library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity ALU is

generic

(

DATA\_WIDTH : natural := 4

);

port(

sw : in std\_logic\_vector(9 downto 0);

LEDR : out std\_logic\_vector(4 downto 0);

HEX0 : out std\_logic\_vector(6 downto 0);

HEX1 : out std\_logic\_vector(6 downto 0);

HEX2 : out std\_logic\_vector(6 downto 0);

HEX3 : out std\_logic\_vector(6 downto 0);

HEX4 : out std\_logic\_vector(6 downto 0);

HEX5 : out std\_logic\_vector(6 downto 0);

HEX6 : out std\_logic\_vector(6 downto 0));

end ALU;

architecture archConc of ALU is

signal a : std\_logic\_vector(DATA\_WIDTH-1 downto 0);

signal b : std\_logic\_vector(DATA\_WIDTH-1 downto 0);

signal mode : std\_logic\_vector(1 downto 0);

signal cout : std\_logic;

signal s : std\_logic\_vector(DATA\_WIDTH-1 downto 0);

signal somme : std\_logic\_vector(DATA\_WIDTH downto 0);

component d7seg is

port(

hexa : in std\_logic\_vector(3 downto 0);

seg : out std\_logic\_vector(6 downto 0));

end component;

begin

a <= SW(3 downto 0);

b <= SW(7 downto 4);

mode <= SW(9 downto 8);

LEDR(4) <= cout;

LEDR(3 downto 0) <= s;

entreeA0 : d7seg port map("0000",HEX5);

entreeA : d7seg port map(a,HEX4);

entreeB0 : d7seg port map("0000",HEX3);

entreeB : d7seg port map(b,HEX2);

sortie0 : d7seg port map("0000",HEX1);

sortie : d7seg port map(s,HEX0);

with mode select

somme <= std\_logic\_vector(unsigned('0'&a) + unsigned('0'&b)) when "00",

std\_logic\_vector(unsigned('0'&a) - unsigned('0'&b)) when "01",

'0' & a and '0' & b when "10",

'0' & a or '0' & b when "11",

'0' & a xor '0' & b when others;

s <= somme(DATA\_WIDTH-1 downto 0);

cout <= somme(DATA\_WIDTH);

end archConc;

======================================================================

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity accum is

port ( CLK,LOAD,RESET : in std\_logic;

DIN : in std\_logic\_vector(7 downto 0);

ACCU : out std\_logic\_vector(7 downto 0));

end accum;

architecture comport of accum is

signal accu\_s : std\_logic\_vector(7 downto 0);

begin

process (CLK)

begin

if (CLK'event and CLK = '1') then

if (RESET = '1') then accu\_s <= "00000000";

elsif (LOAD = '1') then accu\_s <= DIN;

else accu\_s <= std\_logic\_vector(unsigned(accu\_s) + unsigned(DIN));

end if;

end if;

end process;

ACCU <= accu\_s;

end comport;

EXO 1 QUESTION 4 ????

Marche pas dans le scin https://codeshare.io/scin