Aleksas Murauskas

ECSE 429

Assignment 1

Question 1 Part 1

Tabular Method

First Solve initial Finish state with no faults

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Inputs  A B C | AND 1 | XNOR 1  AND1 B | XNOR 2  And1,XNOR1 | NOT 1  XNOR1 | Not 2  XNOR2 | NAND1 [X]  And1,XNOR2 | XNOR3  Not2 B | OR1 [y]  Not2,not1 | NAND2 [Z]  XNOR1,XNOR3,C |
| 0 0 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 0 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 1 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 1 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 1 0 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 0 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 1 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 1 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |

Table Mutant on line e s-a-1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Inputs  A B C | AND 1 | XNOR 1  AND1 B | XNOR 2  And1,XNOR1 | NOT 1  XNOR1 | Not 2  XNOR2 | NAND1 [X]  And1,XNOR2 | XNOR3  1,B | OR1 [y]  1,not1 | NAND2 [Z]  XNOR1,XNOR3,C |
| 0 0 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 0 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 1 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 1 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 0 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 0 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 1 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 1 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |

Table with mutant on line g s-a-0

Assumption: Now the line g inputs for XNOR 2, Not 1, and NAND are stuck at 0

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Inputs  A B C | AND1 | XNOR 1  AND1 B | XNOR 2  And1,0 | NOT1  0 | Not 2  XNOR2 | NAND1 [X]  And1,XNOR2 | XNOR3  Not2 B | OR1 [y]  Not2,not1 | NAND2 [Z]  0,XNOR3,C |
| 0 0 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 0 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 1 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 1 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 1 0 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 0 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 1 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 1 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |

Final Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input  A B C | F(A,B,C)  X Y Z | X Y Z | )  X Y Z |  |
| 0 0 0 | 1 1 1 | 1 1 1 | 1 1 1 |  |
| 0 0 1 | 1 1 1 | 1 1 1 | 1 1 1 |  |
| 0 1 0 | 1 1 1 | 1 1 1 | 1 1 1 |  |
| 0 1 1 | 1 1 1 | 1 1 1 | 1 1 1 |  |
| 1 0 0 | 1 1 1 | 1 1 1 | 1 1 1 |  |
| 1 0 1 | 1 1 1 | 1 1 1 | 1 1 1 |  |
| 1 1 0 | 0 0 1 | 0 1 1 | 1 1 1 |  |
| 1 1 1 | 0 0 1 | 0 1 0 | 1 1 1 |  |

