Bahria University,

Karachi Campus

LAB EXPERIMENT NO.

06

LIST OF TASKS

TASK NO	OBJECTIVE
1.	Write a provide MIPS assembly program that performs bitwise operations between two 32-bit hexadecimal values. Complete the following table by calculating the results of AND, OR, XOR, XNOR, NOR, and NAND operations for each pair of given values.
2.	Write an MIPS assembly program that clears the 1st , 3 rd and 8th bit of a binary value (00000010 00100000 00000001 11110011) stored in a register using the bitwise 'AND' operation.
3.	Create a program that sets the 3rd , 5 th and 11 th bit of a binary value (00000001 00001000 10110000 1010 0000) in a register using the 'OR' operation.
4	Can you demonstrate how to toggle the last 8 bits in a binary value (00000001 00000001 00000100 10000101) using the 'XOR' operation in MIPS?
5	Implement a MIPS program that multiplies a binary value (any) in a register by 28 using the 'SLL' operation and divides a binary value by 23 using the 'SRL' operation.

Submitted On:

Date: 17/10/2025

Task No. 01:

"Write a provide MIPS assembly program that performs bitwise operations between two 32-bit hexadecimal values. Complete the following table by calculating the results of AND, OR, XOR, XNOR, NOR, and NAND operations for each pair of given values."

Values		Operational values					
Value(A)	Value(B)	AND	OR	XOR	NOR	XNOR	NAND
0x00000000	0x00000000						
0x00000000	0x00000001						
0x00000001	0x00000000						
0x00000001	0x00000001						

Solution:

```
.data
And: .asciiz " And "
Or : .asciiz " Or "
Xor : .asciiz " Xor "
Nor : .asciiz " Nor "
Xnor: .asciiz " Xnor(not OR) "
Nand : .asciiz " Nand (not AND) "
newLine: .asciiz "\n"
valueA: .word 0x00000000
valueB: .word 0x00000001

.text
.globl main
main:
 lw $t0,valueA    #valueA=0
 lw $t1,valueB    #valueB=0

 li $v0,4
 la $a0,And
 syscall

 and $t2,$t0,$t0
 and $t3,$t0,$t1
 and $t4,$t1,$t0
 and $t5,$t1,$t1

 li $v0,1
 move $a0,$t2
```

```
syscall
li $v0,1
move $a0,$t3
syscall
li $v0,1
move $a0,$t4
syscall
li $v0,1
move $a0,$t5
syscall

li $v0,4
la,$a0,newLine
syscall

li $v0,4
la $a0,Nand
syscall

not $t2,$t2
not $t3,$t3
not $t4,$t4
not $t5,$t5

li $v0,1
move $a0,$t2
syscall
li $v0,1
move $a0,$t3
syscall
li $v0,1
move $a0,$t4
syscall
li $v0,1
move $a0,$t5
syscall

li $v0,4
la $a0,newLine
syscall

li $v0,4
la,$a0,Or
syscall

or $t2,$t0,$t0
or $t3,$t0,$t1
or $t4,$t1,$t0
or $t5,$t1,$t1
```

```
li $v0,1
move $a0,$t2
syscall
li $v0,1
move $a0,$t3
syscall
li $v0,1
move $a0,$t4
syscall
li $v0,1
move $a0,$t5
syscall

li $v0,4
la $a0,newLine
syscall

li $v0,4
la $a0,Xor
syscall

xor $t2,$t0,$t0
xor $t3,$t0,$t1
xor $t4,$t1,$t0
xor $t5,$t1,$t1

li $v0,1
move $a0,$t2
syscall
li $v0,1
move $a0,$t3
syscall
li $v0,1
move $a0,$t4
syscall
li $v0,1
move $a0,$t5
syscall

li $v0,4
la $a0,newLine
syscall

li $v0,4
la $a0,Xnor
syscall

xor $t6,$t0,$t0
```

```
xor $t7,$t0,$t1
xor $t8,$t1,$t0
xor $t9,$t1,$t1

li $v0,1
move $a0,$t6
syscall
li $v0,1
move $a0,$t7
syscall
li $v0,1
move $a0,$t8
syscall
li $v0,1
move $a0,$t9
syscall

li $v0,4
la $a0,newLine
syscall

li $v0,4
la $a0,Nor
syscall

nor $t2,$t0,$t0
nor $t3,$t0,$t1
nor $t4,$t1,$t0
nor $t5,$t1,$t1

li $v0,1
move $a0,$t2
syscall
li $v0,1
move $a0,$t3
syscall
li $v0,1
move $a0,$t4
syscall
li $v0,1
move $a0,$t5
syscall

li $v0,4
la $a0,newLine
syscall
```

