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

Name: Unnikrishnan Malavika Lab Group: SEP4 Date: 25/08/2020

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

**ANS: there is a division in the conversion equation so a SDIV instruction is required. SDIV is signed division which divides a signed integer register value by another signed integer register value and writes the result to the destination register. This is done in such a way that condition flags are not affected.**

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

**ANS: The link register LR is used to hold the address to which a function should return when it finishes executing. If one function calls another, it must save the existing value of LR somewhere, otherwise it will be overwritten and lost forever.**

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?

**ANS: They are sent to the register R0-R3**

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

**ANS: it loads the value at the address location (address in pc) +0x2e4 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?

**ANS: No. Because even though the offset value is the same, the PC value is different so it’ll point to different addresses**

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.

**ANS. R0 is used to store the argument of Port1\_Output(data) function as the function has only one argument.**

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

**ANS. 2320 bytes**

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

**ANS. system\_msp432p401r.obj and InputOutput.obj both have size 812**

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

**ANS. 64996 bytes**

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?

**ANS. No. They’re not the same. The address of Port1\_Init() in map file is 000000e5. Whereas in the disassembly window, it is 000000e4. This is because Bit[0] of any address you write to the PC with a BX,BLX,LDM,LDR, POP instruction must be 1 for correct execution because this indicates the required instruction set – which for the Cortex M3-M4F processor is Thumb instructions.**