabled. Output is controlled by the peripheral of Function 0, and the default output is 0.
- **5** - input disabled. Output is controlled by the peripheral of Function 0, and the default output is 1.

We recommend pulling high or low GPIO pins in high impedance state to avoid unnecessary power consumption. You may add pull-up and pull-down resistors in your PCB design, or enable internal pull-up and pull-down resistors during software initialization.

### GPIO Input Mode

The input function of GPIO can be configured as hysteresis or normal mode:

- **Hysteresis mode**: There is a small delay between the rising and falling edges of the GPIO input signal. During this delay period, the GPIO ignores any input changes to eliminate the impact of unstable electrical signals. In the hysteresis mode, the threshold voltage for flipping between high and low levels of GPIO input depends on the direction of level flipping. Specifically, the voltage threshold for flipping from high to low level is slightly lower than the voltage threshold for flipping from low to high level.
- **Normal mode**: The threshold voltage for flipping between high and low levels of GPIO input is independent of the direction of level flipping. In other words, the voltage threshold for flipping from high to low level is the same as the voltage threshold for flipping from low to high level.

**Notes**

- **R** - These pins have analog functions.
- **USB** - The pull-up value of a USB pin is controlled by the pin's pull-up value together with the USB pull-up value. If any of the two pull-up values is 1, the pin's pull-up resistor will be enabled. The pull-up resistors of USB pins are controlled by the USB_SERIAL_JTAG_DP_PULLUP bit.

### 3.4.2 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)

### 3.4.3 Pixel-Processing Accelerator

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

- SR 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 4-bit integer part and 8-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

### 3.4.4 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), Lens Shading Correction (LSC), 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)

### 3.4.5 H264 Encoder

ESP32-P4 contains a baseline H264 encoder, which has the following features:

- YUV420 progressive video with the maximum encoding performance of 1920×1088@24fps
- 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 1920x1088@24fps)
- 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.

### 3.4.6 Serial Peripheral Interface (SPI)

ESP32-P4 features four SPI interfaces, i.e. FLASH SPI, PSRAM SPI, SPI2, and SPI3. FLASH SPI and PSRAM SPI can be configured to operate in SPI memory mode, while SPI2 and SPI3 can be configured to operate in general-purpose SPI mode.

- **SPI Memory mode**

  In SPI memory mode, FLASH SPI and PSRAM SPI interfaces connect with in-package/out-package NOR FLASH and PSRAM respectively, which are used as the external memory of CPU. 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.

- **General-purpose SPI (GP-SPI) mode**

  When SPI2 and SPI3 act as general-purpose SPIs, they can operate in master and slave modes. SPI2 and SPI3 support two-line full-duplex communication and single-/two-/four-line half-duplex communication in both master and slave modes. In addition, SPI2 supports eight-line communication. The host's clock frequency is configurable. Data is transferred in bytes. The clock polarity (CPOL) and phase (CPHA) are also configurable. SPI2 and SPI3 interfaces can be connected to PDMA.

  - In master mode, the clock frequency is 80 MHz at most, and the four modes of SPI transfer format are supported.
  - In slave mode, the clock frequency is 60 MHz at most, and the four modes of SPI transfer format are also supported.

For GPIOs assigned to SPI, please refer to Table 3-3.

### 3.4.7 LP SPI Interface

ESP32-P4 includes an LP SPI interface, which is a low-speed SPI interface. The LP SPI interface has the following features:

- configurable to operate in master or slave mode
- two-line full-duplex communication and single-/two-line half-duplex communication in both master and slave modes
- one CS line in master mode
- host's clock frequency is configurable in master mode

- data is transferred in bytes
- configurable clock polarity (CPOL) and phase (CPHA)
- wakeup on start bit and predefined sequence detection in slave mode

For GPIOs assigned to LP SPI Interface, please refer to Table 3-3.

### 3.4.8 Universal Asynchronous Receiver Transmitter (UART)

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.

For GPIOs assigned to UART, please refer to Table 3-3.

### 3.4.9 LP UART Interface

ESP32-P4 has one LP UART interface, which provides hardware flow control (CTS and RTS signals) and software flow control (XON and XOFF).

LP UART only supports asynchronous communication (RS232) at a speed of up to 1.25 Mbps.

