

## 7. マイクロコンピュータボードによる実験

### 1. 実験目的

マイクロコンピュータ Z80 が搭載されたボードを用いて、マイクロコンピュータの動作を理解するためのプログラムの作成を行う。

### 2. 実験内容

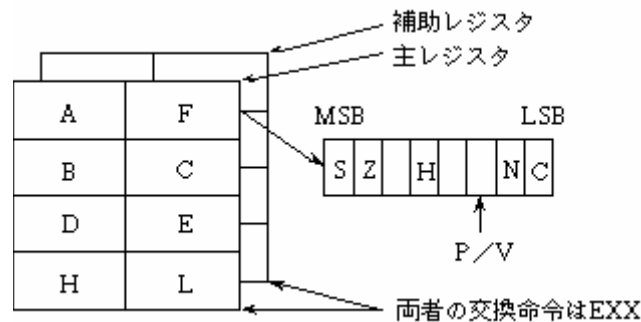
(第1回) 8ビットマイクロコンピュータ Z-80 に用いられる各種の命令を簡単なプログラムを動作させることによって理解する。

(第2回) パラレルデータ用 I/O ポートを用いて LED を点灯、消灯させる方法を修得した後、その LED を右端から左端まで順に1つだけ発光させてゆく動作を繰り返すプログラムを作成する。点灯している LED を移動させるタイミングの決定にはステート数を利用したタイマルーチンを作成して対処する。また、意欲のある学生のためにパラレルデータ用 I/O ポートを用いたステッピングモータの制御に関する実験を用意した。

(第3回) 割り込みについて学習する。モード2の割り込みを利用する。

### 3. 実験に必要な参考事項

Z-80 CPU は、下図に示すようなレジスタ構成を2組持っている。このうち A はアキュムレータである。B, C, D, E, H, L は、それぞれが8ビットの汎用レジスタであると同時に、(B と C), (D と E), (H と L) というペアにより16ビットレジスタとして利用することもできる。F はフラグレジスタで、命令の実行により現われる様々な状態を記憶する。



## § 第1回(簡単なプログラムについて)

### 4. 基本操作

別添の資料を参照して操作方法を理解せよ。

#### 課題1-1

別添の資料の13節<ステップ実行>を実行してみよ。各命令後のAレジスタとBレジスタの状態を確認せよ。

```

                                ORG    08000H
                                main:
8000 3E 02                      LD     A, 02H
8002 06 0B                      LD     B, 0BH
8004 80                          ADD    A, B
8005 76                          HALT
                                END

                                例 3 のプログラム (アセンブラ版)

```

## 5. Z-80 命令に関する実習

### 課題 1-2

下の各プログラムの動作を十分検討した後、ハンドアセンブルし、実機で動作を確認せよ。

(1) レジスタの働き, データ転送命令

- |      |     |          |         |
|------|-----|----------|---------|
| 1. 1 | LD  | A,       | 33H     |
|      | LD  | B,       | 55H     |
|      | LD  | C,       | 77H     |
|      | LD  | A,       | 11H     |
|      | LD  | BC,      | 2233H   |
| 1. 2 | LD  | (8000H), | A       |
|      | LD  | A,       | B       |
|      | LD  | (8001H), | A       |
|      | LD  | A,       | C       |
|      | LD  | (8002H), | A       |
|      | LD  | (8003H), | BC      |
| 1. 3 | LD  | A,       | (8001H) |
|      | LD  | B,       | A       |
|      | LD  | A,       | (8000H) |
|      | ADD | A,       | B       |
|      | LD  | (8002H), | A       |

(2) 加算、減算命令

- |      |     |    |         |
|------|-----|----|---------|
| 2. 1 | LD  | A, | (8001H) |
|      | CPL |    |         |
|      | ADD | A, | 01H     |

```

LD    B,          A
LD    A,          (8000H)
ADD   A,          B
LD    (8002H),   A
2. 2  LD    A,          (8001H)
LD    B,          A
LD    A,          (8000H)
SUB   B
LD    (8002H),   A

```

(3) 比較命令

```

3. 1  LD    A,          (8000H)
      CP    3AH

```

(4) ジャンプ命令

```

4. 1  8000:
      LD    A,          00H
      LD    B,          20H
      8004:
      ADD   A,          B
      JP    8004H
4. 2  LD    A,          00H
      LD    B,          20H
      L1:
      ADD   A,          B
      JP    P,          L1
4. 3  LD    A,          00H
      LD    B,          20H
      L1:
      ADD   A,          B
      JP    NC,         L1
4. 4  LD    A,          00H
      LD    B,          20H
      L1:
      ADD   A,          B
      JP    NZ,         L1

```

(5) 相対ジャンプ命令

```
5.1  LD    A,      00H
      LD    B,      20H
      L1:
      ADD  A,      B
      JR   L1
```

