

# 高速 8 ビットマイクロコントローラ KL5C80A16CFP ハードウェアマニュアル

1. 概要
  2. ブロック図
  3. 端子説明
  4. 外部バス・インターフェース・ユニット
  5. KC82 CPU
  6. 割込みコントローラ
  7. DMA コントローラ
  8. UART(非同期シリアルインターフェース)
  9. タイマ/カウンタ
  10. クロック同期シリアル I/O
  11. パラレルポート
  12. 動作モード設定
  13. アドレスマッピング
  14. 発振回路
  15. 電気的特性
  16. 外形寸法図
- 付録 A インストラクション表

1995. 7 Ver. 1.0  
川崎製鉄株式会社  
LSI 事業部

100  
100

## 1. 概要

KL5C80A16 は最新の CMOS プロセスを駆使して開発された高速 8 ビットマイクロコントローラです。心臓部の CPU コアに Zilog 社の Z80 とバイナリ・レベルでコンパチブルで、アドレス空間を拡張する MMU を内蔵した KC82 を採用しています。KC82 は Z80(10MHz) の約 4 倍の超高速性能を持ち、今までの 8 ビットマイコンの常識を越える性能を発揮する新世代の CPU コアです。

KL5C80A16 は CPU コア以外にも高速 DMA コントローラ、割り込みコントローラ、16 ビット高機能タイマ/カウンタ、非同期シリアルインターフェース (UART)、クロック同期シリアル I/O、パラレル・ポート 32 本、DRAM コントローラを内蔵しており、高性能かつ小型化が要求されるシステムに最適です。また、KL5C80A16 は低消費電力で、携帯装置への応用にも適しています。

## 特長

- ・ Z80 とバイナリ・コンパチブル
- ・ アドレス空間を 1M バイトに拡張する MMU
- ・ 高速 DMA コントローラ、2 チャンネル
- ・ UART (非同期シリアルインターフェース)、2 チャンネル
- ・ クロック同期シリアル I/O、2 チャンネル
- ・ 16 ビット高機能タイマ/カウンタ、4 本
- ・ 割り込み 内部 外部 16 本 (優先順位設定可能)、ノン・マスクブル割り込み 1 本
- ・ 32 本のパラレル・ポート
- ・ DRAM チップを直接接続可能にした DRAM コントローラ
- ・ 外部メモリ・チップ・セレクト回路内蔵
- ・ 水晶発振バッファ搭載
- ・ 最高動作周波数 10 MHz
- ・ 低消費電力

KL5C80A16 の CPU コア (KC82) の内部バスはクロック同期型のバスを採用しています。KL5C80A16 の内部 I/O はこの KC82 のクロック同期型バスに対応したバス・インターフェースを持っており、そのまま接続されています。KL5C80A16 の外部に I/O やメモリを接続する場合は、このままでは接続できませんので KC82 のクロック同期型バスを通常のメモリや I/O が接続できる非同期バスに変換する必要があります。KL5C80A16 にはこの変換を行う回路 (外部バス・インターフェース・ユニット) が搭載されており、KL5C80A16 外部に出力されているリード/ライト信号 (EMRD<sub>1</sub>, EMWR<sub>1</sub>, EIORD<sub>1</sub>, EIOWR<sub>1</sub>) は外部非同期バスに変換されたリード/ライト信号です。この外部バスサイクルのリード/ライト信号は内部バスサイクルのクロック同期バスのリード/ライト信号と異なり、外部の通常のメモリを直接接続できるリード/ライト用のストロブ型の信号となっているため使いやすくなっています。本データシートでは KC82 のクロック同期型バスを内部バスサイクル、外部バス・インターフェース・ユニットにより変換された外部非同期バスを外部バスサイクルと呼んでいます。

CPU の章の図はすべて内部バスサイクルを基本として描かれています。すなわち、KL5C80A16 内部の KC82 の動作説明という形で記述されています。あらかじめ御了承ください。したがって、CPU の章にある説明の図と同様な動作で外部メモリや外部 I/O をアクセスする場合には外部ウェイト入力あるいはウェイト・ステート・コントローラによるウェイトが挿入されることがあります。外部バスサイクルに関しては外部バス・インターフェース・ユニットの章を参照下さい。

また、外部のメモリを効率良く接続するため外部バス・インターフェース・ユニットはウェイト・ステート・コントローラを内蔵しています。これはシステム・コントロール・レジスタの設定により、外部メモリアクセスや外部 I/O アクセス時に自動的に設定に応じたウェイトを挿入しながら、外部バス・タイミング・チャートの外部メモリアクセス、外部 I/O アクセスを発生させるものです。このウェイト・ステート・コントローラは本チップの内部物理アドレス空間 1M byte を 2 つに分割して制御しており、高速 SRAM と EPROM といったアクセス・タイムの違う 2 つのメモリを効率良く接続できるように工夫されています。また外部バス・インターフェース・ユニットは DRAM コントローラ回路を内蔵しています。この DRAM コントローラはリフレッシュカウンタ、タイミング回路、ロウ/カラムアドレスマルチプレクサから構成されており、DRAM を直接接続することが可能でシステムをコンパクトに構成できます。アドレス空間の一部をバンク切替することで大容量の DRAM も対応しています。

システムコントロールレジスタの他にモード設定用の外部入力端子 BFMOD 端子があります。BFMOD 端子に “H” を入力し、外部端子 BFSIO にバグ・ファインダ・アダプタを接続することで簡易型デバッグツール (バグ・ファインダ) がリセット直後に立ち上がります。このとき外部の ROM の代わりに RAM を接続しますと ICE のようにパソコンから RAM ヘブプログラムのダウンロードができ、この RAM 上でシングル・ステップ実行等のデバッグ作業が可能です。バグ・ファインダ・アダプタは複数のベンダーから発売されています。問い合わせ先等は弊社営業部まで問い合わせ下さい。

### 2. ブロック図

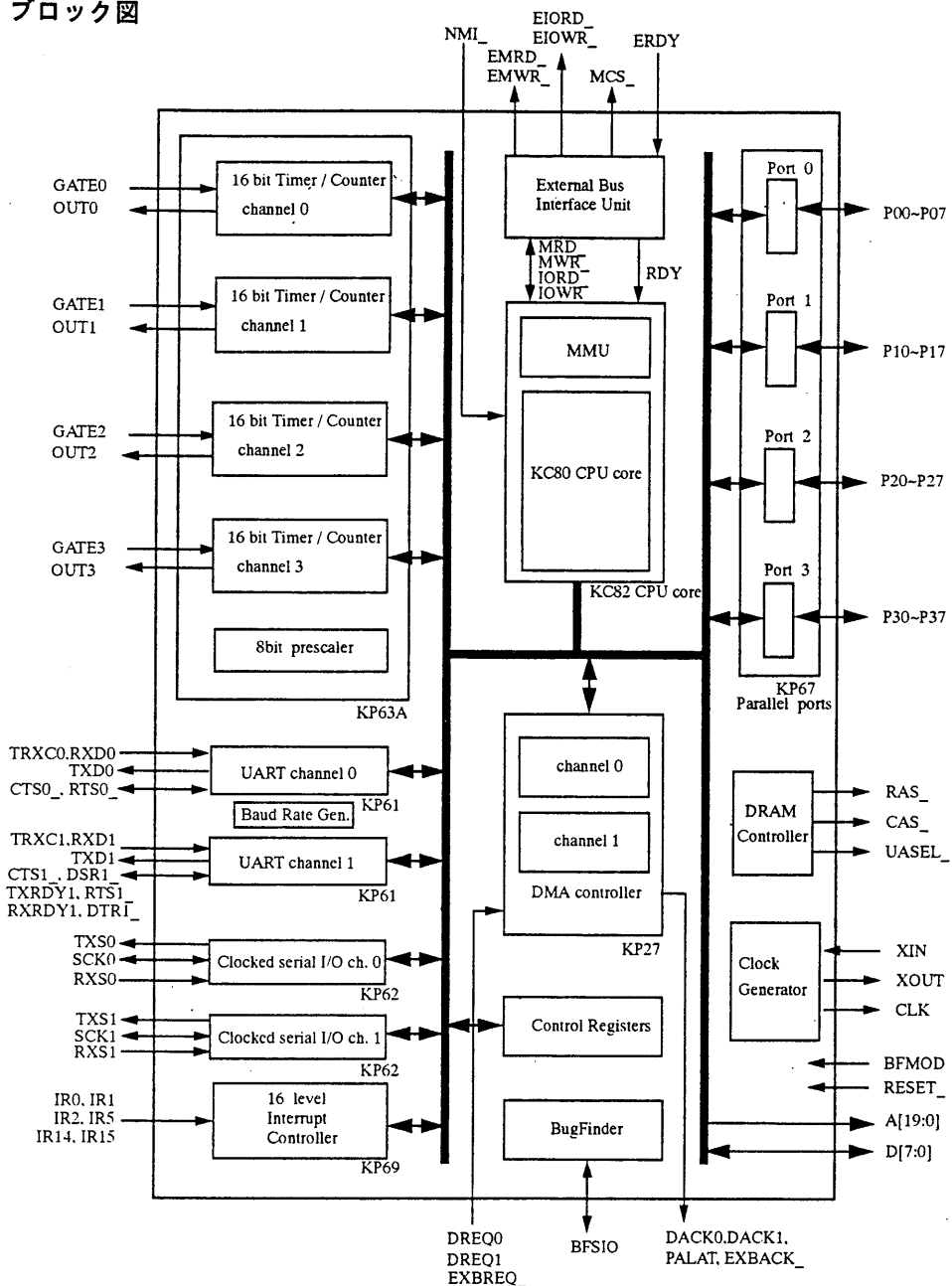


図 2-1 KL5C80A16 のブロック図

### 3. 端子説明

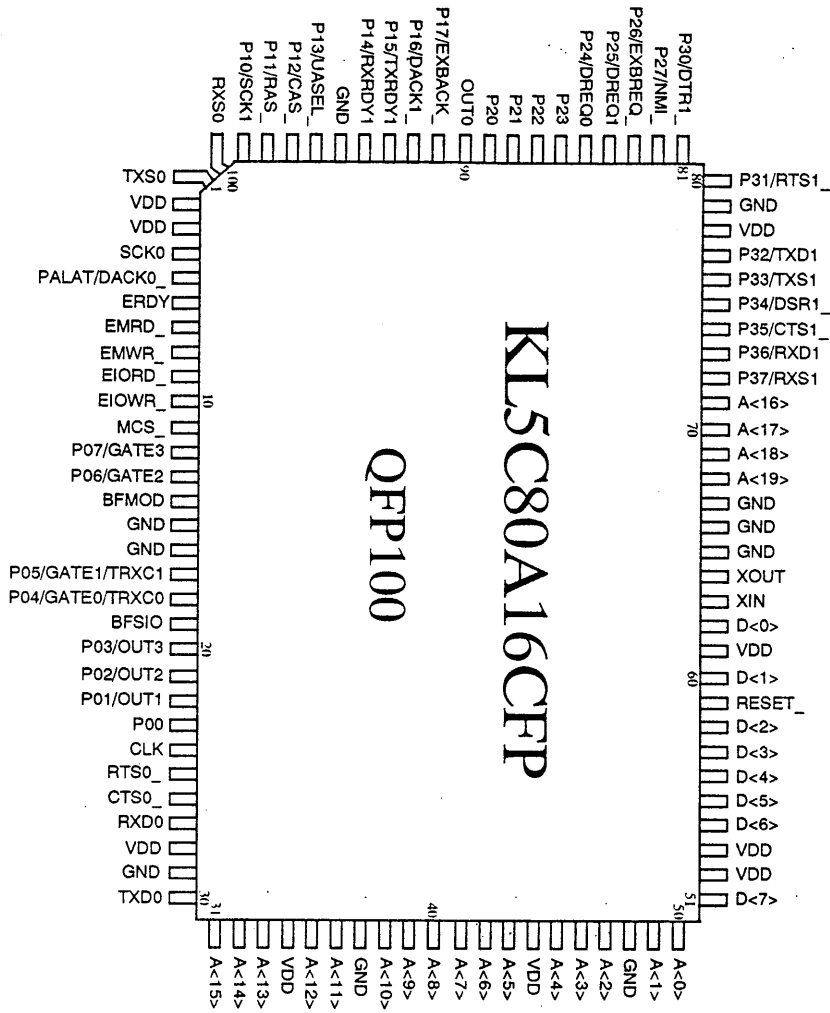


図 3-1 ピン配置図 (上面図)

KL5C80A16の一部の端子はパラレルポート端子とマルチプレクスされる端子があります。パラレルポート端子とどの端子がマルチプレクスされるかは前ページのピン配置図を参照して下さい。

表 3-1 端子説明

端子名	I/O	機能説明
GND	電源	0V を接続して下さい。
VDD	電源	電源を接続して下さい。
EMRD_	O	外部メモリリード信号。KC82 の内部バス信号 MRD_ を外部バス・インターフェース・ユニットで外部バス用に変換した信号です。
EMWR_	O	外部メモリライト信号。KC82 の内部バス信号 MWR_ を外部バス・インターフェース・ユニットで外部バス用に変換した信号です。
EIORD_	O	外部 I/O デバイスリード信号。KC82 の内部バス信号 IORD_ を外部バス・インターフェース・ユニットで外部バス用に変換した信号です。内部 I/O アクセスのときは“H”のままです。
EIOWR_	O	外部 I/O デバイスライト信号。KC82 の内部バス信号 IOWR_ を外部バス・インターフェース・ユニットで外部バス用に変換した信号です。内部 I/O アクセスのときは“H”のままです。
CLK	O	水晶発振子で発生した内部クロックをチップ外へ出力します。
BFSIO	I/O	デバッグ・ツールのバグ・ファインダ専用ポートです。バグ・ファインダを利用しないときは“H”を入力して下さい。
BFMOD	I	モード設定用の入力端子。KL5C80A16 の動作モードを設定します。
ERDY	I	外部からのウェイト要求入力。
A[19:0]	O	アドレス出力。CPU の A[19:0] が接続されています。
D[7:0]	I/O	外部データバス。
RESET_	I	リセット入力。“L”で KL5C80A16 はリセットされます。

端子名	I/O	機能説明
NMI_	I	CPU のノンマスクابل割り込み入力端子です。 立ち下がりエッジ検知。このノンマスクابل割り込み要求の優先順位は、INT_ よりも高く、BREQ_ よりも低くなっています。現在実行中の命令処理終了後、割り込み許可フラグの状態に関係なく、0066H 番地から始まる割り込み処理プログラムを実行します。内部 DMA とのバスリクエスト優先順位については7章を参照して下さい。
XIN	水晶発振 子用端子	内蔵のクロック発生回路に外部の水晶発振子を接続します。周波数はシステムクロックの2倍のものを使用して下さい。水晶発振子を使用しない場合はシステムクロックの2倍の周波数のクロックを入力して下さい。
XOUT	水晶発振 子用端子	内蔵のクロック発生回路に外部の水晶発振子を接続します。周波数はシステムクロックの2倍のものを使用して下さい。
RAS_	O	DRAM コントローラ RAS_ 出力 外部に DRAM を接続する場合、DRAM の RAS_ に接続します。詳細は4章を参照して下さい。
CAS_	O	DRAM コントローラ CAS_ 出力 外部に DRAM を接続する場合、DRAM の CAS_ に接続します。詳細は4章を参照して下さい。
UASEL_	O	DRAM コントローラ上位アドレス選択出力 この出力が“L”のときアドレスバスの下位側がロウアドレスを出力していることを示します。ICE を使用する時に使います。
MCS_	O	外部 RAM チップセレクト出力 アドレス = C0000H~FFFFFH で“L”となります。
GATE3~0	I	タイマ/カウンタチャンネル3~0へのGATE入力 詳細は9章を参照して下さい。
OUT3~0	O	タイマ/カウンタチャンネル3~0のOUT出力 詳細は9章を参照して下さい。
DREQ0 DREQ1	I	DMA 要求信号入力端子 各チャンネルに対する DMA 要求信号を入力します。有効極性は H/L 設定可です。詳細は7章を参照して下さい。

端子名	I/O	機能説明
DACK0_ DACK1_	O	DMA 要求アクノリッジ信号出力端子 DREQ を受けて、DMA 対象デバイスに対するアクノリッジ信号を出力します。“L”で DMA 要求承認となります。詳細は7章を参照して下さい。
EXBREQ_	I	外部デバイスバス権要求信号入力端子 外部に DMA 機能を持ったデバイスを接続する場合、そのバス権要求信号を入力します。”L”アクティブです。詳細は7章を参照して下さい。
EXBACK_	O	外部デバイスバス権要求アクノリッジ信号出力端子 外部に DMA 機能を持ったデバイスを接続する場合、そのデバイスのバス権要求アクノリッジ信号入力端子に接続します。詳細は7章を参照して下さい。
PALAT	O	I/O アドレス等ラッチ信号出力端子 詳細は7章を参照して下さい。
TRXC0 TRXC1	I	送受信クロック入力端子 データ送受信の際のボーレートを制御するクロックを入力する端子です。詳細は8章を参照して下さい。
TXD0 TXD1	O	送信データ出力端子 送信データをシリアルに出力する端子です。詳細は8章を参照して下さい。
RXD0 RXD1	I	受信データ入力端子 外部からのシリアルデータを入力する端子です。詳細は8章を参照して下さい。
CTS0_ CTS1_	I	クリアトゥセンド信号入力端子 詳細は8章を参照して下さい。
DSR1_	I	データセットレディ信号入力端子 詳細は8章を参照して下さい。
RTS0_ RTS1_	O	リクエストトゥセンド信号出力端子 詳細は8章を参照して下さい。
DTR1_	O	データターミナルレディ信号出力端子 詳細は8章を参照して下さい。

端子名	I/O	機能説明
TXRDY1	O	送信レディ信号出力端子 詳細は8章を参照して下さい。
RXRDY1	O	受信レディ信号出力端子 詳細は8章を参照して下さい。
SCK0 SCK1	I/O	クロック同期シリアルI/Oのシリアルクロック入出力端子です。 詳細は10章を参照して下さい。
RXS0 RXS1	I	クロック同期シリアルI/Oの受信データ入力端子です。 詳細は10章を参照して下さい。
TXS0 TXS1	O	クロック同期シリアルI/Oの送信データ出力端子です。 詳細は10章を参照して下さい。
P00~P07	I/O	パラレルポートのP0ポートです。詳細は11章を参照して下さい。
P10~P17	I/O	パラレルポートのP1ポートです。詳細は11章を参照して下さい。
P20~P27	I/O	パラレルポートのP2ポートです。詳細は11章を参照して下さい。
P30~P37	I/O	パラレルポートのP3ポートです。詳細は11章を参照して下さい。

端子名の0, 1, 2, 3はそれぞれチャンネル0、チャンネル1、チャンネル2、チャンネル3を表します。

BFMOD端子にはプルダウン抵抗が、またRESET\_端子にはプルアップ抵抗がついています。それらの特性(電流値)については、15章をご参照ください。

### 4. 外部バス・インターフェース・ユニット

#### 4.1 概要

外部バス・インターフェース・ユニットはウェイト・ステート・コントローラを内蔵しており、指定されたウェイトを挿入しながら CPU の内部バスサイクルを外部バスサイクルに変換する回路です。ウェイト

数の指定は SCR4 のビット 7～4 で行ないます。このユニットは外部 I/O や外部メモリのアクセスのときのみ動作し、外部バスのリード/ライト信号は内部 I/O のアクセス時には“H”のままです。SCR4 はリセット時は‘00H’になります。

SCR4(I/O アドレス = 1FH)

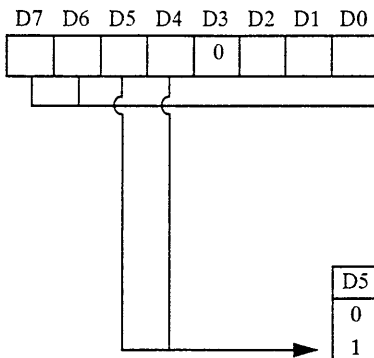


表 4-1 外部 I/O・ウェイト・コントロール

D7	D6	外部I/O
0	0	1 wait (3 clock/バスサイクル)
0	1	2 wait (4 clock/バスサイクル)
1	0	3 wait (5 clock/バスサイクル)
1	1	4 wait (6 clock/バスサイクル)

4waitの時、EIORD\_/EIOWR\_の開始エッジが1/2clock遅れます。

表 4-2 外部メモリ・ウェイト・コントロール

D5	D4	外部メモリ(00000H~7FFFFH)	外部メモリ(80000H~FFFFFFH)
0	X	1 wait	1 wait*
1	0	1 wait	0 wait*
1	1	0 wait	0 wait*

\*DRAM 使用時はこの指定には従いません。

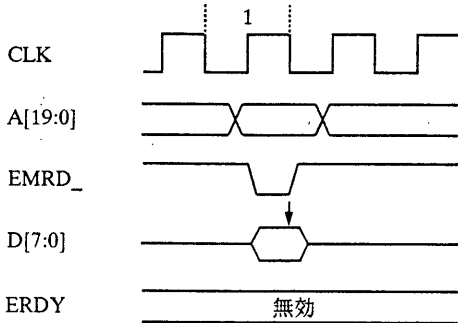
#### 4.2 CPU 外部バスサイクル

KL5C80A16 の CPU コア, KC82 のシステムバスはクロック同期型のバスを採用しています。内部 I/O はこの KC82 のクロック同期型バスに対応したバス・インターフェースを持っており、そのまま接続されています。KL5C80A16 の外部に I/O やメモリを接続する場合は、外部バス・インターフェース・ユニットが内部クロック同期バスのリード/ライト信号(MRD\_, MWR\_, IORD\_, IOWR\_)を次ページ以降のバス・タイミング・チャートにある外部非同期バスのリード/ライト信号(EMRD\_, EMWR\_, EIORD\_, EIOWR\_)に変換します。外部バス・インターフェース・ユニットはウェイト・ステート・コントローラを内蔵しており、指定されたウェイトを挿入しながら CPU の内部バスサイクルを外部バスサイクルに変換します。ウェイト数の指定は SCR4 のビット 7~4 で行ないます。外部バス・インターフェース・ユニット

は外部ウェイト入力端子(ERDY)から入力された外部ウェイト要求に内蔵のウェイト・ステート・コントローラからのウェイト要求を加えて CPU に渡します。

ウェイト・ステート・コントローラは KL5C80A16 の物理アドレス空間 1M byte を 2 つに分割して制御しており、高速 SRAM と中速 EPROM といったアクセス・タイムの違う 2 つのメモリを効率良く接続できるように工夫されています。また、0 ウェイト・外部メモリアクセスでは外部ウェイト信号(ERDY)は無視されますので注意が必要です。外部バス・インターフェース・ユニットは DRAM コントローラも内蔵しており、高速 SRAM、DRAM、中速 EPROM といった 3 つのアクセスタイムの異なるメモリを直接接続できます。内蔵 DRAM コントローラによる DRAM アクセスは SCR4 の設定に関係なく常に 3 クロックサイクルで実行されます。詳しくは 4.4 節を参照して下さい。

### 4.3 CPU 外部バスサイクル



メモリアクセスのウェイト数の設定は 0 ウェイトと 1 ウェイトですが、0 ウェイトの設定では ERDY 入力が無視されますので御注意下さい。またリセット時にメモリアクセスのウェイト数の設定は 1 ウェイトに初期化されます。

図 4-1 外部メモリリード(ウェイト・ステート・コントローラを 0 ウェイトに設定した時)

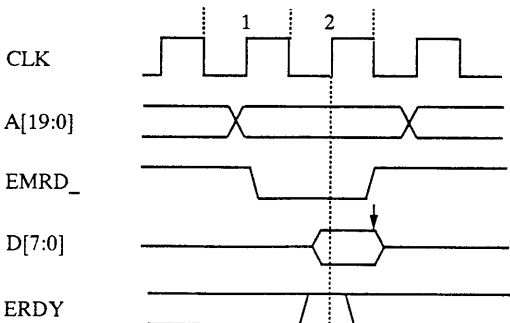
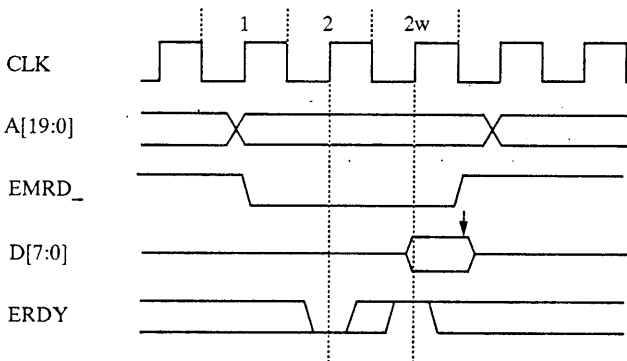
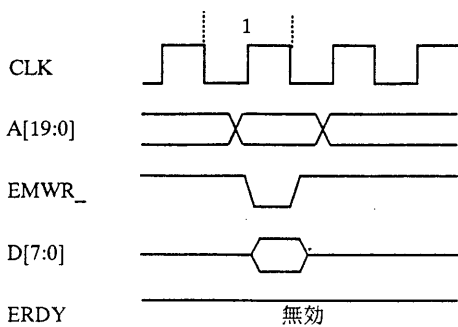


図 4-2 外部メモリリード(ウェイト・ステート・コントローラを 1 ウェイトに設定した時)



2w: 外部ウェイト要求によるウェイトサイクル

図 4-3 外部メモリリード(ウェイト・ステート・コントローラを 1 ウェイトに設定し、1 ウェイト追加した時)



メモリアクセスのウェイト数の設定は 0 ウェイトと 1 ウェイトですが、0ウェイトの設定ではERDY入力が無視されますので御注意下さい。またリセット時にメモリアクセスのウェイト数の設定は 1 ウェイトに初期化されます。

図 4-4 外部メモリライト (ウェイト・ステート・コントローラを 0 ウェイトに設定した時)

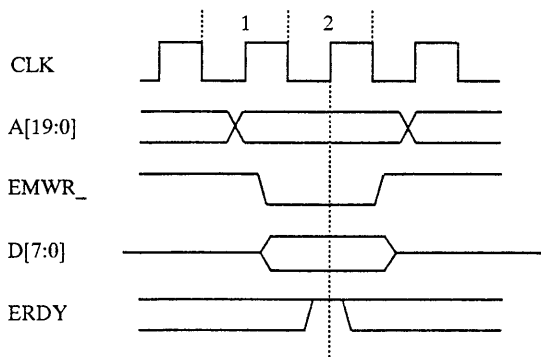
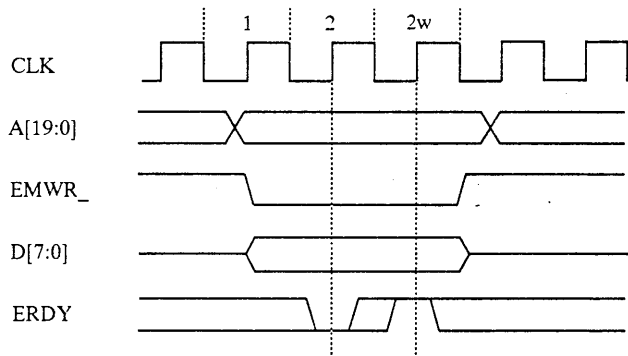
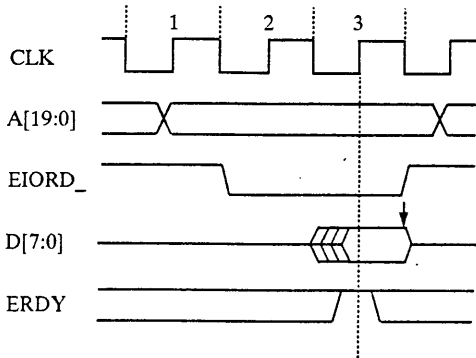


図 4-5 外部メモリライト (ウェイト・ステート・コントローラを 1 ウェイトに設定した時)



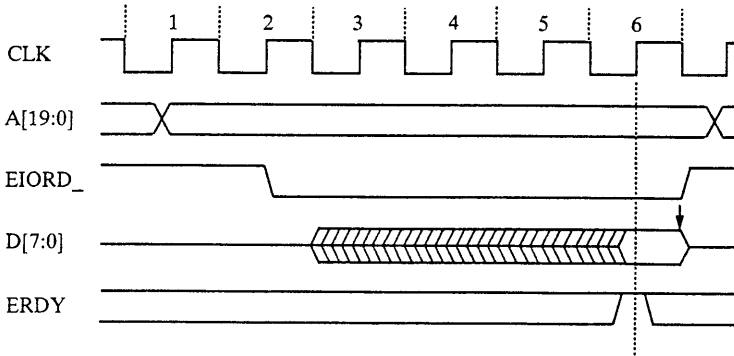
2w: 外部ウェイト要求によるウェイトサイクル

図 4-6 外部メモリライト (ウェイト・ステート・コントローラを 1 ウェイトに設定し、1 ウェイト追加した時)



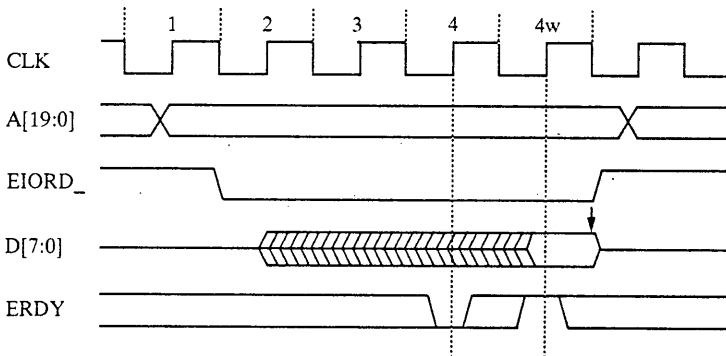
外部 I/O リードのウェイト設定は SCR4 で 1 ウェイト～ 4 ウェイトまで設定できます。1 ウェイトでは左の図にあるように 3 クロックで外部 I/O をリードします。2 ウェイト、3 ウェイトの場合はサイクル 2 の後にそれぞれ 1 クロック、2 クロックのウェイトサイクルが挿入されます。SCR の設定に関係なく内部 I/O アクセスは常に 0 ウェイトです。また、内部 I/O にリードする場合は EIORD\_ は “H” のままです。

図 4-7 外部 I/O リード (ウェイト・ステート・コントローラを 1 ウェイトに設定した時)



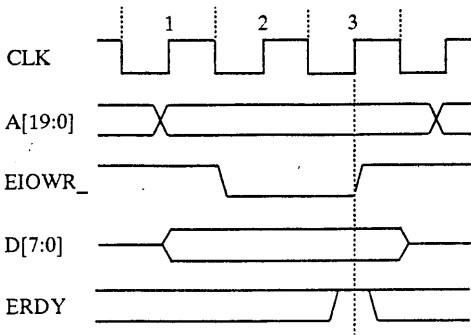
外部 I/O リードの 4 ウェイト設定は 1 ウェイト～ 3 ウェイト設定の外部 I/O リードと異なり、EIORD\_ の立ち下がりが半クロック遅れます。アドレスのセットアップの大きい外部 I/O デバイスを接続する場合に御利用下さい。

図 4-8 外部 I/O リード (ウェイト・ステート・コントローラを 4 ウェイトに設定した時)



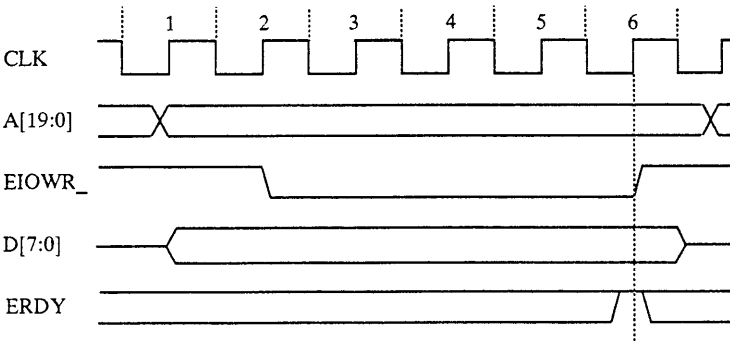
4w: 外部ウェイト要求によるウェイトサイクル

図 4-9 外部 I/O リード (ウェイト・ステート・コントローラを 2 ウェイトに設定し、1 ウェイト追加した時)



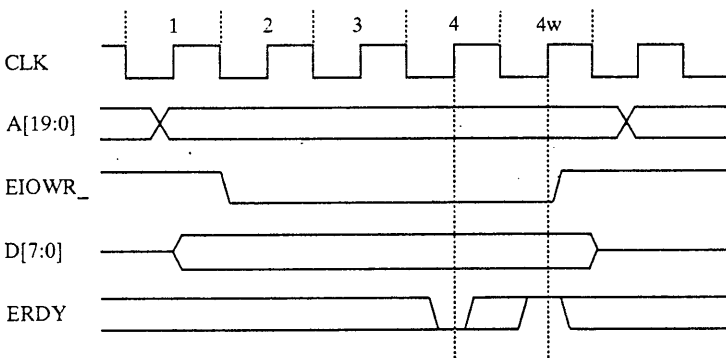
外部 I/O ライトのウェイト設定は SCR4 で 1 ウェイト～ 4 ウェイトまで設定できます。1 ウェイトでは左の図にあるように 3 クロックで外部 I/O をライトします。2 ウェイト、3 ウェイトの場合はサイクル 2 の後にそれぞれ 1 クロック、2 クロックのウェイトサイクルが挿入されます。SCR の設定に関係なく内部 I/O アクセスは常に 0 ウェイトです。また、内部 I/O にライトする場合は EIOWR\_ は “H” のままです。

図 4-10 外部 I/O ライト (ウェイト・ステート・コントローラを 1 ウェイトに設定した時)



外部 I/O ライトの 4 ウェイト設定は 1 ウェイト～ 3 ウェイト設定の外部 I/O ライトと異なり、EIOWR\_ の立ち下がりが半クロック遅れます。アドレスのセットアップの大きい外部 I/O デバイスを接続する場合はこれを御利用下さい。

図 4-11 外部 I/O ライト (ウェイト・ステート・コントローラを 4 ウェイトに設定した時)



4w: 外部ウェイト要求によるウェイトサイクル

図 4-12 外部 I/O ライト (ウェイト・ステート・コントローラを 2 ウェイトに設定し、1 ウェイト追加した時)

### 4.4 DRAMコントローラ・バス・タイミング

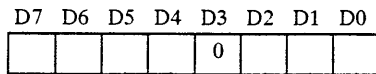
KL5C80A16はDRAMコントローラを内蔵しています。このDRAMコントローラはリフレッシュカウンタ、タイミング発生回路、バンク切り替え機能を持っており、KL5C80A16に大容量のDRAMを直接接続できます。本DRAMコントローラの設定はSCR3とSCR0で行ないます。DRAM用のアドレス領域は80000H～BFFFFHです。SCR3のD5の設定によりC0000H～FFFFFFHはDRAMの最上位バンクあるいはSRAM領域のどちらかが選択可能です。

このDRAMコントローラはDRAMのロウアドレスとカラムアドレスを切り替えるマルチプレクサを内蔵

しています。DRAMへのアドレスバスの接続は以下のようにして下さい。

- 256Kバイト(非拡張時) : A16, A7～A0をDRAMに接続してください。
- 1Mバイト拡張時 : A17,A16, A7～A0をDRAMに接続してください。
- 4Mバイト拡張時 : A18～A16, A7～A0をDRAMに接続してください。

SCR3 (I/O アドレス = 1EH)



Refresh Rate

D2	D1	D0	リフレッシュ要求発生間隔
0	0	0	128 clock 毎
0	0	1	88 clock 毎
0	1	0	64 clock 毎
0	1	1	44 clock 毎
1	0	0	32 clock 毎
1	0	1	22 clock 毎
1	1	0	16 clock 毎
1	1	1	11 clock 毎

D5= '1' のとき、C0000H～FFFFFFHを

0: SRAM 領域

1: DRAM 最上位バンク領域

として使用します。D5= '0' のときは意味を持ちません。

80000H～BFFFFHをDRAM領域として

0: 使用しません。

端子 96 は P13

端子 97 は P12

端子 98 は P11 として機能します。

1: 使用します。

端子 96 は UASEL<sub>-</sub>

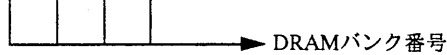
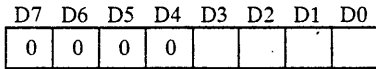
端子 97 は CAS<sub>-</sub>

端子 98 は RAS<sub>-</sub> として機能します。

(UASEL<sub>-</sub> は ICE 使用時に使用します。)

SCR0(I/Oアドレス = 1BH)

KL5C80A16 で DRAM 領域として使用できるのは 80000H~FFFFFFH のみですが、SCR0 でコントロールできるバンク切り換え機能により最大 4M バイトまで DRAM をマップできます。



80000H~BFFFFHにDRAMを割り付けてアドレス拡張を行なう場合、ここにバンク番号を設定します。

拡張するバンク数に応じて、LSBから必要なビット数 (D<1:0>またはD<3:0>)を使用し、0から最大15までのバンク番号を設定して下さい。例えば

1M バイト拡張時: 4M ビットの × 4 DRAM を 2 個使用して、D<1:0>の 2 ビットで 4 バンク

4M バイト拡張時: 16M ビットの × 4 DRAM を 2 個使用して、D<3:0>の 4 ビットで 16 バンク

を設定します。

図 4-13 に DRAM のリフレッシュサイクルを示します。KL5C80A16 の DRAM コントローラは CAS ビフォア RAS リフレッシュをサポートしています。リフレッシュサイクルは DRAM リード、DRAM ライト以外のバスサイクル (DRAM 以外のメモリアドレス、DRAM 以外のメモリアドレス、I/O リード、I/O ライト、CPU のアイドルサイクル) と並行して進行します。したがって、リフレッシュサイクル中のアドレス、EMRD<sub>0</sub>、EMWR<sub>0</sub>、EIORD<sub>0</sub>、EIOWR<sub>0</sub>、D[7:0]、ERDY の各状態はリフレッシュサイクルと並行して進行するバスサイクルによって異なります。また、リフレッシュ要求と DRAM リード

あるいは DRAM ライトサイクルが同時に発生した場合、リフレッシュサイクルが優先され CPU はウェイト状態になります。すでに DRAM リードあるいは DRAM ライトサイクルが進行中にリフレッシュ要求が発生した場合、実行中の DRAM リードあるいは DRAM ライトサイクルが終了した時点でリフレッシュサイクルが始まります。逆にリフレッシュサイクル実行中に DRAM リードあるいは DRAM ライトサイクルが発生した場合は、リフレッシュサイクル終了まで CPU はウェイト状態になります。

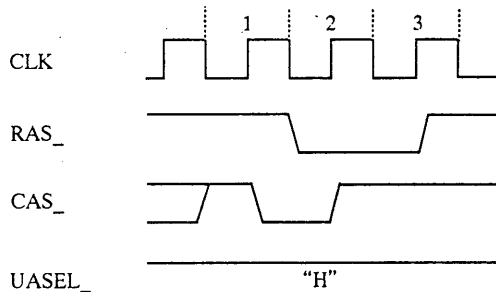
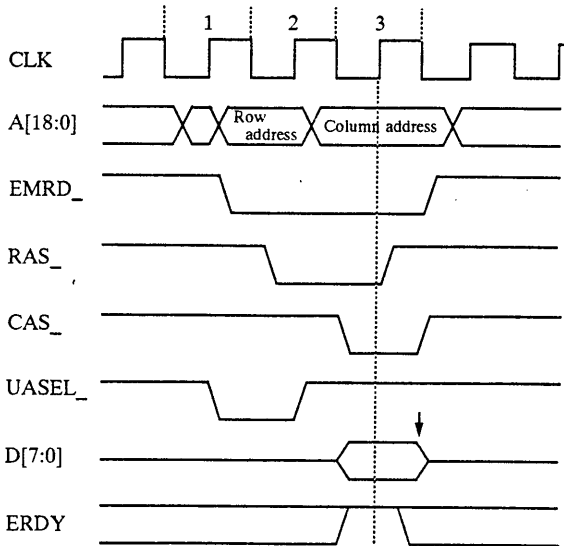
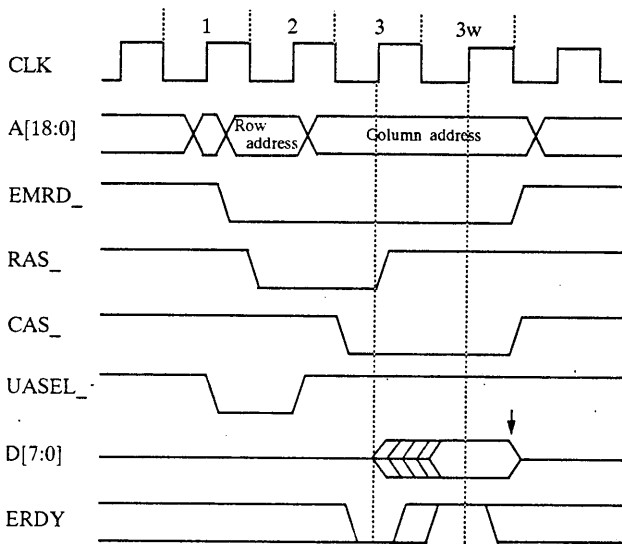


図 4-13 DRAM refresh



DRAM リードサイクルは SCR の設定 (メモリアクセスのウェイト設定) に関係なく 3 クロックのバスサイクルです。アクセスタイムが足りない場合は外部ウェイト要求を ERDY 端子から入れてください。ERDY 入力は SCR の設定に関係なく有効です。

図 4-14 DRAM リード (外部ウェイトを追加しない時)



3w: 外部ウェイト要求によるウェイトサイクル

図 4-15 DRAM リード (外部ウェイトを 1 ウェイト追加した時)

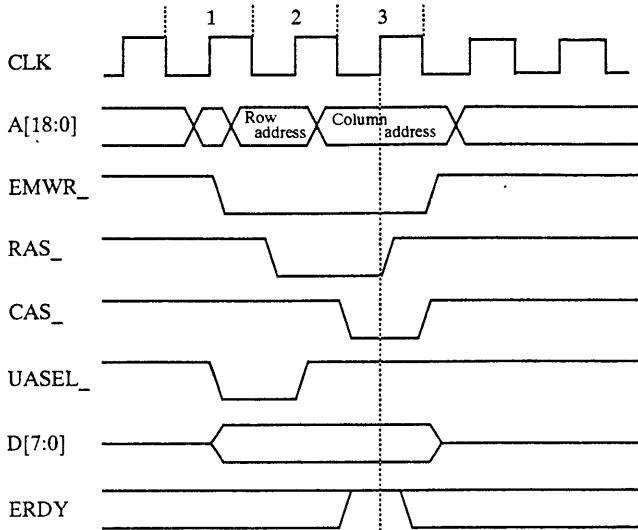
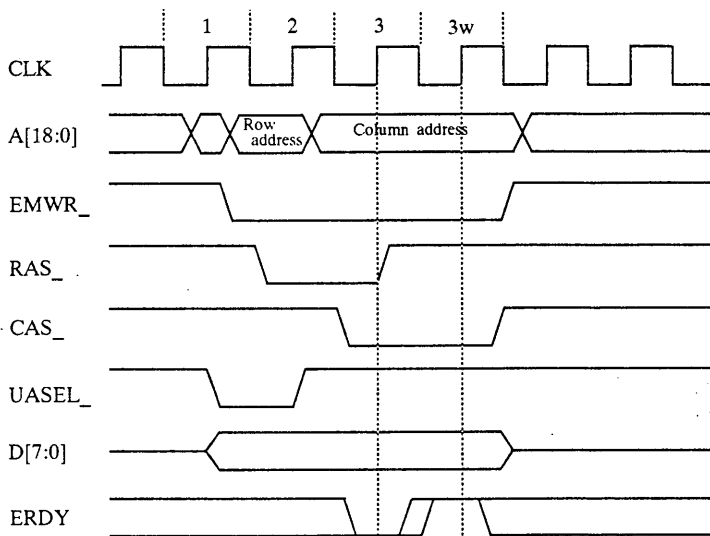


図 4-16 DRAM ライト (外部ウェイトを追加しない時)

DRAM ライトサイクルは SCR の設定 (メモリアクセスのウェイト設定) に関係なく 3 クロックのバスサイクルです。アクセスタイムが足りない場合は外部ウェイト要求を ERDY 端子から入れてください。ERDY 入力は SCR の設定に関係なく有効です。



3w: 外部ウェイト要求によるウェイトサイクル

図 4-17 DRAM ライト (外部ウェイトを 1 ウェイト追加した時)

### 4.5 DMA コントローラ・外部バス・タイミング

本外部バス・インターフェース・ユニットは内蔵 DMA コントローラの外部デバイスに対する DMA 転送サイクルも自動的に外部バスサイクルに変換します。DMA によるメモリメモリ転送は CPU の 2 つのメモリアクセスと同じ外部バスサイクルに変換されます。メモリメモリ転送のメモリアクセスタイミングに関しては、CPU と DRAM コントローラのメモリアクセスタイミングを参照下さい。DMA の動作について詳しくは 7 章を参照下さい。

#### DMA コントローラ・外部バスの基本動作

内蔵の DMA コントローラは DMA 転送のサイクル 1 で A19~A0 に I/O アドレスを出力します。サイクル 2 以降ではメモリアドレスを A19~A0 に出力しつつ、EMRD\_ と EIOWR\_ あるいは EMWR\_ と EIORD\_ を同時にアクティブにします。(内部 I/O とメモリの間で行なわれる DMA 転送では EIOWR\_ あるいは EIORD\_ は

アクティブになりません。) このときデータはメモリから I/O へあるいは I/O からメモリへ直接転送されます。このためメモリと外部 I/O デバイスとの間で DMA 転送を行なう際、メモリと外部 I/O デバイスを同時に選択する必要があります。メモリの選択方式は CPU による通常のメモリアクセスと変わりありませんが、DMA 転送における外部 I/O デバイスの指定方式は 2 つあります。一つはサイクル 1 のアドレスを PALAT 信号の立ち下がりにより外部回路でラッチし、これを外部の I/O アドレスとして外部 I/O デバイスに供給します。もう一つの方法は、DMA コントローラの DMA 要求アックノリッジ信号 (DACK0\_、DACK1\_ 出力) を使って外部 I/O デバイスのチップセレクトを直接選択する回路を外部に用意する方法です。

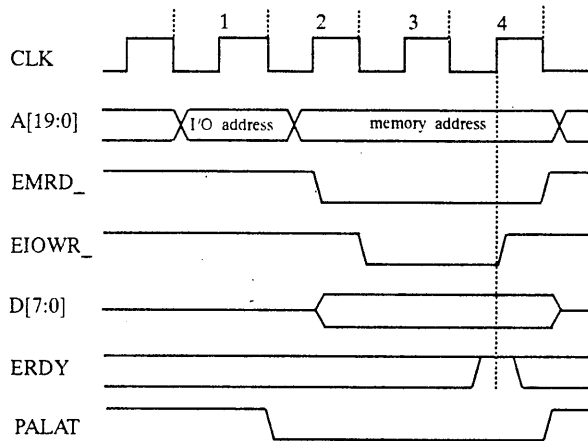


図 4-18 DMA 転送の例 (memory to 外部 I/O, メモリは 0 ウェイト、I/O は 1 ウェイト設定時)

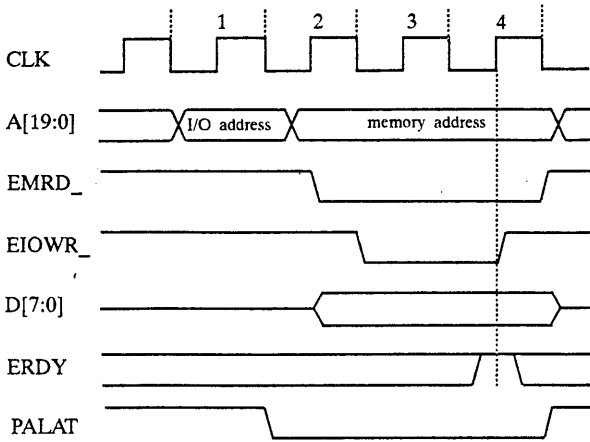


図 4-19 DMA 転送 (memory to 外部 I/O, メモリは 0 ウェイト、I/O は 1 ウェイト設定時)

DRAM以外のメモリから外部I/OへのDMA転送サイクルでは転送元のメモリと転送先のI/Oのウェイト数の設定によってDMA転送サイクルのクロック数が変わってきます。転送元のメモリが0ウェイト、転送先のI/Oが1ウェイトの設定では図4-19のように4クロックサイクルになります。転送元のメモリが1ウェイトの場合はサイクル2の終わりにウェイトサイクルが挿入されます。転送先のI/Oのウェイト設定が2、3、4ではサイクル3の終わりにそれぞれ1、2、3クロックのウェイトサイクルが挿入されます。図4-20の例では転送元のメモリ1ウェイト、転送先のI/Oの2ウェイトの設定でのDMA転送サイクルを示しています。

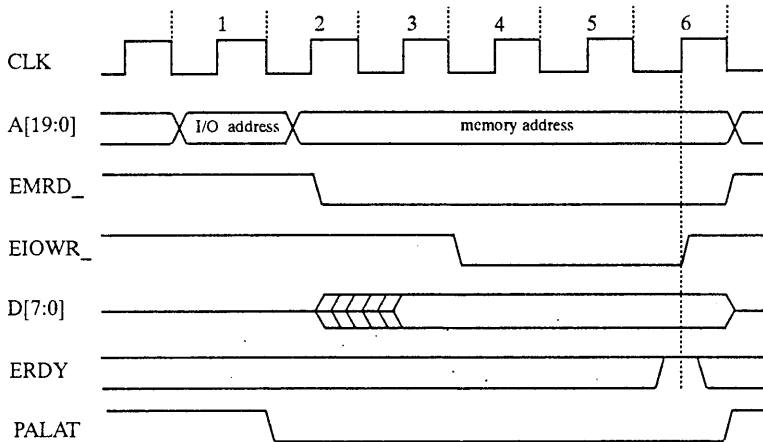


図 4-20 DMA 転送 (memory to 外部 I/O, メモリは 1 ウェイト、I/O は 2 ウェイト設定時)

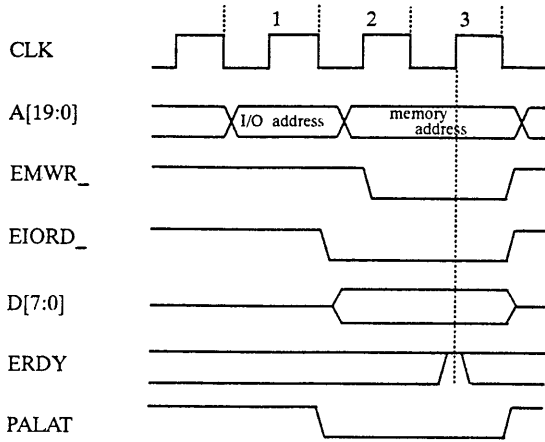


図 4-21 DMA 転送 (外部 I/O to memory, メモリは 0 ウェイト、I/O は 1 ウェイト設定時)

外部I/OからDRAM以外のメモリへのDMA転送サイクルではSCR4による転送元のI/Oと転送先のメモリのウェイト数の設定によってDMA転送サイクルのクロック数が変わってきます。転送元のI/Oが1ウェイト、転送先のメモリが0ウェイトの設定の場合、図4-21のように3クロックサイクルになります。転送元のI/Oのウェイト設定が2、3、4の場合、サイクル2の終わりにそれぞれ1、2、3クロックのウェイトサイクルが挿入されます。転送先のメモリが1ウェイトの場合はサイクル3の終わりに1クロックのウェイトサイクルが挿入されます。図4-22の例では転送元のI/Oの2ウェイト、転送先のメモリ1ウェイトの設定でのDMA転送サイクルを示しています。

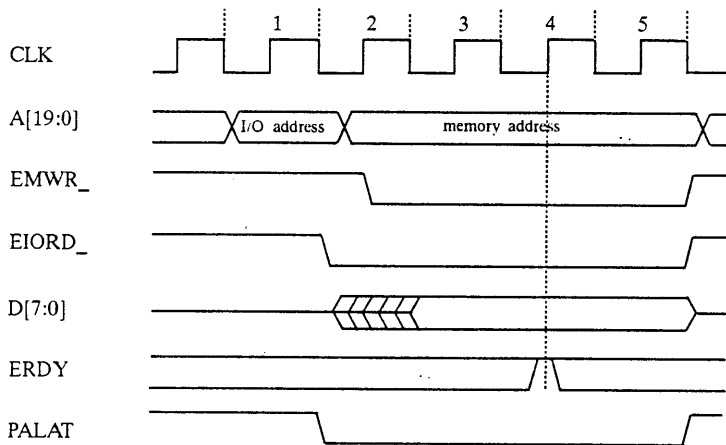
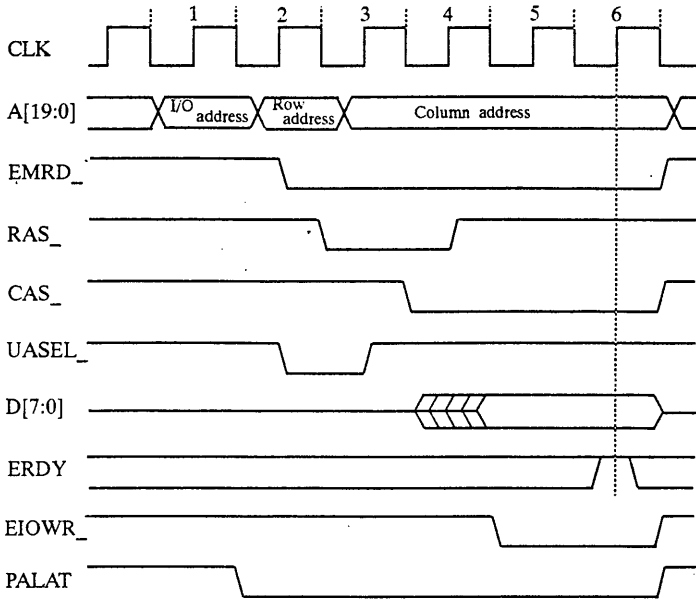
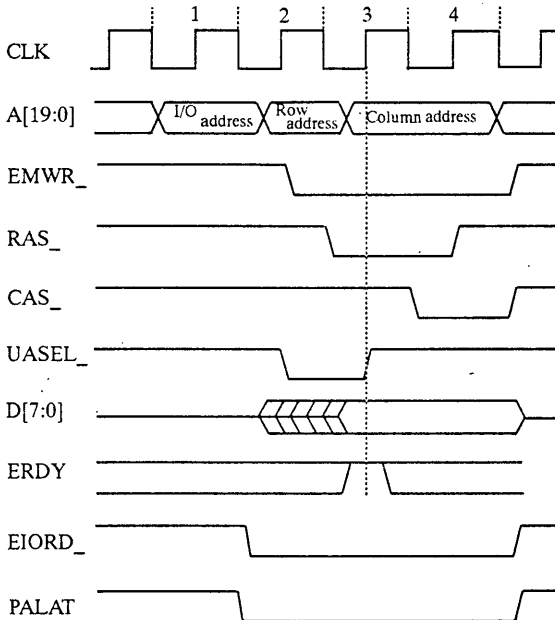


図 4-22 DMA 転送 (外部 I/O to memory, メモリは 1 ウェイト、I/O は 2 ウェイト設定時)



DRAMから外部I/OへのDMA転送サイクルではSCR4による転送先のI/Oのウェイト数の設定によってDMA転送サイクルのクロック数が変わってきます。転送先のI/Oが1ウェイトの設定の場合、図4-23のように6クロックサイクルになります。転送先のI/Oのウェイト設定が2、3、4の場合、サイクル5の終わりにそれぞれ1、2、3クロックのウェイトサイクルが挿入されます。転送元メモリがDRAMの場合、DMA転送サイクルのクロック数はメモリのウェイト数の設定には影響されません。

図4-23 DMA転送 (DRAM to 外部I/O, I/Oは1ウェイト, 外部ウェイトを追加しない時)



外部I/OからDRAMへのDMA転送サイクルではSCR4による転送元のI/Oのウェイト数の設定によってDMA転送サイクルのクロック数が変わってきます。転送元のI/Oが1ウェイトの設定の場合、図4-24のように4クロックサイクルになります。転送元のI/Oのウェイト設定が2、3、4の場合、サイクル3の終わりにそれぞれ1、2、3クロックのウェイトサイクルが挿入されます。転送先メモリがDRAMの場合、DMA転送サイクルのクロック数はメモリのウェイト数の設定には影響されません。

図4-24 DMA転送 (外部I/O to DRAM, I/Oは1ウェイト, 外部ウェイトを追加しない時)

