

お客様各位

カタログ等資料中の旧社名の扱いについて

2010 年 4 月 1 日を以って NEC エレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願い申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010 年 4 月 1 日
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

32ビット・バス対応 CSMA/CD コントローラ

μ PD72932は、32ビットおよび16ビット・バス・システムに対応した CSMA/CD (Carrier Sense Multiple Access with Collision Detection) コントローラです。

μ PD72932のシステム・インタフェースは、2サイクル DMA モード動作において標準的な CPU のバス・バンド幅に対し3%以下しかバスを占有しません。バス・モードの切り替えによって、ほとんどの標準マイクロプロセッサとインタフェースすることができます。リンク・リスト・バッファ管理システムにより、パーソナル・コンピュータのアダプタ・ボードから高速、高性能マザー・ボードに至るまで、さまざまなアプリケーションに柔軟に対応します。さらに、IEEE 802.3に準拠したエンコーダ/デコーダ/PLL (ENDEC) を集積しており、 μ PC8392 (10BASE5/2 トランシーバ) と組み合わせ、2チップでイーサネット™にも対応します。

受信バッファ管理は、(1) パケット・データをバッファする受信バッファ・エリア、(2) ステータス情報を示す受信ディスクリプタ・エリア、(3) 追加メモリを割り当てる受信リソース・エリアの3つのメモリ・エリアで行います。送信バッファ管理は、(1) ステータスおよび送信情報を記録する送信ディスクリプタ・エリア、(2) パケット・データをバッファする送信バッファ・エリアの2つのメモリ・エリアで行います。システムは送信データの送信リストを生成して、1回の送信コマンドで複数のパケットを送信することができます。送信パケット・データはバイト境界にストアすることも、いくつかに分割された非連続なロケーションにストアすることもできます。

特 徴

- 独立したそれぞれ32ビットのアドレス/データ・バス
- 中断可能な高速2サイクル DMA
- 独立した32バイト送信/受信 FIFO
- ビッグ/リトル・エンディアン・フォーマットをサポート
- IEEE 802.3 ENDEC 内蔵
- 16個までの物理/マルチキャスト・アドレス、あるいはそのいずれかを組み合わせたアドレス・フィルタリング
- 32ビット汎用タイマ
- 全二重ループバック診断
- 802.3レイヤ管理規格をサポートする完全なネットワーク管理機能
- ブリッジおよびリピータ・アプリケーションのサポート

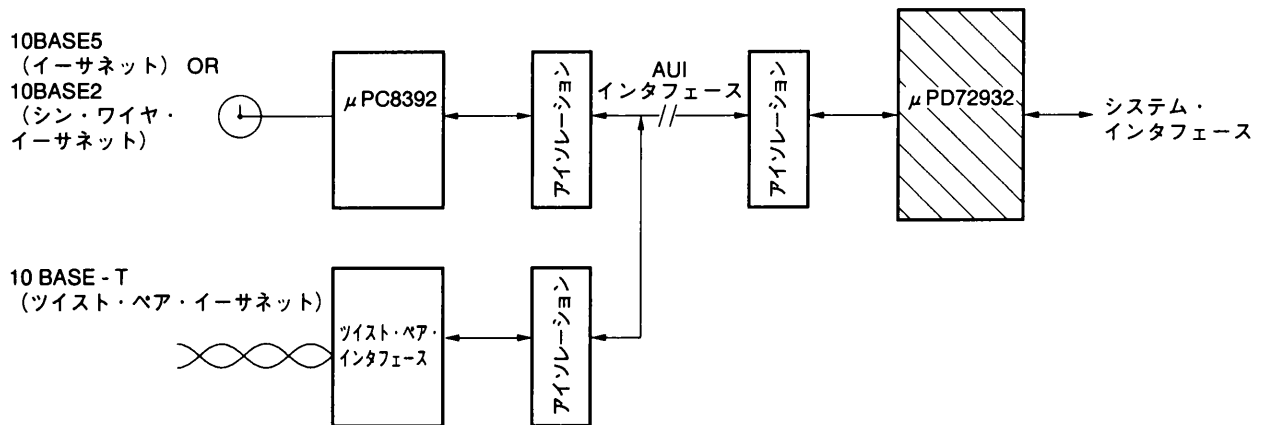
本資料の内容は、後日変更する場合があります。

★ オーダ情報

オーダ名称	パッケージ	動作周波数 (MHz)
μPD72932CVF-20	132ピン・プラスチック QFP	20
μPD72932CVF-25	//	25
μPD72932CVF-33	//	33

システム構成例

IEEE 802.3 イーサネット / シン・イーサネット / 10BASE - Tステーション

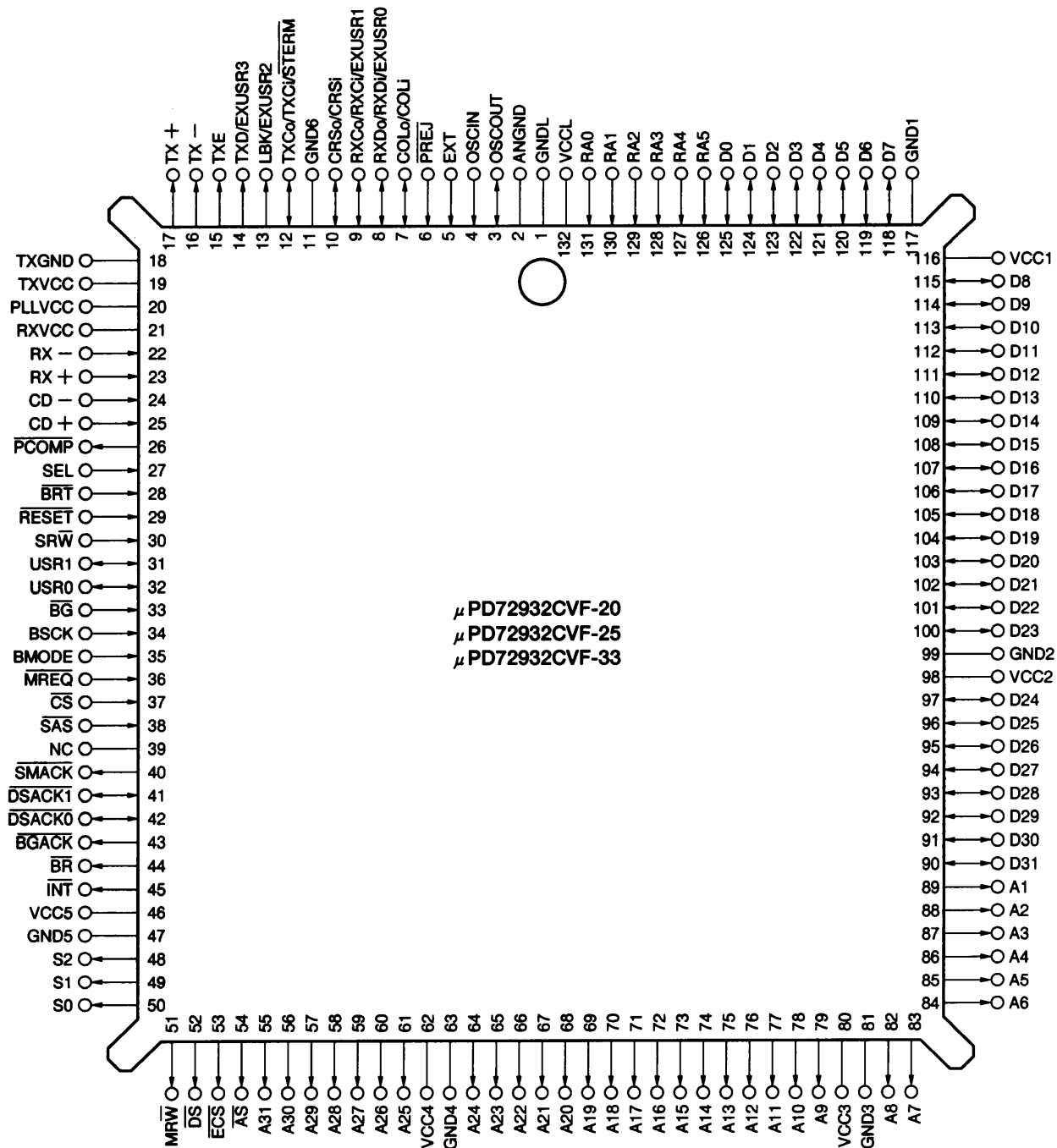


端子接続図 (Top View)

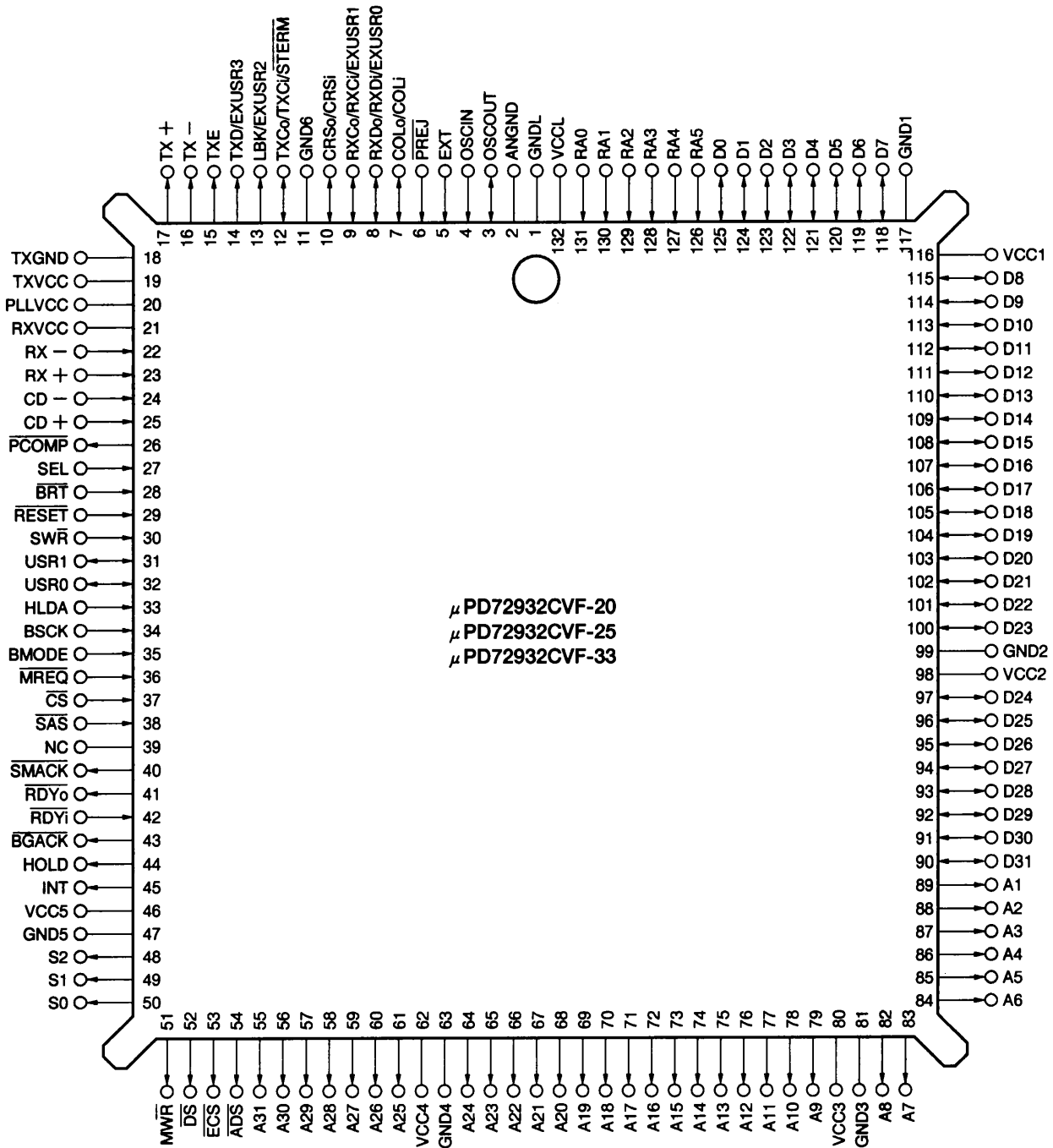


132ピン・プラスチックQFP

(a) BMODE = 1 (モトローラ型バスに対応)

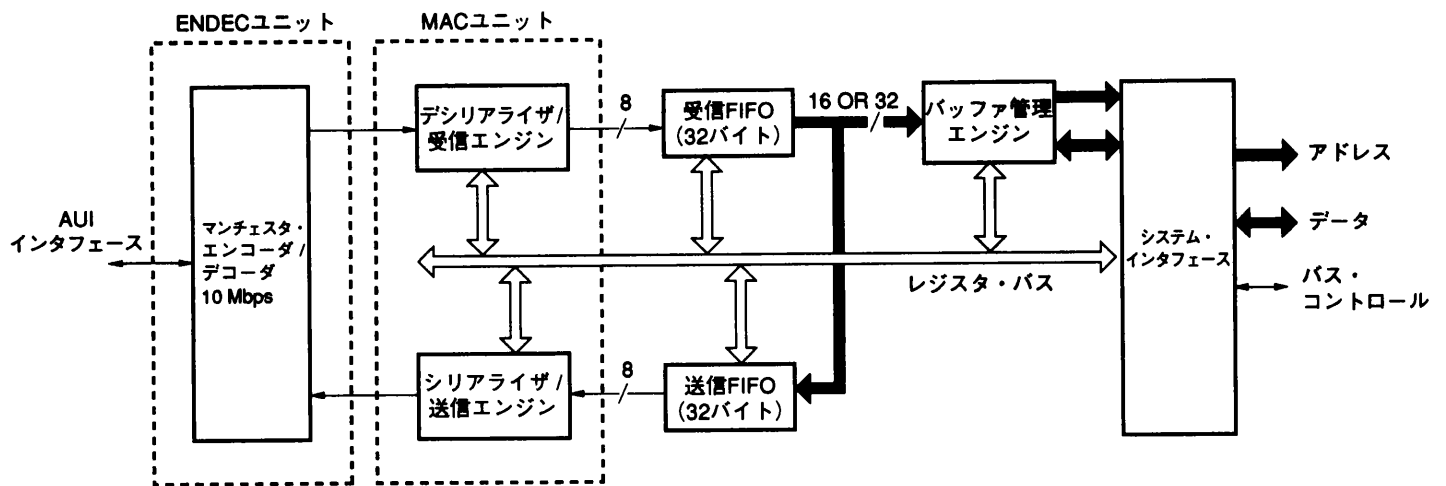


(b) BMODE = 0 (ナショナル セミコンダクター (NSC) / NEC / インテル型バスに対応)



A31-A1	: Address Bus	OSCOUNT	: Oscillator Output
$\overline{AS}$, $\overline{ADS}$	: Address Strobe	$\overline{PCOMP}$	: Packet Compression
$\overline{BG}$	: Bus Grant	$\overline{PREJ}$	: Packet Reject
$\overline{BGACK}$	: Bus Grant Acknowledge	RA5-RA0	: Register Address Bus
BMODE	: Bus Mode	RDYi	: Ready Input
$\overline{BR}$	: Bus Request	RDYo	: Ready Output
$\overline{BRT}$	: Bus Retry	$\overline{RESET}$	: Reset
BSCK	: Bus Clock	RX+, RX-	: Receive
CD+, CD-	: Collision	RXCi	: Receive Clock Input
COLi	: Collision Detect Input	RXCo	: Receive Clock Output
COLo	: Collision Output	RXD _i	: Receive Data Input
CRSi	: Carrier Sense Input	RXD _o	: Receive Data Output
CRSo	: Carrier Sense Output	S2-S0	: Bus Status
$\overline{CS}$	: Chip Select	$\overline{SAS}$	: Slave Address Strobe
D31-D0	: Data Bus	SEL	: Mode Select
$\overline{DS}$	: Data Strobe	$\overline{SMACK}$	: Slave and Memory Acknowledge
$\overline{DSACK1}$, $\overline{DSACK0}$	: Data and Size Acknowledge	SRW	: Slave Read/Write
ECS	: Early Cycle Start	$\overline{STERM}$	: Synchronous Termination
EXT	: External ENDEC Select	SWR	: Slave Read/Write Strobe
EXUSR3-EXUSRO	: Extended User Output	TX+, TX-	: Transmit
GND6-GND1, GNDL	: Ground	TXCi	: Transmit Clock Input
HLDA	: Hold Acknowledge	TXCo	: Transmit Clock Output
HOLD	: Hold Request	TXD	: Transmit Data
$\overline{INT}$, INT	: Interrupt	TXE	: Transmit Enable
LBK	: Loopback	TXGND, ANGND	: Ground for ENDEC
$\overline{MREQ}$	: Memory Request	TXVCC, RXVCC, PLLVCC	: Power Supply for ENDEC
MRW, MWR	: Memory Read/Write Strobe	USR1, USR0	: User Define
NC	: No Connection	VCC5-VCC1, VCCL	: Power Supply
OSCIN	: Oscillator Input		

ブロック図



目 次

1. 端子機能	11
1.1 ネットワーク・インタフェース端子	11
1.2 バス・インタフェース端子	15
1.3 共有メモリ・アクセス端子	19
1.4 ユーザ定義端子	20
1.5 電源およびグランド端子	20
2. 機能説明	21
2.1 IEEE 802.3 ENDEC ユニット	21
2.1.1 ENDEC 動作	23
2.1.2 外部 ENDEC の選択	24
2.2 MAC ユニット	24
2.2.1 MAC 受信部	24
2.2.2 MAC 送信部	26
2.3 データ幅とバイト順序	28
2.4 FIFO および制御回路	29
2.4.1 受信 FIFO	29
2.4.2 送信 FIFO	31
2.5 ステータス・レジスタおよびコンフィギュレーション・レジスタ	31
2.6 バス・インタフェース	31
2.7 ループバックおよび診断	32
2.7.1 ループバック機能	33
2.7.2 ループバック手順	33
2.8 ネットワーク管理機能	34
3. 送信/受信 IEEE 802.3 フレーム・フォーマット	35
3.1 プリアンブルおよびフレーム開始デリミタ (SFD)	35
3.2 デスティネーション・アドレス	35
3.3 ソース・アドレス	36
3.4 レンクス/タイプ・フィールド	36
3.5 データ・フィールド	36
3.6 FCS フィールド	36
3.7 MAC (メディア・アクセス・コントロール) への適合	37
4. バッファ管理	38
4.1 バッファ管理の概要	38
4.2 ディスクリプタ・エリア	38
4.2.1 ディスクリプタのネーミング規則	38
4.2.2 省略形	38

4.2.3	バッファ管理エリアのベース・アドレス	40
4.3	ディスクリプタ・データの配列	40
4.4	受信バッファ管理	40
4.4.1	受信リソース・エリア (RRA)	41
4.4.2	受信バッファ・エリア (RBA)	42
4.4.3	受信ディスクリプタ・エリア (RDA)	44
4.4.4	受信バッファ管理の初期化	45
4.4.5	受信の開始	50
4.4.6	パケットの終了処理	51
4.4.7	オーバフロー状態	51
4.5	送信バッファ管理	54
4.5.1	送信ディスクリプタ・エリア (TDA)	55
4.5.2	送信バッファ・エリア (TBA)	57
4.5.3	送信準備	57
4.5.4	ダイナミックな TDA ディスクリプタ追加	59
5.	μPD72932レジスタ	60
5.1	CAM ユニット	60
5.1.1	CAM の構成	60
5.1.2	ロード CAM コマンド	61
5.2	ステータス/制御レジスタ	64
5.3	レジスタの説明	66
5.3.1	コマンド・レジスタ	66
5.3.2	データ・コンフィギュレーション・レジスタ	69
5.3.3	受信制御レジスタ	73
5.3.4	送信制御レジスタ	76
5.3.5	割り込みマスク・レジスタ	79
5.3.6	割り込みステータス・レジスタ	81
5.3.7	データ・コンフィギュレーション・レジスタ 2	84
5.3.8	送信レジスタ	86
5.3.9	受信レジスタ	86
5.3.10	CAM レジスタ	88
5.3.11	タリー・カウンタ	90
5.3.12	汎用タイマ	91
5.3.13	シリコン・リビジョン・レジスタ (SRR)	91
5.4	レジスタの初期化について	92
6.	バス・インタフェース	97
6.1	端子構成	97
6.2	システム構成	97
6.3	バス・オペレーション	100
6.3.1	バスの獲得	100
6.3.2	ブロック転送	102

6.3.3	バス・ステータス	...	102
6.3.4	バス・モードの互換性	...	104
6.3.5	マスタ・モードのバス・サイクル	...	105
6.3.6	バスの特殊実行例(バス・リトライ)	...	115
6.3.7	スレーブ・モードのバス・サイクル	...	116
6.3.8	オンチップ・メモリ・アービタ	...	121
6.3.9	チップ・リセット	...	122
7.	ネットワークへのインタフェース	...	124
7.1	内部/外部 ENDEC	...	124
7.2	マンチェスタ・エンコーダおよび差動出力ドライバ	...	126
7.3	マンチェスタ・デコーダ	...	126
7.4	コリジョン・トランスレータ	...	127
7.5	電源の考慮事項	...	127
8.	BバージョンからCバージョンへの変更点	...	128
8.1	オーダ名称およびドキュメントの変更	...	128
8.2	動作の改善	...	128
8.3	タイミングの改善	...	129
8.3.1	BMODE=0 (NSC/NEC/インテル・モード) のタイミング修正	...	129
8.3.2	BMODE=1 (モトローラ・モード) のタイミング修正	...	132
8.4	データ・シートの記述修正	...	136
9.	電気的特性	...	137
10.	外形図	...	176
11.	半田付け推奨条件	...	177
付録 A.	使用上の注意事項	...	178
A.1	プログラマブル割り込み (Bバージョン, Cバージョンともに適用)	...	178
A.2	RDA オーバフロー (Bバージョン, Cバージョンともに適用)	...	178
A.3	バス・グラント後の $\overline{DSACK1}$, $\overline{DSACK0}$ の状態 (Bバージョン, Cバージョンともに適用)	...	179
A.4	PLL $\overline{VCC}$ 端子についての考慮 (Bバージョン, Cバージョンともに適用)	...	179
A.5	レディ出力 ($\overline{RDY0}$) の状態 (Bバージョン, Cバージョンともに適用)	...	180
A.6	ラッチト・レディ・モード (Bバージョン, Cバージョンともに適用)	...	180
A.7	バイト・カウンタのミスマッチ (BCM) ビットの状態 (Bバージョン, Cバージョンともに適用)	...	180
A.8	$\overline{SAS}$ ハイ・セットアップ時間の規格 (Bバージョン, Cバージョンともに適用)	...	180
A.9	ループバック・モード時の PMB ビットの状態 (Bバージョン, Cバージョンともに適用)	...	180
A.10	CMOS 負荷テストの削除 (Bバージョン, Cバージョンともに適用)	...	180
A.11	RDA の不正データ書き込み (Bバージョンにのみ適用)	...	181
A.12	TXP のスタック (Bバージョンにのみ適用)	...	181

A.13	RXEN および RXDIS のセット, リセット (B バージョン, C バージョンともに適用) ...	181
A.14	最低バス・クロック速度 (B バージョン, C バージョンともに適用) ...	181
A.15	PCOMP タイミング (B バージョン, C バージョンともに適用) ...	181
A.16	VCCL 端子, GNDL 端子 (B バージョン, C バージョンともに適用) ...	182
A.17	DSACK1 および DSACK0 (B バージョン, C バージョンともに適用) ...	182
A.18	ホスト CPU による μPD72932DMA 動作の強制中断 (B バージョン, C バージョンともに適用) ...	182
A.19	RBAE の処理 (B バージョン, C バージョンともに適用) ...	182
A.20	AC 特性値 (C バージョンにのみ適用) ...	184
A.21	1 つの RBA に複数のバケットを格納する場合 (C バージョンにのみ適用) ...	184
付録 B.	B バージョンでの問題 ...	185
B.1	RDA の不正データ書き込み ...	185
B.2	TXP のスタック ...	186
付録 C.	C バージョンへの置き換えに関する Q & A ...	188

1. 端子機能

1.1 から 1.5 に端子機能一覧を示します。ドライバ・タイプの記号は次の意味です。

記 号	意 味
記号なし（入力端子）	TTL コンパチブル。
ECL	AUI にインタフェースするための ECL 信号レベルのドライバ。
TP	トータム・ボール・ドライバ。ハイまたはロウ・レベルにドライブされる。ドライブ・レベルは CMOS コンパチブル。
TRI	3 ステート・ドライバ。ハイまたはロウ・レベルにドライブされるか、3 ステート状態になる。ドライブ・レベルは CMOS コンパチブル。
OC	オープン・コレクタ・タイプのドライバ。インアクティブのときは 3 ステート状態になり、アクティブのときはロウ・レベルにドライブされる。端子によっては入力端子としても使用する。

1.1 ネットワーク・インタフェース端子 (1/4)

端子名	ドライバ・タイプ	I/O	機 能
EXT		I	外部 ENDEC セレクト この端子を V_{CC} に接続すると (EXT=1)、内部 ENDEC をディスエーブルし、外部 ENDEC を使用できるようにします。この端子をグランドに接続すると (EXT=0)、内部 ENDEC をイネーブルします。この端子は必ず V_{CC} またはグランドのいずれかに接続してください。 CRS_o/CRS_i , COL_o/COL_i , RXD_o/RXD_i , RXC_o/RXC_i および TXC_o/TXC_i 端子は、EXT 端子の入力レベルによって機能が変わります。それぞれの端子の機能説明を参照してください。EXT=0 のときは最初に説明する機能となり、EXT=1 のときは 2 番目に説明する機能となります。
CD+		I	コリジョン+ トランシーバからの正の差動コリジョン入力です。この端子は外部 ENDEC 使用時 (EXT=1) には何も接続しないでください。
CD-		I	コリジョン- トランシーバからの負の差動コリジョン入力です。この端子は外部 ENDEC 使用時 (EXT=1) には何も接続しないでください。
RX+		I	受信+ トランシーバからの正の受信データ入力です。この端子は外部 ENDEC 使用時 (EXT=1) には何も接続しないでください。
RX-		I	受信- トランシーバからの負の受信データ入力です。この端子は外部 ENDEC 使用時 (EXT=1) には何も接続しないでください。
TX+	ECL	O	送信+ トランシーバへの正の送信データ出力です。この端子は外部 ENDEC 使用時 (EXT=1) には何も接続しないでください。
TX-	ECL	O	送信- トランシーバへの負の送信データ出力です。この端子は外部 ENDEC 使用時 (EXT=1) には何も接続しないでください。

1.1 ネットワーク・インタフェース端子 (2/4)

端子名	ドライバ・ タイプ	I/O	機 能
CRSo CRSi	TP	○ I	内部 ENDEC (EXT=0) からのキャリア検知出力 (CRSo) EXT=0 のとき, CRSo 信号は ENDEC と MAC ユニットとの間で内部接続されます。RX+, RX- に入力される受信データの最初の有効なハイ・レベルからロウ・レベルへの変化によって, CRSo 信号はハイ・レベルになります。データの最後のビットから 1.5 ビット・タイムの間はハイ・レベルのままになります。この信号は内部で使用しますが, ユーザへの出力としても供給します。 外部 ENDEC (EXT=1) からのキャリア検知入力 (CRSi) CRSi 端子には, 外部 ENDEC からのキャリア検知信号を入力します。外部 ENDEC が受信入力に有効なデータを検出したとき, ハイ・レベルを入力してください。
COLo COLi	TP	○ I	内部 ENDEC (EXT=0) からのコリジョン出力 (COLo) EXT=0 のとき, COLo 信号は ENDEC と MAC ユニットとの間で内部接続されます。内部 ENDEC がトランシーバから約 10 MHz のコリジョン信号を検出すると, COLo 信号はハイ・レベルになります。この信号は内部で使用しますが, ユーザへの出力としても供給します。 外部 ENDEC (EXT=1) からのコリジョン検出入力 (COLi) 外部 ENDEC がコリジョンを検出したとき, COLi 端子にハイ・レベルを入力してください。COLi 端子はフレーム開始デリミタからパケットの終わりまでの送信中モニタされます。送信の終わりでは, この信号は CD ハートビートのモニタとしても使われます。
RXDo/ RXDi/ EXUSRO	TP TRI	○ I ○, 3 ステート	この端子は, DCR を設定するまで 3 ステートに保たれます (詳細については 5.3.2 EXBUS 参照)。 内部 ENDEC (EXT=0) からの受信データ出力 (RXDo) NRZ データ出力。EXT=0 のときには, RXDo 信号は ENDEC と MAC ユニット間で内部接続されます。この信号は受信クロック出力 (RXCo) の立ち上がりエッジでサンプルしています。この信号は内部で使用しますが, ユーザへの出力としても供給します。 外部 ENDEC (EXT=1) からの受信データ入力 (RXDi) 外部 ENDEC がデコードした NRZ データを入力します。このデータは RXCi の立ち上がりエッジでサンプリングします。 拡張ユーザ出力 (EXUSRO) EXBUS がセットされると (5.3.2 参照), この端子はプログラマブル出力になります。μPD72932 がバス・マスタになると, DCR2 にプログラムされた値に従ってドライブします。μPD72932 がバス・マスタのとき以外は 3 ステート状態になります (5.3.7 参照)。

1.1 ネットワーク・インタフェース端子 (3/4)

端子名	ドライバ・タイプ	I/O	機能
RXCo/ RXCi/ EXUSR1	TP TRI	○ I ○, 3 ステート	この端子は、DCR を設定するまで 3 ステート状態になります (詳細については 5.3.2 EXBUS 参照)。 内部 ENDEC (EXT=0) からの受信クロック出力 (RXCo) EXT=0 のときには、RXCo 信号は ENDEC および MAC ユニット間で内部接続されます。この信号は、マンチェスタ・データから分離された受信クロックです。CRSo がロウ・レベルになったあと、受信クロックは 5 クロック分アクティブになります。この信号は内部で使用しますが、ユーザへの出力としても供給します。 外部 ENDEC (EXT=1) からの受信クロック入力 (RXCi) 外部 ENDEC がマンチェスタ・データから分離した受信クロックを入力します。この信号は外部 ENDEC が生成します。 拡張ユーザ出力 (EXUSR1) EBUS がセットされると (5.3.2 参照), この端子はプログラマブル出力になります。μPD72932 がバス・マスタになると、DCR2 にプログラムされた値に従ってドライブします。μPD72932 がバス・マスタのとき以外は 3 ステート状態になります (5.3.7 参照)。
TXD/ EXUSR3	TP TRI	○ ○, 3 ステート	この端子は、DCR を設定するまで 3 ステート状態になります (詳細については 5.3.2 EXBUS 参照)。 送信データ (TXD) MAC ユニットからのシリアル NRZ データ出力で、外部 ENDEC に入力してデコードします。データは TXC の立ち上がりエッジで有効になります。この信号は内部で使用しますが、ユーザへの出力としても供給します。 拡張ユーザ出力 (EXUSR3) EBUS がセットされると (5.3.2 参照), この端子はプログラマブル出力になります。μPD72932 がバス・マスタになると、DCR2 にプログラムされた値に従ってドライブします。μPD72932 がバス・マスタのとき以外は 3 ステート状態になります (5.3.7 参照)。
TXE	TP	○	送信イネーブル この端子は μPD72932 が送信を開始するとハイ・レベルになり、最後のビットを送信するまでハイ・レベルを維持します。この信号は内部で使用しますが、ユーザへの出力としても供給します。
TXCo/ TXCi/ STERM	TRI	○, 3 ステート I I	この端子は、DCR を設定するまで 3 ステート状態になります (詳細については 5.3.2 EXBUS 参照)。 内部 ENDEC (EXT=0) からの送信クロック出力 (TXCo) この 10 MHz 送信クロック出力は、20 MHz の発振器から得ています。EXT=0 のときには、TXCo 信号は ENDEC と MAC ユニット間で内部で接続されます。この信号は内部で使用しますが、ユーザへの出力としても供給します。 送信クロック入力 (TXCi) (EXT=1) この端子には、外部 ENDEC からのクロックを入力します。MAC ユニットのシリアルライザからデータをシフト・アウトするのに使用します。このクロックは 10 MHz です。 同期停止 (Synchronous Termination) (STERM) μPD72932 はバス・マスタのとき、メモリ・サイクルを停止する前にこの端子をサンプルします。この端子は同期してサンプルされ、BMODE=1 の非同期バス・モードの場合に限り使用できます。詳細については 6.3.5 を参照してください。

1.1 ネットワーク・インタフェース端子 (4/4)

端子名	ドライバ・ タイプ	I/O	機 能
LBK/ EXUSR2	TP TRI	○ ○, 3 ステート	この端子は、DCR を設定するまで 3 ステート状態になります (詳細については 5.3.2 EXBUS 参照)。 ループバック (LBK) ENDEC ループバックがプログラムされると、この端子がハイ・レベルになります。この信号は内部で使用しますが、ユーザへの出力としても供給します。 拡張ユーザ出力 (EXUSR2) EXBUS がセットされると (5.3.2 参照)、この端子はプログラマブル出力になります。μPD72932 がバス・マスタになると、DCR2 にプログラムされた値に従ってドライブします。μPD72932 がバス・マスタのとき以外は 3 ステート状態になります (5.3.7 参照)。
PCOMP	TRI	○, 3 ステート	パケット圧縮 この端子は、DCR2 を設定するまで 3 ステート状態になります (詳細については 5.3.7 PCM, PCNM 参照)。DCR2 の設定により、受信パケットのデスティネーション・アドレスが CAM に設定されたアドレスと整合したとき (PCM=1) あるいは整合しないとき (PCNM=1) に PCOMP をロウ・レベルにします。PCOMP をロウ・レベルにするタイミングは到着したパケットの 7 バイト目の第 4 ビットの直後です。この端子はリピータ・インタフェース・コントローラ (μPD72950) の管理バスとともに使用します。μPD72950 はこの信号を使用して、ネットワーク管理のために受信パケットを圧縮 (短縮) し、メモリ使用量を低減することができます。
SEL		I	モード・セレクト (EXT=0) この端子はアイドル時のネットワーク・インタフェース上の絶縁トランス一次側における TX+ と TX- 間の電圧関係を決定するために使用します。SEL=1 のとき、アイドル時には TX+ と TX- は同電圧になります。SEL=0 のとき、アイドル時には TX+ が TX- に対して正になります (図 7-2 参照)。
PRE $\overline{\text{I}}$		I	パケット・リジェクト この信号は受信パケットをリジェクトするために使用します。受信クロック (RXC) の 2 クロック分以上の間ロウ・レベルにすると、μPD72932 は着信パケットをリジェクトします。この端子を用いてパケットをリジェクトする場合、受信の最終ビットまでにロウ・レベルにしてください。
OSCO $\overline{\text{U}}$ T	TP	I, ○	水晶振動子フィードバック出力 この信号は内部 ENDEC ヘックロックを供給するために使用します。水晶振動子をこの端子と OSCIN に接続します。外部クロック発振器または水晶振動子の使用についての詳細は、9. 電気的特性を参照してください。
OSCIN		I	水晶振動子または外部発振器入力 この信号は内部 ENDEC へのクロックを供給するために使用します。水晶振動子をこの端子と OSCOUT に接続しますが、外部クロック発振器を接続することもできます。外部クロック発振器または水晶振動子の使用についての詳細は、9. 電気的特性を参照してください。

1.2 バス・インタフェース端子 (1/5)

端子名	ドライバ・タイプ	I/O	機能
BMODE		I	バス・モード この入力は、マイクロプロセッサと接続する際の μPD72932 のバス・モードを決定します。この端子はバイト順序 (リトルまたはビッグ・エンディアン) を決定し、バス・インタフェース制御信号の動作を制御します。ハイ・レベル (V_{CC} へ接続) でモトローラ・モード (ビッグ・エンディアン) を選択し、ロウ・レベル (グランドへ接続) でナショナル セミコンダクター (NSC)/NEC/インテル・モード (リトル・エンディアン) を選択します。AS/ADS, MRW/MWR, INT/INT, BR/HOLD, BG/HLDA, SRW/SWR, DSACK0/RDY, および DSACK1/RDY。端子は、BMODE 端子の入力レベルによって機能が変わります。それぞれの端子の機能説明を参照してください。BMODE=1 のときは最初に説明する機能となり、BMODE=0 のときは 2 番目に説明する機能となります。6.3.1, 6.3.4 および 6.3.5 を参照してください。
D31-D0	TRI	I, O, 3 ステート	データ・バス これらのバス信号は、システム・バス上でデータを転送するために使用します。μPD72932 がバス・マスタのときには、D15-D0 上で 16 ビット・データを転送し、D31-D0 上で 32 ビット・データを転送します。μPD72932 をスレーブとしてアクセスするときは、D15-D0 を使用してレジスタ・データを転送します。μPD72932 が 16 ビット・モードのときは、D31-D16 は 3 ステートになります。μPD72932 が 32 ビット・モードでスレーブ・サイクルを実行しているときは、D31-D16 をドライブします。
A31-A1	TRI	O, 3 ステート	アドレス・バス μPD72932 はこれらの信号を使用して、μPD72932 がバスを獲得したあと DMA アドレスを出力します。μPD72932 はデータをワードの境界にそろえるので、31 本のアドレス・ラインしか必要ありません (A0 は必要ありません)。
RA5-RA0		I	レジスタ・アドレス・バス これらの信号は、μPD72932 の内部レジスタをアクセスするために使用します。μPD72932 をアクセスする際、システムはこれらのラインにより必要な μPD72932 レジスタを選択します。
AS ADS	TRI TRI	O, 3 ステート O, 3 ステート	アドレス・ストロープ (AS) BMODE=1 のとき、立ち下がりがエッジでステータスおよびアドレスが有効であることを示します。立ち上がりエッジは、メモリ・サイクルの終了を示します。 アドレス・ストロープ (ADS) BMODE=0 のとき、立ち上がりエッジでステータスおよびアドレスが有効であることを示します。
MRW MWR	TRI TRI	O, 3 ステート O, 3 ステート	μPD72932 がバスを獲得すると、この信号はデータの転送方向を示します。 メモリ・リード/ライト・ストロープ (MRW) BMODE=1 のとき、この信号はリード・サイクル中にはハイ・レベルになり、ライト・サイクル中にはロウ・レベルになります。 メモリ・リード/ライト・ストロープ (MWR) BMODE=0 のとき、この信号はリード・サイクル中にはロウ・レベルになり、ライト・サイクル中にはハイ・レベルになります。

1.2 バス・インタフェース端子 (2/5)

端子名	ドライバ・ タイプ	I/O	機 能
$\overline{\text{INT}}$ INT	OC TP	○ ○, 3 ステート	割り込みステータス・レジスタに示される割り込み要因 (IMR でイネーブルに設定されたもの) が 1 つ以上ペンディングされていることを示します。割り込みマスク・レジスタでディスエーブルされている割り込み要因によってこの信号がアクティブになることはありません。 割り込み ($\overline{\text{INT}}$) BMODE=1 のとき、この信号はアクティブ・ロウになります。 割り込み (INT) BMODE=0 のとき、この信号はアクティブ・ハイになります。
$\overline{\text{RESET}}$		I	リセット この信号は μPD72932 のハードウェア・リセットに使用します。ロウ・レベルにすると、一定時間経過後に μPD72932 はリセット・ステートに入ります。リセット・ステートに入るまでの時間は、バス・クロック (BSCK) 周期が送信クロック周期より大きい場合には 10 バス・クロック (BSCK) 後、その他の場合には 10 送信クロック後になります。
S2-S0	TP	○	バス・ステータス これら 3 つの信号は、μPD72932 のバス動作の現在のステータスを表します。ステータスの定義については、6.3.3 を参照してください。
BSCK		I	バス・クロック この入力クロックは μPD72932 の DMA エンジンのタイミングを生成します。
$\overline{\text{BR}}$ HOLD	OC TP	○, 3 ステート ○	バス要求 ($\overline{\text{BR}}$) BMODE=1 のとき、μPD72932 はバスへのアクセスを獲得しようとするときにこの端子をロウ・レベルにします。インアクティブのときにはこの信号は 3 ステートになります。 ホールド・リクエスト (HOLD) BMODE=0 のとき、μPD72932 はバスを使用しようとするときにこの端子をハイ・レベルにし、インアクティブのときはロウ・レベルにします。
$\overline{\text{BG}}$ HLDA		I I	バス許可 ($\overline{\text{BG}}$) BMODE=1 のとき、この信号はシステムからのバス許可信号になります。システムはこの端子をロウ・レベルにして、バスの制御権を許可したことを示します。 ホールド・アクノリッジ (HLDA) BMODE=0 のとき、この信号はシステムが μPD72932 にバスを付与したことを通知します。システムがこの端子をハイ・レベルにすると、μPD72932 はバスの制御権を獲得します。

1.2 バス・インタフェース端子 (3/5)

端子名	ドライバ・タイプ	I/O	機能
$\overline{\text{BGACK}}$	TRI	O, 3 ステート	バス許可アクノリッジ BMODE=1 のとき、μPD72932はバス使用権が獲得できることを確認すると、この端子をロウ・レベルにします。μPD72932が$\overline{\text{BGACK}}$ 応答をセットする前に今回バス獲得要求を出したバス・マスタ（この場合、μPD72932）および前回のバス・マスタは、各制御信号を(1) から(4) で示す状態にする必要があります。 (1) 今回、バス要求を発生した μPD72932の$\overline{\text{BR}}$ 信号は、バス調停プロセスを通してシステムに受信されたことを示している。 (2) 前回のバス・マスタが$\overline{\text{AS}}$ 端子をハイ・レベルにし、バスの使用を終了したことを示している。 (3) 前回のバス・マスタは$\overline{\text{DSACK0}}$ 端子および$\overline{\text{DSACK1}}$ 端子をハイ・レベルにし、バスから切り離されたことを示している。 (4) 前回のバス・マスタは$\overline{\text{BGACK}}$ 端子をハイ・レベルにし、バスを切り離したことを示している。 この端子はモトローラ・モードのときのみ使用します。
$\overline{\text{CS}}$		I	チップ・セレクト^注 システムはこの端子をロウ・レベルにして、μPD72932のレジスタにアクセスします。レジスタの選択はRA5-RA0 端子にアドレスを入力して行います。
$\overline{\text{SAS}}$		I	スレーブ・アドレス・ストローブ システムはこの端子を使用して、RA5-RA0 端子のレジスタ・アドレスをラッチします。 BMODE=1 のとき、アドレスは$\overline{\text{SAS}}$ 端子の立ち下がりエッジでラッチします。BMODE=0 のとき、アドレスは$\overline{\text{SAS}}$ 端子の立ち上がりエッジでラッチします。
$\overline{\text{SRW}}$ $\overline{\text{SWR}}$		I I	システムはこの端子を使用して、μPD72932の内部レジスタに対して読み出しを行うのか書き込みを行うのかを知らせます。 スレーブ・リード/ライト (SRW) BMODE=1 のとき、この端子には読み出し時にはハイ・レベル、書き込み時にはロウ・レベルを入力します。 スレーブ・リード/ライト・ストローブ (SWR) BMODE=0 のとき、この端子には読み出し時にはロウ・レベル、書き込み時にはハイ・レベルを入力します。
$\overline{\text{DS}}$	TRI	O, 3 ステート	データ・ストローブ μPD72932がバス・マスタのとき、この端子を使用します。 リード・サイクル中、μPD72932はこの端子をロウ・レベルにして、スレーブ・デバイスがデータをバスに送出できることを示します。ライト・サイクル中は、この信号は μPD72932が有効なデータをバスに送出していることを示します。

★

注 $\overline{\text{CS}}$ と $\overline{\text{MREQ}}$ を同時にロウ・レベルにしないでください。これらの信号を連続してロウ・レベルにする場合、あとからロウ・レベルにする信号は、先にロウ・レベルにした信号の立ち上がりエッジから2バス・クロック以上の間ハイ・レベルに保持したうえでロウ・レベルにしてください。

1.2 バス・インタフェース端子 (4/5)

端子名	ドライバ・ タイプ	I/O	機 能
DSACK0 RDYi	TRI	I, O, 3 ステート I	データおよびサイズ・アクノリッジ 0 および 1 (DSACK0, DSACK1 : BMODE=1) これらの端子は、ホスト CPU が μPD72932 レジスタをアクセスしているときは、システムへの出力スレーブ・アクノリッジとなり、μPD72932 がバス・マスタのときは入力スレーブ・アクノリッジとなります。 ホスト CPU がレジスタをアクセスしているとき、μPD72932 は DSACK0 端子および DSACK1 端子をロウ・レベルにすることでスレーブ・サイクル終了を知らせます。μPD72932 は 32 ビット・ペリフェラルとして応答しますが、D15-D0 端子にのみデータを送出するので注意してください。D31-D16 端子はドライブしますが、無効になります。 μPD72932 は 32 ビット・バス・マスタのとき、メモリ・サイクルを終了するために DSACK0 端子および DSACK1 端子をサンプリングします。これらの端子がロウ・レベルであれば、μPD72932 はマスタ・サイクルを終了します。 μPD72932 は 16 ビット・バス・マスタのとき、メモリ・サイクルを終了するために DSACK1 端子をサンプリングします。この端子がロウ・レベルであれば、μPD72932 はマスタ・サイクルを終了します。そのとき、DSACK0 端子をロウ・レベルにする必要はありません。この端子を未使用とする場合、抵抗を介してハイ・レベルにプルアップしてください。 データ・コンフィギュレーション・レジスタの SBUS ビットの値により、同期または非同期でサンプルを行います。詳細については 6.3.5 を参照してください。 μPD72932 では、ダイナミック・バス・サイジングを行うことはできません。バス・サイズは、データ・コンフィギュレーション・レジスタに設定したバス幅に固定されます (5.3.2 参照)。 レディ入力 (RDYi : BMODE=0) μPD72932 がバス・マスタのとき、システムがこの信号をハイ・レベルにするとうェイト・ステートを挿入し、ロウ・レベルにするとメモリ・サイクルを終了します。この信号は、SBUS ビットの値により同期または非同期でサンプルされます。詳細については 5.3.2 および 6.3.5 を参照してください。 レディ出力 (RDYo : BMODE=0) レジスタをアクセスしているとき、μPD72932 はこの信号をロウ・レベルにしてスレーブ・サイクルを終了します。この端子は 3 ステートとはなりません。
DSACK1 RDYo	TRI TP	I, O, 3 ステート O	

★

1.2 バス・インタフェース端子 (8/8)

端子名	ドライバ・タイプ	I/O	機能
$\overline{\text{BRT}}$		I	バス・リトライ μPD72932がバス・マスタのとき、システムはバス・エラーを検出するとこの信号をロウ・レベルにして、バス・エラーとなったサイクルを繰り返させます。この端子には2つのモードがあります。 モード1(データ・コンフィギュレーション・レジスタのLBRビットを0にセット) この端子をロウ・レベルにすると、μPD72932はカレント・バス・サイクルを終了し、$\overline{\text{BRT}}$がハイ・レベルになったあとに同じサイクルを繰り返します。 モード2(データ・コンフィギュレーション・レジスタのLBRビットを1にセット) この信号をロウ・レベルにすると、μPD72932はバス・サイクルをモード1と同様にリトライします。ただし、μPD72932は割り込みステータス・レジスタのBRビットをリセットするまでDMA動作を継続しません。
$\overline{\text{ECS}}$	TRI	O, 3ステート	アーリ・サイクルの開始 この出力は、システムに対してメモリ操作をしていることを最も早く知らせる信号です。この信号はT1の立ち上がりエッジでロウ・レベルになり、T1の立ち下がりエッジでハイ・レベルになります。

1.3 共有メモリ・アクセス端子

端子名	ドライバ・タイプ	I/O	機能
$\overline{\text{MREQ}}$		I	メモリ要求^注 システムが共有バッファRAMにアクセスするとき、この信号をロウ・レベルにします。オンチップ・アービタはシステムとμPD72932間のアクセス競合を解決します。詳細は6.3.8を参照してください。
$\overline{\text{SMACK}}$	TP	O	スレーブおよびメモリ・アクノリッジ μPD72932のレジスタまたはバッファ・メモリがアクセス可能なとき、チップ・セレクト($\overline{\text{CS}}$)またはメモリ・リクエスト($\overline{\text{MREQ}}$)に応答して、μPD72932はこの端子をロウ・レベルにします。この端子を使用して、デュアル・バス・システムのバス・ドライバをイネーブルすることができます。

注 $\overline{\text{CS}}$ と $\overline{\text{MREQ}}$ を同時にロウ・レベルにしないでください。これらの信号を連続してロウ・レベルにする場合、あとからロウ・レベルにする信号は、先にロウ・レベルにした信号の立ち上がりエッジから2バス・クロック以上の間ハイ・レベルに保持したうえでロウ・レベルにしてください。

1.4 ユーザ定義端子

端子名	ドライバ・ タイプ	I/O	機 能
USR1, 0	TRI	I, O, 3 ステート	ユーザセット 1, 0 これらの信号は $\overline{\text{RESET}}$ がロウ・レベルのとき入力になります。入力されたデータは $\overline{\text{RESET}}$ の立ち上がりエッジでラッチされ、データ・コンフィギュレーション・レジスタ (DCR) のビット 8 と 9 にそれぞれ直接入力されます。バス・マスタ動作 (HLDA または $\overline{\text{BGACK}}$ がアクティブ) 中には、これらの端子は出力となり、そのレベルはそれぞれ DCR のビット 11 と 12 を通してプログラムできます。USR1, 0 端子は V_{CC} にプルアップするか、グラウンドにプルダウンします。4.7 k Ω のプルアップ抵抗を推奨します。

1.5 電源およびグラウンド端子

端子名	ドライバ・ タイプ	I/O	機 能
VCC5- VCC1 VCCL			電源 μ PD72932 のデジタル部への +5 V 電源です。
TXVCC RXVCC PLLCC			電源 μ PD72932 の ENDEC ユニットへの +5 V 電源です。これらの端子は内部 ENDEC を使用しない場合にも必ず V_{CC} に接続しておいてください。
GND6- GND1 GNDL			グラウンド μ PD72932 のデジタル部へのグラウンド接続端子です。
TXGND ANGND			グラウンド μ PD72932 の ENDEC ユニットへのグラウンド接続端子です。これらの端子は内部 ENDEC を使用しない場合にも必ずグラウンドに接続しておいてください。

2. 機能説明

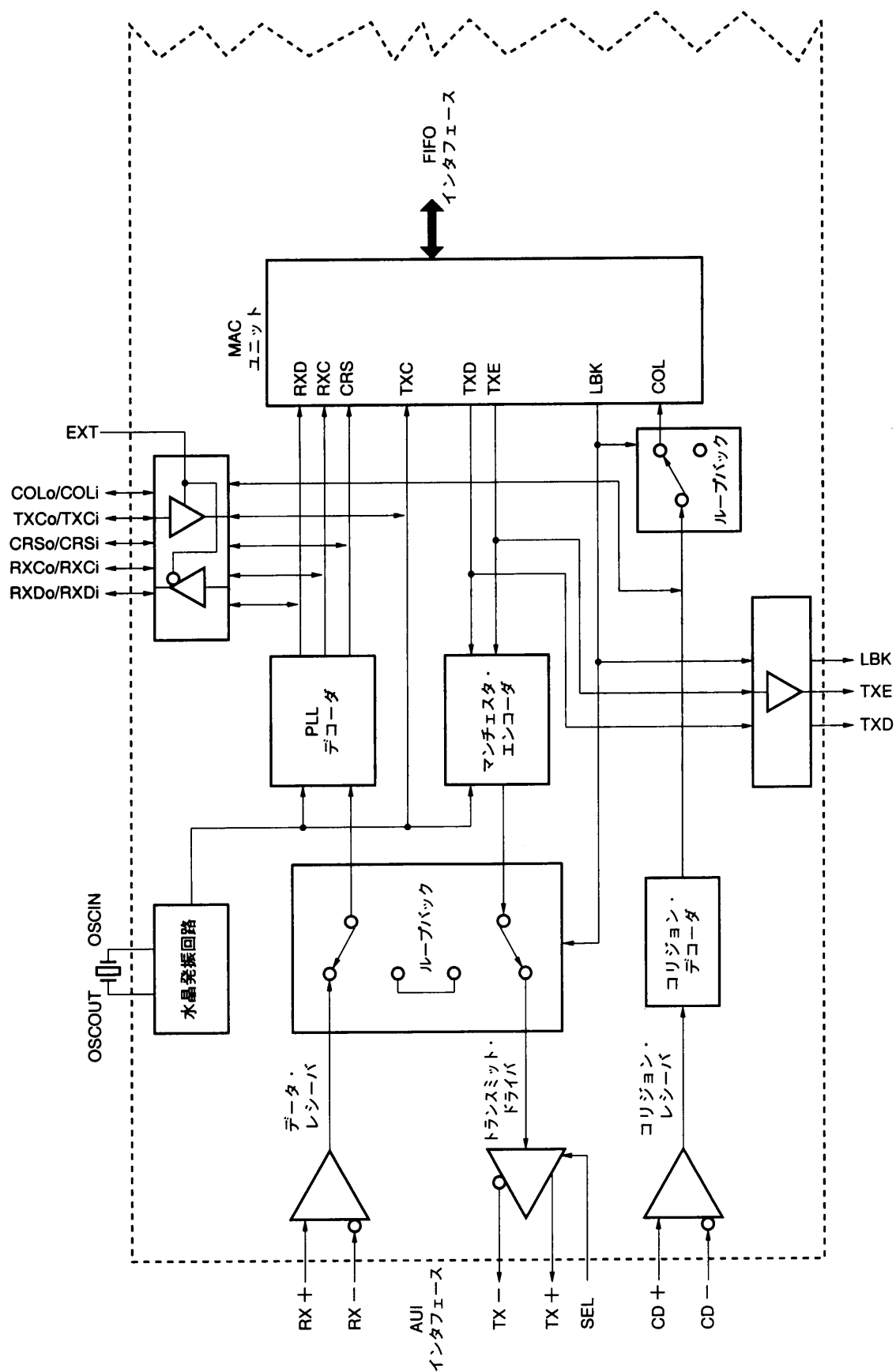
μPD72932はエンコーダ/デコーダ (ENDEC) ユニット, メディア・アクセス・コントロール (MAC) ユニット, 独立した送信および受信 FIFO, システム・バッファ管理のためのエンジン, およびユーザ・プログラマブルなシステム・バス・インタフェース・ユニットから構成されています。μPD72932はシステム・レベルの最大性能を提供するために, 高度にパイプライン化されています。このセクションでは, μPD72932の機能概要について説明します。

2.1 IEEE 802.3 ENDEC ユニット

ENDEC (エンコーダ/デコーダ) ユニットは, AUI (アタッチメント・ユニット・インタフェース) と MAC ユニット間のインタフェースを行います。ENDEC は IEEE 802.3 およびイーサネット/シン・イーサネット型ローカル・エリア・ネットワークの規格に合った, マンチェスタ・データへのエンコーディングおよびデコーディングを行います。送信中, ENDEC ユニットは MAC 部からの non-return-zero (NRZ) データとクロック・パルスをマンチェスタ・データ変換し, その変換データを AUI へ差動出力します。逆に, 受信中には, アナログ PLL がマンチェスタ・データを NRZ フォーマットと受信クロックに分離します。ENDEC ユニットは差動ドライバおよびレシーバ, 水晶発振回路, コリジョン信号トランスレータ, およびループバック診断回路を内蔵したマンチェスタ・エンコーダ/デコーダです。特長は次のとおりです。

- Ethernet I および II, IEEE 802.3 10BASE5 および 10BASE2 とコンパチブル
- 受信クロック・リカバリ機能を備えた 10 Mbps マンチェスタ・エンコーディング/デコーディング回路
- 高精度な外部部品が不要
- システム診断のためのループバック機能
- 送信出力でハーフまたはフル・ステップ・モードを外部設定可能
- 受信入力およびコリジョン入力はノイズを除去するためのスケルチ回路を持つ
- 外部パルス・トランスを経由してトランシーバ (AUI) ケーブルに接続

図2-1 イーサネットENDECのブロック図



2.1.1 ENDEC 動作

ENDEC ユニット (図 2-1) の主要機能は、トランシーバへの差動入出力ペア上のマンチェスタ・エンコード・データと MAC ユニットの non-return-zero (NRZ) シリアル・データとを変換するために必要な、エンコーディングおよびデコーディングを実行することです。また、データのエンコーディングおよびデコーディングに加えて、MAC ユニットに必要なすべての制御信号 (コリジョン検出信号, キャリア検出信号, クロックなど) を供給します。

これらの信号は、オンチップ ENDEC から MAC ユニットに供給されるだけでなく、ユーザにも出力として供給されます。

(1) マンチェスタ・エンコーダおよび差動出力ドライバ

ネットワークへの送信中, ENDEC ユニットは MAC ユニットからの NRZ シリアル・データを、差動マンチェスタ・コードに変換し、10BASE5/2 トランシーバ (μPC8392) への送信ペア上に出力します。この動作を行うために、MAC ユニットからの NRZ ビット・ストリームは、ENDEC ユニットのマンチェスタ・エンコーダ・ブロックに渡されます。ビット・ストリームをエンコードしたあと、送信ドライバを通して、差動送信ペアに出力します。

(2) マンチェスタ・デコーダ

ネットワークからの受信中, 10BASE5/2 トランシーバ (μPC8392) から差動受信したマンチェスタ・コード・データから受信クロックを分離し、データを NRZ シリアル・データに変換します。受信クロックと NRZ シリアル・データは MAC ユニットの受信データとクロック入力に送ります。この動作を行うために、信号はいったん差動レシーバで受信すると、フェーズ・ロック・ループ (PLL) デコーダ・ブロックに渡します。PLL はデータ受信クロックと NRZ シリアル・データ・ストリームを分離して MAC ユニットに送ります。

(3) 特殊信号

マンチェスタ・エンコーディングおよびデコーディング機能に加えて、ENDEC ユニットは MAC ユニットに制御信号とクロックを供給します。データがネットワークから ENDEC の差動受信ペアに送られると、ENDEC はキャリア検出 (CRS) 信号を送って MAC ユニットに知らせます。また、トランシーバがネットワーク上のコリジョンを検出して 10 MHz のコリジョン信号を発生すると、ENDEC はこの信号からコリジョン検出信号 (COL) を生成し、MAC ユニットに知らせます。

ENDEC はさらに、MAC ユニットに受信クロックと送信クロックも供給します。送信クロックは発振回路入力を 2 分周したクロックです。受信クロックは PLL によって入力データから生成されます。

(4) 発振回路

発振回路は、ネットワーク・タイミング用に 10 MHz の送信クロック信号を生成します。発振回路は水晶振動子または外部クロック発振器に接続されます (9. 電気的特性参照)。発振回路の出力は 2 分周され、MAC ユニットのために 10 MHz 送信クロック (TXC) を生成します。発振回路はさらにエンコーディング回路およびデコーディング回路に内部クロック信号を供給します。

(5) ループバック機能

μPD72932 には 3 つのループバック・モードがあります。この 3 つのモードによって、MAC, ENDEC, および外部トランシーバ・レベルでループバック・テストが可能です (詳細は 2.7 を参照してください)。注意しなければならない点は、μPD72932 の送信パケットは、通常の送信時においても外部トランシーバによりループバックされる点です。μPD72932 は送信パケットをモニタするために、このループバックを利用します。送信パケットのモニタに関する詳細は、2.2.1 (1) 受信ステート・マシン (RSM) の説明を参照してください。

2.1.2 外部 ENDEC の選択

μPD72932ではオプションで内部の ENDEC ユニットをディスエーブルし、外部 ENDEC を使用することができます。EXT 端子を V_{CC} に接続する (EXT=1) ことによって、内部 ENDEC をバイパスすることができます。このモードでは、送受信データおよびクロック、コリジョン信号は μPD72932から入出力され、外部 ENDEC の使用が可能です。このモードでの端子機能については、1. 端子機能を参照してください。

2.2 MAC ユニット

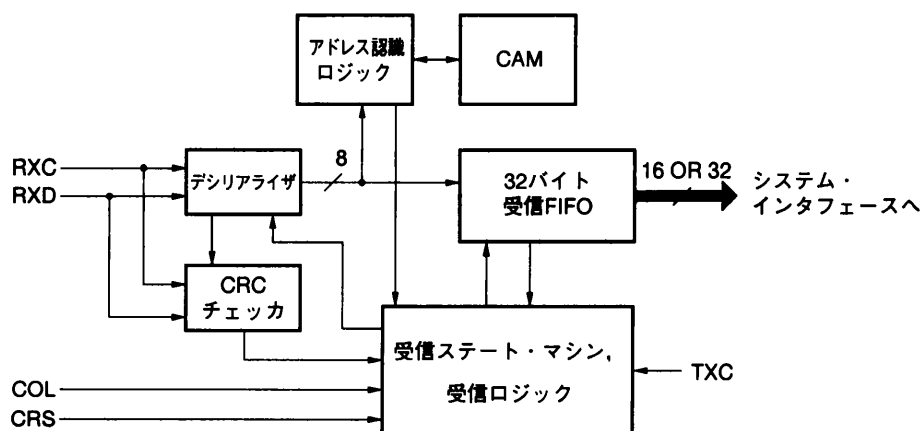
MAC (メディア・アクセス・コントロール) ユニットは、パケット送信および受信のためのメディア・アクセス制御を行います。送信中、MAC ユニットは送信 FIFO からのデータをパラレルからシリアルに変換し、ENDEC ユニットに供給します。受信中は ENDEC ユニットからの受信データをシリアルからパラレルに変換し、受信が有効かどうかを確認するために、フレーム・チェックを行い、データを受信 FIFO に書き込みます。μPD72932の制御レジスタおよびステータス・レジスタが MAC ユニットの動作を管理します。

2.2.1 MAC 受信部

受信部 (図 2-2) は受信、ループバック、および送信中の MAC (メディア・アクセス・コントロール) 受信動作を制御します。受信時、2ビットの SFD (フレーム開始デリミタ) パターンを検出すると、デシリアライザがアクティブになります (3.1 参照)。次に、受信ビットを8ビットごとにパラレル・データ化し、そのデータを32バイト受信 FIFOへ転送します。同時に、アドレス・コンパレータが、デスティネーション・アドレス・フィールドと CAM アドレス・レジスタ (コンテンツ・アドレス・メモリ) に記憶されたアドレスとを比較します。一致すれば、デシリアライザは受信 FIFO へパケットの残りを送ります。キャリア検知信号 (CRS) がインアクティブ状態になると、FIFO の中に残っているパケットを読み出し、システム側に送ります。受信の終わりに、受信部は次のチェックを行い、適切なステータスを受信制御レジスタに示します (5.3.3 参照)。

- フレーム配列エラー
- CRC エラー
- 64オクテット以下のフレーム・エラー (ラント・パケット)

図 2-2 MAC受信部



ループバック動作でも、受信部は通常の受信中と同様に動作します。

送信中においても、受信部は自身が送信したパケットをモニタするためにアクティブになります。サイクリック・リダンダンシ・コード (CRC) チェッカは通常どおり動作し、ソース・アドレス・フィールドと CAM アドレス・エントリを比較します。CRC チェックとアドレス比較の結果は、送信制御レジスタの PMB ビットに示します。送信動作中は受信 FIFO にはデータを書き込みません。

受信部は次に示すブロックで構成されています。

(1) 受信ステート・マシン (RSM)

通常の受信における正しいシーケンスと自身が送信したパケットの受信を保証します。ネットワークがインアクティブ状態のとき、RSM はアイドル状態のまま、ネットワークの状態をモニタします。ネットワークがアクティブになったとき、RSM はデシリアライザのデータを受信 FIFO に転送します。このステート中に、以下のような状態が発生するとパケットを完全に受信できない可能性があります。

- FIFO オーバラン

μPD72932がメモリにデータをバッファする前に、受信 FIFO が満杯になった。

- CAM アドレスが不整合

パケットのデスティネーション・アドレスと CAM のアドレスが不一致のため、パケットを拒否。

- メモリ・リソース・エラー

受信パケットをバッファするためのリソース (バッファ) が存在しない。

- 衝突、およびその他のエラー

ネットワーク上に衝突が発生したか、または CRC エラーなどの他のエラーが発生 (μPD72932がコリジョン・パケット、あるいはエラー・パケットを拒否するように設定された場合)。

これらの状態が起こらなければ、RSM はパケットを処理し受信制御レジスタに適切なステータスをセットします。

μPD72932からパケットが送信されている間は、外部トランシーバは常にパケットを μPD72932にループバックします。μPD72932はこのループバックを利用して、送信中のパケットをモニタします。ループバック・パケットと、送信パケットの CRC およびソース・アドレスを比較します。これらの CRC とソース・アドレスが一致しなければ、エラー・ビットを送信パケットのステータスにセットします。**5.3.4** の送信制御レジスタ (PMB: モニタしたパケットの不良) を参照してください。このモニタ動作の間、トランシーバ・ループバック・モードを選択していないと、受信 FIFO へのデータの書き込みは行いません (**2.7** 参照)。

(2) 受信ロジック

受信ロジックには、受信部の動作を管理する、コマンド・レジスタ、制御レジスタ、およびステータス・レジスタがあります。受信ロジックは受信 FIFO へデータを書き込むための制御信号を生成し、CRC チェッカとデシリアライザからのエラー信号を処理し、内部の $\overline{\text{PREJ}}$ (パケット・リジェクト) 信号をアクティブにしてパケット拒否の条件が整えば RSM にパケットを拒否させて、受信制御レジスタ中に該当するステータスを表示します。

(3) デシリアライザ

シリアル入力データをシリアルからパラレルに変換し、アドレス・コンパレータと受信ロジックに供給します。また CRC チェッカの動作を同期して開始します。受信終了後、CRS がインアクティブとなったあとフレーム配列をチェックします。

(4) アドレス・コンパレータ

アドレス・コンパレータは、デスティネーション・アドレス（受信またはループバック中）あるいはソース・アドレス（送信中）をラッチし、そのアドレスがCAM エントリのいずれかと一致するかどうか判断します。

(5) CRC チェッカ

受信パケットから4バイトのCRCを計算し、それを受信パケットの最後の4バイト（FCS フィールド）と比較します。CRC チェッカは通常受信と送信中のループバック受信の両方の動作に対して有効です。

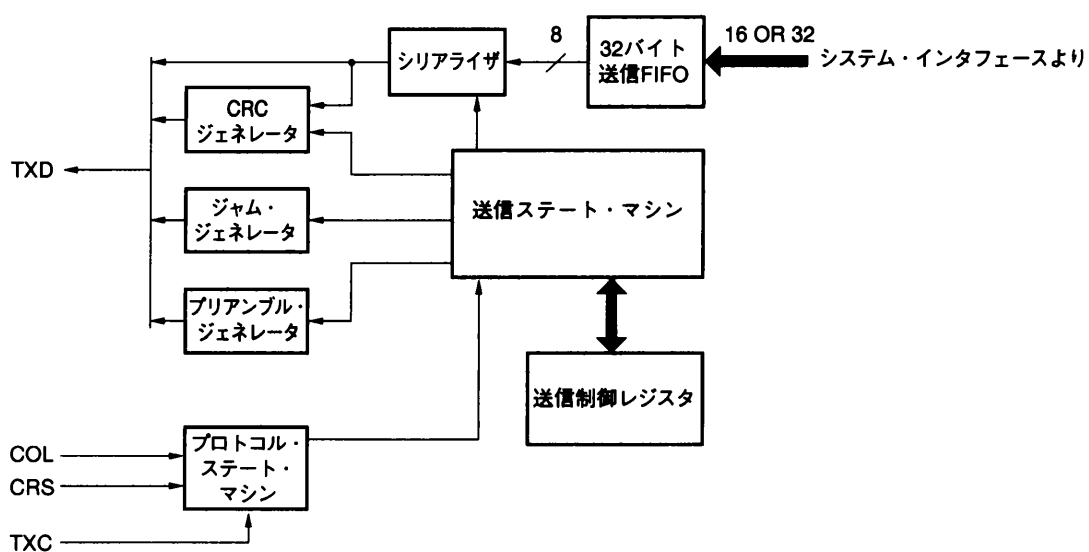
(6) コンテント・アドレス・メモリ (CAM)

16個のユーザ・プログラマブル・エントリと1つの初期化時にプログラムされたブロードキャスト・アドレス・エントリによって、受信パケットを完全にフィルタリングします。CAMには物理およびマルチキャスト・アドレス（3.2 参照）の任意の組み合わせをロードすることができます。CAM レジスタのロード手順については、5.1 を参照してください。

2.2.2 MAC 送信部

送信部（図 2-3）は送信 FIFO からデータを読み出し、ネットワークに IEEE 802.3 CSMA/CD 規格に適合したシリアル・データ・ストリームを送信します。送信部は次のブロックから構成されています。

図 2-3 MAC送信部



(1) 送信ステート・マシン (TSM)

このステート・マシンはシリアライザ、プリアンプル・ジェネレータ、およびジャム・ジェネレータの機能を制御します。このステート・マシンは、ネットワークのさまざまな条件において送信部がどのように動作するかを決定します。衝突がない場合、送信部は各パケットの先頭に62ビットのプリアンプルと2ビットのフレーム開始デリミタ（SFD）を付加し、シリアル変換したデータを送信します。パケットの終わりにはオプションで4バイトのCRCパターンを追加します（ディスエーブル可能）。衝突が発生した場合、送信部はデータ送信から4バイト・ジャム・パターン送信に切り替え、全ノードに衝突が発生したことを知らせます。万一、プリアンプル中に衝突が発生した場合、送信部はプリアンプル送信が終了するのを待ってからジャム・パターンを発生します。送信が完了すると、送信部は送信制御レジスタにステータスを書き込みます（5.3.4参照）。

(2) プロトコル・ステート・マシン

プロトコル・ステート・マシンは、μPD72932が CSMA/CD プロトコルに従うことを保証します。送信前に、プロトコル・ステート・マシンはネットワークの状況を確認するためにキャリア検出と衝突をモニタします。別のノードが送信中の場合、μPD72932はネットワークのキャリアがなくなるまで送信を延期し、フレーム間ギャップ・タイマ (9.6 μs) が満了したあとで送信を行います。フレーム間ギャップ時間は、2つの部分に分けられます。最初の6.4 μsの間にネットワークがアクティブ（他が使用中）であればフレーム間ギャップ・タイマを再スタートします。しかし、この時間を過ぎるとネットワークのアクティビティ（ネットワークが使用中）は無視され、ステート・マシンは残りの3.2 μsを待ってから送信を行います。μPD72932が送信中に衝突を検知すると、データ送信から4バイト・ジャム・パターン（すべてのビットが1である4バイト）に切り替わって送信を停止します。その後、μPD72932は“2値指数関数的休止時間増加アルゴリズム”によって決定されるバックオフ時間（休止時間）、すなわちランダム数 $r \times$ スロット時間 (51.2 μs) 待ってから、再び送信を試みます。このアルゴリズムでは n 番目の再送信が待つバックオフ時間は、スロット時間と 0 以上 2^k 以下の範囲内で決定されるランダム整数 r の積になります。

$$0 \leq r \leq 2^k$$

ここで $k = \min(n, 10)$ ……つまり、再送信の回数 n が 1 から 10 の場合は $k=n$ となりますが、 n が 11 以上の場合は $k=10$ となります。

