Michael Ghisilieri

CPE301 – SPRING 2016

Design Assignment 1

**DO NOT REMOVE THIS PAGE DURING SUBMISSION:**

The student understands that all required components should be submitted in complete for grading of this assignment.

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **SUBMISSION ITEM** | **COMPLETED (Y/N)** | **MARKS**  **(/MAX)** |
| 0. | INITIAL CODE OF TASK 1 (with every section) |  |  |
| 1. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |
| 2. | GITHUB LINK OF THE DA |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | INITIAL CODE OF TASK 1 (with every section) |  |  |

;

; DA1\_Task1.asm

;

; Created: 2/21/2016 8:15:50 PM

; Author : Michael

;

; Macro that initializes stack pointer

.MACRO INITSTACK

LDI R16, HIGH(RAMEND)

OUT SPH, R16

LDI R16, LOW(RAMEND)

OUT SPL, R16

.ENDMACRO

INITSTACK ; Call macro to initialize stack

LDI R21, 0x19 ; Value of 25 for counter to store numbers

LDI XH, HIGH(RAMEND) ; Load high bits of RAMEND into X pointer

LDI XL, LOW(RAMEND) ; Load low bits of RAMEND into X pointer

MOV R20, XH ; Move XH into R20 for comparisons

ANDI R20, 0x01 ; Keep LSB of XH

LSR XH ; Divide XH by 2

LSR XL ; Divide XL by 2

CPI R20, 1 ; Compare LSB of XH with 1

BRNE ClearHigh ; If not equal, branch to CLEARHIGH

ORI XL, 128 ; Set MST of SL high for proper RAMEND/2

RJMP Store ; Jump to STORE to skip CLEARHIGH label

ClearHigh:

ANDI XL, 127 ; Clear MSB of XL for proper RAMEND/2

Store:

MOV R22, XL ; Move lower 8 bits of address into R22

ST X+, R22 ; Store the 8 bit value into memory

DEC R21 ; Decrement counter

BRNE Store ; It equal to zero (25th loop) drop out of loop

; Beginning of Task B and C

LDI R16, 0x19 ; Value of 25 for counter to count sums

LDI R20, 0x00 ; Sum for divisible by 7 (low)

LDI R21, 0x00 ; Sum for divisible by 7 (high)

LDI R23, 0x00 ; Sum for divisible by 3 (low)

LDI R24, 0x00 ; Sum for divisible by 3 (high)

Divisible:

LD R17, -X ; Load top value from stack of 25 numbers

MOV R18, R17 ; Hold copy of current value

Seven:

SUBI R18, 7 ; Subtract 7 from current value

BRCS DoneSeven ; If value is negative, not divisible by 7

CPI R18, 0 ; Compare if value is currently 0

BREQ AddToSeven ; If equal to zero, divisible by 7

RJMP Seven

AddToSeven:

ADD R20, R17 ; Add value to running sum

BRCS CarrySeven ; If carry bit occured, jump to CarrySeven

RJMP DoneSeven ; Skip increment of R21 for no carry bit

CarrySeven:

INC R21 ; Increment R21 for carry bit in R20

CLC ; Clear carry flag

DoneSeven:

CLC

MOV R18, R17 ; Reset current value for divisible by 3 check

Three:

SUBI R18, 3 ; Subtract 3 from current value

BRCS DoneThree ; If value is negative, not divisible by 3

CPI R18, 0 ; Compare if value is currently 0

BREQ AddToThree ; If equal to zero, divisible by 3

RJMP Three

AddToThree:

ADD R23, R17 ; Add value to running sum

BRCS CarryThree ; If carry bit occured, jump to CarryThree

RJMP Three ; Skip increment of R24 for no carry bit

CarryThree:

INC R24 ; Increment R24 for carry bit in R23

CLC ; Clear carry flag

DoneThree:

CLC

DEC R16 ; Decrement counter

BRNE Divisible ; It equal to zero (25th loop) drop out of loop

; Beginning of Task D

CPI R21, 0 ; Check if high register of divisible by 7 is 0

BRNE Greater ; Jump to Greater and set R7.3 high

LDI R16, 0 ; Move zero into R16

MOV R7, R16 ; Set R7.3 to zero

RJMP Done ; Go to end and stay in infinite loop

CPI R23, 0 ; Check if high register of divisible by 3 is 0

BREQ Greater ; Jump to Greater and set R7.3 high

LDI R16, 0 ; Move zero into R16

MOV R7, R16 ; Set R7.3 to zero

RJMP Done ; Go to end and stay in infinite loop

Greater:

LDI R16, 8 ; Move 8 into R16

MOV R7, R16 ; Set R7.3 high

Done:

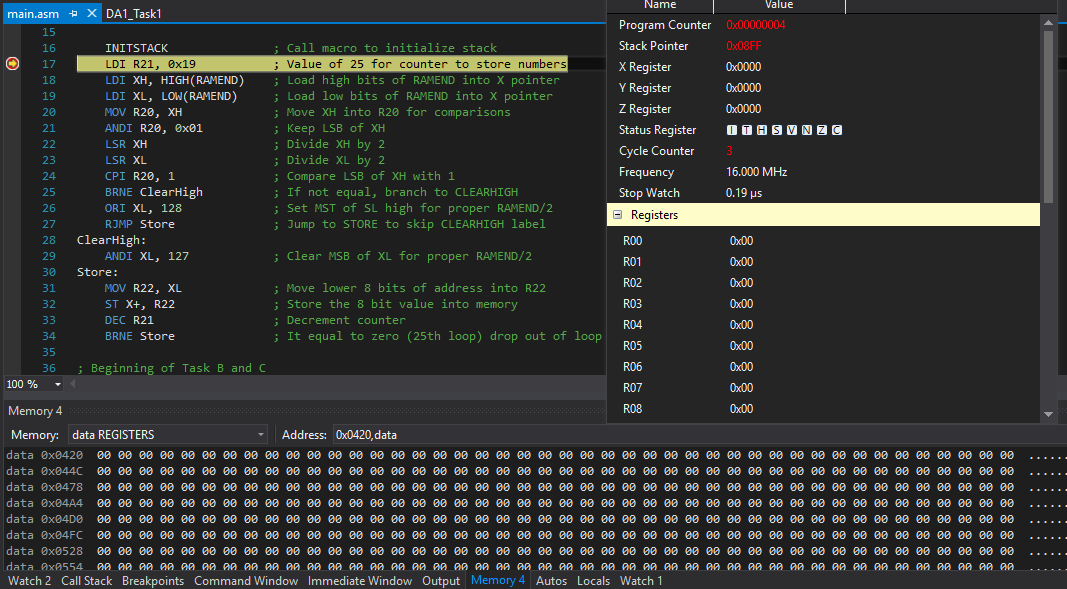
RJMP Done ; Infinite loop

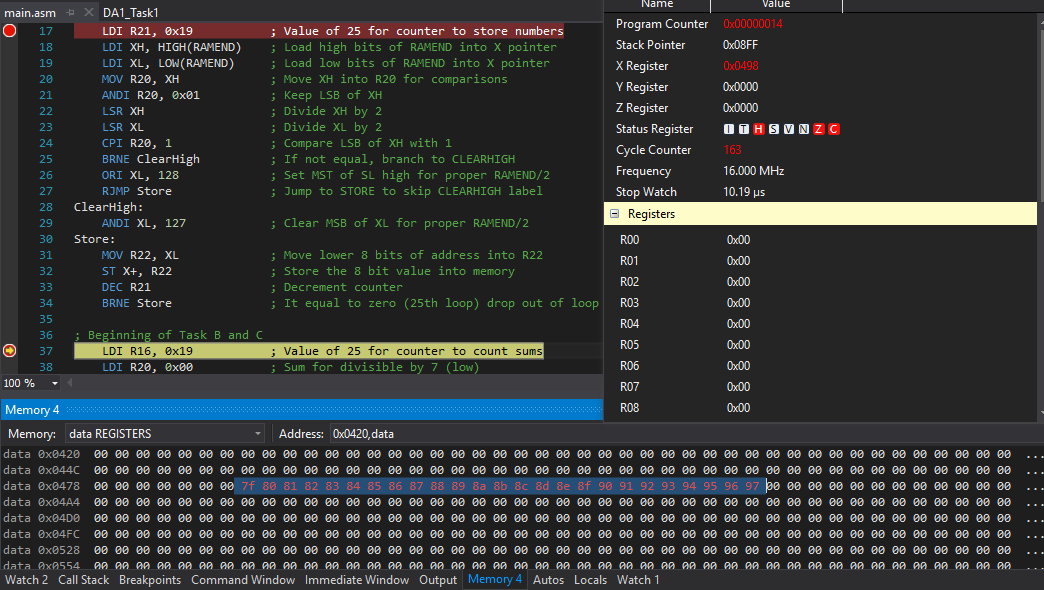
|  |  |  |  |
| --- | --- | --- | --- |
| 2. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |

TASK 1/A:

Store 25 numbers starting from the RAMEND/2 location. Use the X/Y/Z registers as pointers.

