

MSX-TIMER

仕様書

目次

1. 概要.....	3
2. I/O ポート.....	4
2.1. レジスタの読み書き.....	4
2.2. 割り込み要因のチェック.....	5
2.3. カウンターの読みだし.....	6
3. レジスタ.....	6
3.1. モードレジスタ.....	6
3.2. カウントレジスタ.....	8
3.3. コントロールレジスタ.....	8
4. 詳細な動作説明.....	9

1. 概要

本書は、MSX-TIMER の仕様書である。制御方法、動作タイミングを規定するものである。
MSX-TIMER には、大まかに下記の機能がある。

- (1) ワンショットタイマー
- (2) コンティニューアスタイマー
- (3) フリーランカウンター
- (4) 割り込み
- (5) 精度指定
- (6) 4つのタイマー

各機能について、次章以降で詳細に説明する。

MSX-TIMER は、Y8960 Cartridge、MSX2++以降の本体に搭載される。
基本的に、インクリメントカウンターによるシンプルなタイマーハードウェアである。

2. I/O ポート

I/O ポートは、4 ポート占有する。下記に表 1 I/O ポート一覧を示す。

表 1 I/O ポート一覧

ポート番号	機能
74h	レジスタ番号を指定する。
75h	74h で指定したレジスタを読み書きする。
76h	現在発生中の割り込みフラグの読み出しとクリア。
77h	カウンターの読みだし。

2.1. レジスタの読み書き

レジスタへの書き込みは、下記のように 74h に書き込みたいレジスタ番号を書き込み後、75h にそのレジスタへ書き込みたい値を書き込む。

```
DI  
LD A, register_number  
OUT (74h), A  
LD A, value  
OUT (75h), A  
EI
```

レジスタの読みだしは、下記のように 74h に読みだしたいレジスタ番号を書き込み後、75h を読みだす。

```
DI  
LD A, register_number  
OUT (74h), A  
IN A, (75h)  
EI
```

74h に書いた値は、74h を読みだすことによって読むことができるため、例えば割り込み処理の中で、もとの値を読みだして保持しておき、割り込みから抜けるときにもとの値へ戻すことも出来る。

interrupt_handler:

```
(レジスタの待避など)  
IN      A, (74h)  
PUSH    AF  
(割り込み内の処理)  
POP    AF  
OUT    (74h), A  
(レジスタの復帰など)  
RET
```

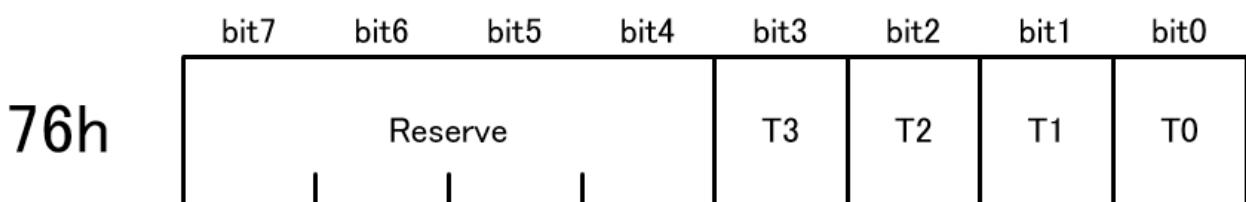
2.2. 割り込み要因のチェック

MSX は、標準的には Z80 の IM1 で動作する。割り込みが発生すると 0038h がコールされ、そこから BIOS の割り込み処理ルーチンへジャンプする。そして、真っ先にフック H.KEYI (FD9Ah) にジャンプする。通常は、H.KEYI に割り込み処理ルーチンをフックすることで、割り込みを受けるが、VDP の割り込みも、MSX-TIMER の割り込みも、全てここを通過するため、割り込み処理ルーチンは「自身が処理すべき割り込みなのか?」を判定して、そうで無い場合は速やかに戻る必要がある。

MSX-TIMER は、独立した 4 つのタイマーを持っているが、この 4 つのどの割り込みが発生したかを、I/O 76h を一回読むだけで判断する情報を得ることができる。

読みだした値(8bit) の各ビットには、T0~T3 の名前が付けられている。

図1. 76h のビットマップ



読みだした値のうち、bit4-7 の Reserve には 0 が入っている。

bit0 が T0, bit1 が T1, bit2 が T2, bit3 が T3 に対応しており、それぞれ Timer0~Timer3 の割り込み要因フラグに対応している。つまり「T0=1 なら Timer0 の割り込みが発生している状態」を示しているのである。「TN=1 なら TimerN の割り込みが発生している状態」である。

このどれか1つでも1である状態の時に、CPUが割り込みを受けられる状態になると、割り込みが発生することになる。割り込みハンドラーは、EIする前に「自身が処理したタイマー割り込み」をクリアする必要がある。