For GPIOs assigned to LP UART Interface, please refer to Table 3-3.

### 3.4.10 I3C Interface

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

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
- DMA transfer

For GPIOs assigned to I3C Interface, please refer to Table 3-3.

### 3.4.11 I2C Interface

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

Users can configure instruction registers to control the I2C interfaces for more flexibility.

For GPIOs assigned to I2S Interface, please refer to Table 3-3.

### 3.4.12 LP I2C Interface

ESP32-P4 has one LP I2C bus interface which can only be used for the master mode. The LP I2C interface supports:

- 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

Users can configure instruction registers to control the LP I2C interface for more flexibility.

For GPIOs assigned to LP I2C Interface, please refer to Table 3-3.

### 3.4.13 I2S Interface

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.

For GPIOs assigned to I2S Interface, please refer to Table 3-3.

### 3.4.14 LP I2S Interface

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

For GPIOs assigned to LP I2S Interface, please refer to Table 3-3.

### 3.4.15 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.

For GPIOs assigned to RMT, please refer to Table 3-3.

### 3.4.16 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

For GPIOs assigned to LED PWM, please refer to Table 3-3.

### 3.4.17 Motor Control PWM (MCPWM)

ESP32-P4 integrates two MCPWMs that can be used to drive digital motors and smart light. Every MCPWM has a clock divider (prescaler), three PWM timers, three PWM operators, 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.

For GPIOs assigned to MCPWM, please refer to Table 3-3.

### 3.4.18 Pulse Count Controller (PCNT)

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

  For GPIOs assigned to PCNT, please refer to Table 3-3.

### 3.4.19 TWAI® Controller

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

For GPIOs assigned to TWAI®, please refer to Table 3-3.

### 3.4.20 USB 2.0 OTG Full-Speed Interface

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 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.

For GPIOs assigned to USB 2.0 OTG Full-Speed Interface, please refer to Table 3-3.

### 3.4.21 USB 2.0 OTG High-Speed Interface

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
- HNP and SRP as A-device or B-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.

For GPIOs assigned to USB 2.0 OTG High-Speed Interface, please refer to Table 3-3.

### 3.4.22 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

For GPIOs assigned to USB Serial/JTAG Controller, please refer to Table 3-3.

### 3.4.23 Ethernet MAC

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

For GPIOs assigned to Ethernet MAC, please refer to Table 3-3.

### 3.4.24 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

For GPIOs assigned to SD/MMC Host Controller, please refer to Table 3-3.

### 3.4.25 MIPI CSI Interface

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

For GPIOs assigned to MIPI CSI, please refer to Table 3-3.

### 3.4.26 MIPI DSI Interface

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

For GPIOs assigned to MIPI DSI, please refer to Table 3-3.

### 3.4.27 LCD Interface

ESP32-P4 supports 8-bit ~ 24-bit parallel RGB, I8080, and MOTO6800 interfaces. These interfaces operate at 40 MHz or lower, and support conversion among RGB565, YUV422, YUV420, and YUV411.

For GPIOs assigned to LCD Interface, please refer to Table 3-3.

### 3.4.28 CAM Interface

ESP32-P4 supports an 8-bit ~ 16-bit DVP image sensor, with clock frequency of up to 40 MHz. The camera interface supports conversion among RGB565, YUV422, YUV420, and YUV411.

For GPIOs assigned to CAM Interface, please refer to Table 3-3.

### 3.4.29 General DMA 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

### 3.4.30 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

### 3.4.31 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

### 3.4.32 Bit-Scrambler

The Bit-Scrambler controller includes transmit and receive channels, located in the transmit and receive paths of GDMA, respectively. The Bit-Scrambler 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.

### 3.4.33 SoC Event Task Matrix (ETM)

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

### 3.4.34 Parallel IO (PARLIO) 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

For GPIOs assigned to PARLIO, please refer to Table 3-3.

### 3.4.35 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.

## 3.5 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.

## 3.6 Timers and Watchdogs

### 3.6.1 General Purpose Timers

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

### 3.6.2 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

### 3.6.3 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.

### 3.6.4 LP General Purpose 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

### 3.6.5 Super Watchdog Timer

ESP32-P4 integrates an analog super watchdog timer. The analog super watchdog timer can trigger a reset of the digital system in the event of an abnormality in the digital system that cannot be self-detected.

