



INSTITUTO POLITÉCNICO
NACIONAL
ESCUELA SUPERIOR DE
COMPUTO



DISEÑO DE SISTEMAS DIGITALES

PRÁCTICA 4. CONTADOR CON FSM MOORE.

PROFESOR:

Testa Nava Alexis

Alumno:

Ramírez Cotonieto Luis Fernando

GRUPO:

2CV18

FECHA: 07/Mayo/2021

PRÁCTICA 4. CONTADOR CON FSM

MOORE.

Los contadores pueden tener varias señales de control dependiendo de las operaciones que realiza. Las más comunes son:

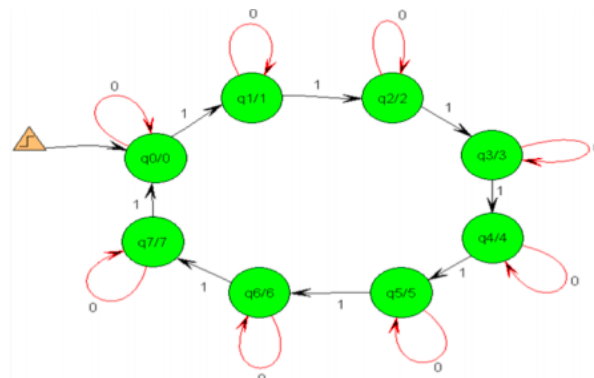
Señal de carga (L - load). Cuando esta activa, esta señal permite la carga del dato colocado en el bus de datos de entrada del contador en el bus de salida Q.

Señal de habilitación (E - Enable). Cuando esta activa, esta señal permite la operación del contador, en caso contrario el contador permanece sin cambio.

Señal de dirección de conteo (UD – Up/Down). Cuando esta activa, esta señal permite el conteo ascendente del contador, en caso contrario el conteo es descendente.

Señal de acarreo (C – Carry). Esta es una señal de salida. Se activa cuando el contador pasa del número mayor que se puede representar con los bits del contador, al menor en un conteo ascendente. También, cuando el contador pasa del número menor que se puede representar con los bits del contador al mayor en un conteo descendente. Por ejemplo, en el caso de un contador de 3 bits, el acarreo se activa cuando el contador pasa del 7 al 0 o cuando pasa del 0 al 7.

Aplicar la metodología para diseño de circuitos secuenciales y obtener las ecuaciones de entrada de excitación usando FF-D, FF-JK y FF-T que permitan implementar el diseño de un Contador ascendente/descendente con señal de control, tomando en cuenta el autómata mostrado en la ilustración.



Una vez realizado el diseño funcional y simulado en FALSTAD, realizar en HDL el diseño de un contador genérico como se muestra a continuación.

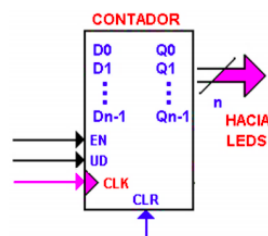


Tabla de operación del contador

EN	UD	OPERACIÓN
0	X	RETENCIÓN
1	1	CONTEO ASCENDENTE
1	0	CONTEO DESCENDENTE

Tabla de estados

X	Q2	Q1	Q0	D2	J1	K1	T0	Q2N	Q1N	Q0N
0	0	0	0	0	0	X	0	0	0	0
0	0	0	1	0	0	X	0	0	0	1
0	0	1	0	0	X	0	0	0	1	0
0	0	1	1	0	X	0	0	0	1	1
0	1	0	0	1	0	X	0	1	0	0
0	1	0	1	1	0	X	0	1	0	1
0	1	1	0	1	X	0	0	1	1	0
0	1	1	1	1	X	0	0	1	1	1
1	0	0	0	0	0	X	1	0	0	1
1	0	0	1	0	1	X	1	0	1	0
1	0	1	0	0	X	0	1	0	1	1
1	0	1	1	1	X	1	1	1	0	0
1	1	0	0	1	0	X	1	1	0	1
1	1	0	1	1	1	X	1	1	1	0
1	1	1	0	1	X	0	1	1	1	1
1	1	1	1	0	X	1	1	0	0	0

