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

Name: Mervyn Chiong Jia Rong Lab Group: SEP4 Date: 25/08/20

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

**SDIV represents signed division while UDIV represents unsigned division, It is possible that unsigned division will give unexpected and uncontrollable results hence using SDIV is better**

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

**To know where to resume the function after running another function. LR is saved before overwritten**

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?

**Inputs are stored in registers R0 to R3, output is then stored in R0 at the end**

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

**It combines the current pc value with #0x2e4 as an offset then that address value, is loaded into R1**

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?

**No, as the program counter will always change. The code is dependant on the pc value hence it will not always give the same result despite having #0x2e4 as a constant**

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 for Port1\_Input(),return value**

**R0 to store the argument of the Port1\_Output(data)**

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, it is 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

**0xFDE4 is left**

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?

**Map file shows address of Port1\_init() is 0x000000e5 but disassemble window address with 0x is 0x000000e4.Map file(more accurate address) contains absolute addresses but in assembly only symbolic address references are stored.**