ND8080　ND80Zモニタプログラム操作説明書

(有)中日電工

目次

〒463-0067　名古屋市守山区守山2－8－14
パレス守山305
有限会社中日電工
℡052-791-6254　Fax052-791-1391
E-mail　thisida@alles.or.jp
Homepage　http://www.tyunitidenko.x0.com/

2016. 4. 29　Rev.　1. 0

1章　基本操作

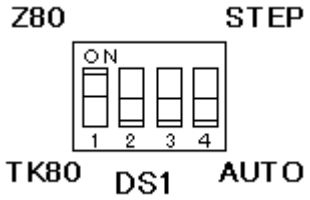
1．　はじめに

ND80ZモニタプログラムはTK80モニタプログラムの機能を拡張したものです。
ND80ZモニタプログラムはCPUがZ80であることを前提としています。
ND8080のCPUは8080なので、ND80Zモニタの一部の機能は使うことができません。
そのことを理解していただいたうえで、ND80Zモニタを使ってください。

ND80Zモニタプログラムの基本的な機能はTK80モニタプログラムと変わりません。
そこで、ND80ZモニタプログラムとTK80モニタプログラムとで共通している事柄については、TK80モニタプログラム操作説明書でまとめて説明することにしました。ですから、まず先にTK80モニタプログラム操作説明書を読んで、基本的な操作や基礎知識について理解してから、本書を読むようにしてください。

モニタプログラムは、ディップスイッチDS1で選択するようになっています。
ND80ZモニタプログラムはディップスイッチDS1のNo.1がON、No.2がOFFのときに選択されます。
ディップスイッチの設定は、電源投入時かまたはリセットスイッチを押したときに検出されます。
それ以外のときに設定を変更しても、状態は変化しませんから、電源が入っているときに設定を変更した場合、その設定を有効にするためには、リセットスイッチ（MONキー）を押す必要があります。

電源を入れる前に、図のようにディップスイッチDS1のNo.1がON、No.2がOFFになっていることを確認してください。
もし図のようになっていないビットがあれば、小型のマイナスドライバなどで、図と同じになるようにしてください。
電源ON後に設定を変更した場合には、変更後にリセットスイッチ（MONキー）を押すと、設定が有効になります。
No.3、No.4はRS232C通信のボーレート設定用です。通常はOFFにしておいてください。



2．　キーボード

ND80Zモニタプログラムに色々な指示を与えたり、メモリの中身を読んだり書いたりするときには、それらの作業は全てキーボードからの入力によって行います。
ND8080のキーは、５×５配列のキー25個です。この25個のキーは、その働きによって、次の3つのグループに分けられます。

2．1　データキー

［0］［1］［2］［3］［4］［5］［6］［7］［8］［9］［A］［B］［C］［D］［E］［F］

メモリアドレスを指定して、データを書き込んだり、プログラムを入力するときの、16進数のキーです。
0～Fのほかにも、色々な記号がついていますが、この記号は［＊（I／O）］キーや［REG］キーを使うときのものです。
それについては2章以降で説明します。ここでは16進数の入力キーとして、覚えて下さい。

2．2　ファンクションキー

［CONT］［RUN］［＊（I／O）］［REG］［ADRSSET］［READINC］［READDEC］［WRITEINC］

［＊（I／O）］キーや［REG］キー以外のキーの操作方法や、キーの機能はTK80モニタプログラムと同じです。
基本的なキー操作については、「TK80モニタプログラム操作説明書」を参照してください。

2．3　リセット（モニタエントリ）キー

［MON］

実行中のプログラムを中止してモニタプログラムに戻ります。
もう少し詳しい説明が「TK80モニタプログラム操作説明書」にありますから参照してください。

2章　プログラムデバッグの仕方

1．ステップ動作

ND80Zモニタプログラムのステップ動作は、ワークアドレスが異なっていることおよび退避されるレジスタがZ80の全レジスタであることを除いてはTK80モニタプログラムと同じです。
ワークアドレスや対象になるレジスタについては、2．ブレイク動作で説明をします。
そのほかの基本的な操作については、ここでは省略しますから、具体的な操作については「TK80モニタプログラム操作説明書」を参照してください。

2．　ブレイク動作

ND80Zモニタプログラムのブレイク動作についても、基本的な機能はTK80モニタプログラムと同じです。
ただワークアドレスは異なります。
ブレイクカウンタやブレイクアドレスの設定、退避されたレジスタの値の確認などがTK80モニタプログラムに比べて簡単なキー操作で行うことができます。
ここではTK80モニタプログラムの操作例と同じように、「TK80モニタプログラム操作説明書」の2章で作ったプログラムをブレイクさせてみます。

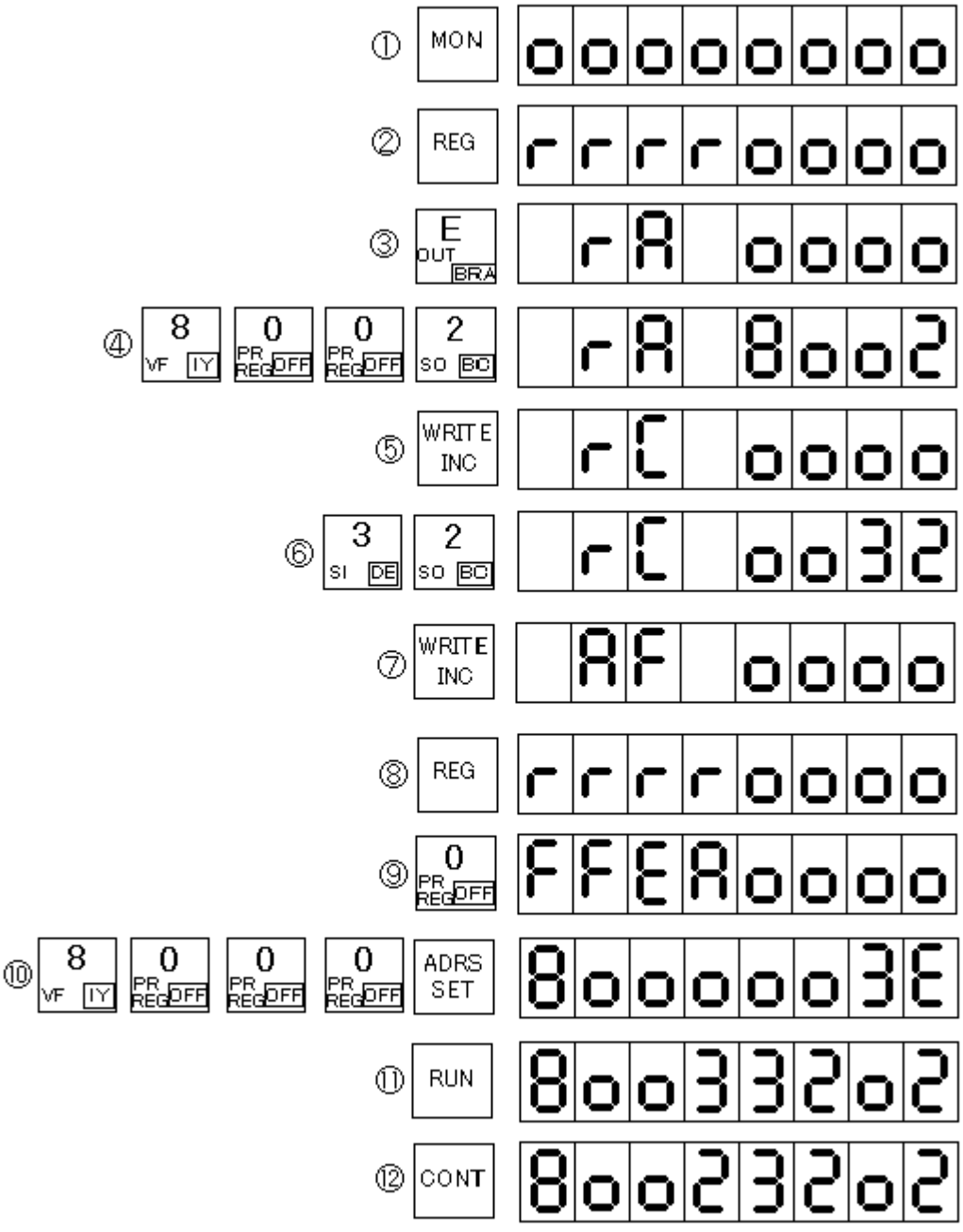
```
8000  3E00          MVI  A, 00
8002  3C            INR  A
8003  C30280        JMP  ＄8002
```

8002番地を50回実行したあとステップ動作に移る(ブレイクする)ようにセットします。
図2－1を参照しながら、以下の説明を読んで下さい。

①まずリセットします。(ブレイクの時は必ずしもリセットから始めなくてもよいのですが、この方が確実です)
②ブレイクアドレスをセットします。
まず［REG］キーを押します。
このキーについては後で詳しく説明します。いまは図2－1 の通りに操作して下さい。
［REG］キーを押すと、アドレス表示部にはrrrrが表示されます。
③データキーの［E］(白抜き文字の［BR　A］)を押します。するとアドレス表示部に　rA　と表示され、データ表示部にはブレイクアドレスが表示されます(リセツト後は0000になっています)。
④ブレイクしたいアドレスを入れます。今回は8002をセットします。
⑤［WRITE　INC］キーを押します。
これでブレイクアドレスがセットされ、アドレス表示部には　rC　(ブレイクカウンタ)が表示されます。
このときデータ表示部にはブレイクカウンタの値が表示されます(リセット後は0000になっています) 。
⑥ブレイクカウンタに値をセットします。今回は50回にします。10進の50は16進では32になります。そこで［3］［2］と入力します。
ブレイクカウンタは16進2桁です。LEDのデータ表示部は4桁ありますが、上位2桁に何が表示されていても、その上位2桁の値は無視されます。
ブレイクカウンタにセットできる値は01～FF(十進数の0～255)です。
⑦［WRITE　INC］キーを押します。
これでブレイクカウンタのセットができました。
⑧もう一度［REG］キーを押します。
⑨最後にデータキーの［0］(白抜き文字の［OFF］)を押します。
以上で準備完了です。このときアドレスレジスタにはFFEAが表示されますが、気にしないで下さい。
⑩プログラムの開始アドレスをセットして下さい。［8］［0］［0］［0］［ADRSSET］と入力します。
⑪ディップスイッチDS1の№4がON(STEP側)になっていることを確認して下さい。
OFF(AUTO側)になっていると、ブレイクできません。
［RUN］キーを押すと瞬間にプログラムが50回実行されて、ブレイクします。50回実行した証拠に、データ表示部の上2桁にはAレジスタの値、32(十進数の50)が表示されています。

これ以後はステップ操作と同じです。[CONT]キーを押すと1ステップずつ進みます。
ブレイクカウンタはブレイク時点で0になります。ブレイクアドレスは[MON](RESET)キーを押さない限りクリアされないでそのまま残ります。

〔注意1〕(削除)
〔注意2〕ステップ動作は割り込み(RST7)を利用しています。したがってもしユーザープログラムの中で、割り込み禁止命令(DI)を使うと、それ以後は割り込みが禁止されるため、ステップ動作ができなくなります。
〔注意3〕ブレイクカウンタが0になっている時は、ブレイク動作ではなくてステップ動作になります。
〔注意4〕ブレイクアドレスは各命令の1バイト目でなければいけません。今回の例では8000、8002、8003は指定できますが、8001、8004、8005を指定してはいけません。
〔注意5〕ブレイクアドレスを設定した場合、ブレイクするまでの間のプログラム実行時間は、通常処理の場合の数十倍かかります。これはブレイクアドレス以外のプログラム部分でも1ステップ実行する毎に、ブレイク処理プログラムが実行されているためです。



(図2－1)

3. レジスタモード

レジスタモードはZ80のレジスタを対象にしています。

Z80は8080に比べて、非常に豊富なレジスタをもっています。
ND8080では8080にはないレジスタに対する機能は使うことができません。
レジスタモードのキー操作ではZ80固有のレジスタに対する操作はしても意味がありませんが、プログラムの整合性を維持するために、Z80のレジスタに対する操作を除外するようにはプログラムしてありません。
そのことを理解したうえで以下の説明を読んでください。

レジスタモードの機能を使うことにより、ステップ動作やブレイク時に、その時点でのレジスタの値を確認したり、変更したりすることが簡単にできます。
またレジスタに特定の値をセットしてから、プログラムをスタートさせることもできます。
具体的な操作方法について説明する前に、まずZ80Aのレジスタについて簡単に説明しておきます。

3．1　8080のレジスタ

8080は内部に下記のレジスタを持っていて、これらのレジスタはプログラムの中で色々な処理に利用されます。

← 8ビット→	← 8ビット→
A	F
B	C
D	E
H	L

← 16ビット →
SP
PC

[A]
一般にアキュムレータ(加算器)と呼ばれているように、演算命令はこのレジスタを中心に行われます。
[F]
フラグレジスタ。命令の実行により現れる色々な状態を1ビットずつに記録して保持します。各ビットの意味は3．2で説明します。
[B][C]
共に 8 ビットのレジスタとして独立して使うこともできますが、つないで 16 ビットのレジスタ[BC]として使うこともできます。その場合には[B]が上位8ビット、[C]が下位8ビットになります。[B]または[BC]をカウンタとして使っている命令がいくつかあります。
[D][E]
B、Cと同じですがカウンタとして使う命令はありません。
[H][L]
B、Cと同じですがカウンタとして使う命令はありません。16ビットレジスタ[HL]はメモリアドレスを入れて、メモリを間接的に示すときにも使われます(間接アドレッシングモード)。また[HL]は16ビットの加減算命令で加算器(アキュムレータ)としても使われます。
[SP]
スタックポインタ。現在のスタックのトップ・アドレスを示しています。スタックについては、3．3で説明します。
[PC]
プログラムカウンタ。現在実行中のアドレス(正しくは、次のアドレス)を管理しています。

3．2　8080のフラグ

フラグは8ビットのフラグレジスタに、下図のように割りつけられています。

S	Z		H		P		C	
7	6	5	4	3	2	1	0	(ビット位置)

各記号の意味は下の通りです(ビット1、3、5は使用されません)。
なお、フラグがセットされたときは、そのビットが1になり、リセットされたときは0になります。

C

キャリ・フラグ。計算の結果、上位桁へのキャリー、ボローが発生したときにセットされます。ローテイト命令でもセット、リセットされます。

P

パリティフラグ。論理、算術演算およびINR、DCR命令を実行した結果、1のビットが偶数個あるときにセットされ、奇数個のときはリセットされます。

H

ハーフ・キャリ・フラグ。算術演算でビット3からビット4へのキャリーや、ビット4からビット3へのボローがあったときセットされます。このフラグはCフラグとともに、BCD演算後のDAA命令で利用されます。

Z

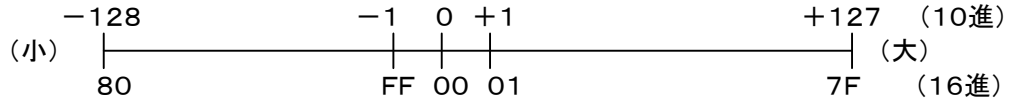
ゼロ・フラグ。結果がゼロのときセットされます。

S

サイン・フラグ。結果が負のときセット、正またはゼロのときリセットされます。

8ビットの数00～FFは符号なしでは10進の0～255として扱われますが、符号付の数として扱ったときには、－128～＋127の数になり、これは16進では80～7Fになります（ビット7が0のときはその数は正で、ビット7が1のときは負になります）。

