

Práctica N° 6. Contadores

1. Datos de la práctica

Carrera	INGENIERÍA ELECTRÓNICA		
Semestre		Grupo	
Tipo de Práctica	<input type="checkbox"/> Laboratorio <input type="checkbox"/> Simulación	Fecha	
Asignatura	Electrónica Digital I		
Unidad Temática			
N° Alumnos por práctica	2	N° Alumnos por reporte	2
Nombre del Profesor			
Nombre(s) de Alumno(s)	1. 2.		
Tiempo estimado		Vo. Bo. Profesor	
Comentarios			

2. Objetivos

- Describir la diferencia entre los sistemas síncronos y asíncronos.
- Comprender el funcionamiento de los flip flops a través de los diagramas de tiempo de sus salidas.
- Utilizar los diagramas de transición de estados para describir la operación de un contador.

3. Medios a utilizar

Por cada práctica y por cada puesto de laboratorio, los materiales a utilizar son:

Cantidad	Descripción
1	Computadora
1	Tarjeta de desarrollo Basys2 Digilent
1	Software Xilinx ISE® Webpackv14.7

4. Actividades previas

Diseñar un contador ascendente de 0 al 7 que deba incluir:

- Diagrama de estado
- Tablas de estado del sistema
- Diagrama del circuito del contador con flip flop tipo D

5. Introducción

Un contador digital corresponde a una máquina secuencial que recibe pulsos de una señal de reloj, cuenta los flancos de bajada (o de subida) de la misma señal y entrega dicha cuenta, en binario, por las salidas. La cantidad máxima que se alcance a contar dependerá del número de bit de salida que se dispongan. Los contadores de década, del tipo BCD, se utilizan ampliamente en aplicaciones donde los pulsos o sucesos van a ser contados y los resultados exhibidos en algún tipo de dispositivo de visualización numérica decimal.

En este laboratorio se realizará el código para modelar dos tipos de Flip Flops (Tipo D y JK). Una vez comprendido su funcionamiento, se hará un contador digital binario que cuente desde 0000 a 1111(0 – F). Utilizaremos el módulo x7seg de la práctica de laboratorio 4 para visualizar estos dígitos en un display de la tarjeta Basys2 y el módulo Bin2BCD para apreciarlo en formato decimal. El diseño incluye una señal de reloj y una señal de reset (clr) que se utiliza para poner a cero dichas cuentas.

6. Desarrollo de la práctica

Flip Flop Tipo D

1. Crear un nuevo proyecto en ISE® llamado **FlipFlops**.
2. Crear el módulo **DFF** para el Flip Flop tipo D descrito en el siguiente programa.

```
entity DFF is
    Port ( clk : in  STD_LOGIC;
          clr : in  STD_LOGIC;
          set : in  STD_LOGIC;
          D  : in  STD_LOGIC;
          Q  : out STD_LOGIC;
          notQ : out STD_LOGIC);
end DFF;

architecture Behavioral of DFF is
begin
    process(clk,clr,set)
    begin
        if(clr = '1') then
            Q <= '0';
            notQ <= '1';
        elsif (set = '1') then
            Q <= '1';
            notQ <= '0';
        elsif (rising_edge(clk)) then
            Q <= D;
            notQ <= not D;
        end if;
    end process;
end Behavioral;
```

Flip Flop Tipo JK

3. Crear el módulo **JKFF** para el tipo de Flip Flop JK descrito en el siguiente programa.

```
entity JKFF is
    Port ( J : in  STD_LOGIC;
          K : in  STD_LOGIC;
          set : in  STD_LOGIC;
          clr : in  STD_LOGIC;
          clk : in  STD_LOGIC;
          Q : inout STD_LOGIC;
          notQ : inout STD_LOGIC);
end JKFF;

architecture Behavioral of JKFF is
begin
    process(clk,clr,set)
    begin
        if(clr = '1') then
            Q <= '0';
            notQ <= '1';
        elsif (set = '1') then
            Q <= '1';
            notQ <= '0';
        elsif (rising_edge(clk)) then
            if(J='0' and K='0')then
                Q <= Q;
                notQ <= not Q;
            elsif(J='0' and K='1')then
                Q <= '0';
                notQ <= '1';
            elsif(J='1' and K='0')then
                Q <= '1';
                notQ <= '0';
            else
                Q <= not Q;
                notQ <= Q;
            end if;
        end if;
    end process;
end Behavioral;
```

