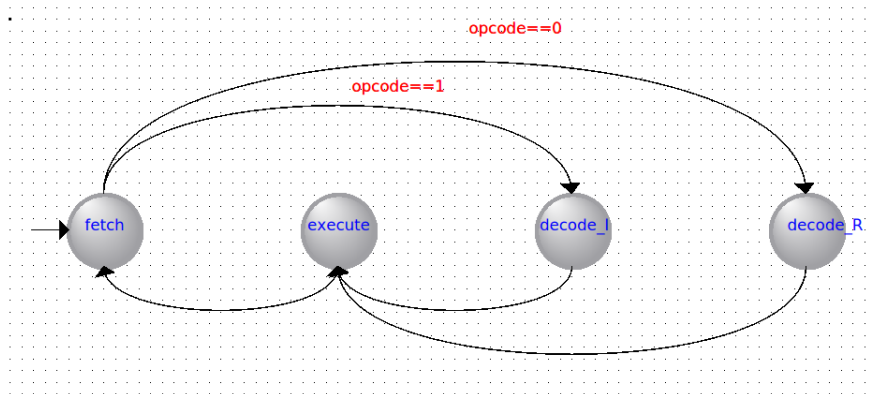


Apêndice Um Modelo de Máquina de Estados VHDL (Opcional)

No seu processador, o mais fácil é mesmo usar um contador, como descrito no laboratório #5. Os alunos que tentaram fazer uma máquina como a descrita a seguir tiveram bastante dificuldade, talvez por questões meramente conceituais.

Há formas padrão de se fazer uma máquina de estados propriamente com transições condicionais, ou seja, uma máquina de Moore. Por exemplo, vamos implementar a máquina a seguir:¹



Um dos códigos VHDL possíveis é simples² e usa uma nova estrutura sequencial “case-when”:

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity maquina is
    port( clk      : in std_logic;
          rst      : in std_logic;
          opcode   : in std_logic;
          estado   : out unsigned(1 downto 0)
    );
end entity;

architecture a_maquina of maquina is
    signal estado_s : unsigned(1 downto 0);
begin
    process(clk, rst)
    begin
        if rst='1' then
            estado_s <= "00";
        elsif rising_edge(clk) then
            case estado_s is
                when "00" => -- fetch
                    if opcode='0' then
                        estado_s <= "01";
                    else
                        estado_s <= "10";
                    end if;
            end case;
        end if;
    end process;
end architecture;
  
```

- 1 No Quartus II, você pode fazer a máquina de estados desenhando-a com *File => New... => State Machine File*, ou ainda fazê-la com tabelas e propriedades através de um *Wizard* (menu do botão direito). Depois é só mandar gerar o fonte VHDL. Tem um *caveat*: as condições e sintaxe não são VHDL, são Verilog >:(
- 2 Alternativamente você pode criar um .vhd e usar *Edit => Insert Template...* para usar arquivos padrão de máquinas de estados.

```

        end if;
    when "01" => -- decode_R
        estado_s <= "11";
    when "10" => -- decode_I
        estado_s <= "11";
    when "11" => -- execute
        estado_s <= "00";
    when others => -- cobre casos como "UU" ou "XX"
        null;
    end case;
end if;
end process;
estado <= estado_s;
end architecture;

```

Perceba que a transição condicional no estado “00” (*fetch*) é feita com um simples *if*. Lembre-se no entanto, **lembre-se**, de que o *if* deve ser usado apenas dentro de um *process*, ou seja, quando houver *flip-flops* na jogada (código sequencial). Usar *if* pra fazer blocos de portas lógicas é uma fonte de onde jorram *bugs* para aprendizes da linguagem.³

Uma máquina de estados VHDL mais típica, no entanto, definiria um novo tipo de dados ao invés de usar o estado numérico:

```

type type_fstate is (fetch,write_back,formato_I,formato_R);

```

E depois é só criar sinais do tipo “type_fstate.” Mas isso não é necessário.

³ Evite usar *process* a não ser que seja necessário mesmo, e faça portas lógicas e *muxes* com a estrutura *when-else*