プログラミングマニュアル　No.454

SHARP

Z-80
プログラミングマニュアル

アセンブリ語

SHARP

Z-80 プログラミング
マニュアル

目　　次

付　　録

第1章　序　　章

プログラムを作る場合、実際のメモリのアドレスや、複雑な機械語を使うよりも、アセンブリ語を用いる方がずっと楽に作成できる。

メモリ・アドレスをシンボリックなラベルで表わしたり、命令語をニーモニック・コードで表わしたりする。

ラベルはプログラム中の特定のステップに付けるシンボルで、プログラマがそのステップの位置を、容易に見つけ出せるようにする効果がある。

オペランドを使って命令語をより細分化し、具体的に表現することができる。メモリ、レジスタ、定数などがオペランドになる。

アセンブリ語には、機械語に対応するニーモニック以外に、アセンブラ自身に対する命令語も含まれている。

たとえば、アセンブラに指示を与える擬似命令があるが、これは機械語には変換されず、単にアセンブリ作業のための指示となる。

アセンブリ語で書いたものをソース・プログラムと呼び、多数行のステートメント（シンボリックな命令の集まり）から成っている。

各ステートメントは1ないし4の要素（ラベル、オペレーション、オペランド、コメント）から成り、1行以内で書く。

このソース・プログラムをアセンブラでアセンブルすることにより、機械語のオブジェクト・プログラムが得られ、実行ができるようになる。

第2章　アセンブリ語の説明

2.1　アセンブリ語

Ｚ－80のアセンブリ語はオペコード（オペレーション・コード）の数を少なくし、オペランドに意味を持たせるという考えで設計されていて、覚えやすく、また読みやすいという特長を持っている。

たとえば、データの転送に関しては１つのオペコードＬＤを使い、転送元（ソース）と転送先（デスティネーション）の指定はオペランドで行う。この場合、第１オペランドをデスティネーション、第２オペランドをソースとしている。

```
LD    A，B          ；A←B
```

これはレジスタＢの内容をレジスタＡに転送する命令である。

```
LD    C，３ＦＨ
```

これは値３ＦH（Hは16進数の意味）をレジスタＣにロードする命令である。

括弧は“その値に対応するメモリまたは入出力ポートの位置”という意味で使用する。

たとえば、次のような命令、

```
LD    HL，(1000)
```

はメモリ1000（10進数）番地（および1001番地）の内容を、16ビットのレジスタ・ペアＨＬにロードする命令である。

```
LD    （ＩＸ＋６），Ｂ
```

この命令はレジスタＢの内容を、16ビッドのインデックス・レジスタＩＸの現在の内容に、６を加えた値で指定されるメモリにストアするものである。

アセンブリ語の命令は覚えやすく、使いやすいものでなければならない。また、誤りの見つけやすさ、ソフトウエアの保守の容易さも重要なことである。

Ｚ－80のアセンブリ語はそれらの点について、良く配慮されている。

2.2　オペランド

オペランドはオペコードにさらに細かい意味付けをするためのもので、特定の予約語（キー・ワード）、数値、ラベルなどである。

予約語は次のとおりである。

１）８ビットのレジスタ名：Ａ，Ｂ，Ｃ，Ｄ，Ｅ，Ｈ，Ｌ，Ｉ，Ｒ

２）16ビットのレジスタ名：ＩＸ，ＩＹ，ＳＰ，ＡＦ，ＢＣ，ＤＥ，ＨＬ
（8ビットのレジスタ・ペアを含む）

3）補助レジスタ・ペア名：AF′, BC′, DE′, HL′

4）フラグの状態名：C（キャリ），NC（ノン・キャリ），Z（ゼロ），NZ（ノン・ゼロ）
M（マイナス），P（プラス），PE（パリティ偶数），PO（パリティ奇数）

a．＜オペランドの記号＞

オペランドに用いる記号を次のように定めておく。

1）r　：レジスタA，B，C，D，E，H，Lのいずれかを指す。実際の操作はそのレジスタの内容について行われる。

2）(HL)：レジスタ・ペアHLの内容で指定されるメモリを指す。実際の操作はそのメモリの内容について行われるので注意すること。

3）n　：1バイトの値で、0≦n≦255。

4）nn　：2バイトの値で、0≦n≦65535。

5）d　：1バイトの値で、−128≦d≦127。

6）(nn)　：2バイトの値nnで指定されるメモリを指す。実際の操作はそのメモリの内容について行われる。

7）(n)　：1バイトの値nで指定される入出力ポートを指す。実際の操作はその入出力ポートの内容について行われる。

8）b　：0から7までの範囲の値。

9）e　：1バイトの値で、−126≦e≦129。

10）cc　：条件ジャンプJP，CALL，RET命令などにおけるフラグの状態を指す。

11）qq　：レジスタ・ペアBC，DE，HL，AFのいずれかを指す。実際の操作はそのレジスタの内容について行われる。

ss　：レジスタ・ペアBC，DE，HL，SPのいずれかを指す。

pp　：レジスタ・ペアBC，DE，IX，SPのいずれかを指す。

rr　：レジスタ・ペアBC，DE，IY，SPのいずれかを指す。

dd　：レジスタ・ペアBC，DE，HL，SPのいずれかを指す。
（ss とは、機械語レベルで異なっている）

12）s　：r，n，(HL)，(IX+d)，(IY+d) のいずれかを指す。

13）m　：r，(HL)，(IX+d)，(IY+d) のいずれかを指す。

2.3 アセンブリ記法(シンタクス)

アセンブリ語で書くソース・プログラムは1行内に、ラベル、オペコード、オペランド、コメントのいくつかを含んでいる。

ソース・プログラムは擬似命令ORGによって、それ以降の命令群のメモリ上での位置が定められる。ORG

命令がなければ、自動的にメモリ0000番地から配列されるが、ソース・プログラムの先頭にはＯＲＧを入れた方がよい。

Ｚ－80の命令セットには、包括的な74のオペコード（ＬＤなど）と、25の予約語（Ａ，ＮＺなど）があり、オペコードとオペランドを残らず組み合わせると、６９４の命令になる。

ａ．＜アセンブラの書式＞

ソース・プログラムは一定の書式で書く必要がある。その代表的な命令行を示す。

```
ラベル   オペコード   オペランド              コメント
ＬＯＯＰ：  ＬＤ    ＢＣ，ＭＡＸ        ；ＧＥＴ，ＭＡＸ
```

ここで、ラベルＬＯＯＰはＬＯＡＤ(ＬＤ)命令の場所を特徴づけるためのもので、この命令行が置かれているアドレスを指すのに使うことができる。

オペランドとして、１ないし２個の語があり、これらは１ないしそれ以上のコンマ、またはスペースで区切る。

コメントはセミコロンで始め、ラベルは第１コラムから書き始めるか、最後にコロンを置く。

2.4 アセンブラ規約

ａ．＜ラベル＞

ラベルは16ビットまでの値を意味するシンボルであり、アドレスまたはデータに代えて用いられる。

このラベルはプログラマの識別が容易な単語あるいは文字列で書けばよい。文字列のうち、初めの6文字のみが有効で、それ以上の分は無視される。最初の文字は英文字でなければならない。

ラベルは行のどの位置から書き始めてもよいが、後尾にコロン(：)を付さねばならない。行の第１コラムから書き始めれば、コロンは省略できる。

ラベルはそれが置かれている行のロケーション値、またはそれと等価であると指定した値とともに、ラベル・テーブルに貯えられ、ラベルがオペランドとして書かれている箇所があれば、テーブルから引いてきて、その値を充当する。

ｂ．＜表現式＞

表現式は１ないし数個の項から成っていて、定数、変数、関数などが、演算子でつながれたものである。

この式はアセンブラの解読能力によって種々の制限があるが、Ｚ－80アセンブラでは、算術演算と論理演算の数種が許される。式は左から右へ順次計算され、括弧などで優先順位をつけることはできない。

演算子の種類を次に示す。

演算子	機　能
＋	加　算
－	減　算
＊	乗　算
／	除　算

&　　論理積

!　　論理和

スペースやタブのようなデリミタ(境界子、分離子)が式の中にあってもよいが、コンマは式の区切りを意味するので、使用の際には注意を要する。

演算はすべて16ビットで行われる。

８ビットの値しか許されない箇所でも、演算結果が８ビット以内であれば、16ビットの値を含んだ式を使用してもよい。この場合下位８ビットだけが採用される。

特別の場合として、JR(相対ジャンプ)あるいはDJNZ(Decrement and Jump if not Zero) 命令では、オペランドにアドレスを意味するラベルが書かれていれば、アセンブラは自動的にアドレス値の差を充当する。この差は－126から＋129までの範囲でなければならない。次に例を示す。

```
LOOP:  LD      BC, MAX   ┐
       ---               │
       ---               ├ (差) < 129バイト
       JR      C, LOOP   ┘
```

シンボル$はその時点でのアドレス値として使える。

たとえば、

```
JR    $+5
```

は、この命令のあるアドレスから５バイト後のアドレスへ、相対ジャンプする命令である。

c．＜擬似命令(アセンブラ命令)＞

擬似命令はＺ－80自身に対する通常の命令と異なり、アセンブラに対する命令である。

アセンブリ作業に指示を与えるだけで、命令そのものが、Ｚ－80の機械語に直接変換されることはない。ただ、その命令のオペランドが機械語に変換されることはある。たとえば、DEFB命令など。

アセンブラ命令を次に示す。

ＯＲＧ	ｎｎ	アドレスをｎｎにセットせよ。
ＥＱＵ	ｎｎ	この行のラベルの値をｎｎにし、プログラム内で、そのラベルがオペランドとして使われている所には、その値ｎｎを充当せよ。
ＥＮＤ		ソース・プログラムの終端とせよ。この命令をソースの最後に付しておかないと、アセンブリ作業は正しく終了しない。
ＤＥＦＢ	ｎ	この行の位置(アドレス)に、値ｎをそのままセットせよ。
ＤＥＦＢ	“Ｓ”	この行の位置(アドレス)に、１文字Ｓのアスキー・コードをセットせよ。
ＤＥＦＷ	ｎｎ	この行の位置(アドレス)と次の位置に、２バイトの値ｎｎを下位バイト、上位バイトの順にセットせよ。
ＤＥＦＳ	ｎｎ	この行の位置(アドレス)から、ｎｎバイト分だけ、メモリ領域を確保せよ。
ＤＥＦＭ	“Ｓ”	この行の位置(アドレス)から、文字列Ｓをアスキー・コードでセットしていけ。文字列Ｓの文字数は１から63までの範囲である。

注）“各行の位置”は、アドレスを割り振っていくためにアセンブラ自身が持っているリファレンス・カウンタの内容に対応している。

d．＜デリミタ(境界子)＞

デリミタはソース・プログラム行内の各要素の区切りとして用いられ、コンマ、スペース、タブなどである。

通常、スペース、タブは区切りとなるが、表現式の中の演算子の前後では無視される。

e．＜コメント(注釈)＞

コメントはプログラムの見やすさ、ソフトウエアの保守の容易さを得るために、ソース・プログラムに記入しておく説明書きであり、オブジェクト・プログラムには出力されない。

コメントは初めにセミコロンを付せば、行のどの位置からでも書き始められる。

第3章　Z－80　CPU　フラグ

フラグ・レジスタ（FとF′）は、適時、プログラマが、CPUのステータス（状態）を調べることができるように設けられている。

各フラグのビット位置は次のとおりである。

7	6	5	4	3	2	1	0
S	Z	X	H	X	P/V	N	C

C　：キャリ・フラグ
N　：加/減算フラグ
P/V　：パリティ/オーバ・フロー・フラグ
H　：ハーフ・キャリ・フラグ
Z　：ゼロ・フラグ
S　：サイン・フラグ
X　：未使用

これらのフラグはCPUの動作により、セットあるいはリセットされる。

C，P／V，Z，Sはプログラマが、条件ジャンプ、コールなどの命令によって調べることができるが、BCD演算に用いられるH，Nは直接調べることはできない。

a．＜キャリ・フラグ C＞

キャリ・フラグのセット、リセットは行われる演算操作によって異なる。

“ADD”命令でキャリが出た場合、“SUB”命令でボローが出た場合に、キャリ・フラグはセットされる。それぞれ、キャリ、ボローが出なければ、キャリ・フラグはリセットされてしまう。

“DAA”命令で10進補正を行う条件が揃ったときに、キャリ・フラグはセットされる。

“RLA，RRA，RLS，RRS”命令において、キャリ・フラグはリンク(連結)の中の1ビットとして含められる。

“RLCA，RLC，SLA”命令において、キャリ・フラグに、レジスタ、メモリのビット7の内容がシフトされてきて残る。

“RRCA，RRC，SRA，SRL”命令において、キャリ・フラグに、レジスタ、メモリのビット0の内容がシフトされてきて残る。

“AND，OR，XOR”命令で、キャリ・フラグはリセットされる。

キャリ・フラグは“SCF”命令でセットされ、“CCF”命令で反転する。

b．＜加/減算フラグ N＞

このフラグは“ＤＡＡ”命令を実行する際にチェックされる。

“ＡＤＤ”命令で“０”にリセットされ、“ＳＵＢ”命令で“１”にセットされる。

c．＜パリティ/オーバ・フロー・フラグ P/V＞

算術演算において、アキュムレータに置かれるべき結果が、－128から＋127までの範囲を超えていれば、このフラグがオーバ・フローによりセットされる。２の補数表現においてオーバ・フローが生じた場合にセットされる。

以下に、このフラグがセットまたはリセットされる条件について述べる。

１）たがいに符号の異なる値の加算ではリセットされる。

２）たがいに同じ符号の値の加算で、結果が異符号になれば、セットされる。

	デシマル	バイナリ
例	＋１２０	０１１１　１０００
（＋）	＋１０５	０１１０　１００１
	－　３１	１１１０　０００１（OVER）

３）たがいに同じ符号の値の減算ではリセットされる。

４）たがいに異なる符号の値の減算では、各値の大きさによって異なる。

次のような場合、セットされる。

	デシマル	バイナリ
例	＋１２７	０１１１　１１１１
（－）	－　６４	１１００　０００
	－　６５	１０１１　１１１１（OVER）

このＰ／Ｖフラグはまた、論理演算、ローテイト命令において、結果のパリティ（１バイト中の“１”の数）を調べるのに使用される。

１の総数が奇数ならば、Ｐ＝０（パリティ奇数）となり、総数が偶数ならば、Ｐ＝１（パリティ偶数）となる。

“ＣＰＩ，ＣＰＩＲ，ＣＰＤ，ＣＰＤＲ”のサーチ命令や、“ＬＤＩ，ＬＤＩＲ，ＬＤＤ，ＬＤＤＲ”のようなブロック転送命令の実行時、このＰ／Ｖフラグはバイト・カウンタ（ＢＣ）の状態をモニタしていて、バイト・カウンタが零でない場合、フラグは“１”で、カウンタが零になったとき“０”となる。

“ＬＤ Ａ，Ｉ”と“ＬＤ Ａ，Ｒ”命令の実行時、ＩＦＦ２（割り込みイネーブル・フリップ・フロップ2）の内容が、このＰ／Ｖフラグに移されるので、ＩＦＦ２の内容をセーブしたり、テストしたりできる。

入出力デバイスから、“ＩＮ ｒ，（Ｃ）”命令で、１バイトずつ読み込む場合、このＰ／Ｖフラグはデータのパリティに応じて、セットかリセットされる。

d．＜ハーフ・キャリ・フラグ H＞

ハーフ・キャリ・フラグは、８ビットの算術演算でのビット３とビット４のあいだのキャリ、ボローの有無によって、セットかリセットされる。

このフラグはＢＣＤの加算あるいは減算の結果を、“ＤＡＡ”命令で補正する際に利用される。

キャリ、ボローがあれば“１”にセットされ、なければ“０”にリセットされる。

e．＜ゼロ・フラグ Z＞

ゼロ・フラグは、ある命令の実行の結果が零であるか否かによって、セットかリセットされる。

８ビットの算術、論理演算の結果、アキュムレータの内容が零ならば、フラグは“１”にセットされ、零でなければ、“０”にリセットされる。

サーチ命令において、アキュムレータの値とレジスタ・ペアＨＬで指定されたメモリの値とが一致した場合、ゼロ・フラグは“１”にセットされる。

ビット・テスト命令において、指定されたビットの補数値が、このゼロ・フラグに置かれる。

入出力命令“ＩＮＩ，ＩＮＤ，ＯＵＴＩ，ＯＵＴＤ”において、入出力データＢから１を減じた値、Ｂ－１が零ならば、ゼロ・フラグはセットされ、零でなければ、リセットされる。

また、“ＩＮ r,（Ｃ）”命令では、入力データが零ならば、フラグはセットされる。

f．＜サイン・フラグ(符号)S＞

サイン・フラグはアキュムレータのビット７の内容を保持し、演算の際に利用する。

符号付演算では、２の補数を使い、ビット７が“０”ならば正数、ビット７が“１”ならば負数とする。

正の数の大きさは７ビット分（０～ 127）で、負の数の大きさは、同じく、７ビット分(－1～－128)で与えられる。

入力命令“ＩＮ r,（Ｃ）”で、データを読み込む場合、サイン・フラグに、データの正、負がそれぞれ“０”、“１”でセットされる。

第4章　Z－80命令セット

Z－80の命令グループ

1）8ビット　ロード　グループ

2）16ビット　ロード　グループ

3）交換、ブロック転送、サーチ　グループ

4）8ビット　算術、論理演算　グループ

5）汎用算術演算、ＣＰＵ制御　グループ

6）16ビット　算術演算　グループ

7）ローテイト、シフト　グループ

8）ビット操作（セット、リセット、テスト）　グループ

9）ジャンプ　グループ

10）コール、リターン　グループ

11）入出力　グループ

注）命令の説明において、次のように定めておく。

実行サイクル、ステート、時間として、

Mサイクル＝ i　　；i個のマシン・サイクル

Tステート＝ j（$m_0+m_1+\cdots+m_i$）；j個のTステート（各マシン・サイクルにおけるTステートの総和）

実行時間＝ tμs（f_C）　　；実行時間〔Tステートの総数とクロック周期$1/f_C$との積〕

（クロック周波数）

フラグとは、条件フラグF（F′）の内容を意味する。

ＣＹ（キャリ）はＣと略記することもある。

8ビット　ロード　グループ

LD r, r′ ; r←r′

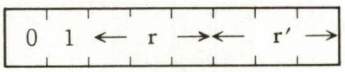
（オペランド r, r′）

レジスタr′の内容をレジスタrにロードする。

r, r′とオブジェクト・コードは次のとおり。

レジスタ		r, r′
A	=	1 1 1
B	=	0 0 0
C	=	0 0 1
D	=	0 1 0
E	=	0 1 1
H	=	1 0 0
L	=	1 0 1

オブジェクト・コード



実　行 ： Mサイクル：1，Tステート：4，実行時間：1.00μs（4MHz）

フラグ ： 変化せず。

例　 ： レジスタEの内容が10Hのとき

```
LD H, E
```

を実行すると、レジスタHの内容も10Hとなる。

LD r, n ; r←n

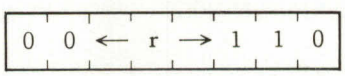
（オペランド r, n）

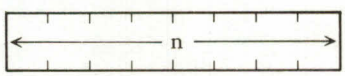
8ビット整数nをレジスタrにロードする。

rとオブジェクト・コードは次のとおり。

レジスタ		r
A	=	1 1 1
B	=	0 0 0
C	=	0 0 1
D	=	0 1 0
E	=	0 1 1
H	=	1 0 0
L	=	1 0 1

オブジェクト・コード





実　行 ： Mサイクル：2，Tステート：7（4+3），実行時間：1.75μs（4MHz）

フラグ ： 変化せず。

例　 ： レジスタEの内容が何であっても、

```
LD E, 5AH
```

を実行すると、レジスタEの内容は5AHとなる。

LD r,(HL)　　; r←(HL)

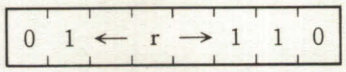
（オペランド　r，（HL））

レジスタ・ペアHLの内容で指定されるメモリの8ビットの内容をレジスタrにロードする。

rとオブジェクト・コードは次のとおり。

レジスタ	r
A	= 111
B	= 000
C	= 001
D	= 010
E	= 011
H	= 100
L	= 101

オブジェクト・コード



実　行：Mサイクル：2，Tステート：7(4+3)，実行時間：1.75μs（4MHz）

フラグ：変化せず。

例　：レジスタ・ペアHLの内容が202AHで、メモリの202AH番地の内容が58Hのとき

LD C, (HL)

を実行すると、レジスタCの内容も58Hとなる。

LD r,(IX+d)　　; r←(IX+d)

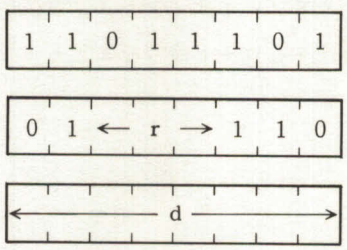
（オペランド　r，（IX+d））

インデックス・レジスタIXの内容にディスプレイスメントdを加えた値で指定されるメモリの8ビットの内容をレジスタrにロードする。

rとオブジェクト・コードは次のとおり。

レジスタ	r
A	= 111
B	= 000
C	= 001
D	= 010
E	= 011
H	= 100
L	= 101

オブジェクト・コード



実　行：Mサイクル：5，Tステート：19(4+4+3+5+3)，実行時間：4.75μs（4MHz）

フラグ：変化せず。

例　：インデックス・レジスタIXの内容が25AFHのとき、ディスプレイスメントdを19Hとすると

LD B, (IX+19H)

を実行すると、メモリの25C8H番地の内容がレジスタBにロードされる。
ディスプレイスメントdは直接指定できるが、インデックス・レジスタＩＸの内容は後述するロード命令でまえもって指定しておく必要がある。

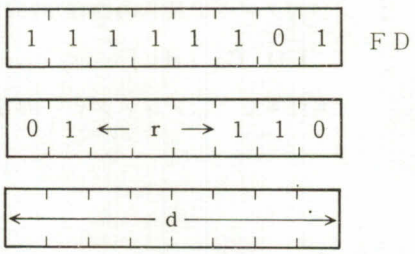
ＬＤ　r,(ＩＹ＋ｄ)　；　r←(ＩＹ＋ｄ)

（オペランド　r，(IY＋d)）

前述のＬＤ r,（IX＋ｄ）と同様で、もう１つのインデックス・レジスタＩＹの内容に、ディスプレイスメントｄを加えた値で指定されるメモリの８ビットの内容を、レジスタｒにロードする。
ｒとオブジェクト・コードは次のとおり。

レジスタ		r
A	＝	1 1 1
B	＝	0 0 0
C	＝	0 0 1
D	＝	0 1 0
E	＝	0 1 1
H	＝	1 0 0
L	＝	1 0 1

オブジェクト・コード



実　行：Mサイクル：５，Ｔステート：19(4＋4＋3＋5＋3)，実行時間：4.75μS（4MHz）
フラグ：変化せず。
例　：前述のLD r,（IX＋ｄ）の例と同様で、インデックス・レジスタＩＹの内容が25AFHのとき、ディスプレイスメントｄを19Hとすると、

```
LD B, (IY＋19H)
```

を実行すると、メモリの25C8H番地の内容がレジスタBにロードされる。
ディスプレイスメントｄは直接指定できるが、インデックス・レジスタＩＹの内容は後述するロード命令でまえもって指定しておく必要がある。

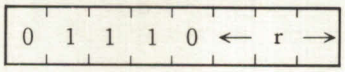
ＬＤ　(ＨＬ)，ｒ　　；(ＨＬ)←ｒ

（オペランド（HL)，ｒ）

レジスタｒの内容をレジスタ・ペアＨＬの内容で指定されるメモリにロードする。
ｒとオブジェクト・コードは次のとおり。

レジスタ		r
A	=	1 1 1
B	=	0 0 0
C	=	0 0 1
D	=	0 1 0
E	=	0 1 1
H	=	1 0 0
L	=	1 0 1

オブジェクト・コード



実　行 ： Mサイクル：2，Tステート：7(4+3)，実行時間：1.75μs(4MHz)

フラグ ： 変化せず。

例　　： レジスタ・ペアＨＬの内容が0055Hで、レジスタＢの内容が3AHのとき

```
LD (HL), B
```

を実行すると、メモリの0055H番地の内容も3AHとなる。

LD (IX+d), r ; (IX+d) ←r

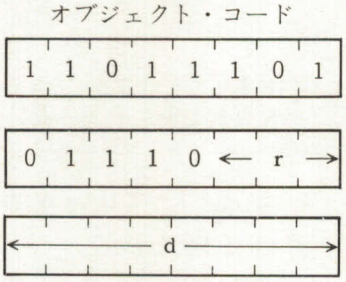
（オペランド（IX＋d），r）

レジスタrの内容を、インデックス・レジスタＩＸの内容にディスプレイスメントdを加えた値で指定されるメモリにロードする。

rとオブジェクト・コードは次のとおり。

レジスタ		r
A	=	1 1 1
B	=	0 0 0
C	=	0 0 1
D	=	0 1 0
E	=	0 1 1
H	=	1 0 0
L	=	1 0 1

オブジェクト・コード



実　行 ： Mサイクル：5，Tステート：19(4+4+3+5+3)，実行時間：4.75μs(4MHz)

フラグ ： 変化せず。

例　　： レジスタＣの内容が3BHで、インデックス・レジスタＩＸの内容が2100Hのとき

```
LD (IX+5H), C
```

を実行すると、メモリの2105H番地の内容も3BHとなる。

LD (IY＋d), r ; (IY+d)←r

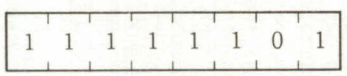
（オペランド（IY＋d), r）

レジスタrの内容を、インデックス・レジスタIYの内容にディスプレイスメントdを加えた値で指定されるメモリにロードする。

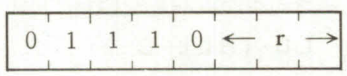
rとオブジェクト・コードは次のとおり。

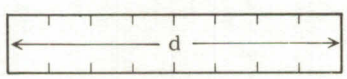
レジスタ	r
A	111
B	000
C	001
D	010
E	011
H	100
L	101

オブジェクト・コード



FD





実　行：Mサイクル：5，Tステート：19(4+4+3+5+3)，実行時間：4.75μs（4MHz）

フラグ：変化せず。

例　：レジスタDの内容が4CHで、インデックス・レジスタIYの内容が3200Hのとき、

LD (IY＋6H), D

を実行すると、メモリの3206H番地の内容も4CHとなる。

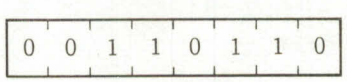
LD (HL), n ; (HL)←n

（オペランド（HL), n）

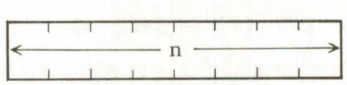
8ビット整数nをレジスタ・ペアHLの内容で指定されるメモリにロードする。

オブジェクト・コードは次のとおり。

オブジェクト・コード



36



実　行：Mサイクル：3，Tステート：10(4+3+3)，実行時間：2.50μs（4MHz）

フラグ：変化せず。

例　：レジスタ・ペアＨＬの内容が7700Hのとき、

LD (HL), 77H

を実行すると、メモリの7700H番地の内容が77Hとなる。

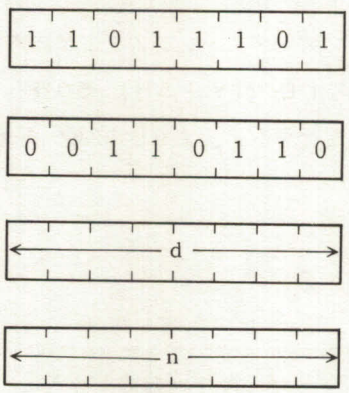
LD (IX＋d), n　；(IX+d) ←n

（オペランド（IX＋d），n）

８ビット整数nを、インデックス・レジスタＩＸの内容にディスプレイスメントｄを加えた値で指定されるメモリにロードする。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行：Mサイクル：5，Tステート：19(4＋4＋3＋5＋3)，実行時間：4.75μs (4MHz)

フラグ：変化せず。

例　：インデックス・レジスタＩＸの内容が3300Hのとき、

LD (IX＋5H), 50H

を実行すると、メモリの3305H番地の内容が50Hとなる。

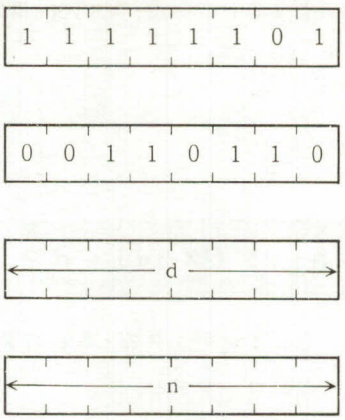
LD (IY＋d), n　；(IY+d) ←n

（オペランド（IY＋d），n）

８ビット整数nを、インデックス・レジスタＩＹの内容にディスプレイスメントｄを加えた値で指定されるメモリにロードする。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行 ： Mサイクル：5，Tステート：19(4+4+3+5+3)，実行時間：4.75μs (4MHz)

フラグ ： 変化せず。

例　 ： インデックス・レジスタＩＹの内容が3300Hのとき、

```
LD (IY+5H), 50H
```

を実行すると、メモリの3305H番地の内容が50Hとなる。

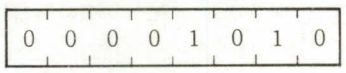
LD A,(BC)　；A←(BC)

（オペランド A,（ＢＣ））

レジスタ・ペアＢＣの内容で指定されるメモリの内容をアキュムレータにロードする。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行 ： Mサイクル：2，Tステート：7（4＋3），実行時間：1.75μs (4MHz)

フラグ ： 変化せず。

例　 ： レジスタ・ペアＢＣの内容が200AHで、メモリの200AH番地の内容が59Hのとき、

```
LD A, (BC)
```

を実行すると、アキュムレータの内容も59Hとなる。

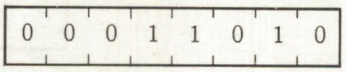
LD A, (DE) ; A←(DE)

（オペランド A, (DE)）

レジスタ・ペアDEの内容で指定されるメモリの内容をアキュムレータにロードする。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行 ： Mサイクル：2，Tステート：7（4＋3），実行時間：1.75μs（4MHz）

フラグ ： 変化せず。

例　 ： レジスタ・ペアDEの内容が201AHで、メモリの201AH番地の内容が5AHのとき、

```
LD A, (DE)
```

を実行すると、アキュムレータの内容も5AHとなる。

LD A, (nn) ; A← (nn)

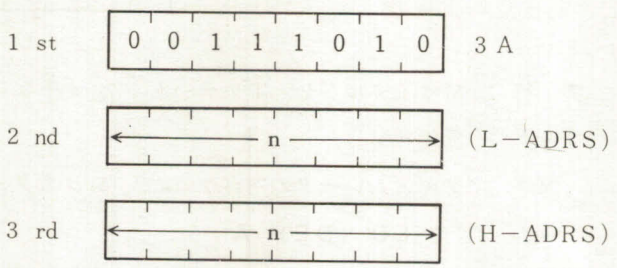
（オペランド A, (nn)）

16ビット整数nnで指定されるメモリの内容をアキュムレータにロードする。

オブジェクト・コードの2バイト目に番地の下位バイトが入り、3バイト目に上位バイトが入る。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行 ： Mサイクル：4，Tステート：13(4+3+3+3)，実行時間：3.25μs（4MHz）

フラグ ： 変化せず。

例　 ： メモリの4455H番地の内容が10Hのとき、

```
LD A, (4455H)
```

を実行すると、アキュムレータの内容も10Hとなる。

LD (BC), A ; (BC)←A

（オペランド（ＢＣ），Ａ）

アキュムレータの内容をレジスタ・ペアＢＣの内容で指定されるメモリにロードする。
オブジェクト・コードは次のとおり。

オブジェクト・コード

0	0	0	0	0	0	1	0	02

実　行：Mサイクル：2，Tステート：7（4＋3），実行時間：1.75μs（4MHz）
フラグ：変化せず。
例　：アキュムレータの内容が66Hで、レジスタ・ペアＢＣの内容が300AHのとき、

```
LD (BC), A
```

を実行すると、メモリの300AH番地の内容も66Hとなる。

LD (DE), A ; (DE)←A

（オペランド（ＤＥ），Ａ）

アキュムレータの内容をレジスタ・ペアＤＥの内容で指定されるメモリにロードする。
オブジェクト・コードは次のとおり。

オブジェクト・コード

0	0	0	1	0	0	1	0	12

実　行：Mサイクル：2，Tステート：7（4＋3），実行時間：1.75μs（4MHz）
フラグ：変化せず。
例　：アキュムレータの内容が67Hで、レジスタ・ペアＤＥの内容が301AHのとき、

```
LD (DE), A
```

を実行すると、メモリの301AH番地の内容も67Hとなる。

LD (nn), A ; (nn)←A

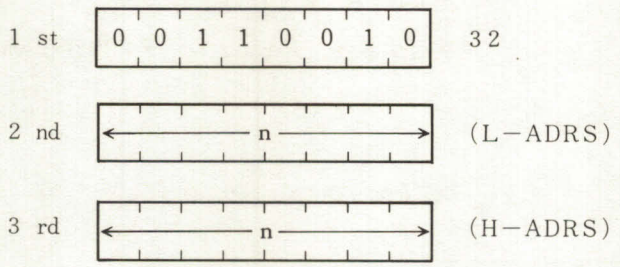
（オペランド（nn），A）

アキュムレータの内容を16ビット整数nnで指定されるメモリにロードする。

オブジェクト・コードの2バイト目に番地の下位バイトが入り、3バイト目に上位バイトが入る。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行：Mサイクル：4，Tステート：13(4+3+3+3)，実行時間：3.25μs (4MHz)