## 3.7 Cryptography/Security Components

### 3.7.1 AES Accelerator (AES)

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

The following functionality is supported:

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

### 3.7.2 ECC Accelerator (ECC)

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

ESP32-P4's ECC accelerator can complete various calculations based on different elliptic curves, thus accelerating the ECC algorithm and ECC-derived algorithms (such as ECDSA).

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

- two different elliptic curves, namely P-192 and P-256 defined in FIPS 186-3
- six working modes
- interrupt upon completion of calculation

### 3.7.3 HMAC Accelerator (HMAC)

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

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)

### 3.7.4 RSA Accelerator (RSA)

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

The following functionality is supported:

- large-number modular exponentiation with two optional acceleration options
- large-number modular multiplication, up to 3072 bits
- large-number multiplication, with operands up to 1536 bits
- operands of different lengths
- interrupt on completion of computation

### 3.7.5 SHA Accelerator (SHA)

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

### 3.7.6 Digital Signature (DS)

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 3072 bits
- encrypted private key data, only decryptable by DS module
- SHA-256 digest to protect private key data against tampering by an attacker

### 3.7.7 Elliptic Curve Digital Signature Algorithm (ECDSA)

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

ESP32-P4's ECDSA accelerator provides a secure and efficient environment for computing ECDSA signatures. It 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.

### 3.7.8 External Memory Encryption and Decryption (XTS_AES)

The ESP32-P4 integrates an External Memory Encryption and Decryption module that complies with the XTS-AES standard algorithm specified in IEEE Std 1619-2007, providing security for users' application code and data stored in the external memory (flash). 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

### 3.7.9 Secure Boot

ESP32-P4 integrates a hardware Secure Boot (V2) verification scheme based on either RSA-PSS or ECDSA algorithm. Secure Boot feature ensures that only authenticated software signed by a trusted entity can execute on ESP32-P4. Following are a few highlights of this feature:

- RSA-PSS (3072-bit key) or ECDSA P192/P256 curve-based signing scheme
- up to 3 secure signing keys are supported with independent digest slots in the eFuse
- key revocation support through eFuse

### 3.7.10 Access Permission Management (APM)

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

### 3.7.11 True Random Number Generator (TRNG)

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.

### 3.7.12 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.

## 3.8 Peripheral Pin Configurations

**Table 3-3. Peripheral Pin Configurations**

| Interface | Signal | Pin | Function |
|---|---|---|---|
| ADC | ADC1_CH0 | GPIO16 | 12-bit SAR ADC |
| | ADC1_CH1 | GPIO17 | |
| | ADC1_CH2 | GPIO18 | |
| | ADC1_CH3 | GPIO19 | |
| | ADC1_CH4 | GPIO20 | |
| | ADC1_CH5 | GPIO21 | |
| | ADC1_CH6 | GPIO22 | |
| | ADC1_CH7 | GPIO23 | |
| | ADC2_CH0 | GPIO49 | |
| | ADC2_CH1 | GPIO50 | |
| | ADC2_CH2 | GPIO51 | |
| | ADC2_CH3 | GPIO52 | |
| | ADC2_CH4 | GPIO53 | |
| | ADC2_CH5 | GPIO54 | |
| TOUCH | TOUCH_CH0 | GPIO2 | 14-channel TOUCH |
| | TOUCH_CH1 | GPIO3 | |
| | TOUCH_CH2 | GPIO4 | |
| | TOUCH_CH3 | GPIO5 | |
| | TOUCH_CH4 | GPIO6 | |
| | TOUCH_CH5 | GPIO7 | |
| | TOUCH_CH6 | GPIO8 | |
| | TOUCH_CH7 | GPIO9 | |
| | TOUCH_CH8 | GPIO10 | |
| | TOUCH_CH9 | GPIO11 | |
| | TOUCH_CH10 | GPIO12 | |
| | TOUCH_CH11 | GPIO13 | |
| | TOUCH_CH12 | GPIO14 | |
| | TOUCH_CH13 | GPIO15 | |
| JTAG | MTCK | GPIO2 | JTAG for software debugging |
| | MTDI | GPIO3 | |
| | MTMS | GPIO4 | |
| | MTDO | GPIO5 | |
| UART | U0RXD_in | GPIO38 | 5 UART channels with hardware flow control and GDMA |
| | U0CTS_in | Any GPIO pins | |
| | U0DSR_in | | |
| | U0TXD_out | GPIO37 | |
| | U0RTS_out | Any GPIO pins | |
| | U0DTR_out | | |
| | U1RXD_in | | |
| | U1CTS_in | | |
| | U1DSR_in | | |
| | U1TXD_out | | |
| | U1RTS_out | | |