16回目の再送信で衝突が発生した場合、μPD72932はパケット送信をアボートし、送信制御レジスタに“過剰コリジョン”エラーを報告します。

(3) シリアライザ

データが32バイト送信 FIFO に書き込まれたあと、シリアライザは FIFO からバイト幅でデータを読み出し、マンチェスタ・エンコーダへ NRZ データ・ストリームを送ります。データ送信レートは送信クロック (TXC) で決まります。SFD に続いてシリアルに変換されたデータを送信します。

(4) プリアンブル・ジェネレータ

プリアンブル・ジェネレータは各パケットの先頭に62ビットの“1, 0, 1, 0, …”の交互パターンと2ビットの“1, 1”の SFD パターンを付加します。これにより受信ノードは受信データに同期することができます。プリアンブルは衝突が発生した場合でも、常に完全に送信します。これは最小コリジョン・フラグメントが96ビット（64ビットの通常のプリアンブル+32ビットのジャム・パターン）であることを保証します。

(5) CRC ジェネレータ

CRC ジェネレータは送信シリアル・データ・ストリームから4バイトの FCS フィールドを計算します。TCR レジスタの CRCI がイネーブルにセットされた場合、4バイトの FCS フィールドは送信パケットの終わりに付加します (3.6 参照)。

(6) ジャム・ジェネレータ

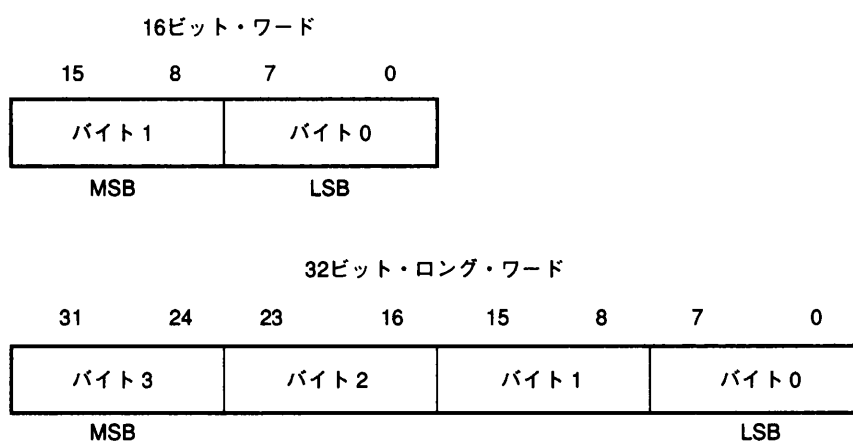
ジャム・ジェネレータは、すべてのデータが1である4バイト（32ビット）パターンを生成し、ネットワーク上の全ノードが衝突を検出できるようにします。衝突が発生すると、μPD72932はデータ送信を停止して、ジャム・ジェネレータをイネーブルします。プリアンブル中に衝突が発生した場合、μPD72932はプリアンブルの送信を終えてからジャム・ジェネレータをイネーブルします ((4) プリアンブル・ジェネレータの項参照)。

2.3 データ幅とバイト順序

μPD72932は32ビットまたは16ビット幅のいずれのメモリとでも動作するようにプログラムできます。初期化中にデータ・コンフィギュレーション・レジスタ (DCR, **5.3.2** 参照) の DW ビットをプログラムすることによって、データ幅を設定することができます。16ビット幅を選択した場合は、D15-D0 端子がデータ入出力に使用されます。また、μPD72932は BMODE 端子に適切なレベルを入力することによって、リトルまたはビッグ・エンディアン・バイト順序付けを選択することができます、ナショナル セミコンダクター (NSC)/NEC/インテル型またはモトローラ型のマイクロプロセッサとインタフェースすることができます。バイト順序付けについて次に説明します。

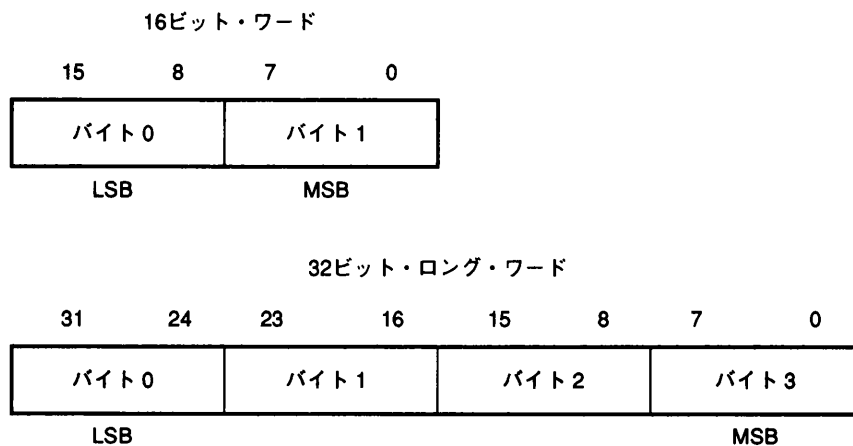
(1) リトル・エンディアン・モード (BMODE=0)

システム・メモリの受信バッファ・エリア (RBA) と送信バッファ・エリア (TBA) 内の送受信データのバイト方向は次のとおりです。



(2) ビッグ・エンディアン・モード (BMODE=1)

RBA および TBA 内の送受信データのバイト方向は次のとおりです。

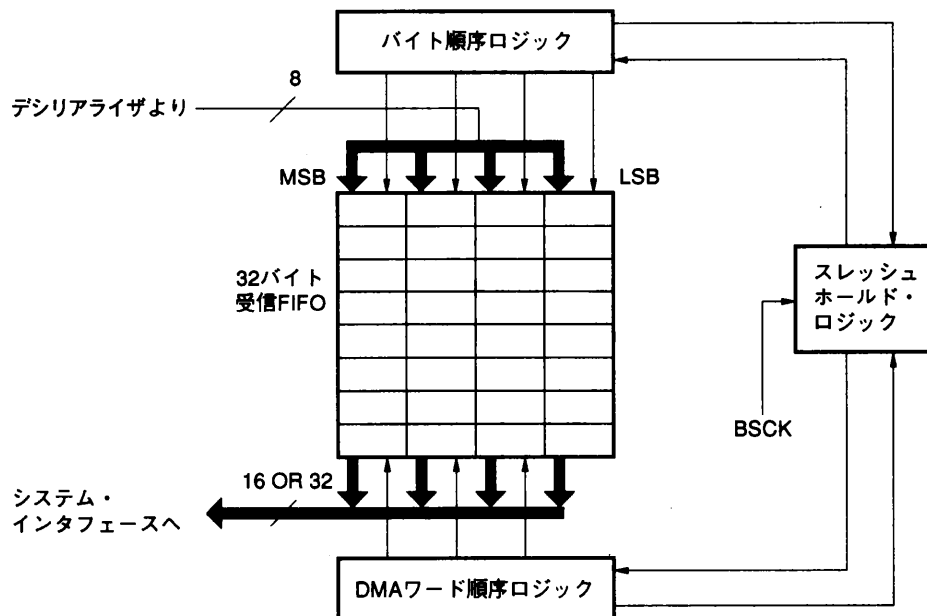


2.4 FIFO および制御回路

μPD72932はネットワークとシステム・インタフェース間でデータ転送を行うために、送信/受信それぞれに独立した32バイト FIFO を内蔵しています。FIFO はデータを一時的に記憶し、ネットワークからのリアルタイムなデータ処理要求からホスト・システムを解放します。

FIFO に対してデータを満たしたり、空けたりするための動作は、FIFO のスレッシュホールド値とブロック・モード選択ビット (BMS, 5.3.2 参照) によって制御されます。μPD72932がメモリに対して送信データのリードや受信データのライトを行うときの FIFO のフィル/エンプティ量は、スレッシュホールド値によって決まります。ブロック・モードがセットされると、転送されるバイト数はスレッシュホールド値によって決定されます。たとえば、受信 FIFO のスレッシュホールドが4ワードだと、ブロック・モード時は μPD72932は常に受信 FIFO からメモリに4ワードを転送します。エンプティ/フィル・モードがセットされると、転送されるバイト数は送信 FIFO を満たすため、および受信 FIFO を空にするために必要な数になります。パケットの受信と送信におけるスレッシュホールドの影響についての詳細は、2.4.1 と 2.4.2を参照してください。

図 2-4 受信FIFO



2.4.1 受信 FIFO

ネットワークとシステム・バスの異なる転送速度に対応するために、受信 FIFO (図 2-4) は8ビットのネットワーク (デシリアライザ) インタフェースと16/32ビット・システム・インタフェースとの間でバッファとして働きます。FIFO は3つのロジック・セクションによって制御される、幅4バイト深さ8ワードのメモリ・アレイで構成されています。受信時、バイト順序ロジックは4つのライト・ポイントのうちの1つを使用して、デシリアライザからのバイト・データを適切な順序で FIFO へ送ります。すなわち、データはビッグ・エンディアンまたはリトル・エンディアンのいずれかに対応するように、指定されたバイト順序モードに従って最下位バイト (LSB) または最上位バイト (MSB) を最初に書き込みます。

スレッシュホールド・ロジックはデシリアライザから FIFO に書き込まれるバイト数を監視します。システム・メモリに対する受信データ転送のための DMA 要求を行う前に、プログラマブル・スレッシュホールド (データ・コンフィギュレーション・レジスタ内の制御ビット RFT1, RFT0) によって、MAC ユニットから FIFO へ書き込まれるワード (またはロング・ワード) 数を決めます。受信 FIFO の中にあるデータの数にスレッシュホールド値に達すると、スレッシュホールド・ロジックはバッファ管理エンジンをイネーブルして FIFO から16または32ビット・ワー

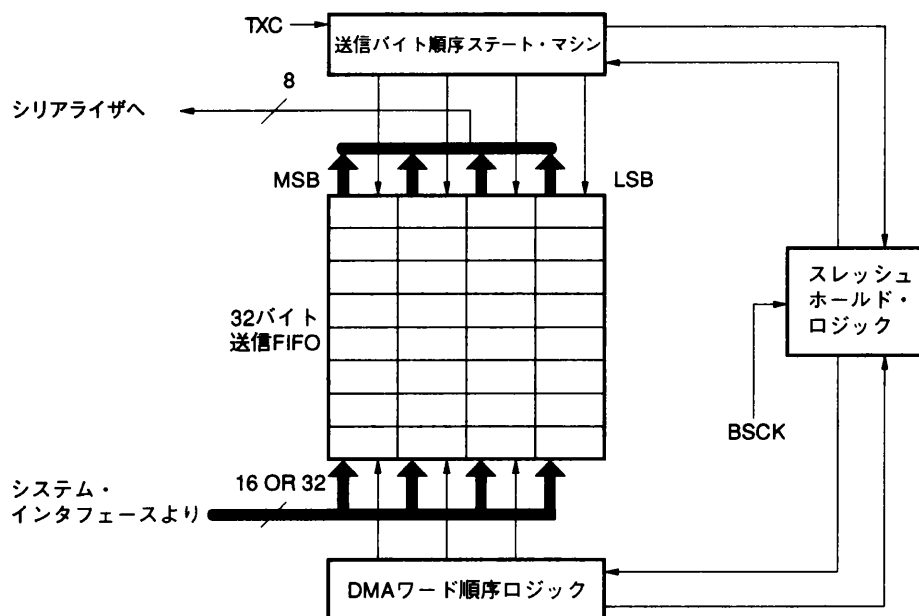
ドのデータをプログラムされた数だけ（選ばれたワード幅に従って）読み出し、DMA によってそれをシステム・インタフェース側（システム・メモリ）に転送します。たとえば、スレッシュホールドが4ワード（8バイト）だと、FIFO 内の受信データが8バイトより多くなるまで、スレッシュホールド・ロジックはバッファ管理エンジンによるメモリへの書き込みを行いません。

バッファ管理エンジンは、16ビット・モードでは FIFO の上位または下位半分（16ビット）を読み出し、32ビット・モードでは完全なロング・ワード（32ビット）を読み出します。バッファ管理エンジンが転送を終えると、FIFO の中にあるデータのバイト数はスレッシュホールド値以下になります。μPD72932がエンプティ/フィル・モードの場合、常にスレッシュホールド値以上になると、受信 FIFO を空にする分のデータを転送します。

ブロック・モードでは μPD72932がバス要求を出すたびに、スレッシュホールド値に等しいバイト数だけを転送します。スレッシュホールド・ロジックは、デシリアライザから書き込まれるバイト数を監視し続け、再びスレッシュホールド値に達するとバッファ管理エンジンをイネーブルします。ただし、ブロック・モードの場合、何かの理由（バスの待ち時間など）で送信データの DMA 転送が終わっても FIFO の中にあるデータのバイト数がスレッシュホールド値より大きいままになっていると、スレッシュホールド・ロジックはバッファ管理エンジンにメモリ書き込みを再度行うように DMA 要求を出します。このプロセスはパケットの終わりまで続きます。

パケットの終わりに達したあと、最終バイトがワード境界あるいはロング・ワード境界で終わっていなければ、シリアライザは最終ワード（16ビット・モード）あるいは最終ロング・ワード（32ビット・モード）にフィル・バイトを挿入します。フィル・バイトは OFFH です。また、FIFO の最終バイト（あるいはフィル・バイト）の直後に、受信パケットのステータスを FIFO に書き込みます。フィル・バイトや受信パケット・ステータスを含むすべてのバイトをメモリにバッファします。パケットがバッファ管理エンジンによってメモリにバッファされるときは、必ずワードあるいはロング・ワードで FIFO から取り出され、ワード（16ビット・モード）境界やロング・ワード（32ビット・モード）境界のメモリにバッファします。したがってパケットの最初のデータは16ビット・モードの奇数バイト境界にバッファせず、また32ビット・モードの奇数ワード境界にバッファしません（4.3 参照）。受信パケットのバッファ手順についての詳細は、4.4 を参照してください。

図 2-5 送信FIFO



2.4.2 送信 FIFO

受信 FIFO と同様、送信 FIFO (図 2-5) は、16/32ビット・システム・インタフェースとネットワーク (シリアルライザ) インタフェース間でバッファとして働きます。送信 FIFO も幅 4 バイト深さ 8 ワードのメモリ・アレイ (8 ロング・ワード、あるいは 32 バイト) で構成され、3 つのロジック・セクションによって制御されます。送信を開始する前に、バッファ管理エンジンはメモリから 16 または 32 ビット・ワードのデータをプログラムされている数だけフェッチして、それを FIFO に書き込みます。バッファ管理エンジンは、16 ビット・モードなら上位または下位半分 (16 ビット) を FIFO へ書き込み、32 ビット・モードでは完全なロング・ワード (32 ビット) を書き込みます。

FIFO に書き込みを行うと、スレッショールド・ロジックはそのバイト数を監視します。FIFO の中にあるデータのバイト数がスレッショールド値より大きくなると、送信バイト順序ステート・マシンが FIFO からデータの読み出しを開始し、連続バイト・ストリームを作り出してシリアルライザに送ります。たとえば、送信スレッショールドが 4 ワード (8 バイト) だとすると、送信バイト順序ステート・マシンは、FIFO の中にあるデータが 9 バイト以上になるまで FIFO からデータの読み出しを行いません。

バッファ管理エンジンは、パケットの終わりまで FIFO に送信データを補充し続けます。これは、システム・インタフェースに複数の DMA 要求を出すことによって行います。FIFO の中にあるデータのバイト数がスレッショールド値と同じかそれ以下だと、常にバッファ管理エンジンは DMA 要求を出します。ブロック・モードがセットされ、その後システムによって各 DMA 要求が認められると、バッファ管理エンジンはシステムのメモリからデータを読んでスレッショールド値に等しいバイト数を FIFO に書き込みます。エンプティ/フィル・モードがセットされると、1 回の DMA 要求で FIFO を完全に満たします。

データはビッグまたはリトル・エンディアン・バイト順序フォーマットで構成できるため、送信バイト順序ステート・マシンは 4 つのリード・ポインタのうちの 1 つを使用して 4 バイト幅の FIFO 内に適切な順序でバイト・データを書き込みます。送信バイト順序ステート・マシンは、FFIFO の有効なバイト数を同様に決定します。奇数バイト・アドレスで開始または終了するパケットについては、バッファ管理エンジンが偶数アドレス側にデータとは異なるバイトを書き込みます。送信バイト順序ステート・マシンはこれらのバイトを検出し、有効なバイトだけをシリアルライザに転送します。バッファ管理エンジンはどのバイト境界のメモリからもデータを読み出すことができます (4.3 参照)。送信バッファ管理についての詳細は、4.5 を参照してください。

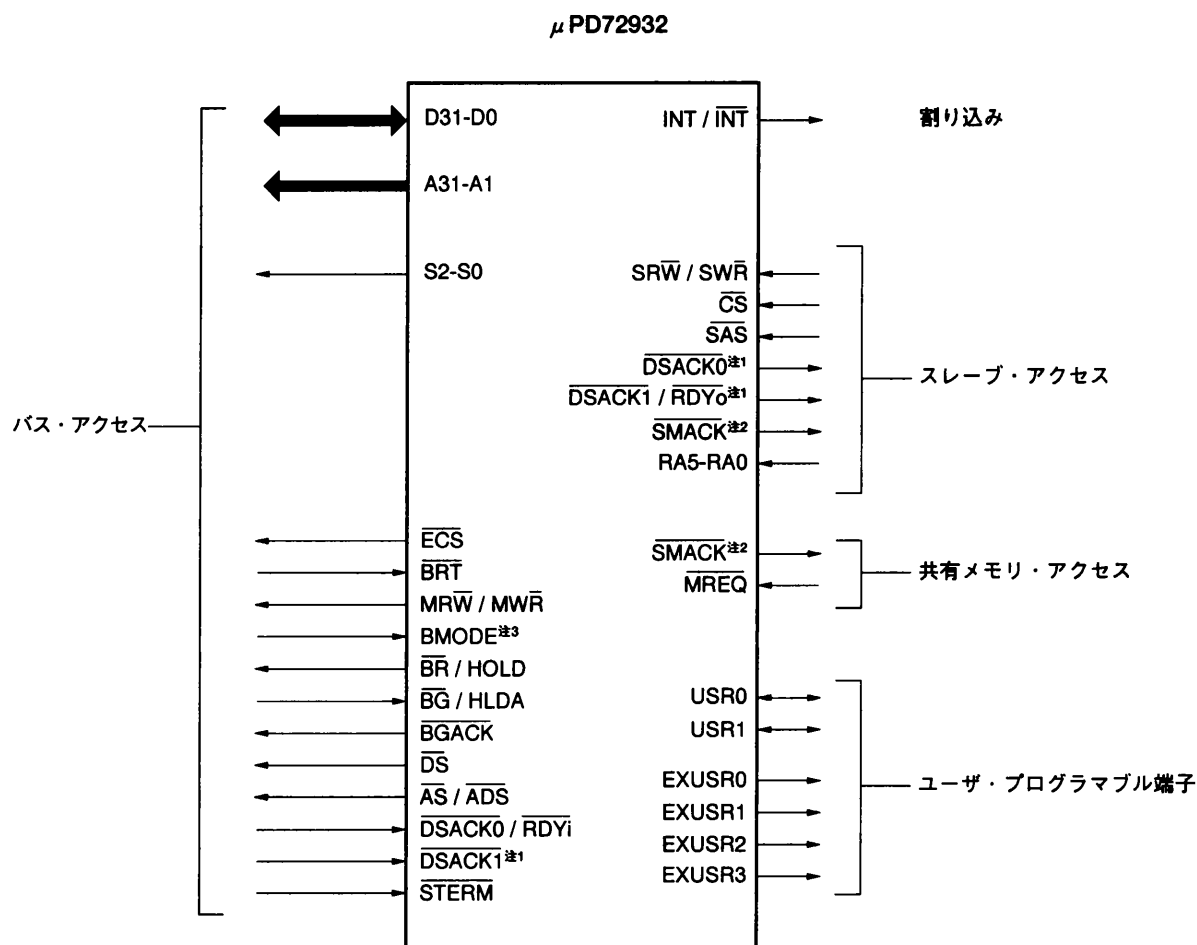
2.5 ステータス・レジスタおよびコンフィギュレーション・レジスタ

μPD72932 は、ホスト・システムとの間でステータスおよび制御情報を伝達するために、ステータス・レジスタおよび制御レジスタをそれぞれ 1 個内蔵しています。μPD72932 はこれらのレジスタを使用して、システムから生成されるコマンドのロード、送受信ステータスの表示、データのバッファ、および割り込み制御を行います。各レジスタは 16 ビット長です。レジスタの詳細については、5. μPD72932 レジスタを参照してください。

2.6 バス・インタフェース

システム・インタフェース (図 2-6) は、各種バスとインタフェースするために必要な端子で構成されています。システム・インタフェースは、データとアドレス・ラインのための I/O ドライバ、マイクロプロセッサのためのバス・アクセス制御、同期または非同期システムのためのレディ (ready) ロジック、スレーブ・アクセス制御、割り込み制御、および共有メモリ・アクセス制御回路を持っています。図 2-6 に機能別に分けて、これらの信号線を示します。μPD72932 バス・インタフェースについての詳細は、6. バス・インタフェースを参照してください。

図 2-6 バス・インタフェース信号



注1. $\overline{DSACK0}$, $\overline{DSACK1}$ は、バス・アクセス制御とスレーブ・アクセス制御の両方に使用される双方向の信号です。

2. $\overline{SMACK}$ は、スレーブ・アクセスと共有メモリ・アクセスの両方に使用します。

3. BMODE 端子によって、ナショナル セミコンダクター (NSC) / NEC / インテル型またはモトローラ型のバスを選択します。

2.7 ループバックおよび診断

μPD72932は MAC ユニットからトランシーバまでを自己テストするための3つのループバック機能を備えています。ループバック機能によって、チップの内部送受信動作を自己テストすることができます。ループバック中、μPD72932は送信したパケットを受信部に戻します。そこでアドレス認識ロジックでパケットをフィルタし、受け入れ可能な場合はメモリにバッファします。送受信ステータスおよび割り込みは、ループバック中もアクティブのままです。つまり、システムにとって、ループバック使用時には、パケットの送受信はあたかも同じシステム・バスとメモリに接続された2個の独立したチップ間で行われているように見えます。

2.7.1 ループバック機能

(1) MAC ループバック

送信データは MAC ユニットでループバックされます。内部 ENDEC や外部 ENDEC (外部 ENDEC のインタフェース端子はドライブされない) のいずれに対してもデータを送信しません。したがって、μPD72932 からデータを送信しません。MAC ループバックでは ENDEC を使用しませんが、ENDEC はクロック (内部 ENDEC では発振器や水晶振動子、外部 ENDEC では TXC) を供給しなければなりません。また MAC ループバックは衝突などのネットワークの影響を受けません。したがって MAC ループバックでは CSMA/CD のプロトコルに基づいて動作していません。

(2) ENDEC ループバック

送信データは ENDEC でループバックされます。内部 ENDEC の使用時には、データは ENDEC の送信部から受信部に送られます (図 2-1 参照)。μPD72932 からデータは送信されず、COL 信号および CD 土は無視されるので、ENDEC のループバックはネットワークの影響を受けません。MAC からの LBK 信号によって ENDEC はループバック・モードになります。

外部 ENDEC の使用時には、ENDEC は LBK 信号をアクティブにし、外部 ENDEC ループバック・モードで動作するように指示します。μPD72932 からデータを送信しないにもかかわらず、MAC ユニットは CSMA/CD の MAC プロトコルに従って動作します。

(3) トランシーバ・ループバック

送信データは、(μPD72932 のループバック・モードの設定に関係なく、常に) 外部トランシーバでループバックされます。データがチップから送信されるため、CSMA/CD の MAC プロトコルに従います。このため、トランシーバ・ループバックはネットワークの影響を受けます。トランシーバ・ループバックと通常の動作時のループバックでは、μPD72932 の動作には基本的な違いがあります。トランシーバ・ループバックの場合、μPD72932 は受信 FIFO を使用しパケットをメモリにバッファしますが、通常動作の場合、トランシーバによってループバックされているパケットをモニタするだけで、受信 FIFO にデータを入れたりパケットをバッファしたりしません。

2.7.2 ループバック手順

ループバック動作は次の手順で行います。

- (1) 送受信エリアを 4.4 および 4.5 で説明するとおりに初期化します。
- (2) CAM アドレス・レジスタ (5.1 参照) の 1 つにアドレスをロードしますが、μPD72932 のアドレス認識能力を確認するときはパケットのデスティネーション・アドレスをロードします。
- (3) CAM アドレス・レジスタの 1 つにアドレスをロードしますが、デスティネーション・アドレスと異なる場合はパケットのソース・アドレスをロードし、送信ステータス (5.3.4 参照) のモニタ・パケットの不良 (PMB) エラーの発生を防ぎます。
- (4) 任意の受信フィルタおよびループバック・モード (LB1, LB0) で受信制御レジスタをプログラムします。
- (5) 送信コマンド (TXP) を発行すると同時に、コマンド・レジスタ内のレシーバ (RXEN) をイネーブルします。

μPD72932 はパケットを完全に受信 (または、アドレス不一致の場合は拒否) したあとで、ループバック動作を終了します。送信制御レジスタおよび受信制御レジスタはループバック・パケットを通常動作の場合と同様に扱い、ステータスを表示します。割り込みマスク・レジスタ内で割り込みがイネーブルされていれば、割り込みも発生します。

注意 MAC ループバックでは、パケットを 1 つだけしかキューできません。これは、送信 MAC 部が送信されたパケット間に対しフレーム間ギャップ・タイム (IFG) を生成しないため、受信 MAC 部が受信ステータスを更新しないことによるものです。その他のループバック・モードについては特別な制御は必要ありません。

2.8 ネットワーク管理機能

μPD72932は、ノードがネットワークの機能全体を監視できるように、レイヤ管理のための IEEE 802.3 規格 (LME : Layer Management) を完全にサポートします。受信や送信の終わりに、1 パケットごとに統計が得られます。さらに、μPD72932は CRC エラー、フレーム配列エラー、ミス・パケットの数を計数するための 3 つのタリー・カウンタを備えています。表 2-1 に μPD72932 によって得られる統計を示します。

表 2-1 ネットワーク管理統計

統計量	使用レジスタ	使用ビット
フレーム送信完了 (正常)	TCR ^{注1}	PTX
シングル・コリジョン・フレーム	注1	NC0-NC4
マルチプル・コリジョン・フレーム	注1	NC0-NC4
コリジョン・フレーム	注1	NC0-NC4
遅延送信フレーム	TCR ^{注1}	DEF
ウィンドウ外コリジョン	TCR ^{注1}	OWC
過剰コリジョン	TCR ^{注1}	EXC
過剰遅延	TCR ^{注1}	EXD
内部 MAC 転送エラー	TCR ^{注1}	BCM, FU
受信フレーム数 (正常)	RCR ^{注2}	PRX
マルチキャスト・フレーム受信数	RCR ^{注2}	MC
ブロードキャスト・フレーム受信数	RCR ^{注2}	BC
FCS エラー数	CRCT RCR	全ビット CRC
フレーム構成エラー	FAET RCR	全ビット FAE
内部 MAC により消失したエラー数	MPT ISR	全ビット RFO

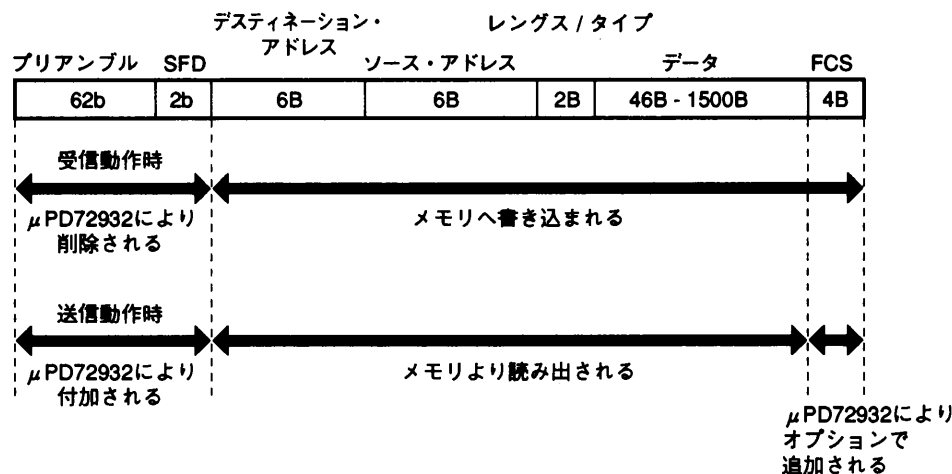
注 1. コリジョン数と送信制御レジスタの内容は TXpkt.status フィールド (4.5.1 (1) 参照) で確認できます。

2. 受信制御レジスタの内容は RXpkt.status フィールド (4.4.3 (1) 参照) で確認できます。

3. 送信/受信 IEEE 802.3フレーム・フォーマット

IEEE 802.3標準で規定されたパケット・フォーマット(図3-1)は、プリアンプル、フレーム開始デリミタ(SFD)、デスティネーション・アドレス、ソース・アドレス、レングス、データ、およびフレーム・チェック・シーケンス(FCS)の各フィールドで構成されています。代表的なフォーマットを図3-1に示します。フィールドは、データ・フィールドを除いてすべて固定長です。μPD72932は送信中、プリアンプル、SFDとFCSフィールドを生成し付加します。受信中、プリアンプルとSFDフィールドを削除します(CRCは受信中バッファ・メモリに転送します)。

図3-1 IEEE 802.3パケット構成



備考 B = バイト, b = ビット

3.1 プリアンプルおよびフレーム開始デリミタ (SFD)

ENDEC はマンチェスタ・コードにエンコードされた62ビットの“1, 0, 1, 0, …”パターンのプリアンプル・フィールドを使用して、受信パケットに同期化します。パケットがネットワークを通過して移動する際、このプリアンプルの数ビットは失われる可能性があります。“1”が2つ連続したフレーム開始デリミタ (SFD) を検出すると、バイト整列を開始します。

3.2 デスティネーション・アドレス

デスティネーション・アドレスは、ネットワーク上のパケットの宛先を示し、不要なパケットがシステムに転送されないようにフィルタするのに使用します。μPD72932によってサポートされるアドレス・フォーマットには、物理アドレス、マルチキャスト・アドレス、ブロードキャスト・アドレスの3種類があります。

(1) 物理 (フィジカル) アドレス

物理アドレスは単一ノードに対応する固有のアドレスです。すべての物理アドレスは、アドレスの最初のバイトのLSBが“0”にセットされています。このアドレスは内部に記憶されたCAM (コンテンツ・アドレス・メモリ) アドレス・エントリと比較されます。RCRレジスタのPROビットが“0”の場合、μPD72932がパケットを受信するためには、デスティネーション・アドレスの全ビットがCAMのエントリと一致しなければなりません。

(2) マルチキャスト・アドレス

マルチキャスト・アドレスは、アドレスの最初のバイトの LSB が“1”にセットされています。マルチキャスト・アドレスも物理アドレス同様、RCR レジスタの AMC ビットが“0”の場合 CAM のエントリと一致しなければなりません。これによってマルチキャスト・パケットの完全なフィルタリングが行われ、マルチキャスト・パケットをマップするためのハッシング・アルゴリズムが不要になります。

(3) ブロードキャスト・アドレス

アドレスがすべて“1”で構成されている場合、ブロードキャスト・アドレスとなり、パケットが全ノードを対象としていることを示します。RCR レジスタの BRD ビットが“1”にセットされている場合、 μ PD72932はこのフレームを受信します。

μ PD72932はすべての物理アドレス・パケットを受信する“透過モード”(受信制御レジスタの PRO ビットをセット)も備えています。物理、マルチキャスト、ブロードキャスト、および透過モードは、受信制御レジスタによって選択できます。

3.3 ソース・アドレス

ソース・アドレスは送信ノードの物理アドレスです。ソース・アドレスがマルチキャスト・アドレスやブロードキャスト・アドレスとなることはありません。このフィールドは、システム・ソフトウェアによって μ PD72932の送信バッファに転送されなければなりません。送信中、 μ PD72932は自己受信パケットの CRC を監視する前に、ソース・アドレスと内部 CAM アドレス・エントリを比較します。

送信されたパケットのソース・アドレスが CAM の値と一致しなければ、モニタ・パケット不良フラグ (PMB) が送信ディスクリプタの送信ステータス・フィールドにセットされます (4.5.1 (1) と 5.3.4 参照)。

μ PD72932はソース・アドレスの生成を行いません。しかし、ソース・アドレスだけを含む送信ディスクリプタ・フラグメントを各パケットに生成することは可能です (4.5.1 参照)。

3.4 レングス/タイプ・フィールド

IEEE 802.3のパケットでは、このフィールドはパケットのデータ・フィールドに含まれるデータのバイト数を示します。Ethernet I および II では、このフィールドはパケットのタイプを示します。 μ PD72932はこのフィールドに対しては操作を行いません。

3.5 データ・フィールド

データ・フィールドには、46から1500バイトまで、IEEE 802.3およびイーサネット仕様によって決定されるさまざまなオクテット長が含まれています。1500バイト以上の長さのメッセージは、IEEE 802.3ネットワークでは複数のパケットに分割する必要があります。46バイトより短いデータ・フィールドは、パッド(任意のキャラクタ)を追加してフレーム長を64バイトにすることが必要です。データ・フィールドにパッドが含まれる場合、有効バイト数は前述のレングス・フィールドで示されます。 μ PD72932は送信中にショート・パケットにパッド・バイトを追加したり、受信中にオーバサイズ・パケットをチェックすることはありません。しかし、ユーザのドライバ・ソフトウェアは、TXpkt.pkt_size フィールドと TXpkt.frag_size フィールドを最低64バイトまでに設定すれば、簡単にパッドを追加できます (4.5.1 参照)。IEEE 802.3標準およびイーサネット仕様においてデータ・フィールドの最大バイト数が規定されていますが、 μ PD72932はそれらの規定にかかわらず最大64 K バイトのパケットを送受信することができます。

3.6 FCS フィールド

フレーム・チェック・シーケンス (FCS) は、エラーのないパケットを検出するために、送信中に計算され、パケットの終わりに追加される32ビットの CRC フィールドです。CRC 計算には、AUTODIN II ($X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$)の多項式を使用します。 μ PD72932はオプションで送信中に CRC シーケンスを追加することができ、通常受信と送信中のループバック受信 (2.2.1 参照)の両方で CRC をチェックします。

3.7 MAC (メディア・アクセス・コントロール) への適合

μPD72932は、IEEE 802.3 MAC 仕様に準拠するよう設計されています。μPD72932は大部分の MAC 機能をチップ上に実現し、残りの機能をユーザのソフトウェアで実現できるようにしています。MAC の仕様を表 3-1 に示します。

表 3-1 MAC 仕様

適合テスト項目	μPD72932がサポート	ユーザのドライバ・ソフトウェアで設定
最小フレーム長	○	
最大フレーム長 ^{注1}	○	○
アドレス生成 ^{注2}	○	○
アドレス認識	○	
パッド長生成 ^{注3}	○	○
フレーム開始デリミタ	○	
レングス・フィールド	○	
プリアンプル生成	○	
送信ビット順序	○	
フレーム長の不一致 ^{注1}	○	○
オクテット境界でない	○	
FCS 誤り	○	
フレーム組み立て	○	
FCS 生成/追加	○	
キャリア相違	○	
フレーム間隔	○	
コリジョン検出	○	
コリジョン処理	○	
コリジョンによる送信停止と再送信	○	
FCS 確認	○	
フレーム解体	○	
バック・トゥ・バック・フレーム	○	
フロー制御	○	
アテンプト・リミット	○	
ジャム・サイズ (SFD 後)	○	
ジャム・サイズ (プリアンプル中)	○	

- 注 1. μPD72932はパケット全体のバイト・カウントを RXpkt.byte_count で供給します (4.4.3 参照)。ユーザのドライバ・ソフトウェアはバイト・カウントに基づいてパケットをさらにフィルタすることが可能です。
2. μPD72932はソース・アドレスの生成を行いません。ただし、ソース・アドレスだけを含む送信ディスクリプタ・フラグメントを各パケットに生成することは可能です。4.5.1 を参照してください。
3. μPD72932はパッド生成を行いません。ただし、ユーザのドライバ・ソフトウェアで TXpkt.pkt_size フィールドと Txpkt.frag_size フィールドを最低64バイトに設定すれば、簡単にパッドを追加できます。4.5.1 を参照してください。

4. バッファ管理

4.1 バッファ管理の概要

μPD72932のバッファ管理方式はデータ・バッファとディスクリプタをベースにしています(図4-2および図4-11参照)。受信、あるいは送信されるパケットは、受信バッファ・エリア(RBA)と送信バッファ・エリア(TBA)に置かれます。システムは、受信リソース・エリア(RRA)と受信ディスクリプタ・エリア(RDA)と送信ディスクリプタ・エリア(TDA)にある情報により、これらのバッファのパケット管理を行います。

4.2 ディスクリプタ・エリア

ディスクリプタは、μPD72932が使用するバッファ管理方式の基本となっています。RDAはRBA内の受信パケットをポイントし、RRAはRBAをポイントし、TDAは送信されるパケットを含んでいるTBAを指します。次の3つのセクションでは、これらのディスクリプタを記述するための規約とレジスタについて説明します。

4.2.1 ディスクリプタのネーミング規則

ディスクリプタを構成するフィールドは、各ディスクリプタの使用法を連想させるよう一貫した方法で命名します。各ディスクリプタ名は次のフォーマットで示される3つの部分で構成されています。

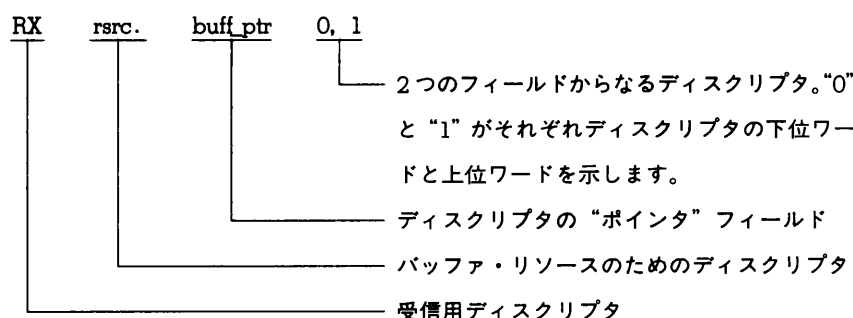
[RX/TX] [ディスクリプタ名] . [フィールド]

最初の大文字2文字はディスクリプタが送信(TX)と受信(RX)のいずれに使用されるかを示し、次に以下のいずれかのディスクリプタ名が続きます。

rsrc=リソース・ディスクリプタ

pkt=パケット・ディスクリプタ

最後の構成部分にはフィールド名が入り、ディスクリプタの他のフィールドと区別します。フィールド名はピリオド(".")でディスクリプタ名と分けられています。ディスクリプタの一例を以下に示します。



4.2.2 省略形

μPD72932レジスタやメモリ内のデータ構成の説明には、表4-1に示す省略形を使用します。省略形の“0”と“1”は、レジスタまたはディスクリプタの下位ワードと上位ワードを示します。ディスクリプタの省略形も表4-1に示します。

表 4-1 ディスクリプタの省略形

送信/受信エリア	
RRA	受信リソース・エリア
RDA	受信ディスクリプタ・エリア
RBA	受信バッファ・エリア
TDA	送信ディスクリプタ・エリア
TBA	送信バッファ・エリア
バッファ管理レジスタ	
RSA	リソース・スタート・エリア・レジスタ
REA	リソース・エンド・エリア・レジスタ
RRP	リソース・リード・ポインタ・レジスタ
RWP	リソース・ライト・ポインタ・レジスタ
CRDA	カレント受信ディスクリプタ・アドレス・レジスタ
CRBA0, 1	カレント受信バッファ・アドレス・レジスタ
TCBA0, 1	テンポラリ・カレント・バッファ・アドレス・レジスタ
RBWC0, 1	リメイニング・バッファ・ワード・カウント・レジスタ
TRBWC0, 1	テンポラリ・リメイニング・バッファ・ワード・カウント・レジスタ
EOBC	エンド・オブ・バッファ・カウント・レジスタ
TPS	送信パケット長レジスタ
TSAO, 1	送信開始アドレス・レジスタ
CTDA	カレント送信ディスクリプタ・アドレス・レジスタ
TFC	送信フラグメント・カウント・レジスタ
TFS	送信フラグメント長レジスタ
UTDA	送信ディスクリプタ上位アドレス・レジスタ
URRA	受信リソース上位アドレス・レジスタ
URDA	受信ディスクリプタ上位アドレス・レジスタ
送信/受信ディスクリプタ	
RXrsrc.buff_ptr0, 1	RRA バッファ・ポインタ・フィールド
RXrsrc.buff_wc0, 1	RRA バッファ・ワード・カウント・フィールド
RXpkt.status	RDA 受信ステータス・フィールド
RXpkt.byte_count	RDA パケット・バイト・カウント・フィールド
RXpkt.buff_ptr0, 1	RDA バッファ・ポインタ・フィールド
RXpkt.link	RDA 受信ディスクリプタ・リンク・フィールド
RXpkt.in_use	RDA イン・ユース・フィールド
TXpkt.frag_count	TDA フラグメント・カウント・フィールド
TXpkt.pkt_size	TDA パケット・サイズ・フィールド
TXpkt.pkt_ptr0, 1	TDA パケット・ポインタ・フィールド
TXpkt.frag_size	TDA フラグメント・サイズ・フィールド
TXpkt.link	TDA 送信ディスクリプタ・リンク・フィールド

4.2.3 バッファ管理エリアのベース・アドレス

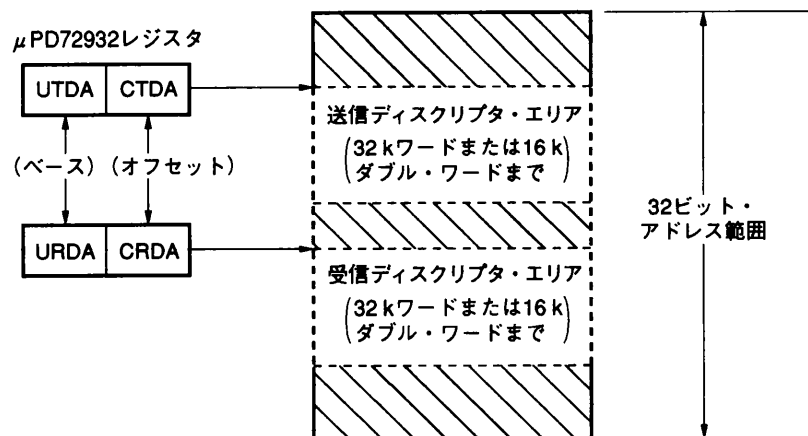
μPD72932はディスクリプタ情報を記憶するためにメモリ内で送信ディスクリプタ・エリア (TDA), 受信ディスクリプタ・エリア (RDA), および受信リソース・エリア (RRA) の3つのエリアを使用します。μPD72932は16ビット・ベース・アドレス・レジスタを16ビット・オフセット・レジスタに連結してこのエリアにアクセスします。アドレスの固定上位16ビットをベース・アドレス・レジスタが供給し、アドレスの下位16ビットをオフセット・レジスタが供給します。ベース・アドレス・レジスタとは上位送信ディスクリプタ・アドレス (UTDA) レジスタ, 上位受信ディスクリプタ・アドレス (URDA) レジスタ, および上位受信リソース・アドレス (URRA) レジスタを指します。これに対応するオフセット・レジスタを以下に示します。

上位アドレス・レジスタ	オフセット・レジスタ
URRA	RSA, REA, RWP, RRP
URDA	CRDA
UTDA	CTDA

レジスタ名と種類については表 4-1 を参照してください。

図 4-1 は UTDA レジスタおよび URDA レジスタによって配置される送信ディスクリプタ・エリアと受信ディスクリプタ・エリアの例です。ディスクリプタ・エリア, RDA, TDA, および RRA のベース・アドレスは同一にすることもできます (URRA=URDA=UTDA)。しかし、これらのエリアがお互いに重ね書きしないよう注意してください。

図 4-1 送信/受信ディスクリプタ・エリア・ポインタ



4.3 ディスクリプタ・データの配列

ディスクリプタ (RXpkt.×××, RXsrc.×××, TXpkt.×××) によって使用されるフィールドはすべてワード・サイズ (16ビット) であり、16ビット・メモリに対してはワードの境界 (A0=0) に、32ビット・メモリに対してはロング・ワードの境界 (A1, A0=0, 0) に配列しなければなりません。受信バッファ・エリア (RBA) も16ビット・モードではワードの境界、32ビット・モードではロング・ワードの境界に配列しなければなりません。しかし、送信バッファ・エリア (TBA) のフラグメントのみは任意のバイト境界に配列することができます。

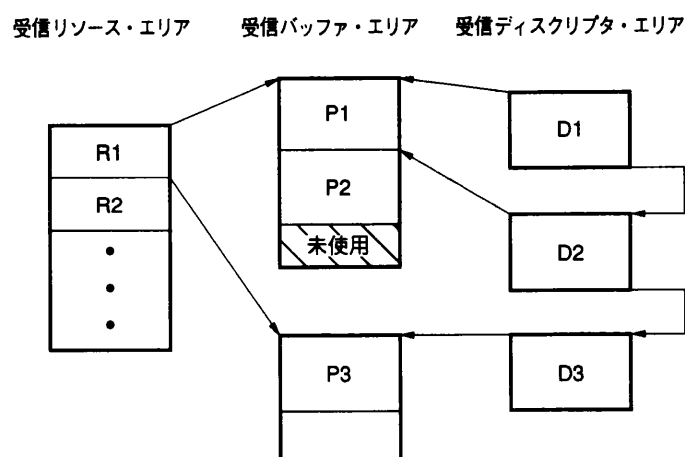
4.4 受信バッファ管理

受信バッファ管理は、受信中にデータ、ステータス、および制御情報を書き込むメモリ内の3つのエリアを操作

します(図4-2)。これらのエリアはレシーバをイネーブルする(コマンド・レジスタのRXENビットをセットする)前に初期化しておかなければなりません(4.4.4参照)。受信リソース・エリア(RRA)には、受信バッファ・エリアをシステム・メモリに配置するディスクリプタが格納されます。図4-2のR1, R2などがこのディスクリプタを示しています。パケット(P1, P2などで表示)はそのRDAに、対応するRBAにバッファされます。各バッファ・エリア・サイズとパケット・サイズによって、一個あるいは複数のパケットが各RBAにバッファされます。受信ディスクリプタ・エリア(RDA)には各受信パケットに対応する各パケット(図4-2のD1, D2など。D1はP1, D2はP2に対応)のステータスおよび制御情報が格納されます。

パケットが到着するとアドレス認識ロジックが、物理アドレス、マルチキャスト・アドレス、またはブロードキャスト・アドレスの一致をチェックし、パケットが受け入れ可能ならμPD72932は選択されている受信バッファ・エリア(RBA)に書き込みます。エンド・オブ・パケット処理のあと、μPD72932は完全なパケットが1つの連続ブロックに書き込まれたことを確認し、受信ステータス、バイト・カウント、パケット格納アドレスを受信ディスクリプタ・エリア(RDA)に書き込みます。μPD72932はその後、ポインタを更新して次のディスクリプタを設定するとともに、RBA内で使用できる残りのワード数をチェックします。スペースが十分に残っていれば、μPD72932は受信したパケットのすぐあとに次のパケットをバッファします。カレント・バッファのスペースが十分でなければ、システムによって割り当てられている追加のバッファを得るために、μPD72932はリソース・ディスクリプタを受信リソース・エリア(RRA)からフェッチします。

図4-2 受信バッファ管理の概要



4.4.1 受信リソース・エリア (RRA)

受信リソース・エリア(RRA)は、μPD72932が受信データ転送によってバッファ・メモリを消費すると、システムがμPD72932に追加バッファ・スペースを割り当てるようにさせます。システムは、μPD72932がカレント・バッファ・エリアを使い切ると受信ディスクリプタとともに、このエリア(RRA)をロードします。各リソース・ディスクリプタは、RBAの開始点を指す32ビット・バッファ・ポインタと、バッファ・サイズをワード数で示す(ワード当たり2バイト)32ビット・ワード・カウントからなります。バッファ・ポインタとワード・カウントは、図4-3に示すように、各構成部分が16ビット・フィールドからなるフォーマットを使用して連続して配置されます。μPD72932はこの情報を内部に記憶し、バッファ・ポインタとワード・カウントの対応するフィールドを連結して32ビット・ロング・ワードを生成します。32ビット・モードでは、メモリ中の上位ワード(D31-D16)は使用されないことに注意してください。μPD72932はRRAへの書き込みを行わないので、このエリア(メモリ中の上位ワード)を他の用途に使用することができます。

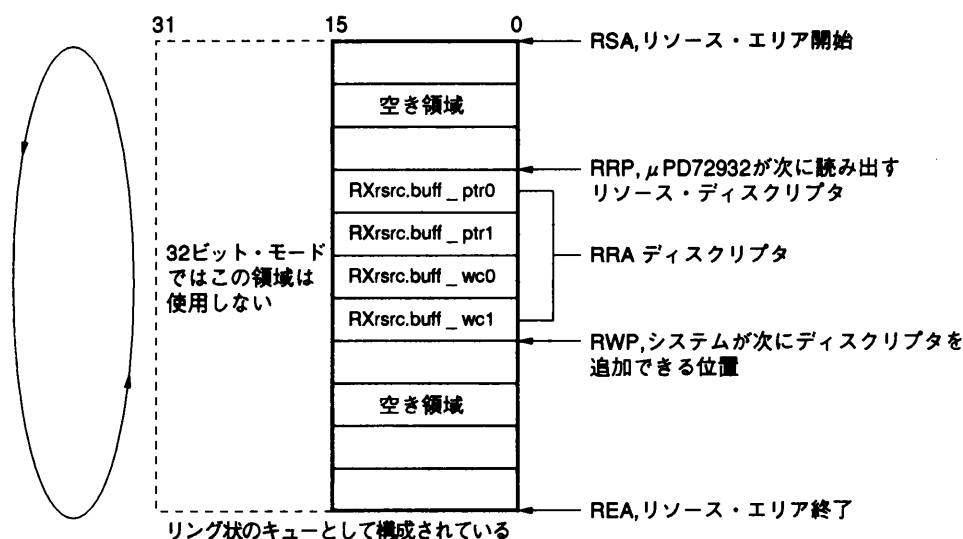
μPD72932はディスクリプタを効果的に処理するために、RRAをリング状のキューとして構成します。RRAは4

つのレジスタで定義されます。リソース・スタート・エリア (RSA) レジスタとリソース・エンド・エリア (REA) レジスタが、RRA の開始位置と終了位置を決定し、残り 2 つのレジスタが RRA を更新します。システムはリソース・ライト・ポインタ (RWP) で指定されたアドレスでディスクリプタを追加し、μPD72932 がリソース・リード・ポインタ (RRP) で指定された、次のディスクリプタを読み出します。μPD72932 は RRA の内容を読み出すと、16 ビット・モードなら 4 ワード (32 ビット・モードなら 4 ロング・ワード) だけ RRP を進め、REA と等しくなると自動的に RSA をセットします。

RRA のディスクリプタが読み出されたあと、RXrsrc.buff_ptr0, 1 が CRBA0, 1 レジスタにロードされ、RXrsrc.buff_wc0, 1 が RBWC0, 1 レジスタにロードされます。

RRA の配置はデータ幅のモードにより、ワードの境界かロング・ワードの境界のいずれかに限定されます。16 ビット・モードでは RRA はワードの境界 (A0 は常に 0) に、32 ビット・モードではロング・ワードの境界 (A0 および A1 は常に 0) に配置しなければなりません。

図 4-3 受信リソース・エリアのフォーマット



4.4.2 受信バッファ・エリア (RBA)

μPD72932 は受信パケット・データを、前述のとおり RRA のリソース・ディスクリプタによって指定された RBA に書き込みます。RRA の RXrsrc.buff_wc0, 1 フィールドが RBA の長さを示します。

μPD72932 が RRA から RBA を取得するとき、バッファ内の残存スペース (ワード) の記録を行うリメインিং・バッファ・ワード・カウント (RBWC0, 1) レジスタに、RXrsrc.buff_wc0, 1 の値がロードされます。これらのレジスタはバッファ内の残存ワード数を保持しています。パケットが RBA へ書き込まれるときは、連続して書き込まれます (μPD72932 は複数のバッファやフラグメントにパケットを分散しません)。したがって、バッファにパケットを収納したあと、さらに次の最大サイズのパケット (ネットワークから送信される規格内のパケット、たとえば 1500 バイト + α) をバッファする十分なスペースがない場合、あらたにバッファが必要になります。エンド・オブ・バッファ・カウント (EOBC) レジスタを使用して、バッファしなければならない最大パケット・サイズを μPD72932 に通知します。

(1) エンド・オブ・バッファ・カウント (EOBC)

EOBC は RBA 中でバッファのボトムを基にして決められた境界を指します。EOBC に書き込まれた値は μPD72932 によってバッファされるネットワークからのパケットの最大予測サイズ (ワード) を表します。このワード・

カウントによってRBA内に境界が生成され、これを越えると、 μ PD72932がRRAから新しいRBAリソースをフェッチします。

注意 EOBCはワード・カウントであり、バイト・カウントではありません。また、EOBCに設定する値は、 μ PD72932が32ビット・モードの場合、ダブル・ワード（32ビット）サイズでなければなりません（すなわち、IEEE 802.3規格の最大パケット・サイズは759ワードですが、32ビット・モードでは、EOBCを759ワードでなく760ワードに設定します）。

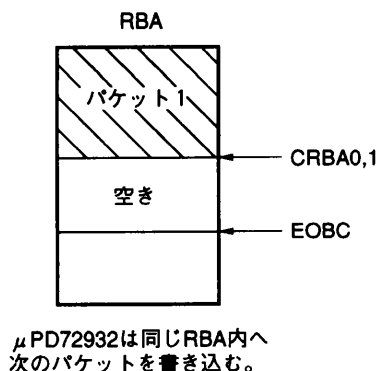
(2) RBA内での最終パケットのバッファ管理

受信を開始すると、 μ PD72932はパケットをカレント受信バッファ・アドレス（CRBA0, 1）レジスタで示される値で始まるアドレスに格納し、受信が完了するまで続行します。受信と同時に、 μ PD72932はリメイニング・バッファ・ワード・カウント（RBWC0, 1）レジスタを、16ビット・モードなら1つずつ、32ビット・モードなら2つずつデクリメントします。受信の終了時に、パケットがエンド・オブ・バッファ・カウント・レジスタ（EOBC）の指し示す境界を越えていると、 μ PD72932は次のパケットがRBAの残りのエリアに入らないかもしれないことを認識します。このチェックはRBWC0, 1レジスタをEOBCと比較することにより行われます。RBWC0, 1がEOBCよりも小さい（最終パケットがEOBCの境界を越えてバッファされている）場合、 μ PD72932はRRA内の次のリソース・ディスクリプタをフェッチします。RBWC0, 1がEOBCと等しいかあるいはそれより大きい（EOBCの境界を越えていない）場合は、 μ PD72932は次のパケット受信を同じRBA内のCRBA0, 1レジスタで指し示されている位置から受信します。図4-4に（a）RBWC0, 1 $\geq$ EOBCの場合、（b）RBWC0, 1 < EOBCの場合の μ PD72932の動作を示します。EOBCの設定についての詳細は、4.4.4（4）を参照してください。

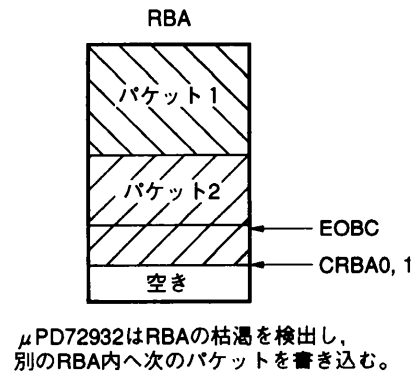
注意 EOBCの境界を越える場合注意が必要です。すなわち、図4-4において、（b）の状態になる前に（a）の状態になっていなければなりません。最初に（a）の状態にならずに（b）の状態になると、RBWC0, 1 < EOBCのテストは正常に行われず、 μ PD72932は新しいバッファをフェッチしません。この結果、バッファのオーバーフローが発生します（5.3.6 割り込みステータス・レジスタのRBAEを参照）。

図4-4 受信バッファ・エリア

（a）RBWC0, 1 $\geq$ EOBCの場合



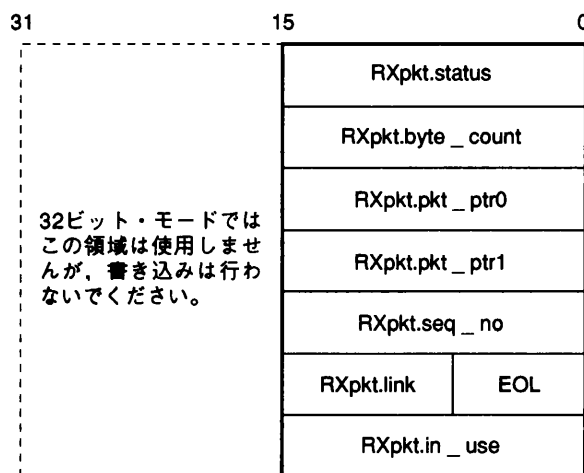
（b）RBWC0, 1 < EOBCの場合



4.4.3 受信ディスクリプタ・エリア (RDA)

μPD72932はパケットをメモリへバッファしたあと、ステータスおよび制御情報の6ワードをRDAに書き込み、さらに次の受信ディスクリプタへのリンク・フィールドを読み出します。またさらにカレント・ディスクリプタのインユース・フィールドにも書き込みます。32ビット・モードでは上位ワード、D31-D16は使用しません。しかし、この場所にμPD72932が書き込みを行う場合もあるので、他の用途にこのメモリ領域を使用することはできません。各受信ディスクリプタは、次のセクションから成ります (図4-5参照)。

図4-5 受信ディスクリプタのフォーマット



(1) RXpkt.status (受信ステータス)

受信パケットのステータスを示します。μPD72932は受信制御レジスタの内容をこのフィールドへ書き込みます。図4-6に受信ステータスのフォーマットを示します。このフィールドには受信制御レジスタの内容がロードされます。ERR, RNT, BRD, PRO, および AMC はコンフィギュレーション・ビットであり、初期化時にプログラムしてください。受信制御レジスタの詳細については5.3.3を参照してください。

図4-6 受信ステータスのフォーマット

15	14	13	12	11	10	9	8
ERR	RNT	BRD	PRO	AMC	LB1	LB0	MC
7	6	5	4	3	2	1	0
BC	LPKT	CRS	COL	CRCR	FAER	LBK	PRX

(2) RXpkt.byte_count (バイト・カウント)

受信パケット中のデスティネーション・アドレスの1バイト目からフレーム・チェック・シーケンス (FCS) の4バイト目までの長さ (バイト・カウント) を示します。

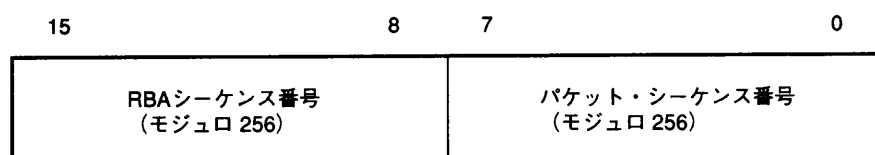
(3) RXpkt.pkt_ptr0, 1 (パケット・ポインタ)

パケットをRBA内に配置する32ビットのポインタです。μPD72932はCRBA0, 1レジスタの内容をこのフィールドへ書き込みます。

(4) RXpkt.seq_no (シーケンス番号)

このフィールドは、受信データの格納に使用した RBA とバッファされたパケットを順に並べるための 2 つの 8 ビット・カウンタ (256 ビット) で構成されています。これらのカウンタはシステムが RBA の処理完了を決定する際に利用します。シーケンス番号によってシステムは、特定の RBA 内で処理されたパケット数を記録することができます。パケット数を記録するシーケンス番号には RBA シーケンス番号とパケット・シーケンス番号の 2 つがあります。パケットをメモリへバッファするとき、μPD72932 は RBA 内の全パケットに対して 1 つの RBA シーケンス番号を維持し、RBA 内の連続したパケットにはパケット番号を順に付けます。次の RBA を使用する際には、μPD72932 が RBA シーケンス番号をインクリメントし、パケット・シーケンス番号をクリアします。リード RRA コマンドがコマンド・レジスタで発行されたときは RBA のシーケンス・カウンタはインクリメントされません。受信シーケンス番号のフォーマットを図 4-7 に示します。ハードウェア・リセットを行うか、または受信シーケンス・カウンタ・レジスタに “0” を書き込むと、このカウンタはリセットされます (5.3.9 参照)。

図 4-7 受信シーケンス番号のフォーマット



(5) RXpkt.link (受信リンク・フィールド)

次の受信ディスクリプタのロケーションを指定する 15 ビットのポインタ (A15-A1) です。このフィールドの LSB はエンド・オブ・リスト (EOL) ビットであり、リスト内の最後のディスクリプタに対してセットします (システムによって初期化します)。

(6) RXpkt.in_use (イン・ユース・フィールド)

このフィールドはディスクリプタのオーナーシップを示すために、システムと μPD72932 間のハンドシェイク機能を供給します。システムが μPD72932 に対してディスクリプタの使用許可を与えるときは、フィールドに “0” ではない値を書き込みます。μPD72932 がディスクリプタの処理を終えたとき (CRDA レジスタが次の受信ディスクリプタをポイントしたとき) には、フィールドをすべて “0” に設定します。一般に、μPD72932 は RDA にステータスおよび制御情報を書き込んだあと制御を解きます。しかし、μPD72932 がリスト内の最後のディスクリプタに到達した場合、システムがディスクリプタを追加するまではディスクリプタのオーナーシップを維持します (このフィールドへの “0” の書き込みは行いません)。μPD72932 のこのフィールドへの書き込みについての詳細は、4.4.6(1) を参照してください。受信パケット・ディスクリプタのフォーマットを図 4-5 に示します。

4.4.4 受信バッファ管理の初期化

μPD72932 がパケットの受信と書き込みを開始する前に、メモリ内の受信リソース・エリア、受信ディスクリプタ・エリア、受信バッファ・エリア (RRA, RDA, RBA) に対応する μPD72932 レジスタを、正しく初期化しておかなければなりません。このセクションではこの初期化について説明します。

(1) ディスクリプタ・ページの初期化

μPD72932が使用するディスクリプタ・エリア (RRA, RDA, および TDA) はすべて、32 k (ワード) ページか16 k (ロング・ワード) ページまでの範囲で設定できます。このページは上位16ビット・アドレス値を URDA および URRRA の各レジスタにロードすれば、32ビット・アドレス範囲内のどこにでも配置することができます。

(2) RRA の初期化

RRA の初期化は、4 つの μPD72932 の RRA レジスタをロードし、リソース・ディスクリプタ情報をメモリに書き込むことにより実行します。

次の値の RRA レジスタをロードします。

(a) リソース・スタート・エリア (RSA) レジスタ

RRA の開始点の下位16ビット・アドレスを RSA にロードします。

(b) リソース・エンド・エリア (REA) レジスタ

RRA の終わりの下位16ビット・アドレスを REA にロードします。最後の RXsrc.buff_ptr0 フィールドに、16ビット・モードなら 4 ワード、32ビット・モードなら 4 ロング・ワードを加えたアドレスを RRA の終わりとして定義します (図 4-3 参照)。

(c) リソース・リード・ポインタ (RRP) レジスタ

μPD72932が読み出す最初のリソース・ディスクリプタの下位16ビット・アドレスを RRP にロードします。

(d) リソース・ライト・ポインタ (RWP) レジスタ

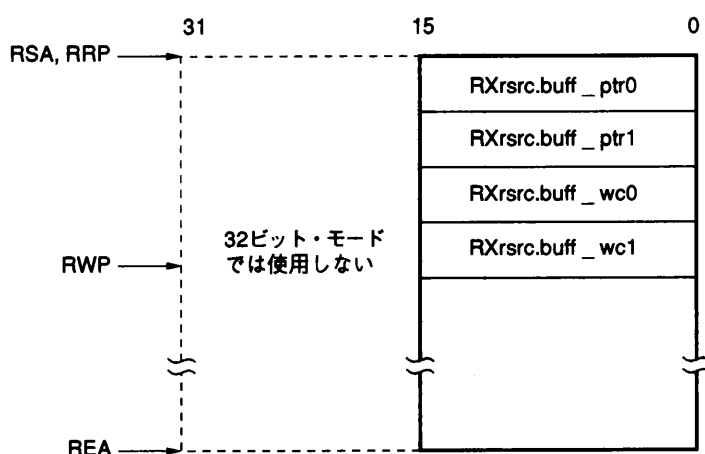
システムがリソース・ディスクリプタを設定する次の空きロケーションの下位16ビット・アドレスを RWP にロードします。

備考 RWP レジスタは (1) RRA ディスクリプタのうちのいずれかの RXsrc.buff_ptr0 フィールド、(2) RSA レジスタが指すメモリ・アドレス (RRA の開始)、(3) REA レジスタが指すメモリ・アドレス (RRA の終わり) のいずれかのみをポイントしていなければなりません。RWP=RRP のチェックは RRA ディスクリプタを読み出したあとに行います。RRA フェッチ中には RWP=RRP のチェックは行いません。RWP を上記のいずれかの値に設定しないと、RWP=RRP のチェックは正確に行われません。

すべての RRA レジスタは URRRA レジスタとともに、32ビット・アドレスを生成します。

システムが RRA に書き込むリソース・ディスクリプタは、(1) RXsrc.buff_ptr0, (2) RXsrc.buff_ptr1, (3) RXsrc.buff_wc0 および (4) RXsrc.buff_wc1 の 4 つのフィールドから成ります。フィールドは連続していなければならない、図 4-8 に示す順序で書き込みます。ディスクリプタの "0" および "1" は、バッファ・ポインタとワード・カウントの下位ワードおよび上位ワードを示しています。最初の 2 つのフィールドは受信バッファ・エリア (RBA) の開始位置を32ビットで示し、次の 2 つのフィールドは RBA が占有する16ビット・ワード数を定義します。バッファ・ポインタとワード・カウントには 2 つの制限事項がありますので注意してください。まず、32ビット・モードでは、μPD72932は常にロング・ワードを書き込むため、RXsrc.buff_wc0 に偶数カウントを書き込まなければなりません。さらに、バッファ・ポインタは16ビット・モード (A0=0) ではワードの境界を、32ビット・モード (A0, A1=0, 0) ではロング・ワードの境界を指し示していなければなりません。また、ディスクリプタは 4.3 で述べたとおり、RRA に正しく配列されていなければなりません。

図 4-8 RRAの初期化



RRA を構成したあとで、RRA のリード・コマンドを発行する（コマンド・レジスタの RRRRA ビットをセットする）と、μPD72932が1回のブロック転送で RRP によって示される RRA ディスクリプタを読み出し、次のレジスタにロードします（レジスタ名については 5.2 参照）。

- CRBA0レジスタ← RXsrc.buff_ptr0
- CRBA1レジスタ← RXsrc.buff_ptr1
- RBWC0レジスタ← RXsrc.buff_wc0
- RBWC1レジスタ← RXsrc.buff_wc1

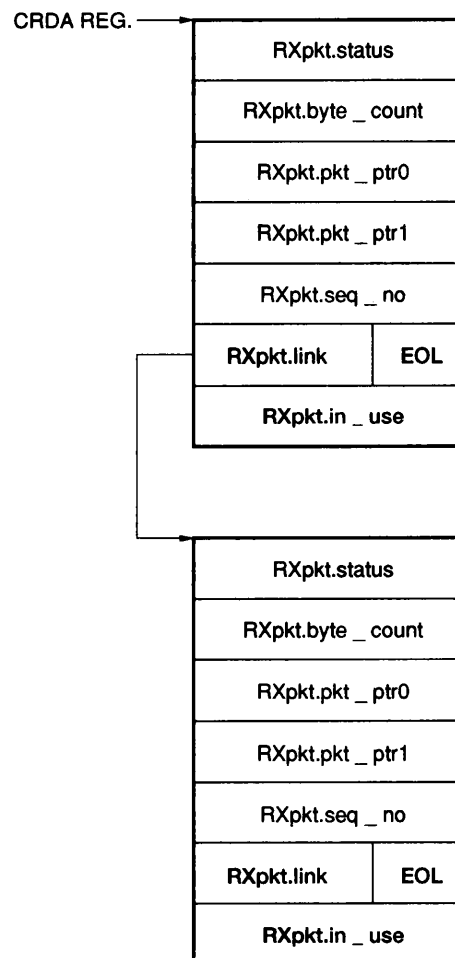
コマンドが完了すると、コマンド・レジスタの RRRRA ビットが“0” にリセットされます。通常このコマンドは初期化時にのみ発行してください。それ以外のときは常に、μPD72932が RBA の使用を終えると自動的に次の RRA を読み出します。

(3) RDA の初期化

システムは RDA の初期化として、図 4-9 に太字で示す RXpkt.link フィールド、EOL ビット、RXpkt.in_use フィールドにそれぞれ適切な値を書き込み、その後受信イネーブルする直前に上位受信ディスクリプタ・アドレス（URDA）レジスタとカレント受信ディスクリプタ・アドレス（CRDA）レジスタをセットします。

まずネットワークから複数のパケットを受信するために、15ビットの RXpkt.link フィールドに次のディスクリプタのロケーション（RXpkt.status）の下位15ビット（A15-A1、A0は0に固定）を書き込んでそれぞれの受信パケット・ディスクリプタをリンクしなければなりません。RXpkt.link フィールドの LSB は、エンド・オブ・リスト（EOL）ビットであり、ディスクリプタ・リストの終わりを示すために使用します。ディスクリプタ・リスト中の最後のディスクリプタの EOL は 1 に、それより前のディスクリプタの EOL は 0 にしてください。RXpkt.in_use フィールドは、μPD72932がディスクリプタを使用しているかどうかを示します。初期化時およびディスクリプタ使用が可能になったときに、システムで“0”でない値を書き込んでください。μPD72932がディスクリプタを使用し終えると、RXpkt.in_use フィールドに“0”を書き込みます。RDA の初期化が完了したあと受信イネーブルする直前に、最初の RXpkt.status フィールドの下位16ビット・アドレスを CRDA レジスタにロードし、上位16ビット・アドレスを URDA レジスタにロードして、μPD72932が最初のディスクリプタで受信を開始できるようにします。リンクされた2つのディスクリプタの例を図 4-9 に示します。太字で示すフィールド以外のフィールドは、パケットを受け入れたあとで μPD72932が書き込みます。また、RXpkt.in_use フィールドは、最初にシステムが書き込んだあと μPD72932が変更します。4.3 で述べたとおり、ディスクリプタの配列を適切に行ってください。

図 4-9 RDAの初期化例



(4) RBA の下限領域の初期化

受信しうる最大サイズの packets に相当する値 (16ビット・ワード数) をエンド・オブ・バッファ・カウンタ (EOBC) レジスタにロードすれば, RBA に “仮のボトム値” を設定することができます。これが RBA の下限境界になります。パケット受信完了後, リメイニング・バッファ・ワード・カウンタ (RBWC0, 1) レジスタが EOBC レジスタを下回ったとき, μPD72932 は次の packets を別の RBA に書き込みます。μPD72932 は 1 つの packets を複数の RBA に分散してバッファすることはありません。32ビット・モードでは, μPD72932 は RBWC0, 1 レジスタと EOBC レジスタの比較を正しく行うために, 常にアドレスの最下位 (A1) をロウ・レベルに保持する必要があることに注意してください。

