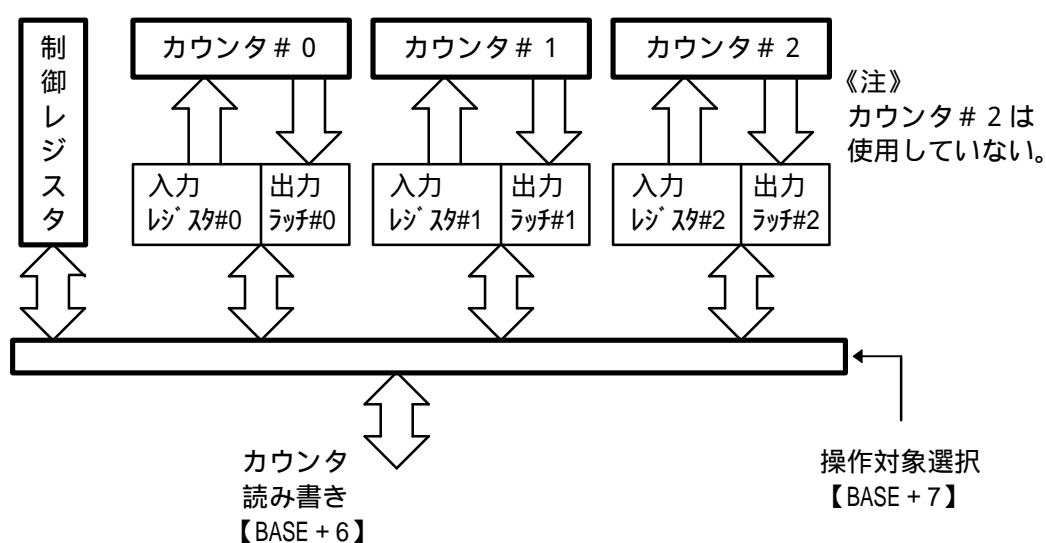


カウンタ素子の構造

各カウンタ素子（ $\mu\text{PD71054} = \text{i8253A}$ 上位互換）の構造を図3-10に記します。3本の各カウンタ（16BITバイナリ・ダウンカウンタ）は制御レジスタにより動作モードを指定されます。なお書き込まれるデータは実際には各カウンタ専用のレジスタに書き込み・保持されており、計数開始時等（動作モードに応じて）、必要なタイミングでカウンタ本体にロード（転送）されます。またデータをパソコン側から読み込むときは全カウンタを同時にラッチする構造となっています。

【3-11項・分周比（クロック値）の設定】および【3-12項・サンプリングデータ点数の設定】では当素子の特定動作モードのみを使用します。同素子自体の詳細については素子メーカー発行の仕様書等を御参照ください。（本ボードの制御には不要です。）

図3-10. $\mu\text{PD71054}$ （ i8253A 上位互換）の構造



3-11．クロック値（クロック源分周比）の設定

F I F Oバッファ出力モードで使用するクロック値は前3 - 9項で選択したクロック源を分周して得ます。分周は32 BITカウンタ（16 BITバイナリカウンタを2本直列接続）で行います。

各16 BITカウンタに設定できる値は1 ~ 65535で、
その設定値を【DV#0】，【DV#1】とすると、

分周比【DV】 = 【DV#0】 × 【DV#1】 (a)

得られるクロック周波数【CKF】 = 【クロック源周波数】 ÷ 【DV】 (b)

得られるクロック周期 【CKP】 = 【クロック源周期】 × 【DV】 (c)

となります。

《例》 クロック源10MHz（周期 = 100ns）から10μsのクロック値（周期）を得るには、

式(c)より、 【DV】 = (10μs) ÷ (100ns) = 100

式(a)から 【DV#0】 = 100 & 【DV#1】 = 1、
【DV#0】 = 50 & 【DV#1】 = 2、
【DV#0】 = 20 & 【DV#1】 = 5、等々。

