

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ひ申し上げます。

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と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。

保守/廃止

V40™, V50™, V40HL™, V50HL™

16/8, 16ビット・マイクロプロセッサ

ソフトウェア編

μPD70208

μPD70216

μPD70208(A)

μPD70216(A)

μPD70208H

μPD70216H

— 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ーやマガジン・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。

ハイボラや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、内部で漏れ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。

インターツールは米国Intermetrics Microsysteme Software, Inc.の商標です。

V20, V30, V20HL, V30HL, V40, V50, V40HL, V50HL, Vシリーズは日本電気株式会社の商標です。

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。

- 本資料の内容は、後日変更する場合があります。
- 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ください。
- 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。

本版で改訂された主な箇所

箇所	内容
全般	対象品種に V40HL, V50HL を追加
p. 4	2.1.2 SCTL (System Control register) 追加
p. 10	2.1.9 RFC (Refresh Control register) ・ビット7の名称を RFE に変更 ・注意追加
p. 49-54	3.1.5 プログラム・リスト 全面変更 (RA70116-I を使用)
p. 62-69	3.2.5 プログラム・リスト ”
p. 77-83	3.3.5 プログラム・リスト ”

本文欄外の★印は、本版で改訂された主な箇所を示しています。

巻末にアンケート・コーナーを設けております。このドキュメントに対するご意見をお気軽にお寄せください。

はじめに

対象者 このアプリケーション・ノートは、次の製品の機能を理解し、それらを使用した応用システムを設計するユーザを対象とします。

- μ PD70208, 70208(A) (別名称V40)
- μ PD70208H (別名称V40HL)
- μ PD70216, 70216(A) (別名称V50)
- μ PD70216H (別名称V50HL)

目的 このアプリケーション・ノートでは、上記の各製品が持つソフトウェア機能の応用方法をユーザに即解していただくことを目的としています。

構成 このアプリケーション・ノートは大きく分けて次の内容で構成しています。

- 概要
- 内蔵ペリフェラル
- 応用例
 - ・ タイマ割り込みを用いた時間カウント
 - ・ RS-232-C インタフェース
 - ・ DMA U とリフレッシュ機能の応用例

読み方 このアプリケーション・ノートの読者には、電気、論理回路、およびマイクロコンピュータに関する一般的知識を必要とします。

このアプリケーション・ノートでは、各製品の機能面に違いのない場合、 μ PD70208, 70208(A), 70208H を μ PD70208 に、 μ PD70216, 70216(A), 70216H を μ PD70216 に名称を統一してあります。 μ PD70208(A), 70216(A), 70208H, 70216H のアプリケーション・ノートとしてお使いの際は、それぞれの品名を次のように読み替えてください。

本文の品名	対応する品名	
μ PD70208	μ PD70208(A)	μ PD70208H
μ PD70216	μ PD70216(A)	μ PD70216H

各製品に内蔵される I/O レジスタの機能を知りたいとき

→ 第 2 章 内蔵ペリフェラルを参照してください。

各製品の内蔵ペリフェラルに関するプログラム例を知りたいとき

→ 第 3 章 応用例を参照してください。

各製品の電気的特性を知りたいとき

→各製品に対応した別冊のデータ・シートを参照してください。

各製品のハードウェア機能を知りたいとき

→各製品に対応した別冊のユーザーズ・マニュアル ハードウェア編を参照してください。

各製品の命令機能を知りたいとき

→別冊の16ビットVシリーズ™ユーザーズ・マニュアル 命令編を参照してください。

μPD70208H, 70216Hのハードウェア設計に関する応用例を知りたいとき

→別冊のV40HL, V50HL アプリケーション・ノート ハードウェア設計編を参照してください。

- 凡 例
- データ表記の重み : 左が上位桁, 右が下位桁
 - アクティブ・ロウの表記 : $\overline{\text{xxx}}$ (端子, 信号名称に上線)
 - メモリ・マップのアドレス : 上部-上位, 下部-下位
 - 注 : 本文中につけた注の説明
 - 注意 : 気をつけて読んでいただきたい内容
 - 備考 : 本文の補足説明
 - 数の表記 : 2進数... xxx または xxx B
 10進数... xxx
 16進数... xxx H
 - 2のべき数を示す接頭語 (アドレス空間, メモリ容量) :
 K (キロ) : $2^{10} = 1024$
 M (メガ) : $2^{20} = 1024^2$

関連資料 ● μPD70208, 70208(A), 70216, 70216(A)に関する資料

資 料 名	資料番号
データ・シート	U10154J
ユーザーズ・マニュアル ハードウェア編	U10666J
16ビットVシリーズ ユーザーズ・マニュアル 命令編	IEU-804

● μPD70208H, 70216Hに関する資料

資 料 名	資料番号
データ・シート	IC-8221
ユーザーズ・マニュアル ハードウェア編	IEU-770
16ビットVシリーズ ユーザーズ・マニュアル 命令編	IEU-804
アプリケーション・ノート ハードウェア設計編	U10037J
レジスタ活用表	IEM-5575

●開発ツールに関する資料（ユーザズ・マニュアル）

製 品 名		資料番号
IE-70208H-BX, IE-70216H-BX (インサーキット・エミュレータ) ^注		EEU - 864
EB-70216H-98, EB-70216H-PC (エバリュエーション・ボード) ^注		EEU - 819
RA70116-1 (インターツール TM ・アセンブラ)	操作編	EEU - 869
SP70116-1 (インターツール・ソフトウェア・パッケージ)	言語編	EEU - 861
LC70116 (ロケータ)		EEU - 834
TD208H-BX, TD216H-BX (ターボ・ディバッガ) ^注		EEU - 889
RX116 Ver. 2.1 (リアルタイム OS)	基礎編	IEM - 5069
	テクニカル編	EEM - 719
	ユーザズ・インストールソフト編	EEU - 747

注 対応製品：μ PD70208H, 70216H

目 次

第1章 概 要 … 1

第2章 内蔵ペリフェラル … 3

2.1 システムI/Oレジスタ … 3

2.1.1 OPCN (On-chip Peripheral Connection register) … 3

2.1.2 SCTL (System Control register) … 4

★

2.1.3 OPSEL (On-chip Peripheral Selection register) … 5

2.1.4 内蔵ペリフェラル・リロケーション・レジスタ … 5

2.1.5 TCKS (Timer Clock Selection register) … 7

2.1.6 WMB (Programmable Wait,Memory Boundary register) … 8

2.1.7 WCY1 (Programmable Wait, cycle register 1) … 9

2.1.8 WCY2 (Programmable Wait, cycle register 2) … 9

2.1.9 RFC (Refresh Control register) … 10

2.2 TCU (タイマ/カウンタ・ユニット) … 11

2.2.1 カウンタのプログラム … 11

2.2.2 カウンタからの読出し … 13

2.3 SCU (シリアル・コントロール・ユニット) … 16

2.3.1 SMD (Serial Mode register) … 16

2.3.2 SCM (Serial Command register) … 18

2.3.3 SST (Serial Status register) … 19

2.3.4 SIMK (Serial Interrupt Mask register) … 21

2.4 ICU (割込みコントロール・ユニット) … 22

2.4.1 イニシャライズ&コマンド・ワード・レジスタ群 … 22

2.4.2 IMK (Interrupt Mask register) … 27

2.4.3 IRQ (Interrupt Request register) … 28

2.4.4 IIS (Interrupt In Service register) … 28

2.4.5 各種モード、コマンド … 28

2.5 DMAU (DMAコントロール・ユニット) … 34

2.5.1 レジスタ … 34

2.5.2 コマンド … 35

第3章 応 用 例 … 43

3.1 タイマ割込みを用いた時間カウント … 43

3.1.1 概 要 … 43

3.1.2 ハードウェアの説明 … 43

3.1.3 ソフトウェアの説明 … 44

3.1.4 フローチャート … 46

3.1.5 プログラム・リスト … 49

3.2 RS-232-Cインタフェース … 55

3.2.1 概 要 … 55

3.2.2 ハードウェアの説明 … 55

3.2.3 ソフトウェアの説明 … 56

- 3.2.4 フローチャート … 58
- 3.2.5 プログラム・リスト … 62
- 3.3 DMAUとリフレッシュ機能の応用例 … 70**
 - 3.3.1 概 要 … 70
 - 3.3.2 ハードウェアの説明 … 70
 - 3.3.3 ソフトウェアの説明 … 70
 - 3.3.4 フローチャート … 75
 - 3.3.5 プログラム・リスト … 77

図の目次(1/2)

図番号	タイトル, ページ
2-1	OPCNの設定フォーマット ... 3
2-2	SCTLの設定フォーマット ... 4
2-3	OPSELの設定フォーマット ... 5
2-4	リロケーション・レジスタの設定フォーマット ... 6
2-5	内蔵ペリフェラルのリロケーション ... 7
2-6	TCKSの設定フォーマット ... 8
2-7	WMBの設定フォーマット ... 8
2-8	WCY1の設定フォーマット ... 9
2-9	WCY2の設定フォーマット ... 9
2-10	RFCの設定フォーマット ... 10
2-11	モード・ワード ... 12
2-12	カウント・ラッチ・コマンド・フォーマット ... 14
2-13	マルチプル・ラッチ・コマンド・フォーマット ... 14
2-14	ステータス・フォーマット ... 15
2-15	SMDの設定フォーマット ... 17
2-16	SCMの設定フォーマット ... 18
2-17	$\overline{\text{SRDY}}$ 信号 ... 18
2-18	SSTのフォーマット ... 19
2-19	オーバラン・エラー ... 20
2-20	SIMKの設定フォーマット ... 21
2-21	IIW1のフォーマット ... 23
2-22	IIW2のフォーマット ... 23
2-23	IIW3のフォーマット ... 24
2-24	IIW4のフォーマット ... 25
2-25	IPFWのフォーマット ... 25
2-26	INTL要求の優先順位 ... 26
2-27	IMDW (Interrupt Mode Word) ... 27
2-28	IMKW (Interrupt Mask Word) ... 27
2-29	通常ネスト・モード ... 29
2-30	例外ネスト・モード ... 30
2-31	イニシャライズ・コマンド ... 36
2-32	チャンネル・レジスタ・リード ... 36
2-33	チャンネル・レジスタ・ライト ... 37
2-34	カウント・レジスタ・リード/ライト ... 38
2-35	アドレス・レジスタ・リード/ライト ... 38
2-36	デバイス・コントロール・レジスタ・リード/ライト ... 39
2-37	拡張ライト・タイミング ... 40
2-38	モード・コントロール・レジスタ・リード/ライト ... 40
2-39	ステータス・レジスタ・リード ... 41
2-40	マスク・レジスタ・リード/ライト ... 42
3-1	CPUとTCU,ICUの関係 ... 43
3-2	応用例2 RS-232-Cインタフェース ... 55

図の目次(2/2)

図番号	タイトル, ページ
3-3	応用例3(1) DMAUと μ PD71051との接続 ... 73
3-4	応用例3(2) DRAM周辺 ... 74

表の目次

表番号	タイトル, ページ
2-1	TCUへの書込み ... 11
2-2	カウンタ数書込み ... 13
2-3	TCUからの読出し ... 13
2-4	SCUレジスタ・アクセス ... 16
2-5	ICUレジスタ・アドレッシング ... 22
2-6	割込みベクタ番号の発生 ... 24
2-7	レジスタ構成 ... 34
2-8	コマンド構成 ... 35
2-9	初期化されたレジスタの状態 ... 36

第1章 概要

μPD70208, 70208(A), 70216, 70216(A)は, μPD70108(V20™), μPD70116(V30™)相当のCPUと, 数種類のペリフェラルを内蔵したマイクロプロセッサです.

μPD70208H, 70216HはμPD70108H(V20HL™), μPD70116H(V30HL™)相当のCPUを内蔵し, μPD70208, 70208(A), 70216, 70216(A)を高速/低消費電力/高機能化したマイクロプロセッサです.

オンチップで内蔵しているペリフェラルには以下のものがあります.

- ・クロック・ジェネレータ(CG)
- ・プログラマブル・ウェイト・コントロール・ユニット(WCU)
- ・リフレッシュ・コントロール・ユニット(REFU)
- ・タイマ・カウンタ・ユニット(TCU)
- ・シリアル・コントロール・ユニット(SCU)
- ・割込みコントロール・ユニット(ICU)
- ・DMA・コントロール・ユニット(DMAU)

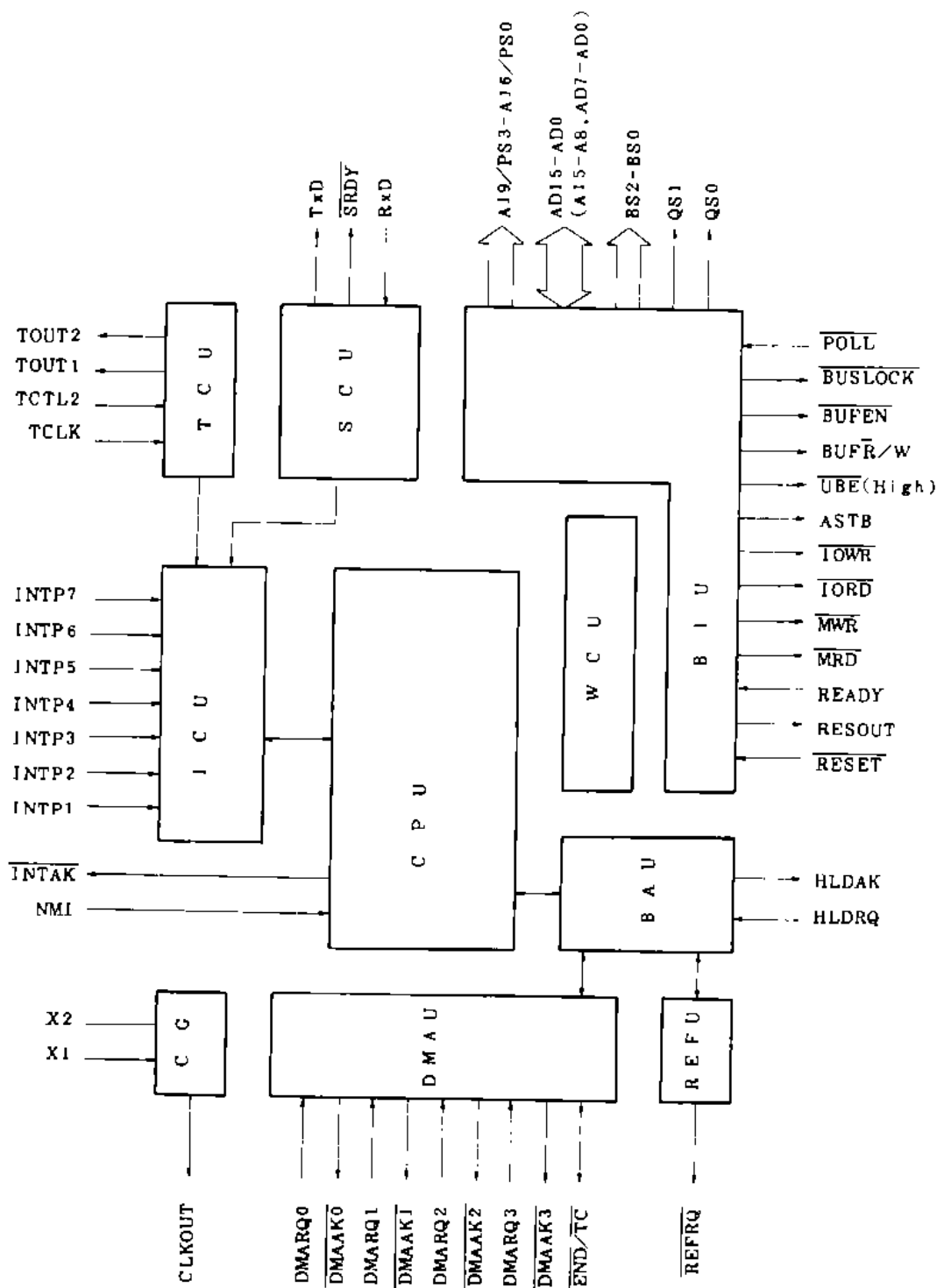
このほかに兼用端子の設定を行ったり, 内蔵ペリフェラルのリロケーションを行うシステムI/Oレジスタがあります.

このアプリケーション・ノートでは各製品を使用する際に必要となるこれら各I/Oのイニシャライズ方法について説明します.

なお, このアプリケーション・ノートで紹介している応用例では, μPD70208H, 70216Hに追加された次の機能は使用していません.

- ・STOPモード
- ・μPD71037モード
- ・ポー・レート・ジェネレータ
- ・REFRQ拡張タイミング
- ・インストラクション・サイクル可変
- ・WCU制御によるメモリ, I/O拡張分割

μPD70208, μPD70216ブロック図



() 内社 #PD70208

第2章 内蔵ペリフェラル

この章では #PD70208, #PD70216 に内蔵されているペリフェラルについてその概要を説明します。

2.1 システムI/Oレジスタ

I/O空間の FFF0H-FFFFH の 16 バイトはシステム領域となっています。この領域には、DMAU, ICU, TCU, SCU の I/O アドレスを決定するレジスタや、REFU, WCU 用のレジスタなどが割当てられています。

2.1.1 OPCN(On-chip Peripheral Connection register)

このレジスタは ICU の INTL1, INTL2 への入力信号の切換えと DMARQ/RxD, DMAAK/TxD, INTAK/SRDY/TOUT1 の 3 本の端子の機能の選択を行います。

図 2-1 OPCN の設定フォーマット

7	6	5	4	3	2	1	0
-	-	-	-	IRSW	PF		

- - don't care

IRSW (Interrupt Request Switch)

IRSW	INTL1	INTL2
00	INTP1 端子	INTP2 端子
01	SCU	INTP2 端子
10	INTP1 端子	TCU (TCT#1)
11	SCU	TCU (TCT#1)

PF (Pin Function)

PF	DMARQ3/RxD	DMAAK3/TxD	INTAK/SRDY/TOUT1
00	DMARQ3	DMAAK3	INTAK
01	DMARQ3	DMAAK3	TOUT1
10	RxD	TxD	INTAK
11	RxD	TxD	SRDY

★ 2.1.2 SCTL (System Control register)
[対象: μPD70208H.70216H]

μPD70208H.70216Hが内蔵しているレジスタです。
μPD70208.70216から追加された各機能の設定を行います。

図2-2 SCTLの設定フォーマット

7	6	5	4	3	2	1	0
0	0	EREF	SC	0	CE	DMAM	IOAG

IOAG

IOAG	機 能
0	内蔵I/Oアドレスを奇数/偶数に固定する(16ビット・バウンダリ)
1	内蔵I/Oアドレスを連続にする(8ビット・バウンダリ)

DMAM

DMAM	DMAUモード指定
0	μPD71071モード
1	μPD71037モード

CE

CE	キャリー伝搬の指定
0	キャリーをA16に伝搬しない
1	キャリーをA16に伝搬する

注 μPD71037モード時のみ

SC

SC	SCUの入力クロック指定
0	TOUT1を使用する
1	専用ポーレート・ジェネレータを使用する

EREF

EREF	REFRQ出力の指定
0	T1の立ち上がりに同期して出力する
1	T4の立ち上がりに同期して出力する

注意 ビット3、ビット6、ビット7は必ず0に設定してください。

2.1.3 OPSEL(On-chip Peripheral Selection register)

このレジスタは、SCU,TCU,ICU,DMAUの各内蔵ペリフェラルの選択をします。対応するビットを0にすれば使用不可、1にすれば使用可になります。使用不可状態では、そのペリフェラルはI/O空間上には存在しなくなります。

このレジスタはリセットでビット3-ビット0が0に初期化されます。

図2-3 OPSELの設定フォーマット

7	6	5	4	3	2	1	0
-	-	-	-	SS	TS	IS	DS

SS, TS, IS, DS	選 択
0	使 用 不 可
1	使 用 可

SS:SCU Selection
 TS:TCU Selection
 IS:ICU Selection
 DS:DMAU Selection

2.1.4 内蔵ペリフェラル・リロケーション・レジスタ

SCU,TCU,ICU,DMAUの四つの内蔵ペリフェラルのI/O空間上でのアドレス・ロケーションは、256バイト境界の任意のブロック内に設定できます。また、その256バイトのブロック内での各ペリフェラルのアドレスも数バイト間隔(4,8,16バイト)で任意に設定できます。

256バイトごとのブロックはOPHAで設定し、そのブロック内の位置はSULA,TULA, IULA,DULAで設定します。

ペリフェラルの占めるI/O領域は、DMAUが16バイト、TCU,SCU,ICUが4バイトです。アドレスが重ならないように注意してください。

内蔵ペリフェラルのうちDMAU以外へのアクセスはバイト・アクセス(バイトIN/OUT命令)で行なってください。

なお、μPD70216, 70216(A), またはSCTLのIOAGビットを0に設定したμPD70208H, 70218HではDMAU以外はTULA, SULA, IULAのA0ビットによって偶数または奇数アドレスに配置されます。

図 2-4 リロケーション・レジスタの設定フォーマット

OPHA(On-chip Peripheral High Address register)

7	6	5	4	3	2	1	0
A15	A14	A13	A12	A11	A10	A9	A8

DULA(DMAU Low Address register)

7	6	5	4	3	2	1	0
A7	A6	A5	A4	-	-	-	-

IULA(ICU Low Address register)

7	6	5	4	3	2	1	0
A7	A6	A5	A4	A3	A2	-	-

注 1

A7	A6	A5	A4	A3	-	-	A0
----	----	----	----	----	---	---	----

注 2

TULA(TCU Low Address register)

7	6	5	4	3	2	1	0
A7	A6	A5	A4	A3	A2	-	-

注 1

A7	A6	A5	A4	A3	-	-	A0
----	----	----	----	----	---	---	----

注 2

SULA(SCU Low Address register)

7	6	5	4	3	2	1	0
A7	A6	A5	A4	A3	A2	-	-

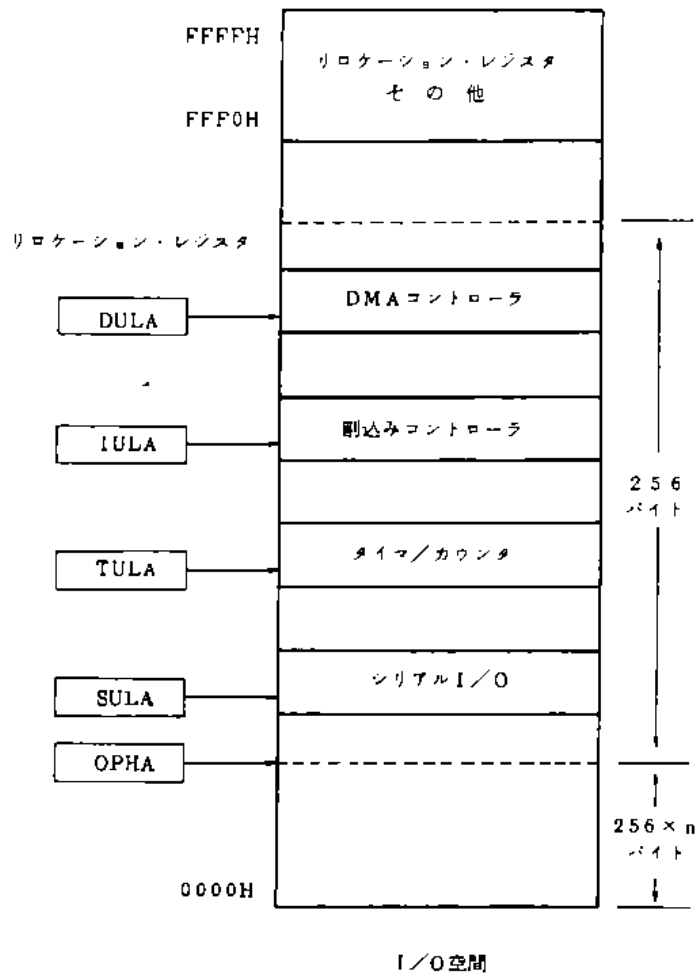
注 1

A7	A6	A5	A4	A3	-	-	A0
----	----	----	----	----	---	---	----

注 2

- 注 1. μ PD70208, 70208(A), またはSCTLのIOAGビットを1に設定した μ PD70208H, 70216H
2. μ PD70216, 70216(A), またはSCTLのIOAGビットを0に設定した μ PD70208H, 70216H

図 2-5 内蔵ペリフェラルのリロケーション



2.1.5 TCKS (Timer Clock Selection register)

このレジスタはTCUの各カウンタ (TCT#2-0)に供給するクロックを外部TCLK端子からの入力にするか、または内部のクロック信号を分周した信号にするかの選択をします。また、内部クロックを供給する場合のプリスケアラの分周数を設定します。

図 2-6 TCKS の設定フォーマット

7	6	5	4	3	2	1	0
-	-	-	CS2	CS1	CS0	PS	

CSn (Clock Selection of TCT#n)

CS2/CS1/CS0	供給クロック
0	内部クロック
1	TCLK 端子入力

PS (Prescale of internal clock)

PS	内部クロック分周数
00	2
01	4
10	8
11	16

2.1.6 WMB (Programmable Wait, Memory Boundary register)

このレジスタはメモリ・サイクルでのウェイト・ステート数をメモリ・ブロック (メモリの種類) ごとに指定するためメモリ空間の分割を行います。

LMB, UMB の各 3 ビットによって下位メモリ・ブロックと上位メモリ・ブロックのサイズを決定します。下位ブロックと上位ブロックの間のブロックが中位ブロックとなります。

図 2-7 WMB の設定フォーマット

7	6	5	4	3	2	1	0
-	LMB			-	UMB		

LMB (size of Lower Memory Block)

UMB (size of Upper Memory Block)

LMB/UMB	メモリ・ブロック・サイズ
000	32KB
001	64KB
010	96KB
011	128KB
100	192KB
101	256KB
110	384KB
111	512KB

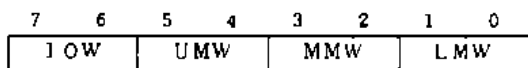
2.1.7 WCY1 (Programmable Wait, cycle register 1)

このレジスタは三つのメモリ領域および外部I/Oのアクセス・サイクルにおけるウェイト・ステート数の設定を行います。

WCY1はリセットで全ビットが1に初期化されます。

内部I/Oをアクセスする場合、WCUおよびREADY入力は無効となり、バス・サイクルは常に4クロックで終了します。

図 2 - 8 WCY1 の設定フォーマット



IOW (Wait for the I/O)
 UMW (Wait for the Upper Memory block)
 MMW (Wait for the Middle Memory block)
 LMW (Wait for the Lower Memory block)