ハードウェア・リセット後, EOBC レジスタは自動的に IEEE 802.3 で規定された最大パケット・サイズである, 2F8H (760ワード, または 1520バイト) に初期化されます。32ビット・モードで μPD72932 を使用する場 合, これが EOBC の推奨パケット・サイズになります。1518バイトはロング・ワード (32ビット) 境界でないため, EOBC は 759ワード (1518バイト) ではなく 760ワード (1520バイト) のデフォルト・サイズを取ります (4.4.2(1) 参照)。16ビット・モードで使用する場合は, 1518バイトはワード (16ビット) 境界なので 759ワード (1518バイト) にセットします。

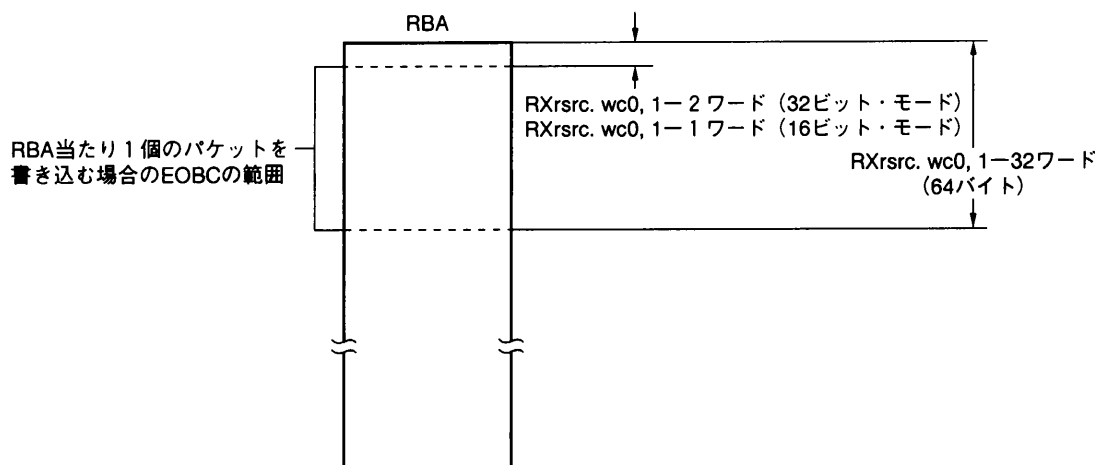
ソフトウェアの処理効率を向上させるため, 1 つの RBA 当たり 1 個の packets をバッファするのが望ましいことがあります。このとき, EOBC とバッファ・サイズを正しくセットすることが大切です。その際の EOBC の設定は, 32ビット・モードでは最低 4 バイト, 16ビット・モードでは 2 バイトをバッファ・サイズから引いた値にセットしてください。32ビット・モードでは EOBC を 760ワード (1520バイト), バッファ・サイズを 762

ワード (1524バイト) にセットしてください。16ビット・モードでは、EOBC を759ワード (1518バイト)、バッファ・サイズを760ワード (1520バイト) にセットしてください。

バッファ・サイズには制限がありませんが、EOBC が32ビット・モードで2ワード、16ビット・モードで1ワードのバッファ・サイズ以下だと、そのRBAには1個のケットだけしかバッファされません。

- 備考 1.** バッファ・サイズを、32ビット・モードで760ワード (1520バイト) に、また16ビット・モードで759ワード (1518バイト) にセットすると、ほとんどのオーバサイズ・ケットを除去することができます。この32ビットと16ビットの両モードでは、EOBC は758ワード (1516バイト) になります。この構成の場合、ケットが32ビット・モードで1520バイト以上、16ビット・モードで1518バイト以上あるケットはすべて、バッファをオーバフローしバッファされません。ケットのオーバフローが起きると、受信バッファ・エリアの超過割り込み (**5.3.6 割り込みステータス・レジスタのRBAE**) が発生します。
- 2.** バッファ当たり1個のケットをバッファする場合、備考1の値を使用してください。しかし、イーサネットの最小規格サイズは64バイトなので、バッファ・サイズから64バイトを引いた値までEOBCをセットでき、バッファ当たり1個のケットをバッファできます。図4-10はこの“範囲”を示しています。

図4-10 イーサネットにおいてRBAに1個のケットを書き込む場合のEOBCの設定



EOBCの範囲 = (RXsrc. wc0, 1 - 2からRXsrc. wc0, 1 - 32)

(5) 1つのRBAに複数のケットを格納する場合

ケットの受信後、μPD72932はステータス情報を受信バッファ・エリア (RBA) に追加します。1つのRBAに複数のケットを格納する場合、メモリの重ね書きが生じる危険性があります。その原因と解決方法を次に説明します。

- 1つのRBAに1つのケットを受信するアプリケーション・システムでは問題ありません。
- 1つのRBAに複数のケットを受信するメモリ・アーキテクチャでμPD72932を使用するアプリケーション・システムは、次の指示に従ってこの問題を解決する必要があります。

ケットの受信時、μPD72932はネットワークから受信したケットを受信FIFOにバッファします。受信バ

ケットがFIFO スレッシュホールドに達するとすぐに、μPD72932はバスを要求し、データをRBA にバッファします。このプロセスはパケットの終わりまで繰り返されます。

μPD72932はパケットの終端で、CRC 情報を書き込んだ直後に1ワードまたは2ワードのステータス情報(16ビット・モードでは1ワード、32ビット・モードでは2ワード)を書き込みます。μPD72932のカレント受信バッファ・アドレス (CRBA0, 1) レジスタおよびリメイニング・バッファ・ワード・カウント (RBWC0, 1) レジスタは次のようになります。

- ① バッファの最後に達した場合 (RBWC0, 1 < EOBC), μPD72932 は受信リソース・エリア (RRA) からの RXsrc.buff_ptr0, 1 および RXsrc.buff_wc0, 1 を CRBA0, 1 および RBWC0, 1 へそれぞれロードします。
- ② バッファの最後に達していない場合 (RBWC0, 1 ≥ EOBC), μPD72932は CRBA0, 1 をインクリメントし、RBWC0, 1 をデクリメントして、バッファの最後に達するまでパケットをRBA にバッファし続けます。

②の場合、μPD72932はパケットのデータに加え、ステータス情報をRBA に書き込んでCRBA0, 1 をインクリメントしますが、RBWC0, 1 はこのステータス情報分をデクリメントしません。したがって、バッファされるパケット1つについてRBWC0, 1 は本来よりも1ないし2ワード少なくデクリメントされることになります。このため、1つのRBA に多くのパケットがバッファされるほど、RBA 外のメモリに対して重ね書きをする危険性がしだいに高くなります。RBA 外のメモリに重ね書きをしないようにするために、EOBC の設定を次のように変更してください。

(a) 16ビット・モードの場合

$$EOBC = N + 759$$

N はRBA に収まるイーサネット・パケットの最小数 (N=RBA サイズ/64) です。

(b) 32ビット・モードの場合

$$EOBC = 2 \times N + 760$$

N は16ビット・モードの場合と同様です。

4.4.5 受信の開始

パケット受信を開始すると、μPD72932は前回のRXpkt.link フィールドを読み出し、そこにあるEOLビットが“1”かどうかをチェックします。EOL=1であると、μPD72932は前回の受信のあとに受信パケット・ディスクリプタが残っていなかったことを認識します。μPD72932は同じRXpkt.link フィールドを再び読み出し、前回の受信のあとシステムがこのフィールドを更新(EOL=0)したかどうかをチェックします。まだEOL=1であると、受信を停止します(リストへのディスクリプタの追加については4.5を参照してください)。EOL=0であると、μPD72932はカレント受信バッファ・アドレス (CRBA0, 1) レジスタで示されるアドレスからRBA へのパケットの格納を開始し、パケット受信が完了するまでこれを継続します。パケット受信と同時に、μPD72932はリメイニング・バッファ・ワード・カウント (RBWC0, 1) レジスタをデクリメントします。16ビット・モードの場合は受信データ1ワードをメモリに書き込むたびに1ずつ、32ビット・モードの場合は受信データ1ロング・ワードを書き込むたびに2ずつRBWC0, 1レジスタをデクリメントします。RBWC0, 1レジスタは受信終了時に、RBA 内に残っている空き領域のワード数を示します。

4.4.6 パケットの終了処理

受信終了時に、μPD72932はパケットの終了処理シーケンスに入り、受信エラーとパケット・サイズに基づいてパケットを受け入れるべきか拒否すべきかを決定します。受信終了時に、μPD72932は次のいずれかのシーケンスに入ります。

- 成功受信シーケンス
- ラント・パケットまたはエラーのあるパケットに対するバッファ・リカバリ

(1) 受信の成功

μPD72932がパケットの受信を完了した場合まず、カレント受信ディスクリプタ・アドレス (CRDA) レジスタが指す RDA 内のディスクリプタ情報の最初の 5 ワードを書き込みます。そして RXpkt.link フィールドを読み出して、CRDA レジスタを次の受信ディスクリプタへ送ります。μPD72932はこのフィールドの EOL ビットが“1”になっているかどうか同時にチェックします。EOL=1 の場合、それ以上ディスクリプタは使用できません。μPD72932はカレント RXpkt.link フィールドのアドレスを (仮のレジスタから) 復元し、割り込みステータス・レジスタに“受信ディスクリプタの枯渇”を表示します (ディスクリプタの追加方法については 4.4.7 を参照してください)。一方、EOL=0 の場合、次のディスクリプタ・アドレスへ CRDA レジスタが更新され、RXpkt.in_use フィールドをすべて“0”にリセットします。

μPD72932は 1 回のブロック転送で 7 ワード RDA ディスクリプタすべてにアクセスします。

μPD72932はさらに、RBA 内にスペースが残っているかどうかチェックします。リメイニング・バッファ・ワード・カウント (RBWC0, 1) レジスタとエンド・オブ・バッファ・カウント (EOBC) を比較します。RBWC が EOBC 以下であれば、最大サイズの packets を RBA の残りスペースに書き込むことはもうできないことになります。したがって、μPD72932が RRA から新たなリソース・ディスクリプタをフェッチし、そのレジスタに次に使用可能な RBA のポインタとワード・カウントをロードします。

(2) ラント・パケットまたはエラー・パケットに対するバッファ・リカバリ

ラント・パケット (64バイト以下のパケット) またはエラー付きパケットを受信し、しかも受信制御レジスタがこれらのパケットを受信しないように設定されている場合、μPD72932は受信前にテンポラリ受信バッファ・アドレス (TRBA0, 1) レジスタに退避させたバッファ・アドレスを CRBA0, 1 レジスタに戻し、テンポラリ・バッファ・ワード・カウント (TBWC0, 1) レジスタに退避させたワード・カウントを RBWC0, 1 レジスタに戻します。

4.4.7 オーバフロー状態

オーバフロー状態が発生すると、μPD72932は DMA 動作を中止してシステムから使用を許可されていないメモリへの書き込みを防止します。μPD72932は、受信リソースが空になったときに発生する 3 種類のオーバフロー状態を示すために、割り込みステータス・レジスタ (ISR) を使用します。システムは空になったリソースを補給しなければなりません。オーバフロー状態 (ディスクリプタ・リソースの枯渇 (RDE)、バッファ・リソースの枯渇 (RBE)、RBA の超過 (RBAE)) は割り込みステータス・レジスタに示されます。これらを以下に詳述します。

(1) ディスクリプタ・リソースの枯渇

μPD72932がリストの最後の受信ディスクリプタに到達したときに発生し、EOL=1を検出したことを意味します。受信を続けるためにはシステムはディスクリプタを追加しなければなりません。追加方法には、(a) リストにディスクリプタを追加する、(b) 別のリストを生成する、という2つの方法があります。

(a) ディスクリプタを追加する方法は最も簡単でよく用いられる方法です。この方法では、システムが新しいリストを生成すると、その先頭アドレスをRXpkt.link フィールドに書き込んでEOL=0に設定し、新しいリストを現在のリストにつなぎます。次の受信時にμPD72932が最後のRXpkt.link フィールドを再び読み取って、CRDA レジスタが次のディスクリプタを指すように更新します。

(b) 別のリストを生成するには、リストがつながっていないのでさらにステップをふまなければならない、また CRDA レジスタに新しいリストのRXpkt.link フィールドのアドレスをロードしなければなりません。

オーバフロー状態の間、μPD72932はディスクリプタのオーナーシップを維持し(RXpkt.in_use≠00H)、システムがリストにディスクリプタを追加するまで待ちます。システムがディスクリプタを追加し、新たなパケットの受信を開始すると、μPD72932はRXpkt.in_use フィールドに0000H を書き込みオーナーシップを解きます。

(2) バッファ・リソースの枯渇

リソース・リード・ポインタ(RRP)レジスタとリソース・ライト・ポインタ(RWP)レジスタが等しい(つまり、RRA ディスクリプタがすべて使用された状態)ことをμPD72932が検出したときに発生します。μPD72932が最後から二番目の受信バッファの使用を終え、最後のRRA ディスクリプタを読み出すと、割り込みステータス・レジスタのRBEビットが設定されます。実際には、μPD72932のリソースが完全に空になっているわけではありませんが、もうすぐリソースが空になることをシステムに対し早期に警告します。最後のRBAを使用したあと受信を継続するには、システムはRRA ディスクリプタを追加し、RWP レジスタを更新し、ISR内のRBEビットをクリアしなければなりません。このビットがクリアされるとRRAを再び読み取ります。

(3) RBA 限界オーバ

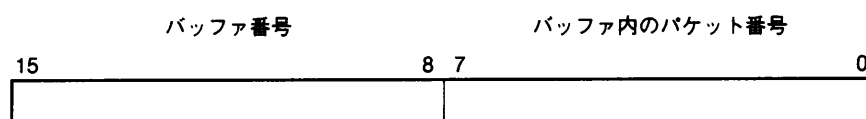
パケットがRBAの残りスペースにおさまらなくなるときに発生します。EOBCレジスタが受信可能な最大パケット以上の値にプログラムされていない場合にも発生することがあります。こうした状況になると、パケットは切り捨てられ、RRAが読み出されて別のRBAが生成されます。RBAの限界オーバは、受信バッファ・エリア超過(RBAE)の割り込みがセットされることにより示されます(5.3.6参照)。RDAは切り捨てられたパケット用には生成されず、そのバッファ空間は再度使用されません。このオーバフロー状態を修正するには、受信可能な最大パケットと等しいかそれ以上の値をEOBCレジスタにロードしなければなりません。4.4.2を参照してください。

(4) RBAE の処理

μPD72932でRBAEを処理する方法を次に説明します。通常、μPD72932はエラー・パケット(CRCエラー、フレーム配列エラー、ラント・パケットなど)を受け取ると、その該当パケットを受信するように設定されていないかぎり、パケットを廃棄し、バッファ・ポインタをそのパケットの受信前の位置に戻します。しかし、バッファの長さを越えるパケット(RBAE)の場合は、μPD72932はバッファ・ポインタを戻さず、バッファをそのままの状態にして新しいパケットを受信します。この動作自体は問題ではありませんが、問題はμPD72932がこのRBAEに対してRDAを作成しないということです。つまり、対応するRDAを持たないバッファ(1つのバッファに1つのパケットを格納する場合)やバッファの一部(1つのバッファに複数のパケットを格納する場合)が生じてしまいます。このようなバッファは見つけ出して割り当て解除するかRRAに戻して、μPD72932

が再び使用できるようにしなければなりません。

失われたバッファを発見する方法としては、RDAのシーケンス番号を使用する方法があります。RDAのシーケンス番号フィールドは16ビット長のフィールドで、次に示すように2つのカウンタで構成されています。



上位8ビットはバッファ番号のカウンタ値を表し、下位8ビットはそのバッファ内のパケット番号のカウンタ値を表します。カウンタはそれぞれ0から255までカウントして0に戻ります。したがって、シーケンス番号は次のように変化します。

0000H…バッファ0のパケット0
 0001H…バッファ0のパケット1
 0002H…バッファ0のパケット2
 0100H…バッファ1のパケット0
 0101H…バッファ1のパケット1
 0102H…バッファ1のパケット2
 0103H…バッファ1のパケット3
 ↓
 0200H…バッファ2のパケット0

ソフトウェア・ドライバは、バッファ番号がたとえば0から1へ変化したことを確認したとき、バッファ0が完全に処理されたと認識し、そのバッファの割り当てを解除することができます。バッファ0のパケット3でRBAEが発生した場合、このバッファ・シーケンス番号は同じようにインクリメントされる(0から1へ変化する)ため、問題は生じません。

1つのバッファに複数のパケットをバッファリングするとき、RBAEについて考慮しなければならないのは、RBAEパケットがバッファ内の最初で唯一のパケットである場合です。この場合、そのバッファを指し示すRDAは存在せず、シーケンス番号は次のようになります。

0000H…バッファ0のパケット0
 0001H…バッファ0のパケット1
 0002H…バッファ0のパケット2
 0200H…バッファ2のパケット0
 0201H…バッファ2のパケット1
 0202H…バッファ2のパケット2
 0203H…バッファ2のパケット3
 ↓
 0300H…バッファ3のパケット0

この場合、バッファ・シーケンス番号が0から2へ飛んでいます。これは、RBAEがバッファ1のパケット0で発生した場合に生じます。RBAEバッファ(バッファ1)を割り当て解除するには、2つのことを行ってください。まず、RBAEの発生を見つけてください。これはバッファ・シーケンス番号2から前のバッファ・シーケンス番号0を引いて、その結果の値が1より大きい(2-0=2)ことから導き出せます。このことは、バッファ

2 とバッファ 0 の間に RBAE の発生したバッファが存在し、そのバッファを指し示す RDA がないことを意味します。ソフトウェアでこの現象を確認したら、次に RBAE バッファ (バッファ 1) のアドレスまたはポインタを見つけて、バッファの割り当てを解除するか RRA に戻してください。

この方法は μPD72932 が 1 バッファあたり 1 パケットだけをバッファリングする場合にも容易に応用することができます。その場合、RBAE が発生するたびに対応する RDA がないバッファが生成されます。この場合も、シーケンス番号を引き算することによってそのバッファを見つけることができます。その後の手順は前述の場合と同じです。

RBAE 割り込みを使用すれば、チェックする RDA 1 つ 1 つについて引き算を行わなくても RBAE の発生がわかります。割り込みが発生したら、ただちに RBAE をクリアしてシーケンス番号のチェックが必要なことを知らせるフラグをセットします。RBAE バッファが見つかったらフラグをクリアします。

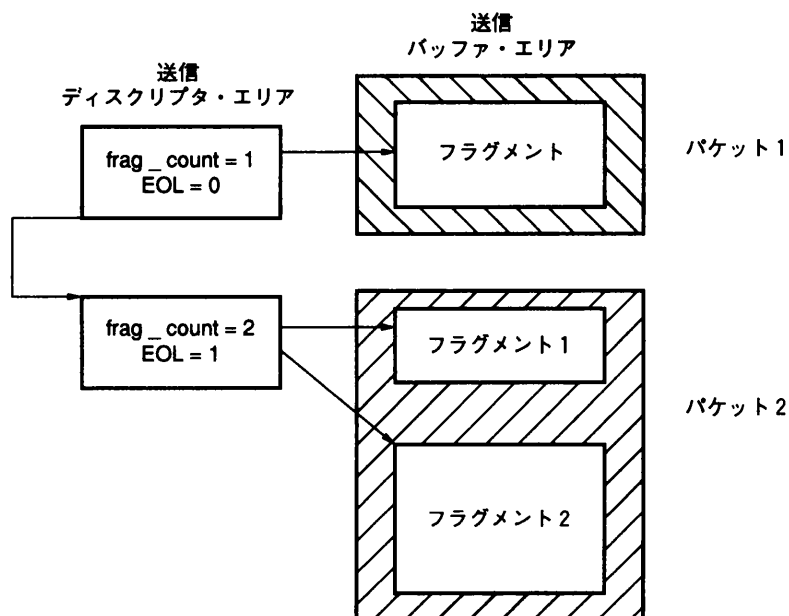
RBAE を処理するとき考慮しなければならないもう 1 つの問題として、複数の RBAE バッファが連続する場合があります。複数の RBAE バッファが連続している場合、同じ手順を使用できますが、シーケンス番号を引き算するとき結果の値が 2 よりも大きくなります。また、RBAE バッファを見つける前に複数の RBAE 割り込みが発生する可能性が高いため、発生した RBAE 割り込みの数をカウントすることが必要になります。したがって、RBAE 割り込みが発生したらフラグを設定するのではなく、カウンタをインクリメントして割り込みをクリアします。そして、RBAE バッファが見つかったらカウンタをデクリメントし、カウンタ値が 0 になるまで繰り返します。

割り込みがカウントされる前に 2 つの RBAE 割り込みが発生する可能性がありますので注意してください。この場合、RBAE バッファを処理するルーチンは、連続する複数の RBAE バッファの発生をチェックする必要があり、さらに 1 つおきのバッファでの RBAE バッファの発生をもチェックする必要があります (どちらの場合も RBAE バッファの発生が 1 回としてカウントされた可能性があります)。ここで必要となるチェックの量は、割り込み処理ルーチンがどの程度頻繁に応答するか、ISR をどの程度速くポーリングするかによって決まります。

4.5 送信バッファ管理

送信バッファ管理においては、パケット送信のためにメモリ内の 2 つのエリア、送信ディスクリプタ・エリア (TDA) と送信バッファ・エリア (TBA) を使用します (図 4-11)。システム・ソフトウェアが送信コマンド (CR 内の TXP=1) をセットすると、送信を開始します。送信中、μPD72932 は TDA から各フィールドをフェッチし、適切なレジスタにロードし、TBA のデータを送信します。送信が完了するとステータス情報を TDA に書き込みます。1 個の送信コマンドにより 1 つのパケットを送信できますが、複数のディスクリプタをリンクすることによって、グループ (複数のパケット) でも送信できます。

図 4-11 送信バッファ管理の概要



4.5.1 送信ディスクリプタ・エリア (TDA)

TDA には、ステータスおよびコンフィギュレーションをリード/ライトするためにシステムが生成したディスクリプタが含まれています。各ディスクリプタは1つのパケットに対応し、以下の16ビット・フィールドからなります。

(1) TXpkt.status (送信ステータス)

μPD72932によって書き込まれ、送信パケットのステータスを設定します。

各送信の終わりに、送信制御レジスタ (TCR) のステータス・ビット (10-0) と送信中のコリジョン回数を書き込みます (図 4-12 の “res” は予約ビットを示します)。NC4-NC0ビット (NC4がMSB) はコリジョン数を示します。TCR については 5.3.4 を参照してください。

図 4-12 送信ステータスのフォーマット

15	14	13	12	11	10	9	8
NC4	NC3	NC2	NC1	NC0	EXD	DEF	NCRS
7	6	5	4	3	2	1	0
CRSL	EXC	OWC	res	PMB	FU	BCM	PTX

(2) TXpkt.config (送信コンフィギュレーション)

μPD72932は TXpkt.config フィールドで送信の前にいずれかの送信モードにプログラムできます。各送信の始めに、μPD72932はこのフィールドを読み出し、PINTR, POWC, CRCI, EXDIS ビットを送信制御レジスタ (TCR) にロードします。図 4-13 で示すように、TCR のコンフィギュレーション・ビットは TXpkt.config フィールドのビットと直接対応します。TCR については 5.3.4 を参照してください。

図 4-13 送信コンフィギュレーションのフォーマット

15	14	13	12	11	10	9	8
PINTR	POWC	CRCI	EXDIS	×	×	×	×
7	6	5	4	3	2	1	0
×	×	×	×	×	×	×	×

備考 ⅓ : don't care

(3) TXpkt.pkt_size (パケット・サイズ)

パケット全体のバイト・カウントを書き込みます。

(4) TXpkt.frag_conut (フラグ・カウント)

パケットを分割するフラグメントの数を書き込みます。

(5) TXpkt.frag_ptr0, 1 (フラグ・ポインタ)

送信バッファ・エリア (TBA) 内で送信されるパケット・フラグメントの位置を示す32ビット・ポインタ。

このポインタはいずれのバイト配列に対しても制限はありません。

(6) TXpkt.frag_size (フラグ・サイズ)

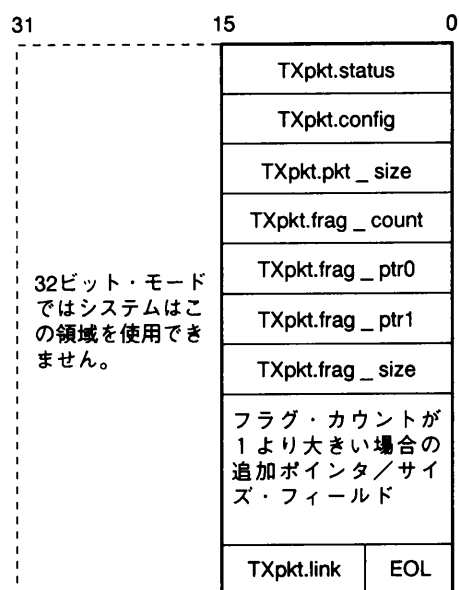
パケット・フラグメントのバイト・カウントが書き込まれます。フラグメント・サイズはシステムの DMA 転送能力に左右されます。最小 1 バイトをセットすることができますが、できるだけ大きなサイズを設定してください。

(7) TXpkt.link (送信リンク・フィールド)

次の TDA ディスクリプタに対する15ビット・ポインタ (A15-A1) を書き込みます。このフィールドの LSB であるエンド・オブ・リスト (EOL) ビットが "1" に設定されていればリスト内の最後のディスクリプタであることを示します。ディスクリプタがリンクされているときは、一個の送信コマンドによって続けざまにパケットを送信できます。

パケット・データは連続している必要はなく、メモリ内のいくつかの位置 (フラグメント) に分散することができます。この場合、TXpkt.frag_count フィールドは 1 よりも大きく、各フラグメントに対応する追加の TXpkt.frag_ptr0, 1 フィールドと TXpkt.frag_size フィールドが使用されます。ディスクリプタ・フォーマットを図 4-14 に示します。32ビット・モードでは上位ワード (D31-D16) は使用できないことに注意してください。

図 4-14 送信ディスクリプタのフォーマット



4.5.2 送信バッファ・エリア (TBA)

TBA には TDA のディスクリプタによって定義されるパケットのフラグメントを書き込みます。パケットは1つのフラグメントからでも複数のフラグメントからでも構成できますが、そのいずれになるかは TDA ディスクリプタのフラグメント・カウントによって決まります。フラグメントもまた、32ビットの全アドレス範囲のどこにでも設定することができ、いずれのバイトの境界へも配列することができます。最初の送信データのロケーションが奇数アドレスから始まる場合には、16ビット・モードならそれに対応するワードの境界で、32ビット・モードならロング・ワードの境界を選んで、μPD72932が自動的にデータ読み出しを始めます。奇数バイト配列のフラグメントで始まった場合は FIFO に書き込まれた無関係のバイトは無視されます。図 4-11 にシングルおよび複数のフラグメント・パケットに対する TDA と TBA 間の関係を示します。

4.5.3 送信準備

送信コマンド（コマンド・レジスタの TXP ビット）をセットする前に、TDA ディスクリプタとカレント送信ディスクリプタ・アドレス（CTDA）レジスタのすべてのフィールドを初期化しておかなければなりません。複数のパケットを送信キューに設定する場合、ディスクリプタを TXpkt.link フィールドでリンクしなければなりません。最後のディスクリプタでは EOL=1、その他のディスクリプタでは EOL=0 になっていなければなりません。送信を開始するには、システムが最初の TXpkt.status フィールドのアドレスを CTDA レジスタにロードしなければなりません。アドレスの上位16ビットは上位送信ディスクリプタ（UTDA）レジスタにロードされることに注意してください。ユーザは以下の送信初期化を行います。

- (1) TDA の初期化
- (2) CTDA レジスタに最初の送信ディスクリプタのアドレスをロードする
- (3) 送信コマンドをセットする

送信されるパケットのソース・アドレスが CAM になれば、TXpkt.status フィールドのモニタ・パケット不良（PMB）ビットがセットされることに注意してください（5.3.4 参照）。

(1) 送信プロセス

送信コマンド(コマンド・レジスタの TXP=1)がセットされると、μPD72932は TDA ディスクリプタをフェッチし、適切なレジスタ(下記参照)にロードし、送信を開始します(レジスタ名については 5.2 を参照してください)。

- TCR ← TXpkt.config
- TPS ← TXpkt.pkt_size
- TFC ← TXpkt.frag_count
- TSA0 ← TXpkt.frag_ptr0
- TSA1 ← TXpkt.frag_ptr1
- TFS ← TXpkt.frag_size
- CTDA ← TXpkt.link

CTDA はすべてのフラグメントの読み出し、送信が完了したあとでロードされます。送信停止コマンドがセット(コマンド・レジスタの HXT ビットがセット)されると、CTDA レジスタはロードされません。

送信中、μPD72932は TDA のポインタ、サイズ情報を読み出し、TBA のデータを FIFO に転送します。TXpkt.frag_count が 1 よりも大きければ、フラグメント送信のあと、μPD72932は次の TXpkt.frag_ptr0, 1 フィールドと TXpkt.frag_size フィールドをフェッチし、引き続き次のフラグメントを送信します。パケットの全フラグメントが送信されるまで、このプロセスは継続します。パケット送信が終了すると、ステータスが TXpkt.status フィールドに書き込まれます。そのあと μPD72932が TXpkt.link フィールドを読み出して、EOL=0 かどうかをチェックします。“0”であれば次のディスクリプタをフェッチし、次のパケットを送信します。EOL=1 であれば、μPD72932は“送信完了”の表示を割り込みステータス・レジスタに生成し、コマンド・レジスタの TXP ビットをリセットします。

μPD72932はテンポラリ送信ディスクリプタ・アドレス(TTDA)レジスタに CTDA レジスタをコピーし、コリジョンが発生すると TDA 内のポインタをリカバリしてパケットを15回まで再送信します。

μPD72932が送信プロセスのどの状態であるかによって、TDA 内の 6 つ、3 つまたは 2 つのフィールドのうちのいずれかのフィールド・アクセスを 1 回のブロック操作で転送します。最初のフラグメントに対しては TXpkt.config から TXpkt.frag_size までの 6 フィールド(6 アクセス)が読み出されます。もし次のフラグメントがあれば、TXpkt.frag_ptr0 から TXpkt.frag_size までの 3 フィールド(3 アクセス)が読み出されます。送信終了時には、ステータス情報が TXpkt.status に書き込まれ、TXpkt.link フィールドが読み出されます(2 アクセス)。

(2) 送信終了

以下の 2 つの状況になると送信を停止します。1 つは、μPD72932が TDA 内のディスクリプタ・リストをすべて正常に送信し、EOL=1 を検出すると停止する場合です。もう 1 つは、送信エラーによって送信がアボートされる場合です。FIFO アンダーラン、バイト・カウント不一致、過剰衝突、または過剰遅延(もしイネーブルされていれば)などのエラーが発生すると、送信は停止します。CTDA レジスタは最後に送信されたパケットを指します。また、コマンド・レジスタの HTX ビットをセットすると、μPD72932が TXpkt.status フィールドに書き込んだあと、送信を停止します。

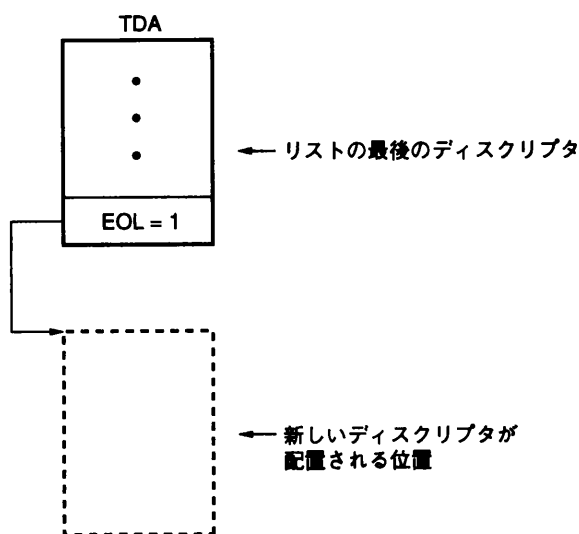
4.5.4 ダイナミックな TDA ディスクリプタ追加

送信中に μPD72932 を停止せずに、ディスクリプタをダイナミックに追加することができます。また、次の規則に従えば、(送信エラーがなければ) 新しく追加されたディスクリプタを含んだ完全なリストを送信することもできます。その規則とは、最後の TXpkt.link フィールドが次のディスクリプタ追加位置を指していなければならない、ということです (以下の3つのステップと図 4-15 を参照してください)。ディスクリプタ追加の手順は以下のステップとなります。

- (1) TXpkt.link が次の空きディスクリプタ位置を指し、EOL ビットが "1" に設定されている新しいディスクリプタを生成します。
- (2) 既存の最後のディスクリプタの EOL ビットを "0" にリセットします。
- (3) 送信コマンド (コマンド・レジスタの TXP ビットをセットする) を再びセットします。

ステップ (3) でリスト内のパケットはすべて送信されます。μPD72932 が送信中の場合、(3) の送信コマンドは何の影響も与えず、EOL=1 が検出されるまで送信を継続します。送信がちょうど終了したところであれば、最後に停止した場所から送信を継続します。

図 4-15 最終リンク・フィールドの初期化



5. μPD72932レジスタ

μPD72932は2組のレジスタを内蔵しています。ステータス/制御レジスタとCAMメモリ・セルです。ステータス/制御レジスタは、μPD72932の動作を規定、制御、モニタするために使用します。このレジスタは直接アドレス可能なレジスタで、システム・メモリ・スペース内の(RA5-RA0アドレス端子によって選択される)64の連続したアドレス・ロケーションを占有します。合計64のステータス/制御レジスタがあり、以下のカテゴリーに分類されます。

(1) ユーザ・レジスタ

ユーザがμPD72932の動作を構成、制御、モニタするためにアクセスします。ユーザがアクセスする必要があるレジスタはこれだけです。図5-3にプログラマズ・モデルを示し、表5-1には各レジスタの一覧表を示します。

(2) 内部使用レジスタ

通常の動作でμPD72932が使用するレジスタ(表5-2)で、ユーザはアクセスできません。

(3) 工場テスト用レジスタ

工場専用のレジスタ(表5-3)で、ユーザがアクセスしてはなりません。通常の動作で、これらのレジスタにアクセスすると、μPD72932が誤動作することがあります。

5.1 CAMユニット

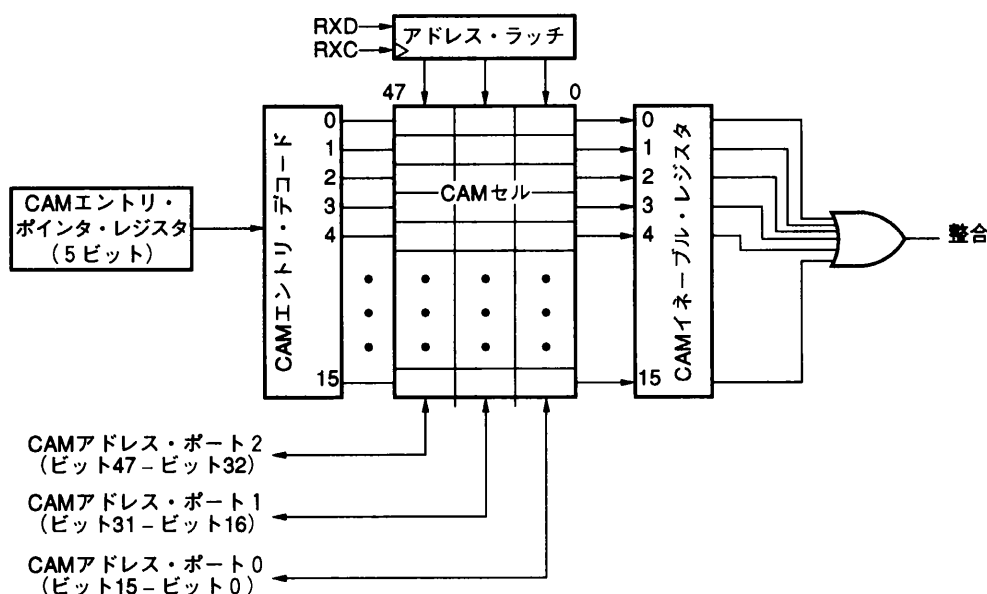
CAMユニット・メモリ・セルは、システム・メモリのCAMディスクリプタ・エリアをプログラムし、LCAMコマンドをセットする(制御レジスタのLCAMビットをセットする)ことによって、間接的にアクセスされます。CAMセルはシステム・メモリ空間内を占有しないため、RA5-RA0のアドレス端子でアクセスすることはできません。CAM制御レジスタはユーザ・レジスタ・セットの一部なので、LCAMコマンドをセットする前に初期化しておかなければなりません(5.3.10参照)。

5.1.1 CAMの構成

コンテンツ・アドレス・メモリ(CAM)は16個の48ビット・エントリからなり、ネットワーク・パケットの完全なアドレス・フィルタリングを実行します(図5-1)。各エントリは、ユーザ・プログラムが可能な48ビットのデスティネーション・アドレスに対応しており、物理アドレスまたはマルチキャスト・アドレスのどのような組み合わせでも格納することができます。各エントリは、CAMアドレス・ポート(CAP2, CAP1およびCAP0)からリード可能な、3つの16ビットCAMセルに分割することができ、CAP0はデスティネーション・アドレスの最下位16ビットに、CAP2は最上位ビットに対応します。CAMにアクセスするには2つのステップの処理が必要です。まず、CAMエントリ・ポインタをロードして16のエントリのうちの一つを指すようにします。ついで、各CAMアドレス・ポートにアクセスして、CAMセルを選択します。CAMイネーブル・レジスタで16のユーザ・プログラマブルCAMエントリをマスクすることができます(5.3.10参照)。

備考 ブロードキャスト・パケットを受信するために、受信制御レジスタのBRDビットをセットする必要がありますが、ブロードキャスト・アドレスをCAMにプログラムする必要はありません。BRDビットがセットされていても、CAMはアクティブのままになります。これは、CAMの物理アドレスに一致するパケットを受信しながら、同時にブロードキャスト・パケットの受信が可能なことを意味します。

図 5-1 CAMの構成



5.1.2 ロードCAMコマンド

μPD72932は受信中に比較的長い間CAMを使用するため、CAMセルへの書き込みはCAMディスクリプタ・エリアを作成し、LCAMコマンドを実行することによってのみ可能です。また、CAP2-CAP0の読み出しはμPD72932がソフトウェア・リセット状態にあるときのみ可能です。CDAは受信リソース・エリア(RRA)と同じ、64Kバイトのメモリ・ブロックにあり、CAMレジスタをロードするためのディスクリプタを格納します。各ディスクリプタは4つの16ビット・フィールドからなり、連続しています(図5-2)。32ビット・モードでは、上位ワードD31-D16は使用しません。最初のフィールドには、CAMのエントリ・ポインタにロードする値が格納されており、残りのフィールドは3つのCAMアドレス・ポートに対応しています(5.3.10参照)。さらに、最後のディスクリプタのあとにもう1つフィールドがあり、CAMイネーブル・レジスタのセットを行います。各CAMディスクリプタは、CAMディスクリプタ・ポインタ(CDP)レジスタでアドレス指定されます。

システムは、CDAを初期化したあと、LCAMコマンドを発行して、CDAから、CAMをロードするようにμPD72932をプログラムします。LCAMコマンドをセットする手順は以下のとおりです。

- (1) 上位受信リソース・アドレス(URRA)レジスタを初期化します。

CAMディスクリプタ・エリアが、受信リソース・エリアと同じ64Kバイト・エリアになければならないことに注意してください(5.3.9参照)。

- (2) CDAを上記のとおり初期化します。

- (3) CAMディスクリプタ・カウントをCAMディスクリプタ数で初期化します。このレジスタの下位5ビットだけを使用することに注意してください。その他のビットは任意の値でかまいません(5.3.10参照)。

- (4) 最初のディスクリプタをCDAにセットするためにCAMディスクリプタ・ポインタを初期化します。新しいLCAMコマンドをセットするたびに、このレジスタを再設定しなければなりません。

- (5) コマンド・レジスタでロードCAMコマンド(LCAM)をセットします(5.3.1参照)。

送信または受信が行われている場合は、CAMのDMA転送はそれらが完了するまで動作しません。μPD72932がLCAMコマンドを完了すると、CDPレジスタがCAMイネーブル・フィールドのあとの次のロケーションを指し、CDCが0になります。最後にμPD72932はコマンド・レジスタのLCAMビットをリセットし、ISRのロードCAM完了(LCD)ビットをセットします。

図 5-2 CAMディスクリプタのフォーマット

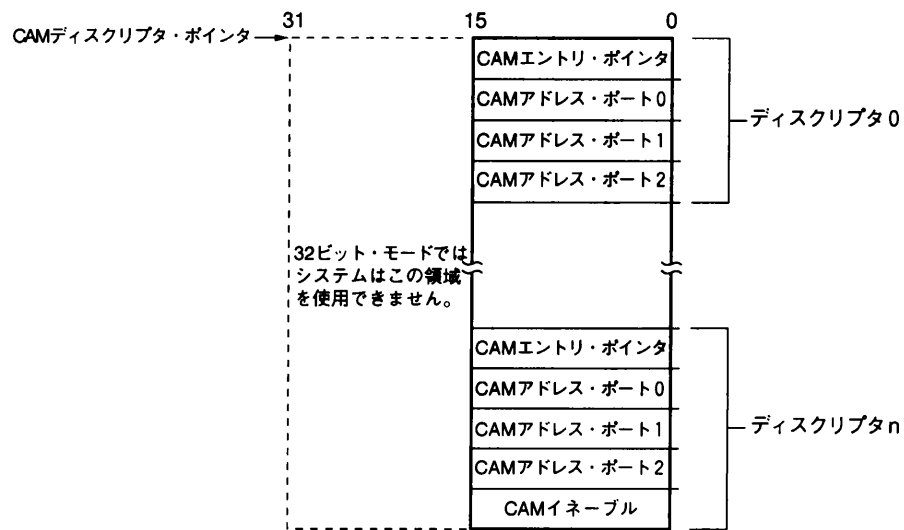


図5-3 レジスタ・プログラミング・モデル

	RA5 - RA0		15	0
ステータス/ 制御レジスタ	00H	コマンド・レジスタ	ステータス/コントロール・フィールド	
	01H	データ・コンフィギュレーション・レジスタ	コントロール・フィールド	
	02H	受信制御レジスタ	ステータス/コントロール・フィールド	
	03H	送信制御レジスタ	ステータス/コントロール・フィールド	
	04H	割り込みマスク・レジスタ	マスク・フィールド	
	05H	割り込みステータス・レジスタ	ステータス・フィールド	
	3FH	データ・コンフィギュレーション・レジスタ2	コントロール・フィールド	
送信レジスタ	06H	上位送信ディスクリプタ・アドレス・レジスタ	上位16ビット・アドレス・ベース	
	07H	カレント送信ディスクリプタ・アドレス・レジスタ	下位16ビット・アドレス・オフセット	
受信レジスタ	0DH	上位受信ディスクリプタ・アドレス・レジスタ	上位16ビット・アドレス・ベース	
	0EH	カレント受信ディスクリプタ・アドレス・レジスタ	下位16ビット・アドレス・オフセット	
	14H	上位受信リソース・アドレス・レジスタ	上位16ビット・アドレス・ベース	
	15H	リソース・スタート・アドレス・レジスタ	下位16ビット・アドレス・オフセット	
	16H	リソース・エンド・アドレス・レジスタ	下位16ビット・アドレス・オフセット	
	17H	リソース・リード・ポインタ・レジスタ	下位16ビット・アドレス・オフセット	
	18H	リソース・ライト・ポインタ・レジスタ	下位16ビット・アドレス・オフセット	
	2BH	受信シーケンス・カウンタ	8 7 カウンタ値 カウンタ値	
CAMレジスタ	21H	CAMエントリ・ポインタ	4 ポインタ	
	22H	CAMアドレス・ポート2	CAMエントリ上位16ビット	
	23H	CAMアドレス・ポート1	CAMエントリ中位16ビット	
	24H	CAMアドレス・ポート0	CAMエントリ下位16ビット	
	25H	CAMイネーブル・レジスタ	マスク・フィールド	
	26H	CAMディスクリプタ・ポインタ	下位16ビット・アドレス・オフセット	
	27H	CAMディスクリプタ・カウンタ・レジスタ	5 カウンタ値	
	2CH	CRCエラー・タリー・カウンタ	カウンタ値	
タリー・カウンタ	2DH	フレーム配列エラー・タリー・カウンタ	カウンタ値	
	2EH	消失パケット・タリー・カウンタ	カウンタ値	
ウォッチドッグ・ タイマ	29H	ウォッチドッグ・タイマ0	下位16ビット・カウンタ値	
	2AH	ウォッチドッグ・タイマ1	上位16ビット・カウンタ値	
	28H	シリコン・リビジョン・レジスタ	チップ・リビジョン・ナンバ	

5.2 ステータス/制御レジスタ

このレジスタ・セットは、ホスト・システムとの間でステータス/制御情報を受け渡し、μPD72932の動作を制御するために使用します。これらのレジスタは、システムで生成されたコマンドのロード、送受信ステータスの表示、メモリ間でのデータのバッファ、および割り込み制御のために使用します。μPD72932のレジスタ・アドレス端子 RA5-RA0 に必要なアドレスを設定し、チップ・セレクト端子 $\overline{CS}$ をロウ・レベルにすればレジスタが選択されます。表 5-1 から表 5-3 にすべてのレジスタの位置と、データ・シートでの説明箇所を示します。

表 5-1 ユーザ・レジスタ

レジスタ種類	アドレス	R/W	レジスタ名	略号	説明箇所
コマンド/ 制御レジスタ	00H	R/W	コマンド	CR	5.3.1
	01H ^{注1}	R/W	データ・コンフィギュレーション	DCR	5.3.2
	02H	R/W	受信制御	RCR	5.3.3
	03H	R/W	送信制御	TCR	5.3.4
	04H	R/W	割り込みマスク	IMR	5.3.5
	05H	R/W	割り込みステータス	ISR	5.3.6
	3FH ^{注1}	R/W	データ・コンフィギュレーション 2	DCR2	5.3.7
送信レジスタ	06H	R/W	上位送信ディスクリプタ・アドレス	UTDA	5.3.8, 4.4.4 (1)
	07H	R/W	カレント送信ディスクリプタ・アドレス	CTDA	5.3.8, 4.5.3
受信レジスタ	0DH	R/W	上位受信ディスクリプタ・アドレス	URDA	5.3.9, 4.4.4 (1)
	0EH	R/W	カレント受信ディスクリプタ・アドレス	CRDA	5.3.9, 4.4.4 (3)
	13H	R/W	エンド・オブ・バッファ・カウンタ	EOBC	5.3.9, 4.4.2
	14H	R/W	上位受信リソース・アドレス	URRA	5.3.9, 4.4.4 (1)
	15H	R/W	リソース・スタート・アドレス	RSA	5.3.9, 4.4.1
	16H	R/W	リソース・エンド・アドレス	REA	5.3.9, 4.4.1
	17H	R/W	リソース・リード・ポインタ	RRP	5.3.9, 4.4.1
	18H	R/W	リソース・ライト・ポインタ	RWP	5.3.9, 4.4.1
	2BH	R/W	受信シーケンス・カウンタ	RSC	5.3.9, 4.4.3 (4)
CAM レジスタ	21H	R/W	CAM エントリ・ポインタ	CEP	5.1, 5.3.10
	22H ^{注2}	R	CAM アドレス・ポート 2	CAP2	5.1, 5.3.10
	23H ^{注2}	R	CAM アドレス・ポート 1	CAP1	5.1, 5.3.10
	24H ^{注2}	R	CAM アドレス・ポート 0	CAP0	5.1, 5.3.10
	25H ^{注3}	R/W	CAM イネーブル	CE	5.1, 5.3.10
	26H	R/W	CAM ディスクリプタ・ポインタ	CDP	5.1, 5.3.10
	27H	R/W	CAM ディスクリプタ・カウンタ	CDC	5.1, 5.3.10
タリー・カウンタ	2CH ^{注4}	R/W	CRC エラー・タリー	CRCT	5.3.11
	2DH ^{注4}	R/W	FAE タリー	FAET	5.3.11
	2EH ^{注4}	R/W	消失パケット・タリー	MPT	5.3.11
ウォッチドッグ・ カウンタ	29H	R/W	ウォッチドッグ・タイマ 0	WTO	5.3.12
	2AH	R/W	ウォッチドッグ・タイマ 1	WT1	5.3.12
シリコン・リビジョン	28H	R	シリコン・リビジョン	SRR	5.3.13

- 注 1. データ・コンフィギュレーション・レジスタ (DCR と DCR2) は、μPD72932がリセット・モード (CR レジスタの RST ビットがセットされている) のときのみ書き込むことができます。リセット・モード以外の際にこのレジスタに書き込みを行っても、レジスタの値は変わりません。
2. これらのレジスタは、μPD72932がリセット・モード (CR の RST ビットがセットされている) のときのみ読み出すことができます。これらのレジスタをリセット・モード以外で読み出すと、無効なデータが読み出されます。
3. このレジスタは μPD72932がリセット・モードのときのみ書き込むことができます。通常、このレジスタは LCAM コマンドによる CAM ディスクリプタ転送でのみロードできます。
4. これらのレジスタに書き込まれたデータはレジスタにラッチされる前に反転します。つまり、FFFFH の値が書き込まれた場合、これらのレジスタには 0000H の値が入り、リード時はその値が読み出されます。リード動作ではデータは反転しません。

表 5-2 内部使用レジスタ (ユーザ書き込み禁止)

レジスタ種類	アドレス	R/W	レ ジ ス タ 名	略 号	説明箇所
送信レジスタ	08H ^{注1}	R/W	送信パケット・サイズ	TPS	4.5
	09H	R/W	送信フラグメント・カウント	TFC	4.5
	0AH	R/W	送信スタート・アドレス 0	TSA0	4.5
	0BH	R/W	送信スタート・アドレス 1	TSA1	4.5
	0CH ^{注2}	R/W	送信フラグメント・サイズ	TFS	4.5
	20H	R/W	テンポラリ送信ディスクリプタ・アドレス	TTDA	4.5.4
	2FH	R	最大遅延タイマ	MDT	5.3.4
受信レジスタ	0FH	R/W	カレント受信バッファ・アドレス 0	CRBA0	4.4.2, 4.4.4 (2)
	10H	R/W	カレント受信バッファ・アドレス 1	CRBA1	4.4.2, 4.4.4 (2)
	11H	R/W	リメイニング・バッファ・ワード・カウント 0	RBWC0	4.4.2, 4.4.4 (2)
	12H	R/W	リメイニング・バッファ・ワード・カウント 1	RBWC1	4.4.2, 4.4.4 (2)
	19H	R/W	テンポラリ受信バッファ・アドレス 0	TRBA0	4.4.6 (2)
	1AH	R/W	テンポラリ受信バッファ・アドレス 1	TRBA1	4.4.6 (2)
	1BH	R/W	テンポラリ・バッファ・ワード・カウント 0	TBWC0	4.4.6 (2)
	1CH	R/W	テンポラリ・バッファ・ワード・カウント 1	TBWC1	4.4.6 (2)
	1FH	R/W	最終リンク・フィールド・アドレス	LLFA	なし
アドレス・ジェネレータ	1DH	R/W	アドレス・ジェネレータ 0	ADDR0	なし
	1EH	R/W	アドレス・ジェネレータ 1	ADDR1	なし

- 注 1. これらのレジスタから読み出されるデータは、書き込まれていたデータを反転したものです。
2. このレジスタに書き込まれる値は、16ビット・モードでは 1 回、32ビット・モードでは 2 回シフトされます。

表 5-3 工場テスト用レジスタ

アドレス	R/W	レ ジ ス タ 名	略号	説明箇所
30H	R/W	このレジスタは工場でのテスト用です。ユーザはアクセスしないでください。このレジスタにアクセスすると μPD72932に誤動作が生じることがあります。	なし	なし
3EH				

5.3 レジスタの説明

5.3.1 コマンド・レジスタ (RA5-RA0=00H)

このレジスタ (図 5-4) は, μPD72932にコマンドを発行するために使用します。これらのコマンドは各機能に対応するビットをセットすると発行されます。RST ビットを除くすべてのビットは, コマンドが完了すると μPD72932 が自動的にリセットします。RST ビット, RXEN ビット, RXDIS ビット以外のビットにはいつ “0” を書き込んでも影響はありません。コマンドを実行する前には, 必ず RST ビットを “0” にリセットしておかなければなりません。つまり, RST ビットがセットされているときにコマンドを実行する場合は, まず RST ビットをクリアし, 次にコマンドをセットする, という2回の書き込みをコマンド・レジスタに行う必要があります。

このレジスタは, 汎用32ビット・ウォッチドッグ・タイマの制御も行います。ウォッチドッグ・タイマ・レジスタをロードしたあと, ST ビットを “1” にセットすると, レジスタはデクリメントを始めます。IMR のタイマ完了割り込みがイネーブルされていると, カウントが0になったときに割り込みが発生します。

ハードウェア・リセット時, ビット 7, 4, 2 は “1” にセットされ, その他のビットはクリアされます。ソフトウェア・リセット時, ビット 9, 8, 1, 0 はクリアされ, ビット 7, 2 は “1” にセットされ, その他のビットは影響を受けません。

図 5-4 コマンド・レジスタ

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	LCAM	RRRA	RST	0	ST	STP	RXEN	RXDIS	TXP	HTX
						R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W

R/W = Read/Write

フィールド	意 味
LCAM	CAMのロード
RRRA	RRAの読み出し
RST	ソフトウェア・リセット
ST	タイマの始動
STP	タイマの停止
RXEN	レシーバ・イネーブル
RXDIS	レシーバ・ディスエーブル
TXP	パケットの送信
HTX	送信の停止

(1/2)

ビット	機 能
15-10	0 でなければなりません。
9	LCAM : CAM のロード^{注1} このビットをセットすると、μPD72932はCAM ディスクリプタ・ポインタ・レジスタが指すディスクリプタをCAM にロードします。
8	RRRA : RRA の読み出し このビットをセットすると、μPD72932はリソース・リード・ポインタ (RRP) レジスタが指す次のRRA ディスクリプタを読み出します。このビットは初期化時にのみセットします。通常の動作中にセットすると、受信動作が正しく行われない可能性があります。
7	RST : ソフトウェア・リセット このビットをセットすると、内部ステート・マシンがすべてリセットされます。CRC ジェネレータがディスエーブルされ、タリー・カウンタが停止しますが、クリアはされません。このビットが0にリセットされると、μPD72932が動作可能な状態になります。このビットはハードウェア・リセットで“1”になります。μPD72932の動作前に“0”にリセットしなければなりません。
6	0 でなければなりません。
5	ST : タイマの始動 このビットをセットすると、汎用ウォッチドッグ・タイマがカウントを開始、または停止している場合はカウントを再開します。このビットはタイマが停止 (STP をセット) するとリセットされます。このビットをセットすると STP はリセットされます。
4	STP : タイマの停止^{注2} このビットをセットすると、汎用ウォッチドッグ・タイマが停止し、ST ビットがリセットされます。ST ビットがセットされるとタイマはカウントを再開します。電源投入時には“1”にセットされます。

注 1. このビットは送信中(TXP がセットされているとき)にセットしないでください。LCAM と TXP を同時にセットすると μPD72932はロックします。

2. ST ビットと STP ビットを同時にセットするとタイマが停止します。

(2/2)

ビット	機 能
3	RXEN ：レシーバ・イネーブル ^{注1} このビットをセットすると、受信バッファ管理エンジンがメモリへのデータのバッファリングを開始します。パケット受信中にこのビットをセットすると、パケット受信完了後 RXDIS ビットがリセットされます。
2	RXDIS ：レシーバ・ディスエーブル ^{注1} このビットをセットすると、レシーバのメモリや受信 FIFO へのデータのバッファリングをディスエーブルします。パケット受信中にこのビットをセットすると、レシーバはパケット受信完了後にディスエーブルします。レシーバがディスエーブルされると、RXEN ビットはリセットされます。このビットの状態に関係なく、タリー・カウンタはアクティブのままです。
1	TXP ：パケットの送信 ^{注2} このビットをセットすると、μPD72932は送信ディスクリプタ・エリア (TDA) にセットされたパケットを送信します。μPD72932は適切なレジスタを TDA からロードしてから送信を開始します。次のいずれかの状態が発生すると、このビットはクリアされます。 (1) 送信が完了 (μPD72932が EOL=1 を検出) (2) 送信停止コマンド (HTX) をセット (3) 送信アボート状態が発生 (TCR の EXC, EXD, FU, BCM ビットのいずれかがセット)
0	HTX ：送信の停止 このビットをセットすると、現在の送信が完了したあとで送信コマンドが停止します。送信が停止すると TXP はリセットされます。カレント送信ディスクリプタ・アドレス (CTDA) レジスタが、最後に送信されたディスクリプタを指します。μPD72932は TXpkt.status フィールドにステータスを書き込んだあと、このビットをサンプルします。

注 1. RXEN ビットが“1”で、かつパケット受信中に受信動作停止コマンド (RXEN ビット = “0”, RXDIS ビット = “1”) を発行すると、パケット受信が終了するまで RXEN ビットと RXDIS ビットの両方がセットされた状態になります。しかし、受信動作停止コマンド発行のあと、パケット受信が終わっていないうちに別のコマンドを発行しようとして不用意に RXEN ビットに“0”を書き込むと、RXEN ビットが強制的にクリアされてしまい、受信中のパケットが消失してしまいます。したがって、受信停止コマンドを発行した直後に別のコマンドを発行する場合、RXEN ビットを強制的にクリアしないようにするため、コマンド・レジスタを読み出して RXEN ビットが“0”になるのを待ってから次のコマンドを発行してください。

受信動作を再開するときにも同じことがいえます。つまり、受信動作再開コマンド (RXEN ビット = “1”, RXDIS ビット = “0”) を発行した直後に別のコマンドを発行する場合、不用意に RXDIS ビットに“0”を書き込まないようにしてください。そのためには、受信動作再開コマンド発行のあとコマンド・レジスタを読み出して RXDIS ビットが“0”になるのを待ってから次のコマンドを発行してください。

- 2.** このビットはロード CAM が作動しているとき (LCAM がセットされているとき) にセットしないでください。TXP と LCAM を同時にセットすると μPD72932はロックします。

5.3.2 データ・コンフィギュレーション・レジスタ (RA5-RA0=01H)

このレジスタ (図 5-5) は、メモリ・システムと μPD72932 とのデータのリード/ライト時のバス・モードの設定を行います。

ハードウェア・リセット時はビット15およびビット13がクリアされます。その他のビットは影響を受けません (このため、ドライバ・ソフトウェアは μPD72932 に対して、まず最初にこのレジスタのセットアップを行わなければなりません)。ソフトウェア・リセットではすべてのビットは影響を受けません。このレジスタは、μPD72932 がリセット・モードのとき (コマンド・レジスタの RST ビットがセットされているとき) のみアクセスできます。

図 5-5 データ・コンフィギュレーション・レジスタ

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXBUS	0	LBR	PO1	PO0	SBUS	USR1	USR0	WC1	WC0	DW	BMS	RFT1	RFT0	TFT1	TFT0
R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

R/W = Read/Write

フィールド	意 味
EXBUS	拡張バス・モード
LBR	ラッチされたバスの再実行
PO1, PO0	プログラマブル出力
SBUS	同期バス・モード
USR1, USR0	ユーザ定義端子
WC1, WC0	ウェイト・ステート制御
DW	データ幅の選択
BMS	DMAブロック・モードの選択
RFT1, RFT0	受信FIFOスレッシュホールド
TFT1, TFT0	送信FIFOスレッシュホールド

(1/3)

ビット	機 能
15	EXBUS : 拡張バス・モード このビットをセットすると、次の (1) - (3) をイネーブルする拡張バス・モードになります。 (1) 拡張プログラマブル出力, EXUSR3-EXUSRO TXD, LBK, RXC, RXD 端子を、外部 ENDEC インタフェースから USR1, USR0 と同様の 4 本のプログラマブル出力, EXUSR3-EXUSRO に変更します。これらの出力のプログラムには DCR2 のビット 15-ビット 12 を使用します (5.3.7 参照)。ハードウェア・リセット時、この 4 本の端子はハイ・インピーダンスになり、DCR を変更するまでその状態を保持します。EXBUS をイネーブルにすると、これらの端子は μPD72932 がバス・マスタのとき以外はハイ・インピーダンスを保持し、μPD72932 がバス・マスタになったときのみ DCR2 に従ってドライブされます。EXBUS をディスエーブルにすると、この 4 本の端子は通常の外部インタフェース端子として動作します。 (2) 同期停止, $\overline{\text{STERM}}$ TXC_o/TXC_i/$\overline{\text{STERM}}$ 端子を外部 ENDEC インタフェースからモトローラ型プロセッサに対応する同期メモリの終端入力である $\overline{\text{STERM}}$ 端子機能に設定します。この入力、BMODE=1 (モトローラ・モード) で非同期バス・モードを選択 (ビット 10 を "0" にセット) しているときのみ有効です。ハードウェア・リセット時、この端子はハイ・インピーダンスになり、DCR を変更するまでその状態を保持します。EXBUS をイネーブルにすると、この端子は μPD72932 がバス・マスタのとき以外はハイ・インピーダンスを保持し、μPD72932 がバス・マスタのときは $\overline{\text{STERM}}$ 入力になります。EXBUS をディスエーブルにすると、この端子は通常の外部 ENDEC 用の TXC 端子として動作します。 (3) 非同期バスの再実行 $\overline{\text{BRT}}$ をバス・クロックの立ち下がりエッジから非同期にサンプリング実行します。これは、μPD72932 が非同期モード (ビット 10 を "0" にセット) で動作しているときにかぎり適用されます。$\overline{\text{BRT}}$ はバス・クロックの立ち上がりエッジに同期して取り込みます (6.3.6 参照)。
14	0 でなければなりません。
13	LBR : ラッチされたバスの再実行^注 LBR ビットは $\overline{\text{BRT}}$ 信号 (1.2 バス・インタフェース端子参照) の動作モードを制御します。$\overline{\text{BRT}}$ の立ち下がりエッジからバスの再実行動作をラッチまたはアンラッチすることができます。 0 : アンラッチ・モード $\overline{\text{BRT}}$ 端子をロウ・レベルにすると μPD72932 が現在の DMA 動作を終了し、ハイ・レベルにすると再実行します。 1 : ラッチ・モード $\overline{\text{BRT}}$ 端子をロウ・レベルにすると、アンラッチ・モードと同様に μPD72932 が現在の DMA 動作を終了します。しかし、ISR の BR ビット (5.3.6 参照) がリセットされ、かつ $\overline{\text{BRT}}$ 端子がハイ・レベルになるまで、μPD72932 は再実行しません。したがって、BR ビットがクリアされるまでラッチされた状態になります。
12, 11	PO1, PO0 : プログラマブル出力 PO1 ビット, PO0 ビットは μPD72932 がバス・マスタのとき (HLDA または $\overline{\text{BGACK}}$ がアクティブのとき)、それぞれ USR1, USR0 端子を制御します。PO1 または PO0 を 1 にセットすると USR1 端子または USR0 端子はバス・マスタ動作の間ハイ・レベルになり、これらのビットを 0 にセットすると、USR1, 0 端子はバス・マスタ動作の間ロウ・レベルになります。

注 LBR が "0" にセットされている場合、μPD72932 がアイドル状態になるまで $\overline{\text{BRT}}$ をロウ・レベルに保持する必要があります。6.3.6 および 9. 電気的特性のバス再実行のタイミングを参照してください。

(2/3)

ビット	機 能															
10	SBUS：同期バス・モード SBUS ビットは μ PD72932がバス・マスタのとき、システム・バス動作モードを選択するために使用します。このビットは、ブロック転送 DMA 動作中に μ PD72932に入力される内部レディ・ラインを、同期または非同期に設定します。 0：非同期モード $\overline{RDY}_i$ (BMODE=0) または $\overline{DSACK}_0$, $\overline{DSACK}_1$ (BMODE=1) がそれぞれバス・クロックの立ち上がりエッジ (DMA サイクルの T2) で内部的に同期されます。セットアップ時間やホールド時間をクロックのエッジに合わせなくてもバス動作は保証されます。 1：同期モード バス動作を正しく行うために、 $\overline{RDY}_i$ (BMODE=0)または $\overline{DSACK}_0$, $\overline{DSACK}_1$ (BMODE=1)は T1 または T2 の立ち上がりエッジに対してセットアップ時間およびホールド時間を合わせなければなりません。															
9, 8	USR1, USRO：ユーザ定義端子 USR1 ビット, USRO ビットはそれぞれ、チップ・ハードウェア・リセット後の USR1 端子, USRO 端子のレベルを反映します。ハードウェア・リセット時に USR1, USRO 端子が論理 1 (V_{CC} に接続) の場合, USR1, USRO ビットは 1 にセットされます。論理 0 (グランドに接続) の場合は 0 にセットされます。これらのビットは $\overline{RESET}$ 端子の立ち上がりエッジでラッチされます。いったんセットされると、次のハードウェア・リセットまで新たにラッチ動作を行いません。															
7, 6	WC1, WCO：ウェイト・ステート制御 これらのビットは各 DMA サイクル中に追加されるバス・サイクル (T2 ステート) の数を決定します。 <table><tr><th>WC1</th><th>WCO</th><th>追加されるバス・サイクル</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>2</td></tr><tr><td>1</td><td>1</td><td>3</td></tr></table>	WC1	WCO	追加されるバス・サイクル	0	0	0	0	1	1	1	0	2	1	1	3
WC1	WCO	追加されるバス・サイクル														
0	0	0														
0	1	1														
1	0	2														
1	1	3														
5	DW：データ幅の選択 このビットは DMA 動作のためのデータ幅を選択します。 <table><tr><th>DW</th><th>データ幅</th></tr><tr><td>0</td><td>16ビット</td></tr><tr><td>1</td><td>32ビット</td></tr></table>	DW	データ幅	0	16ビット	1	32ビット									
DW	データ幅															
0	16ビット															
1	32ビット															
4	BMS：DMA ブロック・モードの選択 受信 FIFO または送信 FIFO に対するデータのフィル/エンプティの方法を決定します。 0：エンプティ/フィル・モード 受信 FIFO が完全に空になるか、送信 FIFO が完全に満たされるまで、DMA 転送が継続されます。 1：ブロック・モード プログラムされたバイト数 (受信時は RFT0, RFT1。送信時は TFT0, TFT1) の転送が終わるまで、DMA 転送が継続されます (次頁の注2 参照)。															

(3/3)

ビット	機 能															
3, 2	RFT1, RFT0 : 受信 FIFO スレッシュホールド^{注1} これらのビットは受信 DMA 要求が発生する前に MAC ユニットから受信 FIFO へ書き込まれるワード（またはロング・ワード）数を決定します（ 2.4 参照）。 <table><tr><th>RFT1</th><th>RFT0</th><th>スレッシュホールド</th></tr><tr><td>0</td><td>0</td><td>2ワードまたは1ロング・ワード（4バイト）</td></tr><tr><td>0</td><td>1</td><td>4ワードまたは2ロング・ワード（8バイト）</td></tr><tr><td>1</td><td>0</td><td>8ワードまたは4ロング・ワード（16バイト）</td></tr><tr><td>1</td><td>1</td><td>12ワードまたは6ロング・ワード（24バイト）</td></tr></table>	RFT1	RFT0	スレッシュホールド	0	0	2ワードまたは1ロング・ワード（4バイト）	0	1	4ワードまたは2ロング・ワード（8バイト）	1	0	8ワードまたは4ロング・ワード（16バイト）	1	1	12ワードまたは6ロング・ワード（24バイト）
RFT1	RFT0	スレッシュホールド														
0	0	2ワードまたは1ロング・ワード（4バイト）														
0	1	4ワードまたは2ロング・ワード（8バイト）														
1	0	8ワードまたは4ロング・ワード（16バイト）														
1	1	12ワードまたは6ロング・ワード（24バイト）														
1, 0	TFT1, TFT0 : 送信 FIFO スレッシュホールド^{注2} これらのビットは、DMA 部が管理する送信 FIFO 中の最小ワード（またはロング・ワード）数を決定します。ワード数が送信 FIFO スレッシュホールド以下になるとバス・リクエストが発生します（ 2.4 参照）。 <table><tr><th>TFT1</th><th>TFT0</th><th>スレッシュホールド</th></tr><tr><td>0</td><td>0</td><td>4ワードまたは2ロング・ワード（8バイト）</td></tr><tr><td>0</td><td>1</td><td>8ワードまたは4ロング・ワード（16バイト）</td></tr><tr><td>1</td><td>0</td><td>12ワードまたは6ロング・ワード（24バイト）</td></tr><tr><td>1</td><td>1</td><td>14ワードまたは7ロング・ワード（28バイト）</td></tr></table>	TFT1	TFT0	スレッシュホールド	0	0	4ワードまたは2ロング・ワード（8バイト）	0	1	8ワードまたは4ロング・ワード（16バイト）	1	0	12ワードまたは6ロング・ワード（24バイト）	1	1	14ワードまたは7ロング・ワード（28バイト）
TFT1	TFT0	スレッシュホールド														
0	0	4ワードまたは2ロング・ワード（8バイト）														
0	1	8ワードまたは4ロング・ワード（16バイト）														
1	0	12ワードまたは6ロング・ワード（24バイト）														
1	1	14ワードまたは7ロング・ワード（28バイト）														

注 1. ブロック・モード (BMS ビット=1) では, 受信 FIFO スレッシュホールドは受信 DMA ブロック・サイクルでメモリへ書き込まれるワード (またはロング・ワード) 数を設定します。

2. ブロック・モード (BMS ビット=1) において, μPD72932が1回の DMA バースト転送で行うバイト読み取り数は, 送信 FIFO スレッシュホールド値に等しくなります。FIFO を満たすのに必要なワード(ロング・ワード) 数がスレッシュホールド値以下だと, FIFO を満たすだけのバイト数が1回の DMA 転送で行われます。一般に, FIFO に12ワードか14ワードを設定すると, メモリ読み取り数は FIFO スレッシュホールド以下になります。

5.3.3 受信制御レジスタ (RA5-RA0=02H)

このレジスタは着信パケットをフィルタし、受け入れたパケットのステータス情報を得るために使用します (図 5-6)。ビット15-ビット11のうちの任意のビットをセットすると、対応する受信フィルタがイネーブルされます。いずれのビットもセットされない場合、CAM アドレス・レジスタに適合するパケットだけを受け入れます (ノーマル・アドレス・モード)。ビット10およびビット9は、ループバック動作を制御します。

受信後、ビット8-ビット0は受信したパケットについてのステータス情報を示し、対応する条件が発生したときに“1”にセットされます。パケットを受信すると、RCR のビットはすべて RXpkt.status フィールドに書き込まれます。次のパケットを受信すると、ビット8-ビット6, およびビット3-ビット0はクリアされます。

このレジスタはソフトウェア・リセットの影響を受けません。

図 5-6 受信制御レジスタ

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERR	RNT	BRD	PRO	AMC	LB1	LB0	MC	BC	LPKT	CRS	COL	CRCR	FAER	LBK	PRX
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R

