**İ.T.Ü.**

**Faculty of Computer and Informatics**

**Computer Engineering**



**MICROCOMPUTER LAB**

**REPORT**

|  |  |  |
| --- | --- | --- |
| **Lab No** | **:** 04 | |
| **Lab Date** | **:** 24.10.2013 | |
| **Group** | **:** 09 | |
| **Group Members** | **:** 040100014 Teoman Turan  040100018 Mustafa Durmuş  040100117 Tuğrul Yatağan  040100124 Emre Gökrem | |
|  |  | |
|  |  | |
|  |  | |
| **Research Assistant :** Mahiye Uluyağmur | | |  |
|  | |  |

**1. THE AIM/CONTENT of THE EXPERIMENT**

The purpose of doing this experiment is to get information about MC6821 peripheral interface adapter through a simulation of Knight Rider LED.

**2. EQUIPMENT**

ITU-Training kit has been used in the experiment. This kit consists of the following hardware components:

CPU: MC6802

Memory: 24K\*8 R/W + 16K\*8 Read Only

Address decoder

Control unit

Display and Keypad

Parallel Port

Serial Port

Programmable counter

In addition to this kit, for observing the simulation results during the experiment, one of our group members has used his own computer to run the simulation software of the microprocessor. Also, at the end of the experiment, another group member has used his own smartphone’s camera to videotape our Knight Rider LED simulation.

**3. THE PROGRAMS for THE EXPERIMENT**

**3.1. SECTION 2: SIMPLE LED CODE**

The machine codes of the instructions in the simple LED code program below had been obtained before coming to the lab. In the lab, the program has been run on ITU-Training Kit, then its result has been observed. Here are the machine codes of the instructions and comments about what they do:

86 00 LDAA #$00

// It immediately loads a direct data, $00, to Accumulator A.

B7 83 01 STAA $8301

// It stores the value in Accumulator A, 00000000, into $8301 that is the address of Control Register A. So, the third bit of Control Register A becomes 0. Now, it is pointing to Data Direction Register.

86 FF LDAA #$FF

// It immediately loads a direct data, $FF, to Accumulator A.

B7 83 00 STAA $8300

// It stores the value in Accumulator A, 11111111, into $8300 that is the address of Data Direction Register having been pointed by the previous STAA instruction. Now, Port A is completely output.

86 04 LDAA #$04

// It immediately loads a direct data, $04, to Accumulator A.

B7 83 01 STAA $8301

// It stores the value in Accumulator A, 00000100, into $8301 that is the address of Control Register A. So, the third bit of Control Register A becomes 0. Now, it is pointing to Port-A value.

86 55 LDAA #$55

// It immediately loads a direct data, $55, to Accumulator A.

B7 83 00 STAA $8300

// It stores the value in Accumulator A, 01010101, into $8300 that is the address of Data Direction Register. This leads that LEDs 0, 2, 4 and 6 are on, and the rest are off.

3F SWI

// It terminates the program.

As stated in the comment for the penultimate instruction, it has been observed that LEDs 0, 2, 4 and 6 blink whereas the rest don’t.

**3.2. SECTION 3: EXPERIMENT**

Here, “Knight Rider LED” that can be seen on YouTube has been simulated by filling the basic loop within the code fragment on the lab sheet. That code fragment in Assembly programming language forms a program for the following algorithm:

int leds[] = {0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80};

int i=0;

int main()

{

initPortA();

while(1)

{

for(i=0; i<8; i++)

{

writePortA(leds[i]);

delay(); // Approximately 1 second

}

}

}

This experiment consists of two ways: First, LEDs are on by turns with a small delay only in one direction. Then, LEDs shuttle by turns with a small delay both in a direction and its reverse one, which simples a Knight Rider LED.

The machine codes of the program have been obtained from the simulation software of the microprocessor.

One direction

7E 40 00 JMP $4000

.org $4000

86 00 LDAA #$00

B7 83 01 STAA $8301 // Third bit of Control Register A is 0.

// Now, $8300 points to Data Direction Register.

86 FF LDAA #$FF

B7 83 00 $8300 // Port-A is completely output.

86 04 LDAA #$04

B7 83 01 STAA $8301 // Third bit of Control Register A is 1.

86 00 LDAA #$00

B7 83 00 STAA $8300 // Initially all LEDs are off.

C6 00 LOOP LDAB #$00 // These blue codes are wanted.

CE 41 00 LDX #$4100

FF 41 10 STX $4110

4F CLRA

FE 41 10 F LDX $4110

E6 00 LDAB 0,x

F7 83 00 STAB $8300

08 INX

4C INCA

FF 41 10 STX $4110

BD 40 33 JSR DELAY

81 08 CMPA #$08

2D EC BLT F

20 E1 BRA LOOP

CE 00 00 DELAY LDX #$0000

08 CNT INX

8C 00 03 CPX #$0003 // Delay between two LEDs

27 02 BEQ TERM

20 F8 BRA CNT

39 TERM RTS

;------------------------------------------------

.org $4100

.byte $1, $2, $4, $8, $10, $20, $40, $80

Shuttling in two reverse directions

7E 40 00 JMP $4000

.org $4000

86 00 LDAA #$00

B7 83 01 STAA $8301 // Third bit of Control Register A is 0.

// Now, $8300 points to Data Direction Register.

86 FF LDAA #$FF

B7 83 00 $8300 // Port-A is completely output.

86 04 LDAA #$04

B7 83 01 STAA $8301 // Third bit of Control Register A is 1.

86 00 LDAA #$00

B7 83 00 STAA $8300 // Initially all LEDs are off.

C6 00 LOOP LDAB #$00 // These blue codes are wanted.

CE 41 00 LDX #$4100

FF 41 10 STX $4110

4F CLRA

7A 41 11 DEC $4111

FE 41 10 F LDX $4110

E6 00 LDAB 0,x

F7 83 00 STAB $8300

08 INX

4C INCA

FF 41 10 STX $4110

BD 40 33 JSR DELAY

81 08 CMPA #$08

2D EC BLT F

4F CLRA

7A 41 11 DEC $4111

FE 41 10 G LDX $4110 // Loop for reverse direction

E6 00 LDAB 0,x

F7 83 00 STAB $8300

08 DEX // For reverse direction

4C INCA

FF 41 10 STX $4110

BD 40 33 JSR DELAY

81 08 CMPA #$08

2D EC BLT G

20 C9 BRA LOOP

CE 00 00 DELAY LDX #$0000

08 CNT INX

8C 00 03 CPX #$0003 // Delay between two LEDs

27 02 BEQ TERM

20 F8 BRA CNT

39 TERM RTS

;------------------------------------------------

.org $4100

.byte $1, $2, $4, $8, $10, $20, $40, $80

The result is a simulation of Knight Rider LED in two reverse directions. It can be watched on the following YouTube video that has been videotaped via one of our group members’ smartphone: [www.youtube.com/watch?v=urY4WocIZtY](http://www.youtube.com/watch?v=urY4WocIZtY)