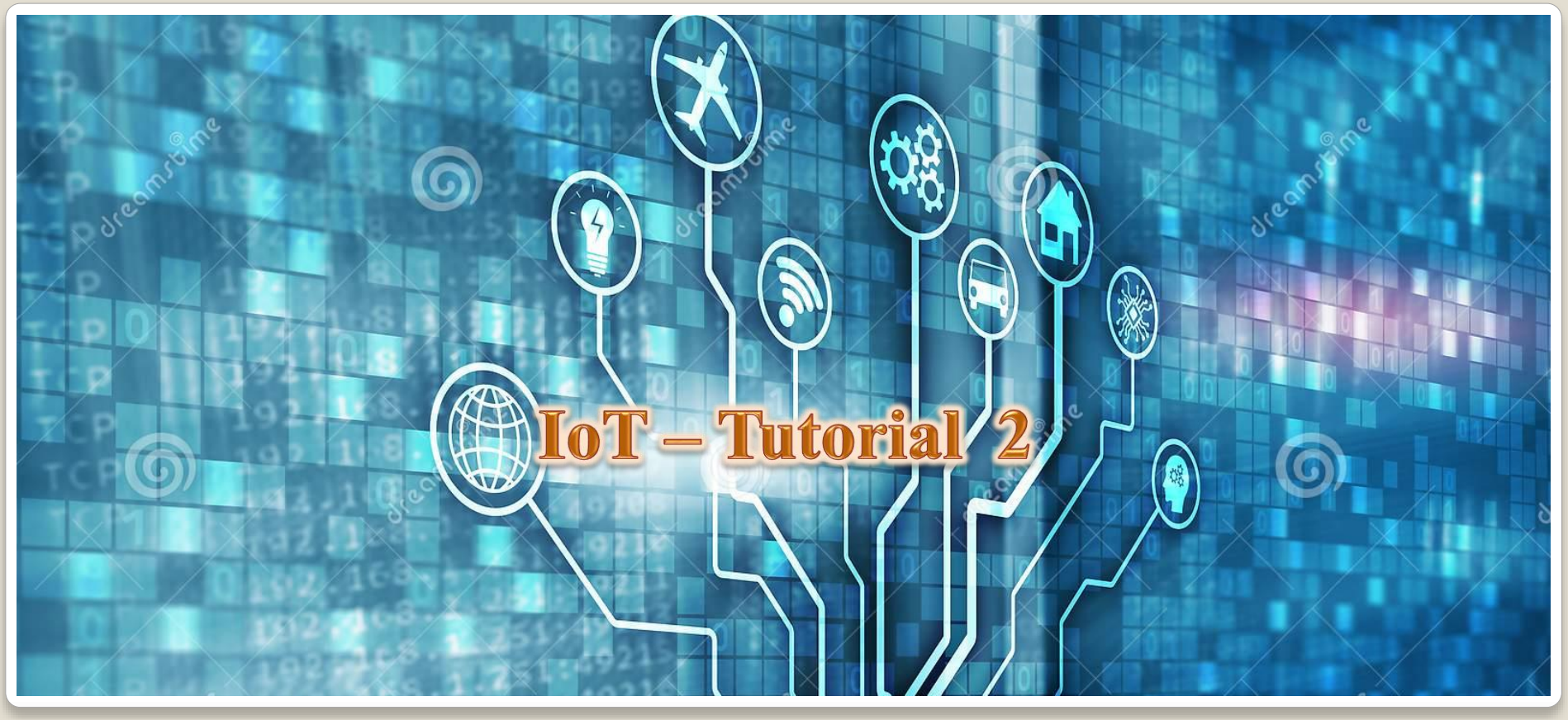


ESP32 Details

Mouli Sankaran

Ref: All about ESP32

Focus: ESP 32 Details

- ESP32-WROOM-32
  - Specification
  - CPU and Internal Memory
- ESP32-IDF
- Compilation/Upload Details
- Modified Harvard Architecture
- Xtensa LX6 CPU
  - Address Ranges
  - Memory Map