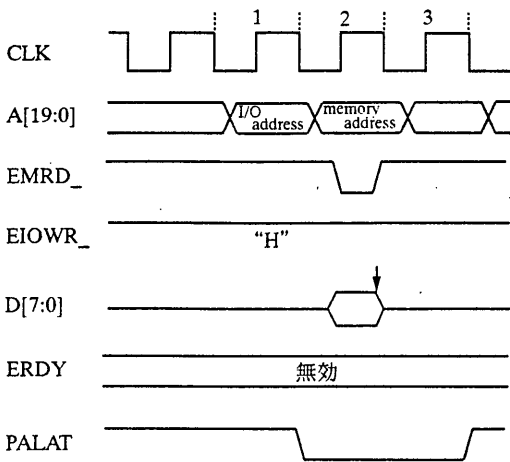


図 4-25 DMA 転送 (memory to 内部 I/O, メモリ 0 ウェイト設定時)

内部 I/O とメモリ間の DMA 転送サイクルは外部 I/O リードライト信号 (EIORD\_, EIOWR\_) が "H" の状態でメモリリードライト信号だけが有効になります。したがって外部には I/O アクセスサイクルは現われず、外部のデバイスには CPU のメモリアクセスサイクルだけが発生しているのと同じ状態になります。左の図の例ではそれぞれ 0 ウェイトのメモリリード、ライトの例を示していますが、転送先のメモリのウェイト設定や種類にあったバスサイクルが発生します。

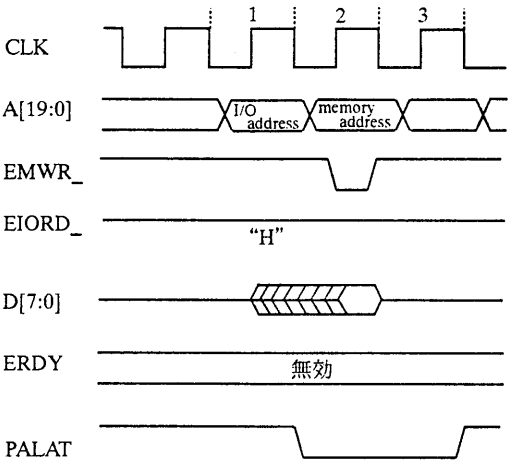


図 4-26 DMA 転送 (内部 I/O to memory, メモリ 0 ウェイト設定時)

### 5. KC82 CPU

#### 5.1 概要

KC82はKC80 CPU コアにMMUを搭載してアドレス空間を1Mバイトに拡張した新開発のCPUコアです。KC80はZilog社Z80とオブジェクトレベルでコンパチブルなCPUコアですが、内部回路は内部16bit高速アーキテクチャーの採用など最新のCMOS技術で新たに設計されており、オリジナルZ80の動作周波数に換算すると約40MHzの高性能を発揮し、かつ低消費電力になっています。本CPUコアは同期バスを採用しており、当社マクロセルであるKP6XシリーズやKP55、KP59、KP54、KP37、KP51と組み合わせることで、効率良くASICマイコン化することが可能です。KC82は非常に高性能で16bitマイコンを越える性能を発揮しますので従来の8bitCPUでは困難であった応用にも応えます。

#### 特徴

KC82は以下の特長を持っています。

1) Zilog社Z80に比較し、KC82は平均で約1/4のクロックサイクルで処理します。

例：命令	Z80	KC82
LD r, r'	4クロック	1クロック
ADD HL, ss	11クロック	1クロック

2) インターフェース用マクロを使用することにより、従来のZ80ペリフェラル・ファミリとの接続も可能です。

3) インストラクション・セットは158種類で、Zilog社Z80とオブジェクトコードレベルでフルコンパチブルです。

#### 5.2 ブロック図

図5-1にKC82の全体ブロック図を示します。

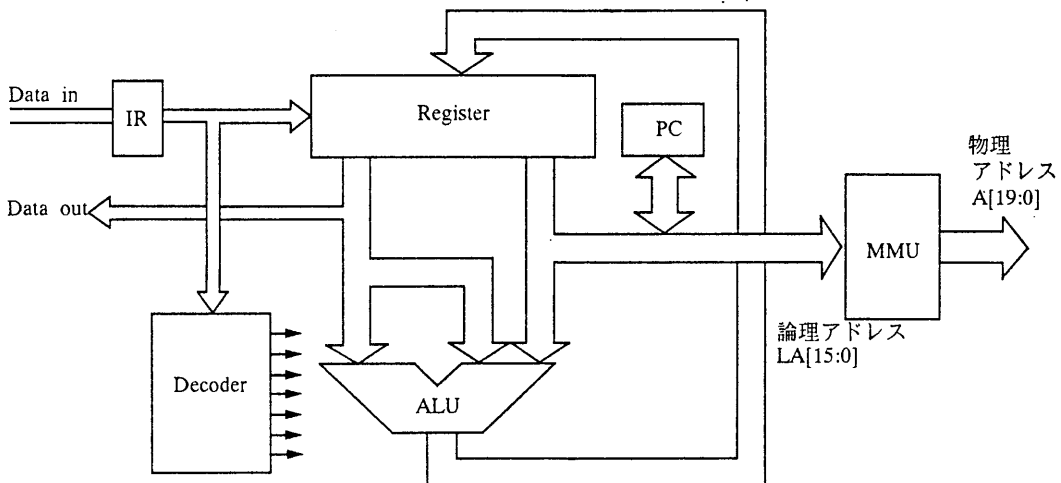


図 5-1 KC82 全体ブロック図

### 5.3 CPU 内部端子機能説明

端子名	I/O	機能説明
MRD_	O	内部バスメモリリード出力端子 アクティブ“L”。メモリからのデータを読み込む際に出力します。
MWR_	O	内部バスメモリライト出力端子 アクティブ“L”。メモリへデータを書き込む際に出力します。
IOR_	O	内部バスI/Oリード出力端子 アクティブ“L”。I/Oデバイスからのデータを読み込む際に出力します。最小2クロック期間アクティブとなります。
IOWR_	O	内部バスI/Oライト出力端子 アクティブ“L”。I/Oデバイスへデータを書き込む際に出力します。最小2クロック期間アクティブとなります。
WAIT_	I	ウェイト信号入力端子 アクティブ“L”。メモリやI/Oデバイスがデータ送出の用意ができていないことをCPUに通知する為の信号です。この信号がアクティブの間、CPUはウェイト状態を続けます。この入力は外部バス・インターフェース・ユニットのウェイト・ステート・コントローラを介して外部入力端子ERDYに接続されています。
INT_	I	マスカブル割り込み入力端子 アクティブ“L”。I/Oデバイスからの割り込み要求信号を入力します。CPU内の割り込み許可フラグがセットされていて、BREQ_入力がインアクティブならば、現在実行中の命令処理終了後、割り込み処理プログラムを実行します。この入力端子は割り込みコントローラのINT_出力に接続されています。
IACK_	O	インタラプトアクノリッジ出力端子 アクティブ“L”。マスカブルインタラプトを認識した時、割り込みベクタや命令をI/Oデバイスに要求する為の信号です。最小2クロック期間アクティブになります。この出力端子は割り込みコントローラに接続されています。この信号に同期して割り込みコントローラから割り込みベクタを読み込みます。

端子名	I/O	機能説明
EOI <sub>-</sub>	O	エンド・オブ・インタラプト信号出力端子 アクティブ“L”。RETI命令(コードED 4D)を再フェッチする際出力する信号です。割り込みコントローラに接続されています。
LA [15:0]	O	論理アドレス出力。KC82内部のMMUにより物理アドレスA[19:0]に拡張されて出力されます。

### 5.4 レジスタ構成

られる 16 ビットのベースアドレスを保持するレジスタで、IX レジスタ、IY レジスタの 2 つがあります。

#### 5.4.1 専用レジスタ

##### プログラム・カウンタ (PC)

次に実行すべきプログラムのアドレス情報 16 ビットを保持するレジスタです。この PC で示されるアドレスから命令をフェッチします。

##### インタラプトページ・アドレス・レジスタ (I)

割り込みモード 2 において、間接サブルーチン・ジャンプする為の間接アドレスの上位 8 ビットの値を保持するレジスタです。

##### スタック・ポインタ (SP)

RAM 上のスタック領域のその時点での先頭アドレス情報 16 ビットを保持するレジスタです。

##### メモリ・リフレッシュ・レジスタ (R)

8 ビットレジスタで、その内容は命令フェッチごとに自動的にインクリメントされます。ただし最上位ビットはインクリメントされません。

##### インデックス・レジスタ (IX, IY)

インデックス・アドレッシングモード用として用い

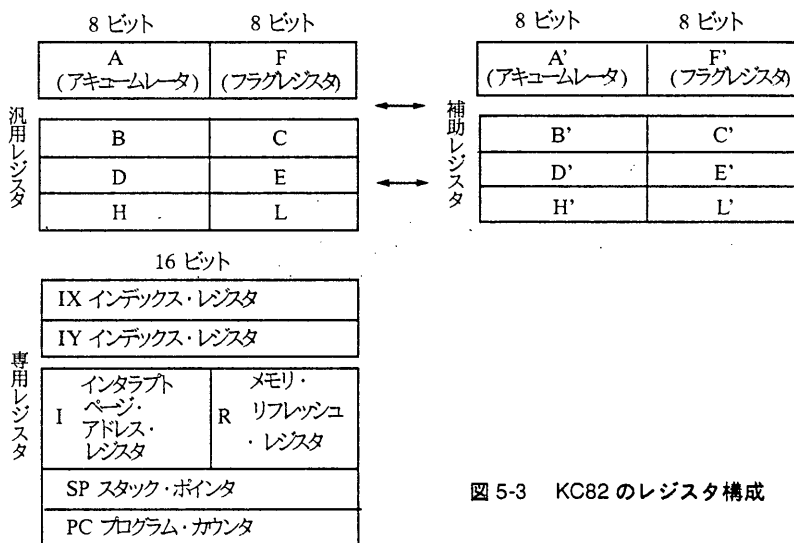


図 5-3 KC82 のレジスタ構成

**アキュムレータ (A, A') とフラグ・レジスタ (F, F')**  
 2組の8ビットアキュムレータ (A, A') と、各々対になった2組のフラグ・レジスタがあります。アキュムレータは8ビットの算術・論理演算の結果を保持するレジスタです。フラグ・レジスタは8ビットまたは16ビットの演算結果の状態を保持するレジスタです。A, F と A', F' の交換は EX AF, AF' で行ないます。

### 5.4.2 汎用レジスタ

2組の汎用レジスタ群があり、それぞれ単独で8ビットレジスタ (B, C, D, E, H, L または B', C', D', E', H', L') として使用可能で、またこれらは16ビットのレジスタペア (BC, DE, HL または BC', DE', HL') としても使用できます。B, C, D, E, H, L と B', C', D', E', H', L' の交換は EXX 命令で行ないます。

### 5.5 フラグについて

#### フラグ・レジスタのビット構成

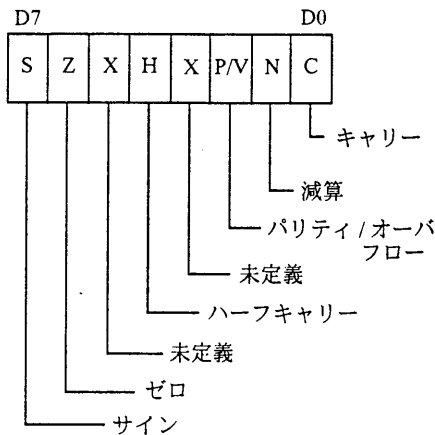


図 5-4 フラグレジスタ

フラグ・レジスタの各ビットの機能を以下に示します。

#### キャリー・フラグ (C)

アキュムレータの最上位ビットからの桁上がりでセットされます。(加算命令、減算命令、シフト・ローテイト命令等)

#### ゼロ・フラグ (Z)

加算/減算/論理演算/INC, DEC/DAA において演算結果がゼロである時にセットされます。またブロック・サーチ命令時の検索データ検出時、ブロック入出力命令時、ローテイト/シフトおよびストリングサーチ、ビットテスト命令等にも用いられます。

#### サイン・フラグ (S)

符号付数値による演算で、演算結果が負ならばセットされます。

#### パリティ・オーバーフロー・フラグ (P/V)

このフラグは2つの機能を持ち、論理演算時にはパリティを示し、算術演算ではオーバーフローが発生したことを示します。

オーバーフローでは、アキュムレータ内の2の補数値が+127を超えるか、-128より小さくなった時にセットされます。またブロック・サーチ命令、ブロック入出力命令、“LD A, I 命令”、“LD A, R 命令”実行時にも用いられます。

#### ハーフ・キャリー・フラグ (H)

BCD 演算の結果、下位4ビットからのキャリー、ポローを示します。

#### サブトラクト・フラグ (N)

減算命令の場合、セットされます。DAA 命令実行時、CPUはこのフラグを見て、先に実行された命令が加算か減算かを自動判定して処理を行います。

### 5.6 機能説明とタイミング

このセクションでは機能説明とタイミングについて説明します。タイミング図に出てくるアドレスバスは説明を簡単にするため MMU の影響を受けない論理アドレス LA[15:0] を使って説明しています。ご注意ください。論理アドレス LA[15:0] は MMU により、物理アドレス A[19:0] に変換されます。またバスサイクルはすべて内部バスサイクルを使って説明しています。外部バスサイクルについては 4 章を参照下さい。

#### 5.6.1 基本動作 (インストラクション・サイクル)

KC82 が命令を実行する過程は、以下の五つの各過程 (マシン・サイクル) の組み合わせにより行います。

- 1) 命令のオペコードをメモリから読み出す (フェッチする) 過程 --- オペコード・フェッチ・サイクル
- 2) メモリからデータを読み出す過程 --- メモリ・リード・サイクル
- 3) メモリにデータを書き込む過程 --- メモリ・ライト・サイクル
- 4) I/O デバイスからデータを読み出す過程 ---

- I/O リード・サイクル
- 5) I/O デバイスにデータを書き込む過程 ---
- I/O ライト・サイクル

以下に各マシン・サイクルごとのタイミングを説明します。

#### オペコード・フェッチ・サイクル

図 5-5 に示すとおり、WAIT\_ の入力がない場合は、1 バイト 1 クロックでフェッチします。また MRD\_ の他に M1\_ がアクティブになります。メモリ・リード・サイクルとの違いは、M1\_ が出力されるかされないかの違いだけです。また、オペコードフェッチサイクルは命令シーケンスの最後のバスサイクルで実行されず。尚、最小 1 クロックサイクルです。

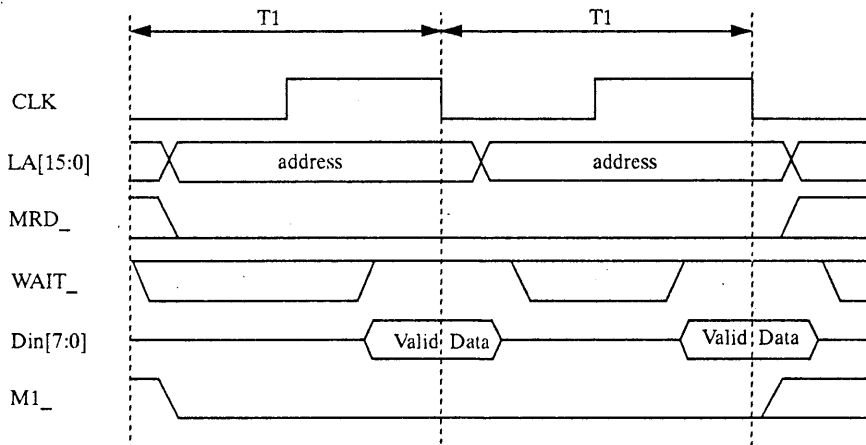


図 5-5 オペコードフェッチサイクル (内部バス・サイクル)  
(上図は、0WAIT の連続リードを表す)

### 内部バス・メモリ・リード・サイクル

オペコード・フェッチ・サイクルとの違いは、  
M1\_がアクティブにならない点です。最小1ク  
ロックサイクルです。

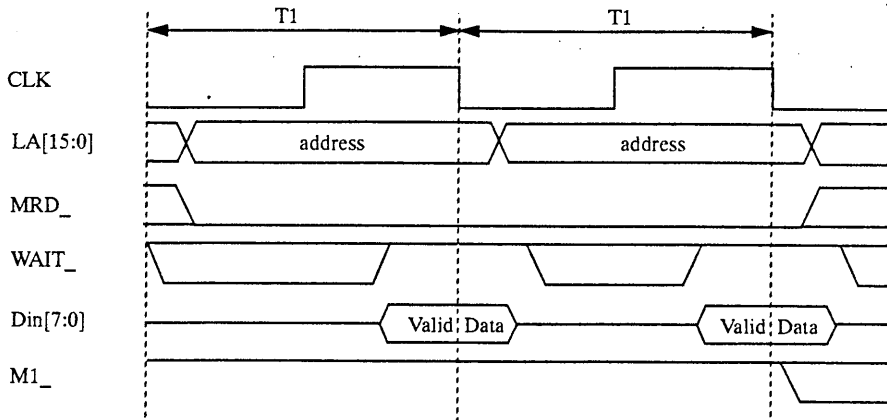


図 5-6 メモリリードサイクル (0 wait)  
(上図は、連続の 0 wait リード動作のタイミング)

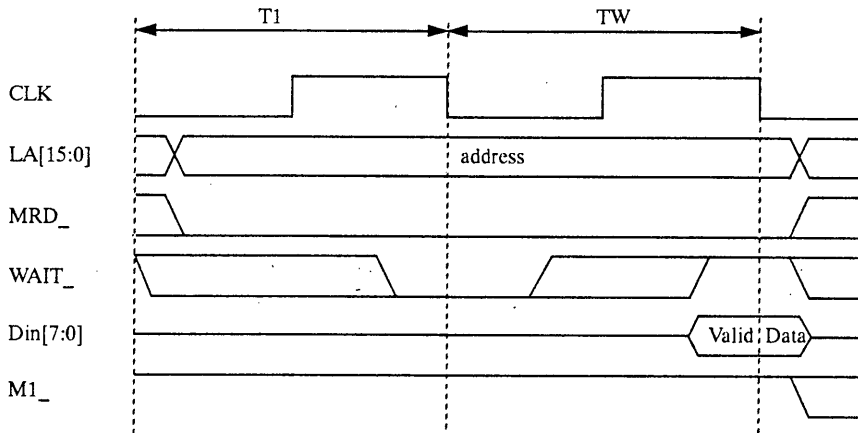


図 5-7 メモリリードサイクル (1 wait)  
(上図は、1 wait リード動作のタイミング)

内部バス・メモリ・ライト・サイクル  
 最小1クロックサイクルです。

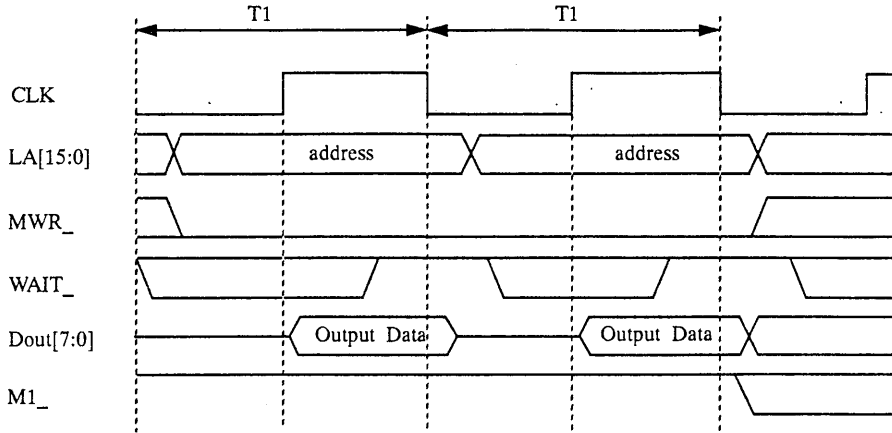


図 5-8 メモリライトサイクル (0 wait)  
 (上図は、連続 0 wait ライト動作のタイミング)

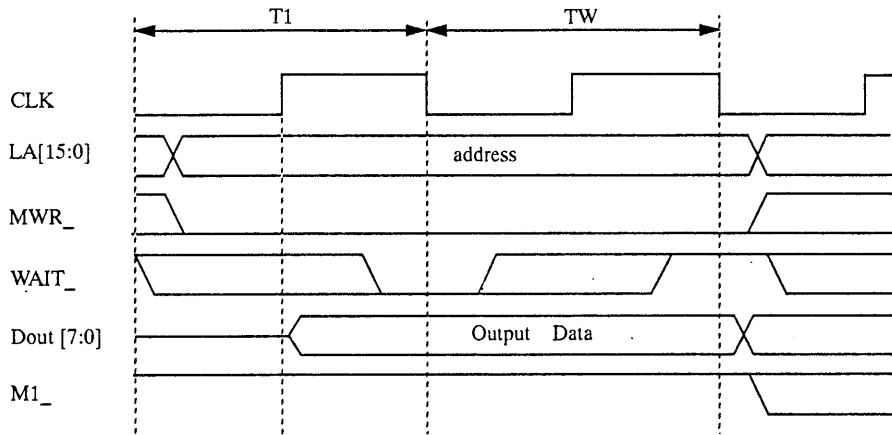


図 5-9 メモリライトサイクル (1 wait)  
 (上図は、1 wait ライト動作のタイミング)

内部バス・I/O リード・サイクル  
 最小2クロックサイクルです。

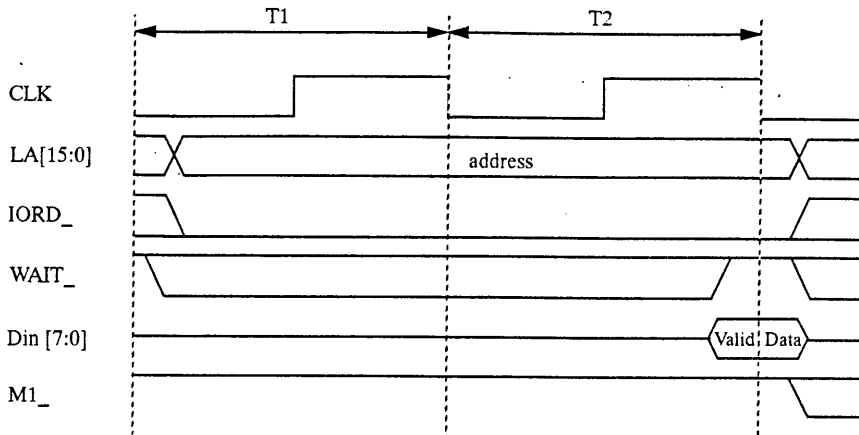


図 5-10 I/O リードサイクル (0 wait)  
 (上図は、0 wait 動作のタイミング)

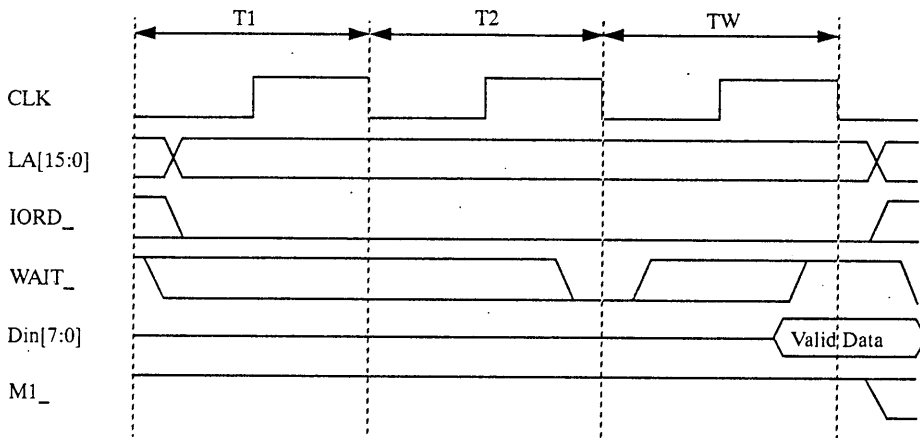


図 5-11 I/O リードサイクル (1 wait)  
 (上図は、1 wait 動作のタイミング)

内部バス・I/O ライト・サイクル  
 最小2クロックサイクルです。

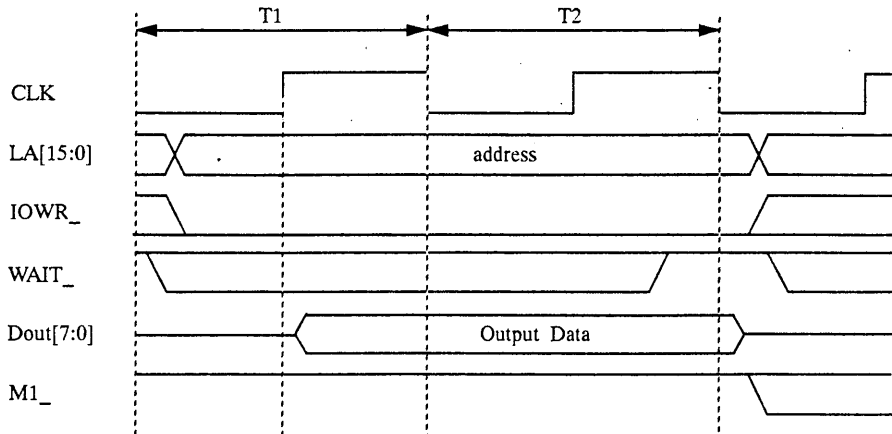


図 5-12 I/O ライトサイクル (0 wait)  
 (上図は、0 wait 動作タイミング)

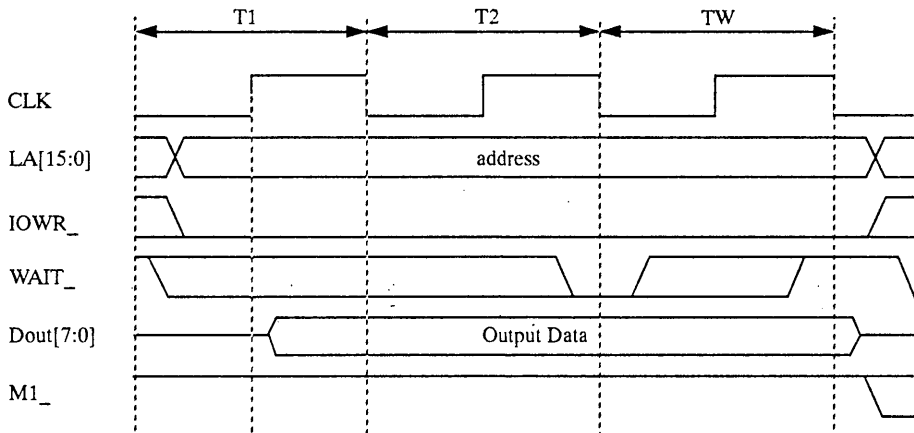


図 5-13 I/O ライトサイクル (1 wait)  
 (上図は、1 wait 動作タイミング)

### 命令プリフェッチサイクル

KC82 の命令サイクルの終りには、必ずプリフェッチサイクルが存在します。図 5-15 に LD A,[1234H] ⇒ ADD A,D ⇒ JP 1000H ⇒ LD [HL], A 命令を実行するタイムチャートを示します。また、図 5-14 には図 5-15 に対応したアセンブラコーディングを示します。図中の (以下“矢印”と呼びます) がプリフェッチサイクルを表します。①の矢印は n+3 番地の '82H' (ADD A,D 命令) を、②の矢印は n+4 番地の 'C3H' (JP xxxx 命令) を、③の矢印は 1000H 番地の '77H' (LD [HL], A 命令) をそれぞれプリフェッチするサイクルで、図でもわかるように次に実行しようとする命令の直前 (現在実行中の命令の終り) にプリフェッチされることがわかります。

address	mnemonic	code
n	LD A, [1234H]	3A 34 12
n+3	ADD A, D	82
n+4	JP 1000H	C3 00 10
1000H	LD [HL], A	77
1234H		5A

図 5-14 図 5-15 のアセンブラコーディング

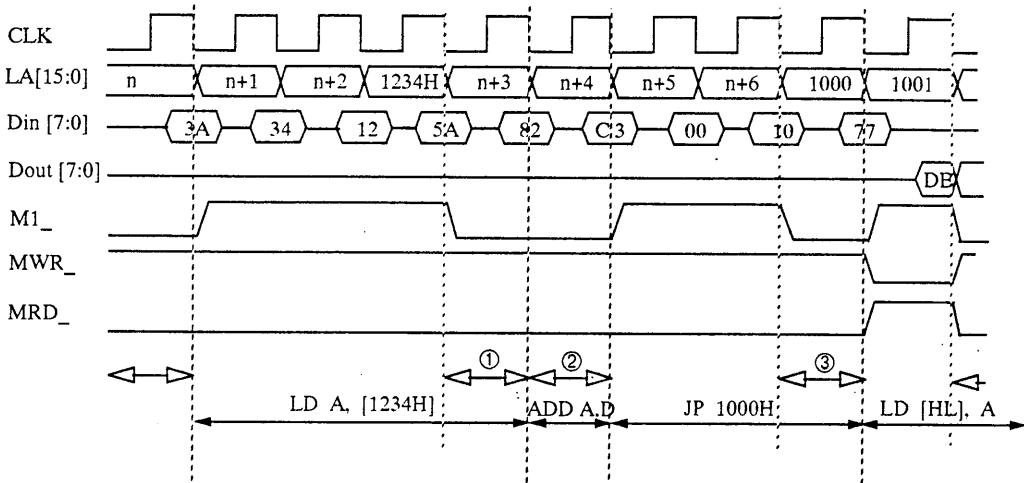


図 5-15 プリフェッチの様子 (◀→ がプリフェッチサイクルを表す。また 1000H 番地、1234H 番地にはそれぞれ '77H', '5AH' が格納されており、D レジスタは '81H' とする)

### 特殊な場合のプリフェッチ(プリフェッチの無効化)

プリフェッチサイクルは例えば図 5-1 の JP 1000H 命令のように④でプリフェッチしたデータ '77H' は内部で保持してデコードされ実行されますが、割り込み系に関してはプリフェッチされたデータは保持せず、廃棄され割り込みルーチンから復帰後に再び廃棄されたデータのフェッチを開始します。つまり、割り込み認識処理で PC をスタックに PUSH する時に PC をマイナス 1 してセーブし、RETI や RETN 命令での POP 動作で再セットされる PC、すなわち廃棄されたデータの格納されているアドレスからプリフェッチをします。図 5-19 に LDIR 命令実行中にモード 1 割り込みが発生したタイミング図を示します。図中の⑥でプリフェッチされたデータ 'ED' は廃棄され割り込み認識処理へ進みます。そして、割り込み復帰後再びこの 'ED' のプリフェッチから開始されます。なお、バスリクエストサイクルでは通常命令と同じく直前にプリフェッチされたデータは保持されバス解放時に next アドレス(ジャンプ命令時はジャンプ先のアドレス)から処理を開始します。図 5-18 に LDIR 命令実行中にバスリクエスト要求が発生した場合のタイミング図を示します。図中の⑥でプリフェッチされたデータ 'ED' は廃棄されずに保持されバス解放時に 'ED' の次のアドレスからプリフェッチがされていることがわかります。

### 5.6.2 バス解放(バス・リクエスト/アクノレッジ・サイクル)

通常 CPU がアドレス・バス、コントロール・バス(MRD\_, MWR\_, IORD\_, IOWR\_)の制御権を握っていますが、外部からバス要求(BREQ\_="L")があると、CPU はアドレスバスをハイ・インピーダンス状態にして、全ての割り込みを禁止し、BACK\_信号を出力して("L")、外部にバスの制御権を譲ります。この機能により、CPU を介さないデータ転送が可能となります。図 5-17 にバスリクエスト基本タイムチャートを示します。このサイクルは、BREQ\_が"L"の期間ずっと続きます。バスリクエストの受け付けタイミングについては、ブロック転送命令、ブロックサーチ命令、入出力命令のうちくりかえし動作のある命令を除いて命令の切れ目です。特殊な例として

は、図 5-18 に示すように LDIR 命令実行中の④でリクエストが発生していますが、LDIR 命令の最後のバスサイクルである⑤で受け付けられます。注意するのは、各バスサイクルの切れ目ではなく命令サイクルの切れ目であることです。

### 5.6.3 割り込み機能とタイミング

KC82 で受け付けられる割り込み要求には、以下の 2通りがあります。

- 1)INT\_によるマスク可能な割り込み(マスカブル割り込み)
  - 2)NMI\_によるマスク不可能な割り込み(ノン・マスカブル割り込み)
- 2)は、1)よりも優先順位が高く、1)、2)同時に発生した場合は、2)の方が先に受け付けられます。

### マスカブル割り込み

マスカブル割り込みは EI 命令によって割り込みを許可され、DI 命令によって割り込みを禁止されます。この制御は 2つのイネーブル・フリップフロップ(IFF1, IFF2)によって制御されています。下図はそのフリップフロップの状態遷移とその要因を示したものです。

要因	IFF1	IFF2	
リセット	0	0	
DI 命令	0	0	
EI 命令	1	1	
INT 受け付け	0	0	
NMI 受け付け	0	-	
RETN 命令;受け付け	IFF2	-	IFF1 に IFF2 がコピー
LD A,I 命令	-	-	P/V に IFF2 がコピー
LD A,R 命令	-	-	P/V に IFF2 がコピー

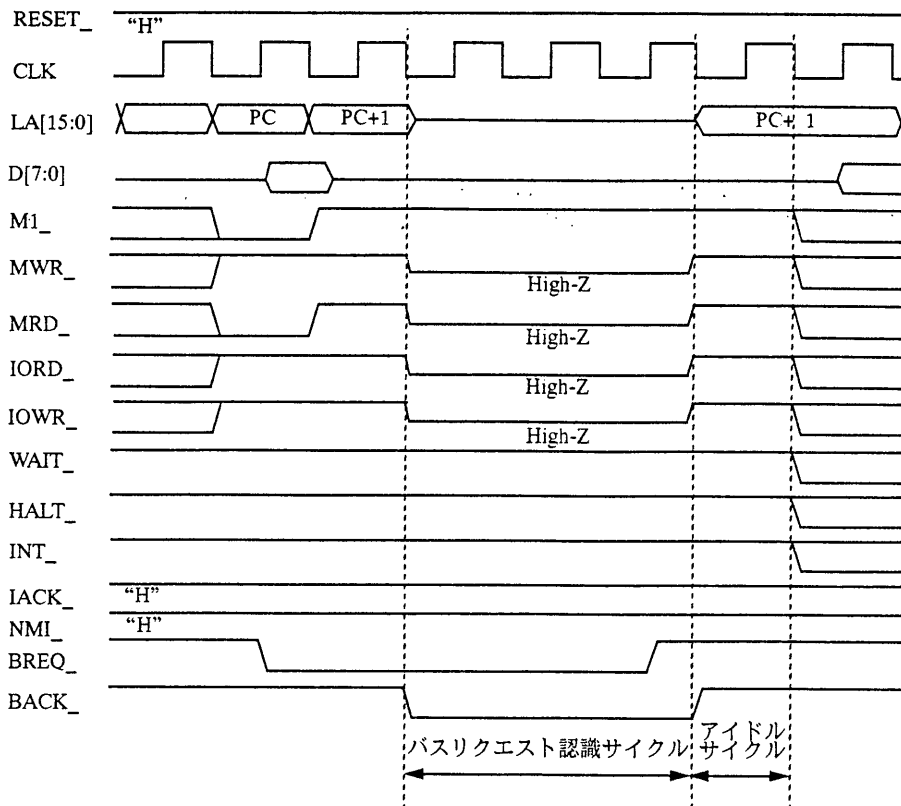


図 5-17 バスリクエストタイミング

マスクابل割り込みが有効なのは、以下の条件を全て満たしている時です。

- 1) 2 つのフリップフロップが共にセットされている。(EI 命令や RETN 命令後は、さらに次の 1 命令実行後に初めて有効になります。従って、EI 命令と DI 命令が連続した場合には、割り込みは受け付けられません。)
- 2) BREQ\_ がインアクティブ ("H") になっている。(BREQ\_ が "L" から "H" に変わった直後では、1 命令実行後に初めて有効になります。)
- 3) NMI\_ の立ち下がりエッジを保存していない。  
従来の Z80 では M1\_ と IORQ\_ とで外部に割り込み許可を認識させていましたが、KC82 では独立した IACK\_ 端子によってこれを実現しています。

### マスクابل割り込みのモード

マスクابل割り込みには 3 種類のモードがあり、それぞれ処理が異なります。以下に各モードでの動作を説明しますが通常はモード 2 を使って下さい。

#### (1) モード 0

リセット時には、自動的にこのモードにセットされます。また他のモード実行中は IM 0 命令によってこのモードに移ります。このモードは、割り込みサイクル中に読み込んだ命令をそのまま実行します。通常この割り込みで使う命令は、RST 00H- RST 38H、または CALL 命令です。RST 命令、CALL 命令を読み込んだ場合のタイミング図をそれぞれ図 5-20、図 5-21 に示します。なおモード 0, 1, 2 共通して、IM X (X = 0,1,2) 命令実行中に INT\_ が入力された場合、IM X 命令実行直後からモードの変更が有効になります。

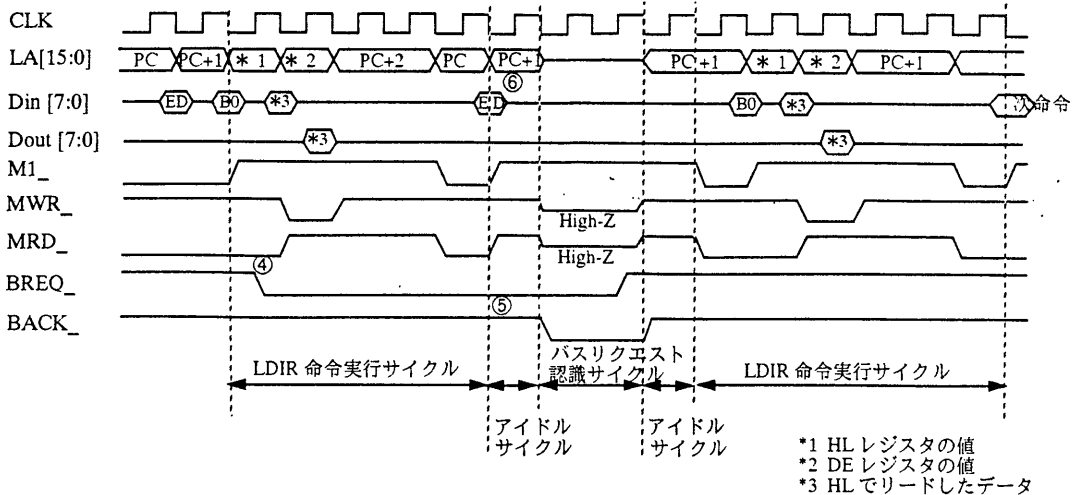


図 5-18 バスリクエスト受け付けタイミング

(上図では、LDIR 命令実行中に BREQ\_ 入力が④で発生し、命令の切れ目である⑤で受け付ける。尚、⑥でフェッチしたデータは保持される。)

### (2) モード 1

IM 1 命令によってこのモードに移ります。このモードでは、PC(プログラム・カウンタ)の内容をスタックに退避後、割り込みサイクル中に読み込んだデータを無視して、強制的に RST 38H 命令を実行するものです。モード 1 の割り込み認識シーケンスのタイミング図を図 5-22 に示します。

### (3) モード 2

IM 2 命令によってこのモードに移ります。このモードでは、図 5-23-A のように割り込みサイクル中に読み込んだ割り込みベクタを下位 8 ビット (ただし、最下位ビットは '0') とし、I レジスタの内容を上位 8 ビットとする 16 ビットアドレスで割り込みプログラムのスタートアドレステーブルをアクセスします。そして、読み込んだスタートアドレスから始まる割り込みプログラムを実行します。モード 2 の割り込み認識シーケンスのタイミング図を図 5-23-B に示します。

### 割り込みの受け付けタイミング

図 5-19 に割り込み受け付けのタイミングを示します。図では、LDIR 命令実行中に INT が発生した例で

すが、④で INT 入力が発生し、LDIR 命令の最後のプリフェッチサイクルである⑤で初めて受け付けられます。この時、⑥でフェッチしたデータ 'ED' は CPU 内部にはとどまらず割り込み復帰後再フェッチされます。

### ノン・マスカブル割り込み

ノン・マスカブル割り込みはソフトウェアでマスクできない割り込みです。NMI\_ の立ち下がリエッジの検出は各命令の最後のステートのクロックの立ち下がりで行ないます。この時点より前に NMI\_ の立ち下がリエッジがあれば、割り込みを受け付けます。ノン・マスカブル割り込みは BREQ\_ がインアクティブ ("H") になっている時に有効になります。(BREQ\_ が "L" から "H" に変わった直後では、1 命令実行後に初めて有効になります。)

ノン・マスカブル割り込みを認識すると、PC の内容をスタックにセーブして、0066H 番地へサブルーチンコールします。この割り込みルーチンからの復帰は、RETN 命令の実行により実現します。NMI\_ 認識シーケンスのタイミング図を図 5-24 に示します。

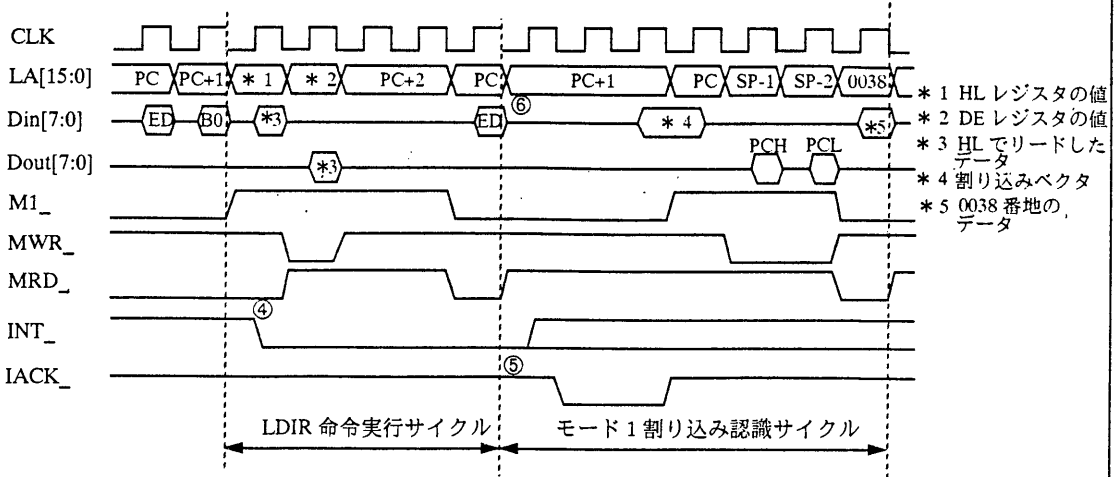


図 5-19 割り込み受け付けタイミング (モード 1)

(上図では、LDIR 命令実行中に INT\_ 入力が④で発生し、命令の切れ目である⑤で割り込みを受け付ける。割り込みリターン後は⑥の ED データのフェッチから再開するケースを想定)

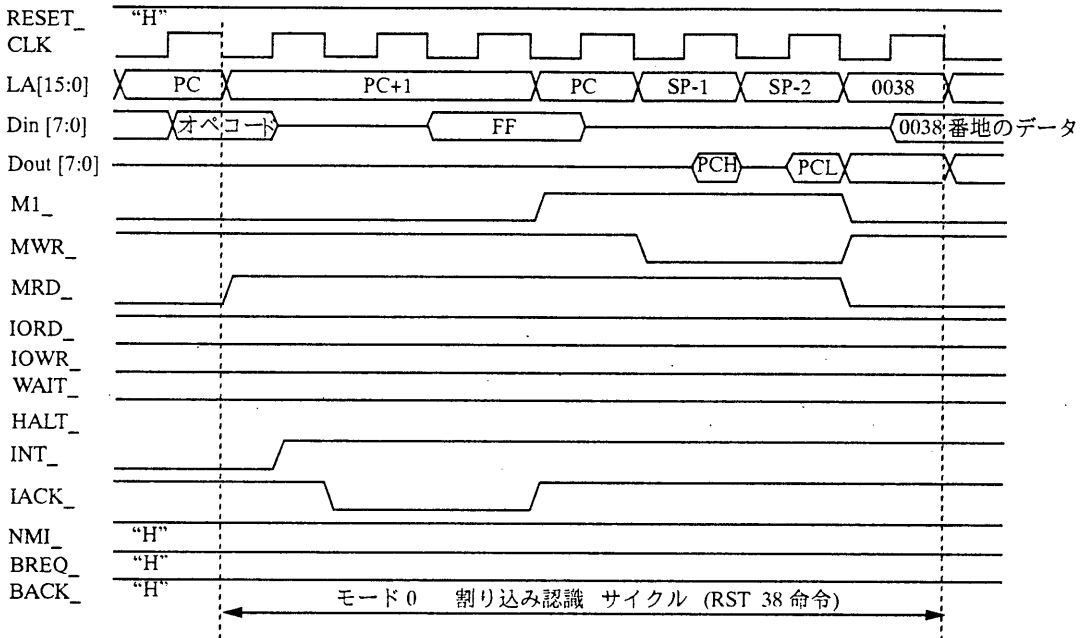


図 5-20 モード 0 タイミング

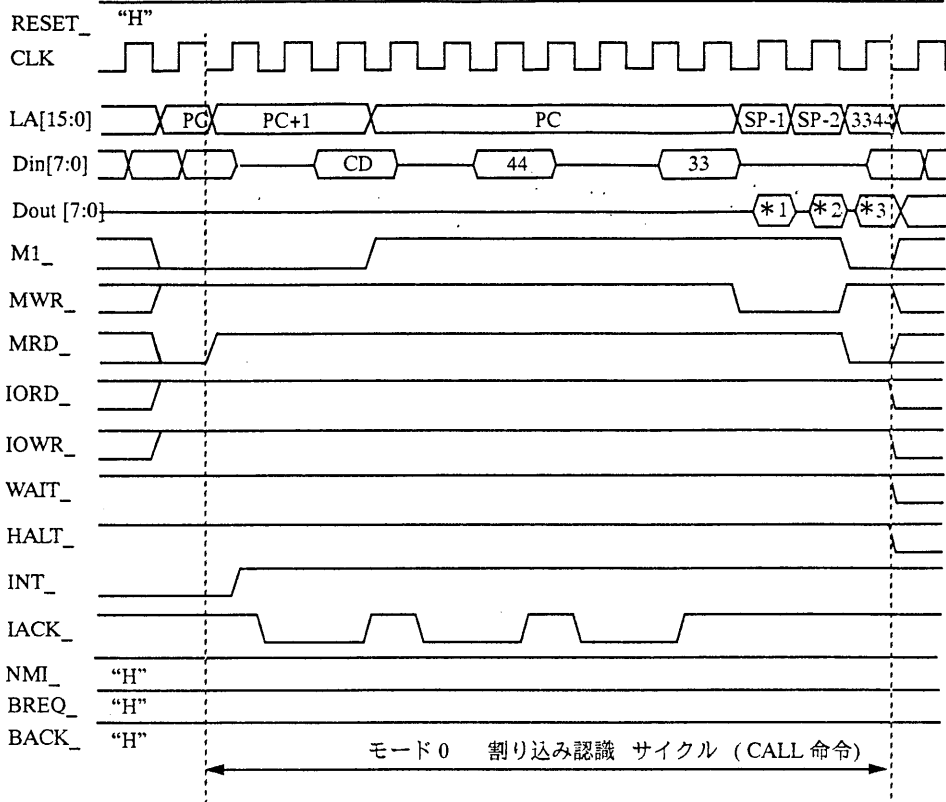


図 5-21 モード0 タイミング

- \* 1 PCH
- \* 2 PCL
- \* 3 3344 番地のデータ

### 割り込みのイネーブルおよびディスエーブルについて

割り込みのイネーブルおよびディスエーブルは内部の IFF1 および IFF2 というフリップフロップで制御されます。図 5-16 に 2 つのフリップフロップの動作の状態を表に示します。マスカブル割り込みが受け付けられるのは、IFF1、IFF2 がともに '1' の状態です。つまり、リセット直後および DI 命令実行直後は '0' でありマスカブル割り込みは受け付けられず、また EI 命令で両フリップフロップが '1' となり割り込みを受け付けられる状態となります。フリップフロップが 2 つあるのはノンマスカブル割り込みが発生したときのイネーブル/ディスエーブル状態を IFF2 に記憶す

るためです。例えば、NMI が発生する直前の IFF の値が IFF1='1'、IFF2='1' (イネーブル状態) とすると NMI の発生により IFF1='0' となり INT の受け付けはできなくなります。この時 IFF2 には '1' が保存されています。そして、RETN 命令により IFF2 の値つまり '1' が IFF1 にコピーされ両フリップフロップとも '1' となりイネーブル状態に復帰出来るわけです。

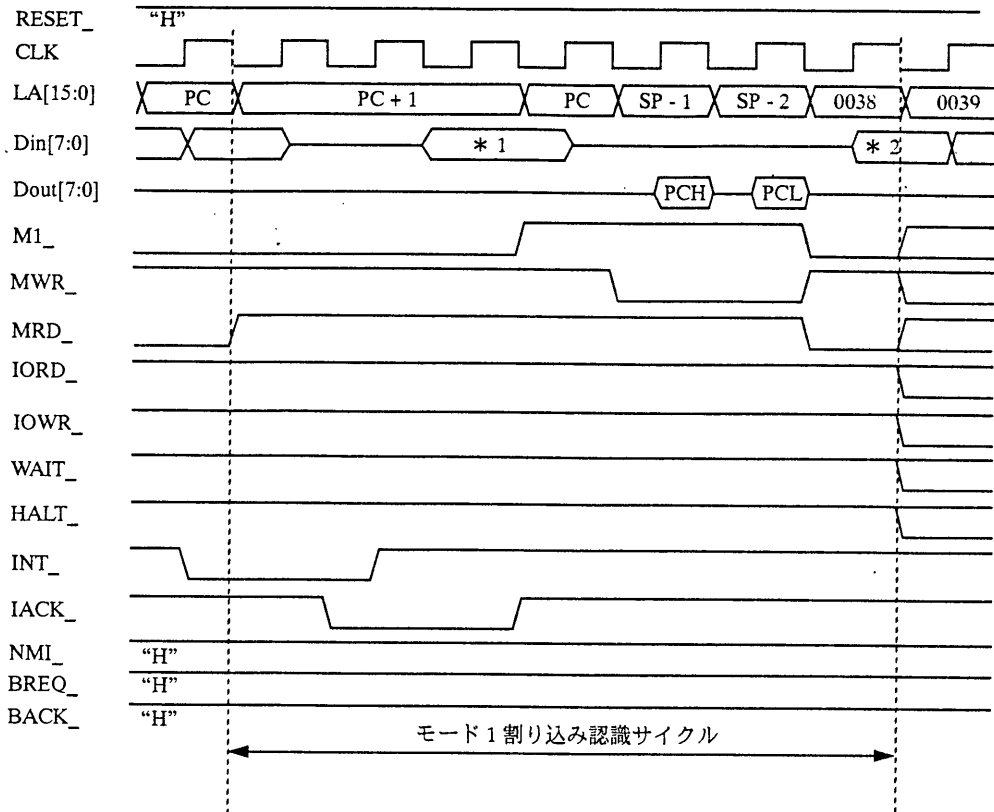


図 5-22 モード1 タイミング

\* 1 割り込みベクトル  
\* 2 0038 番地のデータ

### 5.6.4 HALT の実行と解除

HALT 命令により CPU 内部では、NOP 命令を実行し続けますが、リセット入力や割り込み入力(ノン・マスカブル割り込みまたは、割り込み許可フラグがセットされている状態でのマスカブル割り込み)があれば、HALT 状態は解除されます。図 5-25 は HALT 命令実行中にモード 2 の割り込みが発生し HALT が解除されるタイミング図です。HALT は 2 つのバスサイクルで構成され第 1 サイクルはアイドルサイクルで、第 2 サイクルがオペコードフェッチサイクルです。HALT 実行中のアドレスは HALT 命令が置かれているアドレスの次のアドレスを出力します。この時第 2 サイクルでフェッチしたデータは KC82 内部には取り込まれません。そして、INT\_ 入力により HALT 命令の

第 2 サイクルのクロックの立ち下がりで割り込み (図ではモード 2) を受け付けるのと同時に HALT\_="H" となります。

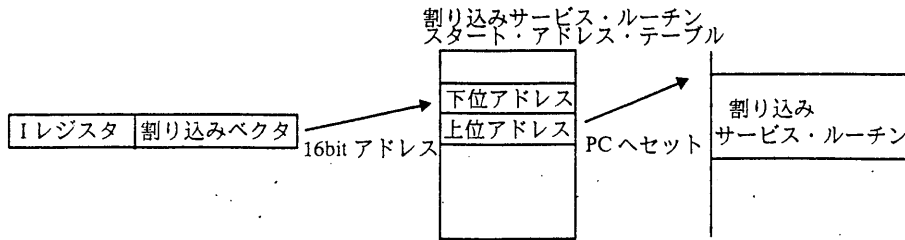


図 5-23-A モード2 割り込み

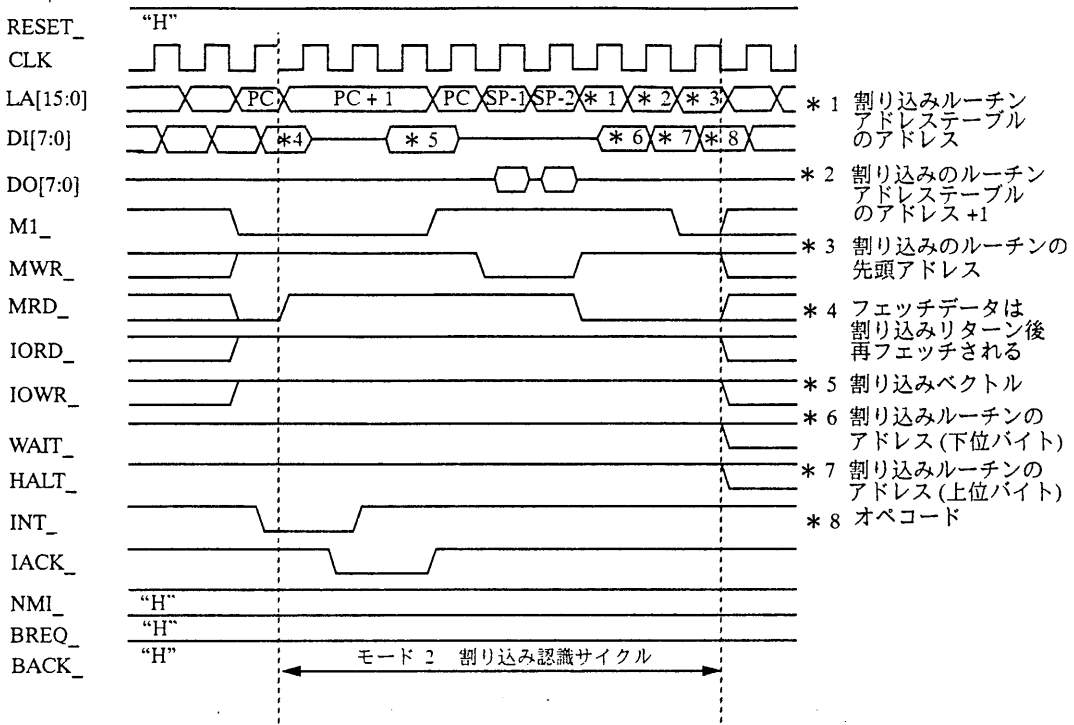


図 5-23-B モード2 タイミング

### 5.6.5 リセット・タイミング

RESET\_入力を最小3クロック期間アクティブ（“L”）にすることによって、CPUはリセットされます。リセット期間中はアドレスバスはハイ・インピーダンス状態となります。RESET\_入力がインアクティブ（“H”）になると、3クロック目の立ち下がりからオペコード・フェッチ・サイクルが開始され、0000H番地の命

令から順次実行を開始します。また、割り込みモードは0にセットされ、割り込みフラグIFF1, IFF2はリセットされます。Iレジスタ、Rレジスタは‘00H’にリセットされます。

