Lab Final Exam (Circuits Portion)
COMP 122/L
Fall 2017

Score breakdown:
Assembly (online): 35
Arithmetic Logic Unit: 33
Register File: 32
Total: 100

Please write your name below, and **wait until told to begin:**

Once told to begin, tear out the last sheet of this document, which contains the questions to answer. You can use this to easily refer back to the question while you answer it.

Ground Rules

1. You may have three 8 1/2 x 11 inch sheets of paper in front of you containing handwritten notes covering both sides of each sheet. It does not need to be the same sheet from last week.

2. If you have a question, raise your hand and I will come to you.

3. You **may not** communicate with anyone else. **Violations of this rule will result in a 0 on the exam.** This exam is purely individual effort.

Arithmetic Logic Units

Write your solution below.

Register Files

Write your solution below.

Arithmetic Logic Units

Design a 3-bit ALU which supports subtraction and bitwise `AND`. This ALU takes the following inputs:

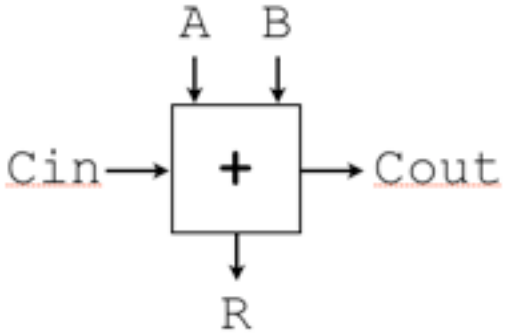
Input Name	Input Description
`A0`	Bit 0 of the first operand
`A1`	Bit 1 of the first operand
`A2`	Bit 2 of the first operand
`B0`	Bit 0 of the second operand
`B1`	Bit 1 of the second operand
`B2`	Bit 2 of the second operand
`Op`	The operation to perform; `0` for subtraction and `1` for `AND`

Given the above inputs, the ALU produces the following outputs:

Output Name	Output Description
`U0`	Bit 0 of the result
`U1`	Bit 1 of the result
`U2`	Bit 2 of the result
`Cout`	Carry-out for subtraction. This output is permitted to take on any value you want if the input operation isn't subtraction.
`V`	Overflow for subtraction. This output is permitted to take on any value you want if the input operation isn't subtraction.

For this task, you may use **only** the following components, in unlimited supply:

- `AND`, `OR`, and `NOT` gates
- 2-input multiplexers
- Single-bit adders, shown to the right. Given two single-bit inputs `A` and `B`, along with a single carry-in bit `Cin`, this produces a single-bit result `R` and a carry-out bit `Cout`



Register Files

Design a register file holding two 3-bit registers. The register file takes the following inputs:

Input Name	Input Description
`WE`	Short for "Write Enable"; set to `1` if we are writing to a register, else `0`
`CLK`	The system clock
`R`	Which register we are reading from / writing to; `0` for the first register and `1` for the second register
`I0`	Bit 0 of the input to write. Ignored if `WE = 0`.
`I1`	Bit 1 of the input to write. Ignored if `WE = 0`.
`I2`	Bit 2 of the input to write. Ignored if `WE = 0`.

Given the above inputs, the register file produces the following outputs:

Output Name	Output Description
`U0`	Bit 0 of the result
`U1`	Bit 1 of the result
`U2`	Bit 2 of the result

For this task, you may use **only** the following components, in unlimited supply:

- `AND`, `OR`, and `NOT` gates
- 2-input multiplexers
- D flip-flops