ESP32-WROOM-32
Specifications

Specifications of ESP-32

- Processors
  - Xtensa Single/Dual-core 32-bit LX6 microprocessor, 160MHz to 240MHz, 600 DMIPS
  - Ultralow power co-processor
  - 448kB ROM
  - 520kB SRAM
- Wireless connectivity
  - Wi-Fi: 802.11 b/g/n/e/i
  - Bluetooth: v4.2 BR/EDR and BLE
- Security
  - IEEE 802.11 standard security WFA, WPA/WPA2 and WAPI
  - Secure boot
  - Flash encryption
  - 1024-bit OTP, up to 768-bit for customers
  - Cryptographic hardware acceleration: AES, SHA-2, RSA, elliptic curve cryptography (ECC), random number generator (RNG)
- Power management
  - Internal low-dropout regulator
  - Individual power domain for RTC
  - 5μA deep sleep current
  - Wake up from GPIO interrupt, timer, ADC measurements, capacitive touch sensor interrupt

- Peripheral Interfaces
  - 12-bit SAR ADC up to 18 channels
  - 2× 8-bit DACs
  - 10× touch sensors (capacitive sensing GPIOs)
  - Temperature sensor
  - 4× SPI
  - 2× I2S interfaces
  - 2× I2C interfaces
  - 3× UART
  - Host controller (SD/SDIO/CE-ATA/MMC/eMMC)
  - Slave controller (SDIO/SPI)
  - Ethernet MAC interface with dedicated DMA and IEEE 1588 support
  - CAN (Controller Area Network) 2.0
  - Infrared remote controller (TX/RX, up to 8 channels)
  - Motor PWM
  - LED PWM up to 16 channels
  - Hall effect sensor
  - Ultralow power analog pre-amplifier

SAR (Successive Approximation Register) ADCs, with18 measurement channels (analog enabled pins).

Ten capacitive touch GPIOs. It can sense variations in anything that holds electrical charges, like the human skin.

Hall effect sensor that detects changes in the magnetic field in its surrounding.

DMIPS: **Dhrystone MIPS,** benchmark that measures integer performance. **MIPS**: Million Instructions per Second

One Time Programmable (**OTP**) memory, based on four blocks of **256 eFuses** (total of 1024 bits)

Memory Layout of ESP32-WROOM-32

ESP32-WROOM-32:
CPU and Internal Memory

ESP32-D0WDQ6 contains two low-power Xtensa® 32-bit LX6 microprocessors. The internal memory includes:

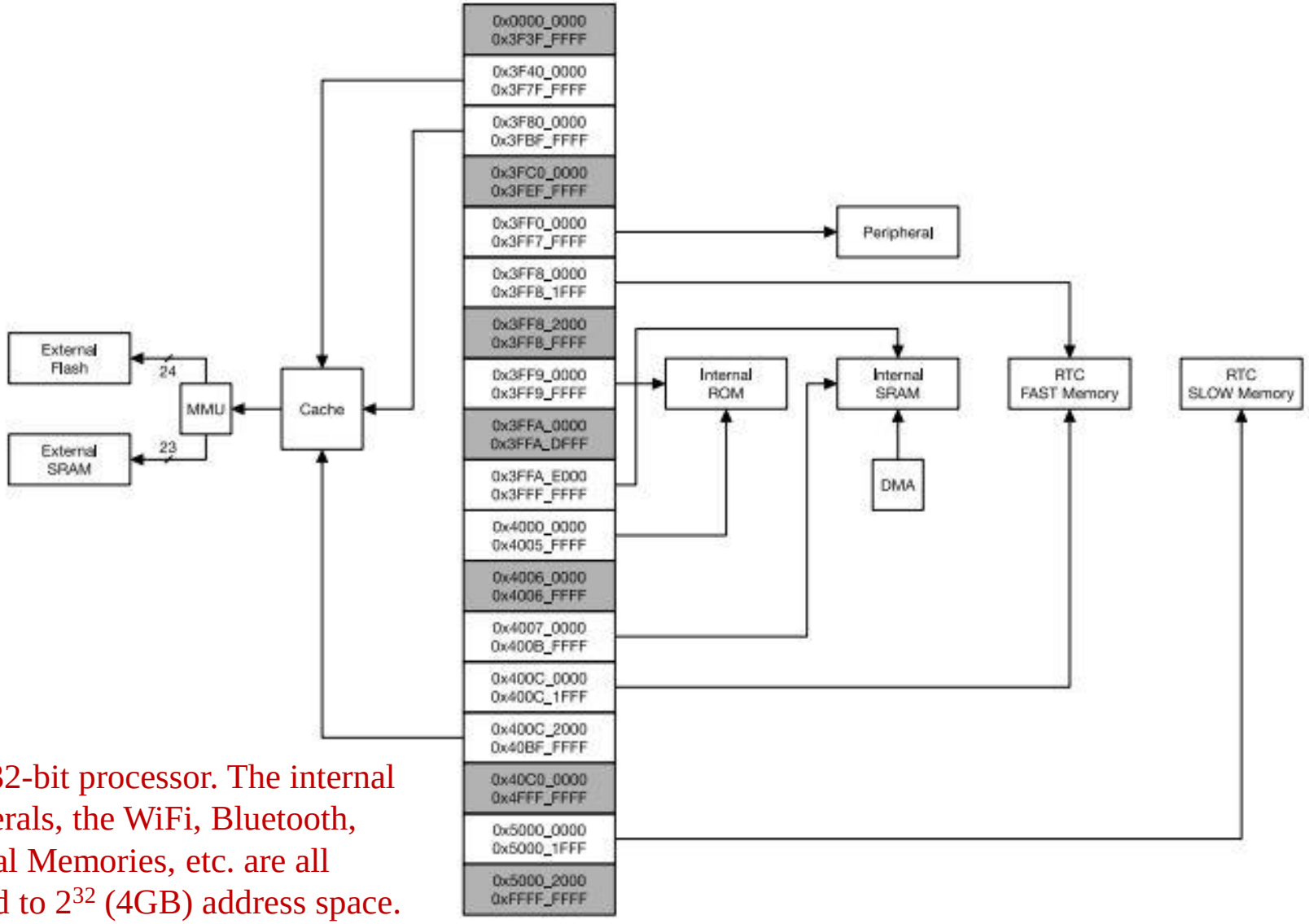
- 448 KB of ROM for booting and core functions.
- 520 KB of on-chip SRAM for data and instructions.
- 8 KB of SRAM in RTC, which is called RTC FAST Memory and can be used for data storage; it is accessed by the main CPU during RTC Boot from the Deep-sleep mode.
- 8 KB of SRAM in RTC, which is called RTC SLOW Memory and can be accessed by the co-processor during the Deep-sleep mode.
- 1 Kbit of eFuse: 256 bits are used for the system (MAC address and chip configuration) and the remaining 768 bits are reserved for customer applications, including flash-encryption and chip-ID.

Note: ESP32-WROOM-32 integrates a 4 MB SPI flash, which is connected to GPIO 6, GPIO 7, GPIO 8, GPIO 9, GPIO 10 and GPIO 11. These six pins cannot be used as regular GPIOs. This needs to be verified.

Ref: ESP32-WROOM-32 Datasheet

ESP32-WROOM32: Memory Layout

Ref: ESP32 Features



It is a 32-bit processor. The internal peripherals, the WiFi, Bluetooth, External Memories, etc. are all mapped to 2^{32} (4GB) address space.

