

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

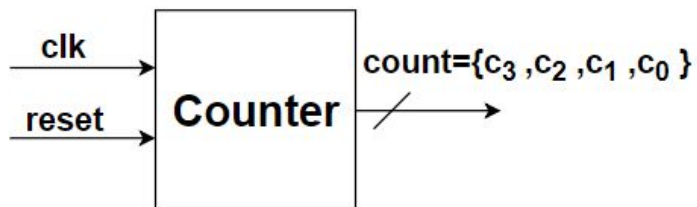
Nombre del módulo: counter_4bit

Entrada: clk, reset

Salida: [3:0] count

Requiere: N/A

Diagrama de alto nivel:



top_module.sv:

integración de los módulos anteriores.

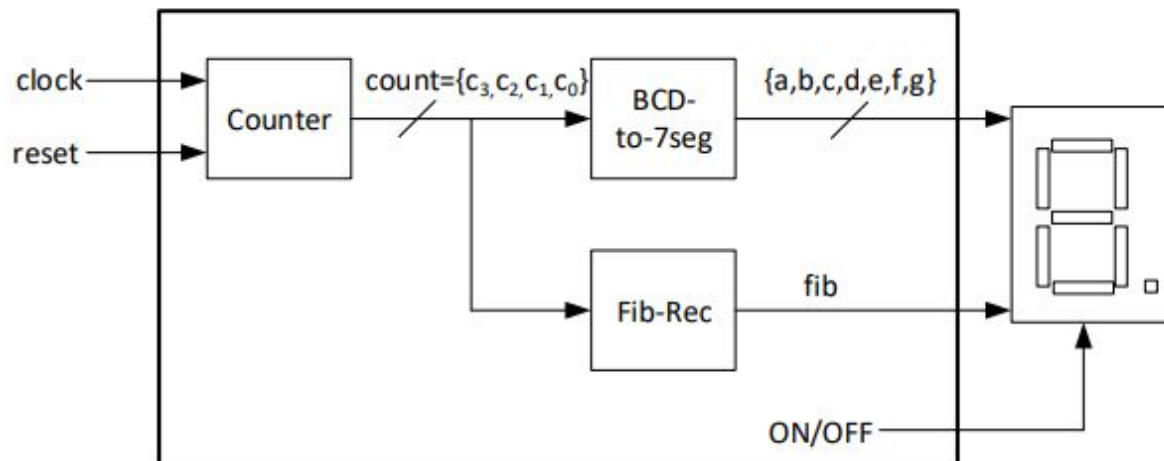
Nombre del módulo: top_module

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.

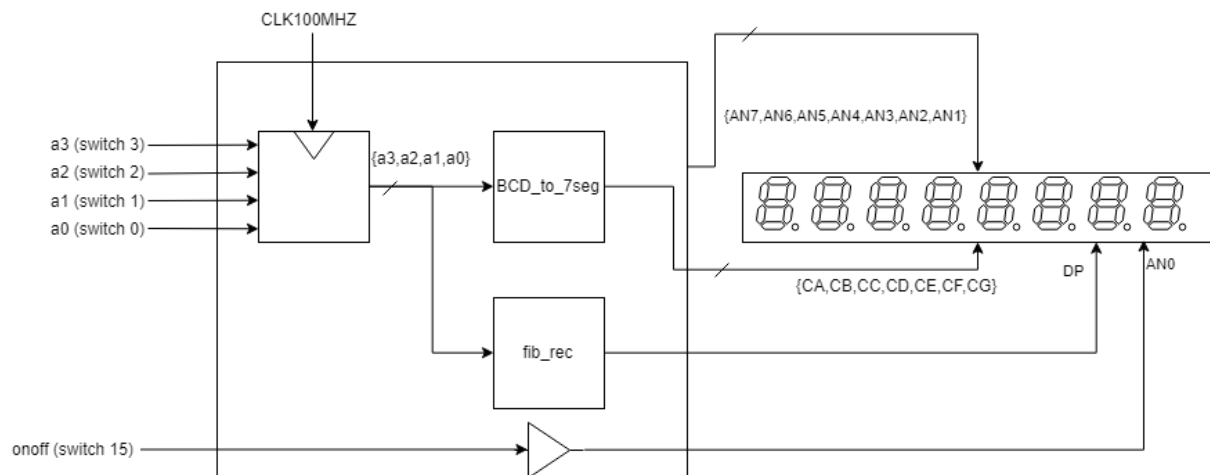
Nombre del módulo: display_BCD

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.