Test Vectors detecting mutant line e s-a-1: 110,110

Test Vectors detecting mutant line g s-a-0: 110,111

Question 1 Part 2

Solving Through Boolean difference

Boolean solutions for X Y and Z

Solving for line e s-a 1

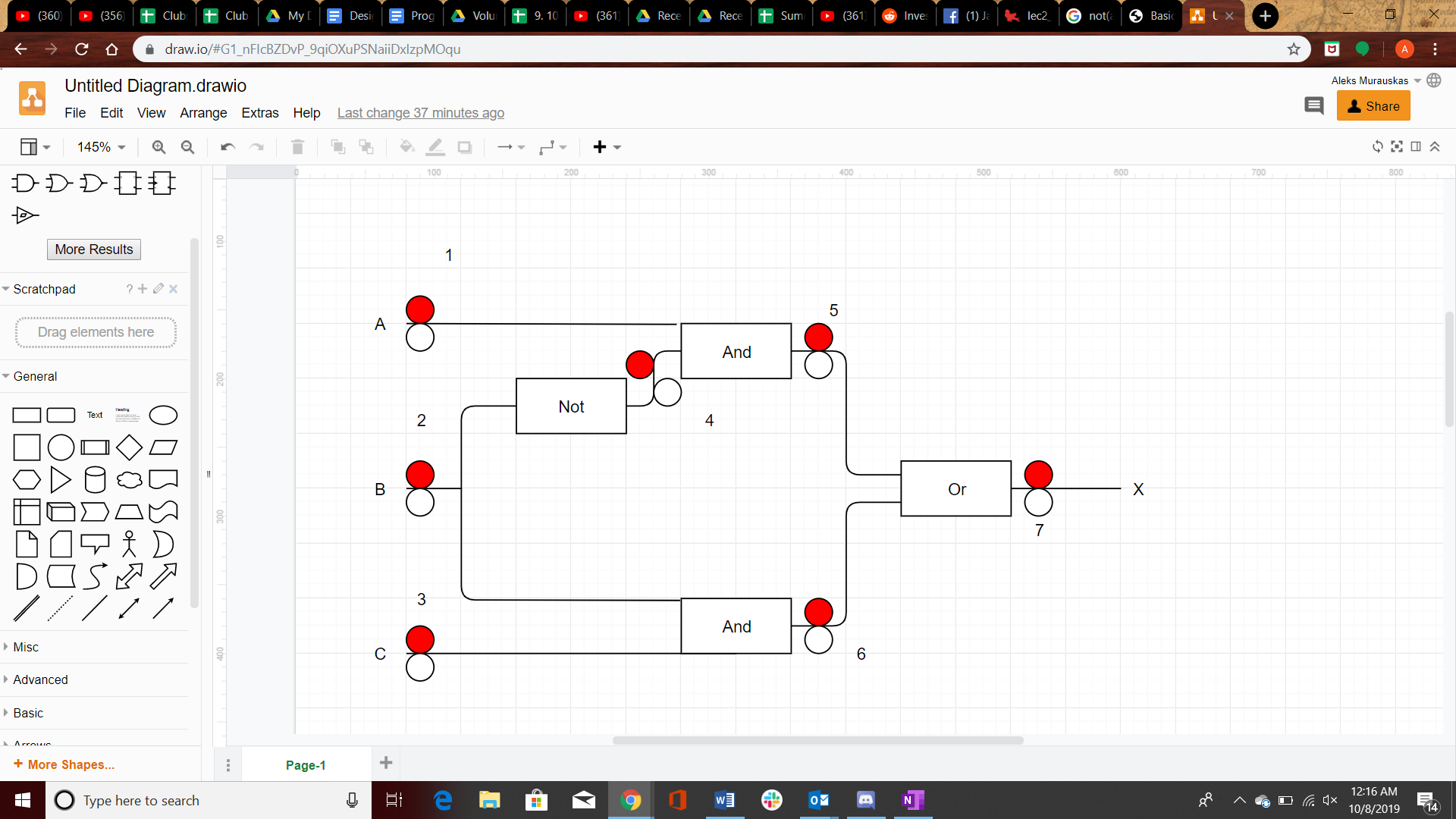
XOR

Solving for line g s-a 0

XOR

Question 2

Simplified circuit- Stuck ats affect the whole wire.



Boolean Equation:

At line 1

* As0:
* As1:

At line 2

* As0:
* As1:

At line 3

* As0:
* As1:

At line 4

* As0:
* As1:

At line 5

* As0:
* As1:

At line 6

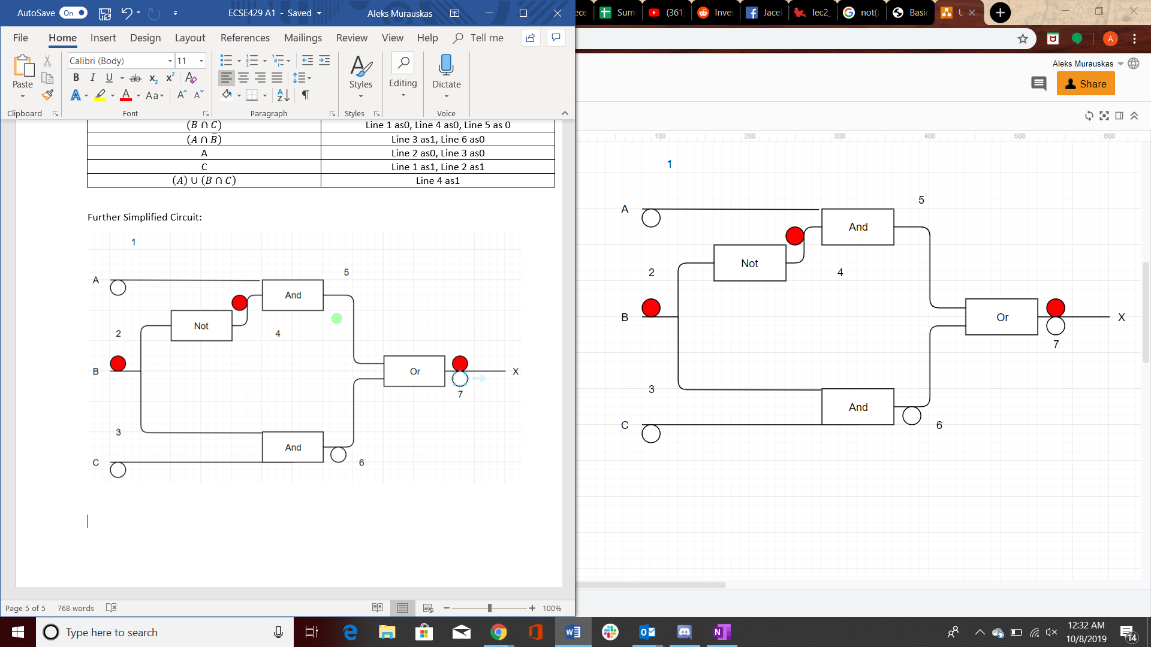
