Homework #1

Owner	A Angel David Sanchez
Tags	Embedded software

▼ **Block Diagram**

- **Sytem bus can operate at the speed up to 180MHZ?** - NO, the max in the case of the STM32F411CEU6 is up to 100MHZ, this is because the features of the STM32F4 arm cortex M4 can only go up to this speed.
- **SRAMS A are connected to System BUS T/F?** - False, it is connected to the bus-matrix
- **APB1 Bus can operate at the speed up to 180MHZ?** - No, its max is 50mhz
- **Let's say I have a peripheral whose datasheet says that, is operating frequency or speed must be above 95MHz, can I connect that peripheral via APB2 bus?** - yes, we can connect it there, because the max frequency for this bus is 100MHZ
- **What is the max HCLK value of the MCU?** - It varies depending with register we are using: on reset we internal oscilator is set to 16MHZ

PWR_CR	Register = 0x01	64MHZ
PWR_CR	Register = 0 0x10	84MHZ - general value
PWR_CR	Register = 0x11	100MHZ

- What is the max P1CLK (APB1) and P2CLK (APB2)value of our MCU? - If a timer is connected the max is of 100MHZ, if we are using it for USART2, then is 50MHZ, the frequency can varieS depending of the use and the peripheral that is being used, but the answer of this question is that internally the APB1 has a max of 50mhz and the APB2 a mas of 100MHZ
- GPIOs and processors communicate over AHB1 bus? Is True, because the GPIOs and processor are connected to the AHB, although in my case my MCU has 2 AHB and the peripherals are connected to both of these.
- USB OTG and processor communicate over AHB2 bus? - Yes, it is connected to the AHB2.

▼ **Memory Map**

- **What's the base address of AHB1 Bus Peripheral?** - 0x4002 0000
- **What is the base address of RCC engine registers of the MCU?** - 0x4002 3800
- **What is the base address of APB1 Peripherals?** - 0x400 0000
- **What is the base address of Flash Memory?** - 0x0800 0000
- **What is the base address of SRAM2?** - 0X2000 0000
- **What is the base address of ADC registers?** - 0X4001 2000

▼ **Glossary**

- **ARM: (Advanced RISC Machine)** ARM is a type of processor architecture commonly used in microcontrollers and other embedded systems. ARM processors are known for their power efficiency, performance, and versatility.

▼ **RISC**

RISC stands for Reduced Instruction Set Computing. It's a CPU design philosophy that focuses on simplifying and optimizing instructions performed by the processor. Here are some key points about RISC:

1. **Simplified Instruction Set:** RISC architectures use a smaller set of simple and frequently used instructions. This simplicity allows faster execution of instructions.
2. **Fixed-Length Instructions:** Instructions in RISC architectures are typically of fixed length, making decoding and executing them more straightforward and faster.
3. **Load/Store Architecture:** RISC architectures often separate load/store instructions for accessing memory from arithmetic/logical instructions. This means data must be loaded from memory into registers before processing and stored back afterward.
4. **Pipelining:** RISC processors commonly use pipelining techniques to execute multiple instructions simultaneously by breaking down instruction execution into stages.
5. **Efficiency and Performance:** RISC architectures are designed for high performance and efficiency by prioritizing common operations and

optimizing the execution of these operations.

6. **ARM Architecture:** ARM processors, commonly used in microcontrollers and mobile devices, are based on the RISC architecture. ARM's focus on power efficiency and performance stems from RISC principles.

