IE-0523 Circuitos Digitales II

I Ciclo 2017

Profe. Enrique Coen

Lennon Núñez Meoño, B34943

Tarea #1

**Contabilización del Tiempo**

Buscar información 20 min

Estudiar información 20 min (“on the go”)

Ejecución 2 h 30 min

Descarga e instalación de las herramientas 30 min

Solucionar problemas para poder usar las herramientas 1h 20 min

Leer documentación para realizar los ejemplos rápidos 30 min

Tomar capturas de pantalla 10 min

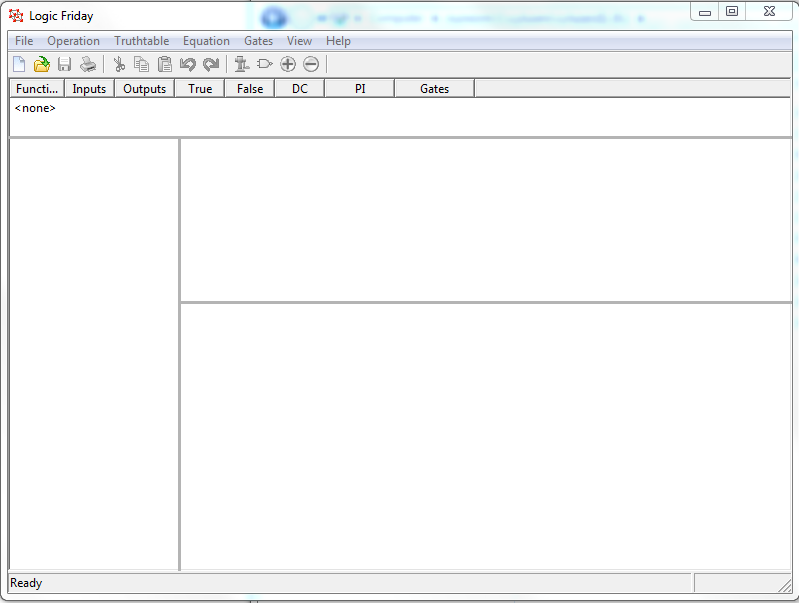
Confección del Reporte y Presentación 30 min (“on the go”)

**Descripción de las herramientas**

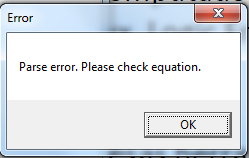
Logic Friday

Es una herramienta con la cual se puede trabajar en diseños de circuitos con lógica digital basada en paquetes IC. Los usuarios meta son estudiantes, entusiastas e ingenieros que trabajan con estos modelos de circuitos. Dentro de las funciones específicas de la herramienta se puede ingresar la lógica digital como tabla de verdad, ecuación o diagrama de compuertas lógicas, sobre los cuales se trabaja el diseño de la aplicación buscando minimizar el número de compuertas, comparar funciones lógicas, entre otros.

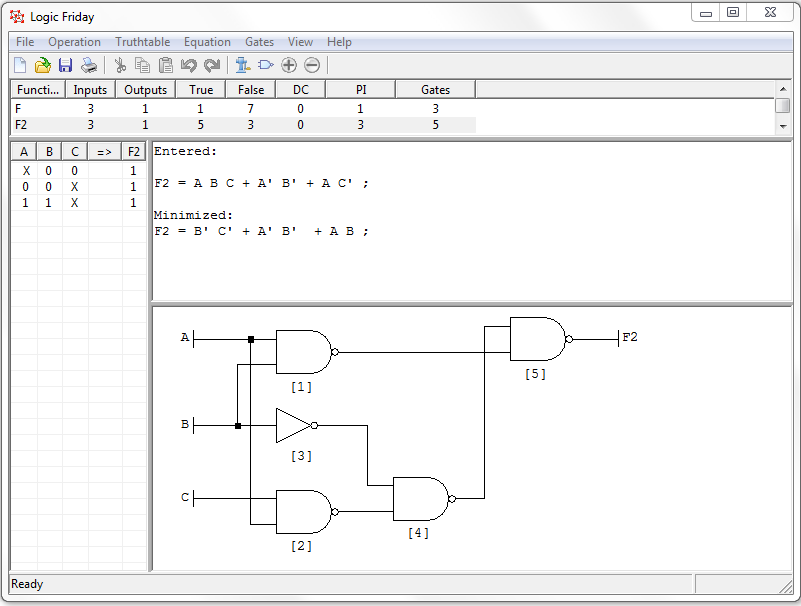
Al abrir, la herramienta despliega una pantalla como la siguiente:



Esta pantalla será el ambiente de trabajo. Al seleccionar la opción “New”, se puede crear una nueva función lógica ya sea como Tabla de Verdad, Ecuación Lógica o Diagrama de Compuertas. Al iniciar con ecuación lógica, se generará automáticamente la tabla de verdad, y visceversa. Para este ejemplo se inició con la Ecuación Lógica: ABC+A’B’+AC’, creada de manera arbitraria. En este punto surgen dos aclaraciones importantes para ingresar la ecuación, primero, esta debe terminar con un “;”, de lo contrario la rechaza y muestra una ventana con el aviso, segundo, para definir que es una función se debe iniciar con “F =” seguido de la función, de lo contrario también indica que hay un error. Al presionar ENTER en el teclado se generará la función lógica como tabla y diagrama.



Cada vez que se ingrese una nueva función con NEW se agregarán a la lista de funciones. Para generar el diagrama de compuertas se debe Mapear la función a compuertas con la opción “Map to Gates…”, con la función a mapear previamente seleccionada. La herramienta sugerirá compuertas para generar la función y si se quiere generar de manera rápida o exacta.

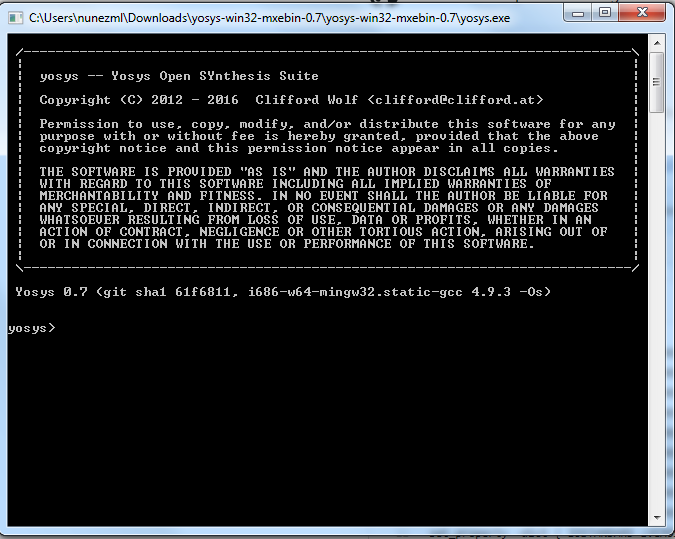


Ya con algún diseño para trabajar se puede aprovechar el resto de funciones que ofrece la herramienta.

Yosys

Es una estructura de trabajo para la síntesis de Verilog RTL, ofreciendo un set básico de algoritmos de síntesis para varias áreas de aplicación. Yosys es controlado utilizando scripts de síntesis. Para realizar la síntesis es necesario tener de previo un documento en Verilog para sintetizar.

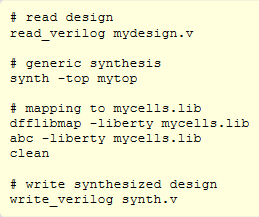
Al ejecutar Yosys se abrirá una ventana como la siguiente, para interactuar con los comandos específicos de Yosys.



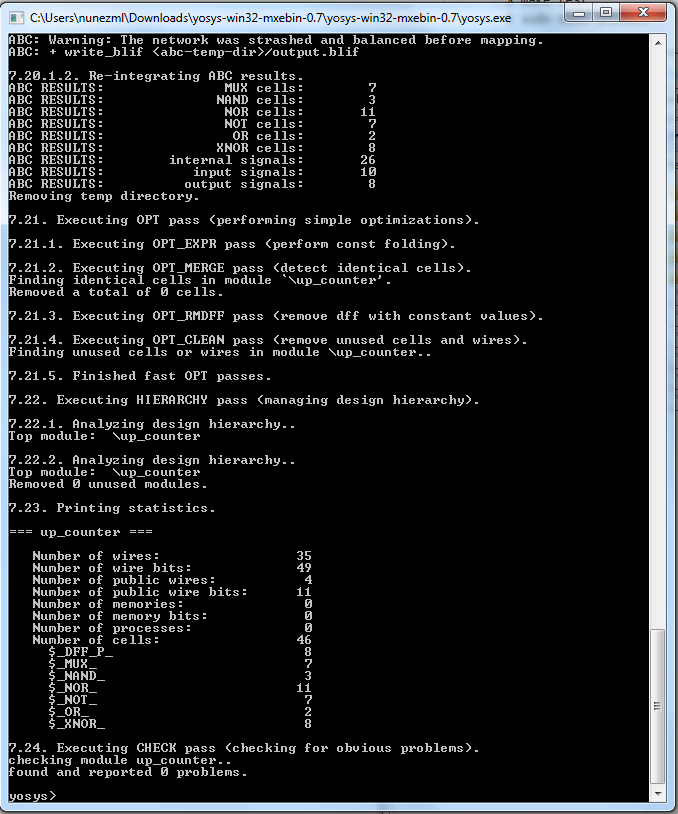
Por ejemplo, si se ingresa el comando “help” se despliega en pantalla la lista de instrucciones. Dentro de las funciones se encuentran las de leer un archivo en Verilog, elaborar la jerarquía de diseño y escribir el diseño sintetizado.



