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

I. Familiarization with Full Adder

II. Familiarization with Parity Generator and Checker

III. Construct Full-Adder using NAND and XOR only logics

IV. Construct NAND equivalent for combinational circuit

V. Construct NOR equivalent for combinational circuit

2. Components Required

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

4. Pre-Lab

4.1 Boolean Function

4.1.1 Use algebraic manipulation to prove

I. x + yz = (x + y) * (x + z)

x + yz = (x + y)(x + z)
= x(x + y) + z(x + y)
= xx + xy + xz + yz
= x + xy + xz + yz (absorption)
= x + xz + yz (absorption)
= x + yz

ii. xy + yz + !xz = xy + !xz

xy + yz + !xz = xy + !xz
xy + !xz = xy + !xz (absorption)

iii. !(((a + b)+ !c + !d)*(!(ab)*!c*!d)) = !(a + b) * c * d + a * b + c + d

!(((a + b) + !c + !d) * (!(ab) * !c * !d)) = !(a + b) * cd + ab + c + d

RHS
!(a + b) * cd + ab + c + d

LHS

!((a + c) + (!c + !d)) + !(!(ab) * !c * !d)
!(a + b) * !(!c + !d) + !!(ab) + !!c + !!d
!(a + b) * c * d + ab + c + d

LHS == RHS

4.1.2 Use algebraic manipulation to find the minimum sum-of-products (SOP) expression for the function:

let a = x1, b = x2, c = x3, d = x4

f = a * !b * !c + abd + a * !b * c * !d
= a(!b!c + bd + !bc!d)
= a(!b!c + !bc!d + bd)
= a(!b(!c + c!d) + bd)
= a(!b(!c + !d) + bd)

i.e
x1 & (((!x2) & ((!x3) | (!x4))) | (x2 & x4))

f(x1, x2, x3) = sum (m(1, 3, 4, 6, 7))
= (!x1 * !x2 * x3) + (!x1 * x2 * x3) + (x1 * !x2 * !x3) + (x1 * x2 * !x3) + (x1 * x2 * x3)
= (!x1 * !x2 * x3) + (!x1 * x2 * x3) + (x1 * !x2 * !x3) + (x1 * x2)
= (!x1 * x3) + (x1 * !x2 * !x3) + (x1 * x2)
= (!x1 * x3) + x1 * (!x2 * !x3 + x2)
= !x1 * x3 + x1 * (x2 + !x3)

4.1.3 Use algebraic manipulation to find the simplest products-of-sums (POS) circuit that implements the function:

f(x1, x2, x3) = mul (M(0, 2, 5))
= (!x1 + !x2 + !x3) * (!x1 + x2 + !x3) * (x1 + !x2 + x3)
= (!x2 + (!x1 + !x3)) * (x2 + (!x1 + !x3)) * (x1 + !x2 + x3)]
= (!x1 + !x3) * (x1 + !x2 + x3)

4.1.4 Find the minimum-cost SOP and POS forms for the function

f(x1, x2, x3) = sum (m(1, 2, 3, 5))

SOP
m(1) = !x1 * !x2 * x3
m(2) = !x1 * x2 * !x3
m(3) = !x1 * x2 * x3
m(5) = x1 * !x2 * x3

	x2/x3			
x1	00	01	11	10
0	0	1	1	1
1	0	1	0	0

f(x1, x2, x3) = !x2 * x3 + !x1 * x2

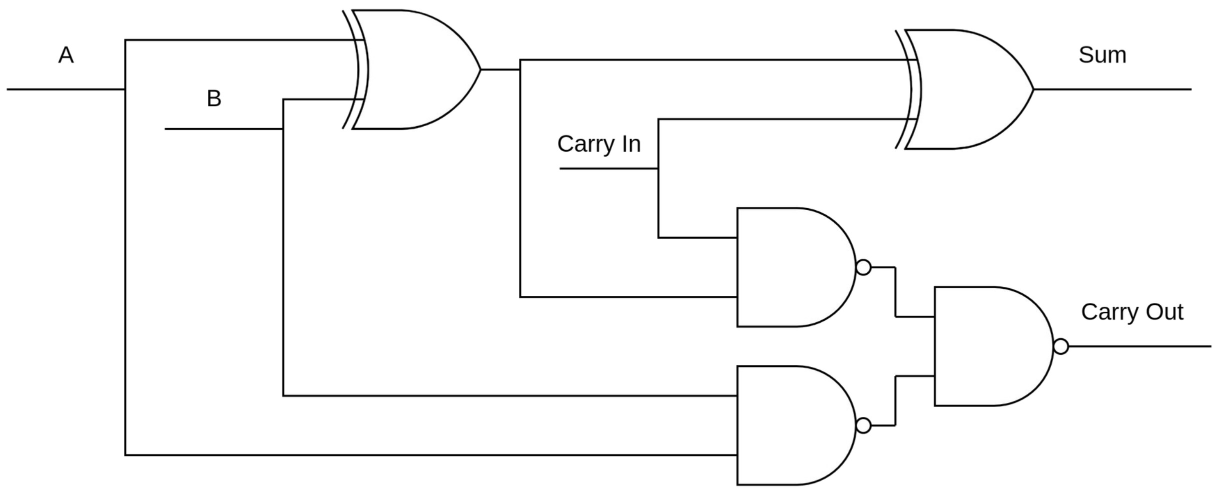
POS
M(0) = x1 + x2 + x3
M(4) = !x1 + x2 + x3
M(6) = !x1 + !x2 + x3
M(7) = !x1 + !x2 + !x3

