

İ.T.Ü.
Faculty of Computer and Informatics
Computer Engineering



MICROCOMPUTER LAB REPORT

Lab No : 4
Lab Date : 31.10.2014 Friday
Group : 14
Group Members : 150110040 - Çağatay Erdiz
150110131 - Mert Cingöz
150110801 - Umut Pilge
150110030 - Duran Eren
150110106 - Ahmet Gözüberk

Research Assistant : Mustafa Ersen

A) Introduction

The MC6821 Peripheral Interface Adapter provides the universal means of interfacing peripheral equipment to the MC6800 family of microprocessors. This device is capable of interfacing the MPU* to peripherals through two 8-bit bidirectional peripheral data buses and four control lines. No external logic is required for interfacing to most peripheral devices.

The functional configuration of the PIA* is programmed by the MPU during system initialization. Each of the peripheral data lines can be programmed to act as an input or output, and each of the four control/interrupt lines may be programmed for one of several control modes. This allows a high degree of flexibility in the overall operation of the interface.

MPU: Micro Processing Unit

PIA: Peripheral Interface Adapter

B) Programing 6821 Peripheral Interface Adapter

MC6821 has two bidirectional 8-bit buses. These two 8-bit buses can be programmed as input or output. MC6802 programs MC6821 using address and data buses.

Adress Table of MC6821	
\$8300	Port-A
\$8300	Data Direction Register A
\$8301	Control Register A
\$8302	Port-B
\$8302	Data Direction Register B
\$8303	Control Register B

If third bit of control register A is 0, \$8300 points to data direction register. If it is 1, \$8300 points to port-A value.

Simple Led Code

```
ldaa #$00
staa $101           ;third bit of control register A is 0.
                   ;Now, $8300 points data direction register.

ldaa #$FF
staa $100           ;Port-A is completely output.
ldaa #$04
staa $101           ;third bit of control register A is 1
ldaa #$55
staa $100           ;led0-2-4-6 are on, led1-3-5-7 is off.
swi
```

Simulator Results

SDK6800 Emulator v1.08 (www.HVRSsoftware.com)

Assembly Program

```

0001  ldaa #$00
0002  staa $101 ;third bi
0003
0004  ldaa #$FF
0005  staa $100 ;Port-A i
0006  ldaa #$04
0007  staa $101 ;third bi
0008  ldaa #$55
0009  staa $100 ;led0-2-4
0010  swi

```

Memory | Display | Reference ADDRESS: 0014

0000:	86 00 B7 01 01 86 FF B7 01 00 86 04 B7 01 01 86
0010:	55 B7 01 00 8F 00 00 00 00 00 00 00 00 00 00 00
0020:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0050:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0070:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0080:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0090:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00A0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00B0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00C0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00D0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00E0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00F0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100:	55 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0110:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0120:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0130:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

PC: 0014 X: 0000
SP: F000
IR: 00B7 STAA
ACCUMULATOR
A: 55 B: 00

Status Flags
0 0 0 0 0 0
H I N Z V C

Base Converter
7 8 9 F
4 5 6 E
1 2 3 D
0 A B C
Hex Dec Bin

Assembling program...
Syntax Check: OK

Break Disabled
Break At: 0000

Clear Load Save Step Run Stop

Note: Address information is changed in order to observe results easily.

- \$0100 is used instead of \$8300
- \$0101 is used instead of \$8301

C) Experiment

Lab Code

```

        ldx #arr
        ldaa #$04
        staa $0101
        lds #$0130

init  ldaa 0,x
      psha
      inx
      incb
      cmpb #8
      bne init
      ldab #8

loop  pula
      staa $0100
      incb
      cmpb #8
      beq return
      jsr delay
      bra loop

delay ldx #$0000

```

```

cnt   inx
      cpx #$000F
      beq term
      bra cnt

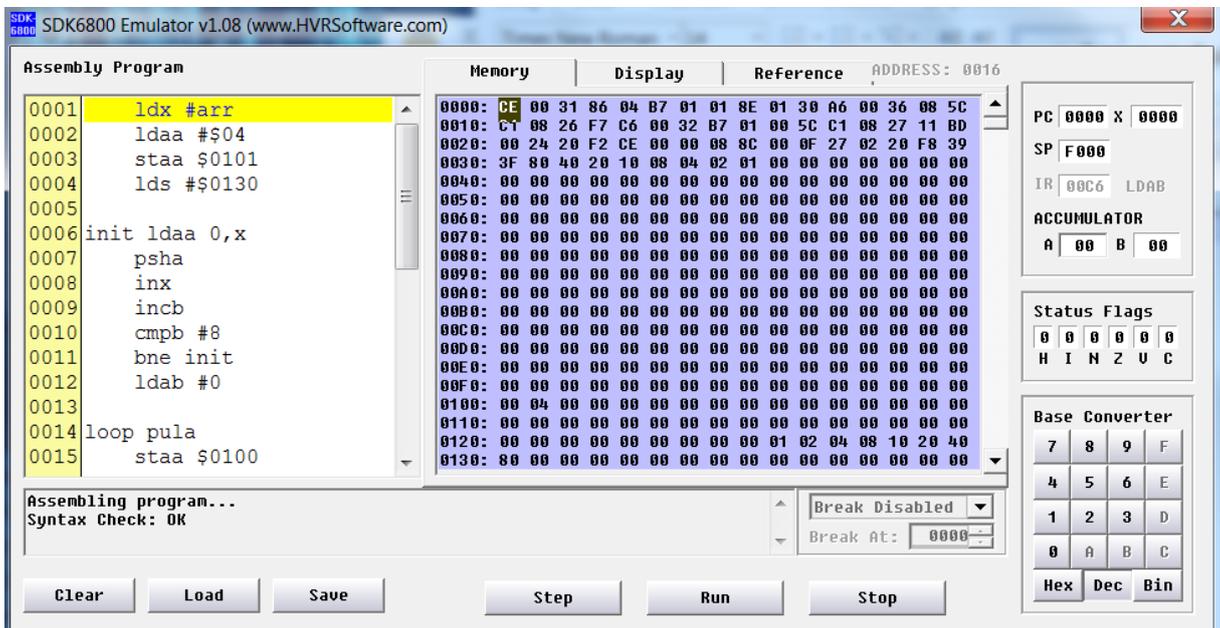
term  rts

return swi

arr .byte $80,$40,$20,$10,$08,$04,$02,$01

```

Simulator Results



Note: Address information is changed in order to observe results easily.

- \$0100 is used instead of \$8300
- \$0101 is used instead of \$8301

Led values are stored in stack and stack pointer starts from \$0130.