

# EL253 - Sistemas Digitales

Semestre 2020-2

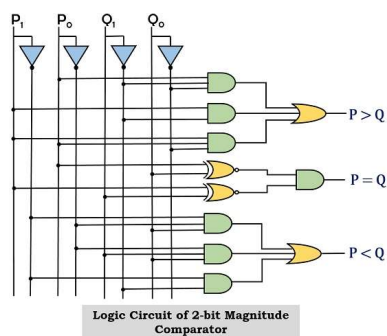
Profesor Kalun José Lau Gan

Sesión de Teoría Semana 4

1

## Preguntas previas:

- ¿Cómo sería un modelo de comparador de magnitud en VHDL?



```

1  LIBRARY ieee;
2  USE ieee.std_logic_1164.ALL;
3
4  ENTITY Lab6 IS
5      PORT(a, b : IN std_logic_vector(7 DOWNTO 0);
6            agb, aeb, alb : OUT std_logic);
7  END Lab6;
8
9  ARCHITECTURE arc OF Lab6 IS
10     SIGNAL result : std_logic_vector(2 DOWNTO 0);
11 BEGIN
12     PROCESS (a,b)
13     BEGIN
14         IF a<b THEN
15             result <= "001";
16         ELSIF a=b THEN
17             result <= "010";
18         ELSIF a>b THEN
19             result <= "100";
20         ELSE
21             result <= "000";
22         END IF;
23         agb <= result(2);
24         aeb <= result(1);
25         alb <= result(0);
26     END PROCESS;
27 END arc;

```

2

## Preguntas previas:

- ¿?

3

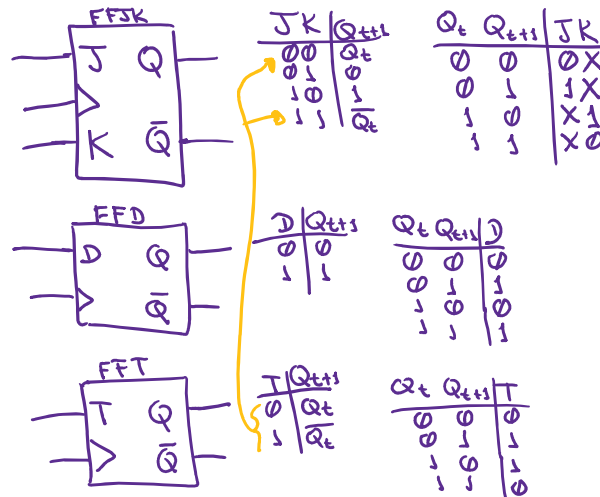
## Agenda

- Sistemas secuenciales
- Flanco vs Nivel
- Modelo de contador binario en VHDL
- Módulo divisor de frecuencia en VHDL
- Descripción estructural en VHDL

4

## Sistemas Secuenciales

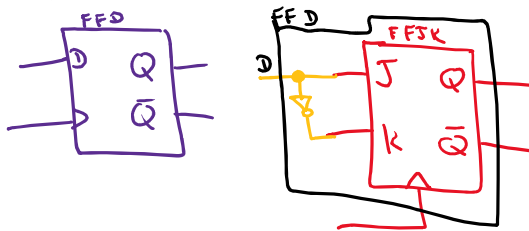
- Un nuevo escenario: el tiempo
- Flip-flops como unidad mínima de un sistema secuencial (SR, JK, T, D)