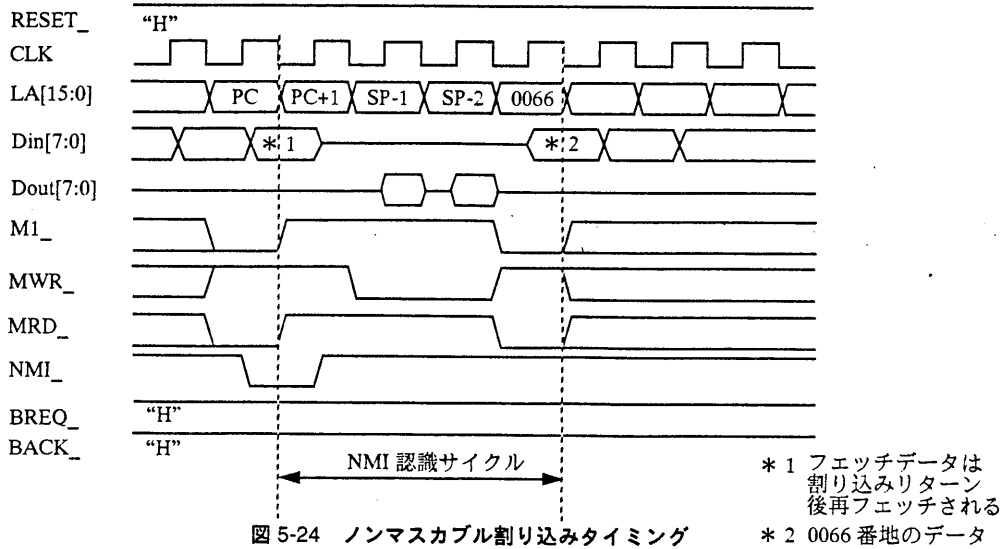


図 5-24 ノンマスクابل割り込みタイミング

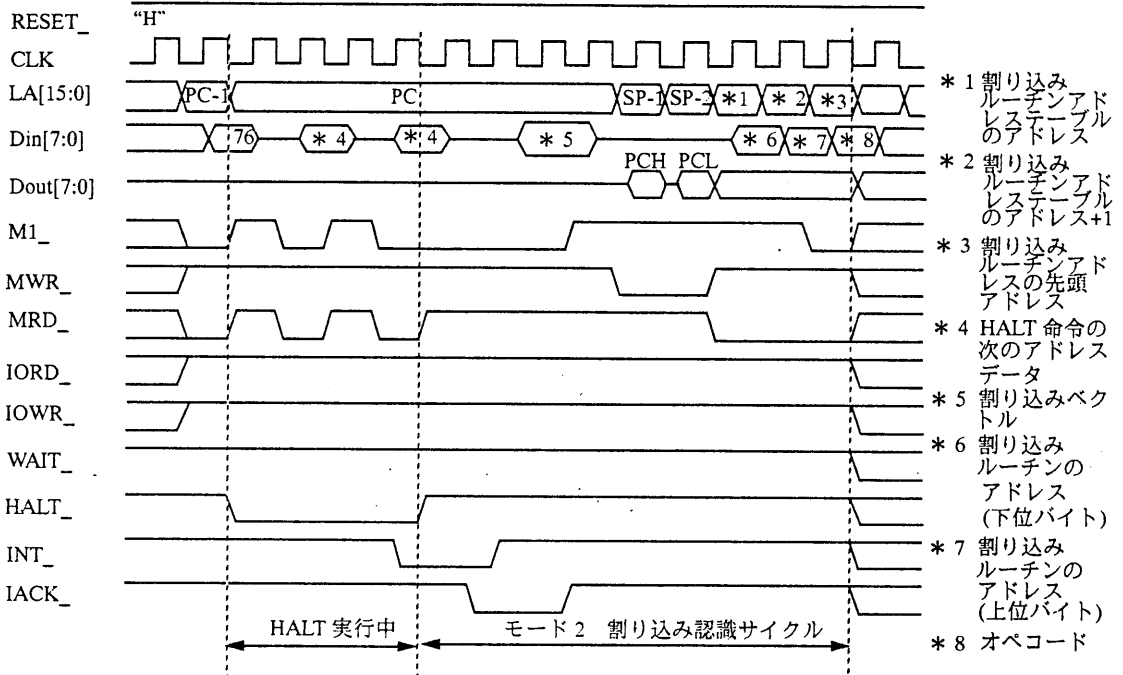


図 5-25 HALT 解除 (モード 2 による)

### 5.7 MMU

#### 5.7.1 概要

MMU ブロックは KC82 の 16 ビットの論理アドレス LA[15:0] を 20 ビットの物理アドレス A[19:0] に変換する回路です。ただし、MMU がアドレスを変換するのはメモリアクセスの場合だけで、I/O アクセスの場合はアドレス空間に何の変換も加えません。MMU は下

のレジスタと物理アドレス生成回路から構成されています。物理アドレス生成回路の動作は後のセクションで説明しています。

表 5-1 レジスタ構成

名称	ビット数	Read / Write
境界/ ベースレジスタ 1 (BBR1)	8 ビット	R/W
"  2 (BBR2)	"	R/W
"  3 (BBR3)	"	R/W
"  4 (BBR4)	8 ビット*	R/W
ベースレジスタ 1 (BR1)	8 ビット	R/W
"  2 (BR2)	"	R/W
"  3 (BR3)	"	R/W
"  4 (BR4)	"	Read Only

\*上位 2 ビットは Read Only

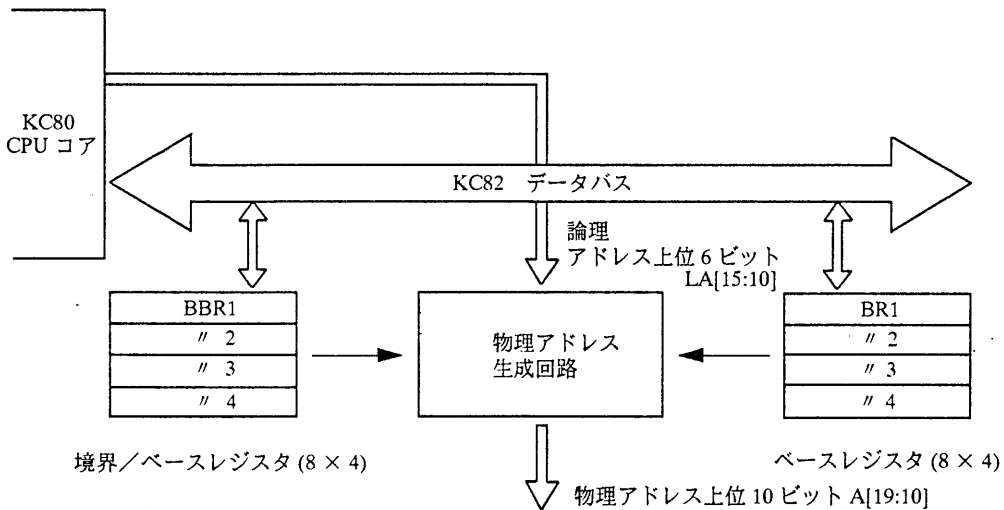


図 5-26 ブロック図

### 5.7.2 各レジスタの説明

MMUブロックには8ビットのレジスタが8本含まれています。この8本のレジスタでMMUブロックの動作に必要な4つの10ビットデータと4つの6ビットデータを保持します。

#### 境界/ベースレジスタ (BBR1 ~ BBR4)

本レジスタの上位2ビット(A1<1:0>~A4<1:0>)はベースレジスタの8ビットとともに物理アドレスのベースアドレス指定に使われます。下位6ビットは論理アドレス境界値(B1<5:0>~B4<5:0>)で、論理空間の領域分割の境界値として使われます。またBBR4の上位2ビットは固定値であり、データを書き込んでもその内容は変化しません。

#### ベースレジスタ (BR1~BR4)

本レジスタは境界/ベースレジスタの上位2ビットとともに10ビットの物理アドレスベース(A1<9:0>~A4<9:0>)を構成し、物理アドレスのベースアドレス指定に使われます。この4本のベースレジスタ(BR1~BR4)のうち、BR4はF0H固定で書き込みできません。

#### 論理アドレス境界値と物理アドレスベース

MMUブロックは論理アドレス空間を5つの領域に分割しており、それぞれを物理アドレス空間にマッピングしています。このためMMUブロックは5つの論理アドレス境界値(B0 ~ B4)と5つの物理アドレスベース(A0 ~ A4)を必要とします。このうちA0とB0は固定でそれぞれA0=000H B0=00Hです。残りのデータは図5-27にあるように境界/ベースレジスタとベースレジスタに割りつけられています。

I/Oアドレス レジスタ名

		bit7	6	5	0
00H	BBR1	A1<1:0>		B1<5:0>	
01H	BR1	A1<9:2>			
02H	BBR2	A2<1:0>		B2<5:0>	
03H	BR2	A2<9:2>			
04H	BBR3	A3<1:0>		B3<5:0>	
05H	BR3	A3<9:2>			
06H	BBR4	A4<1:0>		00B固定	B4<5:0>
07H	BR4	A4<9:2> F0H固定			

図 5-27 MMU ブロックのレジスタ

### 5.7.3 MMUによる物理アドレス生成

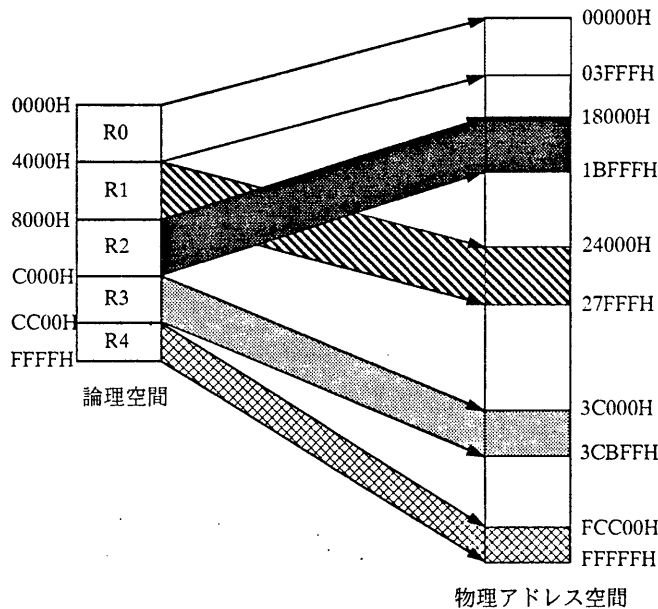
論理アドレス空間は、図5-28に示すように5つの領域(R0~R4)に分割されています。この5つの論理領域は論理アドレス境界値(B0<5:0>~ B4<5:0>)で決められています。Rn領域の下限アドレスは(Bn+1) × 400Hで決定され、Rn領域の上限アドレスは(B(n+1)+1) × 400H-1となります。(ただし、R0領域の下限アドレスは常に00000H、R4領域の上限アドレスは常にFFFFFFHです。)これらの領域の物理空間上の配置は分割されたそれぞれの領域の物理アドレスベース An(10ビット)を論理アドレスの上位6ビットと足し合わせて生成します(図5-29参照)。論理空間の分割は論理アドレスの上位6ビットと論理アドレス境界値(B0~B4)の大小比較で決定されるので、1Kバイト単位で設定が可能です。なおR1領域の下限アドレスの最小値は0400HでA0は固定値000Hのため、R0領域に属する論

理空間の最初の1Kバイト(0000H~03FFFH)は常に物理空間上の00000H~003FFFHに割り付けられます。

参考のため設定例を図5-28に示します。例えば

- B0 = 00H (固定)
- B1 = 0FH
- B2 = 1FH
- B3 = 2FH
- B4 = 32H
- A0 = 000H (固定)
- A1 = 080H
- A2 = 040H
- A3 = 0C0H
- A4 = 3C0H (固定)

と設定した場合、5つの論理領域と物理領域の対応は以下ようになります。



領域 R0	論理アドレス 0000H ~ 3FFFH	物理アドレス 00000H ~ 03FFFH
領域 R1	論理アドレス 4000H ~ 7FFFH	物理アドレス 24000H ~ 27FFFH
領域 R2	論理アドレス 8000H ~ BFFFH	物理アドレス 18000H ~ 1BFFFH
領域 R3	論理アドレス C000H ~ CBFFFH	物理アドレス 3C000H ~ 3CBFFFH
領域 R4	論理アドレス CC00H ~ FFFFH	物理アドレス FCC00H ~ FFFFH

図 5-28 論理アドレスと物理アドレスの対応例

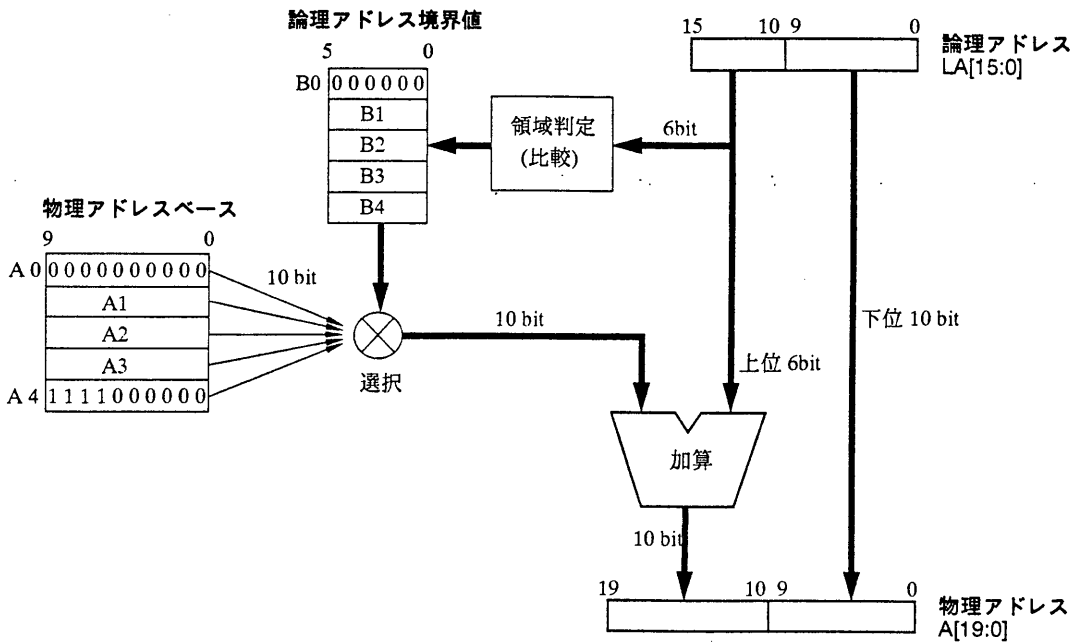


図 5-29 物理アドレス生成

### 5.7.4 MMU の動作

#### (1) メモリ空間

MMU は、CPU がメモリ空間をアクセスするとき  
にそのアドレスを変換します。これは具体的には  
次のような場合です。

1. 命令のフェッチ
2. 命令によるメモリ空間へのリードあるいはラ  
イト
3. 割り込みのリスタートアドレスへのフェッチ
4. モード2 割り込みにおけるスタートアドレス  
テーブル

#### (2) I/O 空間

CPU が I/O 空間をアクセスするときには、MMU  
はアドレスを変換せず、論理アドレスをそのまま  
出力します。このとき、アドレスの上位 4 ビット  
には 0H が出力されます。

### 5.7.5 リセット時

リセット時には下記のように初期化されます。

B0	=	00H (固定)
B1	=	3FH
B2	=	3FH
B3	=	3FH
B4	=	3FH
A0	=	000H (固定)
A1	=	000H
A2	=	000H
A3	=	000H
A4	=	3C0H (固定)

このとき論理アドレス空間は R0 領域のみになり、  
論理アドレス空間の 64K バイトは物理アドレス空間  
先頭の 64K バイトにマップされます。

### 5.7.6 MMU 使用上の注意

- (1) 論理アドレス境界値は、 $B0 < B1 < B2 < B3 < B4$   
となるように設定したとき、すべての領域が有  
効になります。論理アドレス境界値の大小関係が  
逆転あるいは等しくなった場合には数字の大きい  
方が有効となり、数字の小さい方は無効となり、  
この論理アドレス境界値に対応する領域が消滅し  
ます。例えば、 $B1 \geq B2$  に設定したとき、R1 領域  
は消滅します。

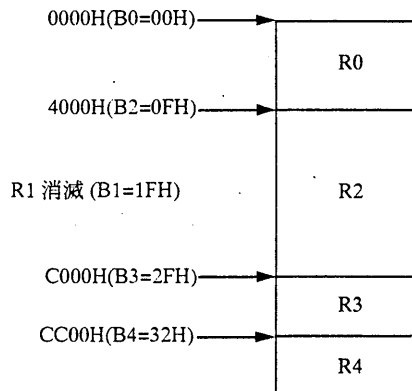


図 5-30 B1  $\geq$  B2 としたときの例

(2) MMU レジスタ (境界/ベースレジスタ、ベースレジスタ) にデータを書き込んだとき、その設定が有効になるのは、MMU レジスタにデータを書き込む I/O ライトサイクルの次のバスサイクルからです。物理アドレスの割付の変更などを行なう場合は、十分注意してください。

00000H ~ 003FFH となります。

(6) B1 ~ B4 を設定するときは 3FH 未満の値を設定してください。3FH は無効設定として扱われ、その領域は消滅します。

(3) 各領域 (R0 ~ R4) の用法についてハードウェア的な制約はありませんが、当社の指定のリンク (LSI-C の knil) の制約から以下のような割り付けをおすすめします。

LSI-C は LSI ジャパン社の登録商標です。

- R0 常駐プログラム領域 (共通プログラム、割り込み処理プログラムなど)
- R1 プログラムバンクウインドウ
- R2 データバンクウインドウ 1
- R3 データバンクウインドウ 2
- R4 常駐データ領域 (スタックなど)

(4) 本 MMU 回路は下記の I/O アドレスを占有します。この I/O アドレスにはユーザ用の I/O は割り付けられないので御注意ください。

表 5-2 MMU 回路が使用する I/O マッピング

I/O アドレス	レジスタ名
00H	BBR1
01H	BR1
02H	BBR2
03H	BR2
04H	BBR3
05H	BR3
06H	BBR4
07H	BR4
08H ~ 0FH	川崎製鉄予約

(5) R1 の論理アドレス境界は R1 論理アドレス境界値 (B1) で決められており、R1 領域の下限アドレスは  $(B1+1) \times 400H$  となります。このため、R1 領域の下限アドレスの下限値は 0400H となります。一方、R0 領域の下限アドレスは 0000H となっており、論理アドレスの最初の 1K バイトは必ず R0 領域に属し、そのマップ先は固定されており

### 6. 割り込みコントローラ

#### 6.1 概要

KL5C80A16は割り込みコントローラとしてKP69マイクロセルを搭載しています。KP69は弊社CPU(KC80あるいはKC82)専用に開発された小型割り込みコントローラです。KP69はKC80あるいはKC82のモード2割り込みに対応して16レベルの割り込み要求入力をサポートできます。

各割り込み要求入力は、HIGH / LOW の2つの優先順位グループに分割設定でき、各グループの中では割り込み要求入力レベルのビット番号が大きいものほど優先順位は高く設定されます。各割り込み要求入力のエッジ/レベル動作、マスク状態、および割り込みベクタの上位3bitはプログラム可能となっています。

KP69は割り込み要求を受け付けるとマスク状態、優先順位を判定し、CPUへINT\_信号を出力します。そしてCPUからIACK\_信号が返ってくると、プログラムされた割り込みベクタをデータバスに出力します。また、CPUのEOI\_信号="L"によって割り込み処理終了を認識します。このCPUのEOI\_信号はRETI命令実行時に"L"になります。したがって割り込み処理ルーチンの最後にRETI命令を置くことにより、KP69は自動的にRETI命令を認識して割り込み処理終了を認識します。さらに不正割り込み検出機能を備えています。

#### 特徴

- ・ クロック同期式の割り込みコントローラです。
- ・ KC82のモード2割り込み対応です。
- ・ 16レベルの割り込み要求入力の、優先順位が制御できます。
- ・ 割り込み入力は個々にマスク可能です。
- ・ 多重割り込み処理が可能です。
- ・ 割り込み要求入力のエッジ/レベル切替が可能です。
- ・ 不正割り込み検出機能を有します。
- ・ CPUのRETI命令実行を検出し、割り込み処理終了を自動的に検出できます。

#### 6.2 ブロック図

KP69の全体ブロック図を以下に示します。KP69のIACK\_入力はCPUのIACK\_出力、EOI\_入力はCPUのEOI\_出力、INT\_出力はCPUのINT\_入力にそれぞれ内部で接続されています。また割り込み要求入力IR[15:0]には表6-1のような信号が接続されています。

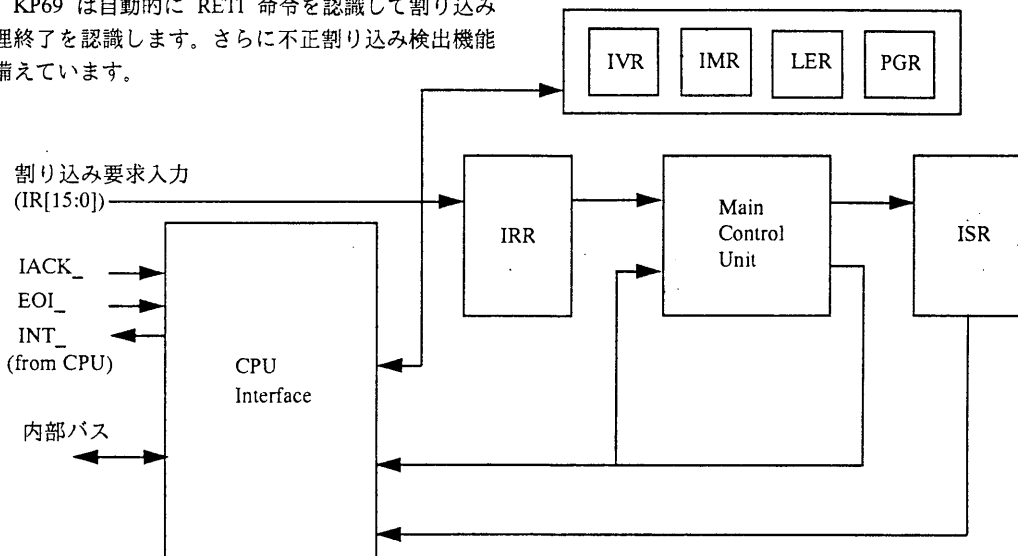


図 6-1 KP69 ブロック図

表 6-1 KL5C80A16 の割り込み

	割り込み要求元
IR[15]	外部入力 (P23)
IR[14]	外部入力 (P22)
IR[13]	タイマ/カウンタチャンネル1 割り込み
IR[12]	タイマ/カウンタチャンネル0 割り込み
IR[11]	UART ブレーク検出/エラー検出割り込み (チャンネル0)
IR[10]	UART RXRDY 出力 (チャンネル0)
IR[9]	UART TXRDY 出力 (チャンネル0)
IR[8]	クロック同期シリアルI/Oの送受信割り込み (チャンネル0)
IR[7]	DMA コントローラのDMTC1
IR[6]	DMA コントローラのDMTC0
IR[5]	UART ブレーク検出/エラー検出割り込み (チャンネル1)/ 外部入力 P21
IR[4]	UART RXRDY 出力 (チャンネル1)
IR[3]	UART TXRDY 出力 (チャンネル1)
IR[2]	クロック同期シリアルI/Oの送受信割り込み (チャンネル1)/ 外部入力 P20
IR[1]	タイマ/カウンタ チャンネル3 割り込み/ 外部入力 P21
IR[0]	タイマ/カウンタ チャンネル2 割り込み/ 外部入力 P20

### 6.3 レジスタの構成とI/Oレジスタマッピング

KP69 は以下のレジスタを備えています。

表 6-2 I/O レジスタマッピング

I/O アドレス	ライト時	リード時
34H	LERL/PGRL	ISRL
35H	LERH/PGRH	ISRH
36H	IMRL	IMRL
37H	IVR / IMRH	IMRH

### IRR (Interrupt Request Register)

書き込み不可／読み出し不可

割り込み要求が発生すると該当するビットがセットされます。割り込み要求の受付がレベルモードの場合は要求の消失によって、またエッジモードの場合は割り込みサービス開始によってリセットされます。リセット直後の初期状態では全ての割り込みレベルのエッジ検出回路がリセットされます。

### ISR (In Service Register)

書き込み不可／読み出し可

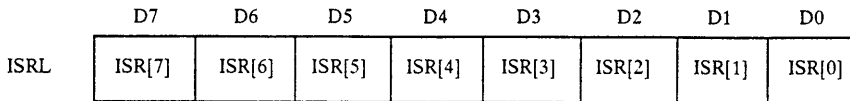
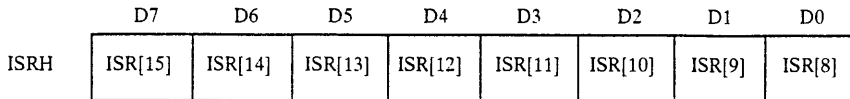
IACK\_ 信号が返って来ると、受け付けた割り込み要求の割り込みレベルに該当するビットがセットされま

す。割り込みサービスが終了するとリセットされます。リセット直後の初期状態では全てのビットがリセットされます。8ビットごとの読み出しが出来ます。

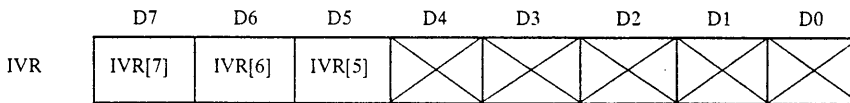
### IVR (Interrupt Vector Register)

書き込み可／読み出し不可

KC82 のモード 2 割り込みの割り込みベクタの上位 3 bit をこのレジスタで指定します。このレジスタに書き込みを行なう前と後で、書き込み可能なレジスタが変化します。



ISR[n]	意味
0	非サービス中
1	サービス中



### LER (Level / Edge Register)

書き込み可／読み出し不可

割り込み要求入力(以下IR入力とします。)のレベル／エッジのモード切替を制御します。各IR入力ごとに設定可能です。リセット直後の初期状態では全てのビットがレベルモードになります。なお、このレジスタへの書き込みはIVR設定“前”に行なって下さい。このレジスタは書き込み専用です。

注意: タイマ/カウンタの割り込みを使用する場合は対応する IR 入力をエッジモードに設定する必要があります。

### PGR (Priority Group Register)

書き込み可／読み出し不可

IR 入力の優先順位グループを設定します。各 IR 入力ごとに設定できます。優先順位グループには“HIGH”と“LOW”の 2 種類があります。リセット直後の初期状態では全てのビットが“LOW”に設定されます。なお、このレジスタへの書き込みはIVR設定“後”に行なって下さい。このレジスタは書き込み専用です

	D7	D6	D5	D4	D3	D2	D1	D0
LERH	LER[15]	LER[14]	LER[13]	LER[12]	LER[11]	LER[10]	LER[9]	LER[8]

	D7	D6	D5	D4	D3	D2	D1	D0
LERL	LER[7]	LER[6]	LER[5]	LER[4]	LER[3]	LER[2]	LER[1]	LER[0]

LER[n]	意味
0	LEVEL mode
1	EDGE mode

	D7	D6	D5	D4	D3	D2	D1	D0
PGRH	PGR[15]	PGR[14]	PGR[13]	PGR[12]	PGR[11]	PGR[10]	PGR[9]	PGR[8]

	D7	D6	D5	D4	D3	D2	D1	D0
PGRL	PGR[7]	PGR[6]	PGR[5]	PGR[4]	PGR[3]	PGR[2]	PGR[1]	PGR[0]

PGR[n]	意味
0	"LOW" group
1	"HIGH" group

### IMR (Interrupt Mask Register)

書き込み可／読み出し可

IR入力のマスク状態を設定します。各IR入力ごとに設定できます。リセット直後の初期状態では全てのビットがマスクセット状態になっています。なお、このレジスタへの書き込みはIVR設定“後”に行なって下さい。このレジスタは書き込み／読み出し可能です。

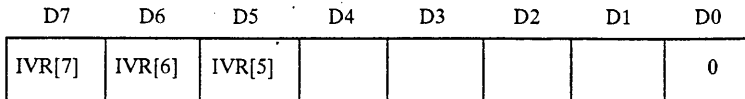
	D7	D6	D5	D4	D3	D2	D1	D0
IMRH	IMR[15]	IMR[14]	IMR[13]	IMR[12]	IMR[11]	IMR[10]	IMR[9]	IMR[8]

	D7	D6	D5	D4	D3	D2	D1	D0
IMRL	IMR[7]	IMR[6]	IMR[5]	IMR[4]	IMR[3]	IMR[2]	IMR[1]	IMR[0]

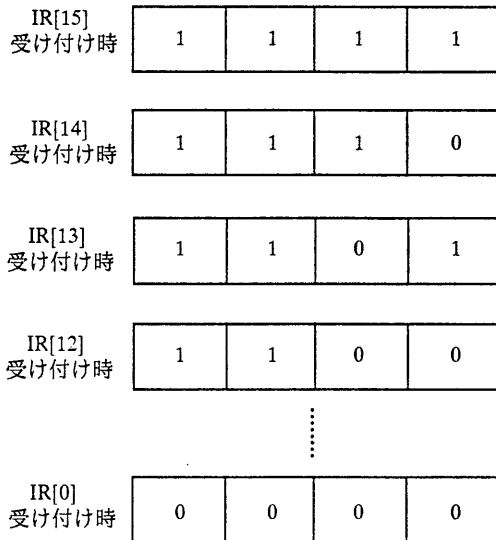
IMR[n]	意味
0	非マスク状態
1	マスク状態

### 6.4 割り込みベクタ出力

割り込みベクタの出力フォーマットを示します。  
 IACK\_ に応答してデータバスに出力される割り込みベクタは図 6-2 のようになります。



割り込み要求入力 (IR[n]) のビット番号に応じて出力



ビット 7～5 : IVR に設定したデータ  
 ビット 4～1 : 受け付けた割り込みレベルを 2 進数にしたコード  
 ビット 0 : 0 に固定

図 6-2 割り込みベクタ

このように割り込みベクタは、割り込み要求入力のビット番号が大きくなるにしたがって大きくなる連続した偶数値を出力します。

### 6.5 割り込み要求の優先順位

KP69 では各割り込み要求入力の優先順位は、リセット直後の初期状態では番号が大きい順となっています。また、優先順位グループという考え方をを用いて、優先順位の変更を可能としています。優先順位グループには“LOW”と“HIGH”の2つがあり、各IRごとに設定可能です。“HIGH”グループのIRの方が“LOW”グループのIRよりも優先順位が高く、また各グループ内ではIRのビット番号が大きい方ほど優先順位が高くなっています。以下に例を挙げておきます。

割り込み要求入力	グループ	優先順位
IR[15]	LOW	
IR[14]	LOW	
IR[13]	LOW	
IR[12]	HIGH	
IR[11]	LOW	
IR[10]	HIGH	
IR[9]	LOW	
IR[8]	HIGH	
IR[7]	HIGH	
IR[6]	LOW	
IR[5]	LOW	
IR[4]	HIGH	
IR[3]	LOW	
IR[2]	HIGH	
IR[1]	HIGH	
IR[0]	HIGH	

⇒

割り込み要求入力	グループ	優先順位
IR[12]	HIGH	最高
IR[10]	HIGH	↓
IR[8]	HIGH	↓
IR[7]	HIGH	↓
IR[4]	HIGH	↓
IR[2]	HIGH	↓
IR[1]	HIGH	↓
IR[0]	HIGH	↓
IR[15]	LOW	↓
IR[14]	LOW	↓
IR[13]	LOW	↓
IR[11]	LOW	↓
IR[9]	LOW	↓
IR[6]	LOW	↓
IR[5]	LOW	↓
IR[3]	LOW	最低

図 6-3 割り込み優先順位

### 6.6 レジスタ設定シーケンス

KP69 の内部レジスタは、表 6-2 のように I/O アドレスが割り付けられています。したがって、リセット解除後の各レジスタの設定は、LER → IVR → IMR(または PGR) という順に行ないます。IVR を設定した後は IMR および PGR の設定のみ可能となりますのでご注意ください。

### 6.7 レジスタの読み出し

KP69 のレジスタのうち、ISR と IMR は常時読みだし可能です。不正割り込み判定のためには ISR の読み出しを行なう必要があります。

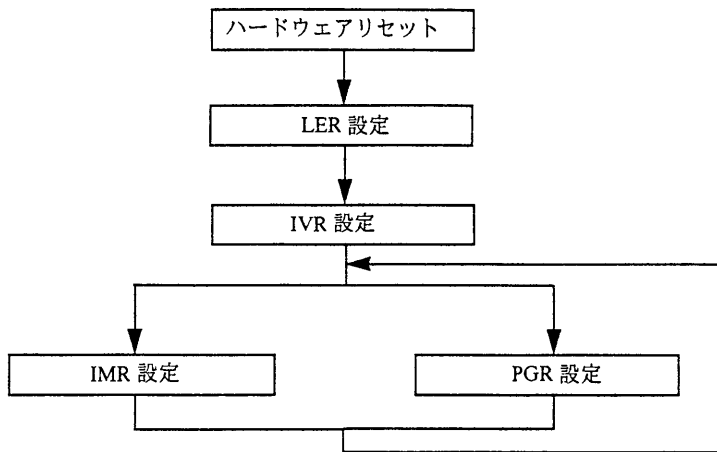


図 6-4 レジスタの設定シーケンス

### 6.8 割り込み要求の受け付け

割り込み要求の受け付けにはレベル/エッジの2種類のモードがあり、LERで設定します。

#### レベルモード

レベルモードではIR入力の“H”レベルを割り込み要求として認識します。リセット後LER未設定の状態ではこのモードです。

#### エッジモード

エッジモードではIR入力の立ち上がりを割り込み要求として認識します。この場合その割り込み要求は受け付けられるまで保持されます。

端子 86~89 から入力される外部割り込み要求のレベル/エッジモードの極性は SCR1 の設定によりビット単位で変更できます。詳しくは 12 章を参照下さい。

### 動作シーケンス

IR[n]端子への割り込み要求が発生した時の動作説明をします。

IR入力が“H”になり割り込み要求が発生すると、IRRの該当ビットがセットされます。この割り込み要求はIMRによるマスクやISR、PGRによる優先順位の判断を経てINT\_信号になります。KC82はそれを受けてIACK\_信号を“L”にします。IACK\_を受けるとKP69は割り込み要求の発生したIR[n]に対応する割り込みベクタを出力し、対応するISRのビットのセットとIRRのビットのリセット（エッジモードの場合）を行ないます。これで割り込みサービス状態に入ります。ISRのセットが行なわれた時点で次のエッジの割り込み要求の受付が可能になります。

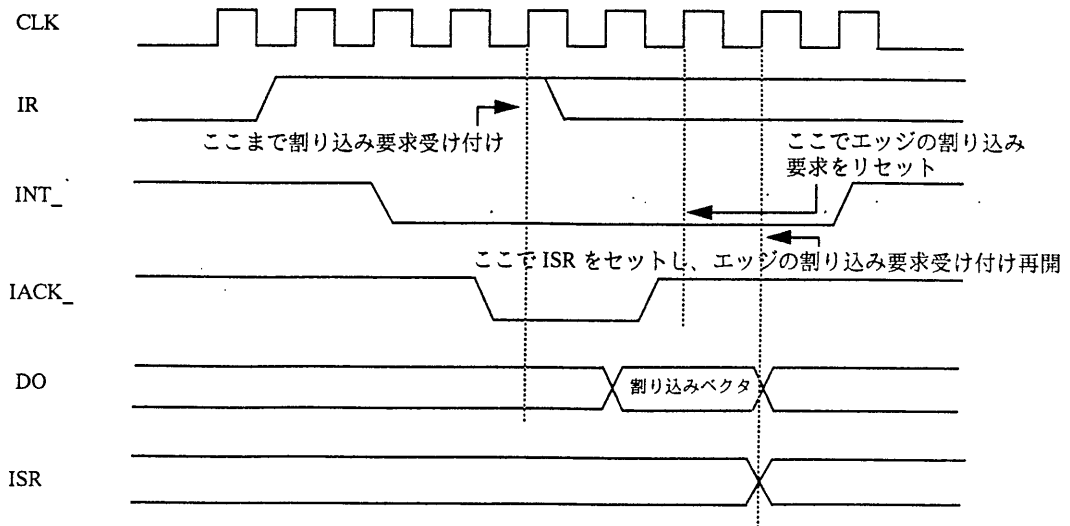


図 6-5 割り込み要求の受け付けタイミング

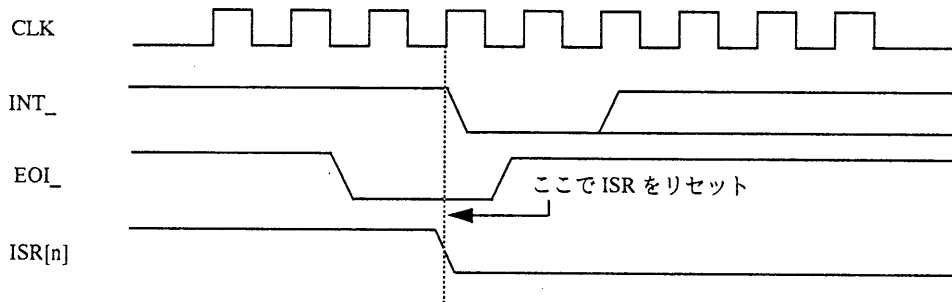


図 6-6 割り込み終了タイミング

### 6.9 割り込みの終了

KP69 は KC82 からの EOI\_ 信号が “L” になることにより、現在サービス中の割り込みの中で最優先レベルに相当する ISR ビットをリセットします。これによりそのレベルの割り込みサービスが終了したことになります。

### 6.10 多重割り込み処理

KP69 は割り込み要求のあった割り込みレベルと現在サービス中の割り込みレベルをあらかじめ設定された優先順位で比較します。そのとき新たに要求のあった割り込みレベルが現在サービス中の割り込みレベルより優先順位が高ければ新しい割り込み要求を受け付け、優先順位が同じかあるいは低ければ割り込み要求を拒否あるいは待たせます。これによって多重割り込

み処理を実現しています。多重割り込みの状況は ISR を読み出して知ることが出来ます。

### 6.11 不正割り込み処理

不正割り込みとは、レベルモードに設定されている割り込み要求入力において、割り込み要求の発生によって INT<sub>0</sub> が“L”になり、それを受けて KC82 が IACK<sub>0</sub> を返した際に、すでにその割り込み要求がなくなってしまう、しかも「現在サービス中の割り込みのうちで優先順位が最高のも」より上位の割り込み要求入力がない場合をいいます。このとき、KP69はISRのセットは行なわず、IR[0]に相当する割り込みベクタを出力し不正割り込み処理状態になります。この状態はKC82からのEOI<sub>0</sub>によって不正割り込み処理ルーチンが終了するまで続きます。このときのEOI<sub>0</sub>においては、ISR はリセットされません。また不正割り込み処理状態においては他のいかなる割り込み要求（正常、不正）も受け付けません。このため、IR[0]の割り込み処理ルーチン内では、ISR[0]がセットされているかどうかによって、正常な割り込み要求だったのか不正な割り込み要求だったのかを判断し処理を分ける必要があります。なお、IR[0]の割り込みサービス中に不正割り込み処理状態になった場合、ISR を読み出すとISR[0]='0'となります。したがって、ISR[0]='1'ならば、正常な割り込み要求、ISR[0]='0'ならば不正割り込み要求、と区別することが可能となっています。

### 6.12 リセット

RESET\_端子を“L”レベルにすると、次のように動作します。

- (1) IMR を 'FFFFH' にセットします。（全レベルマスク状態）
- (2) IRR、ISR、LER、PGR を '0000H' にリセットします。
- (3) 不正割り込み処理状態を無効にします。
- (4) IVR を未設定状態にします。（再設定が必要です。）

### 6.13 注意事項

- (1) CPU の割り込みモードはモード 2 を使って下さい。
- (2) 割り込み処理ルーチンの最後には必ず RETI 命令（コード ED 4D）を置いて下さい。
- (3) タイマ/カウンタの割り込みを使用する場合は対応する IR 入力をエッジモードに設定する必要があります。

### 6.14 応用 (ピン 86 ~ 89 の使い方)

外部端子 86~89 はパラレルポートとマルチプレクスされています。この外部端子 86~89 の入出力設定はパラレルポート入出力制御レジスタによって制御されます。一方、割り込み要求 IR[0]、IR[1]、IR[2]、IR[5] の割り込み要求源の選択は SCR1 によって制御されます。外部端子 86,87 と割り込みコントローラの割り込み要求の回路を図 6-7 に示します。

#### 使用上の注意

- (1) 外部割り込み要求を使用する場合、パラレルポートは入力方向に設定して下さい。
- (2) 外部端子 86,87 をパラレルポート出力として使う場合、IR[14]、IR[15] をマスクして下さい。
- (3) 外部端子 88,89 をパラレルポート出力として使う場合、IR[0]、IR[1]、IR[2]、IR[5] をマスクするか割り込み要求源として内部割り込みを選択して下さい。

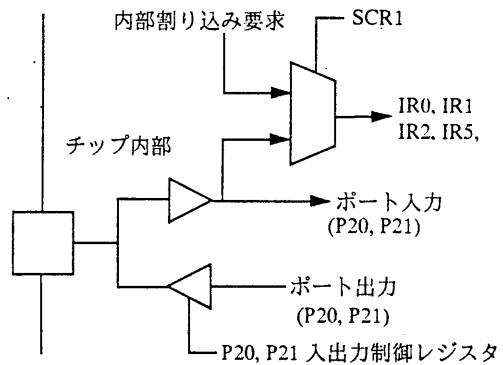


図 6-7 ピン 88, 89 の構造

### 7. DMA コントローラ

#### 7.1 概要

KL5C80A16 は DMA コントローラとして KP27 マクロセルを内蔵しています。KP27 は弊社オリジナルの小型プログラマブル DMA コントローラで、2つの独立した DMA チャンネルを持っています。各チャンネル毎にメモリ→メモリ転送、メモリ→I/O 転送、I/O →メモリ転送の設定が可能です。

メモリ・メモリ間の転送はソフトウェア・リクエストにより起動されます。またメモリ・I/O 間の転送では、DMA 要求信号の有効極性、転送モード (シングル転送/デマンド転送) 等を設定することが出来ます。転送はアドレスを 2 系統出力する方式で行ないますが、アドレスラッチ信号によりラッチにアドレス等を保持することにより、高速な転送を可能としています。また、外部に DMA 機能を持ったデバイスを接続可能で、この時そのデバイスのバス権要求信号と KP27 の各チャンネルへの DMA 要求信号を調停します。さらに NMI による DMA 動作の中断/再開機能を備えています。

KP27 は DMA 要求あるいは外部 DMA デバイスからのバス権要求があると各チャンネルのイネーブル状態、優先順位を判定し CPU に対しバス権を要求します。CPU がバスを解放するとバス権を取得して DMA 動作を実行、あるいは外部 DMA デバイスにバス権を渡してその外部 DMA デバイスの動作終了を待ちます。DMA 動作が終了したらバス権を CPU に返します。

#### 特徴

- ・クロック同期式の DMA コントローラです。クロックレートは最高 10MHz です。
- ・MMU を介さずに 1Mbyte のメモリ空間にアクセス可能です。
- ・アドレスを 2 系統出力して転送を行なう方式の DMA なので、DMA 対象の I/O アドレスがプログラムで指定可能です。
- ・DACK<sub>1</sub> 信号を出力可能なので、DACK<sub>1</sub> 信号による転送対象の I/O 指定も可能です。
- ・優先度は、チャンネル 1 >チャンネル 0 となっています。
- ・各チャンネル毎にメモリ→メモリ転送、メモリ→I/O 転送、I/O →メモリ転送の設定が可能です。
- ・メモリ→I/O 転送、I/O →メモリ転送設定時は、さらにシングル転送/デマンド転送、DREQ の有効極性、Auto Initialize 機能が設定可能です。
- ・最大で 64kbyte の連続転送が可能です。
- ・3 クロックで 1 転送を実行します。したがって最大転送レートは 3.3Mbyte/sec となります。
- ・外部に DMA 機能を持ったデバイスを接続可能です。その際、外部 DMA デバイスの優先度を設定可能です。
- ・NMI による DMA 動作中断/再開が可能です。
- ・チャンネル 1 は SCR2 の設定により UART のチャンネル 0 専用として使用できます。

### 7.2 ブロック図

DMA コントローラの全体ブロック図を以下に示します。図中の各信号については次の 7.3 節を参照して下さい。

図 7-1 中の DMTC1、DMTC0 出力は割り込みコントローラの IR[7]、IR[6] にそれぞれ接続されています。また、バス権要求信号は CPU の BREQ\_ 入力に、バス権承認信号入力は CPU の BACK\_ 出力に内部で接続されています。NMI\_ 入力には CPU の NMI\_ 入力と同じ信号が接続されています。

DREQ0 には、パラレルポート P24 を入力方向に設定することにより、端子 85 から入力される信号を接続できます。DREQ1 には、表 7-1 の通り、端子 84 から入力される外部信号、または KP61 からの TXRDY0、

RXRDY0 信号のいずれかを SCR2 の設定によって内部で接続可能です。

また、パラレルポート P26 を入力方向に設定することにより、端子 83 からの入力を EXBREQ\_ として使用できます。この時、EXBACK\_ は SCR2 の設定によって端子 91 と接続されます。

KP27 は転送対象の I/O 指定を DACK\_ 信号により行なうことも出来ます。DACK0\_ は PALAT と多重化されて端子 5 に割り当てられています。この切替は SCR2 で行ないます。また DACK1\_ も SCR2 の設定によって端子 92 から出力することが出来ます。

詳細は 12 章を参照して下さい。

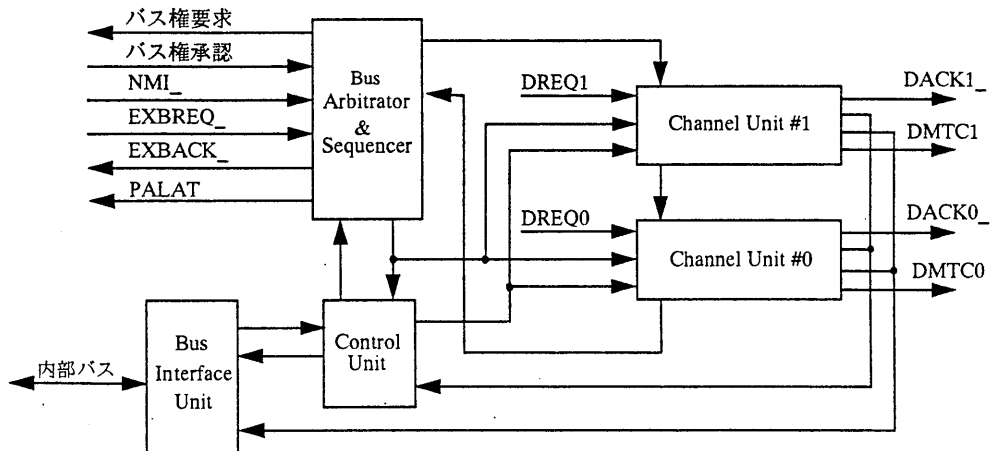


図 7-1 DMA コントローラブロック図

表 7-1 DREQ1 に内部で接続可能な DMA 要求源

DMA 要求源	SCR2 D<1:0>	端子 92 の機能
端子 84	00	P16
	01	DACK1_
UART TXRDY0	10	P16
UART RXRDY0	11	P16

### 7.3 端子機能説明

端子名	I/O	機能説明
DREQ0 DREQ1	I	DMA 要求信号入力端子 各チャンネルに対する DMA 要求信号を入力します。有効極性は“H(↑エッジ)”または“L(↓エッジ)”を設定可能です。
DACK0_ DACK1_	O	DMA 要求アックノリッジ信号出力端子 DREQ を受けて、DMA 対象 I/O デバイスに対するアックノリッジ信号を出力します。“L”で DMA 要求承認となります。
EXBREQ_	I	外部デバイスバス権要求信号入力端子 外部に DMA 機能を持ったデバイスを接続する場合、そのバス権要求信号を入力します。有効極性は“L”です。
EXBACK_	O	外部デバイスバス権要求アックノリッジ信号出力端子 外部に DMA 機能を持ったデバイスを接続する場合、そのデバイスのバス権要求アックノリッジ信号入力端子に接続します。“L”でバス権要求承認となります。
PALAT	O	ペリフェラルアドレス等ラッチ信号出力端子 この信号はメモリ・I/O 間の DMA 転送時、アドレスバスに出力される値が I/O アドレスからメモリアドレスに切り替わる直前に“H”から“L”に変化します。また、その 2 バスサイクル後に“L”から“H”に変化します。この信号を用いて外部ラッチに I/O アドレス等をラッチし、それを各 I/O デバイスに入力することによって、プログラマブルに転送対象の I/O デバイスを選択出来ます。
DMTC0 DMTC1	O	ターミナルカウント出力端子 各チャンネルが TC 状態に達したことを示す信号です。“H”で TC 状態を示します。TC 状態は、あらかじめ設定した転送バイト数の転送が実行された状態です。この信号は、DMTC0 が割り込みコントローラの IR[6] に、DMTC1 が IR[7] にそれぞれ内部で接続されています。

注) 端子名の 0,1 はそれぞれチャンネル 0、チャンネル 1 を表します。

### 7.4 内部レジスタマッピング

KP27 の内部レジスタのマッピングを次ページに示します。各レジスタの機能については、次の 7.5 節を参照して下さい。また、表 7-2 中の BSFF は、各内部レジスタを 8 ビットごとにリード/ライトするために設け

た 2 ビットのカウンタです。これについても 7.5 節の最後に説明してありますので、そちらを参照して下さい。

表7-2 I/Oレジスタマッピング

I/O アドレス	ライト時	リード時	BSFF
10H	チャンネル0 B-PAR0	チャンネル0 C-PAR0	00
	チャンネル0 B-PAR1	チャンネル0 C-PAR1	01
	チャンネル0 B-PAR2	チャンネル0 C-PAR2	10
11H	チャンネル0 B-SAR0	チャンネル0 C-SAR0	00
	チャンネル0 B-SAR1	チャンネル0 C-SAR1	01
	チャンネル0 B-SAR2	チャンネル0 C-SAR2	10
12H	チャンネル0 B-BCR0	チャンネル0 C-BCR0	00
	チャンネル0 B-BCR1	チャンネル0 C-BCR1	01
13H	チャンネル0 CR	チャンネル0 SR0	00
		チャンネル0 SR1	01(リード時のみ)
14H	チャンネル1 B-PAR0	チャンネル1 C-PAR0	00
	チャンネル1 B-PAR1	チャンネル1 C-PAR1	01
	チャンネル1 B-PAR2	チャンネル1 C-PAR2	10
15H	チャンネル1 B-SAR0	チャンネル1 C-SAR0	00
	チャンネル1 B-SAR1	チャンネル1 C-SAR1	01
	チャンネル1 B-SAR2	チャンネル1 C-SAR2	10
16H	チャンネル1 B-BCR0	チャンネル1 C-BCR0	00
	チャンネル1 B-BCR1	チャンネル1 C-BCR1	01
17H	チャンネル1 CR	チャンネル1 SR0	00
		チャンネル1 SR1	01(リード時のみ)

表7-2において、PAR, SAR, BCRの各レジスタ名の最後の数字は、これら8ビットを越える幅のレジスタを分割して表すためのものです。'0'は各レジスタの<7:0>を、'1'は<15:8>を、'2'は<19:16>を示しています。また、“B-PAR0”、“C-PAR0”等の“B”、“C”はそれぞれ“Base”、“Current”を表します。詳しくは次の7.5節を参照して下さい。

また、表7-2中のCR（コマンドレジスタ）は次の表7-3のようになっています。KP27は、書き込みデータのMSB側数ビットによって、書き込みを行なうコマンドレジスタを選択します。以下にその対応表を示します。各レジスタのフォーマットについては7.5節を参照して下さい。

表7-3 コマンドレジスタのマッピング

I/O アドレス	データ	コマンドレジスタ
13H	D<7> = 0	チャンネル0 CR0
	D<7:6> = 10	チャンネル0 CR1
17H	D<7> = 0	チャンネル1 CR0
	D<7:6> = 10	チャンネル1 CR1
13H または 17H	D<7:5> = 110	CR2（共通コマンドレジスタ）
	D<7:4> = 1111	NCC（共通コマンドレジスタ）

### 7.5 内部レジスタの構成

KP27は大きく分けて次のような2種類の内部レジスタを備えています。

- 各チャンネル毎に存在する内部レジスタ。
- チャンネルに依存しない、共通の内部レジスタ。

このそれぞれについて、以下に説明します。

なお、KP27はDMA転送時、アドレスを2系統出力します。時間順に先に出力される方をPrimary Address、後に出力される方をSecondary Addressと呼びます。メモリ→I/O転送、I/O→メモリ転送の場合、

PA=I/Oアドレス

SA=メモリアドレス

となり、メモリ→メモリ転送の場合、

PA=転送元メモリアドレス、

SA=転送先メモリアドレス

となります。

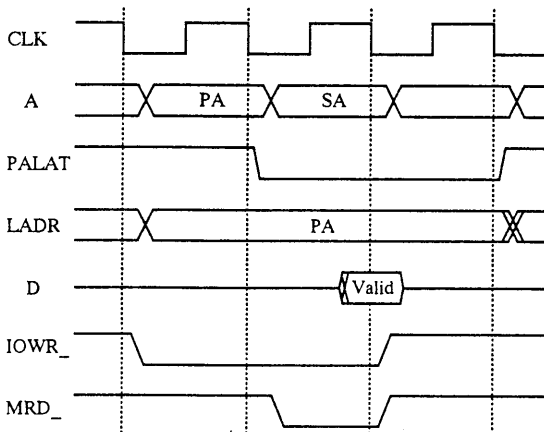


図 7-2 KP27 の基本 DMA 転送タイミング  
(メモリ→チップ内部 I/O)

図7-2中のLADRは、A(アドレス)をPALATでラッチした信号です。また、この図ではI/Oアクセス信号やメモリアccess信号は内部バス信号としてあります。

### ●各チャンネル毎に存在する内部レジスタ

各チャンネル毎に存在する内部レジスタには、PAR, SAR, BCR, CR0, CR1, SR0, SR1があります。このうちPAR, SAR, BCRにはBase(ベース)とCurrent(カレント)の2種類があります。

ベースレジスタは初期値を書き込んで設定するレジスタであり、カレントレジスタはベースレジスタから自動的に初期値が設定された後は、動作中のアドレスや残り転送バイト数を保持するレジスタです。ベースレジスタからカレントレジスタへの初期値のロードは、

- ・ベースレジスタへの書き込み発生時
- ・CR0, CR1(Command Register. 後述)への書き込み発生時
- ・メモリ→I/O転送、I/O→メモリ転送時はAuto Initialize(7.6節参照)有効のときのTC(7.6節参照)到達後

に、PAR, SAR, BCR同時に起こります。

したがって、あるチャンネルの3種類のベースレジスタ(B-PAR, B-SAR, B-BCR)のいずれかにデータライトを行なうと、これによってそのチャンネルの3種類のカレントレジスタ(C-PAR, C-SAR, C-BCR)全てに、そのときの各ベースレジスタの内容がロードされます。同時にそのチャンネルは自動的にディセーブル状態になります。

例えばチャンネル1のB-PARにライトを行なうと、これによってチャンネル1のC-PAR, C-SAR, C-BCRには、それぞれそのときのチャンネル1のB-PAR, B-SAR, B-BCRの内容がロードされ、チャンネル1はディセーブル状態になります。

ただし、このロードは同一チャンネル内でのみ発生します。例えばチャンネル1のベースレジスタに初期値をライトしても、チャンネル0には何ら影響を与えません。

また、共通内部レジスタ CR2, NCCへの書き込みはいかなるチャンネルのいかなる内部レジスタの内容へも影響を与えません。

### B-PAR (Base Primary Address Register)

書き込み可／読み出し不可 20bit

Primary Addressの初期値を保持するレジスタです。メモリ→I/O 転送、I/O→メモリ転送ではI/O アドレスを、メモリ→メモリ転送では転送元メモリアドレスの初期値を設定します。転送対象のI/Oの指定に DACK\_信号を使用する場合も、このレジスタのデータがアドレスバス上に出力されます。

リセット直後は全ビット'0'となります。

	D7	D6	D5	D4	D3	D2	D1	D0
B-PAR2	0	0	0	0				
B-PAR1								
B-PAR0								

### C-PAR (Current Primary Address Register)

書き込み不可／読み出し可 20bit

DMA動作中のPrimary Addressを保持するレジスタです。

B-PAR の設定を行なった際、その値が同時に設定されます。

メモリ→メモリ転送設定時は、ソフトウェア・リクエスト実行後、現在までに転送を終了したメモリアドレスの次のメモリアドレス、すなわち次のDMA転送でデータリードを行なうメモリアドレスが保持されています。各ベースレジスタやCR0, CR1へのライト直後は、B-PARに設定した転送元メモリアドレス(初期値)が自動的に再設定されます。

メモリ→I/O 転送、I/O→メモリ転送設定時はB-PARに設定したI/O アドレスが保持されます。

リセット直後は全ビット'0'となります。

	D7	D6	D5	D4	D3	D2	D1	D0
C-PAR2	0	0	0	0				
C-PAR1								
C-PAR0								

### B-SAR (Base Secondary Address Register)

書き込み可／読み出し不可 20bit

Secondary Addressの初期値を保持するレジスタです。メモリ→I/O 転送、I/O→メモリ転送ではメモリアドレスの初期値を、メモリ→メモリ転送では転送先メモリアドレスの初期値を設定します。