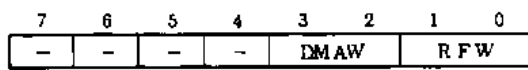
IOW/UMW/MMW/LMW	ウェイト・ステート数
00	0 (No Wait)
01	1
10	2
11	3

2.1.8 WCY2 (Programmable Wait, cycle register 2)

このレジスタはDMAUによるDMAサイクルでのウェイト・ステート数の指定とREFUKによるメモリ・リフレッシュ・サイクルでのウェイト・ステート数の指定を行います。

WCY2はリセットでビット3-0が1に初期化されます。

図 2 - 9 WCY2 の設定フォーマット



DMAW (Wait for the DMA cycle)
 RFW (Wait for the Refresh cycle)

DMAW/RFW	ウェイト・ステート数
00	0 (No Wait)
01	1
10	2
11	3

★ 2.1.9 RFC(Refresh Control register)

RFEビットはREFUによるリフレッシュ機能の許可/禁止を設定します。

RTMの5ビットはリフレッシュ間隔の設定します。リフレッシュ間隔が短かすぎるとCPUのスループットが落ちますので、RTMの設定が00000-00011の場合、タイマ・ファクタNは1-4ではなく17-20になります。実際のリフレッシュ間隔はNによって決まり $8 \times N \times$ (CPUのCLOCK周期)となります。動作クロックが8MHzでN=10の場合は

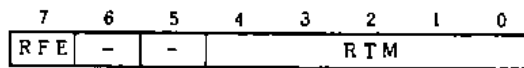
$$8 \times 10 \times \frac{1}{8,000,000} (\text{sec}) = 10 (\mu\text{sec})$$

となります。

リセットによって、RFEビット、RTMビットは次のようになります。

- ・ RFEビット
 - | パワーオン・リセット時…不定
 - | 動作中のリセット入力時…変化しない
- ・ RTMビット：01000 (N=9) に初期化

図2-10 RFCの設定フォーマット



RFE (Refresh Enable) RTM (Refresh Timer)

RFE	機 能
0	リフレッシュ禁止
1	リフレッシュ許可

RTM	N (タイマ・ファクタ)
00000B	17
00001	18
00010	19
00011	20
00100	5
00101	6
00110	7
⋮	⋮
⋮	⋮
11110	31
11111B	32

注意1. RFEビットは、パワーオン・リセット時は不定です（リフレッシュ許可/禁止は規定されません）。

リフレッシュ・サイクルを1回も実行せずにリフレッシュ禁止にしたい場合は、RESOUTがインアクティブ後、72クロック以内にRFEビットを0に設定してください。

2. リフレッシュ・キューにリフレッシュ要求が保留されているとき、RFEビットを0（リフレッシュ禁止）にしても、リフレッシュ・キューの内容が0になるまでリフレッシュ・サイクルを実行します。リフレッシュ・キューはリセットによりクリアされます。

2.2 TCU (タイマ/カウンタ・ユニット)

TCUは3個のカウンタを持っており、タイマ、イベント・カウンタとして使用できます。ここではカウンタのプログラム(設定)とカウンタからの脱出しに分けて説明します。

2.2.1 カウンタのプログラム

TCUはCPUからのプログラムにより制御されますので、カウンタを使用するときはモード・ワードを書込んで使用するカウンタのモードを指定し、さらにそのカウンタにどれだけカウントするか決定するカウント数を書込まなければなりません。

表 2-1 TCUへの書き込み

注 1		注 2		書 込 み 対 象
A1	A0	A2	A1	
0	0	0	0	TCT#0
0	1	0	1	TCT#1
1	0	1	0	TCT#2
1	1	1	1	TMD (モード・レジスタ)

- 注 1. μ PD70208, 70208 (A), またはSCTLのIOAGビットを1に設定した μ PD70208H, 70216H
2. μ PD70216, 70216 (A), またはSCTLのIOAGビットを0に設定した μ PD70208H, 70216H

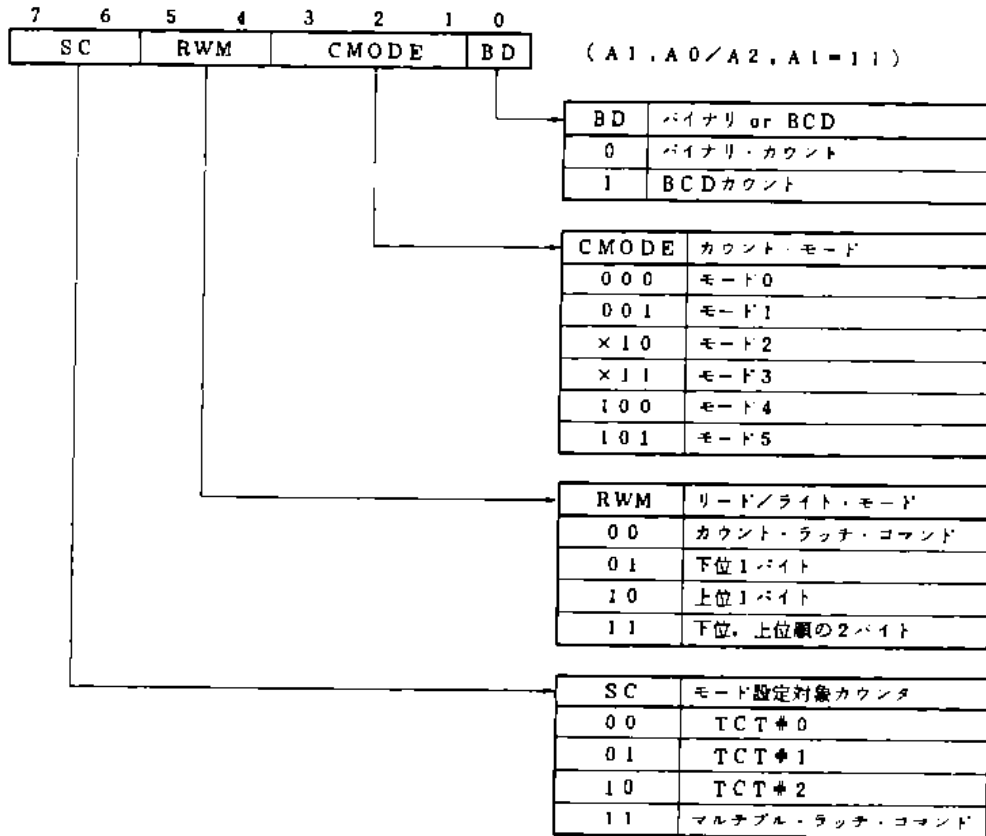
(1) モード・ワード

カウンタを動作させるにはモード・ワードをTMD(Timer Counter Mode register)に書き込み、使用するカウンタのモードを指定します。A1, A0 またはA2, A1を'11'として書き込みます。モード・ワードは8ビット構成で、次に示す四つのブロックに分けることができます。

- SC(Select Counter or Multiple Latch Command)……2ビット
TCT#2-0またはマルチプル・ラッチ・コマンドの指定をします。カウンタを指定した場合、SC以外は指定されたカウンタに対してのみ意味を持ちます。マルチプル・ラッチ・コマンドについては後述します(2.2.2(3)参照)。
- RWM(Read/Write Mode)……2ビット
カウント・レジスタへの脱出し書き込み時のデータ幅を規定するリード/ライト・モード(下位・上位2バイト, 下位1バイト, 上位1バイト)。またはカウント・ラッチ・コマンドの指定をします。カウント・ラッチ・コマンドについては後述します(2.2.2(2)参照)。
- CMODE(Count Mode)……3ビット
カウント・モード5~0を指定します。

- ・ BD(Binary or Decimal)……1ビット
 バイナリ・カウント, またはBCDカウントの指定をします。バイナリの場合は 0H-0FFFFH, BCDの場合は0-9999のカウント数の設定が可能です。

図 2-11 モード・ワード



×: don't care

TMDにモード・ワード(後述のカウント・ラッチ・コマンド, マルチプル・ラッチ・コマンドではない)が書込まれると, SCで選択されたカウンタの入アータス・レジスタの下位6ビットにTMDの下位6ビットがコピーされます。以後, そのカウンタのモードは再びモード・ワードが書込まれない限り変化しません。

② カウント数の書込み

カウンタのモード指定を行なった後, そのカウンタにカウント数を書込みます。A1, A0 または A2, A1を目的のカウンタを指定するように設定してからカウント数を書込めば, そのカウンタのカウント・レジスタに書込まれます。新しいカウント数はどの時点でも書込み可能です。カウント数の書込みでは, モード指定の際に設定されたリード/ライト・モードに注意する必要があります。下位1バイト, 上位1バイト・モードに設定されている場合は, 1回の書込みで, それぞれカウント・レジスタの下位・上位バイトに書込まれてカウント数の書込みは完了します。この場合, それぞれの残りの1バイトは00Hになります。下位・上位2バ

イト・モードでは、1回目の書き込みで下位バイトに書き込まれます。続く2回目の書き込みで上位バイトに書き込んだ時点で、カウント数の書き込みが完了します。たとえば、現在下位1バイト・モードに設定されているカウンタに対して、8801H という2バイトのカウント数を書込もうとして、1回目に01Hを書込み、2回目に88Hを書込んだ場合は、カウント・レジスタには最初0001Hが書き込まれ、続いて新しいカウント数0088Hが書き込まれたこととなります。

表2-2 カウント数書き込み

リード/ライト・モード	書き込み回数	カウント・レジスタ	
		上位バイト	下位バイト
下位1バイト	1	00H	××H
上位1バイト	1	××H	00H
下位・上位2バイト	2	××H (2回目)	××H (1回目)

2.2.2 カウンタからの読出し

TCUにはカウンタの内容を読出す方法が三つありますが、そのうちの二つでは動作中のカウンタに影響を与えずに読出すことができます。特にマルチプル・ラッチ・コマンドを使用した場合は、カウンタのステータスの読出しも可能です。

表2-3 TCUからの読出し

注1		注2		読出し対象
A1	A0	A2	A1	
0	0	0	0	TCT#0, TST0
0	1	0	1	TCT#1, TST1
1	0	1	0	TCT#2, TST2

- 注1. μ PD70208, 70208(A)、またはSCTLのIOAGビットを1に設定した μ PD70208H, 70216H
2. μ PD70216, 70216(A)、またはSCTLのIOAGビットを0に設定した μ PD70208H, 70216H

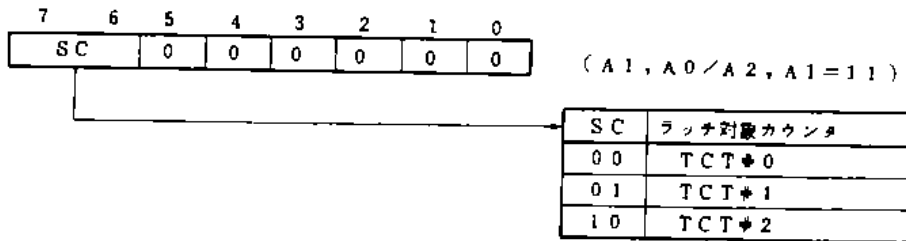
(1) カウント・データの直接読出し

A1, A0またはA2, A1で指定されたカウンタを読出すことによって、現在のカウント・データを読出すことができます。しかし、この方法ではダウンカウンタの現在の状態を読出しますので、読出し動作中にその値が変化することもありますので、得られる値は不正確となります。正確な値を得るには、TCLK入力やTCTL入力を操作してカウントを停止した状態で読出す必要があります。

(2) カウント・ラッチ・コマンド後の読出し

カウント・ラッチ・コマンドを発行すると、SCビットで選択されたカウンタのダウンカウンタの内容、すなわちカウント・データがカウント・ラッチにラッチされます。ラッチされたカウント・データはそのカウンタが読出されるか、またはそのカウンタが新たにモード指定されるまでは保持されます。ラッチされた状態でカウンタを読出せば、ラッチ・コマンド発行時の正確なカウント・データをカウンタの動作に影響を与えずに読出すことができます。

図 2-12 カウント・ラッチ・コマンド・フォーマット

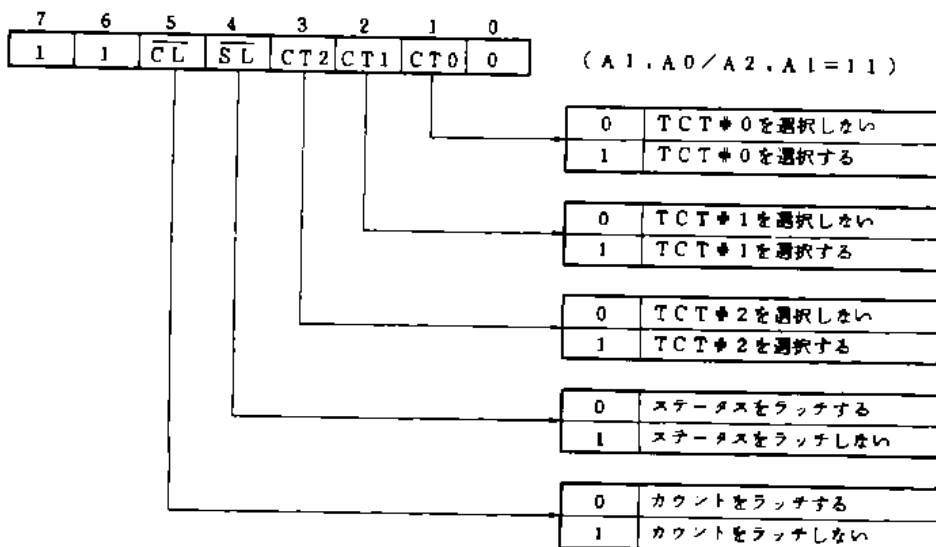


もしカウント・ラッチ・コマンドを発行して、そのカウンタを読まないうちに再び同じカウンタに対してカウントに対して2度目のコマンドを発行すると2度目のコマンドは無視され、以前にラッチされた値が保持されます。ラッチされているカウント・データを読出せばラッチは解除され、カウント・ラッチはダウンカウンタと同じ値を示します。

(3) マルチプル・ラッチ・コマンド後の読出し

マルチプル・ラッチ・コマンドを発行すれば、選択されたカウンタすべての、その時点でのカウント・データ、ステータスのどちらかまたは両方がそれぞれのカウンタ内のカウント・ラッチまたは、ステータス・ラッチにラッチされます。ラッチされた状態でカウンタを読出せば、コマンド発行時でのカウント・データおよびステータスをカウンタの動作に影響を与えずに読出すことができます。

図 2-13 マルチプル・ラッチ・コマンド・フォーマット



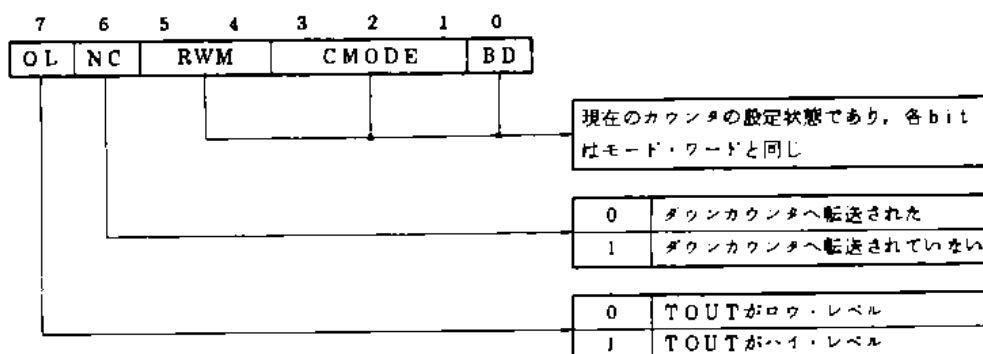
CT2-0の3ビットはそれぞれTCT#2-TCT#0に対応していて、そのビットが1となっているカウンタ全部がコマンドの対象となります。このことにより、1度のマルチプル・ラッチ・コマンドで1度に複数のカウンタのラッチができます。

\overline{CL} (Count Latch) ビットを0とすれば、選択されたカウンタ全部のカウンタ・データがそれぞれのカウンタ・ラッチにラッチされます。

\overline{SL} (Status Latch) ビットを0とすれば、選択されたステータスがラッチされ図2-13のフォーマットで読出されます。ステータスのビット5-0はそのカウンタの状態、OL (Output Level) ビットはそのカウンタの出力状態を示しています。NC (Null Count) ビットはカウント無効フラグで、最も新しく書込まれたカウント数がダウンカウンタへ転送されているか、いないかを示しています。

NC=0で転送されたことを示します。

図2-14 ステータス・フォーマット



2.3 SCU (シリアル・コントロール・ユニット)

SCUを使用するときはSMDにモード・ワードを書込み、使用するモードを設定します。

また、モードを設定してからも、SCUからの割込みの発生制御とか、送受信の許可/禁止、エラー・フラグのクリア、 $\overline{\text{SRDY}}$ 出力端子の制御等を、割込みマスク・レジスタSIMK、コマンド・レジスタSCMへの書込みによってプログラムします。ステータス・レジスタSSTの読出しはデータの入出力を情報として必要とします。

表2-4 SCUレジスタ・アクセス

注1		注2		レジスタ名	操作
A1	A0	A2	A1		
0	0	0	0	SRB	リード
				STB	ライト
0	1	0	1	SST	リード
				SCM	ライト
1	0	1	0	SMD	ライト
1	1	1	1	SIMK	リード/ライト

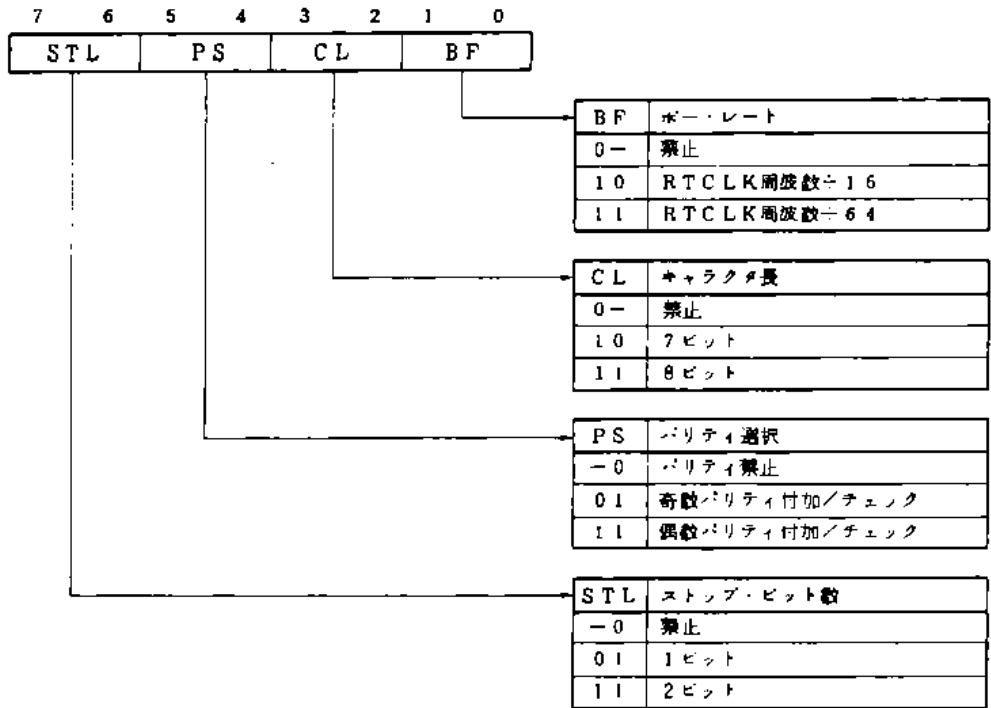
注1. $\mu\text{PD70208}$, $70208(A)$, またはSCTLのIOAGビットを1に設定した $\mu\text{PD70208H}$, $70216H$

2. $\mu\text{PD70216}$, $70216(A)$, またはSCTLのIOAGビットを0に設定した $\mu\text{PD70208H}$, $70216H$

2.3.1 SMD (Serial Mode register)

このレジスタには、送受信のモードを決めるモード・ワードを設定します。

図 2 - 15 SMDの設定フォーマット



BF(Baud rate factor)は、RTCLK(TOUT1)の周波数とボー・レートとの関係を規定します。送受信クロックに対してボー・レートを16分の1にするか64分の1にするかを選択します。

CL(Character Length)は、1キャラクタのビット数の設定で、7ビットまたは8ビットの選択をします。7ビットを選択した場合、SCUはCPUの書込んだ8ビット・データの下位7ビットを受け取り、SCUがCPUへ出力するデータは上位1ビットが0に固定されます。

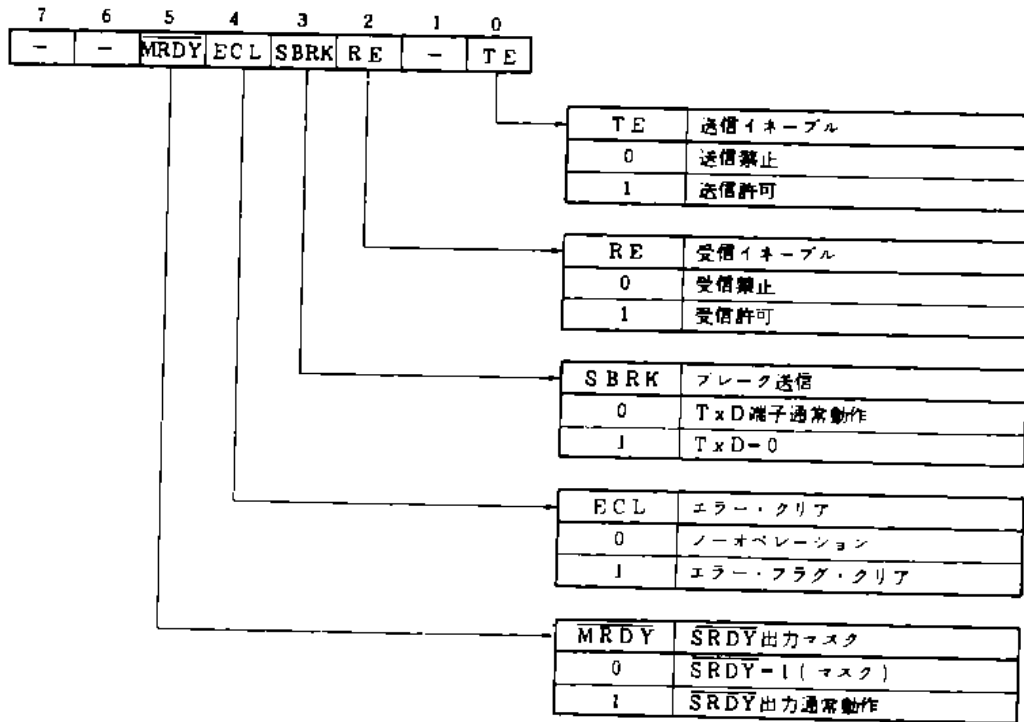
PS(Parity Select)は、パリティ・ビットの設定を行ないます。パリティ禁止にすると、送信時のパリティ・ビットの付加、受信時のパリティ・チェックは行なわれません。パリティ許可の場合には偶数パリティが奇数パリティかの選択をします。

STL(Stop bit Length)は送信するストップ・ビット数の設定で、1ビットまたは2ビットの選択をします。

2.3.2 SCM (Serial Command register)

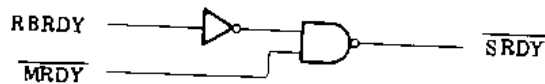
このレジスタには送受信の許可/禁止、エラー・クリアなどを行なうコマンド・ワードを設定します。

図 2-16 SCMの設定フォーマット



$\overline{\text{MRDY}}$ (Mask Ready) ビットは $\overline{\text{SRDY}}$ 端子の出力のマスクを制御します。 $\overline{\text{SRDY}}$ 出力は図 2-17 のようになっており、 $\overline{\text{MRDY}}=0$ の場合は常にハイ・レベルを出力し、送信側に対してデータの送信の禁止を示します。 $\overline{\text{MRDY}}=1$ の場合は RBRDY 信号を出力します。

図 2-17 $\overline{\text{SRDY}}$ 信号



ECL (Error Clear) ビットを 1 にするとステータス・レジスタの中のエラー・フラグ (PE, OVE, FE) がクリアされ 0 になります。 受信許可 (RE=1) にするときは同時に ECL=1 としてエラー・フラグをクリアする必要があります。

SBRK (Send Break) ビットはブレイク状態の送信に用います。 SBRK=1 とすると、データの送信状態に関係なく TxD 端子はロウ・レベルになりブレイク送信を行ないます。ブレイク状態を解除するには SBRK=0 とします。 なお、この機能は送信禁止中 (TE=0) でも有効です。

