

**Digital Logic Lab Assignment # 5 & 6**

5. To verify the universality of NAND gate and NOR gate.

6. To verify 2 and 3 variable De-Morgan’s Law.

**Submitted By**

Gaurav Chaulagain

Year I / SEM I

017BSCIT014

**Submitted To**

Signature Remarks

Er. Rajan Karmacharya

Lecturer, Dept. of Computer Science

Er. Saugat Sigdel

Lecturer, Dept. of Computer Science

**OBJECTIVE 5.1:**

**TO VERIFY THE UNIVERSALITY OF NAND GATE.**

**THEORY:** The NAND gate is a combination of the AND gate with that of an inverter or NOT gate. Here in this gate the output is opposite of the AND gate and output of AND gate is connected to the input of NOT gate.

**Boolean Expression: F=(A.B)’**

**5.1.1 : NOT GATE USING NAND GATE:**

**CIRCUIT DIAGRAM:**



**TRUTH TABLE:**

|  |  |
| --- | --- |
| **INPUT** | **OUTPUT** |
| **A** | **A’** |
| 0 | 1 |
| 1 | 0 |

**OBSERVATIONS:**

****

**OBSERVATION TABLE:**

|  |  |
| --- | --- |
| **INPUT** | **OUTPUT** |
| **A** | **A’** |
| 0 | 1 |
| 1 | 0 |

**CONCLUSION:**

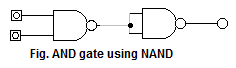
Hence, NAND gate was used as NOT gate.

**REFERENCE:**

http://www.electronics-tutorials.ws/logic/logic\_2.html

**5.1.2 : AND GATE USING NAND GATE:**

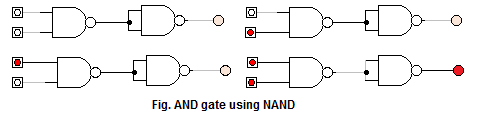
**CIRCUIT DIAGRAM:**

****

**TRUTH TABLE:**

|  |  |  |
| --- | --- | --- |
| **INPUTS** | | **OUTPUT** |
| **A** | **B** | **A.B** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**OBSERVATIONS:**



**OBSERVATION TABLE:**

|  |  |  |
| --- | --- | --- |
| **INPUTS** | | **OUTPUT** |
| **A** | **B** | **A.B** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**CONCLUSION:**

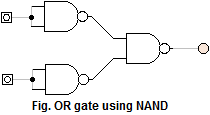
Hence, NAND gate was used as AND gate.

**REFERENCE:**

http://www.electronics-tutorials.ws/logic/logic\_2.html

**5.1.3 : OR GATE USING NAND GATE:**

**CIRCUIT DIAGRAM:**

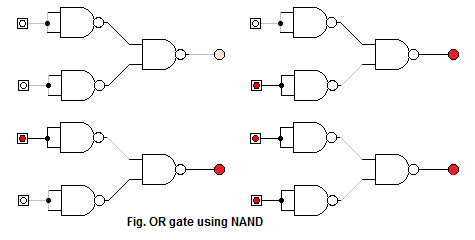
****

****

**TRUTH TABLE:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **(A+B)** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**OBSERVATIONS:**



**OBSERVATION TABLE:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **(A+B)** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**CONCLUSION:**

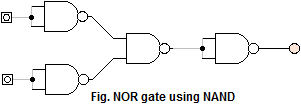
Hence, NAND gate was used as OR gate.

**REFERENCE:**

http://www.electronics-tutorials.ws/logic/logic\_2.html

**5.1.4 : NOR GATE USING NAND GATE:**

**CIRCUIT DIAGRAM:**

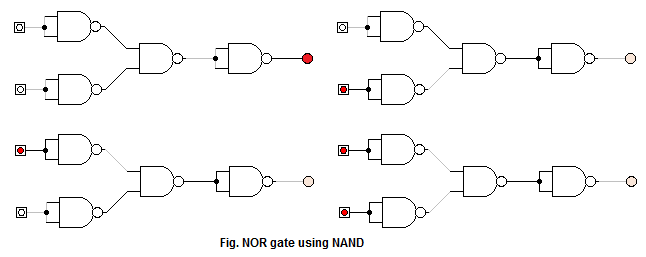
****

****

**TRUTH TABLE:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **(A+B)’** |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

**OBSERVATIONS:**



**OBSERVATION TABLE:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **(A+B)’** |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

**CONCLUSION:**

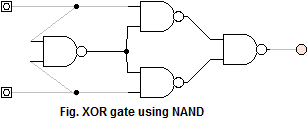
Hence, NAND gate was used as NOR gate.

**REFERENCE:**

http://www.electronics-tutorials.ws/logic/logic\_2.html

**5.1.5 : X-OR GATE USING NAND GATE:**

**CIRCUIT DIAGRAM:**

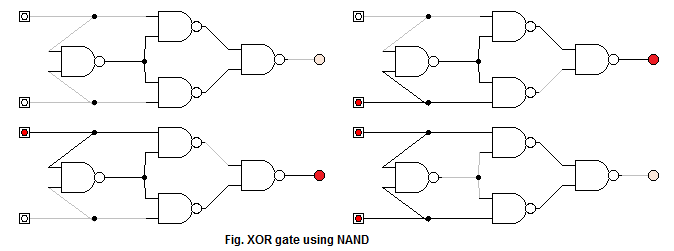
****

****

**TRUTH TABLE:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **(A**⊕**B)=AB’+A’B** |
| **0** | **0** | **0** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **0** |

**OBSERVATIONS:**



**OBSERVATION TABLE:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **(A**⊕**B)=AB’+A’B** |
| **0** | **0** | **0** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **0** |

**CONCLUSION:**

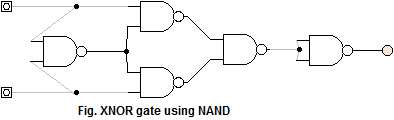
Hence, NAND gate was used as X-OR gate.

**REFERENCE:**

http://www.electronics-tutorials.ws/logic/logic\_2.html

**5.1.6 : X-NOR GATE USING NAND GATE:**

**CIRCUIT DIAGRAM:**

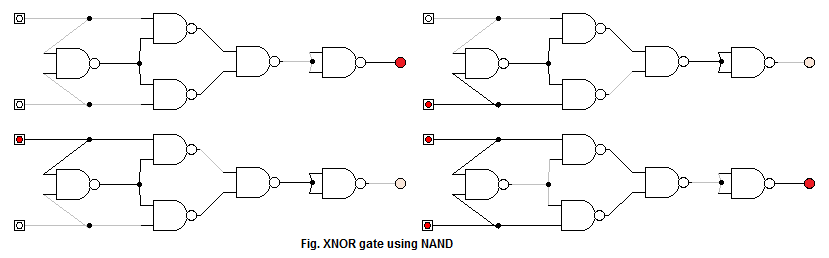
****

****

**TRUTH TABLE:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **AB+A’B’** |
| **0** | **0** | **1** |
| **0** | **1** | **0** |
| **1** | **0** | **0** |
| **1** | **1** | **1** |

**OBSERVATIONS:**

****

**OBSERVATION TABLE:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **AB+A’B’** |
| **0** | **0** | **1** |
| **0** | **1** | **0** |
| **1** | **0** | **0** |
| **1** | **1** | **1** |

**CONCLUSION:**

Hence, NAND gate was used as X-NOR gate.

