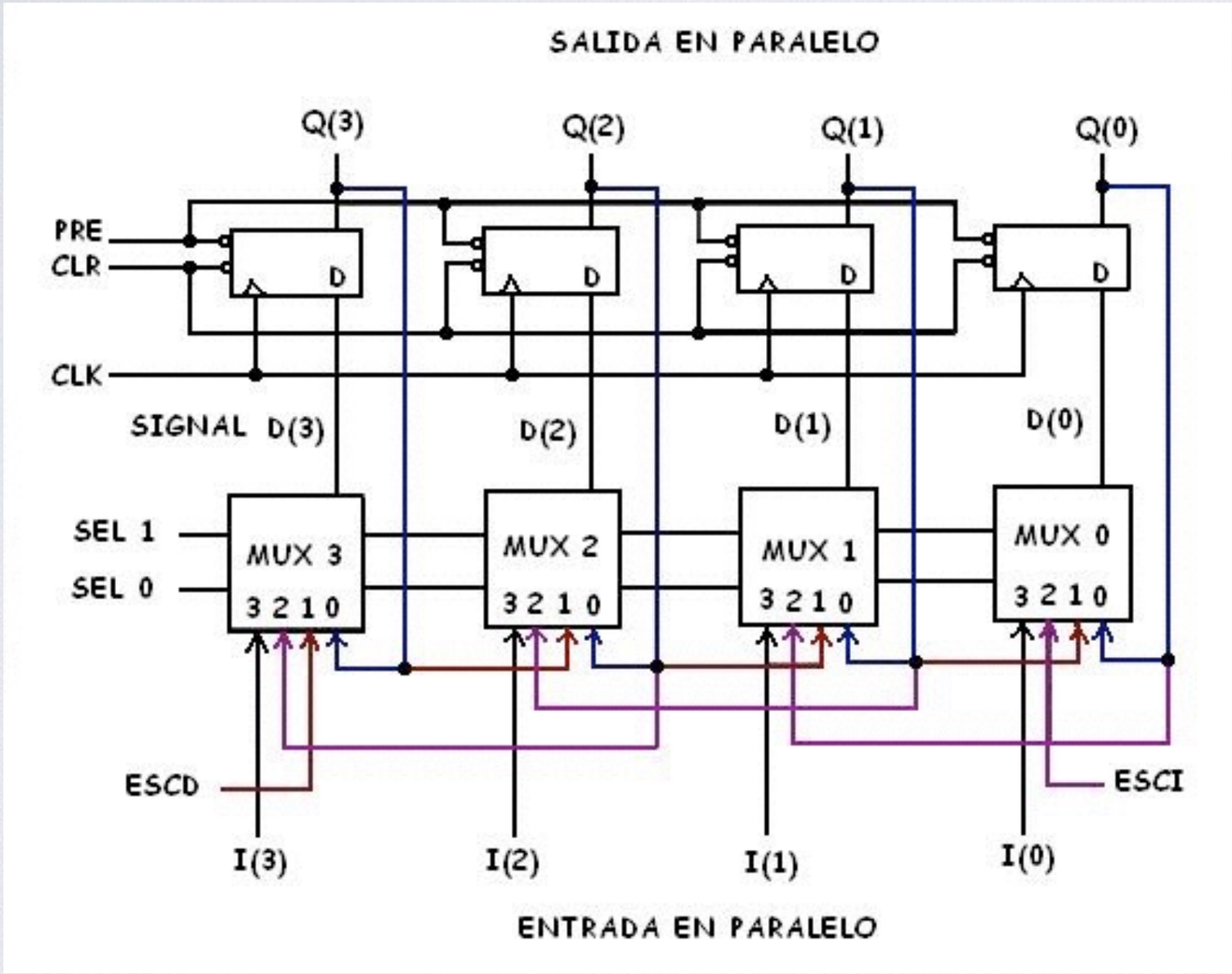
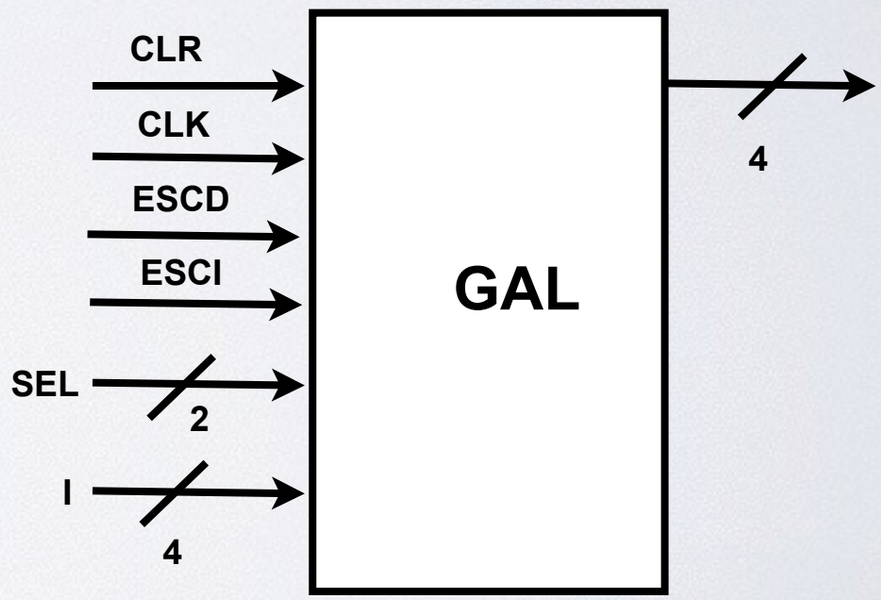
REGISTRO UNIVERSAL

