Jett Guerrero

2/16/2019

CPE 301

HW1 – DA1A

DA1A Assembly Code using Iterative Addition

;Jett Guerrero

;2/16/19

;DA1A

.EQU multiplicand = 0xFFFF ;equates variable name multiplicand with hex value 0xFFFF

.EQU multiplier = 0xFF ;equates variable name multiplier with hex value 0xFF

ldi r24, low(multiplicand) ;loads the lower bits of the 16 bit multiplicand in r24

ldi r25, high(multiplicand) ;loads the higher bits of the 16 bit multiplicand in r25

ldi r22, multiplier ;loads the hex value in multiplier into r22

ldi r16, 0x00 ;loads 0 in r16

add\_loop: ;label for looping

add r18, r24 ;add hex value that is in r24 into r18

adc r19, r25 ;add hex value that is in r25 into r19 including any carry

adc r20, r16 ;add any carry if any into r20

dec r22 ;decrement value in r22 to count the number of times addition

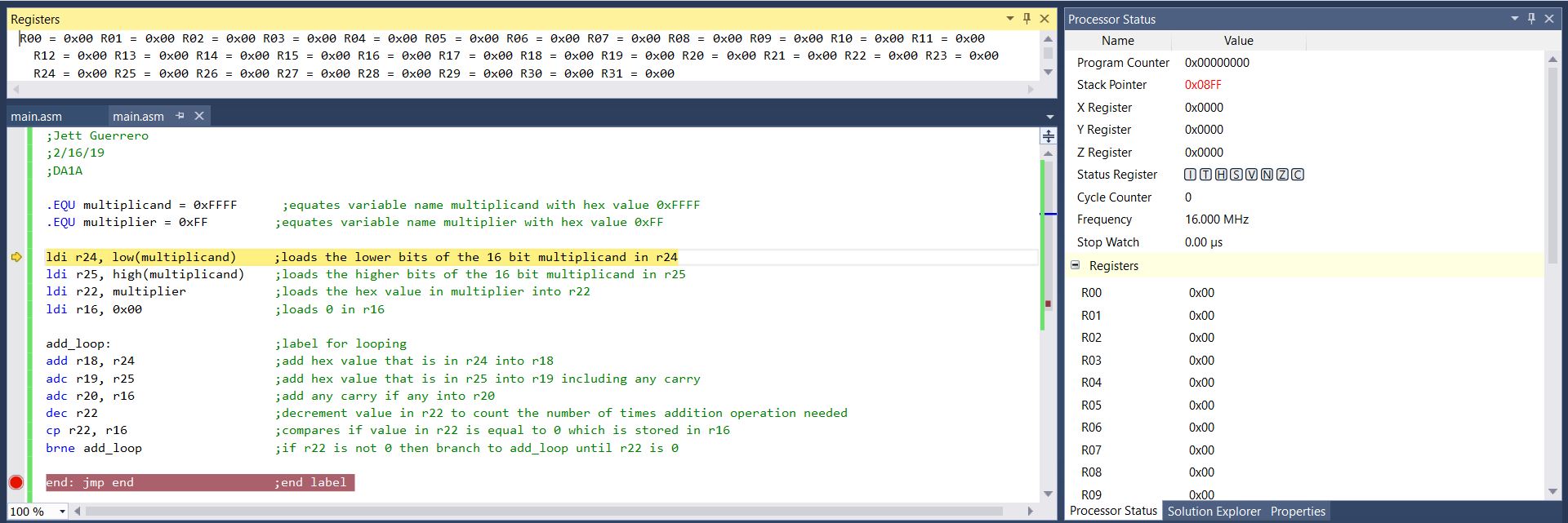
operation needed

cp r22, r16 ;compares if value in r22 is equal to 0 which is stored in r16

