

## CONTADOR DE 0 A 4

VERSIÓN 1: SIN PROBLEMAS EN LA SIMULACIÓN

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity nueve is
    Port ( clk : in bit;
          salida : out integer range 0 to 8);
end nueve;

architecture algoritmo of nueve is
begin
process (clk)
variable temporal: integer range 0 to 8;
begin
    if clk='1'and clk'event then
        temporal:=temporal +1;
        if temporal=5 then
            temporal:=0;
        end if;
    end if;
salida<=temporal;
end process;
end algoritmo;
```

ARCHIVO DE SIMULACIÓN

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;

ENTITY testbench IS
END testbench;

ARCHITECTURE behavior OF testbench IS

    COMPONENT nueve
    PORT(
        clk : IN bit;
        salida : OUT integer range 0 to 8
    );
    END COMPONENT;

    SIGNAL clk : bit;
```

```

    SIGNAL salida : integer range 0 to 8;

BEGIN

    uut: nueve PORT MAP(
        clk => clk,
        salida => salida
    );

    clk<=not clk after 1 ms;
    tb : PROCESS
    BEGIN
        Wait;
    END PROCESS;

END;
```

## VERSIÓN 2: SIN PROBLEMAS EN SÍNTESIS, PERO CON TRUNCAMIENTO EN SIMULACIÓN

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity conta4 is
port (clk: in std_logic;
      q: out integer range 0 to 4);
end conta4;
architecture algoritmo of conta4 is
begin
    process (clk)
    variable temporal: integer range 0 to 4;
    begin
        if clk='1' and clk'event then
            temporal:=temporal+1;
        end if;
        q<=temporal;
    end process;
end algoritmo;
```

## GRÁFICA OBTENIDA DE LA VERSIÓN 1 DEL CONTADOR



Sin problemas en la simulación (no hay glitches), y en la parte de hardware, se implementa con 3 bits de salida y uno de entrada (clk), para ambos casos (VERSION 1 y VERSION 2)