



Análisis de la respuesta temporal del diseño de un controlador de SSRAM

Dilaila Criado Cruz, Víctor Escartín Fernández, Sonnia Pavoni Oliver

RESUMEN / ABSTRACT

Los sistemas digitales son cada día más rápidos y complejos. Por esta razón, desde la concepción de un diseño hasta la validación de su funcionamiento, es necesario realizar un detallado estudio de la respuesta temporal de los mismos. En ese sentido, este trabajo tiene como objetivo central analizar la respuesta temporal de un sistema digital, desde el diseño hasta la verificación. El sistema desarrollado se implementó en un FPGA EP3C25F324C6 de la familia Cyclone III de Altera y consistió en un controlador para la memoria SSRAM IS61LPS25636A, cuya frecuencia de reloj máxima es 200 MHz. Se emplearon el programa Quartus II y su herramienta *TimeQuest Timing Analyzer*. Con el método de diseño utilizado, corroborado por los análisis y herramientas empleadas, se garantizó un controlador con un margen de seguridad mínimo en el tiempo de *setup* de 155 ps a una frecuencia de reloj de 190 MHz.

Palabras claves: SSRAM pipeline, FPGA, Controlador

Abstract: The digital systems are every day faster and more complex. Therefore, from the conception of a design until the validation of their operation, it is necessary to carry out a detailed time analysis. In that sense, this work has as central objective the analysis of the timing performance of a digital system, from the design until the verification. The memory controller was implemented in the EP3C25F324C6, FPGA of the family Cyclone III from Altera. The memory IS61LPS25636A, manufactured by ISSI (Integrated Silicon Solution, Inc.) was used. The design was focused to work with the greater clock frequency as possible. The analysis of the design was carried out using the simulation tools Modelsim and TimeQuest Timing Analyzer from Quartus II. It was obtained an SSRAM controller with a worst setup slack of 155 ps for a clock frequency of 190 MHz.

Key words: pipelined SRAM, FPGA controller

Título en Inglés Timing analysis of a SSRAM controller design.

INTRODUCCIÓN

El desarrollo alcanzado en las tecnologías de fabricación de circuitos integrados, ha permitido la construcción de transistores más pequeños y consecuentemente, la disminución de los tiempos de respuesta¹. Es por ello que hoy en día es posible diseñar circuitos digitales más rápidos, que pueden realizar más operaciones en . Sin embargo, las aplicaciones y los requerimientos actuales resultan grandes retos para el diseño y la verificación de los sistemas digitales debido a su creciente complejidad. Para desarrollar sistemas digitales complejos la metodología de diseño sincrónico es la más importante³. Los circuitos globalmente sincrónicos o simplemente circuitos sincrónicos utilizan biestables controlados por una única señal de reloj. Esto no solo facilita la síntesis del circuito sino, sobre todo, la prueba y verificación del sistema final. Según los sistemas digitales se vuelven más complejos, aumenta la necesidad de tener en cuenta desde la concepción del diseño, la repercusión de los parámetros temporales, así como la utilización de las herramientas para el análisis de la respuesta temporal como método de validación⁴.

En sistemas digitales es común la utilización de circuitos integrados de memorias para aumentar la capacidad de almacenamiento de datos. Los requerimientos exigidos para la selección de estas memorias dependen enteramente de la aplicación donde se utilizan. En particular, las memorias estáticas y sincrónicas de acceso aleatorio (SSRAM) son las más rápidas, por lo que se seleccionan para las aplicaciones donde se requiera de baja latencia y para cuando se realizan varias lecturas o escrituras seguidas, como por ejemplo en los sistemas de comunicaciones^{5,6}. En la literatura se reportan diseños de controladores para diferentes tipos de memoria^{2,7,8,9}. Sharmay Singh presentan el diseño de un controlador de SRAM ZBT (*Zero Bus Turnaround*), utilizando la herramienta Xilinx ISE Design Suite 12.4². En la nota de aplicación de Altera⁹, se describe la funcionalidad de un controlador para memorias SRAM ZBT, donde se explica la estructura del controlador así como su instalación, compilación y simulación. Este diseño está optimizado para trabajar a 200 MHz en dispositivos Stratix y Stratix GX, que son dentro de los FPGA de ALTERA, la familia de mayores prestaciones. En general, los diseños que involucran interfaces con memorias externas constituyen grandes desafíos para lograr el cumplimiento de las especificaciones temporales, elemento clave para garantizar el correcto funcionamiento del sistema digital¹⁰.

Este trabajo tiene como objetivo central el análisis de la respuesta temporal de un circuito digital, desde su diseño hasta la verificación. El diseño desarrollado se implementó en un FPGA EP3C25F324C6, disponible en la tarjeta de desarrollo Cyclone III FPGA Starter Board del fabricante Altera¹¹ y consistió en un controlador para una memoria SSRAM IS61LPS25636A fabricada por ISSI (Integrated Silicon Solution, Inc.), cuya frecuencia de reloj máxima es 200 MHz. Se estableció que el diseño debe cumplir los requerimientos temporales en condiciones de peor caso en cuanto a la dispersión de los parámetros del proceso de fabricación, temperatura y tensión de alimentación (PVT por sus siglas en inglés). Se emplearon el programa Quartus II y su herramienta *TimeQuest Timing Analyzer*. Con el método de diseño utilizado, corroborado por los análisis y herramientas empleadas, se garantizó un controlador con un margen de seguridad mínimo en el tiempo de *setup* de 155 ps a una frecuencia de reloj de 190 MHz.