(6) 論理演算命令

```
6.1  LD    A,      55H
      AND  FOH
      LD    A,      55H
      OR   FOH
      LD    A,      55H
      XOR  FOH
```

(7) ローテイト、シフト命令

```
7.1  LD    A,      1FH
      SRL  A
      SRL  A
      SRL  A
      SRL  A
      LD    A,      1FH
      RRC  A
      RRC  A
      RRC  A
      RRC  A
```

検印

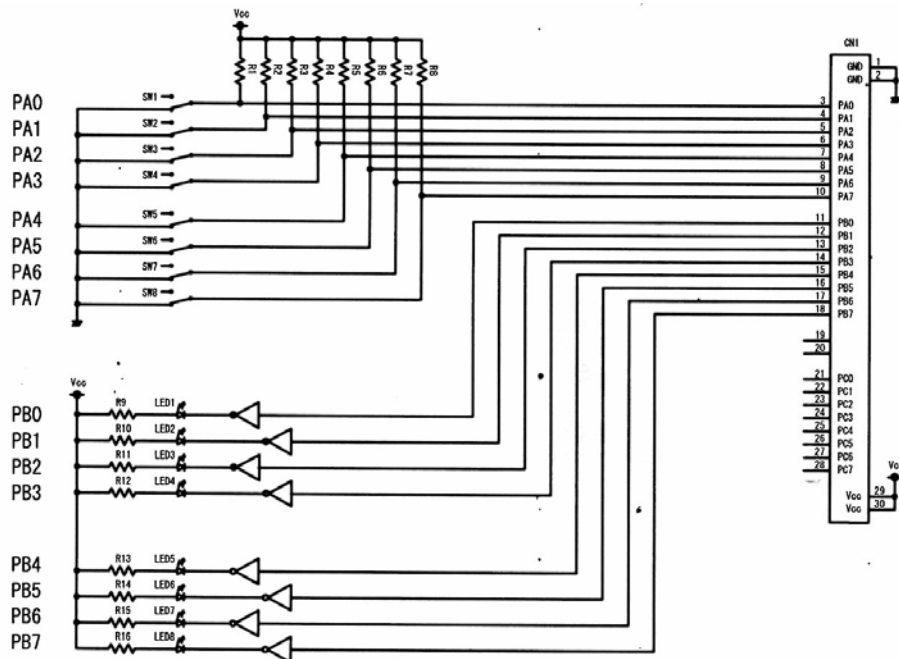
## § 第2回 (I/O ポートの制御)

### 6. LED の点灯に関する実習

入出力装置とのデータのやり取りは、I/O ポートを介して行われる。本実習ボードにも 8255

というパラレルデータ用 I/O ポート LSI が接続されている (IC5). 8255 には A, B, C の 3 つのポートがあり, それぞれのポートは 8 ビットで構成されている. 各ポートはビットごとに入出力を指定できる. LED ボード中のスイッチ PA0-PA7 は A ポートに, LED1-LED8 は B ポートに接続されている.

(回路図)



名称	I/O アドレス	入出力設定アドレス	入出力 (ビットごとの入出力は省略するがわかるだろう)
ポート A	04H	14H	00 出力 FF 入力
ポート B	05H	15H	00 出力 FF 入力

### 課題 2-1

次のプログラムは LED の右 4 ビットを点灯するプログラムである.

- (1) これの動作確認をせよ.
- (2) 左 4 ビットを点灯するように変更せよ.

```

8000      LD A, 00H          3E 00
8002      OUT (15H), A      D3 15 /* B ポートを出力に設定 */
8004      LD A, 0FH          3E 0F
8006      OUT (05H), A      D3 05 /* B ポートに A レジスタの内容 0F を出力する*/

```

8008

HALT

76

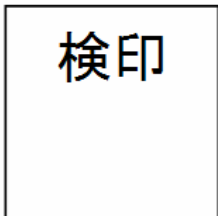
/\* 停止 \*/



課題 2 - 2

A ポートにつながっているディップスイッチを入力に設定してディップスイッチの値を B ポートに接続されている LED に出力する次のプログラムを完成せよ。

8000	LD A, 00H	3E00	
8002	OUT (15H), A	D3 15	/* B ポートを出力に設定 */
8004	LD A, █	█	/* A ポートを入力に設定 */
8006	█	█	}
8008	IN A, (04H)	█	
800A	OUT █, A	█	/* A レジスタの内容を B ポート (LED) に出力 */
800C	JP (8008H)	C3 08 80	/* 8008H へジャンプして次の入力を持つ */



7. ステート数を利用したタイマルーチンの作成

機械語の各命令は、マイクロプロセッサのクロック周波数に同期して行われ、各命令の実行時間はクロック周波数の 1 周期の倍数となっている。クロック周波数の 1 周期の時間を 1 ステートという。この実験ボードでは、クロック周波数=6MHz だから

$$1 \text{ ステート} = 1/6\text{MHz} = 1/(6 \times 10^6) = 0.167 \times 10^{-6} \text{ (sec)} = 0.167 (\mu\text{s})$$

