LAB 1

INTRODUCTION TO ITU-TRAINING KIT

1. INTRODUCTION

This lab is designed to introduce you ITU-Training kit and it gives information about how to program MC6802. ITU-Training Kit uses MC6802 Microprocessor and the programing language is Motorola 6800 assembly language.

ITU-Training Kit provides following features,

* Programs, which are written by machine codes, can be run.

* New peripherals can be added to MC6802.

2. ITU-TRAINING KIT HARDWARE

Basic components of ITU-Training Kit are listed below,

* CPU: MC6802

* Memory: 24K*8 R/W

        16K*8 Read Only

* Address decoder

* Control unit

* Display and keypad

* Parallel Port

* Serial Port

* Programmable counter

3. MC6802

Table 1. Memory addressing

Operating System	$FFFF-$E000
ROM for User	$DFFF-$C000
Not connected	$BFFF-$A000
Not connected	$9FFF-$8000
Not connected	$7FFF-$6000
R/W Memory	$5FFF-$4000
R/W Memory	$3FFF-$2000
R/W Memory	$1FFF-$0000

MC6802 is an 8-bit microcontroller. Operating frequency is 4 MHz. MC6802 has 128Byte on-chip memory. Address of the memory is between $0000-$007F. MC6802 supports external memories. In ITU-Training Kit, 25KB R/W and 16KB Read only memory are externally connected. External memory address spaces are shown in Table 1. When writing your programs, only use address space between $4000-$6000.

