Combinatorial Logic

Submitted by

Caleb Burke

658780838

Department of Computer Science

Faculty of Science and Technology

Vancouver Island University

CSCI 355 Digital Logic and Computer Organization

Submitted to

Prof. Ajay Shrestha

September 24, 2024

Table of Contents

Contents

1. Objectives

1. Construct NAND equivalent for Sum-of-Products
2. Construct NOR equivalent for Product-of-Sums
3. Familiarization with Exclusive OR (XOR) gate
4. Construct XOR using NAND only logics/ NOR only logics

2. Components Required

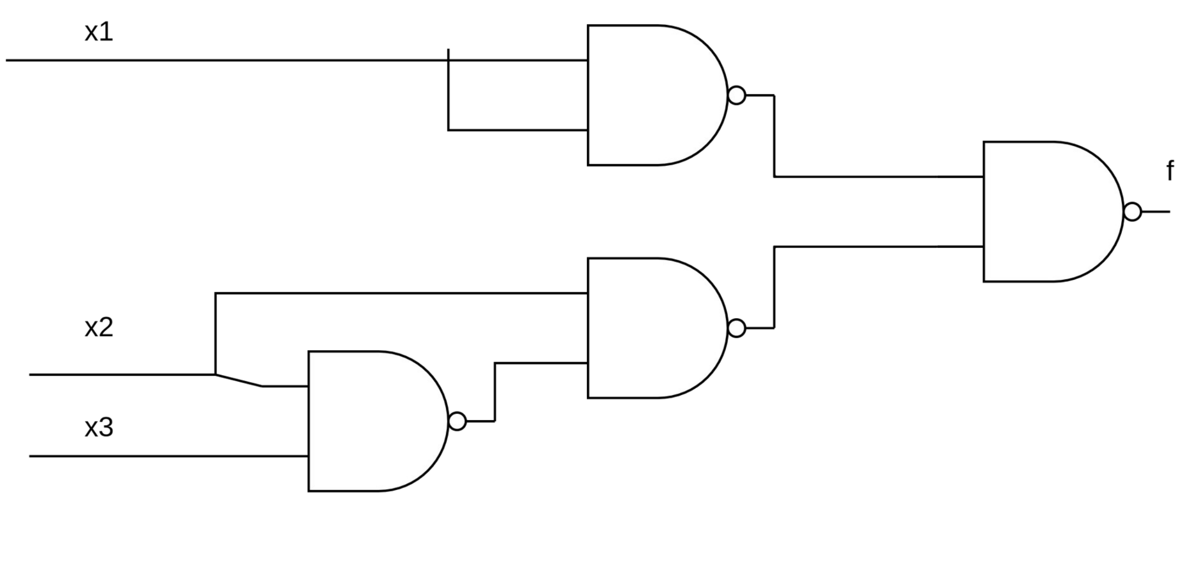
- Power supply
- 1 x 7400 Quad 2-Input NAND Gate
- 2 x 7402 Quad 2-Input NOR Gate
- 1 X 7486 Quad 2-input XOR
- Breadboard
- Digital Circuit Evaluator
- Wiring Kit

4. Pre-Lab

4.1 NAND Equivalent for Sum-of-Products

x1	x2	x3	x2 * ~x3	f
0	0	0	0	0
0	0	1	0	0
0	1	0	1	1
0	1	1	0	0
1	0	0	0	1
1	0	1	0	1
1	1	0	1	1
1	1	1	0	1

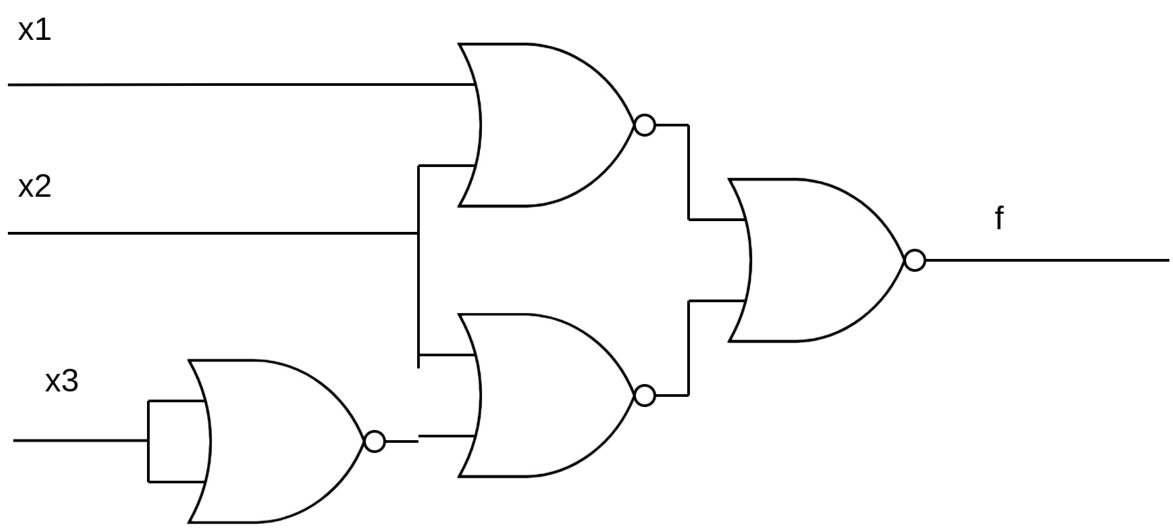
Logical diagram equivalent using only NAND gates