**The highlighted row in the second image shows the 25 numbers loaded into memory.**



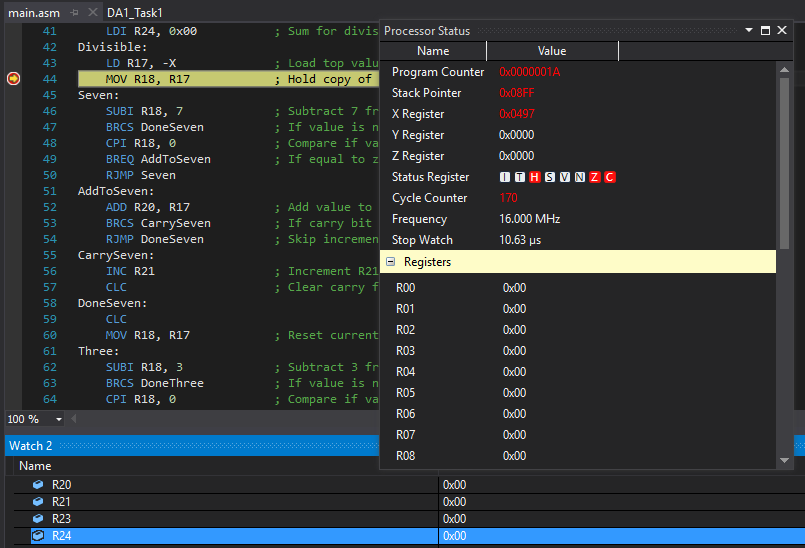


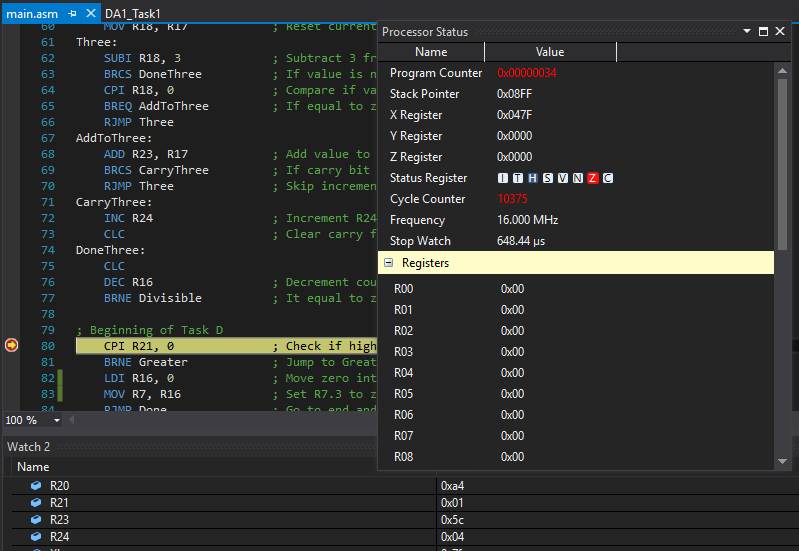
TASK 1/B and 1/C:

Add all numbers divisible by 7 and place the result in R20:21, divisible by 3 and place in R23:24.

Parsing of the numbers is done simultaneously.

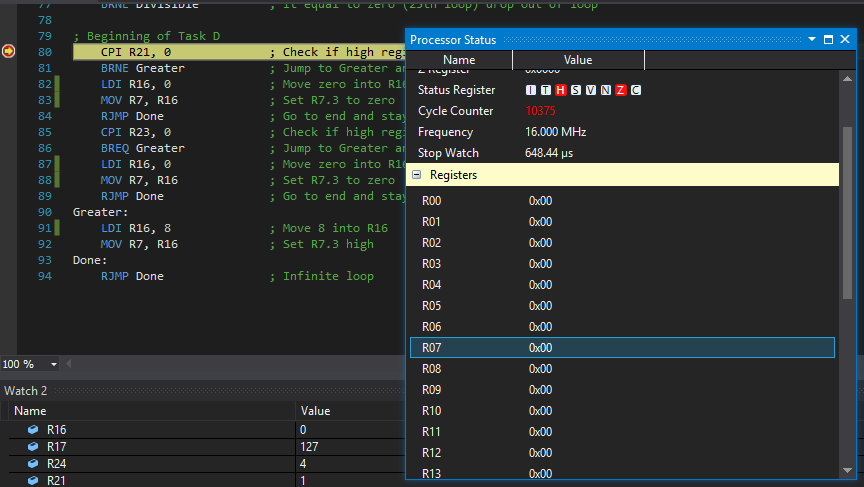
**Sum of numbers shown in Watch2 window.**