4. USAGE of ITU-TRAINING KIT
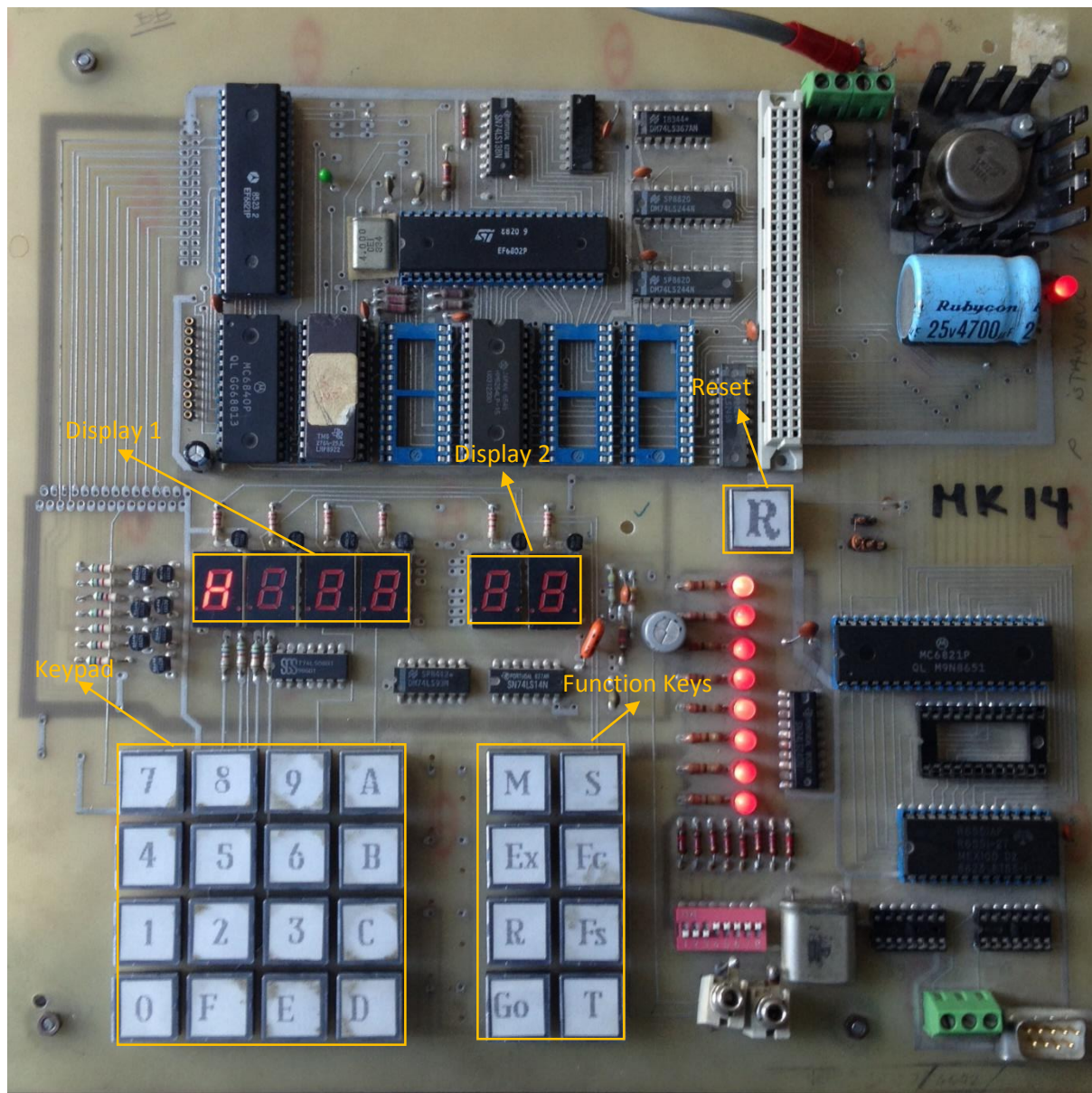
ITU-TRAINING KIT

Basic four operations of ITU-TRANING Kit.

1- Memory Editing
2- Offset Calculation
3- Running
4- Debugging
    - Register Observation
    - Step by Step Running

Memory Editing

When **Display 1** shows **H** , write hexadecimal four-digit address using **Keypad** and press **M**. **Display 1** shows address, **Display 2** shows the contents of the address. **M**(previous address) and **G**(next address) are used for address decrement or increment, respectively. If any hexadecimal 2-digit number is entered, current address location is updated and value is shown in **Display 2**. To exit memory tool press **Ex**.

Note: Before entering memory tool **Display 1** should show **H**, if **Display 1** does not show **H**, press **Ex**.

Offset Calculation

MC6802 uses address offset to jump short distance. For example, your program branches from $4000 to $400A, branch instruction at $4000 only stores ($09) which is the address difference. Difference can be negative or positive. ITU-Training Kit has the capability of offset calculation. Before calculating offset, you should enter memory tool. In memory editing tool, press **Fs**. Now **Display 2** shows **A**, enter address value you want to jump. Press **G**, offset value is shown **Display 2**. If you press **G** again, offset value will be written into current address location. Offset value cannot be more than **7F**. If offset exceeds **7F**, **Display 1** shows **CIS**.

Running

Running is easy. **Display 1** should show **H**. If it does not, press **Ex**. Write beginning address of your program and press **G**.

Debugging

Register observation

Display 1 should show **H**. Press **R** to show value of **AccA, AccB, SP, X, status register, PC**. **Display 2** shows register type and **Display 1** shows the contents. Press **M** and **G** to switch next or previous register. **Display 2** shows **PS**(Program Counter), **du**(status register), **A**(accumulator A), **B**(accumulator B), **YG**(stack pointer), **SI**(index register).

Step by Step Running(Tracing)

Display 1 should show **H**. Press **R** to show value of **AccA, AccB, SP, X, status register, PC**. Select **PS**(Program Counter) and enter beginning address of your program. Then, press **T** to execute one instruction.

5. EXPERIMENT

Please run below code on ITU-Training Kit. In your report what ıt does.

```
4000   4F                    clra
4001   5F                    clrb
4002   CE 44 00              ldx    #$4400
4005   AB 00      label      adda   0,x
4007   08                    inx
4008   5C                    incb
4009   C1 0A                 cmpb   #$0A
400B   2D F8                 blt    label
400D   B7 44 40              staa   $4440
4010   3F                    swi
```

In the report please answer the following question:

What are the values of Program Counter, Accumulator A, Accumulator B, Status Register and Stack Pointer, after execution of each instruction. Explain offset value calculation of the program.

Good Luck

Your Microprocessors' Lab Team