

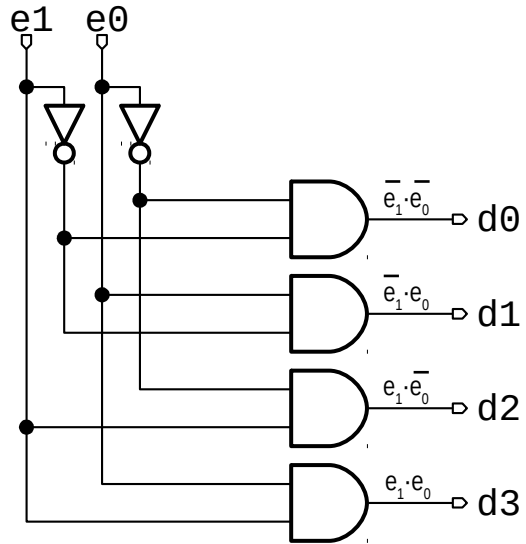
# Diseño de circuitos digitales con VHDL

## *3. Circuitos combinacionales: Decodificador*

Diseño de Sistemas Electrónicos

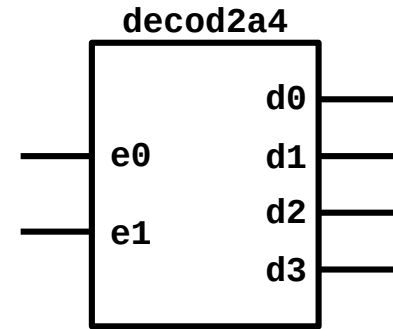
Felipe Machado

# Decodificador

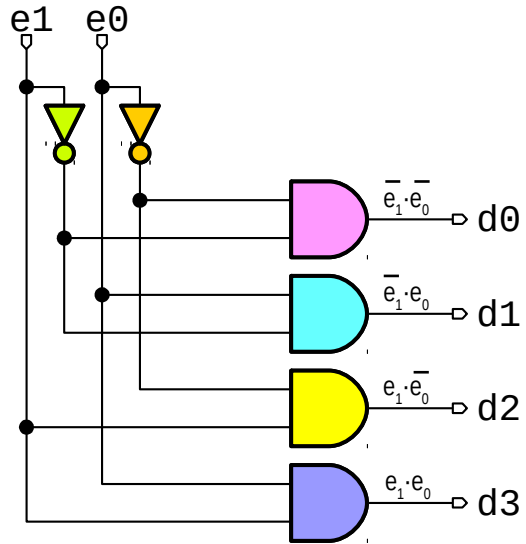


```
entity decod2a4 is
  port (
    e0 : in std_logic;
    e1 : in std_logic;
    d0 : out std_logic;
    d1 : out std_logic;
    d2 : out std_logic;
    d3 : out std_logic);
end decod2a4;
```

```
architecture gate of decod2a4 is
  signal e0n, e1n : std_logic;
begin
  e0n <= not e0;
  e1n <= not e1;
  d0 <= e1n and e0n; --00
  d1 <= e1n and e0;  --01
  d2 <= e1  and e0n; --10
  d3 <= e1  and e0;  --11
end gate;
```

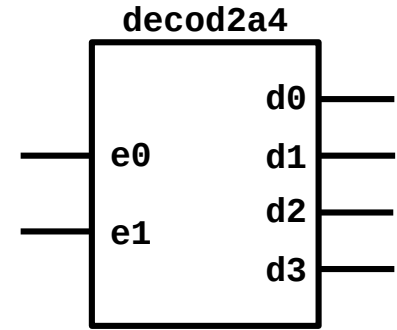


# Decodificador

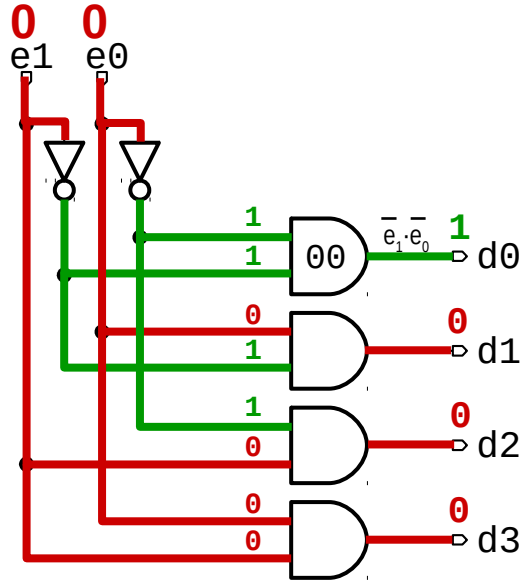


```
entity decod2a4 is
  port (
    e0 : in std_logic;
    e1 : in std_logic;
    d0 : out std_logic;
    d1 : out std_logic;
    d2 : out std_logic;
    d3 : out std_logic);
end decod2a4;
```

```
architecture gate of decod2a4 is
  signal e0n, e1n : std_logic;
begin
  e0n <= not e0;
  e1n <= not e1;
  d0 <= e1n and e0n; --00
  d1 <= e1n and e0;  --01
  d2 <= e1  and e0n; --10
  d3 <= e1  and e0;  --11
end gate;
```

