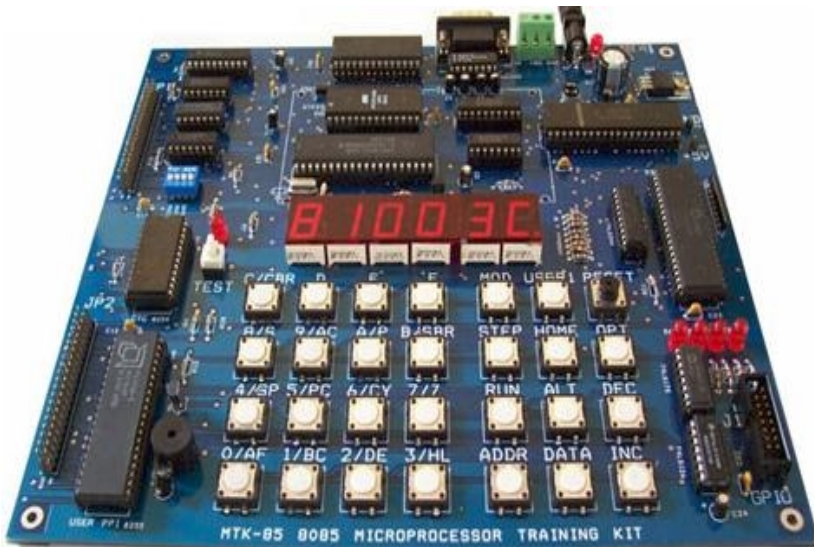


MTK-85TM

8085 Microprocessor Training Kit

Quick Start



The MTK-85 is a single board computer. The board is controlled by the 8-bit microprocessor, 8085. The built-in monitor program provides easy computer code entering using HEX digit to the memory location. The kit is best for learning the fundamental of today's computer.

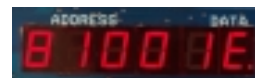
A. Kit Components

1. MTK-85 8085 single board computer
2. AC adapter
3. RS232 cable
4. 20x4 LCD*
5. CD-ROM (User Manual, Terminal Emulator, Monitor Source Code)

*Plugging or removing the LCD must be done when the board is power off.

B. Display and Keypad

When the board is powered with AC adapter, the cold boot message, "8085" will display on the 7-segment display.



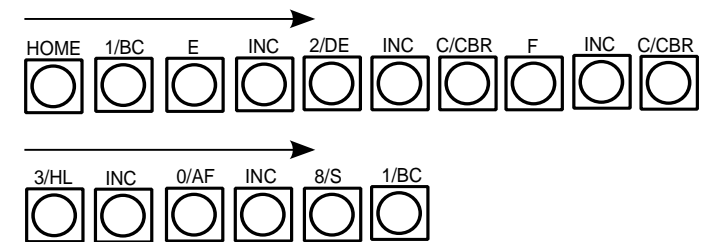
The board has 6-digit 7-segment LED. Four digits display memory location, and two digits for memory contents. Shown the location 8100 has the 8-bit data 1E.



The keypad has 16-key HEX digit, 0-F and 12-function keys. The DATA key or ADDR key selects between data or address entry when HEX key is pressed. HOME returns the home location to 8100. INC increments memory location.

C. Entering the HEX Code

Let us try enter the small program to the memory and run it.



You may check the memory contents with key DEC or INC. It will decrement or increment the memory location. Our program has only 6 bytes, i.e. 1E, 02, CF, C3, 00, 81.

ADDRESS	DATA
8100	1E
8101	02
8102	CF
8103	C3
8104	00
8105	81

Now run the program by pressing key HOME and RUN.



To stop running, press RESET key.

The above program in Assembly Language is as follows.

```

LOOP: MVI E,2 ; load register E with 02
      RST 1 ; call monitor subroutine
      JMP LOOP ; go back to loop location
  
```

The program has only three instructions. Now we will see the relationship between 8085 instructions and the HEX codes.

- Real computer built with the 8085 microprocessor
- von Neumann-architecture computer
- Designed for today's students
- Entering computer code directly to memory
- Full speed code execution
- Single instruction execution

Program listing is shown below.

```

ADDR DATA  INSTRUCTION  COMMENT
8100 1E 02    MVI E,2      ; load register E with 02
8102 CF       RST 1       ; restart 1
8103 C3 00 81 JMP 8100    ; jump back to 8100

```

The small program above demonstrates how to enter the HEX codes that makes text "8085" moving. The program begins with instruction that loads register E with 02 and then calls the subroutine that makes such text running by using RST 1 instruction. To repeat forever, the JUMP instruction will force the CPU to go back to location 8100.

The HEX code of the first instruction MVI E,2 is composed of two bytes, i.e. 1E and 02. CF is for RST 1 instruction and the last instruction, JMP 8100 has three bytes, i.e. C3, 00, and 81.

D.Single Instruction Execution

Now try using the tool for single instruction execution. This feature helps learning the 8085 instruction clearly. The program will be as follows.

```

XRA A ; clear accumulator
LOOP: OUT 0 ; write the accumulator to PORT 0
INR A ; increment accumulator by one
JMP LOOP ; go back LOOP

```

The first instruction, XRA A , is exclusive OR operation with the same value, thus clear the accumulator to 0. The 2nd instruction OUT 0, writes the accumulator to the PORT located at 0. The 3rd instruction, INR A, increments the accumulator by one. And the last is JUMP back to LOOP.

