Yannick Kengne Tatcha

CPE301 – SPRING 2018

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

1.	INITIAL CODE OF Task1		

```
;Code segment that puts 300 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_5H=r23              ;high of sum of 5
.def SUM_5L=r24                   ;low of sum of 5
.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
```

2.	INITIAL CODE OF Task2		

```
;Code segment that parses the numbers and check division by 5
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

division5:
;loop to divide number by 5
sum_5:
;calculates the sum for division by 5
add SUM_5L, number
brvs ovr_flw5
```

3.	INITIAL CODE OF Task3		

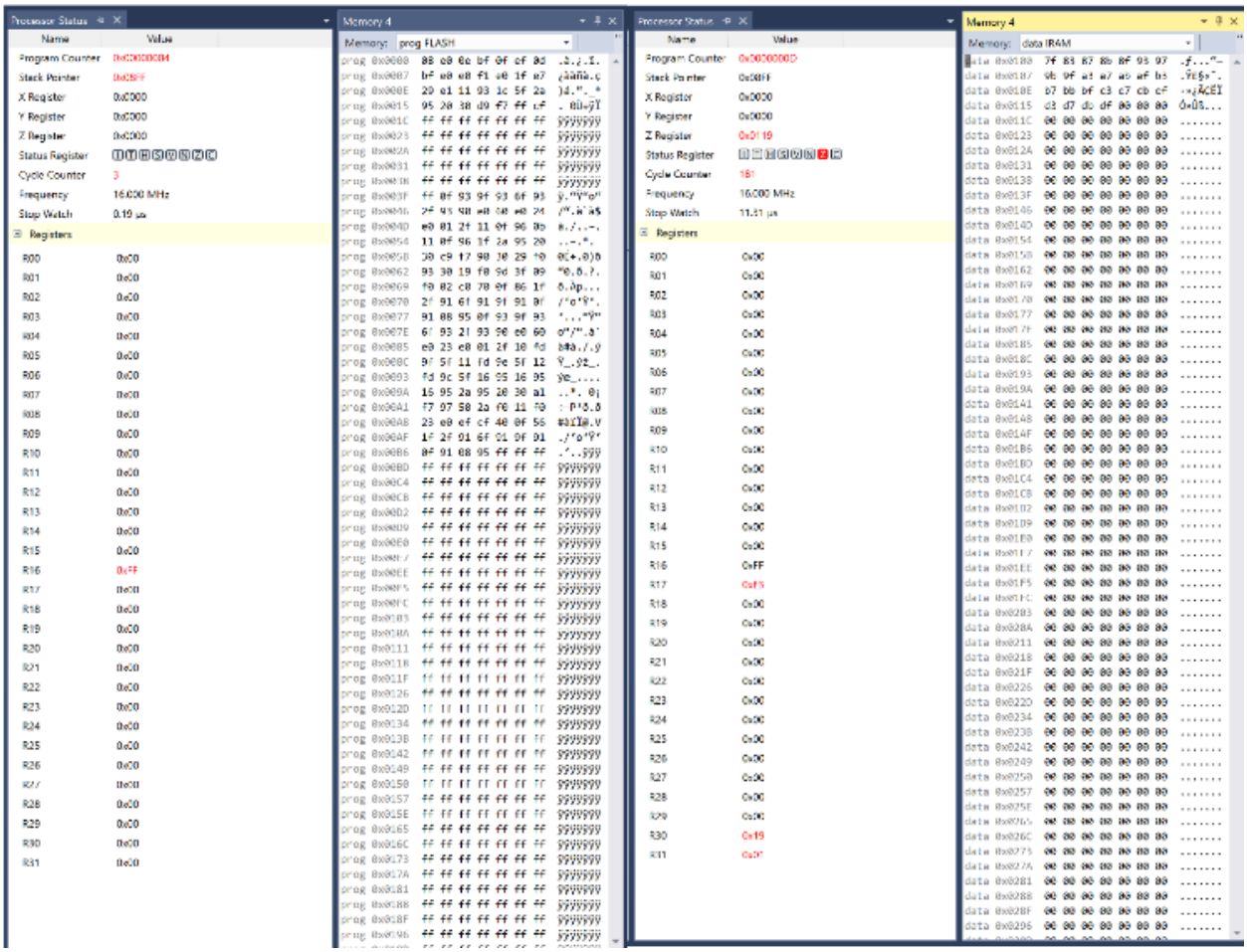
```
;Code segment to set overflow register
ovr_flw5:
;both labels will set overflow register is the sum is greater that 8 bits
ldi r17, 0x08
mov OVERFLOW, r17 ;copies r17 to OVERFLOW register and set bit 5
subi SUM_5H, -1
```

