CE2107 Lab1 Assignment Sheet (to be submitted to NTULearn before next lab)

Name: Ang Kai Jun Lab Group: SEP4 Date: 25/8/20

1. Section 7.5. Why do we use SDIV instead of UDIV when calculating the Distance D?

SDIV is the signed division while UDIV is the unsigned division.

1. Section 7.5. Why must the calling function save LR before calling another function?

To know where to resume at the main function.

1. Section 7.5. If a function has 4 input parameters, how does the calling routine pass these parameters to the function according to AAPCS?

The input parameters are passed in through R0-R3 and the output parameter is returned in R0.

1. Section 7.6. What does “ldr r1, [pc, #0x2e4]” do?

Load contents of memory at location 0x3ca to register 1.

1. Section 7.6. In the code snippets shown, why is the same instruction “ldr r1, [pc, #0x2e4]” used in the initialization of Port1’s SEL0 and SEL1 registers? Does that mean these instructions are all writing to the same location since the same offset (#0x2e4) is used?

PC value is different at each instruction. Instructions are not writing to the same location.

1. Section 7.6. Which register is used to store the return value of Port1\_Input()? Which register is used to store the argument of the Port1\_Output(data) function? Note the C compiler confirms to the AAPCS standard.

R0 is used to store the return value. R0 is used to store the argument of Port1\_Output(data) function.

1. Section 7.7. How large is the code size for this project? Hint: which software section are code allocated to?

0xA1C

1. Section 7.7. Which file consumes the largest code size in this project? Hint: check the map file.

Text file (InputOutput.obj and system\_msp432p401r.obj)

1. Section 7.7. How much SRAM is left for program expansion? Note that SRAM\_CODE and SRAM\_DATA is sharing the same piece of SRAM in the physical memory

0x438

1. Section 7.7. From the map file, what is the starting address of Port1\_Init()? Compare with the address you see in the Disassembly Window, are they the same? If not, why?

From map file, address of Port1\_Init() is 0x000000e5. In the Disassemble window, address is 0x is 0x000000e4. Addresses are different.

The map file contains absolute addresses but in assembly, only symbolic address references are stored. The map file has the more accurate address.