クリアは、T0～T3のクリアしたいbitに1を立てた値を76hへ書き込む事により実現する。

従って、MSX-TIMERの割り込み処理を全て引き受ける割り込みハンドラであれば、76hから読んだ値をそのまま書き込んでやればクリア出来るのである。

2.3. カウンターの読みだし

77hの下位2bitに対して、0～3の値を書き込み、その後に77hを読みだすとカウント値を読みだすことができる。

3. レジスタ

4つのタイマーは、全て同じ機能を持つ。周期等を個別に指定できるため、目的に応じて4つのタイマーを使い分けることができる。

1つのタイマーモジュールには、表1レジスター一覧に示す2個のレジスタが存在する。

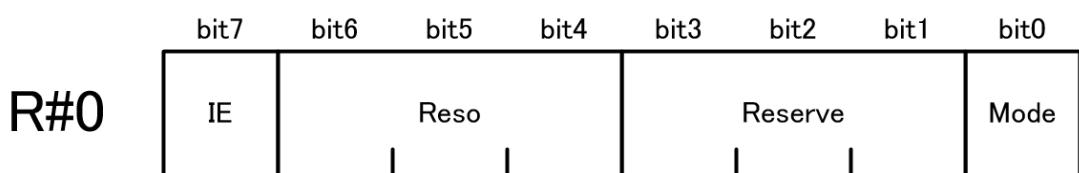
表1 レジスター一覧

レジスタアドレス	機能
0	モードレジスタ。挙動を決めるレジスタである。
1	カウントレジスタ。
2	コントロールレジスタ。

3.1. モードレジスタ

モードレジスタ(0)のビットマップを図1に示す。

図1. モードレジスタのビットマップ



Mode の設定値を、表 2 Mode レジスタの設定値一覧に示す。

表 2 Mode レジスタの設定値一覧

設定値	意味
0	one shot
1	repeat

one shot (0) にすると、所定の時間が経過したらそこで止まる。

repeat (1) にすると、所定の時間が経過したら、0 に戻る。

いずれの場合も、「所定の時間」はカウントレジスタを使って指定する。

Reso の設定値を、表 3 Reso レジスタの設定値一覧に示す。

表 3 Reso レジスタの設定値一覧

設定値	意味
0	11.92 [usec] 単位
1	23.84 [usec] 単位
中略	-
6	71.58 [usec] 単位
7	83.48 [usec] 単位

クロック源は、85.90908MHz である。内部に 32bit のカウンターを持っており、0～4,294,967,295 までカウントできる。

$$4,294,967,295 / 85,909,080 \doteq 49.99 [\text{sec}]$$

最大で、49.99[sec] までカウントできる。

しかし、MSX で 32bit カウンターは扱いにくいことと、1 カウント $\doteq 11.64 [\text{nsec}]$ と、分解能が高すぎるため、これを 1024 分周した 11.92 [usec]（内部 32bit カウンターの下位 10bit を無視）を単位としてカウントレジスタに指定する値の分解能を指定する。

IE の設定値を、表 4 IE レジスタの設定値一覧に示す。

表 4 IE レジスタの設定値一覧

設定値	意味
0	interrupt disable
1	interrupt enable

IE レジスタは、割り込みの発生を制御するレジスタである。

3.2. カウントレジスタ

タイマーの時間を指定するレジスタがカウントレジスタである。

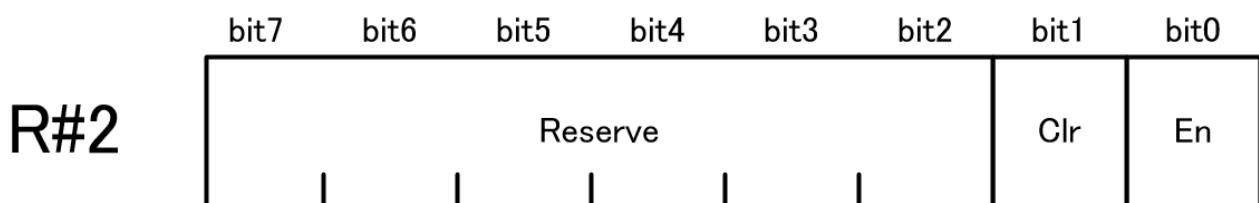
1 カウントに対応する時間は、モードレジスタ内の Reso レジスタで指定する。

カウント値としては、0~255 を指定できるが、0 は 256 の意味になる。

3.3. コントロールレジスタ

コントロールレジスタ (2) のビットマップを図 1 に示す。

図 1. コントロールレジスタのビットマップ



En の設定値を、表 5 En レジスタの設定値一覧に示す。

表 5 En レジスタの設定値一覧

設定値	意味
0	count disable
1	count enable

`count disable (0)` にすると、カウンターのインクリメントが停止する。一次停止。

`count enable (1)` にすると、カウンターがインクリメントする。稼働状態。

`Clr` は、1 を書き込むとカウンターをクリアする。読み出し時は常に 0 になる。

`Reserve` は、将来予約のスペースである。書き込み時には 0 を書き込むこと。読み出し時には 0 が返ってくる。

4. 詳細な動作説明

T.B.D.

更新履歷

2026/Jan/14th v0.1 HRA! 初版