R = Read Only, R/W = Read/Write

フィールド	意 味
ERR	CRCエラー, またはコリジョンのあるパケットの受信
RNT	ラント・パケットの受信
BRD	ブロードキャスト・パケットの受信
PRO	全物理アドレス・パケットの受信
AMC	全マルチキャスト・パケットの受信
LB0, LB1	ループバック制御
MC	マルチキャスト・パケットの受信
BC	ブロードキャスト・パケットの受信
LPKT	RBA内の最終パケット
CRS	キャリア検知
COL	コリジョン
CRCR	CRCエラー
FAER	フレーム配列エラー
LBK	ループバック・パケットの受信
PRX	正常なパケットの受信

(1/2)

ビット	機 能															
15	ERR: CRC エラー, またはコリジョンのあるパケットの受信 0: CRC エラーのあるパケットおよびコリジョン発生時のパケットは受信しません。 1: エラーのあるパケットも受信し, コリジョンも無視します。															
14	RNT: ラント・パケットの受信^注 0: ラント・パケット (64バイト長以下のパケット) は受信しません。 1: ラント・パケット (64バイト長以下のパケット) も受信します。															
13	BRD: ブロードキャスト・パケットの受信^注 0: ノーマル・アドレス・マッチ・モード 1: ブロードキャスト・パケットを受信します (CAM に整合したアドレスをもつパケットも受信します)。															
12	PRO: 全物理アドレス・パケットの受信 0: ノーマル・アドレス・マッチ・モード 1: 透過モード															
11	AMC: 全マルチキャスト・パケットの受信 0: ノーマル・アドレス・マッチ・モード 1: すべてのマルチキャスト・パケットを受信可能にします。ブロードキャスト・パケットも BRD ビットにかかわらずすべて受信します (ブロードキャスト・パケットはマルチキャスト・パケットのサブセットです)。															
10, 9	LB1, LBO: ループバック制御^注 これらのビットは, MAC ループバック, ENDEC ループバック, およびトランシーバ・ループバックに対してその動作を制御します。ループバック動作を正しく行うためには, ループバック・パケットのデスティネーション・アドレスを受け入れるように, CAM アドレス・レジスタと受信制御レジスタを初期化しなければなりません。 <table><tr><th>LB1</th><th>LBO</th><th>機 能</th></tr><tr><td>0</td><td>0</td><td>ループバックなし, 通常動作</td></tr><tr><td>0</td><td>1</td><td>MAC ループバック</td></tr><tr><td>1</td><td>0</td><td>ENDEC ループバック</td></tr><tr><td>1</td><td>1</td><td>トランシーバ・ループバック</td></tr></table>	LB1	LBO	機 能	0	0	ループバックなし, 通常動作	0	1	MAC ループバック	1	0	ENDEC ループバック	1	1	トランシーバ・ループバック
LB1	LBO	機 能														
0	0	ループバックなし, 通常動作														
0	1	MAC ループバック														
1	0	ENDEC ループバック														
1	1	トランシーバ・ループバック														
8	MC: マルチキャスト・パケットの受信 有効なマルチキャスト・アドレス・パケットを受信したときにこのビットがセットされます。															
7	BC: ブロードキャスト・パケットの受信 有効なブロードキャスト・アドレス・パケットを受信したときにこのビットがセットされます。															
6	LPKT: RBA 内の最終パケット パケットが受信バッファ・エリア (RBA) にバッファされた最後のパケットであったとき, すなわちリメイニング・バッファ・ワード・カウント・レジスタ (RBWC0, RBWC1) がエンド・オブ・バッファ・カウント・レジスタ (EOBC) 以下であったとき, このビットをセットします (4.4.2 参照)。															
5	CRS: キャリア検知 CRS がアクティブであるときセットされます。ネットワークがアクティブであることを示します。															
4	COL: コリジョン コリジョンの発生した受信パケットであったことを示します。															

注 ハードウェア・リセットによってクリアされます。

(2/2)

ビット	機 能
3	CRCR : CRC エラー パケットがCRC エラーを含んでいることを示します。パケットがフレーム配列エラーも含んでいる場合、FAER ビットが代わりにセットされます（このとき CRCR ビットはセットされません）。
2	FAER : フレーム配列エラー^注 受信パケットが8 ビット境界で正しく区切られなかったことを示します。
1	LBK : ループバック・パケットの受信 μPD72932がループバック・パケットを正常に受信したことを示します。
0	PRX : 正常なパケットの受信 パケットがCRC エラー、フレーム配列エラー、ラント・パケット・エラーおよびコリジョンなしに受信されたことを示します。

注 CRC エラーが発生していなければ、このビットはセットされません（つまり、このビットがセットされるのはフレーム配列エラーとCRC エラーの両方が発生したときに限ります）。

5.3.4 送信制御レジスタ (RA5-RA0=03H)

このレジスタは μPD72932の送信動作をプログラムし、パケット送信後にステータス情報を得るために使用します (図 5-7)。送信の始めに、TXpkt.config フィールドからビット15, 14, 13およびビット12がTCRにロードされ、さまざまな送信モードを構成します (4.5.1 (2) 参照)。送信が終了すると、ビット10-ビット0がステータス情報を示し、対応する条件が発生したとき“1”にセットされます。コリジョン情報の数に基づいたこれらのビットは、送信終了時にTXpkt.status フィールドに書き込まれます (4.5.1 (1) 参照)。TXpkt.status フィールドに書き込みが終わると、ビット9およびビット5はクリアされます。ビット10, 7, 6およびビット1は次の送信が開始するとクリアされ、ビット8はこのときセットされます。

ハードウェア・リセットでビット8およびビット1は“1”にセットされます。ソフトウェア・リセットではこのレジスタは影響を受けません。

図 5-7 送信制御レジスタ

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PINTR	POWC	CRCI	EXDIS	0	EXD	DEF	NCRS	CRSL	EXC	OWC	0	PMB	FU	BCM	PTX
R/W	R/W	R/W	R/W		R	R	R	R	R	R		R	R	R	R

R = Read Only, R/W = Read/Write

フィールド	意 味
PINTR	プログラマブル割り込み
POWC	ウィンドウ外コリジョン・タイマのプログラム
CRCI	CRC禁止
EXDIS	過剰遅延タイマの禁止
EXD	過剰遅延
DEF	過剰送信
NCRS	CRS未検出
CRSL	CRSのロスト
EXC	過剰コリジョン
OWC	ウィンドウ外コリジョン
PMB	モニタしたパケットの不良
FU	FIFOのアンダーラン
BCM	バイト・カウンタのミスマッチ
PTX	正常パケット送信

(1/2)

ビット	機 能
15	PINTR : プログラマブル割り込み^{注1} このビットによりソフトウェア制御による送信割り込みが生成されます。μPD72932が TDA を読み出し TXpkt.config フィールドの PINTR ビットがセットされていることを検出すると、すぐに割り込みが生成されます (割り込みステータス・レジスタの PINT がセットされます)。
14	POWC : ウィンドウ外コリジョン・タイマのプログラム このビットはウィンドウ外コリジョン・タイマの開始タイミングをプログラムします。 0 : フレーム開始デリミタ (SFD) のあと、タイマが始動します。 1 : プリアンプルの最初のビットのあと、タイマが始動します。
13	CRCI : CRC 禁止 0 : 4 バイト FCS フィールドを付加してパケットを送信します。 1 : 4 バイト FCS フィールドなしでパケットを送信します。
12	EXDIS : 過剰遅延タイマの禁止 0 : 過剰遅延タイマをイネーブルします。 1 : 過剰遅延タイマをディスエーブルします。
11	0 でなければなりません。
10	EXD : 過剰遅延 μPD72932の送信が3.2 ms 遅延したことを示します。過剰遅延タイマがイネーブルされている (EXDIS がリセットされている) ときに遅延が3.2 ms に達すると送信はアボートされます。このビットがセットされるのは、過剰遅延タイマがイネーブルされているときに過剰遅延が発生した場合のみです。
9	DEF : 遅延送信 最初のキャリア・センス動作の間にネットワーク・ビジーを検出したことで、μPD72932の送信が遅延したことを示します。それに続いて衝突が発生すればこのビットはリセットされます。TDA の TXpkt.status フィールドが書き込まれるとこのビットはクリアされます。
8	NCRS : CRS 未検出^{注2} 送信中にキャリア・センス (CRS) が検出されなかったことを示します。CRS はフレーム開始デリミタの先頭から最後の送信バイトまでモニタされます。このビットがセットされても送信はアボートされません。このビットはプリアンプルの開始でセットされ、CRS が検出されるとリセットされます。したがって、パケット送信中に CRS を一度も検出しないと、このビットはセット状態に保たれます。
7	CRSL : CRS のロスト^{注3} 送信中に CRS がロウ・レベルになったか、あるいはキャリアが存在しなかったことを示します。CRS はフレーム開始デリミタから最後の送信バイトまでモニタされます。このビットがセットされても送信はアボートされません。
6	EXC : 過剰コリジョン 16回の衝突が発生したことを示します。送信はアボートされます。

注 1. 複数の TDA において PINTR を正常に動作させるには、TXpkt.config フィールドを交互にセット、リセットしてください。すべての TDA において連続して PINTR をセットすることはできません。なぜならば、送信パケットの割り込み受け付けで ISR の PINT が一度セットされたあと、もう一度セットするために、次のパケット送信時の TXpkt.config フィールドをリセットすることで送信制御レジスタの PINTR を "0" にクリアしておかなければならないからです。

2. NCRS は、MAC ループバック中は常にセットされます。

3. CRS が存在しなかった場合は、NCRS と CRSL の両方が同時にセットされます。また、CRSL は MAC ループバック中は常にセットされます。

(2/2)

ビット	機 能
5	OWC：ウィンドウ外コリジョン プリアンプルの最初のビット (POWC=1) または SFD (POWC=0) の送信から、51.2 μs (ワン・スロット時間) 以降に衝突が発生したことを示します。送信は停止します。TDA の TXpkt.status フィールドが書き込まれるとこのビットはクリアされます。
4	0 でなければなりません。
3	PMB：モニタしたパケットの不良^{注1, 2, 3, 4} このビットは受信ユニットが送信パケットをモニタしたあと、計算した CRC が無効であった場合、フレーム配列エラーが発生した場合、およびソース・アドレスがどの CAM アドレス・レジスタとも一致しない場合にセットされます。
2	FU：FIFO のアンダーラン FIFO が空になる前に、μPD72932がバスにアクセスできなかったことを示します。この状態は過剰なバスの待ち時間およびスロー・バス・クロック、あるいはそのいずれかによって発生します。送信はアボートされます (2.4.2 参照)。
1	BCM：バイト・カウントのミスマッチ このビットは TXpkt.pkt_size フィールドが TXpkt.frag_size フィールドの合計と等しくないことを μPD72932が検出するとセットされます。送信はアボートされます。送信中に過剰コリジョン (EXC=1) が発生するとこのビットがセットされます。
0	PTX：正常パケット送信 次のエラーなしでパケットが送信されたことを示します。 ●過剰コリジョン (EXC) ●過剰遅延 (EXD) ●FIFO アンダーラン (FU) ●バイト・カウントのミスマッチ (BCM)

注 1. μPD72932の CRC チェッカは送信中にアクティブになります。

2. CRC の送信が禁止された場合 (CRCI がセット)、このビットは (フレーム配列エラーやソース・アドレス・ミスマッチ・エラーを問わず) 常に "0" になります。
3. 受信 FIFO オーバランが発生した場合、送信されたパケットは完全にモニタされません。したがって、ISR の RFO ビットとともにこのビットがセットされた場合、このビットは不定です。パケットを完全に受信してからこのビットが有効になります。
4. 3 種類のループバック・モードのいずれかがセットされた場合、このビットは常に "0" になります。

5.3.5 割り込みマスク・レジスタ (RA5-RA0=04H)

このレジスタは ISR から生成可能な割り込みをマスクします (図 5-8)。ビットに “1” を書き込むと対応する割り込みがイネーブルされます。ハードウェア・リセット時にはすべてのマスク・ビットがクリアされます。

図 5-8 割り込みマスク・レジスタ

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	BREN	HLEN	LCDEN	PINTEN	PRXEN	PTXEN	TXEREN	TCEN	RDEEN	RBEEN	RBAEEN	CRCEN	FAEEN	MPEN	RFOEN
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

R = Read Only, R/W = Read/Write

フィールド	意 味
BREN	バス再実行発生イネーブル
HLEN	ハートビート・ロスト・イネーブル
LCDEN	ロードCAM完了割り込みイネーブル
PINTEN	プログラマブル割り込みイネーブル
PRXEN	パケット受信イネーブル
PTXEN	正常パケット受信イネーブル
TXEREN	送信エラー・イネーブル
TCEN	汎用タイマ完了イネーブル
RDEEN	受信ディスクリプタ枯渇イネーブル
RBEEN	受信バッファ枯渇イネーブル
RBAEEN	受信バッファ・エリア超過イネーブル
CRCEN	CRCタリー・カウンタ警告イネーブル
FAEEN	FAEタリー・カウンタ警告イネーブル
MPEN	MPタリー・カウンタ警告イネーブル
RFOEN	受信FIFOオーバラン・イネーブル

(1/2)

ビット	機 能
15	0でなければなりません。
14	BREN : バス再実行発生イネーブル 0 : ディスエーブル 1 : バス再実行動作の要求時に割り込みをイネーブル
13	HLEN : ハートビート・ロスト・イネーブル 0 : ディスエーブル 1 : ハートビート・ロスト状態の発生時に割り込みをイネーブル
12	LCDEN : ロード CAM 完了割り込みイネーブル 0 : ディスエーブル 1 : ロード CAM コマンド完了時に割り込みをイネーブル
11	PINTEN : プログラマブル割り込みイネーブル 0 : ディスエーブル 1 : TXpkt.config フィールドの PINTR ビットが “1” にセットされたときにプログラマブル割り込みのセットをイネーブル

(2/2)

ビット	機 能
10	PRXEN : パケット受信イネーブル 0 : ディスエーブル 1 : 受信されたパケットに対する割り込みをイネーブル
9	PTXEN : 正常パケット送信イネーブル 0 : ディスエーブル 1 : 送信完了に対する割り込みをイネーブル
8	TXEREN : 送信エラー・イネーブル 0 : ディスエーブル 1 : エラーのある送信パケットに対する割り込みをイネーブル
7	TCEN : 汎用タイマ完了イネーブル 0 : ディスエーブル 1 : 汎用タイマが 0000 0000H から FFFF FFFFH にロール・オーバーしたときに割り込みをイネーブル
6	RDEEN : 受信ディスクリプタ枯渇イネーブル 0 : ディスエーブル 1 : RDA のすべての受信ディスクリプタがなくなったときに割り込みをイネーブル
5	RBEEN : 受信バッファ枯渇イネーブル 0 : ディスエーブル 1 : RRA のすべてのリソース・ディスクリプタがなくなったときに割り込みをイネーブル
4	RBAEEN : 受信バッファ・エリア超過イネーブル 0 : ディスエーブル 1 : μPD72932が受信バッファ・エリアの終わりを越えてデータをバッファしようとしたときに割り込みをイネーブル
3	CRCEN : CRC タリー・カウンタ警告イネーブル 0 : ディスエーブル 1 : CRC タリー・カウンタが FFFFH から 0000H にロール・オーバーしたときに割り込みをイネーブル
2	FAEEN : FAE (フレーム配列エラー) タリー・カウンタ警告イネーブル 0 : ディスエーブル 1 : FAE タリー・カウンタが FFFFH から 0000H にロール・オーバーしたときに割り込みをイネーブル
1	MPEN : MP (消失パケット) タリー・カウンタ警告イネーブル 0 : ディスエーブル 1 : MP タリー・カウンタが FFFFH から 0000H にロール・オーバーしたときに割り込みをイネーブル
0	RFOEN : 受信 FIFO オーバラン・イネーブル 0 : ディスエーブル 1 : 受信 FIFO がオーバランしたときに割り込みをイネーブル

5.3.6 割り込みステータス・レジスタ (RA5-RA0=05H)

このレジスタ(図 5-9)は INT 端子がアクティブになったときの割り込みソースを示します。IMR の対応するビットをイネーブルすると、このレジスタの対応するビットが割り込みを発生します。割り込みがアクティブのときには、このレジスタの1つまたは複数のビットが“1”にセットされます。ビットは“1”を書き込むとクリアされます。“0”を書き込んででも影響はありません。

このレジスタはハードウェア・リセットによってクリアされますが、ソフトウェア・リセットの影響は受けません。

図 5-9 割り込みステータス・レジスタ

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	BR	HBL	LCD	PINT	PKTRX	TXDN	TXER	TC	RDE	RBE	RBAE	CRC	FAE	MP	RFO
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

R/W = Read/Write

フィールド	意 味
BR	バス再実行の発生
HBL	CDハートビートの喪失
LCD	ロードCAM完了
PINT	プログラマブル割り込み
PKTRX	パケットの受信
TXDN	送信完了
TXER	送信エラー
TC	汎用タイマ完了
RDE	受信ディスクリプタの枯渇
RBE	受信バッファの枯渇
RBAE	受信バッファ・エリアの超過
CRC	CRCタリー・カウンタのロール・オーバー
FAE	フレーム配列エラー・タリー・カウンタのロール・オーバー
MP	消失パケット・カウンタのロール・オーバー
RFO	受信FIFOオーバーラン

(1/2)

ビット	機 能
15	0 でなければなりません。
14	BR：バス再実行の発生 バス再実行 (BRT) が発生したことを示します。バス再実行のラッチ・モード (DCR の LBR) で BR がセットされるのは、μPD72932 がバス・マスタのときに限ります。μPD72932 が DMA 動作を続ける前に、BR ビットをクリアしなければなりません。アンラッチ・モードでも BR ビットをクリアしなければなりません。μPD72932 は BR ビットがクリアされるのを待たずに再度バス要求を出し DMA 動作を続けます (バス再実行の詳細については 5.3.2 と 6.3.6 を参照してください)。
13	HBL：CD ハートビートの喪失 トランシーバが送信後のフレーム間ギャップの最初の 6.4 μs 中に衝突信号 (ハートビート) を発生しなかった場合、このビットがセットされます。
12	LCD：ロード CAM 完了 ロード CAM コマンドによって CAM 中のすべてのプログラムされたロケーションへのアドレスの書き込みが完了したことを示します (5.1.2 参照)。
11	PINT：プログラマブル割り込み TXpkt.config フィールドを読み出したときに、PINTR ビットが "0" から "1" に変化したことを μPD72932 が検出したことを示します。
10	PKTRX：パケットの受信 パケットを受信し、メモリにバッファしたことを示します。このビットは RXpkt.seq_no フィールドがメモリ (RDA) に書き込まれたあとにセットされます。
9	TXDN：送信完了 次のいずれかの状態を示します。 (1) 送信ディスクリプタ・エリアに送信すべきパケットがない状態 (EOL ビット=1 を検出) (2) 送信停止コマンドが与えられた状態 (CR の HTX ビットが "1" にセット) (3) 送信アボート状態が発生した状態 (TCR の BCM, EXC, FU, EXD ビットのいずれかがセット) このビットは TXpkt.status フィールドが書き込まれたあとにセットされます。
8	TXER：送信エラー 送信されたパケットに、次のエラーのうちの少なくとも 1 つが含まれていたことを示します。 ●バイト・カウントのミスマッチ (BCM) ●過剰コリジョン (EXC) ●FIFO のアンダーラン (FU) ●過剰遅延 (EXD) TXpkt.status フィールドでエラーの原因を知ることができます。
7	TC：汎用タイマ完了 タイマが 0000 0000H から FFFF FFFFH にロール・オーバーしたことを示します (5.3.12 参照)。
6	RDE：受信ディスクリプタの枯渇 RDA にあるすべての受信パケット・ディスクリプタがなくなったことを示します。このビットは μPD72932 が RDA の RXpkt.link フィールドにある EOL ビットが "1" であることを検出したときにセットされます (4.4.7 参照)。

(2/2)

ビット	機 能
5	RBE: 受信バッファの枯渇 ^{注1, 2, 3} リソース・リード・ポインタ (RRP) とリソース・ライト・ポインタ (RWP) が等しいことを μPD72932 が検出したことを示します。このビットは、リソース・エリアから最後のビットを読み出したあとにセットされます。
4	RBAE: 受信バッファ・エリアの超過 受信中、μPD72932 が受信バッファ・エリアの終わりに達したことを示します。受信はアボートされ、μPD72932 は RRA 内で次に利用可能なリソース・ディスクリプタをフェッチします。バッファの残リスペースは再使用されず、また RDA は不完全なバケット用に設定されません (4.4.7 参照)。
3	CRC: CRC タリー・カウンタのロール・オーバ CRC タリー・カウンタが FFFFH から 0000H にロール・オーバしたことを示します (5.3.11 参照)。
2	FAE: フレーム配列エラー・タリー・カウンタのロール・オーバ FAE タリー・カウンタが FFFFH から 0000H にロール・オーバしたことを示します (5.3.11 参照)。
1	MP: 消失バケット・カウンタのロール・オーバ MP タリー・カウンタが FFFFH から 0000H にロール・オーバしたことを示します (5.3.11 参照)。
0	RFO: 受信 FIFO オーバーラン 受信 FIFO がネットワークからのデータでオーバフローした (μPD72932 のバス・アクセスが遅かったため、FIFO のオーバーランが発生した) ことを示します。この状態は、きわめて長いバス待ち時間および低速バス・クロック、またはそのいずれかで発生します。FIFO アンダーランは TCR で示します (2.4.1 参照)。

注 1. RBE ビットは μPD72932 が最後から 2 番目の受信バッファの使用を終了し、最後の RRA ディスクリプタを読み出すとセットされます。これにより、リソースの枯渇が差し迫っていることを早期にシステムに警告します。

- μPD72932 は最後の RBA を使い終わると、別の受信バッファが追加 (RWP が RRP を越えてインクリメントされる) され、RBE ビットがリセットされるまで、パケットの受信を停止します。
- バッファが追加された場合、RBE ビットをリセットすると、μPD72932 は受信リソース・エリアの RRP で示される次のリソース・ディスクリプタを読み出します。この状態で、このビットをリセットすることは、リード RRA コマンドをセットすること (CR の RRA ビットのセット) と同じです。RRA に別のリソースを追加するまで、このビットをリセットしないでください。

5.3.7 データ・コンフィギュレーション・レジスタ2 (RA5-RA0=3FH)

このレジスタ (図 5-10) は、拡張バス・インタフェース・オプションをイネーブルするのに使用します。

このレジスタのビットは、EXPO (拡張プログラマブル出力) ビットを除き、すべてハードウェア・リセットによって“0”にリセットされます。EXPO ビットは書き込まれるまで不定状態です。また、常に0にセットされているビット5-ビット11は、いつ読み出してもかまいません。このレジスタはソフトウェア・リセットの影響を受けません。このレジスタへの書き込みは、μPD72932がソフトウェア・リセット (コマンド・レジスタの $\overline{\text{RST}}$ ビットがセット) 状態にあるときに限ります。

図 5-10 データ・コンフィギュレーション・レジスタ2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXPO3	EXPO2	EXPO1	EXPO0	0	0	0	0	0	0	0	PH	0	PCM	PCNM	RJCM
R/W	R/W	R/W	R/W								R/W		R/W	R/W	R/W

R/W = Read/Write

フィールド	意 味
EXPO3-EXPO0	拡張プログラマブル出力
PH	プログラム・ホールド
PCM	一致時のパケット圧縮
PCNM	不一致時のパケット圧縮
RJCM	CAM一致パケットのリジェクト

ビット	機 能
15-12	EXPO3-EXPO0 : 拡張プログラマブル出力 μPD72932がバス・マスタのとき、これらのビットで拡張ユーザ出力 (EXUSR3-EXUSR0) レベルをプログラムできます。これらのビットのいずれかに“1”を書き込むと、対応する出力をハイ・レベルにプログラムすることができ、“0”を書き込むとロウ・レベルにプログラムすることができます。EXUSR3-EXUSR0 は、拡張バス・モードを使用している場合 (DCR のビット15を“1”にセット, 5.3.2 参照) にかぎり有効という点を除いては USR1, USR0 と同じ機能を持っています。
11-5	0 でなければなりません。
4	PH : プログラム・ホールド このビットを“0”にセットすると、HOLD 要求出力がバス・クロックの立ち下がりエッジに同期してハイまたはロウ・レベルになります。“1”にセットすると、バス・クロックの立ち上がりエッジから1/2クロック分遅れてハイまたはロウ・レベルになります。
3	0 でなければなりません。
2	PCM : 一致時のパケット圧縮^{注1, 2} このビットが“1”にセット (および PCNM ビットが“0”にリセット) され、受信パケットのデスティネーション・アドレスが CAM エントリの 1 つと一致すると、 $\overline{\text{PCOMP}}$ 出力をロウ・レベルにします。このビットは PCNM ビットとともに、μPD72950 ^{注3} のリピータ・インタフェース・コントローラ管理バスで使います。この管理バスについての詳細は、μPD72950のデータ・シートを参照してください。このモードは管理ブリッジ・モードとも呼ばれています。
1	PCNM : 不一致時のパケット圧縮^{注4} このビットが“1”にセット (および PCM ビットが“0”にセット) され、パケットのデスティネーション・アドレスが CAM エントリのいずれにも一致しなければ、 $\overline{\text{PCOMP}}$ 出力をロウ・レベルにします。このモードは管理ハブ・モードとも呼ばれています。
0	RJCM : CAM 一致パケットのリジェクト このビットを“1”にセットすると、μPD72932は CAM 一致のパケットをリジェクトします。RJCM を“0”にセットすると、μPD72932は CAM 一致のパケットを受信して正常に動作します。このモードの設定は、接続されるノードの数を制限した小規模ブリッジの場合に有効です。RJCM は CAM に影響するだけで、RJCM の設定によって、受信制御レジスタ (5.3.3 参照) の BRD ビット、PRO ビットや AMC ビットの機能 (それぞれ、ブロードキャスト・パケット、すべての物理パケット、マルチキャスト・パケットの受信) が反転することはありません。つまり、RJCM と BRD をセットしてブロードキャスト・パケットをすべてリジェクトすることはできません。ただし、RJCM と BRD を同時にセットすれば、すべてのブロードキャスト・パケットを受信することができますが、CAM アドレスに一致するデスティネーション・アドレスを持ったパケットはすべてリジェクトします。

注 1. PCNM と PCM を同時に“1”にセットしないでください。

2. PCNM と PCM がともに“0”だと、PCNM または PCM を変更するまで、 $\overline{\text{PCOMP}}$ 出力はハイ・インピーダンスを保持します。

3. μPD72950は13ポート対応リピータ・インタフェース・コントローラです。

4. デスティネーション・アドレスがブロードキャスト・アドレスであれば、受信制御レジスタの BRD ビットの状態にかかわらず、 $\overline{\text{PCOMP}}$ はロウ・レベルになりません。

5.3.8 送信レジスタ

送信レジスタは、ユーザ・レジスタの一部です。UTDA と CTDA は、コマンド・レジスタで送信コマンドを実行 (TXP ビットをセット) する前に初期化しなければなりません。

(1) 上位送信ディスクリプタ・アドレス・レジスタ (UTDA)

このレジスタには、送信ディスクリプタ・エリア (TDA) にアクセスする上位アドレス・ビット (A31-A16) を格納しており、μPD72932がシステム・メモリ上の TDA にアクセスするときに、CTDA と連結して32ビット・アドレスを生成します。TDA は最大32 Kワードまたは16 Kのロング・ワード・サイズにすることができ、システム・メモリのどこにでも置くことができます。このレジスタはハードウェア・リセットおよびソフトウェア・リセットの影響を受けません。

(2) カレント送信ディスクリプタ・アドレス・レジスタ (CTDA)

この16ビットの CTDA レジスタには、32ビットの送信ディスクリプタ・アドレスの下位アドレス・ビット (A15-A1) を格納します。初期化時には、このレジスタに送信ディスクリプタの下位アドレス・ビットをプログラムします。μPD72932はこのレジスタの内容を UTDA の内容と連結して、送信ディスクリプタを示します。32ビット・メモリ・システムのビット 1 はアドレス信号 A1 に対応しており、ロング・ワードの境界に配置するには、A1 に“0”をセットします。このレジスタのビット 0 は、エンド・オブ・リスト (EOL) ビットで、リストの終わりを示します。このレジスタはハードウェア・リセットおよびソフトウェア・リセットの影響を受けません。

5.3.9 受信レジスタ

受信レジスタは、ユーザ・レジスタの一部です。ソフトウェア・リセットは、これらのレジスタに影響を与えず、ハードウェア・リセットは、EOBC および RSC レジスタだけに影響を与えます。受信レジスタは、コマンド・レジスタで受信コマンドを実行 (RXEN ビットをセット) する前に初期化してください。

(1) 上位受信ディスクリプタ・アドレス・レジスタ (URDA)

このレジスタには、受信ディスクリプタ・エリア (RDA) をアクセスするための上位アドレス・ビット (A31-A16) を格納しており、μPD72932がシステム・メモリの RDA にアクセスするときに、CRDA に連結します。RDA は最大32 Kワードまたは16 Kのロング・ワード・サイズにすることができ、またシステム・メモリのどの位置にでも置くことができます。このレジスタは、ハードウェア・リセットおよびソフトウェア・リセットの影響を受けません。

(2) カレント受信ディスクリプタ・アドレス・レジスタ (CRDA)

CRDA は16ビットのリード/ライト・レジスタで、受信したパケットのディスクリプタ・ロケーションを RDA 内に配置するのに使用します。CRDA には下位アドレス・ビット (A15-A1) を格納します。μPD72932は CRDA の内容を URDA の内容に連結して、完全な32ビット・アドレスを形成します。この32ビット・アドレスは、次に受信パケットを格納するディスクリプタ・ブロックの最初のフィールドを示します。32ビット・メモリ・システムでは、アドレス信号 A1 に対応するビット 1 は、ロング・ワードの境界に配置するために必ず“0”をセットしてください。このレジスタのビット 0 はエンド・オブ・リスト (EOL) ビットで、リストの終わりを示すために使用します。このレジスタはハードウェア・リセットおよびソフトウェア・リセットの影響を受けません。

(3) エンド・オブ・バッファ・カウント・レジスタ (EOBC)

μPD72932はこのレジスタの内容を使用して、次のパケットを同じRBAに置くか、別のRBAに置くか判断します。パケット受信の終了時、μPD72932はEOBCレジスタの内容とリメイニング・バッファ・ワード・カウント・レジスタ (RBWC0, 1) の内容を比較して、次のことを決定します。

- (a) 次のパケットを同じRBAの中に置く
- (b) 次のパケットを別のRBAの中に置く

パケットを受信後、EOBCがRBAの残りのワード数以下 ($EOBC \leq RBWC0, 1$) の場合、μPD72932は次のパケットを同じRBA内にバッファします。EOBCがRBAの残りのワード数よりも大きい ($EOBC > RBWC0, 1$) 場合、μPD72932はRBA内の最終パケット (**5.3.3 受信制御レジスタのLPKTビット**) をセットし、次のリソース・ディスクリプタをフェッチし、次の受信パケットを新しいRBAにバッファします。ハードウェア・リセットを実行すると、このレジスタは02F8H (760ワード、あるいは1520バイト) にセットされます。EOBCの使用についての詳細は、**4.4.2**と**4.4.4 (4)**を参照してください。

(4) 上位受信リソース・アドレス・レジスタ (URRA)

URRAは16ビットのリード/ライト・レジスタで、受信リソース・エリア (RRA) のベース・アドレスをプログラムします。この16ビットの上位アドレス値 (A31-A16) は、受信リソース・エリアをシステム・メモリ上のどこに配置するかを決めます。μPD72932は、4つの受信リソース・レジスタ (RSA, REA, RWP, またはRRP) のいずれか1つの下位アドレス値とURRAを連結して、受信リソース・エリアをアクセスします。

(5) リソース・スタート・アドレス・レジスタ (RSA)

RSAは15ビットのリード/ライト・レジスタです。LSBは使用せず、常に0が読み出されます。RSAには、受信リソース・エリアの開始アドレスの下位15ビット・アドレス (A15-A1) をプログラムします。μPD72932は、このレジスタの内容をURRAの内容と連結して、完全な32ビット・アドレスを形成します。

(6) リソース・エンド・アドレス・レジスタ (REA)

REAは15ビットのリード/ライト・レジスタです。LSBは使用せず、常に0が読み出されます。REAには、受信リソース・エリアの終了アドレスの下位15ビット (A15-A1) がプログラムされます。μPD72932は、このレジスタの内容をURRAの内容に連結して、完全な32ビットのアドレスを形成します。

(7) リソース・リード・ポインタ・レジスタ (RRP)

RRPは15ビットのリード/ライト・レジスタです。LSBは使用せず、常に0が読み出されます。RRPには、μPD72932が読み出す次のリソース・エリアの最初のフィールドの下位15ビット・アドレス (A15-A1) をプログラムします。μPD72932はこのレジスタの内容をURRAの内容と連結して、完全な32ビットのアドレスを形成します。

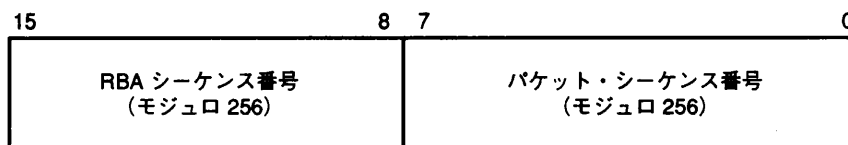
(8) リソース・ライト・ポインタ・レジスタ (RWP)

RWPは15ビットのリード/ライト・レジスタです。LSBは使用せず、常に0で読み出されます。RWPには、このシステムがリソース・エリアを付加できる次の有効ロケーションの下位15ビット・アドレス (A15-A1) をプログラムします。μPD72932はこのレジスタの内容をURRAの内容と連結して、完全な32ビット・アドレスを形成します。32ビット・モードでは、アドレス信号A1に対応するビット1を“0”にして、このレジスタとRRPレジスタの比較が適切に行われるようにします。

(9) 受信シーケンス・カウンタ・レジスタ (RSC)

RSC は 2つのフィールドを含む16ビットのリード/ライト・レジスタです。μPD72932はこのレジスタで、RBA 内のパケット数に関するカウント情報と RBAの番号を示します。RSC レジスタは 2つの 8 ビット (256ビット 構成) カウンタで構成されています。各パケットを受信するたびに、パケット・シーケンス番号をインクリメントします。μPD72932は 1つの RBA に対し 1つの RBA シーケンス番号を維持します。μPD72932が次の RBA を使用するときは、パケット・シーケンス番号を 0 にリセットし、RBA シーケンス番号をインクリメントします。このレジスタはハードウェア・リセット、またはレジスタに 0 を書き込むとリセットされます。ソフトウェア・リセットの影響は受けません。

図 5-11 受信シーケンス・カウンタ・レジスタ



5.3.10 CAM レジスタ

CAM レジスタは、ユーザ・レジスタの一部です。CAM レジスタは、パケットのアドレス・フィルタリングをするコンテンツ・アドレス・メモリ (CAM) のエントリをプログラムするために使用します。これらのレジスタは、CAM イネーブル・レジスタを除いて、ハードウェア・リセットおよびソフトウェア・リセットの影響を受けません。

(1) CAM エントリ・ポインタ・レジスタ (CEP)

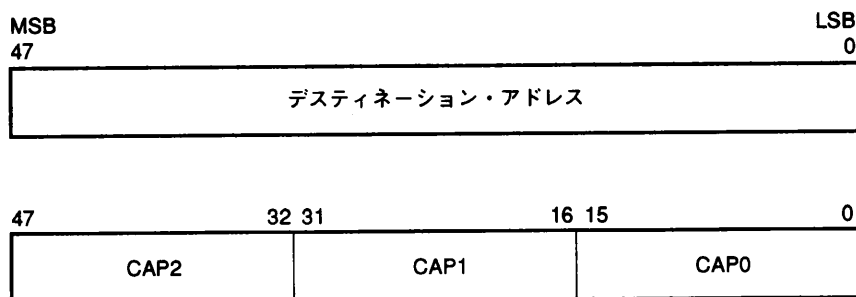
CEP は、μPD72932が16の CAM セル・エントリの 1つを選択するのに使用する 4ビット・レジスタです。μPD72932はこのレジスタの最下位 4 ビットを使用します。0H の値が最初の CAM エントリを示し、FH の値は最後のエントリを指し示します。

(2) CAM アドレス・ポート 2, 1, 0 レジスタ (CAP2, CAP1, CAP0)

各 CAP は 16ビットのリード・オンリ・レジスタで、CAMセルのアクセスに使用します(図 5-12)。各 CAM セルは 16ビット幅で、μPD72932がアドレス・フィルタリングに用いる 48ビットの CAM エントリの 1/3が含まれます。CAP2, CAP1, および CAP0 レジスタは、それぞれ CAM エントリの上位ビット (47-32)、中位ビット (31-16) および下位ビット (15-0) をアクセスするのに使用します。例えば 10 : 20 : 30 : 40 : 50 : 60 の物理アドレスは 6オクテットまたはバイトからなり、10H は最下位バイト、60H は最上位バイトを示します (10H はネットワークから受信する最初のバイト、60H は最後のバイト)。この物理アドレスの場合、CAP0, CAP1, CAP2 には、それぞれ 2010H, 4030H, 6050H をロードします。

CAM エントリを読み出すには、ユーザは最初に μPD72932をソフトウェア・リセット (コマンド・レジスタの RST ビットをセット) し、CEP レジスタをプログラムして 16の CAM エントリの 1つを選択し、CAP2, CAP1, および CAP0 を読み出して、全 48ビットを得るようにします。ユーザは、CAM エントリに直接書き込むことはできません。その代わりに、CAM ディスクリプタ・エリアをシステム・メモリにプログラムし (5.1.2 参照)、次にロード CAM コマンドをセット (コマンド・レジスタの LCAM ビットをセット) します。これによって、μPD72932にメモリからディスクリプタを読み出させ、CAP2-CAP0 に対応する CAM エントリをロードします。

図 5-12 CAMアドレス・ポート2,1,0レジスタ

**(3) CAM イネーブル・レジスタ (CE)**

CEは16ビットのリード/ライト・レジスタで、個々のCAM エントリをマスク・アウトまたはイネーブルするのに使用します。レジスタの各ビット位置はCAM エントリに対応しています。各レジスタ・ビットを“1”にセットすると、対応するCAM エントリがイネーブルされ、“0”にセットするとCAM エントリがディスエーブルされます。このレジスタはソフトウェア・リセットの影響を受けませんが、ハードウェア・リセット時は0にクリア（すべてのエントリがディスエーブル）されます。通常動作時には、ユーザはこのレジスタにアクセスしません。その代わりに、ユーザはCAM ディスクリプタ・エリア中の最後のエントリを通して、このレジスタをセットします。

(4) CAM ディスクリプタ・ポインタ・レジスタ (CDP)

CDPは15ビットのリード/ライト・レジスタです。LSBは使用せず、常に0が読み出されます。CDPには、システム・メモリのCAM ディスクリプタ・エリア (CDA) にあるCAM ディスクリプタ・ブロックの最初のフィールドの下位アドレス (A15-A1) をプログラムします。μPD72932は、CAM ディスクリプタにアクセスするときに、CDP レジスタの内容を使用します。このレジスタは、LCAM コマンドをセットする前に、ユーザがプログラムしなければなりません。LCAM コマンド実行中に、μPD72932はこのレジスタの内容をURRAレジスタの内容に連結して、完全な32ビット・アドレスを形成します。LCAM 実行中、このレジスタはインクリメントされCDAのフィールドにアクセスします。LCAM コマンドが完了したあと、このレジスタはCAM ディスクリプタ・エリアの次のロケーションを指し示します。

(5) CAM ディスクリプタ・カウント・レジスタ (CDC)

CDCは5ビットのリード/ライト・レジスタです。CDCにはCAM ディスクリプタ・エリアのCAM ディスクリプタ・ブロック数をプログラムします。このレジスタは、LCAM コマンドをセットする前に、ユーザがプログラムしなければなりません。μPD72932はこのレジスタ値を用いて、LCAM コマンドの実行中に、CAMの中にいくつのエントリを置くかを決定します。LCAM 実行中に、μPD72932はディスクリプタ・ブロックを読み出すたびに、このレジスタをデクリメントします。CDCが0になると、最後のCAM イネーブル・エントリの値をCEレジスタにロードし、μPD72932はLCAMの実行を終了します。CDCレジスタには、CAM ディスクリプタ・エリアのCAM ディスクリプタ・ブロック数をプログラムするため、CDCレジスタにプログラムする値の範囲は、1から16 (1Hから10H) になります。

5.3.11 タリー・カウンタ

μPD72932は、CRC エラー、フレーム配列エラー、および消失パケット数に関するネットワークの統計をモニターするために使用する3個の16ビット・カウンタを持っています。これらのレジスタは、FFFFHのカウンタに達したあとにロール・オーバーし、割り込みマスク・レジスタ (IMR) でイネーブルされていれば、割り込みを発生します。

これらのカウンタは、CR の RXEN ビットの影響を受けませんが、CR の RST ビットがセットされると停止します。これらのレジスタに書き込まれたデータは、ラッチされる前に反転します。このことは、システムによって FFFFH の値がこれらのレジスタに書き込まれると、実際には 0000H の値がレジスタに書き込まれることを意味しています。リード動作の間、データは反転しません。したがって、このタリー・カウンタは、すべてのビットに“1”を書き込むことによってクリアされます。ソフトウェア・リセットおよびハードウェア・リセットは、このタリー・カウンタに影響を与えません。

(1) CRC タリー・カウンタ・レジスタ (CRCT)

CRCT は16ビットのリード/ライト・レジスタです。このレジスタは、CRC エラー付きの受信パケット数を確実にカウントするために使用します。パケットがアドレス認識ロジックで確認され受け入れが始まったあと、CRC エラーが検出された場合、このレジスタをインクリメントします。このパケットにフレーム配列エラーも含む場合、このカウンタはインクリメントしません。

(2) FAE タリー・カウンタ・レジスタ (FAET)

FAET は16ビットのリード/ライト・レジスタです。このレジスタは、フレーム配列エラーを含む受信パケット数を確実にカウントするために使用します。パケットがアドレス認識ロジックで確認され受け入れが始まったあと、FAE エラーを検出した場合、このレジスタをインクリメントします。

(3) 消失パケット・タリー・カウンタ・レジスタ (MPT)

MPT は16ビットのリード/ライト・レジスタです。パケット受信後、このカウンタは次の場合にインクリメントされます。

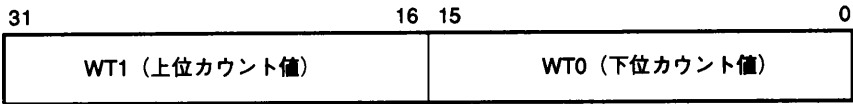
- (a) パケットをバッファするメモリ・リソースの不足
- (b) FIFO のオーバラン
- (c) 有効パケットが受信されたにもかかわらず、レシーバがディスエーブルされていた (コマンド・レジスタの RXDIS がセット) 場合

5.3.12 汎用タイマ

μPD72932は、ユーザが決めるイベントをカウントするための32ビットの汎用ウォッチドッグ・タイマを内蔵しています。ユーザは、2つの16ビットのリード/ライト・レジスタ（WT1 および WT0）を通して、このタイマにアクセスします。下位のカウンタ値は WT0 レジスタを使用してプログラムされ、上位のカウンタ値は WT1 レジスタを使用してプログラムされます。

これらの2つのレジスタを連結すれば、完全な32ビット・タイマが形成されます。このタイマは送信クロック（TXC）周波数の1/2のクロックが使用され、プログラムした値からカウント・ダウンし、0000 0000H から FFFF FFFFH にロール・オーバーするとき、割り込みがイネーブルであれば（割り込みマスク・レジスタで）割り込みが発生します。カウンタがロール・オーバーしたあとは、プログラムで停止させる（STP ビットをセット）場合を除いて、デクリメントし続けます。このタイマはコマンド・レジスタの ST（タイマ 開始）および STP（タイマ停止）ビットで制御できます。ハードウェア・リセットおよびソフトウェア・リセットによってこの汎用タイマは停止しますが、クリアはされません。

図 5－13 汎用タイマ



5.3.13 シリコン・リビジョン・レジスタ（SRR）

これは16ビットのリード・オンリ・レジスタです。このレジスタには、μPD72932の現在のリビジョン（改訂）に関する情報が含まれています。

バージョン	SRR の値
B	4H
C	6H

5.4 レジスタの初期化について

μPD72932の起動時には、電源投入、ハードウェア・リセットに続いて内部レジスタの初期化を行います。レジスタ初期化の手順について次に示します。

- (1) ソフトウェア・リセット・モードにします (CR ← 0080H)。
- (2) DCR および DCR2 をハードウェア仕様に従って設定します。
- (3) IMR および RCR をソフトウェア仕様に従って設定します。
- (4) ISR および各タリー・カウンタ (CRCT, FAET, MPT) をクリアします。
- (5) メモリ空間上に CAM ディスクリプタを作成します。
- (6) CAM ディスクリプタ情報を基に URRRA, CDP, CDC を設定します。
- (7) ソフトウェア・リセット・モードを解除します (CR ← 0000H)。
- (8) LCAM コマンドを発行します (CR ← 0200H)。
- (9) ISR をモニタし、LCD ビット = “1” を確認します。
- (10) ソフトウェア・リセット・モードにします (CR ← 0080H)。
- (11) CEP をセットし、CAP0-CAP2 および CE を読み出し、CAM の内容を確認します。
- (12) メモリ空間上に RRA ディスクリプタを生成します。
- (13) RRA ディスクリプタ情報を基に URRRA, RSA, REA, RWP, RRP を設定します。
- (14) RSC をクリアします。
- (15) ソフトウェア・リセット・モードを解除します (CR ← 0000H)。
- (16) RRRRA コマンドを発行します (CR ← 0100H)。
- (17) メモリ空間上に RDA ディスクリプタを生成します。
- (18) RDA ディスクリプタ情報を基に URDA, CRDA を設定します。
- (19) RXEN コマンドを発行します (CR ← 0008H)。

この処理によって、μPD72932を受信可能な状態にすることができます。

注意 DCR, DCR2 レジスタの設定は、必ずリセット・モード時に行ってください。また、LCAM, RRRRA, TXP コマンドの発行は、必ずリセット・モード解除時に行ってください。

メモリ空間上のディスクリプタの配置例を図 5-14 に示します。また、そのディスクリプタに基づいた初期設定フローの例を図 5-15 に示します。この設定例は、BMODE=0, 16ビット・バス・モードの場合の応用例です。各ディスクリプタをメモリ上に作成し、その配置に従って各レジスタを設定しています。

図 5-14 ディスクリプタ配置例

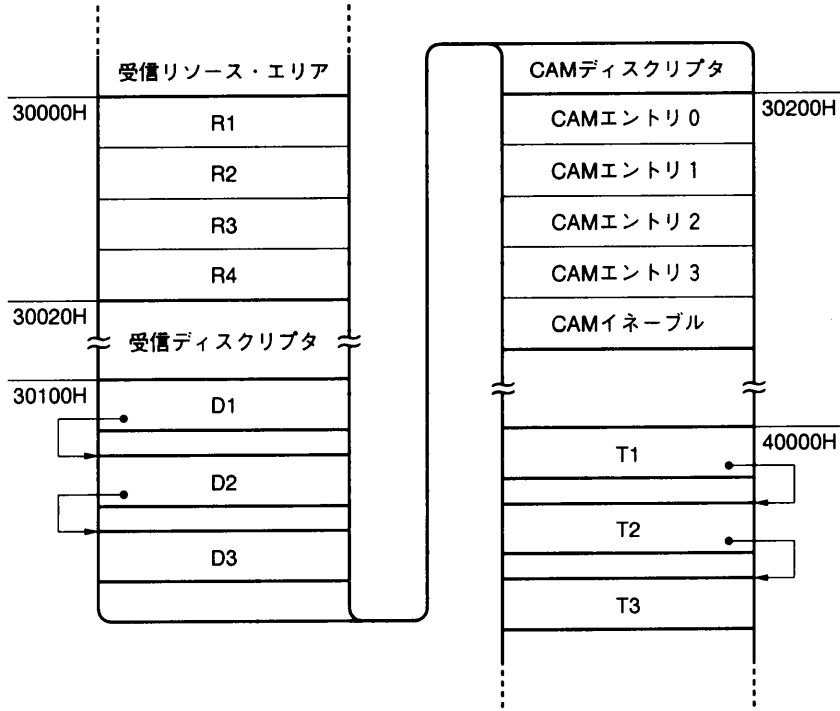


図 5-15 初期設定フロー・チャート (1/3)

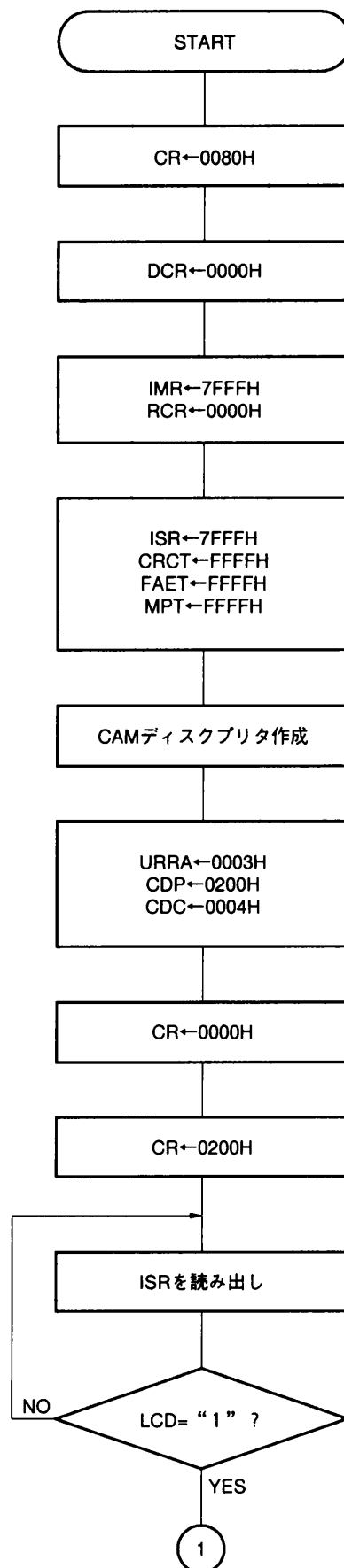


図 5-15 初期設定フロー・チャート (2/3)

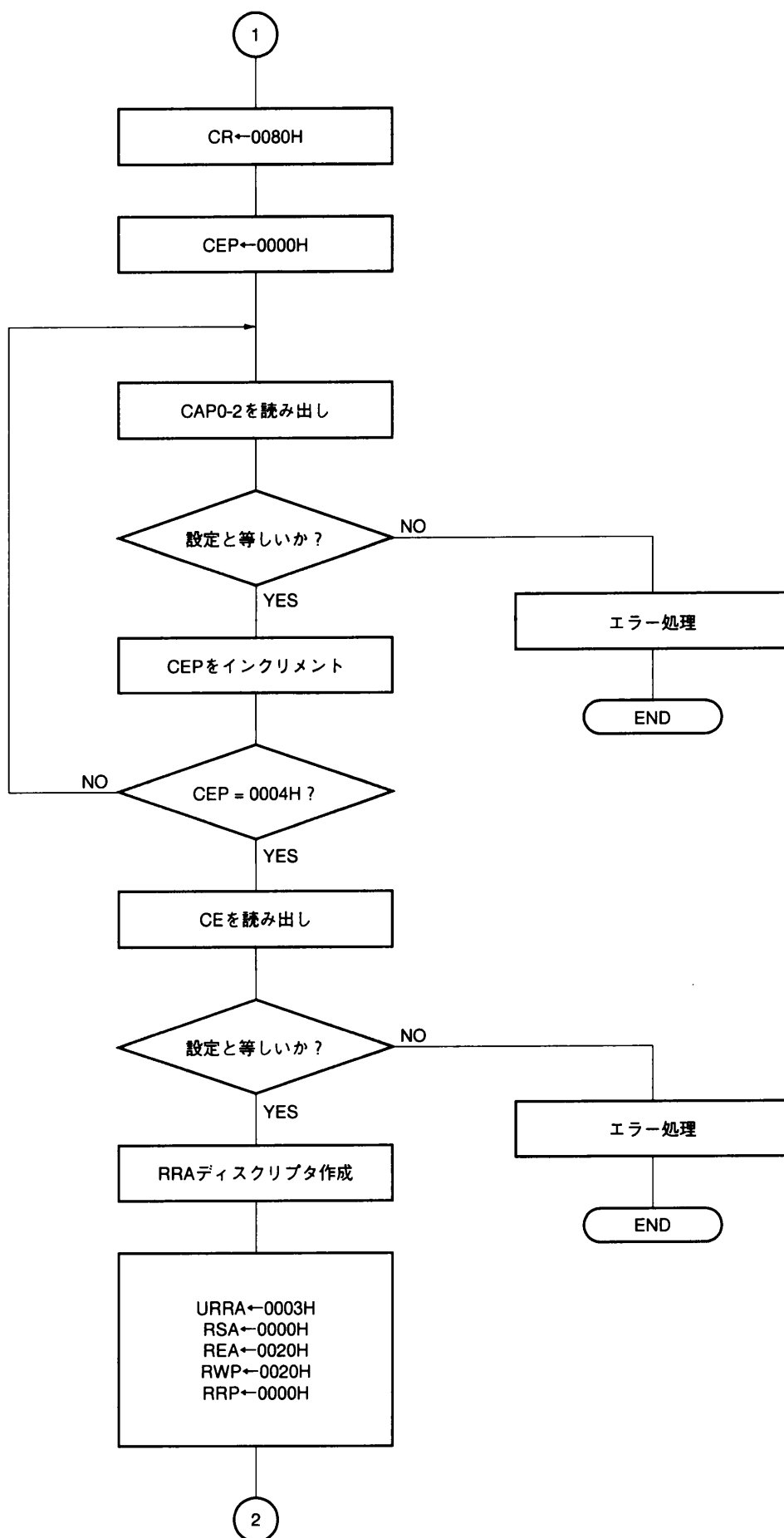
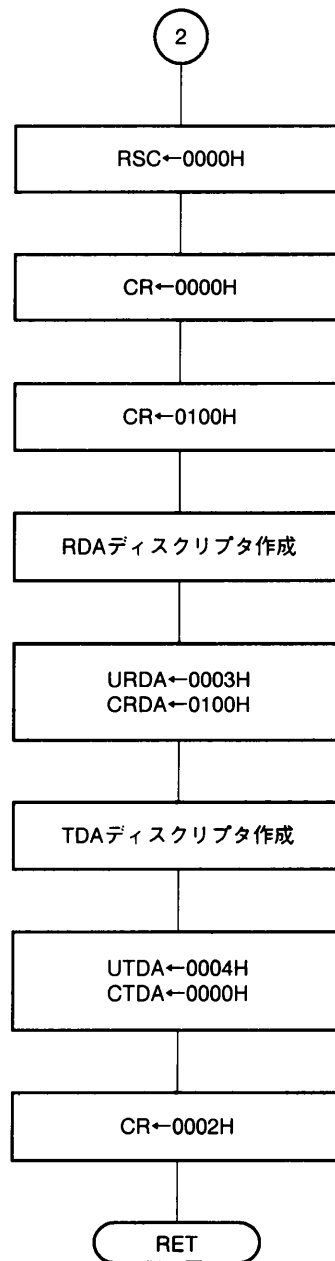


図 5-15 初期設定フロー・チャート (3/3)



6. バス・インタフェース

μ PD72932は、広範なシステム環境に対応するように設計された、高速ノンマルチプレクス・アドレスおよびデータ・バスを持っています。データ・バスは（データ・コンフィギュレーション・レジスタを使用して）、32ビット幅または16ビット幅のいずれにもプログラムできます。 μ PD72932はオンチップDMAを内蔵し、DMA動作に必要なすべての信号を提供します。 μ PD72932は31本のアドレス・ラインによって、2Gワードのアドレス空間すべてにアクセスすることができます。さまざまなメモリ・スピードに対応するために、2通りの方法によって、バス・サイクルにウェイト・ステートを追加できます。1つは、 $\overline{\text{RDYi}}$ (BMODE=0) または $\overline{\text{DSACK0}}$, $\overline{\text{DSACK1}}$ (BMODE=1) をインアクティブにする方法です。この方法によって、簡単にウェイト・ステートを追加できます。もう1つの方法として、データ・コンフィギュレーション・レジスタをプログラムすることによりウェイト・ステートを追加することもできます。

μ PD72932は、ナショナル セミコンダクター (NSC) /NEC/インテルおよびモトローラ型のバスの両方にインタフェースできるように設計されています。最小のチップ数による設計、および完全なバスの互換性を実現する設計のために、ユーザは μ PD72932を次のバス・モードにプログラムできます。

- 同期モードで動作する NSC/NEC/インテル型バス
- 非同期モードで動作する NSC/NEC/インテル型バス
- 同期モードで動作するモトローラ型バス
- 非同期モードで動作するモトローラ型バス

バス・モードの選択は、データ・コンフィギュレーション・レジスタの SBUS ビットとモード端子 (BMODE) を組み合わせて行います。

ここでは、 μ PD72932のシステム・インタフェース例を提示し、各種のバス動作について説明します。

6.1 端子構成

μ PD72932には、ユーザが選択可能な2つの端子構成があり、NSC/NEC/インテルまたはモトローラ型バスのいずれかに、適合するインタフェース信号を供給することができます。BMODE 端子を用いて、各モードの端子構成を設定します。BMODE=1 (V_{CC} に接続) でモトローラ型バスに対応します。BMODE=0 (グランドに接続) で NSC/NEC/インテル型バスに対応します。詳細は端子接続図および 1. 端子機能を参照してください。

6.2 システム構成

μ PD72932のインタフェース・プロトコルおよび電氣的条件（タイミング、スレッシュホールド、および負荷）を満たすマイクロプロセッサと μ PD72932とをインタフェースすることができます。BMODE 端子によって選択可能な2つのバス・プロトコルが用意されているため、ほとんどのマイクロプロセッサに、 μ PD72932を直接インタフェースすることができます。図 6-1 に、NSC/NEC/インテル型バス (BMODE=0) への代表的なインタフェースを示します。また、図 6-2 には、モトローラ型バス (BMODE=1) への代表的なインタフェースを示します。

BMODE 端子は、バイト順序も制御します。BMODE=1 のときには、ビッグ・エンディアン・バイト順序が選ばれ、BMODE=0 のときには、リトル・エンディアン・バイト順序が選ばれます。

図 6-1 μPD72932とV810™ (μPD70732) のインタフェース例

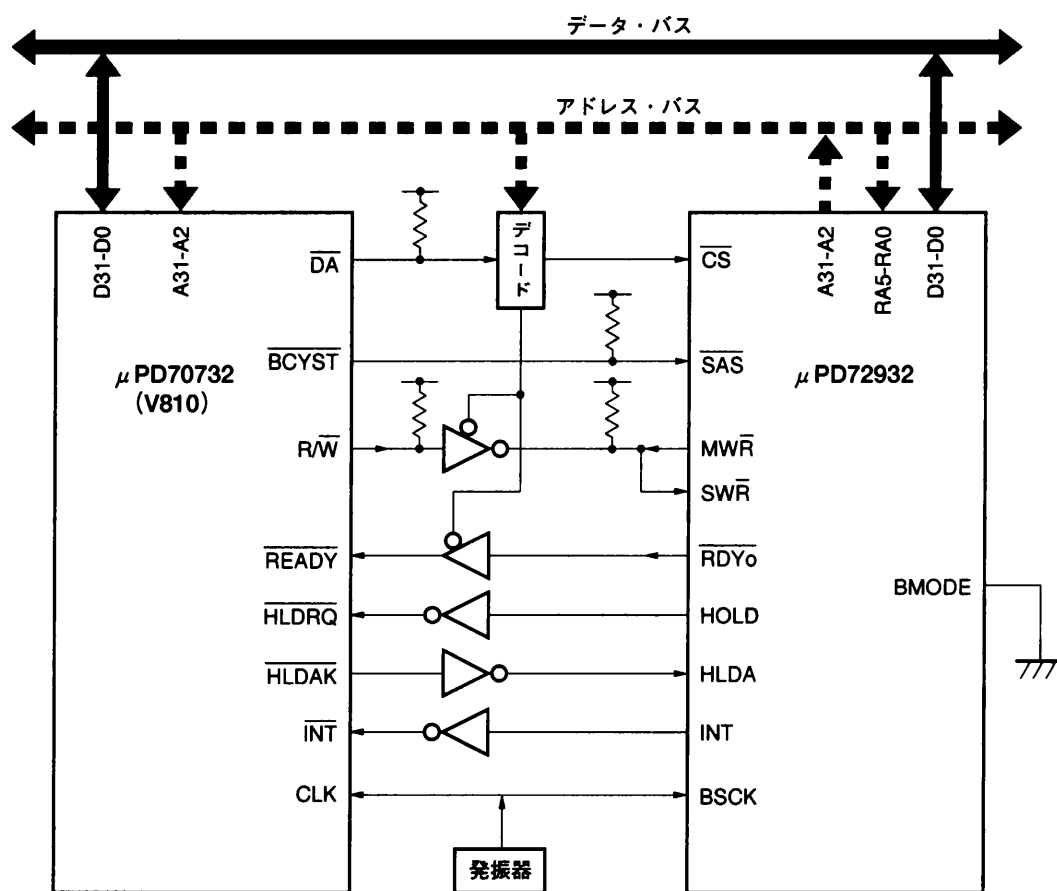
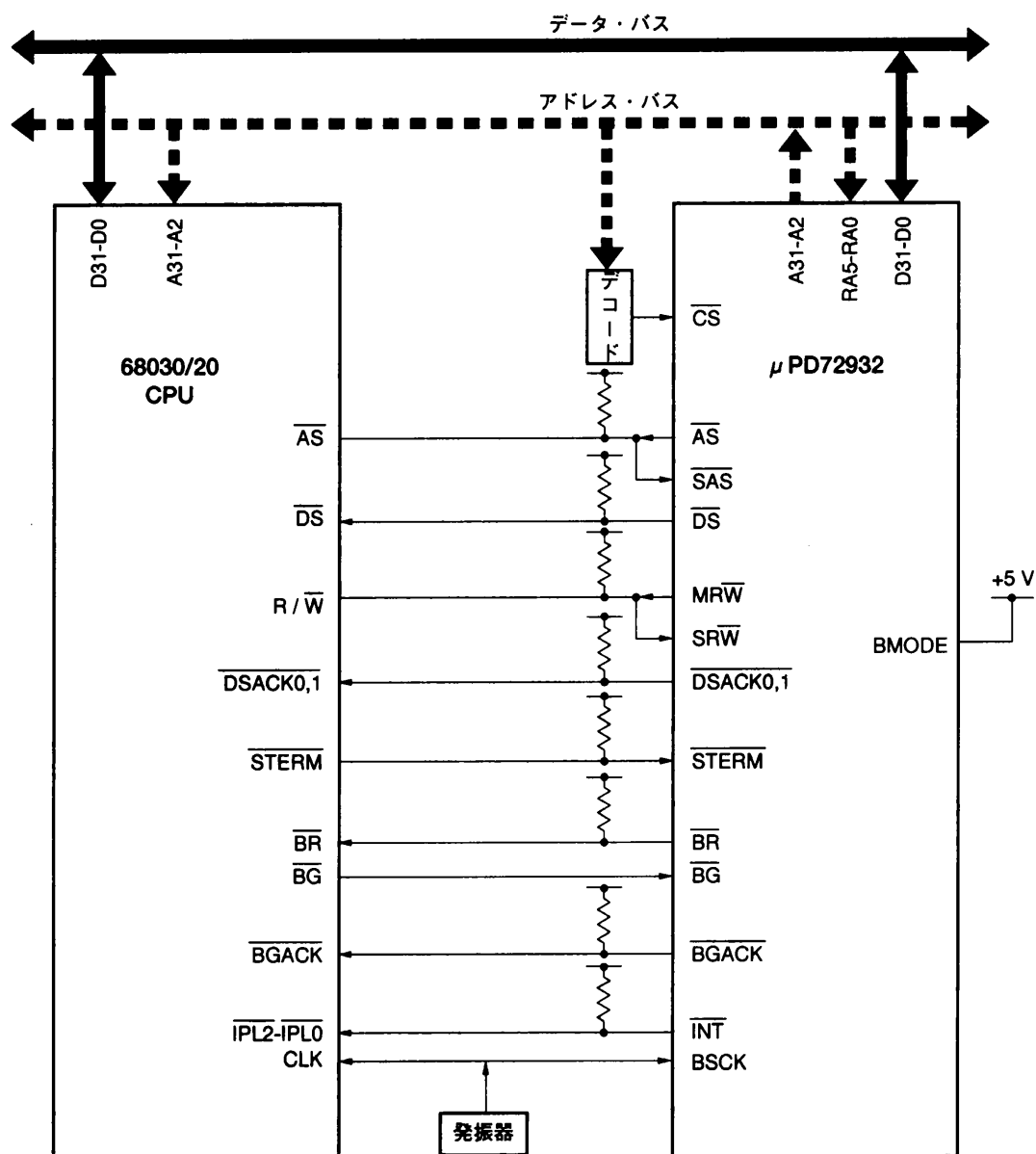


図 6-2 μPD72932とモトローラ社製68030/20のインタフェース例



6.3 バス・オペレーション

μPD72932のシステム・バスのオペレーションには、(1) スレーブと(2) バス・マスタの2つのタイプがあります。μPD72932がスレーブのオペレーションをするとき(たとえば、CPUがμPD72932レジスタをアクセスしているとき)、DMA転送はしません。μPD72932がバス・マスタのオペレーションをするとき(たとえば、μPD72932が受信または送信バッファ/ディスクリプタ・エリアをアクセスするとき)、転送はすべてμPD72932のオンチップDMAを用いたブロック転送となります。

このセクションでは、μPD72932バス動作について説明します。注意事項がある箇所は特に気をつけてください。正しいバス動作を保証するために、それらの条件を満足しなければなりません。

6.3.1 バスの獲得

μPD72932は、(1) FIFO中のデータの数スレッシュホールド値に達したとき、または(2) メモリのディスクリプタ・エリア(つまり、RRA, RDA, CDA, およびTDA)をアクセスするときにバスを要求します。μPD72932がメモリのあるエリアから別のエリア(たとえば、RBAからRDA)に転送するときには、常にバス要求をいったんインアクティブにしてから、メモリの次のエリアをアクセスするときに再度バスを要求することに注意してください。

μPD72932はNSC/NEC/インテルまたはモトローラ型のマイクロプロセッサと互換性を持たせるために、バスを要求する方法を2つ備えています。これらの方法は、BMODE端子に適切なレベルを設定することによって選択します。

図6-3と図6-4にNSC/NEC/インテル型(BMODE=0)とモトローラ型(BMODE=1)のバス要求タイミングを示します。以下に各モードの説明を行います。両モードでは、μPD72932がバスを放棄するとき、最後のDMAサイクル(T2)から1バス・サイクルの間余分なホールド・ステート(Th)があります。これは、μPD72932がバス解放後、別のバス・マスタと競合しないようにするためのものです。

(1) BMODE=0

NSC/NEC/インテル・プロセッサは、ホールド・リクエスト/ホールド・アクノリッジ・プロトコル(図6-3)を用いた、2方向ハンドシェイクを必要とします。μPD72932がバスへのアクセスを必要とするときには、マイクロプロセッサに対してホールド・リクエスト(HOLD)でバスを要求します。マイクロプロセッサはバスの使用权を与えるため、ホールド・アクノリッジ(HLDA)でμPD72932に応答します。すると、μPD72932は該当するバスでメモリ転送を開始します。CPUがHLDAをアクティブに維持する限り、μPD72932はメモリ・ブロック転送が完了するまでそれを継続します。しかし、CPUはμPD72932がHOLDをロウ・レベルにする前に、HLDAをロウ・レベルにすることによって、μPD72932がブロック転送を完了する前にバスを先取りすることができます。これによって、高い優先順位をもつデバイスが、μPD72932によるバスの使用を強制的に中断させてバスを解放させることができるようになります。μPD72932は中断される前の動作を完了するために、あとで再度バス要求を行います。

図6-3に示すように、μPD72932はバス・クロック(BSCK)の立ち下がり、あるいは立ち上がりエッジでHOLDをハイ・レベルにします。デフォルトでは、立ち上がりエッジでHOLDをハイ・レベルにします。DCR2レジスタのPHビットをセット(5.3.7参照)すると、HOLDを立ち上がりエッジから1/2バス・クロック遅れてハイ・レベルにします(点線の部分)。HOLDをハイ・レベルにする前に、μPD72932はHLDAラインをチェックします。HLDAがハイ・レベルになると、HLDAがロウ・レベルになるまでHOLDをハイ・レベルにしません。

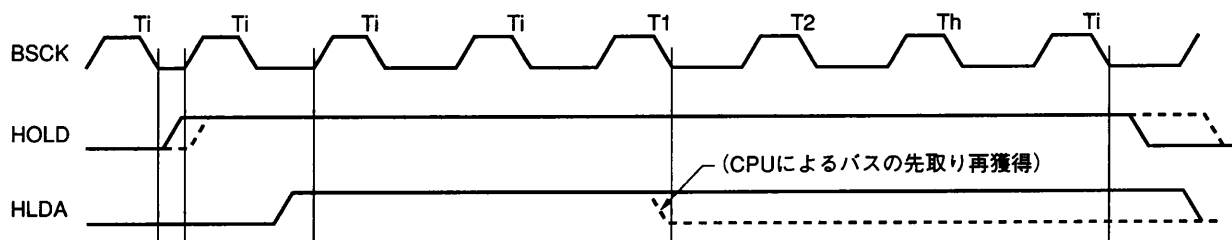
μPD72932がLCAMコマンドによってCAMエリアをアクセスしているときにCPUがHLDAをロウ・レベルにすると、μPD72932からバス使用权を強制的に取ることができます。このときμPD72932はバスから切り離され、バス・ステータス(S2-S0)がアイドル状態を示しますが、HOLDはロウ・レベルになりません。

μPD72932がそのほかのディスクリプタ・エリア(RRA, RDA, TDA)をアクセスしているときにHLDA

がロウ・レベルになると、 μ PD72932は正常にバスから切り離され、HOLD をロウ・レベルにします。HOLD はその 1 バス・クロック後に再びハイ・レベルになります。

μ PD72932がバッファ・エリア (RBA, TBA) をアクセスしているときに HLDA をロウ・レベルにすると、 μ PD72932は正常にバスから切り離され、HOLD をロウ・レベルにします。しかし、再びバッファ・エリアにアクセスする際、FIFO のスレッシュホールドの状態によっては HOLD が再びハイ・レベルにならないこともあります。