●符号付8ビットの数の大小



3．3 スタック

大きなプログラムになると、レジスタもたくさん必要で、とても3．1で説明した数では足りません。そこでレジスタの値をひとまずメモリのワークエリアにしまっておいて、そのレジスタを次の用途に使う、ということが簡単にできると便利になります。

ところがレジスタの値をしまうときに、一々異なるメモリアドレスに割りつけていくのでは大変です。

そんなときにこのスタックを使えば、メモリアドレスを指定しなくても簡単な操作でレジスタの値を保存することができます。

スタックとは積み重ねるという意味です。

ちょうど本などを積み重ねるように、メモリの中にレジスタの値を順番にしまうことができます（PUSH命令を使います）。

取り出すときは、入れたときと逆の順番で取り出します（POP命令を使います）。

そしてそのスタックの現在の位置を管理しているのがSP（スタックポインタ）です。

（例）SP＝8300、BC＝1234、DE＝5678のとき、

```
PUSH  BC
PUSH  DE
```

を実行すると、メモリ内容は下のようになります。

8300		←命令実行前のSPの位置
82FF	12	
82FE	34	
82FD	56	
82FC	78	←命令実行後のSPの位置（SP＝82FC。BC、DEは変化しない）
82FB		

この後でPOP　HLを実行すると、下のようになります。

8300		
82FF	12	
82FE	34	←命令実行後のSPの位置（SP＝82FE。HL＝5678になる。BC、DEは変化しない）
82FD	56	
82FC	78	
82FB		

こうなってしまった後で、POP　DEを実行してもDEにはもとの値は戻りません（1234が入る）。

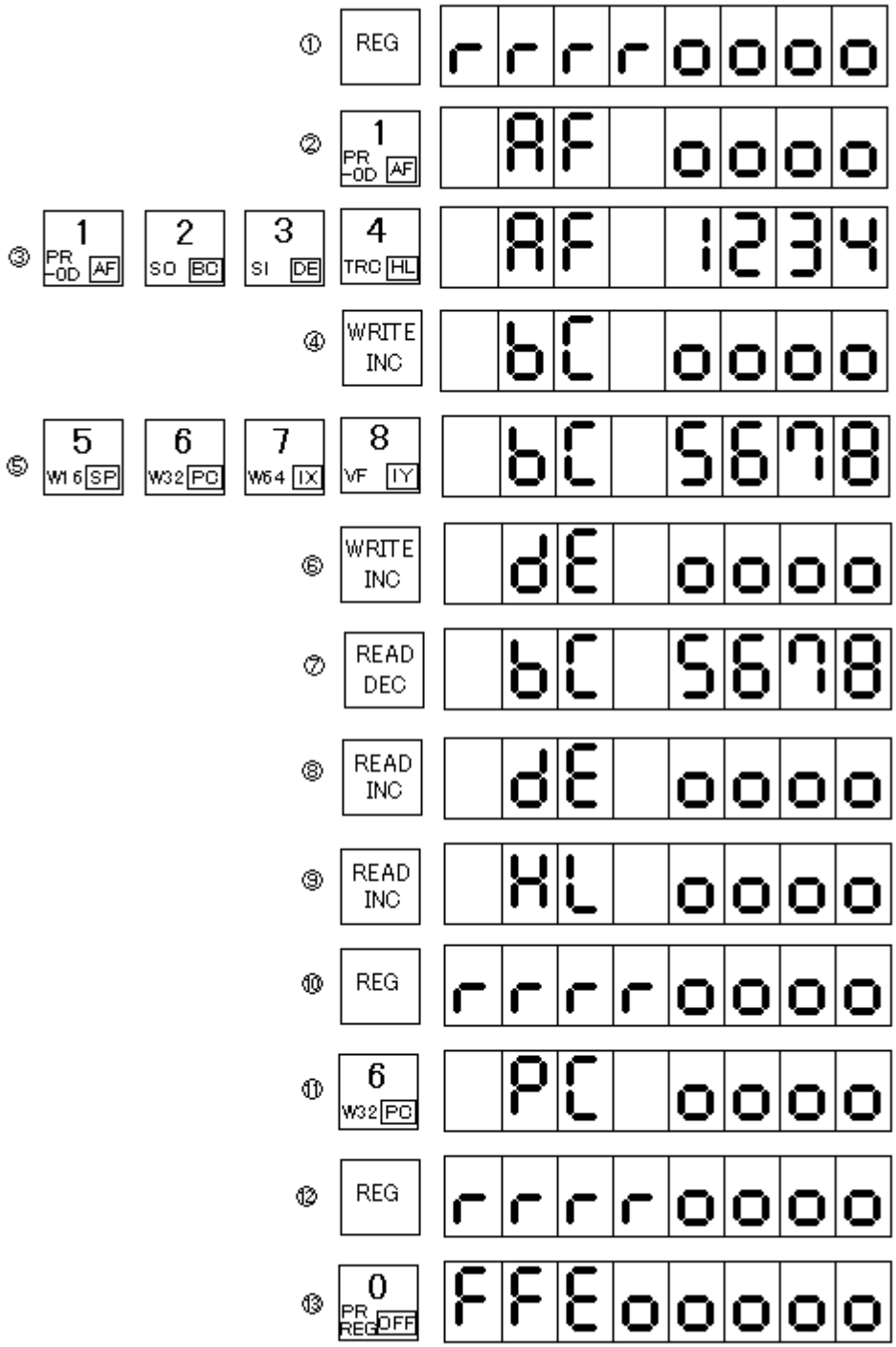
ここではスタックの特殊な使い方として説明しました。

一般的な使い方としては、一時的に退避しておきたいレジスタを、PUSH命令でスタックに保存しておいて、あとでそれをもとのレジスタに戻すときは、PUSHとは「逆の順序」でPOPを実行します。

PUSH BC、PUSH DE の順で実行したら、それをもとに戻すときはPOP DE、POP BCのようにPUSHのときの逆の順序にします。
PUSH、POPは常に順番を覚えておいて、間違わないように使う必要があります。
〔注意1〕スタックの操作はPUSH、POPだけではなくCALL、RET命令や割り込み処理でも使用されます（アドレスがスタックに入れられる）。
〔注意2〕スタックはメモリ上のどこにでも設定することができます（LD SP命令を使う）。
しかし指定場所によってはプログラムやデータの入っている領域と重なってしまい、その結果プログラムやデータが壊されてしまうことがあるので、充分注意が必要です。
マシン語プログラムでは、普通はその先頭部分でスタックポインタのセットが必要ですが、ND80Zモニタプログラムではリセット後はSP＝F800にセットされるのでユーザーがあらためてスタックポインタをセットする必要はありません。

3．4 レジスタモードの操作方法

レジスタモードは[REG]キーとデータキーを組み合わせて使います（ 以下の説明は次頁の図2－2を参照しながら読んで下さい）。
①[REG]キーを押すとアドレス表示部にrrrrが表示され、レジスタモードであることを示します。
rrrrが表示されているときには、データキー以外のキーは押してはいけません。
処理を中止するためにリセット（[MON]）キーを押すことはできます。
②データキーはZ80のレジスタに対応しています。
レジスタモードで、[REG]キーを押して、rrrrが表示されているときは、データキーの白抜き文字が有効になっています。
データキーの[1]（白抜き文字の[AF]）を押して下さい。アドレス表示部にAFと表示され、データ表示部にはステップ動作やブレイク動作で中断した時点のAレジスタとFレジスタの内容が表示されます。（リセット後はSP＝F800以外は全レジスタ＝0000になっています）
この状態のときは、データ入力キーとして[0]～[F]キーと、[READ INC]、[READ DEC]、[WRITE INC]が使用できます。
③レジスタ内容を変更したい場合は、データキーを使って、値をセットします。例として[1][2][3][4]とキー入力してみます。
④最後に[WRITE INC]を押します。これでAレジスタに12が入り、Fレジスタに34が入れられました。
正確に言うと、この時点ではまだメモリ上のレジスタセーブエリアが書き換えられただけで、CPUレジスタに書き込まれた訳ではありません。[CONT]または[RUN]キーを押したときに、レジスタセーブエリアからCPUレジスタに値がセットされます。
アドレスレジスタには次のレジスタペアBCが表示されます。
⑤これも変更したければ、同じように操作します。[5][6][7][8]と入れてみます。
⑥[WRITE INC]を押すと、さらに次のレジスタペアDEが表示されます。
⑦今書き込んだ内容を確認することもできます。
[READ DEC]キーを押して下さい。
表示がBCに戻って、データ表示部には今書き込んだ内容が表示されます。
⑧逆に書き込みをしないで、先に進んでレジスタ内容を見たいときは、[READ INC]キーを押します。
⑨続けて押せば、一つずつ表示が進んでいきます（[READ DEC]も同様です）。
⑩順番に確認するのではなくて、必要なレジスタだけ見たい場合には、もう一度[REG]キーを押します。
⑪続けて確認したいレジスタのキーを押します。たとえば[6]（[PC]）を押すと、PCが表示されます。
⑫レジスタ設定モードを終了するときは、先に[REG]キーを押し、
⑬続いて、データキーの[0]（[OFF]）を押します。これで普通のモニタの状態に戻ります。
アドレス表示部には、OFFになった位置によって異なるアドレスが表示されますが気にしないでください。
以上の作業がブレイク後またはトレース後に行われたときは、このあとで[CONT]キーを押すことによって、上の作業で最終的に書き換えられたデータを各レジスタにセットしたあと、ユーザープログラムに復帰します。
またブレイク動作やトレース動作以外の普通の処理でも、まずこのレジスタモードで必要な値をセットしたあと、RUNすることにより、CPUレジスタに特定の初期値を持たせてユーザープログラムを開始させることができます。
以上のようにこの機能によって、実行途中のCPUレジスタの値を簡単な操作で参照できるばかりではなく、必要ならば途中で各レジスタの値を簡単に変更することもできるため、非常にきめ細かなデバッグ作業が行えます。



（図2－2）

〔注意1〕rrrr表示のときは、データキー以外のキーのうち、[CONT][RUN][＊(I／O)][REG][ADRS SET]の各キーは押しても無視されますが、[READ INC][READ DEC][WRITE INC]の各キーを押すと、異常表示やレジスタセーブエリアの値が書き換えられたりします。

〔注意2〕プログラムカウンタ(PC)やスタックポインタ(SP)の値を不用意に変更すると、以後のユーザープログラムがまともに実行されなくなることがあります。

〔注意3〕ブレイクアドレス(BR A)やブレイクカウンタ(BR C)はCPUレジスタではありませんが、機能から考えてこのレジスタモードに入れてあります。

〔注意4〕USBコネクタにUSBケーブルを接続して、Windowsパソコンと接続することにより、ブレイク時やトレース時に全レジスタの内容を一度にWindowsパソコンのディスプレイに表示させることができます(USB接続については5章で説明します)。

3章　I／Oモード（IN／OUT）

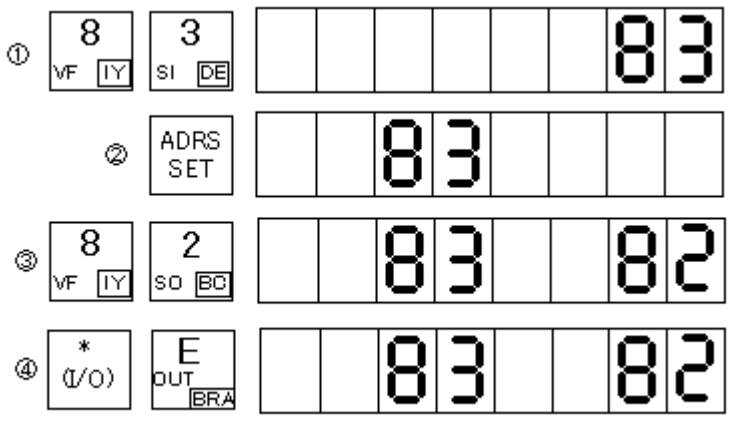
1．　IN、OUTキーの使い方

2章では[REG]キーを使って、レジスタセーブエリアの参照や値の変更を行う操作について説明をしました。
3章では、もうひとつの特殊機能キーの[＊（I／O）]キーの基本的な操作について説明をします。
[＊（I／O]キーはもともとは、メモリアクセスと同じようにI／Oアクセスも簡単なキー操作で行えるように、という目的で、用意したものなのですが、ND80Zモニタの多様な機能を簡単なキー操作で行うときにも、[＊（I／O）]キーが利用されます。そのために、キーシールには[I／O]だけではなくて、その他のいろいろな機能を意味する記号として[＊]が付け加えられています。
その他のいろいろな機能については4章以降で説明を行います。
3章では、[＊（I／O）]キーの基本的な使い方として、I／O回路に対する入出力操作（IN／OUT）について説明をします。

8080はメモリについては0000～FFFFの64KBのアドレス空間をアクセスすることができますが、これとは全く独立して82C55などのI／Oデバイスについても00～FFの256バイトのアドレス空間をアクセスできます。
メモリに対しては、キーボードからアドレスを入力するだけで、簡単にREAD、WTRITEすることができました。
ND80ZモニタプログラムではI／Oに対しても、同様の簡単な操作でIN、OUTができます。
次にその操作例を示します。
ND8080に実装されている82C55に対してIN、OUTの操作をしてみます。

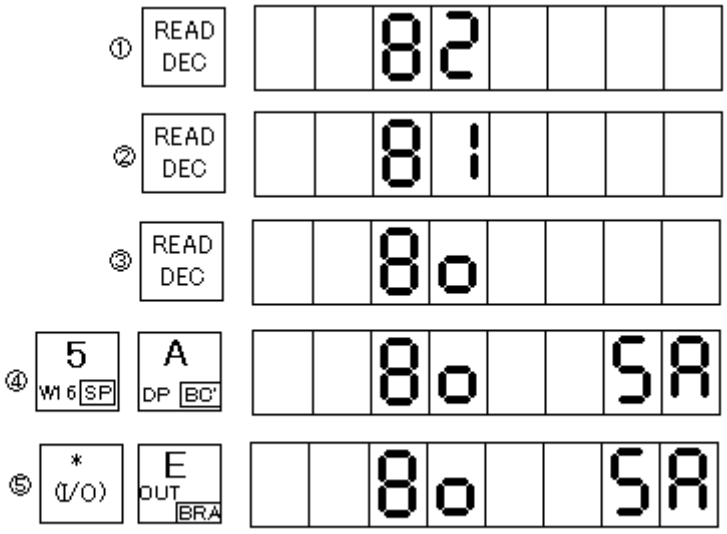
82C55は、最初にAポート～Cポートの入出力の向き決めるためのコントロールワードをコントロールアドレスに与える必要があります。
82C55については、「TK80モニタプログラム操作説明書」の　5章　I／O制御　を参照してください。
ND8080に実装されている82C55のI／Oアドレスは80～83です。
80がAポート、81がBポート、82がCポート、83がコントロールワードアドレスです。
例としてAポートとCポートを出力に、Bポートを入力に設定してみます。
コントロールワードは82（10000010）です。