リセット直後は全ビット'0'となります。

	D7	D6	D5	D4	D3	D2	D1	D0
B-SAR2	0	0	0	0				
B-SAR1								
B-SAR0								

### C-SAR (Current Secondary Address Register)

書き込み不可／読み出し可 20bit

DMA動作中のSecondary Addressを保持するレジスタです。

B-SAR の設定を行なった際、その値が同時に設定されます。

メモリ→I/O転送、I/O→メモリ転送、メモリ→メモリ転送いずれの場合でも、TC 未達時には、現在までに転送を終了したメモリアドレスの次のメモリアドレス、すなわち次のDMA転送でアクセスを行なうメモリアドレスが保持されています。

メモリ→メモリ転送時は各ベースレジスタやCR0, CR1へのライト直後に、またメモリ→I/O 転送、I/O→メモリ転送時は Auto Initialize有効時の TC 到達後あるいは各ベースレジスタやCR0, CR1へのライト直後に、B-SARに設定したメモリアドレス(初期値)が自動的に再設定されます。

リセット直後は全ビット'0'となります。

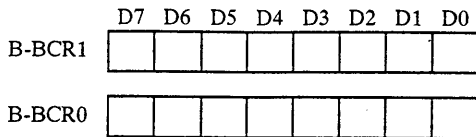
	D7	D6	D5	D4	D3	D2	D1	D0
C-SAR2	0	0	0	0				
C-SAR1								
C-SAR0								

### B-BCR (Base Byte Count Register)

書き込み可／読み出し不可 16bit

DMA転送時の総転送バイト数を設定するレジスタです。このレジスタに設定したバイト数の転送を行なうとTC状態となります。

リセット直後は全ビット'0'となります。



### C-BCR (Current Byte Count Register)

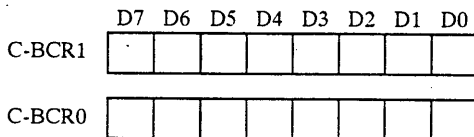
書き込み不可／読み出し可 16bit

DMA転送時の残り転送バイト数を保持するレジスタです。

B-BCR の設定を行なった際、その値が同時に設定されます。メモリ→I/O転送、I/O→メモリ転送、メモリ→メモリ転送いずれの場合でも、TC未達時には残り転送バイト数が保持されています。

メモリ→メモリ転送時は各ベースレジスタやCR0, CR1へのライト直後に、またメモリ→I/O転送、I/O→メモリ転送時はAuto Initialize有効時のTC到達後、あるいは各ベースレジスタやCR0, CR1へのライト直後に、B-BCRに設定した総転送バイト数が自動的に再設定されます。

リセット直後は全ビット'0'となります。



### CR0 (Command Register 0)

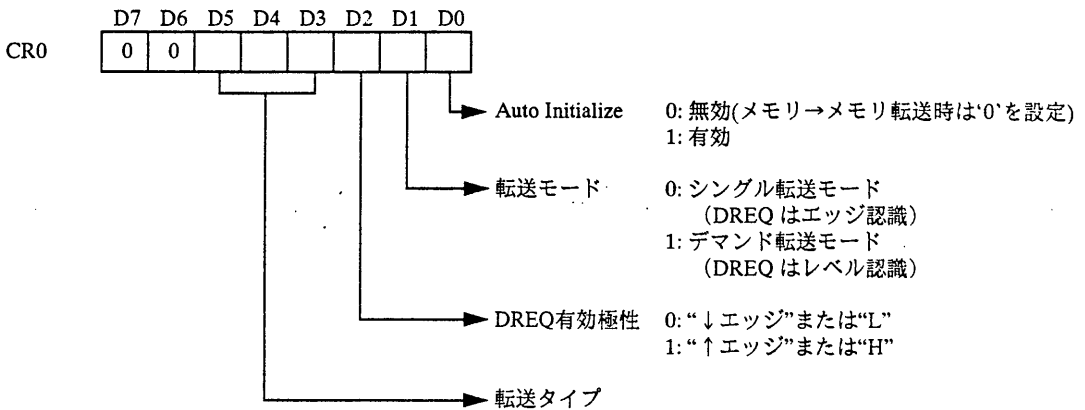
書き込み可／読み出し不可 8bit

各チャンネル毎の転送タイプや転送モードなどを設定記憶するレジスタです。書き込みデータの上位2ビット

トを'00'にすることにより、このレジスタが選択されます。各ビットの意味は次ページの図の通りです。

“転送モード”でシングル転送を選択した時は、“DREQ極性”は“立ち下がりエッジ”または“立ち上がりエッジ”が選択できます。また、デマンド転送を選択した場合、“DREQ極性”は“Lレベル”または“Hレベル”が選択できます。

D<5:4>=00のとき、D<2:1>はdon't careです。すなわち、転送タイプでメモリ→メモリ転送を選択した場合、DREQ極性、転送モードの設定は無視されます。また、このときAuto Initializeは無効に設定して下さい。



D5	D4	D3	転送タイプ	メモリアドレス
0	0	0	メモリ→メモリ	increment
0	0	1	メモリ→メモリ	decrement
0	1	0	メモリ→I/O	increment
0	1	1	メモリ→I/O	decrement
1	0	0	I/O→メモリ	increment
1	0	1	I/O→メモリ	decrement
1	1	X	禁止	_____

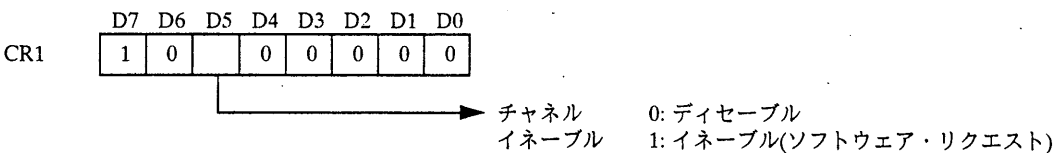
### CR1 (Command Register 1)

書き込み可/読み出し不可    8bit  
(有効ビットはD<7:5>)

各チャンネルのDMA 転送イネーブル/ディセーブルを設定記憶するレジスタです。

書き込みデータの上位2ビットを'10'にすることにより、このレジスタが選択されます。各ビットの意味は以下の通りです。➤

CR0でメモリ→メモリ転送に設定した場合、CR1はソフトウェア・リクエスト・レジスタとなります。すなわち、メモリ→メモリ転送に設定したチャンネルのCR1に'A0H'を書き込むことで、ソフトウェア・リクエストが実行されます。

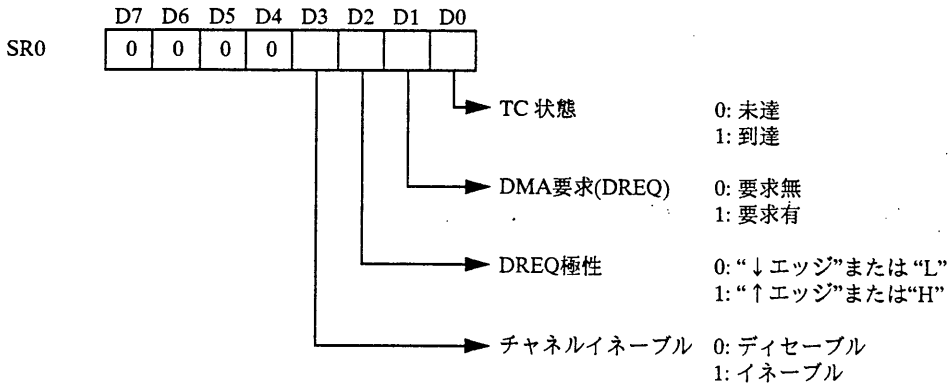


### SR0 (Status Register 0)

書き込み不可/読み出し可    8bit  
(有効ビットはD<3:0>)

各チャンネル毎のステータスを表すレジスタです。D<7:4>からは'0000'が読み出されます。このレジスタからは主に各チャンネルの現在の動作状態を知ることが

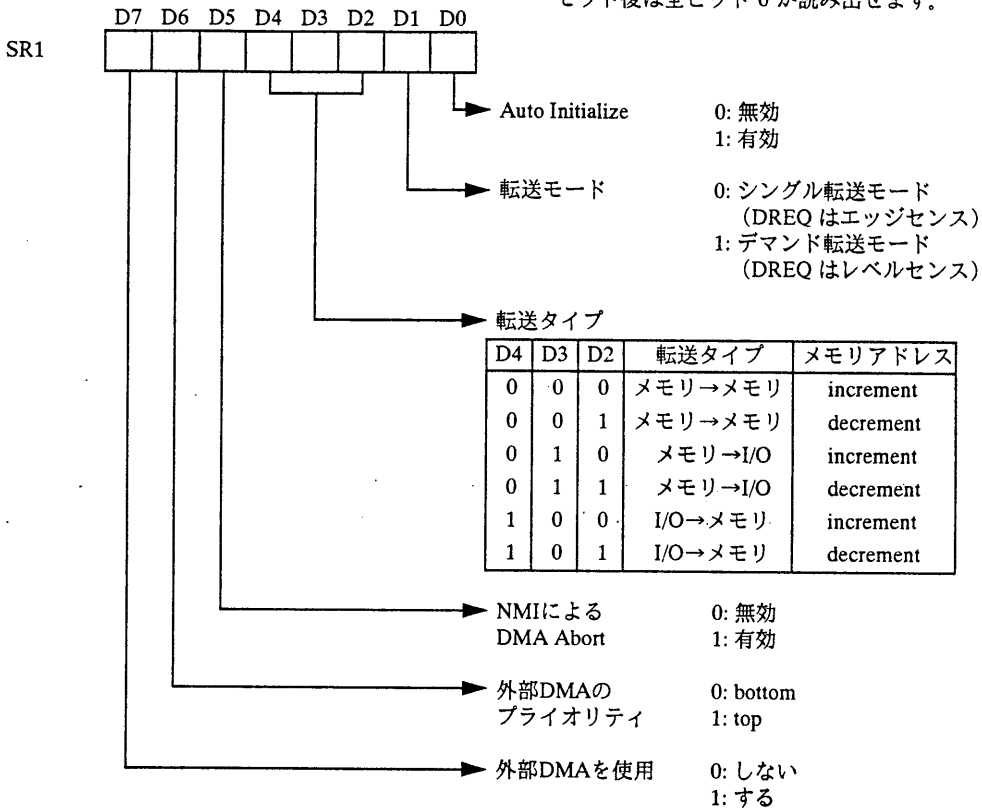
できます。各ビットの意味は次ページの図の通りです。また、リセット直後は全ビット'0'が読み出せます。



### SR1 (Status Register 1)

書き込み不可/読み出し可 8bit

各チャンネル毎のステータスおよびKP27の動作設定を、また D<4:0>はCR0, CR1で設定した内容を、また D<7:5>はCR2(後述)で設定した内容を表示します。リセット後は全ビット'0'が読み出せます。



### ●チャンネルに依存しない、共通の内部レジスタ

以下に示す2種類の内部レジスタCR2, NCCはDMAチャンネルに依存しない、共通の内部レジスタです。チャンネル0、チャンネル1のいずれから書き込んでも構いません。

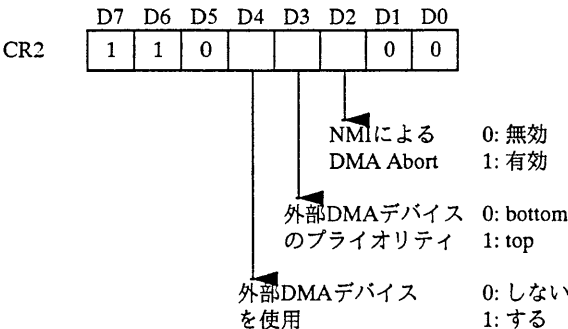
#### CR2 (Command Register 2)

書き込み可/読み出し不可 8bit  
(有効ビットはD<7:2>)

KP27のプログラム可能な動作のうち、チャンネルに依存しない部分の設定をするレジスタです。

D<7:5>='110' にすることにより、このレジスタが選択されます。各ビットの意味を以下の図に示します。

D3の外部DMAデバイスのプライオリティの設定は、D4で外部DMAデバイスを使用するように設定した時のみ有効です。



#### NCC (NMI Clear Command)

書き込み可/読み出し不可 8bit  
(有効ビットはD<7:4>)

CR2でNMIによるDMA Abortを有効に設定した場合、このコマンドによってそのNMIによる中断状態を解除し、DMA転送を再開します。D<7:0>='F0H'にして書き込むことで、このコマンドが実行されます。

NMIによるDMA Abortを無効に設定した場合、NCCコマンドを書き込んでもKP27の動作には何の影響もありません。

D7	D6	D5	D4	D3	D2	D1	D0
1	1	1	1	0	0	0	0

NCC

### BSFF について

KP27は、チャンネル0、チャンネル1にそれぞれ独立したBSFF(Byte Select Flip Flop)を備えています。BSFFとは、複数の内部レジスタが割り当てられているI/Oアドレスにアクセスを行なった時、どの内部レジスタにアクセスを行なうかを選択するための2ビットのカウンタです。PARやSAR、BCRといった16ビット、20ビット長のレジスタに対して8ビットごとにアクセスするために設けられています。

あるチャンネルのBSFFは以下の3つのルールにしたがって動作します。

◎直前のI/Oアクセス時と同じI/Oアドレスに同じアクセス（前回も今回もI/Oリード、あるいは前回も今回もI/Oライト）が行なわれた場合、そのアクセス直後にBSFFをインクリメントします。ただし、

\*A[1]='0'(PAR or SAR)かつBSFF='10'ならば、アクセス直後にBSFF='00'となります。

\*A[1]='1'(BCR or SR)かつBSFF='01'ならば、アクセス直後にBSFF='00'となります。

◎直前のI/Oアクセスと同一チャンネル内の異なるI/Oアドレスにアクセスが行なわれた場合、あるいは直前のI/Oアクセス時と同じI/Oアドレスに直前と異なるI/Oアクセスが行なわれた場合(前回I/Oリードで今回I/Oライト、またはその逆)、直ちにBSFF='00'となり、そのアクセス実行後にBSFFをインクリメントします。

◎A[1:0]='11'へのI/Oライト（CRへのライト）が行なわれた場合、直ちにBSFF='00'となります。

次にKP27の内部レジスタ設定の具体例を示します。この中でBSFF0、BSFF1とあるのはそれぞれチャンネル0およびチャンネル1のBSFFを表しています。またch0,ch1はそれぞれチャンネル0、チャンネル1を表しています。

例1 次の命令シーケンス例の場合、KP27の内部レジスタに対する動作は(1)~(3)のようになります。

```
LD    BC,  FF15H
OUT   (C),  B      ..... (1)
OUT   (C),  B      ..... (2)
OUT   (C),  B      ..... (3)
```

(1)BSFF1='00'で、A[3:0]='0101'にI/Oライト  
→ch1\_B-SAR0に書き込み。その後、BSFF1='01'。

(2)BSFF1='01'で、再びA[3:0]='0101'にI/Oライト  
→ch1\_B-SAR1に書き込み。その後、BSFF1='10'。

(3)BSFF1='10'で、再びA[3:0]='0101'にI/Oライト  
→ch1\_B-SAR2に書き込み。その後、BSFF1='00'。

(1)~(3)により、ch1\_B-SAR<19:0>に'FFFFFFH'が設定されます。

例2 次の命令シーケンス例の場合、KP27の内部レジスタに対する動作は(1)~(6)のようになります。

```
LD    A,  A0H
OUT   (13H), A      ..... (1)
XOR   A
OUT   (16H), A      ..... (2)
IN    A,  (11H)     ..... (3)
IN    A,  (11H)     ..... (4)
LD    A,  80h
OUT   (16H), A      ..... (5)
IN    A,  (10H)     ..... (6)
```

(1)BSFF0='00'で、A[3:0]='0011'にD='A0H'をI/Oライト  
→ch0\_CR1に書き込み。BSFF0='00'でそのまま。

(2)BSFF0='00'、BSFF1='00'で、A[3:0]='0110'にI/Oライト  
→ch1\_B-BCR0に書き込み。その後、BSFF0='00'、BSFF1='01'。

(3)BSFF0='00'、BSFF1='01'で、A[3:0]='0001'をI/Oリード  
→ch0\_C-SAR0を読み出し。その後、BSFF0='01'、BSFF1='01'。

(4)BSFF0='01'、BSFF1='01'で、A[3:0]='0001'を再I/Oリード  
→ch0\_C-SAR1を読み出し。その後、BSFF0='10'、BSFF1='01'。

(5)BSFF0='10', BSFF1='01'で、A[3:0]='0110'にI/Oライト

→ch1\_B-BCR1に書き込み。その後、BSFF0='10', BSFF1='00'。

(6)BSFF0='10', BSFF1='00'で、A[3:0]='0000'をI/Oリード

→直ちにBSFF0='00'となり、ch0\_C-PAR0を読み出し。その後、BSFF0='01', BSFF1='00'。

(1)により、ch0\_CR1の設定(ch0イネーブル)が行なわれます。また(2)~(6)によりch1\_B-BCRに'8000H'が設定され、ch0\_C-SAR<15:0>とch0\_C-PAR<7:0>の読み出しも行なわれます。

例3 次の命令シーケンス例の場合、KP27の内部レジスタに対する動作は(1)~(7)のようになります。

XOR	A		
LD	BC,	4014H	
LD	DE,	0E08H	
OUT	(C),	B	..... (1)
INC	C		
OUT	(C),	A	..... (2)
OUT	(C),	A	..... (3)
OUT	(C),	D	..... (4)
INC	C		
OUT	(C),	E	..... (5)
DEC	C		
IN	A,	(C)	..... (6)
ADD	A,	E	
OUT	(C),	A	..... (7)

(1)BSFF1='00'で、A[3:0]='0100'にI/Oライト  
→ch1\_B-PAR0に書き込み。その後、BSFF1='01'

(2)BSFF1='01'で、A[3:0]='0101'にI/Oライト  
→直ちにBSFF1='00'となってch1\_B-SAR0に書き込み。その後、BSFF1='01'

(3)BSFF1='01'で、A[3:0]='0101'にI/Oライト  
→ch1\_B-SAR1に書き込み。その後、BSFF1='10'

(4)BSFF1='10'で、A[3:0]='0101'にI/Oライト

→ch1\_B-SAR2に書き込み。その後、BSFF1='00'

(5)BSFF1='00'で、A[3:0]='0110'にI/Oライト  
→BSFF1='00'(不変)でch1\_B-BCR0に書き込み。その後、BSFF1='10'

(6)BSFF1='10'で、A[3:0]='0101'をI/Oリード  
→直ちにBSFF1='00'となってch1\_C-SAR0を読み出し。その後、BSFF1='01'

(7)BSFF1='01'で、A[3:0]='0101'にI/Oライト  
→直ちにBSFF1='00'となってch1\_B-SAR0に書き込み。その後、BSFF1='01'

(1)により、ch1\_PAR<19:0>に'00040H'が設定されます。(2)~(4)により、ch1\_SAR<19:0>に'E0000H'が設定されます。(5)により、ch1\_BCR<15:0>に'0008H'が設定されます。(6)により、ch1\_SAR<7:0>を読み出して演算し、その結果を(7)によってch1\_SAR<7:0>に書き戻します。

このようにして、例えばPARやSARにI/Oアクセスを行なう場合、I/Oアドレスを変更することなく同じアドレスに連続してI/Oリード、あるいは連続してI/Oライトすることによって、20ビット長や16ビット長のデータの読み書きが行なえます。

また同一チャンネル内では、直前のアクセスと異なったI/OアドレスにアクセスすることによってBSFFはクリアされますので、初期設定の時など、必要最小限の書き込みでレジスタの設定が行なえます。

### 7.6 KP27 の動作説明

#### 動作概要

KP27は1クロックを一つの基本単位として動作するクロック同期式のDMAコントローラです。以下にその動作の概要を説明します。

ここからの説明は図7-4~図7-6を合わせて御覧下さい。また、これらの図や以下の説明では、メモリアクセス信号やI/Oアクセス信号はMRD\_, MWR\_, IORD\_、

IOWR\_といった内部バス信号を想定しています。これらは外部バス・インターフェース・ユニットで外部バスサイクル信号に変換されます。この変換に関しては4章を参照して下さい。

KP27の動作の基本単位をステートと呼ぶことにします。KP27の動作は基本的に6つのステートから成り立っています。

まずKP27がCPUからのバス権承認信号を認識するまでの期間がステート0です。ステート0はKP27がバス権を持っていない期間とほぼ一致します。ステート0の期間のときにKP27の各内部レジスタへの初期値設定等を行いません。またチャネルイネーブル後でDREQ信号がアサートされた場合、KP27はCPUに対してバス権を要求し、要求が承認されるのを待ちます。

バス権要求がCPUによって承認されると、KP27はステート1へと移行します。このステートはこれから実行する1バイトの転送の準備期間です。このステートではKP27はMRD\_、MWR\_、IORD\_、IOWR\_信号を“H”とし、またDACK\_信号を“L”とします。複数バイトのデータを連続して転送するとき、2回目以降のステート1では現在そのチャネルがチャネルイネーブルされてから何バイトの転送が終了したか、を示す値をアドレスバスに出力します。

次のシステムクロックの立ち下がり、KP27はステート2に移行します。このステート2では内部レジスタC-PARの内容がアドレスとして出力され、またそのアドレスへのアクセス信号(MRD\_、IORD\_、IOWR\_)がアサートされます。また、PALAT信号が“L”であった場合、“H”にします。転送対象のI/OをDACK\_信号で指定する場合もC-PARの内容がアドレスバス上に出力されますので注意して下さい。この場合C-PARには、アドレスバス上に出力されても影響のない未使用のI/Oアドレスを設定して下さい。

次のシステムクロックの立ち下がり、KP27はステート3に移行します。このステート3ではKP27はまずPALAT信号を“L”とします。この信号によって、アドレスバス上に出力されているC-PARの内容等をラッチ回路にラッチします。その後KP27は内部レジスタC-SARの内容をアドレスバス上に出力し、またそのアドレスへのアクセス信号(MRD\_、MWR\_)をアサートします。

このステート3までの転送でTC(Terminal Count)に達

していた場合、次はステート4となります。TCには未達で転送続行の場合、KP27はステート1に戻ります。TCには未達で転送終了の場合（DREQネゲート、またはシングル転送のとき）、KP27はステート5になります。

ステート4ではKP27はステート1とほぼ同様の動作をしますが、そのほかに当該チャネルのSR0のTCフラグをセットし、TC信号を“H”にします。このときアドレスバス上に出力される値は、B-BCRに初期値として設定した値です。

Auto Initialize(後述)が設定されていない場合、あるいは転送終了の場合（DREQネゲート、またはシングル転送のとき）、ステート4は次のシステムクロックの立ち下がり、ステート5となります。Auto Initializeが設定されている場合でかつ転送続行の場合、ステート4は次のシステムクロックの立ち下がり、ステート1に戻ります。

ステート5ではKP27はバス権要求を取り下げ、バスを解放するとともにDACK\_信号も“H”とします。また現在そのチャネルがチャネルイネーブルから数えて何バイトの転送が終了したか、を示すデータをアドレスバス上に出力します。ステート5の後KP27はCPUがバス権を得たのを受けてステート0に戻ります。

以上がKP27の動作の概略となります。

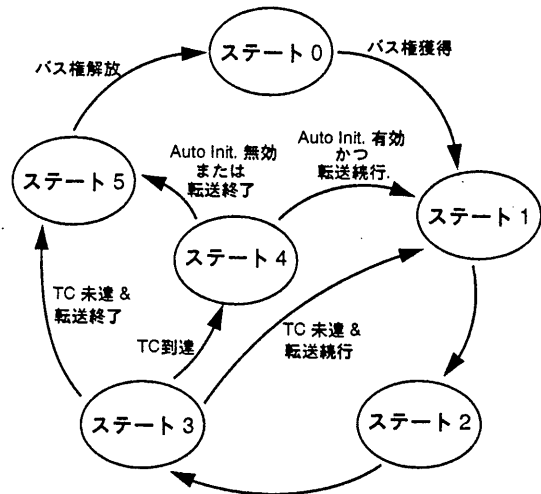


図 7-3 KP27のステート遷移図

## TC(Terminal Count) について

Terminal Countとは、あるデータ転送実行後にC-BCRレジスタの値がカウントダウンされて‘0000H’になることを指します。あるチャンネルがTCに到達した場合、KP27はそのチャンネルのTC信号を“H”にし、またSR0レジスタのビット0(TCフラグ)をセットします。

Auto Initialize有効/無効に関わらず、TCフラグはそのチャンネルのSR0をリードすることによりクリアされます。したがって、Auto Initializeが行なわれたあとでもSR0リードがあるまでTCフラグの状態は保持されます。またCR2、NCC以外の内部レジスタへのライトによっても、そのチャンネルのTCフラグはクリアされます。

## Auto Initializeについて

Auto Initializeは、TC状態が発生したときにそのチャンネルのC-PAR、C-SAR、C-BCRに各ベースレジスタの初期値を設定し直し、チャンネルイネーブル状態を自動的に継続する機能です。

例えばデマンド転送モードに設定されたとき、Auto Initializeを有効に設定していない場合は、TCに到達するとDREQが有効極性であってもチャンネルはディセーブルされますのでそこで転送が終了しますが、Auto Initializeを有効に設定した場合、TC到達後C-PAR、C-SAR、C-BCRには初期値が再設定され、チャンネルはイネーブル状態が継続しますのでDREQが有効極性であれば引き続き転送が実行されます。

Auto Initializeの設定は、当該チャンネルのCR0のビット0を‘0’(Auto Initialize無効)あるいは‘1’(Auto Initialize有効)にすることで行ないます。

## KP27のDREQの有効極性

KP27は各チャンネル毎にDMA要求信号(DREQ)の有効極性を設定可能です。

当該チャンネルのCR0のビット2に‘0’を設定することで有効極性を“立ち下がリエッジ”または“L”に、また‘1’を設定することで有効極性を“立ち上がりエッジ”または“H”に設定できます。

シングル転送モードを設定した場合DREQの上記

エッジがDMA要求に、またデマンド転送モードを設定した場合DREQの上記レベルがDMA要求になります。

## KP27の転送モード

KP27は以下の2つの転送モードを各チャンネル毎に設定可能です。

転送モード設定は、転送タイプ(次項参照)としてメモリ→I/O転送またはI/O→メモリ転送を設定したときに有効で、転送タイプをメモリ→メモリ転送に設定した場合、シングル、デマンドの両転送モードは選択不可となります。この場合、DMA要求はソフトウェア・リクエスト(7.5節のCR1の項参照)のみとなり、またDMA転送はTCに達するまで連続して行なわれます。

### ・シングル転送モード

1回のDMA要求につき1バイトだけDMA転送を実行する転送モードです。この転送モードは当該チャンネルのCR0のビット1を‘0’にすることで設定されます。

本モードではDMA要求(DREQ)はエッジ認識となります。例えばもしチャンネル0がイネーブルされていれば、チャンネル1の転送実行中でもDREQがあったことは記憶されていますので、チャンネル1の転送終了後にチャンネル0のDMAが実行されます。

シングル転送モードにおいては、DREQの有効エッジが発生したステートによって動作が若干異なってきます。

まずステート0の期間でかつそれまでにDREQの有効エッジがなかった場合、DREQ(図7-4のDREQエッジ①)は受け付けられ、DMA動作が開始されます。

ステート0の期間でかつ既にDREQの有効エッジがあった場合(図7-4のDREQエッジ②)、またはステート1の期間にDREQの有効エッジがあった場合(図7-4のDREQエッジ③)、またはステート2の期間にDREQの有効エッジがあった場合(図7-4のDREQエッジ④)、そのDREQは受け付けられません(記憶されません)。

ステート3でDREQの有効エッジがあった場合(図7-4のDREQエッジ⑤)、シングル転送モードながら連続して次の転送動作を行ないます。

ステート5でDREQの有効エッジがあった場合(図7-4のDREQエッジ⑥)、一旦KP27はバスを解放しますが

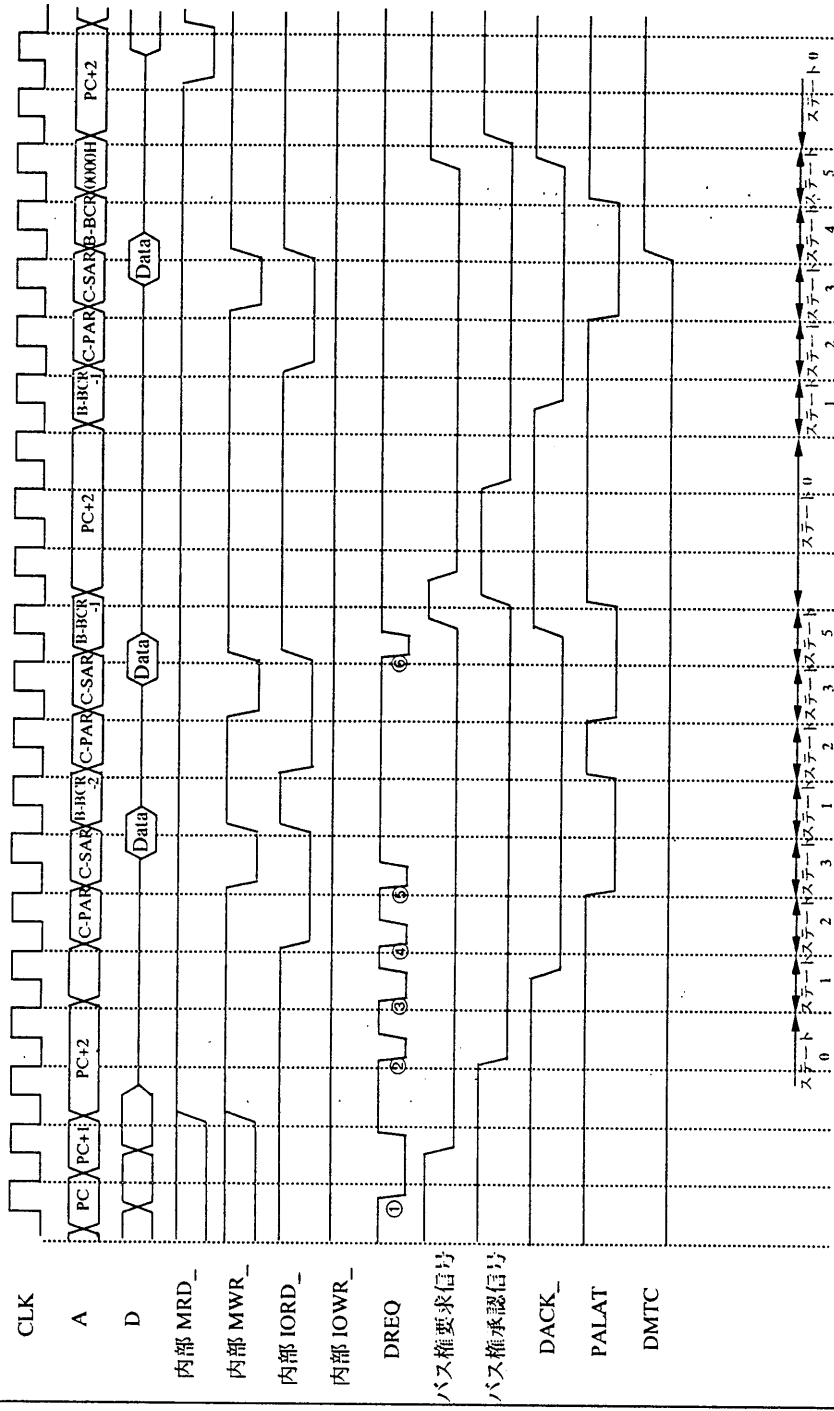


図 7-4 I/O → メモリ転送 (シングル転送モード、DREQ 極性 “↓エッジ”)

その1システムクロック後にすぐバス権を要求するため、CPUが命令の実行を行なわないうちにバス権はKP27に移り、KP27は再びDMA動作を開始します(図7-4参照)。

### ・デマンド転送モード

DMA要求(DREQ)が有効極性である期間、連続してDMA転送を実行する転送モードです。DREQが有効である期間中にTCに到達した場合は、Auto Initializeの設定の有無により動作が異なります(Auto Initializeの項参照)。この転送モードは当該チャンネルのCR0のビット1を'1'にすることで設定されます。

本モードではDMA要求(DREQ)はレベル認識となります。つまり有効極性である期間だけDREQは有効であるので、例えばチャンネル1の転送実行中にDREQ0が有効極性になっても、チャンネル1の転送終了時までその有効極性状態が持続していなければチャンネル0のDMA転送は実行されません。

有効極性のDREQによりDMA転送を実行するかしないかの判断はステート1あるいはステート5へ移行するシステムクロックの立ち上がり時に行ないます。すなわち、その時点でDREQが有効極性であれば(次の)転送を実行し、有効極性でなければ(次の)転送は実行されません(図7-5参照)。

### KP27の転送タイプ

KP27は次の3種類の転送タイプを各チャンネル毎に設定可能です。

#### ・メモリ→I/O転送

メモリ上のデータをI/Oへ転送する転送タイプです。

この転送タイプは当該チャンネルのCR0のD<5:3>を'010'(メモリアドレスインクリメント)または'011'(メモリアドレスデクリメント)にすることで設定されます。I/Oアドレスは変化しません。

この転送タイプでは、B-PARにはデータ転送先のI/Oアドレスを、B-SARにはデータ転送元のメモリアドレスを設定します。B-BCRには転送バイト数を設定します。各ベースレジスタへの設定と同時にその値は各カレントレジスタへも書き込まれます。転送対象の

I/O指定をDACK\_信号で行なう場合はB-PARに設定する値には注意が必要です。

この転送タイプはDREQ端子からのDMA要求によって起動されます。転送モードはシングル転送モード、デマンド転送モードが選択できます。

メモリ→I/O転送では、転送を開始するとまずDACK\_信号をアサートします。そしてC-PARの値で示されるI/Oアドレスを出力しつつ、PALAT\_信号をデータスルー極性("H")にします。またIOWR\_信号をアサートしI/Oライトアクセスを開始します。次にPALAT\_信号をデータラッチ極性("L")にし、これによってI/Oアドレス等をレジスタ(ラッチ等)にラッチしてからC-SARの値で示されるメモリアドレスを出力し、MRD\_信号もアサートします。こうしてメモリリードされたデータを直接I/Oに書き込み、1バイト転送します(図7-5参照)。

#### ・I/O→メモリ転送

I/Oのデータをメモリ上へ転送する転送タイプです。

この転送タイプは当該チャンネルのCR0のD<5:3>を'100'(メモリアドレスインクリメント)または'101'(メモリアドレスデクリメント)にすることで設定されます。I/Oアドレスは変化しません。

このとき、B-PARにはデータ転送元のI/Oアドレスを、B-SARにはデータ転送先のメモリアドレスを設定します。B-BCRには転送バイト数を設定します。各ベースレジスタへの設定と同時にその値は各カレントレジスタへも書き込まれます。転送対象のI/O指定をDACK\_信号で行なう場合は、B-PARに設定する値が他のI/Oアドレスと重ならないよう注意が必要です

この転送タイプもDREQ端子からのDMA要求によって起動されます。転送モードはシングル転送モード、デマンド転送モードが選択できます。

I/O→メモリ転送では、転送を開始するとまずDACK\_信号をアサートします。そしてC-PARの値で示されるI/Oアドレスを出力しつつPALAT\_信号をデータスルー極性("H")にします。またIORD\_信号をアサートしI/Oリードアクセスを開始します。次にPALAT\_信号をデータラッチ極性("L")にしI/Oアドレス等をレジスタ(ラッチ等)にラッチしてからC-SARの値で示されるメモリアドレスを出力し、MWR\_信号

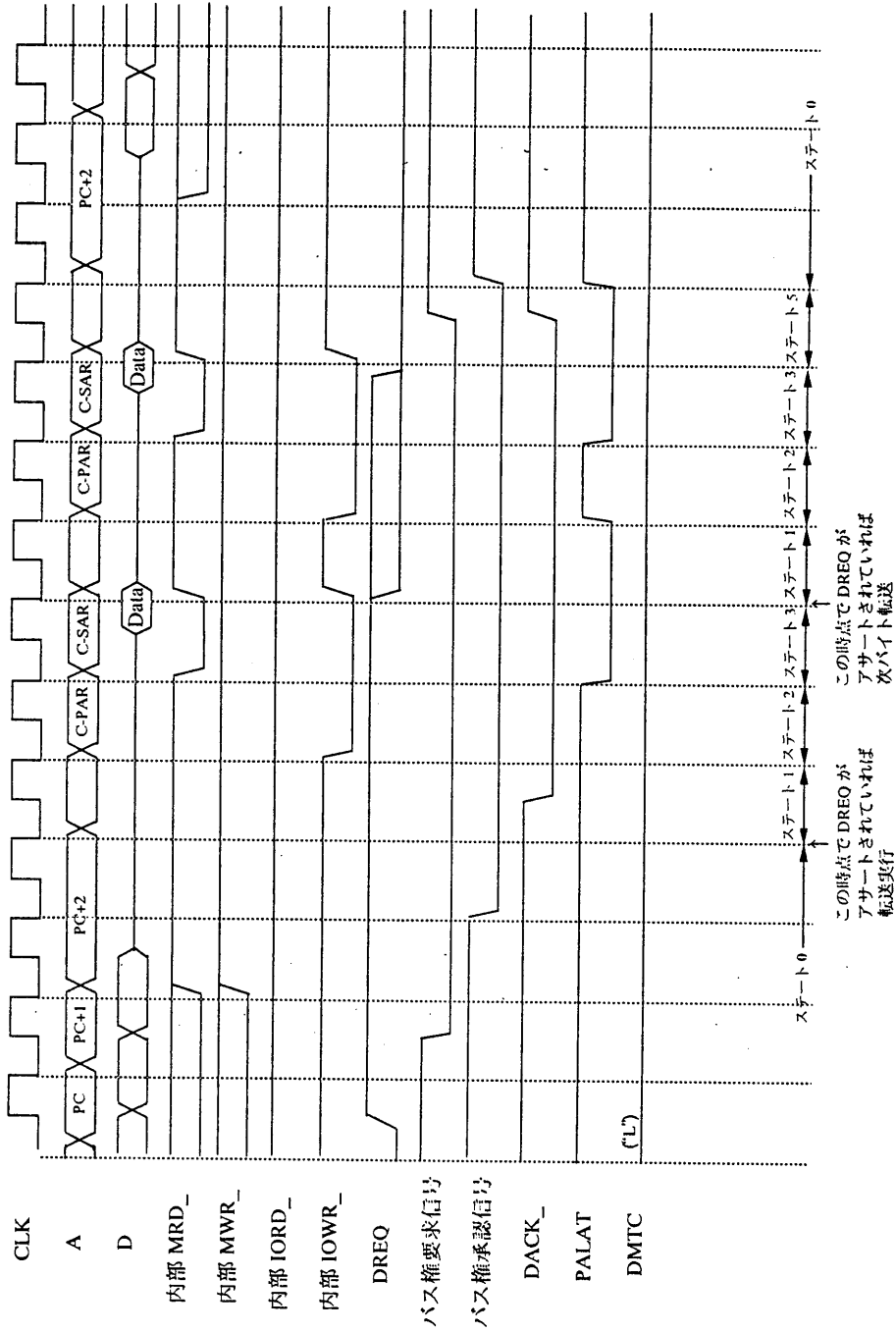


図 7-5 メモリ→I/O 転送 (デマンド転送モード、DREQ 極性 "H")

もアサートします。こうしてI/Oリードされたデータを直接メモリに書き込み、1バイト転送します（図7-4参照。この図ではDREQのエッジのタイミングとDMA転送実行の様子も合わせて説明しています。シングル転送モードの項参照）。

#### ・メモリ→メモリ転送

メモリ上のデータをメモリ上の他のアドレスへ転送する転送タイプです。

この転送タイプは当該チャンネルのCR0のD<5:3>を'000'（メモリアドレスインクリメント）または'001'（メモリアドレスデクリメント）にすることで設定されます。このときCR0のD<2:1>、すなわちDREQ極性、転送モードの設定は無視されます。Auto Initializeも無効になりますが、CR0のD0は'0'にして下さい。

この転送タイプでは、B-PARにはデータ転送元のメモリアドレスを、B-SARにはデータ転送先のメモリアドレスを設定します。B-BCRには転送バイト数を設定します。各ベースレジスタへの設定と同時にその値は各カレントレジスタへも書き込まれます。

メモリ→メモリ転送はソフトウェア・リクエストによってのみ起動されます。ソフトウェア・リクエストはCR1へのコマンドライトで行なわれます。起動されたら、NMIによって中断される場合を除いて、TCに達するまでCPUにバス権が戻ることはありません。アドレスはC-PAR、C-SARともにインクリメントあるいはC-PAR、C-SARともにデクリメントが選択できます。

この転送タイプでは、転送を開始するとまずC-PARの値で示されるアドレスのメモリリードを行ない、読み出したデータをKP27内部に保持してから、次にC-SARの値で示されるアドレスにメモリライトを行なって1バイト転送します（図7-6参照。図7-6では2バイト目の転送時のバスサイクルがバスサイクル延長信号によって延長される様子も合わせて示しています）。

#### チャンネルイネーブルとディセーブル

各DMAチャンネルのイネーブル/ディセーブルはCR1で行ないます。CR1のビット5を'1'にするとそのチャンネルはイネーブル、'0'にするとディセーブルされま

す。またメモリ→メモリ転送設定チャンネルではビット5を'1'にするとソフトウェア・リクエストとなります。

イネーブル中のチャンネルの内部レジスタに書き込みを行なうとそのチャンネルはディセーブルとなりますので、DMA転送を実行する場合は再度イネーブルする必要があります。内部レジスタの読み出しを行なってもイネーブル状態に影響はありません。また、ディセーブル中のチャンネルに対するシングル転送要求、ソフトウェアリクエスト等は記憶されません。

#### DMA要求のプライオリティと外部DMAデバイス

KL5C80A16は2つのDMAチャンネルを持ち、2つの異なる要求源からのDMA要求を調停、処理できます。2つのDMAチャンネルの優先順位は固定で、チャンネル1>チャンネル0となっています。

したがって、例えばチャンネル0がメモリ→メモリ転送実行中にデマンド転送モードのDREQ1が有効極性になったら、チャンネル0の転送を中断してDREQ1の処理を開始します。DREQ1が有効極性でなくなったらチャンネル0の転送を再開し、これがTCに到達したらバス権をCPUに返す、というように動作します。

またKP27は、DMA機能をもったデバイスを外部DMAデバイスとして接続し、そのデバイスのバス権要求とKP27に接続されているI/OのDMA要求とを調停する機能を備えています。これは、外部DMAデバイスのバス権要求信号をEXBREQ\_端子に、またEXBACK\_信号を外部DMAデバイスのバス権要求認識信号端子に接続することで実現されます。なお、EXBREQ\_、EXBACK\_はどちらも有効極性は"L"です。また、EXBREQ\_はレベル認識となります。

この時、KP27はDREQ0、DREQ1とEXBREQ\_の3つのDMA要求を調停可能です。ただし外部DMAデバイスの優先度（プライオリティ）は、DREQ0、DREQ1のいずれよりも高い(top)、あるいはいずれよりも低い(bottom)、のどちらかになります。この設定は、CR2のD<4:3>により行ないます。ここを'11'にすることにより外部DMAデバイスがプライオリティtopで、また'10'にすることによりプライオリティbottomで設定されます。外部DMAデバイスを接続しない場合はCR2のD4を'0'にします。この時はD3の値は無視されます。

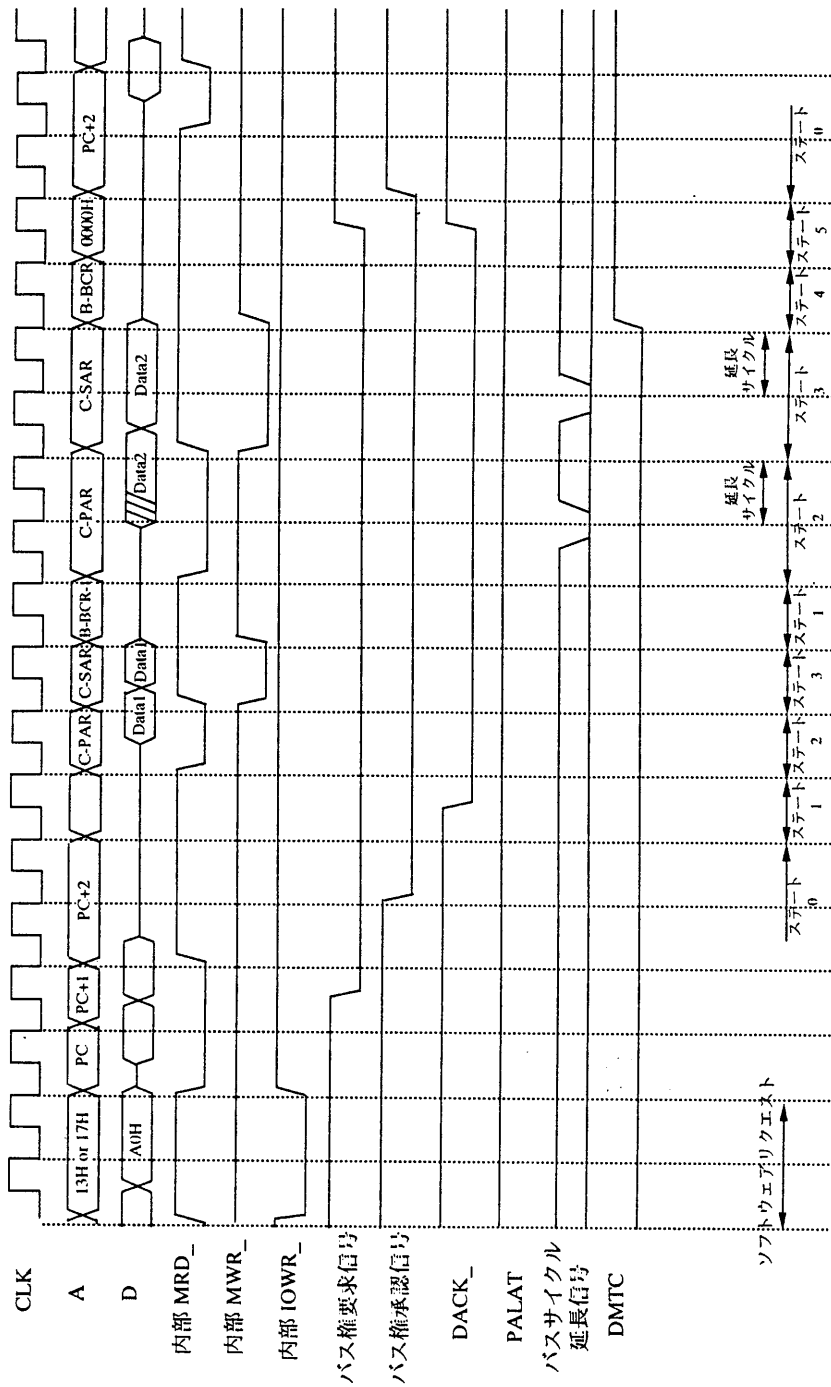


図 7-6 メモリ→メモリ転送

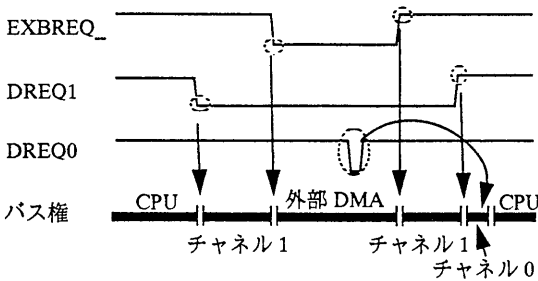
外部DMAデバイスのプライオリティ設定がtopの場合、KP27は

- ・DREQ0またはDREQ1の処理中に外部DMA要求が発生したら、ある1バイトの転送が終了した後のステート1の途中で転送を一時中断して、外部DMAの処理を行ないます。外部DMA処理が終了したのち、中断したDREQを再開します。
- ・外部DMA処理中にDREQ0またはDREQ1があり、外部DMA処理が終了したときにそれが有効のままである（デマンド転送モード）あるいは記憶されている（シングル転送モード）ときは、外部DMA処理が終了したのちにそのDREQの処理にとりかかります。

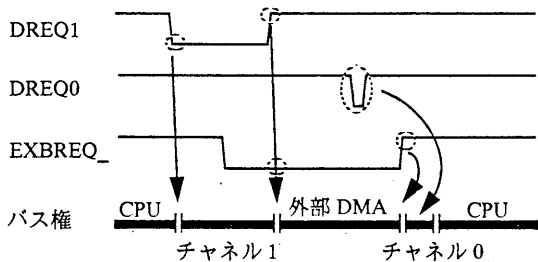
また、外部DMAデバイスのプライオリティがbottomの場合、KP27は

- ・DREQ0およびDREQ1の処理が全て終了した時点で外部DMA要求が存在していたら、外部DMAの処理を行ないます。
- ・外部DMA処理中にDREQ0またはDREQ1が発生したら、外部DMA処理が終了したのちに、有効なDREQについて処理にとりかかります。

いずれにしても、外部DMA処理を開始した場合はそれが終了するまでバス権はKP27にもCPUにも復帰しません。したがって、外部DMAによる長時間のバス専有はCPUやKP27の正常な動作を妨げるおそれがあります。



外部DMAデバイス プライオリティ top の場合



外部DMAデバイス プライオリティ bottom の場合

DREQ1 : デマンド転送、“L”で有効  
DREQ0 : シングル転送、“↓”で有効とする。

図7-7 外部DMAデバイスのプライオリティとバス権の移動の様子

### NMIによるDMAの中断(NMI Abort)

KP27はNMI(Non Maskable Interrupt)によってDMA転送を中断できる機能を備えています。この機能はCR2のビット2で設定します。ここを‘1’にすることで本機能が有効、‘0’にすることで本機能が無効となります。

この機能は、NMI信号をシステムの非常事態対処用に用いる場合、NMI要求が発生したにもかかわらずDMAコントローラがバス権を持っているため迅速なNMI処理が出来ない、というような事態を避けるために用意されています。

KP27がバス権を持ちDMA転送を行なっている最中にNMI信号の立ち下がりが発生すると、KP27は内部のNMIフラグをセットして転送の切れ目（ある1バイトの転送が終了したところ）でバス権をCPUに戻します。この時各DMAチャンネルの内部レジスタはその値を保持したままです。

NMI処理のために、DMA転送を中断してバス権をCPUに戻すか否かの判断は、ステート3と次のステートの境目のシステムクロックの立ち下がりで行なわれます。すなわちNMI\_信号の立ち下がりエッジがここより前にある場合は現在実行している転送を行なった後DMA転送を中断しますが、ここより後にNMI信号の立ち下がりエッジがある場合は次の1バイト転送を実行してからDMA転送を中断し、バス権をCPUに戻します。(図7-8にDMA転送→NMI処理開始のタイミングチャート例を示します。)

バス権を受け取った CPUはNMI処理ルーチンを実行しますが、NMI Abort機能を有効にした場合、DMA転送中以外のNMIでもKP27内部のNMIフラグはセットされますので、NMI処理ルーチンの最後では必ず7.5節で述べたNCCを書き込むようにして下さい。

KP27はNCCのI/Oライトを行なうクロックの立ち下がりから1バスサイクル+1.5CLK後に自動的にバス権要求信号をアサートしてバス権を再要求し、バス権を受けとったら先ほどNMI\_信号により中断したDMA転送を再開します。

NMI Abort機能を用いる場合、次のようなNMIルーチンをプログラムしてお使いになることをおすすめします。また、このプログラム例によるDMA再開時のタイミング図を図7-9に示します。

```

org      0066H
PUSH    AF          {F5H}
.
.
.
LD      A,  F0H     {3EH F0H}
OUT     (13H), A   {D3H 13H}
POP     AF          {F1H}
RET     {EDH 45H}
        {}内は命令コードです。
    
```

この場合、CPUがPOP AF命令実行中(スタックポップ中)にKP27がバス権を再要求しますので、CPUはPOP AF命令実行後、RETN命令の1バイト目('EDH')をフェッチしたまま実行せずにバス権を解放し、これによりDMA転送が再開されます。したがって、この再開した転送を終了してバス権をCPUに返した場合、CPUはRETN命令から実行します。また再開した転送中再びNMI Abortがかかった場合、転送は中断されCPUにバス権が返りますが、ここでCPUは先ほど1バイト目をフェッチしておいたRETN命令の、2バイト目('45H')をフェッチし、RETN命令を実行してからNMIルーチンに入るので、NMIのネストは発生しません。(図7-10参照。)

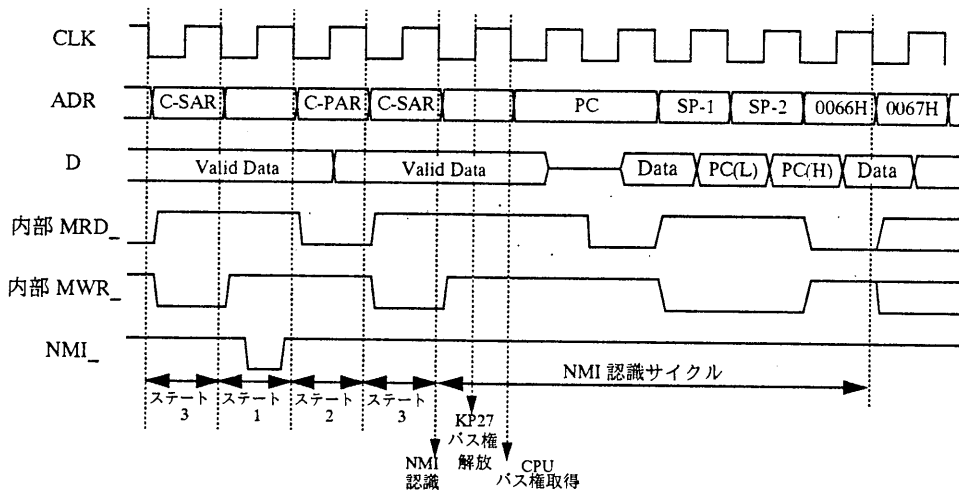


図 7-8 DMA 転送中断→NMI ルーチン開始

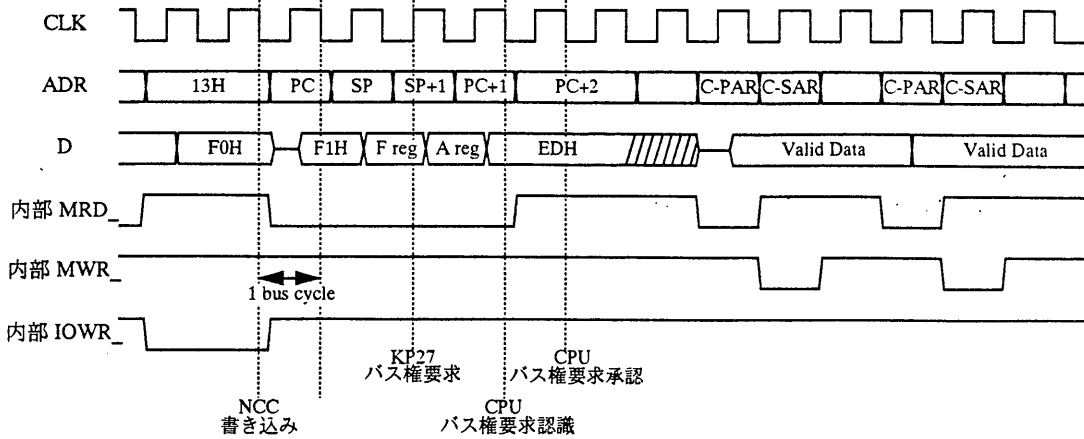


図 7-9 上記プログラム例による NMI 処理ルーチン終了→ DMA 転送再開

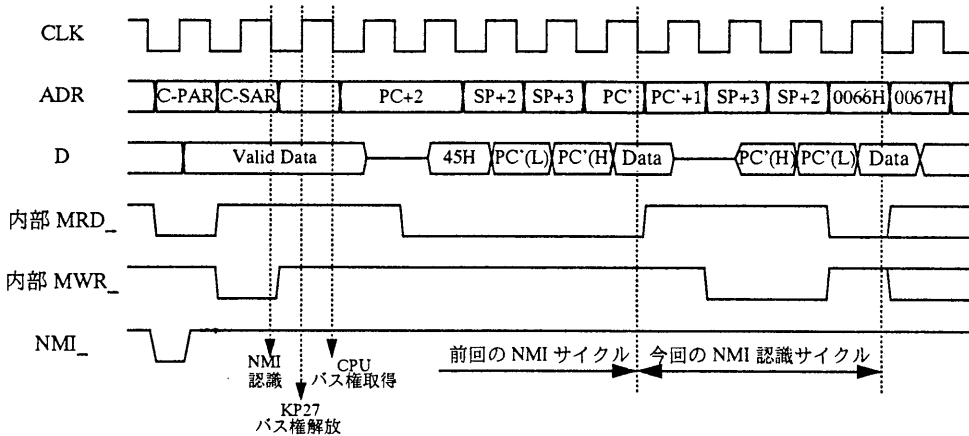


図 7-10 前記プログラム例による DMA 転送再開→ NMI ルーチン開始 (図 7-9 の続き)