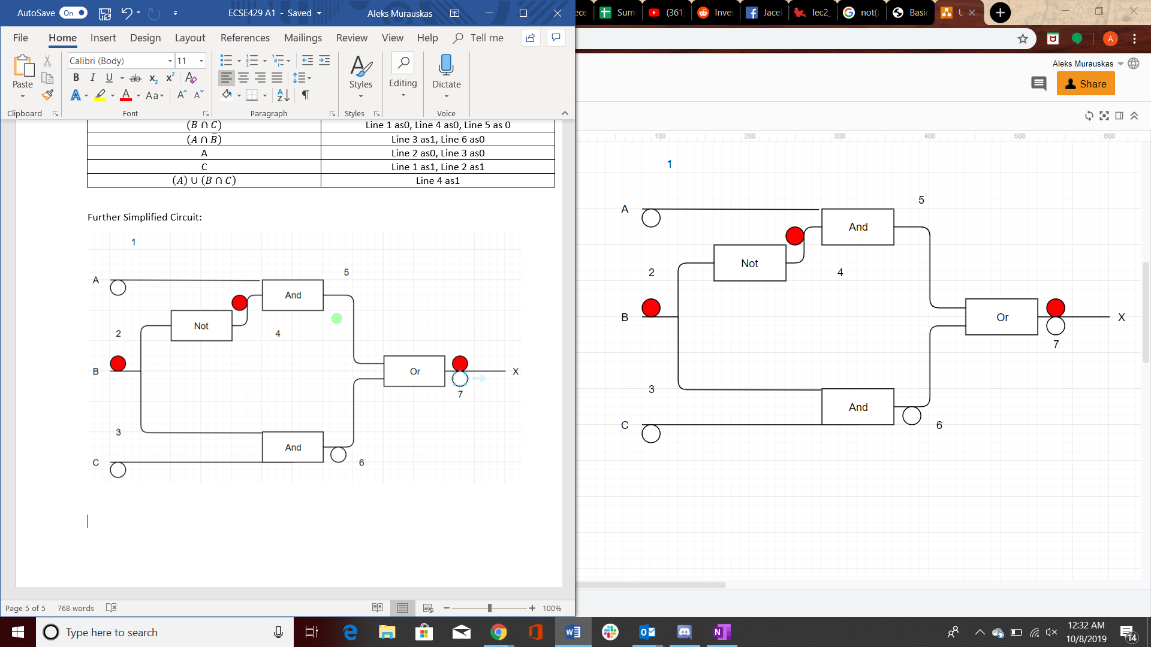
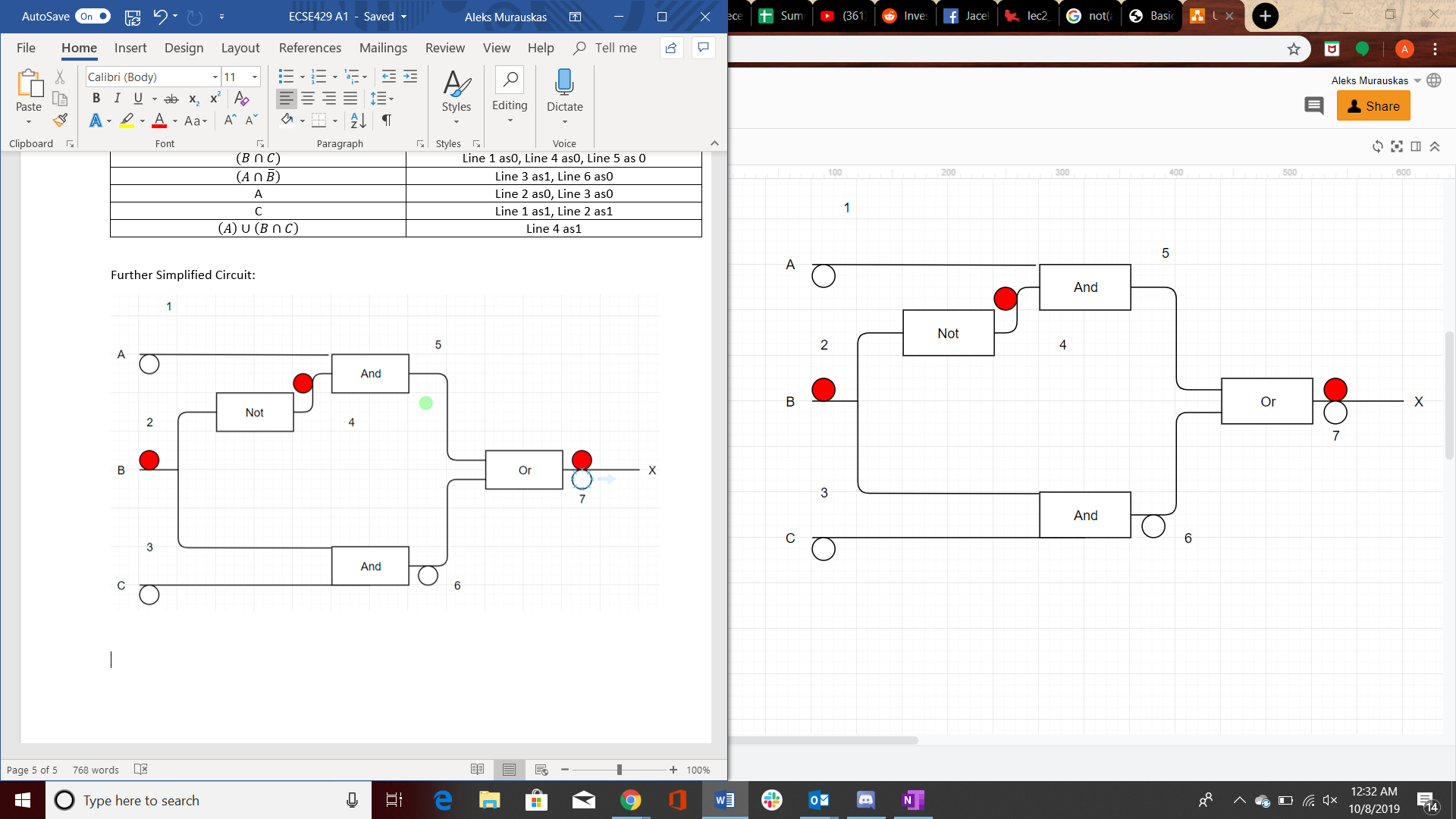
