Aviles, Jean-Ralph
EEL3744
Section 1539
September 14th, 2015
Lab 1

Questions

1. What memory location should **.DSEG** start? Why?
   According to the manual, the data segment should start at memory location **0x2000** because that is the way the architecture was designed.

2. What registers can be used to read from program memory (flash)?
   Only the Z register according to the documentation...

   > The Z-register can also be used as an address pointer to read from and/or write to the flash program memory, signature rows, fuses, and lock bits.

3. What is the difference between program and data memory?
   The biggest difference is how big the pages are; the flash's page size is one word (two bytes) while the data memory's page size is one byte. Also the Z register is used to access Flash memory locations. They also differ in how they are organized and what sections are located within each, a more detailed list is available in the manual. Data memory also affords 1 cycle access times which may be important.

Problems Encountered

In writing the assembly code, the most frustrating thing was that the program memory is indexed by word while the reset of the memory map is indexed by byte. This made it extremely difficult to find where I had .ORG'd TABLE1.

Future Work/Applications

This gave me a good intro to writing assembly for the Atmel. I have written MIPS and x86 assembly before but its nice that I'm going to have an actual Atmel board to test my assembly instructions on.

Pseudocode/Flowcharts

Part B + C — Debug a Simulated ASM project

```
NUL := 0x0
LOWERBOUND := 0x21
UPPERBOUND := 0x40
Table1 := 0x4230
Table2 := 0x2B10
SRC := Address(Table1) - 1
DEST := Address(Table2)
DO
        SRC = SRC + 1
        IF DATA(SRC) <= LOWERBOUND OR DATA(SRC) > UPPERBOUND
        THEN
                DATA(DEST) = DATA(SRC)
                DEST = DEST + 1
        ENDIF
ENDDO
WHILE DATA(SRC) IS NOT NUL
```

Programs

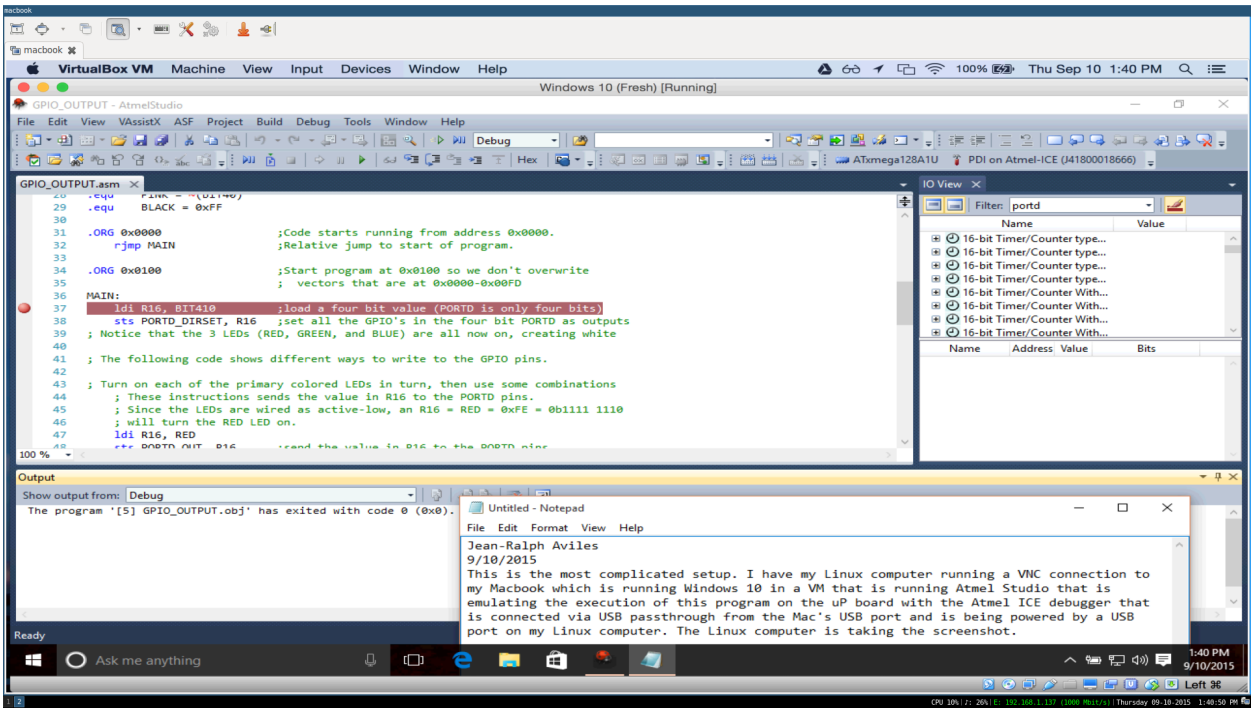
Part B and C – Code

```
/*
 * Table.asm
 * Lab 1 Part B
 *
 *  Created: 9/10/2015 2:24:48 PM
 *   Author: Jean-Ralph Aviles
 *   Section: 1539
 *   TA: Khaled
 *   This program filters values from a Table located at
 *   address 0x1000 in memory and places the filtered table
 *   at address 0x2B10.
 */

; Definitions for all the registers in the processor. ALWAYS
; REQUIRED.View the contents of this file in the Processor
; "Solution Explorer" window under "Dependencies"
.include "ATxmega128A1Udef.inc"
```

