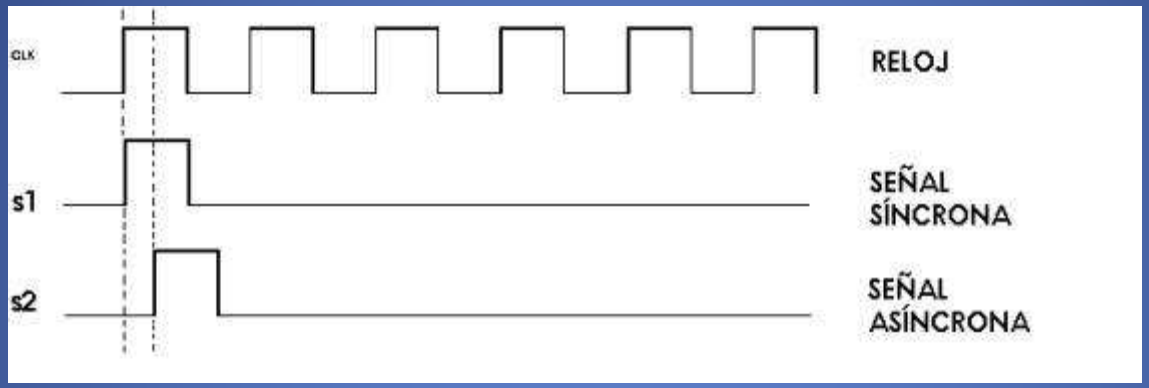
VHDL

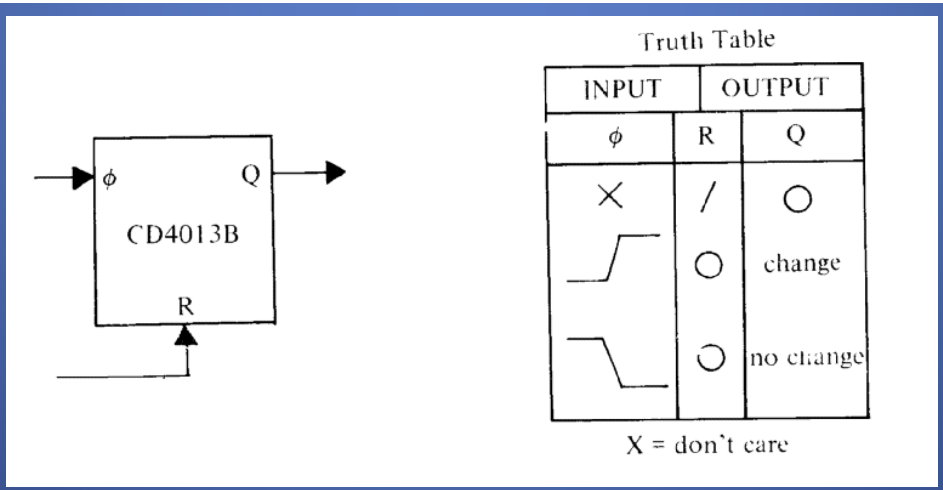
Descripción de sistemas síncronos
Process()
If..then elsif… endif
case() ..when

