

Z80 互換高速 CPU 搭載
通信用コントローラ
KL5C80A20CFP
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. HDLC シリアルインターフェース
 12. パラレルポート
 13. 動作モード設定
 14. アドレスマッピング
 15. 発振回路
 16. 電気的特性
 17. 外形寸法図
- 付録 A インストラクション表

1996.2 Ver. 1.0
川崎製鉄株式会社
LSI 事業部

1. 概要

KL5C80A20 は最新の CMOS プロセスを駆使して開発された高速 HDLC シリアルインターフェース搭載 Z80 互換高速マイクロコントローラです。最大転送速度 5M bps の HDLC シリアルインターフェースを 1 チャンネル搭載しており、HDLC フレームの高速送受信が可能です。HDLC シリアルインターフェースは 8 段の受信 FIFO、4 段の送信 FIFO を内蔵しているため、高速転送時のソフトウェア処理が容易になり、また、送受信割り込みのオーバーヘッドを低減することが可能です。さらに、ボーレート・ジェネレータ、クロック再生のための DPLL (Digital Phase Locked Loop) を搭載しているため、送受信クロックが付加回路なしで生成できます。

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。KC82CPU コアは 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。

KL5C80A20 は CPU コア以外にも高速 HDLC シリアルインターフェース、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、高速シリアル通信が要求されるシステムに最適です。また、KL5C80A20 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

HDLC シリアルインターフェース部

- 最大転送速度 5M bps (システムクロック 10 MHz、DPLL 非 使用時)、DPLL 使用時は 1.25M bps
- 自動 '0' 挿入、削除機能
- アボート送出、検出機能
- CRC 生成、検査 (CRC-16 or CRC-CCITT, Preset '1' or '0')
- 受信 FIFO 8 段、送信 FIFO 4 段
- NRZ, NRZI, FM0, FM1 データ・エンコーディング対応
- ボーレート・ジェネレータ内蔵
- クロック再生用 DPLL 内蔵
- ループ・モード対応
- 送信割り込み、受信割り込み、External/Status 割り込み、Special Condition 割り込み発生機能 (弊社割り込みコントローラ KP69 と接続することにより多重割り込み対応可能)
- 送信 DMA 転送要求、受信 DMA 転送要求発生機能
- Local Loopback、Auto Echo 機能
- 最大動作周波数 10 MHz

その他の部分

- 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、1 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・マスク割込み 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

KL5C80A20 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。KL5C80A20 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。KL5C80A20 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。KL5C80A20 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、KL5C80A20 外部に出力されているリード/ライト信号 (EMRD₀, EMWR₀, EIORD₀, EIOWR₀) 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、KL5C80A20 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 バイトを 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・開発部もしくは弊社代理店まで問い合わせ下さい。

2. ブロック図

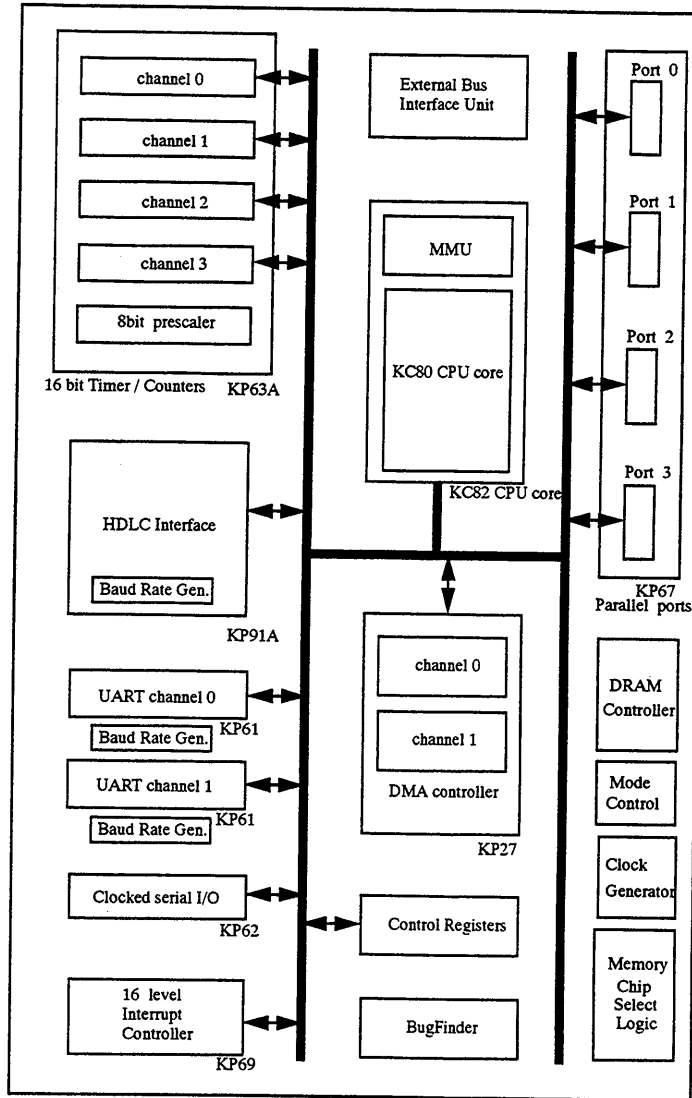


図 2-1 KL5C80A20 のブロック図

3. 端子説明

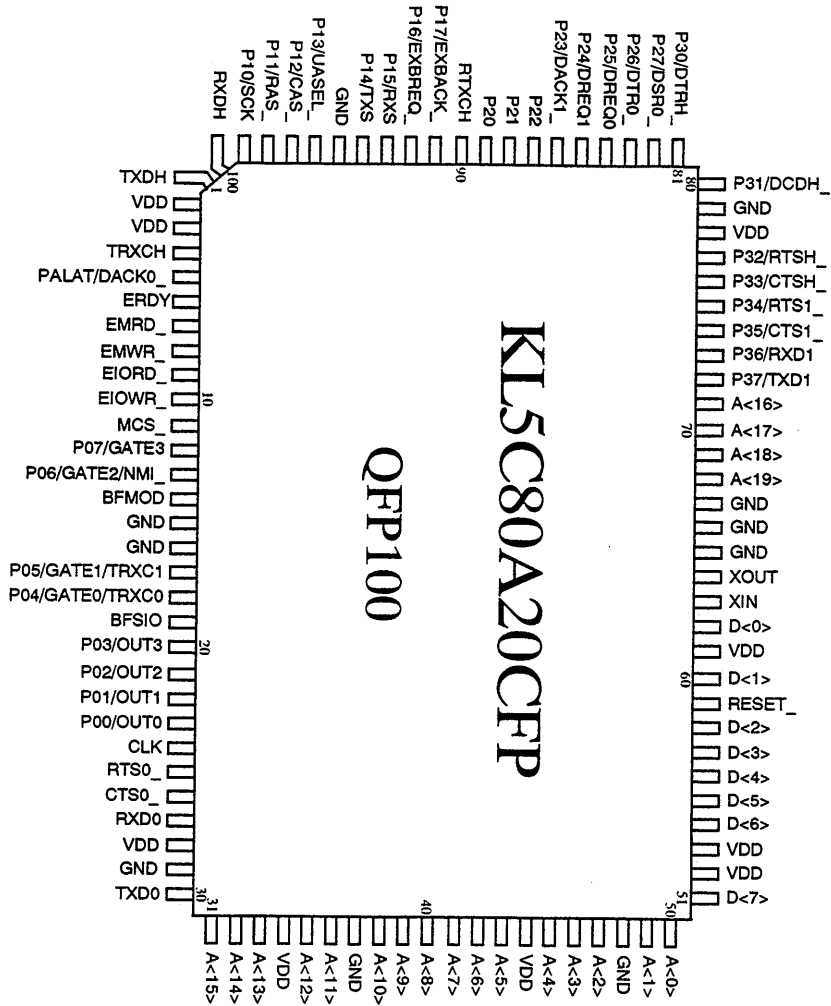


図 3-1 ピン配置図 (上面図)

KL5C80A20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。端子機能の選択はシステムコントロールレジスタの設定により行います。システムコントロールレジスタについては13章を参照して下さい。

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	外部メモリアドレスリード信号。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	外部メモリアドレスライト信号。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。KL5C80A20 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。“L”でウェイトを挿入します。
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 KL5C80A20 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。 立ち下がリエッジ検知。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 要求アッ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。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。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	UART 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	UART 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	UART 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	UART 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。
DSR0_	I	UART 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	UART 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。
DTR0_	O	UART 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
SCK	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。
RXS	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。
TXS	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。
TXDH	O	HDLC シリアルインターフェース・送信データ出力端子 送信データが送信クロックにしたがって、シリアルに出力されます。 詳細は 11 章を参照して下さい。
RXDH	I	HDLC シリアルインターフェース・受信データ入力端子 受信データがシリアルに入力され、受信クロックでサンプルされます。 詳細は 11 章を参照して下さい。
CTSH_	I	HDLC シリアルインターフェース・クリア・トゥ・SEND 入力端子 Auto Enable に設定された時 (WR4, D0 = '1') は送信イネーブルとして機能します。Auto Enable に設定しないときは汎用入力ポートとして機能します。 詳細は 11 章を参照して下さい。
DCDH_	I	HDLC シリアルインターフェース・データ・キャリア・ディテクト入力端子 Auto Enable に設定された時 (WR4, D0 = '1') は受信イネーブルとして機能します。Auto Enable に設定しないときは汎用入力ポートとして機能します。 詳細は 11 章を参照して下さい。
RTSH_	O	HDLC シリアルインターフェース・リクエスト・トゥ・SEND 出力端子 モデムに対するリクエスト・トゥ・SEND 信号として用いますが、汎用出力ポートとして使用することもできます。詳細は 11 章を参照して下さい。
DTRH_	O	HDLC シリアルインターフェース・データ・ターミナル・レディ出力端子 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、汎用出力ポートとして使用することもできます。詳細は 11 章を参照して下さい。

端子名	I/O	機能説明
RTXCH	I	HDLCシリアルインターフェース・送信/受信クロック入力端子 WR6の設定にしたがって、送信クロックあるいは受信クロック入力として 使用します。また、ポーレートジェネレータ、DPLLのソース・クロックと して使用することも出来ます。詳細は11章を参照して下さい。
TRXCH	I/O	HDLCシリアルインターフェース・送信/受信クロック入出力端子 WR6の内容にしたがって、送信クロックあるいは受信クロック入出力とし て使用します。詳細は11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12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12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12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12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、16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 SCR5 のビット 7~4 で行います。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。SCR5 はリセット時は‘00H’になります。

SCR5(I/O アドレス = 1FH)

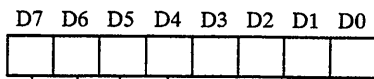


表 4-1 外部 I/O・ウェイト・コントロール

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。

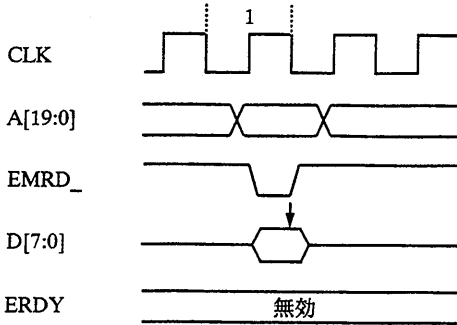
4.2 外部バスインターフェース・ユニットの動作

KL5C80A20 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。KL5C80A20 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 KL5C80A20 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)

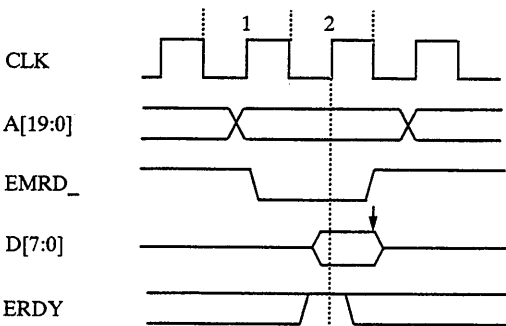
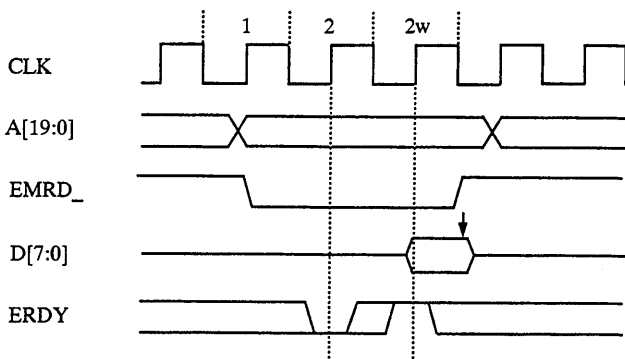
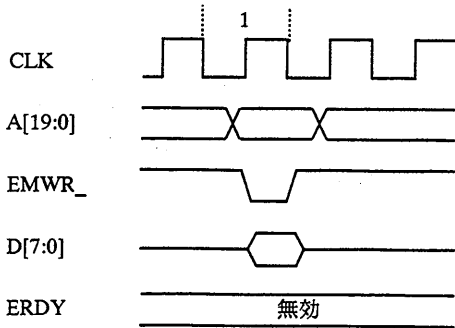


図 4-2 外部メモリリード (ウェイト・ステート・コントローラを 1 ウェイトに設定した時)



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ず。

図 4-4 外部メモライト (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)

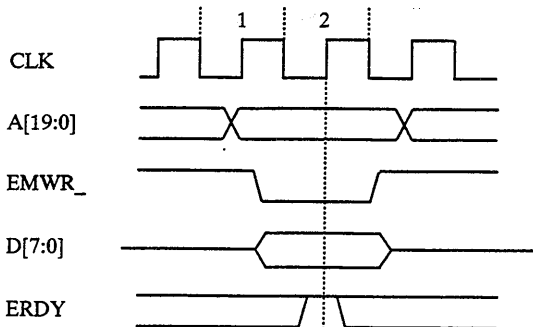
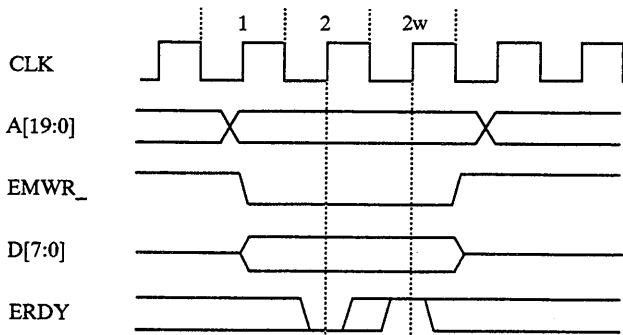
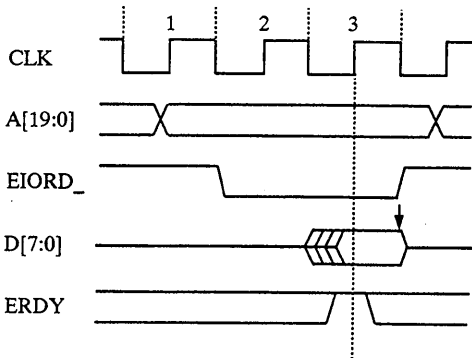


図 4-5 外部メモライト (ウェイト・ステート・コントローラを 1 ウェイトに設定した時)



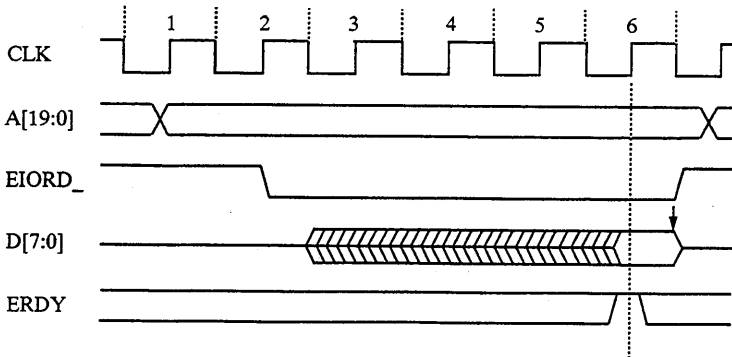
2w: 外部ウェイト要求によるウェイトサイクル

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)



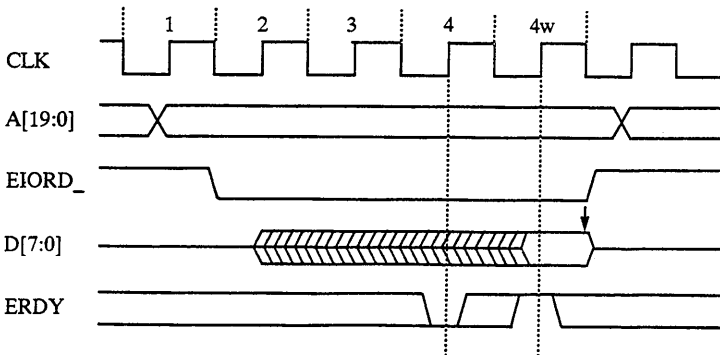
外部 I/O リードのウェイト設定は SCR4 で 1 ウェイト～ 4 ウェイトまで設定できます。1 ウェイトでは左の図にあるように 3 クロックで外部 I/O をリードします。2 ウェイト、3 ウェイトの場合はサイクル 2 の後にそれぞれ 1 クロック、2 クロックのウェイトサイクルが挿入されます。SCR の設定に関係なく内部 I/O アクセスは常に 0 ウェイトです。また、内部 I/O にリードする場合は EIORD_ は “H” のままです。

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)



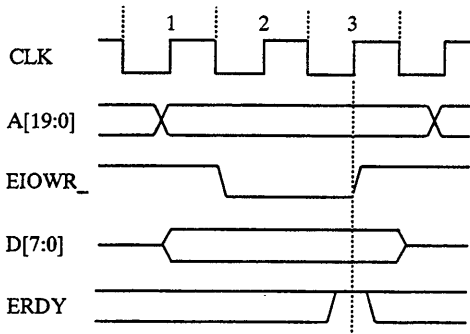
外部 I/O リードの 4 ウェイト設定は 1 ウェイト～ 3 ウェイト設定の外部 I/O リードと異なり、EIORD_ の立ち下がりが半クロック遅れます。アドレスのセットアップの大きい外部 I/O デバイスを接続する場合に御利用下さい。

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)



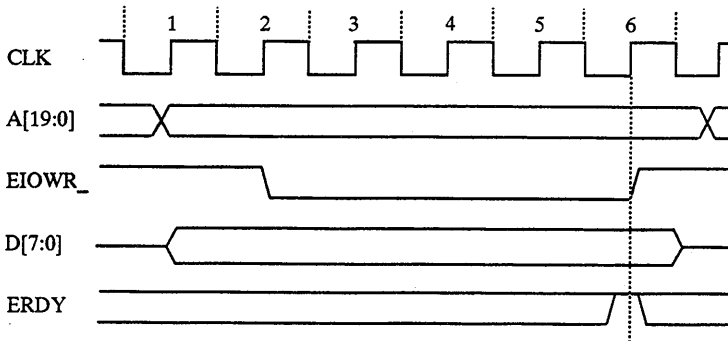
4w: 外部ウェイト要求によるウェイトサイクル

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)



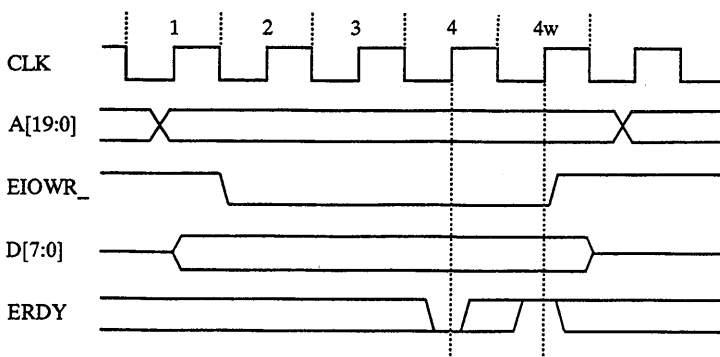
外部 I/O ライトのウェイト設定は SCR4 で 1 ウェイト～ 4 ウェイトまで設定できます。1 ウェイトでは左の図にあるように 3 クロックで外部 I/O をライトします。2 ウェイト、3 ウェイトの場合はサイクル 2 の後にそれぞれ 1 クロック、2 クロックのウェイトサイクルが挿入されます。SCR の設定に関係なく内部 I/O アクセスは常に 0 ウェイトです。また、内部 I/O にライトする場合は EIORW_ は “H” のままです。

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、EIORW_ 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

KL5C80A20 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、KL5C80A20 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 SCR4 と SCR0 で行います。DRAM 用のアドレス領域は '80000H ~ BFFFFH' です。SCR4 の D5 の設定により 'C0000H ~ FFFFFH' 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。

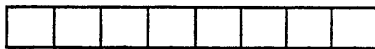
1Mバイト拡張時 : A17,A16, A7~A0をDRAMに接続してください。

4Mバイト拡張時 : A18 ~ A16, A7 ~ A0 を DRAM に接続してください。

尚、DRAM は × 4 ビット構成でロウ・カラムアドレス幅が同一のタイプを使用して下さい。

SCR4 (I/O アドレス = 1EH)

D7 D6 D5 D4 D3 D2 D1 D0



Refresh Rate

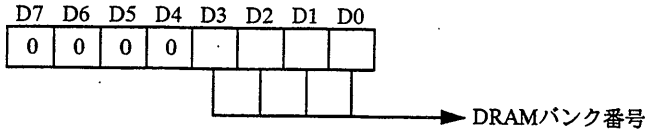
D2	D1	D0	リフレッシュ要求発生間隔
0	0	0	128 clock 毎
0	0	1	88 clock 毎
0	1	0	64 clock 毎
0	1	1	44 clock 毎
1	0	0	32 clock 毎
1	0	1	22 clock 毎
1	1	0	16 clock 毎
1	1	1	11 clock 毎

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_
 端子 97 は CAS_
 端子 98 は RAS_ として機能します。
 (UASEL_ は ICE 使用時に使用します。)

SCR0(I/Oアドレス = 1BH)

KL5C80A20 で DRAM 領域として使用できるのは '80000H ~ FFFFFH' のみですが、SCR0 でコントローラで 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。KL5C80A20 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、EMRD₀、EMWR₀、EIORD₀、EIOWR₀、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。

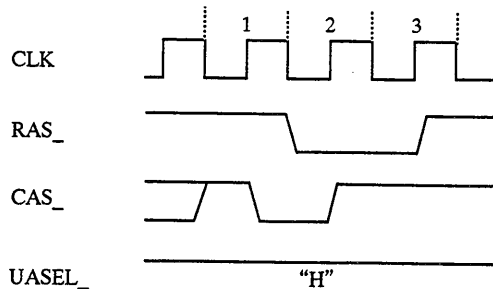
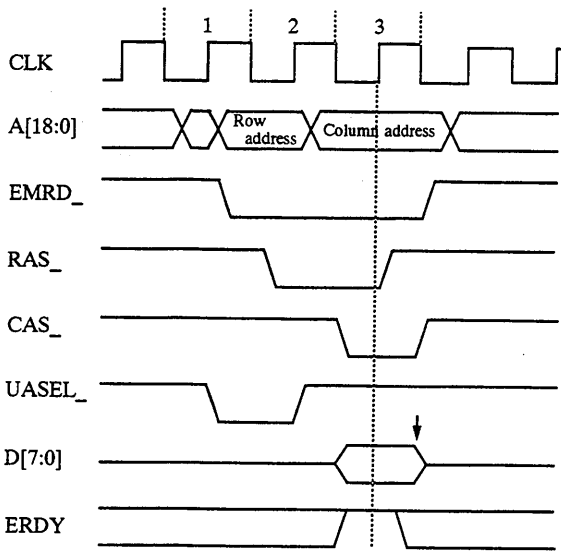
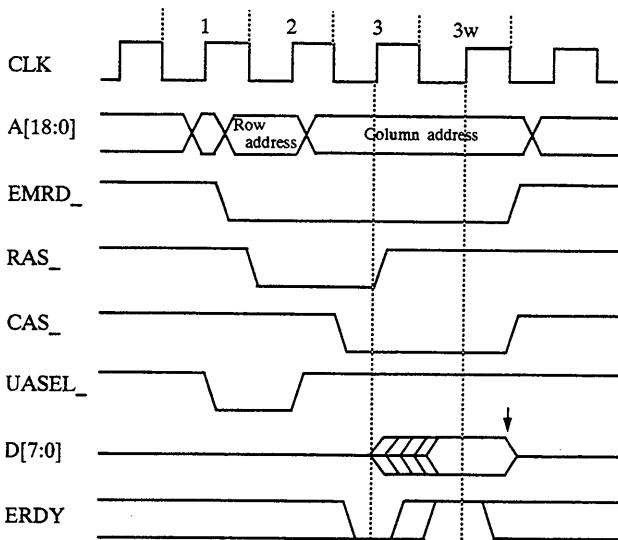


図 4-13 DRAM リフレッシュ



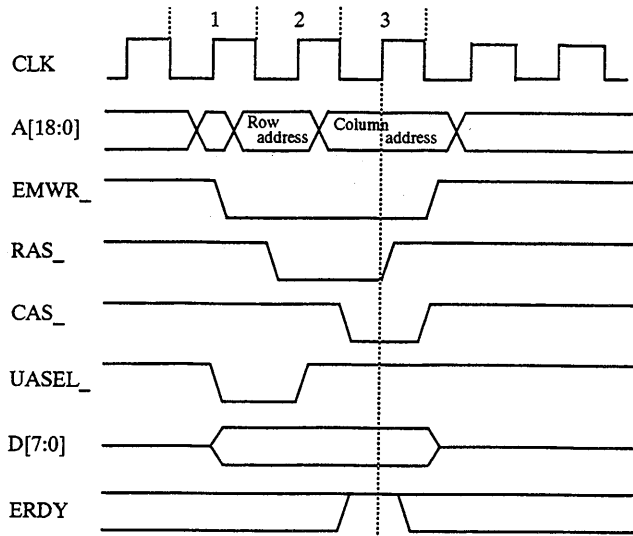
DRAM リードサイクルは SCR の設定 (メモリアクセスのウェイト設定) に関係なく 3 クロックのパスサイクルです。アクセスタイムが足りない場合は外部ウェイト要求を ERDY 端子から入れてください。ERDY 入力は SCR の設定に関係なく有効です。

図 4-14 DRAM リード (外部ウェイトを追加しない時)



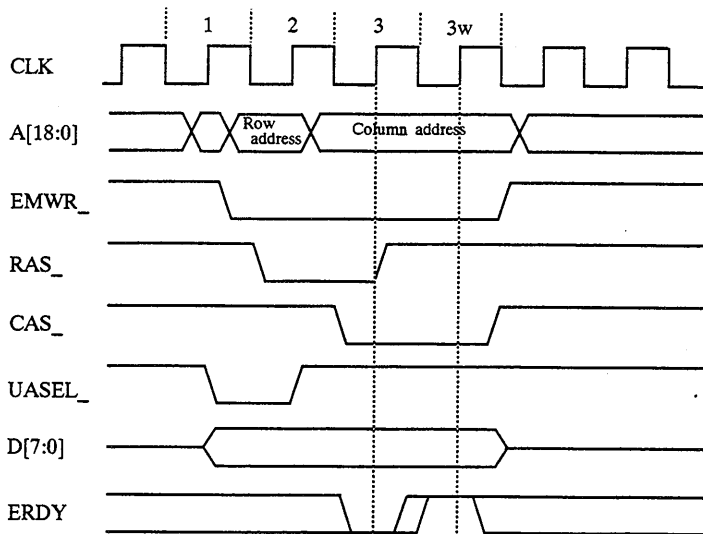
3w: 外部ウェイト要求によるウェイトサイクル

図 4-15 DRAM リード (外部ウェイトを 1 ウェイト追加した時)



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。

図 4-16 DRAM ライト (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 4-17 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要求アクリッジ信号(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。

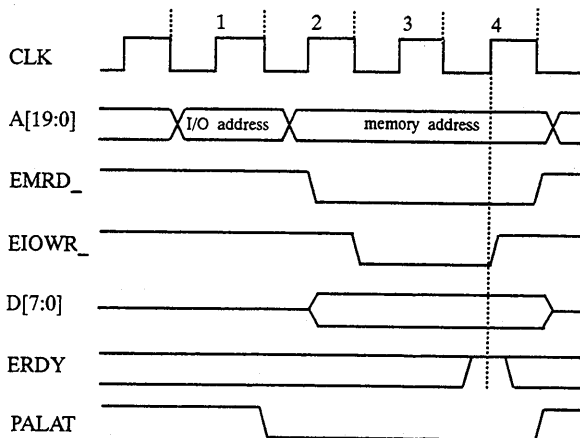
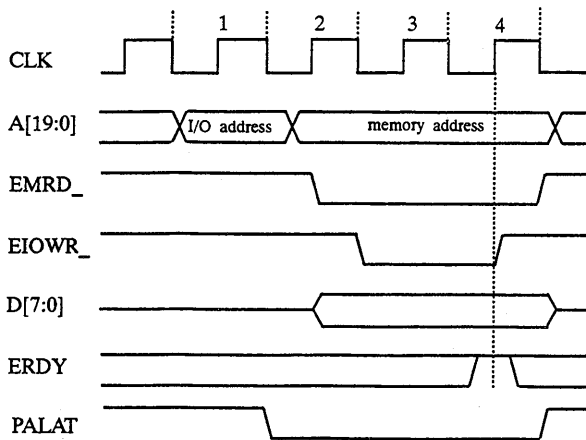


図 4-18 DMA 転送の例 (memory to 外部 I/O、メモリは 0 ウェイト、I/O は 1 ウェイト設定時)



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 4-19 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)

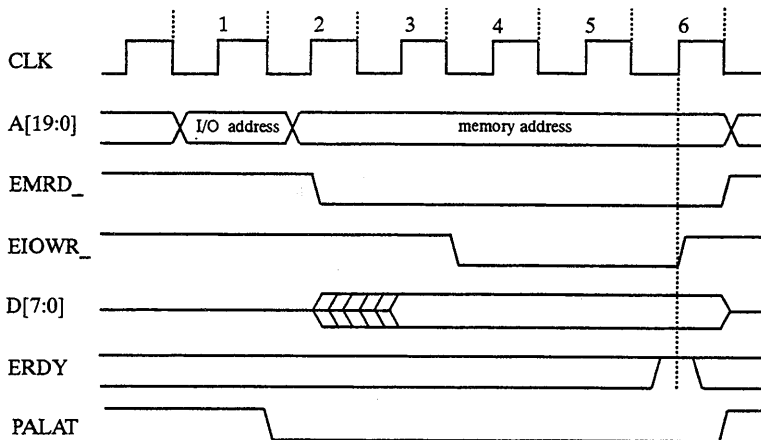


図 4-20 DMA 転送 (memory to 外部 I/O, メモリは 1 ウェイト、I/O は 2 ウェイト設定時)

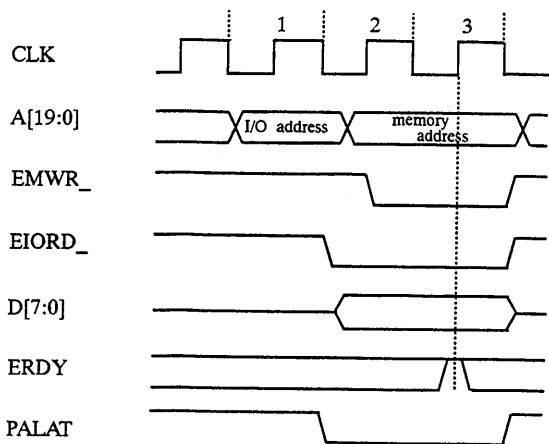


図 4-21 DMA 転送 (外部 I/O to memory, メモリは 0 ウェイト、I/O は 1 ウェイト設定時)

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。

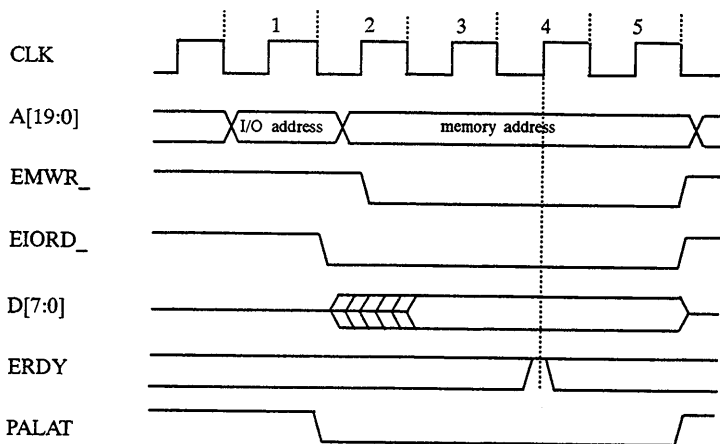


図 4-22 DMA 転送 (外部 I/O to memory, メモリは 1 ウェイト、I/O は 2 ウェイト設定時)

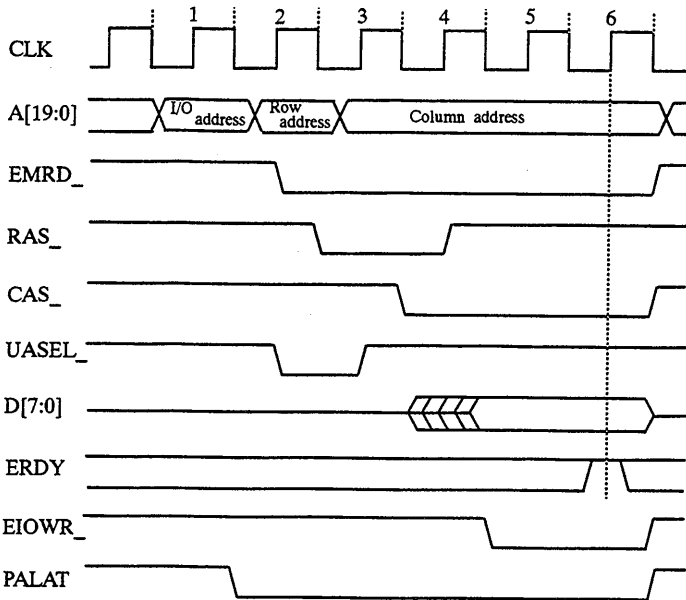


図 4-23 DMA 転送 (DRAM to 外部 I/O, I/O は 1 ウェイト, 外部ウェイトを追加しない時)

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。

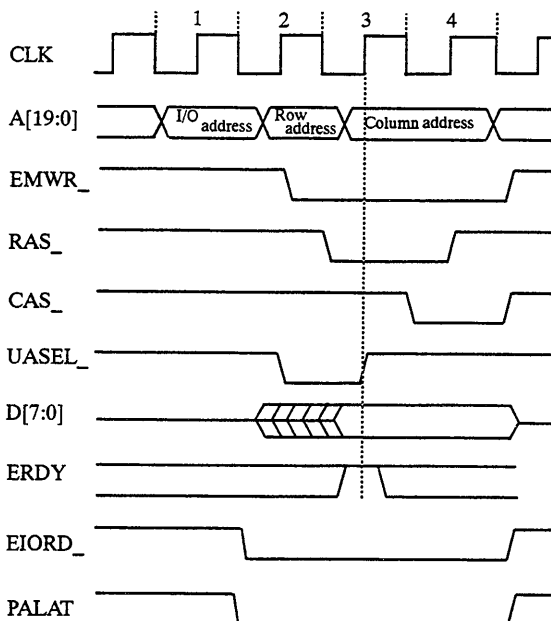


図 4-24 DMA 転送 (外部 I/O to DRAM, I/O は 1 ウェイト, 外部ウェイトを追加しない時)

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。

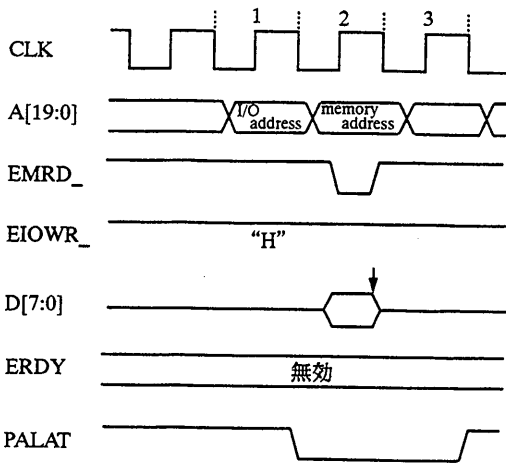


図 4-25 DMA 転送 (memory to 内部 I/O, メモリ 0 ウェイト設定時)

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。

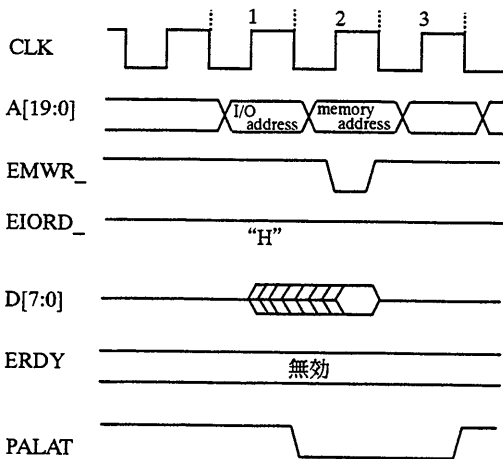


図 4-26 DMA 転送 (内部 I/O to memory, メモリ 0 ウェイト設定時)

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と組み合わせて使用することで、効率良くASICマイコン化することが可能です。そして、従来の非同期バス接続に比べてシステム性能を飛躍的に向上させることが可能です。

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。

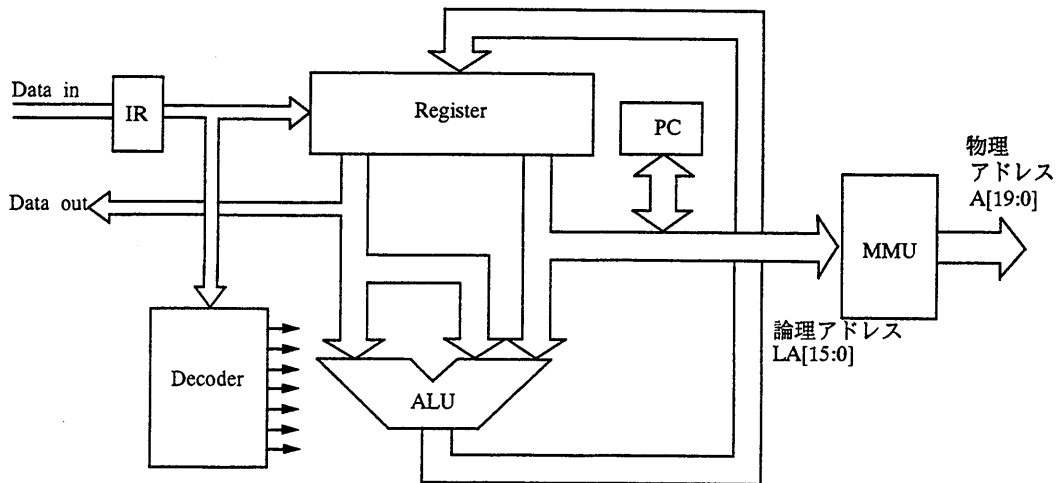


図 5-1 KC82全体ブロック図

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。
IORD_	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用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期間、CPUはウェイト状態を続けます。この入力には外部バス・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_入力が入力アクティブならば、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_	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

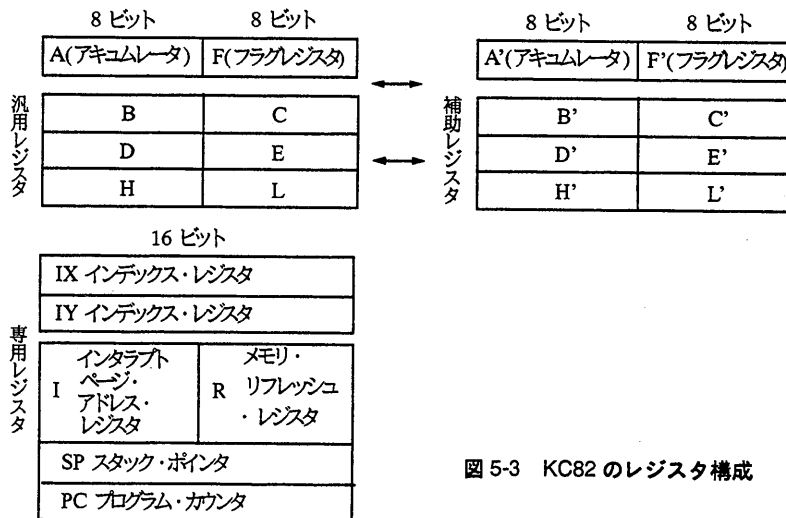


図 5-3 KC82 のレジスタ構成

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

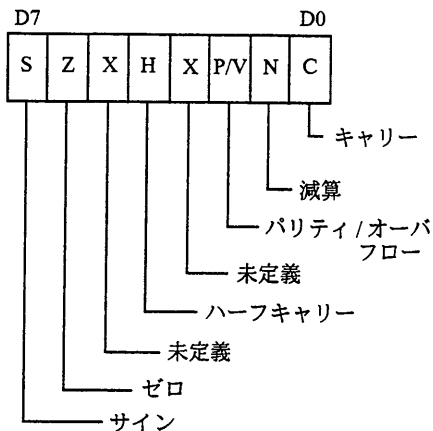


図 5-4 フラグレジスタ

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/M)

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、 Borrow 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、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ます。尚、最小 1 クロックサイクルです。

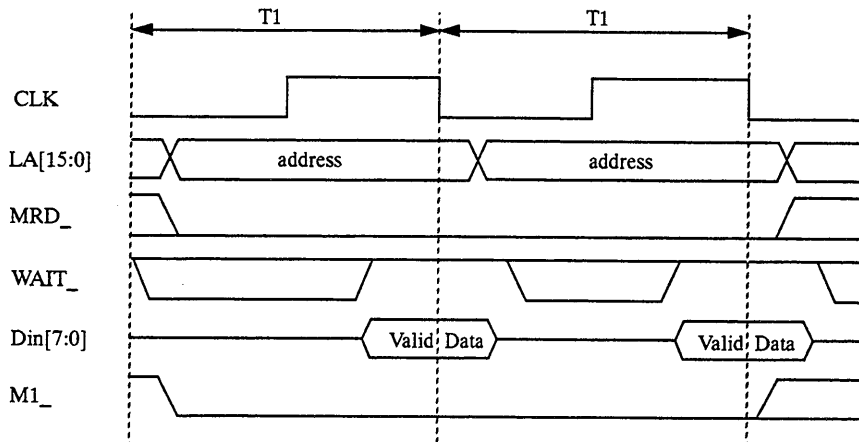


図 5-5 オペコードフェッチサイクル (内部バス・サイクル)
(上図は、0 WAIT の連続リードを表す)

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。

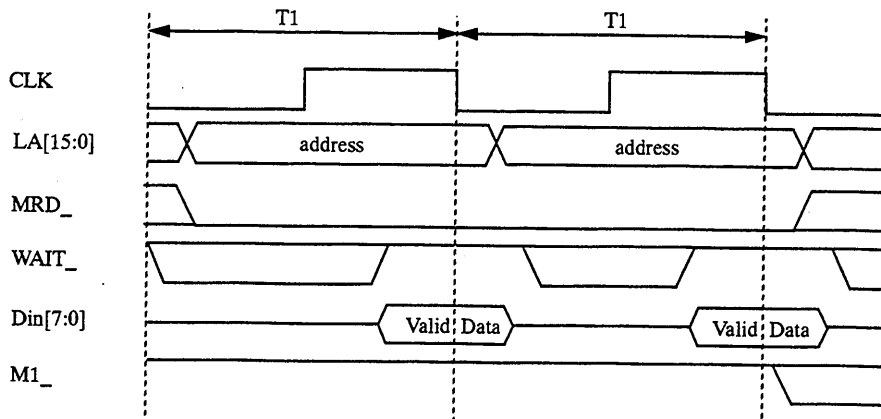


図 5-6 メモリリードサイクル (0 wait)
(上図は、連続の 0 wait リード動作のタイミング)

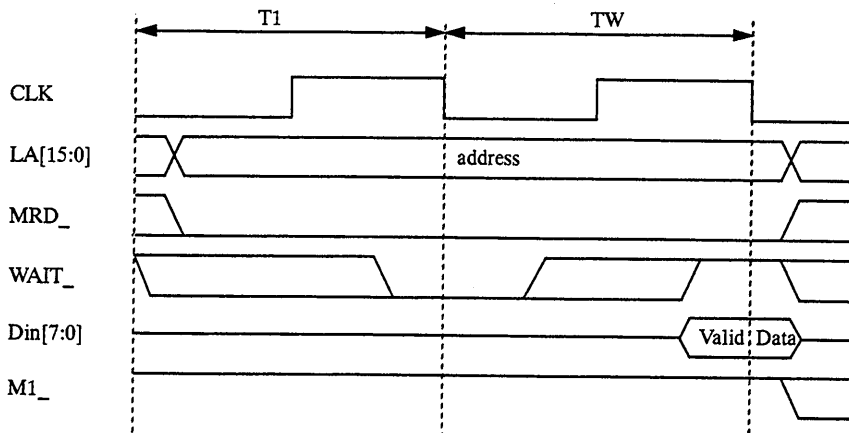


図 5-7 メモリリードサイクル (1 wait)
(上図は、1 wait リード動作のタイミング)

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。

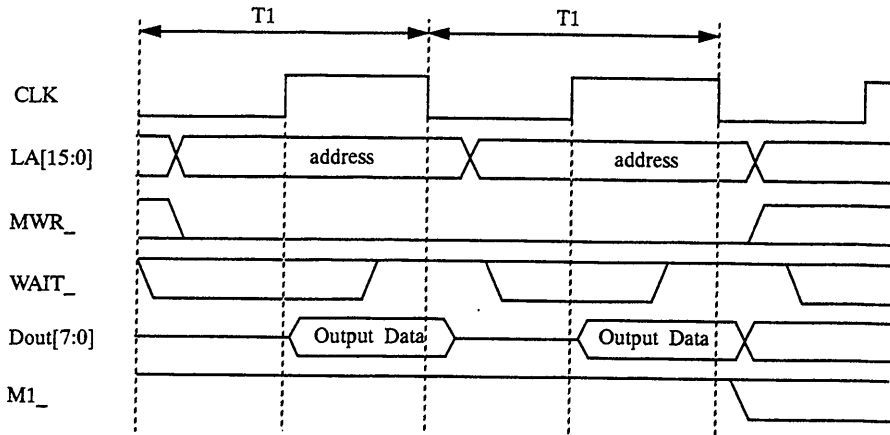


図 5-8 メモリライトサイクル (0 wait)
 (上図は、連続 0 wait ライト動作のタイミング)

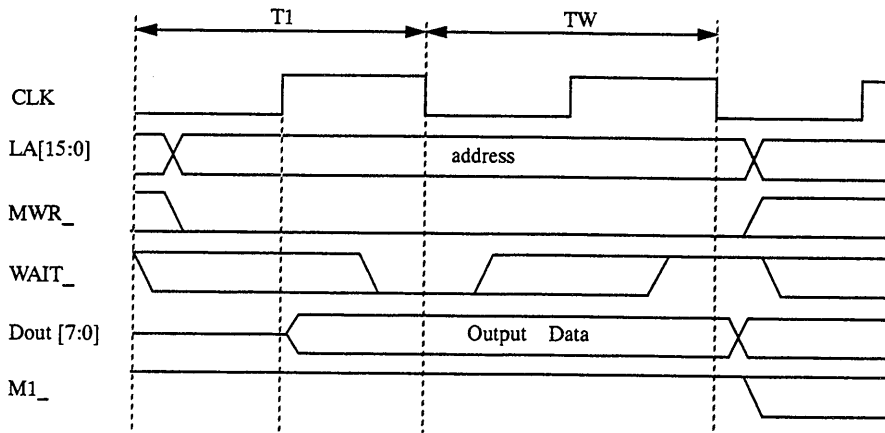


図 5-9 メモリライトサイクル (1 wait)
 (上図は、1 wait ライト動作のタイミング)

