

Alunos: Guilherme Amorim, Matheus Silva, Thiago Santos

Tabela 1. Mapa de memória:

Faixa de endereços (hex)	Tamanho da mem. (bytes)	Utilidade da mem	Nome do componente vhd de memória	Tecnologia da mem
0x0000 0FFF – 0x0000 0C00	1024 B	Periféricos mapeados	Mem_mapped	RAM
0x0000 0BFF – 0x0000 0800	1024 B	Memória de dados estática	Mem_data	RAM
0x0000 07FF – 0x0000 0400	1024 B	Memória de programa	Mem_program	FLASH
0x0000 03FF – 0x0000 0000	1024 B	Memória de dados dinâmica (<i>stack e heap</i>)	Mem_data	RAM

Tabela2. Detalhamento da região de periféricos do mapa de memória:

Endereços (hex)	Nome do registrador	Periférico	Mapa em <i>bits</i> do registrador
0x0000 0ED0	interrupt_ctl_interrupt	interrupt_ctl	0
0x0000 0ED0	interrupt_ctl_Acknowledge	interrupt_ctl	1
0x0000 0ED0	interrupt_ctl_Clear_pending	interrupt_ctl	2
0x0000 0ED1	timer_enable_counter_burst	timer	0
0x0000 0ED2	timer_counter_burst_value	timer	32 – 0
0x0000 0ED6	timer_data	timer	32 – 0
0x0000 0F36	gpio_we	GPIO	0
0x0000 0F36	gpio_addr	GPIO	1
0x0000 0F37	gpio_data_i	GPIO	32 – 0
0x0000 0F3B	gpio_data_o	GPIO	32 – 0
0x0000 0F3F	gpio_port_dir	GPIO	32 – 0
0x0000 0F9B	rden_i	UART	0
0x0000 0F9B	wren_i	UART	1

0x0000 0F9B	ack_o	UART	2
0x0000 0F9B	clkgen_en_o	UART	3
0x0000 0F9B	uart_txd_o	UART	4
0x0000 0F9B	uart_rxd_i	UART	5
0x0000 0F9B	uart_rts_o	UART	6
0x0000 0F9B	uart_cts_i	UART	7
0x0000 0F9C	addr_i	UART	32 – 0
0x0000 0FA0	clkgen_i	UART	8 – 0

Observação: o exemplo acima considera que as palavras (e também os registradores) são todos de 32 bits de tamanho (4 bytes) e que o endereçamento dessa CPU é por byte, ou seja, cada byte tem o seu próprio endereço. Portanto, na região de memória reservada para o mapeamento dos registradores dos periféricos, cabe até 64 registradores de 32 bits. Preencha os detalhes da tabela 2, somente quando for implementar os periféricos do seu MCU.

Etapa 1: Definição de requisitos do projeto

• Quais instruções sua CPU poderá processar, ou seja, o início da definição da ISA:

1. **Arithmetic Instructions:** add, subtract, add immediate;
2. **Logical Instructions:** shift left logical e shift right logical;
3. **Data transfer Instructions:** load word, store word, load byte, store byte;
4. **Conditional branch Instruction:** branch on equal, branch on not equal, set on less than, set less than immediate;
5. **Jump: Instructions:** jump, jump register, jump and link;
6. **Interrupton Instruction:** syscall;
7. Nop (no operation).

• Tamanho(s) da instrução:

32 bits.

• Tamanho do(s) dado(s) que a sua CPU será capaz de processar:

32 bits.

- **Capacidade de memória que a sua CPU será capaz de endereçar (tem a ver com o PC):**

Considerando um *program counter* de 32 bits, no qual somente os 12 bits menos significativos são utilizados no endereçamento:

$$\text{Capacidade de memória} = 2^{12}[\text{bytes}] = 4096[\text{bytes}] \sim 4KB$$

- **Formas de endereçamento que a sua CPU será capaz de tratar:**

- Endereçamento direto (Direct addressing/ Absolute addressing Mode):

- **Direct addressing/ Absolute addressing Mode (symbol []):**

The operand's offset is given in the instruction as an 8 bit or 16 bit displacement element. In this addressing mode the 16 bit effective address of the data is the part of the instruction.

Here only one memory reference operation is required to access the data.



- **Formas de E/S que a sua CPU será capaz de tratar:**

Polling e Interrupção.

- **Priorizará ou não o uso de banco de registradores no processamento dos dados?**

Sim.

- **Modelo RISC ou Modelo CISC:**

RISC.

- **Modelo Von Neumann ou Modelo Harvard:**

Modelo Von Neumann.

- **Modelo de CPU: ciclo único, multiciclo ou pipeline simples?**

Ciclo único, mas a ideia é posteriormente fazer o pipeline.

- **Endianess – ordenamento de bytes dentro da palavra (little endian ou big endian):**

Little endian.