BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Generación de Memorias con Quartus**	**CÓDIGO:** **BT1_A4_P4**

FECHA:			
NOMBRE:		APELLIDOS:	

MODALIDAD:		TIPO:	Presencial	DURACIÓN:	85 minutos

CALENDARIO:	Sesión presencial S14	REQUISITOS:	

CRITERIO DE ÉXITO:	

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTOEVALUACIÓN: [entre 0 y 10 puntos]	No procede

1. Descargas Moodle y creación de carpetas

Además de este documento, para esta sesión de trabajo debe descargar:

Presentación: BT1_A4_P4
Zip: BT1_A4_Z1

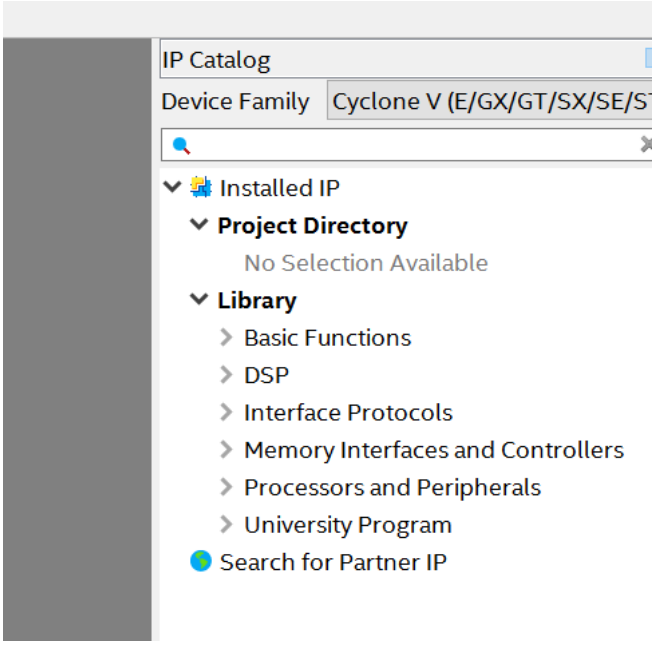
Descomprima BT1_A4_Z1. Verá que contiene 3 carpetas: RAM_altera, FIFO_altera y FIFO_custom. Cada una de estas carpetas contiene un banco de test VHDL que usted tendrá que utilizar para simular las memorias que utilice. Para hacer estas simulaciones no necesita seguir las convenciones que habitualmente utilizamos en la creación de proyectos (carpetas hdl, modelsim y quartus).
Debajo de la carpeta RAM_altera cree dos carpetas al mismo nivel: RAM_SIN_RSAL y RAM_CON_RSAL. Debajo de la carpeta FIFO_custom cree otras dos carpetas con los mismos nombres (RAM_SIN_RSAL y RAM_CON_RSAL). Finalmente, debajo de la carpeta FIFO_altera debe crear, en el mismo nivel, las carpetas NORMAL y AHEAD.

2. Generación de memorias con Quartus

Quartus dispone de un generador de módulos que permite crear distintos tipos de subsistemas que pueden emplearse como bloques funcionales emplazándolos en nuestros diseños.

Arranque Quartus Prime (v 2016). No es necesario que cree ningún proyecto *quartus* para realizar las tareas en esta sesión.

La herramienta que va a utilizar está disponible en una ventana que aparece en la parte derecha (si no está, puede visualizarse seleccionando la opción IP Catalog del menú Tools):



Dentro de la opción Library se definen diferentes grupos de funcionalidades, y dentro de cada una se pueden seleccionar IPs. El acrónimo IP, en este contexto, no es Internet Protocol, sino *Intelectual Property*. Un IP es un módulo prediseñado que se puede comprar o, en ocasiones, obtener de forma

gratuita para incluirlo como parte de nuestro diseño. Dentro del *IP Catalog* hay IPs gratuitos y otros que no lo son.

