

Laboratorio 6

Objetivo

- Utilización de las instrucciones secuenciales, concurrentes y paquetes aprendidos en clase.
- Interpretación de información especificada en hojas de datos o especificaciones de diseño.
- Familiarizarse con el DE2-115 Cyclone IV development kit board. Uso de un constraint file para asignación de pines de I/O, pulsadores, switches y LEDs.
- Uso de la herramienta MegaWizard para la implementación de un DLL o PLL para generar un reloj de baja frecuencia (opcional).
- Utilización de restricciones de tiempo adecuadas a una interface tipo “source synchronous”.
- Descripción en VHDL de un controlador de memoria SRAM (memoria externa al FPGA).
- Simulación a nivel de compuerta (gate level) o post-place and route.
- Configuración del FPGA.

Descripción

Realizar la descripción en VHDL de un sistema digital que controle la lectura y escritura de datos en una memoria tipo SRAM desde un FPGA.

El kit de desarrollo DE2-115 tiene una memoria SRAM (IS61WV102416BLL) disponible para la escritura y lectura de datos controlada por el FPGA. Las conexiones entre el FPGA y la SRAM son detalladas en la siguiente figura.

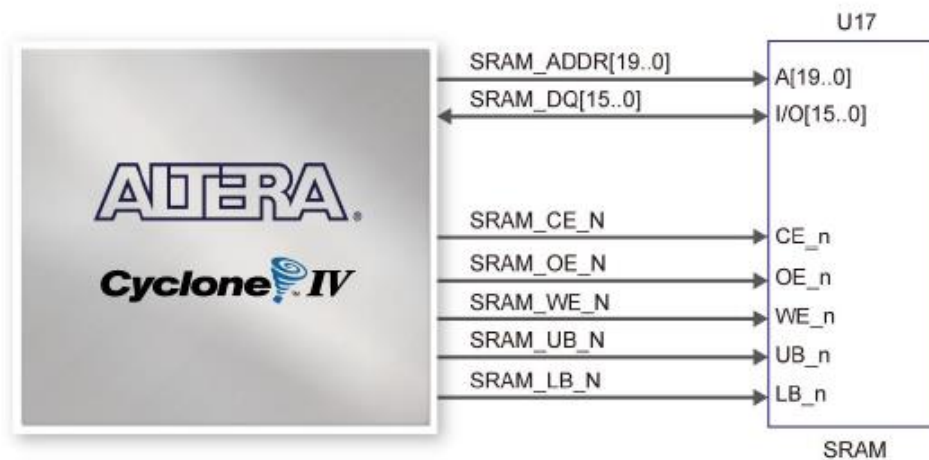


Figura 1 - Interface FPGA-SRAM

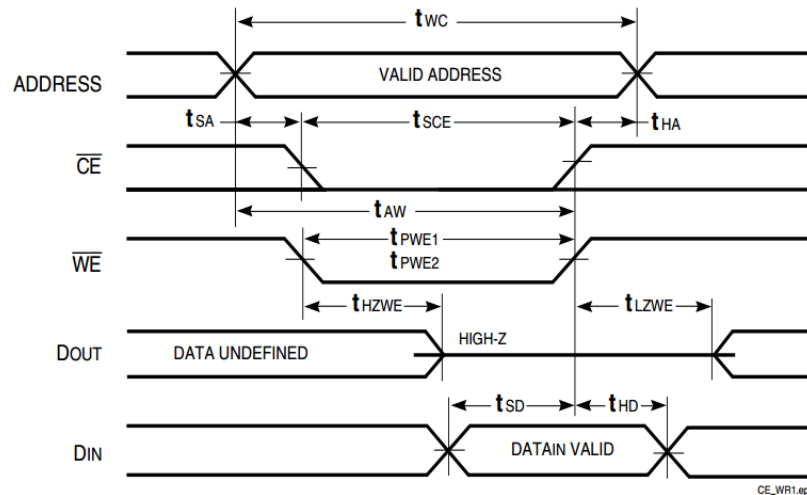
Se debe describir en VHDL el controlador de escritura y lectura ha ser implementado en el FPGA. Este controlador se debe basar en las características de interface (I/O timing) de la SRAM. Para ello se deben entender las descripciones de temporización de entradas y salidas detalladas en la hoja de datos de la memoria (la cual se puede bajar desde el siguiente link: [SRAM ISSI61WV102416BLL](#)).