各データはさらに、2バイトに分割して適用カウンタにかきこむ。

操作手順

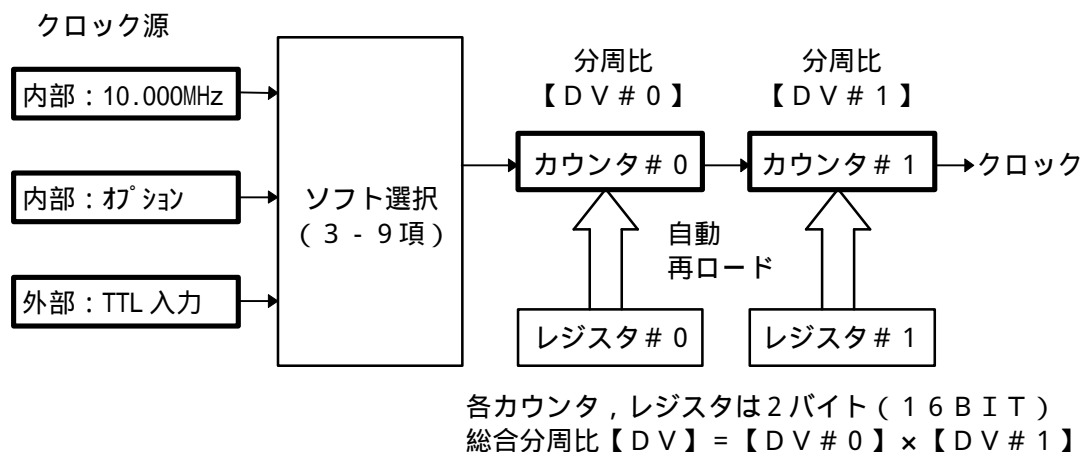
```
outp(BASE + 7, 0x3) ; /* 素子1の制御レジスタを対象に指定 */
outp(BASE + 6, 0x36) ; /* 制御データ【注2】：固定値 */
outp(BASE + 7, 0x0) ; /* 素子1のカウント#0を対象に指定 */
outp(BASE + 6, DV#0L) ; /* 下位バイト書き込み */
outp(BASE + 6, DV#0H) ; /* 上位バイト書き込み */
```

```
outp(BASE + 7, 0x3) ; /* 素子1の制御レジスタを対象に指定 */
outp(BASE + 6, 0x76) ; /* 制御データ【注3】：固定値 */
outp(BASE + 7, 0x1) ; /* 素子1のカウント#1を対象に指定 */
outp(BASE + 6, DV#1L) ; /* 下位バイト書き込み */
outp(BASE + 6, DV#1H) ; /* 上位バイト書き込み */
```

【注1】： 書き込み対象選択については前3 - 10項．参照。
また各カウンタの書き込みは下位バイトを先、続けて上位バイトの順とする。

【注2】 / 【注3】： カウント#0 / #1を“同素子のモード3”で使用する意味の固定データ。（無条件に適用してください。）

図 3 - 1 1 A . クロックの制御構造

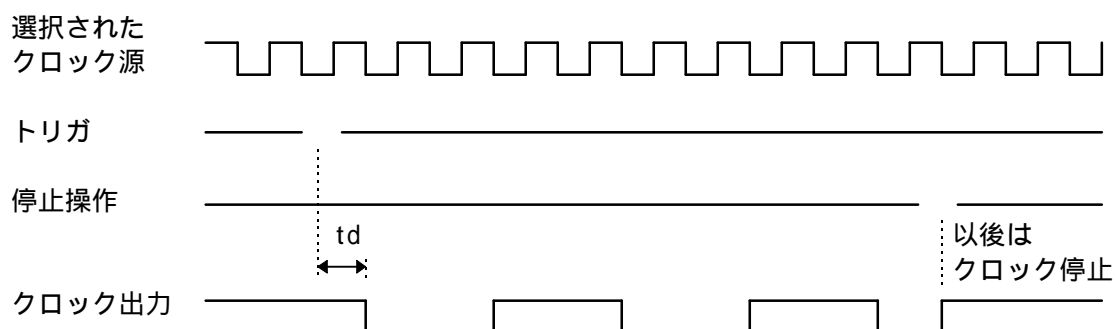


（3 - 9 項で）選択されたクロック源信号は“素子 1”内のカウンタ # 0，# 1 で分周されてクロック値となります。カウンタ # 0，# 1 は直列に接続された各 16 BIT のバイナリ DOWN カウンタで、当素子（ μ PD71054）のモード 3 で動作させます。