フラグ：変化せず。

例　：アキュムレータの内容が11Hのとき、

```
LD (4456H), A
```

を実行すると、メモリの4456H番地の内容も11Hとなる。

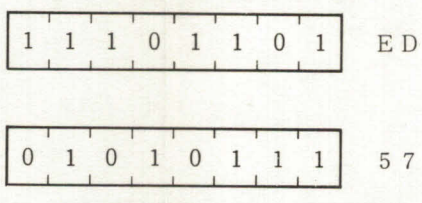
LD A, I ; A←I

（オペランド A，I）

割り込みベクトル・レジスタIの内容をアキュムレータにロードする。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行：Mサイクル：2，Tステート：9（4＋5），実行時間：2.25μs (4MHz)

フラグ：　S　：Iが負ならばセット、他の場合はリセット。

Z　：　Iが零ならばセット、他の場合はリセット。

H　：　リセット。

P／V：　IFF$_2$の内容を保持。

N　：　リセット。

C　：　変化せず。

例　：　割り込みベクトル・レジスタIの内容が30Hのとき、

```
LD A, I
```

を実行すると、アキュムレータの内容も30Hとなる。

LD I, A　　; I←A

（オペランド I, A）

アキュムレータの内容を割り込みベクトル・レジスタIにロードする。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	0	1	1	0	1	ED

0	1	0	0	0	1	1	1	47

実　行：　Mサイクル：2，Tステート：9（4＋5），実行時間：2.25μs（4MHz）

フラグ：　変化せず。

例　：　アキュムレータの内容が31Hのとき、

```
LD I, A
```

を実行すると、割り込みベクトル・レジスタIの内容も31Hとなる。

LD A, R　　; A←R

（オペランド A, R）

メモリ・リフレッシュ・レジスタRの内容をアキュムレータにロードする。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	0	1	1	0	1	ED

0	1	0	1	1	1	1	1	5F

実　行：Mサイクル：2，Tステート：9(4＋5)，実行時間：2.25μs (4MHz)

フラグ：　S　：Rが負ならばセット、他の場合はリセット。

Z　：Rが零ならばセット、他の場合はリセット。

H　：リセット。

P／V：IFF_2の内容を保持。

N　：リセット。

C　：変化せず。

例　：メモリ・リフレッシュ・レジスタRの内容が32Hのとき、

```
LD A, R
```

を実行すると、アキュムレータの内容も32Hとなる。

LD R, A　　; R←A

(オペランド R, A)

アキュムレータの内容をメモリ・リフレッシュ・レジスタRにロードする。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	0	1	1	0	1	ED

0	1	0	0	1	1	1	1	4F

実　行：Mサイクル：2，Tステート：9(4＋5)，実行時間：2.25μs (4MHz)

フラグ：変化せず。

例　：アキュムレータの内容が33Hのとき、

LD R, A

を実行すると、メモリ・リフレッシュ・レジスタRの内容も33Hとなる。

—— 16ビット　ロード　グループ ——

LD dd, nn ; dd←nn

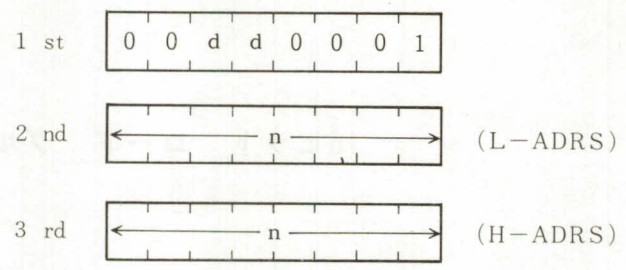
（オペランド dd, nn）

2バイトの整数をレジスタ・ペアddにロードする。

ddとオブジェクトコードは次のとおり。

レジスタ・ペア		dd
BC	=	00
DE	=	01
HL	=	10
SP	=	11

オブジェクト・コード



実　行：Mサイクル：3，Tステート：10(4+3+3)，実行時間：2.50μs（4MHz）

フラグ：変化せず。

例　：レジスタ・ペアHLの内容が何であっても、

```
LD HL, 1000H
```

を実行すると、レジスタ・ペアHLの内容は1000Hとなる。

LD IX, nn ; IX←nn

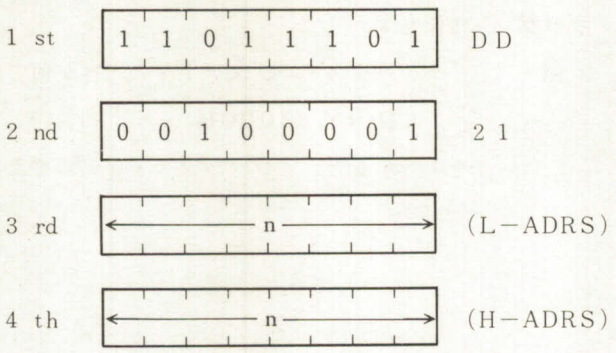
（オペランド IX, nn）

2バイトの整数nnをインデックス・レジスタIXにロードする。

オブジェクト・コードの3バイト目にデータの下位バイトを入れ、4バイト目に上位バイトを入れる。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行　：　Mサイクル：4，Tステート：14(4+4+3+3)，実行時間：3.50μs(4MHz)

フラグ　：　変化せず。

例　　：　インデックス・レジスタＩＸの内容が何であっても、

```
LD IX, 2000H
```

を実行すると、インデックス・レジスタＩＸの内容は2000Hとなる。

LD　IY, nn　　; IY←nn

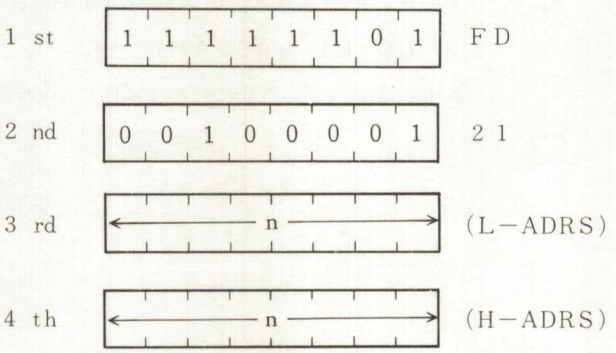
（オペランド　IY, nn）

2バイトの整数ｎｎをインデックス・レジスタＩＹにロードする。

オブジェクト・コードの3バイト目にデータの下位バイトを入れ、4バイト目に上位バイトを入れる。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行 ： Mサイクル：4，Tステート：14(4+4+3+3)，実行時間：3.50μs（4MHz）

フラグ ： 変化せず。

例　 ： インデックス・レジスタＩＹの内容が何であっても、

```
LD IY, 3000H
```

を実行すると、インデックス・レジスタＩＹの内容は3000Hとなる。

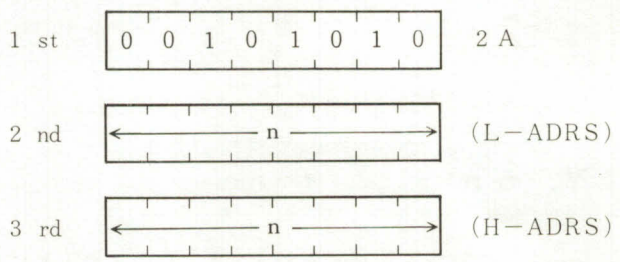
LD HL, (nn)　　; L← (nn), H← (nn+1)　（オペランド HL,(ｎｎ)）

メモリのｎｎ番地とｎｎ＋１番地の内容をレジスタ・ペアＨＬのＬとＨにロードする。

オブジェクト・コードの２バイト目に番地の下位バイトを入れ、３バイト目に上位バイトを入れる。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行 ： Mサイクル：5，Tステート：16(4+3+3+3+3)，実行時間：4.00μs（4MHz）

フラグ ： 変化せず。

例　 ： メモリの2AH番地と2BH番地の内容がそれぞれ12Hと34Hのとき、

```
LD HL, (002AH)
```

を実行すると、レジスタ・ペアＨＬの内容は3412Hとなる。

LD dd, (nn) ; $dd_L \leftarrow (nn)$, $dd_H \leftarrow (nn+1)$

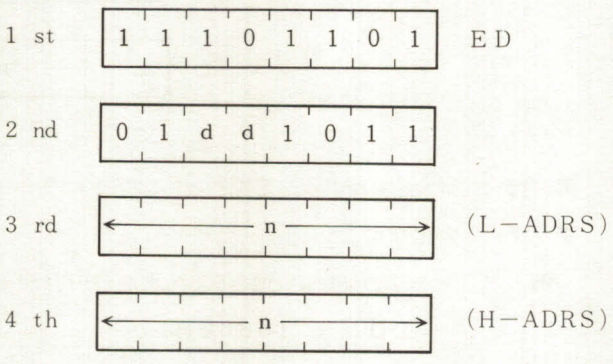
（オペランド dd, (nn)）

メモリのnn番地とnn＋1番地の内容を、レジスタ・ペアの下位側dd_Lと上位側dd_Hにロードする。オブジェクト・コードの3バイト目に番地の下位バイトを入れ、4バイト目に上位バイトを入れる。

ddとオブジェクト・コードは次のとおり。

レジスタ・ペア		dd
BC	=	00
DE	=	01
HL	=	10
SP	=	11

オブジェクト・コード



実　行：Mサイクル：6，Tステート：20(4＋4＋3＋3＋3＋3)，実行時間：5.00 μs (4MHz)

フラグ：変化せず。

例　：メモリの$2A0_H$番地と$2A1_H$番地の内容がそれぞれ34_Hと56_Hのとき、

```
LD BC, (02A0H)
```

を実行すると、レジスタ・ペアBCの内容は5634_Hとなる。

LD IX, (nn) ; $IX_L \leftarrow (nn)$, $IX_H \leftarrow (nn+1)$

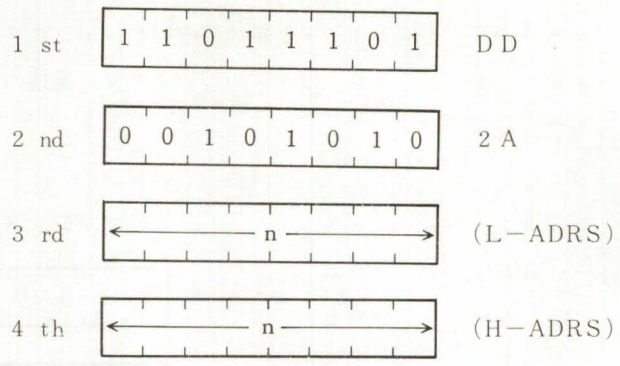
（オペランドIX, (nn)）

メモリのnn番地とnn＋1番地の内容を、インデックス・レジスタの下位側IX_Lと上位側IX_Hにロードする。

オブジェクト・コードの3バイト目に番地の下位バイトを入れ、4バイト目に上位バイトを入れる。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行：Mサイクル：6，Tステート：20(4+4+3+3+3+3)，実行時間：5.00μs(4MHz)

フラグ：変化せず。

例　：メモリの6666H番地と6667H番地の内容がそれぞれ67Hと45Hのとき、

```
LD IX, (6666H)
```

を実行すると、インデックス・レジスタＩＸの内容は4567Hとなる。

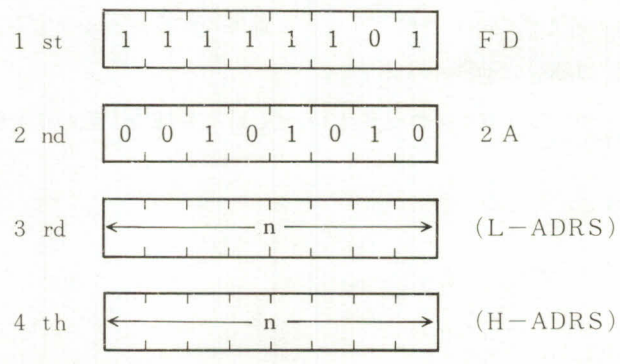
LD IY, (nn)　；IY_L←(nn), IY_H←(nn+1)　（オペランドＩＹ，（ｎｎ））

メモリのｎｎ番地とｎｎ＋１番地の内容を、インデックス・レジスタの下位側IY_Lと上位側IY_Hにロードする。

オブジェクト・コードの３バイト目に番地の下位バイトを入れ、４バイト目に上位バイトを入れる。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行　：　Mサイクル：6，Tステート：20(4+4+3+3+3+3)，実行時間：5.00μs(4MHz)

フラグ　：　変化せず。

例　　：　メモリの6677H番地と6678H番地の内容がそれぞれ78Hと56Hのとき、

```
LD IY, (6677H)
```

を実行すると、インデックス・レジスタＩＹの内容は5678Hとなる。

LD (nn), HL ; (nn)←L, (nn+1)←H

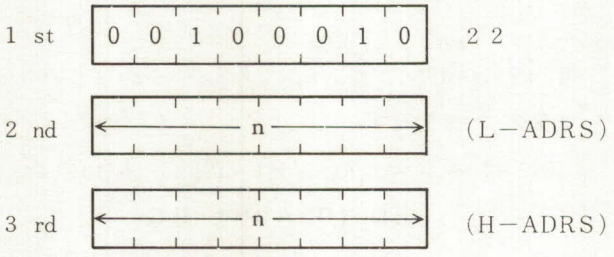
（オペランド (nn), HL）

レジスタ・ペアＨＬのＬとＨの内容を、それぞれメモリｎｎ番地とｎｎ＋１番地にロードする。

オブジェクト・コードの２バイト目に番地の下位バイトを入れ、３バイト目に上位バイトを入れる。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行　：　Mサイクル：5，Tステート：16(4+3+3+3+3)，実行時間：4.00μs(4MHz)

フラグ　：　変化せず。

例　　：　レジスタ・ペアＨＬの内容が3412Hのとき、

```
LD (002AH), HL
```

を実行すると、メモリの2AH番地と2BH番地の内容は、それぞれ12Hと34Hとなる。

LD (nn), dd ; (nn)←dd$_L$, (nn+1)←dd$_H$

（オペランド（nn），dd）

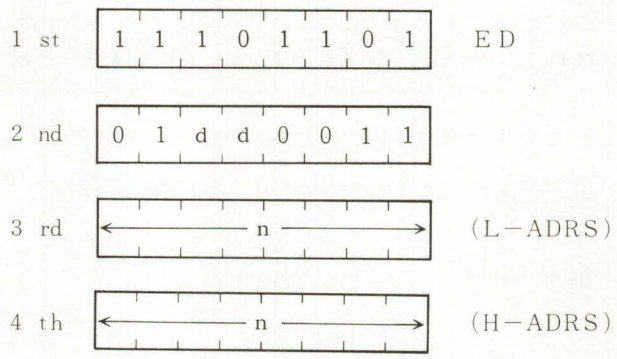
レジスタ・ペアddの下位側dd$_L$と上位側dd$_H$の内容を、それぞれメモリnn番地とnn＋1番地にロードする。

オブジェクト・コードの3バイト目に番地の下位バイトを入れ、4バイト目に上位バイトを入れる。

ddとオブジェクト・コードは次のとおり。

レジスタ・ペア		dd
BC	＝	00
DE	＝	01
HL	＝	10
SP	＝	11

オブジェクト・コード



実　行：Mサイクル：6，Tステート：20(4＋4＋3＋3＋3＋3)，実行時間：5.00μs（4MHz）

フラグ：変化せず。

例　：レジスタ・ペアBCの内容が5634$_H$のとき、

```
LD (02A0H), BC
```

を実行すると、メモリの02A0$_H$番地と02A1$_H$番地の内容はそれぞれ34$_H$と56$_H$となる。

LD (nn), IX ; (nn)←IX$_L$, (nn+1)←IX$_H$

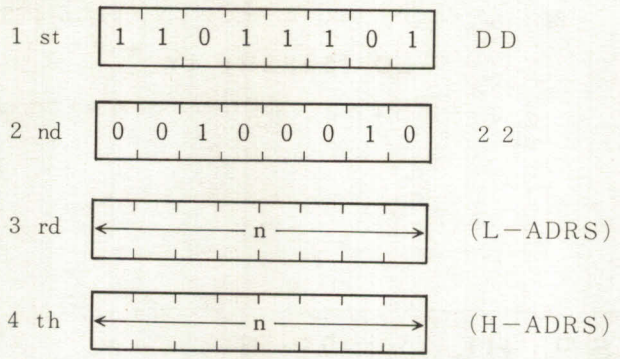
（オペランド（nn），IX）

インデックス・レジスタIXの下位側IX$_L$と上位側IX$_H$の内容を、それぞれメモリnn番地とnn＋1番地にロードする。

オブジェクト・コードの3バイト目に番地の下位バイトを入れ、4バイト目に上位バイトを入れる。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行：Mサイクル：6，Tステート：20(4+4+3+3+3+3)，実行時間：5.00μs (4MHz)

フラグ：変化せず。

例　：インデックス・レジスタＩＸの内容が$2A3B_H$のとき、

```
LD (4321H), IX
```

を実行すると、メモリの4321_H番地と4322_H番地の内容はそれぞれ$3B_H$と$2A_H$となる。

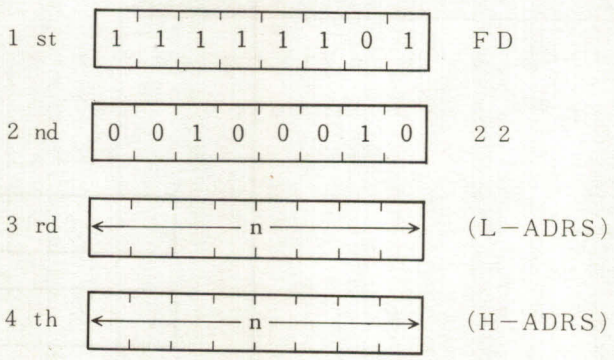
LD (nn), IY　　;(nn)←IY_L, (nn+1)←IY_H　　(オペランド (nn), IY)

インデックス・レジスタＩＹの下位側IY_Lと上位側IY_Hの内容を、それぞれメモリｎｎ番地とｎｎ＋１番地にロードする。

オブジェクト・コードの３バイト目に番地の下位バイトを入れ、４バイト目に上位バイトを入れる。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行　：　Mサイクル：6，Tステート：20(4+4+3+3+3+3)，実行時間：5.00μs(4MHz)

フラグ　：　変化せず。

例　　：　インデックス・レジスタＩＹの内容が4C5DHのとき、

```
LD (5432H), IY
```

を実行すると、メモリの5432H番地と5433H番地の内容はそれぞれ5DHと4CHとなる。

LD SP, HL　；SP←HL

（オペランド ＳＰ，ＨＬ）

レジスタ・ペアＨＬの内容をスタック・ポインタＳＰにロードする。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	1	1	0	0	1	F9

実　行　：　Mサイクル：1，Tステート：6，実行時間：1.50μs（4MHz）

フラグ　：　変化せず。

例　　：　レジスタペアＨＬの内容が1112Hのとき、

```
LD SP, HL
```

を実行すると、スタック・ポインタＳＰの内容も1112Hとなる。

LD SP, IX　；SP←IX

（オペランド ＳＰ，ＩＸ）

インデックス・レジスタＩＸの内容をスタック・ポインタＳＰにロードする。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	0	1	1	1	0	1	DD

1	1	1	1	1	0	0	1	F9

実　行：Mサイクル：2，Tステート：10（4+6），実行時間：2.50μs（4MHz）

フラグ：変化せず。

例　：インデックス・レジスタＩＸの内容が2222_Hのとき、

```
LD SP, IX
```

を実行すると、スタック・ポインタＳＰの内容も2222_Hとなる。

LD SP, IY　；SP←IY

（オペランド ＳＰ，ＩＹ）

インデックス・レジスタＩＹの内容をスタック・ポインタＳＰにロードする。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	1	1	1	0	1	FD
1	1	1	1	1	0	0	1	F9

実　行：Mサイクル：2，Tステート：10（4+6），実行時間：2.50μs（4MHz）

フラグ：変化せず。

例　：インデックス・レジスタＩＹの内容が3222_Hのとき、

```
LD SP, IY
```

を実行すると、スタック・ポインタＳＰの内容も3222_Hとなる。

PUSH qq　；(SP−1)←q q_H, (SP−2)←q q_L

（オペランドｑｑ）

レジスタ・ペアｑｑの内容を外部メモリ・スタックにプッシュする。まずＳＰの内容を１だけ減じて、ｑｑの上位側の内容をＳＰの内容で指定されるスタックへ入れ、次に下位側の内容を次のスタックに入れる。

スタック・ポインタＳＰの内容は自動的に２だけ減少する。

q qとオブジェクト・コードは次のとおり。

レジスタ・ペア		q q
BC	=	0 0
DE	=	0 1
HL	=	1 0
AF	=	1 1

オブジェクト・コード

1	1	q	q	0	1	0	1

実　行 ： Mサイクル：3，Tステート：11（5+3+3），実行時間：2.75 μs（4MHz）

フラグ ： 変化せず。

例　　： レジスタ・ペアAFの内容が1122Hで、スタック・ポインタSPの内容が1002Hのとき、

```
PUSH AF
```

を実行すると、メモリの1001H番地と1000H番地の内容はそれぞれ11Hと22Hとなり、スタック・ポインタSPの内容は1000Hとなる。

PUSH IX ; $(SP-1) \leftarrow IX_H$, $(SP-2) \leftarrow IX_L$ （オペランドIX）

インデックス・レジスタIXの内容を外部メモリ・スタックにプッシュする。まずSPの内容を1だけ減じて、IXの上位側の内容をSPの内容で指定されるスタックへ入れ、次に下位側の内容を次のスタックに入れる。スタック・ポインタSPの内容は自動的に2だけ減少する。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	0	1	1	1	0	1	DD
1	1	1	0	0	1	0	1	E5

実　行 ： Mサイクル：4，Tステート：15(4+5+3+3)，実行時間：3.75 μs（4MHz）

フラグ ： 変化せず。

例　　： インデックス・レジスタIXの内容が2233Hで、スタック・ポインタSPの内容が2002Hのとき、

```
PUSH IX
```

を実行すると、メモリの2001H番地と2000H番地の内容はそれぞれ22Hと33Hとなり、スタック・ポインタSPの内容は2000Hとなる。

PUSH IY　　; (SP−1) ← IY_H, (SP−2) ← IY_L　　（オペランドＩＹ）

インデックス・レジスタＩＹの内容を外部メモリ・スタックにプッシュする。まずＳＰの内容を１だけ減じて、ＩＹの上位側の内容をＳＰの内容で指定されるスタックへ入れ、次に下位側の内容を次のスタックに入れる。スタック・ポインタＳＰの内容は自動的に２だけ減少する。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	1	1	1	0	1	FD
1	1	1	0	0	1	0	1	E5

実　行 ： Mサイクル：4，Tステート：15(4+5+3+3)，実行時間：3.75μs (4MHz)

フラグ ： 変化せず。

例　 ： インデックス・レジスタＩＹの内容が3344Hで、スタック・ポインタＳＰの内容が3002Hのとき、

PUSH IY

を実行すると、メモリの3001H番地と3000H番地の内容はそれぞれ33Hと44Hとなり、スタック・ポインタＳＰの内容は3000Hとなる。

POP qq　　; qq$_L$ ← (SP), qq$_H$ ← (SP+1)　　（オペランド qq）

外部メモリ・スタックの内容をレジスタ・ペアqqにポップする。まず、スタック・ポインタＳＰの内容で指定されるメモリの内容をqqの下位側に、次にＳＰの内容を１だけ増して、その内容で指定されるメモリの内容をqqの上位側にロードする。ＳＰの内容は最終的に２だけ増加する。qqとオブジェクト・コードは次のとおり。

レジスタ・ペア		qq
ＢＣ	＝	00
ＤＥ	＝	01
ＨＬ	＝	10
ＡＦ	＝	11

オブジェクト・コード

1	1	q	q	0	0	0	1

実　行　：　Mサイクル：3，Tステート：10（4+3+3），実行時間：2.50μs（4MHz）

フラグ　：　変化せず。

例　：　スタック・ポインタＳＰの内容が1000Hで、メモリ1000H番地と1001H番地の内容がそれぞれ55Hと44Hのとき、

```
POP HL
```

を実行すると、レジスタ・ペアＨＬの内容は4455Hとなり、スタック・ポインタＳＰの内容は1002Hとなる。

POP IX　；$IX_L \leftarrow (SP)$, $IX_H \leftarrow (SP+1)$　（オペランドＩＸ）

外部メモリ・スタックの内容をインデックス・レジスタＩＸにポップする。まず、スタック・ポインタＳＰの内容で指定されるメモリの内容をＩＸの下位側に、次にＳＰの内容を１だけ増し、その内容で指定されるメモリの内容をＩＸの上位側にロードする。ＳＰの内容は最終的に２だけ増加する。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	0	1	1	1	0	1	DD

1	1	1	0	0	0	0	1	E1

実　行　：　Mサイクル：4，Tステート：14（4+4+3+3），実行時間：3.50μs（4MHz）

フラグ　：　変化せず。

例　：　スタック・ポインタＳＰの内容が2000Hで、メモリの2000H番地と2001H番地の内容がそれぞれ66Hと55Hのとき、

```
POP IX
```

を実行すると、インデックス・レジスタＩＸの内容は5566Hとなり、スタック・ポインタＳＰの内容は2002Hとなる。

POP IY　　; $IY_L \leftarrow (SP)$, $IY_H \leftarrow (SP+1)$　　（オペランドＩＹ）

外部メモリ・スタックの内容をインデックス・レジスタＩＹにポップする。まず、スタック・ポインタＳＰの内容で指定されるメモリの内容をＩＹの下位側に、次にＳＰの内容を１だけ増し、その内容で指定されるメモリの内容をＩＹの上位側にロードする。ＳＰの内容は最終的には２だけ増加する。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	1	1	1	0	1	FD
1	1	1	0	0	0	0	1	E1

実　行　：　Mサイクル：４，Ｔステート：14（4+4+3+3），実行時間：3.50μs（4MHz）

フラグ　：　変化せず。

例　　：　スタック・ポインタＳＰの内容が3000Hで、メモリの3000H番地と3001H番地の内容がそれぞれ77Hと66Hのとき、

```
POP IY
```

を実行すると、インデックス・レジスタＩＹの内容は6677Hとなり、スタック・ポインタＳＰの内容は3002Hとなる。

―― 交換、ブロック転送、サーチ　グループ ――

EX DE, HL ; DE↔HL

（オペランド DE, HL）

レジスタ・ペアＤＥとＨＬの２バイトの内容を交換する。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	0	1	0	1	1

ＥＢ

実　行 ： Mサイクル：１，Ｔステート：４，実行時間：1.00 μs（4MHz）

フラグ ： 変化せず。

例　 ： レジスタ・ペアＤＥとＨＬの内容がそれぞれ0022Hと1100Hのとき、

EX DE, HL

を実行すると、レジスタ・ペアＤＥとＨＬの内容はそれぞれ1100Hと0022Hとなる。

EX AF, AF′ ; AF↔AF′

（オペランド AF, AF′）

レジスタ・ペアＡＦとＡＦ′（実際はＡ′とＦ′とから成る）の２バイトの内容を交換する。

オブジェクト・コードは次のとおり。

オブジェクト・コード

0	0	0	0	1	0	0	0

０８

実　行 ： Mサイクル：１，Ｔステート：４，実行時間：1.00 μs（4MHz）

フラグ ： 変化せず。

例　 ： レジスタ・ペアＡＦとＡＦ′の内容がそれぞれ0020Hと0001Hのとき、

EX AF, AF′

を実行すると、レジスタ・ペアＡＦとＡＦ′の内容はそれぞれ0001Hと0020Hとなる。

EXX　；BC↔BC′, DE↔DE′, HL↔HL′

（オペランド　なし）

レジスタ・ペアＢＣ、ＤＥ、ＨＬの内容をそれぞれＢＣ′、ＤＥ′、ＨＬ′の内容と交換する。
レジスタをたがいに置き換えると考えてもよい。
オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	0	1	1	0	0	1

Ｄ９

実　行：Ｍサイクル：１，Ｔステート：４，実行時間：1.00μs（4MHz）
フラグ：変化せず。
例　：レジスタ・ペアの内容が次のとき、

ＢＣ：００１０Ｈ　　ＢＣ′：００１１Ｈ
ＤＥ：００２０Ｈ　　ＤＥ′：００２１Ｈ
ＨＬ：００３０Ｈ　　ＨＬ′：００３１Ｈ

EXX

を実行すると、各レジスタ・ペアの内容は次のようになる。

ＢＣ：００１１Ｈ　　ＢＣ′：００１０Ｈ
ＤＥ：００２１Ｈ　　ＤＥ′：００２０Ｈ
ＨＬ：００３１Ｈ　　ＨＬ′：００３０Ｈ

EX　(SP), HL　；L↔(SP), H↔(SP+1)

（オペランド（SP), HL）

レジスタ・ペアＨＬの内容をスタック・ポインタＳＰで指定されるメモリの内容と交換する。
オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	0	0	0	1	1

Ｅ３

実　行：Ｍサイクル：５，Ｔステート：19(4+3+4+3+5)，実行時間：4.75μs（4MHz）
フラグ：変化せず。

例　：レジスタ・ペアＨＬの内容が6677Hで、スタック・ポインタＳＰの内容が0818H、メモリの0818H番地の内容が99H、メモリの0819H番地の内容が00Hのとき、

ＥＸ　(ＳＰ)，　ＨＬ

を実行すると、レジスタ・ペアＨＬの内容は0099Hとなり、メモリの0818H番地の内容は77H、メモリの0819H番地の内容は66Hとなる。

スタック・ポインタＳＰの内容は変化しない。

ＥＸ　(ＳＰ)，　ＩＸ　　；$IX_L \leftrightarrow (SP)$，$IX_H \leftrightarrow (SP+1)$

（オペランド（ＳＰ)，ＩＸ）

インデックス・レジスタＩＸの内容とスタック・ポインタの内容で指定されるメモリの内容を交換する。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	0	1	1	1	0	1	ＤＤ
1	1	1	0	0	0	1	1	Ｅ３

実　行　：　Mサイクル：６，Ｔステート：23(4+4+3+4+3+5)，実行時間：5.75μs(4MHz)

フラグ　：　変化せず。

例　：　各レジスタ、メモリの内容が次のとき、

ＩＸ　：４４５５H

ＳＰ　：０１００H

０１００H　：　８８H

０１０１H　：　６６H

ＥＸ　(ＳＰ)，　ＩＸ

を実行すると、各レジスタ、メモリの内容は次のようになる。

ＩＸ　：６６８８H

ＳＰ　：０１００H

０１００H　：　５５H

０１０１H　：　４４H

EX (SP), IY ; $IY_L \leftrightarrow (SP)$, $IY_H \leftrightarrow (SP+1)$

（オペランド（SP), IY）

インデックス・レジスタＩＹの内容とスタック・ポインタＳＰの内容で指定されるメモリの内容を交換する。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	1	1	1	0	1	FD

1	1	1	0	0	0	1	1	E3

実　行：Mサイクル：6，Tステート：23(4+4+3+4+3+5)，実行時間:5.75 μs（4MHz）

フラグ：変化せず。

例　：各レジスタ、メモリの内容が次のとき、

ＩＹ：5566H

ＳＰ：0200H

0200H：0099H

0201H：0077H

EX (SP), IY

を実行すると、各レジスタ、メモリの内容は次のようになる。

ＩＹ：7799H

ＳＰ：0200H

0200H：0066H

0201H：0055H

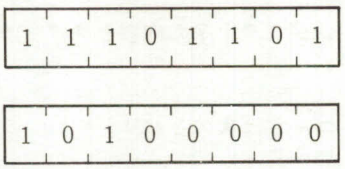
LDI ;(DE)←(HL), DE←DE+1, HL←HL+1, BC←BC−1

（オペランド なし）

レジスタ・ペアＨＬの内容で指定されるメモリの内容を、レジスタ・ペアＤＥの内容で指定されるメモリへ転送する。その後、レジスタ・ペアＨＬ，ＤＥの内容を１だけ増して、レジスタ・ペアＢＣの内容を１だけ減ずる。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行 ： Mサイクル：4，Tステート：16(4+4+3+5)，実行時間：4.00μs(4MHz)

フラグ ：　S　：変化せず。

Z　：変化せず。

H　：リセット。

P／V：BC－1キ0ならばセット、他の場合はリセット。

N　：リセット。

C　：変化せず。

例　：各レジスタ、メモリの内容が次のとき、

HL　：1111H

DE　：2222H

BC　：　6H

1111H 番地：　88H

2222H 番地：　55H

LDI

を実行すると、各レジスタ、メモリの内容は次のようになる。

HL　：1112H

DE　：2223H

BC　：　5H

1111H 番地：　88H

2222H 番地：　88H

ＬＤＩＲ ；(ＤＥ)←(ＨＬ), ＤＥ←ＤＥ＋１, ＨＬ←ＨＬ＋１, ＢＣ←ＢＣ－１ (オペランド　なし)

レジスタ・ペアＨＬの内容で指定されるメモリの内容を、レジスタ・ペアＤＥの内容で指定されるメモリへ転送する。その後、レジスタ・ペアＨＬ，ＤＥの内容を１だけ増し、レジスタ・ペアＢＣの内容を１だけ減じる。そして、この操作をＢＣの内容が零になるまで繰り返す。

ＢＣの内容が零ならば、次に再び零になるまで、64Ｋバイト分を実行し終わるまで繰り返す。

この命令の実行中割り込みは受け付けられる。

各データの転送ごとに２つのリフレッシュ・サイクルを実行する。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	0	1	1	0	1	ＥＤ
1	0	1	1	0	0	0	0	Ｂ０

実　行 ： ＢＣ≠０のとき：(１データ転送当たり)

Ｍサイクル：５，Ｔステート：21(4+4+3+5+5)，実行時間：5.25μs(4MHz)

ＢＣ＝０のとき：

Ｍサイクル：４，Ｔステート：16(4+4+3+5)，実行時間：4.00μs(4MHz)

フラグ ：　Ｓ　：変化せず。

Ｚ　：変化せず。

Ｈ　：リセット。

Ｐ／Ｖ：リセット。

Ｎ　：リセット。

Ｃ　：変化せず。

例　：各レジスタ、メモリの内容が次のとき、

ＨＬ：１１１１Ｈ	１１１１Ｈ 番地：９９Ｈ	２２２２Ｈ 番地：ＣＣＨ
ＤＥ：２２２２Ｈ	１１１２Ｈ 番地：ＡＡＨ	２２２３Ｈ 番地：ＤＤＨ
ＢＣ：　　３Ｈ	１１１３Ｈ 番地：ＢＢＨ	２２２４Ｈ 番地：ＥＥＨ

```
LDIR
```

