Universidad del Valle de Guatemala

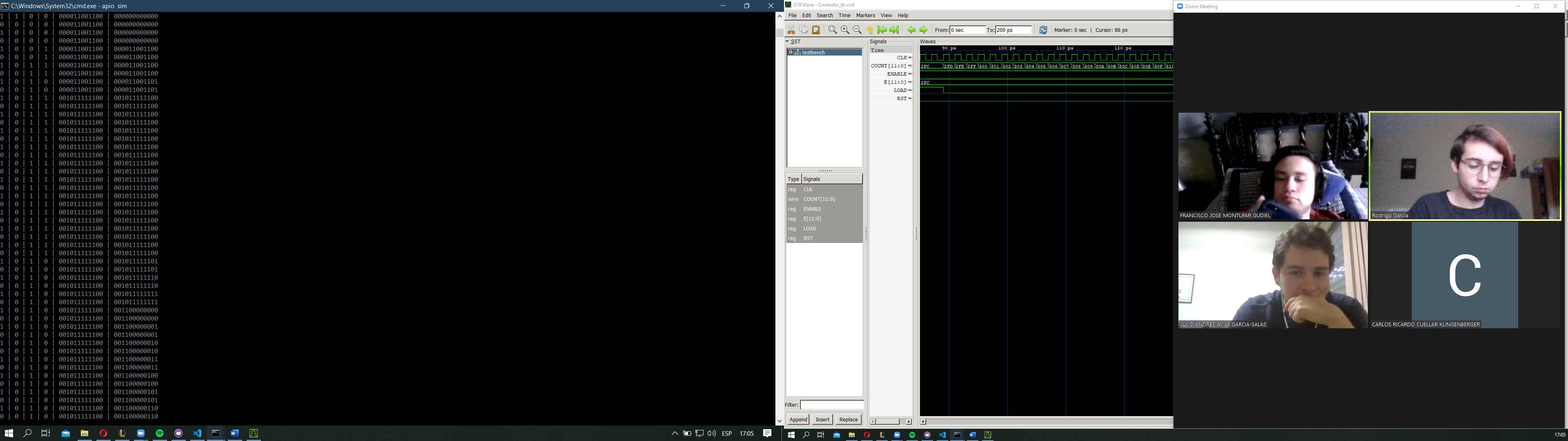
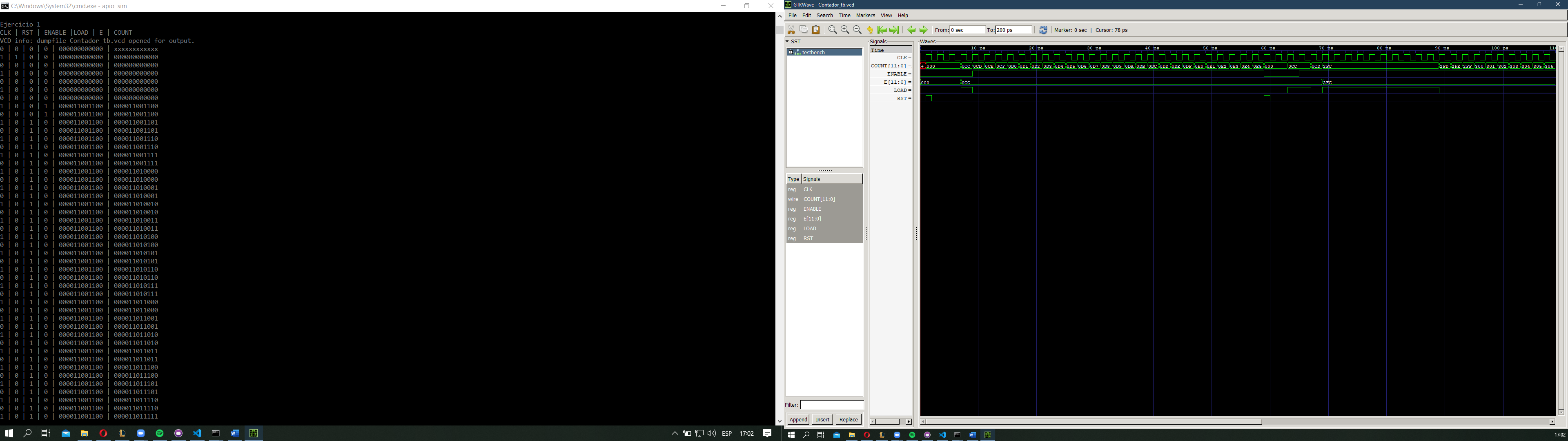
Digital 1