- **SWD: (Serial Wire Debug)** SWD is a debug and programming interface used to communicate with the microcontroller's embedded software during development. It enables features like debugging, flashing firmware, and accessing internal registers.
- **ITM: (Instrumentation Trace Macrocell):** ITM is a part of ARM's CoreSight debug architecture. It allows non-intrusive debugging by providing a mechanism to output trace information from the CPU and peripherals while the system is running.
- **JTAG: (Joint Test Action Group):** JTAG is a standard for testing and debugging hardware, allowing access to on-chip debug functions. It enables operations like boundary scanning, programming, and debugging of microcontrollers.
- **FIFO: (First-In, First-Out):** FIFO is a data structure used to manage data in a way that the first data to enter the structure is the first to be removed. It's commonly used in buffering data for various peripherals or communication interfaces.
- **.elf: (Executable and Linkable Format):** .elf is a file format used for executables, object code, shared libraries, and more. In the context of microcontrollers, .elf files often contain compiled firmware that can be loaded onto the microcontroller.
- **DMA: (Direct Memory Access):** DMA is a feature that enables peripherals to access system memory directly without involving the CPU. It helps in efficient data transfer between peripherals and memory.
- **USB OTG: (On-The-Go):** USB OTG is a specification that allows USB devices to act as either hosts or peripherals. In microcontrollers, USB OTG enables them to switch roles between being a USB host or a USB device.
- **HBA: (Host Bus Adapter):** HBA is an interface used in systems to connect the CPU or host system to other components like storage devices or network adapters.

- **APB: (Advanced Peripheral Bus):** APB is a bus protocol used in ARM-based microcontrollers to connect low-bandwidth peripherals to the system. It operates at a slower speed compared to the AHB (Advanced High-Performance Bus).
- **RCC: (Reset and Clock Control):** RCC is a module in microcontrollers responsible for system clock configuration, enabling and disabling peripheral clocks, and managing the reset and startup sequences.
- **HCLK: (Host Clock):** HCLK refers to the main clock signal used by the microcontroller's core and various bus systems. It determines the operating frequency of the CPU and other synchronous modules within the microcontroller.

▼ **SRAM**

SRAM stands for Static Random-Access Memory. It's a type of volatile memory used in microcontrollers and other computing devices for high-speed data storage and retrieval. Here are some key characteristics of SRAM:

1. **Speed:** SRAM is faster compared to other types of memory like DRAM (Dynamic RAM). It offers fast access times, enabling rapid read and write operations.
2. **Volatility:** SRAM is volatile memory, meaning it loses its stored data when power is turned off. It requires constant power to retain information.
3. **Structure:** SRAM uses flip-flops to store data, which makes it faster but also more expensive in terms of chip area compared to other types of memory.
4. **Low Power Consumption:** While SRAM is faster than DRAM, it generally consumes more power. However, in standby mode, SRAM consumes very little power compared to dynamic memory types.
5. **Used for Caches and Buffers:** SRAM is commonly used in microcontrollers as cache memory to store frequently accessed data. It's also used as buffer memory for various peripherals or temporary storage of data during processing.
6. **Applications:** Due to its speed and random access nature, SRAM finds applications in cache memory, register files within processors, and other

areas where fast access to data is critical.

- **System bus:** This refers to the collective pathways used for communication between different components within a system, such as the CPU, memory, and peripherals. It's the primary data highway used for transferring information between these elements.
- **Matrix bus:** The bus matrix is a part of the microcontroller's architecture that manages and arbitrates access to different buses within the system. It's responsible for routing data and managing the flow of information between various components, peripherals, and memory modules. The bus matrix allows multiple masters (like CPU, DMA controllers, etc.) to access different parts of the system simultaneously, handling bus contention and prioritizing access when multiple devices need to communicate.

▼ **My MCU**

1. **Core:** ARM Cortex-M4 32-bit RISC core running at up to 100 MHz frequency.
2. **Memory:**
   - 512 KB Flash memory for program storage.
   - 128 KB SRAM for data storage.
3. **Peripherals:**
   - Advanced peripherals including timers, ADCs, DACs, SPI, I2C, USART, USB OTG, CAN, and more.
   - DMA (Direct Memory Access) controller for efficient data transfers.
   - GPIO (General Purpose Input/Output) pins for interfacing with external devices.
4. **Clocks:**
   - RCC (Reset and Clock Control) for managing system clocks.
   - HCLK (Host Clock) up to 100 MHz.
5. **Connectivity:**
   - USB OTG (On-The-Go) for USB device/host connectivity.
   - CAN (Controller Area Network) for communication in automotive and industrial applications.