4.	Complete code		

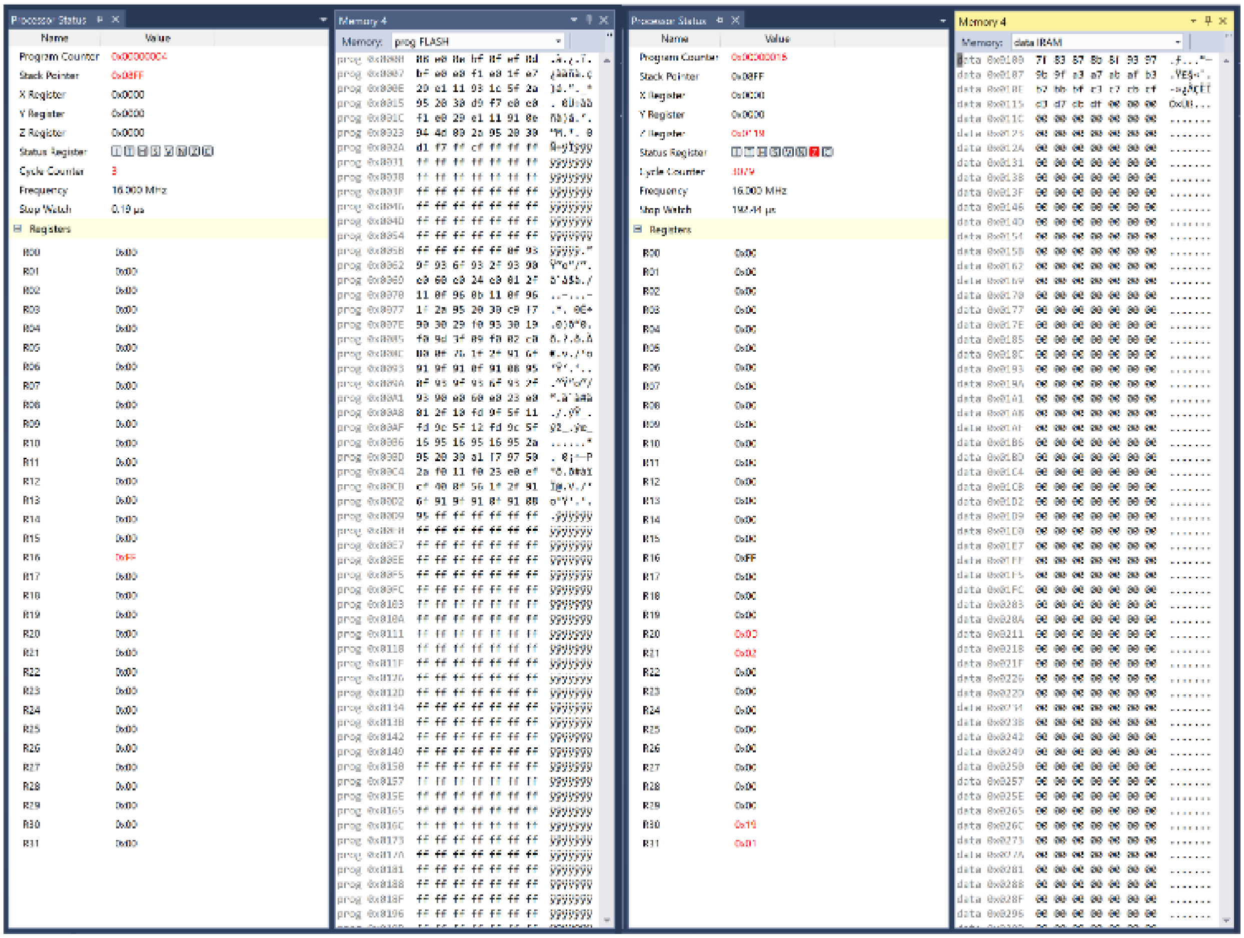
```
; cpe301Assignment1.asm
;
; Created: 2/23/2018 1:25:41 PM
; Author : YKengne
;

;Code segment that puts 300 numbers on to the stack


;Code segment that puts 300 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_5H=r23              ;high of sum of 5
.def SUM_5L=r24                   ;low of sum of 5
.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

;************************************

;Code segment that parses the numbers and check division by 5
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

division5:
;loop to divide number by 5
sum_5:
;calculates the sum for division by 5
add SUM_5L, number
brvs ovr_flw5

;*************************************************
;Code segment to set overflow register
ovr_flw5:
;both labels will set overflow register is the sum is greater that 8 bits
ldi r17, 0x08
mov OVERFLOW, r17 ;copies r17 to OVERFLOW register and set bit 5
subi SUM_5H, -1
```

