Task No 01:

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.

A table with black text

Description automatically generated

Solution:

.data

mask0:.word 0x00000000

mask1:.word 0xffffffff

spaceseq: .asciiz "\n"

.text

li $v0,5

syscall

move $t0,$v0

lw $t1, mask0

lw $t2, mask1

and $t5,$t0,$t2

and $t6,$t0,$t1

move $a0,$t5

li $v0,1

syscall

la $a0,spaceseq

li $v0,4

syscall

move $a0,$t6

li $v0,1

syscall

li $v0,10

syscall

|  |  |
| --- | --- |
| **Logic** | **Mask Bits** |
| AND | 0 |
| OR | 1 |
| NOT | 1 |
| XOR | 1 |
| XNOR | 0 |
| NOR | 1 |
| NAND | 1 |

Output:

Task No 02:

Write an MIPS assembly program that clears the 5th bit of a binary value stored in a register using the bitwise 'and' operation.

Solution:

.data

mask: .word 0xC

.text

li $t0, 0x1C

lw $t1, mask

and $t2, $t0,$t1

li $v0, 1

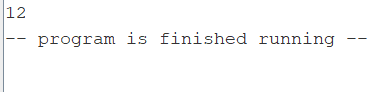
move $a0, $t2

syscall

li $v0, 10

syscall

Output:



Task No 03:

Create a program that sets the 3rd bit of a binary value in a register using the 'or' operation.

Solution:

.data

mask: .word 0x04

.text

li $t0, 0x11

lw $t1, mask

or $t2, $t0,$t1

li $v0, 1

move $a0, $t2

syscall

li $v0, 10

syscall

Output:

A black and white text

Description automatically generated

Task No 04:

Can you demonstrate how to toggle the 4th bit in a binary value using the 'xor' operation in MIPS?

Solution:

.data

.text

li $t0, 0x02

sll $t1, $t0, 2

move $a0, $t1

li $v0, 1

syscall

li $v0, 10

syscall

Output:

A black and white text

Description automatically generated

Task No 05:

Implement a program that multiplies a binary value in a register by 32 (2^5) using the 'sll' operation and divides a binary value by 16 (2^4) using the 'srl' operation.

Solution:

.data

.text

li $t0, 0x10

srl $t1,$t0,4

li $v0, 1

move $a0, $t1

syscall

li $v0, 10

syscall

Output:

A black and white text

Description automatically generated