TABLE OF CONTENTS

1. INTRODUCTION…………………………………………………………………………………..2
2. DESIGN APPROACH……………………………………………………………………………..3
3. CONSTRUCTION…………………………………………………………………………………..6
4. CONCLUSION AND RESULTS………………………………………………………………10

INTRODUCTION

The premise of design project 1 is to design a circuit which will light up a seven segment LED display. The circuit is to be constructed entirely of NAND gates and inverters. The input to the circuit is a decimal digit encoded in 4311 format. The table below shows the proper inputs in 4311 format.

|  |  |
| --- | --- |
| DECIMAL | CODE |
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0011 |
| 3 | 0100 |
| 4 | 0101 |
| 5 | 1010 |
| 6 | 1011 |
| 7 | 1100 |
| 8 | 1110 |
| 9 | 1111 |

The 4 digit input will be labeled A, B, C, and D, each corresponding to a digit, left to right. Six of the sixteen possible inputs are not used. Thus they will be used to represent don’t care conditions. The LED display segment itself is triggered with signals. When it receives a signal of 0, it will light up and a 1 will results in unpowered segment. The segments are labeled in clockwise order, starting with T at the top and ending with Z in the middle. This project is limited to only 2 74HC00 NAND and 3 74HC04 inverter chips. This limits the amount of segments that can actually be programmed. Due to this, 3 segments were chosen to be implemented.DESIGN APPROACH

In order to find the most efficient way to implement the NAND gates and inverters, certain steps had to be performed. In order to discover the most efficient method, a truth table was created to show the inputs and the outputs. In the table below, X represents don’t care conditions. Notice that in the following table, an output of 1 represents a lit segment. In order to prevent confusion, the equations were derived from the original truth tables, and then inverted in the final implementation.

INPUT OUTPUT (0 = OFF, 1 = ON)

A B C D T U V W X Y Z

0 0 0 0 1 1 1 1 1 1 0

0 0 0 1 0 1 1 0 0 0 0

0 0 1 0 X X X X X X X

0 0 1 1 1 1 0 1 1 0 1

0 1 0 0 1 1 1 1 0 0 1

0 1 0 1 0 1 1 0 0 1 1

0 1 1 0 X X X X X X X

0 1 1 1 X X X X X X X

1 0 0 0 X X X X X X X

1 0 0 1 X X X X X X X

1 0 1 0 1 0 1 1 0 1 1

1 0 1 1 0 0 1 1 1 1 1

1 1 0 0 1 1 1 0 0 0 0

1 1 0 1 X X X X X X X

1 1 1 0 1 1 1 1 1 1 1

1 1 1 1 1 1 1 0 0 1 1

Using the truth table, the following Karnaugh maps were created to further help derive the equations.

KARNAUGH MAPS (0 = OFF, 1 = ON)

T

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| INPUT AB/CD | 00 | 01 | 11 | 10 |
| 00 | 1 | 0 | 1 | X |
| 01 | 1 | 0 | X | X |
| 11 | 1 | X | 1 | 1 |
| 10 | 1 | 1 | 0 | 1 |

SUM OF PRODUCTS = A’C’D + BC’D’ + A’B’CD + ABC + ACD

PRODUCT OF SUMS = (A + C + D’) (A’+B+C’+D’)

U

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| INPUT AB/CD | 00 | 01 | 11 | 10 |
| 00 | 1 | 1 | 1 | X |
| 01 | 1 | 1 | X | X |
| 11 | 1 | X | 1 | 1 |
| 10 | X | X | 0 | 0 |

SUM OF PRODUCTS = A’C’ + A’B’D + BC’D’ + ABC

PRODUCT OF SUMS = A’ + B + C’

V

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| INPUT AB/CD | 00 | 01 | 11 | 10 |
| 00 | 1 | 1 | 0 | X |
| 01 | 1 | 1 | X | X |
| 11 | 1 | X | 1 | 1 |
| 10 | X | X | 1 | 1 |
|  |  |  |  |  |

SUM OF PRODUCTS = A’C’ + BC’D’ + AC

PRODUCT OF SUMS = A + B + C’ + D’

W

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| INPUT AB/CD | 00 | 01 | 11 | 10 |
| 00 | 1 | 0 | 1 | X |
| 01 | 1 | 0 | X | X |
| 11 | 0 | X | 0 | 1 |
| 10 | X | X | 1 | 1 |

SUM OF PRODUCTS = A’C’D’ + B’CD + ACD’

PRODUCT OF SUMS = (A’ + B’ + C + D) (A + C + D’) ( A’ + B’ + C’ + D’)

X

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| INPUT AB/CD | 00 | 01 | 11 | 10 |
| 00 | 1 | 0 | 1 | X |
| 01 | 0 | 0 | X | X |
| 11 | 0 | X | 0 | 1 |
| 10 | X | X | 1 | 0 |

