

# DOCUMENTACIÓN MÓDULOS LAB ELO212

## LAB 3

### **fibi\_gates.sv:**

Reconocedor fibinarios con puertas lógicas (usando assign), no está en uso.

### **BCD\_to\_7seg.sv:**

convertor de entrada de 4-bit a salida para display 7 segmentos.

**Nombre del módulo:** BCD\_to\_sevenSeg **\*ojo: difiere del nombre del archivo\***

**Entrada:** [3:0] BCD\_in

**Salida:** [6:0] sevenSeg

**Requiere:** N/A

**Diagrama de alto nivel:**



### **fib\_rec.sv:**

Reconocedor de fibinarios, usando always\_comb.

**Nombre del módulo:** fib\_rec

**Entrada:** [3:0] BCD\_in

**Salida:** fib

**Requiere:** N/A

**Diagrama de alto nivel:**



### counter\_4bit.sv:

contador de 4-bits

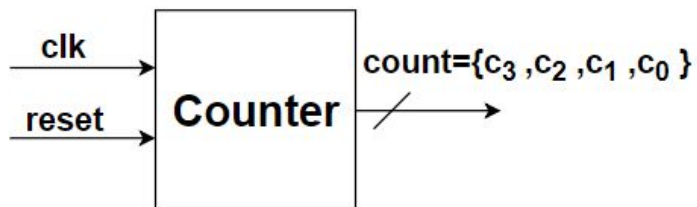
**Nombre del módulo:** counter\_4bit

**Entrada:** clk, reset

**Salida:** [3:0] count

**Requiere:** N/A

**Diagrama de alto nivel:**



### top\_module.sv:

integración de los módulos anteriores.

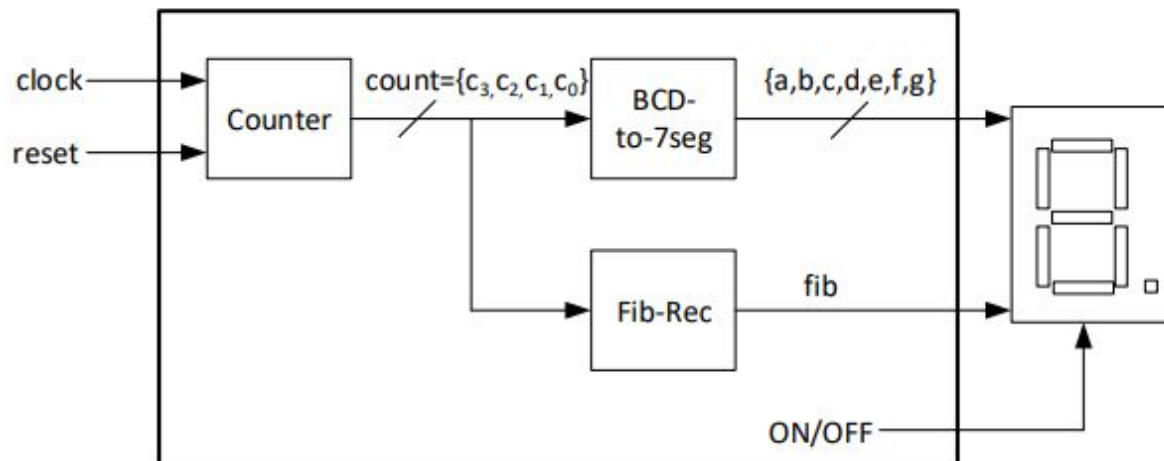
**Nombre del módulo:** top\_module

**Entrada:** clk, reset

**Salida:** [6:0] sevenSeg, fib, onoff

**Requiere:** counter\_4bit.sv, fib\_rec.sv, BCD\_to\_7seg.sv

**Diagrama de alto nivel:**



## LAB 4-5

### 3.1

#### display\_BCD.sv:

recibe una entrada BCD 8421 y onoff desde los switches de la tarjeta de desarrollo, muestra en un display de 7-seg el dígito hexadecimal representado y si es binario (esto no era necesario) enciende el punto decimal. (como usa solo un display fuerza los otros 7 a estar apagados).

las entradas BCD pasan por un flopping stage para que sean sincrónicas.

