

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。



ユーザーズ・マニュアル

μPD77210 ファミリ

デジタル・シグナル・プロセッサ

アーキテクチャ編

μPD77210

μPD77213

資料番号 U15807JJ3V0UM00 (第3版)

発行年月 March 2004 NS CP(K)

© NEC Electronics Corporation 2001

[メ モ]

目 次 要 約

第 1 章	概 要	...	19
第 2 章	端子機能	...	24
第 3 章	使用方法概論	...	39
第 4 章	アーキテクチャ	...	43
第 5 章	ペリフェラル	...	122
第 6 章	ブートアップ機能	...	241
付録 A	索 引	...	250

入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。

CMOSデバイスの入力がノイズなどに起因して、 V_{IL} (MAX.) から V_{IH} (MIN.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定な場合はもちろん、 V_{IL} (MAX.) から V_{IH} (MIN.) までの領域を通過する遷移期間中にチャタリングノイズ等が入らないようご使用ください。

未使用入力の処理

CMOSデバイスの未使用端子の入力レベルは固定してください。

未使用端子入力については、CMOSデバイスの入力に何も接続しない状態で動作させるのではなく、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、未使用の入出力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。

静電気対策

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、当社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。

初期化以前の状態

電源投入時、MOSデバイスの初期状態は不定です。

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。

本製品のうち、外国為替及び外国貿易法の規定により規制貨物等（または役務）に該当するものについては、日本国外に輸出する際に、同法に基づき日本国政府の輸出許可が必要です。

非該当品 : μ PD77210F1-DA2, μ PD77210GJ-8EN

ユーザ判定品 : μ PD77213F1-xxx-DA2, μ PD77213GJ-xxx-8EN

- 本資料に記載されている内容は2004年3月現在のものです、今後、予告なく変更することがあります。量産設計の際には最新の個別データ・シート等をご参照ください。
- 文書による当社の事前の承諾なしに本資料の転載複製を禁じます。当社は、本資料の誤りに関し、一切その責を負いません。
- 当社は、本資料に記載された当社製品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、一切その責を負いません。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。
- 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、一切その責を負いません。
- 当社は、当社製品の品質、信頼性の向上に努めておりますが、当社製品の不具合が完全に発生しないことを保証するものではありません。当社製品の不具合により生じた生命、身体および財産に対する損害の危険を最小限度にするために、冗長設計、延焼対策設計、誤動作防止設計等安全設計を行ってください。
- 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いただく「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ください。

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、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

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。意図されていない用途で当社製品の使用をお客様が希望する場合には、事前に当社販売窓口までお問い合わせください。

(注)

- (1) 本事項において使用されている「当社」とは、NECエレクトロニクス株式会社およびNEC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う。
- (2) 本事項において使用されている「当社製品」とは、(1)において定義された当社の開発、製造製品をいう。

本版で改訂された主な箇所

箇所	内容
p.59	4.3.4(1)(a) 制限事項を追加。
p.88	4.4.4(9) 割り込みの制限事項を追加
p.132	表 5-4 SST1 のビット構成 SSER, SLER, SLEF ビットに注を追加。
p.136	5.3.4 シリアル・ステータス・レジスタのリセット機能についてを追加。
p.140	表 5-8 ASST のビット構成 ASOEN, ASSER, ASLER, ASLEF ビットに注を追加。
p.142	5.4.3 オーディオ・シリアル・インタフェースのタイミングに説明を追加。
p.144	5.4.5 オーディオ・シリアル・ステータス・レジスタのリセット機能についてを追加。
p.147	5.5.2(2) SST1 に説明を追加。
p.150	表 5-11 SST のビット構成 SSER, SLER, SLEF ビットに注を追加。
p.160	5.5.5 シリアル・ステータス・レジスタのリセット機能についてを追加。
p.162, 163	5.6.1(6) HRE, (7) HWE に説明を追加。
p.186	5.7.7 メモリ・アクセスのタイミングに説明を追加。
p.186	図 5-23 MWAIT 端子によるアクセス・ウェイトの挿入のタイミングを修正。
p.190	表 5-22 PCM のビット構成に注意を追加。
p.215	5.12.2 クロックの切り替えタイミングに説明を追加。
p.224	図 5-42 POWC レジスタの設定例 を変更。
p.240	5.18.5 SD Card IF SD カード使用時の注意事項 (μ PD77213 のみ) を追加。
p.249	6.3 リポートに注意 6 を追加。

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。

はじめに

対象者 このマニュアルは、 μ PD77210 ファミリの機能を理解し、それをを用いたソフトウェア、ハードウェアなどのアプリケーション・システムを設計するユーザを対象とします。

μ PD77210 ファミリは、 μ PD77210, 77213, 77214 の総称です。特に機能面に違いがない場合は、 μ PD77210 ファミリを該当する製品に読み替えてご使用ください。機能面に違いがある場合は、製品名をあげて説明しています。 μ PD77214 は、 μ PD77213 の機能拡張品です。 μ PD77214 をご使用になる場合は、このマニュアルと、 μ PD77214 ユーザーズ・マニュアル アーキテクチャ編をあわせてご覧ください。

目的 このマニュアルは、次の構成に示す μ PD77210 ファミリの持つハードウェア、ソフトウェア機能をユーザに理解していただき、これらのデバイスを使用するシステムのハードウェア、ソフトウェア開発の参照用資料として役立つことを目的としています。

構成 このマニュアルでは、大きく分けて次の内容で構成しています。

- 第1章 概説
- 第2章 端子機能
- 第3章 使用方法概論
- 第4章 アーキテクチャ
- 第5章 ペリフェラル
- 第6章 ブートアップ機能
- 付録A 索引

読み方 このマニュアルを読むにあたっては、電気、論理回路、マイクロコンピュータに関する一般的知識が必要となります。

μ PD77210 ファミリの基本機能を一通り理解しようとするとき

第1章 概説から順にお読みください。

ハードウェア・エンジニア/ソフトウェア・エンジニアの方へ

第1章 概説から第6章 ブートアップ機能をお読みください。第4章 アーキテクチャでは、デバイス内部の各機能ブロックについて説明しています。第5章 ペリフェラルでは、各インタフェース、タイマ、割り込みなど、ペリフェラル機能について説明しています。

レファレンス・マニュアルとして利用する方へ

このマニュアルの最後に索引があります。キーワードをもとに検索参照するときには使用してください。第5章 ペリフェラルでは、各インタフェース、タイマ、割り込みなど、ペリフェラル機能について説明しています。

第3章 使用方法概論では、 μ PD77210 ファミリの電源投入からユーザ・プログラム実行までの流れ、および全体の使い方について説明しています。 μ PD77210 ファミリをご使用になる前に、必ずお読みください。

μ PD77214 の拡張機能については、 μ PD77214 ユーザーズ・マニュアル アーキテクチャ編をご覧ください。

μ PD77210 ファミリの命令語は、 μ PD77016 ファミリ ユーザーズ・マニュアル アーキテクチャ編をご覧ください。

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	: <u>xxx</u> (端子, 信号の名称に上線)
	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	: 本文中の補足説明
	数の表記	: 2進数...xxx または 0bxxx 10進数...xxx 16進数...0xxx

関連資料 関連資料は暫定版の場合がありますが、この資料では「暫定」の表示をしておりません。あらかじめご了承ください。

μ PD77210 ファミリに関する資料

資料名 品名	パンフレット	データ・シート	ユーザーズ・マニュアル		アプリケーション・ノート	
			アーキテクチャ編	命令編	基本ソフトウェア編	ライブラリ編
μ PD77210	U12395J	U15203J	このマニュアル	U13116J	U11958J	U12021J
μ PD77213						
μ PD77214	-	S16075J	U16631J			

開発ツールに関する資料

資料名	資料番号	
HSM77016 ユーザーズ・マニュアル	U11602J	
WB77016 ユーザーズ・マニュアル	言語編	U10078J
	操作編	U11506J
ID77016 ユーザーズ・マニュアル	U10118J	
CC77016 ユーザーズ・マニュアル	U15037J	
RX77016 ユーザーズ・マニュアル	機能編	U14397J
	コンフィギュレーション・ツール編	U14404J
RX77016 アプリケーション・ノート	HOST API 編	U14371J

ミドルウェアに関する資料

資料名	資料番号
μ SAP77016-B01 ユーザーズ・マニュアル	U13130J
μ SAP77016-B03 ユーザーズ・マニュアル	U13373J
μ SAP77016-B04 ユーザーズ・マニュアル	U13955J
μ SAP77016-B05 ユーザーズ・マニュアル	U14497J
μ SAP77016-B06 ユーザーズ・マニュアル	U15165J
μ SAP77016-B07 ユーザーズ・マニュアル	U15134J
μ SAP77016-B08 ユーザーズ・マニュアル	U15152J
μ SAP77016-B09 ユーザーズ・マニュアル	U15611J
μ SAP77016-B10 ユーザーズ・マニュアル	U15518J
μ SAP77016-B11 ユーザーズ・マニュアル	U15683J
μ SAP77016-B13 ユーザーズ・マニュアル	U16392J
μ SAP77016-B19 ユーザーズ・マニュアル	U16553J
μ SAP77016-B21 ユーザーズ・マニュアル	U16392J
μ SAP77016-B22 ユーザーズ・マニュアル	U16554J

注意 上記関連資料は、予告なしに内容を変更することがあります。設計などには、必ず最新の資料をご使用ください。

目 次

第1章 概 要 … 19

- 1.1 μ PD77111 ファミリとの比較 … 19
 - 1.1.1 高速動作 … 19
 - 1.1.2 低消費電力 … 19
 - 1.1.3 ペリフェラル機能の拡張 … 19
- 1.2 特 徴 … 20
 - 1.2.1 DSP コア・カーネル部 … 20
 - 1.2.2 ペリフェラル部 … 20
 - 1.2.3 オーダ情報 … 23

第2章 端子機能 … 24

- 2.1 端子接続 … 25
 - 2.1.1 161 ピン・プラスチック・ファインピッチ BGA (10×10mm) … 25
 - 2.1.2 144 ピン・プラスチック LQFP (ファインピッチ)(20×20mm) … 27
- 2.2 端子構成 … 29
- 2.3 端子機能 … 30
- 2.4 未使用端子の処理について … 37

第3章 使用方法概要 … 39

- 3.1 電源投入からプログラム実行まで … 39
 - 3.1.1 電源投入 … 40
 - 3.1.2 リセット … 40
 - 3.1.3 ブート … 40
 - 3.1.4 プログラム実行 … 40
- 3.2 スタンバイ・モード … 41
- 3.3 ブート後のクロック設定 … 41

第4章 アーキテクチャ … 43

- 4.1 全体ブロック構成 … 43
- 4.2 バ ス … 46
 - 4.2.1 メイン・バス … 46
 - 4.2.2 データ・バス … 47
 - 4.2.3 ペリフェラル - メモリ間転送バス … 50
- 4.3 システム制御ユニット … 50
 - 4.3.1 クロック・ジェネレータ … 51
 - 4.3.2 リセット機能 … 52
 - 4.3.3 パイプライン・アーキテクチャ … 56
 - 4.3.4 スタンバイ機能 … 58
- 4.4 プログラム制御ユニット … 60

- 4.4.1 **ブロック構成** … 60
- 4.4.2 **プログラム実行制御部** … 61
- 4.4.3 **フロー制御部** … 68
- 4.4.4 **割り込み** … 76
- 4.4.5 **エラー・ステータス・レジスタ (ESR : error status register)** … 89
- 4.5 **データ・アドレッシング・ユニット** … 90
 - 4.5.1 **ブロック構成** … 90
 - 4.5.2 **データ・メモリ空間** … 90
 - 4.5.3 **命令メモリのエイリアス** … 94
 - 4.5.4 **外部データ・メモリ・マップ** … 95
 - 4.5.5 **アドレッシング・モード** … 96
- 4.6 **演算ユニット** … 107
 - 4.6.1 **ブロック構成** … 108
 - 4.6.2 **汎用レジスタとデータ・フォーマット** … 108
 - 4.6.3 **累乗算器 (MAC) と MAC 入力シフト (MSFT) の演算機能** … 111
 - 4.6.4 **ALU 演算機能** … 117
 - 4.6.5 **BSFT シフト演算機能** … 120

第5章 ペリフェラル … 122

- 5.1 **ブロック構成** … 123
- 5.2 **ペリフェラル・レジスタ** … 124
- 5.3 **時分割 (TDM) シリアル・インタフェース (TSIO)** … 127
 - 5.3.1 **時分割シリアル・インタフェースの端子** … 128
 - 5.3.2 **時分割シリアル・インタフェースのレジスタ** … 129
 - 5.3.3 **時分割シリアル・インタフェースのタイミング** … 133
 - 5.3.4 **シリアル・ステータス・レジスタのリセット機能について** … 136
- 5.4 **オーディオ・シリアル・インタフェース (ASIO)** … 136
 - 5.4.1 **オーディオ・シリアル・インタフェースの端子** … 138
 - 5.4.2 **オーディオ・シリアル・インタフェースのレジスタ** … 139
 - 5.4.3 **オーディオ・シリアル・インタフェースのタイミング** … 142
 - 5.4.4 **起動時の ASIO に関する注意事項** … 144
 - 5.4.5 **オーディオ・シリアル・ステータス・レジスタのリセット機能について** … 144
- 5.5 **スタンダード・シリアル・インタフェース (SIO)** … 145
 - 5.5.1 **スタンダード・シリアル・インタフェースの端子** … 146
 - 5.5.2 **スタンダード・シリアル・インタフェースのレジスタ** … 147
 - 5.5.3 **スタンダード・シリアル・インタフェースのタイミング** … 152
 - 5.5.4 **起動時の TSO に関する注意事項** … 160
 - 5.5.5 **シリアル・ステータス・レジスタのリセット機能について** … 160
- 5.6 **ホスト・インタフェース (HIO)** … 161
 - 5.6.1 **ホスト・インタフェースの端子** … 162
 - 5.6.2 **ホスト・インタフェースのレジスタ** … 163
 - 5.6.3 **ホスト CPU から見たホスト・インタフェースのレジスタ** … 165
 - 5.6.4 **ホスト・インタフェースのタイミング** … 166
- 5.7 **外部データ・メモリ・インタフェース (MIO)** … 172
 - 5.7.1 **メモリ・インタフェースの端子** … 174
 - 5.7.2 **外部データ・メモリ・インタフェースのレジスタ** … 174
 - 5.7.3 **ダイレクト・アクセス** … 180

- 5.7.4 DMA アクセス … 181
- 5.7.5 ダイレクト・アクセスのタイミング … 183
- 5.7.6 DMA アクセスのタイミング … 184
- 5.7.7 メモリ・アクセスのタイミング … 185
- 5.8 **ペリフェラル - メモリ間転送 (PMT)** … 187
 - 5.8.1 PMT のレジスタ … 189
 - 5.8.2 PMT の動作モード … 191
 - 5.8.3 PMT 転送手順 … 192
- 5.9 **汎用入出力ポート (PIO)** … 193
 - 5.9.1 汎用入出力ポートの端子 … 193
 - 5.9.2 汎用入出力ポートのレジスタ … 194
 - 5.9.3 汎用入出力ポートのタイミング … 196
 - 5.9.4 ポート・プログラミングの例 … 199
- 5.10 **割り込みコントローラ (INTC)** … 200
 - 5.10.1 割り込みコントローラのレジスタ … 201
 - 5.10.2 割り込みコントローラの動作モード … 202
 - 5.10.3 割り込みテーブル … 203
 - 5.10.4 外部割り込み端子による割り込みのハードウェア条件 … 204
 - 5.10.5 割り込みに関する注意点 … 205
- 5.11 **タイマ (TIM)** … 208
 - 5.11.1 タイマの端子 … 209
 - 5.11.2 タイマのレジスタ … 209
 - 5.11.3 タイマの動作 … 211
 - 5.11.4 タイマ使用上の注意 … 211
- 5.12 **クロック・コントローラ (CLKC)** … 212
 - 5.12.1 クロック・コントローラのレジスタ … 213
 - 5.12.2 クロックの切り替えタイミング … 215
 - 5.12.3 クロック制御の注意点 … 216
- 5.13 **命令メモリ訂正機能 (IMC)** … 220
 - 5.13.1 命令メモリ訂正機能のレジスタ … 220
 - 5.13.2 命令訂正機能の動作 … 221
- 5.14 **ページング機能** … 221
 - 5.14.1 ページング機能のレジスタ … 222
 - 5.14.2 ページング機能の動作 … 222
- 5.15 **ペリフェラル STOP モード** … 223
 - 5.15.1 ペリフェラル STOP モードのレジスタ … 223
 - 5.15.2 ペリフェラル STOP モードの動作 … 224
- 5.16 **ディバグ・インタフェース (IEIO)** … 224
 - 5.16.1 JTAG ポート … 224
 - 5.16.2 ディバグ・インタフェースの端子 … 225
 - 5.16.3 バウンダリ・スキャン・テスト機能 … 226
 - 5.16.4 ディバグ機能 (インサーキット・エミュレータ機能) … 227
- 5.17 **拡張インタフェース (Additional IO)** … 228
 - 5.17.1 拡張インタフェースのレジスタ … 229
- 5.18 **SD カード・インタフェース (SDCIF)** … 230
 - 5.18.1 SD カード・インタフェースの端子 … 231
 - 5.18.2 SD カード・インタフェースのレジスタ … 232
 - 5.18.3 CRC (Cyclic Redundancy Codes) 回路 … 237
 - 5.18.4 SD カード・インタフェースの動作 … 238

第 6 章 ブートアップ機能 ... 241

- 6.1 **イニシャル・リセット・ブート ... 241**
 - 6.1.1 ブート・モード指定 ... 241
 - 6.1.2 Xメモリ・ブート ... 242
 - 6.1.3 Yメモリ・ブート ... 243
 - 6.1.4 XYメモリ・ブート ... 243
 - 6.1.5 外部データ・メモリ・ブート ... 244
 - 6.1.6 ホスト・ブート ... 245
 - 6.1.7 シリアル・ブート ... 247
 - 6.1.8 ノン・ブート ... 247
- 6.2 **イニシャル・リセット・ブートと PLL ... 248**
- 6.3 **リブート ... 249**

付録 A 索引 ... 250

- A.1 **五十音で始まる語句の索引 ... 250**
- A.2 **アルファベットで始まる語句の索引 ... 252**
- A.3 **レジスタ索引 ... 253**
 - A.3.1 レジスタ名称順 ... 253
 - A.3.2 レジスタ略号順 ... 255

図の目次 (1/3)

図番号	タイトル, ページ
2-1	161 ピン・プラスチック・ファインピッチ BGA … 25
2-2	144 ピン・プラスチック LQFP … 27
2-3	端子構成図 … 29
3-1	電源投入からユーザ・プログラムの実行まで … 39
3-2	PLL の起動と停止 … 42
4-1	全体ブロック構成 … 44
4-2	DSP コア・カーネル … 45
4-3	リセット動作タイミング … 53
4-4	パイプライン・イメージ … 56
4-5	プログラム制御ユニット … 60
4-6	命令メモリ・マップ … 62
4-7	PC の通常動作 … 63
4-8	無条件イミューディエト・ジャンプのタイミング … 66
4-9	無条件間接ジャンプのタイミング … 66
4-10	条件イミューディエト・ジャンプのタイミング (条件成立:分岐) … 67
4-11	条件イミューディエト・ジャンプのタイミング (条件不成立:通過) … 67
4-12	RC のフォーマット … 70
4-13	リピート命令の例 (2 回リピート) … 71
4-14	リピート実行のタイミング (2 回リピート) … 71
4-15	LC のフォーマット … 72
4-16	ループ実行のタイミング (2 回ループ動作例) … 74
4-17	多重割り込みの概念 … 82
4-18	割り込み受け付けタイミング … 84
4-19	RETI 命令によるタイミング … 85
4-20	割り込み遅延タイミング (1 サイクル遅延) … 86
4-21	割り込み遅延タイミング (2 サイクル遅延) … 87
4-22	データ・アドレッシング・ユニット … 90
4-23	データ・メモリ・マップ … 91
4-24	命令メモリのデータ・メモリへのエイリアス … 94
4-25	外部データ・メモリ・マップのアクセス・イメージ … 96
4-26	DPn のビット反転 … 100
4-27	DPn の分割 … 102
4-28	通常のもジュロ演算の写像 … 103
4-29	もジュロ調整の写像 … 103
4-30	演算ユニットの詳細ブロック … 108
4-31	汎用レジスタの分割フォーマット … 109
4-32	汎用レジスタとデータ・メモリ間のデータ転送 … 110

図の目次 (2/3)

図番号	タイトル, ページ
4-33	有符号値同士の乗算 … 112
4-34	有符号値と無符号値の乗算 … 113
4-35	無符号値同士の乗算 … 113
4-36	累乗算 … 115
4-37	1 ビット・シフト累乗算 … 116
4-38	16 ビット・シフト累乗算 … 117
4-39	BSFT 演算 … 121
5-1	ペリフェラル・ユニット … 123
5-2	時分割シリアル・インタフェースのブロック … 127
5-3	スロット … 134
5-4	スロット拡張 … 134
5-5	時分割シリアル・インタフェースのタイミング … 135
5-6	オーディオ・シリアル・インタフェースのブロック … 137
5-7	オーディオ・シリアル・インタフェースの出力タイミング … 142
5-8	オーディオ・シリアル・インタフェースの入力タイミング … 143
5-9	ASIO 動作開始時の ASO 動作 (マスタ・モードの場合) … 144
5-10	スタンダード・シリアル・インタフェース … 145
5-11	シリアル・インタフェース出力タイミング … 153
5-12	シリアル・インタフェース入力タイミング … 155
5-13	シリアル・インタフェース: シリアル・クロック・カウンタの動作 … 156
5-14	ホスト・インタフェースのブロック図 … 161
5-15	ホスト・リード手順 (μ PD77210 ファミリ ホスト): HDT 書き込みウエイトなし … 167
5-16	ホスト・ライト手順 (μ PD77210 ファミリ ホスト): HDT 読み出しウエイトなし … 168
5-17	外部データ・メモリ・インタフェースのブロック図 … 173
5-18	ダイレクト・アクセス … 180
5-19	DMA アクセス … 182
5-20	ダイレクト・アクセスのタイミング … 183
5-21	DMA アクセスのタイミング … 184
5-22	メモリ・アクセスのタイミング … 185
5-23	$\overline{\text{MWAIT}}$ 端子によるアクセス・ウエイトの挿入 … 186
5-24	バス・アービトレーション … 186
5-25	PMT のブロック図 … 187
5-26	PMT の設定 … 191
5-27	汎用入出力ポートのブロック図 … 193
5-28	スタンプ・モード時の割り込みコントローラ … 202
5-29	マスク・モード時の割り込みコントローラ … 203
5-30	外部割り込みタイミング … 204
5-31	2 種類の割り込みマスク・フラグの関係 … 205

図の目次 (3/3)

図番号	タイトル, ページ
5-32	fint 命令使用例: オーディオ・シリアル入力割り込み使用, 32 ビット・モード時 ... 206
5-33	タイマのブロック図 (1 チャンネル分) ... 209
5-34	クロック・コントローラのブロック図 ... 212
5-35	クロック切り替えタイミング ... 215
5-36	クロックの切り替え ... 216
5-37	PLL 動作と分周動作の状態遷移 ... 217
5-38	PLL の設定例 ... 218
5-39	分周器の設定例 ... 218
5-40	クロック・モニタ出力の状態遷移 ... 219
5-41	クロック・モニタ出力の設定例 ... 219
5-42	POWC レジスタの設定例 ... 224
5-43	JTAG 端子処理 ... 227
5-44	拡張インタフェースのブロック図 ... 228
5-45	SD カード・インタフェースのブロック図 ... 230
6-1	X メモリからのブートアップ ... 242
6-2	Y メモリからのブートアップ ... 243
6-3	X メモリおよび Y メモリからの並列ブートアップ ... 243
6-4	外部データ・メモリからのブートアップ ... 244
6-5	ホスト・ブートアップ ... 245
6-6	シリアル・ブートアップ ... 247

表の目次 (1/2)

表番号	タイトル, ページ
1-1	μ PD77210 ファミリの特徴 ... 22
2-1	機能端子の処理 ... 37
2-2	非機能端子の処理 ... 38
4-1	メイン・バス接続レジスタ ... 47
4-2	機能ブロックとバス ... 48
4-3	X データ・バス接続レジスタおよびメモリ ... 49
4-4	Y データ・バス接続レジスタおよびメモリ ... 50
4-5	PLL 逡倍率設定 ... 51
4-6	PLL ロック・レンジ設定 ... 52
4-7	初期化対象 CPU レジスタと初期化値 ... 54
4-8	初期化対象ペリフェラル・レジスタと初期化値 ... 55
4-9	初期化対象端子と初期状態 ... 55
4-10	ホールド中の端子状態 ... 58
4-11	内部命令メモリの容量 ... 61
4-12	分岐命令の分類 ... 65
4-13	割り込みベクタ・テーブル ... 77
4-14	ROM と RAM の容量 ... 92
4-15	外部データ・メモリの容量 ... 92
4-16	X, Y メモリ空間への同時アクセスの可否 ... 93
4-17	データ・ポインタのモディファイ一覧 ... 101
4-18	汎用レジスタのフォーマット ... 109
4-19	累乗算機能 ... 114
5-1	ペリフェラル・レジスタのメモリ・マッピング ... 124
5-2	時分割シリアル・インタフェースのレジスタ ... 128
5-3	TSST のビット構成 ... 130
5-4	SST1 のビット構成 ... 131
5-5	SICM ビットと SIEF ビットの組み合わせ ... 132
5-6	TFMT のビット構成 ... 133
5-7	オーディオ・シリアル・インタフェースのレジスタ ... 137
5-8	ASST のビット構成 ... 140
5-9	スタンダード・シリアル・インタフェースのレジスタ ... 145
5-10	シリアル入出力エラー・フラグ設定条件 ... 147
5-11	SST のビット構成 ... 149
5-12	SICM ビットと SIEF ビットの組み合わせ ... 151
5-13	ホスト・インタフェースのレジスタ ... 161
5-14	HST のビット構成 ... 164

表の目次 (2/2)

表番号	タイトル, ページ
5-15	ホスト・インタフェース用レジスタの選択 … 165
5-16	外部データ・メモリ・インタフェースのレジスタ … 173
5-17	MSHW のビット構成 … 177
5-18	MWAIT のビット構成 … 178
5-19	MCST のビット構成 … 179
5-20	PMT のレジスタ … 188
5-21	PMT 転送チャネルと対象ペリフェラル … 189
5-22	PMC のビット構成 … 190
5-23	汎用入出力ポートのレジスタ … 193
5-24	PCD のビット構成 … 194
5-25	割り込みコントローラのレジスタ … 200
5-26	ICR のビット構成 … 201
5-27	割り込みテーブル … 204
5-28	タイマのレジスタ … 209
5-29	TCSR のビット構成 … 210
5-30	タイマのクロック・ソース (TCSR のビット 5-3) … 211
5-31	クロック・コントローラのレジスタ … 213
5-32	CLKC のビット構成 … 213
5-33	命令メモリ訂正機能のレジスタ … 220
5-34	CEFR のビット構成 … 221
5-35	ページング機能のレジスタ … 222
5-36	DPR の設定と対象領域 … 222
5-37	ペリフェラル STOP モードのレジスタ … 223
5-38	POWC のビット構成 … 223
5-39	テスト・インストラクション … 226
5-40	拡張インタフェースのレジスタ … 228
5-41	APCR のビット構成 … 229
5-42	SD カード・インタフェースのレジスタ … 231
5-43	SDCMD_IDX のビット構成 … 234
5-44	SDCTL のビット構成 … 235
5-45	SDRPR レジスタの構成 … 237
5-46	μPD77213 の APCR レジスタのビット構成 … 238
6-1	イニシャル・リセット・ブート・モード … 242
6-2	PLL のロック・レンジ … 242
6-3	ブート・モードと PLL … 248
6-4	リポート・エントリ・アドレスとパラメータ … 249

第1章 概 要

μ PD77210 ファミリは、16 ビット固定小数点デジタル・シグナル・プロセッサ（以下、DSP）である μ PD 77111 ファミリの後継として位置づけられます。 μ PD77210 ファミリは、 μ PD77210, 77213, 77214 の総称です。

高速低消費電力性能を持ち、モバイル・アプリケーション・セットにおいて音声処理にとどまらず、音楽、画像などの各種メディアの信号処理をターゲットとしています。

μ PD77214 をご使用になる場合は、このマニュアルのほかに、 μ PD77214 ユーザーズ・マニュアル アーキテクチャ編を参照してください。

備考 μ PD77111 ファミリは、 μ PD77110, 77111, 77112, 77113A, 77114, 77115 の総称です。

1.1 μ PD77111 ファミリとの比較

μ PD77210 ファミリは、0.13 μ m プロセスを採用することにより、実行速度は μ PD77111 ファミリの2倍、消費電力は1/2になっています。さらに、ペリフェラル機能を拡張しています。

1.1.1 高速動作

最大動作周波数を 120 MHz (μ PD77213) ~ 160 MHz (μ PD77210) まで引き上げています。

複数チャンネルの音声コーデックや、音声 / 画像コーデックを 1 チップで実現可能です。

1.1.2 低消費電力

MAC 演算（並列ロード / ストア）+ NOP 実行時の消費電流を 0.35 mA/MHz としています。動作電圧は 1.5 V です（ただし μ PD77210 の 120 ~ 160 MHz 動作時は 1.6 V となります）。

1.1.3 ペリフェラル機能の拡張

μ PD77210 ファミリは、 μ PD77111 ファミリのペリフェラル機能をさらに拡張しています。 μ PD77111 ファミリから機能拡張、追加のあったペリフェラルは次のとおりです。

- 8/16 ビット・ホスト・インタフェース（機能拡張：バスの 16 ビット対応）
- 汎用出力ポート（機能拡張：ポート数を最大 16 本に拡張）
- 8 チャンネル DMA 機能内蔵（新規）
- 時分割（TDM）シリアル・インタフェース（新規）
- 32/64 ビット・オーディオ・シリアル・インタフェース（新規）
- 16 ビット・タイマ（新規）
- SD（Secure Digital）カード・インタフェース[※]

注 μ PD77213 のみ

1.2 特徴

演算部，命令体系は μ PD77111 ファミリに準じています。既存の μ PD77111 ファミリのソフトウェア，ミドルウェアとバイナリ・レベルでコンパチブル性を持っています（ただし，メモリ構成，ペリフェラル構成に依存する部分を除きます）。

1.2.1 DSP コア・カーネル部

機能

- デュアル・ロード/ストアによる並列実行
- ハードウェア・ループ
- 条件実行
- 積和演算を1インストラクション・サイクルで実行
- JTAG 対応機能内蔵

プログラミング

- 16ビット×16ビット+40ビット 40ビットのマルチプライ・アキュムレータ
- 8本×40ビットの汎用レジスタ
- 8本のデータ・メモリ・ポインタ・レジスタ（2つのメモリ空間4個ずつ）
- 12レベルの割り込み（1レベルごとに4要因を割り当て，割り込み要因を拡張）
- 3オペランド命令
- 実行ステージはパイプラインなし

メモリ空間（DSP コア・カーネルから見た空間）^注

- | | |
|------------|------------------|
| • 命令メモリ | 32ビット幅×16ビット・ワード |
| • Xデータ・メモリ | 16ビット幅×16ビット・ワード |
| • Yデータ・メモリ | 16ビット幅×16ビット・ワード |

注 論理的に取り扱える空間であり，すべて実装されているわけではありません。

1.2.2 ペリフェラル部

機能

- 8チャンネルDMA機能

ペリフェラル - メモリ間転送機能（PMT）

- 1Mワード×16ビットのSRAMインタフェース
- 時分割（TDM）シリアル・インタフェース（TSIO）
- オーディオ・シリアル・インタフェース（ASIO）
- 8/16ビット・ホスト・インタフェース（HIO）
- 汎用入出力ポート（PIO）
- 16ビット・タイマ
- 割り込み機能拡張
- 周辺回路スタンバイ機能
- 命令訂正機能

- メモリ空間拡張機能 (ページング機能)
- PLL 逡倍, 分周制御機能
- SD カード・インタフェース (μ PD77213 のみ)

メモリ空間 (ページング機能をとおして見た空間)^注

- 命令メモリ 32 ビット幅 × 16 ビット・ワード × 64 ページ
- X データ・メモリ 16 ビット幅 × 16 ビット・ワード × 64 ページ
- Y データ・メモリ 16 ビット幅 × 16 ビット・ワード × 64 ページ
- DMA データ・メモリ 16 ビット幅 × 14 ビット・ワード
- 外部データ・メモリ 16 ビット幅 × 20 ビット・ワード

注 論理的に取り扱える空間であり, すべて実装されているわけではありません。

表 1-1 μ PD77210 ファミリの特徴

		μ PD77210	μ PD77213
インストラクション・サイクル		6.25 ns	8.33 ns
最大動作周波数		160 MHz	120 MHz
クロック回路	PLL 逡倍回路	$\times 10 \sim 32$ ($\times 2$ ステップ), $\times 40 \sim 64$ ($\times 8$ ステップ)	
	分周器出力	$\div 1 \sim 16$ の整数分周	
並列命令実行		三項演算命令 & 並列ロード / ストア, 二項演算命令 & 並列ロード / ストア, 単項演算命令 & 条件命令, レジスタ間転送命令 & 条件命令, 分岐命令 & 条件命令	
ハードウェア・ループ		最大 4 重までネスティング可能	
条件命令		独立した条件命令をほかの命令と組み合わせて, 条件演算, 条件転送, 条件分岐が可能	
マルチプライ・アキュムレータ		16 ビット \times 16 ビット + 40 ビット \rightarrow 40 ビット	
アキュムレータ		40 ビット入出力 (二項演算, 単項演算)	
汎用レジスタ		8 本の 40 ビット・レジスタ	
データ・メモリ・ポインタ		X メモリ / Y メモリ空間用にそれぞれ 4 本, 計 8 本	
割り込み		ベクタ 12 本 (ベクタ内で要因を最大 4 多重対応可)	
3 ステージ・パイプライン処理		命令フェッチ, 命令デコード, 命令実行	
命令メモリ	システム ROM	512 ワード (ブート用)	
	内部 RAM	31.5 K ワード	15.5 K ワード
	内部 ROM	なし	64 K ワード
	外部領域	なし	
データ・メモリ	X 内部 RAM	30 K ワード	18 K ワード
	X 内部 ROM	なし	32 K ワード
	Y 内部 RAM	30 K ワード	18 K ワード
	Y 内部 ROM	なし	32 K ワード
	外部領域 (X/Y 共通)	1M ワード	1M (SD カード使用時は 8 K) ワード
シリアル・インタフェース		オーディオ・シリアル $\times 1$	
		時分割シリアル $\times 1$	
		スタンダード・シリアル $\times 2$ (オーディオ / 時分割と兼用)	
ホスト・インタフェース		8/16 ビット・パラレル	
汎用入出力ポート		最大 16 本	
タイマ		2 チャンネル (16 ビット分解能)	
メモリ・カード・インタフェース		なし	SD カード (1 ビット)
DMA 転送機能		ペリフェラル - 内部データ RAM 間 8 チャンネル	
電源	DSP コア	1.425 ~ 1.65 V, 1.55 ~ 1.65 V (μ PD77210 の 120 ~ 160 MHz 動作時)	
	外部 I/O	2.7 ~ 3.6 V	
スタンバイ・モード		HALT 命令, STOP 命令	
パッケージ		144 ピン LQFP, 161 ピン FBGA	
その他		デバッグ機能 (JTAG)	

1.2.3 オーダ情報

オーダ名称	パッケージ
μ PD77210F1-DA2	161 ピン・プラスチック・ファインピッチ BGA (10×10)
μ PD77210GJ-8EN	144 ピン・プラスチック LQFP (ファインピッチ)(20×20)
μ PD77213F1-xxx-DA2	161 ピン・プラスチック・ファインピッチ BGA (10×10)
μ PD77213GJ-xxx-8EN	144 ピン・プラスチック LQFP (ファインピッチ)(20×20)

備考 xxx は ROM コード番号です。

第2章 端子機能

この章では、 μ PD77210 ファミリの端子接続図および端子機能について説明します。 μ PD77210 ファミリの端子名称は次のとおりです。

ASCK :	Audio Serial Clock Input/Output	$\overline{\text{MWAIT}}$:	External Data Memory Access Wait Input
ASI :	Audio Serial Data Input	NC :	Non-Connection
ASIEN :	Audio Serial Input Enable	P0-P15 :	Port
ASO :	Audio Serial Data Output	PLL0-PLL3 :	PLL Multiple Rate Set
ASOEN :	Audio Serial Output Enable	Reserved :	Reserved
BCLK :	Bit Clock Input/Output	$\overline{\text{RESET}}$:	Reset
CLKIN :	Clock Input	SDCLK :	SD Card Clock Output
CLKOUT :	Clock Output	SDCR :	SD Card Command Output/Response Input
CSTOP :	Clear Stop Mode	SDDAT0 :	SD Card Data Input
EV _{DD} :	Power Supply for I/O Pins	SDMON :	SD Card Access Monitor Output
GND :	Ground	STOPS :	Stop Status Signal Output
HALTS :	Halt Status Signal Output	TCK :	Test Clock Input
HD0-HD15 :	Host Data Bus	TDI :	Test Data Input
$\overline{\text{HCS}}$:	Host Chip Select	TDO :	Test Data Output
HA0, HA1 :	Host Data Access	TICE :	Test In-Circuit Emulator
$\overline{\text{HRD}}$:	Host Read	TIMOUT :	Timer Time Out Monitor Output
$\overline{\text{HRE}}$:	Host Read Enable	TMS :	Test Mode Select
$\overline{\text{HWE}}$:	Host Write Enable	$\overline{\text{TRST}}$:	Test Reset
$\overline{\text{HWR}}$:	Host Write	TSCK :	Time Division Multiplex Serial Clock Input
I.C. :	Internal Connection	TSI :	Time Division Multiplex Serial Data Input
IV _{DD} :	Power Supply for DSP Core	TSIAK :	Time Division Multiplex Serial Input Acknowledge
INT _{mn} :	Interrupt (m,n : 0-3)	TSIEN :	Time Division Multiplex Serial Input Enable
LRCLK :	Left Right Clock Input/Output	TSO :	Time Division Multiplex Serial Data Output
MA0-MA19 :	External Data Memory Address Bus	TSOEN :	Time Division Multiplex Serial Data Output Enable
$\overline{\text{MBSTB}}$:	External Data Memory Bus Strobe	TSORQ :	Time Division Multiplex Serial Output Request
MCLK :	Master Clock Input		
MD0-MD15 :	External Data Memory Bus		
$\overline{\text{MHOLDAK}}$:	External Data Memory Bus Hold Acknowledge		
$\overline{\text{MHOLDRQ}}$:	External Data Memory Bus Hold Request		
$\overline{\text{MRD}}$:	External Data Memory Read Output		
$\overline{\text{MWR}}$:	External Data Memory Write Output		

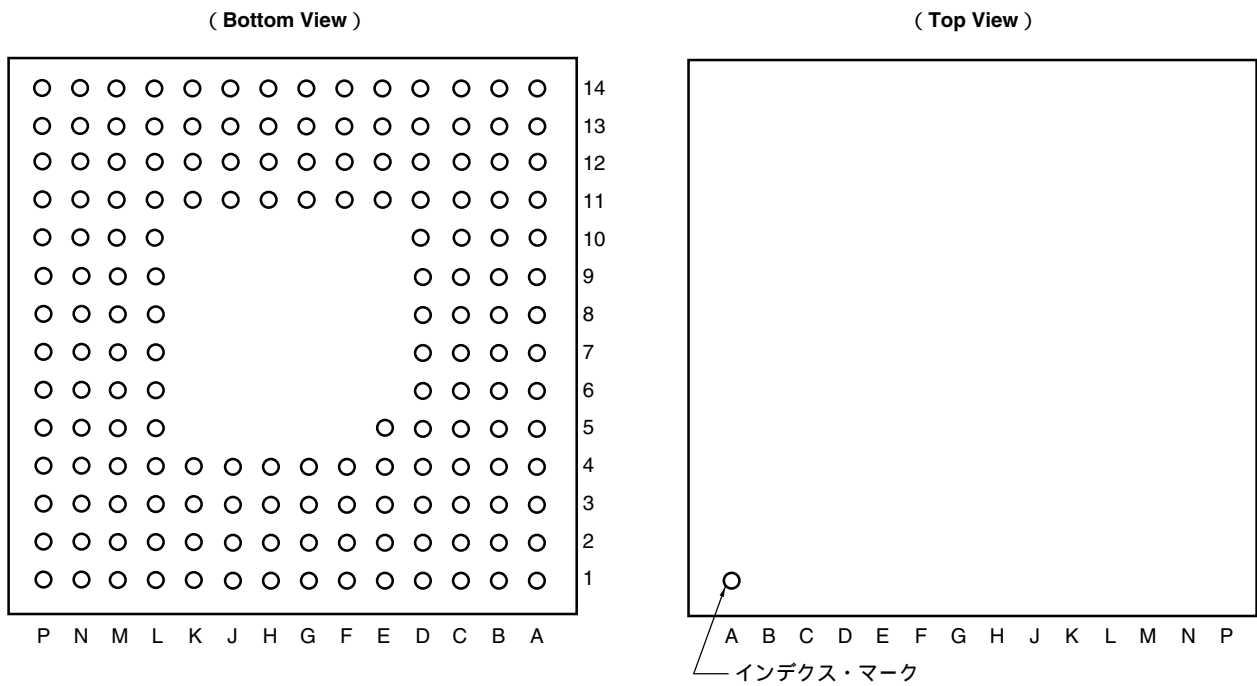
2.1 端子接続

μ PD77213 の端子接続は、一部の外部メモリ端子と SD カード・インタフェースが兼用化されている以外は、 μ PD77210 と同じです。

2.1.1 161 ピン・プラスチック・ファインピッチ BGA (10×10mm)

- μ PD77210F1-DA2
- μ PD77213F1-xxx-DA2

図 2-1 161 ピン・プラスチック・ファインピッチ BGA



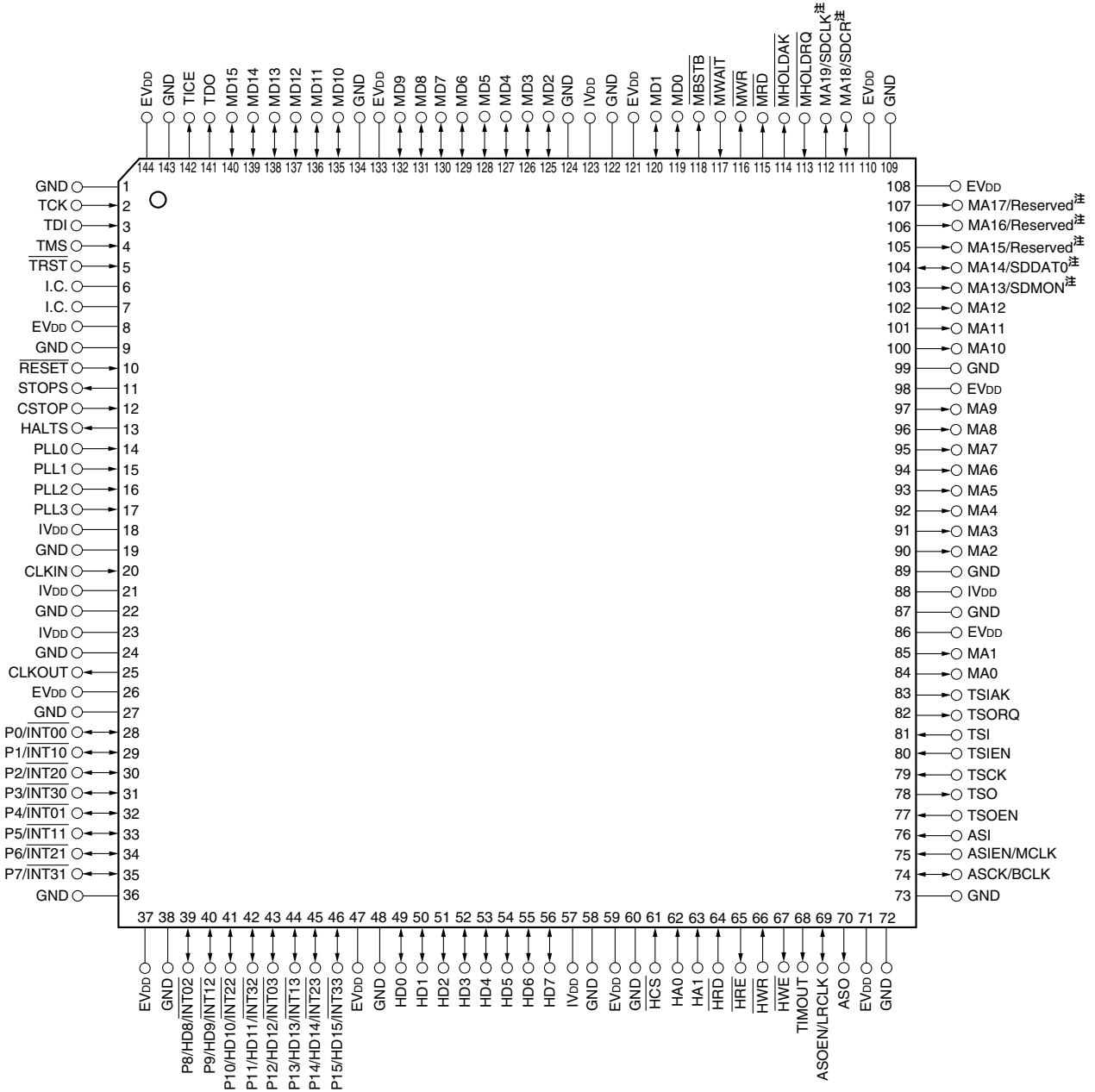
端子番号	端子名	端子番号	端子名	端子番号	端子名	端子番号	端子名
A1	NC	C14	EV _{DD}	H2	HD7	M5	TSORQ
A2	NC	D1	P10/HD10/INT22	H3	HD6	M6	MA0
A3	P5/INT11	D2	P11/HD11/INT32	H4	GND	M7	MA4
A4	P2/INT20	D3	P12/HD12/INT03	H11	MD5	M8	MA5
A5	GND	D4	GND	H12	MD4	M9	MA10
A6	EV _{DD}	D5	GND	H13	MD1	M10	MA12
A7	IV _{DD}	D6	P1/INT10	H14	MD3	M11	MA15/Reserved ^注
A8	IV _{DD}	D7	GND	J1	EV _{DD}	M12	MA19/SDCLK ^注
A9	PLL0	D8	GND	J2	HCS	M13	MA18/SDCR ^注
A10	STOPS	D9	GND	J3	HA1	M14	EV _{DD}
A11	EV _{DD}	D10	GND	J4	HWR	N1	NC
A12	TRST	D11	TMS	J11	GND	N2	NC
A13	NC	D12	TICE	J12	MD0	N3	ASIEN/MCLK
A14	NC	D13	MD12	J13	MBSTB	N4	TSCK
B1	NC	D14	MD15	J14	IV _{DD}	N5	TSIAK
B2	NC	E1	P14/HD14/INT23	K1	HA0	N6	MA1
B3	P7/INT31	E2	P15/HD15/INT33	K2	HRD	N7	MA2
B4	P6/INT21	E3	P13/HD13/INT13	K3	TIMOUT	N8	MA7
B5	P3/INT30	E4	GND	K4	ASO	N9	MA9
B6	CLKOUT	E5	NC	K11	GND	N10	MA11
B7	IV _{DD}	E11	GND	K12	MWR	N11	MA16/Reserved ^注
B8	PLL3	E12	MD14	K13	MWAIT	N12	MA17Reserved ^注
B9	PLL1	E13	MD9	K14	EV _{DD}	N13	NC
B10	CSTOP	E14	MD11	L1	HWE	N14	NC
B11	I.C.	F1	EV _{DD}	L2	HRE	P1	NC
B12	TCK	F2	HD1	L3	GND	P2	NC
B13	NC	F3	HD2	L4	GND	P3	ASI
B14	NC	F4	HD0	L5	TSIEN	P4	TSO
C1	EV _{DD}	F11	MD10	L6	GND	P5	TSI
C2	P8/HD8/INT02	F12	MD13	L7	GND	P6	EV _{DD}
C3	P9/HD9/INT12	F13	MD7	L8	MA8	P7	IV _{DD}
C4	P4/INT01	F14	EV _{DD}	L9	GND	P8	MA3
C5	P0/INT00	G1	HD3	L10	MA14/SDDAT0 ^注	P9	MA6
C6	CLKIN	G2	HD5	L11	GND	P10	EV _{DD}
C7	PLL2	G3	HD4	L12	MHOLDRQ	P11	MA13/SDMON ^注
C8	HALTS	G4	GND	L13	MRD	P12	EV _{DD}
C9	RESET	G11	GND	L14	MHOLDAK	P13	NC
C10	I.C.	G12	MD8	M1	EV _{DD}	P14	NC
C11	TDI	G13	MD2	M2	ASCK/BCLK		
C12	TDO	G14	MD6	M3	ASOEN/LRCLK		
C13	GND	H1	IV _{DD}	M4	TSOEN		

注 μPD77213 の場合は、MA13-MA19 は兼用端子になります。

2.1.2 144ピン・プラスチックLQFP(ファインピッチ)(20×20mm)

- μPD77210GJ-8EN
- μPD77213GJ-xxx-8EN

図 2-2 144ピン・プラスチックLQFP



注 μPD77213の場合は、MA13-MA19は兼用端子になります。

端子番号	端子名	端子番号	端子名	端子番号	端子名	端子番号	端子名
1	GND	37	EV _{DD}	73	GND	109	GND
2	TCK	38	GND	74	ASCK/BCLK	110	EV _{DD}
3	TDI	39	P8/HD8/INT02	75	ASIEN/MCLK	111	MA18/SDCR ^注
4	TMS	40	P9/HD9/INT12	76	ASI	112	MA19/SDCLK ^注
5	TRST	41	P10/HD10/INT22	77	TSOEN	113	MHOLD _{RQ}
6	I.C.	42	P11/HD11/INT32	78	TSO	114	MHOLD _{AK}
7	I.C.	43	P12/HD12/INT03	79	TSCK	115	MRD
8	EV _{DD}	44	P13/HD13/INT13	80	TSIEN	116	MWR
9	GND	45	P14/HD14/INT23	81	TSI	117	MWAIT
10	RESET	46	P15/HD15/INT33	82	TSORQ	118	MBSTB
11	STOPS	47	EV _{DD}	83	TSIAK	119	MD0
12	CSTOP	48	GND	84	MA0	120	MD1
13	HALTS	49	HD0	85	MA1	121	EV _{DD}
14	PLL0	50	HD1	86	EV _{DD}	122	GND
15	PLL1	51	HD2	87	GND	123	IV _{DD}
16	PLL2	52	HD3	88	IV _{DD}	124	GND
17	PLL3	53	HD4	89	GND	125	MD2
18	IV _{DD}	54	HD5	90	MA2	126	MD3
19	GND	55	HD6	91	MA3	127	MD4
20	CLKIN	56	HD7	92	MA4	128	MD5
21	IV _{DD}	57	IV _{DD}	93	MA5	129	MD6
22	GND	58	GND	94	MA6	130	MD7
23	IV _{DD}	59	EV _{DD}	95	MA7	131	MD8
24	GND	60	GND	96	MA8	132	MD9
25	CLKOUT	61	HCS	97	MA9	133	EV _{DD}
26	EV _{DD}	62	HA0	98	EV _{DD}	134	GND
27	GND	63	HA1	99	GND	135	MD10
28	P0/INT00	64	HRD	100	MA10	136	MD11
29	P1/INT10	65	HRE	101	MA11	137	MD12
30	P2/INT20	66	HWR	102	MA12	138	MD13
31	P3/INT30	67	HWE	103	MA13/SDMON ^注	139	MD14
32	P4/INT01	68	TIMOUT	104	MA14/SDDAT0 ^注	140	MD15
33	P5/INT11	69	ASOEN/LRCLK	105	MA15/Reserved ^注	141	TDO
34	P6/INT21	70	ASO	106	MA16/Reserved ^注	142	TICE
35	P7/INT31	71	EV _{DD}	107	MA17/Reserved ^注	143	GND
36	GND	72	GND	108	EV _{DD}	144	EV _{DD}

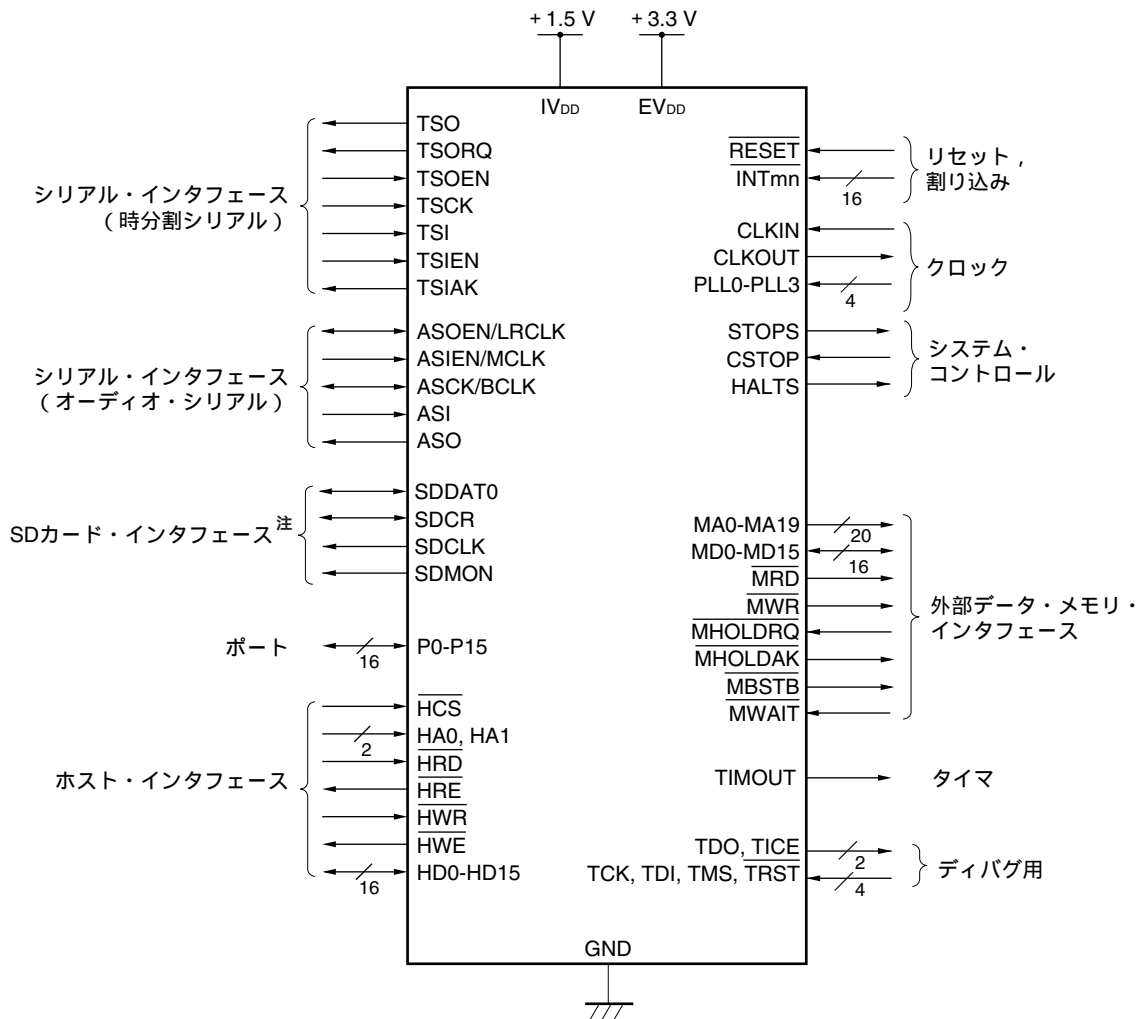
注 μPD77213 の場合は，MA13-MA19 は兼用端子になります。

2.2 端子構成

μ PD77210 ファミリの端子接続を、機能別に群構成して示します。

μ PD77213 の端子構成は、SD カード・インタフェースが追加されている以外は、μ PD77210 の端子構成と同じです。

図 2-3 端子構成図



注 μ PD77213 のみ

注意 ポート端子、ホスト・インタフェース端子、シリアル・インタフェース端子、割り込み端子、SD カード・インタフェース端子は兼用になっているものがあります。

備考 m,n=0-3

2.3 端子機能

(1) 電源

端子名称	端子番号		入出力	端子機能	兼用端子
	144ピン LQFP	161ピン FBGA			
IVDD	18,21,23,57,88, 123	A7,A8,B7,H1,J14, P7	-	DSP コア用電源 (+1.5V) μPD77210 ファミリのコア部分に供給する電源です。	-
EVDD	8,26,37,47,59,71, 86,98,108,110, 121,133,144	A6,A11,C1,C14, F1,F14,J1,K14, M1,M14,P6,P10, P12	-	I/O 端子用電源 (+3.3V) 外部インタフェース端子用の電源です。	-
GND	1,9,19,22,24,27, 36,38,48,58,60, 72,73,87,89,99, 109,122,124,134, 143	A5,C13,D4,D5, D7,D8,D9,D10, E4,E11,G4,G11, H4,J11,K11,L3, L4,L6,L7,L9,L11	-	接地 グランド端子です。	-

備考 IVDD と EVDD の両電源は、同時に投入するようにしてください。

(2) クロック、システム・コントロール

端子名称	端子番号		入出力	機能	兼用端子
	144ピン LQFP	161ピン FBGA			
CLKIN	20	C6	入力	クロック入力 μPD77210 ファミリを動作させるためのクロック 入力です。	-
CLKOUT	25	B6	出力	内部システム・クロック出力 CLKIN から入力されたクロックを PLL 回路で 逡倍した内部システム・クロックを出力します。	-
PLL0-PLL3	14-17	A9,B9,C7,B8	入力	PLL 逡倍率設定入力 PLL 回路のクロック逡倍率を設定します。 PLL3:PLL2:PLL1:PLL0 0000 : ×10 0001 : ×12 0010 : ×14 0011 : ×16 0100 : ×18 0101 : ×20 0110 : ×22 0111 : ×24 1000 : ×26 1001 : ×28 1010 : ×30 1011 : ×32 1100 : ×40 1101 : ×48 1110 : ×56 1111 : ×64	-
HALTS	13	C8	出力	ホールド・モード・ステータス出力 ホールド・モードおよびストップ・モード中にア クティブになります。	-
STOPS	11	A10	出力	ストップ・モード・ステータス出力 ストップ・モード中にアクティブになります。	-
CSTOP	12	B10	入力	ストップ・モード解除信号入力 アクティブにすることによってストップ・モード からの復帰を行います。	-

(3) リセット, 割り込み

端子名称	端子番号		入出力	機能	兼用端子
	144ピン LQFP	161ピン FBGA			
$\overline{\text{RESET}}$	10	C9	入力	内部システム・リセット信号入力 μ PD77210 ファミリを初期化します。	-
$\overline{\text{INT00}}$	28	C6	入力	マスカブル外部割り込み入力 外部割り込み入力端子です。	P0
$\overline{\text{INT01}}$	32	C4	入力		P4
$\overline{\text{INT02}}$	39	C2	入力		P8/HD8
$\overline{\text{INT03}}$	43	D3	入力		P12/HD12
$\overline{\text{INT10}}$	29	D6	入力		P1
$\overline{\text{INT11}}$	33	A3	入力		P5
$\overline{\text{INT12}}$	40	C3	入力		P9/HD9
$\overline{\text{INT13}}$	44	E3	入力		P13/HD13
$\overline{\text{INT20}}$	30	A4	入力		P2
$\overline{\text{INT21}}$	34	B4	入力		P6
$\overline{\text{INT22}}$	41	D1	入力		P10/HD10
$\overline{\text{INT23}}$	45	E1	入力		P14/HD14
$\overline{\text{INT30}}$	31	B5	入力		P3
$\overline{\text{INT31}}$	35	B3	入力		P7
$\overline{\text{INT32}}$	42	D2	入力		P11/HD11
$\overline{\text{INT33}}$	46	E2	入力		P15/HD15

(4) 外部データ・メモリ・インタフェース

端子名称	端子番号		入出力	機能	兼用端子
	144ピン LQFP	161ピン FBGA			
MA0-MA19 ^注	84-85,90-97, 100-107,111-112	M6,N6,N7,P8, M7,M8,P9,N8,L8, N9,M9,N10,M10, P11,L10,M11, N11,N12,M13, M12	出力 (3S)	外部データ・メモリのアドレス・バス 外部データ・メモリをアクセスするときにアドレスを出力します。	SDCLK, SDCR, SDDAT0, SDMON
MD0-MD15	119-120, 125-132, 135-140	J12,H13,G13, H14,H12,H11, G14,F13,G12, E13,F11,E14, D13,F12,E12, D14	入出力 (3S)	16ビット・データ・バス 外部データ・メモリをアクセスするときにデータを入出力します。	-
MWR	116	K12	出力 (3S)	ライト出力 外部データ・メモリのライト・ストロープです。	-
MRD	115	L13	出力 (3S)	リード出力 外部データ・メモリのリード・ストロープです。	-
MHOLDAK	114	L14	出力	ホールド・アクノリッジ信号 外部デバイスに、 μ PD77210ファミリの外部データ・メモリ・バスの使用を許可するとき、ロウ・レベルになります。	-
MHOLDRQ	113	L12	入力	ホールド・リクエスト信号 外部デバイスが、 μ PD77210ファミリの外部データ・メモリ・バスを使用したいときにロウ・レベルを入力します。	-
MWAIT	117	K13	入力	ウエイト信号入力 μ PD77210ファミリが外部データ・メモリのアクセスをするときに、ウエイト・サイクルを挿入する。 ・0：ウエイト挿入 ・1：ウエイト非挿入	-
MBSTB	118	J13	出力	バス・ストロープ信号 μ PD77210ファミリが外部データ・メモリ・バスを使用しているとき、ロウ・レベルになります。	-

注 MA13-MA19が兼用端子になるのは、 μ PD77213の場合のみです。

備考 表中入出力欄に“3S”を付記した端子は、次の状態でハイ・インピーダンスになります。

MA0-MA19, $\overline{\text{MRD}}$, $\overline{\text{MWR}}$: バス解放時 ($\overline{\text{MHOLDAK}}$ = ロウ・レベル)

MD0-MD15 : 外部データ・メモリ非アクセス時、およびバス解放時 ($\overline{\text{MHOLDAK}}$ = ロウ・レベル)

(5) タイマ

端子名称	端子番号		入出力	機能	兼用端子
	144ピン LQFP	161ピン FBGA			
TIMOUT	68	K3	出力	タイム・アウト・モニタ タイマがタイム・アウトしたときにアクティブになります。	-

(6) シリアル・インタフェース

端子名称	端子番号		入出力	機能	兼用端子
	144ピン LQFP	161ピン FBGA			
ASCK/BCLK	74	M2	入出力	オーディオ・シリアル用クロック入力 ASCK : オーディオ・シリアル・クロック入力 BCLK : シリアル・クロック入出力	-
ASO	70	K4	出力 (3S)	オーディオ・シリアル・データ出力	-
ASI	76	P3	入力	オーディオ・シリアル・データ入力	-
ASOEN/ LRCLK	69	M3	入出力	オーディオ・シリアル出力イネーブル/レフト・ ライト・クロック入力 ASOEN : オーディオ・シリアル出力イネーブル 入力 LRCLK : レフト・ライト・クロック入出力	-
ASIEN/MCLK	75	N3	入力	オーディオ・シリアル入力イネーブル/マスタ・ クロック入力 ASIEN : オーディオ・シリアル入力イネーブル入 力 MCLK : マスタ・クロック入力 (マスタ・モード 時)	-
TSCK	79	N4	入力	時分割シリアル用クロック入力	-
TSO	78	P4	出力 (3S)	時分割シリアル・データ出力	-
TSI	81	P5	入力	時分割シリアル・データ入力	-
TSORQ	82	M5	出力	時分割シリアル出力リクエスト	-
TSOEN	77	M4	入力	時分割シリアル出力イネーブル	-
TSIEN	80	L5	入力	時分割シリアル入力イネーブル	-
TSIAK	83	N5	出力	時分割シリアル入力応答	-

備考 表中入出力欄に“3S”を付記した端子は、データ送出完了時、およびハードウェア・リセット ($\overline{\text{RESET}}$) 入力により、ハイ・インピーダンス状態になります。

(7) ホスト・インタフェース

端子名称	端子番号		入出力	機 能	兼用端子
	144 ピン LQFP	161 ピン FBGA			
HA1	63	J3	入力	ホスト・アドレス 1 ホスト・インタフェース端子 (HD7-HD0 または HD15-HD0) がアクセスするレジスタを指定します。 ・1: ホスト・インタフェース・ステータス・レジスタ (HST) をアクセスします。 ・0: 読み出し ($\overline{\text{HRD}} = 0$) のとき, ホスト送信データ・レジスタ (HDT(out)) を, 書き込み ($\overline{\text{HWR}} = 0$) のとき, ホスト受信データ・レジスタ (HDT(in)) をアクセスします。	-
HA0	62	K1	入力	ホスト・アドレス 0 8 ビット・モードのとき, HD7-HD0 がアクセスするレジスタを指定します。16 ビット・モード時は無効になります。 ・1: HST, HDT(in), HDT(out) のビット 15-8 をアクセスする。 ・0: HST, HDT(in), HDT(out) のビット 7-0 をアクセスする。	-
$\overline{\text{HCS}}$	61	J2	入力	チップ・セレクト入力	-
$\overline{\text{HRD}}$	64	K2	入力	ホスト・リード入力	-
$\overline{\text{HWR}}$	66	J4	入力	ホスト・ライト入力	-
$\overline{\text{HRE}}$	65	L2	出力	ホスト・リード・イネーブル出力	-
$\overline{\text{HWE}}$	67	L1	出力	ホスト・ライト・イネーブル出力	-
HD0-HD7	49-56	F4,F2,F3,G1,G3,G2,H3,H2	入出力 (3S)	8 ビット・ホスト・データ・バス 8 ビット・ホスト・モードのときのホスト・データ・バスです。16 ビット・データの入出力は HA0 端子によって 2 回に分けてアクセスします。 16 ビット・モードのときは下位 8 ビットの入出力を行います。	-
HD8-HD15	39-46	C2,C3,D1,D2,D3,E3,E1,E2	入出力 (3S)	ホスト・データ・バス 16 ビット・ホスト・モードのときのホスト・データ・バスです。HD0-HD7 と合わせて 16 ビット・データを入出力します。	P8-P15/ $\overline{\text{INT02}}$, $\overline{\text{INT12}}$, $\overline{\text{INT22}}$, $\overline{\text{INT32}}$, $\overline{\text{INT03}}$, $\overline{\text{INT13}}$, $\overline{\text{INT23}}$, $\overline{\text{INT33}}$

備考 表中入出力欄に “3S” を付記した端子は, ホスト I/F 非アクセス時に, ハイ・インピーダンス状態になります。

(8) 入出力ポート

端子名称	端子番号		入出力	機 能	兼用端子
	144 ピン LQFP	161 ピン FBGA			
P0	28	C5	入出力	汎用入出力ポート	INT00
P1	29	D6	入出力		INT10
P2	30	A4	入出力		INT20
P3	31	B5	入出力		INT30
P4	32	C4	入出力		INT01
P5	33	A3	入出力		INT11
P6	34	B4	入出力		INT21
P7	35	B3	入出力		INT31
P8	39	C2	入出力		INT02/HD8
P9	40	C3	入出力		INT12/HD9
P10	41	D1	入出力		INT22/HD10
P11	42	D2	入出力		INT32/HD11
P12	43	D3	入出力		INT03/HD12
P13	44	E3	入出力		INT13/HD13
P14	45	E1	入出力		INT23/HD14
P15	46	E2	入出力	INT33/HD15	

(9) ディバグ用インタフェース

端子名称	端子番号		入出力	機 能	兼用端子
	144 ピン LQFP	161 ピン FBGA			
TDO	141	C12	出力 (3S)	ディバグ用 ディバグを用いるときのインタフェース端子です。	-
TICE	142	D12	出力		-
TCK	2	B12	入力		-
TDI	3	C11	入力		-
TMS	4	D11	入力		-
TRST	5	A12	入力		-

備考 表中入出力欄に“3S”を付記した端子は、ディバグ I/F 非アクセス時に、ハイ・インピーダンス状態になります。

(10) その他

端子名称	端子番号		入出力	機 能	兼用端子
	144 ピン LQFP	161 ピン FBGA			
I.C.	6,7	B11,C10	-	内部接続端子 オープンにしてください。	-
NC	-	A1,A2,A13,A14, B1,B2,B13,B14, E5,N1,N2,N13, N14,P1,P2,P13, P14	-	ノー・コネクション オープンにしてください。	-

注意 これらの端子になんらかの信号の印加または読み出しを行ったとき、 μ PD77210 ファミリの正常な動作は保証されません。

(11) SD カード・インタフェース：μPD77213 のみ

端子名称	端子番号		入出力	機能	兼用端子
	144 ピン LQFP	161 ピン FBGA			
SDCLK	112	M12	出力	SD カード・クロック出力 ・オープンにしてください。	MA19
SDCR	111	M13	入出力 (3S)	SD カード・コマンド・レスポンス 入力：レスポンス 出力：コマンド ・プルアップしてください。	MA18
SDDAT0	104	L10	入出力 (3S)	SD カード・データ入出力 入力：リード・データ 出力：ライト・データ ・プルアップしてください。	MA14
SDMON	103	P11	出力	SD カード・インタフェース・アクセス・モニタ SD カード・インタフェースをアクセスしているときにハイ・レベルを出力します。 1：SD カード・インタフェースをアクセス中 0：SD カード・インタフェースを非アクセス中	MA13
Reserved	105-107	M11,N11,N12	-	将来の拡張用端子です。 SD カード・インタフェースを使用する場合は、ハイ・インピーダンスになります。	MA15-MA17

備考 表中入出力欄に“3S”を付記した端子は、SD カード I/F 非アクセス時に、ハイ・インピーダンス状態になります。

2.4 未使用端子の処理について

実装時に未使用になる端子は、次の表のとおりに取り扱うことを推奨します。

表 2-1 機能端子の処理

端 子	入出力	推奨接続方法
STOPS, HALTS	出力	オープンにしてください。
CSTOP	入力	プルダウン抵抗を介して GND に接続してください。
CLKOUT	出力	オープンにしてください。
P0-P15	入出力	プルアップ抵抗を介して EV _{DD} に接続、またはプルダウン抵抗を介して GND に接続してください。
HD0-HD7 ^{注1}	入出力	プルアップ抵抗を介して EV _{DD} に接続、またはプルダウン抵抗を介して GND に接続してください。
HA0, HA1	入力	プルアップ抵抗を介して EV _{DD} に接続、またはプルダウン抵抗を介して GND に接続してください。
HCS, HRD, HWR	入力	プルアップ抵抗を介して EV _{DD} に接続してください。
HRE, HWE	出力	オープンにしてください。
TIMOUT	出力	オープンにしてください。
ASCK, TSCK	入力	プルアップ抵抗を介して EV _{DD} に接続、またはプルダウン抵抗を介して GND に接続してください。
ASI, TSI	入力	プルダウン抵抗を介して GND に接続してください。
ASIEN, TSIEN	入力	
ASOEN, TSOEN, LRCLK	入力	
ASO, TSO	出力	オープンにしてください。
SORQ	出力	
TSIAK	出力	
MA0-MA19	出力	オープンにしてください。
MD0-MD15 ^{注2}	入出力	プルアップ抵抗を介して EV _{DD} に接続、またはプルダウン抵抗を介して GND に接続してください。
MRD, MWR	出力	オープンにしてください。
MHOLDRQ	入力	プルアップ抵抗を介して EV _{DD} に接続してください、
MBSTB, MHOLDAK	出力	オープンにしてください。
MWAIT	入力	プルアップ抵抗を介して EV _{DD} に接続してください。
TCK	入力	プルダウン抵抗を介して GND に接続してください。
TDO, TICE	出力	オープンにしてください。
TMS, TDI	入力	オープンにしてください(内部でプルアップされています)。
TRST	入力	オープンにしてください(内部でプルダウンされています)。

注1. $\overline{\text{HCS}}$ 、 $\overline{\text{HRD}}$ 、 $\overline{\text{HWR}}$ がハイ・レベル固定なら、オープン可能です。ただし、ホールド・モード、ストップ・モード時など、消費電流が問題となる場合は、推奨接続方法に従ってください。

2. プログラム中で外部データ・メモリをアクセスしない場合はオープン可能です。ただし、ホールド・モード、ストップ・モード時など、消費電流が問題となる場合は、推奨接続方法に従ってください。

注意 兼用端子でこの表に記載されていないものについては、兼用となっている端子処置に従ってください。

その際、ソフトウェア上のモード切り替えによって兼用端子の機能を切り替える端子については、デフォルト設定にしてください。

表 2-2 非機能端子の処理

端子	入出力	推奨接続方法
I.C.	-	オープンにしてください。
NC	-	オープンにしてください。

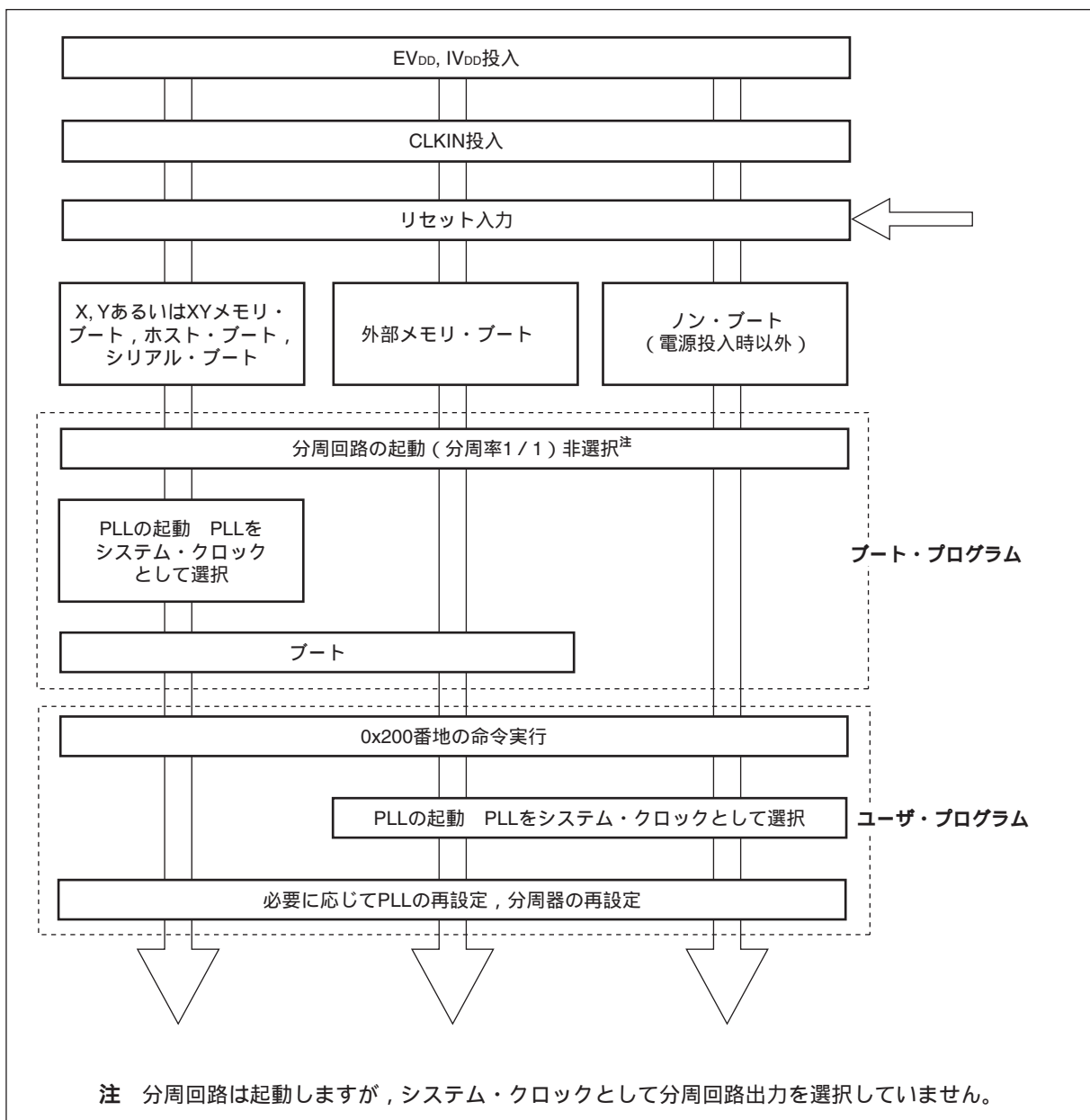
第3章 使用方法概要

この章では、電源投入から実際に μ PD77210ファミリのユーザ・プログラム・コードが実行されるまでの流れ、および全体の使い方について簡単に説明します。詳細については、各機能の説明を参照してください。

3.1 電源投入からプログラム実行まで

電源投入からユーザ・プログラム実行までの流れを次に示します。

図 3-1 電源投入からユーザ・プログラムの実行まで



3.1.1 電源投入

μ PD77210 ファミリの電源は、DSP コア用電源 (IV_{DD}) と I/O 用電源 (EV_{DD}) との 2 電源となっています。 IV_{DD} は 1.5V 系、 EV_{DD} は 3V 系です。これらの電源は同時に投入することをお勧めします。

電源投入後、クロックを CLKIN 端子に、PLL の逡倍率設定値を PLL0-PLL3 端子に、ブート・モードの選択設定値を P0-P2 端子に、PLL のロック・レンジ設定値を P3 端子に加えます。

3.1.2 リセット

規定の電源電圧になったあと、リセット端子に 6 サイクルのロウ・レベルを入力すると、デバイスが初期化されます。CLKIN 端子への入力クロックで 6 サイクルということに注意してください。リセットをしない場合は内部レジスタの状態が初期化されないほか、端子状態も確定しないため外部でバス共有を行う場合などには注意が必要です。また、この段階では PLL は動作していません。

3.1.3 ブート

μ PD77210 ファミリはリセット後、0x200 番地からのユーザ・プログラムを実行します。しかし、 μ PD77210 ファミリでは 0x200 番地が RAM になっているので、電源投入後の値は不定となり、命令を実行できません。このため、ブートアップによって 0x200 番地以降の RAM に命令を書き込む必要があります。

ブートアップはリセット解除後にブートアップ・ルーチン（内蔵の ROM に格納）を介して行われます。ブートアップには命令コードの転送元の種類によっていくつかのモードがあり、ブートアップ・ルーチン実行中の P0-P2 端子の値により決定されます。

リセット後のブート（イニシャル・リセット・ブート）によってブートアップされるのは 0x200 ~ 0x7FFF の命令領域のみです。これ以外の命令領域をブートする場合はプログラム中よりリブート・ルーチンを実行する必要があります。また、データ領域のブートアップはユーザ・プログラムで行う必要があります（制限付きでリブート・ルーチンの利用も可能です）。

ブートに先立って、ブート ROM は PLL を起動してクロック・ソースを PLL 出力に自動的に切り替えます^注。起動から切り替えまでにロックアップ・タイムを必要とするため、入力クロックで約 5,000 サイクルかかります。ユーザはロックアップそのものをケアする必要はありませんが、ブート時にはロックアップ・タイムが加わることになります。切り替え後 PLL クロックで μ PD77210 ファミリは動作をし、ブートを実行します。

