SC2107 Lab1 Assignment Sheet (to be submitted to NTULearn before next lab)

Name: Saravanakaleeswaran Arun Karthick Lab Group: SCEC Date: 28 August 2024

1. Section 7.4. Write one C statement to set bit 7 and 5 of P1SEL0 register, keeping the rest of the bits in the register unchanged.

**Answer Q1 Section 7.4:** P1SEL0 |= (1<<5) | (1<<7);

1. Section 7.4. Write C statement(s) to extract bit 6 and 5 of variable ‘x’ and right align these two bits. Masked off all other bits in variable ‘x’. e.g. if ‘x’ has a value 1101 0111b initially, it should have a value of 0000 0010b after executing the C statement.

**Answer Q2 Section 7.4:** x = (x& ((1 << 6) | (1 << 5))) >> 5;

(1 << 6) and (1 << 5) creates a bitmask of 0100 0000b and 0010 0000b respectively. Combining both like this ((1 << 6) | (1 << 5)) will isolate but 5 and 6 to create 0110 0000b.

“x&” clears the remaining bits and “>>5” moves bit 5 and 6 to positions 1 and 0 by right shifting them.

1. Section 7.4. Why do we need to declare the P1IN register, the register that contain the status of the processor Port1 GPIO input pin logic with a ‘volatile’ keyword qualifier?

**Answer Q3 Section 7.4:** This is to inform the compiler that the value of this register can change outside of the program. If not used, the compiler may assign the variable to a temporary register or even remove the variable during the code optimization process

1. Section 7.5. Why do we use SDIV instead of UDIV when calculating the Distance D? Or does it really matter whether SDIV or UDIV is used for this case?

**Answer Q4 Section 7.5:** We used signed division instead of unsigned division because to calculate D, the equation involves subtracting n from 1058 which could result in a negative result. If you used UDIV, the negative result would be treated as a positive number

1. Section 7.5. What is saved into the LR register when the calling routine calls “BL Convert”? What command is used to return from the sub-routine to the calling routine?

**Answer Q5 Section 7.5:** When BL convert is called, the program branches to the convert subroutine. The address for the program to return back is saved into the LR. To return from the subroutine, “BX LR” command can be used.

1. Section 7.6. If a function has 4 input parameters, which registers does the calling routine used to pass these parameters to the function according to AAPCS?

**Answer Q6 Section 7.6:** If a function has 4 input parameters, it will be stored in registers R0-R3. Additional parameters will be stored in the stack.

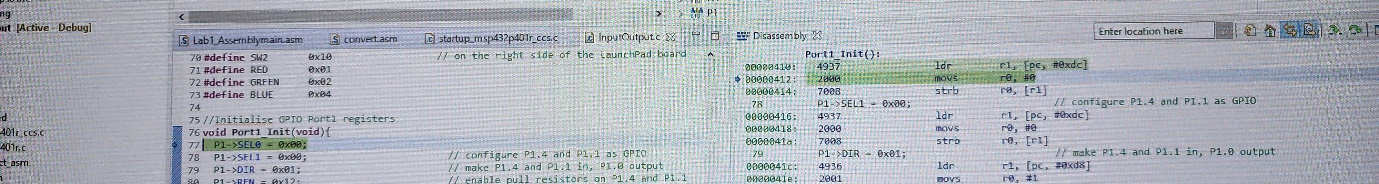
1. Section 7.6. What data content is loaded into R1 by the instruction “ldr r1, [pc, #0x2e4]”? Just the expression will do, need not give the exact value since the offset in your code may be different.

**Answer Q7 Section 7.6:** The instruction LDR R1, [PC, #0xDC] is located at the address 0x00000410. The offset in my code is 0xDC as shown in the pic below.