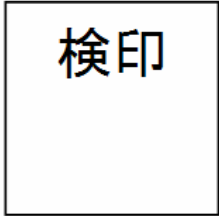
である。このことを利用して 1 秒タイマルーチンを作成する。さらに、LED を右端から左端まで 1 つずつ点灯させて行くプログラムに応用する。

課題 2 - 3

次のプログラムをステップ実行してみよ。その後、AUTO 実行するとどうなるか？

8000	LD A, 00H	3E 00	
8002	OUT (15H), A	D3 15	/* B ポートを出力に設定 */

8004	LD A, 01H	3E 01
8006	OUT (05H), A	D3 05
8008	RLC A	07 /* 左に1ビットシフト */
8009	JP (8006H)	C3 06 80



AUTO 実行した結果は全ての LED が点灯という結果となる。これは、点灯、消灯の速度が速すぎ、人の目では知覚できないためである。そこで、知覚できるようにするため、特に点灯間隔を 1 秒となるようにする。

8000	LD A, 33H	ステート数 T=7
8002	LD B, A	ステート数 T=4

は A レジスタに 33H を格納し、B レジスタに A レジスタの 33H を移すというのだが、付録の命令表のステート数 T を見ると LD r, n が T=7 であるのに対し、LD r, r' は T=4 である。よってこの命令 2 つの実行時間は  $(7+4) \times (1/6) (\mu s) = 1.83 (\mu s)$  となる。以下では 8100 番地から始まるサブルーチン（まあ、C 言語の関数みたいなものです。C 言語の関数がサブルーチンみたいなものと言ったほうがよいかも知れない）で 0.1 秒を作っている。

#### 課題 2-4

以下のプログラム中、WAIT:サブルーチンが 0.1 秒をつくるように 8102 番地の BC レジスタに代入する値  $x$  を求め、プログラムを完成せよ。 $x = \text{FFFF} = 65535$  にしても 1 秒には足りないから、0.1 秒を作っていることに注意する。メインルーチンの方で B レジスタを利用して 1 秒を作っている。

$$(式) \quad (11 + \square + 10 + (6+4+4+10) \times x + \square + \square + \square) \times \frac{1}{6} \times 10^{-6} = 0.1$$

より  $x = \square$  (10 進), 16 進に直すと  $x = \square$  H

8000	LD A, 00H	3E 00
8002	OUT (15H), A	D3 15
8004	LD A, 01H	3E 01

8006	OUT (05H), A	D3 05	
8008	RLC A	07	
8009	LD B, █	█ █	/* 上の注意を読めば何を代入すればよいか わかるはず*/
800B	CALL WAIT:	CD 00 81	/* 8100 から始まるサブルーチン呼び出す。 サブルーチン終了後はこのアドレス 800B に戻 る*/
800E	DEC B	05	
800F	JP NZ █ █	C2 █ █	/* どのアドレスにジャンプするのだろう */
8012	JP █ █	C3 █ █	/* 同じく*/
WAIT:			
8100	PUSH AF	F5	/* PUSH と POP については下の説明を参照 T=11 */
8101	PUSH BC	C5	/* T=█ */
8102	LD BC, █ █	01 █ █	/* T=10 <u>ここが問題の部分</u> . 反転することも 忘れずに*/
8105	DEC BC	█	/* T=6 */
8106	LD A, B	█	/* T=4 */
8107	OR C	█	/* T=4 */
8108	JP NZ (8105H)	C2 05 81	/* T=10 */
810B	POP BC	C1	/* T=█ */
810C	POP AF	F1	/* T=█ */
810D	RET	C9	/* T=█ */

この部分は(8102H)で決めた回数ループする

検印

サブルーチン WAIT の補足 :

(1) 8100 と 8101 の PUSH は AF レジスタと BC レジスタの値をスタックに格納する. このようにしないと, メインルーチンのレジスタの値がサブルーチンで書き換えられてしまい, おかしなことになる. C 言語の変数のローカル宣言と思えばよい.