注 外部メモリ・ブートを選択している場合、自動的に起動しません。ブートは外部から入力されるクロックで処理されます。3.3 **ブート後のクロック設定**を参照してください。

3.1.4 プログラム実行

ブートアップ・ルーチン終了後、 μ PD77210 ファミリは 0x200 番地からの命令を実行します。

3.2 スタンバイ・モード

DSP プログラムのアイドル時、消費電流を低減する方法として2種類のスタンバイ・モードを備えています。スタンバイ・モードへは命令実行によって移行します。

HALT モードへは、HALT 命令の実行によって移行します。自動的にクロック・ソースを分周器側に切り替えます。したがって、あらかじめ分周器側にクロック・ソースを切り替えている場合は、動作クロックに変化はありません。各種割り込み（ただしマスクされていないこと）によって通常動作モードへ復帰します。

STOP モードへは、STOP 命令の実行によって移行します。システム・クロックをマスクし、DSP の動作を停止します。PLL を自動的に停止する方法はありません。PLL を停止する場合は、STOP 命令の実行の前にクロック・ソースを入力クロックに切り替えてから、PLL を停止します。CSTOP 端子に信号を引加することによって通常動作モードへ復帰します。

3.3 ブート後のクロック設定

次の場合、ブート実行時に自動的に PLL が起動しないためブート後にユーザ・プログラムから PLL を起動する必要があります。

- ノン・ブート
- 外部メモリ・ブート

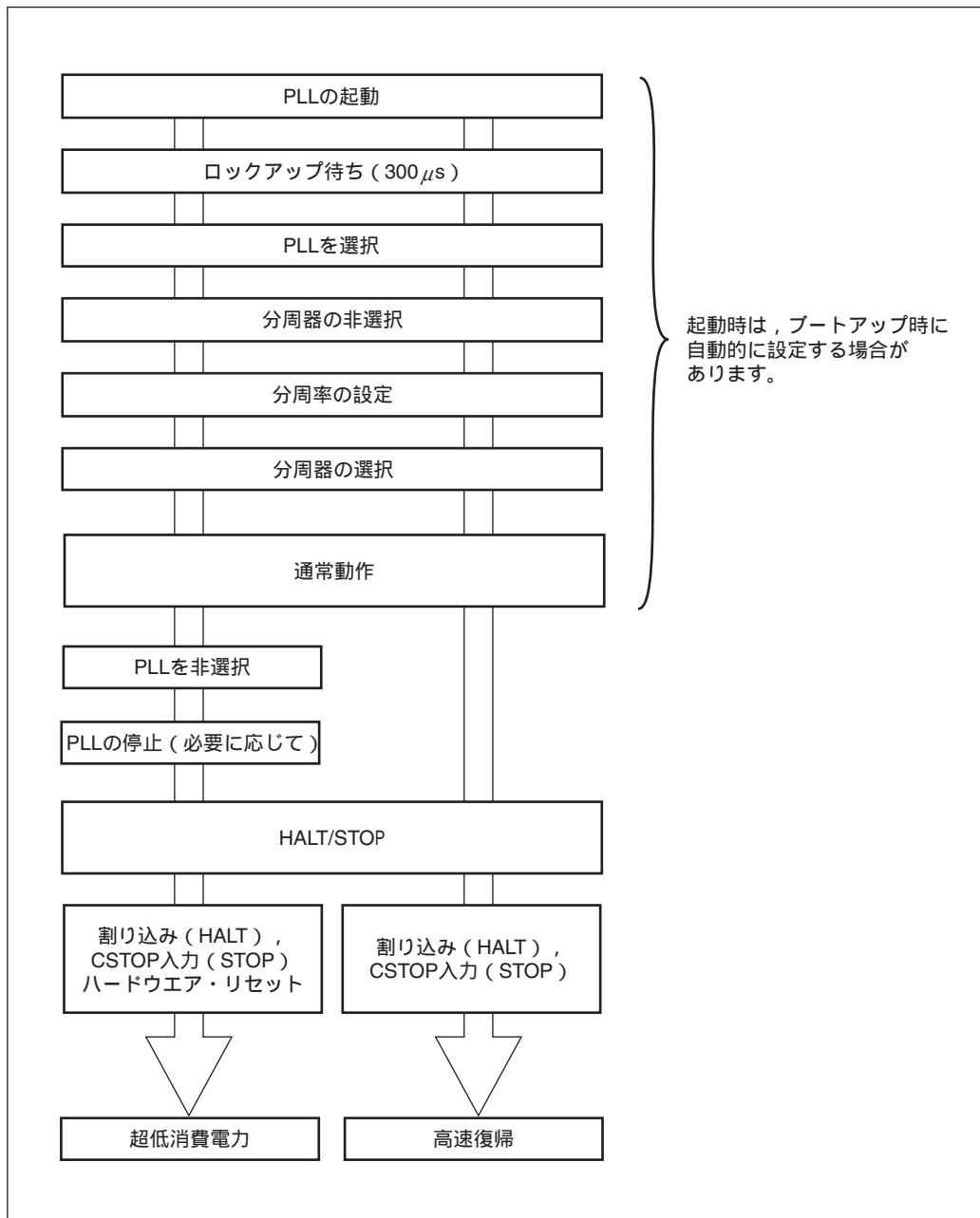
これらの場合は、電源投入後ブート時に PLL がブート ROM によって自動的に起動されず、CLKIN から入力されるクロックを直接システム・クロックとして動作しています。PLL 遜倍クロックをシステム・クロックとして利用する場合は、PLL 起動、PLL のロックを待つ、クロック・ソースを PLL 出力に切り替える、という処理をプログラム中で行う必要があります。

PLL の起動、クロック・ソースの切り替えは、クロック・コントロール用のレジスタ (CLKC) によって行います。PLL 起動後、PLL のロック時間 (300 μ s) を nop 命令などでタイミングを計ってからクロック・ソースを PLL 出力に切り替えます。

このほか CLKC では分周器の設定および選択が可能です。システム・クロックとして CLKIN 端子からの入力クロックあるいは PLL 出力クロックを、分周したクロックも選択可能です。

PLL の起動から停止までの流れを次に示します。

図 3-2 PLL の起動と停止



第4章 アーキテクチャ

μ PD77210 ファミリのアーキテクチャについて説明します。

4.1 全体ブロック構成

ここでは、 μ PD77210 ファミリの物理的な構造を機能ブロックとして区分したものを示します。

バス（メイン・バス，X データ・バス，Y データ・バス）

4.2 バスを参照してください。

システム制御ユニット

4.3 システム制御ユニットを参照してください。

プログラム制御ユニット

4.4 プログラム制御ユニットを参照してください。

データ・アドレッシング・ユニット

4.5 データ・アドレッシング・ユニットを参照してください。

演算ユニット

4.6 演算ユニットを参照してください。

ペリフェラル・ユニット

第5章 ペリフェラルを参照してください。

全体ブロック構成を図 4-1に示します。各機能ブロックについてはそれぞれの節を参照してください。

図 4-1 全体ブロック構成

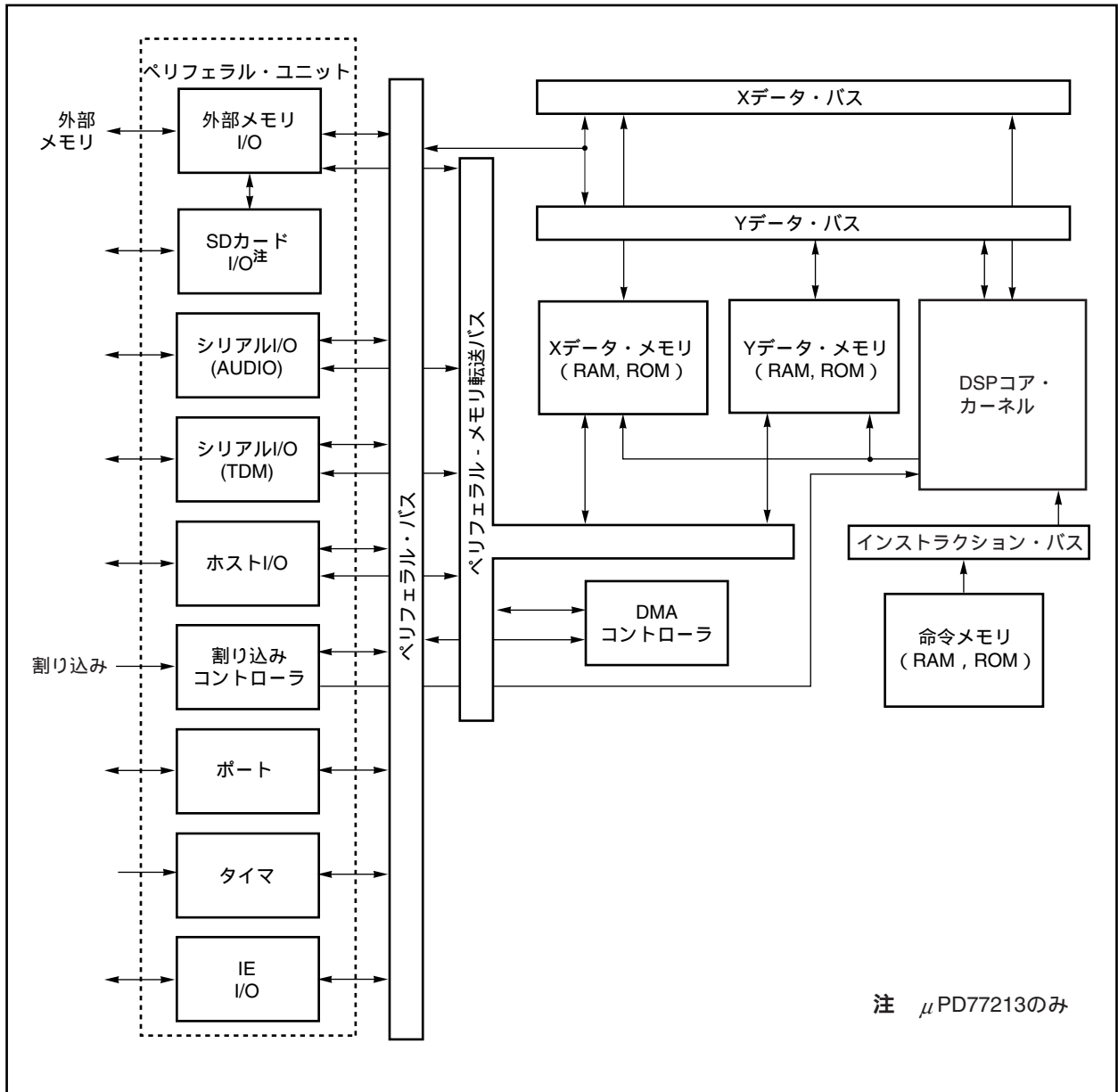
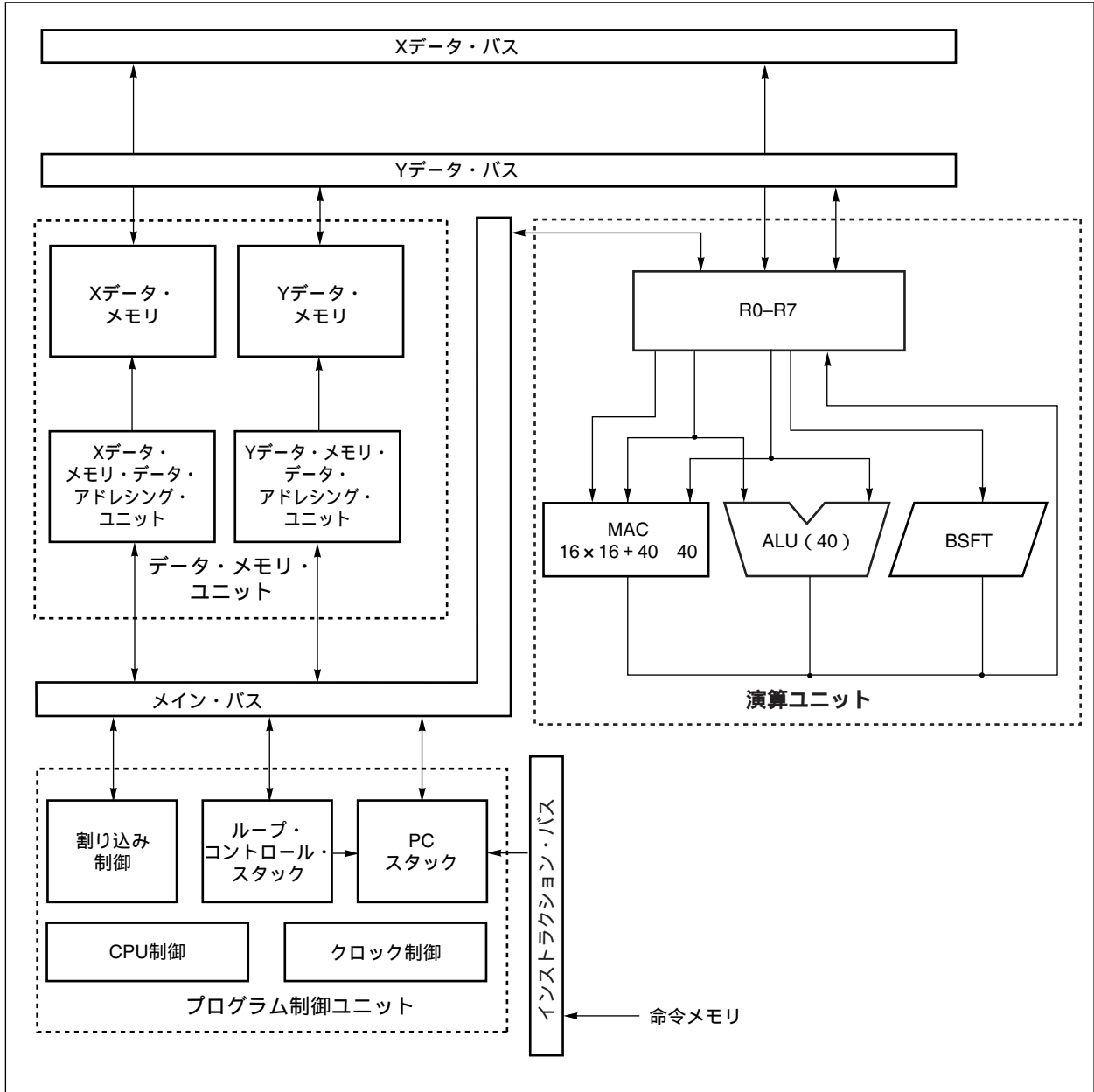


図 4-2 DSP コア・カーネル



4.2 バス

バスはデバイスの内部および外部に対してデータ転送を行います。 μ PD77210 ファミリには次の4つのバスが用意されています。

- メイン・バス
- Xデータ・バス
- Yデータ・バス
- ペリフェラル・メモリ間転送バス

4.2.1 メイン・バス

(1) 機能

汎用レジスタ (R0-R7) と制御レジスタなどを接続する 16 ビットのバスで、次のカテゴリの命令実行によるデータが転送されます。

レジスタ間転送命令

汎用レジスタとその他のレジスタとの転送命令です。そのほかのレジスタとは、表 4-1 に示すもので、汎用レジスタ以外のものを示します。また、この転送で対象としている汎用レジスタの部分は L パートのみです。

レジスタ間転送命令については、 μ PD77016 ファミリ ユーザーズ・マニュアル 命令編を参照してください。

注意 汎用レジスタは 40 ビットで構成されます。この 40 ビットのレジスタは、下位 16 ビット (L パート)、中間の 16 ビット (H パート)、上位 8 ビット (E パート) に分割されます。詳しくは、4.6.2 汎用レジスタとデータ・フォーマットを参照してください。

即値設定命令

イミディエイト・データを指定されたレジスタに設定する命令です。指定されたレジスタとは表 4-1 に示したレジスタのうち次のものを示します。

- 汎用レジスタ (ただし L パート (R0L-R7L) のみ)
- データ・ポインタ (DP0-DP7)
- インデクス・レジスタ (DN0-DN7)
- モジュロ・レジスタ (DMX, DMY)

即値設定命令については、 μ PD77016 ファミリ ユーザーズ・マニュアル 命令編を参照してください。

(2) メイン・バス接続レジスタ

表 4-1に、メイン・バスに接続されているレジスタを示します。

表 4-1 メイン・バス接続レジスタ

レジスタ名	アセンブリの予約名称	ロード(L) / ストア(S)
汎用レジスタ	R0L-R7L (R0-R7 の L パート)	L/S
データ・ポインタ	DP0-DP7	
インデクス・レジスタ	DN0-DN7	
モジュロ・レジスタ	DMX, DMY	
スタック	STK	
スタック・ポインタ	SP	
ループ・カウンタ	LC	
ループ・スタック (LSTK)	LSR1, LSR2, LSR3	
ループ・スタック・ポインタ	LSP	
ステータス・レジスタ	SR	
割り込み許可フラグ・スタック・レジスタ	EIR	
エラー・ステータス・レジスタ	ESR	

4.2.2 データ・バス

(1) 機能

汎用レジスタと X, Y データ・メモリ, および内蔵ペリフェラルを接続する 16 ビットのバスです。次のカテゴリの命令実行によるデータが転送されます。

並列ロード/ストア命令
 部分ロード/ストア命令
 ダイレクト・アドレッシング・ロード/ストア命令
 即値インデクス・ロード/ストア命令

ロード/ストア命令については、*μPD77016 ファミリー ユーザーズ・マニュアル 命令編*を参照してください。

また、データ・バスは、X データ・バス、Y データ・バス、ペリフェラル・バスに分類されます。それらの論理的および物理的な関係を機能ブロックとの関係から表示すると表 4-2のようになります。

表 4-2 機能ブロックとバス

機能ブロック	Xデータ・バス, Yデータ・バス, ペリフェラル・バスの関係
内部メモリ周辺	Xデータ・バスとYデータ・バスは, 論理的かつ物理的に分離されています。したがって, 1つのインストラクションでXデータ・バス, Yデータ・バスを同時に有効化した転送が可能となります。
内部ペリフェラル	Xデータ・バスとYデータ・バスは, 論理的かつ物理的に共通に接続されています。したがって, ペリフェラル関連レジスタに, X, Yどちらのメモリ空間からアクセスしても, アドレスが同じであれば同一のペリフェラル・レジスタにアクセスします。ただし, このとき, 1つのインストラクションでX, Yデータ・メモリ空間から同時にペリフェラル・レジスタにアクセスすることはできません。
外部メモリ	Xデータ・バスとYデータ・バスは, 論理的には分離していますが物理的には共有しています。したがって, 1つのインストラクションでX, Y外部メモリに同時にアクセスすることはできません。

(2) Xデータ・バス

汎用レジスタとXデータ・メモリ, および内蔵ペリフェラルからのバスを接続する16ビットのバスで, 次のカテゴリの命令実行によるデータが転送されます。

並列ロード/ストア命令 (Xデータ・メモリ対象)

部分ロード/ストア命令 (Xデータ・メモリ対象)

ダイレクト・アドレッシング・ロード/ストア命令 (Xデータ・メモリ対象)

即値インデクス・ロード/ストア命令 (Xデータ・メモリ対象)

- 注意 1.** デバイス内部でXデータ・バスとYデータ・バスは分離していますが, 外部では単一のデータ・バスを共用しています。したがって, 同一のインストラクション・サイクルで両方の外部メモリに同時にアクセスする命令は実行できません。
- 2.** 内蔵ペリフェラルは, Xデータ・メモリまたはYデータ・メモリどちらからアクセスしてもアドレスが同じであれば, 同一のペリフェラル・レジスタにアクセスします。
- 3.** 2の場合であっても, 同一のインストラクション・サイクルでX, Y両方のメモリ空間から同時にペリフェラル・レジスタにアクセスすることはできません。

表 4-3に, Xデータ・バスに接続されているレジスタおよびメモリを示します。

表 4-3 X データ・バス接続レジスタおよびメモリ

レジスタまたはメモリ名	アセンブリの予約名称	ロード (L) / ストア (S)
汎用レジスタ	R0-R7 R0E-R7E R0H-R7H R0L-R7L R0EH-R7EH	L/S
X 内部 RAM	-	ROM パス方向のみ
X 内部 ROM		
外部メモリ		L/S
内蔵ペリフェラル		

注意 汎用レジスタは 40 ビットで構成されますが、その特定の部分を転送対象に指定することができます。その場合の分割は下位 16 ビット (L パート)、中間 16 ビット (H パート)、上位 8 ビット (E パート) となります。詳しくは 4.6.2 汎用レジスタとデータ・フォーマットを参照してください。

(3) Y データ・バス

汎用レジスタと Y データ・メモリ、および内蔵ペリフェラルからのバスを接続する 16 ビットのバスで、次のカテゴリの命令実行によるデータが転送されます。

- 並列ロード/ストア命令 (Y データ・メモリ対象)
- 部分ロード/ストア命令 (Y データ・メモリ対象)
- ダイレクト・アドレッシング・ロード/ストア命令 (Y データ・メモリ対象)
- 即値インデクス・ロード/ストア命令 (Y データ・メモリ対象)

- 注意 1.** デバイス内部で X データ・バスと Y データ・バスは分離していますが、外部では単一のデータ・バスを共用しています。したがって、同一のインストラクション・サイクルで両方の外部メモリに同時にアクセスする命令は実行できません。
- 2.** 内蔵ペリフェラルは、X データ・メモリまたは Y データ・メモリどちらからアクセスしてもアドレスが同じであれば、同一のペリフェラル・レジスタにアクセスします。
- 3.** 2 の場合であっても、同一のインストラクション・サイクルで X、Y 両方のメモリ空間から同時にペリフェラル・レジスタにアクセスすることはできません。

表 4-4 に、Y データ・バスに接続されているレジスタおよびメモリを示します。

表 4-4 Y データ・バス接続レジスタおよびメモリ

レジスタまたはメモリ名	アセンブリの予約名称	ロード (L) / ストア (S)
汎用レジスタ	R0-R7 R0E-R7E R0H-R7H R0L-R7L R0EH-R7EH	L/S
Y 内部 RAM	-	ROM パス方向のみ
Y 内部 ROM		
外部メモリ		L/S
内蔵ペリフェラル		

注意 汎用レジスタは 40 ビットで構成されますが、その特定の部分を転送対象に指定することができます。その場合の分割は下位 16 ビット (L パート)、中間 16 ビット (H パート)、上位 8 ビット (E パート) となります。詳しくは 4.6.2 汎用レジスタとデータ・フォーマットを参照してください。

4.2.3 ペリフェラル - メモリ間転送バス

内部ペリフェラルと内部データ・メモリとを接続し DMA 転送を行うバスです。DSP コア・カーネルを介さないバックグラウンド転送をサポートします。

詳細は 5.8 ペリフェラル - メモリ間転送 (PMT) を参照してください。

4.3 システム制御ユニット

μ PD77210 ファミリが DSP として動作するのを、背景で支えている次の基本機能群を、システム制御ユニットといいます。

- クロック・ジェネレータ
- リセット機能
- パイプライン・アーキテクチャ
- スタンバイ機能

4.3.1 クロック・ジェネレータ

クロック・ジェネレータは、CPU に供給されるシステム・クロックを生成および制御する回路です。PLL、分周器およびクロック制御回路から構成されています。

CLKIN 端子に入力された外部クロックから内部システム・クロックを生成し、デバイス内部の基本タイミングの基準にします。同時に内部システム・クロックは CLKOUT 端子からも出力され、外部デバイスとの同期のタイミングを測ることができます（この機能はレジスタ設定により無効とすることもできます）。PLL と出力分周器を備え、逡倍および分周したクロックをデバイス内部に供給することができます。PLL と出力分周器はユーザ・プログラムによって起動および停止が可能です。

リセットによって PLL および出力分周器は起動状態になります。リセット直後の内部システム・クロックは CLKIN 端子からの入力クロックとなっています。ブートのはじめに PLL 出力クロックが内部システム・クロック・ソースとして選択されます（ブート・モードによっては PLL 起動および PLL 出力クロック選択されない）。ブート・モードと PLL の起動に関する詳細は、**第6章 ブートアップ機能**を参照してください。

PLL 逡倍率 m の設定は外部端子（PLL0-PLL3）によって行います。逡倍率の設定値は 10 倍から 64 倍までの範囲で指定でき、端子設定値と逡倍率の関係は表 4-5 のようになります。また、出力分周率 n はレジスタにより 1/1 倍から 1/16 倍の範囲で指定できます。

表 4-5 PLL 逡倍率設定

端子設定値				逡倍率 (m)
PLL3	PLL2	PLL1	PLL0	
0	0	0	0	10
0	0	0	1	12
0	0	1	0	14
0	0	1	1	16
0	1	0	0	18
0	1	0	1	20
0	1	1	0	22
0	1	1	1	24
1	0	0	0	26
1	0	0	1	28
1	0	1	0	30
1	0	1	1	32
1	1	0	0	40
1	1	0	1	48
1	1	1	0	56
1	1	1	1	64

PLL 制御回路では、入力されたクロックを 10～64 倍に逡倍します。逡倍された周波数が、仕様で定められている PLL ロック周波数内に収まるように指定してください。なお、PLL のロック周波数は 80～160 MHz となっており、表 4-6 で示すように 2 つのロック・レンジ帯を持っています。リセット時に P3 端子によってこのロック・レンジ帯を指定してください。

出力分周器では分周器に入力されるクロックを 1/1 から 1/16 の整数分の 1 に分周します。最終的に DSP 内部に供給される外部入力クロックの m/n 倍の周波数が、仕様で定められている μ PD77210 ファミリの動作周波数内に収まるように指定してください。

表 4-6 PLL ロック・レンジ設定

端子設定値	PLL ロック・レンジ
P3	
0	120～160 MHz
1	80～120 MHz

スタンバイ・モード (HALT モード, STOP モード) 時は自動的に分周器出力クロックが内部システム・クロックのソースとして選択されます。したがって、スタンバイ・モードを実行する前に必ず分周器を起動させておく必要があります。通常動作時に分周器出力クロックをクロック・ソースとして動作させている場合は、スタンバイ・モード突入でクロック・ソースが切り替わることはありません。スタンバイ・モードからの復帰時は突入直前のクロック・ソースに戻ります。

内部システム・クロックを CLKOUT 端子から“出力する”,あるいは“出力しない”ことをレジスタによって選択できます。

クロック制御に関する詳細は、5.12 クロック・コントローラ (CLKC) を参照してください。

4.3.2 リセット機能

(1) ハードウェアの初期化

$\overline{\text{RESET}}$ 端子入力を規定の期間 (CLKIN 端子に入力するクロックに対して 6 サイクル以上) アクティブ (ロウ・レベル) にすることで、デバイスをハードウェア・リセットすることができます。プログラム実行の前にデバイスを正しく初期化することがリセットの目的です。端子状態および内部状態を初期化するために必ずリセットが必要です。電源投入時にリセットが行われない場合は端子状態が定まらず、最悪の場合、接続されているほかのデバイスの動作に悪影響を与える可能性があります。

初期化対象のレジスタと信号端子およびそれらの初期化値を表 4-7 から表 4-9 に、また、図 4-3 にリセット動作タイミングを示します。

リセット後に $\overline{\text{RESET}}$ 端子をインアクティブ (ハイ・レベル) にすると P0-P2 端子の設定値に応じてブート動作を行います。ブート動作によって、各端子および各レジスタの値がどう変化するかについては、**第 6 章 ブートアップ機能**を参照してください。同時に P3 端子の設定に応じて PLL のロック・レンジの指定を行います。P0-P3 端子の値はリセットが解除される 3 クロック前からブート終了時までの間、安定している必要があります。

図 4-3 リセット動作タイミング (1/2)

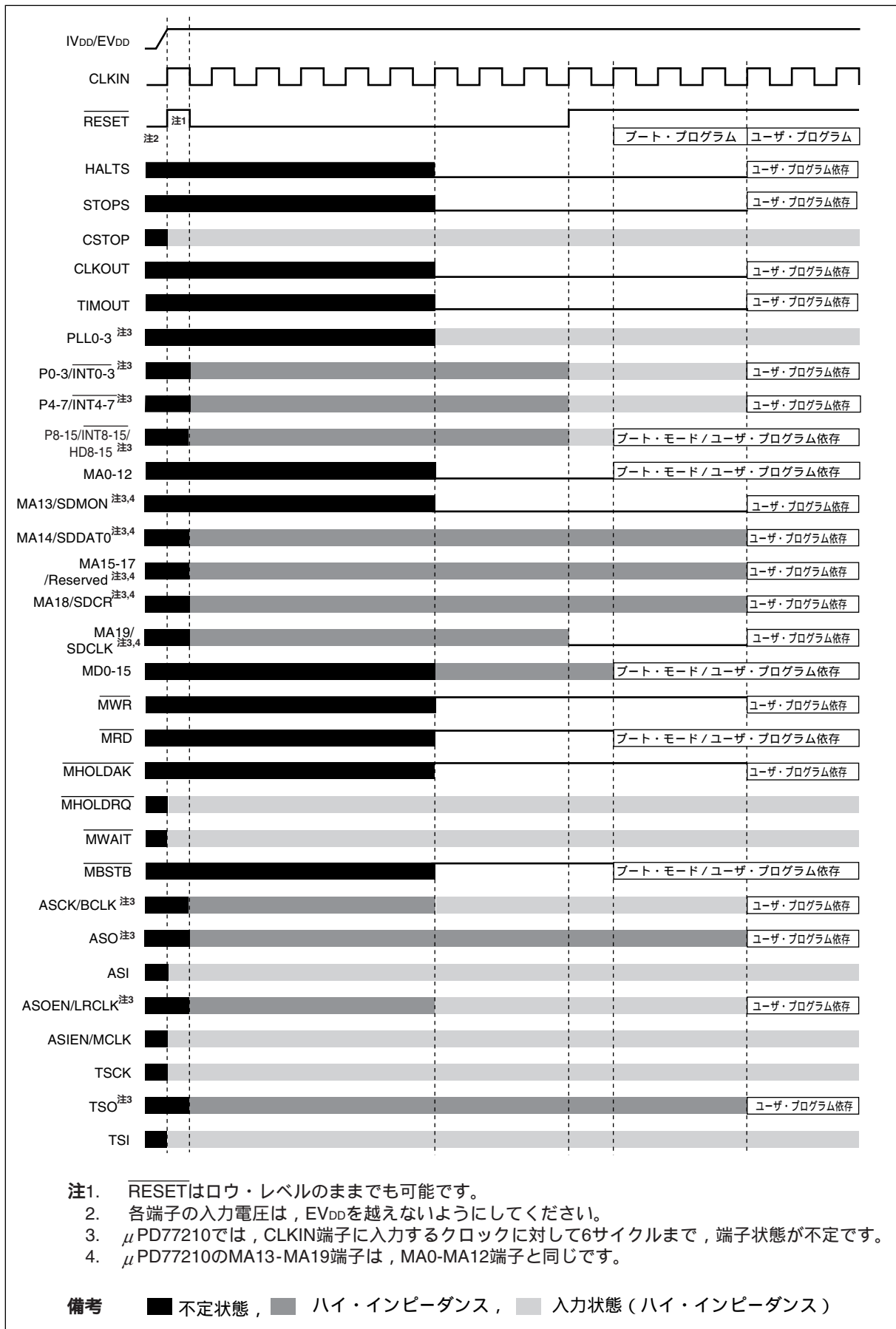


図 4-3 リセット動作タイミング (2/2)

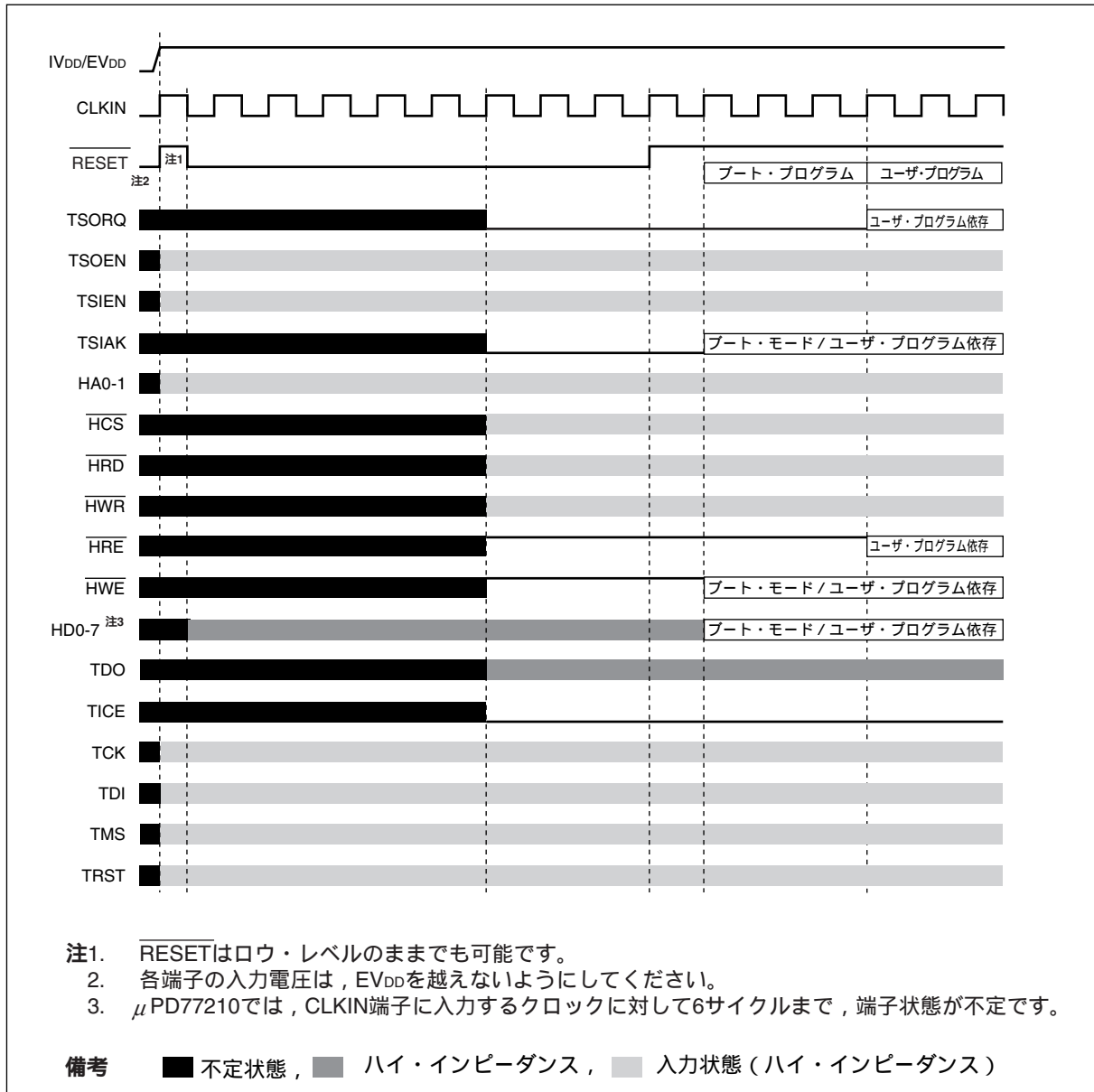


表 4-7 初期化対象 CPU レジスタと初期化値

レジスタ名	初期値
SR	0xF000
PC	0
SP	0
LC	0b1xxx xxxx xxxx xxxx
LSP	0
RC	0b1xxx xxxx xxxx xxxx
EIR	0xFFFF
ESR	0

表 4-8 初期化対象ペリフェラル・レジスタと初期化値

レジスタ名	初期値
SST1	0x00C2
TSST	0x0005
TFMT	0x0000
TTXL, TTXH, TRXL, TRXH	0xFFFF
SST2	0x0002
ASST	0x8012
HST	0x0301
MSHW	0x0000
MCST	0x0000
MWAIT	0xFFFF
PMC0-PMC7	0x0000
PCD0-PCD3	0x0000
ICR0-ICR11	0x0000
TCSR0, TCSR1	0x0000
CFER	0x0000

表 4-9 初期化対象端子と初期状態

端子名	初期状態
MA0-MA19	ロウ・レベル出力 ^{注1,2}
MD0-MD15	ハイ・インピーダンス
$\overline{\text{MWR}}$, $\overline{\text{MRD}}$, $\overline{\text{MBSTB}}$	ハイ・レベル出力 ^{注1}
TSORQ, TSIK	ロウ・レベル出力
ASO, TSO	ハイ・インピーダンス ^{注2}
ASCK/BCLK, ASOEN/LRCLK	ハイ・インピーダンス ^{注2}
HD0-HD7	ハイ・インピーダンス ^{注2}
$\overline{\text{HWE}}$, $\overline{\text{HRE}}$	ハイ・レベル出力
STOPS, HALTS	ロウ・レベル出力
CLKOUT	ロウ・レベル出力
TIMOUT	ロウ・レベル出力
P0-P15	ハイ・インピーダンス(入力状態) ^{注2}
TICE	ロウ・レベル出力

注1. バス解放時 ($\overline{\text{MHOLDAK}} = 0$) に、ハイ・インピーダンス状態になります。 $\overline{\text{MHOLDRQ}} = 0$ にすることによって、リセット中もバス解放が可能になります。

2. μ PD77213 においては、 $\overline{\text{RESET}} = 0$ によってハイ・インピーダンス状態になります(ただし、MA0-MA13 を除きます)。指定クロック入力後のリセット解除で初期状態になります。 μ PD77210 ではクロック入力により初期状態になるまでは不定です。

4.3.3 パイプライン・アーキテクチャ

μ PD77210 ファミリは、実行速度向上のためパイプライン・アーキテクチャを採用しています。

一般に、1つの命令は要素的な処理を担ういくつかのマシン・サイクルを推移することで処理が完結しますが、 μ PD77210 ファミリの場合は次に示す3つのマシン・サイクルがあります。

F：インストラクション・フェッチ・サイクル

命令コードを命令メモリから読み込みます。

D：デコード・サイクル

読み込んだ命令コードを解釈します。

E：実行サイクル

解釈した結果を実行します。

それぞれのマシン・サイクル実行部をパイプライン・ステージといいます。各ステージはそれぞれ独立に同じクロック・サイクル数（1サイクル）で処理を終了しますから、実行される命令に注目すれば、連続するステージを待ち時間なしに遷移し、しかも3つのステージに同時に3つの命令が存在できます。つまり、命令ストリームが障害なくパイプラインを通過しているかぎり、見かけ上1ステージの実行時間で1命令を処理することになります。この1ステージのクロック・サイクル数を1インストラクション・サイクルといいます。

μ PD77210 のインストラクション・サイクルは最小 6.25 ns（160 MHz 動作時）、 μ PD77213 のインストラクション・サイクルは最小 8.33 ns（120 MHz 動作時）です。

次にパイプライン処理のイメージを示します。

図 4-4 (a) ではパイプライン・ステージを固定し、その中を命令が流れてゆくイメージを表しています。

図 4-4 (b) では命令シーケンスに注目し、それぞれの命令が命令として完結したものを単位として、それらが次々と実行されるシーケンス・イメージを表しています。

図 4-4 パイプライン・イメージ (1/2)

(a) パイプライン・イメージ 1

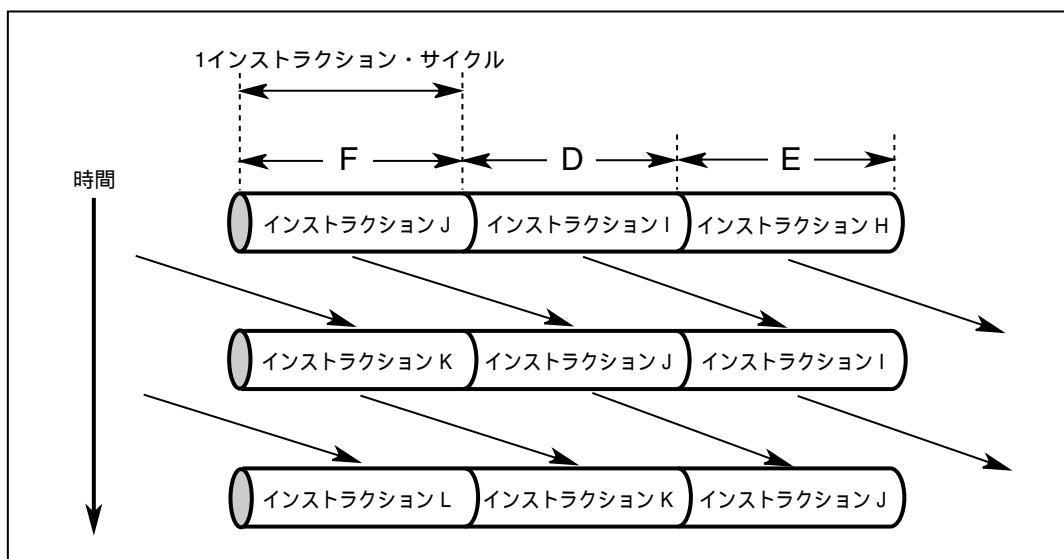
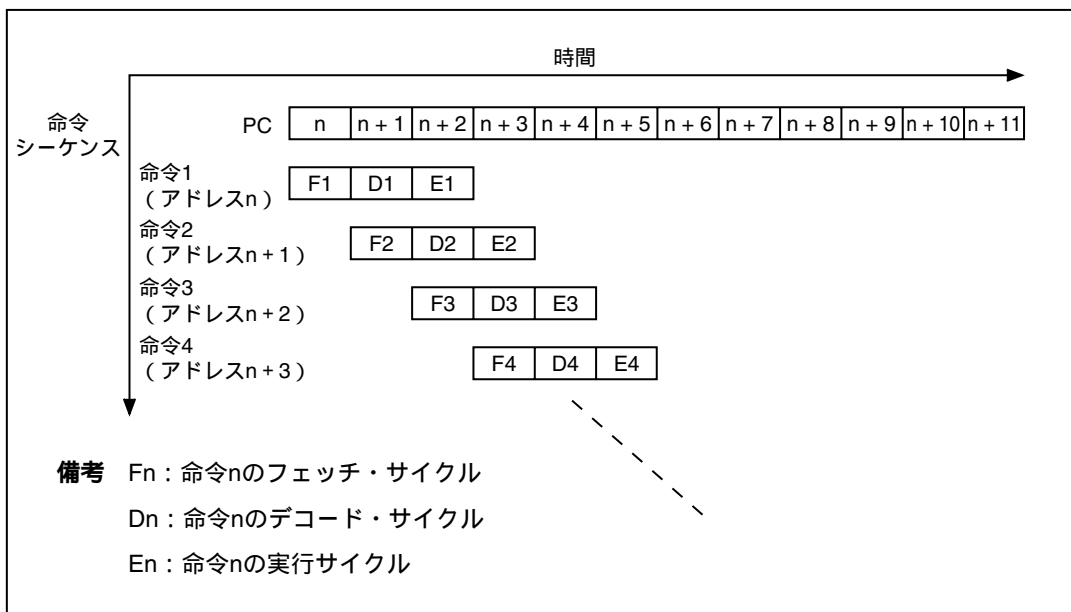


図4-4 パイプライン・イメージ (2/2)

(b) パイプライン・イメージ2



(1) 連続演算 (MAC, ALU, バレル・シフタ) について

連続する演算命令で直前の演算結果を利用する場合には、演算結果を汎用レジスタに書き戻すと同時に今回の演算のために演算器にも入力しますから、パイプラインを意識しないプログラミングが可能です。

(2) 分岐命令について

分岐命令でパイプライン・ハザードが発生した場合、遅延スロットにはNOP命令が挿入されてパイプラインが再充填されるので実行時間はよけいにかかりますが、アプリケーション作成に影響はありません。結果として、分岐命令のプログラミングでもパイプラインを意識する必要がありません。分岐命令とパイプライン・タイミングの詳細については、4.4.2 プログラム実行制御部を参照してください。

注意 次の3つの場合はパイプラインによる処理の遅延を考慮する必要があります。

割り込みを制御する命令 (EIR の設定など) は、割り込み制御情報を更新するまでに2インストラクション・サイクルを必要とします (4.4.4 割り込み参照)。

DPn に、レジスタ間転送命令または即値設定命令で値を設定した場合、それをアドレスとしてメモリをアクセスできるのはDPnを設定した命令の2命令後以降となります。

```
例 : inst#1 DP0 = 0x0100 ;
      inst#2 NOP          ; DP0 をここで使用することはできません。
      inst#3 ROL = *DP0  ;
```

ループの終端から3命令以内に分岐命令を記述することはできません (4.4.3 フロー制御部参照)。

4.3.4 スタンバイ機能

μ PD77210 ファミリは、デバイスの動作を停止して消費電流を抑制するスタンバイ機能を備えています。デバイスは命令によってスタンバイ状態になり、デバイスがスタンバイしている状態をスタンバイ・モードと呼びます。スタンバイ・モードに移行する命令には HALT 命令と STOP 命令の 2 種類があります。

(1) HALT 命令によるスタンバイ・モード

HALT 命令の実行により、CLKC レジスタで設定した分周クロックに自動的に切り替わります。このため HALT 命令を実行する前には必ず分周率の設定を行い、分周器の動作をオンにしておく必要があります。

HALT モード解除の外部割り込み信号は分周クロックで 4 サイクル分が必要です。復帰の際に用いる割り込み要因は許可状態としてください。

リセットによる復帰を行う場合、リセットによって強制的に外部クロックに切り替わるためクロック・ハザードが生じます。このときの内部メモリの内容は保証されませんので、必ずブートを実行してください。

HALT モード中の端子状態は、CLKOUT 端子が分周出力、HALTS 端子がハイ・レベルになる以外は直前の状態を維持します。

表 4-10 ホールト中の端子状態

端子名	$\overline{\text{MHOLDRQ}}$ がアクティブ (ロウ・レベル) な場合	$\overline{\text{MHOLDRQ}}$ がインアクティブ (ハイ・レベル) な場合
CLKOUT ^{注1}	システム・クロック出力 ^{注2}	
MA19-MA0	ハイ・インピーダンス	直前の状態を保持
MD15-MD0	ハイ・インピーダンス	
$\overline{\text{MRD}}$ / $\overline{\text{MWR}}$	ハイ・インピーダンス	ハイ・レベル
$\overline{\text{MHOLDAK}}$	ロウ・レベル	ハイ・レベル
$\overline{\text{MBSTB}}$	ハイ・レベル	
TSORQ, TSIK, TSO, ASO	直前の状態を保持	
$\overline{\text{HRE}}$, $\overline{\text{HWE}}$, HD15-HD0	直前の状態を保持	
P15-P0	直前の状態を保持	
TDO, TICE	直前の状態を保持	

注 1. CLKC レジスタで CLOKOUT 出力を不許可にしている場合は、ロウ・レベルのままです。

2. ホールト・モードでの分周クロックのデューティは、50%ではなくハイ・レベル幅が 1 サイクル分のクロックになります。

注意 ホールト中にハイ・インピーダンスになる端子および入力端子は、ハイ・レベルまたはロウ・レベルに固定してください。

★ (a) 制限事項

HALT 命令を実行した場合、システム・クロックが自動的に分周クロックへ切り替えられます。そのときに割り込み受け付けによる HALT 解除が競合した場合、HALT 解除指示信号をサンプリングできずに、システム・クロック停止によるデッドロックが発生することがあります。

HALT 命令の実行と、割り込みによる HALT 解除が競合しないようにしてください。

問題が発生する条件については次のようになります。

< 正常に動作する場合 >

DSP の HALT 命令実行にて HALTS 端子がセット

HALTS の信号を受け、システム・クロックを分周クロックに切り替え

HALTS 端子がセット以降の割り込みで HALT 解除

< 不具合が発生する場合 >

DSP の HALT 命令実行にて HALTS 端子がセット

同時に、すでにサンプリングされた割り込み信号が存在

HALT セットと解除が競合し、HALTS の信号は、少サイクル信号化

分周回路側でサンプリングできずにデッドロック

(2) STOP 命令によるスタンバイ・モード

STOP 命令を実行することによってこのモードに移行し、このときデバイス消費電流は数百 μ A (PLL 停止時) に低下します。ストップ・モードへ移行、ストップ・モードからの復帰の過程は次のようになります。

(a) STOP 命令でストップ・モードに移行する。

(b) PLL を停止した場合には、デバイスの消費電流は数百 μ A 程度 (PLL 停止時) に低下する。

(c) ストップ・モードからの復帰はハードウェア・リセットあるいは CSTOP 端子による。

STOP 命令の実行により、CLKC レジスタで設定した分周クロックに自動的に切り替わります。このため STOP 命令を実行する前には必ず分周率の設定を行い、分周器の動作をオンにしておく必要があります。

STOP モード中でも PLL は停止しません。消費電流をさらに低減するためには、STOP 命令を実行する前にクロック・ソースを CLKIN 端子ダイレクトにし、PLL を停止させておく必要があります。

PLL を停止しないときは PLL のロックアップを不要とするため、STOP モードからの高速復帰が可能です。ただし、ロック状態を維持するために CLKIN 端子へのクロックを停止しないでください。

命令 ROM 上で STOP 命令を実行した場合、ROM の電流が停止できないために、消費電流が低減されないことがあります。命令 RAM 上での STOP 命令実行を推奨します。

STOP モードの解除は CSTOP 端子にハイ・レベルを入力することで行います。分周クロックで 12 サイクル分のハイ・レベル幅が必要です。

リセットによる復帰を行う場合、リセットによって強制的に外部クロックに切り替わるためクロック・ハザードが生じます。このとき、内部メモリの内容を保証できませんので、必ずブートを実行してください。

STOP モード中の端子状態は、CLKOUT 端子が分周出力、HALTS、STOPS 端子がハイ・レベルになる以外は直前の状態を維持します。ただし、DSP コア・カーネルおよびペリフェラルにはクロックが供給されていないため、ペリフェラルが出力するアクセス要求フラグや割り込みを DSP コア・カーネルが検出することはできません。復帰の際は、CSTOP 端子のハイ・レベル入力でも STOPS 端子をロウ・レベルにし、CSTOP 端子のロウ・レベル入力でも復帰処理を行います。

4.4 プログラム制御ユニット

プログラムの実行制御に広くかかわるユニットです。このユニット内のいろいろなレジスタに対し、メイン・バスを介してデータをロード/ストア可能であるほか、このユニットは次のカテゴリの命令実行にかかわります。

- 命令実行一般
- 分岐命令
- ハードウェア・ループ命令
- 割り込み（命令ではありませんが、PC, STK, SP, SR, EIR が INTC により、自動的に管理されます）

これらの命令実行は、プログラム制御ユニットを機能面から区分した、次の3つの部分に対応しています。

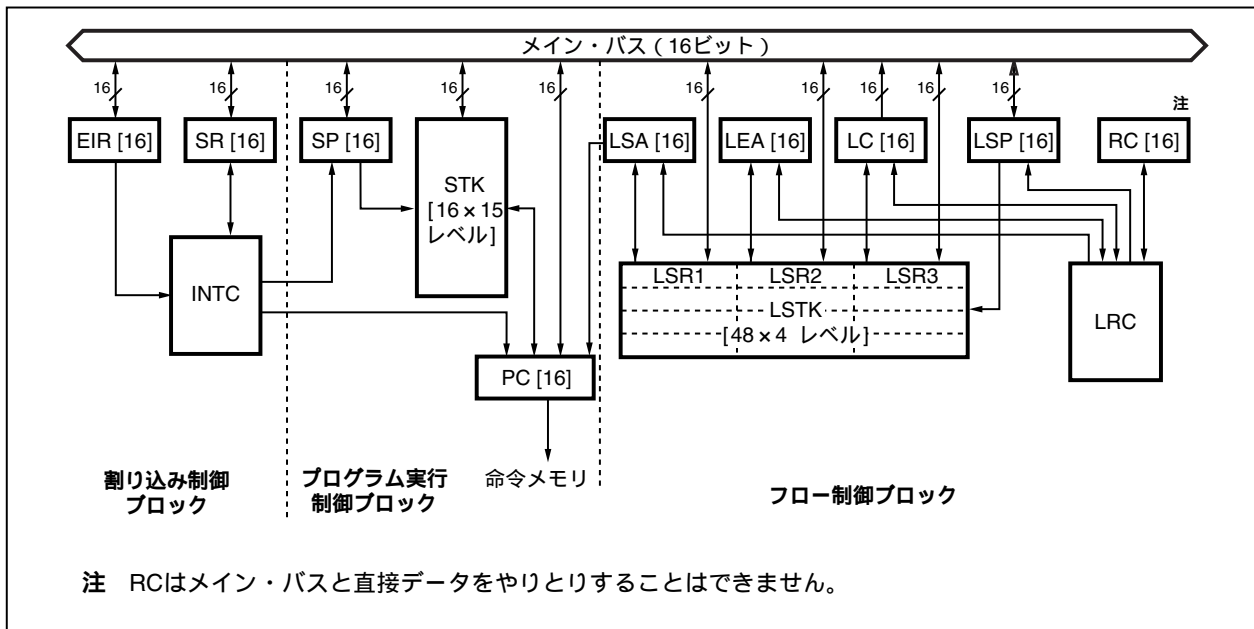
- プログラム実行制御部
- フロー制御部
- 割り込み制御部

4.4.1 **ブロック構成**でプログラム制御ユニットの詳細なブロック図を示し、さらに4.4.2 **プログラム実行制御部**から4.4.4 **割り込み**でそれぞれの機能の詳細を説明します。

4.4.1 ブロック構成

プログラム制御ユニットのブロック構成を図4-5に示します。

図 4-5 プログラム制御ユニット



4.4.2 プログラム実行制御部

プログラム実行は、次のレジスタにより管理され推移します。

- プログラム・カウンタ (PC)
- スタック (STK)
- スタック・ポインタ (SP)

(1) プログラム・カウンタ (PC)

プログラム実行にあたって、インストラクション・アドレスを保持する 16 ビットのレジスタです。したがって、PC に設定できる値の範囲は命令メモリ空間全体に一致します。

注意 PC 自身は 16 ビットの範囲でどのような値をとることもできますが、命令メモリ空間として定義されていない部分、またはシステム用として予約されている部分へアクセスすることは禁止されています。

(a) 内部命令メモリ

μ PD77210 ファミリは、内部命令メモリとして ROM または RAM を内蔵しています。これらの容量は製品ごとに異なります。それらの相違を表 4-11 に示します。

表 4-11 内部命令メモリの容量

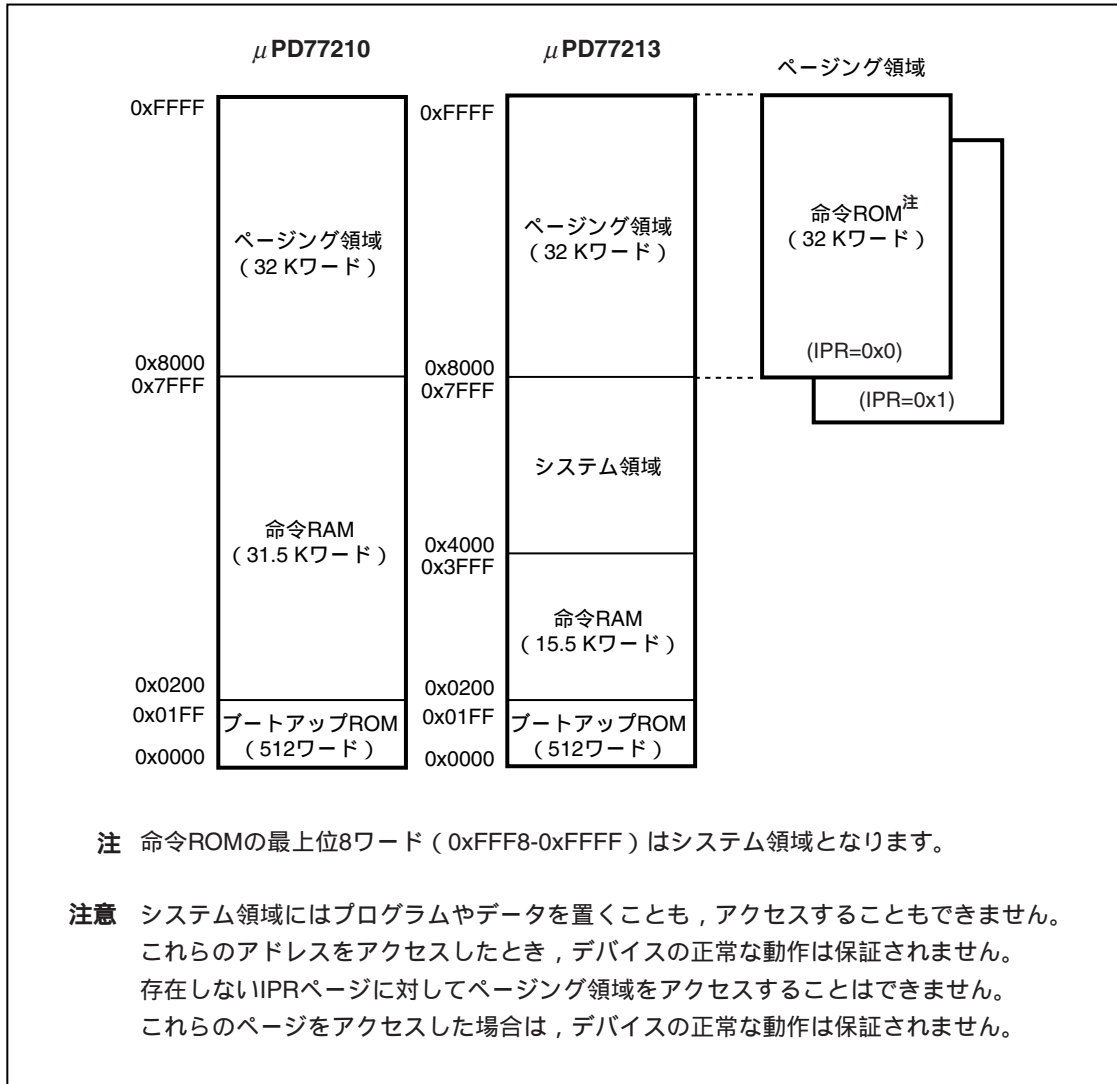
品名	内部 ROM 容量	内部 RAM 容量
μ PD77210	なし	31.5 K ワード
μ PD77213	64 K ワード	15.5 K ワード

(b) 命令メモリ

次にμPD77210ファミリの命令メモリ・マップを示します。

なお、命令領域をデータ領域としてアクセスする方法があります。詳細については、4.5.3 命令メモリのエイリアスを参照してください。

図 4-6 命令メモリ・マップ



(2) スタック (STK) とスタック・ポインタ (SP)

スタック (STK) はプログラム・カウンタ (PC) のセーブ/リストア専用のレジスタ・ファイルで, 16 ビット×15 レベルあります。

用途は次の2点です。

サブルーチン・コールで, 戻りアドレスをセーブする。

割り込みで, 現在実行中のアドレスをセーブする。

割り込みについては4.4.4 **割り込み**を参照してください。

現在アクセス対象となっているスタック・レベル (スタック・トップといいます) を示すポインタ・レジスタがスタック・ポインタ (SP) です。SP は 16 ビットで構成されますが, 0-15 以外の値を設定することは禁じられています。スタック・トップと SP はメイン・バスに接続されていますから, メイン・バスを介して汎用レジスタとデータ交換が可能です。スタックがオーバフローまたはアンダフローすると, エラー・ステータス・レジスタのステータス・エラー・フラグが1にセットされます。

備考 RET または RETI 命令を, STK または SP にロード/ストアするレジスタ間転送命令の直後に記述しないでください。

(3) 関連する命令

プログラム・カウンタ (PC), スタック (STK), スタック・ポインタ (SP) の動作は, 命令実行の側面から次の2種類と考えることができます。

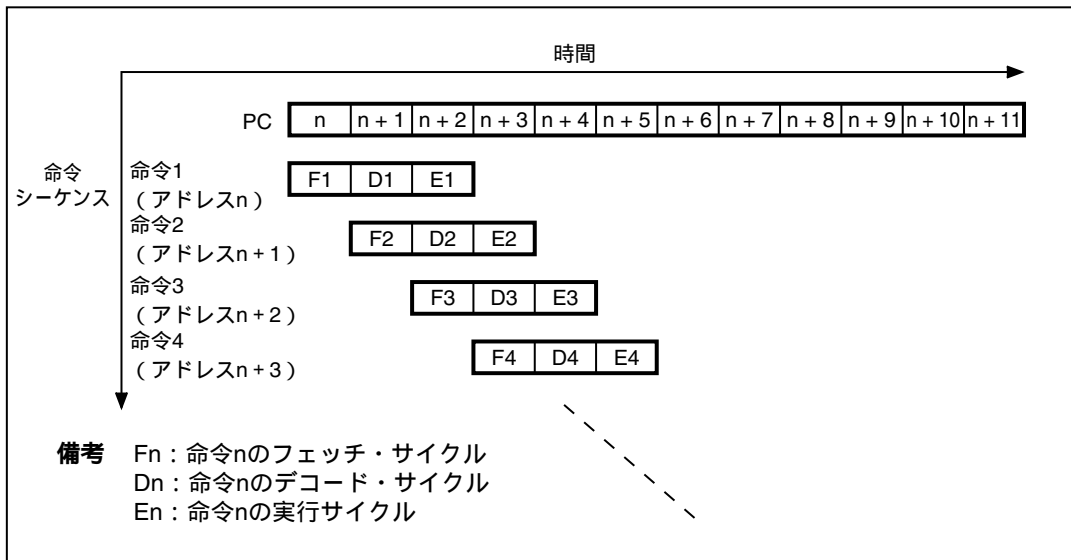
命令実行と PC の動作

分岐命令と PC, SP, STK の動作

(a) 命令実行と PC (通常動作)

すべての命令フェッチごとに PC をインクリメントします。これをパイプライン実行と組み合わせた場合のイメージを図 4-7に示します。

図 4-7 PC の通常動作



(b) 分岐命令と PC, SP, STK の動作

分岐命令は次の3種類に分類できます。

ジャンプとサブルーチン・コール

現在実行中の命令アドレス (PC 値) をスタックに保存するか否かで分類されます。

JMP 命令

実行中の命令アドレスを保存しません。したがって、分岐先から分岐元へ自動的に戻ることはできません。

サブルーチン・コール命令

実行中の命令アドレス (サブルーチン・コール命令の次のアドレス) をスタックに保存します。分岐先から分岐元へ戻るときはリターン命令を使用します。

PC への設定形態から見た分岐

分岐先アドレスが PC へ設定される形態から分岐を見ると、次の2種類に分類されます。

イミディエト・ジャンプ/コール

この形態をイミディエト・ジャンプまたはイミディエト・コールと呼びます。JMP/CALL 命令において、オペランドに数値を記述するタイプはこの形態の分岐です。このとき数値は 16 ビットの 2 の補数として現在の PC 値に加算/減算され、全体で ±32 K ワードの範囲、したがって最終的に 64 K ワード空間全域に分岐できることになります。

注意 この命令をアセンブラで記述する場合、オペランドには直接分岐先アドレス、またはレベルを記述します。この命令は、実際は現在の PC 値からの相対分岐命令です。

レジスタ間接ジャンプ/コール

この形態をレジスタ間接ジャンプまたはレジスタ間接コールと呼びます。

JMP/CALL 命令において、オペランドに DPn レジスタを記述するタイプはこの形態の分岐です。このとき DPn レジスタの値は直接 PC に設定されます。

条件の有無による分類

μPD77210 ファミリには条件分岐命令、条件リターン命令というカテゴリはなく、条件分岐は条件命令と分岐命令、リターン命令の組み合わせで実現されます。これらは次の2つに分類されます。

無条件 JMP/CALL/RET 命令

常に (無条件に) 分岐 (JMP/CALL/RET) します。

条件 JMP/CALL/RET 命令

組み合わせた条件命令の条件が真であるときにかぎり分岐 (JMP/CALL/RET) します。

これらをまとめると表 4-12 のようになります。注意する点は、条件付きか否かで、分岐が発生した場合の処理実行順序に違いはありませんが、実際に分岐にかかる時間は、条件命令と組み合わせた場合 1 インストラクション・サイクル多くかかります。また、表には示されていませんが、条件命令付き分岐で条件不成立の場合は、パイプライン・ハザードによる遅延は発生しません (図 4-8 から図 4-11 を参照)。

表 4-12 分岐命令の分類

命令名称	条件判定	アドレス指定	語長	インストラクション・サイクル
ジャンプ命令	条件なし	PC 相対	1	2
	条件あり			3
	条件なし	レジスタ間接絶対	1	3
	条件あり			
サブルーチン・コール命令	条件なし	PC 相対	1	2
	条件あり			3
間接サブルーチン・コール命令	条件なし	レジスタ間接絶対	1	3
	条件あり			
リターン命令	条件なし	-	1	2
	条件あり			3
割り込みリターン命令	条件なし	-	1	2
	条件あり			3

注意 インストラクション・サイクルの値は、条件が満足したとき、またはプログラムが分岐したときです。条件が満足しない場合、分岐が行われないうえ、パイプライン・ハザードも発生しないので、1 インストラクションになります。

次に、それぞれのタイミングを図 4-8から図 4-11に示します。

無条件イミディエト・ジャンプ

無条件間接ジャンプ

条件イミディエト・ジャンプ (条件成立：分岐)

条件イミディエト・ジャンプ (条件不成立：通過)

それぞれの図の略称の意味は次のとおりです (n = 0, 1, 2, ...)

ifn : 命令フェッチ	jfn : ジャンプ先命令フェッチ	
idn : 命令デコード	exn : 命令実行	ia : 命令アドレス演算
addr : アドレス出力	p : パージ	push : スタック・プッシュ
pop : スタック・ポップ	jdec : ジャンプ先デコード	popi : 割り込みポップ

図 4-8 無条件イミューディート・ジャンプのタイミング

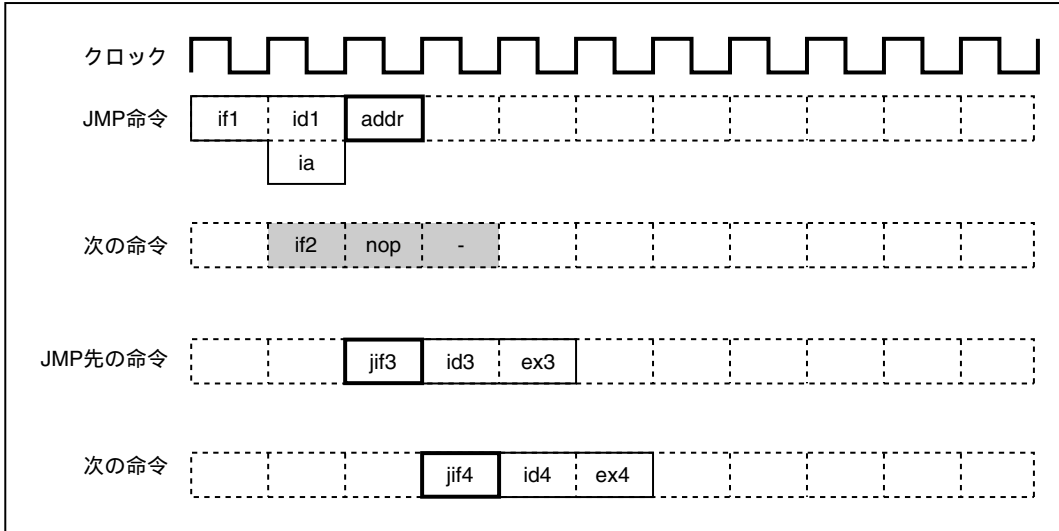


図 4-9 無条件間接ジャンプのタイミング

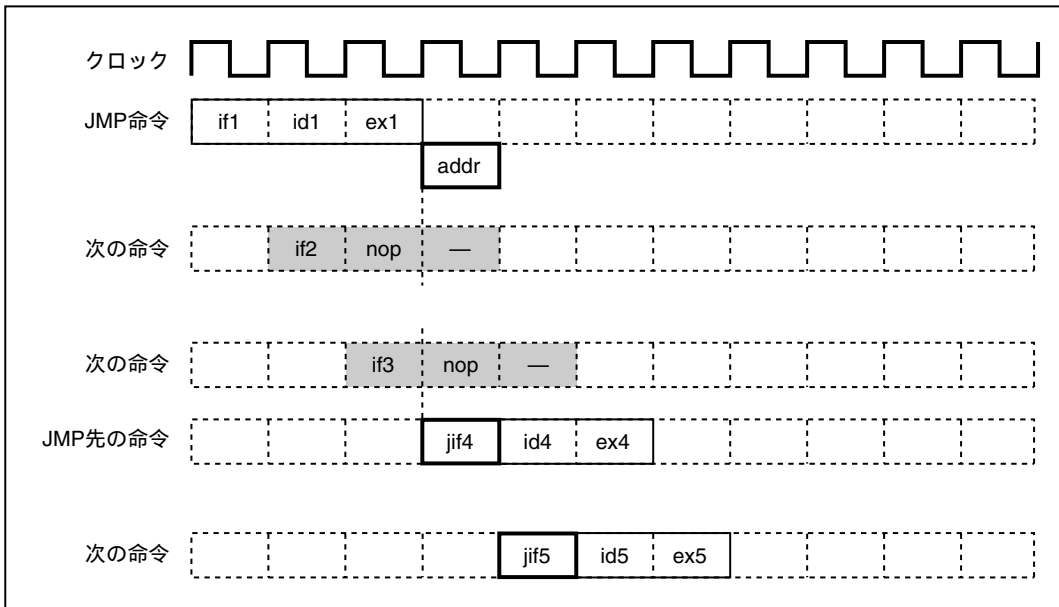


図 4-10 条件イミディエト・ジャンプのタイミング (条件成立: 分岐)

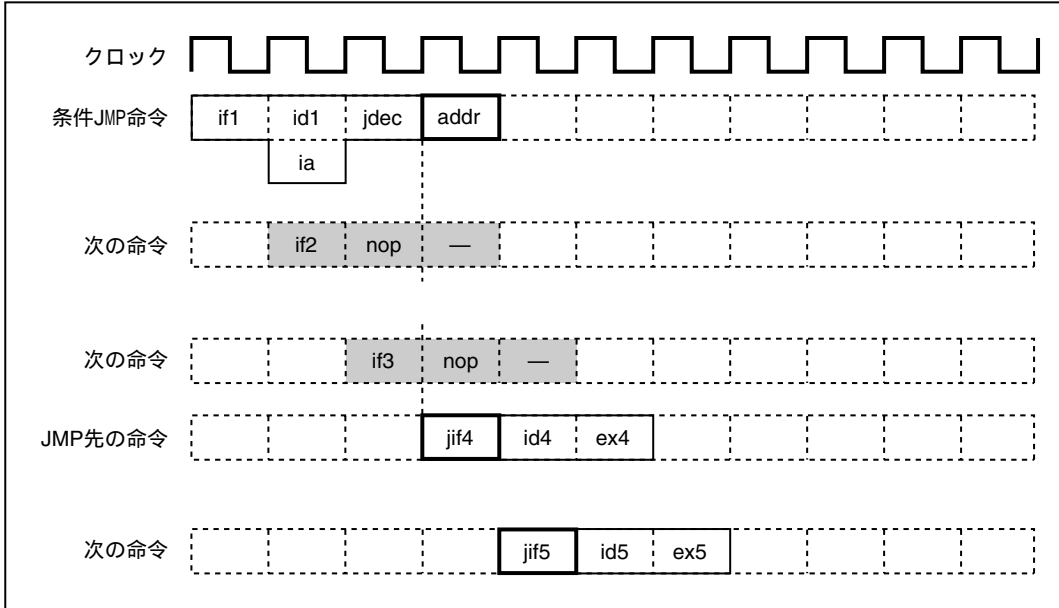
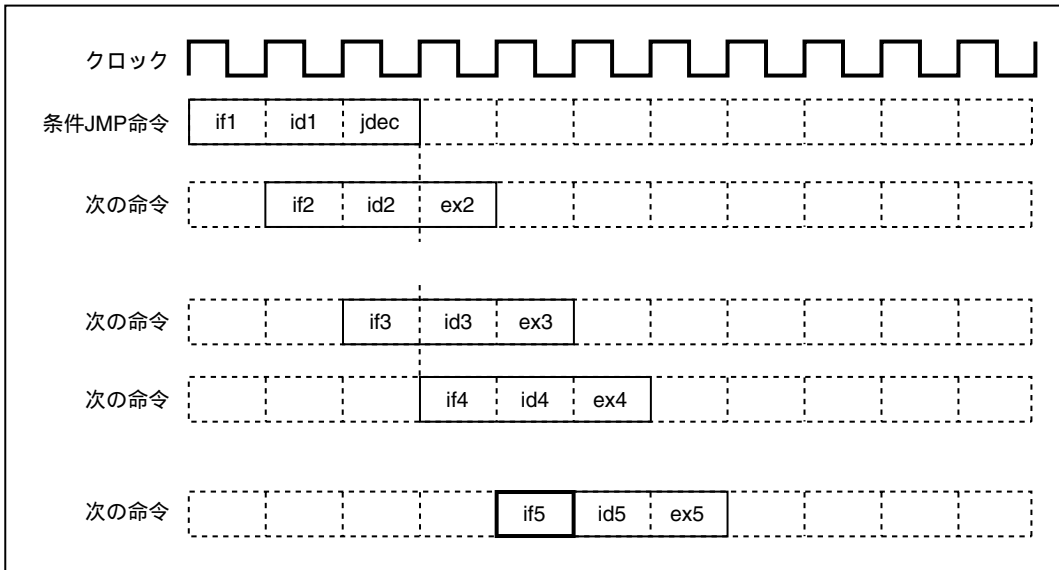


図 4-11 条件イミディエト・ジャンプのタイミング (条件不成立: 通過)



(c) サブルーチン・コール/リターン動作

サブルーチン・コールは CALL 命令によって実行されます。CALL 命令が発生した場合、次の手順で分岐します。

SP がインクリメントされる (プレインクリメント)。
 SP の示す STK に PC の値 (CALL 命令の次のアドレス) がセーブされる。
 PC に分岐先アドレスが設定される。このとき、分岐先が数値で与えられる場合は現在の PC 値に当該数値が 2 の補数形式で加減算され、DPn レジスタで与えられた場合はその値が直接 PC に設定される。

サブルーチンから戻る場合は RET 命令によって実行されます。RET 命令は次の順序で実行されます。

SP の示す STK から PC に値をリストアする。
 SP がデクリメントされる (ポスト・デクリメント)。

備考 CALL 命令のタイミング関係は JMP 命令のタイミングを参考にしてください。

CALL 命令では、リターン・アドレスがスタックにセーブされるほかは JMP 命令と同様のタイミングです。また、リターン命令のタイミングも、イミディエト・ジャンプと同様のタイミングです (2 インストラクション・サイクルかかります)。

(d) 割り込み時の動作

実行中の命令アドレス (割り込みを受け付けた命令のアドレス) をスタックに保存し、分岐先アドレスが PC に設定されます。割り込みから戻るときは RETI (割り込みリターン) 命令を使用します。

割り込みの動作については、4.4.4 **割り込み**を参照してください。

4.4.3 フロー制御部

一般的に高級言語は、洗練されたフロー制御シンタクスを整備しています (例: C 言語の for ループや while ループなど)。μPD77210 ファミリには、これらフロー制御を直接アセンブリ命令として記述できるハードウェアが用意されており、オーバーヘッドなしでループ/リピート動作が行えます。ループ/リピート動作はループ/リピート制御回路が制御します。

フロー制御は次のレジスタ、機能ブロックにより管理されます。

リピート・カウンタ (RC : repeat counter)

リピート命令の繰り返し回数を保持する 16 ビットのカウンタ用レジスタです。

ループ開始アドレス・レジスタ (LSA : loop start address register)

ループ実行中、ループ開始アドレスを保持する 16 ビットのレジスタです。

ループ終端アドレス・レジスタ (LEA : loop end address register)

ループ実行中、ループ終端アドレスを保持する 16 ビットのレジスタです。

ループ・カウンタ (LC : loop counter)

16 ビットのレジスタで、LOOP 命令開始時に初期値が設定され、ループを 1 回実行するごとにデクリメントし、0 になったらループ終了です。

ループ・スタック (LSTK : loop stack)

LSA, LEA および LC の値の退避 / 復帰をする 3 × 16 ビット × 4 レベルのレジスタ・ファイルです。ループ命令により LSA, LEA および LC の値を退避します。ループ終了またはループ・ポップ命令により LSA, LEA および LC に値を復帰します。

ループ・スタックは、次の 3 つの 16 ビット・レジスタとレジスタ間転送命令を使ってデータの入出力を行います。

- ・ LSR1 : ループ開始アドレスの退避 / 復帰 (LSA 用スタック)
- ・ LSR2 : ループ終端アドレスの退避 / 復帰 (LEA 用スタック)
- ・ LSR3 : ループ・カウンタの退避 / 復帰 (LC 用スタック)

レジスタ間転送命令のソースに LSR1 を指定すると、転送後に LSP をデクリメントします。レジスタ間転送命令のデスティネーションに LSR1 を指定すると、LSP をインクリメントしたあとに転送します。

ループ・スタック・ポインタ (LSP : loop stack pointer)

LSTK の現在位置を示すポインタです。16 ビット・レジスタですが設定できる値は 0-4 です。

LSP の値は、レジスタ間転送命令を使ってデータの入出力ができます。

リセットにより LSP の値は 0 になります。

インクリメント / デクリメントはビット 2-0 の 3 ビットで行い、ビット 15-3 は 0 に固定です。

次の場合にインクリメントを行います。

- ・ ループ命令により LSA, LEA および LC の値を LSTK に退避する場合
- ・ レジスタ間転送命令のデスティネーションに LSR1 を指定した場合

また、次の場合にデクリメントを行います。

- ・ ループ終了またはループ・ポップ命令により LSTK の値を LSA, LEA および LC に復帰する場合
- ・ レジスタ間転送命令のソースに LSR1 を指定した場合

注意 1. LSP の値が 0-4 以外するとき、ループ・スタック・オーバフローまたはループ・スタック・アンダフローが発生し、エラーであることを示します。

2. LSP の値に 5-0xFFFF は設定しないでください。

ループ / リpeat制御回路 (LRC : loop repeat controller)

ループ命令、リpeat命令を管理します。

注意 フロー制御用のレジスタのうち、リpeat・カウンタ (RC) 以外のレジスタは、すべてメイン・バスに接続されています。このため汎用レジスタとのデータ転送が可能です。

フロー制御には次の 2 つの機能があります。

リpeat機能 (REP 命令)

ループ機能 (LOOP 命令, LPOP 命令)

(1) リピート機能

REP 命令で記述されるリピート機能は、1 命令のみに対してカウント・ベースによる繰り返し (リピート) を実現します。リピートの対象となる命令は REP 命令の直後の命令です。

(a) リピート・カウンタ (RC) のフォーマット

図 4-12 にリピート・カウンタ (RC) のフォーマットを示します。

図 4-12 RC のフォーマット



注意 リピート動作中は割り込みを受け付けません。詳細については 4.4.4 割り込みを参照してください。

(b) リピート機能の要約

リピート機能は次のように要約できます。

- 単一の命令をリピート対象とする。
- リピート回数は直接数値で与えても、汎用レジスタ (R0L-R7L) で与えてもどちらでもよい。
- リピート回数の範囲は 1 から 32767 である。
- リピート動作中プログラム・カウンタ (PC) のインクリメントは行わない。
- 繰り返しごとに RC をデクリメントし、指定回数実行したらリピートを終了する。
- リピート機能は RC のみに依存するので、ループ命令のネスティングにカウントされない。

