

BAB I

PENDAHULUAN

A. Deskripsi

Modul ini berisi materi latihan memahami Set Instruksi Mikroprosesor Z-80 CPU sebagai perintah-perintah yang digunakan untuk menyusun program bagi peserta didik SMK program keahlian Elektronika Industri dan Teknik Audio Video.

Melalui modul ini anda dapat berlatih memahami perintah transfer data 8 bit, perintah transfer data 16 bit, perintah pertukaran data, perintah pelacakan/search data, perintah aritmetika dan logika, perintah putar dan geser, perintah manipulasi bit, perintah Jump, perintah Call dan Return, perintah Restart, perintah Input Output.

B. Prasyarat

Untuk mempelajari dan melakukan latihan kompetensi dalam modul ini ada dua syarat yaitu :

Syarat Umum :

Anda harus belajar dan berlatih kompetensi dengan rumus “TePUK DisKo” yaitu Teratur, Percaya diri, Ulet, Kreatif, Disiplin dan Konsentrasi.

Syarat Khusus :

Anda harus sudah mempelajari Modul 2 tentang Arsitektur Komputer Mikro MPF-I, Sistem digital, Logika, dan memahami aritmatika Sistem Bilangan Biner, Heksa Desimal

C. Petunjuk Penggunaan Modul

I. Bagi Guru/Fasilitator

- ✓ Baca dan cermati betul deskripsi Silabi
- ✓ Pelajari Kompetensi Dasar, Indikator, Materi Pembelajaran, Kegiatan Pembelajaran, dan Penilaian
- ✓ Pelajari Level Kompetensi Kunci dan Jabarannya
- ✓ Lihat Tujuan Akhir Pembelajaran apakah sudah sesuai dengan Indikator Silabus sebagai tuntutan Kriteria Kinerja deskripsi kompetensi.
- ✓ Cocokkan cakupan kegiatan belajar dengan deskripsi Materi Pembelajaran dan Kegiatan Pembelajaran. Cermati apakah materi kegiatan pembelajaran telah mencakup keseluruhan Kompetensi Dasar dalam aspek Sikap, Pengetahuan, dan Keterampilan.

- ✓ Fasilitasi peserta didik untuk berlatih kompetensi memahami Set Instruksi Mikroprosesor Z-80 CPU sesuai prinsip pembelajaran KBK yaitu berbasis pada siswa, belajar secara terintegrasi, *Individual learning*, *Mastery learning*, *Problem Solving*, *Experience Based Learning*.

2. Bagi Peserta didik

- ✓ Baca dan pahami deskripsi modul dan prasyarat penggunaan modul
- ✓ Baca dan pahami tujuan akhir modul
- ✓ Isikan rencana kegiatan belajar dengan berkonsultasi dengan guru/fasilitator
- ✓ Baca dan laksanakan cek kemampuan dengan berkonsultasi dengan guru/ fasilitator

D. Tujuan Akhir

Menguasai cara kerja seluruh Instruksi Mikroprosesor Zilog Z-80 CPU.

E. KOMPETENSI

BIDANG KEAHLIAN : Teknik Elektronika
 PROGRAM KEAHLIAN : Teknik Elektronika Industri
 STANDAR KOMPETENSI : Memprogram Peralatan Sistem Otomasi Elektronik yang Berkaitan I/O berbantuan : Mikroprosesor dan Mikrokontroler.
 KODE : ELIND1
 JAM PEMBELAJARAN : 45 (90) Jam @ 45 menit

LEVEL KOMPETENSI	A	B	C	D	E	F	G
KUNCI	3	3	3	3	2	3	3

KONDISI KINERJA	Unjuk kerja ketrampilan kognitif namun dengan imajinasi psiko-motorik seperti unit kompetensi ini bisa dicapai dengan kondisi: <ol style="list-style-type: none"> 1. Memiliki kemampuan dasar tentang konsep sistem 2. Memiliki kompetensi dasar elektronika 3. Memiliki kemampuan mengenai petunjuk keselamatan kerja secara umum 4. Memiliki kemampuan menulis laporan kerja yang baik.
-----------------	---

KOMPETENSI DASAR	INDIKATOR	MATERI PEMBELAJARAN	KEGIATAN PEMBELAJARAN	PENILAIAN	ALOKASI WAKTU			SUMBER BELAJAR
					TM	PS	PI	
1. Menguasai Prosedur Penyusunan Algoritma Pemrograman	1.1. Ditunjukkan kaidah-kaidah yang berlaku dalam penyusunan algoritma dan pemrograman.	Kaidah-kaidah penyusunan algoritma	<ul style="list-style-type: none"> ▪ Memahami pengertian Program Komputer ▪ Menguraikan empat langkah Pengembangan Program Komputer ▪ Berlatih mendefinisikan Permasalahan Program ▪ Berlatih mengembangkan Algoritma Program ▪ Berlatih memilih bentuk baku Flowchart ▪ Berlatih menyusun Flowchart 	<ul style="list-style-type: none"> • Tes Lisan • Tes Tulis 	10	10 (20)		<i>Microprocessor and Interfacing Programming and Hardware</i> MCGraw-Hill, 1992

KOMPETENSI DASAR	INDIKATOR	MATERI PEMBELAJARAN	KEGIATAN PEMBELAJARAN	PENILAIAN	ALOKASI WAKTU			SUMBER BELAJAR
					TM	PS	PI	
	1.2. Diujikan algoritma dan pemrograman yang telah dikuasai dengan melakukan beberapa kasus.	Pengujian algoritma dan pemrograman	<ul style="list-style-type: none"> ▪ Menguji algoritma dan pemrograman ▪ Menggunakan algoritma dalam pemrograman ▪ Menguji studi kasus penggunaan algoritma dalam pemrograman ▪ Memahami Bahasa Pemrograman Komputer 	<ul style="list-style-type: none"> • Tes Lisan • Tes Tulis 	10	5 (10)	10 (40)	<i>Microprocessor and Interfacing Programming and Hardware</i> MCGraw-Hill, 1992

LEVEL KOMPETENSI KUNCI

No	Kompetensi Kunci	LEVEL 1	LEVEL 2	LEVEL 3
A	MENGUMPULKAN, MENGANALISA DAN MENGELOLA INFORMASI Kapasitas untuk mengumpulkan informasi, memindahkan dan menyeleksi informasi dalam rangka memilih informasi yang diperlukan untuk dipresentasikan, mengevaluasi sumber dan cara memperoleh informasi tersebut	Mengakses dan menyimpan dari satu sumber	Mengakses, memilih dan menyimpan dari beberapa sumber	<i>Mengakses, mengevaluasi dan mengatur dari berbagai macam sumber</i>
B	MENKOMUNIKASIKAN IDE-IDE DAN INFORMASI Kapasitas untuk berkomunikasi dengan orang lain secara efektif menggunakan beragam bahasa, tulisan, grafik dan ekspresi non verbal lainnya	Sederhana dengan aturan yang telah dikenal	Komplek dengan isi tertentu	<i>Komplek dengan isi beragam</i>
C	MERENCANAKAN DAN MENGORGANISIR KEGIATAN Kapasitas untuk merencanakan dan mengatur kegiatan kerja individu termasuk penggunaan waktu dan sumber yang baik, pemilihan prioritas dan pengawasan prestasi individu	Di bawah pengawasan	Dengan bimbingan	<i>Inisiatif sendiri dan mengevaluasi kegiatan yang kompleks</i>

No	Kompetensi Kunci	LEVEL 1	LEVEL 2	LEVEL 3
D	BEKERJA DENGAN ORANG LAIN SERTA KELOMPOK DALAM SATU TIM Kapasitas untuk berhubungan secara efektif dengan orang lain baik antar pribadi ataupun kelompok termasuk mengerti dan memberikan respon akan keinginan klien dan bekerja secara efektif sebagai anggota kelompok untuk mencapai tujuan bersama.	Aktifitas yang telah diketahui	Membantu merancang dan mencapai tujuan	<i>Kolaborasi dalam kegiatan kelompok</i>
E	MENGGUNAKAN IDE-IDE SERTA TEKNIK DALAM MATEMATIKA Kapasitas untuk menggunakan konsep bilangan, spasi dan ukuran dan teknik seperti perkiraan untuk praktek	Tugas yang sederhana	<i>Memilih tugas yang kompleks dan sesuai</i>	Evaluasi dan mengadaptasi sebagai tugas yang sesuai
F	MENYELESAIKAN MASALAH Kapasitas untuk menjalankan strategi penyelesaian masalah baik untuk situasi di mana masalah dan solusi yang diinginkan memiliki bukti dan dalam situasi yang memerlukan pemikiran kritis dan pendekatan kreatif untuk mencapai hasil	Rutin, sedikit pengawasan eksplorasi – pengawasan melekat	Rutin, mandiri eksplorasi- dengan bimbingan	<i>Masalah yang kompleks, pelaksanaan pendekatan sistematis, menjelaskan proses</i>
G	MENGGUNAKAN TEKNOLOGI Kapasitas untuk menerangkan teknologi, mengkombinasikan keahlian fisik dan sensor yang diperlukan untuk menjalankan peralatan dengan pengertian ilmiah dan prinsip teknologi yang diperlukan untuk mengadaptasi sistem	Produksi ulang atau melaksanakan produk dasar atau jasa	Menyusun, mengatur atau mengoperasikan produk atau jasa	<i>Mendesain atau merangkai produk atau jasa</i>

F. Cek Kemampuan

Berilah tanda thick (✓) pada kolom YA atau TIDAK sesuai dengan pernyataan berikut.

Mintalah Catatan dan Tanda Tangan Guru/Pembimbing.

NO	PERNYATAAN	YA	TIDAK	CATATAN GURU/PEMBIMBING	TANDA TANGAN GURU/PEMBIMBING
01	Apakah saudara sudah mempelajari set instruksi mikroprosesor Z-80 CPU				
02	Apakah saudara dapat mengklasifikasikan sebelas jenis instruksi pada mikroprosesor Z-80 CPU				
03	Apakah saudara dapat menggunakan seluruh instruksi pada mikroprosesor Z-80 CPU				

BAB II PEMBELAJARAN

A. Rencana Belajar Peserta didik

Rencana belajar peserta didik diisi oleh peserta didik dan disetujui oleh Guru. Rencana

belajar tersebut adalah sebagai berikut :

NAMA PESERTA DIDIK :

JENIS KEGIATAN	TANGGAL	WAKTU	TEMPAT BELAJAR	ALASAN PERUBAHAN	TANDA TANGAN GURU

B. Kegiatan Belajar

1. Pendahuluan

Instruksi pada Z-80 CPU dapat digolongkan menjadi 11 kelompok:

- 1) Instruksi Transfer Data 8 bit
- 2) Instruksi Transfer Data 16 bit
- 3) Instruksi Pertukaran Data
- 4) Instruksi Pelacakan/Search Data
- 5) Instruksi Aritmetika dan Logika
- 6) Instruksi Putar dan Geser
- 7) Instruksi Manipulasi Bit
- 8) Instruksi Jump
- 9) Instruksi Call dan Return
- 10) Instruksi RESTART
- 11) Instruksi Input dan Output

2. Kegiatan Belajar I: Memahami Instruksi Transfer Data

Operasi transfer data lebih tepat disebut sebagai operasi copy data. Pada mikroprosesor Z-80 CPU sebagian besar dijalankan menggunakan perintah LD singkatan dari Load. Z-80 CPU memiliki 134 perintah Load.

Disamping juga ada perintah 6 jenis perintah EX, EXX singkatan dari EXCHANGE. Z-80 CPU juga memiliki 12 jenis perintah PUSH, dan POP, dapat digunakan untuk transfer data dalam operasi stack pointer.

Data dapat ditransfer dalam 8 bit atau 16 bit. Perintah transfer data memuat dua operand yaitu operand pertama menunjukkan Lokasi dimana data akan disimpan, apakah dalam register atau di memori. Operand pertama ini disebut **Destinasi**. Operand yang kedua menunjukkan lokasi asli atau asal sebuah data. Operand kedua ini disebut **Source**. Operand dapat berupa register, memori, atau data immediate. Lebar data yang ditransfer dapat berupa data 8 bit atau data 16 bit.

