

Ga-wun Kim

ENGR 270

Scott Koss

Lab Partners: Rebecca, Kantas

Computer and Organization & Microprocessors Lab #2

Introduction

From the lab1, we spent much of time to get used to utilize the chip and the program. Based on this experience, we could program the codes, designed the circuit, and confirm the codes by this circuit output.

Experiment 1

In this experiment, using PICmicro, MPLab IDE and Lab kit, implement and test a circuit that accepts two, 2-bits binary input and output the sum as a 3-bit binary value. For this lab, in programing, we aimed that processor accept two separate 2-bit digital inputs, add them together, and produce a 3-bit digital output. We used the three diodes and they were regarded as the output. The below picture is the codes that we insert into the chip.

```
;FILE: main.asm
;DESC: Lab 2 2-bit Adder
;DATE: 2-3-17
;AUTH: Rebecca Chis, Kantas Zalpys, Ga-wun Kim
;DEVICE: PICmicro (PIC18F1220)
;-----
#include <pic18f1220.inc>

        list      p=18f1220
        radix     hex
        config    WDT=OFF, LVP=OFF

INPUTA   equ 0x080
INPUT    equ 0x081
LASTIN   equ 0x082
org 0

CLRF     PORTB;          clearing PORTB

;initialize pins to digital
MOVLW    0x7f
MOVWF    ADCON1

;initialize I/O pins
MOVLW    0x0f
MOVWF    TRISB

check:    MOVFF    PORTB, INPUT;      moving PORTB into INPUT
          MOVF     INPUT, W; moving INPUT into WREG
          ANDLW    0x0f;              ANDing WREG with 00001111 to get inputs
          XORWF    LASTIN, W;         XORing WREG with LASTIN to see if inputs have
changed
          BNZ      start;             calls start label if there's a change
          GOTO     check;             loops back

start:    ;getting first binary value
          MOVF     INPUT, W; moving INPUT into WREG
          ANDLW    0x03;              ANDing WREG with 00000011 to get A1 and A0
          MOVWF    INPUTA;           storing first binary number in INPUTA

          ;getting second binary value
          MOVF     INPUT, W; moving INPUT into WREG
          RRCF     WREG;              rotating WREG 2 bits to the right to get proper
placing
          RRCF     WREG
          ANDLW    0x03;              ANDing WREG with 00000011 to get B1 and B0

          ADDWF    INPUTA, W;         adding first binary number(INPUTA) to WREG and
storing result WREG
          RRCF     WREG;              rotating bits right three times to get proper
placing
          RRCF     WREG
          RRCF     WREG

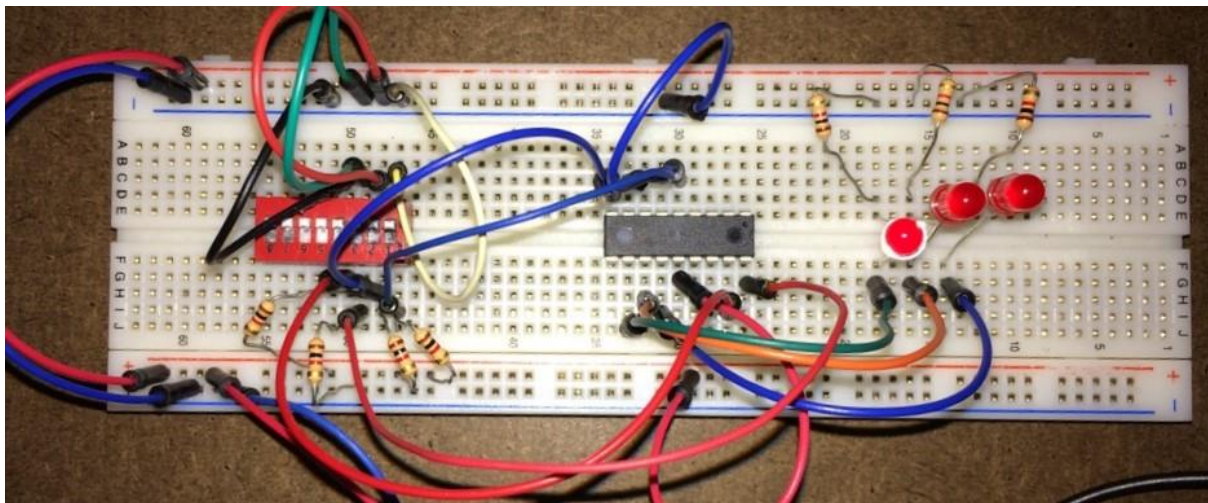
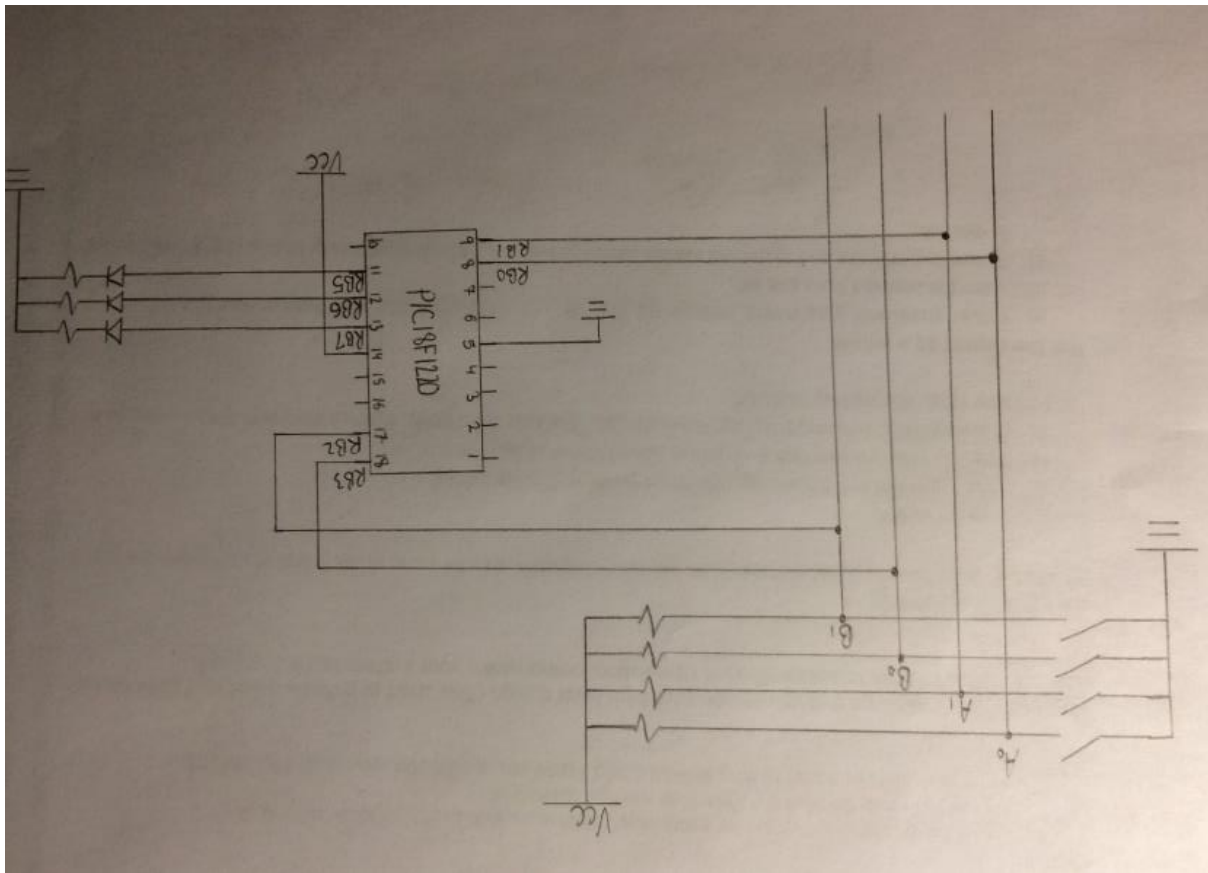
          MOVWF    PORTB;            moving WREG (R2, R1, R0) into PORTB to display
result

          MOVF     INPUT, W; moving INPUT into WREG
          ANDLW    0x0f;              ANDing WREG with 00001111 to get last input
values
          MOVWF    LASTIN;           storing WREG(INPUT) in LASTIN for next test

          GOTO     check

end
```