DISEÑO DE SISTEMAS DIGITALES
OPTATIVA I. ISISA
AUTOR: Claudia A. López R.

ARQUITECTURA



SEL 1	SEL 0	ACCION
0	0	RETENCION
0	1	ESCD
1	0	ESCI
1	1	CARGA DE DATO



DESCRIPCION EN VDHL

SEL 1	SEL 0	ACCION
0	0	RETENCION
0	1	ESCD
1	0	ESCI
1	1	CARGA DE DATO

VERSION I

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.AL

ENTITY   REG   IS
      PORT( CLR, CLK, ESCD, ESCDI: IN STD_LOGIC;
                SEL: IN STD_LOGIC_VECTOR (1 DOWNTO 0);
                 I: IN STD_LOGIC_VECTOR (3 DOWNTO 0);
                Q: INOUT  STD_LOGIC_VECTOR (3 DOWNTO 0)
              );
END  REG;

ARCHITECTURE AREG OF REG IS
SIGNAL D: STD_LOGIC_VECTOR (3  DOWNTO 0):
BEGIN
PROCESS (CLK,CLR)
BEGIN
      IF (CLR='0') THEN
          Q<="0000";
      ELSIF (CLK ' EVENT  AND CLK='1') THEN
           Q<= D;
     END  IF;
END PROCESS;
```

```
MUX3: PROCESS (SEL)
BEGIN
      CASE SEL IS
            WHEN "00" => D(3) <= Q(3);
            WHEN "01" => D(3) <= ESCD;
            WHEN "10" => D(3) <= Q(2);
            WHEN OTHERS => D(3) <= I(3);
      END CASE;
END PROCESS MUX 3;

MUX2: PROCESS (SEL)
BEGIN
      CASE SEL IS
            WHEN "00" => D(2) <= Q(2);
            WHEN "01" => D(2) <= Q(3);
            WHEN "10" => D(2) <= Q(1);
            WHEN OTHERS => D(2) <= I(2);
      END CASE;
END PROCESS MUX 2;

MUX1: PROCESS (SEL)
BEGIN
      CASE SEL IS
            WHEN "00" => D(1) <= Q(1);
            WHEN "01" => D(1) <= Q(2);
            WHEN "10" => D(1) <= Q(0);
            WHEN OTHERS => D(1) <= I(1);
      END CASE;
END PROCESS MUX 1;

MUX0: PROCESS (SEL)
BEGIN
      CASE SEL IS
            WHEN "00" => D(0) <= Q(0);
            WHEN "01" => D(0) <= Q(1);
            WHEN "10" => D(0) <= ESCI;
            WHEN OTHERS => D(0) <= I(0);
      END CASE;
END PROCESS MUX 0;