を実行すると、各レジスタ、メモリの内容は次のようになる。

HL：1114H　　1111H 番地：99H　　2222H 番地：99H
DE：2225H　　1112H 番地：AAH　　2223H 番地：AAH
BC：　　0H　　1113H 番地：BBH　　2224H 番地：BBH

LDD　;(DE)←(HL), DE←DE−1, HL←HL−1, BC←BC−1　（オペランド　なし）

レジスタ・ペアHLの内容で指定されるメモリの内容を、レジスタ・ペアDEの内容で指定されるメモリへ転送して、レジスタ・ペアDE，HL，BCの内容を1だけ減じる。
オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	0	1	1	0	1	ED
1	0	1	0	1	0	0	0	A8

実　行：Mサイクル：4，Tステート：16(4+4+3+5)，実行時間：4.00μs(4MHz)
フラグ：　S　：変化せず。
　　Z　：変化せず。
　　H　：リセット。
　　P／V：BC−1≠0ならばセット、他の場合はリセット。
　　N　：リセット。
　　C　：変化せず。
例　：各レジスタ、メモリの内容が次のとき、
　　HL：1111H　　1111H 番地：55H
　　DE：2222H　　2222H 番地：66H
　　BC：　　3H
　　LDD
　　を実行すると、各レジスタ、メモリの内容は次のようになる。
　　HL：1110H　　1111H 番地：55H
　　DE：2221H　　2222H 番地：55H
　　BC：　　2H

LDDR ；(DE)←(HL), DE←DE－1, HL←HL－1, BC←BC－1 （オペランド　なし）

レジスタ・ペアＨＬの内容で指定されるメモリの内容を、レジスタ・ペアＤＥの内容で指定されるメモリへ転送する。その後、レジスタ・ペアＨＬ、ＤＥの内容を１だけ減じ、レジスタ・ペアＢＣの内容を１だけ減じる。そして、同じ操作をＢＣの内容が零になるまで繰り返す。

ＢＣの内容が始めから零の場合、次に再び零になるまで、64Ｋバイト分を実行し終えるまで繰り返す。

この命令の実行中割り込みは受け付けられ、各データの転送ごとに２つのリフレッシュ・サイクルを実行する。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	0	1	1	0	1	ED
1	0	1	1	1	0	0	0	B8

実　行：ＢＣ≠０のとき：

Ｍサイクル：５，Ｔステート：21(4+4+3+5+5)，実行時間：5.25μs（4MHz）

ＢＣ＝０のとき：

Ｍサイクル：４，Ｔステート：16(4+4+3+5)，実行時間：4.00μs（4MHz）

フラグ：　Ｓ　：変化せず。

Ｚ　：変化せず。

Ｈ　：リセット。

Ｐ／Ｖ：リセット。

Ｎ　：リセット。

Ｃ　：変化せず。

例　：各レジスタ、メモリの内容が次のとき、

ＨＬ：１１１１Ｈ	１１１１Ｈ 番地：Ａ２Ｈ	２２２２Ｈ 番地：９０Ｈ
ＤＥ：２２２２Ｈ	１１１０Ｈ 番地：Ｂ３Ｈ	２２２１Ｈ 番地：Ａ１Ｈ
ＢＣ：　　３Ｈ	１１０ＦＨ 番地：Ｃ４Ｈ	２２２０Ｈ 番地：Ｂ２Ｈ

LDDR

を実行すると、各レジスタ、メモリの内容は次のようになる。

HL：１１０EH　　１１１１H 番地：Ａ２H　　２２２２H 番地：Ａ２H

DE：２２１FH　　１１１０H 番地：Ｂ３H　　２２２１H 番地：Ｂ３H

BC：　　０H　　１１０FH 番地：Ｃ４H　　２２２０H 番地：Ｃ４H

CPI　；A−(HL)，HL←HL＋1，BC←BC−1

（オペランド　なし）

レジスタ・ペアＨＬの内容で指定されるメモリの内容とアキュムレータの内容を比較する。等しい場合、フラグをセットする。その後、ＨＬの内容を１だけ増し、バイト・カウンタＢＣの内容を１だけ減じる。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	0	1	1	0	1	ED
1	0	1	0	0	0	0	1	A1

実　行：Mサイクル：４，Ｔステート：16（4+4+3+5），実行時間：4.00μs（4MHz）

フラグ：　Ｓ　：結果が負ならばセット、他の場合はリセット。

Ｚ　：Ａ＝（ＨＬ）ならばセット、他の場合はリセット。

Ｈ　：ビット４からのボローがあればセット、他の場合はリセット。

Ｐ／Ｖ：ＢＣ－１≠０ならばセット、他の場合はリセット。

Ｎ　：セット。

Ｃ　：変化せず。

例　：各レジスタ、メモリの内容が次のとき、

ＨＬ：１１１２H　　１１１２H 番地：３BH

Ａ　：　３BH

ＢＣ：０００１H

CPI

を実行すると、各レジスタ、メモリの内容は次のようになる。

ＨＬ：１１１３Ｈ　　１１１２Ｈ 番地：３ＢＨ

Ａ　：　３ＢＨ

ＢＣ：００００Ｈ

Ｚ　：セット。

Ｐ／Ｖ：リセット。

CPIR　；A－(HL)，HL←HL＋1，BC←BC－1

（オペランド　なし）

レジスタ・ペアＨＬの内容で指定されるメモリの内容とアキュムレータの内容を比較する。等しい場合、フラグをセットする。その後、ＨＬの内容を１だけ増し、バイト・カウンタＢＣの内容を１だけ減じる。比較した結果が等しい（Ａ＝(ＨＬ)）か、ＢＣの内容が零になるまで、同じ操作を繰り返す。

比較結果が一致した場合、すでにＨＬは＋１、ＢＣは－１されている。

ＢＣの内容が始めから零で、途中比較の結果、一致がない場合、64Ｋバイト分繰り返される。

この命令の実行中割り込みは受け付けられ、各データの転送ごとに２つのリフレッシュ・サイクルを実行する。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	0	1	1	0	1	ED
1	0	1	1	0	0	0	1	B1

実　行　：　ＢＣ≠０で、Ａ≠(ＨＬ)のとき：

Ｍサイクル：５，Ｔステート：21(4+4+3+5+5)，実行時間：5.25μs（4MHz）

ＢＣ＝０か、Ａ＝(ＨＬ)のとき：

Ｍサイクル：４，Ｔステート：16(4+4+3+5)，実行時間：4.00μs（4MHz）

フラグ　：　Ｓ　：結果が負ならばセット、他の場合はリセット。

Ｚ　：Ａ＝(ＨＬ)ならばセット、他の場合はリセット。

Ｈ　：ビット４からのボローがあればセット、他の場合はリセット。

Ｐ／Ｖ：ＢＣ－１≠０ならばセット、他の場合はリセット。

N　：セット。

C　：変化せず。

例　：各レジスタ、メモリの内容が次のとき、

HL：1111H　　1111H 番地：52H

A　：　F3H　　1112H 番地：00H

BC：0007H　　1113H 番地：F3H

CPIR

を実行すると、各レジスタ、メモリの内容は次のようになる。

HL：1114H　　Z　：セット　　1111H 番地：変化せず

A　：　F3H　　P／V：セット　　1112H 番地：変化せず

BC：0004H　　1113H 番地：変化せず

CPD　；A−(HL)，HL←HL−1，BC←BC−1

（オペランド　なし）

レジスタ・ペアHLの内容で指定されるメモリの内容とアキュムレータの内容を比較する。等しい場合、フラグをセットする。その後、HLとバイト・カウンタBCの内容を1だけ減じる。オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	0	1	1	0	1	ED
1	0	1	0	1	0	0	1	A9

実　行：Mサイクル：4，Tステート：16(4+4+3+5)，実行時間：4.00μs（4MHz）

フラグ：　S　：結果が負ならばセット、他の場合はリセット。

Z　：A＝(HL)ならばセット、他の場合はリセット。

H　：ビット4からのボローがあればセット、他の場合はリセット。

P／V：BC−1≠0ならばセット、他の場合はリセット。

N　：セット。

C　：変化せず。

例　：各レジスタ、メモリの内容が次のとき、

HL：1111H　　1111H 番地：3BH

A　：　3BH

BC：0001H

CPD

を実行すると、各レジスタ、メモリの内容は次のようになる。

HL：1110H　　Z　：セット　　1111H 番地：3BH

A　：　3BH　　P／V：リセット

BC：0000H

CPDR　；A−(HL)，HL←HL−1，BC←BC−1

（オペランド　なし）

レジスタ・ペアＨＬの内容で指定されるメモリの内容とアキュムレータの内容を比較する。

等しい場合、フラグをセットする。その後、ＨＬとバイト・カウンタＢＣの内容を１だけ減じる。

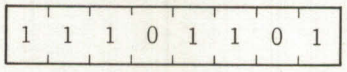
そして、ＢＣの内容が零になるか、比較した結果が等しく（Ａ＝(ＨＬ)）なるまで、同じ操作を繰り返す。

始めからＢＣの内容が零であって、途中Ａ＝(ＨＬ)になることがなければ、この操作を64Ｋバイト分繰り返す。

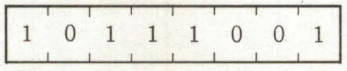
この命令の実行中割り込みは受け付けられ、各データの転送ごとに２つのリフレッシュ・サイクルを実行する。

オブジェクト・コードは次のとおり。

オブジェクト・コード



ED



B9

実　行：ＢＣ≠０で、Ａ≠(ＨＬ)のとき：

Mサイクル：５，Ｔステート：21(4+4+3+5+5)，実行時間：5.25μs（4MHz）

ＢＣ＝０か、Ａ＝(ＨＬ)のとき：

Mサイクル：４，Ｔステート：16(4+4+3+5)，実行時間：4.00μs（4MHz）

フラグ ：　S　：結果が負ならばセット、他の場合はリセット。

Z　：A＝(HL)ならばセット、他の場合はリセット。

H　：ビット4からのボローがあればセット、他の場合はリセット。

P／V：BC－1≠0ならばセット、他の場合はリセット。

N　：セット。

C　：変化せず。

例　：各レジスタ、メモリの内容が次のとき、

HL：1118H　　1118H 番地：53H

A　：　F3H　　1117H 番地：01H

BC：0007H　　1116H 番地：F3H

CPDR

を実行すると、各レジスタ、メモリの内容は次のようになる。

HL：1115H　　Z　：セット　　1118H 番地：変化せず

A　：　F3H　　P／V：セット　　1117H 番地：変化せず

BC：0004H　　1116H 番地：変化せず

——— 8ビット　算術、論理演算　グループ ———

ADD A, r ; A←A+r

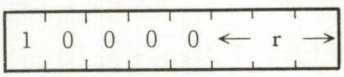
（オペランド A, r）

レジスタrの内容とアキュムレータの内容とを加えて、その結果をアキュムレータにストアする。

rとオブジェクト・コードは次のとおり。

レジスタ	r
A =	111
B =	000
C =	001
D =	010
E =	011
H =	100
L =	101

オブジェクト・コード



実　行：Mサイクル：1，Tステート：4，実行時間：1.00μs（4MHz）

フラグ：　S　：結果が負ならばセット、他の場合はリセット。

Z　：結果が零ならばセット、他の場合はリセット。

H　：ビット3からのキャリがあればセット、他の場合はリセット。

P／V：オーバフローがあればセット、他の場合はリセット。

N　：リセット。

C　：ビット7からのキャリがあればセット、他の場合はリセット。

例　：各レジスタの内容が次のとき、

A：44H

C：11H

ADD A, C

を実行すると、各レジスタの内容は次のようになる。

A：55H

C：11H

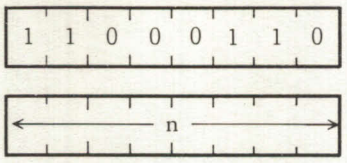
ADD A, n ; A←A+n

（オペランド A, n）

整数nとアキュムレータの内容を加え、その結果をアキュムレータにストアする。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行：Mサイクル：2，Tステート：7(4＋3)，実行時間：1.75μs (4MHz)

フラグ：　S　：結果が負ならばセット、他の場合はリセット。

Z　：結果が零ならばセット、他の場合はリセット。

H　：ビット3からのキャリがあればセット、他の場合はリセット。

P／V：オーバフローがあればセット、他の場合はリセット。

N　：リセット。

C　：ビット7からのキャリがあればセット、他の場合はリセット。

例　：アキュムレータの内容が2Hのとき、

ADD A, 33H

を実行すると、アキュムレータの内容は35Hとなる。

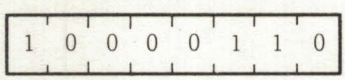
ADD A,(HL)　；A←A+(HL)

(オペランド　A,(HL))

レジスタ・ペアHLの内容で指定されるメモリの1バイトの内容とアキュムレータの内容とを加え、結果をアキュムレータにストアする。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行：Mサイクル：2，Tステート：7(4＋3)，実行時間：1.75μs (4MHz)

フラグ：　S　：結果が負ならばセット、他の場合はリセット。

Z　：結果が零ならばセット、他の場合はリセット。

H　：ビット3からのキャリがあればセット、他の場合はリセット。

P／V：オーバフローがあればセット、他の場合はリセット。

N　：リセット。

C　：ビット7からのキャリがあればセット、他の場合はリセット。

例　：各レジスタ、メモリの内容が次のとき、

A　：A0H　　2233H番地：0AH

HL：2233H

ADD A, (HL)

を実行すると、アキュムレータの内容はAAHとなる。

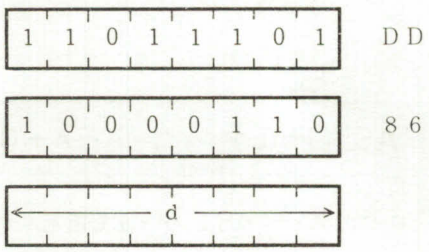
ADD A, (IX+d)　；A←A+(IX+d)

（オペランド　A, (IX+d)）

インデックス・レジスタIXの内容に、ディスプレイスメントdを加えた値で指定されるメモリの内容とアキュムレータの内容とを加え、その結果をアキュムレータにストアする。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行：Mサイクル：5，Tステート：19(4＋4＋3＋5＋3)，実行時間：4.75μs(4MHz)

フラグ：　S　：結果が負ならばセット、他の場合はリセット。

Z　：結果が零ならばセット、他の場合はリセット。

H　：ビット3からのキャリがあればセット、他の場合はリセット。

P／V：オーバフローがあればセット、他の場合はリセット。

N　：リセット。

C　：ビット7からのキャリがあればセット、他の場合はリセット。

例　：各レジスタ、メモリの内容が次のとき

A　：34H　　2005H番地：02H

IX：2000H

ADD A, (IX＋5H)

を実行すると、アキュムレータの内容は36Hとなる。

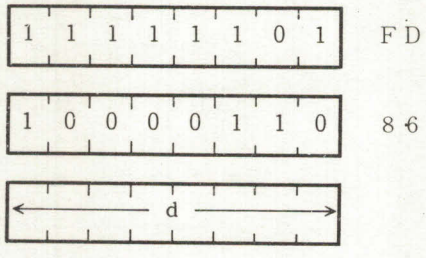
ADD A, (IY+d) ; A←A+(IY+d)

（オペランド A, (IY＋d)）

インデックス・レジスタIYの内容に、ディスプレイスメントdを加えた値で指定されるメモリの内容とアキュムレータの内容とを加え、その結果をアキュムレータにストアする。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行：Mサイクル：5，Tステート：19(4＋4＋3＋5＋3)，実行時間：4.75μs(4MHz)

フラグ：　S　：結果が負ならばセット、他の場合はリセット。

Z　：結果が零ならばセット、他の場合はリセット。

H　：ビット3からのキャリがあればセット、他の場合はリセット。

P／V：オーバフローがあればセット、他の場合はリセット。

N　：リセット。

C　：ビット7からのキャリがあればセット、他の場合はリセット。

例　：各レジスタ、メモリの内容が次のとき、

A　：34H　　2005H番地：02H

IY：2000H

ADD A, (IY＋5H)

を実行すると、アキュムレータの内容は36Hとなる。

ADC A, s ; A←A+s+CY

（オペランド A, s）

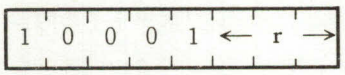
オペランドsの内容に、キャリフラグの内容とアキュムレータの内容とを加え、その結果をアキュムレータにストアする。

オペランドsにはr、n、(HL)、(IX+d)、(IY+d)が用いられる。

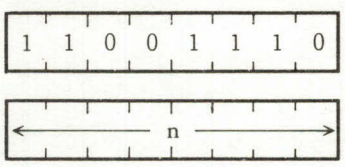
rとオブジェクト・コードは次のとおり。

オブジェクト・コード

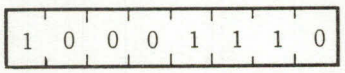
ADC A, r



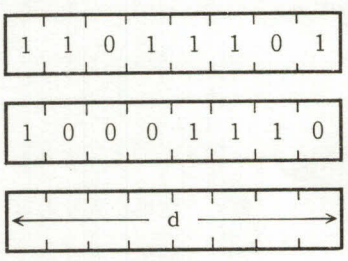
ADC A, n



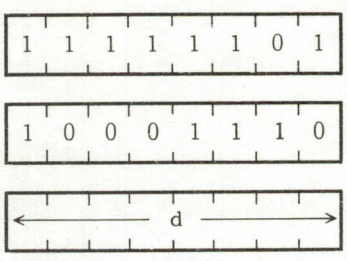
ADC A, (HL)



ADC A, (IX+d)



ADC A, (IY+d)



レジスタ		r
B	=	0 0 0
C	=	0 0 1
D	=	0 1 0
E	=	0 1 1
H	=	1 0 0

L ＝ １０１

A ＝ １１１

実　行	命　令	Mサイクル	Tステート	実行時間(4MHz)
	ＡＤＣ　Ａ，ｒ	1	4	1.00μs
	ＡＤＣ　Ａ，ｎ	2	7（４＋３）	1.75μs
	ＡＤＣ　Ａ，（ＨＬ）	2	7（４＋３）	1.75μs
	ＡＤＣ　Ａ，（ＩＸ＋ｄ）	5	19（４＋４＋３＋５＋３）	4.75μs
	ＡＤＣ　Ａ，（ＩＹ＋ｄ）	5	19（４＋４＋３＋５＋３）	4.75μs

フラグ：　Ｓ　：結果が負ならばセット、他の場合はリセット。

Ｚ　：結果が零ならばセット、他の場合はリセット。

Ｈ　：ビット３からのキャリがあればセット、他の場合はリセット。

Ｐ／Ｖ：オーバフローがあればセット、他の場合はリセット。

Ｎ　：リセット。

Ｃ　：ビット７からのキャリがあればセット、他の場合はリセット。

例　：各レジスタ、メモリの内容が次のとき、

Ａ：１５H　　キャリフラグ：セット

ＨＬ：５５５５H　　５５５５H番地：２０H

ＡＤＣ　Ａ，（ＨＬ）

を実行すると、アキュムレータの内容は36Hとなる。

ＳＵＢ　ｓ　；Ａ←Ａ－ｓ

（オペランド　ｓ）

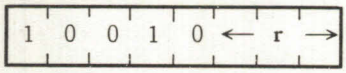
アキュムレータの内容からオペランドｓの内容を減じ、その結果をアキュムレータにストアする。

オペランドｓにはｒ、ｎ、（ＨＬ）、（ＩＸ＋ｄ）、（ＩＹ＋ｄ）が用いられる。

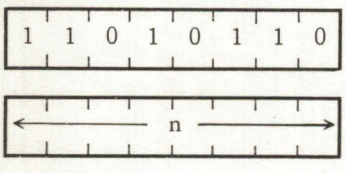
ｒとオブジェクト・コードは次のとおり。

オブジェクト・コード

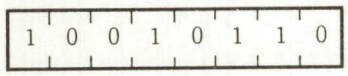
ＳＵＢ　ｒ



ＳＵＢ　ｎ

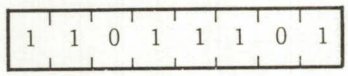


SUB （HL）

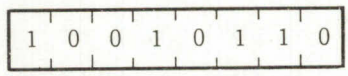


96

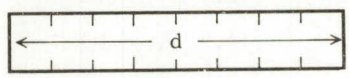
SUB （IX+d）



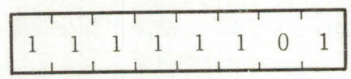
DD



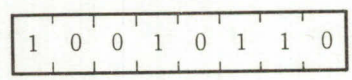
96



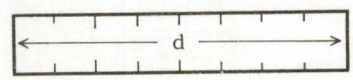
SUB （IY+d）



FD



96



レジスタ	r
B ＝	0 0 0
C ＝	0 0 1
D ＝	0 1 0
E ＝	0 1 1
H ＝	1 0 0
L ＝	1 0 1
A ＝	1 1 1

実　行：

命　令	Mサイクル	Tステート	実行時間(4MHz)
SUB　r	1	4	1.00μs
SUB　n	2	7（4＋3）	1.75μs
SUB　（HL）	2	7（4＋3）	1.75μs
SUB　（IX+d）	5	19（4＋4＋3＋5＋3）	4.75μs
SUB　（IY+d）	5	19（4＋4＋3＋5＋3）	4.75μs

フラグ：

S　：結果が負ならばセット、他の場合はリセット。

Z　：結果が零ならばセット、他の場合はリセット。

H　：ビット4からのボローがあればセット、他の場合はリセット。

P／V：オーバフローがあればセット、他の場合はリセット。

N　：セット。

C　：ボローがあればセット、他の場合はリセット。

例　：アキュムレータの内容が38H、レジスタDの内容が21Hのとき、

SUB D

を実行すると、アキュムレータの内容は17Hとなる。

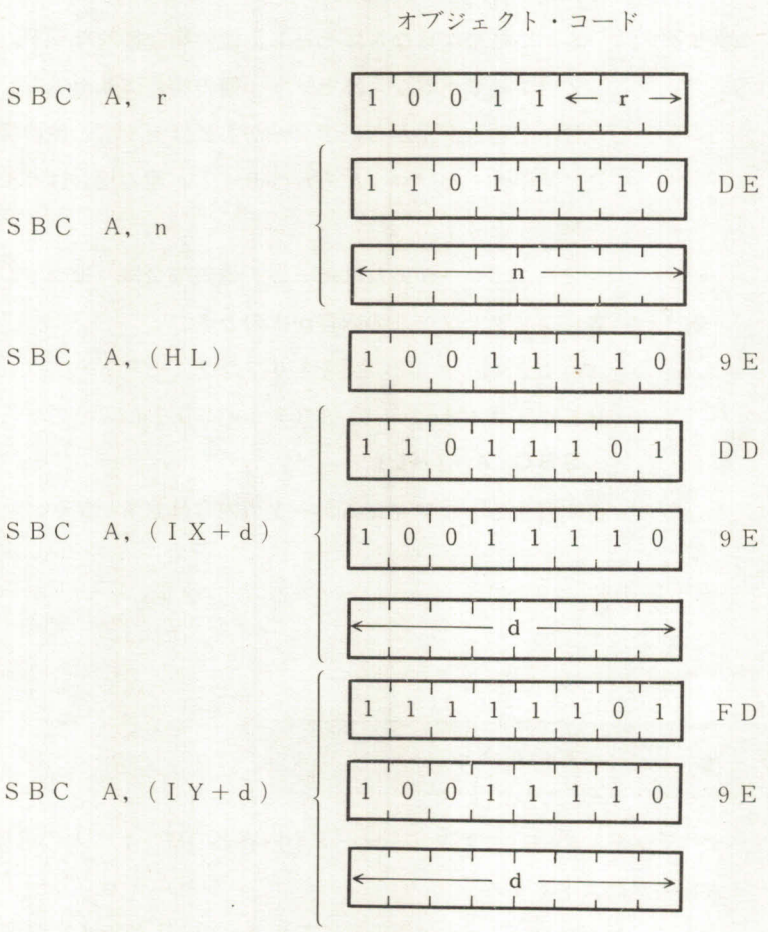
SBC A, s　；A←A−s−CY

（オペランド　A, s）

アキュムレータの内容からオペランドsの内容とキャリフラグの内容とを減じ、その結果をアキュムレータにストアする。

オペランドsにはr、n、(HL)、(IX+d)、(IY+d)が用いられる。

rとオブジェクト・コードは次のとおり。



レジスタ	r
B	000
C	001
D	010
E	011
H	100
L	101
A	111

実行：命令	Mサイクル	Tステート	実行時間(4MHz)
SBC A, r	1	4	1.00μs
SBC A, n	2	7(4＋3)	1.75μs
SBC A,(HL)	2	7(4＋3)	1.75μs
SBC A,(IX＋d)	5	19(4＋4＋3＋5＋3)	4.75μs
SBC A,(IY＋d)	5	19(4＋4＋3＋5＋3)	4.75μs

フラグ：S ：結果が負ならばセット、他の場合はリセット。
Z ：結果が零ならばセット、他の場合はリセット。
H ：ビット4からのボローがあればセット、他の場合はリセット。
P／V：オーバフローがあればセット、他の場合はリセット。
N ：セット。
C ：ボローがあればセット、他の場合はリセット。

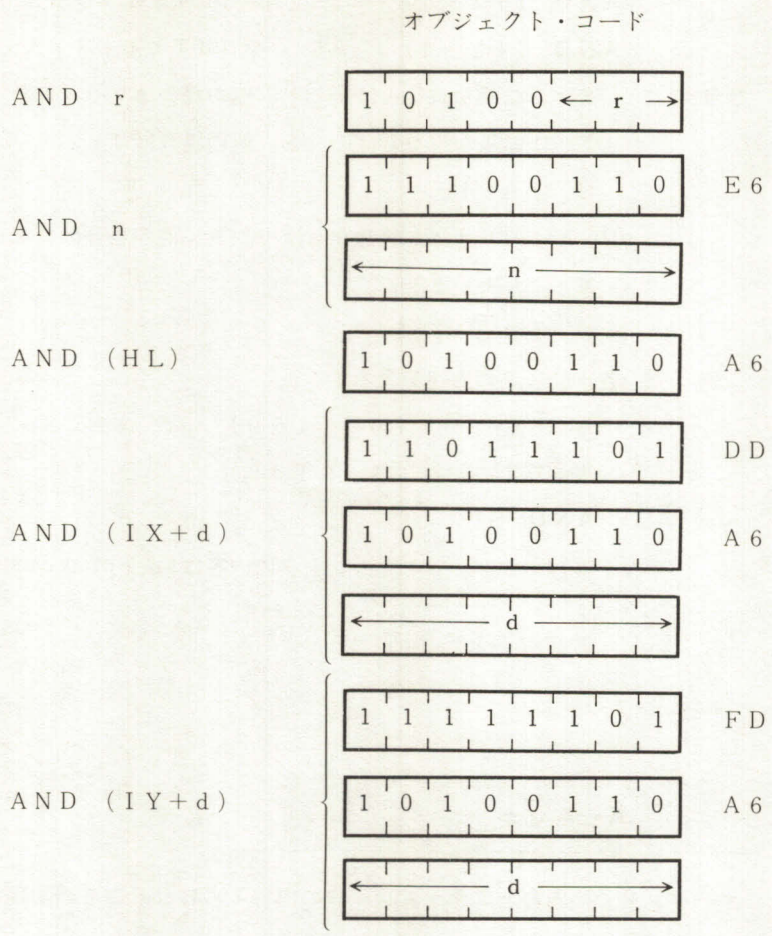
例 ：各レジスタ、メモリの内容が次のとき、
A ：24H　　キャリフラグ：セット
HL：3344H　　3344H：07H
SBC A, (HL)
を実行すると、アキュムレータの内容は1CHとなる。

AND s ; A←A∧s

（オペランド s）

オペランドsの内容とアキュムレータの内容とのビットごとの論理積をとり、その結果をアキュムレータにストアする。
オペランドsにはr、n、(HL)、(IX＋d)、(IY＋d)が用いられる。

rとオブジェクト・コードは次のとおり。



レジスタ	r
B =	0 0 0
C =	0 0 1
D =	0 1 0
E =	0 1 1
H =	1 0 0
L =	1 0 1
A =	1 1 1

実行：命令	Mサイクル	Tステート	実行時間(4MHz)
AND r	1	4	1.00μs
AND n	2	7(4+3)	1.75μs

AND （HL）	2	7（4＋3）	1.75μs
AND （IX＋d）	5	19（4＋4＋3＋5＋3）	4.75μs
AND （IY＋d）	5	19（4＋4＋3＋5＋3）	4.75μs

フラグ ： S ：結果が負ならばセット、他の場合はリセット。

Z ：結果が零ならばセット、他の場合はリセット。

H ：セット。

P／V：パリティが偶数ならばセット、他の場合はリセット。

N ：リセット。

C ：リセット。

例 ： 各レジスタの内容が次のとき、

B：6C$_H$（0110 1100）

A：D4$_H$（1101 0100）

AND B

を実行すると、アキュムレータの内容は44$_H$（0100 0100）となる。

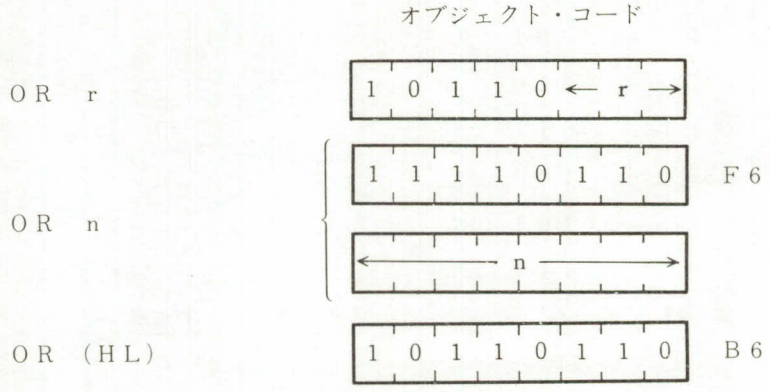
OR s ; A←A∨s

（オペランド s）

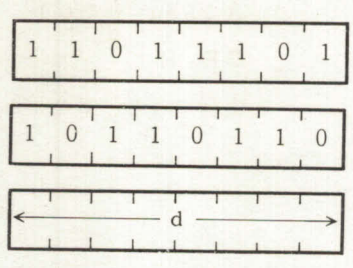
オペランドsの内容とアキュムレータの内容とのビットごとの論理和をとり、その結果をアキュムレータにストアする。

オペランドsにはr、n、（HL）、（IX＋d）、（IY＋d）が用いられる。

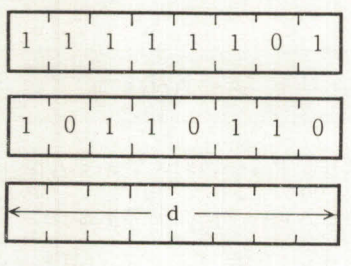
rとオブジェクト・コードは次のとおり。



OR （IX＋d）



OR （IY＋d）



レジスタ	r
B ＝	000
C ＝	001
D ＝	010
E ＝	011
H ＝	100
L ＝	101
A ＝	111

実行：命令	Mサイクル	Tステート	実行時間(4MHz)
OR r	1	4	1.00μs
OR n	2	7（4＋3）	1.75μs
OR （HL）	2	7（4＋3）	1.75μs
OR （IX＋d）	5	19（4＋4＋3＋5＋3）	4.75μs
OR （IY＋d）	5	19（4＋4＋3＋5＋3）	4.75μs

フラグ：　S　：結果が負ならばセット、他の場合はリセット。

Z　：結果が零ならばセット、他の場合はリセット。

H　：リセット

P／V：パリティが偶数ならばセット、他の場合はリセット。

N　：リセット。

C　：リセット。

例　：各レジスタの内容が次のとき、

H：27_H（0010　0111）

A：58H（0101　1000）

OR H

を実行すると、アキュムレータの内容は7FH(0111 1111)となる。

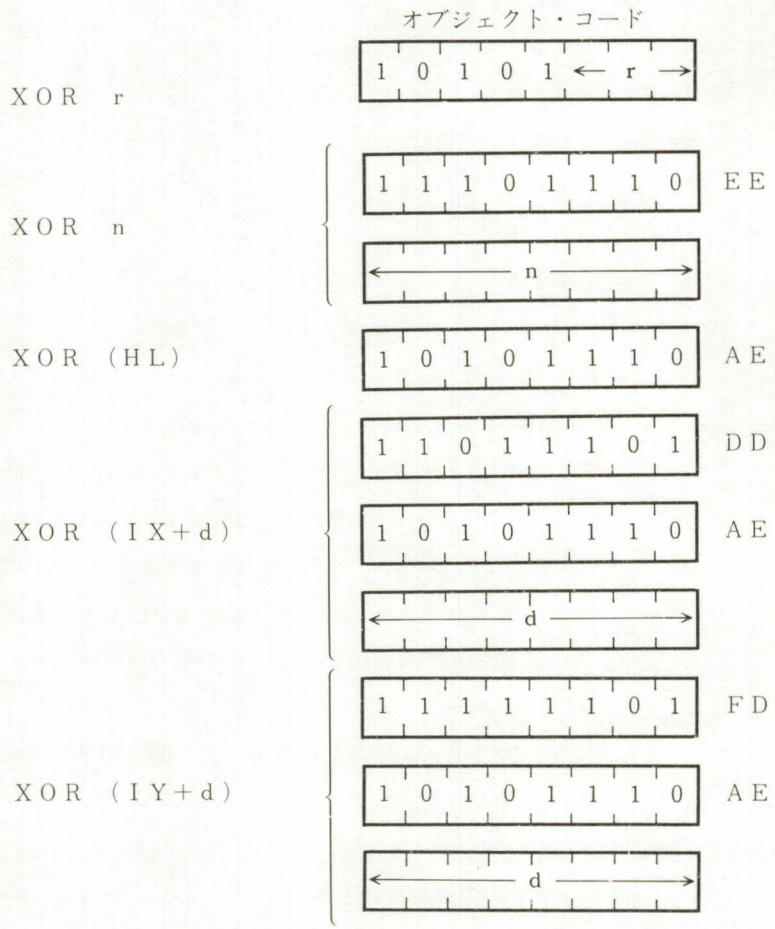
XOR　s　　；A←A⊕s

（オペランド　s）

オペランドsの内容とアキュムレータの内容とのビットごとの排他的論理和をとり、その結果をアキュムレータにストアする。

オペランドsにはr、n、(HL)、(IX+d)、(IY+d)が用いられる。

rとオブジェクト・コードは次のとおり。



レジスタ	r
B	000
C	001
D	010
E	011
H	100
L	101
A	111

実行：命令	Mサイクル	Tステート	実行時間(4MHz)
XOR r	1	4	1.00μs
XOR n	2	7(4＋3)	1.75μs
XOR (HL)	2	7(4＋3)	1.75μs
XOR (IX＋d)	5	19(4＋4＋3＋5＋3)	4.75μs
XOR (IY＋d)	5	19(4＋4＋3＋5＋3)	4.75μs

