Jason Vu

ELEN/COEN 21 Lab 4 Report

- Write an introduction describing the behavior of your circuit for the implementations of all three outputs S_2, S_1 and S_0 that you made.
- Include the K-maps and the logic expressions for the functions you used.
- Include your final schematics, your final Verilog code.
- If, during testing, you found problems with the circuit that required correction, explain what you observed that demonstrated the problem and describe how you determined the cause of the problem and how you fixed it.
- Write a conclusion about the challenges of this lab and what you learned in this lab.

In order to design this circuit, we created the complete truth table with four inputs and three outputs and then drew K-maps to get the Sum of Products (SOP) form for all of the outputs. We made symbols for each of the seven blocks to differentiate between different input and output functions, so we could obviously go back and address any runtime/compile errors and technical uncertainties if needed.

Basically, we made the three following inputs - A0, B1, and B0 - which would connect to their respective block diagrams, executing the program then connecting to multiplexers. For each set of inputs, the block diagram would abstract functions through the form of smaller expressions that the inputs would connect to, each rendering one input (except for the first set of inputs which were connected into two). Keep in mind, that the A1 is a selector that doesn't have it inside the blocks but instead goes to the multiplexer.

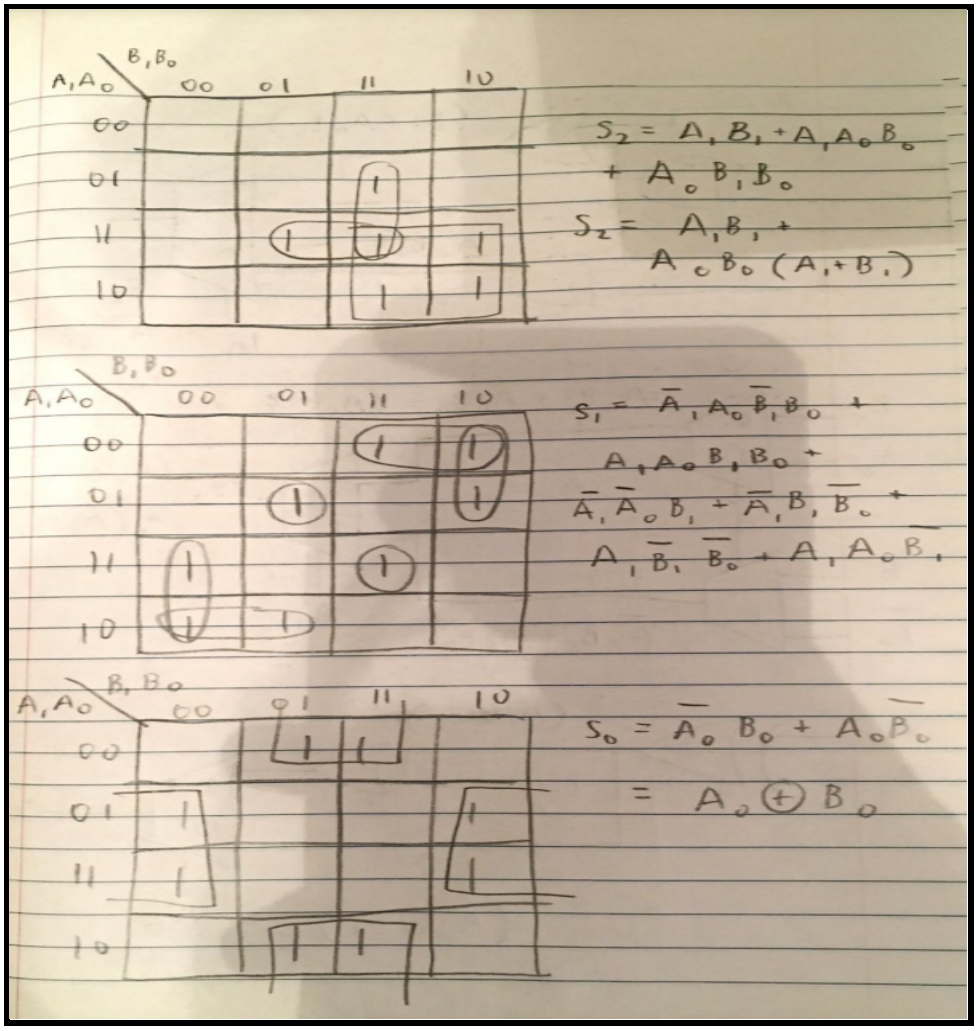
A problem we had was finding the correct way to connect all of our blocks using a multiplexer. Initially, we attempted to use the logic circuit that was referenced in the lab document but quickly reconsidered after it didn't work out in the program. To solve this, we specified the inputs and outputs for our Verilog module which would then be compiled and applied to the entire schematic. Another problem we encountered was for S1F1, we made a mistake with connecting the inputs to a gate. I was supposed to connect inputs B0' to A', but instead I connected B1' to B0'. This was an easily avoidable problem.

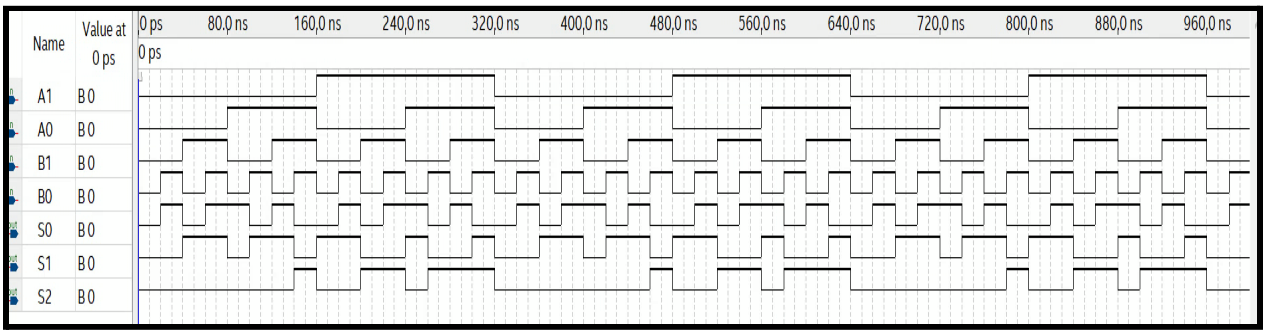
The challenges in this lab included implementing a multiplexer using Verilog, making sure I don't make a mistake when I'm drawing out the schematics that take place inside of the blocks, etc. We learned from this lab how to implement a multiplexer using Verilog and how to create/update a schematic into a symbol/block file and then implementing it into a main file where we then ran waveform and functional simulation tests.

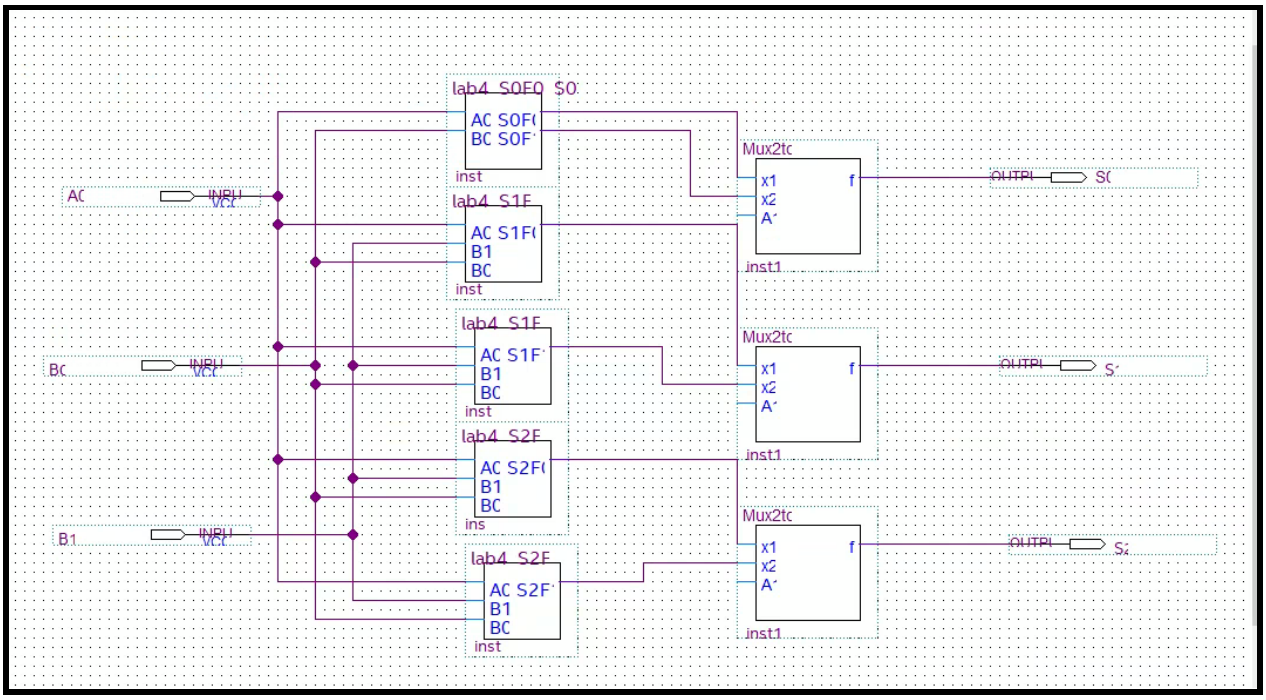
- 2-bit adder circuit:

A1	A0	B1	B0	S2	S1	S0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

- K-maps and the logic expressions




Name
Value at
0 ps
0 ps
80,0 ns
160,0 ns
240,0 ns
320,0 ns
400,0 ns
480,0 ns
560,0 ns
640,0 ns
720,0 ns
800,0 ns
880,0 ns
960,0 ns
A1
A0
B1
B0
S0
S1
S2


lab4_S0F0_S0
AC S0F0
BC S0F1
inst
lab4_S1F
AC S1F0
B1
BC
inst
lab4_S1F
AC S1F1
B1
BC
inst
lab4_S2F
AC S2F0
B1
BC
ins
lab4_S2F
AC S2F1
B1
BC
inst
Mux2tc
x1
x2
A1
f
inst1
Mux2tc
x1
x2
A1
f
inst1
Mux2tc
x1
x2
A1
f
inst1
AC
INPU
VCC
BC
INPU
VCC
B1
INPU
VCC
OUTPU
S0
OUTPU
S1
OUTPU
S2