Descripción de un contador anillo

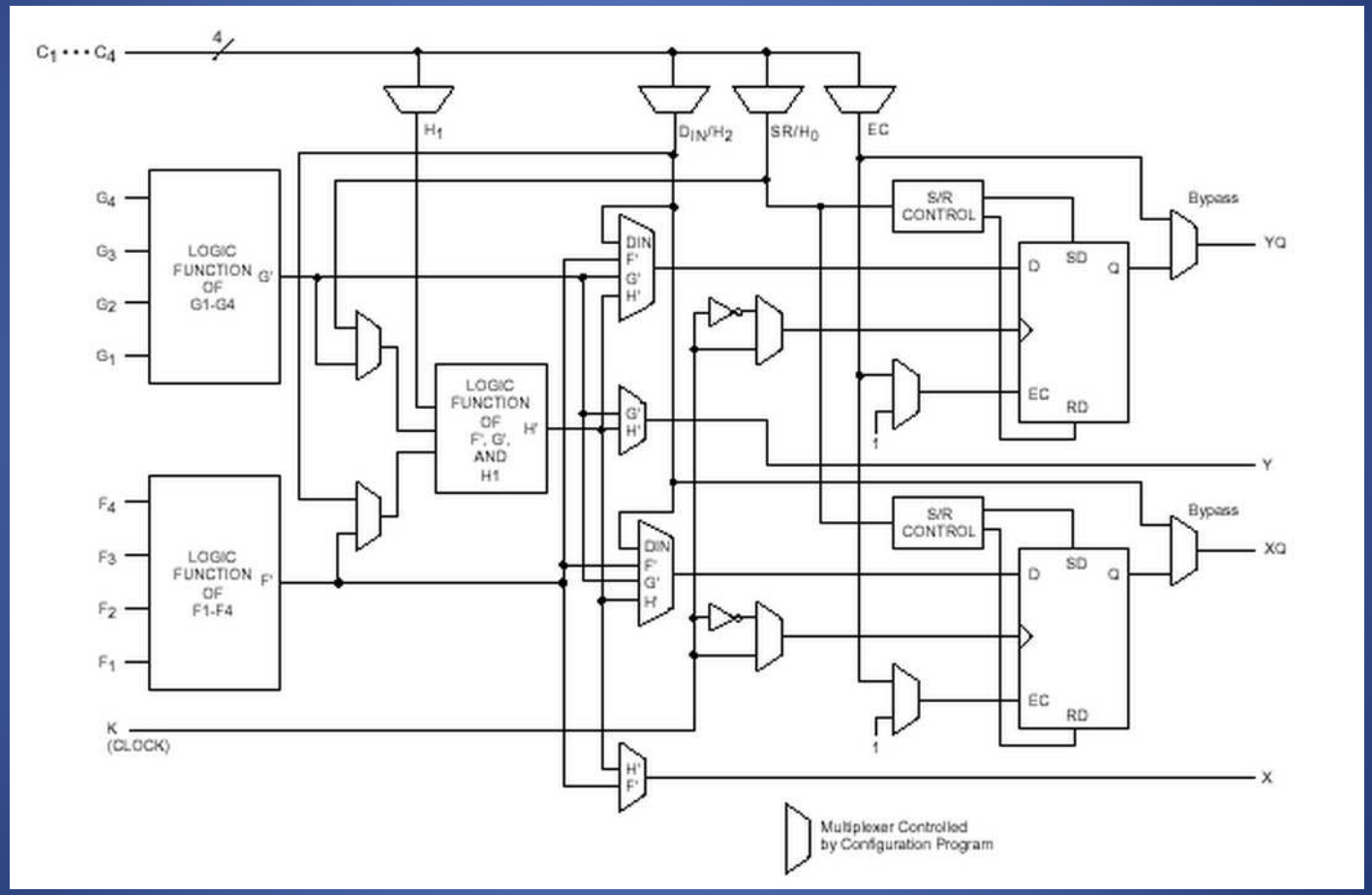
Técnicas Digitales I -Ing. Francisco G. Gutiérrez 2012

SISTEMAS SINCRONOS





MACROCELDA


C1 • • • C4
4
H1
DIN/H2
SR/H0
EC
G4
G3
G2
G1
LOGIC FUNCTION OF G1-G4
G'
LOGIC FUNCTION OF F', G', AND H1
H'
F4
F3
F2
F1
LOGIC FUNCTION OF F1-F4
F'
K (CLOCK)
DIN F' G' H'
G' H'
H' F'
S/R CONTROL
D
SD
Q
EC
RD
1
Bypass
YQ
Y
XQ
X
Multiplexer Controlled by Configuration Program

PROCESS

Plantilla de Xilinx

Lista Sensible

```
process (<all_input_signals_separated_by_commas>)
begin
  <statements>;
end process;
```

CONDICIONAL
IF..THEN

```
if <condition> then
   <statement>
elsif <condition> then
   <statement>
else
   <statement>
end if;
```

Atributos de señales

REFERENCIA A ATRIBUTOS

```
nombre_objeto'nombre_atributo
```

EJEMPLO

```
IF ( clk'event and clk = '1' ) THEN
        A <= '1' ;
END IF ;
```

'event
'active
'last_active
'last_event
'stable

PROCESS

Plantilla de Xilinx –Sistema síncrono con "reset" asíncrono.