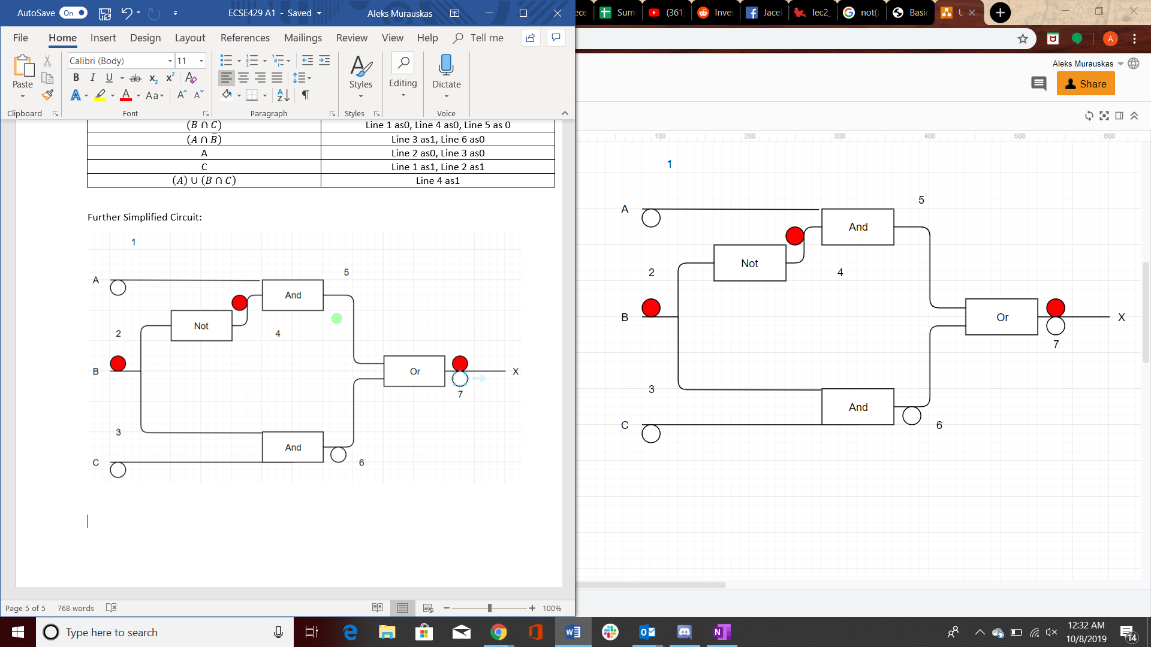