Hence, the universality of NAND gate was verified for all possible gates.

**REFERENCE:**

http://www.electronics-tutorials.ws/logic/logic\_2.html

**OBJECTIVE 5.2:**

**TO VERIFY THE UNIVERSALITY OF NOR GATE.**

**THEORY:** NOR gate can be defined as the combination of OR gate and NOT gate. In other words output of OR gate is connected to the input of NOR gate. Note that output of OR gate is inverted to form NOR gate.

**Boolean expression’s= (A+B)**

**5.2.1 : NOT GATE USING NOR GATE:**

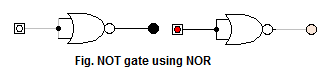
**CIRCUIT DIAGRAM:**

****

**TRUTH TABLE:**

|  |  |
| --- | --- |
| **INPUT** | **OUTPUT** |
| **A** | **A’** |
| 0 | 1 |
| 1 | 0 |

**OBSERVATIONS:**



**OBSERVATION TABLE:**

|  |  |
| --- | --- |
| **INPUT** | **OUTPUT** |
| **A** | **A’** |
| 0 | 1 |
| 1 | 0 |

**CONCLUSION:**

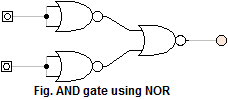
Hence, NOR gate was used as NOT gate.

**REFERENCE:**

<http://www.electronics-tutorials.ws/logic/logic_2.html>

**5.2.2 : AND GATE USING NOR GATE:**

**CIRCUIT DIAGRAM:**

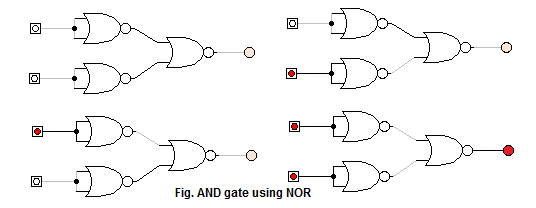
****



**TRUTH TABLE:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **(A.B)** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**OBSERVATIONS:**



**OBSERVATION TABLE:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **(A.B)** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**CONCLUSION:**

Hence, NOR gate was used as AND gate.

**REFERENCE:**

http://www.electronics-tutorials.ws/logic/logic\_2.html

**5.2.3 : OR GATE USING NOR GATE:**

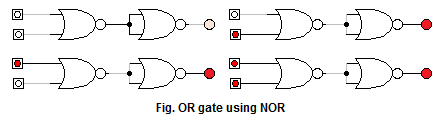
**CIRCUIT DIAGRAM:**

****

**TRUTH TABLE:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **(A+B)** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**OBSERVATIONS:**



**OBSERVATION TABLE:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **(A+B)** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**CONCLUSION:**

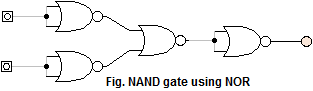
Hence, NOR gate was used as OR gate.

**REFERENCE:**

http://www.electronics-tutorials.ws/logic/logic\_2.html

**5.2.4 : NAND GATE USING NOR GATE:**

**CIRCUIT DIAGRAM:**

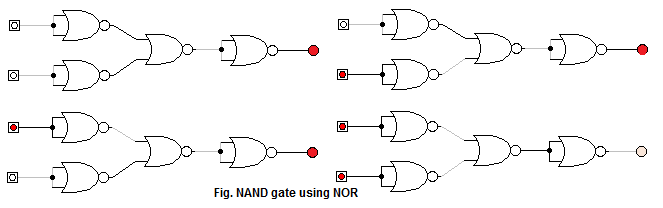
****

****

**TRUTH TABLE:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| A | B | (A.B)’ |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**OBSERVATIONS:**



**OBSERVATION TABLE:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| A | B | (A.B)’ |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**CONCLUSION:**

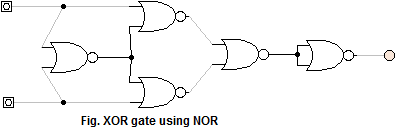
Hence, NOR gate was used as NAND gate.

**REFERENCE:**

http://www.electronics-tutorials.ws/logic/logic\_2.html

**5.2.5 : X-OR GATE USING NOR GATE:**

**CIRCUIT DIAGRAM:**

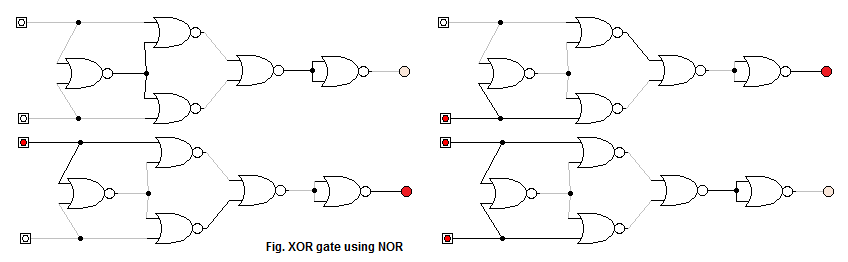
****

****

**TRUTH TABLE:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **(A**⊕**B)=AB’+A’B** |
| **0** | **0** | **0** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **0** |

**OBSERVATIONS:**



**OBSERVATION TABLE:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **(A**⊕**B)=AB’+A’B** |
| **0** | **0** | **0** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **0** |

**CONCLUSION:**

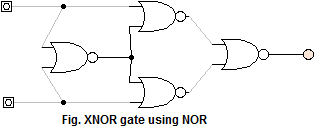
Hence, NOR gate was used as X-OR gate.