ESP32-IDF

ESP-IDF on Arduino Genuinio IDE

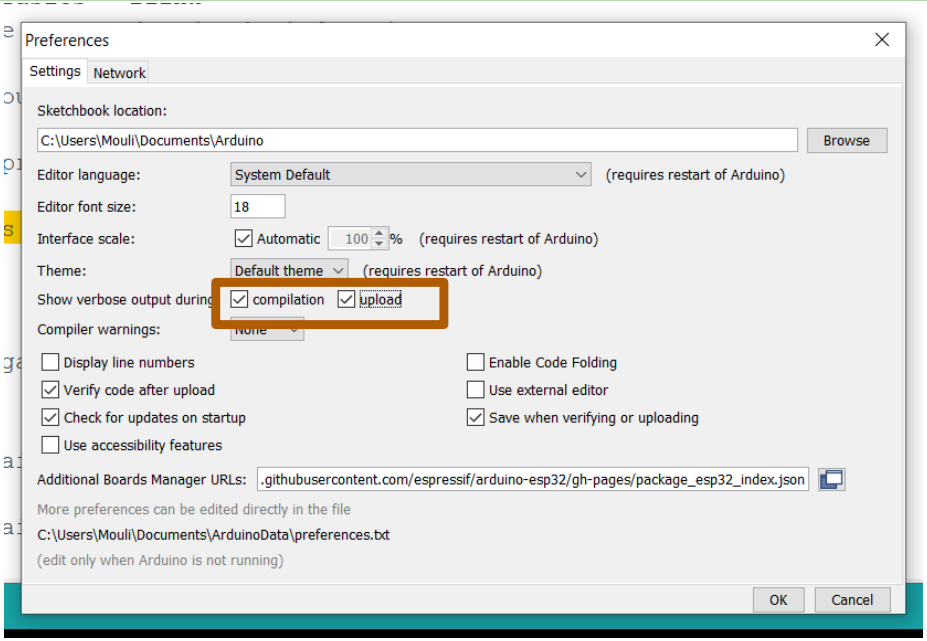
- This differentiation is done in the Espressif Internet Development Framework (ESP-IDF).
  - ESP-IDF is the official software development framework for the chip. Arduino and other implementations for the development will be based on ESP-IDF.
- ESP-IDF uses freeRTOS for switching between the processors and data exchange between them.

- The ESP-IDF development kit (for Espressif IoT Development Framework) for the Arduino IDE allows to develop code for all development boards based on ESP32.
- ESP-IDF is very easy to install from the library manager of the **Arduino IDE**.
- ESP-IDF framework is directly available from the board manager of the Arduino IDE.

ESP32-Booting Sequence

Compilation/Upload Details

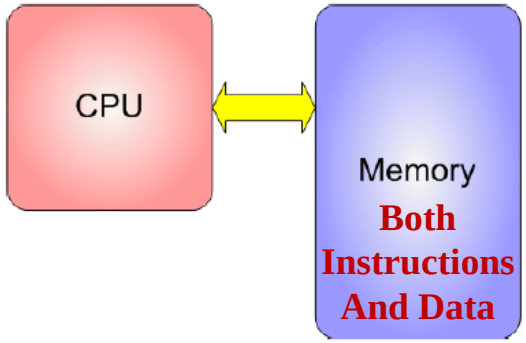
- Select ***File→Preferences***
- Choose both options under "Show verbose output during"
  - Compilation
  - Upload
- This will show the activities during the compilation and download process, on the output window below.