MATERIALES Y MÉTODOS

Se utilizó la tarjeta de desarrollo Cyclone III FPGA Starter Board¹¹, que tiene el FPGA EP3C25F324C6 de la familia Cyclone III del fabricante Altera. La familia Cyclone III es la tercera generación de FPGA de bajo costo y de los primeros FPGA con tecnología de 65 nm. El FPGA EP3C25F324C6, tiene 24624 celdas, 324 terminales, trabaja en el rango de temperatura comercial (0°C - 85°C) y es el más rápido de esta familia¹². La tarjeta dispone de un oscilador de 50 MHz conectado a una de las entradas de reloj del FPGA. Esta tarjeta también dispone de una memoria estática, sincrónica (SSRAM) y una memoria *Flash*, estas memorias comparten 16 bits del bus de datos y el bus de direcciones.

La SSRAM es la IS61LPS25636A¹³ fabricada por ISSI (*Integrated Silicon Solution, Inc.*), está organizada en 262144 palabras de 36 bits cada una. Esta memoria maneja niveles lógicos definidos por el estándar LVCMOS y trabaja a una frecuencia máxima de reloj de 200 MHz. Es una memoria del tipo *pipeline*¹⁴, ya que las entradas y las salidas pasan a través de registros que almacenan con el frente positivo del reloj, por lo que tiene una latencia de un periodo de reloj en la lectura, como se muestra en la figura 1. En la escritura después de una lectura, este tipo de memoria requiere dos ciclos inactivos (IDLE en la figura 1).

La conexión de la SSRAM en la tarjeta de desarrollo tiene inhabilitadas varias señales de control, y en esta situación solo se tiene acceso a las señales de control ADSC (habilitación de la dirección del controlador), CE (selección del circuito), BWE (habilitación de la escritura de bytes), las señales BWx (selección de bytes) y OE (habilitación de salida). No está permitido el modo ráfaga (*burst*) por lo que para escribir o leer es necesario establecer la dirección y activar las señales ADSC y CE.

En caso de escritura también hay que combinar la señal BWE con una o más de las señales BWx, lo que permite escribir una palabra completa o seleccionar cuál o cuáles bytes se escribirán. La lectura permite obtener una palabra completa a partir de activar la señal OE.

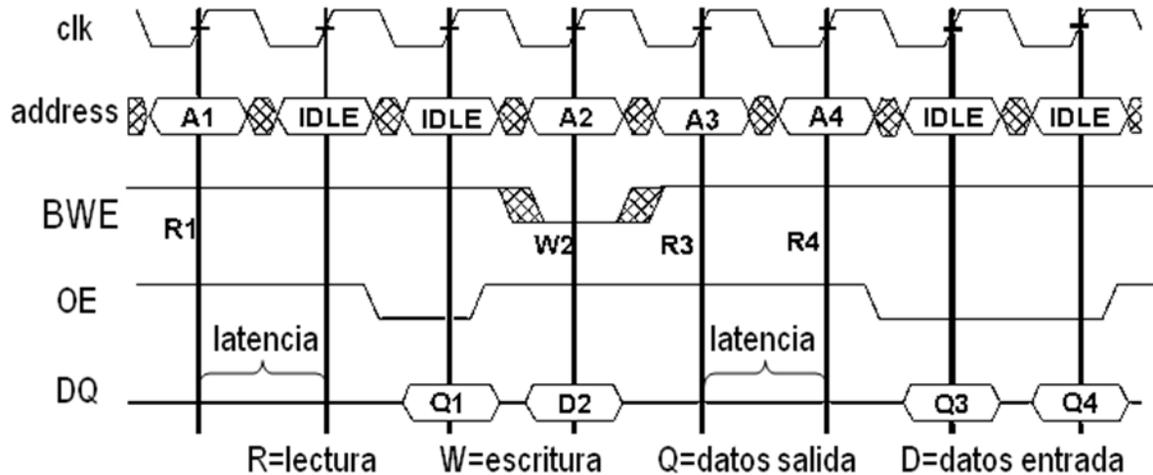


Figura 1. Diagrama temporal de operaciones de lectura y escritura en las SSRAM tipo pipeline.

El diseño del controlador de SSRAM, como todo desarrollo de un sistema digital, consistió en un proceso de refinamiento y validación que comenzó en un nivel de abstracción grande, y se transformó gradualmente hasta llegar a la implementación física. En este caso se siguió el siguiente flujo de diseño ¹⁵:

- Diseño de la arquitectura del sistema
 - Descripción y síntesis de los bloques que integran la arquitectura.
 - Simulación para la verificación funcional (ModelSim).
 - Proceso de refinamiento (de ser necesario).
- Diseño detallado
 - Establecer y añadir las restricciones al diseño.
 - Posicionamiento y ruteo (*fitter*) del diseño para el FPGA seleccionado y verificación temporal (TimeQuest Analyzer y ModelSim).
 - Proceso de refinamiento (de ser necesario)
 - Rediseño y ajuste de las restricciones.
 - Sintetizar el diseño y verificar el cumplimiento de los requerimientos temporales.
- Realizar la implementación
 - Realizar la programación del dispositivo.
 - Realizar la verificación de la implementación.

La arquitectura diseñada se describió en VHDL y se utilizó el programa Quartus II versión 13.0 sp1 (32-bits) de Altera para la implementación del diseño. Para la verificación funcional se implementó un banco de prueba y se utilizó el programa ModelSim.

La etapa de diseño detallado tiene que garantizar el cumplimiento de las especificaciones temporales del controlador. Las restricciones se definieron en un fichero con el formato Synopsys Design Constraints (.sdc) ^{16,17} donde se establecieron parámetros tales como: la frecuencia del reloj de entrada y la frecuencia del reloj interno de trabajo ¹⁸.

Después del posicionamiento y el ruteo se realizó el análisis estático de la respuesta temporal¹⁹ con las peores condiciones (PVT) para la verificación del cumplimiento de los requerimientos de tiempo. El análisis estático se realizó con la herramienta TimeQuest Analyzer del programa Quartus II. También se utilizó el banco de prueba diseñado y la simulación temporal con el programa ModelSim donde se incluyen las demoras de los dispositivos para los diferentes modelos.