**REFERENCE:**

http://www.electronics-tutorials.ws/logic/logic\_2.html

**5.2.5 : X-NOR GATE USING NOR GATE:**

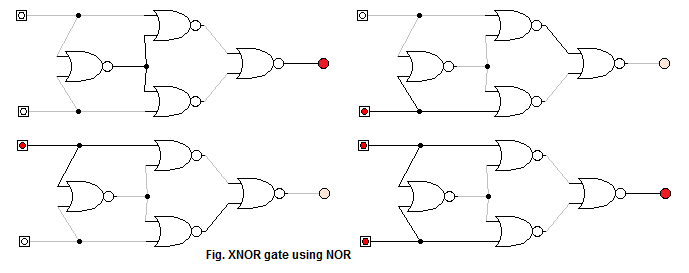
**CIRCUIT DIAGRAM:**



****

**TRUTH TABLE:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **AB+A’B’** |
| **0** | **0** | **1** |
| **0** | **1** | **0** |
| **1** | **0** | **0** |
| **1** | **1** | **1** |

**OBSERVATIONS:**

**OBSERVATION TABLE:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| **A** | **B** | **AB+A’B’** |
| **0** | **0** | **1** |
| **0** | **1** | **0** |
| **1** | **0** | **0** |
| **1** | **1** | **1** |

**CONCLUSION:**

Hence, NOR gate was used as X-NOR gate.

Hence, the universality of NOR gate was verified for all possible gates.

**REFERENCE:**

http://www.electronics-tutorials.ws/logic/logic\_2.html

**OBJECTIVE 6.1:**

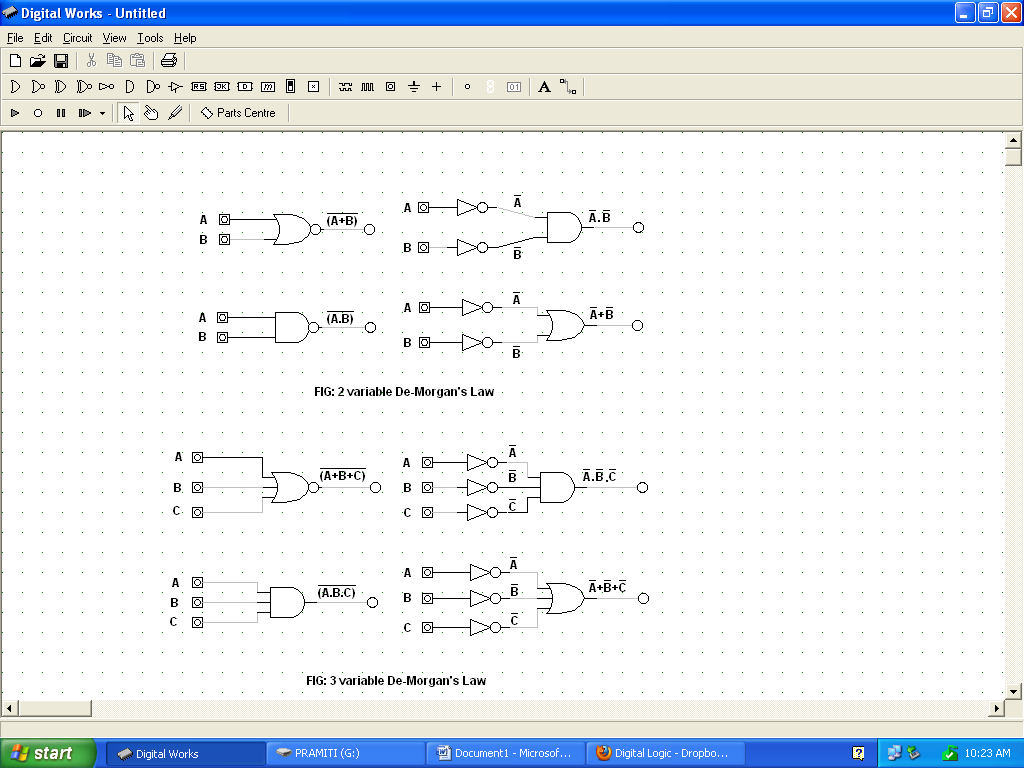
**TO VERIFY 2 VARIABLE DE-MORGAN’S LAW**

**THEORY:**

De- Morgan’s law states that the denial of the conjunction of a class of propositions is equivalent to the disjunction of the denials of a proposition, and the denial of the disjunction of a class of propositions is equivalent to the conjunction of the denials of the propositions.

**Boolean Expressions:**

1. (A+B)’=A’.B’
2. (A.B)’=A’+B’

**CIRCUIT DIAGRAM:**

**TRUTH TABLE:**

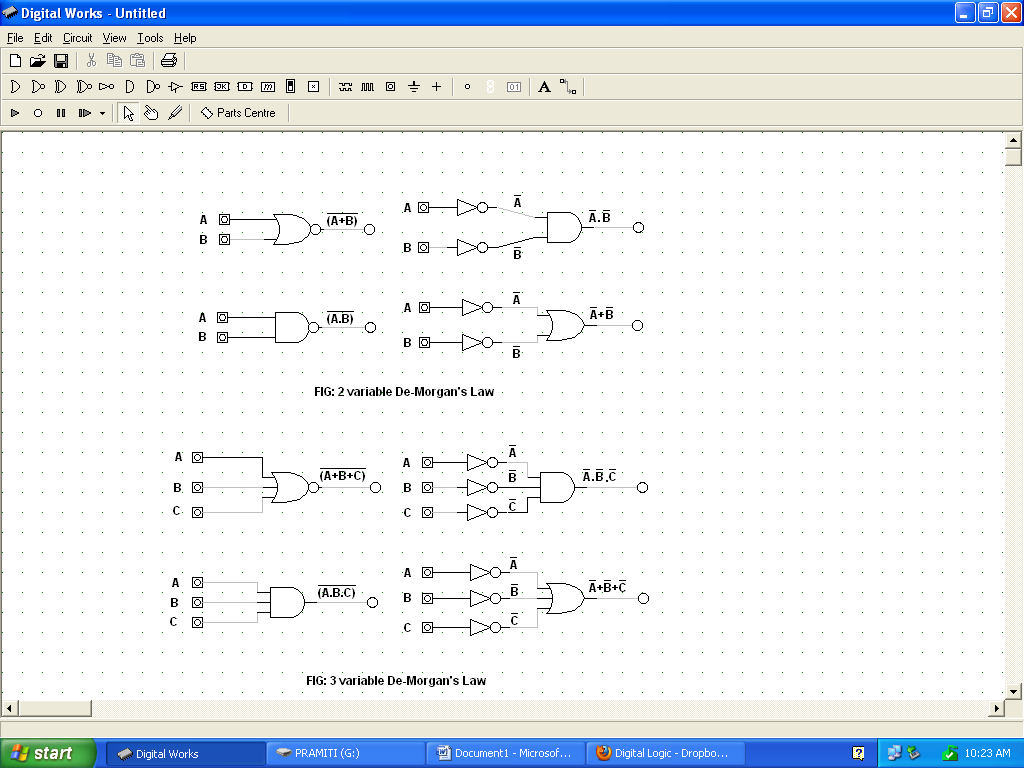
1. (A+B)’=A’.B’