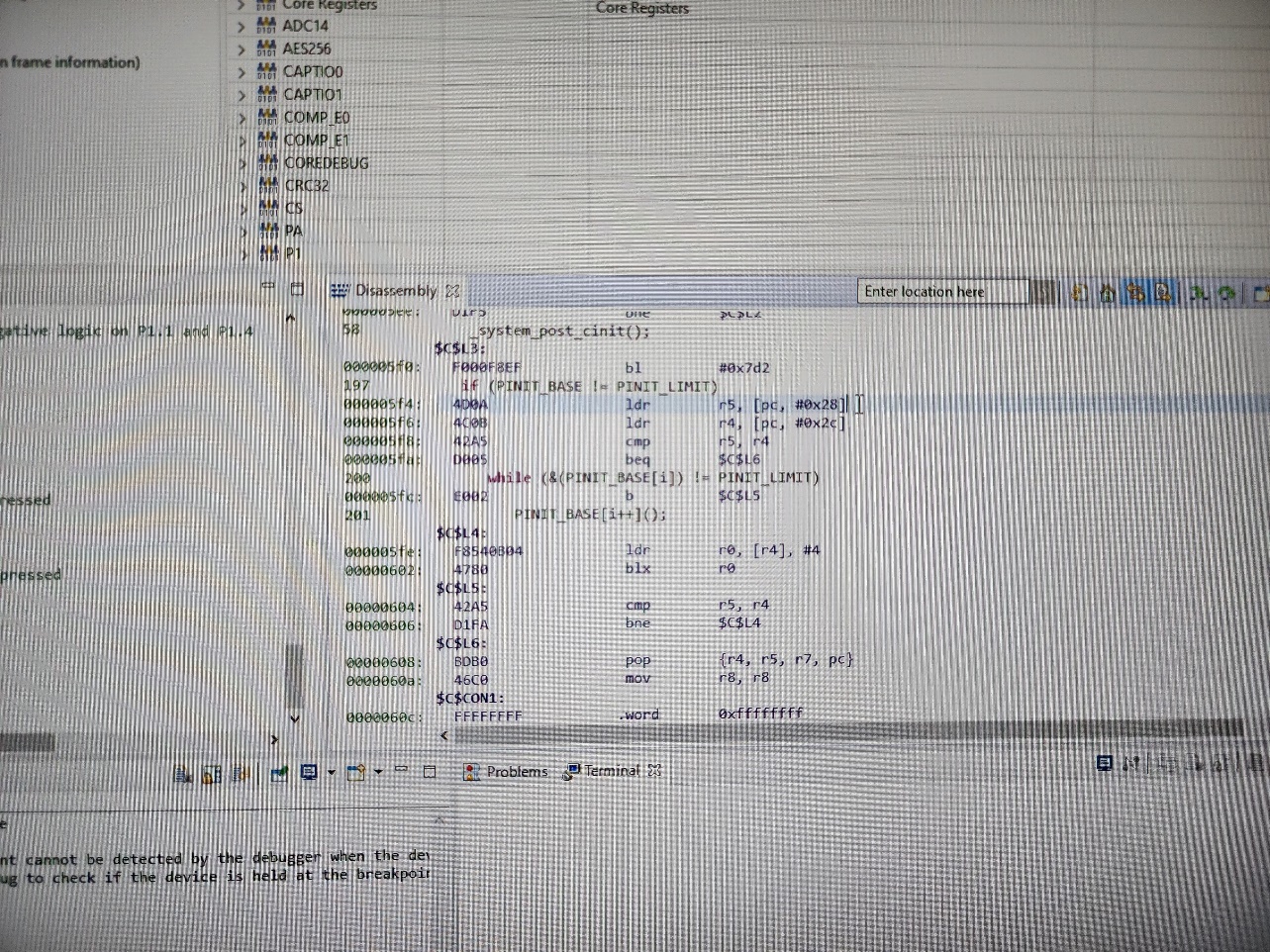
The PC is ahead of the current instruction by 8 bytes. So, 0x00000410 + 8 = 0x00000418.

Adding the offset #0xDC, 0x00000418 + 0xDC = 0x000005F4. The data at address 0x000005F4 is loaded into R1.



