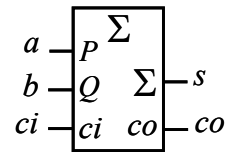


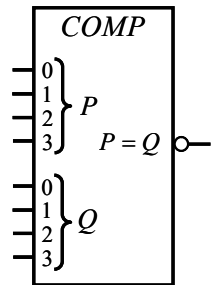
Tarea 7 correspondiente a las actividades no presenciales de la asignatura Hardware de aplicación específica (curso 2019-2020).

1) Lee detenidamente el pdf que está en fatic denominado Notas_VHDL_alumnos_resumidas. Una vez leído dicho pdf, responde a las siguientes preguntas:

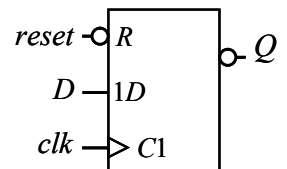
a) Escribe un código en VHDL (entidad y arquitectura) que describa el comportamiento del sumador total de 1 bit indicado en la parte derecha. En fatic están las diapositivas de SD



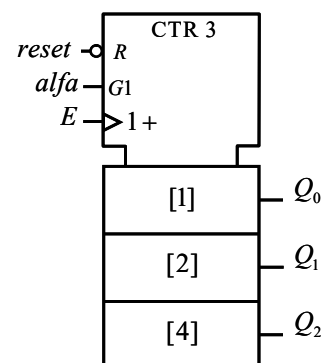
b) Escribe un código en VHDL (entity y architecture) que describa el comportamiento del comparador de magnitud de 4 bits indicado en la parte derecha.



c) Escribe un código en VHDL (entity y architecture) que describa el comportamiento del flip flop D sincronizado con los flancos de subida indicado en la parte derecha.



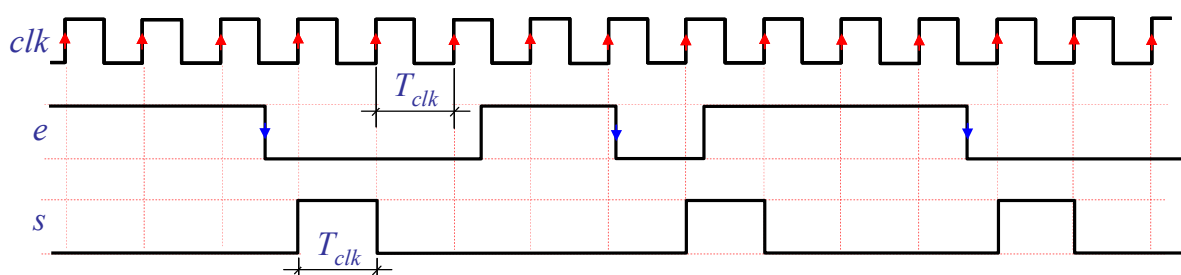
d) Escribe un código en VHDL (entity y architecture) que describa el comportamiento del contador de 3 bits indicado en la parte derecha.



e) Escribe un código en VHDL (entity y architecture) que describa el comportamiento de un circuito con 8 entradas, en cuyas salidas se indica (en binario) el doble del número de entradas que están a 1.

f) Escribe un código en VHDL (entity y architecture) que describa el comportamiento de un circuito al que llegan 4 número binarios, sin signo, $A(a_1 a_0)$, $B(b_1 b_0)$, $C(c_1 c_0)$ y $D(d_1 d_0)$, que proporciona el mayor de los cuatro números.

g) Escribe un código en VHDL (entity y architecture) que describa el comportamiento de un circuito detector de flancos de bajada. Dicho circuito se caracteriza porque su salida (s) se pone a 1 durante un tiempo dado (T_{clk}) cada vez que detecte que la señal de entrada (e) ha descrito un flanco de bajada. A continuación se muestra un ejemplo de la relación entre la señal de entrada y la señal de salida del circuito a diseñar.