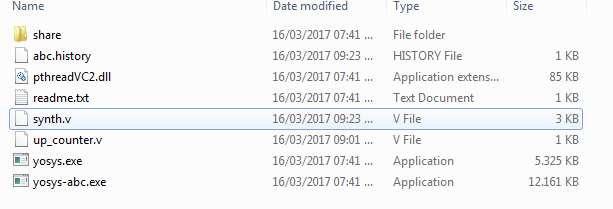
La página oficial sugiere un script modelo como base para síntesis sencillas.



Para realizar la prueba de síntesis de utilizó el código up\_counter.v, proporcionado por ASIC WORLD ([www.asic-world.com](http://www.asic-world.com)). Para no hacer un script, cada línea se corrió en la terminal de Yosys paso a paso. El siguiente es el resultado de pasar por “read\_verilog path\up\_counter.v” y “synth –top up\_counter”. Con esto se lee el diseño en verilog y se genera la síntesis del diseño.



Como Yosys no ha sido configurado, para abrir documentos trabaja con direcciones absolutas, de modo que para utilizar las cells.lib del script modelo la dirección queda muy larga. Luego de ello, con el comando “write\_verilog synth.v” se genera el archivo en Verilog con el sintetizado del diseño inicial.



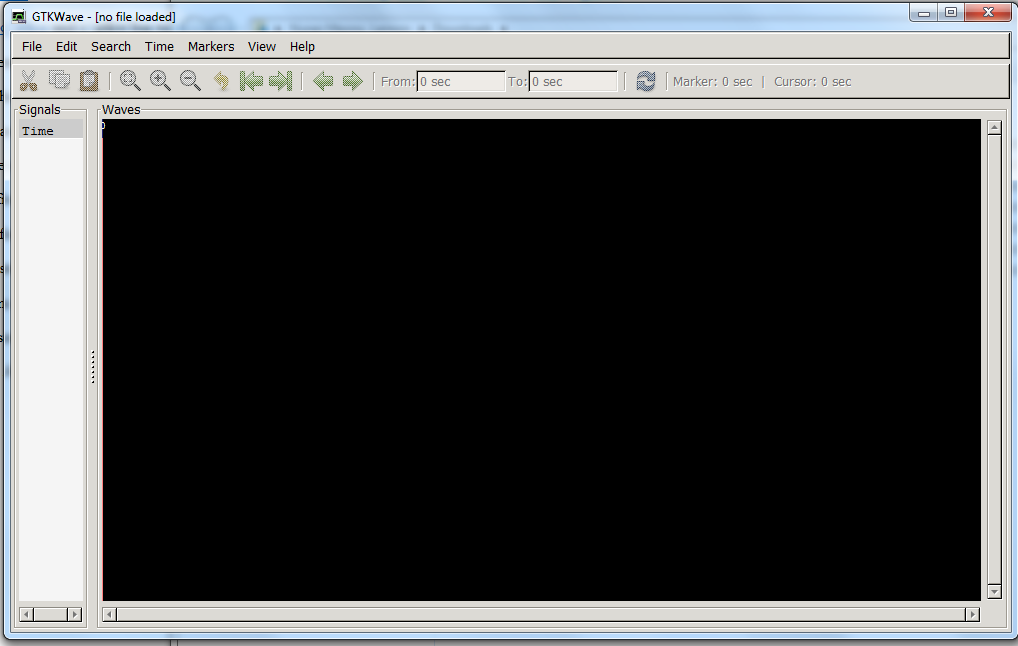
Para comprobar el sintetizado, se requiere de una herramienta que permita correr los archivos en Verilog, como Icarus. Con otras funciones de Yosys se podría optimizar, etc, pero para ellos se requiere realmente saber lo que se está haciendo con la herramienta.