### NMIによる中断と外部DMA

CPUがバスマスタであるときに発生したNMI処理中に外部DMAデバイスのバス権要求があった場合、KP27はEXBREQ\_信号が有効極性になったのを認識し、CPUに対してバス権を要求します。

一方、KP27がバスマスタであるとき(DMA転送実行中)に発生したNMIの処理中に外部DMAデバイスのバス権要求があった場合は、外部DMAデバイスのプライオリティがtopの場合のみ、NMIルーチン実行中でも外部DMAデバイスへのバス権の移動が起りま

す。

また、さきにも述べたとおり外部DMAデバイスがひとたびバス権を得ると、そのデバイスがバス権を解放するまでKP27にもCPUにもバス権は復帰しません。したがって、このときNMI要求が発生しても外部DMAデバイスがバス権を解放するまでCPUによるNMI処理は開始されません。

これらのことから、長時間バスを専有する可能性のある外部DMAデバイスを御使用になる場合は注意が必要です。

### バスサイクルの延長

KP27は遅いメモリやI/Oを使用する場合のために、バスサイクル延長信号を入力することによってバスサイクルを延長することが出来ます。バスサイクル延長信号は、KL5C80A16の端子6から入力されるERDY信号が、外部バス・インターフェース・ユニットによってシステムクロックに同期化されて接続されています。

システムクロックの立ち下がりでバスサイクル延長信号が“L”であった場合、そのシステムクロックの立ち下がりから開始されるバスサイクルが延長サイクルとなります。ただし延長可能なのは

- ・メモリ→メモリ転送ではステート2およびステート3の期間
- ・メモリ→I/O転送、I/O→メモリ転送ではステート3の期間

のみです（図7-6参照）。

バスサイクルの延長による外部バス信号の出力の様子については4章を参照して下さい。

### リセット

RESET\_端子を“L”レベルにすると、KP27は次のように動作します。

- (1)全ての内部レジスタの全ビットを‘0’にクリアします。すなわち、PAR, SAR, BCR は全て‘00H’となり、また各設定は、

転送タイプ : メモリ→メモリ転送  
 アドレス Increment  
 DREQ極性 : 有効極性“↓エッジ”または“L”  
 転送モード : シングル転送モード  
 (上記2つの設定はメモリ→メモリ転送により無効)  
 Auto Initialize : 無効  
 各チャンネル : ディセーブル  
 外部DMA : 未使用  
 外部DMA優先度 : bottom  
 (この設定は外部 DMA 未使用であるので無効)  
 NMIによる中断 : 無効

となります。したがって、リセット解除後DMAを動作させるには各レジスタに設定を行なってください。

- (2)実行中の DMA 転送は全て中止されます。

### 7.7 注意事項

DMA 動作を実行しているチャンネルの DACK\_ 信号は、そのチャンネルの転送タイプ設定にかかわらず有効極性 (“L”) になります。また、転送対象の I/O 指定に DACK\_ 信号を用いる場合でも I/O アドレスや PALAT 信号は出力されますので、注意して下さい。このとき B-PAR には出力されても影響のない I/O アドレスを設定して下さい。

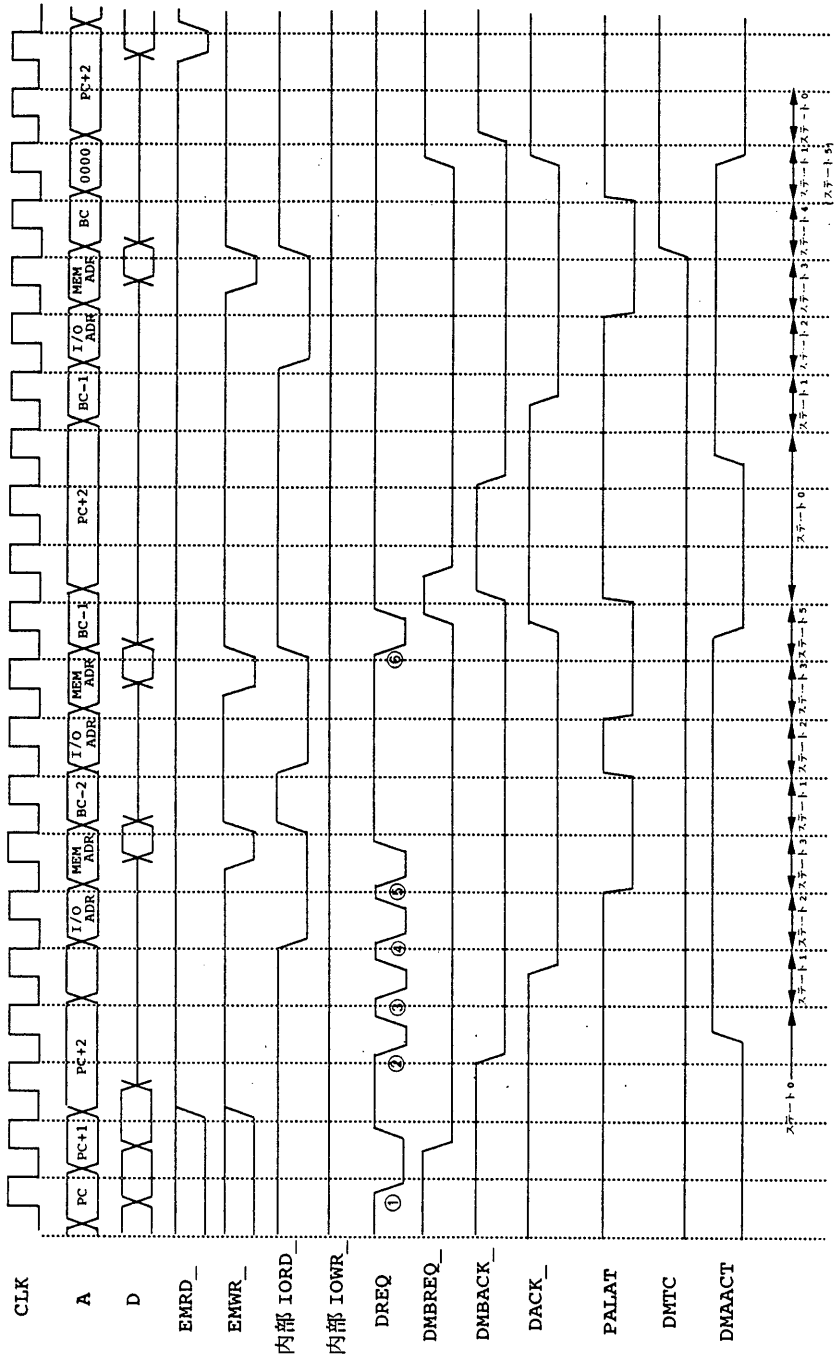


図 7-2 I/O → メモリ転送 (シングル転送モード、DREQ 極性 "L")

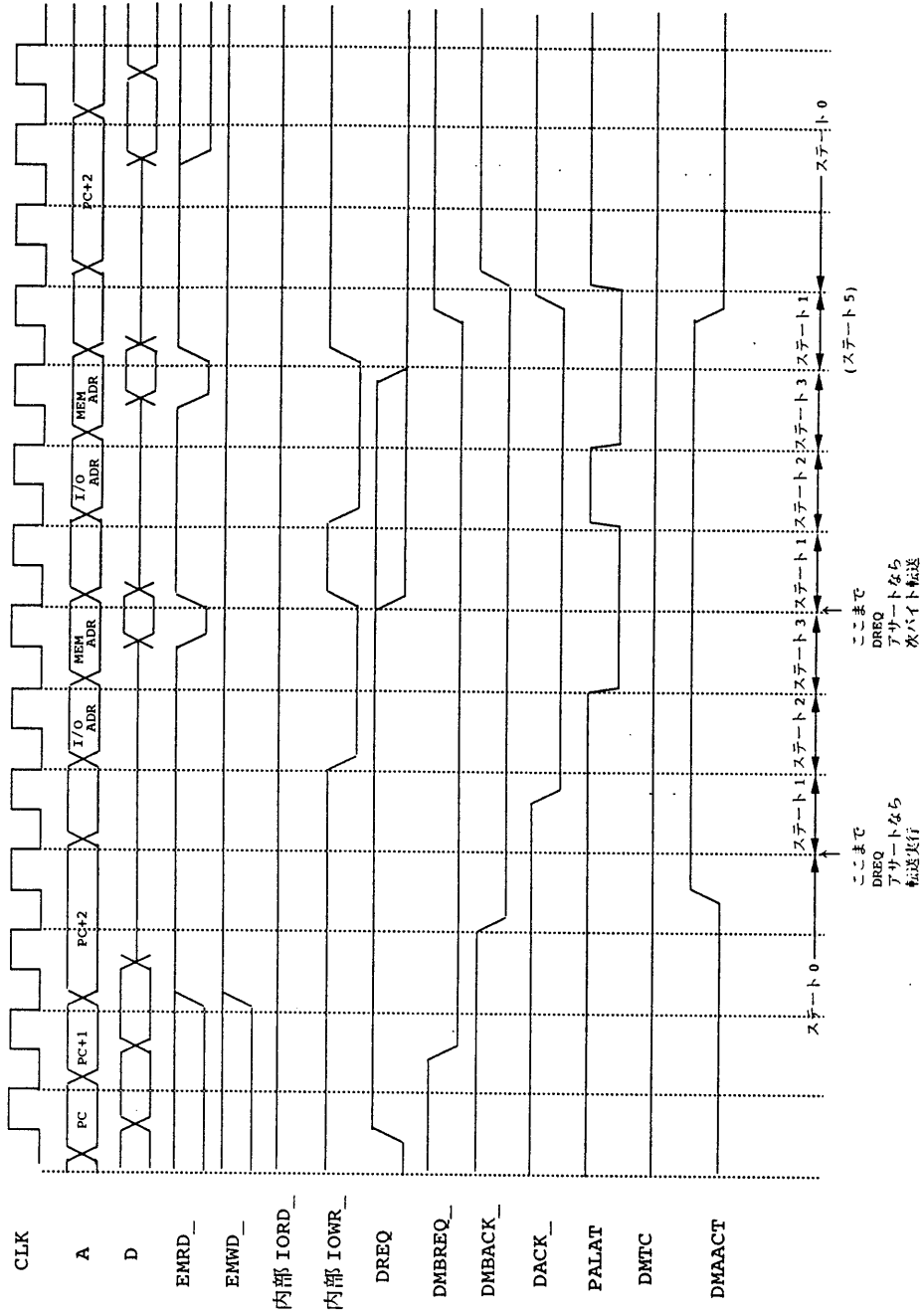


図 7-3 メモリ → I/O 転送 (デマンド転送モード、DREQ 極性 “H”)

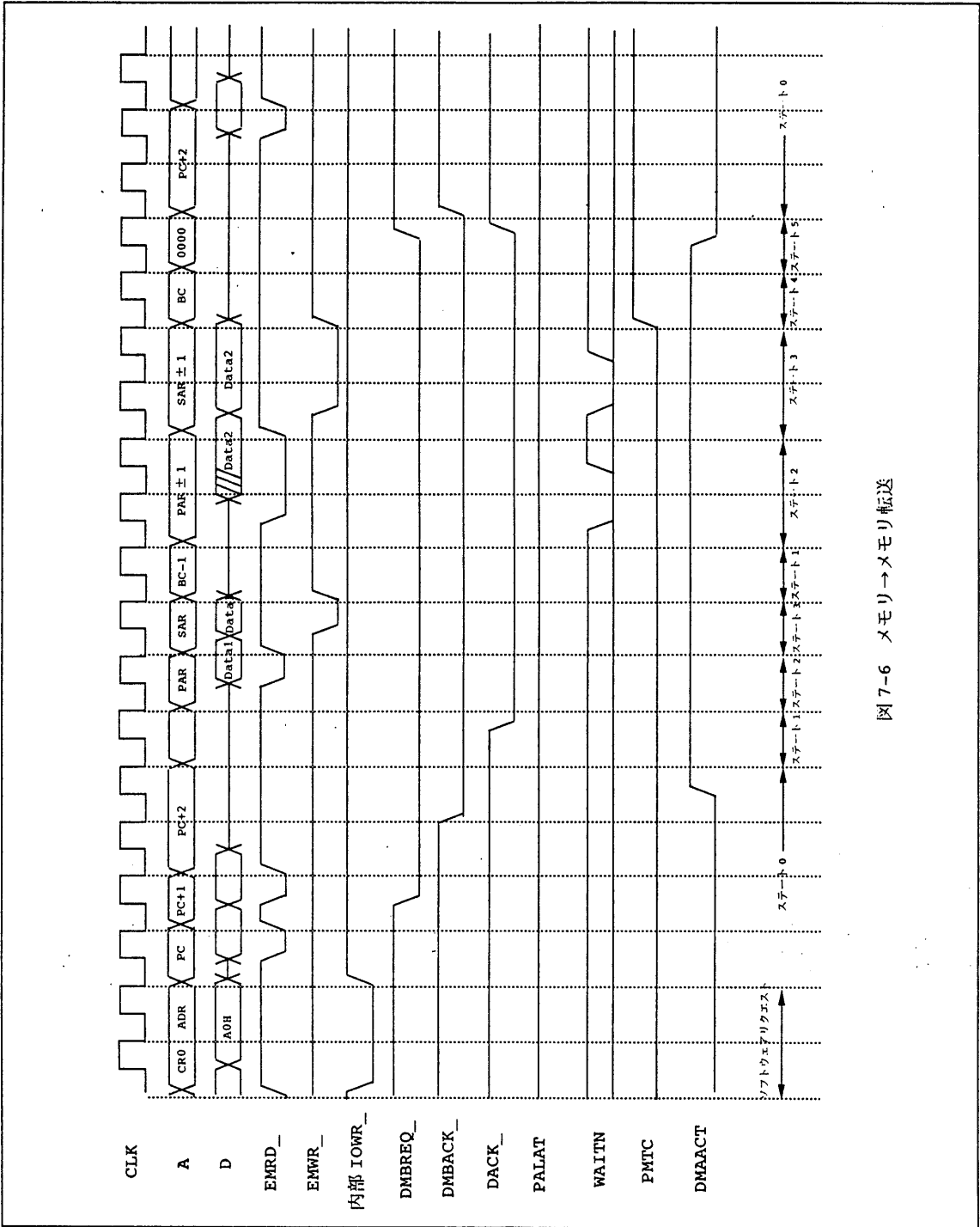


図 7-6 メモリ→メモリ転送

### 8. UART

#### 8.1 概要

KL5C80A16は非同期シリアルポートとしてKP61マクロセルを搭載しています。本マクロは非同期シリアルポートを2チャンネル内蔵しており、CPUからの動作設定にしたがって、シリアルデータの送信・受信を行ないます。また、各チャンネルとも送信時、受信時独立に割り込み発生が可能な他、SCR2の設定によりチャンネル0の送信側または受信側を内蔵DMAのチャンネル1に内部で接続することができます。

#### 特徴

・送信・受信で独立したバッファレジスタを備えており、全二重通信を行ないます。

- ・通信チャンネルは2つ用意されています。
- ・キャラクタ長は7,8,9ビットを選択できます。
- ・ストップビットとして1または2が選択できます。
- ・パリティビットの付加(偶数/奇数)/無付加が選択できます。ただしキャラクタ長が9ビットの時はパリティ無付加となります。
- ・送受信クロック源としてチャンネル0,1に共通のボーレートジェネレータを内蔵しています。ボーレートはチャンネルごとに設定可能です。
- ・送受信クロックを内部/外部選択可能です。送信クロックと受信クロックは共通です。
- ・サンプリングクロックはシリアルデータのビットレートの×16クロックです。
- ・スリーピングモードの使用により、複数のマイクロコントローラと通信が可能です。

#### 8.2 ブロック図

KP61の全体ブロック図を以下に示します。

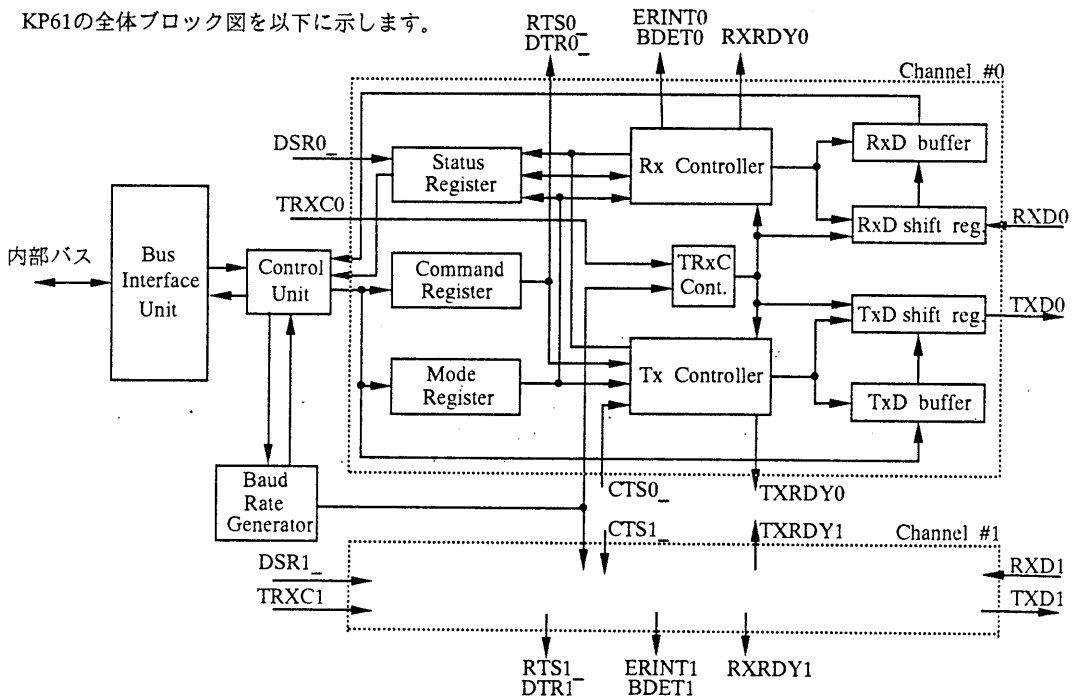


図 8-1 KP61 ブロック図

図8-1において、チャンネル1はチャンネル0と同じ構成になっています。

チャンネル0の信号のうち、TXRDY0は割り込みコントローラの IR[9] に、RXRDY0 は IR[10] に、ERINT0+BDET0はIR[11]にそれぞれ内部で接続されています。また、DSR0\_ は内部で“L”に固定され、DTR0\_ はKL5C80A16外部へは出力されません。これ以外のチャンネル0の信号にはKL5C80A16の外部端子が専用端子として割り当てられています。

チャンネル1の信号のうち、TXRDY1 は IR[3] に、

RXRDY1 は IR[4] に内部で接続されています。また ERINT1+BDET1は、SCR1の設定によってIR[5]に内部で接続できるようになっています。ERINT1、BDET1以外のチャンネル1の信号はすべてパラレルポートと多重化されています。これらの信号の詳しい使い方は12章を参照して下さい。

SCR2の設定によって、チャンネル0のTXRDY0、あるいはRXRDY0のいずれかをDMAコントローラのチャンネル1のDMA要求として使用することができます。これについても12章を参照して下さい。

### 8.3 端子機能説明

端子名の0,1はそれぞれチャンネル0、チャンネル1を表します。

端子名	I/O	機能説明
TRXC0 TRXC1	I	送受信クロック入力端子 データ送受信の際の×16クロックを入力する端子です。送受信クロックとして外部入力を選択する場合に使用します。
RXD0 RXD1	I	受信データ入力端子 外部からのシリアルデータを入力する端子です。受信データはビットの中央付近で3回サンプルされ、多数決論理により受信シフトレジスタ内に取り込まれます。
RXRDY0 RXRDY1	O	受信レディ信号出力端子 1キャラクタのデータ受信が終了して受信バッファからの読み出しが可能になると、“H”を出力します。また、データの読み出しによって“L”にリセットされます。 RXRDY0は、KL5C80A16内部でIR[10]に接続されています。またDMAコントローラのDREQ1として使用することもできます。(12章参照) RXRDY1は、KL5C80A16内部でIR[4]に接続されています。
TXD0 TXD1	O	送信データ出力端子 送信データをシリアルに出力する端子です。CTS_="L"かつ送信イネーブル(コマンドレジスタのビット0='1')でデータ送出可能となり、送受信クロック16サイクル毎に、その送受信クロックの立ち下がりに同期してLSBから順に送出されます。 送信中にCTS_="H"または送信ディセーブル(コマンドレジスタのビット0='0')となった場合、送信バッファ内のデータ送出終了後にマーキング状態(送信データ='1')となって送信動作を中止します。 また、送信ブレイクコマンド書き込み(コマンドレジスタのビット3を'1'にする)によって、TXDからは“L”が出力されます。このブレイク状態は送信ブレイク解除(コマンドレジスタのビット3を'0'にする)まで継続します。

端子名	I/O	機能説明
TXRDY0 TXRDY1	○	<p>送信レディ信号出力端子</p> <p>KP61のデータ送信系は二重バッファ構造のため、データの送信を開始すると次の送信データの書き込みが可能となります。この送信データ書き込み可能な状態でかつ送信可能状態（CTS_="L" かつ送信イネーブル状態）であるとき TXRDY="H" となります。このとき送信データの書き込みが行なわれると TXRDY="L" となり、この状態は、再び送信データ書き込み可能な状態でかつ送信可能状態となるまで続きます。この端子はステータスレジスタのビット0(TXRDYフラグ)とは以下のように異なります。</p> <p>TXRDYビット：送信データの書き込みが可能である時 "H"</p> <p>TXRDY端子：送信データの書き込みが可能で、かつ送信可能状態である時 "H"</p> <p>TXRDY0は、KL5C80A16の内部でIR[9]に接続されています。また内蔵DMAコントローラのDREQ1として使用することもできます。(12章参照。)</p> <p>TXRDY1は、KL5C80A16の内部でIR[3]に接続されています。</p>
CTS0_ CTS1_	I	<p>クリアトゥセンド信号入力端子</p> <p>外部からの送信要求信号を入力する端子です。通常モデムのリクエストトゥセンド信号を接続します。送信シフトレジスタ内にデータがあるとき、送信イネーブル状態（コマンドレジスタのビット0='1'）でかつCTS_="L"のとき、送信データがTXD端子から出力されます。データ送中にCTS_="H"となっても送信の中断はしません。この場合、送信シフトレジスタ内のデータ送終了後にマーキング状態（TXD="H"）を保持します。</p> <p>この端子の値の反転が拡張ステータスレジスタBのビット6に反映されます。</p>
DSR0_ DSR1_	I	<p>データセットレディ信号入力端子</p> <p>通常モデムのステータスをチェックするためにモデムのデータターミナルレディ信号を接続しますが、汎用の入力ポートとしても使用できます。</p> <p>この端子の値の反転がステータスレジスタのビット7(DSR)に反映されます。</p> <p>DSR0_は内部でGNDに固定されています。</p>
RTS0_ RTS1_	○	<p>リクエストトゥセンド信号出力端子</p> <p>通常モデムに対するリクエストトゥセンド信号として用いますが、汎用の出力ポートとしても利用できます。</p> <p>コマンドレジスタのビット5(RTS)に設定した値の反転が出力されます。</p>
DTR0_ DTR1_	○	<p>データターミナルレディ信号出力端子</p> <p>通常モデムに対するデータターミナルレディ信号として用いますが、汎用の出力ポートとしても利用できます。</p> <p>コマンドレジスタのビット1(DTR)に設定した値の反転が出力されます。なお、DTR0_はKL5C80A16の外部へは出力されません。</p>

端子名	I/O	機能説明
ERINT0 ERINT1	○	エラー発生割り込み要求信号出力端子 パリティ・エラー、フレーミング・エラー、オーバーラン・エラーのうち少なくとも一つが発生すると“H”を出力します。この信号を割り込み要求信号として使用することができます。
BDET0 BDET1	○	ブレイク検出信号出力端子 ブレイクの検出を示す信号の出力端子です。 受信データのストップビット='0'を検出した後連続して8ビットの'0'をRXD端子から受信すると、それをブレイク信号と認識して、“H”を出力します。この状態は次にRXD端子から'1'を検出するまで継続されます。

### 8.4 内部レジスタマッピングと設定方法

KP61の内部レジスタのマッピングを以下に示します。

表8-1 I/Oレジスタマッピング

I/O アドレス	ライト時	リード時
28H	RATE設定	RATE設定
29H	川崎製鉄予約	川崎製鉄予約
2AH	チャンネル0送信データ	チャンネル0受信データ/拡張ステータスA
2BH	チャンネル0モード/コマンド	チャンネル0ステータス/拡張ステータスB
2CH	チャンネル1送信データ	チャンネル1受信データ/拡張ステータスA
2DH	チャンネル1モード/コマンド	チャンネル1ステータス/拡張ステータスB
2EH	川崎製鉄予約	川崎製鉄予約
2FH	川崎製鉄予約	川崎製鉄予約

### レジスタ設定／読み出しシーケンス

リセット後、KP61 でデータ送受信を行なう前に、使用するチャンネルに対して以下の手順でモード設定とコマンド書き込みを行なって下さい。またステータスレジスタの読み出し方法も合わせて説明します。

#### モード設定

リセット（ハードリセットまたはコマンド書き込みによるソフトウェアリセット）後、モード/コマンド（'2BH'、'2DH'）に書き込みを行なうと、必ずモードレジスタへの書き込みとなります。ここではキャラクタ

長やパリティの設定等を行ないます。8.5節のフォーマットにしたがってモード設定を行なって下さい。

#### コマンド設定

モード設定後、I/Oアドレス'2BH'、'2DH'への書き込みは全てコマンドレジスタへの書き込みとなります。このコマンド設定によりデータ送受信が可能となります。コマンドレジスタは3種類あります。8.5節のフォーマットにしたがってコマンド設定を行なって下さい。

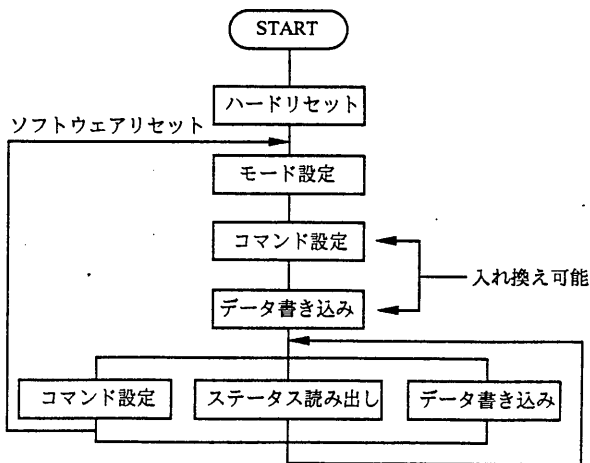


図 8-2 レジスタ設定、読み出しシーケンス

### 読み出しレジスタの切替え

ここでは読み出すレジスタの切替え方法について、チャンネル0を例に説明します。

リセット後、I/O アドレス 2AH, 2BH からの読み出しではそれぞれチャンネル0の受信データ、ステータスが読み出せます。ここで、I/O アドレス 2BH へレジスタ切替えコマンドのビット 0='1' にして書き込みますと、レジスタ切替え F/F がセットされて、I/O アドレス '2AH', '2BH' から読み出せるレジスタがチャンネル0の拡張ステータスレジスタとなります。拡張ステータスレジスタ A, B それぞれのフォーマットは 8.5 節を参照して

下さい。

2つの拡張ステータスレジスタのうち、拡張ステータスレジスタ A の読み出しを行なってもレジスタ切替え F/F はクリアされません。したがって、続いて拡張ステータスレジスタ B の読み出しを行なうことができます。拡張ステータスレジスタ B の読み出しを行なうとレジスタ切替え F/F はクリアされますので、続いて読み出しを行なうと受信データ/ステータスレジスタを読み出すことになります。レジスタ切替え F/F のクリアはレジスタ切替えコマンドのビット 0='0' にして書き込むことでも行なえます。

チャンネル 1 についても同様です。なお、この読み出しレジスタの切替えはチャンネル間で独立に行なわれます。但し、拡張ステータス B の読み出しによるレジスタ切替え F/F のクリアについては、チャンネル 0 とチャンネル 1 両方同時に行なわれますのでご注意ください。

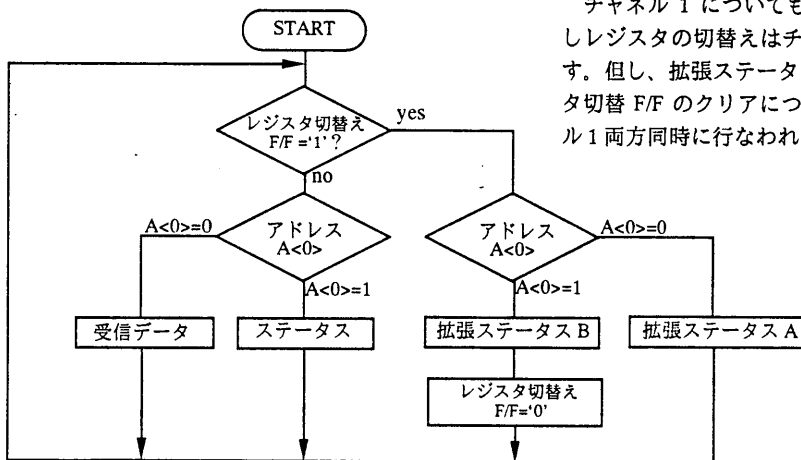
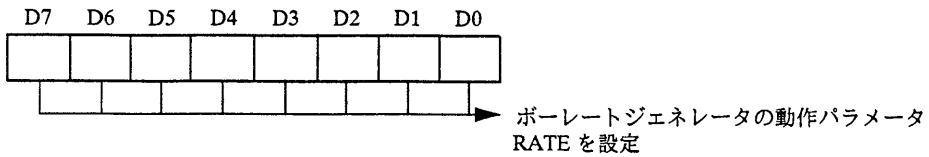


図 8-3 読み出しレジスタ切替えシーケンス

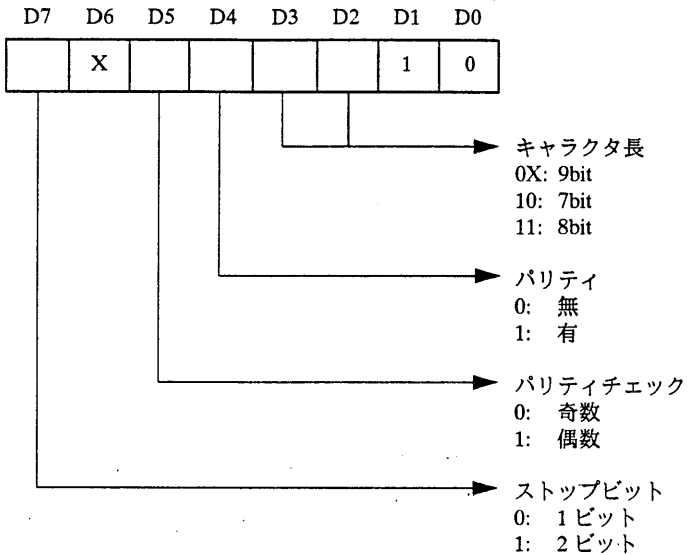
### 8.5 内部レジスタの構成

#### RATE 設定レジスタ

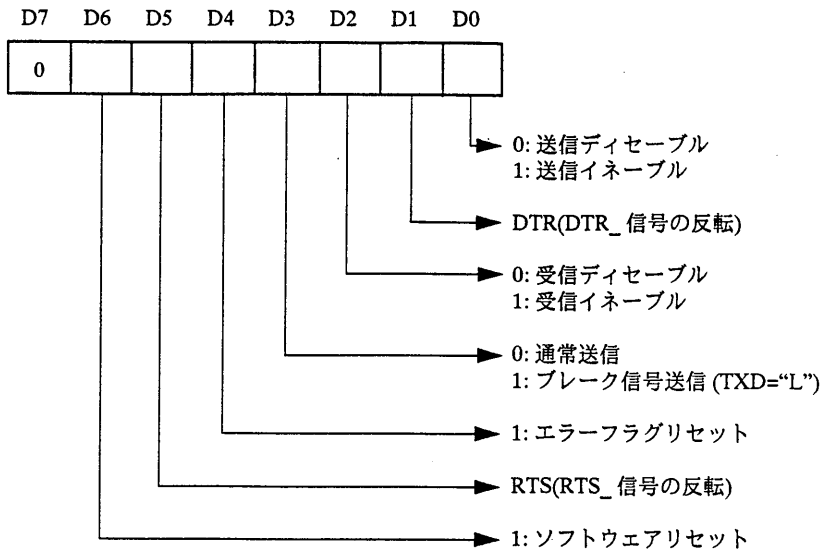
内蔵ボーレートジェネレータの動作を決定するパラメータ RATE<7:0> を設定します。8.6 節中の「ボーレートジェネレータと送受信クロックの選択」の項を参照して下さい。



#### モードレジスタ

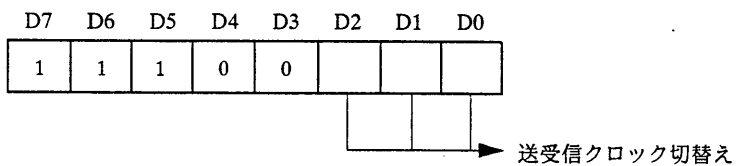


### コマンドレジスタ A



D6, D4 は単発動作のため、その動作の解除のために '0' を書き込む必要はありません。

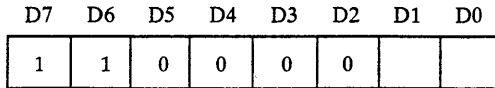
### コマンドレジスタ B



D2	D1	D0	送受信クロック
0	0	0	外部クロック (from TRXC PIN)
0	0	1	BCK<0>
0	1	0	BCK<1>
0	1	1	BCK<2>
1	0	0	BCK<3>
1	0	1	BCK<4>
1	1	0	BCK<5>
1	1	1	BCK<6>

BCK<6:0> に関しては 8.6 節中の「ボーレートジェネレータと送受信クロック選択」の項を参照して下さい。

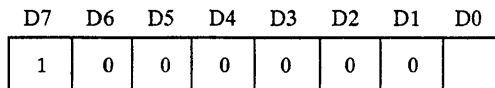
### コマンドレジスタ C



送信データビット 8  
(キャラクタ長 9 ビット時のみ有効)

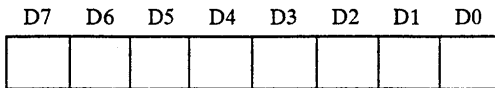
スリープモード  
0: 解除  
1: 設定

### レジスタ切替えコマンド



0: レジスタ切替え F/F クリア  
1: レジスタ切替え F/F セット

### ステータスレジスタ



TXRDY(外部 PIN の TXRDY とは若干意味が異なります。8.3 節の端子機能説明参照)

RXRDY  
RXRDY 端子と同じ機能

TXEMP  
1: 送信データバッファと送信シフトレジスタの  
どちらにも送信すべきデータが無い状態

PE(パリティエラー検出) フラグ  
1: 検出

OE(オーバーランエラー検出) フラグ  
1: 検出

FE(フレーミングエラー検出) フラグ  
1: 検出

BDET(ブ레이크検出) フラグ  
1: 検出

DSR(DSR\_ 信号の反転)

受信ディセーブル時（コマンドレジスタ A のビット 0='0'）は PE, OE, FE フラグは立ちません。またエラーフラグが立っている時に受信ディセーブルしても、エラーフラグはクリアされません。エラーフラグのクリアはエラーリセット（コマンドレジスタ A のビット 4 を '1' にする）時かハードウェア/ソフトウェアリセット時に限られます。

### 拡張ステータスレジスタ A

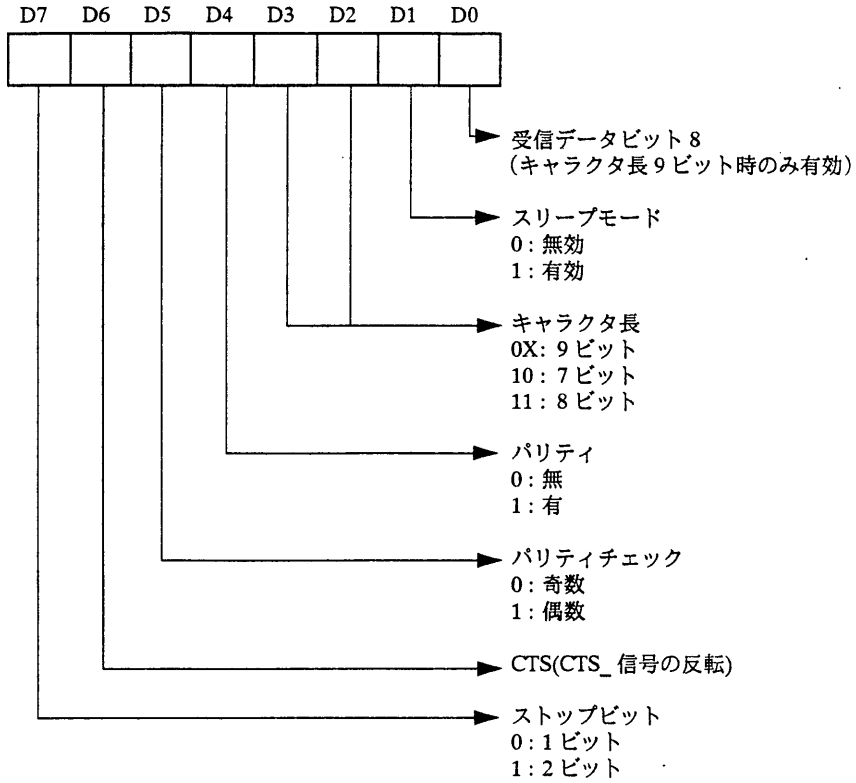
D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0				

使用送受信クロック設定

D2	D1	D0	送受信クロック
0	0	0	外部クロック (from PIN)
0	0	1	BCK<0>
0	1	0	BCK<1>
0	1	1	BCK<2>
1	0	0	BCK<3>
1	0	1	BCK<4>
1	1	0	BCK<5>
1	1	1	BCK<6>

テスト用ビット (川崎製鉄予約)。  
値は不定。

### 拡張ステータスレジスタ B



## 8.6 動作説明

### 送信動作

CPUから送信データが書き込まれると、そのデータは送信データバッファに設定されます。ここで、ブレイク信号送信状態に設定されておらず、かつ送信ディセーブル状態ならTXD端子からは“H”が出力され続けます（マーキング状態）。

CTS<sub>0</sub>=“L”かつ送信イネーブル(コマンドレジスタのビット0=“1”)状態(これを送信可能状態といいます)になると、KP61は送信データバッファ内のデータを送信シフトレジスタに設定してデータ送信を開始するとともに、ステータスレジスタのTXRDYフラグを“1”にします。これはTXRDY端子にも反映されます。

送信シフトレジスタ内のデータ送出が終了する度にKP61は送信可能状態のチェックを行いません。このとき送信データバッファ内に次の送信データが存在し、かつ送信可能状態であったならそのデータを送信シフトレジスタに設定し、連続して送信を行いません。送信データが存在しない場合は送信動作を行わず、マーキング状態を保持するとともにステータスレジスタのTXEMPフラグを“1”にします。データ送信中に送信可能状態でなくなっても、現在実行中の送信動作に影響はありません。前述のようにKP61は送信動作完了時に送信可能状態のチェックを行ない、このとき送信可能状態でなければ送信動作を停止します。

ブレイク信号送信状態(コマンドレジスタのビット3=“1”)に設定することによって、TXD出力は“L”になります。このブレイク状態は送信ブレイク解除(コマンドレジスタのビット3を“0”にする)まで継続しますが、その間も送信動作やTXRDY出力は、設定されたキャラクタ長等にしがって通常の送信状態と同じタイミングで動作します。

### 受信動作

KP61は受信イネーブル(コマンドレジスタのビット2=“1”)状態にすることによって、設定モードにしたがって受信動作を行いません。RXD端子から“H”を検出すると、スタートビット待ち状態となります。その後RXD=“L”となってから数えて送受信クロック7,8,9クロック目でRXDをサンプルし、このうち2回以上“L”であったならばスタートビットと認識します

(図8-7参照)。“L”の検出が1回以下だった場合、スタートビット待ち状態に戻ります。

スタートビット認識後、データビットを受信シフトレジスタに取り込んでいきますが、このときもスタートビット認識時と同様に、ある受信ビットにおける送受信クロックの7,8,9クロック目でRXDをサンプルし、このうち2回以上検出された値をそのビットのデータとして確定します。

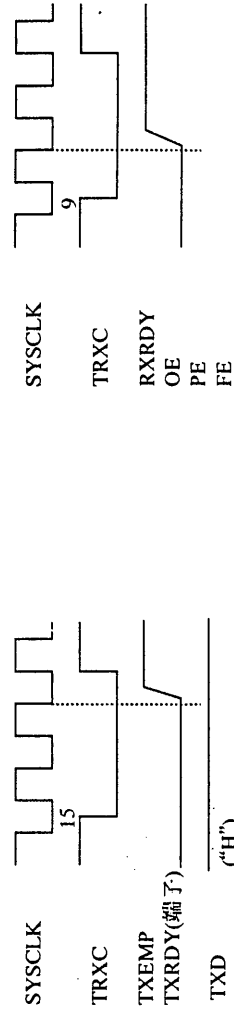
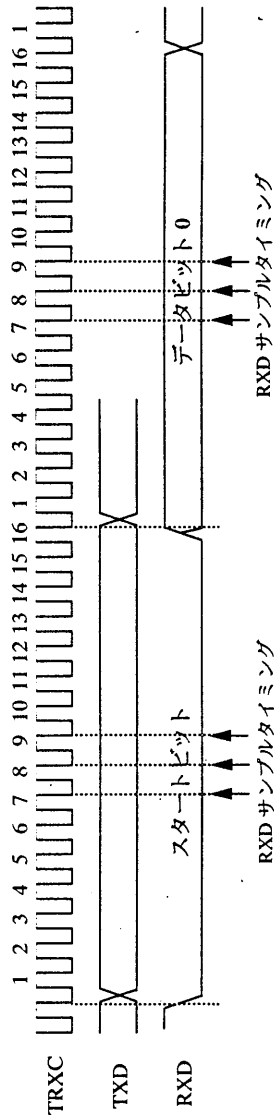
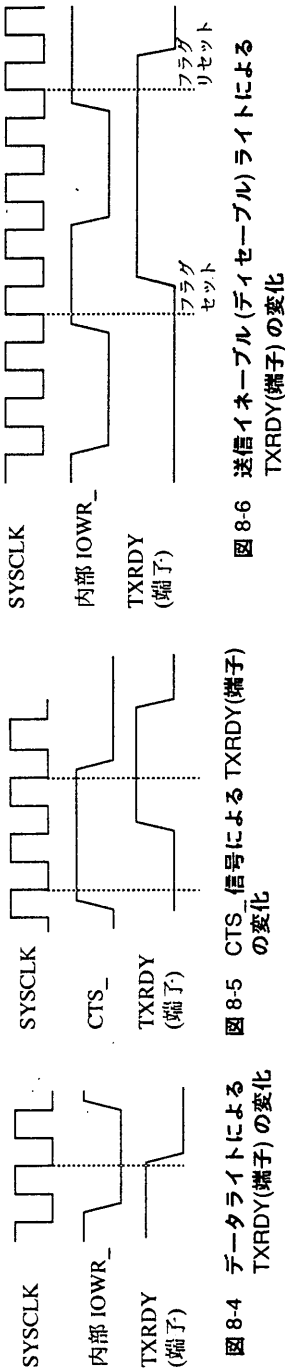
規定ビット数のキャラクタとパリティビットを受信した後ストップビットを検出すると、受信シフトレジスタから受信データバッファへ受信データが転送され、RXRDYフラグが“1”にセットされるとともにRXRDY端子からは“H”が出力されます。このとき各エラーが検出されると、そのエラーフラグも“1”にセットされます。そのタイミングは図8-9の通り、最初のストップビットの中央付近です。

なお、フレーミングエラーは最初のストップビットが“0”と認識されることによって検出されます。またオーバーランエラーは受信シフトレジスタから受信データバッファへの受信データの転送時に以前のデータが読み出されていない場合検出され、このとき受信データバッファには新しい受信データが上書きされてしまいます。これらのエラー検出は受信動作そのものには影響しません。ただしフレーミングエラー後の受信データについては保証されません。

セットされたRXRDYフラグは、CPUからの受信データバッファ読み出しにより“0”にクリアされます。このときRXRDY信号も“L”となります。

受信ディセーブル(コマンドレジスタのビット2=“0”)状態ではKP61は受信動作を行いません。またデータ受信中に受信ディセーブル状態にされると現在受信中のデータの受信完了を待たずに直ちに受信動作を中止します。したがってこのとき一切のエラー検出は行なわれません。

KP61はフレーミングエラー検出後、連続して8ビットの“0”をRXD端子から受信すると、これをブレイク信号と認識してBDET=“H”を出力します。この状態は次にRXD端子から“1”を検出するまで継続されます。したがってKP61にブレイク信号を送信する時は、少なくとも20ビット程度連続して“0”を送信することをお奨めします。なお受信ディセーブル状態ではブレイク検出は行なわれません。



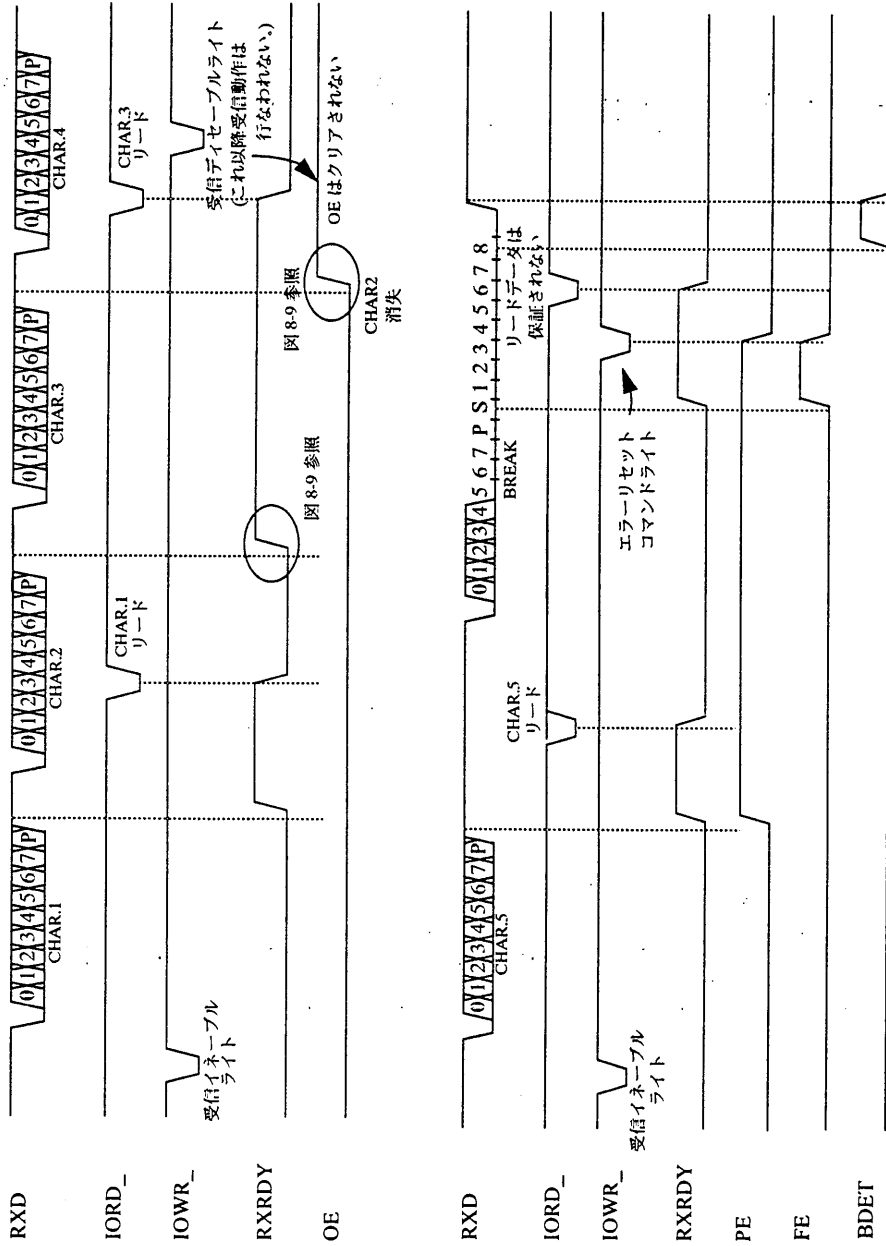


図 8-10 受信動作とフラグタイミング (キャラクタ長8ビット、パリティ付、ストップビット2の例)

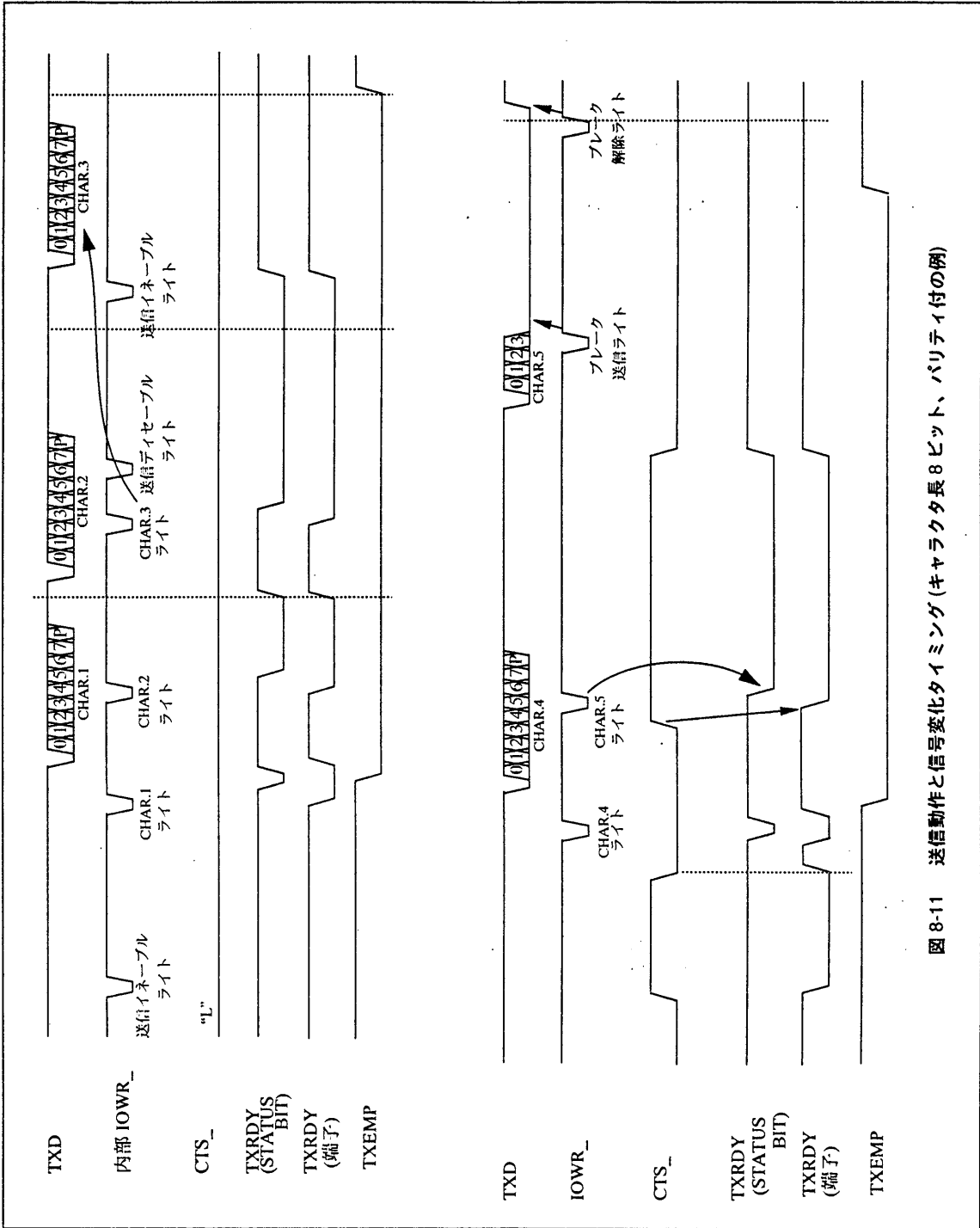


図 8-11 送信動作と信号変化タイミング (キヤラクタ長 8 ビット、パリティ付の例)

### スリーピングモード

KP61 はスリーピングモードをサポートしています。UART を用いて複数のマイコン間で通信を行なう際にこのスリーピングモードを使用します。スリーピングモードは、モードレジスタでキャラクタ長 9 ビットを選択し、かつコマンドレジスタ C の D1='1' としたときに有効になります。スリーピングモードでは、受信データのビット 8='0' のときは受信シフトレジスタの受信データを受信データバッファへ転送しません。また RXRDY もフラグ、信号ともに変化しません。

以下にスリーピングモードの使用例を示します。

- (1) 各スレーブコントローラをスリープモードに設定します。
- (2) マスターコントローラの UART からスレーブコントローラ選択アドレスを送信します。ただしその送信データのビット 8 は '1' としておきます。
- (3) 各スレーブコントローラは今受信したアドレスが自分を示すものであるか判定し、そうであればプログラムにより自分のスリーピングモードを解除します。
- (4) マスターコントローラの UART からデータを送信します。ただしその送信データのビット 8 は '0' とします。こうすることにより、指定された以外のスレーブコントローラはスリーピングモードのままなので受信データはシフトレジスタから受信データバッファに転送されません。

こうして特定のスレーブコントローラとのみ通信を行なうことができます。

スリーピングモードでは、

- ・フレーミングエラーは検出されません。
- ・パリティエラーは検出されません。
- ・オーバーランエラーは、ビット 8='1' のデータを 2 回受信してその間 1 度も受信データバッファの読み出しが行なわれないと成立します。

### ボーレートジェネレータと送受信クロック選択

KP61 は通信チャンネル 0, 1 に共通のボーレートジェネレータを内蔵しています。そして、送受信クロックとして TRXC 端子から入力される外部クロックと、ボーレートジェネレータによってシステムクロックを分周して得られる内部クロックのいずれかを選択することができます。この設定はコマンドレジスタ B によって行ないます。

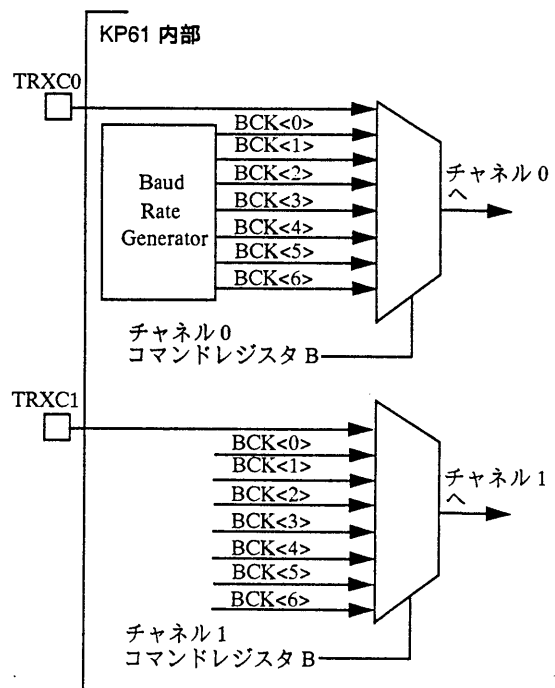


図 8-12 使用送受信クロック選択

ボーレートジェネレータの動作設定は RATE 設定レジスタに設定する RATE<7:0>によって行ないます。

RATE<7>='0' とすることにより比較的低速の内部送受信クロックが得られ、また RATE<7>='1' とすることで高速の内部送受信クロックが得られます。またボーレートジェネレータの出力は、BCK<6:0> として得られますが、このうち BCK<0> がもっとも高速で、以下

BCK<1>はBCK<0>の1/2、BCK<2>はBCK<1>の1/2となり、もっとも低速なBCK<6>からはBCK<0>の1/64の内部送受信クロックが得られます。

RATE<6:0>はボーレートジェネレータに設定する動作定数です。この値をn(10進数)とし、またシステムクロックの周波数をF<sub>sysclk</sub>[Hz]とすると、ボーレートジェネレータのBCK<i>から得られる送受信クロックによるボーレートは以下の表の式ようになります。

