David Flores

CPE 301

Homework DA1A

February 16, 2019

1. Perform a multiplication of a 16-bit multiplicand with an 8-bit multiplier without using the MUL instruction. Use iterative addition to perform the above multiplication.

2. Registers R25:R24 hold the 16-bit multiplicand, R22 hold 8-bit multiplier, and R20:R19:R18 should hold the result.

3. Verify your algorithm and answers using the AVR mul instruction or C or any high-level program.

Code for both multiplication using iterative addition and mul function.

Comments: I put both methods on the same project R24 R25 hold the 16-bit multiplicand and R22 holds the multiplier. The answer is stored in R18 R19 and R20 For the second part using mul I use R17 as my multiplier since it gets changed in the previous code. When using mul the bits gets separated into r0 and r1 I move those result into R26 R27 and R28

I multiplied the highest possible value 0xFFFF and 0xFF which gives a result of FE FF 01

;

; DA1A\_MUL.asm

;

; Created: 2/16/2019 10:17:17 PM

; Author : David Flores

; Replace with your application code

.EQU multiplier = 0xFF ;assign value to mulitplier

.EQU multiplicand = 0xFFFF ;assign value to multiplicand

ldi r24, low(multiplicand) ;load low bits of multiplicand to r24

ldi r25, high(multiplicand) ;load high bits of multiplicand to r25

ldi r22, multiplier ;load multiplier bits in r22

ldi r21, 0 ;load value 0 to r21

ldi r17, multiplier ;multiplier 2

loop:

add r18, r24 ;move value in r24 to r18

adc r19, r25 ;add value in r25 into r19 and carry from r18 if necessary

adc r20, r21 ;add value in r21 into r20 and carry from r19 if necessary

dec r22 ;decrement r22

brne loop ;branch if not equal to 0

mul r24, r17 ;multiplies value in r17 with value in r24 and stores in r0 and r1

add r26, r0 ;move value in r0 to r26

add r27, r1 ;move value in r1 to r27

mul r25, r17 ;multiplies value in r17 with value in r28 and stores in r0 and r1

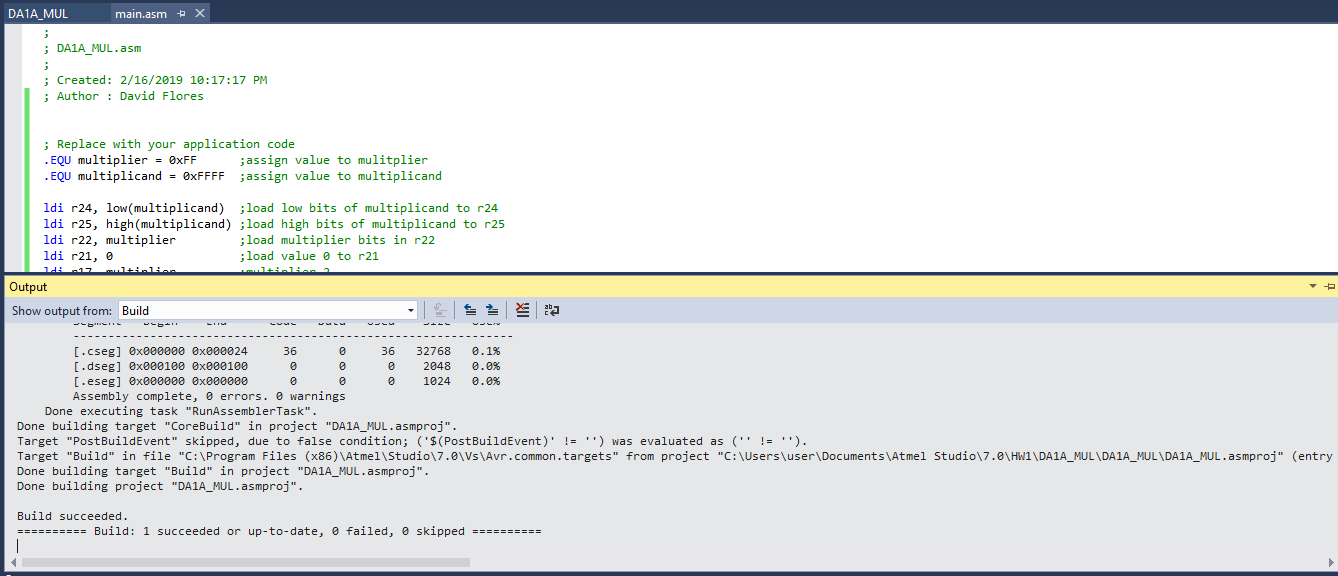
add r28, r1 ;move value in r1 to r28

add r27, r0 ;move value in r0 to r27

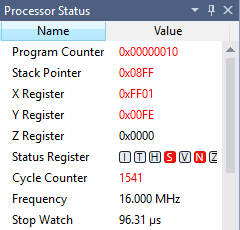
end: jmp end

4. Determine the execution time @ 16MHz/#cycles of your algorithm using the simulation.

Here I built my project



Here is my stop execution time which is 96.31us



Here are my Results

R20 R19 R18 which equal and verified by R28 R27 R26