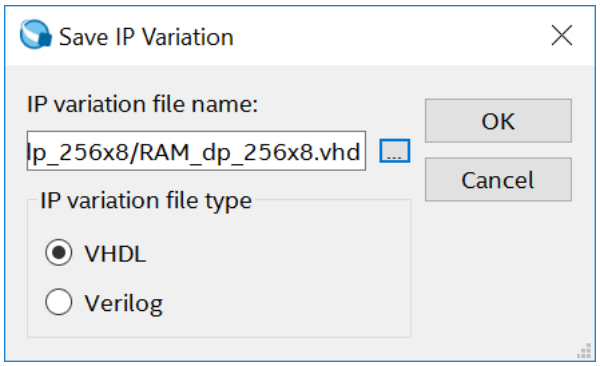
Los IPs tienen cierta dependencia con la familia de FPGAs utilizada. Seleccione ahora la MAX 10 como *Device Family*.

Seleccione (*library -> basic function -> on chip memory*), y observe que hay diferentes tipos de memorias síncronas que se pueden seleccionar.

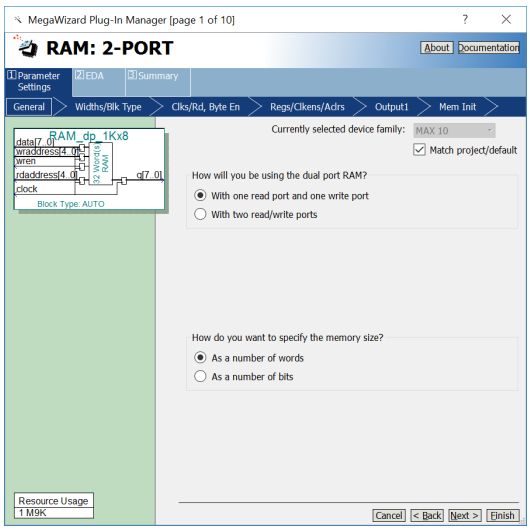
3. Generación de una RAM de doble puerto sin registro de salida

A continuación, va a generar una memoria RAM de doble puerto de 256 bytes sin salida registrada.

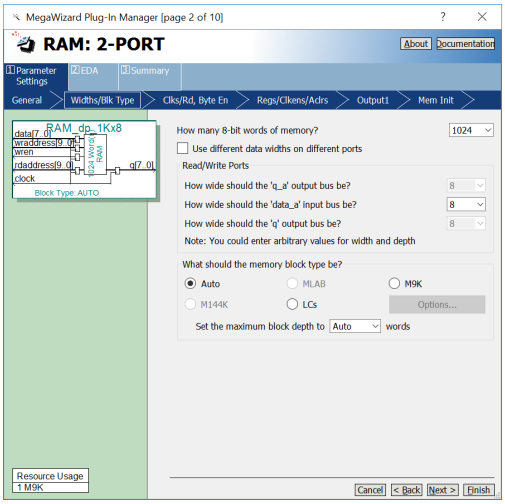
Para ello, haga doble *click* sobre las memorias RAM:2-PORT y en la ventana que aparece, indique la carpeta donde se va a almacenar el modelo generado (la carpeta RAM_SIN_RSAL que ha creado dentro de RAM_altera) y su nombre (*RAM_dp_256x8,* es importante que no se confunda en esto) y también que el modelo se desea generar en VHDL. Pulse *OK* para continuar.