SUM OF PRODUCTS = A’B’C’D’ + B’CD + ABCD’

PRODUCT OF SUMS = (A + C + D’) (B ‘ + C + D) (A’ + B’ + C’ + D’) (A’ + B + C’ + D)

Y

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| INPUT AB/CD | 00 | 01 | 11 | 10 |
| 00 | 1 | 0 | 0 | X |
| 01 | 0 | 1 | X | X |
| 11 | 0 | X | 1 | 1 |
| 10 | X | X | 1 | 1 |

SUM OF PRODUCTS = A’B’C’D’ + A’BC’D + AC

PRODUCT OF SUMS = (B’ + C + D) (A +B + D’)

Z

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| INPUT AB/CD | 00 | 01 | 11 | 10 |
| 00 | 0 | 0 | 1 | X |
| 01 | 1 | 1 | X | X |
| 11 | 0 | X | 1 | 1 |
| 10 | X | X | 1 | 1 |

SUM OF PRODUCTS = AC + A’BC’ + A’B’CD

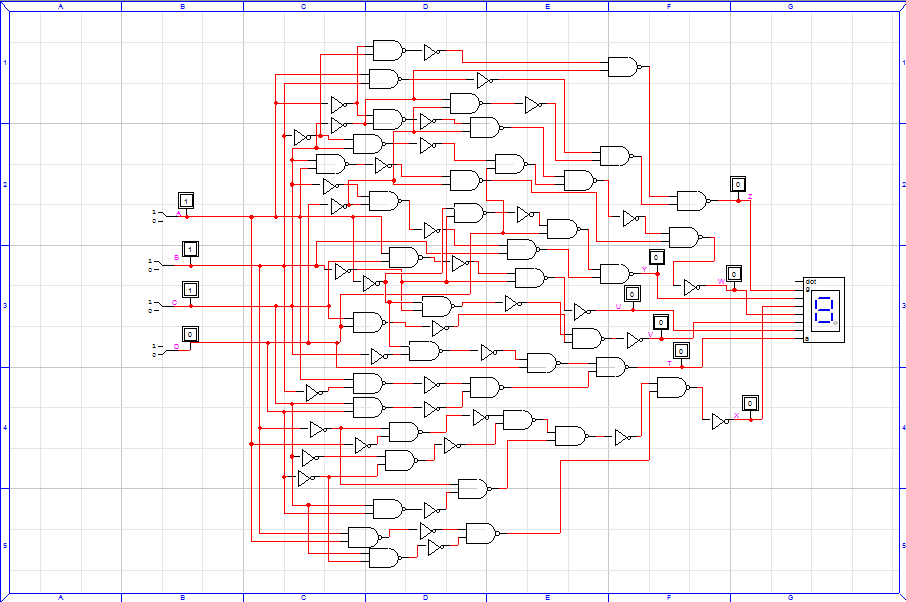
PRODUCT OF SUMS = (A + B + C) ( A’ + B’ + C + D)

By first grouping the 1’s in each K-map, I was able to derive the sum-of-products equation for each segment. The 0’s were then grouped for each segment to derive the product-of-sums (POS) equation. No X’s were grouped when the equations were derived.

After deriving the equations, I decided on segments U, V, and Y. The reason I chose this was because they had the simplest POS equations. Any of the SOP forms would have required too many gates to implement. The POS forms of U, V, and Y required the least amount of NAND gates and inverters.

CONSTRUCTION

The first step in implementing the circuit was to create it in Logicworks. The following gate level circuit diagram is an implementation of all 7 segments of the LED display



As shown, implementing all 7 segments of the display would have required a massive amount of inverters and NAND gates. The below is an input timing table that was used to simulate the running circuit.

$T $D $I A $I B $I C $I D

0 200 0 0 0 0

200 200 0 0 0 1

400 200 0 0 1 1

600 200 0 1 0 0

800 200 0 1 0 1

1000 200 1 0 1 0

1200 200 1 0 1 1

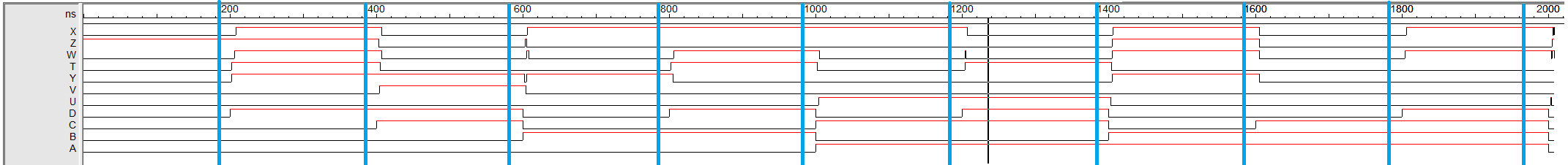
1400 200 1 1 0 0

1600 200 1 1 1 0

1800 200 1 1 1 1

2000 200 0 0 0 0

Running the previous timing inputs, the following timing diagram resulted:



0 1 2 3 4 5 6 7 8 9 0

The inputs were spaced every 200 nanoseconds, with the range of time going from zero to 2000 nanoseconds. The lines are placed right before the input was changed and the number under corresponds to the input number. Exporting the output table would give us exact results at each step and allow verification of proper operation. The exported output timing table is shown below. From it, you can see that there is a small propagation delay from step to step. Note that the circuits have been constructed to account for the LED display lighting up when receiving an input of 0. This is reflected in the timing table.

$T $D $O X $O Z $O W $O T $O Y $O V $O U $O D $O C $O B $O A

0 200NS 0 1 0 0 0 0 0 0 0 0 0

200NS 2NS 0 1 0 0 0 0 0 1 0 0 0

202NS 4NS 0 1 0 1 1 0 0 1 0 0 0

206NS 2NS 0 1 1 1 1 0 0 1 0 0 0

208NS 192NS 1 1 1 1 1 0 0 1 0 0 0

400NS 3NS 1 1 1 1 1 0 0 1 1 0 0

403NS 1NS 1 0 1 1 1 0 0 1 1 0 0

404NS 1NS 1 0 1 1 1 1 0 1 1 0 0

405NS 2NS 1 0 1 0 1 1 0 1 1 0 0

407NS 193NS 0 0 0 0 1 1 0 1 1 0 0

600NS 2NS 0 0 0 0 1 1 0 0 0 1 0

602NS 1NS 0 0 0 0 0 1 0 0 0 1 0

603NS 1NS 0 1 0 0 0 1 0 0 0 1 0

604NS 1NS 0 1 0 0 0 0 0 0 0 1 0

605NS 1NS 0 0 1 0 1 0 0 0 0 1 0

606NS 2NS 1 0 1 0 1 0 0 0 0 1 0

608NS 192NS 1 0 0 0 1 0 0 0 0 1 0

800NS 2NS 1 0 0 0 1 0 0 1 0 1 0

802NS 3NS 1 0 0 1 1 0 0 1 0 1 0

805NS 1NS 1 0 0 1 0 0 0 1 0 1 0

806NS 194NS 1 0 1 1 0 0 0 1 0 1 0

1US 2NS 1 0 1 1 0 0 0 0 1 0 1

1002NS 2NS 1 0 1 0 0 0 0 0 1 0 1

1004NS 1NS 1 0 1 0 0 0 1 0 1 0 1

1005NS 195NS 1 0 0 0 0 0 1 0 1 0 1

1200NS 4NS 1 0 0 0 0 0 1 1 1 0 1

1204NS 1NS 1 0 1 1 0 0 1 1 1 0 1

1205NS 2NS 1 0 0 1 0 0 1 1 1 0 1

1207NS 193NS 0 0 0 1 0 0 1 1 1 0 1

1400NS 3NS 0 0 0 1 0 0 1 0 0 1 1

1403NS 1NS 0 0 0 1 0 0 0 0 0 1 1

1404NS 1NS 0 0 0 0 0 0 0 0 0 1 1

1405NS 1NS 0 1 1 0 1 0 0 0 0 1 1

1406NS 194NS 1 1 1 0 1 0 0 0 0 1 1

1600NS 5NS 1 1 1 0 1 0 0 0 1 1 1

1605NS 195NS 0 0 0 0 0 0 0 0 1 1 1

1800NS 4NS 0 0 0 0 0 0 0 1 1 1 1

1804NS 2NS 0 0 1 0 0 0 0 1 1 1 1

1806NS 194NS 1 0 1 0 0 0 0 1 1 1 1

2US 3NS 1 0 1 0 0 0 0 0 0 0 0

2003NS 1NS 1 0 1 0 0 0 1 0 0 0 0

2004NS 1NS 1 0 0 0 0 0 0 0 0 0 0

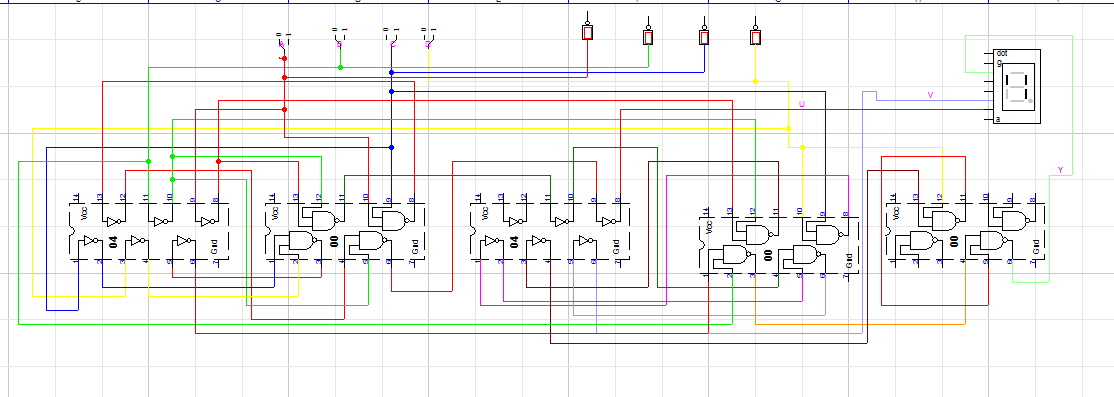
2005NS 1NS 1 1 1 0 0 0 0 0 0 0 0

2006NS 1NS 0 1 1 0 0 0 0 0 0 0 0

2007NS 1NS 1 1 1 0 0 0 0 0 0 0 0

2008NS 0 0 1 0 0 0 0 0 0 0 0 0

Following the construction of the gate level diagram, the chip level diagram below was created in order to facilitate the actual construction of the board.