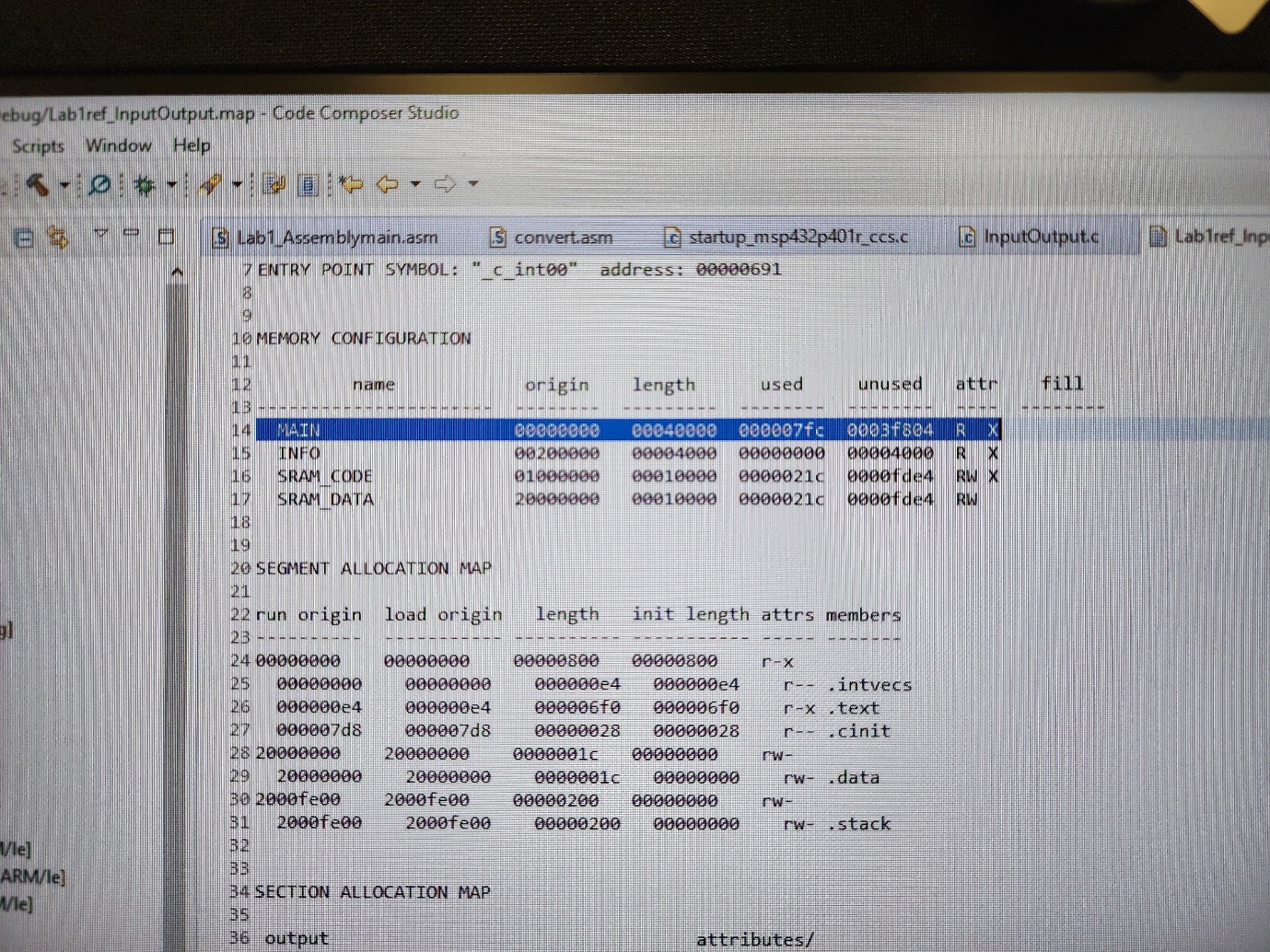






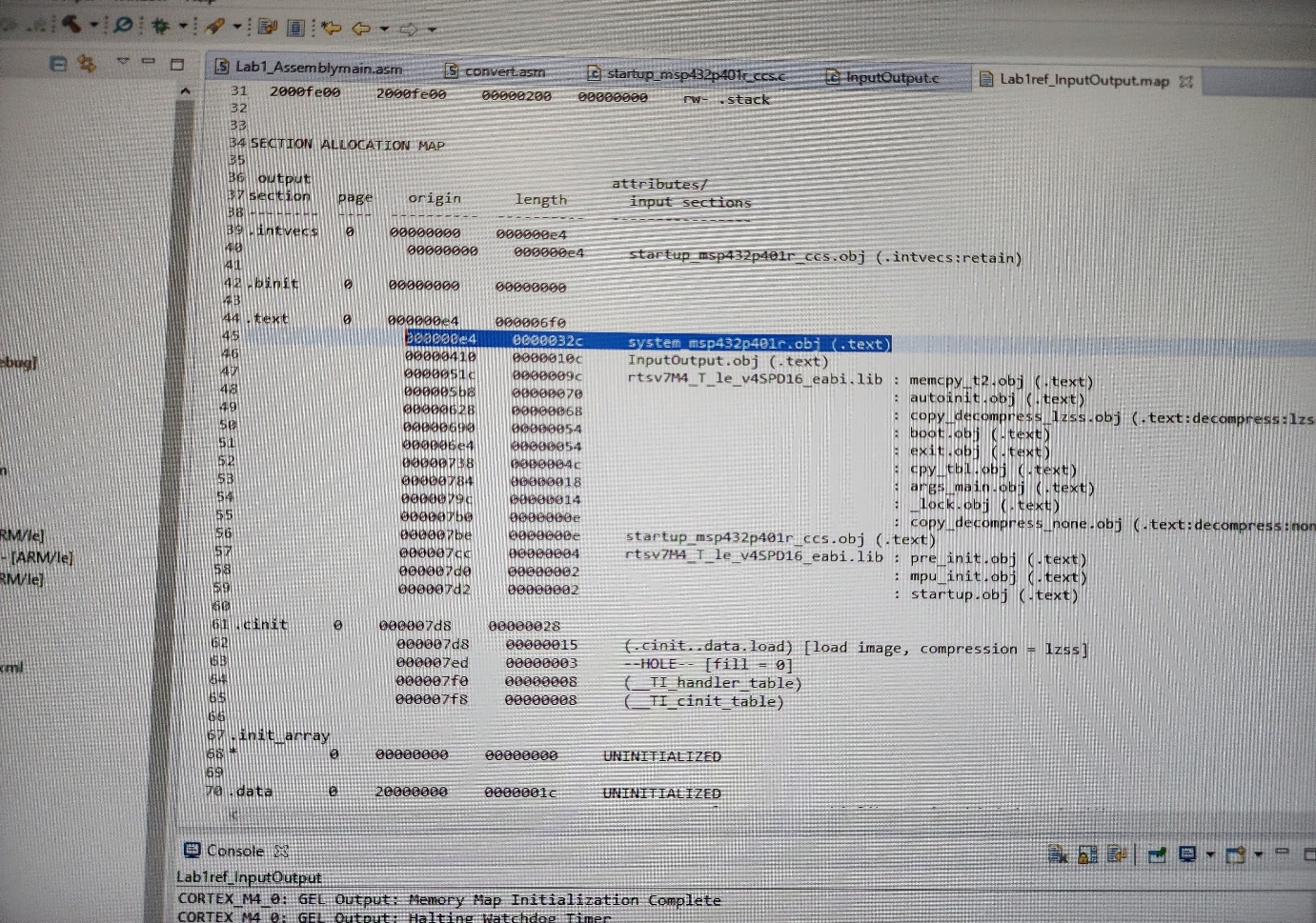


1. Section 7.7. The Memory Section “MAIN” correspond to the On-Chip Flash Memory in MSP432. How much on-chip flash memory is available for future code development? Cut and paste the screen shot of the relevant content in the map file and highlight where you extract your answer from. Hint: Check the map file.  
   **Answer Q8 Section 7.7:** Main unused is 3F804 memory is available for future coed development





1. Section 7.7. Which software section are code allocated to by default? Which file consumes the largest code size in this project? Hint: check the map file.

**Answer Q9 Section 7.7:** .text file. System\_msp432p401r.obj consumes the largest code size in this project of length 0x0000032C. 



1. Section 7.7. From the map file, what is the starting address of Port2\_Init()? Compare with the address you see in the Disassembly Window, are they the same? If not, why?

**Answer Q10 Section 7.7**: 43b in map file while the disassembly window shows 43a(shown in the picture below). They are not the same. This could be due to ARM 32bit instructions or thumb 16bit instructions or alignment issues/differences.





