Derek Nhan

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. | CODE SEGMENT OF TASK 1/A |  |  |
| 2. | CODE SEGMENT OF TASK 2/B/C |  |  |
| 3. | CODE SEGMENT OF TASK 3/D |  |  |
| 4. | COMPLETE CODE |  |  |
| 5. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |
| 6. | FLOW CHART OF ALGORITHM |  |  |
| 7. | VIDEO LINKS OF EACH DEMO |  |  |
| 8. | GOOGLECODE LINK OF THE DA |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

;Code segment that puts 25 numbers on to the stack

.def COUNT=r25 ;counter

.def dividend=r22 ;dividend register

.def number=r12 ;number to be added is divided

.def SUM\_7H=r20 ;high of sum of 7

.def SUM\_7L=r21 ;low of sum of 7

.def SUM\_3H=r23 ;high of sum of 3

.def SUM\_3L=r24 ;low of sum of 3

.def OVERFLOW=r7 ;overflow register for sum

.macro STACK

ldi @0, high(@1)

out SPH, @0

ldi @0, low(@1)

out SPL, @0

.endmacro

STACK r16, RAMEND

ldi XH, high(RAMEND/2) ;set X pointer to high bits of middle of ramend

ldi XL, low(RAMEND/2) ;set X pointer to low bits of middle of ramend

ldi COUNT, 0 ;set counter to 0

loop:

;loop to store numbers in to RAMEND/2 location

ldi r17, low(RAMEND/2)

add r17, COUNT

st X+, r17

inc COUNT

cpi COUNT, 25

brne loop

|  |  |  |  |
| --- | --- | --- | --- |
| 2. | INITIAL CODE OF TASK 1/B |  |  |

;Code segment that parses the numbers and check division by 7 and 3 and adds the ;corresponding values

ldi XH, high(RAMEND/2)

ldi XL, low(RAMEND/2)

ldi YH, high(RAMEND/2)

ldi YL, low(RAMEND/2)

ldi ZH, high(RAMEND/2)

ldi ZL, low(RAMEND/2)

again:

ld number, Z+ ;loads number in to the number var

ld dividend, X+ ;loads number to the dividend to be divided

division7:

;loop to divide number by 7

subi dividend, 7

cpi dividend, 7

brsh division7

cpi dividend, 0

ld dividend, Y+ ;if remainder is 0, then the number is divisible by 7

breq sum\_7

division3:

;loop to divide number by 3

subi dividend, 3

cpi dividend, 3

brsh division3

cpi dividend, 0 ;if remainder is 0, then the number is divisible by 3

breq sum\_3

div\_lp:

dec COUNT

cpi COUNT, 0 ;count of the numbers already used

brne again

jmp done

sum\_7:

;calculates the sum for division by 7

add SUM\_7L, number

brvs ovr\_flw7

jmp division3

sum\_3:

;calculates the sum for division by 3

add SUM\_3L, number

brvs ovr\_flw3

jmp div\_lp

|  |  |  |  |
| --- | --- | --- | --- |
| 3. | INITIAL CODE OF TASK 1/D |  |  |

;Code segment to set overflow register

ovr\_flw7:

;both labels will set overflow register is the sum is greater that 8 bits

ldi r17, 0x08

mov OVERFLOW, r17 ;copies r17 to OVERFLOW(r7) register and set bit 3

subi SUM\_7H, -1

jmp division3

ovr\_flw3:

ldi r17, 0x08

mov OVERFLOW, r17

subi SUM\_3H, -1

jmp div\_lp

|  |  |  |  |
| --- | --- | --- | --- |
| 4. | Complete Code |  |  |

.def COUNT=r25 ;counter

.def dividend=r22 ;dividend register

.def number=r12 ;number to be added is divided

.def SUM\_7H=r20 ;high of sum of 7

.def SUM\_7L=r21 ;low of sum of 7

.def SUM\_3H=r23 ;high of sum of 3

.def SUM\_3L=r24 ;low of sum of 3

.def OVERFLOW=r7 ;overflow register for sum

.macro STACK

ldi @0, high(@1)

out SPH, @0

ldi @0, low(@1)

out SPL, @0

.endmacro

STACK r16, RAMEND

ldi XH, high(RAMEND/2) ;set X pointer to high bits of middle of ramend

ldi XL, low(RAMEND/2) ;set X pointer to low bits of middle of ramend

ldi COUNT, 0 ;set counter to 0

loop:

;loop to store numbers in to RAMEND/2 location

ldi r17, low(RAMEND/2)

add r17, COUNT

st X+, r17

inc COUNT

cpi COUNT, 25

brne loop

;set the X,Y,Z pointers to the first number on the stack

ldi XH, high(RAMEND/2)

ldi XL, low(RAMEND/2)

ldi YH, high(RAMEND/2)

ldi YL, low(RAMEND/2)

ldi ZH, high(RAMEND/2)

ldi ZL, low(RAMEND/2)

again:

ld number, Z+ ;loads number in to the number var

ld dividend, X+ ;loads number to the dividend to be divided

division7:

;loop to divide number by 7

subi dividend, 7

cpi dividend, 7

brsh division7

cpi dividend, 0

ld dividend, Y+ ;if remainder is 0, then the number is divisible by 7

breq sum\_7

division3:

;loop to divide number by 3

subi dividend, 3

cpi dividend, 3

brsh division3

cpi dividend, 0 ;if remainder is 0, then the number is divisible by 3

breq sum\_3

div\_lp:

dec COUNT

cpi COUNT, 0 ;count of the numbers already used

brne again

jmp done

sum\_7:

;calculates the sum for division by 7

add SUM\_7L, number

brvs ovr\_flw7

jmp division3

sum\_3:

;calculates the sum for division by 3

add SUM\_3L, number

brvs ovr\_flw3

jmp div\_lp

ovr\_flw7:

;both labels will set overflow register is the sum is greater that 8 bits

ldi r17, 0x08

mov OVERFLOW, r17 ;copies r17 to OVERFLOW(r7) register and set bit 3

subi SUM\_7H, -1

jmp division3

ovr\_flw3:

ldi r17, 0x08

mov OVERFLOW, r17

subi SUM\_3H, -1

jmp div\_lp

done:

|  |  |  |  |
| --- | --- | --- | --- |
| 5. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |

Task 1/A

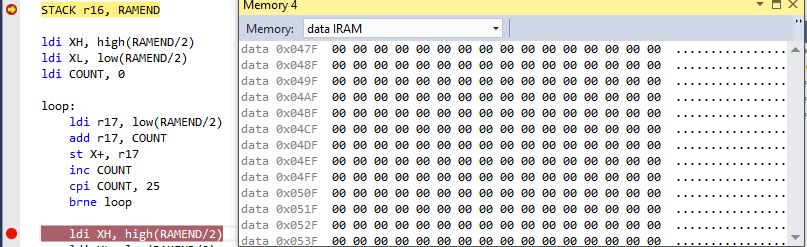


Figure 1: Before storing values

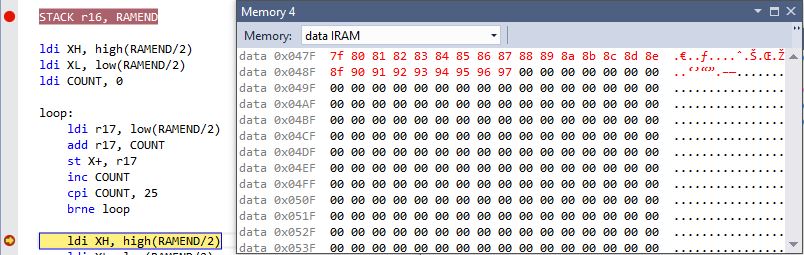


Figure 2: After storing values

Task 1/B/C/D

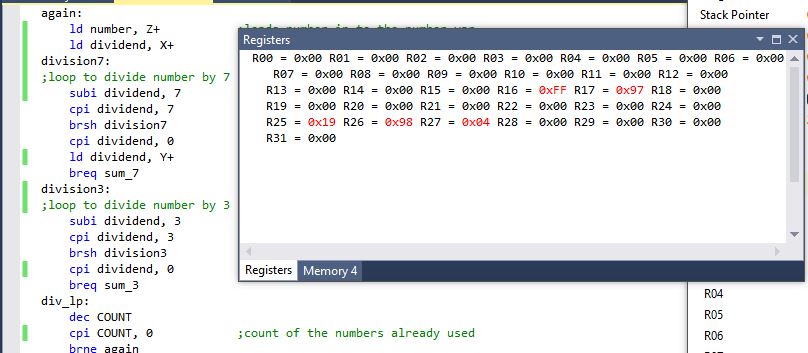


Figure 3: Task B,C,D before performing arithmetic/parse

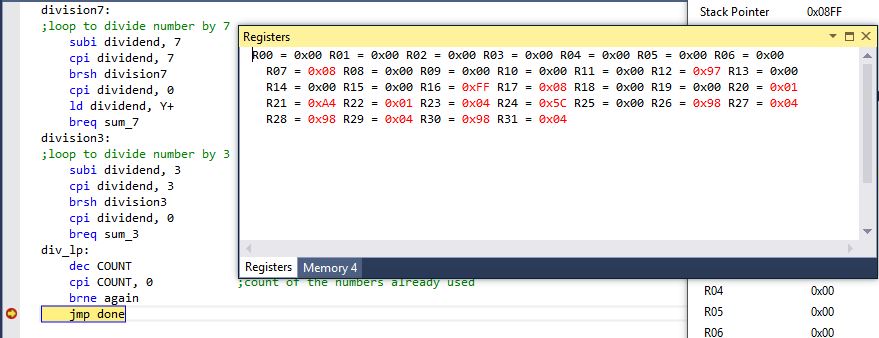


Figure 4: Task B,C,D after performing arithmetic/parse

Task 1/E

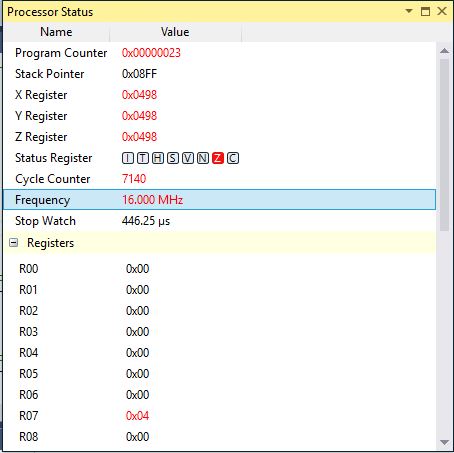


Figure 5: Clock at 16MHz, execution done in 446.25 microseconds

|  |  |  |  |
| --- | --- | --- | --- |
| 7. | FLOW CHART OF ALGORITHM |  |  |

Store number at mem location + count

Increment count

NO

Count=25?

Get value at pointer location then increment pointers X, Z

YES

Add number to sum\_7

Divisible by 7?

NO

NO

Get value at pointer Y and increment Y

YES

Divisible by 3?

Add number to sum\_3

NO

Sum>8bits?

NO

Decrement COUNT

Count=0?

YES

Set overflow

YES

DONE

|  |  |  |  |
| --- | --- | --- | --- |
| 9. | VIDEO LINKS OF EACH DEMO |  |  |
| http:// @youtube | | | |
| 10. | GOOGLECODE LINK OF THE DA |  |  |
| hhttps://github.com/nhand2/CPE301S16.git | | | |

**Student Academic Misconduct Policy**

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

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

Derek Nhan