

**LAPORAN PRAKTIK
MATA KULIAH MIKROPROSESOR**



PENGOPERASIAN TRAINER MPF – 1

Disusun oleh :

MUSLIKHIN

NIM. 05507134012

**TEKNIK ELEKTRONIKA FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
YOGYAKARTA 2006**

TEKNIK ELEKTRONIKA			
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA			
MIKROPROSESOR	PENGOPERASIAN TRAINER MPF – 1	LK 01	4X50”

A. TUJUAN

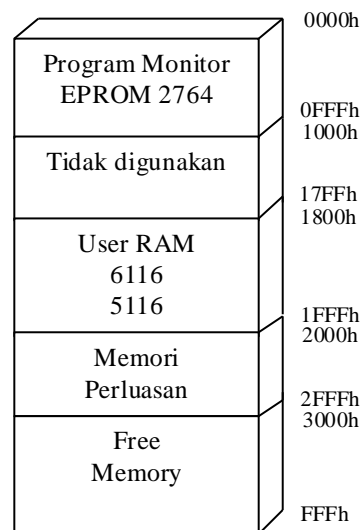
1. Menguraikan peta memori trainer MPF-1
2. Menampilkan dan mengganti data suatu register
3. Menampilkan data pada suatu lokasi ROM
4. Menampilkan data pada suatu lokasi RWM
5. Memasukkan, mengeksekusi, dan menguji program

B. PERALATAN YANG DIPERLUKAN

Trainer MPF – 1

C. TEORI DASAR

Komputer mikro MPF-1 merupakan suatu trainer khusus sebagai alat bantu mempelajari mikroprosesor Zilog Z-80 CPU. Peta memori MPF-1 ditunjukkan seperti Gambar berikut :



Peta Memori MPF-1

Program monitor menempati daerah EPROM/ROM yang hanya dapat dibaca (Read Only) dan tidak dapat ditulisi. Program monitor menempati daerah Address 0000h - 0FFFh.

Untuk program aplikasi dan data terolah dapat ditempatkan pada daerah RAM (Random Access Memory) pada Address 1800h-1FFFh. Dalam daerah tersebut dapat saja dimasukkan lebih dari satu program asal saja penempatannya tidak tumpang tindih. Jadi jika akan memasukkan program, dapat dipilih Address awal 1800h sampai dengan Address akhir 1FFFh. Daerah perluasan memori pada Address 2000h sampai dengan 2FFFh dapat diisi EPROM atau RAM yang sesuai.

D. LANGKAH PERCOBAAN

Pada percobaan ini mahasiswa akan mempelajari cara pengoperasian trainer MPF-1. Anda akan mempelajari kegunaan tombol keyboard yang ada pada MPF-1.

1. Cara Menghidupkan Trainer MPF - 1

Dalam pengoperasian MPF-1 pertama menyiapkan catu daya (adaptor) dan memeriksa apakah catu daya tersebut bekerja baik dengan keluaran DC 5 Volt. Soket DC dihubungkan paralel ke konektor daya disebelah kanan atas. Bila sumber daya telah masuk dengan benar pada MPF-1 maka pada monitor akan muncul pola sbb:

Tekan sekali tombol RESET (RS) yang terletak di pojok kiri atas, jika ditekan maka unit peraga akan menunjukkan pola yang sama. Pola tersebut akan selalu ditampilkan setiap kali prosesor memasuki program monitor.

2. Melihat dan Merubah Isi Suatu Register

Untuk melihat data yang ada pada suatu register, maka pertama-tama menekan tombol REG. Setelah tombol ditekan, peraga tujuh segment akan menampilkan tulisan "REG" yang menyatakan MPF-1 siap untuk menyajikan Register. Kemudian menekan tombol sesuai dengan nama register yang akan dilihat isinya. Setiap tombol mewakili register 16 bit.

Contoh :

Misalnya kita akan melihat isi register AF maka menekan tombol sebagai berikut:



Maka peraga akan menampilkan pola :



Dimana XX di sebelah kiri adalah data 8 bit yang akan berada di register A dan XX di kanannya adalah data 8 bit yang berada di register F. Untuk melihat isi register lainnya dengan menekan tombol pasangan register yang dikehendaki, maka unit peraga akan menampilkan pola yang sama dengan contoh diatas.

2.1 Memeriksa dan Mencatat Isi Register:

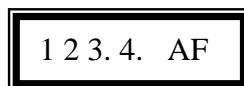
Register	AF	BC	DE	HL	AF'	BC'	DE'	HL'	IX	IY	SP
Data											

Untuk mengganti isi (data) yang berada pada suatu register pertama-tama melakukan prosedur seperti yang dilakukan untuk melihat isi register. Kemudian menekan tombol "DATA" dan menekan tombol heksa-desimal sebagai data baru sesuai dengan besarnya data yang dikehendaki.