Bentuk umum transfer data pada Z-80 CPU adalah :

LD (operand I : destinasi), (operand II : Source)

Sebagai contoh : LD A, B menunjukkan perintah untuk meng-copy data yang ada di Register B ke Register A. Jadi Register A disebut destinasi atau tujuan dan Register B disebut Source atau asal/sumber.

a. Transfer Data 8 Bit

Transfer data 8 bit dapat terjadi diantara :

- 1) Register Ke Register
- 2) Memori Ke Register
- 3) Data Immediate Ke Register
- 4) Register Ke Memori
- 5) Memori Ke Memori
- 6) Data Immediate Ke Memori

1). Transfer data 8 bit dari Register ke Register

Dapat terjadi diantara register 8 bit yaitu register , A,B,C,D,E,H,L,dan I

Contoh:

No	Assembly	Simbol Operasi	Keterangan
1.	LD A,B	$A \leftarrow B$	muati register A dengan data dari register B
2.	LD B,C	$B \leftarrow C$	muati register B dengan data dari register C
3.	LD B,A	$B \leftarrow A$	muati register B dengan data dari register A
4.	LD B, E	$B \leftarrow E$	muati register B dengan data dari register E

2). Transfer data 8 bit dari Memori ke Register

Transfer data dari memori ke register mencakup persyaratan bahwa harus ada cara atau mekanisme pemegangan alamat memori. Dalam Z-80 CPU alamat memori ada dua byte atau 16 bit. Pemegang alamat memori menggunakan salah satu register 16 bit.

Transfer data dari memori dapat terjadi dari lokasi EPROM atau dari lokasi RWM karena kedua memori ini memiliki sifat baca. Untuk operasi ini ada tanda “ () ” sebagai tanda operasi memori.

Contoh:

No	Assembly	Simbol Operasi	Keterangan
1.	LD A, (1902)	$A \leftarrow (1902)$	muati register A dengan data dari memori lokasi alamat 1902 (RWM)
2.	LD A, (0066)	$A \leftarrow (0066)$	muati register A dengan data dari memori lokasi alamat 0066 (ROM)
3.	LD B, (HL)	$B \leftarrow (HL)$	muati register B dengan data dari memori lokasi alamat sama dengan isi register HL
4.	LD D, (IX+02)	$D \leftarrow (IX+02)$	muati register D dengan data dari memori lokasi alamat sama dengan isi register IX+02

3). Transfer data Immediate 8 bit ke Register

Transfer data immediate 8 bit ke register dapat terjadi terhadap register A, B, C,D,E,H, dan L.

Contoh:

No	Assembly	Smbol Operasi	Keterangan
1.	LD A, 19	A ← 19h	muati register A dengan data 19h
2.	LD A, 00	A ← 00h	muati register A dengan data 00h
3.	LD B, 3F	B ← 3Fh	muati register B dengan data 3Fh
4.	LD C, FF	C ← FFh	muati register C dengan data FFh

4). Transfer data 8 bit dari Register ke Memori

Transfer data dari register ke memori mencakup persyaratan bahwa harus ada cara atau mekanisme pemegangan alamat memori. Dalam Z-80 CPU alamat memori ada dua byte atau 16 bit.

Transfer data dari register ke memori dapat terjadi hanya ke lokasi RWM karena ROM tidak bisa diisi data baru. Untuk operasi ini ada tanda “() “ sebagai tanda operasi memori menggunakan salah satu register 16 bit atau angka alamat.

Contoh:

No	Assembly	Simbol Operasi	Keterangan
1.	LD(1902), A	(1902) ← A	muati memori lokasi alamat 1902 (RWM) dengan data dari register A
2.	LD (HL), B	(HL) ← B	muati memori lokasi alamat sama dengan isi register HL dengan data dari register B
3.	LD (IX+02), D	(IX+02) ← D	muati memori lokasi alamat sama dengan isi register IX+ 02 dengan data dari register D

5). Transfer data 8 bit dari Memori ke Memori

Transfer data dari memori ke memori mencakup persyaratan bahwa harus ada cara atau mekanisme pemegangan alamat memori. Dalam Z-80 CPU alamat memori ada dua byte atau 16 bit.

Transfer data dari memori ke memori dapat terjadi hanya ke lokasi RWM karena ROM tidak bisa diisi data baru. Untuk operasi ini ada tanda “() “ sebagai tanda operasi memori.

Contoh:

No	Assembly	Operasi	Keterangan
1.	LDI	(DE) ← (HL) DE ← DE+1 HL ← HL+1 BC ← BC-1	Transfer 1 byte data dari lokasi memori yang alamatnya dicatat oleh HL ke lokasi memori yang alamatnya dicatat oleh DE
2.	LDIR	(DE) ← (HL) DE ← DE+1 HL ← HL+1 BC ← BC-1 Diulang sampai reg. BC = 0000	Transfer 1 byte data dari lokasi memori yang alamatnya dicatat oleh HL ke lokasi memori yang alamatnya dicatat oleh DE , Diulang sampai isi reg BC sama dengan nol (alamat naik)
3.	LDD	(DE) ← (HL) DE ← DE-1 HL ← HL-1 BC ← BC-1	Transfer 1 byte data dari lokasi memori yang alamatnya dicatat oleh HL ke lokasi memori yang alamatnya dicatat oleh DE
4.	LDDR	(DE) ← (HL) DE ← DE-1 HL ← HL-1 BC ← BC-1 Diulang sampai reg. BC = 0000	Transfer 1 byte data dari lokasi memori yang alamatnya dicatat oleh HL ke lokasi memori yang alamatnya dicatat oleh DE , Diulang sampai isi reg BC sama dengan nol (alamat turun)

6). Transfer data Immediate 8 bit ke Memori

Contoh:

No	Assembly	Operasi	Keterangan
1.	LD (HL), FF	(HL) ← FF	muati memori lokasi alamat sama dengan isi register HL dengan data FFh
2	LD (IX+02), 64	(IX+02) ← 64	muati memori lokasi alamat sama dengan isi register IX+ 02 dengan data 64h
3	LD (IY+02), 19	(IY+02) ← 19	muati memori lokasi alamat sama dengan isi register IY+ 02 dengan data 19h

7). Latihan Kasus

Berikut ini ada duabelas perintah transfer data. Identifikasi untuk setiap perintah tergolong kategori transfer data 8 bit yang mana diantara enam kategori (register ke register, memori ke register, data immediate ke register, register ke memori, memori ke memori, atau data immediate ke memori). Kemudian nyatakan hasil dari masing-masing perintah tersebut.

Ulangi sekali lagi sampai saudara memahami cara menentukan kategori jenis perintah transfer data 8 bit dan hasil dari masing-masing perintah tersebut.

No.	ADDRESS	Kode operasi	Assembly	Simbol Operasi
1.	1800	3E 93	LD A , 93H	A ← 93H
2.	1802	47	LD B , A	B ← A
3.	1803	48	LD C , B	C ← B
4.	1804	51	LD D , C	D ← C
5.	1805	5A	LDE , D	E ← D
6.	1806	26 19	LD H , 19H	H ← 19H
7.	1808	2E 90	LD L , 90H	L ← 90H
8.	180A	36 64	LD (HL) , 64H	(HL) ← 64H
9.	180C	46	LD B , (HL)	B ← (HL)
10.	180D	3A 90 19	LD A , (1990)	A ← (1990)
11.	1810	32 91 19	LD (1991) , A	(1991) ← A
12.	1813	FF	RST 38	STOP

8). Isikan hasil identifikasi transfer data pada tabel berikut. Gunakan kolom simbol operasi untuk merumuskan hasilnya.

No.	Assembly	Kategori	Hasil
1.	LD A , 93H
2.	LD B , A
3.	LD C , B
4.	LD D , C
5.	LDE , D
6.	LD H , 19H
7.	LD L , 90H
8.	LD (HL) , 64H
9.	LD B , (HL)
10.	LD A , (1990)
11.	LD (1991) , A
12.	RST 38

9). Eksekusi program di atas dengan perintah “GO” lalu baca isi register dan memori seperti tabel berikut :

Reg./Memori	A	F	B	C	D	E	H	L	1990	1991
Data										

10). Cocokkan hasil identifikasi saudara pada langkah 8) dengan hasil eksekusi langkah 9). Apakah ada perbedaan atau sama nilai akhirnya. Jika berbeda mengapa hal itu bisa terjadi. Seharusnya hasil analisis saudara dengan hasil eksekusi adalah sama.

k). Uraian penjelasan kasus 10)

.....
.....
.....
.....
.....
.....
.....
.....

l). Ulangi kaji sekali lagi kasus 7) tersebut. Jika saudara sudah memahami dengan baik silahkan meneruskan ke sub kegiatan belajar transfer data 16 bit berikut ini.

Kunci Jawaban :

Kategori dan hasil transfer data

No.	Assembly	Kategori	Hasil
1.	LD A , 93H	data immediate ke register	A = 93
2.	LD B , A	register ke register	B = 93
3.	LD C , B	register ke register	C = 93
4.	LD D , C	register ke register	D = 93.
5.	LD E , D	register ke register	E = 93
6.	LD H , 19H	data immediate ke register	H = 19
7.	LD L , 90H	data immediate ke register	L = 90 ; HL =1990
8.	LD (HL) , 64H	data immediate ke memori	(1990) = 64
9.	LD B , (HL)	memori ke register	B = 64
10.	LD A , (1990)	memori ke register	A = 64
11.	LD (1991) , A	register ke memori	(1991) = 64
12.	RST 38		

9). Hasil akhir Eksekusi program dengan perintah "GO"

Reg./Memori	A	F	B	C	D	E	H	L	1990	1991
Data	64	XX	64	93	93	93	19	90	64	64

b. Transfer Data 16 Bit

Transfer data 16 bit dapat terjadi diantara :

- 1) Register ke Register,
- 2) Register ke Memori,
- 3) Memori ke Register,
- 4) Data Immediate ke Register
- 5) Memori ke Memori

Pada transfer data 16 bit, transfer data immediate ke memori tidak ada sehingga transfer data immediate ke memori hanya dapat dilakukan dengan cara 8 bit.

1). Transfer data 16 bit dari Register ke Register

Dapat terjadi diantara register 16 bit yaitu register SP, HL, IX, dan IY. Perintah yang digunakan adalah LD = Load

Contoh:

No	Assembly	Operasi	Keterangan
1.	LD SP,HL	SP ← HL	muati register SP dengan data dari register HL
2	LD SP, IX	SP ← IX	muati register SP dengan data dari register IX
3	LD SP, IY	SP ← IY	muati register SP dengan data dari register IY

2). Transfer data 16 bit dari Register ke Memori

Transfer data dari register ke memori mencakup persyaratan bahwa harus ada cara atau mekanisme pemegangan alamat memori. Dalam Z-80 CPU alamat memori ada dua byte atau 16 bit. Transfer data dari register ke memori dapat terjadi hanya ke lokasi RWM karena ROM tidak bisa diisi data baru. Untuk operasi ini ada tanda “ () “ sebagai tanda operasi memori menggunakan salah satu register 16 bit atau angka alamat. Transfer data 16 bit dari register ke memori dapat terjadi dari register BC, DE, HL, IX, IY, dan AF. Perintah yang digunakan adalah LD = Load dan PUSH

Contoh:

No	Assembly	Operasi	Keterangan
1.	LD(1902), BC	(1903) ← B (1902) ← C	Muati memori lokasi alamat 1902 (RWM) dengan data dari register C dan memori alamat 1903 dengan data register B
2.	LD (1800), HL	(1801) ← H (1800) ← L	Muati memori lokasi alamat 1800 dengan isi register L dan alamat 1801 dengan data dari register H
3.	PUSH IX	(SP-1) ← IX _H (SP-2) ← IX _L SP ← SP-2	Muati memori lokasi alamat sama dengan isi register SP-1 dengan data dari register IX _H dan SP-2 dengan data dari register IX _L

3). Transfer data 16 bit dari Memori ke Register

Transfer data dari memori ke register mencakup persyaratan bahwa harus ada cara atau mekanisme pemegangan alamat memori. Dalam Z-80 CPU alamat memori ada dua byte atau 16 bit. Pemegang alamat memori menggunakan salah satu register 16 bit.

Transfer data dari memori dapat terjadi dari lokasi EPROM atau dari lokasi RWM karena kedua memori ini memiliki sifat baca. Untuk operasi ini ada tanda “()” sebagai tanda operasi memori. Transfer data 16 bit dari memori ke register dapat terjadi terhadap register IX, IY, BC, DE, HL, SP, dan AF. Perintah yang digunakan adalah LD dan POP.