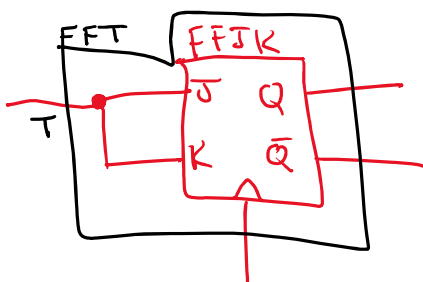
5

## Conversión entre tipos de flip flops

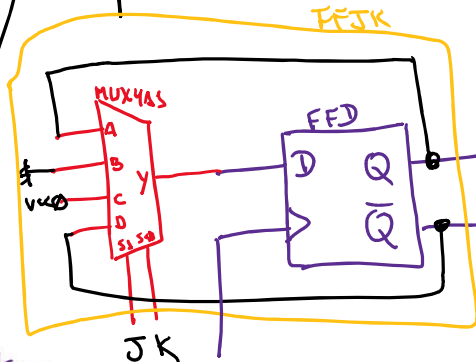
- Implementar un FFD con FFJK



- FFT con FFJK



Implementar un FFJK con FFD

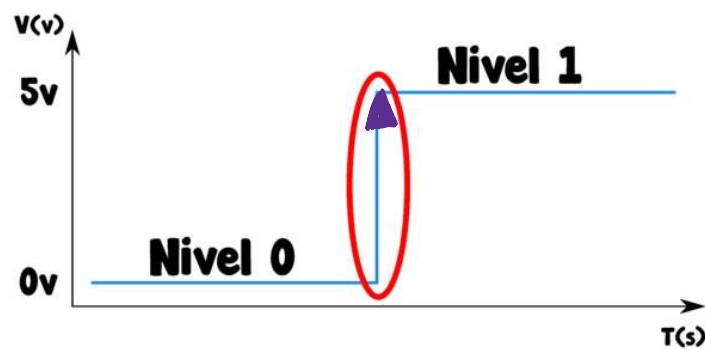


$J$	$K$	$Q_t$	$Q_{t+1}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

6

## Flanco vs nivel lógico

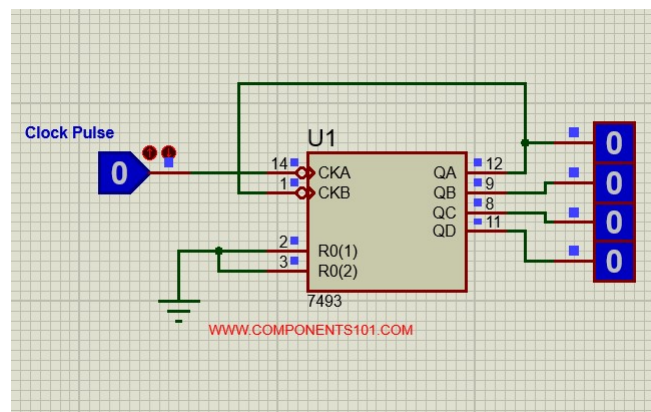
- Flanco es la transición de un nivel a otro nivel lógico
- Transición positiva (paso de nivel 0 a nivel 1)
- Transición negativa (paso de nivel 1 a nivel 0)



7

## Contadores binarios

- Circuitos secuenciales
- Tienen un registro interno donde se almacena la cuenta



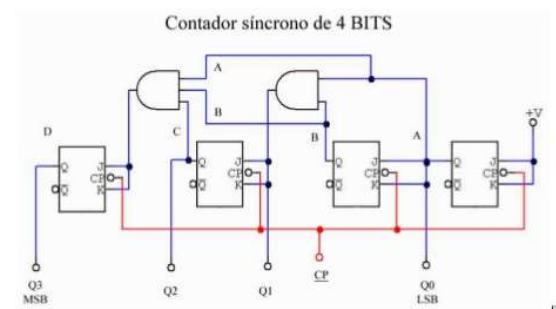
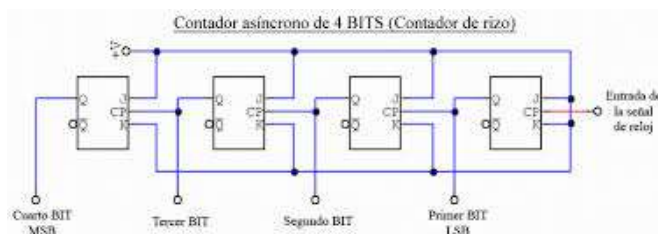
8