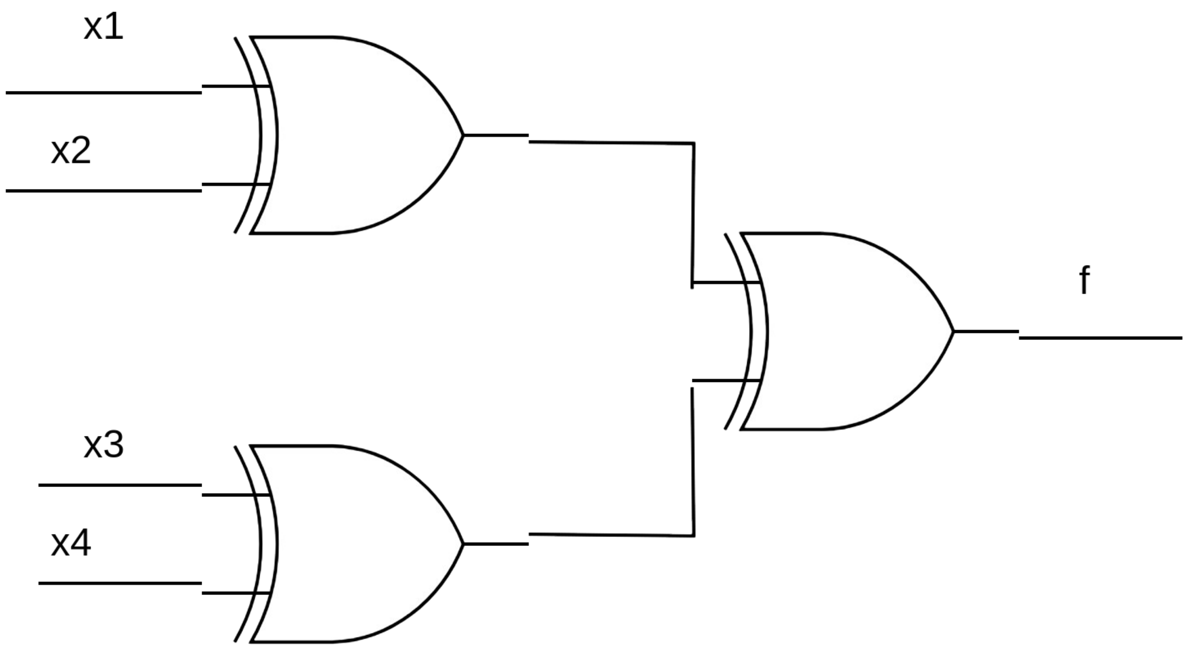
4.2 NOR Equivalent of Product-of-Sums

x1	x2	x3	x1 | x2	~x3 | x2	f
0	0	0	0	1	0
0	0	1	0	0	0
0	1	0	1	1	1
0	1	1	1	1	1
1	0	0	1	1	1
1	0	1	1	0	0
1	1	0	1	1	1
1	1	1	1	1	1

Logical diagram equivalent using only NOR gates



4.3 A 4-input XOR using 2-input XOR Gates



Explaination:

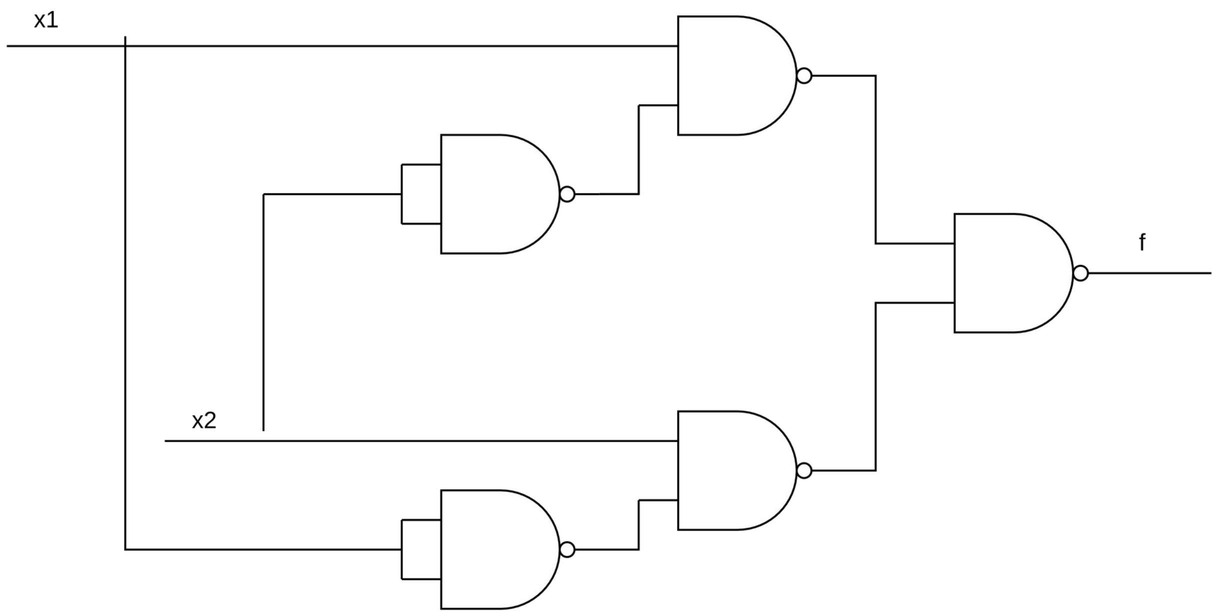
The truth table of the partiy of a pair of bits it equal to the truth table of XOR on a pair of bits. Given this, to get the parity of the entire input just XOR each pair of bits in the input, this will give you partitions of partiy. Taking pairs of these partitions and XOR them will give you the partiy of that part of the input, repeat XOR on pairs until one XOR left and thats the parity of the entire input. (This works for inputs with an odd number of bits, just choose one bit to leave for the last XOR and you will get the correct parity)