表 8-2 ボーレート計算式

RATE<7>	BCK<i> から得られる送受信クロックによるボーレート(BPS)
0	$\frac{F_{\text{sysclk}}}{(n+1) \times 2^{(i+1)}} \times \frac{1}{16}$
1	$\frac{F_{\text{sysclk}} \times (n+128)}{256 \times 2^{(i+1)}} \times \frac{1}{16}$

i: 0~6

n = RATE<6:0>

表 8-3 に、システムクロック周波数が 10MHz、7.159MHz、3.932MHz の場合について、上の式で計算したボーレート例とそのときの RATE 設定レジスタへの設定値を示します。

### 8.7 リセット

KP61 はハードウェア/ソフトウェアリセットにより以下の状態に初期化されます。ただし、ソフトウェアリセットでリセットされるのはそのチャンネルに属する信号、フラグ等のみです。

送信ディセーブル

受信ディセーブル

各エラーフラグ='0'

TXRDY(フラグ)='1'、 TXRDY(端子)='L'

RXRDY(フラグ)='0'、 RXRDY(端子)='L'

BDET(フラグ)='0'、 BDET(端子)='L'  
TXEMP='0' (これはモードライトによって'1'となります。)

DTR(フラグ)='0'、 DTR(端子)='H'

RTS(フラグ)='0'、 RTS(端子)='H'

DSR(フラグ)='0'、 DSR(端子)='H'

レジスタ切替えF/F : クリア('0')

RATE 設定レジスタ、コマンドレジスタ B の送受信クロック選択ビット (D2,D1,D0)、およびコマンドレジスタ C のスリーピングモードビット (D1) はハードウェアリセットではリセットされますがソフトウェアリセットではリセットされません。この場合ソフトウェアリセット直前の設定データが残っていますのでご注意ください。

### 8.8 注意事項

- ・各ステータスレジスタやデータレジスタを読み出した場合、リードを開始した時点でのデータが読み出されます。読み出し中に各フラグや受信データバッファの内容が変化しても、読み出しデータにその影響は現れません。
- ・通信キャラクタ長で 7 ビットを選択した場合、KL5C80A16 搭載の KP61 の現バージョンでは、受信データのビット7の値は不定となります。
- ・読み出しレジスタの切替えは基本的にチャンネル間で独立に行なわれますが、KL5C80A16 搭載の KP61 の現バージョンでは、拡張ステータス B の読み出し後に行なわれるレジスタ切替えF/Fのクリアのみ、チャンネル0とチャンネル1両方同時に行なわれます。
- ・パラレルポートと多重化されている各信号端子については、3章を参照して下さい。また各端子の詳しい使い方は12章を参照して下さい。

表 8-3 ボーレート例一覧表 (ボーレート精度は誤差0.5%未満)

システムクロック 周波数(MHz)	RATE<7>	RATE<6:0>	BCK<i> i=0-6	ボーレート (BPS)	備考
10	1	124	BCK<0>	307200	RATE<7:0>=FCH
			BCK<1>	153600	
			BCK<2>	76800	
			BCK<3>	38400	
			BCK<4>	19200	
			BCK<5>	9600	
	0	64	BCK<0>	4800	RATE<7:0>=40H
			BCK<1>	2400	
			BCK<2>	1200	
			BCK<3>	600	
			BCK<4>	300	
			BCK<5>	150	
7.159	1	48	BCK<0>	153600	RATE<7:0>=B0H
			BCK<1>	76800	
			BCK<2>	38400	
			BCK<3>	19200	
			BCK<4>	9600	
			BCK<5>	4800	
	0	92	BCK<0>	2400	RATE<7:0>=5CH
			BCK<1>	1200	
			BCK<2>	600	
			BCK<3>	300	
			BCK<4>	150	
			BCK<5>	75	
3.932	1	32	BCK<0>	76800	RATE<7:0>=A0H
			BCK<1>	38400	
			BCK<2>	19200	
			BCK<3>	9600	
			BCK<4>	4800	
			BCK<5>	2400	
	0	50	BCK<0>	2400	RATE<7:0>=1AH
			BCK<1>	1200	
			BCK<2>	600	
			BCK<3>	300	
			BCK<4>	150	
			BCK<5>	75	
		BCK<6>	-----		

### 9. タイマ/カウンタ

#### 9.1 概要

KL5C80A16のタイマ/カウンタはKP63A(16ビット高機能タイマ/カウンタマクロセル)です。KP63Aはマイクロコンピュータ用に設計された8ビットプリスケアラつきクロック同期式16bitプログラマブル・バイナリカウンタ/タイマです。KP63Aは、4種類の動作モード(パルス幅変調(PWM)モード、連続カウントモード、単発カウントモード、ウォッチドックタイマモード)をもつダウンカウンタ4チャンネルで構成され、CPUからのモード制御命令によりモード設定できます。カウント値は、バスから読み出すことができます。また、ステータスリードコマンドで設定モードやOUT出力端子の状態を読み出すことができます。

#### 特徴

- ・8ビットプリスケアラつき。GATE入力から外部クロックをとりこみカウント可能。
- ・4チャンネルの16bitダウンカウンタ/タイマを内蔵し各々4種類の動作モード設定、動作が可能
- ・カウント動作に影響を与えずに、安定したカウント値の読み出しが可能
- ・設定モード、OUT出力の状態等のステータスの読み出しが可能

#### 9.2 ブロック図

タイマ/カウンタの全体ブロック図とI/Oレジスタマップを示します。各外部端子以外に各チャンネルの割り込み要求信号が内蔵の割り込みコントローラに接続されています。チャンネル3の割り込み要求信号はCPUのNMI\_入力にも接続されています。

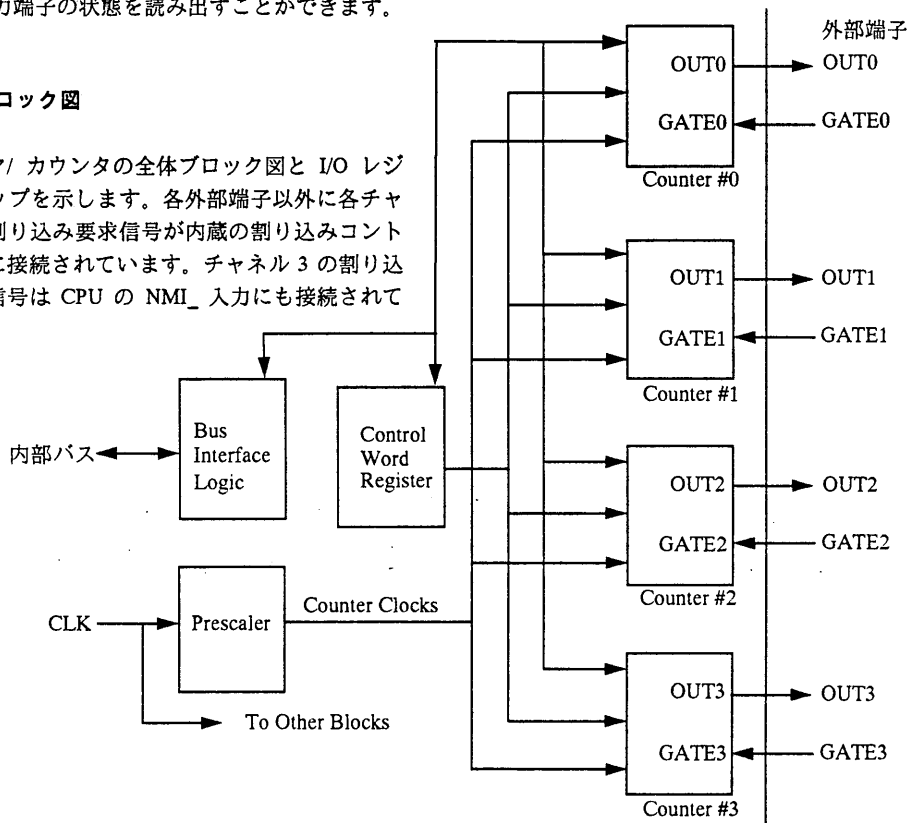


図 9-1 タイマ/カウンタのブロック図

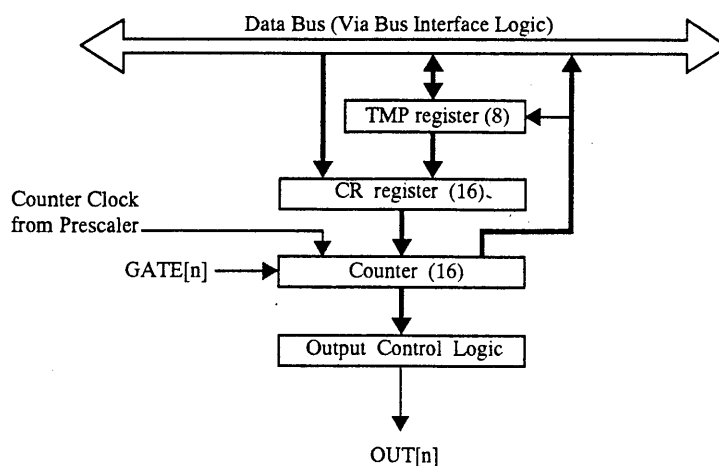


図 9-2 各チャンネルカウンタ・ユニット内のブロック図

表 9-1 I/O レジスタマッピング

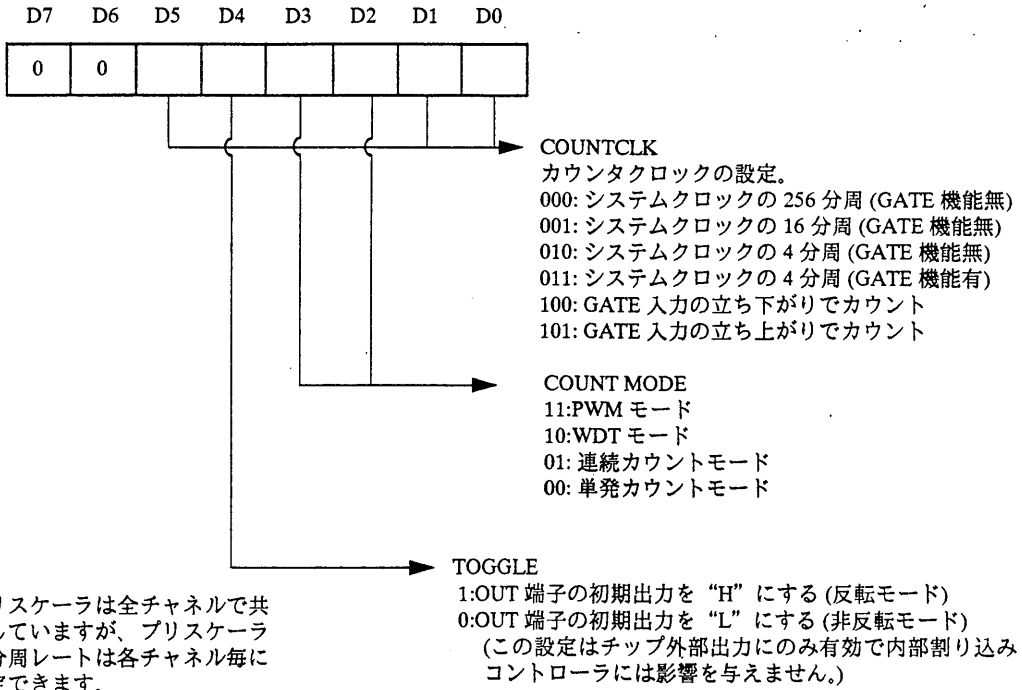
I/O アドレス	ライト時	リード時
20H	チャンネル 0 カウンタ	チャンネル 0 カウンタ
21H	チャンネル 0 コントロールワード	チャンネル 0 ステータス
22H	チャンネル 1 カウンタ	チャンネル 1 カウンタ
23H	チャンネル 1 コントロールワード	チャンネル 1 ステータス
24H	チャンネル 2 カウンタ	チャンネル 2 カウンタ
25H	チャンネル 2 コントロールワード	チャンネル 2 ステータス
26H	チャンネル 3 カウンタ	チャンネル 3 カウンタ
27H	チャンネル 3 コントロールワード	チャンネル 3 ステータス

### 9.3 端子機能説明

端子名	I/O	機能説明
GATE3~GATE0	I	ゲート入力端子 各チャンネルのカウンタに対するカウント動作イネーブル/ディセーブルの指定を与えます。またモード設定により外部カウンタクロックとしても機能します。
OUT3~OUT0	O	パルスアウト出力端子 モードの指定によって、分周出力、PWM 出力が得られます。リセット信号入力により本出力はシステムクロックとは非同期に“L”になります。

### 9.4 モード設定

モード設定は各チャンネルにコントロールワードを書き込むことで行われます。



### 9.5 カウンタへのカウント初期値の設定

カウンタは 16bit 長のダウンカウンタですが、データバスが 8bit 幅なので、データ書き込みは下位バイト、上位バイトの順に 2 回に分けて行います。ただし、書き込み用と読み込み用で TMP レジスタを共用しているので、書き込みを完了する前にカウンタの読み出しを行うと、書き込みデータが壊れます。

下位側:

D7	D6	D5	D4	D3	D2	D1	D0
7	6	5	4	3	2	1	0

(数字はデータビットを表す)

上位側:

D7	D6	D5	D4	D3	D2	D1	D0
15	14	13	12	11	10	9	8

(数字はデータビットを表す)

### 9.6 カウンタの読み出し

カウンタの読み出しは、各カウンタのアドレスをアクセスして読み出すことになります。その際、8bit 幅で 2 回に分けて、下位バイト、上位バイトの順で行います。上位側は下位側を読み出す時に TMP レジスタに移されたカウンタの上位バイトなので、2 回の読み出し動作中にカウンタの値が変化しても、読み出され

る値は変化しません。ただし、読み出し用 TMP レジスタと書き込み用 TMP レジスタを共用しているため、読み出しを完了する前にカウント初期値の書き込みを行うと、読み出しデータが壊れます。

下位側：

D7	D6	D5	D4	D3	D2	D1	D0
7	6	5	4	3	2	1	0

(数字はデータビットを表す)

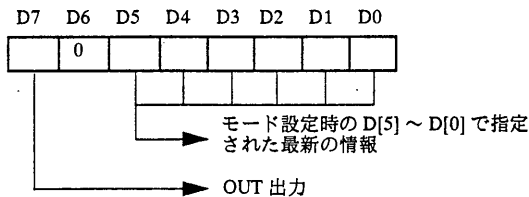
上位側：

D7	D6	D5	D4	D3	D2	D1	D0
15	14	13	12	11	10	9	8

(数字はデータビットを表す)

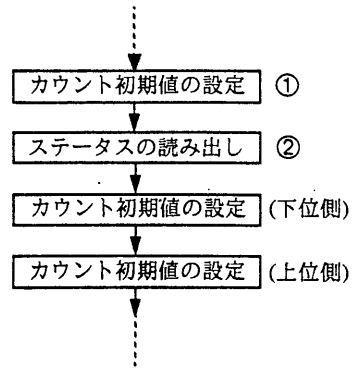
### 9.7 ステータスの読み出し

ステータスは各チャンネルごとに用意されています。ステータスのフォーマットは以下の通りです。



なお、ステータスの読み出しを行うと、リードライト・シーケンスがクリアされます(下例参照)。したがってカウント初期値の設定やカウンタの読み出しを行う前に一度ステータスの読み出しを行うことにより、誤りを防ぐことができます。

(例)



- ①は上位側、下位側、どちらでも良い。
- ②でリードライト・シーケンス・クリアされる。

### 9.8 動作モード

以下の説明はモード設定時、OUT 非反転モード扱の場合について行います。

#### 連続カウントモード

連続カウントモードでは、カウント初期値をロードした後デクリメントし、カウント値がゼロになると OUT 出力が変化すると共に初期値をリロードした後デクリメントを繰り返します。

モード設定によって、OUT 端子は“L”を出力します。カウント動作はカウント初期値をロードすることで開始し、OUT 端子に“H”を出力します。カウント値がゼロになる度に OUT 出力はトグルします。

リロードはカウント値がゼロになることでのみ発生し、カウント中に初期値を書き換えてもリロードは発生しません。

#### 単発カウントモード

単発カウントモードでは、カウント初期値をロードした後デクリメントし、カウント値がゼロになると OUT 出力が変化し、カウント動作は次にカウント初期値が設定されるまで停止します。

モード設定によって OUT 端子は“L”を出力します。カウント動作はカウント初期値の設定により開始

し、OUT 端子はカウント値がゼロになるまで“H”を出力します。リロードは、カウント初期値の設定により行われます。

#### ウォッチドッグタイマ(WDT)モード

ウォッチドッグタイマモードでは、CRレジスタ(カウント初期値書き込み用レジスタ)への書き込みはできません。CRレジスタへの書き込み動作はカウント開始およびリトリガとして認識されます。

モード設定により、カウンタは動作を停止し、OUT端子は“L”を出力します。カウント動作はCRレジスタへの書き込み動作(書き込み動作時のデータ値には無関係; 1回のライト動作)により、CRレジスタからデータをロードして開始し、カウントを開始すると、OUT端子は“H”を出力します。

また、カウント中にCRレジスタへの書き込み動作で再び、CRレジスタからデータをロードしてカウントダウンを行います。

ただし、一旦カウント値がゼロになると、OUT端子は“L”になり、次にCRレジスタへの書き込み動作が行われるまでカウント動作は停止します。

なお、モード設定では、CRレジスタが初期化されません。CRレジスタにカウント初期値を設定するには次の様な手順を用います。すなわち、単発カウントモードにモード設定後、CRレジスタにカウント初期値を設定し、再度モード設定によりWDTモードにします。そして、CRレジスタへの書き込み動作を行うと、先に設定したCRレジスタの値をロードして、カウントダウンを開始します。

また、カウント中にCRレジスタへの書き込み動作を行うことにより、再度CRレジスタの値をロードしてカウントダウンを実行します。

以上の手順によれば、WDTモードにおいて、任意のカウント初期値を設定することが可能となります。

なお、本タイマ/カウンタをウォッチドッグタイマとして使用する場合はこのモード設定をするだけでなく、本タイマ/カウンタ出力をKC82のNMI<sub>1</sub>入力に内部で接続するようにSCRで設定をする必要があります。詳しくは12章を参照下さい。またウォッチドッグタイマモードは全チャンネルで使用可能ですが、チップ内部でタイマ/カウンタの割り込み要求をKC82のNMI<sub>1</sub>入力に接続できるのはチャンネル3のみです。

#### パルス幅変調(PWM)モード

パルス幅変調モードでは、カウント初期値により設定する、パルス幅およびパルス周期により構成される繰り返しパルスをOUT端子に出力します。

パルス幅はCRレジスタの上位バイトで、パルス周期はCRレジスタの下位バイトによって設定します。まず、パルス周期の設定に関して説明します。

パルス周期は、CRレジスタの下位側バイトをカウンタにロードした後、上位バイトと独立にデクリメントし、下位バイトのカウント値がゼロになるまでの時間となります。すなわち、(1回目のCRレジスタへの書き込みにおける設定値+1)の時間を意味します。

一方、パルス幅はCRレジスタの上位側バイトをカウンタにロードした後、下位バイトと独立にデクリメントし、上位バイトのカウント値がゼロになるまでの時間となります。すなわち、(2回目のCRレジスタへの書き込みにおける設定値+1)の時間を意味します。

このように上位、下位ともに同時にカウントダウンを独立に行いますので、任意のパルスがOUT端子から得られることとなります。

モード設定によってカウンタは停止し、OUT端子は“L”を出力します。カウント動作はカウント初期値を上位、下位共に設定後、ロードすることにより開始し、カウントダウンは上位バイト、下位バイトずつ各々独立に行います。ただし、設定した値が(パルス幅 $\geq$ パルス周期)の場合には、常にOUT=“H”となります。

リロードは下位バイトがゼロになることでのみ発生し、カウント中に初期値を書き込むことによるリロードは発生しません。

(例:4分周設定時、カウント初期値 = '0308H' とした場合

パルス幅は  $4(=3+1) \times 4=16$  システムクロック

パルス周期は  $9(=8+1) \times 4=36$  システムクロック  
となります。)

#### 9.9 OUT出力と割り込み要求について

OUT出力は、モード設定により反転が可能な信号です。以下の説明はモード設定時、OUT非反転モード選択の場合について行ないません。連続モード時に

は、カウント値がゼロになる度にトグルします。単発モードおよび WDT モードでは、通常時カウント中に“H”を出力し、カウント値がゼロになると“L”を出力します。PWM モードでは、所望のパルス幅および周期のパルス信号を出力します。以上の様に OUT 出力は様々なパルス信号を得る為に利用することが可能です。

割り込み要求は OUT 出力と異なり、モード設定によらず常にカウント値がゼロになったとき発生します。

### 9.10 動作説明

ここでは、タイマ/カウンタの動作に関し、図を用いて説明します。

#### カウントダウンタイミング

タイマ/カウンタのカウントダウンはチャンネル 0、チャンネル 1、チャンネル 2、チャンネル 3 の順に 1 システムクロック遅れて、システムクロックの立ち下がりで行われます。

#### 各カウンタクロック選択時のカウントダウンタイミング (図 9-3)

図 9-3 はチャンネル 0、チャンネル 1 に GATE 機能無の 4 分周、チャンネル 2 に 16 分周、チャンネル 3 に 256 分周のカウントクロックを設定した場合です。

チャンネル 0、チャンネル 1 では、それぞれ 1 から 1'、2 から 2' 迄の 1 回のカウントダウン迄の時間が 4 システムクロックであり、3 から 3' 迄が 16 システムクロック、4 から 4' 迄が 256 システムクロックとなります。

#### GATE 信号のサンプルタイミング (図 9-4-A)

図 9-4-A は全チャンネルに GATE 機能無の 4 分周を設定した場合です。GATE 信号のサンプルは、4 システムクロックに 1 回 (図中内部 GATE 入力サンプル信号の立ち上がり)、全チャンネル分を同時に行います。したがって、この時サンプルした GATE 信号の極性により、次のカウントダウンタイミングでカウントダウンするかしないかが判断されます。GATE 機能が使

えるのは 4 分周の分周レートのみです。

#### 外部クロックのカウントタイミング (図 9-4-B)

9-4-B は外部クロックとしてチャンネル 0 に外部カウントクロックの立ち下がり、チャンネル 1 に外部カウントクロックの立ち上がりを設定した場合の例を示しています。外部カウントクロックは GATE 入力から取り込まれます。カウントエッジはモード設定により立ち下がりが立ち上がりか選べます。外部カウントクロックの H パルス幅、H パルス幅は 4 システムクロック以上にして下さい。また実際のカウントは GATE 入力のエッジから 1~3 クロック遅れる場合があります。

#### カウントダウンタイミングとコントロールワードの書き込み

各チャンネルに対するコントロールワード書き込みが反映されるのはタイマ/カウンタのカウントダウンと同様にチャンネル 0、チャンネル 1、チャンネル 2、チャンネル 3 の順に 1 システムクロック遅れて、システムクロックの立ち下がりで行われます。したがって実際にコントロールワード書き込みが反映されるまでの時間はその場合によって異なり、最も短い場合 1 クロック、最も長い場合 4 クロックです。

#### カウントダウンタイミングとカウント値書き込み

連続カウントモードと PWM モードの場合の各チャンネルに対するカウント値書き込みが反映されるのはタイマ/カウンタのカウントダウンと同様にチャンネル 0、チャンネル 1、チャンネル 2、チャンネル 3 の順に 1 システムクロック遅れて、システムクロックの立ち下がりで行われます。したがって実際にカウント値書き込みが OUT 出力に反映されるまでの時間はその場合によって異なり、最も短い場合 1 クロック、最も長い場合 4 クロックです。

#### 連続カウントモードの動作説明 (図 9-5)

連続カウントモードについて、GATE 機能無の 4 分周、OUT 非反転モード選択時を例にとり、図 9-5 を用いて説明します。これはチャンネル 0 の場合です。

モード設定を行うと、次のカウントダウンタイミングで OUT 信号が初期化され、“L”になります。下位カウント値 '02H'、上位カウント値 '00H' を書き込

むと、次のカウントダウンタイミングでカウンタに '0002H' がロードされ、カウントダウンを開始すると共に、OUT 出力がトグルします。

カウント中、すなわち図中の 1 から 2 の間にカウント初期値 'F0ABH' を再設定すると、次にカウント値がゼロとなった次のカウントダウンタイミング、すなわち 2 の時点でカウント値が 'F0ABH' になります。

また、それと同時に OUT 出力がトグルします。

### 単発カウントモードの動作説明 (図 9-6)

単発カウントモードについて、GATE 機能無の 4 分周、OUT 非反転モード選択時を例にとり、図 9-6 を用いて説明します。これはチャンネル 0 の場合です。

モード設定を行うと、次のカウントダウンタイミングで OUT 信号が初期化され、"L" になります。下位カウント値 '02H'、上位カウント値 '00H' を書き込むと、次のカウントダウンタイミングでカウンタに '0002H' がロードされ、カウントダウンを開始すると共に OUT 出力が "H" となります。

カウント中、すなわち図中の 1 から 2 の間にカウント初期値 '0003H' を再設定すると、次のカウントダウンタイミング (図中の 2) で、リトリガがかかります。そして、この再設定値からカウントダウンを続けます。そして 2 から 3 の間にカウント初期値の再設定がなく、カウンタがゼロになると (図中の 3) 'FFFFH' となり、カウントダウンを停止し、OUT 出力は "L" になります。

なお、1 にあるような場合、すなわちカウント値がゼロから 'FFFFH' になるタイミングでカウント初期値 'ABCDH' の再設定を認識した場合にはリトリガと判断し、OUT 信号は変化しません。

### ウォッチドッグタイマ (WDT) モードの動作説明 (図 9-7)

ウォッチドッグタイマ (WDT) モードについて GATE 機能無の 4 分周、OUT 反転モード選択時を例にとり、図 9-7 を用いて説明します。

WDT モードでは、カウント初期値の書き込み (CR レジスタへの書き込み) 動作がリトリガと解釈され、カウント初期値レジスタの値は変化しません。カウント初期値レジスタに所望の値を設定するためには、一旦、単発カウントモードに設定し、CR レジスタに値

を書き込み、WDT モードに設定し直します。

WDT モードの基本動作は単発カウントモードに準じます。まず、単発カウントモードにモード設定を行うと、次のカウントダウンタイミングで OUT 出力が "H" になります。下位カウント値 '03H'、上位カウント値 '00H' を書き込むと、次のカウントダウンタイミングでカウンタに '0003H' がロードされ、カウントダウンを開始すると共に OUT 出力が "L" になります。

次に WDT モードに再度モード設定を行います (図中の 1)。すると、次のカウントダウンタイミングで OUT 出力が "H" になり、このときカウンタは停止状態です。次にカウント値書き込み動作 (データは任意) を 1 回行うと、次のカウントダウンタイミングでカウンタに CR レジスタの値 '0003H' がロードされ、WDT モードとして初めて、カウントダウンを開始すると共に OUT 出力が "L" となります (図中の 2)。

カウント中、すなわち図中の 2 から 3 の間にカウント書き込み動作 (データは任意) を行うと、リトリガとして認識され、次のカウントダウンタイミングで再び '0003H' をロードします (図中の 3)。そしてまたカウントダウンを開始し、カウント値がゼロになると、次のカウントダウンタイミングで OUT 端子に "H" を出力し、カウント値を 'FFFFH' にクリアし、停止します (図中の 4)。

その後、カウント値書き込み動作 (データは任意) を 1 回行うと、次のカウントダウンタイミングでカウンタに '0003H' がロードされ、再度カウントダウンを開始します。

OUT 出力は、一旦単発カウントモードに設定することにより、WDT としては不要な変化をしますが、内部で CPU の NMI<sub>1</sub> に接続されているダイマ/カウンタチャンネル 3 の割り込み要求はカウント値がゼロになるまで、パルスを出力しません。

### パルス幅変調 (PWM) モードの動作説明 (図 9-8)

パルス幅変調 (PWM) モードについて、GATE 機能無の 4 分周、OUT 非反転モード選択時を例にとり、図 9-8 を用いて説明します。これは、チャンネル 0 の場合です。

基本動作は連続カウントモードに準じます。モード

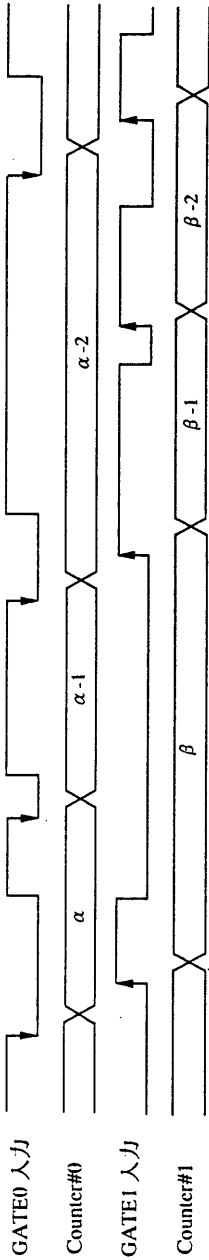


図 9-4-B GATE 入力からのクロック入力タイミング (チャネル 0 は GATE の立ち下がりでカウント、チャネル 1 は立ち上がりでカウント)

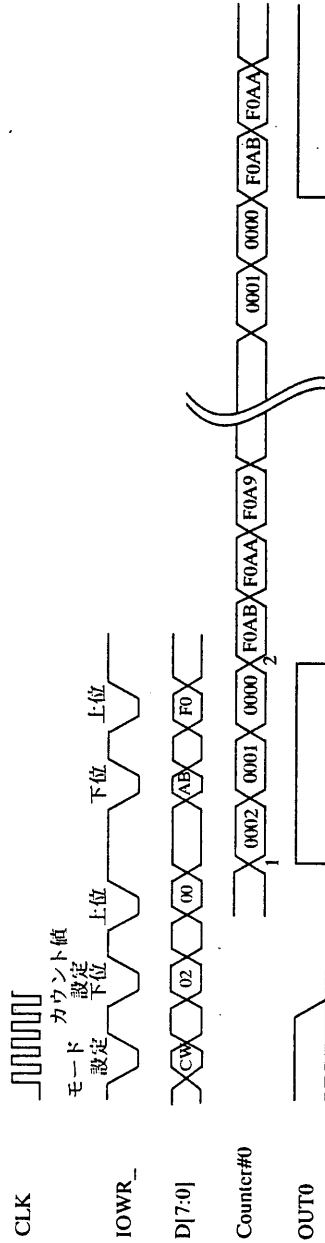


図 9-5 連続カウントモード (GATE 機能無 4 分周、OUT 非反転モード選択時)

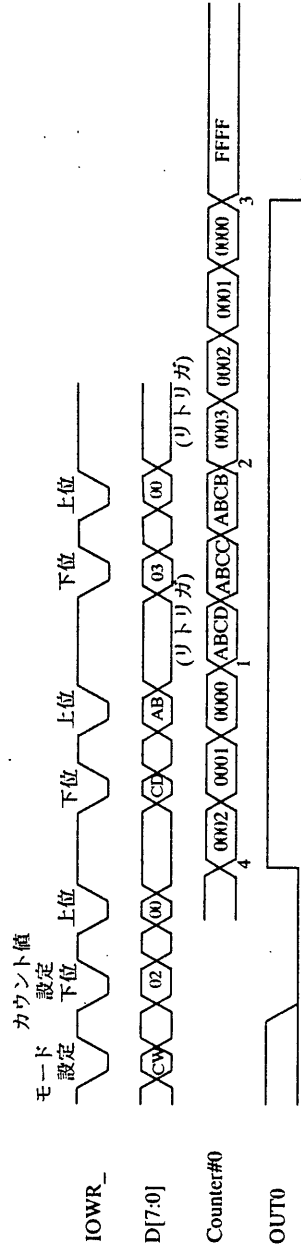


図 9-6 単発カウントモード (GATE 機能無 4 分周、OUT 非反転モード選択時)

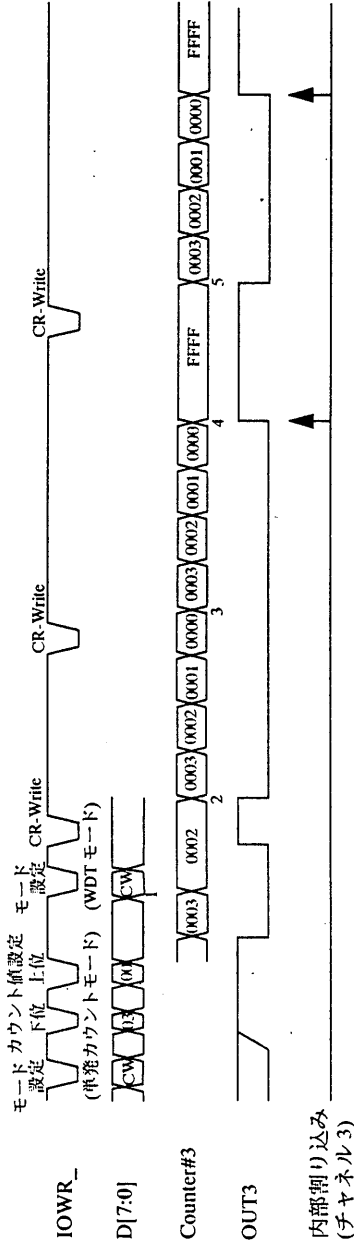


図 9-7 WDT モード (GATE 機能無 4 分周、OUT 反転モード選択時)

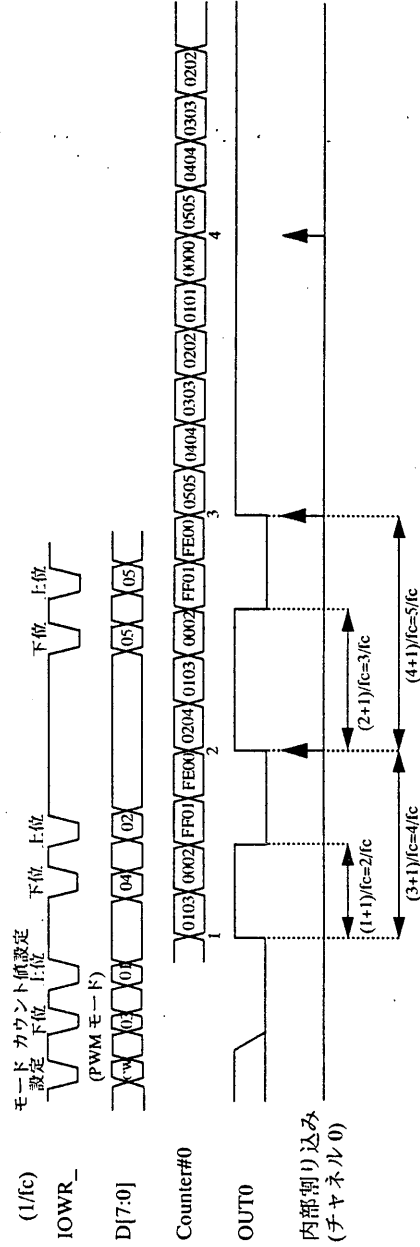


図 9-8 PWM モード (GATE 機能無 4 分周、OUT 非反転モード選択時)

### 10. クロック同期シリアル I/O

#### 10.1 概要

KL5C80A16 のクロック同期シリアル I/O は KP62 マクロセルを 2 チャンネル使って構成しています。キャラクタ長は 8 ~ 16 ビットが指定可能で、キャラクタ転送方向も LSB 先頭 / MSB 先頭が選択できます。また、シリアルクロックとして外部クロックに加え内部クロックも選択でき、シングルチップマイコンや AD/DA コンバータ等との接続に、柔軟に対応できます。このクロック同期シリアル I/O は 2 チャンネルとも同じ構成ですのでこの章では 1 チャンネル分の説明を載せています。

#### 特徴

- ・クロック同期式半二重通信方式
- ・最大ボーレートは 5Mbps (外部シリアルクロック選択時、5MHz)
- ・キャラクタ長は 8 ~ 16 ビットが指定可能
- ・キャラクタ転送方向は LSB 先頭 / MSB 先頭が選択可能
- ・内部シリアルクロック (システムクロックの 8 分周) / 外部シリアルクロックが選択可能

#### 10.2 ブロック図

クロック同期シリアル I/O の全体ブロック図を以下に示します。

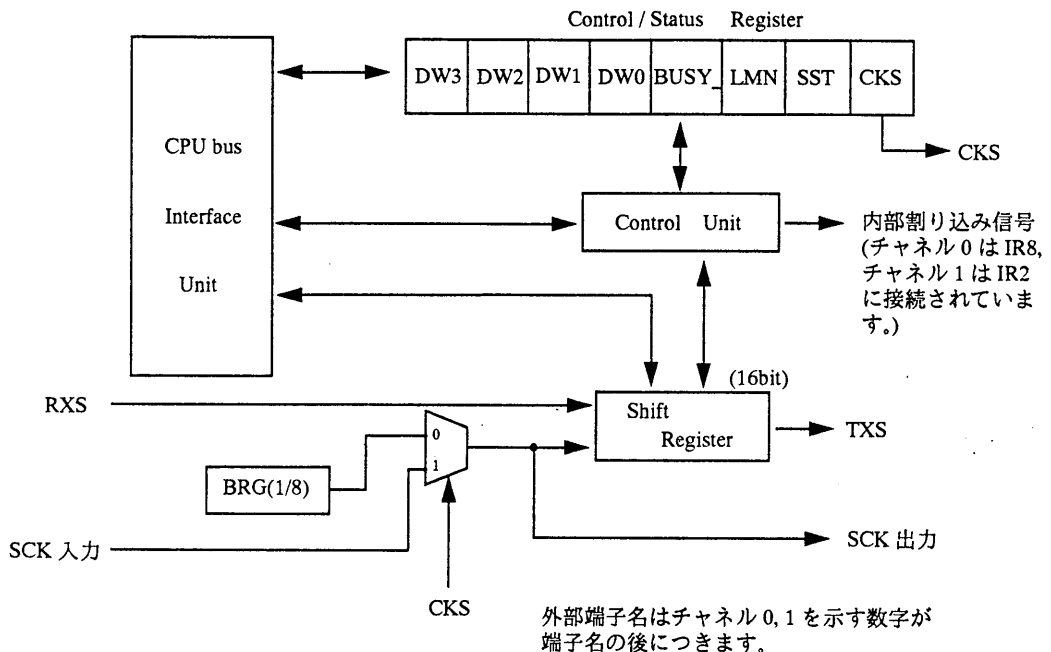


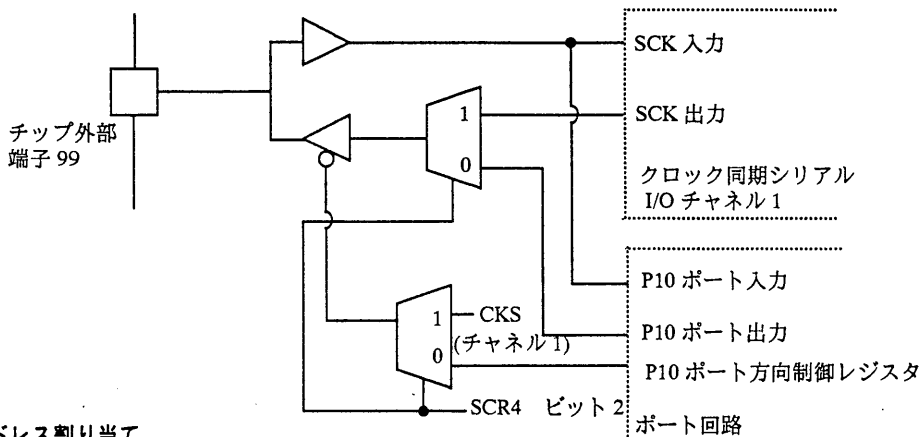
図 10-1 クロック同期シリアル I/O のブロック図

### 10.3 端子機能

端子名	I/O	機能説明
SCK0 SCK1	I/O	シリアルクロック入出力端子 入力設定時は外部シリアルクロックを選択して、通信を行う際のボーレート を制御するクロック入力です。出力設定時はクロック同期シリアル I/O 内部 で生成するシリアルクロックを出力します。
RXS0 RXS1	I	受信データ入力端子 受信データをシリアル入力する端子です。
TXS0 TXS1	O	送信データ出力端子 送信データをシリアル出力する端子です。

#### チャンネル1 SCKの内部接続

チャンネル1のSCK端子はポートと兼用になっており、以下のように内部で接続されています。



### 10.4 I/Oアドレス割り当て

レジスタのI/Oアドレス割り当てを表10-1に示します。

表10-1 I/Oアドレス割り当て

I/Oアドレス	ライト時	リード時
30H	チャンネル0送信データ	チャンネル0受信データ
31H	チャンネル0モード/コマンド	チャンネル0ステータス
32H	チャンネル1送信データ	チャンネル1受信データ
33H	チャンネル1モード/コマンド	チャンネル1ステータス

### 10.5 初期設定

データの送受信を始める前に、リセット後図10-2に示す様にモード設定とコマンドの書き込みを行う必要があります。

#### モード設定

リセット後、図10-2にしたがってモード設定を行ってください。クロック同期シリアル I/O はリセット後は必ずこのモード設定待ちの状態になっています。再度モード設定を行う際にはステータスの BUSY\_ が '1' であることを確認してから再設定してください。但し、「外部シリアルクロック、8ビットキャラクタ長、SST=0、MSB先頭」を選択する場合は、このモード設定を省略することができます。

#### モードレジスタ

##### D0: CKS

このビットを '0' にすると、クロック同期シリアル I/O 内の B.R.G. (ポーレートジェネレータ) 出力である、システムクロックを 8 分周した信号がシリアルクロックとして選択されます。SCK端子にはこの8分周信号が出力されます。

'1' にすると、SCK端子に入力される信号をシリアルクロックとして使用します。クロック同期シリアル I/O 内では、この信号をシステムクロックの立ち上がりでサンプルして使用しています。

##### D1: SST

このビットを '0' にすると、送信タイミングがシリアルクロックの立ち下がり、受信時サンプルタイミングがシリアルクロックの立ち上りに指定されます。なお、シリアルクロックに内部クロックを指定すると SCK 端子に出力される信号の初期値が "H" になります。

'1' にすると、送信タイミングがシリアルクロックの立ち上がり、受信時サンプルタイミングがシリアルクロックの立ち下りに指定されます。なお、シリアルクロックに内部クロックを指定すると、SCK 端子に出力される信号の初期値は "L" になります。

##### D2: LMN

このビットを '0' にすると、通信における先頭ビットがMSBとなり、'1' にするとLSBが先頭ビットとなります。

##### D7~D4: DW3~DW0

キャラクタ長を指定します。指定可能なキャラクタ長は8~16ビットです。

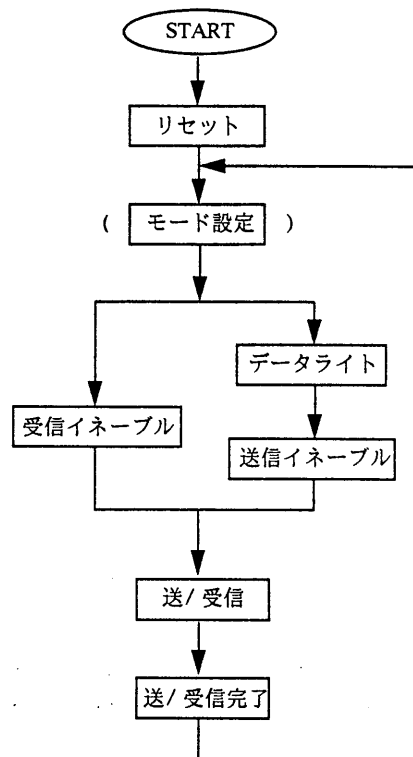
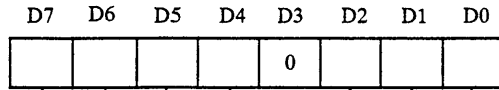


図 10-2 初期設定フロー

### モードレジスタ



**CKS**  
シリアルクロック選択  
0: 内クロック (システムクロックの 8 分周)  
1: 外クロック (SCK 入力)

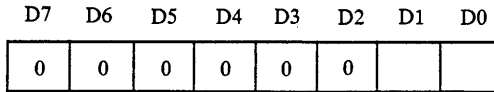
**SST**  
シフトタイミング/ サンプルタイミング  
0: 内部クロック選択時  
SCK 出力初期値 = "H"  
1: 内部クロック選択時  
SCK 出力初期値 = "L"

**LMN**  
LSB/MSB 先頭切替  
0: MSB 先頭  
1: LSB 先頭

DW3	DW2	DW1	DW0	キャラクタ長選択
0	1	1	1	8
1	0	0	0	9
1	0	0	1	10
1	0	1	0	11
1	0	1	1	12
1	1	0	0	13
1	1	0	1	14
1	1	1	0	15
1	1	1	1	16

(注) ここにあげた以外のビット組合せは予約済です。  
使用された場合の動作保証はいたしかねますので、御注意下さい。

### コマンドレジスタ

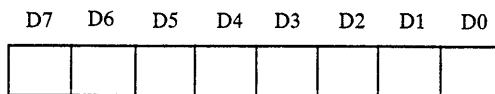


D1	D0	コマンド
0	0	キャンセル
1	0	送信イネーブル
1	1	受信イネーブル
0	1	使用禁止

(注) 上にあげた以外のビット組合せは予約済です。使用された場合の動作保証はいたしかねますので、御注意下さい。

### ステータスレジスタ

ステータス情報は、I/O アドレス =31H(チャンネル 0),  
33H(チャンネル 1) で読み出すことができます。  
ステータスのフォーマットは以下の通りです。



モードレジスタに対応する  
ビットと同一

- D0: CKS
- D1: SST
- D2: LMN
- D4: DW0
- D5: DW1
- D6: DW2
- D7: DW3

} キャラクタ長

D3: BUSY\_  
0: 送信中または受信  
1: 待機中

### コマンド設定

コマンドとして送信イネーブル、受信イネーブル、キャンセルが用意されています。送信イネーブルによりクロック同期シリアル I/O は送信を開始します。同様に、受信イネーブルにより受信を開始します。キャンセルは通信中にも受け付けられ、クロック同期シリアル I/O はモード設定直後の待機状態に移ります。これに対し、送信イネーブルと受信イネーブルコマンドはどちらも通信中には受け付けられませんので、ステータスのBUSY\_が'1'の時に書き込んでください。

## 10.6 動作の説明

### 受信動作

図 10-2にあるように、待機状態、すなわちモード設定後、ステータスの BUSY\_ が '1' のときに、受信イネーブルコマンドを書き込むことにより、受信動作を開始します。受信動作の開始によりステータスの BUSY\_ が '0' になります。また、シリアルクロックとして内部クロックを選択している場合は、SCK 出力端子にシリアルクロックが出力開始されます。

モード設定時に指定したサンプルタイミングで指定ビット数分の受信を行い、完了すると、ステータスの BUSY\_ が '1' になり、内部割り込みが発生し、受信動作を完了したことを示します。そしてクロック同期シリアル I/O は待機状態になります。この状態で、受信データをデータバスから読み出すことができます。

### 送信動作

図 10-2にあるように、待機状態、すなわちモード設定後、ステータスの BUSY\_ が '1' のときに送信データを書き込みます。送信データの書き込みは、この待機状態においてのみ受け付けられます。

次に送信イネーブルコマンドを書き込むと送信動作を開始し、ステータスの BUSY\_ は '0' になります。また、シリアルクロックとして内部クロックを選択している場合は、SCK 出力端子にシリアルクロックが出力開始されます。モード設定時に指定したシフトタイミングで指定ビット数だけ送信し、送信動作が完了すると、ステータスの BUSY\_ が '1' になり、内部割り込みが発生し、クロック同期シリアル I/O は待機状態とな

ります。

なお、受信完了後に送信イネーブルコマンドを書き込むことにより、受信データをそのまま送信することもできます。

### 送受信動作の注意事項

(1) 8 ビットのキャラクタ長を指定した場合には、データの書き込み/読み出しがシフトレジスタの下位 8 ビットに固定されています。誤って2回書き込むと、1 回目のデータが失われ、2 回目のデータが有効となります。

キャラクタ長として 9 ビット以上を指定した場合、データの書き込み/読み出し動作は下位側 8 ビットと上位側の 2 回に分けて行います。どちらの場合も下位側、上位側の順に行いますが、下位側か上位側かを記憶する素子が書き込み/読み出しで共通化されていますので、注意が必要です。例えば、送信データの低位側を書き込んだ後、データ読み出しを行うと、上位側が読み出されてしまいます。この記憶素子はリセット、モード/コマンド書き込み、ステータス読み出しによってクリア（下位側を選択）されますから、送信データの設定や受信データの読み出しを行う前に、ステータス読み出しを行うことをお奨めします。

またキャラクタ長として9ビット以上15ビットまでを指定した場合、読み出しデータの上位側 8 ビットのうち上から数ビットの値は不定となります。例えばキャラクタ長10ビットを指定した場合、読み出しデータの上位側8ビットのうちD7～D2までは不定となります。

(2) クロック同期シリアルI/Oのシフトレジスタは1本ですので、送受信同時には行えません（半二重通信）。また、バッファレジスタをもたないので、送信データは待機中にしか書き込めません。

(3) 送信または受信動作中に受信データの読み出しを行っても、そのデータ内容は保証されません。

(4) 送信動作後、シフトレジスタの内容は壊れますので、前回と同一データを送信する場合も再度、データ設定を行ってから送信して下さい。

## 10.7 動作タイミング

### 内部シリアルクロック選択時

内部シリアルクロック、12ビットキャラクタ長、SST=0、LSB先頭のモード設定を行った場合を例にとり説明します(図10-3, 10-4)。まず、送信の場合です。送信データを2回のデータ書き込みにより設定した後、送信イネーブルを書き込みます。ここで、ステータスのBUSY\_が'0'になります。また、モード設定により“H”になっていたSCK出力端子が“L”になると同時に、最下位ビットがTXSに現れます。同様にシリアルクロックが“H”から“L”になる度にシフトレジスタが1ビットシフトされ、TXSに送信データが現れます。12個のデータをTXSに出力した後、シリアルクロックは“H”になり、次の通信開始まで“H”を保持します。送信完了でステータスのBUSY\_は'1'になり、内部割り込みが発生し、クロック同期シリアルI/Oは待機状態になります。

受信の場合も基本的に同様です。受信データサンプルタイミングはシリアルクロックの“L”から“H”への変化時となり、内部シフトレジスタに取り込まれます。

### 外部シリアルクロック選択時

次に外部シリアルクロック、8ビットキャラクタ長、SST=1、MSB先頭のモード設定を行った場合を例にとります(図10-5, 10-6)。まず、送信の場合です。あらかじめ、SCK入力端子に“L”を入力しておきます。送信データを1回のデータ書き込みにより設定した後、送信イネーブルを書き込みます。ここで、ステータスのBUSY\_が'0'になります。この後、SCK入力端子にシリアルクロックを入力します。

入力するシリアルクロックの立ち上がりに同期して、シフトレジスタが1ビットシフトされ、TXS端子へのデータが、MSBから順に出力されます。8個のデータをTXS端子に出力した後、ステータスのBUSY\_が'1'になり、内部割り込みが発生し、クロック同期シリアルI/Oは待機状態になります。SCK入力端子に入力するシリアルクロックは8回の立ち下がりの後、もしくは、ステータスのBUSY\_が'1'になったのを受けて、“L”に保持します。

受信の場合は、クロック同期シリアルI/Oが待機状態(SCK入力端子は“L”にしておきます)で受信イネーブル

を書き込むと、ステータスのBUSY\_が'0'になります。これを受けて、SCK入力端子にシリアルクロックを入力します。入力するシリアルクロックの立ち下がりをサンプルタイミングとして、RXS端子のデータを1ビットシフトしながら、内部シフトレジスタに取り込みます。8個のデータを取り込むと、ステータスのBUSY\_が'1'になり、内部割り込みが発生し、クロック同期シリアルI/Oは待機状態になります。SCK入力端子に入力するシリアルクロックは8回の立ち上がりの後、もしくは、ステータスのBUSY\_が'1'になったのを受けて“L”に保持します。

### 動作タイミングの注意事項

(1)外部シリアルクロック選択時に必要パルス数を越えて、入力されるシリアルクロックは無視されます(図10-7)。

(2)外部シリアルクロック選択時、ステータスのBUSY\_が'1'の場合、すなわち待機中に入力される外部シリアルクロックは無視されます(図10-8)。

(3)外部シリアルクロック通信時において、ステータスBUSY\_が'0'になったことをステータス読み出しで確認した後、シリアルクロックをスタートさせて下さい。

## 10.8 リセット

RESET\_端子を“L”レベルにすると、外部シリアルクロック、8ビットキャラクタ長、SST=0、MSB先頭のモードになり、待機状態に入ります。

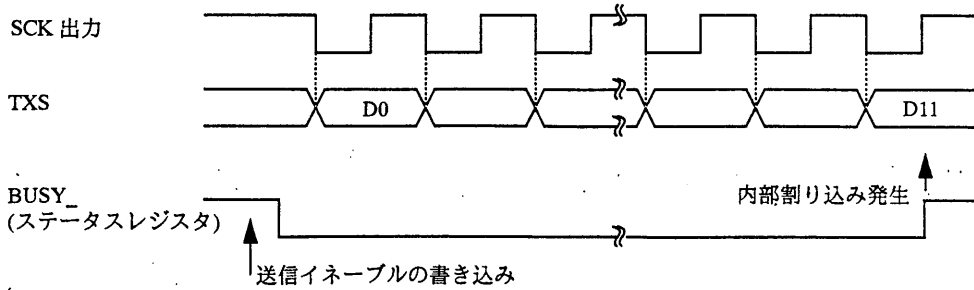


図 10-3 内部シリアルクロック、12ビットキャラクタ長、SST=0、LSB先頭時の送信動作タイミング

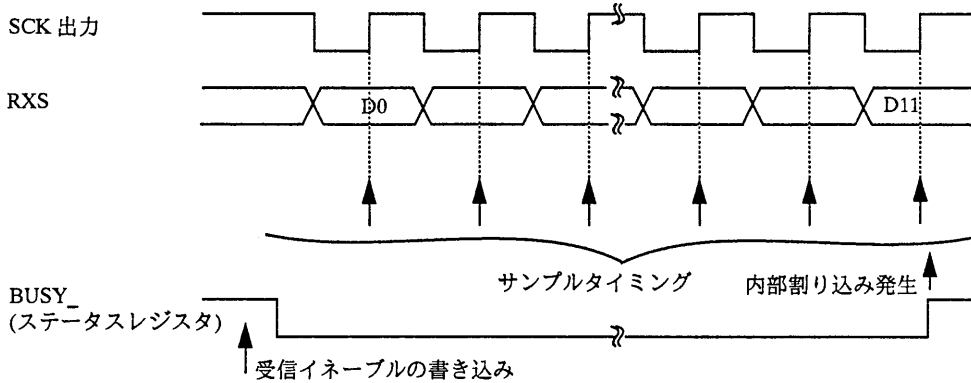


図 10-4 内部シリアルクロック、12ビットキャラクタ長、SST=0、LSB先頭時の受信動作タイミング

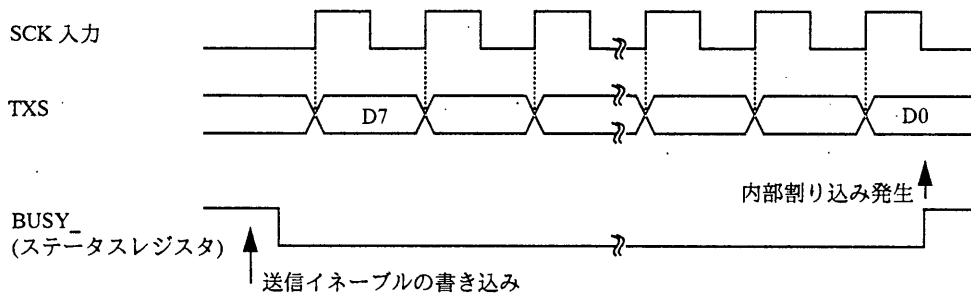


図 10-5 外部シリアルクロック、8ビットキャラクタ長、SST=1、MSB先頭時の送信動作タイミング

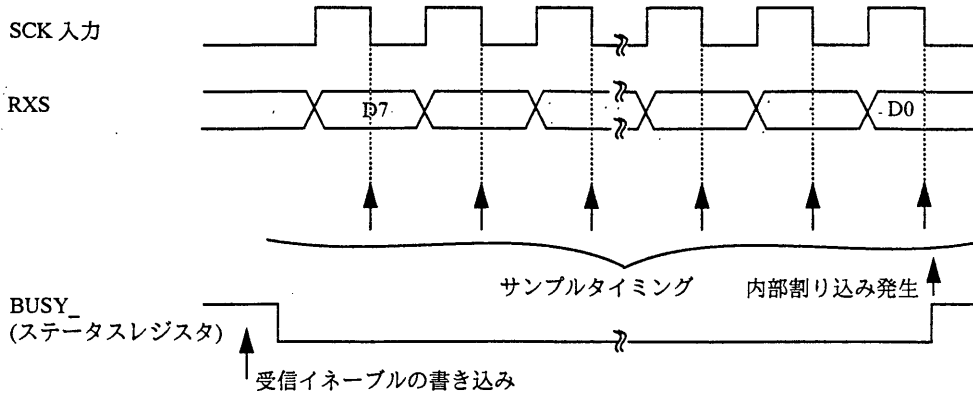


図 10-6 外部シリアルクロック、8ビットキャラクタ長、SST=1、MSB先頭時の受信動作タイミング

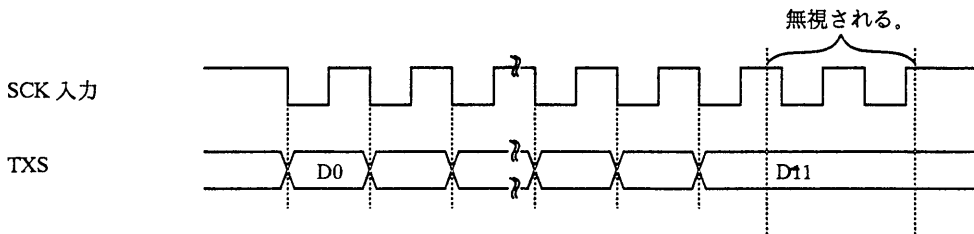


図 10-7 必要以上に入力される外部シリアルクロックの例 (送信時)