6. **Debugging and Interface:**
    - SWD (Serial Wire Debug) for programming and debugging.
    - JTAG (Joint Test Action Group) for boundary scanning and debugging.
7. **Operating Conditions:**
    - Voltage supply typically ranging from 2.0V to 3.6V.
    - Operating temperature range specified by the manufacturer.
8. **Package:**
    - Available in various package options (e.g., LQFP, UFQFPN) with different pin counts and form factors.

Libraries and Frameworks:

1. **STM32Cube HAL (Hardware Abstraction Layer):**
    - Provides a high-level API for STM32 microcontrollers.
    - Abstraction of hardware details for easier software development.
    - Offers APIs for GPIOs, timers, communication protocols (SPI, I2C, UART), and more.
2. **CMSIS (Cortex Microcontroller Software Interface Standard):**
    - ARM's standardized interface for Cortex-M processors.
    - Offers core access and peripheral drivers.
    - Provides standardized APIs for device configuration and access.
3. **ST Peripheral Drivers:**
    - STM32 microcontrollers have individual peripheral libraries, providing detailed APIs for specific peripherals like ADC, DAC, DMA, etc.

Features
• Dynamic Efficiency Line with BAM (Batch
Acquisition Mode)
– 1.7 V to 3.6 V power supply
– - 40°C to 85/105/125 °C temperature range
• Core: Arm® 32-bit Cortex®-M4 CPU with FPU,
Adaptive real-time accelerator (ART
Accelerator™) allowing 0-wait state execution

from Flash memory, frequency up to 100 MHz, memory protection unit,
125 DMIPS/1.25 DMIPS/MHz (Dhrystone 2.1), and DSP instructions
• Memories
– Up to 512 Kbytes of Flash memory
– 128 Kbytes of SRAM
• Clock, reset and supply management
– 1.7 V to 3.6 V application supply and I/Os
– POR, PDR, PVD and BOR
– 4-to-26 MHz crystal oscillator
– Internal 16 MHz factory-trimmed RC
– 32 kHz oscillator for RTC with calibration
– Internal 32 kHz RC with calibration
• Power consumption
– Run: 100 µA/MHz (peripheral off)
– Stop (Flash in Stop mode, fast wakeup time): 42 µA Typ @ 25C; 65 µA max @25 °C
– Stop (Flash in Deep power down mode, slow wakeup time): down to 9 µA @ 25 °C; 28 µA max @25 °C
– Standby: 1.8 µA @25 °C / 1.7 V without RTC; 11 µA @85 °C @1.7 V
– VBAT supply for RTC: 1 µA @25 °C
• 1×12-bit, 2.4 MSPS A/D converter: up to 16 channels
• General-purpose DMA: 16-stream DMA controllers with FIFOs and burst support
• Up to 11 timers: up to six 16-bit, two 32-bit timers up to 100 MHz, each with up to four IC/OC/PWM or pulse counter and quadrature (incremental) encoder input, two watchdog timers (independent and window) and a SysTick time.

Debug mode
– Serial wire debug (SWD) & JTAG interfaces

– Cortex®-M4 Embedded Trace Macrocell™
• Up to 81 I/O ports with interrupt capability
– Up to 78 fast I/Os up to 100 MHz
– Up to 77 5 V-tolerant I/Os
• Up to 13 communication interfaces
– Up to 3 x I2C interfaces (SMBus/PMBus)
– Up to 3 USARTs (2 x 12.5 Mbit/s, 1 x 6.25 Mbit/s), ISO 7816 interface, LIN, IrDA, modem control)
– Up to 5 SPI/I2Ss (up to 50 Mbit/s, SPI or I2S audio protocol), SPI2 and SPI3 with muxed full-duplex I2S to achieve audio class accuracy via internal audio PLL or external clock
– SDIO interface (SD/MMC/eMMC)
– Advanced connectivity: USB 2.0 full-speed device/host/OTG controller with on-chip PHY