フラグ：　S　：結果が負ならばセット、他の場合はリセット。
　　Z　：結果が零ならばセット、他の場合はリセット。
　　H　：リセット。
　　P／V：パリティが偶数ならばセット、他の場合はリセット。
　　N　：リセット。
　　C　：リセット。

例　：アキュムレータの内容が36_H(0011 0110)のとき、

```
XOR 6DH          (注：6DH＝0110 1101)
```

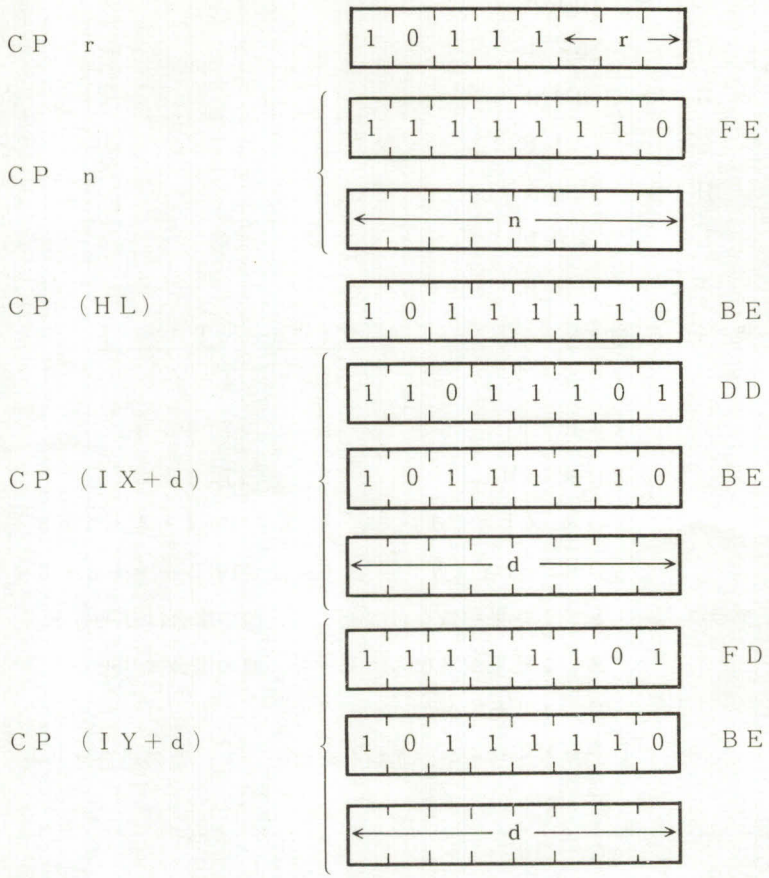
を実行すると、アキュムレータの内容は$5B_H$(0101 1011)となる。

CP s　；A－s

（オペランド　s）

オペランドsの内容とアキュムレータの内容を比較し、等しい場合フラグをセットする。
オペランドsにはr、n、(HL)、(IX＋d)、(IY＋d)が用いられる。
rとオブジェクト・コードは次のとおり。

オブジェクト・コード



レジスタ	r
B =	0 0 0
C =	0 0 1
D =	0 1 0
E =	0 1 1
H =	1 0 0
L =	1 0 1
A =	1 1 1

実行：命令	Mサイクル	Tステート	実行時間(4MHz)
CP r	1	4	1.00μs
CP n	2	7(4+3)	1.75μs
CP (HL)	2	7(4+3)	1.75μs
CP (IX+d)	5	19(4+4+3+5+3)	4.75μs
CP (IY+d)	5	19(4+4+3+5+3)	4.75μs

フラグ：　S　：結果が負ならばセット、他の場合はリセット。

Z　：結果が零ならばセット、他の場合はリセット。

H　：ビット4からのボローがあればセット、他の場合はリセット。

P／V：オーバフローならばセット、他の場合はリセット。

N　：セット。

C　：ボローがあればセット、他の場合はリセット。

例　：各レジスタ、メモリの内容が次のとき、

A　：54H

HL：5000H　　5000H番地：50H

CP (HL)

を実行すると、P／Vフラグはリセットされる。

INC　r　　；r←r+1

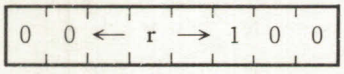
（オペランド　r）

レジスタrの内容を1だけ増す。

rとオブジェクト・コードは次のとおり。

レジスタ		r
A	=	111
B	=	000
C	=	001
D	=	010
E	=	011
H	=	100
L	=	101

オブジェクト・コード



実　行：Mサイクル：1，Tステート：4，実行時間：1.00μs（4MHz）

フラグ：　S　：結果が負ならばセット、他の場合はリセット。

Z　：結果が零ならばセット、他の場合はリセット。

H　：ビット3からのキャリがあればセット、他の場合はリセット。

P／V：実行前にrの内容が7FHであればセット、他の場合はリセット。

N　：リセット。

C　：変化せず。

例　：レジスタDの内容が37Hであるとき、

INC D

を実行すると、レジスタDの内容は38Hとなる。

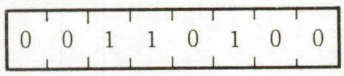
INC (HL)　；(HL)←(HL)+1

（オペランド (HL)）

レジスタ・ペアHLの内容が指定するメモリの内容を1だけ増す。

オブジェクト・コードは次のとおり。

オブジェクト・コード



34

実　行：Mサイクル：3，Tステート：11(4＋4＋3)，実行時間：2.75μs (4MHz)

フラグ：　S　：結果が負ならばセット、他の場合はリセット。

Z　：結果が零ならばセット、他の場合はリセット。

H　：ビット3からのキャリがあればセット、他の場合はリセット。

P／V：実行前に（HL）が7FHであればセット、他の場合はリセット。

N　：リセット。

C　：変化せず。

例　：各レジスタ、メモリの内容が次のとき、

HL：3344H　　3344H番地：76H

INC (HL)

を実行すると、メモリの3344H番地の内容は77Hとなる。

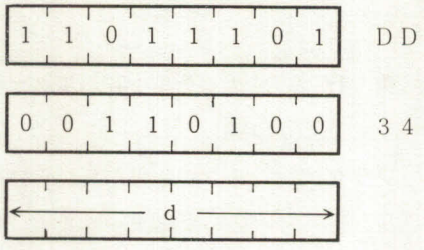
INC (IX+d)　；(IX+d)←(IX+d)+1

（オペランド (IX+d)）

インデックス・レジスタIXの内容にディスプレイスメントdを加えた値で指定されるメモリの内容を1だけ増す。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行：Mサイクル：6，Tステート：23(4＋4＋3＋5＋4＋3)，実行時間：5.75μs (4MHz)

フラグ：　S　：結果が負ならばセット、他の場合はリセット。

Z　：結果が零ならばセット、他の場合はリセット。

H　：ビット3からキャリがあればセット、他の場合はリセット。

P／V：実行前に（ＩＸ＋ｄ）が7FHであればセット、他の場合はリセット。

N　：リセット。

C　：変化せず。

例　：各レジスタ、メモリの内容が次のとき、

ＩＸ：１０１０H　　１０２０H番地：５６H

INC (IX＋10H)

を実行すると、メモリの1020H番地の内容は57Hとなる。

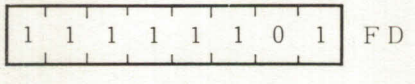
ＩＮＣ　(ＩＹ＋ｄ)　；(ＩＹ＋ｄ)←(ＩＹ＋ｄ)＋1

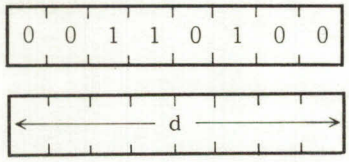
（オペランド　(ＩＹ＋ｄ)）

インデックス・レジスタＩＹの内容にディスプレイスメントｄを加えた値で指定されるメモリの内容を１だけ増す。

オブジェクト・コードは次のとおり。

オブジェクト・コード





34

実　行　：　Mサイクル：6，Tステート：23（4＋4＋3＋5＋4＋3），実行時間：5.75μs（4MHz）

フラグ　：　S　：結果が負ならばセット、他の場合はリセット。

Z　：結果が零ならばセット、他の場合はリセット。

H　：ビット3からのキャリがあればセット、他の場合はリセット。

P／V：実行前に（IY＋d）が7FHであればセット、他の場合はリセット。

N　：リセット。

C　：変化せず。

例　：各レジスタ、メモリが次のとき、

IY：1010H　　1020H番地：45H

INC (IY＋10H)

を実行すると、メモリの1020H番地の内容は46Hとなる。

DEC　m　　；m←m－1

（オペランド　m）

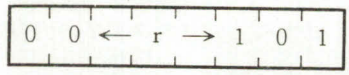
オペランドmの内容を1だけ減じる。

オペランドmにはINC命令と同様に、r、(HL)、(IX＋d)、(IY＋d)が用いられる。

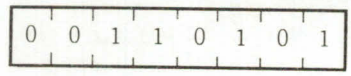
rとオブジェクト・コードは次のとおり。

オブジェクト・コード

DEC　r

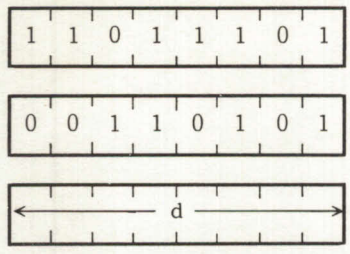


DEC　(HL)

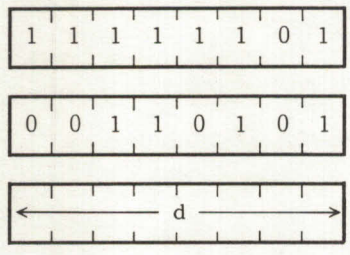


35

DEC （IX＋d）



DEC （IY＋d）



レジスタ	r
B ＝	000
C ＝	001
D ＝	010
E ＝	011
H ＝	100
L ＝	101
A ＝	111

実　行：

命　令	Mサイクル	Tステート	実行時間(4MHz)
DEC　r	1	4	1.00μs
DEC　（HL）	3	11(4＋4＋3)	2.75μs
DEC　（IX＋d）	6	23(4＋4＋3＋5＋4＋3)	5.75μs
DEC　（IY＋d）	6	23(4＋4＋3＋5＋4＋3)	5.75μs

フラグ：
S　：結果が負ならばセット、他の場合はリセット。
Z　：結果が零ならばセット、他の場合はリセット。
H　：ビット4からのボローがあればセット、他の場合はリセット。
P／V：実行前にオペランドmの内容が80Hならばセット、他の場合はリセット。
N　：セット。
C　：変化せず。

例　：レジスタDの内容が4BHのとき、

DEC D

を実行すると、レジスタDの内容は4AHとなる。

―― 汎用算術演算、ＣＰＵ制御　グループ ――

ＤＡＡ　；Decimal adjust acc

（オペランド　なし）

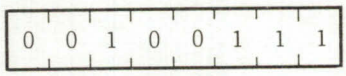
ＢＣＤ（２進化10進法）による加減算の際に、アキュムレータの内容を補正する。

加算（ＡＤＤ, ＡＤＣ, ＩＮＣ）、減算（ＳＵＢ, ＳＢＣ, ＤＥＣ, ＮＥＧ）に対する操作を次表に示す。

オペレーション	ＤＡＡ実行前のＣの内容（キャリフラグ）	アキュムレータの4~7ビットの値（16進）	ＤＡＡ実行前のＨの内容（ハーフキャリフラグ）	アキュムレータの0~3ビットの値（16進）	アキュムレータに加えられる数（16進）	ＤＡＡ実行後のＣの内容（キャリフラグ）
	0	0－9	0	0－9	0 0	0
	0	0－8	0	A－F	0 6	0
	0	0－9	1	0－3	0 6	0
ＡＤＤ	0	A－F	0	0－9	6 0	1
ＡＤＣ	0	9－F	0	A－F	6 6	1
ＩＮＣ	0	A－F	1	0－3	6 6	1
	1	0－2	0	0－9	6 0	1
	1	0－2	0	A－F	6 6	1
	1	0－3	1	0－3	6 6	1
ＳＵＢ	0	0－9	0	0－9	0 0	0
ＳＢＣ	0	0－8	1	6－F	F A	0
ＤＥＣ	1	7－F	0	0－9	A 0	1
ＮＥＧ	1	6－F	1	6－F	9 A	1

オブジェクト・コードは次のとおり。

オブジェクト・コード



２７

実　行：Ｍサイクル：１，Ｔステート：４，実行時間：1.00μs（4MHz）

フラグ：　Ｓ　：実行後アキュムレータのビット７が１ならばセット、他の場合はリセット。

Ｚ　：実行後アキュムレータの内容が零ならばセット、他の場合はリセット。

Ｈ　：前記表を参照。

Ｐ／Ｖ：実行後アキュムレータのパリティが偶数ならばセット、他の場合はリセット。

Ｎ　：変化せず。

Ｃ　：前記表を参照。

例　：16（ＢＣＤ）と28（ＢＣＤ）の加算は10進算術演算では

```
  １６
＋ ２８
------
  ４４
```

となる。

2進表示したものを2進算術演算で加えると、

```
    0 0 0 1   0 1 1 0
+   0 0 1 0   1 0 0 0
-------------------------
    0 0 1 1   1 1 1 0      3EH
```

となる。

DAA命令はこの結果を補正して正しいBCD表示にする。

```
    0 0 1 1   1 1 1 0
+   0 0 0 0   0 1 1 0
-------------------------
    0 1 0 0   0 1 0 0   =  44 (BCD)
```

CPL　；A←$\overline{A}$

（オペランド　なし）

アキュムレータの内容について1の補数をとる。

オブジェクト・コードは次のとおり。

オブジェクト・コード

0	0	1	0	1	1	1	1

2F

実　行：Mサイクル：1，Tステート：4，実行時間：1.00μs（4 MHz）

フラグ：　S　：変化せず。
　　Z　：変化せず。
　　H　：セット。
　　P／V：変化せず。
　　N　：セット。
　　C　：変化せず。

例　：アキュムレータの内容が1001 1010（9AH）のとき、

```
CPL
```

を実行すると、アキュムレータの内容は、

0 1 1 0　0 1 0 1　（65H）

となる。

NEG　；A←0－A

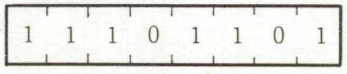
（オペランド　なし）

アキュムレータの内容について２の補数をとる。

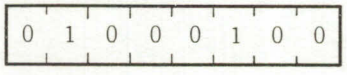
これは零からアキュムレータの内容を減じるのと同様である。80Hの場合変化しないので注意すること。

オブジェクト・コードは次のとおり。

オブジェクト・コード



ED



44

実　行：Mサイクル：2，Tステート：8（4＋4），実行時間：2.00μs(4MHz)

フラグ：　S　：結果が負ならばセット、他の場合はリセット。

Z　：結果が零ならばセット、他の場合はリセット。

H　：ビット4からボローがあればセット、他の場合はリセット。

P／V：実行前にアキュムレータの内容が80Hならばセット、他の場合はリセット。

N　：セット。

C　：実行前にアキュムレータの内容が00Hでなければセット、他の場合はリセット。

例　：アキュムレータの内容が次のとき、

1010　1100　(ACH)

NEG

を実行すると、アキュムレータの内容は次のようになる。

0101　0100　(54H)

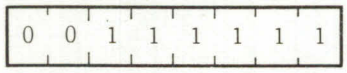
CCF　；CY←$\overline{CY}$

（オペランド　なし）

レジスタF中のC（キャリ）フラグの1の補数をとる。

オブジェクト・コードは次のとおり。

オブジェクト・コード



3F

実　行：Mサイクル：1，Tステート：4，実行時間：1.00μs（4MHz）

フラグ：　S　：変化せず。

Z　：変化せず。

H　：実行以前のキャリに等しい。

P／V：変化せず。

N　：リセット。

C　：実行前にキャリ・フラグが零ならばセット、他の場合はリセット。

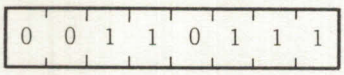
SCF　；CY←1

（オペランド　なし）

Fレジスタ中のC（キャリ）フラグをセットする。

オブジェクト・コードは次のとおり。

オブジェクト・コード



3 7

実　行：Mサイクル：1，Tステート：4，実行時間：1.00μs（4MHz）

フラグ：　S　：変化せず。

Z　：変化せず。

H　：リセット。

P／V：変化せず。

N　：リセット。

C　：セット。

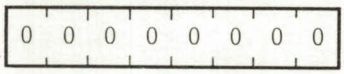
NOP　；No operation

（オペランド　なし）

プログラム・カウンタを1進めるだけで他に影響を与える動作はしない。リフレッシュ信号は出る。

オブジェクト・コードは次のとおり。

オブジェクト・コード



００

実　行 ： Mサイクル：1，Tステート：4，実行時間：1.00μs（4MHz）

フラグ ：変化せず。

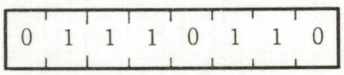
HALT　；Halt

（オペランド　なし）

次の割り込みかリセットが来るまでＣＰＵを停止させる。停止中はメモリ内容を保持するためＮＯＰを実行しているがＰＣは＋１されない。

オブジェクト・コードは次のとおり。

オブジェクト・コード



７６

実　行 ： Mサイクル：1，Tステート：4，実行時間：1.00μs（4MHz）

フラグ ： 変化せず。

DI　；IFF←0

（オペランド　なし）

イネーブル・フリップフロップ（ＩＦＦ１とＩＦＦ２）をリセットして、ＩＮＴ割り込みを無効にする。

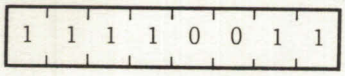
この命令の実行中に入ったＩＮＴ割り込みも無効にするので注意すること。

この命令の実行後、ＣＰＵは割り込み要求（ＩＮＴ）信号には応答しなくなる。

次にＥＩ命令が実行されるまで解除されない。

オブジェクト・コードは次のとおり。

オブジェクト・コード



F3

実　行 ： Mサイクル：1，Tステート：4，実行時間：1.00μs（4MHz）

フラグ ： 変化せず。

EI　　；IFF←1

（オペランド　なし）

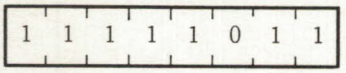
イネーブル・フリップフロップ（IFF1とIFF2）をセットして、INT割り込みを有効にする。

この命令の実行中に入ったINT割り込みは無効にするので注意すること。

この命令の実行後、CPUは割り込み要求（INT）信号に応答するようになる。

オブジェクト・コードは次のとおり。

オブジェクト・コード



FB

実　行 ： Mサイクル：1，Tステート：4，実行時間：1.00μs（4MHz）

フラグ ： 変化せず。

IM 0　　；Interrupt mode zero

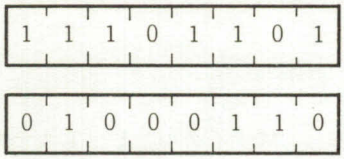
（オペランド　0）

割り込みをモード0にセットする。

このモードの割り込みによって割り込みアクノリッジの期間にデータ・バス上に任意の命令を挿入し、CPUに実行させることができる。

オブジェクト・コードは次のとおり。

オブジェクト・コード



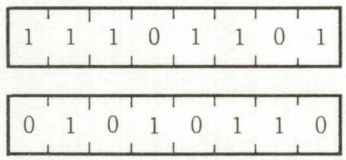
実　行：Mサイクル：2，Tステート：8(4＋4)，実行時間：2.00μs（4MHz）
フラグ：変化せず。

IM 1　　; Interrupt mode one

（オペランド　1）

割り込みをモード1にセットする。
このモードの割り込みによって0038H番地へのリスタート命令を実行する。
オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行：Mサイクル：2，Tステート：8(4＋4)，実行時間：2.00μs（4MHz）
フラグ：変化せず。

IM 2　　; Interrupt mode two

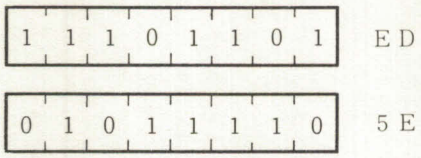
（オペランド　2）

割り込みをモード2にセットする。
このモードでは、メモリ中の任意の番地へ割り込みコールを実行できる。この任意の実行開始番地は、あらかじめメモリ内に作られたテーブルから参照される。このテーブルは割り込みベクトル・レジスタIの内容を上位バイトとし、割り込みをかけた機器から与えられる8ビットを下位

バイトとするベクトル・アドレスによって指定される。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行：Mサイクル：2，Tステート：8(4＋4)，実行時間：2.00μs（4MHz）

フラグ：変化せず。

——— 16ビット　算術演算　グループ ———

ADD HL，ss ；HL←HL+ss

（オペランド HL，ss）

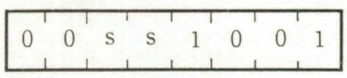
レジスタ・ペアssの内容とレジスタ・ペアＨＬの内容とを加え、その結果をＨＬにストアする。

オペランドssにはＢＣ、ＤＥ、ＨＬ、ＳＰが用いられる。

ssとオブジェクト・コードは次のとおり。

レジスタ・ペア	ss
ＢＣ ＝	００
ＤＥ ＝	０１
ＨＬ ＝	１０
ＳＰ ＝	１１

オブジェクト・コード



実　行：Mサイクル：３，Ｔステート：11(４＋４＋３)，実行時間：2.75μs（4MHz）

フラグ：　Ｓ　：変化せず。

Ｚ　：変化せず。

Ｈ　：ビット11からのキャリがあればセット、他の場合はリセット。

Ｐ／Ｖ：変化せず。

Ｎ　：リセット。

Ｃ　：ビット15からのキャリがあればセット、他の場合はリセット。

例　：レジスタ・ペアの内容が次のとき、

ＨＬ：２５２５H　　ＤＥ：３３３３H

ADD HL, DE

を実行すると、レジスタ・ペアＨＬの内容は5858Hとなる。

ADC HL, ss ；HL←HL+ss+CY

（オペランド HL，ss）

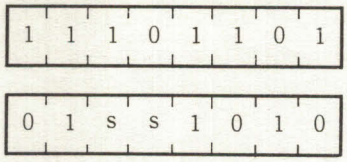
レジスタ・ペアｓｓの内容とレジスタ・ペアＨＬの内容およびキャリ・フラグの内容とを加え、その結果をＨＬにストアする。

オペランドｓｓにはＢＣ、ＤＥ、ＨＬ、ＳＰが用いられる。

ｓｓとオブジェクト・コードは次のとおり。

レジスタ・ペア		ｓｓ
ＢＣ	＝	００
ＤＥ	＝	０１
ＨＬ	＝	１０
ＳＰ	＝	１１

オブジェクト・コード



実　行：Mサイクル：４，Ｔステート：15(４＋４＋４＋３)，実行時間：3.75μs（4MHz）

フラグ：　Ｓ　：結果が負ならばセット、他の場合はリセット。

Ｚ　：結果が零ならばセット、他の場合はリセット。

Ｈ　：ビット11からのキャリがあればセット、他の場合はリセット。

Ｐ／Ｖ：オーバフローがあればセット、他の場合はリセット。

Ｎ　：リセット。

Ｃ　：ビット15からのキャリがあればセット、他の場合はリセット。

例　：レジスタ・ペア、フラグの内容が次のとき、

ＨＬ：２０３７H　　ＢＣ：１４８６H

キャリ・フラグ：セット

```
ADC HL, BC
```

を実行すると、レジスタ・ペアＨＬの内容は34BEHとなる。

SBC HL, ss　；HL←HL−ss−CY

（オペランド　ＨＬ，ss）

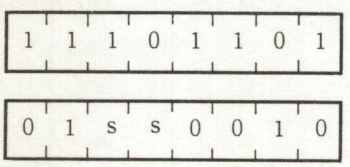
レジスタ・ペアssの内容とキャリ・フラグの内容とをレジスタ・ペアＨＬの内容から減じ、その結果をＨＬにストアする。

オペランドssにはＢＣ、ＤＥ、ＨＬ、ＳＰが用いられる。

ssとオブジェクト・コードは次のとおり。

レジスタ・ペア		ｓｓ
ＢＣ	＝	００
ＤＥ	＝	０１
ＨＬ	＝	１０
ＳＰ	＝	１１

オブジェクト・コード



実　行：Mサイクル：４，Ｔステート：15(４＋４＋４＋３)，実行時間：3.75μs（4MHz）

フラグ：　Ｓ　：結果が負ならばセット、他の場合はリセット。

Z　：結果が零ならばセット、他の場合はリセット。

H　：ビット12からのボローがあればセット、他の場合はリセット。

P／V：オーバフローがあればセット、他の場合はリセット。

N　：セット。

C　：ボローがあればセット、他の場合はリセット。

例　：レジスタ・ペア、フラグが次のとき、

HL：９８７６H　　DE：１２３４H

キャリ・フラグ：セット

SBC HL, DE

を実行すると、レジスタ・ペアHLの内容は8641Hとなる。

ADD　IX, pp　；IX←IX+pp

（オペランド　IX, pp）

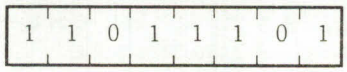
レジスタ・ペアppの内容とインデックス・レジスタIXの内容とを加え、その結果をIXにストアする。

オペランドppにはBC、DE、IX、SPが用いられる。

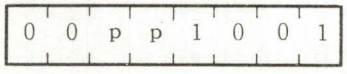
ppとオブジェクト・コードは次のとおり。

レジスタ・ペア		pp
BC	=	00
DE	=	01
IX	=	10
SP	=	11

オブジェクト・コード



DD



実　行：Mサイクル：4，Tステート：15(４＋４＋４＋３)，実行時間：3.75μs（4MHz）

フラグ：　S　：変化せず。

Z　：変化せず。

H　：ビット11からのキャリがあればセット、他の場合はリセット。

P／V：変化せず。

N　：リセット。

C　：ビット15からのキャリがあればセット、他の場合はリセット。

例　：レジスタの内容が次のとき、

IX：２２４４H　　BC：６６６６H

ADD IX, BC

を実行すると、インデックス・レジスタＩＸの内容は88AAHとなる。

ADD　ＩＹ, rr　　；ＩＹ←ＩＹ+rr

（オペランド　ＩＹ, ｒｒ）

レジスタ・ペアｒｒの内容とインデックス・レジスタＩＹの内容とを加え、その結果をＩＹにストアする。

オペランドｒｒにはＢＣ、ＤＥ、ＩＹ、ＳＰが用いられる。

ｒｒとオブジェクト・コードは次のとおり。

レジスタ・ペア		ｒｒ
ＢＣ	＝	００
ＤＥ	＝	０１
ＩＹ	＝	１０
ＳＰ	＝	１１

オブジェクト・コード

1	1	1	1	1	1	0	1	FD
0	0	r	r	1	0	0	1	

実　行：Mサイクル：４，Ｔステート：15(４＋４＋４＋３)，実行時間：3.75μs（4MHz）

フラグ：　Ｓ　：変化せず。

Ｚ　：変化せず。

Ｈ　：ビット11からのキャリがあればセット、他の場合はリセット。

Ｐ／Ｖ：変化せず。

Ｎ　：リセット。

Ｃ　：ビット15からのキャリがあればセット、他の場合はリセット。

例　：各レジスタの内容が次のとき、

ＩＹ：２２２２H　　ＢＣ：４４４４H

ADD IY, BC

を実行すると、インデックス・レジスタＩＹの内容は6666Hとなる。

ＩＮＣ　ｓｓ　　；ｓｓ←ｓｓ＋１

（オペランド　ｓｓ）

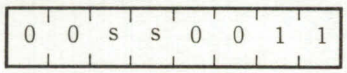
レジスタ・ペアｓｓの内容を１だけ増す。

オペランドｓｓにはＢＣ、ＤＥ、ＨＬ、ＳＰが用いられる。

ｓｓとオブジェクト・コードは次のとおり。

レジスタ・ペア	ｓｓ
ＢＣ	００
ＤＥ	０１
ＨＬ	１０
ＳＰ	１１

オブジェクト・コード



実　行：Ｍサイクル：１，Ｔステート：６，実行時間：1.50μｓ（4MHz）

フラグ：変化せず。

例　：レジスタ・ペアＨＬの内容が2200Hのとき、

```
INC HL
```

を実行すると、レジスタ・ペアＨＬの内容は2201Hとなる。

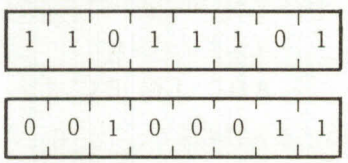
ＩＮＣ　ＩＸ　　；ＩＸ←ＩＸ＋１

（オペランド　ＩＸ）

インデックス・レジスタＩＸの内容を１だけ増す。

オブジェクト・コードは次のとおり。

オブジェクト・コード



ＤＤ

２３

実　行：Ｍサイクル：２，Ｔステート：10（４＋６），実行時間：2.50μｓ（4MHz）

フラグ：変化せず。

例　：インデックス・レジスタＩＸの内容が1200Hのとき、

```
INC IX
```

を実行すると、インデックス・レジスタＩＸの内容は1201Hとなる。

INC IY ; IY←IY+1

（オペランド IY）

インデックス・レジスタＩＹの内容を１だけ増す。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	1	1	1	0	1	FD
0	0	1	0	0	0	1	1	23

実　行 ： Mサイクル：2，Ｔステート：10(４＋６)，実行時間：2.50μs（4MHz）

フラグ ： 変化せず。

例　 ： インデックス・レジスタＩＹの内容が1384Hのとき、

```
INC IY
```

を実行すると、インデックス・レジスタＩＹの内容は1385Hとなる。

DEC ss ; ss←ss−1

（オペランド ss）

レジスタ・ペアｓｓの内容を１だけ減じる。

オペランドｓｓにはＢＣ、ＤＥ、ＨＬ、ＳＰが用いられる。

ｓｓとオブジェクト・コードは次のとおり。

レジスタ・ペア		ｓｓ
ＢＣ	＝	００
ＤＥ	＝	０１
ＨＬ	＝	１０
ＳＰ	＝	１１

オブジェクト・コード

0	0	s	s	1	0	1	1

実　行 ： Mサイクル：1，Ｔステート：6，実行時間：1.50μs（4MHz）

フラグ ： 変化せず。

例　 ： レジスタ・ペアＨＬの内容が2003Hのとき、

```
DEC HL
```

を実行すると、レジスタ・ペアＨＬの内容は2002Hとなる。

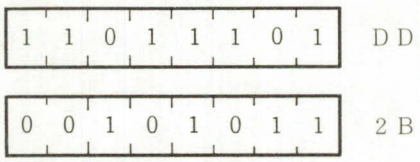
DEC　IX　　; IX←IX−1

（オペランド　IX）

インデックス・レジスタＩＸの内容を１だけ減じる。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行：Mサイクル：2，Tステート：10(４＋６)，実行時間：2.50μs（4MHz）

フラグ：変化せず。

例　　：インデックス・レジスタＩＸの内容が1822Hのとき、

```
DEC IX
```

を実行すると、インデックス・レジスタＩＸの内容は1821Hとなる。

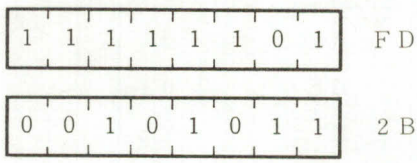
DEC　IY　　; IY←IY−1

（オペランド　IY）

インデックス・レジスタＩＹの内容を１だけ減じる。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行：Mサイクル：2，Tステート：10(４＋６)，実行時間：2.50μs（4MHz）

フラグ：変化せず。

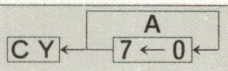
例　　：インデックス・レジスタＩＹの内容が4800Hのとき、

```
DEC IY
```

を実行すると、インデックス・レジスタＩＹの内容は47FFHとなる。

—— ローテイト、シフト　グループ ——

RLCA ；



（オペランド　なし）

アキュムレータ（レジスタA）の内容を左へローテイトする。

すなわち、ビット0の内容をビット1へコピーし、コピー前のビット1の内容をビット2へコピーする。

同様な操作を順次行う。ビット7の内容はキャリ・フラグとビット0の両方に入る。

オブジェクト・コードは次のとおり。

オブジェクト・コード

0	0	0	0	0	1	1	1

07

実　行：Mサイクル：1，Tステート：4，実行時間：1.00μs（4MHz）

フラグ：　S　：変化せず。

Z　：変化せず。

H　：リセット。

P／V：変化せず。

N　：リセット。

C　：アキュムレータのビット7からのデータ。

例　：アキュムレータの内容が次のとき、

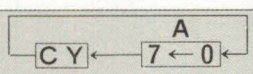
CY	7	6	5	4	3	2	1	0
	1	1	0	0	1	0	1	0

RLCA

を実行すると、アキュムレータとキャリ・フラグの内容は次のようになる。

CY	7	6	5	4	3	2	1	0
1	1	0	0	1	0	1	0	1

RLA ；



（オペランド　なし）

アキュムレータ（レジスタA）の内容を左へローテイトする。すなわち、ビット0の内容をビット1へコピーし、コピー前のビット1の内容をビット2へコピーする。同様な操作を順次行う。