Truth table

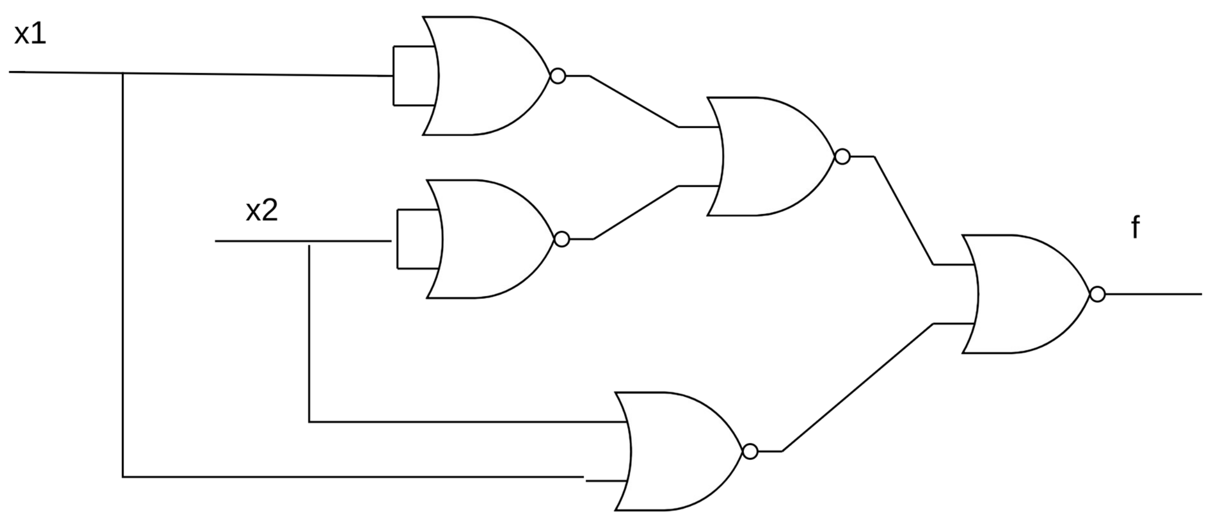
x1	x2	x3	x4	f	Number of 1's parity
0	0	0	0	0	even
0	0	0	1	1	odd
0	0	1	0	1	odd
0	0	1	1	0	even
0	1	0	0	1	odd
0	1	0	1	0	even
0	1	1	0	0	even
0	1	1	1	1	odd
1	0	0	0	1	odd
1	0	0	1	0	even
1	0	1	0	0	even
1	0	1	1	1	odd
1	1	0	0	0	even
1	1	0	1	1	odd
1	1	1	0	1	odd
1	1	1	1	0	even