RE (Receiver Enable) ビットは受信の許可/禁止を制御します。 RE=1 で受信許可、

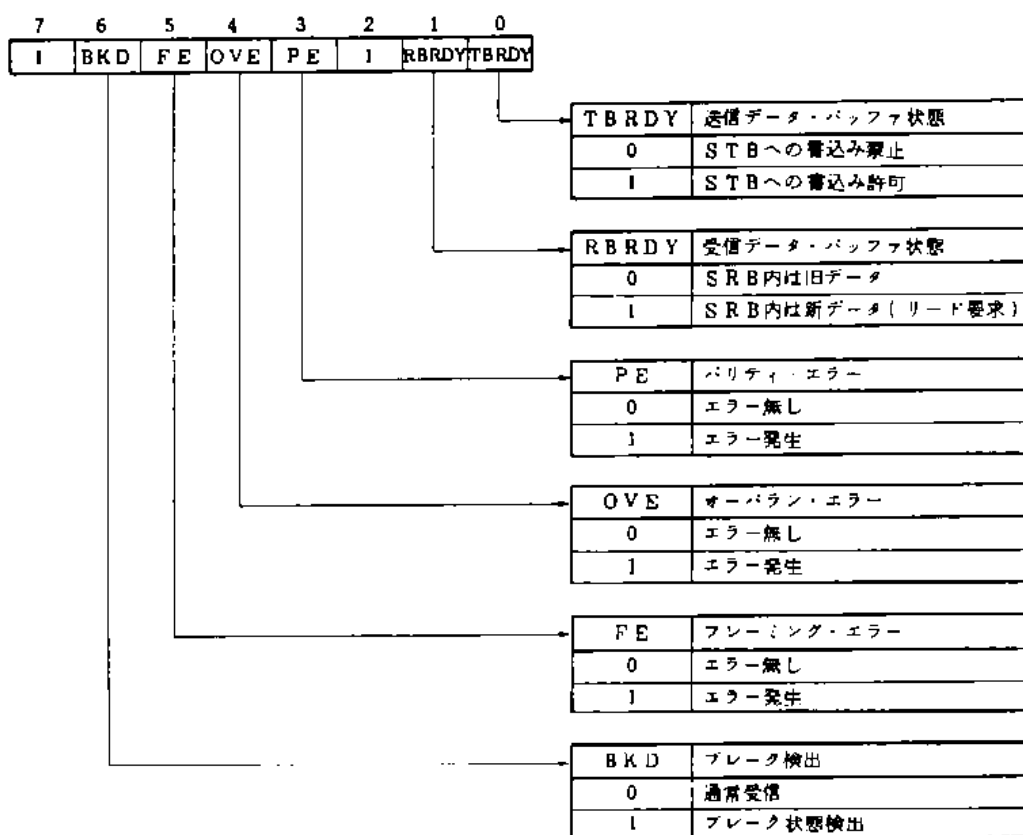
RE=0で受信禁止となります。

TE(Transmitter Enable)ビットは送信の許可/禁止を制御します。TE=1で送信許可、TE=0で送信禁止となります。送信禁止の場合、TE=0とする前にCPUから蓄込まれたデータをすべて送出してから送信を停止し、TxD端子はハイ・レベル(マーク状態)になります。

2.3.3 SST(Serial Status register)

このレジスタは、SCUのステータスを格納しているレジスタです。ステータスにはエラー状況、データ・バッファSRB、STBの状態、ブレーク検出等があります。

図2-18 SSTのフォーマット



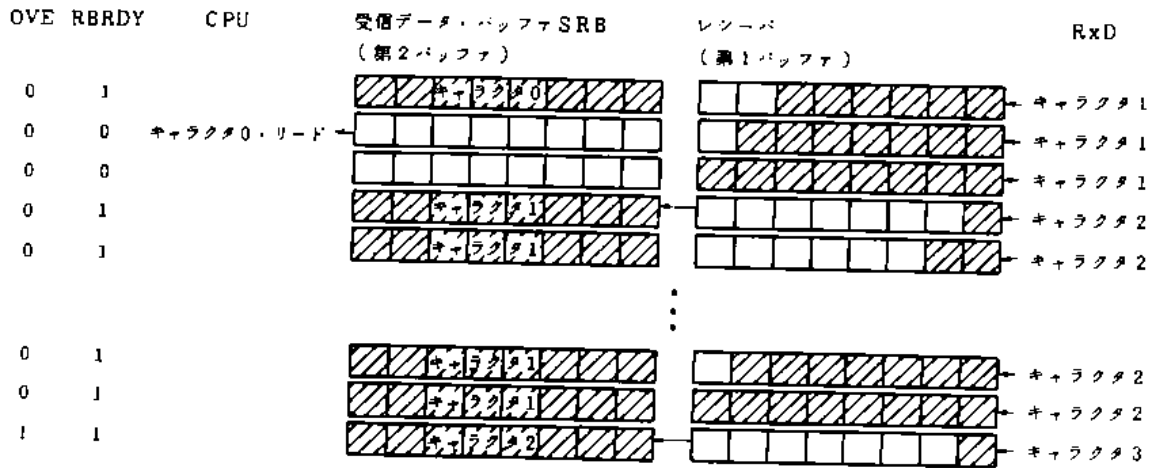
BKD(Break Detect)ビットは、ブレーク状態を検出したときに1になります。ブレーク状態とは、常にロウ・レベル状態のことですが、この検出はスタート・ビットからストップ・ビットの位置まで連続してロウ・レベルが入力されたときに発生します。

一度セットされたBKDビットはRxD入力がハイ・レベルに戻るか、またはRESET 入力が入力にならない限りクリアされません。

FE(Framing Error)ビットは、ストップ・ビットがあるべきタイミングにハイ・レベルが検出されなかったときに'1'となり、フレーミング・エラー発生を示します。フレーミング・エラーは、ブレーク状態の受信時に発生するほか、送信側と受信側のクロックがずれている場合や、伝送路中でデータが変化するときなどに発生します。

OVE(Overrun Error)ビットは受信時にCPUの受信データの脱出しが遅れたときに'1'となり、オーバーラン・エラー発生を示します。このとき、SRBの旧データが新データの書き込みによって失われます。

図2-19 オーバラン・エラー



キャラクター1はCPUに読まれることなくキャラクター2の受信によって失われます。

PE(Parity Error)ビットはパリティが有効なときの受信時にパリティ・チェックでエラーが発生すると'1'になります。

以上三つのエラー・ビットはエラー・クリア・コマンド(SCMのECLビットを1にする)を発行することによりクリアされます。

RBRDY(Receive data Buffer Ready)ビットは、レシーバが1キャラクター分のデータを受信し、そのデータが受信データ・バッファSRBに転送されたとき、すなわち受信データをCPUが読み出し可能になったときに1となります。そしてCPUがSRBを読出したときにRBRDYは0にクリアされます。

RBRDY=1となってから、次の1キャラクターが受信されてSRBに転送されるまでにCPUが受信データの脱出しを完了していなければ、オーバーラン・エラーが発生してステータスのOVEビットがセットされます。

TBRDY(Transmit data Buffer Ready)ビットは送信データ・バッファSTBが空であるとき、すなわちCPUからの送信データの書き込みが可能なときに1となります。CPUがSTBに送信データを書き込むと、TBRDYは0にクリアされます。

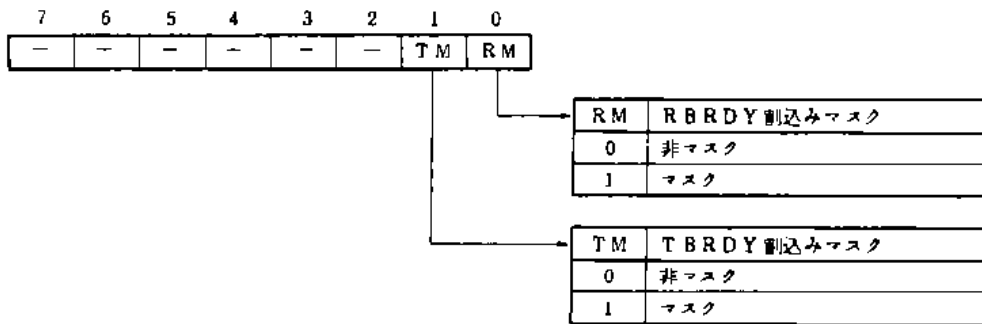
TBRDY=0の状態では送信データを書き込むと、STB内にあるまだ送出されていないデータは破壊されてしまいます。

コマンドによって送信禁止にされるとTBRDYは1にはなりません。

このレジスタはリセット時にビット7、ビット2以外が0に初期化されます。

2.3.4 SIMK (Serial Interrupt Mask register)

図 2-20 SIMK の設定フォーマット



SIMKはSCUからの割込み要求の原因となるRBRDYとTBRDYを独立にマスクします。RM(RBRDY interrupt Mask)ビット、TM(TBRDY interrupt Mask)ビットは'1'になると、それぞれRBRDY=1、TBRDY=1となっても割込みは発生しません。このレジスタはリセットによってRM=1、TM=1に初期化されます。

2.4 ICU (割込みコントロール・ユニット)

ICUの内部レジスタはアドレス信号^注によって選択されます。しかし、アドレス信号は「本しか使用していないため、同じアドレスに複数のレジスタが割り当てられています。

注 μ PD70208, 70208(A), またはSCTLのIOAGビットを1に設定した μ PD70208H, 70216Hの場合: A0
 μ PD70216, 70216(A), またはSCTLのIOAGビットを0に設定した μ PD70208H, 70216Hの場合: A1

表 2-5 ICUレジスタ・アドレッシング

	アドレス	その他の条件	動作	
リ ド	0	IMDでIRQを設定	CPU←IRQデータ	
		IMDでIISを設定	CPU←IISデータ	
		ポーリング・フェーズ	CPU←ポーリング・データ	
	1	-	CPU←IMKW	
ラ イ ト	0	D4=1	CPU→IIW1	
		(D4=0)・(D3=0)	CPU→IPFW	
		(D4=0)・(D3=1)	CPU→IMDW	
	1	イニシャライズ・シーケンス		CPU→IIW2
				CPU→IIW3
				CPU→IIW4
	イニシャライズの終了後	CPU→IMKW		

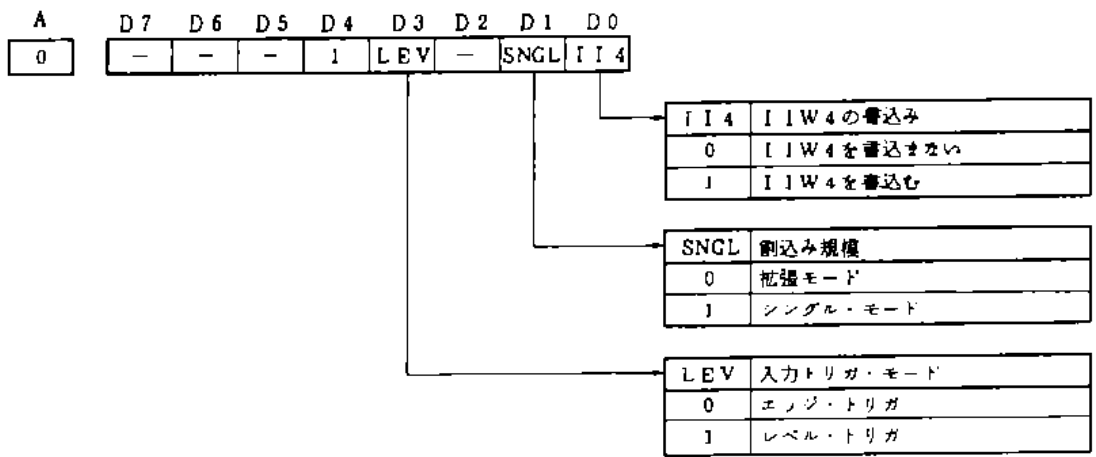
ICUはリセットによって初期化を行ないませんので電源投入後は必ず初期化プログラムを実行します。

2.4.1 イニシャライズ&コマンド・ワード・レジスタ群

これらのレジスタは、イニシャライズ・ワードのIIW1-IIW4, コマンド・ワードのIPFW (Interrupt Priority and Finish Word)およびIMDW(Interrupt Mode Word)を格納しています。読出しは出来ません。

- (1) IIW1(Interrupt Initialize Word1)

図 2-21 I1W1のフォーマット

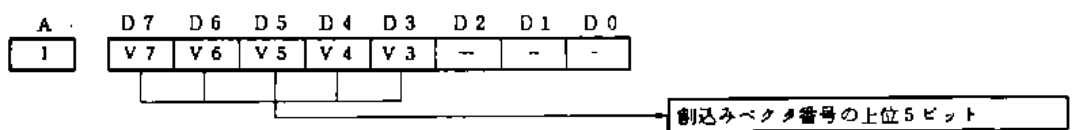


LEV(Level trigger mode)ビットは、INTP端子からの入力とSCUからの入力のトリガ・モードを設定します。LEV=1のときにレベル・トリガ、LEV=0のときに立上がりエッジ・トリガとなります。TCUからの要求(INTL0およびTCUに接続された場合のINTL2)はこの設定に関係なく常に、立上がりエッジ・トリガとなります。

SNGL(Single mode)ビットは、割込みシステムの規模を指定します。SNGL=1の場合はμPD70208、μPD70216内蔵のICUだけを使用するシングル・モード、SNGL=0の場合は外部に1~7個のμPD71059をカスケード接続して、割込み要求を最大57本にまで拡張できる拡張モードを設定します。

(2) I1W2(Interrupt Initialize Word)

図 2-22 I1W2のフォーマット



V7-V3は、割込みアドレスを生成するためにICUがCPUへ与える8ビット割込みベクタ番号の上位5ビットです。ICUはCPUに受け付けられた割込みレベルに応じて下位3ビットを追加し、8ビット・ベクタ番号を発生しCPUに与えます。

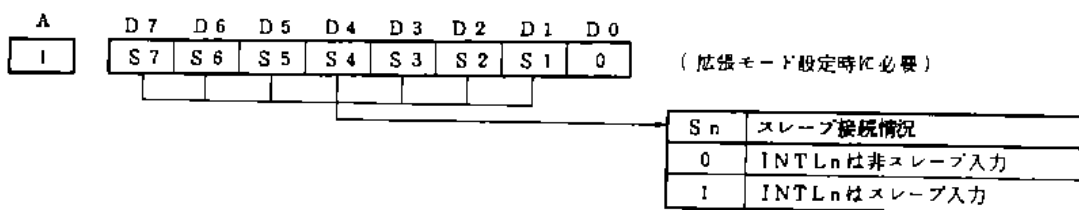
CPUはICUから受け取ったベクタ番号を4倍して、該当する割込みベクタ・テーブルのアドレスを生成し、テーブルを参照して目的の割込みルーチンを実行します。

表 2-6 割込みベクタ番号の発生

割込みレベル	D7	D6	D5	D4	D3	D2	D1	D0
INTL0	V7	V6	V5	V4	V3	0	0	0
INTL1	V7	V6	V5	V4	V3	0	0	1
INTL2	V7	V6	V5	V4	V3	0	1	0
INTL3	V7	V6	V5	V4	V3	0	1	1
INTL4	V7	V6	V5	V4	V3	1	0	0
INTL5	V7	V6	V5	V4	V3	1	0	1
INTL6	V7	V6	V5	V4	V3	1	1	0
INTL7	V7	V6	V5	V4	V3	1	1	1

(3) IIW3 (Interrupt Initialize Word3)

図 2-23 IIW3のフォーマット

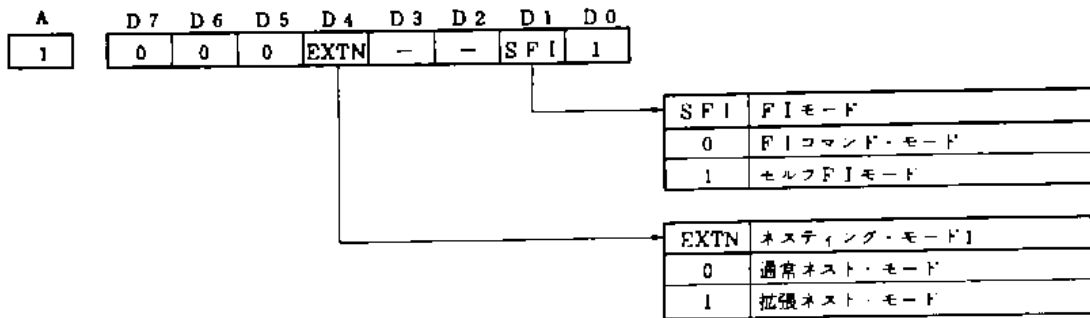


このワードは拡張モードの場合にのみ必要でかつ意味を持ちます。

IIW3のS7-S1の7ビットはINTL7-1への入力がそれぞれスレーブが接続されているか、接続されていないかを設定します。たとえばS7=1の場合、ICUはINTL7への割込み要求をスレーブからの要求と判断し、A10-A8端子に7を出力し割込みベクタ番号の出力は行ないません。ICUの代わりにINTP7端子にカスケード接続されているスレーブのμPD71059が割込みベクタ番号の出力を行なうからです。S7=0の場合は、INTL7からの要求に対してICUは必要なすべての動作を行ないます。

(4) IIW4(Interrupt Initialize Word4)

図 2-24 IIW4のフォーマット

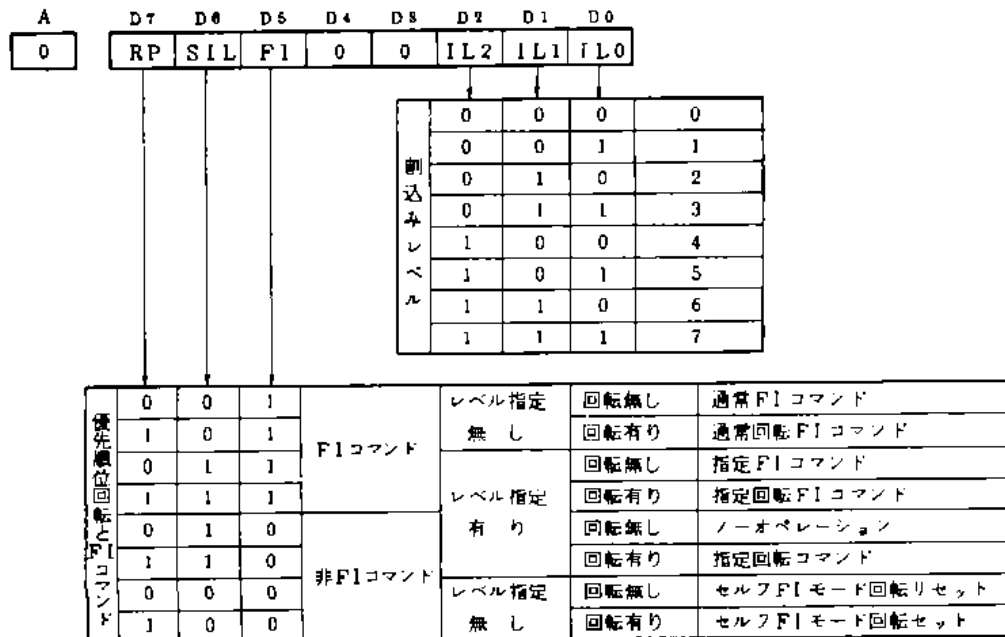


EXTN(Extended Nesting mode)ビットはネスティングのモードを設定します。EXTN=0の場合は通常ネスティング・モード、EXTN=1の場合は拡張ネスティング・モードとなります。拡張ネスト・モードは拡張モードのときに使用します。

SFI(Self Finish Interrupt)ビットはFIモードの設定をします。SFI=0の場合は、CPUがICUにFIコマンドを発行して割込み処理の終了を宣言するFIコマンド・モードに設定されます。SFI=1の場合はセルフFIモードが設定され、CPUからはFIコマンドを発行する必要はありません。

(5) IPFW(Interrupt Priority and Finish Word)

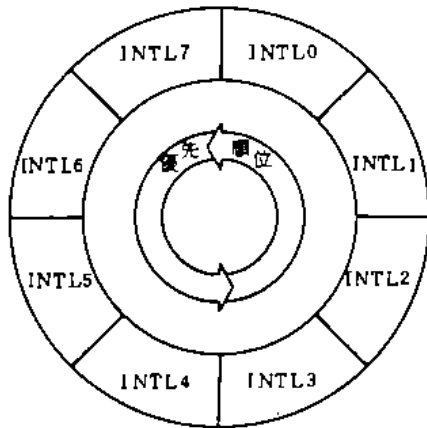
図 2-25 IPFWのフォーマット



このワードは、割り込み処理（ルーチン）の終了を宣言するコマンドや、要求の優先順位を変更するコマンドを発行します。

RP (Rotate Priority) ビットは 8 本の要求の優先順位を変更（回転）するためのビットで、RP=1 のときに変更させます。ICU の持つ 8 本の割り込み要求の優先順位はリング状になっており、任意の 1 レベルを最低優先に設定すると残りの 7 レベルの優先順位は一意的に決定されます（初期化終了時は INTL7 が最低優先）。RP=1 とすることにより最低優先となるレベルを変更することができます。

図 2-26 INTL 要求の優先順位



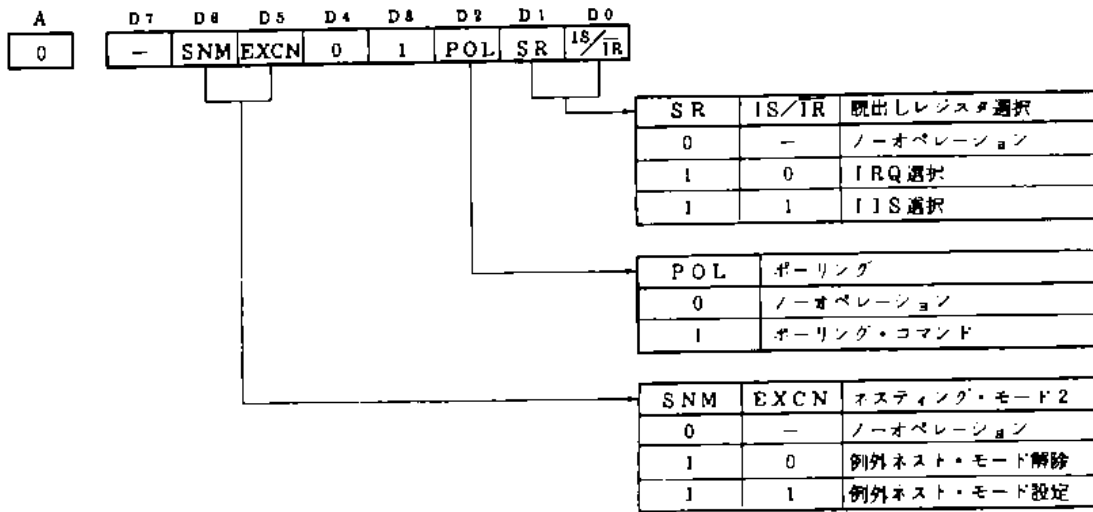
SIL (Specify Interrupt Level) ビットは優先順位の変更、FI コマンドに対して割り込みレベルを指定するとき 1 にします。SIL=1 の場合、IL2-IL0 ビットで表わされるレベルが最低優先レベル (RP=1 の場合) や終了割り込みレベル (FI=1 の場合) に指定されます。

FI (Finish Interrupt) ビットを 1 にすると FI コマンドとなります。

IL2-IL0 (Interrupt Level) ビットは SIL=1 の場合に、指定する割り込みレベル (7-0) を示します。

(6) IMDW (Interrupt Mode Word)

図 2-27 IMDW (Interrupt Mode Word)



このワードは、読出しレジスタの選択、例外ネスト・モードの設定/解除、ポーリングを行なうときに用います。

SNM (Set Nesting Mode) ビットと EXCN (Exceptional Nesting Mode) ビットはペアで用いられ、例外ネスト・モードの設定/解除を行ないます。SNM=0 の場合は EXCN ビットは無視され何の動作もしません。SNM=1 として、EXCN = 1 ならば例外ネスト・モードの設定、EXCN = 0 ならば例外ネスト・モードの解除を行います。

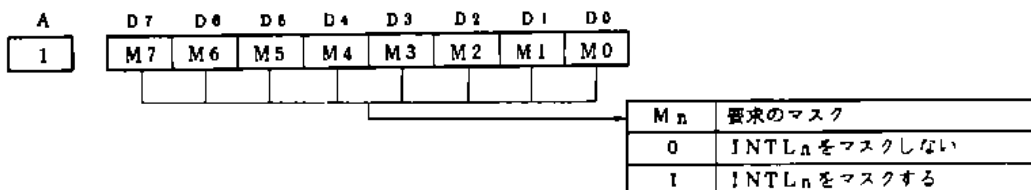
POL (Polling) ビットはポーリング動作をするときに 1 とします。

SR (Set Register) ビットと IS/IR (In Service/Interrupt Request) ビットはペアで用いられ読出すレジスタの選択を行ないます。SR=0 の場合は IS/IR ビットは無視され何の動作もしません。SR=1 として、IS/IR = 1 ならば IIS、IS/IR = 0 ならば IRQ が読出すレジスタとして選択されます。

2.4.2 IMK (Interrupt Mask register)

IMK (Interrupt Mask register) は、マスク・ワードを設定するリード/ライト可能なレジスタです。

図 2-28 IMKW (Interrupt Mask Word)



このIMKWはIRQをマスクして該当するINTLn入力からの割込み要求を禁止します。ビットnが1ならばINTLnへの要求はマスクされて受け付けられることはありません。また、例外ネスト・モードが設定されている場合はIISもマスクします。

2.4.3 IRQ (Interrupt Request register)

IRQ (Interrupt Request register) は、8本の割込み入力の状態を持っています。IRQのビットnが1ならばINTLnに要求があることを示します。

IRQの情報は優先決定ロジックへ伝えられますが、IMKによって任意のビットがマスクされます。CPUからのIRQの読出しの場合はIMKによってマスクされることはありません。

2.4.4 IIS (Interrupt In Service register)

IIS (Interrupt In Service register) は、8本の割込みに対するCPUのサービス状況を示す情報を持っています。IISのビットnが'1'ならばINTLnに対する割込みルーチンがサービス中であることを示します。

IISの情報は優先決定ロジックへ伝えられますが、例外ネスト・モードが設定されている場合はIMKによって任意のビットがマスクされます。CPUからのIISの読出しの場合はIMKによってマスクされることはありません。

