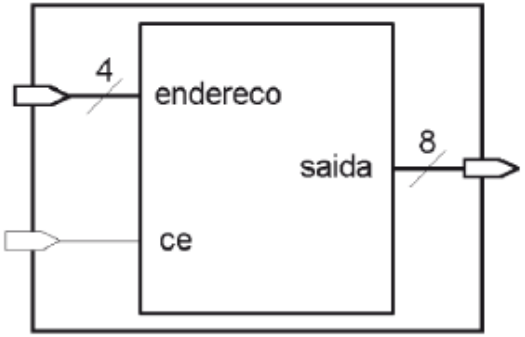
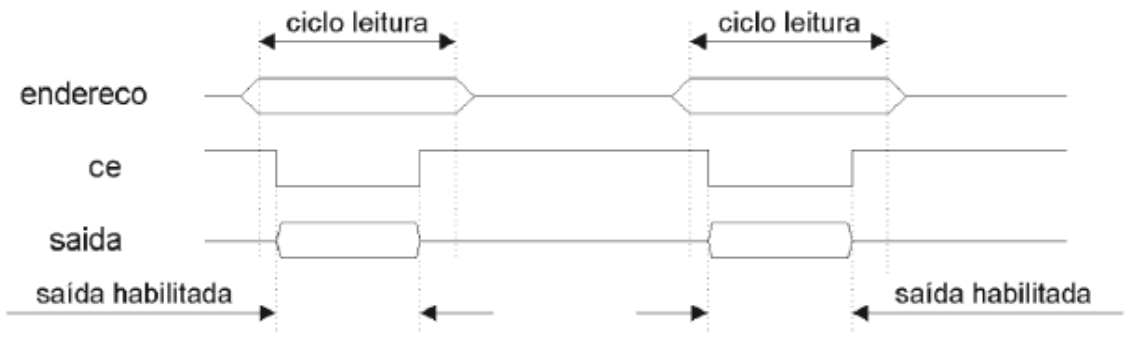
Aula 10 – Memórias

Tópicos da aula

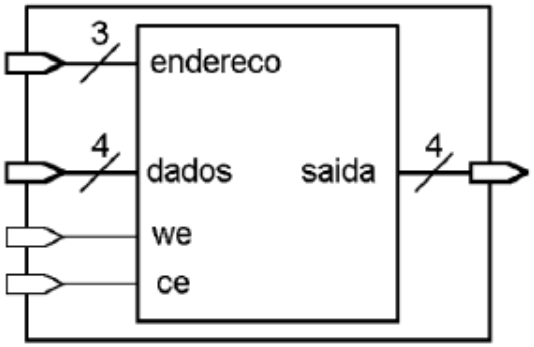
- **Read-only memory (ROM)**
- **Random-acces memory (RAM)**

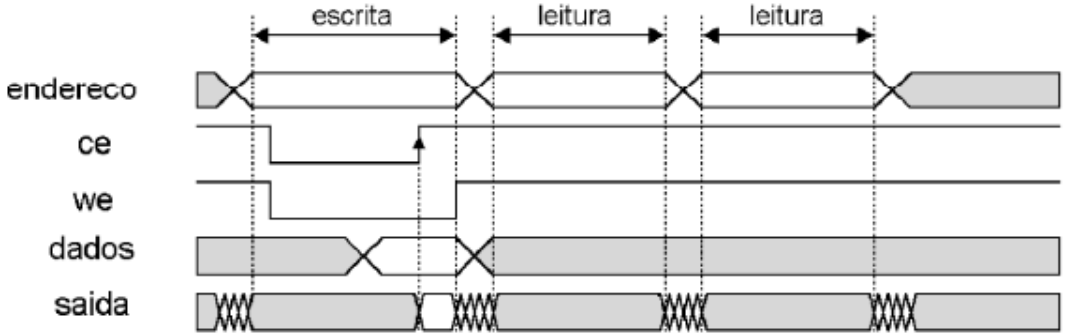
ROM


4
endereco
saida
8
ce


ciclo leitura
ciclo leitura
endereco
ce
saida
saída habilitada
saída habilitada

RAM


3
endereco
4
dados
saida
4
we
ce


escrita
leitura
leitura
endereco
ce
we
dados
saida

Linguagem de descrição de hardware


```
1   library IEEE;
2   use IEEE.STD_LOGIC_1164.ALL;
3   use IEEE.STD_LOGIC_arith.ALL;
4   use IEEE.STD_LOGIC_unsigned.ALL;
5
6   entity MEMORIA is
7       Generic(
8               p_DATA_WIDTH    : INTEGER := 16;        -- Número de bits dos dados.
9               p_ADD_WIDTH     : INTEGER := 6          -- Número de bits dos endereços.
10      );
11      Port (
12            i_CLK     : in  STD_LOGIC;
13            i_DATA    : in  STD_LOGIC_VECTOR ((p_DATA_WIDTH-1) downto 0);
14            i_WE      : in  STD_LOGIC;
15            i_ADDR    : in  STD_LOGIC_VECTOR ((p_ADD_WIDTH-1) downto 0);
16            i_ADDW    : in  STD_LOGIC_VECTOR ((p_ADD_WIDTH-1) downto 0);
17            o_DATA    : out STD_LOGIC_VECTOR ((p_DATA_WIDTH-1) downto 0)
18      );
19  end MEMORIA;
20
21  architecture Behavioral of MEMORIA is
22
23      type MEM_TYPE is array(i_ADDR'range) of std_logic_vector(i_DATA'range);
24      signal w_MEMORIA_RAM : MEM_TYPE;
25
26  begin
27
28          -- Process de escritura
29          process(i_CLK)  begin
30              if rising_edge(i_CLK)  then
31                  if (i_WE = '1')  then
32                      w_MEMORIA_RAM(conv_integer(i_ADDW)) <= i_DATA;
33                  end if;
34
35                  o_DATA <= w_MEMORIA_RAM(conv_integer(i_ADDR));
36              end if;
37          end process;
38
39  end Behavioral;
```

FIM AULA X