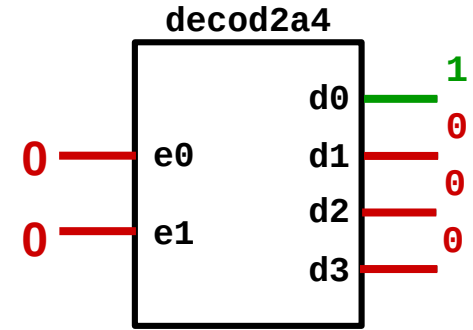


# Decodificador

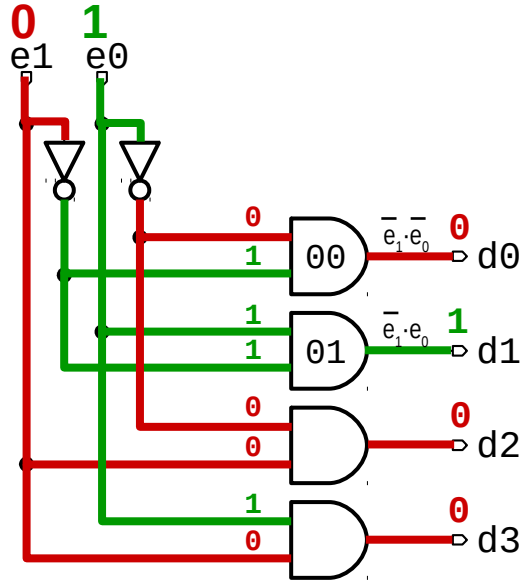


```
entity decod2a4 is
  port (
    e0 : in std_logic;
    e1 : in std_logic;
    d0 : out std_logic;
    d1 : out std_logic;
    d2 : out std_logic;
    d3 : out std_logic);
end decod2a4;
```

```
architecture gate of decod2a4 is
  signal e0n, e1n : std_logic;
begin
  e0n <= not e0;
  e1n <= not e1;
  d0 <= e1n and e0n; --00
  d1 <= e1n and e0;  --01
  d2 <= e1  and e0n; --10
  d3 <= e1  and e0;  --11
end gate;
```

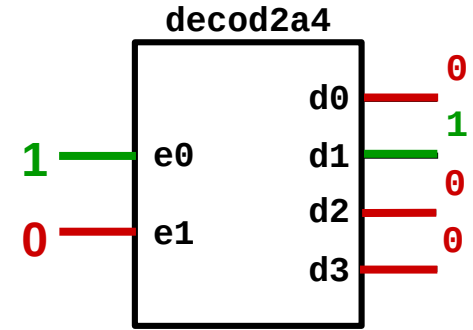


# Decodificador

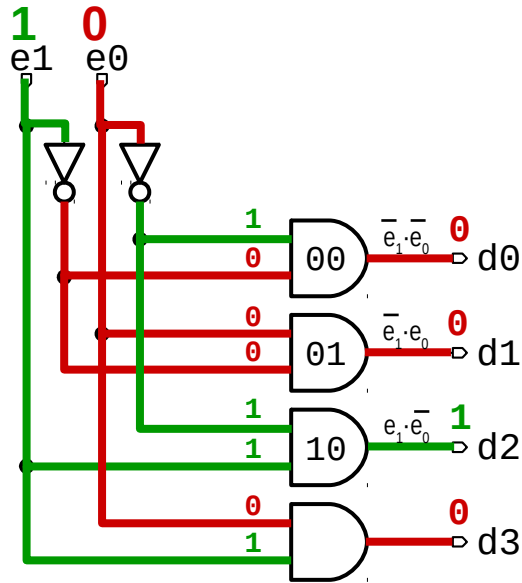


```
entity decod2a4 is
  port (
    e0 : in std_logic;
    e1 : in std_logic;
    d0 : out std_logic;
    d1 : out std_logic;
    d2 : out std_logic;
    d3 : out std_logic);
end decod2a4;
```

```
architecture gate of decod2a4 is
  signal e0n, e1n : std_logic;
begin
  e0n <= not e0;
  e1n <= not e1;
  d0 <= e1n and e0n; --00
  d1 <= e1n and e0;  --01
  d2 <= e1  and e0n; --10
  d3 <= e1  and e0;  --11
end gate;
```