END AREG;
```


AUTOR: CLAUDIA A. LOPEZ RODRÍGUEZ

SENTENCIA FOR

La sentencia o declaración FOR-LOOP o FOR-GENERETE es usada siempre que una operación necesita ser repetida n veces.

FOR LOOP → **DENTRO DE PROCESOS**

Sintaxis

FOR VAR IN RANGO LOOP

---------- CODIGO

END LOOP;

Ejemplos:

FOR I IN 0 TO 3 LOOP
---------- CODIGO
END LOOP;

FOR I IN 3 DOWNTO 0 LOOP
---------- CODIGO
END LOOP;

FOR GENERATE → **S E N T E N C I A S CONCURRENTES**

Sintaxis

Etiqueta: FOR VAR IN RANGO GENERATE

---------- CODIGO

END GENERATE;

Ejemplos:

CICLO I: FOR I IN 0 TO 3 GENERATE
---------- CODIGO
END GENERATE;

CICLO II: FOR I IN 3 DOWNTO 0 GENERATE
---------- CODIGO
END GENERATE;

DESCRIPCION EN VDHL

SEL 1	SEL 0	ACCION
0	0	RETENCION
0	1	ESCD
1	0	ESCI
1	1	CARGA DE DATO

VERSION 2

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

Entity Registro is
Port(
    D : in std_logic_vector (7 downto 0);
    SEL : in std_logic_vector (1 downto 0);
    ESCI, ESCD, CLK, CLR : in std_logic;
    Q : INOUT std_logic_vector (7 downto 0)
        );
end Registro;
```

```
Architecture  Arq_Registro of Registro is
Begin

PREG : Process ( CLK, CLR )
Begin

    if( CLR = '1' ) then
        Q <= X"00";
          elsif (CLK'event and CLK ='1') then
             for I in 0 to 7 loop
                    case SEL is
                           when "00" => Q(I) <= Q(I);
                           when "01" => Q(I) <= D(I);
                           when "10" =>
                                  if (I = 0) then
                                  Q(I) <= ESCI;
                                  else
                                  Q(I) <= Q(I-1);
                                  end if;
                    when others =>
                                  if  (I = 7 )  then
                                  Q(I) <= ESCD;
                                  else
                                  Q(I) <= Q(I+1);
                                  end if;
                    end case;
             end loop;
          end if;
end process PREG;

end Arq_Registro;
```

OPERADORES DE CORRIENTO

- **SLL.** Desplazamiento lógico hacia la izquierda, llenado con ceros.
- **SRL**. Desplazamiento lógico hacia la derecha, llenado con ceros.
- **SLA**. Desplazamiento aritmético hacia la izquierda, llenado con el bit de menor peso.
- **SRA**. Desplazamiento aritmético hacia la derecha, llenado con el bit de mayor peso.
- **ROL**. Rotación a la izquierda.
- **ROR**. Rotación a la derecha.

NOTA: LOS OPERADORES DE CORRIMIENTO SOLO PUEDEN SER USADOS CON TIPO DE DATROS BIT O BIT_VECTOR.

AUTOR: CLAUDIA A. LOPEZ RODRÍGUEZ

DESCRIPCION EN VDHL

SEL 1	SEL 0	ACCION
0	0	RETENCION
0	1	ESCD
1	0	ESCI
1	1	CARGA DE DATO

VERSION 3

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

Entity Registro is
Port(
        D : in std_logic_vector (7 downto 0);
        SEL : in std_logic_vector (1 downto 0);
        ESCI, ESCD, CLK, CLR : in std_logic;
        Q : INOUT std_logic_vector (7 downto 0)
       );
end Registro;
```

```
Architecture Arq_Registro of Registro is
Begin

PREG : Process( CLK, CLR )
Begin

    if ( CLR = '1' ) then
          Q <= X"00";
    elsif (CLK'event and CLK ='1') then

       for I in 0 to 7 loop

             case SEL  is
              when "00" => Q <= Q;
              when "01" => Q <= D;
              when "10" => Q <=TO_STDLOGICVECTOR(TO_BITVECTOR(Q) SLL 1);
                           Q(0) <= ESCI;
              when others => Q <= TO_STDLOGICVECTOR(TO_BITVECTOR(Q) SRL 1);
                           Q(7) <= ESCD;
             end case;
       end loop;
      end if;
end process PREG;

end Arq_Registro;
```


AUTOR: CLAUDIA A. LOPEZ RODRÍGUEZ

DESCRIPCION EN VDHL

SE PUEDEN TENER MAS VERSIONES DE ESTA ARQUITECTURA

AUTOR: CLAUDIA A. LOPEZ RODRÍGUEZ