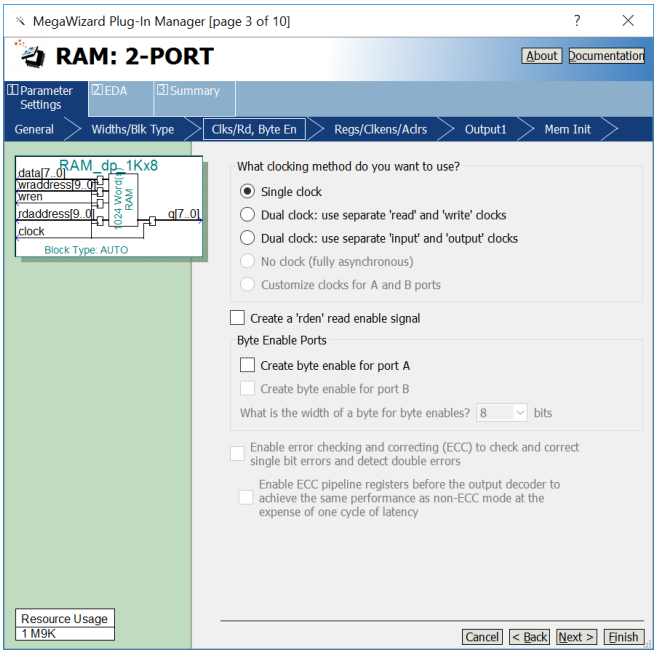
En la primera ventana que aparece a continuación se pregunta si se desea generar una *true double port* o *simple double port* (que sea simple: un puerto de escritura y otro de lectura). Hacer *click* en *Next* para continuar.



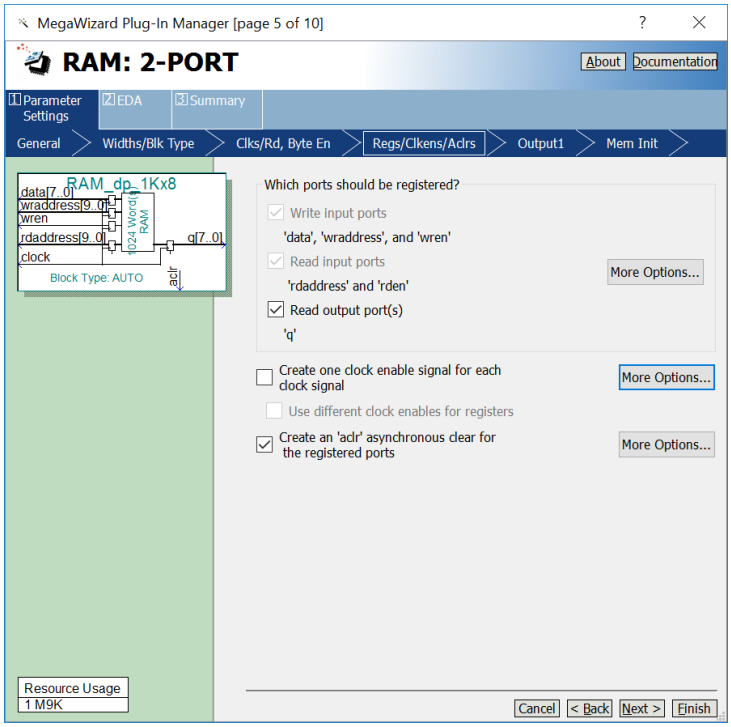
En la siguiente, lo relevante es seleccionar un tamaño de 256 bytes (no como en la figura, que son 1024). El resto de opciones se quedan como están. Haga *click* en *Next*.



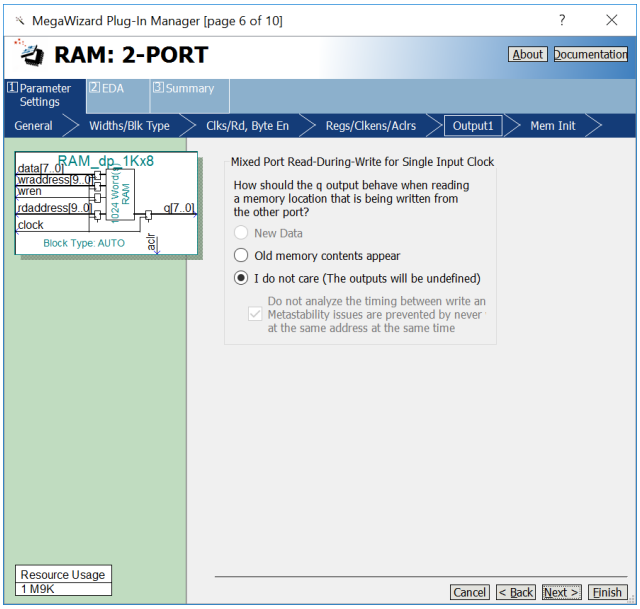
A continuación, se elige entre 1 y 2 relojes, Seleccione uno (*single clock*); el resto de opciones quedan como están.