ビット７の内容はキャリ・フラグにコピーし、コピー前のキャリ・フラグの内容はビット０にコピーする。

オブジェクト・コードは次のとおり。

オブジェクト・コード

0	0	0	1	0	1	1	1

１７

実　行：Mサイクル：１，Tステート：４，実行時間：1.00μs（4MHz）

フラグ：　S　：変化せず。

Z　：変化せず。

H　：リセット。

P／V：変化せず。

N　：リセット。

C　：アキュムレータのビット７からのデータ。

例　：アキュムレータとキャリ・フラグの内容が次のとき、

CY	7	6	5	4	3	2	1	0
1	0	1	0	1	1	1	0	1

RLA

を実行すると、アキュムレータとキャリ・フラグの内容は次のようになる。

CY	7	6	5	4	3	2	1	0
0	1	0	1	1	1	0	1	1

RRCA　；A　7→0　→CY

（オペランド　なし）

アキュムレータ（レジスタA）の内容を右にローテイトする。

すなわち、ビット７の内容をビット６にコピーし、コピー前のビット６の内容をビット５にコピーする。

同様な操作を順次行う。ビット０の内容はビット７とキャリ・フラグの両方にコピーする。

オブジェクト・コードは次のとおり。

オブジェクト・コード

0	0	0	0	1	1	1	1	0F

実　行：Mサイクル：1，Tステート：4，実行時間：1.00μs（4MHz）

フラグ：　S　：変化せず。

Z　：変化せず。

H　：リセット。

P／V：変化せず。

N　：リセット。

C　：アキュムレータのビット0からのデータ。

例　：アキュムレータの内容が次のとき、

CY	7	6	5	4	3	2	1	0
	0	0	1	0	0	1	0	1

RRCA

を実行すると、アキュムレータとキャリ・フラグの内容は次のようになる。

CY	7	6	5	4	3	2	1	0
1	1	0	0	1	0	0	1	0

RRA　；A 7→0 → CY

（オペランド　なし）

アキュムレータ（レジスタA）の内容を右にローテイトする。すなわちビット7の内容をビット6にコピーし、コピー前のビット6の内容をビット5にコピーする。同様な操作を順次行う。ビット0の内容はキャリ・フラグにコピーし、コピー前のキャリ・フラグの内容をビット7にコピーする。

オブジェクト・コードは次のとおり。

オブジェクト・コード

0	0	0	1	1	1	1	1	1F

実　行：Mサイクル：1，Tステート：4，実行時間：1.00μs（4MHz）

フラグ：　S　：変化せず。

Z　：変化せず。

H　：リセット。

P／V：変化せず。

N　：リセット。

C　：アキュムレータのビット0からのデータ。

例　：アキュムレータとキャリ・フラグの内容が次のとき、

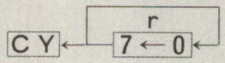
CY	7	6	5	4	3	2	1	0
1	0	1	1	1	0	1	0	1

RRA

を実行すると、アキュムレータとキャリ・フラグの内容は次のようになる。

CY	7	6	5	4	3	2	1	0
1	1	0	1	1	1	0	1	0

RLC　r　；



（オペランド　r）

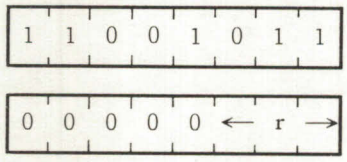
レジスタrの8ビットの内容を左へローテイトする。

ビット0の内容をビット1にコピーし、コピー前のビット1の内容をビット2にコピーする。同様な操作を順次行う。ビット7の内容はビット0とキャリ・フラグの両方にコピーする。

rとオブジェクト・コードは次のとおり。

レジスタ		r
B	=	0 0 0
C	=	0 0 1
D	=	0 1 0
E	=	0 1 1
H	=	1 0 0
L	=	1 0 1
A	=	1 1 1

オブジェクト・コード



実　行：Mサイクル：2，Tステート：8（4＋4），実行時間：2.00μs（4MHz）

フラグ：　S　：結果が負ならばセット、他の場合はリセット。

Z　：結果が零ならばセット、他の場合はリセット。

H　：リセット。

P／V：パリティが偶数ならばセット、他の場合はリセット。

N　：リセット。

C　：レジスタｒのビット７からのデータ。

例　：レジスタｒの内容が次のとき、

CY	7	6	5	4	3	2	1	0
	0	1	1	1	0	1	1	1

RLC r

を実行すると、レジスタｒとキャリ・フラグの内容は次のようになる。

CY	7	6	5	4	3	2	1	0
0	1	1	1	0	1	1	1	0

RLC (HL)　；CY ← 7 ← 0 ← (HL)

（オペランド　(HL)）

レジスタ・ペアＨＬの内容が指定するメモリの内容を左にローテイトする。

ビット０の内容をビット１にコピーし、コピー前のビット１の内容をビット２にコピーする。同様な操作を順次行う。ビット７の内容はキャリ・フラグとビット０の両方にコピーする。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	0	0	1	0	1	1	CB
0	0	0	0	0	1	1	0	06

実　行：Mサイクル：４，Tステート：15(４＋４＋４＋３)，実行時間：3.75μs（4MHz）

フラグ：　S　：結果が負ならばセット、他の場合はリセット。

Z　：結果が零ならばセット、他の場合はリセット。

H　：リセット。

P／V：パリティが偶数ならばセット、他の場合はリセット。

N　：リセット。

C　：前記メモリのビット７からのデータ。

例　：レジスタ・ペアＨＬの内容が3636Hで、メモリの3636H番地の内容が次のとき、

CY	7	6	5	4	3	2	1	0
	1	0	1	0	0	1	1	0

RLC (HL)

を実行すると、メモリの3636H番地とキャリ・フラグの内容は次のようになる。

CY	7	6	5	4	3	2	1	0
1	0	1	0	0	1	1	0	1

RLC (IX+d)　；CY ← (IX+d) 7 ← 0

（オペランド(IX+d)）

インデックス・レジスタＩＸの内容にディスプレイスメントｄを加えた値で指定されるメモリの内容を左にローテイトする。

ビット０の内容をビット１にコピーし、その前のビット１の内容をビット２にコピーする。同様な操作を１バイト行う。ビット７の内容はキャリ・フラグとビット０の両方にコピーする。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1 1 0 1 1 1 0 1	DD
1 1 0 0 1 0 1 1	CB
← d →	
0 0 0 0 0 1 1 0	06

実　行：Mサイクル：６，Tステート：23(４＋４＋３＋５＋４＋３)，実行時間：5.75μs (4MHz)

フラグ：　S　：結果が負ならばセット、他の場合はリセット。

Z　：結果が零ならばセット、他の場合はリセット。

H　：リセット。

P／V：パリティが偶数ならばセット、他の場合はリセット。

N　：リセット。

C　：前記メモリのビット７からのデータ。

例　：インデックス・レジスタＩＸの内容が1200Hで、メモリの1202H番地の内容が次のとき、

CY	7	6	5	4	3	2	1	0
	1	0	1	0	1	1	1	0

RLC (IX＋2H)

を実行すると、メモリの1202H番地とキャリ・フラグの内容は次のようになる。

CY	7	6	5	4	3	2	1	0
1	0	1	0	1	1	1	0	1

RLC (ＩＹ＋d)　；CY ← (IY＋d) 7 ← 0

(オペランド(ＩＹ＋ｄ))

インデックス・レジスタＩＹの内容に、ディスプレイスメントｄを加えた値で指定されるメモリの内容を左へローテイトする。

ビット０の内容をビット１にコピーし、コピー前のビット１の内容をビット２にコピーする。同様な操作を順次行う。ビット７の内容はキャリ・フラグとビット０の両方にコピーする。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	1	1	1	1	0	1	FD
1	1	0	0	1	0	1	1		CB
←				d				→	
0	0	0	0	0	1	1	0		06

実　行：Mサイクル：6，Tステート：23(４＋４＋３＋５＋４＋３)，実行時間：5.75μs
(4MHz)

フラグ：　S　：結果が負ならばセット、他の場合はリセット。
Z　：結果が零ならばセット、他の場合はリセット。
H　：リセット。
P／V：パリティが偶数ならばセット、他の場合はリセット。
N　：リセット。
C　：前記メモリのビット７からのデータ。

例　：インデックス・レジスタＩＹの内容が3300Hで、メモリの3306H番地の内容が次のとき、

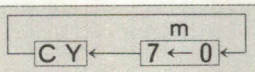
CY	7	6	5	4	3	2	1	0
	0	1	0	0	1	1	0	1

RLC (IY＋6H)

を実行すると、メモリの3306H番地とキャリ・フラグの内容は次のようになる。

CY	7	6	5	4	3	2	1	0
0	1	0	0	1	1	0	1	0

RL　m　；

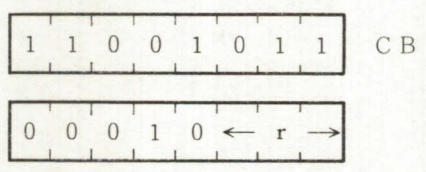


（オペランド　m）

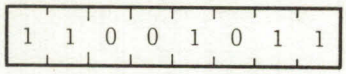
オペランドmの内容を左へローテイトする。
ビット０の内容をビット１にコピーし、コピー前のビット１の内容をビット２にコピーする。同様な操作を順次行う。ビット７の内容はキャリ・フラグにコピーし、コピー前のキャリ・フラグの内容はビット０にコピーする。
オペランドmにはr、(ＨＬ)、(ＩＸ+d)、(ＩＹ+d) が用いられる。
ｒとオブジェクト・コードは次のとおり。

オブジェクト・コード

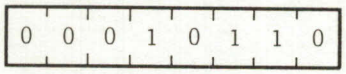
RL　r



RL (HL)

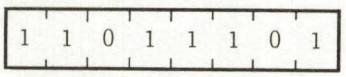


CB

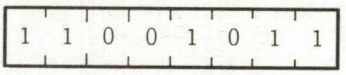


16

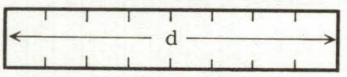
RL (IX+d)

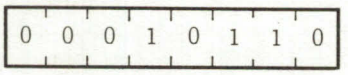


DD



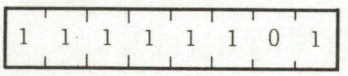
CB



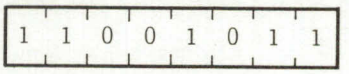


16

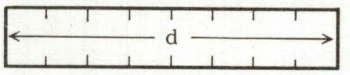
RL (IY+d)

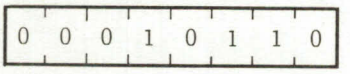


FD



CB





16

レジスタ	r
B =	0 0 0
C =	0 0 1
D =	0 1 0
E =	0 1 1
H =	1 0 0
L =	1 0 1
A =	1 1 1

実行： 命令	Mサイクル	Tステート	実行時間(4MHz)
RL r	2	8(4+4)	2.00μs
RL (HL)	4	15(4+4+4+3)	3.75μs
RL (IX+d)	6	23(4+4+3+5+4+3)	5.75μs
RL (IY+d)	6	23(4+4+3+5+4+3)	5.75μs

フラグ： S ：結果が負ならばセット、他の場合はリセット。

Z　：結果が零ならばセット、他の場合はリセット。

H　：リセット。

P／V：パリティが偶数ならばセット、他の場合はリセット。

N　：リセット。

C　：ビット７からのデータ。

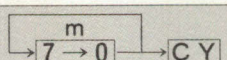
例　：レジスタＤとキャリ・フラグの内容が次のとき、

CY	7	6	5	4	3	2	1	0
0	1	1	0	0	1	0	0	0

RL D

を実行すると、レジスタＤとキャリ・フラグの内容は次のようになる。

CY	7	6	5	4	3	2	1	0
1	1	0	0	1	0	0	0	0

RRC　m　； 

（オペランド　m）

オペランドｍの内容を右にローテイトする。

ビット７の内容をビット６にコピーし、コピー前のビット６の内容をビット５にコピーする。同様な操作を順次行う。ビット０の内容はキャリ・フラグとビット７の両方にコピーする。

オペランドｍにはｒ、（ＨＬ）、（ＩＸ＋d）、（ＩＹ＋d）が用いられる。

ｒとオブジェクト・コードは次のとおり。

オブジェクト・コード

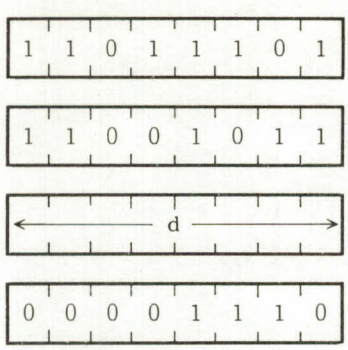
RRC　r

1	1	0	0	1	0	1	1	CB
0	0	0	0	1	←	r	→	

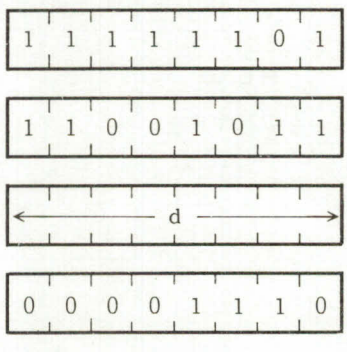
RRC　(HL)

1	1	0	0	1	0	1	1	CB
0	0	0	0	1	1	1	0	0E

RRC （IX+d）



RRC （IY+d）



レジスタ	r
B ＝	0 0 0
C ＝	0 0 1
D ＝	0 1 0
E ＝	0 1 1
H ＝	1 0 0
L ＝	1 0 1
A ＝	1 1 1

実　行：命　令	Mサイクル	Tステート	実行時間(4MHz)
RRC　r	2	8（4＋4）	2.00μs
RRC（HL）	4	15（4＋4＋4＋3）	3.75μs
RRC（IX＋d）	6	23（4＋4＋3＋5＋4＋3）	5.75μs
RRC（IY＋d）	6	23（4＋4＋3＋5＋4＋3）	5.75μs

フラグ：　S　：結果が負ならばセット、他の場合はリセット。

Z　：結果が零ならばセット、他の場合はリセット。

H　：リセット。

P／V：パリティが偶数ならばセット、他の場合はリセット。

N　：リセット。

C　：ビット0からのデータ。

例　：レジスタAの内容が次のとき、

CY	7	6	5	4	3	2	1	0
	0	1	1	0	1	0	0	1

RRC A

を実行すると、レジスタAとキャリ・フラグの内容は次のようになる。

CY	7	6	5	4	3	2	1	0
1	1	0	1	1	0	1	0	0

RR m　；m →7→0→ CY

（オペランド　m）

オペランドmの内容を右にローテイトする。

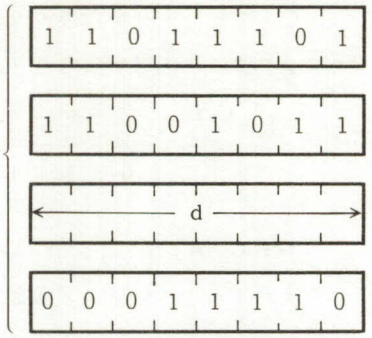
ビット7の内容をビット6にコピーし、コピー前のビット6の内容をビット5にコピーする。同様な操作を順次行う。ビット0の内容はキャリ・フラグにコピーし、コピー前のキャリ・フラグの内容をビット7にコピーする。

オペランドmにはr、(HL)、(IX+d)、(IY+d) が用いられる。

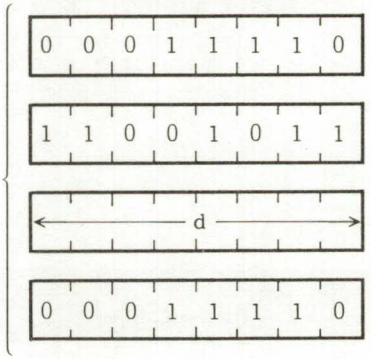
rとオブジェクト・コードは次のとおり。

	オブジェクト・コード	
RR r	1 1 0 0 1 0 1 1	CB
	0 0 0 1 1 ← r →	
RR (HL)	1 1 0 0 1 0 1 1	CB
	0 0 0 1 1 1 1 0	1E

RR （IX+d）



RR （IY+d）



レジスタ	r
B ＝	0 0 0
C ＝	0 0 1
D ＝	0 1 0
E ＝	0 1 1
H ＝	1 0 0
L ＝	1 0 1
A ＝	1 1 1

実　行：

命　令	Mサイクル	Tステート	実行時間(4MHz)
RR　r	2	8（4＋4）	2.00μs
RR　（HL）	4	15（4＋4＋4＋3）	3.75μs
RR　（IX＋d）	6	23（4＋4＋3＋5＋4＋3）	5.75μs
RR　（IY＋d）	6	23（4＋4＋3＋5＋4＋3）	5.75μs

フラグ：

S　：結果が負ならばセット、他の場合はリセット。

Z　：結果が零ならばセット、他の場合はリセット。

H　：リセット。

P／V：パリティが偶数ならばセット、他の場合はリセット。

N　：リセット。

C　：ビット0からのデータ。

例　：レジスタ・ペアＨＬの内容が3200Hで、メモリの3200H番地とキャリ・フラグの内容が次のとき、

CY	7	6	5	4	3	2	1	0
1	1	0	0	0	1	0	0	1

RR (HL)

を実行すると、メモリの3200H番地とキャリ・フラグの内容は次のようになる。

CY	7	6	5	4	3	2	1	0
1	1	1	0	0	0	1	0	0

SLA m　；CY ← 7 ← 0 ← 0（m）　（オペランド　m）

オペランドmの内容を左へシフトする。

すなわち、ビット０の内容をビット１にコピーし、コピー前のビット１の内容をビット２にコピーする。

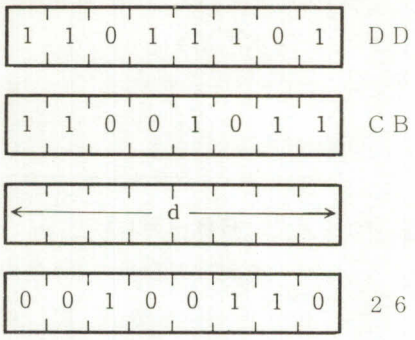
同様な操作を順次行う。ビット７の内容はキャリ・フラグにコピーし、ビット０はリセットする。

オペランドmにはｒ、(ＨＬ)、(ＩＸ+d)、(ＩＹ+d) が用いられる。

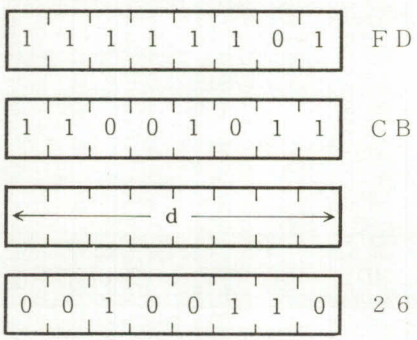
ｒとオブジェクト・コードは次のとおり。

	オブジェクト・コード	
SLA r	1 1 0 0 1 0 1 1	CB
	0 0 1 0 0 ← r →	
SLA (HL)	1 1 0 0 1 0 1 1	CB
	0 0 1 0 0 1 1 0	26

SLA （IX+d）



SLA （IY+d）



レジスタ	r
B	000
C	001
D	010
E	011
H	100
L	101
A	111

実行：命令	Mサイクル	Tステート	実行時間(4MHz)
SLA r	2	8(4＋4)	2.00μs
SLA （HL）	4	15(4＋4＋4＋3)	3.75μs
SLA （IX＋d）	6	23(4＋4＋3＋5＋4＋3)	5.75μs
SLA （IY＋d）	6	23(4＋4＋3＋5＋4＋3)	5.75μs

フラグ：
S ：結果が負ならばセット、他の場合はリセット。
Z ：結果が零ならばセット、他の場合はリセット。
H ：リセット。
P／V：パリティが偶数ならばセット、他の場合はリセット。
N ：リセット。

C　：ビット７からのデータ。

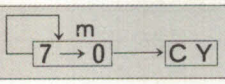
例　：レジスタＬの内容が次のとき、

CY	7	6	5	4	3	2	1	0
	1	0	0	1	1	1	0	1

SLA L

を実行すると、レジスタＬとキャリ・フラグの内容は次のようになる。

CY	7	6	5	4	3	2	1	0
1	0	0	1	1	1	0	1	0

SRA　m　； 

（オペランド　m）

オペランドｍの内容を右へシフトする。

ビット７の内容をビット６にコピーし、コピー前のビット６の内容をビット５にコピーする。同様な操作を順次行う。ビット０の内容はキャリ・フラグにコピーし、ビット７の内容は変わらない。

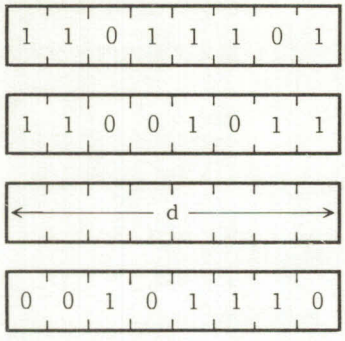
オペランドｍにはｒ、（ＨＬ）、（ＩＸ＋d）、（ＩＹ＋d）が用いられる。

ｒとオブジェクト・コードは次のとおり。

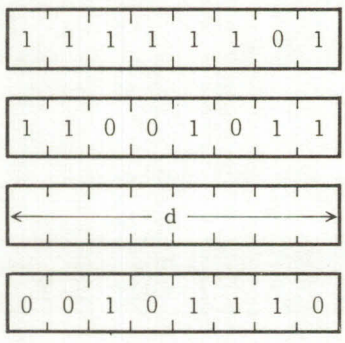
オブジェクト・コード

SRA　r	1 1 0 0 1 0 1 1	CB
	0 0 1 0 1 ← r →	
SRA　(HL)	1 1 0 0 1 0 1 1	CB
	0 0 1 0 1 1 1 0	2E

SRA （IX+d）



SRA （IY+d）



レジスタ	r
B ＝	0 0 0
C ＝	0 0 1
D ＝	0 1 0
E ＝	0 1 1
H ＝	1 0 0
L ＝	1 0 1
A ＝	1 1 1

実　行：命　令	Mサイクル	Tステート	実行時間(4MHz)
SRA　r	2	8（4＋4）	2.00μs
SRA （HL）	4	15（4＋4＋4＋3）	3.75μs
SRA （IX+d）	6	23（4＋4＋3＋5＋4＋3）	5.75μs
SRA （IY+d）	6	23（4＋4＋3＋5＋4＋3）	5.75μs

フラグ：　S　：結果が負ならばセット、他の場合はリセット。

Z　：結果が零ならばセット、他の場合はリセット。

H　：リセット。

P／V：パリティが偶数ならばセット、他の場合はリセット。

N　：リセット。

C　：ビット０からのデータ。

例　：インデックス・レジスタＩＸの内容が2400Hで、メモリの2403H番地の内容が次のとき、

CY	7	6	5	4	3	2	1	0
	1	0	1	0	1	1	1	0

SRA (IX＋3H)

を実行すると、メモリの2403H番地とキャリ・フラグの内容は次のようになる。

CY	7	6	5	4	3	2	1	0
0	1	1	0	1	0	1	1	1

SRL　m　；$0 \rightarrow \boxed{7 \overset{m}{\rightarrow} 0} \rightarrow \boxed{CY}$

（オペランド　m）

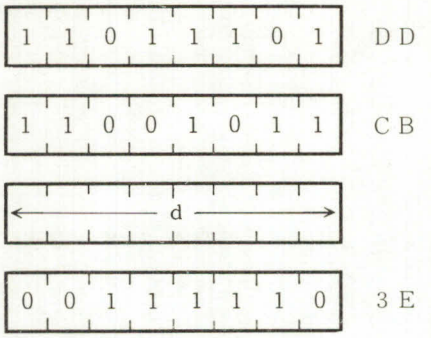
オペランドmの内容を右へシフトする。ビット７の内容はビット６へ、ビット６の内容はビット５へ、と順次シフトする。ビット０の内容はキャリ・フラグに移され、ビット７はリセットされる。

オペランドmにはＲＬＣ命令と同じように、r（A、B、C、D、H、Lレジスタ）、（HL）、（ＩＸ+d）、（ＩＹ+d）が用いられる。

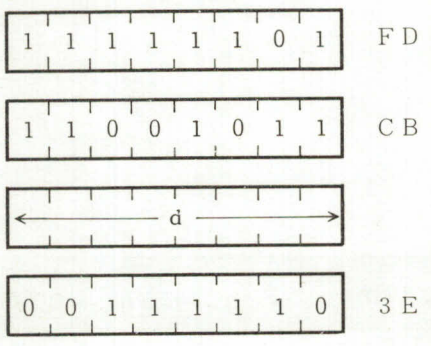
rとオブジェクト・コードは次のとおり。

	オブジェクト・コード	
SRL　r	1 1 0 0 1 0 1 1	CB
	0 0 1 1 1 ← r →	
SRL （HL）	1 1 0 0 1 0 1 1	CB
	0 0 1 1 1 1 1 0	3E

SRL （IX＋d）



SRL （IY＋d）



レジスタ	r
B ：	0 0 0
C ：	0 0 1
D ：	0 1 0
E ：	0 1 1
H ：	1 0 0
L ：	1 0 1
A ：	1 1 1

実　行： 命　令	Mサイクル	Tステート	実行時間(4MHz)
SRL　r	2	8（4＋4）	2.00μs
SRL　（HL）	4	15（4＋4＋4＋3）	3.75μs
SRL　（IX＋d）	6	23（4＋4＋3＋5＋4＋3）	5.75μs
SRL　（IY＋d）	6	23（4＋4＋3＋5＋4＋3）	5.75μs

フラグ：　S　：結果が負ならばセット、他の場合はリセット。

Z　：結果が零ならばセット、他の場合はリセット。

H　：リセット。

P／V：パリティが偶数ならばセット、他の場合はリセット。

N　：リセット。

C　：ビット０の内容が入る。

例　：Ｂレジスタの内容が次のようなとき、

CY	7	6	5	4	3	2	1	0
	1	0	0	1	0	0	1	1

SRL B

を実行すると、Ｂレジスタとキャリ・フラグは次のようになる。

CY	7	6	5	4	3	2	1	0
1	0	1	0	0	1	0	0	1

RLD　；A 7 4 3 0　7 4 3 0 (HL)

（オペランド　なし）

レジスタ・ペアＨＬの内容で指定されるメモリの内容の下位４ビット（０～３ビット）の内容をその上位４ビット（４～７ビット）へ移し、上位４ビットの内容をアキュムレータの下位４ビットに移す。そしてアキュムレータの下位４ビットの内容をレジスタ・ペアＨＬの内容で指定されるメモリの下位４ビットに移す。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	0	1	1	0	1	ED
0	1	1	0	1	1	1	1	6F

実　行：Mサイクル：５，Ｔステート：18(４＋４＋３＋４＋３)，実行時間：4.50μs(4MHz)

フラグ：　S　：アキュムレータの内容が負ならばセット、他の場合はリセット。

Z　：アキュムレータの内容が零ならばセット、他の場合はリセット。

H　：リセット。

P／V：アキュムレータのパリティが偶数ならばセット、他の場合はリセット。

N　：リセット。

C　：変化せず。

例　：アキュムレータおよびメモリが次のようなとき、

7	6	5	4	3	2	1	0	
0	1	1	0	1	1	1	1	アキュムレータ
1	0	1	1	0	0	1	1	３５３５H番地

ＨＬレジスタ：３５３５H

ＲＬＤ

を実行すると、アキュムレータとメモリの3535H番地の内容は次のようになる。

7	6	5	4	3	2	1	0	
0	1	1	0	1	0	1	1	アキュムレータ
0	0	1	1	1	1	1	1	３５３５H番地

ＲＲＤ　； A 7 4 3 0 ⇄ 7 4 3 0 (HL)　　（オペランド　なし）

レジスタ・ペアＨＬの内容で指定されるメモリの内容の下位４ビット（０～３ビット）の内容をアキュムレータの下位４ビットに移し、アキュムレータの下位４ビットの内容をレジスタ・ペアＨＬの内容で指定されるメモリの内容の上位４ビット（４～７ビット）に移す。そして、上位４ビットの内容を下位４ビットに移す。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	0	1	1	0	1	ＥＤ
0	1	1	0	0	1	1	1	６７

実　行：Mサイクル：５，Ｔステート：18(４＋４＋３＋４＋３)，実行時間：4.50μs(4MHz)

フラグ：　Ｓ　：アキュムレータの内容が負ならばセット、他の場合はリセット。

Ｚ　：アキュムレータの内容が零ならばセット、他の場合はリセット。

Ｈ　：リセット。

P／V：アキュムレータのパリティが偶数ならばセット、他の場合はリセット。

N　：リセット。

C　：変化せず。

例　：アキュムレータおよびメモリの内容が次のようなとき、

7	6	5	4	3	2	1	0	
1	1	0	0	0	0	0	1	アキュムレータ
0	1	0	0	0	0	1	0	３４３４H番地

ＨＬレジスタ：３４３４H

ＲＲＤ

を実行すると、アキュムレータとメモリの3434H番地の内容は次のようになる。

7	6	5	4	3	2	1	0	
1	1	0	0	0	0	1	0	アキュムレータ
0	0	0	1	0	1	0	0	３４３４H番地

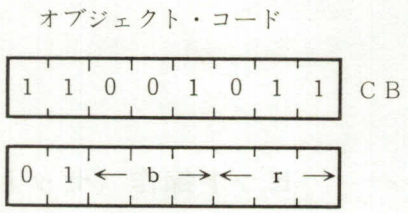
ビット操作（セット，リセット，テスト）グループ

BIT b, r ; Z←$\overline{r_b}$

（オペランド b, r）

この命令の実行後、指示されたレジスタのうち、指示されたビットの内容の補数がFレジスタのZフラグに入る。

オブジェクト・コードは次のとおり。



bとrは次のとおり。

テスト・ビット	b	レジスタ	r
0	0 0 0	B	0 0 0
1	0 0 1	C	0 0 1
2	0 1 0	D	0 1 0
3	0 1 1	E	0 1 1
4	1 0 0	H	1 0 0
5	1 0 1	L	1 0 1
6	1 1 0	A	1 1 1
7	1 1 1		

実　行：Mサイクル：2，Tステート：8（4＋4），実行時間：2.00μs（4MHz）

フラグ：　S　：不定。
　　Z　：指定されたビットの内容が零ならばセット、他の場合はリセット。
　　H　：セット。
　　P／V：不定。
　　N　：リセット。
　　C　：変化せず。

例　：Bレジスタのビット2が零のとき、

```
BIT 2, B
```

を実行すると、Zフラグは1となり、Bレジスタの内容は変わらない。

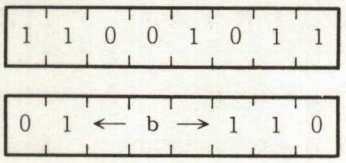
BIT b, (HL) ; Z← $\overline{(HL)_b}$

（オペランド b,（HL））

この命令の実行後、レジスタ・ペアＨＬの内容で示されるメモリの内容のうち、指示されたビットの補数がＦレジスタのＺフラグに入る。

オブジェクト・コードは次のとおり。

オブジェクト・コード



bについては次のとおり。

テスト・ビット	b
0	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1

実　行：Mサイクル：3，Tステート：12（4＋4＋4），実行時間：3.00μs（4MHz）

フラグ：　S　：不定。
　　Z　：指定されたビットが零ならばセット、他の場合はリセット。
　　H　：セット。
　　P／V：不定。
　　N　：リセット。
　　C　：変化せず。

例　：ＨＬレジスタおよびメモリの内容が次のようなとき、
　　ＨＬ：１０００H
　　１０００H 番地のビット４：１
　　BIT 4, (HL)
　　を実行すると、次のようになる。

ＨＬ：１０００Ｈ

１０００Ｈ 番地のビット４：１

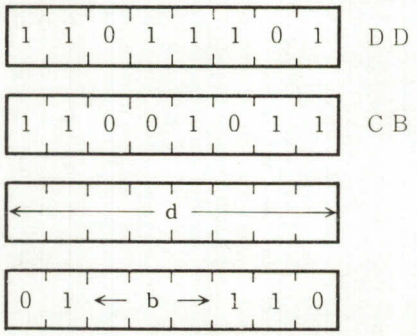
Ｚフラグ：０

ＢＩＴ　ｂ，（ＩＸ＋ｄ）　；Ｚ←$\overline{(IX+d)_b}$　（オペランド　ｂ，（ＩＸ＋ｄ））

この命令の実行後、インデックス・レジスタＩＸの内容とディスプレイスメントｄとの和によって示されるメモリの内容のうち、指示されたビットの補数が、Ｆレジスタのフラグに入る。

オブジェクト・コードは次のとおり。

オブジェクト・コード



ｂについては次のとおり。

テスト・ビット	ｂ
0	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1

実　行　：　Mサイクル：5，Tステート：20(4＋4＋3＋5＋4)，実行時間：5.00μs（4MHz）

フラグ　：　S　：不定。

Z　：指定されたビットが零ならばセット、他の場合はリセット。

H　：セット。

P／V：不定。

N　：リセット。

C　：変化せず。

例　　：ＩＸレジスタおよびメモリの内容が次のようなとき、

ＩＸ：４０００H

４００４H 番地のビット６：１

BIT 6, (IX＋4H)

を実行すると、次のようになる。

ＩＸ：４０００H

４００４H 番地のビット６：１

Ｚフラグ：０

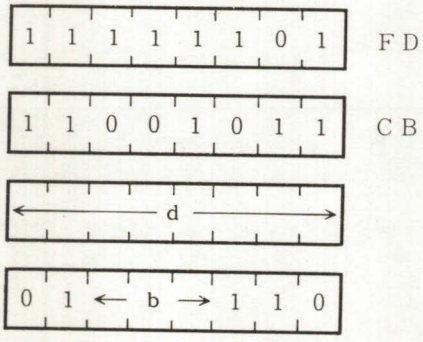
BIT b, (IY＋d)　；Z←$\overline{(IY+d)_b}$

（オペランド　b, (ＩＹ＋ｄ)）

この命令の実行後、インデックス・レジスタＩＹの内容とディスプレイスメントｄとの和によって示されるメモリの内容のうち、指示されたビットの補数が、Ｆレジスタのフラグに入る。