(c) リピート機能実行の手順

REP 命令に出会ったとき、リピート機能は次の手順で実行されます。

- REP 命令のパラメータとして与えられたリピート回数を RC に設定する。
- PC をインクリメントし、REP 命令の直後の命令をリピート対象とする。このとき 1 インストラクション・サイクルの無効サイクルが発生する。
- リピート実行中 PC は、リピート対象命令の次のアドレスを保持し続ける。
- リピート対象命令を 1 回実行するごとに RC をデクリメントする。指定回数実行したら、リピートを終了する。
- リピート終了で PC はインクリメントされる。このときリピート対象命令から次の命令実行への遷移過程で、パイプラインは連続する。したがってリピート終了時にオーバーヘッドは発生しない。
- リピート命令については、μPD77016 ファミリー ユーザーズ・マニュアル 命令編を参照。

(d) リピート実行のタイミング

REP 命令実行のタイミング (2 回リピートの例) を、次に示します。

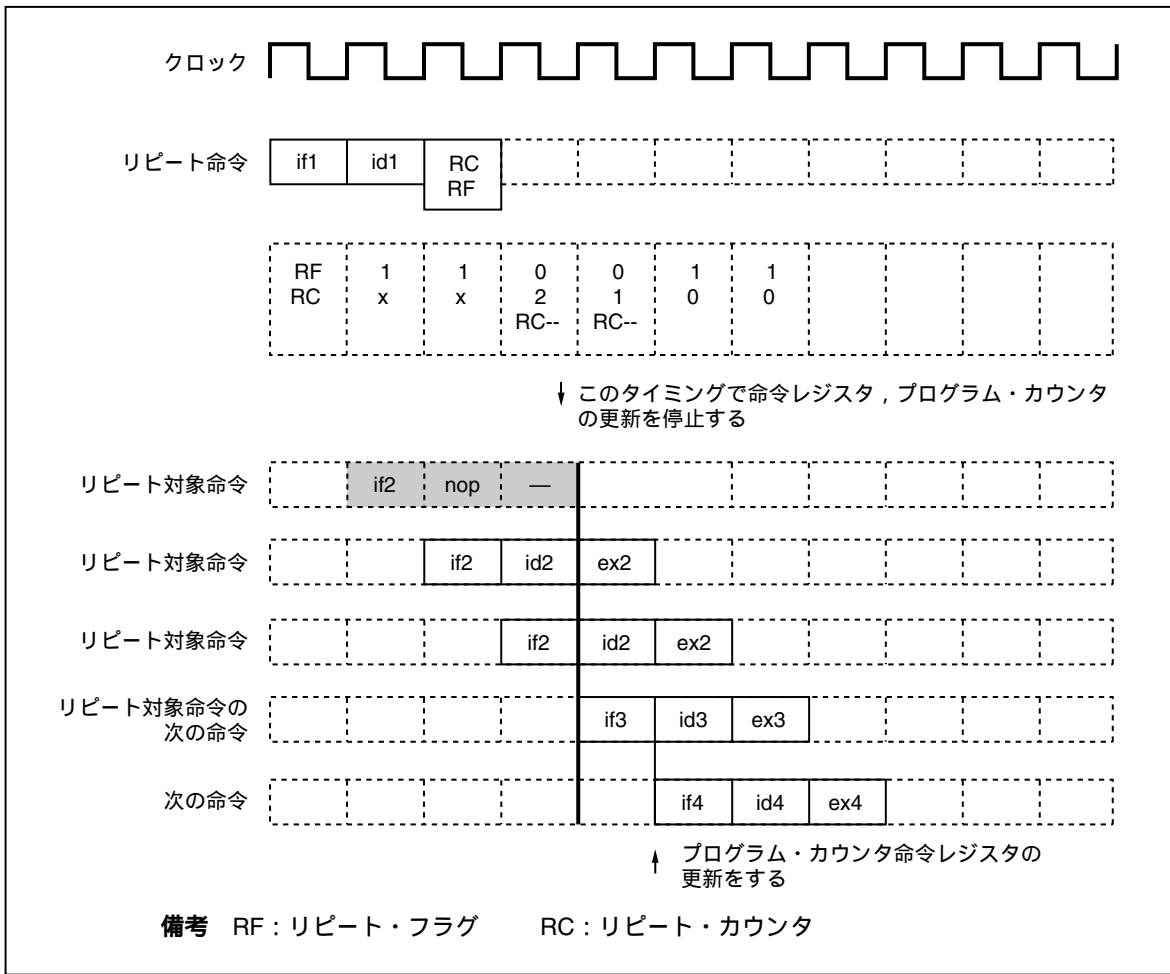
図 4-13 にアセンブリ・プログラムを、図 4-14 にその実行タイミングを、それぞれ示します。

図 4-13 リピート命令の例 (2 回リピート)

```

REP    2 ;
R0    /= R1 ;
    
```

図 4-14 リピート実行のタイミング (2 回リピート)



(2) ループ機能

LOOP 命令で記述されるループ機能は、2 命令以上 255 命令以下の命令群に対してカウント・ベースによるループ・フローを実現します。ループのネスティングは 4 レベルのハードウェア・ループ・スタックによってサポートされます。さらに、ループ途中の任意の時点でループを脱出できる LPOP 命令が用意され、柔軟なループ制御を可能にしています。

(a) ループ・カウンタ (LC) のフォーマット

図 4-15にループ・カウンタ (LC) のフォーマットを示します。

図 4-15 LC のフォーマット



備考 ループ・フラグ LC はステータス・レジスタ (SR) にも含まれています (4.4.4 割り込み参照)。

(b) ループ機能の要約

ループ機能は次のように要約できます。

2 命令以上 255 命令以下をグルーピングしてループ要素とする。

ループ回数は直接数値で与えても、汎用レジスタ (R0L-R7L) で与えてもどちらでもよい。

ループ回数の範囲は 1 から 32767 までである。

ループ・スタックがあるので 4 重までのネスティングが可能である。

ループ脱出は次の 2 つのうちのどちらかである。

- (1) カウント値が 1 になった。
- (2) LPOP 命令を実行し、JMP 命令を実行した。

備考 ループ操作と関連した割り込み処理については、4.4.4 割り込みを参照してください。

(c) ループ機能実行の手順

LOOP 命令に出会ったとき、ループ機能は次の手順で実行されます。

ループの開始時

1. LSP の値をインクリメントする (プレ・インクリメント)。
2. 現在の LSA, LEA, LC を LSP の指す LSTK にセーブする。
3. LSA にループ開始アドレスを設定する。
4. ループ終端アドレスを計算し, LEA に設定する。
5. LC にループ回数を設定する。

ループ動作中

1. PC と LEA の値が等しければ LC の値をデクリメントする。
2. LC が 1 でなければ LSA の値を PC に設定する。
LC が 1 であればループ終端処理を実行する。

ループ終端処理

1. PC の値をインクリメントする。
2. LSP の指す LSTK の値を LSA, LEA, LC にリストアする。
3. LSP の値をデクリメントする (ポスト・デクリメント)。

LPOP 命令によるループ終端処理

LPOP 命令では次の処理をして 1 レベルのループを破棄します。

1. LSP の指す LSTK の値を LSA, LEA, LC にリストアする。
2. LSP の値をデクリメントする (ポスト・デクリメント)。

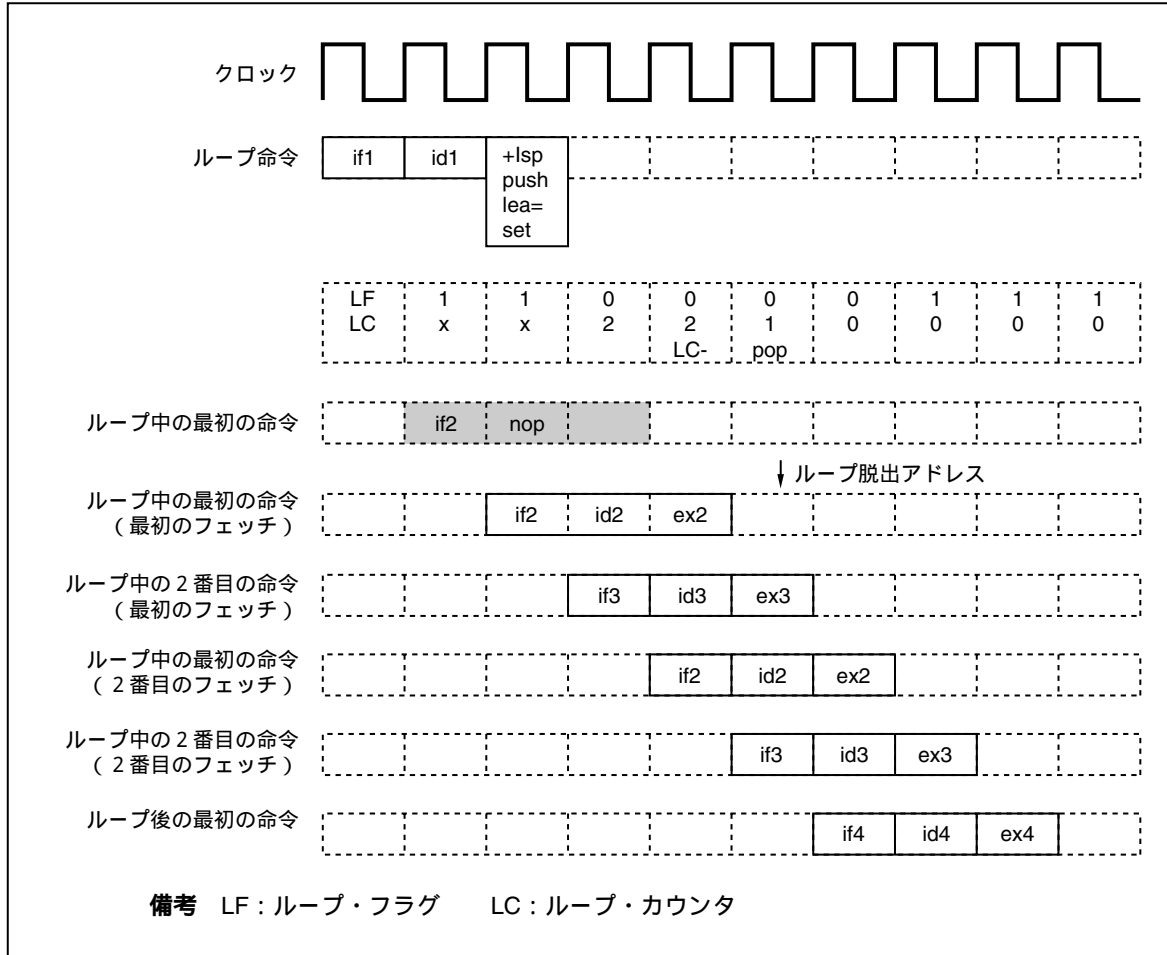
LOOP 命令, LPOP 命令については, μ PD77016 ファミリ ユーザーズ・マニュアル 命令編を参照してください。

注意 LPOP 命令では, ループ脱出のための自動的な PC 管理は行われません。したがって, JMP 命令でループを脱出したあとに, LPOP 命令を実行するか, または LPOP 命令を実行してから JMP 命令でループから抜けてください (μ PD77016 ファミリ ユーザーズ・マニュアル 命令編を参照)。

(d) ループ実行のタイミング (2 回ループ動作例)

LOOP 命令の実行タイミングを、2 命令のグループを 2 回実行するループ動作の例について図 4-16に示します。

図 4-16 ループ実行のタイミング (2 回ループ動作例)



(e) ソフトウェア・ループ・スタック

5 レベル以上のループを行うと、ループ・スタック・オーバーフローが起こります。その結果、戻り番地が失われるので、正常なループ動作が行われなくなります。

あらかじめ5レベル以上のループ処理を行うことがわかっているとき、ループ・スタック(LSTK)がオーバーフローする前に、LSTKの内容をメモリに退避しておくこと、さらにループを行っても正常に動作することができます。これをソフトウェア・ループ・スタックといいます。

ただし、LSTKの内容をメモリに退避したときは、スタックのレベルに対応して保存しておいた内容をLSTKに書き戻さなければなりません。ソフトウェア・ループ・スタックのプログラム例を次に示します。

ソフトウェア・ループ・スタックの例

・プッシュ (DP0: 退避アドレス)

```
R0L = LSR3;  
*DP0-- = R0L;  
R0L = LSR2;  
*DP0-- = R0L;  
R0L = LSR1;  
*DP0-- = R0L;
```

・ポップ (DP0: 復帰アドレス)

```
R0L = *DP0++;  
LSR1 = R0L;  
R0L = *DP0++;  
LSR2 = R0L;  
R0L = *DP0++;  
LSR3 = R0L
```

4.4.4 割り込み

μ PD77210 ファミリには強力な割り込み機能があります。割り込み機能は、DSP コア・カーネル側の機能と、ペリフェラルである割り込みコントローラ側の機能の2種類があります。ここでは、DSP コア・カーネルの側の割り込み機能について説明します。ペリフェラルである割り込みコントローラの機能については、5.10 **割り込みコントローラ (INTC)** を参照してください。

割り込み要因

割り込み制御機能

割り込みの受け付け条件

割り込みベクタ

(1) 割り込み要因

DSP コア・カーネルには、合計 12 個の割り込み要因があります。すべての割り込みは、ペリフェラルである割り込みコントローラを介して DSP コア・カーネルに入力されます。DSP コア・カーネルとしての割り込み要因は、ペリフェラルである割り込みコントローラからの割り込み要求ということになります。

(2) 割り込み制御機能

すべての割り込み要因は独立の事象、独立のレベルとして取り扱われ、それらについての制御機能は次のようにまとめられます。

要因ごとにイネーブル/ディスエーブルを設定できる。

割り込み全体をイネーブル/ディスエーブルできる。

全体イネーブル/ディスエーブル状態についてのスタック機能があり、多重レベル割り込みに対応できる。

割り込みベクタ（割り込みを受け付けた場合の割り込み要因ごとのエントリ・ポイント）は固定されている。

割り込みを受け付けると、実行中の命令は中断されて指定されたエントリ・ポイントにプログラム実行制御を移す。

割り込みプログラムから復帰すると、割り込まれた命令に制御を移す。

ジャンプ命令などの実行中は、割り込み受け付けが遅延される。

(3) 割り込みの受け付け条件

ある割り込み要因について割り込み要求が発生したとき、次の条件がすべて成立しているときにかぎり、その割り込みは受け付けられます。

グローバル割り込み許可フラグ (EI) が 0 (許可状態) である。

対象となる割り込みの要因別許可フラグが 0 (許可状態) である。

ただし、次に示すいずれかの場合には割り込み受け付けが遅延されます。

ジャンプ命令をフェッチ/デコード/実行中

リピート命令およびリピート対象命令をフェッチ/デコード/実行中

ループ命令をフェッチ/デコード/実行中

ループ終端命令をフェッチ中

(4) 割り込みベクタ

すべての割り込み要因には、要因ごとに固定したエントリ・ポイント（ベクタ）があります。各要因に対するベクタは内部命令領域先頭位置（0x200 番地）から順に設定され、64 ワードのテーブルを構成します。各要因にはそれぞれ 4 命令アドレスが割り当てられますが、割り込み処理が、割り込みリターン命令（RETI）を含み 4 命令以内で終了しない場合は、0x240 番地以降に分岐して処理する必要があります。

(a) 割り込みベクタ・テーブル

表 4-13に割り込みベクタ・テーブルを示します。

割り込み要因拡張のために μ PD77210 ファミリでは割り込みペリフェラルを追加しています。詳細は 5.10 割り込みコントローラ（INTC）を参照してください。

表 4-13 割り込みベクタ・テーブル

ベクタ・アドレス	割り込み番号
0x200	リセット
0x204	予約
0x208	
0x20C	
0x210	0
0x214	1
0x218	2
0x21C	3
0x220	4
0x224	5
0x228	6
0x22C	7
0x230	8
0x234	9
0x238	10
0x23C	11

- 注意**
1. リセットは割り込みではありませんが、ベクタのエントリとして統一的に扱われます。
 2. 使用しない割り込み要因のベクタは、異常処理ルーチンへの分岐などの処置をすることをお勧めします。
 3. 各割り込みベクタの先頭アドレスに RETI 命令を記述しないでください。記述した場合に、実際にその割り込みが発生すると μ PD77210 ファミリは正常に動作しません。
 4. マスク ROM 品においてもベクタ領域は内部 RAM 領域に存在します。したがって、この領域のブートアップが必要になります。また、リセット後のエントリが 0x200 番地となるため、内部命令 RAM、割り込みを使用しない場合でも 0x200 番地のブートアップは最低限必要です。

(b) 割り込みベクタの処理例

割り込みベクタの処理例を次に示します。

```

; Definitions
#define SI1 0x3800 ; シリアル入力レジスタのアドレス
#define SO1 0x3800 ; シリアル出力レジスタのアドレス
#define ICR4 0x3884 ; 割り込み番号 4 に対する割り込みコントロール・レジスタ
#define ICR5 0x3885 ; 割り込み番号 5 に対する割り込みコントロール・レジスタ

; Interrupt vector table
int_vec imseg at 0x200 ; ベクタ・テーブル
:
:
org 0x220 ; シリアル入力#1 (TSI) 割り込みベクタ
(0x220) JMP INPUT ; 4 命令以上のためアプリケーション領域に分岐
(0x221) NOP ;
(0x222) NOP ;
(0x223) NOP ;
; シリアル出力#1(TSO)割り込みベクタ
; 4 命令以内の割り込み処理例
(0x224) R0H=*DP4 + + ; Yメモリよりデータをフェッチ
(0x225) *SO1 : y=R0H ; シリアル出力#1 へ転送
(0x226) RETI ; 割り込みからの復帰
(0x227) NOP ;
; シリアル入力#2(ASI)割り込みベクタ
(0x228) NOP ; SI2 は未使用のため (ベクタ先頭へ RETI 命令は記述
(0x229) RETI ; しないでください)
(0x22A) NOP ;
(0x22B) NOP ;
:
:
; Main program segment
main imseg ;
:
:
R0L=SR ;
R0=R0 & 0x7FCF ; 全体および割り込み番号 4, 5 の割り込みを許可
SR=R0L ;
R0L=0x0010 ; ペリフェラルである割り込みコントローラの設定
ICR4:x=R0L ; マスク・モード, TSI 割り込み許可
ICR5:x=R0L ; マスク・モード, TSO 割り込み許可
:
; シリアル入力#1 割り込み処理ルーチン
INPUT : R0H=*SI1 : y ; シリアル入力#1 よりデータをフェッチ
R1=*DP0 ;
R1=R1 + R0H*R2H ;
*DP0=R1H ;
RETI ; 割り込みからの復帰

```


(5) 割り込み制御のソフトウェア

割り込みは次のレジスタによって制御されます (図 4-5 プログラム制御ユニット参照)

- ステータス・レジスタ (SR)
- 割り込み許可フラグ・スタック・レジスタ (EIR)

(a) ステータス・レジスタ (SR : status register)

すべての割り込みの許可 / 禁止および要因ごとの割り込みの許可 / 禁止を示す 16 ビットのレジスタです。レジスタのビットの値が 0 のとき割り込みを許可し, 1 のとき割り込みを禁止します。

SR の値は, レジスタ間転送命令を使ってデータの入出力ができます。

リセットにより SR の値は, 0xF000 になります。

割り込み許可フラグ			注	要因別割り込み許可フラグ (割り込み番号)												
EI	EP	EB	LF	11	10	9	8	7	6	5	4	3	2	1	0	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

注 ループ・フラグ

割り込み許可フラグ (EI : enable interrupt, EP : enable interrupt previous, EB : enable interrupt before) すべての割り込みの許可 / 禁止を示すフラグです。フラグのビットの値が 0 のとき, 割り込みを許可し, 1 のとき割り込みを禁止します。EI (enable interrupt), EP (enable interrupt previous) および EB (enable interrupt before) の 3 ビットで現在, 1 レベル前および 2 レベル前の割り込み許可 / 禁止の状態を示します。

EI, EP, EB フラグは割り込み許可フラグ・スタック・レジスタ (EIR) のビット 15-13 にある EI, EP, EB フラグと同じものです。SR レジスタと EIR レジスタのビット 15-13 は常に同じ値が入ります。

次に示す割り込みのネスティング, スタック操作は SR レジスタの EI, EP, EB フラグおよび割り込み許可フラグ・スタック・レジスタ (EIR) の E3-E15 フラグで扱われます。

- 割り込みが受け付けられた場合 :
 - EB の値 EIR レジスタの E3
 - EP の値 EB ヘシフト
 - EI の値 EP ヘシフト
 - EI 1 にセット (すべての割り込みを禁止)

- RETI 命令が実行された場合 :
 - EI の値 破棄
 - EP の値 EI ヘシフト
 - EB の値 EP ヘシフト
 - EIR レジスタの E3 の値 EB へ

多重割り込みについては, (b) **割り込み許可フラグ・スタック・レジスタ**を参照してください。SR をデスティネーションに指定した転送命令をフェッチしてから実行するまでの期間, すなわち, 転送命令と直後の命令の間および直後の命令とその命令の間は, 更新前の割り込み許可フラグが有効となります。

割り込み許可フラグを変更する例 (許可状態 禁止) を次に示します。

初期状態 : EI = 0 ; (割り込み許可)	
R0L = EIR ;	
R0 = R0 0x8000 ;	
EIR = R0L ;	} 割り込み処理に分岐することがあります。
直後の命令 ;	
その次の命令 ;	

注意 EP, EB フラグの書き換えはすべての割り込みを禁止した状態 (EI = 1)で行ってください。

ループ・フラグ (LF : loop flag)

ループ中かどうかを示すフラグです。ループ・フラグの値が 0 のときはループ中です。1 のときはループ中ではありません。

注意 割り込み許可フラグおよび要因別割り込み許可フラグをモディファイするとき、このフラグを変更しないでください。これらのフラグをモディファイする場合は、常に現在の SR の値を読み込んで、指定されたフラグだけをマスクするようにしてください (次の例を参照)。

予備フラグ

書き込み時は無視、読み出し時は不定です。

要因別割り込み許可フラグ

要因ごとの割り込みの許可 / 禁止を示すフラグです。フラグのビットの値が 0 のとき割り込みを許可し、1 のとき割り込みを禁止します。割り込みを受け付けても値は変化しません。

注意 要因別割り込み許可フラグの書き換えはすべての割り込みを禁止した状態 (EI = 1)で行ってください。

要因別割り込み許可フラグを変更する例 (許可状態 禁止) を次に示します。

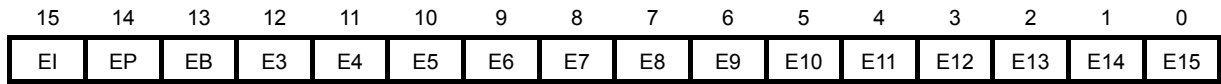
R0L = EIR ;	EIR を用いてすべての割り込みをディスエーブルにする
R0 = R0 0x8000 ;	EI = 1
EIR = R0L ;	EIR に書き戻す
NOP ;	EIR にセットした値が有効になるまで待つ
R0L = SR ;	SR を用いて割り込み番号 0 の割り込みをディスエーブルにする
R0 = R0 0x0001 ;	(割り込み番号 0) = 1
SR = R0L ;	SR に書き戻す

(b) 割り込み許可フラグ・スタック・レジスタ (EIR : enable interrupt stack register)

割り込み許可フラグをスタックする 16 ビットのレジスタです。レジスタのビットが 0 のとき、割り込みは許可されます。1 のとき、割り込み禁止を示します。

EIR の値は、レジスタ間転送命令を使ってデータの入出力ができます。

リセットにより、EIR の値は 0xFFFF になります。



割り込みを受け付けると、1 ビット右シフトしたあと EI を 1 にし、すべての割り込みを禁止します。レジスタの値は割り込みリターン命令により 1 ビット左シフトされ、同時に E15 に 1 をセットします。こうして、16 レベルまでの多重割り込みの復帰を保証します。

ビット 15-13 (EI, EP, EB) は、SR のビット 15-13 と同じです。

レジスタ間転送命令で EIR に書き込みを行い、割り込みの許可または禁止状態を変更することができます。このとき、変更が有効になるのは 3 命令後からです。

割り込みを許可する例 (禁止状態 → 許可) を次に示します。

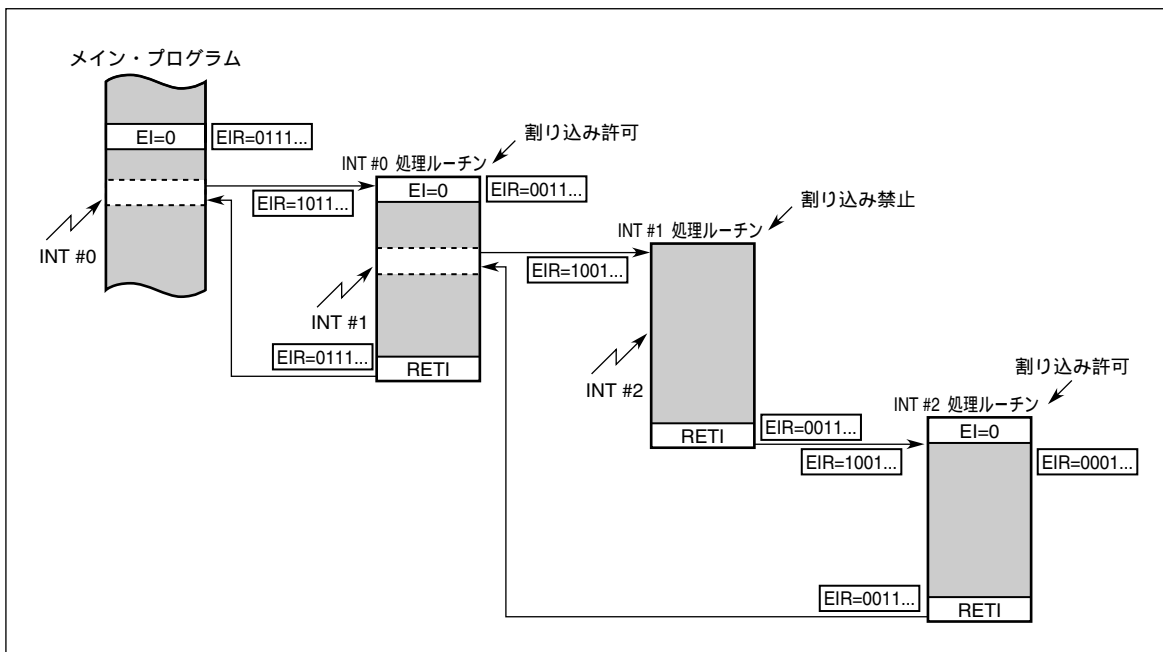
```

初期状態 : EI = 1 (割り込み禁止)
ROL = EIR      ;
R0 = R0&0x7FFF ;
EIR = R0L     ;
命令 1        ; } この間は割り込み禁止
命令 2        ; }
命令 3        ; } 割り込み許可
    
```

(c) EIR と多重割り込み

割り込み許可フラグ・スタック・レジスタ (EIR) を利用して多重割り込みシステムを構成することができます。図 4-17 に多重割り込みの例を示し、EIR に注目した多重割り込みの概念を説明します (すべての要因別割り込み許可フラグ (SR のビット 9-0) はイネーブル状態とします)。

図 4-17 多重割り込みの概念



EI に 0 をセットし、割り込みイネーブルとする。

割り込み番号 0 の割り込みが受け付けられ、割り込み番号 0 の処理ルーチンに制御が移行する。このとき EIR は右に 1 ビット・シフトされ、割り込み状態は 1 レベル分スタックされる。同時にビット 15 (EI) には 1 がセットされ、割り込み禁止となる。

割り込み番号 0 の処理ルーチン中で、EI = 0 (イネーブル) とする。

割り込み番号 1 の割り込みが受け付けられ、割り込み番号 1 の処理ルーチンに制御が移行する。前と同様に EIR は右に 1 ビット・シフトされ、EI は 1 にセットされる。この状態で割り込みは禁止である。

割り込み番号 1 の処理ルーチン中で割り込み番号 2 の割り込み要求が発生したが、ディスエーブル状態なのでこの割り込みは記憶される。

割り込み番号 1 の処理ルーチンを RETI 命令で終了すると、EIR は左に 1 ビット・シフトされ、割り込み番号 1 の割り込み前の状態が回復する。この状態は EI = 0 で割り込み許可である。

ここでただちに、先ほどの割り込み番号 2 の割り込み記憶が有効化され、割り込み番号 2 の処理ルーチンに制御が移行する。同様の手順で EIR は右に 1 ビット・シフトされ、EI は 1 にセットされる。必要であれば EI = 0 とする。

割り込み番号 2 処理ルーチンを RETI 命令で終了すると、EIR は左に 1 ビット・シフトされ、割り込み番号 2 の割り込み前の状態が回復する (割り込み番号 0 の割り込み処理中)。

割り込み番号 0 の処理ルーチンを続行し、最後に RETI 命令で処理すると割り込み番号 0 の割り込みを受け付ける前の状態に回復する。

(d) SR と EIR の使い分け

SR と EIR の最上位 3 ビット (EI, EP, EB) は共通のビットとしてアクセスされます。特に EI ビットは現在の割り込み許可状態に直接かかっていますから状態変更には注意してください。EI ビットを取り扱うときの SR と EIR の違いは次のように考えられます。

禁止 許可への変更は, SR, EIR のどちらを使用してもかまいません。

許可 禁止への変更は, EIR レジスタの使用を推奨します。

禁止 許可の場合では, いずれにしても変更は禁止状態で推移しますから問題は起こりません。許可禁止のケースでは, 次のような状態が考えられます。

```

R0L = SR                ; SR を用いてすべての割り込みをディスエーブルにする

                        ; ここでいずれかの割り込みが発生し, 割り込み処理に移行する

                        ; 割り込み処理ルーチン
                        ; このルーチンでは要因別に使用禁止設定を行う
R1L=SR                ;
R1=R1 | 0x0001        ; 割り込み番号 0 の割り込みを禁止
SR=R1L                ; SR へ書き戻す
RETI                  ; 割り込みからの復帰

                        ; SR の値が変更されている

R0=R0 | 0x8000        ; EI=1
SR=R0L                ; SR へ書き戻す
:
:
    
```

この場合, 割り込み中での SR の書き込みは無視されます。このような危険を避けるために許可禁止では, EIR の使用を推奨します。

(6) 割り込みシーケンス

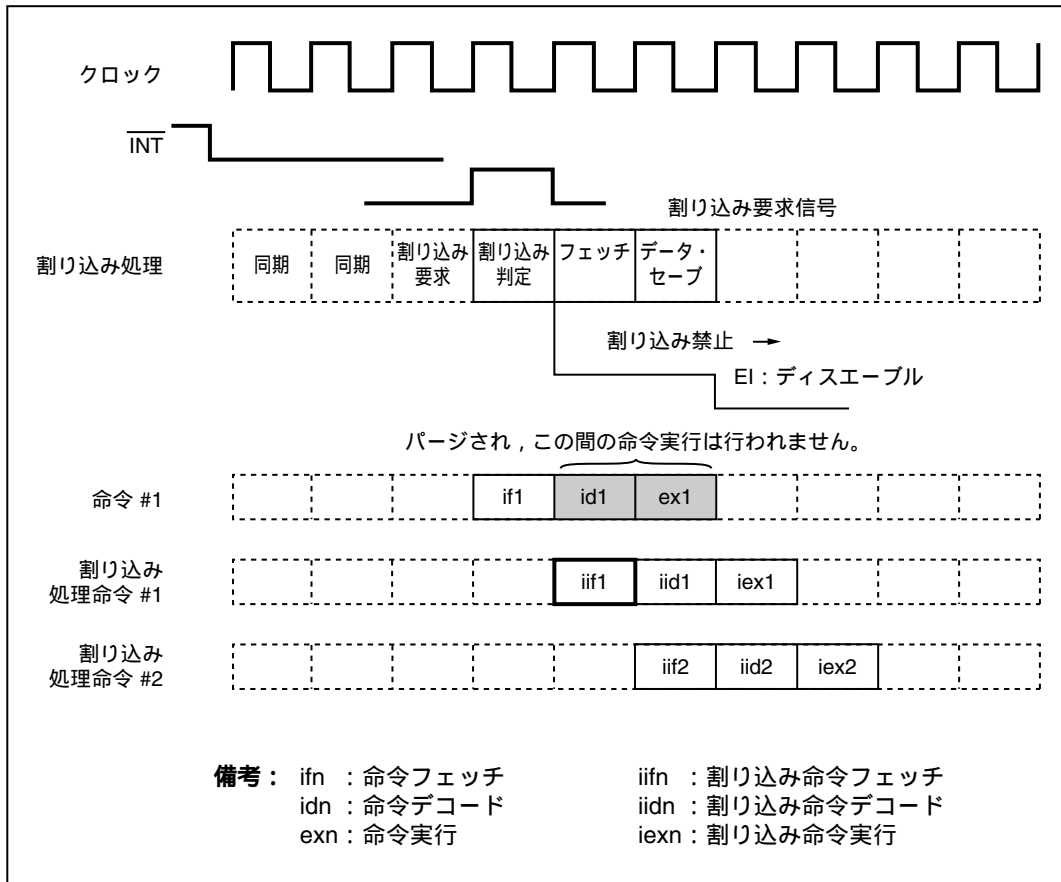
(a) 割り込みの受け付け

割り込みが受け付けられたとき, 次のように動作します。

- 割り込み受け付け直前にフェッチした命令を保留する。
- EIR を右に 1 ビット・シフトして 1 レベル・スタックする。
- EI = 1 として割り込み禁止とする。
- SP をインクリメントする。
- 保留した命令のアドレスを SP の指す STK にセーブする。
- PC に所定の割り込みベクタ・アドレスを設定し, 割り込み処理に分岐する。

図 4-18に割り込み受け付けのタイミングを示します。

図 4-18 割り込み受け付けタイミング



(b) 割り込みからの復帰

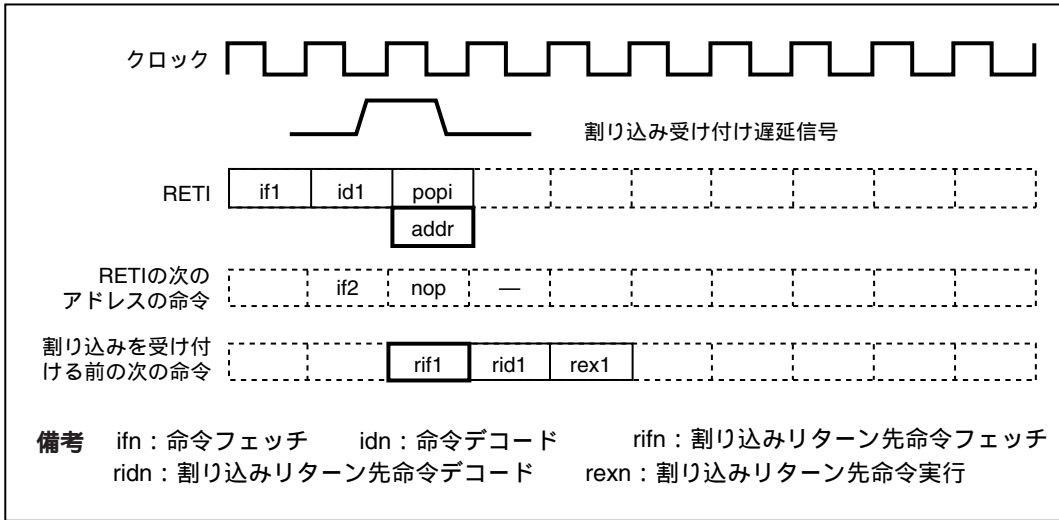
RETI (割り込みリターン) 命令により、次の処理を2または3インストラクション・サイクルで実行して割り込みから復帰します。

- SPの指すSTKの値をPCにリストアする。
- SPをデクリメントする。
- EIRレジスタを左シフトして割り込み許可フラグを回復する。
- 戻り番地(割り込み受け付け時に保留した命令)に分岐する。

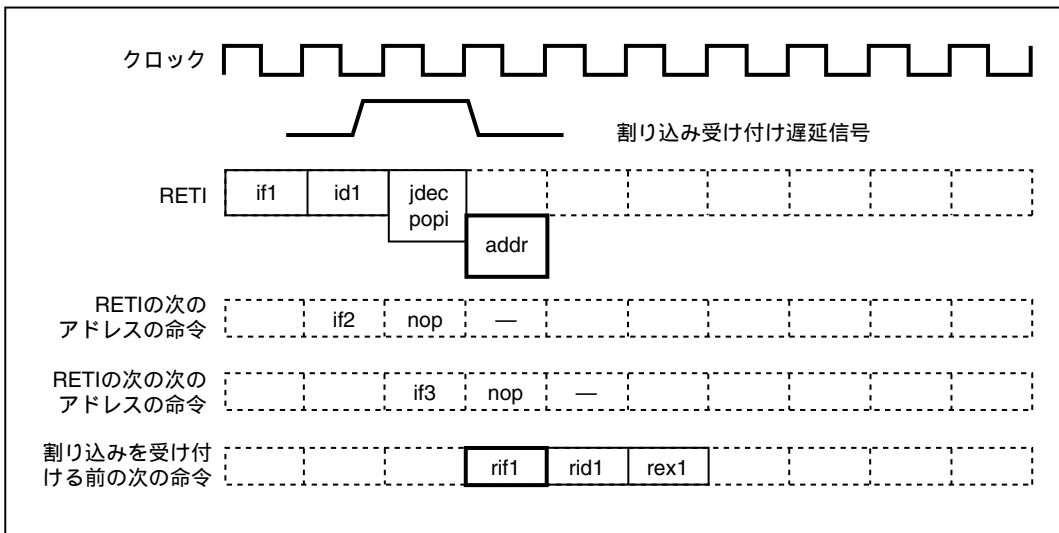
図 4-19(a)に無条件 RETI 命令によってリターンするタイミングを、図 4-19(b)に条件付き RETI 命令で、条件成立によってリターンするタイミングをそれぞれ示します。

図 4-19 RETI 命令によるタイミング

(a) 無条件



(b) 条件付き：条件成立



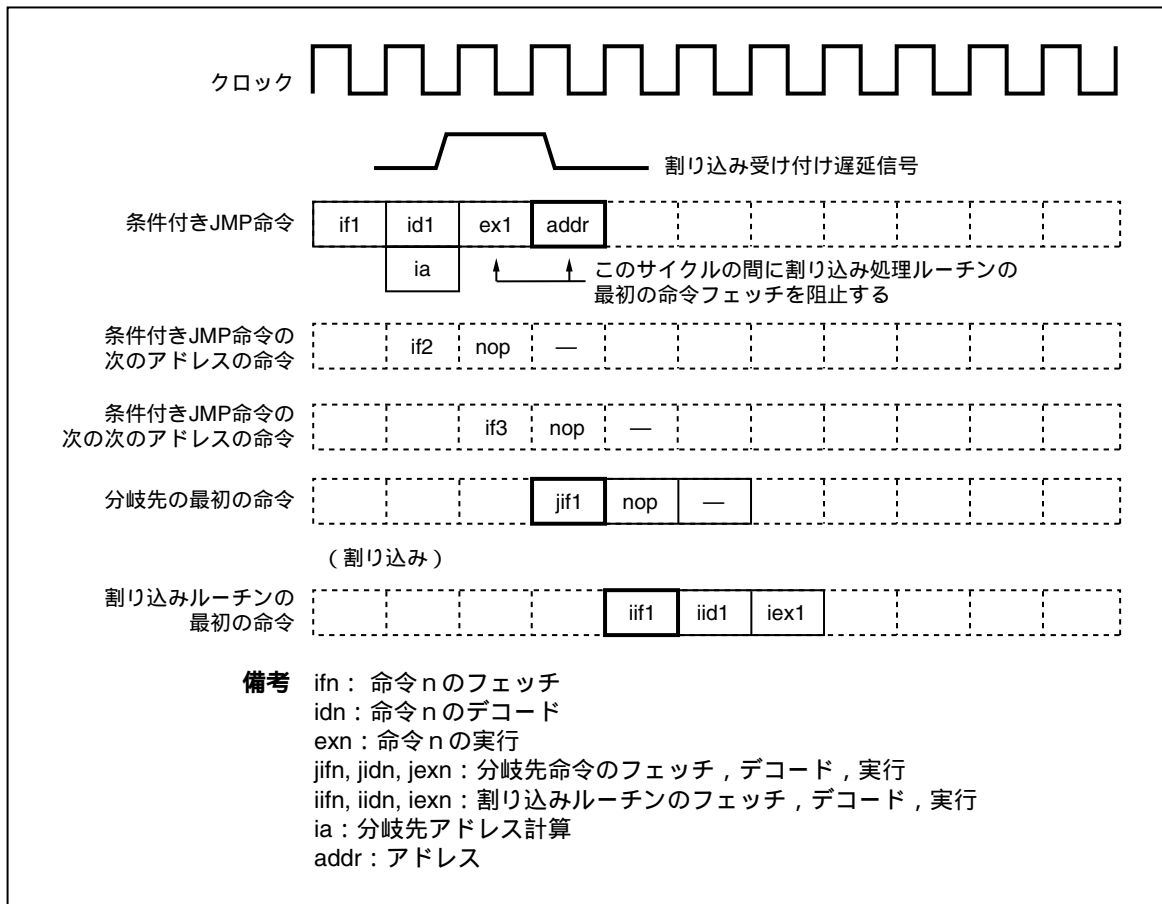
(b) 2 インストラクション・サイクルの遅延を生ずる命令

次に示す命令は、2 インストラクション・サイクルの割り込み遅延を生じます。

- 条件付き JMP 命令（即値データによる PC 相対ジャンプ）のデコード
- 条件付き CALL 命令（即値データによる PC 相対ジャンプ）のデコード
- 条件付き RET 命令のデコード
- 条件付き RETI 命令のデコード
- 無条件 / 条件付きレジスタ間接 JMP 命令のデコード
- 無条件 / 条件付きレジスタ間接 CALL 命令のデコード
- REP 命令のデコード
- LOOP 命令のデコード

これらの命令実行中に発生した割り込みが遅延される様子を図 4-21 に示します。

図 4-21 割り込み遅延タイミング（2 サイクル遅延）



(8) 割り込みの競合と記憶**(a) 割り込みの記憶**

ある要因の割り込みが受け付けられると、自動的に EI = 1 (グローバル割り込み許可フラグがディスエーブル) の状態で割り込み処理プログラムが実行されます。その状態でさらに別の割り込みが発生した場合、新たな割り込みがその時点で受け付けられることはありませんが、発生した割り込みに対して要因別に割り込みを記憶します。実行中の割り込み処理を終了して RETI (割り込みリターン) 命令を実行すると EI が 0 (イネーブル) に回復し、この時点で先ほど記憶されていた要因が有効化されて割り込み受け付けとして処理されます。この割り込み記憶は、EI = 1 の状態ばかりでなく、要因別の割り込みがディスエーブル状態でも同様に機能します。

- 注意 1.** 割り込みの状態がイネーブル、ディスエーブルにかかわらず、すべての割り込みが記憶されます。
- 2.** 割り込み記憶は 1 要因について 1 レベルのみです。
- 3.** 割り込みが発生したことを記憶するフラグは、該当する割り込みが受け付けられるまで保持されます。
- 4.** FINT(フォークット・インタラプト)命令によってすべての割り込みを破棄することができます。詳細については、*μPD77016 ファミリ ユーザーズ・マニュアル 命令編*を参照してください。

(b) 割り込みの優先順位

割り込みに優先順位はありません。同時に 2 つ以上の割り込みが競合した場合、どの割り込みが最初に実行されるかは不定です。

★ (9) 割り込みの制限事項

ノイズなどにより外部割り込み端子にデータ・シートに明記されている $\overline{\text{INTmn}}$ ロウ・レベル幅の MIN スペック (6 ttc ns) を満たさない信号が入力された場合でも、割り込みと認識する場合があります。

AC 特性の割り込みタイミング必要条件に明記されているスペックを満たすようにしてください。

4.4.5 エラー・ステータス・レジスタ (ESR : error status register)

μ PD77210 ファミリのエラー状態を示す 16 ビットのレジスタです。ビット 15-4 は、書き込み時は無視され、読み出し時は不定となります。

ESR のビット 3-0 の値は、エラーが発生したとき 1 になります。ハードウェア・リセットまたはレジスタ間転送命令で 0 を設定するまで、値は変化しません。

ESR の値は、レジスタ間転送命令を使ってデータの入出力ができます。

リセットにより、ESR の値は 0 になります。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	-	ovf	ste	lse	-

ovf : オーバフロー・エラー・フラグ

演算ユニットで、演算結果が 40 ビットの 2 の補数表現のデータ・フォーマットでオーバフローしたとき 1 になります。

ste : スタック・エラー・フラグ

スタックがオーバフローおよびアンダフローしたとき 1 になります。

lse : ループ・スタック・エラー・フラグ

ループ・スタックがオーバフローおよびアンダフローしたとき 1 になります。

注意 μ PD7701 x ファミリでは ESR のビット 0 は bac(バス・アクセス・エラー・フラグ)となっていますが、 μ PD77210 ファミリに本フラグは存在しません。したがって禁止しているメモリ領域の組み合わせでアクセスしたときのエラーを検出することはできません。

4.5 データ・アドレッシング・ユニット

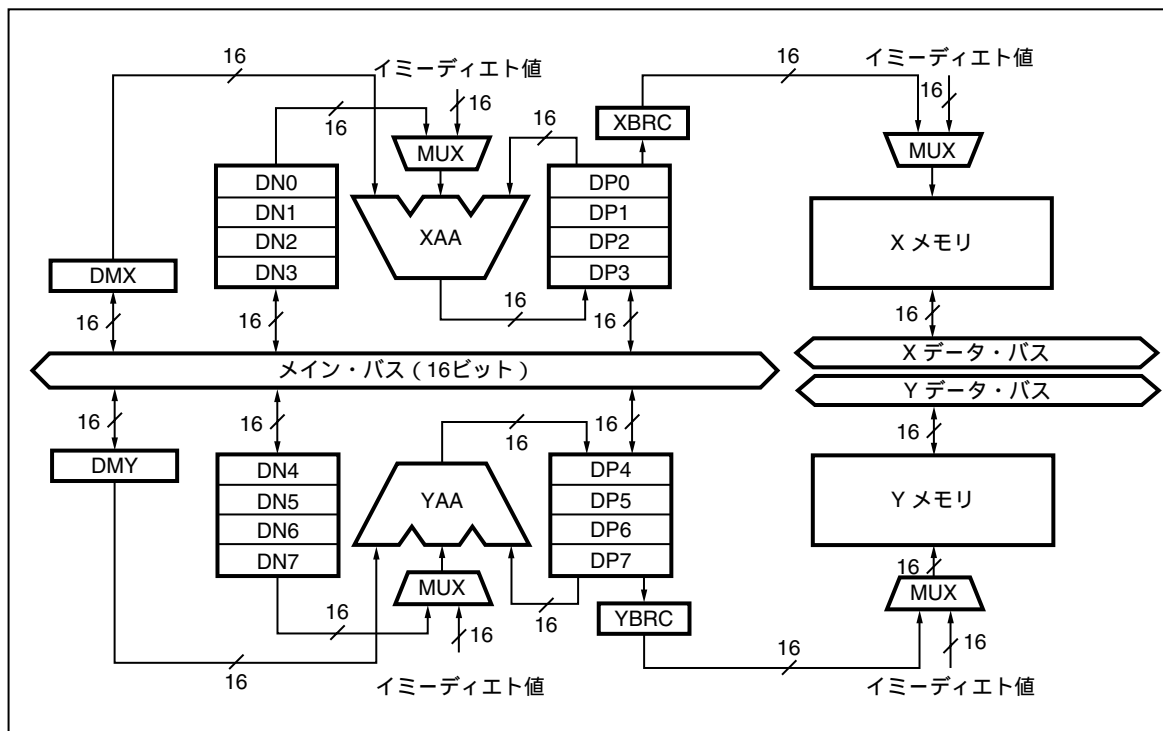
一般に DSP では、大量のデータを柔軟に効率よくアクセスすることを求められます。

μ PD77210 ファミリは、データ・メモリ空間に対するこれらの要求を満たすために、専用のデータ・アドレッシング・ユニットを備えています。

4.5.1 ブロック構成

図 4-22にデータ・アドレッシング・ユニットのブロック図を示します。

図 4-22 データ・アドレッシング・ユニット



4.5.2 データ・メモリ空間

μ PD77210 ファミリは独立した対等の2つのデータ・メモリ空間(X, Yメモリ空間)を持ち、柔軟なデータ・アクセスを可能にしています。これらのメモリ空間はそれぞれが、内部メモリ(デバイス内部のリソースとして常に高速なアクセスが保証されます。しかも、X, Yメモリ空間への同時アクセスが可能です)と外部メモリ(ソフトウェア・ウエイトおよびハードウェア・ウエイト機能を利用し、多様なスピード・レンジのメモリを接続できます)に区画されています。また内部メモリには、ROM領域およびRAM領域が設定されています。

ここでは、メモリ空間について説明します。

(1) Xメモリ空間とYメモリ空間

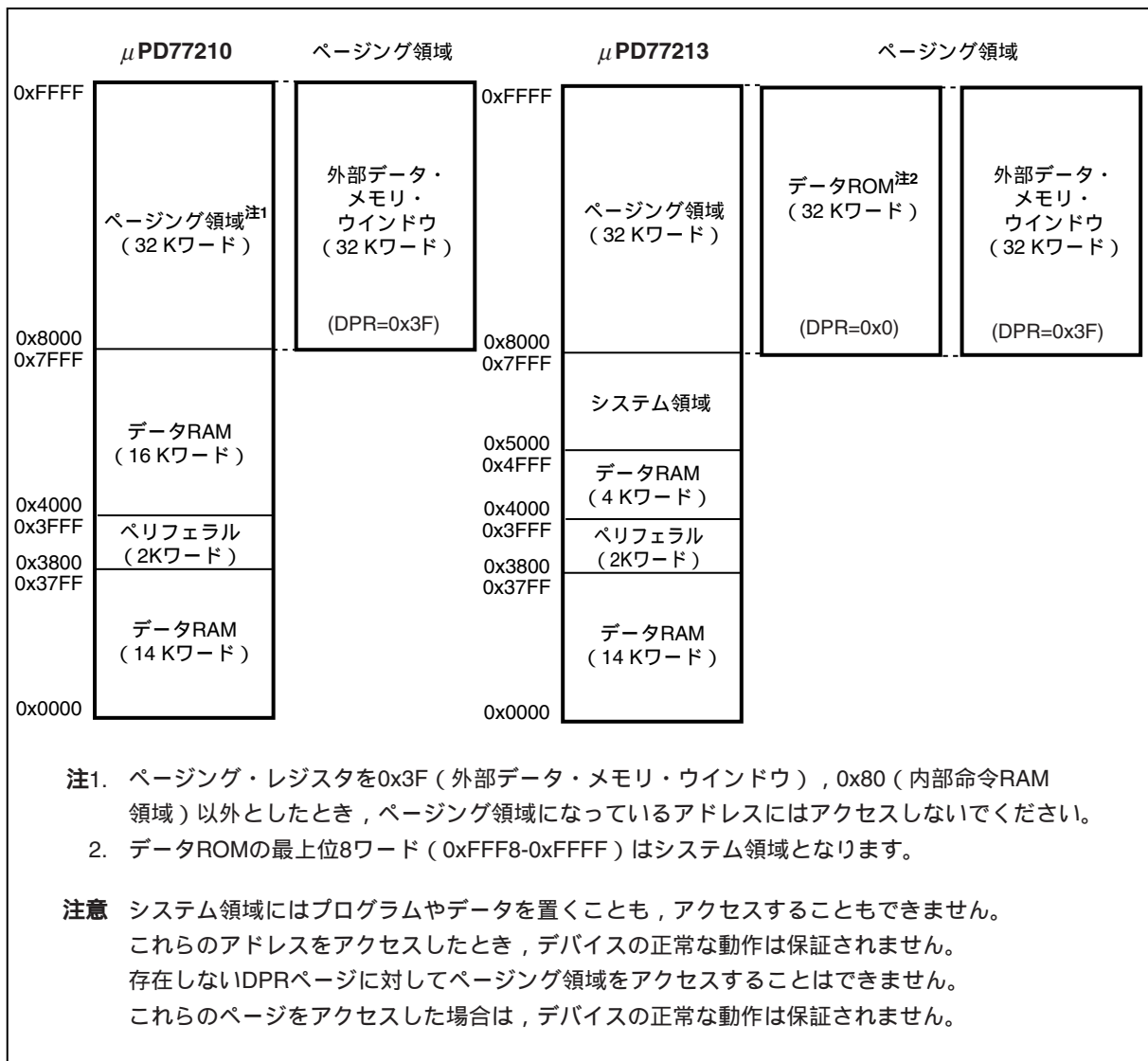
μPD77210ファミリは、独立した対等の2つのデータ・メモリ空間、Xメモリ空間およびYメモリ空間を持っています。これらは、Xデータ・バス、Yデータ・バスを経由してそれぞれアクセスされるメモリ空間です(4.2.2 データ・バス参照)。メモリ空間全体の特長は次のとおりです。

1ワードを16ビットで構成

Xメモリ、Yメモリそれぞれについて64Kワードの空間

Xメモリ空間とYメモリ空間とは互いに同等なメモリ・マップ構成になっていますが、μPD77210ファミリの製品間では違いがあります。図4-23に、各製品ごとのデータ・メモリ・マップを示します。

図4-23 データ・メモリ・マップ



(2) 内部データ・メモリ

図 4-23に示されるように、 μ PD77210 は 0 番地から始まる 32K ワードの領域、 μ PD77213 は 0 番地から始まる 18K ワードの領域が、デバイス内部にマッピングされた内部領域として機能します (0x3800 番地から 0x3FFF 番地はペリフェラル領域です)。内部領域はさらに ROM 領域、RAM 領域、ペリフェラル領域およびシステム領域に分かれており、このうちデータ・メモリ部分は ROM と RAM です。これら内部データ・メモリの容量は製品ごとに異なっていますので、アプリケーションに最適なものを選択することができます。

ペリフェラル領域についての詳細は 5.2 ペリフェラル・レジスタを参照してください。

注意 システム領域へのアクセスは禁止されています。

(a) 内部 ROM と内部 RAM

上に述べたように、内部領域の ROM および RAM は製品ごとに容量が異なります。ここでは、それら容量のバリエーションを次に示します。

表 4-14 ROM と RAM の容量

製品名	ROM		RAM	
	X	Y	X	Y
μ PD77210	なし	なし	30 K ワード	30 K ワード
μ PD77213	32 K ワード	32 K ワード	18 K ワード	18 K ワード

(3) 外部データ・メモリ・インタフェース

μ PD77210 ファミリでは外部メモリ・インタフェースをペリフェラルとして取り扱います。詳細は、5.7 外部データ・メモリ・インタフェース (MIO) を参照してください。

(a) 外部データ・メモリ容量

図 4-23に示したように、 μ PD77210 ファミリは外部データ・メモリを拡張できます。また、拡張可能なメモリ容量は製品によって異なり、表 4-15にそれらの違いを示します。

表 4-15 外部データ・メモリの容量

製品名	メモリ容量
μ PD77210	1 M ワード
μ PD77213	1M ワード (SD カード・インタフェース使用時は 8 K ワード)

(4) 同時アクセスの制限

μ PD77210 ファミリは並列ロードなど X, Y メモリ空間に同時に 2 つのオブジェクトにアクセスする機能を持っていますが、次のような制限があります。

- ペリフェラル (0x3800-0x3FFF) に対する並列アクセスはできない。
- DPR=0x3F とした 0x8000-0xFFFF への外部メモリ・アクセスは、MIO ペリフェラルを介してアクセスするためペリフェラルとの並列アクセスはできない。
- DPR は X, Y とも共通で用いるため、異なるページに対するアクセスはできない。

表 4-16に同時アクセスの可否を示す。

表 4-16 X, Y メモリ空間への同時アクセスの可否

X メモリ \ Y メモリ		0x8000-0xFFFF						
		0x0000-0x37FF	0x3800-0x3FFF	0x4000-0x7FFF	DPR=0x0-0x3E	DPR=0x3F	DPR=0x80	DPR=0xC0-0xFF
0x0000-0x37FF		OK	-	OK	OK	OK	OK	OK
0x3800-0x3FFF		-	-	-	-	-	-	-
0x4000-0x7FFF		OK	-	OK	OK	OK	OK	OK
0x8000-0xFFFF	DPR=0x0-0x3E	OK	-	OK	OK	-	-	-
	DPR=0x3F	OK	-	OK	-	-	-	-
	DPR=0x80	OK	-	OK	-	-	OK	-
	DPR=0xC0-0xFF	OK	-	OK	-	-	-	OK

また、命令メモリに対してデータとしてアクセスを行う場合は次のような制限があります。

- 0x0000-0x7FFF の領域にある命令において、DPR = 0x80 としたアクセスは、アクセス競合が生じるため 1 ウェイトが挿入される。
- 0x8000-0xFFFF の領域にある命令において、DPR = 0xC0-0xFF としたアクセスは、アクセス競合が生じるため 1 ウェイトが挿入される。

4.5.3 命令メモリのエイリアス

データ・ページ・メモリには命令メモリにエイリアスされているページがあり、このページをデータ・ページ・レジスタにより指定することで、命令メモリをデータとしてアクセスすることができます。

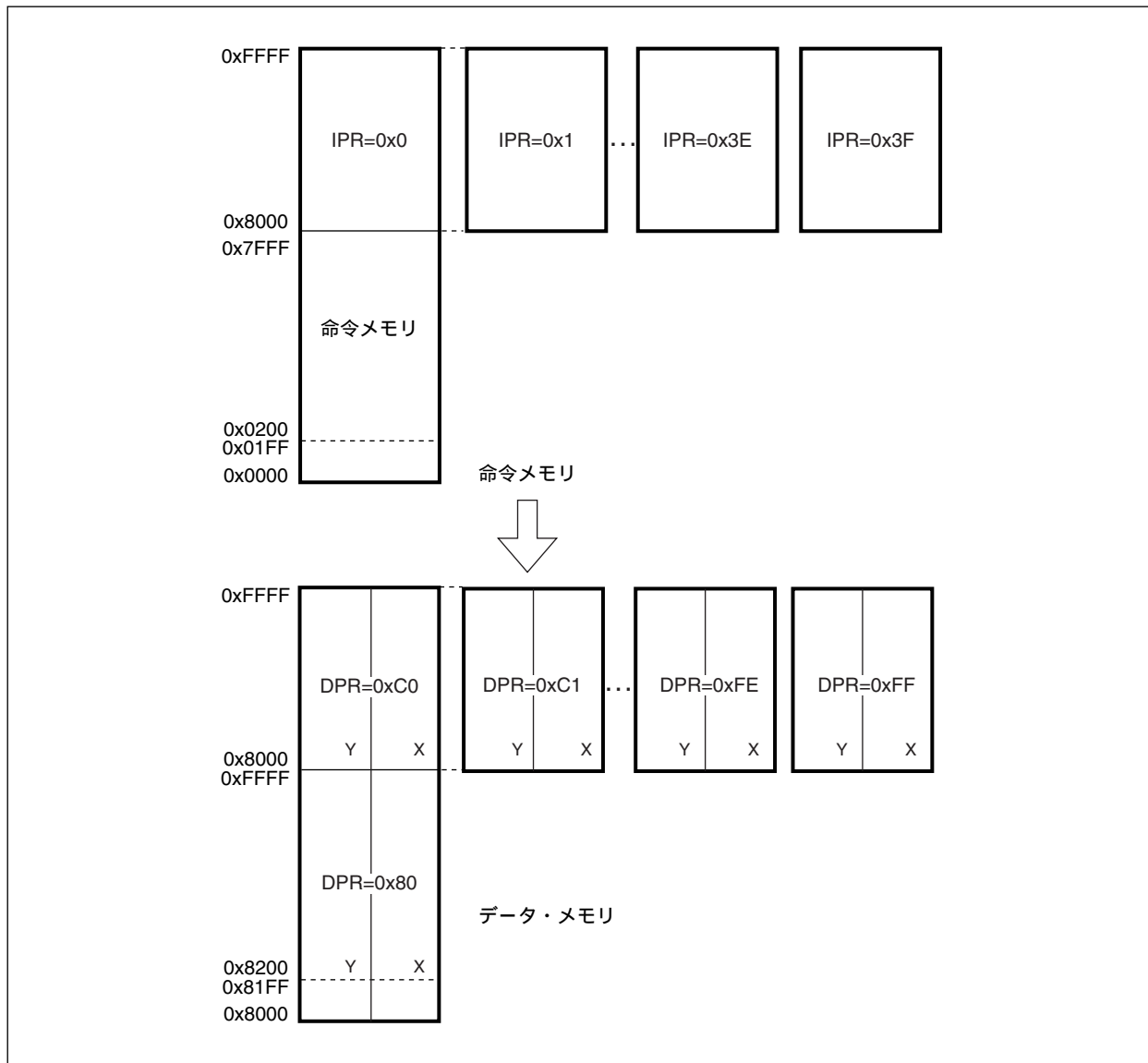
図 4-24にエイリアスのイメージを示します。命令メモリは32ビット幅、データ・メモリは16ビット幅であるため、命令メモリ下位16ビットをXメモリ、上位16ビットをYメモリとしてエイリアスします。

ページ・メモリからアクセスするため、命令メモリの下位32Kワード空間(0x0000-0x7FFF)に対するアクセスは0x8000のオフセットが必要になります。

命令メモリをデータとしてアクセスする場合、アクセス命令と同一のメモリ領域(たとえば、0x0000-0x7FFFの命令領域上の命令からDPR=0x80のアクセスなど)へのアクセスでは競合が発生することがあります。この場合は、1ウエイトを挿入し、命令フェッチ後にデータ・アクセスを行います。

注意 μ PD77210 ファミリには命令例外割り込み機能が存在しないため、データとして書き込みを行った値を命令としてフェッチした場合、それが正しくない命令であっても検出できません。その結果、プログラムが暴走する可能性があるため注意してください。

図 4-24 命令メモリのデータ・メモリへのエイリアス



4.5.4 外部データ・メモリ・マップ

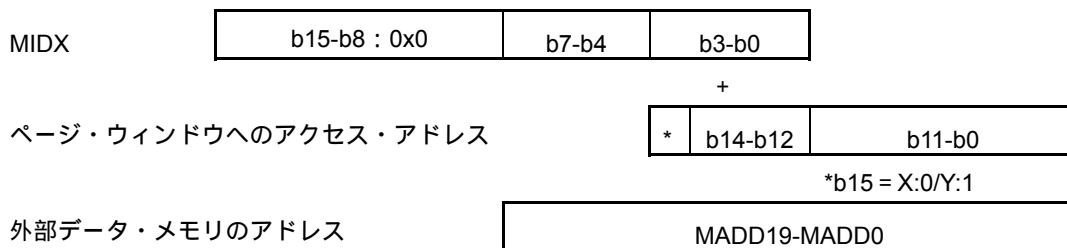
外部データ・メモリ空間として 1Mワード×16ビットをサポートしています。

アクセス方法としては、次の2種類があります。

- (1) DPR = 0x3F としたページ・メモリをウインドウとしてアクセス
- (2) MIO のデータ・レジスタ (MDT) を介してアクセス

(2) については、5.7 外部データ・メモリ・インタフェース (MIO) を参照してください。ここでは (1) について補足します。

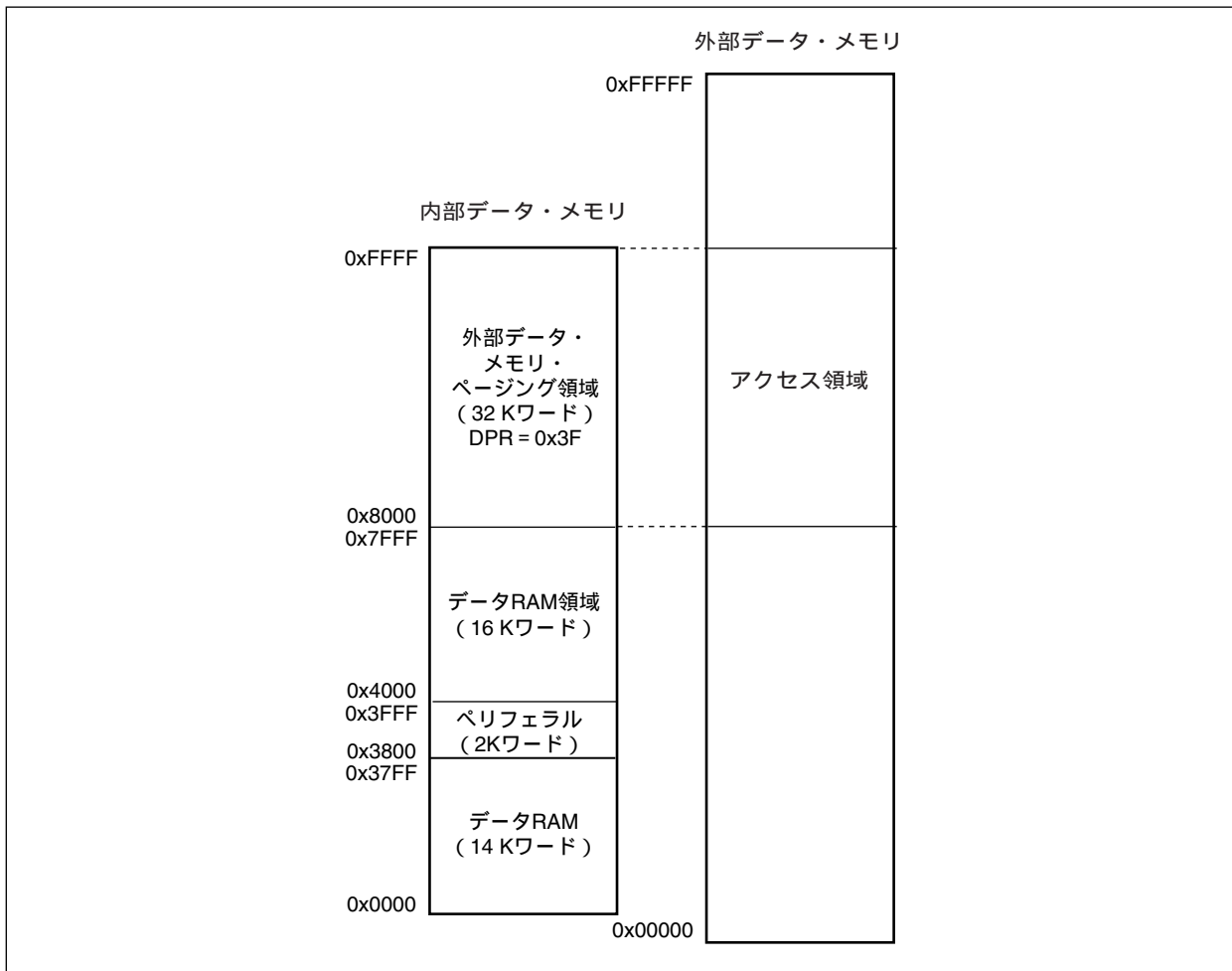
図 4-25に外部データ・メモリを、ページ・メモリをウインドウとしてアクセスするイメージを示します。ページ・メモリをウインドウとしてアクセスする場合、アドレッシングはアクセスするページ・メモリのアドレスと MIO インデクス・レジスタ (MIDX) とを加算した 20 ビット・アドレスとしてアクセスします。ただし、X メモリに対するアクセスの場合は、ページ・メモリへのアクセスを、アドレスの MSB を 0 として、Y メモリに対するアクセスの場合はページ・メモリへのアクセスをアドレスの MSB を 1 として、加算を行います。このイメージを次に示します。



たとえば、DPR = 0x3F において MIDX = 0x11 とした場合は、次のようになります。

- 0x8000:X へのアクセス → 外部メモリ 0x11000 へのアクセス
- 0x8000:Y へのアクセス → 外部メモリ 0x19000 へのアクセス

図 4-25 外部データ・メモリ・マップのアクセス・イメージ



4.5.5 アドレッシング・モード

μ PD77210 ファミリは、高速かつ柔軟なデータ・メモリ・アクセスを実現するためにさまざまなメカニズムを備えています。X メモリ空間と Y メモリ空間は完全に独立してアドレスされていますが、アドレッシング・ユニットは機能的には同一です。ここでは、それらのメカニズムとそれによって実現されるアドレッシング・モードを説明します。

(1) アドレッシング・ユニットの各部の機能

アドレッシング・ユニットのレジスタとブロック (図 4-22 データ・アドレッシング・ユニット参照) の機能は次のとおりです。

(a) データ・ポインタ (DP0-DP7)

間接アドレッシングに使用する 8 つの 16 ビット・レジスタです。X メモリ空間のアドレスを DP0-DP3 で、Y メモリ空間のアドレスを DP4-DP7 で指定します。

DP0-DP7 の値は、メイン・バスで入出力できます。

(b) インデクス・レジスタ (DN0-DN7)

DP0-DP7 をモディファイするための 8 つの 16 ビット・レジスタです。メモリ・アクセス後、DNn の値で DPn (添え字が同一) をモディファイします。DN0-DN7 の値は、メイン・バスで入出力できます。

このレジスタに設定できる値は - 32768 (0x8000) ~ + 32767 (0x7FFF) です。

(c) モジュロ・レジスタ (DMX, DMY)

DP0-DP7 をリング・カウンタ動作させながらモディファイするときに、リング・カウンタの範囲を指定するための2つの16ビット・レジスタです。

DP0-DP3 の範囲指定を DMX で、DP4-DP7 の範囲指定を DMY で行います。

DMX および DMY の値は、メイン・バスで入出力できます。

このレジスタに設定できる値は +1 (0x0001) ~ +32767 (0x7FFF) です。

(d) アドレス ALU (XAA, YAA : X, Y Address ALU)

DP0-DP7 をモディファイするための2つの16ビット ALU です。

DP0-DP3 のモディファイを XAA で、DP4-DP7 のモディファイを YAA で行います。

(e) ビット・リバース回路 (XBRC, YBRC : X, Y Bit Reverse Circuit)

ビット・リバース・アクセスの場合、DP0-DP7 の値の上位と下位を反転したアドレスを出力します。

最上位ビットの値が最下位ビットに、最下位ビットの値が最上位ビットになるよう反転します。

(f) マルチプレクサ (MUX : MUltipleXer)

複数の信号から1本を選択して出力する回路です。

(2) アドレッシング・モードの種類

データ・メモリ・アドレッシング・モードを階層的に分類すると次のようになります。

直接アドレッシングは1種類、間接アドレッシングは DP をアドレッシングの中心にした7種類から構成されません。

直接アドレッシング

間接アドレッシング

* DPn (ノー・チェンジ)

* DPn++ (ポスト・インクリメント)

* DPn-- (ポスト・デクリメント)

* DPn## (ポスト・インデクス加算)

* DPn%% (ポスト・モジュロ・インデクス加算)

* !DPn## (プレ・ビット・リバース・アンド・ポスト・インデクス加算)

* DPn##imm (イミディエイト加算)

(a) 直接アドレッシング

直接アドレッシングは、命令ワードの中で直接にアドレス値とアドレス区分(X/Y)が表現されています。指定された区分(X/Y)の指定されたアドレスと汎用レジスタとの間で、X/Y データ・バスを經由して16ビット・データを交換します。

命令ワードについては、**μ PD77016 ファミリ ユーザーズ・マニュアル 命令編**を参照してください。

記述例1：ロード

```
R0H = *0x1234:X;
```

汎用レジスタ R0 の H パート (中間 16 ビット) に、X メモリの 0x1234 番地から 16 ビット・データをロードします。

記述例2：ストア

```
*0x1234:X=R0H;
```

汎用レジスタ R0 の H パート (中間 16 ビット) から、X メモリの 0x1234 番地へ 16 ビット・データをストアします。

注意 直接アドレッシングでは、X、Y メモリ空間へ同時にアクセスすることはできません。

(b) 間接アドレッシング

間接アドレッシングは、すべて DPn レジスタ (データ・ポインタ) を使用したアドレッシングで、基本特性を次のようにまとめることができます。

アドレス値は、ビット・リバース・インデクス加算を除くすべてのモードで指定された DPn の現在値が出力される。ビット・リバース・インデクス加算モードでは、指定された DPn の現在値をビット・リバース (図 4-26 参照) した値が出力される。

DPn のモディファイが指示されている場合、データ・メモリ・アクセスが終了したあとで DPn がモディファイされる。

モディファイされた DPn の値、すなわち新しいアドレスは次の命令から有効になる。

DPn のモディファイを単独で実行することはできない。

レジスタ間転送命令または即値設定命令によって DPn にイミューディエト値が設定された場合、新しいアドレスは 2 命令後から有効になる (**μ PD77016 ファミリ ユーザーズ・マニュアル 命令編** 参照)。

X メモリ空間に対しては DP0-DP3 を、Y メモリ空間に対しては DP4-DP7 を、それぞれ使用する。

次に、個別の間接アドレッシング・モードについて説明します。

*DPn (ノー・チェンジ)

DPn の値でメモリをアクセスし、アクセス終了後 DPn の値は保存されます。

記述例：

```
R1L=*DP0;
```

R1 の L パート (下位 16 ビット) に、DP0 の値の示す X メモリのアドレスから 16 ビット・データをロードします。

*DPn++ (ポスト・インクリメント)

DPn の値でメモリをアクセスし、アクセス終了後 DPn をインクリメント (+1) します。

記述例：

```
R2H=*DP4++;
```

R2 の H パート (中間 16 ビット) に、DP4 の値の示す Y メモリのアドレスから 16 ビット・データをロードし、その後 DP4 をインクリメントします。

*DPn-- (ポスト・デクリメント)

DPn の値でメモリをアクセスし、アクセス終了後 DPn をデクリメント (-1) します。

記述例：

```
R3E=*DP1--;
```

R3 の E パート (上位 8 ビット) に、DP1 の値の示す X メモリのアドレスから 8 ビット・データ (16 ビット中の下位 8 ビット) をロードし、その後 DP1 をデクリメントします。

*DPn## (ポスト・インデクス加算)

DPn の値でメモリをアクセスし、アクセス終了後 DPn に DNn の値を加算します。

インデクス・レジスタ DNn は同じ添え字のデータ・ポインタ DPn に対応しています。

たとえば DN1 に対応するのは DP1 です。

DNn に設定できる値は - 32768 (0x8000) ~ + 32767 (0x7FFF) です。

記述例：

```
R4L=*DP5##;
```

R4 の L パート (下位 16 ビット) に、DP5 の値の示す Y メモリのアドレスから 16 ビット・データをロードし、その後 DP5 に DN5 の値を加算します。

*DPn%% (ポスト・モジュロ・インデクス加算)

DPn の値でメモリをアクセスし、アクセス終了後 DPn に DNn の値を加算し、さらに DMX/DMY (n = 0-3 のとき DMX, n = 4-7 のとき DMY を選択) でモジュロ調整されます。

インデクス・レジスタ DNn は同じ添え字のデータ・ポインタ DPn に対応しています。

たとえば DN1 に対応するのは DP1 です。

DNn に設定できる値は - 32768 (0x8000) ~ + 32767 (0x7FFF) です。

モジュロ・インデクス加算とモジュロ調整については、**モジュロ・インデクス加算と巡回バッファ**を参照してください。

記述例：

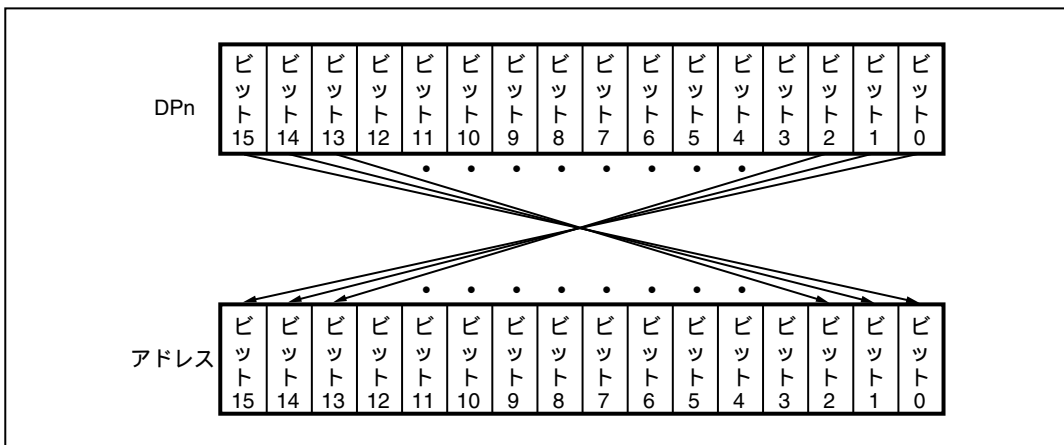
```
R5H=*DP3%%;
```

R5 の H パート (中間 16 ビット) に、DP3 の値の示す X メモリのアドレスから 16 ビット・データをロードし、その後 DP3 に DN3 の値を加算して、さらに DMX によりモジュロ調整します。

*!DP## (プレ・ビット・リバース・アンド・ポスト・インデクス加算)

DPn の上位-下位ビットを反転した値 (図 4-26 参照) でメモリをアクセスし、アクセス終了後 DPn に DNn の値を加算します。このとき、加算される DPn の値はビット反転されないことと、DP と DN とで n の値が等しいことに注意してください。この機能は FFT などのアプリケーションに使用します。

図 4-26 DPn のビット反転



記述例：

```
R6H=*DP6##;
```

R6 の H パート (中間 16 ビット) に、DP6 をビット反転した値の示す Y メモリのアドレスから 16 ビット・データをロードし、そのあと DP6 に DN 6 の値を加算します。

備考 アドレスのビット反転アクセスによって DPn はモディファイされず、ビット反転されたアドレスは DPn にフィードバックされません。ビット反転アクセス後、ビット反転する前の DP6 の値 (もとの DP6 の値) に対して DNn の値を加算します。

*DPn##imm (ポスト・イミディエト加算)

DPn の値でメモリをアクセスし、アクセス終了後 DPn に imm として表されるイミディエト値を加算します。

イミディエト値に設定できる値は - 32768 (0x8000) ~ + 32767 (0x7FFF) です。

記述例：

```
R7L=*DP2##100;
```

R7 の L パート (下位 16 ビット) に、DP2 の値の示す X メモリのアドレスから 16 ビット・データをロードし、その後 DP2 に 100 を加算します。

注意 イミディエト加算アドレッシングでは、X、Y メモリ空間へ同時にアクセスすることはできません。

データ・ポインタのモディファイー一覧

前述のアドレッシング・モードについてのデータ・ポインタ・モディファイを次の表にまとめます。

表 4-17 データ・ポインタのモディファイー一覧

(a) オペレーション

記述例	オペレーション
DPn	何もしません (DPn の値を変化させません)
DPn + +	DPn DPn + 1
DPn - -	DPn DPn - 1
DPn##	DPn DPn + DNn (DP0-DP7 に対応する DN0-DN7 の値を加算します。) 例 DP0 DP0 + DN0
DPn%%	(n=0-3) DPn = ((DPL + DNn) mod (DMX + 1)) + DPn
	(n=4-7) DPn = ((DPL + DNn) mod (DMY + 1)) + DPn
!DPn##	DPn をビット・リバース後メモリ・アクセスする。 メモリ・アクセス後 DPn DPn + DNn
DPn##imm	DPn DPn + imm