## Sobre contadores:

- Los contadores pueden tener uno no mas entradas de reloj
- Las entradas de reloj pueden ser **positivas o negativas**
- Hay contadores de cuenta ascendente y de cuenta descendente
- Pueden tener entradas adicionales de control (Enable, Reset, Load, etc)
- Pueden tener un puerto de entrada de datos para precargar la cuenta.
- Pueden contar de uno en uno o de manera personalizada.
- En VHDL no se especifica si va a ser asíncrono o síncrono

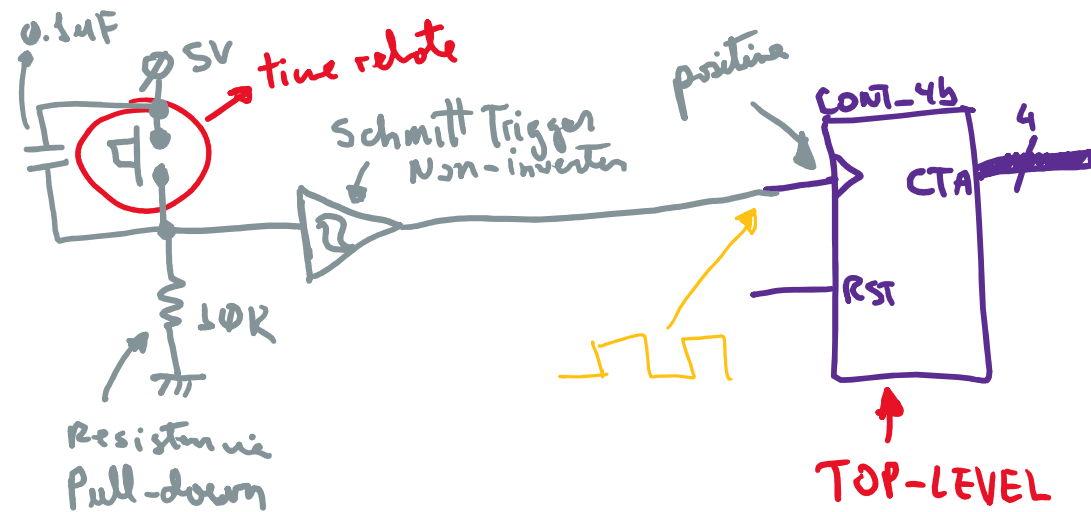
9

## Contador síncrono vs asíncrono:



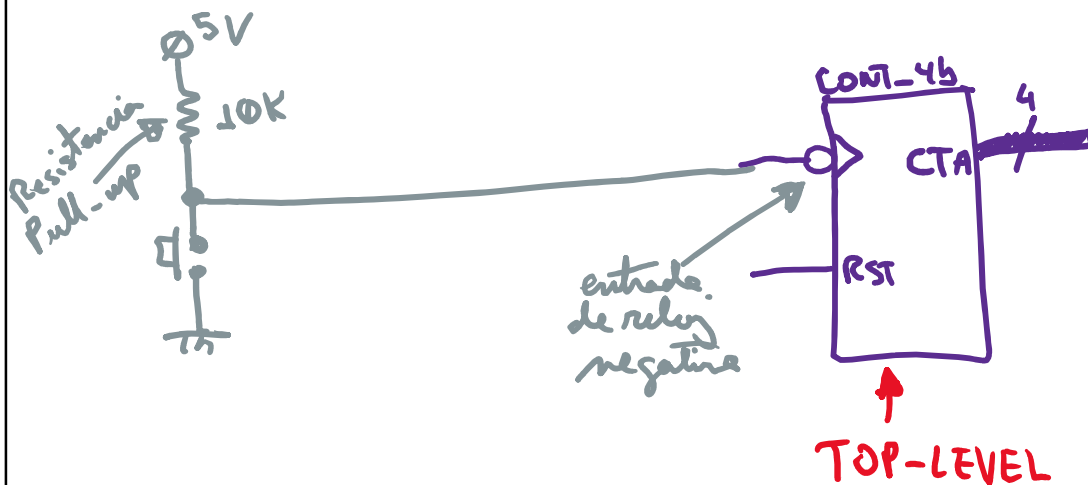
10

Modelo de contador binario básico de 4 bits ascendente con entrada de reset y reloj positivo