Try translate above program to 8085 machine code using the instruction table shown in page 4 and 5. Enter the hex code to memory started at HOME address (8100). When finish press HOME. Push STEP to execute the instruction, we will see the binary counting up on LED when the OUT 0 is executed.

8085 Instruction HEX code

MOVE, LOAD and STORE

40 MOV B,B	6B MOV L,E
41 MOV B,C	6C MOV L,H
42 MOV B,D	6D MOV L,L
43 MOV B,E	6E MOV L,M
44 MOV B,H	6F MOV L,A
45 MOV B,L	70 MOV M,B
46 MOV B,M	71 MOV M,C
47 MOV B,A	72 MOV M,D
48 MOV C,B	73 MOV M,E
49 MOV C,C	74 MOV M,H
4A MOV C,D	75 MOV M,L
4B MOV C,E	77 MOV M,A
4C MOV C,H	78 MOV A,B
4D MOV C,L	79 MOV A,C
4E MOV C,M	7A MOV A,D
4F MOV C,A	7B MOV A,E
50 MOV D,B	7C MOV A,H
51 MOV D,C	7D MOV A,L
52 MOV D,D	7E MOV A,M
53 MOV D,E	7F MOV A,A
54 MOV D,H	
55 MOV D,L	3E nn MVI A,byte
56 MOV D,M	06 nn MVI B,byte
57 MOV D,A	0E nn MVI C,byte
58 MOV E,B	16 nn MVI D,byte
59 MOV E,C	1E nn MVI E,byte
5A MOV E,D	26 nn MVI H,byte
5B MOV E,E	2E nn MVI L,byte
5C MOV E,H	36 nn MVI M,byte
5D MOV E,L	

COMPARE

FE nn CPI byte	01 nnnn LXI B,db1e
B8 CMP B	11 nnnn LXI D,db1e
B9 CMP C	21 nnnn LXI H,db1e
BA CMP D	31 nnnn LXI SP,db1e
BB CMP E	
BC CMP H	02 STAX B
BD CMP L	12 STAX D
BE CMP M	0A LDAX B
BF CMP A	1A LDAX D
	32 nnnn STA adr
	3A nnnn LDA adr
	22 nnnn SHLD adr
	2A nnnn LHLD adr
	EB XCHG

ROTATE

07 RLC
17 RAL
0F RRC
1F RAR

STACK

C5 PUSH B
D5 PUSH D
E5 PUSH H
F5 PUSH PSW
C1 POP B
D1 POP D
E1 POP H
F1 POP PSW
E3 XTHL
F9 SPHL
33 INX SP
3B DCX SP

ARITHMETICS

C6 nn ADI byte	80 ADD B
CE nn ACI byte	81 ADD C
	82 ADD D
	83 ADD E
	84 ADD H
	85 ADD L
	86 ADD M
	87 ADD A
	88 ADC B
	89 ADC C
	8A ADC D
	96 SUB M
	97 SUB A
	98 SBB B
	99 SBB C
	9A SBB D
	9B SBB E
	9C SBB H
	9D SBB L
	9E SBB M
	9F SBB A
	09 DAD B
	19 DAD D
	29 DAD H
	39 DAD SP

8085 Instruction HEX code

CALL

CD nnnn CALL adr
DC nnnn CC adr
D4 nnnn CNC adr
CC nnnn CZ adr
C4 nnnn CNZ adr
F4 nnnn CP adr
FC nnnn CM adr
EC nnnn CPE adr
E4 nnnn CPO adr

RETURN

C9 RET
D8 RC
D0 RNC
C8 RZ
C0 RNZ
F0 RP
F8 RM
E8 RPE
E0 RPO

RESTART

C7 RST 0
CF RST 1
D7 RST 2
DF RST 3
E7 RST 4
EF RST 5

SPECIALS

2F CMA
37 STC
3F CMC
27 DAA

INPUT/OUTPUT

DB nn IN byte	04 INR B
D3 nn OUT byte	0C INR C
	14 INR D
	1C INR E
	24 INR H
	2C INR L
	34 INR M
	3C INR A
	03 INX B
	13 INX D
	23 INX H
	05 DCR B
	0D DCR C
	15 DCR D
	1D DCR E
	25 DCR H
	2D DCR L
	35 DCR M
	3D DCR A
	0B DCX B
	1B DCX D
	2B DCX H

JUMP

C3 nnnn JMP adr
DA nnnn JC adr
D2 nnnn JNC adr
CA nnnn JZ adr
C2 nnnn JNZ adr
F2 nnnn JP adr
FA nnnn JM adr
EA nnnn JPE adr
E2 nnnn JPO adr
E9 PCHL

LOGICAL

E6 nn ANI byte	A0 ANA B
EE nn XRI byte	A1 ANA C
F6 nn ORI byte	A2 ANA D
	A3 ANA E
	A4 ANA H
	A5 ANA L
	A6 ANA M
	A7 ANA A
	A8 XRA B
	A9 XRA C
	AA XRA D
	AB XRA E
	AC XRA H
	AD XRA L
	AE XRA M
	AF XRA A
	B0 ORA B
	B1 ORA C
	B2 ORA D
	B3 ORA E
	B4 ORA H
	B5 ORA L
	B6 ORA M
	B7 ORA A

NOTE

nn is 8-bit constant
 nnnn is 16-bit constant
 Exp. MVI A,5B = 3E 5B
 JMP 8102 = C3 02 81

More details, please study the Kit User Manual.

Contact Information

E-mail: wiclit.sirichote@yahoo.com