Contoh:

No	Assembly	Simbol Operasi	Keterangan
1.	LD IX,(1902)	$IX_H \leftarrow (1903)$ $IX_L \leftarrow (1902)$	muati register IX dengan data dari memori lokasi alamat 1903 dan 1902
2.	LD IY,(0066)	$IY_H \leftarrow (0067)$ $IY_L \leftarrow (0066)$	muati register IY dengan data dari memori lokasi alamat 0067 dan 0066
3.	LD BC,(1800)	$B \leftarrow (1801)$ $C \leftarrow (1800)$	muati register BC dengan data dari memori lokasi alamat 1801 dan 1800
4.	POP DE	$D \leftarrow (SP+1)$ $E \leftarrow (SP)$ $SP \leftarrow SP+2$	muati register DE dengan data dari memori lokasi alamat sama dengan isi register SP+1 dan SP,

4). Transfer data Immediate 16 bit ke Register

Transfer data immediate 16 bit ke register dapat terjadi terhadap register BC, DE, HL, SP, IX, dan IY. Perintah yang digunakan adalah LD = Load

Contoh:

No	Assembly	Operasi	Keterangan
1.	LD BC,1900	$BC \leftarrow 1900H$	muati register BC dengan data 1900h
2.	LD DE,1800	$DE \leftarrow 1800H$	muati register DE dengan data 1800h
3.	LD IX, 203F	$IX \leftarrow 203FH$	muati register IX dengan data 203Fh
4.	LD IY, EEF	$IY \leftarrow EEFH$	muati register IY dengan data EEFh

5). Transfer data 16 bit dari Memori ke Memori

Transfer data dari register ke memori mencakup persyaratan bahwa harus ada cara atau mekanisme pemegangan alamat memori. Dalam Z-80 CPU alamat memori ada dua byte atau 16 bit.

Transfer data dari register ke memori dapat terjadi hanya ke lokasi RWM karena ROM tidak bisa diisi data baru. Untuk operasi ini ada tanda “()” sebagai tanda operasi memori. Perintah yang dapat digunakan hanya LDIR dan LDDR. Dalam hal ini jumlah

byte data yang dapat ditransfer satu byte , dua byte atau lebih dengan kemampuan maksimum 64 K byte bergantung isi register BC

Contoh:

No	Assembly	Simbol Operasi	Keterangan
1.	LDIR	(DE) ← (HL) DE ← DE+1 HL ← HL+1 BC ← BC-1 Diulang sampai reg. BC = 0000	Transfer 1 byte data dari lokasi memori yang alamatnya dicatat oleh HL ke lokasi memori yang alamatnya dicatat oleh DE , Diulang sampai isi reg BC sama dengan nol (alamat naik)
2.	LDDR	(DE) ← (HL) DE ← DE-1 HL ← HL-1 BC ← BC-1 Diulang sampai reg. BC = 0000	Transfer 1 byte data dari lokasi memori yang alamatnya dicatat oleh HL ke lokasi memori yang alamatnya dicatat oleh DE , Diulang sampai isi reg BC sama dengan nol (alamat turun)

6). Latihan Kasus

Berikut ini ada dua puluh dua perintah transfer data. Identifikasi untuk setiap perintah tergolong kategori transfer data 16 bit yang mana diantara lima kategori (register ke register, register ke memori, memori ke register, data immediate ke register, atau memori ke memori). Kemudian nyatakan hasil dari masing-masing perintah tersebut.

No.	ADDRESS	Kode operasi	Assembly	Simbol Operasi
1.	1820	01 63 19	LD BC , 1963H	BC ← 1963H
2.	1823	11 64 19	LD DE , 1964H	DE ← 1964H
3.	1826	21 95 19	LD HL , 1995H	HL ← 1995H
4.	1829	31 91 19	LD SP , 1991H	SP ← 1991H
5.	182C	ED 43 93 19	LD (1993) , BC	(1993) ← C
6.				(1994) ← B
7.	1830	ED 53 95 19	LD (1995) , DE	(1995) ← E
8.				(1996) ← D
9.	1834	DD 21 00 18	LD IX , 1800H	IX ← 1800H
10.	1838	FD 21 13 18	LD IY , 1813H	IY ← 1813H
11.	183C	DD E5	PUSH IX	(SP-2) ← IXL
12.				(SP-1) ← IXH
13.				SP ← SP - 2
14.	183E	FD E5	PUSH IY	(SP-2) ← IYL
15.				(SP-1) ← IYH
16.				SP ← SP-2
17.	1840	FD E1	POP IY	IYH ← (SP+1)
18.				IXL ← (SP)
19.				SP ← SP+2
20.	1842	DD E1	POP IX	IYH ← (SP+1)
21.				IYL ← (SP)
22.				SP ← SP+2
23.	1844	FF	RST 38	STOP

7). Isikan hasil identifikasi transfer data pada tabel berikut. Gunakan kolom simbol operasi untuk merumuskan hasilnya.

No.	Assembly	Kategori	Hasil
1.	LD BC ,1963H
2.	LD DE ,1964H
3.	LD HL ,1995H
4.	LD SP ,1991H
5.	LD (1993) , BC
6.		
7.	LD (1995) , DE
8.		
9.	LD IX ,1800H
10.	LD IY ,1813H
11.	PUSH IX
12.		
13.		
14.	PUSH IY
15.		
16.		
17.	POP IY
18.		
19.		
20.	POP IX
21.		
22.		
23.	RST 38

8). Eksekusi program di atas dengan perintah “GO” lalu baca isi register dan memori seperti tabel berikut :

Reg./Memori	BC	DE	HL	SP	IX	IY	198D	198E	198F	1990
Data										

9). Cocokkan hasil identifikasi saudara pada langkah 7) dengan hasil eksekusi langkah 8). Apakah ada perbedaan atau sama nilai akhirnya. Jika berbeda mengapa hal itu bisa terjadi. Seharusnya hasil analisis saudara dengan hasil eksekusi adalah sama.

10). Uraian penjelasan kasus 6)

.....

.....

.....

.....

.....

.....

.....

.....

11). Ulangi kaji sekali lagi kasus 6) tersebut. Jika saudara sudah memahami dengan baik silahkan meneruskan ke sub kegiatan belajar Pertukaran Data berikut ini.

Kunci Jawaban:

No.	Assembly	Kategori	Hasil
1.	LD BC , 1963h	data immediate ke register	BC = 1963
2.	LD DE , 1964h	data immediate ke register	DE = 1964
3.	LD HL , 1995h	data immediate ke register	HL = 1995
4.	LD SP , 1991h	data immediate ke register	SP = 1991
5.	LD (1993) , BC	register ke memori	(1993) = 63
6.			(1994) = 19
7.	LD (1995) , DE	register ke memori	(1995) = 64
8.			(1996) = 19
9.	LD IX , 1800H	data immediate ke register	IX = 1800
10.	LD IY , 1813H	data immediate ke register	IY = 1813
11.	PUSH IX	register ke memori	(198F) = 00
12.			(1990) = 18
13.			SP = 198F
14.	PUSH IY	register ke memori	(198D) = 13
15.			(199E) = 18
16.			SP = 198D
17.	POP IY	memori ke register	IY = 1813
18.			SP = 198F
19.			
20.	POP IX	memori ke register	IX = 1800
21.			SP = 1991
22.			
23.	RST 38		

8). Eksekusi program di atas dengan perintah "GO" lalu baca isi register dan memori seperti tabel berikut :

Reg./Memori	BC	DE	HL	SP	IX	IY	198D	198E	198F	1990
Data	1963	1964	1995	1991	1800	1813	13	18	00	18

c. Pertukaran Data

Pertukaran data dapat dilakukan diantara dua register, kelompok pasangan register dan antara register dengan memori. Instruksi yang digunakan adalah EX dan EXX. Pertukaran data dapat terjadi diantara register DE, HL, BC, BC', DE', HL'.

Contoh

Assembly	Simbol Operasi	Jenis Transfer Data
EX DE, HL	DE \leftrightarrow HL	Register \leftrightarrow Register
EX AF, AF'	AF \leftrightarrow AF'	Register \leftrightarrow Register
EXX	BC \leftrightarrow BC'	
	DE \leftrightarrow DE'	
	HL \leftrightarrow HL'	
EX (SP), HL	H \leftrightarrow (SP+1)	Register \leftrightarrow Memori
	L \leftrightarrow (SP)	
EX (SP), IX	IXH \leftrightarrow (SP+1)	
	IXL \leftrightarrow (SP)	

1). Latihan Kasus

Berikut ini ada enambelas perintah campuran antara perintah transfer data dan perintah pertukaran data. Identifikasi untuk setiap perintah tergolong kategori transfer data 8 bit yang mana diantara enam kategori (register ke register, memori ke register, data immediate ke register, register ke memori, memori ke memori, atau data immediate ke memori). Kategori transfer data 16 bit yang mana diantara lima kategori (register ke register, memori ke register, data immediate ke register, register ke memori, atau memori ke memori). Atau kategori pertukaran data. Kemudian nyatakan hasil dari masing-masing perintah tersebut.

Ulangi sekali lagi sampai saudara memahami cara menentukan kategori jenis perintah transfer data 8 bit dan hasil dari masing-masing perintah tersebut.

Analisis program berikut dan tentukan hasil untuk setiap perintah !!

No.	ADDRESS	Kode operasi	Assembly	Simbol Operasi
1.	1800	11 01 01	LD DE , 0101H	DE ← 0101H
2.	1803	21 FF FF	LD HL , FFFFH	HL ← FFFFH
3.	1806	EB	EX DE , HL	HL ← DE
4.	1807	01 02 02	LD BC , 0202H	BC ← 0202H
5.	180A	11 03 03	LD DE , 0303H	DE ← 0303H
6.	180D	78	LD A , B	A ← B
7.	180E	08	EX AF , AF'	AF ↔ AF'
8.	180F	31 90 19	LD SP , 1990H	SP ← 1990H
9.	1812	E3	EX (SP) , HL	H ↔ (SP+1)
10.				L ↔ (SP)
11.	1813	DD 21 22 00	LD IX , 0022H	IX ← 0022H
12.	1817	31 92 19	LD SP , 1992H	SP ← 1992H
13.	181A	DD E3	EX (SP) , IX	IXH ↔ (SP+1)
14.				IXL ↔ (SP)
15.	181C	FF	RST 38	STOP
16.				

2). Isikan hasil identifikasi perintah kasus 1) pada tabel berikut. Gunakan kolom simbol operasi untuk merumuskan hasilnya.

No.	Assembly	Kategori	Hasil
1.	LD DE , 0101H
2.	LD HL , FFFFH
3.	EX DE , HL
4.	LD BC , 0202H
5.	LD DE , 0303H
6.	LD A , B
7.	EX AF , AF'
8.	LD SP , 1990H
9.	EX (SP) , HL
10.	
11.	LD IX , 0022H
12.	LD SP , 1992H
13.	EX (SP) , IX
14.	
15.	RST 38
16.	

3). Dengan perintah eksekusi STEP pelajari program latihan kasus 1) step demi step. Cocokkan hasil identifikasi saudara dengan hasil eksekusi

No.	Assembly	Hasil Identifikasi	Hasil Eksekusi STEP
1.	LD DE , 0101H
2.	LD HL , FFFFH
3.	EX DE , HL
4.	LD BC , 0202H
5.	LD DE , 0303H
6.	LD A , B
7.	EX AF , AF'
8.	LD SP , 1990H
9.	EX (SP) , HL
10.	
11.	LD IX , 0022H
12.	LD SP , 1992H
13.	EX (SP) , IX
14.	
15	RST 38
16	

4). Apakah hasil identifikasi saudara sama dengan hasil eksekusi. Jika ya apa kesimpulan saudara. Jika tidak sama jelaskan apa sebabnya.

.....

.....

.....

.....

.....

.....

.....

.....

5). Ulangi kaji sekali lagi kasus 1) tersebut. Jika saudara sudah memahami dengan baik silahkan meneruskan ke sub kegiatan belajar Pelacakan Data berikut ini.

Kunci Jawaban

No.	Assembly	Kategori	Hasil
1.	LD DE , 0101H	data immediate 16 bit ke register	DE = 0101
2.	LD HL , FFFFH	data immediate 16 bit ke register	HL = FFFF
3.	EX DE , HL	pertukaran data	DE=FFFF ; HL=0101
4.	LD BC , 0202H	data immediate 16 bit ke register	BC = 0202
5.	LD DE , 0303H	data immediate 16 bit ke register	DE = 0303.
6.	LD A , B	register ke register 8 bit	A = 02
7.	EX AF , AF'	pertukaran data register -register	AF = XXXX;
8.			AF' = 02XX
9.	LD SP , 1990H	data immediate 16 bit ke register	SP = 1990
10.	EX (SP) , HL	pertukaran data memori-register	HL = XXXX
11.			(1990) = FF; (1991)=FF
12.	LD IX , 0022H	data immediate 16 bit ke register	IX = 0022
13.	LD SP , 1992H	data immediate 16 bit ke register	SP = 1992
14.	EX (SP) , IX	pertukaran data memori-register	IX = FFFF
15.			(1990) = 00; (1991)=22
16.	RST 38	BERHENTI	

No.	Assembly	Hasil Identifikasi	Hasil Eksekusi STEP
1.	LD DE , 0101H	DE = 0101
2.	LD HL , FFFFH	HL = FFFF
3.	EX DE , HL	DE=FFFF ; HL=0101
4.	LD BC , 0202H	BC = 0202
5.	LD DE , 0303H	DE = 0303.
6.	LD A , B	A = 02
7.	EX AF , AF'	AF = XXXX;
8.		AF' = 02XX
9.	LD SP , 1990H	SP = 1990
10.	EX (SP) , HL	HL = XXXX
11.		(1990) = FF; (1991)=FF
12.	LD IX , 0022H	IX = 0022
13.	LD SP , 1992H	SP = 1992
14.	EX (SP) , IX	IX = FFFF
15.		(1990) = 00; (1991)=22
16.	RST 38		

d. Pelacakan Data

Pelacakan atau searching data sangat diperlukan dalam pengembangan program untuk mengetahui nilai sebuah data pada suatu lokasi memori atau menemukan ada tidaknya sebuah nilai dari sekelompok data dapat ditempuh dengan melakukan searching. Perintah search yang digunakan adalah Compare Increment (CPI), Compare Increment Repeat (CPIR), Compare Decrement (CPD), dan Compare Decrement Repeat (CPDR).