Contoh :

Maka unit peraga akan menampilkan :



Untuk mengganti isi register BC cukup menekan tombol "+" kemudian memasukkan data heksa desimal seperti yang dikehendaki. Demikian seterusnya dengan pola susunan urutan register selengkapnya sebagai berikut :

AF BC DE HL AF' BC' DE' HL' IX' IY SP IF FH FI FH' FL'

Di samping untuk maju satu langkah register, MPF-1 juga menyediakan tombol untuk mundur satu langkah yaitu tombol “-“.

2.2 Memeriksa dan mengganti isi register-register dengan data berikut :

Register	AF	BC	DE	HL	AF'	BC'	DE'	HL'	IX	IY	SP
Data	0123	4567	890a	0B0C	0D0E	0FF0	F1F2	F3F4	F5F6	F7F8	1800

3. Melihat dan Merubah Isi suatu Lokasi Memori

Untuk melihat data yang ada pada suatu lokasi memori baik ROM/EPROM maupun RAM, caranya adalah dengan menekan tombol



dan peraga akan menampilkan pola :



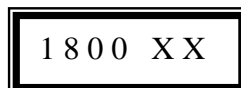
X.X.X.X. menunjukkan lokasi Address dari memori yang dilihat isinya. Dan XX menunjukkan isi pada Address tersebut. Untuk merubah lokasi atau Address memori cukup menekan tombol “ AADR” diikuti dengan menekan tombol heksa-desimal empat kali.

Contoh :

Melihat data yang berada pada suatu lokasi memori Address 1800h, menekan tombol berikut

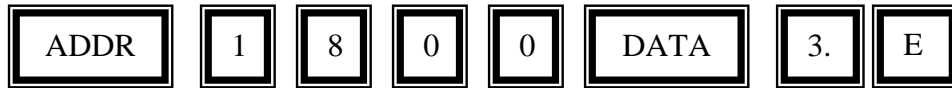


Maka peraga akan menampilkan :

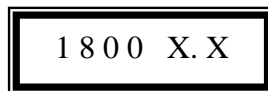


Data pada suatu lokasi memori RAM (1800 – 1FFF) dapat dirubah, sedangkan data pada suatu lokasi memori ROM (0000 – 0FFF) hanya dapat dibaca tanpa dapat dirubah. Untuk merubah data pada lokasi RAM digunakan tombol “DATA” dan

memasukkan dua kode angka heksadesimal sebagai data baru. Untuk memasukkan atau mengganti data pada lokasi Address 1800 dengan 3E dengan menekan tombol berikut :



Pada unit peraga akan menampilkan pola :



3.1. Memeriksa dan mencatat data yang berada pada lokasi memori Address berikut :

Memori	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	000A
Data	06	00	10	FF	3E	90	D3	03	3E	C0	D3

3.2 Kemudian data tersebut diganti dengan data baru seperti tabel berikut :

Memori	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	000A
Data	00	11	22	33	44	55	66	77	88	99	AA

Ternyata data pada Address tersebut tidak dapat diganti atau dirubah karena address tersebut adalah address ROM. Untuk melihat atau merubah data pada suatu blok memori (beberapa lokasi memori dengan Address berurutan) dapat dilakukan dengan lebih cepat menggunakan tombol “+” untuk menunjuk satu lokasi di atas Address yang sedang aktif.

Untuk mengisi data pada lokasi memori seperti pada tabel berikut, maka pertama mengisi data 3Eh pada alamat 1800h seperti cara yang telah diuraikan sebelumnya. Kemudian menekan tombol “+” untuk menuju ke address 1801h dan isikan data 00h pada Address tersebut kemudian mengulangi menekan tombol “+” untuk menunjuk Address 1802. Demikian seterusnya sampai data terakhir pada Address 180A terisikan.

Memori	1800	1801	1802	1803	1804	1805	1806	1807	1808	1809	1800A
Data	3E	00	3C	47	04	48	FB	FF	00	00	00

3.3. Memeriksa kembali data-data pada blok Address Memori 1800 sampai dengan 1804 dan mencocokkan kembali dengan tabel diatas.

4. Menyisipkan dan Menghapus Data Pada Memori.

Untuk menyisipkan data pada suatu Address memori dapat dilakukan dengan menekan tombol “INS” = insert dilanjutkan dengan menekan tombol data heksadesimal yang disisipkan.