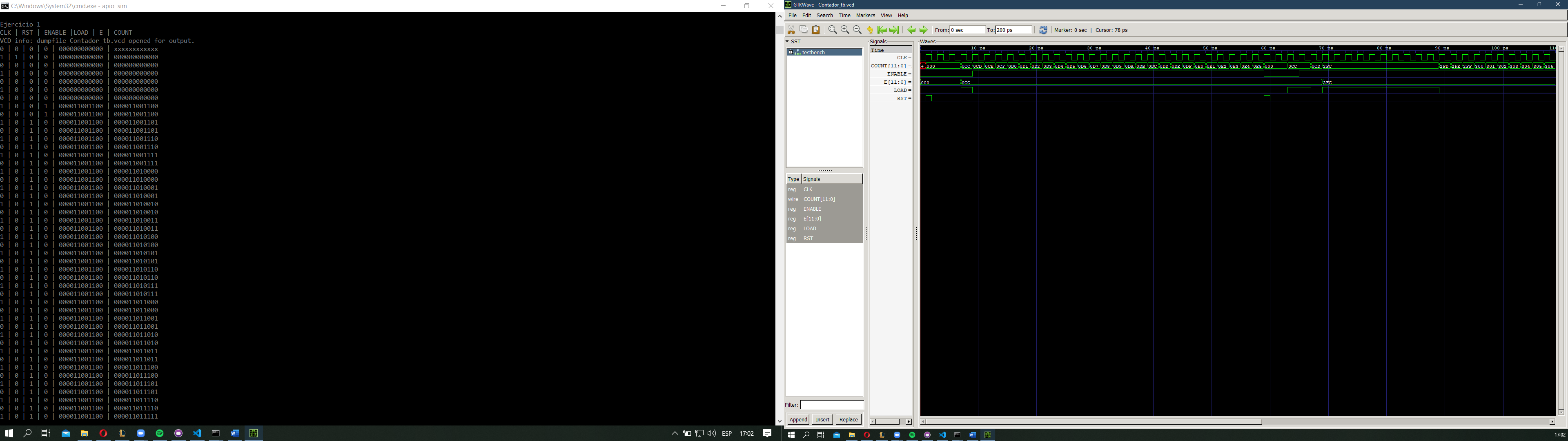
Rodrigo García 19085

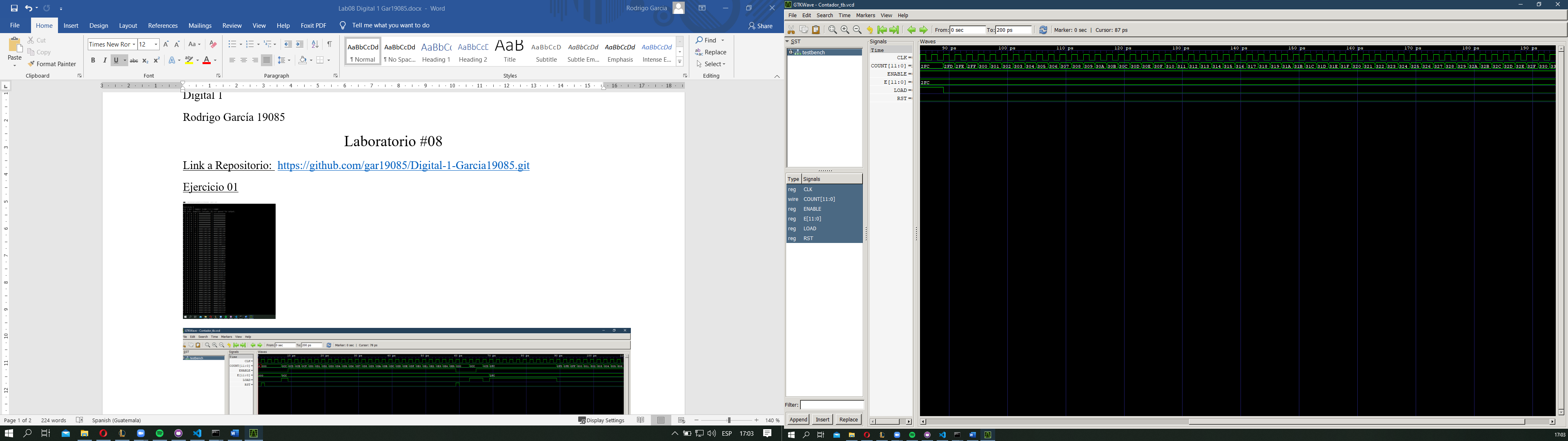
Laboratorio #08

Link a Repositorio: <https://github.com/gar19085/Digital-1-Garcia19085.git>

Ejercicio 01

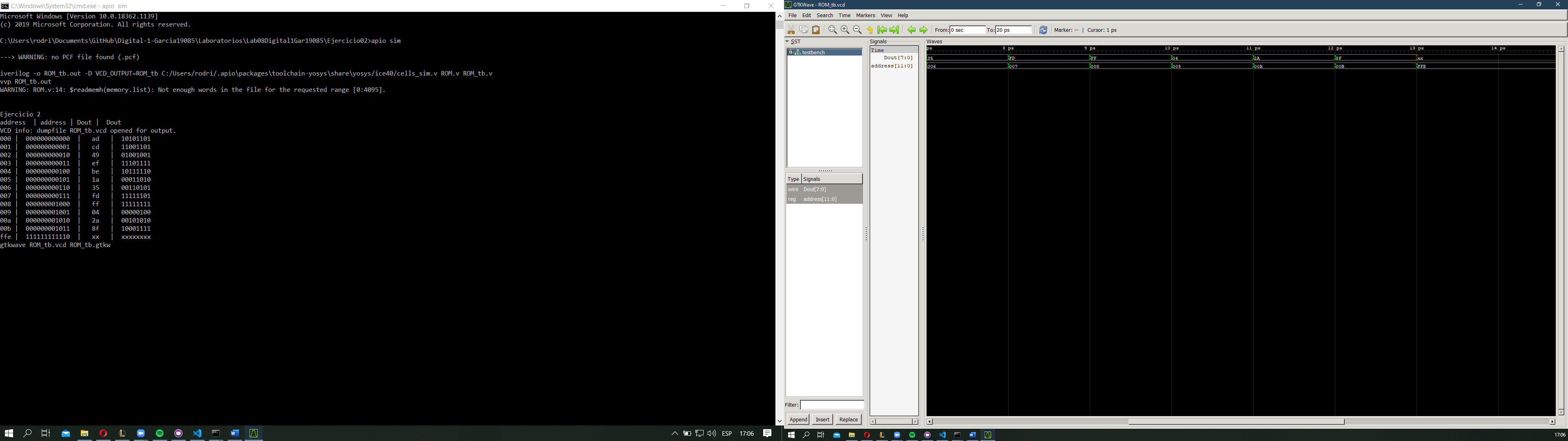


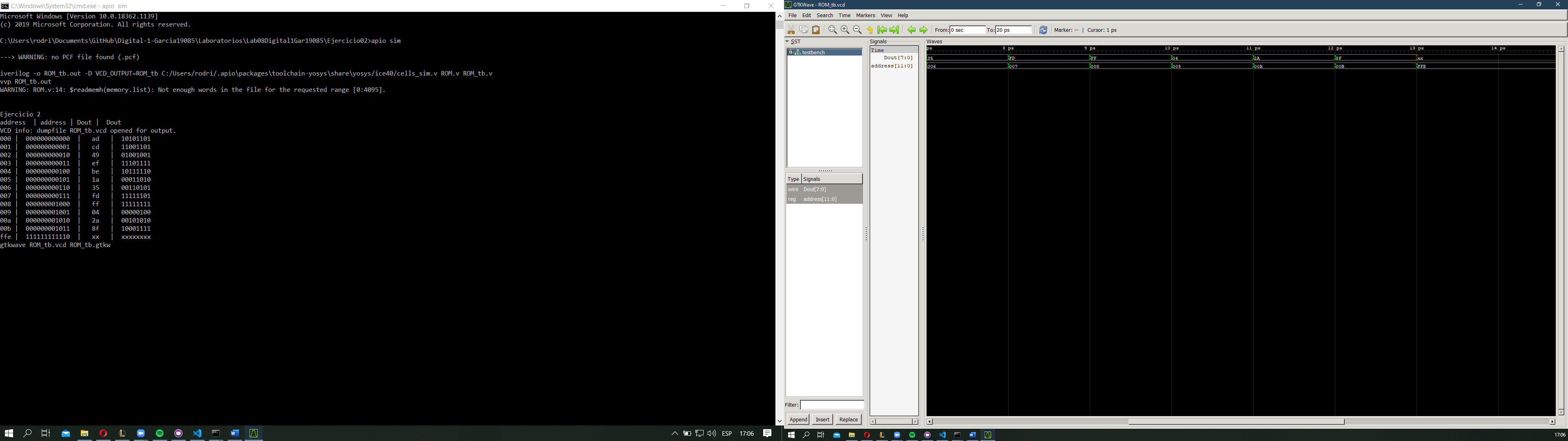




Mi contador consistió en crear un módulo llamado “Counter” el cual consiste con una serie de comandos que dependen de