Contoh:

Assembly	Simbol Operasi	Keterangan
CPI	A ← (HL) HL ← HL + 1 BC ← BC - 1	Bandingkan isi A dengan data di memori lokasi alamat dicatat register HL. Register HL bertambah satu, register BC berkurang satu.
CPIR	A ← (HL) HL ← HL + 1 BC ← BC - 1 Repeat until A = (HL) or BC=0	Bandingkan isi A dengan data di memori lokasi alamat dicatat HL. Berhenti sampai nilai A=(HL) atau BC = 0. Register HL bertambah satu, register BC berkurang satu.
CPD	A ← (HL) HL ← HL - 1 BC ← BC - 1	Bandingkan isi A dengan data di memori lokasi alamat dicatat HL. Register HL berkurang satu, register BC berkurang satu.
CPDR	A ← (HL) HL ← HL - 1 BC ← BC - 1 Repeat until A = (HL) or BC=0	Bandingkan isi A dengan data di memori lokasi alamat dicatat HL. Berhenti sampai nilai A=(HL) atau BC = 0. Register HL berkurang satu, register BC berkurang satu.

Pada perintah CPI dan perintah CPD isi register A dibandingkan dengan data pada memori yang alamatnya dicatat oleh register HL. Perbandingan ini akan menghasilkan dua kemungkinan yaitu data pada register A sama dengan data pada memori yang alamatnya dicatat oleh register HL atau data pada register A tidak sama dengan data pada memori yang alamatnya dicatat oleh register HL. Kesamaan atau ketidaksamaan dicatat statusnya pada sebuah bit Zero pada register F. Flag Z=0 menunjukkan data pada register A tidak sama dengan data pada memori yang alamatnya dicatat oleh register HL. Flag Z=1 menunjukkan data pada register A sama dengan data pada memori yang alamatnya dicatat oleh register HL.

Dalam proses pelacakan data ini CPI bekerja melacak data menuju memori alamat lebih tinggi sedangkan CPD bekerja melacak data menuju memori alamat lebih rendah.

Register BC digunakan sebagai pembatas jumlah data di memori yang akan dilacak. Karena register BC bernilai 16 bit maka pelacakan data dapat dilakukan diseluruh luasan memori sebanyak 64 K byte lokasi.

Perintah CPIR dan CPDR bekerja secara otomatis menemukan sebuah data yang bernilai sama dengan data yang ada di register A. Proses pelacakan akan berhenti jika telah ditemukan sebuah data di memori yang bernilai sama dengan data yang ada di register A. Alamat dimana data itu berada dicatat oleh register HL. Pelacakan data otomatis ini akan berlangsung terus sampai ditemukan A = (HL) atau nilai register BC=0000h.

1). Kasus :

- a). Jelaskan perbedaan CPI dengan CPD
- b). Jelaskan perbedaan CPIR dengan CPDR
- c). Jelaskan perbedaan CPI dengan CPIR
- d). Jelaskan perbedaan CPD dengan CPDR

2). Jawaban

- a).
.....
.....
.....
- b).
.....
.....
.....
- c).
.....
.....
.....
- d).
.....
.....
.....

Kunci Jawaban:

- a). Perbedaan : CPI bekerja menambahkan isi register HL jika digunakan untuk melakukan pelacakan data di memori dimulai dari alamat terendah sesuai data alamat pada register HL. Sedangkan CPD bekerja mengurangi register HL jika digunakan untuk melakukan pelacakan data di memori dimulai dari alamat tertinggi sesuai data alamat pada register HL:
- b). Perbedaan : CPIR bekerja menambahkan isi register HL jika digunakan untuk melakukan pelacakan data di memori dimulai dari alamat terendah sesuai data alamat pada register HL diulangi secara otomatis sampai nilai data di memori sama dengan nilai atau data register A atau nilai register BC = 0000. Sedangkan CPDR bekerja mengurangi register HL jika digunakan untuk melakukan pelacakan data di memori dimulai dari alamat tertinggi sesuai data alamat pada register HL diulangi secara otomatis sampai nilai data di memori sama dengan nilai atau data register A atau nilai register BC = 0000:
- c). Perbedaan : CPI bekerja menambahkan isi register HL jika digunakan untuk melakukan pelacakan data di memori dimulai dari alamat terendah sesuai data alamat pada register HL. CPIR bekerja menambahkan isi register HL jika digunakan untuk melakukan pelacakan data di memori dimulai dari alamat terendah sesuai data alamat pada register HL diulangi secara otomatis sampai nilai data di memori sama dengan nilai atau data register A atau nilai register BC = 0000.
- d). Perbedaan : CPD bekerja mengurangi register HL jika digunakan untuk melakukan pelacakan data di memori dimulai dari alamat tertinggi sesuai data alamat pada register HL. CPDR bekerja mengurangi register HL jika digunakan untuk melakukan pelacakan data di memori dimulai dari alamat tertinggi sesuai data alamat pada register HL diulangi secara otomatis sampai nilai data di memori sama dengan nilai atau data register A atau nilai register BC = 0000

3. Kegiatan Belajar II: Memahami Instruksi Aritmetika

Dalam mikroprosesor Zilog Z-80 CPU instruksi-instruksi aritmetika yang tersedia jumlahnya terbatas pada instruksi penjumlahan (ADD dan ADC) dan pengurangan (SUB dan SBC) saja. Bagaimana dengan operasi aritmetika perkalian dan pembagian?. Persoalan aritmetika perkalian dan pembagian dapat diselesaikan dengan menggabungkan beberapa instruksi yang tersedia dalam program subroutin. Perkalian adalah penjumlahan berulang. Pembagian adalah pengurangan berulang.

Kebutuhan hitung menghitung lazimnya didasarkan bilangan desimal. Pada mikroprosesor operasi penjumlahan dan pengurangan dilakukan dalam sistim komplemen berbasis dua. Instruksi Decimal Adjust Accumulator (DAA) disediakan untuk memberikan faktor koreksi pada saat kita bekerja dalam sistim bilangan desimal dalam kode BCD.

Instruksi CP,s disediakan untuk membandingkan isi akumulator dengan sebuah data tanpa merubah isi akumulator. Instruksi ini memberikan akibat pada perubahan register flag sebagai status pembandingannya. Status tersebut diantaranya adalah (S=Sign, Z=Zero, H=Half Carry, dan C=Carry). Dalam melaksanakan instruksi pembandingan, mikroprosesor menggunakan sistim bilangan komplemen dua.

Pada sistim komplemen dua bilangan terkecil adalah $80H = 1000\ 0000B = -128$ dan bilangan terbesar adalah $7FH = 0111\ 1111 = +127$.

a. Instruksi ADD

Instruksi ADD digunakan untuk melakukan operasi penjumlahan 8 bit dan 16 bit. Ada 38 jenis perintah penjumlahan pada mikroprosesor Z-80 CPU. Pada operasi 8 bit register A (akumulator) ditambahkan dengan isi sebuah register 8 bit atau data immediate 8 bit, atau data pada satu lokasi memori yang alamatnya dicatat oleh register HL, IX, atau IY.

Pada operasi aritmetika 16 bit register HL, IX, dan IY berfungsi sebagai akumulator yang dapat ditambahkan dengan isi register BC, DE, HL, SP. Untuk lebih jelasnya perhatikan tabel berikut:

Contoh:

Operasi	Assembly	Simbol Operasi	Keterangan
8 Bit	ADD A , A	A ← A + A	➤ Mempengaruhi Flag S, Z, H, V, C ➤ N = data 8 bit
	ADD A , B	A ← A + B	
	ADD A , C	A ← A + C	
	ADD A , D	A ← A + D	
	ADD A , E	A ← A + E	
	ADD A , H	A ← A + H	
	ADD A , L	A ← A + L	
	ADD A , N	A ← A + N	
	ADD A , (HL)	A ← A + (HL)	
	ADD A , (IX+d)	A ← A + (IX+d)	
	ADD A , (IY+d)	A ← A + (IY+d)	
	16 Bit	ADD HL , BC	
ADD HL , DE		HL ← HL + DE	
ADD HL , HL		HL ← HL + HL	
ADD HL , SP		HL ← HL + SP	
ADD IX , BC		IX ← IX + BC	
ADD IX , DE		IX ← IX + DE	
ADD IX , IX		IX ← IX + IX	
ADD IX , SP		IX ← IX + SP	
ADD IY , BC		IY ← IY + BC	
ADD IY , DE		IY ← IY + DE	
ADD IY , IY		IY ← IY + IY	
ADD IY , SP		IY ← IY + SP	

Perhatikan semua perintah Assembly dan maknanya pada kolom simbol operasi. Misalnya perintah ADD A,B proses operasinya disimbolkan $A \leftarrow A + B$. Artinya nilai register A yang baru sama dengan nilai register A semula ditambahkan dengan nilai data register B. Misalnya jika semula A = 2Ch dan B = 35h maka nilai yang baru A = 2C + 35 = 61h.

1). Kasus

No.	ADDRESS	Kode operasi	Assembly	Simbol Operasi
1.	1800	3E 4E	LD A , 4Eh	A ← 4Eh
2.	1802	06 1F	LD B , 1Fh	B ← 1Fh
3.	1804	0E 3D	LD C , 3Dh	C ← 3Dh
4.	1806	16 F4	LD D , F4h	D ← F4H
5.	1808	80	ADD A, B	A ← A + B
6.	1809	81	ADD A, C	A ← A + C
7.	180A	82	ADD A, D	A ← A + D
8.	180B	21 56 24	LD HL,2456h	HL ← 2456h
9.	181E	1E 45	LD E, 45h	E ← 45h
10.	1810	09	ADD HL,BC	HL ← HL + BC
11.	1811	19	ADD HL,DE	HL ← HL + DE
12.	1812	FF	RST 38	STOP

2). Isikan hasil identifikasi perintah kasus 1) pada tabel berikut. Gunakan kolom simbol operasi untuk merumuskan hasilnya.

No.	Assembly	Kategori	Hasil
1.	LD A , 4Eh
2.	LD B , 1Fh
3.	LD C , 3Dh
4.	LD D , F4h
5.	ADD A, B
6.	ADD A, C
7.	ADD A, D
8.	LD HL,2456h
9.	LD E, 45h
10.	ADD HL,BC
11.	ADD HL,DE
12.	RST 38

- 3). Dengan perintah eksekusi STEP pelajari program latihan kasus 1) step demi step. Cocokkan hasil identifikasi saudara dengan hasil eksekusi

No.	Assembly	Hasil Identifikasi	Hasil Eksekusi STEP
1.	LD A , 4Eh
2.	LD B , 1Fh
3.	LD C , 3Dh
4.	LD D , F4h
5.	ADD A, B
6.	ADD A, C
7.	ADD A, D
8.	LD HL,2456h
9.	LD E, 45h
10.	ADD HL,BC
11.	ADD HL,DE
12.	RST 38

- 4). Apakah hasil identifikasi saudara sama dengan hasil eksekusi. Jika ya apa kesimpulan saudara. Jika tidak sama jelaskan apa sebabnya.

.....

.....

.....

.....

.....

.....

.....

.....

- 5). Ulangi kaji sekali lagi kasus 1) tersebut. Jika saudara sudah memahami dengan baik silahkan meneruskan ke sub kegiatan belajar Instruksi SUB berikut ini.