(b) 設定できる値

レジスタ名	16 進数	10 進数
DPn	0x0000-0xFFFF	0 ~ +65535
DNn	0x8000-0x7FFF	- 32768 ~ + 32767
DMX / DMY	0x0001-0x7FFF	+ 1 ~ + 32767
イミディエト値	0x8000-0x7FFF	- 32768 ~ + 32767

モジュロ・インデクス加算と巡回バッファ

モジュロ・インデクス加算モードは、巡回バッファ（サイクリック・バッファ、またはリング・バッファとも呼びます）を実現するための機構です。

・演算規約

DPn でメモリをアクセスしたあと、DPn はモディファイされますが、そのとき次の規約にしたがって演算されます。

(1) $DP_L = DP_L + DN_n$ を計算する。

(2) その結果、 $DP_L \geq DMA$ であれば、 $DP_n = DP_L + DP_H$ を結果とする。

そうでなければ ($DP_L < DMA$ のとき)、

$DP_n = (DP_L + DN_n) \bmod (DMA + 1) + DP_H$ を結果とする。

ここで、

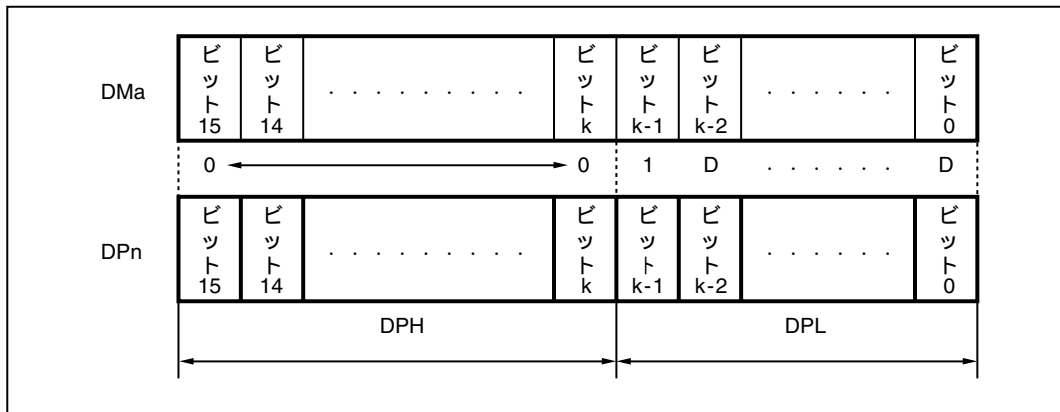
DP_H : DMA の値が $[2k, 2(k - 1)]$ の範囲にある場合、DPn 初期値の下位 k ビットを 0 にした値 (図 4-27 参照)

DP_L : 上記の場合で、DPn の下位 k ビットの値 (図 4-27 参照)

DMA : 指定された DPn において、DMX または DMY のうち対応するもの

備考 (2) の過程をモジュロ調整と呼びます。

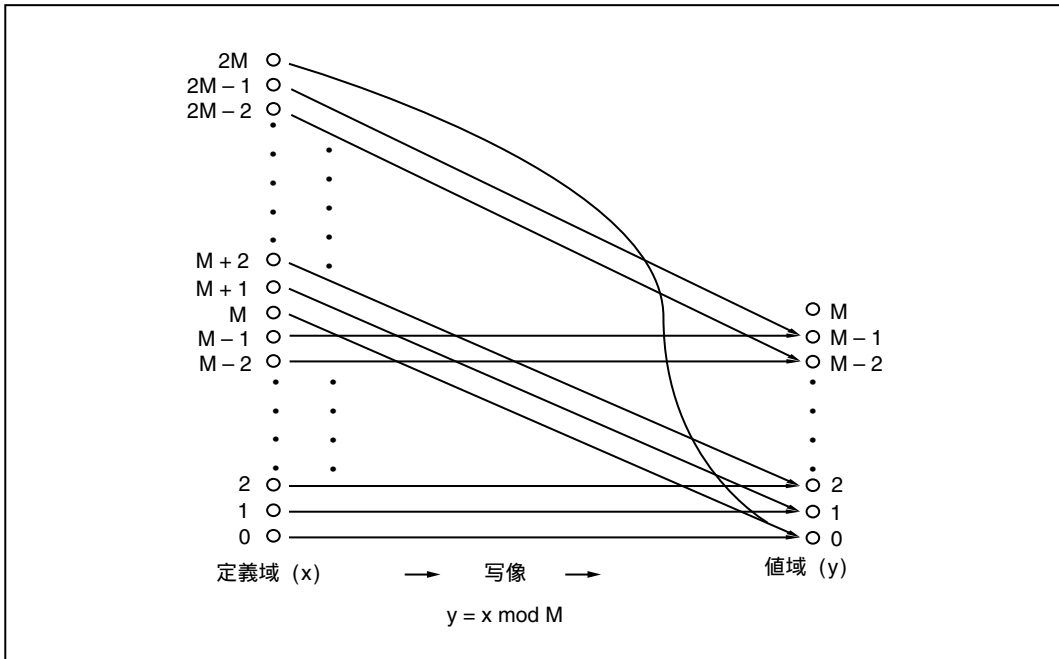
図 4-27 DPn の分割



・意味

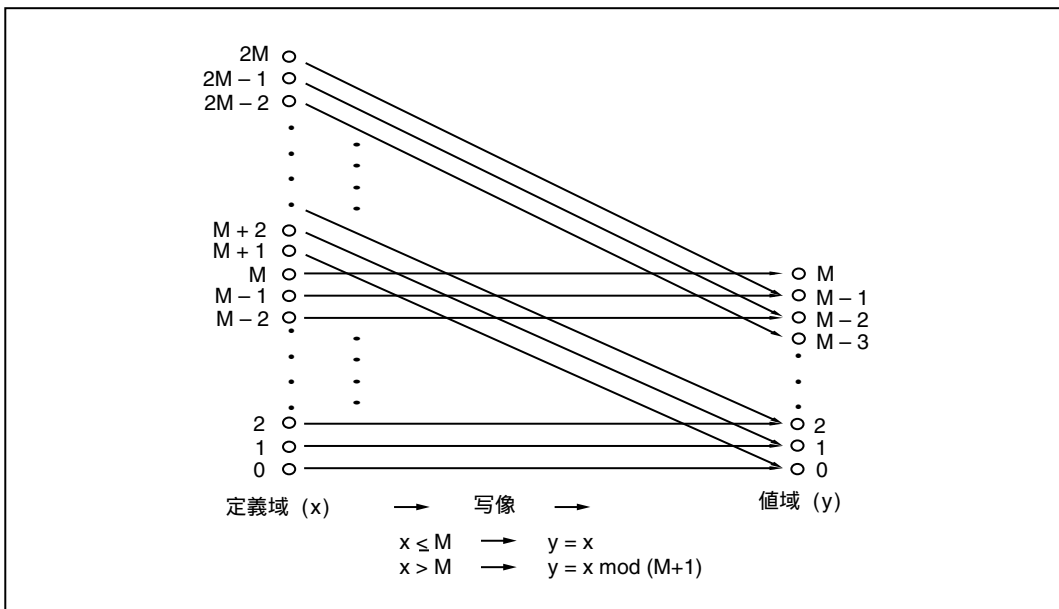
まず、通常のもジュロ演算は図 4-28に表される写像と考えることができます。

図 4-28 通常のもジュロ演算の写像



それに対して上に示したもジュロ調整は、図 4-29に表される写像と見るすることができます。

図 4-29 もジュロ調整の写像



ここで両者の相違を値域(この場合の値域はバッファの大きさに対応しています)の相違に求めると、 DMa への設定値を M とすれば、結局、図 4-28で利用可能なバッファ・サイズは M であるのに対し、図 4-29では $M+1$ になります、したがって、後述するように DMa の設定可能最大値が $0x7FFF$ であるにもかかわらず、最大バッファ・サイズとして $0x8000$ を利用できることになります。

・モジュロ・インデクス加算の例

モジュロ・インデクス加算を使用して、巡回バッファを構成した例の動作過程を次に示します。

例 1 :

DMX=0x7;

DN0=1;

DP0=0x0;

このとき、モジュロ・インデクス加算により DP0 の値を次のように更新します。

DP0=0x0

↓ 0x0+1

DP0=0x1

↓ 0x1+1

DP0=0x2

↓ 0x2+1

DP0=0x3

↓ 0x3+1

DP0=0x4

↓ 0x4+1

DP0=0x5

↓ 0x5+1

DP0=0x6

↓ 0x6+1

DP0=0x7

↓ $0x7+1=0x8 \rightarrow 0x8-(0x7+1)=0x0$

DP0=0x0

↓ 0x0+1

DP0=0x1

↓ 0x1+1

DP0=0x2

↓ 0x2+1

DP0=0x3

↓ 0x3+1

DP0=0x4

:

例 2 :

DMX=0xA;

DN0=3;

DP0=0x10;

このとき、モジュロ・インデクス加算により DP0 の値を次のように更新します。

DP0=0x10

↓ 0x10+3

DP0=0x13

↓ 0x13+3

DP0=0x16

↓ 0x16+3

DP0=0x19

↓ $0x19+3=0x1C \rightarrow 0x1C-(0xA+1)=0x11$

DP0=0x11

↓ 0x11+3

DP0=0x14

↓ 0x14+3

DP0=0x17

↓ 0x17+3

DP0=0x1A

↓ $0x1A+3=0x1D \rightarrow 0x1D-(0xA+1)=0x12$

DP0=0x12

:

4.6 演算ユニット

このユニットで、汎用レジスタは、すべてのオペランドのソース、および数値演算、論理演算結果のデスティネーションとなります。汎用レジスタは次のものと接続しています。

メイン・バス（レジスタ間転送命令実行時）

XY データ・バス（データ・メモリとペリフェラル・レジスタのデータ転送時）

このユニットで実行される数値演算 / 論理演算の種類は次のとおりです。

三項演算命令：3 オペランド入力が必要とするすべての操作

例： MADD : $R0 = R0 + R1H * R2H$

二項演算命令：2 オペランド入力が必要とするすべての操作

例： ADD : $R0 = R2 + R3$

単項演算命令：1 オペランド入力が必要とするすべての操作

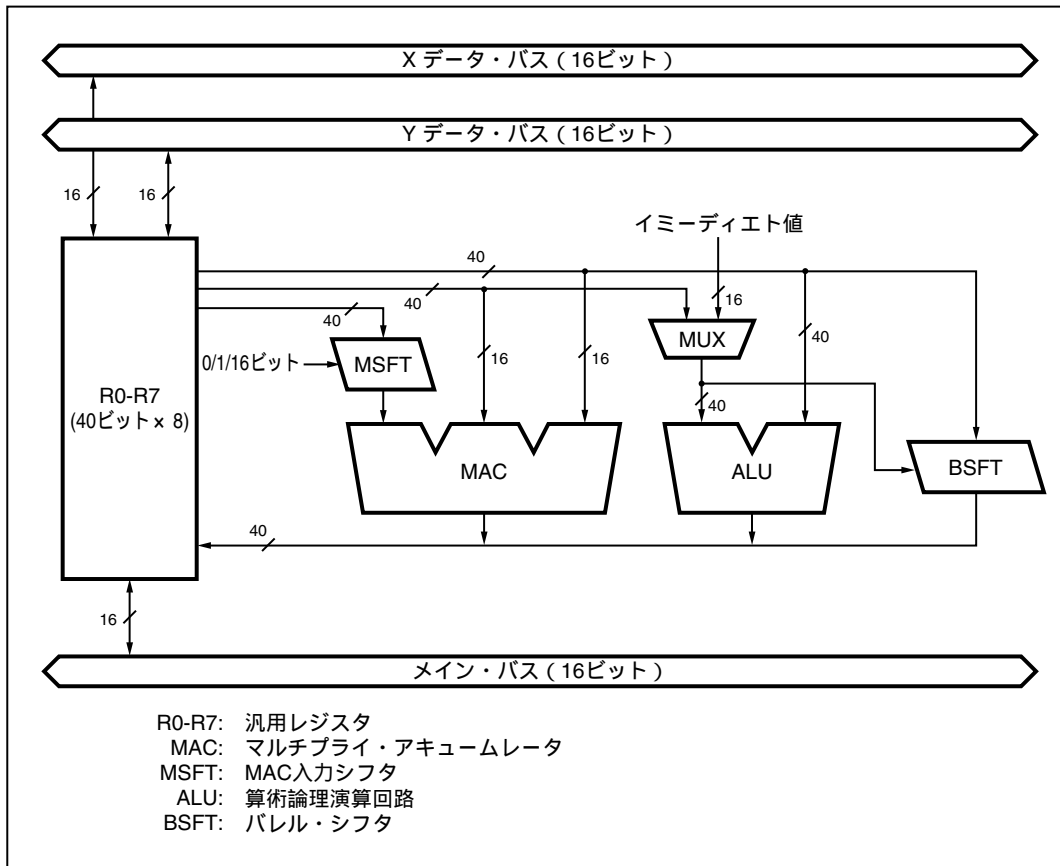
例： NEG : $R0 = -R1$

ここでは、4.6.1 **ブロック構成**でこのユニットの詳細なブロック図を示し、次に汎用レジスタ（R0-R7）とデータ・フォーマット、累乗算器（MAC）と MAC 入力シフタ（MSFT）の演算機能、ALU と平行・シフタの演算機能の順にそれぞれ詳しく説明します。

4.6.1 ブロック構成

図 4-30に演算ユニットの詳細ブロックを示します。

図 4-30 演算ユニットの詳細ブロック



4.6.2 汎用レジスタとデータ・フォーマット

汎用レジスタの特長は次のとおりです。

40 ビットのレジスタです。

トータルで 8 個 (R0-R7) あります。

演算命令の入出力パラメータとして機能します (演算命令では、即値データのほか汎用レジスタのみがパラメータとして記述可能です)。

X/Y データ・メモリとペリフェラル・レジスタの間でデータ転送を行います (ロード/ストア機能)。

ほかのレジスタとの間でデータ転送を行います。

(1) 汎用レジスタの分割フォーマット

汎用レジスタは全 40 ビットで構成されていますが、転送、ロード/ストア、演算などでは、このうちの特定の部分を対象とすることがあります。要素的に分割した場合には、お互いに排他的な 3 つの部分で構成されます。

Lパート：ビット 15-0（下位 16 ビット）

Hパート：ビット 31-16（中間 16 ビット）

Eパート：ビット 39-32（上位 8 ビット）

演算対象となる汎用レジスタのパートは数値演算 / 論理演算の種類によって異なります（表 4-18参照）。

図 4-31に R0HL-R7HL を除いた 5 種類のフォーマットを示します。

表 4-18 汎用レジスタのフォーマット

	R0-R7 40 ビット	R0L-R7L 16 ビット	R0H-R7H 16 ビット	R0E-R7E 8 ビット	R0HL-R7HL 32 ビット	R0EH-R7EH 24 ビット
MAC 累乗算				-	-	-
MAC 乗算		-		-	-	-
ALU		-	-	-		
BSFT			-	-	-	-
X/Y バス転送					-	
レジスタ間転送	-		-	-	-	-

図 4-31 汎用レジスタの分割フォーマット

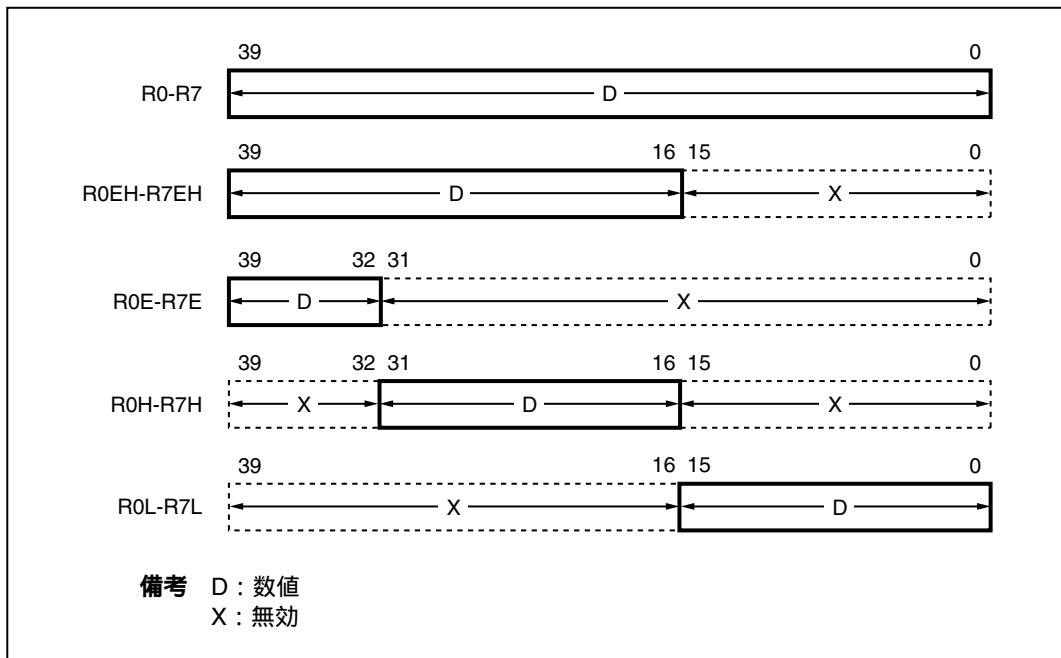
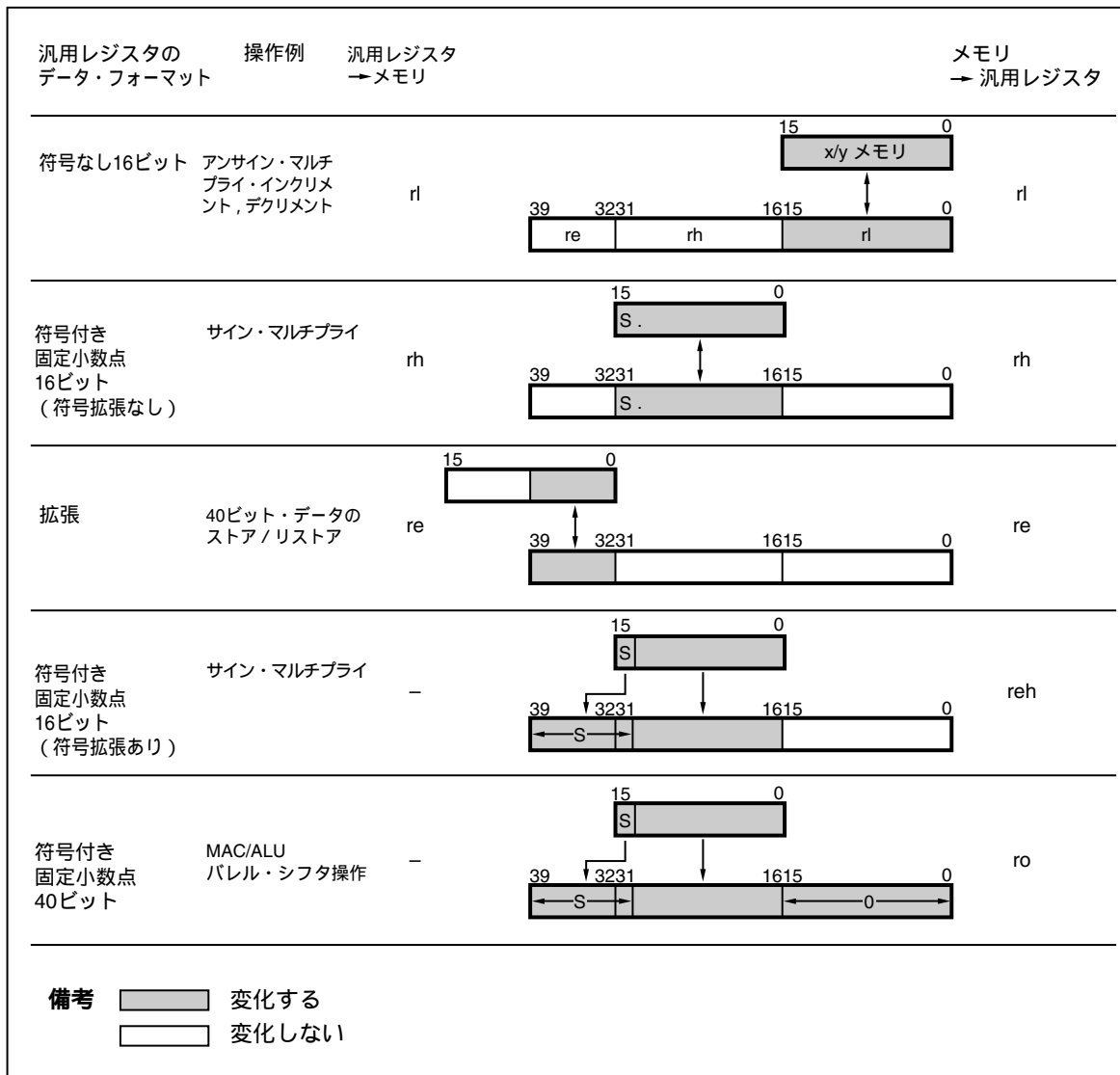


図 4-32に汎用レジスタとデータ・メモリ間のデータ転送を示します。

図 4-32 汎用レジスタとデータ・メモリ間のデータ転送



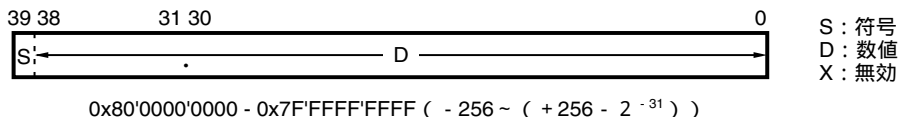
(2) 数値フォーマット

μPD77210 ファミリの汎用レジスタでは、固定小数点データと整数データを取り扱うことができます。ただし、固定小数点データを演算の主体と考えたアーキテクチャになっています。

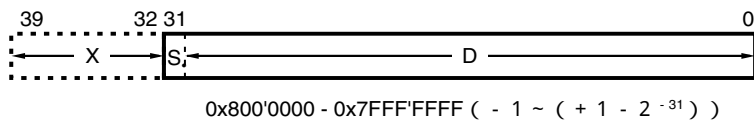
(a) 固定小数点フォーマット

固定小数点データ・フォーマットは、ビット 31 とビット 30 の間が小数点位置となるデータ・フォーマットです。40 ビット表現、32 ビット表現および 16 ビット表現の 3 種類があります。

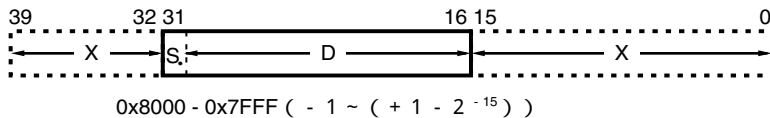
40 ビット・データ・フォーマット (加減算などの入力)



32 ビット・データ・フォーマット (指数命令の入力)



16 ビット・データ・フォーマット (乗算命令の入力)

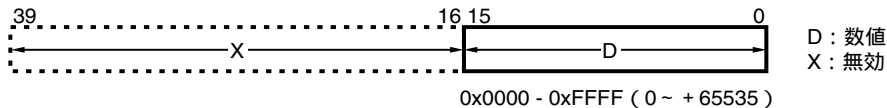


備考 32 ビット固定小数点フォーマットおよび 16 ビット固定小数点フォーマットで、データの絶対値は 1 を越えることはありません。したがって、それらのフォーマットをパラメータとした累算を汎用レジスタ上で実行するかぎりでは、E パートはオーバーフロー吸収領域 (ヘッド・ルーム) として機能します。これは 1 累算で 1LSB (E パート換算) のオーバーフローの可能性のあるものと仮定しても、256 回の連続累算に対してオーバーフロー判定を省略することができます。

(b) 整数フォーマット

整数フォーマットを次に示します。

16 ビット・データ・フォーマット (乗算命令, シフト命令の入力)



4.6.3 累乗算器 (MAC) と MAC 入力シフタ (MSFT) の演算機能

累乗算器には、次の機能があります。

乗算

MPY : $ro = rh * rh'$

乗算とその結果を 40 ビットに拡張し、指定された汎用レジスタへの累加算

MADD : $ro = ro + rh * rh'$ (signed-signed multiply)

MSUB : $ro = ro - rh * rh'$ (signed-signed multiply)

SUMA : $ro = ro + rh * rl$ (signed-unsigned multiply)

UUMA : $ro = ro + rl * rl'$ (unsigned-unsigned multiply)

乗算とその結果を 40 ビットに拡張し、指定された汎用レジスタを 1 ビットまたは 16 ビット右シフトした結果への累加算

MAS1 : $ro = (ro \gg 1) + rh * rh'$

MAS16 : $ro = (ro \gg 16) + rh * rh'$

注意 MAC, ALU, BSFT は同時に動作することはできません。

(1) 乗算機能

乗算機能はマルチプライ・アキュムレータ (MAC : Multiply Accumulator) によって実行されます。乗算機能には取り扱うデータ型によって、次の 3 種類があります。

有符号値同士の乗算

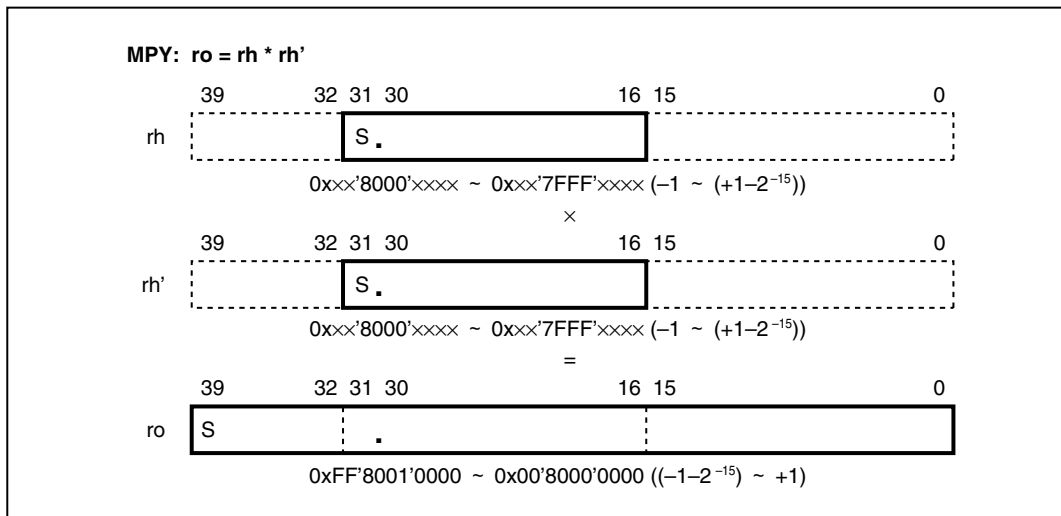
有符号値と無符号値の乗算

無符号値同士の乗算

(a) 有符号値同士の乗算

2 つのパラメータはともに 16 ビット固定小数点型です。つまりデータは汎用レジスタの H パートに設定され、ビット 31 は符号を表しています。この演算過程のイメージを、図 4-33 に示します。

図 4-33 有符号値同士の乗算



- 備考 1. $0x8000(-1)$ 同士の乗算を行うと、結果は $0x00'8000'0000$ となりますが、32 ビット固定小数点フォーマットの範囲では +1 の表現ができないため、オーバーフローになります (拡張ビット $re = 0x00$ はこの場合 32 ビット・フォーマットの符号ビットとは異なります)。ただしこの場合でも、40 ビット・フォーマットとして見た場合の値は正確です ($0x00'8000'0000 = +1$)。
2. 16 ビット値同士を乗算すると、有効となるのは最大で 31 ビットです。したがって、乗算結果を示すレジスタの LSB は常に 0 になります。

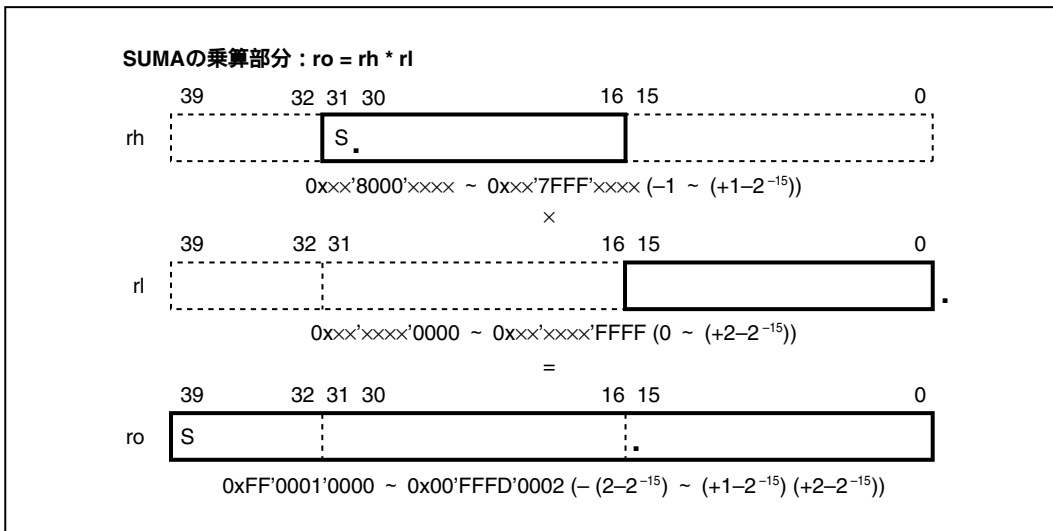
(b) 有符号値と無符号値の乗算

一方のパラメータは16ビット固定小数点型で汎用レジスタのHパートに設定され、ビット31は符号を表しています。他方のパラメータは整数型フォーマットで汎用レジスタのLパートに設定されます。

この演算過程のイメージを、図4-34に示します。

注意 この演算を単独で実行する命令はありません。サイン・アンサイン・マルチプライ・アド命令の一部として機能します。

図4-34 有符号値と無符号値の乗算

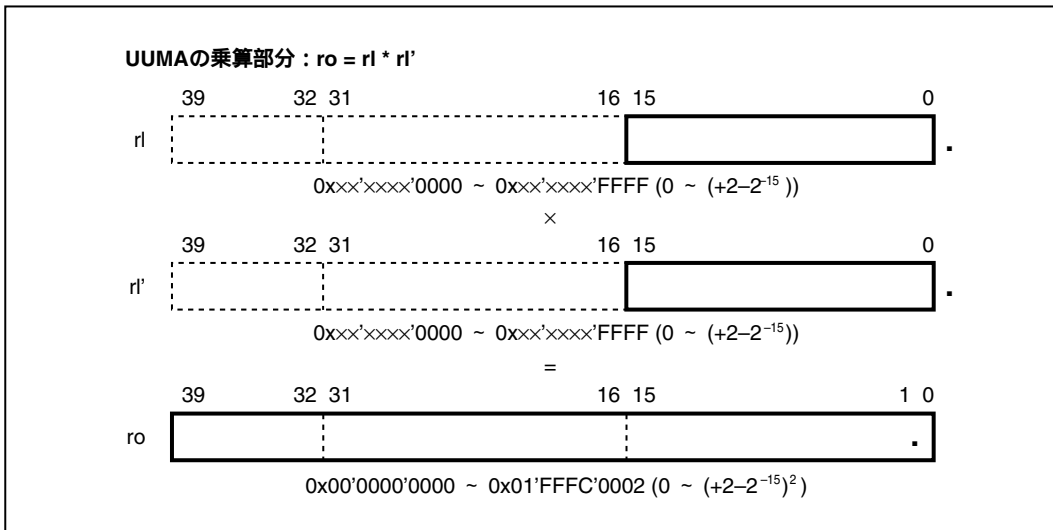


(c) 無符号値同士の乗算

2つのパラメータはともに整数型フォーマットで汎用レジスタのLパートに設定されます。この演算過程のイメージを、図4-35に示します。

注意 この演算を単独で実行する命令はありません。アンサイン・アンサイン・マルチプライ・アド命令の一部として機能します。

図4-35 無符号値同士の乗算



(2) 累乗算機能 (三項演算)

μ PD77210 ファミリで実行される三項演算は、すべて累乗算です。累乗算機能は、累算を指定するレジスタへのシフト指示によって、次の3種類があります (ただし、累乗算には累加算と累減算の2種類があります)。このとき、シフト処理はMAC入力シフタ (MSFT : MAC input shifter) が実行します。

累乗算

1ビット・シフト累乗算

16ビット・シフト累乗算

また、乗算にかかわるパラメータのデータ型から分類すると次の3種類があります。

有符号値同士の乗算

有符号値と無符号値の乗算

無符号値同士の乗算

これらを組み合わせて、次に示す6個の三項演算命令があります。

マルチプライ・アド (有符号値同士の乗算と累加算)

マルチプライ・サブ (有符号値同士の乗算と累減算)

サイン・アンサイン・マルチプライ・アド (有符号値と無符号値の乗算と累加算)

アンサイン・アンサイン・マルチプライ・アド (無符号値と無符号値の乗算と累加算)

1ビット・シフト・マルチプライ・アド (有符号値同士の乗算と1ビット・シフト後への累加算)

16ビット・シフト・マルチプライ・アド (有符号値同士の乗算と16ビット・シフト後への累加算)

累乗算機能は、3つの項 (パラメータ) が関係する三項演算です。そのうちの2つが乗算のパラメータであり、もう1つが累算用のパラメータです。3つのパラメータにはいずれも汎用レジスタが指定されますが、それら汎用レジスタは任意に重複してかまいません。

表 4-19にこれらの組み合わせを示します。

表 4-19 累乗算機能

		有符号値同士	有符号値と無符号値	無符号値同士
累乗算	MSFT0ビット	$ro = ro \pm rh * rh'$ (MADD, MSUB)	$ro = ro + rh * rl$ (SUMA)	$ro = ro + rl * rl'$ (UUMA)
	MSFT1ビット	$ro = (ro \gg 1) + rh * rh'$ (MAS1)	-	-
	MSFT16ビット	$ro = (ro \gg 16) + rh * rh'$ (MAS16)	-	-
乗算 (二項演算)		$ro = rh * rh'$ (MPY)	-	-

(a) 累乗算

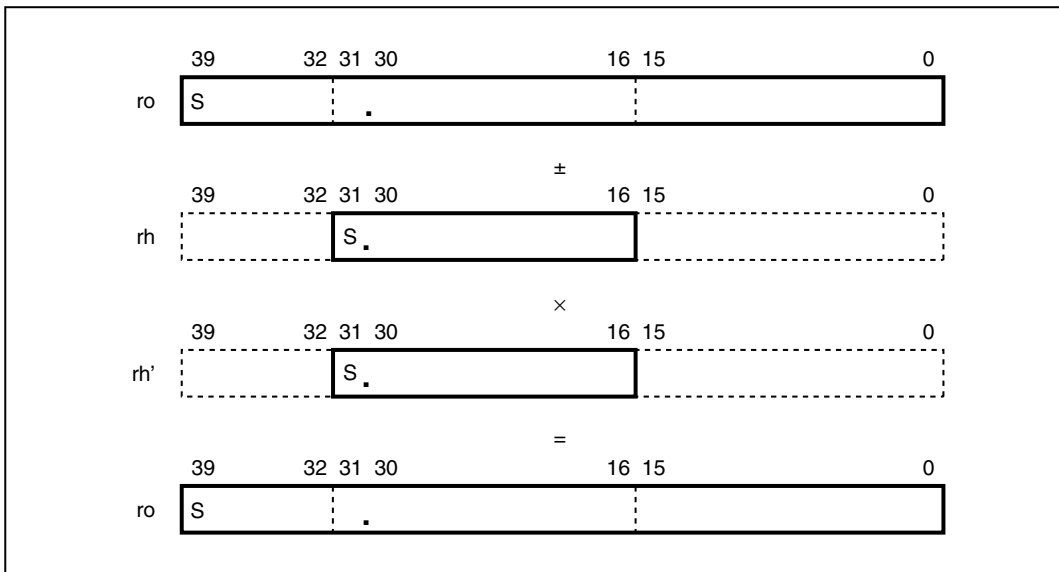
2つの乗算の入力は(符号付き)16ビット固定小数点型です。これらの乗算結果と累算用の40ビット固定小数点データを加算します。関連する命令は次のとおりです。

MADD : $ro = ro + rh * rh'$

MSUB : $ro = ro - rh * rh'$

図 4-36にこの操作のイメージを示します。

図 4-36 累乗算



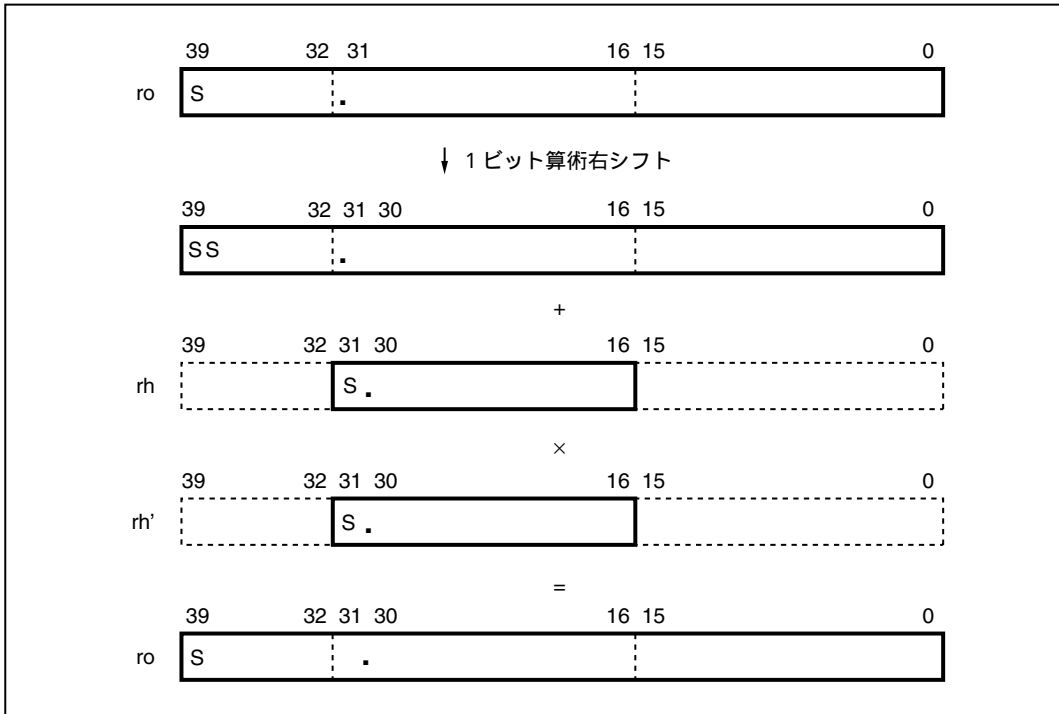
(b) 1ビット・シフト累乗算

2つの乗算の入力は(符号付き)16ビット固定小数点型です。これらの乗算結果と1ビット算術右シフトした累算用の40ビット固定小数点データを加算します。関連する命令は次のとおりです。

MAS1 : ro = (ro >> 1) + rh*rh'

図 4-37にこの操作のイメージを示します。

図 4-37 1ビット・シフト累乗算



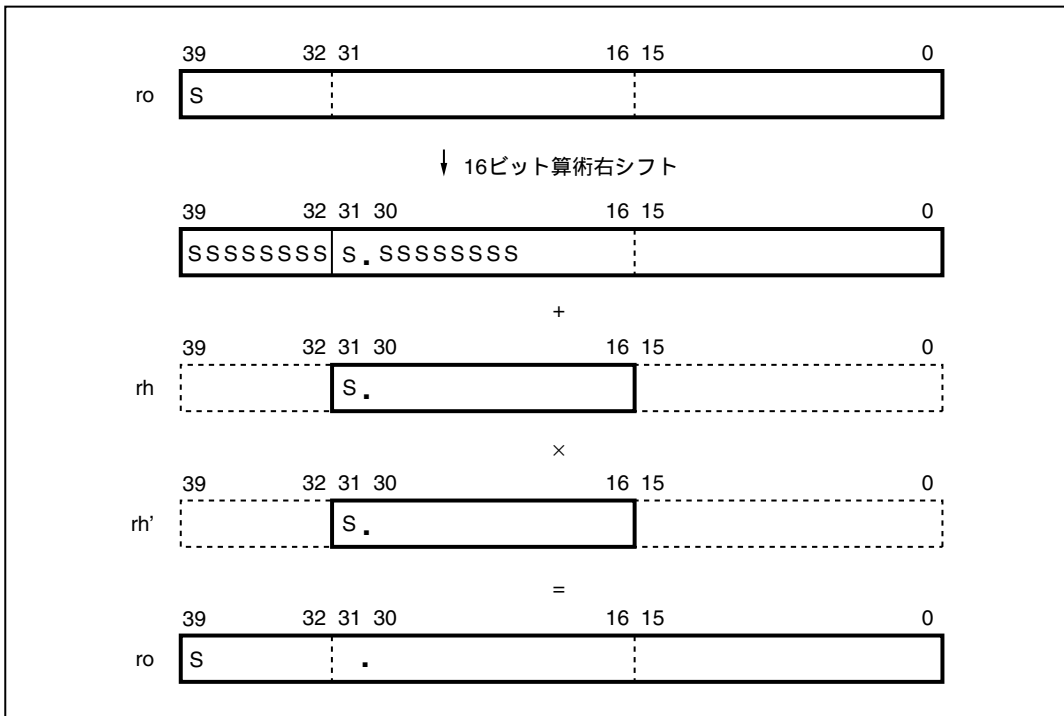
(c) 16 ビット・シフト累乗算

2つの乗算の入力は(符号付き)16ビット固定小数点型です。これらの乗算結果と16ビット算術右シフト累算用40ビット固定小数点データを加算します。関連する命令は次のとおりです。

$$\text{MAS16} : \text{ro} = (\text{ro} \gg 16) + \text{rh} * \text{rh}'$$

図 4-38にこの操作のイメージを示します。

図 4-38 16 ビット・シフト累乗算



4.6.4 ALU 演算機能

ALU (Arithmetic and logic unit) は、2つまたは1つの40ビット入力に対して算術演算、論理演算を実行し、1つの40ビット・データを出力します。

二項演算のパラメータは、2つとも汎用レジスタに、または1つを汎用レジスタにもう1つをイミューディエト・データにそれぞれ指定します(ただし、LT命令ではイミューディエト・データは使用できません)。2つとも汎用レジスタに指定した場合、それらは重複してもかまいません。単項演算では任意の汎用レジスタをパラメータとして指定します。さらに、これらの演算で結果レジスタとして任意の汎用レジスタを指定できます。

注意 MAC, ALU, BSFT は同時に動作することはできません。

(1) 算術演算**(a) 算術二項演算**

算術演算のうち、二項演算命令は次のものがあります。

それぞれの命令については、 μ PD77016 ファミリ ユーザーズ・マニュアル 命令編を参照してください。

マルチプライ命令 (MPY : MAC で実行されます)

アド命令 (ADD)

イミディエト・アド命令 (IADD)

サブ命令 (SUB)

イミディエト・サブ命令 (ISUB)

レスザン命令 (LT)

(b) 算術単項演算

算術演算で単項演算命令は次のものがあります。

それぞれの命令については、 μ PD77016 ファミリ ユーザーズ・マニュアル 命令編を参照してください。

クリア命令 (CLR)

インクリメント命令 (INC)

デクリメント命令 (DEC)

絶対値命令 (ABS)

2つの補数命令 (NEG)

クリップ命令 (CLIP)

丸め命令 (RND)

指数命令 (EXP)

代入命令 (PUT): この命令が実質的な汎用レジスタ間の転送命令となります。

累加算命令 (ACA)

累減算命令 (ACS)

除算命令 (DIV)

(2) 論理演算**(a) 論理二項演算**

論理演算のうち、二項演算命令は次のものがあります。

それぞれの命令については、 μ PD77016 ファミリ ユーザーズ・マニュアル 命令編を参照してください。

アンド命令 (AND)

イミーディエト・アンド命令 (IAND)

オア命令 (OR)

イミーディエト・オア命令 (IOR)

イクスクルーシブ・オア命令 (XOR)

イミーディエト・イクスクルーシブ・オア命令 (IXOR)

(b) 論理単項演算

論理演算で単項演算命令は次のものがあります。

それぞれの命令については、 μ PD77016 ファミリ ユーザーズ・マニュアル 命令編を参照してください。

1の補数命令 (NOT)

注意 イミーディエト値として設定できる値の範囲は 0-0xFFFF (0-65536) で、ビット 15-ビット 0 に設定されます。それぞれの演算は、この 16 ビットのイミーディエト値に対して、ビット 39-ビット 16 を 0 拡張した 40 ビット・データとの間で行われます。

4.6.5 BSFT シフト演算機能

BSFT (Barrel shifter) は、シフト動作を実行します。シフト動作はすべて二項演算で行われます。BSFT は、40 ビット入力データに対して、1 インストラクション・サイクルで任意のシフト・パターンを 40 ビット・データとして出力します。

二項演算のパラメータは、2 つとも汎用レジスタに、または 1 つを汎用レジスタにもう 1 つをイミディエト・データにそれぞれ指定します。2 つとも汎用レジスタに指定した場合、それらは重複してもかまいません。

単項演算では任意の汎用レジスタをパラメータとして指定します。さらに、これらの演算で結果レジスタとして任意の汎用レジスタを指定できます。

注意 MAC, ALU, BSFT は同時に動作することはできません。

(1) シフト演算命令

シフト演算はすべて二項演算です。シフト演算には次のものがあります。

それぞれの命令については *μ PD77016 ファミリー ユーザーズ・マニュアル 命令編* を参照してください。

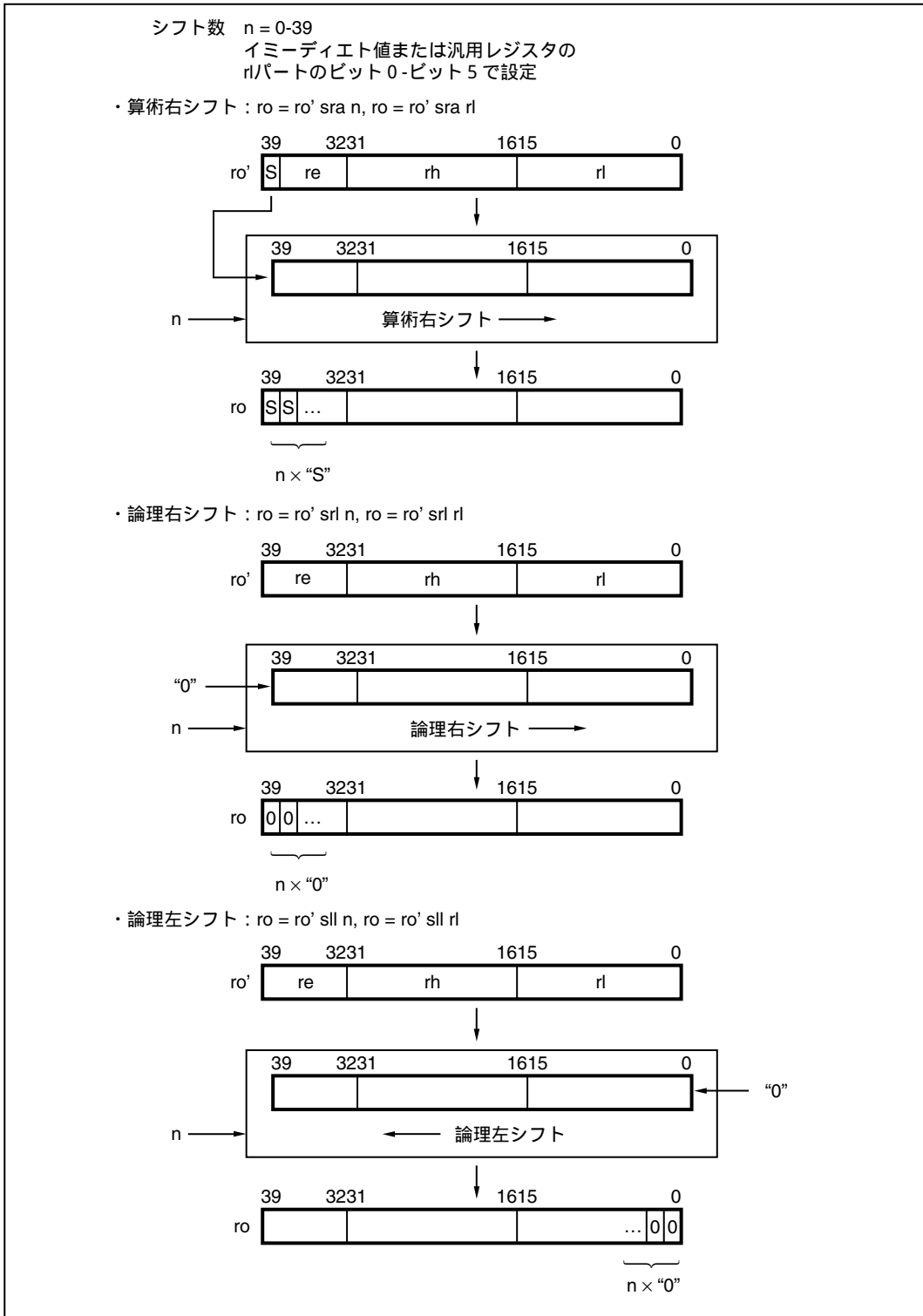
- 算術右シフト命令 (SRA)
- イミディエト算術右シフト命令 (ISRA)
- 論理右シフト命令 (SRL)
- イミディエト論理右シフト命令 (ISRL)
- 論理左シフト命令 (SLL)
- イミディエト論理左シフト命令 (ISLL)

注意 汎用レジスタまたはイミディエト・データに、シフト量として設定できる値の範囲は 0-0x27(0-39) で、ビット 5-ビット 0 に設定されます。ビット 15-ビット 6 の値は無視されます。

(2) シフト演算機能

図 4-39に BSFT の各機能を示します。

図 4-39 BSFT 演算



第5章 ペリフェラル

μ PD77210 ファミリは、次に示すペリフェラル・インタフェース機能を内蔵しています。

これらのペリフェラルは、 μ PD77210 ファミリの内部データ領域にマッピングされたレジスタを介して取り扱うことができます。

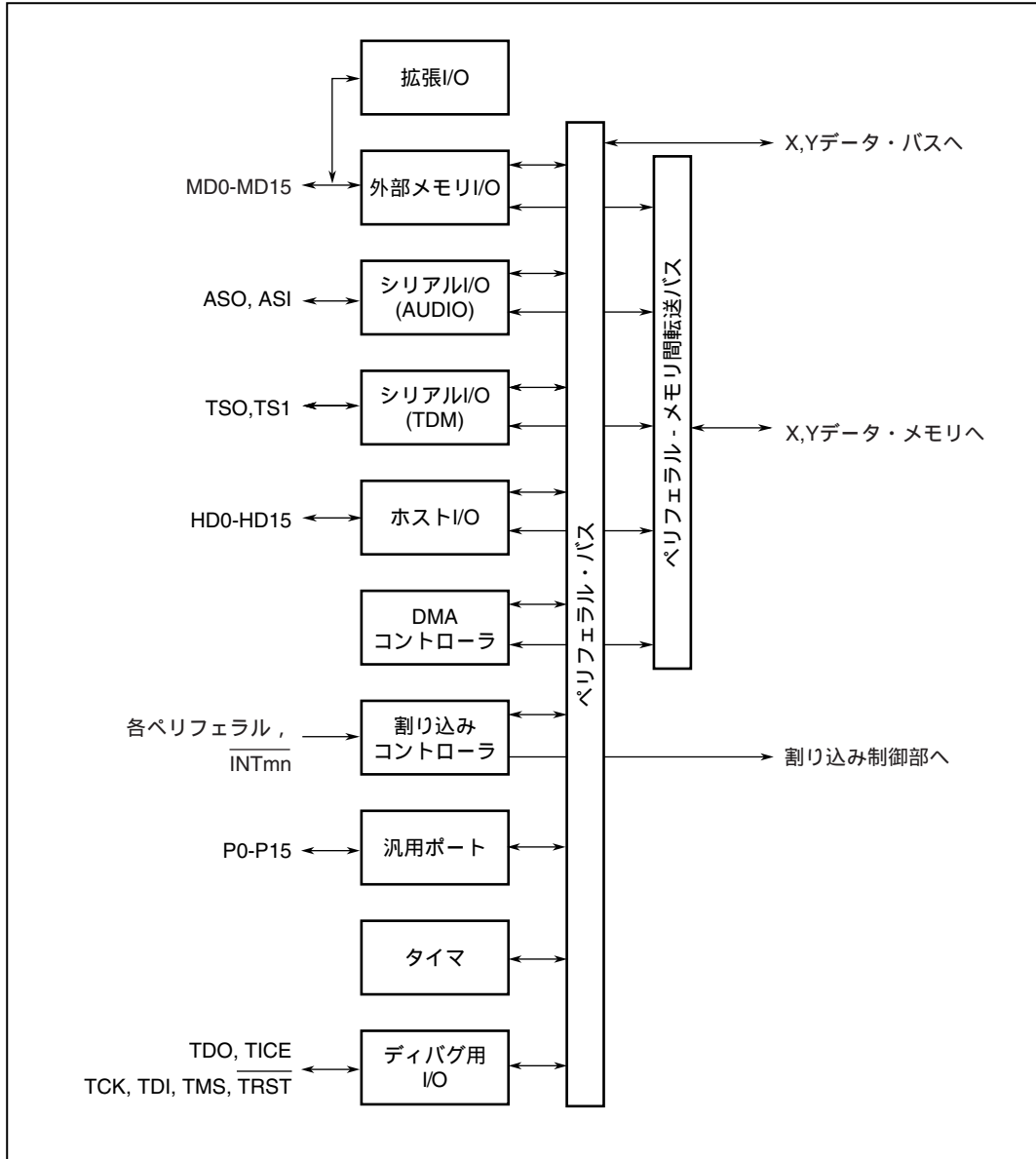
- 時分割 (TDM) シリアル・インタフェース : TSIO
- オーディオ・シリアル・インタフェース : ASIO
- スタンダード・シリアル・インタフェース : SIO
- ホスト・インタフェース (パラレル・インタフェース) : HIO
- 外部データ・メモリ・インタフェース : MIO
- ペリフェラル・メモリ間転送 (DMA) : PMT
- 汎用入出力ポート : PIO
- 割り込みコントローラ : INTC
- タイマ : TIM
- クロック・コントローラ : CLKC
- ディバグ用インタフェース : IEIO^注
- 拡張インタフェース : Additional IO (μ PD77213 のみ)
 - SD カード・インタフェース : SDCIF

注 ユーザ・プログラムから取り扱うことはできません。

5.1 ブロック構成

図 5-1にペリフェラル・ユニットのブロック構成を示します。

図 5-1 ペリフェラル・ユニット



5.2 ペリフェラル・レジスタ

表 5-1にペリフェラル・レジスタとメモリ空間へのマッピングを示します。

表 5-1 ペリフェラル・レジスタのメモリ・マッピング (1/3)

X/Y メモリ・アドレス	レジスタ名	機 能	ペリフェラル名	
0x3800	TSDT/SDT1	TDM シリアル・データ・レジスタ/シリアル・データ・レジスタ 1	TSIO (SIO1)	
0x3801	SST1	シリアル・ステータス・レジスタ 1		
0x3802	TSST	TDM シリアル・ステータス・レジスタ		
0x3803	TFMT	TDM フレーム・フォーマット・レジスタ		
0x3804	TTXL	TDM 送信スロット・レジスタ (下位)		
0x3805	TTXH	TDM 送信スロット・レジスタ (上位)		
0x3806	TRXL	TDM 受信スロット・レジスタ (下位)		
0x3807	TRXH	TDM 受信スロット・レジスタ (上位)		
0x3808-0x380F	予約領域	注意 この領域にはアクセスしないでください。	-	
0x3810	ASDT/SDT2	オーディオ・シリアル・データ・レジスタ/シリアル・データ・レジスタ 2	ASIO (SIO2)	
0x3811	SST2	シリアル・ステータス・レジスタ 2		
0x3812	ASST	オーディオ・シリアル・ステータス・レジスタ		
0x3813-0x381F	予約領域	注意 この領域にはアクセスしないでください。	-	
0x3820	HDT	ホスト・インタフェース・データ・レジスタ	HIO	
0x3821	HST	ホスト・インタフェース・ステータス・レジスタ		
0x3822-0x383F	予約領域	注意 この領域にはアクセスしないでください。	-	
0x3840	MDT	メモリ・データ・レジスタ	MIO	
0x3841	MSHW	メモリ I/F セットアップ/ホールド幅設定レジスタ		
0x3842	MCST	メモリ I/F コントロール/ステータス・レジスタ		
0x3843	MWAIT	メモリ I/F ウェイト・レジスタ		
0x3844	MIDX	ダイレクト・アクセス・インデクス・レジスタ		
0x3845	MADRLI	メモリ I/F 入力用スタート・アドレス・レジスタ (下位)		
0x3846	MADRHI	メモリ I/F 入力用スタート・アドレス・レジスタ (上位)		
0x3847	MOFSI	メモリ I/F 入力用ライン・オフセット・レジスタ		
0x3848	MLENI	メモリ I/F 入力用ライン・レンジ・レジスタ		
0x3849	MADRLO	メモリ I/F 出力用スタート・アドレス・レジスタ (下位)		
0x384A	MADRHO	メモリ I/F 出力用スタート・アドレス・レジスタ (上位)		
0x384B	MOFSO	メモリ I/F 出力用ライン・オフセット・レジスタ		
0x384C	MLENO	メモリ I/F 出力用ライン・レンジ・レジスタ		
0x384D-0x384F	予約領域	注意 この領域にはアクセスしないでください。		-
0x3850	PMSA0	PMT スタート・アドレス・レジスタ 0		PMT ch0
0x3851	PMS0	PMT サイズ・レジスタ 0		
0x3852	PMC0	PMT 制御レジスタ 0		
0x3853	PMP0	PMT アドレス・ポインタ 0		

表 5-1 ペリフェラル・レジスタのメモリ・マッピング (2/3)

X/Y メモリ・アドレス	レジスタ名	機 能	ペリフェラル名
0x3854	PMSA1	PMT スタート・アドレス・レジスタ 1	PMT ch1
0x3855	PMS1	PMT サイズ・レジスタ 1	
0x3856	PMC1	PMT 制御レジスタ 1	
0x3857	PMP1	PMT アドレス・ポインタ 1	
0x3858	PMSA2	PMT スタート・アドレス・レジスタ 2	PMT ch2
0x3859	PMS2	PMT サイズ・レジスタ 2	
0x385A	PMC2	PMT 制御レジスタ 2	
0x385B	PMP2	PMT アドレス・ポインタ 2	
0x385C	PMSA3	PMT スタート・アドレス・レジスタ 3	PMT ch3
0x385D	PMS3	PMT サイズ・レジスタ 3	
0x385E	PMC3	PMT 制御レジスタ 3	
0x385F	PMP3	PMT アドレス・ポインタ 3	
0x3860	PMSA4	PMT スタート・アドレス・レジスタ 4	PMT ch4
0x3861	PMS4	PMT サイズ・レジスタ 4	
0x3862	PMC4	PMT 制御レジスタ 4	
0x3863	PMP4	PMT アドレス・ポインタ 4	
0x3864	PMSA5	PMT スタート・アドレス・レジスタ 5	PMT ch5
0x3865	PMS5	PMT サイズ・レジスタ 5	
0x3866	PMC5	PMT 制御レジスタ 5	
0x3867	PMP5	PMT アドレス・ポインタ 5	
0x3868	PMSA6	PMT スタート・アドレス・レジスタ 6	PMT ch6
0x3869	PMS6	PMT サイズ・レジスタ 6	
0x386A	PMC6	PMT 制御レジスタ 6	
0x386B	PMP6	PMT アドレス・ポインタ 6	
0x386C	PMSA7	PMT スタート・アドレス・レジスタ 7	PMT ch7
0x386D	PMS7	PMT サイズ・レジスタ 7	
0x386E	PMC7	PMT 制御レジスタ 7	
0x386F	PMP7	PMT アドレス・ポインタ 7	
0x3870	PDT0	ポート・データ・レジスタ 0	PIO
0x3871	PCD0	ポート・コマンド・レジスタ 0	
0x3872	PDT1	ポート・データ・レジスタ 1	
0x3873	PCD1	ポート・コマンド・レジスタ 1	
0x3874	PDT2	ポート・データ・レジスタ 2	
0x3875	PCD2	ポート・コマンド・レジスタ 2	
0x3876	PDT3	ポート・データ・レジスタ 3	
0x3877	PCD3	ポート・コマンド・レジスタ 3	
0x3878, 0x3879	予約領域	注意 この領域にはアクセスしないでください。	-
0x387A-0x387B	POWC	パワー制御レジスタ	Peripheral STOP mode
0x387C-0x3879	予約領域	注意 この領域にはアクセスしないでください。	-
0x3880	ICR0	割り込み制御レジスタ 0	INTC
0x3881	ICR1	割り込み制御レジスタ 1	
0x3882	ICR2	割り込み制御レジスタ 2	

表 5-1 ペリフェラル・レジスタのメモリ・マッピング (3/3)

X/Y メモリ・アドレス	レジスタ名	機 能	ペリフェラル名	
0x3883	ICR3	割り込み制御レジスタ 3	INTC	
0x3884	ICR4	割り込み制御レジスタ 4		
0x3885	ICR5	割り込み制御レジスタ 5		
0x3886	ICR6	割り込み制御レジスタ 6		
0x3887	ICR7	割り込み制御レジスタ 7		
0x3888	ICR8	割り込み制御レジスタ 8		
0x3889	ICR9	割り込み制御レジスタ 9		
0x388A	ICR10	割り込み制御レジスタ 10		
0x388B	ICR11	割り込み制御レジスタ 11		
0x388C-0x388F	予約領域	注意 この領域にはアクセスしないでください。		-
0x3890	TIR0	タイマ初期値設定レジスタ 0		TIM0
0x3891	TCR0	タイマ・カウント・レジスタ 0		
0x3892	TCSR0	タイマ制御レジスタ 0		
0x3893	予約領域	注意 この領域にはアクセスしないでください。	-	
0x3894	TIR1	タイマ初期値設定レジスタ 1	TIM1	
0x3895	TCR1	タイマ・カウント・レジスタ 1		
0x3896	TCSR1	タイマ制御レジスタ 1		
0x3897-0x389F	予約領域	注意 この領域にはアクセスしないでください。	-	
0x38A0	CEFR	訂正イネーブル・フラグ・レジスタ	IMC	
0x38A1	CPR0	訂正ページ・レジスタ 0		
0x38A2	CAR0	訂正アドレス・レジスタ 0		
0x38A3	CLIR0	訂正命令コード・レジスタ (上位) 0		
0x38A4	CUIR0	訂正命令コード・レジスタ (下位) 0		
0x38A5	CPR1	訂正ページ・レジスタ 1		
0x38A6	CAR1	訂正アドレス・レジスタ 1		
0x38A7	CLIR1	訂正命令コード・レジスタ (上位) 1		
0x38A8	CUIR1	訂正命令コード・レジスタ (下位) 1		
0x38A9-0x38AF	予約領域	注意 この領域にはアクセスしないでください。		-
0x38B0	CLKC	クロック制御レジスタ		CLKC
0x38B1-0x38BF	予約領域	注意 この領域にはアクセスしないでください。	-	
0x38C0	IPR	命令ページング・レジスタ	Page Register	
0x38C1	DPR	データ・ページング・レジスタ		
0x38C2-0x38CF	予約領域	注意 この領域にはアクセスしないでください。	-	
0x38D0	APCR [※]	拡張インタフェース制御レジスタ	Additional IO	
0x38D1-0x3FFF	予約領域	注意 この領域にはアクセスしないでください。	-	

注 μ PD77213 のみ。μ PD77210 では予約領域です。

注意 1.ここに示されたレジスタ名称は、アセンブラやC言語の予約語ではありません。したがって、アセンブラやC言語でこれらの名前を取り扱う場合には、ユーザによる定義付けが必要です。

2. これらのレジスタは、Xメモリ空間、Yメモリ空間のどちらのメモリ空間からアクセスしても、アドレスが同じであれば同一のレジスタにアクセスします。
3. 異なるレジスタであっても、Xメモリ空間、Yメモリ空間の両方から同時にアクセスすることはできません。

5.3 時分割 (TDM) シリアル・インタフェース (TSIO)

時分割シリアル・インタフェース (TSIO) は、標準的な音声 CODEC と直接接続が可能なシリアル・インタフェース (スタンダード・モード) であるとともに、最大 128 スロットを設定できる時分割多重 (Time Division Multiplexing モード) のシリアル転送機能を持つ周辺回路です。TDM 機能によりマルチプロセッサ間の通信を容易に実現できます。

スタンダード・モードについては5.5 スタンダード・シリアル・インタフェース (SIO) を参照してください。主な特徴を次に示します。

- シリアル・クロック供給
外部からの供給可、入力、出力でクロック共用
- ビット長
8 ビット、16 ビット設定、MSB ファースト、LSB ファーストのフォーマット選択可能
- 内部ハンドシェーク
ポーリング、ウェイト、および割り込みによるハンドシェークが可能
- T1, E1 フォーマット・サポート
フレーム信号に対する遅延入出力の設定、1 ビットのダミー・ビットの入出力、フレーム信号の極性の選択設定が可能
- 最大 128 スロット・サポート
1 フレームあたり 32 スロットまで定義可能であり、またフレーム位置は 16 スロット刻みで最大 128 スロットまで設定可能

時分割シリアル・インタフェースのブロック図を図 5-2 に、レジスタ群を表 5-2 示します。

図 5-2 時分割シリアル・インタフェースのブロック図

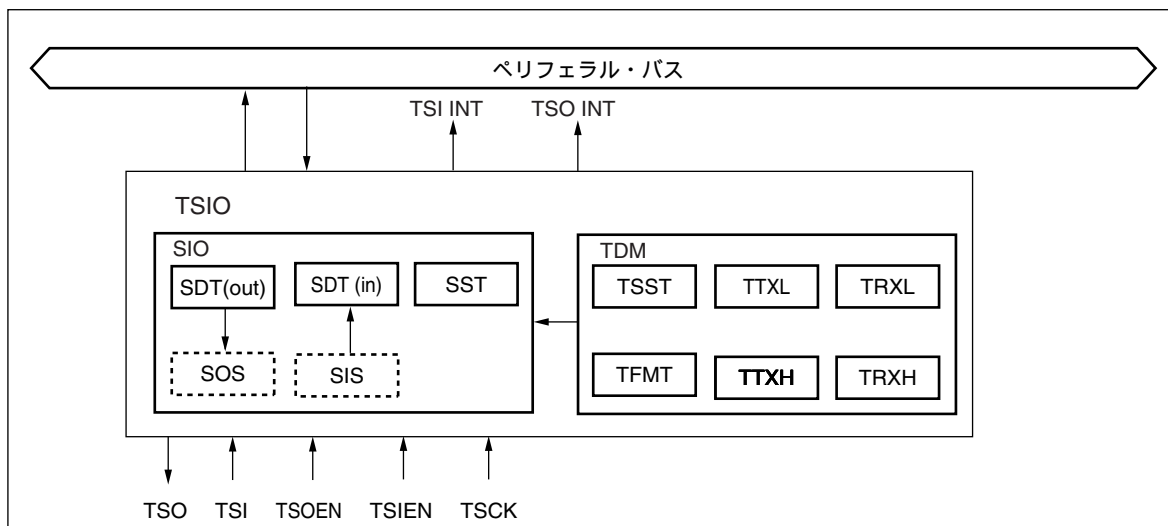


表 5-2 時分割シリアル・インタフェースのレジスタ

X/Y メモリ・アドレス	レジスタ名	機 能	ロード/ストア
0x3800	TSDT	TDM シリアル・データ・レジスタ	L/S
0x3801	SST1	シリアル・ステータス・レジスタ 1	L/S
0x3802	TSST	TDM シリアル・ステータス・レジスタ	L/S
0x3803	TFMT	TDM フォーマット・レジスタ	L/S
0x3804	TTXL	TDM 転送スロット・レジスタ (下位)	L/S
0x3805	TTXH	TDM 転送スロット・レジスタ (上位)	L/S
0x3806	TRXL	TDM 受信スロット・レジスタ (下位)	L/S
0x3807	TRXH	TDM 受信スロット・レジスタ (上位)	L/S

5.3.1 時分割シリアル・インタフェースの端子

(1) TSCK (シリアル・クロック - 入力)

時分割シリアル・データ入出力用のクロック端子です。

シリアル・データの入出力, 各種シリアル・インタフェース用信号の出力およびサンプリングは TSCK に同期して行います。

(2) TSOEN (時分割シリアル出力イネーブル - 入力)

時分割シリアル・データの出力許可信号の入力端子です。

TSCK の立ち下がりに同期してサンプリングします。

外部デバイスが, シリアル出力データを入力する用意が整ったときに, アクティブ (ハイ・レベル) にします。

(3) TSIEN (時分割シリアル入力イネーブル - 入力)

時分割シリアル・データの入力許可信号の入力端子です。

TSCK の立ち下がりに同期してサンプリングします。

外部デバイスが, シリアル入力データを出力する用意が整ったときに, アクティブ (ハイ・レベル) にします。

(4) TSI (時分割シリアル・データ入力 - 入力)

時分割シリアル・データの入力端子です。

TCLK の立ち下がりに同期してサンプリングします。

(5) TSO (時分割シリアル・データ出力 - 出力)

時分割シリアル・データの出力端子です。

TCLK の立ち上がりに同期して出力が変化します。

5.3.2 時分割シリアル・インタフェースのレジスタ

(1) TSdT (TDM シリアル・データ・レジスタ : TDM Serial Data Transfer Register)

時分割シリアル・データを入出力するための 16 ビット・レジスタです。出力用と入力用のレジスタを別々に持っています。

SDT(out)は、出力するデータを設定する 16 ビットのレジスタです。TSdT に対するストア命令を実行すると SDT(out)にペリフェラル・バスからデータを入力します。

シリアル出力シフト・レジスタ (SOS) が空になると TSdT の値を SOS に設定し、TSO 端子から出力します。SSEF フラグが「0」のときに SDT に対するストア命令を実行すると、SSER フラグが「1」になります (ストア・エラー)。

SDT(in)は、入力されたデータを読み出す 16 ビットのレジスタです。TSdT に対してロード命令を実行すると、ペリフェラル・バスにデータを出力します。

TSI 端子から入力し、シリアル入力シフト・レジスタ (SIS) に最後のビットが入力されると、SIS の値を TSdT に設定します。SLEF フラグが「0」のときに SDT からのロード命令を実行すると、SLER フラグが「1」になります (ロード・エラー)。

(2) TSST (TDM シリアル・ステータス・レジスタ : TDM Serial StaTus Register)

時分割シリアル入出力モードの設定と、ステータスを表す 16 ビットのレジスタです。スタンダード / 時分割シリアル・インタフェース・モードの切り替え、フレーム信号の取り扱い方など、 μ PD77210 ファミリの DSP コア・カーネル部分とのインタフェースの指定やオーバーラン、アンダランの表示を行います。

TSST のビット構成については表 5-3に示します。リセット時の値は、0x5 です。

(3) SST1 (シリアル・ステータス・レジスタ : Serial StaTus Register)

シリアル入出力モードの設定と、ステータスを表す 16 ビットのレジスタです。MSB/LSB ファーストの設定、 μ PD77210 ファミリの DSP コア・カーネル部分とのインタフェースの指定やオーバーラン、アンダランの表示を行います。SST1 のビット構成については表 5-4に示します。リセット時の値は、0x00C2 です。

(4) TFMT (TDM フレーム・フォーマット・レジスタ : TDM FraMe Format Register)

時分割シリアル入出力のフォーマット設定を行う 16 ビットのレジスタです。

TFMT のビット構成については表 5-6に示します。リセット時の値は、0x0 です。

(5) TTXL, TTXH (TDM 送信スロット・レジスタ : TDM Transmit Slot Register)

時分割シリアル・データ出力の送信アクセス・フレームを設定する 16 ビットのレジスタです。上位 16 スロットを指定する TTXH と下位 16 スロットを指定する TTXL があります。

0 が設定されているビットに対応するスロットが送信に割り当てられるスロットになります。1 が設定されているスロットではデータを出力しません。複数の μ PD77210 ファミリを接続している場合は、複数の μ PD77210 ファミリで同じスロットに対して 0 を設定しないでください。リセット時の値は、0xFFFF です。

(6) TRXL, TRXH (TDM 受信スロット・レジスタ : TDM Receive Slot Register)

時分割シリアル・データ入力の受信アクセス・フレームを設定する 16 ビットのレジスタです。上位 16 スロットを指定する TRXH と下位 16 スロットを指定する TRXL があります。

0 が設定されているビットに対応するスロットが受信に割り当てられるスロットになります。1 が設定されているスロットではデータを無視します。リセット時の値は、0xFFFF です。

(7) SOS (シリアル出力シフト・レジスタ : Serial Output Shift Register)

時分割シリアル・データを TSO 端子から出力しながらシフトする 16 ビットのシフト・レジスタです。指定したビット数の出力が終わると TSDT から新しいデータを入力します。

SOS は、ペリフェラル・バスとは接続していません。

(8) SIS (シリアル入力シフト・レジスタ : Serial Input Shift Register)

TSI 端子から入力されるビット列を入力する 16 ビットのシフト・レジスタです。指定したビット数の入力が終わると、TSDT にデータを出力します。

SIS は、ペリフェラル・バスとは接続していません。

表 5-3 TSST のビット構成

ビット	名 称	機 能	ロード / ストア
15-8	予約	<ul style="list-style-type: none"> ・ 0 以外の書き込みは行わないこと。 ・ 読み出し時は不定。 	-
7	FSP	フレーム同期信号の極性設定ビット <ul style="list-style-type: none"> ・ 0 : フレーム同期信号立ち上がり同期 (デフォルト) ・ 1 : フレーム同期信号立ち下がり同期 	L/S
6	FSD	データ入出力遅延設定ビット <ul style="list-style-type: none"> ・ 0 : フレーム同期信号に対して次の SCK クロックからデータの入出力を始める (デフォルト) ・ 1 : フレーム同期信号に対して 2SCK クロック後からデータの入出力を始める。 	L/S
5	OBP	1 ビットのダミー・ビット挿入許可ビット <ul style="list-style-type: none"> ・ 0 : ダミー・ビットを挿入しない (デフォルト) ・ 1 : のダミー・ビットを挿入する。 	L/S
4	TDME	スタンダード / 時分割シリアル・インタフェース切り替えビット <ul style="list-style-type: none"> ・ 0 : スタンダード・シリアル・モード (デフォルト) ・ 1 : 時分割シリアル・モード 	L/S
3	ERRTX	送信フレーム・エラー・フラグ <ul style="list-style-type: none"> ・ 0 : TFSx フレーム同期エラーなし (デフォルト) ・ 1 : TFSx フレーム同期エラーあり 設定したフレームのクロック数より短い時間で次のフレーム信号が入力されたときに 1 になる。 一度セットされると、 μ PD77210 ファミリが 0 を書き込むまで変化しない。	L/S
2	予約	<ul style="list-style-type: none"> ・ 0 以外の書き込みは行わないこと。 ・ 読み出し時は不定。 	-
1	ERRRX	受信フレーム・エラー・フラグ <ul style="list-style-type: none"> ・ 0 : RFSx フレーム同期エラーなし (デフォルト) ・ 1 : RFSx フレーム同期エラーあり 設定したフレームのクロック数より短い時間で次のフレーム信号が入力されたときに 1 になる。 一度セットされると、 μ PD77210 ファミリが 0 を書きこむまで変化しない	L/S
0	予約	<ul style="list-style-type: none"> ・ 0 以外の書き込みは行わないこと。 ・ 読み出し時は不定。 	-

表 5-4 SST1 のビット構成 (1/2)

ビット	名称	機能	ロード/ ストア
15	SOTF	シリアル出力転送フォーマット設定ビット ・0: MSB ファーストでシリアル出力 (デフォルト) ・1: LSB ファーストでシリアル出力。	L/S
14	SITF	シリアル入力転送フォーマット設定ビット ・0: MSB ファーストでシリアル入力 (デフォルト) ・1: LSB ファーストでシリアル入力	L/S
13	SOBL	シリアル出力ワード長設定ビット ・0: 16 ビットでシリアル出力 (デフォルト) ・1: 8 ビットでシリアル出力	L/S
12	SIBL	シリアル入力ワード長設定ビット ・0: 16 ビットでシリアル入力 (デフォルト) ・1: 8 ビットでシリアル入力	L/S
11	SSWE	TSDT ストア・ウエイト許可ビット ・0: ストア・ウエイトを使用しない (デフォルト) ・1: ストア・ウエイトを使用する。 SSEF が 0 のときに, μ PD77210 ファミリが TSDT にデータをストアしようとした場合に, ウエイトを挿入する。	L/S
10	SLWE	TSDT ロード・ウエイト許可ビット ・0: ロード・ウエイトを使用しない (デフォルト) ・1: ロード・ウエイトを使用する。 SLEF が 0 のときに, μ PD77210 ファミリが TSDT からデータをロードしようとした場合に, ウエイトを挿入する。	L/S
9	SICM	シリアル入力連続モード設定フラグ ・0: 実行中のシリアル入力処理終了後, シリアル入力モードが単発モードになる。 ・1: シリアル入力モードが連続モードになり, シリアル入力処理を開始する。	L/S
8	SIEF	シリアル単発入力許可フラグ ・0: (デフォルト) ・1: シリアル入力単発モードでシリアル入力処理を開始する (1 回のみ) 1 にセットした SIEF は次の命令サイクルで自動的にリセットされる。	L/S
7	SOE	シリアル出力許可ビット ・0: シリアル出力使用不可 ・1: シリアル出力使用可 (デフォルト)	L/S
6	SIE	シリアル入力許可ビット ・0: シリアル入力使用不可 ・1: シリアル入力使用可 (デフォルト)	L/S
5	SORST ^注	シリアル出力リセット許可ビット シリアル出力回路をリセットする。SSER=0, SSEF=1 となる。シリアル出力リセット後, 自動的にゼロクリアされる。 0: (デフォルト) 1: SO リセット要求	L/S

表 5-4 SST1 のビット構成 (2/2)

ビット	名称	機 能	ロード/ ストア
4	SIRST ^{注1}	シリアル入力リセット許可ビット シリアル入力回路をリセットする。SLER=0, SLEF=0 となる。シリアル入力リセット後、自動的にゼロクリアされる。 0:(デフォルト) 1:SIリセット要求	L/S
3	SSER ^{注2}	TSDT ストア・エラー・フラグ ・0:エラーなし(デフォルト) ・1:エラー SSEF が 0 のときにμ PD77210 ファミリが SDT に書き込んだ場合に 1 になる。 一度セットされると、μ PD77210 ファミリが 0 を書き込むまで変化しない。	L/S
2	SLER ^{注2}	SDT ロード・エラー・フラグ ・0:エラーなし(デフォルト) ・1:エラー SLEF が 0 のときにμ PD77210 ファミリが SDT を読み出した場合に 1 になる。 一度セットされると、μ PD77210 ファミリが 0 を書き込むまで変化しない。	L/S
1	SSEF	SDT ストア許可フラグ ・SDT の値をシリアル出力用シフト・レジスタに転送した場合に 1 になる。 ・μ PD77210 ファミリが SDT にデータをストアすると 0 になる	L
0	SLEF ^{注3}	SDT ロード許可フラグ ・シリアル入力用シフト・レジスタの値を SDT に転送した場合に 1 になる。 ・μ PD77210 ファミリが SDT からデータをロードすると 0 になる	L

