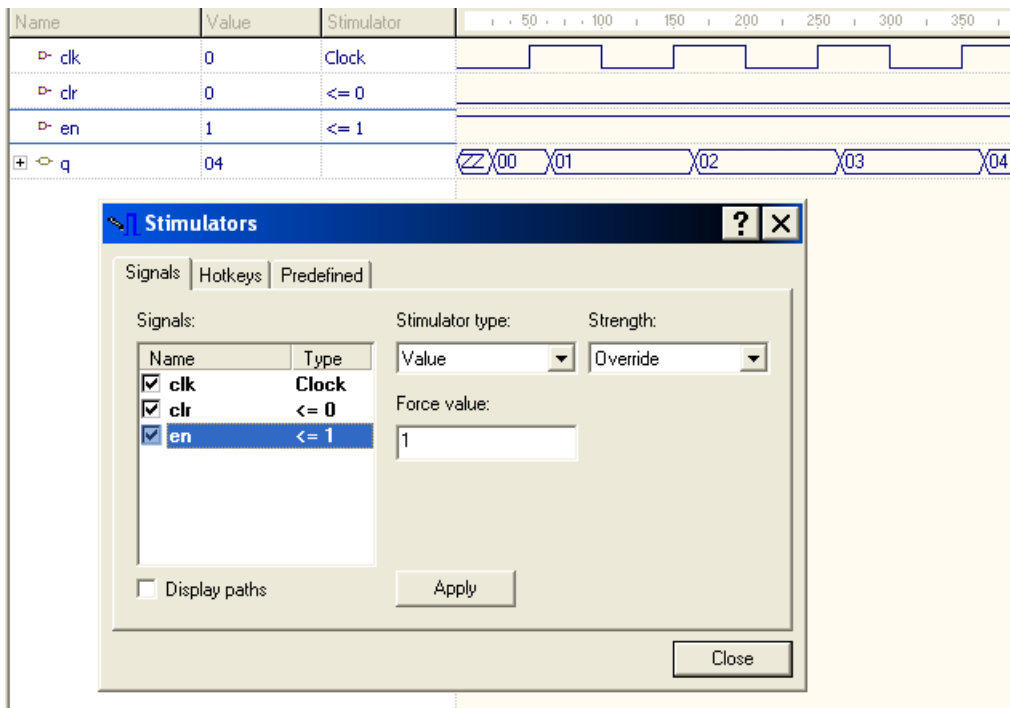


CIRCUITO 1



```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity contador is port (
    q: out std_logic_vector(7 downto 0);
    clk, clr, en: in std_logic
); end contador;
architecture acontador of contador is
    signal aux: std_logic_vector(7 downto 0);
begin process(clk, clr)
    variable aux : std_logic; begin
        if (clr = '1') then
            q <= (others => '0');
        elsif rising_edge(clk) then
            for i in 0 to 7 loop
                aux := en;
                for j in 0 to i-1 loop
                    aux := aux and q(j);
                end loop;
                q(i) <= q(i) xor aux;
            end loop;
        end if;
    end process;
end acontador;

```

CIRCUITO 2

Name	Value	Stimulator	
clk	0	Clock	
clr	0	<= 0	
en	1	<= 1	
l	0	<= 0	
ud	1	<= 1	
d	FE	<= "11111110"	FE
q	02		00 FE FF 00 01 02

Stimulators

Signals

Hotkeys

Predefined

Signals:

Name	Type
<input checked="" type="checkbox"/> clk	Clock
<input checked="" type="checkbox"/> clr	<= 0
<input checked="" type="checkbox"/> en	<= 1
<input checked="" type="checkbox"/> l	<= 0
<input checked="" type="checkbox"/> ud	<= 1
<input checked="" type="checkbox"/> d	<= "1...

Stimulator type:

Value

Strength:

Override

Force value:

0

☐ Display paths

Apply

Close

Stimulators

Signals

Hotkeys

Predefined

Signals:

Name	Type
<input checked="" type="checkbox"/> clk	Clock
<input checked="" type="checkbox"/> clr	<= 0
<input checked="" type="checkbox"/> en	<= 1
<input checked="" type="checkbox"/> l	<= 0
<input checked="" type="checkbox"/> ud	<= 0
<input checked="" type="checkbox"/> d	<= "0...

Stimulator type:

Value

Strength:

Override

Force value:

0

☐ Display paths

Apply

Close

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

entity contGen is port (
    d: in std_logic_vector(7 downto 0);
    q: out std_logic_vector(7 downto 0);
    clk, clr, en, ud, l: in std_logic
);
end contGen;

architecture aCont of contGen is
    signal aux: std_logic_vector(7 downto 0);
begin

    process(clk, clr) begin
        if (clr = '1') then
            aux <= (others => '0');
        elsif rising_edge(clk) then
            if (en = '0') then
                aux <= aux;
            else
                if (l = '1') then
                    aux <= d;
                else
                    if(ud = '0') then
                        if(aux = "00000000") then
                            aux <= "11111111";
                        else aux <= aux - 1;
                        end if;
                    else
                        if(aux = "11111111") then
                            aux <= "00000000";
                        else aux <= aux + 1;
                        end if;
                    end if;
                end if;
            end if;
        end if;
    end process;

    q <= aux;

end aCont;
```

CUESTIONARIO

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

R= Uno por circuito

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

R= 10-21 para cada uno

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

R= En el primero 3 de entrada y 8 de salida. En el segundo 12 de entrada y 8 de salida.

4. ¿Cuántos términos producto ocupan las ecuaciones para cada señal de salida y que porcentaje se usa en total del PLD 22V10?

R= En el primero $10 / 22 = 45 \%$. En el segundo $21 / 22 = 95 \%$

5. ¿Por qué se tienen que usar variables para implementar la ecuación genérica del contador con señal de control enable?

R= Para que el valor de la variable vaya cambiando cada vez que de vuelta el ciclo

6. ¿Qué nivel de diseño se implementó al usar los operadores + y – en el contador?

R= Estructurado.

7. ¿Cuáles son las señales que funcionan de manera síncrona y cuáles de manera asíncrona?

R= Síncrona clk; asíncrona clr, en, l, d, q, du, etc.

8. ¿Qué puedes concluir de esta práctica?

R= Los contadores funcionan de una manera similar a los registros, tienen una función para resetear los valores y otra para cargar un valor de entrada, con la diferencia de que pueden aumentar o disminuir su valor, en función del parámetro que introduzcamos. Cabe mencionar que estos se pueden realizar de varias maneras, la mejor opción es encontrar un patrón en su comportamiento para modelar un algoritmo que simplifique las operaciones lógicas y por lo tanto pueda funcionar en el dispositivo que estemos utilizando.