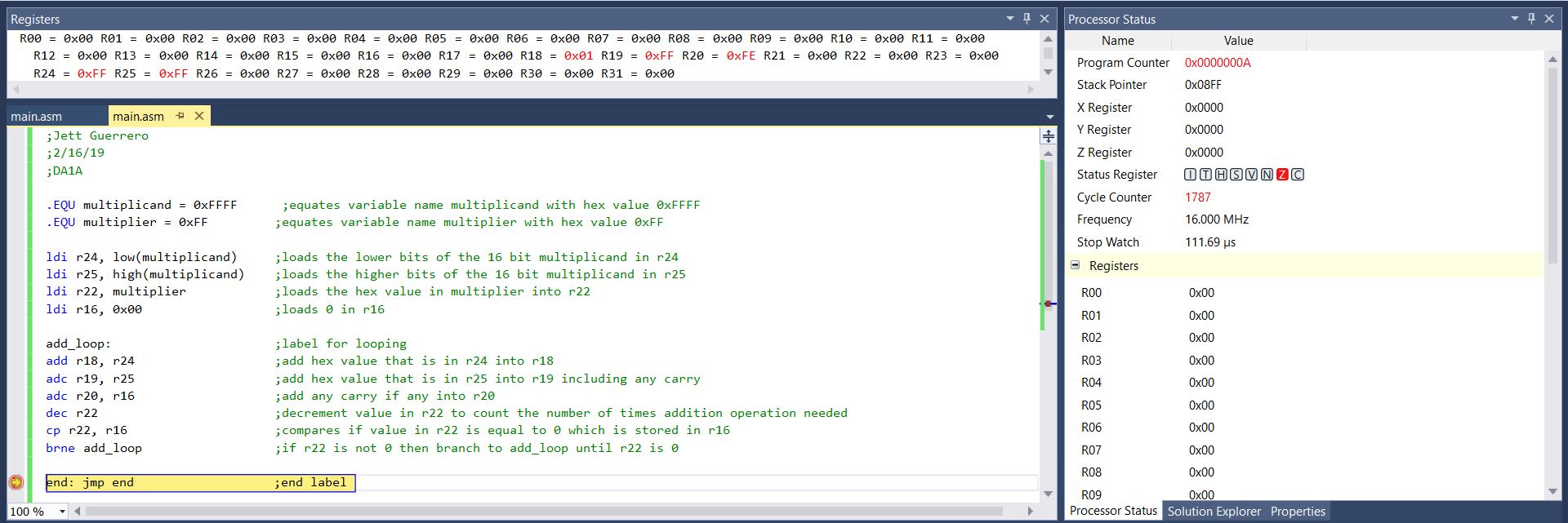
brne add\_loop ;if r22 is not 0 then branch to add\_loop until r22 is 0

end: jmp end ;end label

Debugging Beginning of Task 1



Debugging End of Task 1



Register Results



FFFF x FF = FEFF01

Execution Time @ 16MHz: 111.69us = 0.11ms

Cycle Counter: 1,787

**Verification Using Mul Instruction**

DA1A\_Mul Assembly Code

;Jett Guerrero

;2/16/19

;DA1A\_Mul

.EQU multiplicand = 0xFFFF ;equates variable name multiplicand with hex value 0xFFFF

.EQU multiplier = 0xFF ;equates variable name multiplier with hex value 0xFF

ldi r24, low(multiplicand) ;loads the lower bits of the 16 bit multiplicand in r24

ldi r25, high(multiplicand) ;loads the higher bits of the 16 bit multiplicand in r25

ldi r22, multiplier ;loads the hex value in multiplier into r22

ldi r16, 0x00 ;loads 0 in r16

mul r24, r22 ;multiplies the 8 bit value in r22 with 8 bit value in r24

movw r18:r19,r1:r0 ;moves the 16 bit result that is stored in r1 and r0 into r18 and r19

