

Tarea #7

(Entrega 23 de junio)

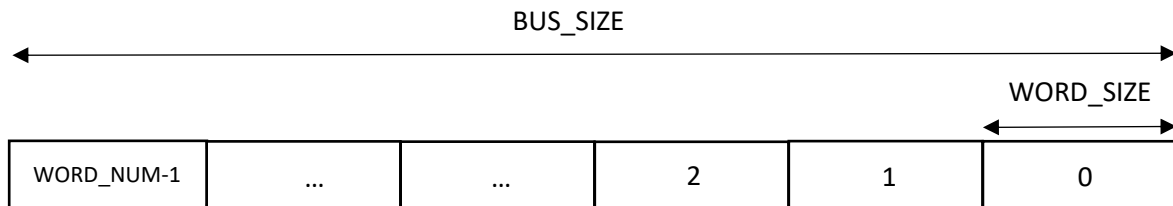
Diseño de una máquina de estados y un multiplexor parametrizado.

*****OJO***** Al igual que en la **Tarea #1** tome el tiempo que demora en hacer cada una de las cosas solicitadas: búsqueda de información, diseño, elaboración de las pruebas, ejecución de las simulaciones, etc.

Evaluación

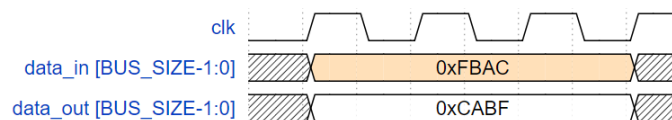
1. Funcionamiento del diseño:
 - a. Descripción conductual 35%
 - b. Descripción estructural 35%
 - c. Pruebas y verificador 15%
 - d. Makefile 5%
2. Documentación 10%

Su bloque cuenta con una entrada de reset, un reloj y un bus de datos de tamaño `BUS_SIZE`, `BUS_SIZE` divisible por `WORD_SIZE`. Tiene una salida de datos de tamaño `BUS_SIZE`, una salida de control de tamaño `WORD_NUM = BUS_SIZE / WORD_SIZE`, y una salida de error.



La salida de control de tamaño `WORD_NUM` tendrá un bit mapeado a cada palabra del bus. El valor de cada bit será una bitwise OR de los `WORD_SIZE` bits de cada palabra del bus. Para construir la lógica de esta salida, utilice un `genvar`, un `generate` y un `for loop`. Todos los tamaños son parametrizados.

La salida de los datos invierte el orden de las palabras de tamaño `WORD_SIZE` en el bus de datos, utilizando `genvar`, `generate` y un `for loop` (el mismo del párrafo anterior). Por ejemplo, si se tuviera `BUS_SIZE=16` y `WORD_SIZE=4`:



La máquina de estados se encargará de informar si hay algún error en los paquetes de entrada, considerando el siguiente protocolo de paquetes de datos. Todos los paquetes deben tener 1's en la palabra más significativa. Todos los paquetes deben seguir una secuencia consecutiva que inicia en cero en la palabra menos significativa. Por ejemplo, manteniendo BUS_SIZE=16 y WORD_SIZE=4 (los asteriscos (*) pueden ser cualquier valor de datos):

