



66.17 - SISTEMAS DIGITALES

Trabajo Práctico 1 - Contador BCD de 4 dígitos con salida a display 7 segmentos

Andrew Parlane

15 de abril de 2018

Índice

1. Introducción	3
2. Herramientas	3
3. Implementación	3
3.1. Contador	3
3.1.1. Señales y Parámetros	4
3.1.2. Diagrama de Bloques	4
3.2. sevenSegmentDisplay	4
3.2.1. Señales y Parámetros	5
3.2.2. Diagrama de Bloques	5
3.3. tp1	5
3.3.1. Señales y Parámetros	5
3.3.2. Diagrama de Bloques	6
4. Simulación y Verificación	7
5. Síntesis	7
5.1. Resumen de síntesis	7

Índice de figuras

1. Diagrama de Bloques de componente contador.	4
2. Diagrama de Bloques de componente sevenSegmentDisplay.	5
3. Diagrama de Bloques de componente tp1.	6
4. El contador solo cuenta cuando es activa.	7
5. Las cuatro cifras contando hasta 9999.	7

1. Introducción

El objetivo de este trabajo es a diseñar, simular y implementar en FPGA un circuito digital que cuenta en código binario decimal (BCD para binary coded decimal) desde cero hasta 9999 y muestra el valor corriente en cuatro displays de siete segmentos.

Diseñé el circuito para funcionar con una Altera DE2 placa de desarrollo que tiene un cyclone II 2C35 FPGA. Para el propósito de este trabajo la única diferencia notable entre la DE2 y la Spartan-3 Starter Board es que la DE2 tiene siete cátodos por cada display en vez de siete compartido entre los cuatro displays.

2. Herramientas

Las herramientas que usé para este trabajo son:

- Questa Sim v10.2c
- Quartus II v13.0sp1
- GNU Make v4.2.1

3. Implementación

Los componentes principales de mi diseño son:

- Un contador que cuenta desde cero hasta un parámetro genérico MAX.
- Un convertidor que produce un señal para mostrar los números cero a nueve en un display de siete segmentos.
- Un componente que junta todas las partes.

Los recursos de la placa DE2 que uso son:

- Dos botones con supresión de rebotes.
- Un reloj de 50MHz.
- Ocho displays de siete segmentos.

He usado un botón para un señal reset y el otro para acelerar la salida a 1KHz en vez de 1Hz. Solo uso cuatro de los displays de siete segmentos. He asignado a los otros cuatro solo para apagarlos.

3.1. Contador

Este componente es un contador genérico que incrementa su valor una vez cada tick del reloj, si el enable está alto. Tiene un rango de cero hasta MAX y vuelve a cero después de max. Hay una salida que indica cuando el valor está igual a MAX.

3.1.1. Señales y Parámetros

Nombre	Tipo	Bits	Descripción
Parámetros			
WIDTH	natural		Cuántos bits hay en la cuenta.
MAX	natural		El valor máximo.
Entradas			
clk	std_logic	1	El reloj.
rst	std_logic	1	Reset asíncrono.
en	std_logic	1	Enable.
load	std_logic	1	Iniciar la cuenta. No es usado en ese proyecto.
loadData	unsigned	WIDTH	El valor asignado a la cuenta si load es alto.
Salidas			
count	unsigned	WIDTH	El valor corriente del contador.
atMax	std_logic	1	Se indica cuando count es igual a MAX.