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。

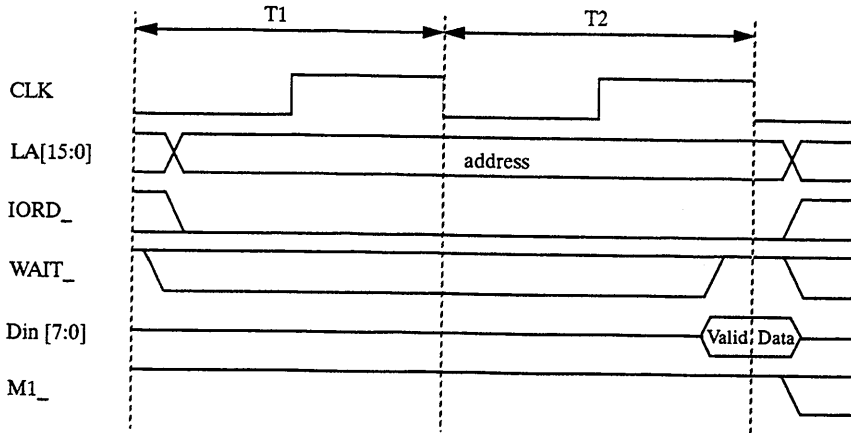


図 5-10 I/O リードサイクル (0 wait)
 (上図は、0 wait 動作のタイミング)

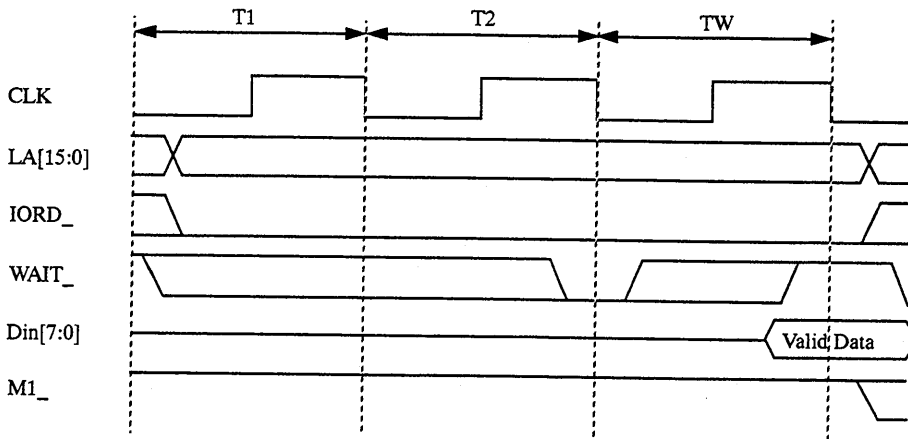


図 5-11 I/O リードサイクル (1 wait)
 (上図は、1 wait 動作のタイミング)

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。

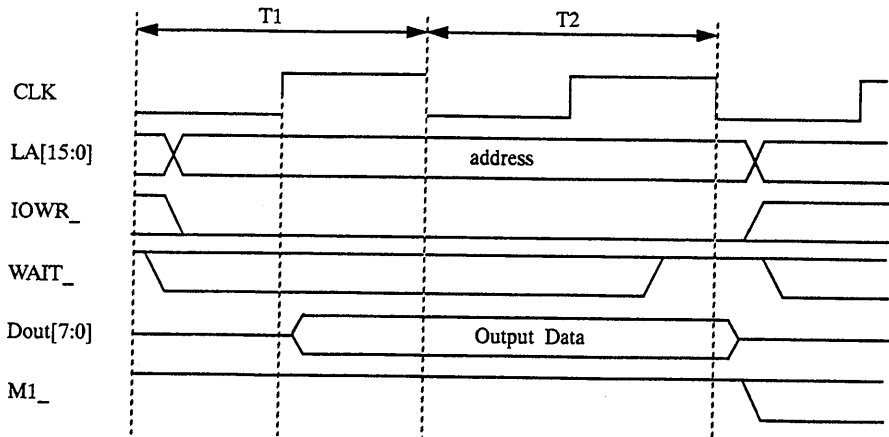


図 5-12 I/O ライトサイクル (0 wait)
 (上図は、0 wait 動作タイミング)

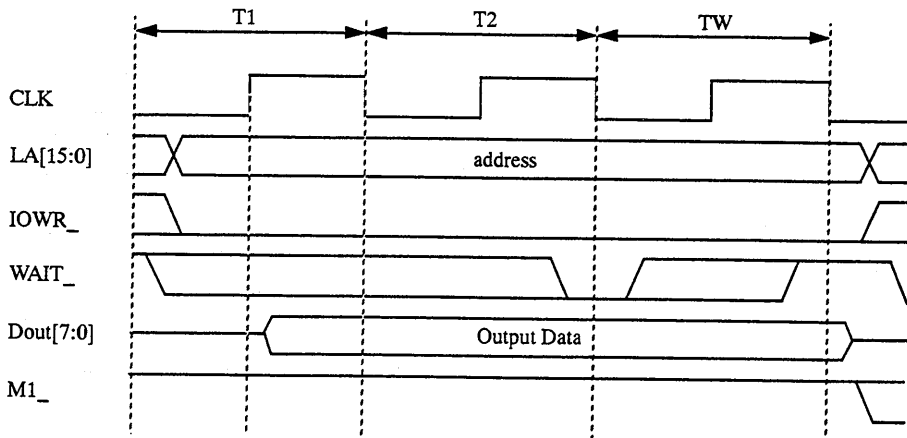


図 5-13 I/O ライトサイクル (1 wait)
 (上図は、1 wait 動作タイミング)

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

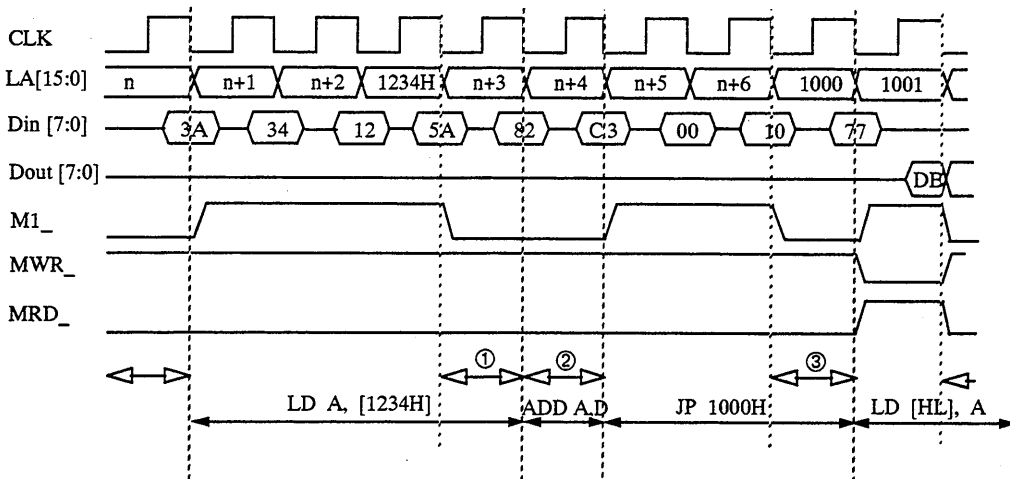


図 5-15 プリフェッチの様子 (←→ がプリフェッチサイクルを表す。また 1000H 番地、1234H 番地にはそれぞれ '77H', '5AH' が格納されており、D レジスタは '81H' とする)

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③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に、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④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④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 (MRD₀, MWR₀, IORD₀, IOWR₀) 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、BACK₀ 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" の期間ずっと継続します。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) INT₀ 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。図 5-16 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

(‘—’は不変を表す)

図 5-16 イネーブル・フリップフロップの状態図

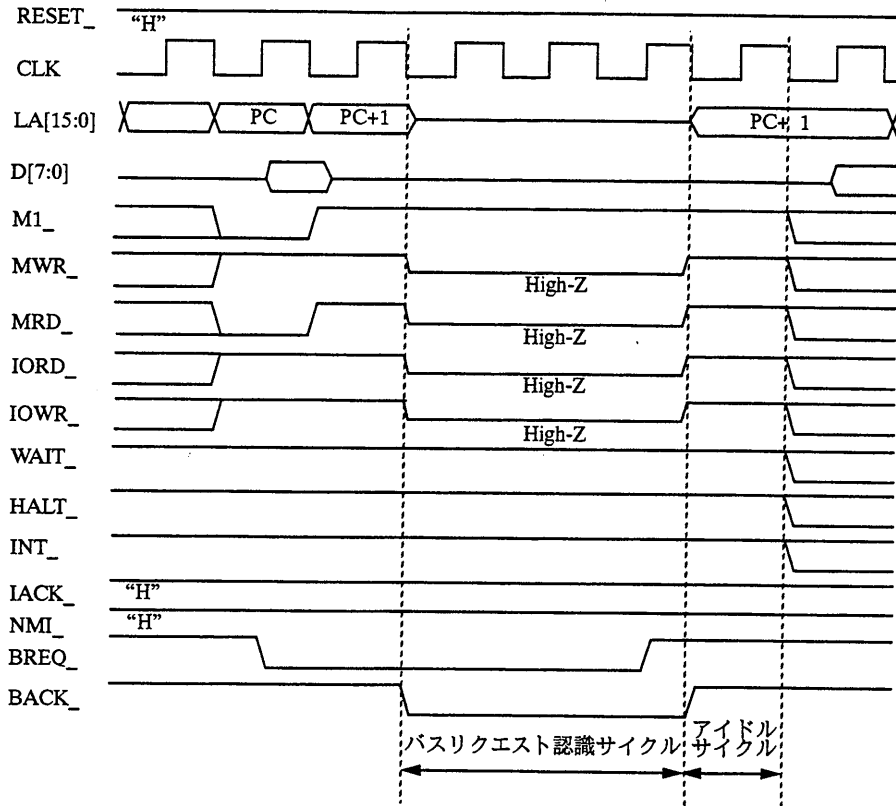


図 5-17 バスリクエストタイミング

マスカブル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_ の立ち下がりが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。

マスカブル割り込みのモード

マスカブル割り込みには 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。

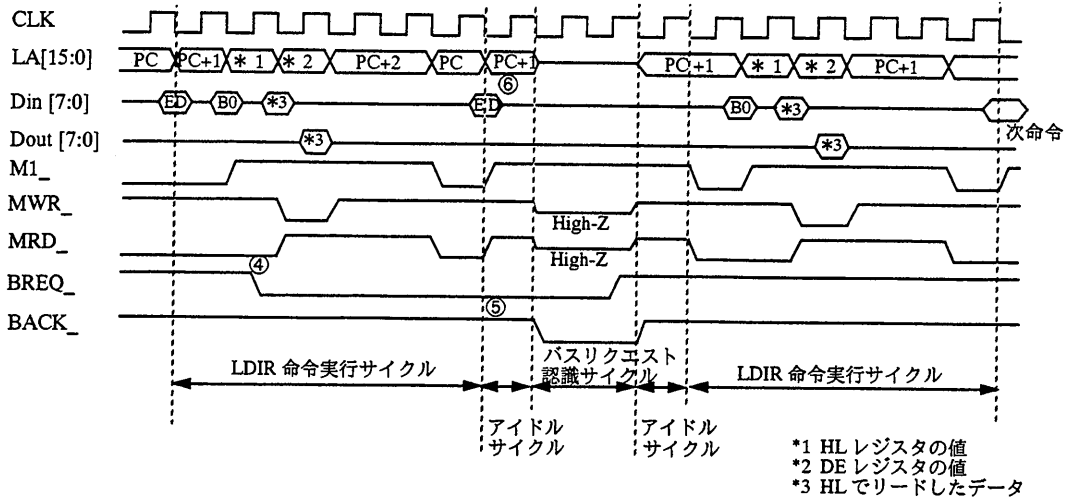


図 5-18 バスリクエスト受け付けタイミング

(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_の立ち下がり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_の立ち下がりエッジがあれば、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。

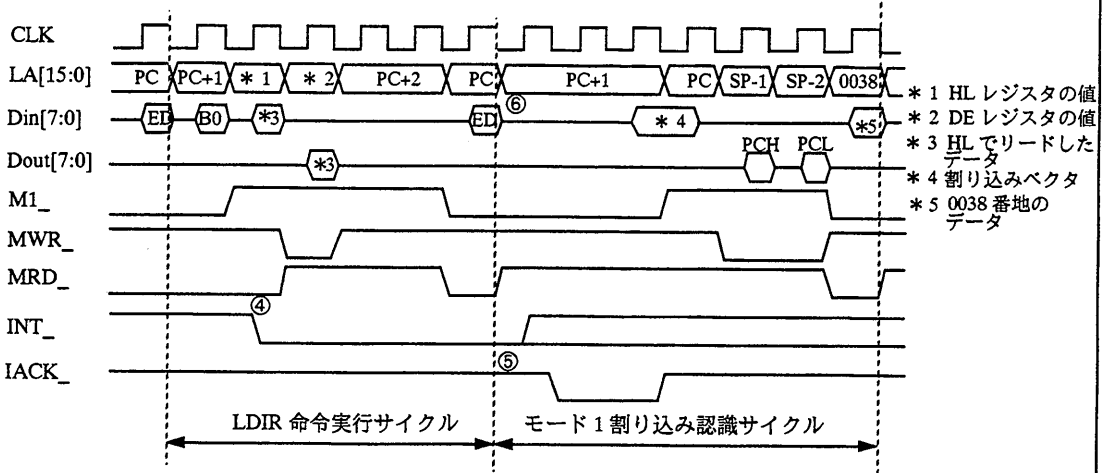


図 5-19 割り込み受け付けタイミング (モード 1)

(上図では、LDIR 命令実行中に INT_ 入力^④が発生し、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)

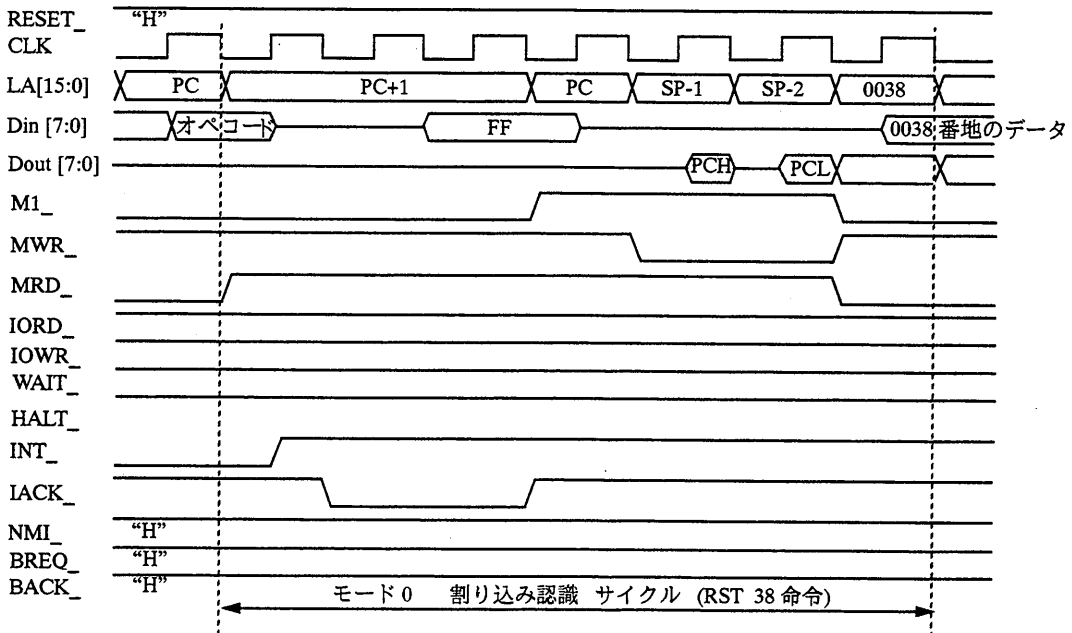


図 5-20 モード 0 タイミング

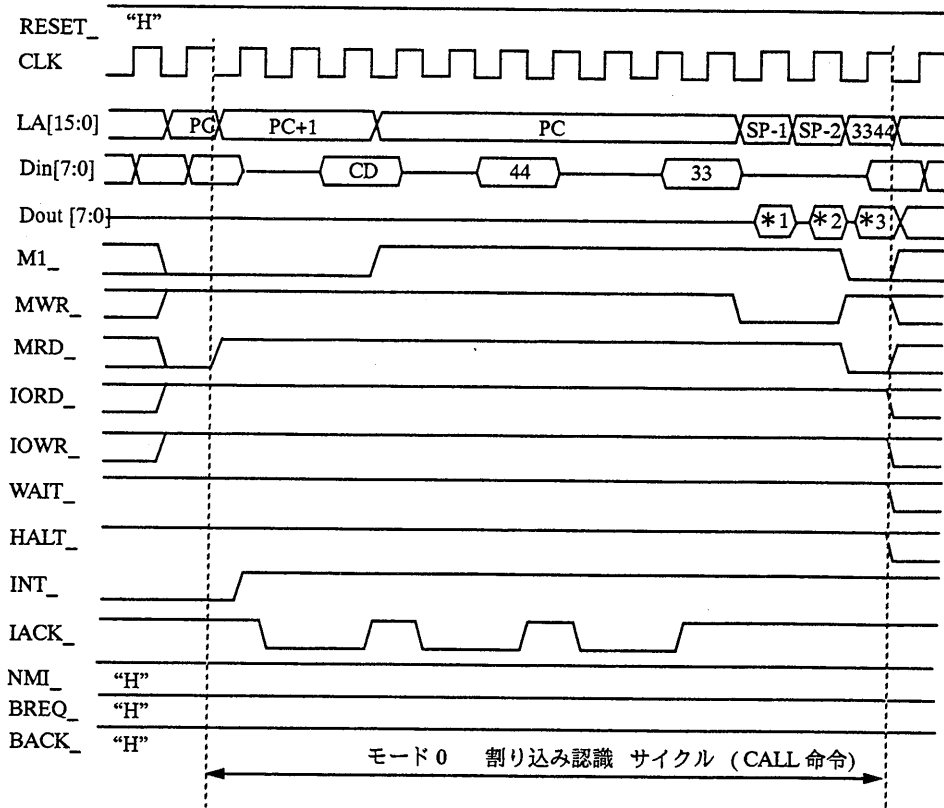


図 5-21 モード0 タイミング

- * 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。マスク割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' でありマスク割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ク割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。

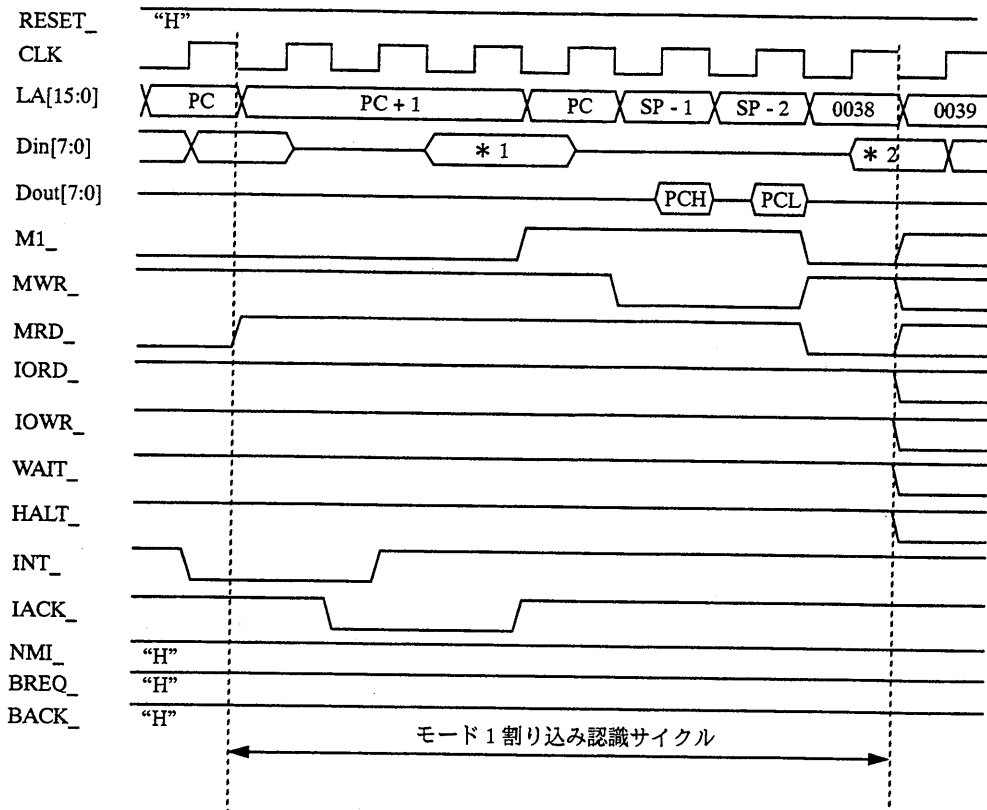


図 5-22 モード1 タイミング

* 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。

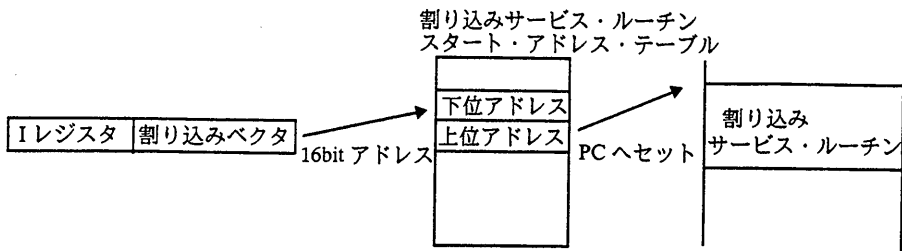
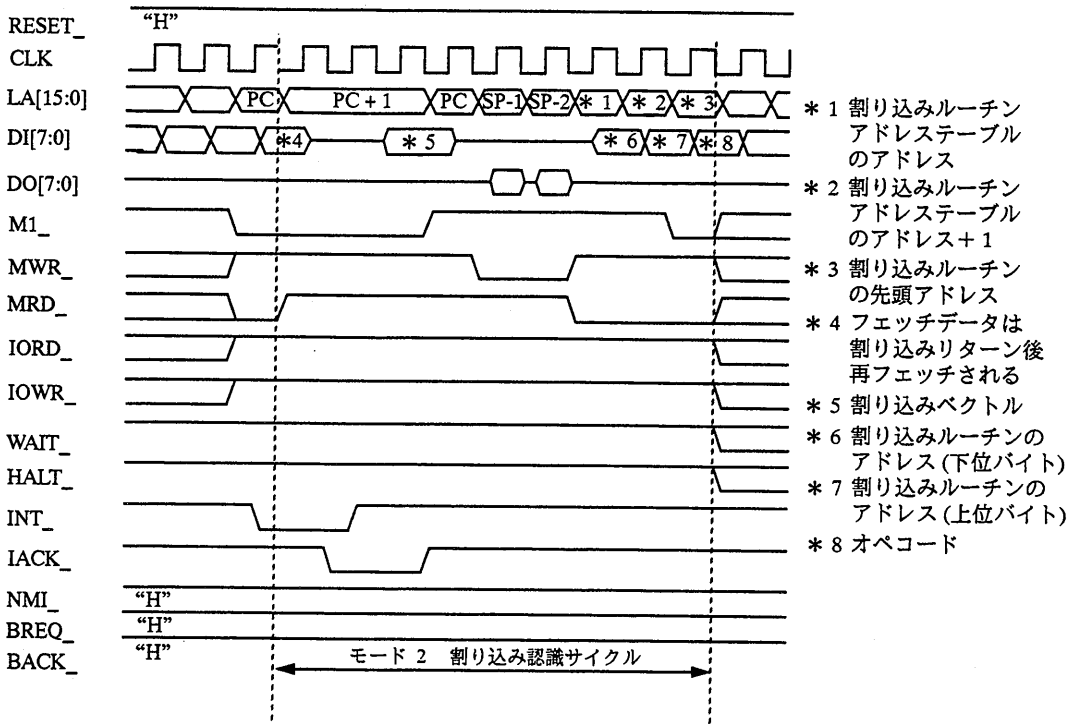


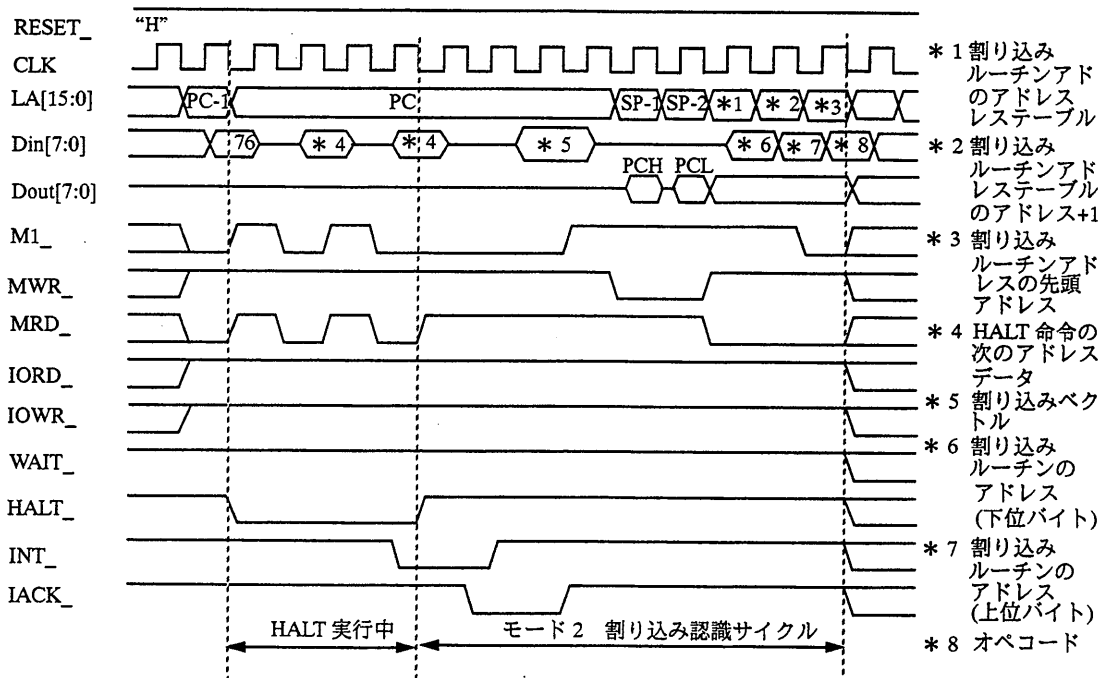
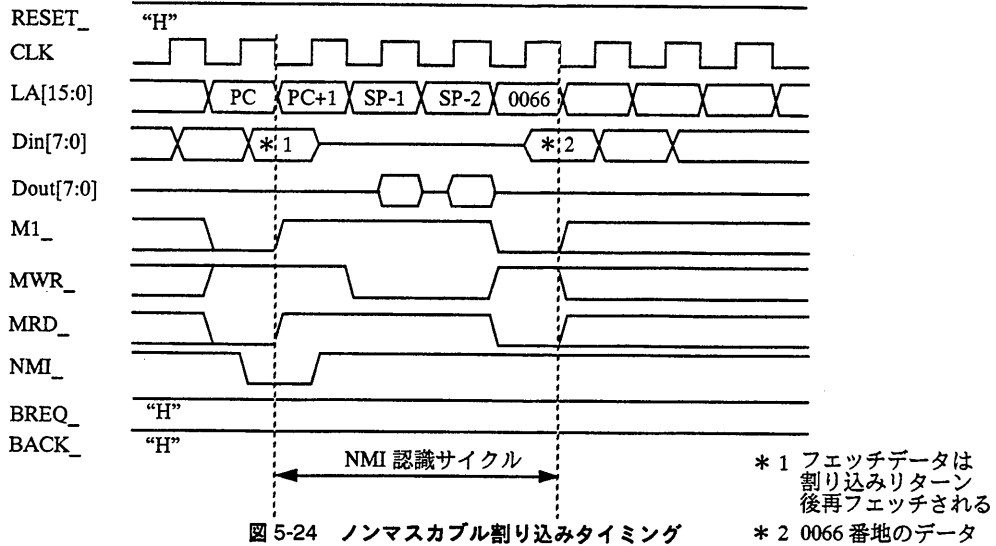
図 5-23-A モード 2 割り込み



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。



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

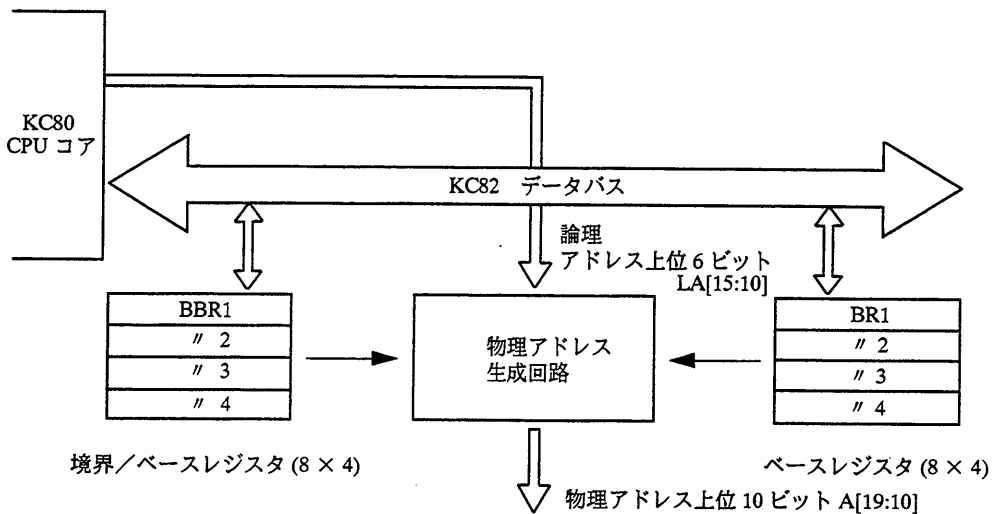


図 5-26 ブロック図

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

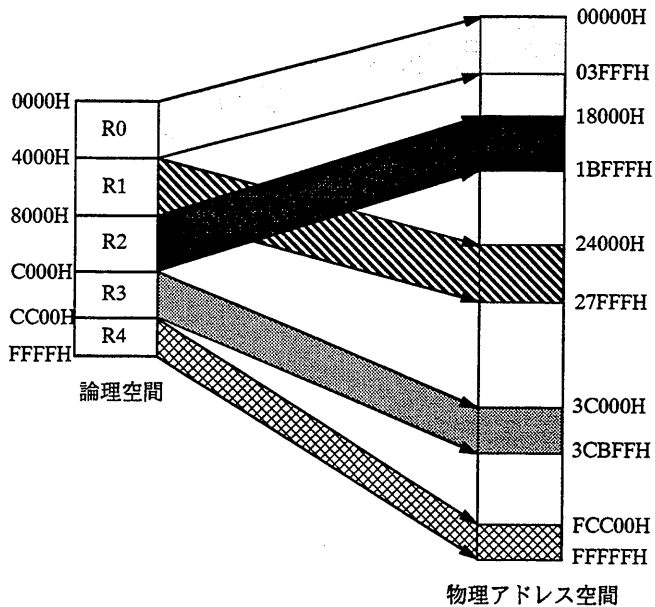
論理アドレス空間は、図 5-28 に示すように 5 つの領域 (R0~R4) に分割されています。この 5 つの論理領域は論理アドレス境界値 (B0<5:0>~ B4<5:0>) で決められています。Rn 領域の下限アドレスは $(B_{n+1}) \times 400H$ で決定され、Rn 領域の上限アドレスは $(B_{n+1}+1) \times 400H-1$ となります。(ただし、R0 領域の下限アドレスは常に 00000H、R4 領域の上限アドレスは常に FFFFH です。) これらの領域の物理空間上の配置は分割されたそれぞれの領域の物理アドレスベース An(10 ビット) を論理アドレスの上位 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

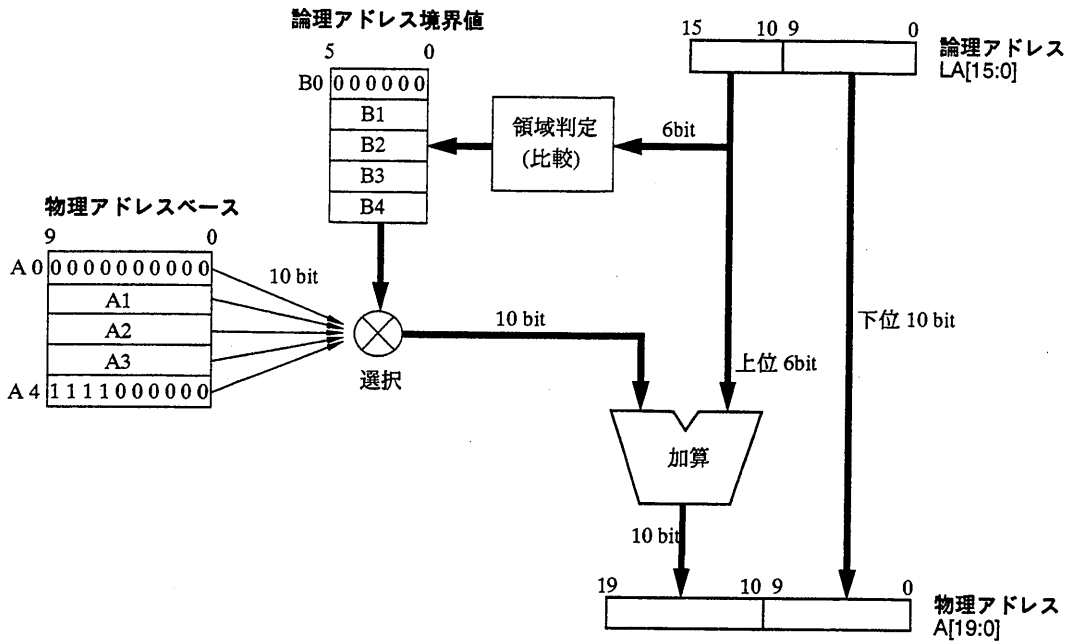


図 5-29 物理アドレス生成

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。

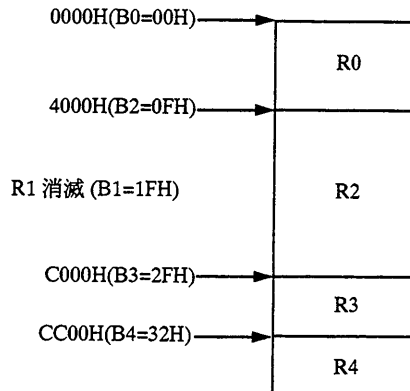


図 5-30 $B1 \geq B2$ としたときの例

(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) の使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ィンドウ
- R2 データバンクウィンドウ 1
- R3 データバンクウィンドウ 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

KL5C80A20は割り込みコントローラとしてKP69マクロセルを搭載しています。KP69は弊社CPU(KC80あるいはKC82)専用に関与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 = "1"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"1"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。

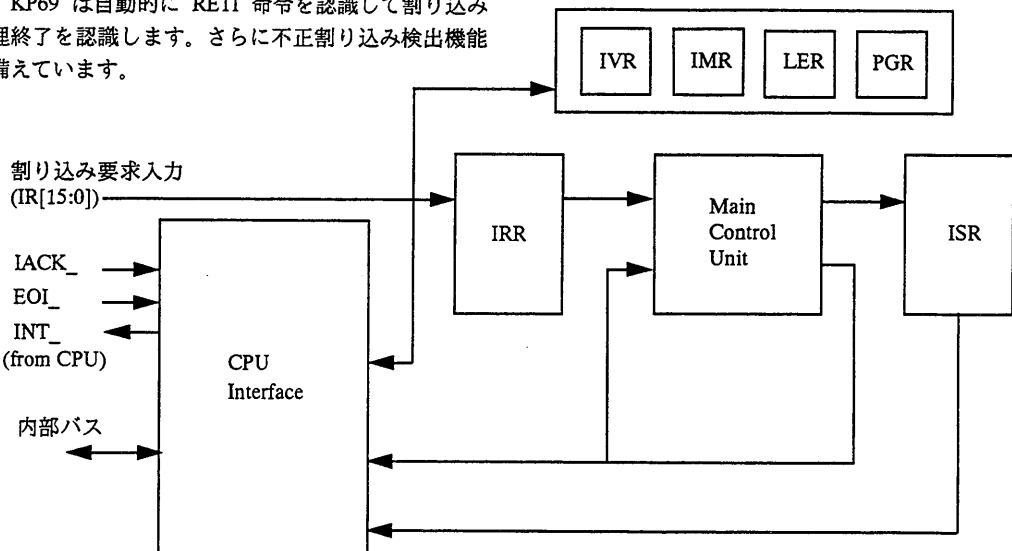


図 6-1 KP69 ブロック図

表 6-1 KL5C80A20 の割り込み

	割り込み要求元
IR[15]	外部端子 P20
IR[14]	タイマ/カウンタチャンネル1 割り込み/ 外部端子 P07
IR[13]	タイマ/カウンタチャンネル0 割り込み/ 外部端子 P21
IR[12]	HDLC シリアルインターフェース SPINT 出力
IR[11]	HDLC シリアルインターフェース RXINT 出力
IR[10]	HDLC シリアルインターフェース TXINT 出力
IR[9]	HDLC シリアルインターフェース ESINT 出力
IR[8]	UART チャンネル0 の RXRDY 出力 +BDET 出力 (+TXRDY 出力)
IR[7]	UART チャンネル0 の TXRDY 出力 / 外部端子 P07
IR[6]	クロック同期シリアル I/O 送受信割り込み / 外部端子 P07
IR[5]	DMA コントローラ チャンネル1 の DMTC 出力 / 外部端子 P21
IR[4]	DMA コントローラ チャンネル0 の DMTC 出力 / 外部端子 P22
IR[3]	タイマ/カウンタチャンネル3 / 外部端子 P22
IR[2]	UART チャンネル1 の RXRDY 出力 +BDET 出力 (+TXRDY 出力)
IR[1]	UART チャンネル1 の TXRDY 出力 / 外部端子 P21
IR[0]	タイマ/カウンタチャンネル2 / 外部端子 P22

各 IR 入力に複数の割り込み要求源があるものは SCR(システムコントロールレジスタ)により選択できるようになっています。SCR については 13 章を参照下さい。

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

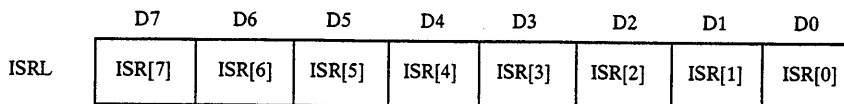
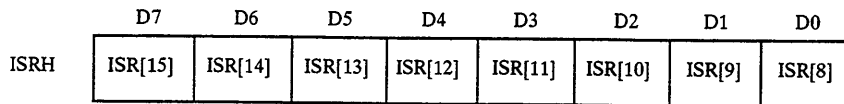
IACK_ 信号が返って来ると、受け付けた割り込み要求の割り込みレベルに該当するビットがセットされま

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。

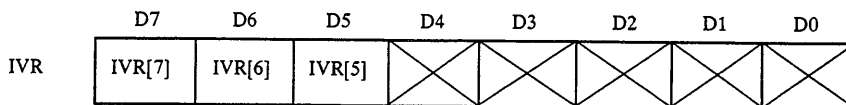
IVR (Interrupt Vector Register)

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”に行ってください。このレジスタは書き込み専用です。

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”に行ってください。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。

	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”に行ってください。このレジスタは書き込み／読み出し可能です。

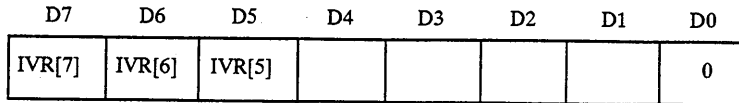
	D7	D6	D5	D4	D3	D2	D1	D0
IMRH	IMR[15]	IMR[14]	IMR[13]	IMR[12]	IMR[11]	IMR[10]	IMR[9]	IMR[8]

	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]

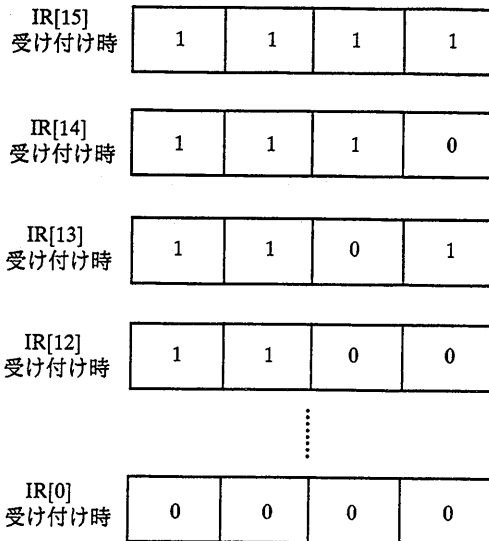
IMR[n]	意味
0	非マスク状態
1	マスク状態

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。以下に例を示します。

割り込み要求入力	グループ
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。

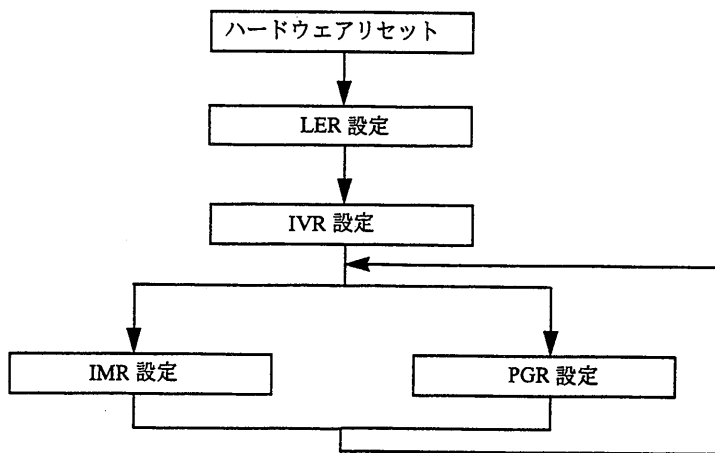


図 6-4 レジスタの設定シーケンス

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。

また、マスク状態であってもIR入力の立ち上がりがあった場合割り込み要求があったと認識し、マスク状態が解除されて受け付けられるまで保持されます。ご注意ください。

端子 12, 87 ~ 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 SCR2 の設定によりビット単位で変更できます。詳しくは13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。

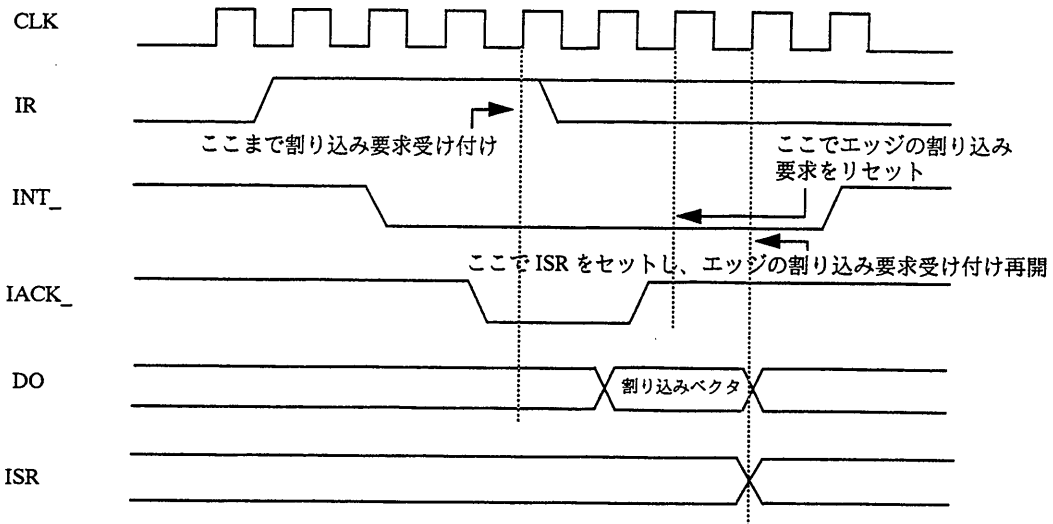


図 6-5 割り込み要求の受け付けタイミング

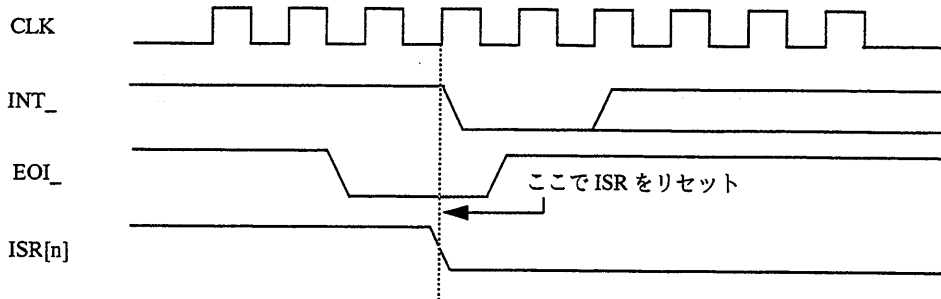


図 6-6 割り込み終了タイミング

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_ が “L” になり、それを受けて KC82 が IACK_ 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の」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_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_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 12, 87 ~ 89 の使い方）

外部端子 12, 87 ~ 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 12, 87 ~ 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[3], IR[4], IR[5], IR[6], IR[7], IR[13], IR[14]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 87,88 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 89 をパラレルポート出力として使う場合、IR[15] をマスクして下さい。
- (3) 外部端子 12, 88, 89 をパラレルポート出力として使う場合、該当 IR 入力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。

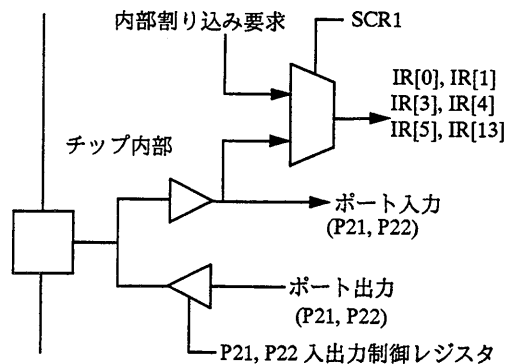


図 6-7 ピン 87, 88 の構造

7. DMA コントローラ

7.1 概要

KL5C80A20 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。
- ・DACK₀ 信号を出力可能なので、DACK₀ 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、DREQ1 には 13 章の SCR3 の設定により、端子 84、85、UART 及び HDLC シリアルインターフェースからの信号を接続することが可能です。

また、パラレルポート P16 を入力方向に設定することにより、端子 92 からの入力を EXBREQ_ として使用できます。この時、EXBACK_ は SCR3 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。この切替えは SCR2 で行います。また DACK1_ も SCR3 の設定によって端子 86 から出力することが出来ます。

詳細は 13 章を参照して下さい。

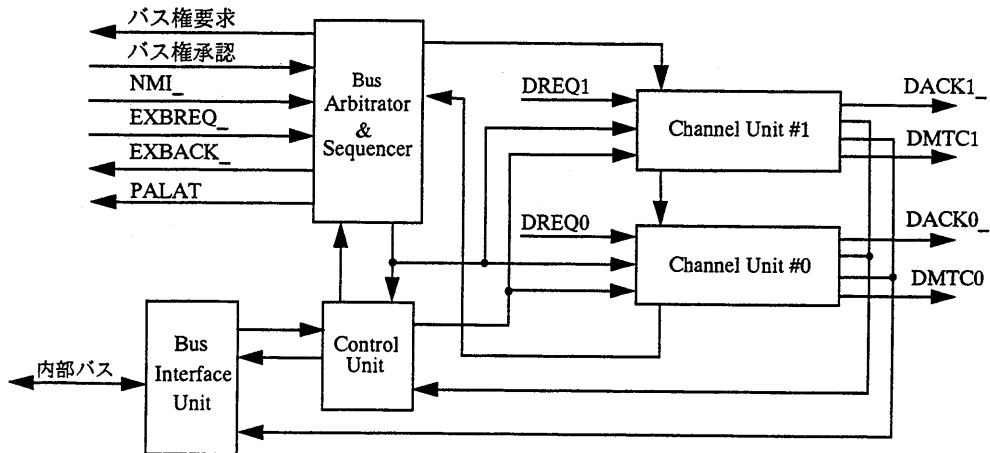


図 7-1 DMA コントローラブロック図

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。“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[4] に、DMTC1 が IR[5] 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。

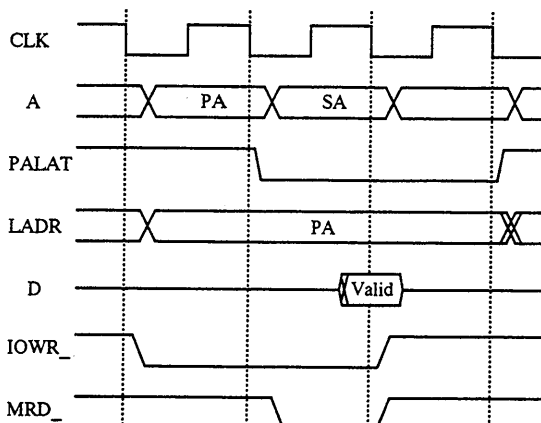


図7-2 KP27の基本DMA転送タイミング
(メモリ→チップ内部I/O)