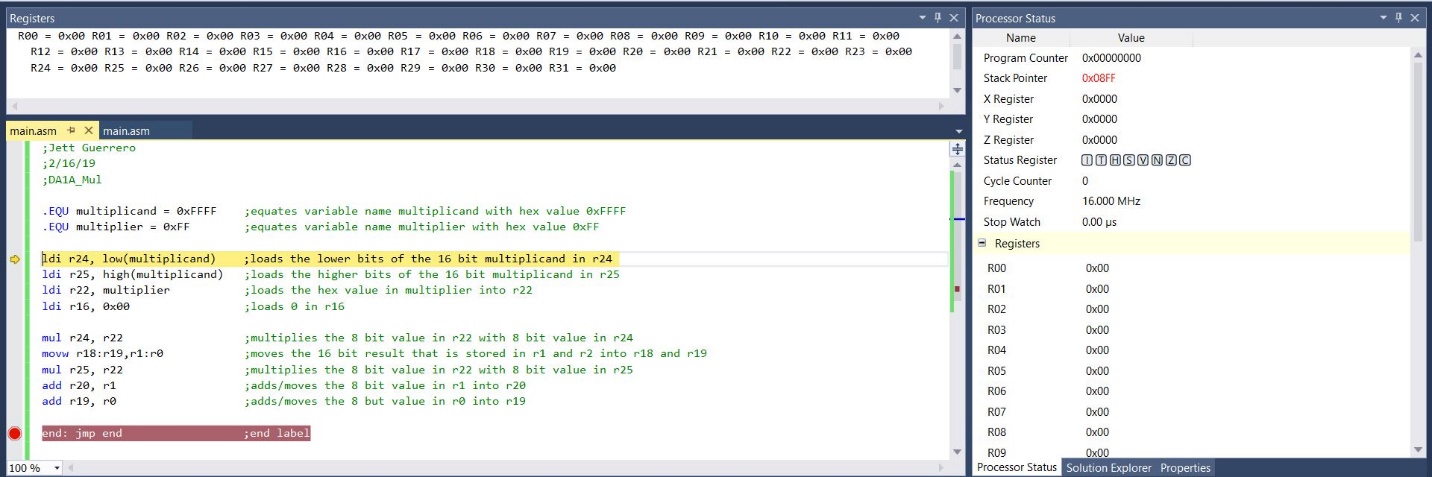
mul r25, r22 ;multiplies the 8 bit value in r22 with 8 bit value in r25

add r20, r1 ;adds/moves the 8 bit value in r1 into r20

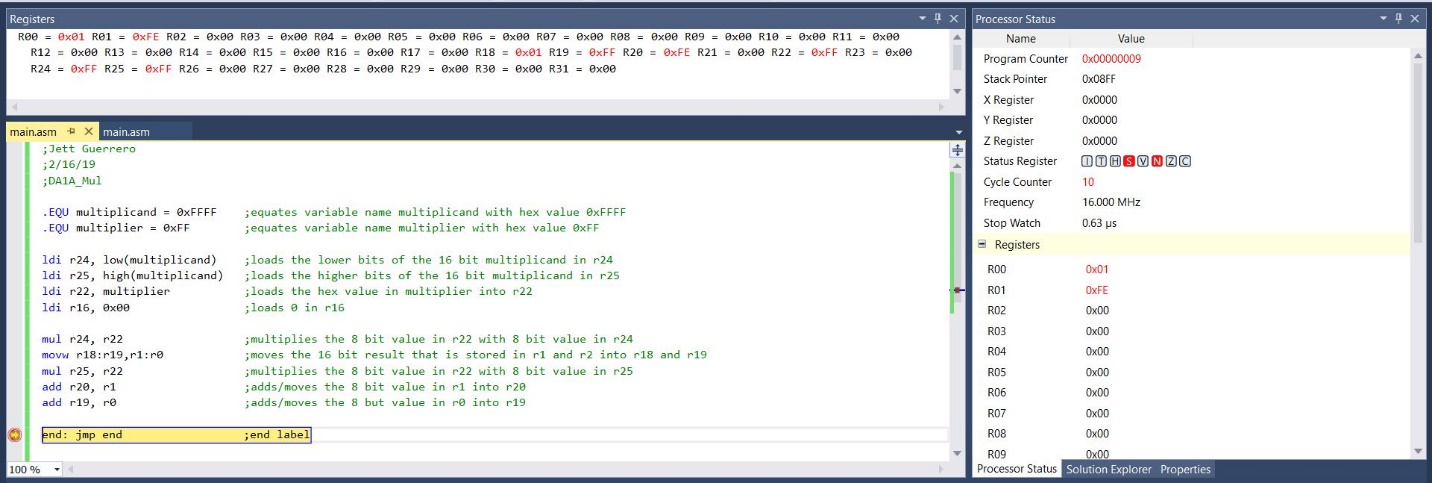
add r19, r0 ;adds/moves the 8 but value in r0 into r19

end: jmp end ;end label

Debugging Beginning of Task 1



Debugging End of Task 1



Register Results



FFFF x FF = FEFF01

Execution Time @ 16MHz: 0.63us

Cycle Counter: 10