4.4 XOR Gate using only NAND logic and only NOR logics

Only NAND



Only NOR

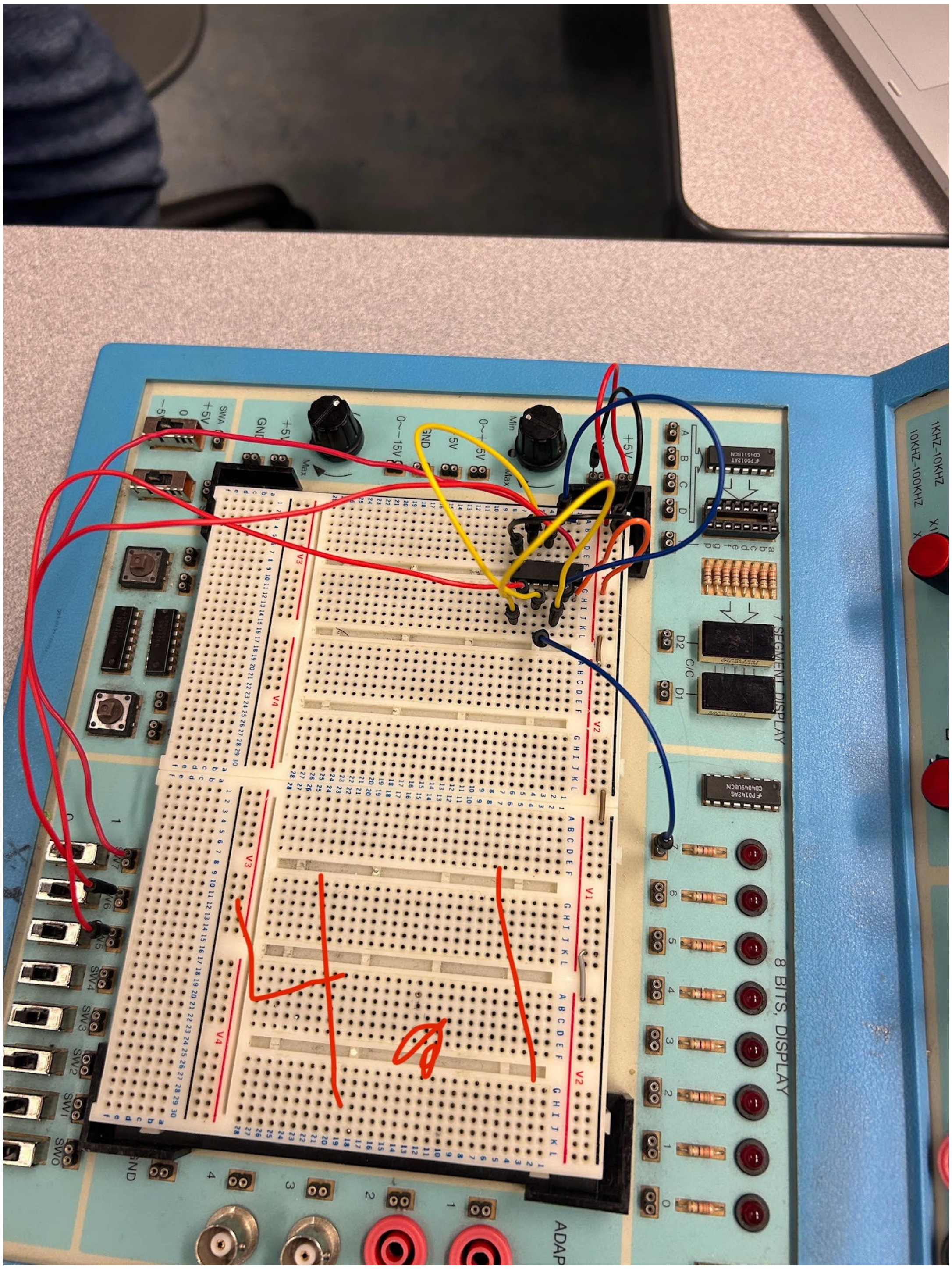


7. Lab Procedure with Deliverables

7.1 NAND Equivalent for Sum-of-Products

I. NAND-only version of the Sum-of-Products circuit developed in Prelab 4.1.

II. Picture of breadboard with the completed circuit



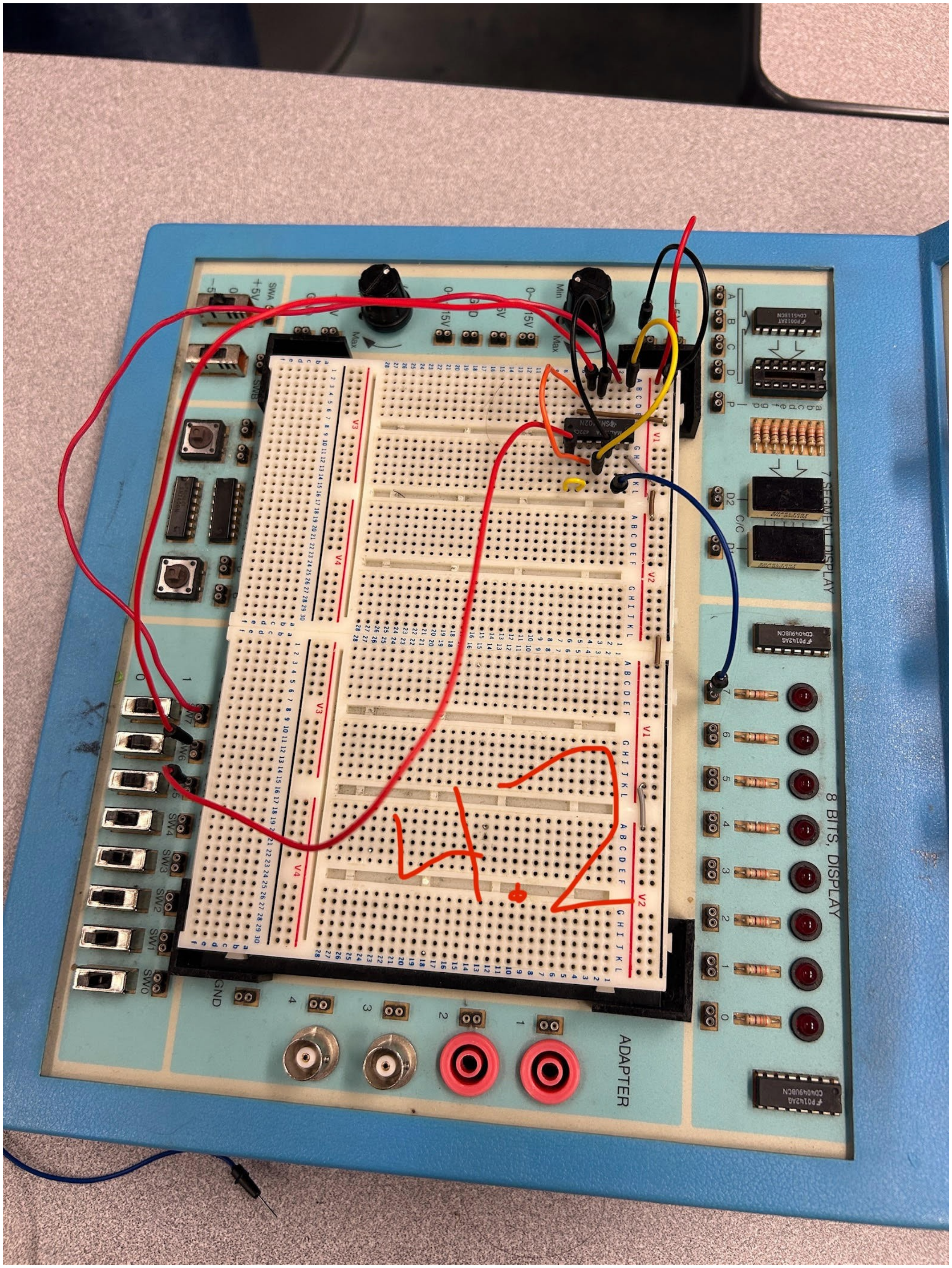
