Luis Ruiz

Assignment 1

CPE 301 – 1001

1. **PART 0**

The assignment took me an overall 3 hours for the fact that I had to solder on a few header and I never soldered before and plus the divide functions I made had to change due to wrong results.

1. **PART A**

My design consists of two loops and two functions to check if the integer stored are divisible by 7 and 3. The first loop stores the elements into the array with base address RAMEND/2, while the second loop goes through the array to check if the integer stored is divisible by 7 and or 3. If any element in the array is divisible by 7 the sum is stored into R21:R20, a similar concept goes for numbers divisible by 3; however, the sum is stored into R24:R23. In addition, if both sums are greater than 8-bits, basically if R21 and or R24 are greater than 0, then PORTB pin 4 and 3 should output HIGH. Otherwise if R21 <= 0 pin 4 should be LOW; similar concept goes for PIN 3, if R24 <= 0 then pin 3 is LOW. The design was simulated on Atmel Studio 7.0 and tested on Atmel 328p X-Mini.

**PART B: Code**

;

; DA1.asm

;

; Created: 2/1/2017 6:02:46 PM

; Author : Luis

;

.CSEG ;CODE SEGMENT

main:

.DEF COUNT = R16 ;Define COUNT as R16

.DEF NUM1 = R19 ;Define NUM as R19

.DEF ZERO = R0 ;Define ZERO as R0

.DEF SUM7H = R21

.DEF SUM7L = R20

.DEF SUM3H = R24

.DEF SUM3L = R23

CLR ZERO ;R0 = 0

LDI COUNT,0 ;R16 = 0

;Z contains RAMEND

LDI ZH, high(RAMEND) ;ZH = RAMEND[15:8]

LDI ZL, low(RAMEND) ;ZL = RAMEND[7:0]

;Z contains RAMEND/2

LSR ZH ;{0,[15:9]} C = [8]

ROR ZL ;{C,[7:1]} C = [0]

MOVW X,Z ;X = Z get a copy of Z

LDI r18, 25 ;R18 = 25

store\_loop: ;loop which will store the numbers in the array

cp COUNT, r18 ;compare r16 to 25

BRGE exit\_store ;check if R16 >= 25

mov R17, XL ;R17 = XL

ST X+,R17 ;mem[X] = XL ; X = X + 1

INC COUNT ;r16 = r16 + 1

RJMP store\_loop

exit\_store:

CLR COUNT ;r16 = 0

sum\_loop: ;parse through the array

;do{

LD NUM1,Z+ ;R19 = MEM[X]

CALL DIV\_7 ;go to function DIV\_7

CALL DIV\_3 ;go to function DIV\_3

INC COUNT ;R16 = R16 + 1

CP COUNT,R18

BRLT sum\_loop ;}while(COUNT < 25);

LDI R17, 0x18 ;R17 = 0X18

OUT DDRB,R17 ;PORTB PIN 4 AND 3 OUTPUTS

CP R21,ZERO ;compare r21 to zero

BRLO DONT\_OUT\_4 ;R21 < Zero Unsigned

BREQ DONT\_OUT\_4 ;R21 == Zero

LDI R17,0x10 ;R17 = 0X10

OUT PORTB,R17 ;PIN 4 OUTPUT HIGH

DONT\_OUT\_4:

CP R24,ZERO ;compare R24

BRLO DONT\_OUT\_3 ;

BREQ DONT\_OUT\_3 ;R24 == Zero

ORI R17, 0x08 ;R17 = R17 | 0x08

OUT PORTB,R17 ;PIN 3 OUTPUT HIGH

DONT\_OUT\_3:

end:

nop

rjmp end

;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

; FUNCTION

DIV\_7: ;Function to divide by 7

CLR R26 ;R26 = 0

SUBI R26,-7 ;R26 = 7

MOV R27, R26 ;R27 = 7

div\_loop\_7:

CP NUM1,R26 ;CHECK IF R26%R27

BREQ POSTIVE\_7 ;If R26%27 = 0 then it is divisible

BRLO NEGATIVE\_7 ;If R27 < R26 then it’s not divisible by 7

ADD R26,R27 ;R26 = R26 + 7

BRCS NEGATIVE\_7 ;If the Addition operation above creates a carry

;then the number being divided isn’t divisible by 7

RJMP div\_loop\_7

POSTIVE\_7:

ADD SUM7L,NUM1 ;ADD SUM7 = SUM7 + MEM[X]

ADC SUM7H,ZERO

NEGATIVE\_7:

RET

DIV\_3: ;Function to divide by 3

CLR R26 ;R26 = 0

SUBI R26,-3 ;R26 = 3

MOV R27,R26 ;R27 = 3

div\_loop\_3:

CP NUM1,R26 ;CHECK IF R26%R27

BREQ POSTIVE\_3 ;IF R26%27 = 0 then it is divisible

BRLO NEGATIVE\_3 ;IF R27 < R26 then its not divisble by 3

ADD R26,R27 ;R26 = R26 + 3

BRCS NEGATIVE\_3 ;If the Addition operation above creates a carry

;then the number being divided isnt divisble by 3

RJMP div\_loop\_3

POSTIVE\_3:

ADD SUM3L,NUM1 ;ADD SUM3 = SUM3 + MEM[X]

ADC SUM3H,ZERO

NEGATIVE\_3:

RET

**PART C: Flow Chart**

The running sum was being held by reg16 (0x10) which indicates a value of 150, it is pass the signed value of 127 thus an overflow

**The flow chart is on the next page**

Start

Count = 0

X = RAMEND/2

MEM[X++] = XL

Count ++

Count => 25

false

true

X = RAMEND/2

Count = 0

false

(MEM [X]%7 = 0) == 1

Function DIV 7

Function DIV 3

true

(MEM [X]%3 = 0) == 1

Count ++; X++

SUM7 += MEM[X]

SUM3 += MEM [X]