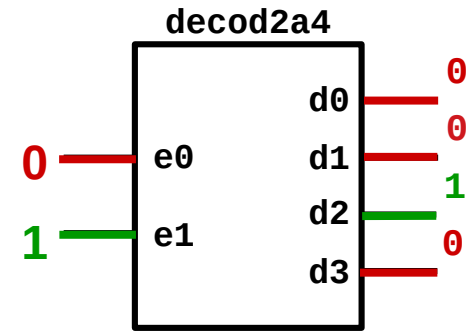


# Decodificador

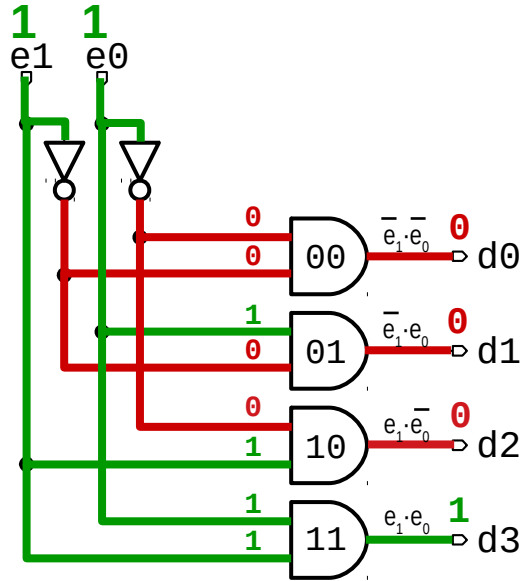


```
entity decod2a4 is
  port (
    e0 : in std_logic;
    e1 : in std_logic;
    d0 : out std_logic;
    d1 : out std_logic;
    d2 : out std_logic;
    d3 : out std_logic);
end decod2a4;
```

```
architecture gate of decod2a4 is
  signal e0n, e1n : std_logic;
begin
  e0n <= not e0;
  e1n <= not e1;
  d0 <= e1n and e0n; --00
  d1 <= e1n and e0;  --01
  d2 <= e1  and e0n; --10
  d3 <= e1  and e0;  --11
end gate;
```

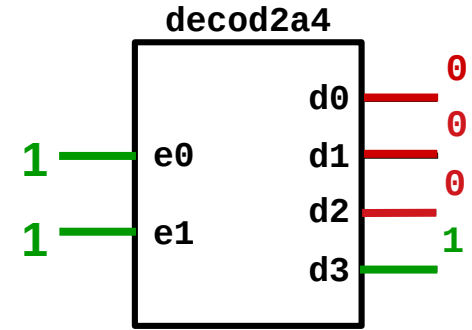


# Decodificador

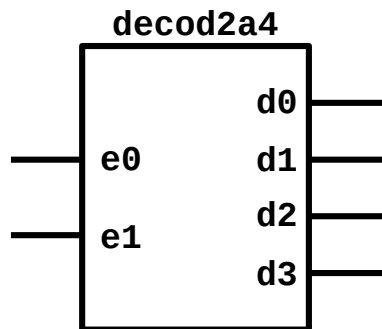
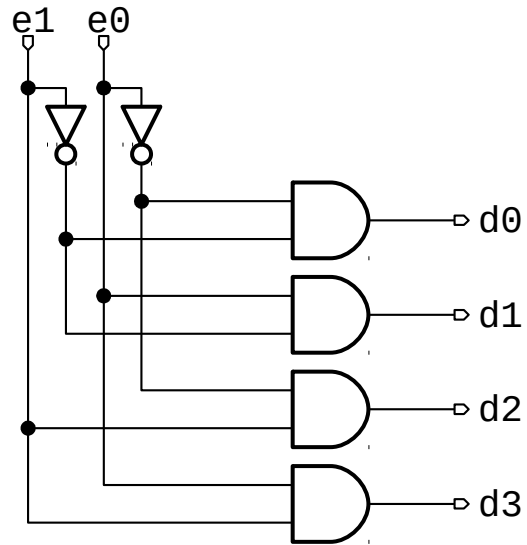


```
entity decod2a4 is
  port (
    e0 : in std_logic;
    e1 : in std_logic;
    d0 : out std_logic;
    d1 : out std_logic;
    d2 : out std_logic;
    d3 : out std_logic);
end decod2a4;
```

```
architecture gate of decod2a4 is
  signal e0n, e1n : std_logic;
begin
  e0n <= not e0;
  e1n <= not e1;
  d0 <= e1n and e0n; --00
  d1 <= e1n and e0;  --01
  d2 <= e1  and e0n; --10
  d3 <= e1  and e0;  --11
end gate;
```