注 1. シリアル・リセット・ビットを 1 にしたあと 3 サイクルの間は TSIO へのアクセスはできません。アクセスした場合、正常に結果が反映されません。

SST のほかのビットへの設定と同時にシリアル・リセット・ビットを 1 にすることを禁止します。

- ★ 2. ストア・エラー・フラグ/ロード・エラー・フラグが 1 のとき、TSDT へのストア/ロードを実行しても、シリアル割り込みは発生しません。シリアル・ステータス・レジスタのストア・エラー・フラグ/ロード・エラー・フラグを直接クリア(0)にしてから使用してください(シリアル・ステータス・レジスタのリセットによるクリアは使用しないでください)。
- ★ 3. シリアル・インタフェースのシリアル・データ・レジスタ・ロード許可フラグが許可(0)のとき、FINT 命令を実行すると、それ以降のロード許可の割り込みを受け付けなくなります。

表 5-5 SICM ビットと SIEF ビットの組み合わせ

SICM	SIEF	機 能
0	0	状態遷移モード ・シリアル入力を行わないときもこのモードになる。
1	0	シリアル連続入力モード
0	1	シリアル単発入力モード ・1 にセットされた SIEF は次の命令サイクルで自動的に 0 にリセットされる。
1	1	設定禁止

表 5-6 TFMT のビット構成

ビット	名称	機能	ロード/ ストア
15-13	SRTX	送信スロット位置設定ビット ・ 000 : 0~31 スロット (デフォルト) ・ 001 : 16~47 スロット ・ 010 : 32~63 スロット ・ 011 : 48~79 スロット ・ 100 : 64~95 スロット ・ 101 : 80~111 スロット ・ 110 : 96~127 スロット ・ 111 : 112~143 スロット	L/S
12-8	FLTX	送信スロット数設定ビット ・ 00000 : 32 スロット/フレーム (デフォルト) ・ 00001~11111 : 1~31 スロット/フレーム	L/S
7-5	SRRX	受信スロット位置設定ビット ・ 000 : 0~31 スロット (デフォルト) ・ 001 : 16~47 スロット ・ 010 : 32~63 スロット ・ 011 : 48~79 スロット ・ 100 : 64~95 スロット ・ 101 : 80~111 スロット ・ 110 : 96~127 スロット ・ 111 : 112~143 スロット	L/S
4-0	FLRX	受信スロット数設定ビット ・ 00000 : 32 スロット/フレーム (デフォルト) ・ 00001~11111 : 1~31 スロット/フレーム	L/S

5.3.3 時分割シリアル・インタフェースのタイミング

1つの入力ポートと出力ポートを、複数の DSP で時分割して使用する方法です。

シリアル・データを 8 ビットまたは 16 ビットずつ区切り、スロットを定義します。スロット数は 1 つ以上 32 個以下に設定できます。拡張仕様として、最大スロット数は 128 まで設定できます。この場合、フラグを設定できるのは任意の 16 の倍数のスロットから始まる 32 スロットのみに制限されます。

スロットと転送のイメージを図 5-3に、スロット数の拡張のイメージを図 5-4に示します。

注意 1 フレーム内では最大 128 スロット (Slot 0~127) までを管理します。このため、図 5-4において Slot region 7 のように次のフレームへとまたがるようなスロット定義では、スロット 128 以降の使用を禁止します。

図 5-3 スロット

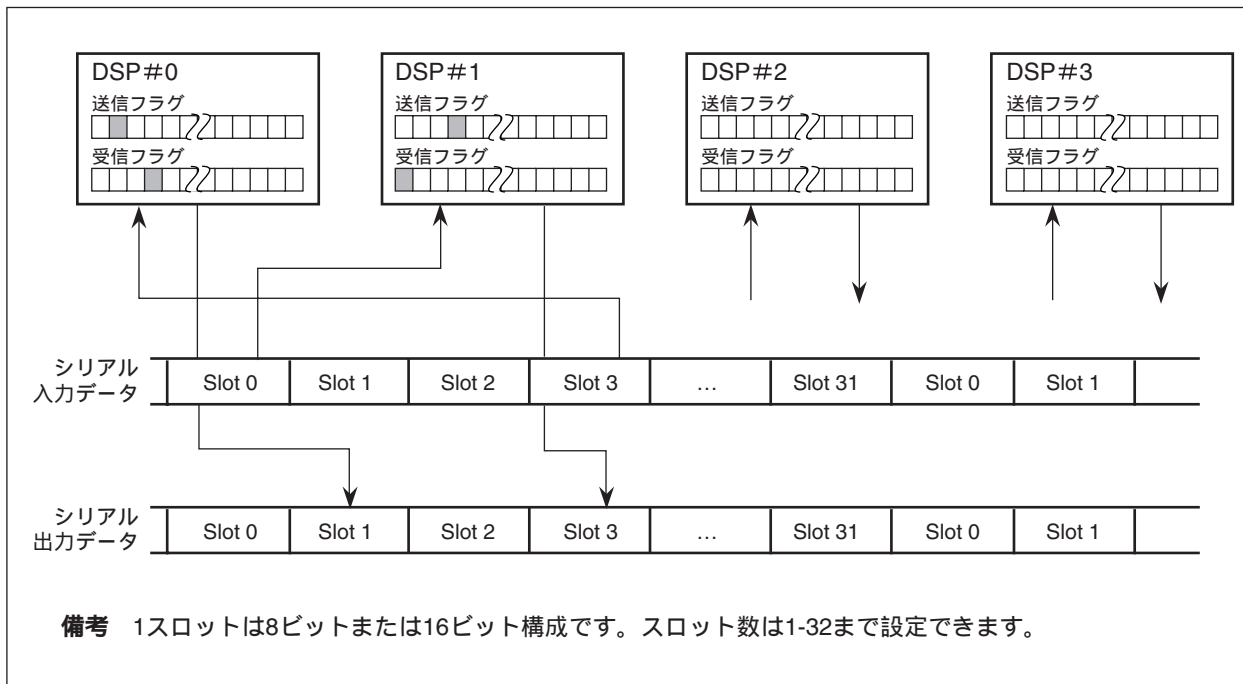
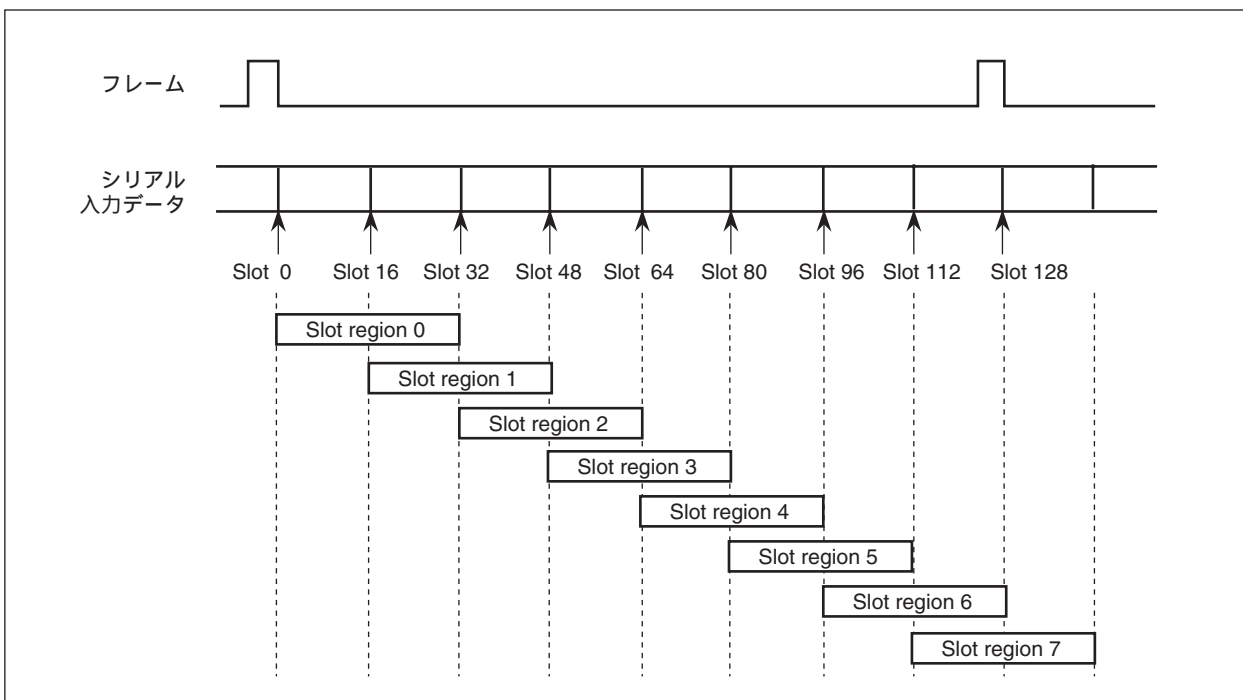


図 5-4 スロット拡張



シリアル入力とシリアル出力を別々に使用すれば、1チャンネルのTSIOに対して、時分割データ入力系と出力系を定義できます。ただし、クロック周波数は同じなので、データ・レートは入出力とも一緒になります。

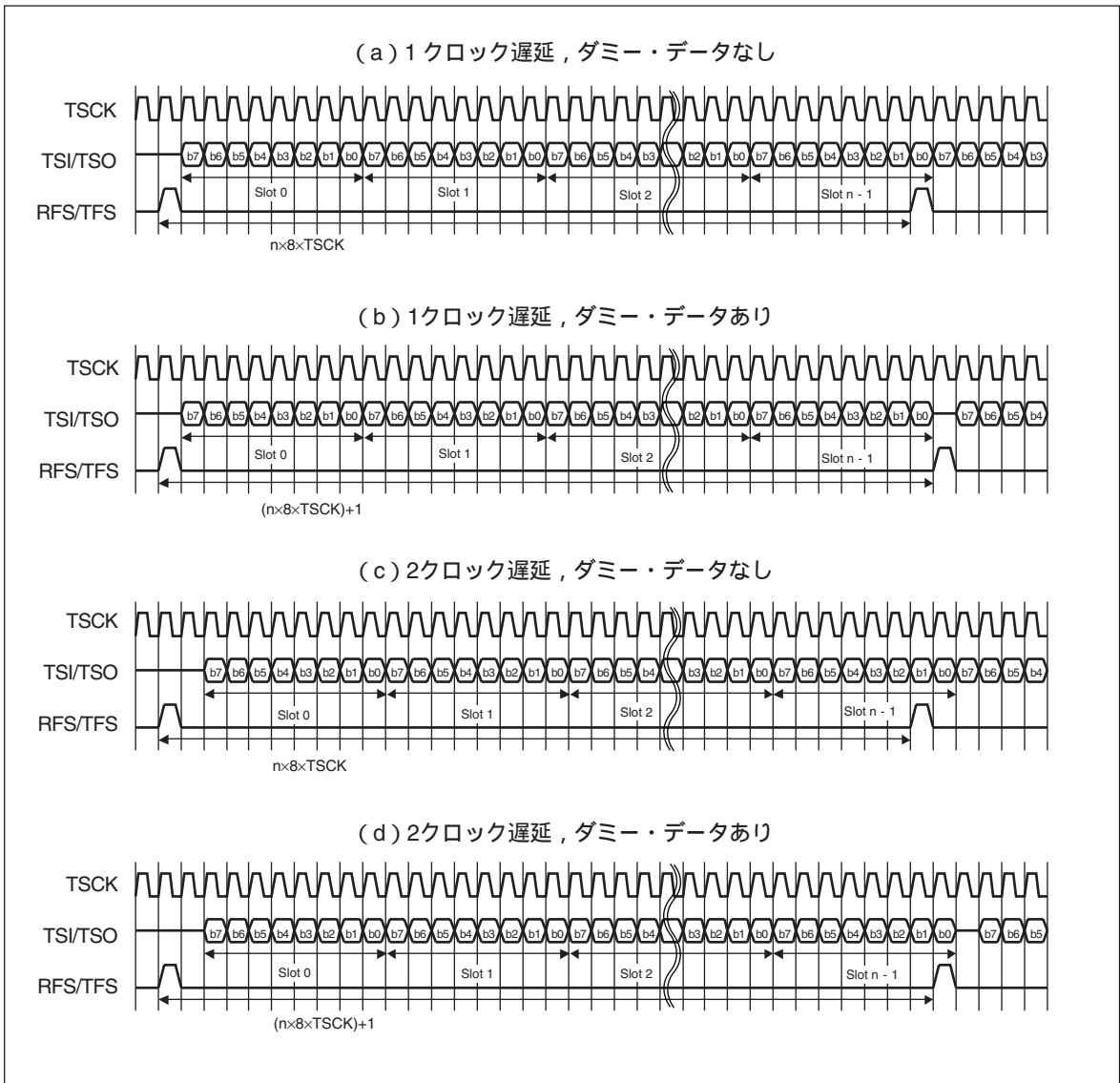
どのスロットからデータを取り出すか、どのスロットにデータを置くかを、それぞれ32ビットの送信フラグと受信フラグに定義できます。また、複数のフラグを設定することも可能です。フラグをセットしたスロットからデータを取り込んだり、データを送信したりします。

時分割のシリアル・データが送られてくる場合に、必要なデータを取り出して入力したり、時分割データを生成するため、必要なタイミングでデータを出力することができます。

TDM機能では、規格であるT1 frameやE1 frame信号が扱いやすいようになっています。

転送タイミングを図 5-5に示します。

図 5-5 時分割シリアル・インタフェースのタイミング



シリアル信号のデータの始まりは、フレーム信号の次のクロックが最初のデータ（標準的なシリアル信号やE1対応）となるタイミングと、フレーム信号の次の次のクロックが最初のデータ（T1対応）となるタイミングを切り替えます。

1フレームの長さは、(8の倍数)ビットと、(8の倍数)+1ビットに切り替えられます。後者は、T1のフレーム・ビットの読み飛ばしを実現するためのものです。1ビットのダミー・ビットを挿入します。

以上の組み合わせで4通りのタイミングが構成されます。図 5-5の(a)はフレームの次のクロックからデータを開始し、ダミー・ビットがない場合、(b)はフレームの次のクロックからデータを開始し、1ビットのダミー・ビットを挿入する場合、(c)はフレームの2クロック後からデータを開始し、ダミー・ビットがない場合、(d)はフレームの2クロック後からデータを開始し、ダミー・ビットを挿入する場合です。E1の信号を扱うには(a)のタイミングで32スロットに設定し、T1の信号を扱うには(d)のタイミングで24スロットに設定してください。

フレーム信号は、図 5-5に示したようなRTZのパルスと、これを反転したRTOのパルスの両方をサポートします。

備考 E1は32スロット×8ビットの256ビット/フレーム(欧州標準),T1は24スロット×8ビット+1の193ビット/フレーム(米国,日本標準)です。

★ 5.3.4 シリアル・ステータス・レジスタのリセット機能について

シリアル入出力に関するフラグの初期化,内部カウンタの初期化を行ないませんが,入出力シフト・レジスタの初期化は行ないません。

割り込みを含む内部同期信号は内部カウンタを基準にしており,場合によってはリセットによる同期ずれが発生するおそれがあります。

シリアル・ステータス・レジスタのリセットは,使用しないでください。

リセット解除後,TSIOの初期化シーケンスの先頭でシリアル・ステータス・レジスタのリセットをすることは問題ありません。

5.4 オーディオ・シリアル・インタフェース(ASIO)

オーディオ・シリアル・インタフェース(ASIO)は,標準的な音声コーデックと直接接続が可能なシリアル・インタフェース(スタンダード・モード)であるとともに,32/64ビットのステレオ・オーディオ信号の入出力が可能なオーディオ・シリアル・インタフェース持つ周辺回路です。

オーディオ・シリアル・インタフェースには2つのモードがあり,μPD77210ファミリからオーディオ・コーデックにシリアル・クロックを供給するマスタ・モードと,オーディオ・コーデックからμPD77210ファミリにシリアル・クロックを供給するスレーブ・モードをサポートしています。

スタンダード・モードについては5.5 スタンダード・シリアル・インタフェース(SIO)を参照してください。主な特徴を次に示します。

- シリアル・クロック供給

マスタ・モード

MCLK 端子: マスタ・クロック(入力)

BCLK 端子: オーディオ・シリアル・クロック(出力)

LRCLK 端子: レフト/ライト・クロック(出力)

スレーブ・モード

MCLK 端子: 未使用(チップ外部でプルダウン)

BCLK 端子: オーディオ・シリアル・クロック(入力)

LRCLK 端子: レフト/ライト・クロック(入力)

- ビット長

入力/出力独立に,32ビットまたは64ビットを選択

- 内部ハンドシェイク

ポーリング,ウエイトまたは割り込みによるハンドシェイク

オーディオ・シリアル・インタフェースのブロック図を図 5-6に,レジスタ群を表 5-7に示します。

図 5-6 オーディオ・シリアル・インタフェースのブロック図

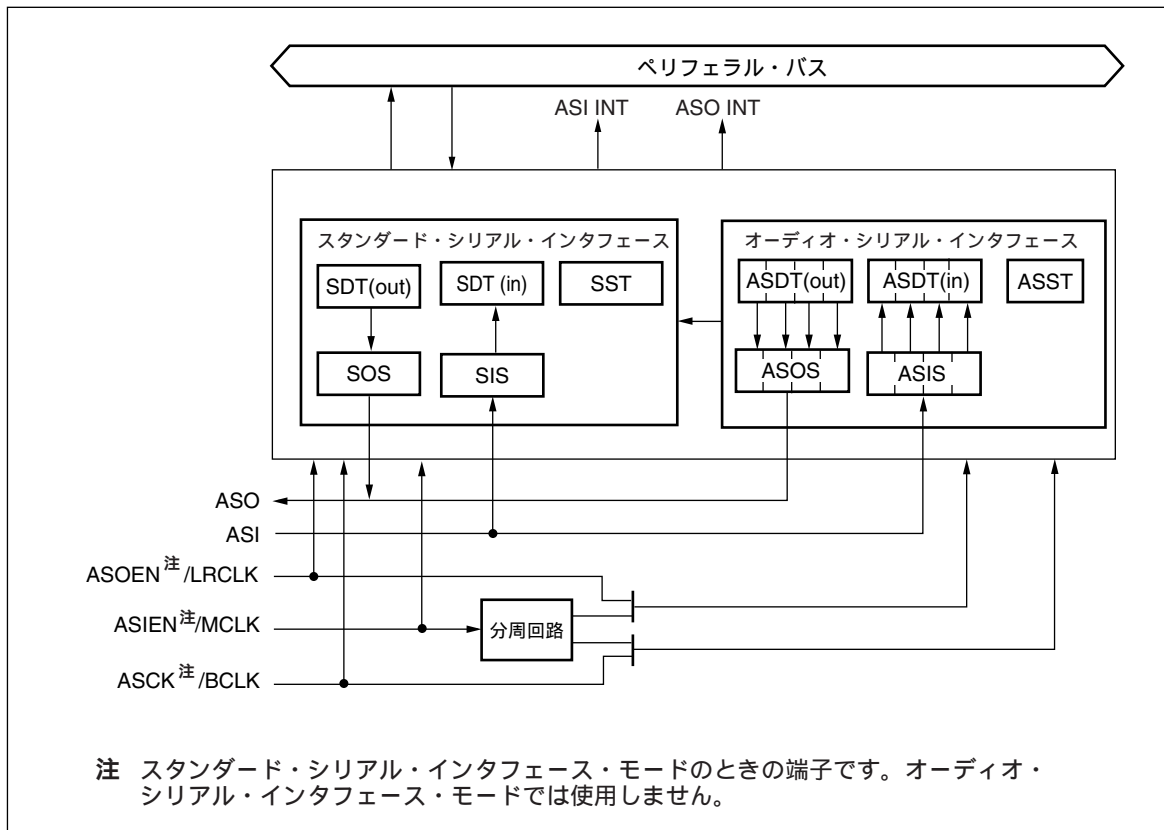


表 5-7 オーディオ・シリアル・インタフェースのレジスタ

X/Y メモリ・アドレス	レジスタ名	機能	ロード/ストア
0x3810	ASDT	オーディオ・シリアル・データ・レジスタ	L/S
0x3811	SST2	シリアル・ステータス・レジスタ 2	L/S
0x3812	ASST	オーディオ・シリアル I/F 設定レジスタ	L/S

5.4.1 オーディオ・シリアル・インタフェースの端子

(1) BCLK (シリアル・ビット・クロック - 入出力)

オーディオ・シリアル・データ入出力用のクロック端子です。マスタ・モードのときに出力、スレーブ・モードのときに入力（デフォルト）となります。

マスタ・モードでは、MCLK 端子から入力したマスタ・クロックを分周した BCLK クロックを出力し、オーディオ・コーデックの BCLK 端子に供給します。

スレーブ・モードでは入力された BCLK クロックを直接シリアル・クロックとして使用します。

シリアル・データの入出力、各シリアル・インタフェース用信号の出力およびサンプリングは BCLK に同期して行います。

(2) LRCLK (レフト・ライト・クロック - 入出力)

オーディオ・シリアル・データのフレーム同期用の信号端子です。マスタ・モードのときに出力、スレーブ・モードのときに入力となります。

オーディオ・シリアル・データの 32 ビットあるいは 64 ビットに同期し、BCLK の 32 周期あるいは 64 周期が LRCLK の 1 周期となります。

マスタ・モードでは、MCLK 端子から入力したマスタ・クロックを分周した LRCLK クロックを出力し、オーディオ・コーデックの LRCLK 端子に供給します。

スレーブ・モードでは、入力された LRCLK クロックを直接フレーム・クロックとして使用します。

シリアル・データの入出力、各シリアル・インタフェース用信号の出力およびサンプリングは BCLK に同期して行います。

(3) MCLK (マスタ・クロック - 入力)

BCLK と LRCLK のマスタ・クロックとなる入力端子です。マスタ・モードのときのみを使用し、スレーブ・モードでは未使用（プルダウン処理を推奨）です。

MCLK 端子 から入力したクロックを分周して、BCLK と LRCLK をオーディオ・シリアル・インタフェース内部で生成します。

(4) ASI (シリアル・データ入力 - 入力)

オーディオ・シリアル・データの入力端子です。

BCLK の立ち上がりに同期してサンプリングします。

(5) ASO (シリアル・データ出力 - 出力)

オーディオ・シリアル・データの出力端子です。

BCLK の立ち下がりに同期して出力が変化します。

5.4.2 オーディオ・シリアル・インタフェースのレジスタ

(1) ASDT (オーディオ・シリアル・データ・レジスタ : Audio Serial Data Transfer Register)

オーディオ・シリアル・データを入出力するための 64 ビット・レジスタです。出力用と入力用のレジスタを別々に持っています。

SDT (out) は、出力するデータを設定する 64 ビットのレジスタです。ASDT に対するストア命令を実行すると SDT (out) にペリフェラル・バスからデータを入力します。

シリアル出力シフト・レジスタ (SOS) が空になると ASDT の値を SOS に設定し、ASO 端子から MSB ファーストで出力します。ASSEF フラグが「0」のときに ASDT に対するストア命令を実行すると、ASSER フラグが「1」になります (ストア・エラー)。

64 ビットのデータ転送を行うときは、ASDT へ 16 ビット・データを 4 回に分けてストアします。4 回のストアで、ビット 63:48、ビット 47:32、ビット 31:16、ビット 15:0 の順に書き込まれます。

32 ビットのデータ転送を行うときは、ASDT へ 16 ビット・データを 2 回に分けてストアします。2 回のストアで、ビット 31:16、ビット 15:0 の順に書き込まれます。

ASSEF フラグは 4 回 (64 ビット転送) または 2 回 (32 ビット転送) のストアが行われたあとに「0」(ストア不許可状態) になります。

SDT (in) は、入力されたデータを読み出す 64 ビットのレジスタです。ASDT に対してロード命令を実行すると、ペリフェラル・バスにデータを出力します。

ASI 端子からは MSB ファーストで入力し、シリアル入力シフト・レジスタ (SIS) に最後のビットが入力されると、SIS の値を ASDT に設定します。ASLEF フラグが「0」のときに ASDT からのロード命令を実行すると、ASLER フラグが「1」になります (ロード・エラー)。

64 ビットのデータ転送を行うときは、SDT から 16 ビット・データを 4 回に分けてロードします。4 回のロードで、ビット 63:48、ビット 47:32、ビット 31:16、ビット 15:0 の順に読み出されます。

32 ビットのデータ転送を行うときは、ASDT から 16 ビット・データを 2 回に分けてロードします。2 回のロードで、ビット 31:16、ビット 15:0 の順に読み出されます。

ASLEF フラグは 4 回 (64 ビット転送) または 2 回 (32 ビット転送) のロードが行われたあとに「0」(ロード不許可状態) になります。

(2) ASST (オーディオ・シリアル・ステータス・レジスタ : Audio Serial StaTus Register)

オーディオ・シリアル入出力モードの設定と、ステータスを表す 16 ビットのレジスタです。スタンダード / オーディオ・シリアル・インタフェース・モードの切り替え、マスタ・クロック分周比設定など、 μ PD77210 ファミリの DSP コア・カーネル部分とのインタフェースの指定やオーバーラン、アンダランの表示を行います。

ASST のビット構成については表 5-8 に示します。リセット時の値は、0x8012 です。

(3) SOS (シリアル出力シフト・レジスタ : Serial Output Shift Register)

オーディオ・シリアル・データを ASO 端子から出力しながらシフトする 64 ビットのシフト・レジスタです。指定したビット数の出力が終わると ASDT から新しいデータを入力します。

SOS は、ペリフェラル・バスとは接続していません。

(4) SIS (シリアル入力シフト・レジスタ : Serial Input Shift Register)

ASI 端子から入力されるビット列を入力する 64 ビットのシフト・レジスタです。指定したビット数の入力が終わると、ASDT にデータを出力します。

SIS は、ペリフェラル・バスとは接続していません。

(5) SST1 (シリアル・ステータス・レジスタ : Serial StaTus Register)

オーディオ・シリアル・インタフェース・モードのときは使用しません。

表 5-8 ASST のビット構成 (1/2)

ビット	名称	機能	ロード / ストア
15	SOAD	スタンダード / オーディオ・シリアル・インタフェース切り替えビット ・0 : スタンダード・シリアル ・1 : オーディオ・シリアル (デフォルト)	L/S
14	ASOEN ^{注1}	オーディオ・シリアル出力使用許可 / 不許可ビット ・0 : 出力使用不許可 (デフォルト) ・1 : 出力使用許可	L/S
13	ASIEN	オーディオ・シリアル入力使用許可 / 不許可ビット ・0 : 入力使用不許可 (デフォルト) ・1 : 入力使用許可	L/S
12-10	BDV ^{注2}	マスタ・クロックから BCLK/LRCLK を生成する時の分周比設定ビット ・000 : 4 分周 / 256 分周 (デフォルト) ・001 : 8 分周 / 256 分周 ・010 : 6 分周 / 384 分周 ・011 : 12 分周 / 384 分周 ・100 : 8 分周 / 512 分周 ・101 : 16 分周 / 512 分周	L/S
9	ADRST ^{注3}	オーディオ・シリアル・リセット許可ビット 0 : (デフォルト) 1 : オーディオ・シリアル入出力リセット要求 入出力回路をリセットし, ASSER=0, ASLER=0, ASSEF=1, ASLEF=0 とする。ADRST はリセット後, 自動的に 0 にクリアされる。	L/S
8	ASSWE	ASDT ストア・ウェイト許可ビット ・0 : ストア・ウェイトを使用しない (デフォルト) ・1 : ストア・ウェイトを使用する。 SDT (out) にデータが残っている状況でストアした場合に, μ PD77210 ファミリにウェイトを挿入する。	L/S
7	ASLWE	ASDT ロード・ウェイト許可ビット ・0 : ロード・ウェイトを使用しない。(デフォルト) ・1 : ロード・ウェイトを使用する。 SDT (in) にデータがない状況でロードした場合に, μ PD77210 ファミリにウェイトを挿入する。	L/S
6	ADOBL	オーディオ・データ出力 1 フレーム・ビット長設定ビット ・0 : 64 ビット (デフォルト) ・1 : 32 ビット	L/S

- ★ 注 1. マスタ・モードにてオーディオの出力をする場合, BCLK/LRCLK の供給 (ASST MSSEL = 0) と同時に ASOEN を 1 にしてください。また, それ以降は ASST のリセットや, ASOEN による出力制御などは同期がずれるおそれがあるため, 使用しないでください。
2. マスタ・モードへの変更時に値を設定してください。マスタ・モード時に再設定 (変更) を行うことはできません。マスタ・モード時に変更した場合は, ASIO は正常に動作しません。
3. シリアル・リセット・ビットを 1 にしたあと 3 サイクルの間は ASIO へのアクセスを実行できません。アクセスした場合, 結果は正常に反映されません。
- ASST のほかのビットへの設定と同時にシリアル・リセット・ビットを 1 にしないでください。

表 5-8 ASST のビット構成 (2/2)

ビット	名称	機能	ロード/ ストア
5	ADIBL	オーディオ・データ入力1フレーム・ビット長設定ビット ・0: 64 ビット (デフォルト) ・1: 32 ビット	L/S
4	MSSEL ^{注1}	オーディオ・シリアル・クロック・モード設定ビット ・0: マスタ・モード ・1: スレーブ・モード (デフォルト)	L/S
3	ASSER ^{注2}	ASDT ストア・エラー・フラグ ・0: エラーなし (デフォルト) ・1: エラー ASSEF が 0 のときに μ PD77210 ファミリが ASDT に書き込んだ場合に 1 になる。 一度セットされると、 μ PD77210 ファミリが 0 を書き込むまで変化しない。	L/S
2	ASLER ^{注2}	ASDT ロード・エラー・フラグ ・0: エラーなし (デフォルト) ・1: エラー ASLEF が 0 のときに μ PD77210 ファミリが ASDT を読み出した場合に 1 になる。 一度セットされると、 μ PD77210 ファミリが 0 を書き込むまで変化しない。	L/S
1	ASSEF	ASDT ストア許可フラグ ・ASDT の値を SOS に転送した場合に 1 になる (デフォルト) ・ μ PD77210 ファミリが ASDT に書き込むと 0 になる	L
0	ASLEF ^{注3}	ASDT ロード許可フラグ ・SIS の値を ASDT に転送した場合に 1 になる。 ・ μ PD77210 ファミリが ASDT を読み出すと 0 になる (デフォルト)	L

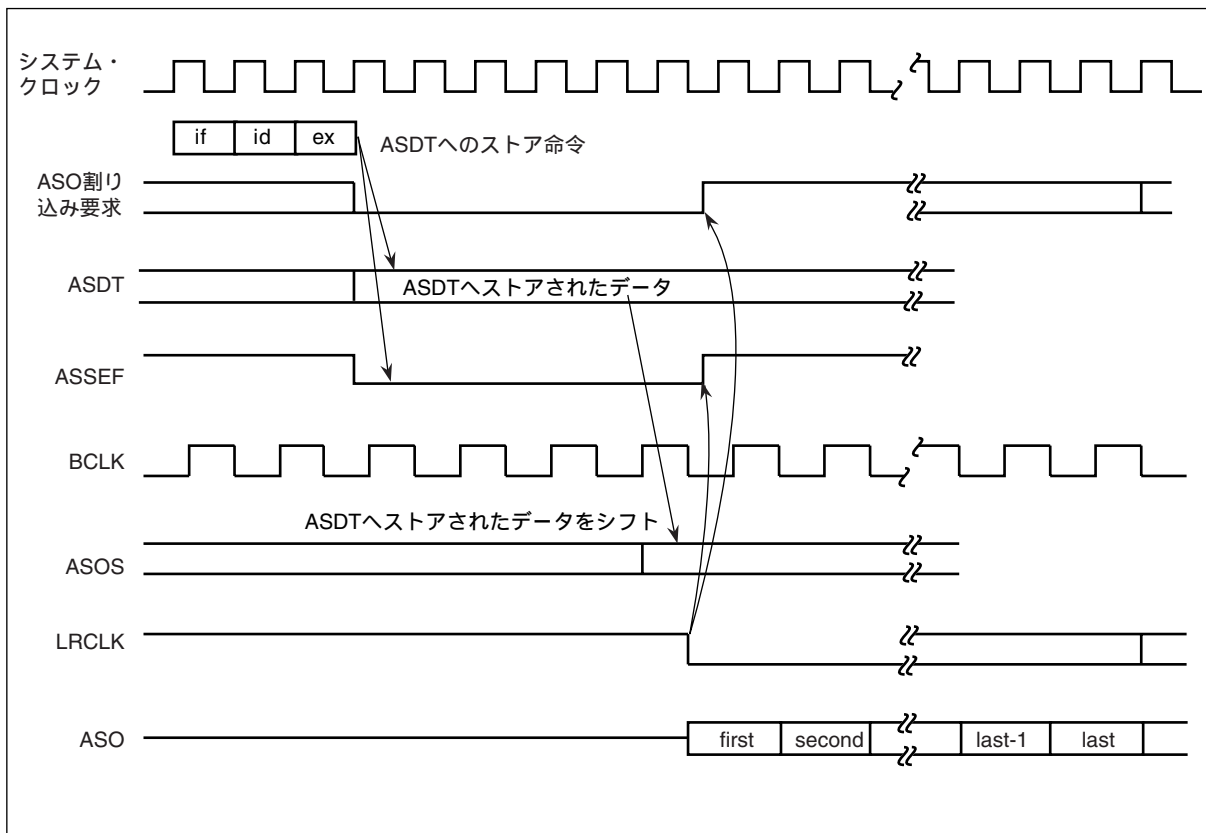
注 1. マスタ・モードに変更後、再度スレーブ・モードに変更しないでください。変更した場合は、ASIO は正常に動作しません。

- ★ 2. ストア・エラー・フラグ/ロード・エラー・フラグが 1 のとき、ASDT へのストア/ロードを実行しても、シリアル割り込みは発生しません。シリアル・ステータス・レジスタのストア・エラー・フラグ/ロード・エラー・フラグを直接クリア (0) にしてから使用してください (シリアル・ステータス・レジスタのリセットによるクリアは使用しないでください)。
- ★ 3. シリアル・インタフェースのシリアル・データ・レジスタ・ロード許可フラグが許可 (0) のとき、FINT 命令を実行すると、それ以降のロード許可の割り込みを受け付けなくなります。

5.4.3 オーディオ・シリアル・インタフェースのタイミング

オーディオ・シリアル出力タイミングを図 5-7に示します。

図 5-7 オーディオ・シリアル・インタフェースの出力タイミング



ASDT に対して、32 ビット・モード時は 2 回のストア、64 ビット・モード時は 4 回のストアにより、割り込みおよびストア許可フラグがインアクティブとなります。

ASDT にストアしたデータは、LRCLK の立ち下がり後の BCLK 立ち上がり時に ASDT から ASOS へ転送され、BCLK 立ち下がりに同期して MSB ファーストで ASO より出力されます。

LRCLK の立ち下がり後、最初のシステム・クロック CLK の立ち上がりタイミングでストア許可フラグおよび割り込み信号がアクティブとなり、次のシリアル出力データを要求します。

- ★ ASST ASOEN = 1 (オーディオ・シリアル出力許可) および、HA2R ASOQ = 1 (ドライバはドライブオン[※]) の状態で、LRCLK1 フレームの間に ASDT へのストア・アクセスがない場合は、ASO はハイ・インピーダンスにはなりません。ASO は、ASOS に格納されているデータを LRCLK 立ち下がりに同期して出力を繰り返します。

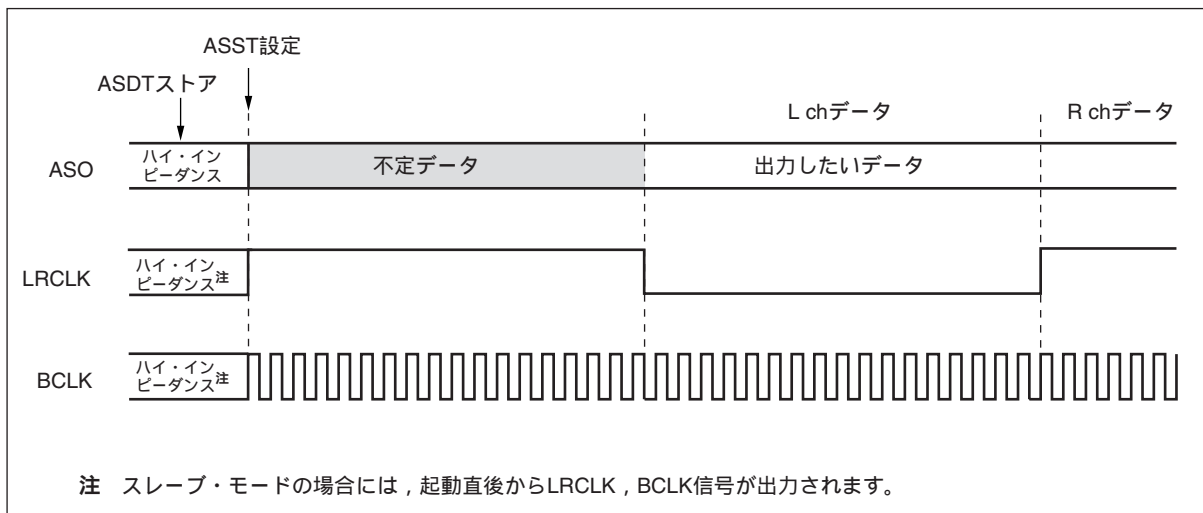
オーディオ・シリアル入力タイミングを図 5-8に示します。

注 μPD77214 のみ

5.4.4 起動時の ASIO に関する注意事項

μ PD77210 ファミリを起動後、ASIO 動作開始時に、ASO 端子から想定されない不定データが出力されます。これは、電源投入直後には、ASOS レジスタは初期化されず、その内容が不定となるためです。これにより、ASDT レジスタに出力したいデータをストアしても、LRCLK の立ち下がり前に、不定データが出力されてしまいます。この不定データはマスタ・モード時、スレーブ・モード時のいずれの場合にも出力されます。この現象の回避策はありません。ASO を使用するときは、コーデック側でミュートをかけるなど、適切な処置をしてください。

図 5-9 ASIO 動作開始時の ASO 動作 (マスタ・モードの場合)



★ 5.4.5 オーディオ・シリアル・ステータス・レジスタのリセット機能について

シリアル入出力に関するフラグの初期化、内部カウンタの初期化を行ないませんが、入出力シフト・レジスタの初期化は行ないません。

マスタ・モードにおける ASO 割り込みは内部カウンタを基準にしており、場合によってはリセットによる同期ずれが発生するおそれがあります。

オーディオ・シリアル・ステータス・レジスタのリセットは、使用しないでください。

リセット解除後、ASIO の初期化シーケンスの先頭でオーディオ・シリアル・ステータス・レジスタのリセットをすることは問題ありません。

5.5 スタンダード・シリアル・インタフェース (SIO)

スタンダード・シリアル・インタフェース (SIO) は、従来の μ PD7701x ファミリおよび μ PD77111 ファミリと互換性のあるシリアル・インタフェースです。オーディオ・シリアル・インタフェースあるいはTDMシリアル・インタフェースの設定を切り替えることによってこのモードになります。

主な特徴を次に示します。

- シリアル・クロック供給
外部からの供給可，入力，出力でクロック共用
- ビット長
8ビット，16ビット設定，MSB ファースト，LSB ファーストのフォーマット選択可能
- ポーリング，ウェイト，および割り込みによるハンドシェークが可能
- 外部ハンドシェーク
専用ステータス信号 (TSIAK，TSORQ) によるハンドシェークが可能

スタンダード・シリアル・インタフェースのブロック図を図 5-10に，レジスタ群を表 5-9に示します。

図 5-10 スタンダード・シリアル・インタフェース

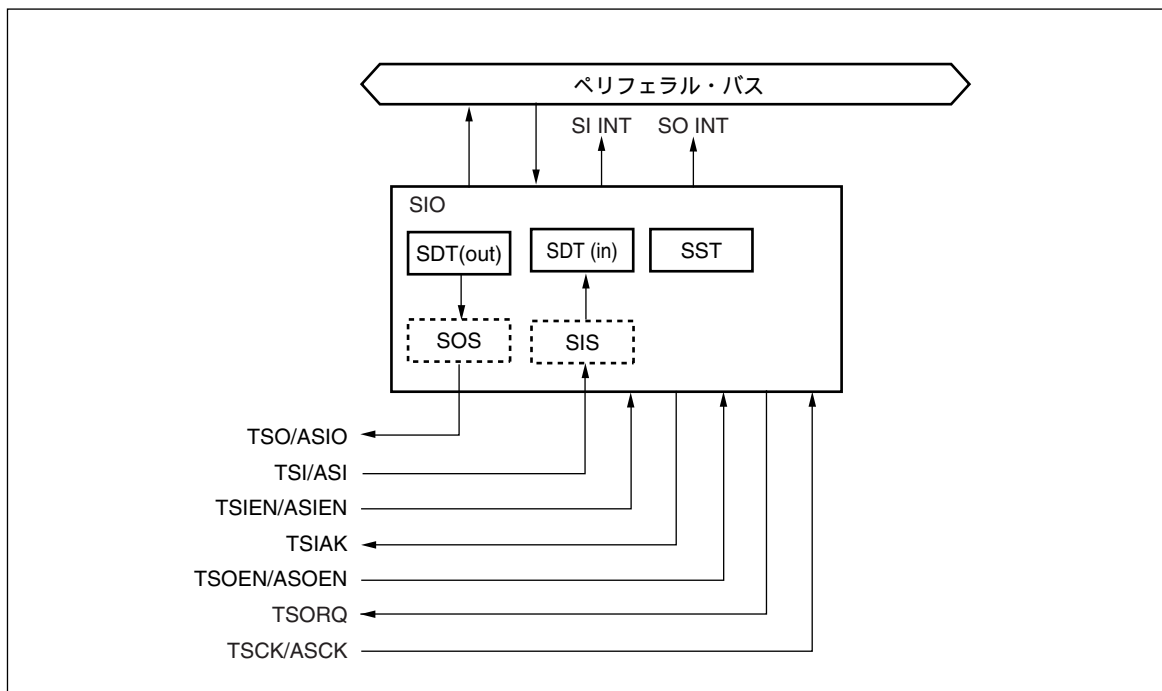


表 5-9 スタンダード・シリアル・インタフェースのレジスタ

X/Y メモリ・アドレス	レジスタ名	機能	ロード/ストア
0x3800	SDT1	シリアル・データ・レジスタ 1	L/S
0x3801	SST1	シリアル・ステータス・レジスタ 1	L/S
0x3810	SDT2	シリアル・データ・レジスタ 2	L/S
0x3811	SST2	シリアル・ステータス・レジスタ 2	L/S

5.5.1 スタンダード・シリアル・インタフェースの端子

(1) TSCK / ASCK (シリアル・クロック - 入力)

スタンダード・シリアル・データ入出力用のクロック端子です。

シリアル・データの入出力，各種シリアル・インタフェース用信号の出力およびサンプリングは TSCK に同期して行います。

(2) TSOEN / ASOEN (シリアル出力イネーブル - 入力)

シリアル・データの出力許可信号の入力端子です。

TSCK の立ち下がりに同期してサンプリングします。

外部デバイスが，シリアル出力データを入力する用意が整ったときにアクティブ (ハイ・レベル) にします。

(3) TSIEN / ASIEN (スタンダード・シリアル入力イネーブル - 入力)

シリアル・データの入力許可信号の入力端子です。

TSCK の立ち下がりに同期してサンプリングします。

外部デバイスが，シリアル入力データを出力する用意が整ったときにアクティブ (ハイ・レベル) にします。

(4) TSORQ (スタンダード・シリアル出力リクエスト - 出力)

シリアル・データの出力要求信号の出力端子です。

TSCK の立ち上がりに同期して出力が変化します。

シリアル・データを SOS に書き込むとアクティブ (ハイ・レベル) になります。シリアル出力を開始するとインアクティブ (ロウ・レベル) になります。

オーディオ・シリアル・インタフェースをスタンダード・シリアル・インタフェースとして使用する場合は，この端子が存在しません。

(5) TSIK (スタンダード・シリアル入力アクノリッジ - 出力)

シリアル・データの入力受け付け信号の出力端子です。

TSCK の立ち上がりに同期して出力が変化します。

シリアル入力が可能になるとアクティブ (ハイ・レベル) になります。シリアル入力を開始するとインアクティブ (ロウ・レベル) になります。

オーディオ・シリアル・インタフェースをスタンダード・シリアル・インタフェースとして使用する場合は，この端子が存在しません。

(6) TSI / ASI (スタンダード・シリアル・データ入力 - 入力)

シリアル・データの入力端子です。

TSCK/ASCK の立ち下がりに同期してサンプリングします。

(7) TSO / ASO (スタンダード・シリアル・データ出力 - 出力)

シリアル・データの出力端子です。

TSCK/ASCK の立ち上がりに同期して出力が変化します。

5.5.2 スタンダード・シリアル・インタフェースのレジスタ

(1) SDT (シリアル・データ・レジスタ : Serial Data Transfer Register)

スタンダード・シリアル・データを入出力するための 16 ビット・レジスタです。出力用と入力用のレジスタを別々に持っています。

SDT (out) は、出力するデータを設定する 16 ビットのレジスタです。SDT に対するストア命令を実行すると SDT (out) にペリフェラル・バスからデータを入力します。

シリアル出力シフト・レジスタ (SOS) が空になると SDT の値を SOS に設定し、TSO/ASO 端子から出力します。SSEF フラグが「0」のときに SDT に対するストア命令を実行すると、SSER フラグが「1」になります (ストア・エラー)。

SDT (in) は、入力されたデータを読み出す 16 ビットのレジスタです。SDT に対してロード命令を実行すると、ペリフェラル・バスにデータを出力します。

TSI/ASI 端子から入力し、シリアル入力シフト・レジスタ (SIS) に最後のビットが入力されると、SIS の値を SDT に設定します。SLEF フラグが「0」のときに SDT からのロード命令を実行すると、SLER フラグが「1」になります (ロード・エラー)。

(2) SST1 (シリアル・ステータス・レジスタ : Serial StaTus Register)

シリアル入出力モードの設定と、ステータスを表す 16 ビットのレジスタです。MSB/LSB ファーストの設定、

μ PD77210 ファミリの DSP コア・カーネル部分とのインタフェースの指定やオーバラン、アンダランの表示を行います。

★ SST のビット構成については表 5-11に示します。リセット時の値は、0x0002 です。

表 5-10 シリアル入出力エラー・フラグ設定条件

エラー・フラグ名	設定条件	リセット条件
SSER	SSEF = 0 のときに SDT にストア	ハードウェア・リセットまたは、プログラムによるリセット
SLER	SLEF = 0 のときに SDT からロード	

シリアル出力モードの変更：

シリアル出力のモード (データ長：8/16 ビット、LSB/MSB ファーストなど) はデータを SDT (out) にストアしたときの SST の設定によって決定されます。

SSEF = 0 (SDT (out) にデータが存在) のときは SST の値を変更しないでください。

SST の値の変更は SSEF = 1 (SDT (out) が空の状態)で行ってください。

シリアル入力モードの変更：

シリアル入力が行われている状態で SST の値を変更しないでください。

シリアル連続入力モードが設定されている場合 (SICM = 1) は、SLEF = 1 の状態で SICM = 0 としてからシリアル入力のモード (データ長：8/16 ビット、LSB/MSB ファーストなど) を変更し、その後再び SICM = 1 にしてください。

SST の値の変更が有効になるのは、SDT (in) および SIS に入力されている 2 つのデータをロードしたあとの入力データからです。

(3) SOS (シリアル出力シフト・レジスタ : Serial Output Shift Register)

シリアル・データを TSO/ASO 端子から出力しながらシフトする 16 ビットのシフト・レジスタです。指定したビット数の出力が終わると SDT から新しいデータを入力します。

SOS は、ペリフェラル・バスとは接続していません。

(4) SIS (シリアル入力シフト・レジスタ : Serial Input Shift Register)

TSI/ASI 端子から入力されるビット列を入力する 16 ビットのシフト・レジスタです。指定したビット数の入力が終わると、SDT にデータを出力します。

SIS は、ペリフェラル・バスとは接続していません。

表 5-11 SST のビット構成 (1/2)

ビット	名称	機能	ロード/ ストア
15	SOTF	シリアル出力転送フォーマット設定ビット ・0: MSB ファーストでシリアル出力 (デフォルト) ・1: LSB ファーストでシリアル出力。	L/S
14	SITF	シリアル入力転送フォーマット設定ビット ・0: MSB ファーストでシリアル入力 (デフォルト) ・1: LSB ファーストでシリアル入力。	L/S
13	SOBL	シリアル出力ワード長設定ビット ・0: 16 ビットでシリアル出力 (デフォルト) ・1: 8 ビットでシリアル出力。	L/S
12	SIBL	シリアル入力ワード長設定ビット ・0: 16 ビットでシリアル入力 (デフォルト) ・1: 8 ビットでシリアル入力。	L/S
11	SSWE	SDT ストア・ウエイト許可ビット ・0: ストア・ウエイトを使用しない (デフォルト) ・1: ストア・ウエイトを使用する。 SSEF が 0 のときに、 μ PD77210 ファミリが SDT にデータをストアしようとした場合に、ウエイトを挿入する。	L/S
10	SLWE	SDT ロード・ウエイト許可ビット ・0: ロード・ウエイトを使用しない (デフォルト) ・1: ロード・ウエイトを使用する。 SLEF が 0 のときに、 μ PD77210 ファミリが SDT からデータをロードしようとした場合に、ウエイトを挿入する。	L/S
9	SICM	シリアル入力連続モード設定フラグ ・0: 実行中のシリアル入力処理終了後、シリアル入力モードが単発モードになる。 ・1: シリアル入力モードが連続モードになり、シリアル入力処理を開始する。	L/S
8	SIEF	シリアル単発入力許可フラグ ・0: (デフォルト) ・1: シリアル入力単発モードでシリアル入力処理を開始する (1 回のみ)。 1 にセットした SIEF は次の命令サイクルで自動的にリセットする。	L/S
7 ^注	SOE	シリアル出力許可ビット ・0: シリアル出力使用不可 ・1: シリアル出力使用可 (デフォルト)	L/S
6 ^注	SIE	シリアル入力許可ビット ・0: シリアル入力使用不可 ・1: シリアル入力使用可 (デフォルト)	L/S

注 ビット 6, 7 は TSIO 側の SIO (SST1) のみの機能です。ASIO 側の SIO (SST2) では「予約ビット」になります。

表 5-11 SST のビット構成 (2/2)

ビット	名称	機能	ロード/ ストア
5	SORST ^{注1}	シリアル出力リセット許可ビット シリアル出力回路をリセットする。SSER=0, SSEF=1 となる。シリアル出力リセット後、自動的にゼロクリアされる。 0:(デフォルト) 1:SO リセット要求	L/S
4	SIRST ^{注1}	シリアル入力リセット許可ビット シリアル入力回路をリセットする。SLER=0, SLEF=0 となる。シリアル入力リセット後、自動的にゼロクリアされる。 0:(デフォルト) 1:SI リセット要求	L/S
3	SSER ^{注2}	SDT ストア・エラー・フラグ ・0:エラーなし(デフォルト) ・1:エラー SSEF が 0 のときに μ PD77210 ファミリが SDT に書き込んだ場合に 1 になる。 一度セットされると、 μ PD77210 ファミリが 0 を書き込むまで変化しない。	L/S
2	SLER ^{注2}	SDT ロード・エラー・フラグ ・0:エラーなし(デフォルト) ・1:エラー SLEF が 0 のときに μ PD77210 ファミリが SDT を読み出した場合に 1 になる。 一度セットされると、 μ PD77210 ファミリが 0 を書き込むまで変化しない。	L/S
1	SSEF	SDT ストア許可フラグ ・SDT の値をシリアル出力用シフト・レジスタに転送した場合に 1 になる。 ・ μ PD77210 ファミリが SDT にデータをストアすると 0 になる。	L
0	SLEF ^{注3}	SDT ロード許可フラグ ・シリアル入力用シフト・レジスタの値を SDT に転送した場合に 1 になる。 ・ μ PD77210 ファミリが SDT からデータをロードすると 0 になる。	L

注 1. シリアル・リセット・ビットを 1 にしたあと 3 サイクルの間は SIO へのアクセスを実行できません。アクセスした場合、結果は正常に反映されません。

SST のほかのビットへの設定と同時にシリアル・リセット・ビットを 1 にすることを禁止します。

- ★ 2. ストア・エラー・フラグ/ロード・エラー・フラグが 1 のとき、SDT へのストア/ロードを実行しても、シリアル割り込みは発生しません。シリアル・ステータス・レジスタのストア・エラー・フラグ/ロード・エラー・フラグを直接クリア(0)にしてから使用してください(シリアル・ステータス・レジスタのリセットによるクリアは使用しないでください)。
- ★ 3. シリアル・インタフェースのシリアル・データ・レジスタ・ロード許可フラグが許可(0)のとき、FINT 命令を実行すると、それ以降のロード許可の割り込みを受け付けなくなります。

表 5-12 SICM ビットと SIEF ビットの組み合わせ

SICM	SIEF	機 能
0	0	状態遷移モード。シリアル入力を行わないときもこのモードになる。
1	0	シリアル連続入力モード。
0	1	シリアル単発入力モード。1 にセットされた SIEF は次の命令サイクルで自動的に 0 にリセットされる。
1	1	設定禁止

5.5.3 スタンダード・シリアル・インタフェースのタイミング

(1) シリアル出力のタイミング

一般的なシリアル出力は、次のような手順で動作します。過程の から までで、SDT 書き込みウエイトがかからない場合を図 5-11 に、連続データ出力、単一データ出力の場合に分けて示します。

アプリケーション・プログラムが SDT (シリアル・データ・レジスタ) に対するストア命令を実行する。

により、SST (シリアル・ステータス・レジスタ) の SSEF (SDT ストア許可フラグ) が 0 になり、SDT に対してそれ以上の書き込みをしないようアプリケーションに知らせる。このとき SSWE (SDT ストア・ウエイト許可ビット) がセットされていれば、同時に SDT ストア・ウエイト機能を有効化し、SDT に対する書き込みを自動的にブロックする。

SOS (シリアル出力シフト・レジスタ) が空の場合、SDT に設定されたデータは 3 シリアル・クロック後 SOS に転送され、TSORQ (シリアル出力リクエスト端子) がアクティブ (ハイ・レベル) になり、シリアル出力要求の発生を外部に知らせる。

外部から TSOEN/ASOEN (シリアル出力イネーブル端子) をアクティブ (ハイ・レベル) にすると (a)、直後の TSCK/ASCK (シリアル・クロック端子) の立ち下がりエッジでサンプルされ^{注1} (b)、TSCK/ASCK の続く立ち上がりエッジで TSORQ がロウ・レベルになり (c)、同じ立ち上がりエッジから TSO/ASO (シリアル・データ出力端子) にデータ出力が開始される (d)。

SDT が空になったあと、SSEF は 1 にセットされて、次のデータ書き込みが可能であることをアプリケーションに知らせ (a)、SSWE = 1 のもとで有効化されていた SDT ストア・ウエイト機能は解除される。またこのとき TSO/ASO による割り込み要求を発生する (b) が割り込みの要因別許可フラグや EI の状態により、有効な割り込みとして処理されたり、または記憶されることになる (4.4.4 割り込み参照)。

最後のビット・データ出力を完了したところでまだ次のデータが供給されていないならば、続く TSCK/ASCK の立ち上がりエッジで、TSO/ASO はハイ・インピーダンスとなる^{注2}。

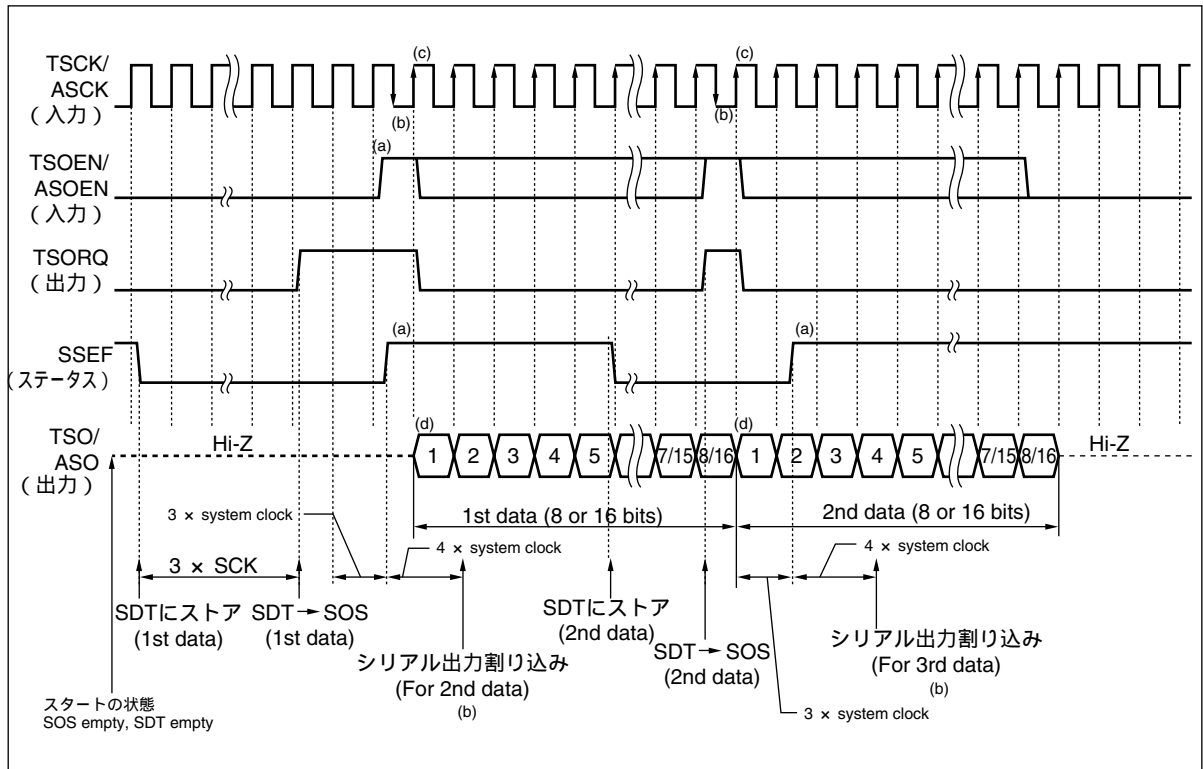
注 1. TSOEN/ASOEN がアクティブになるまえに、少なくとも 3 回の TSCK/ASCK 立ち上がりエッジが経過していなければなりません。データの入出力時のみにクロックをバースト的に供給するシステム構成ではこのことに十分注意してください。

2. 厳密には、最終ビット出力までにすでに次のデータが供給されており、かつ最終ビット出力サイクル中、TSCK/ASCK の立ち下がりまでに TSOEN/ASOEN がアクティブとなり、有効にサンプルされた場合、TSO/ASO はハイ・インピーダンスとならずに連続して次のデータを出力します (図 5-11 シリアル・インタフェース出力タイミング参照)。

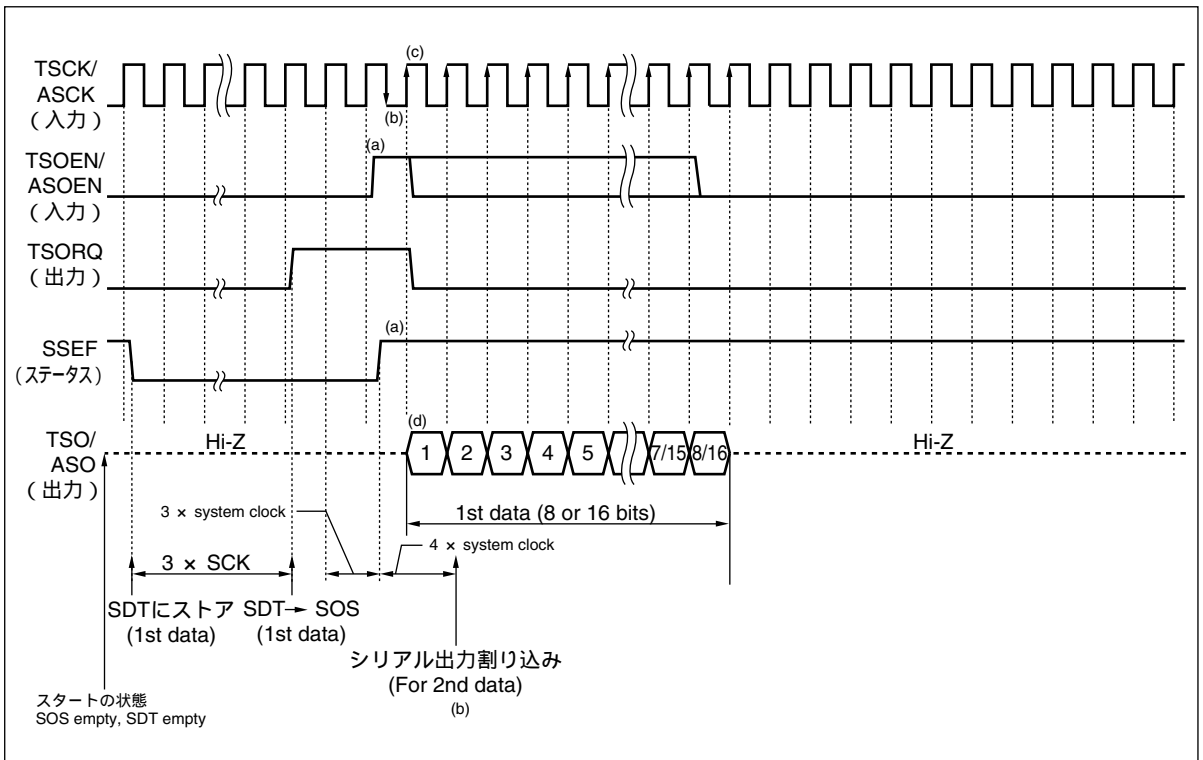
したがって、最終ビットを出力したあと、少なくとも 1 回の TSCK/ASCK 立ち上がりエッジを供給しなければなりません。

図 5-11 シリアル・インタフェース出力タイミング

(a) 連続データ



(b) 単一データ



(2) シリアル入力のタイミング

一般的なシリアル入力は、次のような手順で動作します。過程の から までで、SDT 読み出しウエイトがかからない場合を図 5-12に、連続データ出力、単一データ出力の場合に分けて示します。

TSIAK (シリアル入力許可端子) アクティブ (ハイ・レベル) のもとで、外部から TSIEN/ASIEN (シリアル入力イネーブル端子) をアクティブ (ハイ・レベル) にすることで、シリアル・データ入力手順を開始する。

による TSIEN/ASIEN の変化は、直後の TSCK/ASCK の立ち下がりエッジでサンプリングされ^{※1} (a)、TSCK/ASCK の続く立ち上がりエッジで TSIAK がロウ・レベルになり (b)、同じ TSCK/ASCK サイクルの立ち下がりエッジから TSI/ASI (シリアル・データ入力端子) に与えられているデータの取り込みを開始する (c)。データは、TSI/ASI から SIS (シリアル入力シフト・レジスタ) へ 1 ビットずつ TSCK/ASCK の立ち下がりエッジに同期して取り込まれる。

指定されたビット数の最後のビットを取り込む TSCK/ASCK サイクルで、取り込み直前の立ち上がりエッジに同期して TSIAK がアクティブになり (a)、次のデータ入力が可能であることを外部に知らせる。

ビットを取り込むと^{※2} (b)、SLEF (SDT ロード許可フラグ) = 0 の場合、取り込まれた最後のデータは SIS からただちに SDT^{※3} に転送される。

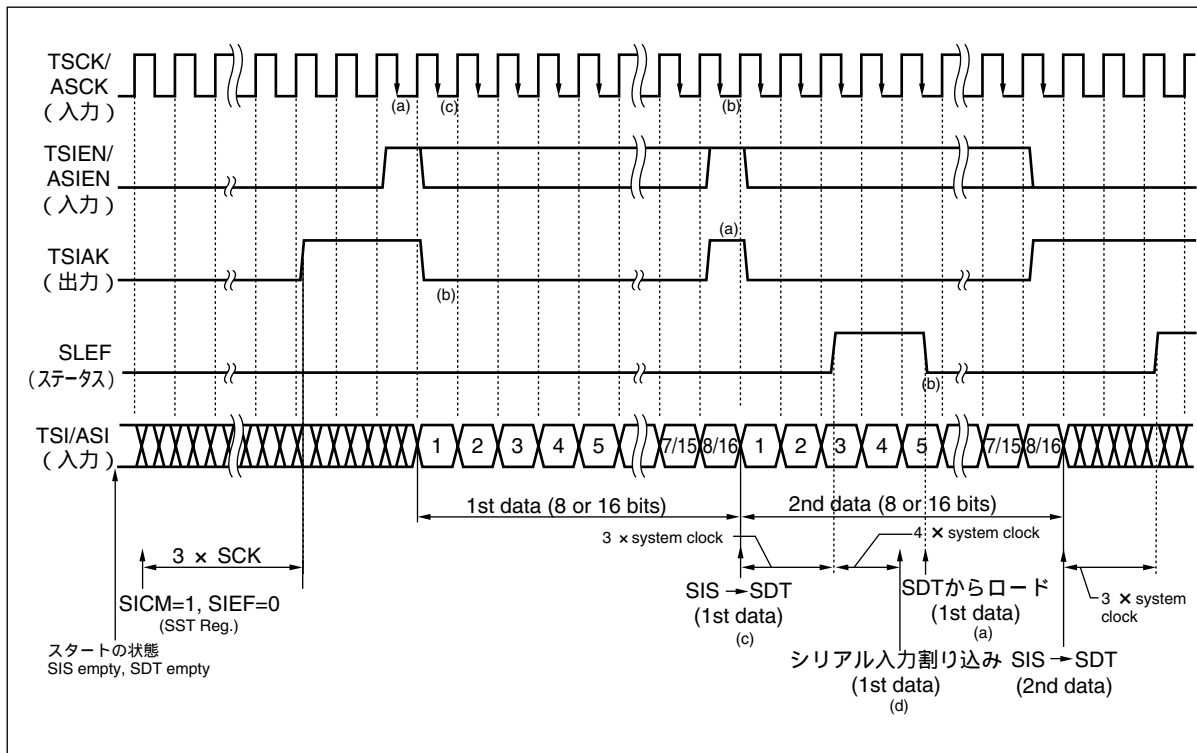
その後、SLEF = 1 になってアプリケーション・プログラムにシリアル入力データ・ワードの成立を知らせる (c)。このとき SLWE (SDT : ロード・ウエイト許可ビット) = 1 でデータ・ウエイト状態にあった場合、ウエイトは解除される。またこのとき、TSI/ASI による割り込み要求を発生するが (d)、割り込みの要因許可フラグや EI の状態により、有効な割り込みとして処理されたり、または記憶されることになる (4.4.4 割り込み参照)。

アプリケーション・プログラムが SDT からのロード命令を実行すると (a)、SLEF は 0 になり入力データが空であることを示す (b)。また、SLWE = 1 であれば、同時に SDT ロード・ウエイト機能を有効化し、SDT に対するそれ以上の読み出しを自動的にブロックする。

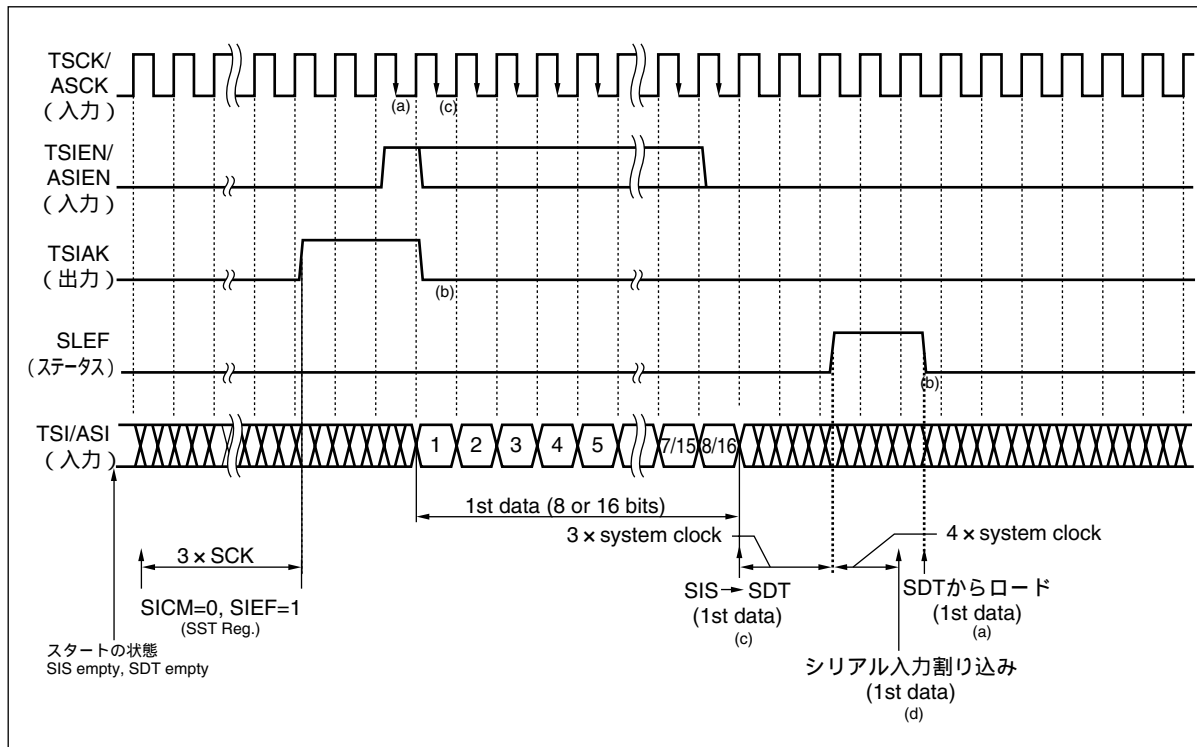
- 注 1. TSIEN/ASIEN がアクティブになる前に、少なくとも 3 回の TSCK/ASCK 立ち上がりエッジが経過していなければなりません。シリアル入出力部のハードウェアは、TSCK/ASCK をタイミング・クロックとしてパイプライン状に動作します。データの入出力時のみにクロックをバースト的に供給するシステム構成では、このことに十分注意してください。
- 連続してクロックを供給するようなシステム構成の場合は、リセット後の最初のデータについてだけ注意してください。
- また、このような理由から、最終ビットが取り込まれたあと、少なくとも 2 回の TSCK/ASCK 立ち上がりエッジを供給しなければなりません。
2. 最終ビット入力サイクル中、TSCK/ASCK の立ち下がりまでに TSIEN/ASIEN がアクティブとなって有効にサンプリングされた場合、連続する次の TSCK/ASCK サイクルから次のデータを取り込みます (図 5-12 シリアル・インタフェース入力タイミング参照)。
3. SDT は、シリアル入力用とシリアル出力用とで別々のレジスタです。

図 5-12 シリアル・インタフェース入力タイミング

(a) SICM = 1, SIEF = 0 : 連続モード



(b) SICM = 1, SIEF = 0 : 単一モード



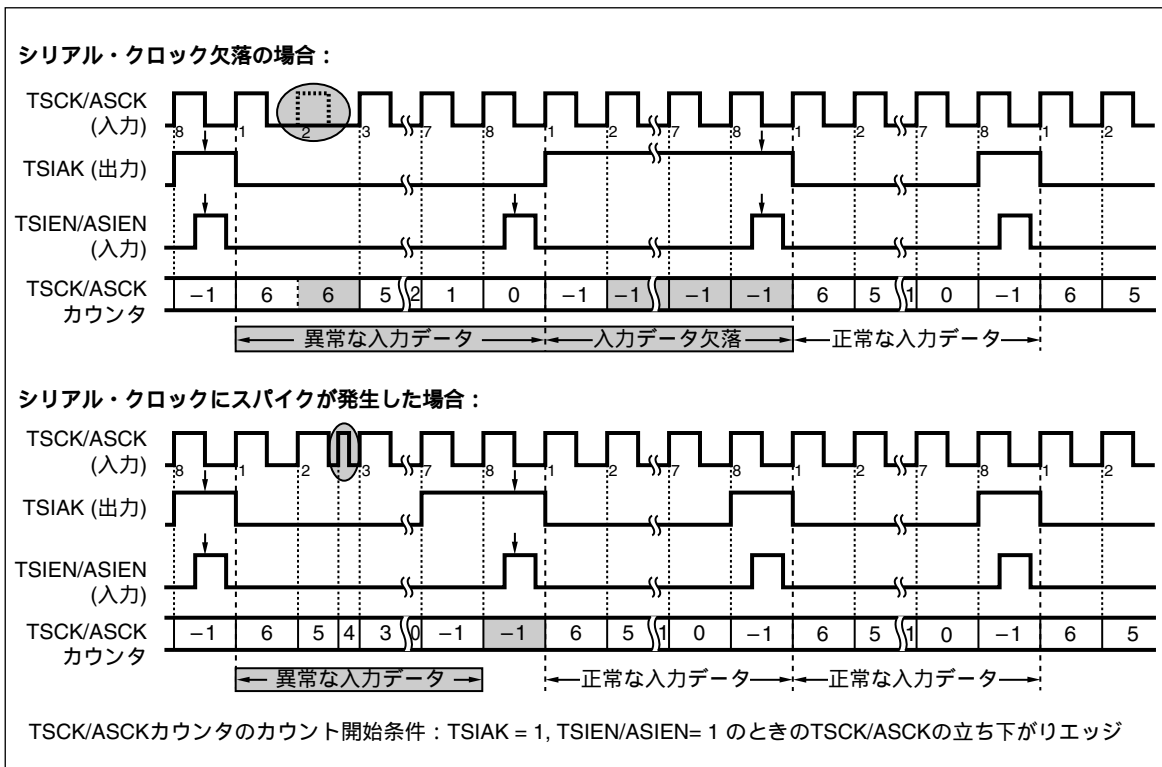
(3) 規格外のシリアル・クロックによる入出力タイミング

図 5-13に規格外のシリアル・クロックによって発生したシリアル・クロック・カウンタの様子を示します。

TSIEN/ASIEN, TSOEN/ASOEN をアクティブにしたままでもデータの入出力は可能ですが、規格外のシリアル・クロックを入力したためにビットにずれが生じると、入出力タイミングを修正できなくなります。

図 5-13のように、TSIEN/ASIEN, TSOEN/ASOEN をインアクティブにすることで、このようなビットのずれを修正することができます。ここでは、TSCK/ASCK をカウントすることによって、TSIEN/ASIEN を入力している例を示しましたが、TSLAK の状態をみて TSIEN/ASIEN を入力する方が確実です。

図 5-13 シリアル・インタフェース：シリアル・クロック・カウンタの動作



(4) ハンドシェーク

μ PD77210 ファミリのアプリケーション・プログラムとシリアル・インタフェースとのハンドシェークは、次の3つの形式があります。

- ポーリング
- ウェイト
- 割り込み

(a) ポーリング

SST(シリアル・ステータス・レジスタ)のSSEF(SDTストア許可フラグ),およびSLEF(SDTロード許可フラグ)を,常時監視/評価することでハンドシェークの同期をとります。次に,ポーリングによるシリアル出力の例を示します。

```

/* SST1, SO1 は予約語ではないので, 明示的に定義する */
#define SST1      0x3801
#define SO1      0x3800

/* 内部割り込み SO1, SI1 を禁止する */
R0L = SR          ;
R0 = R0 | 0x0030  ;
SR = R0L         ;

R0L = 0x0         ; シリアル・ステータスを次のように設定する
*SST1 : X = R0L   ; ・MSB ファースト出力
                  ; ・MSB ファースト入力
                  ; ・16 ビット・ワード出力
                  ; ・16 ビット・ワード入力
                  ; ・SDT 書き込みウエイト機能を使用しない
                  ; ・SDT 読み出しウエイト機能を使用しない
                  ; ・シリアル入力を行わない
                  ; ・シリアル入出力のエラー・フラグをクリア

POLL : R0L = *SST1 : X      ; SSEF を判定し, ストア・イネーブルになるまでループして待ち合わせ
      ;                    ; ます。
      R0 = R0 & 0x2        ;
      if ( R0 == 0 ) jmp POLL ;

*SO1 : X = R1H            ; ストア・イネーブルになったので R1H のデータを出力します。

```

(b) ウェイト

次のような場合に SDT に対してロード/ストアを実行すると、命令実行にウェイトが発生します。

- ・書き込みウェイト機能許可 (SSWE = 1) 時、SSEF = 0 (SDT (out) に有効なデータが存在) にもかかわらず、SDT (out) に対してシリアル出力データのストアが行われた場合。
- ・読み出しウェイト機能許可 (SLWE = 1) 時、SLEF = 0 (SDT (in) に有効なデータが存在しない) にもかかわらず SDT (in) からシリアル入力データのロードが行われた場合。

この形式は、ハンドシェイク手順がハードウェアで自動的に実行されますから、アプリケーション・プログラムで特別なハンドシェイク手順記述を要しないところにメリットがあります。次に、ウェイトによる SDT 書き込み例を示します。

```

/* SST1, SO1 は予約語ではないので、明示的に定義する */
#define SST1      0x3801
#define SO1       0x3800

/* 内部割り込み SO1, SI1 を禁止する */
R0L = SR          ;
R0 = R0 | 0x0030  ;
SR = R0L         ;

R0L = 0x800       ;シリアル・ステータスを次のように設定する
*SST1 : X = R0L   ;・MSB ファースト出力
                ;・MSB ファースト入力
                ;・16 ビット・ワード出力
                ;・16 ビット・ワード入力
                ;・SDT 書き込みウェイト機能を使用する
                ;・SDT 読み出しウェイト機能を使用しない
                ;・シリアル入力を行わない
                ;・シリアル入出力のエラー・フラグをクリア

*SO1 : X = R1H    ; SSEF = 1 になり次第, R1H のデータを出力する

```

注意 アプリケーション・プログラムから SDT へ書き込んだ場合、SDT を SOS に転送しないかぎり (つまり SOS の前データの全ビットを外部へシフト・アウトしないかぎり) ウェイトが解除されることはありません。したがって SDT に対して μ PD77210 ファミリー内部からの書き込みと、外部からの読み出しが 1 対 1 に対応しない場合は、ハングアップする可能性があります。また、ウェイト中は割り込みが遅延されます (4.4.4 割り込み参照)。

(c) 割り込み

SDT (out) にデータを書き込むことができる場合、および、SDT (in) からデータを読み出すことができる場合に割り込みがかかることでハンドシェイクを確立します。したがって、ほかの処理を実行中でも、それらの処理とは無関係（非同期）にシリアル入出力を実行することができます。次に、割り込みを使用したシリアル入出力の例を示します。

```

/*シリアルI/Oを定義する                                     */
#define SST1      *0x3801:X
#define SI1       *0x3800:X
#define SO1       *0x3800:X

/* 割り込みベクタ・テーブルのエントリ                       */
SegSI1           IMSEG AT 0x220    ; SIO#1 入力割り込みルーチン

                R0H = SI1          ; SDT (in) から読み出し
                R0 = R0H*R1H        ;
                *DP0 + + = R0H      ; バッファへ待避
                RETI                ; 割り込みから復帰

SegSO1           IMSEG AT 0x224    ; SIO#1 出力割り込みルーチン

                R0H = *DP4 + +      ; バッファから読み出し
                SO1 = R0H           ; SDT (out) へ書き込み
                RETI                ; 割り込みから復帰
                NOP                 ;

/* シリアル入出力を初期化するために割り込みを禁止         */
                R1L = EIR ; すべての割り込みを禁止
                R1 = R1 | 0x8000    ; EI = 1
                EIR=R1L            ;
                NOP                 ; EI = 1 が有効になるまで2 ウェイト・サイクル必要
                NOP                 ;

                R0L = SR            ; SI1 および SO1 割り込み許可
                R0 = R0 & 0xFFCF    ;
                SR = R0L            ;

                R1 = R1 & 0x7FFF    ; すべての割り込みを許可
                EIR = R1L          ;
                FINT                ; 前の割り込みを廃棄

                ; SST1 を初期化
                R0L = 0x0200        ; input/output : MSB-first , 16-bit
                ; ロード/ストア・ウェイト機能なし
                SST1 = R0L          ; シリアル入力連続モード
                SO1 = R0L           ; ダミー・ストア (注意参照)