2.4.5 各種モード、コマンド

ICUはイニシャライズおよびコマンドによって様々なモードに設定されます。ここではモードとコマンドについて説明します。

(1) 割込みシステム規模に関するモード

ICUは割込みシステムの規模に応じてシングル・モードか拡張モードのどちらかに設定されます。

① シングル・モード

割込み要求が8本以下の時に使用します。

② 拡張モード

割込み要求が9本以上の時に使用します。μPD71059を最大7個までカスケード接続し、57割込み要求入力を可能とします。

スレーブに入力された要求の場合、動作はスレーブがベクタ番号を与えますのでICUはベクタ番号を出力すべきスレーブを指定します。

(2) ネスティングに関するモード

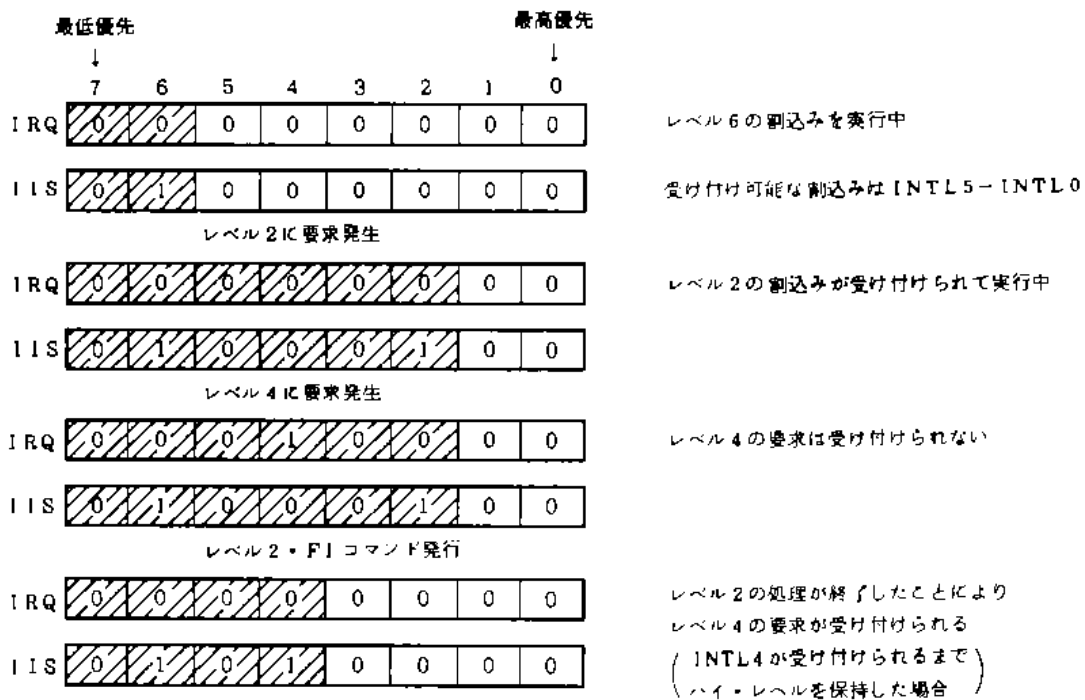
ICUは、割込みの多重化のためにネスティングの方法を変えることができます。これらは

① 通常ネスト・モード

このモードはIIW4を書込まない場合、またはIIW4でEXTN=0とした場合に設定されます。

このモードでは、ある割込みが実行中である(IISの該当ビットが'1')間は、そのレベルより優先順位が高い割込み要求しか受け付けません。

図 2-29 通常ネスト・モード



② 拡張ネスト・モード

このモードは、拡張モードのときに意味を持ちます。

あるスレーブの一つの割込みがサービス中の場合に、同じスレーブから、より優先順位の高い割込み要求は出されても、ICUから見れば同じレベルなので通常ネスト・モードでは受け付けられません。そこで拡張ネスト・モードを設定すると、スレーブからの割込み要求の場合に限り同じレベルでも受け付けられるようになり、完全なネスティング動作が可能になります。

拡張モードでFIコマンドを発行する場合、まずスレーブに対してFIコマンドを発行します。次にスレーブのIISを読み出し、そのスレーブにまだサービス中の割込みがあるかどうか調べます。そして、サービス中の割込みが無かったらICUに対してFIコマンドを発行します。内部周辺からの割込みの場合はICUにのみFIコマンドを発行します。

③ 例外ネスト・モード

通常ICUはサービス中の割込みレベルより低い優先順位の割込み要求は受け付けられません。しかし場合によっては低いレベルの割込み要求を優先させたい場合があります。このような場合に例外ネスト・モードを設定します。

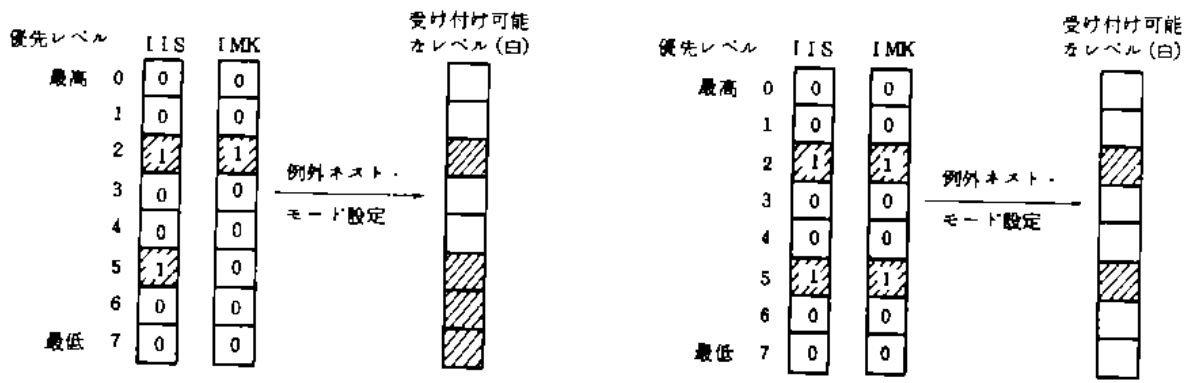
例外ネスト・モードでは割込みマスク・レジスタIMKがマスクする対象がIRQとIISの二つになります。

例外ネスト・モードを設定するときの手順は次の手順になります。

- 1) IISを読み出す
- 2) 読出した値をIMKに書込む。

3) 例外ネスト・モードを設定する。

図 2-30 例外ネスト・モード



IMKのビット2のみを1にした場合はINTL6, INTL7の要求は受け付けられない。

IMKをISRと同じにした場合はサービス中以外のすべての要求が受け付けられる。

(3) FIコマンドと優先順位の変更

FIコマンド、優先順位の変更はすべてIPFWの番込みで行ないます。

① 通常FIコマンド

	D7	D6	D5	D4	D3	D2	D1	D0
IPFW=	0	0	1	0	0	-	-	-

通常FIコマンドを発行すると、ICUはサービス中の割込みの中で、優先順位が一番高いものに該当するIISのビットをリセットします。この動作によって最後に受け付けられた割込みが終了したものと見なされます。

割込みルーチン内で優先順位の変更を行った場合や例外ネスト・モードが設定されている場合は、通常FIコマンドでは正しい動作が行なわれませんので注意が必要です。

② 指定FIコマンド

	D7	D6	D5	D4	D3	D2	D1	D0
IPFW=	0	1	1	0	0	IL2	IL1	IL0

指定FIコマンドを発行すると、ICUはIPFWのIL2-IL0ビットによって指定されるIISのビットをリセットします。このコマンドは、通常FIコマンドが使用できない場合において特に有効です。

③ 通常回転FIコマンド

	D7	D6	D5	D4	D3	D2	D1	D0
IPFW=	1	0	1	0	0	—	—	—

通常回転 F I コマンドを発行すると、ICU は通常 F I コマンドの場合と同じ動作を行ない、さらにそのときに終了した割込みレベルを最低優先に設定します。

④ 指定回転 F I コマンド

	D7	D6	D5	D4	D3	D2	D1	D0
IPFW=	1	1	1	0	0	IL2	IL1	IL0

指定回転 F I コマンドを発行すると、ICU は指定 F I コマンドの場合と同じ動作を行ない、さらに IL2-IL0 ビットで指定される割込みレベルを最低優先に設定します。この場合、優先順位の変更は規則的ではなく、ユーザによって任意に行なわれますので、ユーザはネスティングの管理を行なう必要があります。

⑤ 指定回転コマンド

	D7	D6	D5	D4	D3	D2	D1	D0
IPFW=	1	1	0	0	0	IL2	IL1	IL0

指定回転コマンドを発行すると、ICU は IPFW の IL2-IL0 ビットによって指定される割込みレベルを最低優先に設定します。この場合もユーザはネスティングの管理をする必要があります。

⑥ セルフ F I モード

セルフ F I 回転付加	D7	D6	D5	D4	D3	D2	D1	D0
IPFW=	1	0	0	0	0	—	—	—

セルフ F I 回転非付加	D7	D6	D5	D4	D3	D2	D1	D0
IPFW=	0	0	0	0	0	—	—	—

IIW4 の SFI ビットが 1 に設定された場合、ICU はセルフ F I モードになります。このモードでは INTAK シーケンスの最初の INTAK パルスの初めに IIS の該当ビットがセットされ、その INTAK パルスの終了直後に同じ該当ビットがリセットされます。このため割込みルーチンの終了時に CPU から F I コマンドを発行する必要がなくなります。しかし、このモードでは IIS にサービス中かどうかの情報を持っていませんので、CPU が割込み許可になっていると新しく発生する割込み要求はどんどん受け付けられます（もちろん優先の高いものから）。そのため割込み要求の発生頻度が大きい場合やレベルトリガモードの場合は割込み多重あふれ（スタック領域が足りなくなる）が起こる可能性がありますので注意が必要です。

セルフ FI モードにおいて、優先順位の回転動作を付加することができます。この回転動作を付加させると、 $\overline{\text{INTAK}}$ シーケンスの終了時に IIS の該当ビットのリセットと同時に、該当の割込みレベルを最低優先に設定します。優先順位の回転動作を付加するか、しないかも IPFW によって設定されます。それぞれの IPFW のフォーマットを示します。セルフ FI モードでないときは、これらの二つのコマンドは意味を持ちません。

(4) INTP 端子入力のトリガ・モード

割込み要求入力のトリガ・モードはイニシャライズ・ワードの I I W I で設定されますが、これは外部入力端子または内部接続された S C U からの要求に対して有効となります。内部接続された T C U からの要求の場合は常に立上がりエッジ・トリガ・モードとなります。

① エッジ・トリガ・モード

エッジ・トリガ・モードでは INTP の入力は立上がりエッジによってトリガされ、I R Q の該当ビットが 1 になります。I R Q の内容は $\overline{\text{INTAK}}$ シーケンスにならないとラッチされませんので、要求に対応する CPU からの $\overline{\text{INTAK}}$ パルスが I C U に返されるまでは、INTP をハイ・レベルに保つ必要があります。次の割込み要求を発生するためには INTP を 1 度ロウ・レベルに下げてから再びハイ・レベルに立上げをしなければなりません。

② レベル・トリガ・モード

レベル・トリガ・モードでは、INTP の入力はハイ・レベルによってトリガされ、I R Q の該当ビットが 1 になります。エッジ・トリガ・モードの場合と同様に、要求に対応する CPU からの $\overline{\text{INTAK}}$ パルスが I C U に返されるまでは、INTP をハイ・レベルに保つ必要があります。

注意：I C U の CPU への要求がインアクティブなときに CPU から $\overline{\text{INTAK}}$ パルスが入力されると、I C U はレベル 7 の割込みがあったかのように動作します。この場合、I I S のビット 7 はセットされません。このため、ある INTP に要求が発生して、その要求に対する CPU からの $\overline{\text{INTAK}}$ パルスが返る前に、その INTP 入力がロウ・レベルに落ちてしまうような不完全な割込み要求が起り得る場合は、レベル 7 の処理は不完全割込み要求に対する処理をする必要があります。I I S のビット 7 で判断します。また不完全割込み要求の場合は I I S のビット 7 がセットされませんので、その処理では FI コマンドは発行してはいけません。

(5) その他のコマンド

① 内部レジスタ選択コマンド

内部レジスタの中で I R Q、I I S、I M K の三つのレジスタは CPU から読出すことができます。

I M K はアドレス^注を 1 とすることにより読出すことができますが、I R Q と I I S はどちらもアドレス = 0 として読出すためこれらを選択する必要があります。この選択は

IMDWのSR, IS/ \overline{IR} ビットによって制御します。SR=1, IS/ \overline{IR} =1とするとIISが、IS/ \overline{IR} =0とするとIRQが読出されます。

注 μ PD70208, 70208(A), またはSCTLのIOAGビットを1に設定した μ PD70208H, 70216Hの場合: A0
 μ PD70216, 70216(A), またはSCTLのIOAGビットを0に設定した μ PD70208H, 70216Hの場合: A1

② ポーリング・コマンド

ポーリング・データ

D7	D6	D5	D4	D3	D2	D1	D0
INT	0	0	0	0	PL2	PL1	PL0

ICUに対してポーリングを行なうことによりCPUは、割込みアクノリッジ・サイクルを行なうことなく現在サービスをしなければならない要求はどれかという情報を得ることができます。

ポーリングを行なう場合、CPUは割込み禁止状態になっていなければなりません。次にPOLビットを1としたIMDWによりICUへポーリング・コマンドを発行します。このコマンドにより、ICUはCPUから読出しが行なわれるまでの間ポーリング・フェーズに入ります。ポーリング・フェーズ中にアドレス=0として読出しを行えば、IRQ, IISではなくポーリング・データが読出されます。そしてICUはポーリング・フェーズを終了します。

読出されたデータのINT (Interrupt) ビットが1ならば、ICUはPL2-PL0 (Permitted Level) ビットで示される割込みレベルに該当するIISのビットをセットし、その割込みがサービスされたと見なします。CPUは、読出されたポーリング・データを基に必要な処理を行ない、その終了時にはFIコマンドを発行する必要があります。

2.5 DMAU (DMAコントロール・ユニット)

DMAUは内部に24個のレジスタを持っています。またこれらのレジスタをリード/ライトするためのコマンドとDMAUを初期化するためのコマンドを持っています。

2.5.1 レジスタ

表 2-7 レジスタ構成

レ ジ ス タ 名	ビット・サイズ
チャンネル・レジスタ (DCH)	5-bit
ベース・アドレス・レジスタ (DBA)	20-bit×4
カレント・アドレス・レジスタ (DCA)	20-bit×4
ベース・カウント・レジスタ (DBC)	16-bit×4
カレント・カウント・レジスタ (DCC)	16-bit×4
モード・コントロール・レジスタ (DMD)	7-bit×4
デバイス・コントロール・レジスタ (DDC)	5-bit
ステータス・レジスタ (DST)	8-bit
マスク・レジスタ (DMK)	4-bit

(1) アドレス・レジスタ

アドレス・レジスタは20ビットのDMAアドレスを格納するレジスタで、各チャンネルごとにベース・アドレスとカレント・アドレスの二つのレジスタがあります。カレント・アドレスは、1回の転送(バイト/ワード)ごとに更新され常に次の転送アドレスを示しています。これに対してベース・アドレスはCPUによって置換えられるまでは変化せず、オートイニシヤライズ時に新たなサービスのDMA開始アドレスとしてカレント・アドレス・レジスタへコピーされます。

(2) カウント・レジスタ

カウント・レジスタはDMA転送回数を格納する16ビットのレジスタで、各チャンネルごとにベース・カウントとカレント・カウントの二つのレジスタがあります。カレント・カウントは、1回の転送(バイト/ワード)ごとに1だけデクリメントされ、残りの転送回数を示しています。また、0からアンダフローした時点でターミナル・カウントが発生し指定された回数の転送が終了したことを示します。ベース・カウントはCPUによって置換えられるまでは変化せず、オートイニシヤライズ時に次のサービスの転送回数としてカレント・カウント・レジスタへコピーされます。

(3) コントロール・レジスタ群

チャンネル、デバイス・コントロール、ステータス、モード・コントロール、マスクの5種類のレジスタはDMAUの動作モード等を制御したり、各チャンネルのステータスを示します。

2.5.2 コマンド

DMAUのコマンドは下位4ビットA3-A0によって選択されます。

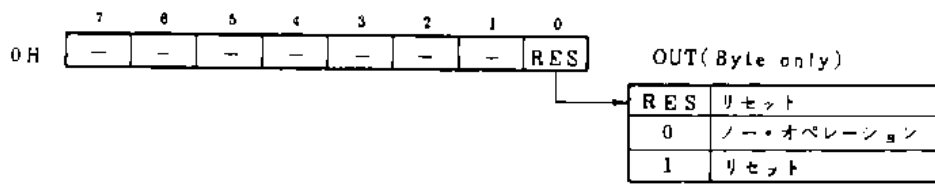
表2-8 コマンド構成

アドレス	R/W	コマンド名	MSB	フォーマット				LSB		
0H	R		禁 止							
	W(B)	イニシャライズ	-	-	-	-	-	-	-	RES
1H	R(B)	チャンネル・レジスタ・リード	-	-	-	BASE	SEL3	SEL2	SEL1	SEL0
	W(B)	チャンネル・レジスタ・ライト	-	-	-	-	-	BASE	SELCH	
2H	R/W	カウント・レジスタ	C7	C6	C5	C4	C3	C2	C1	C0
3H	R/W	・リード/ライト	C15	C14	C13	C12	C11	C10	C9	C8
4H	R/W	アドレス・レジスタ	A7	A6	A5	A4	A3	A2	A1	A0
5H	R/W	タ・リード/ライ	A15	A14	A13	A12	A11	A10	A9	A8
6H	R/W(B)	ト	-	-	-	-	A19	A18	A17	A16
7H	-		禁 止							
8H	R/W	デバイス・コントロール・レジスタ・リード/ライト	-	-	EXW	ROT	-	DDMA	-	-
			-	-	-	-	-	-	WEV	BHLD
0AH	R/W(B)	モード・コントロール・レジスタ・リード/ライト	TMODE		ADIR	AUTI	TDIR		-	W/R
0BH	R(B)	ステータス・レジスタ・リード	RQ3	RQ2	RQ1	RQ0	TC3	TC2	TC1	TC0
	W		禁 止							
0CH	-		禁 止							
0DH	-		禁 止							
0EH	-		禁 止							
0FH	R/W(B)	マスク・レジスタ・リード/ライト	-	-	-	-	M3	M2	M1	M0

表2-8で '(B)' の印がついているコマンドは、バイト IN/OUT命令で発行しなければならぬものです。

- (1) イニシャライズ・コマンド

図 2-31 イニシャライズ・コマンド



このコマンドはDMAUの初期化を行います。このコマンドの発行はバイトOUT命令によって行なわなければなりません。

表 2-9 初期化されたレジスタの状態

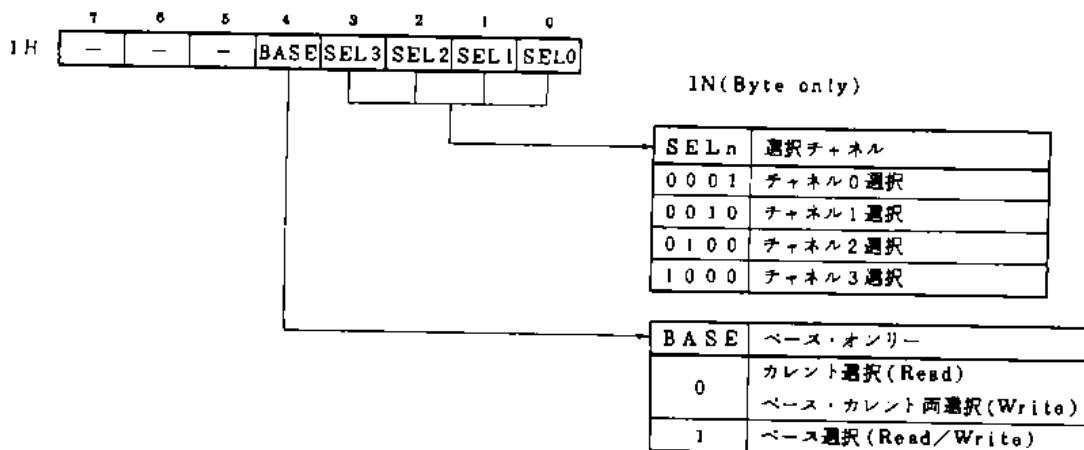
レジスタ名	初期化内容																
アドレス・レジスタ	変化なし																
カウント・レジスタ	変化なし																
チャンネル・レジスタ	<table border="1"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>-</td><td>-</td><td>-</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td> </tr> </table> (CH0選択)	7	6	5	4	3	2	1	0	-	-	-	0	0	0	0	1
7	6	5	4	3	2	1	0										
-	-	-	0	0	0	0	1										
モード・コントロール・レジスタ	全ビット・クリア																
デバイス・コントロール・レジスタ	全ビット・クリア																
ステータス・レジスタ	全ビット・クリア																
マスク・レジスタ	全ビット・セット(全チャンネル・マスク)																

② チャンネル・レジスタ・リード/ライト・コマンド

このコマンドは、四つあるDMAチャンネルからCPUによるプログラミングの対象となる一つのチャンネルを選択するチャンネル・レジスタのリード/ライトを行います。このコマンドの読出し書込みはバイトIN/OUT命令によって行なわなければなりません。

① チャンネル・レジスタ・リード

図 2-32 チャンネル・レジスタ・リード



• ビット 3 - 0 (Select)

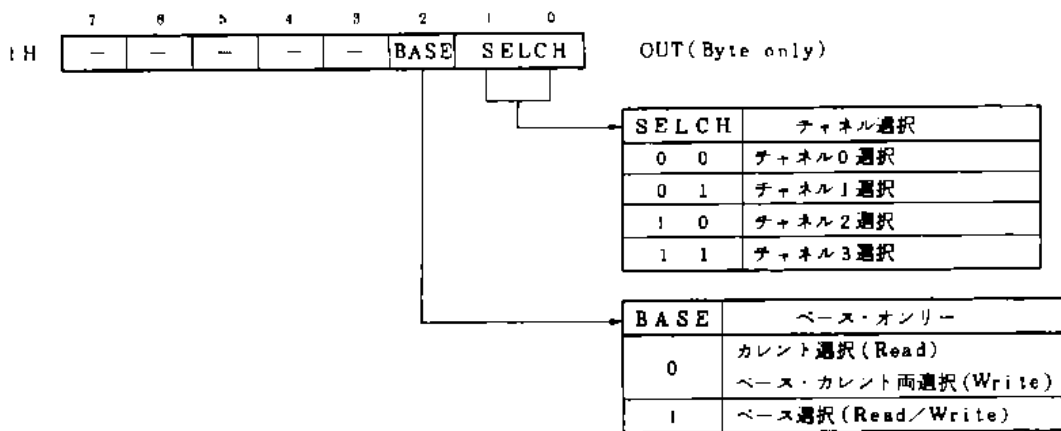
現在のアドレス/カウント/モード・コントロールの各レジスタへのリード/ライトがどのチャンネルに設定されているかを示しています。

• ビット 4 BASE (Base only)

このビットがセットされているとアドレス/カウント・レジスタへのリード/ライトはベース・レジスタにのみ行われます。反対にリセットされていると、読出しはカレント・レジスタから、書込みはベース/カレントの両方のレジスタに対して行われます。

② チャンネル・レジスタ・ライト

図 2 - 3 3 チャンネル・レジスタ・ライト



• ビット 1 - 0 (Select Channel)

アドレス/カウント/モード・コントロールの各レジスタへのリード/ライトをどのチャンネルにするかを選択します。

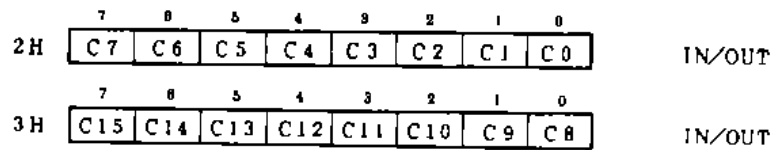
• ビット 2 (Base only)

アドレス・レジスタとカウント・レジスタは、各チャンネルごとにベース・レジスタとカレント・レジスタの2組のレジスタから成り立っています。このビットはベース・レジスタとカレント・レジスタの選択に用います。BASE = 1 の場合、アドレス/カウント・レジスタへのリード/ライトはベース・レジスタにのみ行われます。

BASE = 0 の場合は、読出されるのはカウント・レジスタになりますが、書込みはベース/カレントの両レジスタに行われます。

(3) カウント・レジスタ・リード/ライト・コマンド

図 2-34 カウント・レジスタ・リード/ライト

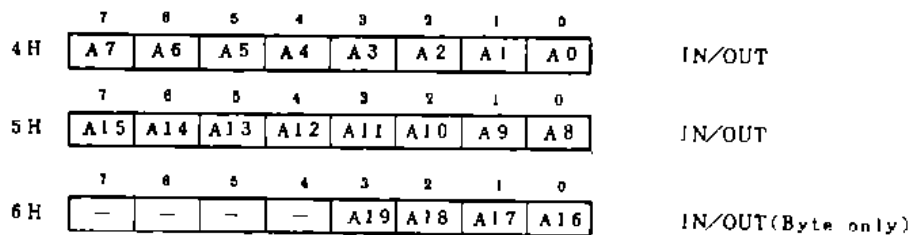


このコマンドの発行は、ワード IN/OUT 命令で行なうことができます。アクセスされるカウント・レジスタはチャンネル・レジスタにより決定されます。

ベース・カウント・レジスタは設定された値を新たな設定が行なわれるまで保持し、オート・イニシャライズ時にはこの内容がカレント・カウント・レジスタに転送されます。カレント・カウント・レジスタは1回のDMA転送ごとに1だけカウント・ダウンされます。

(4) アドレス・レジスタ・リード/ライト・コマンド

図 2-35 アドレス・レジスタ・リード/ライト

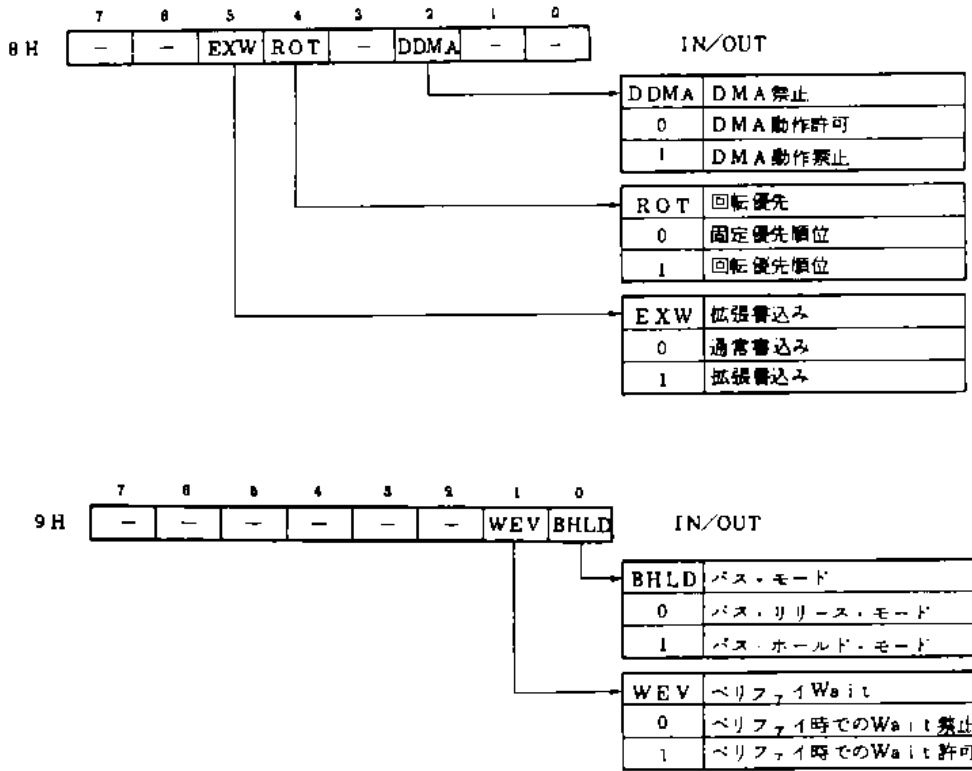


このコマンドの発行は、レジスタの下位2バイト(4H、5H)についてはワード IN/OUT 命令ができますが、上位バイト(6H)に対してはバイト IN/OUT 命令で行なわなければなりません。アクセスされるアドレス・レジスタはチャンネル・レジスタによって決定されます。