| Interface | Signal | Pin | Function |
|---|---|---|---|
| | U1DTR_out | | |
| | U2RXD_in | | |
| | U2CTS_in | | |
| | U2DSR_in | | |
| | U2TXD_out | | |
| | U2RTS_out | | |
| | U2DTR_out | | |
| | U3RXD_in | | |
| | U3CTS_in | | |
| | U3DSR_in | | |
| | U3TXD_out | | |
| | U3RTS_out | | |
| | U3DTR_out | | |
| | U4RXD_in | | |
| | U4CTS_in | | |
| | U4DSR_in | | |
| | U4TXD_out | | |
| | U4RTS_out | | |
| | U4DTR_out | | |
| LP UART | U0RXD_in | Any GPIO pins | 5 UART channels with hardware flow control |
| | LPUCTS_in | | |
| | LPUDSR_in | | |
| | LPUTXD_out | | |
| | LPURTS_out | | |
| | LPUDTR_out | | |
| | LPURXD_in | | |
| I2C | I2CEXT0_SCL_in | Any GPIO pins | 2 I2C channels in slave or master mode |
| | I2CEXT0_SDA_in | | |
| | I2CEXT1_SCL_in | | |
| | I2CEXT1_SDA_in | | |
| | I2CEXT0_SCL_out | | |
| | I2CEXT0_SDA_out | | |
| | I2CEXT1_SCL_out | | |
| | I2CEXT1_SDA_out | | |
| LP I2C | LPI2C_SCL_in/out | Any GPIO pins | One LP I2C channel in slave or master mode |
| | LPI2C_SDA_in/out | | |
| I3C master | I3CMST_SCL | GPIO32 | One I3C master for controlling the pull-up resistor. If the built-in pull-up resistor is used, SCL can only use GPIO32 and SDA can only use GPIO33. There is no limitation in pins if the external pull-up is used |
| | I3CMST_SDA | GPIO33 | |
| | I3CMST_SCL_in/out | Any GPIO pins | |
| | I3CMST_SDA_in/out | | |
| | I3CMST_SCL_pullup | | |
| | I3CMST_SDA_pullup | | |
| I3C slave | I3CSLV_SCL_in/out | Any GPIO pins | One I3C slave |
| | I3CSLV_SDA_in/out | | |

| Interface | Signal | Pin | Function |
|---|---|---|---|
| LED PWM | ledc_ls_sig_out0~7 | Any GPIO pins | Eight independent PWM channels |
| I2S | I2S0O_BCK_in | Any GPIO pins | Stereo input and output from/to the audio codec. Three I2S interfaces support full-duplex/half-duplex TDM and PDM input/output. Among these, I2S0 supports PDM-PCM input and PCM-PDM output. |
| | I2S0_MCLK_in | | |
| | I2S0O_WS_in | | |
| | I2S0I_SD_in | | |
| | I2S0I_BCK_in | | |
| | I2S0I_WS_in | | |
| | I2S0I_SD1_in | | |
| | I2S0I_SD2_in | | |
| | I2S0I_SD3_in | | |
| | I2S0O_BCK_out | | |
| | I2S0_MCLK_out | | |
| | I2S0O_WS_out | | |
| | I2S0O_SD_out | | |
| | I2S0I_BCK_out | | |
| | I2S0I_WS_out | | |
| | I2S0O_SD1_out | | |
| | I2S1O_BCK_in | | |
| | I2S1_MCLK_in | | |
| | I2S1O_WS_in | | |
| | I2S1I_SD_in | | |
| | I2S1I_BCK_in | | |
| | I2S1I_WS_in | | |
| | I2S1O_BCK_out | | |
| | I2S1_MCLK_out | | |
| | I2S1O_WS_out | | |
| | I2S1O_SD_out | | |
| | I2S1I_BCK_out | | |
| | I2S1I_WS_out | | |
| | I2S2O_BCK_in | | |
| | I2S2_MCLK_in | | |
| | I2S2O_WS_in | | |
| | I2S2I_SD_in | | |
| | I2S2I_BCK_in | | |
| | I2S2I_WS_in | | |
| | I2S2O_BCK_out | | |
| | I2S2_MCLK_out | | |
| | I2S2O_WS_out | | |
| | I2S2O_SD_out | | |
| | I2S2I_BCK_out | | |
| | I2S2I_WS_out | | |
| LP I2S | LPI2SO_BCK_in | Any GPIO pins | Stereo input and output from/to the audio codec, supporting TDM 16-bit data input, PDM-PCM input, and enabling built-in VAD to generate interrupt and wake-up signals. |
| | LPI2SI_BCK_in | | |
| | LPI2SI_WS_in | | |
| | LPI2SI_BCK_out | | |