Output:

```
And 0001
Nand (not AND) -1-1-1-2
Or 0111
Xor 0110
Xnor(not OR) 0110
Nor -1-2-2-2

-- program is finished running (dropped off bottom) --
```

Task No. 02:

"Write an MIPS assembly program that clears the 1st , 3 rd and 8th bit of a binary value (00000010 00100000 00000001 11110011) stored in a register using the bitwise 'AND' operation."

Solution:

```
.data
actualBits: .word 0x022001F3
maskingBits: .word 0xFFFFFF7A
.text
.globl main
main:
 lw $t0,actualBits
 lw $t1,maskingBits

 and $t3,$t0,$t1

 li $v0,1
 move $a0,$t3
 syscall

 li $v0,10
 syscall
```

Output:

```
35651954
-- program is finished running --
```

Task No. 03:

"Create a program that sets the 3rd , 5 th and 11 th bit of a binary value (00000001 00001000 10110000 1010 0000) in a register using the 'OR' operation."

Solution:

```
.data
actualBits: .word 0x0108B0A0
maskingBits:.word 0x00000414
.text
.globl main
main:
 lw $t0,actualBits
 lw $t1,maskingBits

 or $t3,$t0,$t1

 li $v0,1
 move $a0,$t3
 syscall

 li $v0,10
 syscall
```

Output:

```
17347764
-- program is finished running --
```

Task No. 04:

"Can you demonstrate how to toggle the last 8 bits in a binary value (00000001 00000001 00000100 10000101) using the 'XOR' operation in MIPS?"

Solution:

```
.data
actualBits: .word 0x01010485
maskingBits:.word 0xFF000000.text
.globl main
main:
 lw $t0,actualBits
 lw $t1,maskingBits

 xor $t3,$t0,$t1

 li $v0,1
 move $a0,$t3
 syscall

 li $v0,10
 syscall
```

Output:

```
-33487739
-- program is finished running --
```

Task No. 05:

"Implement a MIPS program that multiplies a binary value (any) in a register by 2^8 using the 'SLL' operation and divides a binary value by 2^3 using the 'SRL' operation."

Solution:

```
.data
msg1: .asciiz "Original value: "
msg2: .asciiz "After multiplying: "
msg3: .asciiz "After dividing: "
newline: .asciiz "\n"
val: .word 0x00000015
.text
.globl main
main:

   lw $t0, val
   li $v0, 4
   la $a0, msg1
   syscall

   li $v0, 1
   move $a0, $t0
   syscall

   li $v0, 4
   la $a0, newline
   syscall

   sll $t1, $t0, 8

   li $v0, 4
   la $a0, msg2
   syscall

   li $v0, 1
   move $a0, $t1
   syscall

   li $v0, 4
   la $a0, newline
   syscall

   srl $t2, $t0, 3

   li $v0, 4
```

```
la $a0, msg3
syscall

li $v0, 1
move $a0, $t2
syscall

li $v0, 4
la $a0, newline
syscall

li $v0, 10
syscall
```

Output:

```
Original value: 21
After multiplying: 5376
After dividing: 2

-- program is finished running --
```