REGISTRO

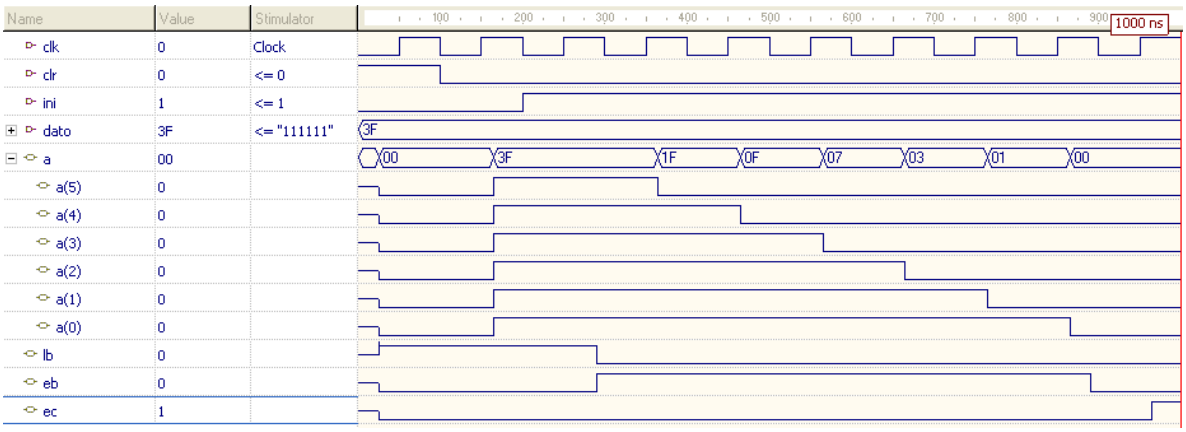
```
library ieee;
use ieee.std_logic_1164.all;

entity registro is port (
    dato : in std_logic_vector (5 downto 0);
    clk, clr, ini : in std_logic;
    a: out std_logic_vector (5 downto 0);
    lb, eb, ec : out std_logic);
end entity;

architecture aregistro of registro is
    signal la, ea, z: std_logic;
    type estados is (e0, e1, e2);
    signal act, sig : estados;
begin
    process (clk, clr) begin
        if (clr = '1') then
            act <= e0;
        elsif (rising_edge(clk)) then
            act <= sig;
        end if;
    end process;
    process (act, z, a , ini) begin
        la <= '0';
        lb <= '0';
        ea <= '0';
        eb <= '0';
        ec <= '0';
        z <= a(5) or a(4) or a(3) or a(2) or a(1) or a(0);
        case act is
            when e0 => lb <= '1';
                if (ini = '1') then sig <= e1;
                else --detiene contador
                    la <= '1';
                    sig <= e0;
                end if;
            when e1 => ea <= '1';
                if (z='1' and a(0)='1') then
                    eb <= '1'; --aumenta contador
                    sig <= e1;
                elsif (z='1' and a(0)='0') then sig <= e1;
                else sig <= e2; --termina contador
                end if;
            when others => ec <= '1'; --muestra contador
```

```
                if (ini = '1') then sig <= e2;
                else sig <= e0; --reinicia contador
            end if;
        end case;
    end process;
    process (clk, clr) begin
        if (clr = '1') then a <= "000000";
        elsif (rising_edge(clk)) then --retencion
            if (ea = '0' and la = '1') then a <= dato;
            elsif (ea = '1' and la = '0') then --corrimiento
            a(4 downto 0) <= a(5 downto 1);
            a(5) <= '0';
            end if;
        end if;
    end process;
end architecture;
```

SIMULACION



FUNCIONAMIENTO

- Empezando por poner el clear en uno para inicializar los valores por defecto
- Después se pone el ini en cero para asignar al arreglo “a” el valor del dato de entrada
- Una vez que “a” ya está inicializado se procede a poner ini en cero para ir recorriendo
- Cuando “a” solo tiene ceros en su arreglo se manda la señal para encender el display

NOTAS

- Observe que hasta que el arreglo empiece a recorrerse eb se enciende y lb se apaga
- Cuando ya no hay ceros en el arreglo eb se apaga para que ec se encienda
- En cualquier momento se puede encender el clear para prender lb y apagador los otros