Además se estableció con la herramienta Board Trace Model del Quartus, a cada terminal de la interfaz la capacidad de carga externa asociadas a la SSRAM y la FLASH que comparte los 16 bits menos significativos del bus de datos. Estas asignaciones son tenidas en cuenta por la herramienta de análisis para estimar con mayor exactitud el tiempo que demoran las señales desde el circuito de salida correspondiente en el FPGA hasta los terminales de la SSRAM.

Verificación del diseño

La verificación es el proceso de comprobar si un diseño cumple con las especificaciones que se le establecieron. La verificación tiene dos aspectos: funcionalidad y prestaciones. La verificación funcional comprueba cuándo un sistema genera la respuesta deseada. Las prestaciones son representadas por las restricciones temporales y se utiliza la verificación temporal para comprobar si la respuesta es generada dentro de las restricciones temporales. Estas restricciones comunican los requerimientos del diseño a la herramienta de implementación y además permiten que la herramienta de análisis temporal pueda reportar correctamente las demoras de las señales en todos los nodos del diseño realizado. La verificación temporal normalmente se expresa en términos de máxima demora de propagación y mínima frecuencia de reloj. La verificación se realiza en diferentes fases del diseño y a diferentes niveles de abstracción³. Existen varios métodos de verificación: la simulación funcional y de la respuesta temporal, el análisis de la respuesta temporal y la emulación del hardware.

La simulación funcional permite verificar el correcto comportamiento del circuito sin tener en cuenta las demoras de los componentes. Para esto se crea un banco de prueba donde se definen vectores de entrada y se chequea que los vectores de las salidas cumplan con la funcionalidad del diseño.

La simulación de la respuesta temporal permite comprobar el correcto comportamiento del diseño considerando también el tiempo que tardan en responder las diferentes componentes e interconexiones para las peores condiciones (PVT)²⁰. Este tipo de simulación es más compleja y demora más en su ejecución pero es más realista que la simulación funcional. Las simulaciones de la respuesta temporal de un diseño son útiles para:

- Permitir verificar la validez de algunas de las restricciones temporales establecidas como pueden ser falsos camino (*false path*) y caminos de ciclos múltiples. (*multi cycles path*).
- Verificar las condiciones iniciales del sistema.
- Permite analizar la inclusión de puntos de prueba (*design for test*)
- Realizar simulaciones para estimar el factor de conmutación (*switching factor*) para el cálculo de la disipación de potencia.

El análisis estático de la respuesta temporal chequea, para las peores condiciones, los requerimientos de las demoras estáticas del circuito sin utilizar vectores de entrada. Este involucra el análisis de registro a registro, de registros a salidas, de entradas a registros además de los caminos de los *reset* asincrónicos, solo se aplica para circuitos sincrónicos. Para el análisis estático de la respuesta temporal en las peores condiciones (PVT), se requirió de la herramienta TimeQuest Analyzer del programa Quartus II. Esta herramienta determina las relaciones de los tiempos que tienen que lograrse para el correcto funcionamiento del diseño, y chequea los tiempos de llegada de los datos (*data arrival time*) en relación con los tiempos requeridos (*data required time*) para verificar el funcionamiento del circuito y detectar las posibles violaciones de los tiempos²¹. Para la verificación de la respuesta temporal, la herramienta calcula las demoras de propagación de todos los caminos, chequea las restricciones de tiempo y reporta los resultados. Uno de estos resultados son los márgenes de seguridad (*slack*) en las peores condiciones (PVT). Los márgenes negativos indican violaciones de los requerimientos temporales. El TimeQuest Analyzer realiza el análisis estático del sistema no solo teniendo en cuenta los tiempos internos del FPGA sino también tiempos de dispositivos externos y parámetros de la tarjeta²².

ALTERA provee para el cálculo de la respuesta temporal para diferentes condiciones de operación (PVT) los tres modelos que se muestra en la tabla 1. Cada condición de operación es utilizada para modelar los tiempos de demora bajo condiciones de temperatura, tensión de alimentación y condiciones del proceso de fabricación²³. Los modelos Slow 85°C y el Fast 0°C describen el comportamiento típico de los dispositivos con largo de canal mayores de 65 nm, donde el primero de ellos reporta la respuesta más lenta y el segundo la respuesta más rápida de los dispositivos. El modelo Slow 0°C es para considerar la situación conocida como inversión de la temperatura²³, que ocurre para dispositivos con largo de canal de 65 nm o menos como el Cyclone III, en donde la tensión de umbral del transistor disminuye con la temperatura lo que puede provocar una respuesta más lenta por esta condición.

Tabla 1 Características de los tres modelos.

Modelo	Dispersión de parámetros	Tensión de alimentación	Temperatura
Slow 85°C	Los que producen la respuesta más lenta	Mínima	85°C
Slow 0°C	Los que producen la respuesta más lenta	Mínima	0°C
Fast 0°C	Los que producen la respuesta más rápida	Máxima	0°C

RESULTADOS Y DISCUSIÓN

Descripción y síntesis de los bloques que integran la arquitectura del Módulo Controlador

Se diseñó y se describió en VHDL la arquitectura del circuito siguiendo una estructura jerárquica, donde el bloque de mayor nivel se identificó como Módulo Controlador, como se muestra en la figura 2. Las señales que interactúan con la memoria tienen el prefijo “M_” y las señales internas del FPGA el prefijo “S_”. El módulo controlador se diseñó integrado por dos bloques: el bloque de control del reloj (clk_ctrl) y el controlador SSRAM. Todas las señales de control, excepto las de la memoria, se diseñaron activas en UNO. El controlador es parametrizable, lo que permite definir la cantidad de bits de datos y de direcciones de la memoria con lo que se facilita su utilización con otras SSRAM.