①まずI／Oアドレスをセットします。コントロールワートアドレスは83です。
[8][3]と入力します。メモリと違って下2桁だけが有効です（上2桁は何になっていても構いません）。
LED表示の空白部分には何が表示されていても構いません。誤解を避けるために空白にしました。
②[ADRS　SET]キーを押すとそのアドレスの「メモリ」の値が、データ表示部に表示されます（このときはまだ「I／Oの値ではありません」）。このとき何が表示されても、気にしないで下さい。
③コントロールワード82を入力します（[WRITE　INC]キーを押してしまわないように注意してください）。
④[＊（I／O）]キーに続いて、データキーのE（[OUT]）を押します。
このとき表示は変化しませんが指定したI／Oアドレスに対してデータの出力が行われます。
82C55のコントロールワードアドレスにコントロールワードの82が与えられましたから、82C55のAポートとCポートが出力に、Bポートが入力に設定されました。
82C55の入出力コネクタCN1の各端子をテスターで測定してみてください（CN1の端子接続図は「ND8080取扱説明書」にあります）。
82C55の、出力に設定されたポートは、設定直後は全ビットが0になります。
入力に設定されたポートはハイインピーダンスになりますが、ND8080に実装してある82C55は、各ポートの全ビットが抵抗（10KΩ）で＋5Vにプルアップしてありますから、入力に設定されたポートのビットは＋5Vになります。



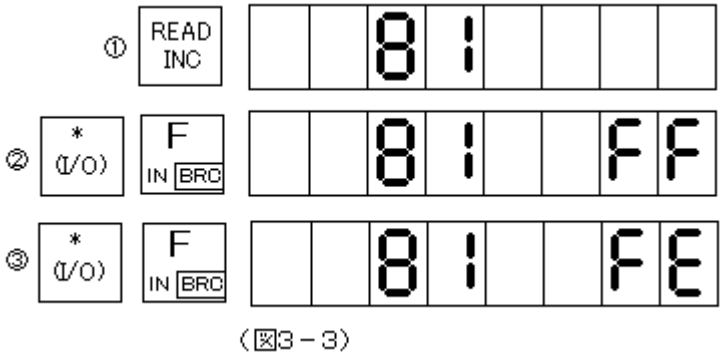
（図3－1）

Aポートにデータを出力してみます。
①Aポートのアドレスは80です。I／Oアドレスの80を、さきほどの例のように[ADRS SET]キーでセットしてもよいのですが、せっかくI／Oアドレスの83が表示されているのですから、[READ DEC]キーを使ってみます。
アドレスが－1されて82が表示されます。
②もう一度[READ DEC]キーを押すと表示が81になります。
③もう一度[READ DEC]キーを押すと表示が80になります。
これでAポートのアドレスが選択されました。
④出力データをセットします。データ5Aをセットしてみます（ここで[WRITE INC]キーを押してしまわないように注意してください）。
⑤[＊(I／O)]キーに続いてデータキーの[E]([OUT])を押すと、表示は変化しませんがこのときAポートから5A(01011010)が出力されています。
Aポートの出力をテスターで確認してみて下さい。1のビットは＋5V、0のビットは0Vになっています。



（図3－2）

Bポートが入力に設定されているので、このBポートからデータを入力してみます。
①Bポートのアドレスは81です。今はI／Oアドレスの80が表示されていますから、[READ INC]を押して、アドレスを81にします。
②[＊(I／O)]キーに続いて、データキーの[F]([IN])を押します。するとデータ表示部の下2桁にFFが表示されます。
Bポートは抵抗(10KΩ)で＋5Vにプルアップしてありますから、入力がない場合には、全ビットが1になります。
③附属品の26pinフラットケーブルを利用して、pin№.25(PB0)とpin№.10(GND)とをつないでください。そうしておいて、もう一度[＊(I／O)]キーに続いて、データキーの[F]([IN])を押します。今度はデータ表示部の下2桁がFEに変わります。これは PB0 だけが0になったためです(11111110＝FE)。



（図3－3）

4章　プログラム、データのSAVE、LOAD

ND8080の側のキー操作がTK80モニタでの操作と異なっています。

1. USB接続

ND8080のRAMはボタン電池でバックアップをしていますから、RAMに書き込んだプログラムやデータは電源を切っても消えずにそのまま残っています。
しかし複数のプログラムをRAMに常駐させて保存するというのはあまり感心できる方法ではありません。プログラムが暴走したりすれば、プログラムもデータも一瞬で破壊されてしまいます。
ND80Zモニタにはプログラムやデータを SAVE、LOADする機能があります。
USBコネクタにUSBケーブルをつないで、パソコンと接続することによって、RAMにあるプログラムやデータをパソコンに送り、ファイルとして保存することができます。
逆にパソコン上で作成したマシン語のプログラムをUSB接続でND8080のRAMに送ることもできます。
そのためには、パソコン側でもUSB（HID）READ、WRITEをするプログラムを用意する必要があります。
そのようなプログラムを自作することも可能ですが、ND8080組立キットには附属ソフトウェアとして、ND8080をUSB（HID）で接続して、パソコンのキーボードからND8080を操作する、リモートモード＋ZB3BASICプログラムがついていますから、それを使った方が簡単です。
しかしここでは、ND80Zモニタプログラムの基本的な機能として、リモートプログラムではなくて、ただのUSB（HID）送信、受信プログラムをパソコン側で実行する場合について、説明をします。

ここではパソコン側のハードディスクにHID受信プログラム（HIDRD. EXE）、HID送信プログラム（HIDWR. EXE）がND8080附属CDROMからCOPY済みで、ND8080とWindowsパソコンがUSBケーブルで接続されているものとして説明します。
それらのプログラムをCDROMからハードディスクにCOPYする作業については、「USB接続説明書」を参照してください。

1－1. プログラムのSAVEの仕方

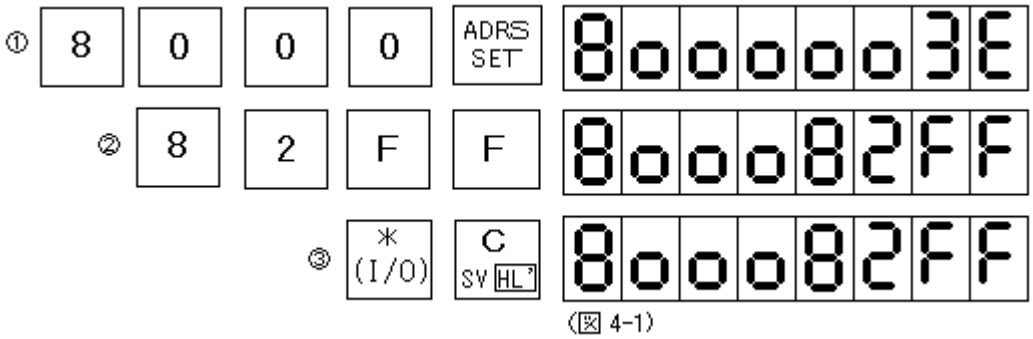
［Windowsパソコン側の操作］
ND8080の操作をする前に、Windowsパソコン側でHID受信プログラムを実行しておいてください。
コマンドプロンプト画面で、

hidrd8　xxxxx. btk［Entr］

と入力します。xxxxx. btkはプログラムを保存したい任意のファイルネームにします。
. 以下の拡張子はbtkでなければならない、ということではありませんが、TK80のSAVE／LOAD機能では、データの先頭に4バイトの開始アドレス、終了アドレスが置かれますから、一般的なバイナリファイル（拡張子bin）と区別する意味で、btkを使うことをおすすめします。btkは、binary file　for　TK80の意味です。
画面は以下の表示になって、データ受信待ちになります。

［入力例］
C:¥nd8080>hidrd8　test. btk[Enter]
送信を開始してください

［ND8080の操作］
図 4-1 の通りに操作して下さい。ここでは例として8000～82FFの内容をSAVEすることにします。



(図 4-1)

①SAVE開始アドレス（この例では8000）をアドレス表示部にセットします。データ表示部には8000番地の内容が表示されます（ここでは3Eになっています）。
②続いてデータ表示部に、SAVE終了アドレスを入力します（[WRINC]キーは絶対に押さないように!!） 。
③最後に[＊（I／O）]に続けて[C SV]を押すと送信が開始されます。
送信中や送信が終了しても7セグメントLEDの表示は変化しませんが、Windowsパソコンのコマンドプロンプト画面には、受信が完了する以下のように表示されます。

```
s=8000,e=82ff
received data 772(=304) bytes
```

1－2. プログラムのLOADの仕方

SAVEの場合とは逆に、先にND8080のキー操作を行います。

[ND8080の操作]
LOADの操作は簡単で、[＊（I／O）]に続けて[D LD]を押すだけです。
7セグメントLEDの表示は変化しませんが、[RESET]キー以外は受け付けなくなります（[D LD]キーはすぐに離してください）。
下はリセット後に[＊（I／O）][D LD]とキー入力したときの例です。

（図4－2）

[Windowsパソコン側の操作]
hidwr8 xxxxx. btk[Entr]
と入力します。xxxxx. btkはND8080に送りたいプログラム、データのファイルです。

[入力例]
C:¥nd8080>hidwr8 test. btk[Enter]

受信が完了すると、ND8080の7セグメントLEDには、メモリに格納された先頭のアドレスと終りのアドレスが表示されます。

（図4－3）

[注記1]
SAVE、LOAD終了後、アドレスが表示されている状態では普通のキー入力モードになっています。したがってこの状態ですぐにRUNキーを押せば、今SAVE（またはLOAD）したプログラムをただちに実行させることができます。またRDINCキーなど他のキーを使うこともできます（キー入力に先立ってリセットする必要はありません）。
なおSAVE、LOADの説明では、「プログラム」のSAVE、LOADということで説明してきましたが、プログラムでも単なるデータでも、扱いは全く同じです。
[注記2]
ND80ZモニタプログラムのSAVEプログラムは、送信データの先頭に2バイトの送信開始アドレスと同じく2バイトの送信終了アドレスを送ります。
[注記3]
LOADは受信したアドレス情報に従って、指定されたメモリアドレスに受信したデータを書き込みます。
指定されたアドレスがROMのアドレス（0000～7FFF）であってもエラーにはなりませんが、結果としては受信しなかったのと同じことになります。
また受信したアドレス情報が、モニタプログラムのワークエリア（83xxまたはF800～FFFF）を示していた場合にはデータの受信によってモニタプログラムが暴走してしまうことがあります。
暴走したときはリセットすれば初期状態に戻ります。

2. RS232C接続

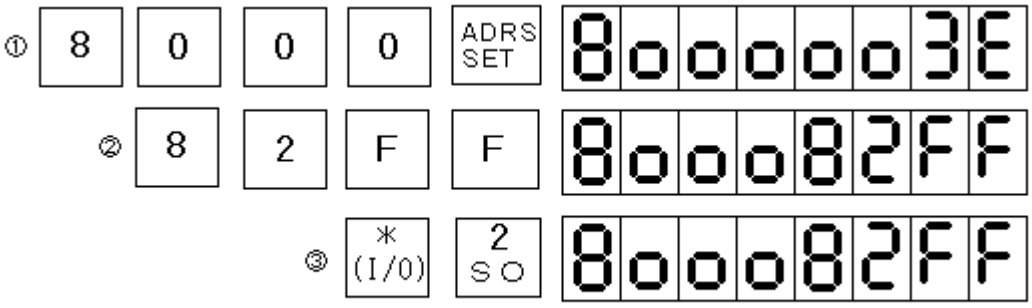
ND8080にはUSBインターフェースのほかに、RS232Cインターフェースも内蔵していますから、RS232Cケーブルで他のパソコンやRS232Cインターフェースを持った機器との間で、プログラムやデータの送受信を行うことができます。
TK80モニタでもこの機能は使えますが、送信受信のために短いプログラムを書く必要があります。
ND80Zモニタでは前項のSAVE／LOADと同様の簡単なキー操作で送信、受信を行なうことができます。

［注記］
送信データはASCIIコードに変換されません。
メモリに書かれている通りの8ビットデータとして送信します。
また先頭に送信開始、終了アドレスは付加されません。
受信についても同様です。

2－1. RS232Cの送信

RS232Cの送信は、［＊（I／O）］キーに続いて、データキーの［2（SO）］を押すことで行われます。
SOはSerial Outです。
送信前に、送信したいデータ（またはプログラム）が格納されているメモリエリアの先頭アドレスを7セグメントLEDのアドレス表示部に表示し、終了アドレスをデータ表示部に表示しておくことで、その指定した範囲のデータが送信されます。
USBを経由してプログラム、データを送信するときと違い、先頭アドレス、終了アドレスそのものは送信されません。
送信データはASCIIコードに変換することなく、バイナリデータのまま送信されます。

データの送信開始アドレスが8000、終了アドレスが82FFであるときの、キー操作です。

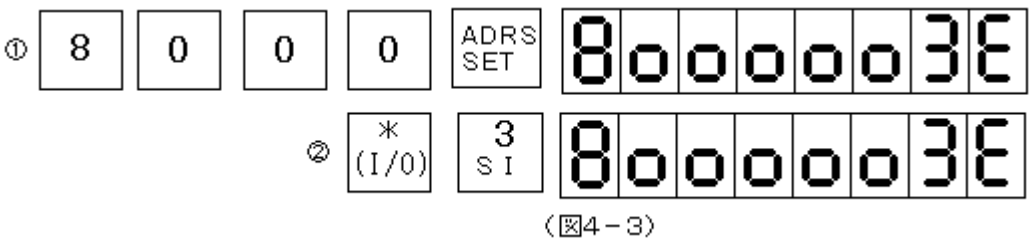


（図4－2）

送信中は7セグメントLEDのアドレス表示部には送信開始アドレスが表示され、データ表示部には現在送信中のアドレスがモニタ表示されます。
送信が完了すると、送信開始アドレスと終了アドレスが7セグメントLEDに表示された状態で静止しますが、この時点でRS232C送信プログラムは終了してTK80モニタのエントリルーチンに戻っていますから、普通にキー操作をすることができます。

2－2. RS232Cの受信

RS232Cの受信は、［＊（I／O）］キーに続いて、データキーの［3（SI）］を押すことで行われます。
SI はSerial Inです。
受信前に、受信データを格納したいメモリエリアの先頭アドレスを7セグメントLEDのアドレス表示部に表示しておくことで、受信データが指定したメモリアドレスから格納されます。



（図4－3）

受信中は7セグメントLEDのアドレス表示部には送信開始アドレスが表示され、データ表示部には現在受信中のアドレスがモニタ表示されます。

受信が終了（送信データが途切れる）しても、そのまま受信待ちで停止しています。ふたたび送信が開始されると、受信動作が再開されます。

送信プログラムと違い、受信の場合には受信データが完了したことを知ることができません。

RS232Cの送信、受信では、[＊(I／O)][C(SV)]、[＊(I／O)][D(LD)]キーを使ってUSB通信を行うときのように、データの先頭に開始アドレスと終了アドレスを置くことはしません。

実際に送信、受信するデータのみを送受信しますから、受信プログラムでは送信が完了したのか、それとも送信側の都合で一時的に送信データが途切れているのかを判断することができません。

もし送信が完了していて、これ以上受信を待つ必要がない状態になったら、リセットすることで、通常のモニタ操作に戻ることができます。

5章　USB接続

1．リモートプログラム

ND8080はWindowsパソコンとUSBケーブルで接続して、データやプログラムをWindowsパソコンのハードディスクにSAVEしたり、逆にあらかじめ作成してハードディスクに保存しておいた、ND8080用のマシン語のプログラムファイルやデータファイルをUSB経由でND8080にLOADすることができます。
そのほかデバッグのためのモニタプログラムの機能のうちのいくつかは、USB接続によって、Windowsパソコンのディスプレイに表示させることで、より効果的に使うこともできます。

USBケーブルでWindowsパソコンと接続して、それによって拡張されたモニタプログラムの機能を使うためには、Windowsパソコン側にもそのためのプログラムをロードして実行させる必要があります。
そのプログラムはND8080組立キットに附属しているCDROMにあります。
ND8080リモート＋ZB3BASICプログラムです。