III. Truth Table

x1	x2	x3	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

7.2 NOR Equivalent of Product-of-Sums

I. NOR-only version of the Products-of-Sums circuit developed in Prelab 4.2.

II. Picture of breadboard with the completed circuit



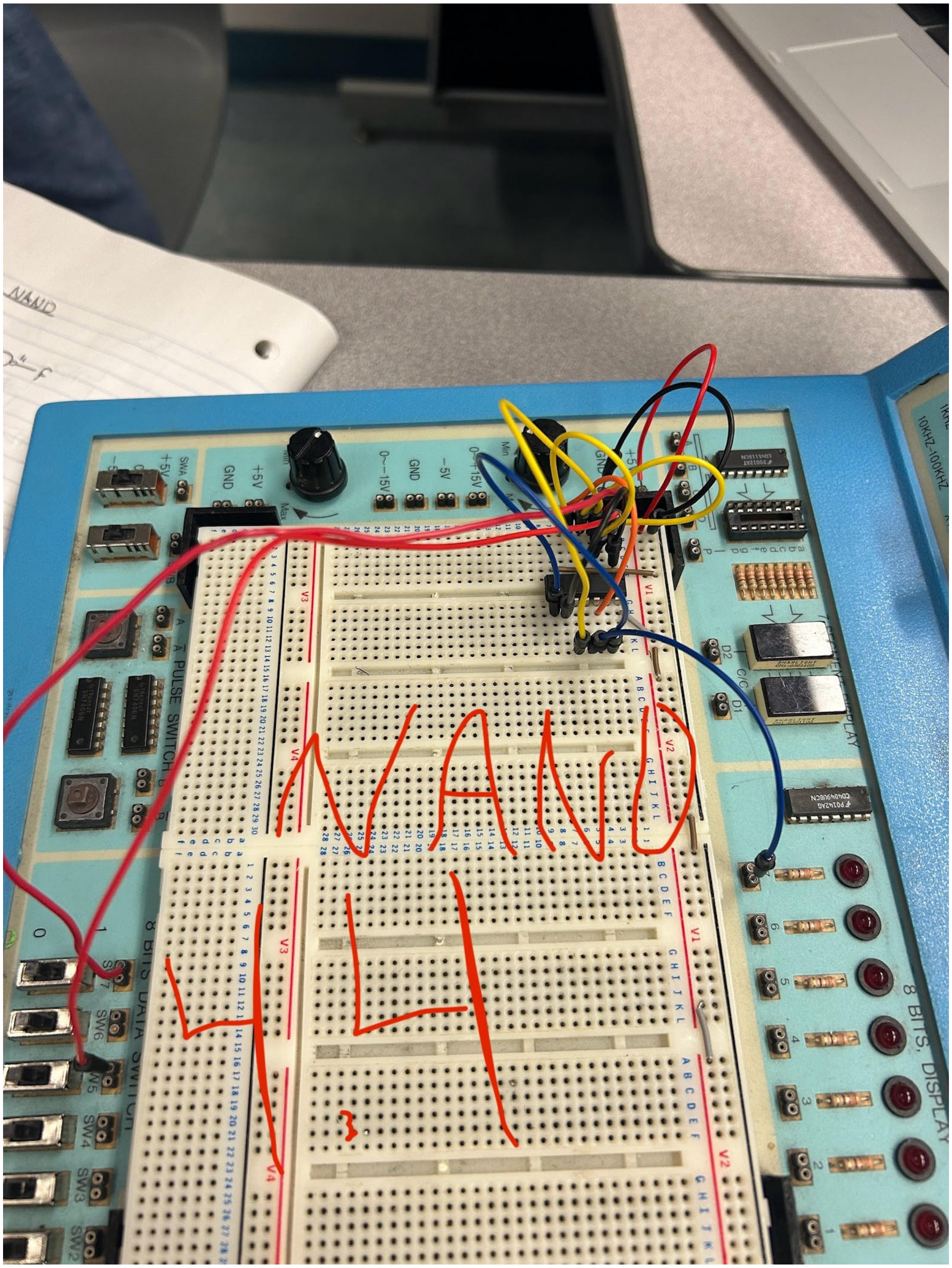
III. Truth Table

