**Program No:** 1

**Date:** 13-07-2009

**STUDY OF TRAINER KIT**

**PROBLEM DEFINITION**

The study the 8085 microprocessor Trainer Kit.

**THEORETICAL BACKGROUND**

A microprocessor is a multipurpose, programmable logic device that reads binary instructions from a storage device called memory, accepts binary data as input and processes data according to these instructions, and provides results as output. A typical programmable machine can be represented with three components microprocessor, memory and I/O. These three components work together or interact with each other to perform a given task, thus they comprise a system. The physical components of the system are called hardware .A set of instructions written for the microprocessor to perform a task is called a program, and a group of programs is called software.

**8085 Microprocessor Architecture**

Architecture reveals the internal logic of a microprocessor. The 8085 architecture comprises of the following blocks:

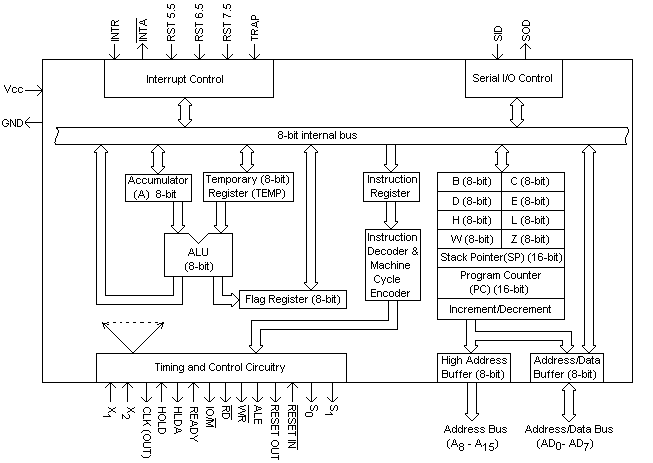
1. ALU logic
2. Register logic
3. Timing and Execution logic
4. Interrupt logic
5. Serial I/O logic

Internal architecture

The 8085 microprocessor has the following features :

* It is a 40 pin LSI chip
* It is an 8 – bit general purpose microprocessor with addressing capacity of 64K
* It operates with 3MHz single phase clock

The internal architecture of 8085 microprocessor is shown in the figure given .



ARITHMETIC LOGIC UNIT

The arithmetic-logic unit (ALU) performs all arithmetic operations(addition,subtraction, multiplication,and division) and logic operations. Logic operations test various conditions encountered during processing and allow for different actions to be taken based on the results. The data required to perform the arithmetic and logical functions are inputs from the designated CPU registers and operands. Logical operations compare numbers, letters and special characters.

Comparison operations test for three conditions:

1. equal-to condition in which two values are the same .
2. less-than condition in which one value is smaller than the other .
3. greater-than condition in which one value is larger than the other.

Relational operations (=, <, >) are used to describe the comparison operations used by the arithmetic logic unit. The arithmetic logic unit performs logic functions such as AND, OR and NOT.

CONTROL UNIT

The control unit maintains order within the computer system and directs the flow of traffic (operations) and data. The control unit selects one program statement at a time from the program storage area, interprets the statement, and sends the appropriate electronic impulses to the arithmetic-logic unit and storage section to cause them to carry out the instruction. The control unit does not perform the actual processing operations on the data. Specifically, the control unit manages the operations of the CPU, be it a single-chip microprocessor or a fill-size mainframe. It controls the flow of all data entering and leaving the computer. It accomplishes this by communicating or interfacing with the arithmetic-logic unit, memory, and I/O areas. It provides the computer with the ability to function under program control. The control unit is also capable of shutting down the computer when the power supply detects abnormal conditions.

**Registers**

ACCUMULATOR

The accumulator is an 8-bit register that is a part of arithmetic/logic unit (ALU). This register is used to store 8-bit data and to perform arithmetic and logical operations. The result of an operation is stored in the accumulator. The accumulator is also identified as register A. In a computer's central processing unit (CPU), an accumulator is a register in which intermediate arithmetic and logic results are stored. Without a register like an accumulator, it would be necessary to write the result of each calculation (addition, multiplication, shift, etc.) to main memory, perhaps only to be read right back again for use in the next operation. Access to main memory is slower than access to a register like the accumulator because the technology used for the large main memory is slower (but cheaper) than that used for a register.