```

注意 シリアル出力の割り込みは、SDT レジスタからシリアル出力シフト・レジスタに転送されてから割り込みが発生するため、次の点に注意してください。

- (1) 最初にデータを転送するときは、割り込みのモードによっては、ダミーのデータを転送して強制的に割り込みを発生させるか、最初のデータの転送は割り込みを使用しないで転送します。
- (2) バーストでデータを転送するときに、バースト・データの最終ワードに発生する割り込みタイミングでは、次のバースト転送の最初のデータは生成されていない可能性が高いため、最終ワードを SDT に転送する直前の命令で割り込みを禁止し、次のバースト・データの生成が終わった時点で(1)と同様の命令を実行し、次のバースト・データを転送してください。

```

例)                ;/* 最終ワードの SDT 書き込み時 */
R0L = SR           ;/* (割り込み処理中 DI 状態) */
R0 = R0 | 0x0020   ;/* SO1 割り込み禁止 */
SR = R0L           ;
*SO1 : X = R0H     ;

```

5.5.4 起動時の TSO に関する注意事項

μ PD77210 ファミリを起動後、TSO 動作開始時に、TSO 端子から想定されない不定データが出力されます。詳しい説明は、5.4.4 起動時の ASIO に関する注意事項を参照してください。

★ 5.5.5 シリアル・ステータス・レジスタのリセット機能について

シリアル入出力に関するフラグの初期化、内部カウンタの初期化を行ないませんが、入出力シフト・レジスタの初期化は行ないません。

マスタ・モードにおける SO 割り込みは内部カウンタを基準にしており、場合によってはリセットによる同期ずれが発生するおそれがあります。

シリアル・ステータス・レジスタのリセットは、使用しないでください。

リセット解除後、SIO の初期化シーケンスの先頭でシリアル・ステータス・レジスタのリセットをすることは問題ありません。

5.6 ホスト・インタフェース (HIO)

ホスト・インタフェース (HIO) は外部のホスト CPU とのデータ転送を行うためのインタフェース回路です。主な特徴を次に示します。

• バス幅

バイト (8 ビット) モード: 外部バスを 8 ビットとし, ホスト・インタフェース用レジスタ (HDT) の上位 8 ビットまたは下位 8 ビットを外部からアドレス選択してアクセス

ワード (16 ビット) モード: 外部バスを 16 ビットとし, ホスト・インタフェース用レジスタ (HDT) にホスト CPU から直接 16 ビット・アクセス

• 内部ハンドシェーク

ポーリング, ウェイト, および割り込みによるハンドシェークが可能

• 外部ハンドシェーク

専用ステータス信号 ($\overline{\text{HRE}}$, $\overline{\text{HWE}}$) によるハンドシェークが可能

ホスト・インタフェースのブロック図を図 5-14 に, レジスタ群を表 5-13 に示します。

図 5-14 ホスト・インタフェースのブロック図

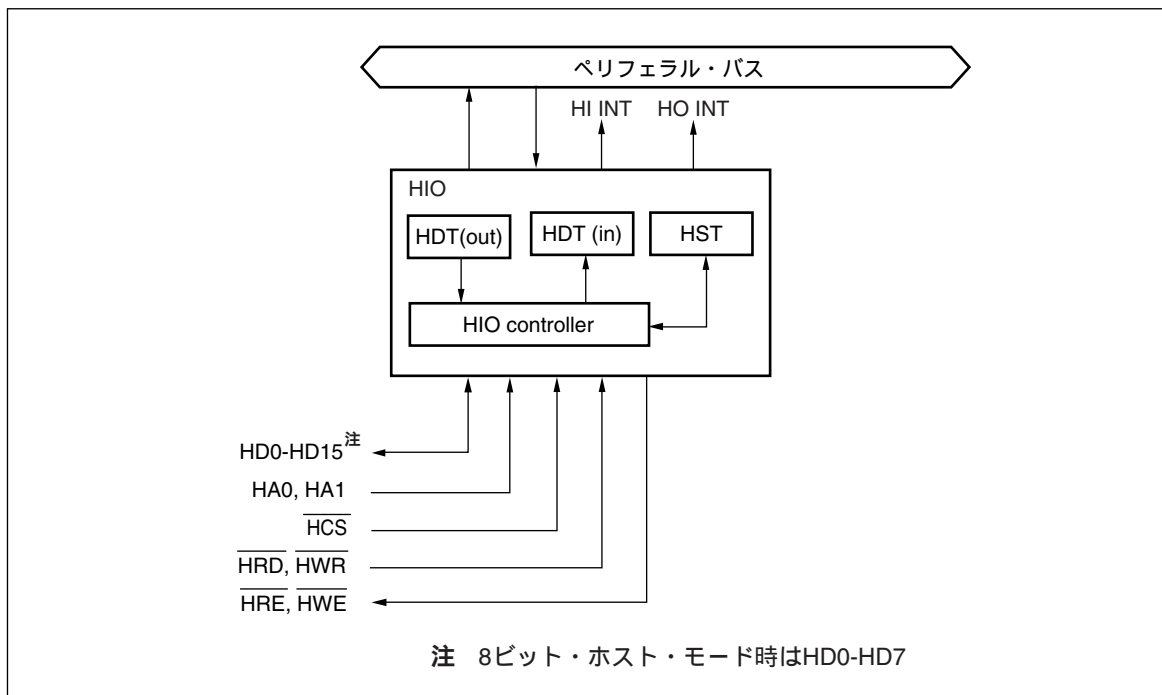


表 5-13 ホスト・インタフェースのレジスタ

X/Y メモリ・アドレス	レジスタ名	機能	ロード/ストア
0x3820	HDT	ホスト・インタフェース・データ・レジスタ	L/S
0x3821	HST	ホスト・インタフェース・ステータス・レジスタ	L/S

5.6.1 ホスト・インタフェースの端子

(1) $\overline{\text{HCS}}$ (ホスト・チップ・セレクト - 入出力)

ホスト・インタフェース・セレクト信号の入力端子です。

ホストがホスト・インタフェースのレジスタをアクセスする間、アクティブ(ロウ・レベル)にします。

(2) HA0, HA1 (ホスト・アドレス - 入力)

ホスト・インタフェースのアドレス入力端子です。

アクセスするホスト・インタフェースのレジスタを指定します。

ホストがホスト・インタフェースのレジスタをアクセスしている間は変化させないでください。

(3) $\overline{\text{HRD}}$ (ホスト・リード・ストローク - 入力)

ホスト・インタフェースのリード・ストローク信号の入力端子です。

ホストがホスト・インタフェースのレジスタを読み出す場合に、アクティブ(ロウ・レベル)にします。

$\overline{\text{HWR}}$ 端子と同時にアクティブにすることを禁止します。

(4) $\overline{\text{HWR}}$ (ホスト・ライト・ストローク - 入力)

ホスト・インタフェースのライト・ストローク信号の入力端子です。

ホストがホスト・インタフェースのレジスタに書き込む場合に、アクティブ(ロウ・レベル)にします。

$\overline{\text{HRD}}$ 端子と同時にアクティブにすることを禁止します。

(5) HD0-HD15 (ホスト・データ - 入出力)

ホスト・インタフェースのデータ入出力端子です。8ビット・バス・モードのときはHD0-HD8を入出力端子として使用します。16ビット・バス・モードのときのHD8-HD15は汎用入出力ポート端子と兼用の端子を使うため、兼用となっている端子は汎用入出力端子としては利用できません。

ホストがホスト・インタフェースのレジスタをアクセスするとき、データの入出力を行います。

$\overline{\text{HCS}}$ がインアクティブ(ハイ・レベル)の場合、ハイ・インピーダンスとなります。

(6) $\overline{\text{HRE}}$ (ホスト・リード・イネーブル - 出力)

HDTの読み出し許可を示す信号を出力する端子です。

HDTを読み出し可能な場合にアクティブ(ロウ・レベル)になり、HDTの読み出し(8ビット・モード時は上位バイトの読み出し)のとき $\overline{\text{HRD}}$ 端子の立ち下がりに同期してインアクティブ(ハイ・レベル)になります。

ハードウェア・リセットによりインアクティブになります。

★ μ PD77210ファミリがHDTにデータをストアしたあと、マスク実行(HST HREM = 1)→マスク解除(HST HREM = 0)とすることによって、 $\overline{\text{HRE}}$ 端子がアクティブ(ロウ・レベル)になります。

μ PD77210ファミリがHDTにデータをストアしていないとき、マスク実行(HST HREM = 1)→マスク解除(HST HREM = 0)としても、 $\overline{\text{HRE}}$ 端子はインアクティブ(ハイ・レベル)のままです。また、HST HREMはマスク解除(HST HREM = 0)→マスク実行(HST HREM = 1)とすることによって、 $\overline{\text{HRE}}$ 端子の状態にかかわらず、 $\overline{\text{HRE}}$ 端子はインアクティブ(ハイ・レベル)となります。

(7) $\overline{\text{HWE}}$ (ホスト・ライト・イネーブル - 出力)

HDT への書き込み許可を示す信号を出力する端子です。

HDT に書き込み可能な場合にアクティブ (ロウ・レベル) になり, HDT への書き込み (8 ビット・モード時は上位バイトへの書き込み) のとき $\overline{\text{HWR}}$ 端子の立ち下がりに同期してインアクティブ (ハイ・レベル) になります。

ハードウェア・リセットによりインアクティブになります。

★ ホスト CPU が HDT にデータをライトしたあと, マスク実行 (HST HWEM = 1) → マスク解除 (HST HWEM = 0) とすることによって, $\overline{\text{HWR}}$ 端子がアクティブ (ロウ・レベル) になります。

ホスト CPU が HDT にデータをライトしていないとき, マスク実行 (HST HWEM = 1) → マスク解除 (HST HWEM = 0) としても, $\overline{\text{HWR}}$ 端子はインアクティブ (ハイ・レベル) のままです。また, HST HREM はマスク解除 (HST HWEM = 0) → マスク実行 (HST HWEM = 1) とすることによって, $\overline{\text{HWR}}$ 端子の状態にかかわらず, $\overline{\text{HWR}}$ 端子はインアクティブ (ハイ・レベル) となります。

5.6.2 ホスト・インタフェースのレジスタ

(1) HDT (ホスト・データ・レジスタ : Host Data Transfer Register)

ホスト・インタフェースからホスト・データを入出力するための 16 ビット・レジスタです。出力用と入力用のレジスタを別々に持っています。

HDT (out) は, 出力するデータを設定する 16 ビットのレジスタです。HDT に対するストア命令を実行すると HDT (out) にペリフェラル・バスからデータを入力します。

SDT (in) は, 入力されたデータを読み出す 16 ビットのレジスタです。HDT に対してロード命令を実行すると, ペリフェラル・バスにデータを出力します。

(2) HST (ホスト・ステータス・レジスタ : Host Status Register)

ホスト・インタフェースのモード設定と, ステータスを表す 16 ビットのレジスタです。8 ビット / 16 ビット・バス・モードの切り替え, μ PD77210 ファミリの DSP コア・カーネル部分とのインタフェースの指定や書き込み, 読み出しエラーの表示を行います。

HST のビット構成については表 5-14 に示します。リセット時の値は, 0x301 です。

表 5-14 HST のビット構成 (1/2)

ビット	名称	機能	ロード/ ストア (DSP から)	リード/ ライト (ホストから)
15-12	予約	<ul style="list-style-type: none"> ・0 以外の書き込みは行わないこと。 ・読み出し時は不定。 	-	-
11	HBM	ホスト・インタフェース・バス幅設定ビット <ul style="list-style-type: none"> ・0 : 8 ビット・モード (デフォルト) ・1 : 16 ビット・モード 	L/S	R
10	HAWE	HDT アクセス・ウエイト許可ビット <ul style="list-style-type: none"> ・0 : ウエイトを使用しない (デフォルト) ・1 : ウエイトを使用する。 HREF=1 のときに HDT (out) にデータをストアしようとした場合に、 μ PD77210 ファミリにウエイトを挿入する。 HWEF=1 のときに HDT (in) からデータをロードしようとした場合に、 μ PD77210 ファミリにウエイトを挿入する。	L/S	R
9	HREM	HRE マスク・ビット 0 : マスクしない 1 : マスクする (デフォルト)	L/S	R
8	HWEM	HWE マスク・ビット 0 : マスクしない 1 : マスクする (デフォルト)	L/S	R
7	UF1	ユーザ・フラグ	L/S	R
6	UF0			
5	HRER	ホスト・リード・エラー・フラグ <ul style="list-style-type: none"> ・0 : エラーなし (デフォルト) ・1 : エラー HREF が 0 のときにホスト CPU が HDT を読み出した場合に 1 になる。 一度セットされると、 μ PD77210 ファミリが 0 を書き込むまで変化しない。	L/S	R
4	HWER	ホスト・ライト・エラー・フラグ <ul style="list-style-type: none"> ・0 : エラーなし (デフォルト) ・1 : エラー HWEF が 0 のときにホスト CPU が HDT に書き込んだ場合に 1 になる。 一度セットされると、 μ PD77210 ファミリが 0 を書き込むまで変化しない。	L/S	R
3	HSER	HDT ストア・エラー・フラグ <ul style="list-style-type: none"> ・0 : エラーなし (デフォルト) ・1 : エラー HREF が 1 のときに μ PD77210 ファミリが HDT に書き込んだ場合に 1 になる。 一度セットされると、 μ PD77210 ファミリが 0 を書き込むまで変化しない。	L/S	R

表 5-14 HST のビット構成 (2/2)

ビット	名称	機能	ロード/ ストア (DSP から)	リード/ ライト (ホストから)
2	HLEP	HDT ロード・エラー・フラグ ・0: エラーなし (デフォルト) ・1: エラー HWEF が 1 のときに μ PD77210 ファミリが HDT を読み出した場合に 1 になる。 一度セットされると、 μ PD77210 ファミリが 0 を書き込むまで変化しない。	L/S	R
1	HREF	ホスト・リード・イネーブル・フラグ ・0: リード禁止 ・1: リード許可 μ PD77210 ファミリが HDT に書き込んだ場合に 1 になる。ホスト CPU が HDT を読み出した場合に 0 になる。	L	R
0	HWEF	ホスト・ライト・イネーブル・フラグ ・0: ライト禁止 ・1: ライト許可 μ PD77210 ファミリが HDT から読み出した場合に 1 になる。ホスト CPU が HDT に書き込んだ場合に 0 になる。	L	R

5.6.3 ホスト CPU から見たホスト・インタフェースのレジスタ

ホスト CPU は μ PD77210 ファミリの HA0, HA1 端子を用いてアクセス対象となるレジスタを指定します。

表 5-15 に外部からアクセスする場合のホスト・インタフェース用レジスタを示します。

表 5-15 ホスト・インタフェース用レジスタの選択

$\overline{\text{HCS}}$	$\overline{\text{HRD}}$	$\overline{\text{HWR}}$	HA1	HA0	転送対象のレジスタ	8 ビット・モード時の 転送対象バイト	16 ビット・モード時の 転送対象バイト
0	0	0	x	x	Reserved(設定禁止)	-	-
0	0	1	0	0	HDT (out)	下位 8 ビット	16 ビット
0	0	1	0	1		上位 8 ビット	
0	0	1	1	0	HST	下位 8 ビット	16 ビット
0	0	1	1	1		上位 8 ビット	
0	1	0	0	0	HDT (in)	下位 8 ビット	16 ビット
0	1	0	0	1		上位 8 ビット	
0	1	0	1	x	Reserved(設定禁止)	-	-
0	1	1	x	x	対象なし	-	-
1	x	x	x	x			

5.6.4 ホスト・インタフェースのタイミング

(1) ホスト・リードの動作 (μ PD77210 ファミリ ホスト)

μ PD77210 ファミリからホストへデータを転送する動作手順は次のようになります (次の過程を 16 ビット・データ転送を例にとり, HDT 書き込みウエイトを使用しない例を図 5-15 に示します)。

μ PD77210 ファミリのアプリケーション・プログラムが, HDT (ホスト・データ・レジスタ) へデータを書き込む (a), (b)。

により, HST (ホスト・インタフェース・ステータス・レジスタ) の HREF (ホスト・リード・イネーブル・フラグ) が 1 になる (a)。

また, HST の HREM (HRE マスク・ビット) = 0 であれば, $\overline{\text{HRE}}$ 端子アクティブ (ロウ・レベル) になり, ハードウェア信号として外部に知らせる (b)。

ホストは次のいずれかの方法で, HDT にデータが提示されていることを認識できる。

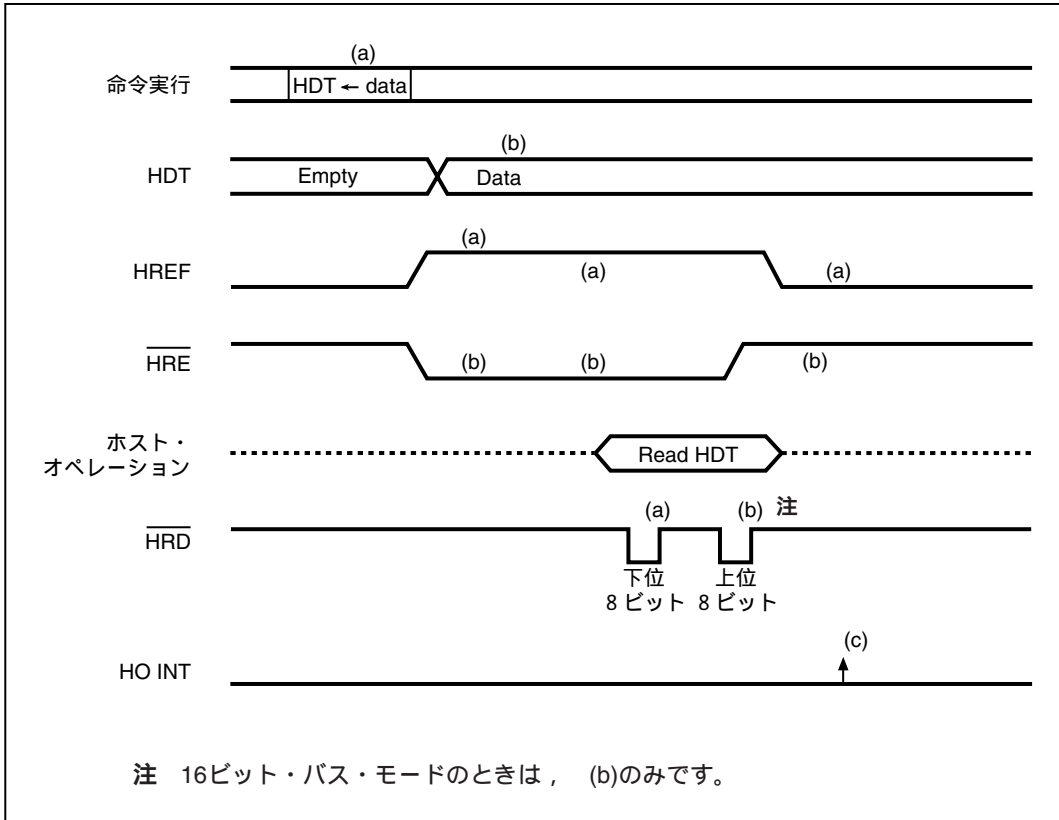
(1) HST を読み出し, ソフトウェア的に HREF = 1 を検出する (a)。

(2) $\overline{\text{HRE}}$ 端子のロウ・レベルを検知する (b)。

ホストは HDT をリードする。このとき, 16 ビット転送であれば最初に下位 8 ビット (a), 次に上位 8 ビット (b) の順番でリードしなければならない。また, 8 ビット転送の場合は常に上位 8 ビットをリードする (HREF, および $\overline{\text{HRE}}$ の論理を参照)。

により HST の HREF は 0 になり (a), $\overline{\text{HRE}}$ 端子はインアクティブ (ハイ・レベル) となる (b)。このとき HO による割り込み要求を発生するが (c), 割り込みの要因別許可フラグや EI の状態により有効な割り込みとして処理されたり, または記憶されることになる (4.4.4 割り込みを参照)。

図 5-15 ホスト・リード手順 (μ PD77210 ファミリ ホスト): HDT 書き込みウエイトなし



(2) ホスト・ライトの動作 (μ PD77210 ファミリ ホスト)

ホストから μ PD77210 ファミリーにデータを転送する動作手順は次のようになります (次の過程を 16 ビット・データ転送を例にとり, HDT 読み出しウエイトを使用しない例を図 5-16に示します)。

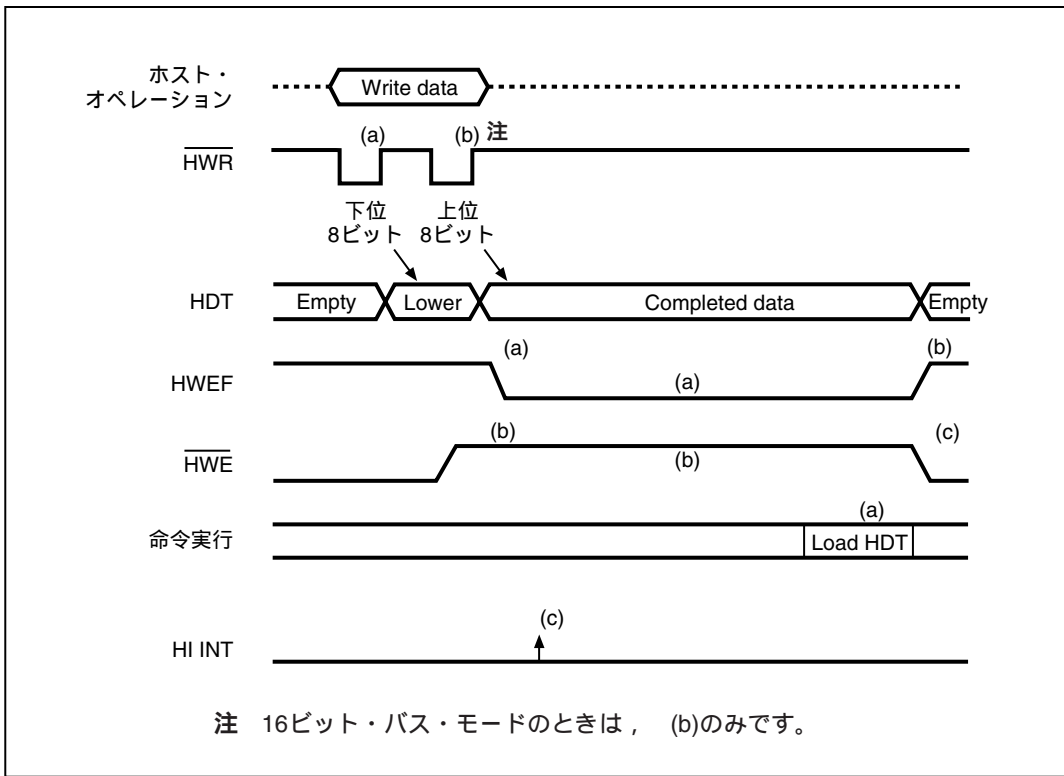
ホストは μ PD77210 ファミリーの HDT にデータを書き込む。このとき, 16 ビット転送であれば下位 8 ビット (a), 上位 8 ビット (b) の順序で, また 8 ビット転送であれば常に上位 8 ビットにデータを書き込む (HWEF, および $\overline{\text{HWE}}$ の論理を参照)。

により HST の HWEF = 0 となって μ PD77210 ファミリーのアプリケーション・プログラムに対し, HDT にデータが提示されたことを知らせる (a)。同時に $\overline{\text{HWE}}$ 端子はインアクティブ (ハイ・レベル) になり, HDT がビジーであることを外部に知らせる (b)。

また, HI を要因とする割り込み要求を発生するが (c), 割り込みの要因別許可フラグや EI の状態により, 有効な割り込みとして処理されたり, または記憶される (4.4.4 割り込み参照)。 μ PD77210 ファミリーのアプリケーション・プログラムは, 次のいずれかの方法で, HDT にホストからのデータが準備されたことを認識できる。(1) HST の HWEF = 0 となったことを検出する (a)。(2) HI による割り込みを待つ (b)。

によりアプリケーション・プログラムは HDT から読み出す (a)。これにより HWEF = 1 になる (b)。同時に $\overline{\text{HWE}}$ 端子はアクティブ (ロウ・レベル) になり (c), 外部回路は HDT に対して書き込み可能であることを認識する。

図 5-16 ホスト・ライト手順 (μ PD77210 ファミリ ホスト): HDT 読み出しウエイトなし



(3) ハンドシェーク

μ PD77210 ファミリとホストとのあいだでインタフェースする場合のハンドシェークには、次の3つの形式があります。

ポーリング

ウェイト

割り込み

(a) ポーリング

HST (ホスト・インタフェース・ステータス・レジスタ) の HREF (ホスト・リード・イネーブル・フラグ), および HWEF (ホスト・ライト・イネーブル・フラグ) を、常に監視, 評価することでハンドシェークの同期をとります。次に、ポーリングによるホスト・リード (μ PD77210 ファミリ ホスト) の例を示します。

```

/* HST, HDO は予約語ではないので, 明示的に定義する */
#define HST      0x3807
#define HDO      0x3806

/* 内部割り込み HO, HI を禁止する */
R0L = SR      ;
R0 = R0 | 0x0300 ;
SR = R0L      ;

R0L = 0x0      ; ホスト・ステータスを次のように設定する
*HST : X = R0L ; ・HDT アクセス・ウェイト機能を使用しない
              ; ・HRE 機能をマスクしない
              ; ・HWE 機能をマスクしない
              ; ・ユーザ・フラグをすべてクリア
              ; ・エラー・フラグをすべてクリア

POLL : R0L = *HST : X      ; HREF を判定し, ホストが HDT を読み出すまでループして待ち合わせる
R0 = R0 & 0x2      ;
if ( R0! = 0 ) jmp POLL ;

*HDO : X = R1H      ; HDT が空になったので R1H のデータを出力する

```

(b) ウェイト

次のような場合 HDT に対してロード/ストアを実行すると、命令実行にウェイトが発生します。

- ウェイト機能許可 (HAWE = 1) 時, HREF = 1 (HDT (out) に有効なデータが存在) にもかかわらず, HDT (out) に対してデータのストアが行われた場合。
- ウェイト機能許可 (HAWE = 1) 時, HWEF = 1 (HDT (in) に有効なデータが存在しない) にもかかわらず, HDT (in) からデータのロードが行われた場合。

この形式は、ハンドシェイク手順がハードウェアで自動的に実行されますから、アプリケーション・プログラムで特別なハンドシェイク手順記述を要しないところにメリットがあります。次に、ウェイトによるホスト・リード例を示します。

```

/* HST, HDO は予約語ではないので、明示的に定義する */
#define HST      0x3807
#define HDO      0x3806

/* 内部割り込み HO, HI を禁止する */
R0L = SR      ;
R0 = R0 | 0x0300 ;
SR = R0L      ;

R0L = 0x0400   ; ホスト・ステータスを次のように設定する
*HST : X = R0L ; ・HDT アクセス・ウェイト機能を使用する
               ; ・HRE 機能をマスクしない
               ; ・HWE 機能をマスクしない
               ; ・ユーザ・フラグをすべてクリア
               ; ・エラー・フラグをすべてクリア

*HDO : X = R1H   ; R1H のデータを出力する。HDT ビジィであれば
               ; ウェイトがかかる

```

注意 アプリケーション・プログラムから HDT へ書き込んだ場合、HDT を外部から読み出さないかぎりウェイトが解除されることはありません。したがって HDT に対して、 μ PD77210 ファミリ内部からの書き込みと外部からの読み出しが 1 対 1 に対応しない場合は、ハングアップする可能性があります。また、ウェイト中は割り込みが遅延されます (4.4.4 割り込み参照)。

(c) 割り込み

μ PD77210 ファミリ側で HDT (out) にデータを書き込むことができる場合、および、HDT (in) からデータを読み出すことができる場合に、割り込みがかかることでハンドシェークを確立します。したがって、ほかの処理を実行中でも、それらの処理とは無関係に（非同期に）ホスト入出力を実行することができます。次に、割り込みを使用したホスト入出力の例を示します。

```

/* ホスト I/O を定義する */
#define HST      *0x3807 : X
#define HDO      *0x3806 : X
#define HDI      *0x3806 : X

/* 割り込みベクタ・テーブルのエントリ */
SegHi IMSEG AT 0x230      ; HIO 入力割り込みルーチン

        R0H = HDI          ; HDT (in) から読み出し
        *DP0 + + = R0H     ; バッファへ待避
        RETI               ; 割り込みから復帰
        NOP                ;

SegHo IMSEG AT 0x234      ; HIO 出力割り込みルーチン
        R0H = *dp4 + +     ; バッファから読み出し
        HDO = R0H          ; HDT (out) へ書き込み
        RETI               ; 割り込みから復帰
        NOP                ;

/* ホスト入出力を初期化するために割り込みを禁止 */
        R1L = EIR          ; すべての割り込みを禁止
        R1 = R1 | 0x8000   ; EI = 1
        EIR = R1L         ;

        NOP                ; EI = 1 が有効になるまで 2 ウェイト・サイクル必要
        NOP                ;

        R0L = SR           ; HI および HO 割り込み許可
        R0 = R0 & 0xFCFF   ;
        SR = R0L          ;

        R1 = R1 & 0x7FFF   ; すべての割り込み許可
        EIR = R1L         ;
        FINT               ; 前の割り込みを廃棄

                                ; HDT を初期化
        R0L = 0x0          ; HDT アクセス・ウェイト機能なし
        HST = R0L         ; HRE, HWE マスクなし, ユーザ・エラー・フラグをクリア
        HD0 = R0L         ; ダミー・ストア (注意参照)

```

注意 ホスト出力の割り込みは、HDT レジスタの上位バイト・アクセス時の $\overline{\text{HRD}}$ 端子の立ち上がりによって発生するため、次の点に注意してください。

- (1) 最初にデータを転送するときは、割り込みのモードによっては、ダミーのデータを転送して強制的に割り込みを発生させるか、最初のデータの転送は割り込みを使用しないで行います。
- (2) パーストで転送するときに、パースト・データの最終ワードに発生する割り込みタイミングでは、次のパースト転送の最初のデータは生成されていない可能性が高いため、最終ワードを HDT に転送する直前の割り込みで割り込みを禁止し、次のパースト・データの生成が終わった時点で(1)と同様の命令を実行し、次のパースト・データを転送してください。

5.7 外部データ・メモリ・インタフェース (MIO)

外部データ・メモリ・インタフェース (MIO) は、外部メモリとのインタフェースを行います。MIO により 1M ワード (20 ビット・アドレス) の外部データ・メモリ空間へのアクセスが可能になります。

主な特徴を次に示します。

- **ダイレクト・アクセス**

データ・ページ・レジスタ DPR を 0x3F に設定することにより、 μ PD77210 ファミリ内部データ・メモリ空間 0x8000-0xFFFF を介して外部データ・メモリへアクセスします。

アクセス時にインデクス・アドレス加算することにより、20 ビットのアドレッシングを行います。

- **MDT (Memory Data register) アクセスとアドレス自動生成機能**

あらかじめアクセス開始のアドレスを設定し、MDT に対するロード/ストアにより外部データ・メモリへアクセスします。

アドレス生成には、初期アドレスをホールド (+0 インクリメント) する 0 次元アクセス、初期アドレスから +1 インクリメントする 1 次元アクセス、ライン長とオフセット長をパラメータとした 2 次元アクセスがあります。

- **タイミング調整機能**

外部データ・メモリ・インタフェースでは、アクセス・サイクル、メモリ・ライトに対するデータ/アドレス信号のセットアップ、ホールド・サイクルをレジスタ設定によりプログラマブルに設定可能です。

外部データ・メモリ・インタフェースのブロック図を図 5-17に、レジスタ群を表 5-16に示します。

図 5-17 外部データ・メモリ・インタフェースのブロック図

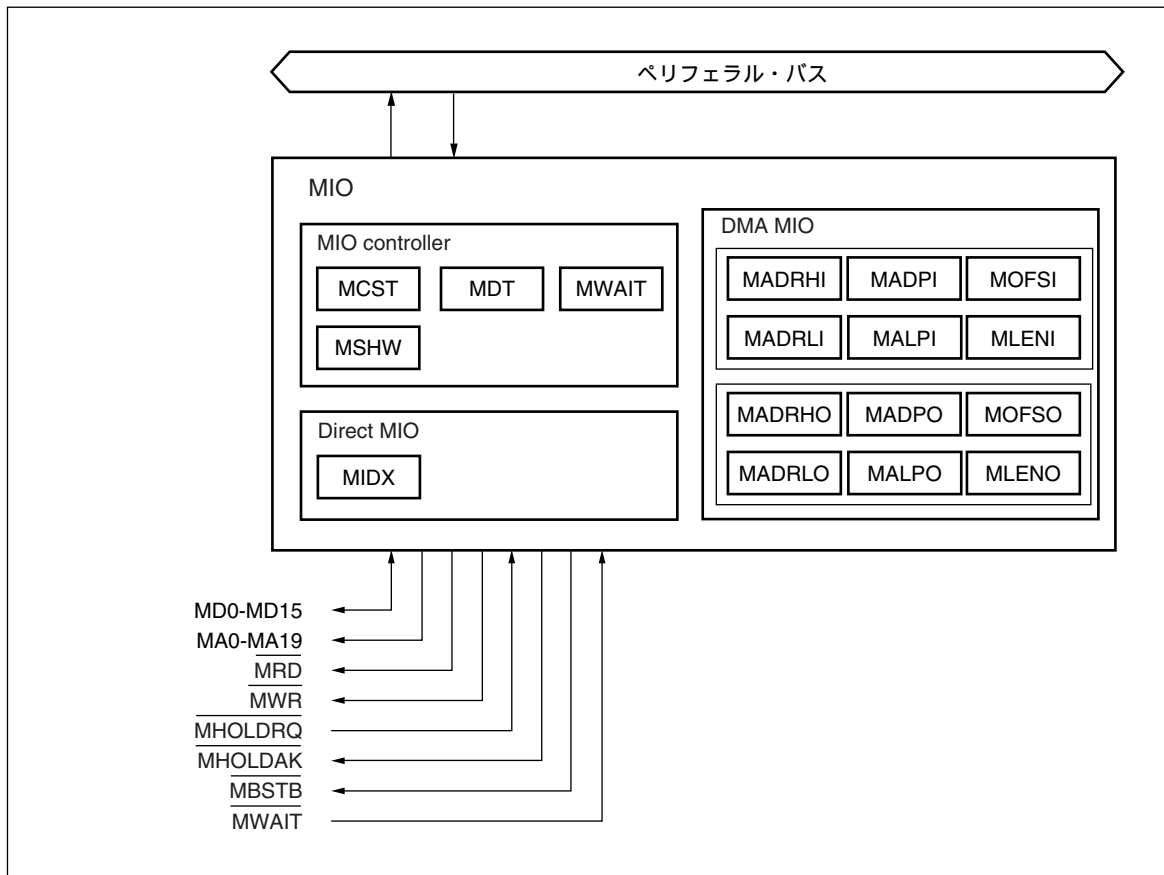


表 5-16 外部データ・メモリ・インタフェースのレジスタ

X/Y メモリ・アドレス	レジスタ名	機能	ロード/ストア
0x3840	MDT	メモリ・データ・レジスタ	L/S
0x3841	MSHW	メモリ I/F セットアップ / ホールド幅設定レジスタ	L/S
0x3842	MCST	メモリ I/F コントロール / ステータス・レジスタ	L/S
0x3843	MWAIT	メモリ I/F ウェイト・レジスタ	L/S
0x3844	MIDX	ダイレクト・アクセス・インデクス・レジスタ	L/S
0x3845	MADRLI	メモリ I/F 入力用スタート・アドレス・レジスタ (下位)	L/S ^注
0x3846	MADRHI	メモリ I/F 入力用スタート・アドレス・レジスタ (上位)	L/S ^注
0x3847	MOFSI	メモリ I/F 入力用ライン・オフセット・レジスタ	L/S ^注
0x3848	MLENI	メモリ I/F 入力用ライン・レンジス・レジスタ	L/S ^注
0x3849	MADRLO	メモリ I/F 出力用スタート・アドレス・レジスタ (下位)	L/S ^注
0x384A	MADRHO	メモリ I/F 出力用スタート・アドレス・レジスタ (上位)	L/S ^注
0x384B	MOFSO	メモリ I/F 出力用ライン・オフセット・レジスタ	L/S ^注
0x384C	MLENO	メモリ I/F 出力用ライン・レンジス・レジスタ	L/S ^注

注 μ PD77210 ではロード不可

5.7.1 メモリ・インタフェースの端子

(1) MA0-MA19 (メモリ・アドレス - 出力)

20 ビットのアドレス端子です。バス解放中はハイ・インピーダンスになります。

(2) MD0-MD15 (メモリ・データ - 入出力)

16 ビットのデータ入出力端子です。バス解放中、および外部データ・メモリをアクセスしないときはハイ・インピーダンスになります。

(3) $\overline{\text{MRD}}$ (メモリ・リード・ストロープ - 出力)

外部データ・メモリのリード・ストロープ出力端子です。バス解放中はハイ・インピーダンスになります。

(4) $\overline{\text{MWR}}$ (メモリ・ライト・ストロープ - 出力)

外部データ・メモリのライト・ストロープ出力端子です。バス解放中はハイ・インピーダンスになります。

(5) $\overline{\text{MHOLDRQ}}$ (バス・ホールド・リクエスト - 入力)

外部データ・メモリ・バスの占有要求信号の入力端子です。複数のデバイスでバスを共有するシステムで、バス・アービトレーションを行うときに使用します。

$\overline{\text{MHOLDRQ}}$ をロウ・レベルにすると、進行中のバス・サイクル終了後にバスは解放されます。

(6) $\overline{\text{MHOLDAK}}$ (バス・ホールド・アクノリッジ - 出力)

外部データ・メモリ・バスを共有しているデバイスに対して、バスの使用許可信号を出力する端子です。外部デバイスにバスを解放している間、ロウ・レベルを出力します。

(7) $\overline{\text{MBSTB}}$ (バス・ストロープ - 出力)

外部データ・メモリ・バスの使用要求信号を出力する端子です。

(8) $\overline{\text{MWAIT}}$ (メモリ・ウエイト - 入力)

アクセス・ウエイト・サイクル挿入の入力端子です。外部データ・メモリをアクセスするときに、この端子をロウ・レベルにするとウエイト・サイクルを挿入します。

$\overline{\text{MRD}}$ 、 $\overline{\text{MWR}}$ の立ち下がり前および、バス解放中の $\overline{\text{MWAIT}}$ 端子に、ウエイト・サイクルを挿入することを禁止します。

5.7.2 外部データ・メモリ・インタフェースのレジスタ

(1) MDT (メモリ・データ・レジスタ : MIO Data Transfer Register)

外部データ・メモリ・インタフェースとのデータを入出力するための 16 ビット・レジスタです。

(2) MSHW (メモリ・インタフェース・セットアップ/ホールド幅レジスタ : MIO Setup Hold Width Register)

メモリ・リード/ライト・ストロープのセットアップ/ホールド・サイクルを設定するための 16 ビットのレジスタです。サイクル単位で、メモリ・リード/ライト・ストロープのセットアップ/ホールドのタイミングを調整することができます。

MSHW の各ビットの機能については表 5-17に示します。デフォルトは 0xFFFF です。次のような注意があります。

- μ PD77213 の場合、メモリ・リード時は、ホールド・サイクルは制御しません（0 サイクル固定）。
- μ PD77210 の場合、メモリ・リード時はセットアップ/ホールド・サイクルともに制御しません（0 サイクル固定）。
- セットアップ/ホールドの設定値を 0 としたアクセスを禁止します。

(3) MWAIT (メモリ・インタフェース・ウエイト・レジスタ : MIO WAIT Register)

外部データ・メモリ・アクセス時にウエイトを挿入するウエイト・サイクル数を設定するための 16 ビットのレジスタです。

MWAIT の各ビットの機能については表 5-18に示します。デフォルトは 0xFFFF です。

ウエイトの設定値を 0 としたアクセスを禁止します。

(4) MCST (メモリ・インタフェース制御/ステータス・レジスタ : MIO Control/Status Register)

メモリ・インタフェースのステータスや DMA 転送の設定を行う 8 ビットのレジスタです。

MCST の各ビットの機能については表 5-19に示します。

(5) MIDX (ダイレクト・アクセス・インデクス・レジスタ : MIO Direct Access InDeX Register)

ダイレクト・アクセス時のアドレス修飾用の 8 ビットのレジスタです。

DPR=0x3F としたデータ・メモリ 0x8000-0xFFFF を外部データ・メモリ・アクセスのウィンドウとし、このレジスタの値を上位 8 ビットとした値をアドレスに加算した 20 ビット・アドレスでアクセスを行います。

(6) MADR (メモリ・インタフェース・スタート・アドレス・レジスタ : MIO Start Address Register)

DMA アクセス時の外部メモリのスタート・アドレスを保持するレジスタです。

20 ビット・アドレッシングを行うため、上位 4 ビットと下位 16 ビットとの 2 本のレジスタがあります。また、入力用、出力用を別々に持っています。MCST レジスタ制御により MAPI あるいは MAPO に転送されません。

MADRLI, MADRHI は入力用のスタート・アドレス・レジスタです。MADRLI が下位 16 ビット, MADRHI が上位 4 ビットを示します。

MADRLO, MADRHO は出力用のスタート・アドレス・レジスタです。MADRLO が下位 16 ビット, MADRHO が上位 4 ビットを示します。

注意 μ PD77210 はロード不可

(7) MOFS (メモリ・インタフェース・ライン・オフセット・レジスタ : MIO Line OffSet Register)

MDT アクセス用のライン・インターバル・オフセットを設定するレジスタです。0 次元アクセス時は 0x0, 1 次元アクセス時は 0x1, 2 次元アクセス時はオフセット長を設定します。

入力用 (MOFSI), 出力用 (MOFSO) を別々に持っています。

注意 μ PD77210 はロード不可

(8) MLEN (メモリ・インタフェース・ライン・レングス・レジスタ : MIO Line LENgth Register)

MDT アクセス用のライン長を設定するレジスタです。0 次元, 1 次元アクセス時は 0x1, 2 次元アクセス時はライン長を設定します。

入力用 (MLENI), 出力用 (MLENO) を別々に持っています。

注意 μ PD77210 はロード不可

(9) MADP (DMA アクセス・データ・アドレス・レジスタ)

DMA アクセス時のアドレスを示す 20 ビットのレジスタです。入力用, 出力用を別々に持っています。

MADPI は入力用のアドレス・レジスタ, MADPO は出力用のアドレス・レジスタです。

MCST へのストア命令により, MADR の値を MADP に転送します。MADP の値は, MDT のアクセスによって自動的に更新されます。

MADP はペリフェラル・バスとは接続していません。

(10) MALP (DMA アクセス・データ残りレングス・レジスタ)

DMA アクセス時の残りの転送ライン数を示す 16 ビットのレジスタです。入力用, 出力用を別々に持っています。

MALPI は入力用の残り転送ライン数レジスタ, MALPO は出力用の残り転送ライン数レジスタです。

MCST へのストア命令により, MLEN の値を MALP に転送します。MALP の値は, MDT のアクセスによって自動的に更新されます。

MALP はペリフェラル・バスとは接続していません。

表 5-17 MSHW のビット構成

ビット	名称	機能	ロード/ ストア
15, 14	Dside	D 面 (0xC0000-0xFFFFF) アクセス・セットアップ・サイクル ・ 0x0 : 0 サイクル ・ 0x1 : 1 サイクル ・ 0x2 : 2 サイクル ・ 0x3 : 3 サイクル (デフォルト)	L/S
13, 12		D 面 (0xC0000-0xFFFFF) アクセス・ホールド・サイクル ・ 0x0 : 0 サイクル ・ 0x1 : 1 サイクル ・ 0x2 : 2 サイクル ・ 0x3 : 3 サイクル (デフォルト)	L/S
11, 10	Cside	C 面 (0x80000-0xBFFFF) アクセス・セットアップ・サイクル ・ 0x0 : 0 サイクル ・ 0x1 : 1 サイクル ・ 0x2 : 2 サイクル ・ 0x3 : 3 サイクル (デフォルト)	L/S
9, 8		C 面 (0x80000-0xBFFFF) アクセス・ホールド・サイクル ・ 0x0 : 0 サイクル ・ 0x1 : 1 サイクル ・ 0x2 : 2 サイクル ・ 0x3 : 3 サイクル (デフォルト)	L/S
7, 6	Bside	B 面 (0x40000-0x7FFFF) アクセス・セットアップ・サイクル ・ 0x0 : 0 サイクル ・ 0x1 : 1 サイクル ・ 0x2 : 2 サイクル ・ 0x3 : 3 サイクル (デフォルト)	L/S
5, 4		B 面 (0x40000-0x7FFFF) アクセス・ホールド・サイクル ・ 0x0 : 0 サイクル ・ 0x1 : 1 サイクル ・ 0x2 : 2 サイクル ・ 0x3 : 3 サイクル (デフォルト)	L/S
3, 2	Aside	A 面 (0x00000-0x3FFFF) アクセス・セットアップ・サイクル ・ 0x0 : 0 サイクル ・ 0x1 : 1 サイクル ・ 0x2 : 2 サイクル ・ 0x3 : 3 サイクル (デフォルト)	L/S
1, 0		A 面 (0x00000-0x3FFFF) アクセス・ホールド・サイクル ・ 0x0 : 0 サイクル ・ 0x1 : 1 サイクル ・ 0x2 : 2 サイクル ・ 0x3 : 3 サイクル (デフォルト)	L/S

注意 セットアップ/ホールドを 0 にしたときの該当メモリ領域へのアクセスを禁止します。

表 5-18 MWait のビット構成

ビット	名称	機能	ロード/ ストア
15-12	Dside	D 面 (0xC0000-0xFFFF) アクセス・ウェイト・サイクル ・0x0 : 0 サイクル ・0x1 : 1 サイクル ・0x2 : 2 サイクル ・0x3 : 3 サイクル ・0x4 : 4 サイクル ・0x5 : 5 サイクル ・0x6 : 6 サイクル ・0x7 : 7 サイクル ・0x8 : 8 サイクル ・0x9 : 9 サイクル ・0xA : 10 サイクル ・0xB : 11 サイクル ・0xC : 12 サイクル ・0xD : 13 サイクル ・0xE : 14 サイクル ・0xF : 15 サイクル (デフォルト)	L/S
11-8	Cside	C 面 (0x80000-0xBFFFF) アクセス・ウェイト・サイクル ・0x0 : 0 サイクル ・0x1 : 1 サイクル ・0x2 : 2 サイクル ・0x3 : 3 サイクル ・0x4 : 4 サイクル ・0x5 : 5 サイクル ・0x6 : 6 サイクル ・0x7 : 7 サイクル ・0x8 : 8 サイクル ・0x9 : 9 サイクル ・0xA : 10 サイクル ・0xB : 11 サイクル ・0xC : 12 サイクル ・0xD : 13 サイクル ・0xE : 14 サイクル ・0xF : 15 サイクル (デフォルト)	L/S
7-4	Bside	B 面 (0x40000-0x7FFFF) アクセス・ウェイト・サイクル ・0x0 : 0 サイクル ・0x1 : 1 サイクル ・0x2 : 2 サイクル ・0x3 : 3 サイクル ・0x4 : 4 サイクル ・0x5 : 5 サイクル ・0x6 : 6 サイクル ・0x7 : 7 サイクル ・0x8 : 8 サイクル ・0x9 : 9 サイクル ・0xA : 10 サイクル ・0xB : 11 サイクル ・0xC : 12 サイクル ・0xD : 13 サイクル ・0xE : 14 サイクル ・0xF : 15 サイクル (デフォルト)	L/S
3-0	Aside	A 面 (0x00000-0x3FFFF) アクセス・ウェイト・サイクル ・0x0 : 0 サイクル ・0x1 : 1 サイクル ・0x2 : 2 サイクル ・0x3 : 3 サイクル ・0x4 : 4 サイクル ・0x5 : 5 サイクル ・0x6 : 6 サイクル ・0x7 : 7 サイクル ・0x8 : 8 サイクル ・0x9 : 9 サイクル ・0xA : 10 サイクル ・0xB : 11 サイクル ・0xC : 12 サイクル ・0xD : 13 サイクル ・0xE : 14 サイクル ・0xF : 15 サイクル (デフォルト)	L/S

注意 ウェイトを 0 にしたときの該当メモリ領域へのアクセスを禁止します。

表 5-19 MCST のビット構成

ビット	名称	機能	ロード/ ストア
7	IDS	入力 MADP セット要求フラグ ・0: リクエストなし (デフォルト) ・1: リクエスト中 1 にすると, MADRLI+MADRHI→MADPI を行う。実行後に 0 になる。	L/S
6	ODS	出力 MADP のセット要求フラグ ・0: リクエストなし (デフォルト) ・1: リクエスト中 1 にすると, MADRLO+MADRHO→MADPO を行う。実行後に 0 になる。	L/S
5	ILS	入力 MALP のセット要求フラグ ・0: リクエストなし (デフォルト) ・1: リクエスト中 1 にすると, MLENI→MALPI を行う。実行後に 0 になる。	L/S
4	OLS	出力 MALP のセット要求フラグ ・0: リクエストなし (デフォルト) ・1: リクエスト中 1 にすると, MLENO→MALPO を行う。実行後に 0 になる。	L/S
3	WM	メモリ・ウエイト・リクエスト・モニタ $\overline{\text{MWAIT}}$ 端子のモニタ。 ・0: リクエスト中 ・1: リクエスト中ではない	L
2	HR	メモリ・ホールド・リクエスト・モニタ $\overline{\text{MHOLDRQ}}$ 端子のモニタ。 ・0: リクエスト中ではない ・1: リクエスト中	L
1	SB	メモリ・アクセス・モニタ $\overline{\text{MBSTB}}$ 端子のモニタ。 ・0: アクセス中ではない ・1: アクセス中	L
0	HA	メモリ・ホールド・アクノリッジ・モニタ $\overline{\text{MHOLDAK}}$ 端子のモニタ。 ・0: アクセス中ではない ・1: アクセス中	L

5.7.3 ダイレクト・アクセス

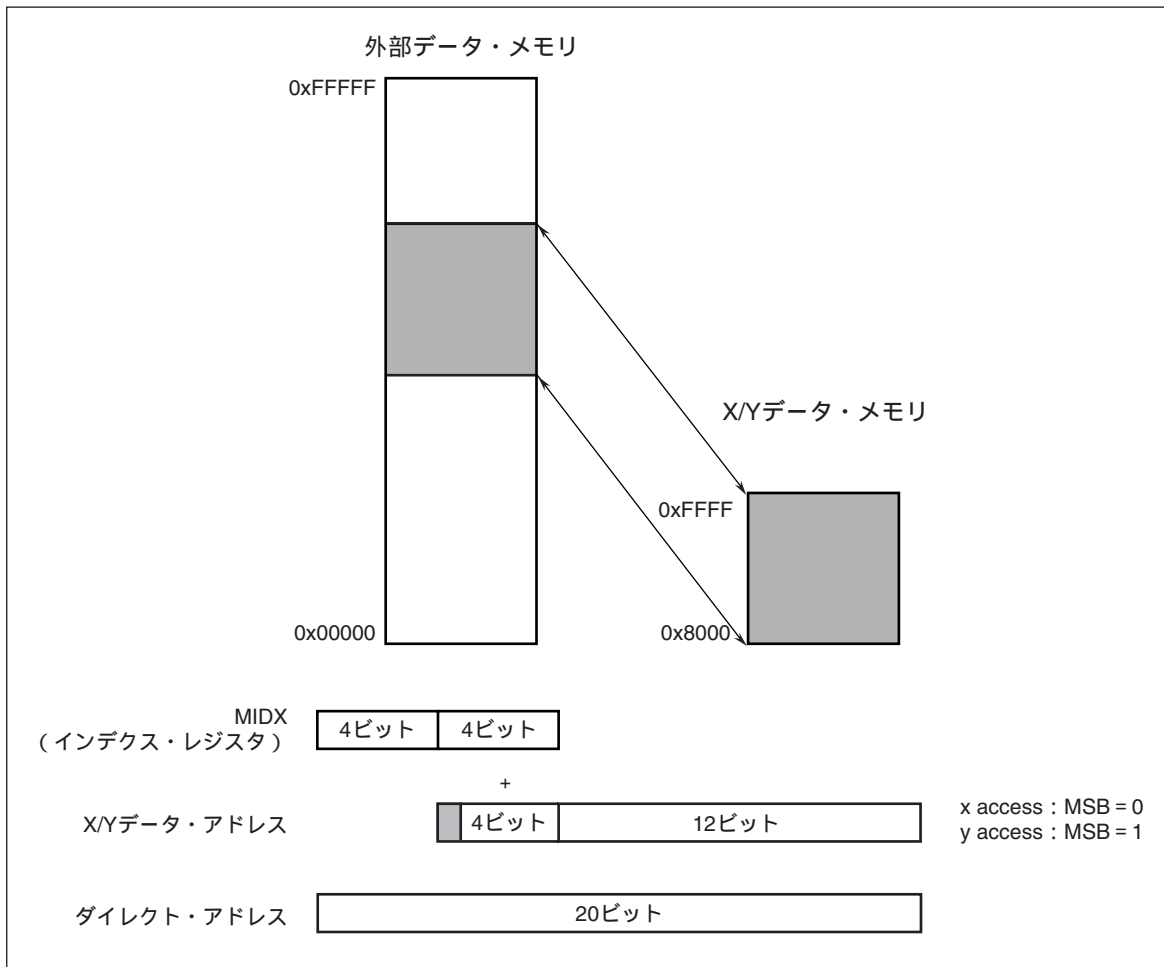
外部データ・メモリ・アクセス・ウィンドウを介して直接外部データ・メモリをアクセスする方法です。外部データ・メモリ・ウィンドウは DPR を 0x3F としたときの X あるいは Y データ・メモリの 0x8000-0xFFFF に割り当てられます。

外部データ・メモリ空間は 20 ビット・アドレス空間 (1M ワード × 16 ビット) であり, アドレスの生成は次のようにして行います。

- MIDX (ダイレクト・アクセス・インデクス・レジスタ) を 20 ビット・アドレスの上位 8 ビットとします。
- X あるいは Y データ・メモリの 0x8000-0xFFFF (DPR = 0x3F) をアクセスするときのアドレスを MIDX に加算し, 20 ビットのアドレスとします。
- この際, X データ側からのアクセスの場合は 16 ビットのアドレスの MSB を 0 とし 0x0000-0x7FFF で MIDX と加算, Y データ側からのアクセスの場合は 16 ビットのアドレスの MSB を 1 とし 0x8000-0xFFFF で MIDX と加算します。

図 5-18にダイレクト・アクセスのイメージを示します。

図 5-18 ダイレクト・アクセス



5.7.4 DMA アクセス

メモリ・マップトされている MDT (メモリ・データ・レジスタ) を介して外部データ・メモリにアクセスする方法です。

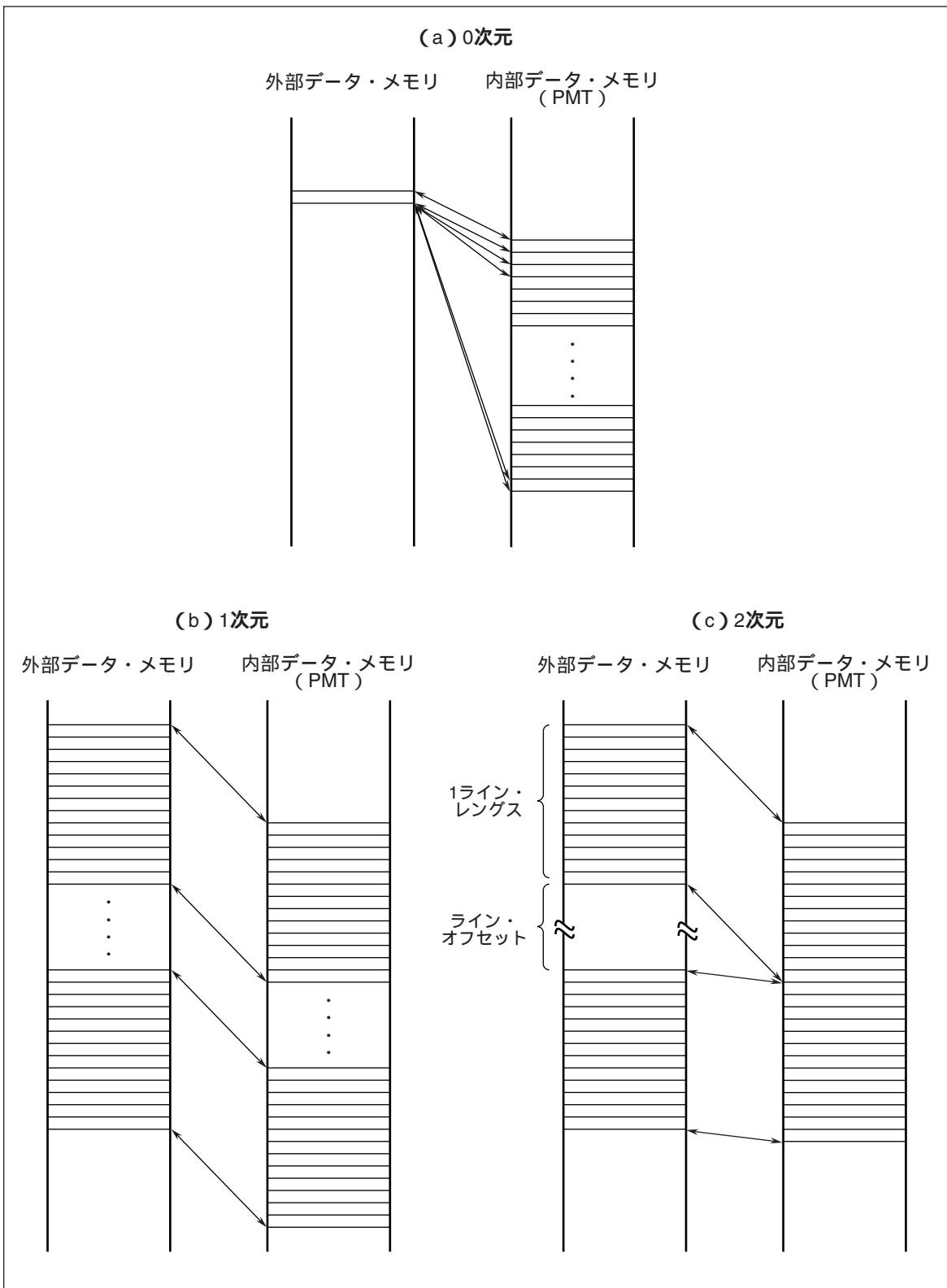
MADR によって設定された MADP の値をアドレスにして、外部データ・メモリにアクセスします。MADR は DMA 転送の開始アドレスを示すレジスタであり、MCST の該当ビットを設定することにより MADR→MADP の転送を行います。MDT に対するアクセスがあると、自動的に MADP の値を更新します。更新の方法には次の 3 種類があります。

- 現在の値をホールドする 0 次元のアドレッシング (0D)
特定の外部メモリ・マップト・ペリフェラルなどへのアクセスを想定しています。
- 単純にインクリメントする 1 次元のアドレッシング (1D)
外部メモリと内部メモリの間のブロック転送などを想定しています。
- ライン長ごとにオフセット加算する 2 次元アドレッシング (2D)
画像を取り扱うフレーム・メモリのようなメモリの矩形領域へのアクセスを想定しています。

2 次元アドレッシングの際のライン長およびオフセット長は、MOFS および MLEN で設定します。0 次元アドレッシング時はそれぞれ 0x0, 0x1 を設定します。1 次元アドレッシング時はそれぞれ 0x1, 0x1 を設定します。

図 5-19 に DMA アクセスのイメージを示します。

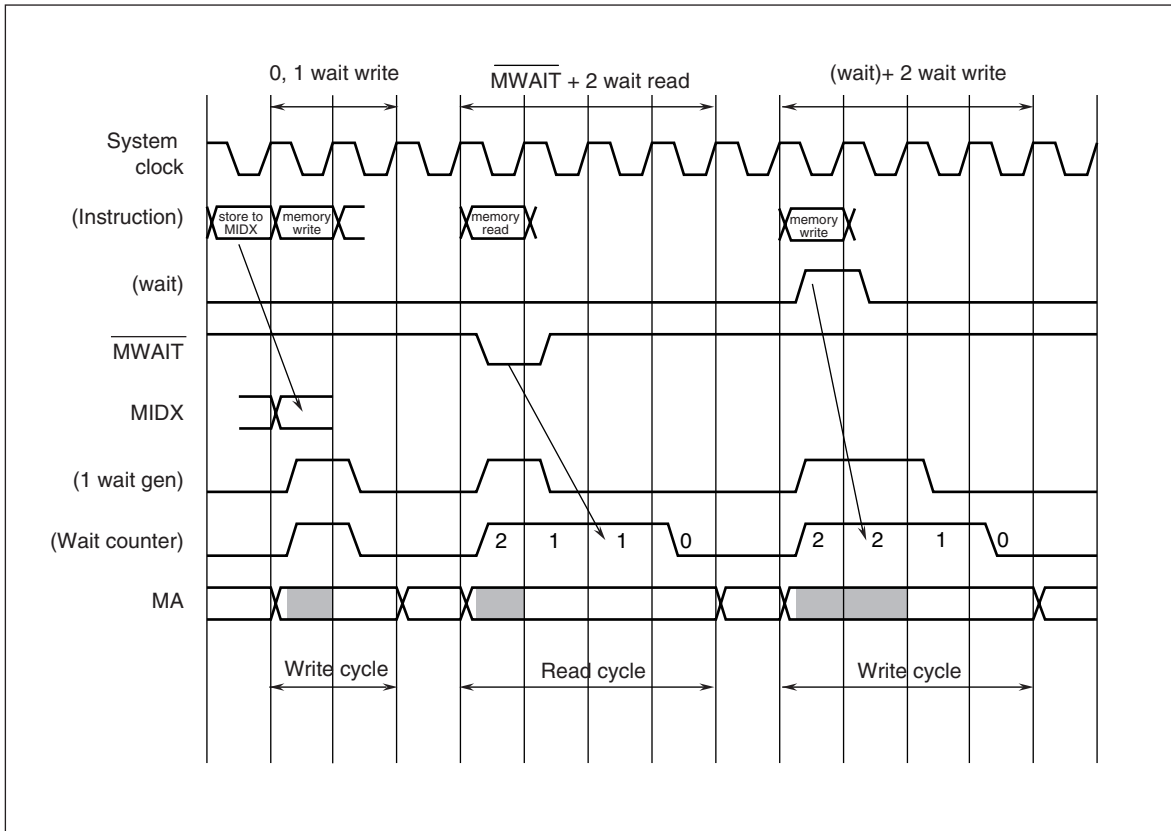
図 5-19 DMA アクセス



5.7.5 ダイレクト・アクセスのタイミング

図 5-20にダイレクト・アクセスのタイミングを示します。

図 5-20 ダイレクト・アクセスのタイミング



MIDX へのストア命令はダイレクト・アクセスの直前で可能です。

MWAIT レジスタによるプログラマブル・ウエイト (Wait counter) とは別に 1 ウエイト発生回路 (1wait gen) を持ち、ロード/ストアの先頭サイクルでインデックス加算しアドレス・ポインタを更新します。したがって、プログラマブル・ウエイトの設定にかかわらず 1 ウエイトは発生します。

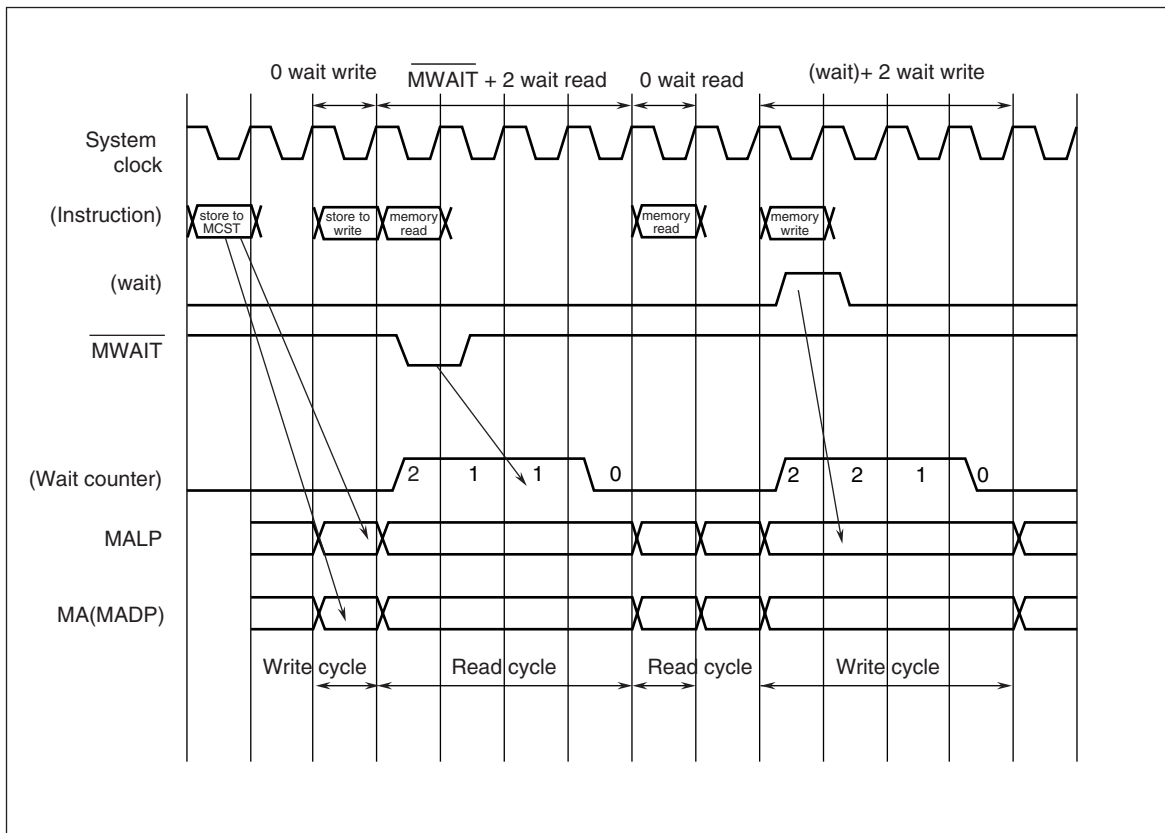
外部 M $\overline{\text{WAIT}}$ 端子がアサートされている間は、アクセス・サイクルを延長します (M $\overline{\text{WAIT}}$ 端子は、アクセス・サイクル中にサンプリングされ、プログラマブル・ウエイトのカウンタ更新を保留します。 M $\overline{\text{WAIT}}$ 端子はプログラマブル・ウエイト以上のアクセス・サイクルを実現するものであり十分なウエイト・サイクルを設定していることを前提としています)。

DSP コア・カーネル・アクセスによるウエイトがアサートされている間は、プログラマブル・ウエイトおよびアドレス加算ステージの 1 ショット・ウエイトも合わせアクセス・サイクルを延長します。

5.7.6 DMA アクセスのタイミング

図 5-21に DMA アクセスのタイミングを示します。

図 5-21 DMA アクセスのタイミング



MCST レジスタ・ストアにより初期アドレス MADP、残り転送ライン数 MALP をセットします(MADR、MOFS、MLEN へのストアを先行して行っておく)。確定するのは2 サイクル後となり、MCST ストア直後でのアクセスは不可能になります。アクセス・サイクル完了とともに MADP、MALP は更新されます。

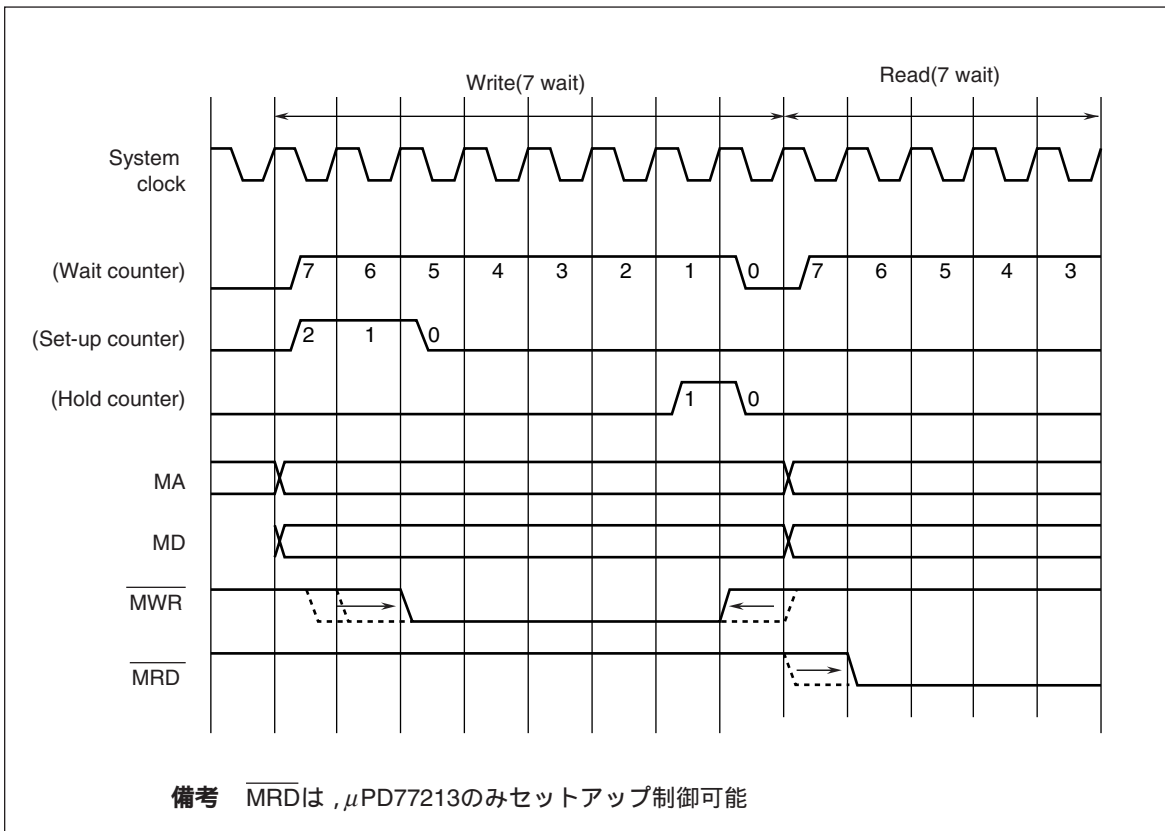
外部 M $\overline{\text{WAIT}}$ 端子がアサートされている間は、アクセス・サイクルを延長します(M $\overline{\text{WAIT}}$ 端子は、アクセス・サイクル中にサンプリングされ、プログラマブル・ウエイトのカウンタ更新を保留します。M $\overline{\text{WAIT}}$ 端子はプログラマブル・ウエイト以上のアクセス・サイクルを実現するものであり十分なウエイト・サイクルを設定していることを前提としています)。

DSP コア・カーネル・アクセスによるウエイトがアサートされている間は、プログラマブル・ウエイトのカウンタ更新を保留し、アクセス・サイクルを延長します。

5.7.7 メモリ・アクセスのタイミング

図 5-22にメモリ・アクセスのタイミングを示します。

図 5-22 メモリ・アクセスのタイミング



メモリ・ライト・ストロープ ($\overline{\text{MWR}}$) は、アドレス・バス (MADD)、メモリ・データ・バス (MD) に対して、誤書き込みの危険性があるため、セットアップ、ホールド時間を確保する必要があります。

本 MIO では、MSHW レジスタにセットアップ、ホールドの時間をマシン・サイクルで設定可能としています。ライト・サイクルの前後のストロープをマスクします。図の例はセットアップ 0x2、ホールド 0x1 とした場合です。マスク信号は、プログラマブル・ウエイトのカウント更新を保留するケースに合わせて延長します。なお、セットアップ側は、デコードによる髭を防止するためセットアップ 0 サイクル設定で 0.5 サイクルのセットアップ・サイクルを発生しています。

メモリ・リード・ストロープ ($\overline{\text{MRD}}$) に関しては、次のような制御になります。

- $\mu\text{PD77210}$: セットアップ、ホールドとも制御をしない
- $\mu\text{PD77213}$: セットアップのみ制御をする (ホールドは制御しない)

連続リード時はリード・アクセス・サイクル間アクティブ・ロウとなる信号とします。 $\mu\text{PD77213}$ においてセットアップが 0 以外で設定されている場合はハイ・レベルが挿入されます。また、DMA 転送時は 1 回ごとの転送でハイ・レベルが挿入されるが、DMA 転送の間にプログラム中からのリードが発生した場合は連続リードと同じになります。

ウエイト、セットアップ、ホールドを 0 とした設定での該当領域のアクセスを禁止します。最短でのメモリ・アクセスは、セットアップ、ホールドをともに 1 とし、ウエイトを 2 とした 3 サイクル・アクセスとなります。

外部端子 $\overline{\text{MWAIT}}$ を用いたウエイトの挿入と動作について説明します。 $\overline{\text{MRD}}$ 、 $\overline{\text{MWR}}$ 端子の立ち下がり後に

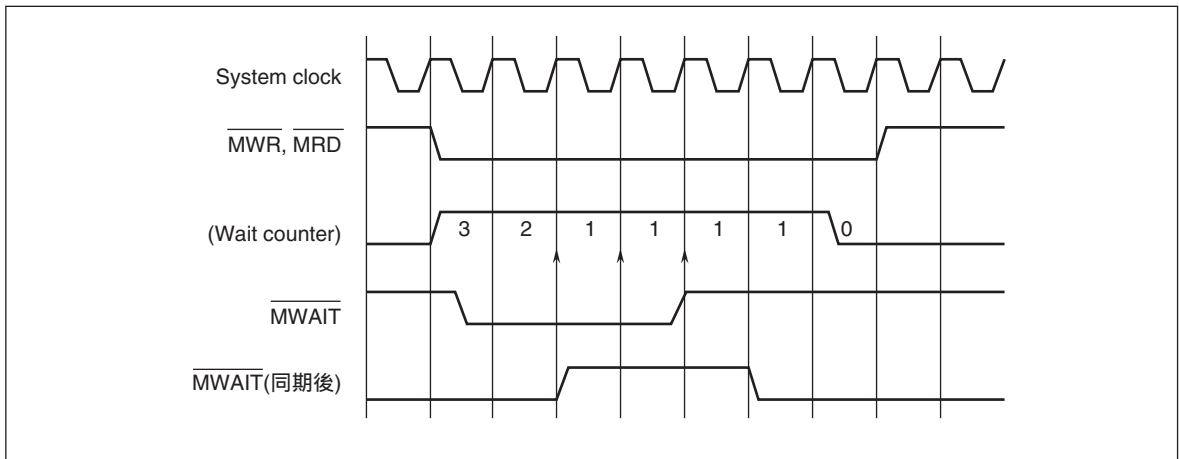
$\overline{\text{MWAIT}}$ 端子をアクティブ (ロウ・レベル) にすると同期化を行った信号でプログラブル・ウエイトのカウンタの更新を停止します。 $\overline{\text{MWAIT}}$ 端子をインアクティブ (ハイ・レベル) にするとカウンタの更新を再開するようになります。

★ $\overline{\text{MWAIT}}$ (同期後) は $\overline{\text{MWAIT}}$ を 1 回ではなく、2 回ラッチして生成する仕様となります。

$\overline{\text{MWAIT}}$ レジスタによるアクセス・ウエイト・サイクルの設定が 3 サイクル以上でなければ、 $\overline{\text{MWAIT}}$ によるウエイト挿入はできません。

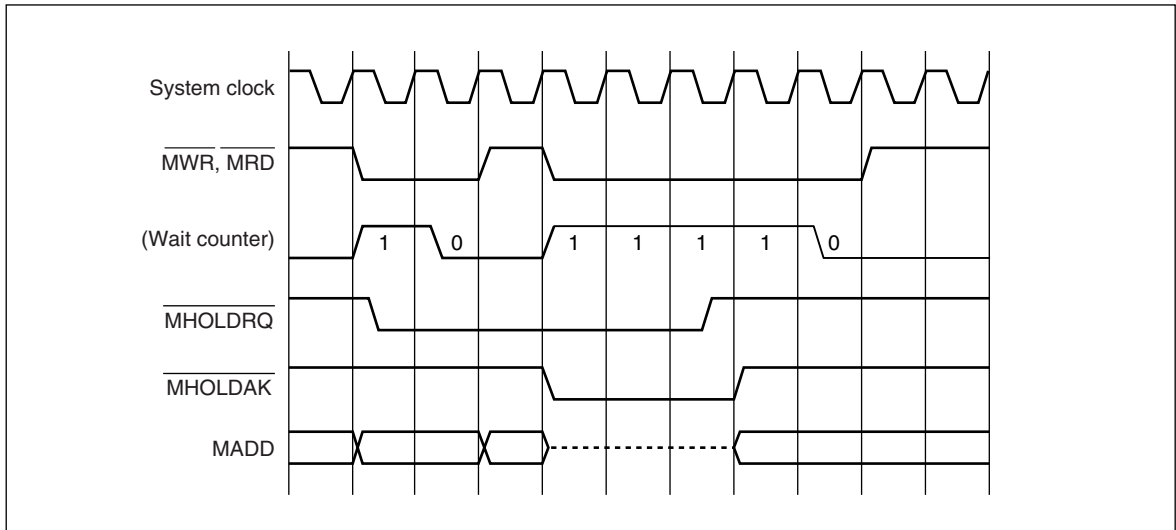
$\overline{\text{MRD}}$, $\overline{\text{MWR}}$ 端子の立ち下がり前あるいはバス解放中に $\overline{\text{MWAIT}}$ 端子をアクティブにすることを禁止します。

★ 図 5-23 $\overline{\text{MWAIT}}$ 端子によるアクセス・ウエイトの挿入



バス・アービトレーションについて説明します。 $\overline{\text{MHOLDRQ}}$ 端子をアクティブ (ロウ・レベル) にすると同期化を行ったのち、 μ PD77210 ファミリが外部データ・メモリを非アクセスである場合に $\overline{\text{MHOLDAK}}$ 端子を立ち下げます。するとバスを解放 (ハイ・インピーダンス) し、外部のデバイスがバスを使用することが可能になります。解放中に μ PD77210 ファミリが外部データ・メモリをアクセスする場合は、ウエイト・カウンタ更新を停止します。

図 5-24 バス・アービトレーション



5.8 ペリフェラル - メモリ間転送 (PMT)