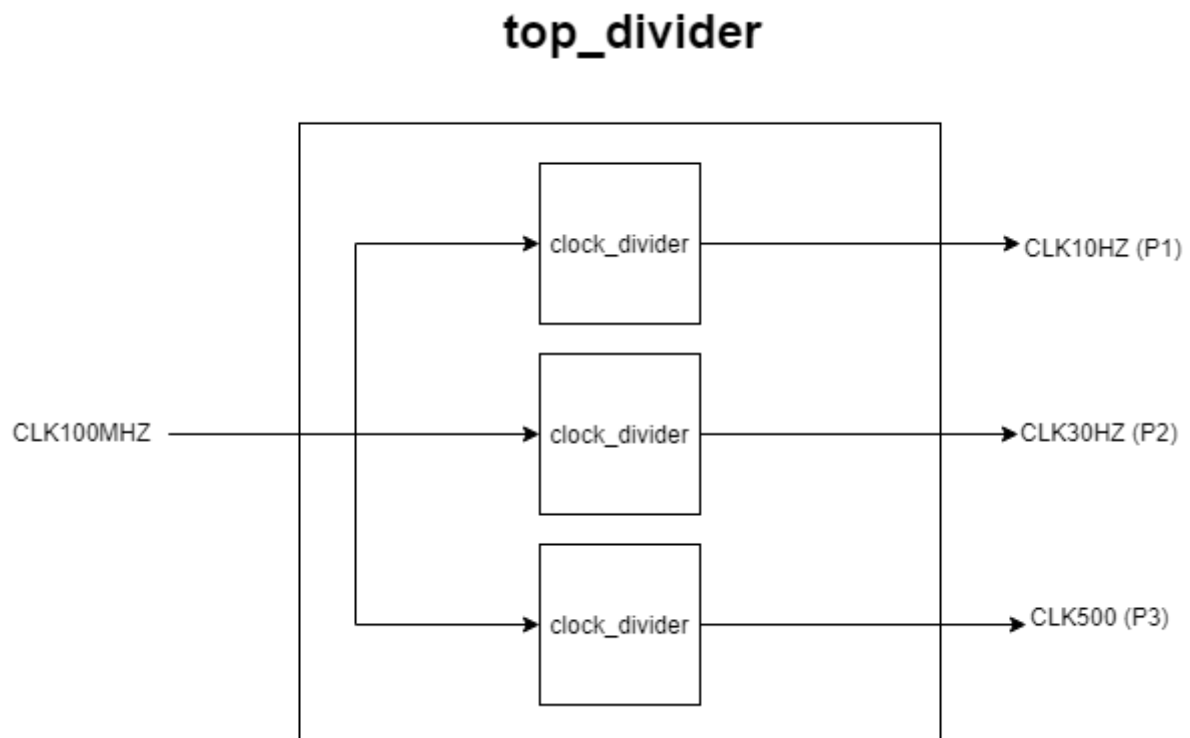
Nombre del módulo: top_divider

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.