C
B
F
A

スタック上ではデータは積み上げられてゆく（だからスタックな訳だが）．  
 そして、取り出すときは上のものから取り上げられている．よって取り出す命令  
 POP は POP BC, POP AF の順となる．

(2) 8107 は何をやっているかという、A レジスタに入った B レジスタの値と C レジスタの値の論理和をとっている．これが 0 となるのは、B と C が共に 0 のときとなる．

レポート用の問題として点灯間隔をより 1 秒に近いようにするにはどうすればよいかという問題をあげておこう．

8. ステッピングモータの制御（この課題は 15:30 までに 7 を終えた者だけが希望により取り組むことができる．もちろんこの課題を終えることができれば成績に加算される．課題 2-5 のみ取り組むのも可とする．）

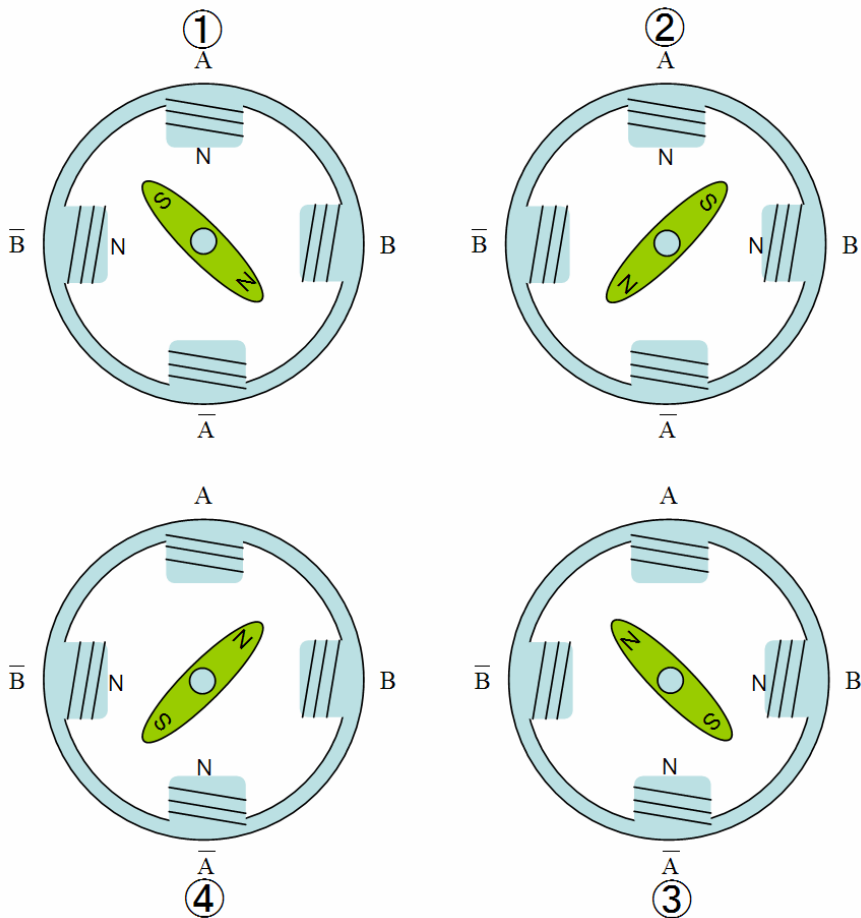
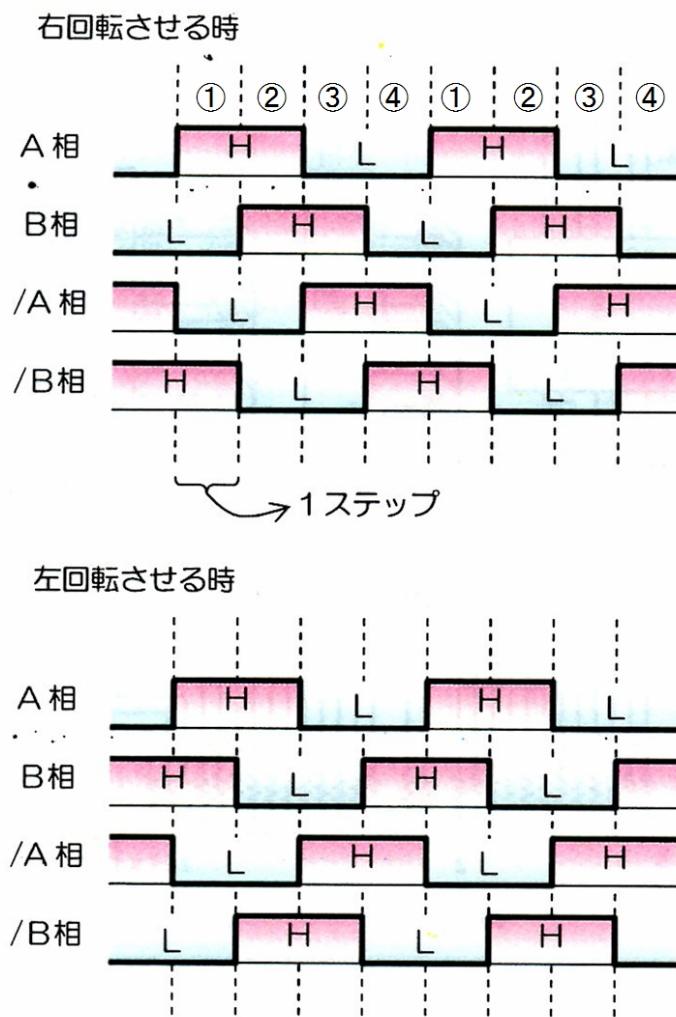