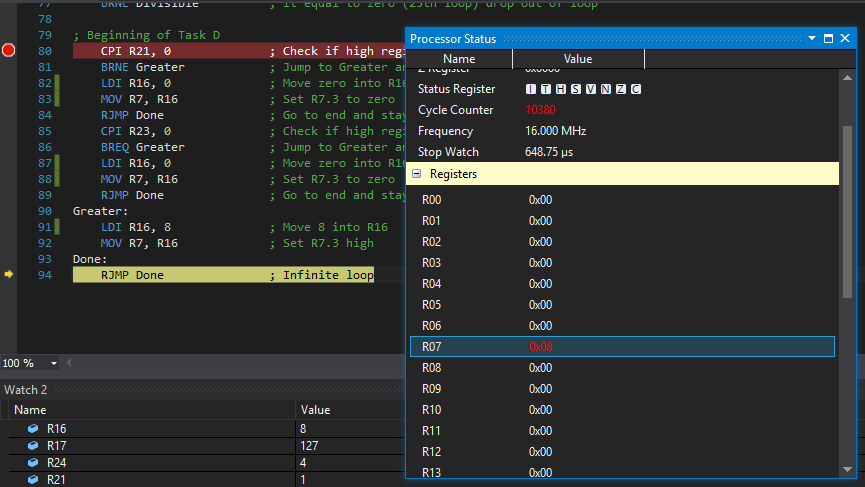




TASK 1/D:

Check and set register R7.3 high if either sum from TASK 1/B or 1/C is greater than 8-bits.

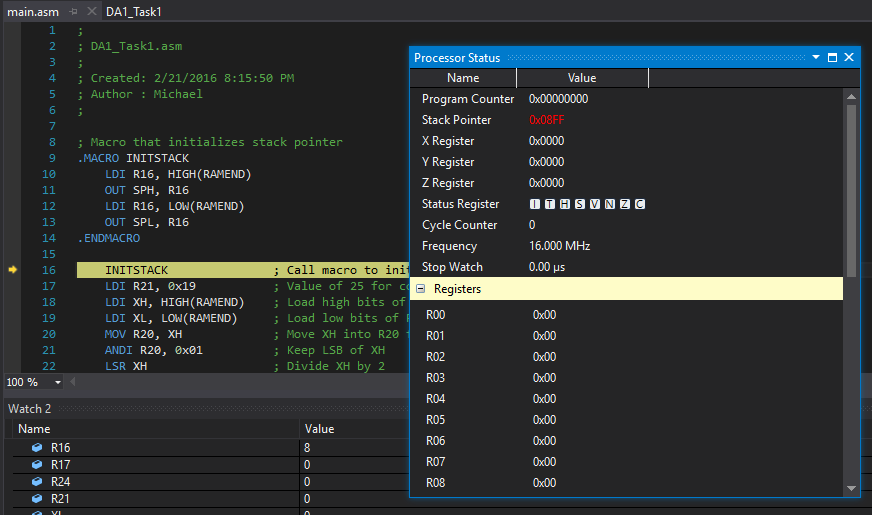


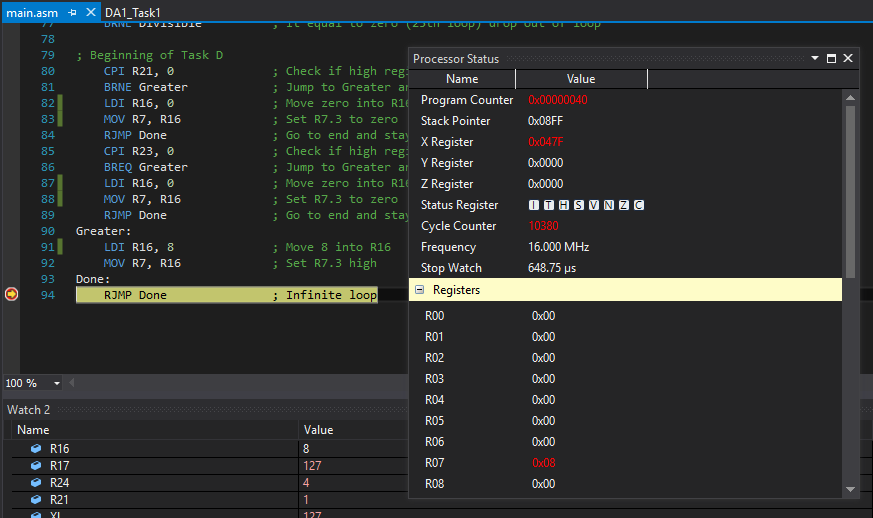


TASK 1/E:

Determine the execution time @ 16MHz / # cycles.

10,850 cycles / 16MHz = **648.75 µs**





|  |  |  |  |
| --- | --- | --- | --- |
| 10. | GITHUB LINK OF THE DA |  |  |
| https://github.com/michael-ghisilieri/CpE301\_DAs.git | | | |

**Student Academic Misconduct Policy**

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

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

Michael Ghisilieri