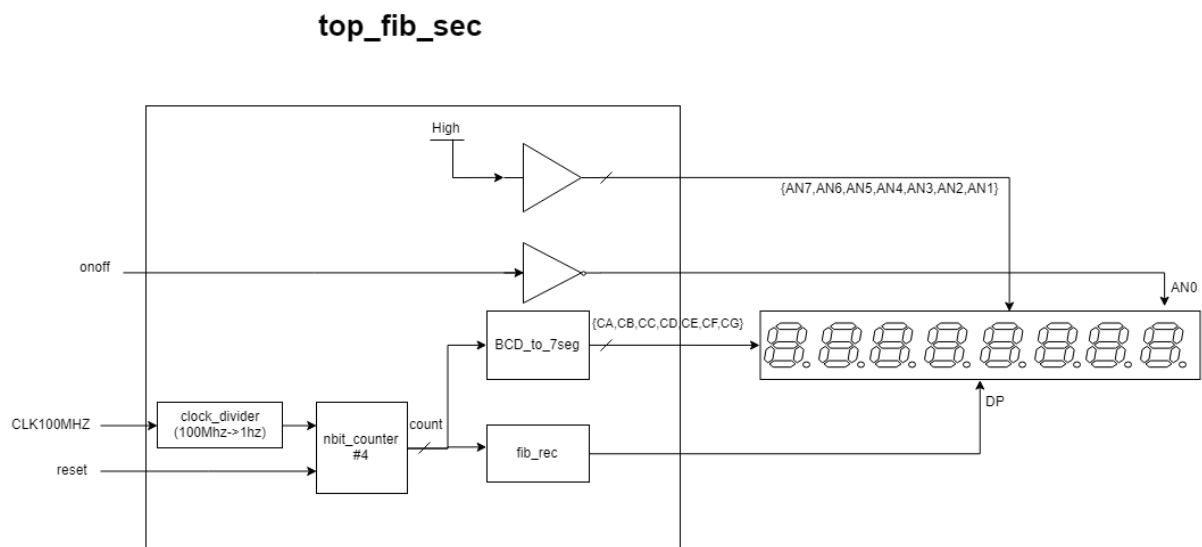
Nombre del módulo: top_fib_seq

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 = opcode. en todo otro caso la salida es 8'd0 y el status es 4'd0.

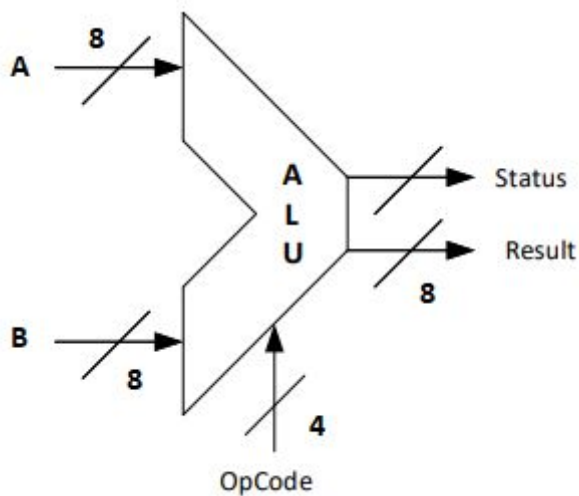
tiene un parámetro n, que determina los bits de entradas y result, default es 8bits.

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.

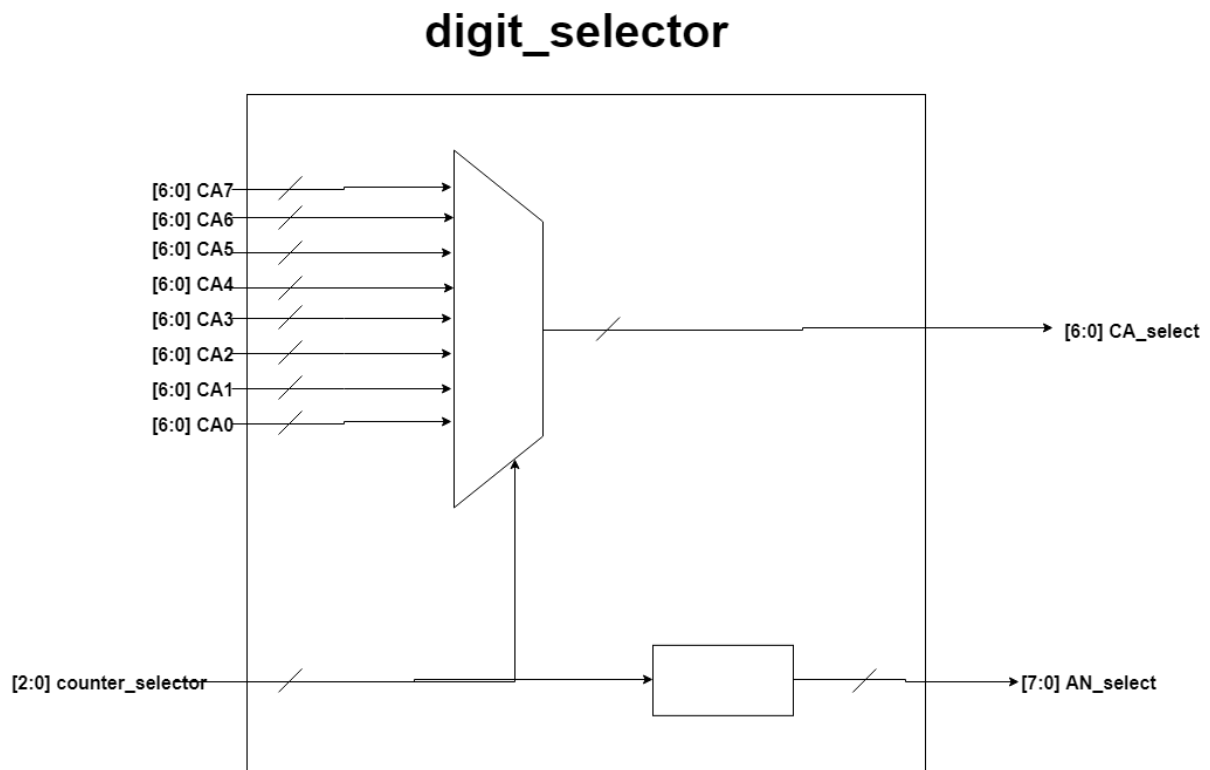
Nombre del módulo: digit_selector

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.