Para simular los códigos de los ejercicios anteriores puedes utilizar el programa *ModelSim PE Student Edition*. Es un programa con un tamaño razonablemente pequeño (en el laboratorio de Electrónica se utiliza el ISE WebPack de Xilinx, cuyo tamaño es de 6,17GB). Aunque el programa *ModelSim* sólo vale para simular código, es suficiente en lo que se refiere a esta tarea. Para poder utilizar el programa *ModelSim* debes hacer lo siguiente:

i) Baja el programa *ModelSim PE Student Edition* de la siguiente página:

https://www.mentor.com/company/higher_ed/modelsim-student-edition

El fabricante (Mentor) obliga a registrarse. La licencia que proporciona es gratuita pero sólo es válida durante 6 meses (pasado dicho tiempo habría que volver a solicitar al fabricante una nueva licencia gratuita).

Haz clic en *Download student edition*, cubre el formulario que aparece en pantalla y haz clic en *Submit*. Esto hará que se inicie la descarga a tu ordenador del programa *Modelsim_pe_student_edition*. A la cuenta de correo que has tenido que indicar te deberá llegar un archivo de nombre '*student_license.dat*'.

ii) Instala el programa y guarda el archivo *Student_license.dat* en la carpeta en la que hayas instalado el programa *Modelsim_pe_student_edition*. Si no ha habido ningún problema con la instalación, el programa debería funcionar

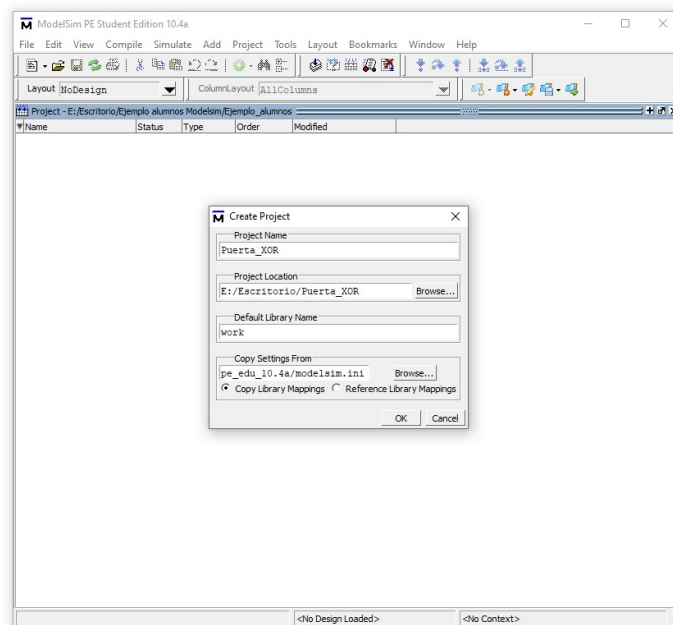
Para simular un código (la descripción en vhd de un circuito), primero hay que escribirlo y para ello hay que hacer lo siguiente:

1º Crear un proyecto: vete a *File – New Project*. En la ventana que se abre indica:

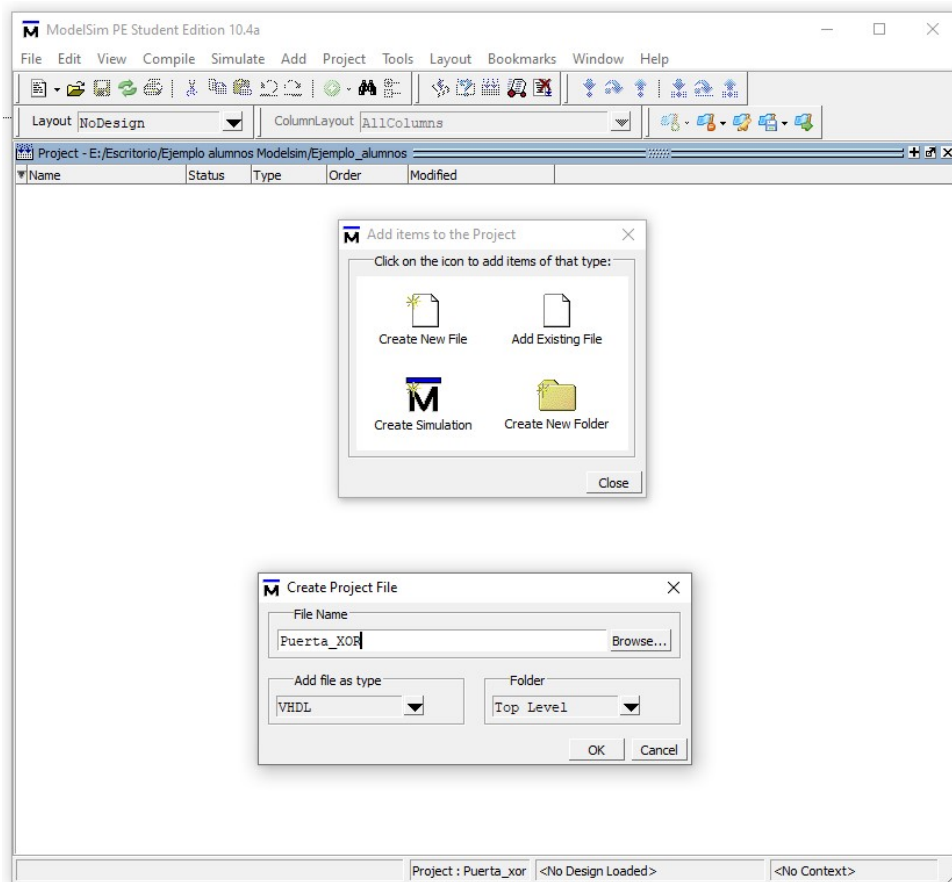
Project name: el nombre del proyecto

Project location: la carpeta en la que quieres que se guarden los archivos del proyecto.

A continuación haz clic en OK (si ya hay un proyecto abierto, al crear un nuevo proyecto te va a preguntar si quieres que se cierre el proyecto actual. Dile que si)



2º Escribir el código a simular: la creación del nuevo proyecto provoca que se abra una ventana (*Add items to the project*). Haz clic en el icono *Create new file*. En la ventana que se abre (*Create project file*), en *File name* indica el nombre del archivo vhd que guardará el código a simular. Dicho nombre deberá ser el mismo que el nombre de la entidad que figure en el código.



A continuación haz clic en OK y cierra la ventana *Add items to the project*.

En la parte izquierda de la pantalla aparecerá el nombre del archivo creado. Haz un doble clic sobre él con el ratón. En la parte derecha de la pantalla aparecerá una ventana en la que hay que escribir el código (en vhdl) que se quiere simular con el programa *ModelSim*.

Ejemplo:

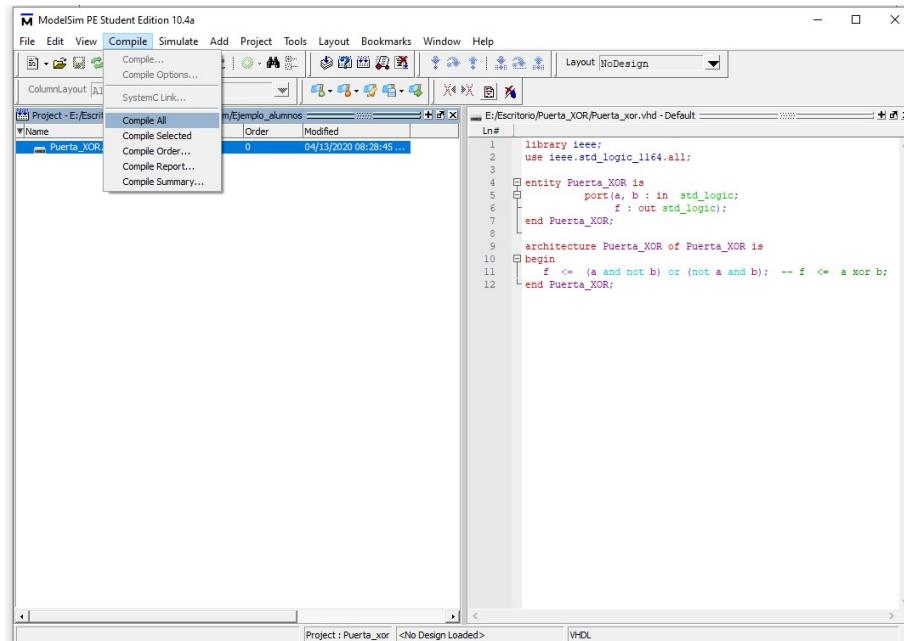
```
library ieee;
use ieee.std_logic_1164.all;

entity Puerta_XOR is
    port(a, b : in std_logic;
         f : out std_logic);
end Puerta_XOR;

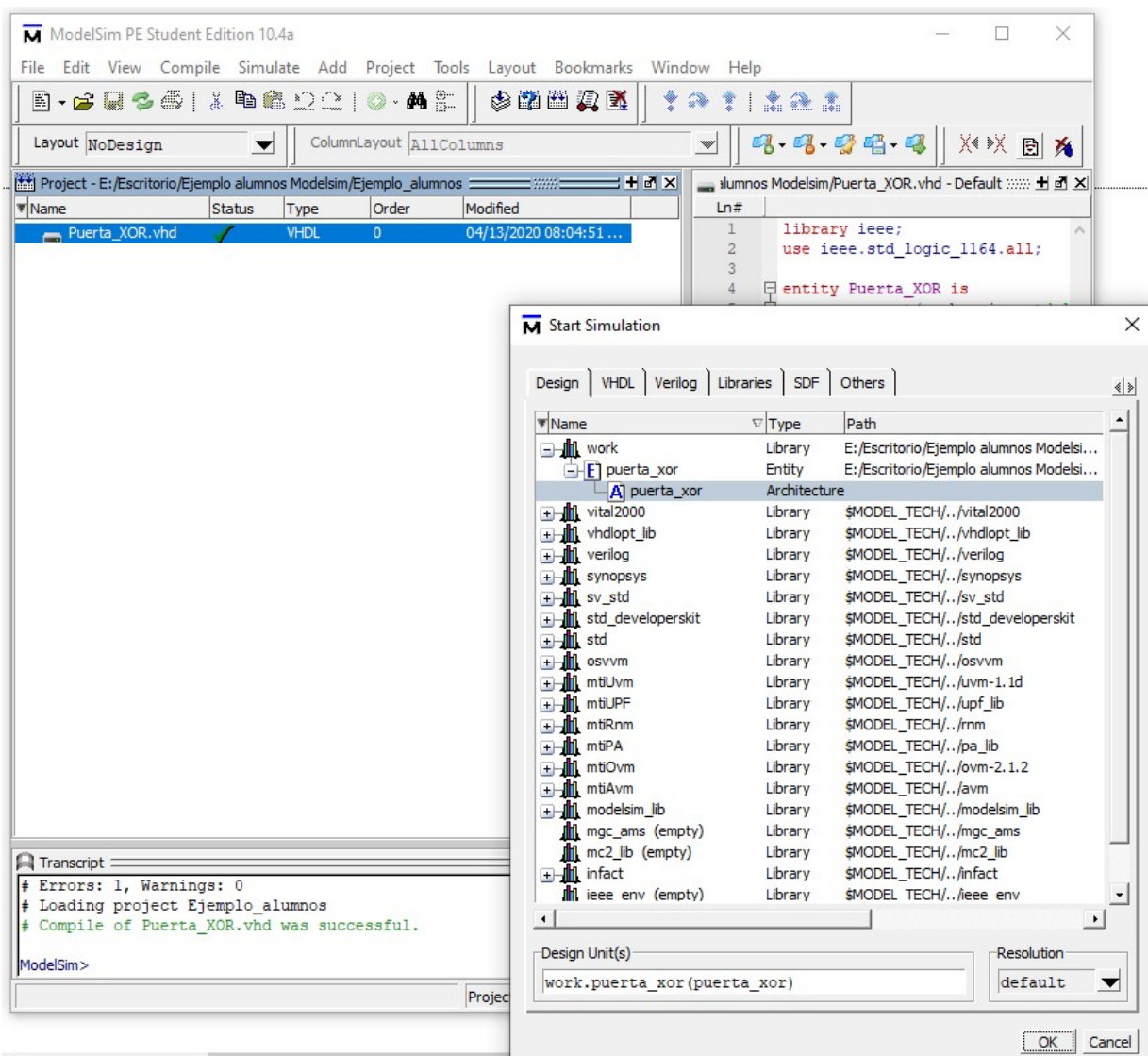
architecture Puerta_XOR of Puerta_XOR is
begin
    f <= (a and not b) or (not a and b); -- f <= a xor b;
end Puerta_XOR;
```

