



Tarea 1

22 de marzo de 2020

Agustín Campeny

1. Introducción

En este reporte se presentan los resultados de los módulos *RTL* implementados en el lenguaje *verilog*, presentándose una breve introducción sobre la implementación, y las formas de onda resultantes de la simulación de los testbench respectivos.

Se utiliza el software *Icarus Verilog* para la compilación y simulación de los testbench, y *GTKWave* para la visualización de las formas de onda resultantes.

2. Módulos RTL

2.1. Left Barrel Shifter

Este módulo realiza la operación shift left de manera completamente combinacional. La implementación se encuentra basada en un documento por Matthew Rudolf Pillmeier¹. Se implementa un módulo de largo variable definido por el parámetro *N*. A continuación se presentan formas de onda para varias entradas.

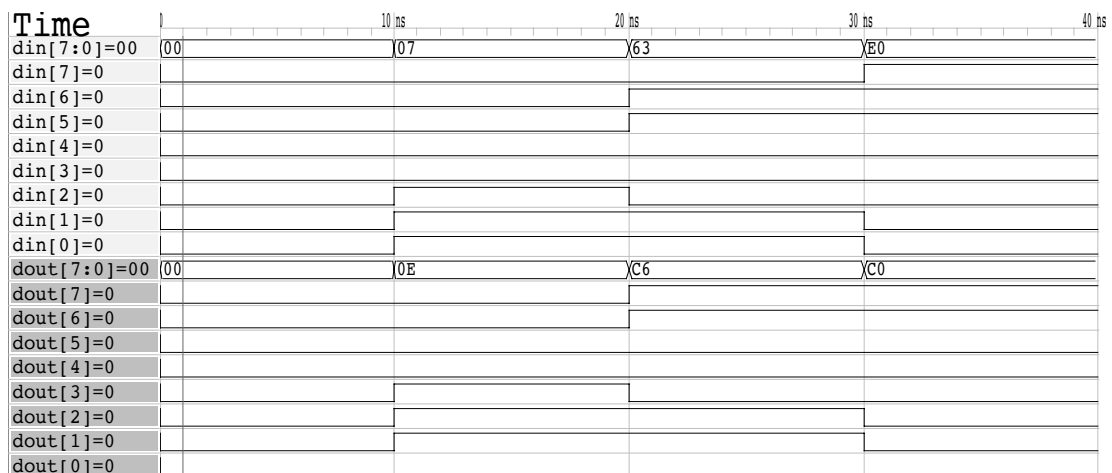


Figura 1: Entrada y salida para diversos valores de largo 8 bits

¹Barrel shifter design, optimization, and analysis, 2001

2.2. Addressable shift register

Este módulo consiste en un shift register de 8 registros de largo variable. Se realiza un shift de los contenidos de cada registro en cada ciclo de reloj, y la salida corresponde a los contenidos del registro señalado por la dirección en la entrada **addr**. Este módulo fue basado en la descripción provista por Xilinx.²

La implementación se realizó mediante un arreglo de 8 registros de largo definido por el parámetro N. A continuación se presentan las formas de onda para distintas entradas y valores de dirección de salida.

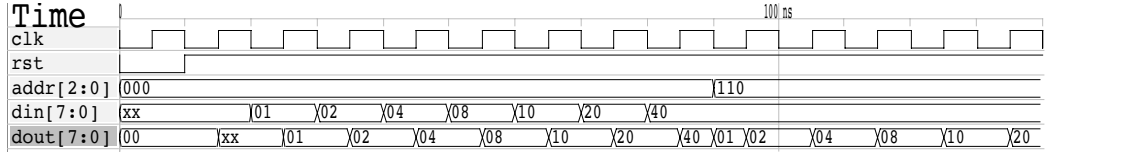


Figura 2: Valores de salida correspondientes a la dirección en **addr**

2.3. Linear-feedback shift register

Se implementa un LFSR de tipo *Fibonacci*, en el que el bit realimentado corresponde al XOR entre los bits de varias posiciones (denominadas “taps”) dentro del registro.

La elección de los taps en este módulo fueron tal que el polinomio de realimentación fuese primitivo, con tal de asegurar un periodo máximo de $2^n - 1$, con n el largo en bits.³ Se utilizó una tabla de polinomios publicada por E.J. Watson⁴, y se seleccionó el siguiente polinomio:

$$x^{16} + x^5 + x^3 + x^2 + 1 \quad (2.1)$$

Al ser un LFSR de largo 16 bits, el periodo máximo es de 65535 valores diferentes antes de que se repita la secuencia. A continuación se presentan las formas de onda que prueban que esto se cumple.

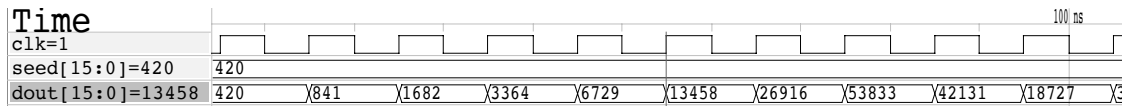


Figura 3: Seed inicial y primeros valores

²<https://www.manualslib.com/manual/195956/Xilinx-System-Generator-V2-1.html?page=26>

³https://en.wikipedia.org/wiki/Linear-feedback_shift_register

⁴Primitive Polynomials (mod 2)

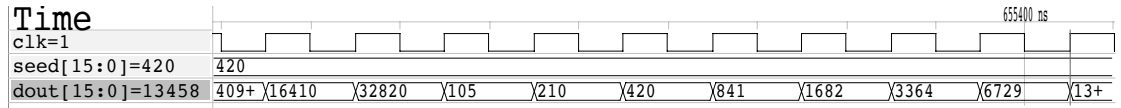


Figura 4: Reinicio del ciclo luego de 65535 ciclos

2.4. Multiplier accumulator

Este módulo multiplica los valores en sus entradas, y guarda el valor de la operación en un acumulador, el que corresponde a la salida de este⁵. En los ciclos siguientes el valor del acumulador corresponde al valor anterior sumado al resultado de la multiplicación en dicho ciclo. Cada 8 ciclos se reinicia el valor del acumulador.

A continuación se presentan las formas de onda en donde se aprecia el valor del acumulador, y su reinicio en el ciclo 8.



Figura 5: Salida del módulo con respecto a las entradas y el número de ciclos

⁵https://en.wikipedia.org/wiki/Multiply-accumulate_operation