**CS 224**

**COMPUTER ORGANIZATION**

**PRELIMINARY DESIGN REPORT**

**LAB 02**

**BERK YILDIZ**

**21502040**

**SECTION 4**

* 1. **Converting Hex to Decimal**

.text

convertHexToDecimal:

move $t1, $a0 # moves string to $t1

# Initialize of the variables

lbu $t7, 0($t1) # get the first character from the

#string

addi $t2, $zero, 48 # 0's ASCII value

addi $t3, $zero, 70 # 15's ASCII value

addi $t4, $zero, 0 # sum of the digits in decimal

addi $t6, $zero, 0 # initilize $t6 which is the index

addi $t5, $zero, 1 # initilize $t5, j, each digit is going to be multiplied by $t5

firstj: # count the first factor to multiply the most sign dig by

beq $t6, 1, lastj

mul $t5, $t5, 16

addi $t6, $t6, -1 #decrements the index

j firstj

counter: # length of string

beq $t7, $zero, firstj

addi $t6, $t6, 1 # increments the index

addi $t1, $t1, 1

lbu $t7, 0($t1)

j counter

lastj:

lbu $t7, 0($t1) # get the first character

loop: # calculate decimal number

beq $t7, $zero, endLoop

slt $t9, $t7, $t2

beq $t9, 1, err

slt $t9, $t3, $t7

beq $t9, 1, err

#convert character to integer

addi $t8, $t7, -48

mul $t8, $t8, $t5

add $t4, $t4, $t8

addi $t1, $t1, 1

lbu $t7, 0($t1)

mul $t5, $t5, 16

j loop

endLoop:

move $v0, $t4

jr $ra

err: #gives an error if there is not a hexademical number

addi $t0, $zero, -1

.data

hexNo: .asciiz "1A"

* 1. **User Interaction**

.data

prompt: .asciiz "Please enter the number in hexadecimal form: "

errPrompt:.asciiz "Number is not in hexadecimal form!\n"

outputPrompt: .asciiz "The decimal value of the given number is "

hexadecimal: .space 40

.text

# Prompt

li $v0, 4

la $a0, prompt

syscall

# Get the input in string form

li $v0, 8

la $a0, hexadecimal

move $t1, $a0 # move the input in to $t1

jal convertHexToDecimal

move $t0, $v0

# Print output

li $v0, 4

la $a0, outputPrompt

syscall

li $v0, 1

la $a0, ($s0)

syscall

li $v0, 10

syscall

convertHexToDecimal:

move $t1, $a0 # moves string to $t1

# Initialize of the variables

lbu $t7, 0($t1) # get the first character from the string

addi $t2, $zero, 48 # 0's ASCII value

addi $t3, $zero, 70 # 15's ASCII value

addi $t4, $zero, 0 # sum of the digits in decimal

addi $t6, $zero, 0 # initilize $t6 which is the index

addi $t5, $zero, 1 # initilize $t5, j, each digit is going to be multiplied by $t5

firstj: # count the first factor to multiply the most sign dig by

beq $t6, 1, lastj

mul $t5, $t5, 16

addi $t6, $t6, -1 #decrements the index

j firstj

counter: # length of string

beq $t7, $zero, firstj

addi $t6, $t6, 1 # increments the index

addi $t1, $t1, 1

lbu $t7, 0($t1)

j counter

lastj:

lbu $t7, 0($t1) # get the first character

loop: # calculate decimal number

beq $t7, $zero, endLoop

slt $t9, $t7, $t2

beq $t9, 1, err

slt $t9, $t3, $t7

beq $t9, 1, err

#convert character to integer

addi $t8, $t7, -48

mul $t8, $t8, $t5

add $t4, $t4, $t8

addi $t1, $t1, 1

lbu $t7, 0($t1)

mul $t5, $t5, 16

j loop

endLoop:

move $v0, $t4

jr $ra

err: #gives an error if there is not a hexademical number

addi $t0, $zero, -1

**2) Generating Object Code**

Jump instruction is a J-Type

**opcode** 6 bits – **address** 26 bits

beq and bne instructions are I-type

**opcode** 6 bits – **rs** 5 bits – **rt** 5 bits – **imm** 16 bits

10 01 00 30 again: add ...

10 01 00 34 add ...

10 01 00 38 add ...

10 01 00 3C beq $t0, $t1, next

10 01 00 40 bne $t0, $t1, again

10 01 00 44 add ...

10 01 00 48 add ...

10 01 00 4C next: j again

beq $t0, $t1, next

000100 01001 01000 0000000000010000 = 0x11280010

bne $t0, $t1, again

000101 01001 01000 1111111111111110 = 0x1528FFFE

j again

000010 00000000000100000000001100 = 0x0800400C