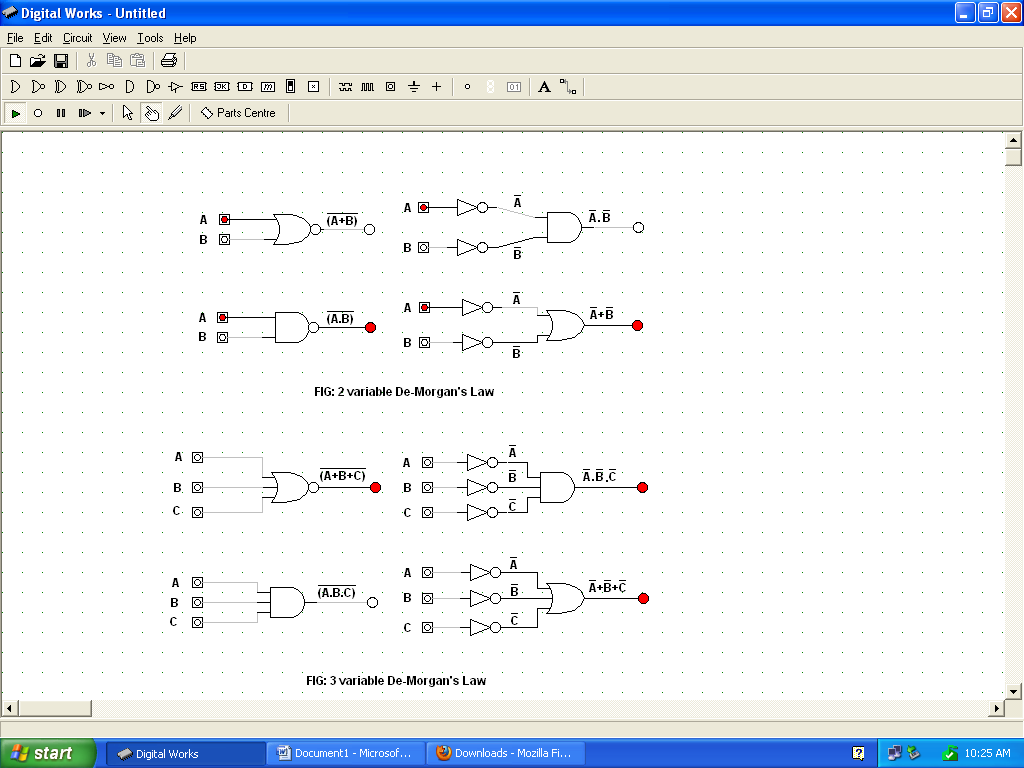
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Input** | | **Output** | | | | |
| **A** | **B** | **A’** | **B’** | **(A+B)** | **(A+B)’** | **A’.B’** |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |

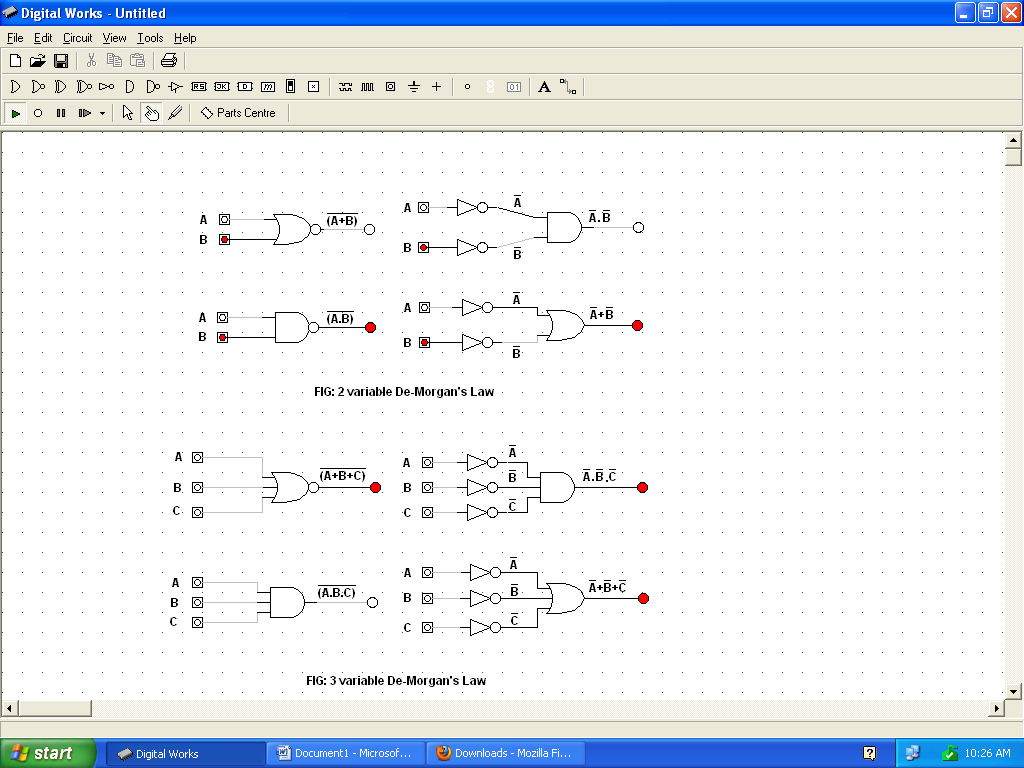
1. (A.B)’=A’+B’

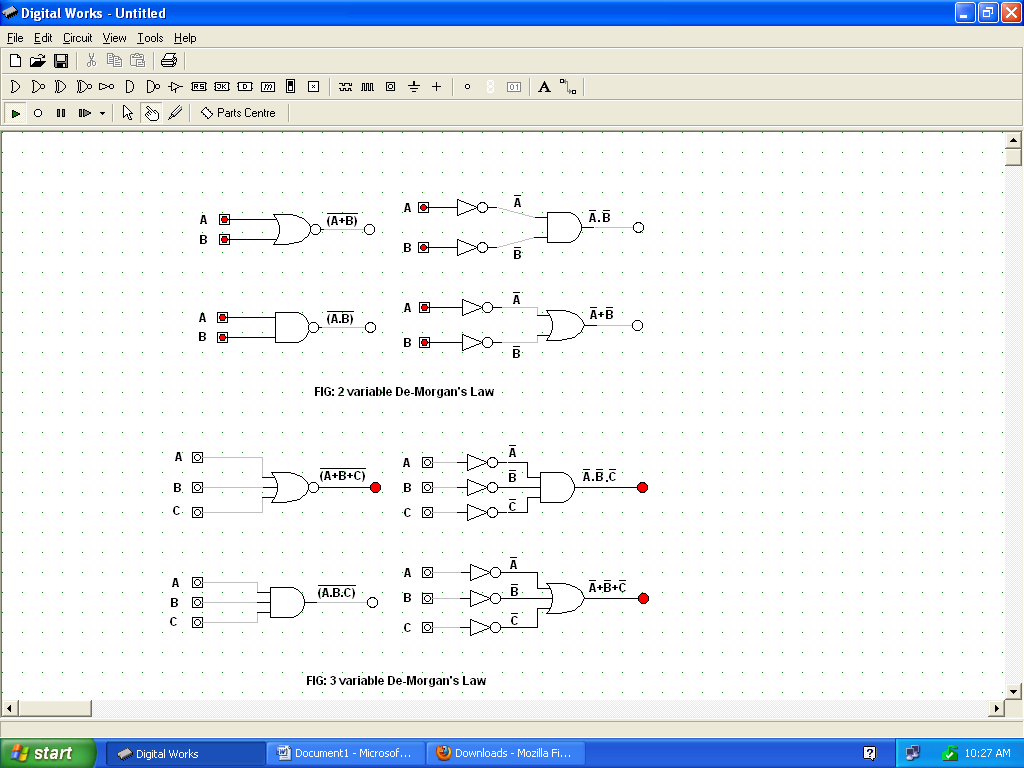
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Input** | | **Output** | | | | |
| **A** | **B** | **A’** | **B’** | **(A.B)** | **(A.B)’** | **A’+B’** |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |

**OBSERVATIONS:**









**OBSERVATION TABLE:**

1. (A+B)’=A’.B’

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Input** | | **Output** | | | | |
| **A** | **B** | **A’** | **B’** | **(A+B)** | **(A+B)’** | **A’.B’** |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |

1. (A.B)’=A’+B’

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Input** | | **Output** | | | | |
| **A** | **B** | **A’** | **B’** | **(A.B)** | **(A.B)’** | **A’+B’** |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |

**CONCLUSION:**

Hence, the 2 variable De-Morgan’s Law is verified.

**REFERENCE:**

<http://dictionary.reference.com/browse/de+morgan%27s+laws>

**OBJECTIVE 6.2:**

**TO VERIFY 3 VARIABLE DE-MORGAN’S LAW**

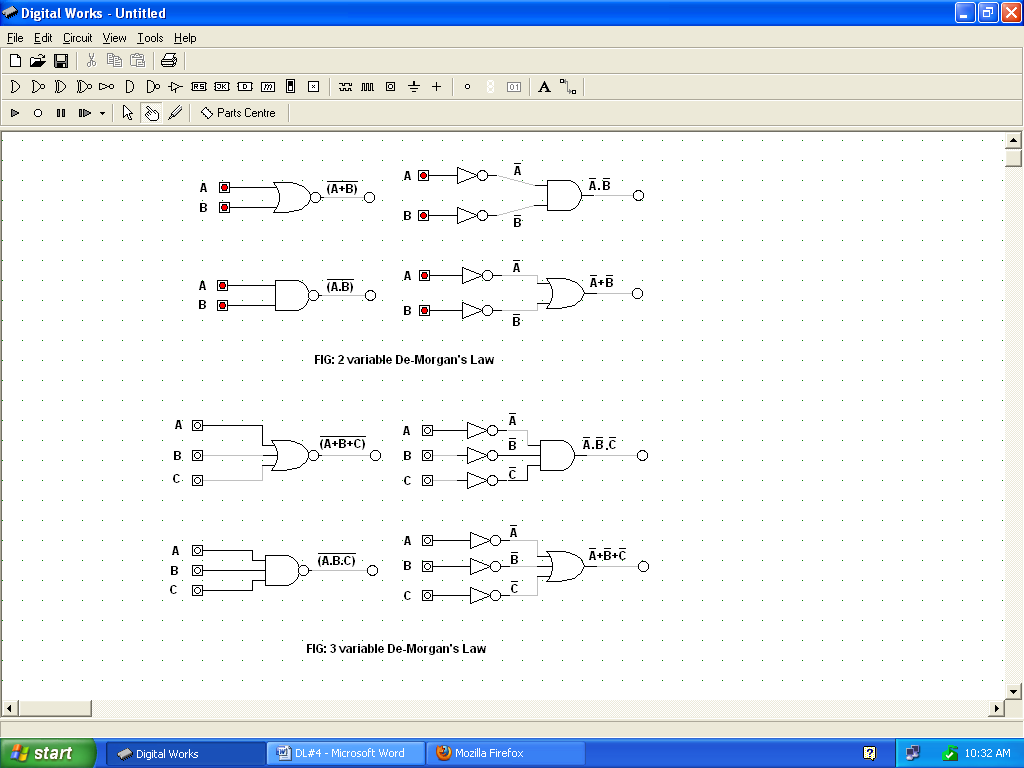
**THEORY:**

De- Morgan’s law states that the denial of the conjunction of a class of propositions is equivalent to the disjunction of the denials of a proposition, and the denial of the disjunction of a class of propositions is equivalent to the conjunction of the denials of the propositions.

**Boolean Expressions:**

1. (A+B+C)’=A’.B’.C’
2. (A.B.C)’=A’+B’+C’

**CIRCUIT DIAGRAM:**



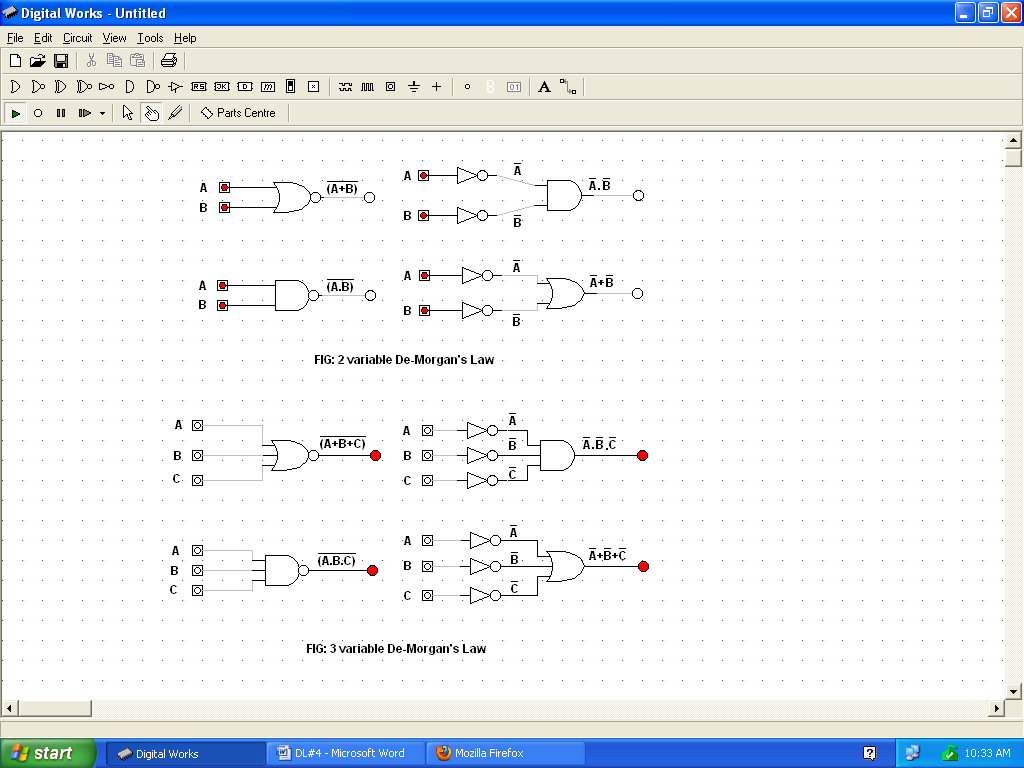
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | | | **Output** | | | | | |
| **A** | **B** | **C** | **A’** | **B’** | **C’** | **(A+B+C)** | **(A+B+C)’** | **A’.B’.C’** |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

