Laboratório

RAM - ROM

RAM Assíncrona

```
entity SingleRAM_asyn is
   Generic(
      ADDR_WIDTH: integer:=12;
      DATA_WIDTH: integer:=8
   );
    Port ( clk : in  STD_LOGIC;
           we : in  STD_LOGIC;
           addr : in  STD_LOGIC_VECTOR (ADDR_WIDTH-1 downto 0);
           din : in  STD_LOGIC_VECTOR (DATA_WIDTH-1 downto 0);
           dout : out  STD_LOGIC_VECTOR (DATA_WIDTH-1 downto 0));
end SingleRAM_asyn;

architecture ram_arch of SingleRAM_asyn is
   type ram_type is array (2**ADDR_WIDTH-1 downto 0)
        of std_logic_vector (DATA_WIDTH-1 downto 0);
   signal ram : ram_type;
begin
   process (clk)
   begin
      if (clk'event and clk = '1') then
         if (we='1') then
            ram(to_integer(unsigned(addr))) <= din ;
         end if;
      end if;
   end process;
   dout <= ram(to_integer(unsigned(addr_reg)));
end ram_arch;
```

RAM Síncrona

```
entity SingleRAM_syn is
   Generic(
      ADDR_WIDTH: integer:=12;
      DATA_WIDTH: integer:=8
   );
    Port ( clk : in  STD_LOGIC;
           we : in  STD_LOGIC;
           addr : in  STD_LOGIC_VECTOR (ADDR_WIDTH-1 downto 0);
           din : in  STD_LOGIC_VECTOR (DATA_WIDTH-1 downto 0);
           dout : out  STD_LOGIC_VECTOR (DATA_WIDTH-1 downto 0));
end SingleRAM_syn;

architecture ram_arch of SingleRAM_syn is
   type ram_type is array (2**ADDR_WIDTH-1 downto 0)
        of std_logic_vector (DATA_WIDTH-1 downto 0);
   signal ram : ram_type;
   signal addr_reg: std_logic_vector (ADDR_WIDTH-1 downto 0);
begin
   process (clk)
   begin
      if (clk'event and clk = '1') then
         if (we='1') then
            ram(to_integer(unsigned(addr))) <= din ;
         end if;
         addr_reg <= addr;
      end if;
   end process;
   dout <= ram(to_integer(unsigned(addr_reg)));
end ram_arch;
```

ROM Assíncrona

```
entity Rom_asyn is
   port(
      addr: in std_logic_vector(3 downto 0);
      data: out std_logic_vector(6 downto 0)
   );
end Rom_asyn;

architecture arch_rom of Rom_asyn is
   constant ADDR_WIDTH: integer:=4;
   constant DATA_WIDTH: integer:=7;
   type rom_type is array (0 to 2**ADDR_WIDTH-1)
      of std_logic_vector(DATA_WIDTH-1 downto 0);
   -- ROM definition
   constant DATA_ROM: rom_type:=(
      "0000000", -- addr 0000
      "1000001", -- addr 0001
      "0100010", -- addr 0010
      "1100011", -- addr 0011
      "0100001", -- addr 0100
      "0010001", -- addr 0101
      "0001001", -- addr 0110
      "0000101", -- addr 0111
      "0000011", -- addr 1000
      "0110001", -- addr 1001
      "0001101", -- addr 1010
      "1100001", -- addr 1011
      "0000000", -- addr 1100
      "0000111", -- addr 1101
      "0001111", -- addr 1110
      "1111101"  -- addr 1111
      );
begin
   data <= DATA_ROM(to_integer(unsigned(addr)));
end arch_rom;
```