Carleton University

Department of Systems and Computer Engineering

SYSC 3006 (Computer Organization) fall 2020

Lab / Assignment 2 – Answers file

Student Name: Youssef Ibrahim

Part I – Answer those questions [1-mark/3]

1-1

a) [0.25-mark] Why can a single binary value represents both a signed integer value and an unsigned integer value?

A single binary value can represent both a signed integer value and an unsigned integer value since the semantic of a binary number is important.

b) [0.25-mark] For each component supplied in Logisim's Arithmetic Library, give a simple one-sentence summary (in your own words) of the function performed by the component.

- Adder: A component that adds two input values
- Subtractor: A component that calculates the difference between two input values
- Multiplier: A component that multiplies two input values
- Divider: A component that divides two input values
- Negator: A component that changes the sign of an input
- Comparator: A component that compares two voltages and outputs 1 if the voltage at positive side is greater or outputs 0 if the voltage at the negative side is greater
- Shifter: A component that has two inputs which include data and dist, and one output which is the result of shifting data by dist places. Both data and output have the same number of bits
- Bit Adder: A component that determines how many 1 bits are in the input and outputs the total number of 1 bits
- Bit Finder: A component that determines the index a bit where the index is computed from 0 as the lowest-order bit

c) [0.25-mark] Why are separate UMAX and SMAX operations included? Give an example in which the same input values (X and Y) would result in different output values (R) from the functions.

Since UMAX is used unsigned values, but SMAX is used for signed values. Example: if x=111 and y=011, using UMAX output values (R) would be 111 since 111(7) is bigger than 011(3); but if SMAX operation was used, output values (R) would be 011 since 011(3) is greater than 111(-3).

d) [0.25-mark] For each operation, propose test cases (i.e. X and Y input values) that can be used to test the ALU to demonstrate that it complies with the requirements of the operation given above. Each test case should expose different important aspects of the required functionality. There should be enough test cases to expose all important aspects of the required functionality. Describe each test in terms of any initial conditions, the inputs, the expected outputs, the particular aspects of the requirements that are tested, and why the test validates the aspects of the requirements being tested. Chose reasonable tests that give reasonable requirements coverage of both output values and flags, and reasonable descriptions.

1. NOP: 4-bit Op = 0000, Use a probe to check that C, V, N and, Z are don't care
2. ADD: 4-bit Op = 0001, Use 32-bit register for both inputs X and Y of the ALU, then connect bot these inputs to an Adder, then connect the output of the Adder and ALU to a Comparator, then connect a probe to the equal pin in the comparator to check for a number of toggled random values , both adders' outputs are equal. Then connect the carry in and carry out pins to thee comparator again to check the values for C and V. Then to check for N and Z values, ADD X and Y values that belong to R<0 and R=0.
3. SUB: 4-bit Op = 0010, then same as the steps followed for ADD
4. RY: 4-bit Op = 0011, then same as the steps followed for ADD
5. NEG: 4-bit Op = 1110, then same as the steps followed for ADD
6. UMIN: 4-bit Op = 1001, then same as the steps followed for ADD
7. SMIN: 4-bit Op = 1010, then same as the steps followed for ADD

Part II – Implementation of the ALU [2-mark/3]

Implement the required ALU. Use 32-bit registers for the inputs (X and Y). Use a 4-bit register as the operation selector (Op). Use a 32-bit probe to monitor the R output. Use individual 1-bit probes to monitor the Status Flag outputs.

2.1 - Circuit wiring [0.50-mark]

Insert a screenshot of your final ALU circuit here below and save a copy of your Logisim circuit (Save as) and name it “Lab2.circ” *(your “Lab2.circ” Logisim circuit must be included with your submission. We may need to be able to verify your circuit and rum few test cases in order to give you the marks for this part).* Briefly explain your circuit or some parts of your circuits as required