ND8080のモニタプログラムの通常の機能は、ND8080のキーボードを操作して使いますが、WindowsパソコンとUSBで接続して、ND8080モニタの拡張機能を使うためには、キー操作はND8080のキーで行うのではなくて、Windowsパソコンのキーボードから行います。

ND8080リモートプログラムの準備や起動方法、使い方などについては、ND8080リモートプログラム操作説明書を参照してください。

2．レジスタダンプ

2章で説明したステップ動作やブレイク動作を、WindowsパソコンとUSBで接続して、Windowsパソコン側でND8080リモートプログラムを起動させることによって、[REG]キー＋データキーの操作で一つずつレジスタの中身を確認しなくても、全レジスタの値を一度にWindowsパソコンのコマンドプロンプト画面に表示させることができます。
USB接続でこの機能を使うときは、ND8080のキーから操作をするのではなくて、Windowsパソコンのキーボードからキー入力操作を行います。
具体的な操作の仕方については、ND8080リモートプログラム操作説明書を参照してください。

3．トレース（TRACE）機能

ステップ動作は、[CON]キーを繰り返し必要なだけ押す必要がありましたが、トレース機能を使うと[CON]キーを押さなくても、レジスタダンプをUSBで接続したWindowsパソコンのコマンドプロンプト画面に表示しながら自動的に10ステップ分のステップ動作が行われます。
トレース機能の具体的な操作の仕方については、ND8080リモートプログラム操作説明書を参照してください。

4．メモリダンプ

USBで接続したWindowsパソコンのコマンドプロンプト画面に、指定範囲のメモリ内容を、16進で表示するとともに、JIS（ASCII）コードとみなして、キャラクタ（文字） でも表示します。
メモリダンプ機能の具体的な操作の仕方については、ND8080リモートプログラム操作説明書を参照してください。

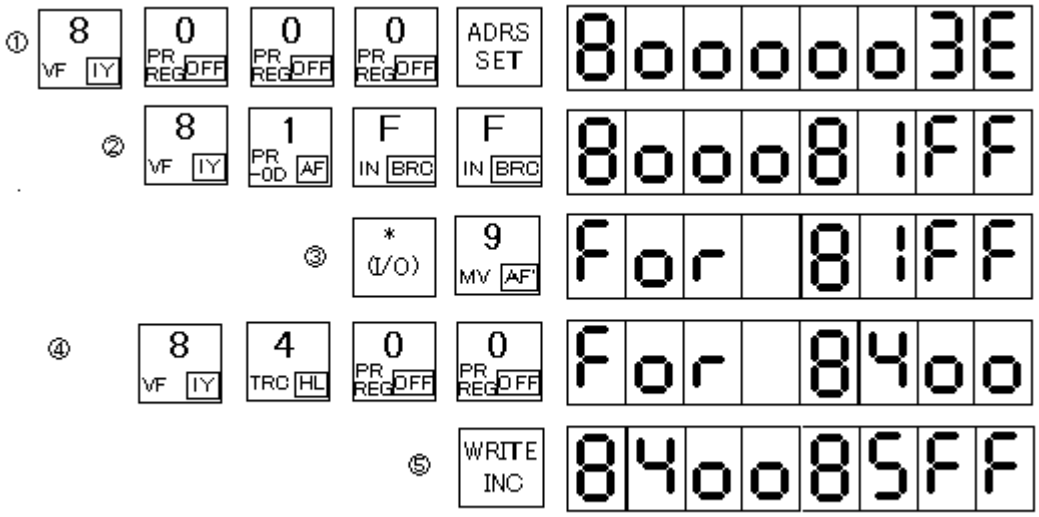
6章　その他の機能

1．MOVE MEMORY

プログラムの変更や追加のために、ある範囲のプログラムを少し移動させたり、ある範囲のデータをそっくりそのまま、別のアドレスに転送したい場合があります。

そんなときにこの機能を使えば、簡単にメモリ内容の転送（COPY）ができます。

例として、8000～81FFのデータを8400からはじまるメモリエリアに転送する仕方を説明します。

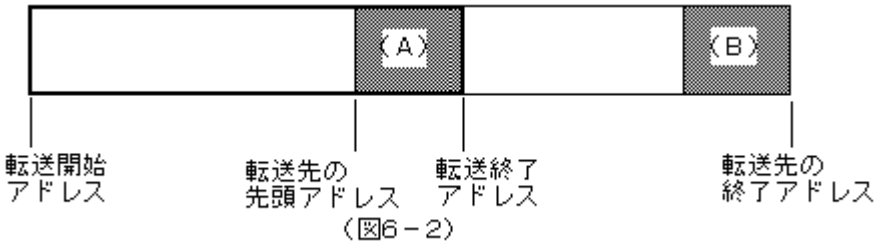


（図6－1）

①転送開始アドレス（ここでは8000）をアドレス表示部に入れます。
②続いて転送終了アドレス（81FF）を入力します（[WRITE　INC]キーを押してしまわないように注意して下さい）。
③[＊（I／O）]キーに続いてデータキーの[9]（[MV]）を押します。するとアドレス表示部にForの表示が出ます（このときデータ表示部は、変化しません。転送終了アドレスを表示したままです）。
④今度は転送先の先頭アドレスを入力します。
⑤[WRITE　INC]キーを押すと、瞬時に転送が完了して、転送後の先頭アドレスと終わりのアドレスを表示します。

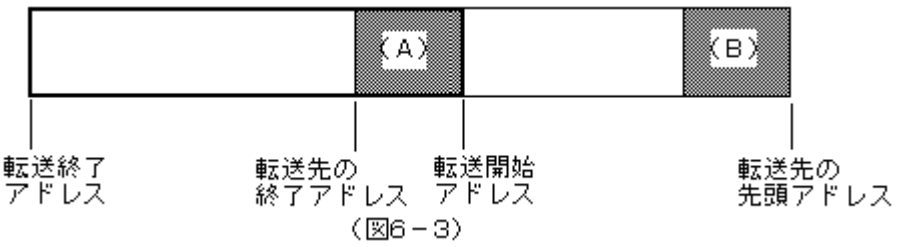
●転送元のメモリ範囲と転送先のメモリ範囲が重なっていても転送は正しく行われます。

普通は小さいアドレスから大きいアドレスの順に転送しますが、図6－2のように転送元と転送先のアドレスに重なりがあると、転送によって重なった部分のデータが壊されてしまうため、正しく転送が行われません。
転送によって（A）が先に壊されてしまうので（B）の部分には壊されたデータが転送されてしまいます。



（図6－2）

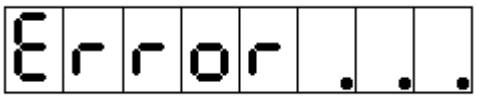
そこでこの転送プログラムは、与えられたアドレスをチェックして、上図のような重なりがある場合には、逆に大きいアドレスから小さいアドレスの順に転送するように考えてあります。



（図6－3）

図6－3 のようにすると、A部分は転送の終わりの方で壊されますがそのときにはBへの転送は完了しているため、正しく転送されます。

〔注意1〕転送部分の重なりの判断およびその処理は全てプログラムが行います。従ってアドレスの入力については必ず、転送開始アドレス＜転送終了アドレスという条件で入れて下さい。例えば開始アドレス＝81FF、終了アドレス＝8000と指定すると7セグメントLEDにエラーの表示が出されます。



　転送先の先頭アドレスについては制限はありません。例えば開始アドレス＝8400終了アドレス＝85FFのとき、転送先の先頭アドレス＝8000であっても問題はありません。
〔注意2〕メモリのF800～FFFFの部分はモニタプログラムが使用しています。もし転送先のメモリ範囲に、このF800～FFFFの一部でも含むような指定をすると、モニタプログラムが暴走することがあります。
〔注意3〕この転送機能は転送先でのチェックは行いません。転送先として指定した範囲がROMアドレス（0000～7FFF）の場合、転送は結果的に無意味ですが、特にエラー表示は出ません（転送元がROMのアドレスであることは支障ありません）。
〔注意4〕この転送機能はデータをそのままのかたちで移動するものです。もし移動の対象がプログラムである場合、転送元ではうまく動作していても転送先では正しく動作しなくなる可能性があります（特にジャンプやCALL命令に注意して下さい）。

```
8000  3E00    MVI A,00
8002  3C      INR A
8003  C30280  JMP $8002
```

このプログラムを8150に転送した場合

```
8150  3E00
8152  3C
8153  C30280  →C35281  に直さなければ、このアドレスでは正しく動作しません。
```

7章　モニタサブルーチン

ND80Zモニタプログラムには、幾つかのサブルーチンが含まれており、この中にはユーザーが利用すると便利なものもあります。

なお、ND80Zモニタプログラムはキー入力やLED表示などの基本的な処理については0400～07FFにある、TK80モニタプログラムのサブルーチンをCALLしています。

0400～07FFにあるサブルーチンは、0000スタートのTK80モニタプログラムのサブルーチンと同じ機能です（ワークエリアだけが83xxではなくてFFxxになっています）。

0400～07FFにあるサブルーチンのエントリアドレスは、0000スタートのTK80モニタプログラムのサブルーチンのエントリアドレスに0400を加算したアドレスになります。

たとえばキー入力ルーチン①のエントリアドレスは0000スタートのTK80モニタプログラムでは0216ですが、0400スタートのTK80モニタプログラムでは0616です。

これらのサブルーチンはエントリアドレスとワークエリアのアドレスが異なるだけですから、ここでの説明は省略します（「TK80モニタプログラム操作説明書」を参照してください）。