Ecuaciones

D2

Q1,Q0

	00	01	11	10
X,Q2\				
00	0	0	0	0
01	1	1	1	1
11	1	1	0	1
10	0	0	1	0

$$X' Q2 + Q2 Q1' + Q2 Q0' + X Q2' Q1 Q0$$

J1

Q1,Q0

	00	01	11	10
X,Q2\				
00	0	0	X	X
01	0	0	X	X
11	0	1	X	X
10	0	1	X	X

X Q0

K1

Q1,Q0					
		00	01	11	10
X,Q2\	00	0	0	X	X
	01	0	0	X	X
	11	0	1	X	X
	10	0	1	X	X

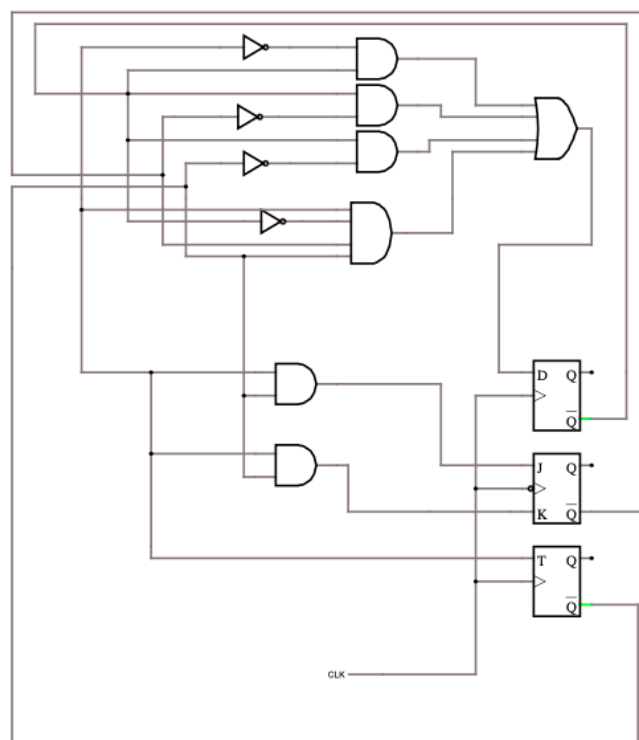
X Q0

Q1,Q0					
		00	01	11	10
X,Q2\	00	0	0	0	0
	01	0	0	0	0
	11	1	1	1	1
	10	1	1	1	1

X

DIAGRAMA (FALSTAD)

<https://tinyurl.com/yzy7mj72>



t = 0 s
intervalo tiempo = 5 μs

VHDL

```

library ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
USE ieee.std_logic_unsigned.all;

entity Prac4 is port(
    clr,clk,E,L,UD: in std_logic;
    D: in std_logic_vector(4 downto 0);
    Q: inout std_logic_vector(4 downto 0)
);
attribute pin_numbers of Prac4: entity is
    "clr:2 E:3 L:4 UD:5 D(4):6 D(3):7 D(2):8 D(1):9 D(0):10 Q(4):18 Q(3):17 Q(2):16 Q(1):15 Q(0):14 ";
end Prac4;

architecture A_Prac4 of Prac4 is
begin
    process(clk,clr)
        variable aux: std_ulogic := '1';
    begin
        if(clr='1') then
            Q<=(others=>'0');
        elsif(clk'event and clk='1') then
            if(E='1' and L='1') then          --CARGA
                Q<=D;
            elsif(E='0') then                --RETENCIÒN
                Q<=Q;
            elsif(E='1' and L='0' and UD='1') then    --ASCENDENTE
                Q<=Q+1;
            elsif(E='1' and L='0' and UD='0') then    --DESCENDENTE
                Q<=Q-1;
            end if;
        end if;
    end process;
end A_Prac4;

```