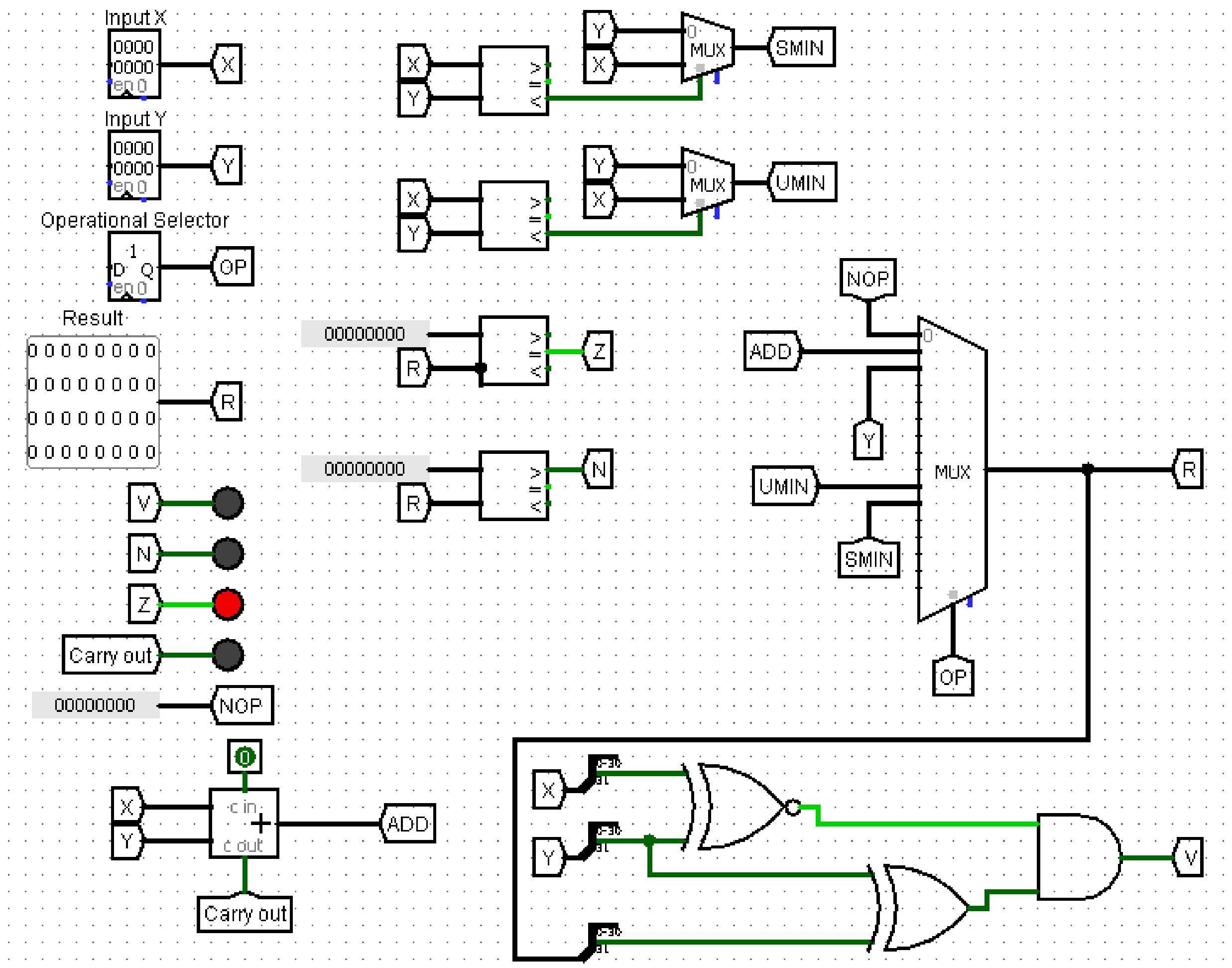

Figure 1: 32-Bit ALU

2.2 [1.5-mark]

Simulate and Log the test cases that you proposed in Part I above (unless you think that other test cases might be more appropriate ☺). Log the inputs and outputs of the circuit, including the flags (check Logisim documentation and the posted video about logging, they will give you an idea). Briefly comment your log to explain your results. If your solution is not complete before the deadline, try to log at least the working parts and give some explanations.

NOP OUTPUTS 0-32 BIT

Adder:

X	Y	R	C	V	N	Z
00000000	00000000	00000000	0	0	0	1
ffffffff	ffffffff	fffffffe	1	1	1	0

RY:

X	Y	R	C	V	N	Z
00000000	00000000	00000000	0	0	0	1

UMIN

X	Y	R	C	V	N	Z
10000000	00000000	00000000	0	0	0	1

SMIN

X	Y	R	C	V	N	Z
00000000	00000000	00000000	0	0	0	1

The circuit shown above IN Figure 1 takes input X, Y, and OP, then decides on which operation to use based on the input and Outputs the result of the operation (R), Negative (N), Overflow(V), Zero (Z), Carry Out (C)

Submission deadline

Must be submitter on cuLearn, locate (Assignment 2 submission) and follow instructions. Submission exact deadline (date and time) is displayed clearly within the Assignment 2 submission on cuLearn.

Note: If you have any question please ask them during your TA online live session. Do not wait till the deadline to ask questions as there in no warranty you will get answered before the deadline.

Good Luck