# Vectores

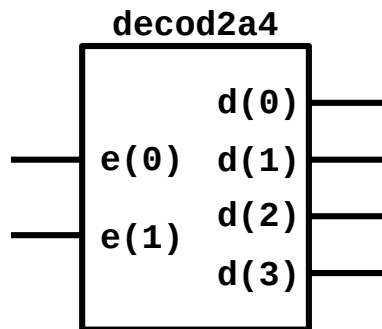
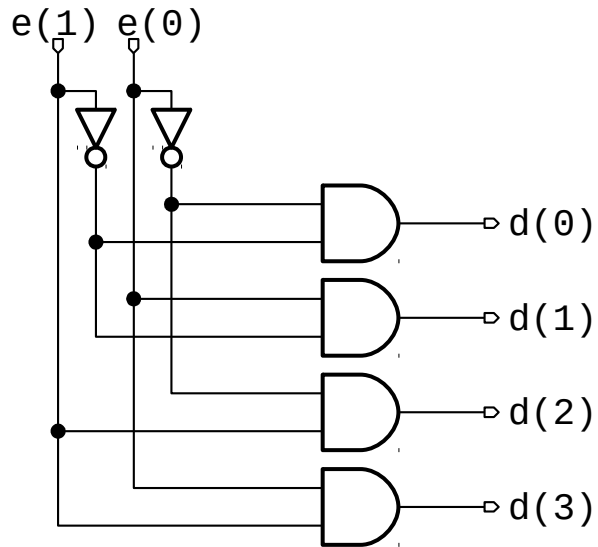


```
entity decod2a4 is
  port (
    e0 : in std_logic;
    e1 : in std_logic;
    d0 : out std_logic;
    d1 : out std_logic;
    d2 : out std_logic;
    d3 : out std_logic);
end decod2a4;
```

```
architecture gate of decod2a4 is
  signal e0n, e1n : std_logic;
begin
  e0n <= not e0;
  e1n <= not e1;
  d0 <= e1n and e0n; --00
  d1 <= e1n and e0; --01
  d2 <= e1 and e0n; --10
  d3 <= e1 and e0; --11
end gate;
```



# Vectores



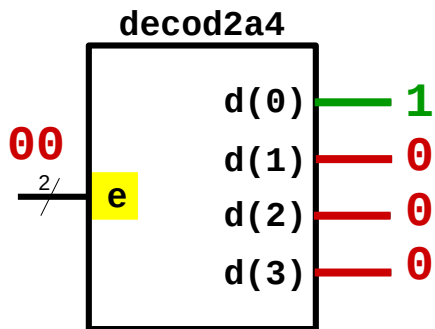
```
entity decod2a4 is
  port (
    e : in std_logic_vector(1 downto 0);
    d : out std_logic_vector(3 downto 0));
end decod2a4;
```

```
architecture gate of decod2a4 is
  signal e_n: std_logic_vector(1 downto 0);
begin
  e_n <= not e;
  d(0) <= e_n(1) and e_n(0); --00
  d(1) <= e_n(1) and e(0); --01
  d(2) <= e(1) and e_n(0); --10
  d(3) <= e(1) and e(0); --11
end gate;
```

Siempre usar **downto**  
y terminar en **0**

Hace operación bit a bit:  
e\_n(0) <= not e(0);  
e\_n(1) <= not e(1);

# Decodificador comportamental



```
entity decod2a4 is
  port (
    e : in std_logic_vector(1 downto 0);
    d : out std_logic_vector(3 downto 0));
end decod2a4;
```

## Sentencia concurrente con *select*

```
architecture behav1 of decod2a4 is
begin
  with e select
    d <= "0001" when "00",
        "0010" when "01",
        "0100" when "10",
        "1000" when others;
end behav1;
```

## Proceso con *case*

```
architecture behav2 of decod2a4 is
begin
  p_decod: process(e)
  begin
    case e is
      when "00" =>
        d <= "0001";
      when "01" =>
        d <= "0010";
      when "10" =>
        d <= "0100";
      when others =>
        d <= "1000";
    end case;
  end process;
end behav2;
```

# Decodificador