El bloque clk_ctrl contiene el circuito de inicialización (reset) de todos los registros del sistema. También utiliza un PLL para generar la señal de reloj interno (Clk) de 200 MHz para todo el módulo controlador a partir de los 50 MHz que recibe el FPGA del oscilador externo. Esta misma señal de reloj se envía por un terminal externo (M_Clk) a la entrada de reloj de la memoria. Donde único se utilizaron bibliotecas propias del fabricante ALTERA fue en la generación del reloj interno en el FPGA asociado con el PLL, por lo que para implementar este diseño con un dispositivo de otro fabricante solo hay que adecuar la descripción de este bloque.

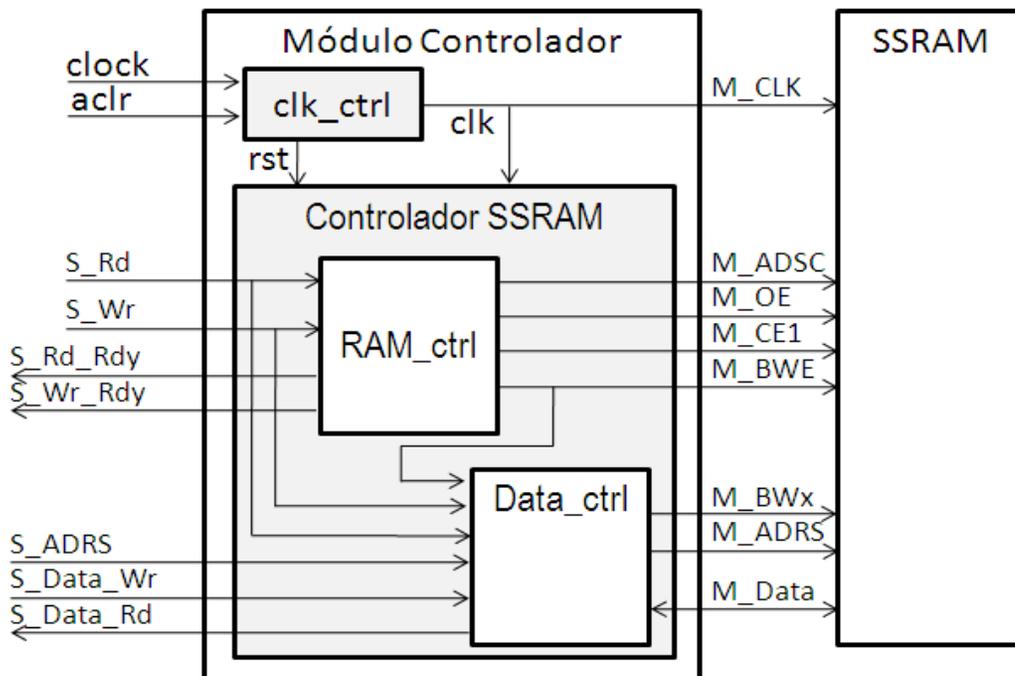


Figura 2. Diagrama de bloques del Módulo Controlador y su conexión con la SSRAM.

El segundo bloque es el Controlador SSRAM y se conformó por los módulos denominados RAM_ctrl y Data_ctrl (ver figura 2). El módulo RAM_ctrl, genera las señales de control para trabajar con la memoria y las señales que brindan la información necesaria para la utilización del controlador. Se implementó con una máquina de estados finita tipo Mealy con sus salidas registradas para que estas señales de salida demoren lo menos posible. El bloque Data_ctrl se dedica al manejo de los datos, en él se registra la dirección de memoria a la que se quiere tener acceso. Además, este bloque tiene un registro para almacenar el dato a escribir y otro para almacenar el dato leído. El resultado de la síntesis del diseño reportó un total de 114 elementos lógicos y 109 registros utilizados.

Simulación funcional

Para la verificación del módulo controlador se diseñó un banco de prueba que consta de tres bloques funcionales, que se muestran en la figura 3. El bloque Generador de Estímulos se encargó de la generación de las señales para cada una de las comprobaciones que se llevaron a cabo. El bloque del centro de la figura 3 corresponde al dispositivo bajo prueba, es decir el Módulo Controlador. El tercer bloque describe a la SSRAM. Para la verificación de circuitos como este que tiene una componente externa es imprescindible contar con un modelo de la misma. El modelo de SSRAM empleado se diseñó a partir de las informaciones que aparecen en la hoja de datos del fabricante de la SSRAM ¹³.

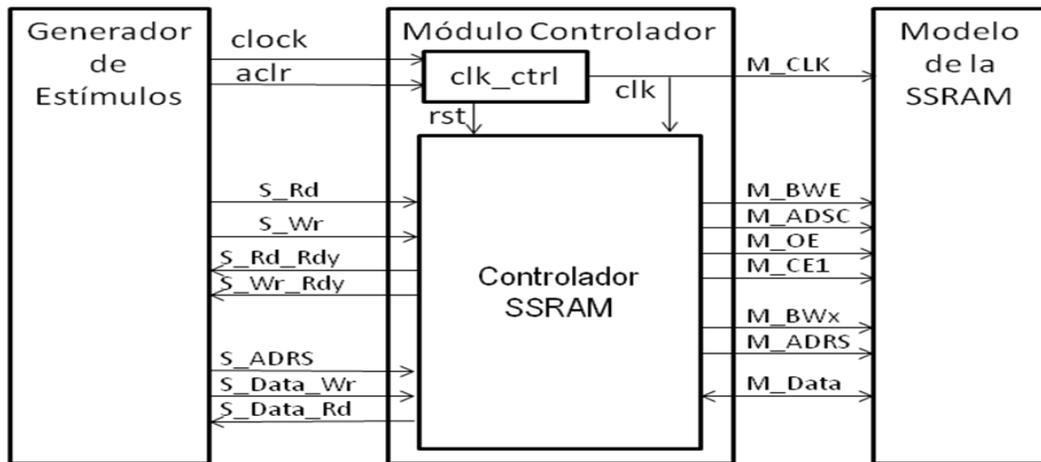


Figura 3. Diagrama de bloques del banco de prueba.