3.1.2. Diagrama de Bloques

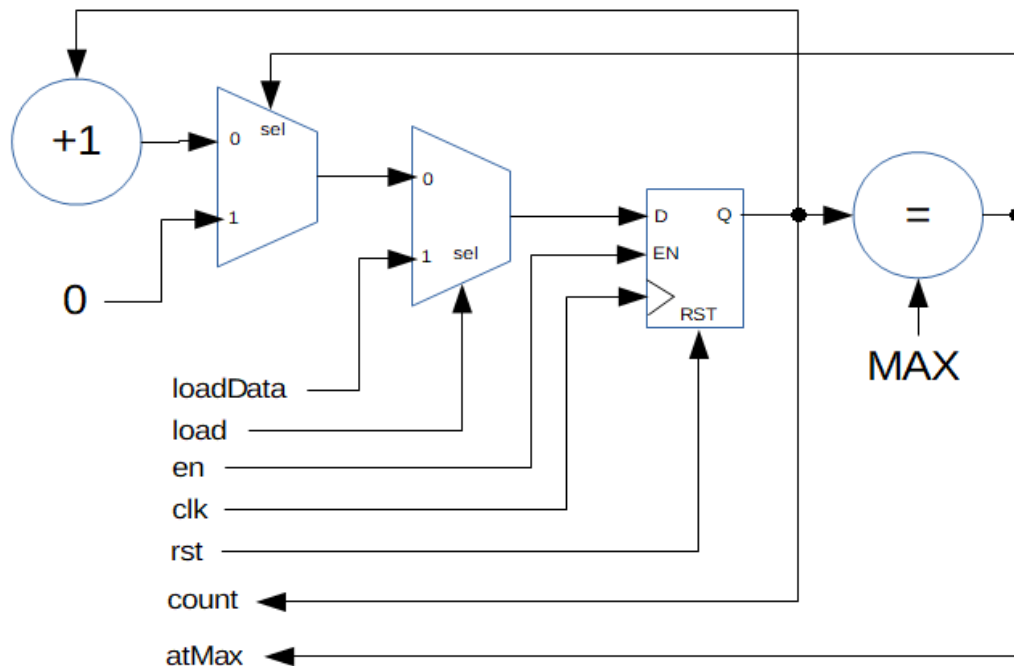


Figura 1: Diagrama de Bloques de componente contador.

3.2. sevenSegmentDisplay

Este componente es un convertidor entre un cuatro bit BCD y los siete bits necesarios a mostrar la cifra en un display de siete segmentos.

3.2.1. Señales y Parámetros

Nombre	Tipo	Bits	Descripción
Entradas			
bcd	unsigned	4	Un numero entre 0 y 9.
Salidas			
sevenSegmentOutput	unsigned	7	El patrón para mostrar el número BCD en el display de siete segmentos.

3.2.2. Diagrama de Bloques

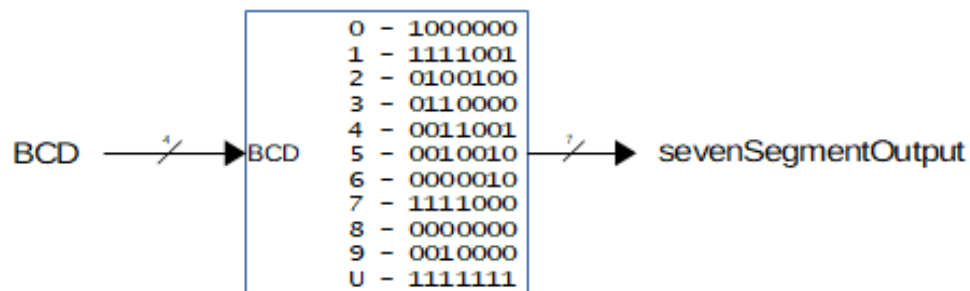


Figura 2: Diagrama de Bloques de componente sevenSegmentDisplay.

3.3. tp1

Este componente es el módulo de nivel más alto. Hay dos contadores que son generadores de enables, uno que genera un enable a 1Hz y el otro a 1KHz. Uso el de 1KHz para tener un modo rápido para mejor probar el diseño. Elijo cual a usar con un botón en la placa DE2.

Después hay cuatro contadores más que cuentan entre cero y nueve. Cada uno corresponde a una cifra BCD. El enable del primero viene del generador de enable elegido por el botón. Los otros están activos solo cuando el contador previo está activo y su valor es el máximo (9).