4.1 Masukkan data 00h sampai dengan 05h pada Address 1900h sampai dengan 1905h dan mencatat hasilnya seperti tabel berikut :

Memori	1900	1901	1902	1903	1904	1905	1906	1907	1908	1909	190A
Data	00	01	02	03	04	05	FF	FF	00	00	00

4.2 Menyisipkan dua byte data 10h dan 20h pada Address 1901 dan 1902 dengan cara menekan tombol sebagai berikut :



4.3 Memeriksa data pada Address memori 1900h sampai dengan 190Ah dan mencatat hasilnya pada tabel berikut ini :

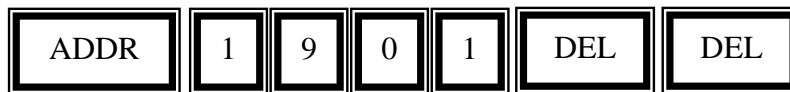
Memori	1900	1901	1902	1903	1904	1905	1906	1907	1908	1909	190A
Data	00	10	20	01	02	03	04	05	FF	FF	00

4.4. Jelaskan apa yang terjadi pada Langkah 4.1 s.d Langkah 4.3

Dapat dijelaskan bahwa yang terjadi pada langkah 4.1 sampai dengan langkah 4.3. adalah cara untuk menyisipkan suatu data pada alamat tertentu.

4.5. Untuk menghapus data pada suatu Address memori dapat dilakukan dengan menekan tombol “ DEL” = delete.

Untuk menghapus data 10h pada alamat 1901 dan data 20h pada Address 1902, maka dengan cara menekan tombol berikut ini :

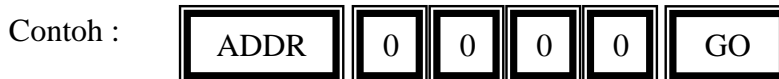


Selanjutnya dapat dibaca data pada Address berikut ini :

Memori	1900	1901	1902	1903	1904	1905	1906	1907	1908	1909	190A
Data	00	01	02	03	04	05	FF	FF	00	00	00

5. Menjalankan (Execute/Run/Go) Program

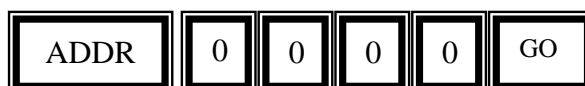
Untuk menjalankan atau execute program, dengan cara memasukkan Address awal program yang akan dieksekusi, lalu menekan tombol “GO” sebagai perintah mulainya pelaksanaan eksekusi.



5.1. Contoh memasukkan program sederhana mulai dari Address 1800h dengan memasukkan Op-Code (sandi operasi) ke dalam unit memori.

No	Address	Op-Code	Mnemonic	Keterangan
1.	1800	3E 00	LD A, 00H	A ← 00H
2.	1802	3C	INC A	A ← A+1
3.	1803	47	LD B, A	B ← A
4.	1804	04	INC	B ← B +1
5.	1805	48	LD C, B	C ← B
6.	1806	FF	RST 38	STOP / END

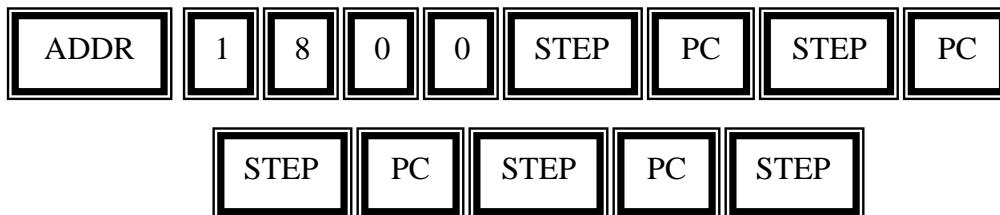
5.2. Menjalankan program dengan menekan tombol



5.3. Memeriksa isi register A, B, C

Register	A	B	C
Data	01	02	02

Di samping tombol eksekusi “GO” MPF-1 dilengkapi dengan tombol “STEP” yang digunakan untuk eksekusi Step demi Step, dengan cara menekan tombol berikut ini :



5.4. Mencatat data register-register dan mengisinya pada tabel berikut :

Data register	A	B	C
Step 1	00	XX	XX
Step 2	01	XX	XX
Step 3	01	01	XX
Step 4	01	02	XX
Step 5	01	02	02