図 6-3 バス要求タイミング, BMODE = 0



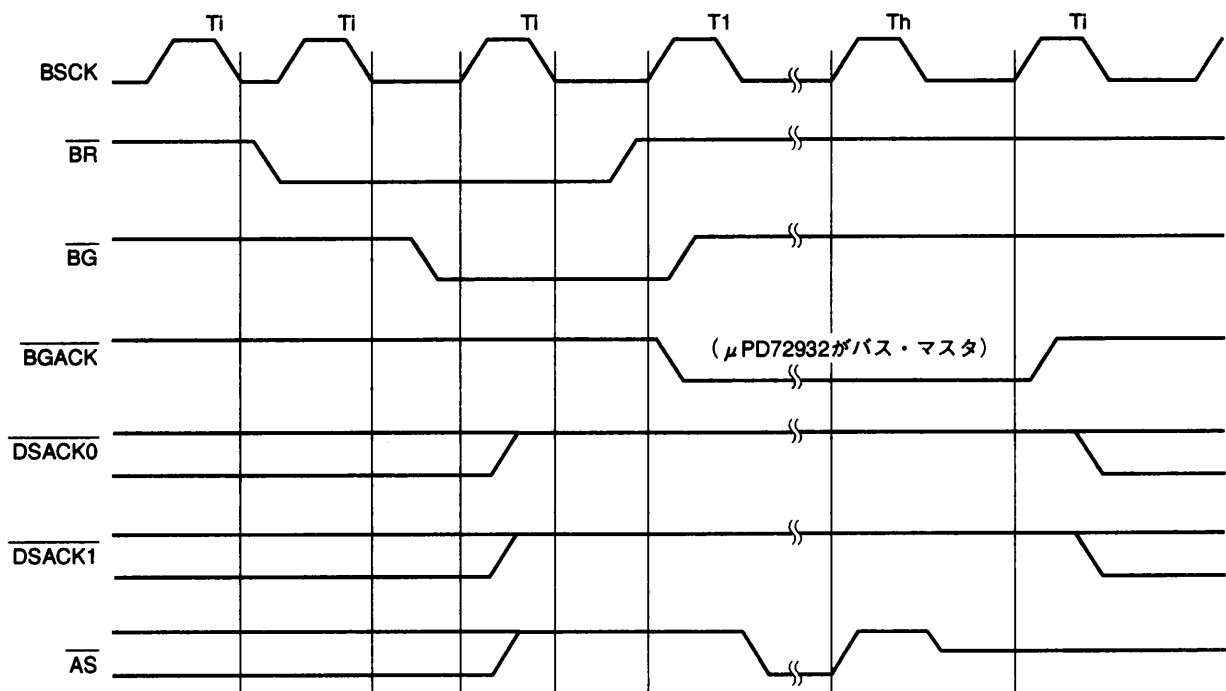
(2) BMODE=1

モトローラ・プロトコルは、バス要求、バス許可およびバス許可アクリッジ・ハンドシェークの 3 つの信号を用いた 3 方向ハンドシェークを必要とします (図 6-4)。このプロトコルを使用するときには、バス要求 ($\overline{\text{BR}}$) をロウ・レベルにしてバスを要求します。CPU はバス許可 ($\overline{\text{BG}}$) をロウ・レベルにしてこれに応答します。 μ PD72932は $\overline{\text{BG}}$ を受信することにより、バスを使用する前にすべてのデバイスがバスの使用权を放棄したことを確認します。 μ PD72932のバスの獲得前に、次の信号はインアクティブになっていなければなりません。

- $\overline{\text{BGACK}}$
- $\overline{\text{AS}}$
- $\overline{\text{DSACK0}}$, $\overline{\text{DSACK1}}$
- $\overline{\text{STERM}}$ (非同期モードの場合のみ)

$\overline{\text{BGACK}}$ のインアクティブ状態は前のマスタがバスを放棄したことを表します。 $\overline{\text{AS}}$ のインアクティブ状態は前のマスタがそのサイクルを完了したことを表します。 $\overline{\text{DSACK0}}$, $\overline{\text{DSACK1}}$ と $\overline{\text{STERM}}$ のインアクティブ状態は前のスレーブが前のマスタへの接続を終了したことを表します。 μ PD72932は自分で $\overline{\text{BGACK}}$ をインアクティブにするまで、バスの使用权を保持します。バスからのバス使用权の先取り再獲得はできません。

図 6-4 バス要求タイミング, BMODE = 1



6.3.2 ブロック転送

μPD72932はすべてのバス動作中に、ブロック転送を用いることによって、メモリへ効率的に転送を行います。ブロック・サイクルは、3つの部分で構成されています。最初の部分は前述したようなバスの獲得段階で、μPD72932はバスへのアクセス権を獲得します。第2に、バスにアクセスすると、μPD72932は内部FIFOとメモリ間、またはメモリとレジスタ間でデータ転送を行う段階に入ります。最後はバスを終了する段階です。データ転送時、μPD72932はブロック・モードまたはエンプティ/フィル・モードのいずれかで、FIFOからのデータを転送します。

(1) ブロック・モード

このモードでは、ブロック転送中に転送するワード（またはロング・ワード）数は、データ・コンフィギュレーション・レジスタにプログラムされた送信または受信FIFOのスレッシュホールド値のどちらかで決定します。

(2) エンプティ/フィル・モード

このモードでは、DMAは送信の場合、送信FIFOを完全に満たすか、または受信の場合、受信FIFOを完全に空にするまで続きます。これにより、μPD72932のバスの待ち時間を長くすることができます。

μPD72932はディスクリプタ・エリア（RRA, RDA, CDA, およびTDA）にアクセスするときには、レジスタとメモリ間でデータの転送を行います。使用する必要のあるすべてのフィールドを、1ブロック動作でアクセスします。したがって、1ブロック転送でμPD72932はRRA（4.4.4(2)参照）で4回のアクセス、RDA（4.4.6(1)参照）で7回のアクセス、TDA（4.5.4参照）で、2, 3, または6回のアクセス、そしてCDAで4回のアクセスを実行します。

6.3.3 バス・ステータス

μPD72932は現在実行中のバス動作のタイプを表す3ビットのステータス情報を、S2-S0端子に出力します（表6-1）。バス・ステータスは、 $\overline{AS}$ の立ち下がりエッジまたは $\overline{ADS}$ の立ち上がりエッジで有効になります。

表 6-1 バス・ステータス

S2	S1	S0	ステータス
1	1	1	このバス・サイクルはアイドルです。μPD72932はこのバス・サイクルでは転送をしません。
1	0	1	現在、送信ディスクリプタ・エリア (TDA) をアクセス中であることを表します。
0	0	1	現在、送信バッファ・エリア (TBA) を読み出し中であることを表します。
0	1	1	現在、受信バッファ・エリア (RBA) に書き込み中ですが、デスティネーションまたはソース・アドレス以外のデータの書き込みを行っています。
0	1	0	現在、受信バッファ・エリア (RBA) に書き込み中ですが、デスティネーションおよびソース・アドレスをメモリに書き込み中です。
1	1	0	現在、受信リソース・エリア (RRA) を読み出し中であることを表します。
1	0	0	現在、受信ディスクリプタ・エリア (RDA) にアクセス中であることを表します。
0	0	0	現在、CAM ディスクリプタ・エリア (CDA) にアクセス中であることを表します。

μPD72932はバスを獲得すると、メモリのいずれかのエリア (TDA, TBA, RDA, RBA, RRA, または CDA) との間でのみデータを転送します。したがって、バス・ステータス端子は、次の3つの場合を除いて、ブロック転送サイクル中は安定状態 (変化しない) となります。

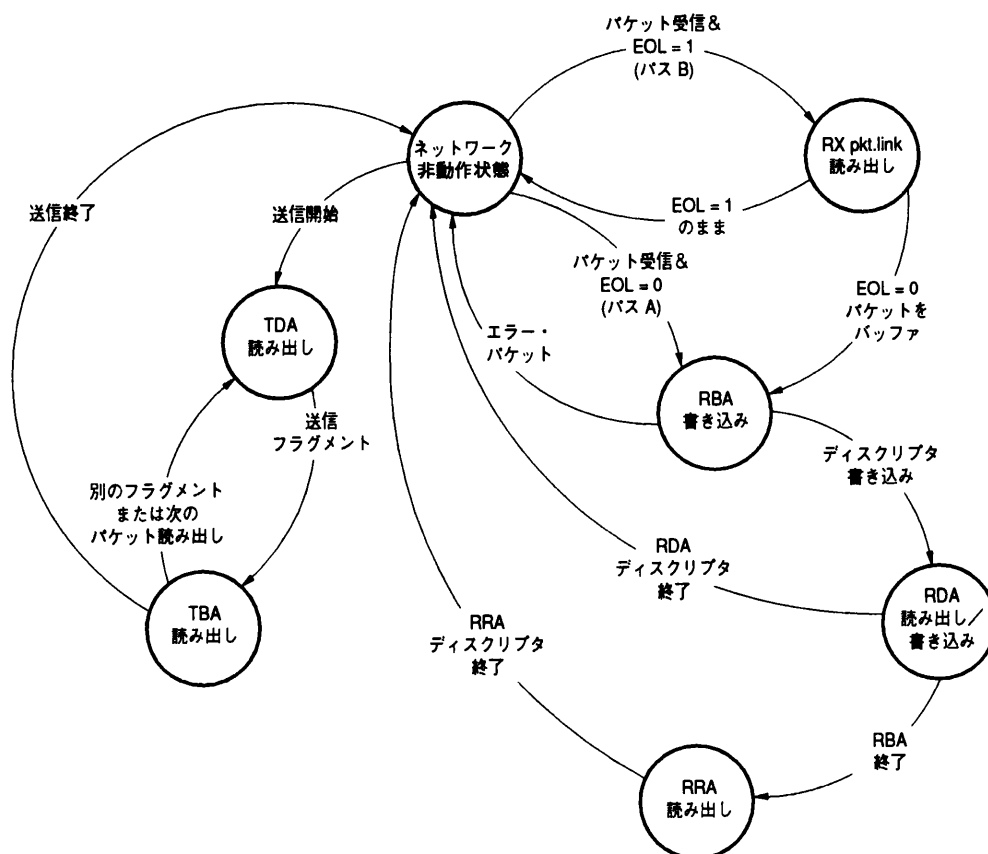
- (a) ブロック転送中に μPD72932の内部レジスタがアクセスされた場合、S2-S0 はレジスタ・アクセス中にバス・アイドルを示し、そのあとで前のステータスに戻ります。
- (b) ブロック転送中に、μPD72932が RBA に対してソース・アドレスの書き込みを終了した場合、S2-S0 は [0, 1, 0] から [0, 1, 1] に変化します。
- (c) 次に示すときに S2-S0 はそれぞれ2または1バス・クロックにわたりアイドル [1, 1, 1] を示します。

- RXpkt.seq-no へのアクセスと RXpkt.link へのアクセスとの間
- RXpkt.link へのアクセスと RXpkt.in_use へのアクセスとの間での RDA アクセス中

ステータスは $\overline{AS}$ の立ち下がリエッジまたは $\overline{ADS}$ の立ち上がりエッジで有効になります。

図 6-5 は、送信および受信処理中の、メモリに対する μPD72932の状態変化を示します。送信中、μPD72932は TDA からディスクリプタ情報を読み出ししてから、TBA にあるデータ・パケットを送信します。μPD72932はすべてのフラグメントとパケットの送信を終了するまで、TDA と TBA 間を繰り返してアクセスします。受信中には、μPD72932は2つのバスのうち1つを使用します。最初の場合 (バス A)、μPD72932は前の受信で EOL=0 を検出すると、受信したパケットを RBA にバッファし、次にディスクリプタ情報を RDA に書き込みます。RBA が空になった場合 (RBWCQ, 1 < EOBC) は、RRA のリソース・ディスクリプタを読み出します。2番目の場合 (バス B)、μPD72932は前の受信から EOL=1 を検出すると、再度 RXpkt.link フィールドを読み出し、システムが最後の受信以降に EOL ビットをリセットしたか確認します。システムが EOL ビットをリセットしていれば、μPD72932はバス A の場合と同様に、パケットをバッファします。そうでなければ、μPD72932は該当するパケットをリジェクトして、アイドルに戻ります。

図 6-5 バス・ステータスの変化



6.3.4 バス・モードの互換性

μPD72932には2種類のバス・モードがあり、複数の異なるマイクロプロセッサやバス・アーキテクチャとの互換性があります。バス・モードの設定はBMODE端子で行います。BMODE=0にすると、NSC/NEC/インテル・モード（リトル・エンディアン・モード）になります。BMODE=1にすると、モトローラ・モード（ビッグ・エンディアン・モード）になります。バス・モードによって、機能が変わる端子があります（表 6-2 参照）。バス・モードによる機能の変化はバス・マスタおよびスレーブ動作のどちらにも起こります。

表 6-2 バス・モードによる端子機能の変化

端子名	BMODE=0 (NSC/NEC/インテル)	BMODE=1 (モトローラ)
$\overline{\text{BR}}/\text{HOLD}$	HOLD	$\overline{\text{BR}}$
$\overline{\text{BG}}/\text{HLDA}$	HLDA	$\overline{\text{BG}}$
$\text{MR}\overline{\text{W}}/\text{M}\overline{\text{WR}}$	$\text{M}\overline{\text{WR}}$	$\text{MR}\overline{\text{W}}$
$\text{SR}\overline{\text{W}}/\text{S}\overline{\text{WR}}$	$\text{S}\overline{\text{WR}}$	$\text{SR}\overline{\text{W}}$
$\overline{\text{DSACK0}}/\text{RDY}_i$	RDY_i	$\overline{\text{DSACK0}}$
$\overline{\text{DSACK1}}/\text{RDY}_o$	RDY_o	$\overline{\text{DSACK1}}$
$\overline{\text{AS}}/\text{ADS}$	ADS	$\overline{\text{AS}}$
$\overline{\text{INT}}/\text{INT}$	INT	$\overline{\text{INT}}$

6.3.5 マスタ・モードのバス・サイクル

異なるバス・アーキテクチャとの互換性をさらに高めるために、バス・モードのほかにバス動作に影響する2つのモード（同期モードと非同期モード）があります。これらのモードはデータ・コンフィギュレーション・レジスタ（DCR）のSBUSビットでプログラムします。それぞれのバス動作は μ PD72932へのスレーブ・アクセスには影響しませんが、マスタ・モード動作に影響します。特定のバス/プロセッサ・モード内では、同期モードと非同期モードは極めて似ています。このセクションでは、バス・マスタ時の μ PD72932の4つの動作モード（NSC/NEC/インテルとモトローラ、同期と非同期）について説明します。

このセクションでは、T1 または T2 の立ち上がりエッジは T1 ステート、T2 ステートの始まりを意味し、T1 および T2 の立ち下がりエッジは T1 ステート、T2 ステートの中央を意味します。

(1) ウェイト・ステートの追加

異なるメモリ速度に対応するために、 μ PD72932は二通りの方法でバス動作にウェイト・ステートを追加することができます。これらの方法はどちらも、単独でまたは組み合わせて用いることができます。メモリ・サイクルは、T2 ステートを追加することによって拡張されます。

1つの方法は、 $\overline{\text{DSACK0}}$, $\overline{\text{DSACK1}}$, $\overline{\text{STERM}}$, または $\overline{\text{RDYi}}$ をインアクティブにすることによってウェイト・ステートを挿入します。もう1つの方法では、ソフトウェアによりウェイト・ステートをプログラムできます。データ・コンフィギュレーション・レジスタの WC0, WC1 ビットをプログラムすることにより、各メモリ・サイクルごとに1-3 ウェイト・サイクルを追加することができます。これらのウェイト・ステートは T1 と T2 バス・ステート間に挿入され、T2（ウェイト）バス・ステートと呼ばれます。 μ PD72932は、プログラムされたウェイト・ステートが過ぎるまで、 $\overline{\text{DSACK0}}$, $\overline{\text{DSACK1}}$, $\overline{\text{STERM}}$, $\overline{\text{RDYi}}$ をサンプルしません。したがって、プログラムされたウェイト・ステートを含んでいるバス動作を完了させるためには、プログラムされたウェイト・ステートではなく最後の T2 のサイクル終了時に、 $\overline{\text{DSACK0}}$, $\overline{\text{DSACK1}}$, $\overline{\text{STERM}}$, $\overline{\text{RDYi}}$ ラインをそれぞれロウ・レベルにします。この場合、唯一の例外は非同期モードです。非同期モードでは、最後のプログラムされたウェイト・ステート、すなわち T2（ウェイト）の間に $\overline{\text{DSACK0}}$, $\overline{\text{DSACK1}}$ や $\overline{\text{RDYi}}$ をロウ・レベルにします。詳細については、タイミング・チャートの信号タイミングを参照してください。プログラムされたウェイト・ステートはスレーブ・モードのバス・サイクルに影響しません。

(2) BMODE=1 の場合のメモリ・サイクル、同期モード

μ PD72932は T1 の立ち上がりエッジで $\overline{\text{ECS}}$ をロウ・レベルにして、メモリ・サイクルが開始したことを示し、アドレス (A31-A1), バス・ステータス (S2-S0), およびディレクション・ストローブ ($\overline{\text{MRW}}$) をセットし、残りのメモリ・サイクルの間これらのラインをドライブし続けます。 μ PD72932は、T1 の立ち下がりエッジで $\overline{\text{ECS}}$ をハイ・レベルにし、 $\overline{\text{AS}}$ をロウ・レベルにします。

同期モードでは、T2 の立ち上がりエッジで $\overline{\text{DSACK0}}$, $\overline{\text{DSACK1}}$ をサンプルし、正しくロウ・レベルでサンプルされるまで T2 ステートを繰り返します。正しい動作のためには、 $\overline{\text{DSACK0}}$, $\overline{\text{DSACK1}}$ はバス・クロックの立ち上がりエッジに対しセットアップおよびホールド時間条件を満たさなければなりません。

リード・サイクル（図 6-6）では、T1 の立ち下がりエッジで $\overline{\text{DS}}$ をロウ・レベルにし、T2 の立ち下がりエッジでデータ (D31-D0) をラッチします。ライト・サイクル（図 6-7）では、T1 の立ち下がりエッジでデータをドライブします。挿入するウェイト・ステートがあると、T2（ウェイト）の立ち下がりエッジで $\overline{\text{DS}}$ をロウ・レベルにします。0 ウェイト・ステートのライト・サイクルでは、 $\overline{\text{DS}}$ をロウ・レベルにしません。 μ PD72932は、T2 の立ち下がりエッジで $\overline{\text{AS}}$ と $\overline{\text{DS}}$ をハイ・レベルにして、メモリ・サイクルを終了します。

図 6-6 メモリ・リード, BMODE = 1, 同期モード (1 ウェイト・ステート)

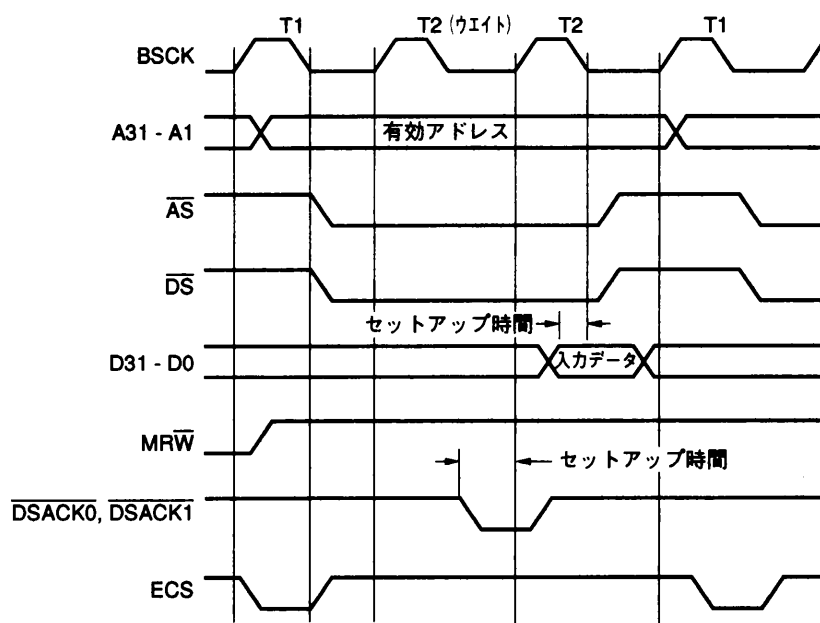
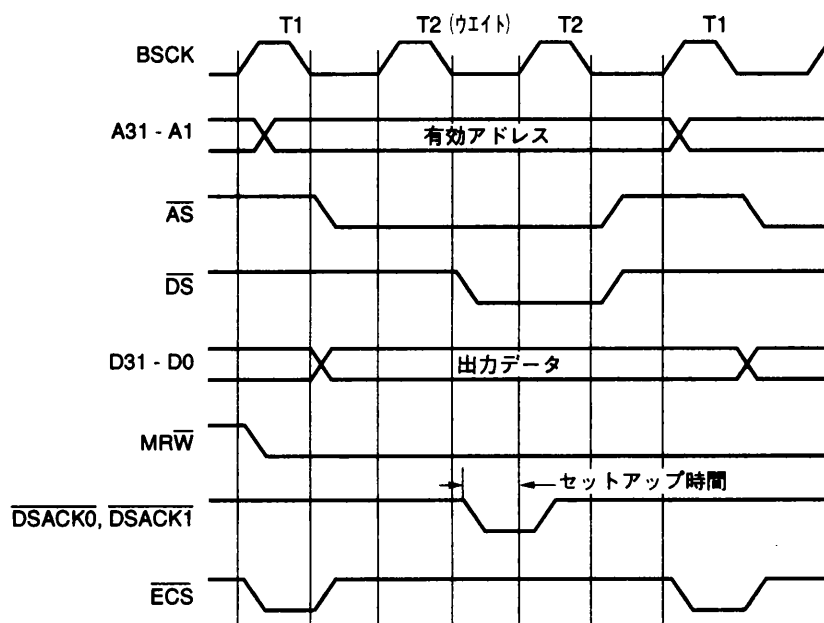


図 6-7 メモリ・ライト, BMODE = 1, 同期モード (1 ウェイト・ステート)



(3) BMODE=1 の場合のメモリ・サイクル、非同期モード

μPD72932はT1の立ち上がりエッジで $\overline{ECS}$ をロウ・レベルにして、メモリ・サイクルを開始したことを示し、アドレス(A31-A1)、バス・ステータス(S2-S0)、およびディレクション・ストロブ(MRW)をドライブし、残りのメモリ・サイクルの間これらを変更しません。μPD72932は、T1の立ち下がりエッジで $\overline{ECS}$ をハイ・レベルにし、 $\overline{AS}$ をロウ・レベルにします。

非同期モードでは、T1とT2の立ち下がりエッジで $\overline{DSACK0}$ 、 $\overline{DSACK1}$ を非同期でサンプルします。デバイスが常に信号のハイ状態、およびロウ状態を決めるので、 $\overline{DSACK0}$ 、 $\overline{DSACK1}$ をバス・クロックに同期させる必要はありません。ただし、同期をとってバス・サイクルを終了させる必要があるときは、 $\overline{STERM}$ を使用します。 $\overline{STERM}$ はT2の立ち上がりエッジでサンプルされ、正しい動作のためには、その立ち上がりエッジでセットアップおよびホールド時間条件を満たさなければなりません。 $\overline{DSACK0}$ 、 $\overline{DSACK1}$ や $\overline{STERM}$ のセットアップ時間条件が満たされていると、μPD72932は $\overline{DSACK0}$ 、 $\overline{DSACK1}$ をサンプルしてから1.5バス・クロック後に、または $\overline{STERM}$ をサンプルしてから1バス・クロック後にメモリ・サイクルを終了します。

$\overline{DSACK0}$ 、 $\overline{DSACK1}$ のあとのバス・クロックまたは $\overline{STERM}$ から1サイクル後のバス・クロックがサンプルされます。 $\overline{DSACK0}$ 、 $\overline{DSACK1}$ や $\overline{STERM}$ をロウ・レベルでサンプルするまでT2ステートを繰り返します。

リード・サイクル(図6-8、図6-9)では、T1の立ち下がりエッジで $\overline{DS}$ をロウ・レベルにし、T2の立ち下がりエッジでデータ(D31-D0)をラッチします。ライト・サイクル(図6-10、図6-11)では、T1の立ち下がりエッジでデータがドライブされます。挿入するウェイト・ステートがあると、最初のT2(ウェイト)の立ち下がりエッジで $\overline{DS}$ をロウ・レベルにします。0ウェイト・ステートのライト・サイクルでは、 $\overline{DS}$ をロウ・レベルにしません。μPD72932は、T2の立ち下がりエッジで $\overline{AS}$ と $\overline{DS}$ をハイ・レベルにして、メモリ・サイクルを終了します。

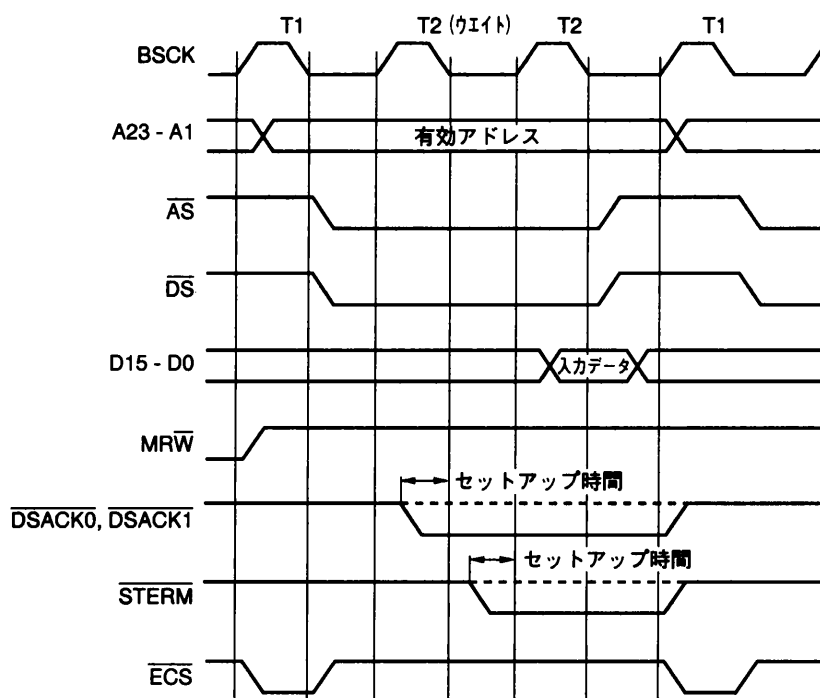
なお、 $\overline{DSACK1}$ 、 $\overline{STERM}$ のどちらか1つを使用することでサイクルを終了させることができます。たとえば、 $\overline{DSACK1}$ のみを使用する場合、 $\overline{STERM}$ は常時ハイ・レベルにプルアップしてください。

★

注意 T1で $\overline{DSACK0}$ 、 $\overline{DSACK1}$ のセットアップ時間条件を、または初めのT2で $\overline{STERM}$ のセットアップ時間条件を満たしていると、非同期の全バス・サイクルは2バス・クロックとすることができます。この状態が不要な場合は、T1の間に $\overline{DSACK0}$ 、 $\overline{DSACK1}$ 、T2の開始で $\overline{STERM}$ をハイ・レベルにしてください。

★

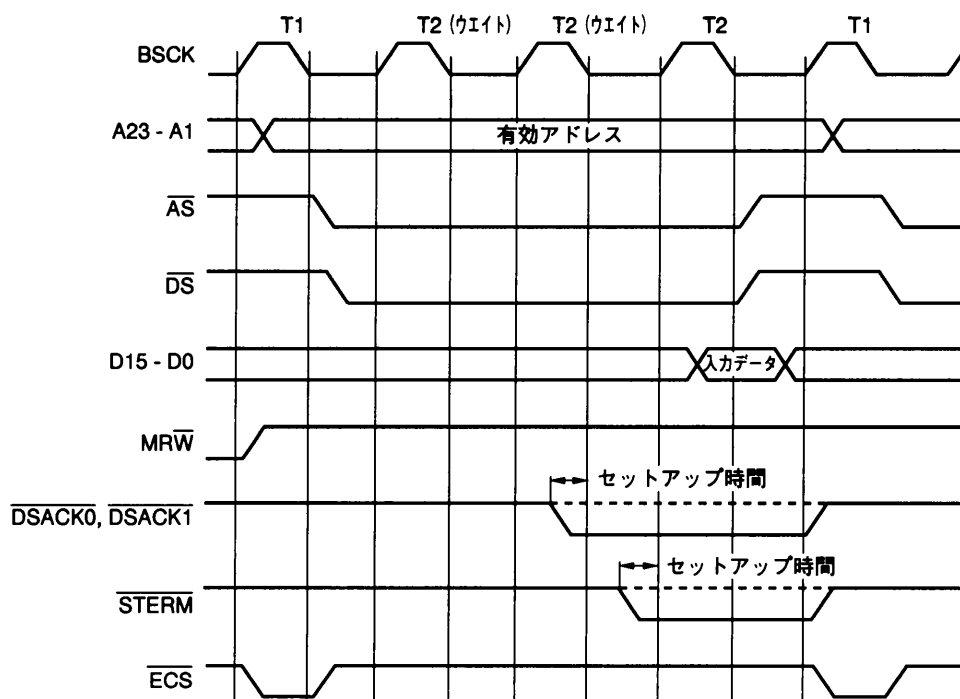
図 6-8 メモリ・リード, BMODE = 1, 非同期モード (1 ウェイト・ステート)



備考 破線は、未使用時のタイミングを示します。

★

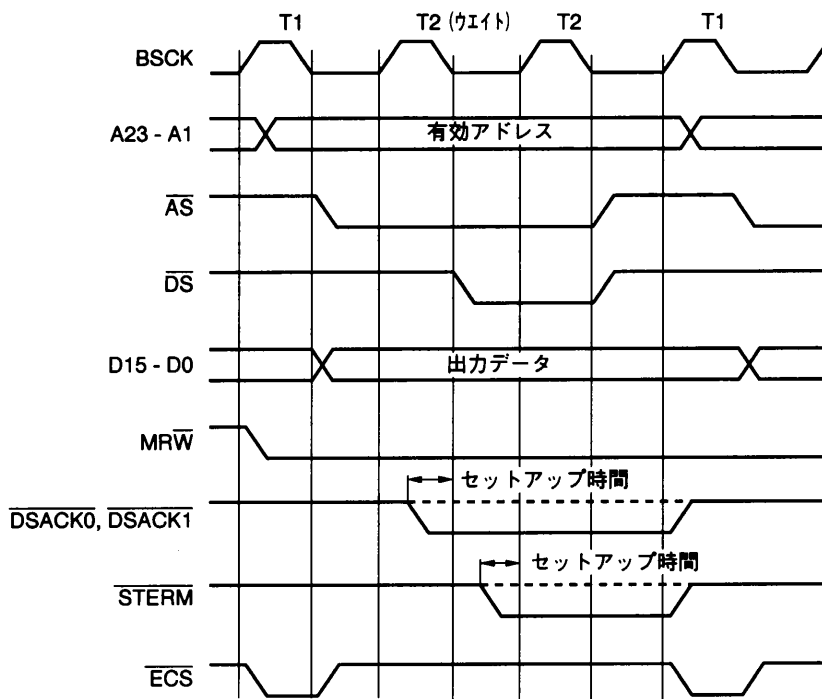
図 6-9 メモリ・リード, BMODE = 1, 非同期モード (2 ウェイト・ステート)



備考 破線は、未使用時のタイミングを示します。

図 6-10 メモリ・ライト, BMODE = 1, 非同期モード (1 ウェイト・ステート)

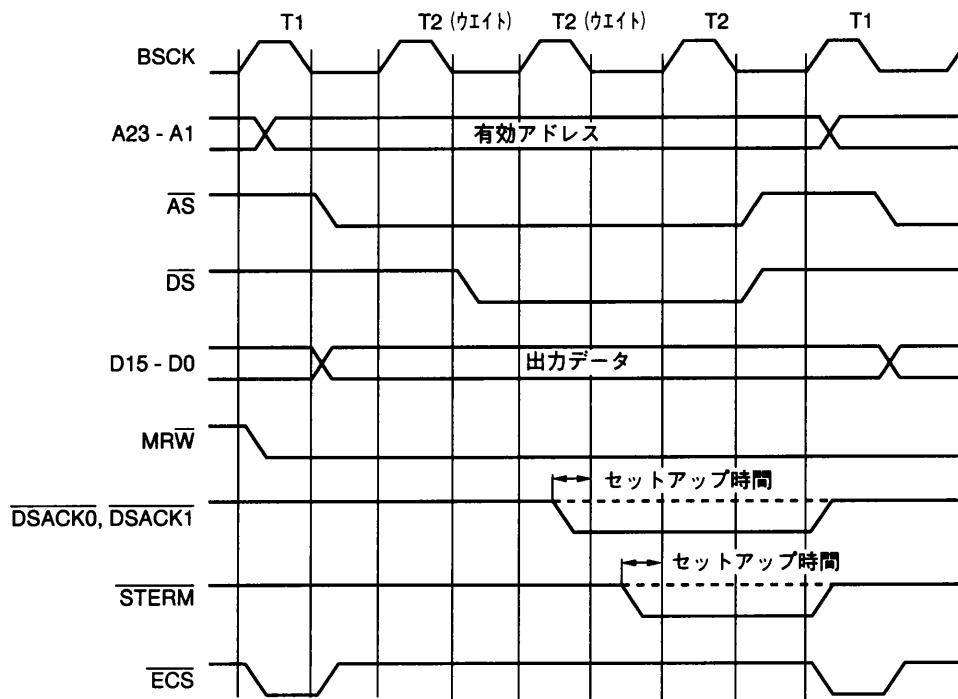
★



備考 破線は、未使用時のタイミングを示します。

図 6-11 メモリ・ライト, BMODE = 1, 非同期モード (2 ウェイト・ステート)

★



備考 破線は、未使用時のタイミングを示します。

(4) BMODE=0の場合のメモリ・サイクル，同期モード

μPD72932はT1の立ち上がりエッジで $\overline{\text{ADS}}$ と $\overline{\text{ECS}}$ をロウ・レベルにしメモリ・サイクルを開始したことを示します。アドレス(A31-A1)，バス・ステータス(S2-S0)，およびディレクション・ストローブ(MWR)をドライブし，残りのメモリ・サイクルの間これらのラインを変更しません。μPD72932は，T1の立ち下がりエッジで $\overline{\text{ECS}}$ をハイ・レベルにし，T2の立ち上がりエッジで $\overline{\text{ADS}}$ をハイ・レベルにします。

同期モードでは，T2の終わりの立ち上がりエッジ(次のT1の立ち上がりエッジ)で $\overline{\text{RDYi}}$ をサンプルします。 $\overline{\text{RDYi}}$ をロウ・レベルでサンプルするまでT2ステートを繰り返します。正しい動作のためには， $\overline{\text{RDYi}}$ はバス・クロックの立ち上がりエッジでセットアップおよびホールド時間条件を満たさなければなりません。

リード・サイクル(図6-12)では，T2の終わりの立ち上がりエッジでデータ(D31-D0)をラッチします。ライト・サイクル(図6-13)では，T1の立ち下がりエッジでデータをドライブし，サイクル終了まで続きます。

図 6-12 メモリ・リード, BMODE = 0, 同期モード (1 ウェイト・ステート)

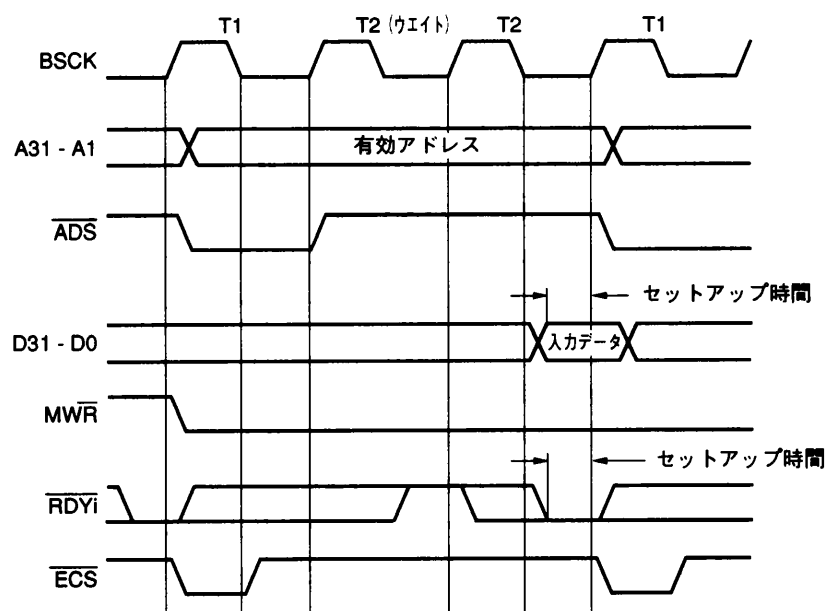
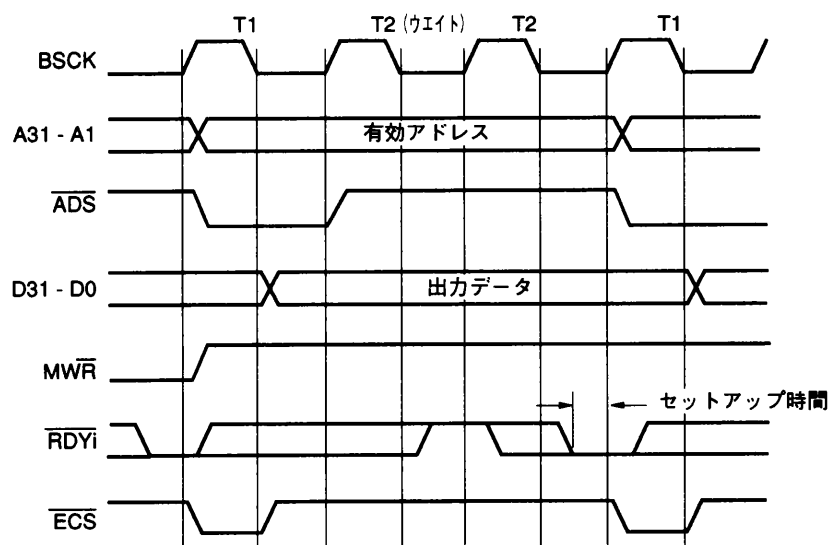


図 6-13 メモリ・ライト, BMODE = 0, 同期モード (1 ウェイト・ステート)



(5) BMODE=0の場合のメモリ・サイクル、非同期モード

μ PD72932は T1 の立ち上がりエッジで $\overline{\text{ADS}}$ と $\overline{\text{ECS}}$ をロウ・レベルにして、メモリ・サイクルを開始したことを示します。アドレス (A31-A1)、バス・ステータス (S2-S0)、およびディレクション・ストロブ (MWR) をドライブし、残りのメモリ・サイクルの間これらを変更しません。 μ PD72932は、T1 の立ち下がりエッジで $\overline{\text{ECS}}$ をハイ・レベルにし、T2 の立ち上がりエッジで $\overline{\text{ADS}}$ をハイ・レベルにします。

非同期モードでは、T1 と T2 の立ち下がりエッジで $\overline{\text{RDYi}}$ を非同期でサンプルします。デバイスが常に信号のハイ状態、およびロウ状態を決めるので、 $\overline{\text{RDYi}}$ をバス・クロックに同期させる必要はありません。

$\overline{\text{RDYi}}$ のセットアップ時間条件を満たしていると、 μ PD72932は $\overline{\text{RDYi}}$ がサンプルされてから 1.5 バス・クロック後にメモリ・サイクルを終了することを保証します。 $\overline{\text{RDYi}}$ をロウ・レベルでサンプルするまで T2 ステートを繰り返します^注。

リード・サイクル (図 6-14, 図 6-15) では、T2 の立ち下がりエッジでデータ (D31-D0) をラッチし、T1 の立ち下がりエッジで $\overline{\text{DS}}$ をロウ・レベルにします。ライト・サイクル (図 6-16, 図 6-17) では、T1 の立ち下がりエッジでデータをドライブします。挿入するウェイト・ステートがあると、最初の T2 (ウェイト) の立ち下がりエッジで $\overline{\text{DS}}$ をロウ・レベルにします。0 ウェイト・ステートのライト・サイクルでは、 $\overline{\text{DS}}$ をロウ・レベルにしません。 μ PD72932は、T2 の立ち下がりエッジで $\overline{\text{DS}}$ をハイ・レベルにして、メモリ・サイクルを終了します。

注 T1 で $\overline{\text{RDYi}}$ のセットアップ時間条件を満たしていると、非同期の全バス・サイクルは 2 バス・クロックとすることができます。この状態が不要な場合は、T1 で $\overline{\text{RDYi}}$ をハイ・レベルにしてください。

図 6-14 メモリ・リード, BMODE = 0, 非同同期モード (1 ウェイト・ステート)

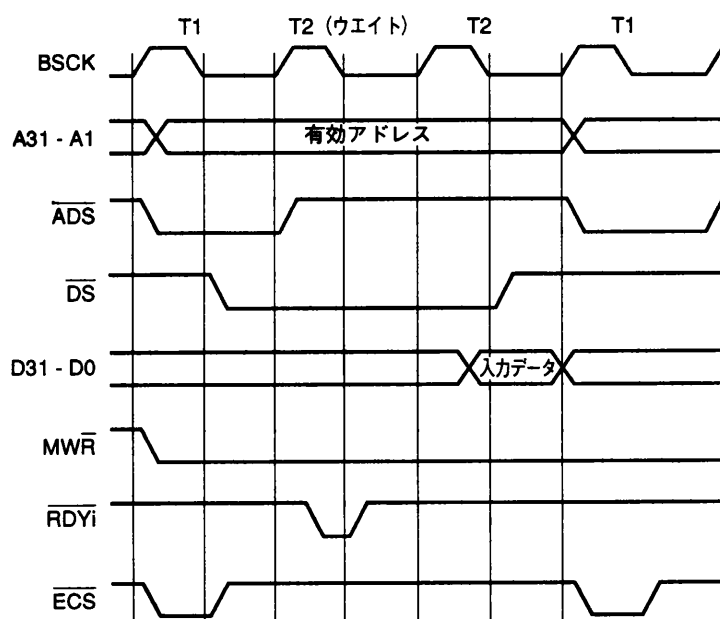


図 6-15 メモリ・リード, BMODE = 0, 非同同期モード (2 ウェイト・ステート)

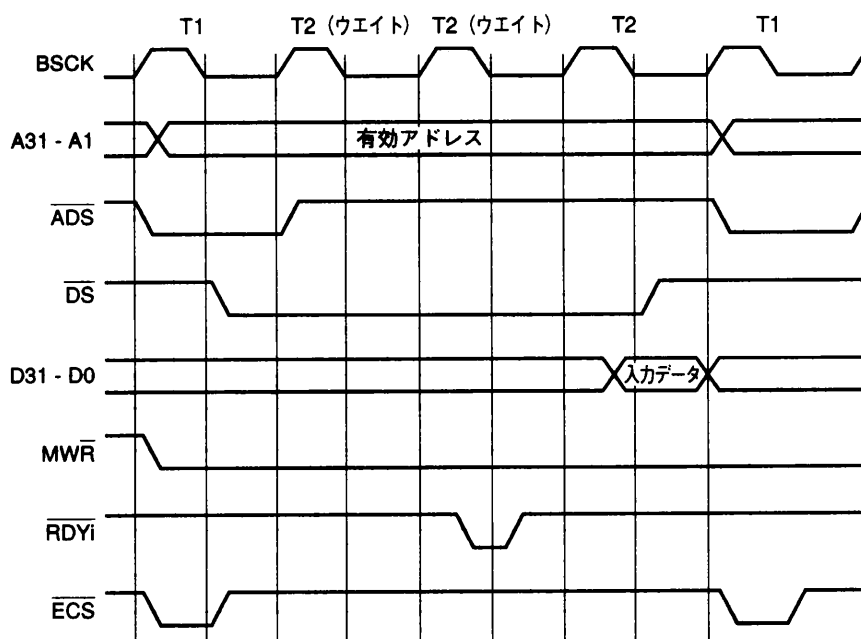


図 6-16 メモリ・ライト, BMODE = 0, 非同期モード (1 ウェイト・ステート)

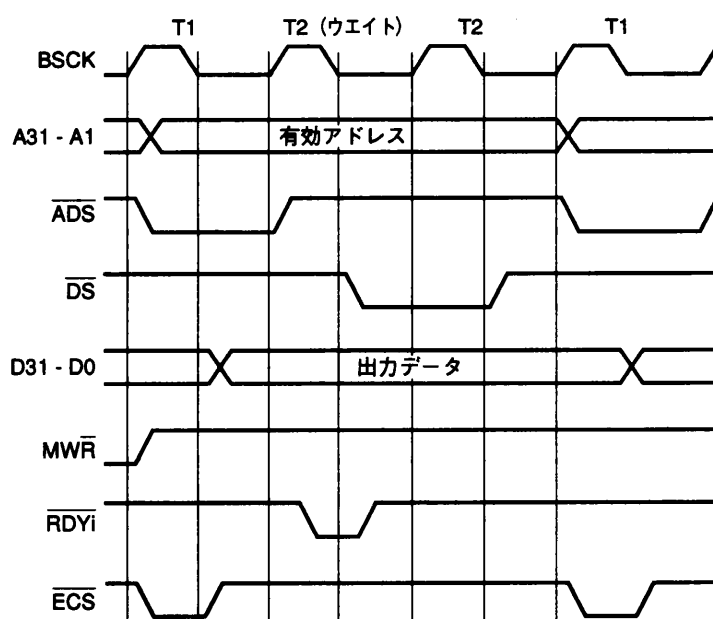
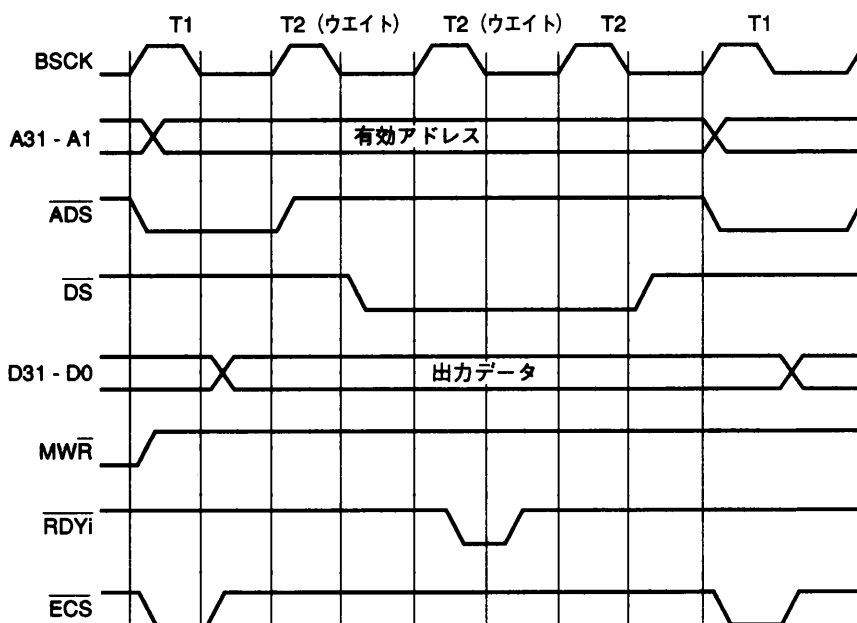


図 6-17 メモリ・ライト, BMODE = 0, 非同期モード (2 ウェイト・ステート)



6.3.6 バスの特殊実行例（バス・リトライ）

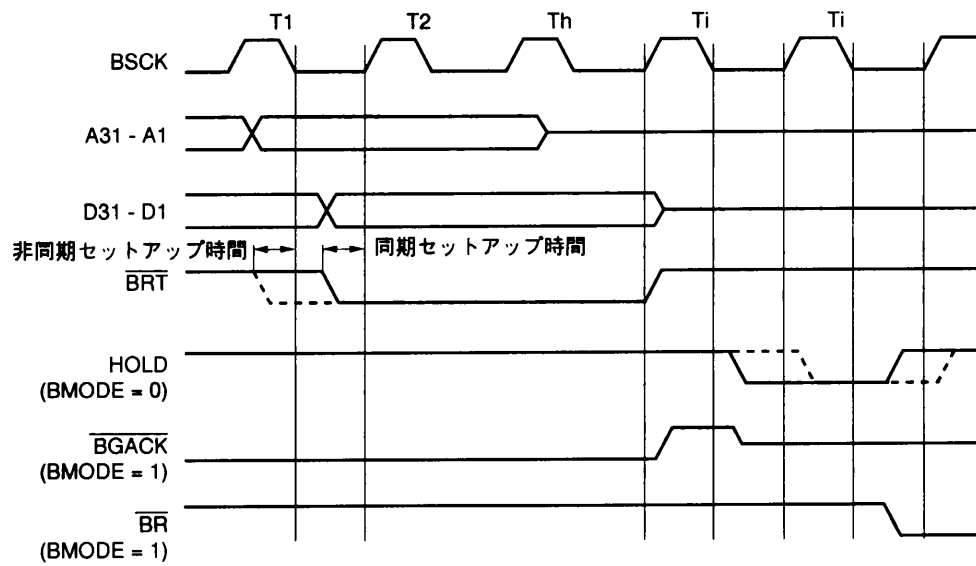
μPD72932はバス・サイクル実行中に、バス・エラーのリトライ処理をする機能を備えています（図 6-18）。

システムは、 $\overline{\text{BRT}}$ （バス・リトライ）をロウ・レベルにして、強制的に μPD72932にカレント・メモリ・サイクルを反復させることができます。 $\overline{\text{BRT}}$ をサンプルしてロウ・レベルであると、μPD72932はT2の終わりでメモリ・サイクルを完了し、 $\overline{\text{BGACK}}$ または $\overline{\text{HOLD}}$ をインアクティブにしてバスを解放します。次に、ラッチト・バス・リトライ・モードがセットされていない場合（5.3.2 データ・コンフィギュレーション・レジスタのLBR）、再度バス要求を行い同じメモリ・サイクルをリトライします。ただし、ラッチト・バス・リトライ・モードがセットされている場合、ISRのBRビット（5.3.6参照）がリセットされ、 $\overline{\text{BRT}}$ がハイ・レベルになるまで、μPD72932はバス・リトライを要求しません。 $\overline{\text{BRT}}$ は、 $\overline{\text{DSACK0}}$ 、 $\overline{\text{DSACK1}}$ 、 $\overline{\text{STERM}}$ または $\overline{\text{RDYi}}$ よりメモリ・サイクル完了の優先権が高くなっています。

DCRのEXBUSビット（5.3.2）をセットすることで、 $\overline{\text{BRT}}$ を同期、または非同期でサンプルすることができます。同期のバス・リトライをセットすると、 $\overline{\text{BRT}}$ はT2の立ち上がりエッジでサンプルされます。非同期のバス・リトライをセットすると、 $\overline{\text{BRT}}$ はT1の立ち下がりエッジで同期化されます。非同期のセットアップ時間を守る必要はありませんが、この条件を満たすことにより、バスの特殊実行を次のバス・サイクルではなく、カレント・バス・サイクルで行うことができます。μPD72932が非同期モードの場合に限り、非同期バス・リトライが使用できます。

- 注意 1. $\overline{\text{HOLD}}$ の立ち下がりエッジのタイミングはDCR2のPHビット（5.3.7参照）で決まります。また、 $\overline{\text{BGACK}}$ もハイ・インピーダンスになる前に約1/2バス・クロックの間ハイ・レベルになります。
2. ラッチト・バス・リトライをセットする場合、 $\overline{\text{BRT}}$ はセットアップ時間条件を満たす必要がありますが、ホールド時間は重要ではありません。また、ノンラッチ・バス・リトライの場合、 $\overline{\text{BRT}}$ はThステートのあとまでハイ・レベルにしなければなりません。
3. $\overline{\text{BRT}}$ のあと、 $\overline{\text{DSACK0}}$ 、 $\overline{\text{DSACK1}}$ 、 $\overline{\text{STERM}}$ または $\overline{\text{RDYi}}$ がロウ・レベルのままであると、次のメモリ・サイクルに影響を与えることがあります。

図 6-18 バスの特殊実行例 (バス・リトライ)



6.3.7 スレーブ・モードのバス・サイクル

CPU は、2つの方法 (BMODE=1 または BMODE=0) のどちらかで、μPD72932の内部レジスタにアクセスします。どちらの方法でも、μPD72932はバス上でスレーブとしてはたります。このセクションでは、μPD72932のスレーブ・モードでのバス・オペレーションを説明します。

(1) BMODE=1 のスレーブ・サイクル

BMODE=1 のとき、システムは $\overline{SAS}$ 、 $\overline{SRW}$ および RA5-RA0 を操作することによって μPD72932 にアクセスします。これらの信号はバス・サイクルごとにサンプルされます。 $\overline{CS}$ をサンプルするクロックの次のクロックの立ち下がりエッジより前に $\overline{SAS}$ をロウ・レベルにして、データのリード/ライト制御、アドレス指定を行ってください。 $\overline{SAS}$ と $\overline{CS}$ をロウ・レベルにしたあと 1-2 バス・クロック後に $\overline{SMACK}$ をロウ・レベルにし、μPD72932 がスレーブ・サイクルを開始したことを示します。 $\overline{CS}$ は非同期入力ですが、セットアップ時間を合わせることで、 $\overline{CS}$ を同期して入力したバス・クロックの立ち下がりエッジの、1 バス・クロック後に $\overline{SMACK}$ をロウ・レベルにします (図 6-19, 図 6-20 参照)。これは、 $\overline{CS}$ がロウ・レベルになったときに μPD72932 がバス・マスタでないことを前提としています。μPD72932 がバス・マスタの場合、 $\overline{CS}$ がロウ・レベルになると、μPD72932 はカレント・マスタ・バス・サイクルを終了して一時的にバスを解放します (6.3.8 参照)。この場合 $\overline{SMACK}$ は、 $\overline{CS}$ を同期して入力したバス・クロックの立ち下がりエッジの、5 バス・クロック後にロウ・レベルになります。これは、カレント・マスタ・モードのアクセスにウエイト・ステートがないことを前提としています。サイクル中にウエイト・ステートがあると、その数の分だけ $\overline{SMACK}$ がロウ・レベルになる時間が遅くなります。

スレーブ・サイクルがリード・サイクルであれば、データは $\overline{SMACK}$ と同じエッジでセットされます (図 6-19)。スレーブ・アクセスがライト・サイクルであれば、 $\overline{SMACK}$ がロウ・レベルになってから正確に 2 バス・クロック後にデータがラッチされます (図 6-20)。どちらの場合においても、 $\overline{DSACK0}$ 、 $\overline{DSACK1}$ は $\overline{SMACK}$ の 2 バス・クロック後にロウ・レベルになり、スレーブ・サイクルを終了します。

リード・サイクルの場合、レジスタ・データが有効になると $\overline{DSACK0}$ 、 $\overline{DSACK1}$ がロウ・レベルになり、ライト・サイクルの場合、μPD72932 がデータをラッチすると $\overline{DSACK0}$ 、 $\overline{DSACK1}$ がロウ・レベルになります。リード・サイクルの場合、 $\overline{SAS}$ または $\overline{CS}$ のうち、どちらか先にハイ・レベルになったほうの立ち上がりエッジで、μPD72932 は $\overline{DSACK0}$ 、 $\overline{DSACK1}$ 、 $\overline{SMACK}$ およびデータをインアクティブにします。

注意 1. μPD72932 は $\overline{DSACK0}$ 、 $\overline{DSACK1}$ をロウ・レベルにするときは 32 ビットのペリフェラルとして応答しますが、D15-DO 上でのみデータを転送します。

2. 複数レジスタ・アクセスの場合、 $\overline{CS}$ をロウ・レベルに保持し、 $\overline{SAS}$ を用いて、スレーブ・サイクルの境界を定めることができます ($\overline{SAS}$ の前に $\overline{CS}$ をロウ・レベルにできる唯一のケースです)。この場合、 $\overline{CS}$ がすでにロウ・レベルになっているため、 $\overline{SAS}$ がロウ・レベルになることによって $\overline{SMACK}$ がロウ・レベルになります。 $\overline{CS}$ がロウ・レベルであっても $\overline{SMACK}$ がロウ・レベルになっていない場合があることに注意してください ($\overline{SMACK}$ がずっとロウ・レベルを保持するのは $\overline{MREQ}$ による場合。6.3.8 参照)。
3. チップ・セレクト ($\overline{CS}$) のあとにメモリ要求 ($\overline{MREQ}$) が続く場合、 $\overline{CS}$ をハイ・レベルにしたあと少なくとも 2 バス・クロック経過してから $\overline{MREQ}$ をロウ・レベルにしてください。 $\overline{CS}$ と $\overline{MREQ}$ を同時にロウ・レベルにしないでください。
4. $\overline{CS}$ をインアクティブにする場合、少なくとも 1 バス・クロックの間ハイ・レベルに保持してください。
5. $\overline{SMACK}$ がロウ・レベルになる動作は $\overline{CS}$ による場合と $\overline{MREQ}$ による場合とで異なります。 $\overline{SMACK}$ は、 $\overline{CS}$ だけでなく $\overline{CS}$ と $\overline{SAS}$ の両方がロウ・レベルになることによってロウ・レベルになります。 $\overline{MREQ}$ による場合については 6.3.8 を参照してください。 $\overline{CS}$ による場合と $\overline{MREQ}$ による場合とを混同しないよう注意してください。

図 6-19 レジスタ・リード, BMODE = 1

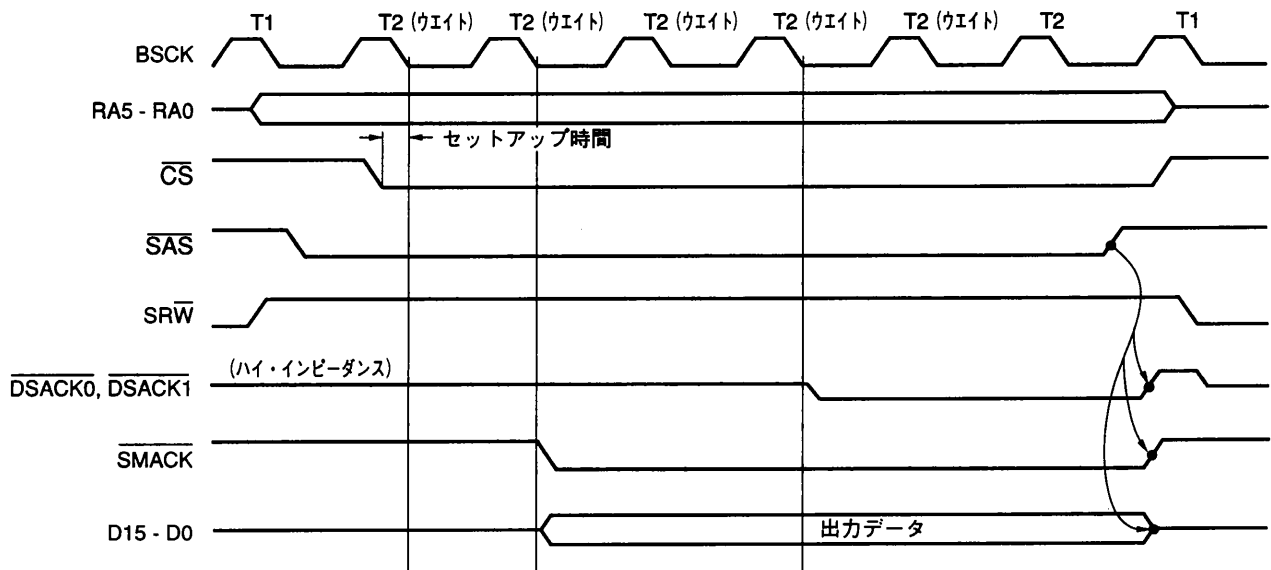
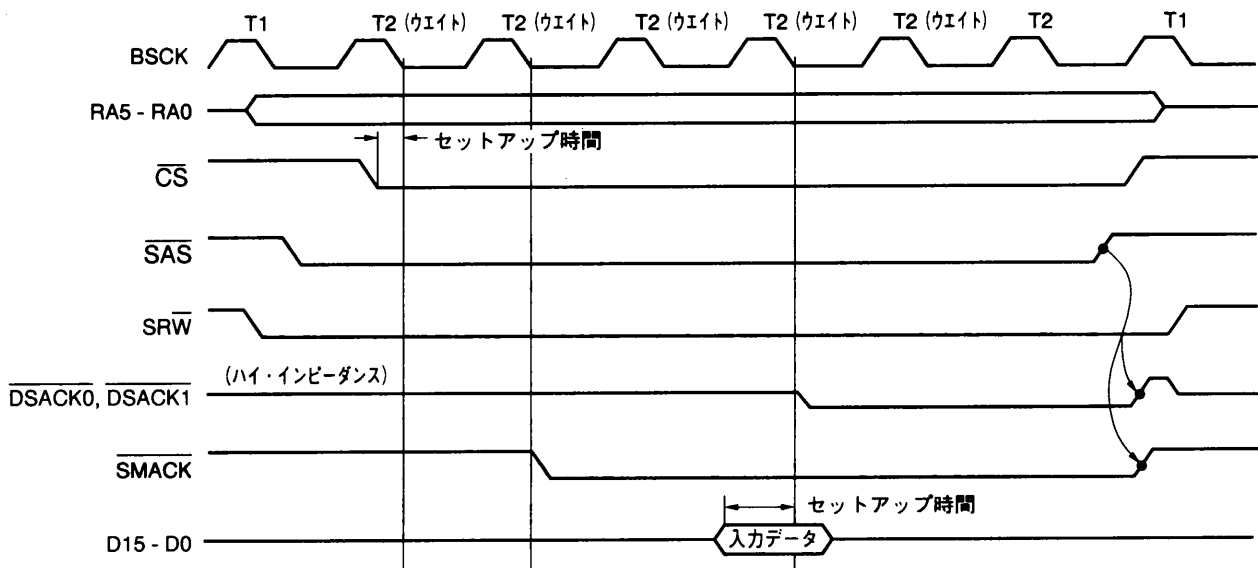


図 6-20 レジスタ・ライト, BMODE = 1



(2) BMODE=0のスレーブ・サイクル

BMODE=0のとき、システム(CPU)は $\overline{SAS}$ 、 $\overline{CS}$ 、 $\overline{SWR}$ およびRA5-RA0をセットすることによって μ PD72932にアクセスします。これらの信号はバス・サイクルごとにサンプルされますが、 $\overline{CS}$ がロウ・レベル、 $\overline{SAS}$ がハイ・レベルになっていないと、 μ PD72932は実際にはスレーブ・サイクルを開始しません。 $\overline{SAS}$ をロウ・レベルにする前に $\overline{CS}$ をロウ・レベルにすると、誤ったスレーブ動作を行ってしまいます。しかし、 $\overline{SAS}$ の立ち上がりエッジの前に $\overline{CS}$ をロウ・レベルにすることができます。この場合、 $\overline{CS}$ の立ち下がりエッジのあとの1バス・クロック以内に $\overline{SAS}$ をハイ・レベルにしてください。 $\overline{SAS}$ をハイ・レベルにして、 $\overline{CS}$ をロウ・レベルにしたあと1-2バス・クロック後に $\overline{SMACK}$ をロウ・レベルにし、 μ PD72932がスレーブ・サイクルを開始したことを示します。 $\overline{CS}$ は非同期入力ですが、セットアップ時間を合わせることによって、 $\overline{CS}$ を同期して入力したバス・クロックの立ち下がりエッジの、1バス・クロック後に $\overline{SMACK}$ をロウ・レベルにします(図6-21、図6-22参照)。これは、 $\overline{CS}$ がロウ・レベルになったときに μ PD72932がバス・マスタでないことを前提としています。 μ PD72932がバス・マスタの場合、 $\overline{CS}$ がロウ・レベルになると、 μ PD72932はカレント・マスタ・バス・サイクルを終了して一時的にバスを解放します(6.3.8参照)。この場合 $\overline{SMACK}$ は、 $\overline{CS}$ を同期して入力したバス・クロックの立ち下がりエッジの、5バス・クロック後にロウ・レベルになります。これは、カレント・マスタ・モードのアクセスにウェイト・ステートがないことを前提としています。サイクル中にウェイト・ステートがあると、その数の分だけ $\overline{SMACK}$ がロウ・レベルになる時間が遅くなります。

スレーブ・サイクルがリード・サイクルであれば、データは $\overline{SMACK}$ と同じエッジでセットされます(図6-21)。スレーブ・アクセスがライト・サイクルであれば、 $\overline{SMACK}$ がロウ・レベルになってから正確に2バス・クロック後にデータがラッチされます(図6-22)。どちらの場合においても、 $\overline{RDY}_O$ は $\overline{SMACK}$ の2.5バス・クロック後にロウ・レベルになり、スレーブ・サイクルを終了します。

リード・サイクルの場合、レジスタ・データが有効になると $\overline{RDY}_O$ がロウ・レベルになり、ライト・サイクルの場合、 μ PD72932がデータをラッチすると $\overline{RDY}_O$ がロウ・レベルになります。リード・サイクルの場合、 $\overline{SAS}$ の立ち下がりエッジまたは $\overline{CS}$ の立ち上がりエッジのうち、どちらか先になったほうで、 μ PD72932は $\overline{RDY}_O$ 、 $\overline{SMACK}$ およびデータをインアクティブにします。

注意 1. μ PD72932はスレーブ・モード・アクセスの間D15-D0上でのみデータを転送します。

2. 複数レジスタ・アクセスの場合、 $\overline{CS}$ をロウ・レベルに保持し、 $\overline{SAS}$ を用いて、スレーブ・サイクルの境界を定めることができます($\overline{SAS}$ の前に $\overline{CS}$ をロウ・レベルにできる唯一のケースです)。この場合、 $\overline{CS}$ がすでにロウ・レベルになっているため、 $\overline{SAS}$ がハイ・レベルになることによって $\overline{SMACK}$ がロウ・レベルになります。 $\overline{CS}$ がロウ・レベルであっても $\overline{SMACK}$ がロウ・レベルになっていない場合があることに注意してください($\overline{SMACK}$ がずっとロウ・レベルを保持するのは $\overline{MREQ}$ による場合。6.3.8参照)。
3. チップ・セレクト($\overline{CS}$)のあとにメモリ要求($\overline{MREQ}$)が続く場合、 $\overline{CS}$ をハイ・レベルにしたあと少なくとも2バス・クロック経過してから $\overline{MREQ}$ をロウ・レベルにしてください。 $\overline{CS}$ と $\overline{MREQ}$ を同時にロウ・レベルにしないでください。
4. $\overline{CS}$ をインアクティブにする場合、少なくとも1バス・クロックの間ハイ・レベルに保持してください。
5. $\overline{SMACK}$ がロウ・レベルになる動作は $\overline{CS}$ による場合と $\overline{MREQ}$ による場合とで異なります。 $\overline{SMACK}$ は、 $\overline{CS}$ だけでなく $\overline{CS}$ と $\overline{SAS}$ の両方がロウ・レベルになることによってロウ・レベルになります。 $\overline{MREQ}$ による場合については6.3.8を参照してください。 $\overline{CS}$ による場合と $\overline{MREQ}$ による場合とを混同しないよう注意してください。

図 6-21 レジスタ・リード, BMODE = 0

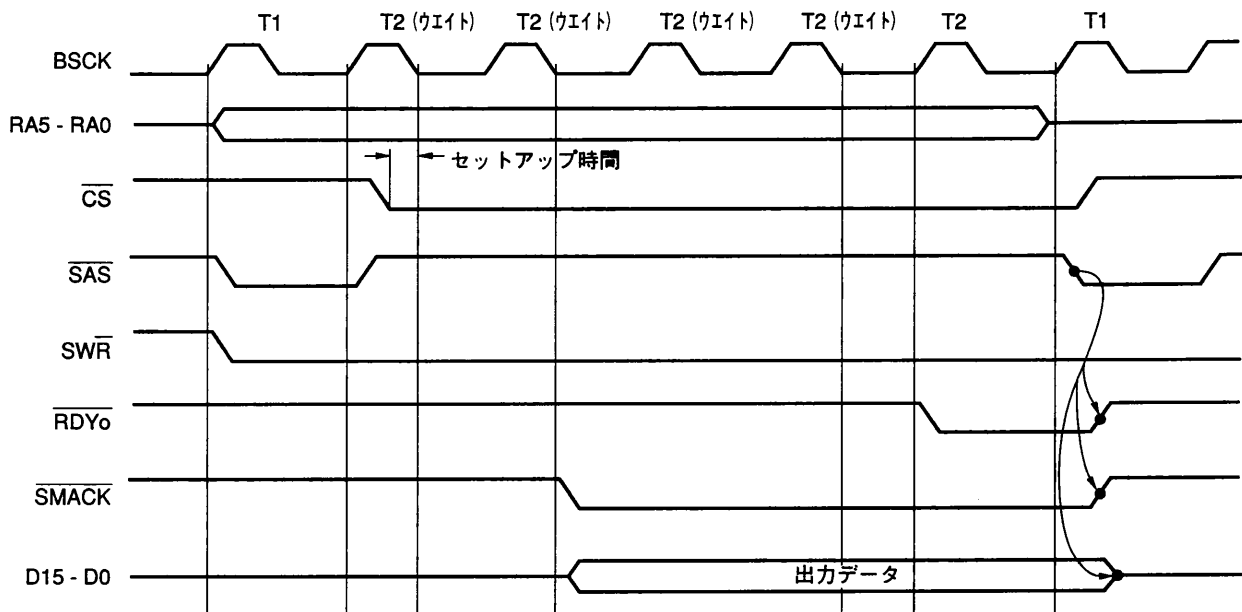
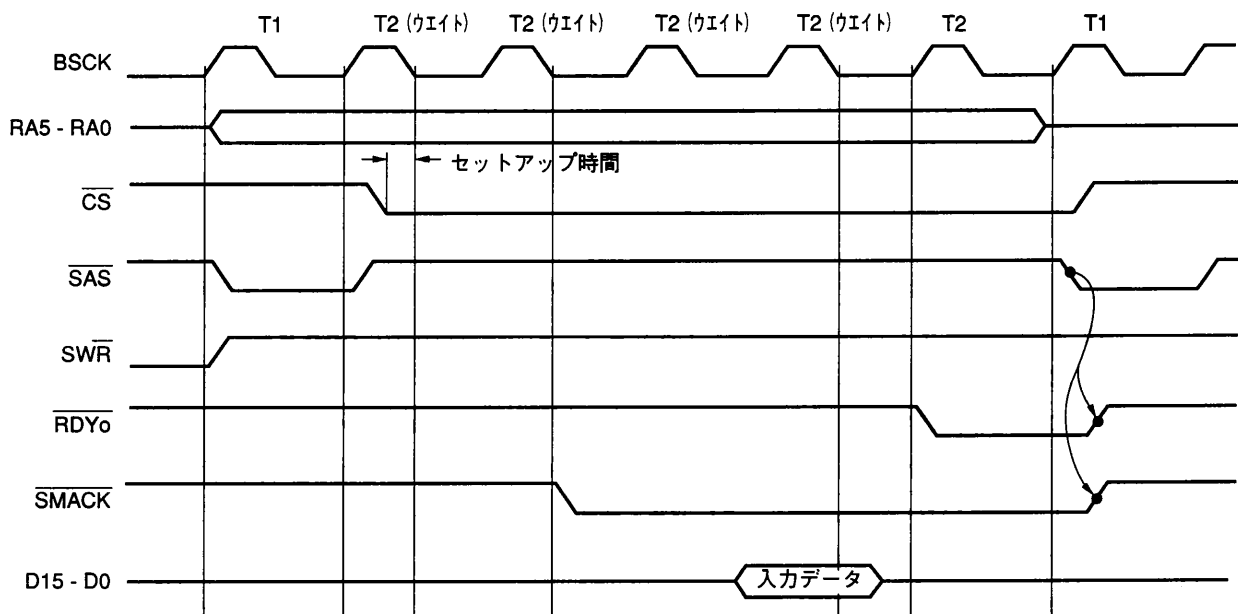