Below, there are the two picture; schematics and actual circuit.



Input operands(A1, A0) and (B1, B0) are simulated using DIP switches and should be connected to PICmicro input pins 9, 8, 18. and 17 respectively; and the resulting output(R2,R1,R0) is produced on PICmicro pins 13, 12, 11, and displayed by using LEDs. For example, if we insert input ((A1, A0) \rightarrow (1,1) and (B1, B0) \rightarrow (1,1)), the output is (110) and the diodes are (on,on,off)

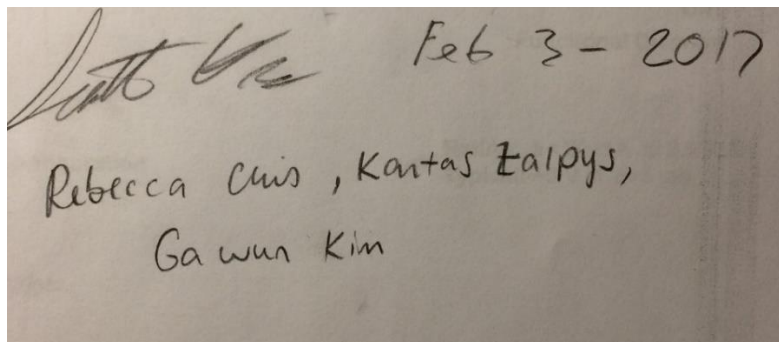
Learn from these experiments.

- * Set up external pin as general purpose I/O ports.
- * How to make an output that add the input values by using high level codes.

Conclusion (New experiment)

In this lab, though we helped from others, we built the codes by ourselves. Actually, before the lab1, we just pasted the code that was in the book. Honestly, at that time, after finishing the lab, I still could not figure it out. However, in this lab, I could understand how is this code going because we really strived to figure it out. This was a good experience and I will keep trying to studying for this study.

Lab tutor signature for certificate.



Feb 3 - 2017

Rebecca Chris, Kantas Ealpyis,
Gawun Kim