

Maria Alejandra Cabezas - Daniel Pinilla Paris

BIBLIOTECA DE COMPONENTES

PUBLISHED BY UNIVERSIDAD DE LOS ANDES - COLOMBIA



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

September 2018



Contenido

1	Componente secuencial generador de números pseudoaleatorios	5
1.1	Componente secuencial generador de números pseudosecuenciales	5
1.1.1	Descripción del componente	5
1.1.2	Diagrama	5
1.1.3	Resultados y lecciones aprendidas	6

Componente secuencial generador de números pseudosecuenciales

Descripción del componente

Diagrama

Resultados y lecciones aprendidas

1. Componente secuencial generador de números pseudoaleatorios

1.1 Componente secuencial generador de números pseudosecuenciales

A continuación se presenta el componente secuencial **generador de números pseudoaleatorios** mostrando su descripción ...

1.1.1 Descripción del componente

El elemento diseñado es un componente secuencial. Este componente es un registro de propósito general que tiene implementado un generador de números pseudoaleatorios de 8 bits. El componente recibe 1 byte de entrada de forma paralela proveniente de un bloque que genera números pseudoaleatorios. La salida del componente se da paralelamente cada vez que se carga un dato en el registro. Este componente mantiene las señales del registro general Reset, Clear, Load y Clk.

1.1.2 Diagrama

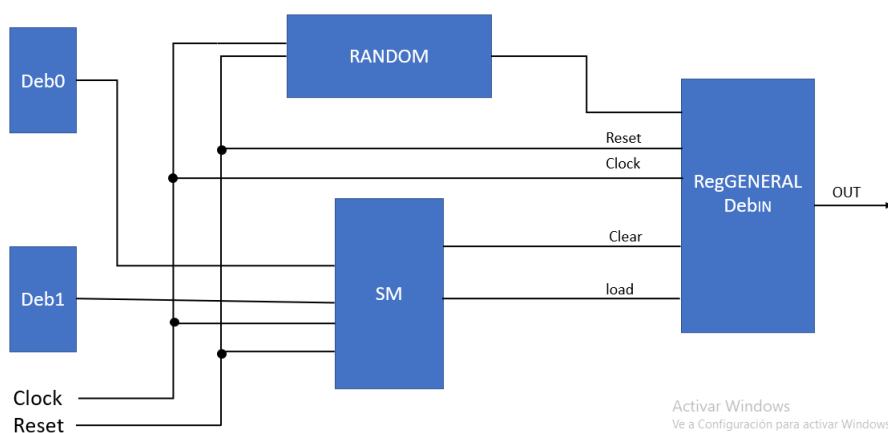


Figure 1.1: Diagrama del componente secuencial



1.1.3 Resultados y lecciones aprendidas

A través de Quartus se logró implementar este componente de secuencia. Para poder hacer esto se creó un componente llamado RANDOM que es el componente que genera números aleatorios. Una vez se creó este elemento se implementó junto a los otros componentes (Deb0, Deb1, SM, RegGENERAL). De esta forma, se creó el registro el cual genera números pseudoaleatorios de 8 bites.

Se eligió el resultado presentado puesto que, tras la comparación con los demás resultados evaluados, se logra una mayor cantidad de combinaciones de números pseudo-aleatorios mediante la implementación del componente desarrollado. De esta forma, se define que para la implementación del componenete realizado, contando con la mayor cantidad de núemros generados, se realiza una operación **XOR** entre los bits 2 y 4 de la salida del subcomponenete *Reg_Shifter*, que permitió una salida de 9 números pseudoaleatorios.