Kunci Jawaban :

No.	Assembly	Kategori	Hasil
1.	LD A , 4Eh	data immediate ke register 8 bit	A = 4E
2.	LD B , 1Fh	data immediate ke register 8 bit	B = 1F
3.	LD C , 3Dh	data immediate ke register 8 bit	C = 3D
4.	LD D , F4h	data immediate ke register 8 bit	D = F4
5.	ADD A, B	Aritmetika jumlah 8 bit	A = 6D
6.	ADD A, C	Aritmetika jumlah 8 bit	A = AA
7.	ADD A, D	Aritmetika jumlah 8 bit	A = 9E ; Cy=1
8.	LD HL,2456h	data immediate ke register 16 bit	HL = 2456
9.	LD E, 45h	data immediate ke register 8 bit	E = 45
10.	ADD HL,BC	Aritmetika jumlah 8 bit	HL = 4393
11.	ADD HL,DE	Aritmetika jumlah 8 bit	HL = 37D8 C=1
12.	RST 38	STOP	

b. Instruksi SUB.

Instruksi SUB digunakan hanya untuk melakukan operasi pengurangan 8 bit. Pada operasi SUB isi register A dikurangkan dengan salah satu isi register A, B, C, D, E, H, L, atau data immediate 8 bit. Disamping juga isi register A dapat dikurangi dengan data pada suatu lokasi memori yang alamatnya dicatat oleh register HL, IX, dan IY.

Contoh:

Operasi	Assembly	Simbol Operasi	Keterangan
8 Bit	SUB , A	$A \leftarrow A - A$	<ul style="list-style-type: none"> ➤ Mempengaruhi Flag S, Z, H, V, C ➤ N = data 8 bit
	SUB , B	$A \leftarrow A - B$	
	SUB , C	$A \leftarrow A - C$	
	SUB , D	$A \leftarrow A - D$	
	SUB , E	$A \leftarrow A - E$	
	SUB , H	$A \leftarrow A - H$	
	SUB , L	$A \leftarrow A - L$	
	SUB , N	$A \leftarrow A - N$	
	SUB , (HL)	$A \leftarrow A - (HL)$	
	SUB , (IX+d)	$A \leftarrow A - (IX+d)$	
	SUB , (IY+d)	$A \leftarrow A - (IY+d)$	

Instruksi SUB dapat mempengaruhi status Sign, Zerro, HalfCarry, Overflow, dan Carry pada Register Flag. Pada instruksi ADD flag N = 0 dan pada instruksi SUB flag N = 1. Dua keadaan ini digunakan untuk menyatakan fungsi flag C sebagai carry atau borrow.

1). Kasus

No.	ADDRESS	Kode operasi	Assembly	Simbol Operasi
1.	1800	3E EE	LD A , EEh	$A \leftarrow EEh$
2.	1802	06 1F	LD B , 1Fh	$B \leftarrow 1Fh$
3.	1804	0E 3D	LD C , 3Dh	$C \leftarrow 3Dh$
4.	1806	16 24	LD D , 24h	$D \leftarrow 24H$
5.	1808	90	SUB , B	$A \leftarrow A - B$
6.	1809	91	SUB , C	$A \leftarrow A - C$
7.	180A	92	SUB , D	$A \leftarrow A - D$
8	180B	FF	RST 38	STOP

2). Isikan hasil identifikasi perintah kasus 1) pada tabel berikut. Gunakan kolom simbol operasi untuk merumuskan hasilnya.

No.	Assembly	Kategori	Hasil
1.	LD A , EEh
2.	LD B , 1Fh
3.	LD C , 3Dh
4.	LD D , 24h
5.	SUB , B
6.	SUB , C
7.	SUB , D
8.	RST 38

3). Dengan perintah eksekusi STEP pelajari program latihan kasus 1) step demi step. Cocokkan hasil identifikasi saudara dengan hasil eksekusi

No.	Assembly	Hasil Identifikasi	Hasil Eksekusi STEP
1.	LD A , EEh
2.	LD B , 1Fh
3.	LD C , 3Dh
4.	LD D , 24h
5.	SUB , B
6.	SUB , C
7.	SUB , D
8.	RST 38

4). Apakah hasil identifikasi saudara sama dengan hasil eksekusi. Jika ya apa kesimpulan saudara. Jika tidak sama jelaskan apa sebabnya.

.....

.....

.....

.....

.....

.....

.....

.....

5). Ulangi kaji sekali lagi kasus 1) tersebut. Jika saudara sudah memahami dengan baik silahkan meneruskan ke sub kegiatan belajar Instruksi ADC dan SBC berikut ini.

Kunci Jawaban :

No.	Assembly	Kategori	Hasil
1.	LD A , EEh	data immediate ke register 8 bit	A = EE
2.	LD B , 1Fh	data immediate ke register 8 bit	B = 1F
3.	LD C , 3Dh	data immediate ke register 8 bit	C = 3D
4.	LD D , 24h	data immediate ke register 8 bit	D = 24
5.	SUB , B	Aritmetika jumlah 8 bit	A = CF
6.	SUB , C	Aritmetika jumlah 8 bit	A = 92
7.	SUB , D	Aritmetika jumlah 8 bit	A = 6E
8.	RST 38	STOP	

c. Instruksi ADC (ADD With Carry) dan SBC (Sub With Carry)

Instruksi ADC digunakan untuk menambahkan isi register A dengan data 8 bit yang berada pada suatu register atau data immediate atau data suatu memori dan mengikut sertakan bit Carry (C). Instruksi ADC juga digunakan untuk menambahkan isi register HL dengan data 16 bit yang berada pada register BC, DE, HL, dan SP dengan mengikut sertakan bit Carry Flag (C).

Contoh :

Operasi	Assembly	Simbol Operasi	Keterangan
8 Bit	ADC A , A	$A \leftarrow A + A + Cy$	<ul style="list-style-type: none"> ➤ Mempengaruhi Flag S, Z, H, V, C ➤ N = data 8 bit
	ADC A , B	$A \leftarrow A + B + Cy$	
	ADC A , C	$A \leftarrow A + C + Cy$	
	ADC A , D	$A \leftarrow A + D + Cy$	
	ADC A , E	$A \leftarrow A + E + Cy$	
	ADC A , H	$A \leftarrow A + H + Cy$	
	ADC A , L	$A \leftarrow A + L + Cy$	
	ADC A , N	$A \leftarrow A + N + Cy$	
	ADC A , (HL)	$A \leftarrow A + (HL) + Cy$	
	ADC A , (IX+d)	$A \leftarrow A + (IX+d) + Cy$	
	ADC A , (IY+d)	$A \leftarrow A + (IY+d) + Cy$	
16 Bit	ADC HL , BC	$HL \leftarrow HL + BC + Cy$	<ul style="list-style-type: none"> ➤ Hanya Mempengaruhi Flag carry
	ADC HL , DE	$HL \leftarrow HL + DE + Cy$	
	ADC HL , HL	$HL \leftarrow HL + HL + Cy$	
	ADC HL , SP	$HL \leftarrow HL + SP + Cy$	

Instruksi SBC digunakan untuk mengurangi isi register A dengan data 8 bit yang berada pada suatu register atau data immediate atau data suatu memori dengan mengikutsertakan bit carry flag. Instruksi SBC juga digunakan untuk mengurangi isi register HL dengan data 16 bit yang berada pada register BC, DE, HL, dan SP dengan mengikutsertakan bit Carry Flag (Cy). Hasil dari kedua bentuk pengurangan tersebut dicatat di Register A dan Register HL.

Contoh :

Operasi	Assembly	Simbol Operasi	Keterangan
8 Bit	SBC A , A	$A \leftarrow A - A - Cy$	<ul style="list-style-type: none"> ➤ Mempengaruhi Flag S, Z, H, V, C ➤ N = data 8 bit
	SBC A , B	$A \leftarrow A - B - Cy$	
	SBC A , C	$A \leftarrow A - C - Cy$	
	SBC A , D	$A \leftarrow A - D - Cy$	
	SBC A , E	$A \leftarrow A - E - Cy$	
	SBC A , H	$A \leftarrow A - H - Cy$	
	SBC A , L	$A \leftarrow A - L - Cy$	
	SBC A , N	$A \leftarrow A - N - Cy$	
	SBC A , (HL)	$A \leftarrow A - (HL) - Cy$	
	SBC A , (IX+d)	$A \leftarrow A - (IX+d) - Cy$	
	SBC A , (IY+d)	$A \leftarrow A - (IY+d) - Cy$	
	16 Bit	SBC HL , BC	
SBC HL , DE		$HL \leftarrow HL - DE - Cy$	
SBC HL , HL		$HL \leftarrow HL - HL - Cy$	
SBC HL , SP		$HL \leftarrow HL - SP - Cy$	

1). Kasus

No.	ADDRESS	Kode operasi	Assembly	Simbol Operasi
1.	1800	3E 4E	LD A , 4Eh	$A \leftarrow 4Eh$
2.	1802	06 1F	LD B , 1Fh	$B \leftarrow 1Fh$
3.	1804	0E 3D	LD C , 3Dh	$C \leftarrow 3Dh$
4.	1806	16 F4	LD D , F4h	$D \leftarrow F4H$
5.	1808	80	ADD A , B	$A \leftarrow A + B$
6.	1809	89	ADC A , C	$A \leftarrow A + C + Cy$
7.	180A	8A	ADC A , D	$A \leftarrow A + D + Cy$
8.	180B	98	SBC A , B	$A \leftarrow A - B - Cy$
9.	180C	99	SBC A , C	$A \leftarrow A - C - Cy$
10.	180D	9A	SBC A , D	$A \leftarrow A - D - Cy$
11.	180E	FF	RST 38	STOP

2). Isikan hasil identifikasi perintah kasus 1) pada tabel berikut. Gunakan kolom simbol operasi untuk merumuskan hasilnya.

No.	Assembly	Kategori	Hasil
1.	LD A , 4Eh
2.	LD B , 1Fh
3.	LD C , 3Dh
4.	LD D , F4h
5.	ADD A, B
6.	ADC A, C
7.	ADC A, D
8.	SBC A, B
9.	SBC A, C
10.	SBC A, D
11	RST 38

- 3). Dengan perintah eksekusi STEP pelajari program latihan kasus 1) step demi step.
Cocokkan hasil identifikasi saudara dengan hasil eksekusi

No.	Assembly	Hasil Identifikasi	Hasil Eksekusi STEP
1.	LD A , 4Eh
2.	LD B , 1Fh
3.	LD C , 3Dh
4.	LD D , F4h
5.	ADD A, B
6.	ADC A, C
7.	ADC A, D
8.	SBC A, B
9.	SBC A, C
10.	SBC A, D
11.	SBC A, D
12.	RST 38

- 4). Apakah hasil identifikasi saudara sama dengan hasil eksekusi. Jika ya apa kesimpulan saudara. Jika tidak sama jelaskan apa sebabnya.

.....

.....

.....

.....

- 5). Ulangi kaji sekali lagi kasus 1) tersebut. Jika saudara sudah memahami dengan baik silahkan meneruskan ke sub kegiatan belajar Pelacakan Data berikut ini.

Kunci Jawaban :

No.	Assembly	Kategori	Hasil
1.	LD A , 4Eh	data immediate ke register 8 bit	A = 4E
2.	LD B , 1Fh	data immediate ke register 8 bit	B = 1F
3.	LD C , 3Dh	data immediate ke register 8 bit	C = 3D
4.	LD D , F4h	data immediate ke register 8 bit	D = F4
5.	ADD A, B	aritmetika jumlah 8 bit	A = 6D; Cy=0
6.	ADC A, C	aritmetika jumlah 8 bit dengan Cy	A = AA; Cy=0
7.	ADC A, D	aritmetika jumlah 8 bit dengan Cy	A = 9E; Cy=1
8.	SBC A, B	aritmetika kurang 8 bit dengan Cy	A = 7E; Cy=0
9.	SBC A, C	aritmetika kurang 8 bit dengan Cy	A = 41; Cy=0
10.	SBC A, D	aritmetika kurang 8 bit dengan Cy	A = 4D; Cy=0
11	RST 38	STOP	

No.	Assembly	Hasil Identifikasi	Hasil Eksekusi STEP
1.	LD A , 4Eh	A = 4E
2.	LD B , 1Fh	B = 1F
3.	LD C , 3Dh	C = 3D
4.	LD D , F4h	D = F4
5.	ADD A, B	A = 6D; Cy=0
6.	ADC A, C	A = AA; Cy=0
7.	ADC A, D	A = 9E; Cy=1
8.	SBC A, B	A = 7E; Cy=0
9.	SBC A, C	A = 41; Cy=0
10.	SBC A, D	A = 4D; Cy=0
11.	RST 38		

d. Instruksi INC (Increment) dan DEC (Decrement)

Instruksi INC digunakan untuk menambah isi suatu register atau memori dengan satu nilai. Instruksi ini sangat potensial digunakan untuk membuat counter cacah naik.