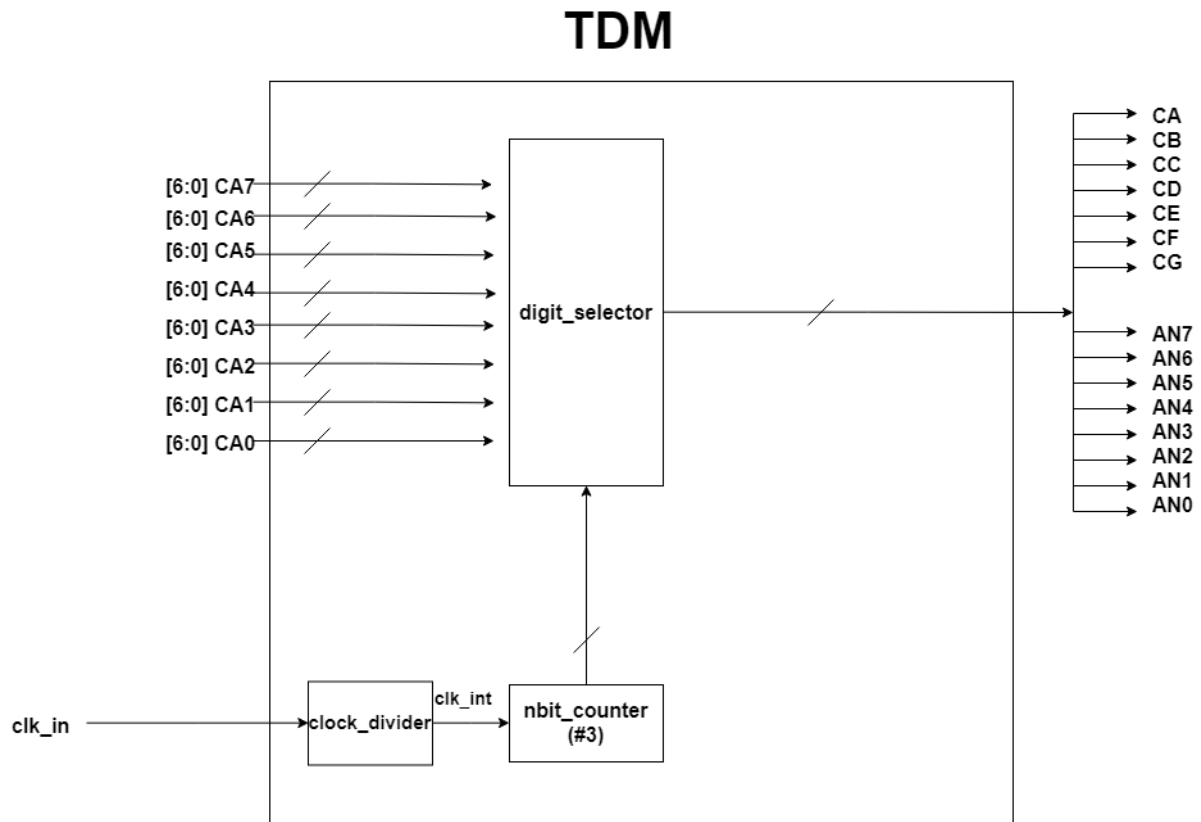
Nombre del módulo:

Entradas: clk_in, CA0-CA7

Salidas: CA-CG, AN7-AN0

Requiere: clock_divider, nbit_counter, digit_selector

Diagrama de alto nivel:



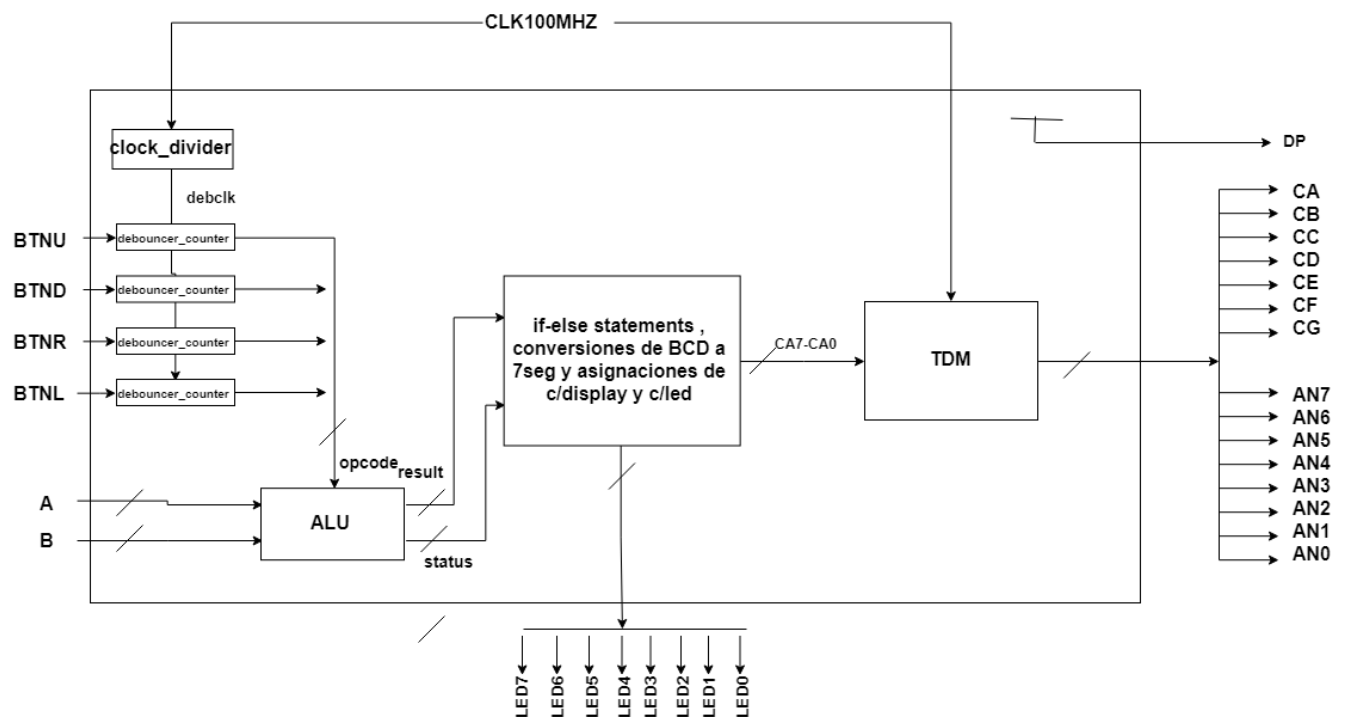
Display_ALU.sv

se conecta a los led y 7 segmentos para mostrar los resultados de un ALU cuyas entradas vienen de los switch y opcode de los botones.

Entradas: A7-A0,B7-B0 (switches),CLK100MHZ,BTNU,BTND,BTNR,BTNL (botones)

Salidas: CA-CG,AN7-AN0,DP,LED7-LED0

Requiere: clock_divider.sv, PB_Debouncer_counter.sv, ALU.sv, BCD_to_7seg.sv, TDM.sv, digit_selector.sv, nbit_counter.sv



4.1

group_display.sv

Usando dos switch como entrada, alterna entre mostrar en el display ELO212 y olivares, panatt y quevedo.