ペリフェラル - メモリ間転送 (PMT : Peripheral Memory Transfer) は、TSIO, ASIO, HIO, MIO のペリフェラル回路を介していたデータを直接 μ PD77210 ファミリの内蔵メモリに転送する DMA コントローラです。入力, 出力それぞれ 4 チャンネルずつあり, 合計 8 チャンネルで構成されます。

PMT コントローラでアクセスできる μ PD77210 ファミリの内蔵メモリ領域は, 0x0000-0x37FF のデータ空間 (14K ワード) です。

この空間にデータをロード/ストアするだけで, ペリフェラル回路との入出力を自動的に行うことができます。

データ入力の場合は, 1 ワードのデータがペリフェラル回路に入力されたら, メモリに転送します。

出力の場合は, データがペリフェラル回路から出力されたら, 次のデータをメモリから 1 ワード持ってきます。

ペリフェラル回路は 1 ワードごとの入出力で割り込み信号を発生し, PMT への転送リクエストにします。指定したワード数を入出力したら, DSP コア・カーネルに割り込みが入ります。転送中の割り込み発生によって転送を中断することはありません。

PMT のブロック図を図 5-25 に, レジスタ群を表 5-20 に示します。

図 5-25 PMT のブロック図

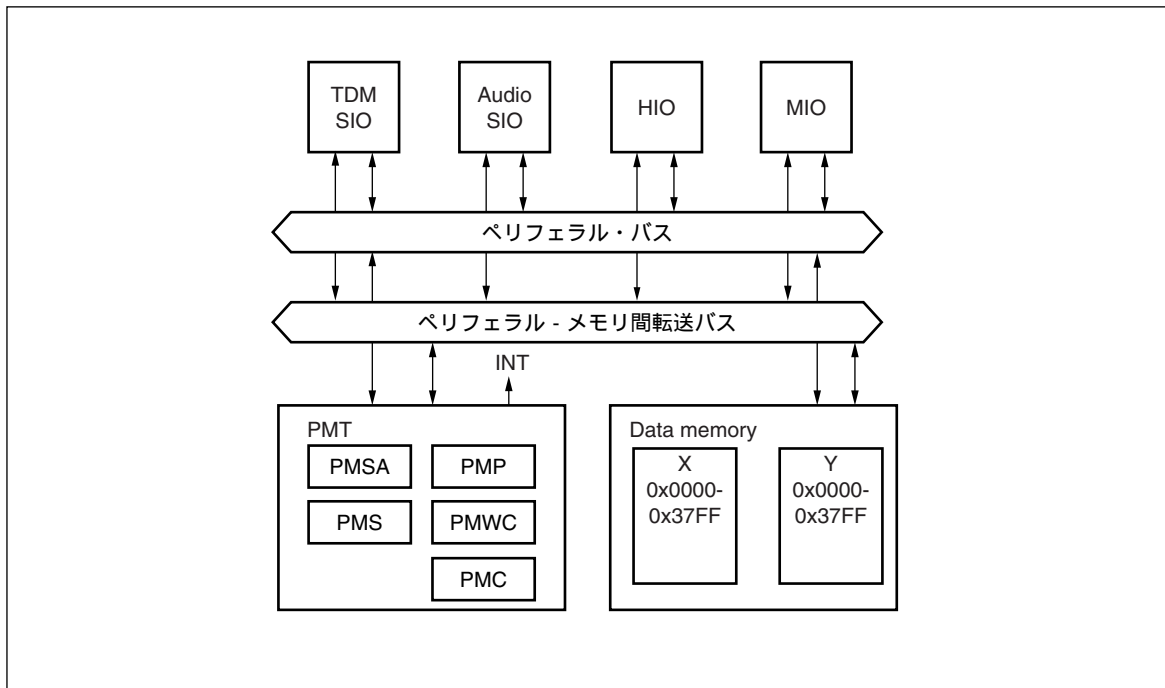


表 5-20 PMT のレジスタ

X/Y メモリ・アドレス	レジスタ名	機 能	ロード/ストア
0x3850	PMSA0	PMT スタート・アドレス・レジスタ 0	L/S
0x3851	PMS0	PMT サイズ・レジスタ 0	L/S
0x3852	PMC0	PMT 制御レジスタ 0	L/S
0x3853	PMP0	PMT アドレス・ポインタ 0	L
0x3854	PMSA1	PMT スタート・アドレス・レジスタ 1	L/S
0x3855	PMS1	PMT サイズ・レジスタ 1	L/S
0x3856	PMC1	PMT 制御レジスタ 1	L/S
0x3857	PMP1	PMT アドレス・ポインタ 1	L
0x3858	PMSA2	PMT スタート・アドレス・レジスタ 2	L/S
0x3859	PMS2	PMT サイズ・レジスタ 2	L/S
0x385A	PMC2	PMT 制御レジスタ 2	L/S
0x385B	PMP2	PMT アドレス・ポインタ 2	L
0x385C	PMSA3	PMT スタート・アドレス・レジスタ 3	L/S
0x385D	PMS3	PMT サイズ・レジスタ 3	L/S
0x385E	PMC3	PMT 制御レジスタ 3	L/S
0x385F	PMP3	PMT アドレス・ポインタ 3	L
0x3860	PMSA4	PMT スタート・アドレス・レジスタ 4	L/S
0x3861	PMS4	PMT サイズ・レジスタ 4	L/S
0x3862	PMC4	PMT 制御レジスタ 4	L/S
0x3863	PMP4	PMT アドレス・ポインタ 4	L
0x3864	PMSA5	PMT スタート・アドレス・レジスタ 5	L/S
0x3865	PMS5	PMT サイズ・レジスタ 5	L/S
0x3866	PMC5	PMT 制御レジスタ 5	L/S
0x3867	PMP5	PMT アドレス・ポインタ 5	L
0x3868	PMSA6	PMT スタート・アドレス・レジスタ 6	L/S
0x3869	PMS6	PMT サイズ・レジスタ 6	L/S
0x386A	PMC6	PMT 制御レジスタ 6	L/S
0x386B	PMP6	PMT アドレス・ポインタ 6	L
0x386C	PMSA7	PMT スタート・アドレス・レジスタ 7	L/S
0x386D	PMS7	PMT サイズ・レジスタ 7	L/S
0x386E	PMC7	PMT 制御レジスタ 7	L/S
0x386F	PMP7	PMT アドレス・ポインタ 7	L

PMT は 8 チャンネルの転送チャンネルを持っています。転送方向は転送チャンネルごとに固定されており、偶数チャンネルが「ペリフェラル→メモリ」、奇数チャンネルが「メモリ→ペリフェラル」になります。

表 5-21に PMT 転送チャンネルとペリフェラルの関係を示します。

表 5-21 PMT 転送チャンネルと対象ペリフェラル

PMT チャンネル	対象ペリフェラル	転送方向
0	TDM シリアル・インタフェース (入力)	ペリフェラル → メモリ
1	TDM シリアル・インタフェース (出力)	メモリ → ペリフェラル
2	オーディオ・シリアル・インタフェース (入力)	ペリフェラル → メモリ
3	オーディオ・シリアル・インタフェース (出力)	メモリ → ペリフェラル
4	ホスト・インタフェース (入力)	ペリフェラル → メモリ
5	ホスト・インタフェース (出力)	メモリ → ペリフェラル
6	メモリ・インタフェース (入力)	ペリフェラル → メモリ
7	メモリ・インタフェース (出力)	メモリ → ペリフェラル

5.8.1 PMT のレジスタ

(1) PMSA0-PMSA7 (PMT スタート・アドレス・レジスタ : PMT Start Address Register)

内部データ RAM 領域中に設けるバッファ領域の先頭アドレスを設定、保持する 16 ビット・レジスタです。PMT 転送開始時に PMP へ転送されます。このレジスタはチャンネルごとに存在し、デフォルトは 0x0 です。

(2) PMS0-PMS7 (PMT サイズ・レジスタ : PMT Size Register)

内部データ RAM 領域中に設けるバッファのサイズを設定する 16 ビットのレジスタです。PMT 転送開始時に PMWC へ転送されます。このレジスタはチャンネルごとに存在し、デフォルトは 0x0 です。

(3) PMC0-PMC7 (PMT 制御レジスタ : PMT Control Register)

PMT の動作を設定したり、ステータスを表すレジスタです。PMC の各ビットの機能については表 5-22 PMCのビット構成を参照してください。このレジスタはチャンネルごとに存在し、デフォルトは 0x0 です。

(4) PMP0-PMP7 (PMT アドレス・ポインタ : PMT Pointer)

PMT により転送対象となる内部データ RAM のアドレスを保持するレジスタです。ロードのみ可能です。このレジスタはチャンネルごとに存在し、デフォルトは 0x0 です。

(5) PMWC (PMT ワード・カウンタ : PMT Word Counter)

PMT により転送される残りの転送数を示す 16 ビット・レジスタです。このレジスタはチャンネルごとに存在します。

PMWC はペリフェラル・バスとは接続していません。ロード、ストアはできません。

表 5-22 PMC のビット構成

ビット	名称	機能	ロード/ ストア
15-7	予約	<ul style="list-style-type: none"> ・ 0 以外の書き込みは行わないこと。 ・ 読み出し時は不定。 	-
6	BM	転送モニタ・フラグ <ul style="list-style-type: none"> ・ 0 : 転送中 (ただし初期状態も 0) ・ 1 : 非転送中 EN を 1 に設定すると 0 になる。プログラム中で PMT 転送完了を判定する場合は{BM, EN}=10 を判断する。	L
5	ST	転送モード設定ビット <ul style="list-style-type: none"> ・ 0 : 連続転送 (デフォルト) ・ 1 : 転送停止 バッファが完了するたびに転送を停止するか、連続して転送するかを指定する。	L/S
4	EN	バッファリング・イネーブル・ビット <ul style="list-style-type: none"> ・ 0 : PMT 転送停止指示 (デフォルト) ・ 1 : PMT 転送開始指示 バッファリング動作の開始を設定します。ST が 1 の場合、転送完了後は 0 になる。	L/S
3, 2	IPT	割り込み周期設定ビット <ul style="list-style-type: none"> ・ 00 : フル, バッファ・サイズの 1/1 転送終了時 (デフォルト) ・ 01 : ハーフ, バッファ・サイズの 1/2 転送終了時 ・ 10 : クォータ, バッファ・サイズの 1/4 転送終了時 ・ 11 : リザーブ (設定禁止) 割り込み間隔を設定する。バッファの全部, 1/2, 1/4 フィルで割り込みを発生する。	L/S
1, 0	MSXY	バッファ・メモリ設定ビット <ul style="list-style-type: none"> ・ 00 : Xデータ・メモリ (デフォルト) ・ 01 : Yデータ・メモリ ・ 10, 11 : リザーブ (設定禁止) バッファとして使用するメモリを設定する。	L/S

- ★ 注意 PMT 転送停止指示において、EN ビットを 0 に設定しても、BM ビットは 1 になりません。EN ビットが 0 のとき、PMT 動作は、バッファリングを中断し、転送中状態で待機します。PMT の転送停止は、ST ビットを 1 にすることで行うことを推奨します。PMT の転送停止指示は ST ビット=1 (転送モードを転送停止) として PMT を停止し、BM ビット=1 (非転送中) かつ EN ビット=0 (PMT 転送停止指示) で転送が完了したことを判断してください。また、内部メモリ ベリフェラルへの PMT では、転送停止はベリフェラルへ最終データが転送された段階で完了となります。ベリフェラルから最終データが出力されたかどうかは、必要に応じて、ユーザ・プログラムで判断してください。

5.8.2 PMT の動作モード

先頭アドレスと、バッファのサイズを指定します（図 5-26を参照）。

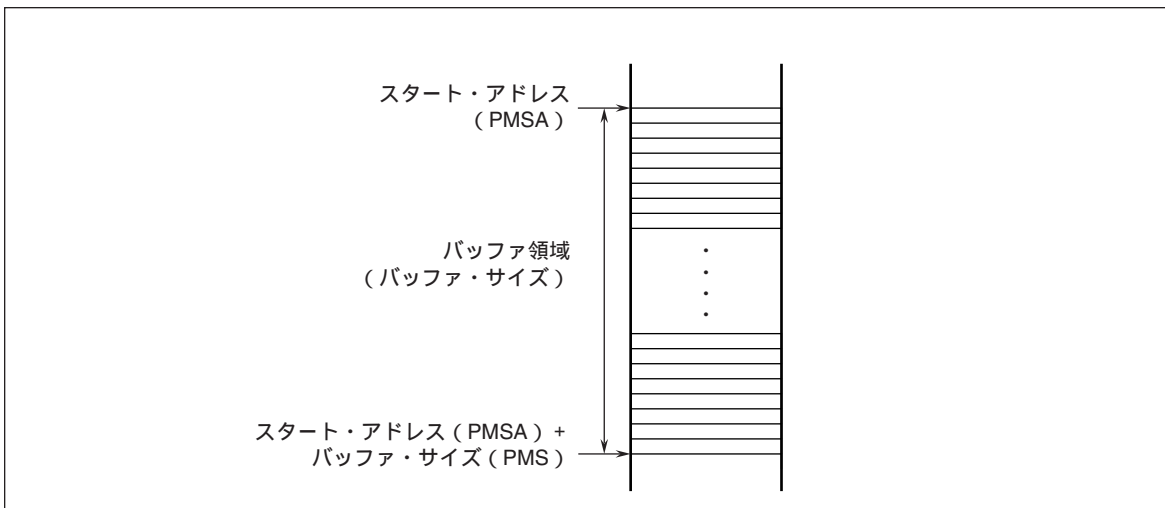
動作を開始すると、入力の場合はペリフェラルから送られてきたデータを先頭番地から順に格納し、出力の場合は先頭番地から順にデータを読み出してペリフェラルに転送します。バッファの最後までアクセスしたら、先頭アドレスに戻ります。

割り込み発生タイミングを、バッファ・サイズの2分の1、または4分の1に設定できます。バッファの2分の1、または4分の1のデータが転送されたら割り込みを発生します。

バッファ・サイズの2分の1で割り込み発生するモードの場合は、バッファ・サイズ指定の下位1ビットを0にしてください。バッファ・サイズの4分の1で割り込み発生するモードでは、バッファ・サイズ指定の下位2ビットを0にしてください。

割り込みが発生した時点で転送を中断するモードと、続けて転送を行うモードがあります。シリアル・インタフェースのように、同期信号の場合は続けて転送しても、バッファが埋まるタイミングが決まっているので、割り込み発生間隔を決めることができます。ホスト・インタフェースのように非同期のタイミングでデータが転送されてくる場合は、割り込み発生間隔が動作時までわからないので、割り込みのたびに転送を中断し、プログラムがデータを受け取ったあとに動作を再開させます。

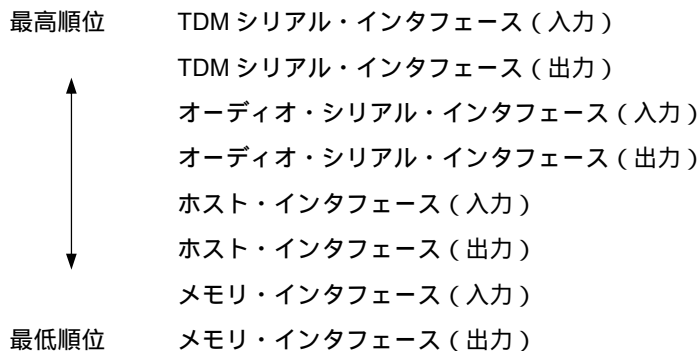
図 5-26 PMT の設定



5.8.3 PMT 転送手順

- (1) PMSA レジスタに、アクセスするデータ RAM 領域の先頭アドレスをセットします。PMT 動作中に値を変更できますが、設定された値が有効になるのは、次回起動時からです。
- (2) PMS レジスタにバッファ・サイズを設定します。PMT 動作中に値を変更できますが、設定された値が有効になるのは次回起動時からです。
- (3) PMC レジスタの EN を 1 に設定することにより PMSA ,PMS が PMP ,PMWC にそれぞれセットされレディ状態になります。
- (4) ペリフェラル回路からの転送要求により 1 ワードの転送を実行します。複数チャンネルから転送要求が発生した場合は、調停を行います。

転送要求の優先順位は、次のとおりです。ただし、直前に受理されたチャンネル要求は最低順位として扱われず(ラウンド・ロビン方式)。



PMT の転送対象となっているペリフェラルに対して、直接アクセスすることを禁止します。たとえば、PMT でホスト・インタフェースの転送を行っているときにホスト・レジスタをアクセスすると、PMT アクセスと直接アクセスとでどちらが優先されるか、保証できなくなります。

PMT 転送停止指示は、EN を 0 とすると転送残り語数にかかわらず停止します。基本的には ST を 1 としてバッファリング完了時点で停止することを推奨します。

PMT の転送は、内部 RAM を 0x0000-0x0FFF (4K ワード)、0x1000-0x1FFF (4K ワード)、0x2000-0x2FFF (4K ワード)、0x3000-0x37FF (2K ワード) の 4 つのメモリ・バンクで行います。各メモリ・バンクにて DSP コア・カーネルによるアクセス競合が発生すると DSP コア・カーネル動作にウエイトが入りアービトレーションを行います。

したがって、ユーザはプログラム中で DMA 転送中のバンクを避けてプログラミングした方が、ウエイトのないデータ転送を実行できます。X, Y は区別するので、たとえば X メモリの 0x0000-0x0FFF を DMA 転送中に、DSP コア・カーネルによる Y メモリの 0x0000-0x0FFF へのアクセスでは競合しません。

5.9 汎用入出力ポート (PIO)

汎用入出力ポート (PIO) は 4 ビットの汎用入出力ポートです。μ PD77210 ファミリではこれを 4 セット (P0-P3, P4-P7, P8-P11, P12-P15) 内蔵していて最大 16 ビットのポートとして利用可能です。ここでは P0-P3 を例にあげて説明しますが、ほかの汎用入出力ポートも同じ構成になっています。

汎用入出力ポートのブロック図を図 5-27 に、レジスタ群を表 5-23 に示します。

図 5-27 汎用入出力ポートのブロック図

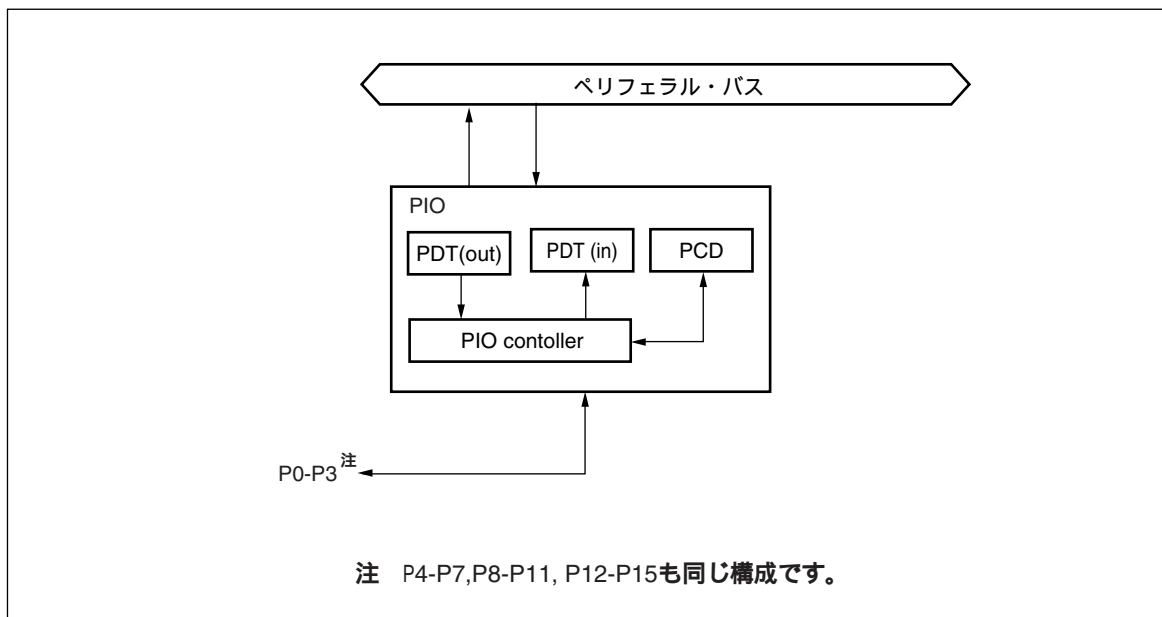


表 5-23 汎用入出力ポートのレジスタ

X/Y メモリ・アドレス	レジスタ名	機 能	ロード / ストア
0x3870	PDT0	ポート・データ・レジスタ 0	L/S
0x3871	PCD0	ポート・コマンド・レジスタ 0	L/S
0x3872	PDT1	ポート・データ・レジスタ 1	L/S
0x3873	PCD1	ポート・コマンド・レジスタ 1	L/S
0x3874	PDT2	ポート・データ・レジスタ 2	L/S
0x3875	PCD2	ポート・コマンド・レジスタ 2	L/S
0x3876	PDT3	ポート・データ・レジスタ 3	L/S
0x3877	PCD3	ポート・コマンド・レジスタ 3	L/S

5.9.1 汎用入出力ポートの端子

(1) P0-P15 (汎用入出力ポート - 入出力)

汎用入出力端子です。

P8-P15 はホスト・インタフェースの HD8-HD15 端子と兼用となっているため、ホスト・インタフェースが 16 ビット・バス・モードのときは汎用入出力ポート端子としては利用できません。

5.9.2 汎用入出力ポートのレジスタ

(1) PDT0-PDT3 (ポート・データ・レジスタ : Port Data Transfer Register)

汎用入出力ポートからデータを入出力するための 16 ビット・レジスタです (下位 4 ビットのみ有効)。汎用入出力ポート端子 4 本ごとに 1 つのレジスタが割り当てられ、PDT0-PDT3 の 4 つのレジスタがあります。P0-P3 端子に対応するのが PDT0、同様に P4-P7、P8-P11、P12-P15 端子に対応するのがそれぞれ PDT1、PDT2、PDT3 となります。それぞれ出力用と入力用のレジスタを別々に持っています。

PDT (out) は、出力するデータを設定する 16 ビットのレジスタです。PDT に対するストア命令を実行すると PDT (out) にペリフェラル・バスからデータを入力します。汎用入出力ポート端子が出力に設定されている場合、下位 4 ビットの値が出力端子の出力値となり、ビットの値が 1 のときはハイ・レベル、0 のときはロウ・レベルを出力します。入力に設定されている端子については対応するビットの値は無効になります。リセット時の値は、0x0 です。

PDT (in) は、入力されたデータを読み出す 16 ビットのレジスタです。PDT に対してロード命令を実行すると、ペリフェラル・バスにデータを出力します。汎用入出力ポート端子が入力に設定されている場合、下位 4 ビットの値が入力端子の入力値となり、ハイ・レベル入力のときはビットの値が 1、ロウ・レベル入力のときは 0 になります。出力に設定されている端子については対応するビットの値は不定です。

(2) PCD0-PCD3 (ポート・コマンド・レジスタ : Port Command Register)

汎用入出力ポートの、入出力の方向および出力端子のビット操作を指定する 16 ビットのレジスタです。PCD の各ビットの機能については表 5-24 に示します。リセット時の値は、0x0 です。

表 5-24 PCD のビット構成 (1/2)

ビット	名称	種類	ビットの機能	ロード/ ストア (L/S)
15	BE	ビット操作	ビット操作許可ビット ・ 0 : ビット操作しない ・ 1 : ビット操作する 操作内容は、B1、B0、PSR で指定する ・ 読み出し時は不定	S
14	PSR	ビット操作	ポート・セット/リセット指定ビット ・ 0 : リセット (ロウ・レベル) ・ 1 : セット (ハイ・レベル) ・ 操作ポートは、B1、B0 で指定する ・ BE=1 のとき、有効 ・ 読み出し時は不定	S
13	ME	モード設定	モード設定許可ビット ・ 0 : モード設定はしない ・ 1 : モード設定する 設定内容は、IO、M3-M0 で指定する ・ 読み出し時は不定	S

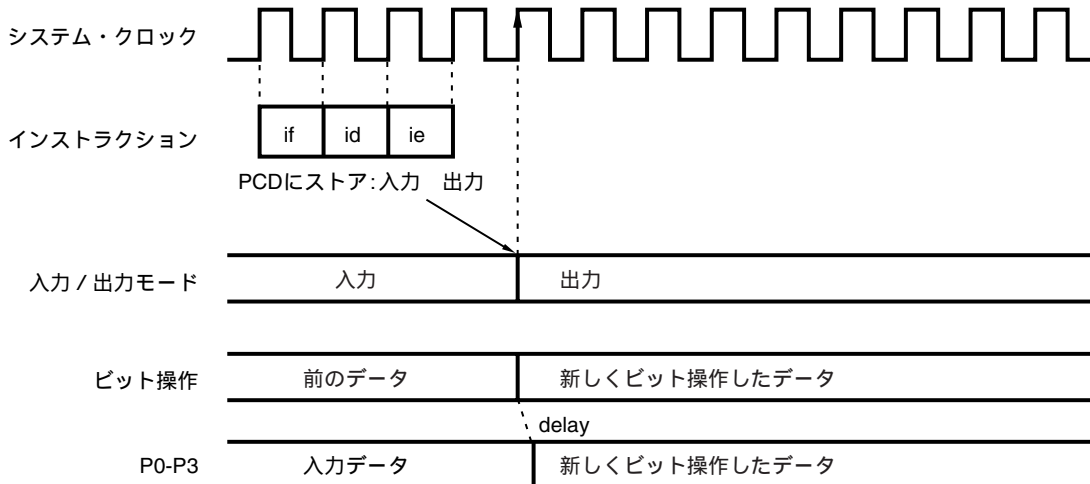
表 5-24 PCD のビット構成 (2/2)

ビット	名 称	種 類	ビットの機能	ロード/ ストア (L/S)
12	IO	モード設定	入出力指定ビット ・ 0 : 入力ポートに指定 ・ 1 : 出力ポートに指定 ・ 設定ポートは, M3-M0 で指定する ・ ME = 1 のとき, 有効 ・ 読み出し時は不定	S
11, 10	予約	-	予約ビット ・ これらのビットには値を設定できない ・ 読み出し時は不定	-
9, 8	B1 B0	ビット操作	ビット操作ポート指定ビット ・ B1, B0 = 00 : P0 01 : P1 10 : P2 11 : P3 ・ セット/リセットは, PSR で指定する ・ BE = 1 のとき, 有効 ・ 読み出し時は不定	S
7-4	予約	-	予約ビット ・ これらのビットには値を設定できない ・ 読み出し時は不定	-
3 2 1 0	M3 M2 M1 M0	モード設定	モード設定ポート指定ビット ・ M3 = 0 : P3 非選択, 1 : P3 選択 ・ M2 = 0 : P2 非選択, 1 : P2 選択 ・ M1 = 0 : P1 非選択, 1 : P1 選択 ・ M0 = 0 : P0 非選択, 1 : P0 選択 独立して選択することが可能	S
		モード・ ステータス	入出力モード・ステータス・ビット ・ M3 = 0 : P3 入力, 1 : P3 出力 ・ M2 = 0 : P2 入力, 1 : P2 出力 ・ M1 = 0 : P1 入力, 1 : P1 出力 ・ M0 = 0 : P0 入力, 1 : P0 出力	L

5.9.3 汎用入出力ポートのタイミング

汎用入出力ポートは、本来は定期的な使用を想定したインタフェースではありませんが、入出力についてはCLKOUTの立ち上がりエッジに対して同期関係にあります。

(1) 入力からの出力へのモード変更



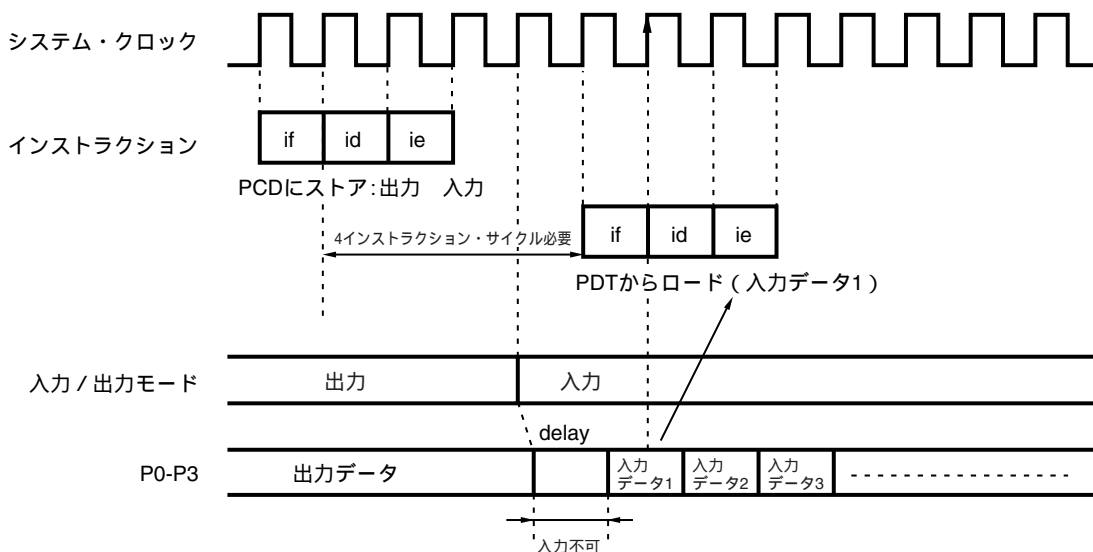
各端子のモードは、PCDレジスタにストアする命令の実行サイクルから、2システム・クロック後に入力から出力に変更されます。

プログラム例：

```
#define PCD = 0x3805
#define PDT = 0x3804
R1L = 0x0000 ;
* PDT : x = R1L ; PDT を初期化
R0L = 0x3001 ;
* PCD : x = R0L ; P0 を出力に設定
```

注意 PDTレジスタは、ハードウェア・リセット後は不定なので、PCDレジスタにストアする前に、PDTレジスタにデータをあらかじめ設定しておいてください。

(2) 出力から入力へのモード変更



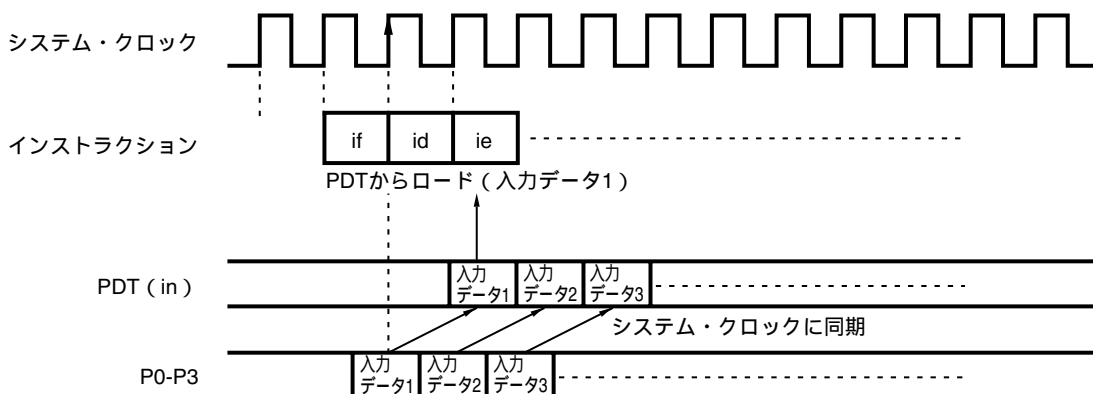
各端子のモードは、PCD レジスタにストアする命令の実行サイクルから、2 システム・クロック後に、出力から入力に変更されます。しかし μ PD77210 ファミリは、入力に変更されてから、2 システム・クロックの間は入力を受け付けません。よって、PCD レジスタにストアする命令から PDT レジスタよりロードする命令の間は、最低 4 システム・クロックの時間を空けてください。

プログラム例:

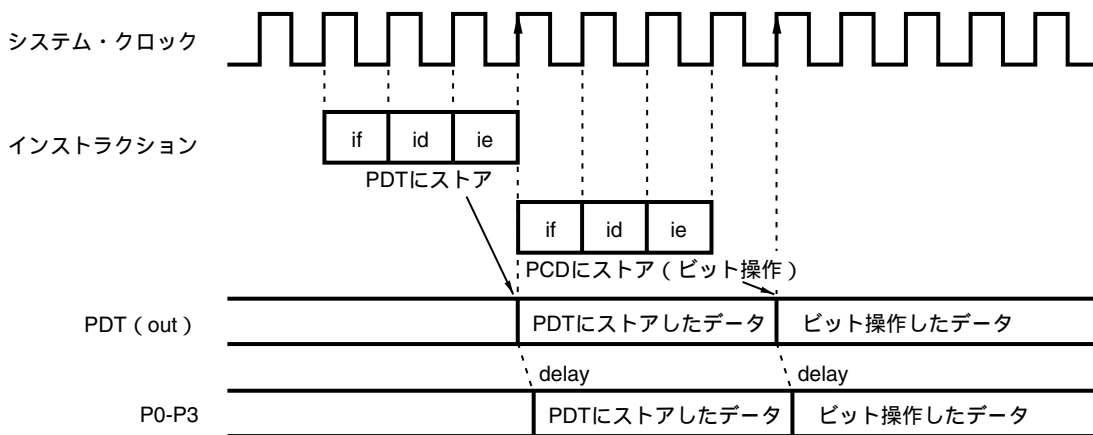
```
#define PCD = 0x3805
#define PDT = 0x3804
R0L = 0x200f      ;
*PCD : x = R0L    ; P0-P3 出力 入力
    この間は最低 4 システム・クロック空けてください
R1L = *PDT : x    ; PDT からロード
```

(3) 入力ポートのタイミング

端子から入力されたデータは、システム・クロックの 2 回の立ち上がりエッジで同期したあとでロードされます。



(4) 出力ポートのタイミング



(a) PDT レジスタにストアする命令の場合

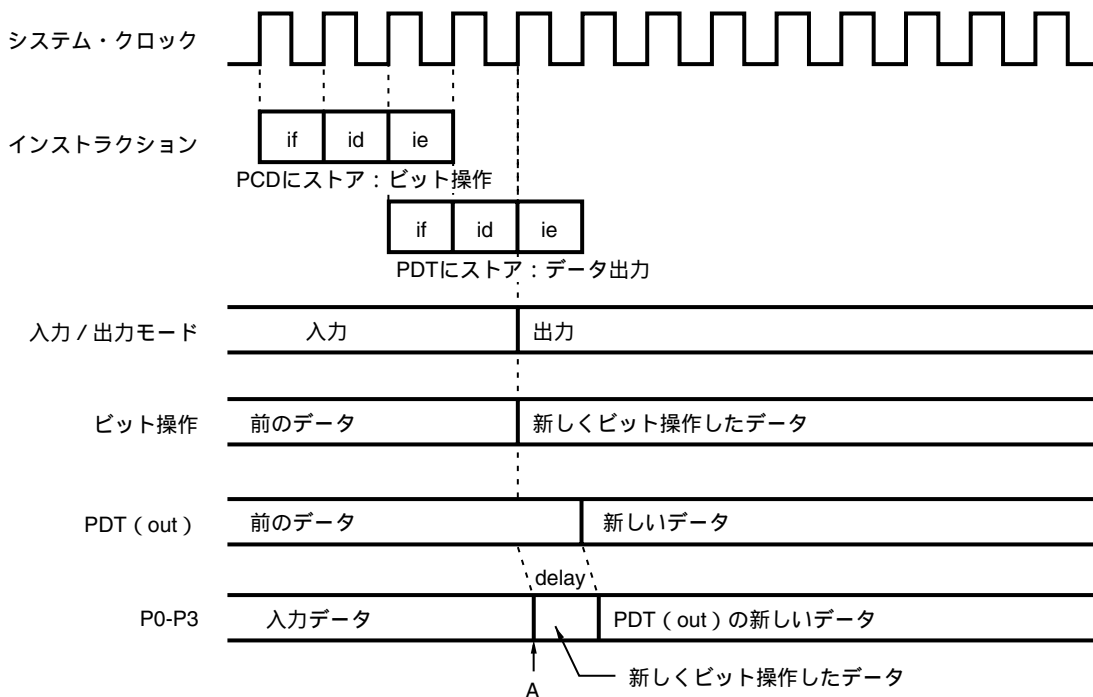
出力されるデータは、PDT レジスタにストアする命令の実行サイクルから、1 システム・クロック後に出力されます。

(b) PCD レジスタにストアする命令の場合

ビット操作をしたデータは、PCD レジスタにストアする命令の実行サイクルから、2 システム・クロック後に出力されます。

注意 PCD レジスタによるビット操作と PDT レジスタのストアによる出力データが同じタイミングになった場合、PDT レジスタのデータが PCD レジスタによるビット操作より優先されます。

(5) 出力ポートの設定 (PCD レジスタと PDT レジスタを使用する場合)



ビット操作をしたデータは、PCD レジスタにストアする命令の実行サイクルから、2 システム・クロック後に出力されます。次に、PDT レジスタにストアする命令の実行サイクルから、1 クロック後にデータが出力されるとき、A ポイントでスパイクが発生する可能性があります。よって、PCD レジスタにストアする命令として PDT レジスタにストアする命令の間に、最低 1 システム・クロック空ける必要があります。

プログラム例：

```
#define PCD = 0x3805
#define PDT = 0x3804
R0L = 0xf00f      ;
*PCD : x = R0L    ; P0-P3 入力 出力, P0 にハイ・レベルを出力
R1L = 0x0000      ; この間は最低 1 システム・クロック空けてください
*PDT : x = R1L    ; P0-P3 にロウ・レベルを出力
```

- 注意 1. PCD レジスタにストアする命令と PDT レジスタにストアする命令の間に、最低 1 システム・クロック空けない場合、A ポイントでスパイクが発生する可能性があります。
2. PDT レジスタは、ハードウェア・リセット後は不定なので、PCD レジスタにストアする前に、PDT レジスタをあらかじめ設定しておいてください。

5.9.4 ポート・プログラミングの例

汎用入出力を使用したプログラミング例を次に示します。この例では次のように実行されます。

P0 と P1 は出力端子に設定される。

P2 と P3 は入力端子に設定される。

P0 にはロウ・レベル、P1 にはハイ・レベルが出力される。

汎用入出力ポートのプログラミング例

```
#define PDT 0x3804
#define PCD 0x3805
#define All_In_mode 0x200F
#define P0_Out_mode 0x3001
#define P1_Out_mode 0x3002
#define Out_P0_Low 0x8000
#define Out_P1_High 0xC100

R0L = All_In_mode      ; P3-P0 入力端子
*PCD : x = R0L        ;
R0L = P0_Out_mode + Out_P0_Low ; P0 出力端子 (ロウ・レベル)
*PCD : x = R0L        ;
R0L = P1_Out_mode + Out_P1_High ; P1 出力端子 (ハイ・レベル)
*PCD : x = R0L        ;
```

5.10 割り込みコントローラ (INTC)

DSP コア・カーネルには 12 本の割り込みポートが用意されています。

割り込みコントローラ (INTC) は、この 12 本の割り込みポートを拡張して、各ポートに 4 要因を割り当て、最大 48 要因の割り込みを取り扱います。

割り込みコントローラのレジスタを表 5-25 に示します。

表 5-25 割り込みコントローラのレジスタ

X/Y メモリ・アドレス	レジスタ名	機 能	ロード/ストア
0x3880	ICR0	割り込み制御レジスタ 0 (ベクタ・アドレス: 0x210 用)	L/S
0x3881	ICR1	割り込み制御レジスタ 1 (ベクタ・アドレス: 0x214 用)	L/S
0x3882	ICR2	割り込み制御レジスタ 2 (ベクタ・アドレス: 0x218 用)	L/S
0x3883	ICR3	割り込み制御レジスタ 3 (ベクタ・アドレス: 0x21C 用)	L/S
0x3884	ICR4	割り込み制御レジスタ 4 (ベクタ・アドレス: 0x220 用)	L/S
0x3885	ICR5	割り込み制御レジスタ 5 (ベクタ・アドレス: 0x224 用)	L/S
0x3886	ICR6	割り込み制御レジスタ 6 (ベクタ・アドレス: 0x228 用)	L/S
0x3887	ICR7	割り込み制御レジスタ 7 (ベクタ・アドレス: 0x22C 用)	L/S
0x3888	ICR8	割り込み制御レジスタ 8 (ベクタ・アドレス: 0x230 用)	L/S
0x3889	ICR9	割り込み制御レジスタ 9 (ベクタ・アドレス: 0x234 用)	L/S
0x388A	ICR10	割り込み制御レジスタ 10 (ベクタ・アドレス: 0x238 用)	L/S
0x388B	ICR11	割り込み制御レジスタ 11 (ベクタ・アドレス: 0x23C 用)	L/S

5. 10. 1 割り込みコントローラのレジスタ

(1) ICR0-ICR11 (割り込み制御レジスタ: Interrupt Control Register)

割り込みモード設定や割り込みマスクを設定する 16 ビット・レジスタです。DSP コア・カーネルの割り込みポート (割り込みベクタ・アドレス) ごとに存在します。

ICR の各ビットの機能については表 5-26に示します。デフォルトは 0x0 です。

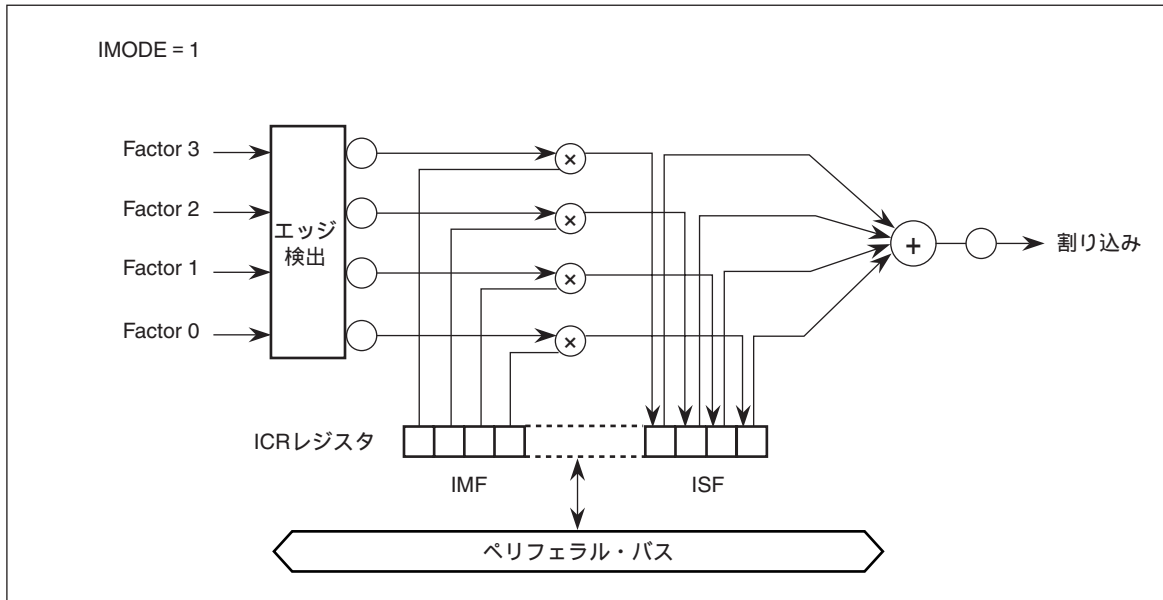
表 5-26 ICR のビット構成

ビット	名称	機能	ロード/ストア
15-9	予約	<ul style="list-style-type: none"> ・0 以外の書き込みを行わないこと。 ・読み込み時は不定。 	-
8	IMODE	割り込みモード設定ビット <ul style="list-style-type: none"> ・0: マスク・モード (デフォルト) ・1: スタンプ・モード 	L/S
7-4	IMF	割り込みマスク・フラグ <ul style="list-style-type: none"> ・0: マスクする (デフォルト) ・1: マスクしない 各割り込みポートに入力された割り込みを DSP コア・カーネルに送るか送らないかを設定する。 ビット 7 が Factor3, ビット 6 が Factor2, ビット 5 が Factor1, ビット 4 が Factor0 に対応する。	L/S
3-0	ISF	割り込みスタンプ・フラグ <ul style="list-style-type: none"> ・0: 割り込みなし (デフォルト) ・1: 割り込みあり 割り込みが入ると対応するビットが 1 になる。ICR を読み出すと 0 になる。 ビット 3 が Factor3, ビット 2 が Factor2, ビット 1 が Factor1, ビット 0 が Factor0 に対応する。	L

5.10.2 割り込みコントローラの動作モード

スタンプ・モード (IMODE = '1') では、DSP コア・カーネルの割り込みポートを要因拡張する動作モードとなります (図 5-28 参照)。

図 5-28 スタンプ・モード時の割り込みコントローラ



4 ビットの割り込みマスク・フラグ (IMF) と 4 ビットの割り込みスタンプ・フラグ (ISF) を持つ ICR レジスタ (割り込みコントロール・レジスタ) を DSP コア・カーネルの割り込みポートごとに 1 セット、合計 12 セット持っています。これによって DSP コア・カーネルの割り込みポート 1 本に対して 4 つの要因拡張を行います。

4 つの要因を各 4 本の割り込み信号に対して立ち下がりを検出し、マスクされていない場合は該当するスタンプ信号をセットし割り込みを記録します。DSP コア・カーネルに対する割り込み信号は、4 本すべてのスタンプ・フラグの論理和で出力し、いずれかの割り込み入力で割り込みベクタへ分岐処理します。割り込みベクタで ICR レジスタを参照すれば、どの割り込み要因かを識別できます。

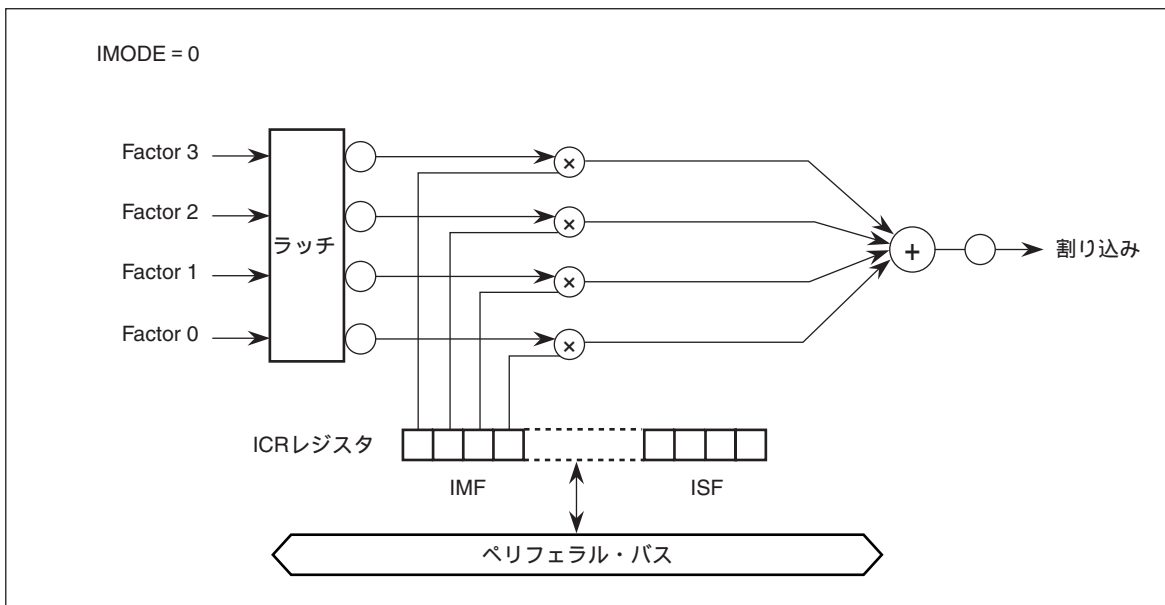
スタンプ・フラグは ICR のロードによりリセットされ、ICR ロード後 4 サイクル間 DSP コア・カーネルに対する割り込み信号を強制マスクします。ICR のロードとスタンプ・フラグのセットが同時もしくはセットが遅れるケースでは、フラグはセット優先であり、DSP コア・カーネルがロードしたあと、再度割り込みを発生します。

割り込みベクタ内で ICR のロード処理をしないかぎり、DSP コア・カーネルへの割り込み信号はロウ・レベルを維持するため、以降の割り込み信号がマスクされてしまいます。したがって、ICR のロードが必要です。

また、ICR のロード前に同一要因で多重に割り込み発生する場合は、1 つの割り込みとみなされます。

マスク・モード (IMODE = '0') では、単純に 4 本の要因をマスクする動作モードとなります (図 5-29 参照)。

図 5-29 マスク・モード時の割り込みコントローラ



ICR によって設定されたマスクされない割り込み要因に対しては、単純に論理和を取り、DSP コア・カーネルに割り込み信号として送じます。要因の選択を 1 つにすることにより μ PD77111 ファミリーと互換性を持った取り扱いが可能です。このモードでは、ISF の値は不定になります。

5. 10. 3 割り込みテーブル

リセットと 12 本の DSP コア・カーネル割り込み用の割り込みベクタ先頭アドレスです。

リセットや割り込みの入力により、その要因に対応するベクタのアドレスにジャンプし、割り込みルーチンを実行します。

リセットは 16 ワード、各割り込み要因は 4 ワードのベクタ領域を持っています。それ以上の長さの処理を行う場合は、割り込み処理が書かれたルーチンの先頭にジャンプする命令を、ここに記述する必要があります。

外部割り込みは DSP コア・カーネルの割り込みポート 0-3 が割り当てられているので、これを要因拡張して 16 ビット化します。外部割り込み端子は 4 ビット PIO の入力信号にアサインし、4 セット 16 本の割り込みを取り扱うことができます。

ペリフェラル、TSIO、ASIO、HIO は入力/出力に対してそれぞれ 4-9 の割り込みポートが割り当てられています。要因 1 は端子にアサインし、使用しないペリフェラルは外部割り込み端子として使用可能です。

表 5-27 割り込みテーブル

割り込み番号	ベクタ・アドレス	割り込み要因			
		0	1	2	3
RESET	0x200	リセット	予約	予約	予約
0	0x210	$\overline{\text{INT00}}$	$\overline{\text{INT01}}$	$\overline{\text{INT02}}$	$\overline{\text{INT03}}$
1	0x214	$\overline{\text{INT10}}$	$\overline{\text{INT11}}$	$\overline{\text{INT12}}$	$\overline{\text{INT13}}$
2	0x218	$\overline{\text{INT20}}$	$\overline{\text{INT21}}$	$\overline{\text{INT22}}$	$\overline{\text{INT23}}$
3	0x21C	$\overline{\text{INT30}}$	$\overline{\text{INT31}}$	$\overline{\text{INT32}}$	$\overline{\text{INT33}}$
4	0x220	TSI 入力	TSIEN	PMT ch0	SDCR 入力 ^注
5	0x224	TSO 出力	TSOEN	PMT ch1	SDCR 出力 ^注 (ビジィ解除)
6	0x228	ASI 入力	ASIEN	PMT ch2	SDDAT 入力 ^注
7	0x22C	ASO 出力	ASOEN	PMT ch3	SDDAT 出力 ^注
8	0x230	HI 入力	$\overline{\text{HWR}}$	PMT ch4	予約
9	0x234	HO 出力	$\overline{\text{HRD}}$	PMT ch5	予約
10	0x238	TIMER ch0	TIMER ch1	PMT ch6	予約
11	0x23C	TIMER ch1	TIMER ch0	PMT ch7	予約

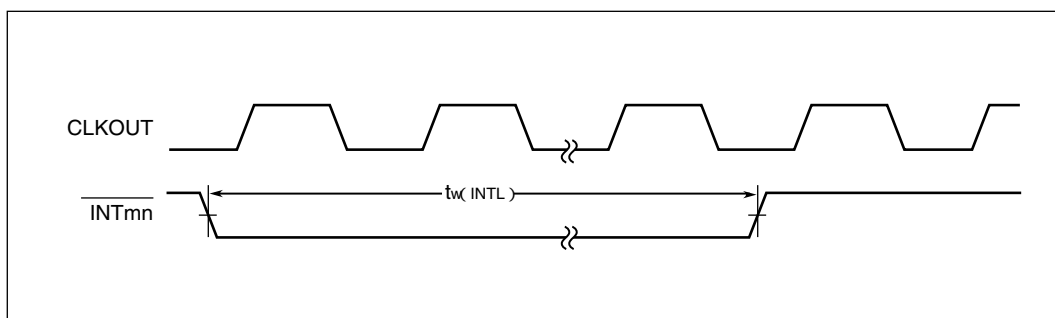
注 μ PD77213 の場合のみ。 μ PD77210 はすべて「予約」です。

5. 10. 4 外部割り込み端子による割り込みのハードウェア条件

外部割り込み ($\overline{\text{INTmn}}$) は、各信号端子の立ち下がりエッジを検出して受け付けられます。したがって次々に割り込みをかけるためには、そのたびにいったんハイ・レベルに戻し、その後ロウ・レベルにすることで立ち下がりエッジを与えてください。ただし、ロウ・レベル、ハイ・レベルとも、検出するための時間が必要です。

図 5-30 に外部割り込みタイミングをそれぞれ示します。

図 5-30 外部割り込みタイミング



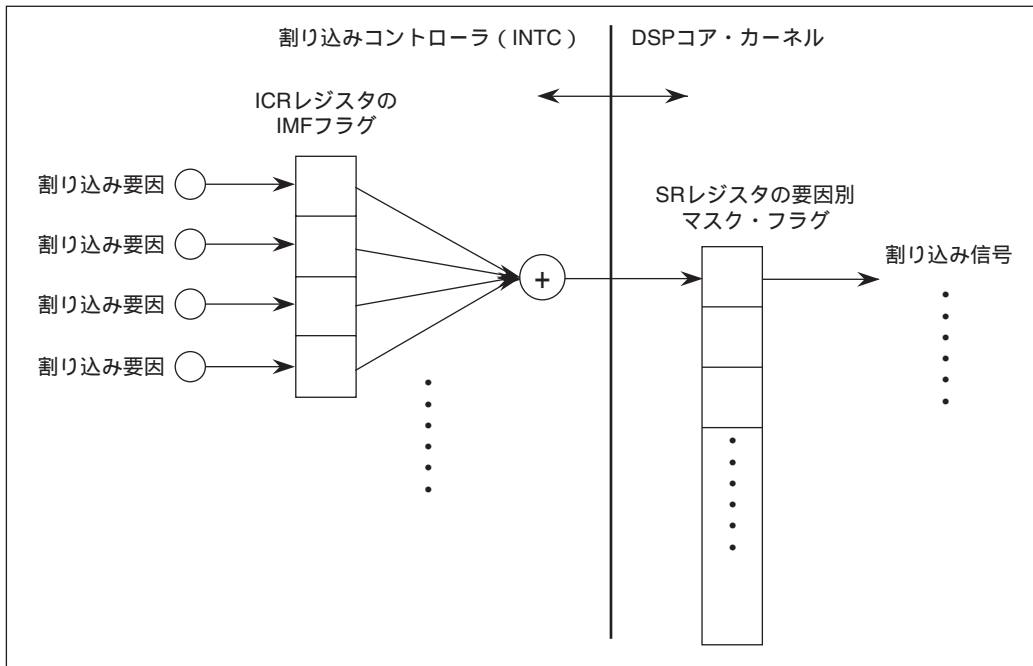
注意 外部端子による割り込み ($\overline{\text{INTmn}}$) のアクティブ時間は、システム・クロックにより定義されています。ホールド・モード時など、分周クロックで動作している場合は、分周クロックがシステム・クロックになります。

5.10.5 割り込みに関する注意点

(1) SR レジスタと ICR レジスタ

割り込みのマスク・フラグには2種類あります。図 5-31に2種類のマスク・フラグの関係について示します。

図 5-31 2種類の割り込みマスク・フラグの関係



SR レジスタは DSP コア・カーネルの内部にあり、割り込み要因別のマスク制御を行うレジスタです。DSP コア・カーネルの外（ペリフェラルである割り込みコントローラ）より入力された割り込みは、マスクされている、いないに関わらず 1 回分の割り込み要求として記憶されます。この場合、マスク解除によって記憶されていた割り込みが実行されます。

ICR レジスタはペリフェラルである割り込みコントローラの内部にあり、その中の IMF フラグは割り込みコントローラにより拡張された割り込み要因のマスク制御を行うフラグです。マスクされた割り込み要因は、DSP コア・カーネルへは入力されません。したがって、IMF フラグでマスクした割り込みについては記憶されません。

(2) fint 命令

全体あるいは個別の割り込みを不許可にしているときに割り込みが入った場合、割り込み不許可になっているために実際には割り込み処理に分岐することはありませんが、DSP コア・カーネルは割り込み要因ごとに1回分の割り込みを記憶しています。この記憶され保留している割り込みは、割り込み許可になった時点で有効になり、割り込み処理を実行します。

fint 命令は、この DSP コア・カーネルに保留している割り込み要求信号をクリアする命令です。保留されている不要な割り込み要求を破棄するときに使用します。

ただし、この命令を使用するには次の点に注意してください。

- ペリフェラルの割り込み信号は、ペリフェラルに対するアクセス許可信号をもとに作られます。
- これらのアクセス許可信号は fint 命令ではクリアされません。

したがって、割り込み保留状態で fint 命令を実行した場合、割り込み要求を出したペリフェラル側は出した要求に対してする処理を受けられないので、両方とも次のような待機状態になります。

- DSP コア・カーネル：

割り込み要求を待っている（割り込み要求をクリアしたため今行うべき割り込み処理はなく、次の割り込み信号を待っている状態）

- ペリフェラル：

割り込みに対する処理を待っている（今出している割り込み要求の処理を行ってもらえないので次の割り込み要求を出せない状態）

この場合、ペリフェラルのアクセス許可フラグをクリアすることによってこの状態を回避することができます。ペリフェラルに対してダミーのアクセスを行うことで、アクセス許可フラグをクリアします。オーディオ・シリアル・インタフェースの入力割り込みを用いる場合のプログラム例を図 5-32に示します。ほかのペリフェラルの場合も同様です。割り込みに使用している要因と同一の機能の許可フラグをクリアするように、ダミー出力やダミー入力を行ってください。ペリフェラル・ステータス・レジスタに許可フラグが存在しないペリフェラルに関しては問題はありません。

図 5-32 fint 命令使用例：オーディオ・シリアル入力割り込み使用，32 ビット・モード時

```

;割り込み禁止
:
fint                               ;
clr (r0)                            ;
r0l = *ASST:x                       ;
r0 = r0 & 0x0001                    ; check "ASDT Load Enable Flag"
if (r0 == 0) jmp $+3                ;
r0l = *ASDT:x                       ; dummy access to clear "ASDT Load Enable Flag"
r0l = *ASDT:x                       ; dummy access to clear "ASDT Load Enable Flag"
any instruction                      ;
:
;割り込み許可

```

なお、fint 命令は不要な割り込みを破棄することを目的とするので、割り込み禁止状態で行うことが適当です。

(3) 割り込み許可と禁止の設定順序

割り込み要因ごとの許可 / 不許可は SR レジスタと ICR レジスタとで設定できます (5. 10. 5 (1) SR レジスタと ICR レジスタを参照)。最終的に割り込みを許可にするためには両方を許可にする必要がありますが、ここでは許可にする順序について説明します。

SR レジスタでは割り込み要因をマスク (不許可) した場合にも、割り込みが入力されたときにその割り込みを記憶しています。したがって、不要な割り込みが記憶されている場合にも、割り込みを許可にした段階で割り込み処理を行います。これに対して ICR での割り込み要因マスクは割り込みのおおもとからの信号がマスクされるため DSP コア・カーネルには割り込みそのものが入力されないことになります。

したがって不要な割り込みを破棄する必要がある場合は、次の順序で行ってください。

- SR ICR の順で許可
- ICR SR の順で不許可

もちろん、単に割り込みを一時的に不許可にするような場合には SR, ICR とも不許可にする必要はありません。SR の設定だけ行ってください。

(4) リポート後の割り込み

リポートを行う場合、リポート方法によってはリポート後に割り込みが発生することがあるので注意してください。なお、リポートに関しては割り込み禁止状態で行う必要があり、割り込み許可状態でリポートを行った場合の動作は保証できません。

- ホスト・リポート
- シリアル・リポート

これらの方法でリポートを行った場合、リポート終了後に割り込み許可状態に変更するとリポート方法に応じてホストあるいはシリアル入力割り込みが発生します。

これは、割り込みの使用 / 未使用に関わらずホストあるいはシリアルの入力アクセスにおいて割り込み信号が発生し、1 回分の割り込みについては保留されているためです。

したがって、ホストあるいはシリアル入力割り込みを使用する場合は、リポート終了後に、fint 命令を実行するか、ダミー入力を行う必要があります (5. 10. 5 (2) fint 命令を参照)

なお、ブート時は ICR レジスタがマスク状態で行われるので問題はありません。また、リポート時でも ICR によって該当の割り込みをマスクしている場合は問題ありません。

(5) リセット時のマスク・モードとスタンプ・モードとの動作の違い

リセット解除から割り込みコントローラの割り込みマスク・フラグ解除までの間に発生する割り込みについて、マスク・モードとスタンプ・モードでは動作が違うので注意してください。

DSP コア・カーネルは割り込み要因の立ち下がりエッジを検出して割り込みと認識します。したがって、割り込みをかけるには立ち下がりエッジが必要です。ICR レジスタのマスク・フラグがマスクされているときは、割り込みコントローラはハイ・レベルを出力しています。このため、リセット解除からマスク・フラグ解除までに発生する割り込みについては、マスク・モードとスタンプ・モードでは次のような動作の違いが生じます。

• マスク・モード

マスク・モードではマスク・フラグ解除時に割り込みコントローラに入力される割り込み信号は、そのまま DSP コア・カーネルの割り込みへと入力されます。マスク・モードで割り込みマスク・フラグを解除した場合、内部ペリフェラルによる割り込み^注はロウ・レベルを維持しているために、マスク・フラグ解除の時点で割り込みコントローラの出力がロウ・レベルになります。このため、この内部ペリフェラルによる割り込み^注は DSP コア・カーネルに対して発生します。したがって、該当のペリフェラルに対して割り込みを用いる場合は、ダミーのアクセスを必要としません。このモードでは、 μ PD77111 ファミリと異なるので注意してください。

• スタンプ・モード

スタンプ・モードではマスク・フラグ解除時に割り込みコントローラに入力される割り込み信号の立ち下がりを検出して割り込みスタンプ・フラグをセットします。

スタンプ・モードで割り込みマスク・フラグを解除した場合、内部ペリフェラルによる割り込み^注はその立ち下がりエッジを検出できないために割り込みスタンプ・フラグにセットできません。このため、この内部ペリフェラルによる割り込み^注は DSP コア・カーネルに対して発生しません。したがって、該当のペリフェラルに対して割り込みを用いる場合は最初の 1 回についてダミーのアクセスを必要とします。このモードでは、 μ PD77111 ファミリと同様です。

注 ここでいう「内部ペリフェラルによる割り込み」とは、シリアル・インタフェースの出力割り込み、ホスト・インタフェースの出力割り込みを示します。あるいは、ロウ・レベルに固定した割り込み端子（たとえばブート・モード設定に用いるポート端子）についても同様です。

5.11 タイマ (TIM)

16 ビット分解能のタイマを 2 チャンネル搭載しています。インターバル・タイマ、イベント・カウンタ、フリー・ランニング・タイマ、ウォッチドッグ・タイマとして利用できます。

ソース・クロックとして 5 種類から選択可能であり、さらに 8 種類のプリスケアラ機能を持っています。

2 チャンネルのタイマを結合することにより 32 ビット・タイマを実現可能です。

主な特徴を次に示します。

• タイマ・ソース

システム・クロック、外部割り込み (2 系統)、シリアル・クロック、タイム・アップ信号 (他方のタイマ出力) の 5 種類から選択可能

• ソース・プリスケアラ

1, 2, 4, 8, 16, 32, 64, 128 分周を選択可能

• リピート・カウント

単発動作とカウント・アップ後に初期値をロードして、再カウント起動するリピート動作が可能

• カウント・モディファイ機能

カウント動作中にカウント・データを書き換え可能 (本機能によりタイム・アップ前に初期値を設定することでウォッチドッグ・タイマを実現できます。)

• 割り込み発生機能

タイム・アップ信号を割り込み信号として発生

タイマのブロック図を図 5-33に、レジスタ群を表 5-28に示します。

図 5-33 タイマのブロック図 (1チャンネル分)

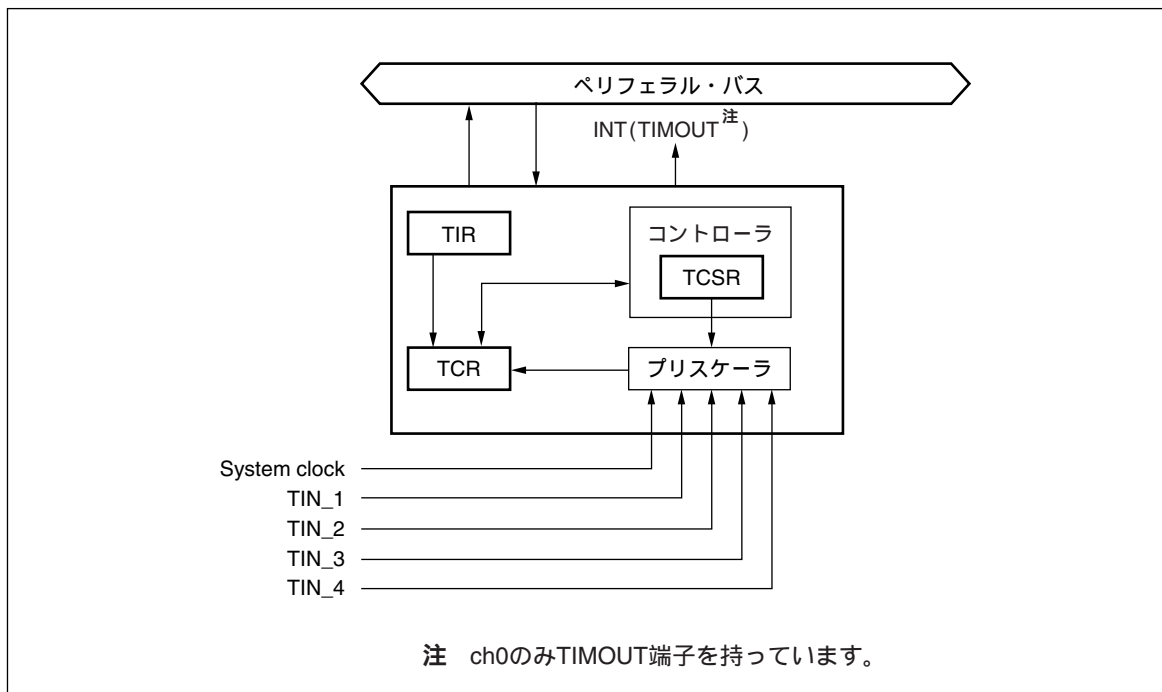


表 5-28 タイマのレジスタ

X/Y メモリ・アドレス	レジスタ名	機 能	ロード/ストア
0x3890	TIR0	タイマ初期値設定レジスタ 0	L/S
0x3891	TCR0	タイマ・カウント・レジスタ 0	L/S
0x3892	TCSR0	タイマ制御レジスタ 0	L/S
0x3894	TIR1	タイマ初期値設定レジスタ 1	L/S
0x3895	TCR1	タイマ・カウント・レジスタ 1	L/S
0x3896	TCSR1	タイマ制御レジスタ 1	L/S

5. 11. 1 タイマの端子

(1) TIMOUT (タイム・アップ・アウトプット - 出力)

タイマのタイム・アップを示す端子です。ch0 のみ端子として存在します。

TCR レジスタが 0 になる (タイム・アップする) と、ハイ・レベルを 4 サイクル出力します。

5. 11. 2 タイマのレジスタ

(1) TIR0, TIR1 (タイマ初期値設定レジスタ : Timer Initial Register)

タイマの初期値を設定する 16 ビット・レジスタです。タイマの起動時に TCR レジスタへ転送されます。

TIR はカウントをしても値は変化しません。デフォルトは 0xFFFF です。

(2) TCR0, TCR1 (タイマ・カウント・レジスタ : Timer Count Register)

現在のタイマ・カウント値を保持する 16 ビットのレジスタです。タイマのカウントによりデクリメントされます。デフォルトは 0xFFFF です。ウォッチドッグ・タイマとして使用する場合は、タイム・アップする前に TIR と同じ値をストアしてください。

(3) TCSR0, TCSR1 (タイマ制御レジスタ : Timer Control/Status Register)

タイマの動作の設定、およびステータスを表す 16 ビットのレジスタです。デフォルトは 0x0000 です。TCSR の各ビットの機能については表 5-29 に示します。

表 5-29 TCSR のビット構成

ビット	名称	機能	ロード / ストア
15-8	予約	<ul style="list-style-type: none"> ・ 0 以外の書き込みは行わないこと。 ・ 読み出し時は不定。 	-
7	TEN	タイマ・イネーブル・フラグ <ul style="list-style-type: none"> ・ 0 : ディスエーブル (デフォルト) ・ 1 : イネーブル 1 でカウント開始, 0 でカウント停止。	L/S
6	TFNC	タイマ機能設定ビット <ul style="list-style-type: none"> ・ 0 : リピート (デフォルト) ・ 1 : ワン・タイム タイマ動作のリピート指定。1 回動作で終了か、繰り返すかを指定する。	L/S
5-3	TCLKSEL	タイマ・クロック・ソース・セレクト・ビット <ul style="list-style-type: none"> ・ 000 : システム・クロック (デフォルト) ・ 001 : TIN_1 ・ 010 : TIN_2 ・ 011 : TIN_3 ・ 100 : TIN_4 ・ 101 ~ 111 : リザーブ 	L/S
2-0	TCLKPS	タイマ・クロック・プリスケアラ・セレクト・ビット <ul style="list-style-type: none"> ・ 000 : 1/1 (デフォルト) ・ 001 : 1/2 ・ 010 : 1/4 ・ 011 : 1/8 ・ 100 : 1/16 ・ 101 : 1/32 ・ 110 : 1/64 ・ 111 : 1/128 タイマ・クロック・ソースの分周比を設定する。	L/S

- 注意 1. クロック・ソースの切り替え, プリスケアラの切り替えを, TEN, TFNC の更新と同時に行わないでください。クロック・ソース, プリスケアラの設定は TEN = 0 として行ってください。
2. クロック・ソースの切り替え時にハザード信号により誤カウントするケースが考えられるため, クロック・ソースを確定したあと, タイマを起動させてください。

表 5-30 タイマのクロック・ソース (TCSR のビット 5-3)

ビット 5-3	クロック・ソース	ch0	ch1
000	システム・クロック	システム・クロック	システム・クロック
001	TIN_1	TSIO のシリアル・クロック	ASIO のシリアル・クロック
010	TIN_2	P0 端子	P2 端子
011	TIN_3	P1 端子	P3 端子
100	TIN_4	ch1 の TIMEOUT	ch0 の TIMEOUT

5.11.3 タイマの動作

タイマを起動するには、カウント初期値（周期）を TIR レジスタに、カウント・ソースおよびプリスケアラのパラメータを TCSR に、それぞれストアします。そのあと、タイマ動作をイネーブルするために再度 TCSR にストアします。イネーブル化することにより、TCR へ TIR データが転送されます。TCR はカウント・ソースにより順次デクリメントしていきます。

タイマの終了は、イネーブル信号をモニタするか、割り込み発生により判別します。TCR が 0 となると TIMEOUT が 4 サイクル間ハイ・レベルとなり、またワン・タイム・モードでは、イネーブル信号がインアクティブになります。また、同時に TCR に再度 TIR データが転送されます。リピート・モード時は、イネーブル信号はアクティブのまま保持し、再度 TCR をデクリメントしていきます。

5.11.4 タイマ使用上の注意

システム・クロックの切り替えが行われるとき、調整用のクロックが挿入されます。この調整クロックについては、5.12 クロック・コントローラ (CLKC) を参照してください。

タイマの動作がこの調整クロックをまたぐ場合、タイマのクロック・ソース入力をカウント（サンプリング）できない可能性が生じます。PLL や分周回路の切り替え、またはホールド・モードなどを遷移するケースが該当します。これは、システム・クロックによってタイマのカウントを更新するが、調整クロックの途中においてはタイマの動作をサンプリングできないために起こるものです。タイマ切り替えの前後のクロック周期、逡倍率や分周率、タイマのクロック・ソース周期によってカウントされないタイマ入力数は変わります。

プリスケアラの値が 1/1 のときは、タイマのクロック・ソースの立ち下がりエッジによってカウントします。プリスケアラの値が 1/1 以外のときは、タイマのクロック・ソースの立ち上がりエッジによってカウントします。

5.12 クロック・コントローラ (CLKC)

外部クロック入力 (CLKIN) から入力したクロック信号を制御します。PLL と出力分周器を持っています。PLL では通倍, 出力分周器では分周して, システム・クロックを生成します。

通倍率は, 外部端子 PLL0-PLL3 で設定します。

CLKC レジスタは, PLL のパワー・オン/オフ制御, クロック・ソース選択 (外部クロックと PLL クロック, 分周と非分周出力), 出力用分周回路のリセット制御と分周率設定, CLKOUT イネーブル制御などを行います。本レジスタによりユーザは, 外部クロック動作, PLL 通倍動作, 分周動作, STOP などのクロック動作を制御します。

スタンバイ・モード (HALT / STOP モード) 時は, 自動的にクロック・ソースを分周回路出力に選択します。スタンバイ・モードに入る前に分周器の設定を行う必要があります。あらかじめ分周動作実行されるプログラムでは HALT モードでのクロックは不変です。

STOP モード時では, システム・クロックをマスクします。自動的に PLL をオフにすることはありません。これは PLL のロックアップ時間を不要とし, STOP モードからの早期の立ち上がりなどを実現するためです。この場合, PLL 動作分の消費電流が流れます。また, この間 CLKIN 端子へのクロック供給を停止することはできません。消費電流を低減するためには, ユーザ・プログラムにて PLL を停止する必要があります。この場合, STOP モード中に CLKIN 端子へのクロック供給を停止してもかまいません。

リセットでは, PLL はスタンバイとなり, 外部クロックが直接システム・クロックとして供給される状態になります。このため PLL 通倍あるいは分周器出力をクロック・ソースとして使用している場合, リセットによってハザードが発生します。PLL 通倍あるいは分周器出力をクロック・ソースとした状態でのリセットにおいては, 内部メモリおよび内部レジスタの内容保持について保証できません。

クロック・コントローラのブロック図を図 5-34 に, レジスタを表 5-31 に示します。

図 5-34 クロック・コントローラのブロック図

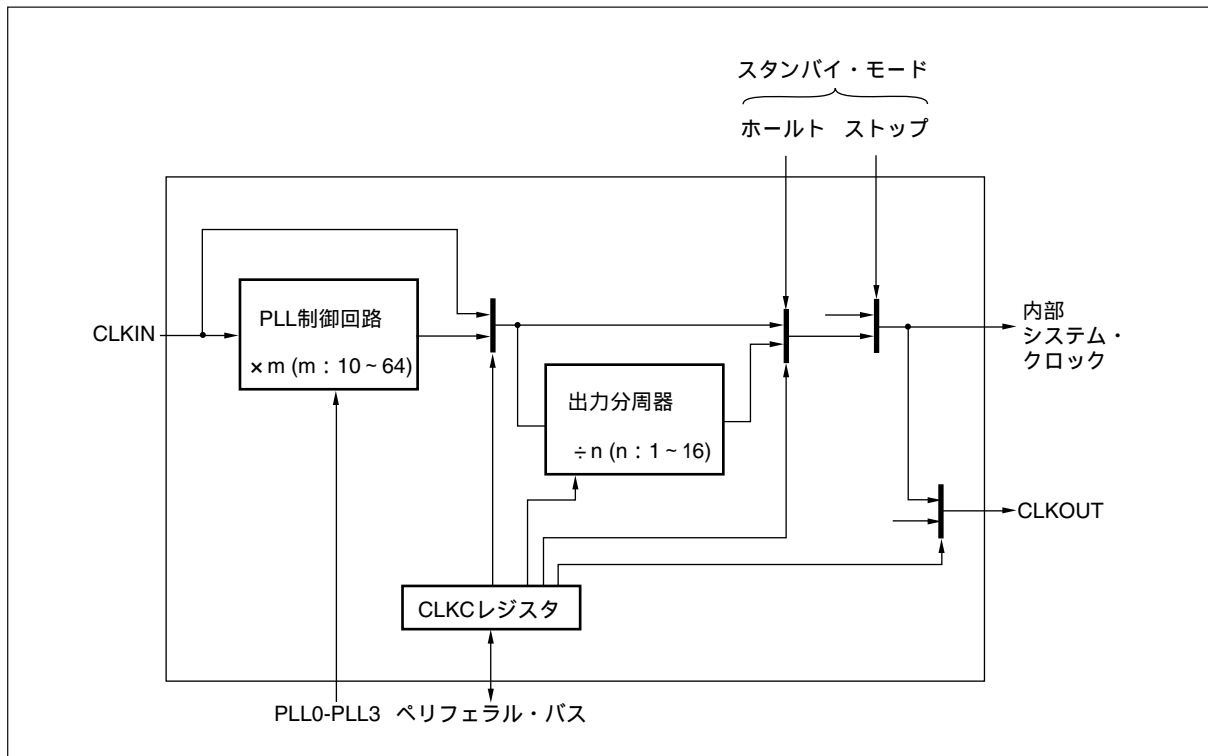


表 5-31 クロック・コントローラのレジスタ

X/Y メモリ・アドレス	レジスタ名	機 能	ロード/ストア
0x38B0	CLKC	クロック制御レジスタ	L/S

5. 12. 1 クロック・コントローラのレジスタ

(1) CLKC (クロック制御レジスタ : CLoCK Control Register)

PLL や分周器の設定を行うレジスタです。PLL のオン/オフ、クロック・ソースの選択、分周器の分周率、CLKOUT 端子の設定を行います。

CLKC の各ビットの機能については表 5-32に示します。リセットにより 0x0 になりますが、ブート中に変更されます。ODIVENA=1, ODIV=0001, またブート・モードに応じて PLLENA, PLLSEL が変更されます。

表 5-32 CLKC のビット構成 (1/2)

ビット	名称	機 能	ロード/ ストア
15-9	予約	0 以外の書き込みを行わないこと。 読み込み時は不定。	-
8	PLLENA	PLL イネーブル・ビット ・0: スタンバイ (デフォルト) ・1: 逓倍回路動作 PLL の動作を切り替えるビット。PLL を停止 / 起動することができる。	L/S
7	PLLSEL	PLL クロック選択ビット ・0: 外部クロック端子ダイレクト (デフォルト) ・1: PLL 逓倍クロック出力 システム内部に供給するクロックを切り替えるレジスタ。	L/S
6	CKOEN	CLKOUT イネーブル・ビット ・0: CLKOUT 端子をロウ・レベル出力固定 (デフォルト) ・1: CLKOUT 端子から内部システム・クロックを出力 CLKOUT 端子からシステムへの供給クロックを出力するか、停止するかを設定するビット。	L/S
5	ODIVENA	分周器イネーブル・ビット ・0: スタンバイ (デフォルト) ・1: 分周回路動作 分周器の動作を切り替えるビット。分周器を停止 / 起動することができる。	L/S
4	ODIVSEL	分周器クロック選択ビット ・0: 非分周クロック (デフォルト) ・1: 分周クロック システム内部に供給するクロックを切り替えるレジスタ。0 で外部クロック / PLL クロックを供給し、1 で分周クロックをシステムに供給する。	L/S