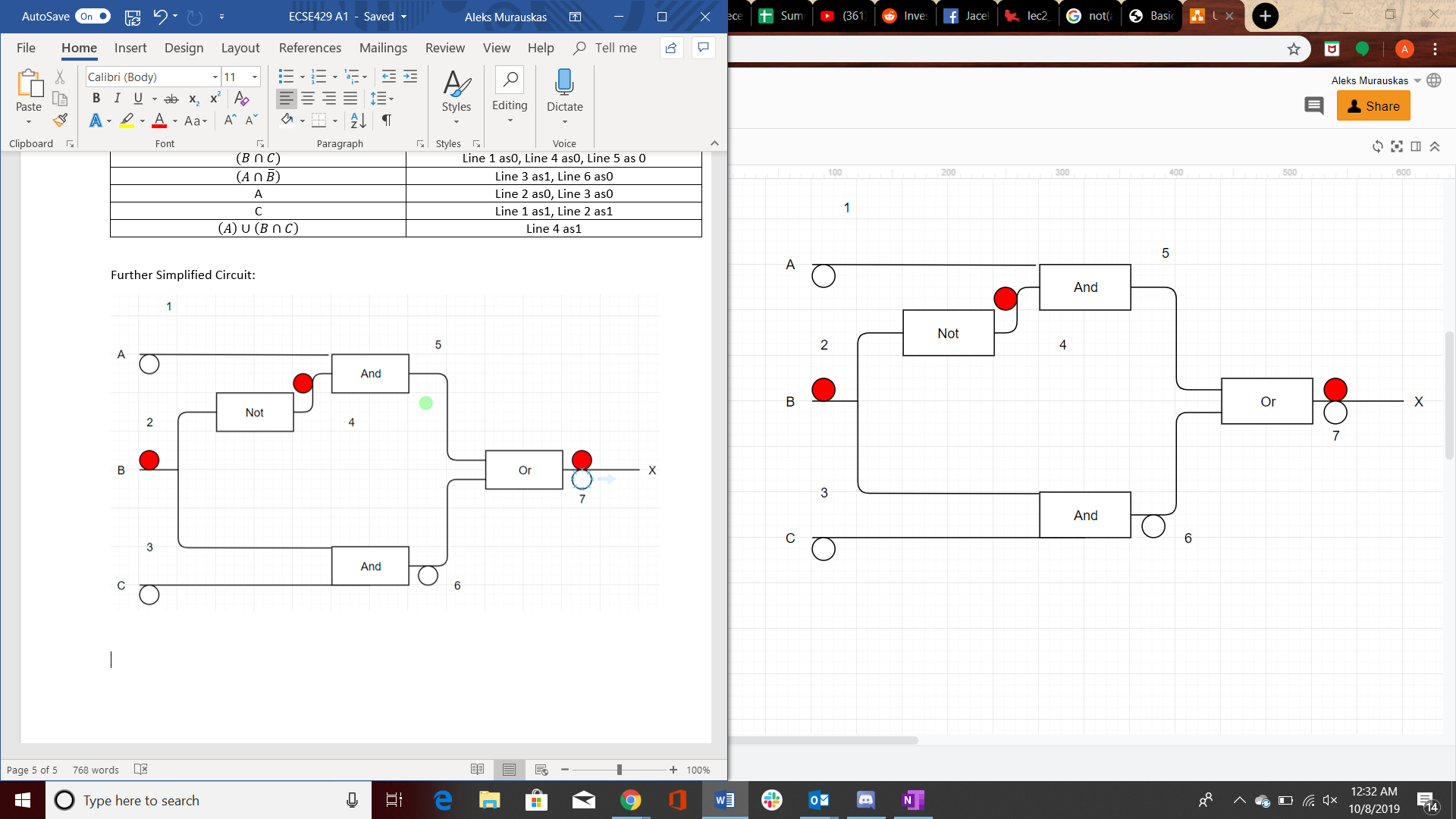
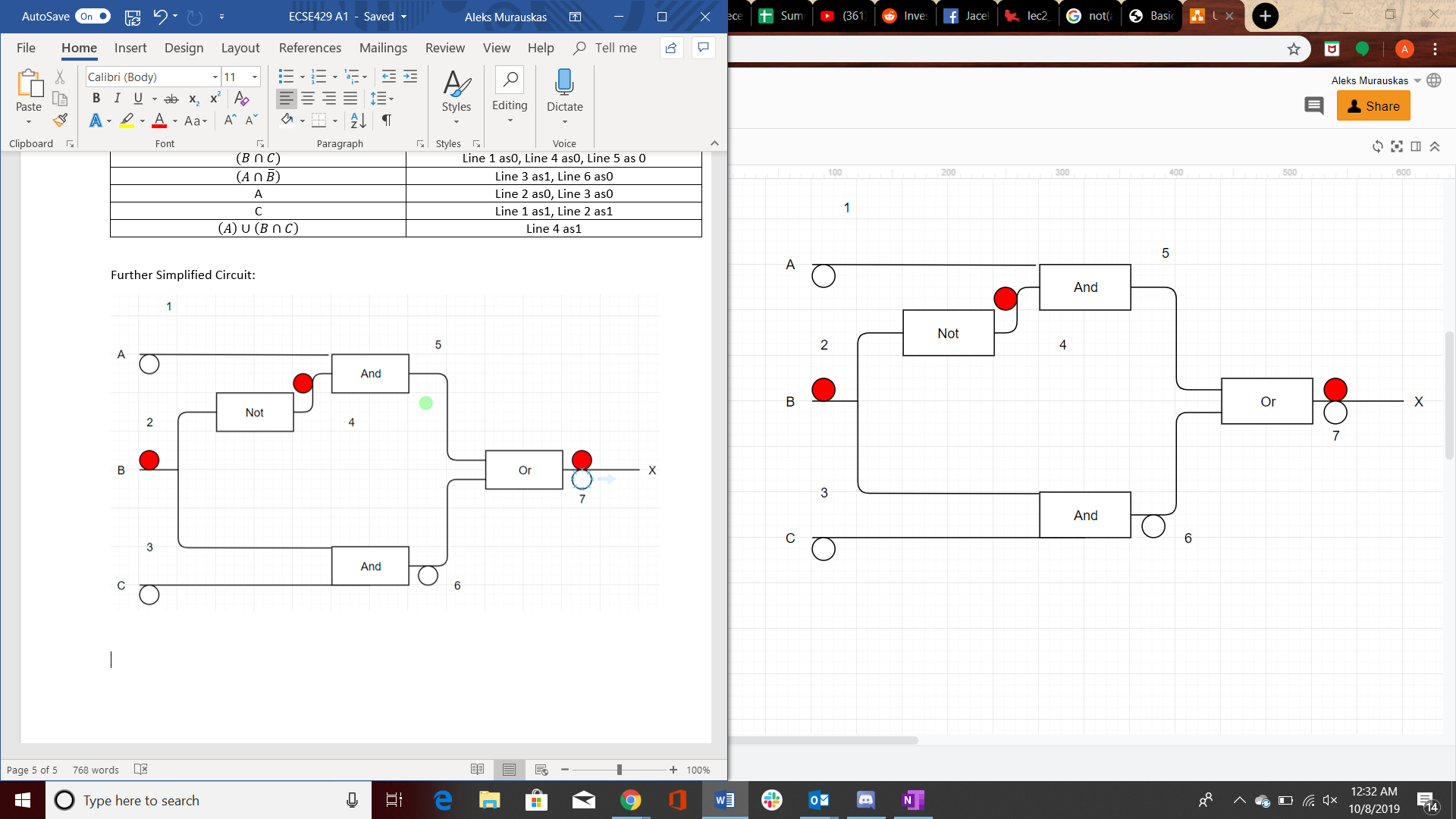