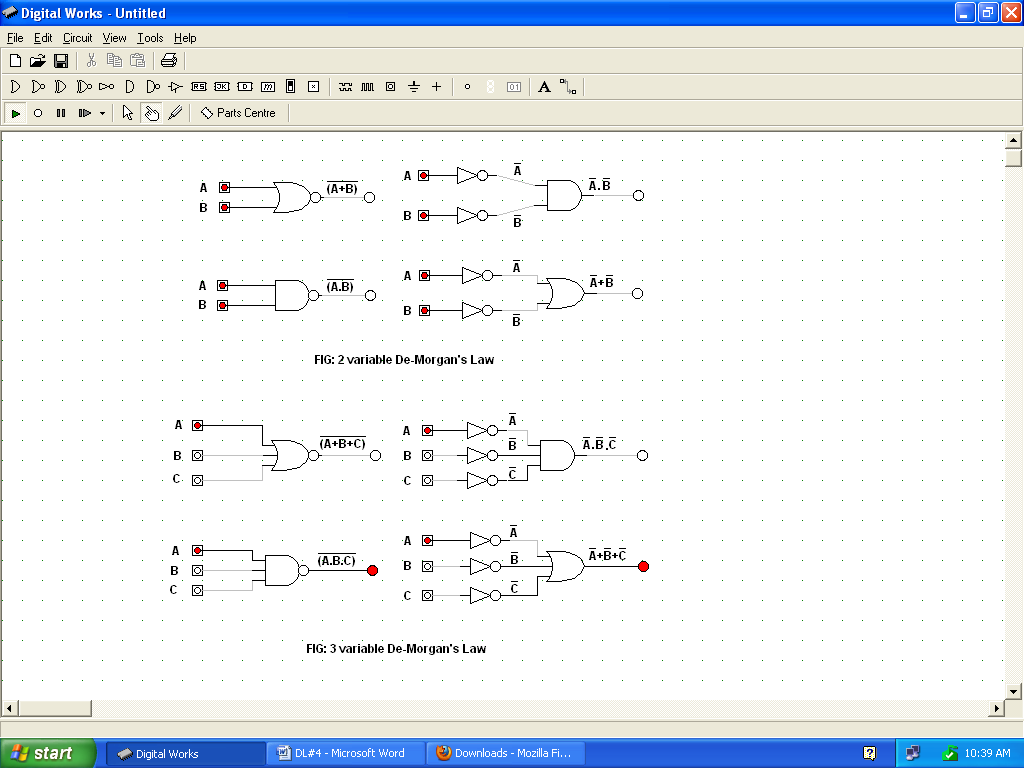
**TRUTH TABLE:**

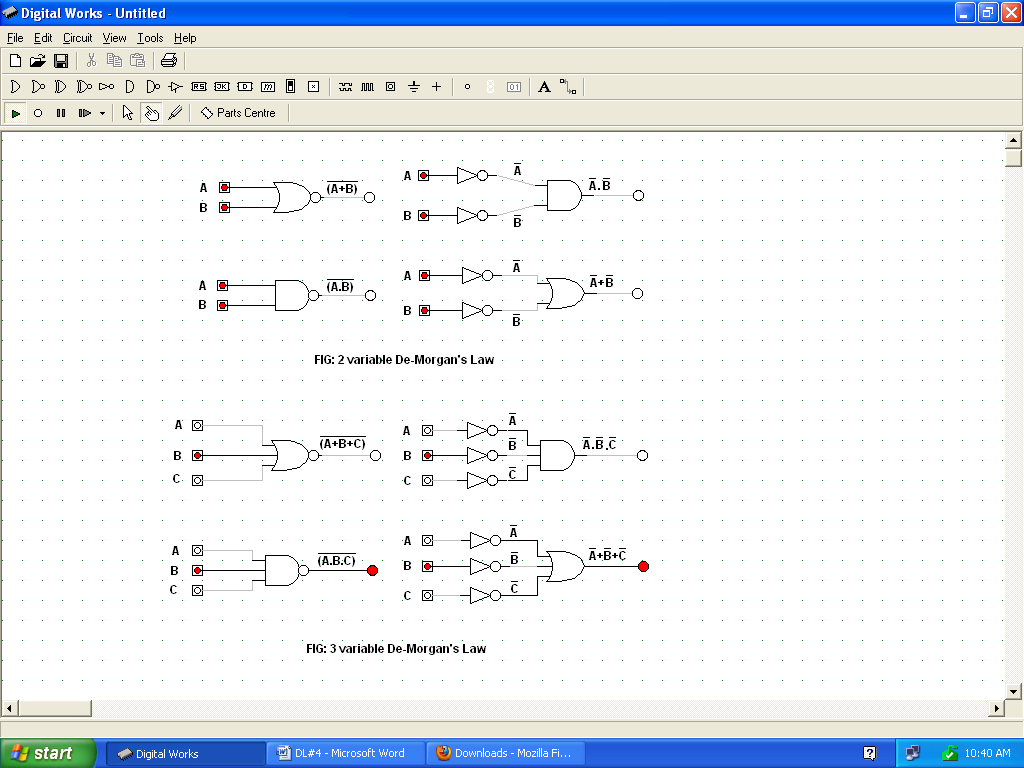
1. (A+B+C)’=A’.B’.C’
2. (A.B.C)’=A’+B’+C’