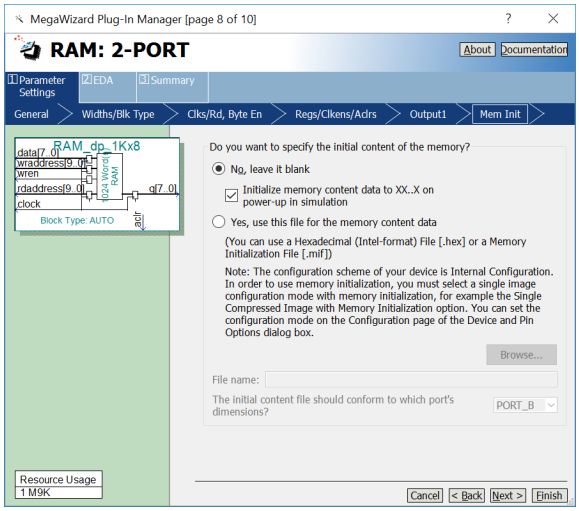
En la siguiente pantalla se elige entre lectura registrada o sin registrar (es la única opción de registro que se puede tocar). Modifique la selección en la casilla *read output ports* y observe el efecto que tiene esa modificación sobre el modelo de la memoria que puede verse en la esquina superior izquierda.
Si elige registrar la salida de la memoria, entonces se introducirá un retardo de un ciclo de reloj en la generación del dato de salida (*latencia*); a cambio, el camino (*path*) de retardo se segmenta, lo que favorece la obtención de una frecuencia máxima de trabajo (*fmax*) superior.
Seleccione la opción en la que no se registra la salida (casilla *read output ports* en blanco, no como en la figura). Seleccione también en esta ventana un clear asíncrono (reset asíncrono) para los registros y continúe (*Next*).



En la siguiente ventana se plantea la cuestión de cómo debe comportarse la memoria cuando se lee una posición que se está escribiendo; seleccione la opción *I don't care*. Continúe (*Next*).



A continuación, viene la configuración inicial para simulación. Realice la selección tal y como aparece en la siguiente figura para que el contenido de la memoria se inicialice a 'X' para la simulación. Continúe (*Next*).



En la penúltima ventana, haga *click* en *Next* sin modificar nada y en la última ventana, deje todas las casillas en blanco y haga *click* en *Finish*.

Si ha realizado correctamente todo el proceso, en la carpeta RAM_SIN_RSAL debajo de RAM_altera debería haber dos ficheros: *RAM_dp_256x8.vhd*, que contiene el modelo de la memoria que acaba de generar y *RAM_dp_256x8.qip*; este último no es necesario y puede borrarlo si lo desea. Si los ficheros se han generado en otra carpeta es porque no ha seguido las indicaciones al inicio del proceso; no importa, mueva el fichero *RAM_dp_256x8.vhd* a la carpeta RAM_SIN_RSAL y continúe con la tarea.

A continuación, deberá observar el comportamiento de la memoria en una simulación demostrativa que se le ha suministrado a tal efecto. Para ello:

- Cree un proyecto ModelSim en RAM_altera/RAM_SIN_RSAL (llámelo RAM_SIN_RSAL) e incluya el fichero generado con Quartus (que debe estar también en RAM_SIN_RSAL) y el test (que está en RAM_altera). Compile ambos ficheros.

- Abra el fichero que contiene el modelo de la memoria y observe su contenido. Se trata de un modelo que solo sirve para simulación, NO es un modelo RTL para síntesis. Al realizar el diseño físico, el sintetizador lógico utiliza uno o más bloques de memoria interna (M9K, en el caso de la MAX10) para implementar la memoria.