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クセス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。

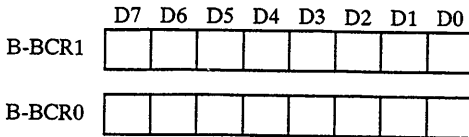
	D7	D6	D5	D4	D3	D2	D1	D0
C-PAR2	0	0	0	0				
C-PAR1								
C-PAR0								

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)

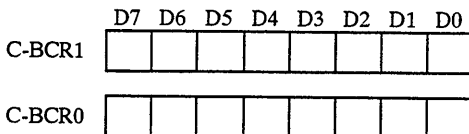
書き込み不可／読み出し可 16bit

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、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。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)

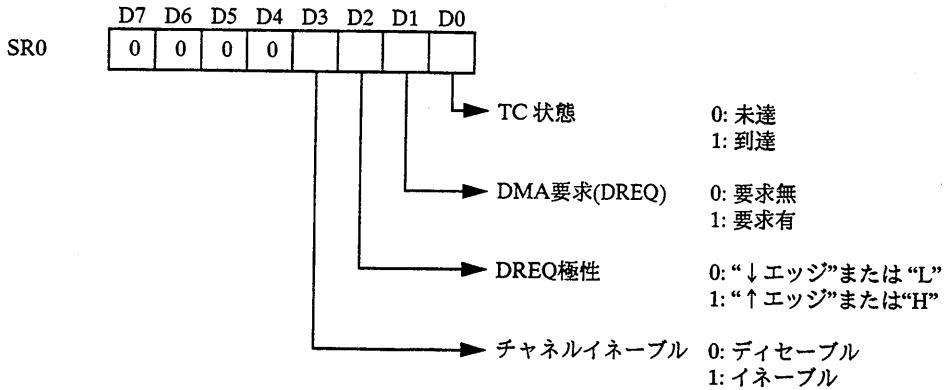
書き込み可／読み出し不可 8bit

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。

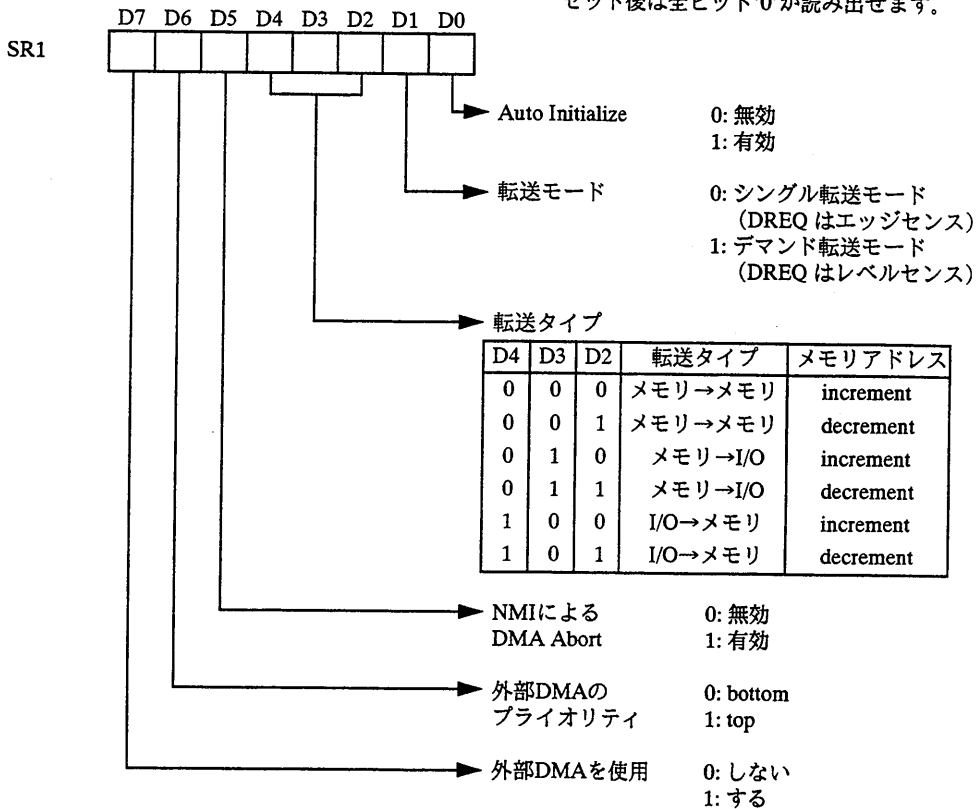
D<5:4>=00のとき、D<2:1>はdon't careです。すなわち、転送タイプでメモリ→メモリ転送を選択した場合、DREQ 極性、転送モードの設定は無視されます。また、このとき Auto Initialize は無効に設定して下さい。



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の動作設定を表すレジスタです。各ビットの意味を以下の図に示します。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。

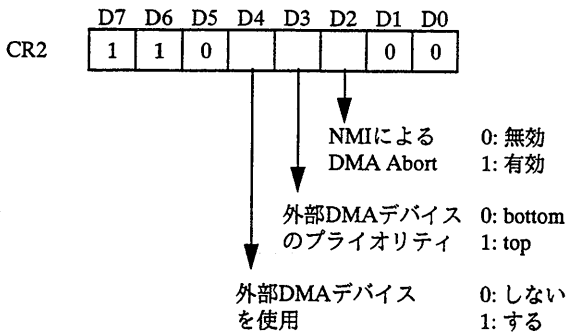
CR2 (Command Register 2)

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

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（前回は今回も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はそれぞれチャンネル0、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,    0FF15H
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,    0A0H
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>の読み出しもされます。

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す。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。

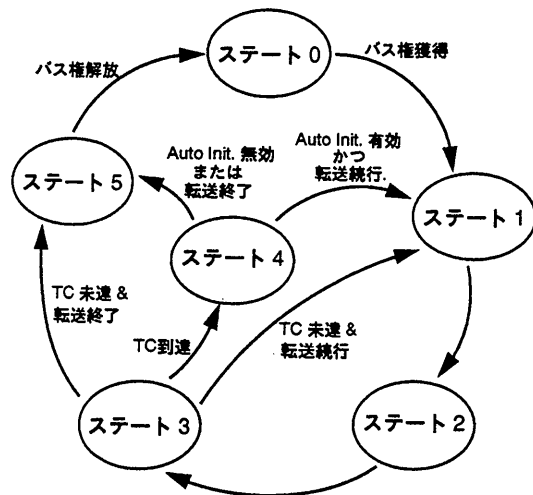


図 7-3 KP27 のステート遷移図

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 が行われた後も 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 "立ち下がりエッジ" 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

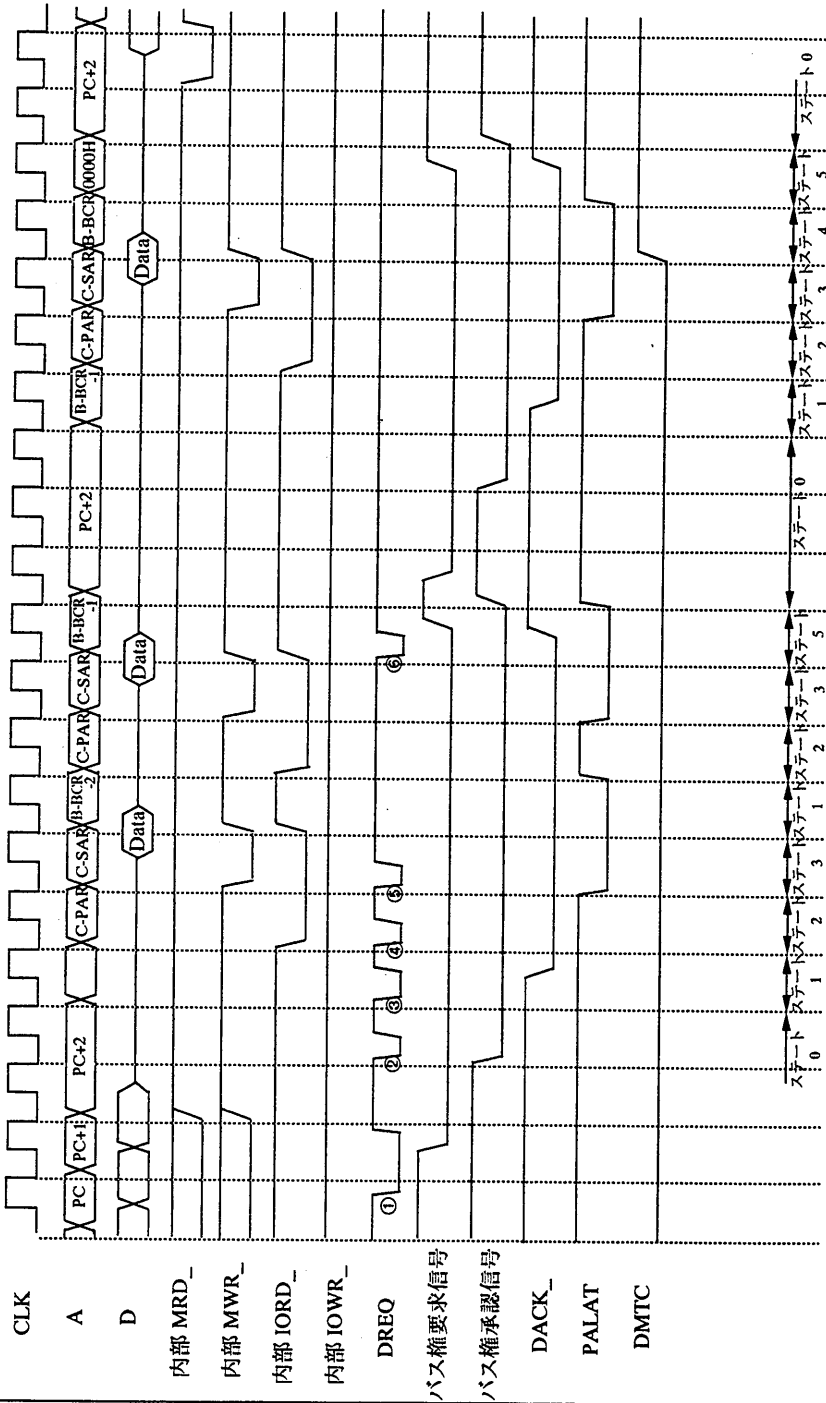


図 7-4 I/O → メモリ転送 (シングル転送モード、DREQ 極性 “↓エッジ”)

が、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ってきます(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DREQ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下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_

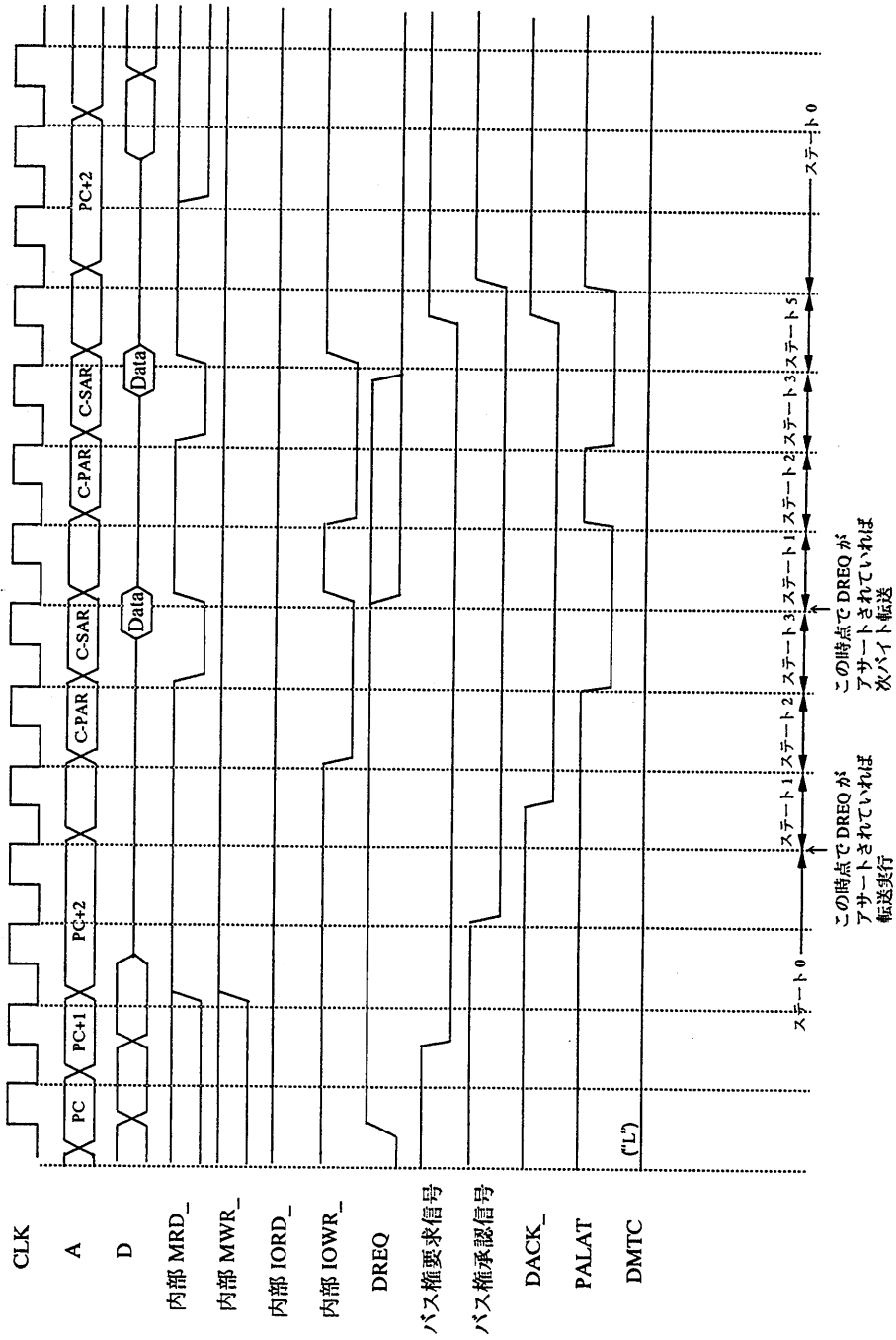


図 7-5 メモリ→I/O 転送 (デマンド転送モード、DREQ 極性 "H")

信号もアサートします。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

KL5C80A20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>により行います。ここを '1' 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 '0' 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。

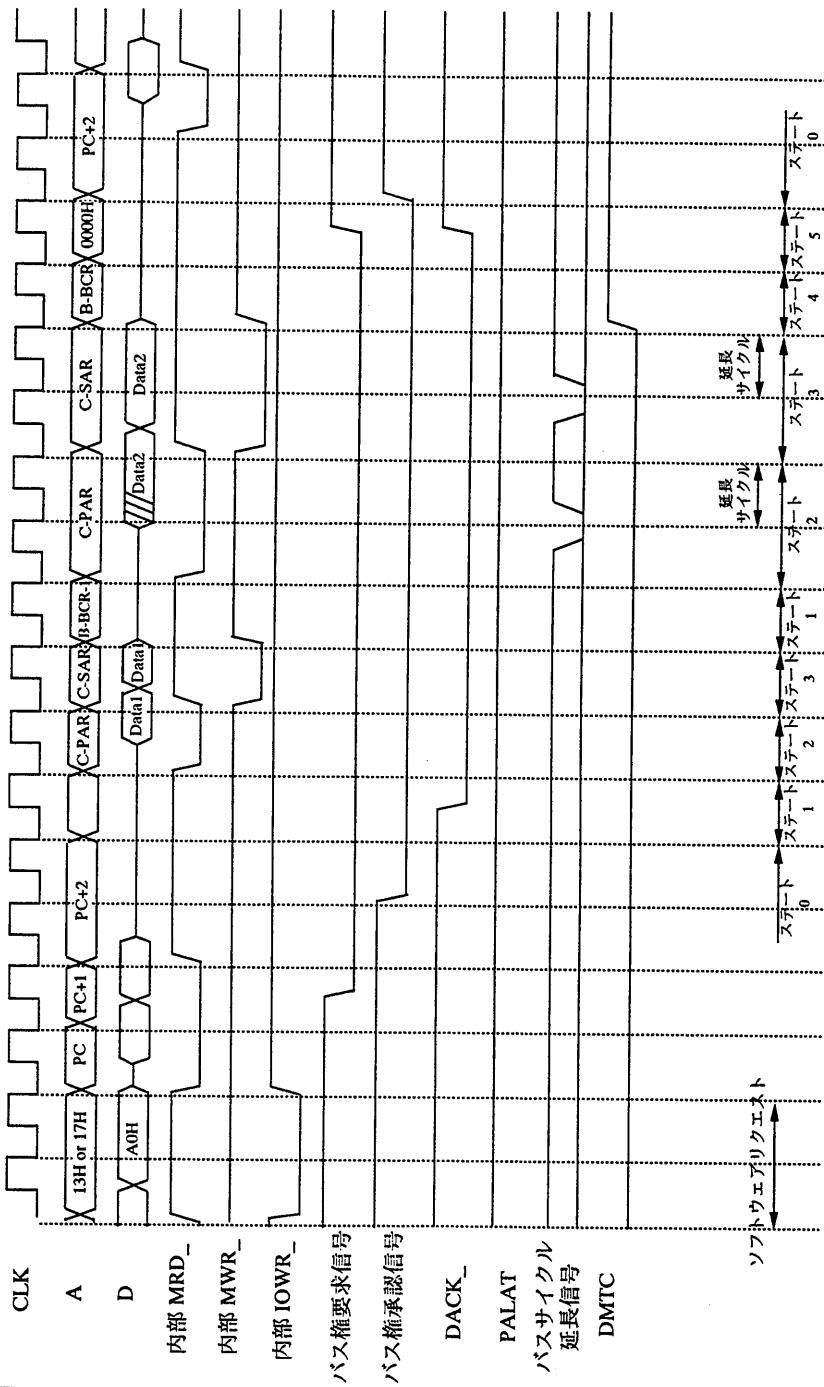
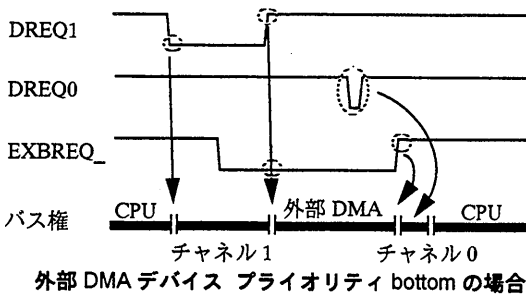
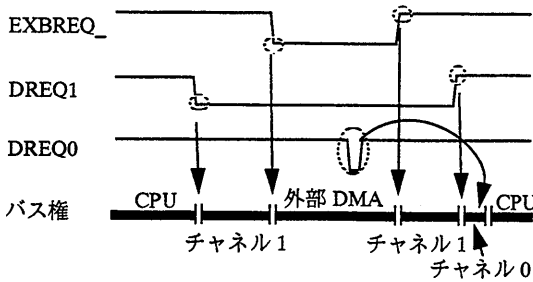


図 7-6 メモリ→メモリ転送

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。



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

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の処理を行いません。
- ・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。

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 ライトを行うクロックの立ち上がりから 1 バスサイクル + 1.5 CLK 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}
RETN    {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。)

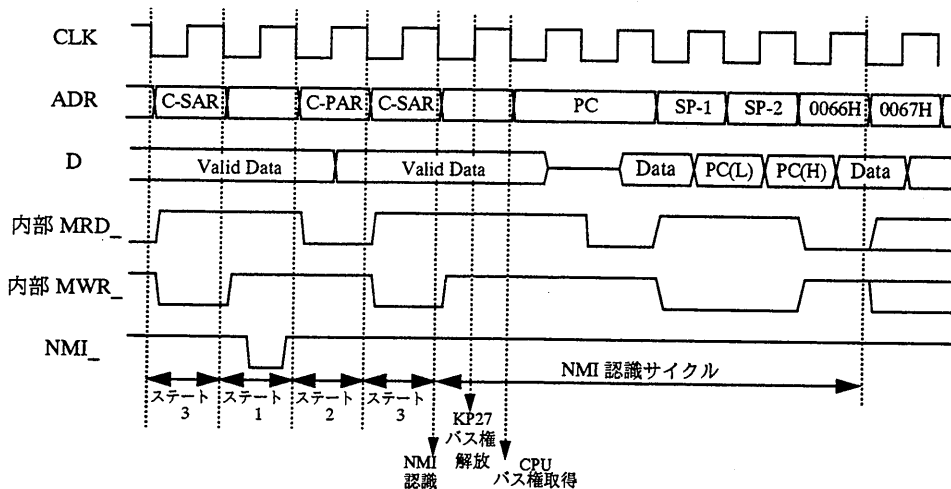


図 7-8 DMA 転送中断→ NMI ルーチン開始

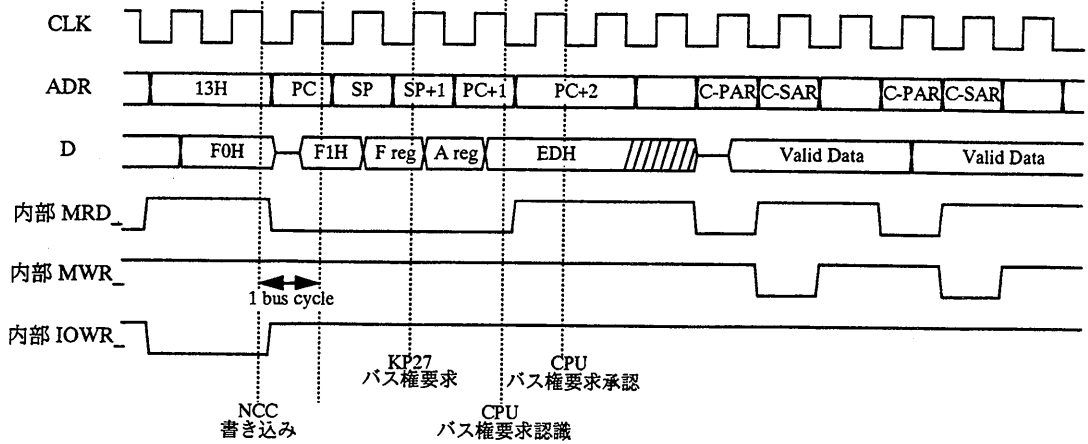


図 7-9 上記プログラム例による NMI 処理ルーチン終了→ DMA 転送再開

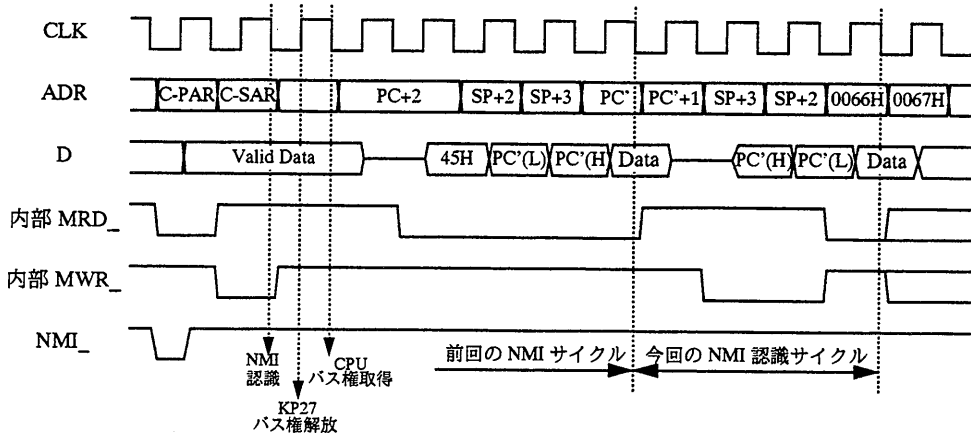


図 7-10 前記プログラム例による DMA 転送再開→ NMI ルーチン開始 (図 7-9 の続き)

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こりま

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、KL5C80A20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によって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。

8. UART

8.1 概要

KL5C80A20は非同期シリアルポートとしてKP61マイ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それぞれに専用のポーレートジェネレータを内蔵しています。ポ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。

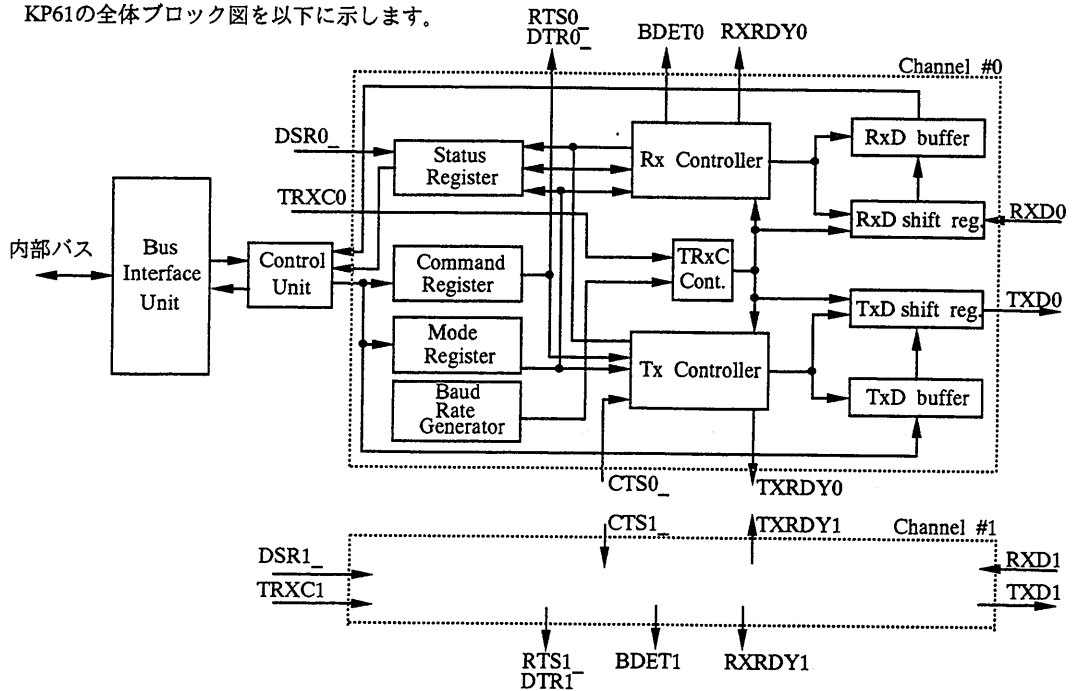


図 8-1 KP61 ブロック図

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[7], IR[8] に、RXRDY0 は IR[8] に、BDET0 は IR[8] にそれぞれ内部で接続されています。RXD0, TXD0, CTS0, RTS0 はKL5C80A20の外部端子が専用端子として割り当てられています。TRXC0, DTR0, DSR0 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13章を参照して下さい。

チャンネル1の信号のうち、TXRDY1はIR[1], IR[2]に、RXRDY1はIR[2]に内部で接続されています。またBDET1は、SCR1の設定によってIR[2]に内部で接続できるようになっています。また、DSR1_は内部で“L”に固定され、DTR1_はKL5C80A20外部へは出力されません。これら以外の信号はパラレルポートと多重化されています。詳しくは13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13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、KL5C80A20内部でIR[8]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。(13章参照) RXRDY1は、KL5C80A20内部でIR[2]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、送信ブレー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。このブレーク状態は送信ブレーク解除(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	O	<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継続します。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、KL5C80A20の内部でIR[7], IR[8]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。(13章参照)</p> <p>TXRDY1は、KL5C80A20の内部でIR[1], IR[2]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出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>DSR1_は内部でGNDに固定されています。</p>
RTS0_ RTS1_	O	<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_	O	<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、DTR1_はKL5C80A20の外部へは出力されません。</p>

端子名	I/O	機能説明
BDET0 BDET1	○	ブレーク検出信号出力端子 ブレー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、それをブレーク信号と認識して、“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 (チャンネル0)	RATE設定(チャンネル0)
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	RATE設定 (チャンネル1)	RATE設定 (チャンネル1)
2DH	川崎製鉄予約	川崎製鉄予約
2EH	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
2FH	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

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'、'2FH'）に書き込みを行うと、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いません。8.5 節のフォーマットにしたがってモード設定を行って下さい。

コマンド設定

モード設定後、I/Oアドレス'2BH'、'2FH'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。

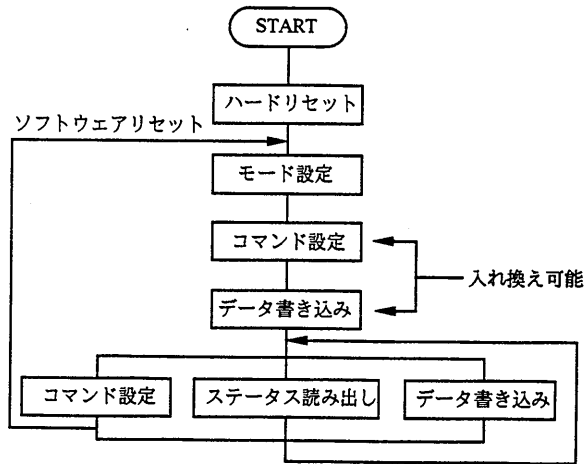


図 8-2 レジスタ設定、読み出しシーケンス

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' ヘ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われます。

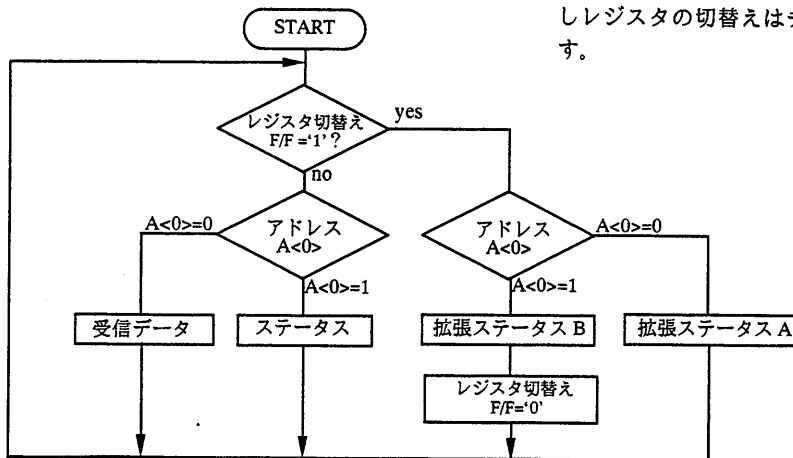
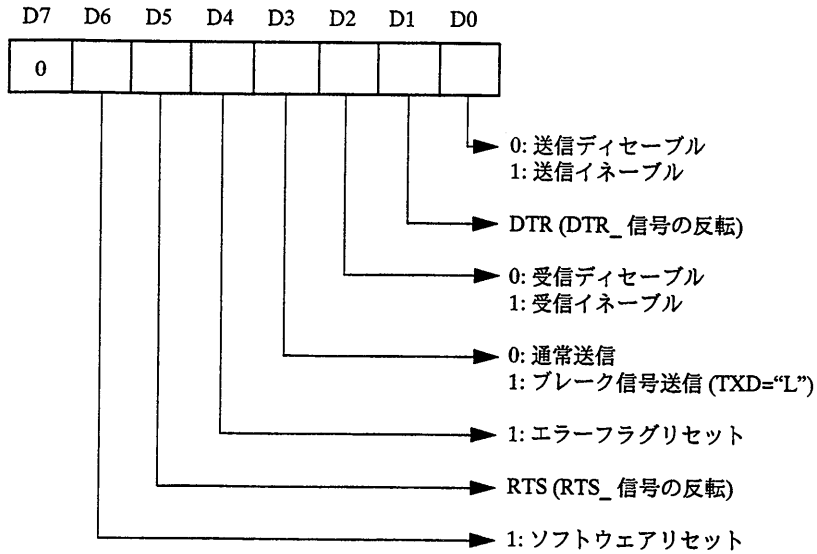


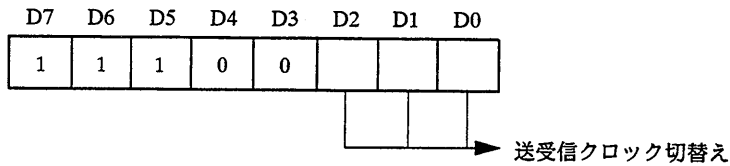
図 8-3 読み出しレジスタ切替えシーケンス

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。

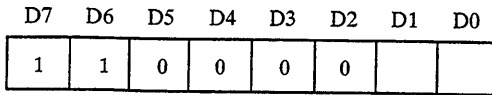
コマンドレジスタ B



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「ポ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。

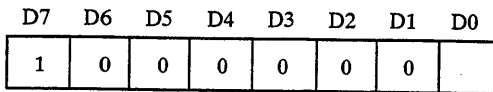
コマンドレジスタ C



送信データビット 8
(キャラクタ長 9 ビット時のみ有効)

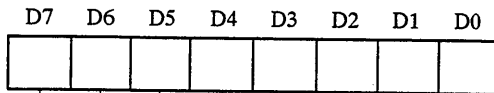
スリープモード
0: 解除
1: 設定

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

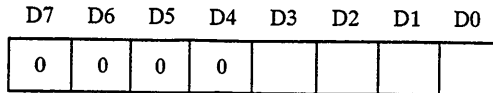
BDET (ブレーク検出) フラグ
1: 検出

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

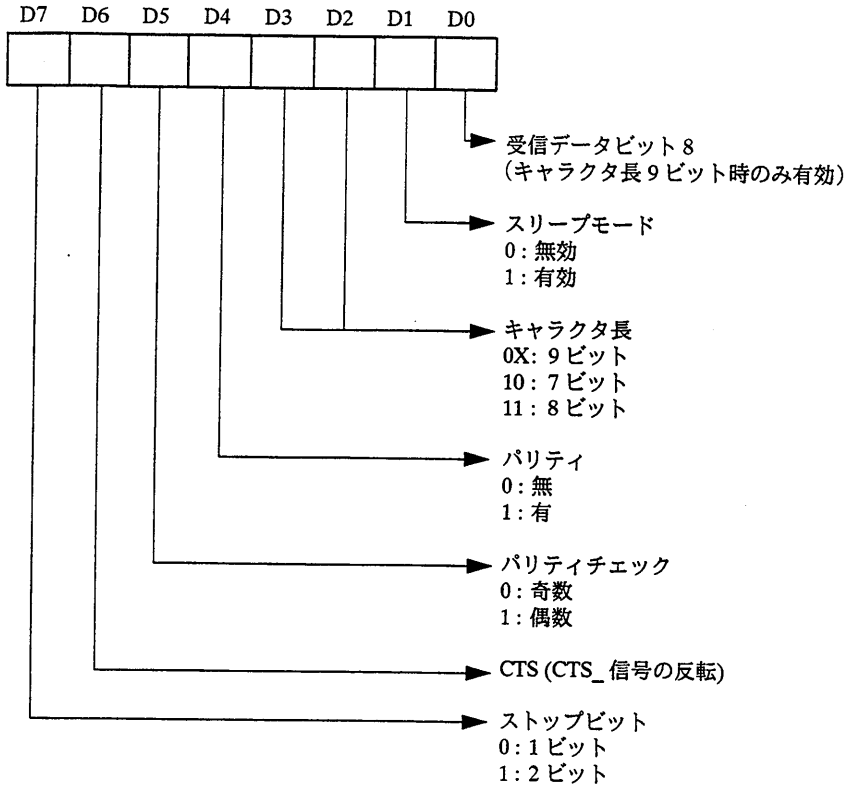


→ 使用送受信クロック設定

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_ = “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す。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す。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す。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われません。

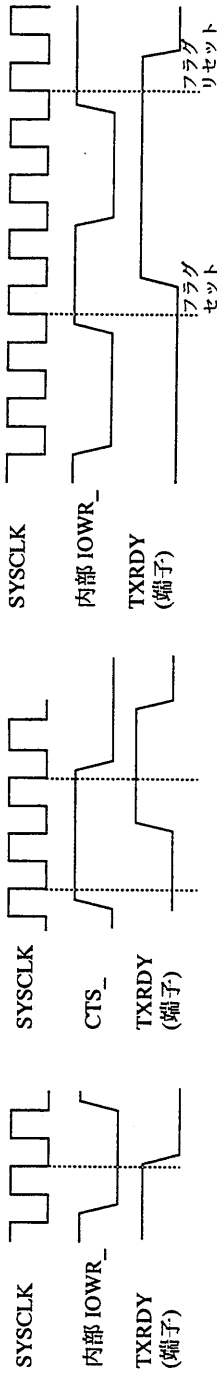


図 8-4 データライトによる TXRDY(端子)の変化

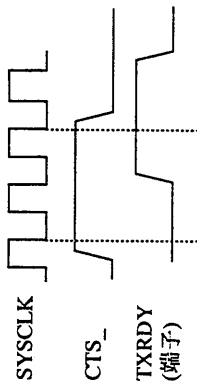


図 8-5 CTS_信号による TXRDY(端子)の変化

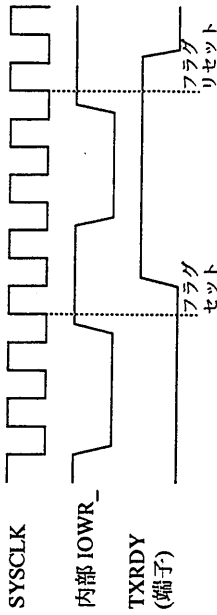


図 8-6 送信イネーブル(ディセーブル)ライトによる TXRDY(端子)の変化

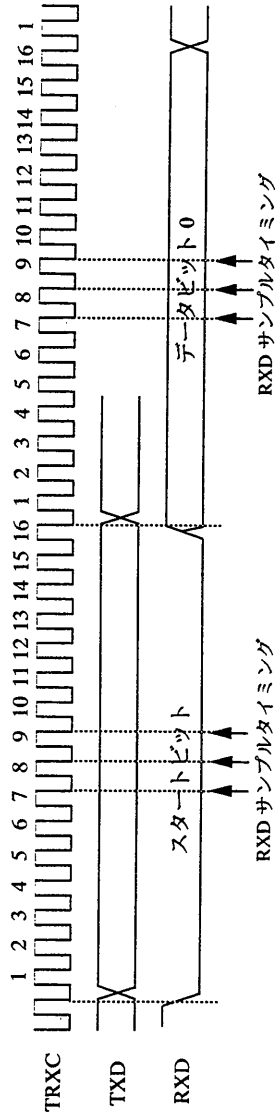


図 8-7 送受信クロックと送受信データ (送受信クロックは外部 TRXC を選択)

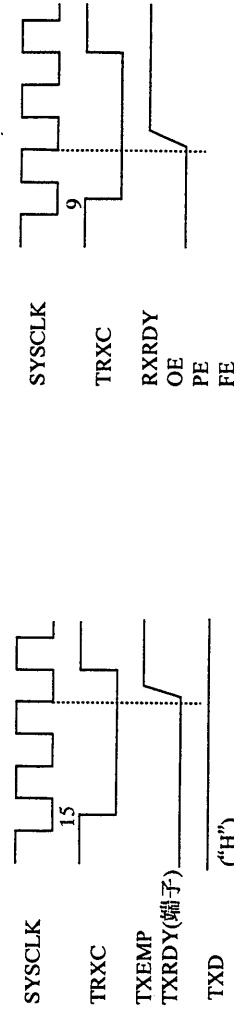


図 8-8 送受信クロックと TXEMP, TXRDY 信号変化タイミング

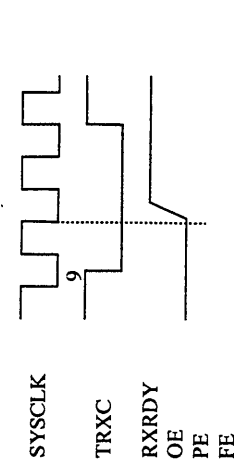


図 8-9 最初のストップビット受信中の RXRDY, エラーフラグセットタイミミング (送受信クロックは外部 TRXC を選択)

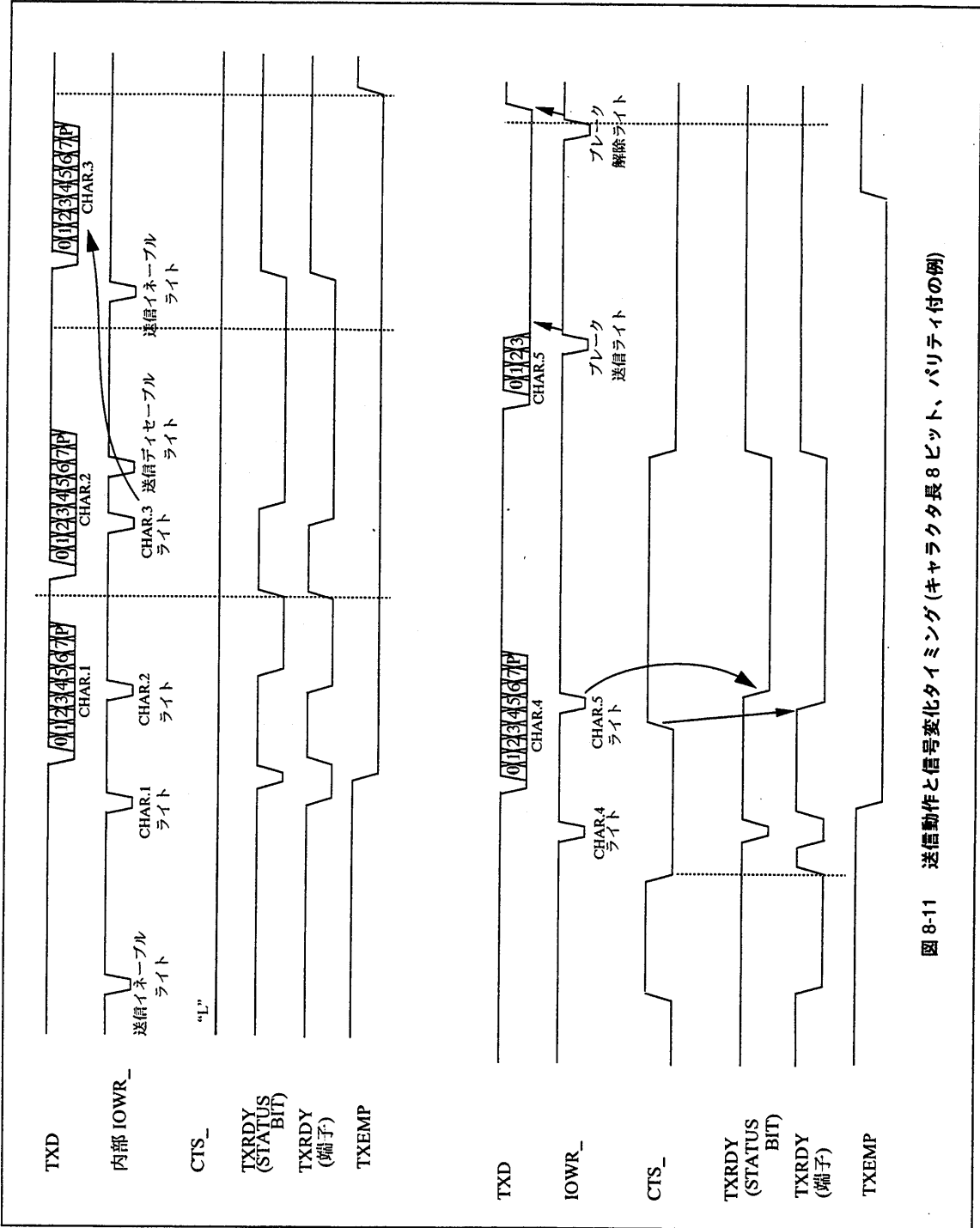


図 8-11 送信動作と信号変化タイミング (キヤラクタ長 8 ビット、パリティ付の例)

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ピング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 それぞれに専用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。

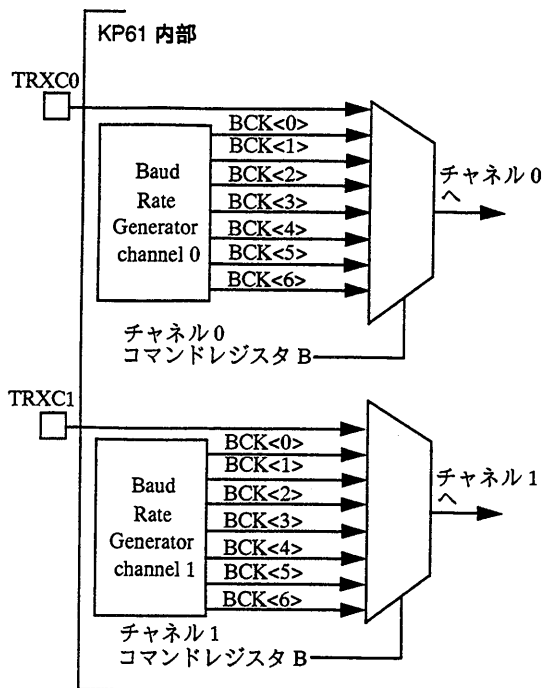


図 8-12 使用送受信クロック選択

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>によって行います。

RATE<7>='0' 及び RATE<7>='1' それぞれにおいて計算することで、あらゆるシステムクロックに対して最適な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_{sysclk}[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表8-2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、KL5C80A20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チャンネル間で独立に行われます。
- ・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。また各端子の詳細な使い方は13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