ベース・アドレス・レジスタは設定された値を新たな設定が行なわれるまで保持し、オート・イニシャライズ時にはこの内容がカレント・アドレス・レジスタに転送されます。カレント・アドレス・レジスタは1回のDMA転送ごとに、ワード転送時は±2、バイト転送時は±1更新されます。

(5) デバイス・コントロール・レジスタ・リード/ライト・コマンド

図 2-36 デバイス・コントロール・レジスタ・リード/ライト



このコマンドは、すべてのDMAチャネルに適用されるモードや、DMA動作の許可/禁止などを制御するデバイス・コントロール・レジスタのリード/ライトを行ないます。

このコマンドの発行は、ワードIN/OUT命令で行なうことができます。

・ビット2 (Disable DMA)

このビットをセットするとDMA動作は禁止されます。具体的にはDDMA=1の間BAUに対するバス要求信号を出力しなくなります。この機能は、レジスタのプログラムの途中で誤ったDMA動作が実行されるのを防ぐために用いられます。

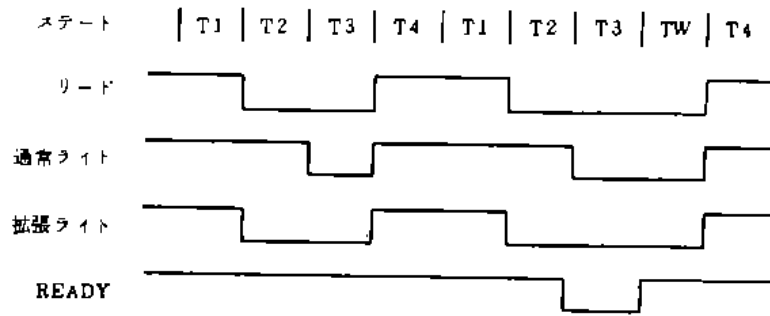
・ビット4 (Rotate Priority)

このビットをセットすると回転優先順位になります。

・ビット5 (Extended Write Timing)

このビットはライト信号の出力タイミングを制御するために用います。EXW=0のときライト信号はT3, TW間アクティブとなり(通常書込み)、EXW=1のときはリード信号と同じくT2, T3, TW間でアクティブとなります(拡張書込み)。

図 2-37 拡張ライト・タイミング



・ビット0 (アドレス 9H) (Bus Hold Mode)

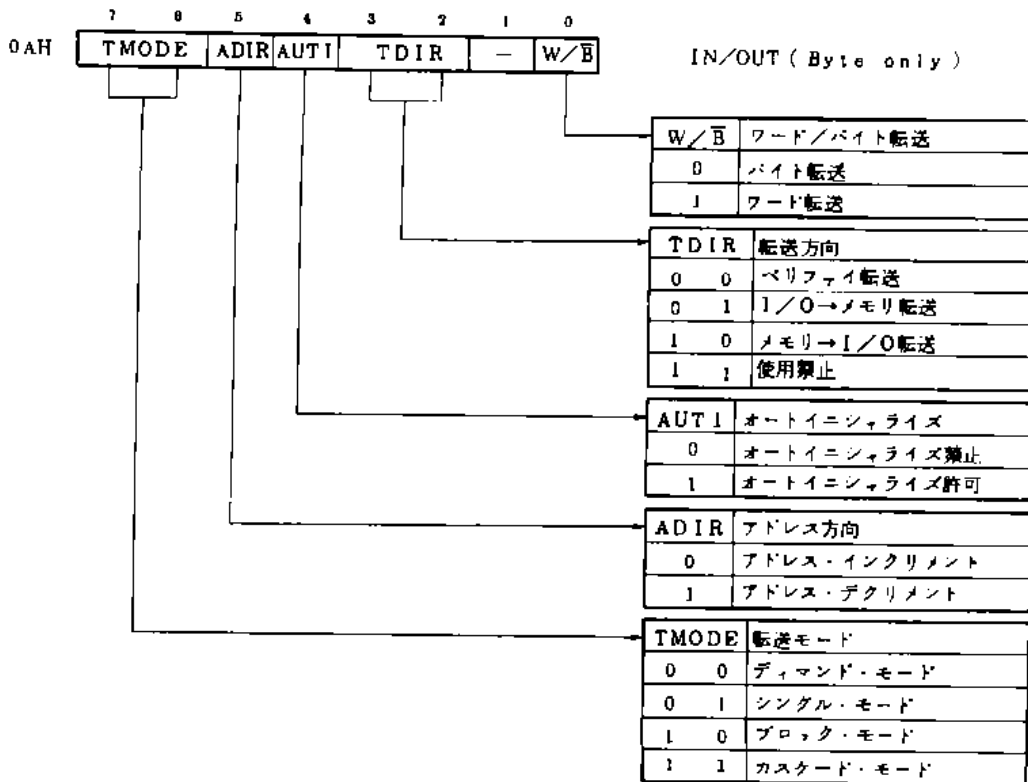
このビットによってDMA転送のバス・モードを選択します。BHL D=1の場合はバス・ホールド・モード、BHL D=0の場合はバス・リリース・モードとなります。

・ビット1 (アドレス 9H) (Wait Enable at Verify)

このビットはベリファイ転送時のREADY信号またはWCUによって発生するウェイト・ステートの挿入の許可/禁止を制御します。WEV=0の場合にウェイト・ステートの挿入は禁止されます。

(6) モード・コントロール・レジスタ・リード/ライト・コマンド

図 2-38 モード・コントロール・レジスタ・リード/ライト



このコマンドは、各チャネルごとの動作モードを規定するモード・コントロール・レジスタのリード/ライトを行います。

アクセスされるモード・コントロール・レジスタはチャネル・レジスタによって決定されます。

このコマンドの発行はバイト IN/OUTで行わなければなりません。

・ビット 0 (Word/Byte Transfer)

このビットは #PD70216 の場合にのみ意味を持ち、DMA 転送をワード転送で行なうか、バイト転送で行なうかを指定します。W/B=1 の場合にワード転送となります。

・ビット 3-2 (Transfer Direction)

これらのビットはメモリー I/O 間の DMA 転送の転送方向を決定します。

・ビット 4 (Auto-initialize)

このビットをセットすることによりオートイニシャライズが許可されます。

・ビット 5 (Address Direction)

このビットはカレント・アドレス・レジスタの更新する方向を決定します。

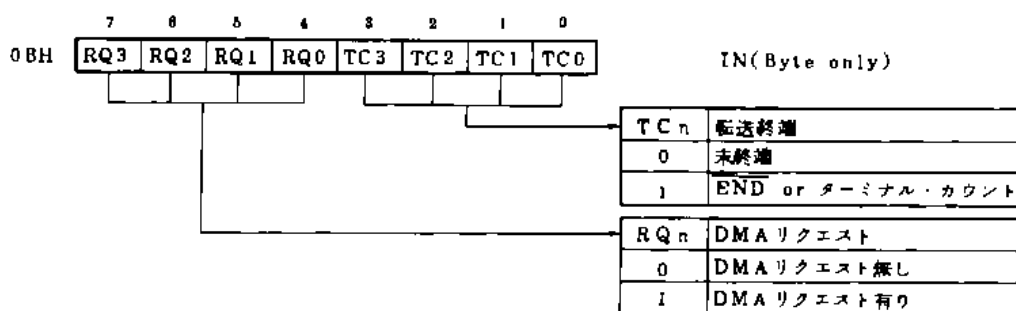
ADIR=0 ならば増加方向に更新 (ワード転送時は、+2+1; バイト転送時) し、ADIR=1 ならば減少方向に更新 (ワード転送時は、-2-1; バイト転送時) します。

・ビット 7-6 (Transfer Mode)

これらのビットはメモリー I/O 間の DMA 転送の転送モードを決定します。

(7) ステータス・レジスタ・リード・コマンド

図 2-39 ステータス・レジスタ・リード



このコマンドは、各チャネルの DMA 要求状態、ターミナル・カウントまたは $\overline{\text{END}}$ 入力の発生状態の情報を持っているステータス・レジスタを読出します。

このコマンドの発行はバイト命令で行わなければなりません。

・ビット 3-0 (Terminal Count)

これらのビットは各 DMA チャネルのターミナル・カウントまたは $\overline{\text{END}}$ 入力を示

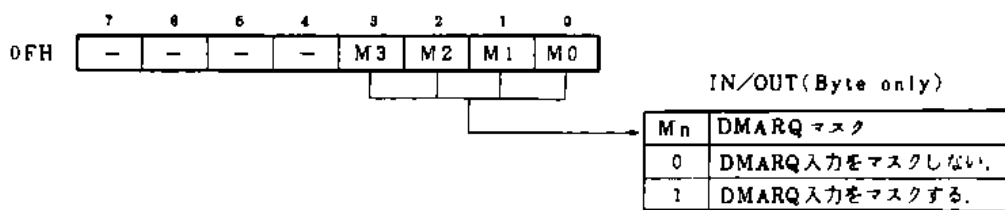
しています。DMAサービス中にターミナル・カウントが発生するか、END 入力があれば該当のビットがセットされます。このコマンドによって読出すごとに、これらのビットはクリアされます。

・ビット7-4 (DMA Request)

これらのビットは各DMAチャンネルのDMAリクエストの状況を示しています。マスク状態に関係なく各DMARQ端子の状態(アクティブなら1)を示していますので、マスクによって保留されているDMA要求を調べることができます。

(8) マスク・レジスタ・リード/ライト・コマンド

図 2-40 マスク・レジスタ・リード/ライト



このコマンドは、DMARQ3-DMARQ0入力によるDMAリクエストのマスクを制御するマスク・レジスタへのリード/ライトを行います。

このコマンドの発行はバイト IN/OUT 命令で行なわなければなりません。

・ビット3-0 (Mask DMARQ input)

これらのビットは4本のDMAリクエスト入力端子DMARQ3-0の入力をマスクするかしないかを決定します。MnビットをセットするとDMARQn端子の入力をマスクしてそのチャンネルへのDMAリクエストを禁止します。

第3章 応用例

3.1 タイマ割込みを用いた時間カウント

3.1.1 概要

μPD70208, μPD70216 に内蔵されるTCUとICUを用いて1/1000秒単位のカウントを行ないます。CPU Clock(5MHz)を、内蔵プリスケアラにより8分周し、この出力をTCUのカウンタ#0により更に625分周し1kHz(周期: $t_{cyc}=1ms$)のパルスが発生させます。

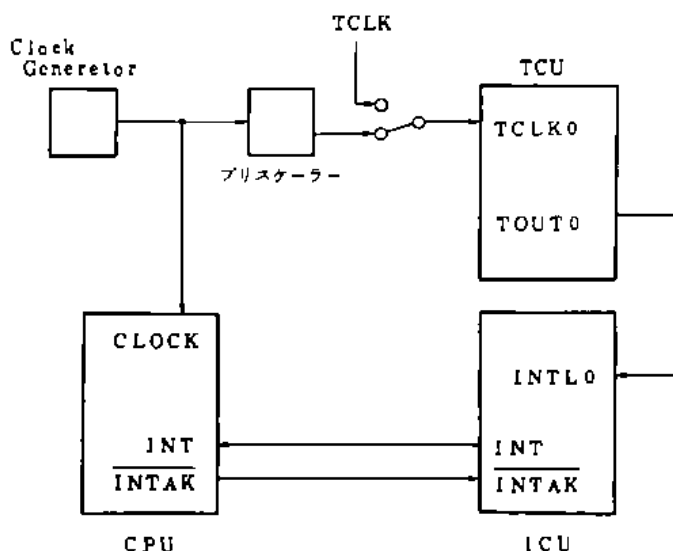
この信号をICUのレベル0割込み要求入力に接続し、1msごとに割込みを発生させます。この割込み処理ルーチンでは、呼び出されるごとにカウントを行ない時間をカウントします。

3.1.2 ハードウェアの説明

μPD70208, μPD70216にはCPU Clockを分周するプリスケアラ、TCU(μPD71054相当)およびICU(μPD71059相当)が内蔵されています。このプリスケアラの出力はシステムI/O領域のTCKSレジスタの設定により分周数を変えたり、TCUのCLK入力に接続することができます。またTCUのカウンタ#0出力がICUのレベル0割込み要求入力に内部接続されており、このTCUからの入力トリガはエッジ・トリガに固定されています。

ICUのINT要求信号はCPUに内部接続されています。以上の接続関係を示すと図3-1のようになります。

図3-1 CPUとTCU, ICUの関係



各ペリフェラルおよび相互接続はすべてμPD70208, μPD70216 内部で行なわれるので、

この応用例での外付けの回路は不用です。

3.1.3 ソフトウェアの説明

プログラムは次の三つの部分からなっています。

- (1) メインルーチン
- (2) 内蔵ペリフェラルの初期設定
- (3) 割込み処理ルーチン

以下、各部分について概要を説明します。

(1) メイン・ルーチン

最初に内蔵ペリフェラルの初期設定ルーチンをコールし、ウェイト・サイクルの設定、TCU、ICUの初期化および動作モードの設定、割込みベクタ・テーブルの設定を行ないます。

初期化後、MSEC(1/1000秒単位のカウンタ)およびSEC(1秒単位のカウンタ)の各カウンタを0にします。次にタイマ割込みを発生させるために、IMKWレジスタのM0ビットを0にして、INTL0からの割込み要求のマスクを解除します。IMKWのマスクビットの設定は、他の処理で使用されているマスクビットを変化させないようにするために現在のIMKWを読み出し、AND/OR命令あるいはCLR1/SET1命令により該当ビットのみをリセット/セットし、これをIMKWレジスタに書き込むことにより行なっています。またこの設定中に割込みが発生し、その割込み処理ルーチンの中でIMKWが書き換えられることを防ぐためにIMKWの設定は、割込み禁止状態で行ないます。

以上の設定により、1/1000秒ごとに割込みが発生します。この割込み処理ルーチンではコールされるとごとにカウンタをカウント・アップします。したがってCWレジスタにウェイトさせたい秒数をセットし、これとSEC(秒カウンタ)が一致するまでループさせることにより、プログラム中で一定時間ウェイトさせることができます。

タイマ割込みを停止するには、割込み禁止状態にして現在のIMKWを読み出しM0ビットをセットしてIMKWレジスタに書き込みます。

(2) 内蔵ペリフェラルの初期設定ルーチン

TCUおよびICUのI/OアドレスをOPHA、TULA、IULAレジスタに設定し、OPSELに各ユニットの使用、不使用を設定します。

I/Oおよびメモリ・サイクルでのウェイト数を設定します。メモリ・サイクルのウェイト数は、メモリ空間の三つのブロックに分けて設定します。ここではRAMおよびROMの実装されているエリア、各32KBをLMB、UMBとして設定し、これらへのメモリ・アクセスとI/Oアクセスのサイクルでは1ウェイト・サイクルを挿入するようにWCY1を設定しています。ウェイト・サイクルの設定には、この他にWCY2がありますが、この応用例では、DMAサイクルおよびリフレッシュ・サイクルは使用しないので省略しています。

リフレッシュ・ユニットの動作モードを設定します。この応用例ではリフレッシュ機能は使用しないので特に設定しなくとも動作しますが、リフレッシュ・サイクルが実行されることによりバス使用効率が下がるためにリフレッシュ動作を禁止しています。

システム I/Oレジスタにはこの他に OPCN, SULA, DULA がありますが、この応用例には関係しないので設定を省略しています。

内蔵プリスケアラおよび TCU の動作モードを設定します。TCKS レジスタによりプリスケアラの分周数を 8, TCU のクロック入力をプリスケアラ出力に接続するよう設定しています。TCU のカウンタ 0 はモード 2 でカウント数 625 で動作するようモード・ワードおよびカウント数を設定します。これにより、CPU クロックが 5MHz の場合 TOUT0 出力からは 1kHz のパルス出力が得られます。

ICU に対しては初期化と動作モードの設定を行いません。エッジ・トリガ、拡張なし、通常ネスト・モードに設定し、全要求をマスクします。I I W 1 の番込みによって行なわれるデフォルト・イニシャライズの設定を変更する必要がないので I I W 4 の番込みを省略しています。

ICU の設定後、使用される割込みベクタを設定します。

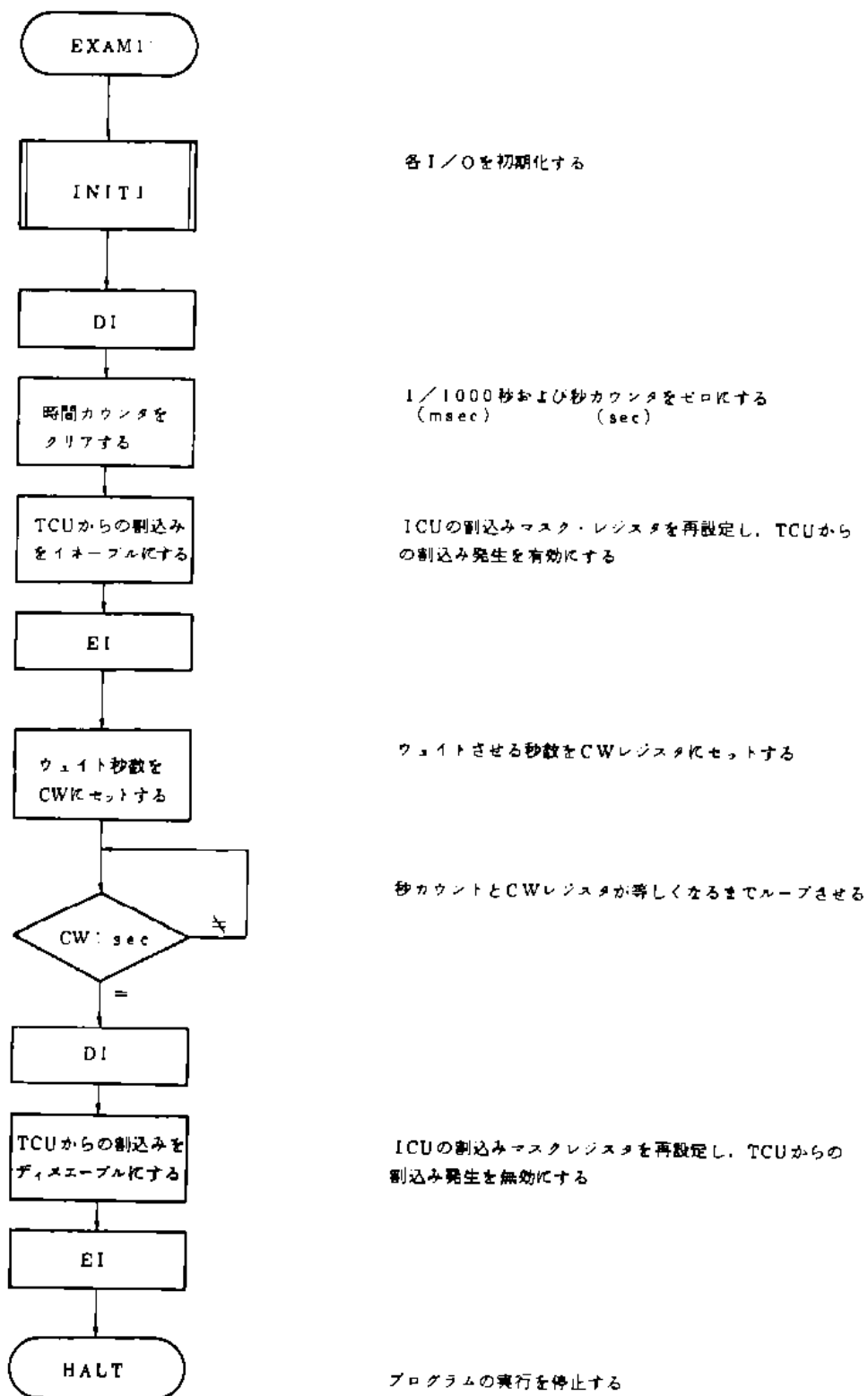
③ 割込み処理ルーチン

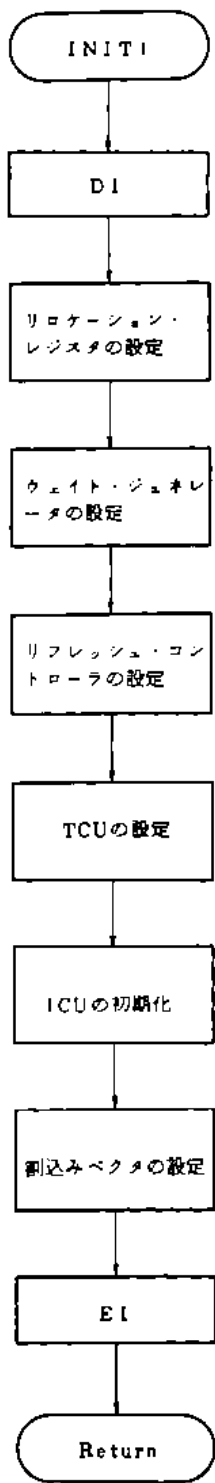
割込み発生によりこのルーチンがコールされると MSEC および SEC を一つカウントします。

処理は以下の手順で行なっています。

- ① このルーチンで使用されるレジスタ AW および DS0 をスタックに退避する。
- ② MSEC を一つカウントする。
- ③ MSEC が 1000 になったら MSEC を '0' にして SEC を一つカウントする。
- ④ ICU に FI コマンドを出す。
- ⑤ レジスタ AW, DS0 を元に戻す。
- ⑥ 割込み前のプログラムへ戻る。

3.1.4 フローチャート





イニシャライズ中は、割り込みを禁止する。

TCU、ICUのアドレスを設定し、次に、両ユニットを使用可能にする。

メモリ、I/Oのウェイト数を設定する。

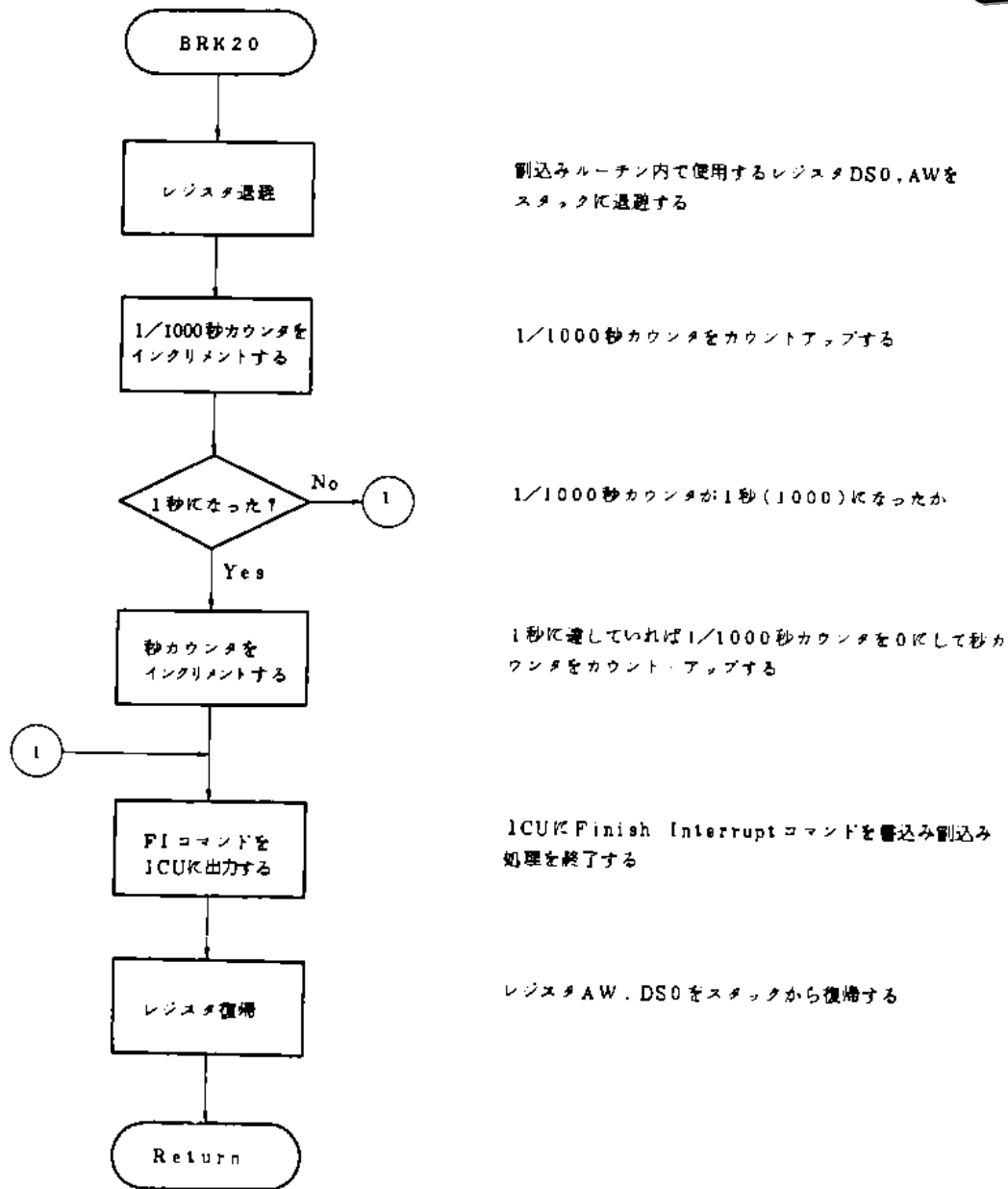
リフレッシュ機能を禁止する。

TCUとTCUのカウント#0を設定する。
プリスケール8分周、
カウンタ#0 モード2、カウント数=625

ICUのイニシャライズ
シングル・モード、エッジ・トリガ、
ベクタ番号20H、全入ガマスク。

ベクタ番号20Hに対応するベクタ・テーブルを設定する。

割り込みを許可する。



3.1.5 プログラム・リスト

```

LOC      OBJECT      LL#  N  SL#  SOURCE
1        1          ;*****
2        2          ;
3        3          ; PROGRAM EXAMPLE 1
4        4          ;
5        5          ; TIMER INTERRUPT
6        6          ;
7        7          ;*****
8        8          ;
9        9          NAME    EXAM1
10       10         ;
11       11         ; DEFINITION INPUT/OUTPUT PORT ADDRESS
12       12         ;
13       13         ; SYSTEM I/O ADDRESS
14       14         ;
15       15 OPON    EQU    OFFFEH ; Peripheral Connection
16       16 OPSEL   EQU    OFFFDH ; Peripheral Selection
17       17 OPHA    EQU    OFFFDH ; Peripheral High Address
18       18 DULA    EQU    OFFFBH ; DMAU Low Address
19       19 IULA    EQU    OFFFAH ; ICU Low Address
20       20 TULA    EQU    OFFF9H ; TCU Low Address
21       21 SULA    EQU    OFFF8H ; SCU Low Address
22       22 WCY2    EQU    OFFF6H ; Wait control 2 (RPM, DMAM)
23       23 WCY1    EQU    OFFF5H ; Wait control 1 (LPM, MPM, UPM, TPM)
24       24 YMB     EQU    OFFF4H ; Memory Boundary
25       25 RFC     EQU    OFFF2H ; Refresh Control
26       26 TCKS    EQU    OFFFDH ; Timer Clock Selection
27       27         ;
28       28         ; TCU I/O Address
29       29         ;
30       30 ICU     EQU    00C0H ; ICU Base Address
31       31 IIM1    EQU    TCU ; Interrupt Initialize 1
32       32 IIM2    EQU    TCU+2 ; Interrupt Initialize 2
33       33 IIM3    EQU    TCU+2 ; Interrupt Initialize 3
34       34 IIM4    EQU    TCU+2 ; Interrupt Initialize 4
35       35 IMKM    EQU    TCU+2 ; Interrupt Mask Register
36       36 IPFM    EQU    TCU ; Priority & FI
37       37 IMDM    EQU    TCU ; Interrupt Mode Register
38       38         ;
39       39         ; TCU I/O Address
40       40         ;
41       41 TCU     EQU    00D0H ; TCU Base Address
42       42 TCT0    EQU    TCU ; TCU#0 Counter
43       43 TST0    EQU    TCU ; TCU#0 Status
44       44 TCT1    EQU    TCU+2 ; TCU#1 Counter
45       45 TST1    EQU    TCU+2 ; TCU#1 Status
46       46 TCT2    EQU    TCU+4 ; TCU#2 Counter
47       47 TST2    EQU    TCU+4 ; TCU#2 Status
48       48 TMD     EQU    TCU+6 ; TCU Mode Register
49       49         ;
50       50 $EJECT