Parte A

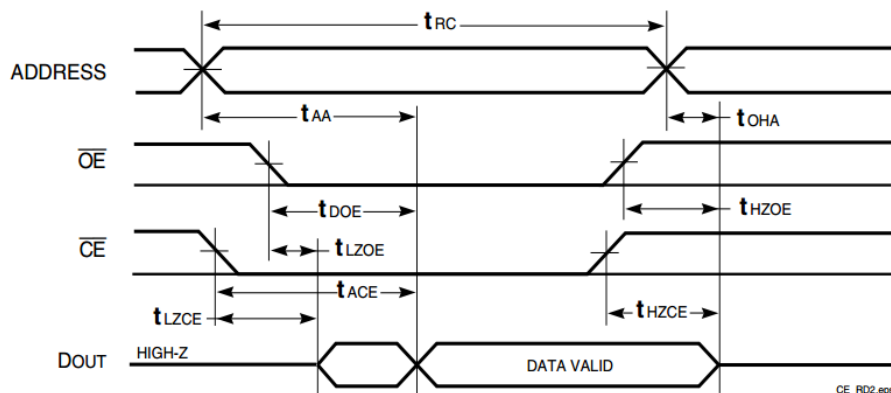
Describir en VHDL un controlador de escritura y lectura de la SRAM IS61WV102416BLL. El controlador se debe basar en la hoja de datos de la memoria.

Tener en cuenta los siguientes requerimientos:

- El controlador debe tener una entrada que indicará si se comienza un ciclo de lectura o un ciclo de escritura.
- Para el ciclo de escritura, implementar el ciclo *Write Cycle 1*.



- Para el ciclo de lectura, implementar el ciclo *Read Cycle 2*.



- Puede agregar las entradas y salidas que crea conveniente.

Parte B

Escribir el test bench respectivo para realizar una simulación a nivel de compuertas (gate level) del controlador de la memoria SRAM IS61WV102416BLL. Verificar los parámetros de tiempo de la hoja de datos de la memoria.

Opcional: describir en VHDL un **modelo** de la memoria SRAM IS61WV102416BLL, basado en la respectiva hoja de datos. Ejecutar el Test Bench del paso anterior, pero esta vez conectando el modelo de la SRAM.

Parte C

Describir en VHDL un sistema, a ser implementado en el FPGA, que cumpla con los siguientes requerimientos:

1. Usando MegaWizard generar una ROM de 4K x 16.
2. Generar el correspondiente archivo .mif.
 - a. Asignar valores para cada dirección de la memoria ROM usando la interface gráfica del .mif disponible en Quartus.
3. Utilice el archivo .mif para inicializar la ROM (en MegaWizard).
4. Cuando se presione un pulsador escribir todos los datos de la ROM a la SRAM externa.
5. Indicar con un LED el fin del proceso de escritura.
6. Al presionar otro pulsador comenzar a leer de la SRAM, de a un dato, y comparar con el respectivo dato en la memoria ROM. *El direccionamiento de la lectura debe ser usando un generador aleatorio (LFSR u otro).*
7. Encender un LED rojo indicando error en caso que los datos comparados no sean iguales. En este caso mostrar la dirección del error en los display 7 segmentos. Hasta que no se presione otro pulsador no continuar con la lectura y comparación.
8. Si la comparación finaliza sin errores, hacer titilar un LED verde con una frecuencia de 1Hz.
9. Escribir el respectivo archivo de restricciones, teniendo en cuenta los parámetros de tiempos respectivos. El sistema en FPGA es “source synchronous”.
10. Escribir un test bench que permita una simulación funcional de todo el sistema. Verifique la correcta funcionalidad del sistema.

