Juan Tarrat

LAB 6 REPORT

1. Theory:
    a. Interrupts are a way to control the flow of the program. As the program is executing, if an interrupt is detected (has to be detected in some of the I/O peripherals no MSP430 the program jumps to the interrupt, executes whatever code is there and returns where it left. An example of using interrupts is a switch. We have to set up the port we are going to use (1 or 2) in MSP430 and then set the edge select. This is an important part since with this what you are setting up is what signal (hi to low or low to high) is going to trigger the interrupt.
    An interrupt vector is where the address of the ISR is defined. Interrupt vectors are placed in a table in memory and have a certain priority. In this lab we have used P1 and P2 interrupt vectors since those are the ports we have used.
    b. Clock Module: Processors have clocks that in some way control the speed of execution. With the clock module we can control different clocks to adjust the program. There are 5 types of clocks in the MSP430: XT1CLK, VLOCLK, REFOCLK, DCOCLK, XT2CLK. These clocks vary in their function and their frequency. Then the module also provides a set of clock signals which are meant for different purposes and frequencies. There are four registers that control the UCS module: SCG0, SCG1, OSCOFF and CPUOFF. The clock module has also a digital controlled oscillator and a frequency locked loop.
2. Questions:
    a. In program 1, if we press switch 2 while led 1 is blinking what will happen is that when we release the led will finish its blink but led 2 will not change its state. This is because the way the program is written, it is constantly checking for changes in the registers that get changed if the interrupt is entered. If you enter an interrupt while another segment of the code is executing, that segment will finish, in this case the led will continue blinking and then the other led will change its state.
    b. In program 2 the blinking rate for the different frequencies are as follow: Formula: delay(1000000)/2*frequency
        i. 1MHz = 0.5s between blinks
        ii. 2MHz = 0.25s between blinks
        iii. 4MHz = 0.125s between blinks
        iv. 8MHz = 0.0625s between binks

*No output needs to be shown so I will just upload the source files