Name: Nguyễn Thanh Long

ID: ITITRG18012

**Lab 03**

**3.1.1)**

.data

.text

.globl main

main:

# Load half-word DEAD into $t1

lui $t1, 0xDEAD

# t1 = t1 OR 0xBEEF

ori $t1, $t1, 0xBEEF

jr $ra

****

**3.1.2)**

.data # the data segment

.text # the code segment

.globl main

main:

ori $t1, $t1, 0xD

sll $t1, $t1, 4

ori $t1, $t1, 0xE

sll $t1, $t1, 4

ori $t1, $t1, 0xA

sll $t1, $t1, 4

ori $t1, $t1, 0xD

sll $t1, $t1, 4

ori $t1, $t1, 0xB

sll $t1, $t1, 4

ori $t1, $t1, 0xE

sll $t1, $t1, 4

ori $t1, $t1, 0xE

sll $t1, $t1, 4

ori $t1, $t1, 0xF

jr $ra # return to caller (\_\_start)

****

**3.1.3)**

.data # the data segment

.text # the code segment

.globl main

main:

# Load 0xDEADBEEF into $t1

li $t1, 0xDEADBEEF

# t3 = t1 AND 0xF = F

andi $t3, $t1, 0xF

# t2 = t3 OR t2 = F

or $t2, $t3, $t2

# t2 << 1 --> t2 = F0

sll $t2, $t2, 4

# t1 >> 1 --> t1 = DEADBEE

srl $t1, $t1, 4

# t3 = t1 AND 0xF = E

andi $t3, $t1, 0xF

or $t2, $t3, $t2

sll $t2, $t2, 4

srl $t1, $t1, 4

# t3 = t1 AND 0xF = E

andi $t3, $t1, 0xF

or $t2, $t3, $t2

sll $t2, $t2, 4

srl $t1, $t1, 4

# t3 = t1 AND 0xF = B

andi $t3, $t1, 0xF

or $t2, $t3, $t2

sll $t2, $t2, 4

srl $t1, $t1, 4

# t3 = t1 AND 0xF = D

andi $t3, $t1, 0xF

or $t2, $t3, $t2

sll $t2, $t2, 4

srl $t1, $t1, 4

# t3 = t1 AND 0xF = A

andi $t3, $t1, 0xF

or $t2, $t3, $t2

sll $t2, $t2, 4

srl $t1, $t1, 4

# t3 = t1 AND 0xF = E

andi $t3, $t1, 0xF

or $t2, $t3, $t2

sll $t2, $t2, 4

srl $t1, $t1, 4

# t3 = t1 AND 0xF = D

andi $t3, $t1, 0xF

or $t2, $t3, $t2

jr $ra # return to caller (\_\_start)

****

**3.1.4)**

.data # the data segment

.text # the code segment

.globl main

main:

# Load 0xDEADBEEF into $t1

li $t1, 0xDEADBEEF

li $t4, 0xF

# t3 = t1 AND 0xF = F

and $t3, $t1, $t4

# t2 = t3 OR t2 = F

or $t2, $t3, $t2

# t2 << 1 --> t2 = F0

rol $t2, $t2, 4

# t1 >> 1 --> t1 = 0DEADBEE

ror $t1, $t1, 4

and $t3, $t1, $t4

or $t2, $t3, $t2

rol $t2, $t2, 4

ror $t1, $t1, 4

and $t3, $t1, $t4

or $t2, $t3, $t2

rol $t2, $t2, 4

ror $t1, $t1, 4

and $t3, $t1, $t4

or $t2, $t3, $t2

rol $t2, $t2, 4

ror $t1, $t1, 4

and $t3, $t1, $t4

or $t2, $t3, $t2

rol $t2, $t2, 4

ror $t1, $t1, 4

and $t3, $t1, $t4

or $t2, $t3, $t2

rol $t2, $t2, 4

ror $t1, $t1, 4

and $t3, $t1, $t4

or $t2, $t3, $t2

rol $t2, $t2, 4

ror $t1, $t1, 4

and $t3, $t1, $t4

or $t2, $t3, $t2

jr $ra # return to caller (\_\_start)