```

LOC	OBJECT	LL#	N	SL#	SOURCE
		51	51	PROG1	SEGMENT
		52	52		ASSUME PS:PROG1, DSO:DATA1
		53	53		;*****
		54	54		;*
		55	55		MAIN ROUTINE 1
		56	56		;
		57	57		;*****
		58	58		
0000		59	59	EXAM1	PROC
		60	60		
0000	E827D0	61	61		CALL INIT1 ; I/O Initialize
		62	62		
0003	C706{0200}0000	63	63		MOV WORD PTR MSEC.0 : Timer Counter Clear
0009	C706{0400}0000	64	64		MOV WORD PTR SEC.0
		65	65		
000F	FA	66	66		DI
0010	E4C2	67	67		IN AL,IMKN ; Timer Interrupt Enable
0012	24FE	68	68		AND AL,DFEH
0014	E6C2	69	69		OUT IMKN,AL
0016	FB	70	70		EI
		71	71		
0017	880E{0000}	72	72		MOV CW,WORD PTR COUNT ; Wait Time(second) Set
		73	73		
		74	74		WAIT:
001B	380E{0400}	75	75		CMP CW,WORD PTR SEC ; Check Wait Time
		76	76		
001F	75FA	77	77		BNE WAIT
		78	78		
0021	FA	79	79		DI
0022	E4C2	80	80		IN AL,IMKN ; Timer Interrupt Disable
0024	00D1	81	81		OR AL,D1H
0026	E6C2	82	82		OUT IMKN,AL
0028	FB	83	83		EI
		84	84		
0029	F4	85	85		HALT ; Program Stop
		86	86	EXAM1	ENDP
		87	87		REJECT

LOC	OBJECT	LL#	N	SL#	SOURCE
		88	88		;*****
		89	89		;*
		90	90		;* INITIALIZE ROUTINE 1
		91	91		;*
		92	92		;*****
		93	93		
002A		94	94		INIT1 PROC
002A	FA	95	95		01
		96	96		
0028	BAFCFF	97	97		MOV DW,OPHA ; OPHA: 00H
002E	B000	98	98		MOV AL,00H
0030	EE	99	99		OUT DW,AL
		100	100		
0031	BAF9FF	101	101		MOV DW,TULA ; TULA: TCU=0000H
0034	B000	102	102		MOV AL,TCU
0036	EE	103	103		OUT DW,AL
		104	104		
0037	BAFAFF	105	105		MOV DW,IULA ; IULA: ICU=0000H
003A	B0C0	106	106		MOV AL,ICU
003C	EE	107	107		OUT DW,AL
		108	108		
003D	BAF0FF	109	109		MOV DW,OPSEL ; OPSEL: TCU,ICU Select Active
0040	B006	110	110		MOV AL,06H
0042	EE	111	111		OUT DW,AL
		112	112		
0043	BAF4FF	113	113		MOV DW,WMB ; WMB: LWB=32KB,LWB=32KB
0046	B000	114	114		MOV AL,00H
0048	EE	115	115		OUT DW,AL
		116	116		
0049	BAF5FF	117	117		MOV DW,WCY1 ; WCY1: LWN,UMN,ICM=1,MMN=0
004C	B051	118	118		MOV AL,51H
004E	EE	119	119		OUT DW,AL
		120	120		
004F	BAF2FF	121	121		MOV DW,RFC ; RFC: Refresh Disable
0052	B000	122	122		MOV AL,00H
0054	EE	123	123		OUT DW,AL
		124	124		;
		125	125		; TCU Initialize
		126	126		;
0055	BAF0FF	127	127		MOV DW,TCKS ; TCKS: Internal,8 Divide
0058	B01A	128	128		MOV AL,1AH
005A	EE	129	129		OUT DW,AL
		130	130		
0058	B034	131	131		MOV AL,34H ; Counter #0 LSB/MSB, Mode 2
005D	E606	132	132		OUT TMD,AL
		133	133		
005F	B87102	134	134		MOV AN,625 ; Count=625
0062	E600	135	135		OUT TCT0,AL
0064	BAC4	136	136		MOV AL,AH
0066	E600	137	137		OUT TCT0,AL

LOC	OBJECT	LL#	N	SL#	SOURCE
		138	138	:	
		139	139	:	ICU initialize
		140	140	:	
0068	B012	141	141	MOV	AL,12H ; Edge Trigger, Single
006A	E6C0	142	142	OUT	IIM1,AL
		143	143		
006C	B020	144	144	MOV	AL,20H ; Base Vector=20H
006E	E6C2	145	145	OUT	IIM2,AL
		146	146		
0070	B0FF	147	147	MOV	AL,0FFH ; All Interrupt Mask
0072	E6C2	148	148	OUT	IMKN,AL
		149	149	:	
		150	150	:	Interrupt Vector Initialize
		151	151	:	
0074	B8{----}	152	152	MOV	AM,VECTOR
0077	8E00	153	153	MOV	DS1,AM
0079	26C706{8000}8900	154	154	MOV	DS1:WORD PTR VECTOR_AREA[4*20H],OFFSET BRK20
0080	26C706{8200}{----}	155	155	MOV	DS1:WORD PTR VECTOR_AREA[4*20H+2],SEG BRK20
		156	156		
0087	FB	157	157	EJ	
		158	158		
0088	C3	159	159	RET	
		160	160	INIT1	ENDP
		161	161	\$EJECT	

LOC	OBJECT	LL#	M	SL#	SOURCE
		162	162	:	*****
		163	163	:	* *
		164	164	:	* INTERRUPT ROUTINE 1 *
		165	165	:	* *
		166	166	:	*****
		167	167	:	
0089		168	168	BRK20	PROC
		169	169	:	
0089	1E	170	170	PUSH	DS0 ; Save DS0,AW Register
008A	50	171	171	PUSH	AW
008B	2E8E1E{A000}	172	172	MOV	DS0,PS:DS0SEG ; Set DS0
		173	173	:	
0090	A1{0200}	174	174	MOV	AW,MSEC ; MSEC Count Increment
0093	40	175	175	INC	AW
0094	A3{0200}	176	176	MOV	MSEC,AW
		177	177	:	
0097	3DE803	178	178	COMP	AW,1000
009A	750A	179	179	BNE	SKIP1
		180	180	:	
009C	C706{0200}0000	181	181	MOV	WORD PTR MSEC,0
		182	182	:	
00A2	FF06{0400}	183	183	INC	WORD PTR SEC
		184	184	:	
		185	185	SKIP1:	
00A5	8020	186	186	MOV	AL,20H ; FI Command
00A8	E6C0	187	187	OUT	IPFW,AL
		188	188	:	
00AA	58	189	189	POP	AW ; Restore DS0,AW Register
00AB	1F	190	190	POP	DS0
		191	191	:	
00AC	CF	192	192	RETI	
		193	193	:	
00AD	[----]	194	194	DS0SEG	DATA1
		195	195	:	
		196	196	BRK20	ENDP
		197	197	:	
		198	198	PROC1	ENDS
		199	199	\$EJECT	

LOC	OBJECT	LL#	N	SL#	SOURCE
		200	200	DATA1	SEGMENT
		201	201		
0000	1400	202	202	COUNT	DW 20
		203	203		
0002	????	204	204	MSEC	DW ?
0004	????	205	205	SEC	DW ?
		206	206		
		207	207	DATA1	ENDS
		208	208		
		209	209	VECTOR	SEGMENT AT 0000H
		210	210		
0000	(256 ????????)	211	211	VECTOR_AREA	DDS 256
		212	212		
		213	213	VECTOR	ENDS
		214	214		
		215	215	END	EXAM1

3.2 RS-232-C インタフェース

3.2.1 概要

μPD70208, μPD70216 に内蔵される SCU は、若干の周辺デバイスを付加することにより RS-232-C 対応のシリアル・インタフェースとして使用することができます。また ICU を組み合わせ、データ転送の終了を割込みにより CPU に知らせることができます。

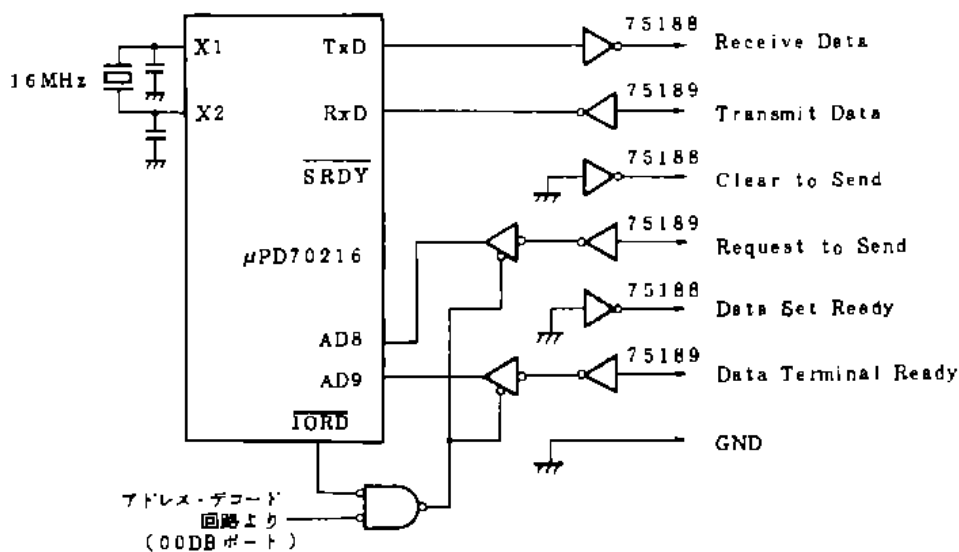
この応用例では μPD70216 外部に入力ポートを付加し、RS-232-C に対応させたシリアル・インタフェースを使用して、割込み処理によりブロック単位でのデータ転送を行なう例について示します。

3.2.2 ハードウェアの説明

この応用例では μPD70216 の他に、RS-232-C のステータス入力用ポートおよびレベル変換回路を必要とします。

RS-232-C インタフェースの回路例を図 3-2 に示します。

図 3-2 応用例 2 RS-232-C インタフェース



μPD70208, μPD70216 に内蔵される SCU は調歩同期モードによるシリアル転送ができます。SCU の端子は TxD 出力、RxD 入力、 $\overline{\text{SRDY}}$ 出力の 3 端子しかないので RS-232-C に対応させるためには、Data Terminal Ready, Request to Send などの信号を入力するポート (μPD71051 における $\overline{\text{DSR}}$, $\overline{\text{CTS}}$ 端子に相当) が必要となります。

この回路では RS-232-C の各信号に対して次のように対処しています。

1) Transmit Data

RxD 端子から入力します。

2) Receive Data

TxD端子から出力します。

3) Clear to Send

相手側に対して送信を要求する信号です。SRDY端子を接続し、受信バッファ・レディ時(RBRDY=1)の新たなデータ送信を禁止させることもできますが、この例ではつねにアクティブな信号が出力されるようにしています。

4) Request to Send

相手側ターミナルの送信要求信号で、受信動作が可能であることを示す信号として使われるため、外付けの入力ポートを設け、この信号を入力します。

データを送信する場合にはこの信号がアクティブであることを調べてからデータを出力します。

5) Data Terminal Ready

相手側に対してケーブルが接続されていることおよびこちら側の電源が入っていることを示す信号です。レベル変換回路の入力をロウ・レベルに固定し、電源が入ればアクティブな信号が出力されるようにします。

6) Data Set Ready

相手側のターミナルの電源が入っていることおよびケーブルが接続されていることを検知する信号です。外付けの入力ポートを設け、この信号を入力します。

7) GND

信号源のグラウンドを接続します。

シリアル転送のボー・レートはTCUのカウンタ#1出力により決定されます。この例では、CPU Clock (8MHz) をプリスケラおよびTCUで52分周し、この出力をSCU内部で16分周し、ボー・レートを約9600ボーにしています。

またデータ転送の終了をTxRDY信号またはRxRDY信号による割込みにより処理するにはこれらの信号をICUへ入力する必要があります。

μPD70208、μPD70216では、TxRDY信号およびRxRDY信号のOR出力が切換えSWを介してICUのINTL1へ接続されています。このSWはOPCNのIRSWビットで設定します。

3.2.3 ソフトウェアの説明

(1) メイン・ルーチン

イニシャライズ・ルーチンでI/Oの初期化を行なった後、送信データの先頭オフセットとデータ数、受信データ用バッファの先頭アドレスと大きさを設定します。

ウェイト・カウンタをクリアし、SCUからの割込みを許可します。

送信、受信の準備ができると割込みが発生し、割込み処理ルーチンがコールされます。

割込み処理ルーチンの中ではSCUのステータスを調べ、送信データのセットまたは受信データの読込みを行ない更にウェイト・カウンタをクリアします。

ウェイト・カウンタをインクリメントし465になるまでループします(CPUクロックが8MHzの場合、2.5msec間ループします)。

割込み処理ルーチンの中でウェイト・カウンタには0がセットされますから、2.5 msec以上データの転送が行なわれないと、このループを抜け出します。

SIMKレジスタに03HをセットしてSCUからの割込みを禁止します。

(2) イニシャライズ・ルーチン

メモリ・ブロックの分割サイズとメモリ、I/Oのウェイト・サイクル数を設定します。

この応用例ではリフレッシュ動作を必要としないので、バスの使用効率を上げるためリフレッシュ動作を禁止します。

外部兼用端子にはSCUの端子を接続します。

内蔵ペリフェラルのうちTCU(00D0H)、ICU(00C0H)、SCU(00D8H)のI/Oアドレスを設定します。

各I/Oのベース・アドレスは、TCUがD0H、SCUがD8H、ICUがC0Hです。

TCUはカウンタ#1をモード3、カウント数26に設定し、CLK入力には、CPUクロックをプリスケアラで2分周した信号を入力します。

ICUは、レベル・トリガ、シングル・モード、ベクタ番号(20H-27H)、SCUからのベクタ番号に対する割込みベクタの設定とマスク・レジスタ(INTL1のマスク解除、INTL0、2-7のマスクセット)を設定し、SCUからの割込み要求に対応する割込みベクタを書込みます。

SCUのイニシャライズはモード・ワードにより、データ長8ビット、パリティ無し、ストップ・ビット2ビット、ポー・レート1/16を設定した後、コマンド・ワードにより送受信可、 $\overline{\text{SRDY}}$ 出力通常動作を設定します。SIMKのTM、RMを1にして割込み発生をマスクします。

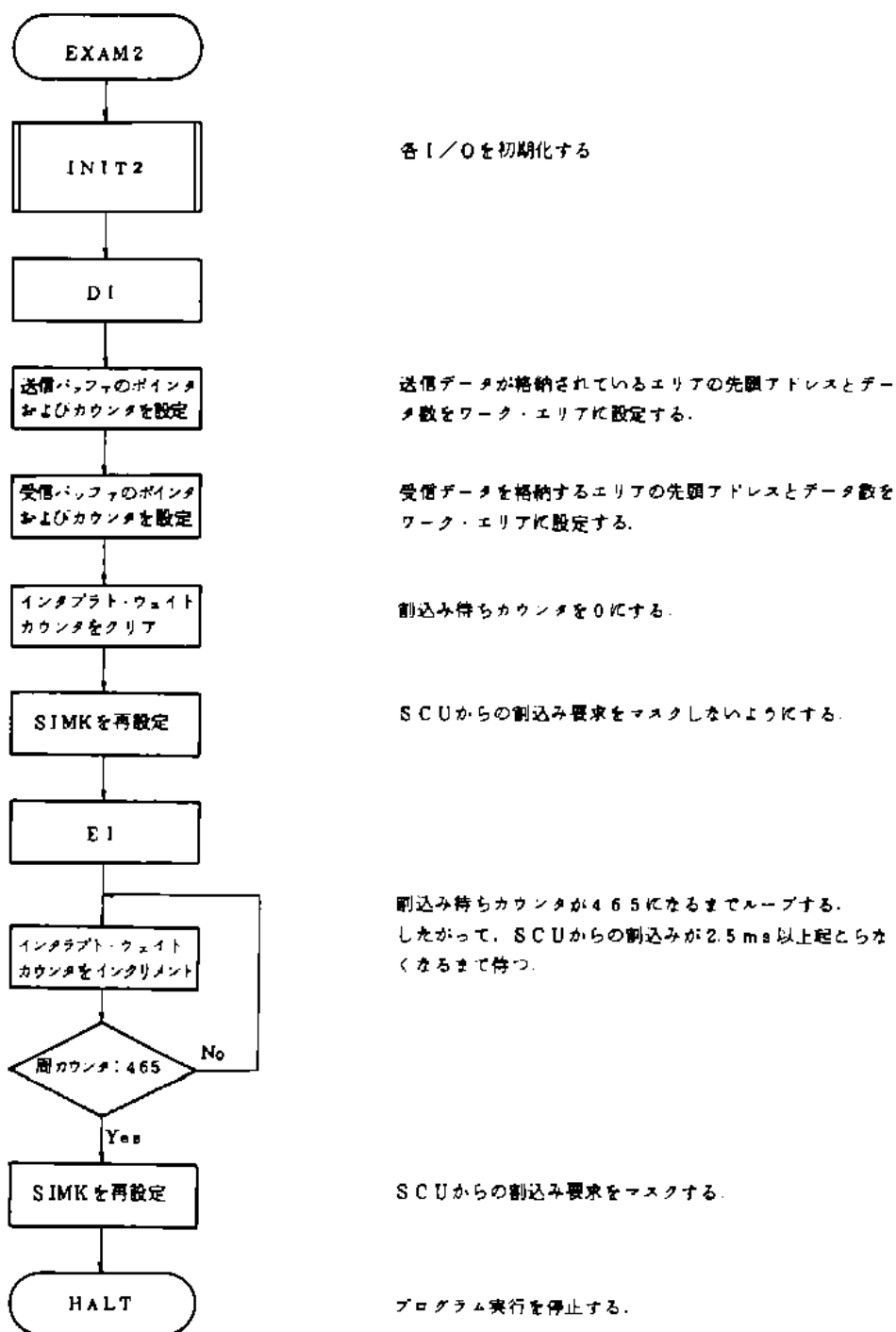
(3) 割込み処理ルーチン

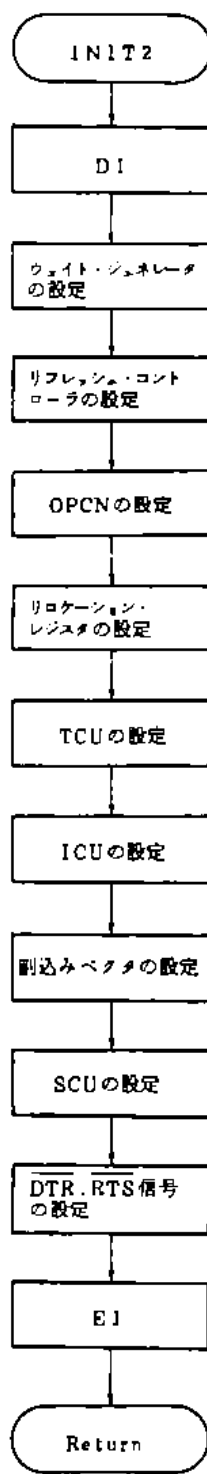
SCUのステータスを調べて、受信データがあれば、これを読み取り、ポインタの示すメモリ中のバッファへ格納します。バッファ領域が一杯になった時には、受信データの読込みは行なわず、SIMKのRMを1にしてRBRDYによる割込み要求をマスクします。

再度SCUのステータスを調べて送信データ待ちであれば送信データをバッファから取出し、STBに書込みます。送信すべきデータがない場合にはSIMKのTMビットを1にして以後のTBRDYによる割込み要求を禁止します。

また、この割込み処理ルーチンでは、呼び出されるごとにウェイト・カウンタを0にしています。

3.2.4 フローチャート





イニシャライズ中は割込みを禁止する。

メモリ、I/Oサイクルでのウェイト数を設定する。

リフレッシュ動作を禁止する。

兼用端子は、SCUの入出力端子として使用し、INTL1には、SCUからの割込み要求信号を接続する。

TCU、ICU、SCUのI/Oアドレスを設定し、次に各ユニットを使用可能にする。

TCCKSとTCUのカウント#1を設定する。
 内蔵プリスケアラ……2分周
 カウント#1……モード3，カウント数=26

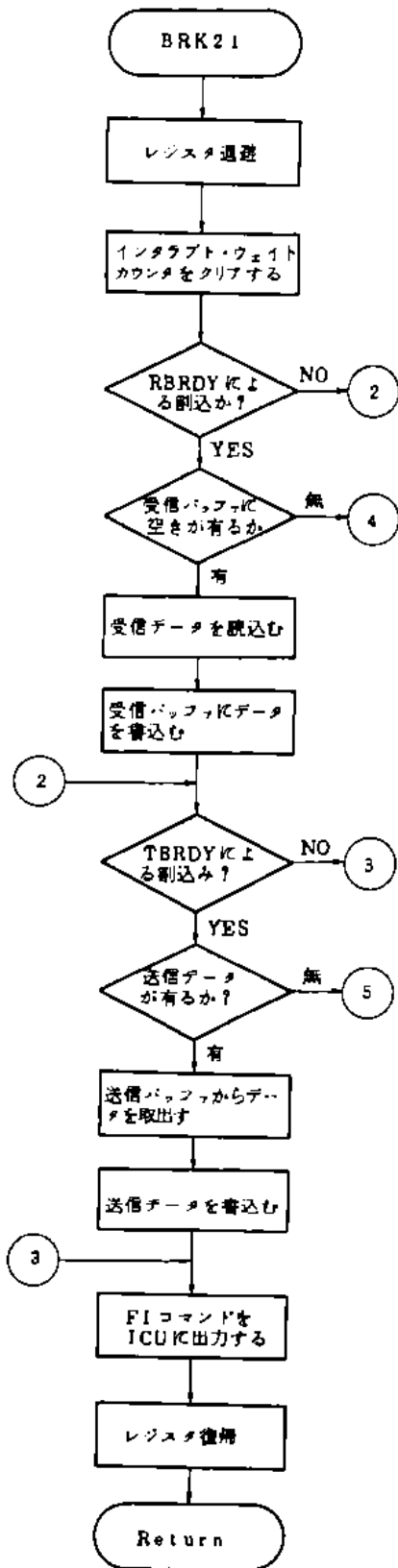
ICUのイニシャライズ
 シングルモード，エッジ・トリガベクタ番号20H，INTL1以外マスク

ベクタ番号21H(SCUからの割込み)に対応するベクタ・テーブルを設定する。

SCUのイニシャライズ8bit，パリティ無し，16分周，ストップ・ビット2ビット，送受信可

$\overline{\text{DTR}}$ 、 $\overline{\text{RTS}}$ 信号をアクティブにする

割込みを許可する



割込みルーチン内で使用するレジスタ DS0, AW, BW, CWをスタックに退避する

割込み待ちカウンタをゼロ化する

(RBRDY=1)かつ(RM=0)であるか調べ、RBRDYによる割込みでなければ②へ分岐する

受信バッファに空きが有るか調べ空きが無ければ④へ分岐し、RBRDYによる割込み要求をマスクする

SRBから受信データを読み込む

データを受信バッファに格納し、ポインタおよびカウンタを更新する

(TBRDY=1)かつ(TM=0)であるか調べTBRDYによる割込みでなければ③へ分岐する

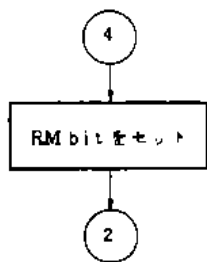
送信すべきデータが有るか調べデータが無ければ⑤へ分岐し、TBRDYによる割込み要求をマスクする