- Abra el fichero que contiene el test. Observe cómo se instancia la memoria. Este procedimiento de emplazamiento es el mismo que se utiliza en los diseños para incluir en ellos una memoria. Observe también el conjunto de pruebas que se realizan.

- Arranque la simulación, incluya las señales de la interfaz de la memoria, configure como unsigned las direcciones y datos para mayor comodidad y ejecute la simulación. Observe los resultados. En particular, fíjese en cómo se escribe en la memoria y cómo se lee de la memoria. Tenga en cuenta que las simulaciones que está realizando son funcionales (sin retardos). Así, en las operaciones de lectura, aunque el dato aparezca inmediatamente en la salida de la memoria tras un cambio en la dirección, ese dato solo podrá leerse de forma síncrona en el siguiente flanco activo de la señal de reloj.

- No cierre ModelSim

4. Generación de una RAM de doble puerto con registro de salida

A continuación, va a generar una memoria RAM con las mismas características que tiene la que ha generado en el apartado 2.1, pero con registro de salida. Para ello:

- Genere con Quartus una nueva memoria con las mismas características que la generada en el apartado 2.1, pero con registro de salida. La única diferencia con la anterior es que en esta ocasión debe seleccionar la casilla *read output ports*. Inicialmente debe indicar que el modelo generado debe guardarse en la carpeta RAM_CON_RSAL.

- Arranque un nuevo ModelSim sin cerrar el anterior.

- Cree un nuevo proyecto en la carpeta RAM_CON_RSAL (llámelo RAM_CON_RSAL). Incluya en ese proyecto el modelo generado con Quartus (que debe estar también en RAM_CON_RSAL) y el test (que está en RAM_ALTERA). El test sirve para ambos modelos (con o sin registro de salida) porque ambos se llaman igual. Compile los dos ficheros.

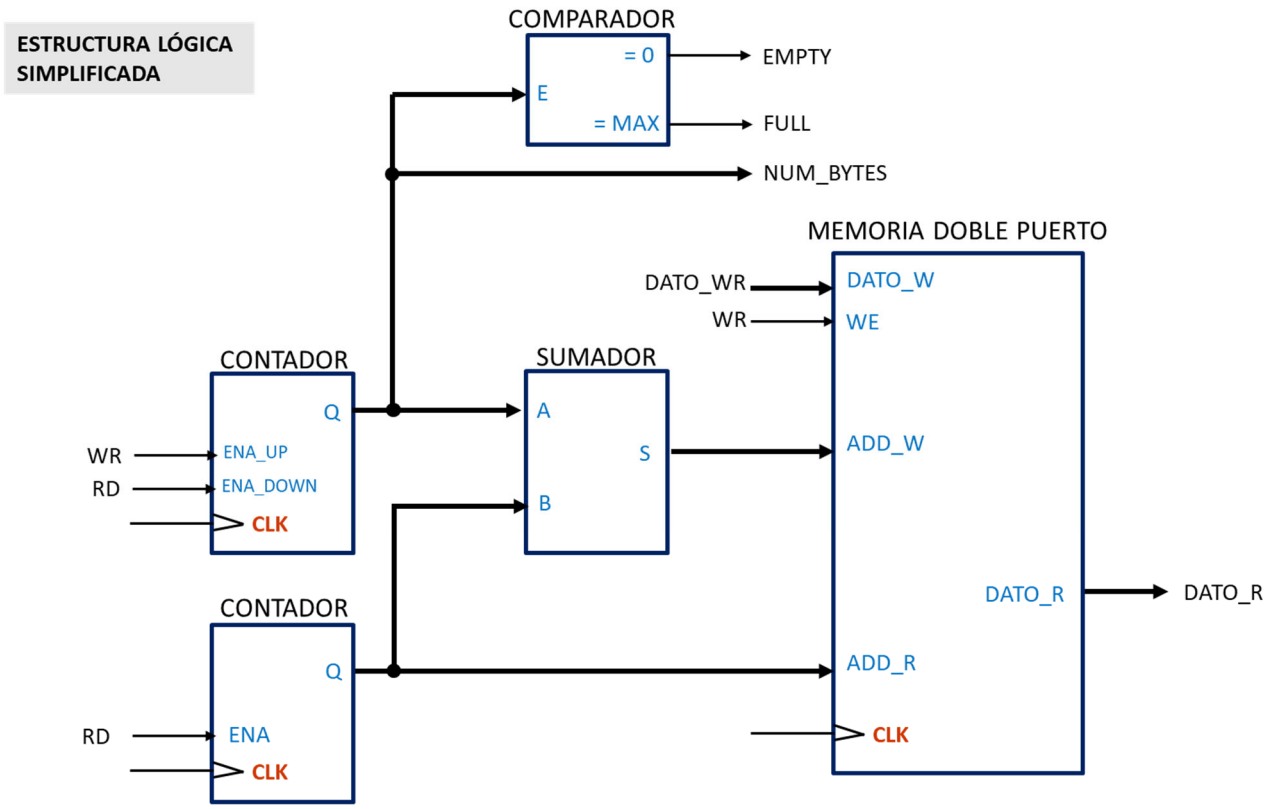
- Arranque la simulación, incluya las señales de la interfaz de la memoria, ejecute la simulación y observe los resultados. En particular, fíjese en cómo se escribe y cómo se lee de la memoria.