Count => 25

true

false

false

END

(SUM\_7H > 0) ==1

true

false

true

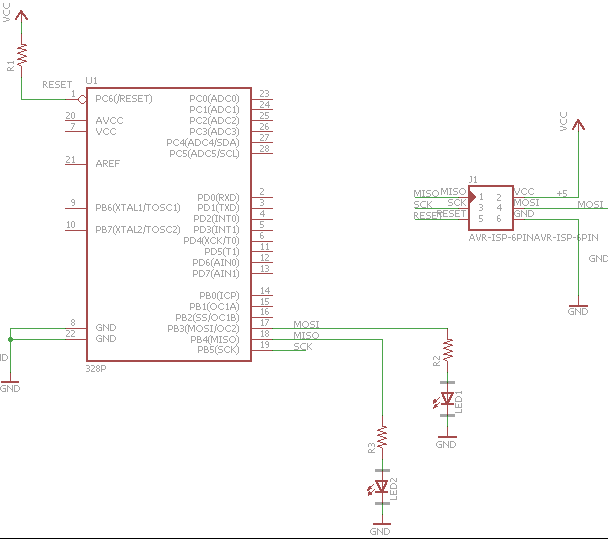
false

PORT B, PIN4 = High

true

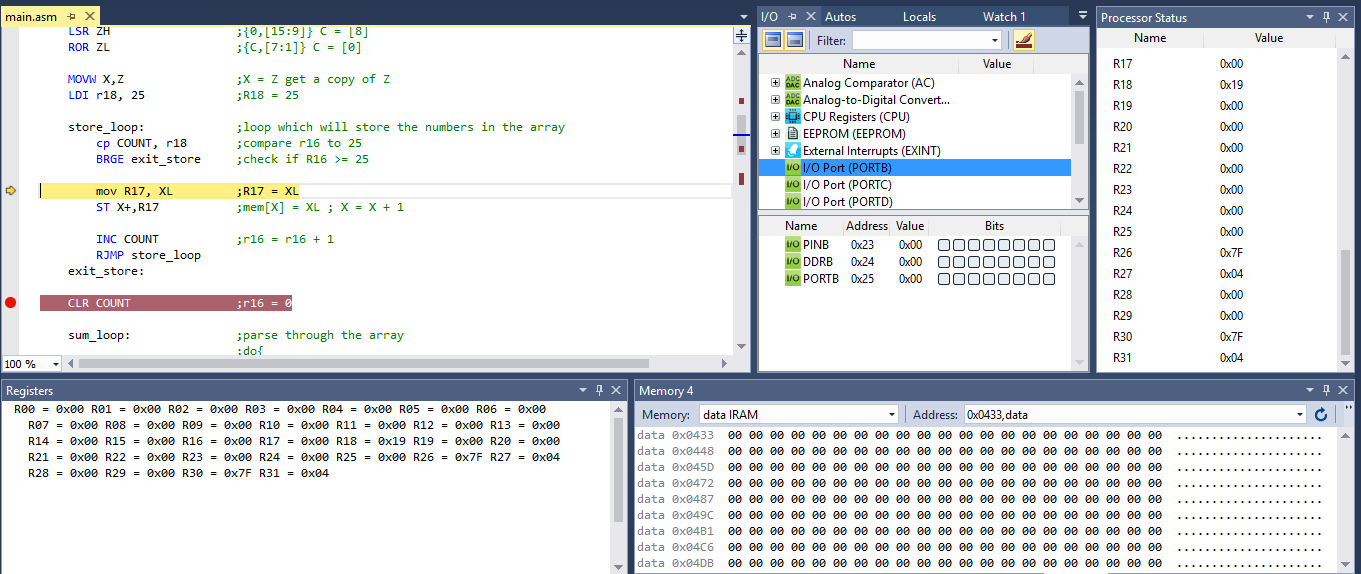
PORT B, PIN3 = High

(SUM\_3H > 0) ==1

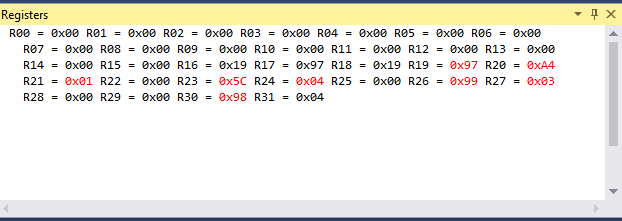
**PART D: Schematic**

**PART E**

The image below shows me go through my first loop to store the elements in the array.



The below image shows the integers stored after going through the first loop



The image below shows the content of the register after checking if the integer value in the array are divisible by 7 and divisible by 3. The sum held in R21:R20 indicate the integers that were divisible by 7 all added together and R24:R23 holds the sum of the integers divisible by 3.

