DESCRIPCION EN VHDL

1. DESCRIPCION EN VHDL DE UN BIESTABLE D SINCRONO POR FLANCO ASCENDENTE CON RESET SINCRONO

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity biestableD is
    Port ( reloj : in  STD_LOGIC;
           reset : in  STD_LOGIC;
               D : in  STD_LOGIC;
               Q : out  STD_LOGIC);

end biestableD;

architecture Behavioral of biestableD is

begin
process (reloj)
begin
   if reloj'event and reloj='1' then
      if reset='1' then
         Q <= '0';
      else
         Q <= D;
      end if;
   end if;
end process;
end Behavioral;
```

2. DESCRIPCION EN VHDL DE UN BIESTABLE T SINCRONO POR FLANCO ASCENDENTE CON RESET SINCRONO

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity biestableT is
```

```
    Port ( reloj : in  STD_LOGIC;
           T : in  STD_LOGIC;
           Q : out  STD_LOGIC;
                  reset: in  STD_LOGIC);
end biestableT;

architecture Behavioral of biestableT is
signal Dp, Qp : STD_LOGIC;
begin
process (reloj)
begin
  if reloj'event and reloj='1' then
    if reset='1' then
      Qp <= '0';
    else
      Qp <= Dp;
    end if;
  end if;
end process;

--- Lógica del estado siguiente

  with T select
    Dp <= Qp when '0',
          not (Qp) when others;

--- Lógica de salida
Q <= Qp ;

end Behavioral;
```

3. DESCRIPCION EN VHDL DE UN BIESTABLE JK SINCRONO POR FLANCO ASCENDENTE CON RESET SINCRONO

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

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity biestableJK is
    Port ( reloj : in  STD_LOGIC;
           reset : in  STD_LOGIC;
           J,K : in  STD_LOGIC;
           Q : out  STD_LOGIC);
end biestableJK;

architecture Behavioral of biestableJK is
signal Qp ,Dp : STD_LOGIC;
signal JK : STD_LOGIC_VECTOR (1 downto 0);
begin
```

```
process (reloj)
begin
  if reloj'event and reloj='1' then
    if reset='1' then
      Qp <= '0';
    else
      Qp <= Dp;
    end if;
  end if;
end process;
--- Lógica del estado siguiente
JK <= J&K ;
with JK select
DP <= Qp when "00",
     '0' when "01",
                '1' when "10",
    not (Qp) when others;
---- Lógica de salida
Q <= Qp ;
end Behavioral;
```

4. SUMADOR DE DOS NUMEROS DE 4 BITS C/U (VHDL)

a. MODULO SUMADOR DE 4 BITS

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

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity sumador_4bits is
    Port ( Ent1 : in  STD_LOGIC_VECTOR (3 downto 0);
           Ent2 : in  STD_LOGIC_VECTOR (3 downto 0);
           Resultado : out  STD_LOGIC_VECTOR (3 downto 0);
           Carry : out  STD_LOGIC);
end sumador_4bits;

architecture Behavioral of sumador_4bits is

signal C1,C2,C3 : std_logic;

  COMPONENT sumador_medio
  PORT (
  A : IN std_logic;
  B : IN std_logic;
  Cout : OUT std_logic;
  Suma : OUT std_logic
  ) ;
  END COMPONENT
```

```
  COMPONENT sumador_completo
  PORT (
  A : IN std_logic;
  B : IN std_logic;
  Cin : IN std_logic;
  Cout : OUT std_logic;
  Suma : OUT std_logic
  ) ;
  END COMPONENT
begin

Inst_sumador_medio : sumador_medio PORT MAP (
A => Ent1(0),
B => Ent2(0),
Cout => C1,
Suma => Resultado(0)
);

Inst_sumador_completo1 : sumador_completo PORT MAP (
A => Ent1(1),
B => Ent2(1),
Cin => C1,
Cout => C2,
Suma => Resultado(1)
);

Inst_sumador_completo2 : sumador_completo PORT MAP (
A => Ent1(2),
B => Ent2(2),
Cin => C2,
Cout =>C3,
Suma => Resultado(2)
);

Inst_sumador_completo3 : sumador_completo PORT MAP (
A => Ent1(3),
B => Ent2(3),
Cin =>C3,
Cout =>Carry,
Suma =>Resultado(3)
);

end Behavioral;
```

b. MODULO SUMADOR MEDIO

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

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
```

```
entity sumador_medio is
   Port ( A : in  STD_LOGIC;
          B : in  STD_LOGIC;
          Suma : out  STD_LOGIC;
          Cout : out  STD_LOGIC);
end sumador_medio;

architecture Behavioral of sumador_medio is

begin
SUMA <= (B xor A) ;
COUT <= (B and A) ;

end Behavioral;
```

c. MODULO SUMADOR TOTAL (1 Instancia)

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

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity sumador_completo is
   Port ( A : in  STD_LOGIC;
          B : in  STD_LOGIC;
          Cin : in  STD_LOGIC;
          Cout : out  STD_LOGIC;
          Suma : out  STD_LOGIC);
end sumador_completo;

architecture Behavioral of sumador_completo is

begin
SUMA <= (B xor A) xor CIN ;
COUT <= (B and A) or (( B xor A ) and CIN ) ;

end Behavioral;
```

o o o O o o o