KL5C80A20 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。

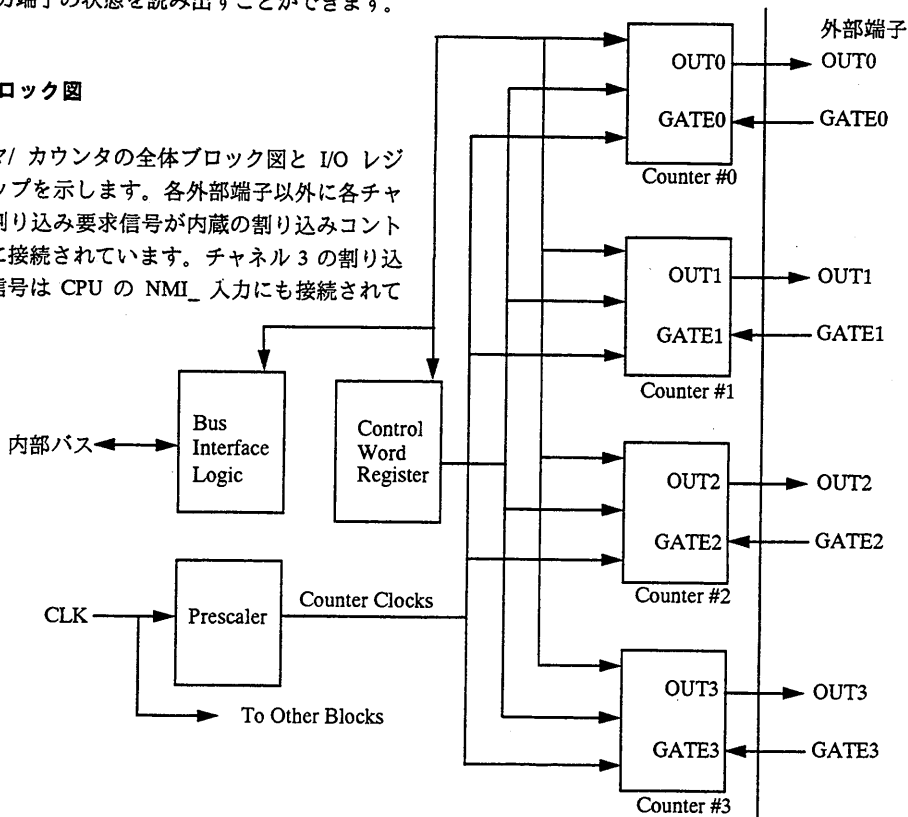


図 9-1 タイマ/カウンタのブロック図

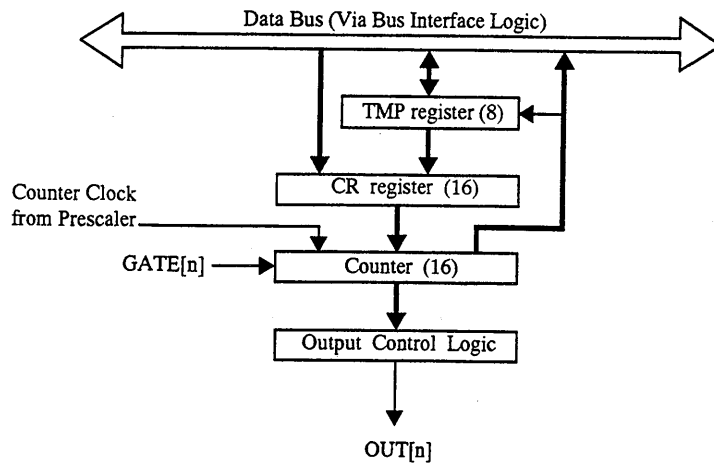


図 9-2 各チャンネルカウンタ・ユニット内のブロック図

表 9-1 I/O レジスタマッピング

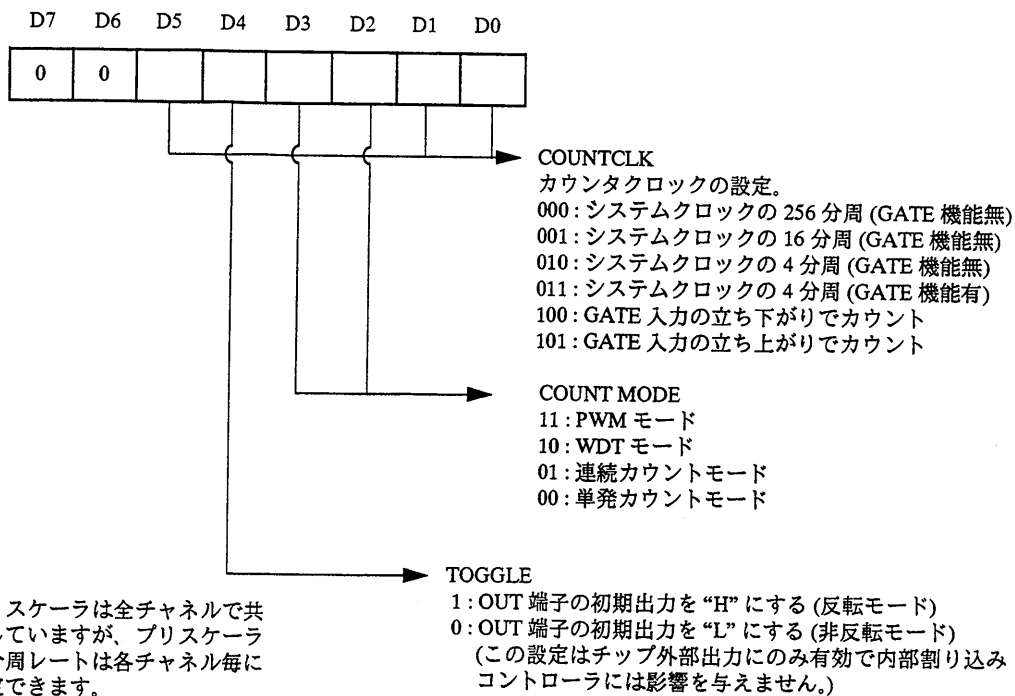
I/O アドレス	ライト時	リード時
20H	チャンネル0 カウンタ	チャンネル0 カウンタ
21H	チャンネル0 コントロールワード	チャンネル0 ステータス
22H	チャンネル1 カウンタ	チャンネル1 カウンタ
23H	チャンネル1 コントロールワード	チャンネル1 ステータス
24H	チャンネル2 カウンタ	チャンネル2 カウンタ
25H	チャンネル2 コントロールワード	チャンネル2 ステータス
26H	チャンネル3 カウンタ	チャンネル3 カウンタ
27H	チャンネル3 コントロールワード	チャンネル3 ステータス

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。



プリスケアラは全チャンネルで共有していますが、プリスケアラの分周レートは各チャンネル毎に設定できます。

9.5 カウンタへのカウント初期値の設定

カウンタは 16 ビット長のダウンカウンタですが、データバスが 8 ビット幅なので、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ため、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ので、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)

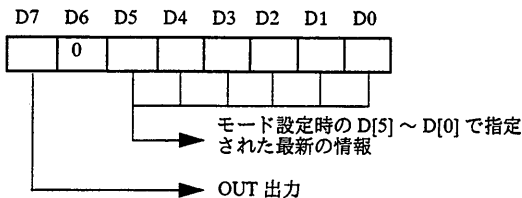
上位側：

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)

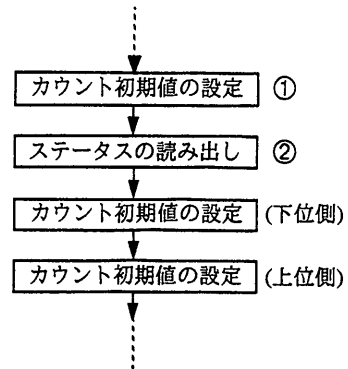
9.7 ステータスの読み出し

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 非反転モード選択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₋ 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₋ 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すパルスを 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になります。

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す。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。以上のように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 分周のカウンタ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 機能有の 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)

図 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か立ち上がりを選択できます。外部カウントクロックの 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 NMI₁ に接続されているタイマ/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

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（パルス周期データ）‘03H’、上位カウント値（パルス幅データ）‘01H’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‘0103H’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。

図にあるように、カウントダウンは上位バイト、下位バイトで独立に、かつ同時に行われ、上位側がゼロになるとOUT出力がトグルします。1回のカウントダウンは1から2、すなわちカウント初期値がロードされてから、下位側がゼロとなった後までを示します。

このように、1から2の間において、パルス幅 $(1+1)/fc = 2/fc$ 、パルス周期 $(3+1)/fc = 4/fc$ (fc はカウントダウン周波数) のパルスがOUT端子に出力されます。2の時点で内部の割り込みコントローラに割り込みが発生します。

また、カウント中にカウント初期値‘0204H’を再設定すると次に下位の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として‘0204H’がロードされます。

また、上位バイトと下位バイトに等しい、あるいは、上位バイトに下位バイトよりも大きい値を設定した場合は、3から4のようにOUT端子は常に“H”を出力します。下位バイトがゼロになった次のカウントダウンサイクル（図中の4）において、カウンタにはカウント初期値‘0505H’がロードされ、再びカウントダウンが連続して実行されます。

9.11 リセット

RESET_入力“L”により次の機能に影響を与えません。

- (1)カウンタの動作を停止します。また、リセット解除後もカウンタは停止状態を保持します。
- (2)カウンタ内部のレジスタ読み出しの際のリードシーケンス、およびカウント初期値書き込みの際のライトシーケンスをクリアします。
- (3)カウンタ内部のCRレジスタおよびカウンタレジスタは‘FFFFH’にクリアされます。
- (4)単発カウントモードで256分周カウンタクロックOUT出力非反転モード選択となり、OUT出力を“L”に設定し、リセット解除後もモード設定が行われるまで、その値を保持します。

9.12 注意事項

カウントの停止方法

カウント中にモードの再設定を行なうとカウント動作を停止します。

カウント初期値の最大/最小値

カウント初期値の最大/最小値を表9-2に示します。

表9-2 カウント初期設定値

モード	最小値	最大値
連続カウント	0001H	FFFFH
単発カウント	0001H	FFFFH
パルス幅変調(PWM)	上位:01H 下位:01H	上位:FFH 下位:FFH
ウォッチドッグタイマ (WDT)	設定不可*	設定不可*

*: 他のモードで設定後、WDTモードに再設定した上で、使用する際は、カウント初期値設定時のモードでの条件に従います。

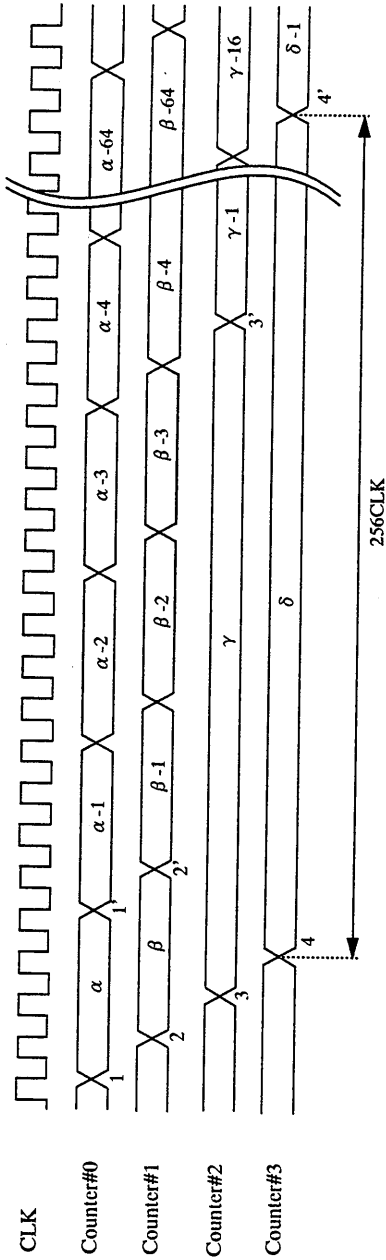


図 9-3 カウントダウンタイミング (チャンネル 0,1: GATE 機能無 4 分周, チャンネル 2: 16 分周, チャンネル 3: 256 分周選択時)

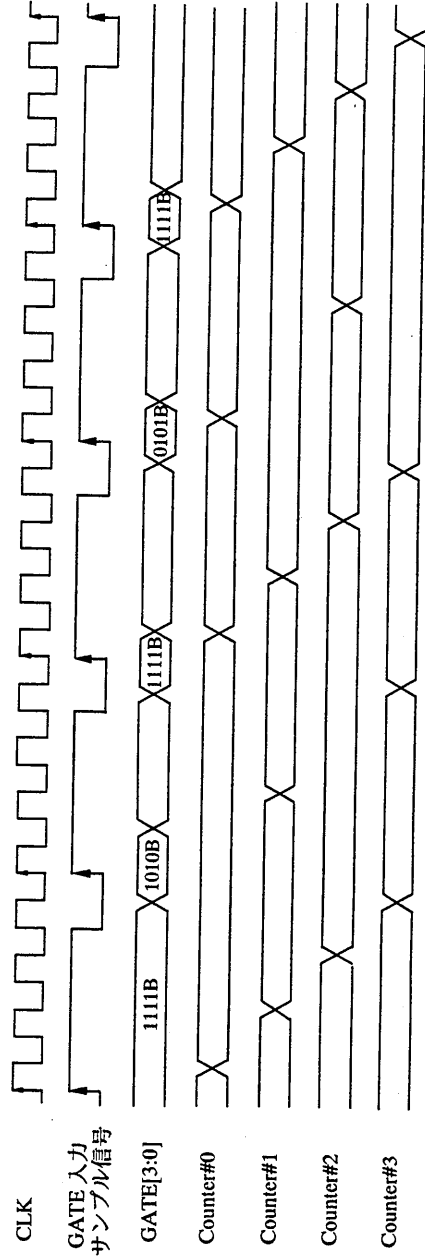


図 9-4-A GATE 信号のサンプルタイミング (GATE 機能有 4 分周)

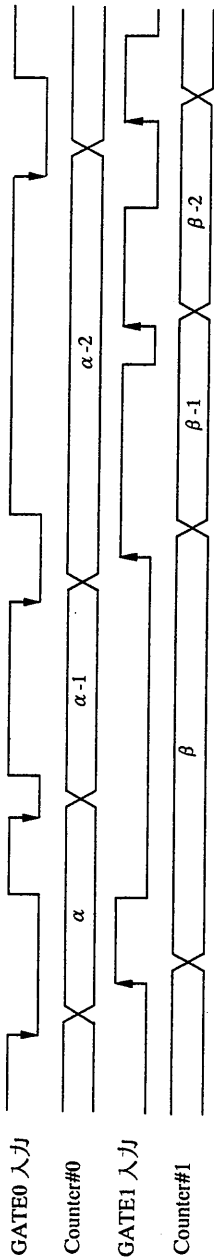


図 9-4-B GATE 入力からのクロック入カタイミング (チャンネル 0 は GATE の立ち下がりでカウント、チャンネル 1 は立ち上がりでカウント)

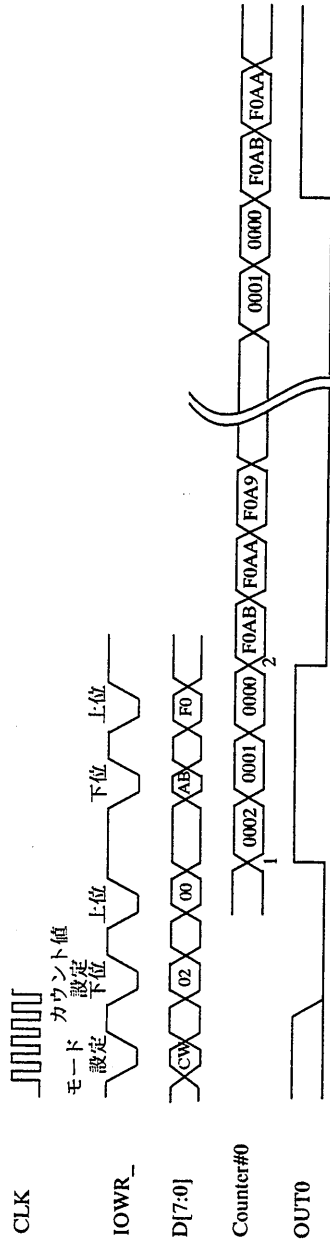


図 9-5 連続カウントモード (GATE 機能無 4 分周、OUT 非反転モード選択時)

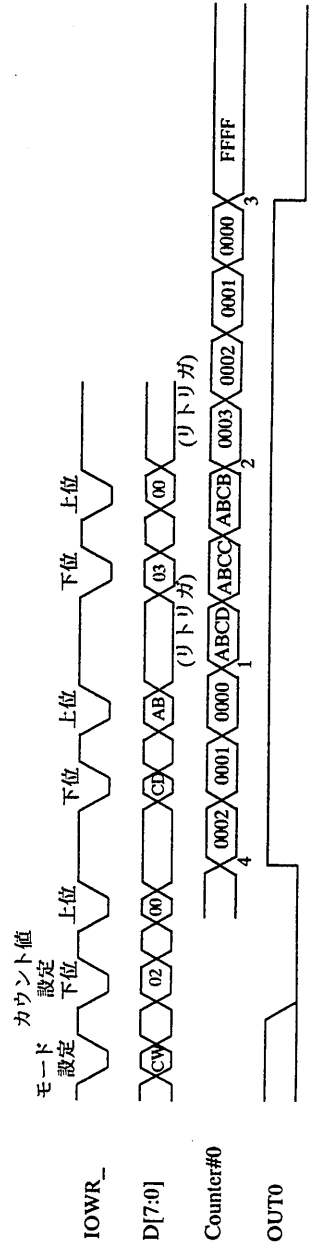


図 9-6 単発カウントモード (GATE 機能無 4 分周、OUT 非反転モード選択時)

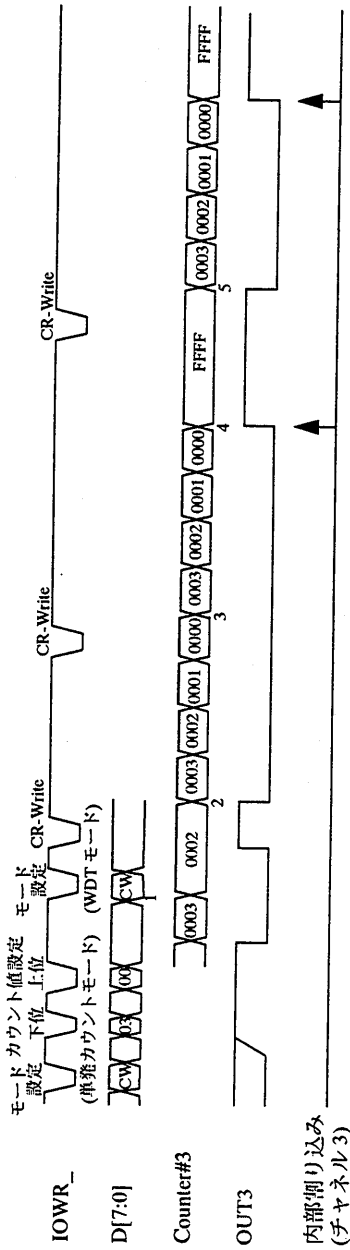


図 9-7 WDT モード (GATE 機能無 4 分周、OUT 反転モード選択時)

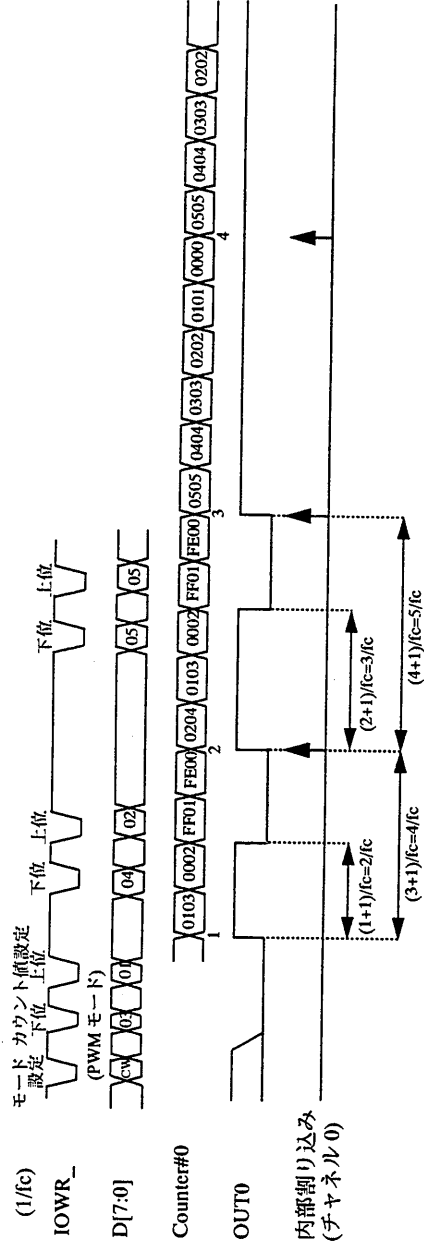


図 9-8 PWM モード (GATE 機能無 4 分周、OUT 非反転モード選択時)

10. クロック同期シリアル I/O

10.1 概要

KL5C80A20 のクロック同期シリアル I/O は KP62 マクロセルを 1 チャンネル搭載しています。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。

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。

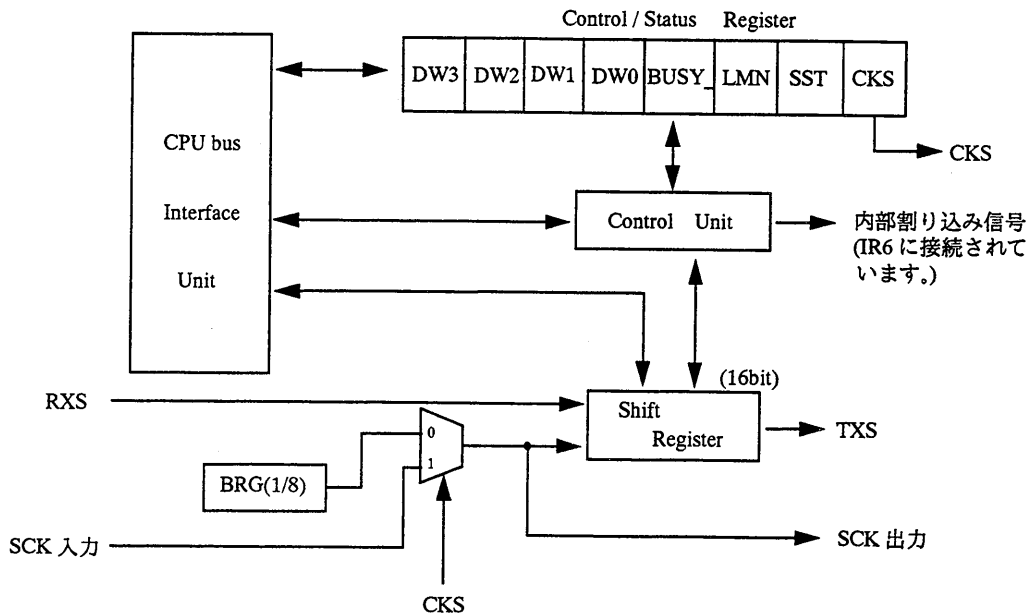


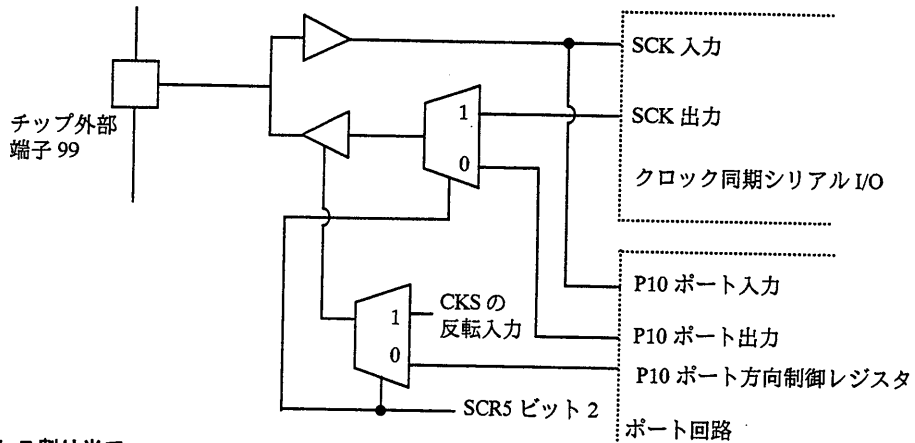
図 10-1 クロック同期シリアル I/O のブロック図

10.3 端子機能

端子名	I/O	機能説明
SCK	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。
RXS	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。
TXS	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。

SCKの内部接続

クロック同期シリアル I/O 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	川崎製鉄予約	川崎製鉄予約
33H	川崎製鉄予約	川崎製鉄予約

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 内のポーレートジェネレータ (BRG) 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に、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がりに指定されます。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がりに指定されます。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。

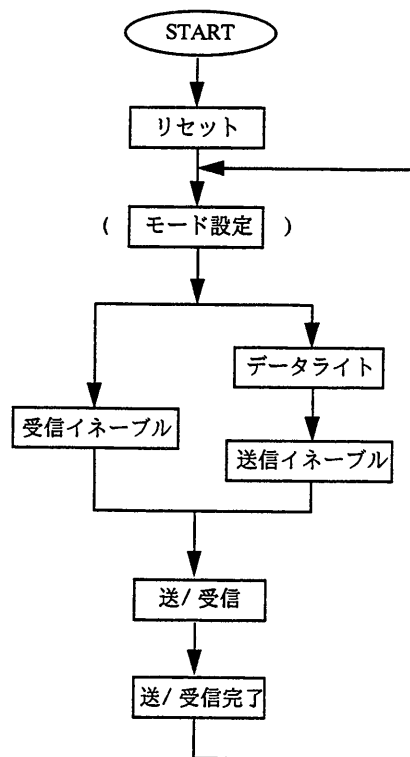
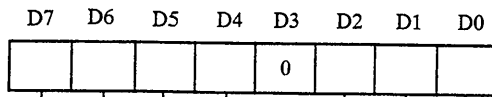


図 10-2 初期設定フロー

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)

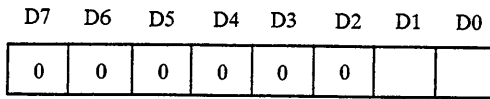
SST
シフトタイミング/サンプルタイミング
0: 内部クロック選択時
SCK 出力初期値 = "H"
1: 内部クロック選択時
SCK 出力初期値 = "L"

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ご注意ください。

コマンドレジスタ



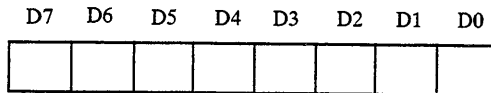
D1	D0	コマンド
0	0	キャンセル
1	0	送信イネーブル
1	1	受信イネーブル
0	1	使用禁止

(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ご注意ください。

ステータスレジスタ

ステータス情報は、I/O アドレス = '31H' 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: 送信中または受信中

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を書き込み/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、送信データの下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勧めします。

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。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超えて、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。

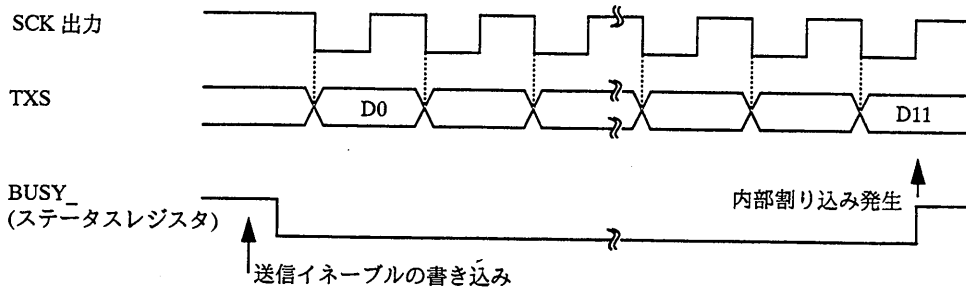


図 10-3 内部シリアルクロック、12ビットキャラクタ長、SST = '0'、LSB 先頭時の送信動作タイミング

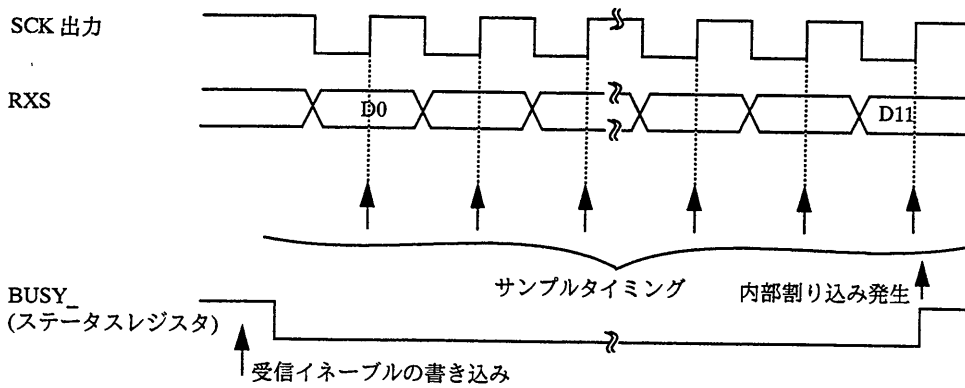


図 10-4 内部シリアルクロック、12ビットキャラクタ長、SST = '0'、LSB 先頭時の受信動作タイミング

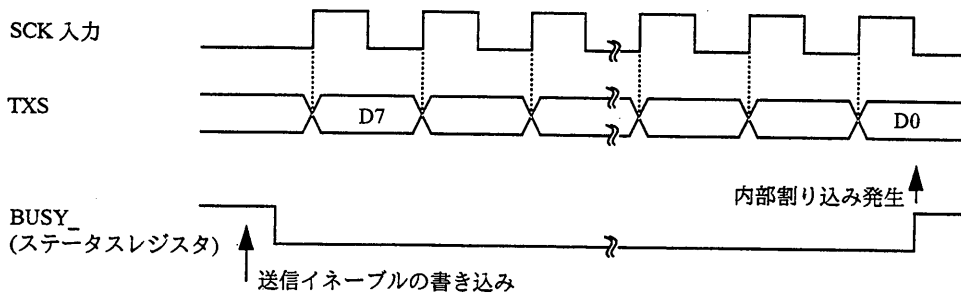


図 10-5 外部シリアルクロック、8ビットキャラクタ長、SST = '1'、MSB 先頭時の送信動作タイミング

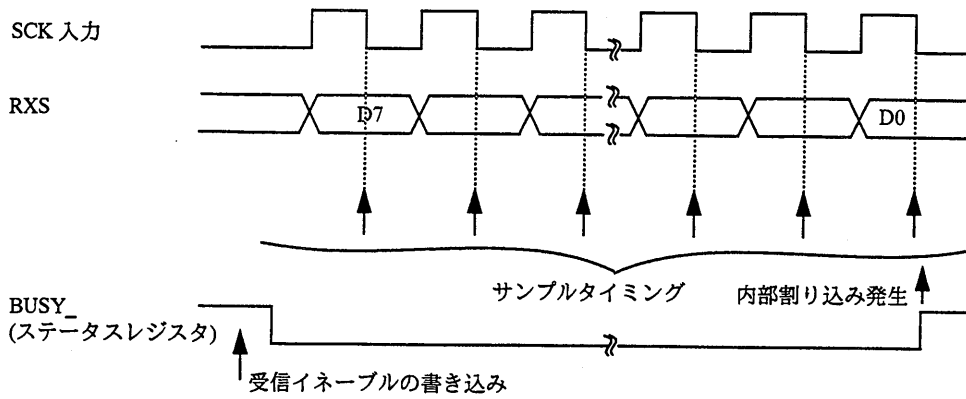


図 10-6 外部シリアルクロック、8ビットキャラクタ長、SST='1'、MSB先頭時の受信動作タイミング

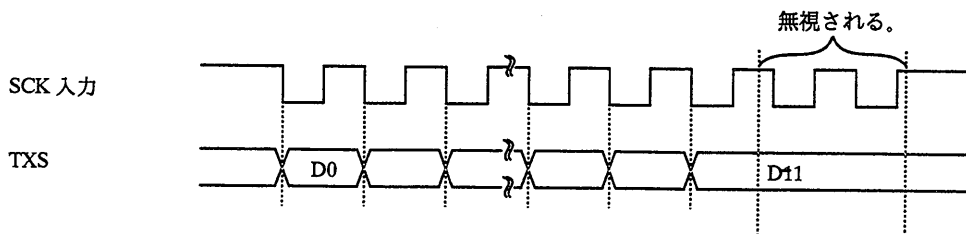


図 10-7 必要以上に入力される外部シリアルクロックの例 (送信時)

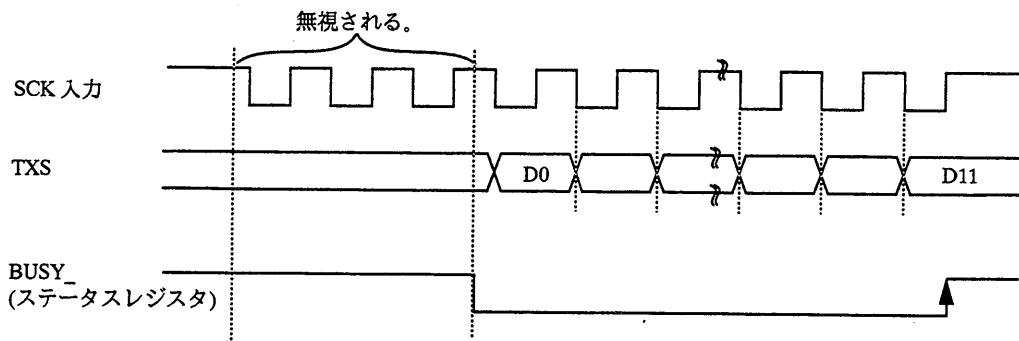


図 10-8 待機中に入力される外部シリアルクロックの例 (送信時)

11. HDLC インターフェース

11.1 概要

KL5C80A20 は HDLC インターフェースとして KP91A マクロセルを搭載しています。本マクロセルは最大転送速度 5M bps の HDLC シリアルポートを 1 チャンネル搭載しており、HDLC フレームの高速送受信が可能です。8 段の受信 FIFO、4 段の送信 FIFO を内蔵しているため、高速転送時のソフトウェア処理が容易になり、また、送受信割り込みのオーバーヘッドを低減することが可能です。さらに、ボーレートジェネレータ、クロック再生のための DPLL(Digital Phase Locked Loop) を搭載しているため、送受信クロックが付加回路なしで生成できます。

特徴

- ・最大転送速度 5M bps
- ・自動 '0' 挿入、削除機能
- ・アポート送出、検出機能
- ・CRC 生成、検査 (CRC-16 / CRC-CCITT、プリセット '1' / '0')
- ・受信 FIFO 8 段、送信 FIFO 4 段
- ・NRZ、NRZI、FM0、FM1 データ・エンコーディング対応
- ・ボーレートジェネレータ内蔵
- ・クロック再生用 DPLL 内蔵 (DPLL 使用時の最大転送速度は 1.25Mbps)
- ・ループ・モード対応
- ・送信割り込み、受信割り込み、外部 / ステータス割り込み、Special Condition 割り込み発生機能
- ・送信 DMA 転送要求、受信 DMA 転送要求発生機能
- ・ローカル・ループバック機能、オートエコー機能

11.2 ブロック図

KP91A 全体のブロック図を以下に示します。

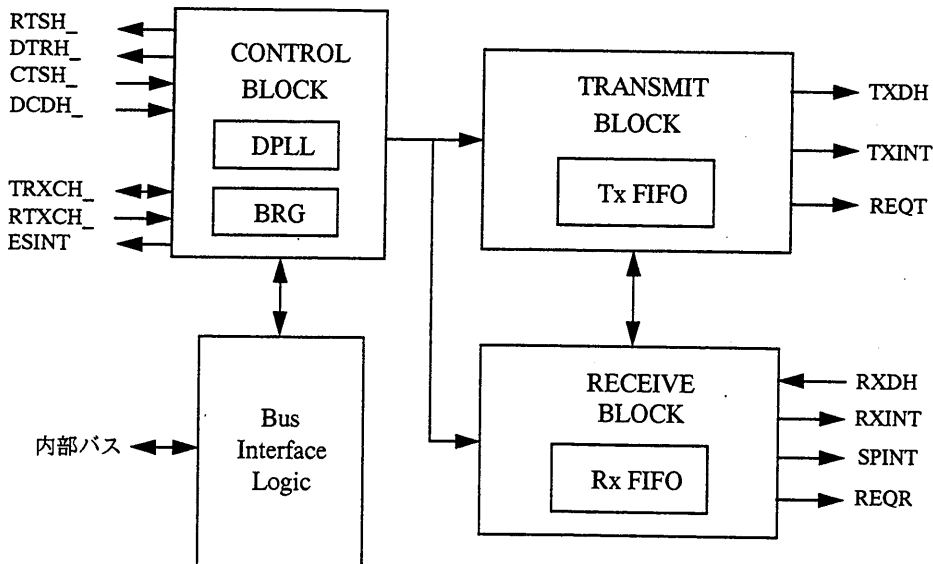


図 11-1 KP91A ブロック図

11.3 端子機能説明

下表に KP91A の端子機能を示します。

端子名	I/O	機能説明
TXDH	O	送信データ出力端子 送信データが送信クロックにしたがって、シリアルに出力されます。
RXDH	I	受信データ入力端子 シリアルな受信データが入力され、受信クロックでサンプルされます。
CTSH_	I	クリア・トゥ・センド入力端子 Auto Enable に設定された時 (WR4 D0 = '1') は送信イネーブルとして機能します。Auto Enable に設定しないときは汎用入力ポートとして機能します。
DCDH_	I	データ・キャリア・ディテクト入力端子 Auto Enable に設定された時 (WR4 D0 = '1') は受信イネーブルとして機能します。Auto Enable に設定しないときは汎用入力ポートとして機能します。
RTSH_	O	リクエスト・トゥ・センド出力端子 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、汎用出力ポートとして使用することもできます。
DTRH_	O	データ・ターミナル・レディ出力端子 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、汎用出力ポートとして使用することもできます。
RTXCH_	I	送信/受信クロック入力端子 WR6 の設定にしたがって、送信クロックあるいは受信クロック入力として使用します。また、ポーレートジェネレータ、DPLL のソース・クロックとして使用することも出来ます。
TRXCH_	I/O	送信/受信クロック入力端子 WR6 の内容にしたがって、送信クロックあるいは受信クロック入出力として使用します。
RXINT	O	受信割り込み要求出力端子 受信データが到着し、読み出し可能になったときに RXINT = "H" となり、受信割り込み要求信号を発生します。
TXINT	O	送信割り込み要求出力端子 送信データが書き込み可能状態になると TXINT = "H" となり、送信割り込み要求信号を発生します。

端子名	I/O	機能説明
SPINT	O	Special Condition 割り込み要求出力端子 受信データに Special Condition が発生すると SPINT = "H" となり、Special Condition 割り込み要求を発生します。
ESINT	O	External/Status 割り込み要求出力端子 WR10 で設定する External/Status 割り込みが発生したときに ESINT = "H" となります。
REQR	O	受信 DMA 転送要求出力端子 受信データが到着して DMA 転送が可能になったとき、REQR = "H" となり受信 DMA 転送要求を発生します。
REQT	O	送信 DMA 転送要求出力端子 送信データが書き込み可能になると REQT = "H" となり、送信 DMA 転送要求を発生します。

11.4 内部レジスタマッピング

KP91A の内部レジスタのマッピングを表 11-1 に示します。各レジスタの詳細説明は後述します。

表 11-1 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
40H	WR0	RR0
41H	WR1	RR1
42H	WR2	RR2
43H	WR3	RR3
44H	WR4	RR4
45H	WR5	RR5
46H	WR6	RR6
47H	WR7	RR7
48H	WR8	RR8
49H	WR9	RR9
4AH	WR10	RR10
4BH	川崎製鉄予約	RR2'
4CH	川崎製鉄予約	RR9'

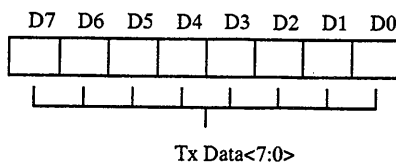
WRn : 送信データ・レジスタ、RRn : 受信データ・レジスタ

11.5. レジスタ構成

以下に KP91A の内部レジスタの説明を行います。KP91Aの送受信動作等は既知のものとして説明を行いますので、不明な点は適宜「11.6 送受信動作の説明」を参照して下さい。WR0 から WR10 まではライト・レジスタ、RR0～RR10、RR2'、RR9' はリード・レジスタです。

また、レジスタの設定方法については「11.12 レジスタ設定」を参照してください。

WR0 送信データ・レジスタ



送信データ・レジスタに送信データをライトすると、Tx FIFO に入った後、送信シフト・レジスタでパラレル→シリアル変換され、必要に応じて'0'挿入が行われた後、TXDH からシリアル・データが送信されます。ただし、Tx Enable (WR1 で設定) 状態でなければ、データを書き込んでも送信は行われません。

WR1 各種コマンド・レジスタ

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

Null Code
Tx Enable
Tx Disable
Reset Tx Underrun/EOM Latch
Send Abort
Tx FIFO Reset
Rx Enable
Rx Disable
Enter Hunt Mode
Error Reset
Rx FIFO Reset

Null Code
Reset E/S Int
Enable Int On Next Rx Character
Reset Tx Int
Software Reset

WR1 は送受信、割り込みに関する各種コマンドを発行するのに用います。

の状態等はハードウェア・リセット後と同じです。

D7 このビットへの書き込みはKP91Aの動作に影響を与えません。

D5,D4 割り込み関係のコマンド

これらのビットの組合せによって以下のような動作を行います。

D6 Software Reset

このビットを '1' にするとソフトウェア・リセットが行われます。ソフトウェア・リセット後のレジスタ

00 Null Code

KP91Aの動作には影響を与えません。

01 Reset E/S Int

External/Status Int をリセットします。このコマンドを発行すると External/Status Int Latch を解放し、新たな割り込み要因の取り込みを開始します。

10 Enable Int On Next Rx Character

Rx Int のモードが "Rx Int On First Rx Character And Sp Int" のときに用いるコマンドで、通常はフレーム受信終了の割り込み (End Of Frame の Sp Int) ルーチン中で発行します。Rx FIFO に受信データがない場合は、このコマンドを発行した後、最初のデータを受信したときに Rx Int を発生します。既に Rx FIFO にデータが存在する場合にはこのコマンドを発行するとただちに Rx Int が発生します。

11 Reset Tx Int

Tx Int をリセットします。フレームの送信の最後で、フレームの送信を終了させたいときに、このコマンドを発行します。このコマンドを発行した後の Tx Int の動きは次の 2 つの場合に分けることができます。CRC の 2 バイト目が送出されるまでに、次のフレームの送信データが書かれない場合は、CRC の 2 バイト目が送出されているときに Tx Int が発生します。CRC の 2 バイト目が送出される前に次のフレームの送信データが書かれた場合は、次に Tx FIFO が、Tx Int Mode (WR2 D1) の設定で決まる所定の状態になったときに、再び Tx Int が発生します。

D3~D0 送受信制御コマンド

これらのビットの組合せによって以下のような動作を行います。

0000 Null Code

KP91A の動作には影響を与えません。

0001 Tx Enable

Tx Enable 状態になります。Tx FIFO にデータがある場合は、送信データが TXDH から送出されます。Tx FIFO にデータがない場合は、Flag もしくは Mark が送出されます。

0010 Tx Disable

Tx Disable 状態になります。送信データの送出は

中止され、Mark 送出状態になります。

0011 Reset Tx Underrun/EOM Latch

Tx Underrun/EOM Latch をリセットします。フレーム送信の終了時に CRC を送出させるためには、Underrun が発生する前にこのコマンドで Tx Underrun/EOM をリセットしておく必要があります。このコマンドは Tx Enable 状態のときしか有効になりません。また、Auto Enable (WR4 D0) に設定した場合は、Tx Enable コマンドを発行した後も、CTS_ = "L" でなければ、このコマンドを発行しても Tx Underrun/EOM Latch はリセットされませんので注意が必要です。

0100 Send Abort

Abort (8 個の '1') を送出します。このとき Tx Underrun/EOM = '1' となり Tx FIFO はリセットされます。

0101 Tx FIFO Reset

Tx FIFO をリセットし、Empty 状態にします。Tx FIFO にライトしたデータをキャンセルしたいときに用います。

1000 Rx Enable

Rx Enable 状態になり、Flag 検出、Abort 検出を開始します。

1001 Rx Disable

Rx Disable 状態になり、Hunt (RR1 D4) = '1' となります。データ受信中にこのコマンドを発行するとただちに受信を中止します。また、Rx Disable 状態では Flag 検出、Abort 検出は行いません。

1010 Enter Hunt Mode

Flag 検出状態になり、Hunt = '1' となります。通常はフレーム受信を中止し、次のフレームの Flag 検出に移行する場合に用います。

1011 Error Reset

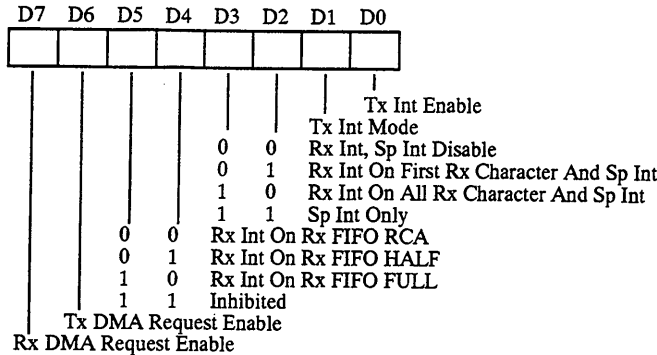
RR2 の End Of Frame ビット、Rx Overrun Error ビットをリセットします。Rx Int のモードが "Rx Int On First Rx Character And Sp Int" または "Sp Int Only" の場合に、Special Condition が発生すると、Rx FIFO はロックされて Special Condition (Rx Overrun Error および End Of Frame) を含む受信データとそれに対応する RR2 のス

データがRx FIFOに保持されます。Rx FIFOのロックの解除はこのコマンドで行います。また、このコマンドでSp Intもリセットされます。

1100 Rx FIFO Reset

Rx FIFOをリセットします。受信データの読み出しをキャンセルする場合に用います。

WR2 割り込み、DMA制御



WR2は送信、受信割り込みのモード設定、DMA転送要求のイネーブル制御を行います。

D7 Rx DMA Request Enable

このビットが‘1’のとき、受信データがRx FIFOにあるとRx DMA Request (REQR 信号)を発生します。ただし、Special ConditionのためにRx FIFOがロックされているときはRx FIFOにデータが存在してもRx DMA Requestは発生しません。

D6 Tx DMA Request Enable

このビットが‘1’のとき、Tx FIFOがEmpty状態ならばTx DMA Request (REQT 信号)を発生します。これはTx Enable、Disableの状態に関係なく発生します。

D5,D4 Rx Intモード設定

Rx FIFOがどのような状態になったとき、Rx Intが発生するかを設定します。これらのビットの設定とRx Int、Sp Int発生の関係は表8-1にまとめてありますので、適宜参照して下さい。

00 Rx Int On Rx FIFO RCA

Rx Character Available(RCA)状態すなわちRx FIFO

にデータが1つでも存在する場合にRx Intを発生します。このモードはビットレートが速い場合に、割り込みで受信データを転送する際に用います。Sp IntはSpecial Conditionを含む受信データがRx FIFOの出口まで来たとき、あるいはSpecial Conditionを含むデータが読み出されたときに発生します。

01 Rx Int On Rx FIFO Half

Rx FIFOに半分(4段)まで受信データが入った場合にRx Intを発生します。比較的ビットレートが遅く、割り込みの回数を減らしたい場合に用います。Sp IntはSpecial Conditionを含むデータがRx FIFOに入るとただちに発生します。どの受信データがSpecial Conditionを含むデータであるかは、データを読み出す前にRR2を読み出して確認する必要があります。

10 Rx Int On Rx FIFO Full

Rx FIFOにFull状態(8段)まで受信データが入った場合にRx Intを発生します。ビットレートが遅く、割り込みの回数を減らしたい場合に用います。Sp IntはSpecial Conditionを含むデータがRx FIFOに入るとただちに発生します。どの受信データがSpecial Conditionを含むデータであるかは、データを読み出す前にRR2を読み出して確認する必要があります。

11 Inhibited

このビットの組合せは禁止されています。

D3,D2 Rx Int, Sp Int制御

Rx Int, Sp Int の発生を制御します。これらのビットの設定とRx Int, Sp Int発生の関係は表8-1にまとめてあります。

00 Rx Int, Sp Int Disable

Rx Int, Sp Intは発生しません。ポーリングでRR1をリードし、RCA ビット (RR1 D0) の状態を確認することで受信データの有無を判断する場合に用います。

01 Rx Int On First Rx Character And Sp Int

ハードウェア・リセットあるいはソフトウェア・リセット後にはじめてデータを受信し RCA 状態になったとき、および“Enable Int On Next Rx Character”コマンド (WR1) 発行後、はじめてデータを受信し RCA状態になったときにのみRx Intを発生します。Sp IntはSpecial Conditionを含む受信データがRx FIFOからリードされた直後に発生します。このときRx FIFOはロックされた状態になり、最後にリードされた受信データおよびそれに対応する受信ステータス RR2 がRx FIFOの出口に保持されます。Special ConditionがRx Overrun Errorによるものか、End Of Frameによるものかの判断はRR2を読み出すことによって行います。Rx FIFOのロックの解除はError Reset コマンド (WR1)を発行することによって行います。RR2をリードする前にError Resetコマンドを発行するとRR2が消失してしまうため、注意が必要です。このモードはフレームの最初の受信データのみ割り込みで転送し、残りはDMAを用いて転送する場合に用います。Rx Intモード(WR2 D5,D4)は“Rx Int On Rx FIFO RCA”に設定して下さい。

10 Rx Int On All Rx Character And Sp,Int

Rx Intモード(WR2 D5,D4)での設定にしたがってRx Intを発生します。Rx Int On Rx FIFO HalfあるいはRx Int On Rx FIFO Fullに設定した場合は、一回のRx Intルーチンで4バイトあるいは8バイトの受信データをまとめて読み出すことができます。Sp IntもRx Int

モードの設定により、以下のように発生の仕方が異なります。

Rx Int On Rx FIFO RCAに設定した場合は、Special Conditionを含む受信データがRx FIFOの出口に表れると(受信データをリードしなくても)Sp Intが発生します。これは“Rx Int On First Rx Character And Sp Int”や“Sp Int Only”の場合と動作が異なるため注意が必要です。また、Sp Int発生時にRx FIFOはロックされないため、受信データをリードすると受信データおよびそれに対応するRR2はRx FIFOから消失します。したがって、Sp Intの割り込みルーチンの中では、受信データをリードする前にRR2をリードしてSpecial Conditionの原因を判断する必要があります。このモードは割り込みを用いて受信データを転送する場合に用います。

Rx Int On Rx FIFO HalfあるいはRx Int On Rx FIFO Fullに設定した場合は、Special Conditionを含む受信データがRx FIFOに入るとただちにSp Intを発生します。Sp Intルーチンの中では、Rx FIFOに存在する受信データのうち、どの受信データにSpecial Conditionが含まれるかをRR2を読み出すことによって判断する必要があります。

11 Sp Int Only

Rx Intは発生しません。Sp Intに関しては“Rx Int On First Rx Character And Sp Int”に設定した場合と全く同様です。このモードは受信データを全てDMAで転送する場合に用います。Rx Intモード(WR2 D5,D4)は“Rx Int On Rx FIFO RCA”に設定して下さい。