	x2/x3			
x1	00	01	11	10
0	1	0	1	1
1	0	0	1	0

f(x1, x2, x3) = (x1 + !x2) * (!x2 + x3) * (x1 + x2 + !x3)

4.2 Full Adder Implementation

Just NAND and XOR



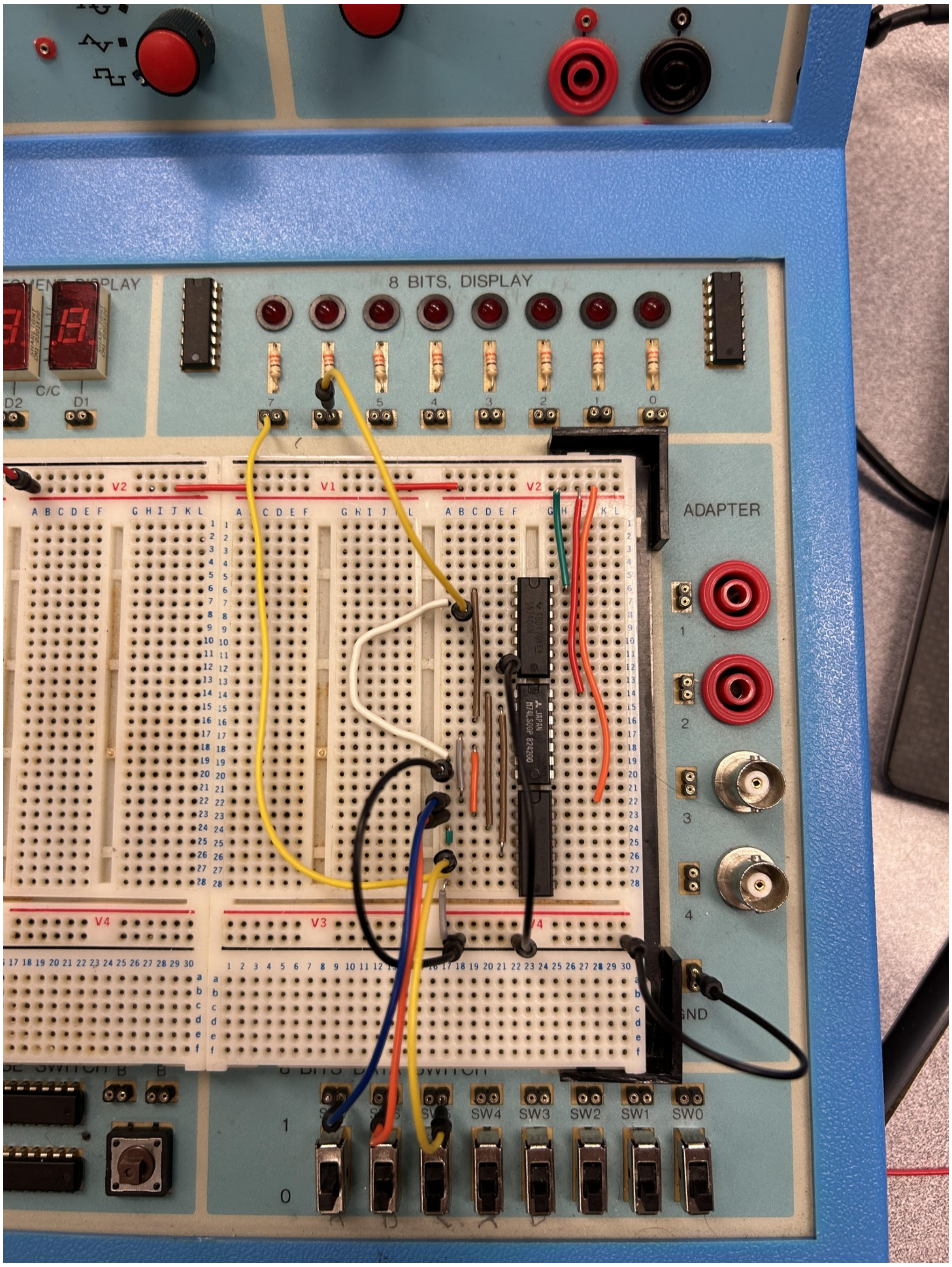
4.3 Parity Generator

The circuit show in 7.2 tells if the 5 input bits are a odd number of 1's. Return of 1 is odd return of 0 is even. Partiy bit is for error detection, if the number of 1's is even but parity is 1 some error has occurred. This integrity check works even if the partiy bit its self is corrupted. Note that the partiy bit only gives us info that some data corruption has occurred.