Cada uno de estos contadores conecta a la entrada de un sevenSegmentDisplay y la salida va al display de siete segmentos.

3.3.1. Señales y Parámetros

Nombre	Tipo	Bits	Descripción
Parámetros			
CLOCK_DIVIDER	natural		Esto es usado por el banco de pruebas para dejarme simular el diseño más rápidamente.
Entradas			
KEY	std_logic_vector	2	Dos botones, uso uno para el reset, y el otro para activar modo rápido. Los dos están activa baja.
Salidas			
HEX0 - HEX7	std_logic_vector	7	Displays de siete segmentos.

3.3.2. Diagrama de Bloques

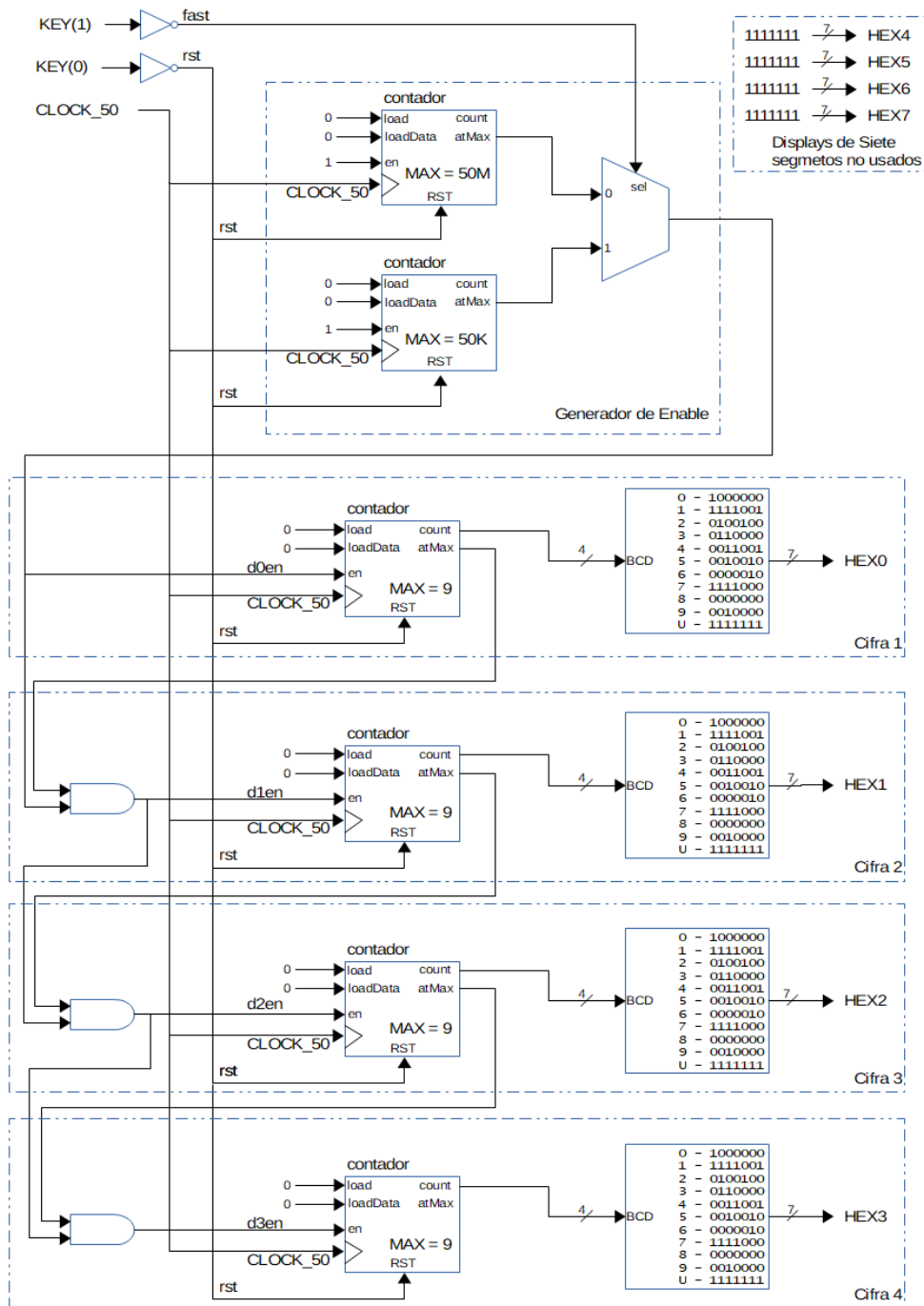


Figura 3: Diagrama de Bloques de componente tp1.

4. Simulación y Verificación

Tengo bancos de pruebas para el contador y el tp1 que me permiten verificarlos. He usado asserts en PSL para especificar los requisitos y después estímulo las entradas para comprobar que todas las pruebas aprueban.