Simulación

1. Verificamos la sintaxis, agregamos una nueva fuente “New Source” y escogemos el tipo de archivo “VHDL Test Bench”. Se tendrá que crear una simulación para cada archivo de Flip Flop que se ha realizado.
2. Recordemos pasarnos al modo Simulación y en el archivo VHDL Test Bench **no comentar todas las líneas que hacen referencia a clock**.
3. Agregue los siguientes estímulos para cada código correspondiente:

Flip Flop D	Flip Flop JK
<code>clr <= '1';</code> <code>wait for 100 ns;</code>	<code>clr <= '1';</code> <code>wait for 100 ns;</code>
<code>clr <= '0';</code> <code>set <= '1';</code> <code>wait for 100 ns;</code>	<code>clr <= '0';</code> <code>set <= '1';</code> <code>wait for 100 ns;</code>
<code>clr <= '0';</code> <code>set <= '0';</code> <code>D <= '0';</code> <code>wait for 100 ns;</code>	<code>clr <= '0';</code> <code>set <= '0';</code> <code>J <= '0';</code> <code>K <= '0';</code> <code>wait for 100 ns;</code>
<code>D <= '1';</code> <code>wait for 100 ns;</code>	<code>J <= '0';</code> <code>K <= '1';</code> <code>wait for 100 ns;</code>
	<code>J <= '1';</code> <code>K <= '0';</code> <code>wait for 100 ns;</code>
	<code>J <= '1';</code> <code>K <= '1';</code> <code>wait for 100 ns;</code>

4. Una vez agregado el estímulo deberíamos obtener las formas de onda para cada flip flop. Enseñe su simulación al profesor.

7. Actividades propuestas

Práctica: Diseño del Contador

Una vez realizado el análisis de los dos tipos de flip flop más utilizados, se reutilizará el módulo del Flip Flop tipo D para poder diseñar el contador de 0 a 15.

1. Crear un nuevo proyecto en ISE® llamado **Contador**.
2. Agregar el módulo del flip flop D que se ha elaborado previamente.

3. Instanciar el modulo las veces que sea necesario para diseñar el contador final según el diagrama de la Figura 1 y las ecuaciones de la Tabla.