7. Lab Procedure with Deliverables

7.1 Full adder

Picture of circuit

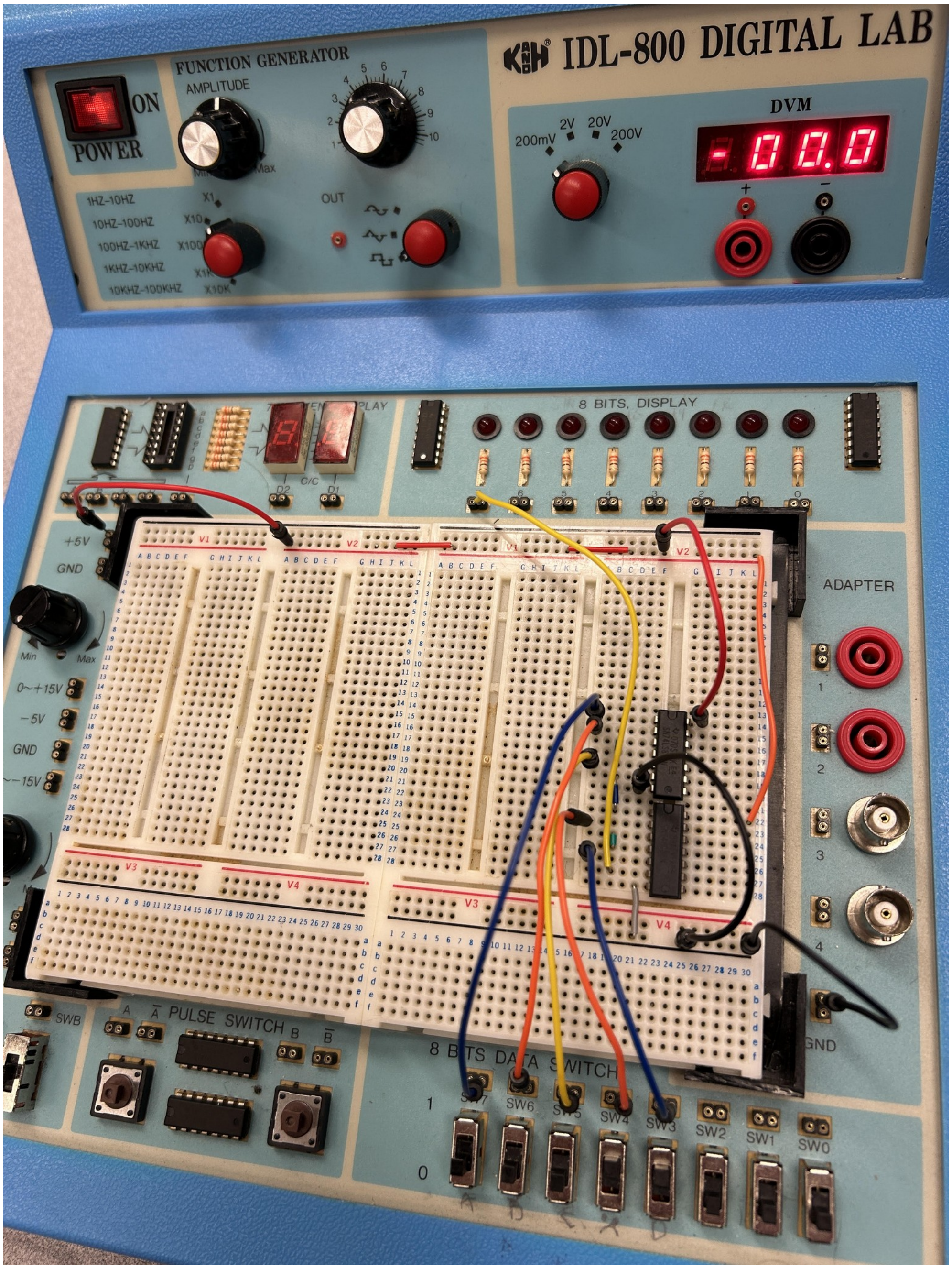


Truth Table

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

7.2 Parity Generator

Picture of circuit



Truth Table

A0	A1	A2	A3	A4	X
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	1
0	0	0	1	1	0
0	0	1	0	0	1
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	0	1	0
0	1	0	1	0	0
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	0	1	0
1	0	0	1	0	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	0	1	1

1	0	1	1	0	1
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	0	1	0
1	1	1	1	0	0
1	1	1	1	1	1

Appendixs

Signature: Caleb Burke