ciertos inputs para que funcionen. Primero es el load que el cual si este se activa pasara la información del input al contador. Luego el Enable cuando este se active comenzara a contar si el Load no esta activado, si esta activado este mantendrá el valor del load, y por último el reset que cuando este se active limpia el contador.

Ejercicio 02





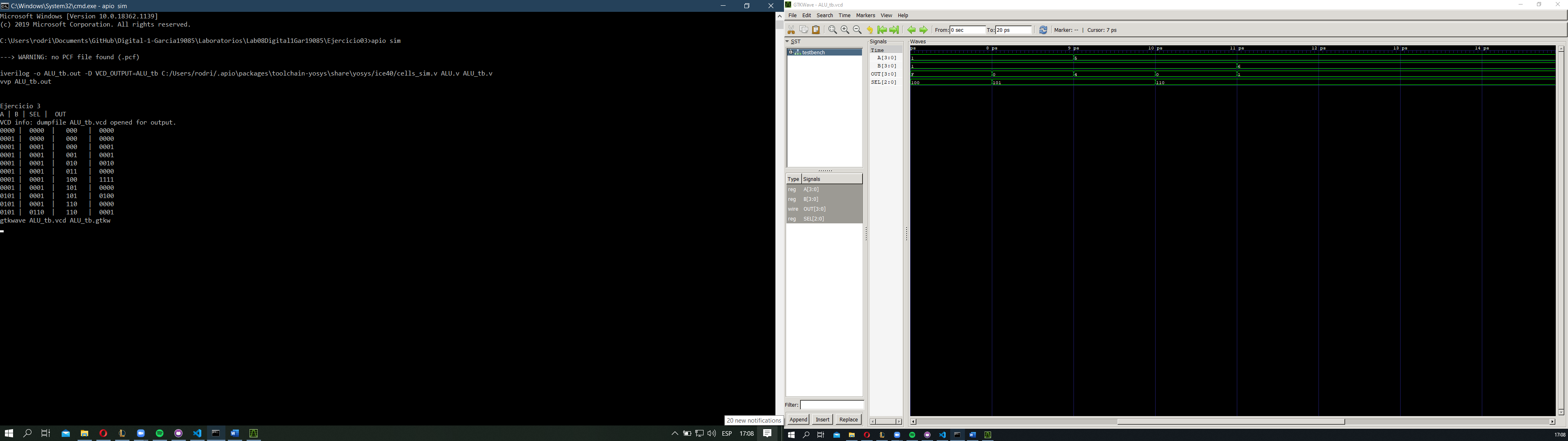
Mi ROM de 4k x 8 consistió en crear un modulo llamado “ROM4Kx8” el cual consiste en un input (address) y un output (Dout), los cuales son necesarios para poder leer la información almacenada en el archivo memory.list. Para indicar el tamaño de la ROM indique las columnas y filas con el siguiente código: “reg [7:0] memory[0:4095]” la cuál corresponde a 8 columnas y 4096 filas. El address llama el dato guardado en el archivo, y el Dout lo muestra.