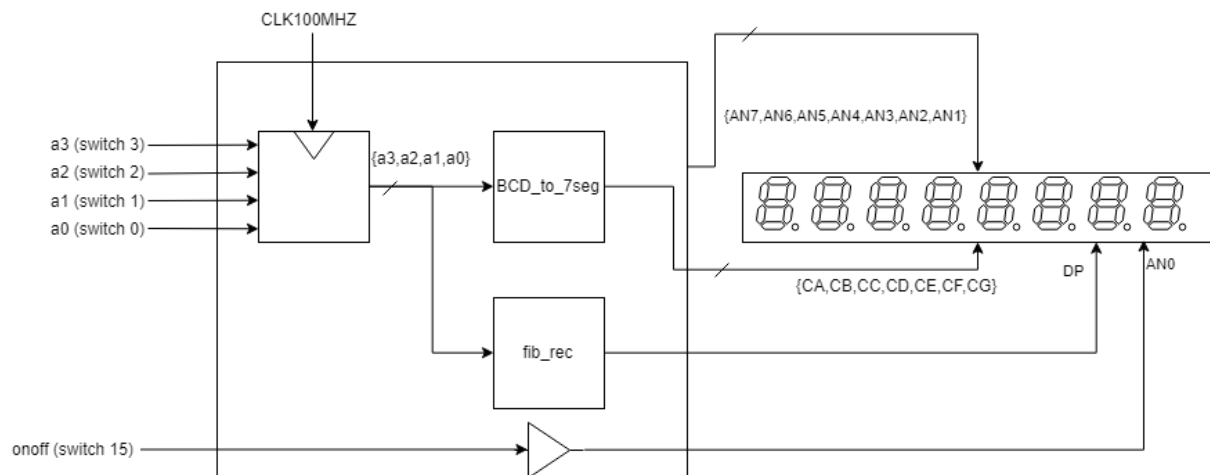
**Nombre del módulo:** display\_BCD

**Entradas:** a3,a2,a1,a0,onoff,clk

**Salidas:** CA,CB,CC,CD,CE,CF,CG,DP,AN0,AN1,AN2,AN3,AN4,AN5,AN6,AN7

**Requiere:** BCD\_to\_7seg.sv, fib\_rec.sv

**Diagrama de alto nivel:**



## 3.2

### top\_divider.sv

Recibe el reloj de 100MHZ y lo divide a frecuencias de 10hz,30hz y 500hz, las cuales se mapean a tres pines del PMOD.