8章　モニタプログラムリスト

　ND80Zモニタプログラムは、0400番地台のTK80モニタプログラム内のサブルーチンをCALLしています。0400番地台のTK80モニタプログラム内のサブルーチンは、0000番地台のTK80モニタプログラム内のサブルーチンアドレスに0400を加算したアドレスになっていて、機能もワークアドレスが83xxの代わりにFFxxになっているだけですが、ごく一部のみ、ND80Zモニタの機能のために変更しています。

```
2016/4/22  11:56  n8mon4t.txt
END=07D7
                ;;; TK80 MONITOR PROGRAM FOR ND80Z(WORK AREAR FFxx)
                ; 09/5/28 09/6/1 6/3 6/5
                ; 10/2/25 3/5 3/6 3/19 3/22 3/23
                ; 10/5/5 SERIAL IN/OUT
                ;10/5/9 PIC INTERFACE 4BITS
                ;5/31SAVE/LOAD ENTRY TO ND3MON FROM NDMON4_G
                ;NDMON4J for ND80Z3 10/6/15 6/16
                ;10/8/21 NDMON4M
                ;8/22 NDMON4N
                ;8/23 NDMON4O
                ;8/24 NDMON4P
                ;
                ;16/1/6 KEYIN D3="0"
                ;4/2 n8mon4r
                ;4/15 D2 for RND
                ;4/22 D2
                ;
                ; FFFF-FFF8 7SEGMENT DISPLAY ADDRESS
                ;
                DIG=$FFF8
                ;
                DISP=$FFF4
                KFLAG=$FFF3
                BRKCT=$FFF2
                BRKAD=$FFF0
                ADRES1=$FFEF
                ADRES=$FFEE
                DATA1=$FFED
                DATA=$FFEC
                FSAVE=$FFEA
                BSAVE=$FFE9
                CSAVE=$FFE8
                DSAVE=$FFE7
                ESAVE=$FFE6
                HSAVE=$FFE5
                LSAVE=$FFE4
                SSAVE=$FFE2
                PSAVE=$FFE0
                ;
                ;FFDF-FFCF are used by ND80Z MONITOR
                ;
                R=$FFD2
                ;
                RST7=$FFCC
                RST6=$FFC9
                RST5=$FFC6
                RST4=$FFC3
                RST3=$FFC0
                RST2=$FFBD
```

```
                RST1=$FFBA
                ;REMOTEWK=$FFB9
                SINERMK=$FFB8
                ;
                MONSP=$FFB8
                USRSP=$F800
                ;
                NDZENT1=$080F
                NDZENT2=$081B
                ;
                ;
                ORG $0400
                ;
0400 C33B04     JMP MONST
                ;
                ;       ORG $0008
                ;       JMP START
                ;
                ;       ORG $0010
                ;       JMP RST2
                ;
                ;       ORG $0018
                ;       JMP RST3
                ;
                ;       ORG $0020
                ;       JMP RST4
                ;
                ;       ORG $0028
                ;       JMP RST5
                ;
                ;       ORG $0030
                ;       JMP RST6
                ;
                ;       ORG $0038
                ;       JMP RST7
                ;
                ; INITIALIZE ROUTINE
                ;
                ORG $043B
                ;
043B 3EFF       MONST:MVI A,FF
043D D398       OUT 98
043F 21ECFF     LXI H,DATA
0442 060C       MVI B,0C
0444 AF         XRA A
0445 77         MONST2:MOV M,A
0446 23         INX H
0447 05         DCR B
0448 C24504     JNZ MONST2
044B 2100F8     LXI H,USRSP
044E 22E2FF     SHLD SSAVE
                ;
                ; MONITOR START
                ;
0451 3EFF       START:MVI A,FF
0453 D398       OUT 98
0455 31B8FF     LXI SP,MONSP
0458 CDC005     CALL SEGCG
```

```
045B CD1606   START2:CALL KEYIN
045E 47       MOV B,A
045F E610     ANI 10
0461 CA8404   JZ DIGIT
0464 78       MOV A,B
0465 E60F     ANI 0F
0467 0600     MVI B,00
0469 87       ADD A
046A 4F       MOV C,A
046B 217404   LXI H,TABL
046E 09       DAD B
046F 7E       MOV A,M
0470 23       INX H
0471 66       MOV H,M
0472 6F       MOV L,A
0473 E9       PCHL
              ;
0474 CC04     TABL:DW GOTO
0476 F905     DW RESRG
0478 9404     DW ADSET
047A B804     DW ADDCX
047C 9D04     DW ADINX
047E C204     DW MEMW
0480 D504     DW SDATA
0482 0705     DW LDATA
              ;
0484 CDB505   DIGIT:CALL SHIFT
0487 3AECFF   LDA DATA
048A B0       ORA B
048B 32ECFF   STA DATA
048E CDA105   CALL RGDSP
0491 C35104   JMP START
              ;
              ; ADDRESS SET
              ;
0494 2AECFF   ADSET:LHLD DATA
0497 22EEFF   SHLD ADRES
049A C3A104   JMP ADINX2
              ;
              ; MEMORY READ & ADDRESS INCREMENT
              ;
049D 2AEEFF   ADINX:LHLD ADRES
04A0 23       INX H
04A1 CDAD04   ADINX2:CALL MEMR
04A4 22EEFF   ADSTR:SHLD ADRES
04A7 CDA105   CALL RGDSP
04AA C35104   JMP START
              ;
04AD 3AECFF   MEMR:LDA DATA
04B0 32EDFF   STA DATA1
04B3 7E       MOV A,M
04B4 32ECFF   STA DATA
04B7 C9       RET
              ;
              ; MEMORY READ & ADDRESS DECREMENT
              ;
04B8 2AEEFF   ADDCX:LHLD ADRES
04BB 2B       DCX H
```

```
04BC CDAD04    CALL MEMR
04BF C3A404    JMP ADSTR
               ;
               ; MEMORY WRITE
               ;
04C2 2AEEFF    MEMW:LHLD ADRES
04C5 3AECFF    LDA DATA
04C8 77        MOV M,A
04C9 C39D04    JMP ADINX
               ;
               ; MONITOR TO USER CONTROL ROUTINE
               ;
04CC 2AEEFF    GOTO:LHLD ADRES
04CF 22E0FF    SHLD PSAVE
04D2 C3F905    JMP RESRG
               ;
               ; STORE DATA
               ;
04D5 06F7      SDATA:MVI B,F7
04D7 2AECFF    SDATA1:LHLD DATA
04DA EB        XCHG
04DB 2AEEFF    LHLD ADRES
04DE 78        MOV A,B
04DF D398      OUT 98;SET 94=OUT
04E1 7C        MOV A,H
04E2 CD7C06    CALL SOUT
04E5 7D        MOV A,L
04E6 CD7C06    CALL SOUT
04E9 7A        MOV A,D
04EA CD7C06    CALL SOUT
04ED 7B        MOV A,E
04EE CD7C06    CALL SOUT
04F1 2B        DCX H
04F2 23        SDATA2:INX H
04F3 7E        MOV A,M
04F4 CD7C06    CALL SOUT
04F7 CD0107    CALL HDCMP
04FA C2F204    JNZ SDATA2
04FD C30F08    JMP NDZENT1
0500 00        NOP
0501 00        NOP
0502 00        NOP
0503 00        NOP
0504 00        NOP
0505 00        NOP
0506 00        NOP
               ;
               ;LOAD DATA
               ;
0507 06FF      LDATA:MVI B,FF
0509 78        LDATA1:MOV A,B;need this!
050A D398      OUT 98;need this!
050C CDA006    CALL SIN
050F 67        MOV H,A
0510 CDA006    CALL SIN
0513 6F        MOV L,A
0514 CDA006    CALL SIN
0517 57        MOV D,A
```

```
0518 CDA006    CALL SIN
051B 5F        MOV E,A
051C 22EEFF    SHLD ADRES
051F EB        XCHG
0520 22ECFF    SHLD DATA
0523 EB        XCHG
0524 2B        DCX H
0525 23        LDATA2:INX H
0526 CDA006    CALL SIN
0529 77        MOV M,A
052A CD0107    CALL HDCMP
052D C22505    JNZ LDATA2
0530 CDA105    CALL RGDSP
0533 C30F08    JMP NDZENT1
               ;
               ; BREAK ENTRY
               ; BREAK & ONE STEP OPERATION
               ;
               ORG $0551
               ;
0551 E3        BRENT:XTHL
0552 22E0FF    SHLD PSAVE
0555 F5        PUSH PSW
0556 210400    LXI H,$0004
0559 39        DAD SP
055A F1        POP PSW
055B 22E2FF    SHLD SSAVE
055E E1        POP H
055F 31ECFF    LXI SP,DATA
0562 F5        PUSH PSW
0563 C5        PUSH B
0564 D5        PUSH D
0565 E5        PUSH H
0566 31B8FF    LXI SP,MONSP
0569 3AF2FF    LDA BRKCT
056C A7        ANA A
056D CA8B05    JZ BSTOP
0570 2AF0FF    LHLD BRKAD
0573 EB        XCHG
0574 2AE0FF    LHLD PSAVE
0577 7D        MOV A,L
0578 BB        CMP E
0579 C28505    JNZ NOBRK
057C 7C        MOV A,H
057D BA        CMP D
057E C28505    JNZ NOBRK
0581 21F2FF    LXI H,BRKCT
0584 35        DCR M
0585 CD9105    NOBRK:CALL ADDSP
0588 C3F905    JMP RESRG
058B CD9105    BSTOP:CALL ADDSP
058E C35104    JMP START
0591 2AEAFF    ADDSP:LHLD FSAVE
0594 22ECFF    SHLD DATA
0597 2AE0FF    LHLD PSAVE
059A 22EEFF    SHLD ADRES
059D CDA105    CALL RGDSP
05A0 C9        RET
```

```
                ;
                ;
                ;;; SUBROUTINE
                ;
05A1 21EFFF     RGDSP:LXI H,ADRES1
05A4 11F4FF     LXI D,DISP
05A7 0604       MVI B,04
05A9 7E         RGDSP2:MOV A,M
05AA 12         STAX D
05AB 2B         DCX H
05AC 13         INX D
05AD 05         DCR B
05AE C2A905     JNZ RGDSP2
05B1 CDC005     CALL SEGCG
05B4 C9         RET
                ;
                ;DATA REG SHIFT(4 BITS)
                ;
05B5 2AECFF     SHIFT:LHLD DATA
05B8 29         DAD H
05B9 29         DAD H
05BA 29         DAD H
05BB 29         DAD H
05BC 22ECFF     SHLD DATA
05BF C9         RET
                ;
                ; SEGMENT CONVERT SUB
                ;
05C0 21F4FF     SEGCG:LXI H,DISP
05C3 11F8FF     LXI D,DIG
05C6 01E905     LXI B,SEGD
05C9 7E         SEGCG2:MOV A,M
05CA 23         INX H
05CB E5         PUSH H
05CC F5         PUSH PSW
05CD E6F0       ANI F0
05CF 0F         RRC
05D0 0F         RRC
05D1 0F         RRC
05D2 0F         RRC
05D3 2600       MVI H,00
05D5 6F         MOV L,A
05D6 09         DAD B
05D7 7E         MOV A,M
05D8 12         STAX D
05D9 13         INX D
05DA F1         POP PSW
05DB E60F       ANI 0F
05DD 2600       MVI H,00
05DF 6F         MOV L,A
05E0 09         DAD B
05E1 7E         MOV A,M
05E2 12         STAX D
05E3 E1         POP H
05E4 1C         INR E
05E5 C2C905     JNZ SEGCG2
05E8 C9         RET
                ;
```

```
                ; SEGMENT DATA
                ;
05E9 5C         SEGD:DB 5C
05EA 06         DB 06
05EB 5B         DB 5B
05EC 4F         DB 4F
05ED 66         DB 66
05EE 6D         DB 6D
05EF 7D         DB 7D
05F0 27         DB 27
05F1 7F         DB 7F
05F2 6F         DB 6F
05F3 77         DB 77
05F4 7C         DB 7C
05F5 39         DB 39
05F6 5E         DB 5E
05F7 79         DB 79
05F8 71         DB 71
                ;
                ; REGISTER RESTORE
                ;
05F9 2AE2FF     RESRG:LHLD SSAVE
05FC F9         SPHL
05FD 2AE0FF     LHLD PSAVE
0600 E5         PUSH H
0601 2AE4FF     LHLD LSAVE
0604 E5         PUSH H
0605 2AEAFF     LHLD FSAVE
0608 E5         PUSH H
0609 2AE8FF     LHLD CSAVE
060C 4D         MOV C,L
060D 44         MOV B,H
060E 2AE6FF     LHLD ESAVE
0611 EB         XCHG
0612 F1         POP PSW
0613 E1         POP H
0614 FB         EI
0615 C9         RET
                ;
                ; KEY INPUT
                ;
0616 CD2306     KEYIN:CALL INPUT
0619 47         MOV B,A
061A 3AF3FF     LDA KFLAG
061D A7         ANA A
061E CA1606     JZ KEYIN
0621 78         MOV A,B
0622 C9         RET
                ;
                ; KEY INPUT SUB
                ;
0623 CD4706     INPUT:CALL KEY
0626 3C         INR A
0627 CA4206     JZ NOKEY
062A CDEA06     INPUT2:CALL D2
062D CD4706     CALL KEY
0630 47         MOV B,A
0631 3C         INR A
```

```
0632 CA4206    JZ NOKEY
0635 3AF3FF    LDA KFLAG
0638 A7        ANA A
0639 C22A06    JNZ INPUT2
063C 3D        DCR A
063D 32F3FF    INPUT3:STA KFLAG
0640 78        MOV A,B
0641 C9        RET
0642 06FF      NOKEY:MVI B,FF
0644 C33D06    JMP INPUT3
               ;
               ; KEY SCAN & CONVERT HEX DATA SUB
               ;
0647 1600      KEY:MVI D,00
0649 42        MOV B,D
064A 3EF6      MVI A,F6
064C D39C      OUT 9C
064E DB9C      IN 9C
0650 EEFF      XRI FF
0652 C27106    JNZ KEYI
0655 0608      MVI B,08
0657 3EF5      MVI A,F5
0659 D39C      OUT 9C
065B DB9C      IN 9C
065D EEFF      XRI FF
065F C27106    JNZ KEYI
0662 0610      MVI B,10
0664 3EF3      MVI A,F3
0666 D39C      OUT 9C
0668 DB9C      IN 9C
066A EEFF      XRI FF
066C C27106    JNZ KEYI
066F 3D        DCR A
0670 C9        RET
0671 0F        KEYI:RRC
0672 DA7906    JC KEYI2
0675 14        INR D
0676 C37106    JMP KEYI
0679 7A        KEYI2:MOV A,D
067A B0        ORA B
067B C9        RET
               ;
               ;SERIAL OUTPUT ROUTINE
               ;
067C 4F        SOUT:MOV C,A
067D DB94      SOUT2:IN 94
067F E640      ANI 40
0681 CA7D06    JZ SOUT2
0684 CD2607    CALL SOUTSB
0687 78        MOV A,B; I/Oaddress 94 "out" set & STROBE ON
0688 E6FD      ANI FD;bit1=0
068A D398      OUT 98
068C DB94      SOUT3:IN 94
068E E640      ANI 40
0690 C28C06    JNZ SOUT3
0693 78        MOV A,B; I/Oaddress 94 "out" set & STROBE OFF
0694 D398      OUT 98
0696 C9        RET
```

```
                ;
                ;SERIAL INPUT ROUTINE
                ;
                ORG $06A0
                ;
06A0 78         SIN:MOV A,B
06A1 D398       OUT 98
06A3 CD8D07     SIN2:CALL SINSB
06A6 CAA306     JZ SIN2
06A9 79         MOV A,C
06AA C9         RET
                ;
                ;CHATTERING TIMER
                ;
                ;D1=45112ms
                ;D2=9.0176ms
                ;D3=27.0176ms
                ;
                ORG $06DD
06DD 1624       D1:MVI D,24;=36 ck=7 (7+266*36+10)/2.048=9593/2.048=4684.08microsec
06DF 1E10       D1_2:MVI E,10;=16 ck=7  7+15*16+15=266
06E1 1D         D1_3:DCR E;       ck=5
06E2 C2E106     JNZ D1_3; ck=10
06E5 15         DCR D;    ck=5
06E6 C2DF06     JNZ D1_2; ck=10
06E9 C9         RET;      ck=10
06EA 1648       D2:MVI D,48;=72  (7+266*72+10+10)/2.048=19179/2.048=9364.75microsec
06EC C3CF07     JMP D2_2
06EF 16D8       D3:MVI D,D8;=216 (7+266*216+20)/2.048=57483/2.048=28067.87microsec
06F1 C3DF06     JMP D1_2
                ;
06F4 22ECFF     HLDTDP:SHLD DATA
06F7 C5         PSHRGDP:PUSH B
06F8 D5         PUSH D
06F9 E5         PUSH H
06FA CDA105     CALL RGDSP
06FD E1         POP H
06FE D1         POP D
06FF C1         POP B
0700 C9         RET
                ;
0701 7D         HDCMP:MOV A,L
0702 BB         CMP E
0703 C0         RNZ
0704 7C         MOV A,H
0705 BA         CMP D
0706 C9         RET
                ;
0707 32B8FF     LDERR:STA SINERMK
070A 211E07     LXI H,ERRT
070D CD1307     CALL SEGDP
0710 C31B08     JMP NDZENT2
                ;
0713 11F8FF     SEGDP:LXI D,DIG
0716 7E         SEGDP2:MOV A,M
0717 12         STAX D
0718 23         INX H
0719 1C         INR E
```

```
071A C21607    JNZ SEGDP2
071D C9        RET
               ;
071E 79        ERRT:DB 79;E
071F 50        DB 50;r
0720 50        DB 50;r
0721 5C        DB 5C;o
0722 50        DB 50;r
0723 80        DB 80
0724 80        DB 80
0725 80        DB 80
               ;
0726 79        SOUTSB:MOV A,C
0727 D394      OUT 94
0729 78        MOV A,B
072A E6FD      ANI FD
072C D398      OUT 98
072E DB94      SOUTSB2:IN 94
0730 E640      ANI 40
0732 C22E07    JNZ SOUTSB2
0735 78        MOV A,B
0736 D398      OUT 98
0738 DB94      SOUTSB3:IN 94
073A E640      ANI 40
073C CA3807    JZ SOUTSB3
073F 79        MOV A,C
0740 0F        RRC
0741 0F        RRC
0742 0F        RRC
0743 0F        RRC
0744 D394      OUT 94
0746 C9        RET
               ;
               ; RS232C SAVE & LOAD
0747 2AECFF    RSAVE:LHLD DATA
074A EB        XCHG
074B 2AEEFF    LHLD ADRES
074E 06F3      MVI B,F3
0750 2B        DCX H
0751 23        RSAVE2:INX H
0752 7E        MOV A,M
0753 CD7C06    CALL SOUT
0756 3EFF      MVI A,FF
0758 D398      OUT 98
075A 22ECFF    SHLD DATA
075D CDF706    CALL PSHRGDP
0760 CD0107    CALL HDCMP
0763 C25107    JNZ RSAVE2
0766 C30F08    JMP NDZENT1
               ;
0769 2AEEFF    RLOAD:LHLD ADRES
076C 06FB      MVI B,FB
076E 2B        DCX H
076F 23        RLOAD2:INX H
0770 CD7D07    CALL RSIN
0773 77        MOV M,A
0774 22ECFF    SHLD DATA
0777 CDF706    CALL PSHRGDP
```

```
077A C36F07    JMP RLOAD2
               ;
077D 78        RSIN:MOV A,B
077E D398      OUT 98
0780 CD8D07    RSIN1:CALL SINSB
0783 79        MOV A,C
0784 C0        RNZ
0785 FEFF      CPI FF
0787 C20707    JNZ LDERR
078A C38007    JMP RSIN1
               ;
078D DB94      SINSB:IN 94
078F E620      ANI 20
0791 CA8D07    JZ SINSB
0794 78        MOV A,B;BUSY
0795 E6FE      ANI FE;bit0=0
0797 D398      OUT 98
0799 DB94      SINSB2:IN 94
079B E620      ANI 20
079D C29907    JNZ SINSB2
07A0 DB94      IN 94
07A2 E610      ANI 10
07A4 F5        PUSH PSW
07A5 DB94      IN 94
07A7 E60F      ANI 0F
07A9 4F        MOV C,A
07AA 78        MOV A,B;READY
07AB D398      OUT 98
07AD DB94      SINSB3:IN 94