- Compare los resultados de la simulación con los de la que realizó en el apartado 2.1 (si tiene dudas acerca de cuál es cuál, en la parte superior de la ventana correspondiente a la pestaña *Project* en ModelSim aparece el nombre del proyecto). Observe cómo en las operaciones de lectura, en la simulación de la memoria con salida registrada se observa un retraso de un ciclo de reloj en la generación de la salida con relación a la salida que se obtiene con la memoria que tiene la salida sin registrar. Observe también que esta latencia no impide que puedan realizarse operaciones de lectura consecutivas a razón de una lectura en cada ciclo de reloj.

- Cuando termine puede cerrar las aplicaciones ModelSim abiertas si lo desea.

5. Generación de una FIFO con una RAM de doble puerto sin registro de salida

En este apartado se va a generar una memoria FIFO a partir de la RAM sin registro de salida generada en el apartado 2.1. El circuito tiene la estructura que se muestra en la figura de forma simplificada.



Podrá observar que:

- ✓ el circuito está compuesto por una RAM de doble puerto, un sumador, un contador ascendente con entrada de habilitación, un contador up/down y un comparador

- ✓ sin tener en cuenta el reloj y el reset asíncrono, la interfaz está compuesta por: las habilitaciones de lectura (RD) y escritura (WR), los datos de escritura (DATO_WR) y lectura (DATO_R) así como las señales que indican FIFO vacía (EMPTY), FIFO llena (FULL) y el número de bytes que hay en la FIFO (NUM_BYTES).

- ✓ para escribir un dato en la FIFO es necesario activar WR al tiempo que se introduce el dato por DATO_WR. En el siguiente ciclo de reloj el dato se escribe en la FIFO en ADD_W y el contador up/down se incrementa en una unidad, de forma que ADD_W apunta a la siguiente dirección.

- ✓ para leer un dato de la FIFO es necesario activar RD y el dato que esté en ADD_R podrá leerse en DATO_R en el siguiente ciclo de reloj. Además, ADD_R se incrementa porque así lo hace el contador ascendente pero ADD_W no cambia porque el contador up/down se decrementa al mismo tiempo.

- ✓ el contador up/down lleva la cuenta del número de bytes que hay en la FIFO porque se incrementa cuando hay una escritura y se decrementa cuando hay una lectura. Además, la salida EMPTY se activa cuando este contador está a cero y la salida FULL se activa cuando llega a MAX, que debe coincidir con el número de bytes que caben en la memoria.