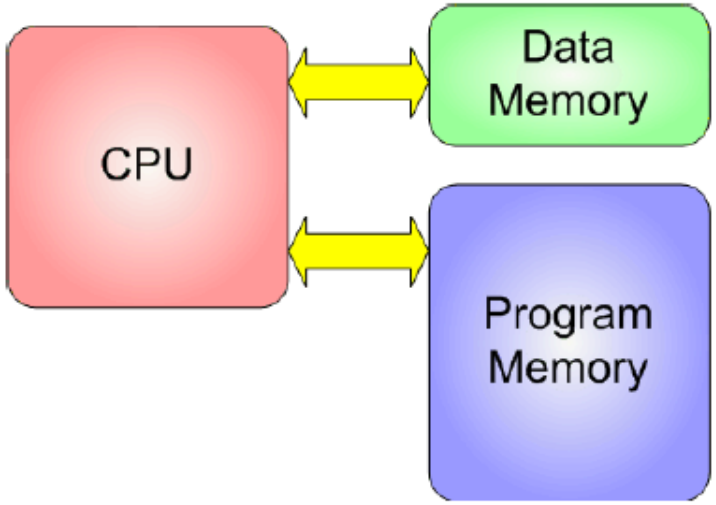
- Observe the output given by the tools while the sketch (program) is being compiled and uploaded.
- You can high light all the output and take a copy of that with Ctrl+C and copy it into a text file to analyze it.
- It will give more information what is happening in the background. You can go over the directories to know the tools and configuration files associated with ESP32.
- Make sure that you do not delete or modify any of the files while going over it.

Harvard Architecture

Harvard Architecture



Von Neumann
(a.k.a. Princeton)
architecture



Harvard
architecture

- **Harvard Architecture** has separate program and data memory address spaces and buses
  - Separate buses for the program memory (for instructions) and data memory, which are accessed by the processor in parallel

Harvard Vs Modified Harvard Architecture

- The **Harvard architecture** is a computer architecture with physically separates storage and signal pathways for instructions and data
- The **modified Harvard architecture** is a variation of the Harvard computer architecture that allows the contents of the instruction memory to be accessed as if it were data.
  - Which means that the constant data values are stored as part of Instruction memory and accessed as data.
- Most modern computers that are documented as Harvard architecture are, in fact, modified Harvard architecture.
- A "modified Harvard" architecture has multiple memory buses, but they share the same address space.
- By being able to simultaneously fetch instructions on one bus and data on the other bus, performance improvements achieved.
- Also, instructions stored in data (SRAM) memory can also be executed.
- ESP32 falls under the modified Harvard architecture.

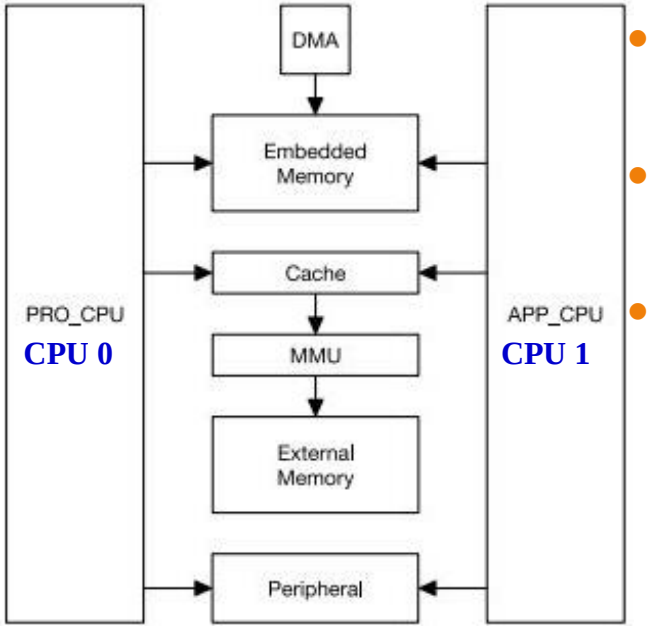
Xtensa LX6 CPU

(Address map)