Contoh :

Operasi	Assembly	Simbol Operasi	Keterangan
8 Bit	INC A	$A \leftarrow A + 1$	➤ Mempengaruhi Flag S, Z, H, V, C
	INC B	$B \leftarrow B + 1$	
	INC C	$C \leftarrow C + 1$	
	INC D	$D \leftarrow D + 1$	
	INC E	$E \leftarrow E + 1$	
	INC H	$H \leftarrow H + 1$	
	INC L	$L \leftarrow L + 1$	
16 Bit	INC BC	$BC \leftarrow BC + 1$	
	INC DE	$DE \leftarrow DE + 1$	
	INC HL	$HL \leftarrow HL + 1$	
	INC IX	$IX \leftarrow IX + 1$	
	INC IY	$IY \leftarrow IY + 1$	
	INC SP	$SP \leftarrow SP + 1$	
Memori	INC (HL)	$(HL) \leftarrow (HL) + 1$	
	INC (IX+d)	$(IX+d) \leftarrow (IX+d) + 1$	
	INC (IY+d)	$(IY+d) \leftarrow (IY+d) + 1$	

Instruksi DEC digunakan untuk mengurangi isi register atau data suatu memori dengan nilai 1.

Contoh:

Operasi	Assembly	Simbol Operasi	Keterangan
8 Bit	DEC A	$A \leftarrow A - 1$	Mempengaruhi Flag S, Z, H, V, C
	DEC B	$B \leftarrow B - 1$	
	DEC C	$C \leftarrow C - 1$	
	DEC D	$D \leftarrow D - 1$	
	DEC E	$E \leftarrow E - 1$	
	DEC H	$H \leftarrow H - 1$	
	DEC L	$L \leftarrow L - 1$	

Operasi	Assembly	Simbol Operasi	Keterangan
16 Bit	DEC BC	BC ← BC - 1	
	DEC DE	DE ← DE - 1	
	DEC HL	HL ← HL - 1	
	DEC IX	IX ← IX - 1	
	DEC IY	IY ← IY - 1	
	DEC SP	SP ← SP - 1	
Memori	DEC (HL)	(HL) ← (HL) - 1	
	DEC (IX+d)	(IX+d) ← (IX+d) - 1	
	DEC (IY+d)	(IY+d) ← (IY+d) - 1	

1). Kasus

No.	ADDRESS	Kode operasi	Assembly	Simbol Operasi
1.	1800	3E 4E	LD A , 4Eh	A ← 4Eh
2.	1802	06 1F	LD B , 1Fh	B ← 1Fh
3.	1804	0E 3D	LD C , 3Dh	C ← 3Dh
4.	1806	16 F4	LD D , F4h	D ← F4H
5.	1808	3C	INC A	A ← A + 1
6.	1809	04	INC B	B ← B + 1
7.	180A	0C	INC C	C ← C + 1
8.	180B	14	INC D	D ← D + 1
9.	180C	15	DEC D	D ← D - 1
10.	180D	0D	DEC C	C ← C - 1
11.	180E	FF	RST 38	STOP

2). Isikan hasil identifikasi perintah kasus 1) pada tabel berikut. Gunakan kolom simbol operasi untuk merumuskan hasilnya.

No.	Assembly	Kategori	Hasil
1.	LD A , 4Eh
2.	LD B , 1Fh
3.	LD C , 3Dh
4.	LD D , F4h
5.	INC A
6.	INC B
7.	INC C
8.	INC D
9.	DEC D
10.	DEC C
11	RST 38

Kunci Jawaban :

No.	Assembly	Kategori	Hasil
1.	LD A , 4Eh	data immediate ke register 8 bit	A = 4E
2.	LD B , 1Fh	data immediate ke register 8 bit	B = 1F
3.	LD C , 3Dh	data immediate ke register 8 bit	C = 3D
4.	LD D , F4h	data immediate ke register 8 bit	D = F4
5.	INC A	aritmetika increament	A = 4F
6.	INC B	aritmetika increament	B = 20
7.	INC C	aritmetika increament	C = 3E
8.	INC D	aritmetika increament	D = F5
9.	DEC D	aritmetika decreament	D = F4
10.	DEC C	aritmetika decreament	C = 3D
11	RST 38	STOP/BERHENTI	

e. Instruksi Aritmetika Khusus

Dalam operasi aritmetika disediakan beberapa instruksi khusus yaitu :

- DAA mnemonic dari Decimal Adjust Accumulator
- CPL mnemonic dari Complement Accumulator (Komplemen 1)
- NEG mnemonic dari Negate Accumulator (Komplemen 2)

1). Instruksi DAA

Instruksi DAA digunakan untuk merubah isi register A ke bentuk BCD. Instruksi DAA digunakan untuk memberi faktor koreksi pada saat bekerja dengan bilangan desimal.

DAA dalam melakukan koreksi bekerja sbb :

Jika Bit b3, b2, b1, b0 > 9 atau ada Half Carry (H = 1) maka bit b3, b2, b1, b0 ditambah dengan 0110 = 6.

Jika Bit b7, b6, b5, b4 > 9 atau ada Carry (C = 1) maka bit b7, b6, b5, b4 ditambah dengan 0110 = 6.

Contoh :

	Desimal	BCD
	29 ₁₀	0010 1001
	26 ₁₀	0010 0110
+	<hr/>	<hr/>
	55 ₁₀	0100 1111 = 4F

Nilai hasil BCD seharusnya 55 tidak sesuai dengan hasil penjumlahan biner 0100 1111 = 4F.

Koreksi DAA

0100 1111
<hr/> 0000 0110
0101 0101 = 55

2). Instruksi CPL (Complement)

Instruksi CPL digunakan untuk merubah isi akumulator menjadi bentuk komplemen 1 yaitu dengan menginverse semua bit yang ada di akumulator.

$$A \longleftarrow \text{NOT } A$$

3). Instruksi NEG (Negate)

Instruksi NEG digunakan untuk merubah isi akumulator (register A) menjadi bentuk negatifnya yaitu dengan merubahnya menjadi nilai komplemen dua.

$$A \leftarrow \text{NOT } A + 1$$

4). Instruksi CP (Compare)

Digunakan untuk membandingkan isi akumulator dengan data immediate 8 bit atau isi salah satu register 8 bit atau isi/data suatu lokasi memori *tanpa merubah isi akumulator*. Instruksi CP membangun keadaan pada status Flag pada Bit Sign, Zero, Over Flow, Half Carry dan Carry pada Register Flag. Instruksi CP sangat baik digunakan untuk menguji sebuah data apakah data tersebut sama dengan suatu nilai tertentu atau lebih atau lebih kecil dari suatu nilai tertentu.

Contoh:

Operasi	Assembly	Simbol Operasi	Keterangan
8 Bit	CP A	A - A	Mempengaruhi Flag S, Z, H, V, C
	CP B	A - B	
	CP C	A - C	Nilai A tetap atau tidak berubah
	CP D	A - D	
	CP E	A - E	
	CP H	A - H	
	CP L	A - L	
	CP N	A - N	
	CP (HL)	A - (HL)	
	CP (IY + d)	A - (IY + d)	
	CP (IX + d)	A - (IX + d)	

4. Kegiatan Belajar III: Memahami Instruksi Logika

a. Instruksi LOGIKA AND, OR, dan XOR

Instruksi AND, OR, dan XOR digunakan untuk melakukan operasi logika isi dari akumulator terhadap data suatu register 8 bit atau data immediate, atau data suatu lokasi memori.

Contoh:

Operasi	Assembly	Simbol Operasi	Keterangan
8 Bit memori	AND A	$A \leftarrow A \wedge A$	➤ Mempengaruhi Flag S, Z, H, V, C
	AND B	$A \leftarrow A \wedge B$	
	AND C	$A \leftarrow A \wedge C$	
	AND D	$A \leftarrow A \wedge D$	
	AND E	$A \leftarrow A \wedge E$	
	AND H	$A \leftarrow A \wedge H$	
	AND L	$A \leftarrow A \wedge L$	
	AND (HL)	$A \leftarrow A \wedge (HL)$	
	AND (IX+d)	$A \leftarrow A \wedge (IX +d)$	
	AND (IY+d)	$A \leftarrow A \wedge (IY +d)$	

Pola di atas berlaku juga pada operasi LOGIKA OR dan XOR. Simbol operasi Logika adalah sbb :

\wedge : untuk LOGIKA AND

\vee : untuk LOGIKA OR

\oplus : untuk LOGIKA XOR

1). Kasus

No.	ADDRESS	Kode operasi	Assembly	Simbol Operasi
1.	1800	3E 4E	LD A , 4Eh	$A \leftarrow 4Eh$
2.	1802	06 1F	LD B , 1Fh	$B \leftarrow 1Fh$
3.	1804	0E 3D	LD C , 3Dh	$C \leftarrow 3Dh$
4.	1806	16 F4	LD D , F4h	$D \leftarrow F4H$
5.	1808	A7	AND A	$A \leftarrow A \wedge A$
6.	1809	A0	AND B	$A \leftarrow A \wedge B$
7.	180A	B0	OR B	$A \leftarrow A \vee B$
8.	180B	B7	OR A	$A \leftarrow A \vee B$
9.	180C	AA	XOR D	$A \leftarrow A \oplus D$
10.	180D	AF	XOR A	$A \leftarrow A \oplus A$
11.	180E	FF	RST 38	STOP

2). Isikan hasil identifikasi perintah kasus 1) pada tabel berikut. Gunakan kolom simbol operasi untuk merumuskan hasilnya.

No.	Assembly	Kategori	Hasil
1.	LD A , 4Eh
2.	LD B , 1Fh
3.	LD C , 3Dh
4.	LD D , F4h
5.	AND A
6.	AND B
7.	OR B
8.	OR A
9.	XOR D
10.	XOR A
11.	RST 38

Kunci Jawaban :

No.	Assembly	Kategori	Hasil
1.	LD A , 4Eh	data immediate ke register 8 bit	A = 4E
2.	LD B , 1Fh	data immediate ke register 8 bit	B = 1F
3.	LD C , 3Dh	data immediate ke register 8 bit	C = 3D
4.	LD D , F4h	data immediate ke register 8 bit	D = F4
5.	AND A	logika And	A = 4E
6.	AND B	logika And	A = 0E
7.	OR B	logika Or	A = 1F
8.	OR A	logika Or	A = 1F
9.	XOR D	logika XOR	A = EB
10.	XOR A	logika XOR	A = 00
11.	RST 38	STOP	

lampiran

Daftar Alfabetik

Set Instruksi Z-80

No	OP-CODE	ASSEMBLY		No	OP-CODE	ASSEMBLY	
1.	8E	ADC	A, (HL)	49.	E6 20	AND	n
2.	DD 8E XX	ADC	A, (IX+d)	50.	CB 46	BIT	0, (HL)
3.	FD 8E XX	ADC	A, (IX+d)	51.	DD CB XX 46	BIT	0, (IX+d)
4.	8F	ADC	A, A	52.	FD CB XX 46	BIT	0, (IY+d)
5.	88	ADC	A, B	53.	CB 47	BIT	0, A
6.	89	ADC	A, C	54.	CB 40	BIT	0, B
7.	8A	ADC	A, D	55.	CB 41	BIT	0, C
8.	8B	ADC	A, E	56.	CB 42	BIT	0, D
9.	8C	ADC	A, H	57.	CB 43	BIT	0, E
10.	8D	ADC	A, L	58.	CB 44	BIT	0, H
11.	CE XX	ADC	A, n	59.	CB 45	BIT	0, L
12.	ED 4A	ADC	HL, BC	60.	CB 4E	BIT	1, (HL)
13.	ED 5A	ADC	HL, DE	61.	DD CB XX 4E	BIT	1, (IX+d)
14.	ED 6A	ADC	HL, HL	62.	FD CB XX 4E	BIT	1, (IY+d)
15.	ED 7A	ADC	HL, SP	63.	CB 4F	BIT	1, A
16.	86	ADD	A, (HL)	64.	CB 48	BIT	1, B
17.	DD 86 XX	ADD	A, (IX + d)	65.	CB 49	BIT	1, C
18.	FD 86 XX	ADD	A, (IY + d)	66.	CB 4A	BIT	1, D
19.	87	ADD	A, A	67.	CB 4B	BIT	1, E
20.	80	ADD	A, B	68.	CB 4C	BIT	1, H
21.	81	ADD	A, C	69.	CB 4D	BIT	1, L
22.	82	ADD	A, D	70.	CB 56	BIT	2, (HL)
23.	83	ADD	A, E	71.	DD CB XX 56	BIT	2, (IX+d)
24.	84	ADD	A, H	72.	FD CB XX 56	BIT	2, (IY+d)
25.	85	ADD	A, L	73.	CB 57	BIT	2, A
26.	C6 XX	ADD	A, n	74.	CB 50	BIT	2, B
27.	09	ADD	HL, BC	75.	CB 51	BIT	2, C
28.	19	ADD	HL, DE	76.	CB 52	BIT	2, D
29.	29	ADD	HL, HL	77.	CB 53	BIT	2, E
30.	39	ADD	HL, SP	78.	CB 54	BIT	2, H
31.	DD 09	ADD	IX, BC	79.	CB 55	BIT	2, L
32.	DD 19	ADD	IX, DE	80.	CB 5E	BIT	3, (HL)
33.	DD 29	ADD	IX, IX	81.	DD CB XX 5E	BIT	3, (IX+d)
34.	DD 39	ADD	IX, SP	82.	FD CB XX 5E	BIT	3, (IY+d)
35.	FD 09	ADD	IY, BC	83.	CB 5F	BIT	3, A
36.	FD 19	ADD	IY, DE	84.	CB 58	BIT	3, B
37.	FD 29	ADD	IY, IY	85.	CB 59	BIT	3, C
38.	FD 39	ADD	IY, SP	86.	CB 5A	BIT	3, D
39.	A6	AND	(HL)	87.	CB 5B	BIT	3, E
40.	DD A6 XX	AND	(IX + d)	88.	CB 5C	BIT	3, H
41.	FD A6 XX	AND	(IY + d)	89.	CB 5D	BIT	3, L
42.	A7	AND	A	90.	CB 66	BIT	4, (HL)
43.	A0	AND	B	91.	DD CB XX 66	BIT	4, (IX+d)
44.	A1	AND	C	92.	FD CB XX 66	BIT	4, (IY+d)
45.	A2	AND	D	93.	CB 67	BIT	4, A
46.	A3	AND	E	94.	CB 60	BIT	4, B
47.	A4	AND	H	95.	CB 61	BIT	4, C
48.	A5	AND	L	96.	CB 62	BIT	4, D

