Microprocessor based System Design Laboratory (Gy)

Department of Computer Science and Technology, IIEST Shibpur

Date: July 26, 2024

Experiment No. 1: Familiarization with the Development Environment

Objective: To be familiar with microprocessor based System Development KIT (SDK).

SDK is a platform for developing **Microprocessor Based Systems**. Like, for development of **software applications**, you have your **laptop** (together with its **Operating System**, **Compiler** (gcc), **Debugger** (gdb), etc.). In this class you are required to get familiar with the given SDK so that you can use it for development of 8085 based systems.

Preamble: For microprocessor based small system design we need a minimum hardware environment which uses a particular processor (Eg, Intel 8086 in our case) and some supporting components like memory (both **RAM** and **EPROM**), **I/O** ports, buffers etc. This environment is commercially available as a platform for the small system design and development and known as System Development Kit (or simply, **SDK** or **KIT**). Normally, a small program (called Monitor progam/software) is preloaded in the EPROM of the kit. Monitor program is like the operating system (Eg, Microsoft Windows, Linux) of our Laptop or Desktop computers. When such a kit is powered on, its processor starts executing the monitor program, which in turn interacts with the user through the keypad/display for loading, running and debugging user programs.

Users may

1. first write (pen and paper) **8085 Assembly Language** programs using the instructions from **Instruction Set** of the processor of the kit,
2. and then consult the Instruction Set to manually translate (pen and paper) the Assembly Language program into **8085 Machine Language**.
3. This machine language program then can be “loaded” into the (RAM section of the) main memory of the SDK through its keypad.
4. Finally the user may execute the program (s)he has loaded at the given location of the main memory.

SDKs also provide a set of “connectors” using which one can connect external Input/Output devices.

Specification of a Typical KIT (Your KIT may have different specification):

- The Microprocessor - A CPU (8/16 bit popular general purpose CPU).
- Memory: RAM – 2K, EPROM – 2K. Memory may be enhanced 4K/8K or more.
- Man-Machine Interface: Input: Keypad with 24 (or more) keys; Output: 7 segment LED Display Panel with 6 or more display modules.
- I/O ports: 2 to 8 through standard Programmable Input Output (PIO) devices for I/O operations and peripheral interfaces.
- Communication : RS-232C compatible (e.g., a PC may be connected to the KIT through this serial channel) serial line.
- Monitor routine: Machine language routine (1K–2K in size) Residing in EPROM.
- Optional features: On-board EPROM programmer, Analog to Digital Conversion & Digital to Analog Converter etc.
- Storage (Secondary) : None

Facilities in the KIT: The keyboard along with the LED display unit serve as basic I/O media for communicating with the user. Programs are loaded in the RAM (user area) through key-press; and executed. Debugging, if required, is also done by using appropriate command/numeric key(s). The following can be done in a KIT:

- i. Reset the system (Cold Start).
- ii. Reset the system with preservation of processor status in memory (Warm Start).
- iii. Modify/examine the content of a memory location.
- iv. Execute a program.
- v. Debug a program by verifying/changing the contents of the accessible registers.
- vi. Set breakpoints and run the program in single step mode.

Tasks you should attempt

Get familiar with the SDK and its manual given to you.

1. Identify different hardware components
   - (a) **CPU** (the 8085 chip)
   - (b) **Main Memory** (ROM and RAM sections) – the chips, memory size, where in the memory space (0000H to FFFFH) they are interfaced. (Ref. Chapter 2, page 4 and Chapter 6, page 43 of the SDK user manual)
   - (c) **Input/Output (IO) devices** – The devices, chips, and where where in the memory space (0000H to FFFFH) or IO space (00H to FFH), they are interfaced. (Ref. Chapter 2, page 4 and Chapter 6, page 45 of the SDK user manual)
2. **How to use (interact with) the monitor program of the SDK**. (Ref. Chapter 3, page 7 of the SDK user manual)
   - (a) Examine/Modify Memory – mainly to be used to load/modify programs (Ref. Chapter 3, page 11 of the SDK user manual)
   - (b) Execute a progam which has already been loaded in the memory (Ref. Chapter 3, page 13 of the SDK user manual)
   - (c) Single Instruction or Single Stepping – Used to debug a program (Ref. Chapter 3, page 14 of the SDK user manual)
   - (d) Block Move - (Ref. Chapter 3, page 15 of the SDK user manual)
   - (e) Delete - (Ref. Chapter 3, page 16 of the SDK user manual)
   - (f) Insert - (Ref. Chapter 3, page 18 of the SDK user manual)
   - (g) Relocate - (Ref. Chapter 3, page 20 of the SDK user manual)
   - (h) Fill - (Ref. Chapter 3, page 21 of the SDK user manual)
   - (i) String - (Ref. Chapter 3, page 22 of the SDK user manual)
   - (j) Memory Compare - (Ref. Chapter 3, page 24 of the SDK user manual)
   - (k) Insert Data - (Ref. Chapter 3, page 25 of the SDK user manual)
   - (l) SuperB - (Ref. Chapter 3, page 26 of the SDK user manual)
   - (m) Simple Delete - (Ref. Chapter 3, page 27 of the SDK user manual)
3. **Load and execute the sample programs given in Chapter 6 Page 60 of the Manual.**

Report: The format of the Laboratory Report and the submission mechanism will be stated during class hours.