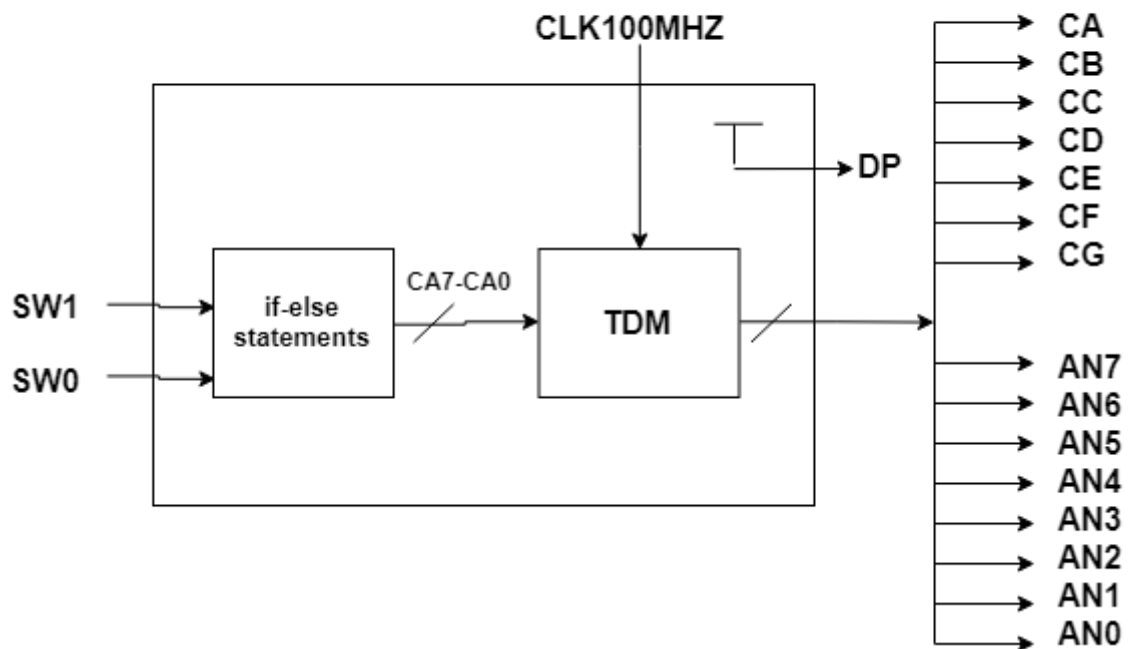
Entradas: SW1, SW0, CLK100MHZ

Salidas: AN7-AN0, CA-CG, DP

Requiere: TDM.sv

Diagrama de alto nivel:

group_display.sv



4.2

bin8bit_to_3decBCD.sv

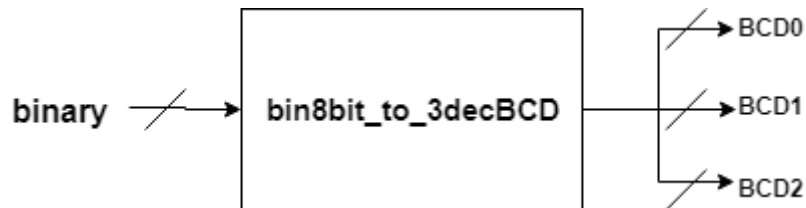
recibe una entrada binaria de 8 bits y entrega cada dígito del número decimal en BCD8421.

Entradas: binary

Salidas: BCD2, BCD1, BCD0

Requiere: N/A

Diagrama de alto nivel:



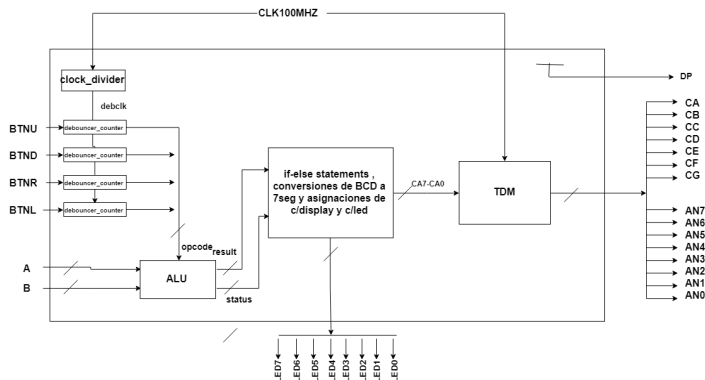
display_alu_dec.sv

Igual a Display_ALU.sv pero con números decimales.

Entradas: A7-A0, B7-B0 (switches), CLK100MHZ, BTNU, BTND, BTNR, BTNL (botones)

Salidas: CA-CG, AN7-AN0, DP, LED7-LED0

Requiere: clock_divider.sv, PB_Debouncer_counter.sv, ALU.sv, BCD_to_7seg.sv, TDM.sv, digit_selector.sv, nbit_counter.sv, bin8bit_to_3decBCD



*idem Display_ALU.sv pero el bloque de if else, etc contiene además conversiones de bin8bit_to_3decBCD.sv.

4.3

button_counter8hex.sv

Muestra en el display en hexadecimal cuantas veces se ha presionado un botón.

Entradas: CLK100MHZ, reset, BTNC

Salidas: CA-CG, AN7-AN0, DP

Requiere: clock_divider.sv, PB_Debouncer_counter.sv, nbit_counter.sv, BCD_to_sevenSeg, TDM.sv

Diagrama de alto nivel:

