Digital Logic Design 3441 – Spring 2021

Lab #2: The Digital Design Process with SimUAid

Cody Betzner, Thomas Nguyen

Overview: Goals of the Lab

The goal of this lab is to practice simplifying boolean expressions and to observe why simplification is important. In this lab we learned how to calculate the number of input combinations, and simplified two boolean expressions to find that the simplified versions output the same values, but contain less gates, reducing the cost of the circuit. We modeled the circuits in SimUAid, making use of clocks and timing diagrams to visualize computation timings.

Part 1: Simplifying Boolean Expressions Using Table 2-3 and 2-4

F1(ABC)=(A+B')'·C+C'·(C+B'+A)'	
F1(ABC)=(A'B)·C+C'·(C'BA')	DeMorgan's Law
F1(ABC)=(A'BC)+(C'BA')	
F1(ABC)=A'(BC+BC')	1st Distributive Law
F1(ABC)=A'(B(C+C'))	1st Distributive Law
F1(ABC)=A'(B·(1))	Law of Complementarity
F1(ABC)=A'·B	
F1*S*(ABC)=A'·B	

1. Original: AND, OR, NOT

   Simplified: NOT, AND
2. 74LS04 (NOT), 74LS08 (AND), 74LS32 (OR)
3. Reducing F1, or any logic circuit, allows the manufacturer to create less unnecessary connections thus reducing the number of resources used. This helps to reduce cost of manufacturing and makes the circuit easier to use and configure.

Part 2: Simplification using New Logic Gates using Section 3.2

$F2(ABC)=A\oplus B\oplus AB$

$F2(ABC)=(A'B+AB')\oplus AB$

$F2(ABC)=(A'B+AB')'(AB)+(A'B+AB')(AB)'$

$F2(ABC)=(A+B')(A'+B)(AB)+(A'B+AB')(A+B)$	DeMorgan's Law
$F2(ABC)=(A+B')(ABA'+ABB)+(A'B+AB')(A+B)$	1st Distributive Law
$F2(ABC)=(A+B')(0+AB)+(A'B+AB')(A+B)$	Laws of Complementarity
$F2(ABC)=(AAB+B'AB)+(A'B+AB')(A+B)$	1st Distributive Law
$F2(ABC)=(AB+0)+(A'B+AB')(A+B)$	Laws of Complementarity
$F2(ABC)=(AB)+(A'BA+A'BB+AB'A+AB'B)$	Multiplying Out
$F2(ABC)=(AB)+(0+A'B+AB'+0)$	Laws of Complementarity
$F2(ABC)=AB+A'B+AB'$	
$F2(ABC)=A(B+B')+A'B$	1st Distributive Law
$F2(ABC)=A(1)+A'B$	Laws of Complementarity
$F2(ABC)=A+A'B$	
$F2(ABC)=A+B$	Elimination Theorem

