

## **Trabajo Integrador Final**

Realice un divisor de números enteros de cuatro bits mediante el uso de una **FSM** (Finite State Machine), compuesta por:

- Módulo de procesamiento de datos que permita al usuario cargar los valores del numerador, denominador y obtener el resultado.
- Bloques combinacionales y/o secuenciales que procesan la información.
- FSM que realice la operación de división entera.

Para dividir dos números enteros de N bits, se propone el siguiente algoritmo:

1. Cargar los datos, en este caso numerador y denominador de la división.
2. Determinar si el denominador es distinto de cero. Si el denominador fuera igual a cero, la división no sería posible. En este caso el proceso termina, pudiendo informar esta situación mediante algún mensaje o señal de error. De lo contrario, es posible realizar la división.
3. Para obtener el resultado y el resto de la división, es posible restar al numerador una cantidad igual al denominador, iterando sucesivamente hasta que el numerador sea menor al denominador. Cuando esto sucede, el resultado es igual al número de iteraciones realizadas, y el resto de la división es igual al valor del numerador tras la última iteración. Por ejemplo, para hacer 5 dividido 2 el proceso es el siguiente, dando como resultado 2 y resto 1:

|                    |             |
|--------------------|-------------|
| <b>Iteración 1</b> | $5 - 2 = 3$ |
| <b>Iteración 2</b> | $3 - 2 = 1$ |

Realice la FSM junto con su **diagrama de estados y transiciones**. Este diagrama será el que describe el comportamiento de la FSM. Con el diagrama de estados y transiciones, y las operaciones a realizar, el paso final es determinar qué elementos de hardware (registros, flip-flops, contadores, memorias, etc.) se requieren para llevar a cabo el algoritmo propuesto. Un diagrama en bloques puede resultar muy útil en esta instancia, ya que ayuda a visualizar el hardware que se va a implementar, con sus respectivas entradas, salidas y señales de control.

Implemente el diseño en verilog, compruebe su correcto funcionamiento mediante testbench y finalmente impacte el diseño en la FPGA provista por la cátedra. Recuerde que las FPGA disponen de 4 LEDs que permitirán visualizar el numerador y el denominador en el momento de carga, así como el resultado. Asimismo, posee cuatro botones para que el usuario determine la carga de los operandos, el valor numérico de los mismos y la muestra de resultados.

Las pautas para la realización del TPI son las siguientes:

- El **TPI** se podrá realizar en **grupo** de hasta **dos** alumnos/as
- Deberán realizar un **informe** bien completo en donde expliquen cómo abordaron el problema a resolver. Pueden agregar los **recursos** que consideren necesarios (imágenes, diagramas, test bench, etc.) siempre que les sean de **utilidad** para explicar la solución a la que abordaron. Pueden incluir algo de código si consideran que es necesario para explicar alguna parte del informe. No incluyan TODO el código.

- Todo **recurso** que incluyan (imágenes, diagramas, test bench, código, etc.) **debe estar explicado** en alguna parte del informe.
- Desde la cátedra, les sugerimos que utilicen el **template** para **informes** que se encuentra en la sección **Materiales del curso -> Material para informes**. Allí podrán descargar el *template* para Word y LaTeX..
- Se recomienda limitar la **cantidad de páginas** a 12.
- El informe es **grupal**.
- Para **entregar** todo el material, deberán subir **un único archivo en formato .zip** (o .rar) con el nombre **ApellidoAlumno1\_ApellidoAlumno2.zip** que deberá incluir:
  1. El **informe** en formato en **pdf**
  2. El **código fuente** de la implementación del problema resuelto (solo los archivos **.v** de los módulos en verilog y de los módulos de test-bench)

Subir **un archivo** .zip (.rar) por **grupo**.

Fecha de entrega: 13/06/2023

Defensa: 20/06/2023 en horario de clase