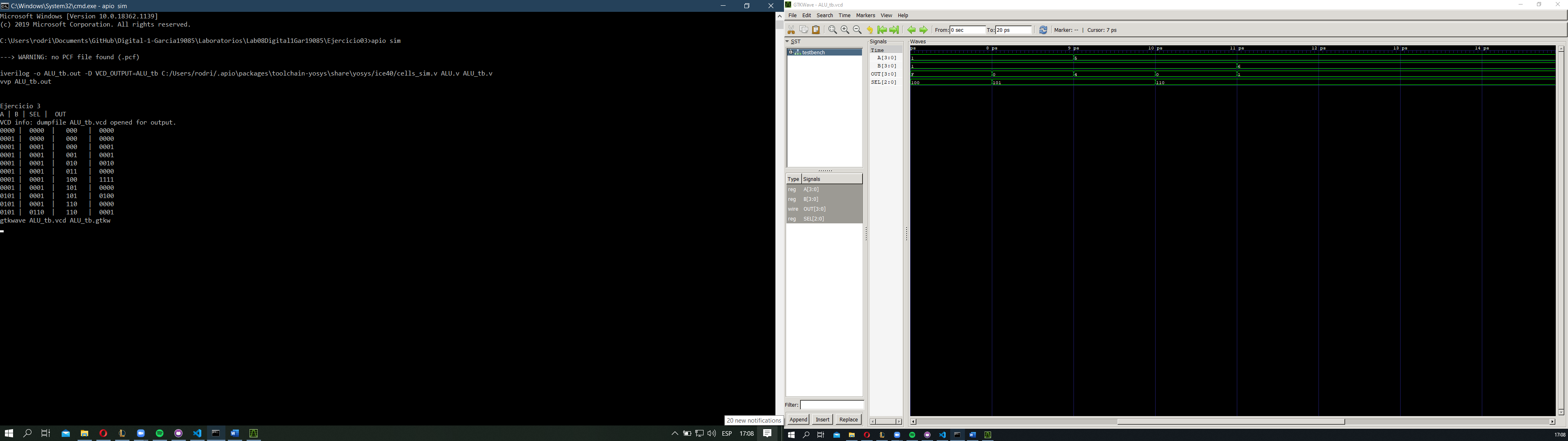
Array: Para implementar un array de datos se le da un nombre el cual va a tener declarado el mínimo y máximo índice, para establecer los límites.

$readmemh: Funciona para leer datos en hexadecimal.

$readmemb: Funciona para leer datos en binario.

Ejercicio 03





La implementación de la ALU consistió en crear un modulo llamado “ALU” en el cuál se indicaron las entradas y la salida de 4 bits, y el selector de 3 bits por el tamaño de operaciones que utiliza la ALU. El funcionamiento del mismo depende de la configuración “case()” cuyo funcionamiento es realizar una tabla en la que se le asigna un espacio en el selector a cada operación de la ALU para qué cuando se llame a dicha posición del selector realice la operación almacenada ahí.