LAB 6 // Hardware

MULTIPLEXORS AND ALUS

INSTRUCTIONS and RUBRIC

CSE 2301 – Fall 2024

INTRODUCTION

Prerequisites

In addition to the expectation that you be familiar with binary operators and arithmetic, you will also need to know about Arithmetic Logic Units, adders, binary multiplication, and multiplexors.

Objectives

In this three-part lab, you will first design an ALU using adders, multiplexors, and standard logic gates.

Background

Computers must perform arithmetic operations millions of times per second to perform their expected functions. The exact mechanism by which the processor schedules and executes these tasks is beyond the scope of this class, but the ALU, or Arithmetic Logic Unit, is critical to such operations, as it is responsible for performing nearly all of the processor's mathematical and logical functions. Real ALUs are capable of handling 32-bit, 64-bit, and yet larger pairs of inputs, but we will be sticking with considerably smaller values.

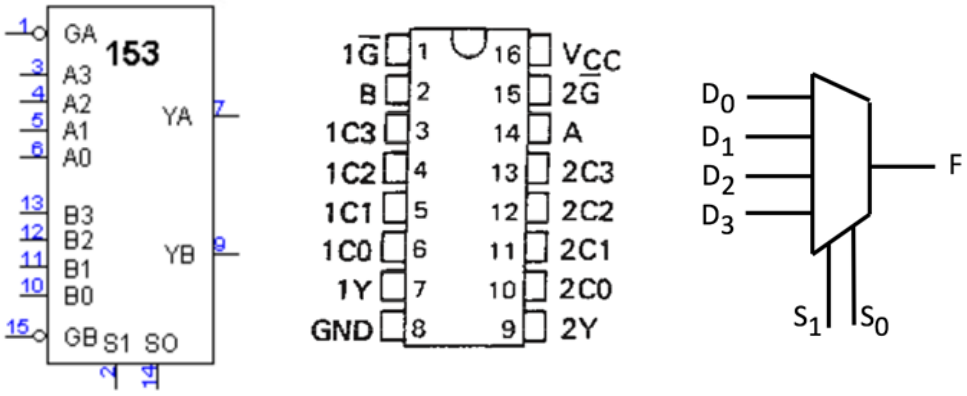
PART 1:

Design an ALU

Critical to understanding ALUs are multiplexors, full adders, and binary multiplication. You have already used adders.

Multiplexing

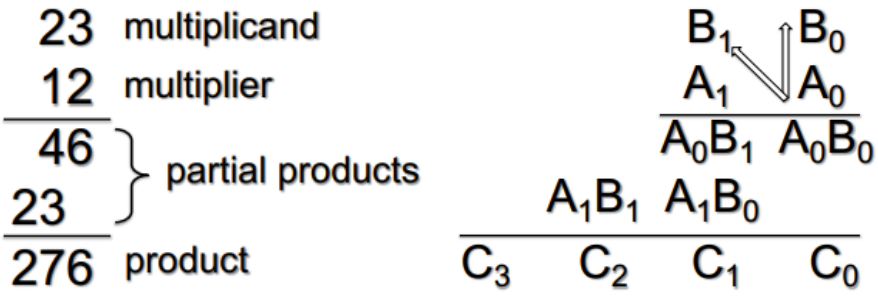
Multiplexors are like railroads: they can switch which pin to output based on selection instructions. In this part of the lab a 74153 will be used, which is a dual 4-1 MUX. We have two sets of four pins, but from each set, only one pin is selected as output (based on the selector). A 74153 has two selector bits, which means that it makes a selection for both sets of inputs, not just one. Three photos are provided below. The first is a 74153 as seen in LogicWorks, the second is a pin-out diagram for the physical chip, and the third is the schematic representation of a generic 4-1 multiplexor.



Multiplication

Additionally, the concept of binary multiplication is important, although for part 1, since we are only multiplying two 2-bit numbers, the implementation will be relatively simple.

- When we perform long multiplication in decimal, we work by first deciding on the partial products, and then add them together. The same is true of binary multiplication.
- First, we compute each product, making sure to shift left for each new partial product, and then add. In the case of binary it is difficult to add more than two numbers together, so we often add the first two partial products, then add each additional partial product to the previous sum.
- Try to convert this exact process into combinational logic when designing your multiplier.



YOUR TASK – Design an ALU circuit. This circuit will take two 2-bit numbers as input. You must be able to select between and display AND, OR, addition, or multiplication operations. You will have to design the multiplier yourself using nothing more than basic logic gates and adders (i.e. 7483/74283). Your implementation should have two 74153 chips (quad 4-1 MUX) and a 4-bit output display. Selection codes are shown below.

S1	So	Operation
0	0	AND
0	1	OR
1	0	Addition
1	1	Multiplication

PART 2:

Multiplication Extension

Multiplying two 2-bit numbers is rather simple, but what if we had something bigger, like a 4-bit number and a 4-bit number? You are not required to implement this in hardware.

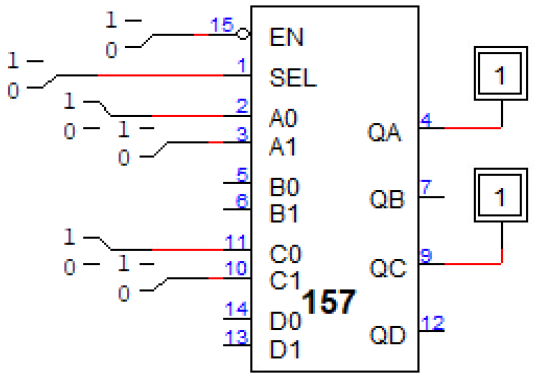
YOUR TASK – Using the principles of binary multiplication that have already been discussed, design a circuit that takes a 4-bit input and a 4-bit input and multiplies. Your implementation should be done in a separate LogicWorks file. The functionality of this design will be worth one of the five demo points for this lab, and part of your report.

PART 3:

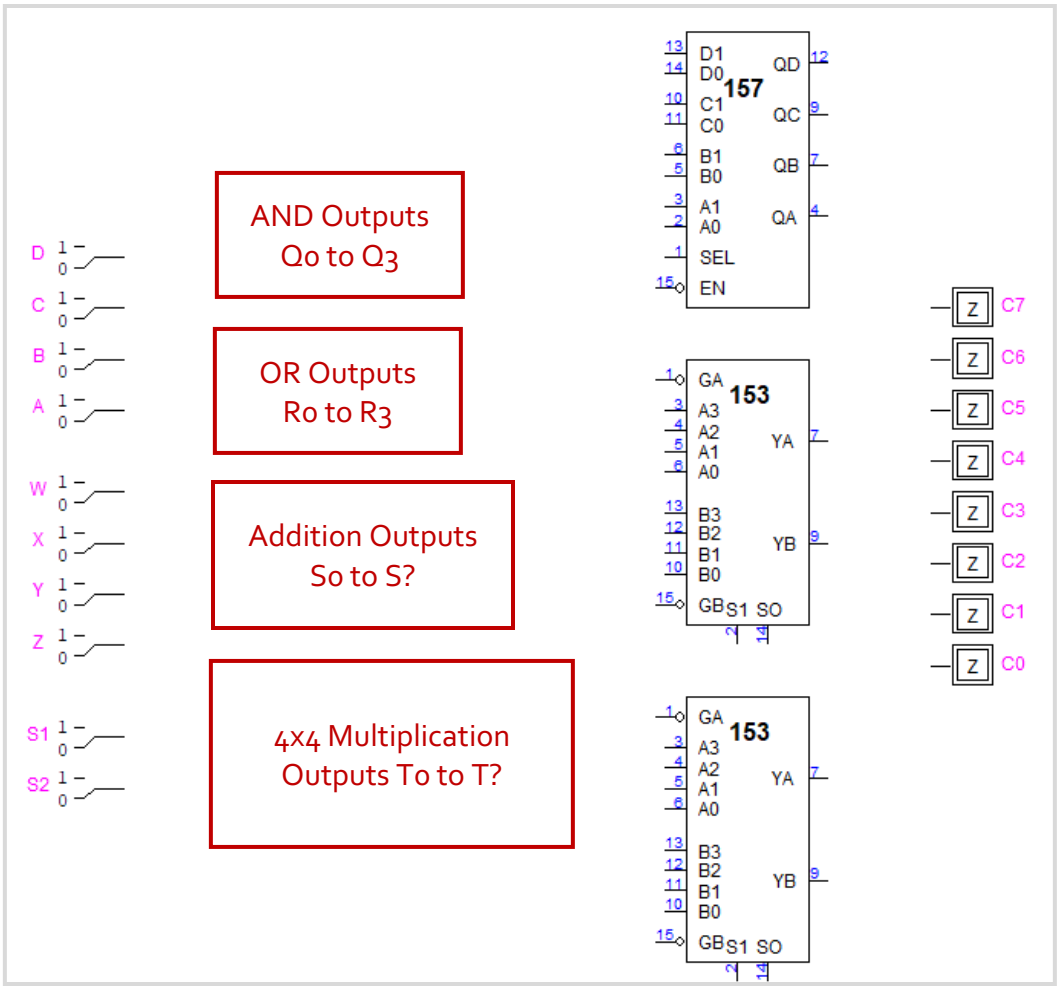
ALU Extension

Now let's imagine that we had a much larger part 1 design where we had to design an ALU for a pair of 4-bit inputs. You can imagine that the multiplication part of the circuit would be even larger than one from part 2, but what's important in this exercise is how to handle the output. Recognize that, again, we need to able to handle AND, OR, addition, and multiplication operations. When we do basic logical operations on two 4-bit numbers, the output is obviously 4 bits, but what of addition and multiplication?

For this task you will also have to use a 74157 multiplexor. In principle it is just like any other multiplexor, but it is often specifically used for choosing between two sets of 4-bit inputs. This is because the 74157 is a quad 2-1 MUX, which means that there is only one selector bit to handle four pairs of inputs. If you had to choose between 4-bit values PQRS and WXYZ, for example, you might wire the former to $D_0C_0B_0A_0$ and the latter to $D_1C_1B_1A_1$ (see diagram). In this way, when SEL = **0**, the output is PQRS, and when SEL = **1**, the output is WXYZ.



YOUR TASK – Conceptually design (but **do NOT implement**) an ALU circuit that can handle the output from two 4-bit numbers. You are limited to using two 74153 chips (dual 4-1 MUX) and one 74157 (quad 2-1 MUX). Specifically, using the diagram below, tell us what the inputs to and outputs from all of the pins on the three multiplexors shown should be. Do not demo. This is part of your report.



SCORING

Demo Requirements [5 pts]:

- ✓ Develop an ALU design (probably on paper to start)
- ✓ Implement your ALU design in LogicWorks
- ✓ Use your LogicWorks design to develop the circuit on hardware **(4 pts)**
- ✓ In part 2, create a 4-bit by 4-bit multiplier in LogicWorks **(1 pt)**

Report Requirements [2.6 pts]:

- ✓ [1.0] Theory:
  - [0.6] You have created a 2x2 binary multiplier and a 3x4 binary multiplier. Explain the theory behind creating multiplication circuits for any input size, and how to determine the maximum size of the output (in bits) from those inputs.
  - [0.4] 74153 chips have two 4-1 multiplexors, but they are bound to the same selector bits, so they are not entirely independent. Why is this beneficial, in our case? What would we have to do if we wanted different selectors for each MUX?
- ✓ [1.2] Deliverables:

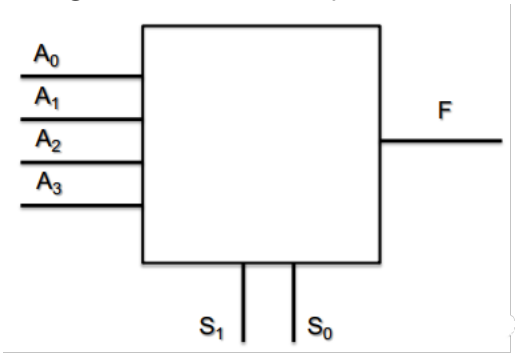
- [1.2] Present your part 3 design. It is recommended that you include a diagram or two, which may be drawn, so long as the drawings are legible. Otherwise your explanations should be sufficiently detailed to be replicated on hardware.

✓ [0.4] Discussion section. Should conform to standard lab report guidelines.

Practice Questions [2.4 pts]:

✓ [0.7] *Question 1:*
Consider F = ∑m(0, 1, 6, 7) for input binary variables C, B, A , where CBA is weighted 421 and m indicates minterms. Implement this circuit using an 8-1 MUX (multiplexor).

✓ [0.7] *Question 2:*
Imagine the following circuit: a 4-1 MUX (multiplexor) gets input from A0 through A3, is controlled by S0 and S1 (where S0 is the LSB) and produces output at F. A0 and A1 are connected to ground, and A3 is connected to Vcc. A2 is connected to an XOR gate which produces $X' \oplus Y$. What is F when S0 and X equal 0 and S1 and Y equal 1?



✓ [1.0] *Question 3:*
Multiply the following numbers together using binary multiplication. Show your work.

a) 11_{10} (1011_2) and 9 (1001_2) Final answer should be 8 bits (binary) (0.5 pts).
b) -15_{10} and 7_{10}. Express your final result as a 2's Complement number with 8 bits (0.5 pts). Don't multiply negative numbers!