Universidad del Valle de Guatemala

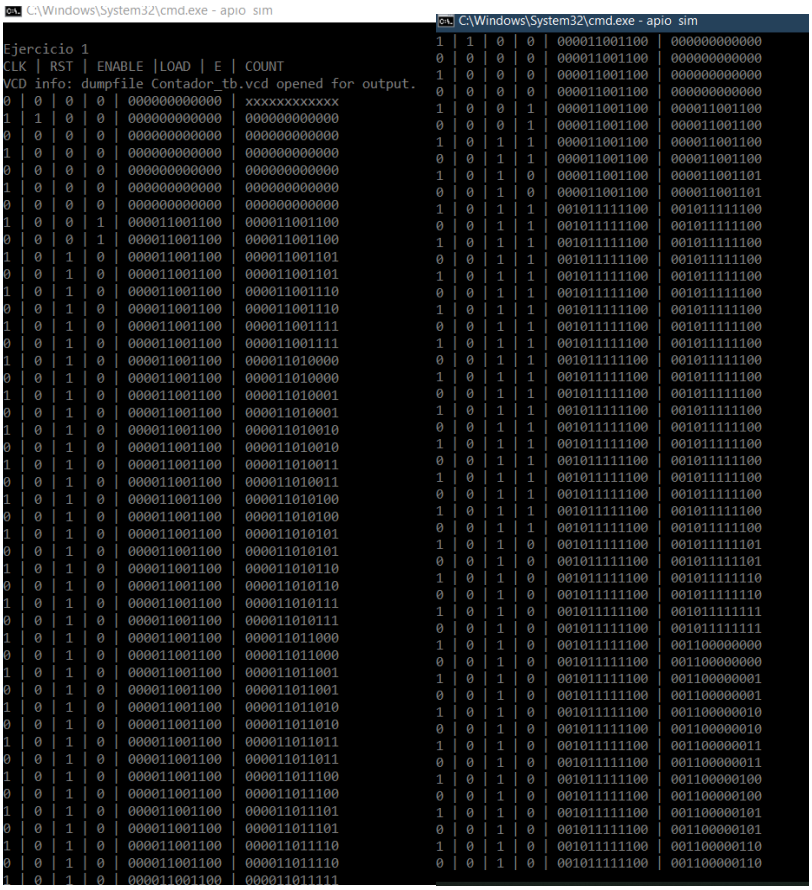
Digital 1

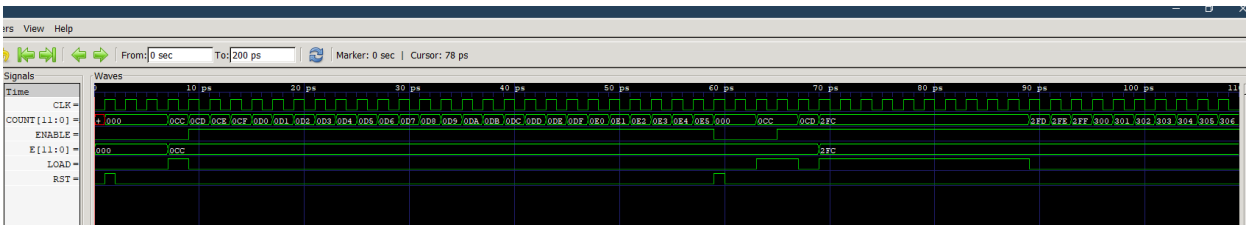
Rodrigo García 19085

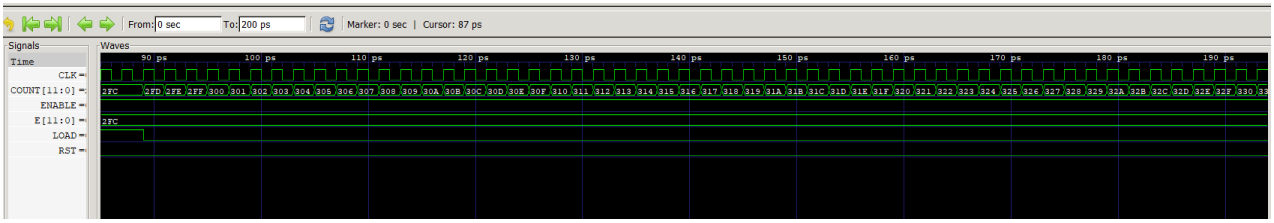
Laboratorio #08

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

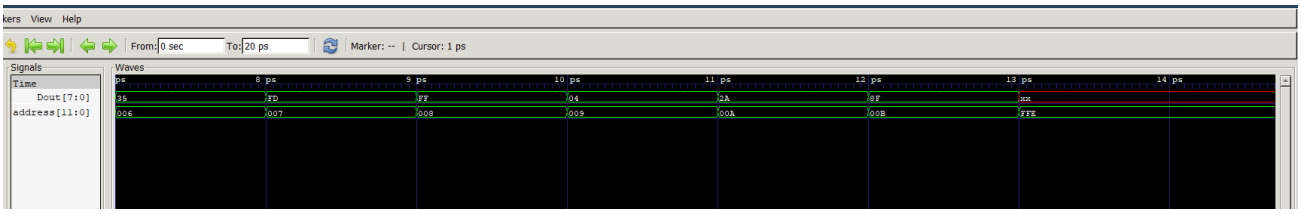
```
Microsoft Windows [Version 10.0.18362.1139]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\rodri\Documents\GitHub\Digital-1-Garcia19085\Laboratorios\Lab08Digital1Gar19085\Ejercicio02>apio sim

---> WARNING: no PCF file found (.pcf)

iverilog -o ROM_tb.out -D VCD_OUTPUT=ROM_tb C:/Users/rodri/.apio\packages\toolchain-yosys\share\yosys/ice40/cells_sim.v ROM.v ROM_tb.v
vvp ROM_tb.out
WARNING: ROM.v:14: $readmemh(memory.list): Not enough words in the file for the requested range [0:4095].


Ejercicio 2
address  | address | Dout |  Dout
VCD info: dumpfile ROM_tb.vcd opened for output.
000 |  000000000000  |   ad   |  10101101
001 |  000000000001  |   cd   |  11001101
002 |  000000000010  |   49   |  01001001
003 |  000000000011  |   ef   |  11101111
004 |  000000000100  |   be   |  10111110
005 |  000000000101  |   1a   |  00011010
006 |  000000000110  |   35   |  00110101
007 |  000000000111  |   fd   |  11111101