5.5. Perbedaan antara tombol eksekusi “GO” dengan tombol “Step”

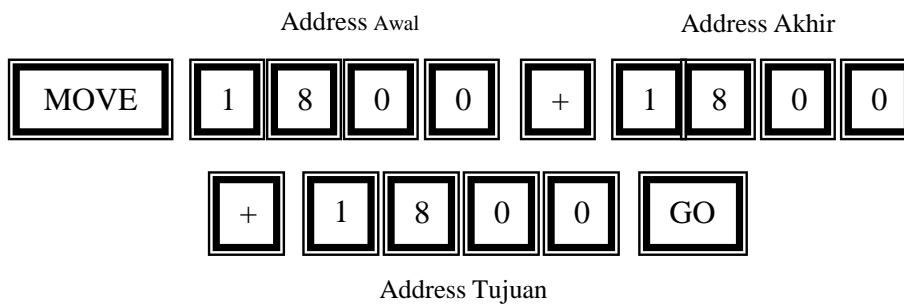
Jika tombol menekan “GO” maka program akan dieksekusi secara menyeluruh sampai selesai, tetapi jika menekan tombol “Step” maka program akan dijalankan secara satu persatu, sebelum kita menekan tombol “Step” lagi program akan tetap diam tidak akan jalan, jadi kita dapat melihat isi registernya.

5.6. Cara kerja program diatas

Register A diisi dengan data 00h kemudian register A ditambah satu, lalu isi dari register A dicopikan ke register B, kemudian register B ditambah satu, lalu isi register B dicopikan ke register C kemudian program berhenti karena perintah RST 38. Arti dari setiap instruksi yang digunakan adalah LD untuk mengcopikan data, kemudian INC adalah menambahkan data yang ada dengan nilai satu (1). RST 38 adalah perintah untuk menghentikan eksekusi program.

6. Menggandakan Suatu Blok Memori

Untuk memindahkan atau menggandakan suatu blok memori yang berisi data dapat dilakukan dengan menggunakan instruksi tombol "MOVE". Misalnya untuk menggandakan program sederhana diatas yang mulai dari Address 1800h sampai dengan 1806h ke Address baru 1900h sampai dengan 1906h dapat dilakukan dengan menekan tombol sebagai berikut :



6.1. Membandingkan isi blok memori Address 1800h sampai dengan 1806 dengan blok memori Address 1900 sampai dengan 1906h dan hasilnya dicatat pada tabel berikut :

Address	Data	Address	Data
1800	3E	1900	3E
1801	00	1901	00

1802	3C	1902	3C
1803	47	1903	47
1804	04	1904	04
1805	48	1905	48
1806	FF	1906	FF

6.2. Program yang ada pada Address 1900h samapi dengan 1906 dieksekusi dengan eksekusi “STEP” dan mencatat isi register untuk setiap instruksi.

Data register	A	B	C
Step 1	00	XX	XX
Step 2	01	XX	XX
Step 3	01	01	XX
Step 4	01	02	XX
Step 5	01	02	02

E. KESIMPULAN

Dari praktikum yang telah dilakukan dapat disimpulkan bahwa :

MPF-1 merupakan suatu trainer khusus sebagai alat bantu mempelajari mikroprosesor Zilog Z-80 CPU.

1. Untuk menghidupkan MPF-1 pertama-tama menyiapkan catu daya (adaptor) dan memeriksa apakah catu daya tersebut bekerja baik dengan keluaran 5 VDC. Soket DC dihubungkan paralel ke konektor daya disebelah kanan atas.
2. Untuk melihat data yang ada pada suatu register, maka pertama-tama menekan tombol REG. Setelah tombol ditekan, peraga tujuh segment akan menampilkan tulisan “REG” yang menyatakan MPF-1 siap untuk menyajikan Register. Kemudian menekan tombol sesuai dengan nama register yang akan dilihat isinya. Setiap tombol mewakili register 16 bit.

3. Untuk melihat data yang ada pada suatu lokasi memori baik ROM/EPROM maupun RAM, caranya adalah dengan menekan tombol **ADDR** dan peraga akan menampilkan pola : **X.X.X.X.** menunjukkan lokasi Address dari memori yang dilihat isinya. Dan **XX** menunjukkan isi pada Address tersebut. Untuk merubah lokasi atau Address memori cukup menekan tombol “ **ADDR**” diikuti dengan menekan tombol heksa-desimal empat kali.
4. Untuk menyisipkan data pada suatu Address memori dapat dilakukan dengan menekan tombol “**INS**” = insert dilanjutkan dengan menekan tombol data heksa-desimal yang disisipkan.
5. Untuk menjalankan atau execute program, dengan cara memasukkan Address awal program yang akan dieksekusi, lalu menekan tombol “**GO**” sebagai perintah mulainya pelaksanaan eksekusi.
6. Untuk memindahkan atau menggandakan suatu blok memori yang berisi data dapat dilakukan dengan menggunakan instruksi tombol “**MOVE**”.