D1 Tx Int Mode

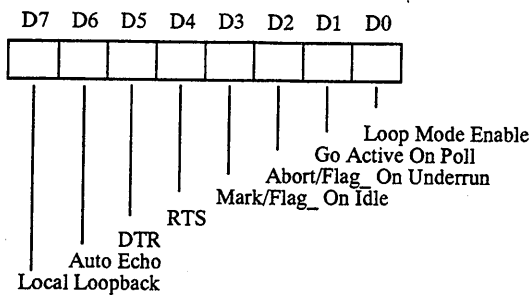
このビットを‘0’に設定するとTx FIFOが完全にEmpty状態になったときにTx Intを発生します。比較的ビットレートが低速のときに、Tx Intの割り込みルーチンの中で1度に2~4byteの送信データをライティングすることにより、Tx Intが発生する回数を減らすことが出来ます。このビットを‘1’に設定するとTx FIFOがFull状態でなくなったときにTx Intを発生します。シリアル・データのビットレートが速く、Tx FIFOが完全にEmptyになってから送信データをライティングしているとUnderrunが発生してしまうような場合

に用います。DMA を用いて送信データの転送を行う場合も、Tx DMA Requestの発生はこのビットに従います。ビットレートが高速の場合は、このビットは‘1’に設定して下さい。いずれの場合もTx Buffer Readyビット (RR1 D2、以下TXBRDYと略す) はTx FIFOがFull状態でないとき、すなわち、まだ送信データがライト出来る状態のときに‘1’になります。

D0 Tx Int Enable

このビットを‘1’にするとTx Int Mode(WR2 D1)の設定にしたがってTx Intを発生します。これはTx Enable、Disableの状態に関係なく発生します。このビットを‘0’にするとTx Intは一切発生しません。

WR3 送信制御



WR3はFlag、Abortの送信およびループモードの制御を行います。

D7 Local Loopback

このビットを‘1’に設定するとTXDHがRXDHに内部でループ・バックされます。外部から入力されるシリアル・データは受信しません。

D6 Auto Echo

このビットを‘1’にするとRXDHに入力されるシリアル・データをそのままTXDHに出力します。RXDHに入力されるシリアル・データは受信可能ですが、送信データ・レジスタにライトした送信データはTXDHから出力されません。

D5 DTR(Data Terminal Ready)

このビットの反転がDTRH_ピンに出力されます。

D4 RTS(Request To Send)

このビットの反転がRTSH_ピンに出力されます。

D3 Mark/Flag_On Idle

Tx Enable 状態でかつフレームの送信が行われてい

ないIdle状態のときにMark(‘1’の連続)を送信するか、Flagを送信するかを設定するビットです。このビットを‘1’に設定するとMarkを送出します。ただし、フレームの始めと終りには Opening Flag、Closing Flag が自動的に付加されます。このビットを‘0’に設定するとIdle状態ではFlagを送出し続けます。

D2 Abort/Flag_On Underrun

Tx Underrunが発生したときにAbort + Flagを送出するか、CRC + Flag を送受するかを設定します。このビットを‘1’に設定すると Underrun 発生時に Abort + Flag を送受します。このビットを‘0’にすると Underrun発生時にCRC + Flagを送出します。通常は送信フレームの最後の送信データをライトするまではこのビットを‘1’にしておき、不用意にUnderrunが発生した際に Abort を送受するように設定しておきます。最後の送信データをライトした時点でこのビットを‘0’にし、正しくCRCが送受されるように設定します。

D1 Go Active On Poll

このビット(以下、GAOPビット)でLoop Modeでの状態の移行を制御します。Loop Mode Enableでゲート遅延状態(RXDHからTXDHへゲート遅延で直結)にあるとき、GAOP='1'でかつEOPパターン('01111111')を受信すると、1ビット遅延状態に移ります。GAOP='0'ではEOPパターンを受信しても1ビット遅延状態には移行しません。GAOP='1'にして1ビット遅延状態に移る場合は、必ずTx Disable状態にして下さい。このタイミングでTx Enableになっている場合の動作は保証しませんので御注意下さい(1ビット遅延で中継してTXDから出力する送信データはTx Enable、Tx Disableに関係しません)。

さらにGAOPビットは1ビット遅延状態からLoop Sending状態への移行も制御します。1ビット遅延状態でEOPパターン受信したとき、GAOP='1'であればLoop Sending状態に移ります。GAOP='0'であれば、Loop Sending状態には移らず、1ビット遅延状態にとどまります。

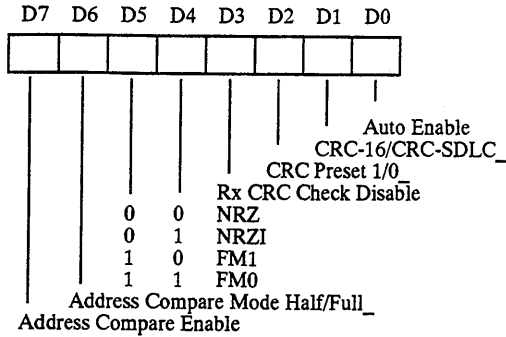
Loop Sending状態から1ビット遅延状態への移行もGAOPビットが制御します。Loop Sendingでのフレーム送りの終りでClosing Flagを送出するときにGAOP='0'ならば、Closing Flag送後後に1ビット遅延状態に移行します。このタイミングでGAOP='1'ならばFlagを送出し続けます。そのまま送信データを書き込むと次のフレームを送出します。

D0 Loop Mode Enable

このビットを'1'にするとLoop Modeに入ります。このビットはRx Enable状態にした後に'1'にして下さい。Rx Disable状態でこのビットを'1'にすると正しくループモードに移行できない可能性がありますので御注意下さい。ゲート遅延状態→1ビット遅延状態→Loop Sending状態へと正しく移行するためには、このビットを'1'にしておく必要があります。Loop Modeから抜ける場合は、1ビット遅延状態でこのビットを'0'にする必要があります。このビットを'0'にしたときにHunt状態であればただちに1ビット遅延状態から抜けて通常モードに戻ります。Hunt状態でなければHunt状態になるのを待って1ビット遅延状態から抜けます。

Loop Sending状態でこのビットを'0'にした場合の動作は保証しません。必ず一度1ビット遅延状態に戻ってから、このビットを'0'にして下さい。

WR4 受信制御、送受信共通



WR4 は受信アドレスの比較を行うか否かの制御、データ・エンコーディング・モードの設定、CRCの計算式の設定等を行います。

D7 Address Compare Enable

このビットが‘1’のときはFlagの次の受信アドレスをチェックし、WR5 で設定したアドレスと一致するフレームのみを受信します。ただし、アドレス‘FFH’のみは例外で、‘FFH’で始まるフレームは無条件に受信します。このビットが‘0’のときはFlagの次の受信アドレスの値に関係なく全てのフレームを受信します。

D6 Address Compare Mode Half/Full_

このビットが‘1’でかつAddress Compare Enable=‘1’

のときは、WR5 で設定したアドレスの上位4ビットのみで受信アドレスの判定を行います。このビットが‘0’の場合は受信アドレスとWR5で設定したアドレスが全ビット一致しない限り、フレームの受信は行われません。また、Address Compare Enable=‘0’のときはこのビットはKP91Aの動作に影響を与えません。

D5,D4 Data Encoding Mode

送受信のデータのエンコーディング・モードを設定します。これらのビットを設定すると、Tx Enable, Disable の状態に関係なく、ただちに設定したエンコーディング・モードになります。4つのエンコーディング・モードを図11-2に示します。

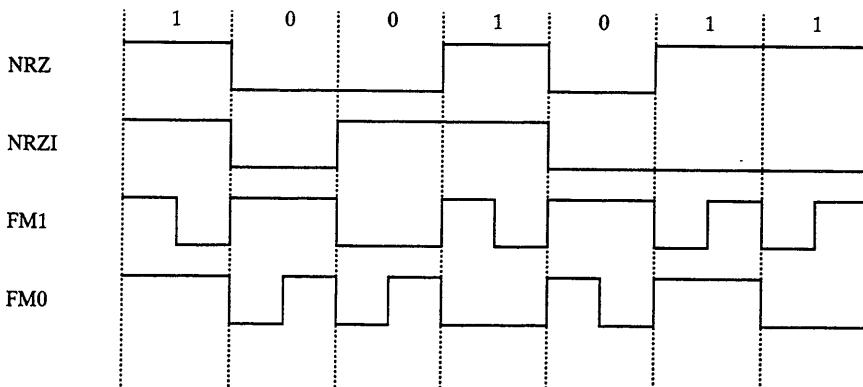


図 11-2 データ・エンコーディング・モード

00 NRZ

送受信するデータが '0' のときは、1ビット期間中 TXDH="L" あるいは RXDH="L" になります。送受信するデータが '1' のときは、1ビット期間中 TXDH="H" あるいは RXDH="H" になります。

01 NRZI

送受信するデータが '0' のときは、TXDH または RXDH は1つ前のビットの値の反転になります。送受信するデータが '1' のときは、TXDH または RXDH は1つ前のビットの値を保持します。

10 FM1

送受信するデータが '0' のときは、ビット境界のみで値を反転させます。送受信するデータが '1' のときは、ビット境界およびビットの中央で値を反転させます。

11 FM0

送受信するデータが '0' のときは、ビット境界およびビットの中央で値を反転させます。送受信するデータが '1' のときは、ビット境界のみで値を反転させます。

D3 RxCRC Check Disable

このビットを '1' にすると受信側での CRC チェックを行いません。Opening Flag と Closing Flag ではさまれた部分を全て受信データとして扱いたい場合に用います。このビットを '0' にすると Closing Flag 直前の

2 byte は CRC とみなし、CRC チェックを行います。CRC チェックの結果、エラーが発生した場合には CRC Error ビット (RR2 D6) がセットされます。

D2 CRC Preset 1/0

送信側の CRC Generator および受信側の CRC Checker の初期値を設定します。このビットを '1' に設定すると CRC Generator および CRC Checker の初期値は All '1' に設定されます。このビットを '0' に設定すると CRC Generator および CRC Checker の初期値は All '0' に設定されます。

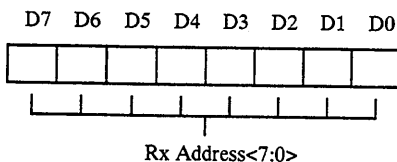
D1 CRC-16/CRC-SDLC

CRC Generator および CRC Checker の生成多項式を選択します。このビットが '1' のときは生成多項式として $CRC-16(X^{16}+X^{15}+X^2+1)$ が使用されます。このビットが '0' のときは生成多項式として $CRC-CCITT(X^{16}+X^{12}+X^5+1)$ が使用されます。

D0 Auto Enable

このビットが '1' の場合は、CTSH_端子="L" かつ Tx Enable コマンド (WR1) が発行されているときに送信可能になり、DCDH_端子="L" かつ Rx Enable コマンドが発行されているときに受信可能になります。このビットが '0' の場合は DCDH_、CTSH_ 端子の状態は送信制御、受信制御に関係なく、WR1 のコマンドだけで制御できます。なお、Local Loopback モードのときは、このビットの設定は無効で、WR1 のコマンドだけで送信イネーブル、受信イネーブルの制御を行います。

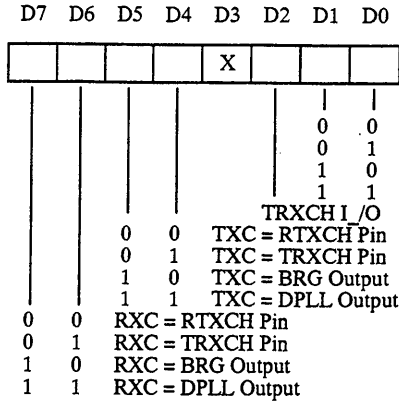
WR5 受信アドレス



Address Compare Enable='1' (WR4 D7) に設定したときは、受信フレームの Opening Flag の次の 1 バイトが、このレジスタで設定した値と一致した場合のみフ

レームの受信を行います。Address Compare Mode Half/Full_='1' (WR4 D6) に設定した場合は上位 4 ビットのみを比較します。Address Compare Mode Half/Full_='0' に設定した場合は 8 ビット全てを比較します。Address Compare Enable='0' の場合は、このレジスタの値に関係なく全てのフレームを受信します。なお、このレジスタで設定した値は、送信フレームのアドレスには全く関係ありません。送信フレームのアドレスはデータとして送出する必要があります。

WR6 クロック選択



TRXCOUT = Receive Clock
 TRXCOUT = Transmit Clock
 TRXCOUT = BRG Output
 TRXCOUT = DPLL Output

WR6 は受信クロック、送信クロックとして外部入力、BRG出力、DPLL出力のいずれを用いるかを選択します。

D7,D6 受信クロック選択

00 RXC=RTXCH Pin

受信クロックとして外部ピンの RTXCH 入力を用います。

01 RXC=TRXCH Pin

受信クロックとして外部ピンの TRXCH 入力を用います。

10 RXC=BRG Output

受信クロックとしてポーレートジェネレータの出力を用います。

11 RXC=DPLL Output

受信クロックとしてDPLLの出力を用います。

D5,D4 送信クロック選択

00 TXC=RTXCH Pin

送信クロックとして外部ピンの RTXCH 入力を用います。

01 TXC=TRXCH Pin

送信クロックとして外部ピンの TRXCH 入力を用い

ます。

10 TXC=BRG Output

送信クロックとしてポーレートジェネレータソース・クロックの出力を用います。

11 TXC=DPLL Output

送信クロックとしてDPLLの出力を用います。

D3 このビットへの書き込みはKP91Aの動作に影響を与えません。

D2 TRXCH Output/Input_

TRXCIO_ ピンの極性切り替えを制御します。このビットが'1'でかつWR6のD7~D4で受信クロックまたは送信クロックとしてTRXCH端子が選択されていない場合に、TRXCIO_="L" となります。それ以外の場合は TRXCIO_="H" となります。TRXCIO_ ピンはTRXCHとTRXCOUTを双方向バッファに接続する際の出カインエーブル信号として用います。

D1,D0 TRXCOUT選択

TRXCH 端子を出力に設定したときに出力する信号を選択します。

00 TRXCOUT=Receive Clock

WR6のD7、D6で選択されている受信クロックを出

力します。

01 TRXCOUT=Transmit Clock

WR6のD5、D4で選択されている送信クロックを出力します。

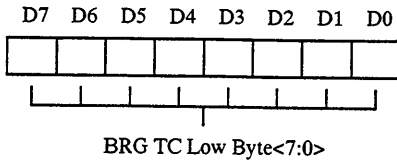
10 TRXCOUT=BRG Output

ポーレートジェネレータのOutputを出力します。

11 TRXCOUT=DPLL Clock

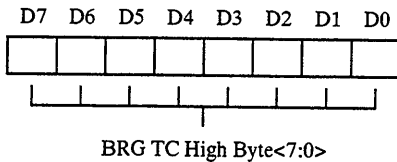
DPLLのOutputを出力します。DPLLをFMモードに設定した場合は DPLL の受信クロックが出力されま
す。

WR7 BRG Time Constant Low Byte



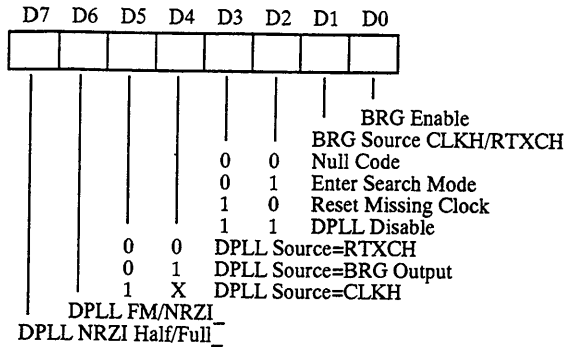
WR7 はポーレートジェネレータの Time Constant の下位バイトを設定します。このレジスタは初期化されませんので、ポーレートジェネレータをイネーブルにする前には必ずこのレジスタの設定を行う必要があります。

WR8 BRG Time Constant High Byte



WR8 はポーレートジェネレータの Time Constant の上位バイトを設定します。このレジスタは初期化されませんので、ポーレートジェネレータをイネーブルにする前には必ずこのレジスタの設定を行う必要があります。

WR9 DPLL, BRG 制御



WR9はポーレートジェネレータ (BRG)、DPLLの制御を行います。

D7 DPLL NRZI Half/Full_

NRZI モードのときに、DPLL の Source Clock として、シリアルデータの 1 ビット期間の何倍のクロックを入力するかを設定します。このビットを '0' に設定した場合は、シリアルデータのビットレートの32倍の周波数のクロックをDPLLのSource Clockとして用います。このビットを '1' に設定した場合は、シリアルデータのビットレートの 16 倍の周波数のクロックをDPLLのSource Clockとして用います。

FMモードのときはこのビットは必ず '0' に設定して下さい。

D6 DPLL FM/NRZI_

DPLL の動作モードの設定を行います。このビットを '0' に設定した場合はNRZIモードになります。このビットを '1' に設定した場合はFMモードになります。

D5,D4 DPLL Source Clock選択

00 DPLL Source=RTXCH

DPLL のソース・クロックとして RTXCH の入力を用います。

01 DPLL Source=BRG Output

DPLLのソース・クロックとしてBRGの出力を用います。

1x DPLL Source=CLKH

KP91A では、基本動作クロックとしてシステムク

ロックの 2 倍の周波数のクロックを内部で使用することができます。

この設定時には、DPLL のソース・クロックとして KP91Aの基本動作クロックCLKHを用います。

D3,D2 DPLL制御コマンド

00 Null Code

KP91Aの動作には影響を与えません。

01 Enter Search Mode

DPLLがディセーブル状態のとき(リセット後、もしくはDPLL Disable コマンドを発行した後)にこのコマンドを発行すると RXDH のエッジの検出を開始します。DPLL 動作中にこのコマンドを発行すると、新たに RXDH のエッジ検出を開始します。Missing Clock ビット (RR9 D7,D6) がセットされているときにこのコマンドを発行すると、これらのビットをリセットします。

10 Reset Missing Clock

DPLL が FM モードで動作している際には、RXD のエッジを見失うとMissing Clockビット(RR9 D7,D6)をセットします。これらのビットをリセットするのにこのコマンドを用います。

11 DPLL Disable

動作中の DPLL を停止させます。このとき Missing Clock ビットがセットされていれば、これらのビットをリセットします。

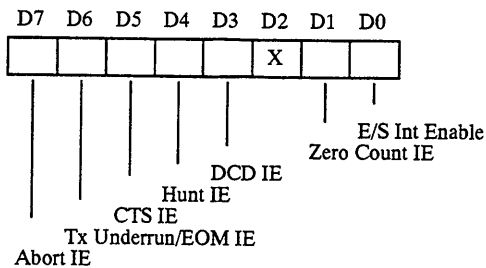
D1 BRG Source CLKH/RTXCH

このビットが‘1’のときBRGのソース・クロックは基本動作クロック CLKH になります。このビットが‘0’のときBRGのソース・クロックはRTXCH入力になります。ハードウェア・リセット後は BRG のソース・クロックはRTXCH入力に設定されます。

D0 BRG Enable

このビットを‘1’に設定すると BRG Counter がカウント・ダウンを開始します。BRG の動作を開始させる前には必ずWR7、WR8のBRG Time Constantを設定する必要があります。

WR10 External/Status 割り込み制御



WR10はExternal/Status Intのイネーブルを制御します。

D7 Abort IE

このビットを‘1’に設定するとAbortビット(RR1 D7)が‘0’→‘1’、および‘1’→‘0’に変化する際にE/S Intを発生します。このビットはハードウェア・リセットおよびソフトウェア・リセットで‘1’に初期化されます。

D6 Tx Underrun/EOM IE

このビットを‘1’に設定するとTx Underrun/EOMビット(RR1 D6)が‘0’→‘1’に変化する際にE/S Intを発生します。このビットはハードウェア・リセットおよびソフトウェア・リセットで‘1’に初期化されます。

D5 CTS IE

このビットを‘1’に設定するとCTSビット(RR1 D5)が‘0’→‘1’、および‘1’→‘0’に変化する際にE/S Intを発生します。このビットはハードウェア・リセットおよびソフトウェア・リセットで‘1’に初期化されます。

D4 Hunt IE

このビットを‘1’に設定するとHuntビット(RR1 D4)が‘0’→‘1’、および‘1’→‘0’に変化する際にE/S Intを発生します。このビットはハードウェア・リセットおよびソフトウェア・リセットで‘1’に初期化されます。

D3 DCD IE

このビットを‘1’に設定するとDCDビット(RR1 D3)が‘0’→‘1’、および‘1’→‘0’に変化する際にE/S Intを発生します。このビットはハードウェア・リセットおよびソフトウェア・リセットで‘1’に初期化されます。

D2 このビットへの書き込みはKP91Aの動作に影響を与えません。

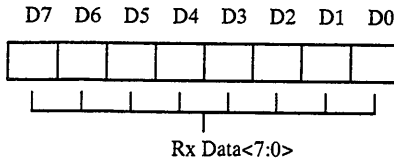
D1 Zero Count IE

このビットを‘1’に設定するとZCビット(RR1 D1)が‘0’→‘1’に変化する際にE/S Intを発生します。このビットはハードウェア・リセットおよびソフトウェア・リセットで‘1’に初期化されます。

D0 E/S Int Enable

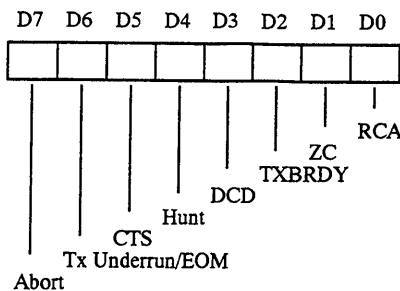
E/S IntはWR10のD7～D3、D1の設定でイネーブルされる割り込み要因の論理和で発生します。このビットは論理和をとった後のマスターのE/S Intの発生を制御します。このビットが‘1’のとき、各要因の論理和でE/S Intを発生します。このビットが‘0’のときはE/S Intは発生しません。ただし、このビットが‘0’の場合もRR1に反映される各要因のラッチ機能は働きます。割り込み要因のラッチ機能については「11.8 割り込み」を参照して下さい。

RR0 受信データ・レジスタ



RXD から入力されるシリアル・データは受信シフト・レジスタでシリアル→パラレル変換されたあと、Rx FIFO に転送されます。Rx FIFO にある受信データはこのレジスタから読み出すことができます。

RR1 送受信ステータス



送信、受信の各ステータスはこのレジスタから読み出すことができます。Abort、Tx Underrun/EOM、CTS、Hunt、DCDについては、WR10でE/S Intの要因として設定するかしないかで、読み出した時点での状態が読み出されるか、ラッチされた状態が読み出せるかが異なりますので注意が必要です。例えば、Abort IE と Tx Underrun/EOM IE が '1'、CTS IE、Hunt IE、DCD IE が '0' に設定されている場合は、CTS、Hunt、DCDについては読み出した時点での状態が読み出されますが、Abort と Tx Underrun/EOM については割り込み要因となる変化(Abortの '0'→'1'、'1'→'0'およびTx Underrun/EOMの '0'→'1'の変化)が生じた時点でラッチされた状態が読み出されます。割り込み要因となる変化が生じていない場合はその時点での状態が読み出されます。

D7 Abort

Abort IE='0'のときは7個以上の連続した'1'を受信したときにこのビットは'0'→'1'に変化し、'0'を受信すると直ちにこのビットは'1'→'0'に変化します。Abort IE='1'のときは7個以上の連続した'1'を受信し

たときにこのビットは'0'→'1'に変化し、その状態がラッチされるとともに、E/S Intを発生します。その後 Reset External/Status Intコマンド(WR1)を発行するまでは、受信ブロック内部で Abort 状態がなくなってもこのビットは'1'を保持します。Reset External/Status Int コマンドを発行するとラッチは解除されて、その時点でAbort状態でなければ、このビットは'1'→'0'に変化します。この変化により、再び AbortによるE/S Intが発生します。Abort ビットは Tx Underrun/EOM、CTS、Hunt、DCD 等のビットと次の点で異なるため、注意が必要です。Tx Underrun/EOM、CTS、Hunt、DCD 等は E/S Int ラッチが閉じている状態では、割り込み要因となる変化があっても Reset External/Status Int コマンドを発行した時点で元の状態と同じであれば、さらなるE/S Intは発生しません。ところが、Abort ビットだけはE/S Int ラッチが閉じている状態でもビットの変化を記憶する機構を備えており、ビットの変化があった場合には Reset External/Status Int コマンドを発行した後に再びE/S Intが発生します。例えば、Abort IEとTx Underrun/EOM IEが'1'のとき、Tx Underrun/EOM '0'→'1'でE/S Intが発生し、

Abort が '0' でラッチされたとします。このあと、Abort ビットが '0' → '1' → '0' と変化した後、ResetExternal/Status Int コマンド発行すると、まず Abortの'0'→'1'の変化によって再びE/S Intが発生します。さらに Reset External/Status Int コマンドを発行すると、今度はAbortの'1'→'0'の変化によるE/S Intが発生します(図 11-11 参照)。なお、いずれの場合も Rx Enable状態になっていない限り、Abortビットは'1'になりません。詳しくは「11-8 割り込み」を参照下さい。

D6 Tx Underrun/EOM

フレーム送信中あるいは送信終了時に、送信ブロック内部に送信すべきデータがなくなったとき(この状態を Underrun/EOM 状態といいます。)にこのビットが'1'になります。Underrun/EOM状態になる前にこのビットが'0'で Underrun/EOM の発生によって'0'→'1'に変化する場合にはのみCRCあるいはAbortが送出されます。Underrun/EOM 状態になったときにこのビットが'1'のまま変化しない場合は、CRCあるいは Abort は送出されず、送信データに続いて Closing Flag が送出されます。すなわち、CRCあるいは Abort を Closing Flag の前に送出するためには、Underrun/EOM 状態になる前にこのビットをリセットしておく必要があります。このビットのリセットは Reset Tx Underrun/EOM Latch コマンド (WR1) で行います。また、Tx Disable コマンドおよび Send Abort コマンド(いずれも WR1) を発行した場合にもこのビットは'1'になります。このビットは'0'→'1'の変化のみE/S Int の要因になります。これは、Tx Disable コマンドおよび Send Abort コマンドで'0'→'1'に変化した場合も同様です。

D5 CTS(Clear To Send)

外部ピン CTSH₋ の状態の反転がこのビットに表れます。CTS IE(WR10 D5)が'1'のときはこのビットの'0'→'1'、'1'→'0'の変化がE/S Int の要因になります。E/S Int Latch が解放されている状態でこのビットに'0'→'1'の変化があった場合(CTSH₋端子が'H'→'L'に変化した場合)、E/S Int が発生し、CTS='1'がラッチされます。Reset External/Status Int コマンドによりE/S Int Latchを解放する時点でCTSH₋端子='L'で

あれば、さらなる E/S Int は発生しません。CTSH₋='H'に変化していた場合は、CTS ビットの'1'→'0'の変化によってE/S Int が再び発生します。このビットはAbortビットのようにE/S Int Latchが閉じている間に生じた変化を記憶する機能はありません。

D4 Hunt

Flag 同期がとれていないときに、このビットは'1'になります。アイドル状態でFlagを受信しているときやフレーム受信(Opening Flag から Closing Flag まで)には Hunt='0'になります。また、Rx Disable コマンド、Enter Hunt Mode コマンド発行時、および Abort 受信によっても Hunt='1'になります。Hunt IE(WR10 D4)が'1'のときは、このビットの'0'→'1'、'1'→'0'の変化がE/S Int の要因になります。E/S Int 関連の動作についてはCTSビットと全く同様です。

D3 DCD(Data Carrier Detect)

外部ピン DCDH₋ の状態の反転がこのビットに表れます。E/S Int 関連の動作についてはCTSビットと全く同様です。

D2 TXBRDY(Tx Buffer Ready)

Tx FIFOがFull状態でないときにこのビットは'1'になります。すなわち、このビットは Tx FIFO に送信データが上書きされずに書けることを示します。Tx Int Mode(WR2 D1)が'1'のときは、TXBRDY='1'と Tx Int の発生は一致しますが、Tx Int Mode='0'の場合は両者の動きは必ずしも一致しません。また、Tx DMA Requestも Tx Int Modeの設定に従うため、TXBRDYの動きと一致しない場合があります。

D1 ZC(Zero Count)

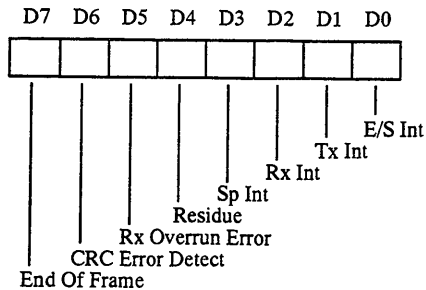
Zero Count IE(WR10 D1)が'0'のときはこのビットは常に'0'になります。Zero Count IE='1'の場合はBRG カウンタがZero Count状態になったときにこのビットが'1'になります。Zero Count IE='1'のときは、このビットの'0'→'1'の変化でE/S Int が発生し、他のE/S Int の要因をラッチします。ただし、このビット自体はラッチされる機能がありません。すなわち、Zero Count IE='1'の場合でもこのビットは常にRR1を読み出した時点での状態が反映されます。

D0 RCA(Receive Character Available)

Rx FIFOに読み出し可能な受信データが存在するときにこのビットが'1'になります。Rx Int On Rx FIFO RCAモード(WR2)のときは、Rx Intの発生とRCAビットの動きは一致しますが、その他のモードのときはRCA='1'とRx Intの発生は必ずしも一致しません。RCAビットは常にRx FIFOに読み出し可能なデータがあるかどうかを示します。したがって、Rx Intの割り

込みルーチン中で所定のバイト数(Rx Int On Rx FIFO RCAモードのときには1バイト、Rx Int On Rx FIFO Halfモードのときは4バイト、Rx Int On Rx FIFO Fullモードのときには8バイト)の受信データを読み出した後、割り込みルーチンからリターンする前にもう一度RCAビットをチェックすることにより、Rx Intが発生する頻度を低くすることができます。

RR2 受信エラー・ステータス



フレーム受信終了時の各種ステータスとオーバーラン・エラーのステータスを示します。RR2のステータスはRR0の受信データと対でRx FIFOに保持され、Rx FIFOの出口にある受信データに対応するステータスがRR2を通して読み出されます。Rx FIFOのリード・ポインタはRR0のリードによって動くため、Sp Intルーチン中で、ある受信データに対応するステータスを読み出す必要があるような場合は、受信データの読み出しの前にRR2を読み出さなければなりません。ただし、Rx Int On First Rx Character And Sp IntモードやSp Int Onlyモードの場合は、Sp Intを含む受信データを読み出したときにRx FIFOがロックされるため、このような必要はありません。

D7 End Of Frame

このビット(以下、EOFビット)が'1'のとき、受信ブロックがClosing Flagを検出したことを示します。EOFビットはError Resetコマンドまたは次のフレームのデータを受信することにより、リセットされます。

Sp Intの発生とEOFビットの関係はRx Intモード(WR2 D5,D4)とRx Int、Sp Int制御(WR2 D3,D2)の設定状態によって異なります。

Rx Int On Rx FIFO RCAモードに設定した場合は、Sp Intの発生とEOFビットの関係は次のようになります。Rx Int On All Rx Character And Sp IntモードのときはEOFに対応する受信データ(Closing Flagの直前のデータ)がRx FIFOの出口に表れると直ちにSp Intを発生します。Rx Int On First Rx Character And Sp IntモードまたはSp Int OnlyモードのときはEOFに対応する受信データを読み出した後にSp Intを発生します。これらのモードでSp Intが発生するとRx FIFOはロックされ、Rx FIFOの出口にはEOFに対応する受信データが保持されます。Rx FIFOのロックはError Resetコマンドを発行することで解除されます。

Rx Int On Rx FIFO HalfモードあるいはRx Int On Rx FIFO Fullモードに設定した場合は、Sp Intの発生とEOFビットの関係は次のようになります。ただし、Rx Int、Sp Int制御の設定はRx Int On All Rx Character

And Sp Int モードになっていることを前提とします。この場合、EOFに対応する受信データが受信シフトレジスタから Rx FIFO に転送されると、ただちに Sp Int が発生します。この場合に限り、EOFビット='0'にもかかわらず、EOFによる Sp Int が発生します。Sp Int ルーチンではどの受信データが EOF に対応する受信データかを判断するために、受信データを読み出す前に必ずRR2を読み出す必要があります。

D6 CRC Error Detect

フレーム受信終了時にCRC Errorが検出されると'1'になります。このビットはEOFビットが'1'のときのみ有効です。このビットが'1'のときは、CRCの2バイト目は受信した値が読めません（CRCの2バイト目に相当する受信データは'C0H'または'E0H'になります→表6-1参照）。また、このビットはError Reset コマンドではリセットされず、次のフレームの受信を始めると自動的に'0'になります。

D5 Rx Overrun Error

受信データに Overrun Error が発生したとき'1'にな

ります。このビット(以下、OVREビット)が'1'のときはRR0から読めるデータは'20H'、'E0H'、'A0H'のいずれかとなり、Overrun で上書きされた受信データを読むことは出来ません(表11-4参照)。OVREビットはError Reset コマンドを発行しない限り、リセットされません。Sp Int と OVRE ビットの関係は、Sp Int と EOFビットの関係と全く同様です。詳しくはEOFビットの説明の項を参照して下さい。

D4 Residue

バイト単位になっていない端数フレームを受信したときに'1'になります。このビットもCRC Errorビットと同様に EOF='1' のときのみ有効です。ただし、EOF='1'のときにOverrun ErrorやCRC Errorが発生しているときは端数フレームであっても Residue='1' にはなりません。Residue='1' のとき、端数ビットが何ビットであったかの情報はRR0に受信データとともに図 11-3 のように書き込まれます。したがって、端数ビットが何ビットであったかは、図11-3の●以外の部分をソフトウェアでデコードして決定する必要があります。

最終データ	データ領域	Residue
8ビット	●●●●●●●●	0
7ビット	●●●●●●● 0	1
6ビット	●●●●●● 0 1	1
5ビット	●●●●● 0 1 1	1
4ビット	●●●● 0 1 1 1	1
3ビット	●●● 0 1 1 1 1	1
2ビット	●● 0 1 1 1 1 1	1
1ビット	● 0 1 1 1 1 1 1	1

● : 受信データ

図11-3 端数ビットとRR0のデータの関係

D3 Sp Int

SPINTピンの状態をこのビットから読み出すことができます。KP91Aが出力する4つの割り込みの論理和をCPUの割り込み端子に接続し、割り込み要因はCPUからのポーリングで調べるような場合にこのビット(D2~D0も同様)を使用します。

D2 Rx Int

RXINTピンの状態をこのビットから読み出すこと

ができます。

D1 Tx Int

TXINTピンの状態をこのビットから読み出すことができます。

D0 E/S Int

ESINTピンの状態をこのビットから読み出すことができます。

RR3 WR3 の設定内容の読み出し

WR3に設定した値が読み出されます。

RR4 WR4 の設定内容の読み出し

WR4に設定した値が読み出されます。

RR5 WR5 の設定内容の読み出し

WR5に設定した値が読み出されます。

RR6 WR6 の設定内容の読み出し

WR6に設定した値が読み出されます。ただし、D3は常に'0'です。

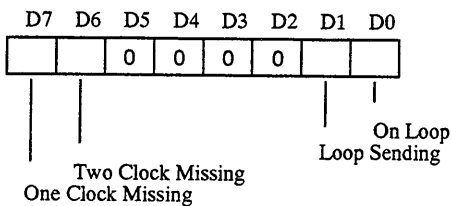
RR7 WR7 の設定内容の読み出し

WR7に設定した値が読み出されます。

RR8 WR8 の設定内容の読み出し

WR8に設定した値が読み出されます。

RR9 DPLL ステータス、ループ・モード・ステータス



DPLL の Missing Clock のステータスおよびループ・モードのステータスを示します。

D7 One Clock Missing

DPLLがFMモードで動作している場合に、RXDの変化を1回見失うとこのビットが'1'になります。このビットのリセットはMissing Clock Resetコマンド(WR9)で行います。また、Enter Search Mode コマンドおよびDPLL Disable コマンド (いずれも WR9) を発行した場

合にもこのビットはリセットされます。DPLLがNRZIモードで動作している場合はこのビットは常に'0'です。

D6 Two Clock Missing

DPLLがFMモードで動作している場合に、RXDの変化を2回連続して見失うとこのビットが'1'になります。このビットが'1'になるとDPLLは自動的にサーチ・モードに入り、RXDHの新たなエッジの検出を開

始します。このビットのリセットは Missing Clock Reset コマンド (WR9) で行います。また、Enter Search Mode コマンドおよび DPLL Disable コマンド (いずれも WR9) を発行した場合にもこのビットはリセットされます。DPLLがNRZIモードで動作している場合はこのビットは常に'0'です。

D5~D2 これらのビットは常に '0' が読み出されます。

D1 Loop Sending

ループ・モードで Opening Flag 送出中あるいはデータ送出中に '1' になります。1 ビット遅延状態で EOP パターン '11111110' を検出したときに GAOP(WR3 D1)= '1' であれば、このビットは '0' → '1' に変化します。また、ループ・モードでのデータ送出の終りで

Closing Flag を送出するときに GAOP='0' であれば、このビットは '1' → '0' に変化します。

D0 On Loop

ループ・モードで 1 ビット遅延状態あるいは Loop Sending 状態にあるときに '1' になります。ゲート遅延状態 (RXDH から TXDH にゲート遅延だけでつながっている状態) で EOP パターンを検出したときに GAOP='1' であれば、このビットは '0' → '1' に変化します。1 ビット遅延状態のときに、Loop Mode Enable (WR3 D0)= '0' にすると、このビットは '1' → '0' に変化し、ループモードから抜けます。ただし、Hunt 状態にあるときは Loop Mode Enable (WR3 D0)= '0' にするとただちにループモードから抜けますが、Hunt 状態でない場合は Hunt 状態になるのを待ってループモードから抜けます。

RR10 WR10 の設定内容の読み出し

WR10 に設定した値が読み出されます。ただし、D2 は常に '0' です。

RR2' WR2 の設定内容の読み出し

WR2 に設定した値が読み出されます。

RR9' WR9 の設定内容の読み出し

WR9 に設定した値が読み出されます。ただし、D3、D2 は常に '0' です。

11.6 送受信動作の説明

HDLCのフレーム形式は一般に図11-4のようになっています。

フラグ "01111110"	アドレス 8ビット	制御部 8ビット	情報フィールド 任意長	CRC 16ビット	フラグ "01111110"
-------------------	--------------	-------------	----------------	--------------	-------------------

図11-4 HDLCフレームの形式

KP91A の送信部は HDLC プロトコルの中の次の機能をサポートしています。

- Opening Flag、Closing Flag送出
- Flag以外のフィールドでのゼロ挿入
- CRCコード生成、送出

送信部はアドレス、制御部、情報フィールドを区別しません。アドレス、制御部で送るべきビット列はソフトウェアで予め設定した上で、情報フィールドと同じ扱いで送信データとして送出する必要があります。CRCはUnderrunが発生する前にTx Underrun/EOMビット(RR1 D6)を'0'にしておけば、自動的に送出されます。Tx Underrun/EOMビットが'1'のまま Underrun が発生すると、CRC は送出されず情報フィールドに続いてClosing Flagが送出されます。一般に HDLC プロトコルでは任意のビット数の情報フィールドの送受信を想定していますが、KP91A はバイト単位での送信しかサポートしていないため注意が必要です。

KP91A の受信部は HDLC プロトコルの中の次の機能をサポートしています。

- Flagパターン検出によるFlag同期
- Flag以外のフィールドでのゼロ削除
- アドレス比較機能
- CRCコード検査

受信部はアドレス、制御部とも情報フィールドと同じように受信データとして扱います。また、2バイトの CRC コードもデータとして受信されます。データとして受信したアドレスや CRC が不要な場合

はソフトウェアで破棄する必要があります。

KP91A は任意のビット数のフレームが受信できます。バイト単位でない端数ビットのフレームを受信した場合は、最後の受信データに端数ビットに関する情報が書き込まれますので、端数ビットが何ビットであったかはソフトウェアで判断する必要があります。

HDLC フレームの送信

アイドル・パターン

送信クロック、データ・エンコーディング・モードの設定を行った後、Tx Enable状態にすると、Mark/Flag_On Idleビット(WR3 D3)にしたがってMark(連続した'1')あるいはFlagパターンを送出します。この状態で送信データ・レジスタ WR0 に送信データをライトすると、ライトされたパラレル・データは送信シフト・レジスタでパラレル→シリアル変化された後、TXDH から送出されます。Mark/Flag_On Idle='1'の場合もOpening Flagは自動的に付加されるため、Mark/Flag_On Idle='0'にしてから送信データをライトする必要はありません。

割り込みによる送信データ転送

Tx Intには2つのモードがあり、Tx Int を発生する条件が異なります。Tx Int Mode(WR2 D1)を'0'に設定するとTx FIFOが完全にEmptyになったときにTx Intを発生します(以下、このモードを完全Emptyモードと呼ぶことにします)。一方、Tx Int Mode='1'に設定すると、Tx FIFOの入口がEmptyになり、送信データが

少なくとも1つライト可能になったときにTx Intを発生します(以下、このモードをEL Emptyモードと呼びます。ELはEntry Locationの略です)。完全Emptyモードはシリアル・データの転送速度が比較的遅い場合に、割り込みによるオーバーヘッドを少なくするために用います。EL Emptyモードはシリアル・データの転送速度が速い場合に、常にTx FIFOをFullに近い状態に保つことで送信アンダーランが発生しないようにするために用います。

いずれのモードの場合も、リセット後、最初の送信データをライトするまではTx Intは発生しません。その後は、Tx FIFOが所定の状態になるたびにTx Intを発生します。また、Tx Intの発生はTx Enable, Disableとは無関係で、Tx Int Enable(WR2 D0)とTx FIFOの状態のみで決まります。フレームの最後の送信データをライトした後は、Reset Tx Intコマンド(WR1)を発行してTx Intをリセットします。Tx FIFO内に送信データがなくなるとTx Underrun/EOM状態になり、Tx Underrun/EOMビット(RR1 D6)が'1'になります。Tx Underrun/EOM状態になる前にTx Underrun/EOMビットが'0'にリセットされており、Tx Underrun/EOM発生で'0'→'1'に変化する場合には、その時点のAbort/Flag_On Underrunビット(WR3 D2)の値によって、(Abort + Closing Flag)もしくは(CRC + Closing Flag)が送出されます。通常は所定のバイト数の送信データを転送するまではAbort/Flag_On Underrun='1'にしておき、フレーム送信の途中でUnderrunが発生した場合にAbortを送出するようにします。そして、フレームの最後の送信データを転送した後にAbort/Flag_On Underrun='0'として、正しくCRCが送出されるようにします。Tx Underrun/EOM状態になる前にTx Underrun/EOMビットがリセットされておらず、'1'のまま変化がない場合には、最後の送信データに続いてClosing Flagが送出されます。言い換えると正しくCRCを送出するためにはTx Underrun/EOM発生前にReset Tx Underrun/EOM Latchコマンド(WR1)を発行してTx underrun/EOMビットをリセットしておく必要があります。このコマンドはTx Enable状態であればいつ発行しても構いません。ただし、Auto Enable(WR4 D0)に設定した場合は、Tx Enableコマンドを発行した後もCTSH_="L"になら

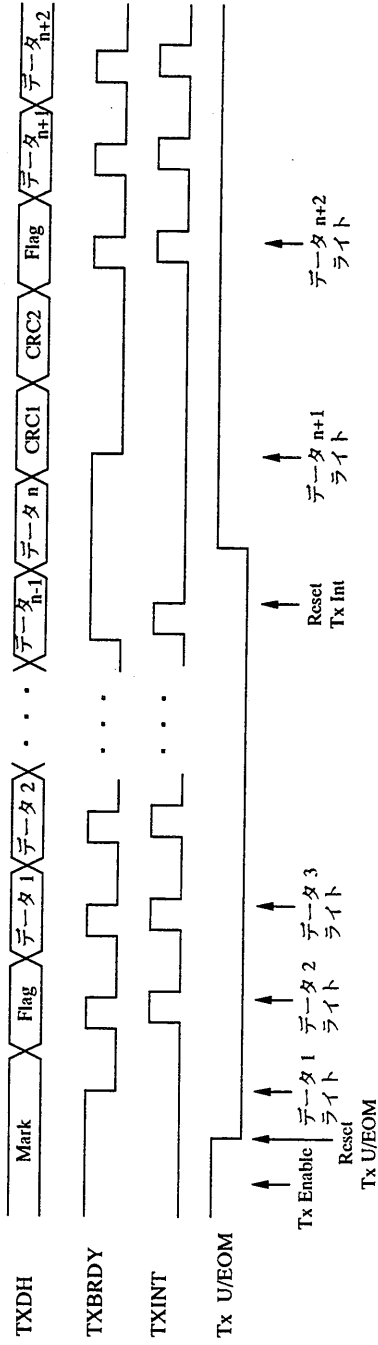
なければ、Reset Tx Underrun/EOM LatchコマンドでTx Underrun/EOMビットをリセットすることは出来ませんので御注意下さい。

最終の送信データをライトし終って、Reset Tx Intコマンドを発行した後、再びTx Intが発生するタイミングは次の2通りにわかれます。Back To Backフレーム(あるフレームのClosing Flagと次のフレームのOpening Flagを共有する送信パターン)送信のために、Tx Underrun/EOMによるE/S Intルーチン中で次のフレームの送信データをライトする場合は、その後Tx FIFOが所定の状態(完全EmptyモードかEL Emptyモードかで異なる)になったときに再びTx Intが発生します。CRCの2バイト目を送出するまでに次のフレームの送信データをライトしない場合は、CRCの2バイト目を送出するタイミングで再びTx Intが発生します。

CRCを送出しない場合は、Closing Flagの送出を開始する直前にTx Intが発生します。

図11-5にTXINTとTXBRDYビット(RR1 D2)の動きを示します。ケース1はTx Underrun/EOMによるE/S Intルーチン中で次のフレームの送信データをライトする場合の例、ケース2はCRCの2バイト目を送出するまでに次のフレームの送信データをライトしない場合の例です。また、ケース1、ケース2ともMark/Flag_On Idle='1'、完全Emptyモードに設定した場合の例です。

ケース 1



ケース 2

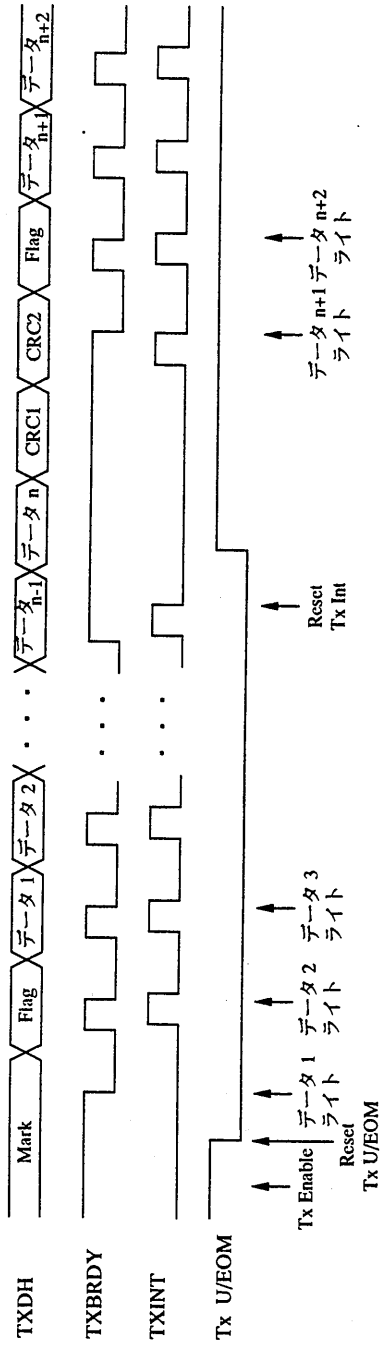


図 11-5 TXBRDY ビットと TXINT の動き

DMAによる送信データ転送

Tx DMA Requestの発生はTx Intのモード設定(WR2 D2)に従います。シリアル・データの転送速度が非常に速い場合 (CLK91の1/4に近い場合)は、EL Emptyモードに設定して下さい。通常は、完全Emptyモードでも十分に転送可能です。