07AF E620      ANI 20
07B1 CAAD07    JZ SINSB3
07B4 78        MOV A,B
07B5 E6FE      ANI FE
07B7 D398      OUT 98
07B9 DB94      SINSB4:IN 94
07BB E620      ANI 20
07BD C2B907    JNZ SINSB4
07C0 DB94      IN 94
07C2 07        RLC
07C3 07        RLC
07C4 07        RLC
07C5 07        RLC
07C6 E6F0      ANI F0
07C8 B1        ORA C
07C9 4F        MOV C,A
07CA 78        MOV A,B
07CB D398      OUT 98
07CD F1        POP PSW
07CE C9        RET
               ;
07CF EB        D2_2:XCHG
07D0 21D2FF    LXI H,R
07D3 34        INR M
07D4 EB        XCHG
07D5 C3DF06    JMP D1_2
               ;END
ADDCX        =04B8  ADDSP        =0591  ADINX        =049D
ADINX2       =04A1  ADRES        =FFEE  ADRES1       =FFEF
```

```
ADSET       =0494  ADSTR       =04A4  BRENT       =0551
BRKAD       =FFF0  BRKCT       =FFF2  BSAVE       =FFE9
BSTOP       =058B  CSAVE       =FFE8  D1          =06DD
D1_2        =06DF  D1_3        =06E1  D2          =06EA
D2_2        =07CF  D3          =06EF  DATA        =FFEC
DATA1       =FFED  DIG         =FFF8  DIGIT       =0484
DISP        =FFF4  DSAVE       =FFE7  ERRT        =071E
ESAVE       =FFE6  FSAVE       =FFEA  GOTO        =04CC
HDCMP       =0701  HLDTDP      =06F4  HSAVE       =FFE5
INPUT       =0623  INPUT2      =062A  INPUT3      =063D
KEY         =0647  KEYI        =0671  KEYI2       =0679
KEYIN       =0616  KFLAG       =FFF3  LDATA       =0507
LDATA1      =0509  LDATA2      =0525  LDERR       =0707
LSAVE       =FFE4  MEMR        =04AD  MEMW        =04C2
MONSP       =FFB8  MONST       =043B  MONST2      =0445
NDZENT1     =080F  NDZENT2     =081B  NOBRK       =0585
NOKEY       =0642  PSAVE       =FFE0  PSHRGDP     =06F7
R           =FFD2  RESRG       =05F9  RGDSP       =05A1
RGDSP2      =05A9  RLOAD       =0769  RLOAD2      =076F
RSAVE       =0747  RSAVE2      =0751  RSIN        =077D
RSIN1       =0780  RST1        =FFBA  RST2        =FFBD
RST3        =FFC0  RST4        =FFC3  RST5        =FFC6
RST6        =FFC9  RST7        =FFCC  SDATA       =04D5
SDATA1      =04D7  SDATA2      =04F2  SEGCG       =05C0
SEGCG2      =05C9  SEGD        =05E9  SEGDP       =0713
SEGDP2      =0716  SHIFT       =05B5  SIN         =06A0
SIN2        =06A3  SINERMK     =FFB8  SINSB       =078D
SINSB2      =0799  SINSB3      =07AD  SINSB4      =07B9
SOUT        =067C  SOUT2       =067D  SOUT3       =068C
SOUTSB      =0726  SOUTSB2     =072E  SOUTSB3     =0738
SSAVE       =FFE2  START       =0451  START2      =045B
TABL        =0474  USRSP       =F800

2016/4/25  20:40  n80mon2d.txt
END=0E7C
```

```
                ;;; ND80Z MONITOR FOR ND80Z3
                ;;; ND3MON
                ;2010/02/24 2/25 2/26 3/2 3/5 3/6 3/15 3/22 3/23
                ;;;5/5
                ;; 5/24 for remote
                ;5/25 5/26 5/27 5/29 5/30 5/31 6/10 6/13 6/14
                ;6/15 for ND80Z3 6/16 6/17 8/20
                ;10/8/21 ND3MON2M
                ;9/5 nd3mon2n
                ;9/9 nd3mon2o
                ;16/1/5 for 8080 n80mon2a
                ;1/9
                ;4/25 SBCHLDE printer
                ;
                      LSEG8=$FFFF
                      LSEG7=$FFFE
                      LSEG5=$FFFC
                      LSEG4=$FFFB
                      LSEG3=$FFFA
                      LSEG2=$FFF9
                      LSEG1=$FFF8
                      DISP3=$FFF6
                      DISP1=$FFF4
```

```
      ;KFLAG=$FFF3
      BRKC=$FFF2
      BRKA=$FFF0
      ADRSH=$FFEF
      ADRSL=$FFEE
      DATAH=$FFED
      DATAL=$FFEC
      AREG=$FFEB
      FREG=$FFEA
      CREG=$FFE8
      EREG=$FFE6
      LREG=$FFE4
      SPL=$FFE2
      PCL=$FFE0
HLWK=$FFD8
BCWK=$FFD6
      ;LDSH=$FFD4
      ;IREG=$FFD3
IOWK=$FFD2;-FFD4
      TRSW=$FFD1
      IOREG=$FFD0
RMODE=$FFCF
      RST7JA=$FFCD
      RST7JC=$FFCC
RST6JA=$FFCA
RST6JC=$FFC9
RST5JA=$FFC7
RST5JC=$FFC6
RST4JA=$FFC4
RST4JC=$FFC3
RST3JA=$FFC1
RST3JC=$FFC0
RST2JA=$FFBE
RST2JC=$FFBD
RST1JA=$FFBB
RST1JC=$FFBA
REMOTEMK=$FFB9
SINERMK=$FFB8
;
      SPTOP=$FFB8
PRCNT=$FF50
PRBF=$FF00
PKETA=$F0B5
      USTOP=$F800
;
TKRST6=$83DD
TKRST5=$83DA
TKRST4=$83D7
TKRST3=$83D4
TKRST2=$83D1
TKRST1=$0051
;
TKMON=$003B
TKBREAK=$0151
D2=$02EA
;
TK2MON=$043B
TK2RST1=$0451
```

```
                TK2BREAK=$0551
                ;
                LEDDPA=$05C0;SEGCG
                LEDDP=$FFC9;=RST6JC
                RGDSP=$05A1
                SEGCG1=$05C6
                KEYINA=$0616
                KEYIN=$FFC6;=RST5JC
                INPUT=$0623
                KEY=$0647
                ;
                LOAD=$0507
                SOUTSB=$067C
                SINSB=$06A0
                SOUT=$0747
                SIN=$0769
                ;
                ;     ND80Z MONITOR
                ;
                      ORG $0800
                ;
0800 C33608           JMP NDMON
0803 C3120B           JMP SEGDP
0806 C31C0B           JMP LPRTSB
0809 C3290B           JMP LPRTSB2
080C C3710B           JMP CRLF
080F C3A408     JMP START1
0812 C3E80B     JMP PRHX4
0815 C3EC0B     JMP PRHX2
0818 C3F50B     JMP PRHX1
081B C3AE08     JMP START2
081E C3310B     JMP LPRTSB22
0821 C3650D     JMP REMOTE
0824 C3FE0D     JMP LDIR
0827 C30B0E     JMP LDDR
082A C3180E     JMP SBCHLBC
082D C3330E     JMP SOUTSB2;16/3/14
0830 C3380E     JMP SBCHLDE
0833 C3530E     JMP PRTJ
                ;
0836 3EFF       NDMON:MVI A,FF
                ;?      LD I,A
0838 D398       OUT 98
083A 31B8FF     LXI SP,SPTOP
083D AF         XRA A
083E 32B9FF     STA REMOTEMK
0841 0607       MVI B,07
0843 21CCFF     LXI H,RST7JC
0846 3EC3       MVI A,C3
0848 77         NDMON2:MOV M,A
0849 2B         DCX H
084A 2B         DCX H
084B 2B         DCX H
                ;?      DJNZ NDMON2
084C 05         DCR B
084D C24808     JNZ NDMON2
0850 DB94       IN 94
0852 E680       ANI 80
```

```
0854 CA7A08    JZ NDMON3;No.1=ON
               ;TK80 MONITOR
0857 215101    LXI H,TKBREAK
085A 22CDFF    SHLD RST7JA
085D 215100    LXI H,TKRST1
0860 22BBFF    SHLD RST1JA
0863 11DD83    LXI D,TKRST6
0866 21CAFF    LXI H,RST6JA
0869 0605      MVI B,05
086B 36DE      NDMON2_2:MVI M,DE
086D 2B        DCX H
086E 2B        DCX H
086F 2B        DCX H
0870 1B        DCX D
0871 1B        DCX D
0872 1B        DCX D
               ;?      DJNZ NDMON2_2
0873 05        DCR B
0874 C26B08    JNZ NDMON2_2
0877 C33B00    JMP TKMON
               ;
087A 21C005    NDMON3:LXI H,LEDDPA
087D 22CAFF          SHLD RST6JA
0880 211606    LXI H,KEYINA
0883 22C7FF    SHLD RST5JA
0886 215104    LXI H,TK2RST1
0889 22BBFF    SHLD RST1JA
               ;
088C 21DC09    LXI H,BREAK
088F 22CDFF    SHLD RST7JA
0892 21CFFF    LXI H,RMODE
0895 0629      MVI B,29
0897 AF        XRA A
0898 77        NDMON32:MOV M,A
0899 23        INX H
               ;?      DJNZ FC
089A 05        DCR B
089B C29808    JNZ NDMON32
089E 2100F8    LXI H,USTOP
08A1 22E2FF    SHLD SPL
               ;
08A4 3EFF      START1:MVI A,FF
08A6 D398      OUT 98
08A8 31B8FF    LXI SP,SPTOP
08AB CDC9FF          CALL LEDDP
08AE CDC6FF    START2:CALL KEYIN
08B1 47              MOV B,A
08B2 E610            ANI 10
08B4 CAF408          JZ KDIN
08B7 3ACFFF          LDA RMODE
08BA B7              ORA A
08BB CAD408          JZ KCIN
08BE 21C408          LXI H,JPT1
08C1 C3D708          JMP KCIN2
               ;
               ;;; KEY JUMP TABLE (1)
08C4 AE08      JPT1:DW START2
08C6 AE08            DW START2
```

```
08C8 AE08           DW START2
08CA EA0A           DW RRDEC
08CC DB0A           DW RRINC
08CE F70A           DW RWINC
08D0 AE08           DW START2
08D2 600A           DW RMOD
                ;
                ;
                ; COMMAND KEYIN
08D4 21E408     KCIN:LXI H,JPT2
08D7 78         KCIN2:MOV A,B
08D8 E607           ANI 07
08DA 0600       KCIN3:MVI B,00
08DC 87             ADD A
08DD 4F             MOV C,A
08DE 09             DAD B
08DF 7E             MOV A,M
08E0 23             INX H
08E1 66             MOV H,M
08E2 6F             MOV L,A
08E3 E9             PCHL
                ;
                ;;; KEY JUMP TABLE (2)
                ;
08E4 BD09       JPT2:DW RUN
08E6 C909           DW CONT
08E8 8B09           DW ADST
08EA AC09           DW RDEC
08EC 9109           DW RINC
08EE B309           DW WINC
08F0 8A0A           DW IOMOD
08F2 600A           DW RMOD
                    ;;;
                ;
                ;0-F KEY INPUT
08F4 3AD0FF     KDIN:LDA IOREG
08F7 B7             ORA A
08F8 CA7B09         JZ KINDP
08FB F22B09         JP RGDIN; =01
                ; =80 or 81
08FE AF             XRA A
08FF 32D0FF         STA IOREG
0902 210B09         LXI H,JPT3
0905 78             MOV A,B
0906 E60F           ANI 0F
0908 C3DA08         JMP KCIN3
                ;
                ;;; KEY JUMP TABLE (3)
                ;
090B 560A       JPT3:DW RPRT
090D AE08           DW START2;LPRT
090F 4707           DW SOUT
0911 6907           DW SIN
0913 510A           DW TRON
0915 AE08           DW START2;W256
0917 AE08           DW START2;R256
0919 AE08           DW START2;START2
091B 650D           DW REMOTE
```

```
091D 510C           DW MMEM
091F F80C           DW DMEM
0921 AE08           DW START2;DAS
0923 CD0C           DW LSAVE
0925 0705           DW LOAD
0927 C50A           DW OUT
0929 920A           DW IN
                ;;;
                ;
                ;0-F KEY INPUT (REG MODE)
092B AF         RGDIN:XRA A
092C 32D0FF         STA IOREG
092F 32F8FF         STA LSEG1
0932 32FBFF         STA LSEG4
0935 78             MOV A,B
0936 87             ADD A
0937 CA7509         JZ RGMDE
093A 32CFFF     RGDI2:STA RMODE
093D 4F             MOV C,A
093E 0600           MVI B,00
0940 21D60D         LXI H,RSGDT
0943 09             DAD B
0944 5E             MOV E,M
0945 23             INX H
0946 56             MOV D,M
0947 EB             XCHG
0948 22F9FF         SHLD LSEG2
094B 21ECFF         LXI H,DATAL
                ;?      SBC HL,BC
094E CD180E     CALL SBCHLBC
0951 FE1C           CPI 1C
0953 DA6809         JC RGDI3
0956 21F0FF         LXI H,BRKA
0959 CA6809         JZ RGDI3
095C 21F2FF         LXI H,BRKC
095F 22EEFF     SHLD ADRSL
0962 6E         MOV L,M
0963 2600       MVI H,00
0965 C36F09     JMP RGDI4
0968 22EEFF     RGDI3:SHLD ADRSL
096B 5E             MOV E,M
096C 23             INX H
096D 56             MOV D,M
096E EB             XCHG
096F 22ECFF     RGDI4:SHLD DATAL
0972 C3A50A         JMP IN1
                ;
                ; REG MODE END
0975 32CFFF     RGMDE:STA RMODE
0978 C39B09         JMP RINC3
                ;
                ;KEY INPUT DATA DISP
097B CD460A     KINDP:CALL DSFTL
097E 3AECFF         LDA DATAL
0981 B0             ORA B
0982 32ECFF     KIDP1:STA DATAL
0985 CD720A     KIDP2:CALL DRDSP
0988 C3AE08         JMP START2;cannot to IN2!!! 10/5/27
```

```
                ;
                ; ADDRESS SET
098B 2AECFF     ADST:LHLD DATAL
098E C39509           JMP RINC2
                ;
                ; READ INC
0991 2AEEFF     RINC:LHLD ADRSL
0994 23               INX H
0995 CDA109    RINC2:CALL DTSET
0998 22EEFF           SHLD ADRSL
099B CDC20D    RINC3:CALL ADDSP
099E C3AE08           JMP START2
                ;
09A1 3AECFF    DTSET:LDA DATAL
09A4 32EDFF           STA DATAH
09A7 7E               MOV A,M
09A8 32ECFF           STA DATAL
09AB C9               RET
                ;
                ; READ DEC
09AC 2AEEFF     RDEC:LHLD ADRSL
09AF 2B               DCX H
09B0 C39509           JMP RINC2
                ;
                ; WRITE INC
09B3 2AEEFF     WINC:LHLD ADRSL
09B6 3AECFF           LDA DATAL
09B9 77               MOV M,A
09BA C39109           JMP RINC
                ;
                ; RUN
09BD 2AEEFF       RUN:LHLD ADRSL
09C0 22E0FF           SHLD PCL
09C3 7C        MOV A,H
09C4 B5        ORA L
09C5 F3        DI
09C6 CA0000    JZ $0000
                ;
                ; CONTINUE
09C9 31E6FF     CONT:LXI SP,EREG
09CC D1               POP D
09CD C1               POP B
09CE F1               POP PSW
                ;?       LD SP,(SPL)
09CF 2AE2FF    LHLD SPL
09D2 F9        SPHL
09D3 2AE0FF           LHLD PCL
09D6 E5               PUSH H
09D7 2AE4FF           LHLD LREG
09DA FB               EI
09DB C9               RET
                ;
                ; BREAK
09DC 22E4FF    BREAK:SHLD LREG
09DF EB        XCHG
09E0 22E6FF    SHLD EREG
09E3 60        MOV H,B
09E4 69        MOV L,C
```

```
09E5 22E8FF    SHLD CREG
09E8 E1              POP H;=PC
               ;?       LD (EREG),DE
               ;?       LD (CREG),BC
09E9 F5              PUSH PSW
09EA 22E0FF    SHLD PCL
09ED 22EEFF          SHLD ADRSL
09F0 210200    LXI H,$0002
09F3 39        DAD SP
09F4 22E2FF    SHLD SPL
09F7 E1              POP H;=AF
09F8 22EAFF          SHLD FREG
09FB 22ECFF          SHLD DATAL
               ;?       LD (SPL),SP
               ;
09FE 31B8FF    LXI SP,SPTOP
0A01 3AB9FF    LDA REMOTEMK
0A04 B7        ORA A
0A05 CCC20D    CZ ADDSP
0A08 3AF2FF    LDA BRKC
0A0B B7        ORA A
0A0C CA310A    JZ BRKON
0A0F 2AF0FF    LHLD BRKA
0A12 EB        XCHG
0A13 2AE0FF    LHLD PCL
               ;?     LD DE,(BRKA)
0A16 B7        ORA A
               ;?     SBC HL,DE
0A17 CD380E    CALL SBCHLDE
0A1A C2210A    JNZ NOBRK
0A1D 3D        DCR A
0A1E 32F2FF    STA BRKC
0A21 CD4706    NOBRK:CALL KEY
0A24 FEFF      CPI FF
0A26 CAC909    JZ CONT
0A29 FE16      CPI 16;*I/O
0A2B C2C909    JNZ CONT
0A2E C3AE08    JMP START2
               ;
0A31 3AB9FF    BRKON:LDA REMOTEMK
0A34 B7        ORA A
0A35 C4A10B    CNZ REGDP
0A38 3AD1FF    LDA TRSW
0A3B B7        ORA A
0A3C CAAE08    JZ START2
0A3F 3D        DCR A
0A40 32D1FF    STA TRSW
0A43 C3C909    JMP CONT
               ;
0A46 2AECFF    DSFTL:LHLD DATAL
0A49 29              DAD H
0A4A 29              DAD H
0A4B 29              DAD H
0A4C 29              DAD H
0A4D 22ECFF          SHLD DATAL
0A50 C9              RET
               ;;;
0A51 3E10      TRON:MVI A,10
```

```
0A53 C3580A          JMP RPRT2
                     ;;;
0A56 3E01      RPRT:MVI A,01
0A58 21D1FF    RPRT2:LXI H,TRSW
0A5B AE        XRA M
0A5C 77        MOV M,A
0A5D C39509          JMP RINC2
                     ;;;
0A60 210201    RMOD:LXI H,$0102
0A63 22CFFF          SHLD RMODE
0A66 215050          LXI H,$5050
0A69 22F8FF          SHLD LSEG1
0A6C 22FAFF          SHLD LSEG3
0A6F C38509          JMP KIDP2
0A72 21EDFF    DRDSP:LXI H,DATAH
0A75 11F6FF          LXI D,DISP3
0A78 D5              PUSH D
0A79 0602            MVI B,02
0A7B 7E             DRDSP2:MOV A,M
0A7C 12              STAX D
0A7D 2B              DCX H
0A7E 13              INX D
                     ;?       DJNZ FA
0A7F 05        DCR B
0A80 C27B0A    JNZ DRDSP2
0A83 E1              POP H
0A84 11FCFF          LXI D,LSEG5
0A87 C3C605          JMP SEGCG1
                     ;
0A8A 3E80      IOMOD:MVI A,80
0A8C 32D0FF          STA IOREG
0A8F C3AE08          JMP START2
                     ;
0A92 3AEEFF    IN:LDA ADRSL
                     ;      MOV C,A
                     ;?       IN A,(C)
0A95 21D2FF    LXI H,IOWK
0A98 36DB      MVI M,DB
0A9A 23        INX H
0A9B 77        MOV M,A
0A9C 23        INX H
0A9D 36C9      MVI M,C9
0A9F CDD2FF    CALL IOWK
0AA2 32ECFF          STA DATAL
0AA5 CD720A    IN1:CALL DRDSP
0AA8 3AB9FF    IN2:LDA REMOTEMK
0AAB B7        ORA A
0AAC CAAE08    JZ START2
0AAF 21EFFF    LXI H,ADRSH
0AB2 11F4FF    LXI D,DISP1
0AB5 0602      MVI B,02
0AB7 7E        IN3:MOV A,M
0AB8 12        STAX D
0AB9 2B        DCX H
0ABA 13        INX D
                     ;?     DJNZ IN3
0ABB 05        DCR B
0ABC C2B70A    JNZ IN3
```

```
0ABF CDA30D    CALL RLEDOUT1
0AC2 C3AE08           JMP START2
               ;;;
0AC5 3AEEFF      OUT:LDA ADRSL
               ;       MOV C,A
