VE 373 Post Lab Report

Introduction to PIC32 Starter Kit

Song Yang 5093709066
Yang LiRen 5093709047
Wang Changming 5093709048

1. OBJECTIVE

- Get familiarized with the PIC32 Starter Kit Ethernet board
- Develop an embedded application using I/O ports
- Learn how to use the MPLAB IDE software for microprocessor based system development

2. INTRODUCTION

I/O Resources

The PIC32 Starter Kit Ethernet board should be used as the development board for all embedded applications in this course. The board has two types of USB connectors, one for programming and debugging, and the other one for data transferring in USB applications. The USB port for programming also supplies power to the board. There is also an Ethernet connector for Ethernet applications. All the I/O ports of the MCU are wired to a HIROSE connector which may be connected to a PIC32 Starter Kit Expansion board to breakout the pins.

3. DESIGN PROCEDURE

In this experiment, we are required to write a C program to produce time delay and use the Digital Logic Analyzer to tune the accuracy of your output. In the program, we first turn on port 4 of PORTD for 2 seconds and turn it off for 1 second, then repeat the same instructions. After discussion and operation on the board, we decide to use for loop to achieve the functionality. According to the suggestion of TAs, we are allowed to choose port 1 of PORTD as the output because it is connected to an LED on the board.

Here we choose to use C function to generate delay instead of using timers. To be precise, we use two "for" loops with empty loop body to generate 1 second delay and 2 second delay. Note that each loop will be disassembly into several assembly instructions, which will take one clock cycle to execute, therefore the number of loop is supposed to be determined carefully. With the help of MPLAB, we are able to find out how many assembly instructions there is for each loop and calculate length of the loop.

Here is the disassembly code of a "for" loop, and there are 11 instructions for each single loop

cycle.(Note that sw at line 9D000064 following beq at 9D000060 is always taken in case of control hazard)

```
7:                      for (i=0;i<=7270000;i++){asm("nop");}
9D000038  AFC00000      sw          zero,0(s8)
9D00003C  8FC30000      lw          v1,0(s8)
9D000040  3C02006E      lui         v0,0x6e
9D000044  3442EE70      ori         v0,v0,0xee70
9D000048  0043102A      slt         v0,v0,v1
9D00004C  14400006      bne         v0,zero,0x9d000068
9D000050  00000000      nop
9D000054  00000000      nop
9D000058  8FC20000      lw          v0,0(s8)
9D00005C  24420001      addiu       v0,v0,1
9D000060  1000FFF6      beq         zero,zero,0x9d00003c
9D000064  AFC20000      sw          v0,0(s8)
```

The PBCLK supplied by PIC32 board is 8MHz and for 1 second delay we have

$$\text{Loop length} = \frac{8MHz}{11} = 7.27\times10^5 = 727000$$

Similarly, to achieve a 2 second delay, we need a "for" loop of length 1454000

The code is given as follows

```
#include <p32xxxx.h>

main()
  {
    TRISD = 0xff00;
    while(1)
      {
      int i=0;
      int j=0;
      for (i=0;i<727000;i++){asm("nop");}
      PORTD = 1;
      for (j=0;j<1454000;j++){asm("nop");}
      PORTD = 0;
      }
  }
```

4. SIMULATION AND TESTING RESULT

First we simulate the code using StopWatch, the breakpoints are given as follows:

```
#include <p32xxxx.h>
main(){
TRISA = 0xff00;
while(1) {
int i=0;
int j=0;
for (i=0;i<727273;i++){asm("nop");}
PORTD = 1;
for (j=0;j<1454545;j++){asm("nop");}
PORTD = 0;
}
}
```

The first breakpoint B1 is set to eliminate some additional clock cycles at the very beginning of the program. And the time between B1 and B2 is expected to be 1 second and that between B2 and B3 is expected to be 2 seconds.