| Interface | Signal | Pin | Function |
|---|---|---|---|
| | LPI2SI_WS_out | | |
| Remote Control Peripheral | RMT_SIG_IN0~3 | Any GPIO pins | Two channels for an IR transceiver of various waveforms |
| | RMT_SIG_OUT0~3 | | |
| GPSPI2 | SPI2CLK_in/out | Any GPIO pins | • Master mode and slave mode of SPI, Dual SPI, Quad SPI and QPI, and master mode of Octal SPI and OPI<br>• Four modes of SPI transfer format<br>• Configurable SPI frequency<br>• 64-byte FIFO or GDMA buffer |
| | SPI2CS_in/out | | |
| | SPI2CS1~5_out | | |
| | SPI2D_in/out | | |
| | SPI2Q_in/out | | |
| | SPI2WP_in/out | | |
| | SPI2HD_in/out | | |
| | SPI2D4_in/out | | |
| | SPI2D5_in/out | | |
| | SPI2D6_in/out | | |
| | SPI2D7_in/out | | |
| | SPI2DQS_out | | |
| GPSPI3 | SPI3CLK_in/out | Any GPIO pins | • Master mode and slave mode of SPI, Dual SPI, Quad SPI, and QPI<br>• Four modes of SPI transfer format<br>• Configurable SPI frequency<br>• 64-byte FIFO or GDMA buffer |
| | SPI3CS_in/out | | |
| | SPI3CS1~2_out | | |
| | SPI3D_in/out | | |
| | SPI3Q_in/out | | |
| | SPI3WP_in/out | | |
| | SPI3HD_in/out | | |
| LP SPI | LPSPICLK_in/out | Any GPIO pins | LP SPI interface |
| | LPSPICS_in/out | | |
| | LPSPID_in/out | | |
| | LPSPIQ_in/out | | |
| TWAI® | TWAI0_RX | Any GPIO pins | Three TWAI® interfaces, which are compatible with ISO 11898-1 protocol (CAN Specification 2.0) |
| | TWAI0_TX | | |
| | TWAI0_BUS_OFF_ON | | |
| | TWAI0_CLKOUT | | |
| | TWAI0_STANDBY | | |
| | TWAI1_RX | | |
| | TWAI1_TX | | |
| | TWAI1_BUS_OFF_ON | | |
| | TWAI1_CLKOUT | | |
| | TWAI1_STANDBY | | |
| | TWAI2_RX | | |
| | TWAI2_TX | | |
| | TWAI2_BUS_OFF_ON | | |
| | TWAI2_CLKOUT | | |
| | TWAI2_STANDBY | | |
| Pulse counter | PCNT_SIG_CH0_in0~3 | Any GPIO pins | Capture pulse and count pulse edges in seven modes |
| | PCNT_SIG_CH1_in0~3 | | |
| | PCNT_CTRL_CH0_in0~3 | | |
| | PCNT_CTRL_CH1_in0~3 | | |