Due to the frequent use of inverters, the best way to create the circuit was in the order shown above. An inverter chip would be first, followed by a NAND gate chip, the second inverter, and the two remaining NAND gate chips. The chip level circuit was then simulated in order to verify results.

TIMING INPUT FILE

$T $D $I A $I B $I C $I D

0 200 0 0 0 0

200 200 0 0 0 1

400 200 0 0 1 1

600 200 0 1 0 0

800 200 0 1 0 1

1000 200 1 0 1 0

1200 200 1 0 1 1

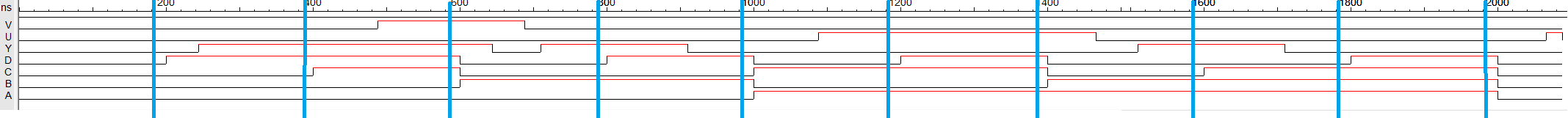
1400 200 1 1 0 0

1600 200 1 1 1 0

1800 200 1 1 1 1

2000 200 0 0 0 0

The following timing diagram resulted. As before, the blue line marks right before an input switch and the numbers correspond to the input. The exported output timing table is also displayed to break down the event at each step. In this circuit, there is also a small propagation delay.



0 1 2 3 4 5 6 7 8 9 0

EXPORTED TIMING TABLE

$T $D $O V $O U $O Y $O D $O C $O B $O A

0 200NS 0 0 0 0 0 0 0

200NS 44NS 0 0 0 1 0 0 0

244NS 156NS 0 0 1 1 0 0 0

400NS 88NS 0 0 1 1 1 0 0

488NS 112NS 1 0 1 1 1 0 0

600NS 44NS 1 0 1 0 0 1 0

644NS 44NS 1 0 0 0 0 1 0

688NS 22NS 0 0 0 0 0 1 0

710NS 90NS 0 0 1 0 0 1 0

800NS 110NS 0 0 1 1 0 1 0

910NS 90NS 0 0 0 1 0 1 0

1US 88NS 0 0 0 0 1 0 1

1088NS 112NS 0 1 0 0 1 0 1

1200NS 200NS 0 1 0 1 1 0 1

1400NS 66NS 0 1 0 0 0 1 1

1466NS 44NS 0 0 0 0 0 1 1

1510NS 90NS 0 0 1 0 0 1 1

1600NS 110NS 0 0 1 0 1 1 1

1710NS 90NS 0 0 0 0 1 1 1

1800NS 200NS 0 0 0 1 1 1 1

2US 66NS 0 0 0 0 0 0 0

2066NS 22NS 0 1 0 0 0 0 0

2088NS 0 0 0 0 0 0 0 0

CONCLUSION AND RESULTS

Both timing simulations resulted in correct outputs, thus verifying the correct operation of my circuit. The last step was to actually implement the NAND gates and inverters as shown in the chip level diagram. The actual implementation proved to be very difficult. With limited and cramped amounts of space, finding the correct place to place wires seemed difficult at times. Backtracking after making a mistake also proved a challenge due to the large amount of wires. However, I was able to eventually correctly create the circuit.

In conclusion, I believe that this design project served as a good learning experience. The limitation on the number of chips that could be used reflected a real life situation, in which one has to minimize resources required for a project.