0AC8 21D2FF    LXI H,IOWK
0ACB 36D3      MVI M,D3
0ACD 23        INX H
0ACE 77        MOV M,A
0ACF 23        INX H
0AD0 36C9      MVI M,C9
0AD2 3AECFF          LDA DATAL
               ;?       OUT (C),A
0AD5 CDD2FF    CALL IOWK
0AD8 C3A80A          JMP IN2
               ;
               ; READ INC(REG MODE)
0ADB 3ACFFF    RRINC:LDA RMODE
0ADE 3C              INR A
0ADF 3C              INR A
0AE0 FE20            CPI 20
0AE2 DA3A09          JC RGDI2
0AE5 3E02      RRINC2:MVI A,02
0AE7 C33A09          JMP RGDI2
               ;
               ; READ DEC(REG MODE)
0AEA 3ACFFF    RRDEC:LDA RMODE
0AED 3D              DCR A
0AEE 3D              DCR A
0AEF C23A09          JNZ RGDI2
0AF2 3E1E            MVI A,1E
0AF4 C33A09          JMP RGDI2
               ;
               ; WRITE INC(REG MODE)
0AF7 2AECFF    RWINC:LHLD DATAL
0AFA EB        XCHG
0AFB 2AEEFF    LHLD ADRSL
               ;?       LD DE,(DATAL)
0AFE 73              MOV M,E
0AFF 3ACFFF    LDA RMODE
0B02 FE1E      CPI 1E
0B04 CAE50A    JZ RRINC2
0B07 23              INX H
0B08 72              MOV M,D
0B09 C3DB0A          JMP RRINC
               ;
0B0C 7C        HDCMP:MOV A,H
0B0D BA              CMP D
0B0E C0              RNZ
0B0F 7D              MOV A,L
0B10 BB              CMP E
0B11 C9              RET
               ;
0B12 11F8FF    SEGDP:LXI D,LSEG1
0B15 010800          LXI B,$0008
0B18 CDFE0D    CALL LDIR
               ;?       LDIR
0B1B C9              RET
```

```
                ;
0B1C 1A         LPRTSB:LDAX D
0B1D 13         INX D
0B1E F5         PUSH PSW
0B1F CD290B           CALL LPRTSB2
0B22 F1         POP PSW
0B23 FE0D       CPI 0D
0B25 C21C0B           JNZ LPRTSB
0B28 C9         RET
                ;
0B29 FE0D       LPRTSB2:CPI 0D
0B2B CA710B     JZ CRLF
0B2E FE0A       CPI 0A
0B30 C8         RZ
0B31 E5         LPRTSB22:PUSH H
0B32 D5         PUSH D
0B33 2150FF     LXI H,PRCNT
0B36 5E         MOV E,M
0B37 16FF       MVI D,FF
0B39 12         STAX D
0B3A 13         INX D
0B3B 7B         MOV A,E
0B3C FE50       CPI 50
0B3E CA4F0B     JZ LPRTSB3
0B41 77         MOV M,A
0B42 D1         POP D
0B43 E1         POP H
0B44 C9               RET
                ;
0B45 E5         BFOUT:PUSH H
0B46 D5         PUSH D
0B47 3A50FF     LDA PRCNT
0B4A 5F         MOV E,A
0B4B B7         ORA A
0B4C CA6E0B     JZ LPRTSB6
                ;
                ; BUFFER FULL
0B4F 2100FF     LPRTSB3:LXI H,PRBF
0B52 C5         PUSH B
0B53 3EF7       MVI A,F7
0B55 47         MOV B,A
0B56 D398       OUT 98
0B58 7B         MOV A,E
0B59 CD7C06     CALL SOUTSB
0B5C 7E         LPRTSB4:MOV A,M
0B5D CD7C06     CALL SOUTSB
0B60 23         INX H
0B61 1D         DCR E
0B62 C25C0B     JNZ LPRTSB4
0B65 AF         LPRTSB5:XRA A
0B66 3250FF     STA PRCNT
0B69 C1         POP B
0B6A 3EFF       MVI A,FF
0B6C D398       OUT 98
0B6E D1         LPRTSB6:POP D
0B6F E1         POP H
0B70 C9         RET
                ;
```

```
0B71 E5       CRLF:PUSH H
0B72 D5       PUSH D
0B73 C5       PUSH B
0B74 3EF7     MVI A,F7
0B76 47       MOV B,A
0B77 D398     OUT 98
0B79 3A50FF   LDA PRCNT
0B7C B7       ORA A
0B7D F5       PUSH PSW
0B7E 5F       MOV E,A
0B7F 3C       INR A;FOR 0D
0B80 3C       INR A;FOR 0A
0B81 CD7C06   CALL SOUTSB
0B84 F1       POP PSW
0B85 CA940B   JZ CRLF3
0B88 2100FF   LXI H,PRBF
0B8B 7E       CRLF2:MOV A,M
0B8C CD7C06   CALL SOUTSB
0B8F 23       INX H
0B90 1D       DCR E
0B91 C28B0B   JNZ CRLF2
0B94 3E0D     CRLF3:MVI A,0D
0B96 CD7C06   CALL SOUTSB
0B99 3E0A     MVI A,0A
0B9B CD7C06   CALL SOUTSB
0B9E C3650B   JMP LPRTSB5
              ;
0BA1 11030C   REGDP:LXI D,REG_PR_T
0BA4 CD1C0B         CALL LPRTSB
0BA7 060D           MVI B,0D
0BA9 11EBFF         LXI D,AREG
0BAC CDE20B   REGDP2:CALL PRDEHX4
0BAF 3E20           MVI A,20
0BB1 CD290B         CALL LPRTSB2
0BB4 05             DCR B
0BB5 C2AC0B         JNZ REGDP2
0BB8 3AEAFF         LDA FREG
0BBB 4F             MOV C,A
0BBC 0608           MVI B,08
              ;? REGDP3::RL C
0BBE 79       REGDP3:MOV A,C
0BBF 17       RAL
0BC0 4F       MOV C,A
0BC1 DAC90B         JC REGDP4
0BC4 3E30           MVI A,30
0BC6 C3CB0B         JMP REGDP5
0BC9 3E31     REGDP4:MVI A,31
0BCB CD290B   REGDP5:CALL LPRTSB2
0BCE 05             DCR B
0BCF C2BE0B         JNZ REGDP3
0BD2 3E20           MVI A,20
0BD4 CD290B         CALL LPRTSB2
0BD7 3AF2FF         LDA BRKC
0BDA 67       MOV H,A
0BDB CDEC0B         CALL PRHX2
0BDE CD710B         CALL CRLF
0BE1 C9             RET
              ;
```

```
0BE2 1A       PRDEHX4:LDAX D
0BE3 67             MOV H,A
0BE4 1B             DCX D
0BE5 1A             LDAX D
0BE6 6F             MOV L,A
0BE7 1B             DCX D
0BE8 CDEC0B   PRHX4:CALL PRHX2
0BEB 65             MOV H,L
0BEC 7C       PRHX2:MOV A,H
0BED 0F             RRC
0BEE 0F             RRC
0BEF 0F             RRC
0BF0 0F             RRC
0BF1 CDF50B         CALL PRHX1
0BF4 7C             MOV A,H
0BF5 E60F     PRHX1:ANI 0F
0BF7 C630           ADI 30
0BF9 FE3A           CPI 3A
0BFB DA290B         JC LPRTSB2
0BFE C607           ADI 07
0C00 C3290B         JMP LPRTSB2
                    ;
                    ; REGISTER DUMP TITLE(for PRINTER)
0C03 41204620 REG_PR_T:"A F "
0C07 20422043 " B C"
0C0B 20204420 "  D "
0C0F 45202048 "E  H"
0C13 204C2020 " L  "
0C17 20535020 " SP "
0C1B 20205043 "  PC"
0C1F 20202049 "   I"
0C23 58202020 "X   "
0C27 49592020 "IY  "
0C2B 41274627 "A'F'"
0C2F 20422743 " B'C"
0C33 27204427 "' D'"
0C37 45272048 "E' H"
0C3B 274C2720 "'L' "
0C3F 49205220 "I R "
0C43 20535A20 " SZ "
0C47 4820504E "H PN"
0C4B 43204252 "C BR"
0C4F 43       "C"
0C50 0D       DB 0D
                    ;;;
                    ;?  MMEM:LD BC,(ADRSL)
0C51 2AEEFF   MMEM:LHLD ADRSL
0C54 22D6FF   SHLD BCWK
0C57 44       MOV B,H
0C58 4D       MOV C,L
0C59 2AECFF         LHLD DATAL
0C5C 22D8FF   SHLD HLWK
                    ;       PUSH B
                    ;       PUSH H
                    ;?        POP IX
                    ;?        POP IY
0C5F B7             ORA A
                    ;?        SBC HL,BC
```

```
0C60 CD180E    CALL SBCHLBC
0C63 DAC40C          JC ERRDP
0C66 23              INX H
0C67 E5              PUSH H
0C68 21715C          LXI H,$5C71;Fo
0C6B 22F8FF          SHLD LSEG1
0C6E 215000          LXI H,$0050;r
0C71 22FAFF          SHLD LSEG3
0C74 CD720A    MMEM2:CALL DRDSP
0C77 CDC6FF    MMEM3:CALL KEYIN
0C7A FE10            CPI 10
0C7C DA870C          JC MMEM4
0C7F FE15            CPI 15
0C81 CA950C          JZ MMEM5
0C84 C3770C          JMP MMEM3
0C87 47        MMEM4:MOV B,A
0C88 CD460A          CALL DSFTL
0C8B 3AECFF          LDA DATAL
0C8E B0              ORA B
0C8F 32ECFF          STA DATAL
0C92 C3740C          JMP MMEM2
                ;? MMEM5:LD DE,(DATAL)
0C95 2AECFF    MMEM5:LHLD DATAL
0C98 EB        XCHG
0C99 C1              POP B
                ;?       PUSH IY
                ;       POP H
0C9A 2AD6FF    LHLD BCWK
0C9D CD0C0B          CALL HDCMP
0CA0 DAAC0C          JC MMEM6
0CA3 D5              PUSH D
                ;?       LDIR
0CA4 CDFE0D    CALL LDIR
0CA7 E1              POP H
0CA8 1B              DCX D
0CA9 C3BA0C          JMP MMEM7
0CAC EB        MMEM6:XCHG
0CAD 09              DAD B
0CAE EB              XCHG
0CAF 1B              DCX D
                ;?       PUSH IX
                ;       POP H
0CB0 2AD8FF    LHLD HLWK
0CB3 D5              PUSH D
                ;?       LDDR
0CB4 CD0B0E    CALL LDDR
0CB7 E1              POP H
0CB8 13              INX D
0CB9 EB              XCHG
0CBA 22EEFF    MMEM7:SHLD ADRSL
                ;?     LD (DATAL),DE
0CBD EB        XCHG
0CBE 22ECFF    SHLD DATAL
0CC1 C39B09          JMP RINC3
                ;;;
0CC4 21F60D    ERRDP:LXI H,ERRT
0CC7 CD120B    CALL SEGDP
0CCA C3AE08    JMP START2
```

```
                ;;;
                LSAVE:;LHLD ADRSL
                ;?      LD DE,(DATAL)
0CCD 2AECFF     LHLD DATAL
0CD0 EB         XCHG
0CD1 2AEEFF     LHLD ADRSL
0CD4 7C         MOV A,H
0CD5 CD310B     CALL LPRTSB22
0CD8 7D         MOV A,L
0CD9 CD310B     CALL LPRTSB22
0CDC 7A         MOV A,D
0CDD CD310B     CALL LPRTSB22
0CE0 7B         MOV A,E
0CE1 CD310B     CALL LPRTSB22
0CE4 7E         SAVE2:MOV A,M
0CE5 CD310B     CALL LPRTSB22
0CE8 CD0C0B     CALL HDCMP
0CEB CAF20C     JZ SAVE3
0CEE 23         INX H
0CEF C3E40C     JMP SAVE2
0CF2 CD450B     SAVE3:CALL BFOUT
0CF5 C3AE08     JMP START2
                ;;;
                 DMEM:;LHLD ADRSL
                ;?       LD DE,(DATAL)
0CF8 2AECFF     LHLD DATAL
0CFB EB         XCHG
0CFC 2AEEFF     LHLD ADRSL
0CFF E5         DMEM2:PUSH H
0D00 E5               PUSH H
0D01 CDE80B           CALL PRHX4
0D04 E1               POP H
0D05 CD600D           CALL PRSP
0D08 0610             MVI B,10
0D0A CD600D     DMEM3:CALL PRSP
0D0D E5               PUSH H
0D0E 66               MOV H,M
0D0F CDEC0B           CALL PRHX2
0D12 E1               POP H
0D13 23               INX H
0D14 05               DCR B
0D15 C20A0D           JNZ DMEM3
0D18 E1               POP H
0D19 0610             MVI B,10
0D1B CD600D           CALL PRSP
0D1E CD600D           CALL PRSP
0D21 7E              DMEM32:MOV A,M
0D22 CD470D           CALL ASCDUMP
0D25 23               INX H
0D26 05               DCR B
0D27 C2210D           JNZ DMEM32
0D2A CD710B           CALL CRLF
0D2D 2B               DCX H
0D2E CD0C0B           CALL HDCMP
0D31 D2440D           JNC DMEM5
0D34 23               INX H
0D35 D5         DMEM4:PUSH D
0D36 CD4706     CALL KEY
```

```
0D39 D1       POP D
0D3A FEFF     CPI FF
0D3C CAFF0C         JZ DMEM2
0D3F FE16           CPI 16
0D41 C2350D         JNZ DMEM4
0D44 C3AE08   DMEM5:JMP START2
              ;
0D47 FE20     ASCDUMP:CPI 20
0D49 DA5B0D         JC ASCDUMP2
0D4C FE7F           CPI 7F
0D4E DA290B         JC LPRTSB2
0D51 FEA0           CPI A0
0D53 DA5B0D         JC ASCDUMP2
0D56 FEE0           CPI E0
0D58 DA290B         JC LPRTSB2
0D5B 3E2E     ASCDUMP2:MVI A,2E
0D5D C3290B         JMP LPRTSB2
              ;
0D60 3E20     PRSP:MVI A,20
0D62 C3290B         JMP LPRTSB2
              ;
              ; REMOTE
              ;
0D65 211234   REMOTE:LXI H,$3412
0D68 22F4FF   SHLD DISP1
0D6B 215678   LXI H,$7856
0D6E 22F6FF   SHLD DISP3
0D71 CDC005   CALL LEDDPA;=SEGCG   not send data to HOST
0D74 21A00D   LXI H,RLEDOUT
0D77 22CAFF   SHLD RST6JA
0D7A 218C0D   LXI H,RSIN
0D7D 22C7FF   SHLD RST5JA
0D80 3EFF     MVI A,FF
0D82 32B9FF   STA REMOTEMK
0D85 AF       XRA A
0D86 3250FF   STA PRCNT
0D89 C3AE08   JMP START2
              ;
0D8C 3EF7     RSIN:MVI A,F7
0D8E C5       PUSH B
0D8F 47       MOV B,A
0D90 D398     OUT 98
0D92 AF       XRA A
0D93 CD7C06   CALL SOUTSB
0D96 3EFF     MVI A,FF
0D98 47       MOV B,A
0D99 D398     OUT 98
0D9B CDA006   CALL SINSB
0D9E C1       POP B
0D9F C9       RET
              ;
0DA0 CDC005   RLEDOUT:CALL LEDDPA
0DA3 3EF7     RLEDOUT1:MVI A,F7
0DA5 C5       PUSH B
0DA6 47       MOV B,A
0DA7 D398     OUT 98
0DA9 3E04     MVI A,04
0DAB CD7C06   CALL SOUTSB
```

```
0DAE 1E04      MVI E,04
0DB0 21F4FF    LXI H,DISP1
0DB3 7E        RLEDOUT2:MOV A,M
0DB4 CD7C06    CALL SOUTSB
0DB7 23        INX H
0DB8 1D        DCR E
0DB9 C2B30D    JNZ RLEDOUT2
0DBC C1        POP B
0DBD 3EFF      MVI A,FF
0DBF D398      OUT 98
0DC1 C9        RET
               ;
0DC2 21EFFF    ADDSP:LXI H,ADRSH
0DC5 11F4FF    LXI D,DISP1
0DC8 0604      MVI B,04
0DCA 7E        ADDSP2:MOV A,M
0DCB 12        STAX D
0DCC 2B        DCX H
0DCD 13        INX D
0DCE 05        DCR B
0DCF C2CA0D    JNZ ADDSP2
0DD2 CDC9FF    CALL LEDDP
0DD5 C9        RET
               ;
               ;;; SEGMENT DATA(REG MODE)
               ;
0DD6 FF        RSGDT:RST 7;DUMMY
0DD7 FF              RST 7;DUMMY
0DD8 77              DB 77;A
0DD9 71              DB 71;F
0DDA 7C              DB 7C;b
0DDB 39              DB 39;C
0DDC 5E              DB 5E;d
0DDD 79              DB 79;E
0DDE 76              DB 76;H
0DDF 38              DB 38;L
0DE0 6D              DB 6D;S
0DE1 73              DB 73;P
0DE2 73              DB 73;P
0DE3 39              DB 39;C
0DE4 06              DB 06;I
0DE5 76              DB 76;X
0DE6 06              DB 06;I
0DE7 6E              DB 6E;y
0DE8 F7              DB F7;A.
0DE9 F1              DB F1;F.
0DEA FC              DB FC;b.
0DEB B9              DB B9;C.
0DEC DE              DB DE;d.
0DED F9              DB F9;E.
0DEE F6              DB F6;H.
0DEF B8              DB B8;L.
0DF0 06              DB 06;I
0DF1 50              DB 50;r
0DF2 50              DB 50;r
0DF3 77              DB 77;A
0DF4 50              DB 50;r
0DF5 39              DB 39;C
```

```
                ;
0DF6 79         ERRT:DB 79;E
0DF7 50         DB 50;r
0DF8 50         DB 50;r
0DF9 5C         DB 5C;o
0DFA 50         DB 50;r
0DFB 80         DB 80
0DFC 80         DB 80
0DFD 80         DB 80
                ;
0DFE F5         LDIR:PUSH PSW
0DFF 7E         LDIR2:MOV A,M
0E00 12         STAX D
0E01 23         INX H
0E02 13         INX D
0E03 0B         DCX B
0E04 78         MOV A,B
0E05 B1         ORA C
0E06 C2FF0D     JNZ LDIR2
0E09 F1         POP PSW
0E0A C9         RET
                ;
0E0B F5         LDDR:PUSH PSW
0E0C 7E         LDDR2:MOV A,M
0E0D 12         STAX D
0E0E 2B         DCX H
0E0F 1B         DCX D
0E10 0B         DCX B
0E11 78         MOV A,B
0E12 B1         ORA C
0E13 C20C0E     JNZ LDDR2
0E16 F1         POP PSW
0E17 C9         RET
                ;
0E18 D5         SBCHLBC:PUSH D
0E19 57         MOV D,A
0E1A 7D         MOV A,L;SBC HL,BC
0E1B 99         SBB C
0E1C 6F         MOV L,A
0E1D 7C         MOV A,H
0E1E 98         SBB B
0E1F 67         MOV H,A
0E20 DA2A0E     JC SBCHLBC1
0E23 FA2A0E     JM SBCHLBC1
0E26 B5         ORA L
0E27 C22D0E     JNZ SBCHLBC2
0E2A 7A         SBCHLBC1:MOV A,D
0E2B D1         POP D
0E2C C9         RET
0E2D 3E01       SBCHLBC2:MVI A,01
0E2F B7         ORA A;clear sf
0E30 7A         MOV A,D
0E31 D1         POP D
0E32 C9         RET
0E33 06E7       SOUTSB2:MVI B,E7
0E35 C37C06     JMP SOUTSB
                ;
0E38 C5         SBCHLDE:PUSH B
```

```
0E39 47       MOV B,A
0E3A 7D       MOV A,L
0E3B 9B       SBB E
0E3C 6F       MOV L,A
0E3D 7C       MOV A,H
0E3E 9A       SBB D
0E3F 67       MOV H,A
0E40 DA4A0E   JC SBCHLDE1
0E43 FA4A0E   JM SBCHLDE1
0E46 B5       ORA L
0E47 C24D0E   JNZ SBCHLDE2
0E4A 78       SBCHLDE1:MOV A,B
0E4B C1       POP B
0E4C C9       RET
0E4D 3E01     SBCHLDE2:MVI A,01
0E4F B7       ORA A;clear sf
0E50 78       MOV A,B
0E51 C1       POP B
0E52 C9       RET
              ;;;
              ; printer out 16/4/25
              ;
0E53 FE20     PRTJ:CPI 20
0E55 DA610E   JC PRTJ2
0E58 3AB5F0   LDA PKETA
0E5B 3C       INR A
0E5C FE50     CPI 50;=80
0E5E DA620E   JC PRTJ3
0E61 AF       PRTJ2:XRA A
0E62 32B5F0   PRTJ3:STA PKETA
0E65 79       MOV A,C
0E66 CD6B0E   CALL PRT0
0E69 C1       POP B
0E6A C9       RET
0E6B F5       PRT0:PUSH PSW
0E6C DB82     PRT1:IN 82
0E6E B7       ORA A
0E6F FA6C0E   JM PRT1
0E72 F1       POP PSW
0E73 D380     OUT 80
0E75 AF       XRA A
0E76 D383     OUT 83
0E78 00       NOP
0E79 3C       INR A
0E7A D383     OUT 83
0E7C C9       RET
              ;END
ADDSP       =0DC2  ADDSP2      =0DCA  ADRSH       =FFEF
ADRSL       =FFEE  ADST        =098B  AREG        =FFEB
ASCDUMP     =0D47  ASCDUMP2    =0D5B  BCWK        =FFD6
BFOUT       =0B45  BREAK       =09DC  BRKA        =FFF0
BRKC        =FFF2  BRKON       =0A31  CONT        =09C9
CREG        =FFE8  CRLF        =0B71  CRLF2       =0B8B
CRLF3       =0B94  D2          =02EA  DATAH       =FFED
DATAL       =FFEC  DISP1       =FFF4  DISP3       =FFF6
DMEM        =0CF8  DMEM2       =0CFF  DMEM3       =0D0A
DMEM32      =0D21  DMEM4       =0D35  DMEM5       =0D44
DRDSP       =0A72  DRDSP2      =0A7B  DSFTL       =0A46
```