| Interface | Signal | Pin | Function |
|---|---|---|---|
| | PCNT_RST_in0~3 | | |
| MCPWM | PWM0_SYNC0~2_in | Any GPIO pins | 2 MCPWM input and output pins. Signals include:<br>• PWM differential output signals<br>• fault input signals to be detected<br>• input signals to be captured<br>• external clock synchronization signals |
| | PWM0_out0a | | |
| | PWM0_out0b | | |
| | PWM0_out1a | | |
| | PWM0_F0~2_in | | |
| | PWM0_out1b | | |
| | PWM0_out2a | | |
| | PWM0_out2b | | |
| | PWM0_CAP0~2_in | | |
| | PWM1_SYNC0~2_in | | |
| | PWM1_out0a | | |
| | PWM1_out0b | | |
| | PWM1_out1a | | |
| | PWM1_F0~2_in | | |
| | PWM1_out1b | | |
| | PWM1_out2a | | |
| | PWM1_out2b | | |
| | PWM1_CAP0~2_in | | |
| PARLIO | PARL_RX_DATA0~15 | Any GPIO pins | A module for parallel data transfer, with<br>• 16 pins to receive parallel data<br>• 16 pins to transmit parallel data<br>• 1 receiver clock pin (clock input)<br>• 2 transmitter clock pins (clock input and output) |
| | PARL_TX_DATA0~15 | | |
| | PARL_RX_CLK_in/_out | | |
| | PARL_TX_CLK_in/_out | | |
| USB Serial/JTAG | USB_D- | GPIO24/26 | USB-to-serial converter and USB-to-JTAG (The pin functions of USB_D- and USB_D+ are interchangeable. GPIO24/25 and GPIO26/27 are the D+ and D- pins of two USB PHY. USB Serial/JTAG can use any one of them, and the default one is GPIO24/25) |
| | USB_D+ | GPIO25/27 | |
| USB 2.0 OTG full-speed | USB_D- | GPIO24/26 | USB 2.0 OTG full-speed (The pin functions of USB_D- and USB_D+ are interchangeable. GPIO24/25 and GPIO26/27 are the D+ and D- pins of two USB PHY. USB 2.0 OTG can use any one of them, and the default one is GPIO26/27) |
| | USB_D+ | GPIO25/27 | |
| | IDDIQ_in | Any GPIO pins | |
| | AVALID_in | | |
| | VBUSVALID_in | | |
| | SRPBVALID_in | | |
| | SRPSESSEND_in | | |
| | SRPDISCHRGVBUS_out | | |
| | SRPCHRGVBUS_out | | |
| | DRVVBUS_out | | |
| | IDPULLUP_out | | |
| | DPPULLDOWN_out | | |

| Interface | Signal | Pin | Function |
|---|---|---|---|
| | DMPULLDOWN_out | | |
| USB 2.0 OTG high-speed | USB_D- | DM | USB 2.0 OTG high-speed interface |
| | USB_D+ | DP | |
| | ADPPRB_in | Any GPIO pins | |
| | ADPSNS_in | | |
| | DRVBUS_out | | |
| | ADPCHRG_out | | |
| | ADPDISCHRG_out | | |
| | ADPPRBEN_out | | |
| | ADPSNSEN_out | | |
| | PHY_REFCLK_in | | |
| USB Serial/JTAG Bridge | USB_JTAG_TDO_BRG_in | Any GPIO pins | ESP32-P4 can be used as a bridge between the USB port and the JTAG port |
| | USB_JTAG_TDI_BRG_out | | |
| | USB_JTAG_TMS_BRG_out | | |
| | USB_JTAG_TCK_BRG_out | | |
| | USB_JTAG_TRST_BRG_out | | |
| HP CORE FAST IO | CORE_GPIO_in/out_0~15 | Any GPIO pins | HP CORE fast GPIO |
| Analog PAD Voltage Comparator | ANA_COMP0_in0 | GPIO51 | Analog PAD voltage comparator |
| | ANA_COMP0_in1 | GPIO52 | |
| | ANA_COMP1_in0 | GPIO53 | |
| | ANA_COMP1_in1 | GPIO54 | |
| | ANA_COMP0_out | Any GPIO pins | |
| | ANA_COMP1_out | | |
| SDIO3.0 | SD1_D0 | GPIO39 | SDIO 3.0 interface |
| | SD1_D1 | GPIO40 | |
| | SD1_D2 | GPIO41 | |
| | SD1_D3 | GPIO42 | |
| | SD1_CLK | GPIO43 | |
| | SD1_CMD | GPIO44 | |
| | SD1_D4 | GPIO45 | |
| | SD1_D5 | GPIO46 | |
| | SD1_D6 | GPIO47 | |
| | SD1_D7 | GPIO48 | |
| | SD1_DETECT_N_in | Any GPIO pins | |
| | SD1_INT_N_in | | |
| | SD1_WRITE_PRT_in | | |
| | SD1_DATA_STRB_in | | |
| | SD1_RST_N_out | | |
| SDIO2.0 | SD2_D0_in/out | Any GPIO pins | SDIO 2.0 interface |
| | SD2_D1_in/out | | |
| | SD2_D2_in/out | | |
| | SD2_D3_in/out | | |
| | SD2_CLK_out | | |
| | SD2_CMD_in/out | | |