En la figura 4 se presentan los resultados de la simulación funcional obtenidos del programa ModelSim, donde se muestran las señales de entrada y salida del controlador para las operaciones de escritura y lectura. Primeramente se realizó la escritura consecutiva de dos datos. Para ello, a partir de los 120 ns se establecieron, por el generador de estímulos, consecutivamente las direcciones 1 y 2 en S_ADRS, respectivamente los datos 0x00000003 y 0xF00F000C en S_Data_Wr y se activó la señal de escritura. S_Wr.

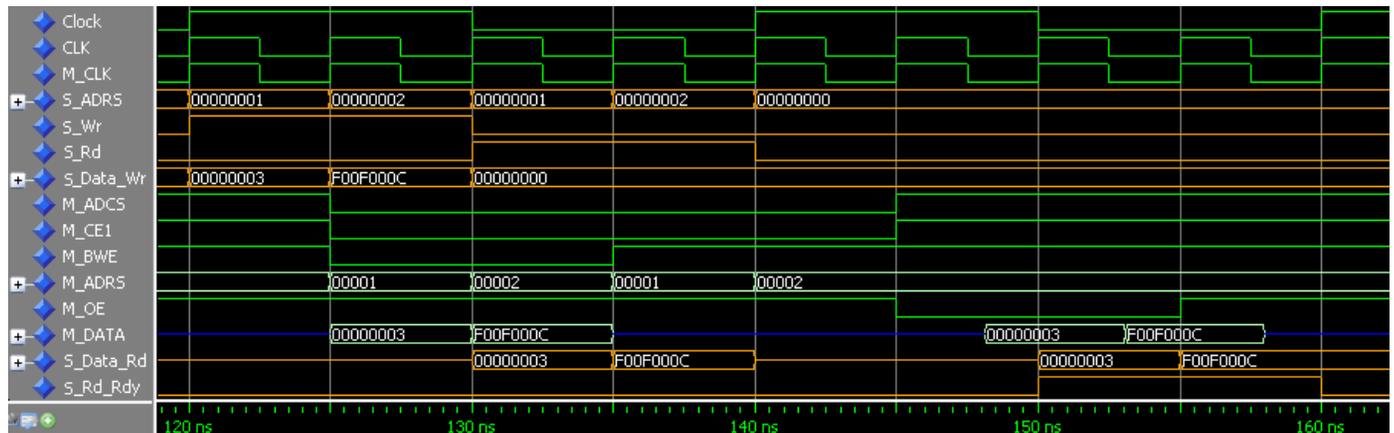


Figura 4. Simulación funcional del circuito controlador de SSRAM.

Como consecuencia de estos estímulos el controlador genera las señales para la memoria relacionadas con la escritura, es decir el bloque Data_ctrl genera las señales M_ADRS, M_DATA y M_BWE y el bloque RAM_ctrl genera las señales MBWE, M_ADSC y M_CE1.

Seguidamente se comprobó el resultado de la escritura mediante la lectura de los datos anteriormente escritos. En este caso, a partir de los 130 ns se establecen nuevamente las direcciones 1 y 2 y se activa la señal de lectura S_Rd. Como puede observarse, cada dirección aparece en la entrada de la memoria M_ADRS en el siguiente frente positivo del reloj. En el próximo frente del reloj la dirección es almacenada en la memoria y después del tercer frente del reloj (desde que el generador de estímulos estableciera la dirección en S_ADRS) el dato está disponible en M_Data, debido al período de latencia propia de la memoria con salidas registradas (*pipeline*). De esta forma se corrobora que el dato leído coincide con el dato escrito. Cuando la señal S_Rdy se activa, el dato leído está disponible en la salida del controlador.

Establecer y añadir las restricciones al diseño

En estos diseños es importante gobernar el proceso de posicionamiento y ruteo mediante restricciones temporales²⁴. Como ya se planteó, estas restricciones comunican los requerimientos del diseño a la herramienta de implementación y además permiten que la herramienta de análisis temporal pueda reportar correctamente las demoras de las señales en todos los nodos del diseño realizado. Debido a que el diseño interactúa con la SSRAM (circuito externo), se le definieron restricciones a todas las señales de entrada y salida del FPGA. Esto se realizó declarando en el fichero sdc los tiempos de *setup* y *hold* (tsu y th) que impone la memoria al FPGA y el tiempo de respuesta de la memoria (tco)¹³.

Se modelaron las conexiones del FPGA que están definidas en la tarjeta de desarrollo asociadas a la SSRAM. A través de la herramienta Board Trace Model del Quartus se tuvo en cuenta los terminales del bus de direcciones y de datos de la SSRAM que están compartidos con la memoria *Flash* por lo que se consideraron las respectivas capacidades acopladas a través de una resistencia de 22 Ω presente en el impreso²⁵. También se asignaron los pines establecidos en la tarjeta de desarrollo.

Se decidió que tanto los registros de salida como los registros de entradas de las señales que interactúan con la memoria fueran posicionados en la periferia del FPGA para minimizar las demoras de los registros a los terminales del FPGA y de los terminales del FPGA a los registros y la memoria. Con este propósito se le asignaron a dichos registros la opción *Fast Output Register*, *Fast Input Register*, y *Fast Output Enable Register* a través del editor de asignaciones del Quartus II.