x1	x2	x3	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

7.3 XOR using NAND logics only

I. NAND-only version of the XOR equivalent circuit developed in Prelab 4.4.

II. Picture of breadboard with the completed circuit



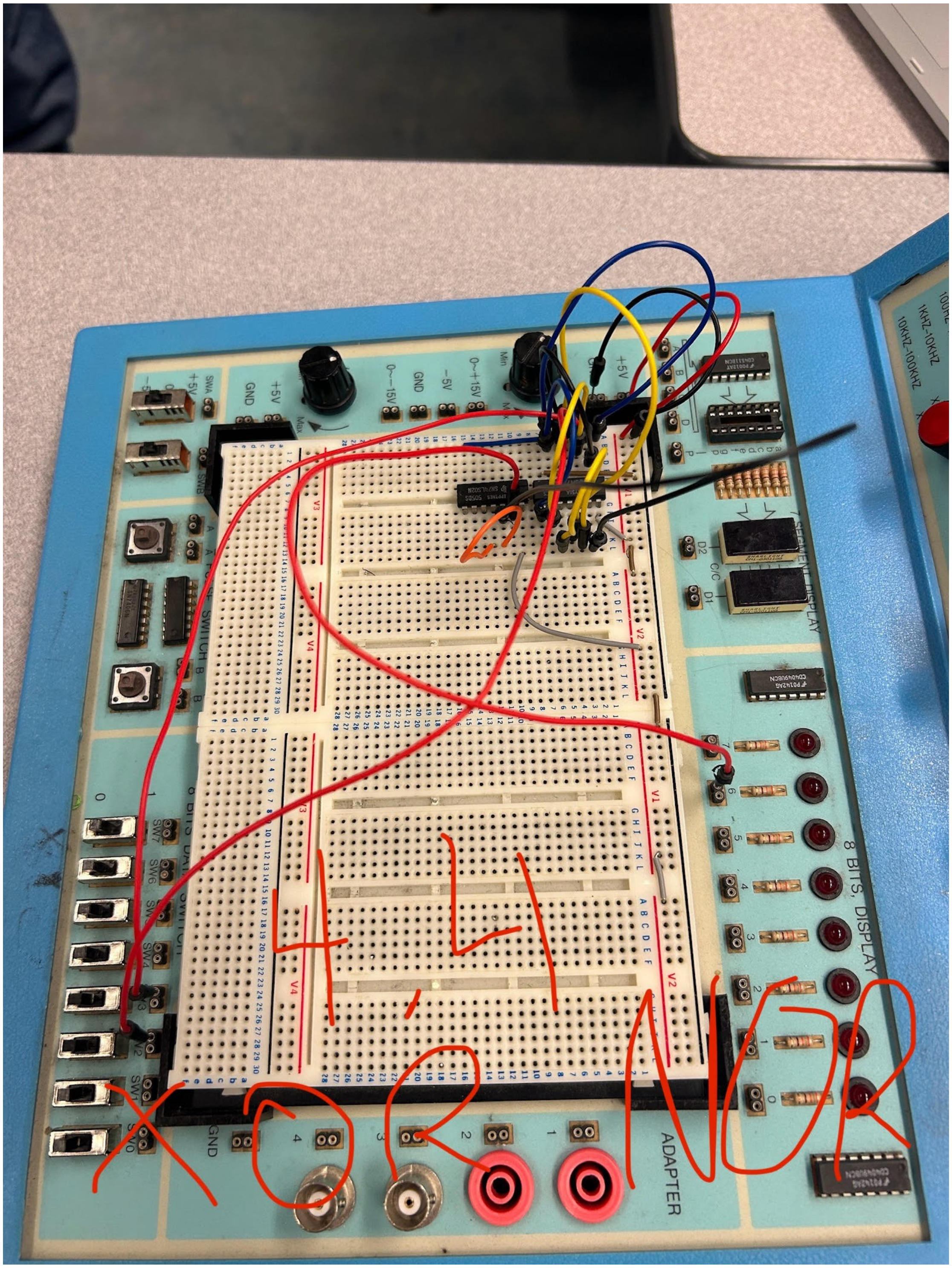
III. Truth Table

x1	x2	f
0	0	0
0	1	1
1	0	1
1	1	0

7.4 XOR using NOR logics only