図 A

ステッピングモータは入力パルス数に比例して正確な回転運動をさせることができるモータであり、プリンタの給排紙など正確な位置決めが必要な場面で使用されている。図Aは4つの電磁石をもつステッピングモータを表す。電磁石（2つ）が電流を流すことでN極に磁化される度に回転子のS極が引かれて行くことがわかる。図Bの①～④を1回行うことにより回転子はちょうど1回転する。

この実験の実際のステッピングモータは図Bの①～④を1回行うことにより回転子を72°回転する仕様になっている。



図B

ステッピングモータ接続に関する 8255 のポート割り当て状況について述べる。Aポートは一応入力に設定する。Aポートからの入力によりキャリアの位置を把握することができるが、今回は使わない。Bポートの下位4ビットがステッピングモータのA, B,  $\bar{A}$ ,  $\bar{B}$  に接続されている。よってBポートは出力に設定する。Cポートは上位4ビットは入力とし、キャリアが左端に到達したか、右端に到達したかを知ることができる。下位4ビットは出力とする。実際使うのは、最下位のビットだけでダイレクトパルスモードを設定するのに用いるだけ

である。詳しくは図Cのようになる。

## A. ポート割り当て

### Port A (モータユニット → マイコン)

PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
/		符号ビット	位置データ (スライドVRの値)				

位置データの値の範囲は、-64 ~ +63 (40H~3FH) です。  
センターが 0 となります。

### Port B (マイコン → モータユニット)

スピードモード時 (PCO : 0の時)

PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
移動方向		移動速度					

移動方向は、左への移動は「1」、右への移動は「0」を指定します。

移動速度は、0 (停止) ~ 127 (最速) で指定します。

ダイレクトパルスモード時 (PCO : 1の時)

PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
/				A相	B相	/A相	/B相

バイポーラ型のモータなので、2相励磁の信号を出力してください。

### Port C (上位: モータユニット → マイコン

### 下位: マイコン → モータユニット)

PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
リミット 左	リミット 右	プッシュ スイッチ SW1	プッシュ スイッチ SW2	/	/	/	Direct mode

リミット左は、キャリアが左端に来た時に 1 となります。

リミット右は、キャリアが右端に来た時に 1 となります。

プッシュスイッチは、押された時に 1 となります。

Direct mode は、「ダイレクトパルスモード」を選択するときに1にします。  
(「スピードモード」選択するときは0にします。)

図 C

## 課題 2-5 (任意)

次のプログラムはキャリアが左端に到達したとき、右側に移動に変更し、右端に到達したときは、左に移動に変更するプログラムである。実行し、プログラムの内容を説明せよ。

検印

ステッピングモータの制御 (繰り返し)

```
8000 LD A, FFH      3E FF      /* A ポートを入力に設定 */
8002 OUT (14H), A  D3 14
8004 LD A, 00H     3E 00
8006 OUT (15H), A  D3 15
8008 LD A, F0H     3E F0      /* C ポート上 4 ビット入力, 下 4 ビット出力
*/

800A OUT (16H), A  D3 16
800C LD A, 01H     3E 01
800E OUT (06H), A  D3 06      /* C ポートを direct mode に設定 */
8010 LD H, 33H     26 33      /* Pulse data */
8012 IN A, (06H)   DB 06
8014 AND 80H       E6 80      /*リミット左のチェック*/
8016 JP NZ, (8024H) C2 24 80
8019 LD A, H       7C
801A OUT (05H), A  D3 05
801C CALL WAIT:    CD 00 81
801F RLCA         07
8020 LD H, A       67
8021 JP (8012H)    C3 12 80
8024 IN A, (06H)   DB 06
8026 AND 40H      E6 40      /*リミット右のチェック*/
8028 JP NZ, (8012H) C2 12 80
802B LD A, H       7C
802C OUT (05H), A  D3 05
802E CALL WAIT:    CD 00 81
8031 RRCA         0F
8032 LD H, A       67
```

8033	JP (8024H)	C3 24 80	
8036	RET	1F	
WAIT:			
8100	PUSH AF	F5	/* PUSH と POP については下の説明を参照 T=11 */
8101	PUSH BC	C5	/* T=■ */
8102	LD BC, 1F 40	01 40 1F	/* T=10 この部分は 1F 40 を設定する*/
8105	DEC BC	■	/* T=6 */
8106	LD A, B	■	/* T=4 */
8107	OR C	■	/* T=4 */
8108	JP NZ (8105H)	C2 05 81	/* T=10 */
810B	POP BC	C1	/* T=■ */
810C	POP AF	F1	/* T=■ */
810D	RET	C9	/* T=■ */

} この部分は(8102H)で決めた回数ループする

課題 2 - 6 (任意)