$F2S(ABC)=A+B$

4. Original: AND, XOR

   Simplified: OR

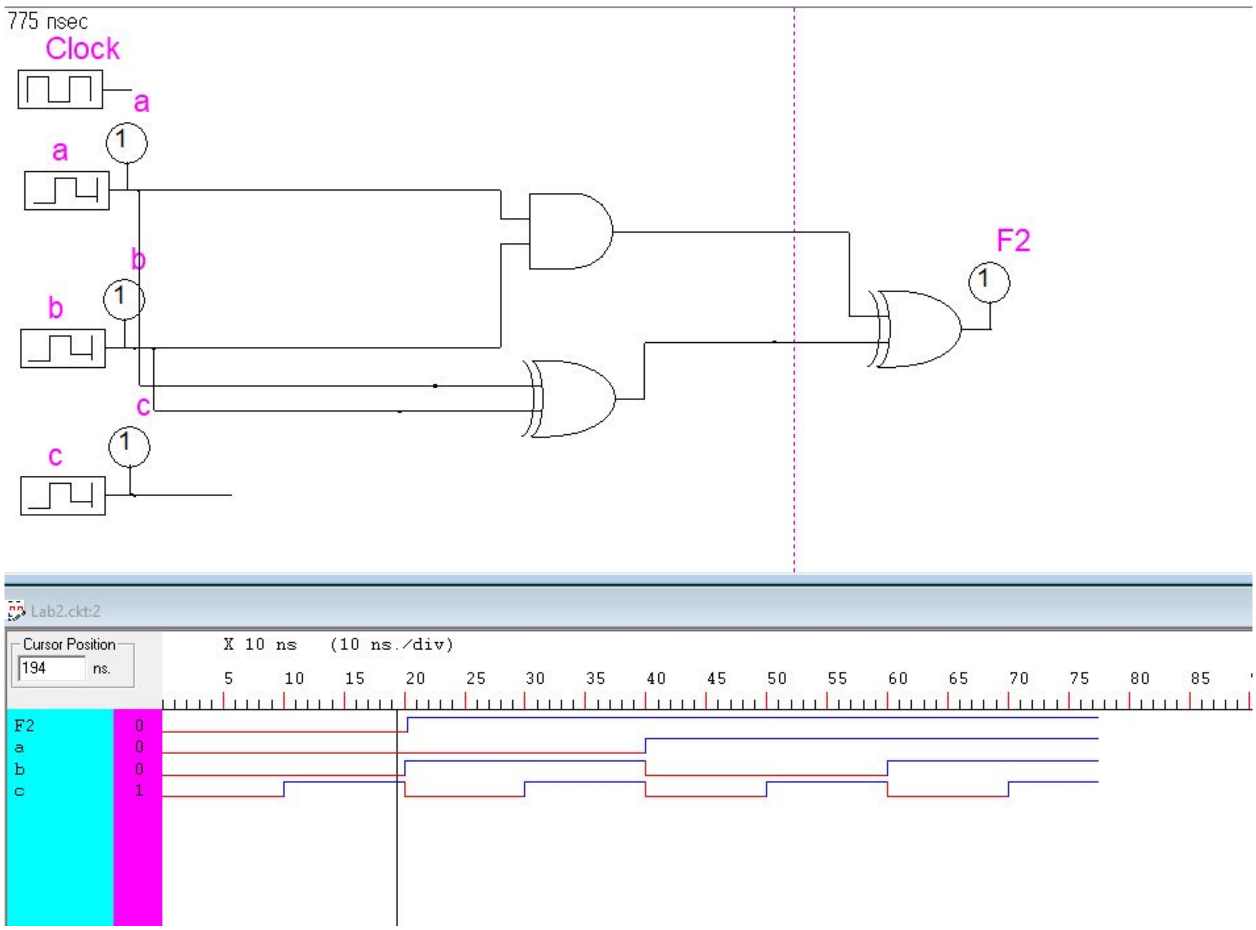
5. 74LS08 (AND), 74LS32 (OR), 74LS86(XOR)