| Interface | Signal | Pin | Function |
|---|---|---|---|
| | SD2_D4_in/out | | |
| | SD2_D5_in/out | | |
| | SD2_D6_in/out | | |
| | SD2_D7_in/out | | |
| | SD2_DETECT_N_in | | |
| | SD2_INT_N_in | | |
| | SD2_WRITE_PRT_in | | |
| | SD2_DATA_STRB_in | | |
| | SD2_RST_N_out | | |
| | SD_CMD_PULLUPEN_N_out | | |
| EMAC | RMII_RXDV | GPIO28/45/51 | Fast Ethernet RMII interface |
| | RMII_RXD0 | GPIO29/46/52 | |
| | RMII_RXD1 | GPIO30/47/53 | |
| | RMII_RXER | GPIO31/48/54 | |
| | RMII_CLK | GPIO32/44/50 | |
| | RMII_TXEN | GPIO33/40/49 | |
| | RMII_TXD0 | GPIO34/41 | |
| | RMII_TXD1 | GPIO35/42 | |
| | RMII_TXER | GPIO36/43 | |
| | MII_RXCLK | Any GPIO pins | Fast Ethernet MII interface |
| | MII_RXDV | | |
| | MII_RXD0 | | |
| | MII_RXD1 | | |
| | MII_RXD2 | | |
| | MII_RXD3 | | |
| | MII_RXER | | |
| | MII_TXCLK | | |
| | MII_TXEN | | |
| | MII_TXD0 | | |
| | MII_TXD1 | | |
| | MII_TXD2 | | |
| | MII_TXD3 | | |
| | MII_TXER | | |
| | MDI_in | Any GPIO pins | Fast Ethernet MDIO interface |
| | MDO_out | | |
| | MDC_out | | |
| | COL_IN | Any GPIO pins | Other Fast Ethernet interfaces |
| | CRS_IN | | |
| CAM Interface | CAMCLK_out | Any GPIO pins | DVP interface for camera |
| | CAMCLK_in | | |
| | CAM_HENABLE_in | | |
| | CAM_HSYNC_in | | |
| | CAM_VSYNC_in | | |
| | CAM_DATA_in0~15 | | |
| LCD Interface | LCDCLK_out | Any GPIO pins | 24-bit display interface |

| Interface | Signal | Pin | Function |
|---|---|---|---|
| | LCD_HENABLE_out | | |
| | LCD_HSYNC_out | | |
| | LCD_VSYNC_out | | |
| | LCD_DATA_out0~23 | | |
| | LCD_CS_out | | |
| | LCD_DC_out | | |
| MIPI CSI | CSI_DATAN0 | CSI_DATAN0 | MIPI CSI interface |
| | CSI_DATAP0 | CSI_DATAP0 | |
| | CSI_CLKN | CSI_CLKN | |
| | CSI_CLKP | CSI_CLKP | |
| | CSI_DATAN1 | CSI_DATAN1 | |
| | CSI_DATAP1 | CSI_DATAP1 | |
| | CSI_REXT | CSI_REXT | |
| MIPI DSI | DSI_DATAN0 | DSI_DATAN0 | MIPI DSI interface |
| | DSI_DATAP0 | DSI_DATAP0 | |
| | DSI_CLKN | DSI_CLKN | |
| | DSI_CLKP | DSI_CLKP | |
| | DSI_DATAN1 | DSI_DATAN1 | |
| | DSI_DATAP1 | DSI_DATAP1 | |
| | DSI_REXT | DSI_REXT | |
| SPI Flash Interface | FLASH_CS | FLASH_CS | Flash MSPI interface |
| | FLASH_Q | FLASH_Q | |
| | FLASH_WP | FLASH_WP | |
| | FLASH_HOLD | FLASH_HOLD | |
| | FLASH_CK | FLASH_CK | |
| | FLASH_D | FLASH_D | |