No	OP-CODE	ASSEMBLY		No	OP-CODE	ASSEMBLY	
97.	CB 63	BIT	4, E	147.	BB	CP	E
98.	CB 64	BIT	4, H	148.	BC	CP	H
99.	CB 65	BIT	4, L	149.	BD	CP	L
100.	CB 6E	BIT	5, (HL)	150.	FE XX	CP	n
101.	DD CB XX 6E	BIT	5, (IX+d)	151.	ED A9	CPD	
102.	FD CB XX 6E	BIT	5, (IY+d)	152.	ED B9	CPDR	
103.	CB 6F	BIT	5, A	153.	ED A1	CPI	
104.	CB 68	BIT	5, B	154.	ED B1	CPIR	
105.	CB 69	BIT	5, C	155.	2F	CPL	
106.	CB 6A	BIT	5, D	156.	27	DAA	
107.	CB 6B	BIT	5, E	157.	35	DEC	(HL)
108.	CB 6C	BIT	5, H	158.	DD 35 XX	DEC	(IX+d)
109.	CB 6D	BIT	5, L	159.	FD 35 XX	DEC	(IY+d)
110.	CB 76	BIT	6, (HL)	160.	3D	DEC	A
111.	DD CB XX 76	BIT	6, (IX+d)	161.	05	DEC	B
112.	FD CB XX 76	BIT	6, (IY+d)	162.	0B	DEC	BC
113.	CB 77	BIT	6, A	163.	0D	DEC	C
114.	CB 70	BIT	6, B	164.	15	DEC	D
115.	CB 71	BIT	6, C	165.	1B	DEC	DE
116.	CB 72	BIT	6, D	166.	1D	DEC	E
117.	CB 73	BIT	6, E	167.	25	DEC	H
118.	CB 74	BIT	6, H	168.	2B	DEC	HL
119.	CB 75	BIT	6, L	169.	DD 2B	DEC	IX
120.	CB 7E	BIT	7, (HL)	170.	FD 2B	DEC	IY
121.	DD CB XX 7E	BIT	7, (IX+d)	171.	2D	DEC	L
122.	FD CB XX 7E	BIT	7, (IY+d)	172.	3B	DEC	SP
123.	CB 7F	BIT	7, A	173.	F3	DI	
124.	CB 78	BIT	7, B	174.	10 XX	DJNZ	DIS
125.	CB 79	BIT	7, C	175.	FB	EI	
126.	CB 7A	BIT	7, D	176.	E3	EX	(SP),HL
127.	CB 7B	BIT	7, E	177.	DD E3	EX	(SP),IX
128.	CB 7C	BIT	7, H	178.	FD E3	EX	(SP),IY
129.	CB 7D	BIT	7, L	179.	08	EX	AF,AF'
130.	DC XX XX	CALL	C, nn	180.	EB	EX	DE,HL
131.	FC XX XX	CALL	M, nn	181.	D9	EXX	
132.	D4 XX XX	CALL	NC, nn	182.	76	HALT	
133.	CD XX XX	CALL	nn	183.	ED 46	IM	0
134.	C4 XX XX	CALL	NZ, nn	184.	ED 56	IM	1
135.	F4 XX XX	CALL	P, nn	185.	ED 5E	IM	2
136.	EC XX XX	CALL	PE, nn	186.	ED 78	IN	A, (C)
137.	E4 XX XX	CALL	PO, nn	187.	DB XX	IN	A, n
138.	CC XX XX	CALL	Z, nn	188.	ED 40	IN	B, (C)
139.	3F	CCF		189.	ED 48	IN	C, (C)
140.	8E	CP	(HL)	190.	ED 50	IN	D, (C)
141.	DD 8E XX	CP	(IX+d)	191.	ED 58	IN	E, (C)
142.	FD 8E XX	CP	(IY+d)	192.	ED 60	IN	H, (C)
143.	BF	CP	A	193.	ED 68	IN	L, (C)
144.	B8	CP	B	194.	34	INC	(HL)
145.	B9	CP	C	195.	DD 34 XX	INC	(IX + d)
146.	BA	CP	D	196.	FD 34 XX	INC	(IY + d)

No	OP-CODE	ASSEMBLY		No	OP-CODE	ASSEMBLY	
197.	3C	INC	A	247.	DD 75 XX	LD	(IX + d),L
198.	04	INC	B	248.	DD 36 XX XX	LD	(IX + d),n
199.	03	INC	BC	249.	FD 77 XX	LD	(IY + d),A
200.	0C	INC	C	250.	FD 70 XX	LD	(IY + d),B
201.	14	INC	D	251.	FD 71 XX	LD	(IY + d),C
202.	13	INC	DE	252.	FD 72 XX	LD	(IY + d),D
203.	1C	INC	E	253.	FD 73 XX	LD	(IY + d),E
204.	24	INC	H	254.	FD 74 XX	LD	(IY + d),H
205.	23	INC	HL	255.	FD 75 XX	LD	(IY + d),L
206.	DD 23	INC	IX	256.	FD 36 XX XX	LD	(IY + d),n
207.	FD 23	INC	IY	257.	32 XX XX	LD	(nn),A
208.	2C	INC	L	258.	ED 43 XX XX	LD	(nn),BC
209.	33	INC	SP	259.	ED 53 XX XX	LD	(nn),DE
210.	ED AA	IND		260.	22 XX XX	LD	(nn),HL
211.	ED BA	INDR		261.	DD 22 XX XX	LD	(nn),IX
212.	ED A2	INI		262.	FD 22 XX XX	LD	(nn),IY
213.	ED B2	INIR		263.	ED 73 XX XX	LD	(nn),SP
214.	E9	JP	(HL)	264.	0A	LD	A, (BC)
215.	DD E9	JP	(IX)	265.	1A	LD	A, (DE)
216.	FD E9	JP	(IY)	266.	7E	LD	A, (HL)
217.	DA XX XX	JP	C, nn	267.	DD 7E XX	LD	A, (IX+d)
218.	FA XX XX	JP	M, nn	268.	FD 7E XX	LD	A, (IY+d)
219.	D2 XX XX	JP	NC, nn	269.	3A XX XX	LD	A, (nn)
220.	C3 XX XX	JP	nn	270.	7F	LD	A, A
221.	C2 XX XX	JP	NZ, nn	271.	78	LD	A, B
222.	F2 XX XX	JP	P, nn	272.	79	LD	A, C
223.	EA XX XX	JP	PE, nn	273.	7A	LD	A, D
224.	E2 XX XX	JP	PO, nn	274.	7B	LD	A, E
225.	CA XX XX	JP	Z, nn	275.	7C	LD	A, H
226.	18 xx	JR	DIS	276.	ED 57	LD	A, I
227.	38 xx	JR	C, DIS	277.	7D	LD	A, L
228.	30 xx	JR	NC, DIS	278.	3E XX	LD	A, n
229.	20 xx	JR	NZ, DIS	279.	46	LD	B, (HL)
230.	28 xx	JR	Z, DIS	280.	DD 46 XX	LD	B, (IX+d)
231.	02	LD	(BC), A	281.	FD 46 XX	LD	B, (IY+d)
232.	12	LD	(DE), A	282.	47	LD	B, A
233.	77	LD	(HL), A	283.	40	LD	B, B
234.	70	LD	(HL), B	284.	41	LD	B, C
235.	71	LD	(HL), C	285.	42	LD	B, D
236.	72	LD	(HL), D	286.	43	LD	B, E
237.	73	LD	(HL), E	287.	44	LD	B, H
238.	74	LD	(HL), H	288.	45	LD	B, L
239.	75	LD	(HL), L	289.	06 XX	LD	B, n
240.	36 XX	LD	(HL), n	290.	ED 4B XX XX	LD	BC, (nn)
241.	DD 77 XX	LD	(IX + d), A	291.	01 XX XX	LD	BC, nn
242.	DD 70 XX	LD	(IX + d), B	292.	4E	LD	C, (HL)
243.	DD 71 XX	LD	(IX + d), C	293.	DD 4E XX	LD	C, (IX+d)
244.	DD 72 XX	LD	(IX + d), D	294.	4F	LD	C, A
245.	DD 73 XX	LD	(IX + d), E	295.	48	LD	C, B
246.	DD 74 XX	LD	(IX + d), H	296.	49	LD	C, C

No	OP-CODE	ASSEMBLY		No	OP-CODE	ASSEMBLY	
297.	4A	LD	C, D	347.	6F	LD	L, A
298.	4B	LD	C, E	348.	68	LD	L, B
299.	4C	LD	C, H	349.	69	LD	L, C
300.	4D	LD	C, L	350.	6A	LD	L, D
301.	0E XX	LD	C, n	351.	6B	LD	L, E
302.	56	LD	D, (HL)	352.	6C	LD	L, H
303.	DD 56 XX	LD	D, (IX+d)	353.	6D	LD	L, L
304.	FD 56 XX	LD	D, (IY+d)	354.	2E XX	LD	L, n
305.	57	LD	D, A	355.	ED 7B XX XX	LD	SP, (nn)
306.	50	LD	D, B	356.	F9	LD	SP, HL
307.	51	LD	D, C	357.	DD F9	LD	SP, IX
308.	52	LD	D, D	358.	FD F9	LD	SP, IY
309.	53	LD	D, E	359.	31 XX XX	LD	SP, nn
310.	54	LD	D, H	360.	ED A8	LDD	
311.	55	LD	D, L	361.	ED B8	LDDR	
312.	16 XX	LD	D, n	362.	ED A0	LDI	
313.	ED 5B XX XX	LD	DE, (nn)	363.	ED B0	LDIR	
314.	11 XX XX	LD	DE, nn	364.	ED 44	NEG	
315.	5E	LD	E, (HL)	365.	00	NOP	
316.	DD 5E XX	LD	E, (IX+d)	366.	B6	OR	(HL)
317.	FD 5E XX	LD	E, (IY+d)	367.	DD B6 XX	OR	(IX+d)
318.	5F	LD	E, A	368.	FD B6 XX	OR	(IY+d)
319.	58	LD	E, B	369.	B7	OR	A
320.	59	LD	E, C	370.	B0	OR	B
321.	5A	LD	E, D	371.	B1	OR	C
322.	5B	LD	E, E	372.	B2	OR	D
323.	5C	LD	E, H	373.	B3	OR	E
324.	5D	LD	E, L	374.	B4	OR	H
325.	1E XX	LD	E, n	375.	B5	OR	L
326.	66	LD	H, (HL)	376.	F6 XX	OR	n
327.	DD 66 XX	LD	H, (IX+d)	377.	ED BB	OTDR	
328.	FD 66 XX	LD	H, (IY+d)	378.	ED B3	OTIR	
329.	67	LD	H, A	379.	ED 79	OUT	(C),A
330.	60	LD	H, B	380.	ED 41	OUT	(C),B
331.	61	LD	H, C	381.	ED 49	OUT	(C),C
332.	62	LD	H, D	382.	ED 51	OUT	(C),D
333.	63	LD	H, E	383.	ED 59	OUT	(C),E
334.	64	LD	H, H	384.	ED 61	OUT	(C),H
335.	65	LD	H, L	385.	ED 69	OUT	(C),L
336.	26 XX	LD	H, n	386.	D3 XX	OUT	n,A
337.	2A XX XX	LD	HL, (nn)	387.	ED AB	OUTB	
338.	21 XX XX	LD	HL, nn	388.	ED A3	OUTI	
339.	ED 47	LD	I, A	389.	F1	POP	AF
340.	D 2A XX X X	LD	IX, (nn)	390.	C1	POP	BC
341.	DD 21 XX XX	LD	IX, nn	391.	D1	POP	DE
342.	FD 2A XX XX	LD	IY, (nn)	392.	E1	POP	HL
343.	FD 21 XX XX	LD	IY, nn	393.	DD E1	POP	IX
344.	6E	LD	L, (HL)	394.	FD E1	POP	IY
345.	DD 6E XX	LD	L, (IX + d)	395.	F5	PUSH	AF
346.	FD 6E XX	LD	L, (IY + d)	396.	C5	PUSH	BC