And here is the result of simulation:

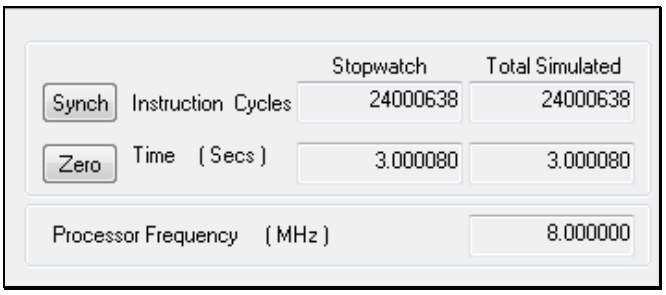
At B1:

		Stopwatch	Total Simulated
Synch	Instruction Cycles	623	623
Zero	Time (uSecs)	77.875000	77.875000
	Processor Frequency (MHz)		8.000000

At B2:

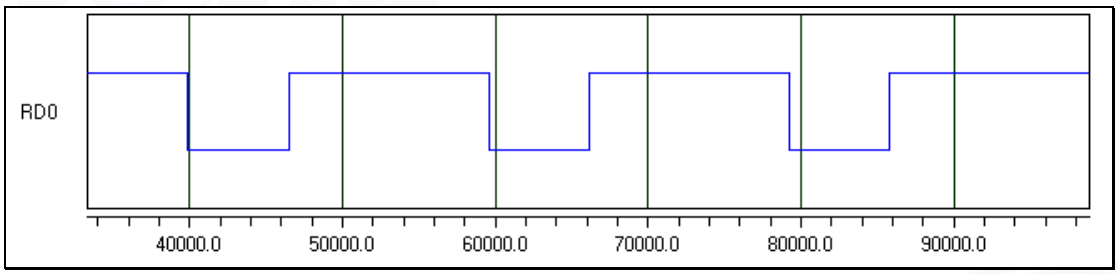
		Stopwatch	Total Simulated
Synch	Instruction Cycles	8000633	8000633
Zero	Time (Secs)	1.000079	1.000079
	Processor Frequency (MHz)		8.000000

At B3:



The simulation is also carried out using Logic Analyzer. Since the range of Logic Analyzer is not big enough to hold 1 second, we modify the code a little bit and let PORTD<0> be on for 2ms and off for 1ms.

The wave generated is given as follows:



From the wave diagram we can observe that the output signal is on for a period of time two times of the time being off.

5. DISCUSSION

During the process of the experiment, we encountered some troubles and spent much time to solve them. At first, we meant to apply timers to produce the delay. According to the instructions in the slides, we calculated the maximum number of the PR register value and set the prescale value, however, when we went to the lab, we are told that timer is not encouraged and there are some parameters which we did not know how to set. So we adapt for loop instead. Perhaps the operating systems are not suitable for the software; the program cannot run on our computers so we perform the program successfully on TA's computers.

There is another suspicious point that we find the MPLAB software may not be accurate enough. We should repeat the loop for 8 million times to produce 1 second delay. However, in reality we

find the actual number of loops is much smaller than that. Perhaps one of the reasons is that one loop cycle is disassembly into several assembly instructions and we must take this into consideration when we determine the length of loop. Sometimes some additional "nop" instructions are added for hazards and we really do not know whether this instruction is executed or not or when it is executed. Therefore perhaps the best way to do this is to use breakpoints to find out exactly how many instructions are executed within one loop cycle.

6. CONCLUSION

In this experiment, we get a fundamental knowledge of the PIC 32 board. Besides, we wrote a simple C program to produce the delay which we desired. And we learned how to use the StopWatch and the Logic Analyzer to simulate. Though we faced some problems which are beyond our expectation, we solved them and gained experience about the actual performance of the board and the software. In spite of these, we find that there may be a lot of incompatibility problem between the MPLAB software and windows 7 OS. After this lab, we decide to install windows XP OS and we hope in that way the incompatibility problem can be solved. Anyway, we are expected to perform better and get a deeper knowledge in the following experiments.