Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

**06**

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| 1 | Complete the table by solving the bitwise instruction of all Logical gates. Add the code and output of the logical gates to show solution of MASK BITS given in the table. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Submitted On:

Date: 12/11/2022

**Task No. 1: Complete the table by solving the bitwise instruction of all Logical gates. Add the code and output of the logical gates to show solution of MASK BITS given in the table.**

**AND Operator**

**Source Code:**

.data

input: .asciiz "Enter integer Number: "

result: .asciiz "The result is: "

.text

.globl main

main:

li $t0,0xffffffff

la $a0,input

li $v0,4

syscall

li $v0,5

syscall

move $t1,$v0

and $t2,$t1,$t0

la $a0,result

li $v0,4

syscall

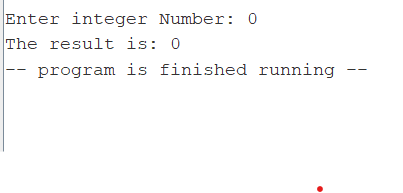
move $a0,$t2

li $v0,1

syscall

li $v0,10

syscall



Graphical user interface, text, application

Description automatically generated with medium confidence

**When we write li $t0,0x00000000 instead of li $t0,0xffffffff then we will get:**

Graphical user interface, text, application

Description automatically generated

**OR Operator**

**Source Code:**

.data

input: .asciiz "Enter integer Number: "

result: .asciiz "The result is: "

.text

.globl main

main:

li $t0,0x00000000

la $a0,input

li $v0,4

syscall

li $v0,5

syscall

move $t1,$v0

or $t2,$t1,$t0

la $a0,result

li $v0,4

syscall

move $a0,$t2

li $v0,1

syscall

li $v0,10

syscall

Text, letter

Description automatically generated

**When we write li $t0,0xffffffff instead of li $t0,0x00000000 then we will get:**

Graphical user interface, text

Description automatically generated

**XOR Operator**

**Source Code:**

.data

input: .asciiz "Enter integer Number: "

result: .asciiz "The result is: "

.text

.globl main

main:

li $t0,0xffffffff

la $a0,input

li $v0,4

syscall

li $v0,5

syscall

move $t1,$v0

xor $t2,$t1,$t0

la $a0,result

li $v0,4

syscall

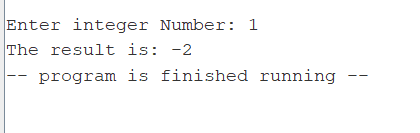
move $a0,$t2

li $v0,1

syscall

li $v0,10

syscall

****

**When we write li $t0,0x00000000 instead of li $t0,0xffffffff then we will get:**

**Text

Description automatically generated**

**NOT Operator**

**Source Code:**

.data

input: .asciiz "Enter Number "

result: .asciiz "The result is "

.text

.globl main

main:

li $v0,4

la $a0,input

syscall

li $v0,5

syscall

move $t0,$v0

not $t1,$t0

li $v0,4

la $a0,result

syscall

move $a0,$t1

li $v0,1

syscall

li $v0,10

syscall

**Graphical user interface, text, application

Description automatically generated**

**XNOR Operator**

**Source Code:**

.data

input: .asciiz "Enter integer Number: "

result: .asciiz "The result is: "

.text

.globl main

main:

li $t0,0xffffffff

la $a0,input

li $v0,4

syscall

li $v0,5

syscall

move $t1,$v0

xor $t2,$t1,$t0

la $a0,result

li $v0,4

syscall

not $t3,$t2

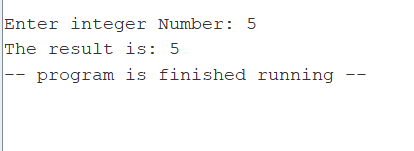
move $a0,$t3

li $v0,1

syscall

li $v0,10

syscall



**When we write li $t0,0x00000000 instead of li $t0,0xffffffff then we will get:**

**Graphical user interface, text, application

Description automatically generatedNOR Operator:**

**Source Code:**

.data

input: .asciiz "Enter Number "

result: .asciiz "The result is "

.text

.globl main

main:

li $t0,0xffffffff

li $v0,4

la $a0,input

syscall

li $v0,5

syscall

move $t1,$v0

nor $t2,$t0,$t1

li $v0,4

la $a0,result

syscall

move $a0,$t2

li $v0,1

syscall

li $v0,10

syscall

Graphical user interface, text, application

Description automatically generated with medium confidence

**When we write li $t0,0x00000000 instead of li $t0,0xffffffff then we will get:**

**Graphical user interface, application

Description automatically generatedNAND Operator:**

**Source Code:**

.data

input: .asciiz "Enter Number "

result: .asciiz "The result is "

.text

.globl main

main:

li $t0,0xffffffff

la $a0,input

li $v0,4

syscall

li $v0,5

syscall

move $t1,$v0

and $t2,$t1,$t0

la $a0,result

li $v0,4

syscall

not $t3,$t2

move $a0,$t3

li $v0,1

syscall

li $v0,10

syscall

Graphical user interface, text, application

Description automatically generated

**When we write li $t0,0x00000000 instead of li $t0,0xffffffff then we will get:**

**Graphical user interface, text, application

Description automatically generated with medium confidence**

**Output:**

|  |  |  |
| --- | --- | --- |
| **Logic** | **Mask Bits** | |
|  | **0** | **1** |
| **AND** | 0 | 9 |
| **OR** | 1 | -1 |
| **NOT** | -6 | -6 |
| **XOR** | 1 | -2 |
| **XNOR** | -6 | 5 |
| **NOR** | 0 | -10 |
| **NAND** | -1 | -6 |