A continuación, copie el modelo de memoria generado en el apartado 2.1, y que se encuentra debajo de RAM_altera/RAM_SIN_RSAL a la carpeta FIFO_custom/ RAM_SIN_RSAL. Cree un nuevo proyecto ModelSim en la carpeta FIFO_custom/ RAM_SIN_RSAL (llámelo FIFO_CUSTOM_RAM_SIN_RSAL) e incluya en él:

- ✓ el modelo de simulación de la memoria RAM de doble puerto de 256 bytes sin registro de salida (*RAM_dp_256x8.vhd*) que se encuentra en FIFO_custom/ RAM_SIN_RSAL.

- ✓ el fichero *fifo_256x8_dp.vhd*, que contiene el diseño de la FIFO, y que se encuentra en la carpeta FIFO_custom.

- ✓ el fichero *test_FIFO_custom.vhd*, que también está en la carpeta FIFO_custom.

Compile todos los ficheros; no debería haber errores.

Ahora abra el fichero *fifo_256x8_dp.vhd* y observe su contenido. En particular:

- ✓ observe cómo dentro del cuerpo de arquitectura se ha emplazado la memoria RAM de doble puerto que utiliza la FIFO.

- ✓ observe cómo se ha implementado la FIFO. En particular, compruebe que su funcionalidad se corresponde con el diagrama de la presentación BT1_A4_P4.

- ✓ observe también que hay alguna funcionalidad que no se ha incluido en el diagrama de bloques, con el objeto de impedir que se realicen lecturas si la FIFO está vacía o escrituras si está llena.

Abra también el fichero *test_FIFO_custom.vhd* y observe el conjunto de pruebas que se realizan.

A continuación, arranque la simulación, incluya las señales de la interfaz de la memoria, ejecute la simulación y observe los resultados. Para poder ver el funcionamiento de la FIFO, debe incluir en la simulación todas las señales internas del DUT.

Observe cómo, en las operaciones de lectura:

- ✓ inicialmente la FIFO muestra el dato correspondiente a la dirección (interna) de lectura que tiene la memoria RAM
- ✓ cuando se activa la habilitación de lectura de la FIFO, en el siguiente flanco activo de reloj, la dirección de lectura se incrementa
- ✓ en el siguiente flanco activo de reloj podrá leerse de forma síncrona el valor almacenado en la FIFO correspondiente al nuevo valor con el que se direcciona la RAM

No cierre ModelSim hasta finalizar el resto de los apartados de esta tarea.

6. Generación de una FIFO con una RAM de doble puerto con registro de salida

A continuación, va a trabajar con una memoria FIFO con las mismas características que tiene la que ha generado en el apartado 2.3, pero basada en una RAM de doble puerto con registro de salida. Para ello:

- ✓ Copie el modelo de memoria generado en el apartado 2.2, y que se encuentra debajo de RAM_altera/RAM_CON_RSAL a la carpeta FIFO_custom/RAM_CON_RSAL.

- ✓ Arranque un nuevo ModelSim (llámelo RAM_CON_RSAL) sin cerrar el anterior.

- ✓ Cree un nuevo proyecto ModelSim en la carpeta FIFO_custom/RAM_CON_RSAL e incluya en él el modelo de simulación de la memoria RAM de doble puerto de 256 bytes con registro de salida (*RAM_dp_256x8.vhd*) que se encuentra en FIFO_custom/ RAM_CON_RSAL, y los ficheros *fifo_256x8_dp.vhd* y *test_FIFO_custom.vhd*, que están en la carpeta FIFO_custom. Compile todos los ficheros; no debería haber errores.

- ✓ Arranque la simulación, incluya las señales internas del DUT, ejecute la simulación y observe los resultados.

- ✓ Compare los resultados de la simulación con los de la que realizó en el apartado 2.3. Observe las diferencias en la lectura de los datos dependiendo de si la FIFO está basada en una memoria RAM con o sin registro de salida.