En el banco de prueba de tp1 en vez de solo comprobar las salidas del componente uso nombres jerárquicos para dejarme leer las salidas de los contadores que están señales internos de tp1.

Durante la simulación Questa Sim produce un mensaje para cada error y al final genera una tabla de resumen indicando cuantas fallas hubieron por cada assert.

También comprobé algunas partes interesantes de las olas generadas por Questa Sim.

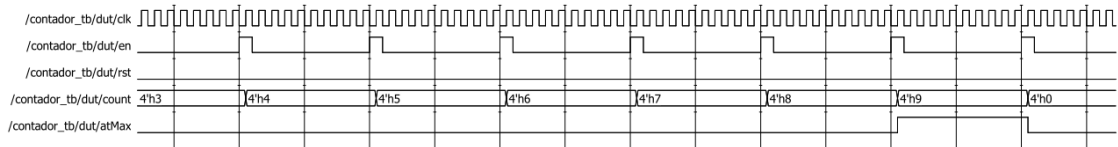


Figura 4: El contador solo cuenta cuando es activa.

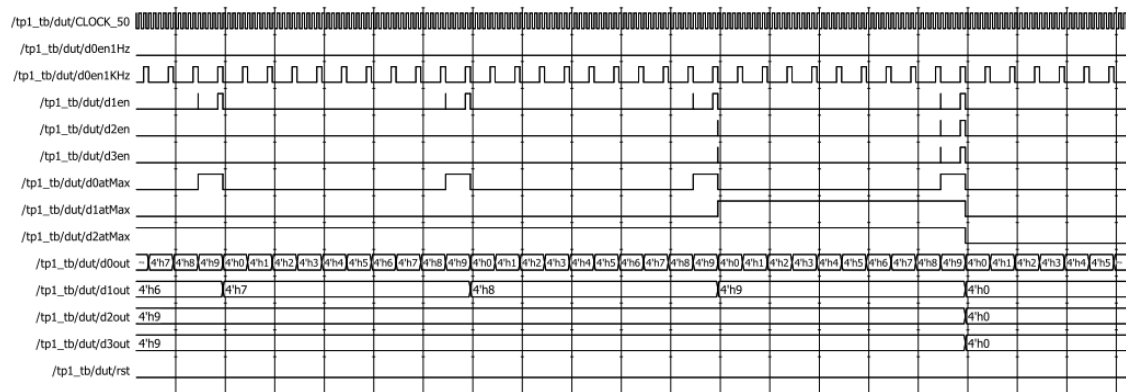


Figura 5: Las cuatro cifras contando hasta 9999.

5. Síntesis

He usado Quartus II para sintetizar el diseño.

5.1. Resumen de síntesis

Ítem	Utilizado	Disponible	Porcentaje Utilizado
Logic Elements:	126	33216	<1 %
Registers:	56	33216	<1 %

TimeQuest Timing Analyzer me informe que el frecuencia máxima de reloj a la que está operable el circuito es 145.54MHz.