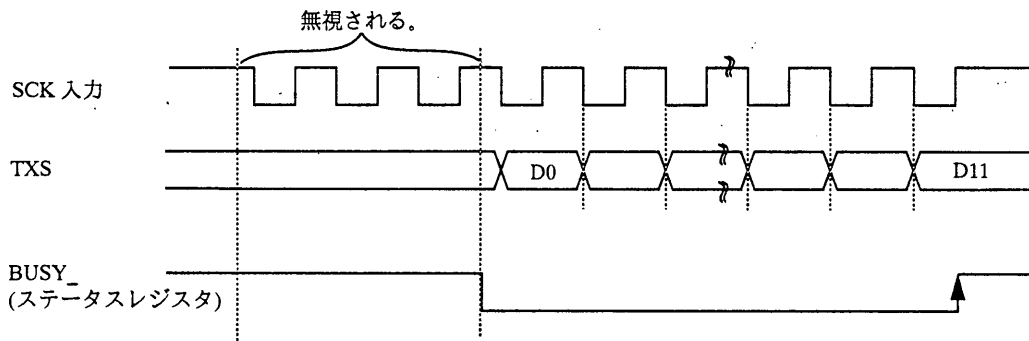


図 10-8 待機中に入力される外部シリアルクロックの例 (送信時)

### 11. パラレルポート

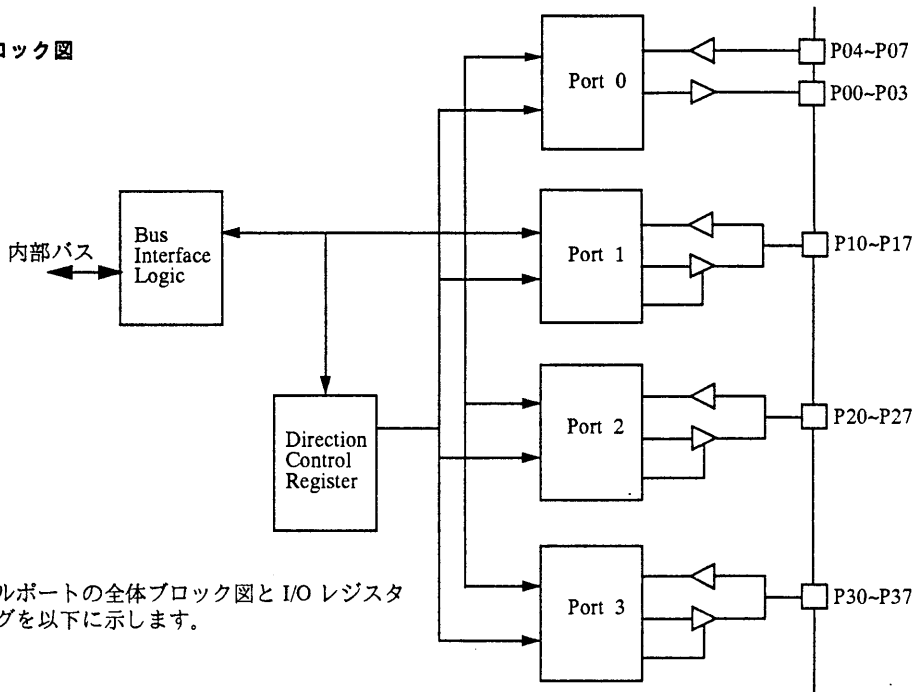
#### 11.1 概要

KL5C80A16 のパラレルポートは KP67 マクロセルで構成されています。ポート数は 32 本でポート 0 以外のポートは、1 ビット単位に入力/出力の方向制御が可能で、この設定はプログラムによるレジスタ設定で行なわれます。またプログラムによるビット単位のセット/リセット機能を用意しています。

#### 特徴

- ・ 32本のパラレルポート搭載
- ・ ビット単位の入力/出力の方向制御が可能
- ・ ソフトウェアによりビット単位のセット/リセットが可能

#### 11.2 ブロック図



パラレルポートの全体ブロック図と I/O レジスタマッピングを以下に示します。

表 11-1 I/O レジスタマッピング

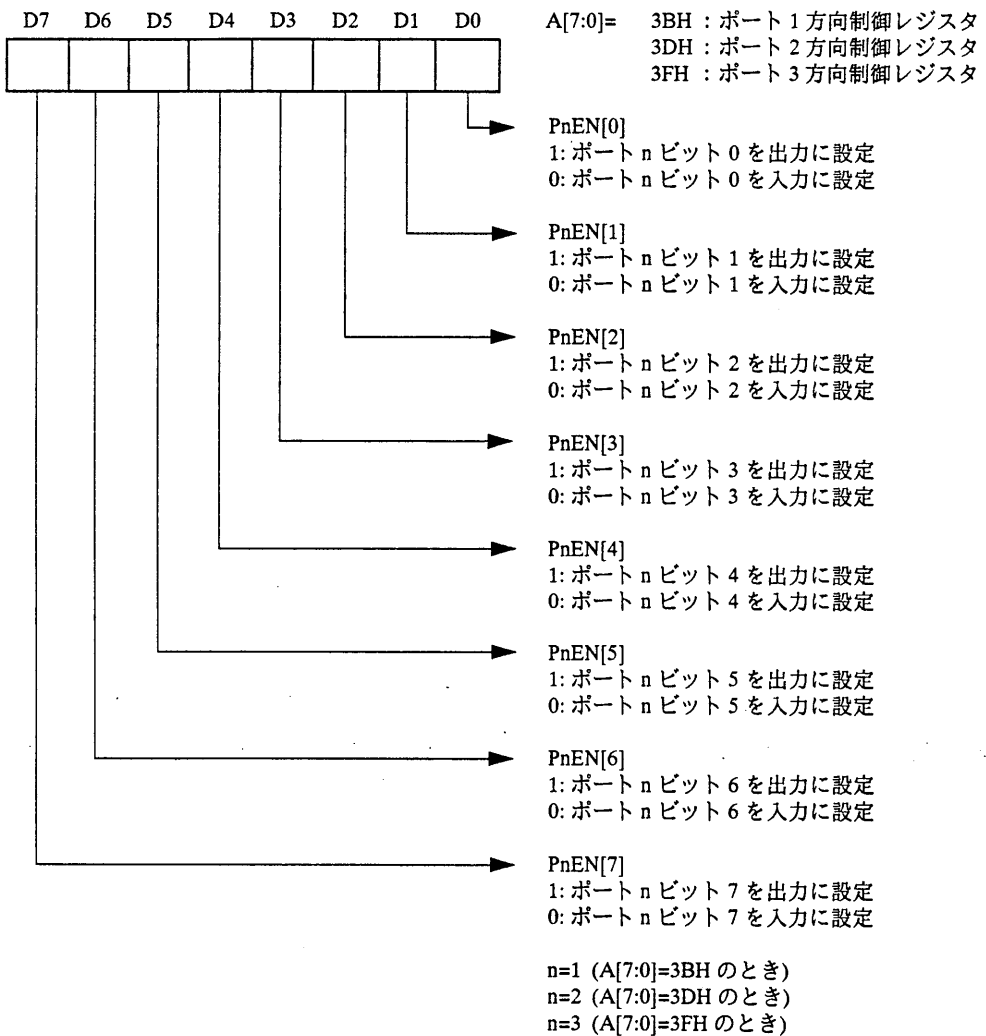
I/O	ライト時	リード時
38H	ポート0	ポート0
39H	ビット操作コマンド	ポート0方向制御レジスタ(固定)
3AH	ポート1	ポート1
3BH	ポート1方向制御レジスタ	ポート1方向制御レジスタ
3CH	ポート2	ポート2
3DH	ポート2方向制御レジスタ	ポート2方向制御レジスタ
3EH	ポート3	ポート3
3FH	ポート3方向制御レジスタ	ポート3方向制御レジスタ

### 11.3 ポート入出力設定

ポート0は上位4本が入力、下位4本が出力に固定されています。ポート1、2、3のポート入出力方向設定はI/Oアドレス=3BH, 3DH, 3FHの各ポート方向制御レジスタ書き込みで行います。ポート1、2、3はパ

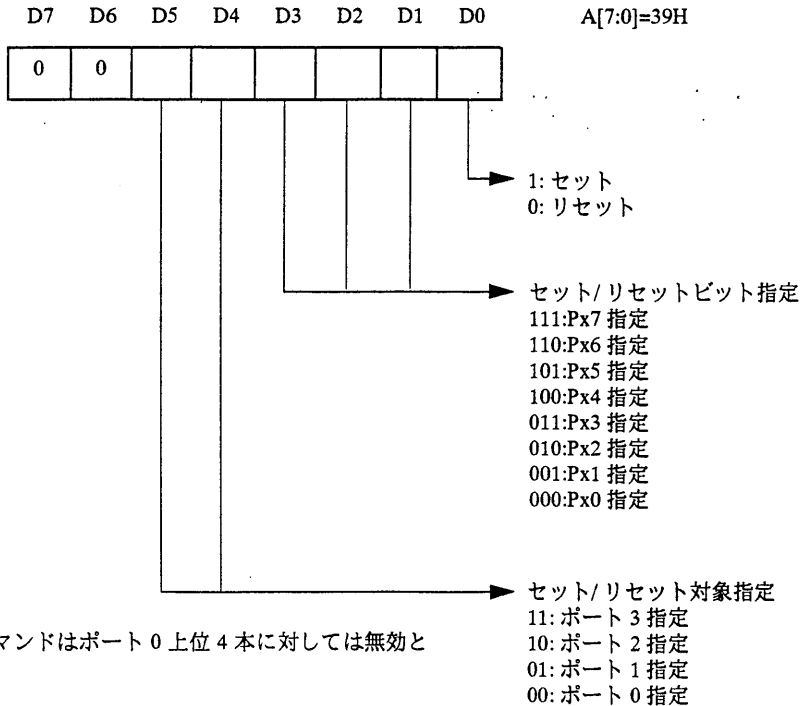
ラレルポートの入出力の方向設定を各ポートの方向制御レジスタにより1ビット単位で設定できます。

#### ポート方向制御レジスタ



### ビット操作コマンド

パラレルポートは全出力ポートを1ビット単位でセット/リセットできます。



このコマンドはポート 0 上位 4 本に対しては無効となります。

#### 11.4 ポート入出力動作

##### ポート入力動作

入力にモード設定されたポートでは双方向バッファから取り込んだデータをデータバスに出力します。

##### ポート出力動作

出力に設定されたポートではデータバスから各ポートに書き込まれたデータが双方向バッファから出力されます。

#### 11.5 リセット

リセット時には以下のレジスタがリセットされます。

- ポート 3-0 全ビット
- ポート 1 方向制御レジスタ
- ポート 2 方向制御レジスタ
- ポート 3 方向制御レジスタ

#### 11.6 使用上の注意

他の周辺回路の信号とパラレルポートがマルチプレクスされている端子をパラレルポートとして使用しない場合、そのパラレルポートは入力方向に設定して下さい。

### 12. 動作モード設定

本チップの動作モードは外部入力端子 BFMOD およびシステムコントロールレジスタ SCR により指定されます。

#### 12.1 BFMOD によるモード設定

BFMOD = 0 : ノーマルモード

1 : バグ・ファインダ Boot-on-RAMモード

##### ノーマルモード

KL5C80A16 をご使用になる時、通常はこのノーマルモードで使用して下さい。➡

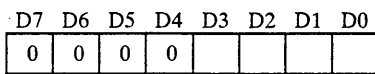
##### バグ・ファインダ Boot-on-RAMモード

バグ・ファインダ Boot-on-RAMモードでは、バグ・ファインダ・アダプタを BFSIO 端子に接続することでバグ・ファインダが外部 RAM 上で立ち上がります。詳しくはバグ・ファインダのマニュアルを参照して下さい。

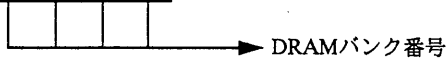
#### 12.2 システムコントロールレジスタ

KL5C80A16 の動作モード設定のためのシステムコントロールレジスタは、SCR0～SCR4の5種類あります。以下、それぞれについて説明します。

SCR0 (I/Oアドレス = '1BH')



ハードウェアリセットにより全ビット'0'となります。



DRAMバンク番号

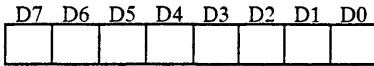
80000H～BFFFFHにDRAMを割り付けてアドレス拡張を行う場合、ここにバンク番号を設定します。

拡張するバンク数に応じて、LSB から必要なビット数 (D<1:0>またはD<3:0>)を使用し、0から最大15までのバンク番号を設定して下さい。例えば、

1M バイト拡張時: 4M ビットの × 4 DRAM を 2 個使用して、D<1:0> の 2 ビットで 4 バンク

4M バイト拡張時: 16M ビットの × 4 DRAM を 2 個使用して、D<3:0> の 4 ビットで 16 バンクのバンク番号を設定します。

SCR1(I/O アドレス = 1CH)



ハードウェアリセットにより全ビット'0'となります。

端子 89 から入力される信号を、D4 または D6 で割り込みコントローラの IR 入力に接続する場合、その信号を

- 0: 反転せず入力します。
- 1: 反転して入力します。

端子 88 から入力される信号を、D5 または D7 で割り込みコントローラの IR 入力に接続する場合、その信号を

- 0: 反転せず入力します。
- 1: 反転して入力します。

端子 87 から入力される信号を、割り込みコントローラの IR[14] に

- 0: 反転せず入力します。
- 1: 反転して入力します。

端子 86 から入力される信号を、割り込みコントローラの IR[15] に

- 0: 反転せず入力します。
- 1: 反転して入力します。

割り込みコントローラの IR[0] に

- 0: 端子 89 を接続します。
- 1: タイマのチャンネル 2 からの割り込み要求\*を接続します。

割り込みコントローラの IR[1] に

- 0: 端子 88 を接続します。
- 1: タイマのチャンネル 3 からの割り込み要求\*を接続します。

割り込みコントローラの IR[2] に

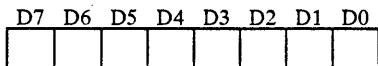
- 0: 端子 89 を接続します。
- 1: クロック同期シリアル I/O のチャンネル 1 の割り込み要求信号を接続します。

割り込みコントローラの IR[5] に

- 0: 端子 88 を接続します。
- 1: UART の BDET1+ERINT1 を接続します。

\*注) この時、割り込みコントローラの割り込み要求はエッジ認識に設定して下さい。

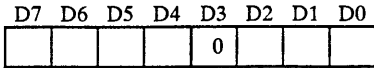
SCR2(I/O アドレス = 1DH)



ハードウェアリセットにより全ビット'0'となります。

- 00 : DREQ1 として端子 84 を使用します。  
端子 92 は P16<sub>1</sub> として機能します。
- 01 : DREQ1 として端子 84 を使用します。  
端子 92 は DACK1<sub>1</sub> として機能します。
- 10 : DREQ1 として UART の TXRDY0 を使用します。  
端子 92 は P16<sub>1</sub> として機能します。
- 11 : DREQ1 として UART の RXRDY0 を使用します。  
端子 92 は P16<sub>1</sub> として機能します。
  
- 0 : 端子 91 を P17 として使用します。  
1 : 端子 91 を EXBACK<sub>1</sub> として使用します。
  
- 0 : 端子 5 を PALAT として使用します。  
1 : 端子 5 を DACK0<sub>1</sub> として使用します。
  
- 00 : 端子 81 を P30  
端子 80 を P31  
端子 77 を P32  
UART の CTS1<sub>1</sub> = "L" (アクティブ)  
UART の DSR1<sub>1</sub> = "L" (アクティブ)  
として使用します。
- 01 : 端子 81 を P30  
端子 80 を P31  
端子 77 を UART の TXD1  
UART の CTS1<sub>1</sub> = "L" (アクティブ)  
UART の DSR1<sub>1</sub> = "L" (アクティブ)  
として使用します。
- 10 : 端子 81 を P30  
端子 80 を UART の RTS1<sub>1</sub>  
端子 77 を UART の TXD1  
UART の CTS1<sub>1</sub> を端子 74  
UART の DSR1<sub>1</sub> = "L" (アクティブ)  
として使用します。
- 11 : 端子 81 を UART の DTR1<sub>1</sub>  
端子 80 を UART の RTS1<sub>1</sub>  
端子 77 を UART の TXD1  
UART の CTS1<sub>1</sub> を端子 74  
UART の DSR1<sub>1</sub> を端子 75  
として使用します。
  
- 0 : 端子 94 を P14 として使用します。  
1 : 端子 94 を UART の RXRDY1 として使用します。
  
- 0 : 端子 93 を P15 として使用します。  
1 : 端子 93 を UART の TXRDY1 として使用します。

SCR3(I/O アドレス = 1EH)



ハードウェアリセットにより全ビット'0'となります。

Refresh Rate

D2	D1	D0	リフレッシュ要求発生間隔
0	0	0	128 clock 毎
0	0	1	88 clock 毎
0	1	0	64 clock 毎
0	1	1	44 clock 毎
1	0	0	32 clock 毎
1	0	1	22 clock 毎
1	1	0	16 clock 毎
1	1	1	11 clock 毎

D5='1' のとき、C0000H~FFFFFH を

0: SRAM 領域

1: DRAM 最上位バンク領域

として使用します。D5='0' のときは意味を持ちません。

80000H~BFFFFH を DRAM 領域として

0: 使用しません。

端子 96 は P13

端子 97 は P12

端子 98 は P11 として機能します。

1: 使用します。

端子 96 は UASEL<sub>-</sub>

端子 97 は CAS<sub>-</sub>

端子 98 は RAS<sub>-</sub> として機能します。

NMI<sub>-</sub> に

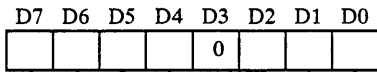
00: "H" (インアクティブ)

01: タイマのチャンネル 3 からの割り込み要求

10: 端子 82

11: (タイマのチャンネル 3 からの割り込み要求) + (端子 82) を接続します。

SCR4(I/O アドレス = 1FH)



ハードウェアリセットにより全ビット'0'となります。

- 00 : 端子 20 を P03  
端子 21 を P02  
端子 22 を P01 として使用します。
- 01 : 端子 20 を P03  
端子 21 を P02  
端子 22 をタイマの OUT1 として使用します。
- 10 : 端子 20 を P03  
端子 21 をタイマの OUT2  
端子 22 をタイマの OUT1 として使用します。
- 11 : 端子 20 をタイマの OUT3  
端子 21 をタイマの OUT2  
端子 22 をタイマの OUT1 として使用します。

- 0 : 端子 76 を P33  
端子 99 を P10 として使用します。
- 1 : 端子 76 をクロック同期シリアル I/O の TXS1  
端子 99 をクロック同期シリアル I/O の SCK1  
として使用します。

外部メモリ・ウェイト・コントロール(表 12-1参照)

外部 I/O・ウェイト・コントロール(表 12-2参照)

表 12-1 外部メモリ・ウェイト・コントロール

D5	D4	外部メモリ(00000H~7FFFFH)	外部メモリ(80000H~FFFFFFH)
0	X	1 wait	1 wait*
1	0	1 wait	0 wait*
1	1	0 wait	0 wait*

\* DRAMを使用する場合、挿入されるウェイト数はこの表には従いません。

表12-2 外部I/O・ウェイト・コントロール

D7	D6	外部I/O
0	0	1 wait (3 clock/バスサイクル)
0	1	2 wait (4 clock/バスサイクル)
1	0	3 wait (5 clock/バスサイクル)
1	1	4 wait (6 clock/バスサイクル)

4 wait時、EIORD\_/EIOWR\_の開始エッジが他の場合より1/2clock遅れます。

### 12.3 各端子の使い方

#### 端子 5 の使い方

端子 5 は出力固定です。この端子の機能は SCR2 によって設定します。

#### 端子 12,13,17,18 の使い方

端子 12, 13, 17, 18 は入力固定です。また、P07 と GATE3、P06 と GATE2、P05 と GATE1 と TRXC1、あるいは P04 と GATE0 と TRXC0 には同じ外部信号が入力されますので特にご注意下さい。いずれかの機能を無効にしたいときは各ブロックのモード設定により行なって下さい。

例えば端子 17 を TRXC1 として使用する場合は、タイマ/カウンタのチャンネル 1 をゲート機能なしで、また端子 18 を GATE0 として使用する場合には、UART のチャンネル 0 の通信クロックの生成には内蔵ポーレートジェネレータを使用して下さい。

#### 端子 20,21,22 の使い方

端子 20, 21, 22 は出力固定です。これらの端子の機能は SCR4 によって設定します。

#### 端子 72,73 の使い方

端子 72, 73 の入出力方向の設定は SCR の設定に関係なく、パラレルポートの方向制御レジスタによってのみ決定されます。したがって、端子 72, 73 を RXS1, RXD1 として使用する場合は P37, P36 を入力方向に設定して下さい。このとき、RXS1 と P37, RXD と P36 には同じ外部信号が入力されます。

また端子 72, 73 を出力ポートとして使用する場合は、パラレルポートの構造上その出力が RXS1, RXD1 に入力されることとなりますので、UART やクロック同期シリアル I/O のチャンネル 1 を受信用に使用する時は注意が必要です。この場合、これらのチャンネルは使用しないように各ブロックのモード設定を行なうことをお奨めします。

#### 端子 74,75 の使い方

端子 74, 75 の入出力方向の設定は、SCR の設定に関係なく、パラレルポートの方向制御レジスタによってのみ決定されます。したがって、端子 74,75 を CTS1\_、

DSR1\_ として使用する場合は P35, P34 を入力方向に設定し、かつ SCR2 の D<5:4> を '10' または '11' に設定して下さい。このとき、CTS1\_ と P35, DSR1\_ と P34 には同じ外部信号が入力されます。

また SCR2 の D<5:4> を '10' または '11' に設定し、かつ端子 74,75 を出力ポートとして使用する場合は、その出力が CTS1\_、DSR1\_ に入力されることとなりますので、UART のチャンネル 1 を使用する時は注意が必要です。

#### 端子 76 の使い方

端子 76 の入出力方向の設定は、SCR の設定に関係なく、パラレルポートの方向制御レジスタによってのみ決定されます。したがって、端子 76 を TXS1 として使用する場合は P33 を出力方向に設定し、かつ SCR4 の D2 を '1' にして下さい。このとき、その TXS1 出力は、パラレルポートの構造上 P33 にも入力されることとなります。すなわち、P33 を読み出すとその時の TXS1 が読めます。(11章参照。)

端子 76 をパラレルポートとして使用する場合はクロック同期シリアル I/O のチャンネル 1 を使用しないよう各ブロックのモード設定を行なうことをお奨めします。

#### 端子 77,80,81 の使い方

端子 77, 80, 81 の入出力方向の設定は、SCR2 の D<5:4> の設定により変化します。これらをパラレルポートとして使用する場合は、そのパラレルポートの方向制御レジスタにより方向を設定して下さい。

パラレルポート以外の機能で使用する場合は、入出力方向は自動的に出力に決定されます。

#### 端子 82 の使い方

端子 82 の入出力方向の設定は、SCR の設定に関係なく、パラレルポートの方向制御レジスタによってのみ決定されます。したがって、端子 82 を NMI\_ 信号入力端子として使用する場合は、P27 を入力方向に設定して下さい。このとき、SCR3 の D<7:6> が '10' に設定された場合、P27 と NMI\_ に同じ信号が入力されます。

端子 82 をパラレルポートとして使用する場合は、SCR3 の D7 を '0' にすることによって NMI\_ と端子 82 を切り離して使用して下さい。

**端子 83,84,85 の使い方**

端子83, 84, 85の入出力方向の設定は、SCRの設定に関係なく、パラレルポートの方向制御レジスタによってのみ決定されます。したがって、端子83, 84, 85をパラレルポート以外の機能で使用する場合は P26 ~ P24 の該当するポートを入力方向に設定して下さい。このときP26とEXBREQ\_、P24とDREQ0には同じ信号が入力されます。端子84に関しては、SCR2のD<1:0>が'00'または'01'に設定された場合、P25とDREQ1に同じ信号が入力されます。

これらの端子をパラレルポートとして使用する場合は、SCR2のD1を'1'にすることによってDREQ1と端子84を切り離し、またこれらのポートの信号がDMAコントローラの動作に影響を与えないようなモード設定をDMAコントローラの設定時に行なって下さい。

**端子 86,87,88,89 の使い方**

端子86, 87, 88, 89の入出力方向の設定は、SCRの設定に関係なく、パラレルポートの方向制御レジスタによってのみ決定されます。したがって、端子86, 87, 88, 89をパラレルポート以外の機能で使用する場合はそれらのP23~P20の該当するポートを入力方向に設定して下さい。このときP23とIR[15]、P22とIR[14]には同じ信号が入力されます。またSCR1のD7,D5の設定により端子88とIR[5]、IR[1]を接続し、またD6,D4の設定により端子89とIR[2]、IR[0]を接続した場合、P21とIR[5] / IR[1]あるいはP20とIR[2] / IR[0]には同じ信号が入力されます。さらにSCR1のD<3:0>により、各端子から入力される信号を反転して対応するIR入力に接続できます。

これらの端子をパラレルポートとして使用する場合、対応するIR入力は割り込みコントローラの設定によってマスクする、SCR1の設定によって各端子と切り離す、などの対策をされることをお奨めします。

**端子 91,92,93,94,96,97,98,99 の使い方**

端子91, 92, 93, 94, 96, 97, 98, 99の入出力方向の設定は、SCRの設定により変化します。これらの端子をパラレルポートとして使用する場合は、そのパラレルポートの方向制御レジスタにより方向を設定して下さい。

パラレルポート以外の機能で使用する場合は方向は自動的に出力に決定されます。なお、このとき端子99のみ、その方向はクロック同期シリアル I/O によって制御されますが、入力方向となっている時はクロック同期シリアル I/O の SCK1 と P10 に同じ信号が入力されます。

### 13. アドレス・マッピング

#### 内部I/Oマッピング

表 13-1 内部I/Oマッピング

I/Oアドレス	ブロック名	ライト時	リード時
00H	KC82(MMU)	BBR1 (境界/ベースレジスタ1)	BBR1 (境界/ベースレジスタ1)
01H		BR1 (ベースレジスタ1)	BR1 (ベースレジスタ1)
02H		BBR2 (境界/ベースレジスタ2)	BBR2 (境界/ベースレジスタ2)
03H		BR2 (ベースレジスタ2)	BR2 (ベースレジスタ2)
04H		BBR3 (境界/ベースレジスタ3)	BBR3 (境界/ベースレジスタ3)
05H		BR3 (ベースレジスタ3)	BR3 (ベースレジスタ3)
06H		BBR4 (境界/ベースレジスタ4)	BBR4 (境界/ベースレジスタ4)
07H		BR4 (ベースレジスタ4)	BR4 (ベースレジスタ4)
08H-0FH	川崎製鉄予約		
10H	DMAコントローラ	チャンネル0 B-PAR	チャンネル0 C-PAR
11H		チャンネル0 B-SAR	チャンネル0 C-SAR
12H		チャンネル0 B-BCR	チャンネル0 C-BCR
13H		チャンネル0 CR	チャンネル0 SR
14H		チャンネル1 B-PAR	チャンネル1 C-PAR
15H		チャンネル1 B-SAR	チャンネル1 C-SAR
16H		チャンネル1 B-BCR	チャンネル1 C-BCR
17H		チャンネル1 CR	チャンネル1 SR
18H	システム制御レジスタ	川崎製鉄予約	川崎製鉄予約
19H		川崎製鉄予約	川崎製鉄予約
1AH		川崎製鉄予約	川崎製鉄予約
1BH		SCR0	SCR0
1CH		SCR1	SCR1
1DH		SCR2	SCR2
1EH		SCR3	SCR3
1FH		SCR4	SCR4
20H	タイマ/カウンタ	チャンネル0カウンタ	チャンネル0カウンタ
21H		チャンネル0コントロールワード	チャンネル0ステータス
22H		チャンネル1カウンタ	チャンネル1カウンタ
23H		チャンネル1コントロールワード	チャンネル1ステータス
24H		チャンネル2カウンタ	チャンネル2カウンタ
25H		チャンネル2コントロールワード	チャンネル2ステータス
26H		チャンネル3カウンタ	チャンネル3カウンタ
27H		チャンネル3コントロールワード	チャンネル3ステータス

注) 内部 I/O アドレスのデコードは8ビットデコードでアドレス上位8ビットは無視してデコードしています。

表 13-1 内部I/Oマッピング(続き)

I/Oアドレス	ブロック名	ライト時	リード時
28H	UART	RATE	RATE
29H		川崎製鉄予約	川崎製鉄予約
2AH		チャンネル0送信データ	チャンネル0受信データ/拡張ステータスA
2BH		チャンネル0モード/コマンド	チャンネル0ステータス/拡張ステータスB
2CH		チャンネル1送信データ	チャンネル1受信データ/拡張ステータスA
2DH		チャンネル1モード/コマンド	チャンネル1ステータス/拡張ステータスB
2EH		川崎製鉄予約	川崎製鉄予約
2FH		川崎製鉄予約	川崎製鉄予約
30H	クロック同期	チャンネル0送信データ	チャンネル0受信データ
31H	シリアルI/O	チャンネル0コマンド/モード設定	チャンネル0ステータス
32H		チャンネル1送信データ	チャンネル1受信データ
33H		チャンネル1コマンド/モード設定	チャンネル1ステータス
34H	割り込みコントローラ	LERL/PGRL	ISRL
35H		LERH/PGRH	ISRH
36H		IMRL	IMRL
37H		IVR/IMRH	IMRH
38H	パラレルポート	ポート0	ポート0
39H		ビット操作コマンド	ポート0方向制御レジスタ(固定)
3AH		ポート1	ポート1
3BH		ポート1方向制御レジスタ	ポート1方向制御レジスタ
3CH		ポート2	ポート2
3DH		ポート2方向制御レジスタ	ポート2方向制御レジスタ
3EH		ポート3	ポート3
3FH		ポート3方向制御レジスタ	ポート3方向制御レジスタ

注) 内部 I/O アドレスのデコードは 8 ビットデコードでアドレス上位 8 ビットは無視してデコードしています。

### メモリ・マッピング

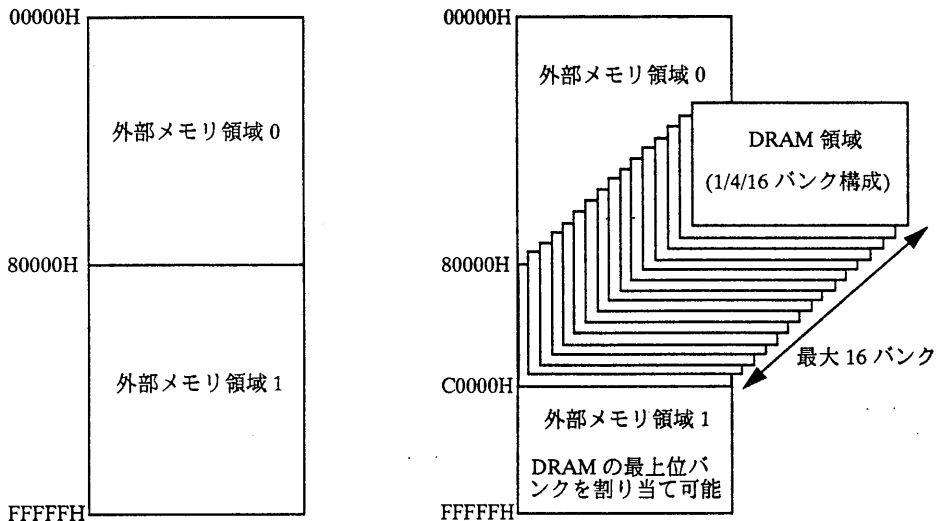
#### 内蔵DRAMコントローラを使用しない場合

外部メモリ領域1 物理アドレス空間 80000H～FFFFFH (512Kbyte)  
 外部メモリ領域0 物理アドレス空間 00000H～7FFFFH (512Kbyte)

#### 内蔵DRAMコントローラを使用する場合

外部メモリ領域1 物理アドレス空間 C0000H～FFFFFH (256Kbyte)  
 (DRAMの最上位バンクを割りあて可能)  
 DRAM領域 物理アドレス空間 80000H～BFFFFH (256Kbyte)  
 (外部メモリ領域1のウェイト設定とは無関係に3クロックバスサイクルになります。)  
 外部メモリ領域0 物理アドレス空間 00000H～7FFFFH (512Kbyte)

システムコントロールレジスタにより外部メモリ領域0と外部メモリ領域1ではそれぞれ別にウェイト設定が可能です。



DRAM を使用しない場合

DRAM を使用する場合

図 13-1 メモリマッピング

### 14. 発振回路

#### 14.1 概要

KL5C80A16 はシステムクロックを発生させるための発振バッファを搭載しています。チップ内部のシステムクロックはこの発振回路が発生させた信号を2分周した信号です。

#### 14.2 回路構成

システムクロックを発生させるためには KL5C80A16 の発振バッファ XIN, XOUT に外部部品として水晶振動子 (あるいはセラミック振動子)、フィードバック抵抗、制限抵抗、コンデンサを右図のように接続することで発振回路を構成できます。外部部品定数は使用する振動子、基板等によって異なります。外部部品定数の最適値は振動子メーカーの推奨値を御使用下さい。チップ内部のシステムクロックはこの発振回路が発生させた信号を2分周した信号です。分周回路はチップ内に搭載されています。

表 14-1 発振周波数

発振周波数	動作電源電圧
2 ~ 20 MHz	5V ± 5%, 5V ± 10%

#### 注意

外部へクロックを取り出す場合は CLK 端子から取り出して下さい。XIN, XOUT から直接信号を取り出さないでください。

外部からクロックを入力する場合は XIN からシステムクロックの2倍の周波数の信号を入力して下さい。このとき XOUT につく寄生容量はできるだけ小さくして下さい。

表 14-2 外部部品定数参考範囲

	Rd	Cl, CO
水晶	100 ~ 800 Ω	5 ~ 30pF
セラミック	30 ~ 300 Ω	5 ~ 100pF

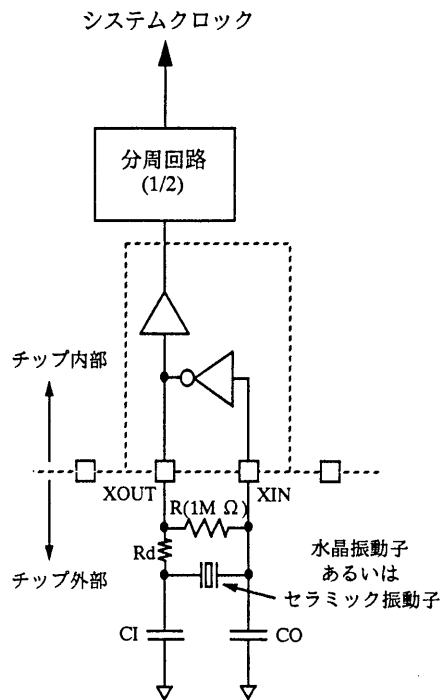


図 14-1 発振回路の構成

### 15. 電気的特性

#### 15.1 絶対最大定格

表 15-1 絶対最大定格 (GND 基準)

項目	記号	定格値	単位
電源電圧	VDD	-0.6 ~ +7.0	V
入力電圧	VIN	-0.6 ~ VDD + 0.6	V
保存温度	TSTG	-40 ~ +125	°C

#### 15.2 DC 特性 (5V ± 10%)

表 15-2 推奨動作条件

項目	記号	定格値	単位
電源電圧	VDD	4.5 ~ 5.5	V
動作周囲温度	TA	0 ~ +70	°C

表 15-3 電気的特性 (推奨動作条件での特性)

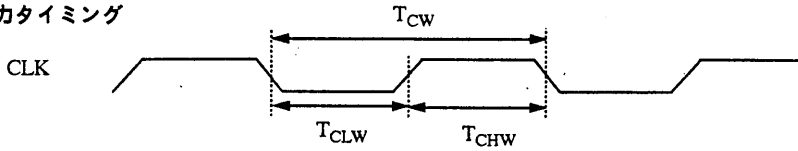
項目	記号	規格値			単位	測定条件
		最小	標準	最大		
入力電圧 (RESET_ 以外の入力端子)	V <sub>IH</sub>	3.6		VDD	V	
	V <sub>IL</sub>	GND		1.4	V	
RESET_ 入力端子 (シュミットトリガ入力)	V <sub>+</sub>	2.4		4.0	V	
	V <sub>-</sub>	0.9		2.3	V	
	V <sub>h</sub>	0.9			V	
出力電圧	V <sub>OH</sub>	3.5			V	I <sub>OH</sub> = -4mA or -6mA
	V <sub>OL</sub>			0.4	V	I <sub>OL</sub> = 4mA or 6mA
出力電流	I <sub>OUT</sub>			±6	mA	(注 1)
				±4	mA	(注 2)
入力リーク電流	I <sub>IL</sub>	-10			μA	V <sub>IN</sub> = GND
	I <sub>IH</sub>			10	μA	V <sub>IN</sub> = VDD
出力リーク電流	I <sub>OZ</sub>	-10		10	μA	ハイ・インピーダンス出力時
ブル・アップ電流	I <sub>PU</sub>	20	95	250	μA	V <sub>IN</sub> = GND
ブル・ダウン電流	I <sub>PD</sub>	20	95	250	μA	V <sub>IN</sub> = VDD
スタンバイ電流	I <sub>DDS</sub>		1.0*	100	μA	CLK 停止時
動作時消費電流	I <sub>DDOP</sub>		30*		mA	f (CLK) = 10MHz

\* TA = 25 °C のとき

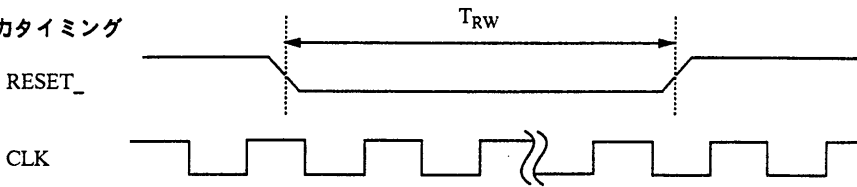
注 1) A[19:0], D[7:0], EMRD\_, EMWR\_, EIORD\_, EIOWR\_, CLK の各出力端子  
注 2) 上記以外の出力端子

### 16.3 AC 特性

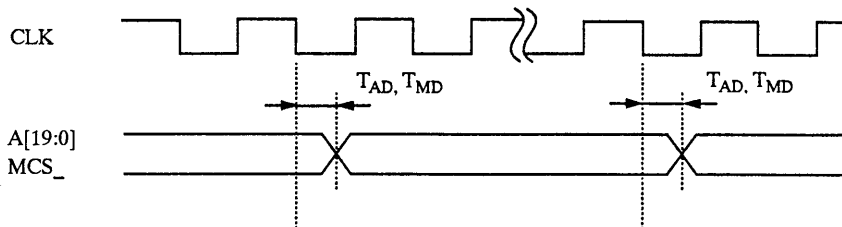
#### クロック出力タイミング



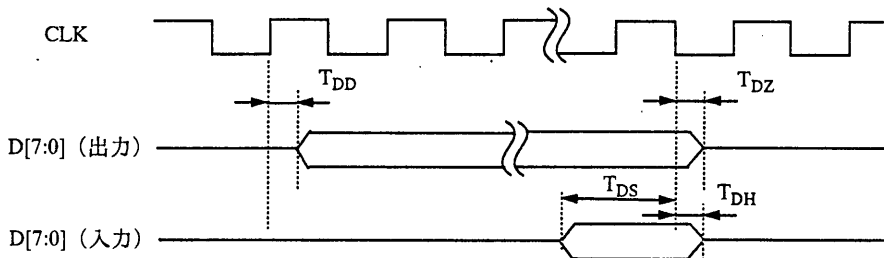
#### リセット入力タイミング



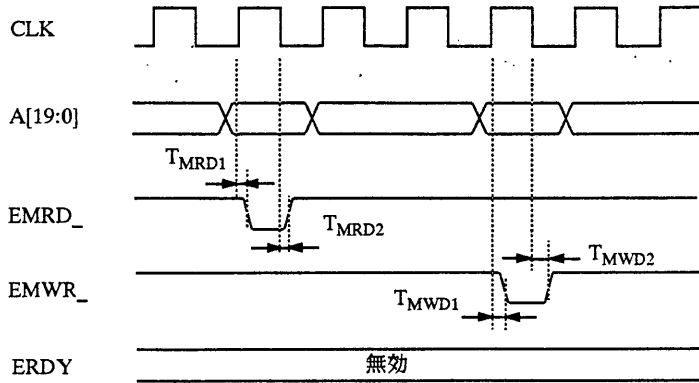
#### アドレス出力タイミング



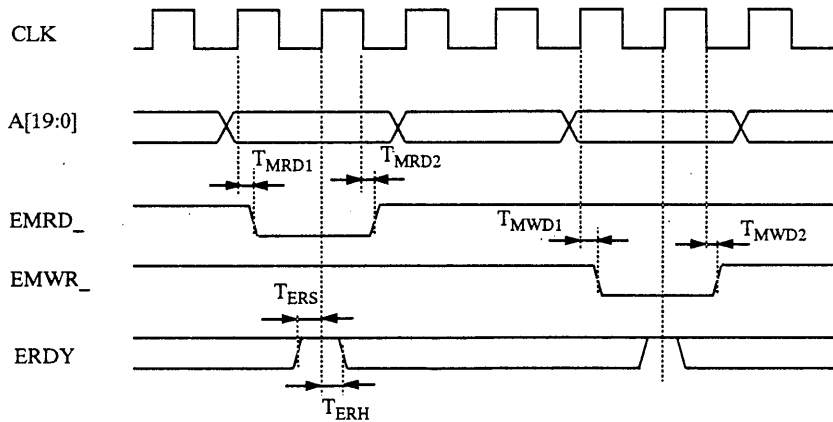
#### データ入出力タイミング



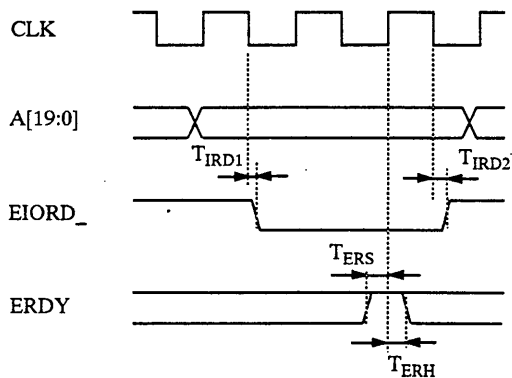
### 外部メモリアクセス (0 ウェイト)



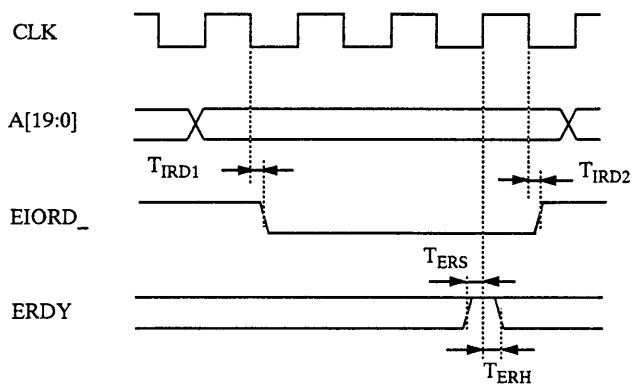
### 外部メモリアクセス (1 ウェイト)



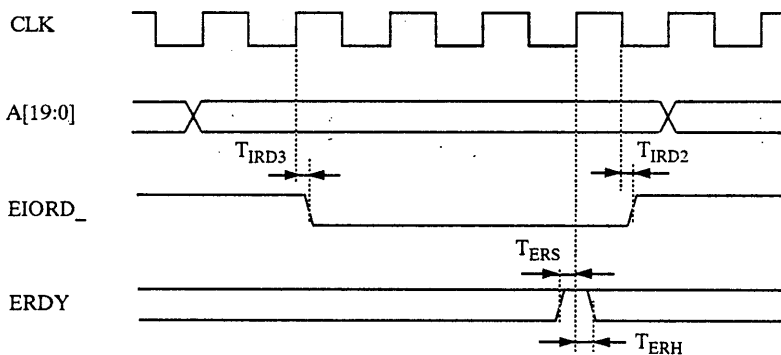
### 外部 I/O リード (1 ウェイト)



### 外部 I/O リード (2 ウェイト)

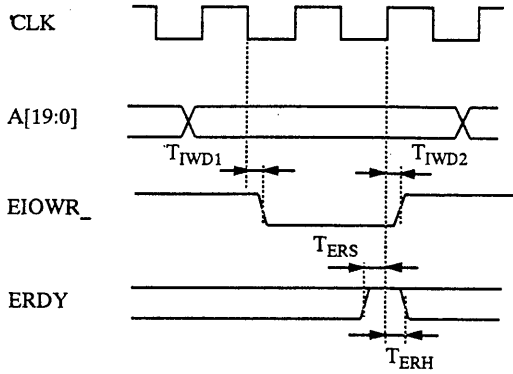


### 外部 I/O リード (4 ウェイト)

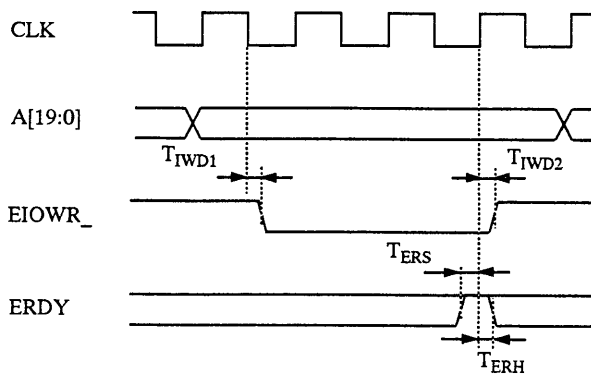


※ ウェイトコントロールレジスタ (SCR4) により 4 ウェイトを設定した場合、EIORD\_ の開始エッジが 1/2 クロック遅れます。

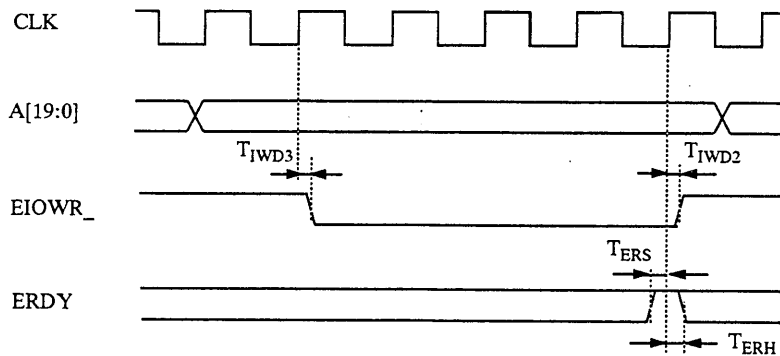
### 外部 I/O ライト (1 ウェイト)