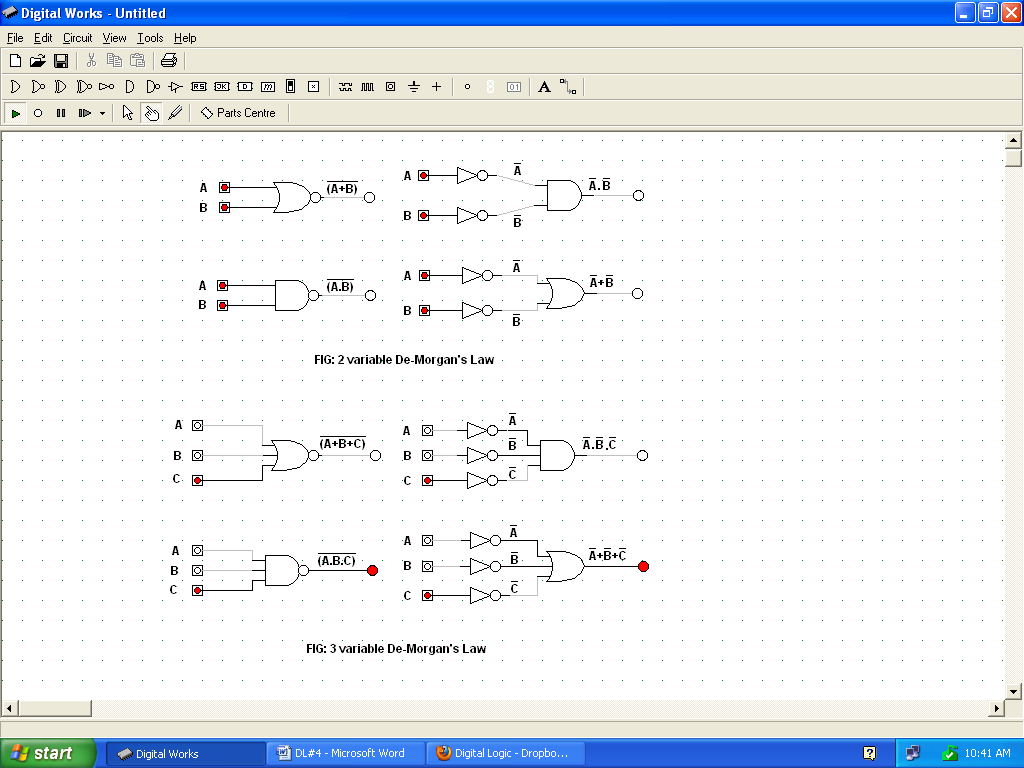
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | | | **Output** | | | | | |
| **A** | **B** | **C** | **A’** | **B’** | **C’** | **(A.B.C)** | **(A.B.C)’** | **A’+B’+C’** |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