I. NOR-only version of the XOR equivalent circuit developed in Prelab 4.4.

II. Picture of breadboard with the completed circuit



III. Truth Table

x1	x2	f
0	0	0
0	1	1
1	0	1
1	1	0

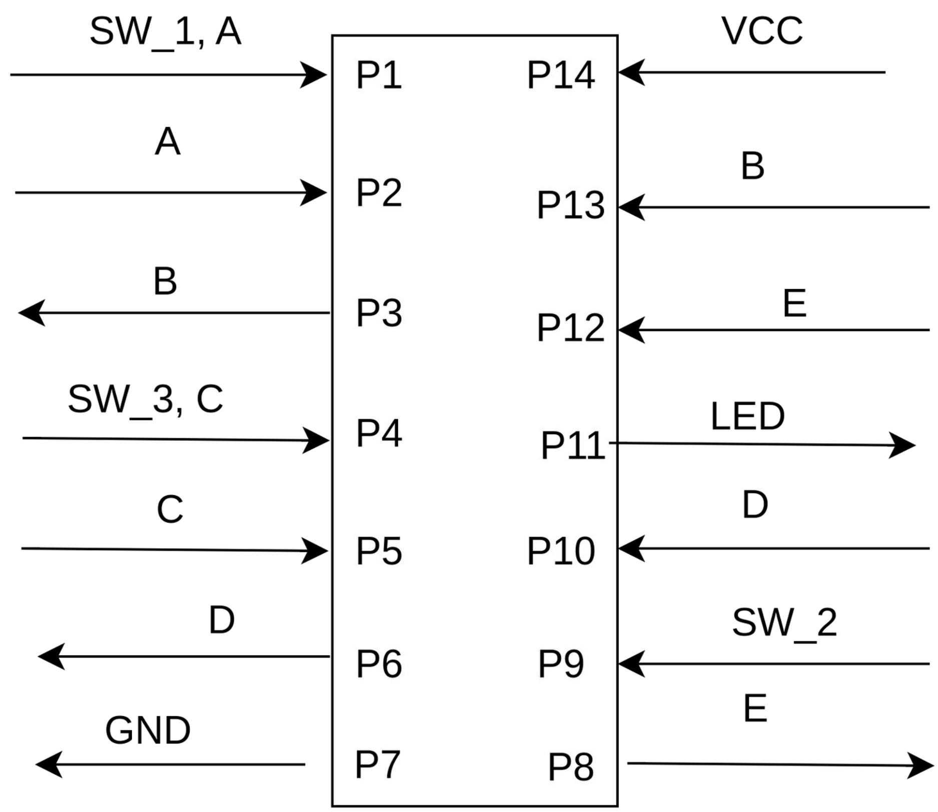
8. Conclusion

Both POS and SOP are valid methods of expressing truth tables. Personally I prefer POS as it is more intiutive solving for 1's then 0's.