初期設定を行った後、Tx DMA Request Enable (WR2 D6)='1'にすると、ただちにTx DMA Requestが発生します。したがって、Tx DMA Request Enable='1'にする前に必ずDMAコントローラに転送アドレス、転送バイト数を設定しておく必要があります。また、Tx DMA RequestはTx Enable、Tx Disableとは無関係で、Tx DMA Request EnableとTx FIFOの状態のみで決まります。

所定のバイト数をDMAコントローラが転送し終ると、DMAコントローラはTx DMA Requestに回答しなくなるため、Tx Underrun/EOMが発生します。Tx Underrun/EOM状態になったときの送信ブロックの動作は、送信割り込みで送信データを転送する場合と全く同様です。Tx DMA RequestはTx Underrun/EOM状態になった後も発生し続けます (REQT端子="H"が保持されます)。したがって、DMAコントローラの再設定を行い、再起動させるときに、DMAコントローラがTx DMA Requestのエッジを必要とする場合は注意が必要です。そのような場合には、一度ソフトウェアでTx DMA Request Enable(WR2 D6)='0'にして、Tx DMA Requestのエッジを作り出す必要があります。

図11-6にREQT端子とTXBRDYビットの動作例を示します。図6-4はMark/Flag_On Idle='1'、完全Emptyモードに設定した場合の例です。

CRCの生成

Tx Underrun/EOM発生で、Tx Underrun/EOMビットが'0'→'1'に変化し、かつ、Abort/Flag On Underrun='0'の場合は、CRCコード2バイトが最終の送信データとClosing Flagの間に付加されます。CRC生成式は次の2通りです。

●CRC-16/CRC-SDLC_(WR4 D1)='0'のとき

$$X^{16}+X^{12}+X^5+1$$

●CRC-16/CRC-SDLC_(WR4 D1)='1'のとき

$$X^{16}+X^{15}+X^2+1$$

どちらの生成式の場合も、CRCコードの部分はビットを反転してMSBから出力されます。CRC生成回路のリセットはOpening Flag送出時に自動的に行われます。また、CRCのレジスタをAll '0'に初期化するかAll '1'に初期化するかはCRC Preset 1/0_(WR4 D2)で設定します。

Abort 送出

Tx Underrun/EOM発生で、Tx Underrun/EOMビットが'0'→'1'に変化し、かつ、Abort/Flag On Underrun='1'の場合は、Abort (連続する8個の'1')が最終の送信データとClosing Flagの間に送出されます。また、Send Abortコマンド(WR1)を発行することによって、任意のタイミングでAbortを送出することもできます。これは、いったん送信し始めたフレームを途中でキャンセルしたいような場合に用います。この場合、Tx FIFOに残っている送信データは全てクリアされます。

Tx FIFOのリセット

Tx FIFO Resetコマンド(WR1)、Send Abortコマンドを発行することにより、Tx FIFOはリセットされ、Tx FIFOに残っていた送信データはすべてクリアされます。いったん、Tx FIFOに書き込んだ送信データをキャンセルする場合にこれらのコマンドを発行します。ハードウェア・リセット、ソフトウェア・リセットでもTx FIFOにあるデータはクリアされます。

送信ディスエーブル

フレーム送信中にTx Disable状態にすると、現在送出中の送信データを送出し終わった後、マーク状態になります。このときTx FIFOに送信データが残っている場合は、そのままの状態になります。再びTx Enable状態にして送信を開始するときには、あらかじめTx FIFO ResetコマンドでTx FIFOに残った送信データをクリアしておく必要があります。

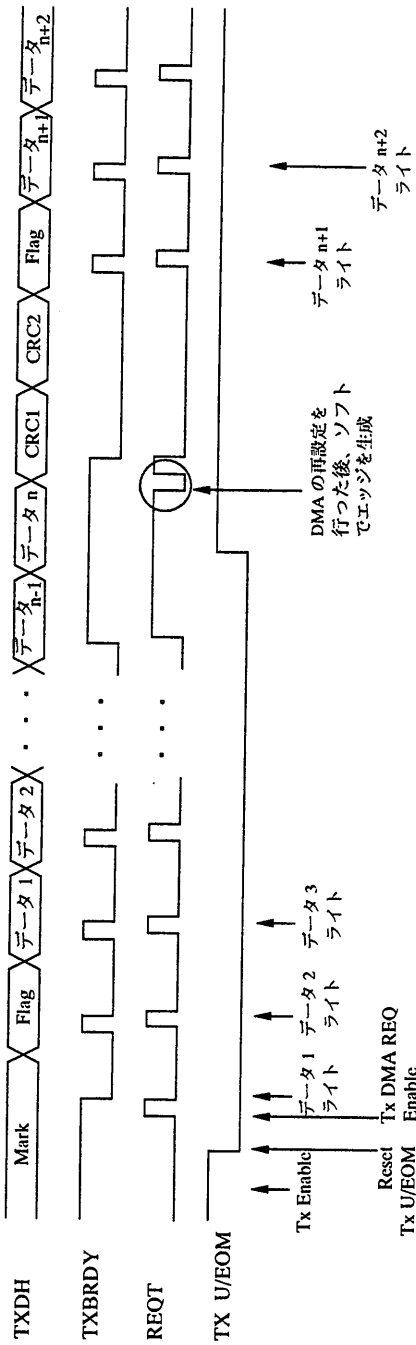


図 11-6 TXBRDY ビットと REQ の動き

HDLC フレームの受信

Flag 検出

初期設定後、Rx Enable 状態になると受信ブロックはFlag検出ステートに移り、Flagパターンの検出を開始します。Flagを検出するとHunt(RR1 D4)='0'となりFlag同期が確立します。フレーム受信中に、受信を中断して新たにFlag検出ステートに入りたい場合は、Enter Hunt コマンド(WR1)を発行します。また、フレーム受信中およびフレーム受信終了後にAbortを受信した場合もFlag検出ステートに入ります。いずれの場合もFlag検出ステートにいるときはHunt='1'となります。Rx Disable(WR1)コマンド発行によってもHunt='1'になりますが、Rx Enable状態にしない限り、Flag検出ステートには入りません。

受信アドレスの比較

Flag同期が確立されている状態でFlag以外のパターンが受信されると、受信ブロックはアドレス・フィールドと判断します。Address Compare Enable (WR4 D7)='0'のときは、受信アドレスとWR5の比較は行われず、無条件にフレームが受信されます。Address Compare Enable='1'のときは、Address Compare Mode Half/Full_(WR4 D6)の設定により、受信アドレスとWR5の全ビットあるいは上位4ビットが比較されます。受信アドレスとWR5が一致した場合には、そのフレームは受信されますが、一致しない場合は全く受信は行われません。また、アドレス・フィールドが'FFH'の場合はグローバル・アドレスとして扱われるため、Address Compare Enable='1'に設定した場合でも'FFH'で始まるフレームは無条件に受信されます。

割り込みによる受信データ転送

フレーム中でOpening FlagとClosing Flagで挟まれた部分は、フィールドの種類によらず、すべてデータとして受信されます。割り込みによって受信データを転送する場合は、割り込みモードをRx Int On All Rx Character And Sp Intモード(WR2 D3, D2='10')に設定します。このモードに設定すると、受信データがRx FIFOに入りRx FIFOが所定の状態になるたびに

Rx Intを発生します。Rx Int On Rx FIFO RCAモード(WR2 D5, D4='00')の場合はRx FIFOに少なくとも1つ受信データがあると、Rx Intを発生します。Rx Int On Rx FIFO Halfモード(WR2 D5, D4='01')の場合はRx FIFOに4バイト以上受信データがあるとRx Intを発生します。Rx Int On Rx FIFO Fullモード(WR2 D5, D4='10')の場合はRx FIFOがFull状態(8バイト受信データが入っている状態)のときにRx Intを発生します。RCAビット(RR1 D0)はRx FIFOに少なくとも1つ受信データがあり、読み出し可能であるときに'1'になります。したがって、Rx Int On Rx FIFO HalfモードあるいはRx Int On Rx FIFO Fullモードの場合は、Rx Intの発生とRCAビットの状態は一致しません。いずれのモードの場合もRx Intの受信ルーチンを抜け出す前に、もう一度RCAビットの状態をチェックして、読み出せる受信データはできるだけ読み出しておくことにより、Rx Intのオーバーヘッドを少なくすることができます。

割り込みモードをRx Int On All Rx Character And Sp Intに設定した場合、Special Condition (Rx Overrun ErrorおよびEnd Of Frame)が発生したときの動作は次のようになります。Rx Int On Rx FIFO RCAモードの場合はSpecial Conditionを含む受信データがRx FIFOの出口に表れたときにSp Intが発生します。Sp Intの割り込みルーチン中では受信データを読み出す前にRR2を読み出して、Sp Intの原因が何であったかを判断する必要があります。受信データを先に読み出すとRR2が消失し、Sp Intの原因が何であったかが分からなくなるため注意が必要です。Rx Int On Rx FIFO HalfモードおよびRx Int On Rx FIFO Fullモードの場合はSpecial Conditionを含む受信データがRx FIFOに入ると直ちにSp Intが発生します。Sp Intの割り込みルーチンでは、どの受信データにSpecial Conditionが含まれていたかを判断するために、RR2を読み出してから受信データを読み出す必要があります。

Rx Intモード設定(WR2 D5, D4)およびRx Int, Sp Int制御(WR2 D3, D2)とRx Int, Sp Intの関係は表8-1にまとめてありますので、適宜参照して下さい。

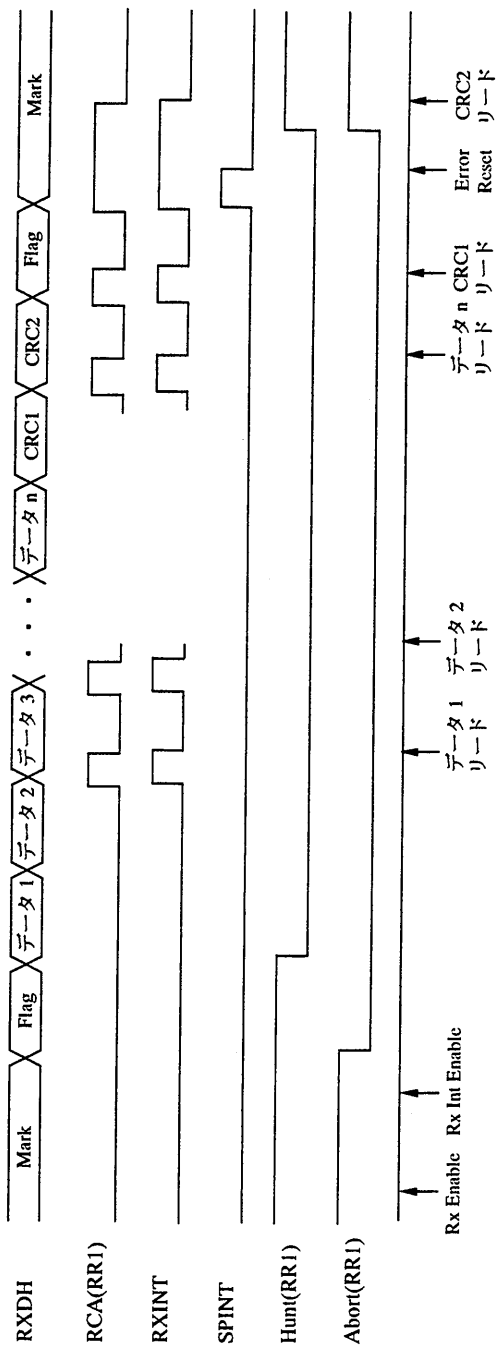


図 11-7 割り込みによって受信データ転送を行う場合のタイミング
(Fx Int On Fx FIFO RCA モード、Hunt IE=Abort IE='0' に設定した場合)

図 11-7 に割り込みによって受信データの転送を行う場合のタイミングの概略を示します。Rx Int On Rx FIFO RCA、Hunt IE=Abort IE='0'に設定し、フレーム受信中にRx Overrun Errorが発生しなかった場合の各信号の動きです。

RR2のEOFビット、Rx Overrun ErrorビットはError Reset コマンドでリセットします。EOF ビットは次のフレームの受信によってもリセットされますが、Rx Overrun ErrorビットはError Resetコマンドを発行しない限りリセットされません。また、Error Resetの発行によって、Sp Int もリセットされます。Overrun Error、CRC Errorが発生したときは受信データ領域にステータス情報が書き込まれるために、本来のデータは正しく受信できません。各エラーが発生したときに受信されるデータは以下のようになります。

表11-2 各エラービットと受信データ

EOF	CRCE	OVRE	受信データ(RR0)
0	0	0	本来の受信データ
1	0	0	本来の受信データ
0	0	1	20H
1	0	1	A0H
1	1	0	COH
1	1	1	E0H

RR2のResidueビットが'1'のときは、端数ビットに関する情報が図 11-3 のように受信データ領域に書き込まれます。端数ビットが何ビットであったかは、この情報をもとにソフトウェアでデコードする必要があります。また、フレームの最後の受信データでOverrun Error、CRC Errorが発生したときは、端数フレームを受信した場合でもResidue='1'にはなりません。この場合、端数ビットがなかった場合と同様に、受信データは正しく受信できません。

DMA による受信データ転送

DMA で受信データを転送する場合は、Rx Int On First Rx Character And Sp IntモードあるいはSp Int Onlyモードに設定します。

Rx Int On First Rx Character And Sp Intモードは、フレームの最初の1バイトは割り込みで、それ以降の

受信データは DMA で転送する場合に用います。Rx Int モード設定は Rx Int On Rx FIFO RCA に設定します。通常は、最初の1バイトの転送ルーチンの中で Rx DMA Request Enable(WR2 D7)='1' にして、DMA コントローラを起動します。Rx DMA Request の発生は RCA ビット (RR1 D0) に従います。すなわち、Rx FIFO に読み出し可能な受信データが存在する場合に Rx DMA Request が発生します。フレームの受信が終了したときにはRx DMA Request Enable='0' とするとともにEnable Int On Next Rx Characterコマンドを発行して、次のフレーム受信に備えます。

Sp Int Only モードは全ての受信データをDMAで転送する場合に用います。この場合もRx Intモード設定はRx Int On Rx FIFO RCAに設定して下さい。

これらのモードに設定した場合、Sp Int は Special Condition を含む受信データを読み出さない限り発生しません。そして、Special Condition を含む受信データを読み出すと Rx FIFO はロックされた状態となり、Rx FIFO の出口には Special Condition を含む受信データとこれに対応する受信ステータス RR2 が保持されます。これにより、DMA で受信データを読み出した後に、Sp Int の割り込みルーチン中でその受信データに対応する受信ステータスを読み出すことが可能になります。また、Rx FIFO がロックされているときは、Rx DMA Request は発生しません。Rx FIFO のロックはError Resetコマンドを発行することにより解除されます。このとき、Rx FIFOに読み出し可能な受信データがあれば、再びRx DMA Requestが発生します。

図6-6にDMAによって受信データ転送を行う場合のタイミングの概略を示します。Sp Int Onlyモードに設定し、フレーム受信中にRx Overrun Errorが発生しない場合の例です。

Overrun Error、CRC Error、端数ビット発生時の受信データは、割り込みで受信データを転送した場合と同様に表11-2および図11-3のようになります。

CRC の検査

受信ブロックには CRC 検査回路が含まれており、

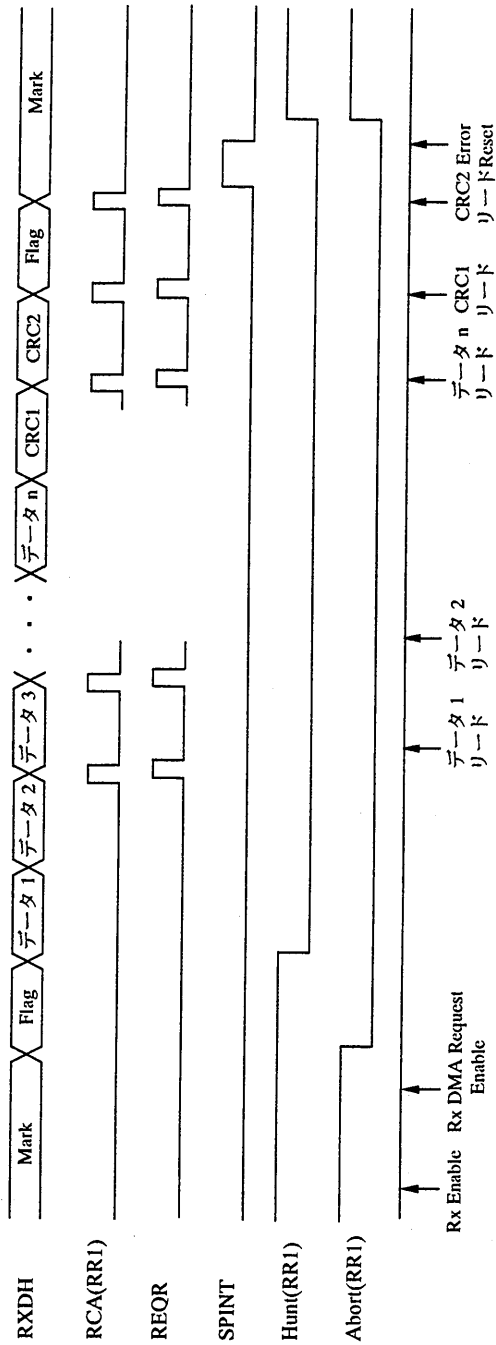


図 11-8 DMA によって受信データ転送を行う場合のタイミング
(Hunt IE=Abort IE='0' の場合)

受信データに誤りがないかどうかを検査します。CRC生成式、およびCRCレジスタの初期値については、送信ブロックのCRC生成回路と同様にCRC-16/CRC-SDLC_ビット(WR4 D1)およびCRC Preset 1/0_ビット(WR4 D2)に従います。また、CRCコードを含まないフレームの受信のためにRx CRC Check Disableビット(WR4 D3)を設けてあります。このビットを‘1’にすると受信フレームのCRC検査は全く行いません。

Abort 受信

Rx Enable状態で7個以上の連続した‘1’を受信すると、受信ブロックはAbort受信と判断します。フレーム受信中にAbortを受信した場合はただちに受信を中止し、受信ブロックはFlag検出ステートに移ります。このときRx FIFOにある受信データはそのままRx FIFOに残るため、次のフレームの受信を始める前に、Rx FIFO Resetコマンド(WR1)でRx FIFOをクリアしておく必要があります。Abortを受信するとAbort(RR1 D7)='1'になりますが、RXDHが‘0’になりAbort状態が消滅するとAbort='0'になります(これはAbort IE='0'の場合で、Abort IE='1'のときはAbort='1'がラッチされます)。また、Abort='1'となるのとほとんど同時にHunt='1'にもなりますが、HuntビットはAbort状態が消失しても次にFlagを検出するまでは‘1’のままです。

リセット後Rx Enableコマンド(WR1)を発行するまでは、外部の受信データに関わらず内部的に受信ラインを‘1’にマスクするため、Abort受信状態になります。したがって、この状態でAbort IE='1'、E/S Int Enable='1'に設定するとAbortによるE/S Intが発生します。これが問題となる場合は、Rx Enableコマンドを発行した後、E/S Int Resetコマンド(WR1)を1回発行してからE/S Int Enableにして下さい。

受信ディスエーブル

フレーム受信中にRx Disable状態になると、ただちに受信を中止します。このときHunt='1'になるため、次にRx Enable状態にしたときはFlag検出ステートに入ります。Rx Disable状態になったときにRx FIFOに受信データがある場合はそのままRx FIFOに残ったままになりますので、次のフレームの受信を始める前にはRx FIFO ResetコマンドでRx FIFOをクリアしておく必要があります。

11.7 ループモード

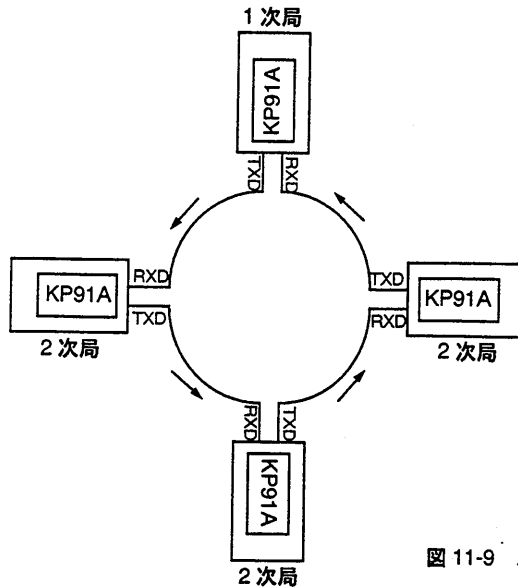


図 11-9 ループモードの概念図

ループモードの概要

ループモードは1次局と複数の2次局の送受信ラインをループ状に接続し、1次局の制御のもとに2次局が送受信を行なうモードです。2次局は1次局からEOP(End Of Poll)パターンと呼ばれる特定のパターンが送られて来るまで送信動作を行うことが出来ません。EOPパターンは‘0’の後に‘1’が7個連続した‘01111111’というパターンです。1次局が“Mark Idle”に設定されていれば、EOPパターンは1次局がフレームを送出し終わったときに自動的に送出されます。1次局からのEOPパターンを検出するまで、2次局は受信したシリアルデータに1ビット時間の遅延を挿入し、そのままTXDから送出します。2次局はEOPパターンを検出したときに、送信すべきデータがあれば、EOPパターンの最後の‘1’を‘0’に変えてFlagを生成し、その後送信すべきフレームを送出します。さらに下流の2次局は上流の2次局が送出したフレームに自分が送出するフレームを付加することができます。

このようにして、1次局が送出したフレームには2次局が送出したフレームが付加され、最終的には再び1次局がこれを受信します。1次局は2次局から返ってきた応答の内容を確認し、次に行うべき動作を決定

します。以上がループモードの簡単な説明です。

次にKP91Aをループモードで用いる場合の具体的な動作の説明を行います。

ゲート遅延状態

KP91Aをループモードの2次局に設定するには、Loop Mode Enable(WR3 D0)='1'にします。Loop Mode Enable='1'の設定は、Rx Enable状態にした後に行ってください。Rx Disable状態でLoop Mode Enable='1'に設定すると正しくループモードに移行できない可能性がありますので御注意下さい。Loop Mode Enable='1'に設定すると、TXDHとRXDHはゲート遅延のみで直結され、RXDHから入ってきた受信データは、そのままTXDHから出力されます。ゲート遅延状態にいるKP91Aは送信も受信も行なうことは出来ません。なお、KP91Aをループモードの1次局として用いる場合は、EOPパターンが正しく送出されるようにMark/Flag_On Idle='1'に設定する必要がある以外は、特別な設定は必要ありません。

1ビット遅延状態

ゲート遅延状態で GAOP(WR3 D1)=‘1’ に設定すると、KP91AはEOPパターンを検出したとき1ビット遅延状態に移行します。1ビット遅延状態では、RXDHで受信した受信データを1ビット時間遅延させてTXDHから送出します。1ビット遅延状態では1次局が送出したデータが受信可能です。1ビット遅延状態に移行するまではTx Disable状態にしておく必要があります(RXDから1ビット遅延でTXDに中継されるシリアルデータは、Tx Disable状態でも送出されます)。ゲート遅延状態から1ビット遅延状態に移行するタイミングでTx Enable状態であった場合の動作は保証できませんので御注意下さい。

GAOP=‘0’のときはEOPパターンを検出しても、1ビット遅延状態には移行しません。

Loop Sending 状態

1ビット遅延状態で GAOP=‘1’ に設定すると、KP91AはEOPパターンを検出したときにEOPパターンの最後の‘0’を‘1’に変えてOpening Flagを生成して、Loop Sending状態に移行します。Loop Sending状態においてのみ、2次局に設定されたKP91Aはデータの送信を行うことができます。Loop Sending状態になったにもかかわらず、送信データがTx FIFOにない場合はOpening Flagを送出し続けます。2次局での遅延を出来るだけ少なくするためには、2次局はすみやかに送信データを書き込む必要があります。フレーム送出の終りでClosing Flagを送出するときにGAOP=‘0’であれば、2次局はLoop Sending状態から抜けます。このタイミングでGAOP=‘1’であれば、Loop Sending状態からは抜けずに次のフレームを送出することが出来ます。

Loop Sending中にGAOP=‘0’にするタイミングには注意が必要です。KP91Aの内部でフレームの送信が開始された後であれば、いつGAOP=‘0’にしても構いません。ただし、内部的にフレームの送信状態になったことを知る手段はありません(Loop Sending=‘1’になるタイミングと内部的な送信開始のタイミングは異なります)。したがって、最初の送信データを書き込んだ後、適当の時間において内部的に送信状態になったのを見計らってからGAOP=‘0’にするか、もしくはTx Underrun/EOMのE/S Int割り込みルーチン中でGAOP=

‘0’にするようにしてください。

1ビット遅延状態でEOPパターンを検出したときに、GAOP=‘0’ならばLoop Sending状態には移行しません。

ループからの離脱

ループモードから離脱するには、1ビット遅延状態でLoop Mode Enable=‘0’としてください。このとき Hunt 状態であれば、2次局はただちにループモードから抜けて通常モードに戻ります。Hunt 状態でなければ Hunt状態になるのを待って通常モードに戻ります。

Loop Sending 状態から通常モードに戻ることは出来ません。必ず、1ビット遅延状態で Loop Mode Enable=‘0’としてください。Loop Sending 状態で Loop Mode Enable=‘0’とした場合の動作は保証致しません

On Loop ビット、Loop Sending ビット

On Loop ビット(RR9 D0)は1ビット遅延状態および Loop Sending 状態のときに‘1’になります。Loop Sending ビットは(RR9 D1)はLoop Sending状態のときに‘0’になります。

図11-10にループモードでの基本的な動作のタイミングチャートを示します。

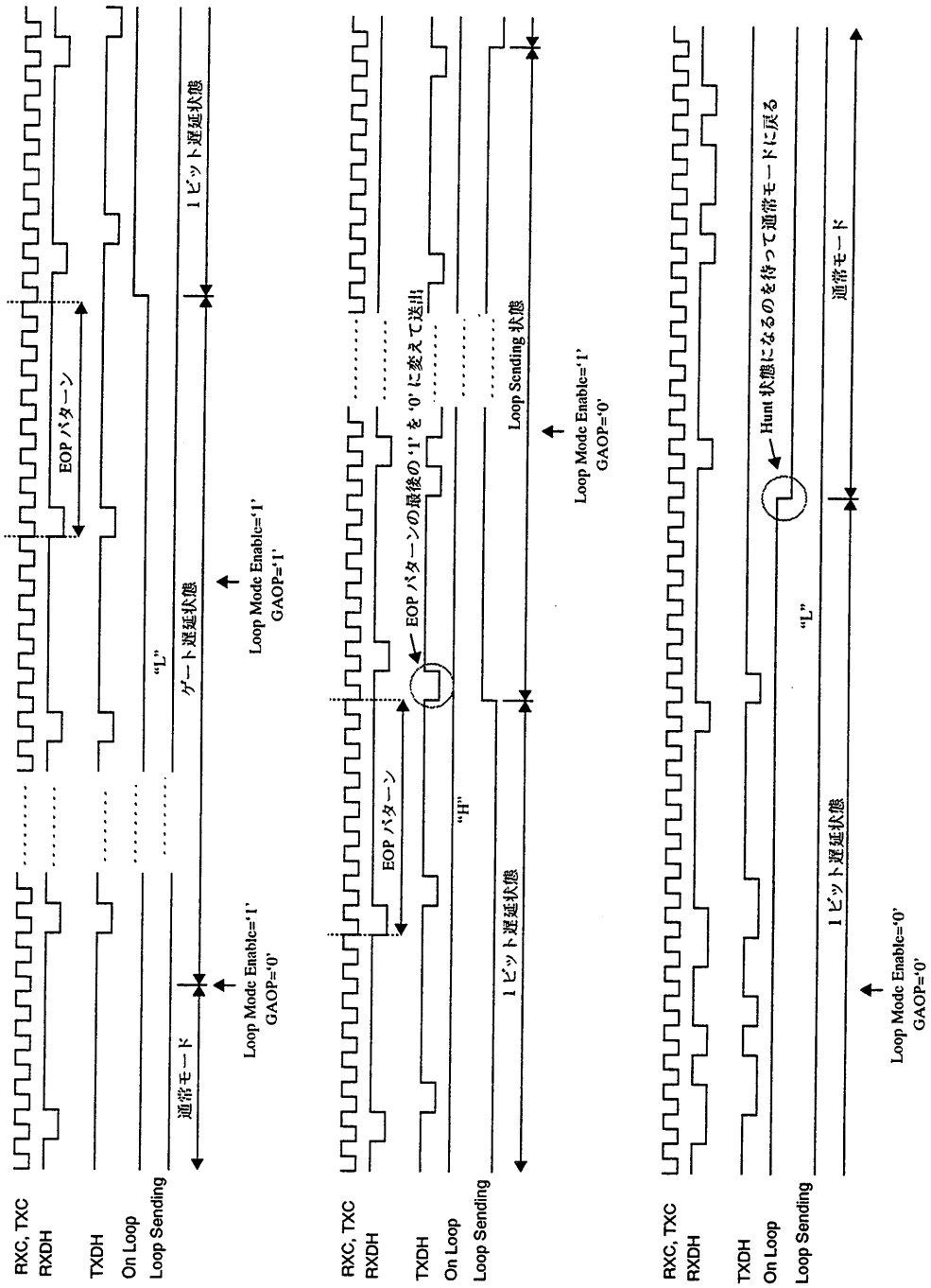


図 11-10 ループモードの基本動作

11.8 割り込み

KP91A は受信割り込み、Special Condition 割り込み、送信割り込み、External/Status 割り込みの4種類の割り込みを発生します。4種類の割り込みの優先順位制御、割り込みベクタの出力等は、KP69の割り込みコントローラで行います。4種類の割り込みの優先順位は、アプリケーションごとに設定することが可能ですが、Special Condition 割り込み>受信割り込み>送信割り込み>E/S割り込み、という順位に設定するのが最も一般的です。

受信割り込み

受信割り込み (Rx Int) は、受信データの読み出しを CPU に要求する割り込みです。Rx Int は「Rx Int, Sp Intモード設定」(WR2 D3,D2)を

“Rx Int On First Rx Character And Sp Int”

“Rx Int On All Rx Character And Sp Int”

に設定した場合に発生します。“Rx Int On First Rx Character And Sp Int”はフレームの最初の受信データのみ割り込みで転送し、残りはDMAを用いて転送する場合に用います。“Rx Int On All Rx Character And Sp Int”は全ての受信データを割り込みで転送する場合に用います。また、Rx FIFOの状態がどのようになったときにRx Intを発生するかは、Rx Intモード設定(WR2 D5,D4)に従い、以下の3通りがあります。

“Rx Int On Rx FIFO RCA”

“Rx Int On Rx FIFO Half”

“Rx Int On Rx FIFO Full”

“Rx Int On Rx FIFO RCA”はRx Character Available状態すなわちRx FIFOにデータが1つでも存在する場合にRx Intを発生します。“Rx Int On Rx FIFO Half”はRx FIFOに半分(4段)まで受信データが入った場合にRx Intを発生します。“Rx Int On Rx FIFO Full”はRx FIFOにFull状態(8段)まで受信データが入った場合にRx Intを発生します。また、Rx Intモードの設定はRx DMA Requestの発生にも影響を与えるため、注意が必要です。DMAで受信データの転送を行なう場合は“Rx Int On Rx FIFO RCA”に設定して下さい。

Rx Intは受信データを読み出すとリセットされます。そして再びRx FIFOが所定の状態になると、Rx

Intが発生します。

Special Condition 割り込み

Special Condition割り込み(Sp Int)は受信ブロックでSpecial Conditionが生じたときに発生します。Special ConditionとはRx Overrun ErrorとEnd Of Frameです。Sp Intルーチンの中では、Sp Intの発生要因によって所定の処理を行ないます。Sp IntはError Resetコマンドを発行するとリセットされます。

Sp Intが発生するタイミングは「Rx Int, Sp Int制御」(WR2 D3,D2)および「Rx Intモード設定」(WR2 D5, D4)によって異なります。

Rx Int, Sp Int制御を“Rx Int On All Rx Character And Sp Int”に設定した場合は、Rx Intモード設定を“Rx Int On Rx FIFO RCA”に設定した時と、“Rx Int On Rx FIFO Half”および“Rx Int On Rx FIFO Full”に設定した時で動作が分かれます。“Rx Int On Rx FIFO RCA”に設定した時は、Special Conditionを含むデータがRx FIFOの出口に現れるとただちにSp Intを発生します。一方、“Rx Int On Rx FIFO Half”および“Rx Int On Rx FIFO Full”に設定した時は、Special Conditionを含むデータが受信シフトレジスタからRx FIFOに転送されるとただちにSp Intを発生します。Sp Intルーチンでは、Rx FIFOに存在するどの受信データにSpecial Conditionが含まれるかを、RR0(受信データ)を読み出す前にRR2を読み出して判断する必要があります。

“Rx Int On First Rx Character And Sp Int”あるいは“Sp Int Only”モードに設定すると、Special Conditionを含む受信データがRx FIFOから読み出されてはじめてSp Intが発生します。この場合、Sp Intが発生するとRx FIFOはロックされた状態になり、RR0, RR2は何度読み出しても同じ値が読み出されます。Rx FIFOのロックはError Resetコマンドを発行すると解除されます。また、“Rx Int, Sp Int Disable”にするといかなる場合でもSp Intは発生しません。

「Rx Int, Sp Int制御」(WR2 D3,D2)および「Rx Intモード設定」(WR2 D5, D4)の設定とRx Int, Sp Intの動きを表11-3にまとめます。

表11-3 WR2の設定とRx Int、Sp Intの関係

Sp Int, Rx Int制御 (WR2 D3,D2)	Rx Int モード (WR2 D5,D4)	Rx Int	Sp Int
0 0	X X	発生しない	発生しない。
0 1	0 0	Reset後およびEnable Int On Next Rx Characterコマンド発行後のみ、受信データがRx FIFOに存在すると発生	Specail Conditionを含む受信データを読み出すと発生。 (Rx FIFOはロックされる)
1 0	0 0	Rx FIFOに受信データが存在すると発生	Special Conditionを含む受信データがRx FIFOの出口に現れると発生
	0 1	Rx FIFOに受信データが4 Byte以上存在すると発生	Special Conditionを含む受信データがRx FIFOに入ると発生
	1 X	Rx FIFOに受信データが8 Byte存在すると発生	Special Conditionを含む受信データがRx FIFOに入ると発生
1 1	0 0	発生しない	Specail Conditionを含む受信データを読み出すと発生 (Rx FIFOはロックされる)

送信割り込み

送信割り込み(Tx Int)はCPUに送信データの書き込みを要求する割り込みです。Tx Int Enable(WR2 D0)= '1' にすることで送信割り込みはイネーブル状態になります。Tx Int Mode(WR2 D1)='0'の場合はTx FIFOが完全にEmptyになったときにTx Intを発生します。Tx Int Mode='1'の場合はTx FIFOの入口がEmptyになった場合にTx Intを発生します。Tx Intは送信データを書き込むか、Tx Int Resetコマンド(WR1)を発行するとリセットされます。Tx Int Reset コマンドはフレーム送信の最後で、送信を終了する場合に用います。ハードウェア・リセットあるいはソフトウェア・リセット後は、最初の送信データを書き込むまではTx Intは発生しません。

External/Status 割り込み

E/S Intは次の割り込み要因のうち、WR10でイネーブルにした割り込み要因の論理和で出力されます。以下のビットはすべてRR1に存在するビットです。

Abort	'0'→'1', '1'→'0'
Tx U/EOM	'0'→'1'
CTS	'0'→'1', '1'→'0'
Hunt	'0'→'1', '1'→'0'
DCD	'0'→'1', '1'→'0'
Zero Count	'0'→'1'

2つ以上の割り込み要因をイネーブルにする場合は、E/S Intが発生したときにどの要因で発生したかを判断する必要があります。このためにはRR1のコピーをメモリやCPU内部のレジスタに保存しておき、E/S Intが発生するたびに現在の内容と比較します。変化のあったビットがE/S Intの発生要因になります。E/S Intが発生するとイネーブルされている割り込み要因がラッチされます。ラッチされた状態はRR1を通して読み出すことが出来ます。E/S Intおよびラッチされた状態の解除は、Reset E/S Intコマンド(WR1)で行います。ラッチを解除した時点で、各ビットの状態がラッチした状態と反転している場合は再びE/S Intが発生します。一般に各ビットがラッチされている間は、各ビットの変化は記憶されず、ラッチを解除した時点でビットの状態がもとに戻っていれば、さら

なる E/S Int は発生しません。ただし、Abort のみは例外でラッチされている間に発生した変化を記憶します。これらの動作の例を図11-11に示します。

ケース1は次のようなシーケンスを説明しています。

- (1) Hunt '1' → '0' による E/S Int 発生
- (2) Hunt '0' → '1'
- (3) E/S Int Reset コマンド発行
- (4) Hunt '0' → '1' による E/S Int 発生
- (5) E/S Int Reset コマンド発行

ケース2は次のようなシーケンスを説明しています。

- (1) Tx Underrun/EOM '0' → '1' による E/S Int 発生
- (2) Abort '0' → '1' → '0'
- (3) E/S Int Reset コマンド発行
- (4) Abort '0' → '1' による E/S Int 発生
- (5) E/S Int Reset コマンド発行
- (6) Abort '1' → '0' による E/S Int 発生
- (7) E/S Int Reset コマンド発行

ZC ビットの '0' → '1' の変化は E/S Int 発生の要因になりますが、他の要因のようにラッチされません。したがって、このビットを読み出してもほとんどの場合は '0' になります。

E/S Int Enable = '0' のときは各要因をイネーブルにしても、E/S Int は発生しません。ただし、割り込みイネーブルにしたビットのラッチ機能は有効となりますので、RR1 の読み出しの際には注意が必要です。ハードウェア・リセット、ソフトウェア・リセット後には各要因はイネーブル状態になります。

なお、リセット後 Rx Enable コマンド (WR1) を発行するまでは、外部の受信データに関わらず内部的に受信ラインを '1' にマスクするため、Abort 受信状態になります。したがって、この状態で Abort IE = '1'、E/S Int Enable = '1' に設定すると Abort による E/S Int が発生します。これが問題となる場合は、Rx Enable コマンドを発行した後、E/S Int Reset コマンド (WR1) を 1 回発行してから E/S Int Enable にして下さい。

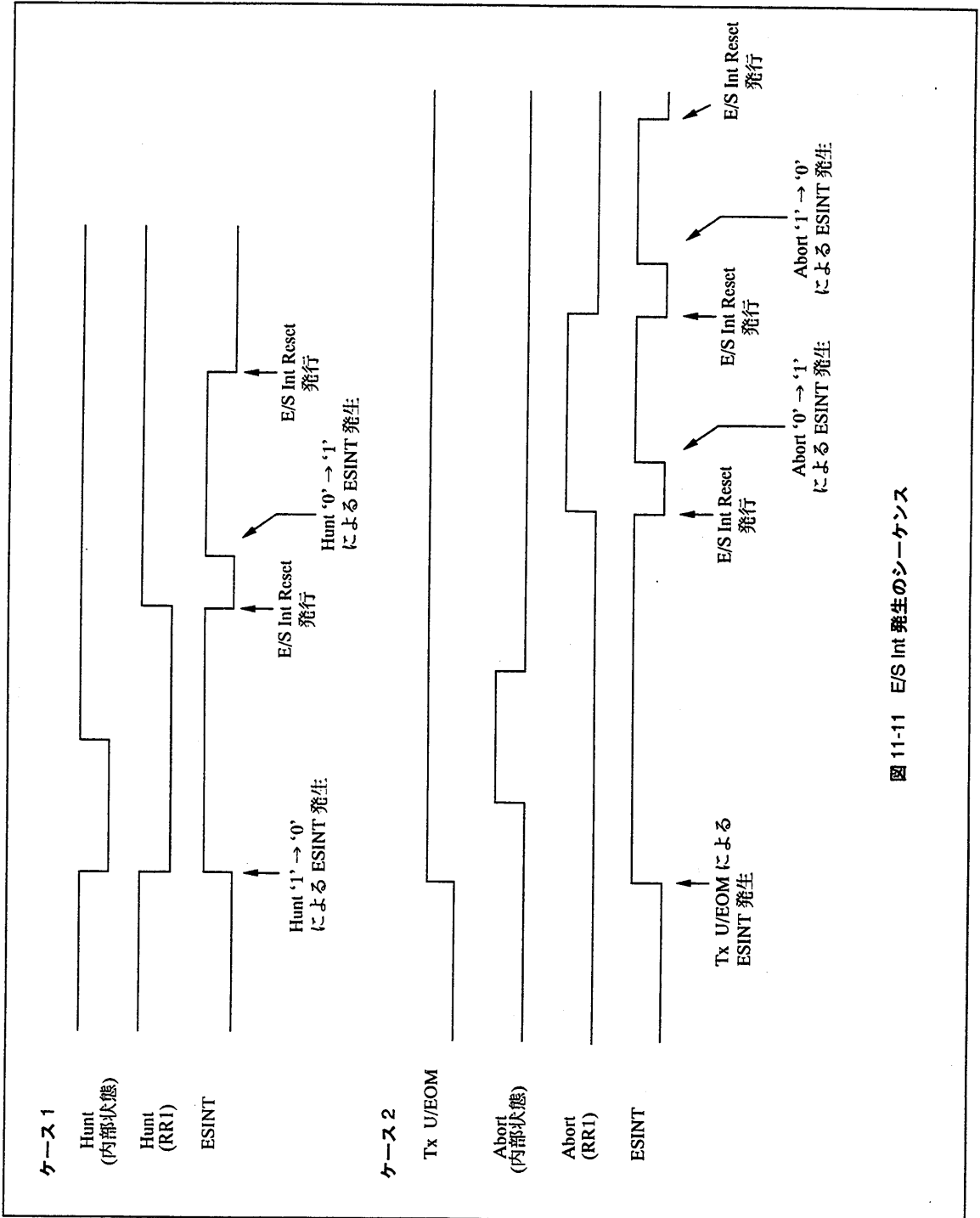


図 11-11 E/S Int 発生シーケンス

11.9 ボーレートジェネレータ

KP91A は送受信クロック生成用にボーレートジェネレータ (BRG) を搭載しています。BRG カウンタのソース・クロックは BRG Source CLKH/RTXCH_ ビット (WR9 D1) で基本動作クロック CLKH と外部入力 RTXCH が選択できます。ここで、基本動作クロック CLKH にはシステムクロックの2倍の周波数の信号が内部で供給されています。BRG の Time Constant は WR7, WR8 に設定します。Time Constant と BRG 出力の周波数 (Baud Rate) の関係は次のようになります。

(BRG 出力の周波数)

$$= (\text{ソース・クロックの周波数}) \\ \div [2 \times (\text{Time Constant} + 2)]$$

すなわち、

(Time Constant)

$$= (\text{ソース・クロックの周波数}) \\ \div [2 \times (\text{BRG出力の周波数})] - 2$$

となります。Time Constant = '0000H' に設定した場合、システムクロック 10MHz 動作時には、ソース・クロックとして基本動作クロック CLKH (20MHz) を選択すると、BRG 出力は4分周されて最大転送速度の 5MHz になります。

Time Constant 設定用レジスタ WR7, WR8 はハードウェア・リセット、ソフトウェア・リセットによって初期化されないため、BRG Enable (WR9 D0) = '1' にするまえに必ず初期化しておく必要があります。

Time Constant は BRG 動作中でも書き換え可能ですが、書き換えた値が反映されるのは次に Zero Count 状態となって Time Constant が BRG カウンタにロードされるときです。

BRG の Zero Count で E/S Int を発生させることができます。Zero CountIE (WR10 D1) = '1' とすると ZC ビット (RR1 D1) の '0' → '1' の変化で E/S Int を発生します。ただし、このビットは他の E/S Int の要因となるビットとは異なり、Zero Count IE = '1' とした場合でもラッチされません。E/S Int 発生前の RR1 と発生

後の RR1 を比較して内容が同じ場合は、Zero Count による割り込みとなります。

11.10 DPLL

KP91A は受信データから受信クロックを再生するためのDPLL(Digital PhaseLocked Loop)を搭載しています。一般に HDLC のような同期通信ではシリアル通信ラインでシリアル・データを送受信するとともに、それらと位相が合った送受信クロックを送るラインが必要となります。ところが KP91A に搭載されたDPLLを用いると、受信側で受信データから受信クロックを自動的に再生できるため、送受信クロック用のラインが不要になります。

DPLLは受信データの変化タイミングを常に監視しており、最適なタイミングで受信データをサンプルできるように受信クロックのエッジを調節します。したがって、受信データが長時間変化しない（‘0’が連続する）可能性がある NRZ モードでは基本的に用いることはできません。しかし、アプリケーションの特性によって‘0’が長時間連続することがない場合はNRZモードでもDPLLを用いることができます。

DPLL の動作の開始

初期設定後Enter Search Mode コマンド(WR9)を発行すると DPLL は RXDH のエッジの検出を開始します。RXDHの変化を検出するまではDPLLカウンタは停止状態で、DPLL Output (送受信クロック)は変化しません。この状態でのDPLL Outputの状態はDPLLのモードによって異なります。NRZI モードのときには“L”、FMモードのときには“H”になります。RXDHのエッジを検出するとDPLLカウンタは動作を開始します。その後の動作はモードによって異なります。

DPLL Disable コマンド(WR9)を発行するとDPLLカウンタは動作を停止します。DPLL Outputはモードに従った初期状態になります。

DPLL ソース・クロック

DPLL カウンタを動作させるソース・クロックはRTXCH端子からの外部入力、BRG出力、基本動作クロックCLKHの3つの中から選択出来ます。DPLLの動作モード(WR9 D7,D6)を設定する前に選択しておく必要があります。

DPLL 動作モード

DPLL FM/NRZI_ビット(WR9 D6)でDPLLをFMモードで動作させるか、NRZI モードで動作させるかを選択します。

DPLLをNRZIモードに設定した場合は、DPLL NRZI Half/Full_ビット(WR9 D7)によって、DPLLのソースクロックとしてビットレートの32倍の周波数のクロックを用いるか、ビットレートの16倍の周波数のクロックを用いるかを選択できます。DPLL NRZI Half/Full_ =‘0’に設定した場合は、ビットレートの32倍の周波数のクロックを DPLL ソースクロックとして用います。DPLL NRZI Half/Full_ =‘1’に設定した場合、ビットレートの16倍の周波数のクロックを DPLL ソースクロックとして用います。

DPLLをFMモードに設定した場合は、DPLL ソースクロックとして、ビットレートの16倍の周波数のクロックを用います。

NRZI モード

NRZIモードでDPLLを動作させる場合は、DPLLのソース・クロックとしてシリアル・データのビット・レートの32倍もしくは16倍の周波数のクロックを入力します。

DPLL NRZI Half/Full_ =‘0’に設定し、32倍クロックでDPLLを動作させる場合のDPLLカウンタの出力、受信シリアル・データのタイミング、受信クロックのタイミングの関係は図11-12のようになります。

カウンタは16を初期値としてカウント動作を行い、シリアル・ビットの領域を図11-12のように分割します。以後の説明では16→31、0→15のサイクルを1カウント・サイクルとします。