図 6-22 レジスタ・ライト, BMODE = 0



6.3.8 オンチップ・メモリ・アービタ

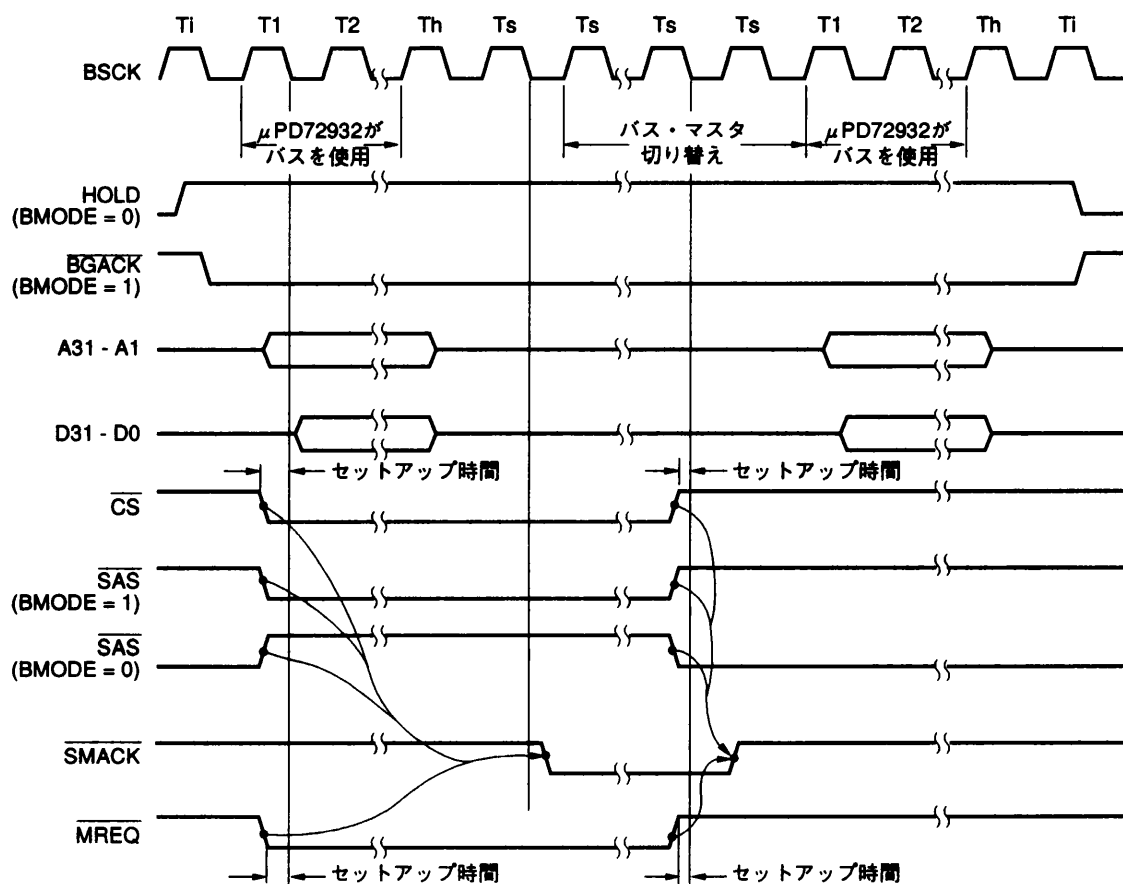
バッファ・エリアをホスト・システムと共用する共用型メモリ・アプリケーションに対し、 μ PD72932は μ PD72932とホスト・システム間のアクセスを効率的に実行するための、高速オンチップ・メモリ・アービタを備えています(図 6-23)。ホスト・システムは、メモリ要求 ($\overline{\text{MREQ}}$) をロウ・レベルにして共用型メモリの使用を要求します。 μ PD72932はホスト・システムの要求に対し、メモリ/スレーブ・アクノリッジ ($\overline{\text{SMACK}}$) で応答し、ホスト・システムが共用型メモリを使用することを許可します。 $\overline{\text{SMACK}}$ がロウ・レベルになると、ホスト・システムは自由に共用型メモリを使用できます。ホスト・システムは共用型メモリを使用したあと、 $\overline{\text{MREQ}}$ をハイ・レベルに戻して共用型メモリを放棄します。

$\overline{\text{MREQ}}$ はバス・クロックの立ち下がりエッジでサンプリングされ、その立ち上がりエッジによって内部で二重に同期化します。 $\overline{\text{SMACK}}$ は T_s バス・サイクルの立ち下がりエッジでロウ・レベルになります。 μ PD72932がメモリにアクセスしていない場合は、 $\overline{\text{MREQ}}$ のクロック入力後ただちに $\overline{\text{SMACK}}$ をロウ・レベルにします。しかし、 μ PD72932がメモリにアクセスしている場合、カレントのメモリ転送を終了したあと、 $\overline{\text{SMACK}}$ をロウ・レベルにします。すなわち、 μ PD72932は、それぞれ $\overline{\text{MREQ}}$ クロック入力の 1 または 5 (注意 2 参照) バス・クロック後に $\overline{\text{SMACK}}$ をロウ・レベルにします。 $\overline{\text{MREQ}}$ を二重に同期化するため、セットアップ時間を規格に合わせる必要はありませんが、セットアップ時間を合わせると、 $\overline{\text{SMACK}}$ がカレント・バス・クロックの次のバス・クロック、あるいは 5 番目のバス・クロック後にロウ・レベルになります。 $\overline{\text{SMACK}}$ は $\overline{\text{MREQ}}$ がハイ・レベルになってから 1 バス・クロック後にハイ・レベルになります。また、 μ PD72932が以前からバスを使用していた場合、 μ PD72932はそのマスタ動作を終了します。

ホスト・システムが共用型メモリの代わりに μ PD72932のレジスタにアクセスする場合、 $\overline{\text{MREQ}}$ の代わりに $\overline{\text{CS}}$ をロウ・レベルにします。 μ PD72932レジスタへのアクセス方法は、 μ PD72932をアイドル状態にする代わりにスレーブ・サイクルに入ることを除いて、ほとんど共用型メモリにアクセスする場合と同じです。レジスタへのアクセス方法についての詳細は、6.3.7 を参照してください。

- 注意 1. $\overline{\text{CS}}$ と $\overline{\text{MREQ}}$ を連続してロウ・レベルにする場合、あとからロウ・レベルにする信号は、先にロウ・レベルにした信号の立ち上がりエッジから 2 バス・クロック以上の間ハイ・レベルに保持したうえでロウ・レベルにしてください。 $\overline{\text{CS}}$ と $\overline{\text{MREQ}}$ を同時にロウ・レベルにしないでください。
2. μ PD72932がマスタ・モード時に $\overline{\text{MREQ}}$ がロウ・レベルになってから $\overline{\text{SMACK}}$ がロウ・レベルになるまでのバス・クロック数は、マスタ・モードのアクセスにウエイト・ステートがない場合、最大 5 バス・クロックになります。サイクル中にウエイト・ステートがあると、その数の分だけ $\overline{\text{SMACK}}$ がロウ・レベルになる時間が遅くなります(バス・クロック数は 5+ウエイト・ステート数になります)。
3. $\overline{\text{SMACK}}$ がロウ・レベルになる動作は $\overline{\text{CS}}$ による場合と $\overline{\text{MREQ}}$ による場合とで異なります。 $\overline{\text{MREQ}}$ をロウ・レベルにした場合はその直接の結果として $\overline{\text{SMACK}}$ がロウ・レベルになりますが、 $\overline{\text{CS}}$ の場合は $\overline{\text{SMACK}}$ がロウ・レベルになるより前に $\overline{\text{SAS}}$ もロウ・レベル ($\text{BMODE}=1$ の場合) またはハイ・レベル ($\text{BMODE}=0$ の場合) にしておかなければなりません。つまり、 $\overline{\text{MREQ}}$ によって $\overline{\text{SMACK}}$ がロウ・レベルになった場合は、 $\overline{\text{MREQ}}$ がハイ・レベルになるまでは $\overline{\text{SMACK}}$ はロウ・レベルを保持します。 $\overline{\text{SMACK}}$ をハイ・レベルにしなくても、共用メモリへの複数のメモリ・アクセスが可能です。しかし $\overline{\text{CS}}$ によって $\overline{\text{SMACK}}$ がロウ・レベルになった場合は、 $\overline{\text{SMACK}}$ がロウ・レベルを保持するのは $\overline{\text{SAS}}$ がロウ・レベル ($\text{BMODE}=1$ の場合) またはハイ・レベル ($\text{BMODE}=0$ の場合) である間に限られます。 $\overline{\text{SAS}}$ はレジスタ・アクセスを行うたびに切り替えなければならないので、 μ PD72932に複数のレジスタ・アクセスを行っている間ずっと $\overline{\text{SMACK}}$ をロウ・レベルに保持しておくことはできません。この点が共有メモリ・デザインを設計する場合に考慮すべき重要な違いです。
4. $\text{BMODE}=1$ (モトローラ・モード) では、バス・マスタが $\overline{\text{MREQ}}$ をロウ・レベルにして μ PD72932にバス解放を要求した場合、 μ PD72932はバスを解放したあとも $\overline{\text{DSACK1}}$ 、 $\overline{\text{DSACK0}}$ をドライブし続けます。このため、3 ステート・バッファを使用するなどして $\overline{\text{DSACK1}}$ 、 $\overline{\text{DSACK0}}$ をバスから切り離す必要があります。

図 6-23 オンチップ・メモリ・アービタ



6.3.9 チップ・リセット

μPD72932には、ハードウェア・リセットとソフトウェア・リセットの2つのリセット・モードがあります。μPD72932は、**RESET** 端子をロウ・レベルにしてハードウェア・リセットするか、あるいはコマンド・レジスタの RST ビットをセット (5.3.1) して、ソフトウェア・リセットすることができます。機能が異なるため、2つのモードには互換性はありません。

電源投入後は、動作状態になるまで、μPD72932をハードウェア・リセットしなければなりません。μPD72932のハードウェア・リセットは、最低10送信クロック時間 (10イーサネット送信クロック周期) **RESET** 端子をロウ・レベルに行います。バス・クロック (BSCCK) 周期が送信クロック周期より大きい場合、10送信クロックの代わりに10バス・クロックを使用します。ハードウェア・リセットによって μPD72932は次に示す状態になります。影響を受けるレジスタを () 内に示します。ハードウェア・リセットによって影響を受けるレジスタについての詳細は、表 6-3 と 5.3 を参照してください。

- (1) レシーバおよびトランスミッタをディスエーブルします (CR)。
- (2) 汎用タイマを停止します (CR)。
- (3) すべての割り込みについてマスク (割り込み禁止) します (IMR)。
- (4) 送信制御レジスタ (TCR) の NCRS および PTX ステータス・ビットをセットします。
- (5) エンド・オブ・バイト・カウント (EOBC) レジスタを 02F8H (760ワード) にセットします。
- (6) パケットおよび RBA シーケンス番号カウンタを 0 にセットします (RSC)。
- (7) すべての CAM エントリをディスエーブルします。ブロードキャスト・アドレスの受信 (BRD) もディスエーブルします (CAM イネーブル・レジスタと RCR)。

- (8) ループバック動作をディスエーブルします (RCR)。
- (9) ラッチト・バス・リトライをアンラッチト・モードにセットします (DCR)。
- (10) すべての割り込みステータス・ビットをリセットします (ISR)。
- (11) 拡張バス・モードをディスエーブルします (DCR)。
- (12) HOLD をクロックの立ち下がりエッジでハイ/ロウ・レベルにします (DCR2)。
- (13) $\overline{\text{PCOMP}}$ を 3 ステート状態にします (DCR2)。
- (14) CAM エントリに一致したパケットは受理されます (リジェクトされない) (DCR2)。

ソフトウェア・リセットはただちに DMA 動作および割り込みを終了させます。μPD72932はアイドル状態になりレジスタにアクセスできますが、この方法以外に DMA 動作および割り込みを停止する方法はありません。レジスタは表 6-3 に示すとおり、ソフトウェア・リセットによって影響を受けます(コマンド・レジスタだけが変化します)。

表 6-3 リセット後の内部レジスタの状態

レ ジ ス タ 名	リセット後の状態	
	ハードウェア・リセット	ソフトウェア・リセット
コマンド・レジスタ	0094H	0094H/00A4H
データ・コンフィギュレーション・レジスタ (DCR, DCR2)	注1	変化しない
割り込みマスク・レジスタ	0000H	変化しない
割り込みステータス・レジスタ	0000H	変化しない
送信制御レジスタ	0101H	変化しない
受信制御レジスタ	注2	変化しない
エンド・オブ・バッファ・カウント・レジスタ	02F8H	変化しない
シーケンス・カウンタ	0000H	変化しない
CAM イネーブル・レジスタ	0000H	変化しない

注 1. ハードウェア・リセット時, DCR のビット 15, ビット 13, DCR2 のビット 4-ビット 0 は 0 にリセットされます。DCR2 のビット 15-ビット 12 は書き込まれるまで不定です。他のビットはすべて変化しません。

2. ハードウェア・リセット時, LB1, LB0, BRD の各ビットは 0 にリセットされます。他のビットはすべて変化しません。

7. ネットワークへのインタフェース

7.1 内部/外部 ENDEC

μ PD72932 は、AUI (アタッチメント・ユニット・インタフェース) と μ PD72932 の MAC ユニット間のネットワーク・インタフェース動作を実行する ENDEC を内蔵しています。端子の選択により、内部 ENDEC をディスエーブルし、外部 ENDEC に接続するための MAC/ENDEC 信号を供給することができます。EXT 端子が接地された場合 (EXT=0)、内部 ENDEC を選択し、EXT が V_{CC} に接続された場合 (EXT=1)、外部 ENDEC が選択されます。

(1) 内部 ENDEC

内部 ENDEC を使用したときは (EXT=0)、ENDEC と MAC ユニット間のインタフェース信号を内部で接続します。これらの信号は μ PD72932 が内部で使用しますが、ユーザにも出力として供給します (図 7-1)。

内部 ENDEC を使用すれば、 μ PC8392 との 2 チップ構成で完全なイーサネット・インタフェースを実現できます。図 7-2 にネットワーク・インタフェースの簡略図を示します。

(2) 外部 ENDEC

EXT=1 のときには、内部 ENDEC をバイパスし、信号は直接ユーザに提供されます。

図 7-1 MACと内部ENDECのインタフェース信号

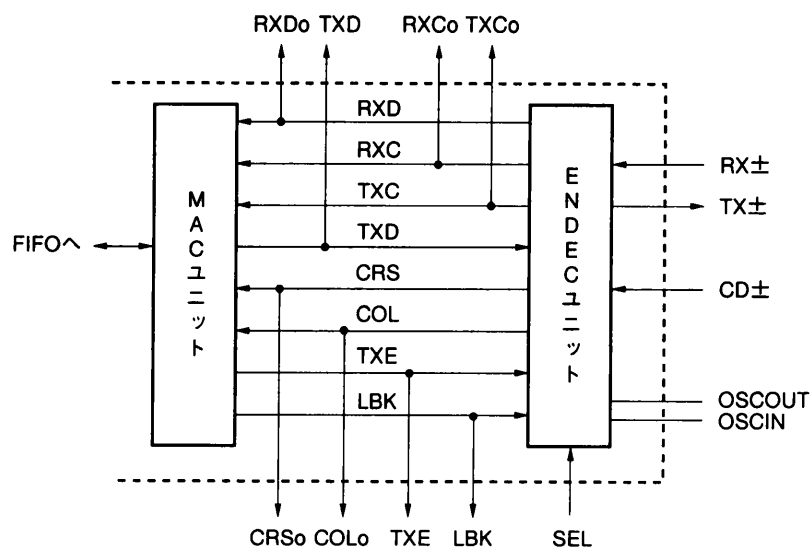
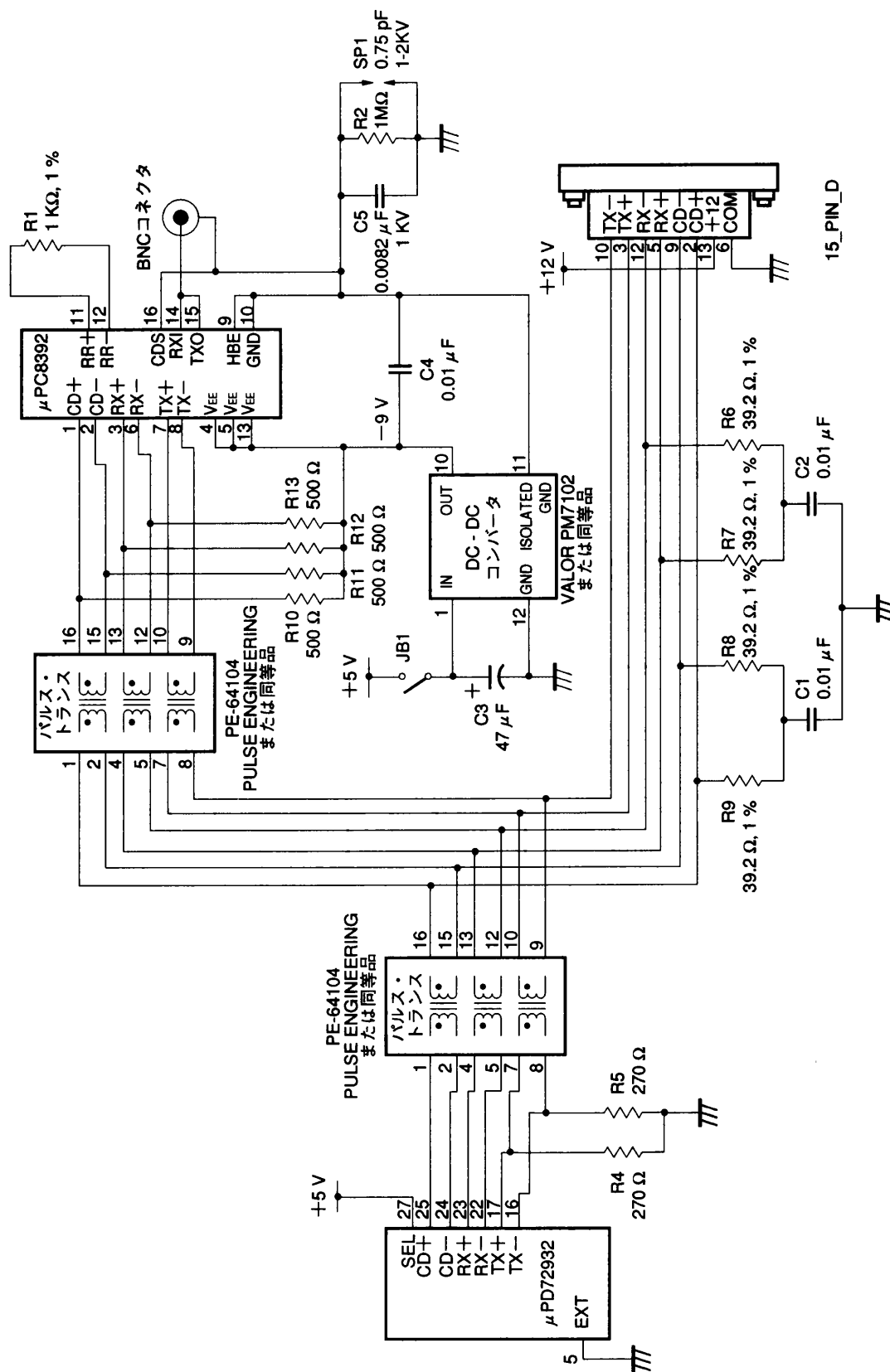


図7-2 ネットワーク・インタフェースの例 (EXT = 0, ネットワーク・インタフェースの選択にシングル・ジャンパ, JB1使用)



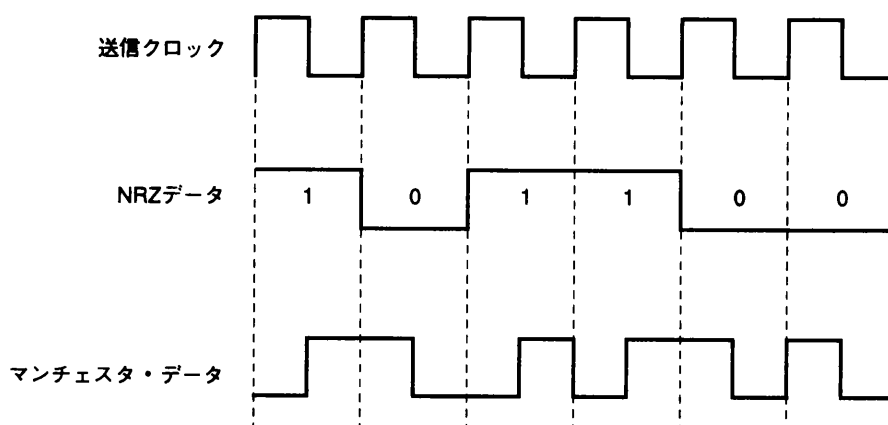
7.2 マンチェスタ・エンコーダおよび差動出力ドライバ

MAC ユニットがシリアル・データ・ストリームの送信を開始すると、ENDEC ユニットのエンコーダが動作を始めます。エンコーダは、差動ドライバ (TX+/-) のために、MAC 部からの NRZ データをマンチェスタ・データに変換します。マンチェスタ・エンコーディングでは、ビット・セルの最初の半分には補数データが含まれ、残りの半分に真データが含まれます (図 7-3)。変化は常にビット・セルの中央で発生します。MAC がデータを送信するかぎり、ENDEC 部は動作を続けます。送信終了時には、最後の変化は常に正になります。この変化は、最後のビットが 1 の場合にはビット・セルの中央で発生し、最後のビットが 0 の場合はビット・セルの終わりで発生します。

差動送信ペアは、最高 50 m の長さのツイスト・ペア AUI ケーブルを駆動します。これらの出力は 2 つの 270 Ω のプルダウン抵抗を接地する必要があるソース・フォロワとなっています。加えて、送信ペア出力と AUI インタフェースの間にはパルス・トランスが必要です。

これらのドライバにより、イーサネット I と IEEE 802.3 の互換性をとるために、ハーフ・ステップ・モードとフル・ステップ・モードの両方が選択可能です。SEL 端子を GND 接地すると (イーサネット I), 絶縁型トランスの一次側でアイドル状態のときは、TX+ は TX- に対して正になります。SEL を V_{CC} に接続すると (IEEE 802.3), TX+ と TX- はアイドル状態のときに等しくなります。

図 7-3 マンチェスタ・エンコード・データ・ストリーム



7.3 マンチェスタ・デコーダ

デコーダは差動レシーバとフェーズ・ロックド・ループ (PLL) から構成され、マンチェスタ・エンコード・データ・ストリームをクロック信号と NRZ データに分離します。差動入力は、直列接続した 2 本の 39 Ω 抵抗により、外部で終端します。加えて、受信入力ペアと AUI インタフェース間にはパルス・トランスが必要です。

ノイズが誤ってデコーダをトリガしないようにするために、入力のスケルチ回路は -175 mV 以下のレベルの信号を除去します。-300 mV よりネガティブな信号もデコードします。

入力がスケルチ要求条件を越えると、デコーダが動作を始めます。デコーダは受信データの 18 ns までのビット・ジッタを許容することができます。デコーダはデータの最終ビットのあと、1.5 ビット・タイム以内にフレームの終わりを検出します。

7.4 コリジョン・トランスレータ

10BASE5/2 トランシーバ(μ PC8392)がコリジョンを検出すると、 μ PD72932の差動衝突入力(CD+および CD-)に対して、10 MHz の信号を生成します。 μ PD72932がこれらの入力がアクティブであることを検出すると、コリジョン・トランスレータはこの10 MHz 信号を内部 ENDEC に送り、さらに MAC ユニットに対してアクティブ・コリジョン (COLo) 信号に変換して送ります。この信号によって、 μ PD72932は現在の送信をアボートし、再送信の実行を再スケジュールします。

コリジョン差動入力、受信差動入力と同様に終端され、コリジョン入力ペアと AUI インタフェース間にはパルス・トランスが必要です。スケルチ回路についても受信ラインと同様に、 -175 mV 以下のレベルのパルスを除去します。

7.5 電源の考慮事項

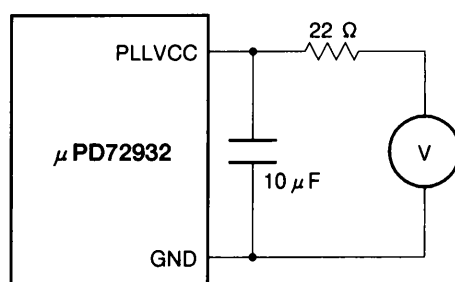
μ PD72932用に電源のパターン・レイアウトを行う際は、ほとんどの場合アナログ電源とデジタル電源を内部接続することができ、一般的な注意事項を守るだけで十分です。しかし、場合によってはアナログ電源(TXVCC, RXVCC, TXGND, ANGND)のレイアウトにおいて、電源ノイズを最小限に抑える必要があります。そうすることによって、 μ PD72932のアナログ機能の最高性能を得ることができます。アナログ電源ノイズを低減するには、次に示す手法のうちのどれかを用いることができます。

- (1) 独自のデカップリング・コンデンサによって、アナログ電源をデジタル電源から独立したトレースおよびプレーンとしてパターン・レイアウトします。
- (2) ロウ・パス・フィルタを挿入してアナログ電源端子にノイズ・フィルタリングを供給します。または、フェライト・ビーズを使用して高周波電源ノイズを低減することもできます。
- (3) 独立したレギュレータを用いて、アナログ電源を生成します。

PLLVCC 端子にロウ・パス・フィルタを挿入した例を図 7-4 に示します。ここでは 1 極 RC フィルタ (遮断周波数が 1 kHz に設計されたもの) を付加しています。この場合、PLLVCC 端子に流れる電流は 3 mA~4 mA となり、抵抗に加わる電圧が 90 mV 以下となるため、PLL の動作は影響を受けません。

PLLVCC 端子 (内部 ENDEC ユニットのフェーズ・ロック・ループ (PLL) 用の +5 V 電源) はアナログ電源のため、PLLVCC 端子に過度のノイズが加わると PLL の性能に悪影響を与えます。また、10 Hz~400 Hz の範囲のノイズが加わると、ENDEC のジッタ性能が低下し、パケット喪失や CRC エラーが発生します。電源ノイズによるパケット受信エラーが深刻な場合は、このような対策によってノイズを除去しジッタ性能を改善してください。

図 7-4 電源ノイズのフィルタリング



8. BバージョンからCバージョンへの変更点

この章では μPD72932BVF (Bバージョン) から μPD72932CVF (Cバージョン) への変更点について説明します。変更内容は大きく分けて次の3つです。

- 動作の改善
- タイミングの改善
- データ・シートの記述修正

動作の改善の項では、Bバージョンで存在し、Cバージョンで修正した不具合動作について説明します。タイミングの改善の項では、BバージョンからCバージョンへ置き換える際に、タイミング上でどのような影響を受けるかを説明します。データ・シートの記述修正の項では、μPD72932の動作をより理解しやすくするために、Cバージョンのデータ・シート（このデータ・シート）で記述を修正した箇所について示します。

8.1 オーダ名称およびドキュメントの変更

次の表は、新旧製品のオーダ名称とドキュメントの一覧です。

表 8-1 新旧製品のオーダ名称と適用ドキュメント

		旧製品	新製品
オーダ名称		μPD72932BVF μPD72932BVF-25	μPD72932CVF-20 μPD72932CVF-25 μPD72932CVF-33
ドキュメント	資料名	μPD72932 データ・シート	μPD72932 データ・シート
	資料番号	IC-8911 (第1版)	S10234JJ3V0 (第3版)
	発行年月	February 1994 P	July 1995 P

8.2 動作の改善

ここでは、スピードおよびタイミングの強化と、Bバージョンでの問題を修正するために行った変更について説明します。

Cバージョンでは、次の項目を修正しました。

- RDA の不正データ書き込み
- TXP のスタック

RDA の不正データ書き込みに関する設計の変更により、パケットの喪失または不正なステータスを引き起こしていた2つの場合について、問題が起こらないようにしました。また、TXP のスタックについても、トランスミッタをロックする原因となっていた2つの場合について、設計を変更して問題が起こらないようにしました。

Bバージョンではこれらの問題を解決するため、ソフトウェアの追加が必要でした。この追加されたソフトウェアによって、Cバージョンの動作に問題は生じません。したがって、BバージョンのソフトウェアはCバージョンと互換性があります。また、追加されたソフトウェアを削除することでコードの最適化が可能です。

スピードおよびタイミングの強化は 9. 電気的特性に反映しています。変更内容の概要について次に説明します。

8.3 タイミングの改善

★

ここでは、μPD72932のタイミングの改善点について説明し、BバージョンからCバージョンへ置き換える際の注意事項を示しています。ほとんどのタイミング（遅延時間、データ・セットアップ時間など）は改善されていますが、ホールド・タイミングが短くなったものがあります。動作を明確にするために新しいタイミングを追加した一方で、冗長性を避けるために削除したタイミングもあります。

Bバージョンの仕様に基づいて設計したもののほとんどは、新しいタイミングを使用しても問題ありません。設計によっては、この新しく改善されたタイミングを使用することによって、バッファ・ロジックの一部を削除したり、システム全体の性能を向上させることもできます。

タイミングの説明は、BMODE=0（NSC/NEC/インテル・モード）の場合とBMODE=1（モトローラ・モード）の場合に分けてあります。それぞれの場合について、さらに次に示す各動作モードに分けて説明しています。

- バス・クロック・タイミング
- メモリ・リード、メモリ・ライト
- レジスタ・リード、レジスタ・ライト
- バス要求、バス・リトライ、メモリ調停/スレーブ・アクセス
- ENDECの送信タイミング/受信タイミング（内部 ENDEC モード）
- ENDEC-MAC 送信シリアル・タイミング/受信シリアル・タイミング（外部 ENDEC モード）

ほとんどのタイミングは以前よりもかなり改善されていますが、Cバージョンへ移行する際には、すべてのシステム・インタフェース・タイミングが有効であるかどうかを確認してください。

8.3.1 BMODE=0（NSC/NEC/インテル・モード）のタイミング修正

ここでは、μPD72932をBMODE=0で使用する場合について説明します。

修正したタイミングの一覧を表8-2に示します。

表 8-2 BMODE=0 のタイミング修正

動作モード	B バージョン	C バージョン	動作モード	B バージョン	C バージョン
バス・クロック・タイミング	T1	T1	レジスタ・リード, レジスタ・ライト	T60	T60a
	T2	T2		T62	T62
メモリ・リード, メモリ・ライト (同期モード)	T9	T9		T62a	—
	T10			T68	T68
	T11	T11 T11b		T70	T70
	T12	T12 T12b		T71	T71
	T36 T40	T36		T73	T73
メモリ・リード, メモリ・ライト (非同期モード)	T9	T10		T76	T76
	T10			T81	T81
	T11b	T11 T11b T11d		T82	T82
	T12b	T12 T12b		T85	T85
	T13	T12d	バス要求, バス・リトライ, メモリ調停/スレーブ・アクセス	T41a	T41a
	T16	T16		T42	T42
	T17	T17		T45	T45
	T36 T40	T36		T46	T46
				T51	T51
				T52	T52
				T55 T55a	T55
				T55b	T55b
				T57	—
				T60	T60
			ENDEC の送信タイミング/受信タイ ミング (内部 ENDEC モード)	T81	T81
				T112	T112
			ENDEC-MAC 送信シリアル・タイ ミング/受信シリアル・タイミング (外部 ENDEC モード)	T118	T118
				T119	T119
				T131	T131
				T132	T132

備考 1. B バージョンと C バージョンで番号が変わっていても、この表に記載されているタイミングはすべて規格を修正しています。

2. B バージョンの欄に 1 つ, C バージョンの欄に複数のタイミングが入っているものは、規格を分けたことを示します。

3. B バージョンの欄に複数, C バージョンの欄に 1 つのタイミングが入っているものは、規格を統合したことを示します。

4. C バージョンの欄に “—” が入っているものは、規格を削除したことを示します。

(1) バス・クロック・タイミング

- T1 “バス・クロック・ロウ・レベル幅” および T2 “バス・クロック・ハイ・レベル幅” の仕様を改善しました。C バージョンへの置き換えの際に影響はありません。

(2) メモリ・リード, メモリ・ライト (同期モード)

- T10 “アドレス・ホールド時間 (対 BSCK↑)” と T9 を統合して T9 にしました。アドレス・ホールド時間は 5 ns から 3 ns に変更しました。C バージョンへの置き換えの際にアドレス・バスにラッチが必要な場合があります。
- T11 を T11 と T11b の 2 つに分けました。2 つともタイミングを改善しました。
- T12 を T12 と T12b の 2 つに分けました。2 つともタイミングを改善しました。
- T36 と T40 “データ・ホールド時間 (対 BSCK↓)” を統合して T36 にしました。新しいデータ遅延時間およびデータ・ホールド時間は、クロックの立ち上がりエッジを基準にしています。データ・ホールド時間は大幅に短くしています。そのため、B バージョンを使用してこのホールド時間に依存する設計を行っているアプリケーション・システムでは、C バージョンへの置き換えの際に、データ・バスにラッチを追加しなければならない場合があります。

(3) メモリ・リード, メモリ・ライト (非同期モード)

- T9 と T10 “アドレス・ホールド時間 (対 BSCK↑)” を統合して T9 にしました。アドレス・ホールド時間は 5 ns から 3 ns に変更しました。C バージョンへの置き換えの際にアドレス・バスにラッチが必要な場合があります。
- T11b を T11, T11b, T11d の 3 つに分けました。3 つともタイミングを改善しました。
- T12b を T12 と T12b の 2 つに分けました。2 つともタイミングを改善しました。
- T13 を T12d に変更し、タイミングを改善しました。
- T16 “DS ハイ・レベル幅” を長くし、T17 “DS ロウ・レベル幅” を短くしました。C バージョンへの置き換えの際に影響はありません。
- T36 と T40 “データ・ホールド時間 (対 BSCK↓)” を統合して T36 にしました。新しいデータ遅延時間およびデータ・ホールド時間は、クロックの立ち上がりエッジを基準にしています。データ・ホールド時間は大幅に短くしています。そのため、B バージョンを使用してこのホールド時間に依存する設計を行っているアプリケーション・システムでは、C バージョンへの置き換えの際に、データ・バスにラッチを追加しなければならない場合があります。

(4) レジスタ・リード, レジスタ・ライト

- $\overline{\text{SAS}}$ タイミング規格およびタイミング・チャートを修正し、スレーブ・サイクルの開始方法を明確にしました。 $\overline{\text{SAS}}$ は $\overline{\text{CS}}$ の立ち下がりエッジと同時またはその前にロウ・レベルにしてください。 $\overline{\text{SAS}}$ と $\overline{\text{CS}}$ をロウ・レベルにすると、 $\overline{\text{SAS}}$ の立ち上がりエッジでレジスタ・アドレスと SWR をラッチします。T62 “ $\overline{\text{SAS}}$ 非同期セットアップ時間 (対 BSCK↓)” の規格を満たすと、スレーブ・サイクルを開始します。この変更により、いくつかのタイミングが変わりました。T60 は T60a に変更し、 $\overline{\text{SAS}}$ がハイ・レベルであることを検出したバス・クロックから $\overline{\text{SMACK}}$ をロウ・レベルにドライブするバス・クロックまでの規格に変更しました。T62 は $\overline{\text{SAS}}$ をハイ・レベルにするときのセットアップ時間を示す規格に変更しました。T62a は妥当でないため削除しました。このタイミングの変更は、B バージョンのタイミング規格よりも柔軟かつ精密なため、C バージョンへの置き換えの際に影響はありません。
- T68, T71 “ SWR ホールド時間 (対 $\overline{\text{SAS}}$ ↑)” および T70, T73 “ SWR セットアップ時間 (対 $\overline{\text{SAS}}$ ↑)” を長くしました。ホスト CPU から出力される SWR は、通常リード/ライト動作に先立ってアクティブとなり、

サイクル全体にわたって保持されるため、C バージョンへの置き換えの際に影響はありません。

- T76 “ $\overline{\text{SAS}} \downarrow / \overline{\text{CS}} \uparrow \rightarrow \overline{\text{RDY}} \circ \uparrow$ 遅延時間” を長くしました。C バージョンへの置き換えの際に影響はありません。
- T81 “ $\text{BSCK} \downarrow \rightarrow \overline{\text{SMACK}} \downarrow$ 遅延時間” を 5 ns 長くしました。
- T82, T85 の規格をそれぞれ改善しました。

(5) バス要求, バス・リトライ, メモリ調停/スレーブ・アクセス

- T41a “ $\overline{\text{BRT}}$ セットアップ時間 (対 $\text{BSCK} \downarrow$)” および T42 “ $\overline{\text{BRT}}$ ホールド時間 (対 $\text{BSCK} \uparrow$)” を長くしました。C バージョンへの置き換えの際に影響はありません。
- T45 “HLDA ハイ同期セットアップ時間” および T46 “HLDA ロウ同期セットアップ時間” を長くしました。HLDA がこれらの規格を満たしていない場合は、同期回路を追加してください。
- T51 “アドレス, $\overline{\text{ADS}}$, $\overline{\text{MWR}}$, $\overline{\text{DS}}$, $\overline{\text{ECS}}$, USR1 , USR0 , EXUSR3-EXUSR0 フロート時間 (対 $\text{BSCK} \uparrow$)” を改善しました。
- T52 “データ・フロート時間 (対 $\text{BSCK} \uparrow$)” を改善しました。
- T55 と T55a を統合して T55 にし、タイミングを改善しました。
- T55b を 10 ns から 3 ns に変更しました。C バージョンへの置き換えの際に影響はありません。
- T57 は冗長なので、データ・シートから削除しました。
- T60 は、 $\overline{\text{MREQ}}$ (スレーブ・モードの場合は $\overline{\text{CS}}$) がロウ・レベルであることを検出したバス・クロックから $\overline{\text{SMACK}}$ をロウ・レベルにドライブするバス・クロックまでの規格に変更しました。これは単に定義の変更であり、C バージョンへの置き換えの際に影響はありません。
- T81 “ $\text{BSCK} \downarrow \rightarrow \overline{\text{SMACK}} \downarrow$ 遅延時間” を改善しました。

(6) ENDEC の送信タイミング/受信タイミング (内部 ENDEC モード)

- T112 “キャリア検知オフ遅延時間” を 155 ns から 250 ns に変更しました。C バージョンへの置き換えの際に影響はありません。

(7) ENDEC-MAC 送信シリアル・タイミング/受信シリアル・タイミング (外部 ENDEC モード)

- T118 “受信クロック・ハイ・レベル幅” および T119 “受信クロック・ロウ・レベル幅” を 35 ns から 40 ns に変更しました。ほとんどの ENDEC 出力は IEEE 802.3 標準に適合するため、C バージョンへの置き換えの際に影響はありません。
- B バージョンのデータ・シートでは T131 と T132 のタイミングが誤っていました。新しく示したタイミングは、十分にテストされた正しい値となっています。

(8) タイミング修正のまとめ

(1) - (7) に示したおもなタイミング変更をすべて調べて問題が見つからなければ、C バージョンへの置き換えを行っても問題ありません。ただし、完全に安全な動作を保証するには、これ以外のタイミングについてもよく調べるのが重要です。

8.3.2 BMODE=1 (モトローラ・モード) のタイミング修正

ここでは、μPD72932 を BMODE=1 で使用する場合について説明します。
修正したタイミングの一覧を表 8-3 に示します。

表 8-3 BMODE=1 のタイミング修正

動作モード	B バージョン	C バージョン	動作モード	B バージョン	C バージョン
バス・クロック・タイミング	T1	T1	バス要求, バス・リトライ, メモリ調停/スレーブ・アクセス	T41a	T41a
	T2	T2		T42	T42
メモリ・リード, メモリ・ライト (同期モード, 非同期モード)	T9	T9		T45a	T45a
	T10			T47	T54
	T11a	T11a T11c T13a		T48	
	T12a	T12a T12c		T49	T54a
	T13a	T13b		T50	
	T14	T14		T51a	T51a
	T15a	T15a		T52	T52
	T16	T16		T53	T53
	T17	T17		—	T54b
	T18	T18		T55	T55
	—	T19		T55a	
	—	T20		T55b	T55b
	T36	T36		T57	—
	T40			T60	T60
	T37	T37a		T81	T81
レジスタ・リード, レジスタ・ライト	T60	T60	ENDEC の送信タイミング/受信タ イミング (内部 ENDEC モード)	T112	T112
	T62	—	ENDEC-MAC 送信シリアル・タイ ミング/受信シリアル・タイミング (外部 ENDEC モード)	T118	T118
	T67	T67		T119	T119
	—	T69		T131	T131
	—	T69a		T132	T132
	T70a	T70a			
	T75a	T75a			
	T75b	T75b			
	—	T77b			
	T81	T81			
	T82	T82			
	T83	T83			
	T86	T86			

備考 1. B バージョンと C バージョンで番号が変わっていても、この表に記載されているタイミングはすべて規格を修正しています。

2. B バージョンの欄に 1 つ、C バージョンの欄に複数のタイミングが入っているものは、規格を分けたことを示します。

3. B バージョンの欄に複数、C バージョンの欄に 1 つのタイミングが入っているものは、規格を統合したことを示します。

4. B バージョンの欄に “—” が入っているものは、規格を追加したことを示します。

5. C バージョンの欄に “—” が入っているものは、規格を削除したことを示します。

(1) バス・クロック・タイミング

- T1 “バス・クロック・ロウ・レベル幅” および T2 “バス・クロック・ハイ・レベル幅” の仕様を改善しました。C バージョンへの置き換えの際に影響はありません。

(2) メモリ・リード、メモリ・ライト（同期モード、非同期モード）

- T9 と T10 “アドレス・ホールド時間（対 B $\overline{\text{SCK}}$ ↑）” を統合して T9 にしました。アドレス・ホールド時間は 5 ns から 3 ns に変更しました。C バージョンへの置き換えの際にアドレス・バスにラッチが必要な場合があります。
- T11a を T11a, T11c, T13a の 3 つに分けました。3 つともタイミングを改善しました。
- T12a を T12a と T12c の 2 つに分けました。2 つともタイミングを改善しました。
- T13a を T13b に変更し、タイミングを改善しました。
- T14 “ $\overline{\text{AS}}$ ロウ・レベル幅” と T15a “ $\overline{\text{AS}}$ ハイ・レベル幅” を長くしました。C バージョンへの置き換えの際に影響はありません。
- T16 “ $\overline{\text{DS}}$ ハイ・レベル幅” を長くし、T17 “ $\overline{\text{DS}}$ ロウ・レベル幅” を短くしました。C バージョンへの置き換えの際に影響はありません。
- T18 “ $\overline{\text{DS}}$ ロウ・レベル幅” を短くしました。C バージョンへの置き換えの際に影響はありません。
- T19 “アドレス・ホールド時間（対 $\overline{\text{AS}}$ ↑）” および T20 “データ・ホールド時間（対 $\overline{\text{AS}}$ ↑）” を新たに追加しました。
- T36 と T40 “データ・ホールド時間（対 B $\overline{\text{SCK}}$ ↓）” を統合して T36 にしました。新しいデータ遅延時間およびデータ・ホールド時間は、クロックの立ち上がりエッジを基準にしています。データ・ホールド時間は大幅に短くしています。そのため、B バージョンを使用してこのホールド時間に依存する設計を行っているアプリケーション・システムでは、C バージョンへの置き換えの際に、データ・バスにラッチを追加しなければならない場合があります。
- T37 を T37a に変更し、タイミングを改善しました。

(3) レジスタ・リード、レジスタ・ライト

- $\overline{\text{SAS}}$ タイミング規格およびタイミング・チャートを修正し、スレーブ・サイクルの開始方法を明確にしました。 $\overline{\text{SAS}}$ は $\overline{\text{CS}}$ をサンプルするクロックの次の立ち下がりエッジの前であればいつでもロウ・レベルにすることができ、 $\overline{\text{SAS}}$ と $\overline{\text{CS}}$ をロウ・レベルにすると、スレーブ・サイクルを開始します。この変更により、いくつかのタイミングが変わりました。T60 は、 $\overline{\text{CS}}$ がロウ・レベルであることを検出したバス・クロックから $\overline{\text{SMACK}}$ をロウ・レベルにドライブするバス・クロックまでの規格に変更しました。T62 は妥当でないため削除しました。 $\overline{\text{SAS}}$ のセットアップ時間を示すため、T69 を追加しました。このタイミングの変更は、B バージョンのタイミング規格よりも柔軟かつ精密なため、C バージョンへの置き換えの際に影響はありません。
- T67, T70a “ $\overline{\text{SRW}}$ セットアップ時間（対 $\overline{\text{SAS}}$ ↓）” を長くしました。ホスト CPU から出力される $\overline{\text{SRW}}$ は、通常リード/ライト動作に先立ってアクティブとなり、サイクル全体にわたって保持されるため、C バージョンへの置き換えの際に影響はありません。
- T75a, T75b の規格をそれぞれ改善しました。
- μPD72932 が $\overline{\text{DSACK1}}$, $\overline{\text{DSACK0}}$ をどのようにハイ・インピーダンスにするのかを明確にするため、T69a および T77b を追加しました。
- T81 “B $\overline{\text{SCK}}$ ↓→ $\overline{\text{SMACK}}$ ↓遅延時間” を 5 ns 長くしました。
- T82, T83, T86 の規格をそれぞれ改善しました。

(4) バス要求, バス・リトライ, メモリ調停/スレーブ・アクセス

- T41a “ $\overline{\text{BRT}}$ セットアップ時間 (対 BSCK↓)” および T42 “ $\overline{\text{BRT}}$ ホールド時間 (対 BSCK↓)” を長くしました。C バージョンへの置き換えの際に影響はありません。
- T45a の規格を “ $\overline{\text{BG}}$, $\overline{\text{AS}}$, $\overline{\text{BGACK}}$, $\overline{\text{DSACK1}}$, $\overline{\text{DSACK0}}$, $\overline{\text{STERM}}$ 非同期セットアップ時間 (対 BSCK↓)” に変更しました。このタイミングを満たすと、μPD72932 はバス・マスタになります。
- T47 と T48 を統合して T54 にし、タイミングを改善しました。
- T49 と T50 を統合して T54a にし、タイミングを改善しました。
- T51a “アドレス, $\overline{\text{AS}}$, $\overline{\text{MRW}}$, $\overline{\text{DS}}$, $\overline{\text{ECS}}$, $\overline{\text{USR1}}$, $\overline{\text{USR0}}$, $\overline{\text{EXUSR3-EXUSR0}}$ フロート時間 (対 BSCK↑)” を改善しました。
- T52 “データ・フロート時間 (対 BSCK↑)” を改善しました。
- T53 の規格を “アドレス, $\overline{\text{AS}}$, $\overline{\text{MRW}}$, $\overline{\text{DS}}$, $\overline{\text{ECS}}$, $\overline{\text{USR1}}$, $\overline{\text{USR0}}$, $\overline{\text{EXUSR3-EXUSR0}}$ 遅延時間 (対 BSCK↑)” に変更しました。このタイミングは短くなっていますが、C バージョンへの置き換えの際に影響はありません。
- T54b “ $\overline{\text{BGACK}}$ 3 ステート遅延時間 (対 BSCK↑)” を追加しました。
- $\overline{\text{BGACK}}$ 信号は、Th ステートから Ti ステートになるとき、いったんハイ・レベルになってからハイ・インピーダンスになります。このタイミングを示すため、T54b を追加しました。
- T55 と T55a を統合して T55 にし、タイミングを改善しました。
- T55b を 10 ns から 3 ns に変更しました。C バージョンへの置き換えの際に影響はありません。
- T57 は冗長なので、データ・シートから削除しました。
- T60 は、 $\overline{\text{CS}}$ または $\overline{\text{MREQ}}$ がロウ・レベルであることを検出したバス・クロックから $\overline{\text{SMACK}}$ をロウ・レベルにドライブするバス・クロックまでの規格に変更しました。これは単に定義の変更であり、C バージョンへの置き換えの際に影響はありません。
- T81 “BSCK↓→ $\overline{\text{SMACK}}$ ↓遅延時間” を改善しました。

(5) ENDEC の送信タイミング/受信タイミング (内部 ENDEC モード)

- T112 “キャリア検知オフ遅延時間” を 155 ns から 250 ns に変更しました。C バージョンへの置き換えの際に影響はありません。

(6) ENDEC-MAC 送信シリアル・タイミング/受信シリアル・タイミング (外部 ENDEC モード)

- T118 “受信クロック・ハイ・レベル幅” および T119 “受信クロック・ロウ・レベル幅” を 35 ns から 40 ns に変更しました。ほとんどの ENDEC 出力は IEEE 802.3 標準に適合するため、C バージョンへの置き換えの際に影響はありません。
- B バージョンのデータ・シートでは T131 と T132 のタイミングが誤っていました。新しく示したタイミングは、十分にテストされた正しい値となっています。

(7) タイミング修正のまとめ

(1) - (6) に示したおもなタイミング変更をすべて調べて問題が見つからなければ、C バージョンへの置き換えを行っても問題ありません。ただし、完全に安全な動作を保証するには、これ以外のタイミングについてもよく調べるのが重要です。

8.4 データ・シートの記述修正

データ・シートの内容をできるだけわかりやすくするため、設計の変更を伴わない改訂を行いました。

(1) OSCIN と OSCOUT

B バージョンのデータ・シートでは、内部 ENDEC で使用するクロック端子の名称は X1 と X2 でした。C バージョンのデータ・シートでは、より意味のある名称にするため、端子名を OSCOUT と OSCIN に変更しました。端子名の対応は X1=OSCOUT, X2=OSCIN となっています。端子の機能を変更していませんので、C バージョンへの置き換えの際に影響はありません。

(2) 外部クロック

9. 電気的特性 外部クロックを、より優れた設計方法を推奨するように改訂しました。 μ PD72932の ENDEC に外部クロック発振器を接続する方法について、B バージョンのデータ・シートでは2つ示していましたが、C バージョンのデータ・シートでは1つだけ示すようにしました。削除した接続方法は、発振回路に追加ドライブ機能が必要で、電力をより多く消費するものでした。この接続方法を使用した古い設計については特にボードの変更を行う必要はありませんが、新たに設計する場合にはC バージョンのデータ・シートで推奨している接続方法を使用してください。

(3) スレーブ・データ・ストロブ ($\overline{\text{SDS}}$) 端子

スレーブ・データ・ストロブ ($\overline{\text{SDS}}$) 端子の説明をC バージョンのデータ・シートでは削除しました。端子接続図に示すように、39番端子は No Connection (NC) になっています。C バージョンはこの端子を内部的にサンプルしません。この端子を V_{CC} や GND に接続しても、何も接続しなくても、信号を入力しても、C バージョンが不正に動作することはありません。既存の設計はこの変更による影響を受けませんが、新たに設計する場合はこの端子に何も接続しないでください。

(4) T72 および T72a の削除

T72 は、設計によって2.5バス・クロック・サイクルであることが保証されているため、データ・シートから削除しました。

T72a は、設計によって2バス・クロック・サイクルであることが保証されているため、データ・シートから削除しました。

(5) AC タイミング・テスト条件

9. 電気的特性 AC タイミング・テスト条件で、出力負荷の図の R_L 値を変更しました。 R_L の変更により、3 ステート・タイミングの精度を改善しています。

9. 電気的特性

絶対最大定格

項 目	略 号	条 件	定 格	単位
電源電圧	V_{CC}		$-0.5 \sim +7.0$	V
入力電圧	V_{IN}		$-0.5 \sim V_{CC} + 0.5$	V
出力電圧	V_{OUT}		$-0.5 \sim V_{CC} + 0.5$	V
動作周囲温度	T_A		$0 \sim 70$	°C
保存温度	T_{stg}		$-65 \sim +150$	°C
消費電力	P_D		500	mW
ESD定格	—	$R_{ZAP} = 1.5 \text{ k}\Omega$, $C_{ZAP} = 120 \text{ pF}$	1.5	kV

注意 各項目のうち1項目でも、また一瞬でも絶対最大定格を越えると、製品の品質を損なう恐れがあります。つまり絶対最大定格とは、製品に物理的な損傷を与えかねない定格値です。必ずこの定格値を越えない状態で、製品をご使用ください。

容 量 ($T_A = 25^\circ\text{C}$)

項 目	略 号	条 件	MIN.	TYP.	MAX.	単位
入力容量	C_{IN}	$f = 1 \text{ MHz}$		7		pF
出力容量	C_{OUT}			7		pF

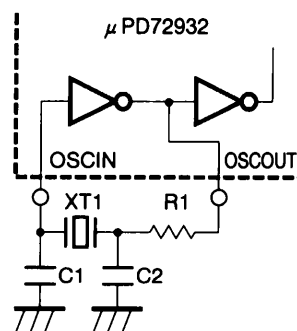
クロック発振回路特性

μPD72932のクロック発振回路入力（OSCOUT および OSCIN）は、水晶振動子（OSCOUT と OSCIN の間に接続）または外部クロック（OSCIN に接続）によって制御します。いずれの場合にも、20 MHz の信号でクロック発振回路の入力を駆動しなければなりません。クロック発振回路の20 MHz 出力は2分周され、MAC 部に送る10 MHz の送信クロック（TXC）を生成します。クロック発振回路はエンコード回路およびデコード回路に内部クロック信号を供給します。

水晶振動子

IEEE 802.3標準で規定されているとおり、送信クロックは0.01 %まで正確でなければなりません。このことは、水晶振動子および他の構成部品を含む発振回路が、クロックを2分周したあとに0.01 %まで正確でなければならないことを意味しています。したがって、水晶振動子を使用する場合、発振回路のすべての面を考慮しなければなりません。発振回路の推奨例と水晶振動子の推奨仕様値を次に示します。負荷容量 C1 と C2 は、すべての浮遊容量を含めてそれぞれ36 pF 以下にします（注意2 参照）。V_{CC} の変化で生じる周波数ドリフトを最小限に抑えるために、抵抗 R1 が必要になります。R1 を使用するとループ利得が低減するので、R1 の値を選択するときには十分に注意してください。抵抗 R1 を大きくしすぎると、ループ利得が大幅に低減し水晶振動子が発振せず、小さくしすぎると、V_{CC} の通常の変動で発振周波数が規格を越えてドリフトすることがあります。最初の目安として、R1 値を水晶振動子の動抵抗の5 倍に設定します。一般に、20 MHz の水晶振動子の動抵抗範囲は10-30 Ω ですから、R1 の適切値は50-100 Ω の範囲になります。各回路のパラメータは不定なので、R1 を含めるか否かは水晶振動子の周波数の測定変動値によって判断します。

水晶振動子の接続例



- 注意 1. OSCOUT端子はTTL互換のロジック出力を供給するように保証されておらず、外部ロジックを駆動するためには使用できません。他のロジックを駆動する必要があるときは、次に説明する外部発振器を使用してください。
2. 水晶振動子に記入されている周波数は、通常水晶振動子のデータ・シートで規定された固定負荷容量で測定されます。使用する実際の負荷容量は、規定値－浮遊容量でなければなりません。

項 目	条 件	MIN.	MAX.	単 位
共振周波数		20		MHz
許容差	T _A = 25 °C	± 0.01		%
精度	T _A = 0 ~ +70 °C	± 0.005		%
基本モード直列抵抗			25	Ω
規定負荷容量			18	pF
タイプ		AT カット		—
回路		並列共振		—

外部クロック

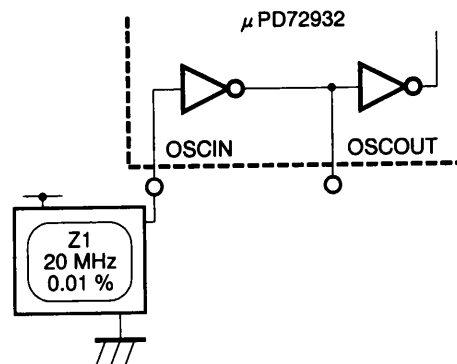
外部クロック発振器を使用する場合、OSCIN に接続することができます。

OSCIN に発振器を接続する場合は、次の特性を備えた発振器を使用しますが、出力ドライブで必要とする CMOS 負荷の数は 1 個のみです。また、この回路構成には消費電力が低くなるという利点があります。この構成では、OSCOUT 端子をオープンに保ち、外部回路を駆動させないでください。

- (1) 0.01 % の周波数許容差を持つ TTL または CMOS
- (2) 40-60 % のデューティ・サイクル

これらの規格は、他の回路が駆動していないことを前提としています。

外部クロックの接続例



PCB のレイアウトに関する考慮事項

水晶振動子を接続する場合には注意が必要です。浮遊容量(たとえばプリント基板のトレースから OSCOUT/OSCIN 端子スルー・ホールまでの)によって水晶振動子の周波数範囲を越え、送信周波数が IEEE で規定された 0.01 % の許容差を越えることがあります。発振回路のレイアウトでは構成部品をすべて OSCOUT, OSCIN 端子の近くに配置し、ショート・トレースを使用して余分な容量とインダクタンスを避けるようにします。同質のグラウンドを使用して 2 個のコンデンサのグラウンド・レッグを接続します。

外部発振器を使用するためのレイアウトは比較的簡潔です。外部発振器を接続するときに考慮すべき点は、発振器をできるかぎり μ PD72932 の近くに配置して浮遊容量とインダクタンスを低減し、発振器にクリーンな V_{CC} と同質のグラウンドを供給することだけです。

DC 特性 ($T_A=0\sim 70^\circ\text{C}$, $V_{CC}=5\text{ V}\pm 5\%$)

項 目	略 号	条 件	MIN.	TYP.	MAX.	単位
ハイ・レベル 入力電圧	V_{IH1}	下記以外	2.0			V
	V_{IH2}	OSCIN (OSCIN は発振器に接続, OSCOUT はオープン)	2.0			V
ロウ・レベル 入力電圧	V_{IL1}	下記以外			0.8	V
	V_{IL2}	OSCIN (OSCIN は発振器に接続, OSCOUT はオープン)			0.8	V
ハイ・レベル出力電圧	V_{OH}	$I_{OH} = -8\text{ mA}$	3.0			V
ロウ・レベル出力電圧	V_{OL}	$I_{OL} = 8\text{ mA}$			0.4	V
TX $\pm$ 差動出力電圧	V_{OD}	終端抵抗78 Ω , 対 GND 間抵抗270 Ω	± 550		± 1200	mV
TX $\pm$ 差動出力電圧 インバランス	V_{OB}	終端抵抗78 Ω , 対 GND 間抵抗270 Ω		40		mV
TX $\pm$ アンダシュート電圧	V_U	終端抵抗78 Ω , 対 GND 間抵抗270 Ω		80		mV
RX $\pm$, CD $\pm$ 差動スケルチ スレッショールド電圧	V_{DS}		-175		-300	mV
入力電流	I_{IN}	$V_{IN} = V_{CC}$ or GND	-1.0		+1.0	μA
3 ステート出力リーク電流	I_{OZ}	$V_{OUT} = V_{CC}$ or GND	-10		+10	μA
電源電流	I_{CC}	$I_{OUT} = 0\text{ mA}$, Freq = f_{MAX}			80	mA
OSCIN 入力ーク電流	I_{OSC}	OSCIN (OSCIN は発振器に接続, OSCOUT はオープン) $V_{IN} = V_{CC}$ or GND	-10		+10	μA

AC 特性 ($T_A=0\sim70\text{ }^{\circ}\text{C}$, $V_{CC}=5\text{ V}\pm5\%$)

★

バス・クロック・タイミング

番 号	項 目	20 MHz		25 MHz		33 MHz		単位
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
T1	バス・クロック・ロウ・レベル幅	22		18		13		ns
T2	バス・クロック・ハイ・レベル幅	22		18		13		ns
T3	バス・クロック・サイクル時間	50	注	40	注	30		ns

注 MAX. 値のテストは行っていないが、116 ns 以上 (8.6 MHz 以下) では、ISR レジスタの TXER ビットが誤ってセットされることが判明しています。

パワーオン・リセット、端子入力によるリセット

番 号	項 目	20 MHz		25 MHz		33 MHz		単位
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
T4	USR1, USR0 セットアップ時間 (対 $\overline{\text{RESET}}\uparrow$)	7		6		5		ns
T5	USR1, USR0 ホールド時間 (対 $\overline{\text{RESET}}\uparrow$)	9		8		7		ns
T6	$\overline{\text{RESET}}$ ホールド時間 (対 $V_{CC}\uparrow$) 注1, 2	10		10		10		TXC
T8	$\overline{\text{RESET}}$ ロウ・レベル幅 注1, 2	10		10		10		TXC

注 1. リセット時間は BSCK と TXC のうち遅いほうによって決まります。BSCK>TXC の場合、T6 と T8 は10 TXC と等しく、BSCK<TXC の場合、T6 と T8 は10 BSCK (T3) に等しくなります。

2. 最大バス・クロック・サイクル時間のテストは実施されていませんが、デバイスの設計上保証されています。

メモリ・ライト, **BMODE=0**, 同期モード (1 ウェイト・ステートを示します)

番 号	項 目	20 MHz		25 MHz		33 MHz		単位
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
T9	アドレス遅延/ホールド時間 (対 BSCK↑)	3	26	3	24	3	22	ns
T11	BSCK↑→ $\overline{\text{ADS}}$ ↓遅延時間		26		24		22	ns
T11b	BSCK↑→ $\overline{\text{ECS}}$ ↓遅延時間		19		17		15	ns
T12	BSCK↑→ $\overline{\text{ADS}}$ ↑遅延時間		24		22		20	ns
T12b	BSCK↓→ $\overline{\text{ECS}}$ ↑遅延時間		29		27		25	ns
T15	$\overline{\text{ADS}}$ ハイ・レベル幅	45		35		25		ns
T32	$\overline{\text{RDYi}}$ セットアップ時間 (対 BSCK↑)	19		17		15		ns
T33	$\overline{\text{RDYi}}$ ホールド時間 (対 BSCK↑)	5		3		3		ns
T36	データ遅延/ホールド時間 (対 BSCK↑)注1	3	50	3	48	3	46	ns
T37	BSCK↑→ $\overline{\text{MWR}}$ ↑遅延時間注2		24		22		20	ns

注 1. RDA および TDA 操作時,最後のライト・サイクルと次のリード・サイクルの間にアイドル・クロック・サイクル (Th) が1つ挿入されます。データ・バスは,アイドル・クロック・サイクル (Th) の次のクロック (Ti) の立ち上がりエッジでハイ・インピーダンスになることに注意してください (データ・フロート時間 (対 BSCK↑) については T52を参照してください)。

2. $\overline{\text{MWR}}$ は連続ライト操作ではハイ・レベルに保持されます。

メモリ・リード, **BMODE=0**, 同期モード (1 ウェイト・ステートを示します)

番 号	項 目	20 MHz		25 MHz		33 MHz		単位
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
T9	アドレス遅延/ホールド時間 (対 BSCK↑)	3	26	3	24	3	22	ns
T11	BSCK↑→ $\overline{\text{ADS}}$ ↓遅延時間		26		24		22	ns
T11b	BSCK↑→ $\overline{\text{ECS}}$ ↓遅延時間		19		17		15	ns
T12	BSCK↑→ $\overline{\text{ADS}}$ ↑遅延時間		24		22		20	ns
T12b	BSCK↓→ $\overline{\text{ECS}}$ ↑遅延時間		29		27		25	ns
T15	$\overline{\text{ADS}}$ ハイ・レベル幅	45		35		25		ns
T23	データ・セットアップ時間 (対 BSCK↑)	5		4		3		ns
T24	データ・ホールド時間 (対 BSCK↑)	5		5		5		ns
T28	BSCK↑→ $\overline{\text{MWR}}$ ↓遅延時間注		26		24		22	ns
T32	$\overline{\text{RDYi}}$ セットアップ時間 (対 BSCK↑)	19		17		15		ns
T33	$\overline{\text{RDYi}}$ ホールド時間 (対 BSCK↑)	5		3		3		ns

注 $\overline{\text{MWR}}$ は連続リード操作ではロウ・レベルに保持されます。

メモリ・ライト, BMODE=0, 非同期モード

番 号	項 目	20 MHz		25 MHz		33 MHz		単位
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
T9	アドレス遅延/ホールド時間 (対 BSCK↑)	3	26	3	24	3	22	ns
T11	BSCK↑→ $\overline{\text{ADS}}$ ↓遅延時間		26		24		22	ns
T11b	BSCK↑→ $\overline{\text{ECS}}$ ↓遅延時間		19		17		15	ns
T11d	BSCK↓→ $\overline{\text{DS}}$ ↓遅延時間		17		15		13	ns
T12	BSCK↑→ $\overline{\text{ADS}}$ ↑遅延時間		24		22		20	ns
T12b	BSCK↓→ $\overline{\text{ECS}}$ ↑遅延時間		29		27		25	ns
T12d	BSCK↓→ $\overline{\text{DS}}$ ↑遅延時間		17		15		13	ns
T15	$\overline{\text{ADS}}$ ハイ・レベル幅	45		35		25		ns
T18	$\overline{\text{DS}}$ ロウ・レベル幅 ^{注1}	40		30		20		ns
T32a	$\overline{\text{RDYi}}$ セットアップ時間 (対 BSCK↓) ^{注2}	5		4		3		ns
T33a	$\overline{\text{RDYi}}$ ホールド時間 (対 BSCK↓)	5		5		5		ns
T36	データ遅延/ホールド時間 (対 BSCK↑) ^{注3}	3	50	3	48	3	46	ns
T37	BSCK↑→ $\overline{\text{MWR}}$ ↑遅延時間 ^{注4}		24		22		20	ns
T39	有効データ→ $\overline{\text{DS}}$ ↓遅延時間	34		21		7		ns

注 1. バス・サイクルにウェイト・ステートが1つ以上挿入されたときのみ、 $\overline{\text{DS}}$ はロウ・レベルになります。

2. このセットアップ時間は、μPD72932が次のバス・クロック (BSCK) でメモリ・サイクルを終了することを保証します。ただし、この場合は非同期入力なので、 $\overline{\text{RDYi}}$ をバス・クロックと同期させる必要はありません。 $\overline{\text{RDYi}}$ はBSCKの立ち下がりエッジの間にサンプルされます。T1 サイクル中に $\overline{\text{RDYi}}$ をサンプルしてロウ・レベルであると、μPD72932は合計2バス・クロックでカレント・アクセスを終了します。T2 (ウェイト) 中に $\overline{\text{RDYi}}$ をサンプルしてロウ・レベルであると、合計3バス・クロックでカレント・アクセスを終了します (プログラム可能なウェイト・ステートを0にセットしている場合)。

3. RDA および TDA 操作時、最後のライト・サイクルと次のリード・サイクルの間にアイドル・クロック・サイクル (Th) が1つ挿入されます。データ・バスは、アイドル・クロック・サイクル (Th) の次のクロック (Ti) の立ち上がりエッジでハイ・インピーダンスになることに注意してください (データ・フロート時間 (対 BSCK↑) については T52を参照してください)。

4. $\overline{\text{MWR}}$ は連続ライト操作ではハイ・レベルに保持されます。

メモリ・リード, BMODE=0, 非同期モード

番 号	項 目	20 MHz		25 MHz		33 MHz		単位
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
T9	アドレス遅延/ホールド時間 (対 BSCK↑)	3	26	3	24	3	22	ns
T11	BSCK↑→ $\overline{\text{ADS}}$ ↓遅延時間		26		24		22	ns
T11b	BSCK↑→ $\overline{\text{ECS}}$ ↓遅延時間		19		17		15	ns
T11d	BSCK↓→ $\overline{\text{DS}}$ ↓遅延時間		17		15		13	ns
T12	BSCK↑→ $\overline{\text{ADS}}$ ↑遅延時間		24		22		20	ns
T12b	BSCK↓→ $\overline{\text{ECS}}$ ↑遅延時間		29		27		25	ns
T12d	BSCK↓→ $\overline{\text{DS}}$ ↑遅延時間		17		15		13	ns
T15	$\overline{\text{ADS}}$ ハイ・レベル幅	45		35		25		ns
T16	$\overline{\text{DS}}$ ハイ・レベル幅	45		35		25		ns
T17	$\overline{\text{DS}}$ ロウ・レベル幅	40		30		20		ns
T23	データ・セットアップ時間 (対 BSCK↑)	5		4		3		ns
T24	データ・ホールド時間 (対 BSCK↑)	5		5		5		ns
T28	BSCK↑→ $\overline{\text{MWR}}$ ↓遅延時間 ^{注1}		26		24		22	ns
T32a	$\overline{\text{RDYi}}$ セットアップ時間 (対 BSCK↓) ^{注2}	5		4		3		ns
T33a	$\overline{\text{RDYi}}$ ホールド時間 (対 BSCK↓)	5		5		5		ns

注 1. $\overline{\text{MWR}}$ は連続リード操作ではロウ・レベルに保持されます。

2. このセットアップ時間は、μPD72932が次のバス・クロック (BSCK) でメモリ・サイクルを終了することを保証します。ただし、この場合は非同期入力なので、 $\overline{\text{RDYi}}$ をバス・クロックと同期させる必要はありません。 $\overline{\text{RDYi}}$ は BSCK の立ち下がりエッジの間にサンプルされます。T1 サイクル中に $\overline{\text{RDYi}}$ をサンプルしてロウ・レベルであると、μPD72932は合計 2 バス・クロックでカレント・アクセスを終了します。T2 (ウェイト) 中に $\overline{\text{RDYi}}$ をサンプルしてロウ・レベルであると、合計 3 バス・クロックでカレント・アクセスを終了します (プログラム可能なウェイト・ステートが 0 にセットしている場合)。

メモリ・ライト, BMODE=1, 同期モード (1 ウェイト・ステートを示します)

番 号	項 目	20 MHz		25 MHz		33 MHz		単位
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
T9	アドレス遅延/ホールド時間 (対 BSCK ↑)	3	26	3	24	3	22	ns
T11a	BSCK ↓ → $\overline{AS}$ ↓ 遅延時間		20		20		20	ns
T11c	BSCK ↑ → $\overline{ECS}$ ↓ 遅延時間		19		17		15	ns
T12a	BSCK ↓ → $\overline{AS}$ ↑ 遅延時間		17		15		13	ns
T12c	BSCK ↓ → $\overline{ECS}$ ↑ 遅延時間		19		17		15	ns
T13a	BSCK ↓ → $\overline{DS}$ ↓ 遅延時間 ^{注1}		16		14		12	ns
T13b	BSCK ↓ → $\overline{DS}$ ↑ 遅延時間 ^{注1}		16		14		12	ns
T14	$\overline{AS}$ ロウ・レベル幅	44		34		24		ns
T15a	$\overline{AS}$ ハイ・レベル幅	45		35		25		ns
T18	$\overline{DS}$ ロウ・レベル幅 ^{注1}	40		30		20		ns
T19	アドレス・ホールド時間 (対 $\overline{AS}$ ↑)	18		14		10		ns
T20	データ・ホールド時間 (対 $\overline{AS}$ ↑)	20		16		12		ns
T22	有効アドレス → $\overline{AS}$ ↓ 遅延時間 ^{注2}	9		6		2		ns
T30	$\overline{DSACK1}$, $\overline{DSACK0}$ セットアップ時間 (対 BSCK ↑) ^{注2}	5		4		3		ns
T31	$\overline{DSACK1}$, $\overline{DSACK0}$ ホールド時間 (対 BSCK ↑)	9		8		7		ns
T36	データ遅延/ホールド時間 (対 BSCK ↑) ^{注3}	3	50	3	48	3	46	ns
T37a	BSCK ↑ → $\overline{MRW}$ ↓ 遅延時間 ^{注4}		26		24		22	ns
T39	有効データ → $\overline{DS}$ ↓ 遅延時間	34		21		7		ns

