

Pablo Alfaro Castro

B20162

Reporte, Tarea 1 Circuitos Digitales 2

1) Tiempo tomado

El tiempo que tome para realizar la tarea consiste en;

1h en búsqueda de información

1h instalando los programas

2) Herramientas utilizadas

Las herramientas utilizadas en el curso fueron instaladas en el sistema operativo Windows.

3) Instalación

Para realizar la instalación de icarus verilog con GTKwave, obtuve un archivo ejecutable desde la página de descarga del programa. Simplemente se ejecutó el .exe y la única complicación que se obtuvo fue agregando la ruta de los binarios para que el cmd de Windows lo pudiera utilizar.

Para la instalación de Yosys se obtuvo un archivo.rar el cual al descomprimirse suministro la herramienta dicha, ya que solo hay que agregar el archivo de binarios al path de Windows.

4) Ejecución

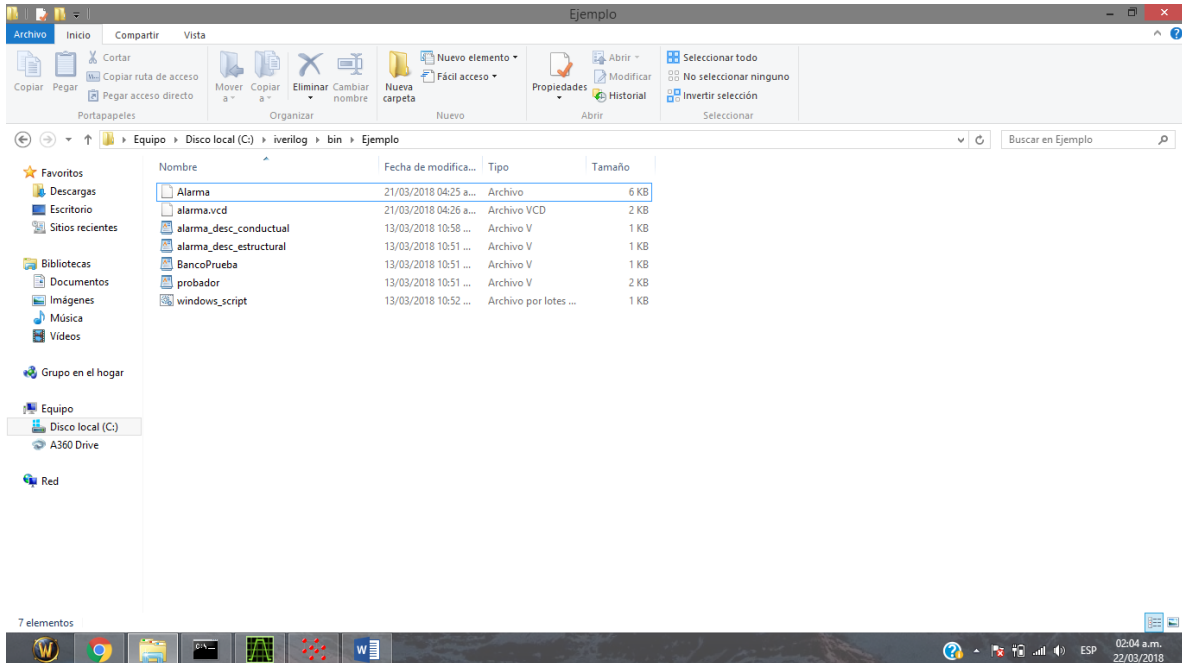
Primeramente, el icarus verilog sirve para poder compilar los códigos programados en el formato.v que representan archivos programados en el lenguaje verilog.

Para ejemplificar, lo que realice fue la compilación y ejecución del programa alarma que se suministro en el curso, donde la compilación se realiza en los siguientes pasos.

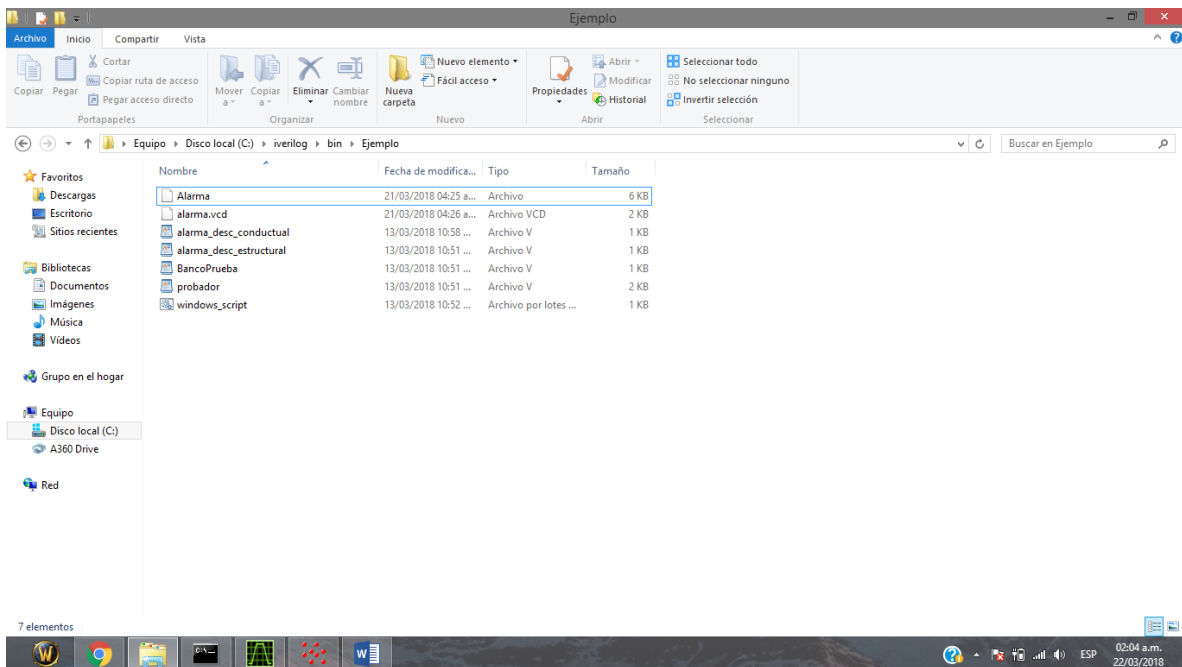
- 1- Abrir la consola de Windows cmd.
- 2- En la carpeta donde se encuentran los archivos, escribir el siguiente comando:

Iverilog –o Alarma alarma_desc_conductual.v alarma_desc_estructural.v
BancoPrueba.v probador.v

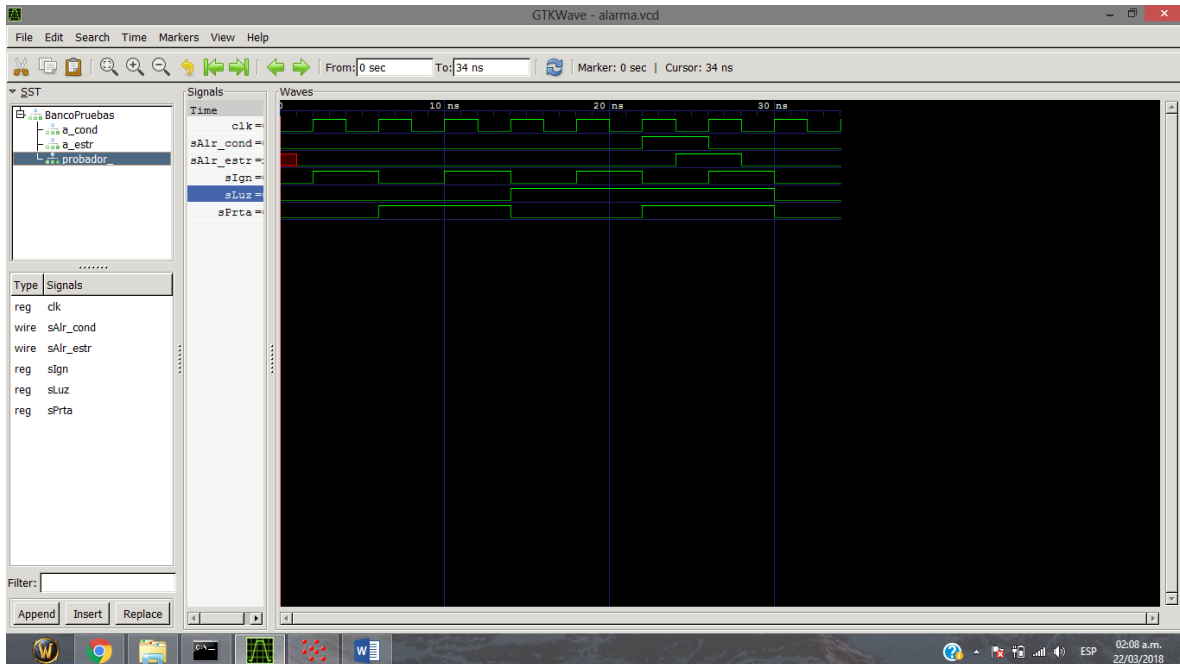
- 3- Al ejecutar este comando se generó un archivo “Alarma” que contiene la compilación de todos los archivos:



- 4- Se utilizo el comando vcd Alarma para generar un archivo que sirve para abrir en el gtkwave el programa compilado:



- 5- Por último se ejecutó el GTKwave con el comando `gtkwave alarma.vcd`, el resultado está visto en la ventana de GTKwave en el cual se pueden ver las señales que se programaron en el ejemplo de la alarma:



Para la ejecución de una síntesis utilizando Yosys, simplemente se modifica estas líneas de texto:

```
read_verilog Conductual.v
hierarchy -check -top Conductual
proc;
opt;
fsm;
opt;
memory;
opt;
techmap;
opt;
write_verilog rtlil.v
```

```
dfflibmap -liberty cmos_cells.lib;  
abc -liberty cmos_cells.lib;  
clean  
write_verilog sintetizado.v
```

Donde Conductual representa el archivo de verilog que se quería sintetizar.

```
/* Generated by Yosys 0.7 (git sha1 61f6811, i686-w64-mingw32.static-gcc 4.9.3 -Os) */  
  
(* top = 1 *)  
(* src = "Conductual.v:1" *)  
module sintetizado(q, s_out, clk, enb, dir, s_in, mode, d);  
  (* src = "Conductual.v:7" *)  
  wire [3:0] _000_;  
  (* src = "Conductual.v:7" *)  
  wire _001_;  
  wire _002_;  
  wire _003_;  
  wire _004_;  
  wire _005_;  
  wire _006_;  
  wire _007_;  
  wire _008_;  
  wire _009_;  
  wire _010_;  
  wire _011_;  
  wire _012_;  
  wire _013_;  
  wire _014_;  
  wire _015_;  
  wire _016_;  
  wire _017_;  
  wire _018_;  
  wire _019_;  
  wire _020_;  
  wire _021_;  
  wire _022_;
```

En la imagen se muestra el archivo sintetizado.