5.	Screenshots of each task		

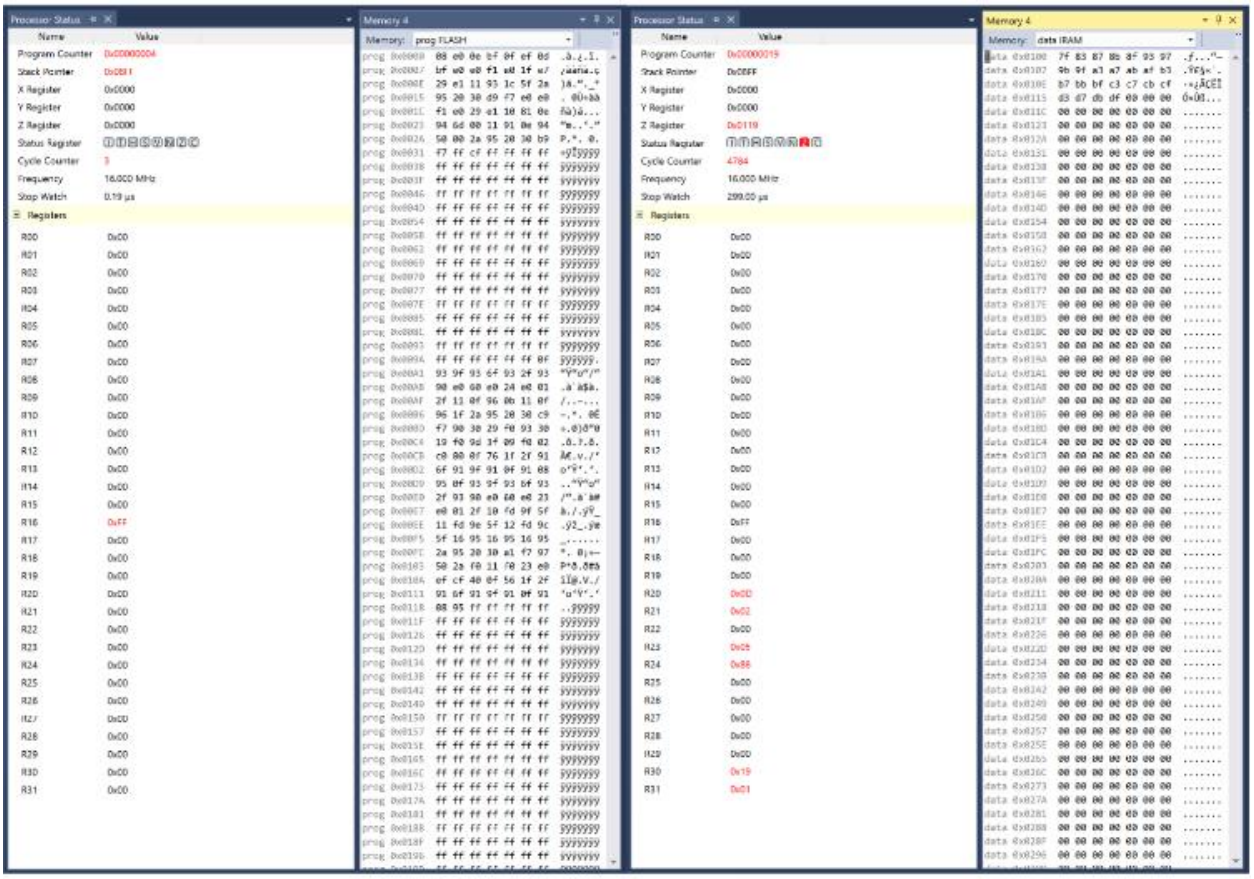
Task1



Task2


Processor Status
Memory 4
Name
Value
Program Counter
Stack Pointer
X Register
Y Register
Z Register
Status Register
Cycle Counter
Frequency
Stop Watch
Registers
Processor Status
Memory 4
Name
Value
Program Counter
Stack Pointer
X Register
Y Register
Z Register
Status Register
Cycle Counter
Frequency
Stop Watch
Registers

Task3



2.	GITHUB LINK OF THE DA		

Student Academic Misconduct Policy
http://studentconduct.unlv.edu/misconduct/policy.html

"*This assignment submission is my own, original work*".
Yannick Kengne Tatcha