En la siguiente figura se muestra un diagrama de bloque (aproximado) del sistema a implementar. Ud. debe dibujar e implementar el diagrama en bloque del sistema que propone como solución a los requerimientos planteados (incluirlo en el informe del laboratorio).

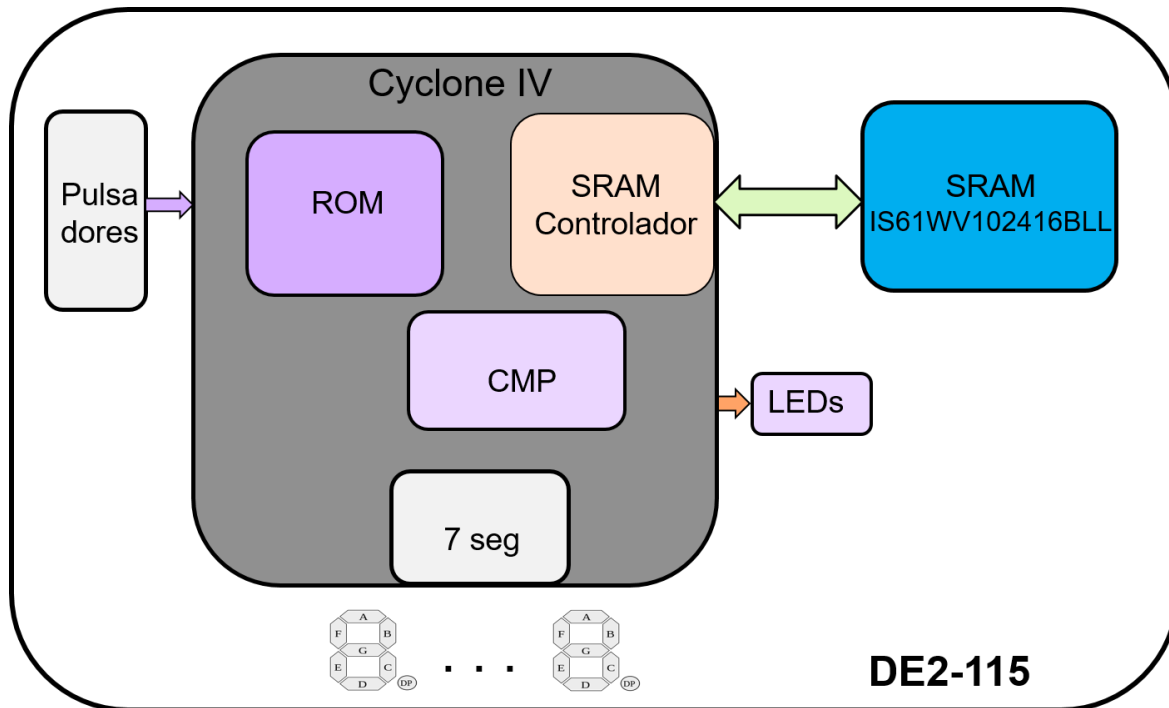


Figura 2 - Ejemplo de diagrama de bloque del sistema a implementar

Parte D

Si se han seguido correctamente los requerimientos y se ha cumplido con los parámetros de tiempo de la SRAM, el sistema implementado en la parte C debería funcionar correctamente. En esta parte del laboratorio se inducirá un dato erróneo a fin de forzar un error en el sistema.

Durante el proceso de lectura de los datos en la SRAM, introduzca un error en la comparación de los datos en la dirección 617_{10} .

Entonces, una vez finalizada la parte C del laboratorio, se comienza un nuevo ciclo de comparación (por ejemplo al presionar un nuevo pulsador), en este nuevo ciclo se debe detectar el error propuesto.

Informe

Presentar un informe completo del diseño siguiendo las reglas especificadas. Los códigos escritos para los distintos componentes deben ser agregados al informe como apéndices. Presentar un ítem describiendo los problemas encontrados y sus respectivas soluciones. Como así también los *warning* generados por el Quartus y sus soluciones.