15の後半から16の前半までにRXDの変化があった場合はカウント値の調整は行いません。

16の後半から31までにRXDの変化があった場合は、そのカウント・サイクル中のカウント値4を2回カウントします。これによりカウント・サイクルを1

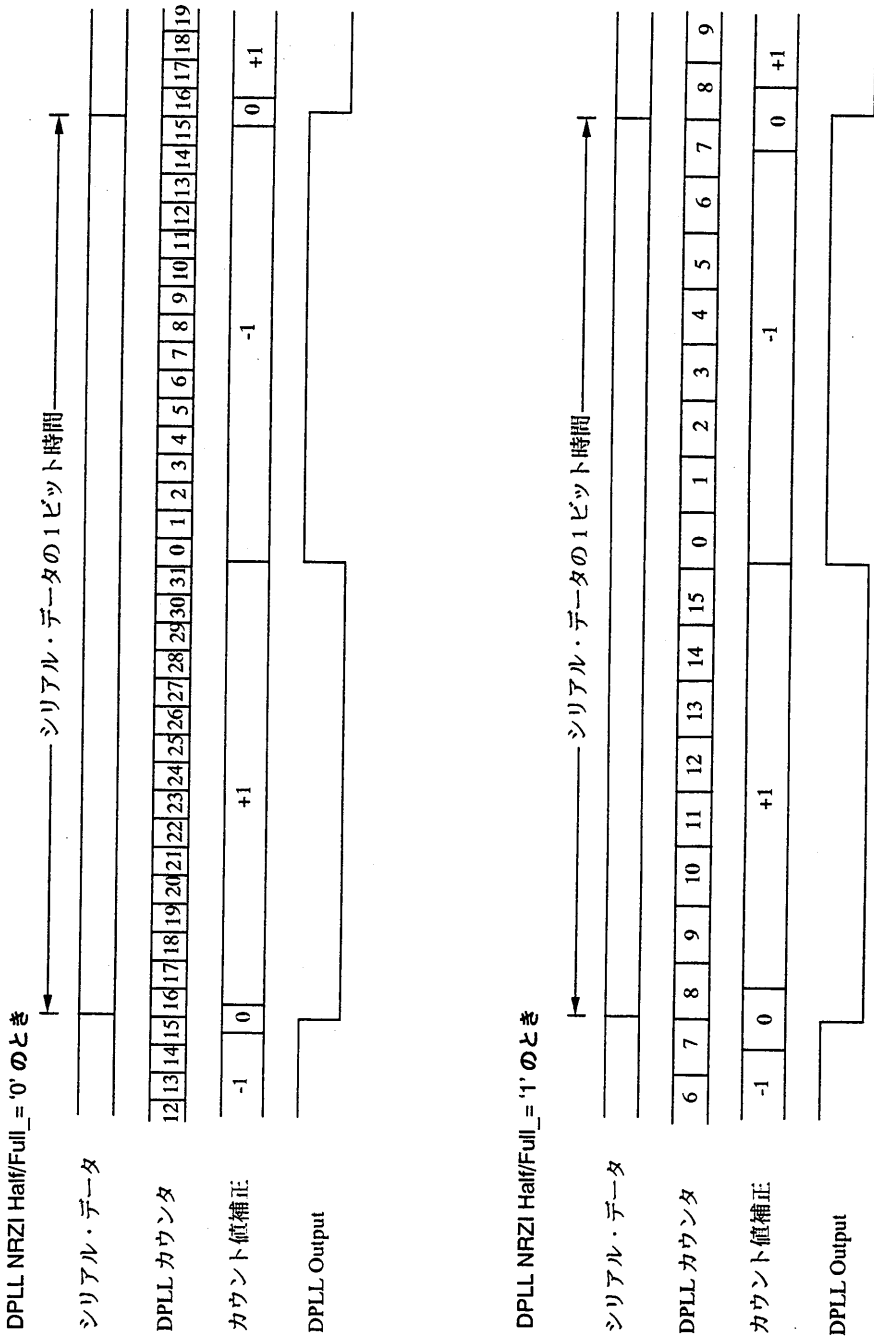


図 11-12 DPLL を NRZI モードで動作させた場合

カウント延長して、RXDのエッジが想定していたタイミングより後ろにあったことを補正します。

0から15の前半までにRXDの変化があった場合は、次のカウント・サイクルのカウント値5をカウントしません。すなわち3、4、6、7、…とカウントします。これによりカウント・サイクルを1カウント短くして、RXDHのエッジが想定したタイミングよりも前にあったことを補正します。

カウント・サイクル中にRXDHの変化がなかった場合は、カウント値の調整は行いません。

DPLL NRZI Half/Full_='1'に設定し、16倍クロックでDPLLを動作させる場合のDPLLカウンタの出力、受信シリアル・データのタイミング、受信クロックのタイミングの関係は図11-12のようになります。

カウンタは8を初期値としてカウント動作を行い、シリアル・ビットの領域を図11-12のように分割します。以後の説明では8→15、0→7のサイクルを1カウント・サイクルとします。

7の後半から8の前半までにRXDHの変化があった場合はカウント値の調整は行いません。

8の後半から15までにRXDHの変化があった場合は、そのカウント・サイクル中のカウント値4を2回カウントします。

0から7の前半までにRXDHの変化があった場合は、次のカウント・サイクルのカウント値5をカウントしません。

カウント・サイクル中にRXDHの変化がなかった場合は、DPLL Half/Full_='0'に設定した場合と同様、カウント値の調整は行いません。

NRZIモードでは受信クロックは常にカウント0になるときに立ち上がります。また、One Clock Missingビット(RR9 D7)、Two Clock Missingビット(RR9 D6)は常に'0'になります。

FMモード

FMモードでDPLLを動作させる場合は、DPLLのソース・クロックとしてシリアル・データのビット・レートの16倍の周波数のクロックを入力します。

DPLLをFMモードで動作させる場合のDPLLカウンタの出力、受信シリアル・データのタイミング、受信クロックのタイミングの関係は図11-13のようになります。DPLLカウンタのカウント・サイクルはNRZIモードと同様16→31、0→15で1カウント・サイクルとなります。すなわち、1カウント・サイクルはシリアル・データの2ビット期間に相当します。DPLLで生成した受信クロックと送信クロックは位相が1/4ずれます。この送受信クロックの位相関係は、FMエンコーディング・モードを用いた場合の理想的な位相関係になっています。DPLLは2ビットごとにRXDHのエッジのタイミングをチェックします。RXDHエッジの位置によってDPLLカウンタのカウント・サイクルを補正し、受信クロックのタイミングを調整します。

15の後半から16の前半までにRXDHの変化があった場合はカウント値の調整は行いません。

16の後半から19の前半までにRXDHの変化があった場合は、そのカウント・サイクル中のカウント値4を2回カウントします。これによりカウント・サイクルを1カウント延長して、RXDHのエッジが想定していたタイミングより後ろにあったことを補正します。

12の後半から15の前半までにRXDHの変化があった場合は、次のカウント・サイクルのカウント値5をカウントしません。すなわち3、4、6、7、…とカウントします。これによりカウント・サイクルを1カウント短くして、RXDHのエッジが想定したタイミングよりも前にあったことを補正します。

19の後半から12の前半までに発生したRXDHのエッジは、DPLLカウンタの動作に影響を与えません。この期間はFMエンコーディング・モードの特性によって必然的にRXDHのエッジが発生します。

12の後半から19の前半までにRXDHの変化がなかった場合は、DPLLはRXDHのエッジを見失ったものと判断し、One Clock Missingビット(RR9 D7)を‘1’にします。DPLL自体はそのまま通常と同じように動作し続けます。2回連続でRXDHのエッジを見失った場合は、Two Clock Missingビット(RR9 D6)を‘1’にします。このときDPLLは自動的に動作を停止し、Search Modeに入ります。そして、最初に Enter Search Mode コマンドを発行した後と同様に、新たなRXDHのエッジの検出を開始します。

One Clock Missingビット、Two Clock Missingビットはカウント値0のときにセットされます。また、これらのビットは一度セットされると、Reset Missing Clock, Enter Search Mode, Disable DPLLのいずれかのコマンド(全てWR9)を発行するまでリセットされません。

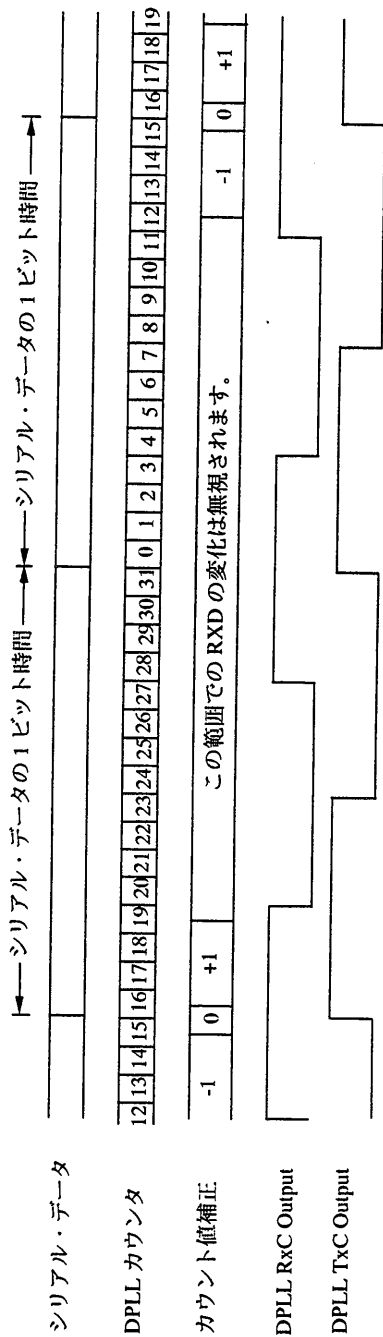


図 11-13 DPLL を FM モードで動作させた場合

11.11 送受信クロック

WR6の設定により、送信クロック、受信クロックはそれぞれ独立に

- (1) RTXCH入力
- (2) TRXCH入力
- (3) BRG出力
- (4) DPLL出力

- (1) RTXCH入力 $1/8 \times f(\text{CLKH})$
 $= 1/4 \times f(\text{CLK})$
- (2) TRXCH入力 $1/8 \times f(\text{CLKH})$
 $= 1/4 \times f(\text{CLK})$
- (3) BRG出力 $1/8 \times f(\text{CLKH})$
 $= 1/4 \times f(\text{CLK})$
- (4) DPLL出力 $1/16 \times f(\text{CLKH})$
 $= 1/8 \times f(\text{CLK})$

のいずれかを選択することが出来ます。

シリアルデータの最大転送速度は、ループモードに設定するか否かで異なってきます。ループモードに設定しない場合(通常モード)では、システムクロック CLK の周波数を $f(\text{CLK})$ 、基本動作クロック CLKH の周波数を $f(\text{CLKH})$ とすると、CLKH には CLK の2倍の信号が供給されているので、各送受信クロックを用いた場合の転送速度(bps)は以下のようになります。

- (1) RTXCH入力 $1/4 \times f(\text{CLKH})$
 $= 1/2 \times f(\text{CLK})$
- (2) TRXCH入力 $1/4 \times f(\text{CLKH})$
 $= 1/2 \times f(\text{CLK})$
- (3) BRG出力 $1/4 \times f(\text{CLKH})$
 $= 1/2 \times f(\text{CLK})$
- (4) DPLL出力 $1/16 \times f(\text{CLKH})$
 $= 1/8 \times f(\text{CLK})$

すなわち、CLKH が 20MHz のとき (システムクロックが 10MHz のとき)、DPLL 出力以外を送受信クロックとする場合は最大転送速度は 5Mbps、DPLL 出力を送受信クロックとする場合は 1.25Mbps となります。

ループモードに設定した場合の最大転送速度は、DPLL の出力以外を送受信クロックとするときに、ノーマルモードの最大転送速度の半分になります。DPLL の出力を送受信クロックとする場合は、最大転送速度はノーマルモードと同じです。すなわち、各送受信クロックを用いた場合の最大転送速度(bps)は以下のようになります。

また、ループモードでは送信クロックと受信クロックの周波数が同じであることが必要です。さらに、NRZ、NRZI モードのときは送信クロックと受信クロックが同相でなければなりません。FM0、FM1 モードの場合は、受信クロックが送信クロックよりも位相が遅いと、正しくループモードに移行することが出来ません。理想的な位相関係は、受信クロックが送信クロックよりも 1/4 だけ進んでいる状態です。送受信クロックを DPLL 出力に設定した場合は、DPLL が理想的な位相の送受信クロックを生成するため、このようなことを気にする必要はありません。

11.12 レジスタ設定

以下にKP91Aのレジスタ設定例を示します。ハードウェア・リセット、ソフトウェア・リセット後のレジスタの初期値については「11.13 リセット時の動作」を参照して下さい。

一般的な設定手順

以下にKP91Aの一般的な設定手順を示します。()の部分は省略可能です。また、必ずしもこの順序で設定する必要はありません。

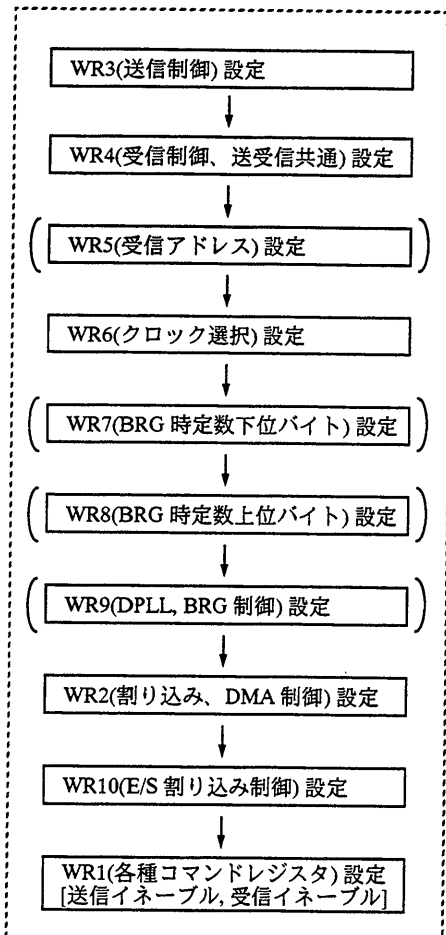


図 11-14 KP91A の一般的な設定手順

【設定例 1】

KP91A が以下のような動作を行うようにするには、表11-4のようにレジスタを設定します。

- ・ Flag Idle
- ・ アドレス比較ディスエーブル
- ・ NRZモード
- ・ CRC Preset 1
- ・ 送信クロック、受信クロック=RTXCH
- ・ Tx Underrun/EOM割り込み、Abort割り込みイネーブル
- ・ 送信データ、受信データは割り込みで転送
- ・ Rx Int On Rx FIFO RCAモード

表 11-4 レジスタ設定例 1

	D7	D6	D5	D4	D3	D2	D1	D0	備考
WR3	0	0	?	?	0	0	0	0	
WR4	0	0	0	0	0	1	0	0	
WR6	0	0	0	0	X	0	0	0	
WR2	0	0	0	0	1	0	0	1	
WR10	0	1	0	0	X	0	1		Tx U/EOM IE
WR1	0	0	0	0	0	0	1		Tx Enable
WR1	0	0	0	0	1	0	0		Rx Enable
WR10	1	1	0	0	X	0	1		Abort IE (注 1)

(注 1) Rx Disable 状態のとき、KP91A の内部では、RXD= '1' にマスクされます。したがって、Abort IE= '1' に設定した後に Rx Enable にすると、RXD 端子の状態に関わらず、Abort による E/S Int が発生します。これを避けたい場合は、設定例1のように Rx Enable にした後に Abort IE= '1' に設定して下さい。

【設定例 2】

設定例2はFMモードでBRG、DPLLを使う場合の例です。

- ・ Mark Idle
- ・ 受信アドレス全ビット比較
- ・ FM0モード
- ・ CRC Preset 1
- ・ 受信アドレス= 'AAH'
- ・ 受信クロック= DPLL Output
- ・ 送信クロック= BRG Output
- ・ BRG時定数= '0006H'
- ・ BRGソースクロック= RTXCH
- ・ DPLLソースクロック= RTXCH
- ・ Tx Underrun/EOM割り込みイネーブル
- ・ 送信データ、受信データともDMAで転送

表 11-5 レジスタ設定例 2

	D7	D6	D5	D4	D3	D2	D1	D0	備考
WR3	0	0	?	?	1	0	0	0	
WR4	1	0	1	1	0	1	0	0	
WR5	1	0	1	0	1	0	1	0	
WR6	1	1	1	0	X	0	0	0	
WR7	0	0	0	0	0	1	1	0	BRG(low)= '06H'
WR8	0	0	0	0	0	0	0	0	BRG(high)= '00H'
WR9	0	1	0	0	0	0	0	1	BRG Enable
WR9	0	1	0	0	0	1	0	1	Enter Search Mode
WR2	1	0	0	0	1	1	0	0	(注 2)
WR10	0	1	0	0	0	X	0	1	
WR1	0	0	0	0	0	0	0	1	Tx Enable
WR1	0	0	0	0	1	0	0	0	Rx Enable

(注 2) KP91A からの送信 DMA 要求は Tx Enable, Tx Disableに関わらず、送信 FIFOがEmpty状態のときに出力されます。したがって、DMAコントローラの設定が完了し、送信の準備が整った後に、Tx DMA Request Enableにするようにして下さい。

【設定例 3】

最後は、NRZIモードでループモード2次局に設定する場合の例です

- ・ ループモード
- ・ 受信アドレス全ビット比較
- ・ NRZIモード
- ・ CRC Preset 1
- ・ 受信アドレス= '55H'
- ・ 受信クロック= RTXCH、送信クロック= TRXCH
- ・ Tx Underrun/EOM割り込みイネーブル
- ・ 送信データ、受信データとも割り込みで転送

表 11-6 レジスタ設定例 3

	D7	D6	D5	D4	D3	D2	D1	D0	備考
WR3	0	0	?	?	1	0	0	0	
WR4	1	0	0	1	0	1	0	0	
WR5	0	1	0	1	0	1	0	1	
WR6	0	0	0	1	X	0	0	0	
WR2	0	0	0	0	1	0	0	1	
WR10	0	1	0	0	0	X	0	1	Tx U/EOM IE
WR1	0	0	0	0	1	0	0	0	Rx Enable (注 3)
WR3	0	0	?	?	1	0	0	1	Loop Mode Enable
WR3	0	0	?	?	1	0	1	1	GAOP= '1'
WR1	0	0	0	0	0	0	0	1	Tx Enable (注 4)

(注 3) 必ず、Loop Mode Enable= '1' にする前に Rx Enableにして下さい。

(注 4) On Loop= '1' になったのを確認してから、Tx Enable にして下さい。On Loop= '0' の状態で、Tx Enableにした場合の動作は保証致しません。

11.13 リセット時の動作

ハードウェア・リセットおよびソフトウェア・リセット時には、KP91Aは以下のようになります。

- (1) Tx FIFO、Rx FIFOはリセットされます(FIFOのポインタがリセットされるだけで、内容が All '0' に初期化されるわけではありません)。
- (2) Tx Disable 状態、Rx Disable 状態になります。
- (3) WR2= '00H' (割り込み要求、DMA要求はすべてディセーブル) になります。
- (4) WR3= '00H' (Local Loopback Disabl.、 Auto Echo Disable、 DTR_端子= "H"、 RTS_端子= "H"、 Flag Idle、 Flag On Underrun、 GAOP= '0'、 Loop Mode Disable) になります。
- (5) WR4='00H' (Address Compare Disable、 Address Compare Mode Full、 NRZ Data Encoding、 Rx CRC Check Enable、 CRC Preset 0、 CRC-SDLC、 Auto Echo Enable= '0') になります。
- (6) WR5 (受信アドレスレジスタ) は初期化されません。
- (7) WR6= '10H' (RXC=RTXCH端子、 TXC=TRXCH端子、 TRXCH Input、 TRXCH Output=RXC) になります。
- (8) WR7 (BRG Time Constant Low Byte) は初期化されません。
- (9) WR8 (BRG Time Constant High Byte) は初期化されません。
- (10) WR9= '00H' (DPLL Half/Full_='0'、 DPLL FM/NRZI_='0'、 DPLL Source= RTXCH、 DPLL Disable、 BRG Source=RTXCH、 BRG Disable) になります。
- (11) WR10= 'F8H' (Abort IE= '1'、 Tx Underrun/EOM= '1'、 CTS IE= '1'、 Hunt IE= '1'、 DCD IE= '1'、 Zero Count IE= '0'、 E/S Int Disable) になります。
- (12) RR1= '01X1_X100' (Abort= '0'、 Tx Underrun/EOM= '1'、 CTS 不定、 Hunt= '1'、 DCD不定、 TXBE= '1'、 ZC= '0'、 RCA= '0'

) になります。

- (13) RR2= '00H' (EOF= '0'、 CRC Error Detect= '0'、 Rx Overrun Error= '0'、 Residue= '0'、 Sp Int='0'、 Rx Int='0'、 Tx Int='0'、 E/S Int='0') になります。
- (14) RR3~RR8はWR3~WR8のリセット後の状態が読み出されます。
- (15) RR9= '00H' (One Clock Missing= '0'、 Two Clock Missing= '0'、 Loop Sending= '0'、 On Loop= '0') になります。
- (16) RR10= 'F8H'(WR10のリセット後の状態)
- (17) RR2='00H'(WR2のリセット後の状態)
- (18) RR9='00H'(WR9のリセット後の状態)

表 11-7 にリセット後の各レジスタの初期値の一覧を示します。

表 11-7 リセット後の各レジスタの初期値

	D7	D6	D5	D4	D3	D2	D1	D0
WR0	X	X	X	X	X	X	X	X
WR1	-	-	-	-	-	-	-	-
WR2	0	0	0	0	0	0	0	0
WR3	0	0	0	0	0	0	0	0
WR4	0	0	0	0	0	0	0	0
WR5	X	X	X	X	X	X	X	X
WR6	0	0	0	1	0	0	0	0
WR7	X	X	X	X	X	X	X	X
WR8	X	X	X	X	X	X	X	X
WR9	0	0	0	0	-	-	0	0
WR10	1	1	1	1	1	0	0	0
RR0	X	X	X	X	X	X	X	X
RR1	0	1	X	1	X	1	0	0
RR2	0	0	0	0	0	0	0	0
RR3	0	0	0	0	0	0	0	0
RR4	0	0	0	0	0	0	0	0
RR5	X	X	X	X	X	X	X	X
RR6	0	0	0	1	0	0	0	0
RR7	X	X	X	X	X	X	X	X
RR8	X	X	X	X	X	X	X	X
RR9	0	0	0	0	0	0	0	0
RR10	1	1	1	1	1	0	0	0
RR2'	0	0	0	0	0	0	0	0
RR9'	0	0	0	0	0	0	0	0

12. パラレルポート

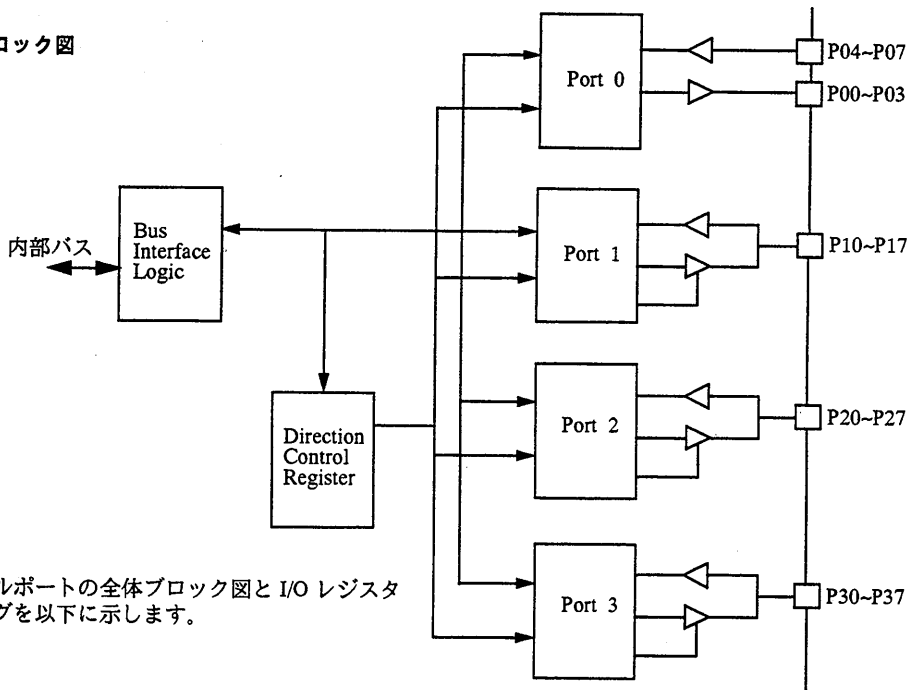
12.1 概要

KL5C80A20 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

12.2 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 12-1 I/O レジスタマッピング

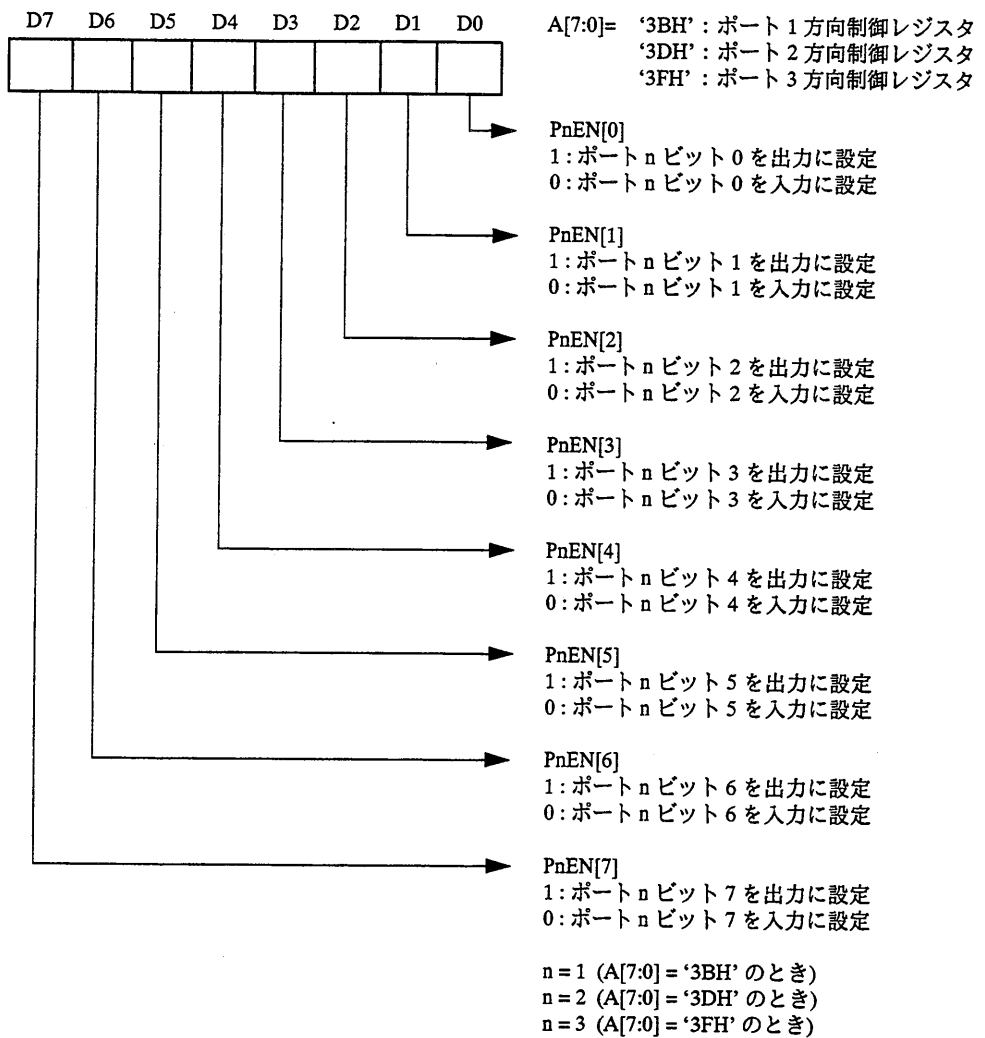
I/O	ライト時	リード時
38H	ポート0	ポート0
39H	ビット操作コマンド	ポート0方向制御レジスタ(固定)
3AH	ポート1	ポート1
3BH	ポート1方向制御レジスタ	ポート1方向制御レジスタ
3CH	ポート2	ポート2
3DH	ポート2方向制御レジスタ	ポート2方向制御レジスタ
3EH	ポート3	ポート3
3FH	ポート3方向制御レジスタ	ポート3方向制御レジスタ

12.3 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

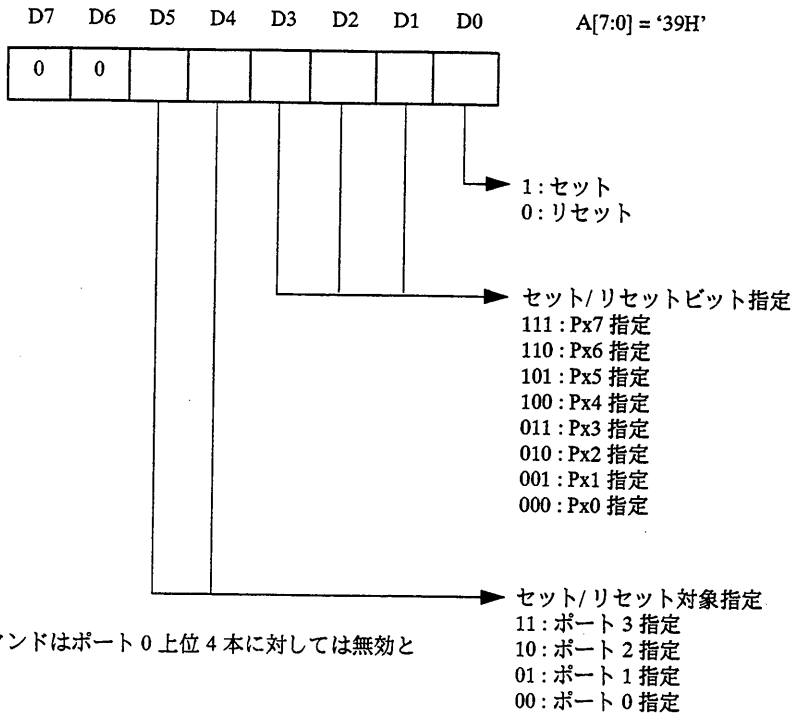
はパラレルポートの入出力の方向設定を各ポートの方向制御レジスタにより 1 ビット単位で設定できます。

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。

12.4 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。

12.5 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。(全λη)

- 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

12.6 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。

13. 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。

13.1 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

KL5C80A20 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。

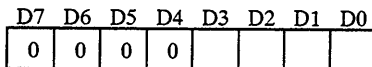
バグ・ファインダ Boot-on-RAM モード

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。

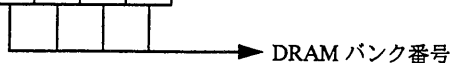
13.2 システムコントロールレジスタ

KL5C80A20 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 ~ SCR5 の6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 = '1AH')



ハードウェアリセットにより全ビット'0'となります。



'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 = '1BH')

D7 D6 D5 D4 D3 D2 D1 D0

0	0						
---	---	--	--	--	--	--	--

ハードウェアリセットにより全ビット'0'となります。

- 00: IR14 はタイマ/カウンタチャンネル1を使用。
IR8 は UART RXRDY0+BDET0を使用。
IR7 は UART TXRDY0を使用。
IR6 はクロックシリアルI/O 送受信割り込みを使用。
 - 01: IR14 は外部入力 P07 を使用。
IR8 は UART RXRDY0+BDET0を使用。
IR7 は UART TXRDY0を使用。
IR6 はクロックシリアルI/O 送受信割り込みを使用。
 - 10: IR14 はタイマ/カウンタチャンネル1を使用。
IR8 は UART RXRDY0+BDET0を使用。
IR7 は UART TXRDY0を使用。
IR6 は外部入力 P07 を使用。
 - 11: IR14 はタイマ/カウンタチャンネル1を使用。
IR8 は UART TXRDY0+RXRDY0+BDET0を使用。
IR7 は外部入力 P07 を使用。
IR6 はクロックシリアルI/O 送受信割り込みを使用。
-
- 00: IR13 はタイマ/カウンタチャンネル0を使用。
IR5 は DMA コントローラチャンネル1 DMTC 出力を使用。
IR2 は UART RXRDY1+BDET1を使用。
IR1 は UART TXRDY1を使用。
 - 01: IR13 は外部入力 P21 を使用。
IR5 は DMA コントローラチャンネル1 DMTC 出力を使用。
IR2 は UART RXRDY1+BDET1を使用。
IR1 は UART TXRDY1を使用。
 - 10: IR13 はタイマ/カウンタチャンネル0を使用。
IR5 は外部入力 P21 を使用。
IR2 は UART RXRDY1+BDET1を使用。
IR1 は UART TXRDY1を使用。
 - 11: IR13 はタイマ/カウンタチャンネル0を使用。
IR5 は DMA コントローラ チャンネル1 DMTC 出力を使用。
IR2 は UART TXRDY1+RXRDY1+BDET1を使用。
IR1 は外部入力 P21 を使用。
-
- 00: IR4 は DMA コントローラチャンネル0 DMTC 出力を使用。
IR3 はタイマ/カウンタチャンネル3を使用。
IR0 はタイマ/カウンタチャンネル2を使用。
 - 01: IR4 は外部入力 P22 を使用。
IR3 はタイマ/カウンタチャンネル3を使用。
IR0 はタイマ/カウンタチャンネル2を使用。
 - 10: IR4 は DMA コントローラチャンネル0 DMTC 出力を使用。
IR3 は外部入力 P22 を使用。
IR0 はタイマ/カウンタチャンネル2を使用。
 - 11: IR4 は DMA コントローラチャンネル0 DMTC 出力を使用。
IR3 はタイマ/カウンタチャンネル3を使用。
IR0 は外部入力 P22 を使用。

SCR2(I/O アドレス = 1CH)

D7	D6	D5	D4	D3	D2	D1	D0
0							

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、SCR1 で割り込みコントローラの IR 入力に接続するよう設定した場合、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、SCR1 で割り込みコントローラの IR 入力に接続するよう設定した場合、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、SCR1 で割り込みコントローラの IR 入力に接続するよう設定した場合、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 12 から入力される信号を、SCR1 で割り込みコントローラの IR 入力に接続するよう設定した場合、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。

00: 端子 81, 80, 77, 76 を P30, P31, P32, P33 として使用。CTSH₋ と DCDH₋ には "L" が入力されます。

01: 端子 81, 80 を P30, P31、端子 77 を RTSH₋、端子 76 を CTSH₋ として使用。

1X: 端子 81 を DTRH₋、端子 80 を DCDH₋、端子 77 を RTSH₋、端子 76 を CTSH₋ として使用。

KP61 チャンネル 0 - KP91 接続選択ビット

0: KP61 チャンネル 0, KP91 が独立。

1: KP61 チャンネル 0 の各信号と HDLC シリアルインターフェースの各信号が内部で接続されます。

RXD0 と RXDH が

TXD0 と TXDH が

CTS0 と CTSH₋ が

RTS0 と RTSH₋ が

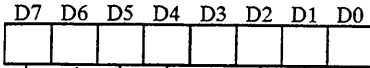
DTR0 と DTRH₋ が

TRXC0 と RTXCH が

それぞれ内部的に接続されます。

※ KP61(UART) チャンネル 0 と KP91(HDLC インターフェイス) を D6 により内部接続する場合、各信号は KP61 チャンネル 0 の信号端子を使用してください。

SCR3(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 : DMA コントローラチャンネル1 DREQ入力として端子85を使用します。
端子86はP23として機能します。
- 01 : DMA コントローラチャンネル1 DREQ入力として端子85を使用します。
端子86はDACK1_として機能します。
- 10 : DMA コントローラチャンネル1 DREQ入力としてUARTのチャンネル0 RXRDY 出力を使用します。
端子86はP23として機能します。
- 11 : DMA コントローラチャンネル1 DREQ入力としてHDLCシリアルインターフェースの REQR 出力を使用します。
端子86はP23として機能します。

※ P24 (端子 85) はこれらの設定に関係なくポートとして機能します。

- 0 : 端子 91 を P17 として使用します。
- 1 : 端子 91 を EXBACK_ として使用します。

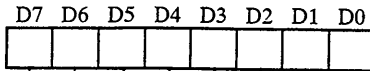
- 00 : DMA コントローラチャンネル0 DREQ入力として端子84使用、端子5はPALATとして使用します。
- 01 : DMA コントローラチャンネル0 DREQ入力として端子84使用、端子5はDACK0として使用します。
- 10 : DMA コントローラチャンネル0 DREQ入力としてUARTチャンネル0 TXRDY 出力を使用、端子5はPALATとして使用します。
- 11 : DMA コントローラチャンネル0 DREQ入力としてHDLCシリアルインターフェース REQT 出力を使用、端子5はPALATとして使用します。

※ P25 (端子 84) はこれらの設定に関係なくポートとして機能します。

- 0 : 端子 82, 83 を P27, P26 として使用。
DSR0_ には "L" が入力されます。
- 1 : 端子 82, 83 を DSR0_, DTR0_ として使用。

- 00 : 端子 72, 74, 75 を P37, P35, P34 として使用。
CTS1_ には "L" が入力されます。
- 01 : 端子 74, 75 を P35, P34、端子 72 を TXD1 として使用。
CTS1_ には "L" が入力されます。
- 1X : 端子 72 を TXD1、端子 74 を CTS1_、端子 75 を RTS1_ として使用。

SCR4(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

refresh enable on external DMA

- 0: 外部 DMA 受け付け中の DRAM リフレッシュを禁止
- 1: 外部 DMA 受け付け中の DRAM リフレッシュを行う。

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。

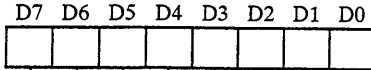
'80000H ~ BFFFFH' を DRAM 領域として

- 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_
端子 97 は CAS_
端子 98 は RAS_ として機能します。

NMI_ に

- 00: "H" (インアクティブ)
- 01: タイマのチャネル 3 からの割り込み要求
- 10: 端子 13
- 11: (タイマのチャネル 3 からの割り込み要求) + (端子 13) を接続します。

SCR5(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
端子 23 を P00 として使用します。
- 01 : 端子 20 を P03
端子 21 を P02
端子 22 を P01
端子 23 をタイマの OUT0 として使用します。
- 10 : 端子 20 を P03
端子 21 を P02
端子 22 をタイマの OUT1
端子 23 をタイマの OUT0 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
端子 23 をタイマの OUT0 として使用します。

- 0 : 端子 99 を P10 として使用します。
- 1 : 端子 99 を SCK 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 TXS として使用します。

外部メモリ・ウェイト・コントロール(表 13-1参照)

外部 I/O・ウェイト・コントロール(表 13-2参照)

表 13-1 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13-2 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。

13.3 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 SCR3 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、タイム/カウンタのチャンネル1をゲート機能無しで、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, 23の使い方

端子 20, 21, 22, 23 は出力固定です。これらの端子の機能は SCR5 によって設定します。

端子 93,73 の使い方

端子93, 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 93, 73 を RXS, RXD1 として使用する場合は P15, P36 を入力方向に設定して下さい。このとき、RXS と P15、RXD1 と P36 には同じ外部信号が入力されます。

また端子 93, 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 RXS, 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 を受信用を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うことをお勧めします。

端子13 の使い方

SCR4 の D<7:6> を '10' に設定した場合、P06 と NMI_ に同じ信号が入力されます。

端子 13 をパラレルポートとして使用する場合、

SCR4 の D7 を '0' にすることによって NMI_ と端子13 を切り離して使用して下さい。

端子 92 の使い方

端子 92 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92を EXBREQ_ の機能で使用する場合は P16 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 P16 と EXBREQ_ には同じ信号が入力されます。

この端子をパラレルポートとして使用する場合は、このポートの信号が 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って下さい。

パラレルポートとマルチプレクスされているすべての入力端子 (端子 99 を除く) は、各々のパラレルポートの入出力方向制御レジスタによって決定されます。したがって、パラレルポート以外の機能で使う場合はパラレルポートの方向制御レジスタを入力方向に設定して下さい。

端子 99 の入出力方向は SCR で選択した回路の設定に従います。

14. アドレス・マッピング

内部I/Oマッピング

表 14-1 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		SCR0	SCR0
1BH		SCR1	SCR1
1CH		SCR2	SCR2
1DH		SCR3	SCR3
1EH		SCR4	SCR4
1FH		SCR5	SCR5
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 14-1 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	チャンネル0 RATE	チャンネル0 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 RATE	チャンネル1 RATE	
2DH		川崎製鉄予約	川崎製鉄予約	
2EH		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	
2FH		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	
30H		クロック同期	送信データ	受信データ
31H		シリアルI/O	コマンド/モード設定	ステータス
32H	割り込みコントローラ	川崎製鉄予約	川崎製鉄予約	
33H		川崎製鉄予約	川崎製鉄予約	
34H		LERL/PGRL	ISRL	
35H	割り込みコントローラ	LERH/PGRH	ISRH	
36H		IMRL	IMRL	
37H		IVR/IMRH	IMRH	
38H		ポート0	ポート0	
39H	パラレルポート	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	
40H		HDLCシリアルインタフェース	WR0(送信データレジスタ)	RR0(受信データレジスタ)
41H	WR1		RR1	
42H	WR2		RR2	
43H	WR3		RR3	
44H	WR4		RR4	
45H	WR5		RR5	
46H	WR6		RR6	
47H	WR7		RR7	
48H	WR8		RR8	
49H	WR9		RR9	
4AH	WR10		RR10	
4BH	川崎製鉄予約	RR2'		
4CH	川崎製鉄予約	RR9'		
4DH~4FH	川崎製鉄予約			

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

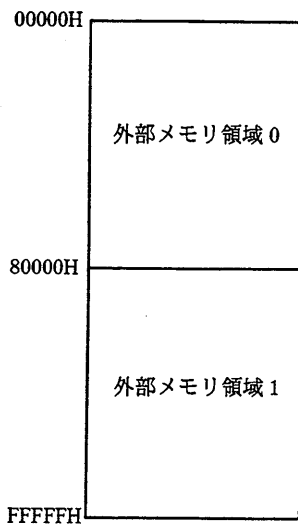
内蔵DRAMコントローラを使用しない場合

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)

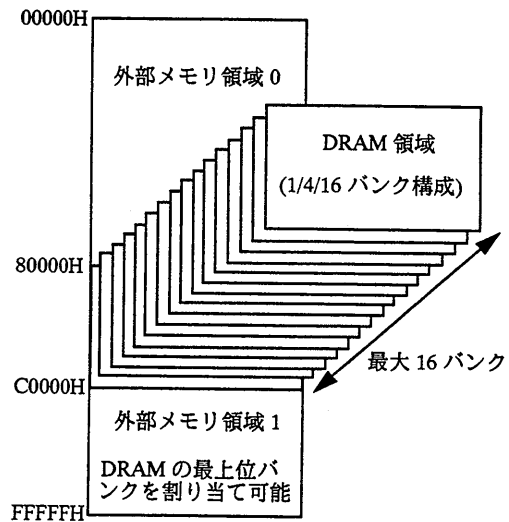
内蔵DRAMコントローラを使用する場合

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当て可能)
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 14-1 メモリマッピング

15. 発振回路

15.1 概要

KL5C80A20 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。

15.2 回路構成

システムクロックを発生させるためには KL5C80A20 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 15-1 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 15-2 外部部品定数参考範囲

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

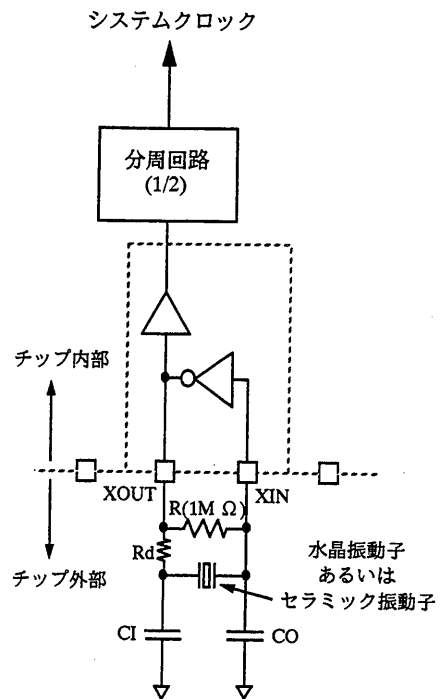


図 15-1 発振回路の構成

16. 電気的特性

16.1 絶対最大定格

表 16-1 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

16.2 DC 特性 (5V ± 10%)

表 16-2 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 16-3 電気的特性 (推奨動作条件での特性)

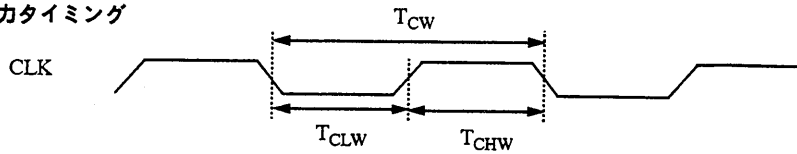
項目	記号	規格値			単位	測定条件
		最小	標準	最大		
入力電圧 (RESET_ 以外の入力端子)	V _{IH}	3.6		VDD	V	
	V _{IL}	GND		1.4	V	
RESET_ 入力端子 (シュミットトリガ入力)	V ₊	2.4		4.0	V	
	V ₋	0.9		2.3	V	
	V _h	0.9			V	
出力電圧	V _{OH}	3.5			V	I _{OH} = -4mA or -6mA
	V _{OL}			0.4	V	I _{OL} = 4mA or 6mA
出力電流	I _{OUT}			±6	mA	(注 1)
				±4	mA	(注 2)
入力リーク電流	I _{IL}	-10			μA	V _{IN} = GND
	I _{IH}			10	μA	V _{IN} = VDD
出力リーク電流	I _{oZ}	-10		10	μA	ハイ・インピーダンス出力時
プル・アップ電流	I _{PU}	20	95	250	μA	V _{IN} = GND
プル・ダウン電流	I _{PD}	20	95	250	μA	V _{IN} = VDD
スタンバイ電流	I _{DDS}		1.0*	100	μA	CLK 停止時
動作時消費電流	I _{DDOP}		45*		mA	f (CLK) = 10MHz

* TA = 25 °C のとき

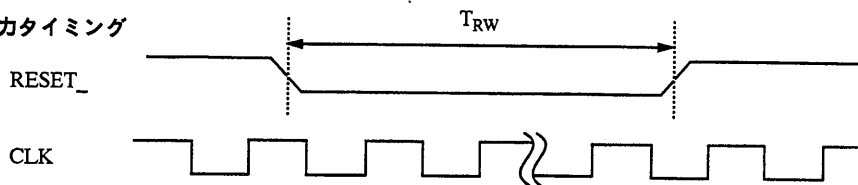
注 1) A[19:0], D[7:0], EMRD_, EMWR_, EIORD_, EIOWR_, CLK の各出力端子
注 2) 上記以外の出力端子

16.3 AC 特性

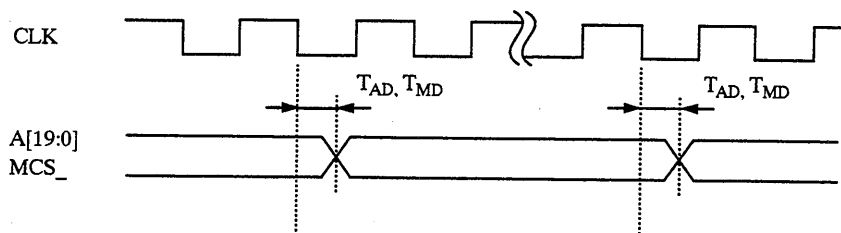
クロック出カタイミング



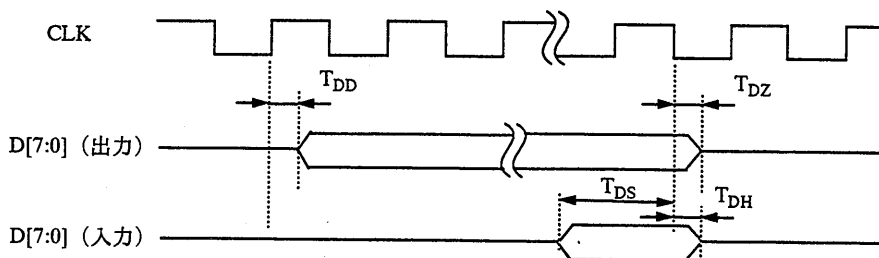
リセット入カタイミング



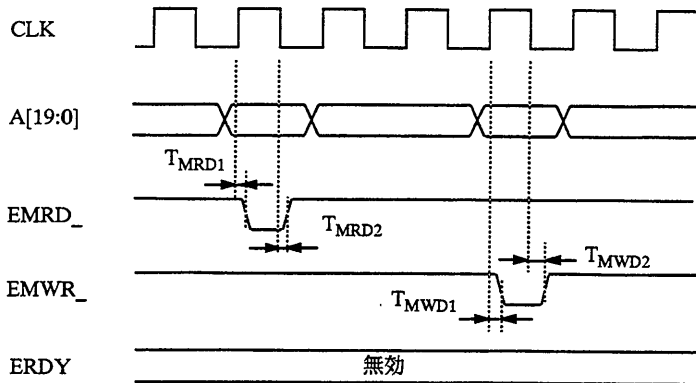
アドレス出カタイミング



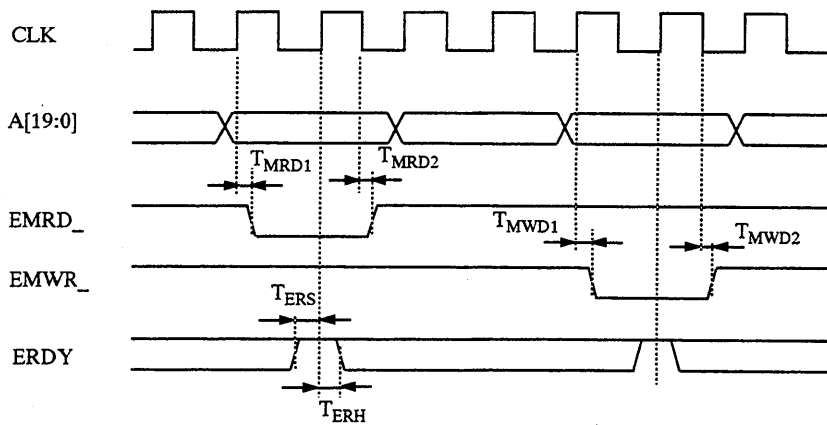
データ入出カタイミング



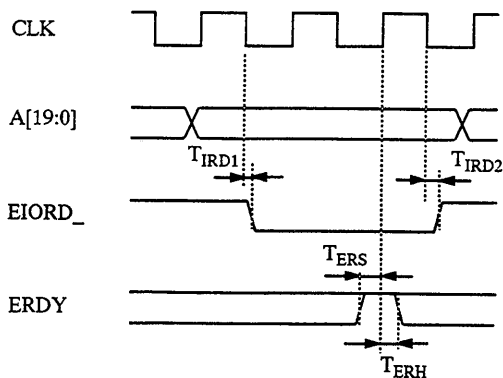
外部メモリアクセス (0 ウェイト)