Posicionamiento y ruteo (*fitter*) del diseño para el FPGA seleccionado y verificación temporal

Después del posicionamiento y ruteo y para cumplir con las restricciones temporales, se utilizaron finalmente 21 elementos lógicos y 137 registros en total, de los cuales 122 son registros de entrada/salida en la periferia del FPGA.

Para verificar el cumplimiento de los requerimientos temporales se analizaron los reportes de la herramienta TimeQuest Analyzer además de realizarse simulaciones considerando las demoras de los componentes y en las interconexiones utilizando ModelSim. El TimeQuest Analyzer reportó violación por 410 ps del ancho mínimo del pulso del reloj de la memoria, M_Clk. Para resolver este problema se aumentó la corriente que maneja el terminal de esta señal a su valor máximo (16 mA), con lo que se logró un margen positivo de 382 ps. En estas condiciones se lograron márgenes de seguridad positivos en los tres modelos para la operación de escritura en la SSRAM, pero en la operación de lectura se obtuvieron márgenes negativos para los tres modelos en el tiempo de *setup*, siendo el peor 1408 ps para el modelo Slow 85°C.

En el reporte del TimeQuest Analyzer se apreció una demora (*skew*) entre los dos relojes, el interno en el FPGA (CLK) y el generado para la memoria (M_CLK), que representa el 41% del periodo del reloj. Esta diferencia es la que más contribuye a los márgenes negativos en la lectura. Para mejorar este resultado se utilizó en el diseño el módulo ALTCLKCLTR²⁶, de biblioteca parametrizada propia de Altera (LPM por su siglas en inglés), que conecta la salida del PLL con el terminal dedicado de salida M_Clk. Esto permitió en la lectura disminuir el peor valor del *skew* a un 12.4 % del periodo del reloj aunque siguieron negativos los márgenes de seguridad en los dos modelos Slow y resultó positivo el margen de seguridad en el modelo Fast 0°C. Esta situación afectó los márgenes de seguridad en la escritura obteniéndose valores negativos para el modelo Slow 85°C solo en los pines de datos 31 y 17. La afectación en la escritura se debe a que el reloj que lanza el dato en la operación de lectura (M_Clk) es el reloj que captura el dato en la operación de escritura por lo que en la medida en que el desplazamiento entre ellos mejore una operación empeorará la otra.

A partir de reconocer que los terminales de datos 31 y 17 son terminales de tensión de referencia (Vref) del FPGA, que tienen mayor capacitancia¹² que los terminales regulares de entrada y salida, se les configuró para que manejen la corriente máxima (16 mA).

Esto disminuye el tiempo de respuesta de los mismos, lo que solucionó el problema en la escritura lográndose en esta operación márgenes positivos para todos los modelos, siendo el peor margen 78 ps para el modelo Slow 85°C.

Con las nuevas condiciones, la herramienta TimeQuest Analyzer reportó márgenes de seguridad positivos para la lectura en el modelo Fast 0°C y el modelo Slow 85°C pero para el modelo Slow 0°C reportó margen de seguridad negativo. En este modelo se violaron por los 32 bits de datos (M_DATA) las restricciones de tiempo en la lectura de la memoria. El valor para el peor caso en el margen de seguridad en el tiempo de *setup* fue de -108 ns. De la evaluación de los resultados que se obtienen de la herramienta TimeQuest Analyzer se determina que en la lectura, los peores márgenes de tiempo están definidos por los terminales de la SSRAM que se comparten con la memoria *Flash* de configuración. Esto teóricamente pudiera mejorarse si se utiliza para la configuración una memoria serie y además no se compartiera con ella ningún terminal. En la tabla 2 se muestran los márgenes de seguridad para los tiempos de *setup* y de *hold* en la lectura y la escritura reportados para cada uno de los modelos utilizados con una frecuencia interna de trabajo de 200 MHz. Una posibilidad reportada en la literatura ⁹ para lograr los requerimientos de tiempo es aprovechar la posibilidad que brindan los PLL de generar relojes con diferentes fases entre ellos. Se rediseñó el PLL con dos salidas de reloj uno para el sistema y otro para la memoria, pero en este caso no se encontró ninguna diferencia de fase que pudiera satisfacer los tres modelos ya que la diferencia entre los márgenes es muy pequeña en el modelo Slow 85°C.

Con estas condiciones una opción para lograr márgenes de seguridad positivos es disminuir la frecuencia del reloj de trabajo. Se determinó, a partir de las ecuaciones que utiliza la herramienta TimeQuest Analyzer para el cálculo de los márgenes de seguridad ²¹ y de los valores reportados por esta, que la frecuencia de reloj máxima para la cual todos los modelos cumplen con las restricciones temporales es 195.77 MHz. Con el mismo procedimiento se calculó que para lograr un margen de seguridad para el tiempo de *setup* (de peor caso) de al menos el 3% del periodo del reloj (155 ps) la frecuencia de trabajo debe ser de 190 MHz. En la tabla 3 se muestran los márgenes de seguridad para los tiempos de *setup* y de *hold* reportados por la herramienta TimeQuest Analyzer para cada uno de los modelos utilizados con una frecuencia de reloj de 190 MHz.

Tabla 2. Peores márgenes de seguridad reportados por TimeQuest Analyzer para una frecuencia de reloj de 200 MHz.

Modelo	Margen <i>Setup</i> (ps)	Margen <i>Hold</i> (ps)
SLOW 85°C		
Reloj que envía CLK, reloj que recibe M_CLK (escritura)	78	1335
Reloj que envía M_CLK, reloj que recibe CLK (lectura)	29	2560
SLOW 0°C		
Reloj que envía CLK, reloj que recibe M_CLK (escritura)	305	1082
Reloj que envía M_CLK, reloj que recibe CLK (lectura)	-108	2743
FAST 0°C		
Reloj que envía CLK, reloj que recibe M_CLK (escritura)	919	560
Reloj que envía M_CLK, reloj que recibe CLK (lectura)	396	2156

Tabla 3. Peores márgenes de seguridad reportados por TimeQuest Analyzer para una frecuencia de reloj de 190 MHz.