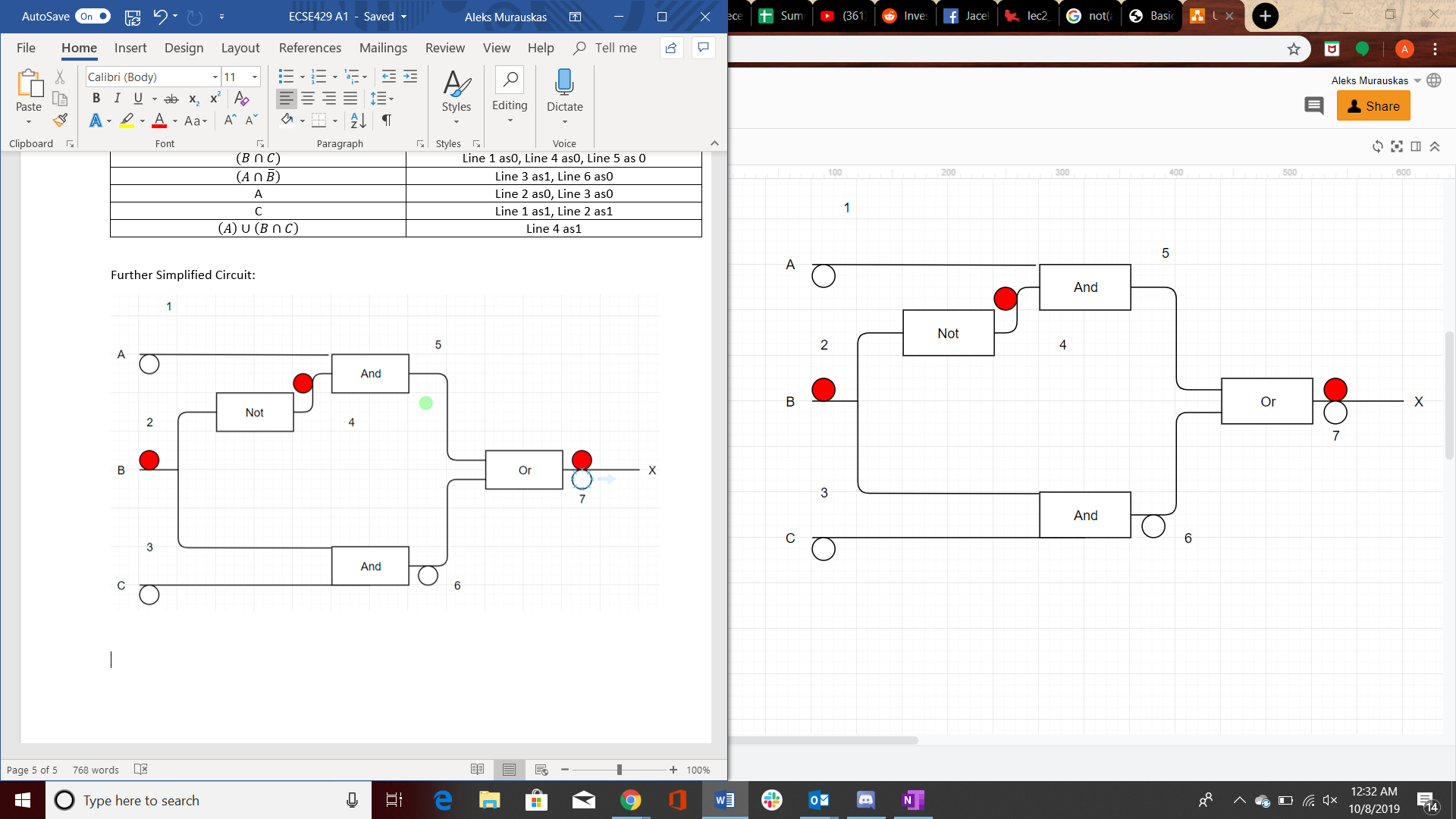
* As0:
* As1:

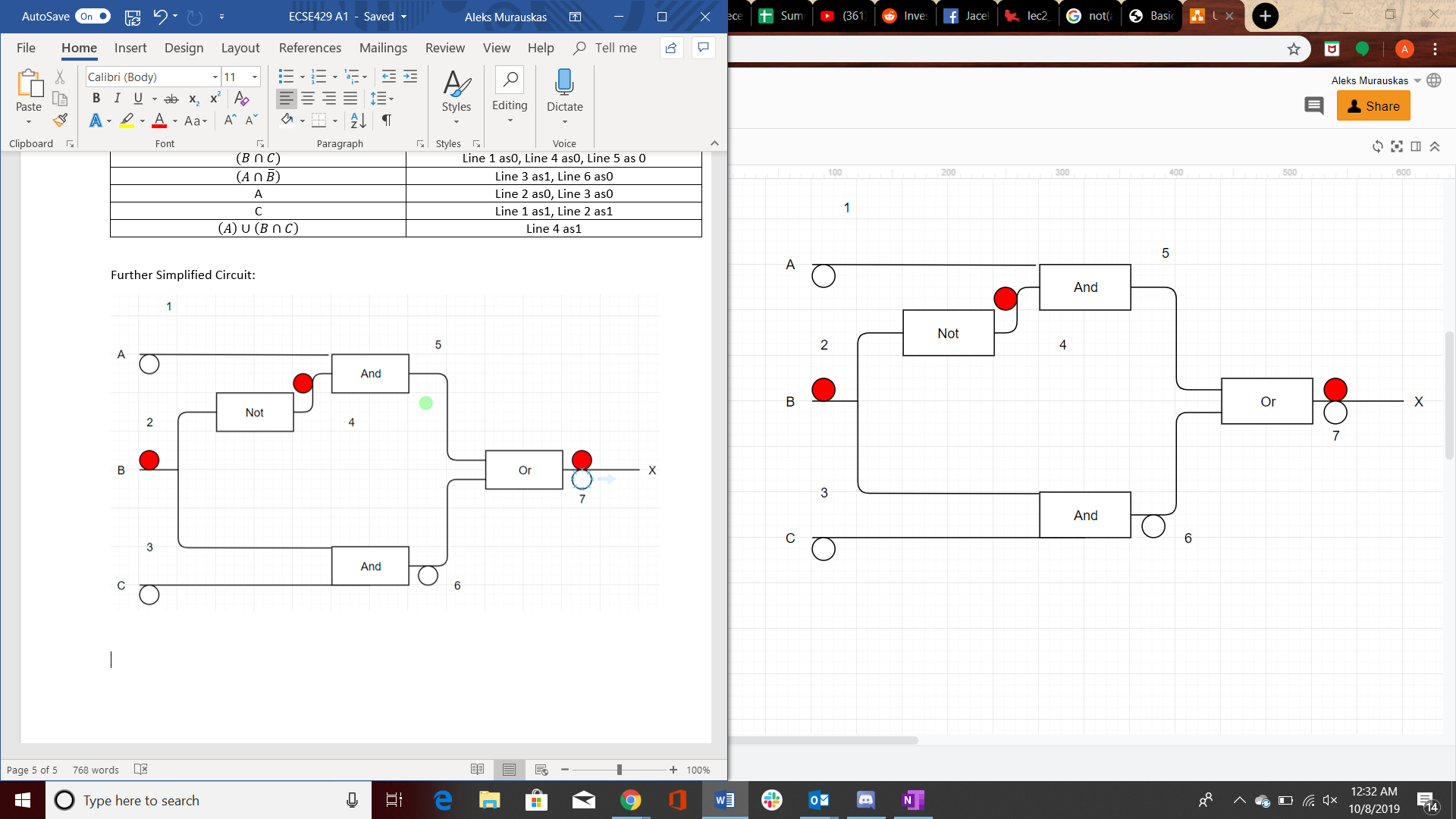
At line 7

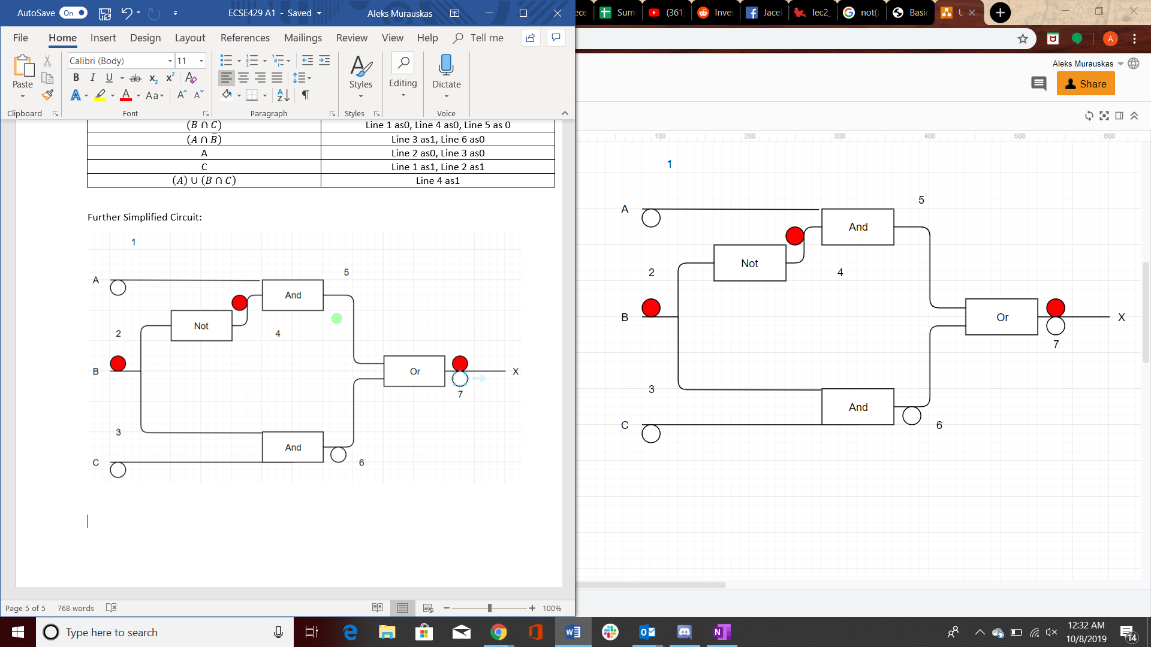
* As0:
* As1:

Equivalent faults:

|  |  |
| --- | --- |
| 0 | Line7 as0 |
| 1 | Line7 as1, Line 6 as1, Line 5 as1 |
|  | Line 1 as0, Line 4 as0, Line 5 as 0 |
|  | Line 1 a1 |
|  | Line 6 as0, Line 3 as0 |
| A | Line 2 as0 |
| C | Line 2 as1 |
|  | Line 4 as1 |
|  | Line 3 a1 |

Further Simplified Circuit:





Question 3:

CODE FRAGMENT

1. Public static int findVal(int numbers[], int val){
2. Int findval=-1;
3. For(inti=0,i<numbers.length,i++) //for(int i=(0+1);i<numbers.lenght;i++)
4. If (numbers[i] == val)
5. findVal=I;
6. return (findVal);
7. }

The mutant is i=1

1. Is it Possible to find an input that does not reach the mutant?
   1. Since the loop initializer [int I =0] is always executed and a null pointer exception would not be thrown until afterwards, there is no input that does not reach the mutant
2. Is it Possible to find an input that satisfies reachability but not infection for the mutant
   1. Since the declaration of the loop variable is independent of the input, no input is reaches the fault but is not infected
3. If possible, find a test input that satisfies infection, but not propagation for the mutant
   1. If the first occurrence of val in numbers is in position 1 or later, then the mutant infects it but does not propagated. [1,2,3,4,5], 3
4. If Possible find a test input that kills mutant m
   1. If the first value of numbers is intentionally not the value you search for, the mutant will never affect the result of findval [-1,2,3,4,5], 3

Question 4:

See attached files