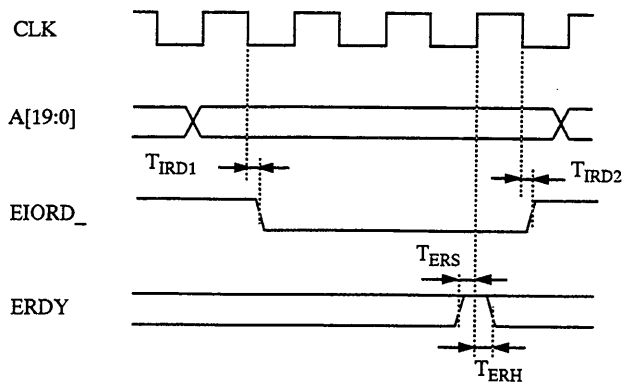
外部メモリアクセス (1 ウェイト)



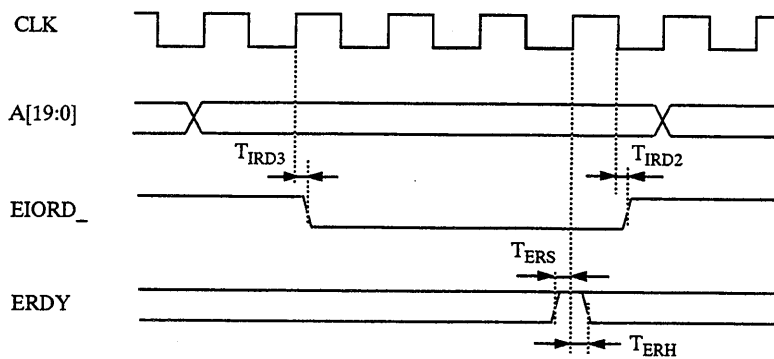
外部 I/O リード (1 ウェイト)



外部 I/O リード (2 ウェイト)

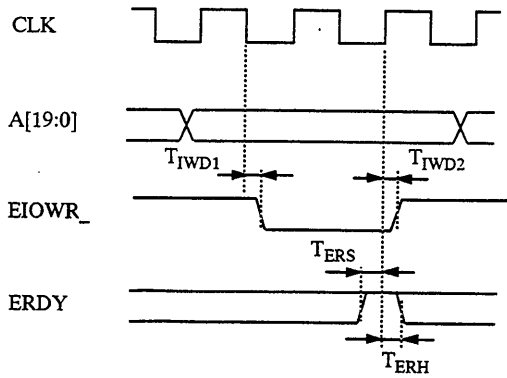


外部 I/O リード (4 ウェイト)

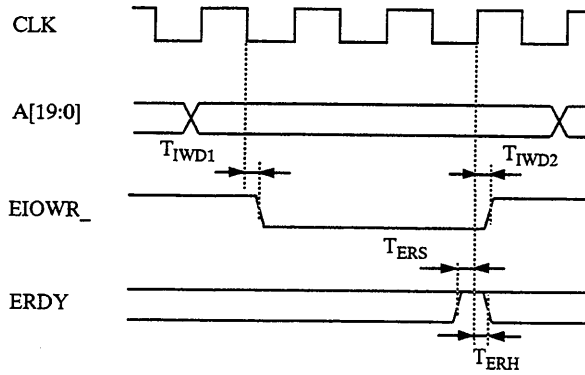


※ ウェイトコントロールレジスタ (SCR5) により 4 ウェイトを設定した場合、EIORD_ の開始エッジが 1/2 クロック遅れます。

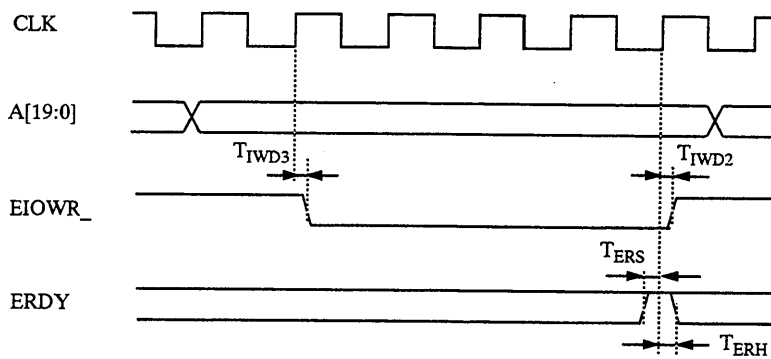
外部 I/O ライト (1 ウェイト)



外部 I/O ライト (2 ウェイト)

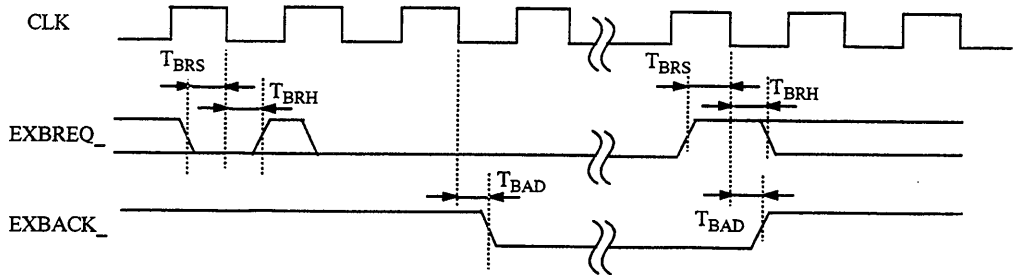


外部 I/O ライト (4 ウェイト)

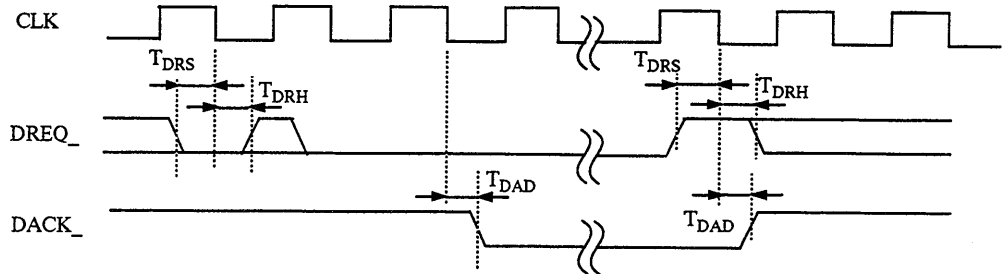


※ ウェイトコントロールレジスタ (SCR5) により 4 ウェイトを設定した場合、EIOWR_ の開始エッジが 1/2 クロック遅れます。

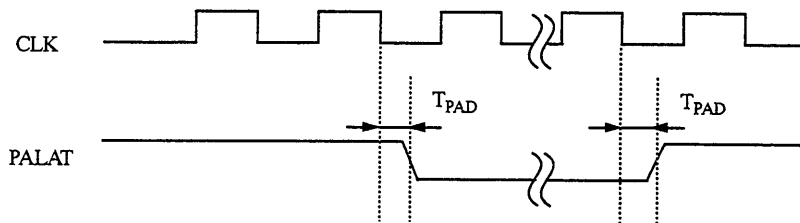
バスコントロールタイミング



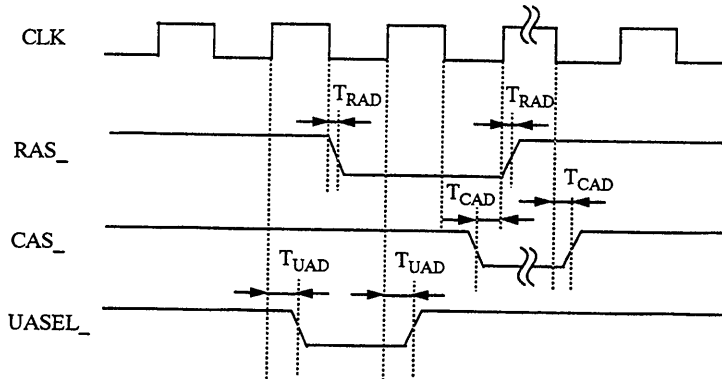
DMA コントロールタイミング



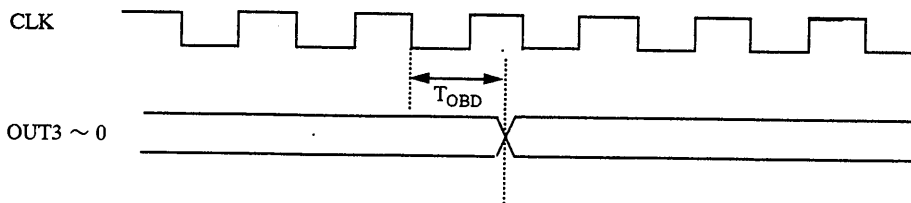
PALAT 出力タイミング



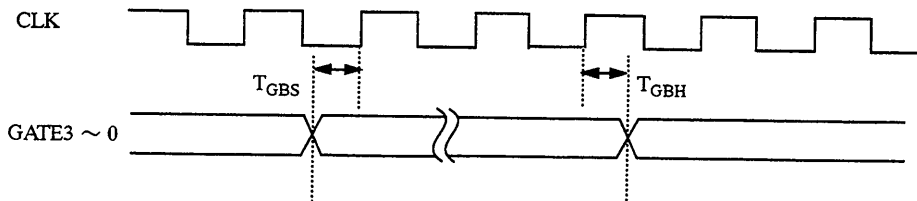
DRAM コントロールタイミング



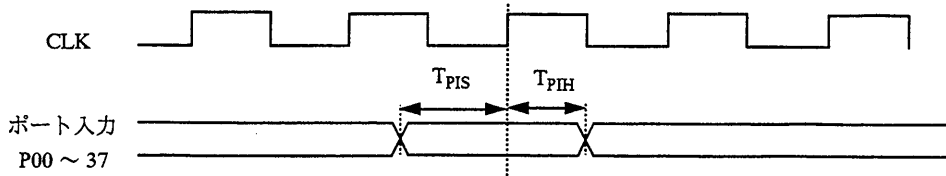
カウンタ出力タイミング



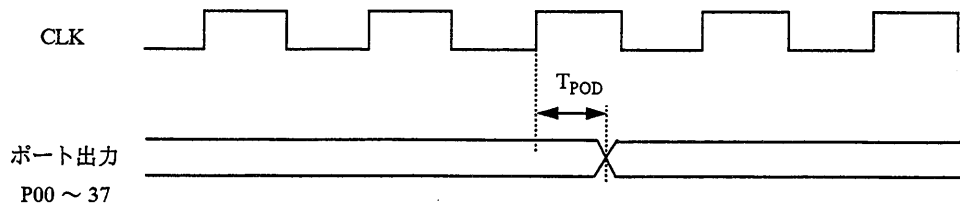
カウンタゲートタイミング



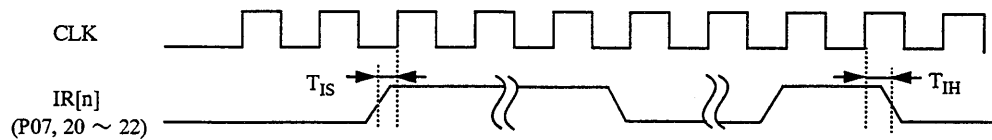
ポート入力タイミング



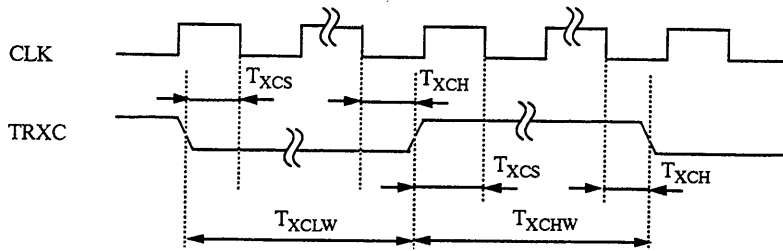
ポート出力タイミング



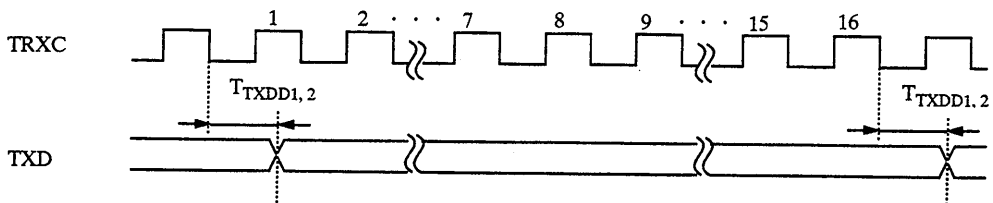
外部割り込み入力タイミング



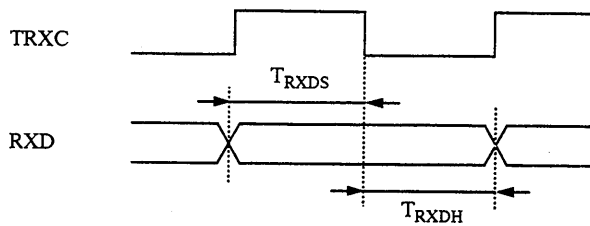
UART 送受信クロック



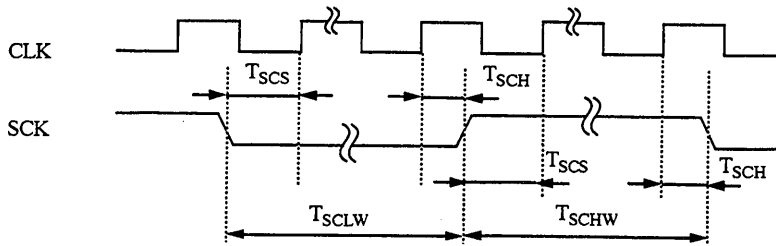
UART 送信クロックと送信データ



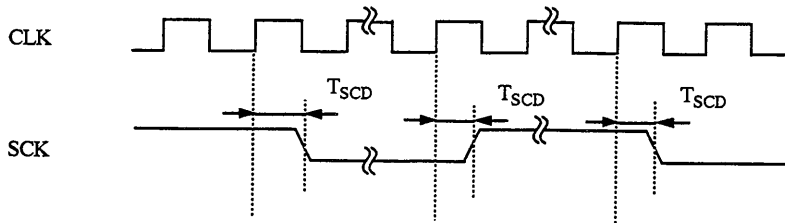
UART 受信クロックと受信データ



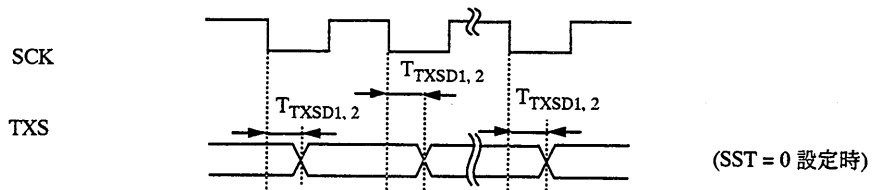
シリアルクロック入力タイミング



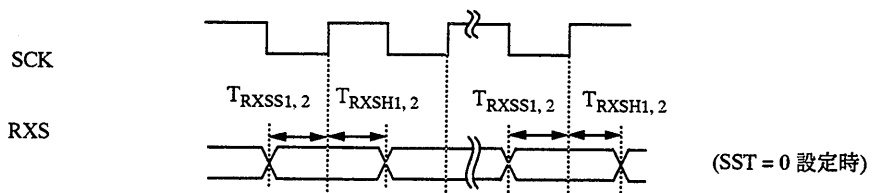
シリアルクロック出力タイミング



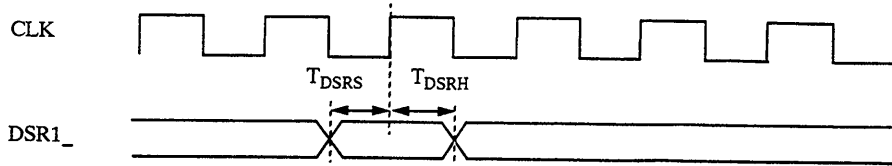
クロック同期シリアル送信データタイミング



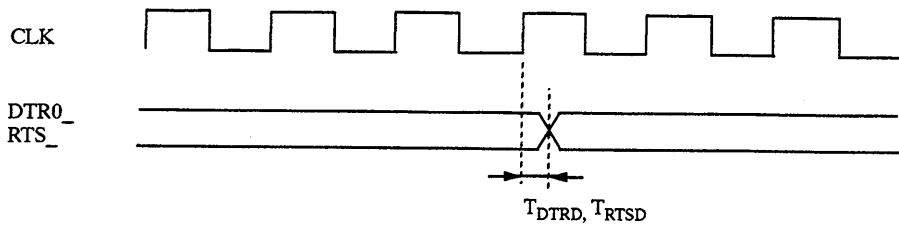
クロック同期シリアル受信データタイミング



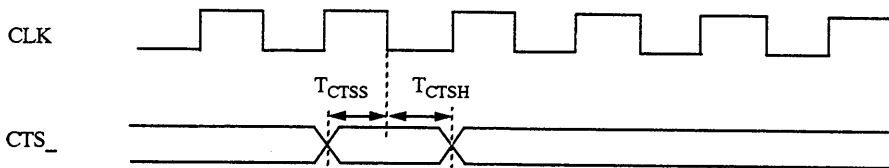
DSR1_ 入カタイミング



DTR0_, RTS_ 出カタイミング

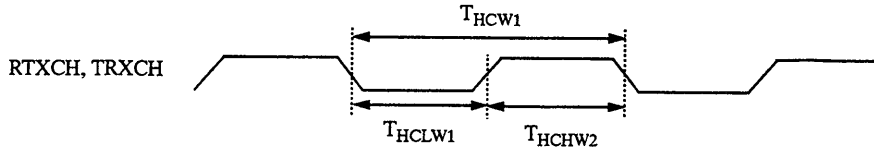


CTS_ 入カタイミング

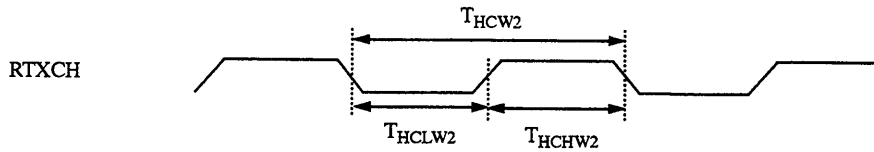


HDLC インターフェース関連 AC 特性

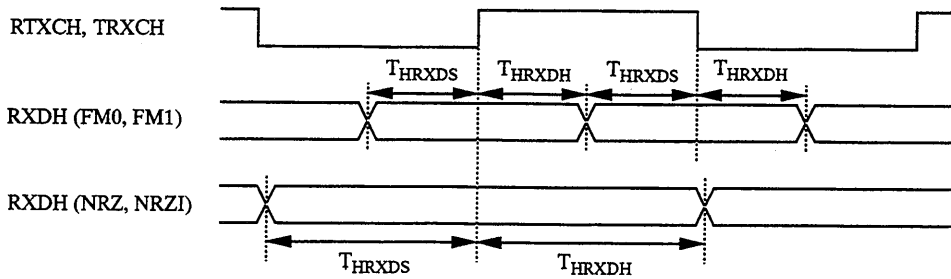
送受信クロックタイミング



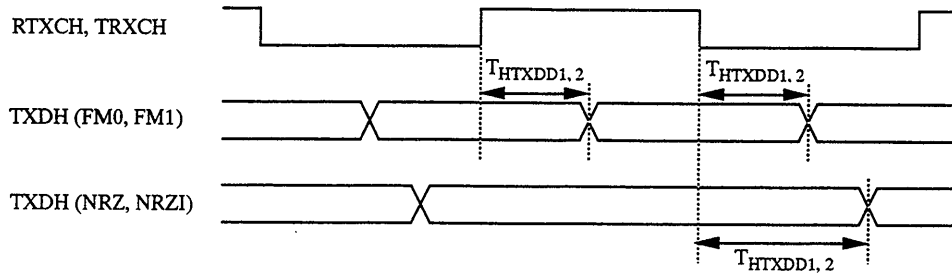
ソースクロックタイミング (DPLL, ボーレートジェネレータ用)



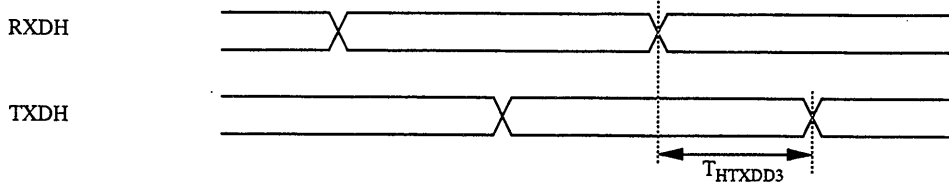
受信データタイミング



送信データタイミング



送信データタイミング (Auto Echo, ループモード・ゲート遅延状態時)



KL5C80A20 AC 特性 (5V ± 10%)

番号	項目	最小	標準	最大	単位
T _{CYC}	XINサイクル時間	50.0			ns
T _{CW}	CLKサイクル時間	100.0			ns
T _{CLW}	CLK “L” パルス幅		50.0		ns
T _{CHW}	CLK “H” パルス幅		50.0		ns
T _{RW}	RESET_パルス幅	3			clk
T _{AD}	アドレス出力遅延時間	9.0		43.0	ns
T _{MD}	チップセレクト出力遅延時間			46.0	ns
T _{DD}	CLK→データ出力遅延時間			30.0	ns
T _{DZ}	CLK→データ出力off遅延時間	8.0			ns
T _{DS}	データ入力セットアップ時間	3.0			ns
T _{DH}	データ入力ホールド時間	5.0			ns
T _{MRD1}	CLK立ち上がりエッジ→EMRD_ “L” 遅延時間			25.0	ns
T _{MRD2}	CLK立ち下がりエッジ→EMRD_ “H” 遅延時間	6.0		25.0	ns
T _{MWD1}	CLK立ち上がりエッジ→EMWR_ “L” 遅延時間			25.0	ns
T _{MWD2}	CLK立ち下がりエッジ→EMWR_ “H” 遅延時間	6.0		25.0	ns
T _{ERS}	ERDY入力セットアップ時間	3.0			ns
T _{ERH}	ERDY入力ホールド時間	5.0			ns
T _{IRD1}	CLK立ち下がりエッジ→EIORD_ “L” 遅延時間			28.0	ns
T _{IRD2}	CLK立ち下がりエッジ→EIORD_ “H” 遅延時間	6.0		25.0	ns
T _{IRD3}	CLK立ち上がりエッジ→EIORD_ “L” 遅延時間			25.0	ns
T _{IWD1}	CLK立ち下がりエッジ→EIOWR_ “L” 遅延時間			29.0	ns
T _{IWD2}	CLK立ち上がりエッジ→EIOWR_ “H” 遅延時間			22.0	ns
T _{IWD3}	CLK立ち上がりエッジ→EIOWR_ “L” 遅延時間			25.0	ns
T _{BRS}	EXBREQ_入力セットアップ時間	3.0			ns
T _{BRH}	EXBREQ_入力ホールド時間	10.0			ns
T _{BAD}	EXBACK_出力遅延時間			40.0	ns
T _{DRS}	DREQ_入力セットアップ時間	3.0			ns
T _{DRH}	DREQ_入力ホールド時間	5.0			ns
T _{DAD}	DACK_出力遅延時間			40.0	ns
T _{PAD}	PALAT出力遅延時間			40.0	ns
T _{RAD}	RAS_出力遅延時間			29.0	ns
T _{CAD}	CAS_出力遅延時間			29.0	ns
T _{UAD}	UASEL_出力遅延時間			24.0	ns
T _{GS}	GATE入力セットアップ時間	3.0			ns
T _{GH}	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 _{OD}	タイマ/カウンタOUT出力遅延時間			40.0	ns
T _{PIS}	ポート入力セットアップ時間	5.0			ns
T _{PIH}	ポート入力ホールド時間	5.0			ns
T _{POD}	ポート出力遅延時間			40.0	ns
T _{IS}	外部割り込み入力セットアップ時間	5.0			ns
T _{IH}	外部割り込み入力ホールド時間	5.0			ns
T _{NS}	NMI_入力セットアップ時間	3.0			ns
T _{NH}	NMI_入力ホールド時間	5.0			ns
T _{XCLW}	TRXC “L” パルス幅	1			clk
T _{XCHW}	TRXC “H” パルス幅	1			clk
T _{XCS}	TRXC入力セットアップ時間	3.0			ns
T _{XCH}	TRXC入力ホールド時間	10.0			ns
T _{TXDD1}	TXD出力遅延時間(内部送受信クロック選択時)			40.0	ns
T _{TXDD2}	TXD出力遅延時間(外部クロックTRXC選択時)			40.0	ns
T _{RXDS}	RXD入力セットアップ時間	3.0			ns
T _{RXDH}	RXD入力ホールド時間	15.0			ns
T _{DSRS}	DSR1_入力セットアップ時間	5.0			ns
T _{DSRH}	DSR1_入力ホールド時間	3.0			ns
T _{CTSS}	CTS_入力セットアップ時間	5.0			ns
T _{CTSH}	CTS_入力ホールド時間	10.0			ns
T _{DTRD}	DTR0_出力遅延時間			40.0	ns
T _{RTSD}	RTS_出力遅延時間			40.0	ns
T _{SCLW}	SCK “L” パルス幅	1			clk
T _{SCHW}	SCK “H” パルス幅	1			clk
T _{SCS}	SCK入力セットアップ時間	3.0			ns
T _{SCH}	SCK入力ホールド時間	5.0			ns
T _{SCD}	SCK出力遅延時間			40.0	ns
T _{TXSD1}	TXS出力遅延時間(内部送受信クロック選択時)			40.0	ns
T _{TXSD2}	TXS出力遅延時間(外部クロックSCK選択時)			40.0	ns
T _{RXSS1}	RXS入力セットアップ時間(内部クロック選択時)	3.0			ns
T _{RXSH1}	RXS入力ホールド時間(内部クロック選択時)	10.0			ns
T _{RXSS2}	RXS入力セットアップ時間(外部SCK選択時)	3.0			ns
T _{RXSH2}	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_L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。

HDLCインターフェース関連

番号	項目	最小	標準	最大	単位
T _{CLKH}	CLKHサイクル時間 (=1/2*T _{CW})	50.0			ns
T _{HCW1}	送受信クロックサイクル時間	4*T _{CLKH}			ns
T _{HCLW1}	送受信クロック“L”パルス幅	2*T _{CLKH}			ns
T _{HCHW1}	送受信クロック“H”パルス幅	2*T _{CLKH}			ns
T _{HCW2}	ソースクロックサイクル時間	2.2*T _{CLKH}			ns
T _{HCLW2}	ソースクロック“L”パルス幅	1.1*T _{CLKH}			ns
T _{HCHW2}	ソースクロック“H”パルス幅	1.1*T _{CLKH}			ns
T _{HRXDS}	RXDH入力セットアップ時間	T _{CLKH}			ns
T _{HRXDH}	RXDH入力ホールド時間	0.0			ns
T _{HTXDD1}	TXDH出力遅延時間(対送受信クロック入力)			30.0	ns
T _{HTXDD2}	TXDH出力遅延時間(対TRXCH出力)	0.0		5.0	ns
T _{HTXDD3}	TXDH出力遅延時間(対RXDH入力、注3)			20.0	ns

注1) ここで用いられている基本動作クロック(CLKH)は、システムクロック(CLK)の1/2のサイクル時間で動作するチップ内部で生成されるHDLCインターフェース専用のクロックです。

注2) 出力端子の負荷容量C_Lは70pFで測定しています。

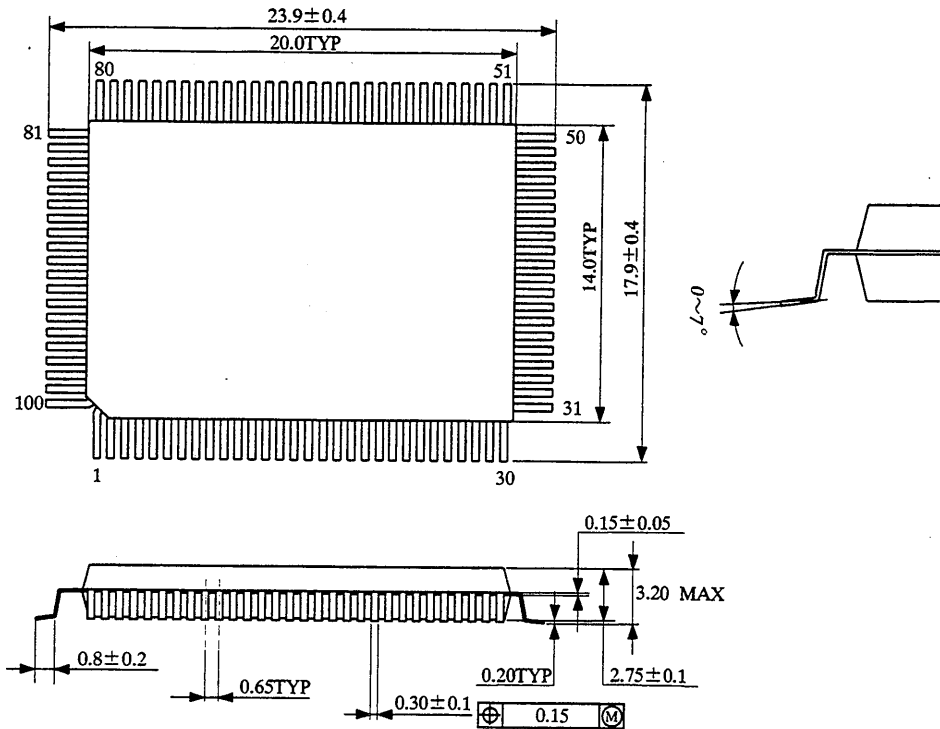
注3) T_{HTXDD3}はAuto Echo、ループモード・ゲート遅延状態時のタイミングです。

17 外形寸法図

KL5C80A20 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、『KL5C80A20C』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、KC80/KC82 CPUコア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”は各命令の名称を表し、“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。例えば、

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。但し、入出力命令の[r]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進数で記しています。また、その右側の“B”、“CL”は、各命令の長さ（バイト数）と内部バスサイクルで命令を実行するのに要するクロック数をそれぞれ表しています。ある命令の実行中に外部バスサイクルが存在するときには、外部バス・インターフェイス・ユニットに設定されているウェイト数だけ“CL”に加算したものが、実際にその命令を実行するのに必要とするクロック数になります。

この他、インストラクション表に出てくる略号に関しては、次の凡例にまとめてありますので参考にして下さい。

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
mn	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	Flags 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		2	2
			<- n ->			
LD r, [HL]	r<-[HL]	● ● ● ● ● ● ● ●	01 r 110		1	2
LD r, [IX+d]	r<-[IX+d]	● ● ● ● ● ● ● ●	11011101	DD	3	5
			01 r 110			
			<- d ->			
LD r, [IY+d]	r<-[IY+d]	● ● ● ● ● ● ● ●	11111101	FD	3	5
			01 r 110			
			<- d ->			
LD [HL], r	[HL]<-r	● ● ● ● ● ● ● ●	01110 r		1	2
LD [IX+d], r	[IX+d]<-r	● ● ● ● ● ● ● ●	11011101	DD	3	5
			01110 r			
			<- d ->			
LD [IY+d], r	[IY+d]<-r	● ● ● ● ● ● ● ●	11111101	FD	3	5
			01110 r			
			<- d ->			
LD [HL], n	[HL]<-n	● ● ● ● ● ● ● ●	00110110	36	2	3
			<- n ->			
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	Opcode		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 ->			
			<- n ->			
			11101101	ED		
LD A, R	A<-R	⇔ ⇔ 0 IF 0 ● *1	01010111	57	2	2
			<- d ->			
			<- n ->			
			11101101	ED		
LD I, A	I<-A	● ● ● ● ● ●	01011111	5F	2	2
			<- d ->			
			<- n ->			
			11101101	ED		
LD R, A	R<-A	● ● ● ● ● ●	01000111	47	2	2
			<- d ->			
			<- n ->			
			11101101	ED		
LD A, [BC]	A<-[BC]	● ● ● ● ● ●	01001111	4F	1	3
			<- d ->			
			<- n ->			
LD A, [DE]	A<-[DE]	● ● ● ● ● ●	00001010	0A	1	3
			<- d ->			
			<- n ->			
LD A, [nn]	A<-[nn]	● ● ● ● ● ●	00011010	1A	3	4
			<- d ->			
			<- n ->			
LD [BC], A	[BC]<-A	● ● ● ● ● ●	00111010	3A	1	3
			<- d ->			
			<- n ->			
LD [DE], A	[DE]<-A	● ● ● ● ● ●	00110010	12	1	3
			<- d ->			
			<- n ->			
LD [nn], A	[nn]<-A	● ● ● ● ● ●	00110010	32	3	4
			<- d ->			
			<- n ->			

*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	Opcode		B	CL
			76543210	Hex		
LD ss, nn	ss<-nn	● ● ● ● ● ●	00ss0001		3	3
			<- n ->			
			<- n ->			
LD IX, nn	IX<-nn	● ● ● ● ● ●	11011101	DD	4	4
			<- d ->			
			00100001	21		
			<- n ->			
LD IY, nn	IY<-nn	● ● ● ● ● ●	<- n ->		4	4
			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	Opcode		B	CL
			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]	● ● ● ● ● ●	11100011	E3	1	5
	H<->[SP+1]					
EX [SP], IX	IXL<->[SP]	● ● ● ● ● ●	11011101	DD	2	6
	IXH<->[SP+1]		11100011	E3		
EX [SP], IY	IYL<->[SP]	● ● ● ● ● ●	11111101	FD	2	6
	IYH<->[SP+1]		11100011	E3		
EXX	BC<->BC'	● ● ● ● ● ●	11011001	D9	1	1
	DE<->DE'					
	HL<->HL'					

PUSH / POP GROUP

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

	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		
	HL<-HL+1					
	BC<-BC-1					
LDD	[DE]<-[HL]	● ● 0 ⇔ 0 ●	11101101	ED	2	5
	DE<-DE-1	*1	10101000	A8		
	HL<-HL-1					
	BC<-BC-1					
LDIR	repeat;	● ● 0 0 0 ●	11101101	ED	2	6
	[DE]<-[HL]		10110000	B0		
	DE<-DE+1					
	HL<-HL+1					
LDDR	BC<-BC-1					
	until BC=0					
	repeat;	● ● 0 0 0 ●	11101101	ED	2	6
	[DE]<-[HL]		10111000	B8		
DE<-DE-1						
HL<-HL-1						
	BC<-BC-1					
	until BC=0					

*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		
	BC<-BC-1					
	until BC=0 or A=[HL]					
CPDR	repeat;A-[HL]	⇔ ⇔ ⇔ ⇔ 1 ●	11101101	ED	2	6
	HL<-HL-1	*2 *1	10111001	B9		
	BC<-BC-1					
	until BC=0 or A=[HL]					

*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	C L
			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
ADD A, [HL]	A<-A+[HL]	⇔⇔⇔V 0⇔	<- n -> 10000110	86	1	2
ADD A, [IX+d]	A<-A+[IX+d]	⇔⇔⇔V 0⇔	11011101 10000110 <- d ->	DD 86	3	5
ADD A, [IY+d]	A<-A+[IY+d]	⇔⇔⇔V 0⇔	11111101 10000110 <- d ->	FD 86	3	5
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
ADC A, [HL]	A<-A+[HL]+C	⇔⇔⇔V 0⇔	<- n -> 10001110	8E	1	2
ADC A, [IX+d]	A<-A+[IX+d]+C	⇔⇔⇔V 0⇔	11011101 10001110 <- d ->	DD 8E	3	5
ADC A, [IY+d]	A<-A+[IY+d]+C	⇔⇔⇔V 0⇔	11111101 10001110 <- d ->	FD 8E	3	5
ADC HL, ss	HL<-HL+ss+C	⇔⇔ ? V 0⇔	11101101 01ss1010	ED	2	2
ADD HL, ss	HL<-HL+ss	● ● ? ● 0⇔	00ss1001		1	1
ADD IX, pp	IX<-IX+pp	● ● ? ● 0⇔	11011101 00pp1001	DD	2	2
ADD IY, rr	IY<-IY+rr	● ● ? ● 0⇔	11111101 00 rr1001	FD	2	2
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 00110100 <- d ->	DD 34	3	7
INC [IY+d]	[IY+d]<-[IY+d]+1	⇔⇔⇔V 0●	11111101 00110100 <- d ->	FD 34	3	7
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	CL
			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	CL
			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 ->			
SBC A, [HL]	A<-A-[HL]-C	⇔ ⇔ ⇔ V 1 ⇔	10011110	9E	1	2
SBC A, [IX+d]	A<-A-[IX+d]-C	⇔ ⇔ ⇔ V 1 ⇔	11011101	DD	3	5
			10011110	9E		
			<- d ->			
SBC A, [IY+d]	A<-A-[IY+d]-C	⇔ ⇔ ⇔ V 1 ⇔	11111101	FD	3	5
			10011110	9E		
			<- d ->			
SBC HL, ss	HL<-HL-ss-C	⇔ ⇔ ? V 1 ⇔	11101101	ED	2	2
			01ss0010			
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	O p c o d e		B	CL
			76543210	Hex		
DEC ss	ss<-ss-1	● ● ● ● ● ●	00ss1011		1	1
DEC IX	IX<-IX-1	● ● ● ● ● ●	11011101	DD	2	2
DEC IY	IY<-IY-1	● ● ● ● ● ●	00101011	2B	2	2
			11111101	FD		
			00101011	2B		

COMPARE 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		
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	O p c o d e		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 ∨ r	⇔ ⇔ 0 P 0 0	10110 r		1	1
OR n	A<-A ∨ n	⇔ ⇔ 0 P 0 0	11110110	F6	2	2
			<- n ->			
OR [HL]	A<-A ∨ [HL]	⇔ ⇔ 0 P 0 0	10110110	B6	1	2
OR [IX+d]	A<-A ∨ [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 - > 01 b 110	DD CB	4	5
BIT b, [IY+d]	Z←-NOT [IY+d]{b}	? ⇔ 1 ? 0 ●	11111101 11001011 < - d - > 01 b 110	FD CB	4	5
SET b, r	r{b}←-1	● ● ● ● ● ●	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 - > 11 b 110	DD CB	4	7
SET b, [IY+d]	[IY+d]{b}←-1	● ● ● ● ● ●	11111101 11001011 < - d - > 11 b 110	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 ⇔	00000111	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	O p c o d e		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 - >			
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 - >			
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		
RL [IX+d]	tmp←-C	⇔ ⇔ 0 P 0 ⇔	11011101	DD	4	7
	C←-[IX+d]{7}		11001011	CB		
	[IX+d]←-[IX+d]*2		< - d - >			
RL [IY+d]	[IX+d]{0}←-tmp		00010110	16	4	7
	tmp←-C	⇔ ⇔ 0 P 0 ⇔	11111101	FD		
	C←-[IY+d]{7}		11001011	CB		
RR r	[IY+d]←-[IY+d]*2		< - d - >		4	7
	[IY+d]{0}←-tmp		00010110	16		
	tmp←-C;C←-r{0}	⇔ ⇔ 0 P 0 ⇔	11001011	CB		
RR [HL]	r←-r/2;r{7}←-tmp	⇔ ⇔ 0 P 0 ⇔	00011 r		2	2
	tmp←-C;C←-r{0}	⇔ ⇔ 0 P 0 ⇔	11001011	CB		
RR [IX+d]	[HL]←-[HL]/2		00011110	1E	2	5
	[HL]{7}←-tmp					
	tmp←-C	⇔ ⇔ 0 P 0 ⇔	11011101	DD		
RR [IY+d]	C←-[IX+d]{0}	⇔ ⇔ 0 P 0 ⇔	11001011	CB	4	7
	[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} A{0..3} <-[HL]{4..7} [HL]{4..7} <-[HL]{0..3} [HL]{0..3}<-tmp	⇔ ⇔ 0 P 0 ⇔	11101101 01101111	ED 6F	2	5
RDD	tmp<-A{0..3} A{0..3} <-[HL]{0..3} [HL]{0..3} <-[HL]{4..7} [HL]{4..7}<-tmp	⇔ ⇔ 0 P 0 ⇔	11101101 01100111	ED 67	2	5

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} r<-r*2	⇔ ⇔ 0 P 0 ⇔	11001011 00100 r	CB	2	2
SLA [HL]	C<-[HL]{7} [HL]<-[HL]*2	⇔ ⇔ 0 P 0 ⇔	11001011 00100110	CB 26	2	5
SLA [IX+d]	C<-[IX+d]{7} [IX+d]<-[IX+d]*2	⇔ ⇔ 0 P 0 ⇔	11011101 11001011 <- d ->	DD CB	4	7
SLA [IY+d]	C<-[IY+d]{7} [IY+d]<-[IY+d]*2	⇔ ⇔ 0 P 0 ⇔	00100110 11111101 11001011 <- d ->	26 FD CB	4	7
SRL r	C<-r{0} r<-r/2	⇔ ⇔ 0 P 0 ⇔	11001011 00111 r	CB	2	2
SRL [HL]	C<-[HL]{0} [HL]<-[HL]/2	⇔ ⇔ 0 P 0 ⇔	11001011 00111110	CB 3E	2	5
SRL [IX+d]	C<-[IX+d]{0} [IX+d]<-[IX+d]/2	⇔ ⇔ 0 P 0 ⇔	11011101 11001011 <- d ->	DD CB	4	7
SRL [IY+d]	C<-[IY+d]{0} [IY+d]<-[IY+d]/2	⇔ ⇔ 0 P 0 ⇔	00111110 11111101 11001011 <- d ->	3E FD CB	4	7
SRA r	tmp<-r{7};C<-r{0} r<-r/2	⇔ ⇔ 0 P 0 ⇔	11001011 00101 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		
SRA [HL]	r{7}<-tmp tmp<-[HL]{7} C<-[HL]{0} [HL]<-[HL]/2 [HL]{7}<-tmp	⇔ ⇔ 0 P 0 ⇔	11001011 00101110	CB 2E	2	5
SRA [IX+d]	tmp<-[IX+d]{7} C<-[IX+d]{0} [IX+d]<-[IX+d]/2 [IX+d]{7}<-tmp	⇔ ⇔ 0 P 0 ⇔	11011101 11001011 < - d - > 00101110	DD CB 2E	4	7
SRA [IY+d]	tmp<-[IY+d]{7} C<-[IY+d]{0} [IY+d]<-[IY+d]/2 [IY+d]{7}<-tmp	⇔ ⇔ 0 P 0 ⇔	11111101 11001011 < - d - > 00101110	FD CB 2E	4	7

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 < - nnl -> < -nnh ->	C3	3	3
JP NZ, nn	if Z=0 PC<-nn	● ● ● ● ● ● ● ●	11000010 < - nnl -> < -nnh ->	C2	3	3
JP Z, nn	if Z=1 PC<-nn	● ● ● ● ● ● ● ●	11001010 < - nnl -> < -nnh ->	CA	3	3
JP NC, nn	if C=0 PC<-nn	● ● ● ● ● ● ● ●	11010010 < - nnl -> < -nnh ->	D2	3	3
JP C, nn	if C=1 PC<-nn	● ● ● ● ● ● ● ●	11011010 < - nnl -> < -nnh ->	DA	3	3
JP PO, nn	if PV=0 PC<-nn	● ● ● ● ● ● ● ●	11100010 < - nnl -> < -nnh ->	E2	3	3
JP PE, nn	if PV=1 PC<-nn	● ● ● ● ● ● ● ●	11101010 < - nnl -> < -nnh ->	EA	3	3
JP P, nn	if S=0 PC<-nn	● ● ● ● ● ● ● ●	11110010 < - nnl -> < -nnh ->	F2	3	3
JP M, nn	if S=1 PC<-nn	● ● ● ● ● ● ● ●	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 ->			
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 - >			
CALL P, nn	SP<-SP-2					
	PC<-nn					
CALL M, nn	if PV=1	● ● ● ● ● ●	11101100	EC	3	3
	[SP-2]<-PCL		< - nml - >			5
RET	[SP-1]<-PCH		< - nnh - >			
	SP<-SP-2					
RET NZ	PC<-nn					
	if S=0	● ● ● ● ● ●	111100100	F4	3	3
RET Z	[SP-2]<-PCL		< - nml - >			5
	[SP-1]<-PCH		< - nnh - >			
RET NC	SP<-SP-2					
	PC<-nn					
RET C	if S=1	● ● ● ● ● ●	11111100	FC	3	3
	[SP-2]<-PCL		< - nml - >			5
RET PO	[SP-1]<-PCH		< - nnh - >			
	SP<-SP-2					
RET PE	PC<-nn					
	PCL<-[SP]	● ● ● ● ● ●	11001001	C9	1	3
RET NZ	PCH<-[SP+1]					
	SP<-SP+2					
RET Z	if Z=0	● ● ● ● ● ●	11000000	C0	1	2
	PCL<-[SP]					4
RET NC	PCH<-[SP+1]					
	SP<-SP+2					
RET C	if Z=1	● ● ● ● ● ●	11001000	C8	1	2
	PCL<-[SP]					4
RET PO	PCH<-[SP+1]					
	SP<-SP+2					
RET PE	if C=0	● ● ● ● ● ●	11010000	D0	1	2
	PCL<-[SP]					4
RET NZ	PCH<-[SP+1]					
	SP<-SP+2					
RET Z	if C=1	● ● ● ● ● ●	11011000	D8	1	2
	PCL<-[SP]					4
RET NC	PCH<-[SP+1]					
	SP<-SP+2					
RET C	if PV=0	● ● ● ● ● ●	11100000	E0	1	2
	PCL<-[SP]					4
RET PO	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	C L
			76543210	Hex		
RET P	PCH<-[SP+1] SP<-SP+2 if S=0	● ● ● ● ● ●	11110000	F0	1	2 4
RET M	PCL<-[SP] PCH<-[SP+1] SP<-SP+2 if S=1	● ● ● ● ● ●	11111000	F8	1	2 4
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	C L
			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 HL<-HL+1	? ⇔ ? ? 1 ● *1	11101101 10100010	ED A2	2	5
IND	[HL]<-[C] B<-B-1 HL<-HL-1	? ⇔ ? ? 1 ● *1	11101101 10101010	ED AA	2	5
INIR	repeat;[HL]<-[C] B<-B-1 HL<-HL+1 until B=0	? 1 ? ? 1 ●	11101101 10110010	ED B2	2	6 6
INDR	repeat;[HL]<-[C] B<-B-1 HL<-HL-1 until B=0	? 1 ? ? 1 ●	11101101 10111010	ED BA	2	6 6
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 HL<-HL+1	*1	10100011	A3		
OUTD	[C]<-[HL]	? ⇔ ? ? ? 1 ●	11101101	ED	2	5
	B<-B-1 HL<-HL-1	*1	10101011	AB		
OTIR	repeat;[C]<-[HL]	? 1 ? ? 1 ●	11101101	ED	2	7
	B<-B-1 HL<-HL+1 until B=0		10110011	B3		7
OTDR	repeat;[C]<-[HL]	? 1 ? ? 1 ●	11101101	ED	2	7
	B<-B-1 HL<-HL-1 until B=0		10111011	BB		7

*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