送信バッファから送信データを取り出し、ポインタおよびカウンタを更新する

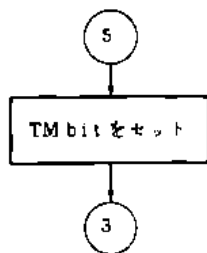
STBに送信データを書き込む

ICUにFinish Interruptコマンドを書込み、割込み処理を終了させる

レジスタDS0, AW, BW, CWをスタックから復帰する



SIMK を読出し、再設定する
RBRDY による割込み要求をマスクする



SIMK を読出し、再設定する
TBRDY による割込み要求をマスクする

★ 3.2.5 プログラム・リスト

LOC	OBJECT	LL#	N	SL#	SOURCE
1		1			*****
2		2			;
3		3			PROGRAM EXAMPLE 2
4		4			;
5		5			SERIAL INTERFACE (RS-232C & INTERRUPT)
6		6			;
7		7			*****
8		8			;
9		9			NAME EXAM2
10		10			;
11		11			DEFINITION INPUT/OUTPUT PORT ADDRESS
12		12			;
13		13			SYSTEM I/O ADDRESS
14		14			;
15		15			OPCN EQU OFFFEH ; Peripheral Connection
16		16			OPSEL EQU OFFFDH ; Peripheral Selection
17		17			OPHA EQU OFFFDH ; Peripheral High Address
18		18			DULA EQU OFFFBH ; DMAU Low Address
19		19			IULA EQU OFFFAH ; ICU Low Address
20		20			TULA EQU OFFF9H ; TCU Low Address
21		21			SULA EQU OFFFBH ; SCU Low Address
22		22			WCY2 EQU OFFF6H ; Wait control 2 (RFW, DMAW)
23		23			WCY1 EQU OFFF5H ; Wait control 1 (LWV, MMW, UMW, TOW)
24		24			MWB EQU OFFFAH ; Memory Boundary
25		25			RFC EQU OFFF2H ; Refresh Control
26		26			TCKS EQU OFFFDH ; Timer Clock Selection
27		27			;
28		28			TCU I/O Address
29		29			;
30		30			ICU EQU 0000H ; ICU Base Address
31		31			IIM1 EQU ICU ; Interrupt Initialize 1
32		32			IIM2 EQU ICU+2 ; Interrupt Initialize 2
33		33			IIM3 EQU ICU+2 ; Interrupt Initialize 3
34		34			IIM4 EQU ICU+2 ; Interrupt Initialize 4
35		35			IIMW EQU ICU+2 ; Interrupt Mask Register
36		36			IPPW EQU ICU ; Priority & FI
37		37			IMDW EQU ICU ; Interrupt Mode Register
38		38			;
39		39			TCU I/O Address
40		40			;
41		41			TCU EQU 0000H ; TCU Base Address
42		42			TCT0 EQU TCU ; TCU#0 Counter
43		43			TST0 EQU TCU ; TCU#0 Status
44		44			TCT1 EQU TCU+2 ; TCU#1 Counter
45		45			TST1 EQU TCU+2 ; TCU#1 Status
46		46			TCT2 EQU TCU+4 ; TCU#2 Counter
47		47			TST2 EQU TCU+4 ; TCU#2 Status
48		48			TMD EQU TCU+6 ; TCU Mode Register

LOC	OBJECT	LL#	N	SL#	SOURCE
49		49			;
50		50			SCU I/O Address
51		51			;
52	SCU	52			EQU 000BH : SCU Base Address
53	SRB	53			EQU SCU : RxData Buffer
54	STB	54			EQU SCU : TxData Buffer
55	SSR	55			EQU SCU+2 : Serial Status Register
56	SCR	56			EQU SCU+2 : Serial Command Register
57	SMD	57			EQU SCU+4 : Serial Mode Register
58	SIMR	58			EQU SCU+6 : Serial Interrupt Mask Register
59	SAUX	59			EQU 000BH : Serial Interface Auxiliary Port
60		60			;
61		61			\$EJECT

LOC	OBJECT	LL#	N	SL#	SOURCE
		62	62	PROG2	SEGMENT
		63	63		ASSUME PS:PROG2, DSO:DATA2
		64	64		:*****
		65	65	:	* * * * *
		66	66	:	MAIN ROUTINE 2 * * * * *
		67	67	:	* * * * *
		68	68		:*****
		69	69		
0000		70	70	EXAM2	PROC
		71	71		
0000	E83500	72	72	CALL	INIT2 ; I/O Initialize
		73	73	:	
0003	FA	74	74		DI
		75	75		
0004	C706{0200}0A00	76	76	MOV	WORD PTR TXPNT,OFFSET TXDATA ; Tx-Buffer Pointer
000A	C706{0400}0900	77	77	MOV	WORD PTR TXCNT,9 ; Counter
		78	78		
0010	C706{0600}1300	79	79	MOV	WORD PTR RXPNT,OFFSET RXDATA ; Rx-Buffer Pointer
0016	C706{0800}8000	80	80	MOV	WORD PTR RXCNT,128 ; Counter
001C	C706{0000}0000	81	81	MOV	WORD PTR WAITCNT,0 ; Interrupt Wait Counter
		82	82		
0022	8000	83	83	MOV	AL,00 ; TM,PM=0 Interrupt Enable
0024	E6DE	84	84	OUT	SIMK,AL
		85	85		
0026	FB	86	86		EI
		87	87	:	
		88	88		LOOP2:
0027	FF06{0000}	89	89	INC	WORD PTR WAITCNT ; Increment Interrupt Wait Counter
		90	90		
0028	813E{0000}D101	91	91	CMP	WORD PTR WAITCNT,465 ; Check Greater Than 465(2.5ms)
		92	92		
0031	72F4	93	93	BL	LOOP2 ; Loop Until
		94	94	:	
0033	8003	95	95	MOV	AL,03H ; TM,PM=1 Interrupt Disable
0035	E6DE	96	96	OUT	SIMK,AL
		97	97	:	
0037	FA	98	98	HALT	; Program Stop
		99	99		
		100	100	EXAM2	ENDP
		101	101		\$EJECT

LOC	OBJECT	LU#	N	SL#	SOURCE
		102	102	:	*****
		103	103	:	0
		104	104	:	INITIALIZE ROUTINE 2
		105	105	:	0
		106	106	:	*****
		107	107	:	
0038		108	108		INIT2 PROC
		109	109	:	
0038	FA	110	110		DJ
		111	111	:	
0039	BAF4FF	112	112		MOV DW, WMB ; WMB: LWB=32KB, UWB=32KB
003C	B000	113	113		MOV AL, 00H
003E	EE	114	114		OUT DW, AL
		115	115	:	
003F	BAF5FF	116	116		MOV DW, WCY1 ; WCY1: LWB, UWB, IOW=1, MWY=0
0042	B051	117	117		MOV AL, 51H
0044	EE	118	118		OUT DW, AL
		119	119	:	
0045	BAF2FF	120	120		MOV DW, RFC ; RFC: Refresh Disable
0048	B000	121	121		MOV AL, 00H
004A	EE	122	122		OUT DW, AL
		123	123	:	
004B	BAFEFF	124	124		MOV DW, OPHN ; INTL1: SCU=INT
004E	B006	125	125		MOV AL, 06H ; Select Tx0, Rx0, SROY/
0050	EE	126	126		OUT DW, AL
		127	127	:	
		128	128	:	I/O Unit Address & Selection
		129	129	:	
0051	BAFCFF	130	130		MOV DW, OPHA ; OPHA: 00H
0054	B000	131	131		MOV AL, 00H
0056	EE	132	132		OUT DW, AL
		133	133	:	
0057	BAF9FF	134	134		MOV DW, TULA ; TCU=0000H
005A	B000	135	135		MOV AL, TCU
005C	EE	136	136		OUT DW, AL
		137	137	:	
005D	BAFAFF	138	138		MOV DW, IULA ; TCU=0000H
0060	B000	139	139		MOV AL, TCU
0062	EE	140	140		OUT DW, AL
		141	141	:	
0063	BAF0FF	142	142		MOV DW, SULA ; SCU=0008H
0066	B008	143	143		MOV AL, SCU
0068	EE	144	144		OUT DW, AL
		145	145	:	
0069	BAF0FF	146	146		MOV DW, OPSEL ; TCU, TCU, SCU Select Active
006C	B00E	147	147		MOV AL, 0EH
006E	EE	148	148		OUT DW, AL

LOC	OBJECT	LL#	N	SL#	SOURCE
		149		149	;
		150		150	; TCU Initialize
		151		151	;
006F	BAF0FF	152		152	MOV DM, TCKS ; Counter #1: Internal Clock Divide by 2
0072	B014	153		153	MOV AL, 14H
0074	EE	154		154	OUT DM, AL
		155		155	
0075	B057	156		156	MOV AL, 57H ; Counter #1 Mode 3, 8C0, LSB
0077	E6D6	157		157	OUT TMD, AL
		158		158	
0079	B01A	159		159	MOV AL, 26 ; Count=26
007B	E6D2	160		160	OUT TCT1, AL
		161		161	;
		162		162	; TCU Initialize
		163		163	;
007D	B01A	164		164	MOV AL, 1AH ; Level Trigger, Single Mode
007F	E6D0	165		165	OUT I1#1, AL
		166		166	
0081	B020	167		167	MOV AL, 20H ; Vector=20H-27H
0083	E6C2	168		168	OUT I1#2, AL
		169		169	
0085	B0FD	170		170	MOV AL, 0FDH ; INTL1 Enable, INTL0 2-7 Disable
0087	E6C2	171		171	OUT I1#N, AL
		172		172	;
		173		173	; Interrupt Vector Initialize
		174		174	;
0089	B8{----}	175		175	MOV AM, VECTOR
008C	BEC0	176		176	MOV DS1, AM
		177		177	
008E	26C706{B400}AA00	178		178	MOV DS1:WORD PTR VECTOR_AREA[4*21H], OFFSET BRK21
0095	26C706{0600}{----}	179		179	MOV DS1:WORD PTR VECTOR_AREA[4*21H+2], SEG BRK21
		180		180	;
		181		181	; SCU Initialize
		182		182	;
009C	B0CE	183		183	MOV AL, 0CEH ; Mode 8 Bit, No Parity,
009E	E6DC	184		184	OUT SMD, AL ; 2 Stop-Bit, Baud Rate 1/16
		185		185	
00A0	B035	186		186	MOV AL, 35H ; Command SPDY=RBRODY, TE=1, RE=1
00A2	E6DA	187		187	OUT SCA, AL
		188		188	
00A4	B003	189		189	MOV AL, 03H ; TM, RM Mask
00A6	E6DE	190		190	OUT SIMK, AL
		191		191	;
00AB	FB	192		192	EI
00A9	C3	193		193	RET
		194		194	
		195		195	INIT2 ENDP
		196		196	\$EJECT

LOC	OBJECT	LL#	N	SL#	SOURCE
		197	197	:	*****XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
		198	198	:	*
		199	199	:	INTERRUPT ROUTINE 2 *
		200	200	:	*
		201	201	:	*****XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
		202	202	:	
00AA		203	203	BRK21	PROC
		204	204	:	
00AA 1E		205	205	PUSH	DSO ; Save DSO, AN, BW, CW Register
00AB 50		206	206	PUSH	AN
00AC 53		207	207	PUSH	BW
00AD 51		208	208	PUSH	CW
		209	209	:	
00AE 88{----}		210	210	MOV	AN, DATA2
00B1 8EDB		211	211	MOV	DSO, AN
		212	212	:	
00E3 C7D6{0000}0000		213	213	MOV	WORD PTR WAITCNT, 0 ; Count 0
		214	214	:	
		215	215	:	Serial Data Receive
		216	216	:	
00B9 E4DE		217	217	IN	AL, SIMK ; Check RM=0 RBRDY Interrupt Enable
00BB A801		218	218	TEST	AL, 1
00BD 7522		219	219	BNZ	TXBRK
		220	220	:	
00BF E4DA		221	221	IN	AL, SST ; Check RBRDY=1 RxData Buffer Ready
00C1 A802		222	222	TEST	AL, 2
00C3 741C		223	223	BZ	TXBRK
		224	224	:	
00C5 880E{0800}		225	225	MOV	CW, WORD PTR RXCNT ; Rx Buffer Area Left ?
00C9 F7C10000		226	226	TEST	CW, 0 ; If None Area, Branch RX_MASK
00CD 7443		227	227	BZ	RX_MASK
		228	228	:	
00CF E4DB		229	229	IN	AL, SAB ; Read Receive Data
		230	230	:	
00D1 881E{0600}		231	231	MOV	BW, WORD PTR RXPNT ; Put to Rx Buffer
00D5 8807		232	232	MOV	BYTE PTR [BW], AL
00D7 43		233	233	INC	BW ; Pointer Increment
00D8 891E{0600}		234	234	MOV	WORD PTR RXPNT, BW
00DC 49		235	235	DEC	CW
00E0 890E{0800}		236	236	MOV	WORD PTR RXCNT, CW ; Counter Decrement
		237	237	:	
		238	238	:	Serial Data Transmit
		239	239	:	
		240	240	TXBRK	:
00E1 E4DB		241	241	IN	AL, SAUX ; Read CTS, DSR Signal
00E3 BAE0		242	242	MOV	AH, AL
00E5 E4DA		243	243	IN	AL, SST ; Read Status
00E7 250103		244	244	AND	AN, 0301H
00EA 300103		245	245	CMP	AN, 0301H ; Check TBRDY, CTS, DTR=1
00ED 751A		246	246	BNE	BRK21F

LOC	OBJECT	LL#	N	SL#	SOURCE	
		247		247	:	
00EF	8B0E{0400}	248		248	MOV	CW, WORD PTR TXCNT ; Tx Buffer Area Empty ?
00F3	85C9	249		249	TEST	CW, CW
00F5	7423	250		250	BZ	TX_MASK ; If Empty, Branch TX_MASK
		251		251	:	
00F7	8B1E{0200}	252		252	MOV	BW, WORD PTR TXPNT ; Get From Tx-Buffer
00FB	8A07	253		253	MOV	AL, BYTE PTR [BW]
		254		254		
00FD	43	255		255	INC	EW ; Tx Buffer Pointer Increment
00FE	891E{0200}	256		256	MOV	WORD PTR TXPNT, BW
		257		257		
0102	49	258		258	DEC	CW ; Tx Buffer Count Decrement
0103	89DE{0400}	259		259	MOV	WORD PTR TXCNT, CW
		260		260		
0107	E60B	261		261	OUT	STB, AL ; Write Transmit Data
		262		262	:	
		263		263	BRK21F :	
0109	8020	264		264	MOV	AL, 20H ; FI Command
010B	E6CD	265		265	OUT	IPFW, AL
		266		266	:	
010D	59	267		267	POP	CW ; Restore DS0, AW, BW, CW Register
010E	5B	268		268	POP	BW
010F	5B	269		269	POP	AW
0110	1F	270		270	POP	DS0
		271		271	:	
0111	CF	272		272	RET1	; Return From Interrupt
		273		273		
		274		274	RX_MASK :	
0112	E4DE	275		275	IN	AL, SIMK ; Set RM=1
0114	0CD1	276		276	OR	AL, 1 ; RRDY Interrupt Mask
0116	E6DE	277		277	OUT	SIMK, AL
011B	EBC7	278		278	BR	SHORT TXBRK
		279		279	:	
		280		280	Tx_MASK :	
011A	E4DE	281		281	IN	AL, SIMK ; Set TM=1
011C	0CD2	282		282	OR	AL, 2 ; TBRDY Interrupt Mask
011E	E6DE	283		283	OUT	SIMK, AL
0120	EBC7	284		284	BR	SHORT BRK21F
		285		285	:	
		286		286	BRK21 ENOP	
		287		287	:	
		288		288	PROG2 ENDS	
		289		289	\$\$EJECT	


```

LOC      OBJECT      LL#  H  SL#  SOURCE
          290  290  DATA2  SEGMENT
          291  291
0000 0000 0000 292  292  WAITCNT DW    ?      ; Interrupt Wait Counter
          293  293
0002 0002 0002 294  294  TXPNT  DW    ?      ; Tx Buffer Pointer(Offset)
0004 0004 0004 295  295  TXCNT  DW    ?      ; Counter
          296  296
0006 0006 0006 297  297  RXPNT  DW    ?      ; Rx Buffer Pointer(Offset)
0008 0008 0008 298  298  RXCNT  DW    ?      ; Counter
          299  299
000A 52532032333243 300  300  TXDATA DB    'RS-232C',00H,0AH ; Transmit Data Buffer Area
0011 0011 0011
0012 0012 0012
          301  301
0013 (128 302  302  RXDATA DBS    128      ; Receive Data Buffer Area
      ??
      )
          303  303
          304  304  DATA2  ENDS
          305  305
          306  306  VECTOR  SEGMENT AT 0000H
          307  307
0000 (256 308  308  VECTOR_AREA DBS    256      ; Interrupt Vector Area
      ???????
      )
          309  309
          310  310  VECTOR  ENDS
          311  311  :
          312  312      END    EXAM2

```

3.3 DMAUとリフレッシュ機能の応用例

3.3.1 概要

μPD70216に内蔵されているDMAUを使用して、外付けされたμPD71051のデータ入出力をDMA転送によって行なう応用例を示します。

また、リフレッシュ機能を用いてDRAMのリフレッシュを行なう例を合わせて示します。

3.3.2 ハードウェアの説明

μPD71051のデータ入出力は、チャンネル2により受信データの入力、チャンネル3により送信データの出力をDMA転送します。

このためμPD70216とμPD71051との間で①DMARQ ② $\overline{\text{DMAAK}}$ ③上位/下位データ・バスを接続しています(図3-3参照)。

DMARQの接続は、DMA転送を要求するためにDMARQ2にTxRDYを、DMARQ3にRxRDYを接続しています。DMARQ3および $\overline{\text{DMAAK3}}$ は兼用端子となっていますので、イニシャライズ・プログラムでOPCNのPFビットを設定しておきます。

$\overline{\text{DMAAK}}$ の接続は、DMAサイクル中のデータ入出力を可能としますが、μPD71051にはこのアクノリッジ信号を受け付ける制御機能はないので、ここでは $\overline{\text{DMAAK2}}$ または $\overline{\text{DMAAK3}}$ がアクティブになるとμPD71051の $\overline{\text{CS}}$ をアクティブに、A0を0にしてデータ入出力が可能となるようにします。

データ・バスの接続は、メモリの偶数番地がデータ・バスの下位に、奇数番地が上位に分かれて接続されており、DMA転送されるデータを連続的に格納するにはμPD71051のデータ・バスを上位/下位両データ・バスに接続する必要があります。ここではバス・トランシーバ(μPD71086)を2個用いて、アドレス・バスのA0によって切替えています。

μPD71051のクロック入力CLKOUTをCLKに、TOUT1をRxCLK, TxCLKに接続し、μPD70216から供給します。

DRAMのアクセスは $\overline{\text{MRD}}$ または $\overline{\text{MWR}}$ とアドレス・デコードの結果とがアクティブになるとアクセスを開始します。このタイミングで $\overline{\text{RAS}}$ をLOWにします。次に遅延回路1, 2によりアドレス切替え信号Sが25 ns, $\overline{\text{CAS}}$ が50 ns遅れてLOWになりDRAMのアクセスが行なわれます。

また $\overline{\text{MWR}}$ とA0, あるいは $\overline{\text{UBE}}$ とのORをとり、これを $\overline{\text{WE}}$ に投入し書き込みサイクルにおける下位バイト・上位バイトの書き込みを制御します(アーリー・ライト・サイクル)。

リフレッシュ・サイクルの場合には $\overline{\text{REFRQ}}$ によりSおよび $\overline{\text{CAS}}$ はハイ・レベルに固定され、 $\overline{\text{RAS}}$ のみがアクティブとなり、アドレス・バスのA8-A0に出力されているリフレッシュ・アドレスをラッチし $\overline{\text{RAS}}$ オンリ・リフレッシュを行ないます。

3.3.3 ソフトウェアの説明

前述したハードウェアを使用した例として、シリアル・インタフェースから20バイトのデータを入力し、次にそのデータをそのままシリアル・インタフェースを通して送り返すプログラム例を

示します。

(1) メイン・ルーチン

まず、イニシャライズ・ルーチンをコールし、各I/Oを初期化します。この初期化には、ウエイト・ジェネレータ、リフレッシュ・コントローラ、TCU、 μ PD71051の初期化および動作モードの設定が含まれます。

μ PD71051の受信データをDMA転送するためDMAUのチャンネル2を設定します。チャンネルの設定は、設定したいチャンネルを選択し、動作モード、転送開始アドレス、転送回数を設定し、マスク・レジスタの該当ビットをクリアします。ここではI/Oからメモリへ、シングル、バイト転送モードを設定し、転送開始アドレスにはBUFFの先頭アドレスを設定します。カウント数には19(20-1)を設定します。チャンネルの設定途中において、誤ってDMA転送が行なわれることを防ぐために設定中のチャンネルは、マスク・レジスタによってマスクするか、ディバイス・コントロール・レジスタによってDMA動作を禁止して設定を行いません。

チャンネル2の設定後、DMA動作を許可すると μ PD71051へデータが受信されるごとに、自動的に20バイトまでDMA転送が行なわれます。20バイト転送が終わるとTCビットがセットされます。プログラムではステータスを読取りTC2(チャンネル2のTCビット)が1になるまでループし、20バイト受信終了を待ちます。

次に μ PD71051へ送信データをDMA転送するためにDMAUのチャンネル3を設定します。チャンネル2の設定と同様にDMA動作を禁止し、チャンネル選択コマンドでチャンネル3を指定します。動作モードはメモリからI/Oへシングル、バイト転送モードを設定します。転送開始アドレスにはバッファ領域(BUFF)の先頭アドレスを設定し、転送回数には19(20-1)を設定します。マスク・レジスタを讀出しM3ビットをOFFし、再びマスク・レジスタへ書込み、チャンネル3のマスクを解除します。その後、ディバイス・コントロール・レジスタを書換え、DMA動作を許可します。

DMA動作が許可されると μ PD71051が送信可となるごとに次の送信データがDMA転送されます。20バイト転送を終えるとTCビット(TC3)がセットされるので、プログラムではステータスを読取りTC3が1となるまでループしデータの転送終了を待ちます。

以上で20バイトのシリアル・データの受信と、同データの送信(送返し)とがDMA転送を使用して行なわれます。

(2) イニシャライズ・ルーチン

ウエイト・ジェネレータを設定する前にメモリ・バンクの分割サイズを設定しています。ここではLMB(DRAM領域)を512KB、UMBを64KBとして設定しています。

ハードウェア・リセットを行なうとすべてのサイクルに3ウエイト・スタートが挿入されます。このままでも動作しますが、不要なウエイト・スタートが挿入されるため処理能力が低下します。ここでは適正なウエイト数とするため、I/OサイクルとMMBのメモリ・サイクルではウエイト無し、LMBおよびUMBのメモリ・サイクルとリフレッシュ・サイクルでは1ウエイト、DMAサイクルでは2ウエイトとなるよう設定しています。

DRAMのリフレッシュを行なうために、 $N=15$ (RTM=0EH)、リフレッシュ動作を許可します。CPUの動作周波数は8MHzですから15 μ s間隔でDRAMのリフレッシュ要求が

発生します。

内蔵ペリフェラルのうち使用するTCU、DMAUのアドレスをそれぞれ00D0H、00B0Hにして、両I/Oを使用可にします。

TCUのカウンタ1を使用するのでTCKSを設定します。ここではCPUクロックをプリスケータで2分周した内部クロックをカウンタ1のクロック入力にします。

μPD71051のポーレート・クロックをTOUT1から供給するためカウンタ1をモード3、カウント数26に設定します。CPUのクロック周波数が8MHzの場合TOUT1出力は約153.8kHzとなり、実際のポーレートはμPD71051内部で更に16分周され9615ポー（約9600ポー）となります。

DMAUはイニシャライズ・ワードを書込むとリセットされ、アドレスおよびカウント・レジスタを除く全レジスタが初期化されます。この初期化によりDMAUの動作モードは、バスリリース・モード、通常書込みタイミング、固定優先順位となり、転送モードはペリファイ転送、ダイヤモンド・モードで全チャンネルがマスクされます。各チャンネルはDMA転送を行なう直前に設定するのでここではDMA動作禁止のみ設定します。

μPD71051をリセットするために、00Hを3回書込み後、40H（ソフトウェア・リセット）を書込みます。ソフトウェア・リセット後、モード・ワード、コマンド・ワードを書込み、転送方式、動作モードを設定します。転送方式は送受信クロックを16分周してポーレートとし、データ長8ビット、パリティ無し、2ストップ・ビットで転送します。動作モードは送受信可能としています。