ESP32 _Ref1: Technical Reference Manual

IoT Tutorials – © 2021 Mouli Sankaran Email: mouli.sankaran@yahoo.com

Dual-Cores: Address Mapping

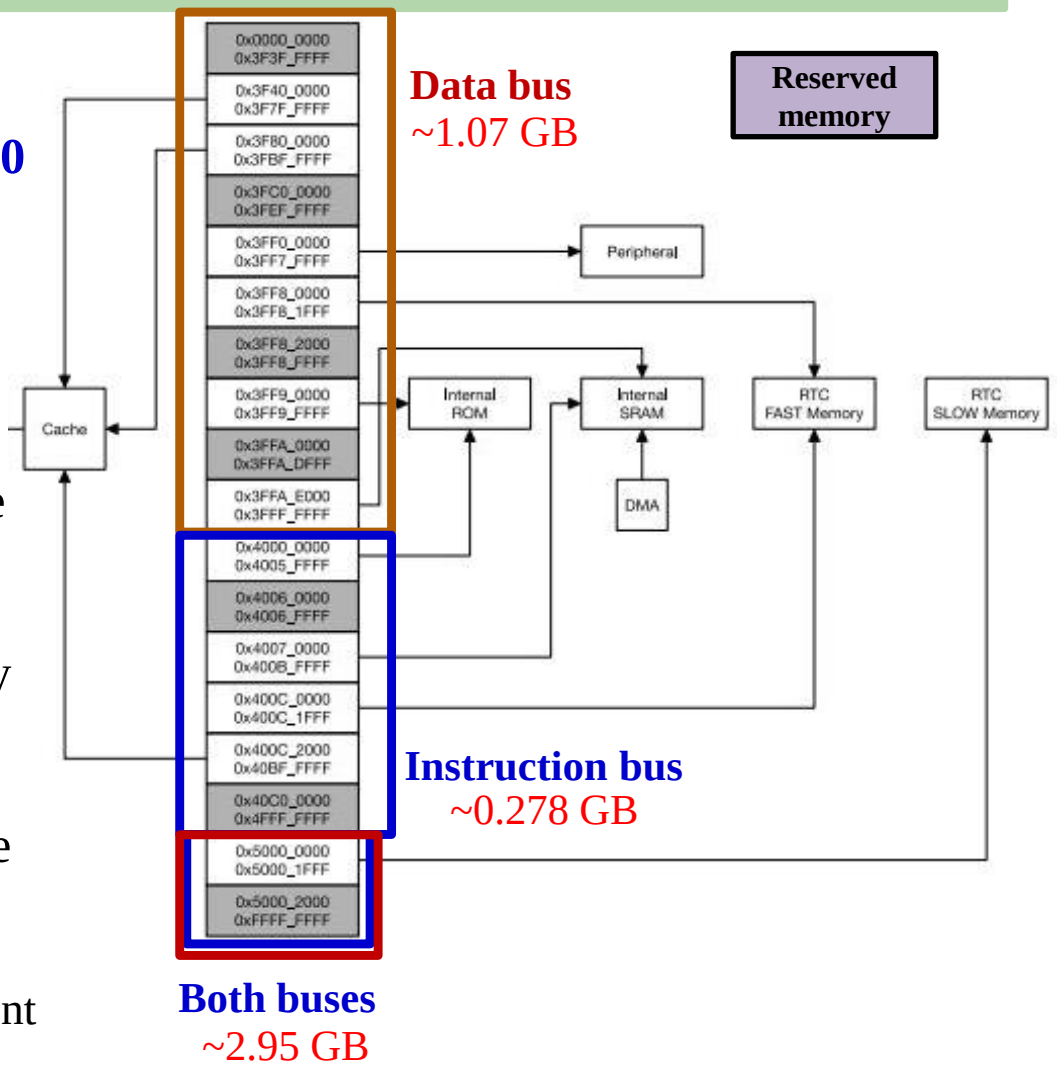


- Each of the two Harvard Architecture Xtensa LX6 CPUs has 4 GB (32-bit) address space.
- Address spaces are symmetric between the two CPUs.
- The two cores are named Protocol CPU (**PRO_CPU**) and Application CPU (**APP_CPU**).
  - PRO_CPU handles the WiFi, Bluetooth and other internal peripherals like SPI, I2C, ADC etc.
  - The APP_CPU is left out for running the application code.