008 |  000000001000  |   ff   |  11111111
009 |  000000001001  |   04   |  00000100
00a |  000000001010  |   2a   |  00101010
00b |  000000001011  |   8f   |  10001111
ffe |  111111111110  |   xx   |  xxxxxxxx
gtkwave ROM_tb.vcd ROM_tb.gtkw
```



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

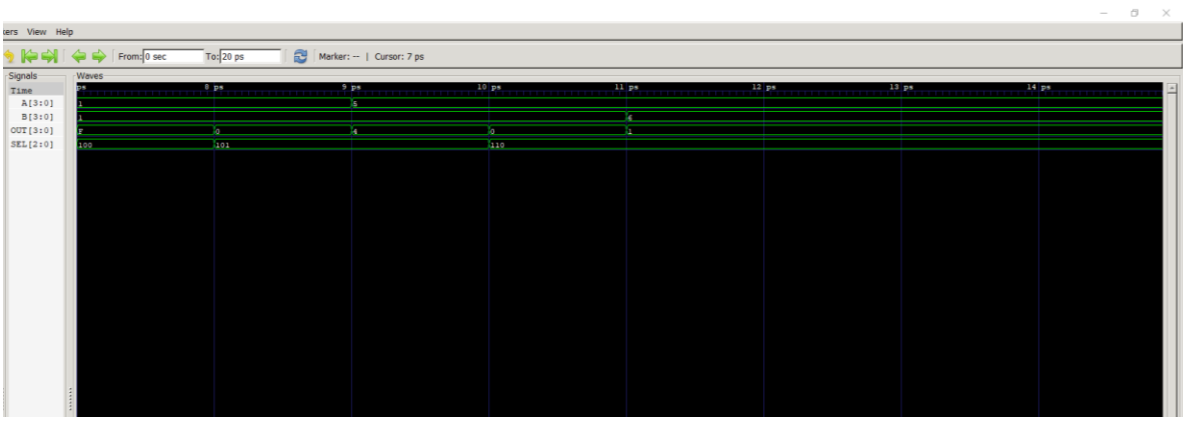
```
Microsoft Windows [Version 10.0.18362.1139]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\rodri\Documents\GitHub\Digital-1-Garcia19085\Laboratorios\Lab08Digital1Gar19085\Ejercicio03>apio sim

---> WARNING: no PCF file found (.pcf)

iverilog -o ALU_tb.out -D VCD_OUTPUT=ALU_tb C:/Users/rodri/.apio\packages\toolchain-yosys\share\yosys/ice40/cells_sim.v ALU.v ALU_tb.v
vvp ALU_tb.out


Ejercicio 3
A | B | SEL |  OUT
VCD info: dumpfile ALU_tb.vcd opened for output.
0000 |  0000  |   000   |  0000
0001 |  0000  |   000   |  0000
0001 |  0001  |   000   |  0001
0001 |  0001  |   001   |  0001
0001 |  0001  |   010   |  0010
0001 |  0001  |   011   |  0000
0001 |  0001  |   100   |  1111
0001 |  0001  |   101   |  0000
0101 |  0001  |   101   |  0100
0101 |  0001  |   110   |  0000
0101 |  0110  |   110   |  0001
gtkwave ALU_tb.vcd ALU_tb.gtkw
```



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.