6. $F2(ABC) = 2^3 = 8$

   $F2S(ABC) = 2^3 = 8$

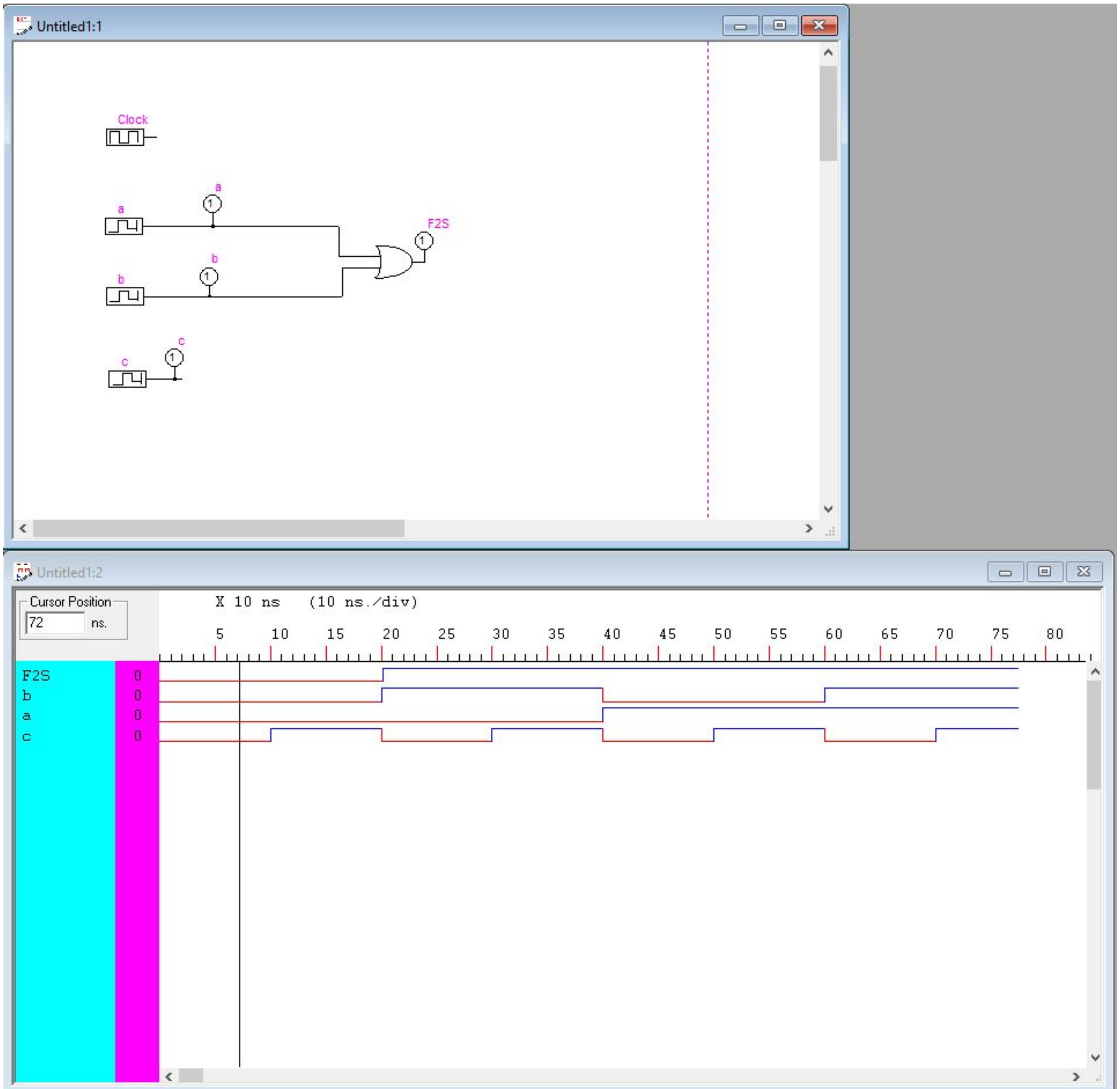
Part 3: Creating both F2 and F2S Boolean Expressions with New SimUAid Techniques

F2(ABC)=A⊕B⊕AB
Input Combinations = 2^3 = 8.
ABC = {000,001,010,011,100,101,110,111}



Input A	Input B	Input C	F2
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

F2S(ABC)=A+B
Input Combinations = 2^3 = 8.
ABC = {000,001,010,011,100,101,110,111}



Input A	Input B	Input C	F2S
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Conclusion

In alignment with the overall objective of the lab, we confirmed and proved the validity of simplification theorems using SimUAid. We learned how to configure a clock as well as input signals for inputs within SimUAid in order to prove that an XOR gate, when simplified, still has the same outputs as if it were not simplified. We also applied many simplification theorems when simplifying more complex gates such as elimination and distributive. Additionally, we also learned how to calculate the quantity of input combinations of an input.