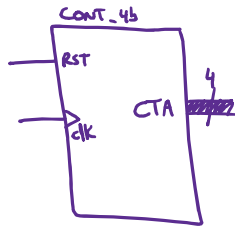
11

Modelo de contador binario básico de 4 bits ascendente con entrada de reset y reloj negativo



12

## Código en VHDL del contador básico binario:

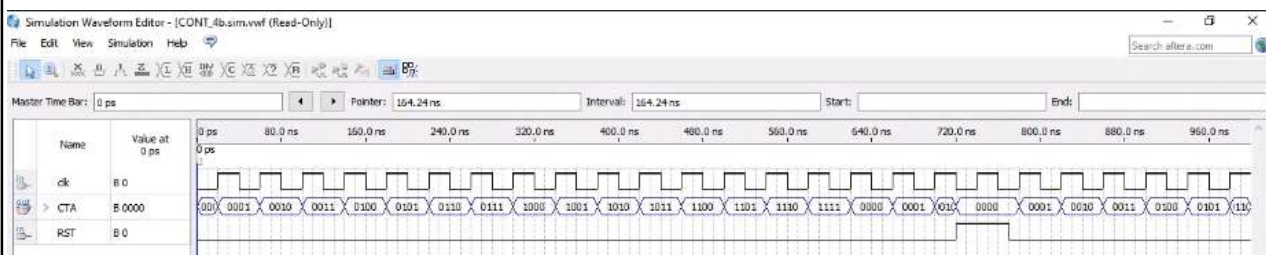


```
entity CONT_4b is
  port(clk, RST: in std_logic;
        CTA: out std_logic_vector(3 downto 0));
end CONT_4b;

architecture Oyuco of CONT_4b is
  signal cta_interno: std_logic_vector(3 downto 0);
begin
  CTA <= cta_interno;
  process(clk, RST)
  begin
    if RST = '1' then
      cta_interno <= (others => '0');
    else
      if rising_edge(clk) then
        if cta_interno = "1111" then
          cta_interno <= (others => '0');
        else
          cta_interno <= cta_interno + 1;
        end if;
      end if;
    end if;
  end process;
end Oyuco;
```

13

## Simulación en VWF:



14

## Divisor de frecuencia

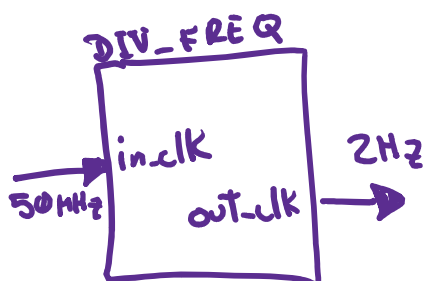
- Se emplea para disminuir la frecuencia de una señal de reloj
- En nuestro caso tenemos que usar un divisor de frecuencia para disminuir el reloj de 50MHz que hay a la entrada del FPGA de la tarjeta EP2C5T144
- Usualmente es un contador binario ( $f/2$ ,  $f/4$ ,  $f/8$ , etc)



15

## Modelo del divisor de frecuencia en VHDL

Divisor de frecuencia  
50 MHz  $\rightarrow$  2 Hz



contador de 26 bits

```

use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

entity DIV_FREQ is
  port(in_clk: in std_logic;
        out_clk: out std_logic);
end DIV_FREQ;

architecture PEPINO of DIV_FREQ is
  signal registron: std_logic_vector(25 downto 0);
  signal pre_out_clk: std_logic;
begin
  process(in_clk)
  begin
    if rising_edge(in_clk) then
      registron <= registron + 1;
      if registron = 25000000 then
        registron <= (others => '0');
        pre_out_clk <= not pre_out_clk;
      end if;
    end if;
  end process;
  out_clk <= pre_out_clk;
end PEPINO;
  
```

16



Fin de la sesión de teoría