# ECE 574 – VLSI System Design

Homework n°2

Amine Gaizi

[gaizia@oregonstate.edu](mailto:gaizia@oregonstate.edu)

Student ID: 934-044-900

*1. The Verilog code:*

//Module Declaration: input and outputs

module alu(

input[7:0] in\_a, //input a

input [7:0] in\_b, //input b

input [3:0] opcode, //opcode input

output reg [7:0] alu\_out, //alu output

output reg alu\_zero, //logic '1' when alu\_output [7:0] is all zeros

output reg alu\_carry //indicates a carry out from AKY

);

//Declaration of the opcodes value in Hexadecimal

parameter c\_add = 4'h1;

parameter c\_sub = 4'h2;

parameter c\_inc = 4'h3;

parameter c\_dec = 4'h4;

parameter c\_or = 4'h5;

parameter c\_and = 4'h6;

parameter c\_xor = 4'h7;

parameter c\_shr = 4'h8;

parameter c\_shl = 4'h9;

parameter c\_onescomp = 4'hA;

parameter c\_twoscomp = 4'hB;

always\_comb begin //a

unique case(opcode) //unique modifier because only one option can match

c\_add: {alu\_carry, alu\_out} = in\_a + in\_b; //Arithmetic addition

c\_sub: {alu\_carry, alu\_out} = in\_a - in\_b; //Arithmetic susbtracion

c\_inc: {alu\_carry, alu\_out} = in\_a + 1; //Arithmetic incrementation

c\_dec: {alu\_carry, alu\_out} = in\_a - 1; //Arithmetic decrementation

c\_or:

begin

alu\_out = in\_a | in\_b; //Logic OR operation

alu\_carry = 0; //Reset carry

end

c\_and:

begin

alu\_out = in\_a & in\_b; //Logic AND operation

alu\_carry = 0; //Reset carry

end

c\_xor:

begin

alu\_out = in\_a ^ in\_b; //Logic XOR operation

alu\_carry = 0; //Reset carry

end

c\_shr:

begin

alu\_out = in\_a>>1; //Right shifting

alu\_carry = 0; //Reset carry

end

c\_shl: {alu\_carry, alu\_out} = 1<<in\_a; //Left shifting

c\_onescomp: alu\_out = ~in\_a;

c\_twoscomp:

begin

alu\_out = ~in\_a; //Invert

{alu\_carry, alu\_out} = alu\_out+1; //Add 1

end

default: {alu\_carry, alu\_out} = 9'b000000000; //If no match, reset alu\_out and alu\_carry

endcase

alu\_zero = ~| alu\_out; // NOR operation so: if alu\_out = 0, alu\_zero=1

end

endmodule

*2. Schematic of the designed ALU:*

Une image contenant circuit

Description générée automatiquement

*3. Waveform of the code working correctly*

*a. RTL level*

Une image contenant équipement électronique, ordinateur

Description générée automatiquement

*b. Gate level*

Une image contenant bâtiment, assis, vert, ordinateur

Description générée automatiquement

*c. Test code*

#undefined start

force in\_a 8'h00

force in\_b 8'h00

force opcode 4'b0000

run 20

#add 8+2 = 0xA

force in\_a 8'h08

force in\_b 8'h02

force opcode 4'b0001

run 20

#add test carry

# 0xE1 + 0x1F = 0x00 and Carry=1

force in\_a 8'hE1

force in\_b 8'h1F

force opcode 4'b0001

run 20

#susbstract 0x0E - Ox07 = 0x07

force in\_a 8'h0E

force in\_b 8'h07

force opcode 4'b0010

run 20

#increment 0x0F + 0x01 = 0x10

force in\_a 8'h0F

force in\_b 8'hXX

force opcode 4'b0011

run 20

#decrement expected result: 0x05

force in\_a 8'h06

force in\_b 8'hXX

force opcode 4'b0100

run 20

#logic or, 0b00001001 | 0b00001111 = 0b00001111

force in\_a 8'b00001001

force in\_b 8'b00001111

force opcode 4'b0101

run 20

#logic and

force in\_a 8'b11111101

force in\_b 8'b10101010

force opcode 4'b0110

run 20

#logic xor

force in\_a 8'b10101111

force in\_b 8'b01010001

force opcode 4'b0111

run 20

#wrong OP code for XOR -> goes to default

force in\_a 8'b10101111

force in\_b 8'b01010001

force opcode 4'b1111

run 20

#logic shift right

force in\_a 8'b10000000

force in\_b 8'hXX

force opcode 4'b1000

run 20

#logic shift left

force in\_a 8'b11111111

force in\_b 8'hXX

force opcode 4'b1001

run 20

#logic ones compe

force in\_a 8'h08

force in\_b 8'hXX

force opcode 4'b1001

run 20

#logic twos comp

force in\_a 8'h4D

force in\_b 8'hXX

force opcode 4'b1010

run 20

*4. Written answers:*

*a. Total area used by the ALU*

The total area use by the ALU is: 1814.079795 .

*b. Number of different cells utilized*

Report hierarchy displays:

AND2X1 saed90nm\_typ

AND3X1 saed90nm\_typ

AO21X1 saed90nm\_typ

AO22X1 saed90nm\_typ

AO221X1 saed90nm\_typ

AO222X1 saed90nm\_typ

AOI222X1 saed90nm\_typ

INVX0 saed90nm\_typ

LATCHX1 saed90nm\_typ

MUX21X1 saed90nm\_typ

NAND2X0 saed90nm\_typ

NAND3X0 saed90nm\_typ

NAND4X0 saed90nm\_typ

NOR2X0 saed90nm\_typ

OR2X1 saed90nm\_typ

OR4X1 saed90nm\_typ

XNOR2X1 saed90nm\_typ

alu\_DW01\_addsub\_0

FADDX1 saed90nm\_typ

XOR2X1 saed90nm\_typ

alu\_DW01\_ash\_0

AND2X1 saed90nm\_typ

AO21X1 saed90nm\_typ

INVX0 saed90nm\_typ

NAND2X0 saed90nm\_typ

NAND3X0 saed90nm\_typ

NOR2X0 saed90nm\_typ

OR3X1 saed90nm\_typ

This totals to 26 different gates utilized.

*c. Number of cells:*

The equivalent gate count is 328 gates.

*d. Hierarchical block:*

The block is an adder substractor (“alu\_DW01\_addsub”). It is implemented using a full adder block and an XOR logic gate.

*e. Delay:*

The maximum delay through the ALU is 2.68ns. The beginning point is “opcode[0](input port)”, and the endpoint is “alu\_zero(output\_port)”.