FLAGS

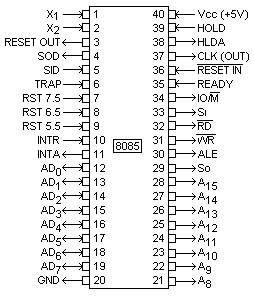
The ALU includes five flip-flops, which are set or reset after an operation according to data conditions of the result in the accumulator and other registers. They are called Zero(Z), Carry (CY), Sign (S), Parity (P), and Auxiliary Carry (AC) flags; their bit positions in the flag register are shown in the Figure below. The most commonly used flags are Zero, Carry, and Sign. The microprocessor uses these flags to test data conditions.

D7 D6 D5 D4 D3 D2 D1 D0

**S Z AC P CY**

For example, after an addition of two numbers, if the sum in the accumulator id larger than eight bits, the flip-flop uses to indicate a carry -- called the Carry flag (CY) -- is set to one. When an arithmetic operation results in zero, the flip-flop called the Zero(Z) flag is set to one. The first Figure shows an 8-bit register, called the flag register, adjacent to the accumulator. However, it is not used as a register; five bit positions out of eight are used to store the outputs of the five flip-flops. The flags are stored in the 8-bit register so that the programmer can examine these flags (data conditions) by accessing the register through an instruction.

**Intel 8085 pin configuration**



**Instruction set classification**

8085 instruction set consists of the following instructions:

1. *Data transfer instructions:*

|  |
| --- |
| MOV Move  MVI Move immediate  LDA Load accumulator directly from memory  STA Store accumulator directly in memory  LHLD Load H&L pair directly from memory  SHLD Store H&L pair directly in memory  LXI Load register pair with immediate data  LDAX Load accumulator from address in register pair  STAX Store accumulator in address in register pair  XCHG Exchange H&L with D&E  XTHL Exchange top of stack with H&L |

|  |
| --- |
| 2.*Arithmetic instructions*  ADD Add to accumulator  ADI Add Immediate data to accumulator  ADC Add to accumulator using carry flag  ACI Add Immediate data to accumulator using carry  SUB Subtract from accumulator  SUI Subtract immediate data to accumulator  SBB Subtract from accumulator using borrow (carry) flag  SBI Subtract Immediate data to accumulator using borrow (carry) flag  INR Increment specified byte by one  INX Decrement specified byte by one  DCR Increment register pair by one  DCX Decrement register pair by one |

3.*Logic instructions*

|  |
| --- |
| ANA Logical AND with accumulator  ANI Logical AND with accumulator using immediate data  ORA Logical OR with accumulator  OR Logical OR with accumulator using immediate data  XRA Exclusive logical OR with accumulator  XRI Exclusive logical OR with accumulator using immediate data  CMP Compare  CPI Compare using immediate data |

*The rotate instructions*

|  |
| --- |
| RLC Rotate accumulator left by one bit  RRC Rotate accumulator right by one bit  RAL Rotate left through carry by one bit  RAR Rotate right through carry by one bit |

*Complement and carry flag instructions*

|  |
| --- |
| CMA Complement accumulator  CMC Complement carry flag  STC Set carry flag |

*4.Branch group:*

The branching instructions alter the normal sequential program flow, either unconditionally or conditionally. The unconditional branching instructions are as follows:

|  |
| --- |
| JMP Jump  CALL Call  RET Return |

Conditional branching instructions examine the status of one of four condition flags to determine whether the specified branch is to be executed. The conditions that may be specified are as follows:

|  |
| --- |
| NZ Not Zero (Z=0)  Z Zero(Z=1)  NC No carry(C=0)  C Carry(C=1)  PO Parity odd(P=0)  PE Parity even(P=1)  P Positive(S=1)  M Negative(S=0) |

Two other instructions can affect a branch by replacing the contents or the program counter:

|  |
| --- |
| PCHL Move H&L to the program counter  RST Special restart instruction used with interrupts |

5. *Stack I/O and Machine control instructions:*

The following instructions affect the Stack and/or the Stack pointer:

|  |
| --- |
| PUSH Push two bytes of data onto the stack  POP Pop two bytes of data off the stack  XTHL Exchange top of the stack with H&L  SPHL Move content of H&L to stack pointer |

*I/O instructions*

|  |
| --- |
| IN Initiate input operation  OUT Initiate output operation |

*Machine control instructions*

|  |  |
| --- | --- |
| EI | Enable interrupt system |
| DI | Enable interrupt system |
| HLT | Halt |
| NOP | No Operation |

**8085 Microprocessor Trainer Kit**

A typical Microprocessor Trainer Kit will contain the following parts:

1. 8085 Microprocessor
2. 8255 Parallel Port Interface
3. 8251 Serial Interface
4. 8279 Keyboard and Display Controller
5. Parallel Port
6. Serial Ports
7. Seven Segment Display Array
8. Keyboard
9. Logic Gates and other electronic components
10. Battery
11. Memory Module

**Functions**

8085 Microprocessor: To process our inputs.

8255 Parallel Port Interface: To interface the 8085 with the parallel port for external inputs and outputs.

8251 Serial Interface: To interface the 8085 with serial input and output devices.

8279 Keyboard and Display Interface: To interface the Keyboard and Seven Segment Displays with The 8085. Sometimes, an additional chip 7447 may also be present to interface the seven segment display array.

Seven Segment Display array: To show the addresses and values in different memory locations and registers.

Keyboard: Unlike normal keyboards, the trainer kit has a built in keyboard, which has a set of 16 keys for specifying the hexadecimal inputs to the 8085 and a set of function keys that determine different operations of the kit.

**Function Keys**

EXMEM (Memory operations)

EXREG (Register operations)

SI (Step by Step Execution)

FILL/. (Execution/Termination)

GO (Start Execution)

NEXT (Next location/Memory)

VCT/INT(Interrupts)

SHIFT (Select between operations if the same key has multiple operations)

BM (Block Movement)

RESET (Refresh or reset the trainer kit)

Logic Gates: They help in different logical operations.

Battery: For backup of data.

Memory Module: To store the data entered for execution.

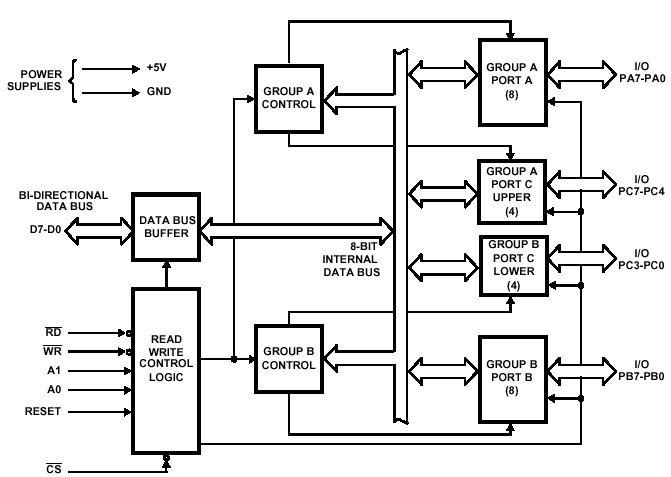
Every microprocessor trainer kit is associated with a user manual where the different possible operations using the kit are given and the procedure or steps to be followed are listed.

**Chips used in trainer kit**

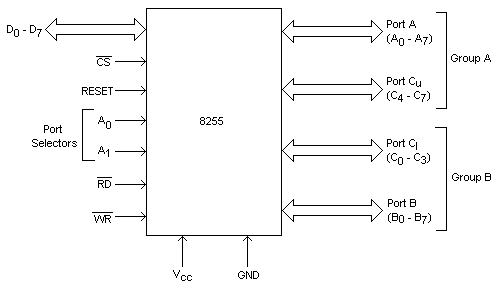
**8255**

8255 is a 40 pin IC. It is used to interface parallel devices (mainly parallel port) that are used in communication with slower devices like printer. It is a general purpose, programmable, parallel I/O device. It has 24 I/O pins that can be grouped primarily in two 8-bit parallel ports: A and B, with the remaining eight bits as port C. The eight bits of port C can be used as individual bits or can be grouped in two 4-bit ports: Cupper & Clower. The functions of these ports are defined by writing a control word in the control register.

Architecture of 8255



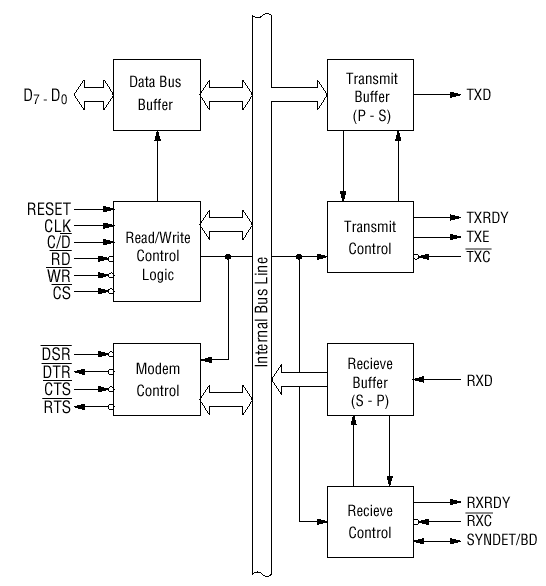
The 8255 chipset is as follows:



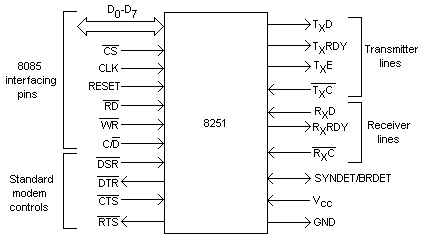
**8251**

USART stands for Universal Synchronous Asynchronous Receiver Transmitter. 8251 is a commonly used USART. It is used for serial data transmission. It is compatible with 8085, 8086 and 8088. As peripheral device of a MP system, the 8251 receives parallel data from the CPU and transmits serial data after conversion. This device also receives serial data from outside and transmits parallel data to the CPU after conversion.

Functional block diagram



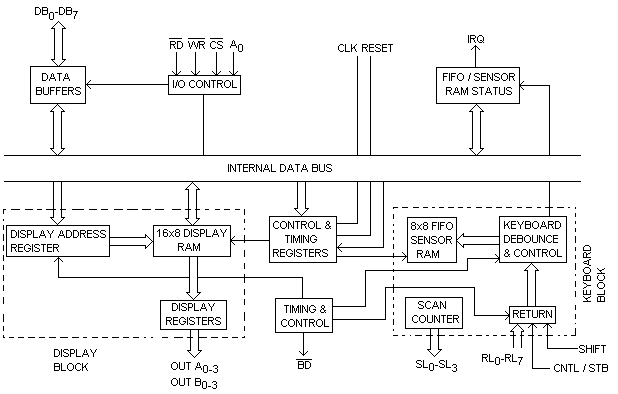
Pin-out of 8251



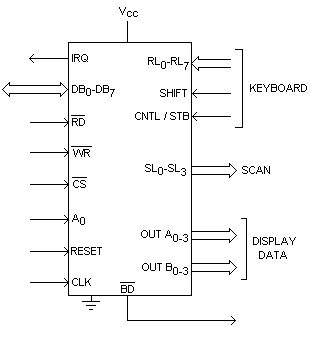
**8279**

8279 is a programmable Keyboard/Display interface. It can be used to interface a keyboard or/and display with 8085 microprocessor.

###### **Functional block diagram**



###### **Pinout**



**I/O operation**

The Microprocessor Trainer Kits can differ in the way in which the I/O operations are handled. This gives rise to 2 types.

1. Memory Mapped I/O
2. I/O mapped I/O

Memory mapped I/O

In this the MP uses16 address lines to identify on I/O device. Here, the I/O device is connected as a memory register itself. The MP uses the same control signal (Memory Read & Memory Write) & instructions as those of the memory. So, in this type of I/O mapping, the MP follows the same steps as if it is accessing a memory register. ie, IO/M’ = 0 Eg:- MOV A,M.

In this example, let HL pair have the value 8000. So, if an I/O device is connected to the location 8000, the value in the I/O device is taken to the accumulator.

I/O mapped I/O

In I/O mapped I/O, the MP uses 8 address lines to identify an I/O device. They are differentiated by the control signals themselves (I/O Read &I/O Write). The range of part ids from 00H to FFH is called I/O map. But here, the devices need interfacing since only one is to be activated at once. This depends on the processor status. The devices use CS (Chip select) to do this. But for memory mapping, due to internal address decoding, this wasn’t needed. Also, only half the address bus is used here. Eg: IN 08H Here, I0/M’ = 1.

Usually, a latch is used to interface O/P devices & a tristate buffer to interface I/P devices.

**Important key combinations in the 8085 Trainer Kit**

*Enter/modify data/code:*

EXMEM + Starting address + Value + NEXT (Continue these 2 steps till the data or code is finished) + RESET

*Execution:*

GO + Starting address of code + FILL + RESET (to check output, follow the previous method)

*Verify registers:*

SHIFT + EXREG + Register (choose using keypad, from A to L)

*Block movement of data/code:*

BM + Starting address of block + NEXT + Ending address of block + NEXT + destination address of block + TERMINATE (.)

**CONCLUSION**

The 8085 Microprocessor Trainer Kit was studied and the different chipsets were familiarized with successfully.