CPE301 – SPRING 2019

Design Assignment 1a

Student Name: Do Le

Student #: 2001183621

Student Email: led2@unlv.nevada.edi

Primary Github address: <https://github.com/DoVietLe>

Directory: <https://github.com/DoVietLe/assignments>

1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

No components or pins were used in this assignment.

1. **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A**

.org 0x00

start:

CLR R1 ; Sets the R1 register to 0x00.

CLR R2 ; Sets the R2 register to 0xFF by first

COM R2 ; clearing it, then complementing it.

; MULTIPLICAND

LDI R19, 0x02 ; Holds the 32-bit multiplicand.

LDI R18, 0xAB ; Since the numbers are 32-bits, and

LDI R17, 0x84 ; each register holds 8 bits, four

LDI R16, 0x5C ; registers are used.

; R19 is the most signficant byte, and R16 is the least.

; MULTIPLIER

LDI R23, 0x00 ; Holds the 32-bit multiplier.

LDI R22, 0x00 ; R23 is the most significant byte, and

LDI R21, 0x3B ; R20 is the least.

LDI R20, 0x5F

; RESULT

LDI R31, 0x00 ; Holds the result.

LDI R30, 0x00 ; Multiplying a 32-bit number by a 32-bit

LDI R29, 0x00 ; number can yield up to a 64-bit number,

LDI R28, 0x00 ; so eight registers are used.

LDI R27, 0x00 ; R31 is the most significant byte, and

LDI R26, 0x00 ; R24 is the least.

LDI R25, 0x00

LDI R24, 0x00

loop:

ADD R24, R16 ; Adds the lower most byte.

ADC R25, R17 ; Adds the corresponding spots to the result,

ADC R26, R18 ; respecting the carry.

ADC R27, R19

ADC R28, R1 ; Adds the rest of the position with zeroes,

ADC R29, R1 ; while respecting the carry.

ADC R30, R1

ADC R31, R1

ADD R20, R2 ; Decrements the counter by adding the 2's

ADC R21, R2 ; complement of 1.

ADC R22, R2

ADC R23, R2

CP R1, R20 ; Compares the first byte of the counter to zero.

BRNE loop ; Branches up to the loop if it's not zero.

CP R1, R21 ; Continues the process for the other three

BRNE loop ; bytes in the counter.

CP R1, R22

BRNE loop

CP R1, R23

BRNE loop

end:

RJMP end

1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

No code provided for this assignment.

1. **SCHEMATICS**

No schematics for this assignment.

1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

• Perform a multiplication of a 32-bit multiplicand with a 32-bit multiplier without using the MUL instruction. Use iterative addition to perform the above multiplication.

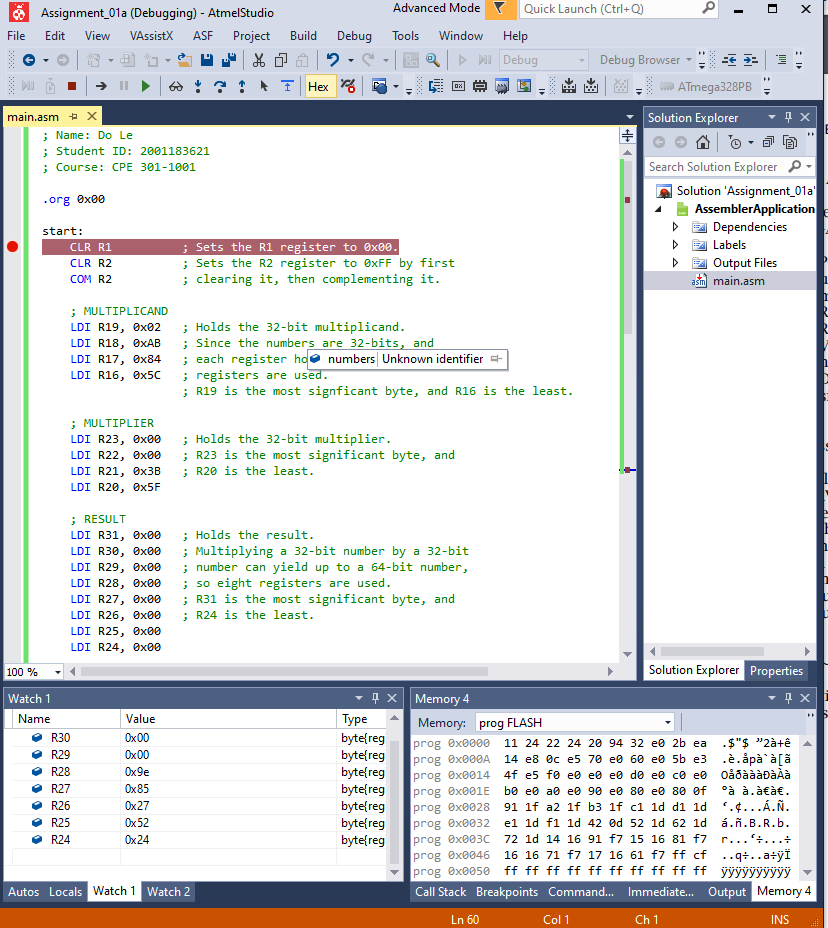


Figure : Program running. In the example I multiply 0x2AB845C with 0x3B5F

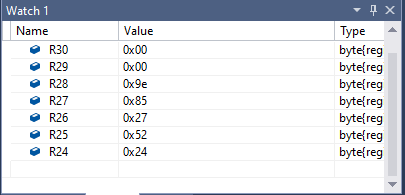


Figure : Result of multiplication stored in registers—0x9E85275224

• Verify your algorithm and answers using the AVRs ‘MUL’ instruction or C or any high-level program.

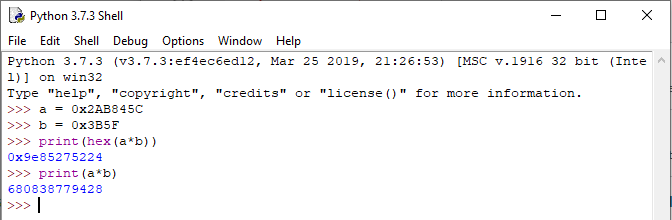


Figure : Verified answer using high-level program

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

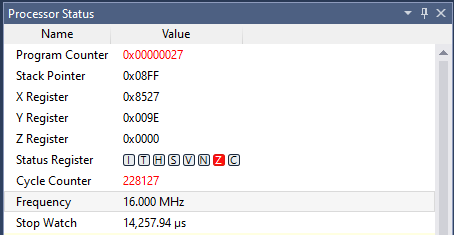


Figure : Program execution time—14,257.94µs

1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**

The board was not used in this assignment.

1. **VIDEO LINKS OF EACH DEMO**

No video recording was made in this assignment.

1. **GITHUB LINK OF THIS DA**

<https://github.com/DoVietLe/assignments/tree/master/ESD301/LAB01a>

**Student Academic Misconduct Policy**

<http://studentconduct.unlv.edu/misconduct/policy.html>

“This assignment submission is my own, original work”.

Do V. Le