DTSET	=09A1	EREG	=FFE6	ERRDP	=0CC4
ERRT	=0DF6	FREG	=FFEA	HDCMP	=0B0C
HLWK	=FFD8	IN	=0A92	IN1	=0AA5
IN2	=0AA8	IN3	=0AB7	INPUT	=0623
IOMOD	=0A8A	IOREG	=FFD0	IOWK	=FFD2
JPT1	=08C4	JPT2	=08E4	JPT3	=090B
KCIN	=08D4	KCIN2	=08D7	KCIN3	=08DA
KDIN	=08F4	KEY	=0647	KEYIN	=FFC6
KEYINA	=0616	KIDP1	=0982	KIDP2	=0985
KINDP	=097B	LDDR	=0E0B	LDDR2	=0E0C
LDIR	=0DFE	LDIR2	=0DFF	LEDDP	=FFC9
LEDDPA	=05C0	LOAD	=0507	LPRTSB	=0B1C
LPRTSB2	=0B29	LPRTSB22	=0B31	LPRTSB3	=0B4F
LPRTSB4	=0B5C	LPRTSB5	=0B65	LPRTSB6	=0B6E
LREG	=FFE4	LSAVE	=0CCD	LSEG1	=FFF8
LSEG2	=FFF9	LSEG3	=FFFA	LSEG4	=FFFB
LSEG5	=FFFC	LSEG7	=FFFE	LSEG8	=FFFF
MMEM	=0C51	MMEM2	=0C74	MMEM3	=0C77
MMEM4	=0C87	MMEM5	=0C95	MMEM6	=0CAC
MMEM7	=0CBA	NDMON	=0836	NDMON2	=0848
NDMON2_2	=086B	NDMON3	=087A	NDMON32	=0898
NOBRK	=0A21	OUT	=0AC5	PCL	=FFE0
PKETA	=F0B5	PRBF	=FF00	PRCNT	=FF50
PRDEHX4	=0BE2	PRHX1	=0BF5	PRHX2	=0BEC
PRHX4	=0BE8	PRSP	=0D60	PRT0	=0E6B
PRT1	=0E6C	PRTJ	=0E53	PRTJ2	=0E61
PRTJ3	=0E62	RDEC	=09AC	REGDP	=0BA1
REGDP2	=0BAC	REGDP3	=0BBE	REGDP4	=0BC9
REGDP5	=0BCB	REG_PR_T	=0C03	REMOTE	=0D65
REMOTEMK	=FFB9	RGDI2	=093A	RGDI3	=0968
RGDI4	=096F	RGDIN	=092B	RGDSP	=05A1
RGMDE	=0975	RINC	=0991	RINC2	=0995
RINC3	=099B	RLEDOUT	=0DA0	RLEDOUT1	=0DA3
RLEDOUT2	=0DB3	RMOD	=0A60	RMODE	=FFCF
RPRT	=0A56	RPRT2	=0A58	RRDEC	=0AEA
RRINC	=0ADB	RRINC2	=0AE5	RSGDT	=0DD6
RSIN	=0D8C	RST1JA	=FFBB	RST1JC	=FFBA
RST2JA	=FFBE	RST2JC	=FFBD	RST3JA	=FFC1
RST3JC	=FFC0	RST4JA	=FFC4	RST4JC	=FFC3
RST5JA	=FFC7	RST5JC	=FFC6	RST6JA	=FFCA
RST6JC	=FFC9	RST7JA	=FFCD	RST7JC	=FFCC
RUN	=09BD	RWINC	=0AF7	SAVE2	=0CE4
SAVE3	=0CF2	SBCHLBC	=0E18	SBCHLBC1	=0E2A
SBCHLBC2	=0E2D	SBCHLDE	=0E38	SBCHLDE1	=0E4A
SBCHLDE2	=0E4D	SEGCG1	=05C6	SEGDP	=0B12
SIN	=0769	SINERMK	=FFB8	SINSB	=06A0
SOUT	=0747	SOUTSB	=067C	SOUTSB2	=0E33
SPL	=FFE2	SPTOP	=FFB8	START1	=08A4
START2	=08AE	TK2BREAK	=0551	TK2MON	=043B
TK2RST1	=0451	TKBREAK	=0151	TKMON	=003B
TKRST1	=0051	TKRST2	=83D1	TKRST3	=83D4
TKRST4	=83D7	TKRST5	=83DA	TKRST6	=83DD
TRON	=0A51	TRSW	=FFD1	USTOP	=F800
WINC	=09B3				