Modelo	Margen <i>Setup</i> (ps)	Margen <i>Hold</i> (ps)
SLOW 85°C		
Reloj que envía CLK, reloj que recibe M_CLK (escritura)	292	1335
Reloj que envía M_CLK, reloj que recibe CLK (lectura)	341	2560
SLOW 0°C		
Reloj que envía CLK, reloj que recibe M_CLK (escritura)	155	1082
Reloj que envía M_CLK, reloj que recibe CLK (lectura)	568	2743
FAST 0°C		
Reloj que envía CLK, reloj que recibe M_CLK (escritura)	659	560
Reloj que envía M_CLK, reloj que recibe CLK (lectura)	1182	2156

En la figura 5 se muestra el resultado de la simulación donde se incluyen las demoras del dispositivo empleando la herramienta ModelSim de ALTERA con el modelo de Slow 0°C, se utilizó el banco de prueba descrito anteriormente. Para evaluar la respuesta temporal de la memoria tanto en la escritura como en la lectura, se generaron estímulos para escribir dos datos (0x00000003 y 0xF00F000C) de forma consecutiva en las direcciones 1 y 2 de la SSRAM, seguido de la lectura del primer dato.

Después de la operación de lectura se esperan tres periodos de reloj, señalados por *idle* en la figura 5, y se escriben los datos 0x00000020 y 0x00000030 en las direcciones 4 y 5 respectivamente.

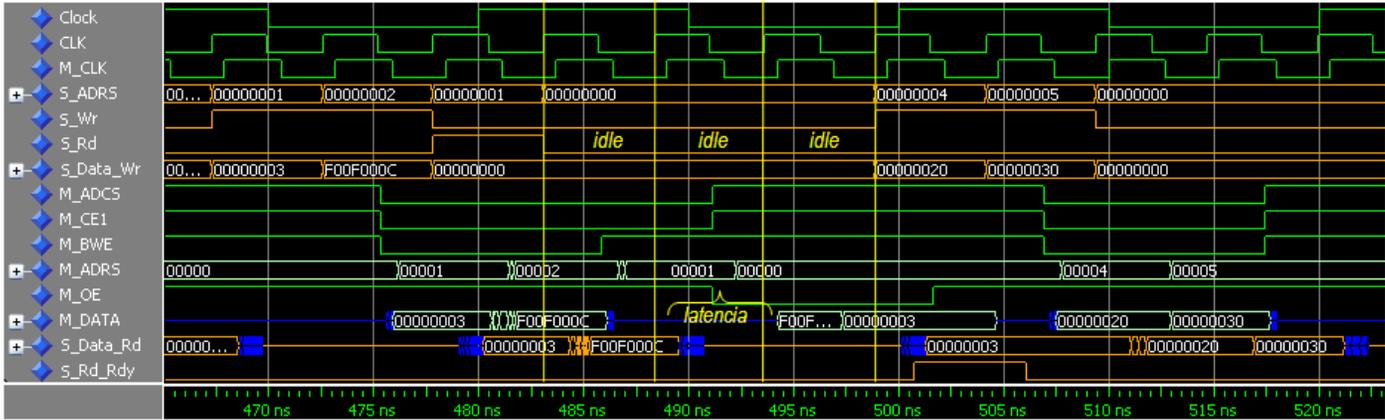


Figura 5. Simulación del circuito controlador de SSRAM incluyendo las demoras del dispositivo para el modelo Slow 0°C a190 MHz.

La simulación de la respuesta temporal, permitió comprobar que el módulo diseñado escribe y lee en la memoria de forma consecutiva como describe el manual de la memoria a 190 MHz. Cuando la operación que se realiza es una lectura seguida por una escritura es necesario incorporar un ciclo de espera adicional a lo reportado en las memorias del tipo *pipeline* ya que puede existir contienda en el bus de datos de la memoria²⁷.

A partir del análisis de la interfaz entre los terminales de datos del controlador y la memoria se concluyó que, para evitar la contienda se debe cumplir, en el peor caso, la siguiente condición:

$$t_{QHZ_{max}} < t_{zx_{min}}(M_Data) \tag{1}$$

donde:

$$t_{QHZ_{max}} = t_{comax}(M_OE) + t_{OEHz_{max}}(SSRAM) \tag{2}$$

$t_{comax}(M_OE)$: es el tiempo máximo de demora con relación al reloj del sistema en que la señal M_OE se desactive.

$t_{OEHz_{max}}(SSRAM)$: es el tiempo máximo en que después de deshabilitarla salida de datos de la SSRAM, esta va a alta impedancia.

$t_{zx_{min}}(M_Data)$: es el tiempo mínimo de demora con relación al reloj del sistema en que las señales M_Data aparecen en los terminales de salida.

A partir de los valores del *Data Sheet Report* de la herramienta TimeQuest Analyzer para el FPGA y los reportados por el manual de la SSRAM¹³, se comprobó en este caso que $t_{QHZ_{max}}$ es 6.825 ns y $t_{zx_{min}}(M_Data)$ es 3.541 ns, por lo que hay contienda al no cumplirse la condición (1).

CONCLUSIONES

En el presente trabajo se han diseñado y evaluados los módulos que integran un controlador para la SSRAM IS61LPS25636A fabricada por ISSI (Integrated Silicon Solution, Inc.) que se encuentra en la tarjeta de desarrollo Cyclone III Starter Board. Este controlador es implementado en el circuito EP3C25F324C6, FPGA de la familia Cyclone III del fabricante Altera, y puede adecuarse a dispositivos lógicos de otros fabricantes e incluso modificarse para satisfacer necesidades específicas, ya que solo se utilizaron funciones propias del fabricante para la generación de la frecuencia del reloj interna.

