Lab 2 Prep

1. Make a copy of the ADCSWTrigger\_4C123 project and call it Lab 2
2. Answer the following questions
   1. **What is the purpose of all the DCW statements?**

DCW stands for “declare constant word” but really puts a 16 bit halfword into memory. LDR and STR load from/store to memory by offsetting from a register or the PC and the offset is not very large (relatively). To fix this, the compiler stores into memory the values it needs to access and stores these nearby the program counter. Since IO ports are memory mapped and PF1 is mapped to 0x40025008, the DCW command is used to store 0x5000 at 0x068C and 0x4002 at 0x68E.

* 1. **The main program toggles PF1. Estimate how fast PF1 will toggle.**

At 80 MHz, each assembly instruction takes about 25 nanoseconds to execute. Toggling PF1 takes 5 assembly instructions, so the execution will take about 125 ns.

* 1. **What is in R0 after the first LDR is executed? And after the second LDR?**

The Cortex M4 processor uses little endian format so when the first LDR executes, the value 0x40025000 is loaded into R0. Next, the value at address 0x40025008 (i.e. the value of PF1) is then loaded into R0 (meaning R0 will be either 0x00 or 0x02 after the second LDR).

* 1. **How would you have written the compiler to remove an instruction?**

You could just LDR the value into a separate register other than R0 since that value (0x40025000) will be used twice.

LDR r1,[pc,#24] ; r1 = 0x40025000

LDR r0,[r1,#0x08] ; r0 = M[0x40025008]

EOR r0,r0,#0x02

~~LDR r1,[pc,#16] ; @0x0000068C~~

STR r0,[r1,#0x08]

* 1. **100-Hz ADC sampling occurs in the Timer0 ISR. The ISR toggles PF2 three times. Do these two read-modify write sequences to Port F create a critical section? If yes, describe how to remove the critical section. If no, justify your answer.**

No critical region because both the interrupt toggle and main toggle use different pins and bit-specific addressing, so the code remains friendly.