前記“操作手順”により書き込まれた分周比データは（実際には）、各カウンタ専用の入力レジスタ # 0，# 1 に保持されており、クロックスタート（トリガ発生）操作により各カウンタにロードされて計数が開始されます。総合 32 BIT の当カウンタが“0”までカウントダウンすると 1 クロックを出力し、次のカウント・タイミングで入力レジスタ内のデータが再ロードされて再びカウントダウンが続けられます。この動作の様子を図 3 - 1 1 B に示します。

【注 4】 クロックスタート（トリガ発生）時にも 1 クロック出力があります。

図 3 - 1 1 B . クロック発生タイミング



《動作説明》 上記はトリガ（クロックスタート＝出力開始）後、3 クロック出力後に停止操作等で動作終了となった例。分周比は 4。なお、

外部クロック源の有効エッジはソフト指定。（3 - 9 項）
クロック出力の有効エッジは常に立下り（ \downarrow ）。
最初のクロック出力遅れ t_d は最大で 1 クロック源周期。

3-12. サンプリングデータ点数の設定・認識

有限モード（3 - 13項）で出力するデータ点数は前3 - 11項で得たクロック出力をカウントして得ます。当モードによる出力データ点数は【1ブロック当りのデータ点数】を【指定ブロック数】または【停止操作等まで無限に】設定・制御する構造です。

ここではカウンタ素子2を使用し、

カウンタ#0に書き込む値W#0：1ブロック当りのデータ点数（2～65535）、
 カウンタ#1に書き込む値W#1：指定ブロック数（1～65535）として、
 指定データ点数【DC】=【W#0】×【W#1】となります。

《例》 指定データ点数【DC】=2000の場合、
 各16BITカウンタに設定する値は上式から、

【W#0】=500 & 【W#1】= 4、
 【W#0】=250 & 【W#1】= 8、
 【W#0】=125 & 【W#1】=16、等。

なお、サイクル出力動作時の【W#0】：1周期の点数、
 【W#1】：繰り返し回数（無限モード時は無視される）

《特注》 F I F Oメモリ内のデータ点数を1周期として循環出力する
 サイクルモードのときは、F I F Oメモリ内の実在データ点数
 とカウンタ#0の値（=1ブロック当りのデータ点数）が一致
 しなければなりません。

設定操作手順

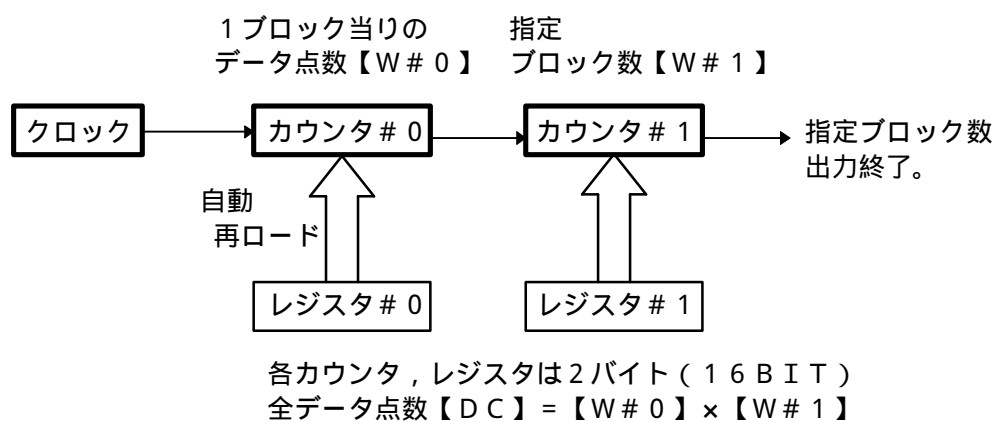
```
o u t p ( B A S E + 7 , 0 x 7 ) ; /* 素子2の制御レジスタを対象に指定 */
o u t p ( B A S E + 6 , 0 x 3 4 ) ; /* 制御データ【注2】：固定値 */
o u t p ( B A S E + 7 , 0 x 4 ) ; /* 素子2のカウンタ#0を対象に指定 */
o u t p ( B A S E + 6 , W # 0 L ) ; /* 下位バイト書き込み */
o u t p ( B A S E + 6 , W # 0 H ) ; /* 上位バイト書き込み */

o u t p ( B A S E + 7 , 0 x 7 ) ; /* 素子2の制御レジスタを対象に指定 */
o u t p ( B A S E + 6 , 0 x 7 2 ) ; /* 制御データ【注3】：固定値 */
o u t p ( B A S E + 7 , 0 x 5 ) ; /* 素子2のカウンタ#1を対象に指定 */
o u t p ( B A S E + 6 , W # 1 L ) ; /* 下位バイト書き込み */
o u t p ( B A S E + 6 , W # 1 H ) ; /* 上位バイト書き込み */
```