- ✓ No cierre ModelSim hasta finalizar el resto de los apartados de esta tarea.

7. Generación de una FIFO con Quartus y con modo de funcionamiento normal

A continuación, se va a generar una FIFO directamente con el generador de IPs de Quartus. El procedimiento es el mismo que el que se emplea en la generación de las memorias RAM. En esta ocasión tendrá que seleccionar en el *IP Catalog* la opción FIFO en lugar de RAM:2-PORT. Los parámetros que debe tener en cuenta son los siguientes:

1.- Nombre del fichero y ubicación: FIFO_altera/NORMAL/*FIFO_dp_256x8* (no se equivoque, tiene que ser exactamente este nombre). Tipo de fichero: VHDL
2.- Familia: MAX 10, Reloj: un solo reloj, Tamaño: 256 bytes (256 palabras de 8 bits)
3.- Seleccionar las siguientes señales: full, empty, usedw, reset (clear) síncrono y reset (clear) asíncrono
4.-Seleccionar modo *normal*
5.-Resto de opciones: las que vienen por defecto; en la última ventana ponga todos los recuadros de selección en blanco y haga *click* en *Finish*.

- Cree un nuevo proyecto ModelSim (llámelo FIFO_NORMAL) en la carpeta FIFO_altera/NORMAL e incluya en él el modelo de simulación de la memoria FIFO que acaba de crear y el *test_FIFO_dp_256x8.vhd*, que está en la carpeta FIFO_altera. Compile ambos ficheros; no debería haber errores.

- Arranque la simulación, incluya las señales de la interfaz de la memoria, ejecute la simulación y observe los resultados. Observe que esta FIFO se comporta prácticamente igual que la FIFO que simuló en el apartado 2.3 (FIFO implementada con una RAM de doble puerto sin registro de salida).

- No cierre ModelSim hasta que no termine el siguiente (y último) apartado de esta tarea.

8. Generación de una FIFO con Quartus y con modo de funcionamiento show-ahead

A continuación, se va a generar una FIFO con el generador de IPs de Quartus, igual que la del apartado anterior, pero que trabaja en el modo *show-ahead*. Como ha visto en el apartado anterior, si la FIFO trabaja en modo *normal*, en una operación de lectura los datos están disponibles en la salida de la FIFO en el siguiente flanco de reloj después de que se activa la señal de lectura. En cambio, con el modo *show-ahead*, el dato estará presente en la salida de la FIFO antes de activarse la señal de lectura; una vez se activa, en el siguiente flanco de reloj, lo que aparece es el siguiente dato almacenado en la FIFO. De esta manera es posible realizar la lectura justo en el flanco de reloj siguiente al flanco en el que se activa la señal de lectura. Todo esto podrá apreciarlo en la simulación que llevará a cabo a continuación.

Genere la FIFO con Quartus siguiendo el mismo procedimiento que empleó en el apartado 2.5, con dos diferencias:

- ✓ la ubicación del modelo generado deberá ser FIFO_altera/ahead (el nombre de la memoria debe mantenerse).

- ✓ el modo de trabajo de la FIFO deberá ser *show-ahead* (en lugar de *normal*).

A continuación,

- Arranque una nueva copia de ModelSim

- Cree un nuevo proyecto ModelSim (llámelo FIFO_AHEAD) en la carpeta FIFO_altera/AHEAD e incluya en él el modelo de simulación de la memoria FIFO que acaba de crear y el *test_FIFO_dp_256x8.vhd*, que está en la carpeta FIFO_altera. Compile ambos ficheros; no debería haber errores.

- Arranque la simulación, incluya las señales de la interfaz de la memoria, ejecute la simulación y observe los resultados. Observe la diferencia en la lectura de los datos con relación a la que pudo ver con la FIFO en modo *normal*: en este caso, inicialmente, el dato que se lee ya está en la salida antes de activar la habilitación de lectura de manera que puede leerse en el siguiente flanco de reloj, momento en el que aparece en la salida el siguiente dato almacenado.