オブジェクト・コードは次のとおり。

オブジェクト・コード



bについては次のとおり。

テスト・ビット	b
0	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1

実　行　：　Mサイクル：5，Tステート：20（4＋4＋3＋5＋4），実行時間：5.00μs(4MHz)

フラグ　：　S　：不定。

Z　：指定されたビットが零ならばセット、他の場合はリセット。

H　：セット。

P／V：不定。

N　：リセット。

C　：変化せず。

例　：　IYレジスタおよびメモリの内容が次のようなとき、

IY：8000H

8006H 番地のビット5：1

```
BIT 5, (IY+6H)
```

を実行すると、次のようになる。

IY：8000H

8006H 番地のビット5：1

Zフラグ：0

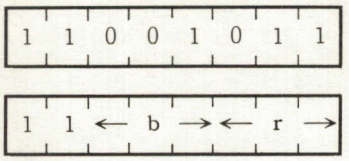
SET b, r　；$r_b \leftarrow 1$

（オペランド　b，r）

レジスタrのビットbがセットされる。

オブジェクト・コードは次のとおり。

オブジェクト・コード



bとrについては次のとおり。

ビット	b	レジスタ	r
0	0 0 0	B	0 0 0
1	0 0 1	C	0 0 1
2	0 1 0	D	0 1 0
3	0 1 1	E	0 1 1
4	1 0 0	H	1 0 0
5	1 0 1	L	1 0 1
6	1 1 0	A	1 1 1
7	1 1 1		

実　行　：　Mサイクル：2，Tステート：8（4＋4），実行時間：2.00μs(4MHz)

フラグ　：　変化せず。

例　：　SET 4，A

を実行すると、Aレジスタのビット4がセットされる。

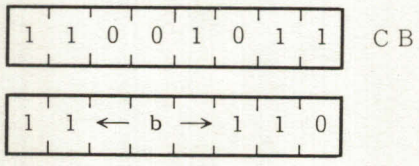
SET b，(HL)　；(HL)$_b$←1

（オペランド　b，(HL)）

レジスタ・ペアHLの内容で示されるメモリのビットbがセットされる。

オブジェクト・コードは次のとおり。

オブジェクト・コード



ビット	b
0	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1

実　行　：　Mサイクル：4，Tステート：15（4＋4＋4＋3），実行時間：3.75μs(4MHz)

フラグ　：　変化せず。

例　　：　レジスタ・ペアＨＬの内容が4400Hのとき、

```
SET 4, (HL)
```

を実行すると、メモリの4400H番地のビット４がセットされる。

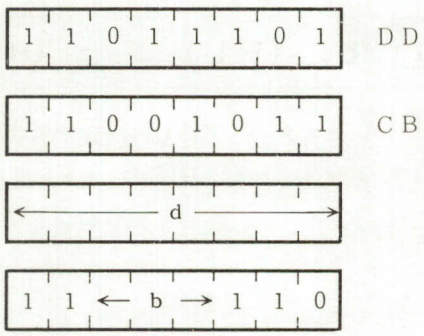
SET b, (IX+d)　　;(IX+d)$_b$←1

（オペランドb，（ＩＸ＋ｄ））

インデックス・レジスタＩＸの内容とディスプレイスメントｄとの和で示されるメモリの内容のビットｂがセットされる。

オブジェクト・コードは次のとおり。

オブジェクト・コード



bは次のとおり。

ビット	b
0	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1

実　行　：　Mサイクル：4，Tステート：23(4+4+3+5+4+3)，実行時間：5.75μs(4MHz)

フラグ　：　変化せず。

例　　：　インデックス・レジスタＩＸの内容が1500Hのとき、

```
SET  0, (IX+3)
```

を実行すると、メモリの1503H番地のビット0がセットされる。

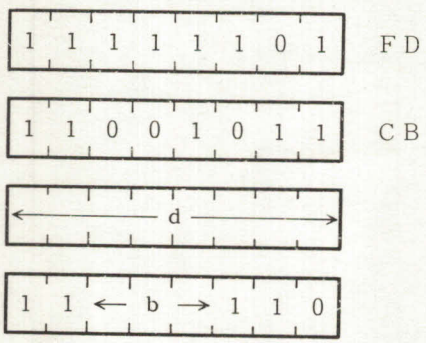
SET　b,（ＩＹ＋ｄ）　；　$(IY+d)_b \leftarrow 1$

（オペランド　b,（ＩＹ＋ｄ））

インデックス・レジスタＩＹの内容とディスプレイスメントｄとの和で示されるメモリの内容のビットｂがセットされる。

オブジェクト・コードは次のとおり。

オブジェクト・コード



bは次のとおり。

ビット	b
0	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1

実　行　：　Mサイクル：6，Tステート：23(4＋4＋3＋5＋4＋3)，実行時間：5.75μs(4MHz)

フラグ　：　変化せず。

例　　：　インデックス・レジスタＩＹの内容が4200Hのとき、

```
SET  0, (IY＋3)
```

を実行すると、メモリの4203H番地のビット0がセットされる。

RES　b，m　　；$m_b \leftarrow 0$

（オペランド　b，m）

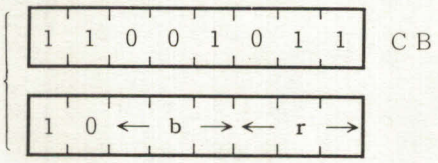
オペランドmのビットbをリセットする。

オペランドmはＳＥＴ命令と同様に、r（A、B、C、D、E、H、Lレジスタ）、(HL)、（ＩＸ＋d)、(ＩＹ＋d）が用いられる。

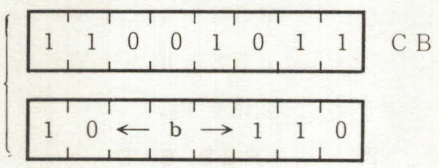
オブジェクト・コードは次のとおり。

オブジェクト・コード

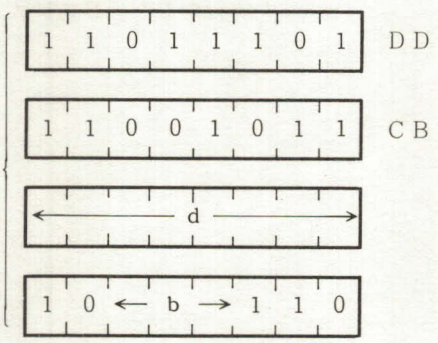
RES　b，r



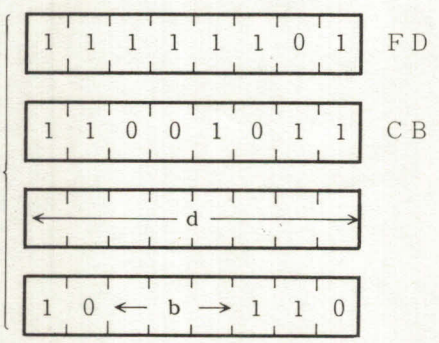
RES　b，（HL）



RES　b，（IX＋d）



RES　b，（IY＋d）



bとrは次のとおり。

リセット・ビット	b	レジスタ	r
0	000	B	000
1	001	C	001
2	010	D	010
3	011	E	011
4	100	H	100
5	101	L	101
6	110	A	111
7	111		

実　行　：　命　令	Mサイクル	Tステート	実行時間(4MHz)
RES　b，r	2	8(4+4)	2.00μs
RES　b，(HL)	4	15(4+4+4+3)	3.75μs
RES　b，(IX+d)	6	23(4+4+3+5+4+3)	5.75μs
RES　b，(IY+d)	6	23(4+4+3+5+4+3)	5.75μs

フラグ　：　変化せず。

例　：　**RES 6，D**

を実行すると、Dレジスタのビット6がリセットされる。

――― ジャンプ　グループ ―――

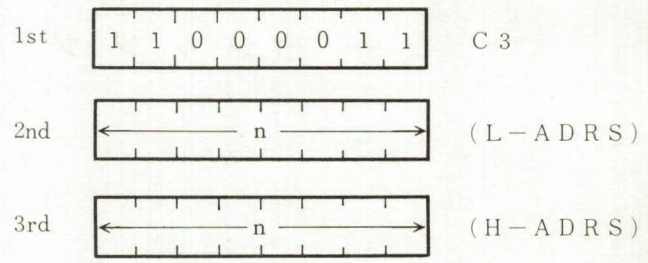
JP nn ; PC←nn

（オペランド nn）

オペランドnnをプログラム・カウンタPCにロードする。nnは次に実行される命令のある番地を指定する。オブジェクト・コードの2バイト目には番地の下位バイトを、3バイト目には上位バイトを入れる。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行 ： Mサイクル：3，Tステート：10(4＋3＋3)，実行時間：2.50μs（4 MHz）

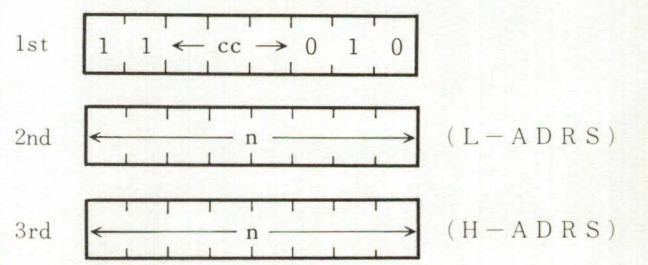
フラグ ： 変化せず。

JP cc, nn ; If cc true, PC←nn

（オペランド cc, nn）

条件ccとフラグの状態が一致すれば、PCにオペランドnnをロードし、次の命令はnn番地から始まる。もし一致しなければ、プログラム・カウンタの内容を増し、このJP命令の次にある命令を実行する。ccはフラグ・レジスタFの4つのステータスのいずれかでありジャンプ命令実行の条件ビットとなる。オブジェクト・コードの2バイト目には番地の下位バイトを、3バイト目には上位バイトを入れる。ccとオブジェクト・コードは次のとおり。

オブジェクト・コード



cc	条　件	フラグ	cc	条　件	フラグ
000	NZ(ノン・ゼロ)	Z	100	PO(パリティ奇数)	P／V
001	Z　(ゼロ)	Z	101	PE(パリティ偶数)	P／V
010	NC(ノン・キャリ)	C	110	P　(正)	S
011	C　(キャリ)	C	111	M　(負)	S

実　行　：　Mサイクル：3，Tステート：10(4＋3＋3)，実行時間：2.50μs（4 MHz）

フラグ　：　変化せず。

例　　：　キャリ・フラグがセットされていて、1600H番地の内容が02Hのとき、

```
JP  C, 1600H
```

を実行すると、プログラム・カウンタの内容は1600Hとなり、次のマシン・サイクルではＣＰＵは1600H番地から命令コード02Hをとり込む。

ＪＲ　e　　；PC←PC＋e

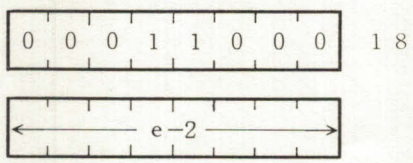
（オペランド　e）

このＪＲ命令の在る番地から、ディスプレイスメントｅだけ離れた番地へ無条件にジャンプする。プログラム・カウンタＰＣの内容にｅを加えた値をＰＣにストアし、この新しい内容の指定する番地から次の命令をとり込む。（次頁【注意】参照）

ジャンプできる範囲（ｅの許される範囲）は、−126バイトから＋129バイトまでである。

ＪＲ命令をフェッチ後、ＰＣの内容は2だけ増しているので、実際は、ＰＣにはｅ−2が加えられる。オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行　：　Mサイクル：3，Tステート：12(4＋3＋5)，実行時間：3.00μs（4 MHz）

フラグ　：　変化せず。

例　　：　520H番地から、5H番地前方へジャンプするとき、次のステートメントを使用する。

```
JR  $＋5
```

【注意】本マニュアルでは、相対アドレッシング・モードの説明において、オブジェクト・コードとディスプレイスメントの関係を明らかにするためにオペランドに記号ｅを用いている。実際のアセンブラではオペランドにディスプレイスメントを表わす数（直接数値、ラベル、式など）を記述することは許されず、ジャンプ先を表わす数（直接数値、ラベル、式など）を記述しなければならない。このオブジェクト・コードとＰＣの指定する位置は次のようになる。

位　置	オブジェクト・コード
５２０	１８　（ＪＲ　＄＋５）
５２１	０３
５２２	－
５２３	－
５２４	－
５２５	←――――ジャンプ後のＰＣ

ＪＲ　Ｃ，ｅ　；If C=0, continue/If C=1, PC←PC+e

（オペランド　Ｃ，ｅ）

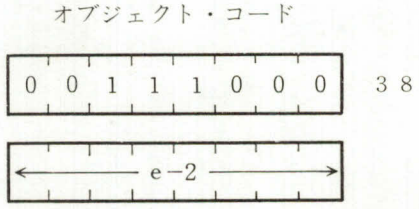
キャリ・フラグの内容によって条件付きジャンプをする。

Ｃフラグが１の場合、プログラム・カウンタＰＣの内容にｅを加えて、このＰＣの新しい値が指定する位置から次の命令をとり込む。（上記【注意】参照）

Ｃフラグが０の場合、次にある命令を実行する。

ジャンプできる範囲（ｅの許される範囲）は、－126バイトから＋129バイトまでである。

オブジェクト・コードは次のとおり。



実　行　：　Ｃ＝１のとき、

Ｍサイクル：３，Ｔステート：12(４＋３＋５)，実行時間：3.00μs（４MHz）

Ｃ＝０のとき、

Ｍサイクル：２，Ｔステート：７(４＋３)，実行時間：1.75μs（４MHz）

フラグ ： 変化せず。

例 ： キャリ・フラグがセットされて、520H番地から4H番地後方へジャンプするとき、次のステートメントを使う。

```
JR  C, $－4
```

オブジェクト・コードとプログラム・カウンタＰＣの内容は次のようになる。

位　置	オブジェクト・コード
５１Ｃ	←――――――ジャンプ後のＰＣ
５１Ｄ	―
５１Ｅ	―
５１Ｆ	―
５２０	３８　（ＪＲ　Ｃ，＄－４）
５２１	ＦＡ　（－６の、２の補数表現）

ＪＲ　ＮＣ，ｅ　　；If Ｃ＝１，continue/If Ｃ＝０，ＰＣ←ＰＣ＋ｅ　（オペランド　ＮＣ，ｅ）

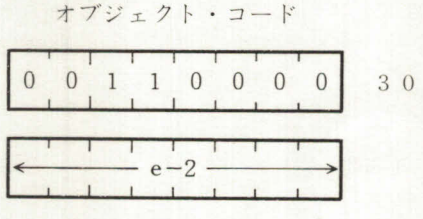
キャリ・フラグの内容によって条件付きジャンプをする。

フラグの内容が０ならば、プログラム・カウンタＰＣの内容にｅを加え、このＰＣの新しい内容が指定する位置から次の命令をフェッチする。【注意】

フラグの内容が１ならば、次にある命令を実行する。

ジャンプできる範囲（ｅの許される範囲）は、－126バイトから＋129バイトまでである。

オブジェクト・コードは次のとおり。



【注意】アセンブラにおけるオペランドの記述法については138ページの注意に同じ。

実　行　：　C＝0のとき、

Mサイクル：3，Tステート：12（4＋3＋5），実行時間：3.00μs（4 MHz）

C＝1のとき、

Mサイクル：2，Tステート：7（4＋3），実行時間：1.75μs（4 MHz）

フラグ　：　変化せず。

例　　：　キャリ・フラグがリセットされて、ジャンプ命令を繰り返させるとき、ステートメントは次のようになる。

```
JR NC, $
```

オブジェクト・コードとプログラム・カウンタPCの内容は次のようになる。

位　置	オブジェクト・コード	
520	30	←――ジャンプ後のPC
521	FE	

JR Z, e　；If Z＝0, continue/If Z＝1, PC←PC＋e　（オペランド　Z，e）

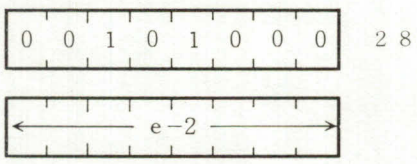
ゼロ・フラグの内容によって条件付きジャンプをする。

フラグの内容が1ならば，プログラム・カウンタPCの内容にeを加え、このPCの新しい内容が指定する位置から次の命令をフェッチする。【注意】

フラグの内容が0ならば、次にある命令を実行する。

ジャンプできる範囲（eの許される範囲）は、－126バイトから＋129バイトまでである。

オブジェクト・コードは次のとおり。



【注意】アセンブラにおけるオペランドの記述法については138ページの注意に同じ。

実　行　：　Z＝1のとき、

Mサイクル：3，Tステート：12（4＋3＋5），実行時間：3.00μs（4 MHz）

Z＝0のとき、

Mサイクル：2，Tステート：7（4＋3），実行時間：1.75μs（4 MHz）

フラグ　：　変化せず。

例　　：　ゼロ・フラグがセットで、450H番地から前方へ5バイトジャンプするとき、ステートメントは次のようになる。

```
JR  Z, $+5
```

オブジェクト・コードとプログラム・カウンタＰＣの内容は次のようになる。

位　置	オブジェクト・コード
450	28　（JR　Z, $+5）
451	03
452	—
453	—
454	—
455	←————ジャンプ後のＰＣ

JR　NZ, e　　; If Z＝1, continue/If Z＝0, PC←PC＋e　（オペランド NZ,e）

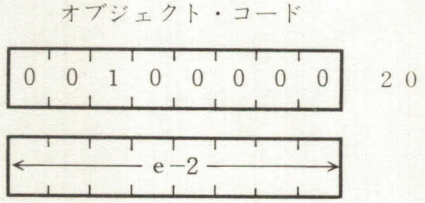
ゼロ・フラグの内容によって条件付きジャンプをする。

フラグの内容が0ならば、プログラム・カウンタＰＣの内容にｅを加え、このＰＣの新しい内容が指定する位置から、次の命令をフェッチする。【注意】

フラグの内容が1ならば、次にある命令を実行する。

ジャンプできる範囲（ｅの許される範囲）は－126バイトから＋129バイトまでである。

オブジェクト・コードは次のとおり。



【注意】アセンブラにおけるオペランドの記述法については138ページの注意に同じ。

実　行：Z＝0のとき、

Mサイクル：3，Tステート：12（4＋3＋5），実行時間：3.00μs（4 MHz）

Z＝1のとき、

Mサイクル：2，Tステート：7（4＋3），実行時間：1.75μs（4 MHz）

フラグ：変化せず。

例　：ゼロ・フラグがリセットされており、450H番地から後方へ4バイトジャンプするとき、ステートメントは次のようになる。

```
JR NZ, $－4
```

オブジェクト・コードとプログラム・カウンタPCの内容は次のようになる。

位　置	オブジェクト・コード
44C	←――――――ジャンプ後のPC
44D	—
44E	—
44F	—
450	20（JR NZ, $－4）
451	FA（－6の、2の補数表現）

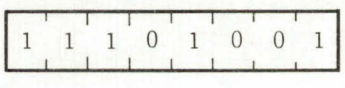
JP（HL）　；PC←HL

（オペランド（HL））

プログラム・カウンタPCにレジスタ・ペアHLの内容をロードする。このPCの新しい内容が指定する位置から次の命令をフェッチする。

オブジェクト・コードは次のとおり。

オブジェクト・コード



E9

実　行　：　Mサイクル：1，Tステート：4，実行時間：1.00μs（4 MHz）

フラグ　：　変化せず。

例　　：　プログラム・カウンタの内容が1200Hとなり、レジスタ・ペアＨＬの内容が2400Hのとき、

```
JP (HL)
```

を実行すると、プログラム・カウンタの内容は2400Hとなり、次に2400Hの命令を実行する。

JP (IX)　；PC←IX

（オペランド（ＩＸ））

プログラム・カウンタＰＣにインデックス・レジスタＩＸの内容をロードする。このＰＣの新しい内容が指定する位置から次の命令をフェッチする。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	0	1	1	1	0	1	DD
1	1	1	0	1	0	0	1	E9

実　行　：　Mサイクル：2，Tステート：8（4＋4），実行時間：2.00μs（4 MHz）

フラグ　：　変化せず。

例　　：　プログラム・カウンタＰＣの内容が2000Hで、インデックス・レジスタＩＸの内容が3600Hのとき、

```
JP (IX)
```

を実行すると、プログラム・カウンタＰＣの内容は3600Hとなり、次に3600Hの命令を実行する。

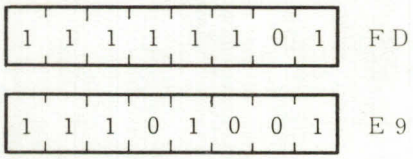
JP （IY） ；PC←IY

（オペランド（IY））

プログラム・カウンタＰＣにインデックス・レジスタＩＹの内容をロードする。このＰＣの新しい内容が指定する位置から次の命令をフェッチする。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行：Mサイクル：2，Tステート：8（4＋4），実行時間：2.00μs（4 MHz）

フラグ：変化せず。

例　：プログラム・カウンタＰＣの内容が1600Hで、インデックス・レジスタＩＹの内容が2200Hのとき、

JP (IY)

を実行すると、プログラム・カウンタＰＣの内容は2200Hとなり、次に2200Hの命令を実行する。

DJNZ e ；————————

（オペランド　e）

レジスタの内容で、ジャンプするか否かが決められることを除けば、他の条件付きジャンプ命令と同様である。

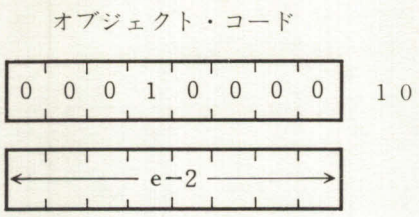
レジスタＢの内容を１だけ減じる。このとき、Ｂの内容が０でなければ、プログラム・カウンタＰＣにディスプレイスメントｅを加え、このＰＣの新しい内容が指定する位置から次の命令をフェッチする。

Ｂの内容が０であれば、次にある命令を実行する。【注意】

ジャンプできる範囲（ｅの許される範囲）は−126バイトから＋129バイトまでである。

【注意】アセンブラにおけるオペランドの記述法については138ページの注意に同じ。

オブジェクト・コードは次のとおり。



実　行：Ｂキ０のとき、

Ｍサイクル：３，Ｔステート：13（５＋３＋５），実行時間：3.25μs（４MHz）

Ｂ＝０のとき、

Ｍサイクル：２，Ｔステート：８（５＋３），実行時間：2.00μs（４MHz）

フラグ：変化せず。

例：入力バッファＩＮＢＵＦから出力バッファＯＵＴＢＵＦへ１バイトずつ移し、ＣＲを発見するかあるいは80バイト移すまで、それを続けるルーチンを以下に示す。

これはＤＪＮＺ命令を使用する典型的ルーチンである。

```
        ＬＤ     B，80          ；カウンタをセット。
        ＬＤ     HL，INBUF      ；ポインタをセット。
        ＬＤ     DE，OUTBUF
ＬＯＯＰ：ＬＤ   Ａ，（ＨＬ）     ；入力バッファから１バイト取ってくる。
        ＬＤ     （ＤＥ），Ａ    ；出力バッファへストア。
        ＣＰ     ０ＤＨ         ；それはＣＲ（キャリッジ・リターン）か？
        ＪＲ     Ｚ，ＤＯＮＥ    ；ＣＲなら、ＤＯＮＥへとべ。
        ＩＮＣ   ＨＬ           ；ポインタを１進める。
        ＩＮＣ   ＤＥ           ；ポインタを１進める。
        ＤＪＮＺ  ＬＯＯＰ      ；80バイト移していなければ、ＬＯＯＰへ戻る。

ＤＯＮＥ：－－－－－－－－－    ；次のルーチンの始まり。
```

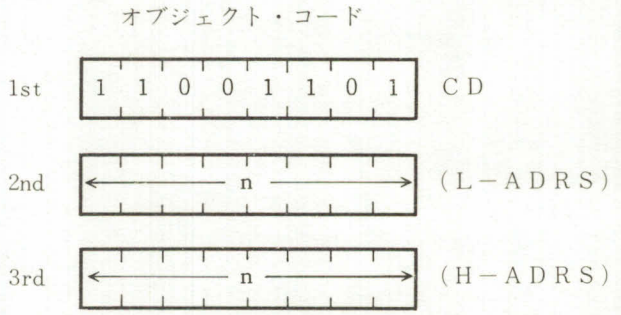
―― コール、リターン　グループ ――

CALL nn ; (SP−1)←PC$_H$, (SP−2)←PC$_L$, PC←nn （オペランド nn）

メモリ上でＣＡＬＬ命令の次にある命令に対応するＰＣの値を、現在のＳＰで指示されるメモリ・バイトの次の２バイトにプッシュした後、オペランドnnをＰＣにロードする。このＰＣの新しい内容で指定される位置以降にサブルーチンをおく。（サブルーチンが終了すると、リターン命令を使用し、先にプッシュした次の命令開始番地をポップしてＰＣに移すことにより、元のプログラムに戻る。）

プッシュはまずスタック・ポインタＳＰのそのときの内容を１減じ、その値が指定する位置にＰＣの上位バイト（１バイト目）をロードする。その後、さらにＳＰの内容を１減じ、その値が指定する位置（スタックのトップ）にＰＣの下位バイトをロードする。

オブジェクト・コードの２バイト目にはnnの下位バイトを入れ、３バイト目には、上位バイトを入れる。オブジェクト・コードは次のとおり。



実　行：Mサイクル：5，Tステート：17（4＋3＋4＋3＋3），実行時間：4.25μs(4MHz)

フラグ：変化せず。

例　：プログラム・カウンタＰＣの内容が2B36Hで、スタック・ポインタＳＰの内容が1002Hであるとき、メモリの内容は次のようになる。

位　置	内　容
2B36H	CDH
2B37H	33H
2B38H	44H

このとき、命令をフェッチすると、ＣＰＵは３バイト命令ＣＤ3344H を得る。これはニーモニックでは、

CALL 4433H

である。これを実行すると、1001H 番地、1000H 番地の内容はそれぞれ2BH 、39Hとなり、ＳＰの内容は1000Hとなる。プログラム・カウンタＰＣの内容は4433H であり、

この番地から次の命令をフェッチする。

（ＰＣは、プッシュ実行前に３だけ増していることに注意）

CALL cc, nn ; If cc true : (SP−1)←PC_H, (SP−2)←PC_L, PC←nn　（オペランド cc, nn）

条件ccがフラグと一致すれば、プログラム・カウンタＰＣの内容をＳＰ－１，ＳＰ－２で指定される外部メモリ・スタックにプッシュした後、オペランドnnをＰＣにロードする。このＰＣの新しい内容が指定する位置からサブルーチンの最初の命令をフェッチする。（サブルーチンが終了すると、リターン命令を使用し先にプッシュした次の命令の開始番地をポップしてＰＣに移すことにより、元のプログラムに戻る。）

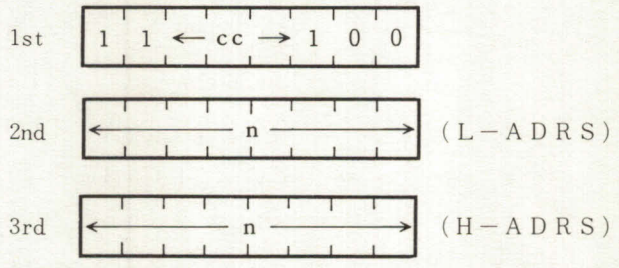
もしccがフラグと一致しなければ、次にある命令をフェッチする。

プッシュはまずスタック･ポインタＳＰのそのときの内容を１減じ、その値が指定する位置にＰＣの上位バイトをロードする。その後、さらにＳＰの内容を１減じ、その値が指定する位置にＰＣの下位バイトをロードする。

オブジェクト・コードの２バイト目にはnnの下位バイトを入れ、３バイト目には上位バイトを入れる。ccはフラグ・レジスタＦのビットの状態に対応する８種類のステータスの１つである。

ccとオブジェクト・コードは次のとおり。

オブジェクト・コード



ｃｃ	条　件	フラグ
０００	ＮＺ(ノン・ゼロ)	Ｚ
００１	Ｚ　(ゼロ)	Ｚ
０１０	ＮＣ(ノン・キャリ)	Ｃ
０１１	Ｃ　(キャリ)	Ｃ
１００	ＰＯ(パリティ奇数)	Ｐ／Ｖ
１０１	ＰＥ(パリティ偶数)	Ｐ／Ｖ
１１０	Ｐ　(正)	Ｓ
１１１	Ｍ　(負)	Ｓ

実　行：ccとフラグが一致すれば、

Mサイクル：5，Tステート：17（４＋３＋４＋３＋３），実行時間：4.25μs（4MHz）

ccとフラグが不一致であれば、

Mサイクル：3，Tステート：10（４＋３＋３），実行時間：2.50μs（4MHz）

フラグ：変化せず。

例　：レジスタの内容が次のとき、

プログラム・カウンタＰＣ：３Ｃ６Ａｈ

スタック・ポインタＳＰ　：２００２ｈ

キャリ・フラグ　　：リセット

CALL NC, 4155H

を実行する場合、メモリは次のようになる。

位　置	内　容
３Ｃ６Ａｈ	Ｄ４ｈ
３Ｃ６Ｂｈ	５５ｈ
３Ｃ６Ｃｈ	４１ｈ

この命令を実行すると、メモリおよびレジスタの内容は次のようになる。

ＰＣ：４１５５ｈ

ＳＰ：２０００ｈ

２００１ｈ番地：３Ｃｈ

２０００ｈ番地：６Ｄｈ

（ＰＣはプッシュの前に３だけ増していることに注意）

RET ; $PC_L \leftarrow (SP)$, $PC_H \leftarrow (SP+1)$, $SP \leftarrow SP+2$ （オペランド なし）

外部メモリ・スタックからプログラム・カウンタＰＣの前の値をポップして元のプログラムに戻る。スタック・ポインタＳＰの内容が指定する位置の内容をＰＣの下位バイトにロードし、ＳＰの内容を１だけ増す。さらにＳＰの新しい内容が指定する位置の内容をＰＣの上位バイトにロードし、ＳＰの内容を１だけ増す。

このＰＣの内容が指定する位置から次の命令をフェッチする。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	0	0	1	0	0	1

C９

実　行：Mサイクル：３，Tステート：10（４＋３＋３），実行時間：2.50μs（４MHz）

フラグ：変化せず。

例　：レジスタおよびメモリの内容が次のとき、

ＰＣ：４３４３H　　ＳＰ：３０００H

３０００H 番地：Ａ６H　　３００１H 番地：２４H

RET

を実行すると、レジスタの内容は次のようになる。

ＰＣ：２４Ａ６H　　ＳＰ：３００２H

RET cc ; If cc true : $PC_L \leftarrow (SP)$, $PC_H \leftarrow (SP+1)$, $SP \leftarrow SP+2$

（オペランド cc）

条件ccとフラグが一致すれば、外部メモリ・スタックからプログラム・カウンタＰＣの前の値をポップして、元のプログラムに戻る。

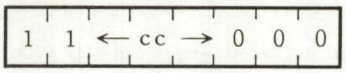
スタック・ポインタＳＰの内容が指定する位置の内容をＰＣの下位バイトにロードし、ＳＰの内容を１だけ増す。さらにＳＰの新しい内容が指定する位置の内容をＰＣの上位バイトにロードし、ＳＰの内容を１だけ増す。このＰＣの内容が指定する位置から次の命令をフェッチする。

ccとフラグが一致しなければ次にある命令を実行する。

ccはフラグ・レジスタＦのビットの状態に対応する８種類のステータスの１つである。

オブジェクト・コードとccは次のとおり。

オブジェクト・コード



cc	条　件	フラグ
000	NZ(ノン・ゼロ)	Z
001	Z　(ゼロ)	Z
010	NC(ノン・キャリ)	C
011	C　(キャリ)	C
100	PO(パリティ奇数)	P／V
101	PE(パリティ偶数)	P／V
110	P　(正)	S
111	M　(負)	S

実　行 ： ccがフラグと一致すれば、

Mサイクル：3，Tステート：11(5＋3＋3)，実行時間：2.75μs（4 MHz）

ccがフラグと不一致であれば、

Mサイクル：1，Tステート：5，実行時間：1.25μs（4 MHz）

フラグ ： 変化せず。

例　 ： レジスタおよびメモリの内容が次のとき、

PC ： 4646H　　SP ： 6000H

6000H番地 ： A7H　　6001H番地 ： 12H

Sフラグ ： セット

RET M

を実行すると、レジスタの内容は次のようになる。

PC ：12A7H　　SP ： 6002H

RETI　；Return from interrupt

（オペランド　なし）

割り込みサービス・ルーチンの終了時に使用する。

その実行内容は次のとおり。

1．割り込みからの復帰番地をプログラム・カウンタＰＣにロードする。
（ＲＥＴ命令と同じ）

2．割り込みルーチンの終了を入出力デバイスに通知する。この命令により、割り込みが重なった場合に優先順位の低いデバイスへのサービスを保留し、優先順位の高いデバイスへのサービスを行うという“割り込みのネスティング”の操作が可能となる。
ＩＦＦ２の状態はＩＦＦ１にコピーされる。