No	OP-CODE	ASSEMBLY		No	OP-CODE	ASSEMBLY	
397.	D5	PUSH	DE	447.	CB A2	RES	4, D
398.	E5	PUSH	HL	448.	CB A3	RES	4, E
399.	DD E5	PUSH	IX	449.	CB A4	RES	4, H
400.	FD E5	PUSH	IY	450.	CB A5	RES	4, L
401.	CB 86	RES	0, (HL)	451.	CB AE	RES	5, (HL)
402.	DD CB XX 96	RES	0, (IX+d)	452.	DD CB XX AE	RES	5, (IX+d)
403.	FD CB XX 96	RES	0, (IY+d)	453.	FD CB XX AE	RES	5, (IY+d)
404.	CB 87	RES	0, A	454.	CB AF	RES	5, A
405.	CB 80	RES	0, B	455.	CB A8	RES	5, B
406.	CB 81	RES	0, C	456.	CB A9	RES	5, C
407.	CB 82	RES	0, D	457.	CB AA	RES	5, D
408.	CB 83	RES	0, E	458.	CB AB	RES	5, E
409.	CB 84	RES	0, H	459.	CB AC	RES	5, H
410.	CB 85	RES	0, L	460.	CB AD	RES	5, L
411.	CB 8E	RES	1, (HL)	461.	CB B6	RES	6, (HL)
412.	DD CB XX 8E	RES	1, (IX+d)	462.	DD CB XX B6	RES	6, (IX+d)
413.	FD CB XX 8E	RES	1, (IY+d)	463.	FD CB XX B6	RES	6, (IY+d)
414.	CB 8F	RES	1, A	464.	CB B7	RES	6, A
415.	CB 88	RES	1, B	465.	CB B0	RES	6, B
416.	CB 89	RES	1, C	466.	CB B1	RES	6, C
417.	CB 8A	RES	1, D	467.	CB B2	RES	6, D
418.	CB 8B	RES	1, E	468.	CB B3	RES	6, E
419.	CB 8C	RES	1, H	469.	CB B4	RES	6, H
420.	CB 8D	RES	1, L	470.	CB B5	RES	6, L
421.	CB 96	RES	2, (HL)	471.	CB BE	RES	7, (HL)
422.	DD CB XX 96	RES	2, (IX+d)	472.	DD CB XX BE	RES	7, (IX+d)
423.	FD CB XX 96	RES	2, (IY+d)	473.	FD CB XX BE	RES	7, (IY+d)
424.	CB 97	RES	2, A	474.	CB BF	RES	7, A
425.	CB 90	RES	2, B	475.	CB B8	RES	7, B
426.	CB 91	RES	2, C	476.	CB B9	RES	7, C
427.	CB 92	RES	2, D	477.	CB BA	RES	7, D
428.	CB 93	RES	2, E	478.	CB BB	RES	7, E
429.	CB 94	RES	2, H	479.	CB BC	RES	7, H
430.	CB 95	RES	2, L	480.	CB BD	RES	7, L
431.	CB 9E	RES	3, (HL)	481.	C9	RET	
432.	DD CB XX 9E	RES	3, (IX+d)	482.	D8	RET	C
433.	FD CB XX 9E	RES	3, (IY+d)	483.	F8	RET	M
434.	CB 9F	RES	3, A	484.	D0	RET	NC
435.	CB 98	RES	3, B	485.	C0	RET	NZ
436.	CB 99	RES	3, C	486.	F0	RET	P
437.	CB 9A	RES	3, D	487.	E8	RET	PE
438.	CB 9B	RES	3, E	488.	E0	RET	PO
439.	CB 9C	RES	3, H	489.	C8	RET	Z
440.	CB 9D	RES	3, L	490.	ED 4D	RETI	
441.	CB A6	RES	4, (HL)	491.	ED 45	RETN	
442.	DD CB XX A6	RES	4, (IX+d)	492.	CB 16	RL	(HL)
443.	FD CB XX A6	RES	4, (IY+d)	493.	DD CB XX 16	RL	(IX+d)
444.	CB A7	RES	4, A	494.	FD CB XX 16	RL	(IY+d)
445.	CB A0	RES	4, B	495.	CB 17	RL	A
446.	CB A1	RES	4, C	496.	CB 10	RL	B

No	OP-CODE	ASSEMBLY		No	OP-CODE	ASSEMBLY	
497.	CB 11	RL	C	547.	FD 9E XX	SBC	A, (IX+d)
498.	CB 12	RL	D	548.	9F	SBC	A, A
499.	CB 13	RL	E	549.	98	SBC	A, B
500.	CB 14	RL	H	550.	99	SBC	A, C
501.	CB 15	RL	L	551.	9A	SBC	A, D
502.	17	RLA		552.	9B	SBC	A, E
503.	DD CB XX 06	RLC	(IX+d)	553.	9C	SBC	A, H
504.	FD CB XX 06	RLC	(IY+d)	554.	9D	SBC	A, L
505.	CB 07	RLC	A	555.	DE XX	SBC	A, n
506.	CB 00	RLC	B	556.	ED 42	SBC	HL,BC
507.	CB 01	RLC	C	557.	ED 52	SBC	HL,DE
508.	CB 02	RLC	D	558.	ED 62	SBC	HL,HL
509.	CB 03	RLC	E	559.	ED 72	SBC	HL,SP
510.	CB 04	RLC	H	560.	37	SCF	
511.	CB 05	RLC	L	561.	CB C6	SET	0, (HL)
512.	07	RLCA		562.	DD CB XX C6	SET	0, (IX+d)
513.	ED 6F	RLD		563.	FD CB XX C6	SET	0, (IY+d)
514.	CB 1E	RR	(HL)	564.	CB C7	SET	0, A
515.	DD CB XX 1E	RR	(IX+d)	565.	CB C0	SET	0, B
516.	FD CB XX 1E	RR	(IY+d)	566.	CB C1	SET	0, C
517.	CB 1F	RR	A	567.	CB C2	SET	0, D
518.	CB 18	RR	B	568.	CB C3	SET	0, E
519.	CB 19	RR	C	569.	CB C4	SET	0, H
520.	CB 1A	RR	D	570.	CB C5	SET	0, L
521.	CB 1B	RR	E	571.	CB CE	SET	1, (HL)
522.	CB 1C	RR	H	572.	DD CB XX CE	SET	1, (IX+d)
523.	CB 1D	RR	L	573.	FD CB XX CE	SET	1, (IY+d)
524.	1F	RRA		574.	CB CF	SET	1, A
525.	CB 0E	RRC	(HL)	575.	CB C8	SET	1, B
526.	DD CB XX 0E	RRC	(IX+d)	576.	CB C9	SET	1, C
527.	FD CB XX 0E	RRC	(IY+d)	577.	CB CA	SET	1, D
528.	CB 0F	RRC	A	578.	CB CB	SET	1, E
529.	CB 08	RRC	B	579.	CB CC	SET	1, H
530.	CB 09	RRC	C	580.	CB CD	SET	1, L
531.	CB 0A	RRC	D	581.	CB D6	SET	2, (HL)
532.	CB 0B	RRC	E	582.	DD CB XX D6	SET	2, (IX+d)
533.	CB 0C	RRC	H	583.	FD CB XX D6	SET	2, (IY+d)
534.	CB 0D	RRC	L	584.	CB D7	SET	2, A
535.	0F	RRCA		585.	CB D0	SET	2, B
536.	ED 67	RRD		586.	CB D1	SET	2, C
537.	C7	RST	0	587.	CB D2	SET	2, D
538.	CF	RST	08H	588.	CB D3	SET	2, E
539.	D7	RST	10H	589.	CB D4	SET	2, H
540.	DF	RST	18H	590.	CB D5	SET	2, L
541.	E7	RST	20H	591.	CB DE	SET	3, (HL)
542.	EF	RST	28H	592.	DD CB XX DE	SET	3, (IX+d)
543.	F7	RST	30H	593.	FD CB XX DE	SET	3, (IY+d)
544.	FF	RST	38H	594.	CB DF	SET	3, A
545.	9E	SBC	A, (HL)	595.	CB D8	SET	3, B
546.	DD 9E XX	SBC	A, (IX + d)	596.	CB D9	SET	3, C

No	OP-CODE	ASSEMBLY		No	OP-CODE	ASSEMBLY	
597.	CB DA	SET	3, D	647.		SLA	D
598.	CB DB	SET	3, E	648.		SLA	E
599.	CB DC	SET	3, H	649.		SLA	H
600.	CB DD	SET	3, L	650.		SLA	L
601.	CB E6	SET	4, (HL)	651.		SRA	(HL)
602.	DD CB XX E6	SET	4, (IX+d)	652.		SRA	(IX+d)
603.	FD CB XX E6	SET	4, (IY+d)	653.		SRA	(IY+d)
604.	CB E7	SET	4, A	654.		SRA	A
605.	CB E0	SET	4, B	655.		SRA	B
606.	CB E1	SET	4, C	656.		SRA	C
607.	CB E2	SET	4, D	657.		SRA	D
608.	CB E3	SET	4, E	658.		SRA	E
609.	CB E4	SET	4, H	659.		SRA	H
610.	CB E5	SET	4, L	660.		SRA	L
611.	CB EE	SET	5, (HL)	661.		SRL	(HL)
612.	DD CB XX EE	SET	5, (IX+d)	662.		SRL	(IX+d)
613.	FD CB XX EE	SET	5, (IY+d)	663.		SRL	(IY+d)
614.	CB EF	SET	5, A	664.		SRL	A
615.	CB E8	SET	5, B	665.		SRL	B
616.	CB E9	SET	5, C	666.		SRL	C
617.	CB EA	SET	5, D	667.		SRL	D
618.	CB EB	SET	5, E	668.		SRL	E
619.	CB EC	SET	5, H	669.		SRL	H
620.	CB ED	SET	5, L	670.		SRL	L
621.	CB F6	SET	6, (HL)	671.		SUB	(HL)
622.	DD CB XX F6	SET	6, (IX+d)	672.		SUB	(IX+d)
623.	FD CB XX F6	SET	6, (IY+d)	673.		SUB	(IY+d)
624.	CB F7	SET	6, A	674.		SUB	A
625.	CB F0	SET	6, B	675.		SUB	B
626.	CB F1	SET	6, C	676.		SUB	C
627.	CB F2	SET	6, D	677.		SUB	D
628.	CB F3	SET	6, E	678.		SUB	E
629.	CB F4	SET	6, H	679.		SUB	H
630.	CB F5	SET	6, L	680.		SUB	L
631.	CB FE	SET	7, (HL)	681.		SUB	n
632.	DD CB XX FE	SET	7, (IX+d)	682.		XOR	(HL)
633.	FD CB XX FE	SET	7, (IY+d)	683.		XOR	(IX+d)
634.	CB FF	SET	7, A	684.		XOR	(IY+d)
635.	CB F8	SET	7, B	685.		XOR	A
636.	CB F9	SET	7, C	686.		XOR	B
637.	CB FA	SET	7, D	687.		XOR	C
638.	CB FB	SET	7, E	688.		XOR	D
639.	CB FC	SET	7, H	689.		XOR	E
640.	CB FD	SET	7, L	690.		XOR	H
641.	CB 26	SLA	(HL)	691.		XOR	L
642.	DD CB XX 26	SLA	(IX+d)	692.		XOR	n
643.	FD CB XX 26	SLA	(IY+d)			XOR	n
644.	CB 27	SLA	A				
645.	CB 20	SLA	B				
646.		SLA	C				

