Desi Battle

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. | COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS |  |  |
| 1. | INITIAL CODE OF TASK 1/A |  |  |
| 2. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 2/B |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 3/C |  |  |
| 4. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 4/D |  |  |
| 5. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 5/E |  |  |
| 6. | SCHEMATICS |  |  |
| 7. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |
| 8. | SCREENSHOT OF EACH DEMO |  |  |
| 9. | VIDEO LINKS OF EACH DEMO |  |  |
| 10. | GOOGLECODE LINK OF THE DA |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 0. | COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | INITIAL CODE OF TASK 1/A |  |  |

Code to push 25 numbers onto RAM to be traversed later

.org 0

;initialize stack and set x as pointer to stack

;INITSTACK

ldi xh, high(RAMEND)

ldi xl, low(RAMEND)

;out sph, xh

;out spl, xl

;shift xh and xl right one bit to simulate division by 2

;due to values being known, no need for overflow checking

lsr xh

lsr xl

;initialize r17 to 1 to be used to increment X

ldi r17, 1

ldi r18, 25

;load a value from address x points to, push it on stack, repeat 25 times

keepgoing:

mov r16, xl

st X, r16

add Xl, r17

dec r18

brbs 1, valsread ;break when z flag set as values have been read

rjmp keepgoing

valsread:

;sub 1 from x1 so that it points to the last of the values read

sub Xl, r17

|  |  |  |  |
| --- | --- | --- | --- |
| 2. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 2/B |  |  |

Code to traverse numbers and add them to appropriate running sums depending on if they are divisible by either 3 or 7

;r17 =1 (still)

;r18 = 25

;r20:r21 == 7 sum, initialized to 0

;r23:r24 == 3 sum, initialized to 0

ldi r18, 25

ldi r20, 0

ldi r21, 0

ldi r23, 0

ldi r24, 0

;beginning of loop

newNum:

mov r19, xl ;set r19 as the #

;The N flag is set to high whenever the msb of a register is high after an operation

;to correct for this I will subtract 133 from any number greater than or eq to 133

;prior to subrtacting 7 and checking for negativeness

cpi r19, 133

brlt noProb7

subi r19, 133

noProb7:

;subtract 7 until either negative or 0 is found then jmp to appropriate loop

sub7:

subi r19,7

brbs 1, divis7

brbs 2, notdivis7

rjmp sub7 ;continue to subtract until 0 or negative is generated

divis7:

add r21, xl ;add number to running sum r21

brcs ovflo7

rjmp notdivis7 ;if no carry generated when item added, finish check

ovflo7:

add r20, r17 ;add 1 to upper portion of 7sum if carry was generated

notdivis7:

mov r19, xl ;repeat routine for 3

cpi r19, 129

brlt noprob3

subi r19, 129

noprob3:

;subtract 3 until either a negative or 0 is discovered

sub3:

subi r19, 3

brbs 1, divis3

brbs 2, notdivis3

rjmp sub3

divis3:

add r24, xl

brcs ovflo3

rjmp notdivis3

ovflo3:

add r23, r17

notdivis3:

subi xl, 1

dec r18

brbs 1, sumdone

jmp newNum

sumdone:

|  |  |  |  |
| --- | --- | --- | --- |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 3/C |  |  |

Code to set r7.4 to 1 if either of the sums is over 8 bits (127)

;if either of the sums has overflowed set r7.3 to 1 (0x08)