- Both the cores are mapped symmetrically to the same address space.
- It basically means, a value stored in the memory can be accessed, using the same memory address from both the CPUs.

Xtensa LX6 CPU: Address Ranges

- Addresses **below 0x4000_0000** are serviced using the data bus.
- Addresses in the range **0x4000_0000 ~0x4FFF_FFFF** are serviced using the instruction bus.
- Finally, **addresses over and including 0x5000_0000** are shared by the **data** and **instruction** bus.
- The data bus and instruction bus are both **little-endian**.
- The CPU can read and write data through the instruction bus, but only in a word aligned manner.
- Each CPU can directly access embedded memory through both the data bus and the instruction bus.
  - External memory, which is mapped into the address space, (via transparent caching & MMU), and peripherals.



Total: 2^{32} GB = ~4.29GB = 0x0000 0000 to 0xFFFF FFFF

Xtensa LX6 CPU : Address Mapping

Bus Type	Boundary Address		Size	Target
	Low Address	High Address		
	0x0000_0000	0x3F3F_FFFF		Reserved
Data	0x3F40_0000	0x3F7F_FFFF	4 MB	External Memory
Data	0x3F80_0000	0x3FBF_FFFF	4 MB	External Memory
	0x3FC0_0000	0x3FEF_FFFF	3 MB	Reserved
Data	0x3FF0_0000	0x3FF7_FFFF	512 KB	Peripheral
Data	0x3FF8_0000	0x3FFF_FFFF	512 KB	Embedded Memory
Instruction	0x4000_0000	0x400C_1FFF	776 KB	Embedded Memory
Instruction	0x400C_2000	0x40BF_FFFF	11512 KB	External Memory
	0x40C0_0000	0x4FFF_FFFF	244 MB	Reserved
Data / Instruction	0x5000_0000	0x5000_1FFF	8 KB	Embedded Memory
	0x5000_2000	0xFFFF_FFFF		Reserved

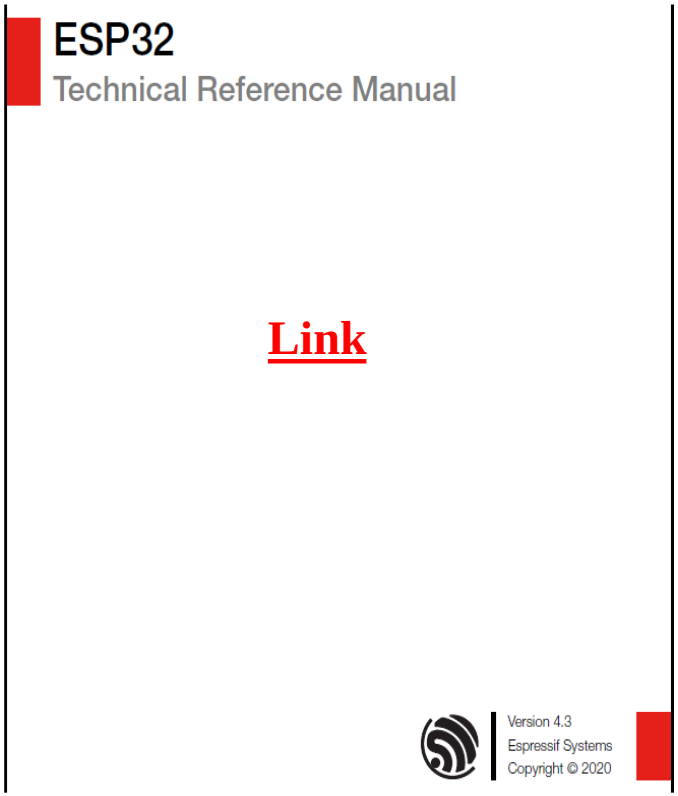
- The table above illustrates address ranges that can be accessed by each CPU's data bus and instruction bus.
- Some embedded memories and some external memories can be accessed via the data bus or the instruction bus.
- In these cases, the same memory is available to either of the CPUs at two address ranges.

