**Christian Sfeir**

**Lab experiment 5**

**Objectives:**

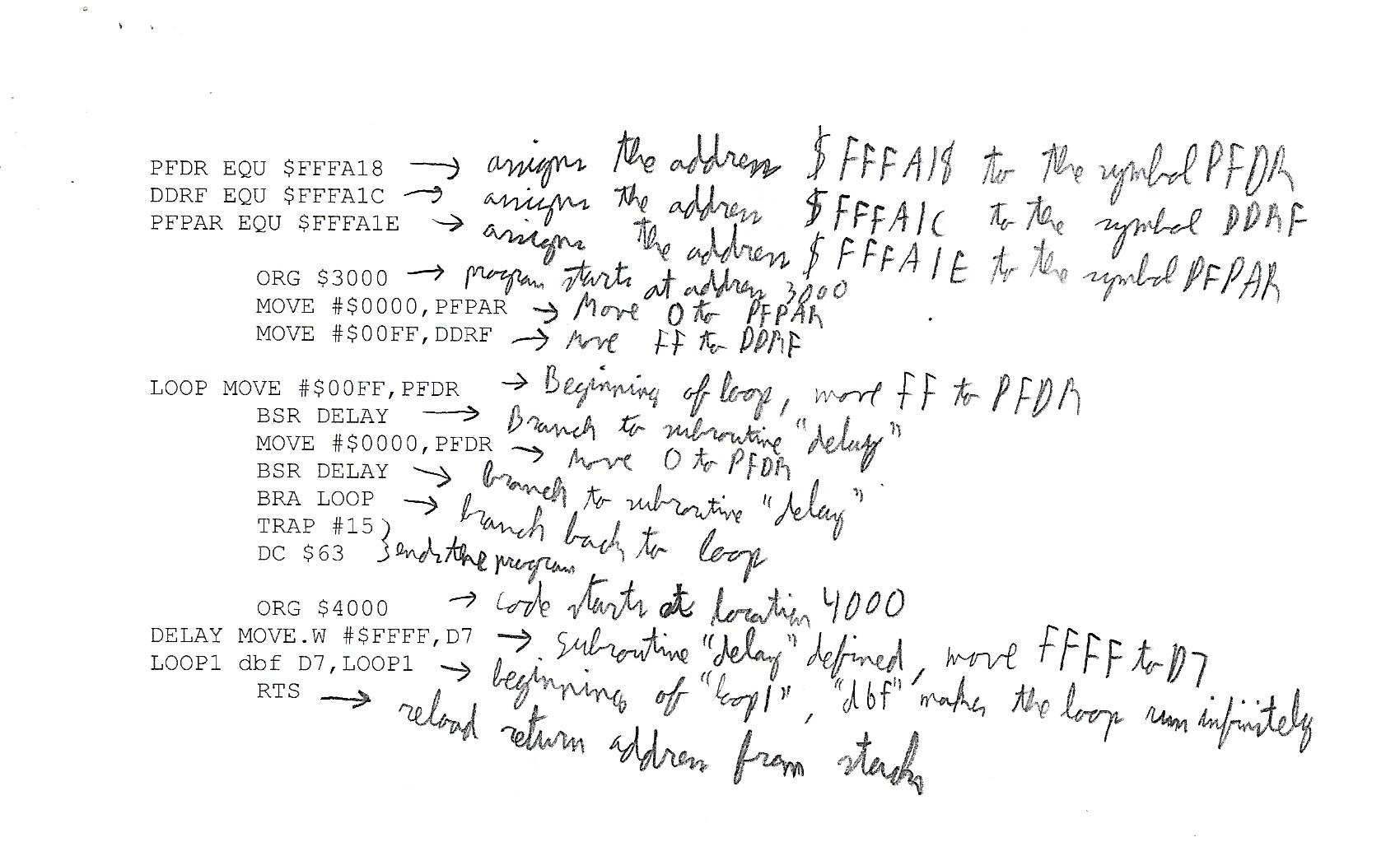
The objectives of this experiment are to use the parallel port in order to control the LED lights and LCD display on the Motorola 68000 board.