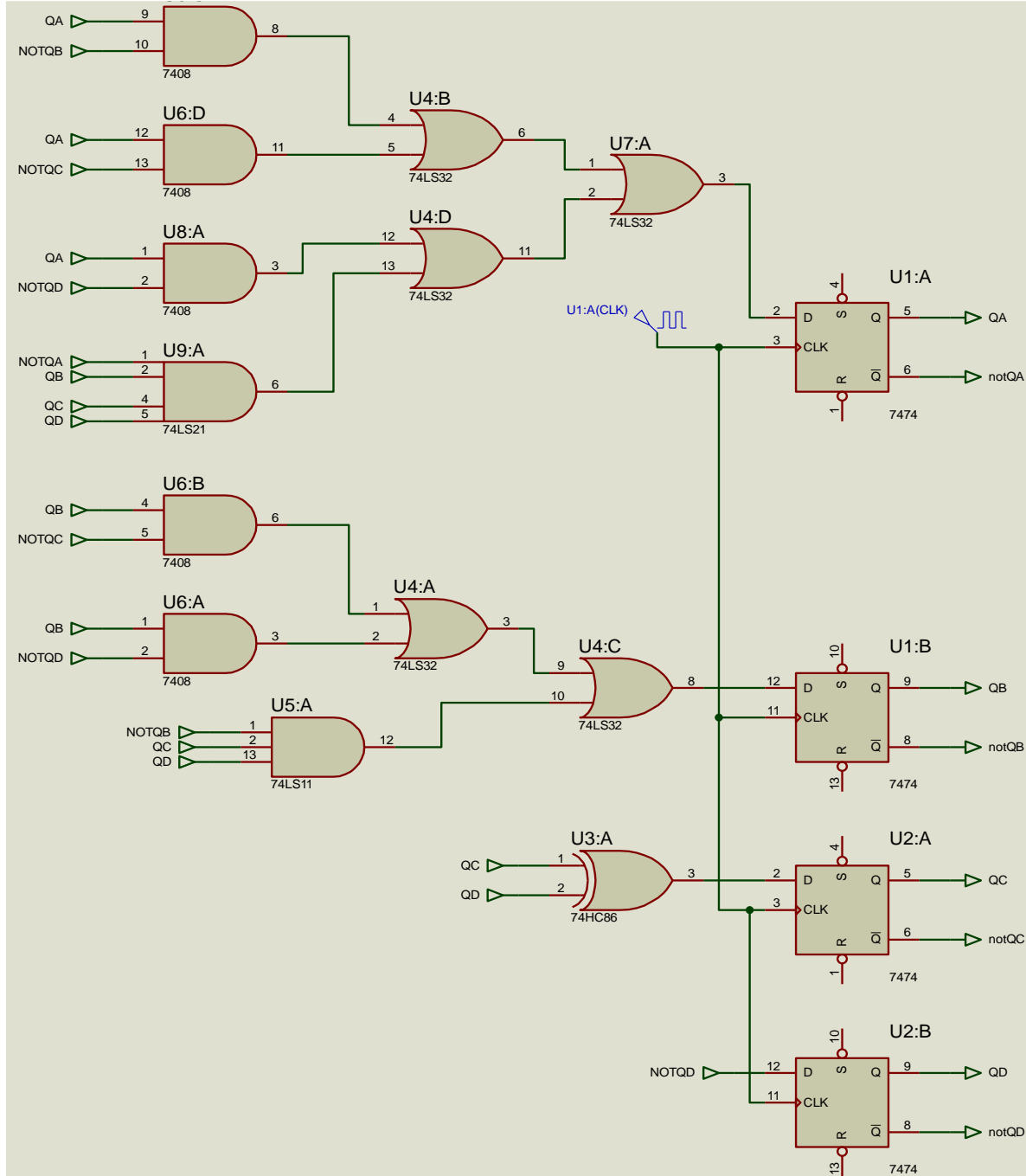


Figura 1. Circuito Contador de 0 a 15

$$DA = (QA \text{ and } \text{not}QB) \text{ or } (QA \text{ and } \text{not}QB) \text{ or } (QA \text{ and } \text{not}QD) \text{ or } (\text{not}QA \text{ and } QB \text{ and } QC \text{ and } QD)$$

$$DB = (QB \text{ and } \text{not}QC) \text{ or } (QB \text{ and } \text{not}QD) \text{ or } (\text{not}QB \text{ and } QC \text{ and } QD)$$

$$DC = QC \text{ xor } QD$$

$$DD = \text{not}QD$$

4. Agregar el Modulo **Bin2BCD** y el **X7Seg** para que el conteo binario realizado por el contador se visualice en los displays de siete segmentos a como se muestra en la Figura 2.

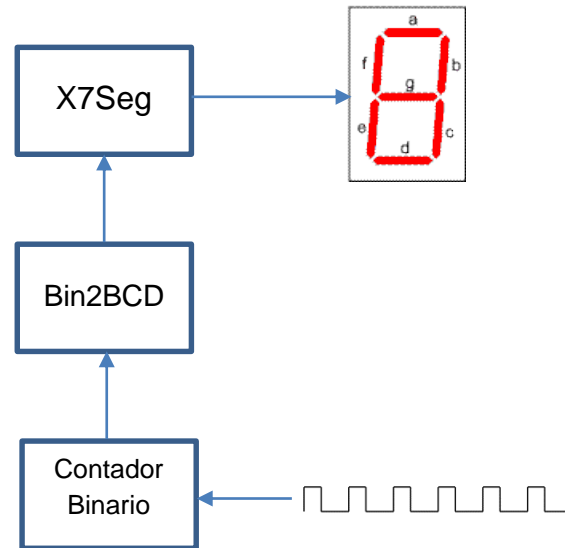


Figura 2. Estructura de un contador BCD con visualización en un display

5. Realizar la simulación del Proyecto y mostrársela al profesor.
6. Implementar su diseño en la tarjeta Basys2.

8. Trabajo a Entregar (Reporte)

Realice el contador asignado en clases en un nuevo proyecto VHDL de tal forma que su conteo se visualice en el display de siete segmentos. Recuerde incluir en el reporte el diagrama de estados, la tabla de estados, mapa de Karnaugh y la carpeta del proyecto generada por el Xilinx.