

Grandes mentiras de las FPGAs: divisores de frecuencia como relojes

Creador: David Rubio G.

Entrada: <https://soceame.wordpress.com/2024/12/17/grandes-mentiras-de-las-fpgas-divisores-de-frecuencia-como-relojes/>

Blog: <https://soceame.wordpress.com/>

GitHub: <https://github.com/DRubioG>

Fecha última modificación: 23/02/2025

En esta entrada hablaré de una mentira que no todo el mundo conoce y es fundamental para el desarrollo en FPGAs.

Las FPGAs conviven diferentes sistemas internos, unos referidos a la electrónica digital en sí, otros a los relojes, otros a los sistemas de comunicaciones de tipo transceiver, etc. Y es importante tener en cuenta que no se deben juntar los sistemas si no se conoce cómo hacerlo convenientemente.

Una cosa que se suele manejar en una FPGA son sistemas con diferentes frecuencias, y lo que se suele hacer es construir sistemas que se dedican a dividir la frecuencia de reloj principal para conseguir estos nuevos «relojes» con la frecuencia deseada. Y cómo se ha construido un nuevo «reloj», éste puede ser utilizado de forma indiscriminada como si de un reloj se tratase.

Bien, pues **esto es mentira**. Es más, es una mentira muy gorda, porque realmente no has construido un reloj, has construido *una señal de habilitación*. Esta mentira se perpetua por el uso del nombre «divisor de frecuencia», que alude a la creación de una nueva frecuencia mediante la división de una frecuencia anterior, creada por un reloj real.

¿Qué es un reloj y qué lo hace tan especial en una FPGA?

Lo primero hay que explicar que una señal de reloj es una señal con parámetros de funcionamiento, con el jitter o el slew-rate, controlados y constantes. De tal forma que la propia FPGA tiende a separar los relojes del resto de electrónica, para que vaya por pistas propias y no por pistas de la electrónica digital (estas pistas suelen estar en un plano dimensional distinto, porque tienen que actuar sobre toda la lógica digital).

Estas pistas por las que va el reloj son pistas controladas, porque son pistas que están hechas para que junto con un PLL puedan autocompensar los retardos por los incrementos de temperatura. Pues estos retardos que se compensan en las pistas de reloj, no se suelen compensar en la electrónica digital, haciendo que el incremento de temperatura afecte a los retardos de las señales.

Dejando más o menos claro lo que es un reloj y el funcionamiento de sus pistas propias. Al construir un divisor de frecuencia lo que se está haciendo es tomar un reloj «puro» (con su compensación, etc) y pasárselo a una lógica digital, para construir un nuevo «reloj», que es dependiente de los retardos que se produzcan por temperatura en la FPGA. Entonces, esta nueva señal no es un reloj, porque ha dejado de serlo.

Entonces, ¿qué es?

La respuesta es simple, es **una señal de habilitación**, que lo que hace es actuar dejando actuar a otra lógica que sí que depende de un reloj real, y esta señal solo dice cuándo funciona esta lógica.

Vale, y si quiero un divisor de frecuencia que genere un reloj real, ¿cómo lo hago?

Utilizando un **PLL**, que se alimentará del reloj principal y externo que tiene la FPGA (*se recomienda bufferear el reloj antes de meterlo en la FPGA, sobretodo si el sintetizador no lo hace*), y generará un nuevo reloj, que sí que tendrá todos los sistemas de un reloj real, y además, este nuevo reloj viajará por las pistas propias del reloj y no por la electrónica.

Y si quiero sacar una frecuencia nueva al mundo exterior, ¿cómo lo hago?

Tanto si esta frecuencia la has generado con un PLL o con un divisor de frecuencia, siempre debes tratar de bufferear internamente este reloj a la salida para que el reloj que salga al exterior sea (o se aproxime lo máximo a) un reloj real.

NOTA FINAL

Cuando generéis un divisor de frecuencia acordaros de meter antes de que salga la señal un biestable D, que garantice que esa señal se sincronice con el reloj que la ha generado. Esto es debido a que el retardo que sufre una FPGA al contar hasta 50 no es el mismo que el que tiene al contar hasta 100.000.000, por lo que para garantizar que ese reloj se ciñe lo máximo al reloj que lo ha creado, y con los retardos constantes, se tiene que resincronizar al final.

Recomendación

Muchas veces podréis ver como en algún FW alguien ha añadido una señal de *Clock Enable* con una AND con el reloj del sistema, para habilitar el reloj, o ha multiplexado dos relojes con un multiplexor de la lógica de la FPGA.

Esto os recomiendo que nunca lo hagáis, porque estáis convirtiendo un reloj «puro» es una señal de habilitación que va por la lógica programable. Lo que se tiene es que hacer uso de las macros de reloj que incorpora el fabricante que estéis manejando, porque esas macros harán que vuestro reloj siga siendo un reloj.

Consejo: Nunca juntéis un reloj con algo que no es de la familia de los relojes, porque si no éste dejará de ser un reloj.