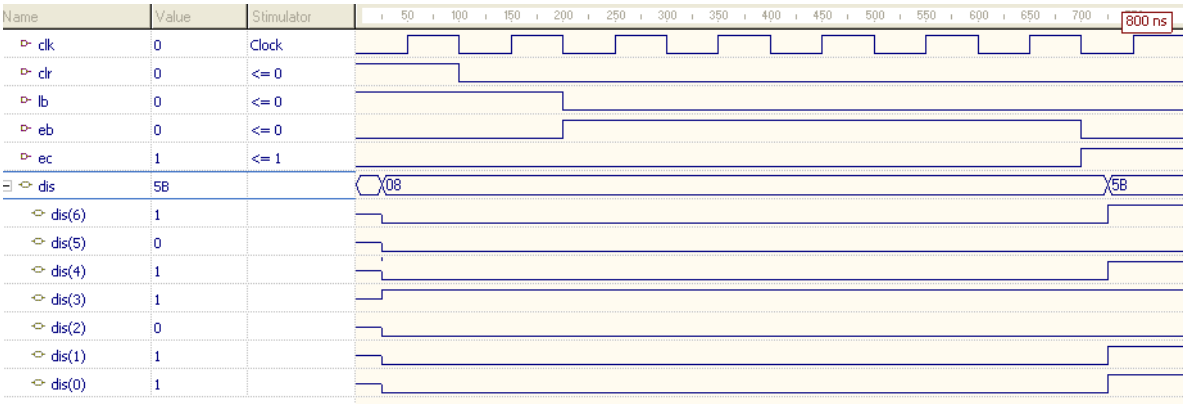
DECODIFICADOR

```
library ieee;
use ieee.std_logic_1164.all;

entity pract11 is port (
    clk, clr, lb, eb, ec : in std_logic;
    dis: out std_logic_vector (6 downto 0));
end entity;

architecture apract11 of pract11 is
    type estados is (e0, e1, e2, e3, e4, e5, e6);
    signal est : estados;
begin
    process (clr, clk) begin
        if(clr='1') then
            est <= e0;
        elsif (rising_edge(clk)) then
            if (lb = '1' and eb = '0') then est <= e0;
            elsif (lb = '0' and eb = '1') then case est is
                    when e0 => est <= e1;
                    when e1 => est <= e2;
                    when e2 => est <= e3;
                    when e3 => est <= e4;
                    when e4 => est <= e5;
                    when others => est <= e6;
                end case;
            else est <= est;
            end if;
        end if;
    end process;
    --asigna valor al display
    process (ec, est) begin
        if(ec = '0') then dis <= "0001000";
        else case est is
                when e0 => dis <= "1111110";
                when e1 => dis <= "0011000";
                when e2 => dis <= "1101101";
                when e3 => dis <= "1111001";
                when e4 => dis <= "0110011";
                when e5 => dis <= "1011011";
                when others => dis <= "1011111";
            end case;
        end if;
    end process;
end architecture;
```

SIMULACION



FUNCIONAMIENTO

- Se pone el clear en uno y todos los demás valores en cero
- Apaga el clear y prende lb para empezar desde el primer estado (0)
- Enciende eb y apaga lb para ir pasando por estados (0,1,2,3,4,5,6)
- Cuantas veces hayas simulado será el resultado en el display (máximo 5F=6)

CUESTIONARIO

1. ¿Cuántos dispositivos PLD 22V10 son necesarios para el desarrollo de esta práctica? R= 2

2. ¿Cuántos dispositivos de la serie 74xx (TTL) ó 40xx (CMOS) hubieras necesitado para el desarrollo de esta práctica? R= aproximadamente 50

3. ¿Cuántos pines de entrada/salida de cada PLD 22V10 se usan en el diseño? R=

- 5 de entrada y 7 de salida en el decodificador

- 9 de entrada y 9 de salida

4. ¿Cuántos términos producto ocupan las ecuaciones para cada señal de salida y que porcentaje se usa en total de los PLD 22V10? R= 25 / 121 = 20 % Y 34 / 121 = 28 %

5. ¿Cuántos FF's ocupa el autómata de control de la microarquitectura? R= 5 - 10

6. ¿Qué puedes concluir de esta práctica?

En esta practica fue posible observar el funcionamiento de un contador optimizado para implementarse en un circuito electrónico. Fue por esto que se necesitaron menos ciclos de reloj para llegar a un resultado de 0 a 6 según la cantidad y la posición de los bits en el dato de entrada.