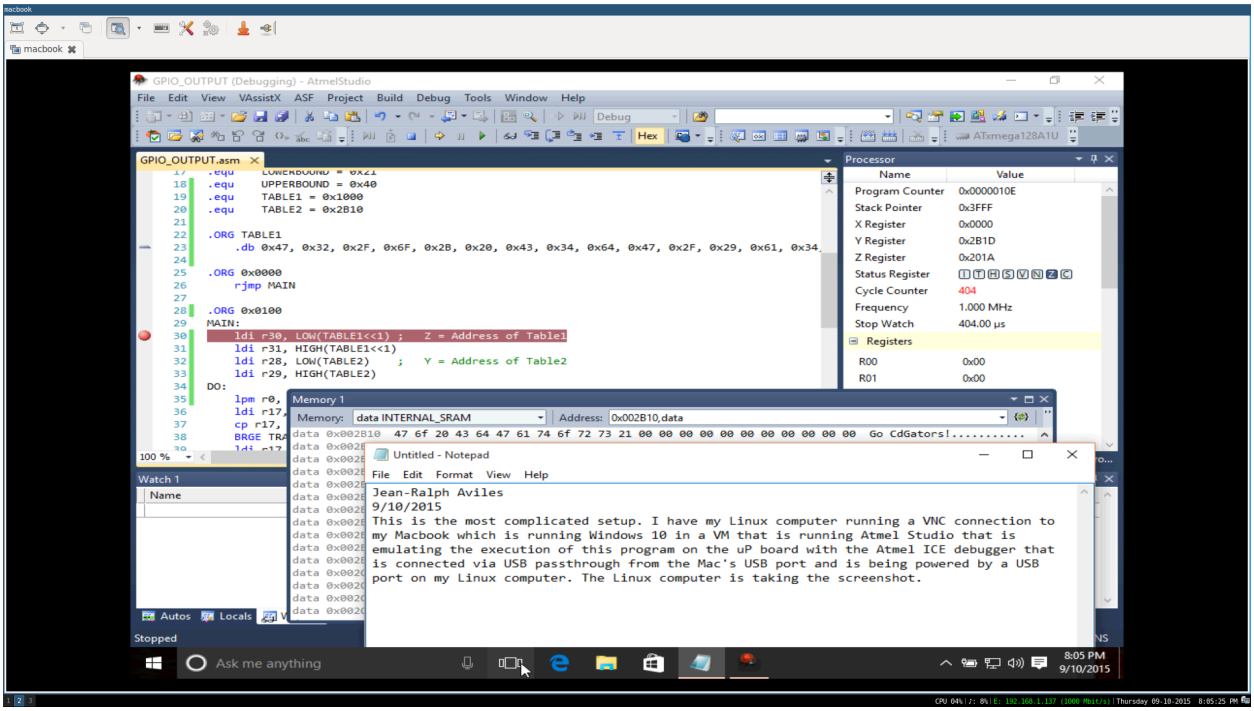
```
.equ  NUL = 0x00
.equ  LOWERBOUND = 0x21
.equ  UPPERBOUND = 0x40
.equ  TABLE1 = 0x1000
.equ  TABLE2 = 0x2B10

.ORG TABLE1
  .db 0x47, 0x32, 0x2F, 0x6F, 0x2B, 0x20, 0x43, 0x34
  .db 0x64, 0x47, 0x2F, 0x29, 0x61, 0x34, 0x26, 0x3C
  .db 0x74, 0x2A, 0x6F, 0x28, 0x2E, 0x72, 0x3F, 0x73
  .db 0x21, 0x0

.ORG 0x0000
  rjmp MAIN

.ORG 0x0100
MAIN:
  ldi r30, LOW(TABLE1<<1)   ; Z = Address of Table1
  ldi r31, HIGH(TABLE1<<1)
  ldi r28, LOW(TABLE2)   ; Y = Address of Table2
  ldi r29, HIGH(TABLE2)
DO:
  lpm r0, Z+   ; ro = *(SRC++)
  ldi r17, LOWERBOUND   ; r17 = LOWERBOUND
  cp r17, r0   ; Compare r17 and r0
  BRGE TRANSFER   ; IF r0 <= LOWERBOUND GOTO TRANSFER
  ldi r17, UPPERBOUND   ; r17 = UPPERBOUND
  cp r17, r0   ; Compare r17 and r0
  BRLT TRANSFER   ; IF r0 > r17 GOTO TRANSFER
RETURN:
  ldi r17, NUL   ; r17 = NUL
  cp r0, r17   ; Compare r0 with NUL
  BRNE DO   ; IF r0 != NUL GOTO DO
DONE:
  rjmp DONE   ; ELSE Loop forever
TRANSFER:
  st Y+, r0   ; *(DEST++) = r0
  rjmp RETURN   ; GOTO RETURN
```

Appendix

Part A - Screenshot



Part B - Screenshot



Part C - Screenshot