cpi r20, 0

breq noCarry

cpi r23, 0

breq noCarry

ldi r19, 0x08

mov r7, r19

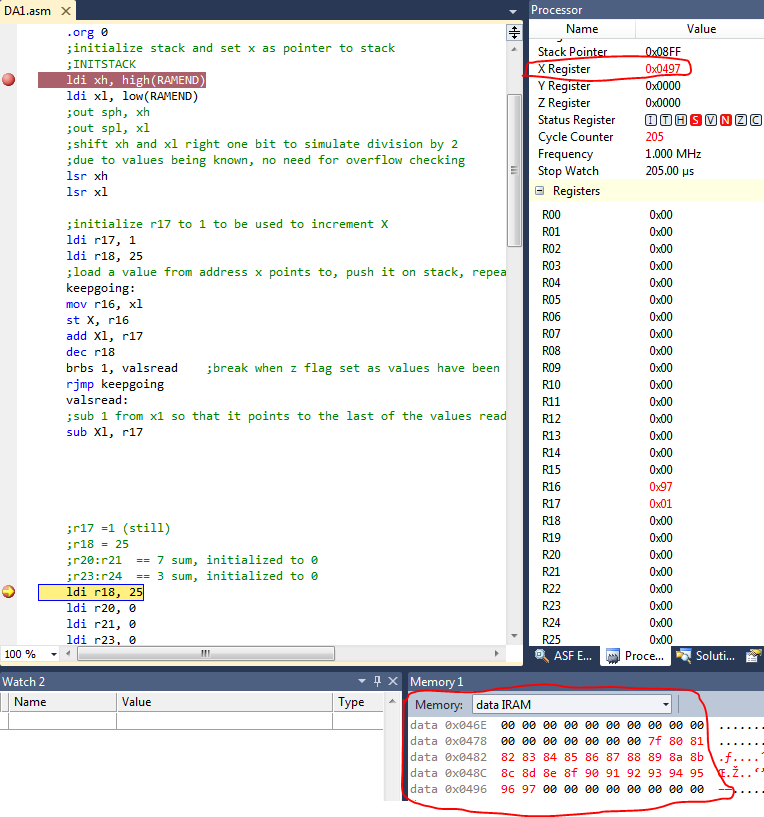
noCarry:

done: rjmp done

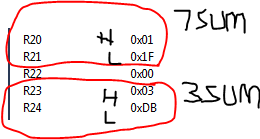
|  |  |  |  |
| --- | --- | --- | --- |
| 6. | SCHEMATICS |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 7. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |

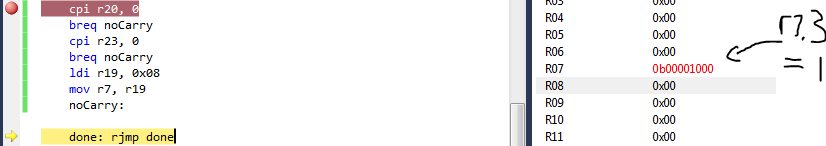
Output of first task showing numbers stored in IRAM and xreg is pointing to middle of RAM



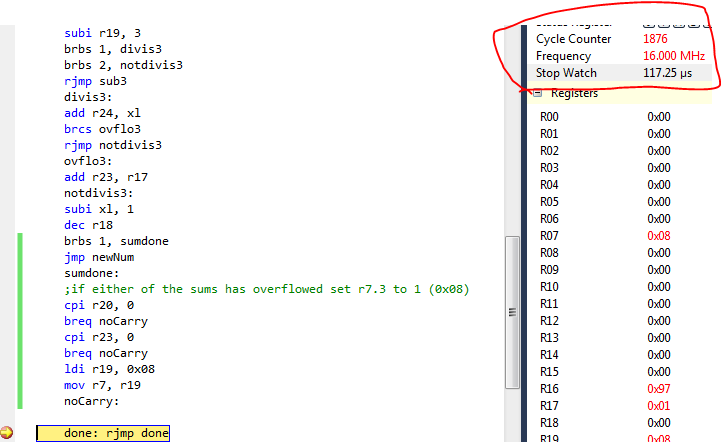
Screen shot of 7sum and 3sum registers after appropriate numbers added to them



Screen shot showing r7.3 set to 1 after execution



Screen shot showing execution time, cycles, and clock freq.



|  |  |  |  |
| --- | --- | --- | --- |
| 8. | SCREENSHOT OF EACH DEMO |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 9. | VIDEO LINKS OF EACH DEMO |  |  |
|  | | | |
| 10. | GOOGLECODE LINK OF THE DA |  |  |
| https://github.com/battled/DA0.git | | | |

**Student Academic Misconduct Policy**

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

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

Desi Battle