Summary: ESP 32 Details

- ESP32-WROOM-32
  - Specification
  - CPU and Internal Memory
- ESP32-IDF
- Compilation/Upload Details
- Modified Harvard Architecture
- Xtensa LX6 CPU
  - Address Ranges
  - Memory Map

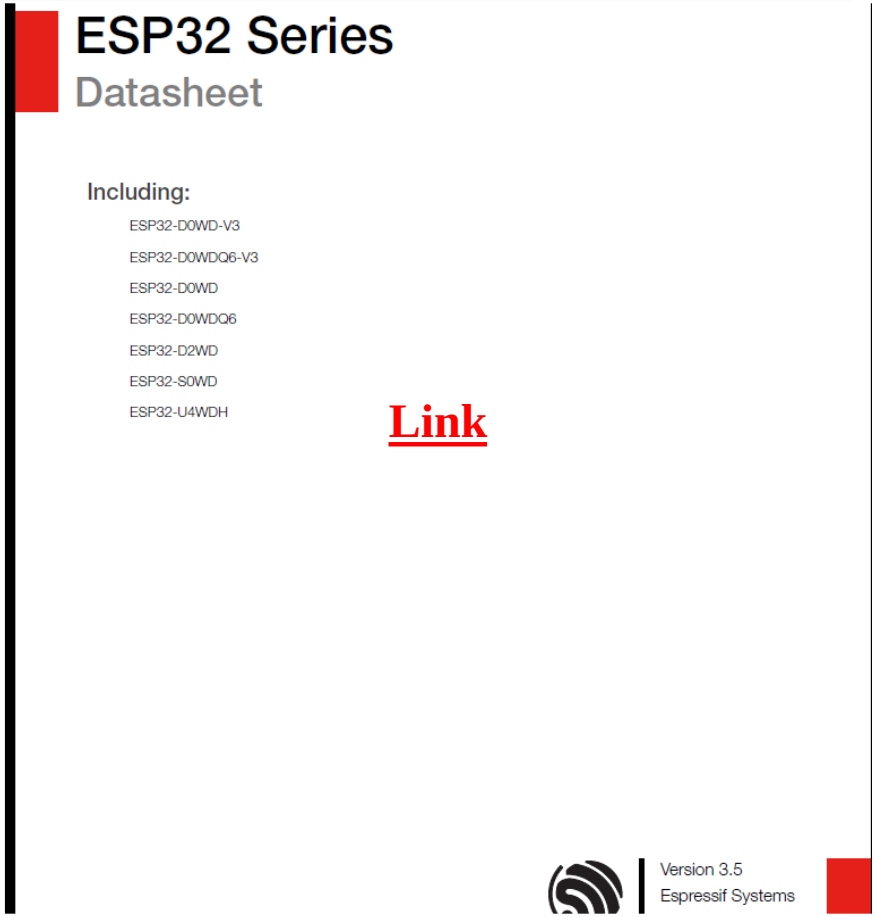
References: For ESP32

Ref 1



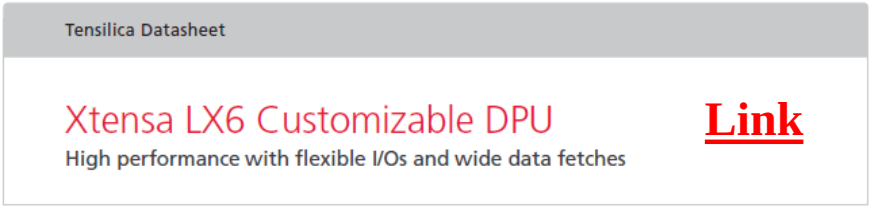
Link

Ref 2



Link

Ref 3



Link

Ref 4

Link