【注1】： 書き込み対象選択については前3 - 10項、参照。
 また各カウンタの書き込みは下位バイトを先、続けて上位バイトの順とする。

【注2】/【注3】：カウンタ#0/#1を同素子の“モード2”/“モード1”で使用
 する意味の固定データ。（無条件に適用してください。）

図 3 - 1 2 A . 出力データ点数の制御構造



（3 - 1 1項で）得たクロック信号はサンプルング動作に使用されると同時に“素子2”内のカウンタ#0、#1でサンプルング実行点数としてカウントされます。

カウンタ#0、#1は直列に接続された各16BITのバイナリDOWNカウンタで、当素子（μPD71054）のモード2またはモード1で動作させます。

前記“操作手順”により書き込まれた各データは（実際には）各カウンタ専用の入力レジスタ#0、#1に保持されており、クロックスタート（トリガ発生）操作により各カウンタにロードされて計数が開始されます。【1ブロック当りのデータ点数カウンタ】#0が“0”までカウントダウンすると1パルスを出力し、次のカウント・タイミングで入力レジスタ内のデータが再ロードされて再びカウントダウンが続けられます。

カウンタ#0からのパルス出力はボード内の制御タイミングに利用されると同時に【指定ブロック数カウンタ】#1の入力ともなります。#1のダウンカウント値が“0”に達すると全データ出力終了なのでクロックを止めます。

なお、#1の出力を無視して【停止操作等まで無限に】クロック動作を続ける無限動作モードもあります。（3 - 1 3項、参照）

出力済みデータ点数の読み込み

F I F Oバッファ出力モードで動作中、素子2のカウンタ# 0 , # 1を読むことにより現在までの出力済み点数を知ることができます。

動作開始時のカウンタ# 0 設定値【W# 0】：1ブロック当りのデータ点数
カウンタ# 1 設定値【W# 1】：指定ブロック数

現在のカウンタ# 0 読み込み値【R# 0】
1 読み込み値【R# 1】

とすると
出力済み点数は、

$$【DON】 = (【W# 1】 - 【R# 1】) \times 【W# 0】 + (【W# 0】 - 【R# 0】)$$

読み込み手順

```

outp ( BASE + 7 , 0 x 7 )      ; /* 素子2の制御レジスタを対象に指定 */
outp ( BASE + 6 , 0 x D 6 )    ; /* 制御データ【注2】：固定値      */

outp ( BASE + 7 , 0 x 4 )      ; /* 素子2のカウンタ# 0を対象に指定 */
R# 0 L = inp ( BASE + 6 )      ; /* 下位バイト読み込み */
R# 0 H = inp ( BASE + 6 )      ; /* 上位バイト読み込み */

outp ( BASE + 7 , 0 x 5 )      ; /* 素子2のカウンタ# 1を対象に指定 */
R# 1 L = inp ( BASE + 6 )      ; /* 下位バイト読み込み */
R# 1 H = inp ( BASE + 6 )      ; /* 上位バイト読み込み */

```

【注1】：読み込み対象選択については前3 - 10項．参照。
また各カウンタの読み込みは下位バイトを先、続けて上位バイトの順とする。

【注2】：カウンタ# 0 / # 1の値を同時ラッチ（出力用ラッチに読み出し）する意味の固定データ。（無条件に適用してください。） 当操作後は必ず続けてカウンタ（ラッチ）の読み込み操作を行うこと。 各ラッチは上位バイト読み出し直後から再びカウンタに追従を開始します。