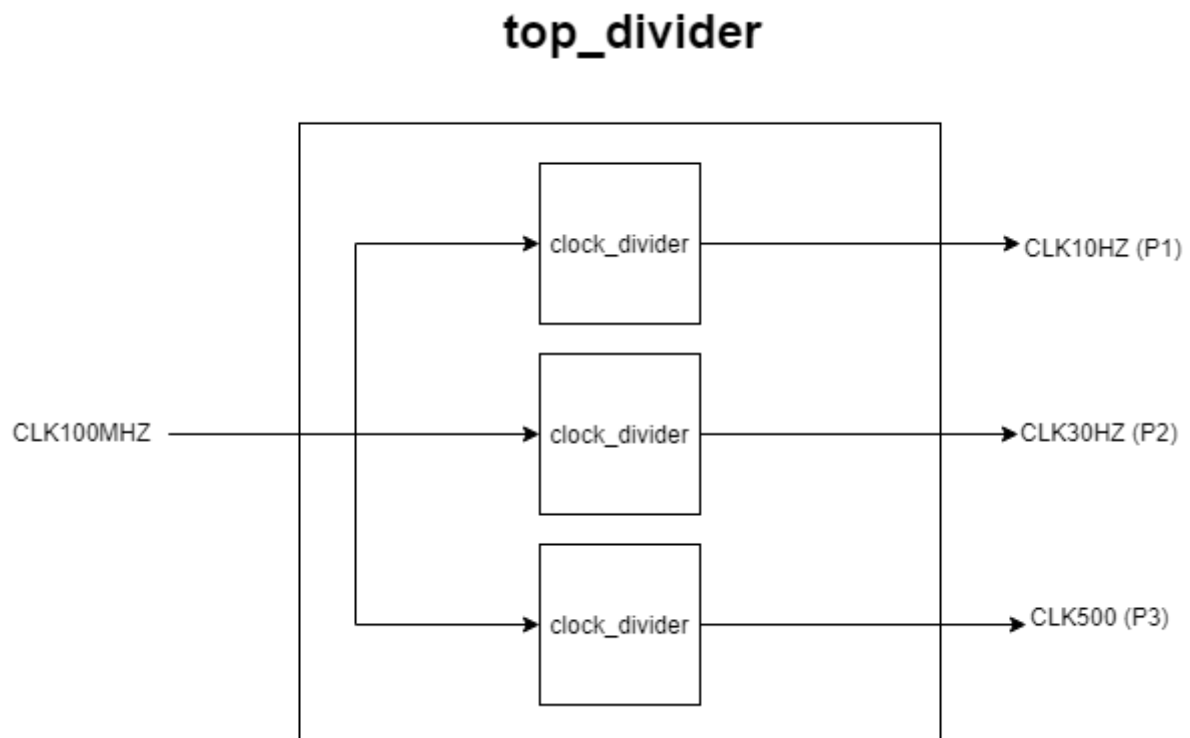
**Nombre del módulo:** top\_divider

**Entradas:** CLK100MHZ

**Salidas:** P1(10hz),P2(30hz),P3(500hz)

**Requiere:** clock\_divider.sv

**Diagrama de alto nivel:**



### 3.3

#### top\_fib\_seq.sv

recibe un reloj de 100Mhz y usa un display para mostrar los valores hexadecimales de un contador de 4 bits y el punto decimal si es fibinario.

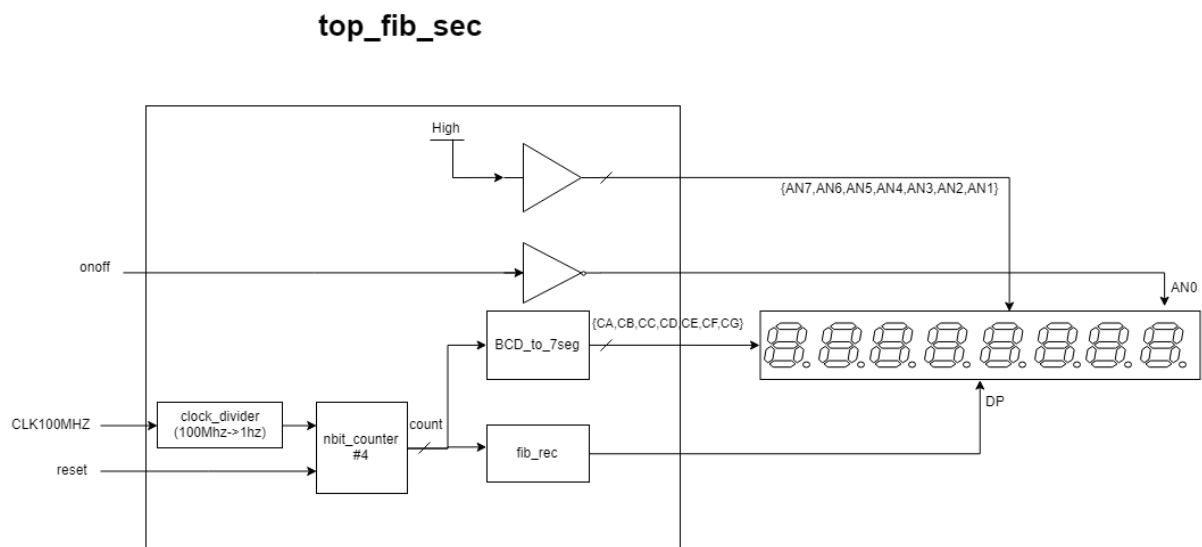
**Nombre del módulo:** top\_fib\_seq

**entradas:** CLK100MHZ,reset,onoff

**Salidas:** CA a CF , AN7 a AN0, DP

**Requiere:** nbit\_counter.sv, clock\_divider.sv, BCD\_to\_7seg.sv, fib\_rec.sv

**Diagrama de alto nivel:**



## 3.4

### ALU.sv

**Nombre del módulo:**

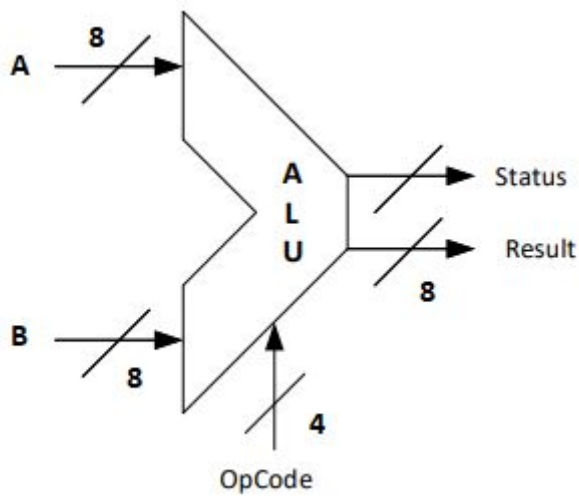
Realiza suma, resta, bitwise OR o bitwise AND si opcode es 1000, 0100, 0010 o 0001 respectivamente y status 1. en todo otro caso la salida es 8'd0 y el status es 0