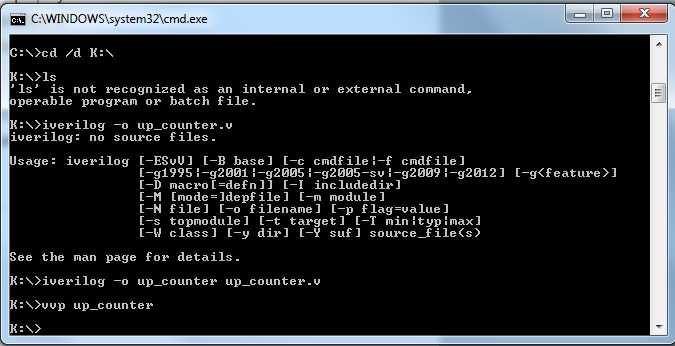
Icarus Verilog con GTkWave

Icarus Verilog es una herramienta para simulación y sintetizado en Verilog. Funciona como un compilador para código escrito en Verilog según los estándares de la IEEE.

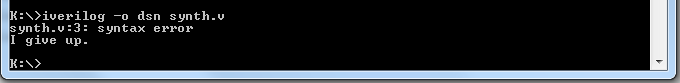
GTkWave es una herramienta para ver señales, basado en GTK+. Por medio de este ambiente desarrollo es posible observar las señales generadas por la simulación de los circuitos digitales descritos en Verilog, y con ella analizar sus comportamientos en el tiempo.

La terminal de GTkWave es la siguiente.

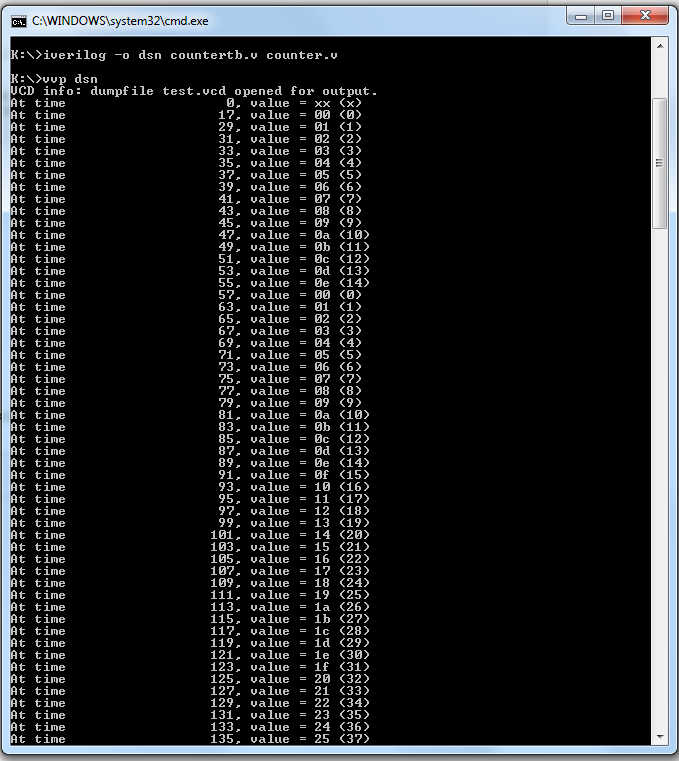


En Windows, es necesario utilizar la Command Promt, cmd, cambiar al path donde esté el archivo en Verilog, y correr los comandos de Icarus Verilog (iverilog). El primer comando es “iverilog –o up\_counter up\_counter.v”, esto compilará el archivo up\_counter.v. Para ejecutar el compilado, se usa el comando “vvp”.

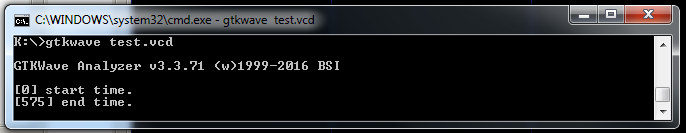
Para observar el resultado en GTkWave, se requiere generar un archivo .vcd. Se ocupan las líneas de código que generarán el archivo .vcd dentro del test bench. Al intentar modifacr el synth.v, ssale el siguiente error.



Se requiere un verdadero test bench para el asunto. Se procede a usar un ejemplo del wikia de iverilog (iverilog.wikia.com).



Una vez que se tiene el archivo .vcd se procede a usar GTkWave para observar las señales.



Para observar las señales se debe seleccionar del árbol de jerarquí y arrastar la señal a observar de la lista generada a la pantalla negra, o bien con doble click. Para este ejemplo se puede ver el resultado (señal 1) del contador.