オブジェクト・コードは次のとおり。

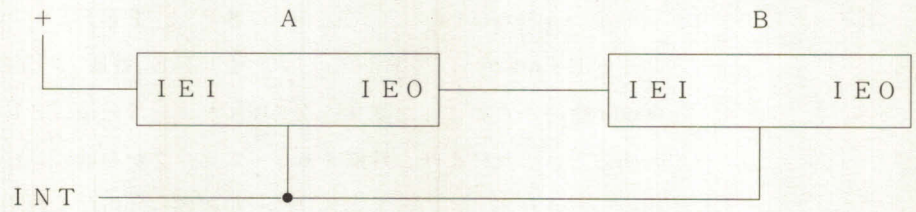
オブジェクト・コード

1	1	1	0	1	1	0	1	ED
0	1	0	0	1	1	0	1	4D

実　行：Mサイクル：4，Tステート：14(４＋４＋３＋３)，実行時間：3.50μs（４MHz）

フラグ：変化せず。

例：２個のデバイスA、Bが、デージー・チェーンの形に連結されており、AはBより優先順位を高くしてある。



Bが割り込みを発生し、それが認められたとする。
（ここで、ＢのＩＥＯが“Ｌ”になり、Ｂへのサービス中はＢより優先順位の低いデバイスからの割り込みのサービスは保留される。）
Aが割り込みを発生し、Bの割り込みのサービスは保留される。
（ＡのＩＥＯが“Ｌ”になり、優先順位の高いデバイスがサービス中であることを示す。）
Aルーチンが終了すると、ＲＥＴＩ命令でＡのＩＥＯがリセットされＢルーチンが再開される。Ｂルーチンが終了すると、ＲＥＴＩ命令でＢのＩＥＯがリセットされ、より順位の低いデバイスの割り込み処理が可能となる。

RETN ; Return from non maskable interrupt

（オペランド　なし）

ノン・マスカブル割り込みに対するサービス終了時に使用し、ＲＥＴ命令と同じく無条件復帰を実行する。プログラム・カウンタＰＣの前の内容を外部メモリ・スタックからポップする。すなわち、スタック・ポインタＳＰの内容が指定するメモリの内容をＰＣの下位バイトにロードし、ＳＰの内容を１だけ増す。このＳＰの新しい内容が指定するメモリの内容をＰＣの上位バイトにロードし、ＳＰの内容を１だけ増す。

制御は元のプログラムに戻り、ＰＣの指定するメモリから次の命令をフェッチする。

また、ＩＦＦ２の状態をＩＦＦ１にコピーし、ＮＭＩ受理前の状態に戻す。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	0	1	1	0	1	ED
0	1	0	0	0	1	0	1	45

実　行：Mサイクル：4，Tステート：14(4＋4＋3＋3)，実行時間：3.50μs（4 MHz）

フラグ：変化せず。

例：ＳＰ、ＰＣの内容がそれぞれ2000H、1B32Hであるとき、ノン・マスカブル割り込み(ＮＭＩ)信号が受理されると、ＣＰＵは次の命令を実行せず、0066H番地からリスタートする。このときのレジスタ・メモリの内容は次のようになる。

ＰＣ：００６６H　　ＳＰ：１ＦＦＥH

１ＦＦＦH番地：１BH　　１ＦＦＥH番地：３２H

0066H番地から始まった割り込み処理ルーチンは最後にＲＥＴＮ命令を実行する必要があるが、これにより、外部メモリ・スタックから前のＰＣの内容がポップされＰＣにロードされる。このとき、ＳＰ、ＰＣの内容は再びそれぞれ、2000H、1B32Hとなり、次の命令は1B32H番地からフェッチされる。

ＲＳＴ　ｐ　　；$(SP-1) \leftarrow PC_H$, $(SP-2) \leftarrow PC_L$, $PC_H \leftarrow 0$, $PC_L \leftarrow P$　（オペランド　ｐ）

そのときのプログラム・カウンタＰＣの内容を外部メモリ・スタックにプッシュし、オペランドｐが与えるページ・ゼロ・メモリの位置をＰＣにロードする。このＰＣの新しい内容が指定するアドレスから次の命令をフェッチする。

プッシュは次の手順で行われる。

スタック・ポインタＳＰの内容を１だけ減じ、このＳＰが指定するメモリにＰＣの上位バイトをロードする。次にＳＰの内容をさらに１だけ減じ、このＳＰが指定するメモリにＰＣの下位バイトをロードする。

ＲＳＴ（リスタート）命令により、８個のアドレス中の１個にジャンプすることができる。

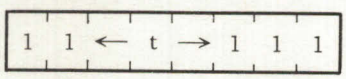
すべてのアドレスはページ・ゼロ・メモリに入っているので、ＰＣの上位バイトには00Ｈがロードされる。

Ｐの値はＰＣの下位バイトにロードされる。

オブジェクト・コードとＰの表は次のとおり。

Ｐ		ｔ
００Ｈ	＝	０００
０８Ｈ	＝	００１
１０Ｈ	＝	０１０
１８Ｈ	＝	０１１
２０Ｈ	＝	１００
２８Ｈ	＝	１０１
３０Ｈ	＝	１１０
３８Ｈ	＝	１１１

オブジェクト・コード



実　行　：　Ｍサイクル：３，Ｔステート：11（５＋３＋３），実行時間：2.75μs（４MHz）

フラグ　：　変化せず。

例　：　プログラム・カウンタＰＣの内容が16Ａ2Ｈのとき、

ＲＳＴ　18Ｈ（オブジェクト・コード 1101 1111）

を実行すると、ＰＣの内容は0018Ｈとなる。

―― 入 出 力　グ ル ー プ ――

IN A, (n) ; A←(n)

（オペランド A, (n)）

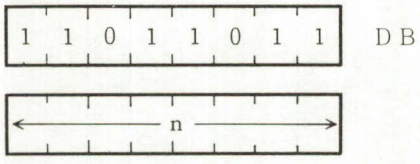
オペランドnをアドレス・バスの下位側(A_0からA_7)に割り付け、(n)で指定可能な256ポートのうちの1つの入出力デバイスを選択する。

本命令実行前のアキュムレータの内容がアドレス・バスの上位側(A_8からA_{15})に出力する。

選ばれたポートから、1バイトがデータ・バス上に置かれ、アキュムレータに取り込まれる。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行 ： Mサイクル：3，Tステート：11(4＋3＋4)，実行時間：2.75μs（4 MHz）

フラグ ： 変化せず。

例　 ： アキュムレータの内容が32Hで、1バイトのデータ7CHが入出力ポート・アドレス01Hに対応する周辺デバイスから得られるならば、

```
IN A, (01H)
```

を実行すると、アキュムレータの内容は7CHとなる。

IN r, (C) ; r←(C)

（オペランド r, (C)）

レジスタCの内容をアドレス・バスの下位側(A_0からA_7)に割り付け、(C)で指定可能な256ポートのうちの1つの入出力デバイスを選択する。

本命令実行前のレジスタBの内容がアドレス・バスの上位側(A_8からA_{15})に出力する。

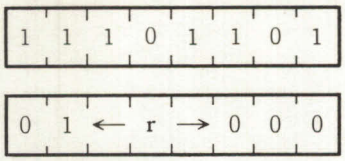
選ばれたポートから、1バイトがデータ・バス上に置かれ、レジスタrに取り込まれる。

フラグは入力データをチェックする。

rとオブジェクト・コードは次のとおり。

レジスタ	r
B ＝	0 0 0
C ＝	0 0 1
D ＝	0 1 0
E ＝	0 1 1
H ＝	1 0 0
L ＝	1 0 1
A ＝	1 1 1

オブジェクト・コード



実　行　：　Mサイクル：3，Tステート：12(4＋4＋4)，実行時間：3.00μs（4 MHz）

フラグ　：　S　：入力データが負ならばセット、他の場合はリセット。

Z　：入力データが零ならばセット、他の場合はリセット。

H　：リセット。

P／V：パリティが偶数ならばセット、他の場合はリセット。

N　：リセット。

C　：変化せず。

例　：　レジスタB、Cの内容がそれぞれ18H、06Hで、7CHが、入出力ポート・アドレス06H に対応する周辺デバイスから得られるならば、

```
IN D, (C)
```

を実行すると、レジスタDの内容は 7CHとなる。

INI　；(HL)←(C), B←B−1, HL←HL+1

（オペランド　なし）

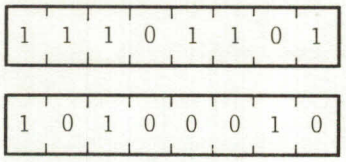
レジスタCの内容を、アドレス・バスの下位側(A_0からA_7)に割り付け、(C)で指定可能な256ポートのうちの1つの入出力デバイスを選択する。

レジスタBをバイト・カウンタとして用い、その内容はアドレス・バスの上位側（A_8からA_{15}）に出力する。選ばれたポートから、1バイトがデータ・バス上に置かれ、CPUに取り込まれる。その後に、レジスタ・ペアHLの内容をアドレス・バス上に送り、それが指定するメモリに取り込んだデータを書き込む。

最後に、バイト・カウンタB、レジスタ・ペアHLの内容をそれぞれ、1だけ減じ、1だけ増す。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行：Mサイクル：4，Tステート：16（4＋5＋3＋4），実行時間：4.00μs（4MHz）

フラグ：　S　：不定。

Z　：B－1＝0ならばセット、他の場合はリセット。

H　：不定。

P／V：不定。

N　：セット。

C　：不定。

例　：レジスタの内容が次のとき、

レジスタB：15H　　レジスタC：08H

レジスタ・ペアHL：2000H

入出力ポート・アドレス08Hに対応する周辺デバイスから7CHが得られるならば、

INI

を実行すると、メモリおよびレジスタの内容は次のようになる。

2000H番地：7CH

B：14H　　HL：2001H

INIR　；(HL)←(C)，B←B−1，HL←HL+1

（オペランド　なし）

レジスタCの内容をアドレス・バスの下位側（A_0からA_7）に割り付け、（C）で指定可能な256ポートのうちの1つの入出力デバイスを選択する。

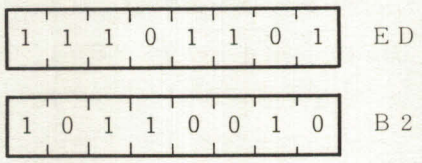
レジスタBをバイト・カウンタとして用い、その内容はアドレス・バスの上位側（A_8からA_{15}）に出力する。

選ばれたポートから、1バイトがデータ・バス上に置かれ、CPUに取り込まれる。

その後に、レジスタ・ペアHLの内容をアドレス・バス上に送り、それが指定するメモリに、取り込まれたデータを書き込む。

ＨＬ、Ｂをそれぞれ１だけ増し、１だけ減じる。このときもしＢが零になれば、命令は終了する。
Ｂが零でなければ、プログラム・カウンタＰＣを２だけ減じ、この命令を繰り返す。
もし命令実行前に、Ｂを零にセットしておけば、256バイトのデータが入力されることに注意する。
割り込みは１バイト転送ごとに受け付けられ、ＣＰＵは割り込みシーケンスに入る。
各データの転送ごとに、２つのリフレッシュ・サイクルを実行する。
オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行　：　Ｂキ０のとき、
Ｍサイクル：５，Ｔステート：21(４＋５＋３＋４＋５)，実行時間：5.25μs(4MHz)
Ｂ＝０のとき、
Ｍサイクル：４，Ｔステート：16(４＋５＋３＋４)，実行時間：4.00μs（４MHz）

フラグ：　Ｓ　：不定。
Ｚ　：セット。
Ｈ　：不定。
Ｐ／Ｖ：不定。
Ｎ　：セット。
Ｃ　：不定。

例　：　レジスタの内容が次のとき、
Ｂ：０３Ｈ　　Ｃ：０６Ｈ
ＨＬ：２０００Ｈ
入出力ポート・アドレス06Ｈ に対応する周辺デバイスから、48Ｈ、Ｂ6Ｈ、02Ｈが得られるならば、

INIR

を実行すると、レジスタ、メモリの内容は次のようになる。
Ｂ：００Ｈ　　ＨＬ：２００３Ｈ
２０００Ｈ 番地：４８Ｈ
２００１Ｈ 番地：Ｂ６Ｈ
２００２Ｈ 番地：０２Ｈ

IND　；(HL)←(C), B←B−1, HL←HL−1　（オペランド　なし）

レジスタCの内容を、アドレス・バスの下位側(A_0からA_7)に割り付け、(C)で指定可能な256ポートのうちの1つの入出力デバイスを選択する。

レジスタBをバイト・カウンタとして用い、その内容はアドレス・バスの上位側(A_8からA_{15})に出力する。選ばれたポートから、1バイトがデータ・バス上に置かれ、CPUに取り込まれる。

その後に、レジスタ・ペアHLの内容をアドレス・バス上に送り、それが指定するメモリに取り込まれたデータを書き込む。

最後に、バイト・カウンタB，レジスタ・ペアHLの内容をともに1だけ減じる。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	0	1	1	0	1	ED
1	0	1	0	1	0	1	0	AA

実　行：Mサイクル：4，Tステート：16(4＋5＋3＋4)，実行時間：4.00μs（4 MHz）

フラグ：　S　：不定。

Z　：B−1＝0ならばセット、他の場合はリセット。

H　：不定。

P／V：不定。

N　：セット。

C　：不定。

例　：レジスタの内容が次のとき、

B：12H　　C：06H

HL：2000H

入出力ポート・アドレス06H に対応する周辺デバイスから7CHが得られるならば、

IND

を実行すると、メモリ、レジスタの内容は次のようになる。

2000H 番地：7CH

HL：1FFFH　　B：11H

INDR　；(HL)←(C), B←B−1, HL←HL−1　　（オペランド　なし）

レジスタCの内容を、アドレス・バスの下位側(A_0からA_7)に割り付け、(C)で指定可能な256ポートのうちの1つの入出力デバイスを選択する。

レジスタBをバイト・カウンタとして用い、その内容はアドレス・バスの上位側(A_8からA_{15})に出力する。選ばれたポートから、1バイトがデータ・バス上に置かれ、CPUに取り込まれる。その後に、レジスタ・ペアHLの内容をアドレス・バス上に送り、それが指定するメモリに取り込んだデータを書き込む。

HL、Bをともに1だけ減じる。

このとき、もしBが零になれば命令は終了する。Bが零でなければプログラム・カウンタPCを2だけ減じこの命令を繰り返す。

もし命令実行前にBを零にセットしておけば、256バイトのデータが入力される。

割り込みは1バイト転送ごとに受け付けられ、CPUは割り込みシーケンスに入る。

各データの転送ごとに、2つのリフレッシュ・サイクルを実行する。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	0	1	1	0	1	ED
1	0	1	1	1	0	1	0	BA

実　行：Bキ0のとき、
Mサイクル：5，Tステート：21(4＋5＋3＋4＋5)，実行時間：5.25μs(4MHz)
B＝0のとき、
Mサイクル：4，Tステート：16(4＋5＋3＋4)，実行時間：4.00μs(4MHz)

フラグ：　S　：不定。
Z　：セット。
H　：不定。
P／V：不定。
N　：セット。
C　：不定。

例　：レジスタの内容が次のとき、

B ： 03H　　C ： 08H

HL ： 2000H

入出力ポート・アドレス08Hに対応する周辺デバイスから55H、A7H、02Hが得られるならば、

INDR

を実行すると、レジスタ、メモリの内容は次のようになる。

B ： 00H　　HL ： 1FFDH

1FFEH 番地：02H

1FFFH 番地：A7H

2000H 番地：55H

OUT (n), A ; (n)←A

（オペランド(n), A）

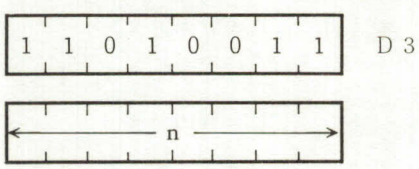
オペランドnをアドレス・バスの下位側(A_0からA_7)に割り付け、(n)で指定可能な256ポートのうちの1つの入出力デバイスを選択する。

本命令実行前のアキュムレータAの内容がアドレス・バスの上位側(A_8からA_{15})に出力する。

アキュムレータの内容は、データ・バス上に置かれ、選択した周辺デバイスに書き込まれる。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行 ： Mサイクル：3，Tステート：11(4＋3＋4)，実行時間：2.75μs（4 MHz）

フラグ ： 変化せず。

例　 ： アキュムレータの内容が42Hであるとき、

OUT 01H, A

を実行すると、入出力ポート・アドレス01Hに対応する周辺デバイスに42Hが書き込まれる。

OUT (C), r ; (c)←r

（オペランド（C），r）

レジスタCの内容を、アドレス・バスの下位側（A_0からA_7）に割り付け、（C）で指定可能な256ポートのうちの1つの入出力デバイスを選択する。

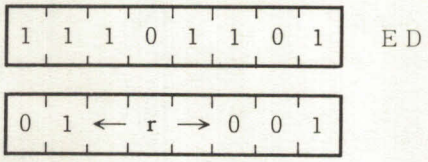
本命令実行前のレジスタBの内容がアドレス・バスの上位側（A_8からA_{15}）に出力する。

レジスタrの内容がデータ・バス上に置かれ、選ばれた周辺デバイスに取り込まれる。

rとオブジェクト・コードは次のとおり。

レジスタ		r
B	=	0 0 0
C	=	0 0 1
D	=	0 1 0
E	=	0 1 1
H	=	1 0 0
L	=	1 0 1
A	=	1 1 1

オブジェクト・コード



実　行：Mサイクル：3，Tステート：12（4＋4＋4），実行時間：3.00μs（4 MHz）

フラグ：変化せず。

例　：レジスタC、Dの内容がそれぞれ06H、3BHであるとき、

```
OUT (C), D
```

を実行すると、入出力ポート・アドレス06Hに対応する周辺デバイスに3BHが書き込まれる。

OUTI ; (C)←(HL), B←B−1, HL←HL+1

（オペランド　なし）

レジスタ・ペアHLの内容を、アドレス・バスに割り付け、メモリを指定する。このメモリの内容は一時的にCPUに取り込まれる。

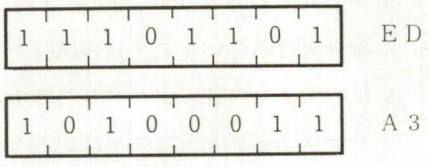
バイト・カウンタBを1だけ減じる。レジスタCの内容をアドレス・バスの下位側（A_0からA_7）に割り付け、（C）で指定可能な256ポートのうちの1つの入出力デバイスを選択する。

レジスタBはバイト・カウンタとして用い、その内容はアドレス・バスの上位側（A_8からA_{15}）に出力する。一時的にCPUに取り込まれたデータは、データ・バス上に置かれ、選ばれた周辺

デバイスに書き込まれる。最後に、ＨＬを１だけ増す。

オブジェクト・コードは次のとおり。

オブジェクト・コード



実　行　：　Mサイクル：4，Tステート：16(４＋５＋３＋４)，実行時間：4.00μs（４MHz）

フラグ　：　Ｓ　：不定。

Ｚ　：Ｂ－１＝０ならばセット、他の場合はリセット。

Ｈ　：不定。

Ｐ／Ｖ：不定。

Ｎ　：セット。

Ｃ　：不定。

例　　：レジスタ、メモリの内容が次のとき、

Ｂ：２０H　　Ｃ：０６H

ＨＬ：２０００H　　２０００H 番地：４８H

ＯＵＴＩ

を実行すると、レジスタＢ、ＨＬの内容はそれぞれ1FH、2001Hとなり、入出力ポート・アドレス06Hに対応する周辺デバイスに48Hが書き込まれる。

ＯＴＩＲ　；(C)←(HL)，B←B－1，HL←HL＋1　　（オペランド　なし）

レジスタ・ペアＨＬの内容を、アドレス・バスに割り付け、メモリを指定する。このメモリの内容は一時的にＣＰＵに取り込まれる。

バイト・カウンタＢを１だけ減じる。レジスタＣの内容をアドレス・バスの下位側(A_0からA_7)に割り付け、(Ｃ)で指定可能な256ポートのうちの１つの入出力デバイスを選択する。

レジスタＢはバイト・カウンタとして用い、その内容はアドレス・バスの上位側(A_8からA_{15})

に出力する。次に、先にＣＰＵに一時的に取り込まれたデータをデータ・バス上に置き、選ばれた周辺デバイスに書き込む。その後に、ＨＬの内容を１だけ増す。

もし、Ｂの内容が零でなければ、プログラム・カウンタＰＣの内容を２だけ減じ、この命令を繰り返す。Ｂの内容が零であれば、この命令は終了する。

もし命令実行前にＢを零にセットしておけば、256バイトのデータが出力される。

割り込みは１バイト転送ごとに受け付けられ、ＣＰＵは割り込みシーケンスに入る。

各データの転送ごとに、２つのリフレッシュ・サイクルを実行する。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	0	1	1	0	1	ED
1	0	1	1	0	0	1	1	B3

実　行：Ｂキ０のとき、

Ｍサイクル：５，Ｔステート：21(４＋５＋３＋４＋５)，実行時間：5.25μs(4MHz)

Ｂ＝０のとき、

Ｍサイクル：４，Ｔステート：16(４＋５＋３＋４)，実行時間：4.00μs（４MHz）

フラグ：　Ｓ　：不定。

Ｚ　：セット。

Ｈ　：不定。

Ｐ／Ｖ：不定。

Ｎ　：セット。

Ｃ　：不定。

例　：レジスタ、メモリの内容が次のとき、

Ｂ：０３Ｈ　　Ｃ：０６Ｈ

ＨＬ：２０００Ｈ

２０００Ｈ 番地：８８Ｈ

２００１Ｈ 番地：Ｂ６Ｈ

２００２Ｈ 番地：０４Ｈ

ＯＴＩＲ

を実行すると、レジスタＢ、ＨＬの内容はそれぞれ00Ｈ、2003Ｈとなり、入出力ポート・アドレス06Ｈに対応する周辺デバイスに、88Ｈ、Ｂ6Ｈ、04Ｈが書き込まれる。

OUTD　；(C)←(HL), B←B−1, HL←HL−1　（オペランド　なし）

レジスタ・ペアＨＬの内容をアドレス・バス上に割り付け、メモリを指定する。このメモリの内容は、一時的にＣＰＵに取り込まれる。

バイト・カウンタＢの内容を１だけ減じた後、レジスタＣの内容をアドレス・バスの下位側（A_0からA_7）に割り付け、（Ｃ）で指定可能な256ポートのうちの１つの入出力デバイスを選択する。

レジスタＢはバイト・カウンタとして用い、その内容はアドレス・バスの上位側（A_8からA_{15}）に出力する。次に、先にＣＰＵに一時的に取り込まれたデータをデータ・バス上に置き、選ばれた周辺デバイスに書き込む。最後に、ＨＬの内容を１だけ減じる。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	0	1	1	0	1	ED
1	0	1	0	1	0	1	1	AB

実　行：Mサイクル：４，Ｔステート：16(４＋５＋３＋４)，実行時間：4.00μs（４ＭＨｚ）

フラグ：　Ｓ　：不定。
　　Ｚ　：Ｂ－１＝０ならばセット、他の場合はリセット。
　　Ｈ　：不定。
　　Ｐ／Ｖ：不定。
　　Ｎ　：セット。
　　Ｃ　：不定。

例　：レジスタ、メモリの内容が次のとき、

Ｂ：２０Ｈ　　Ｃ：０６Ｈ

ＨＬ：２０００Ｈ　　２０００Ｈ 番地：４３Ｈ

OUTD

を実行すると、レジスタＢ、ＨＬの内容はそれぞれ1FH、1FFFHとなり、入出力ポート・アドレス 06Hに対応する周辺デバイスに43Hが書き込まれる。

OTDR　；(C)←(HL), B←B−1, HL←HL−1　（オペランド　なし）

レジスタ・ペアＨＬの内容をアドレス・バス上に割り付け、メモリを指定する。このメモリの内容は、一時的にＣＰＵに取り込まれる。

バイト・カウンタＢの内容を１だけ減じた後、レジスタＣの内容をアドレス・バスの下位側（A_0からA_7）に割り付け、（Ｃ）で指定可能な256ポートのうちの１つの入出力デバイスを選択する。

レジスタＢはバイト・カウンタとして用い、その内容はアドレス・バスの上位側（A_8からA_{15}）に出力する。次に、先に一時的にＣＰＵに取り込まれたデータをデータ・バス上に置き、選ばれた周辺デバイスに書き込む。ＨＬの内容を１だけ減じ、もしＢの内容が零でなければ、プログラム・カウンタＰＣの内容を２だけ減じ、この命令を繰り返す。Ｂの内容が零であれば、命令は終了する。

命令実行前にＢを零にセットしておけば、256バイトのデータが出力されることに注意する。

割り込みは１バイト転送ごとに受け付けられ、ＣＰＵは割り込みシーケンスに入る。

各データの転送ごとに、２つのリフレッシュ・サイクルを実行する。

オブジェクト・コードは次のとおり。

オブジェクト・コード

1	1	1	0	1	1	0	1	ED
1	0	1	1	1	0	1	1	BB

実　行：Ｂ≠０のとき、

Ｍサイクル：５，Ｔステート：21（４＋５＋３＋４＋５），実行時間：5.25μs（4MHz）

Ｂ＝０のとき、

Ｍサイクル：４，Ｔステート：16（４＋５＋３＋４），実行時間：4.00μs（４MHz）

フラグ：　Ｓ　：不定。

Ｚ　：セット。

Ｈ　：不定。

Ｐ／Ｖ：不定。

Ｎ　：セット。

Ｃ　：不定。

例　：レジスタ、メモリの内容が次のとき、

B：03H　　C：08H

HL：2000H

1FFEH 番地：61H

1FFFH 番地：B8H

2000H 番地：04H

OTDR

を実行すると、レジスタB、HLの内容はそれぞれ00H 、 1FFDH となり、入出力ポート・アドレス08Hに対応する周辺デバイスに、04H、B8H、61Hが書き込まれる。