Lista Sensible

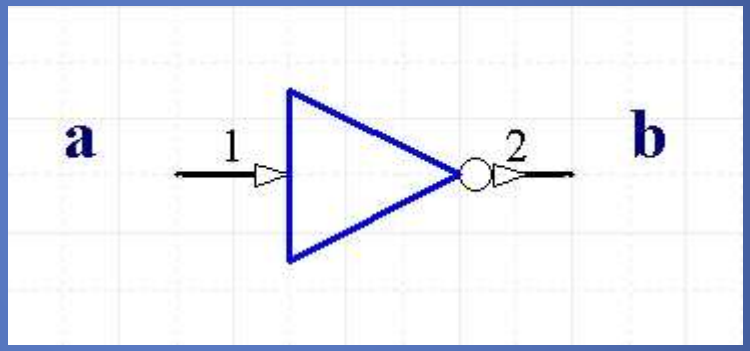
```
process (<clock>,<async_reset>)
begin
  if <async_reset> = '0' then
    <statements>;
  elsif (<clock>'event and <clock> = '1') then
    if <sync_reset> = '0' then
      <statements>;
    else
      <statements>;
    end if;
  end if;
end process;
```

Cuerpo del proceso

PROCESS

Sistema síncrono compuerta NOT.

```
process (a)
begin
  if a = '1' then
    b='0';
  else
    b='1';
  end if;
end process;
```

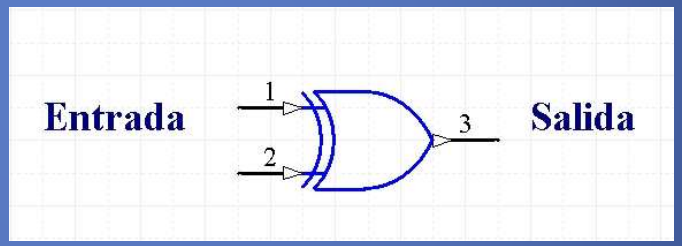


CONDICIONAL CASE()..WHEN

```
case (<2-bit select>) is
  when "00" =>
    <statement>;
  when "01" =>
    <statement>;
  when "10" =>
    <statement>;
  when "11" =>
    <statement>;
  when others =>
    <statement>;
end case;
```

CONDICIONAL CASE()..WHEN XOR

```
process (Entrada)
begin
  case (Entrada) is
    when "00" =>
      salida='0';
    when "01" =>
      salida='1';
    when "10" =>
      salida='1';
    when "11" =>
      salida='0';
    when others =>
      salida='0';
  end case;
end process;
```



Practica

1. **Codificar un Flip-Flop D con reset asincrono activo por alto y clock por flanco descendente.**
2. **Codificar un Flip-Flop D con reset síncrono en bajo y clock por flanco ascendente.**
3. **Codificar un contador anillo modulo 16 utilizando instanciación del flip-flop diseñado en 1.**

Ejemplo de Contador

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_SIGNED.ALL;

entity contador is
   Port ( Reset  : in  STD_LOGIC;
          clock : in  STD_LOGIC;
          Salida : out  STD_LOGIC_VECTOR (3 downto 0));
end contador;

architecture Behavioral of contador is
signal contador: STD_LOGIC_VECTOR (3 downto 0);
begin
process (clock, Reset)
begin
  if Reset='0' then
     contador <= (others => '0');
  elsif clock='0' and clock'event then
       contador <= contador + '1';
  end if;
end process;
Salida<=contador;
end Behavioral;
```