このステッピングモータは図 B, ①~④を 1 回入力するごとに 72° モータが回転する仕様になっている。また, モータ 1 回転あたり, 17.5mm キャリアは移動する。課題 2-5 を利用して原点から出発して

(1) 左に 3.5cm 移動

(2) 右に 7.0cm 移動

を行うプログラムを作成せよ。

課題 2 - 6 (任意)

ステッピングモータの制御

8000	LD A, FFH	3E FF	/* A ポートを入力に設定 */
8002	OUT (14H), A	D3 14	
8004	LD A, 00H	3E 00	
8006	OUT (15H), A	D3 15	
8008	LD A, F0H	3E F0	/* C ポート上 4 ビット入力, 下 4 ビット出力 */
800A	OUT (16H), A	D3 16	

800C LD A, 01H 3E 01

800E OUT (06H), A D3 06

/\* C ポートを direct mode に設定 \*/

検印

## § 第3回(割り込みの概念)

### 9. 割り込み

割り込みとはプログラム中に外部装置からの信号により実行中のプログラムを中断させ、あらかじめ設定しておいた別のプログラムで処理を行うことをいう。パソコンでいえば、マウスが動いたときにどうするかということに相当する。

Z-80 には、ソフトウェアにより割り込みの許可/禁止を切り換えられるマスクابل割り込みと、禁止できないノンマスクابل割り込みがある。さらに前者のマスクابل割り込みには、モード0, 1, 2 の3種の割り込みがある。このうち、今回の実験では、モード2の割り込み(ベクタ割り込みとも呼ばれる)を使う。

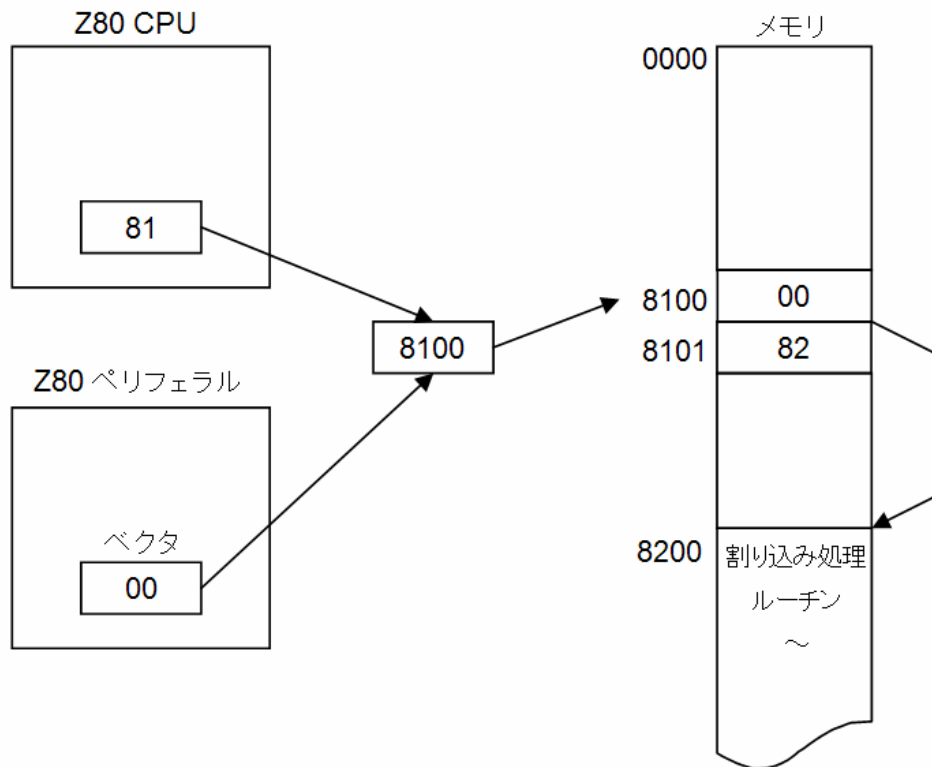


図 D

モード2の割り込みを利用するには、プログラムの始めの部分において“IM 2”という命令を実行することにより宣言される。割り込み処理ルーチンの先頭アドレスの決定過程は、上の図に示されているようなものである。まず、割り込みをかけたペリフェラル(かっこつけない言い方をすれば、この実験の場合だったらジュース80円～ビール230円のボタンとかお金10円～500円のボタンのこと)が、ベクタと呼ばれる8ビットのアドレスデータをCPUに送り、CPUがこの8ビットアドレスとCPU内のIレジスタの8ビットと合わせて16ビットのアドレスデータとし、このアドレスから始まるメモリ2バイトに格納されている数値を割り込み処理ルーチンの先頭アドレスとする方式である。ベクタはペリ