3º Compilar el código: una vez que hayas escrito el código a simular, guárdalo haciendo clic con el ratón en el botón guardar situado en la parte superior izquierda de la pantalla o bien haz clic en *File – Save*.

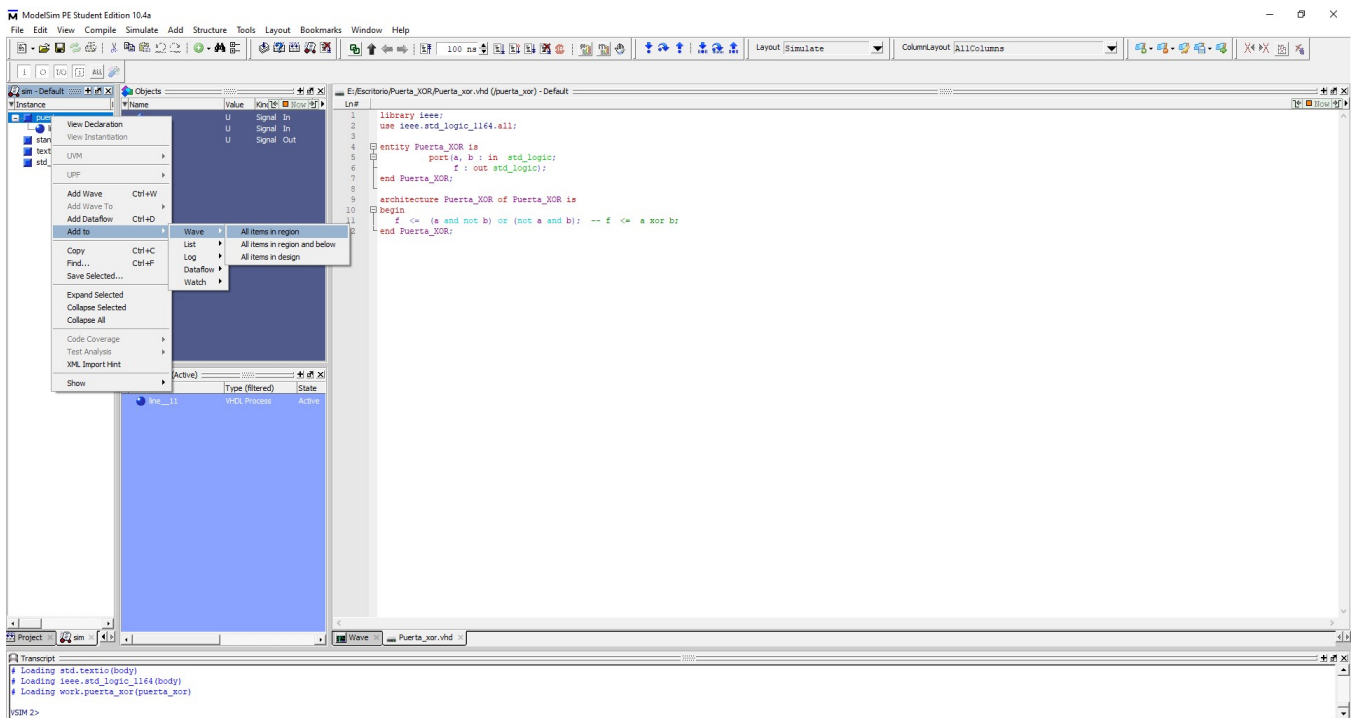
Para compilar el código escrito, en primer lugar selecciona con el ratón (si no lo está ya) el nombre del archivo vhdl que contiene el código a simular y que se muestra en la parte superior izquierda de la pantalla. A continuación haz clic en *Compile – Compile All*. En la parte inferior de la pantalla se indica si el programa ha encontrado algún error sintáctico en el código o no. En el caso de que encuentre algún error debes corregirlo, guardar el archivo con el código corregido y volver a compilarlo.