### 外部 I/O ライト (2 ウェイト)

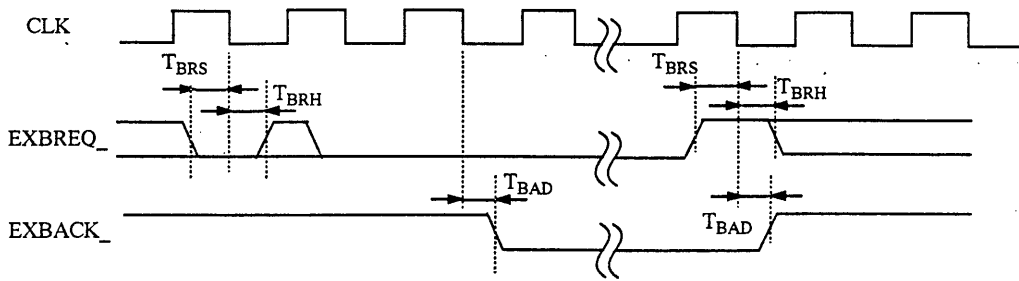


### 外部 I/O ライト (4 ウェイト)

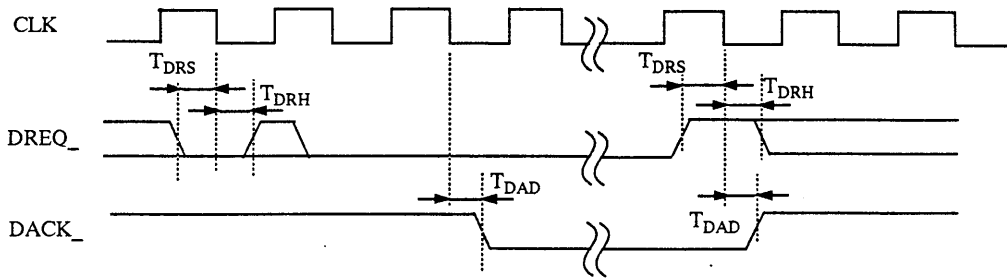


※ ウェイトコントロールレジスタ (SCR4) により 4 ウェイトを設定した場合、EIORW\_ の開始エッジが 1/2 クロック遅れます。

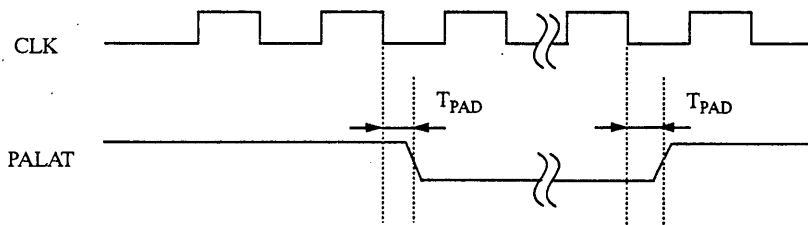
### バスコントロールタイミング



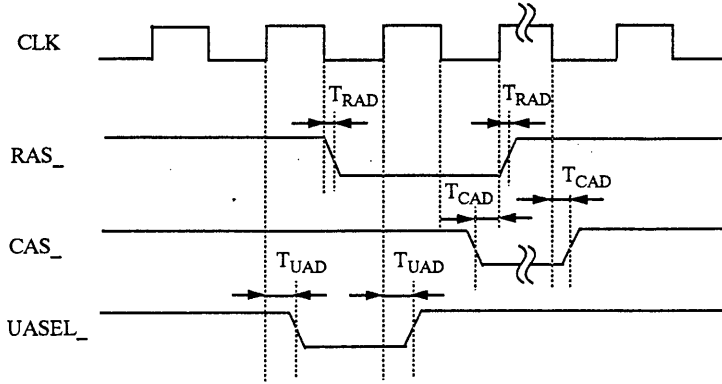
### DMA コントロールタイミング



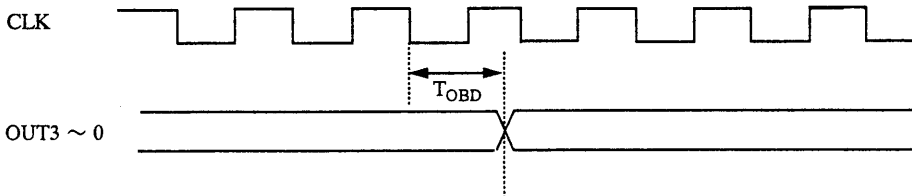
### PALAT 出力タイミング



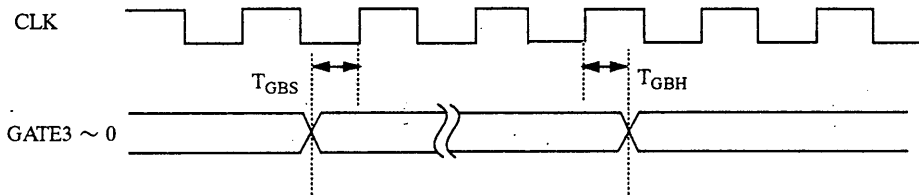
### DRAM コントロールタイミング



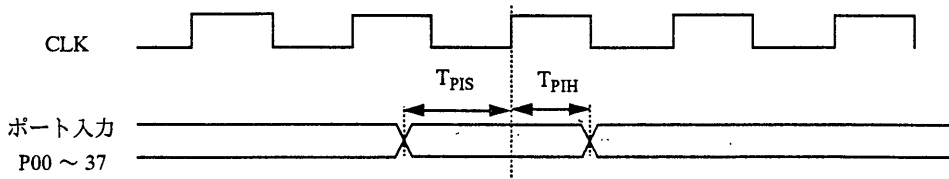
### カウンタ出力タイミング



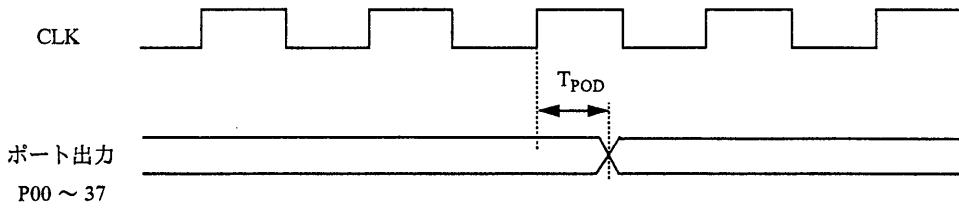
### カウンタゲートタイミング



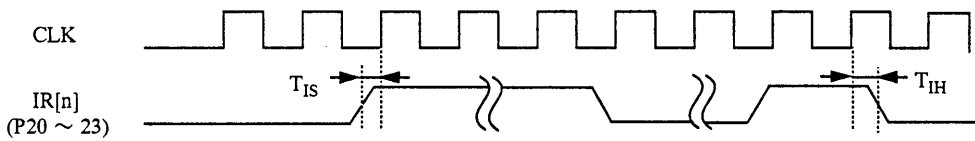
### ポート入力タイミング



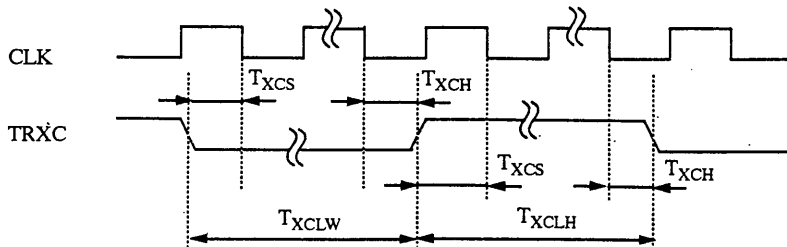
### ポート出力タイミング



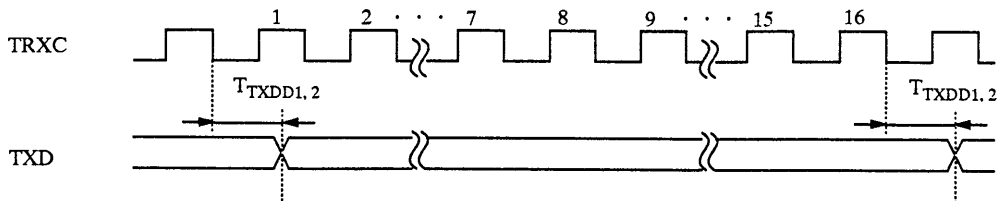
### 外部割り込み入力タイミング



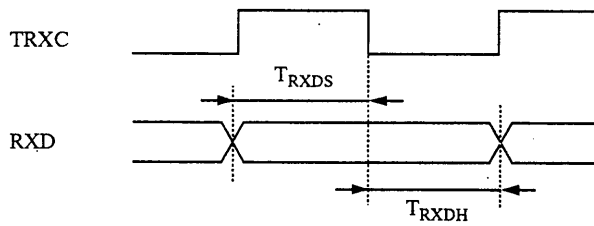
### UART 送受信クロック



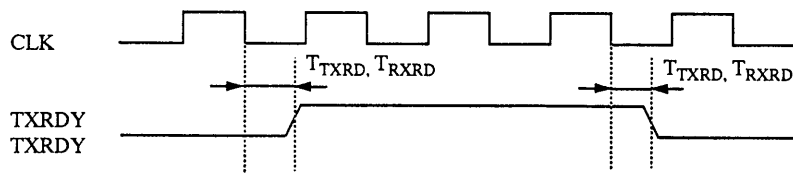
### UART 送信クロックと送信データ



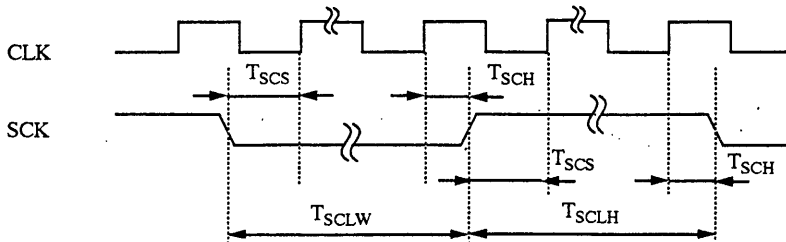
### UART 受信クロックと受信データ



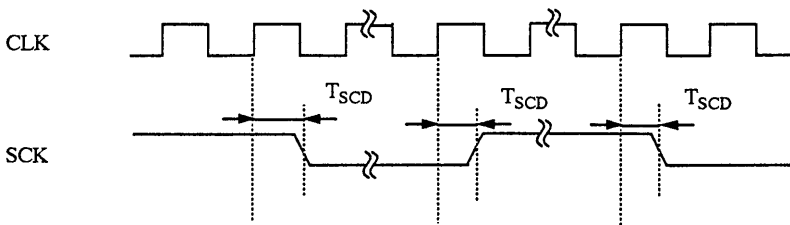
### TXRDY, RXRDY 出力タイミング



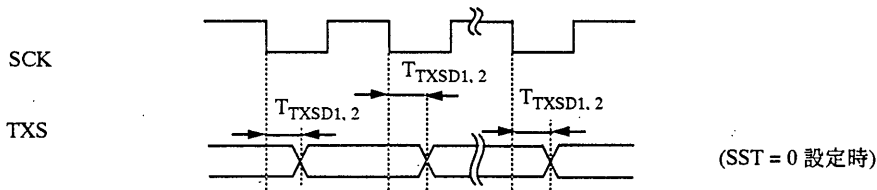
### シリアルクロック入力タイミング



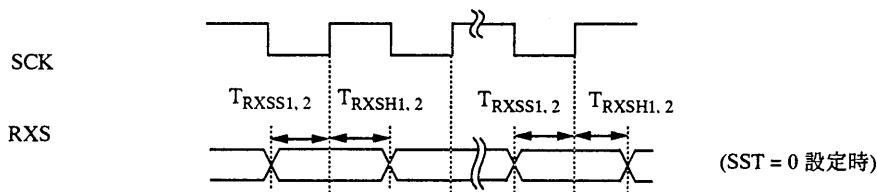
### シリアルクロック出力タイミング



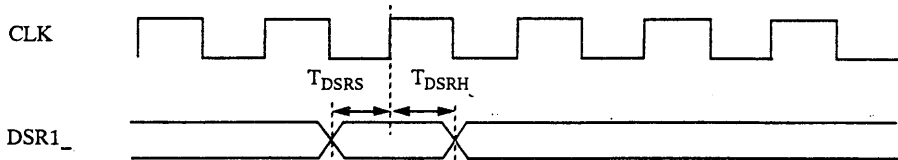
### クロック同期シリアル送信データタイミング



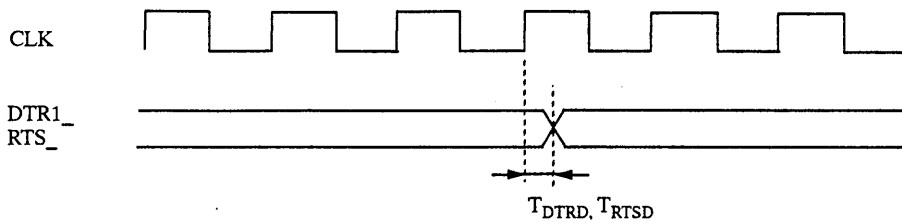
### クロック同期シリアル受信データタイミング



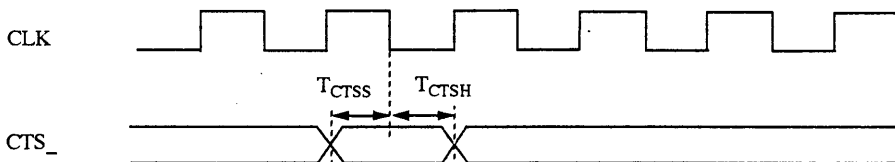
### DSR1\_ 入カタイミング



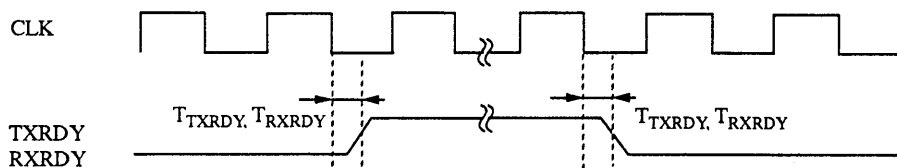
### DTR1\_, RTS\_ 出カタイミング



### CTS\_ 入カタイミング



### TXRDY, RXRDY 出カタイミング



### KL5C80A16 AC 特性 (5V ± 10%)

番号	項目	最小	標準	最大	単位
T <sub>CYC</sub>	XINサイクル時間	50.0			ns
T <sub>CW</sub>	CLKサイクル時間	100.0			ns
T <sub>CLW</sub>	CLK “L” パルス幅		50.0		ns
T <sub>CHW</sub>	CLK “H” パルス幅		50.0		ns
T <sub>RW</sub>	RESET_パルス幅	3			clk
T <sub>AD</sub>	アドレス出力遅延時間	10.0		49.0	ns
T <sub>MD</sub>	チップセレクト出力遅延時間			52.0	ns
T <sub>DD</sub>	CLK→データ出力遅延時間			32.0	ns
T <sub>DZ</sub>	CLK→データ出力off遅延時間	8.0			ns
T <sub>DS</sub>	データ入力セットアップ時間	3.0			ns
T <sub>DH</sub>	データ入力ホールド時間	5.0			ns
T <sub>MRD1</sub>	CLK立ち上がりエッジ→EMRD_ “L” 遅延時間			27.0	ns
T <sub>MRD2</sub>	CLK立ち下がりエッジ→EMRD_ “H” 遅延時間	7.0		26.0	ns
T <sub>MWD1</sub>	CLK立ち上がりエッジ→EMWR_ “L” 遅延時間			27.0	ns
T <sub>MWD2</sub>	CLK立ち下がりエッジ→EMWR_ “H” 遅延時間	7.0		27.0	ns
T <sub>ERS</sub>	ERDY入力セットアップ時間	3.0			ns
T <sub>ERH</sub>	ERDY入力ホールド時間	5.0			ns
T <sub>IRD1</sub>	CLK立ち下がりエッジ→EIORD_ “L” 遅延時間			30.0	ns
T <sub>IRD2</sub>	CLK立ち下がりエッジ→EIORD_ “H” 遅延時間	7.0		25.0	ns
T <sub>IRD3</sub>	CLK立ち上がりエッジ→EIORD_ “L” 遅延時間			27.0	ns
T <sub>IWD1</sub>	CLK立ち下がりエッジ→EIOWR_ “L” 遅延時間			31.0	ns
T <sub>IWD2</sub>	CLK立ち上がりエッジ→EIOWR_ “H” 遅延時間			24.0	ns
T <sub>IWD3</sub>	CLK立ち上がりエッジ→EIOWR_ “L” 遅延時間			28.0	ns
T <sub>BRS</sub>	EXBREQ_入力セットアップ時間	3.0			ns
T <sub>BRH</sub>	EXBREQ_入力ホールド時間	5.0			ns
T <sub>BAD</sub>	EXBACK_出力遅延時間			45.0	ns
T <sub>DRS</sub>	DREQ_入力セットアップ時間	3.0			ns
T <sub>DRH</sub>	DREQ_入力ホールド時間	5.0			ns
T <sub>DAD</sub>	DACK_出力遅延時間			45.0	ns
T <sub>PAD</sub>	PALAT出力遅延時間			40.0	ns
T <sub>RAD</sub>	RAS_出力遅延時間			31.0	ns
T <sub>CAD</sub>	CAS_出力遅延時間			31.0	ns
T <sub>UAD</sub>	UASEL_出力遅延時間			30.0	ns
T <sub>GS</sub>	GATE入力セットアップ時間	3.0			ns
T <sub>GH</sub>	GATE入力ホールド時間	5.0			ns

番号	項目	最小	標準	最大	単位
T <sub>OD</sub>	タイマ/カウンタOUT出力遅延時間			45.0	ns
T <sub>PIS</sub>	ポート入力セットアップ時間	5.0			ns
T <sub>PIH</sub>	ポート入力ホールド時間	5.0			ns
T <sub>POD</sub>	ポート出力遅延時間			40.0	ns
T <sub>IS</sub>	外部割り込み入力セットアップ時間	3.0			ns
T <sub>IH</sub>	外部割り込み入力ホールド時間	5.0			ns
T <sub>NIW</sub>	NMI_ “L” パルス幅	20.0			ns
T <sub>XCLW</sub>	TRXC “L” パルス幅	1			clk
T <sub>XCHW</sub>	TRXC “H” パルス幅	1			clk
T <sub>XCS</sub>	TRXC入力セットアップ時間	3.0			ns
T <sub>XCH</sub>	TRXC入力ホールド時間	5.0			ns
T <sub>TXDD1</sub>	TXD出力遅延時間(内部送受信クロック選択時)			40.0	ns
T <sub>TXDD2</sub>	TXD出力遅延時間(外部クロックTRXC選択時)			40.0	ns
T <sub>RXDS</sub>	RXD入力セットアップ時間	3.0			ns
T <sub>RXDH</sub>	RXD入力ホールド時間	5.0			ns
T <sub>DSRS</sub>	DSR1_入力セットアップ時間	5.0			ns
T <sub>DSRH</sub>	DSR1_入力ホールド時間	5.0			ns
T <sub>CTSS</sub>	CTS_入力セットアップ時間	3.0			ns
T <sub>CTSH</sub>	CTS_入力ホールド時間	5.0			ns
T <sub>DTRD</sub>	DTR1_出力遅延時間			40.0	ns
T <sub>RTSD</sub>	RTS_出力遅延時間			40.0	ns
T <sub>TXRD</sub>	TXRDY1出力遅延時間			40.0	ns
T <sub>RXR</sub>	RXRDY1出力遅延時間			40.0	ns
T <sub>SCLW</sub>	SCK “L” パルス幅	1			clk
T <sub>SCHW</sub>	SCK “H” パルス幅	1			clk
T <sub>SCS</sub>	SCK入力セットアップ時間	3.0			ns
T <sub>SCH</sub>	SCK入力ホールド時間	5.0			ns
T <sub>SCD</sub>	SCK出力遅延時間			40.0	ns
T <sub>TXSD1</sub>	TXS出力遅延時間(内部送受信クロック選択時)			40.0	ns
T <sub>TXSD2</sub>	TXS出力遅延時間(外部クロックSCK選択時)			40.0	ns
T <sub>RXSS1</sub>	RXS入力セットアップ時間(内部クロック選択時)	3.0			ns
T <sub>RXSH1</sub>	RXS入力ホールド時間(内部クロック選択時)	5.0			ns
T <sub>RXSS2</sub>	RXS入力セットアップ時間(外部SCK選択時)	3.0			ns
T <sub>RXSH2</sub>	RXS入力ホールド時間(外部SCK選択時)	5.0			ns

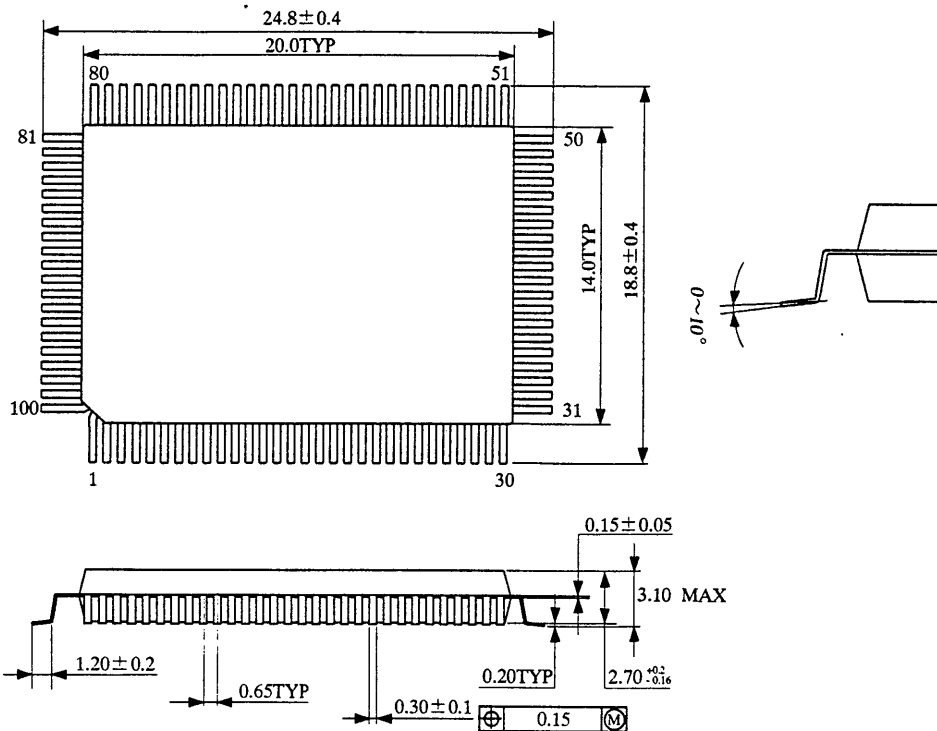
注1) 出力端子の負荷容量C<sub>L</sub>は70pFで測定しています。

注2) 単位の欄に clk とあるのはシステムクロック数を示しています。

### 16 外形寸法図

KL5C80A16 はプラスチック QFP100 パッケージに封止されています。下に QFP100 の外形寸法図を掲載します。

尚、パッケージ上には製品コードとして、「KL5C80A16CF」とマーキングされております。



## 付録 A インストラクション表

この表は、命令の種類ごとに分類して、KC82 のインストラクションをまとめたものです。表中の“ニーモニック”は各命令の名称を現わし、“命令動作”でその動作内容を簡潔に示しています。

命令動作の欄で“<”とあるのは、右側の内容を左側に代入することを、カッコでくくられたものは、くくられたレジスタなどで示されるメモリの内容を、それぞれ意味しています。たとえば

r<-[HL]

とあるのは、HLレジスタで示されるアドレスのメモリの内容を、8ビットレジスタに代入するということです。ただし入出力命令の[n]と[C]は、対応する入出力ポートのデータを意味しています。

“フラグ”の欄は各フラグの動作を、“オペコード”はそれぞれの命令に対するマシン語コードを、2進数と16進数で記したものの。その右側の“B”“CL”は、各命令の長さ（バイト数）と、命令を実行するのに要するクロック数を、それぞれ表わしています。

このほか、インストラクション表に出てくる略号に関して、次の凡例にまとめておいたので参考にしてください。

A{7}	レジスタ A の最上位ビット
A{4..7}	レジスタ A のビット 4-7
;	動作の区切り
[IX+d]	IX に 8 ビットの符号つき変位を足した値が示すアドレスのデータ
C	キャリーフラグ
Z	ゼロフラグ
PV	パリティ・オーバーフローフラグ
S	サインフラグ
N	減算フラグ
H	ハーフキャリーフラグ
●	フラグは変化しない
⇔	フラグは実行結果により変化する
0	フラグは 0
1	フラグは 1
?	不定になる
V	オーバーフローフラグとして使われる
P	パリティフラグとして使われる
IF	割り込みフリップフロップの値が入る
r, r'	8 ビットレジスタ、A, B, C, D, E, H, L
ss	16 ビットレジスタ、BC, DE, HL, SP
pp	16 ビットレジスタ、BC, DE, IX, SP
rr	16 ビットレジスタ、BC, DE, IY, SP

qq	16ビットレジスタ、BC, DE, HL, AF
e	JR系の命令の飛び先アドレスへの差分、8ビットの符号つき即値 (+127~-128)
k	RST命令の飛び先アドレス、00h, 08h, 10h, 18h, 20h, 28h, 30h, 38h
nn	16ビットの即値、もしくは絶対アドレス
n	8ビットの即値
b	ビット演算命令の第何ビットかを示す値
NOT	ビットを反転する
∨	ビットのORをとる
⊕	ビットのXORをとる
∧	ビットのANDをとる
tmp	一時的に値を待避する
B	命令のバイト数
CL	命令の実行に必要な最小クロック数

分岐命令、コール命令でクロック数がふたつ書いてあるものは、上が条件が成立しないとき、下が条件が成立したときを意味します。

また、入出力命令でクロック数がふたつ書いてあるものは、上がまだ転送が終わらないとき、下が転送が終わったときをそれぞれ意味しています。

#### 8 BIT LOAD GROUP

Mnemonic	Operation	F l a g s S Z H P V N C	Opcode		B	CL
			76543210	Hex		
LD r, r'	r<-r'	● ● ● ● ● ● ● ●	01 r r'		1	1
LD r, n	r<-n	● ● ● ● ● ● ● ●	00 r 110 <- n ->		2	2
LD r, [HL]	r<-[HL]	● ● ● ● ● ● ● ●	01 r 110		1	2
LD r, [IX+d]	r<-[IX+d]	● ● ● ● ● ● ● ●	11011101 01 r 110 <- d ->	DD	3	5
LD r, [IY+d]	r<-[IY+d]	● ● ● ● ● ● ● ●	11111101 01 r 110 <- d ->	FD	3	5
LD [HL], r	[HL]<-r	● ● ● ● ● ● ● ●	01110 r		1	2
LD [IX+d], r	[IX+d]<-r	● ● ● ● ● ● ● ●	11011101 01110 r <- d ->	DD	3	5
LD [IY+d], r	[IY+d]<-r	● ● ● ● ● ● ● ●	11111101 01110 r <- d ->	FD	3	5
LD [HL], n	[HL]<-n	● ● ● ● ● ● ● ●	00110110 <- n ->	36	2	3
LD [IX+d], n	[IX+d]<-n	● ● ● ● ● ● ● ●	1011101	DD	4	5

Mnemonic	Operation	F l a g s S Z H P V N C	O p c o d e		B	CL
			76543210	Hex		
LD [IY+d], n	[IY+d]<-n	● ● ● ● ● ●	00110110	36	4	5
			<- d ->			
			<- n ->			
			11111101	FD		
LD A, I	A<-I	⇒ ⇒ 0 IF 0 ● *1	00110110	36	2	2
			<- d ->			
LD A, R	A<-R	⇒ ⇒ 0 IF 0 ● *1	11101101	ED	2	2
			<- n ->			
LD I, A	I<-A	● ● ● ● ● ●	01010111	57	2	2
			11101101	ED		
LD R, A	R<-A	● ● ● ● ● ●	01011111	5F	2	2
			11101101	ED		
LD A, [BC]	A<-[BC]	● ● ● ● ● ●	01000111	47	1	3
			11101101	ED		
LD A, [DE]	A<-[DE]	● ● ● ● ● ●	01001111	4F	1	3
			00001010	0A		
LD A, [nn]	A<-[nn]	● ● ● ● ● ●	00011010	1A	1	3
			00111010	3A		
LD [BC], A	[BC]<-A	● ● ● ● ● ●	<- nnl ->		1	3
			<- nnh ->			
LD [DE], A	[DE]<-A	● ● ● ● ● ●	00000010	02	1	3
			00010010	12		
LD [nn], A	[nn]<-A	● ● ● ● ● ●	00110010	32	3	4
			<- nnl ->			
			<- nnh ->			

\*1 IF indicates IFF2.

	000	001	010	011	100	101	110	111
r	B	C	D	E	H	L		A

### 16 BIT LOAD GROUP

Mnemonic	Operation	F l a g s S Z H P V N C	O p c o d e		B	CL
			76543210	Hex		
LD ss, nn	ss<-nn	● ● ● ● ● ●	00ss0001		3	3
			<- nnl ->			
LD IX, nn	IX<-nn	● ● ● ● ● ●	<- nnh ->		4	4
			11011101	DD		
LD IY, nn	IY<-nn	● ● ● ● ● ●	00100001	21	4	4
			<- nnl ->			
			<- nnh ->			
			11111101	FD		

Mnemonic	Operation	F l a g s S Z H P V N C	Opcode		B	CL
			76543210	Hex		
			00100001	21		
			< -nnl - >			
			< -nnh - >			
LD SP, HL	SP<-HL	● ● ● ● ● ●	11111001	F9	1	1
LD SP, IX	SP<-IX	● ● ● ● ● ●	11011101	DD	2	2
			11111001	F9		
LD SP, IY	SP<-IY	● ● ● ● ● ●	11111101	FD	2	2
			11111001	F9		
LD ss, [nn]	ssh<-[nn+1]	● ● ● ● ● ●	11101101	ED	4	6
	ssl<-[nn]		01ss1011			
			< -nnl - >			
			< -nnh - >			
LD HL, [nn]	H<-[nn+1]	● ● ● ● ● ●	00101010	2A	3	5
	L<-[nn]		< -nnl - >			
			< -nnh - >			
LD IX, [nn]	IXH<-[nn+1]	● ● ● ● ● ●	11011101	DD	4	6
	IXL<-[nn]		00101010	2A		
			< -nnl - >			
			< -nnh - >			
LD IY, [nn]	IYH<-[nn+1]	● ● ● ● ● ●	11111101	FD	4	6
	IYL<-[nn]		00101010	2A		
			< -nnl - >			
			< -nnh - >			
LD [nn], ss	[nn+1]<-ssh	● ● ● ● ● ●	11101101	ED	4	6
	[nn]<-ssl		01ss0011			
			< -nnl - >			
			< -nnh - >			
LD [nn], HL	[nn+1]<-H	● ● ● ● ● ●	00100010	22	3	5
	[nn]<-L		< -nnl - >			
			< -nnh - >			
LD [nn], IX	[nn+1]<-IXH	● ● ● ● ● ●	11011101	DD	4	6
	[nn]<-IXL		00100010	22		
			< -nnl - >			
			< -nnh - >			
LD [nn], IY	[nn+1]<-IYH	● ● ● ● ● ●	11111101	FD	4	6
	[nn]<-IYL		00100010	22		
			< -nnl - >			
			< -nnh - >			

	00	01	10	11
ss	BC	DE	HL	SP

## EXCHANGE GROUP

Mnemonic	Operation	F l a g s S Z H P V N C	O p c o d e		B	C L
			76543210	Hex		
EX DE, HL	DE<->HL	● ● ● ● ● ●	11101011	EB	1	1
EX AF, AF'	AF<->AF'	⇔ ⇔ ⇔ ⇔ ⇔ ⇔	00001000	08	1	1
EX [SP], HL	L<->[SP] H<->[SP+1]	● ● ● ● ● ●	11100011	E3	1	5
EX [SP], IX	IXL<->[SP] IXH<->[SP+1]	● ● ● ● ● ●	11011101 11100011	DD E3	2	6
EX [SP], IY	IYL<->[SP] IYH<->[SP+1]	● ● ● ● ● ●	11111101 11100011	FD E3	2	6
EXX	BC<->BC' DE<->DE' HL<->HL'	● ● ● ● ● ●	11011001	D9	1	1

## PUSH / POP GROUP

Mnemonic	Operation	F l a g s S Z H P V N C	O p c o d e		B	C L
			76543210	Hex		
PUSH qq	[SP-2]<-qql [SP-1]<-qqh SP<-SP-2	● ● ● ● ● ●	11qq0101		1	4
PUSH IX	[SP-2]<-IXL [SP-1]<-IXH SP<-SP-2	● ● ● ● ● ●	11011101 11100101	DD E5	2	5
PUSH IY	[SP-2]<-IYL [SP-1]<-IYH SP<-SP-2	● ● ● ● ● ●	11111101 11100101	FD E5	2	5
POP qq	qql<-[SP] qqh<-[SP+1] SP<-SP+2	● ● ● ● ● ●	11qq0001		1	3
POP IX	IXL<-[SP] IXH<-[SP+1] SP<-SP+2	● ● ● ● ● ●	11011101 11100001	DD E1	2	4
POP IY	IYL<-[SP] IYH<-[SP+1] SP<-SP+2	● ● ● ● ● ●	11111101 11100001	FD E1	2	4

	00	01	10	11
qq	BC	DE	HL	AF

All flags change on POP AF.

BLOCK TRANSFER GROUP

Mnemonic	Operation	F l a g s S Z H P V N C	O p c o d e		B	CL
			76543210	Hex		
LDI	[DE]<-[HL]	● ● 0 ⇔ 0 ●	11101101	ED	2	5
	DE<-DE+1	*1	10100000	A0		
LDD	[DE]<-[HL]	● ● 0 ⇔ 0 ●	11101101	ED	2	5
	DE<-DE-1	*1	10101000	A8		
LDIR	repeat;	● ● 0 0 0 ●	11101101	ED	2	6
	[DE]<-[HL]		10110000	B0		
LDDR	repeat;	● ● 0 0 0 ●	11101101	ED	2	6
	[DE]<-[HL]		10111000	B8		

\*1 If BC=0 PV=0, else PV=1

BLOCK SEARCH GROUP

Mnemonic	Operation	F l a g s S Z H P V N C	O p c o d e		B	CL
			76543210	Hex		
CPI	A-[HL];HL<-	⇔ ⇔ ⇔ ⇔ 1 ●	11101101	ED	2	4
	HL+1;BC<-BC-1	*2 *1	10100001	A1		
CPD	A-[HL];HL<-	⇔ ⇔ ⇔ ⇔ 1 ●	11101101	ED	2	4
	HL-1;BC<-BC-1	*2 *1	10101001	A9		
CPIR	repeat;A-[HL]	⇔ ⇔ ⇔ ⇔ 1 ●	11101101	ED	2	6
	HL<-HL+1	*2 *1	10110001	B1		
CPDR	repeat;A-[HL]	⇔ ⇔ ⇔ ⇔ 1 ●	11101101	ED	2	6
	HL<-HL-1	*2 *1	10111001	B9		

\*1 If BC=0 PV=0, else PV=1

\*2 If A=[HL] Z=1, else Z=0

### ADD GROUP

Mnemonic	Operation	F l a g s S Z H P V N C	O p c o d e		B	CL
			76543210	Hex		
ADD A, r	A←A+r	⇔⇔⇔V 0⇔	10000 r		1	1
ADD A, n	A←A+n	⇔⇔⇔V 0⇔	11000110	C6	2	2
			< - n - >			
ADD A, [HL]	A←A+[HL]	⇔⇔⇔V 0⇔	10000110	86	1	2
ADD A, [IX+d]	A←A+[IX+d]	⇔⇔⇔V 0⇔	11011101	DD	3	5
			10000110	86		
			< - d - >			
ADD A, [IY+d]	A←A+[IY+d]	⇔⇔⇔V 0⇔	11111101	FD	3	5
			10000110	86		
			< - d - >			
ADC A, r	A←A+r+C	⇔⇔⇔V 0⇔	10001 r		1	1
ADC A, n	A←A+n+C	⇔⇔⇔V 0⇔	11001110	CE	2	2
			< - n - >			
ADC A, [HL]	A←A+[HL]+C	⇔⇔⇔V 0⇔	10001110	8E	1	2
ADC A, [IX+d]	A←A+[IX+d]+C	⇔⇔⇔V 0⇔	11011101	DD	3	5
			10001110	8E		
			< - d - >			
ADC A, [IY+d]	A←A+[IY+d]+C	⇔⇔⇔V 0⇔	11111101	FD	3	5
			10001110	8E		
			< - d - >			
ADC HL, ss	HL←HL+ss+C	⇔⇔ ? V 0⇔	11101101	ED	2	2
			01ss1010			
ADD HL, ss	HL←HL+ss	● ● ? ● 0⇔	00ss1001		1	1
ADD IX, pp	IX←IX+pp	● ● ? ● 0⇔	11011101	DD	2	2
			00pp1001			
ADD IY, rr	IY←IY+rr	● ● ? ● 0⇔	11111101	FD	2	2
			00rr1001			
INC r	r←r+1	⇔⇔⇔V 0●	00 r 100		1	1
INC [HL]	[HL]←[HL]+1	⇔⇔⇔V 0●	00110100	34	1	4
INC [IX+d]	[IX+d]←[IX+d]+1	⇔⇔⇔V 0●	11011101	DD	3	7
			00110100	34		
			< - d - >			
INC [IY+d]	[IY+d]←[IY+d]+1	⇔⇔⇔V 0●	11111101	FD	3	7
			00110100	34		
			< - d - >			
INC ss	ss←ss+1	● ● ● ● ● ●	00ss0011		1	1
INC IX	IX←IX+1	● ● ● ● ● ●	11011101	DD	2	2

Mnemonic	Operation	F l a g s S Z H P V N C	O p c o d e		B	C L
			76543210	Hex		
INC IY	IY<-IY+1	● ● ● ● ● ●	00100011	23	2	2
			11111101	FD		
			00100011	23		

	00	01	10	11
ss	BC	DE	HL	SP
pp	BC	DE	IX	SP
rr	BC	DE	IX	SP

SUBTRACT GROUP

Mnemonic	Operation	F l a g s S Z H P V N C	O p c o d e		B	C L
			76543210	Hex		
SUB r	A<-A-r	⇔ ⇔ ⇔ V 1 ⇔	10010 r		1	1
SUB n	A<-A-n	⇔ ⇔ ⇔ V 1 ⇔	11010110	D6	2	2
			<- n ->			
SUB [HL]	A<-A-[HL]	⇔ ⇔ ⇔ V 1 ⇔	10010110	96	1	2
SUB [IX+d]	A<-A-[IX+d]	⇔ ⇔ ⇔ V 1 ⇔	11011101	DD	3	5
			10010110	96		
			<- d ->			
SUB [IY+d]	A<-A-[IY+d]	⇔ ⇔ ⇔ V 1 ⇔	11111101	FD	3	5
			10010110	96		
			<- d ->			
SBC A, r	A<-A-r-C	⇔ ⇔ ⇔ V 1 ⇔	10011 r		1	1
SBC A, n	A<-A-n-C	⇔ ⇔ ⇔ V 1 ⇔	11011110	DE	2	2
			<- n ->			
			10011110	9E		
SBC A, [HL]	A<-A-[HL]-C	⇔ ⇔ ⇔ V 1 ⇔	11011101	DD	3	5
SBC A, [IX+d]	A<-A-[IX+d]-C	⇔ ⇔ ⇔ V 1 ⇔	10011110	9E	3	5
			<- d ->			
			11111101	FD		
SBC A, [IY+d]	A<-A-[IY+d]-C	⇔ ⇔ ⇔ V 1 ⇔	10011110	9E	3	5
			<- d ->			
			11101101	ED		
SBC HL, ss	HL<-HL-ss-C	⇔ ⇔ ? V 1 ⇔	01ss0010		2	2
DEC r	r<-r-1	⇔ ⇔ ⇔ V 1 ●	00 r 101		1	1
DEC [HL]	[HL]<-[HL]-1	⇔ ⇔ ⇔ V 1 ●	00110101	35	1	4
DEC [IX+d]	[IX+d]<-[IX+d]-1	⇔ ⇔ ⇔ V 1 ●	11011101	DD	3	7
			00110101	35		
			<- d ->			
DEC [IY+d]	[IY+d]<-[IY+d]-1	⇔ ⇔ ⇔ V 1 ●	11111101	FD	3	7
			00110101	35		
			<- d ->			

Mnemonic	Operation	F l a g s S Z H P V N C	Opcode		B	CL
			76543210	Hex		
DEC ss	ss<-ss-1	● ● ● ● ● ●	00ss1011		1	1
DEC IX	IX<-IX-1	● ● ● ● ● ●	11011101	DD	2	2
			00101011	2B		
DEC IY	IY<-IY-1	● ● ● ● ● ●	11111101	FD	2	2
			00101011	2B		

### COMPARE GROUP

Mnemonic	Operation	F l a g s S Z H P V N C	Opcode		B	CL
			76543210	Hex		
CP r	A-r	⇔ ⇔ ⇔ V 1 ⇔	10111 r		1	1
CP n	A-n	⇔ ⇔ ⇔ V 1 ⇔	11111110	FE	2	2
			< - n - >			
CP [HL]	A-[HL]	⇔ ⇔ ⇔ V 1 ⇔	10111110	BE	1	2
CP [IX+d]	A-[IX+d]	⇔ ⇔ ⇔ V 1 ⇔	11011101	DD	3	5
			10111110	BE		
			< - d - >			
CP [IY+d]	A-[IY+d]	⇔ ⇔ ⇔ V 1 ⇔	11111101	FD	3	5
			10111110	BE		
			< - d - >			

### LOGICAL GROUP

Mnemonic	Operation	F l a g s S Z H P V N C	Opcode		B	CL
			76543210	Hex		
AND r	A<-A ^ r	⇔ ⇔ 1 P 0 0	10100 r		1	1
AND n	A<-A ^ n	⇔ ⇔ 1 P 0 0	11100110	E6	2	2
			< - n - >			
AND [HL]	A<-A ^ [HL]	⇔ ⇔ 1 P 0 0	10100110	A6	1	2
AND [IX+d]	A<-A ^ [IX+d]	⇔ ⇔ 1 P 0 0	11011101	DD	3	5
			10100110	A6		
			< - d - >			
AND [IY+d]	A<-A ^ [IY+d]	⇔ ⇔ 1 P 0 0	11111101	FD	3	5
			10100110	A6		
			< - d - >			
OR r	A<-A V r	⇔ ⇔ 0 P 0 0	10110 r		1	1
OR n	A<-A V n	⇔ ⇔ 0 P 0 0	11110110	F6	2	2
			< - n - >			
OR [HL]	A<-A V [HL]	⇔ ⇔ 0 P 0 0	10110110	B6	1	2
OR [IX+d]	A<-A V [IX+d]	⇔ ⇔ 0 P 0 0	11011101	DD	3	5
			10110110	B6		
			< - d - >			

Mnemonic	Operation	F l a g s S Z H P V N C	O p c o d e		B	C L
			76543210	Hex		
OR [IY+d]	A<-A ∨[IY+d]	⇔⇔ 0 P 0 0	11111101 10110110 < - d - >	FD B6	3	5
XOR r	A<-A ∨ r	⇔⇔ 0 P 0 0	10101 r		1	1
XOR n	A<-A ∨ n	⇔⇔ 0 P 0 0	11101110 < - n - >	EE	2	2
XOR [HL]	A<-A ∨[HL]	⇔⇔ 0 P 0 0	10101110	AE	1	2
XOR [IX+d]	A<-A ∨[IX+d]	⇔⇔ 0 P 0 0	11011101 10101110 < - d - >	DD AE	3	5
XOR [IY+d]	A<-A ∨[IY+d]	⇔⇔ 0 P 0 0	11111101 10101110 < - d - >	FD AE	3	5

### BIT SET AND TEST GROUP

Mnemonic	Operation	F l a g s S Z H P V N C	O p c o d e		B	C L
			76543210	Hex		
BIT b, r	Z<-NOT r{b}	? ⇔ 1 ? 0 ●	11001011 01 b r	CB	2	2
BIT b, [HL]	Z<-NOT [HL]{b}	? ⇔ 1 ? 0 ●	11001011 01 b 110	CB	2	3
BIT b, [IX+d]	Z<-NOT [IX+d]{b}	? ⇔ 1 ? 0 ●	11011101 11001011 < - d - >	DD CB	4	5
BIT b, [IY+d]	Z<-NOT [IY+d]{b}	? ⇔ 1 ? 0 ●	01 b 110 11111101 11001011 < - d - >	FD CB	4	5
SET b, r	r{b}<-1	● ● ● ● ● ●	01 b 110 11001011 11 b r	CB	2	2
SET b, [HL]	[HL]{b}<-1	● ● ● ● ● ●	11001011 11 b 110	CB	2	5
SET b, [IX+d]	[IX+d]{b}<-1	● ● ● ● ● ●	11011101 11001011 < - d - >	DD CB	4	7
SET b, [IY+d]	[IY+d]{b}<-1	● ● ● ● ● ●	11 b 110 11111101 11001011 < - d - >	FD CB	4	7
RES b, r	r{b}<-0	● ● ● ● ● ●	11001011 10 b r	CB	2	2

Mnemonic	Operation	F l a g s S Z H P V N C	O p c o d e		B	CL
			76543210	Hex		
RES b, [HL]	[HL]{b}<-0	● ● ● ● ● ●	11001011 10 b 110	CB	2	5
RES b, [IX+d]	[IX+d]{b}<-0	● ● ● ● ● ●	11011101 11001011 < - d - > 10 b 110	DD CB	4	7
RES b, [IY+d]	[IY+d]{b}<-0	● ● ● ● ● ●	11111101 11001011 < - d - > 10 b 110	FD CB	4	7

### ROTATE GROUP

Mnemonic	Operation	F l a g s S Z H P V N C	O p c o d e		B	CL
			76543210	Hex		
RLCA	C<-A{7};A<-A*2 A{0}<-C	● ● 0 ● 0 ⇔	00001111	07	1	1
RRCA	C<-A{0};A<-A/2 A{7}<-C	● ● 0 ● 0 ⇔	00001111	0F	1	1
RLA	tmp<-C;C<-A{7} A<-A*2 A{0}<-tmp	● ● 0 ● 0 ⇔	00010111	17	1	1
RRA	tmp<-C;C<-A{0} A<-A/2 A{7}<-tmp	● ● 0 ● 0 ⇔	00011111	1F	1	1
RLC r	C<-r{7} r<-r*2;r{0}<-C	⇔ ⇔ 0 P 0 ⇔	11001011 00000 r	CB	2	2
RLC [HL]	C<-[HL]{7} [HL]<-[HL]*2 [HL]{0}<-C	⇔ ⇔ 0 P 0 ⇔	11001011 00000110	CB 06	2	5
RLC [IX+d]	C<-[IX+d]{7} [IX+d]<-[IX+d]*2 [IX+d]{0}<-C	⇔ ⇔ 0 P 0 ⇔	11011101 11001011 < - d - > 00000110	DD CB 06	4	7
RLC [IY+d]	C<-[IY+d]{7} [IY+d]<-[IY+d]*2 [IY+d]{0}<-C	⇔ ⇔ 0 P 0 ⇔	11111101 11001011 < - d - > 00000110	FD CB 06	4	7
RRC r	C<-r{0} r<-r/2;r{7}<-C	⇔ ⇔ 0 P 0 ⇔	11001011 00001 r	CB	2	2
RRC [HL]	C<-[HL]{0} [HL]<-[HL]/2 [HL]{7}<-c	⇔ ⇔ 0 P 0 ⇔	11001011 00001110	CB 0E	2	5

Mnemonic	Operation	F l a g s S Z H P V N C	Opcode		B	CL
			76543210	Hex		
RRC [IX+d]	C<-[IX+d]{0}	⇔ ⇔ 0 P 0 ⇔	11011101	DD	4	7
	[IX+d]<-[IX+d]/2		11001011	CB		
	[IX+d]{7}<-C		<- d ->			
			00001110	0E		
RRC [IY+d]	C<-[IY+d]{0}	⇔ ⇔ 0 P 0 ⇔	11111101	FD	4	7
	[IY+d]<-[IY+d]/2		11001011	CB		
	[IY+d]{7}<-C		<- d ->			
			00001110	0E		
RL r	tmp<-C;C<-r{7}	⇔ ⇔ 0 P 0 ⇔	11001011	CB	2	2
	r<-r*2;r{0}<-tmp		00010 r			
RL [HL]	tmp<-C;C<-[HL]{7}	⇔ ⇔ 0 P 0 ⇔	11001011	CB	2	5
	[HL]<-[HL]*2		00010110	16		
	[HL]{0}<-tmp					
RL [IX+d]	tmp<-C	⇔ ⇔ 0 P 0 ⇔	11011101	DD	4	7
	C<-[IX+d]{7}		11001011	CB		
	[IX+d]<-[IX+d]*2		<- d ->			
	[IX+d]{0}<-tmp		00010110	16		
RL [IY+d]	tmp<-C	⇔ ⇔ 0 P 0 ⇔	11111101	FD	4	7
	C<-[IY+d]{7}		11001011	CB		
	[IY+d]<-[IY+d]*2		<- d ->			
	[IY+d]{0}<-tmp		00010110	16		
RR r	tmp<-C;C<-r{0}	⇔ ⇔ 0 P 0 ⇔	11001011	CB	2	2
	r<-r/2;r{7}<-tmp		00011 r			
RR [HL]	tmp<-C;C<-r{0}	⇔ ⇔ 0 P 0 ⇔	11001011	CB	2	5
	[HL]<-[HL]/2		00011110	1E		
	[HL]{7}<-tmp					
RR [IX+d]	tmp<-C	⇔ ⇔ 0 P 0 ⇔	11011101	DD	4	7
	C<-[IX+d]{0}		11001011	CB		
	[IX+d]<-[IX+d]/2		<- d ->			
	[IX+d]{7}<-tmp		00011110	1E		
RR [IY+d]	tmp<-C	⇔ ⇔ 0 P 0 ⇔	11111101	FD	4	7
	C<-[IY+d]{0}		11001011	CB		
	[IY+d]<-[IY+d]/2		<- d ->			
	[IY+d]{7}<-tmp		00011110	1E		

Mnemonic	Operation	F l a g s S Z H P V N C	O p c o d e		B	CL
			76543210	Hex		
RLD	tmp<-A{0..3}	⇔ ⇔ 0 P 0 ⇔	11101101	ED	2	5
	A{0..3}		01101111	6F		
RRD	<-[HL]{4..7}	⇔ ⇔ 0 P 0 ⇔	11101101	ED	2	5
	[HL]{4..7}		01100111	67		
	<-[HL]{0..3}	⇔ ⇔ 0 P 0 ⇔	11101101	ED	2	5
	[HL]{0..3}<-tmp		01100111	67		
	tmp<-A{0..3}	⇔ ⇔ 0 P 0 ⇔	11101101	ED	2	5
	A{0..3}		01100111	67		
	<-[HL]{0..3}	⇔ ⇔ 0 P 0 ⇔	11101101	ED	2	5
	[HL]{0..3}		01100111	67		
	<-[HL]{4..7}	⇔ ⇔ 0 P 0 ⇔	11101101	ED	2	5
	[HL]{4..7}<-tmp		01100111	67		

### SHIFT GROUP

Mnemonic	Operation	F l a g s S Z H P V N C	O p c o d e		B	CL
			76543210	Hex		
SLA r	C<-r{7}	⇔ ⇔ 0 P 0 ⇔	11001011	CB	2	2
	r<-r*2		00100 r			
SLA [HL]	C<-[HL]{7}	⇔ ⇔ 0 P 0 ⇔	11001011	CB	2	5
	[HL]<-[HL]*2		00100110	26		
SLA [IX+d]	C<-[IX+d]{7}	⇔ ⇔ 0 P 0 ⇔	11011101	DD	4	7
	[IX+d]<-[IX+d]*2		11001011	CB		
			<- d ->			
			00100110	26		
SLA [IY+d]	C<-[IY+d]{7}	⇔ ⇔ 0 P 0 ⇔	11111101	FD	4	7
	[IY+d]<-[IY+d]*2		11001011	CB		
			<- d ->			
			00100110	26		
SRL r	C<-r{0}	⇔ ⇔ 0 P 0 ⇔	11001011	CB	2	2
	r<-r/2		00111 r			
SRL [HL]	C<-[HL]{0}	⇔ ⇔ 0 P 0 ⇔	11001011	CB	2	5
	[HL]<-[HL]/2		00111110	3E		
SRL [IX+d]	C<-[IX+d]{0}	⇔ ⇔ 0 P 0 ⇔	11011101	DD	4	7
	[IX+d]<-[IX+d]/2		11001011	CB		
			<- d ->			
			00111110	3E		
SRL [IY+d]	C<-[IY+d]{0}	⇔ ⇔ 0 P 0 ⇔	11111101	FD	4	7
	[IY+d]<-[IY+d]/2		11001011	CB		
			<- d ->			
			00111110	3E		
SRA r	tmp<-r{7};C<-r{0}	⇔ ⇔ 0 P 0 ⇔	11001011	CB	2	2
	r<-r/2		00101 r			

Mnemonic	Operation	F l a g s S Z H P V N C	O p c o d e		B	CL
			76543210	Hex		
SRA [HL]	r{7}<-tmp	⇔ ⇔ 0 P 0 ⇔	11001011	CB	2	5
	tmp<-[HL]{7}		00101110	2E		
SRA [IX+d]	C<-[HL]{0}	⇔ ⇔ 0 P 0 ⇔	11011101	DD	4	7
	[HL]<-[HL]/2		11001011	CB		
SRA [IY+d]	[HL]{7}<-tmp	⇔ ⇔ 0 P 0 ⇔	<- d ->		4	7
	tmp<-[IX+d]{7}		00101110	2E		
SRA [IX+d]	C<-[IX+d]{0}	⇔ ⇔ 0 P 0 ⇔	11111101	FD	4	7
	[IX+d]<-[IX+d]/2		11001011	CB		
SRA [IY+d]	[IX+d]{7}<-tmp	⇔ ⇔ 0 P 0 ⇔	<- d ->		4	7
	tmp<-[IY+d]{7}		00101110	2E		
SRA [IY+d]	C<-[IY+d]{0}	⇔ ⇔ 0 P 0 ⇔	11111101	FD	4	7
	[IY+d]<-[IY+d]/2		11001011	CB		
SRA [IY+d]	[IY+d]{7}<-tmp	⇔ ⇔ 0 P 0 ⇔	<- d ->		4	7
	tmp<-[IY+d]{7}		00101110	2E		

### JUMP GROUP

Mnemonic	Operation	F l a g s S Z H P V N C	O p c o d e		B	CL
			76543210	Hex		
JP nn	PC<-nn	● ● ● ● ● ●	11000011	C3	3	3
JP NZ, nn	if Z=0 PC<-nn	● ● ● ● ● ●	<- nnl ->	C2	3	3
			<- nnh ->			
JP Z, nn	if Z=1 PC<-nn	● ● ● ● ● ●	11001010	CA	3	3
			<- nnl ->	<- nnh ->		
JP NC, nn	if C=0 PC<-nn	● ● ● ● ● ●	11010010	D2	3	3
			<- nnl ->	<- nnh ->		
JP C, nn	if C=1 PC<-nn	● ● ● ● ● ●	11011010	DA	3	3
			<- nnl ->	<- nnh ->		
JP PO, nn	if PV=0 PC<-nn	● ● ● ● ● ●	11100010	E2	3	3
			<- nnl ->	<- nnh ->		
JP PE, nn	if PV=1 PC<-nn	● ● ● ● ● ●	11101010	EA	3	3
			<- nnl ->	<- nnh ->		
JP P, nn	if S=0 PC<-nn	● ● ● ● ● ●	11110010	F2	3	3
			<- nnl ->	<- nnh ->		
JP M, nn	if S=1	● ● ● ● ● ●	11111010	FA	3	3

Mnemonic	Operation	F l a g s S Z H P V N C	O p c o d e		B	CL
			76543210	Hex		
	PC<-nn		< - nnl ->			
			< - nnh ->			
JP [HL]	PC<- [HL]	● ● ● ● ● ●	11101001	E9	1	1
JP [IX]	PC<- [IX]	● ● ● ● ● ●	11011101	DD	2	2
			11101001	E9		
JP [IY]	PC<- [IY]	● ● ● ● ● ●	11111101	FD	2	2
			11101001	E9		
JR e	PC<-PC+e	● ● ● ● ● ●	00011000	18	2	3
			< -e-2 ->			
JR NZ, e	if Z=0	● ● ● ● ● ●	00100000	20	2	2
	PC<-PC+e		< -e-2 ->			3
JR Z, e	if Z=1	● ● ● ● ● ●	00101000	28	2	2
	PC<-PC+e		< -e-2 ->			3
JR NC, e	if C=0	● ● ● ● ● ●	00110000	30	2	2
	PC<-PC+e		< -e-2 ->			3
JR C, e	if C=1	● ● ● ● ● ●	00111000	38	2	2
	PC<-PC+e		< -e-2 ->			3
DJNZ e	B<-B-1;if B><0	● ● ● ● ● ●	00010000	10	2	3
	PC<-PC+e		< -e-2 ->			
CALL nn	[SP-2]<-PCL	● ● ● ● ● ●	11001101	CD	3	5
	[SP-1]<-PCH		< - nnl ->			
	SP<-SP-2		< - nnh ->			
	PC<-nn					
CALL NZ, nn	if Z=0	● ● ● ● ● ●	11000100	C4	3	3
	[SP-2]<-PCL		< - nnl ->			5
	[SP-1]<-PCH		< - nnh ->			
	SP<-SP-2					
	PC<-nn					
CALL Z, nn	if Z=1	● ● ● ● ● ●	11001100	CC	3	3
	[SP-2]<-PCL		< - nnl ->			5
	[SP-1]<-PCH		< - nnh ->			
	SP<-SP-2					
	PC<-nn					
CALL NC, nn	if C=0	● ● ● ● ● ●	11010100	D4	3	3
	[SP-2]<-PCL		< - nnl ->			5
	[SP-1]<-PCH		< - nnh ->			
	SP<-SP-2					
	PC<-nn					
CALL C, nn	if C=1	● ● ● ● ● ●	11011100	DC	3	3
	[SP-2]<-PCL		< - nnl ->			5
	[SP-1]<-PCH		< - nnh ->			
	SP<-SP-2					
	PC<-nn					
CALL PO, nn	if PV=0	● ● ● ● ● ●	11100100	E4	3	3

Mnemonic	Operation	F l a g s S Z H P V N C	Opcode		B	CL
			76543210	Hex		
CALL PE, nn	[SP-2]<-PCL		< - nml - >			5
	[SP-1]<-PCH		< - nnh - >			
	SP<-SP-2					
CALL P, nn	PC<-nn					
	if PV=1	● ● ● ● ● ●	11101100	EC	3	3
	[SP-2]<-PCL		< - nml - >			5
CALL M, nn	[SP-1]<-PCH		< - nnh - >			
	SP<-SP-2					
	PC<-nn					
CALL P, nn	if S=0	● ● ● ● ● ●	111100100	F4	3	3
	[SP-2]<-PCL		< - nml - >			5
	[SP-1]<-PCH		< - nnh - >			
CALL M, nn	SP<-SP-2					
	PC<-nn					
	if S=1	● ● ● ● ● ●	11111100	FC	3	3
RET	[SP-2]<-PCL		< - nml - >			5
	[SP-1]<-PCH		< - nnh - >			
	SP<-SP-2					
RET NZ	PC<-nn					
	PCL<-[SP]	● ● ● ● ● ●	11001001	C9	1	3
	PCH<-[SP+1]					
RET Z	SP<-SP+2					
	if Z=0	● ● ● ● ● ●	11000000	C0	1	2
	PCL<-[SP]					4
RET Z	PCH<-[SP+1]					
	SP<-SP+2					
	if Z=1	● ● ● ● ● ●	11001000	C8	1	2
RET NC	PCL<-[SP]					4
	PCH<-[SP+1]					
	SP<-SP+2					
RET C	if C=0	● ● ● ● ● ●	11010000	D0	1	2
	PCL<-[SP]					4
	PCH<-[SP+1]					
RET C	SP<-SP+2					
	if C=1	● ● ● ● ● ●	11011000	D8	1	2
	PCL<-[SP]					4
RET PO	PCH<-[SP+1]					
	SP<-SP+2					
	if PV=0	● ● ● ● ● ●	11100000	E0	1	2
RET PE	PCL<-[SP]					4
	PCH<-[SP+1]					
	SP<-SP+2					
RET PE	if PV=1	● ● ● ● ● ●	11101000	E8	1	2
	PCL<-[SP]					4

Mnemonic	Operation	F l a g s S Z H P V N C	O p c o d e		B	CL
			76543210	Hex		
RET P	PCH<-[SP+1] SP<-SP+2 if S=0	● ● ● ● ● ●	11110000	F0	1	2 4
	PCL<-[SP] PCH<-[SP+1] SP<-SP+2					
RET M	if S=1	● ● ● ● ● ●	11111000	F8	1	2 4
	PCL<-[SP] PCH<-[SP+1] SP<-SP+2					
RETI	interrupt return	● ● ● ● ● ●	11101101 01001101	ED 4D	2	7
RETN	non maskable interrupt return	● ● ● ● ● ●	11101101 01000101	ED 45	2	4
RST k	[SP-2]<-PCL [SP-1]<-PCH SP<-SP-2;PCL<-k PCH<-0	● ● ● ● ● ●	11k/8111		1	4

#### INPUT AND OUTPUT GROUP

Mnemonic	Operation	F l a g s S Z H P V N C	O p c o d e		B	CL
			76543210	Hex		
IN A, [n]	A<-[n]	● ● ● ● ● ●	11011011 <- n ->	DB	2	4
IN r, [C]	r<-[C]	⇔ ⇔ 0 P 0 ●	11101101 01 r 000	ED	2	4
INI	[HL]<-[C] B<-B-1	? ⇔ ? ? 1 ● *1	11101101 10100010	ED A2	2	5
	HL<-HL+1					
IND	[HL]<-[C] B<-B-1	? ⇔ ? ? 1 ● *1	11101101 10101010	ED AA	2	5
	HL<-HL-1					
INIR	repeat;[HL]<-[C] B<-B-1	? 1 ? ? 1 ●	11101101 10110010	ED B2	2	6 6
	HL<-HL+1					
INDR	until B=0 repeat;[HL]<-[C] B<-B-1	? 1 ? ? 1 ●	11101101 10111010	ED BA	2	6 6
	HL<-HL-1					
OUT [n], A	[n]<-A	● ● ● ● ● ●	11010011	D3	2	4

Mnemonic	Operation	F l a g s S Z H P V N C	Opcode		B	CL
			76543210	Hex		
OUT [C], r	[C]<-r	● ● ● ● ● ●	<- n -> 11101101 01 r 001	ED	2	4
OUTI	[C]<-[HL]	? ⇔ ? ? 1 ●	11101101	ED	2	5
	B<-B-1	*1	10100011	A3		
OUTD	HL<-HL+1					
	[C]<-[HL]	? ⇔ ? ? 1 ●	11101101	ED	2	5
	B<-B-1	*1	10101011	AB		
	HL<-HL-1					
OTIR	repeat;[C]<-[HL]	? 1 ? ? 1 ●	11101101	ED	2	7
	B<-B-1		10110011	B3		7
	HL<-HL+1					
	until B=0					
OTDR	repeat;[C]<-[HL]	? 1 ? ? 1 ●	11101101	ED	2	7
	B<-B-1		10111011	BB		7
	HL<-HL-1					
	until B=0					

\*1 If B=0, Z=1, else Z=0

#### GENERAL-PURPOSE ARITHMETIC AND CPU CONTROL GROUPS

Mnemonic	Operation	F l a g s S Z H P V N C	Opcode		B	CL
			76543210	Hex		
DAA	adjust to decimal	⇔ ⇔ ⇔ P ● ⇔	00100111	27	1	1
CPL	A<-NOT A	● ● 1 ● 1 ●	00101111	2F	1	1
NEG	A<-NOT A+1	⇔ ⇔ ⇔ V 1 ⇔	11101101	ED	2	2
			01000100	44		
CCF	C<-NOT C	● ● ? ● 0 ⇔	00111111	3F	1	1
SCF	C<-1	● ● 0 ● 0 1	00110111	37	1	1
NOP	NO operation	● ● ● ● ● ●	00000000	00	1	1
HALT	HALT	● ● ● ● ● ●	01110110	76	1	2
DI	IFF<-0	● ● ● ● ● ●	11110011	F3	1	2
EI	IFF<-1	● ● ● ● ● ●	11111011	FB	1	2
IM 0	interrupt mode 0	● ● ● ● ● ●	11101101	ED	2	2
			01000110	46		
IM 1	interrupt mode 1	● ● ● ● ● ●	11101101	ED	2	2
			01010110	56		
IM 2	interrupt mode 2	● ● ● ● ● ●	11101101	ED	2	2
			01011110	5E		



- 本製品、および本書に記載された、情報・回路の使用に対して、当社は第三者の工業所有権、知的財産権、およびその他の権利に対する保証または実施権の許諾を行うものではありません。
- 本製品は、外国為替および外国貿易管理法に定める戦略物資に該当しますので、本製品を輸出する場合、同法に基づく輸出許可が必要です。
- 本製品を、極めて高い信頼性が要求される用途にご使用の場合は、事前に当社営業までご連絡下さい。

## 川崎製鉄株式会社

LSI事業部 営業・開発部  
〒261-01 千葉市美浜区中瀬1-3 B-11  
TEL (043) 296-7412 (ダイヤルイン)  
FAX (043) 296-7419

お問い合わせ先