**entradas:** A,B,opcode

**Salidas:** result,status

**Requiere:** N/A

**Diagrama de alto nivel:**



## digit\_selector.sv

Recibe un contador y que segmentos quiero mostrar en c/u de los 8 displays (CA0-CA7), usa el contador para sincronizar que cátodo mostrar y que ánodo encender.

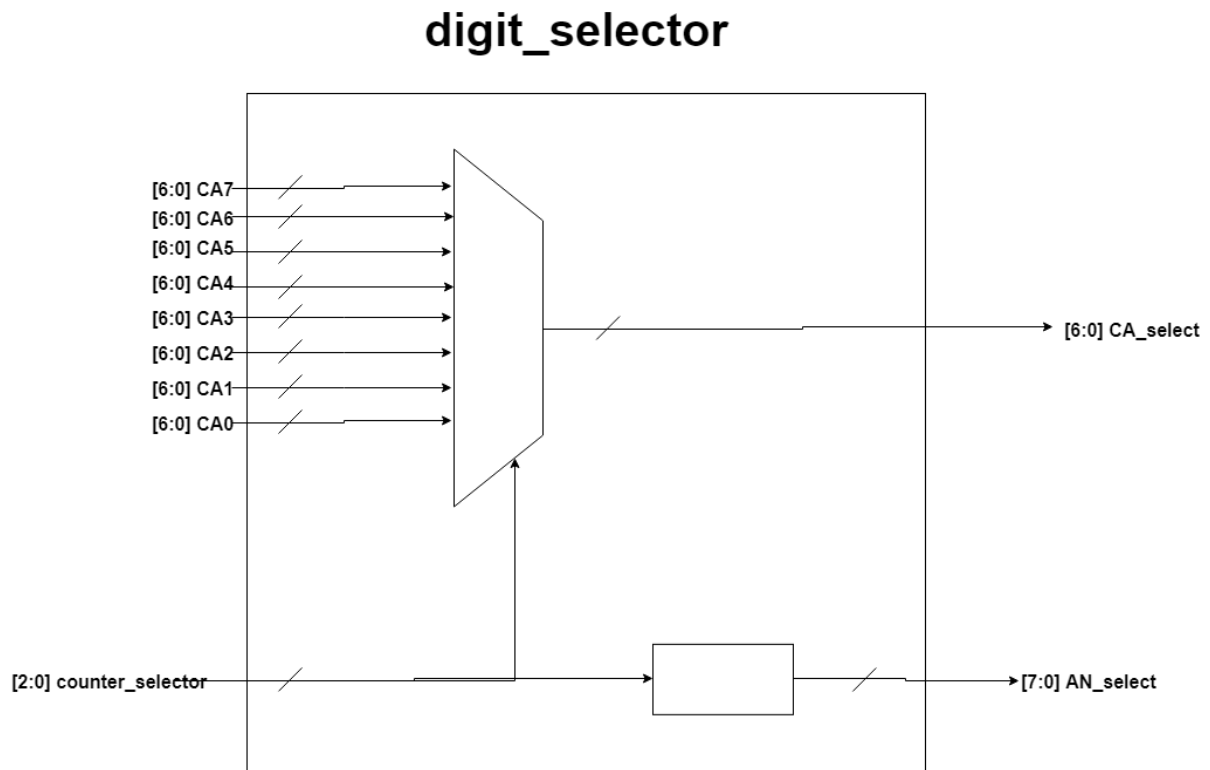
**Nombre del módulo:** digit\_selector

**Entradas:** CA7-CA0, counter\_selector,

**Salidas:** CA\_select, AN\_select

**Requiere:** N/A

**Diagrama de alto nivel:**



## TDM

Incorpora digit\_selector con un divisor de reloj y un contador interno.

**Nombre del módulo:**

**Entradas:** clk\_in, CA0-CA7

**Salidas:** CA-CG, AN7-AN0

**Requiere:** clock\_divider, nbit\_counter, digit\_selector

**Diagrama de alto nivel:**