El método de diseño, los análisis temporales estáticos realizados y las herramientas de análisis empleadas permitieron determinar la frecuencia máxima de trabajo del diseño con márgenes de seguridad positivos para el peor caso, tomando en cuenta la dispersión de parámetros en el FPGA y la SSRAM, el intervalo de temperatura comercial y la variación en la tensión de alimentación.

El diseño realizado garantiza un controlador que trabaja con márgenes de seguridad positivos para frecuencias de reloj hasta 195.77 MHz para el peor caso. Si se utiliza una frecuencia de reloj de 190 MHz el peor de los márgenes de seguridad para el tiempo de *setup* ocurre en el proceso de lectura y es 155 ps.

REFERENCIAS

1. **Kiran Kumar, Kaushik Battachariyya**, "FPGA implementation of System-on-chip architecture for space craft application", *The International Journal of Engineering and Science*, Vol 3, Issue 6, pages 16-24, 2014
2. **Smriti Sharma and Balwinder Singh** "Design and performance analysis of ZBT SRAM controller", *International Journal of VLSI design & Communication Systems (VLSICS)*, Vol.4, No.3, June 2013
3. **Pong P. Chu**, *RTL HARDWARE DESIGN USING VHDL Coding for Efficiency, Portability, and Scalability*, ed. John Wiley & Sons, Inc., 2006
4. **Quartus II Handbook Version 11.0: Section II, Timing Analysis**, Altera Corporation, Vol 3: Verification, May, 2011
5. **AN1116 Introductions to Cypress SRAMs**, Cypress Semiconductor Corporation, June 2006
6. **AN4011 Choosing the right Cypress SRAMs**, Cypress Semiconductor Corporation, November 2006
7. **Kang Li; Hongye Jia; Honghu Gong; Jiangyi Shi; Peijun Ma**, "Optimization of QDR SRAM Controller in Network Processor," Fourth International Symposium on Computational Intelligence and Design (ISCID), 2011, vol.1, pp.254-257. DOI :10.1109/ISCID.2011.7128-30 Oct. 2011
8. **Abhishek Kumar, Yogesh. E. Wankhede, Kirti Shinde P., Nisha Sarwade**, "Design of SDRAM Memory Controller using VHDL", *International Journal of Computer Science and Application Issue*, pp. 184-189, ISSN 0974-0767 2010.
9. **AN329 1.0: ZBT SRAM Controller Reference Design for Stratix & Stratix GX Devices**, Altera Corporation, March 2004
10. **Application Note AN 584-1.0: Timing Closure Methodology for Advanced FPGA Designs**, Altera Corporation, August 2009
11. **Cyclone III FPGA Starter Board Reference Manual**, Altera Corporation, April 2007
12. **Cyclone III Device Handbook**, Vol 1 Chapter 1, Altera Corporation, October 2008
13. **Data Sheet SSRAM, IS61LPS25636A, ISSI**, Integrated Silicon Solution, Inc., February 2009
14. **Application notes, Understanding static RAM operation**, IBM, 1997
15. **Application Note AN204-1.2: Using Modelsim Altera in a Quartus II Design Flow**, Altera Corporation, December 2002.
16. **Ryan Scoville TimeQuest User Guide**, Wiki Release, December 2010
17. **AN 466: Cyclone III Design Guidelines**, Altera Corporation, November 2008
18. **AN 433: Constraining and Analyzing Source Synchronous Interfaces**, Altera Corporation, February 2014
19. **White Paper "TimeQuest Timing Analyzer: Native SDC Support for Timing Analysis of FPGA-Based Designs"**, Altera Corporation, May, 2006
20. **White Paper "Gate-Level Simulation Methodology"**, Cadence Design Systems, Inc., May, 2015
21. **Quartus II Handbook Version 13.1: Timing Analysis Overview TimeQuest Terminology and Concepts**, Altera Corporation, Vol 3 Chapter 6, June,2012
22. **Quartus II Handbook Version 13.1: The Quartus II TimeQuest Timing Analyze**, Altera Corporation, Vol 3, Chapter 7, November 2013
23. **White Paper "Guaranteeing Silicon Performance with FPGA Timing Models"**, Altera Corporation, August, 2010
24. **Jack Marshall**, "RTL Coding and Optimization Guide for use with Design Compiler", Tera Systems Inc., SNUG, San José, 2002
25. **Board schematic and layout**, CycloneIII_SB_3C25, Altera Corporation, April 2007
26. **Clock Control Block (ALTCLKCTRL) Megafunction, User Guide**, Altera Corporation, February, 2014
27. **Application Brief AN-203ZBT SRAMs: System Design Issues and Bus Timing**, Integrated Device Technology, Inc. June 2001

AUTORES

Dilaila Criado Cruz, Ingeniera Física Nuclear, Master en Electrónica, Investigadora Agregada del Centro de Investigaciones en Microelectrónica de la Facultad de Eléctrica de la CUJAE, La Habana, Cuba. dilaila.criado@electronica.cujae.edu.cu

Víctor Escartín Fernández, Ingeniero en Telecomunicaciones, Doctor en Ciencias Técnicas, Profesor Auxiliar del Centro de Investigaciones en Microelectrónica de la Facultad de Eléctrica de la CUJAE, La Habana, Cuba. victor.escartin@electronica.cujae.edu.cu

Sonia Pavoni Oliver, Ingeniera en Telecomunicaciones y Electrónica, Doctora en Ciencias Técnicas, Master en Electrónica, Profesora Auxiliar del Centro de Investigaciones en Microelectrónica de la Facultad de Eléctrica de la CUJAE, La Habana, Cuba sonnia.pavoni@electronica.cujae.edu.cu