

Alunos: Guilherme Amorim, Matheus Silva, Thiago Santos

Tabela 1. Exemplo de um mapa de memória:

| Faixa de endereços (hex) | Tamanho da mem. (bytes) | Utilidade da mem | Nome do componente vhd de memória | Tecnologia da mem |
|---------------------------|-------------------------|---|-----------------------------------|-------------------|
| 0x0000 0B00 – 0x0000 0700 | 1024 B | Memória de dados dinâmica (<i>stack e heap</i>) | Mem_data | RAM |
| 0x0000 06FF – 0x0000 0300 | 1024 B | Memória de dados estática | Mem_data | RAM |
| 0x0000 02FF – 0x0000 0100 | 512 B | Memória de programa | Mem_program | FLASH |
| 0x0000 00FF – 0x0000 0000 | 256 B | Periféricos mapeados | Mem_mapped | RAM |

Tabela2. Exemplo de um detalhamento da região de periféricos do mapa de memória:

| Endereços (hex) | Nome do registrador | Periférico | Mapa em <i>bits</i> do registrador |
|-----------------|---------------------|------------|------------------------------------|
| 0x0000 00C0 | | | |
| 0x0000 0080 | | | |
| 0x0000 0040 | | | |
| 0x0000 0000 | | | |

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.

• 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, load upper immediate;
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. **Interrupt 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 por imediato (Immediate addressing mode);

• **Immediate addressing mode (symbol #):** In this mode data is present in address field of instruction. Designed like one address instruction format.

- Endereçamento por registrador (Register mode);

- **Register mode:** In register addressing the operand is placed in one of 8 bit or 16 bit general purpose registers. The data is in the register that is specified by the instruction.

Here one register reference is required to access the data.



Example: MOV AX,CX (move the contents of CX register

- 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.



- Endereçamento relativo ao PC (PC relative addressing mode);

- **PC relative addressing mode:** PC relative addressing mode is used to implement intra segment transfer of control, In this mode effective address is obtained by adding displacement to PC.

$EA = PC + \text{Address field value}$

$PC = PC + \text{Relative value.}$

- Endereçamento a partir de um registrador base (Base register addressing mode);

- **Base register addressing mode:** Base register addressing mode is used to implement inter segment transfer of control. In this mode effective address is obtained by adding base register value to address field value.

$EA = \text{Base register} + \text{Address field value.}$

$PC = \text{Base register} + \text{Relative value.}$

- 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.

Referências:

- <https://www.geeksforgeeks.org/addressing-modes/>