# 4 Electrical Characteristics

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

## 4.1 Absolute Maximum Ratings

Stresses beyond the absolute maximum ratings listed in the table below may cause permanent damage to the device. These are stress ratings only, and do not refer to the functional operation of the device.

**Table 4-1. Absolute Maximum Ratings**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| VDDPST_1, VDDPST_LDO, VDDPST_DCDC, VDDA, VBAT, VDDPST | Voltage applied to power supply pins per power domain | –0.3 | 3.6 | V |
| VDDPST_2, VDDPST_3, VDDPST_4, VDDPST_5, VDDPST_6 | Voltage applied to I/O power supply pins per power domain | –0.3 | 3.6 | V |
| VDD_HP_0, VDD_HP_1, 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 |

## 4.2 Recommended Operating Conditions

**Table 4-2. Recommended Operating Conditions**

| Symbol | Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| VDDPST_1, VDDPST_LDO, VDDPST_DCDC, VDDA, VBAT | Voltage applied to power supply pins per power domain | 3.0 | 3.3 | 3.6 | V |
| VDDPST_2, VDDPST_3, VDDPST_4, VDDPST_5, VDDPST_6 | 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_1, VDD_HP_2, VDD_HP_3 | Voltage applied to core power supply pins per power domain (from DCDC) | 3.0 | 3.3 | 3.6 | V |
| 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 |

**Cont'd on next page**

Table 4-2 – cont'd from previous page

| Symbol | Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $T_A$ | Ambient temperature | –40 | — | 105 | °C |

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

## 4.3 VFB1_VO1 Output Characteristics

Table 4-3. VDD_SPI Output Characteristics

| Symbol | Parameter | Typ | Unit |
|---|---|---|---|
| $R_{VFB1}$ | On-resistance in 3.3 V mode | 7.5 | Ω |

In real-life applications, when VFB1_VO1 works in 3.3 V output mode, VFB1_VO1 is generally used to drive flash, and VDDPST_LDO may be affected by $R_{SPI}$. For example, when VDDPST_LDO is used to drive a 3.3 V flash, it should comply with the following specifications:

VDDPST_LDO > VDD_flash_min + I_flash_max * $R_{VFB1}$

Among which, VDD_flash_min is the minimum operating voltage of the flash, and I_flash_max the maximum current.

For more information, please refer to section 2.3 *Power Scheme*.

## 4.4 DC Characteristics (3.3 V, 25 °C)

Table 4-4. DC Characteristics (3.3 V, 25 °C)

| Symbol | Parameter | 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 | 0.75 × VDD[1] | — | VDD[1]+ 0.3 | V |
| $V_{IL\_nRST}$ | Chip reset voltage | –0.3 | — | 0.25 × VDD[1] | V |

[1] VDD is the I/O voltage for a particular power domain of pins.

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

# 5 Related Documentation and Resources

## Related Documentation

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

## Developer Zone

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

## Products

- *ESP32-P4 Series SoCs* – Browse through all ESP32-P4 SoCs.
  https://espressif.com/en/products/socs?id=ESP32-P4
- *ESP32-P4 Series Modules* – Browse through all ESP32-P4-based modules.
  https://espressif.com/en/products/modules?id=ESP32-P4
- *ESP32-P4 Series DevKits* – Browse through all ESP32-P4-based devkits.
  https://espressif.com/en/products/devkits?id=ESP32-P4
- *ESP Product Selector* – Find an Espressif hardware product suitable for your needs by comparing or applying filters.
  https://products.espressif.com/#/product-selector?language=en

## Contact Us

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

# Revision History

| Date | Version | Release notes |
|---|---|---|
| 2023-7-26 | 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.