フェラルごとに違っており、この実験ならば

ベクタ 00H : コインスイッチを押す  
ベクタ 02H : 商品選択スイッチを押す  
ベクタ 04H : オペレーションスイッチを押す

となっている。つまり、3種類のスイッチに対応して異なる3つの動作をさせることができることになる。後の課題3-1ではベクタ 00H、3-2ではベクタ 00Hと04Hを用いる。割り込み処理を含むプログラム作成上の留意点を以下に列挙する。

- (1) メインプログラムの始めにおいて、どの種の割り込みを利用するか明確に宣言すること (IM2の宣言)。
- (2) マスカブル割り込みでは、割り込み処理が開始されると自動的にDI(割り込み禁止)の状態になる。そのため、割り込み処理ルーチンから抜ける手前でEI(割り込み許可)する必要がある。そのため、Z80ファミリを使用した場合、EI命令の後RETN命令を用いて割り込み処理ルーチンから抜けることにより、次回の割り込みに確実に対処できる。
- (3) 割り込みテーブルの番地指定は、下位、上位の順である。

### 課題3-1

次のプログラムは、コインスイッチが押されたときに割り込みをかけ、押されたスイッチ (アドレス 0xFBH) の値を取り込み、オペレーション (アドレス 0xFCH) のLEDに出力するプログラムである。プログラムを完成せよ。スイッチのI/Oアドレスは、以下の資料を参考にせよ (後の課題のためにCレジスタは現在自販機に入っている金額、Dレジスタは、現在投入したコインを想定している)。

8000	■	■ ■	/* 上の注意の(1) */
8002	LD A, 81H	3E 81	/* 別に81でなくてもよいのだが、図Dに合わせた */
8004	LD I, A	■ ■	
8006	LD C, 00H	0E 00	/* まだお金が入っていない */
8008	LD D, 00H	16 00	/* まだお金を入れていない */
800A	■	■	/* えいと気合を入れて割り込み可とする */
800B	LD A, D	■	/* 割り込みがかかっていたらDに値が入っている */
800C	OUT (■), A	■ ■	/* オペレーションLEDに出力 */
800E	JP (800BH)	C3 0B 80	/* <u>ここは重要</u> . 800Bから800Eをプログラムが



ループしている間に (いつ押されるかわからない) 割り込みを待つのである\*/

8100  
8101

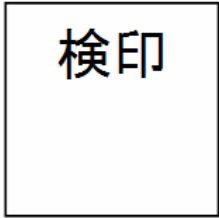


/\*何を書けばよいのだろうか？図Dをみよ\*/

/\* 以下が割り込み処理ルーチンの本体\*/

8200 IN A, (FAH) DB FA  
8202  
8203  
8204 RETI

/\*ここはサービス. コインスイッチの I/O アドレスは(0xFAH)である. \*/  
/\*Aレジスタの値をDレジスタに移す.つまり, お金が入ったことになる\*/  
/\*” えい” と再び気合を入れる\*/  
/\*上の注意の(2)\*/



2. 割り込みベクタ割り当て

本装置は、Z80 のモード 2 割り込みを使用します。

ベクタ#0 : コイン SW 押下  
ベクタ#1 : 商品選択 SW 押下  
ベクタ#2 : オペレーション SW 押下

3. I/O アドレス割り当て

FAh (0xFA): コイン

IN							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	0	0	0	500 円 SW	100 円 SW	50 円 SW	10 円 SW

OUT							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
割り当てなし	割り当てなし	割り当てなし	割り当てなし	割り当てなし	割り当てなし	割り当てなし	割り当てなし

FBh (0xFB): 商品

IN							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	0	0	0	ビール SW	コーヒー SW	紅茶 SW	ジュース SW

OUT							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
割り当てなし	割り当てなし	割り当てなし	割り当てなし	ビール LED	コーヒー LED	紅茶 LED	ジュース LED

FCh (0xFC): オペレーション

IN							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
取消 SW	0	0	釣銭 SW	商品取出 SW	商品取出 SW	商品取出 SW	商品取出 SW

OUT							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
割り当てなし	割り当てなし	割り当てなし	釣銭 LED	取出口 ビール LED	取出口 コーヒー LED	取出口 紅茶 LED	取出口 ジュース LED

FDh (0xFD): 金額表示用 7SEG LED (一の位)  
FEh (0xFE): // (十の位)  
FFh (0xFF): // (百の位)

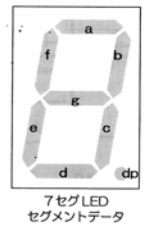
IN							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Don't care	Don't care	Don't care	Don't care	Don't care	Don't care	Don't care	Don't care

OUT							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
dp	g	f	e	d	c	b	a

4. コネクタピン割り当て

50P バスコネクタ (CN1)  
(80Z Mk5)

No.	用途	No.	用途
1	GND	2	GND
3	A0	4	A1
5	A2	6	A3
7	A4	8	A5
9	A6	10	A7
11	DO	12	D1
13	D2	14	D3
15	D4	16	D5
17	D6	18	D7
19	/INT	20	/RESET
21	/RD	22	/WR
23	/M1	24	/ORQ
25	/HALT	26	/WAIT
27	/NMI	28	CLK
29	Vcc	30	Vcc
31	(未接続)	32	(未接続)
33	(未接続)	34	(未接続)
35	/MREQ	36	/RFSH
37	/BUSREQ	38	/BUSACK
39	A8	40	A9
41	A10	42	A11
43	A12	44	A13
45	A14	46	A15
47	Vcc	48	Vcc
49	GND	50	GND



使用コネクタ  
XG4A-5071 (オムロン)  
適合コネクタ  
XG4M-5031-T (オムロン) 等

### 課題 3-2

課題 3-1 と同じ状況で内部状態 (C レジスタ) が  
0 円 (00), 5 0 円 (02), 1 0 0 円 (04)  
をとるものとし,

- (1) 内部状態が 5 0 円で 1 0 0 円が投入された場合, 商品取出の LED を 1 つ点灯
- (2) 内部状態が 1 0 0 円で 5 0 円が投入された場合, 商品取出の LED を 1 つ点灯
- (3) 内部状態が 1 0 0 円で 1 0 0 円が投入された場合, 商品取出の LED を 1 つ点灯, 釣  
銭取出の LED も 1 つ点灯
- (4) 商品取出, 釣銭取出ボタンが押されたら割り込みをかけて消灯

するプログラムを作成せよ. 内部状態は I/O アドレス (0xFBH) につながれている LED に表示  
するものとする. (順序回路の実験と同じ内容)

8000			/* 上の注意の (1) */
8002	LD A, 81H	3E 81	/* 別に 81 でなくてもよいのだが, 図 D に合 わせた */
8004	LD I, A		
8006	LD C, 00H	0E 00	/* まだお金が入っていない */
8008	LD D, 00H	16 00	/* まだお金を入れていない */
800A			/* えいと気合を入れて割り込み可とする*/
800B	LD A, D		/* 割り込みがかかっていたら D に値が入って いる */
800C	CP 00		/* D=0 か?*/
800E	JP Z (802EH)	CA 2E 80	/* D=0 ならば (802EH) へ*/
8011	, 00	00	/* D=0 にしておく*/
8013	,		/* A レジスタに C レジスタの値を加える */
8014	LD C, A	4F	/* C レジスタに A レジスタの値を移しておく */
150 円:			
8015	CP 06H	FE 06	/* 150 円 (=06H) か?*/
8017	JP NZ 200 円:	C2 23 80	/* 150 円でなければ 200 円かを調べるためジ ャンプ */
801A	LD A, 01H	3E 01	/* 商品取出 LED を点灯のため */
801C	OUT (FCH)	D3 FC	
801E	LD C 00	0E 00	/* C レジスタを 0 にクリア */

8020 JP (802EH) C3 2E 80 /\* もちろん 200 円の場合は調べない \*/

200 円 :

8023 ■ ■ ■ ■ /\* 200 円か? \*/

8025 ■ ■ (802EH) ■ ■ ■ /\* 200 円でなかったら (802EH) へ \*/

8028 ■ A, ■ ■ ■ /\* 商品取出と釣銭取出の LED を点灯 \*/

802A ■ ( ■ ) ■ ■

802C ■ , ■ H ■ ■ /\* C レジスタを 0 にクリア \*/

内部状態 :

802E ■ , ■ ■ /\* C レジスタの値を A レジスタに \*/

802F OUT (FBH) D3 FB /\* 商品選択スイッチの LED に出力 \*/

8031 JP ( ■ H) ■ ■ ■ /\* どこにジャンプすればよいのか? \*/

8100 ■ /\* 何を書けばよいのだろうか? 図 D をみよ \*/

8101 ■

81 ■ 00 /\* 商品取出と釣銭取出の LED の消灯用の割り込みアドレスの設定 \*/

81 ■ 83

/\* 以下が割り込み処理ルーチンの本体 \*/

8200 IN A, (FAH) DB FA /\* ここはサービス. コインスイッチの I/O アドレスは (0xFAH) である. \*/

8202 ■ ■ ■ /\* A レジスタの値を D レジスタに移す. つまり, お金が入ったことになる \*/

8204 ■ ■ /\* ” えい ” と再び気合を入れる \*/

8204 RETI ■ ■ /\* 上の注意の (2) \*/

8300	LD A, 00	3E 00	/*商品取出と釣銭取出の LED の消灯用*/
8302	OUT (FCH)	D3 FC	
8304	EI	FB	
8305	RETI	ED 4D	

検印

文責 渡辺 宏太郎 (情報数理研究室), 黒川 恭一 (コンピュータ工学研究室)  
2010/03/16 第 1 版