第5章　Z－80命令セット索引（ＡＢＣ順）

付　　録

1. Z－80 命令セット（ＡＢＣ順）

```
TEST    ASMBL'D BY Z80 ASSEMBLER REV-A.4    02.09.'79

001     TEST
        1 0000 8E                ADC     A,(HL)
        2 0001 DD8E05            ADC     A,(IX+DIS)
        3 0004 FD8E05            ADC     A,(IY+DIS)
        4 0007 8F                ADC     A,A
        5 0008 88                ADC     A,B
        6 0009 89                ADC     A,C
        7 000A 8A                ADC     A,D
        8 000B 8B                ADC     A,E
        9 000C 8C                ADC     A,H
       10 000D 8D                ADC     A,L
       11 000E CE20              ADC     A,N
       12 0010 ED4A              ADC     HL,BC
       13 0012 ED5A              ADC     HL,DE
       14 0014 ED6A              ADC     HL,HL
       15 0016 ED7A              ADC     HL,SP
       16 0018 86                ADD     A,(HL)
       17 0019 DD8605            ADD     A,(IX+DIS)
       18 001C FD8605            ADD     A,(IY+DIS)
       19 001F 87                ADD     A,A
       20 0020 80                ADD     A,B
       21 0021 81                ADD     A,C
       22 0022 82                ADD     A,D
       23 0023 83                ADD     A,E
       24 0024 84                ADD     A,H
       25 0025 85                ADD     A,L
       26 0026 C620              ADD     A,N
       27 0028 09                ADD     HL,BC
       28 0029 19                ADD     HL,DE
       29 002A 29                ADD     HL,HL
       30 002B 39                ADD     HL,SP
       31 002C DD09              ADD     IX,BC
       32 002E DD19              ADD     IX,DE
       33 0030 DD29              ADD     IX,IX
       34 0032 DD39              ADD     IX,SP
       35 0034 FD09              ADD     IY,BC
       36 0036 FD19              ADD     IY,DE
       37 0038 FD29              ADD     IY,IY
       38 003A FD39              ADD     IY,SP
       39 003C A6                AND     (HL)
       40 003D DDA605            AND     (IX+DIS)
       41 0040 FDA605            AND     (IY+DIS)
       42 0043 A7                AND     A
       43 0044 A0                AND     B
       44 0045 A1                AND     C
       45 0046 A2                AND     D
       46 0047 A3                AND     E
       47 0048 A4                AND     H
       48 0049 A5                AND     L
       49 004A E620              AND     N
       50 004C CB46              BIT     0,(HL)
       51 004E DDCB0546          BIT     0,(IX+DIS)
       52 0052 FDCB0546          BIT     0,(IY+DIS)
       53 0056 CB47              BIT     0,A
       54 0058 CB40              BIT     0,B
       55 005A CB41              BIT     0,C
       56 005C CB42              BIT     0,D
       57 005E CB43              BIT     0,E
       58 0060 CB44              BIT     0,H
       59 0062 CB45              BIT     0,L
       60 0064 CB4E              BIT     1,(HL)
```

```
002     TEST
         61 0066 DDCB054E          BIT     1,(IX+DIS)
         62 006A FDCB054E          BIT     1,(IY+DIS)
         63 006E CB4F              BIT     1,A
         64 0070 CB48              BIT     1,B
         65 0072 CB49              BIT     1,C
         66 0074 CB4A              BIT     1,D
         67 0076 CB4B              BIT     1,E
         68 0078 CB4C              BIT     1,H
         69 007A CB4D              BIT     1,L
         70 007C CB56              BIT     2,(HL)
         71 007E DDCB0556          BIT     2,(IX+DIS)
         72 0082 FDCB0556          BIT     2,(IY+DIS)
         73 0086 CB57              BIT     2,A
         74 0088 CB50              BIT     2,B
         75 008A CB51              BIT     2,C
         76 008C CB52              BIT     2,D
         77 008E CB53              BIT     2,E
         78 0090 CB54              BIT     2,H
         79 0092 CB55              BIT     2,L
         80 0094 CB5E              BIT     3,(HL)
         81 0096 DDCB055E          BIT     3,(IX+DIS)
         82 009A FDCB055E          BIT     3,(IY+DIS)
         83 009E CB5F              BIT     3,A
         84 00A0 CB58              BIT     3,B
         85 00A2 CB59              BIT     3,C
         86 00A4 CB5A              BIT     3,D
         87 00A6 CB5B              BIT     3,E
         88 00A8 CB5C              BIT     3,H
         89 00AA CB5D              BIT     3,L
         90 00AC CB66              BIT     4,(HL)
         91 00AE DDCB0566          BIT     4,(IX+DIS)
         92 00B2 FDCB0566          BIT     4,(IY+DIS)
         93 00B6 CB67              BIT     4,A
         94 00B8 CB60              BIT     4,B
         95 00BA CB61              BIT     4,C
         96 00BC CB62              BIT     4,D
         97 00BE CB63              BIT     4,E
         98 00C0 CB64              BIT     4,H
         99 00C2 CB65              BIT     4,L
        100 00C4 CB6E              BIT     5,(HL)
        101 00C6 DDCB056E          BIT     5,(IX+DIS)
        102 00CA FDCB056E          BIT     5,(IY+DIS)
        103 00CE CB6F              BIT     5,A
        104 00D0 CB68              BIT     5,B
        105 00D2 CB69              BIT     5,C
        106 00D4 CB6A              BIT     5,D
        107 00D6 CB6B              BIT     5,E
        108 00D8 CB6C              BIT     5,H
        109 00DA CB6D              BIT     5,L
        110 00DC CB76              BIT     6,(HL)
        111 00DE DDCB0576          BIT     6,(IX+DIS)
        112 00E2 FDCB0576          BIT     6,(IY+DIS)
        113 00E6 CB77              BIT     6,A
        114 00E8 CB70              BIT     6,B
        115 00EA CB71              BIT     6,C
        116 00EC CB72              BIT     6,D
        117 00EE CB73              BIT     6,E
        118 00F0 CB74              BIT     6,H
        119 00F2 CB75              BIT     6,L
        120 00F4 CB7E              BIT     7,(HL)
```

```
003     TEST
      121 00F6 DDCB057E          BIT     7,(IX+DIS)
      122 00FA FDCB057E          BIT     7,(IY+DIS)
      123 00FE CB7F              BIT     7,A
      124 0100 CB78              BIT     7,B
      125 0102 CB79              BIT     7,C
      126 0104 CB7A              BIT     7,D
      127 0106 CB7B              BIT     7,E
      128 0108 CB7C              BIT     7,H
      129 010A CB7D              BIT     7,L
      130 010C DCB001            CALL    C,NN
      131 010F FCB001            CALL    M,NN
      132 0112 D4B001            CALL    NC,NN
      133 0115 CDB001            CALL    NN
      134 0118 C4B001            CALL    NZ,NN
      135 011B F4B001            CALL    P,NN
      136 011E ECB001            CALL    PE,NN
      137 0121 E4B001            CALL    PO,NN
      138 0124 CCB001            CALL    Z,NN
      139 0127 3F                CCF
      140 0128 BE                CP      (HL)
      141 0129 DDBE05            CP      (IX+DIS)
      142 012C FDBE05            CP      (IY+DIS)
      143 012F BF                CP      A
      144 0130 B8                CP      B
      145 0131 B9                CP      C
      146 0132 BA                CP      D
      147 0133 BB                CP      E
      148 0134 BC                CP      H
      149 0135 BD                CP      L
      150 0136 FE20              CP      N
      151 0138 EDA9              CPD
      152 013A EDB9              CPDR
      153 013C EDA1              CPI
      154 013E EDB1              CPIR
      155 0140 2F                CPL
      156 0141 27                DAA
      157 0142 35                DEC     (HL)
      158 0143 DD3505            DEC     (IX+DIS)
      159 0146 FD3505            DEC     (IY+DIS)
      160 0149 3D                DEC     A
      161 014A 05                DEC     B
      162 014B 0B                DEC     BC
      163 014C 0D                DEC     C
      164 014D 15                DEC     D
      165 014E 1B                DEC     DE
      166 014F 1D                DEC     E
      167 0150 25                DEC     H
      168 0151 2B                DEC     HL
      169 0152 DD2B              DEC     IX
      170 0154 FD2B              DEC     IY
      171 0156 2D                DEC     L
      172 0157 3B                DEC     SP
      173 0158 F3                DI
      174 0159 1055              DJNZ    NN
      175 015B FB                EI
      176 015C E3                EX      (SP),HL
      177 015D DDE3              EX      (SP),IX
      178 015F FDE3              EX      (SP),IY
      179 0161 08                EX      AF,AF'
      180 0162 EB                EX      DE,HL
```

```
004     TEST
      181 0163 D9              EXX
      182 0164 76              HALT
      183 0165 ED46            IM      0
      184 0167 ED56            IM      1
      185 0169 ED5E            IM      2
      186 016B ED78            IN      A,(C)
      187 016D DB20            IN      A,(N)
      188 016F ED40            IN      B,(C)
      189 0171 ED48            IN      C,(C)
      190 0173 ED50            IN      D,(C)
      191 0175 ED58            IN      E,(C)
      192 0177 ED60            IN      H,(C)
      193 0179 ED68            IN      L,(C)
      194 017B 34              INC     (HL)
      195 017C FD3405          INC     (IY+DIS)
      196 017F DD3405          INC     (IX+DIS)
      197 0182 3C              INC     A
      198 0183 04              INC     B
      199 0184 03              INC     BC
      200 0185 0C              INC     C
      201 0186 14              INC     D
      202 0187 13              INC     DE
      203 0188 1C              INC     E
      204 0189 24              INC     H
      205 018A 23              INC     HL
      206 018B DD23            INC     IX
      207 018D FD23            INC     IY
      208 018F 2C              INC     L
      209 0190 33              INC     SP
      210 0191 EDAA            IND
      211 0193 EDBA            INDR
      212 0195 EDA2            INI
      213 0197 EDB2            INIR
      214 0199 E9              JP      (HL)
      215 019A DDE9            JP      (IX)
      216 019C FDE9            JP      (IY)
      217 019E DAB001          JP      C,NN
      218 01A1 FAB001          JP      M,NN
      219 01A4 D2B001          JP      NC,NN
      220 01A7 C3B001          JP      NN
      221 01AA C2B001          JP      NZ,NN
      222 01AD F2B001          JP      P,NN
      223 01B0 EAB001          JP      PE,NN
      224 01B3 E2B001          JP      PO,NN
      225 01B6 CAB001          JP      Z,NN
      226 01B9 38F5            JR      C,NN
      227 01BB 18F3            JR      NN
      228 01BD 30F1            JR      NC,NN
      229 01BF 20EF            JR      NZ,NN
      230 01C1 28ED            JR      Z,NN
      231 01C3 02              LD      (BC),A
      232 01C4 12              LD      (DE),A
      233 01C5 77              LD      (HL),A
      234 01C6 70              LD      (HL),B
      235 01C7 71              LD      (HL),C
      236 01C8 72              LD      (HL),D
      237 01C9 73              LD      (HL),E
      238 01CA 74              LD      (HL),H
      239 01CB 75              LD      (HL),L
      240 01CC 3620            LD      (HL),N
```

```
005     TEST
      241 01CE DD7705          LD      (IX+DIS),A
      242 01D1 DD7005          LD      (IX+DIS),B
      243 01D4 DD7105          LD      (IX+DIS),C
      244 01D7 DD7205          LD      (IX+DIS),D
      245 01DA DD7305          LD      (IX+DIS),E
      246 01DD DD7405          LD      (IX+DIS),H
      247 01E0 DD7505          LD      (IX+DIS),L
      248 01E3 DD360520        LD      (IX+DIS),N
      249 01E7 FD7705          LD      (IY+DIS),A
      250 01EA FD7005          LD      (IY+DIS),B
      251 01ED FD7105          LD      (IY+DIS),C
      252 01F0 FD7205          LD      (IY+DIS),D
      253 01F3 FD7305          LD      (IY+DIS),E
      254 01F6 FD7405          LD      (IY+DIS),H
      255 01F9 FD7505          LD      (IY+DIS),L
      256 01FC FD360520        LD      (IY+DIS),N
      257 0200 32B001          LD      (NN),A
      258 0203 ED43B001        LD      (NN),BC
      259 0207 ED53B001        LD      (NN),DE
      260 020B 22B001          LD      (NN),HL
      261 020E DD22B001        LD      (NN),IX
      262 0212 FD22B001        LD      (NN),IY
      263 0216 ED73B001        LD      (NN),SP
      264 021A 0A              LD      A,(BC)
      265 021B 1A              LD      A,(DE)
      266 021C 7E              LD      A,(HL)
      267 021D DD7E05          LD      A,(IX+DIS)
      268 0220 FD7E05          LD      A,(IY+DIS)
      269 0223 3AB001          LD      A,(NN)
      270 0226 7F              LD      A,A
      271 0227 78              LD      A,B
      272 0228 79              LD      A,C
      273 0229 7A              LD      A,D
      274 022A 7B              LD      A,E
      275 022B 7C              LD      A,H
      276 022C ED57            LD      A,I
      277 022E 7D              LD      A,L
      278 022F 3E20            LD      A,N
      279 0231 ED5F            LD      A,R
      280 0233 46              LD      B,(HL)
      281 0234 DD4605          LD      B,(IX+DIS)
      282 0237 FD4605          LD      B,(IY+DIS)
      283 023A 47              LD      B,A
      284 023B 40              LD      B,B
      285 023C 41              LD      B,C
      286 023D 42              LD      B,D
      287 023E 43              LD      B,E
      288 023F 44              LD      B,H
      289 0240 45              LD      B,L
      290 0241 0620            LD      B,N
      291 0243 ED4BB001        LD      BC,(NN)
      292 0247 01B001          LD      BC,NN
      293 024A 4E              LD      C,(HL)
      294 024B DD4E05          LD      C,(IX+DIS)
      295 024E FD4E05          LD      C,(IY+DIS)
      296 0251 4F              LD      C,A
      297 0252 48              LD      C,B
      298 0253 49              LD      C,C
      299 0254 4A              LD      C,D
      300 0255 4B              LD      C,E
```

```
006     TEST
      301 0256 4C              LD      C,H
      302 0257 4D              LD      C,L
      303 0258 0E20            LD      C,N
      304 025A 56              LD      D,(HL)
      305 025B DD5605          LD      D,(IX+DIS)
      306 025E FD5605          LD      D,(IY+DIS)
      307 0261 57              LD      D,A
      308 0262 50              LD      D,B
      309 0263 51              LD      D,C
      310 0264 52              LD      D,D
      311 0265 53              LD      D,E
      312 0266 54              LD      D,H
      313 0267 55              LD      D,L
      314 0268 1620            LD      D,N
      315 026A ED5BB001        LD      DE,(NN)
      316 026E 11B001          LD      DE,NN
      317 0271 5E              LD      E,(HL)
      318 0272 DD5E05          LD      E,(IX+DIS)
      319 0275 FD5E05          LD      E,(IY+DIS)
      320 0278 5F              LD      E,A
      321 0279 58              LD      E,B
      322 027A 59              LD      E,C
      323 027B 5A              LD      E,D
      324 027C 5B              LD      E,E
      325 027D 5C              LD      E,H
      326 027E 5D              LD      E,L
      327 027F 1E20            LD      E,N
      328 0281 66              LD      H,(HL)
      329 0282 DD6605          LD      H,(IX+DIS)
      330 0285 FD6605          LD      H,(IY+DIS)
      331 0288 67              LD      H,A
      332 0289 60              LD      H,B
      333 028A 61              LD      H,C
      334 028B 62              LD      H,D
      335 028C 63              LD      H,E
      336 028D 64              LD      H,H
      337 028E 65              LD      H,L
      338 028F 2620            LD      H,N
      339 0291 2AB001          LD      HL,(NN)
      340 0294 21B001          LD      HL,NN
      341 0297 ED47            LD      I,A
      342 0299 DD2AB001        LD      IX,(NN)
      343 029D DD21B001        LD      IX,NN
      344 02A1 FD2AB001        LD      IY,(NN)
      345 02A5 FD21B001        LD      IY,NN
      346 02A9 6E              LD      L,(HL)
      347 02AA DD6E05          LD      L,(IX+DIS)
      348 02AD FD6E05          LD      L,(IY+DIS)
      349 02B0 6F              LD      L,A
      350 02B1 68              LD      L,B
      351 02B2 69              LD      L,C
      352 02B3 6A              LD      L,D
      353 02B4 6B              LD      L,E
      354 02B5 6C              LD      L,H
      355 02B6 6D              LD      L,L
      356 02B7 2E20            LD      L,N
      357 02B9 ED4F            LD      R,A
      358 02BB ED7BB001        LD      SP,(NN)
      359 02BF F9              LD      SP,HL
      360 02C0 DDF9            LD      SP,IX
```

```
007    TEST
     361 02C2 FDF9          LD      SP,IY
     362 02C4 31B001        LD      SP,NN
     363 02C7 EDA8          LDD
     364 02C9 EDB8          LDDR
     365 02CB EDA0          LDI
     366 02CD EDB0          LDIR
     367 02CF ED44          NEG
     368 02D1 00            NOP
     369 02D2 B6            OR      (HL)
     370 02D3 DDB605        OR      (IX+DIS)
     371 02D6 FDB605        OR      (IY+DIS)
     372 02D9 B7            OR      A
     373 02DA B0            OR      B
     374 02DB B1            OR      C
     375 02DC B2            OR      D
     376 02DD B3            OR      E
     377 02DE B4            OR      H
     378 02DF B5            OR      L
     379 02E0 F620          OR      N
     380 02E2 EDBB          OTDR
     381 02E4 EDB3          OTIR
     382 02E6 ED79          OUT     (C),A
     383 02E8 ED41          OUT     (C),B
     384 02EA ED49          OUT     (C),C
     385 02EC ED51          OUT     (C),D
     386 02EE ED59          OUT     (C),E
     387 02F0 ED61          OUT     (C),H
     388 02F2 ED69          OUT     (C),L
     389 02F4 D320          OUT     (N),A
     390 02F6 EDAB          OUTD
     391 02F8 EDA3          OUTI
     392 02FA F1            POP     AF
     393 02FB C1            POP     BC
     394 02FC D1            POP     DE
     395 02FD E1            POP     HL
     396 02FE DDE1          POP     IX
     397 0300 FDE1          POP     IY
     398 0302 F5            PUSH    AF
     399 0303 C5            PUSH    BC
     400 0304 D5            PUSH    DE
     401 0305 E5            PUSH    HL
     402 0306 DDE5          PUSH    IX
     403 0308 FDE5          PUSH    IY
     404 030A CB86          RES     0,(HL)
     405 030C DDCB0586      RES     0,(IX+DIS)
     406 0310 FDCB0586      RES     0,(IY+DIS)
     407 0314 CB87          RES     0,A
     408 0316 CB80          RES     0,B
     409 0318 CB81          RES     0,C
     410 031A CB82          RES     0,D
     411 031C CB83          RES     0,E
     412 031E CB84          RES     0,H
     413 0320 CB85          RES     0,L
     414 0322 CB8E          RES     1,(HL)
     415 0324 DDCB058E      RES     1,(IX+DIS)
     416 0328 FDCB058E      RES     1,(IY+DIS)
     417 032C CB8F          RES     1,A
     418 032E CB88          RES     1,B
     419 0330 CB89          RES     1,C
     420 0332 CB8A          RES     1,D
```

```
008     TEST
      421 0334 CB8B              RES     1,E
      422 0336 CB8C              RES     1,H
      423 0338 CB8D              RES     1,L
      424 033A CB96              RES     2,(HL)
      425 033C DDCB0596          RES     2,(IX+DIS)
      426 0340 FDCB0596          RES     2,(IY+DIS)
      427 0344 CB97              RES     2,A
      428 0346 CB90              RES     2,B
      429 0348 CB91              RES     2,C
      430 034A CB92              RES     2,D
      431 034C CB93              RES     2,E
      432 034E CB94              RES     2,H
      433 0350 CB95              RES     2,L
      434 0352 CB9E              RES     3,(HL)
      435 0354 DDCB059E          RES     3,(IX+DIS)
      436 0358 FDCB059E          RES     3,(IY+DIS)
      437 035C CB9F              RES     3,A
      438 035E CB98              RES     3,B
      439 0360 CB99              RES     3,C
      440 0362 CB9A              RES     3,D
      441 0364 CB9B              RES     3,E
      442 0366 CB9C              RES     3,H
      443 0368 CB9D              RES     3,L
      444 036A CBA6              RES     4,(HL)
      445 036C DDCB05A6          RES     4,(IX+DIS)
      446 0370 FDCB05A6          RES     4,(IY+DIS)
      447 0374 CBA7              RES     4,A
      448 0376 CBA0              RES     4,B
      449 0378 CBA1              RES     4,C
      450 037A CBA2              RES     4,D
      451 037C CBA3              RES     4,E
      452 037E CBA4              RES     4,H
      453 0380 CBA5              RES     4,L
      454 0382 CBAE              RES     5,(HL)
      455 0384 DDCB05AE          RES     5,(IX+DIS)
      456 0388 FDCB05AE          RES     5,(IY+DIS)
      457 038C CBAF              RES     5,A
      458 038E CBA8              RES     5,B
      459 0390 CBA9              RES     5,C
      460 0392 CBAA              RES     5,D
      461 0394 CBAB              RES     5,E
      462 0396 CBAC              RES     5,H
      463 0398 CBAD              RES     5,L
      464 039A CBB6              RES     6,(HL)
      465 039C DDCB05B6          RES     6,(IX+DIS)
      466 03A0 FDCB05B6          RES     6,(IY+DIS)
      467 03A4 CBB7              RES     6,A
      468 03A6 CBB0              RES     6,B
      469 03A8 CBB1              RES     6,C
      470 03AA CBB2              RES     6,D
      471 03AC CBB3              RES     6,E
      472 03AE CBB4              RES     6,H
      473 03B0 CBB5              RES     6,L
      474 03B2 CBBE              RES     7,(HL)
      475 03B4 DDCB05BE          RES     7,(IX+DIS)
      476 03B8 FDCB05BE          RES     7,(IY+DIS)
      477 03BC CBBF              RES     7,A
      478 03BE CBB8              RES     7,B
      479 03C0 CBB9              RES     7,C
      480 03C2 CBBA              RES     7,D
```

```
009    TEST
     481 03C4 CBBB              RES     7,E
     482 03C6 CBBC              RES     7,H
     483 03C8 CBBD              RES     7,L
     484 03CA C9                RET
     485 03CB D8                RET     C
     486 03CC F8                RET     M
     487 03CD D0                RET     NC
     488 03CE C0                RET     NZ
     489 03CF F0                RET     P
     490 03D0 E8                RET     PE
     491 03D1 E0                RET     PO
     492 03D2 C8                RET     Z
     493 03D3 ED4D              RETI
     494 03D5 ED45              RETN
     495 03D7 CB16              RL      (HL)
     496 03D9 DDCB0516          RL      (IX+DIS)
     497 03DD FDCB0516          RL      (IY+DIS)
     498 03E1 CB17              RL      A
     499 03E3 CB10              RL      B
     500 03E5 CB11              RL      C
     501 03E7 CB12              RL      D
     502 03E9 CB13              RL      E
     503 03EB CB14              RL      H
     504 03ED CB15              RL      L
     505 03EF 17                RLA
     506 03F0 CB06              RLC     (HL)
     507 03F2 DDCB0506          RLC     (IX+DIS)
     508 03F6 FDCB0506          RLC     (IY+DIS)
     509 03FA CB07              RLC     A
     510 03FC CB00              RLC     B
     511 03FE CB01              RLC     C
     512 0400 CB02              RLC     D
     513 0402 CB03              RLC     E
     514 0404 CB04              RLC     H
     515 0406 CB05              RLC     L
     516 0408 07                RLCA
     517 0409 ED6F              RLD
     518 040B CB1E              RR      (HL)
     519 040D DDCB051E          RR      (IX+DIS)
     520 0411 FDCB051E          RR      (IY+DIS)
     521 0415 CB1F              RR      A
     522 0417 CB18              RR      B
     523 0419 CB19              RR      C
     524 041B CB1A              RR      D
     525 041D CB1B              RR      E
     526 041F CB1C              RR      H
     527 0421 CB1D              RR      L
     528 0423 1F                RRA
     529 0424 CB0E              RRC     (HL)
     530 0426 DDCB050E          RRC     (IX+DIS)
     531 042A FDCB050E          RRC     (IY+DIS)
     532 042E CB0F              RRC     A
     533 0430 CB08              RRC     B
     534 0432 CB09              RRC     C
     535 0434 CB0A              RRC     D
     536 0436 CB0B              RRC     E
     537 0438 CB0C              RRC     H
     538 043A CB0D              RRC     L
     539 043C 0F                RRCA
     540 043D ED67              RRD
```

```
010     TEST
      541 043F C7              RST     0
      542 0440 CF              RST     08H
      543 0441 D7              RST     10H
      544 0442 DF              RST     18H
      545 0443 E7              RST     20H
      546 0444 EF              RST     28H
      547 0445 F7              RST     30H
      548 0446 FF              RST     38H
      549 0447 9E              SBC     A,(HL)
      550 0448 DD9E05          SBC     A,(IX+DIS)
      551 044B FD9E05          SBC     A,(IY+DIS)
      552 044E 9F              SBC     A,A
      553 044F 98              SBC     A,B
      554 0450 99              SBC     A,C
      555 0451 9A              SBC     A,D
      556 0452 9B              SBC     A,E
      557 0453 9C              SBC     A,H
      558 0454 9D              SBC     A,L
      559 0455 DE20            SBC     A,N
      560 0457 ED42            SBC     HL,BC
      561 0459 ED52            SBC     HL,DE
      562 045B ED62            SBC     HL,HL
      563 045D ED72            SBC     HL,SP
      564 045F 37              SCF
      565 0460 CBC6            SET     0,(HL)
      566 0462 DDCB05C6        SET     0,(IX+DIS)
      567 0466 FDCB05C6        SET     0,(IY+DIS)
      568 046A CBC7            SET     0,A
      569 046C CBC0            SET     0,B
      570 046E CBC1            SET     0,C
      571 0470 CBC2            SET     0,D
      572 0472 CBC3            SET     0,E
      573 0474 CBC4            SET     0,H
      574 0476 CBC5            SET     0,L
      575 0478 CBCE            SET     1,(HL)
      576 047A DDCB05CE        SET     1,(IX+DIS)
      577 047E FDCB05CE        SET     1,(IY+DIS)
      578 0482 CBCF            SET     1,A
      579 0484 CBC8            SET     1,B
      580 0486 CBC9            SET     1,C
      581 0488 CBCA            SET     1,D
      582 048A CBCB            SET     1,E
      583 048C CBCC            SET     1,H
      584 048E CBCD            SET     1,L
      585 0490 CBD6            SET     2,(HL)
      586 0492 DDCB05D6        SET     2,(IX+DIS)
      587 0496 FDCB05D6        SET     2,(IY+DIS)
      588 049A CBD7            SET     2,A
      589 049C CBD0            SET     2,B
      590 049E CBD1            SET     2,C
      591 04A0 CBD2            SET     2,D
      592 04A2 CBD3            SET     2,E
      593 04A4 CBD4            SET     2,H
      594 04A6 CBD5            SET     2,L
      595 04A8 CBDE            SET     3,(HL)
      596 04AA DDCB05DE        SET     3,(IX+DIS)
      597 04AE FDCB05DE        SET     3,(IY+DIS)
      598 04B2 CBDF            SET     3,A
      599 04B4 CBD8            SET     3,B
      600 04B6 CBD9            SET     3,C
```

```
011     TEST
      601 04B8 CBDA              SET     3,D
      602 04BA CBDB              SET     3,E
      603 04BC CBDC              SET     3,H
      604 04BE CBDD              SET     3,L
      605 04C0 CBE6              SET     4,(HL)
      606 04C2 DDCB05E6          SET     4,(IX+DIS)
      607 04C6 FDCB05E6          SET     4,(IY+DIS)
      608 04CA CBE7              SET     4,A
      609 04CC CBE0              SET     4,B
      610 04CE CBE1              SET     4,C
      611 04D0 CBE2              SET     4,D
      612 04D2 CBE3              SET     4,E
      613 04D4 CBE4              SET     4,H
      614 04D6 CBE5              SET     4,L
      615 04D8 CBEE              SET     5,(HL)
      616 04DA DDCB05EE          SET     5,(IX+DIS)
      617 04DE FDCB05EE          SET     5,(IY+DIS)
      618 04E2 CBEF              SET     5,A
      619 04E4 CBE8              SET     5,B
      620 04E6 CBE9              SET     5,C
      621 04E8 CBEA              SET     5,D
      622 04EA CBEB              SET     5,E
      623 04EC CBEC              SET     5,H
      624 04EE CBED              SET     5,L
      625 04F0 CBF6              SET     6,(HL)
      626 04F2 DDCB05F6          SET     6,(IX+DIS)
      627 04F6 FDCB05F6          SET     6,(IY+DIS)
      628 04FA CBF7              SET     6,A
      629 04FC CBF0              SET     6,B
      630 04FE CBF1              SET     6,C
      631 0500 CBF2              SET     6,D
      632 0502 CBF3              SET     6,E
      633 0504 CBF4              SET     6,H
      634 0506 CBF5              SET     6,L
      635 0508 CBFE              SET     7,(HL)
      636 050A DDCB05FE          SET     7,(IX+DIS)
      637 050E FDCB05FE          SET     7,(IY+DIS)
      638 0512 CBFF              SET     7,A
      639 0514 CBF8              SET     7,B
      640 0516 CBF9              SET     7,C
      641 0518 CBFA              SET     7,D
      642 051A CBFB              SET     7,E
      643 051C CBFC              SET     7,H
      644 051E CBFD              SET     7,L
      645 0520 CB26              SLA     (HL)
      646 0522 DDCB0526          SLA     (IX+DIS)
      647 0526 FDCB0526          SLA     (IY+DIS)
      648 052A CB27              SLA     A
      649 052C CB20              SLA     B
      650 052E CB21              SLA     C
      651 0530 CB22              SLA     D
      652 0532 CB23              SLA     E
      653 0534 CB24              SLA     H
      654 0536 CB25              SLA     L
      655 0538 CB2E              SRA     (HL)
      656 053A DDCB052E          SRA     (IX+DIS)
      657 053E FDCB052E          SRA     (IY+DIS)
      658 0542 CB2F              SRA     A
      659 0544 CB28              SRA     B
      660 0546 CB29              SRA     C
```

```
012     TEST
      661 0548 CB2A                  SRA     D
      662 054A CB2B                  SRA     E
      663 054C CB2C                  SRA     H
      664 054F CB2D                  SRA     L
      665 0550 CB3E                  SRL     (HL)
      666 0552 DDCB053E              SRL     (IX+DIS)
      667 0556 FDCB053E              SRL     (IY+DIS)
      668 055A CB3F                  SRL     A
      669 055C CB38                  SRL     B
      670 055E CB39                  SRL     C
      671 0560 CB3A                  SRL     D
      672 0562 CB3B                  SRL     E
      673 0564 CB3C                  SRL     H
      674 0566 CB3D                  SRL     L
      675 0568 96                    SUB     (HL)
      676 0569 DD9605                SUB     (IX+DIS)
      677 056C FD9605                SUB     (IY+DIS)
      678 056F 97                    SUB     A
      679 0570 90                    SUB     B
      680 0571 91                    SUB     C
      681 0572 92                    SUB     D
      682 0573 93                    SUB     E
      683 0574 94                    SUB     H
      684 0575 95                    SUB     L
      685 0576 D620                  SUB     N
      686 0578 AE                    XOR     (HL)
      687 0579 DDAE05                XOR     (IX+DIS)
      688 057C FDAE05                XOR     (IY+DIS)
      689 057F AF                    XOR     A
      690 0580 A8                    XOR     B
      691 0581 A9                    XOR     C
      692 0582 AA                    XOR     D
      693 0583 AB                    XOR     E
      694 0584 AC                    XOR     H
      695 0585 AD                    XOR     L
      696 0586 EE20                  XOR     N
      697      01B0          NN      EQU     1B0H
      698      0005          DIS     EQU     5
      699      0020          N       EQU     20H
      700 0588                       END
```

アセンブラ 例題

```
        Z80 MACRO ASSEMBLER  V1.2  PAGE    1          80/10/30

 1                      ;SAMPLE PROGRAM
 2                      ;
 3                      ;UNSIGNED 16 BITS BINARY MULTIPLY SUBROUTINE
 4                      ;
 5                      ;(IX+0H)  INDICATES THE ADDRESS OF MALTIPLIER
 6                      ;(IX+2H)  INDICATES THE ADDRESS OF MALTIPLICAND
 7                      ;(IY+0H)  INDICATES THE ADDRESS OF RESULT
 8                      ;DE,DE' : SHIFTING AREA FOR MALTIPLICAND
 9                      ;HL,HL' : PARTIAL RESULT AREA
10                      ;A,C    : SHIFTING AREA FOR MALTIPLIER
11                      ;B      : SHIFT COUNTER
12                      ;
13 0000 DD7E01                  LD      A,(IX+1H)       ;LOAD HIGH ORDER MALTIPLIER TO A
14 0003 DD4E00                  LD      C,(IX+0H)       ;LOAD LOW  ORDER MALTIPLIER TO C
15 0006 DD5603                  LD      D,(IX+3H)       ;LOAD HIGH ORDER MALTIPLICAND TO D
16 0009 DD5E02                  LD      E,(IX+2H)       ;LOAD LOW  ORDER MALTIPLICAND TO E
17 000C 0610                    LD      B,16            ;NUMBER FOR COUNTING
18 000E 210000                  LD      HL,0H           ;CLEAR PARTIAL RESULT LOW ORDER
19 0011 D9                      EXX
20 0012 210000                  LD      HL,0H           ;CLEAR PARTIAL RESULT HIGH ORDER
21 0015 110000                  LD      DE,0H           ;CLEAR SHIFTING AREA
22 0018 D9                      EXX
23                      ;
24 0019 CB3F            LOOP:   SRL     A               ;SHIFT RIGHT MULTIPLIER
25 001B CB19                    RR      C
26 001D 3005                    JR      NC,SHIFT        ;IF NO CARRY,JUMP TO SHIFT
27 001F 19                      ADD     HL,DE           ;ADD SHIFTED MALTIPLICAND
28 0020 D9                      EXX
29 0021 ED5A                    ADC     HL,DE
30 0023 D9                      EXX
31 0024 CB23            SHIFT:  SLA     E               ;SHIFT LEFT MULTIPLICAND
32 0026 CB12                    RL      D
33 0028 D9                      EXX
34 0029 CB13                    RL      E
35 002B CB12                    RL      D
36 002D D9                      EXX
37 002E 10E9                    DJNZ    LOOP            ;REPEAT UNTIL COUNTING IS ZERO
38                                                      ;STORE 32 BITS RESULT
39 0030 FD7500                  LD      (IY+0H),L       ;(IY+0H) : LOWEST   ORDER BYTE
40 0033 FD7401                  LD      (IY+1H),H       ;(IY+1H) : LOWER  ORDER BYTE
41 0036 D9                      EXX
42 0037 FD7502                  LD      (IY+2H),L       ;(IY+2H) : HIGHER ORDER BYTE
43 003A FD7403                  LD      (IY+3H),H       ;(IY+3H) : HIGHEST  ORDER BYTE
44 003D C9                      RET                     ;RETURN
45                              END
```

3．Z-80 CPUレジスタ構成

主レジスタセット/MAIN REG SET　補助レジスタセット/ALTERNATE REG SET

(アキュムレータ) ACCUMULATOR A	(フラグレジスタ) FLAGS REG F	(アキュムレータ) ACCUMULATOR A′	(フラグレジスタ) FLAGS REG F′
B	C	B′	C′
D	E	D′	E′
H	L	H′	L′

汎用レジスタ

I (インタラプトページアドレスレジスタ) INTERRUPT VECTOR	R (メモリリフレッシュレジスタ) MEMORY REFRESH
IX (インデックスレジスタ)	INDEX REGISTER
IY (インデックスレジスタ)	INDEX REGISTER
SP (スタックポインタ)	STACK POINTER
PC (プログラムカウンタ)	PROGRAM COUNTER

専用レジスタ

4．アスキーコード表他

アスキーコード表（7ビットコード）

LSD \ MSD	0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0 0000	NUL	DLE	SP	0	@	P	`	p
1 0001	SOH	DC1	!	1	A	Q	a	q
2 0010	STX	DC2	"	2	B	R	b	r
3 0011	ETX	DC3	#	3	C	S	c	s
4 0100	EOT	DC4	$	4	D	T	d	t
5 0101	ENG	NAK	%	5	E	U	e	u
6 0110	ACK	SYN	&	6	F	V	f	v
7 0111	BEL	ETB	′	7	G	W	g	w
8 1000	BS	CAN	(	8	H	X	h	x
9 1001	HT	EM	)	9	I	Y	i	y
A 1010	LF	SUB	*	:	J	Z	j	z
B 1011	VT	ESC	+	;	K	[	k	{
C 1100	FF	FS	,	<	L	\	l	|
D 1101	CR	GS	−	=	M	]	m	}
E 1110	SO	RS	.	>	N	↑	n	~
F 1111	SI	VS	/	?	O	←	o	DEL

16 進 桁

6		5		4		3		2		1	
HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC
0	0	0	0	0	0	0	0	0	0	0	0
1	1,048,576	1	65,536	1	4,096	1	256	1	16	1	1
2	2,097,152	2	131,072	2	8,192	2	512	2	32	2	2
3	3,145,728	3	196,608	3	12,288	3	768	3	48	3	3
4	4,194,304	4	262,144	4	16,384	4	1,024	4	64	4	4
5	5,242,880	5	327,680	5	20,480	5	1,280	5	80	5	5
6	6,291,456	6	393,216	6	24,576	6	1,536	6	96	6	6
7	7,340,032	7	458,752	7	28,672	7	1,792	7	112	7	7
8	8,388,608	8	524,288	8	32,768	8	2,048	8	128	8	8
9	9,437,184	9	589,824	9	36,864	9	2,304	9	144	9	9
A	10,485,760	A	655,360	A	40,960	A	2,560	A	160	A	10
B	11,534,336	B	720,896	B	45,056	B	2,816	B	176	B	11
C	12,582,912	C	786,432	C	49,152	C	3,072	C	192	C	12
D	13,631,488	D	851,968	D	53,248	D	3,328	D	208	D	13
E	14,680,064	E	917,504	E	57,344	E	3,584	E	224	E	14
F	15,728,640	F	983,040	F	61,440	F	3,840	F	240	F	15
0123		4567		0123		4567		0123		4567	
BYTE				BYTE				BYTE			

2のべき乗

2^n	n
256	8
512	9
1 024	10
2 048	11
4 096	12
8 192	13
16 384	14
32 768	15
65 536	16
131 072	17
262 144	18
524 288	19
1 048 576	20
2 097 152	21
4 194 304	22
8 388 608	23
16 777 216	24

$2^0 = 16^0$
$2^4 = 16^1$
$2^8 = 16^2$
$2^{12} = 16^3$
$2^{16} = 16^4$
$2^{20} = 16^5$
$2^{24} = 16^6$
$2^{28} = 16^7$
$2^{32} = 16^8$
$2^{36} = 16^9$
$2^{40} = 16^{10}$
$2^{44} = 16^{11}$
$2^{48} = 16^{12}$
$2^{52} = 16^{13}$
$2^{56} = 16^{14}$
$2^{60} = 16^{15}$

16のべき乗

16^n	n
1	0
16	1
256	2
4 096	3
65 536	4
1 048 576	5
16 777 216	6
268 435 456	7
4 294 967 296	8
68 719 476 736	9
1 099 511 627 776	10
17 592 186 044 416	11
281 474 976 710 656	12
4 503 599 627 370 496	13
72 057 594 037 927 936	14
1 152 921 504 606 846 976	15

●Ｚ－80ファミリについてのデータとか，最新情報は，下記にお問い合わせ下さい。

シャープ株式会社

本　　社	〒545	大阪市阿倍野区長池町２２番２２号	☎(06) 621-1221(大代表)
ＩＣ事業本部	〒632	奈良県天理市櫟本町２６１３番地の１	☎(07436)5-1321(大代表)
国内電子部品営業本部	〒545	大阪市阿倍野区長池町２２番２２号	☎(06) 621-1221(大代表)
東部地区営業	〒162	東京都新宿区市谷八幡町８番地	☎(03) 260-1161(大代表)
北関東駐在	〒361	埼玉県行田市門井町２丁目５番地	☎(0485)53-3127(代　表)
水戸駐在	〒310	茨城県水戸市千波町１９６３番地	☎(0292)43-7600(代　表)
長野駐在	〒399	松本市芳野８番１４号	☎(0263)27-1677(代　表)
静岡駐在	〒422	静岡市曲金６丁目８番４４号	☎(0542)83-0081(代　表)
中部地区営業	〒454	名古屋市中川区山王３丁目５番５号	☎(052)332-2681(代　表)
浜松駐在	〒430	浜松市植松町１４７６の２	☎(0534)65-1207(代　表)
西部地区営業	〒545	大阪市阿倍野区長池町２２番２２号	☎(06) 621-1221(大代表)
福岡駐在	〒816	福岡市博多区井相田２丁目１２番１号	☎(092)582-5245(代　表)

Z－80 プログラミングマニュアル―アセンブリ語―

1986年５月　　第２版第６刷発行

SHARP

シャープ株式会社

本　　社	〒545	大阪市阿倍野区長池町22番22号	☎(06) 621-1221(大代表)
IC事業本部	〒632	奈良県天理市櫟本町2613番地の1	☎(07436)5-1321(大代表)
国内電子部品営業本部	〒545	大阪市阿倍野区長池町22番22号	☎(06) 621-1221(大代表)
東部地区営業	〒162	東京都新宿区市谷八幡町8番地	☎(03) 260-1161(大代表)
北関東駐在	〒361	埼玉県行田市門井町2丁目5番地	☎(0485)53-3127(代　表)
水戸駐在	〒310	茨城県水戸市千波町1963番地	☎(0292)43-7600(代　表)
長野駐在	〒399	松本市芳野8番14号	☎(0263)27-1677(代　表)
静岡駐在	〒422	静岡市曲金6丁目8番44号	☎(0542)83-0081(代　表)
中部地区営業	〒454	名古屋市中川区山王3丁目5番5号	☎(052)332-2681(代　表)
浜松駐在	〒430	浜松市植松町1476の2	☎(0534)65-1207(代　表)
西部地区営業	〒545	大阪市阿倍野区長池町22番22号	☎(06) 621-1221(大代表)
福岡駐在	〒816	福岡市博多区井相田2丁目12番1号	☎(092)582-5245(代　表)

昭和61年5月作成　電部-1801　Z1(6)