注 1. バス・サイクルにウェイト・ステートが 1 つ以上挿入されたときのみ, $\overline{DS}$ はロウ・レベルになります。

2. $\overline{DSACK1}$, $\overline{DSACK0}$ は同期モードの間はバス・クロック (BSCK) と同期しなければなりません。

3. RDA および TDA 操作時, 最後のライト・サイクルと次のリード・サイクルの間にアイドル・クロック・サイクル (Th) が 1 つ挿入されます。データ・バスは, アイドル・クロック・サイクル (Th) の次のクロック (Ti) の立ち上がりエッジでハイ・インピーダンスになることに注意してください (データ・フロート時間 (対 BSCK ↑) については T52 を参照してください)。

4. $\overline{MRW}$ は連続ライト操作ではロウ・レベルに保持されます。

メモリ・リード, BMODE=1, 同期モード (1 ウェイト・ステートを示します)

番 号	項 目	20 MHz		25 MHz		33 MHz		単位
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
T9	アドレス遅延/ホールド時間 (対 BSCK↑)	3	26	3	24	3	22	ns
T11a	BSCK↓→ $\overline{AS}$ ↓遅延時間		20		20		20	ns
T11c	BSCK↑→ $\overline{ECS}$ ↓遅延時間		19		17		15	ns
T12a	BSCK↓→ $\overline{AS}$ ↑遅延時間		17		15		13	ns
T12c	BSCK↓→ $\overline{ECS}$ ↑遅延時間		19		17		15	ns
T13a	BSCK↓→ $\overline{DS}$ ↓遅延時間 ^{注1}		16		14		12	ns
T13b	BSCK↓→ $\overline{DS}$ ↑遅延時間 ^{注1}		16		14		12	ns
T14	$\overline{AS}$ ロウ・レベル幅	44		34		24		ns
T15a	$\overline{AS}$ ハイ・レベル幅	45		35		25		ns
T16	$\overline{DS}$ ハイ・レベル幅	45		35		25		ns
T17	$\overline{DS}$ ロウ・レベル幅	40		30		20		ns
T19	アドレス・ホールド時間 (対 $\overline{AS}$ ↑)	18		14		10		ns
T22	有効アドレス→ $\overline{AS}$ ↓遅延時間	9		6		2		ns
T23a	データ・セットアップ時間 (対 BSCK↓)	5		4		3		ns
T24a	データ・ホールド時間 (対 BSCK↓)	5		5		5		ns
T28	BSCK↑→ $\overline{MRW}$ ↑遅延時間 ^{注2}		26		24		22	ns
T30	$\overline{DSACK1}$, $\overline{DSACK0}$ セットアップ時間 (対 BSCK↑) ^{注3}	5		4		3		ns
T31	$\overline{DSACK1}$, $\overline{DSACK0}$ ホールド時間 (対 BSCK↑)	9		8		7		ns

注 1. バス・サイクルにウェイト・ステートが 1 つ以上挿入されたときのみ, $\overline{DS}$ はロウ・レベルになります。

2. $\overline{MRW}$ は連続リード操作ではハイ・レベルに保持されます。

3. $\overline{DSACK1}$, $\overline{DSACK0}$ は同期モードの間はバス・クロック (BSCK) と同期しなければなりません。

メモリ・ライト, BMODE=1, 非同期モード

番 号	項 目	20 MHz		25 MHz		33 MHz		単位
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
T9	アドレス遅延/ホールド時間 (対 BSCK↑)	3	26	3	24	3	22	ns
T11a	BSCK↓→ $\overline{AS}$ ↓遅延時間		20		20		20	ns
T11c	BSCK↑→ $\overline{ECS}$ ↓遅延時間		19		17		15	ns
T12a	BSCK↓→ $\overline{AS}$ ↑遅延時間		17		15		13	ns
T12c	BSCK↓→ $\overline{ECS}$ ↑遅延時間		19		17		15	ns
T13a	BSCK↓→ $\overline{DS}$ ↓遅延時間		16		14		12	ns
T13b	BSCK↓→ $\overline{DS}$ ↑遅延時間		16		14		12	ns
T14	$\overline{AS}$ ロウ・レベル幅	44		34		24		ns
T15a	$\overline{AS}$ ハイ・レベル幅	45		35		25		ns
T18	$\overline{DS}$ ロウ・レベル幅 ^{注1}	40		30		20		ns
T19	アドレス・ホールド時間 (対 $\overline{AS}$ ↑)	18		14		10		ns
T20	データ・ホールド時間 (対 $\overline{AS}$ ↑)	20		16		12		ns
T22	有効アドレス→ $\overline{AS}$ ↓遅延時間	9		6		2		ns
T30	$\overline{DSACK1}$, $\overline{DSACK0}$ セットアップ時間 (対 BSCK↓) ^{注2}	5		4		3		ns
T30a	$\overline{STERM}$ セットアップ時間 (対 BSCK↑) ^{注2}	5		4		3		ns
T31	$\overline{DSACK1}$, $\overline{DSACK0}$ ホールド時間 (対 BSCK↓)	9		8		7		ns
T31a	$\overline{STERM}$ ホールド時間 (対 BSCK↑)	8		7		6		ns
T36	データ遅延/ホールド時間 (対 BSCK↑) ^{注3}	3	50	3	48	3	46	ns
T37a	BSCK↑→ $\overline{MRW}$ ↓遅延時間 ^{注4}		26		24		22	ns
T39	有効データ→ $\overline{DS}$ ↓遅延時間	34		21		7		ns

注 1. バス・サイクルにウエイト・ステートが1つ以上挿入されたときのみ、 $\overline{DS}$ はロウ・レベルになります。

2. $\overline{DSACK1}$, $\overline{DSACK0}$ と $\overline{STERM}$ のセットアップ時間を合わせると、 $\overline{DSACK1}$, $\overline{DSACK0}$ をサンプルした1.5 バス・クロック後、または $\overline{STERM}$ をサンプルした1サイクル後に μPD72932はメモリ・サイクルを終了することができます。 $\overline{DSACK1}$, $\overline{DSACK0}$ または $\overline{STERM}$ をサンプルしてロウ・レベルになるまで、T2 ステートを繰り返します。 $\overline{DSACK1}$, $\overline{DSACK0}$ または $\overline{STERM}$ をそれぞれ T1 または一度目の T2 ステート中にサンプルしてロウ・レベルであると、μPD72932は合計2バス・クロック(3バス・クロックでなく)でカレント・アクセスを終了します(プログラム可能なウエイト・ステートを0にセットしている場合)。 $\overline{DSACK1}$, $\overline{DSACK0}$ は非同期でサンプルされ、 $\overline{STERM}$ は同期でサンプルされます。

3. RDA および TDA 操作時、最後のライト・サイクルと次のリード・サイクルの間にアイドル・クロック・サイクル (Th) が1つ挿入されます。データ・バスは、アイドル・クロック・サイクル (Th) の次のクロック (Ti) の立ち上がりエッジでハイ・インピーダンスになることに注意してください(データ・フロート時間(対 BSCK↑)については T52を参照してください)。

4. $\overline{MRW}$ は連続ライト操作ではロウ・レベルに保持されます。

メモリ・リード, BMODE=1, 非同期モード

番 号	項 目	20 MHz		25 MHz		33 MHz		単位
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
T9	アドレス遅延/ホールド時間 (対 BSCK↑)	3	26	3	24	3	22	ns
T11a	BSCK↓→ $\overline{AS}$ ↓遅延時間		20		20		20	ns
T11c	BSCK↑→ $\overline{ECS}$ ↓遅延時間		19		17		15	ns
T12a	BSCK↓→ $\overline{AS}$ ↑遅延時間		17		15		13	ns
T12c	BSCK↓→ $\overline{ECS}$ ↑遅延時間		19		17		15	ns
T13a	BSCK↓→ $\overline{DS}$ ↓遅延時間		16		14		12	ns
T13b	BSCK↓→ $\overline{DS}$ ↑遅延時間		16		14		12	ns
T14	$\overline{AS}$ ロウ・レベル幅	44		34		24		ns
T15a	$\overline{AS}$ ハイ・レベル幅	45		35		25		ns
T16	$\overline{DS}$ ハイ・レベル幅	45		35		25		ns
T17	$\overline{DS}$ ロウ・レベル幅	40		30		20		ns
T19	アドレス・ホールド時間 (対 $\overline{AS}$ ↑)	18		14		10		ns
T22	有効アドレス→ $\overline{AS}$ ↓遅延時間	9		6		2		ns
T23a	データ・セットアップ時間 (対 BSCK↓)	5		4		3		ns
T24a	データ・ホールド時間 (対 BSCK↓)	5		5		5		ns
T28	BSCK↑→ $\overline{MRW}$ ↑遅延時間 ^{注1}		26		24		22	ns
T30	$\overline{DSACK1}$, $\overline{DSACK0}$ セットアップ時間 (対 BSCK↓) ^{注2}	5		4		3		ns
T30a	$\overline{STERM}$ セットアップ時間 (対 BSCK↑) ^{注2}	5		4		3		ns
T31	$\overline{DSACK1}$, $\overline{DSACK0}$ ホールド時間 (対 BSCK↓)	9		8		7		ns
T31a	$\overline{STERM}$ ホールド時間 (対 BSCK↑)	8		7		6		ns

注 1. $\overline{MRW}$ は連続リード操作ではロウ・レベルに保持されます。

2. $\overline{DSACK1}$, $\overline{DSACK0}$ と $\overline{STERM}$ のセットアップ時間を合わせると, $\overline{DSACK1}$, $\overline{DSACK0}$ をサンプルした1.5バス・クロック後, または $\overline{STERM}$ をサンプルした1サイクル後に μPD72932はメモリ・サイクルを終了することができます。 $\overline{DSACK1}$, $\overline{DSACK0}$ または $\overline{STERM}$ をサンプルしてロウ・レベルになるまで, T2 ステートを繰り返します。 $\overline{DSACK1}$, $\overline{DSACK0}$ または $\overline{STERM}$ をそれぞれ T1 または一度目の T2 ステート中にサンプルしてロウ・レベルであると, μPD72932は合計2バス・クロック (3バス・クロックでなく) でカレント・アクセスを終了します (プログラム可能なウェイト・ステートを0にセットしている場合)。 $\overline{DSACK1}$, $\overline{DSACK0}$ は非同期でサンプルされ, $\overline{STERM}$ は同期でサンプルされます。

バス要求タイミング, BMODE=0

番 号	項 目	20 MHz		25 MHz		33 MHz		単位
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
T43	BSCK ↓/BSCK ↑ → HOLD ↑ 遅延時間 ^{注1}		18		16		14	ns
T44	BSCK ↓/BSCK ↑ → HOLD ↓ 遅延時間 ^{注1}		19		17		15	ns
T45	HLDA ハイ同期セットアップ時間 (対 BSCK ↑)	7	注2	6	注2	5		ns
T46	HLDA ロウ同期セットアップ時間 (対 BSCK ↑) ^{注3}	7	注2	6	注2	5		ns
T51	アドレス, $\overline{ADS}$, $\overline{MWR}$, $\overline{DS}$, $\overline{ECS}$, USR1, USR0, EXUSR3-EXUSR0 フロート時間 (対 BSCK ↑) ^{注4}		34		32		35	ns
T52	データ・フロート時間 (対 BSCK ↑)		39		37		35	ns
T53	USR1, USR0, EXUSR3-EXUSR0 遅延時間 (対 BSCK ↑)		34		32		30	ns
T55	S2-S0 遅延時間 (対 BSCK ↑)		29		27		25	ns
T55b	S2-S0 ホールド時間 (対 BSCK ↑)	3		3		3		ns

- 注 1. HOLD の立ち上がりおよび立ち下りのタイミングは DCR2 の PH ビットによって決まります。タイミング・チャートではデフォルト時 (DCR2 : PH=0) の状態を実線で示しています。T43 および T44 の規格はどちらの場合にも適用できます。また、μPD72932 がバスを要求しているときに HLDA をハイ・レベルにすると、先に HLDA がロウ・レベルになるまでは HOLD はハイ・レベルになりません。
2. HDLC は BSCK の立ち上がりエッジに同期して入力する必要があります。この同期をとるために必要な最大遅延時間は、BSCK 周期 - T45 (MIN.) - 5ns となります。この規格はデバイス設計上の値であり、計測は行っていない。
3. カレント・アクセスの最後の T2 の立ち上がりエッジよりも T46 の期間以上前に HLDA をロウ・レベルにすると、μPD72932 によるバス転送よりも優先してバスを占有することができます。
4. このタイミング値には、測定装置の RC 遅延が含まれています。これらの信号は通常 7 ns 早くハイ・インピーダンスになるので、他のデバイスは競合せずにこれらの端子を駆動することができます。

バス要求タイミング, BMODE=1

番 号	項 目	20 MHz		25 MHz		33 MHz		単位
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
T45a	BG, AS, BGACK, DSACK1, DSACK0, STERM 非同期セットアップ時間 (対 BSCK↓)注	7		6		5		ns
T51a	アドレス, AS, MRW, DS, ECS, USR1, USR0, EXUSR3-EXUSR0 フロート時間 (対 BSCK↑)		39		37		35	ns
T52	データ・フロート時間 (対 BSCK↑)		39		37		35	ns
T53	アドレス, AS, MRW, DS, ECS, USR1, USR0, EXUSR3-EXUSR0 遅延時間 (対 BSCK↑)注		34		32		30	ns
T54	BR↓/3 ステート遅延時間 (対 BSCK↓)		23		23		21	ns
T54a	BGACK↓/BGACK↑遅延時間 (対 BSCK↑)		24		22		20	ns
T54b	BGACK 3 ステート遅延時間 (対 BSCK↑)		24		22		20	ns
T55	S2-S0 遅延時間 (対 BSCK↑)		29		27		25	ns
T55b	S2-S0 ホールド時間 (対 BSCK↑)	3		3		3		ns

注 BGACK は、すべての信号 (BG, AS, BGACK, DSACK1, DSACK0, STERM (拡張バス・モード)) が T45a のセットアップ時間を満たした 1 バス・クロック後にロウ・レベルになります (詳細は 6.3.1 参照)。アドレス・バス, AS, MRW, DS, ECS, USR1, USR0, EXUSR3-EXUSR0 も同じクロックでアクティブになります。

バス・リトライ

番 号	項 目	20 MHz		25 MHz		33 MHz		単位
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
T41	$\overline{\text{BRT}}$ セットアップ時間 (対 BSCK↑) 注1	5		4		3		ns
T41a	$\overline{\text{BRT}}$ セットアップ時間 (対 BSCK↓) 注1	6		5		4		ns
T42	$\overline{\text{BRT}}$ ホールド時間 (対 BSCK↑) 注2	7		6		5		ns

注 1. T41 は同期バス・リトライの、T41a は非同期バス・リトライの規格です (5.3.2, ビット15, 拡張バス・モード参照)。T41a は非同期のセットアップ時間なので、必ずしもこの規格に合わせる必要はありませんが、この規格を満たすことによって、バスの例外処理を次のメモリ転送中ではなく現行のメモリ転送中に行うことができます。

2. ラッチト・バス・リトライ・モード (5.3.2 データ・コンフィギュレーション・レジスタの LBR) をセットしていない場合は、Th ステートのあとまで $\overline{\text{BRT}}$ をロウ・レベルに保持してください。ラッチト・バス・リトライ・モードを使用する場合には、 $\overline{\text{BRT}}$ は T42 の規格を満たす必要はありません。

メモリ調停/スレーブ・アクセス

番 号	項 目	20 MHz		25 MHz		33 MHz		単位
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
T56	$\overline{CS}$ ロウ・セットアップ時間 (対 BSCK↓) 注1	8		7		6		ns
T58	$\overline{MREQ}$ ロウ・セットアップ時間 (対 BSCK↓) 注1	8		7		6		ns
T60	$\overline{MREQ}/\overline{CS}$ 有効→ $\overline{SMACK}$ ↓ 遅延時間 注2,3	1	5	1	5	1	5	bcyc
T80	$\overline{MREQ}$ ↑→ $\overline{SMACK}$ ↑ 遅延時間		18		16		14	ns
T81	BSCK ↓→ $\overline{SMACK}$ ↓ 遅延時間		25		25		25	ns

注 1. $\overline{MREQ}$ および $\overline{CS}$ は非同期でサンプルされるので、必ずしもセットアップ時間を規格に合わせる必要はありませんが、規格を満たすことによって、T60 を使用して $\overline{SMACK}$ がロウ・レベルになる時間を正確に求めることができます。

2. $\overline{CS}$ と $\overline{MREQ}$ がロウ・レベルになったときの μPD72932 の状態により、T60 は最小 1 バス・クロックから最大 5 バス・クロックまでの時間になります。これらの数値のテストは実施されていませんが、デバイスの設計上保証されています。この規格は、非同期で入力される $\overline{CS}$ または $\overline{MREQ}$ をラッチするバス・クロックの立ち下がりエッジの前にこれらの信号がロウ・レベルになった場合を仮定しています (T56 および T58 を参照)。タイミングを正確にするためには、 $\overline{SAS}$ をアクティブにしておかなければなりません。レジスタ・リードおよびレジスタ・ライトのタイミング規格に記載されている $\overline{SAS}$ および $\overline{CS}$ のタイミングを参照してください。

3. bcyc = バス・クロック・サイクル時間 (T3)

レジスタ・リード, BMODE=0

番 号	項 目	20 MHz		25 MHz		33 MHz		単位
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
T56	$\overline{CS}$ 非同期セットアップ時間 (対 BSCK↓) 注1,2	8		7		6		ns
T60a	$\overline{CS}$ ↓, $\overline{SAS}$ ↑→ $\overline{SMACK}$ ↓遅延時間注2,3,4	0	4	0	4	0	4	bccyc
T62	$\overline{SAS}$ 非同期セットアップ時間 (対 BSCK↓) 注1,2	7		6		5		ns
T63a	レジスタ・アドレス・セットアップ時間(対 $\overline{SAS}$ ↑)	7		6		5		ns
T64	レジスタ・アドレス・ホールド時間 (対 $\overline{SAS}$ ↑)	8		7		6		ns
T65	$\overline{SAS}$ ロウ・レベル幅注1,2	20		17		15		ns
T68	SWR ホールド時間 (対 $\overline{SAS}$ ↑)	8		7		6		ns
T73	SWR セットアップ時間 (対 $\overline{SAS}$ ↑)	7		6		5		ns
T75	BSCK↑→RDY ₀ ↓遅延時間		20		18		16	ns
T76	$\overline{SAS}$ ↓/ $\overline{CS}$ ↑→RDY ₀ ↑遅延時間注5		34		32		30	ns
T79	$\overline{SAS}$ ↓/ $\overline{CS}$ ↑→ $\overline{SMACK}$ ↑遅延時間注5		18		16		14	ns
T81	BSCK↓→ $\overline{SMACK}$ ↓遅延時間		25		25		25	ns
T82	レジスタ・データ遅延時間 (対 BSCK↓)		59		57		55	ns
T85	データ・フロート時間 (対 $\overline{SAS}$ ↓/ $\overline{CS}$ ↑) 注5,6		39		37		35	ns
T85a	$\overline{CS}$ ハイ・レベル幅注3	1		1		1		bccyc

- 注 1. $\overline{CS}$ (T56) および $\overline{SAS}$ (T62) は非同期でサンプルされるので、必ずしもセットアップ時間を規格に合わせる必要はありませんが、規格を満たすことによって、T60a を使用して $\overline{SMACK}$ がロウ・レベルになる時間を正確に求めることができます。複数のレジスタ・アクセスを行う場合は、 $\overline{CS}$ をロウ・レベルに保持して $\overline{SAS}$ をスレーブ・サイクルの境界を示すために使用することができます。このとき、 $\overline{SAS}$ が T62 の規格を満たしていれば、μPD72932 は T60a に示す時間のあと $\overline{SMACK}$ をロウ・レベルにします。 $\overline{CS}$ をいったんハイ・レベルにするとときは、正常なスレーブ動作を保証するため必ず T85 の規格を満たしてください。
2. $\overline{CS}$ の立ち下がりエッジと同時にまたはその前であればいつでも $\overline{SAS}$ をロウ・レベルにすることができます。RA5-RA0 および SWR は $\overline{SAS}$ の立ち上がりエッジでラッチされます。 $\overline{SAS}$ が T62 の規格を満たしていれば、μPD72932 は T60a に示す時間のあと $\overline{SMACK}$ をロウ・レベルにします。 $\overline{SAS}$ が T62 の規格を満たしていなければ、μPD72932 は次のクロックの立ち下がりエッジで再び $\overline{SAS}$ をサンプリングします。 $\overline{SAS}$ がハイ・レベルになるまで $\overline{SMACK}$ はロウ・レベルになりません。
3. bccyc=バス・クロック・サイクル時間 (T3)
4. T60a の小さいほうの値は μPD72932 がアイドル状態のときにアクセスされた場合の時間を示し、もう一方は μPD72932 が非アイドル状態のときにアクセスされた場合の時間を示します。これらの数値のテストは実施されていませんが、デバイスの設計上保証されています。
5. $\overline{SAS}$ の立ち下がりエッジの前に $\overline{CS}$ がハイ・レベルになると、T76, T79 および T85 は $\overline{CS}$ の立ち上がりエッジを基準にします。
6. このタイミング値には、測定装置の RC 遅延が含まれています。これらの信号は通常 7 ns 早くハイ・インピーダンスになるので、他のデバイスは競合せずにこれらの端子を駆動することができます。

レジスタ・ライト, BMODE=0

番 号	項 目	20 MHz		25 MHz		33 MHz		単位
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
T56	$\overline{CS}$ 非同期セットアップ時間 (対 BSCK↓) 注1,2	8		7		6		ns
T60a	$\overline{CS}$ ↓, $\overline{SAS}$ ↑→ $\overline{SMACK}$ ↓遅延時間注2,3,4	0	4	0	4	0	4	bcyc
T62	$\overline{SAS}$ 非同期セットアップ時間 (対 BSCK↓) 注1,2	7		6		5		ns
T63a	レジスタ・アドレス・セットアップ時間(対 $\overline{SAS}$ ↑)	7		6		5		ns
T64	レジスタ・アドレス・ホールド時間 (対 $\overline{SAS}$ ↑)	8		7		6		ns
T65	$\overline{SAS}$ ロウ・レベル幅注1,2	20		17		15		ns
T70	SWR セットアップ時間 (対 $\overline{SAS}$ ↑)	7		6		5		ns
T71	SWR ホールド時間 (対 $\overline{SAS}$ ↑)	8		7		6		ns
T75	BSCK↑→ $\overline{RDY}_O$ ↓遅延時間		20		18		16	ns
T76	$\overline{SAS}$ ↓/ $\overline{CS}$ ↑→ $\overline{RDY}_O$ ↑遅延時間注5		34		32		30	ns
T79	$\overline{SAS}$ ↓/ $\overline{CS}$ ↑→ $\overline{SMACK}$ ↑遅延時間注5		18		16		14	ns
T81	BSCK↓→ $\overline{SMACK}$ ↓遅延時間		25		25		25	ns
T83	レジスタ・データ・セットアップ遅延時間 (対 BSCK↓)	14		12		10		ns
T84	レジスタ・データ・ホールド時間 (対 BSCK↓)	14		12		10		ns
T85a	$\overline{CS}$ ハイ・レベル幅注3	1		1		1		bcyc

注 1. $\overline{CS}$ (T56) および $\overline{SAS}$ (T62) は非同期でサンプルされるので、必ずしもセットアップ時間を規格に合わせる必要はありませんが、規格を満たすことによって、T60a を使用して $\overline{SMACK}$ がロウ・レベルになる時間を正確に求めることができます。複数のレジスタ・アクセスを行う場合は、 $\overline{CS}$ をロウ・レベルに保持して $\overline{SAS}$ をスレープ・サイクルの境界を示すために使用することができます。このとき、 $\overline{SAS}$ が T62 の規格を満たしていれば、μPD72932 は T60a に示す時間のあと $\overline{SMACK}$ をロウ・レベルにします。 $\overline{CS}$ をいったんハイ・レベルにするときは、正常なスレープ動作を保証するため必ず T85 の規格を満たしてください。

2. $\overline{CS}$ の立ち下がりエッジと同時またはその前であればいつでも $\overline{SAS}$ をロウ・レベルにすることができます。RA5-RA0 および SWR は $\overline{SAS}$ の立ち上がりエッジでラッチされます。 $\overline{SAS}$ が T62 の規格を満たしていれば、μPD72932 は T60a に示す時間のあと $\overline{SMACK}$ をロウ・レベルにします。 $\overline{SAS}$ が T62 の規格を満たしていなければ、μPD72932 は次のクロックの立ち下がりエッジで再び $\overline{SAS}$ をサンプリングします。 $\overline{SAS}$ がハイ・レベルになるまで $\overline{SMACK}$ はロウ・レベルになりません。

3. bcyc=バス・クロック・サイクル時間 (T3)

4. T60a の小さいほうの値は μPD72932 がアイドル状態のときにアクセスされた場合の時間を示し、もう一方は μPD72932 が非アイドル状態のときにアクセスされた場合の時間を示します。これらの数値のテストは実施されていませんが、デバイスの設計上保証されています。

5. $\overline{SAS}$ の立ち下がりエッジの前に $\overline{CS}$ がハイ・レベルになると、T76, T79 および T85 は $\overline{CS}$ の立ち上がりエッジを基準にします。

レジスタ・リード, BMODE=1

番 号	項 目	20 MHz		25 MHz		33 MHz		単位
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
T56	$\overline{CS}$ 非同期セットアップ時間 (対 BSCK↓) 注1,2	8		7		6		ns
T60	$\overline{CS}$ 有効→ $\overline{SMACK}$ ↓遅延時間注1,2,3	1	5	1	5	1	5	bccyc
T63b	レジスタ・アドレス・セットアップ時間(対 SAS↓)	6		5		4		ns
T64	レジスタ・アドレス・ホールド時間 (対 SAS↓)	8		7		6		ns
T67	SRW セットアップ時間 (対 SAS↓)	4		3		2		ns
T69	SAS ロウ非同期セットアップ時間(対 BSCK↓) 注1,2	7		6		5		ns
T69a	SAS ハイ非同期セットアップ時間(対 BSCK↓) 注1,4	5		4		3		ns
T74	SRW ホールド時間 (対 SAS↓)	8		7		6		ns
T75a	BSCK↓→ $\overline{DSACK1}$ ↓, $\overline{DSACK0}$ ↓遅延時間		26		24		22	ns
T77	$\overline{CS}$ ↑→ $\overline{DSACK1}$ ↑, $\overline{DSACK0}$ ↑遅延時間注4		20		18		16	ns
T77a	SAS↑→ $\overline{DSACK1}$ ↑, $\overline{DSACK0}$ ↑遅延時間注4		24		22		20	ns
T77b	$\overline{DSACK1}$, $\overline{DSACK0}$ フロート時間 (対 BSCK↓) 注4		24		22		20	ns
T78	$\overline{DSACK1}$ ↓, $\overline{DSACK0}$ ↓間スキュー		3		3		2	ns
T79a	BSCK↓→ $\overline{SMACK}$ ↑遅延時間注4		19		17		15	ns
T81	BSCK↓→ $\overline{SMACK}$ ↓遅延時間		25		25		25	ns
T82	レジスタ・データ遅延時間 (対 BSCK↓)		59		57		55	ns
T85a	$\overline{CS}$ ハイ・レベル幅注1,3	1		1		1		bccyc
T86	レジスタ・データ・フロート時間 (対 SAS↑) 注5		49		47		45	ns

- 注 1. $\overline{CS}$ (T56) および SAS (T69) は非同期でサンプルされるので、必ずしもセットアップ時間を規格に合わせる必要はありませんが、規格を満たすことによって、T60を使用して $\overline{SMACK}$ がロウ・レベルになる時間を正確に求めることができます。 $\overline{SAS}$ は $\overline{CS}$ をサンプルするクロックの次の立ち下がりエッジの前 (T69の規格を参照してください) であればいつでもロウ・レベルにすることができます。複数のレジスタ・アクセスを行う場合は、 $\overline{CS}$ をロウ・レベルに保持して $\overline{SAS}$ をスレーブ・サイクルの境界を示すために使用することができます (サイクルを終了させて次のサイクルを開始するため、必ず T69a の規格を満たしてください)。この連続したレジスタ・アクセスでは、2つ目のレジスタ以降、 $\overline{SMACK}$ は T69のタイミングが満足されると同時にロウ・レベルになります。
2. $\overline{CS}$ がロウ・レベルになったときの μPD72932の状態により、T60は最小1バス・クロックから最大5バス・クロックまでの時間になります。これらの数値のテストは実施されていませんが、デバイスの設計上保証されています。この規格は、 $\overline{CS}$ が T56の規格を、 $\overline{SAS}$ が T69の規格を満たしている場合を仮定しています。また、T60の規格は、カレント・マスタ・モード・アクセスにウエイト・ステートがないことを仮定しています (μPD72932がマスタ・モードのときに $\overline{CS}$ がロウ・レベルになった場合)。ウエイト・ステートがあると、T60は規格より長くなります。
3. bccyc=バス・クロック・サイクル時間 (T3)
4. SAS (T69a) は非同期でサンプルされるので、必ずしもセットアップ時間を規格に合わせる必要はありませんが、規格を満たすことによって、T1 ステートの立ち下がりエッジで $\overline{DSACK1}$, $\overline{DSACK0}$ をハイ・インピーダンスに (T77b), $\overline{SMACK}$ をハイ・レベルに (T79a) することができます。 $\overline{DSACK1}$, $\overline{DSACK0}$ は、 $\overline{CS}$ と $\overline{SAS}$ のどちらによってもロウ・レベルにすることができますが、 $\overline{DSACK1}$, $\overline{DSACK0}$ をハイ・インピーダンスにすることができるのは $\overline{SAS}$ のみです。
5. このタイミング値には、測定装置の RC 遅延が含まれています。これらの信号は通常7ns 早くハイ・インピーダンスになるので、他のデバイスは競合せずにこれらの端子を駆動することができます。

レジスタ・ライト, BMODE=1

番 号	項 目	20 MHz		25 MHz		33 MHz		単位
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
T56	$\overline{CS}$ 非同期セットアップ時間 (対 $\overline{BSCK}$ ↓) 注1,2	8		7		6		ns
T60	$\overline{CS}$ 有効→ $\overline{SMACK}$ ↓ 遅延時間注1,2,3	1	5	1	5	1	5	bcyc
T63b	レジスタ・アドレス・セットアップ時間 (対 $\overline{SAS}$ ↓)	6		5		4		ns
T64	レジスタ・アドレス・ホールド時間 (対 $\overline{SAS}$ ↓)	8		7		6		ns
T69	$\overline{SAS}$ ロウ非同期セットアップ時間 (対 $\overline{BSCK}$ ↓) 注1,2	7		6		5		ns
T69a	$\overline{SAS}$ ハイ非同期セットアップ時間 (対 $\overline{BSCK}$ ↓) 注1,4	5		4		3		ns
T70a	$\overline{SRW}$ セットアップ時間 (対 $\overline{SAS}$ ↓)	4		3		2		ns
T71a	$\overline{SRW}$ ホールド時間 (対 $\overline{SAS}$ ↓)	8		7		6		ns
T75b	$\overline{BSCK}$ ↓ → $\overline{DSACK1}$ ↓, $\overline{DSACK0}$ ↓ 遅延時間		26		24		22	ns
T77	$\overline{CS}$ ↑ → $\overline{DSACK1}$ ↑, $\overline{DSACK0}$ ↑ 遅延時間注4		20		18		16	ns
T77a	$\overline{SAS}$ ↑ → $\overline{DSACK1}$ ↑, $\overline{DSACK0}$ ↑ 遅延時間注4		24		22		20	ns
T77b	$\overline{DSACK1}$, $\overline{DSACK0}$ フロート時間 (対 $\overline{BSCK}$ ↓) 注4		24		22		20	ns
T78	$\overline{DSACK1}$ ↓, $\overline{DSACK0}$ ↓ 間スキュー		3		3		2	ns
T79a	$\overline{BSCK}$ ↓ → $\overline{SMACK}$ ↑ 遅延時間注4		19		17		15	ns
T81	$\overline{BSCK}$ ↓ → $\overline{SMACK}$ ↓ 遅延時間		25		25		25	ns
T83	レジスタ・データ・セットアップ時間 (対 $\overline{BSCK}$ ↓)	14		12		10		ns
T84	レジスタ・データ・ホールド時間 (対 $\overline{BSCK}$ ↓)	14		12		10		ns
T85a	$\overline{CS}$ ハイ・レベル幅注1,3	1		1		1		bcyc

- 注 1. $\overline{CS}$ (T56) および $\overline{SAS}$ (T69) は非同期でサンプルされるので、必ずしもセットアップ時間を規格に合わせる必要はありませんが、規格を満たすことによって、T60を使用して $\overline{SMACK}$ がロウ・レベルになる時間を正確に求めることができます。 $\overline{SAS}$ は $\overline{CS}$ をサンプルするクロックの次の立ち下がりエッジの前 (T69の規格を参照してください) であればいつでもロウ・レベルにすることができます。複数のレジスタ・アクセスを行う場合は、 $\overline{CS}$ をロウ・レベルに保持して $\overline{SAS}$ をスレーブ・サイクルの境界を示すために使用することができます (サイクルを終了させて次のサイクルを開始するため、必ず T69a の規格を満たしてください)。この連続したレジスタ・アクセスでは、2 つ目のレジスタ以降、 $\overline{SMACK}$ は T69のタイミングが満足されると同時にロウ・レベルになります。
2. $\overline{CS}$ がロウ・レベルになったときの μPD72932 の状態により、T60は最小 1 バス・クロックから最大 5 バス・クロックまでの時間になります。これらの数値のテストは実施されていませんが、デバイスの設計上保証されています。この規格は、 $\overline{CS}$ が T56の規格を、 $\overline{SAS}$ が T69の規格を満たしている場合を仮定しています。また、T60の規格は、カレント・マスタ・モード・アクセスにウエイト・ステートがないことを仮定しています (μPD72932がマスタ・モードのときに $\overline{CS}$ がロウ・レベルになった場合)。ウエイト・ステートがあると、T60は規格より長くなります。
3. bcyc=バス・クロック・サイクル時間 (T3)
4. $\overline{SAS}$ (T69a) は非同期でサンプルされるので、必ずしもセットアップ時間を規格に合わせる必要はありませんが、規格を満たすことによって、T1 ステートの立ち下がりエッジで $\overline{DSACK1}$, $\overline{DSACK0}$ をハイ・インピーダンスに (T77b), $\overline{SMACK}$ をハイ・レベルに (T79a) することができます。 $\overline{DSACK1}$, $\overline{DSACK0}$ は、 $\overline{CS}$ と $\overline{SAS}$ のどちらによってもロウ・レベルにすることができますが、 $\overline{DSACK1}$, $\overline{DSACK0}$ をハイ・インピーダンスにすることができるのは $\overline{SAS}$ のみです。

ENDEC の送信タイミング (内部 ENDEC モード)

番 号	項 目	MIN.	TYP.	MAX.	単位
T87	送信クロック・ハイ・レベル幅 ^注	40			ns
T88	送信クロック・ロウ・レベル幅 ^注	40			ns
T89	送信クロック・サイクル時間 ^注	99.99		100.01	ns
T95	送信出力遅延時間 ^注			55	ns
T96	送信出力立ち下がり時間 (80 %→20 %) ^注			7	ns
T97	送信出力立ち上がり時間 (20 %→80 %) ^注			7	ns
T98	送信出力ジッタ (タイミング・チャートに記載していません)		0.5		ns
T100	アイドル状態 (ハーフ・ステップ) 前の送信出力ハイ・レベル幅	200			ns
T101	送信出力アイドル時間 (ハーフ・ステップ)			8000	ns

注 この規格はあくまでも参考用で、計測されていません。

ENDEC の受信タイミング (内部 ENDEC モード)

番 号	項 目	MIN.	TYP.	MAX.	単位
T102	受信クロック・デューティ・サイクル時間 ^{注1}	40		60	ns
T105	キャリア検知時間			70	ns
T106	データ獲得時間			700	ns
T107	受信データ出力遅延時間			150	ns
T108	有効受信データ遅延時間 (対 RXC↓)			10	ns
T109	受信データ安定有効時間	90			ns
T112	キャリア検知オフ遅延時間 ^{注2}			250	ns
T113	CRS 立ち下がり後の受信クロック数 ^{注3}	5			rcyc
T114	コリジョン・ターン・オン時間			55	ns
T115	コリジョン・ターン・オフ時間			250	ns

注 1. この数値は各クロック・エッジの50 %点を使用して計測されています。

2. CRSi がロウ・レベルになると、最小2 受信クロックの間はロウ・レベルのままになります。

3. rcyc=受信クロック

ENDEC-MAC 受信シリアル・タイミング (外部 ENDEC モード)

番 号	項 目	MIN.	TYP.	MAX.	単位
T118	受信クロック・ハイ・レベル幅	40			ns
T119	受信クロック・ロウ・レベル幅	40			ns
T120	受信クロック・サイクル時間	90		110	ns
T121	受信データ・セットアップ時間 (対 RXC↑)	20			ns
T122	受信データ・ホールド時間 (対 RXC↑)	15			ns
T124	許容されるドリブル・ビット数			6	ビット
T125	受信リカバリ時間		注1		—
T126	RXC↓→CRS↓遅延時間注2,3			1	rcyc

注 1. この数値は μPD72932が受信動作を終了して次のフレーム開始デリミタの準備を終えるまでの最長時間 (ウェイト・ステートを除く) を基準にしています。この時間は $4\text{ tcy} + 36\text{ bcyc}$ (tcyc=送信クロック, bcyc=T3) となり、デバイスの設計上保証されていますが、テストされていません。

2. rcyc=受信クロック

3. 正常な受信動作を保証するため、CRS がロウ・レベルになってから 5 受信クロック以上待つ必要があります。

ENDEC-MAC 送信シリアル・タイミング (衝突なし)

番 号	項 目	MIN.	TYP.	MAX.	単位
T127	送信クロック・ハイ・レベル幅	40			ns
T128	送信クロック・ロウ・レベル幅	40			ns
T129	送信クロック・サイクル時間	90		110	ns
T130	TXC↑→TXE↑遅延時間			40	ns
T131	送信データ遅延時間 (対 TXC↑)			40	ns
T132	送信データ・ホールド時間 (対 TXC↑)	0			ns
T133	TXC↑→TXE↓遅延時間			40	ns
T134	CD ハートビート開始遅延時間 (対 TXE↓)注			64	tcyc
T135	コリジョン検知幅注	2			tcyc

注 tcy=送信クロック

ENDEC-MAC 送信シリアル・タイミング (衝突)

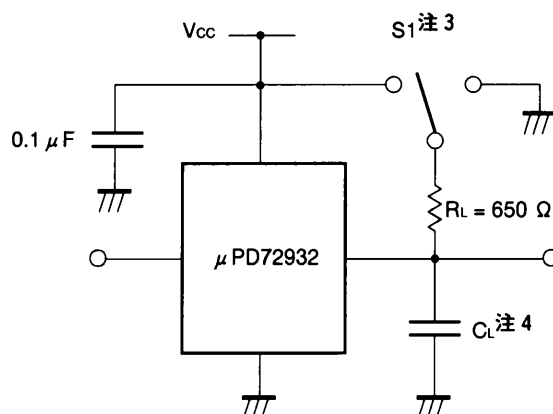
番 号	項 目	MIN.	TYP.	MAX.	単位
T135	コリジョン検知幅注	2			tcyc
T136	ジャム・パターン遅延時間 (対 COL↑)注			8	tcyc
T137	ジャム・パターン出力時間注			32	tcyc

注 tcy=送信クロック

AC タイミング・テスト条件

すべての規格は、絶対条件のアイソレーションを適用し、またすべての差動信号をパルス・トランスの AUI 側に取る場合に限り有効です。

項 目	MIN.	TYP.	MAX.	単位
入力パルス・レベル (TTL/CMOS)	GND		3.0	V
入力立ち上がりおよび立ち下がり時間 (TTL/CMOS)		5		ns
入出力基準レベル (TTL/CMOS)		1.5		V
入力パルス・レベル (差動)	-1315		-350	mV
入出力基準レベル (差動)		注1		V
3 ステート基準レベル	$\Delta V - 0.5$ 注2		$\Delta V + 0.5$ 注2	V
出力負荷	下図参照			-



注1. 差動レベルの50 %点

2. ΔV = 浮動電圧

3. S1の状態

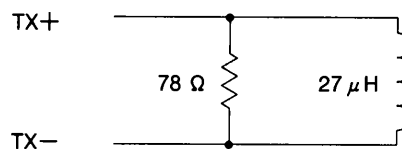
- ・ プッシュプル出力のタイミング・テストではオープン
- ・ VOLテストではVcc
- ・ VOHテストではGND
- ・ ハイ・インピーダンスからアクティブ・ロウ・レベル、および
アクティブ・ロウ・レベルからハイ・インピーダンス測定ではVcc
- ・ ハイ・インピーダンスからアクティブ・ハイ・レベル、および
アクティブ・ハイ・レベルからハイ・インピーダンス測定ではGND

4. 50 pF, スコープおよびジグ容量を含む

ディレーティング係数

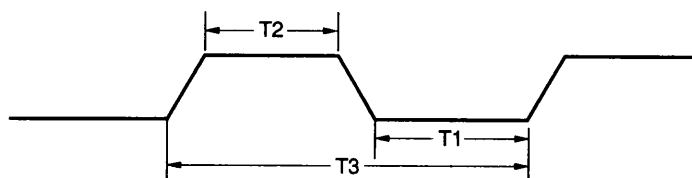
出力タイミングは 50 pF の純粋な容量性負荷で測定します。その他の負荷には $C_L \geq 50 \text{ pF} + 0.05 \text{ ns/pF}$ の補正係数を使用することができます。

AUI送信テスト・ロード

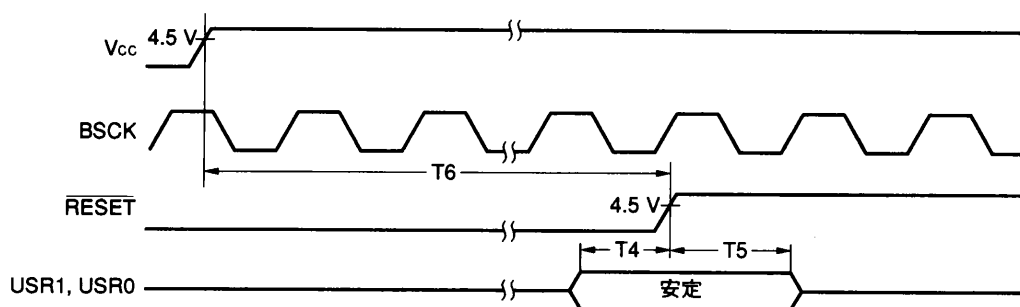


備考 上図で、TX+信号およびTX-信号はアイソレーション（パルス・トランス）のAUI側から取っています。すべてのテストに使用したパルス・トランスは、 $100 \mu\text{H} \pm 0.1\%$ のパルス・エンジニアリング社製PE64103です。

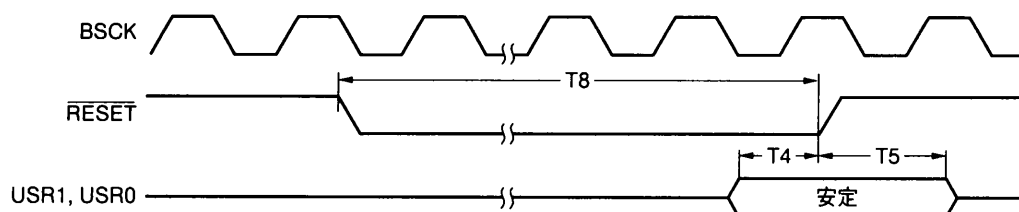
バス・クロック・タイミング



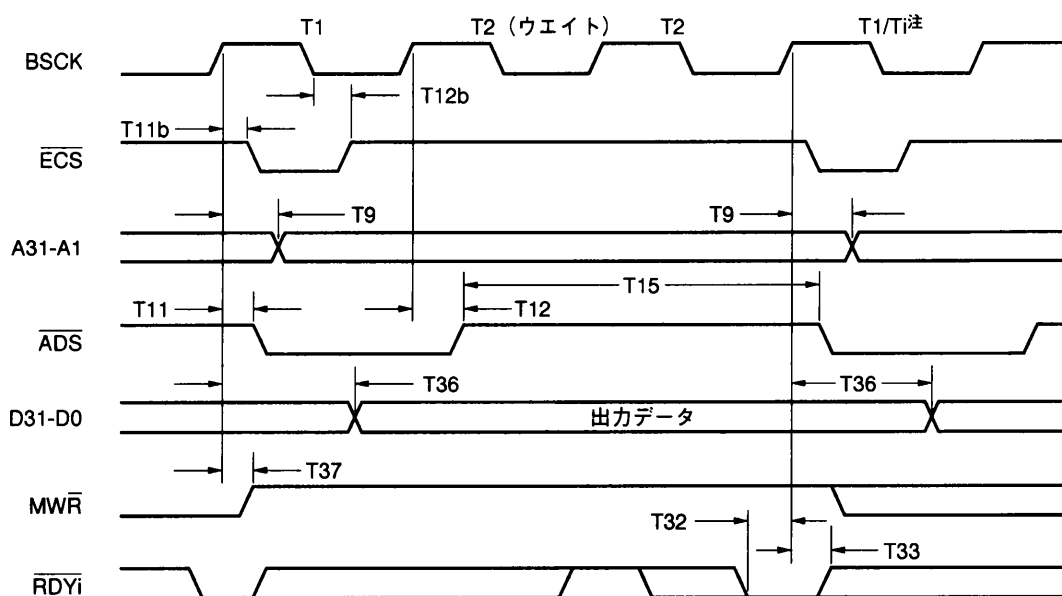
パワー・オン・リセット



端子入力によるリセット

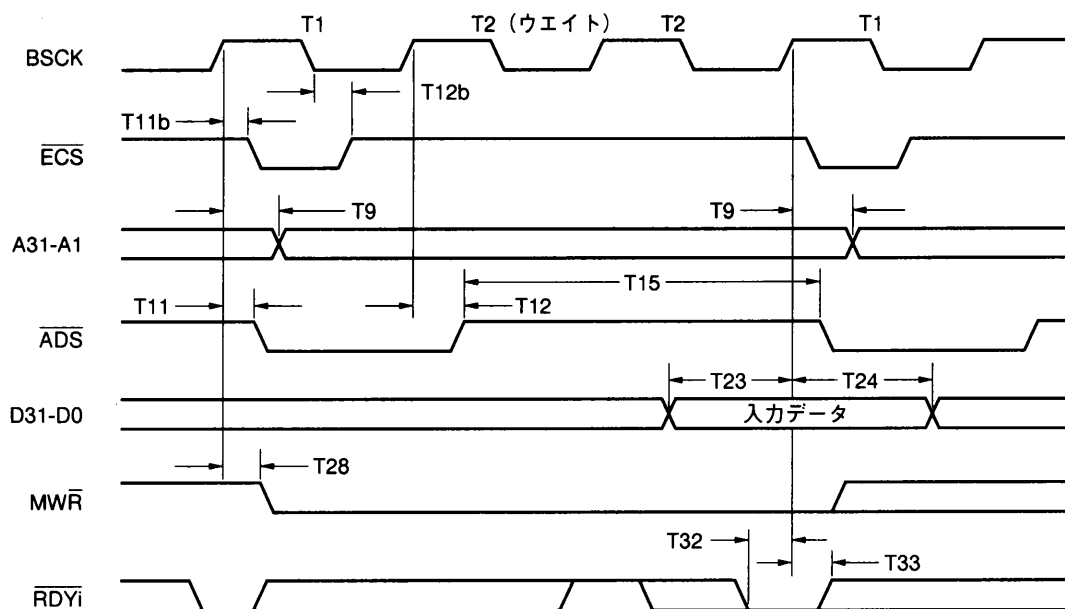


メモリ・ライト, BMODE = 0, 同期モード (1 ウェイト・ステートを示します)

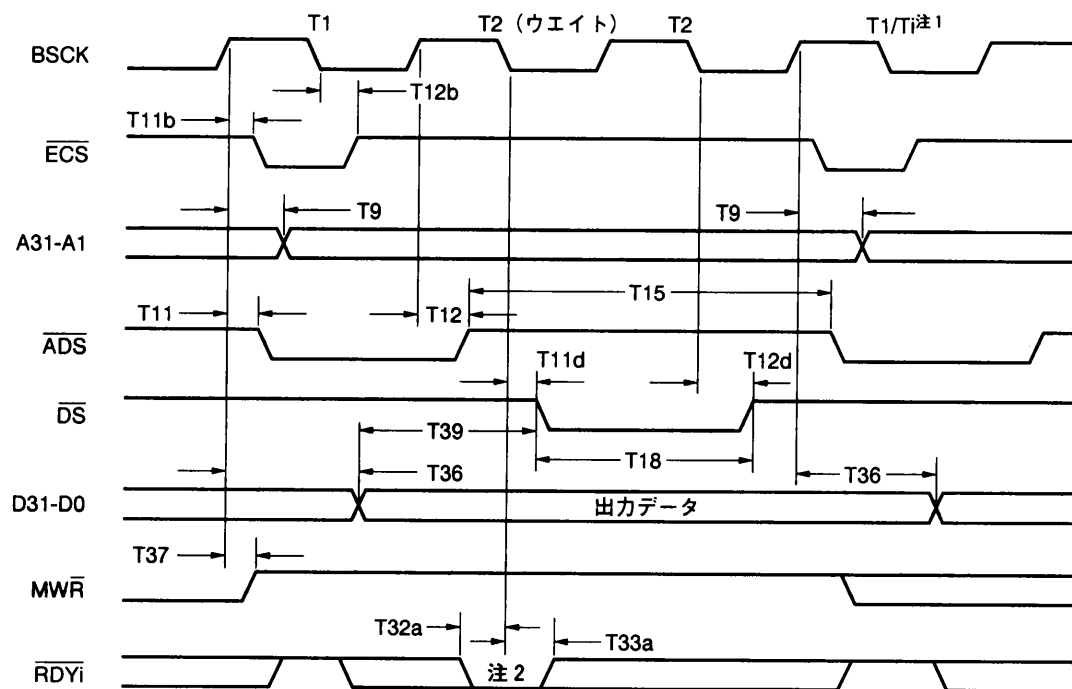


注 RDAおよびTDA操作時、最後のライト・サイクルと次のリード・サイクルの間にアイドル・クロック・サイクル (Ti) が1つ挿入されます。データ・バスは、アイドル・サイクルの次のクロックの立ち上がりエッジでハイ・インピーダンスになることに注意してください (データ・フロート時間 (対BSCK↑) についてはT52を参照してください)。

メモリ・リード, BMODE = 0, 同期モード (1 ウェイト・ステートを示します)



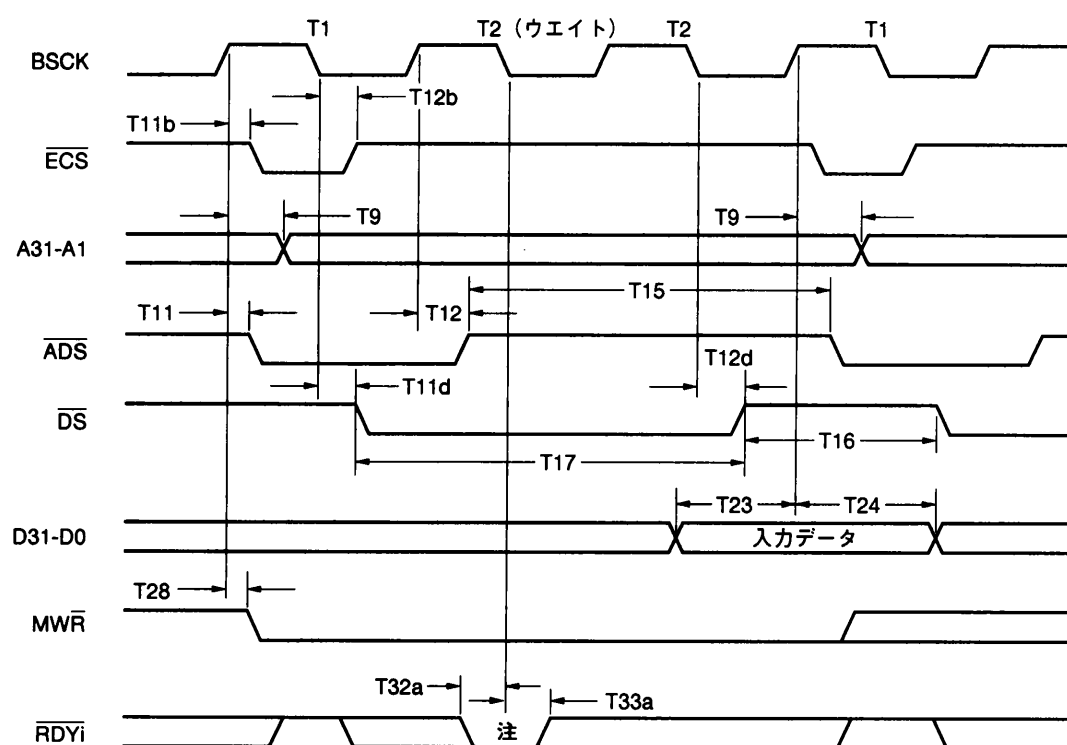
メモリ・ライト, BMODE = 0, 非同期モード



注1. RDAおよびTDA操作時、最後のライト・サイクルと次のリード・サイクルの間にアイドル・クロック・サイクル (Ti) が1つ挿入されます。データ・バスは、アイドル・サイクルの次のクロックの立ち上がりエッジでハイ・インピーダンスになることに注意してください (データ・フロート時間 (対BSCK↑) についてはT52を参照してください)。

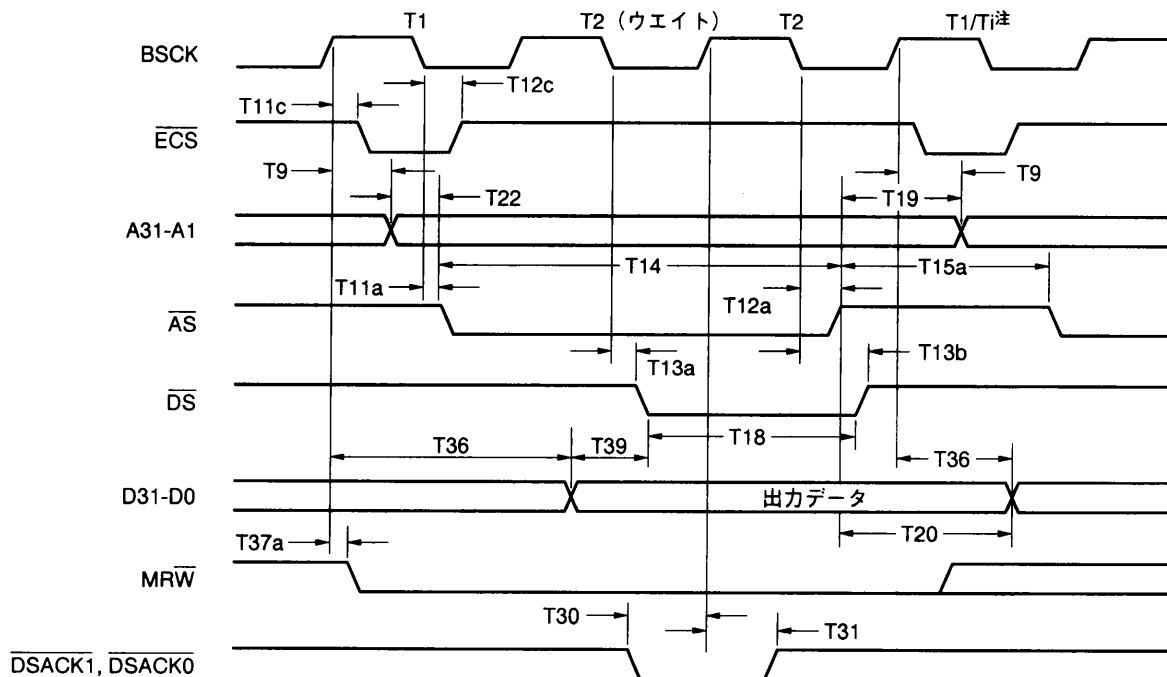
- このセットアップ時間は、μPD72932が次のバス・クロック (BSCK) でメモリ・サイクルを終了することを保証します。ただし、この場合は非同期入力なので、RDYiをバス・クロックと同期させる必要はありません。RDYiはBSCKの立ち下がりエッジの間にサンプルされます。T1サイクル中にRDYiをサンプルしてロウ・レベルであると、μPD72932は合計2バス・クロックでカレント・アクセスを終了します。T2 (ウェイト) 中にRDYiをサンプルしてロウ・レベルであると、合計3バス・クロックでカレント・アクセスを終了します (プログラム可能なウェイト・ステートを0にセットしている場合)。

メモリ・リード, BMODE = 0, 非同期モード



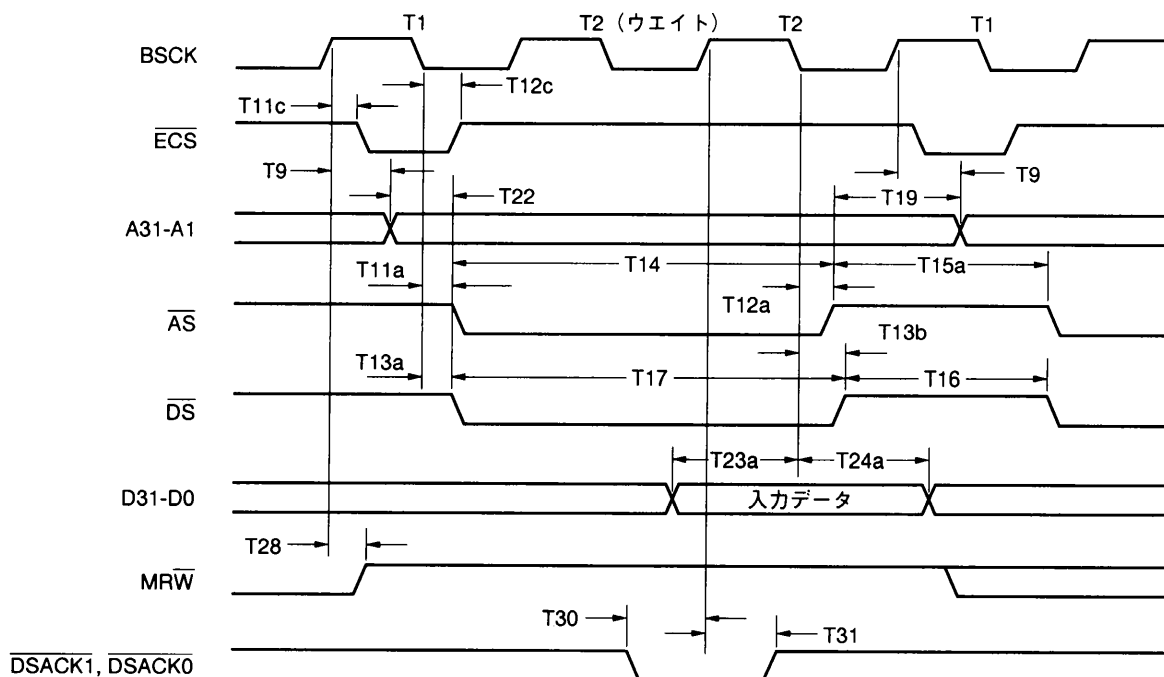
注 このセットアップ時間は、μPD72932が次のバス・クロック（BSCCK）でメモリ・サイクルを終了することを保証します。ただし、この場合は非同期入力なので、RDYiをバス・クロックと同期させる必要はありません。RDYiはBSCCKの立ち下がりエッジの間にサンプルされます。T1サイクル中にRDYiをサンプルしてロウ・レベルであると、μPD72932は合計2バス・クロックでカレント・アクセスを終了します。T2（ウェイト）中にRDYiをサンプルしてロウ・レベルであると、合計3バス・クロックでカレント・アクセスを終了します（プログラム可能なウェイト・ステートを0にセットしている場合）。

メモリ・ライト, BMODE = 1, 同期モード (1 ウェイト・ステートを示します)

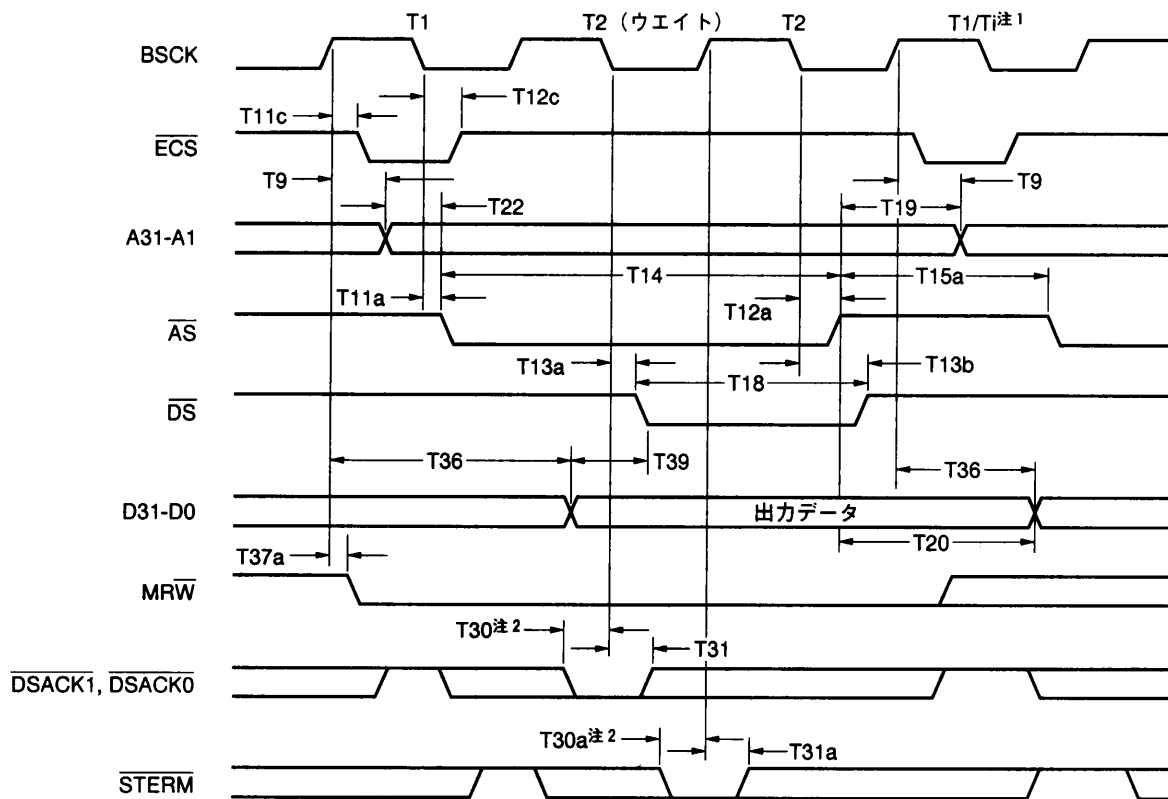


注 RDAおよびTDA操作時、最後のライト・サイクルと次のリード・サイクルの間にアイドル・クロック・サイクル (Ti) が1つ挿入されます。データ・バスは、アイドル・サイクルの次のクロックの立ち上がりエッジでハイ・インピーダンスになることに注意してください (データ・フロート時間 (対BSCK↑) についてはT52を参照してください)。

メモリ・リード, BMODE = 1, 同期モード (1 ウェイト・ステートを示します)



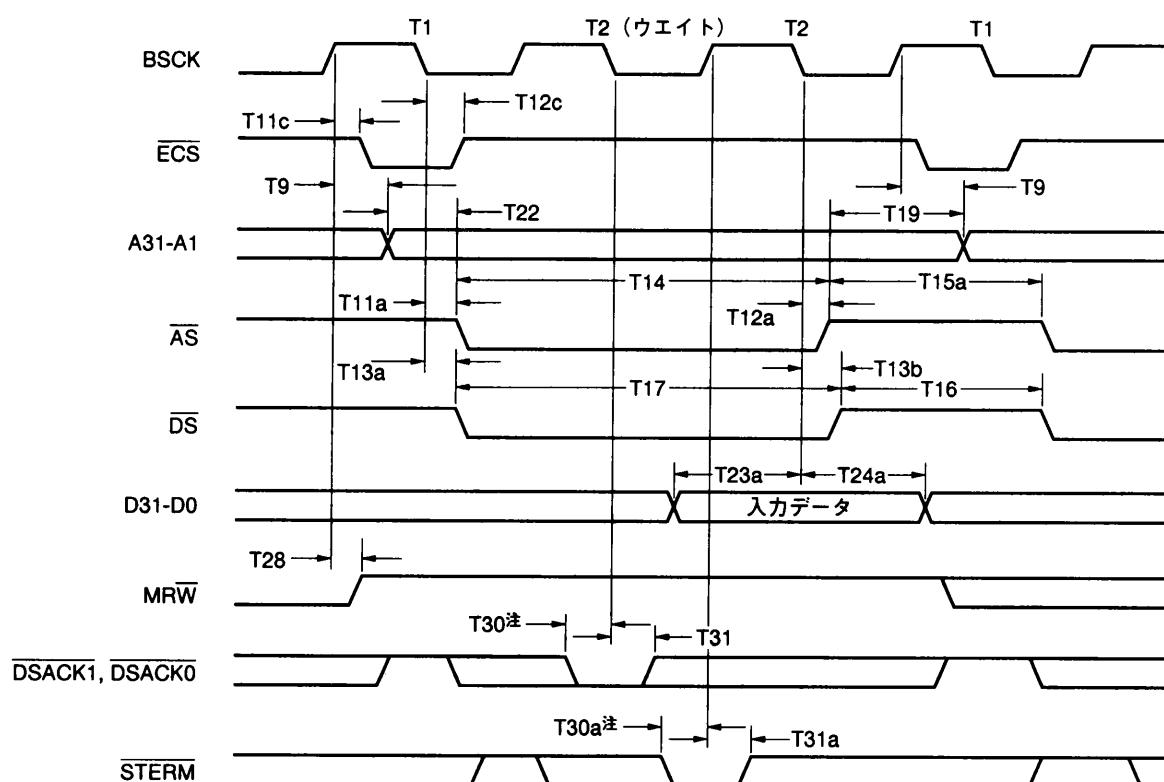
メモリ・ライト, BMODE = 1, 非同期モード



注1. RDAおよびTDA操作時、最後のライト・サイクルと次のリード・サイクルの間にアイドル・クロック・サイクル (Ti) が1つ挿入されます。データ・バスは、アイドル・サイクルの次のクロックの立ち上がりエッジでハイ・インピーダンスになることに注意してください (データ・フロート時間 (対BSCCK↑) についてはT52を参照してください)。

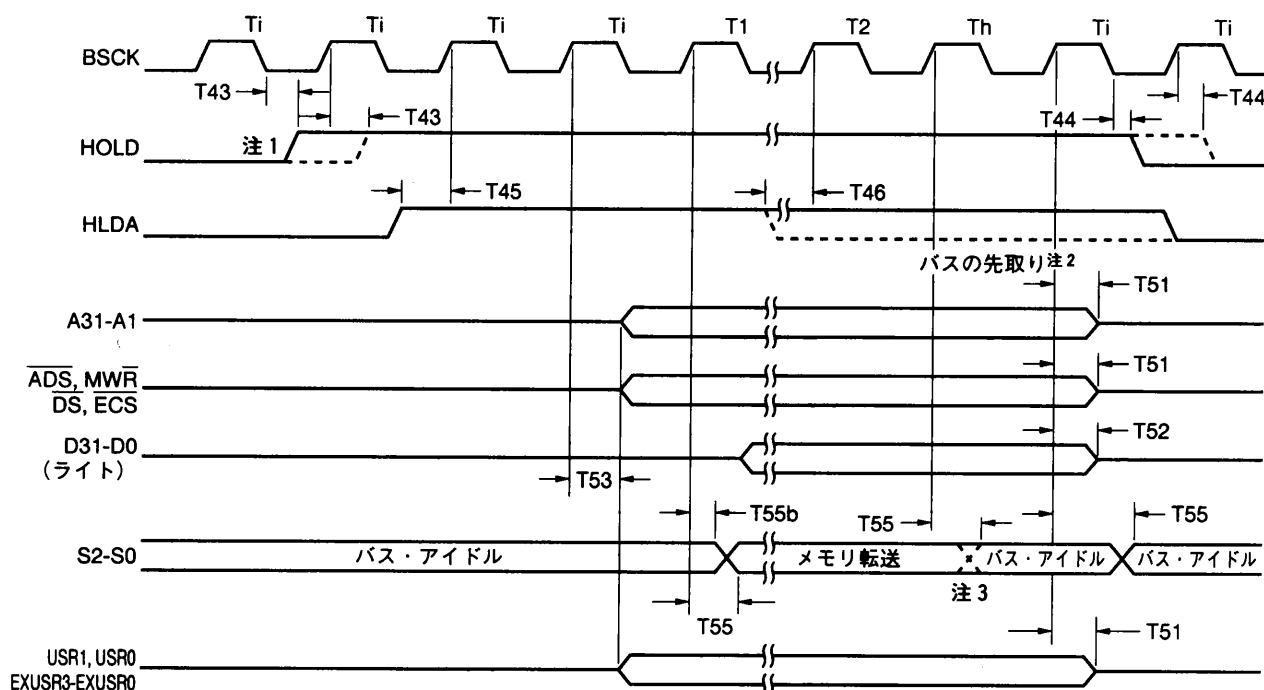
2. $\overline{\text{DSACK1}}$, $\overline{\text{DSACK0}}$ と $\overline{\text{STERM}}$ のセットアップ時間を合わせると、 $\overline{\text{DSACK1}}$, $\overline{\text{DSACK0}}$ をサンプルした1.5バス・クロック後、または $\overline{\text{STERM}}$ をサンプルした1サイクル後にμPD72932はメモリ・サイクルを終了することができます。 $\overline{\text{DSACK1}}$, $\overline{\text{DSACK0}}$ または $\overline{\text{STERM}}$ をサンプルしてロウ・レベルになるまで、T2ステートを繰り返します。 $\overline{\text{DSACK1}}$, $\overline{\text{DSACK0}}$ または $\overline{\text{STERM}}$ をそれぞれT1または一度目のT2ステート中にサンプルしてロウ・レベルであると、μPD72932は合計2バス・クロック (3バス・クロックでなく) でカレント・アクセスを終了します (プログラム可能なウェイト・ステートを0にセットしている場合)。 $\overline{\text{DSACK1}}$, $\overline{\text{DSACK0}}$ は非同期でサンプルされ、 $\overline{\text{STERM}}$ は同期でサンプルされます。