4º Simular el código: una vez que la compilación no indica ningún error ya se puede simular el código escrito. Para ello debes ir a la parte superior de la pantalla y hacer clic con el ratón en *Simulate – Start simulation*. En la ventana que se abre, expande *Work* y a continuación expande la entidad que hay en *Work* para que se muestre la arquitectura. Selecciona la arquitectura y haz clic en OK.



En la pantalla se abren dos nuevas ventanas. En la ventana de la izquierda (*Instance*), haz clic con el botón derecho en el nombre de tu proyecto y en el desplegable que se abre haz clic en *Add to – wave – All items in region*. Esto hace que se abra una nueva ventana en la parte derecha de la pantalla llamada *Wave – Default*.



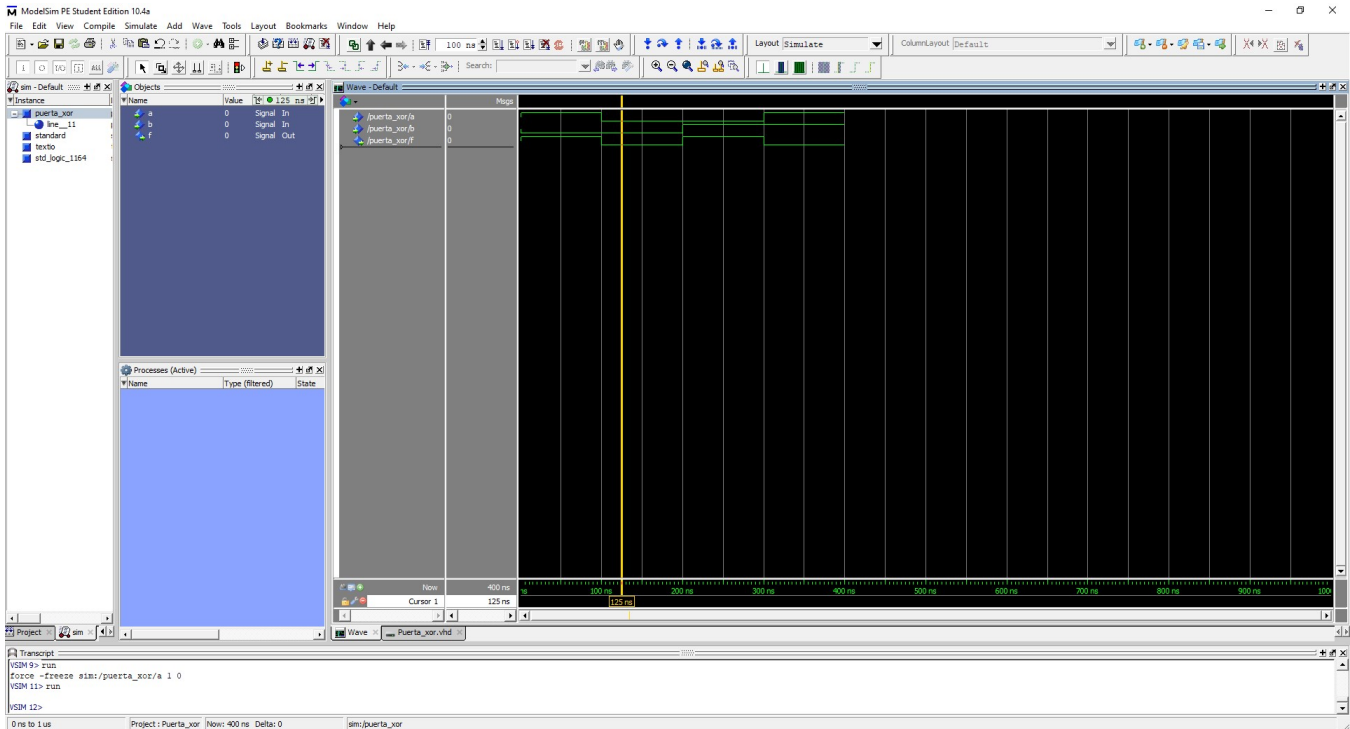
Lo siguiente que hay que hacer es definir unos valores de las señales de entrada del circuito descrito durante un intervalo de tiempo dado. Cuando se realice la simulación, el programa representará el valor de la salida (o salidas) que, de acuerdo con el código vhdl escrito, corresponden a los valores de la entrada (o entradas) indicados.