図 3-3 応用例 3 (t) DMAU と μ PD71051 との接続

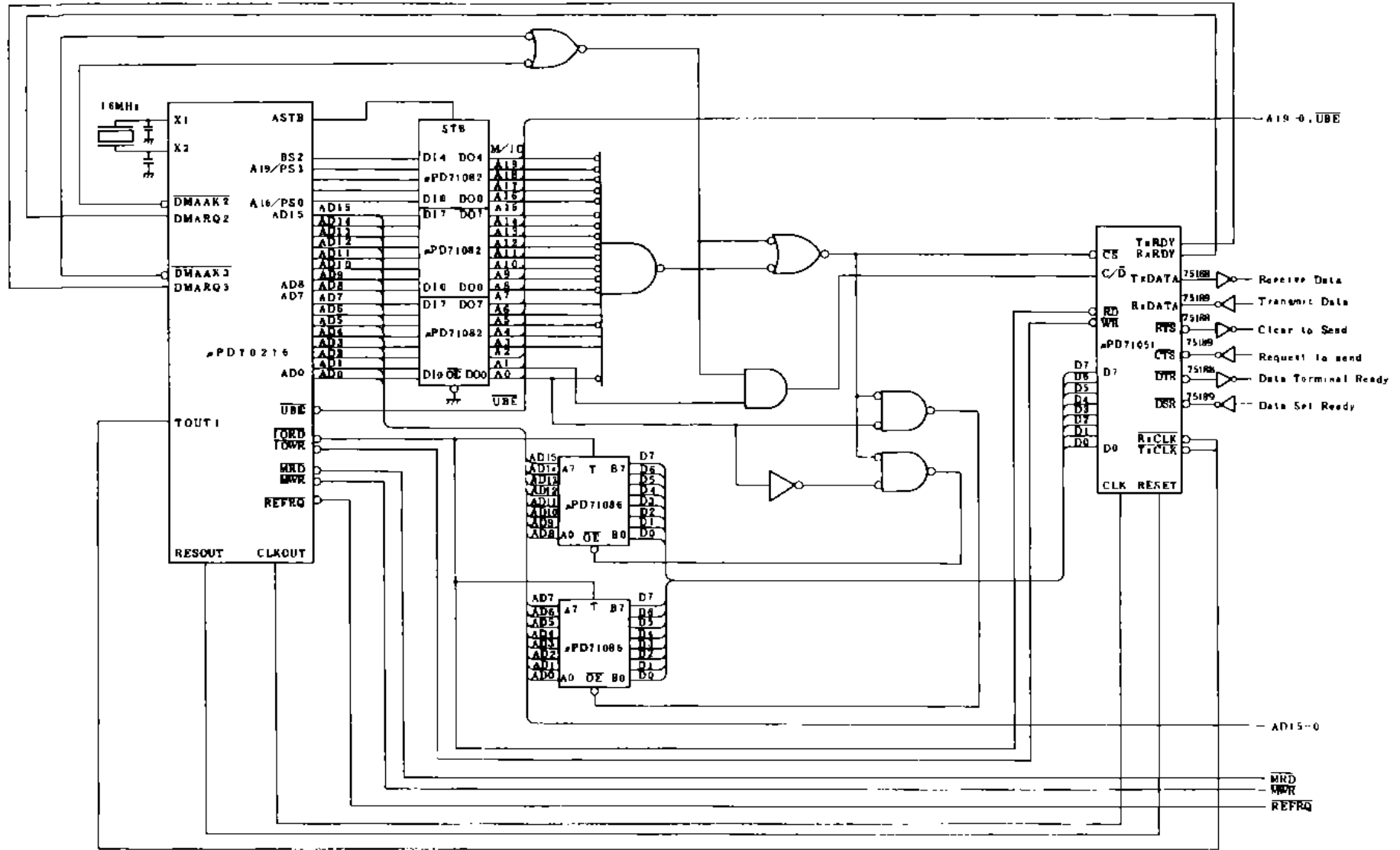
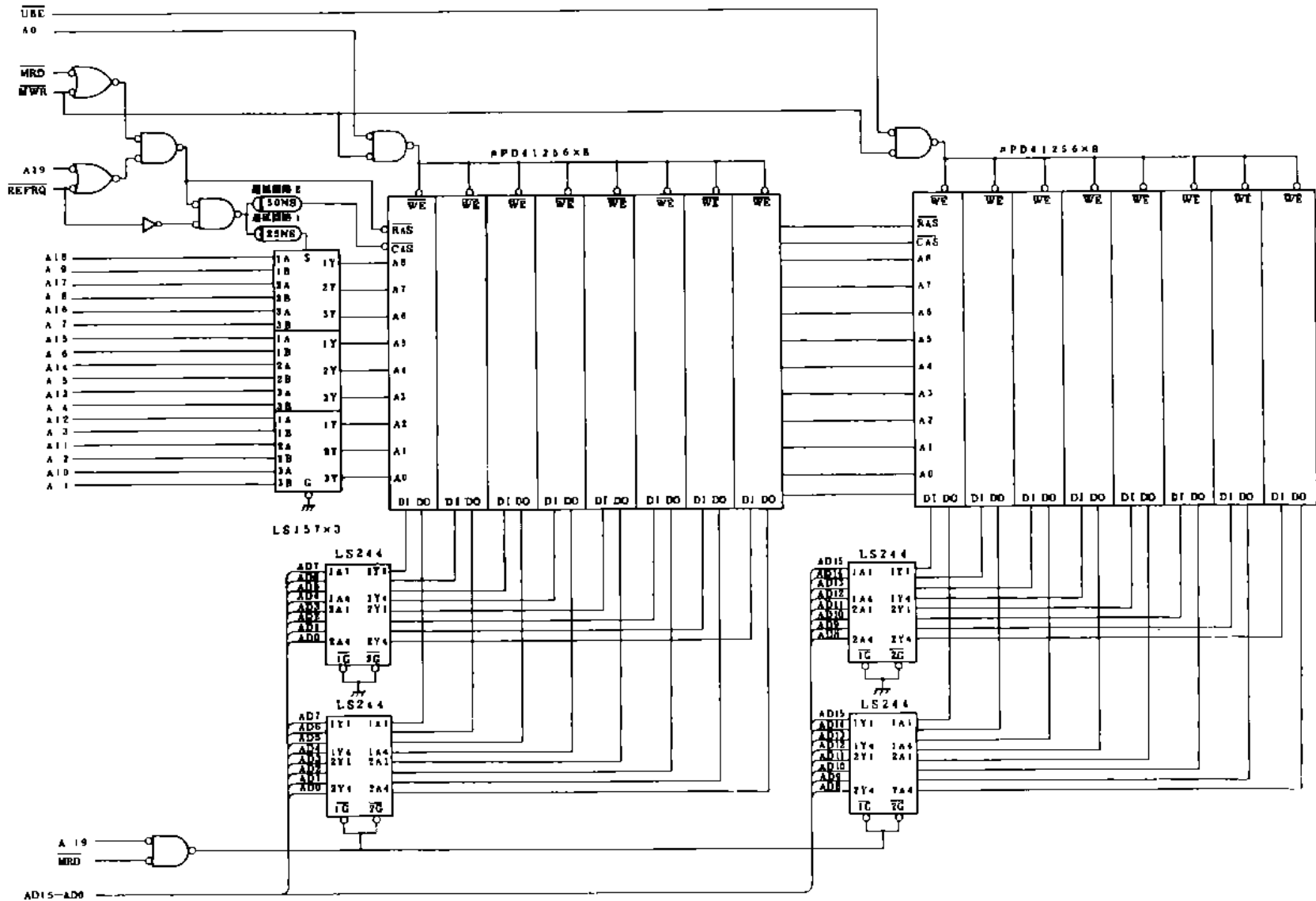
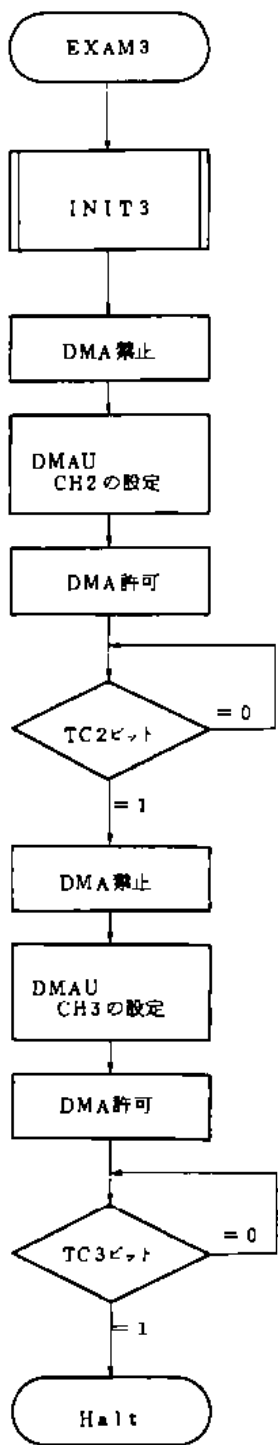


图 3-4 应用例 3(2) DRAM 周边



3.3.4 フローチャート



各 I/O を初期化する

DMAU の設定中は、DMA 転送を禁止する。

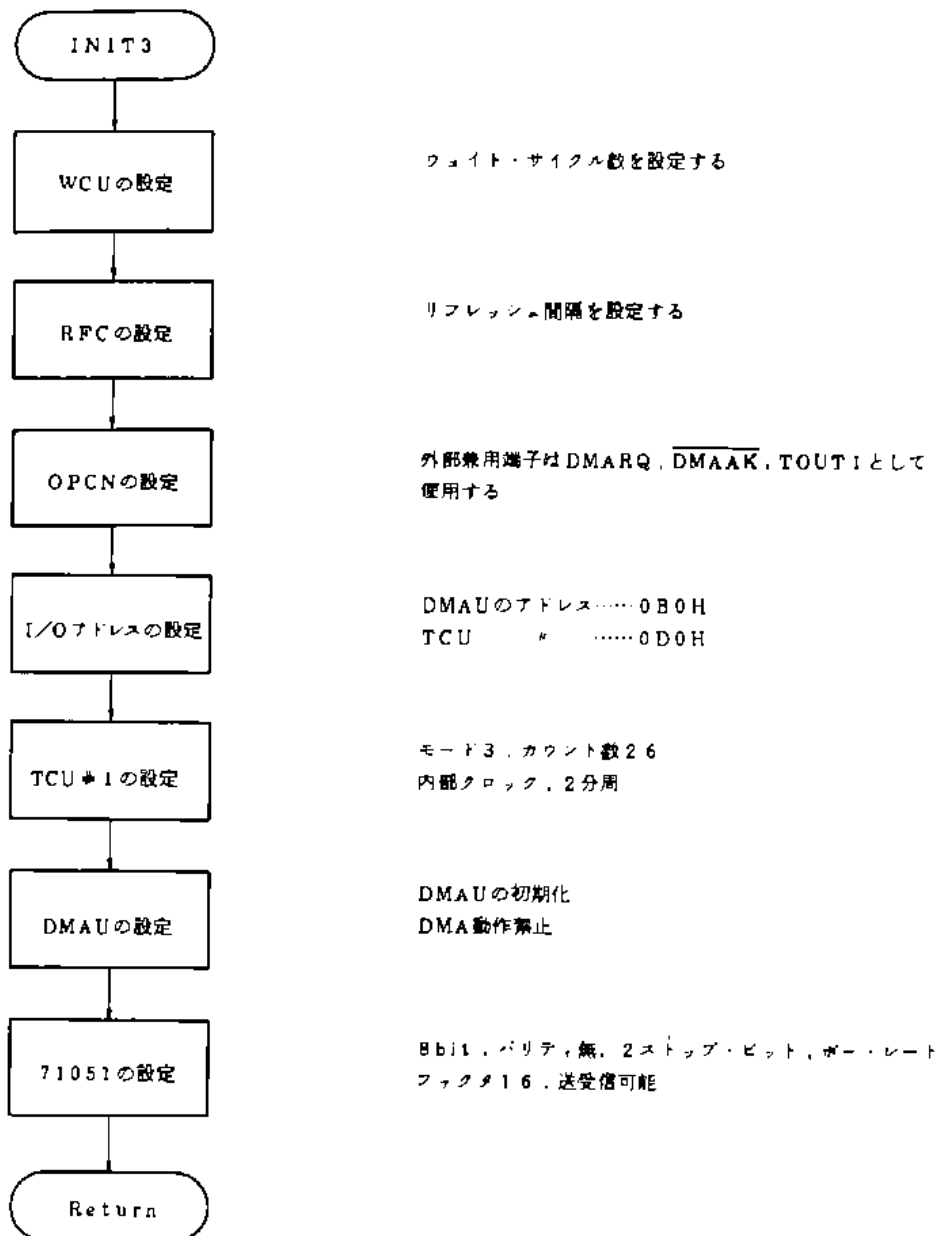
μPD71051 から受信データを読み取るために、チャンネル 2 の動作モード、転送アドレス、カウント数を設定する

チャンネル 2 によるデータ転送（受信）が終了するまでループする

DMAU の設定中は、DMA 転送を禁止する。

μPD71051 へ送信データを書き込むために、チャンネル 3 の動作モード、転送アドレス、カウント数を設定する。

チャンネル 3 によるデータ転送（送信）が終了するまでループする。



3.3.5 プログラム・リスト

LOC	OBJECT	LL#	M	SL#	SOURCE
1		1			;*****
2		2		*	0
3		3		0	PROGRAM EXAMPLE 3
4		4		*	0
5		5		*	DMA TRANSFER & DRAM REFRESH
6		6		*	0
7		7			;*****
8		8			:
9		9			NAME EXAMB
10		10			:
11		11			: DEFINITION INPUT/OUTPUT PORT ADDRESS
12		12			:
13		13			: SYSTEM I/O ADDRESS
14		14			:
15		15		OPCN	EQU OFFFEH : Peripheral Connection
16		16		OPSEL	EQU OFFF0H : Peripheral Selection
17		17		OPHA	EQU OFFF0H : Peripheral High Address
18		18		DULA	EQU OFFFBH : DMAU Low Address
19		19		IULA	EQU OFFFAH : ICU Low Address
20		20		TULA	EQU OFFF0H : TCU Low Address
21		21		SULA	EQU OFFFBH : SCU Low Address
22		22		WCY2	EQU OFFF6H : Wait control 2 (RFW, DMAN)
23		23		WCY1	EQU OFFF5H : Wait control 1 (LMM, MMW, UMW, IOW)
24		24		WMB	EQU OFFF4H : Memory Boundary
25		25		RFC	EQU OFFF2H : Refresh Control
26		26		TCKS	EQU OFFF0H : Timer Clock Selection
27		27			:
28		28			: DMAU I/O Address
29		29			:
30		30		DMAU	EQU 00B0H : DMAU Base Address
31		31		DIW	EQU DMAU : Initialize
32		32		DCR	EQU DMAU+1 : Channel Register
33		33		DCC	EQU DMAU+2 : Count Register
34		34		DCA	EQU DMAU+4 : Address Register
35		35		DDC	EQU DMAU+8 : Device Control Register
36		36		DMD	EQU DMAU+10 : Mode Control Register
37		37		DST	EQU DMAU+11 : Status Register
38		38		DMK	EQU DMAU+15 : Mask Register
39		39			:
40		40			: TCU I/O Address
41		41			:
42		42		TCU	EQU 0000H : TCU Base Address
43		43		TCT0	EQU TCU : TCU#0 Counter
44		44		TSTD	EQU TCU : TCU#0 Status
45		45		TCT1	EQU TCU+2 : TCU#1 Counter
46		46		TST1	EQU TCU+2 : TCU#1 Status
47		47		TCT2	EQU TCU+4 : TCU#2 Counter
48		48		TST2	EQU TCU+4 : TCU#2 Status
49		49		TMD	EQU TCU+6 : TCU Mode Register

LOC	OBJECT	LL#	N	SL#	SOURCE
		50		50	;
		51		51	; EXTERNAL SCU(71051) Address
		52		52	;
		53	53	SDATA EQU 000BH	: Data Register
		54	54	SCMD EQU 000AH	: Mode/Command Register
		55	55	SSTS EQU 000AH	: Status Register
		56		56	;
		57		57	\$EJECT

LOC	OBJECT	LL#	N	SL#	SOURCE
		58	58	PROG3	SEGMENT
		59	59	ASSUME	PS:PROG3, DSO:DATA3
		60	60	:	*****
		61	61	:	* * * * *
		62	62	:	* MAIN ROUTINE 3 * * * * *
		63	63	:	* * * * *
		64	64	:	*****
		65	65		
0000		66	66	EXAM3	PROC
		67	67		
0000	E89100	68	68	CALL	IN173 ; I/O Initialize
		69	69	:	
0003	B004	70	70	MOV	AL,04H ; DMA Disable
0005	E688	71	71	OUT	DDC,AL
		72	72		
0007	B002	73	73	MOV	AL,02H ; Channel 2 Select
0009	E681	74	74	OUT	DCA,AL
		75	75		
0008	B044	76	76	MOV	AL,44H ; Mode Single, I/O to Memory
0000	E68A	77	77	OUT	DMD,AL
		78	78		
000F	BC0B	79	79	MOV	AW,DSO ; Buffer Address Set
0011	BA04	80	80	MOV	DL,AH
0013	C1E004	81	81	SHL	AW,4
0016	0DEA04	82	82	SHR	DL,4
0019	050100	83	83	ADD	AW,OFFSET BUFF
001C	800200	84	84	ADDC	DL,0
001F	E784	85	85	OUT	DCA,AW
0021	8AC2	86	86	MOV	AL,0L
0023	E686	87	87	OUT	DCA+2,AL
		88	88		
0025	BR1300	89	89	MOV	AW,20-1 ; Count=20
0028	E782	90	90	OUT	DCC,AW
		91	91		
002A	E4BF	92	92	IN	AL,DMK ; DMA Mask CH2 Enable
002C	0F1AC002	93	93	CLR1	AL,2
0030	E68F	94	94	OUT	DMK,AL
		95	95		
0032	0F1A06{0000}02	96	96	CLR1	TCFLAG,2 ; TC Flag 2 Off
		97	97		
0038	B000	98	98	MOV	AL,0 ; DMA Enable
003A	E688	99	99	OUT	DDC,AL
		100	100		
		101	101	LOOP1:	
003C	E4BB	102	102	IN	AL,DST ; Loop Until TC2=1
003E	0A06{D000}	103	103	OR	AL,TCFLAG
0042	A2{0000}	104	104	MOV	TCFLAG,AL
0045	0F18C002	105	105	TEST1	AL,2 ; Check TC2 Bit
0049	74F1	106	106	BZ	LOOP1
		107	107		
004B	B004	108	108	MOV	AL,04H ; DMA Disable
004D	E688	109	109	OUT	DDC,AL
		110	110		

LOC	OBJECT	LL#	N	SL#	SOURCE	
004F	B003	111	111	MOV	AL, 03H	; Channel 3 Select
0051	E6B1	112	112	OUT	OCH, AL	
		113	113			
0053	B04B	114	114	MOV	AL, 4BH	; Mode Single, Memory to I/O
0055	E68A	115	115	OUT	DMD, AL	
		116	116			
0057	8CD8	117	117	MOV	AW, D50	; Set Buffer Address
0059	BAD4	118	118	MOV	DL, AH	
005B	C1E004	119	119	SHL	AH, 4	
005E	C0EA04	120	120	SHR	DL, 4	
0061	050100	121	121	ADD	AW, OFFSET BUFF	
0064	800200	122	122	ADDC	DL, 0	
0067	E7B4	123	123	OUT	DCA, AH	
0069	BAC2	124	124	MOV	AL, DL	
006B	E686	125	125	OUT	DCA+2, AL	
		126	126			
006D	B81300	127	127	MOV	AN, 20-1	; Count=20
0070	E7B2	128	128	OUT	DCC, AN	
		129	129			
0072	E4BF	130	130	IN	AL, DMK	; DMA Mask CH3 Enable
0074	DF1AC003	131	131	CLRI	AL, 3	
007B	E6BF	132	132	OUT	DMK, AL	
		133	133			
007A	0F1A06{0000}D3	134	134	CLRI	TCFLAG, 3	; TC3 Flag Off
		135	135			
0080	B000	136	136	MOV	AL, 00H	; DMA Enable
0082	E688	137	137	OUT	DCC, AL	
		138	138			
		139	139	LOOP2:		
0084	E4BB	140	140	IN	AL, DST	; Loop Until TC3=1
0086	0A06{0000}	141	141	OR	AL, TCFLAG	
008A	A2{0000}	142	142	MOV	TCFLAG, AL	
008D	0F1BC003	143	143	TEST1	AL, 3	; Check TC3 Bit
0091	74F1	144	144	BZ	LOOP2	
		145	145			
0093	F4	146	146	HALT		; Program Stop
		147	147			
		148	148	EXAM3	ENDP	
		149	149	\$EJECT		

LOC	OBJECT	LL#	N	SL#	SOURCE
		150	150		;*****
		151	151		;*
		152	152		:* INITIALIZE ROUTINE 3
		153	153		:*
		154	154		;*****
		155	155		
0094		156	156		INIT3 PROC
		157	157		
0094	BAF4FF	158	158		MOV DW, NMB : LMB=512KB, UMB=64KB
0097	B071	159	159		MOV AL, 71H
0099	EE	160	160		OUT DW, AL
		161	161		
009A	BAF5FF	162	162		MOV DW, WCY1 : LMB, UMB=1, IOW, MMW=0
009D	B011	163	163		MOV AL, 11H
009F	EE	164	164		OUT DW, AL
		165	165		
00A0	BAF6FF	166	166		MOV DW, WCY2 : DMAW=2, RFW=1
00A3	B0D9	167	167		MOV AL, 09H
00A5	EE	168	168		OUT DW, AL
		169	169		
00A6	BAF2FF	170	170		MOV DW, RFC : Refresh Enable, N=15(15us, 8MHz)
00A9	B0BE	171	171		MOV AL, BEH
00AB	EE	172	172		OUT DW, AL
		173	173		
00AC	BAFEFF	174	174		MOV DW, OPDN : PF=01(DMARQ3, DMAAK3, TOUT1)
00AF	B001	175	175		MOV AL, 01H
00B1	EE	176	176		OUT DW, AL
		177	177		
00B2	BAFCFF	178	178		MOV DW, OPHA : OPHA= 00H
00B5	B000	179	179		MOV AL, 00H
00B7	EE	180	180		OUT DW, AL
		181	181		
00B8	BAF9FF	182	182		MOV DW, TULA : TCU=0000H
00B9	B000	183	183		MOV AL, TCU
00BD	EE	184	184		OUT DW, AL
		185	185		
00BE	BAFBFF	186	186		MOV DW, DULA : DMAU=00B0H
00C1	B0B0	187	187		MOV AL, DMAU
00C3	EE	188	188		OUT DW, AL
		189	189		
00C4	BAFDFF	190	190		MOV DW, OPSEL : TCU, DMAU Select Active
00C7	B005	191	191		MOV AL, 05H
00C9	EE	192	192		OUT DW, AL
		193	193		
00CA	BAFOFF	194	194		MOV DW, TCKS : Internal, 2 Divide
00CD	B000	195	195		MOV AL, 00H
00CF	EE	196	196		OUT DW, AL
		197	197		

LOC	OBJECT	LL#	M	SL#	SOURCE	
		198		198	:	
		199		199	:	TCU Initialize
		200		200	:	
0000	B056	201		201	MOV	AL, 56H ; Counter #1 Mode 3, LSB
0002	E6D6	202		202	OUT	TMD, AL
		203		203		
0004	B01A	204		204	MOV	AL, 26 ; Count=26
0006	E602	205		205	OUT	TCT1, AL
		206		206	:	
		207		207	:	DMAU Initialize
		208		208	:	
0008	B001	209		209	MOV	AL, 01H ; Initialize
000A	E6B0	210		210	OUT	D1W, AL
		211		211		
000C	B80400	212		212	MOV	AW, 0004H ; DMA Disable
000F	E7B8	213		213	OUT	DDC, AW
		214		214	:	
		215		215	:	External 71051 Initialize
		216		216	:	
00E1	B000	217		217	MOV	AL, 00H ; 71051 Dummy Command
00E3	E6DA	218		218	OUT	SCMD, AL
00E5	E6DA	219		219	OUT	SCMD, AL
00E7	E6DA	220		220	OUT	SCMD, AL
00E9	B040	221		221	MOV	AL, 40H ; Software Reset
00EB	E6DA	222		222	OUT	SCMD, AL
		223		223		
00ED	B0CE	224		224	MOV	AL, 0CEH ; Mode 88bit, No-Parity, 16 Divide.
00EF	E6DA	225		225	OUT	SCMD, AL ; 2-Stop Bit
00F1	B037	226		226	MOV	AL, 37H ; Command Tx/Rx Enable
00F3	E6DA	227		227	OUT	SCMD, AL
		228		228		
00F5	C3	229		229	RET	
		230		230		
		231		231	TMTT3	ENDP
		232		232		
		233		233	PROG3	ENDS
		234		234	\$EJECT	

LOC	OBJECT	LL#	N	SL#	SOURCE
		235	235	DATA3	SEGMENT
		236	236		
0000 (1		237	237	TCFLAG	DBS 1
??					
)					
		238	238		
0001 (20		239	239	BUFF	DBS 20
??					
)					
		240	240		
		241	241	DATA3	ENDS
		242	242		
		243	243	END	EXAM3

— お問い合わせは、最寄りのNECへ —

【営業関係お問い合わせ先】

半導体第一販売事務部 半導体第二販売事務部 半導体第三販売事務部	〒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)
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 半導体第二販売部 半導体第三販売部	〒540 大阪市中央区城見一丁目4番24号 (NEC関西ビル)	大阪 (06) 945-3178 大阪 (06) 945-3200 大阪 (06) 945-3208
北海道支社 札幌 東北支社 仙台 岩手支店 盛岡 山形支店 山形 郡山支店 郡山 いわき支店 いわき 新潟支店 新潟 土浦支店 土浦 水戸支店 水戸 神奈川支社 横浜 群馬支店 高崎 太田支店 太田	札幌 (011)231-0161 仙台 (022)261-5511 盛岡 (0186)51-4344 山形 (0236)23-5511 郡山 (0240)23-5511 いわき (0240)21-5511 新潟 (0250)30-2155 土浦 (0280)23-6161 水戸 (0282)26-1717 横浜 (045)324-5511 高崎 (0273)28-1255 太田 (0276)46-4011	宇都宮支店 宇都宮 (0288)21-2201 小山支店 小山 (0285)24-5011 長野支社 長野 (0262)35-1444 松本支店 松本 (0263)35-1666 諏訪支店 諏訪 (0268)63-6350 甲府支店 甲府 (0552)24-4141 埼玉支店 大宮 (048)641-1411 立川支店 立川 (0425)20-5981 千葉支店 千葉 (043)238-0116 神岡支店 神岡 (054)255-2211 北陸支店 金沢 (0762)23-1621 福井支店 福井 (0776)22-1666
富山支店 富山 三浦支店 津島 京都支社 京都 神戸支社 神戸 中国支店 広島 鳥取支店 鳥取 岡山支店 岡山 四国支店 高松 新居浜支店 新居浜 松山支店 松山 九州支社 福岡 北九州支店 北九州	(0764)31-8481 (0592)25-7341 (075)344-7824 (078)333-3854 (082)242-5504 (0857)27-5311 (086)225-4456 (0878)36-1200 (0897)32-5001 (0898)45-4111 (092)271-7700 (093)541-2887	

【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営業本部 マイクロコンピュータ技術部	〒210 川崎市幸区堀越三丁目484番地	川崎 (044)548-8880	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)
半導体販売技術本部 東日本販売技術部	〒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	
半導体販売技術本部 中部販売技術部	〒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	
半導体販売技術本部 西日本販売技術部	〒540 大阪市中央区城見一丁目4番24号 (NEC関西ビル)	大阪 (06) 945-3383	