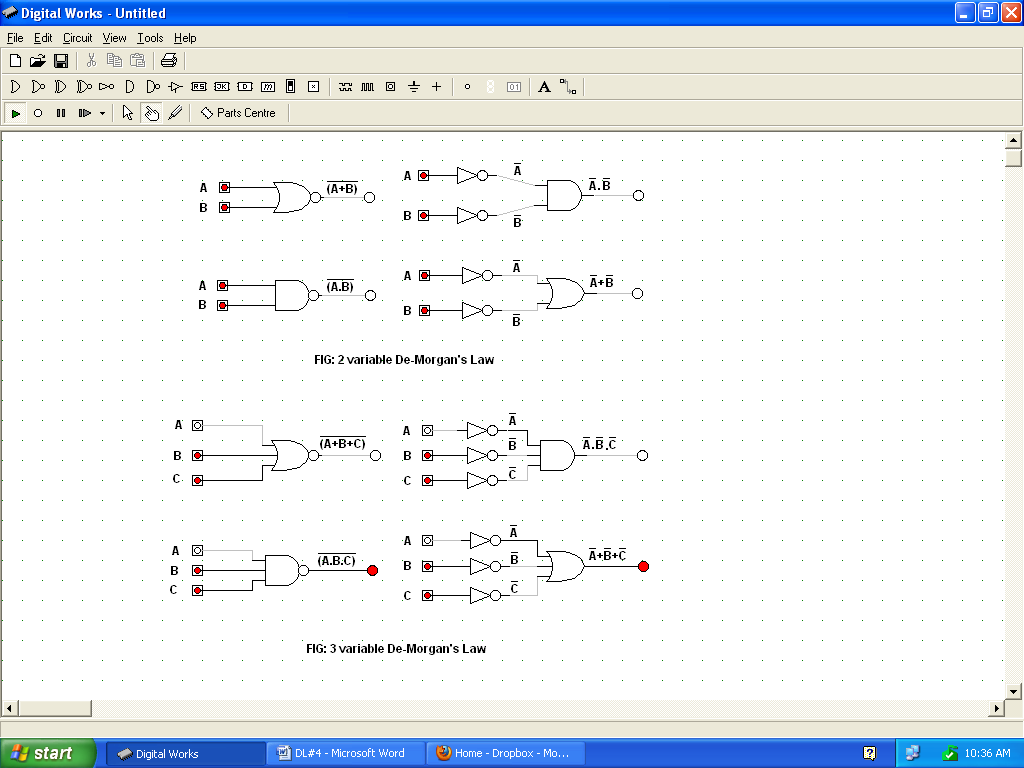
**OBSERVATIONS:**

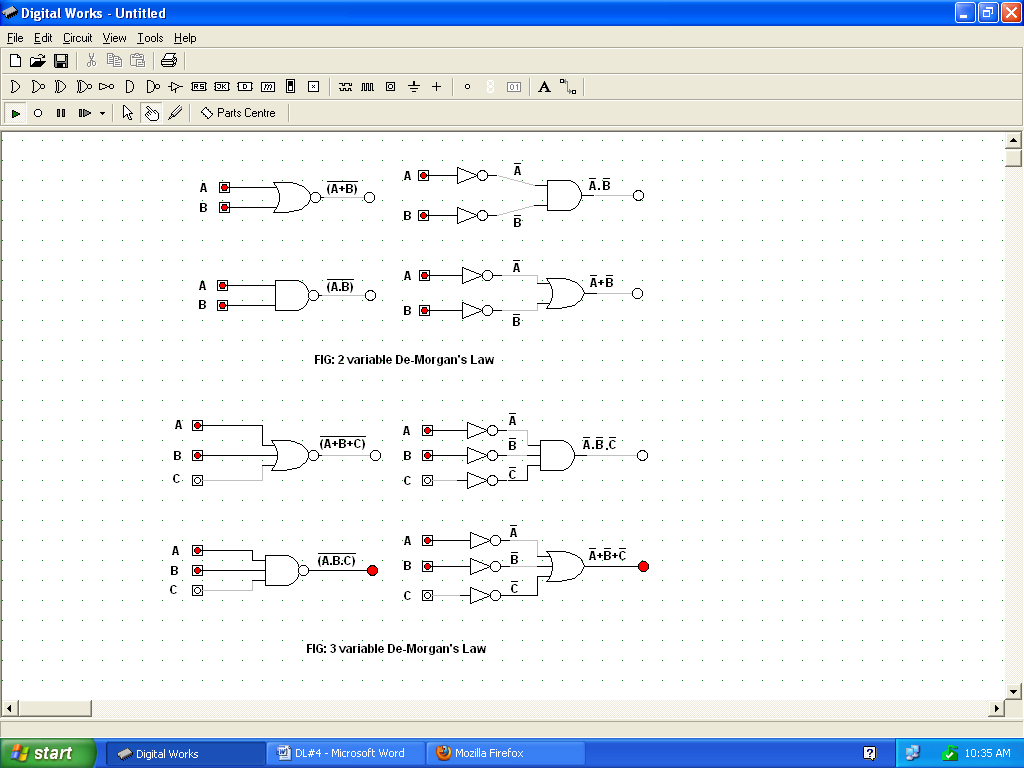


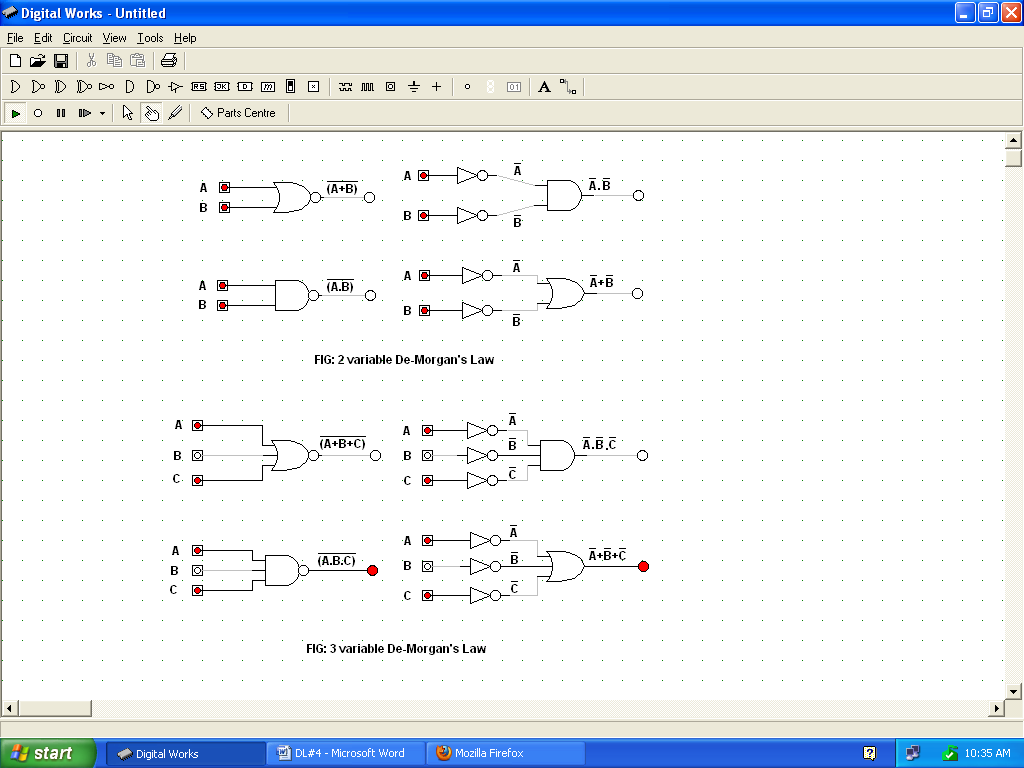


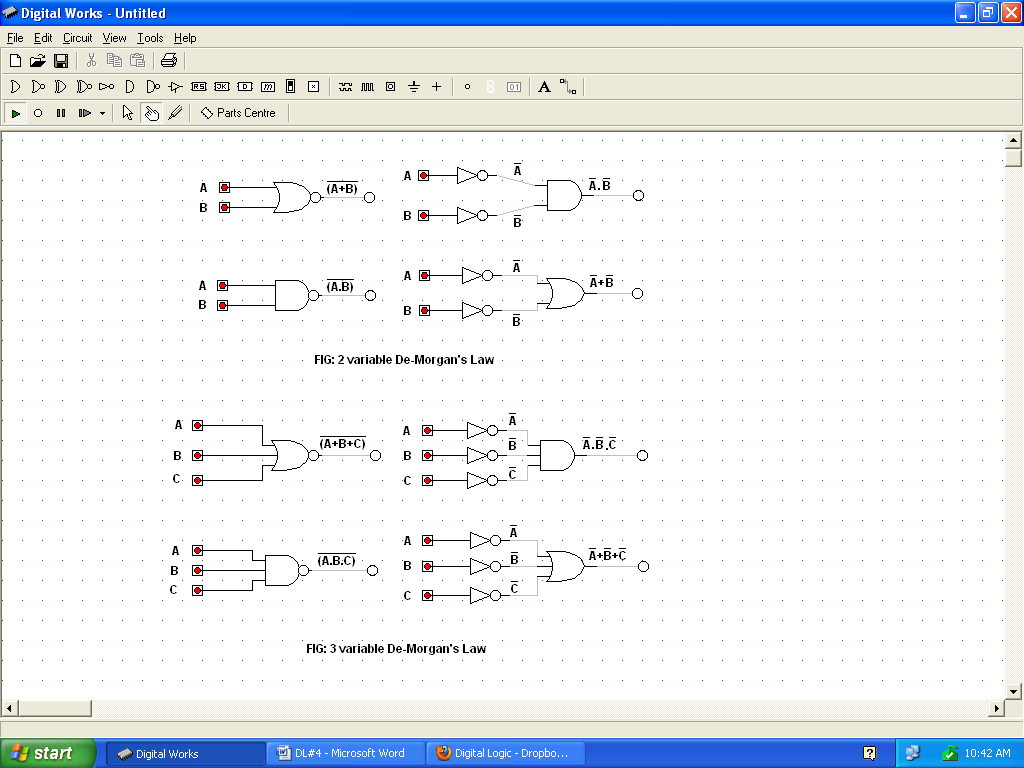












**OBSERVATION TABLE:**

1. (A+B+C)’=A’.B’.C’

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | | | **Output** | | | | | |
| **A** | **B** | **C** | **A’** | **B’** | **C’** | **(A+B+C)** | **(A+B+C)’** | **A’.B’.C’** |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

1. (A.B.C)’=A’+B’+C’

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | | | **Output** | | | | | |
| **A** | **B** | **C** | **A’** | **B’** | **C’** | **(A.B.C)** | **(A.B.C)’** | **A’+B’+C’** |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

**CONCLUSION:**

Hence, the 3 variable De-Morgan’s Law is verified.

**REFERENCE:**

<http://dictionary.reference.com/browse/de+morgan%27s+laws>