****

**3.2.1)**

.data # the data segment

.text # the code segment

.globl main

main:

ori $t1, $t1, 1

sll $t2, $t1, 1

sll $t3, $t2, 1

sll $t4, $t3, 1

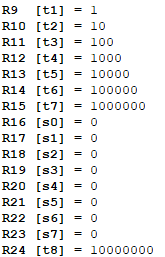
sll $t5, $t4, 1

sll $t6, $t5, 1

sll $t7, $t6, 1

sll $t8, $t7, 1

jr $ra # return to caller (\_\_start)

****

**3.2.2)**

.data # the data segment

.text # the code segment

.globl main

main:

nor $t1, $t1, $t1

jr $ra # return to caller (\_\_start)

****

**Ex 3:**

.data

prompt: .asciiz "Enter the decimal number to convert: "

ans: .asciiz "\nHexadecimal equivalent: "

result: .space 8

.text

.globl main

main:

la $a0, prompt

li $v0, 4

syscall

li $v0, 5

syscall

move $t2, $v0

la $a0, ans

li $v0, 4

syscall

li $t0, 8

la $t3, result

Loop:

beqz $t0, Exit

rol $t2, $t2, 4

and $t4, $t2, 0xf

ble $t4, 9, Sum

addi $t4, $t4, 55

b End

Sum:

addi $t4, $t4, 48

End:

sb $t4, 0($t3)

addi $t3, $t3, 1

addi $t0, $t0, -1

j Loop

Exit:

la $a0, result

li $v0, 4

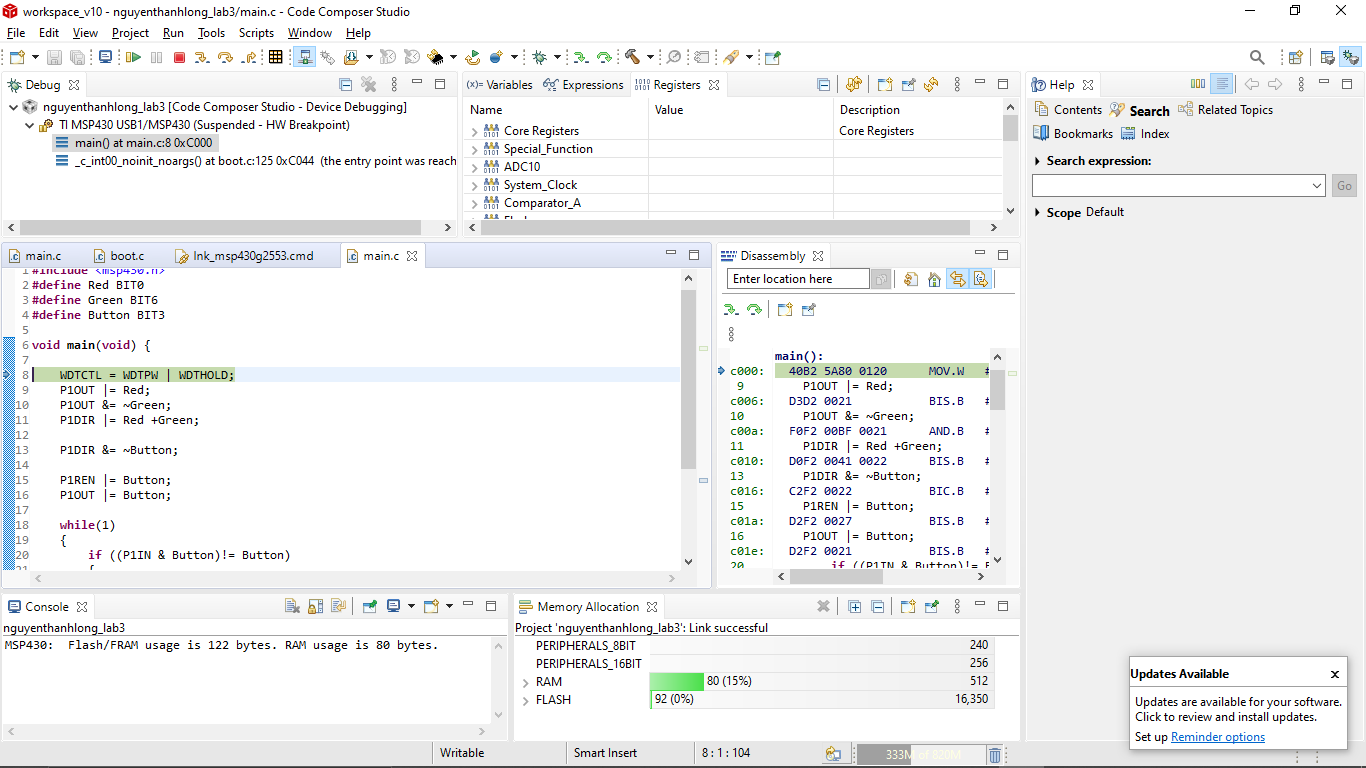
syscall

la $v0, 10

syscall

**II/**

STEP 1 :



STEP 2:

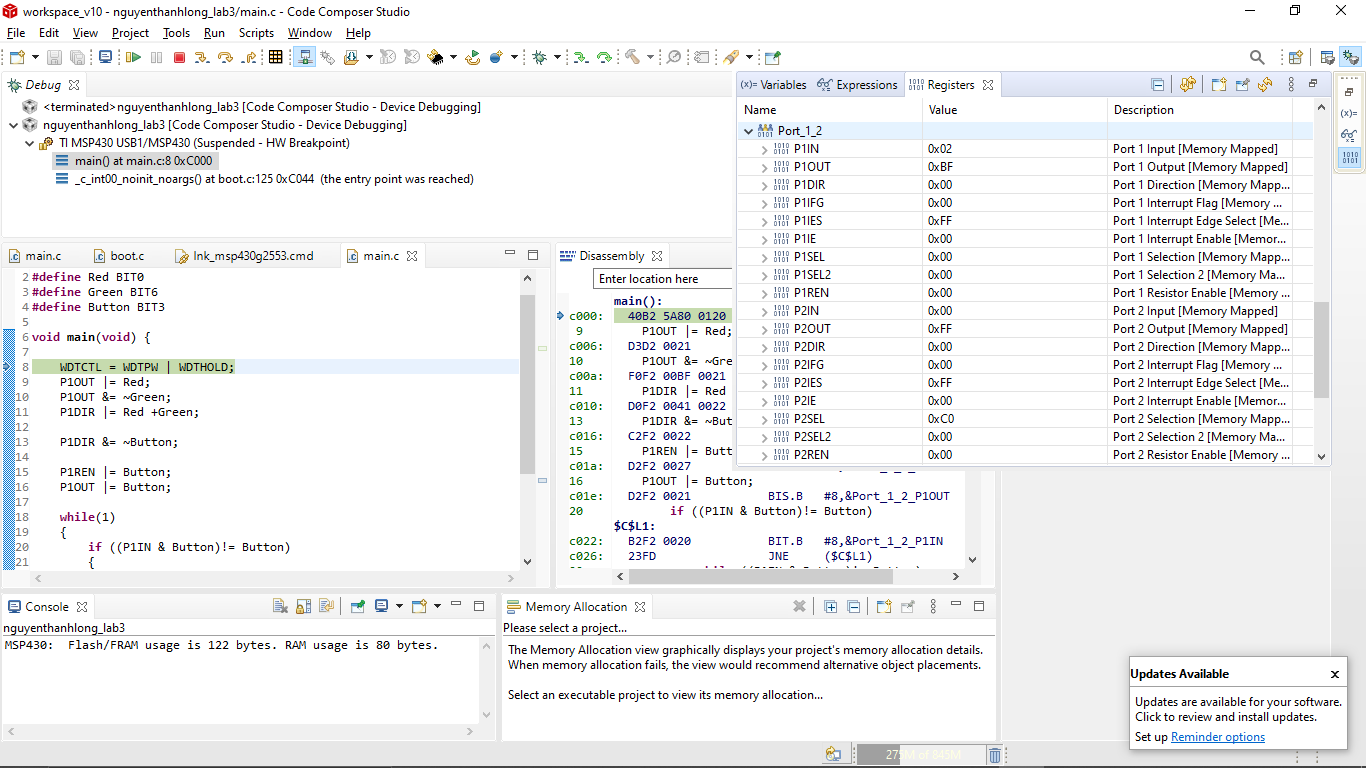
P1OUT: 0xBF

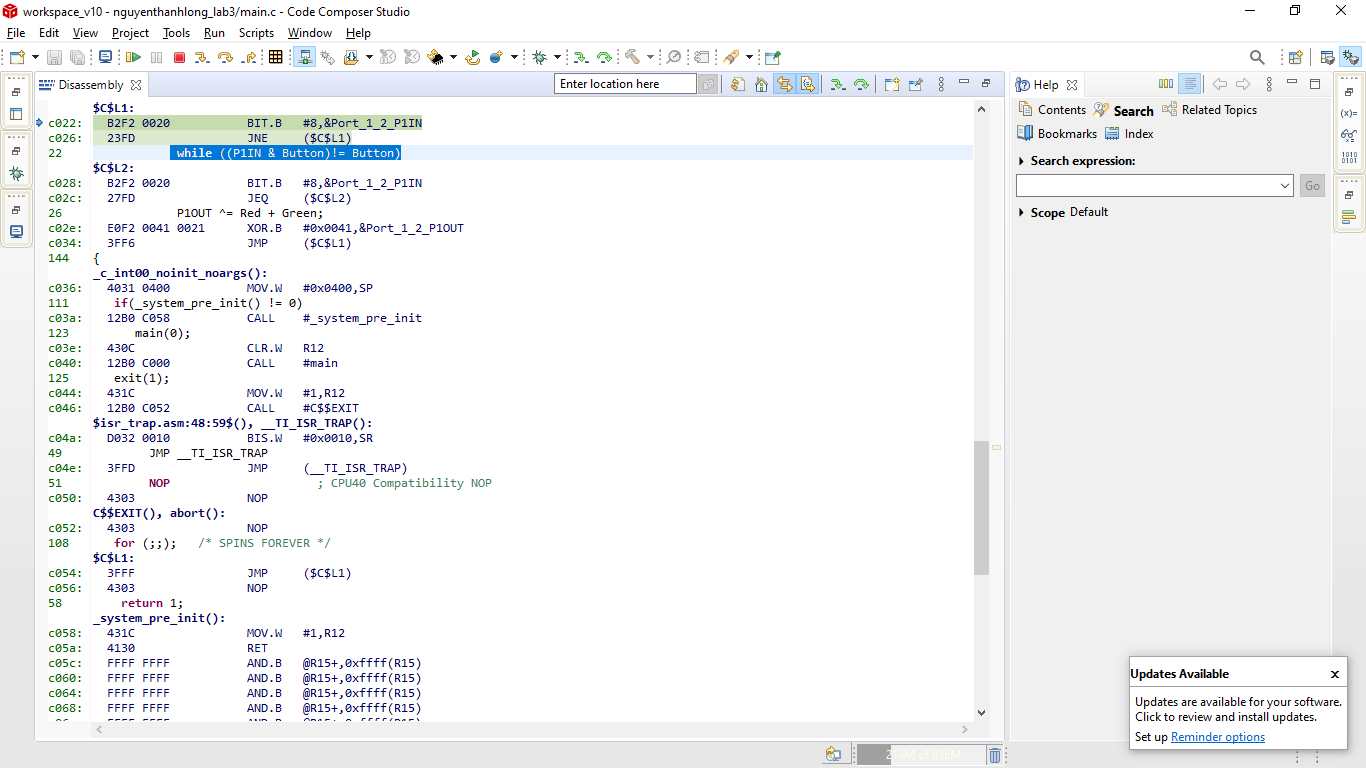
P1IN: 0x02

P1DIR: 0x00

P1REN: 0x00

P1IFG: 0x00





C Code : while ((P1IN & Button)!= Button)

MIPS Code : B2F2 0020 BIT.B #8,&Port\_1\_2\_P1IN

c026: 23FD JNE ($C$L1)