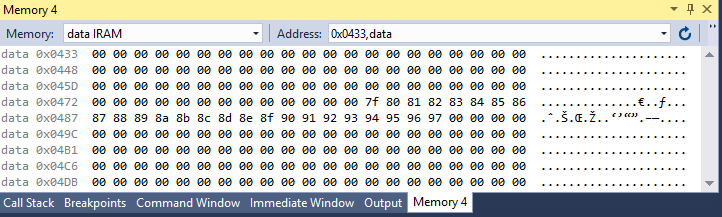
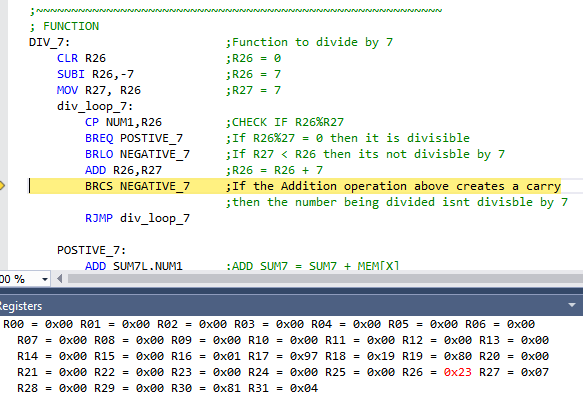
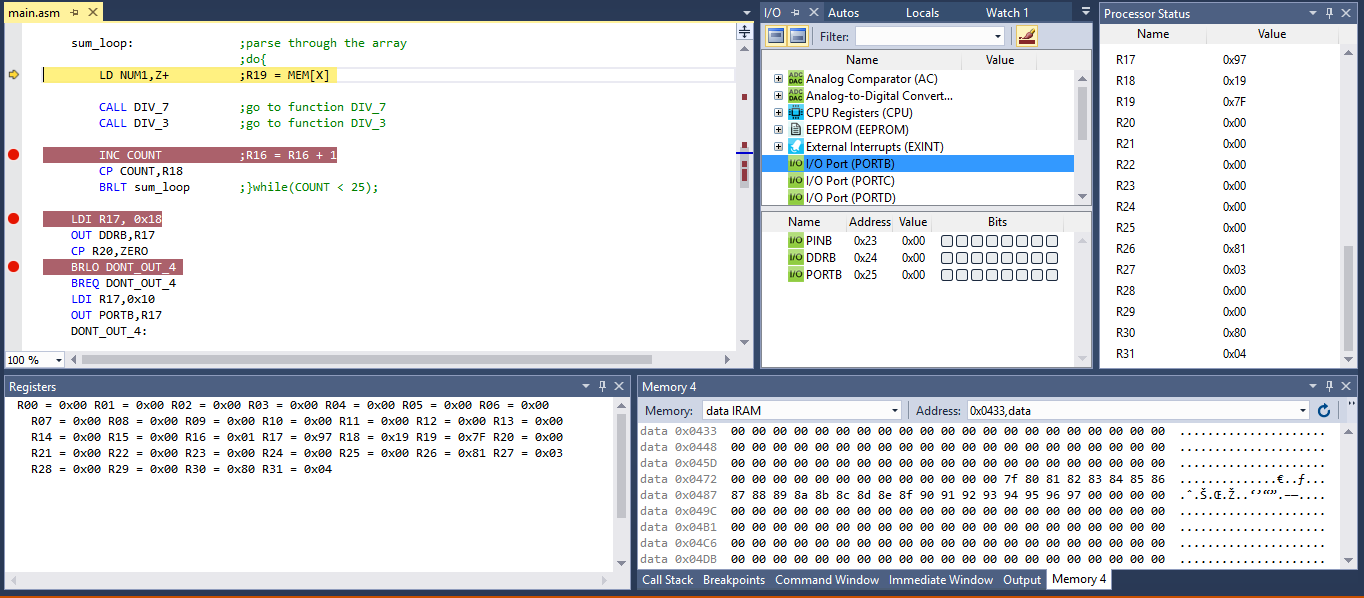
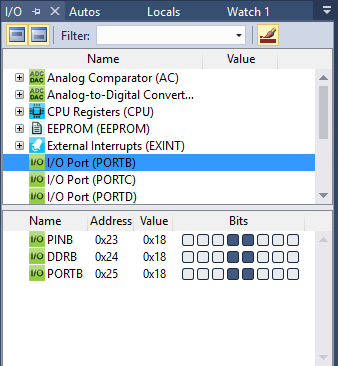


Figure 1: The Values in the Array; The base address is 0x047F



The images above show me going through the second loop where we check if the integers in the array are divisible by 7 and 3

The image below shows PORTB high on PIN 4 and PIN 3, these where set to indicate that both the sums were greater than 16-bits. The PORTB is set after exiting the second loop. I simply check in R21 and R24 > 0 then outputted high if so



**PART F**

URL Video of Design Assignment 1: <https://www.youtube.com/watch?v=IB_ru83uJW4>

**PART G**