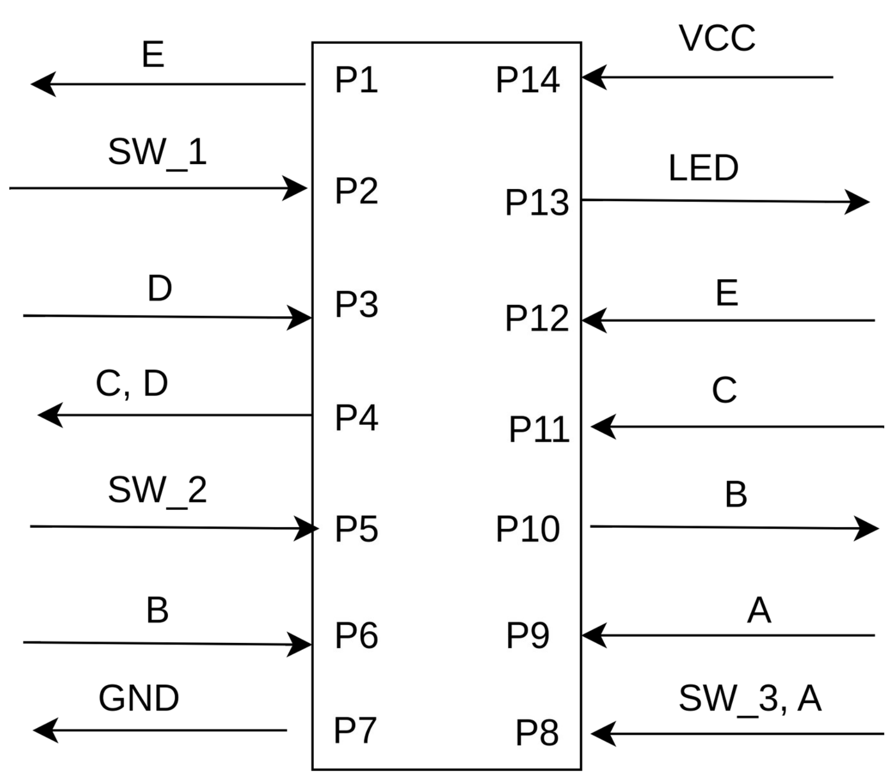
References

Appendix

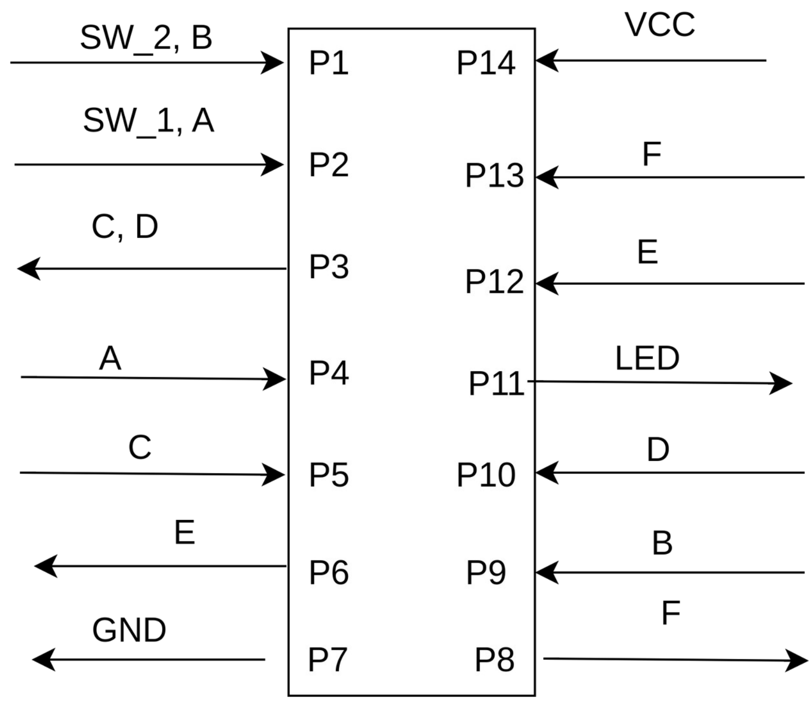
Functional Diagram 4.1



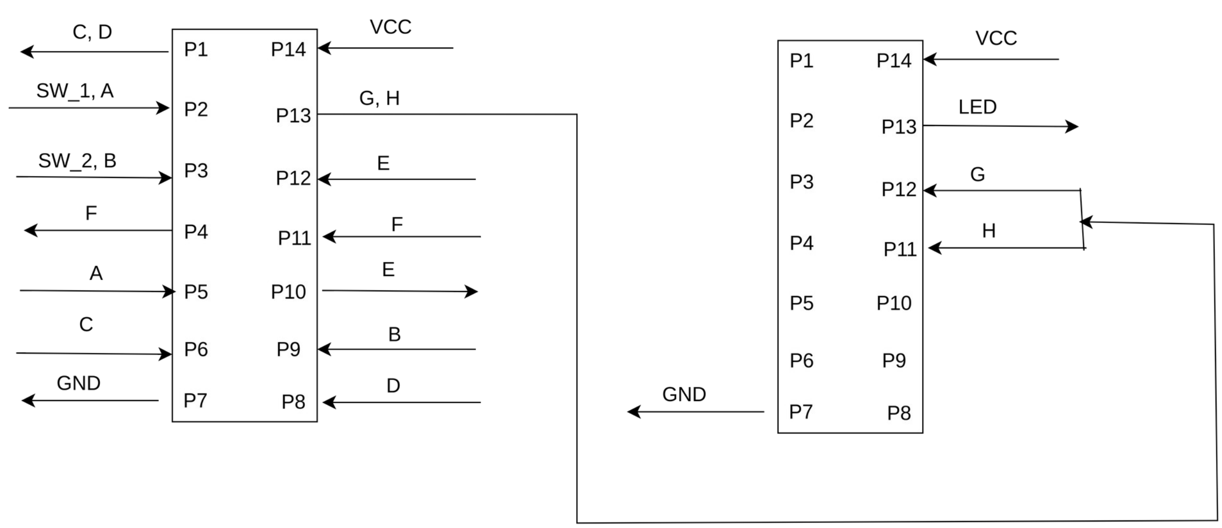
Functional Diagram 4.2



Functional Diagram 4.4 NAND



Functional Diagram 4.4 NOR



Signature: Caleb Burke