メモリ・リード, BMODE = 1, 非同期モード



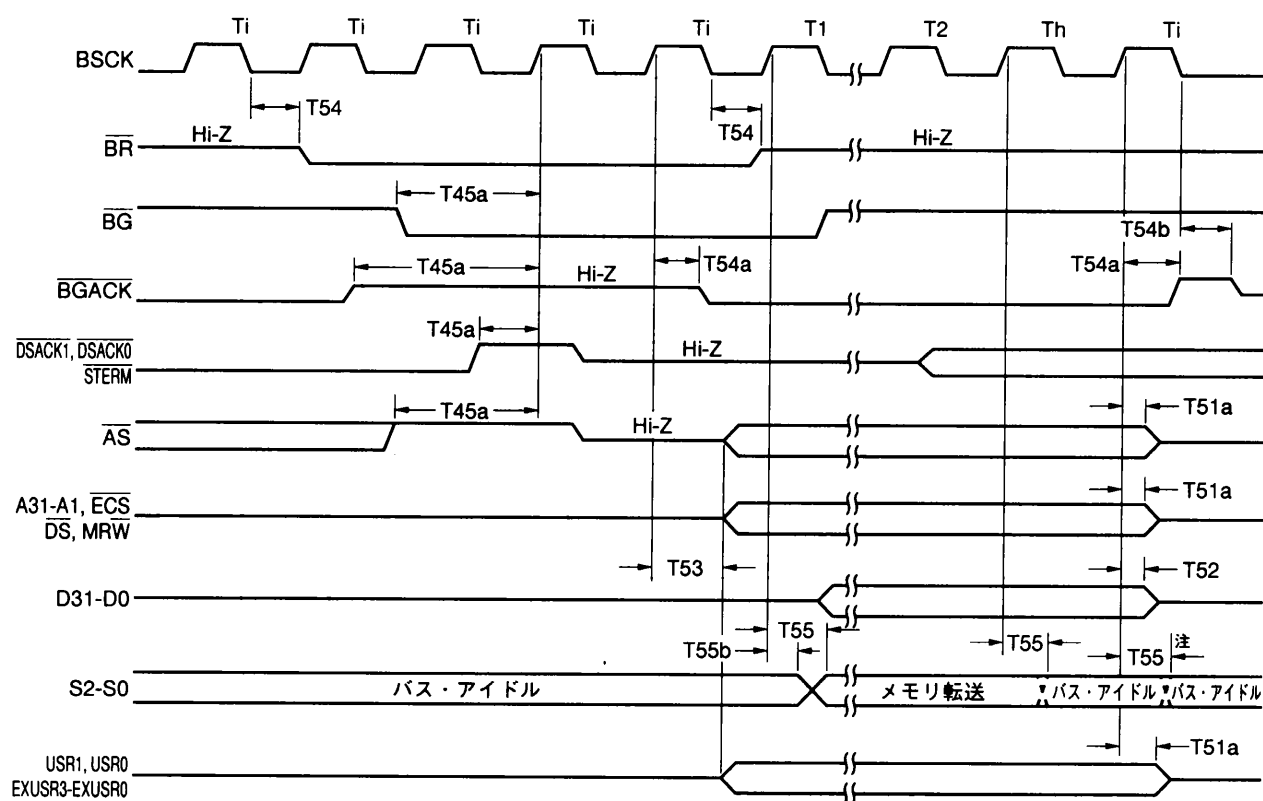
注 DSACK1, DSACK0とSTERMのセットアップ時間を合わせると、DSACK1, DSACK0をサンプルした1.5バス・クロック後、またはSTERMをサンプルした1サイクル後にμPD72932はメモリ・サイクルを終了することができます。DSACK1, DSACK0またはSTERMをサンプルしてロウ・レベルになるまで、T2ステートを繰り返します。DSACK1, DSACK0またはSTERMをそれぞれT1または一度目のT2ステート中にサンプルしてロウ・レベルであると、μPD72932は合計2バス・クロック（3バス・クロックでなく）でカレント・アクセスを終了します（プログラム可能なウェイト・ステートを0にセットしている場合）。DSACK1, DSACK0は非同期でサンプルされ、STERMは同期でサンプルされます。

バス要求タイミング, BMODE = 0



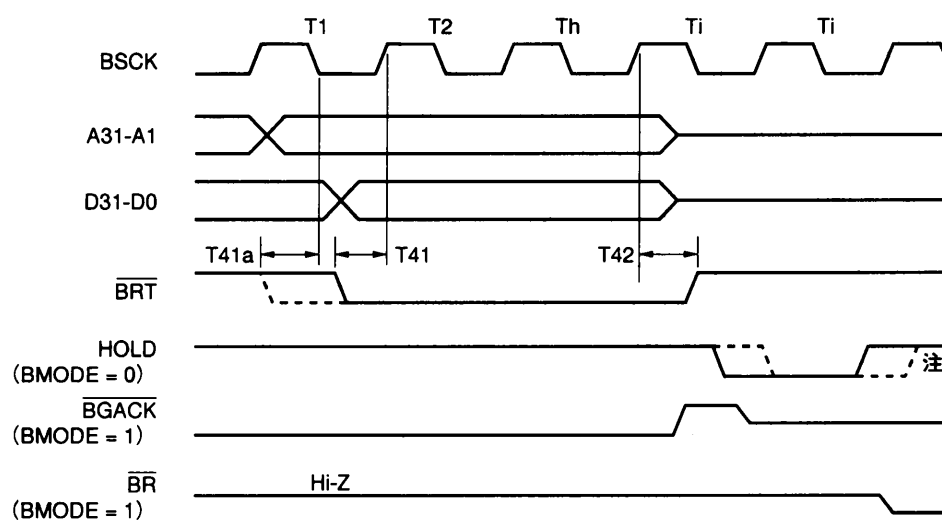
- 注1. HOLDの立ち上がりおよび立ち下りのタイミングはDCR2のPHビットによって決まります。タイミング・チャートではデフォルト時 (DCR2 : PH = 0) の状態を実線で示しています。T43およびT44の規格はどちらの場合にも適用できます。また、μPD72932がバスを要求しているときにHLDAをハイ・レベルにすると、先にHLDAがロウ・レベルになるまではHOLDはハイ・レベルになりません。
2. カレント・アクセスの最後のT2の立ち上がりエッジよりもT46の期間以上前にHLDAをロウ・レベルにすると、μPD72932によるバス転送よりも優先してバスを占有することができます。
3. S2-S0は、最後の動作がリード動作であればT2の終わりでアイドル状態を示します。最後の動作がライト動作であればThの終わりでアイドル状態を示します。

バス要求タイミング, BMODE = 1



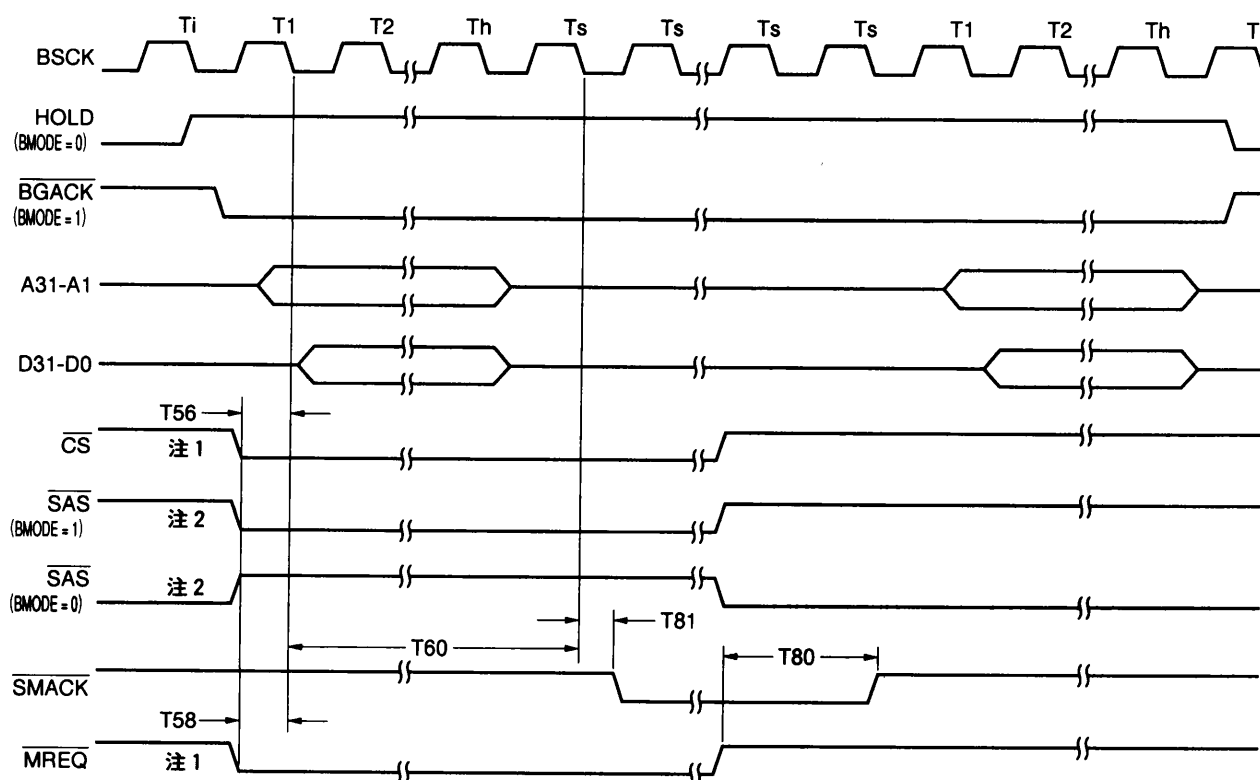
注 S2-S0は、最後の動作がリード動作であればT₂の終わりでアイドル状態を示します。最後の動作がライト動作であればT_hの終わりでアイドル状態を示します。

バス・リトライ

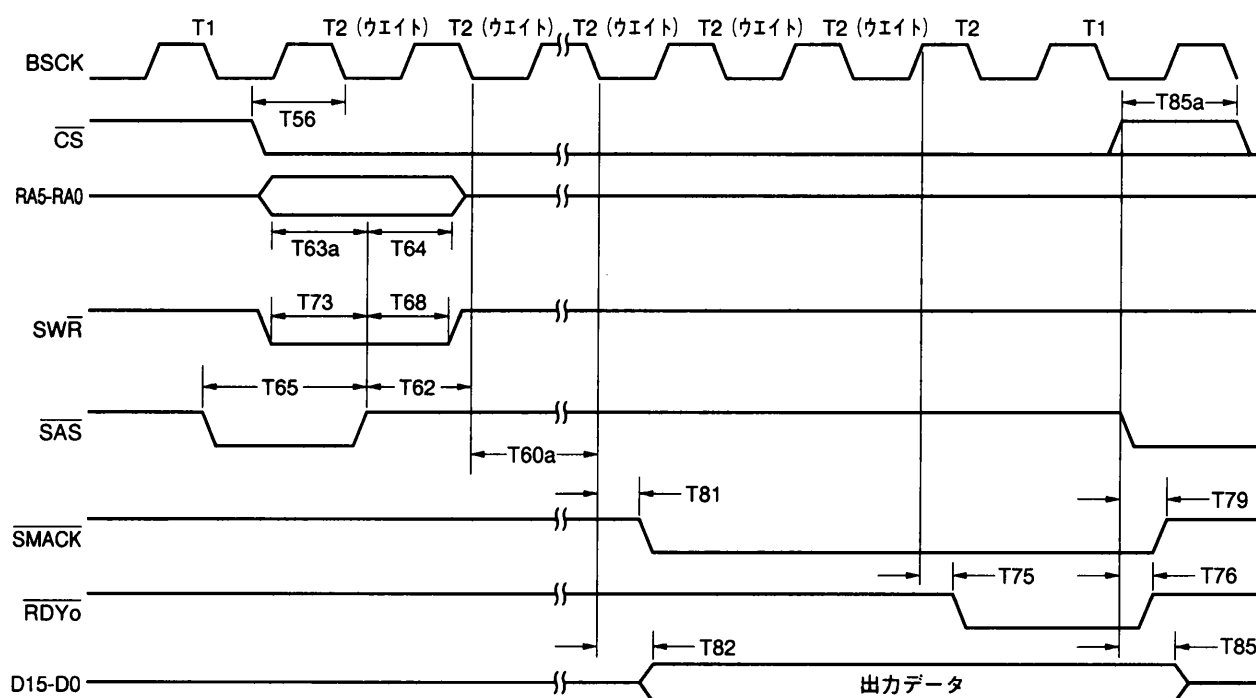


注 μPD72932はモードにより、HOLDをBSCKの立ち上がりエッジでハイ／ロウ・レベルにする場合と、立ち下がりエッジでハイ／ロウ・レベルにする場合があります。

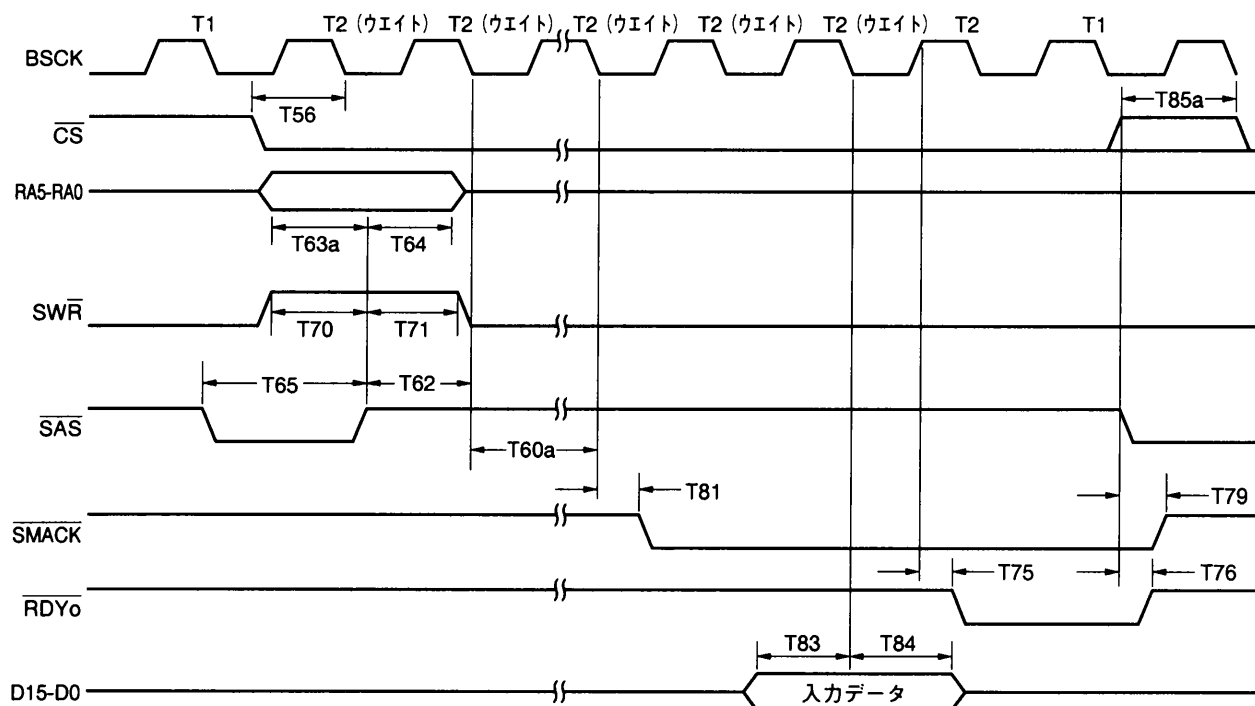
メモリ調停/スレーブ・アクセス



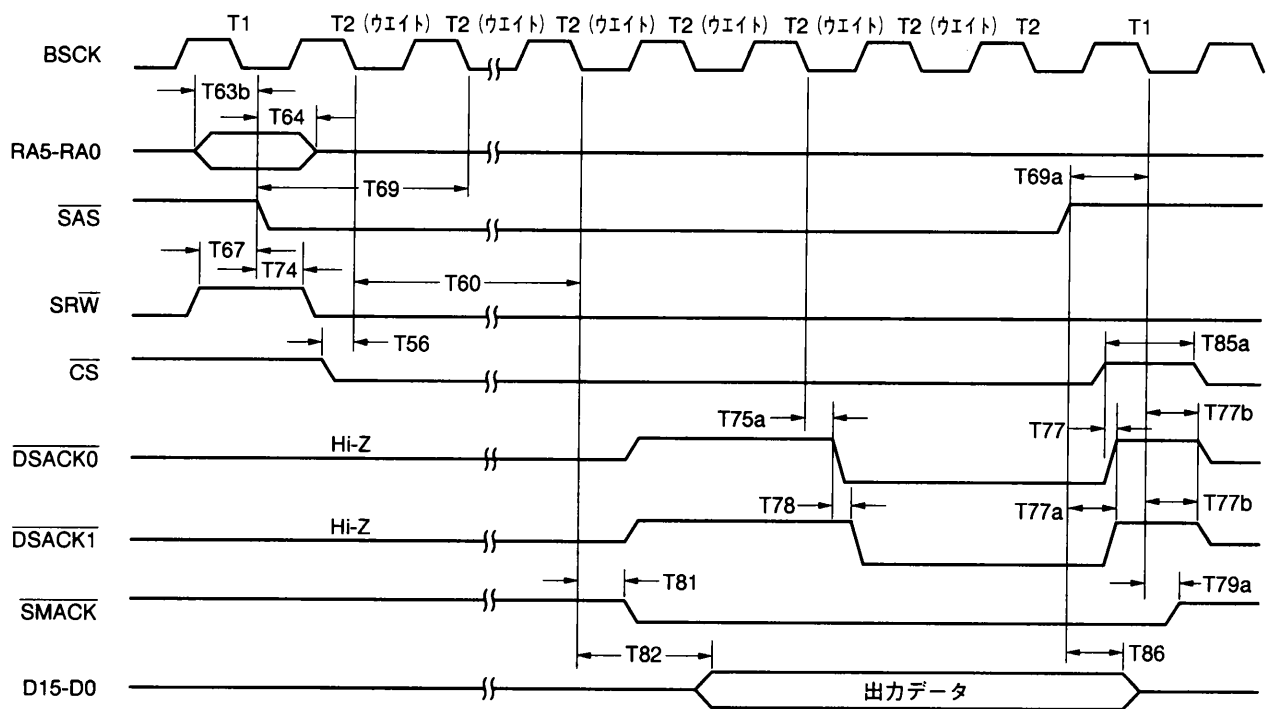
- 注1. $\overline{CS}$ と $\overline{MREQ}$ を同時にロウ・レベルにしないでください。これらの信号を連続してロウ・レベルにする場合、あとからロウ・レベルにする信号は、先にロウ・レベルにした信号の立ち上がりエッジから2バス・クロック以上の間ハイ・レベルに保持したうえでロウ・レベルにしてください。
2. $\overline{SMACK}$ がロウ・レベルになる動作は $\overline{CS}$ による場合と $\overline{MREQ}$ による場合とで異なります。 $\overline{MREQ}$ をロウ・レベルにした場合はその直接の結果として $\overline{SMACK}$ がロウ・レベルになりますが、 $\overline{CS}$ の場合は $\overline{SMACK}$ がロウ・レベルになるより前に $\overline{SAS}$ もロウ・レベル (BMODE = 1 の場合) またはハイ・レベル (BMODE = 0 の場合) にしておかなければなりません。つまり、 $\overline{MREQ}$ によって $\overline{SMACK}$ がロウ・レベルになった場合は、 $\overline{MREQ}$ がハイ・レベルになるまでは $\overline{SMACK}$ はロウ・レベルを保持します。 $\overline{SMACK}$ をハイ・レベルにしなくても、共用メモリへの複数のメモリ・アクセスが可能です。しかし $\overline{CS}$ によって $\overline{SMACK}$ がロウ・レベルになった場合は、 $\overline{SMACK}$ がロウ・レベルを保持するのは $\overline{SAS}$ がロウ・レベル (BMODE = 1 の場合) またはハイ・レベル (BMODE = 0 の場合) である間に限られます。 $\overline{SAS}$ はレジスタ・アクセスを行うたびに切り替えなければならないので、μPD72932に複数のレジスタ・アクセスを行っている間ずっと $\overline{SMACK}$ をロウ・レベルに保持しておくことはできません。この点が共有メモリ・デザインを設計する場合に考慮すべき重要な違いです。

レジスタ・リード, BMODE = 0^注

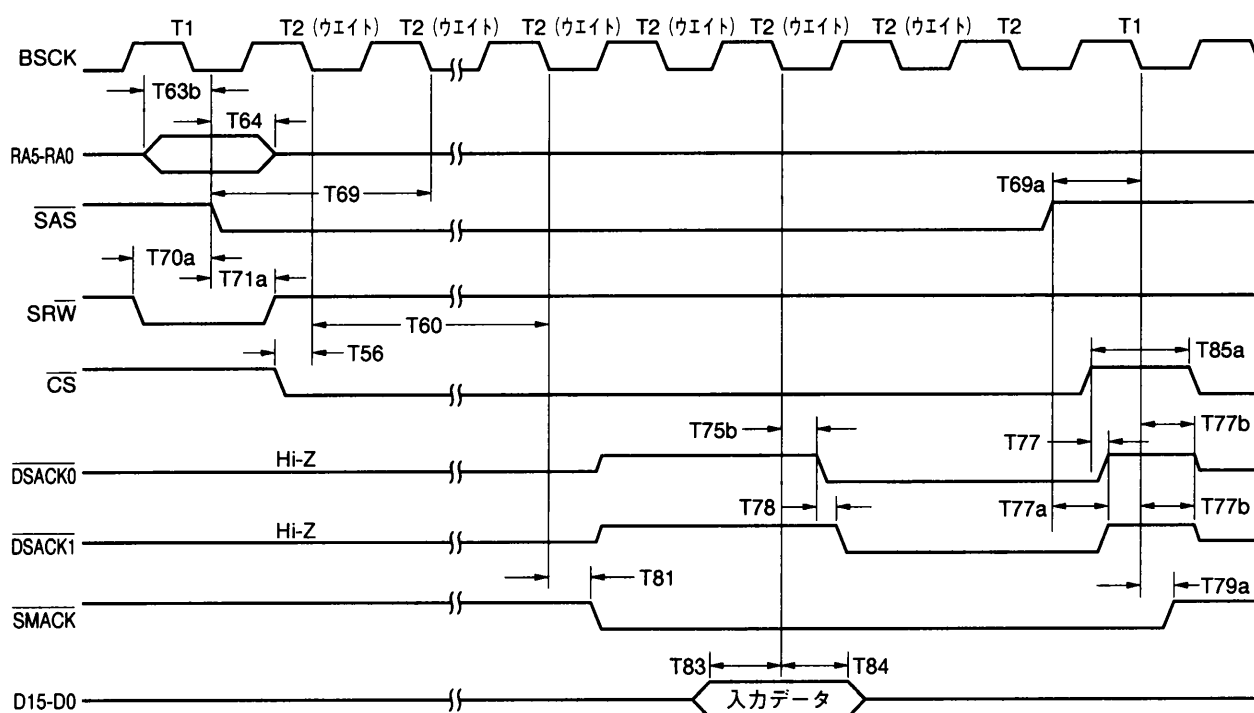
注 この図は、μPD72932へのスレーブ・アクセスを示しています。BSCCKステート (T1, T2など) はスレーブ・アクセスにおける等価のプロセッサ・ステートです。

レジスタ・ライト, BMODE = 0^注

注 この図は、μPD72932へのスレーブ・アクセスを示しています。BSCCKステート (T1, T2など) はスレーブ・アクセスにおける等価のプロセッサ・ステートです。

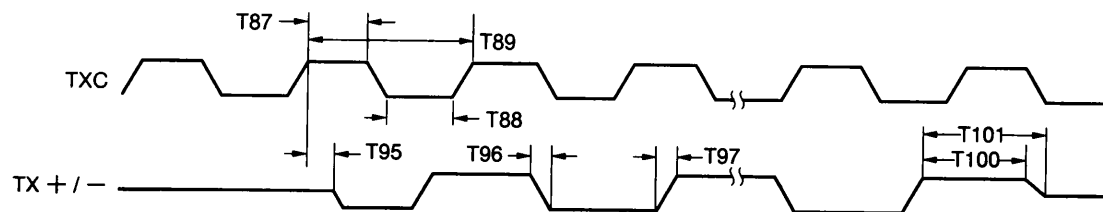
レジスタ・リード, BMODE = 1^注

注 この図は、 μ PD72932がアイドル状態時、正確にはマスタ・モードでない場合の μ PD72932へのスレーブ・アクセスを示しています。 μ PD72932がバス・マスタのときはメモリ調停／スレーブ・アクセスの図で示すように多少異なります。BSCCKステート (T1, T2など) はスレーブ・アクセスにおける等価のプロセッサ・ステートです。

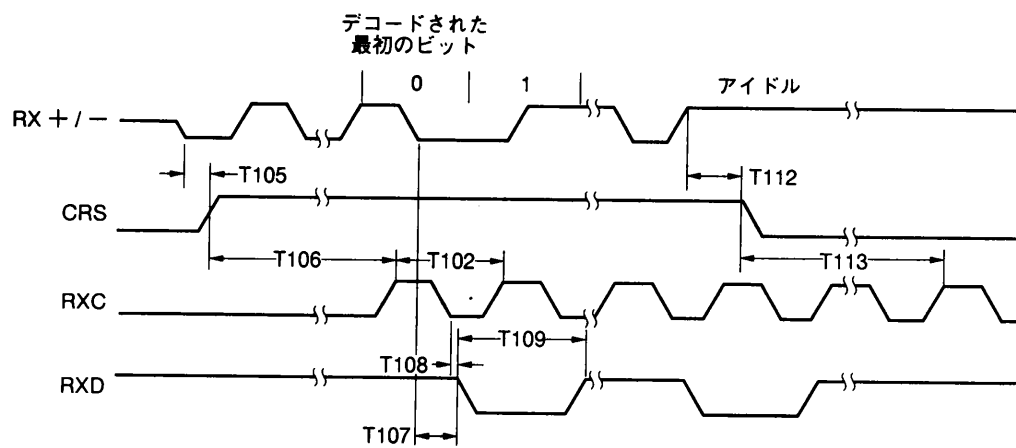
レジスタ・ライト, BMODE = 1^注

注 この図は、μPD72932がアイドル状態時、正確にはマスタ・モードでない場合のμPD72932へのスレーブ・アクセスを示しています。μPD72932がバス・マスタのときはメモリ調停／スレーブ・アクセスの図で示すように多少異なります。BSCCKステート (T1, T2など) はスレーブ・アクセスにおける等価のプロセッサ・ステートです。

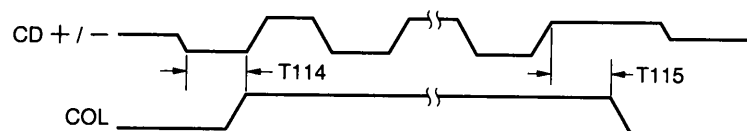
ENDECの送信タイミング (内部ENDECモード)



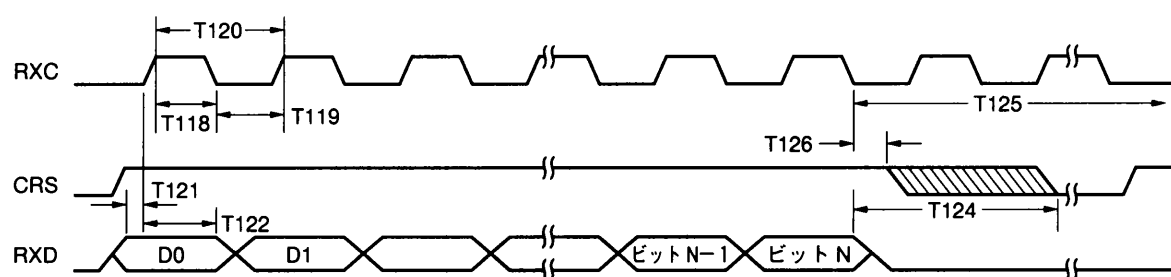
ENDECの受信タイミング (内部ENDECモード)



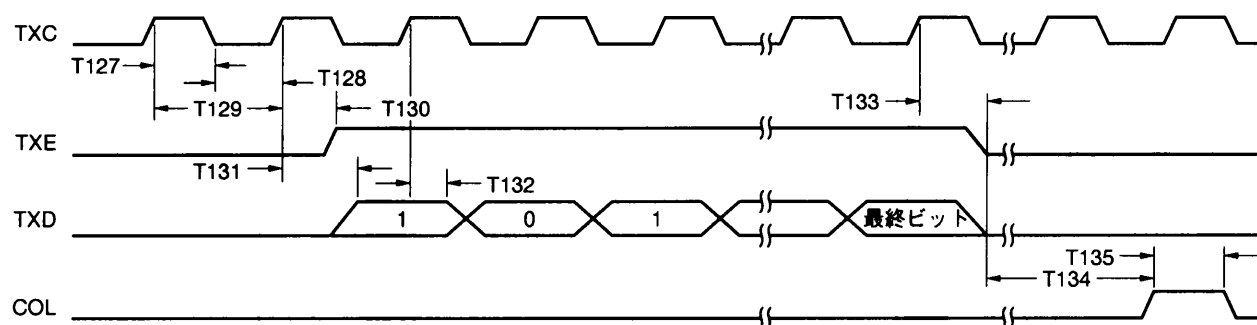
ENDECのコリジョン・タイミング



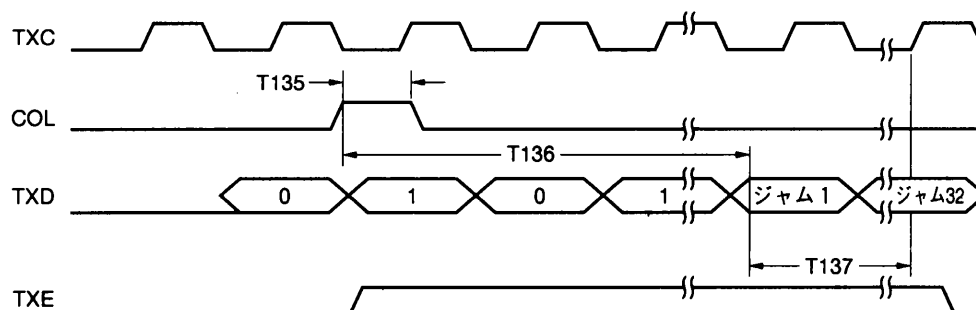
ENDEC - MAC受信シリアル・ タイミング (外部ENDECモード)



ENDEC - MAC送信シリアル・ タイミング (衝突なし)

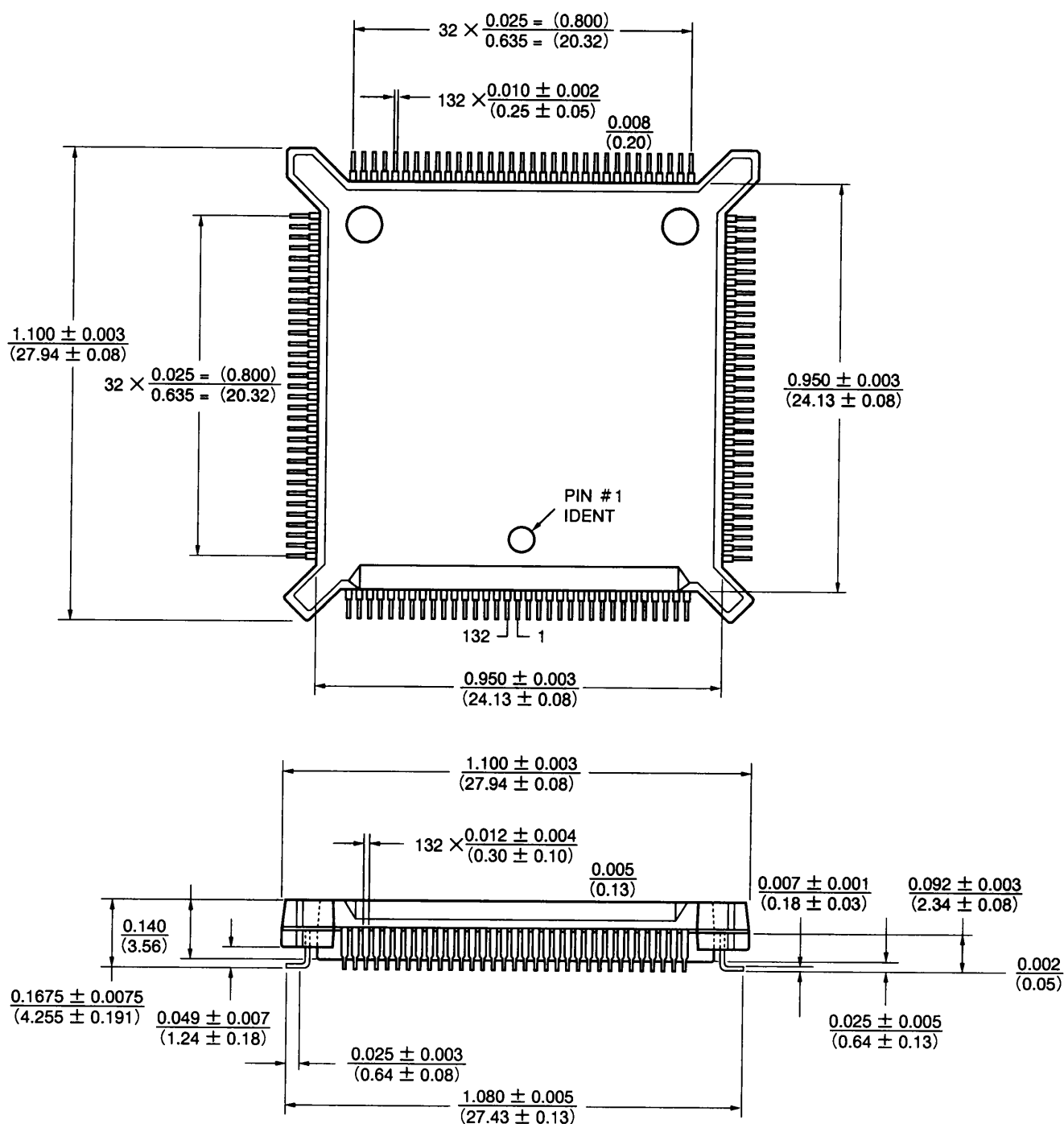


ENDEC - MAC送信シリアル・ タイミング (衝突)



10. 外形図

132ピン・プラスチックQFP外形図 (単位 上段: in., 下段: mm)



11. 半田付け推奨条件

μ PD72932の半田付け条件については、当社販売員にお問い合わせください。

表面実装タイプ

μ PD72932CVF : 132ピン・プラスチック QFP

付録 A. 使用上の注意事項

ここで示す情報の多くは、Bバージョンでは未記載となっていた情報やCバージョンで新たに追加された情報です。このデータ・シートの該当する箇所にはこれらの内容が反映されています。

A.1 プログラマブル割り込み (Bバージョン, Cバージョンともに適用)

送信リストを作成して複数のパケットを送信する場合、プログラムによって、送信ディスクリプタ・エリア(TDA)内に記述された TXpkt.config フィールドを μPD72932が読み出したときに割り込みを発生させることができます。割り込みを発生させるには、TXpkt.config フィールドのプログラマブル割り込み (PINTR) ビットを1にセットし、割り込みマスク・レジスタ (IMR) のプログラマブル割り込みイネーブル (PINTEN) ビットを1にセットします。PINTR は連続するパケットに対してセットすることはできません。1つおきの TDA の TXpkt.config フィールドで PINTR をセット (PINTR を交互にセット, リセットするように設定) することにより、データ・シートに記述されているとおりの動作をします。

A.2 RDA オーバフロー (Bバージョン, Cバージョンともに適用)

μPD72932は受信の最後で受信ディスクリプタ・エリア (RDA) にステータスを書き込み、RDA の RXpkt.link (リンク・フィールド) の EOL ビットを読み出します。このビットが1の場合は、使用可能な RDA がそれ以上ないことを意味します。μPD72932は、新しいパケットが到着し受信データ数が受信 FIFO のスレッシュホールド値を越えるまで、RDA のリンク・フィールドの再読み出しを行いません。新しいパケットが到着し受信データ数が受信 FIFO のスレッシュホールド値を越えると、μPD72932は再び RDA のリンク・フィールドを読み出します。EOL がまだ1の場合、μPD72932はパケットを拒否し、同様に次のパケットが到着して受信 FIFO が再びスレッシュホールド値を越えるまで EOL ビットの読み出しを行いません。

16ビット・モードではごくまれに、新しいパケットが到着するのを待たずに μPD72932が RDA のリンク・フィールドの再読み出しを行うことがあります。この場合、μPD72932は RDA のリンク・フィールドを継続的に読み出します。μPD72932がバス上で最も高いバス・アービトレーション優先順位を与えられているシステムでは、この継続的な読み出しによって CPU がメモリにアクセスできなくなることがあります。すなわち、CPU が RDA の EOL ビットを更新できないため、μPD72932は常にこのビットを1として読み出すことになり、システムがロック・アップしてしまいます。このような状態は、次に示す3つの条件が同時に成立する場合にのみ発生します。

- ① μPD72932が16ビット・モードであること
- ② 受信FIFO を空にしたとき、さらに受信すべきパケット・データが23から30バイト残っていること
- ③ 受信FIFO からメモリに②のデータが DMA 転送されない状態 (ホストがバスを占有している状態) が長時間^注続くこと

注 問題の発生するホストのバス占有時間は、FIFO スレッシュホールドの設定によって異なります。

- 12ワードの FIFO スレッシュホールド… 6.4 μs ((32-24)×800 ns) 以上の場合に発生
- 2ワードの FIFO スレッシュホールド… 22.4 μs ((32-4)×800 ns) 以上の場合に発生

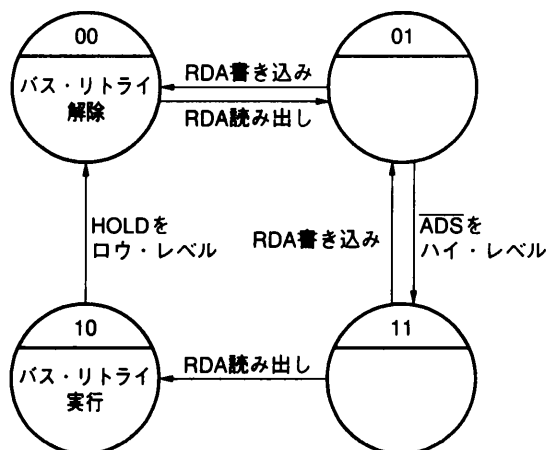
この問題の解決策として、次のいずれかを実施してください。

- (1) RDA が不足しないよう、十分な数の RDA をデバイス・ドライバが割り当てるようにする。
- (2) FIFO スレッシュホールド値を減らす。2ワードの FIFO スレッシュホールドの場合は、ホストによるバスの占有時間が22.4 μs を越える場合のみ問題が発生するので、ホストのバス占有時間をこれより短くする。

- (3) μPD72932のアービトレーション優先順位を CPU より低いレベルにして, CPU が新しい RDA を用意し, RDA の EOL ビットを更新できるようにする。
- (4) μPD72932が RDA の EOL ビットを続けて 2 回読み出したあとは, EOL ビットの読み出し要求を再びバスに出さないように禁止する外部ロジックを使用する。

図 A-1 に非同期ステート・マシンの図を示します。これは, μPD72932に対してラッチト・バス・リトライ (DCR の LBR ビットを 1 にセット) をアクティブとするロジックを実行するためのもので, μPD72932がバスを独占することを防ぎます。ラッチト・バス・リトライにより μPD72932はバスを解放します。ソフトウェアにより割り込みステータス・レジスタ (ISR) の BR ビットをリセットしてラッチト・バス・リトライが解除されると, μPD72932 は再びバスにアクセスします。ソフトウェアによるバス・リトライの解除は, 新しい有効な RDA を生成して前の RDA の EOL ビットをクリアしたあとで行ってください。

図A-1 ラッチト・バス・リトライ EOLステート・マシン



A.3 バス・グラント後の $\overline{DSACK1}$, $\overline{DSACK0}$ の状態 (Bバージョン, Cバージョンともに適用)

BMODE=1 (モトローラ・モード) では, バス・マスタが $\overline{MREQ}$ をロウ・レベルにして μPD72932にバス解放を要求した場合, μPD72932はバスを解放したあとも $\overline{DSACK1}$, $\overline{DSACK0}$ をドライブし続けます。このため, 3 ステート・バッファを使用するなどして $\overline{DSACK1}$, $\overline{DSACK0}$ をバスから切り離す必要があります。

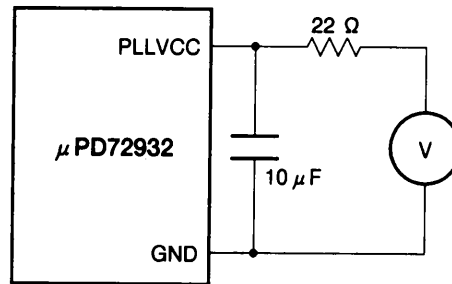
6.3.8 オンチップ・メモリ・アービタを参照してください。

A.4 PLLVCC 端子についての考慮 (Bバージョン, Cバージョンともに適用)

PLLVCC 端子は, 内部 ENDEC ユニットのフェーズ・ロック・ループ (PLL) 用の +5 V 電源です。この PLL 回路はアナログ回路のため, PLLVCC 端子に過度のノイズが加わると PLL の性能に悪影響を与えます。また, 10 Hz - 400 Hz の範囲のノイズが加わると, ENDEC のジッタ性能が低下し, パケット喪失や CRC エラーが発生します。電源ノイズによる パケット受信エラーが深刻な場合は, ノイズを除去してジッタ性能を改善するために, ロウ・パス・フィルタとしてデジタル電源に対して図 A-2 に示すような 1 極 RC フィルタ (遮断周波数が 1 kHz に設計されたもの) を付加することも 1 つの方法です。この場合, PLLVCC 端子に流れる電流は 3 mA-4 mA となり, 抵抗に加わる電圧が 90 mV 以下となるため, PLL の動作は影響を受けません。

7.5 電源の考慮事項を参照してください。

図A-2 電源ノイズのフィルタリング



A.5 レディ出力 ($\overline{\text{RDY}}_0$) の状態 (Bバージョン, Cバージョンともに適用)

レディ出力 ($\overline{\text{RDY}}_0$) は Bバージョンのデータ・シートでは 3 ステートになると記載していましたが、この端子は出力のみでハイ・インピーダンスにはなりません。したがって、 $\text{BMODE}=0$ で使用する場合、 $\overline{\text{RDY}}_0$ に絶縁用の 3 ステート・バッファが必要な場合があります。

1.2 バス・インタフェース端子を参照してください。

A.6 ラッチト・レディ・モード (Bバージョン, Cバージョンともに適用)

ラッチト・レディ・モードは、Bバージョン, Cバージョンともにサポートしていません。このデータ・シートではラッチト・レディ・モードに関する記述を削除しました。

A.7 バイト・カウントのミスマッチ (BCM) ビットの状態 (Bバージョン, Cバージョンともに適用)

バイト・カウントのミスマッチ (BCM) ビット (送信制御レジスタのビット 1) は、転送中に過剰コリジョンが発生して EXC ビット (送信制御レジスタのビット 6) がセットされるたびにセットされます。

5.3.4 送信制御レジスタの BCM ビットの説明を参照してください。

A.8 $\overline{\text{SAS}}$ ハイ・セットアップ時間の規格 (Bバージョン, Cバージョンともに適用)

$\text{BMODE}=1$ の場合、スレーブ・サイクルを終了するために $\overline{\text{SAS}}$ をハイ・レベルにする必要があります。このタイミングは Bバージョンにも存在していましたが、Cバージョンでは T69a として新たに定義しています。9. 電気的特性 AC 特性を参照してください。これは、バス・クロックの立ち下がりエッジに対する $\overline{\text{SAS}}$ セットアップ時間です。このタイミングは Bバージョンでは 15 ns (MIN.) となっていたましたが、Cバージョンではより短いセットアップ時間で済むように改善しました。

A.9 ループバック・モード時の PMB ビットの状態 (Bバージョン, Cバージョンともに適用)

モニタしたパケットの不良 (PMB) ビット (送信制御レジスタのビット 3) は、3 種類のループバック・モードのいずれかがセットされた場合、常に 0 になります。

5.3.4 送信制御レジスタの PMB ビットの説明を参照してください。

A.10 CMOS 負荷テストの削除 (Bバージョン, Cバージョンともに適用)

現在のほとんどのデジタル IC 部品は TTL コンパチブルなため、μPD72932 では CMOS コンパチビリティに関するテストは行っていません。

A.11 RDA の不正データ書き込み (B バージョンにのみ適用)

これは B バージョンで存在した動作であり、C バージョンではこの問題は修正されています。

詳細は B.1 RDA の不正データ書き込みを参照してください。

A.12 TXP のスタック (B バージョンにのみ適用)

これは B バージョンで存在した動作であり、C バージョンではこの問題は修正されています。

詳細は B.2 TXP のスタックを参照してください。

A.13 RXEN および RXDIS のセット、リセット (B バージョン、C バージョンともに適用)

コマンド・レジスタ (CR) の RXEN ビットが "1" で、かつパケット受信中に受信動作停止コマンド (CR の RXEN ビット = "0", RXDIS ビット = "1") を発行すると、パケット受信が終了するまで RXEN ビットと RXDIS ビットの両方がセットされた状態になります (B.3.1 コマンド・レジスタ参照)。しかし、受信動作停止コマンド発行のあと、パケット受信が終わっていないうちに別のコマンドを発行しようとして不用意に RXEN ビットに "0" を書き込むと、RXEN ビットが強制的にクリアされてしまい、受信中のパケットが消失してしまいます。したがって、受信停止コマンドを発行した直後に別のコマンドを発行する場合、RXEN ビットを強制的にクリアしないようにするため、CR を読み出して RXEN ビットが "0" になるのを待ってから次のコマンドを発行してください。

受信動作を再開するときにも同じことがいえます。つまり、受信動作再開コマンド (CR の RXEN ビット = "1", RXDIS ビット = "0") を発行した直後に別のコマンドを発行する場合、不用意に RXDIS ビットに "0" を書き込まないようにしてください。そのためには、受信動作再開コマンド発行のあと CR を読み出して RXDIS ビットが "0" になるのを待ってから次のコマンドを発行してください。

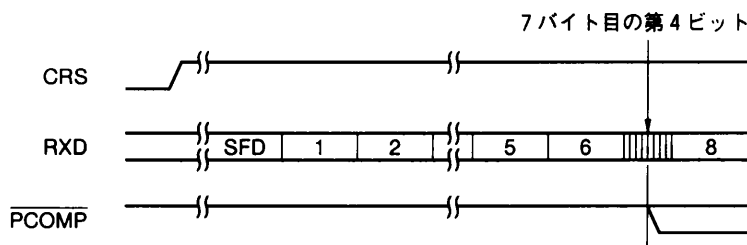
A.14 最低バス・クロック速度 (B バージョン、C バージョンともに適用)

μPD72932の最低クロック速度に関するテストは行っていませんが、8.6 MHz 以下では ISR の TXER ビットが誤ってセットされることが判明しています。

A.15 PCOMP タイミング (B バージョン、C バージョンともに適用)

PCOMP は、データ・コンフィギュレーション・レジスタ 2(DCR2)のビット 1 がセットされているときにはビット 1 の、ビット 2 がセットされているときにはビット 2 の条件を満たすとロウ・レベルになります。ロウ・レベルになるタイミングは到着したパケットの 7 バイト目の第 4 ビットの直後です。

図A-3 PCOMPタイミング



CRS がロウ・レベルになり、1 送信クロック (TXC) 後に PCOMP はハイ・レベルに戻ります。

1.1 ネットワーク・インタフェース端子の PCOMP の説明を参照してください。

A.16 VCCL 端子, GNDL 端子 (B バージョン, C バージョンともに適用)

B バージョンのデータ・シートでは、VCCL 端子およびGNDL 端子は ENDEC 電源端子およびグランド端子の 1 つとなっていたましたが、本来これらの端子はデジタル電源およびグランド端子です。

1.5 電源およびグランド端子を参照してください。

A.17 $\overline{DSACK1}$ および $\overline{DSACK0}$ (B バージョン, C バージョンともに適用)

μPD72932がスレーブ・モードのとき (BMODE=1), 32ビット・モードであっても16ビット・モードであっても, μPD72932はサイクルを終了させるために $\overline{DSACK1}$ と $\overline{DSACK0}$ の両方からロウ・レベルを出力します。

μPD72932が32ビット・マスタ・モードのとき (BMODE=1), サイクルを終了させるには $\overline{DSACK1}$ と $\overline{DSACK0}$ の両方にロウ・レベルを入力しなければなりませんが, 16ビット・マスタ・モードのときは, サイクルを終了させるのに $\overline{DSACK1}$ のみにロウ・レベルを入力するだけでよく, $\overline{DSACK0}$ にロウ・レベルを入力する必要はありません。

A.18 ホスト CPU による μPD72932DMA 動作の強制中断 (B バージョン, C バージョンともに適用)

μPD72932がLCAM コマンドによって CAM エリアをアクセスしているときに CPU が HLDA をロウ・レベルにすると, μPD72932からバス使用权を強制的に取ることができます。このとき μPD72932はバスから切り離され, バス・ステータス (S2-S0) がアイドル状態を示しますが, HOLD はロウ・レベルになりません。

μPD72932がそのほかのディスクリプタ・エリア (RRA, RDA, TDA) をアクセスしているときに HLDA がロウ・レベルになると, μPD72932は正常にバスから切り離され, HOLD をロウ・レベルにします。HOLD はその 1 バス・クロック後に再びハイ・レベルになります。

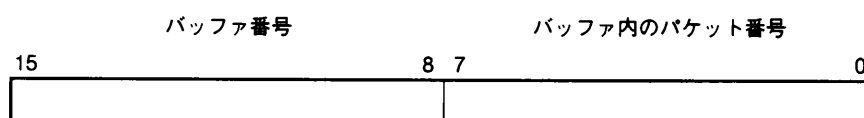
μPD72932がバッファ・エリア (RBA, TBA) をアクセスしているときに HLDA をロウ・レベルにすると, μPD72932は正常にバスから切り離され, HOLD をロウ・レベルにします。しかし, 再びバッファ・エリアにアクセスする際, FIFO のスレッシュホールドの状態によっては HOLD が再びハイ・レベルにならないこともあります。

6.3.1 バスの獲得を参照してください。

A.19 RBAE の処理 (B バージョン, C バージョンともに適用)

μPD72932で RBAE を処理する方法を次に説明します。通常, μPD72932はエラー・パケット (CRC エラー, フレーム配列エラー, ラント・パケットなど) を受け取ると, その該当パケットを受信するように設定されていないかぎり, パケットを廃棄し, バッファ・ポインタをそのパケットの受信前の位置に戻します。しかし, バッファの長さを越えるパケット (RBAE) の場合は, μPD72932はバッファ・ポインタを戻さず, バッファをそのままの状態にして新しいパケットを受信します。この動作自体は問題ではありませんが, 問題は μPD72932がこの RBAE に対して RDA を作成しないということです。つまり, 対応する RDA を持たないバッファ (1 つのバッファに 1 つのパケットを格納する場合) やバッファの一部 (1 つのバッファに複数のパケットを格納する場合) が生じてしまいます。このようなバッファは見つけ出して割り当て解除するか RRA に戻して, μPD72932が再び使用できるようにしなければなりません。

失われたバッファを発見する方法としては, RDA のシーケンス番号を使用する方法があります。RDA のシーケンス番号フィールドは16ビット長のフィールドで, 次に示すように 2 つのカウンタで構成されています。



上位 8 ビットはバッファ番号のカウンタ値を表し、下位 8 ビットはそのバッファ内のパケット番号のカウンタ値を表します。カウンタはそれぞれ 0 から 255 までカウントして 0 に戻ります。したがって、シーケンス番号は次のように変化します。

```

0000H…バッファ 0 のパケット 0
0001H…バッファ 0 のパケット 1
0002H…バッファ 0 のパケット 2
0100H…バッファ 1 のパケット 0
0101H…バッファ 1 のパケット 1
0102H…バッファ 1 のパケット 2
0103H…バッファ 1 のパケット 3
↓
0200H…バッファ 2 のパケット 0

```

ソフトウェア・ドライバは、バッファ番号がたとえば 0 から 1 へ変化したことを確認したとき、バッファ 0 が完全に処理されたと認識し、そのバッファの割り当てを解除することができます。バッファ 0 のパケット 3 で RBAE が発生した場合、このバッファ・シーケンス番号は同じようにインクリメントされる (0 から 1 へ変化する) ため、問題は生じません。

1 つのバッファに複数のパケットをバッファリングするとき、RBAE について考慮しなければならないのは、RBAE パケットがバッファ内の最初で唯一のパケットである場合です。この場合、そのバッファを指し示す RDA は存在せず、シーケンス番号は次のようになります。

```

0000H…バッファ 0 のパケット 0
0001H…バッファ 0 のパケット 1
0002H…バッファ 0 のパケット 2
0200H…バッファ 2 のパケット 0
0201H…バッファ 2 のパケット 1
0202H…バッファ 2 のパケット 2
0203H…バッファ 2 のパケット 3
↓
0300H…バッファ 3 のパケット 0

```

この場合、バッファ・シーケンス番号が 0 から 2 へ飛んでいます。これは、RBAE がバッファ 1 のパケット 0 で発生した場合に生じます。RBAE バッファ (バッファ 1) を割り当て解除するには、2 つのを行ってください。まず、RBAE の発生を見つけてください。これはバッファ・シーケンス番号 2 から前のバッファ・シーケンス番号 0 を引いて、その結果の値が 1 より大きい ($2 - 0 = 2$) ことから導き出せます。このことは、バッファ 2 とバッファ 0 の間に RBAE の発生したバッファが存在し、そのバッファを指し示す RDA がないことを意味します。ソフトウェアでこの現象を確認したら、次に RBAE バッファ (バッファ 1) のアドレスまたはポインタを見つけて、バッファの割り当てを解除するか RRA に戻してください。

この方法は μPD72932 が 1 バッファあたり 1 パケットだけをバッファリングする場合にも容易に応用することができます。その場合、RBAE が発生するたびに対応する RDA がないバッファが生成されます。この場合も、シーケンス番号を引き算することによってそのバッファを見つけることができます。その後の手順は前述の場合と同じです。

RBAE 割り込みを使用すれば、チェックする RDA 1 つ 1 つについて引き算を行わなくても RBAE の発生がわかります。割り込みが発生したら、ただちに RBAE をクリアしてシーケンス番号のチェックが必要なことを知らせるフ

ラグをセットします。RBAE バッファが見つかったらフラグをクリアします。

RBAE を処理するとき考慮しなければならないもう 1 つの問題として、複数の RBAE バッファが連続する場合があります。複数の RBAE バッファが連続している場合、同じ手順を使用できますが、シーケンス番号を引き算するとき結果の値が 2 よりも大きくなります。また、RBAE バッファを見つける前に複数の RBAE 割り込みが発生する可能性が高いため、発生した RBAE 割り込みの数をカウントすることが必要になります。したがって、RBAE 割り込みが発生したらフラグを設定するのではなく、カウンタをインクリメントして割り込みをクリアします。そして、RBAE バッファが見つかったらカウンタをデクリメントし、カウンタ値が 0 になるまで繰り返します。

割り込みがカウントされる前に 2 つの RBAE 割り込みが発生する可能性がありますので注意してください。この場合、RBAE バッファを処理するルーチンは、連続する複数の RBAE バッファの発生をチェックする必要があり、さらに 1 つおきのバッファでの RBAE バッファの発生をもチェックする必要があります（どちらの場合も RBAE バッファの発生が 1 回としてカウントされた可能性があります）。ここで必要となるチェックの量は、割り込み処理ルーチンがどの程度頻繁に応答するか、ISR をどの程度速くポーリングするかによって決まります。

A.20 AC 特性値 (C バージョンにのみ適用)

C バージョンのデータ・シートでは新しいタイミング規格値を記載しています。

9. 電気的特性 AC 特性を参照してください。

A.21 1 つの RBA に複数のパケットを格納する場合 (C バージョンにのみ適用)

C バージョンへの移行において、受信メモリ・アーキテクチャの動作に機能的な変更がありました。この変更は、1 つの受信バッファ・エリア (RBA) に対して複数のパケットを格納するアプリケーション・システムにのみ影響します。これについては、4.4.4 (5) 1 つの RBA に複数のパケットを格納する場合を参照してください。

付録 B. B バージョンでの問題

ここで示す問題は B バージョンから C バージョンへの変更の際に改善されており、C バージョンでは存在しません。

B.1 RDA の不正データ書き込み

ネットワークの通信量が非常に多く、一定のバス待ち時間が生じている状況では、μPD72932が RDA の書き込みの際にステータス・フィールドおよびバイト・カウント・フィールドに無意味な情報を書き込むことがあります。その結果、不正報告のためにパケットが失われることもあります。RDA への不正データ書き込みには次に示す 2 つの場合があります。それぞれについて説明します。

(1) ゼロ長パケットの場合

この場合、μPD72932は 1 つのパケットをバッファに書き込みますが、このバッファを指し示す RDA を 2 つ書き込みます。最初の RDA には不正な値のステータス・フィールドおよびバイト・カウント・フィールドを書き込みます。2 つ目の RDA は正常です。不正 RDA フィールドには、RBA のパケットからのパケット・データを書き込みます。このため、パケットの先頭部分のデータが失われています。2 つ目の RDA にはバッファ内のパケットを指す実際の受信ディスクリプタを書き込みます。最初の RDA も RBA に書き込まれた「ゼロ長パケット」を指す RDA です。ゼロ長パケットとは、μPD72932がパケットのバッファ書き込みを試みても実際にパケット・データを書き込むことができなかったためにそう呼びます。

最初の RDA のステータス・フィールドが不正となっても、受信制御レジスタ (RCR) の RBA 内の最終パケット (LPKT) ビットは有効で、不正とはなりません。ソフトウェアが 1 つのバッファにつき 1 つのパケットを格納するようになっている場合、LPKT ビットは常にセットされます。この動作を利用して、不正となった RDA を見つけることができます。2 個の RDA が同じバッファを指しているため、バッファ内の最後のパケットを指すのは最初の RDA ではなく、2 つ目の RDA になります。したがって、LPKT ビットは最初の RDA では 0 になり、2 つ目の RDA では 1 になります。LPKT ビットが 0 に設定されている RDA は不正な RDA です。不正な RDA が見つかったら、ソフトウェアでその RDA と次の RDA を破棄し、両方の RDA が指していたバッファを解放してください。

ソフトウェアが 1 つの RBA あたり 1 つのパケットをバッファリングしていない場合は、(2) 2 つのバッファに 1 つのパケットを書き込む場合で説明する回避方法を使用してください。

(2) 2 つのバッファに 1 つのパケットを書き込む場合

この場合、μPD72932はパケットのバッファリングを正常に開始しますが、完了しないうちに中断し、RDA の書き込みに移ります。RDA には不正な値のステータス・フィールドおよびバイト・カウント・フィールドを書き込みます。これらのフィールドの不正なデータは、パケットそのものから書き込みます。1 つ目の RDA を書き込んだあと、μPD72932は同じパケットの残りの部分を新しい RBA にバッファリングします。そのあと、この新しいバッファに対応する 2 つ目の RDA を書き込みます。つまり、1 つのパケットを 2 つの RBA に分けて書き込むことになります。2 つの RBA にはそれぞれ対応する RDA がありますが、最初の RDA は不正となっています。しかし、2 つ目の RDA は正常で、ステータス・フィールドおよびバイト・カウント・フィールドはネットワークから受信したパケット全体の正しいステータスを示します。

2 つのバッファに 1 つのパケットを書き込むという問題は、16ビット・モードのアプリケーション・システムでエンプティ/フィル・モードを使用した場合にのみ発生します。ブロック・モードではこの問題は発生しません。

アプリケーション・システムが 16ビット・モードであり、かつブロック・モードを使用することができない

場合、回避方法として受信パケットについて次の3点をテストしてください。

- ① RDAのパケット長が有効かどうか調べます。大きすぎたり小さすぎたりする場合、そのパケットと次のパケットを破棄してください。
- ② RDAのステータス・フィールドが有効かどうか調べます。ドライバ・ソフトウェアはステータス・フィールドがとり得る値を認識していなければなりません。あるはずのないステータス・ワードがRDAに存在する場合、そのパケットと次のパケットを破棄してください。
- ③ μPD72932がCAMのデスティネーション・アドレスと一致するパケットだけを受け取るように設定されている場合、デスティネーション・アドレスを使用して不正となったパケットを調べることもできます。この方法は、2つ目のRBAではパケットの先頭部分が失われており、有効なデスティネーション・アドレスがないことを利用します。ソフトウェアは、パケット処理時に次のパケットを先に調べ、次のパケットのデスティネーション・アドレスとCAMのアドレスとを比較します。それらが一致しない場合、次のパケットと処理中のパケットを破棄してください。

以上のテストのすべて、またはいくつかを組み合わせる使用することによって、不正パケットを見つけることができます。

B.2 TXPのスタック

コマンド・レジスタ (CR) のTXPビットは次の2つの場合にスタックする (TXPビットをセットしてもパケットが送信されない) ことがあります。

(1) パケット長が原因の場合

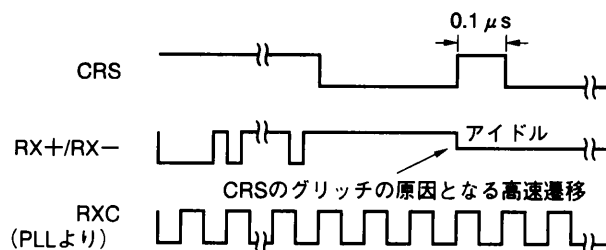
データ・コンフィギュレーション・レジスタ (DCR) に設定した送信 FIFO スレッシュホールド値 (TFT0, TFT1) 以下の長さのパケットを μPD72932に与えた場合、バス・ロック・アップが発生します。このロック・アップ状況を回避するには、常に送信スレッシュホールド値よりも長いパケットを送信してください。

(2) CRS ラインのグリッチが原因の場合

受信の最後で CRS ラインにグリッチ・パルスが加わると、CR の TXP ビットがセットされていても μPD72932 はパケットの送信を停止することがあります。このような状況になると、送信の停止 (HTX) ビットをセットして TXP ビットをクリアしようとしても失敗します。この問題は、受信リソースが不足した場合 (割り込みステータス・レジスタ (ISR) の RDE ビットがセットされているか、最後の受信バッファを使用している場合) にのみ発生します。

CRS 端子でのグリッチは、RX+/RX-ライン上のアンダシュートによって生じます。アンダシュートは、RX+/RX-ライン上でのハイ・レベルからアイドルへの遷移が速い (時間定数が高速である) ことによって生じます (図 B-1 参照)。

図B-1 CRSのグリッチ



この問題を解決する方法の1つとして、ソフトウェア・リセットがあります。ソフトウェア・リセットによって、これらの状況がリセットされ、μPD72932が正常に動作するようになります。ただし、いくつかのレジスタおよびバッファを再初期化しなければなりません。別の解決方法としては、根本的にリソース（RBA および RDA）不足を発生させないようにする方法もあります。最後に3つ目の方法として、RX+/RX-ラインの遷移を遅くする（時間定数を増やす）方法があります。

付録 C. C バージョンへの置き換えに関する Q & A

Q. 1

C バージョンに対応するソフトウェアは、B バージョンのものと互換性があるか？

A. 1

あります。例外は、受信アーキテクチャで 1 つの受信バッファ・エリア (RBA) に複数のパケットを格納する機能を使用しているアプリケーション・システムの場合です。この場合の受信動作については 4.4.4 (5) 1 つの RBA に複数のパケットを格納する場合を参照してください。

Q. 2

タイミングの改善は設計にどのように影響するか？

A. 2

一般に、タイミングを回路全体にわたって改善しています。このことは、古い設計は新しいタイミングを使用しても問題ないことを意味します。ただし、C バージョンでは出力ホールド時間に B バージョンよりも短くなったものがあります。B バージョンのより長い出力ホールド時間に依存している設計では注意が必要です。この点については 8.3 タイミングの改善で説明しています。B バージョンから C バージョンへの置き換えの際に問題が生じた場合に、どのタイミングを最も重要なものとして考慮しなければならないかを記載しています。

Q. 3

B バージョンのデータ・シートを持っていないが、必要か？

A. 3

必要ありません。現在、設計を行うときに潜在的な問題があるかどうかを評価するために必要なのは、このデータ・シートだけです。

Q. 4

C バージョンのシリコン・リビジョン・レジスタ (SRR) は B バージョンから更新されているか？

A. 4

更新されています。SRR は B バージョンでは 4H ですが、C バージョンでは 6H です。

Q. 5

CバージョンでSRRの値が更新されたことにどんな意味があるのか？

A. 5

SRRは μ PD72932のリビジョン・レベルを示します。これにより、ソフトウェア技術者はCバージョンでの改善点を自分たちが開発するソフトウェアに活用することができます。

Q. 6

タイミング・チャートはなぜ修正されたのか？

A. 6

タイミング・チャートは、 μ PD72932の動作を明確化するために修正しました。その結果、不適切なタイミングを削除しました。そのほかに、 μ PD72932の動作を明確に図示するため追加または変更したタイミングがあります。

(メ モ)

CMOSデバイスの一般的注意事項

①静電気対策（MOS全般）

注意 MOSデバイス取り扱いの際は静電気防止を心がけてください。

MOSデバイスは強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、NECが出荷梱包に使用している導電性のトレイやマガジン・ケース、または導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。

また、MOSデバイスを実装したボードについても同様の扱いをしてください。

②未使用入力の処理（CMOS特有）

注意 CMOSデバイスの入力レベルは固定してください。

バイポーラやNMOSのデバイスと異なり、CMOSデバイスの入力に何も接続しない状態で動作させると、ノイズなどに起因する中間レベル入力が生じ、内部で貫通電流が流れて誤動作を引き起こす恐れがあります。プルアップかプルダウンによって入力レベルを固定してください。また、未使用端子が出力となる可能性（タイミングは規定しません）を考慮すると、個別に抵抗を介してV_{DD}またはGNDに接続することが有効です。

資料中に「未使用端子の処理」について記載のある製品については、その内容を守ってください。

③初期化以前の状態（MOS全般）

注意 電源投入時、MOSデバイスの初期状態は不定です。

分子レベルのイオン注入量等で特性が決定するため、初期状態は製造工程の管理外です。電源投入時の端子の出力状態や入出力設定、レジスタ内容などは保証しておりません。ただし、リセット動作やモード設定で定義している項目については、これらの動作ののちに保証の対象となります。

リセット機能を持つデバイスの電源投入後は、まずリセット動作を実行してください。

V810は、日本電気株式会社の商標です。

イーサネットは、米国ゼロックス社の商標です。

本資料に掲載の応用回路および回路定数は、例示的に示したものであり、量産設計を対象とするものではありません。

- 文書による当社の承諾なしに本資料の転載複製を禁じます。
- 本資料に記載された製品の使用もしくは本資料に記載の情報の使用に際して、当社は当社もしくは第三者の知的所有権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。上記使用に起因する第三者所有の権利にかかわる問題が発生した場合、当社はその責を負うものではありませんのでご了承ください。
- 当社は品質、信頼性の向上に努めていますが、半導体製品はある確率で故障が発生します。当社半導体製品の故障により結果として、人身事故、火災事故、社会的な損害等を生じさせない冗長設計、延焼対策設計、誤動作防止設計等安全設計に十分ご注意願います。
- 当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定して頂く「特定水準」に分類しております。また、各品質水準は以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認の上ご使用願います。
 標準水準：コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット
 特別水準：輸送機器（自動車、列車、船舶等）、交通用信号機器、防災／防犯装置、各種安全装置、生命維持を直接の目的としない医療機器
 特定水準：航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器、生命維持のための装置またはシステム等
 当社製品のデータ・シート／データ・ブック等の資料で、特に品質水準の表示がない場合は標準水準製品であることを表します。当社製品を上記の「標準水準」の用途以外でご使用をお考えのお客様は、必ず事前に当社販売窓口までご相談頂きますようお願い致します。
- この製品は耐放射線設計をしておりません。

M4 94.11

— お問い合わせは、最寄りのNECへ —

【営業関係お問い合わせ先】

半導体第一販売事業部	〒108-01	東京都港区芝五丁目7番1号（NEC本社ビル）	東京 (03)3454-1111	（大代表）	
半導体第二販売事業部					
半導体第三販売事業部					
中部支社 半導体販売部	〒460	名古屋市中区錦一丁目17番1号（NEC中部ビル）	名古屋 (052)222-2170		
関西支社 半導体第一販売部	〒540	大阪府中央区城見一丁目4番24号（NEC関西ビル）	大阪 (06) 945-3178		
半導体第二販売部			大阪 (06) 945-3200		
半導体第三販売部			大阪 (06) 945-3208		
北海道支社	札幌 (011)231-0161	小長松支店	小松 (0285)24-5011	富山支店	富山 (0764)31-8461
東北支店	仙台 (022)261-5511	山形支店	山形 (0262)35-1444	山梨支店	山梨 (0592)25-7341
岩手支店	盛岡 (0196)51-4344	長野支店	長野 (0263)35-1666	三重支社	津 (075)344-7824
山形支店	山形 (0236)23-5511	諏訪支店	諏訪 (0266)53-5350	京都支社	京都 (078)333-3854
郡山支店	郡山 (0249)23-5511	甲府支店	甲府 (0552)24-4141	中国支社	広島 (082)242-5504
いわき支店	いわき (0246)21-5511	埼玉支店	大宮 (048)641-1411	鳥取支店	鳥取 (0857)27-5311
長岡支店	長岡 (0258)36-2155	立川支店	立川 (0425)26-5981	岡山支店	岡山 (086)225-4455
土浦支店	土浦 (0298)23-6161	千葉支店	千葉 (043)238-8116	四国支社	高松 (0878)36-1200
水戸支店	水戸 (0292)26-1717	静岡支店	静岡 (054)255-2211	新居浜支店	新居浜 (0897)32-5001
神奈川支社	横浜 (045)324-5511	沼津支店	沼津 (0559)63-4455	松山支店	松山 (0899)45-4111
群馬支店	高崎 (0273)26-1255	浜松支店	浜松 (053)452-2711	九州支社	福岡 (092)271-7700
太田支店	太田 (0276)46-4011	北陸支店	金沢 (0762)23-1621	北九州支店	北九州 (093)541-2887
宇都宮支店	宇都宮 (0286)21-2281	福井支店	福井 (0776)22-1866		

【本資料に関する技術お問い合わせ先】

半導体ソリューション技術本部	〒210	川崎市幸区塚越三丁目484番地	川崎 (044)548-8884	半導体 インフォメーションセンター FAX(044)548-7900 (FAXにてお願い致します)
第一システム技術部				
半導体販売技術本部	〒108-01	東京都港区芝五丁目7番1号（NEC本社ビル）	東京 (03)3798-9619	
東日本販売技術部				
半導体販売技術本部	〒460	名古屋市中区錦一丁目17番1号（NEC中部ビル）	名古屋 (052)222-2125	
中部販売技術部				
半導体販売技術本部	〒540	大阪府中央区城見一丁目4番24号（NEC関西ビル）	大阪 (06) 945-3383	
西日本販売技術部				