Para indicar un valor de una entrada hay que hacer clic con el botón derecho del ratón en el nombre de dicha entrada en la ventana *Wave – Default*. En el desplegable que se abre hay que hacer clic en *Force*. En la ventana que se abre hay que indicar el valor deseado de la entrada (0 ó 1) en el campo *Value* y, a continuación, hay que hacer clic en OK. El valor escrito se “aplicará” a la entrada durante el tiempo que se indica en la parte superior de la pantalla (100 nseg... es un buen valor... no lo cambies).

El procedimiento anterior hay que repetirlo con las demás entradas. Una vez indicados los valores de todas las entradas sólo queda iniciar la simulación. Para ello hay que hacer clic en *Simulate – Run – Run 100*. En la ventana situada en la parte derecha de la pantalla aparecerán dibujadas unas señales binarias que representan los valores que se han indicado para las señales de entrada así como las señales correspondientes a los terminales de salida durante 100nseg.

Si se indica un nuevo valor de la entrada (o entradas) de la misma forma que se hizo antes y se vuelve a hacer clic en *Simulate – Run – Run 100*, en la parte derecha de la pantalla aparecerán dibujadas unas señales binarias representando los nuevos valores especificados para la entrada o entradas así como el correspondiente valor de la salida (o salidas) del circuito descrito en vhdl.

El proceso anterior se puede repetir cuantas veces de quiera hasta comprobar el valor (o valores) que proporciona el circuito descrito en vhdl para todas las combinaciones de entrada que se desee.



Nota: la línea amarilla vertical que aparece en pantalla de fondo negro cuando se hace clic con el ratón en la misma la podemos desplazar horizontalmente con el ratón. En la columna Msgs que hay a la izquierda de la ventana en la que se representan los valores de las señales se indican los valores de las señales de entrada y de salida correspondientes al instante de tiempo en el que se encuentra la línea amarilla.

Nota: para salir del simulador, en *Simulate* hay que elegir *End Simulation*.

Nota: en youtube hay un video que describe los pasos anteriores (algunos pasos los indican de forma diferente a la descrita en los párrafos anteriores pero también son válidos). La página en la que se encuentra dicho video es:

https://www.youtube.com/watch?v=CAwpc_tGWH0

Notas:

1: Espero publicar el enunciado de la tarea 8 a finales de la semana que viene. Como muy tarde se publicará el lunes día 20 de abril. Teóricamente tenéis de tiempo para hacer las tareas de HAE hasta el viernes, día 15 de mayo.

2: Como ya he dicho varias veces por escrito, la materia que entrará en el examen de HAE será la siguiente:

- _ La materia vista en las clases de teoría que se han dado este curso.
- _ La materia vista en las prácticas de laboratorio que se han realizado este curso.
- _ La materia vista en las tareas propuestas a lo largo de este curso y que serán un total de 8 tareas.

3: Sigo sin saber cómo y cuándo se realizará el examen de HAE y como consecuencia de ello aún no puedo decir cómo y cuándo deberéis entregarme las tareas.