**Introduction:**

The LCD display on the Motorola 68k is used to display ASCII characters and numbers. This display is controlled by the HD44780 driver. It is capable of read/write signals, register select signals, enable signal, and has an 8 bit data bus. In order to communicate with the LCD display, parallel port interfaces are used on the CMM332 board, and these ports are used by writing to control registers mapped to specific memory locations. Ports E and F will be used, and the registers PEPAR (Port E pin assignment register), DDRE (Port E data direction), PEDR (Port E data register), PFPAR (Port F pin assignment register), DDRF (port F data direction), and finally PFDR (Port F data register) will be used. The different commands that can be issued to the LCD display are given in the manual of the Motorola 68000 board. In this experiment, two assembly language programs must be written and implemented on the Motorola 68k device. The first program will control the LED’s located on the Motorola 68k and make them flicker, with approximately a 0.5 seconds delay. The second program will print out my name “CHRISTIAN” on the LCD display of the board.

**Results:**

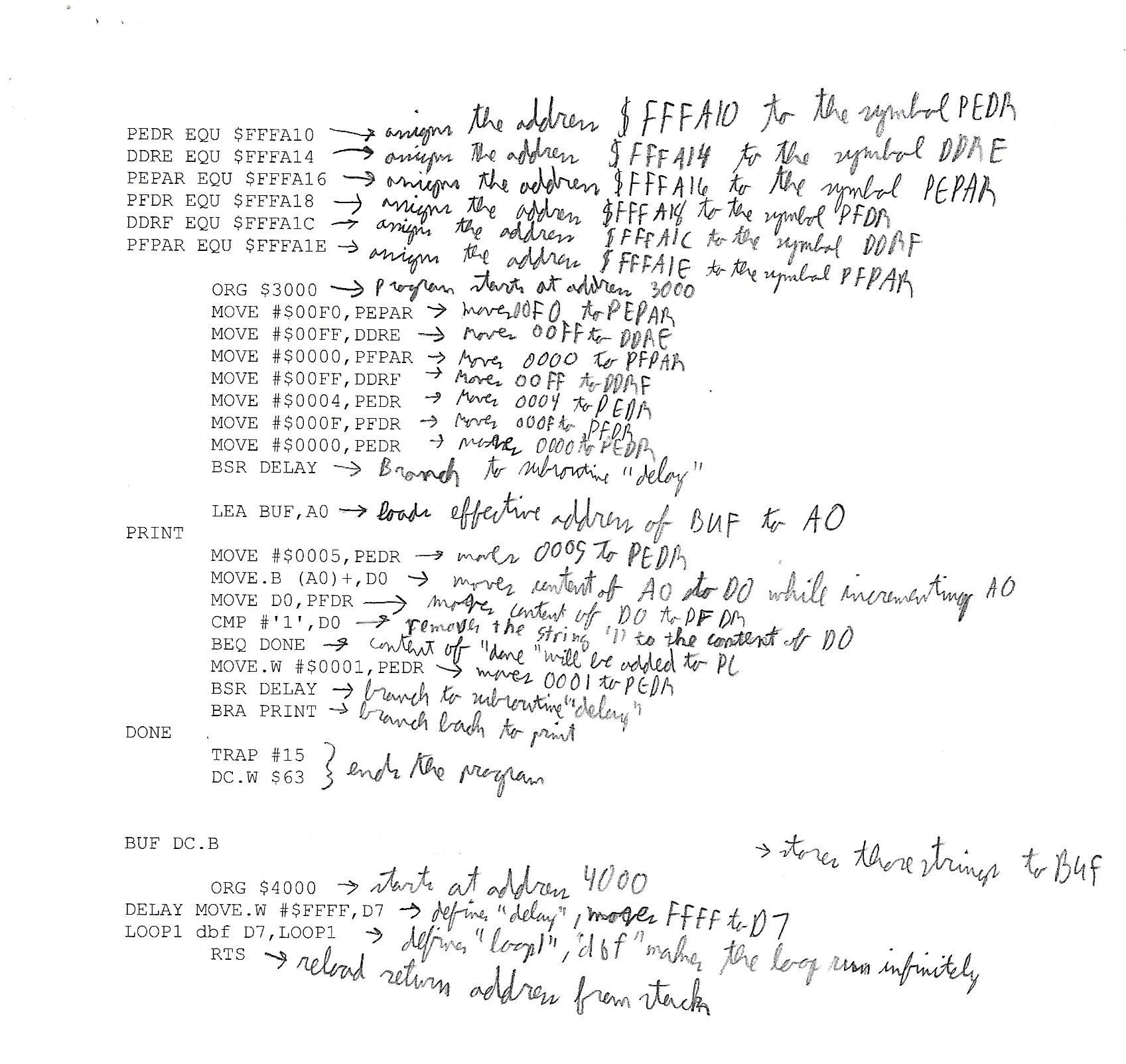
The following code was written for the first program which controls the LED’s:



The associated LST file which was obtained is:

|  |
| --- |
| 00fffa18 PFDR EQU $FFFA18  00fffa1c DDRF EQU $FFFA1C  00fffa1e PFPAR EQU $FFFA1E    00003000 ORG $3000  00003000 33fc 0000 00ff MOVE #$0000,PFPAR  fa1e  00003008 33fc 00ff 00ff MOVE #$00FF,DDRF  fa1c    00003010 33fc 00ff 00ff LOOP MOVE #$00FF,PFDR  fa18  00003018 60f6 BRA LOOP  0000301a 33fc 0000 00ff MOVE #$0000,PFDR  fa18  00003022 6100 0fdc BSR DELAY  00003026 60e8 BRA LOOP  00003028 4e4f TRAP #15  0000302a 0063 DC $63    00004000 ORG $4000  00004000 3e3c ffff DELAY MOVE.W #$FFFF,D7  00004004 51cf fffe LOOP1 dbf D7,LOOP1  00004008 4e75 RTS dbf D7,LOOP1  ===== 0 Error(s)  ===== 0 Warning(s) |

Next, the following code was written for the second program which displayed “Christian” to the LCD board:



And the obtained LST file is the following:

|  |
| --- |
| 00fffa10 PEDR EQU $FFFA10  00fffa14 DDRE EQU $FFFA14  00fffa16 PEPAR EQU $FFFA16  00fffa18 PFDR EQU $FFFA18  00fffa1c DDRF EQU $FFFA1C  00fffa1e PFPAR EQU $FFFA1E    00003000 ORG $3000  00003000 33fc 00f0 00ff MOVE #$00F0,PEPAR  fa16  00003008 33fc 00ff 00ff MOVE #$00FF,DDRE  fa14  00003010 33fc 0000 00ff MOVE #$0000,PFPAR  fa1e  00003018 33fc 00ff 00ff MOVE #$00FF,DDRF  fa1c  00003020 33fc 0004 00ff MOVE #$0004,PEDR  fa10  00003028 33fc 000f 00ff MOVE #$000F,PFDR  fa18  00003030 33fc 0000 00ff MOVE #$0000,PEDR  fa10  00003038 6100 0fc6 BSR DELAY    0000303c 41f9 0000 306c LEA BUF,A0  PRINT  00003042 33fc 0005 00ff MOVE #$0005,PEDR  fa10  0000304a 1018 MOVE.B (A0)+,D0  0000304c 33c0 00ff fa18 MOVE D0,PFDR  00003052 0c40 0031 CMP #'1',D0  00003056 6700 0010 BEQ DONE  0000305a 33fc 0001 00ff MOVE.W #$0001,PEDR  fa10  00003062 6100 0f9c BSR DELAY  00003066 60da BRA PRINT  DONE  00003068 4e4f TRAP #15  0000306a 0063 DC.W $63      0000306c 414c 4553 5341 BUF DC.B  'C','H','R','I','S','T','I','A','N'  4e44 524f 31    00004000 ORG $4000  00004000 3e3c ffff DELAY MOVE.W #$FFFF,D7  00004004 51cf fffe LOOP1 dbf D7,LOOP1  00004008 4e75 RTS dbf D7,LOOP1  ===== 0 Error(s)  ===== 0 Warning(s) |

**Conclusion:**

Two programs needed to be written and uploaded on the Motorola 68k board. Both these programs used parallel ports. The first program was used to control the LED’s and make them flicker with a 0.5 second delay, and the code successfully performed that operation. The second program was used to display the name “CHRISTIAN” to the LCD display, this code was a success since my name was properly displayed. Therefore, parallel ports, as well as infinite loops were successfully used to write and implement these programs.