表 5-32 CLKC のビット構成 (2/2)

ビット	名称	機能	ロード/ ストア
3-0	ODIV	分周率設定ビット ・ 0000 : 1/16 (デフォルト) ・ 0001 : 1/1 ・ 0010 : 1/2 ・ 0011 : 1/3 ・ 0100 : 1/4 ・ 0101 : 1/5 ・ 0110 : 1/6 ・ 0111 : 1/7 ・ 1000 : 1/8 ・ 1001 : 1/9 ・ 1010 : 1/10 ・ 1011 : 1/11 ・ 1100 : 1/12 ・ 1101 : 1/13 ・ 1110 : 1/14 ・ 1111 : 1/15 分周器の分周率を指定するレジスタ。	L/S

プログラムによりクロック・ソースの切り替えが可能ですが、次の点に注意してください。

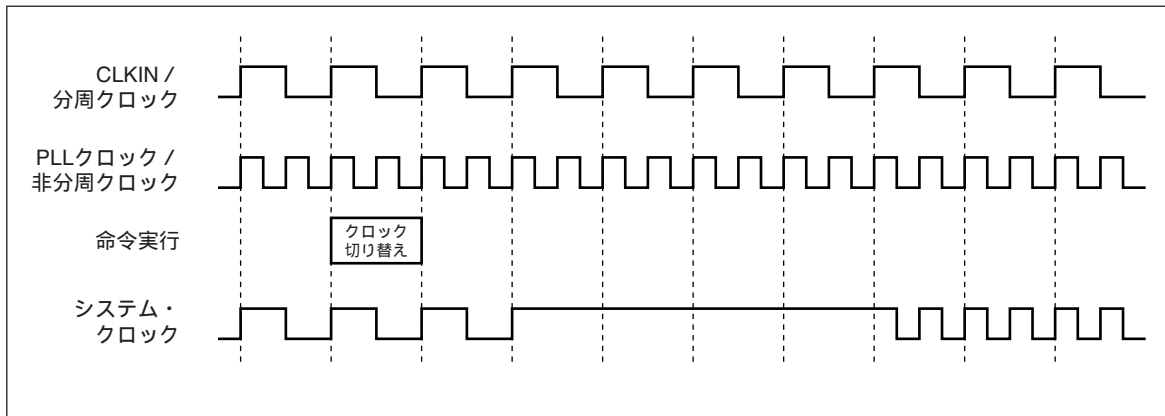
- 停止している分周クロック，未ロックの PLL クロックをソースとして選択しないでください。
- STOP / HALT では無条件に分周クロックをソースとして選択するので，STOP / HALT を実行する前に必ず分周器の設定をしてください。
- クロック・ソース変更後，16 命令サイクル以内に変更前のクロック・ソースを停止しないでください。また，クロック切り替え処理を繰り返すときは，16 命令サイクル以上を離して実行してください。
- 分周比率の変更は，分周器の停止（スタンバイ）状態で行ってください。

5. 12. 2 クロックの切り替えタイミング

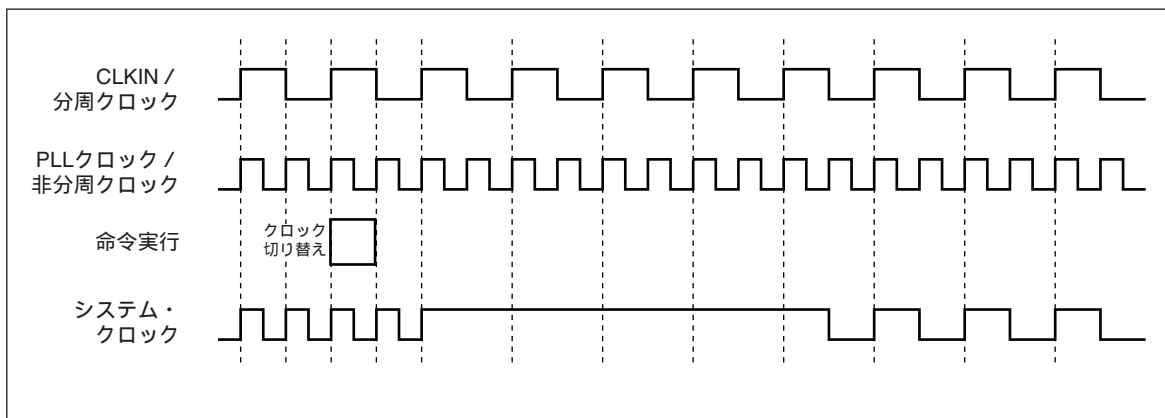
システム・クロックの切り替えが行われるとき、切り替え前後よりも長いハイ・レベル幅を持つ調整用のクロックが挿入されます。その様子を図 5-35に示します。その間に受け付けられた割り込み要求については、正常に検出できない可能性があります。システム・クロックが切り替えられるときには、割り込み要求が発生しない状況で行なってください。

図 5-35 クロック切り替えタイミング

(a) 外部クロック PLL 逡倍クロック (分周クロック 非分周クロック)



(b) PLL 逡倍クロック 外部クロック (非分周クロック 分周クロック)



この調整クロックのハイ・レベル幅は、次のようになります。

- 外部クロック→PLL 逡倍クロック (あるいは分周→非分周) でのクロック切り替え時
 $(2 \times \text{分周あるいは逡倍率} + 4.5) \times \text{非分周クロック周期あるいは PLL 逡倍クロック周期}$
- PLL 逡倍クロック 外部クロック (あるいは非分周→分周) でのクロック切り替え時
 $(8 \times \text{分周あるいは逡倍率} + 1) \times \text{非分周クロック周期あるいは PLL 逡倍クロック周期}$

この間、長期化されたシステム・クロックによるサンプリング欠損が生じる可能性があるため、タイマ、シリアル・インタフェースなどは停止してからクロック切り替え処理を行うようにしてください。

注意 シリアル・クロックのスペックなど、システム・クロックに対する規定がある場合、調整クロック中においては調整クロックもシステム・クロックとしての規定を満たさなければなりません。

たとえば、シリアル・クロック (スタンダード・シリアル・インタフェース) の規定が、「 $2 \times$ システム・ク

ロック (MIN.)」のとき、クロック移行時にシリアル・インタフェースを使用するなら、調整クロックを含めて2サイクル分のシステム・クロックが必要になります。

5.12.3 クロック制御の注意点

ここではクロック制御に関する注意点、誤りやすい点について説明します。

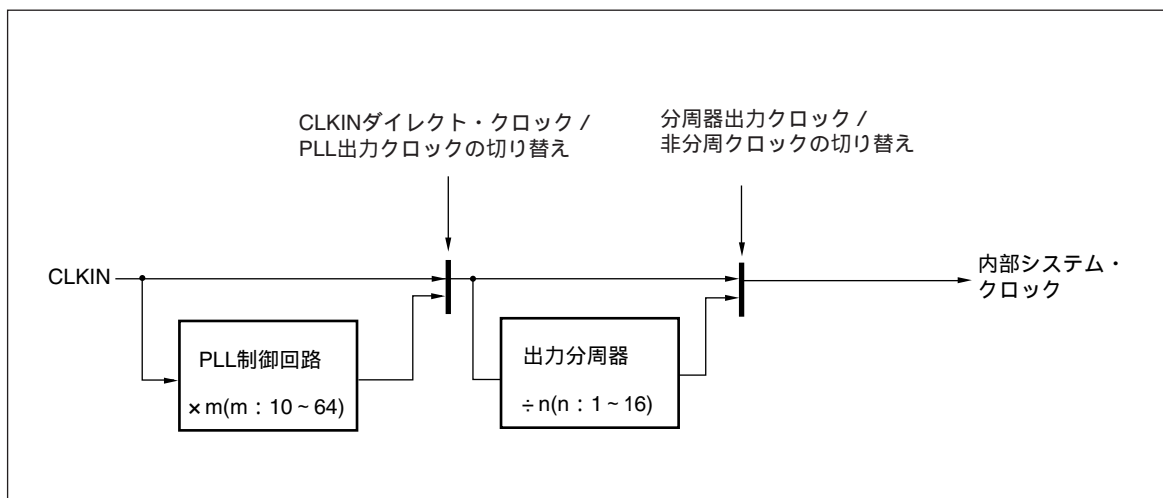
(1) クロックの切り替え

CLKIN 端子から入力されたクロックは、直接あるいは PLL 回路、分周器を通してシステム・クロックとして供給できます。起動時はリセット・アクティブによって CLKIN ダイレクト・クロックをシステム・クロックとして使用しますが、ブート過程において次のようになります。

- PLL 出力クロック (外部メモリ・ブート, ノン・ブート以外)
- CLKIN ダイレクト・クロック (外部メモリ・ブート, ノン・ブート)
- 分周器: 分周器動作 (1/1), 分周器出力クロック非選択

CLKIN 端子から入力されたクロックがシステム・クロックとして供給される関係を図 5-36に示します。PLL 逡倍の設定, CLKOUT 端子出力設定, およびスタンバイ・モードにおけるクロックに関しては 4.3.1 クロック・ジェネレータを参照してください。

図 5-36 クロックの切り替え



クロック・ソース (CLKIN ダイレクト, PLL 出力, 分周器出力) の切り替えは CLKC レジスタによって行います。選択 / 非選択の切り替え, PLL および分周器の切り替え設定にはそれぞれの切り替えから 16 実行サイクル必要であるので注意してください。また, 切り替え時には通常より長い幅を持つクロック (調整クロック) が挿入されるため, システム・クロックを用いてサンプリングしながら動作するペリフェラルでは, サンプリングの欠損が生じる可能性がありますので注意してください。

状態遷移を図 5-37に, PLL と分周期の設定例を図 5-38, 図 5-39に示します。

図 5-37 PLL 動作と分周動作の状態遷移

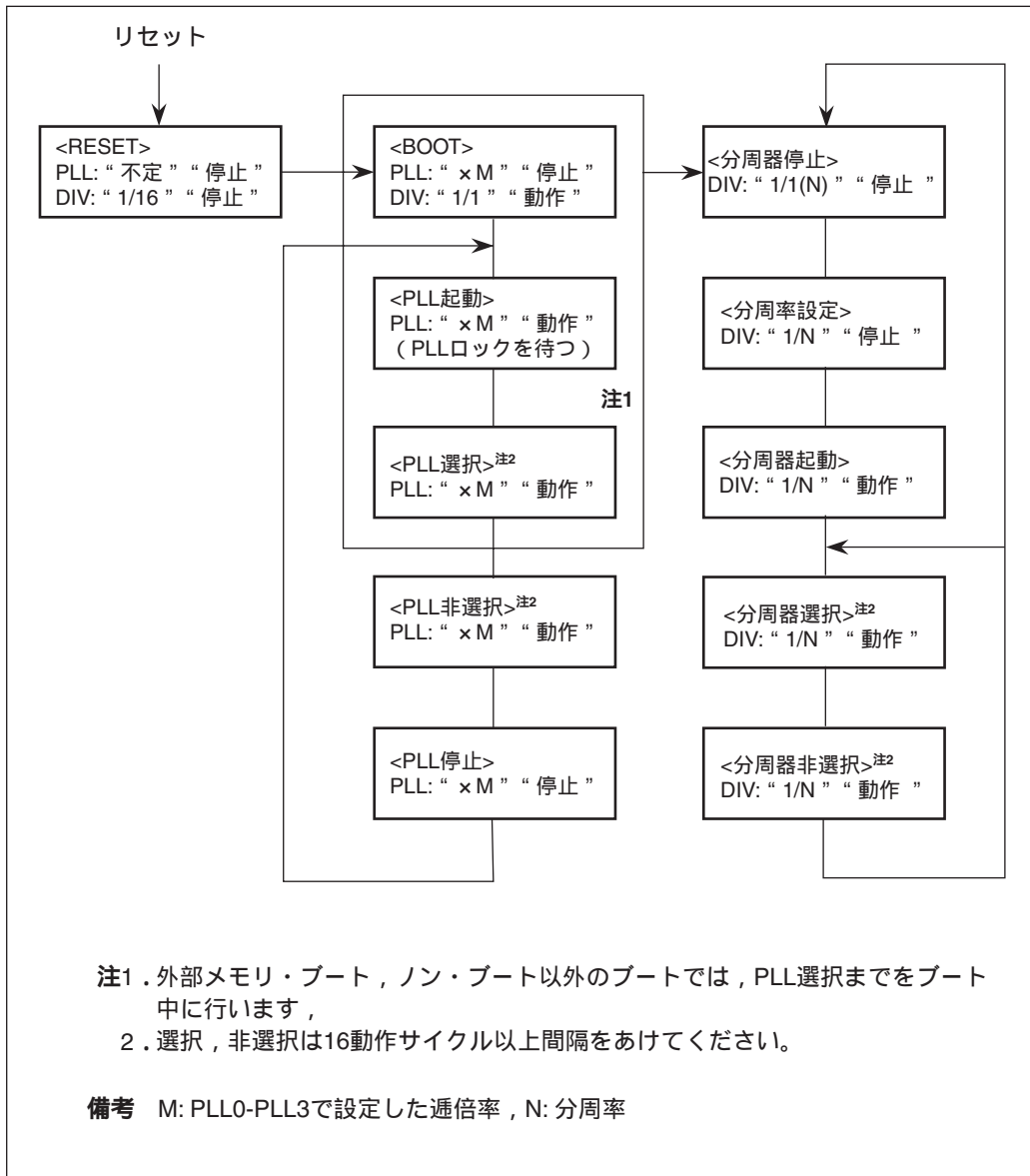


図 5-38 PLL の設定例

```

PLL 回路の起動
clr (r0)                ;
r0l = *CLKC:x          ;
r0 = r0 | 0x0100       ;
*CLKC:x = r0l          ;

PLL 出力クロックの選択 (PLL 回路が起動されていること, PLL がロックしていること)
clr (r0)                ;
r0l = *CLKC:x          ;
r0 = r0 | 0x0180       ;
*CLKC:x = r0l          ;

PLL 出力クロックの非選択
clr (r0)                ;
r0l = *CLKC:x          ;
r0 = r0 & 0x017f      ;
*CLKC:x = r0l          ;

PLL 回路の停止 (PLL 出力クロックが非選択になっていること)
clr (r0)                ;
r0l = *CLKC:x          ;
r0 = r0 & 0x00ff      ;
*CLKC:x = r0l          ;

```

図 5-39 分周器の設定例

```

分周率の設定 (分周器が停止していること)
clr (r0)                ;
r0l = *CLKC:x          ;
r0 = r0 & 0x01f0       ;
*CLKC:x = r0l          ;
r0 = r0 | 0x000z      ; z = divide ratio
*CLKC:x = r0l          ;

分周器の起動
clr (r0)                ;
r0l = *CLKC:x          ;
r0 = r0 | 0x0020       ;
*CLKC:x = r0l          ;

分周器出力クロックの選択 (分周器が起動されていること)
clr (r0)                ;
r0l = *CLKC:x          ;
r0 = r0 | 0x0010       ;
*CLKC:x = r0l          ;

分周器出力クロックの非選択
clr (r0)                ;
r0l = *CLKC:x          ;
r0 = r0 & 0x01ef      ;
*CLKC:x = r0l          ;

分周器の停止 (分周器出力クロックが非選択になっていること)
clr (r0)                ;
r0l = *CLKC:x          ;
r0 = r0 & 0x01df      ;
*CLKC:x = r0l          ;

```

(2) スタンバイ・モード時のクロック

スタンバイ・モード (HALT / STOP) 時は自動的に分周器出力クロックが選択されます。したがって、スタンバイ・モード使用前に必ず分周器を起動しておく必要があります。

分周器出力クロックで動作中に HALT モードを使用するときは、分周器出力クロックのままなのでクロックの切り替えは行われません。

STOP モード時は同時にシステムへのクロック供給を停止します。PLL は起動したままなので、消費電力を削減したい場合は、必要に応じて STOP モードを使用する前に PLL を停止する処理を行ってください。

(3) クロック・モニタ出力

CLKOUT 端子より内部システム・クロックのモニタ出力が可能です。

状態遷移を図 5-40に、CLKOUT 出力の設定例を図 5-41に示します。切り替え時には CLKOUT 端子からの出力クロックにハザードが発生することがあります。

図 5-40 クロック・モニタ出力の状態遷移

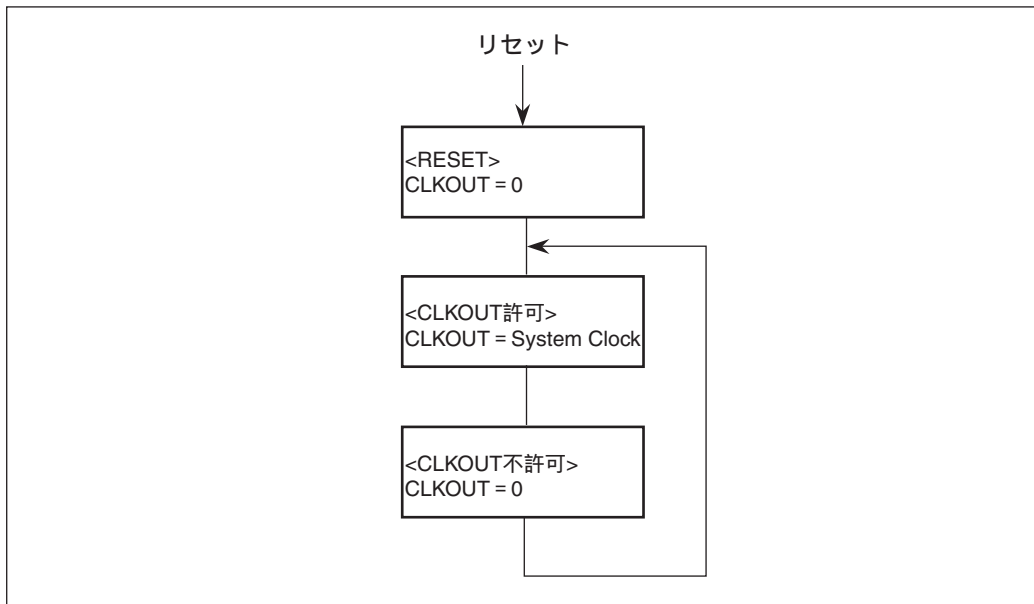


図 5-41 クロック・モニタ出力の設定例

```

CLKOUT 出力許可
clr (r0)           ;
r0l = *CLKC:x     ;
r0 = r0 | 0x0040  ;
*CLKC:x = r0l    ;

CLKOUT 出力不許可
clr (r0)           ;
r0l = *CLKC:x     ;
r0 = r0 & 0xFFDF ;
*CLKC:x = r0l    ;
  
```

5.13 命令メモリ訂正機能 (IMC)

命令メモリ訂正機能(IMC: Instruction Memory Correction)は、命令メモリをあらかじめ設定したレジスタ上のデータと差し替えてフェッチするための機能です。命令メモリを訂正する機能であり、マスク ROM 品である μ PD77213においてROM内容にパッチを当てるときに使用します。

主な特徴を次に示します。

- 訂正可能語数

最大2命令までの訂正が可能

- ノー・オーバヘッド

通常動作と同様にオーバヘッドが生じないノー・ウエイト動作

命令メモリ訂正機能のレジスタ群を表 5-33に示します。

表 5-33 命令メモリ訂正機能のレジスタ

XYメモリ・アドレス	レジスタ名	機能	ロード/ストア
0x38A0	CEFR	訂正イネーブル・フラグ・レジスタ	L/S
0x38A1	CPR0	訂正ページ・レジスタ0	L/S
0x38A2	CAR0	訂正アドレス・レジスタ0	L/S
0x38A3	CLIR0	訂正命令コード・レジスタ(上位)0	L/S
0x38A4	CUIR0	訂正命令コード・レジスタ(下位)0	L/S
0x38A5	CPR1	訂正ページ・レジスタ1	L/S
0x38A6	CAR1	訂正アドレス・レジスタ1	L/S
0x38A7	CLIR1	訂正命令コード・レジスタ(上位)1	L/S
0x38A8	CUIR1	訂正命令コード・レジスタ(下位)1	L/S

5.13.1 命令メモリ訂正機能のレジスタ

(1) CEFR (訂正イネーブル・フラグ・レジスタ: Correct Enable Flag Register)

訂正した内容の有効/無効を設定するレジスタです。

CEFRの各ビットの機能については表 5-34に示します。デフォルトは、0x0です。

(2) CPR0, CPR1 (訂正ページ・レジスタ: Correct Page Register)

訂正対象となる命令メモリのページを指定する16ビットのレジスタです。下位6ビットに対してページを指定します。指定できる値は0x0-0x3Fで、実際にROMが存在するページです。

(3) CAR0, CAR1 (訂正アドレス・レジスタ: Correct Address Register)

訂正対象となる命令メモリのアドレスを指定する16ビットのレジスタです。指定できる値は0x8000-0xFFFFで、実際にROMが存在するアドレスです。

(4) CUIR0, CUIR1, CLIR0, CLIR1 (訂正命令コード・レジスタ: Correct Instruction Register)

訂正する(パッチ処理を行う)命令コードを設定する 16 ビットのレジスタです。命令メモリは 32 ビット幅であるため、上位 16 ビット用(CUIR)、下位 16 ビット用(CLIR)の 2 つを備えています。

表 5-34 CEFR のビット構成

ビット	名称	機能	ロード/ストア
15-2	予約	0 以外の書き込みを行わないこと。 読み込み時は不定。	-
1	Correct1 enable	訂正語 1 イネーブル・ビット ・0: ディスエーブル(デフォルト) ・1: イネーブル 訂正語 1 (CPR1, CAR1, CUIR1, CLIR1) に設定した値で訂正機能を有効/無効にする。	L/S
0	Correct0 enable	訂正語 0 イネーブル・ビット ・0: ディスエーブル(デフォルト) ・1: イネーブル 訂正語 0 (CPR0, CAR0, CUIR0, CLIR0) に設定した値で訂正機能を有効/無効にする。	L/S

5.13.2 命令訂正機能の動作

訂正する命令メモリのページ IPR の値を CPR, アドレスを CAR, 訂正する命令コードを CUIR および CLIR に設定します。そのあとで CEFR のイネーブル・フラグをセットすることにより、命令訂正機能が起動します。命令フェッチ時にインストラクション・ポインタ(IP)と CAR を比較し、一致するとレジスタに設定した命令コードをフェッチします。最大 2 語の訂正が可能です。

- 注意 1.** 命令メモリ訂正のイネーブル制御(CEFR のレジスタへのアクセス)は、0x0000-0x7FFF の命令 RAM 上にて実行してください。
- 2.** イネーブル制御直後の 2 命令を訂正対象にはできません。

5.14 ページング機能

μ PD77210 ファミリの DSP コア・カーネルは 16 ビット・アドレス空間(64 Kワード)をサポートしています。この 64 Kワードを拡張する機能をページングといいます。

命令およびデータ空間のアドレス 0x8000-0xFFFF をページング領域とし、ここに複数のページを割り当てることでメモリ領域拡張を行います。

データ・ページには、拡張用データ・ページ以外にも外部データ・メモリへのアクセス空間、命令メモリ領域へのアクセス空間を割り当てることができます。

ページング機能のレジスタ群を表 5-35に示します。

表 5-35 ページング機能のレジスタ

X/Y メモリ・アドレス	レジスタ名	機能	ロード/ストア
0x38C0	IPR	命令ページ・レジスタ	L/S
0c38C1	DPR	データ・ページ・レジスタ	L/S

5. 14. 1 ページング機能のレジスタ

(1) IPR (命令ページ・レジスタ : Instruction Page Register)

命令ページング領域の命令ページを指定する 16 ビットのレジスタです。下位 6 ビットに対してページを指定します。指定できる値は 0x0-0x3F です。デフォルトは 0x0 です。

(2) DPR (データ・ページ・レジスタ : Data Page Register)

データ・ページング領域のデータ・ページを指定する 16 ビットのレジスタです。下位 8 ビットに対してページを指定します。指定できる値とページ指定対象領域は表 5-36のとおりです。デフォルトは 0x0 です。

表 5-36 DPR の設定と対象領域

DPR	ページ指定対象領域
0x00-0x3E	データ・ページ
0x3F	外部データ・メモリ・ウインドウ
0x80	命令メモリ (0x0000-0x7FFF)
0xC0-0xFF	命令メモリ (0x8000-0xFFFF) ページ

5. 14. 2 ページング機能の動作

アクセス対象とするページを IPR, DPR に設定することによって, ページング領域に対して該当のページが設定されます。

- 注意**
1. ページ切り替え (IPR, DPR レジスタ・アクセス) は, 0x0000-0x7FFF の命令 RAM 上にて実行してください。
 2. ページ切り替え (IPR, DPR レジスタ・アクセス) を, ページング領域 0x8000-0xFFFF のデータ・アクセスと並列に実行することを禁止します。また, ページ切り替え直後の命令で, ページング領域 0x8000-0xFFFF のデータ・アクセスを実行することを禁止します。
 3. 命令アドレス 0x0000-0x7FFF における DPR=0x80 でのデータ・ページ・メモリ・アクセスでは, 同一メモリに対して命令フェッチとデータ・アクセスが発生するため, 1 ウェイトし 2 サイクル・フェッチとなります。命令アドレス 0x8000-0xFFFF, IPR=0x0-0x3F における DPR=0xC0-0xFF でのデータ・ページ・メモリ・アクセスも同様です。

5.15 ペリフェラルSTOPモード

ペリフェラルSTOPモードは、TSIO、ASIO、MIO、PMT に対するクロック供給を個別に停止可能にするモードです。本モードにより、未使用の周辺回路における電力消費を軽減します。

ペリフェラルSTOPモードのレジスタを表 5-37に示します。

表 5-37 ペリフェラルSTOPモードのレジスタ

XYメモリ・アドレス	レジスタ名	機能	ロード/ストア
0x387A-0x387B	POWC	パワー制御レジスタ	S

5.15.1 ペリフェラルSTOPモードのレジスタ

(1) POWC (パワー制御レジスタ : Power Control Register)

ペリフェラルの動作/停止を設定するレジスタです。デフォルトは0x0です。

POWCの各ビットの機能については表 5-38に示します。

表 5-38 POWCのビット構成

ビット	名称	機能	ロード/ストア
15-4	予約	0以外の書き込みを行わないこと。 読み込み時は不定。	-
3	PMTENA	PMT イネーブル・ビット ・0 : アクティブ (デフォルト) ・1 : 停止	S
2	MIOENA	MIO イネーブル・ビット ・0 : アクティブ (デフォルト) ・1 : 停止	S
1	ASIOENA	ASIO イネーブル・ビット ・0 : アクティブ (デフォルト) ・1 : 停止	S
0	TSIOENA	TSIO イネーブル・ビット ・0 : アクティブ (デフォルト) ・1 : 停止	S

5. 15. 2 ペリフェラル STOP モードの動作

POWC レジスタの設定は、0x387A を PDT, 0x387B を PCD と見立てたポート制御として取り扱います。出力モードに設定したのち、各ビット操作を行います。設定例を図 5-42に示します。

出力のみ可能であり、入力値は不定です。

★

図 5-42 POWC レジスタの設定例

```

POWC レジスタを出力設定
clr (r0)                ;
r0l = 0x300F           ;
*PCD(POWC):x = r0l    ;

TSIO 停止
clr (r0)                ;
r0l = 0x0001           ;
*PDT(POWC):x = r0l    ;

TSIO アクティブ
clr (r0)                ;
r0l = 0x000e           ;
*PDT(POWC):x = r0l    ;

```

5. 16 ディバグ・インタフェース (IEIO)

μPD77210 ファミリは、JTAG インタフェースに準拠した次の機能を装備しています。

- JTAG ポート
- バウンダリ・スキャン・テスト機能
- ディバグ機能 (インサーキット・エミュレーション機能)

5. 16. 1 JTAG ポート

JTAG (Joint Test Action Group) は、電子装置基板のテスト容易化手法であるバウンダリ・スキャンの規格化を推進するために組織された団体で、これによる規格化案は IEEE1149.1 として勧告されています。

JTAG に準拠したデバイスはテスト専用のアクセス・ポートを持ち、内部のロジックとは独立にデバイスをテストすることができます。

μPD77210 ファミリでは、勧告で必須としているインストラクション・レジスタ、バイパス・レジスタおよびバウンダリ・スキャン・レジスタに加え、インサーキット・エミュレーション用のレジスタと制御回路を内蔵しています。

JTAG の詳しい内容については、IEEE1149.1 を参照してください。

5. 16. 2 ディバグ・インタフェースの端子

勧告に従う4端子と、インサーキット・エミュレーション用の端子(TICE), テスト用の端子($\overline{\text{TRST}}$)があります。

(1) TCK (入力)

テスト・クロック入力端子です。未使用時は0を入力します(勧告準拠)。

TCK端子に入力するクロックは、ディバグ使用時はCLKIN端子に入力するクロックの1/2未満にしてください。ハードウェア・ディバグの設定から625 kHz-5 MHz(4段階)で選択できます。

注意 TCKをハイ・レベルで停止しないでください。

(2) TMS (入力)

テスト・モード・セレクト入力です。TCKの立ち上がりエッジでサンプリングされます。内部ではプルアップされています。

(3) TDI (入力)

テスト・データ入力端子です。TCKの立ち上がりエッジでサンプリングされます。内部ではプルアップされています。

(4) TDO (出力)

テスト・データ出力端子です。TCKの立ち下がりエッジに同期して出力が変化します。

(5) TICE (出力)

インサーキット・エミュレーションのブレイク・モードを構築するための出力端子です。

(6) $\overline{\text{TRST}}$ (入力)

テスト・リセット入力端子です。通常動作時は0(アクティブ・ロウ)に、ディバグ使用時、バウンダリ・スキャン時に1にします。

内部でプルダウンされています。

5.16.3 バウンダリ・スキャン・テスト機能

バウンダリ・スキャン・テスト方式では、ターゲット・システムのボード・レベル・テスト、チップ・レベル・テストを、一貫したテスト・フェーズとして実現できるため、近年、自動テスト・システムなどで製造現場に広く取り入れられているテスト方式です。

次に、 μ PD77210 ファミリに装備されているバウンダリ・スキャン機能を説明します。

(1) テスト・インストラクション・レジスタ

テスト項目とテスト・データ・レジスタを選択するための、8 ビットのレジスタです。サポートしているインストラクションを表 5-39に示します。

表 5-39 テスト・インストラクション

ビット	命 令
7 6 5 4 3 2 1 0	
0 0 0 0 0 0 0 0	EXTEST 命令
0 0 0 0 0 0 1 0	SAMPLE/PRELOAD 命令
1 1 1 1 1 1 1 0	BYPASS 命令

(2) テスト・バイパス・レジスタ

TDI から入力したデータを、TDO に出力するレジスタです。

5. 16. 4 ディバグ機能（インサーキット・エミュレータ機能）

μ PD77210 ファミリでは、JTAG を利用したディバグ機能を備え、ランタイムでプログラムを監視する機構を装備しています。次のような機能があります。

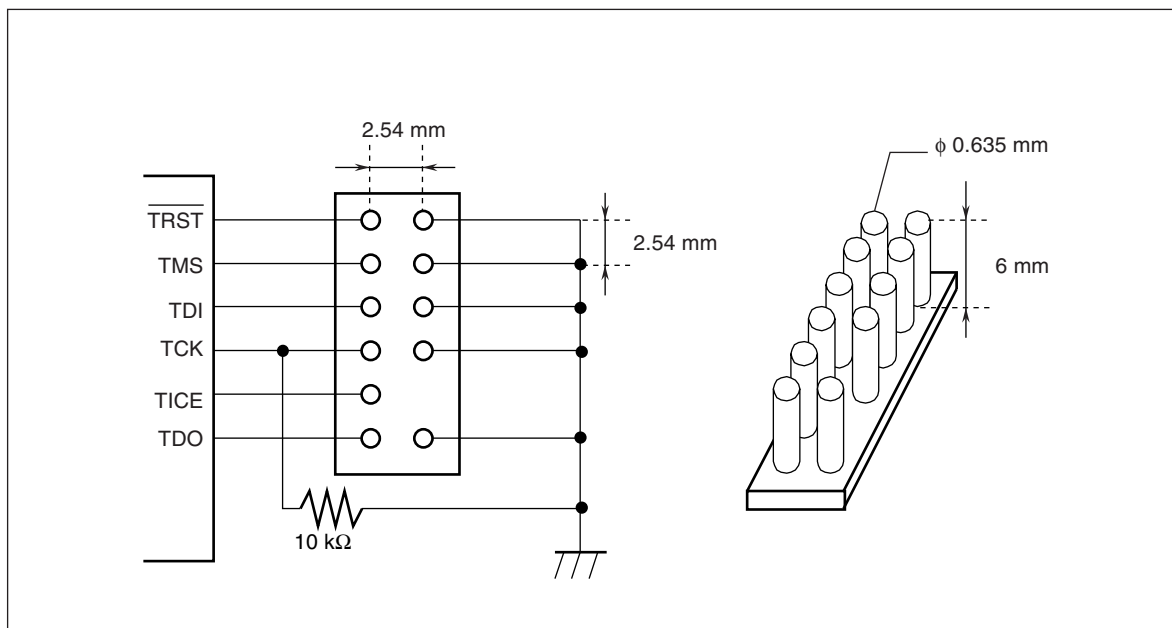
- ブレーク機能
 - 指定した命令メモリ・アドレスのフェッチによるブレーク
 - 指定したデータ・メモリ・アドレスのリード/ライトでブレーク
- ノンブレーク・モニタ機能
 - プログラム実行中にレジスタや、メモリの内容を参照/変更

注意1. ディバグ機能はユーザ向けに公開されたものではありません。

2. 推奨するハードウェア・ディバग्ガは、NDSP_ICE002（アンドール・システム・サポート株式会社製）です。NDSPICE002 を使用する場合の JTAG 端子処理を図 5-43に示します。

ユーザ・システムで NDSP_ICE002 を使用しない場合は、2.4 未使用端子の処理についてにしたがって適切に処理してください。

図 5-43 JTAG 端子処理



5.17 拡張インタフェース (Additional IO)

インタフェース追加に対応する IO です。SD カード・インタフェースはこの IO を利用してインタフェースされます。各拡張インタフェースに関しては次項以降に説明します。ここでは、拡張インタフェースそのものについて説明します。拡張インタフェースを内蔵しているのは μ PD77213 だけです。 μ PD77210 には存在しません。

外部データ・メモリ空間の一部 (0xC0000-0xFFFFF) を拡張インタフェースのレジスタ・マッピング領域として利用します。拡張インタフェースを使用しない場合は、外部データ・メモリ空間として利用可能です。

MIO を介して拡張インタフェースのレジスタをアクセスする構造となっており、ユーザからは外部データ・メモリの一部として拡張インタフェースが存在する形になります。

拡張インタフェースのブロック図を図 5-44 に、レジスタを表 5-40 に示します。

図 5-44 拡張インタフェースのブロック図

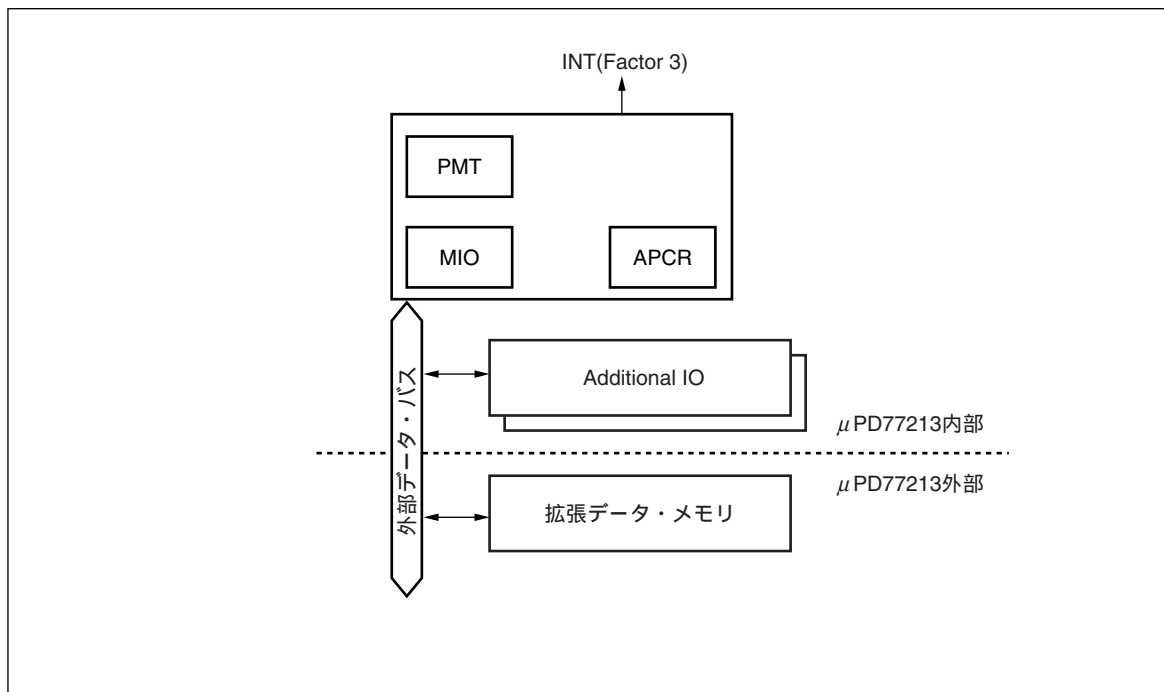


表 5-40 拡張インタフェースのレジスタ

X/Y メモリ・アドレス	レジスタ名	機能	ロード/ストア
0x38D0	APCR	拡張インタフェース制御レジスタ	L/S

5. 17. 1 拡張インタフェースのレジスタ

(1) APCR (拡張インタフェース制御レジスタ: Additional Peripheral Control Register)

拡張インタフェースのイネーブル制御, PMT 要求制御の設定を行うレジスタです。4 ブロックまでのペリフェラルを制御します。1 ブロックあたり 1 組のロード/ストアの割り込みを制御できます。複数のロード/ストアの割り込みを制御する割り込みでは複数のブロックを使用することになります。

APCR の各ビットの機能について表 5-41 に示します。デフォルトは 0x0 です。

表 5-41 APCR のビット構成

ビット	名称	機能	ロード/ストア	
15	Peripheral A	SRE	ストア要求許可フラグ ペリフェラルから PMT チャンネル (ch7) へのストア要求信号イネーブル。 0: ディスエーブル (ストア要求をマスク) 1: イネーブル (ストア要求を透過)	L/S
14		LRE	ロード要求許可フラグ ペリフェラルから PMT チャンネル (ch7) へのストア要求信号イネーブル。 0: ディスエーブル (ロード要求をマスク) 1: イネーブル (ロード要求を透過)	L/S
13		PFE	ペリフェラル機能許可フラグ ペリフェラル機能の動作と停止を設定する。ディスエーブルではペリフェラルにクロックを供給しない。 0: イネーブル 1: ディスエーブル	L/S
12		TOE	端子出力許可フラグ リセット・アクティブ時のペリフェラルの出力端子の状態を設定する。 0: ディスエーブル (ハイ・インピーダンスとする) 1: イネーブル (ペリフェラルに依存する) PFE=1 とした場合, イネーブル時はアドレス出力となる。	L/S
11-8	Peripheral B	Peripheral A と同じ		
7-4	Peripheral C			
3-0	Peripheral D			

注意 PMT 転送を行う場合, LRE, SRE は 4 ブロック中排他的に設定しなければなりません。Peripheral A で “1” に設定したら, Peripheral B,C,D では “1” に設定しないでください。

5.18 SD カード・インタフェース (SDCIF)

μPD77213 は SD (Secure Digital) カード・インタフェース (SDCIF) を内蔵しています。このインタフェースは拡張インタフェース上に構成されています。

SD カード・インタフェース仕様は、SD Memory Card Specifications Part1 PHYSICAL LAYERSPECIFICATION Version 1.0 March 2000 に基づいています。

主な特徴を次に示します。

- 3 端子

クロック (SDCLK), コマンド/レスポンス (SDCR), データ (SDDAT0)

- 内部ハンドシェーク

ポーリングまたは割り込みによるハンドシェーク

SD カード・インタフェースのブロック図を図 5-45 に、レジスタ群を表 5-42 に示します。

図 5-45 SD カード・インタフェースのブロック図

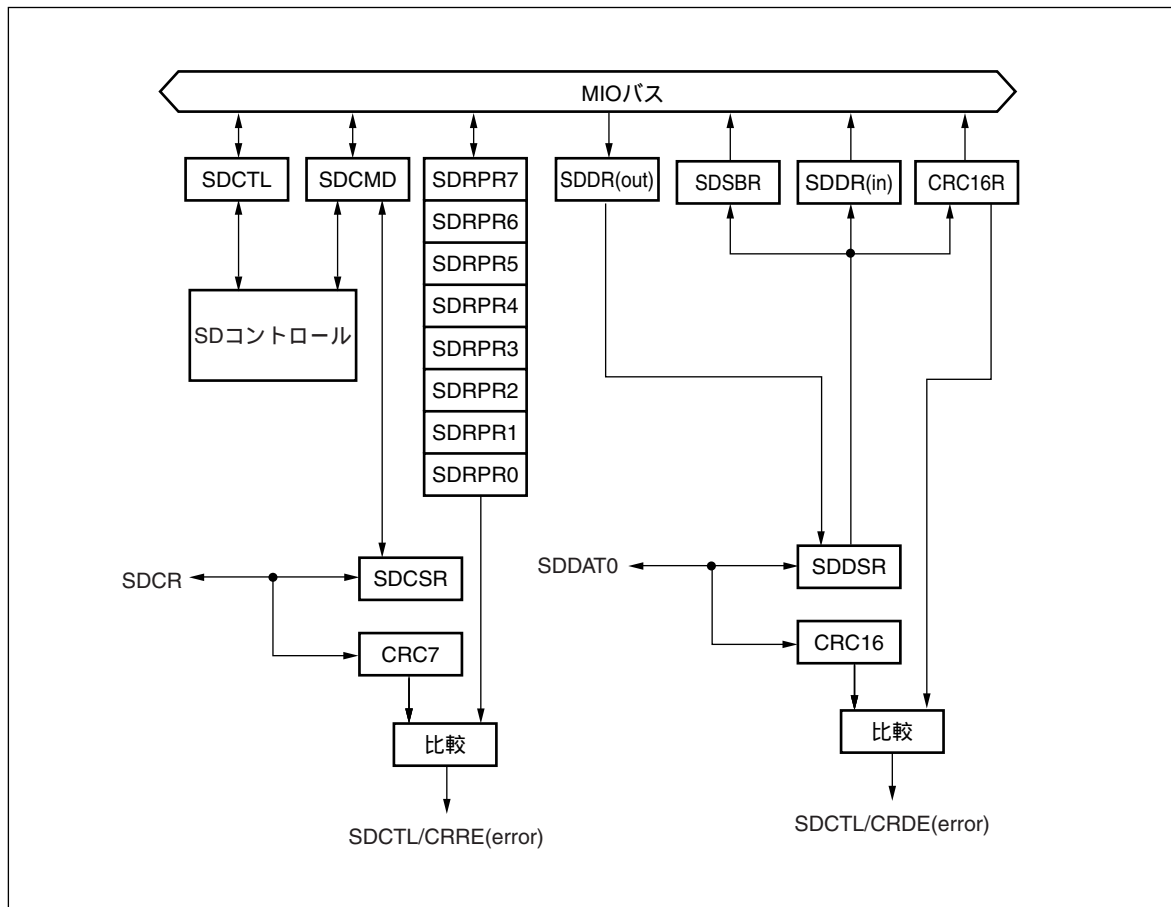


表 5-42 SD カード・インタフェースのレジスタ

外部メモリ・アドレス	レジスタ名	機能	ロード/ストア
0xFC000	SDDR	SD カード・データ・レジスタ	L/S
0xFC001	SDCMD_IDX	SD カード・コマンド・インデクス・レジスタ	L/S
0xFC002	SDCMD_AGH	SD カード・コマンド・アーギュメント・レジスタ (H)	L/S
0xFC003	SDCMD_AGL	SD カード・コマンド・アーギュメント・レジスタ (L)	L/S
0xFC004	SDCTL	SD カード・インタフェース制御レジスタ	L/S
0xFC005	SDRPR	SD カード・インタフェース・レスポンス・レジスタ	L
0xFC006	SDSBR	SD カード・インタフェース CRC ステータス・ビジィ・レジスタ	L
0xFC007	CRC16R	SD カード・インタフェース・コマンド CRC レジスタ	L

5. 18. 1 SD カード・インタフェースの端子

(1) SDCLK (SD カード・クロック - 出力)

SD カードのクロックとして、SD カードの CLK 端子に接続する出力端子です。SDCLK 端子の最大動作周波数は 25 MHz です。

SD カード・インタフェース内のシフト・レジスタも SDCLK クロックに同期してシフト動作を行います。

(2) SDCR (SD カード・コマンド/レスポンス - 入出力)

コマンド出力とレスポンス入力を行う端子で SD カードの CMD 端子に接続します。 μ PD77213 から SD カードに対してコマンドを送信し、SD カードからレスポンスを受信する時分割の双方向バスになっています。また、アクセスしていないときは、デバイス外部ボード上のプルアップにより EV_{DD} レベルとしてください。

(3) SDDAT0 (SD カード・データ - 入出力)

データの入出力を行う端子で SD カードの DAT0 端子に接続します。 μ PD77213 から SD カードに対してリード・コマンド送信後にデータをリード、あるいはライト・コマンド送信後にデータをライトします。コマンドによる切り替え構成で時分割の双方向バスになっています。また、アクセスしていないときは、デバイス外部ボード上のプルアップにより EV_{DD} レベルとしてください。

(4) SDMON (SD カード・アクセス・モニタ - 出力)

SD カード・インタフェースのアクセス・モニタ端子です。SD カード・インタフェースをアクセス中 (MIO にて、0xFC000-0xFFFFF をアクセス中) にハイ・レベルを出力します。このとき MA0-MA12 はロウ・レベルを出力します。

5. 18.2 SD カード・インタフェースのレジスタ

(1) SDDR (SD カード・データ・レジスタ : SD Card Data Register)

SD カードのシリアル・データを入出力するための 16 ビット・レジスタです。出力用と入力用のレジスタを別々に持っています。SDDR の値は MIO を経由して入出力できます。

SDDR (out) は、SD カードへ出力するデータを設定する 16 ビットのレジスタです。SDDR に対するストア命令を実行すると SDDR (out) に MIO バスからデータを入力します。SD カードに対してライト・モードのときに SD カード・データ・シフト・レジスタ (SDDSR) が空になると、SDDR (out) の値を SDDSR に設定します。SDCTL レジスタの SDS ビットが「0」のときに SDDR に対するストア命令を実行すると、SDCTL レジスタの SDSE ビットが 1 になります (ストア・エラー)。

SDDR (in) は、SD カードから入力されたデータを読み出す 16 ビットのレジスタです。SDDR (in) に対するロード命令を実行すると、SDDR (in) から MIO バスにデータを出力します。SD カードに対してリード・モードのときに SD カード・データ・シフト・レジスタ (SDDSR) に最後のビットが入力されると、SDDSR の値を SDDR (in) に設定します。SDCTL レジスタの SDL ビットが 0 のときに SDDR に対するロード命令を実行すると、SDCTL レジスタの SDLE ビットが 1 になります (ロード・エラー)。

(2) SDCMD (SD カード・コマンド・レジスタ : SD Card Command Register)

SD カードを制御するためのコマンドを設定するレジスタで、3つの 16 ビット・レジスタから構成されます。

コマンド解釈およびコマンド・インデクス出力用 (SDCMD_IDX) とコマンド・アークギュメントの上位側用 (SDCMD_AGH) とコマンド・アークギュメントの下位側用 (SDCMD_AGL) のレジスタを別々に持っています。

SDCMD の値は MIO を経由して入出力できます。

(a) SD カード・コマンド・インデクス・レジスタ (SDCMD_IDX)

SD カードへ出力するコマンドのインデクスと、レスポンス・データ制御するためのコマンド解釈情報を設定する 16 ビットのレジスタです。SDCMD_IDX に対するストア命令を実行すると、SDCMD_IDX に MIO バスからデータを入力します。SDCTL の CEN ビットに 1 をセットすると、SDCMD_IDX のビット 5-0 の値を SDCSR に設定し、SD カードへ送信します。リセット初期値は、0x4800 です。SDCMD_IDX の各ビットの機能を表 5-43に示します。

(b) SD カード・コマンド・アークギュメント・ハイ側レジスタ (SDCMD_AGH)

SD カードへ出力するコマンド・アークギュメントの上位側を設定する 16 ビットのレジスタです。SDCMD_AGH に対するストア命令を実行すると、SDCMD_AGH に MIO バスからデータを入力します。SDCTL の CEN ビットに 1 をセットすると SDCMD_AGH の値を SDCSR に設定し、コマンドのビット 39-24 に割り当てて SD カードへ送信します。

(c) SD カード・コマンド・アークギュメント・ロウ側レジスタ (SDCMD_AGL)

SD カードへ出力するコマンド・アークギュメントの下位側を設定する 16 ビットのレジスタです。SDCMD_AGL に対するストア命令を実行すると、SDCMD_AGL に MIO バスからデータを入力します。SDCTL の CEN ビットに 1 をセットすると SDCMD_AGL の値を SDCSR に設定し、コマンドのビット 23-8 に割り当てて SD カードへ送信します。

(3) SDCTL (SD カード・インタフェース制御レジスタ : SD Card Interface Control Register)

SD カード・インタフェースを制御するための値を設定する 16 ビットのレジスタです。SDCTL の値は MIO バスを經由して入出力できます。

リセット初期値は、0x0202 です。SDCTL の各ビットの機能を表 5-44 に示します。

(4) SDRPR (SD カード・インタフェース・レスポンス・レジスタ : SD Card Interface Response Register)

SD カードから受信したレスポンス・データを保持するレジスタです。FIFO (8×16 ビット) で構成されます。

SDRPR は SDRPR7-SDRPR0 のレジスタを別々に持っており、SDCMD_IDX の BSL ビットが 1 (レスポンス・ビット : 136 ビット) の場合、SDRPR7 から順番にレスポンス・データを保持していきます。SDCMD_IDX の BSL ビットが 0 (レスポンス・ビット : 48 ビット) の場合、SDRPR2 から順番にレスポンス・データを保持し、SDRPR3-SDRPR7 には意味のないデータが入っています。

SDRPR は、1 つのベリフェラル・アドレスにマッピングされています。プログラムからは 8 回のロードを繰り返すことで SDRPR0-SDRPR7 の値を MIO バスに出力できます。

(5) SDDSR (SD カード・インタフェース・データ・シフト・レジスタ : SD Card Interface Data Shift Register)

SD カードとシリアル・データをインタフェースするために SDDAT0 端子を經由して、直接データの送受信を行う 16 ビット・レジスタです。SDDSR は MIO バスに接続されていません。

(6) SDCSR (SD カード・インタフェース・コマンド・シフト・レジスタ : SD Card Interface Command Shift Register)

SD カードとインタフェースするため SDCR 端子を經由して、直接コマンド / レスポンスの送受信を行う 16 ビット・レジスタです。SDCSR は MIO バスに接続されていません。

(7) SDCRCR (SD カード・インタフェース CRC レジスタ : SD Card Interface Control Register)

SD カードから受信したレスポンスの CRC7 ビットを保持するレジスタ (SDCRCR7) と、SD カードから受信したリード・データの CRC16 ビットを保持するレジスタ (SDCRCR16) を別々に持っています。

SDCRCR7 は、SD カードから受信したレスポンスの CRC7 ビットを保持するレジスタで SDRPR0 のビット 1-7 と兼用しています。SDCRCR7 で保持した 7 ビットの CRC データと μ PD77213 内で受信したレスポンスの CRC 計算結果を比較し、不一致の場合にエラー・フラグを立てます。SDRPR0 の一部であるためロードは可能であるが、ストアすることはできません。

SDCRCR16 は、SD カードから受信したリード・データの CRC16 ビットを保持する 16 ビット・レジスタです。SDCRCR16 で保持した 16 ビットの CRC データと μ PD77213 内で受信したリード・データの CRC 計算結果を比較し、不一致の場合にエラー・フラグを立てます。SDCRCR16 は、ロード、ストアとも可能です。

(8) SDSBR (SD カード・インタフェース CRC ステータス・ビジィ・レジスタ : SD Card Interface Control Register)

SD カードヘデータを送信したあとに、SD カードから SD カード・インタフェースに対して送る CRC ステータスとビジィ情報を保持するレジスタです。

SDSBR レジスタは 4 ビットで構成され、ビット 3 がビジィ・フラグ、ビット 0-2 が CRC ステータスの機能になっています。ロードのみ可能です。

ビジィ・フラグは、1 のときビジィの状態を示し、0 はビジィではない状態を示します。

CRC ステータス・ビットは、SD カードから送信された CRC ステータスを保持します。

SD カードヘデータを送信したあとは、必ず SDSBR レジスタのビジィ・フラグと CRC ステータス・ビット

をソフトウェアでチェックし、ビジィ・フラグが1の場合は、SDカードに対して次のアクション（コマンド送信，データ送信）を行わないでください。

表 5-43 SDCMD_IDX のビット構成

ビット	名称	機能	ロード/ ストア
15	予約	<ul style="list-style-type: none"> ・0 以外の書き込みは行わないこと。 ・読み出し時は不定。 	-
14-11	DBL	データ・ブロック・レンジ設定ビット <ul style="list-style-type: none"> ・0000 : 1 バイト ・0001 : 2 バイト ・0010 : 4 バイト ・0011 : 8 バイト ・0100 : 16 バイト ・0101 : 32 バイト ・0110 : 64 バイト ・0111 : 128 バイト ・1000 : 256 バイト ・1001 : 512 バイト (デフォルト) ・1010 : 1024 バイト ・1011 : 2048 バイト 上記以外はリザーブ (設定禁止)	L/S
10	RES	レスポンス設定ビット SDカードからのレスポンスの有無を設定する。 <ul style="list-style-type: none"> ・0 : レスポンスなし (デフォルト) ノー・レスポンスのオペレーションとして取り扱う ・1 : レスポンスあり 	L/S
9, 8	RW	リード/ライト設定ビット データのリードあるいはライトおよびデータ転送停止を設定する。 <ul style="list-style-type: none"> ・00 : リード/ライトなし (デフォルト) ノー・データのオペレーションとして取り扱う ・01 : リード・オペレーション ・10 : ライト・オペレーション ・11 : リード/ライトなし 	L/S
7	予約	<ul style="list-style-type: none"> ・0 以外の書き込みは行わないこと。 ・読み出し時は不定。 	-
6	BSL	レスポンス・ビット数設定ビット <ul style="list-style-type: none"> ・0 : 48 ビット (デフォルト) ・1 : 136 ビット 	L/S
5-0	IDX	SDカードへ送信するコマンド・インデクスを設定するビット コマンドを送信時に、コマンドのビット 45-40 に割り当てる。	L/S

表 5-44 SDCTL のビット構成 (1/2)

ビット	名称	機能	ロード/ ストア
15	ALLE	SDCIF 内のすべてのエラー状態を示すフラグ CRRE, CRDE, SDSE, SDLE のいずれかのエラー・フラグが立ったときに 1 にセットされる。 ・0: エラーなし (デフォルト) ・1: エラーあり クリアするには 0 をセットする。	L/S
14	CEN	コマンド実行イネーブル SDCMD レジスタに設定したコマンドを SD カードへ送信するためのビット。1 を設定することにより, SDCMD レジスタに設定しているコマンドを SD カードへ送信し, SD カードはこのコマンドを受信して制御する。コマンド送信終了後, 自動的に CEN は 0 にクリアする。 ・0: コマンド送信以外の状態 (デフォルト) ・1: コマンド送信中の状態	L/S
13-10	DIV	SD カード・クロック分周 システム・クロックから SD カード用のクロックを生成するための分周率を設定する。 ・0000:設定不可 ・0001:設定不可 ・0010:設定不可 ・0011:1/4 (PMT 可) ・0100:1/5 (PMT 可) ・0101:1/7 (PMT 可) ・0110:1/8 (PMT 可) ・0111:1/9 ・1000:1/10 ・1001:1/12 ・1010:1/16 ・1011:1/32 ・1100:1/64 ・1101:1/128 ・1110:1/256 ・1111:1/512	L/S
9	SDSTP	SD カード・インタフェース・ストップ・ビット SDCIF を使用しない場合, または SD カード・アクセスを途中でストップしたい場合に 1 を設定する。SDSTP ビットが 1 になると SDCLK がロウ・レベル固定になり, SD カードのアクセスを停止する。このとき, SDCIF 内は状態を保持する。 ・0: SD カード・インタフェース動作 ・1: SD カード・インタフェース停止 (デフォルト) SDCLK 発振開始時は, 2×SDCLK サイクル以上の安定時間を確保してからアクセスを行うこと。	L/S
8	SDRST	SD カード・インタフェース・リセット許可ビット 1 に設定することで, SDCIF 内 (レジスタ値など) が初期化される。初期化後, 自動的に 0 にクリアされる。 ・0: (デフォルト) ・1: 初期化	L/S
7	予約	・0 以外の書き込みは行わないこと。 ・読み出し時は不定。	L/S
6	CRRE	CRC レスポンス・エラー・フラグ SD カードから受信したレスポンス内の CRC と μ PD77213 内で計算した CRC を比較し, 不一致の場合に CRRE をセットする。 ・0: エラーなし (デフォルト) ・1: エラーあり クリアするには 0 を設定する。	L/S

表 5-44 SDCTL のビット構成 (2/2)

ビット	名称	機能	ロード/ ストア
5	CRDE	CRC データ・エラー・フラグ SD カードから受信したデータ内の CRC と μ PD77213 内で計算した CRC を比較し、不一致の場合に CRDE をセットする。 ・0: エラーなし (デフォルト) ・1: エラーあり クリアするには 0 を設定する。	L/S
4	SDSE	SDDR ストア・エラー・フラグ SDS フラグが 0 のときに SDDOR ヘデータをストアした場合、1 をセットする。 ・0: エラーなし (デフォルト) ・1: エラーあり クリアするには 0 を設定する	L/S
3	SDLE	SDDR ロード・エラー・フラグ SDL フラグが 0 のときに SDDIR からデータをロードした場合、1 をセットする。 ・0: エラーなし (デフォルト) ・1: エラーあり クリアするには 0 を設定する	L/S
2	SDRL	SDRPR ロード許可フラグ SDRPR は、16 ビット \times 8 のレジスタで構成している。 ・0: SDRPR7-0 にデータが入っていないことを示す (デフォルト)、 プログラムから SDRPR0 をロードしたときに 0 になる。 ・1: SDRPR7-0 にデータが入っていることを示す。 SDCSR から SDRPR へすべてのレスポンスが転送されたとき 1 になる。	L/S
1	SDS	SDDR ストア許可フラグ ・0: SDDR (out) にデータが入っていることを示す。 プログラムから SDDR にストアしたときに 0 になる。 ・1: SDDR (out) にデータが入っていないことを示す (デフォルト)、 SDDR から SDDSR ヘデータが転送されたとき 1 になる。	L/S
0	SDL	SDDR ロード許可フラグ ・0: SDDR (in) にデータが入っていないことを示す (デフォルト)、 プログラムから SDDR をロードしたときに 0 になる。 ・1: SDDR (in) にデータが入っていることを示す。 SDDSR から SDDR ヘデータが転送されたとき 1 になる。	L/S

表 5-45 SDRPR レジスタの構成

レジスタ名	機能
SDRPR7	SD カードから受信したレスポンスのビット 135-0 のうち、ビット 127-112 のレスポンスを保持します。 SDCSR に入ったレスポンスを最初に保持し、すべての SDRPR に有効なレスポンスが保持されたあとの最初のロードで MIO バス経由で出力できます。 SDCMD_IDX レジスタの BSL ビットが 1 の場合に有効です。
SDRPR6	SD カードから受信したレスポンスのビット 135-0 のうち、ビット 111-96 のレスポンスを保持します。 SDCSR に入ったレスポンスを SDRPR7 の次に保持し、SDRPR7 をロードした次にロードを行なうと SDRPR6 のレスポンスを MIO バス経由で出力できます。 SDCMD_IDX レジスタの BSL ビットが 1 の場合に有効です。
SDRPR5	SD カードから受信したレスポンスのビット 135-0 のうち、ビット 95-80 のレスポンスを保持します。 SDCSR に入ったレスポンスを SDRPR6 の次に保持し、SDRPR6 をロードした次にロードを行なうと SDRPR5 のレスポンスを MIO バス経由で出力できます。 SDCMD_IDX レジスタの BSL ビットが 1 の場合に有効です。
SDRPR4	SD カードから受信したレスポンスのビット 135-0 のうち、ビット 79-64 のレスポンスを保持します。 SDCSR に入ったレスポンスを SDRPR5 の次に保持し、SDRPR5 をロードした次にロードを行なうと SDRPR4 のレスポンスを MIO バス経由で出力できます。 SDCMD_IDX レジスタの BSL ビットが 1 の場合に有効です。
SDRPR3	SD カードから受信したレスポンスのビット 135-0 のうち、ビット 63-48 のレスポンスを保持します。 SDCSR に入ったレスポンスを SDRPR4 の次に保持し、SDRPR4 をロードした次にロードを行なうと SDRPR3 のレスポンスを MIO バス経由で出力できます。 SDCMD_IDX レジスタの BSL ビットが 1 の場合に有効です。
SDRPR2	SD カードから受信したレスポンスのビット 135-0 または 47-0 のうち、ビット 47-32 のレスポンスを保持します。 SDCMD_IDX レジスタの BSL ビットが 1 の場合、SDCSR に入ったレスポンスを SDRPR3 の次に保持し、SDRPR3 をロードした次にロードを行なうと SDRPR2 のレスポンスを MIO バス経由で出力できます。 SDCMD_IDX レジスタの BSL ビットが 0 の場合、SDCSR に入ったレスポンスを最初に保持し、SDRPR2-0 に有効なレスポンスが保持されたあとの最初のロードで MIO バス経由で出力できます。
SDRPR1	SD カードから受信したレスポンスのビット 135-0 または 47-0 のうち、ビット 31-16 のレスポンスを保持します。 SDCSR に入ったレスポンスを SDRPR2 の次に保持し、SDRPR2 をロードした次にロードを行なうと SDRPR1 のレスポンスを MIO バス経由で出力できます。
SDRPR0	SD カードから受信したレスポンスのビット 135-0 または 47-0 のうち、ビット 15-0 のレスポンスを保持します。 SDCSR に入ったレスポンスを SDRPR1 の次に保持し、SDRPR1 をロードした次にロードを行なうと SDRPR0 のレスポンスを MIO バス経由で出力できます。 SDRPR0 にレスポンスが保持された時点で SDCTL の SDRL ビットが 1 に設定され、レスポンスが SDRPR7-0 に入ったことを示します。また、SDRPR0 をロードした時点で SDCTL の SDRL ビットが 0 に設定され、レスポンスが SDRPR7-0 に入っていないことを示します。

5. 18. 3 CRC (Cyclic Redundancy Codes) 回路

CRC 回路は、 μ PD77213 と SD カード間のデータ・エラーを検出します。

SD カード・インタフェースから送信するコマンドやライト・データに対して、 μ PD77213 内で計算した CRC を付加して出力します。

SD カードから受信したレスポンスやリード・データに付加される CRC データと μ PD77213 内で計算したレスポンスやリード・データの CRC データと比較して、不一致ならエラー・フラグを立てます。

5. 18. 4 SD カード・インタフェースの動作

SD カードに対するアクセスは MIO バスから外部メモリ空間上にマッピングされたレジスタに対して行うことで実現します。アクセスには、ステータスによるポーリング、割り込みを用いることができます。ウェイトによるアクセスはできません。

割り込みは次の 4 つの要因で行います。

- レスポンス・ロード要求：割り込み番号 4 (エントリ：0x220) の要因 3
- コマンド・ストア要求 (ビジィ解除)^注：割り込み番号 5 (エントリ：0x224) の要因 3
- データ・ロード要求：割り込み番号 6 (エントリ：0x228) の要因 3
- データ・ストア要求：割り込み番号 7 (エントリ：0x22C) の要因 3

注 コマンド・ストア要求はビジィ解除を示すものであるため、ビジィを介さない場合は割り込み要求されません。

APCR レジスタは Peripheral C および Peripheral D を SDCIF の制御として使用します。Peripheral A および Peripheral B は外部データ・メモリ・アクセスでの PMT 要求制御に使用します。初期値は 0x0000 です。

表 5-46 μ PD77213 の APCR レジスタのビット構成 (1/2)

ビット	名称		機能	ロード/ ストア
15	Peripheral A (MIO)	SRE	ストア要求許可フラグ 外部データ・メモリ・インタフェースからのデータ・ストア要求イネーブル。 0：ディスエーブル (スストア要求をマスク) 1：イネーブル (スストア要求を透過)	L/S
14		LRE	ロード要求許可フラグ 外部データ・メモリ・インタフェースからのデータ・ロード要求イネーブル。 0：ディスエーブル (ロード要求をマスク) 1：イネーブル (ロード要求を透過)	L/S
13		PFE	未使用 1 に設定。	L/S
12		TOE	未使用 0 に設定。	L/S
11	Peripheral B (MIO)	SRE	ストア要求許可フラグ 外部データ・メモリ・インタフェースからのデータ・ストア要求イネーブル。 0：ディスエーブル (スストア要求をマスク) 1：イネーブル (スストア要求を透過)	L/S
10		LRE	ロード要求許可フラグ 外部データ・メモリ・インタフェースからのデータ・ロード要求イネーブル。 0：ディスエーブル (ロード要求をマスク) 1：イネーブル (ロード要求を透過)	L/S

表 5-46 μ PD77213 の APCR レジスタのビット構成 (2/2)

ビット	名称	機能	ロード/ ストア	
9	Peripheral B (MIO)	PFE	未使用 1に設定。	L/S
8		TOE	未使用 0に設定。	L/S
7	Peripheral C (SDCIF)	SRE	データ・ストア要求割り込み許可フラグ SDCIFからのデータ・ストア要求割り込み信号イネーブル。 0: ディスエーブル(割り込み要求をマスク) 1: イネーブル(割り込み要求を透過)	L/S
6		LRE	データ・ロード要求割り込み許可フラグ SDCIFからのデータ・ロード要求割り込み信号イネーブル。 0: ディスエーブル(割り込み要求をマスク) 1: イネーブル(割り込み要求を透過)	L/S
5		PFE	未使用 1に設定。	L/S
4		TOE	未使用 0に設定。	L/S
3	Peripheral D (SDCIF)	SRE	ビジィ解除割り込み許可フラグ SDCIFからのビジィ解除割り込み信号イネーブル。 0: ディスエーブル(割り込み要求をマスク) 1: イネーブル(割り込み要求を透過)	L/S
2		LRE	レスポンス・ロード要求割り込み許可フラグ SDCIFからのレスポンス・ロード要求割り込み信号イネーブル。 0: ディスエーブル(割り込み要求をマスク) 1: イネーブル(割り込み要求を透過)	L/S
1		PFE	SDカード・インタフェース機能許可フラグ SDCIF機能の動作と停止を設定する。ディスエーブルではSDCIFにク ロックを供給しない。 0: イネーブル 1: ディスエーブル(兼用端子を外部データ・メモリ・アドレスとする)	L/S
0		TOE	SDカード・インタフェース端子出力許可フラグ リセット・アクティブ時のSDCIFの出力端子の状態を設定する。 0: ディスエーブル(ハイ・インピーダンスとする) 1: イネーブル(SDCIF信号とする)	L/S

注意 LRE, SRE ビットは PMT 転送要求に対する設定であり、割り込み制御回路に対するマスク制御ではありません。
PMT 転送を行う場合は、LRE, SRE をセットします。ただし、4 ブロックを排他的に設定する必要があります。
Peripheral A で“1”に設定したら、Peripheral B, C, D では“1”に設定しないでください。

★ 5. 18. 5 SD Card IF SD カード使用時の注意事項(μ PD77213 のみ)

プログラム・シーケンスにおけるビジィ信号の認識方法として、DSP 内の SDSBR レジスタ内のビジィ・フラグに基づいて判定した場合、このフラグで示されるビジィ信号は、SD カードへのデータ・ライト後に発生するビジィ信号^{※1}のみです。このフラグをマルチライト終了時の STOP コマンド (CMD12) 後の programming mode のビジィ信号^{※2}の検出にも使用した場合、ソフトウェア側で busy 信号が検出できなくなり、ソフトウェアの仕様によっては次のコマンドを発行できない状況が発生するので、ソフト側で注意が必要となります。

注1. SD Memory Card Specification Part1 V1.01 P.58 Fig.28 , Fig.29 を参照

2. SD Memory Card Specification Part1 V1.01 P.59,60 の Fig.30-33 を参照

第6章 ブートアップ機能

この章では、 μ PD77210 ファミリのブートアップ機能について説明します。

μ PD77210 ファミリは割り込みベクタをはじめとして、命令メモリの 0x0200 番地以降が RAM として構成されています。この命令 RAM にアプリケーション・プログラムをロードする必要があります。

命令メモリ空間には内部命令 RAM に命令をブートアップするためのブートアップ・ルーチンが ROM として格納されています。 μ PD77210 ファミリはシステム・リセット解除後ブートアップ処理を行い、ブートアップ処理終了後 0x0200 番地より命令を実行します。

ブートアップは、命令コードのソースとして次の種類が選択できます。

- X データ・メモリ・ブート
- Y データ・メモリ・ブート
- XY データ・メモリ・ブート
- 外部データ・メモリ・ブート
- ホスト・ブート
- シリアル・ブート
- ノン・ブート (ブート処理せず 0x0200 ヘジャンプ)

ブートアップ・ルーチンは、ユーザがアプリケーション・プログラムにてコールすることも可能です。イニシャル・リセット・ブート (リセット後の初回ブート) 後、再度命令を書き換えるリブート機能をサポートしています。

6.1 イニシャル・リセット・ブート

6.1.1 ブート・モード指定

リセット解除後の汎用ポート (P0-P2 端子) の設定値によりブート・モードを決定します。汎用ポートの設定とブート・モードの関係を表 6-1 に示します。また、リセット解除後の P3 端子の設定値により PLL のロック・レンジを指定します。ブート・モードと合わせて設定が必要です。PLL のロック・レンジの設定を表 6-2 に示します。

イニシャル・リセット・ブートでは、転送先のページ (DPR=0x80) が固定であるため、0x8000-0xFFFF の命令メモリのブートアップはリブート・ルーチンにて行ってください。

最大ブート・ワード数は、ブート・モードによって異なり、0x4000 (16 K ワード) - 1 ~ 0x7D00 (31.5 K ワード) です。イニシャル・リセット・ブート処理により、DPR は 0x00 に書き換えられます。

表 6-1 イニシャル・リセット・ブート・モード

P2	P1	P0	ブート・モード
0	0	0	ノン・ブート
0	0	1	Xメモリ・ブート
0	1	0	Yメモリ・ブート
0	1	1	XYメモリ・ブート
1	0	0	外部データ・メモリ・ブート
1	0	1	ホスト・ブート
1	1	0	シリアル・ブート
1	1	1	設定禁止

表 6-2 PLL のロック・レンジ

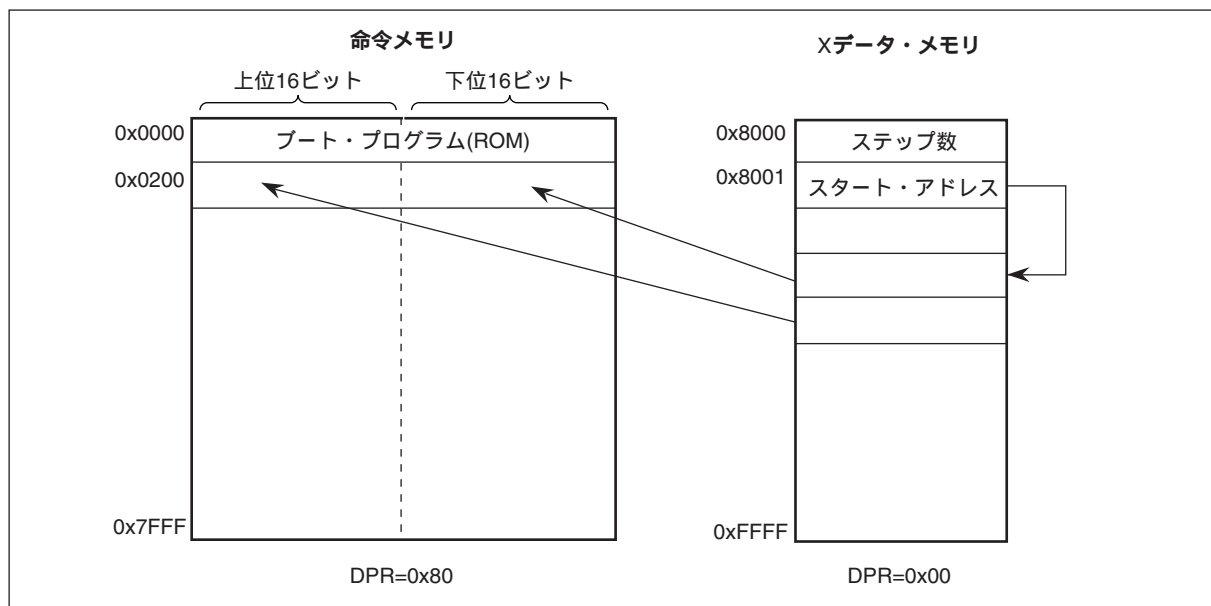
P3	ロック・レンジ
0	120-160 MHz
1	80-120 MHz

注意 ブートアップ時の汎用ポート設定は、システム・リセット解除の最低 3 サイクル以前から、ブート・プログラム動作後の 0x200 番地の命令実行サイクルまでを、固定としてください。

6. 1. 2 Xメモリ・ブート

内部 X データ・メモリの 0x8000 番地 (DPR=0x0) から命令コードをブートするモードです。ブートする命令ステップ数, ソース先頭アドレスがパラメータとして必要です。最大ブート・ワード数は 0x4000 (16 K ワード) - 1 です。

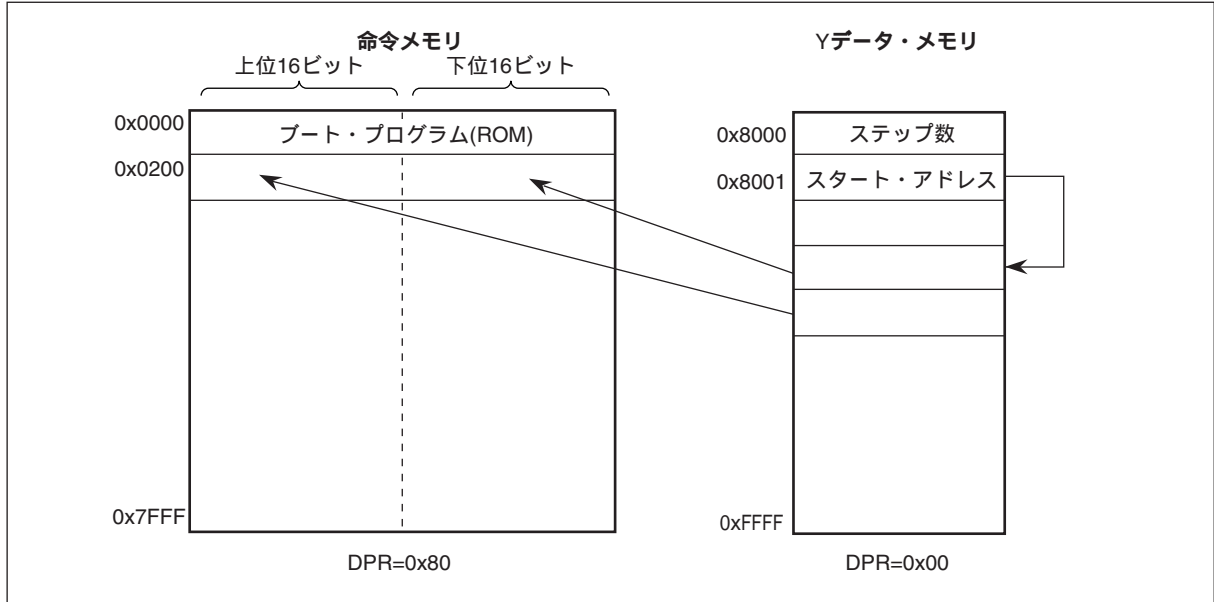
図 6-1 Xメモリからのブートアップ



6.1.3 Yメモリ・ブート

内部Yデータ・メモリの0x8000番地(DPR=0x0)から命令コードをブートするモードです。ブートする命令ステップ数, ソース先頭アドレスがパラメータとして必要です。最大ブート・ワード数は0x4000(16Kワード)-1です。

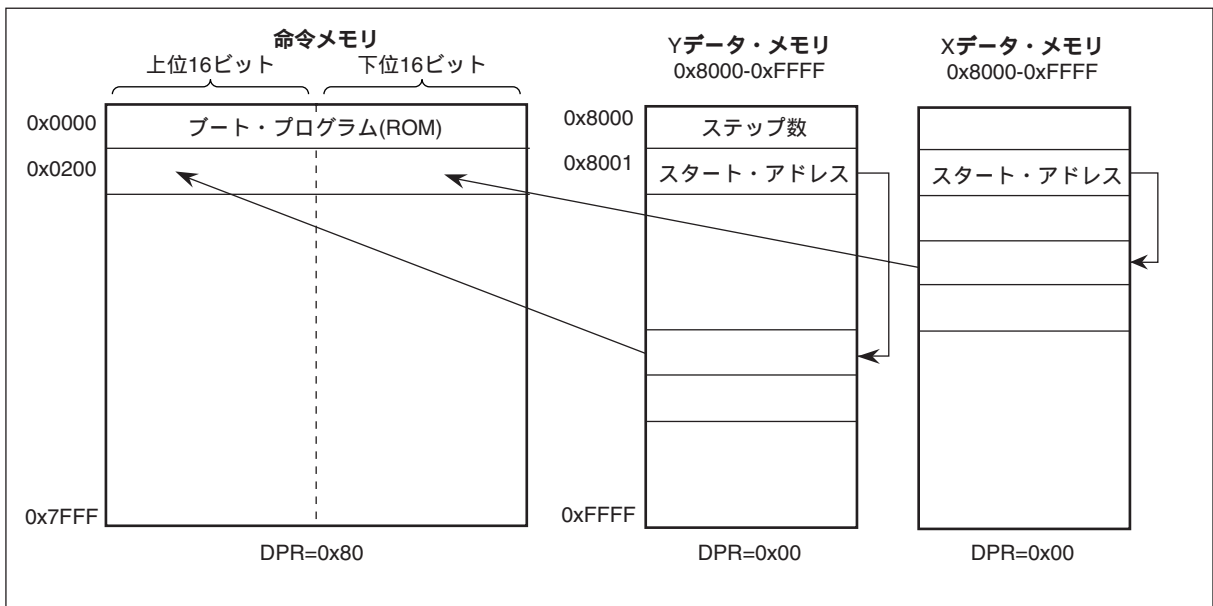
図 6-2 Yメモリからのブートアップ



6.1.4 XYメモリ・ブート

内部XYの両データ・メモリの0x8000番地(DPR=0x0)から命令コードをブートするモードです。命令コード上位16ビットをYメモリから, 下位16ビットをXメモリからブートします。最大ブート・ワード数は0x7D00(31.5Kワード)です。

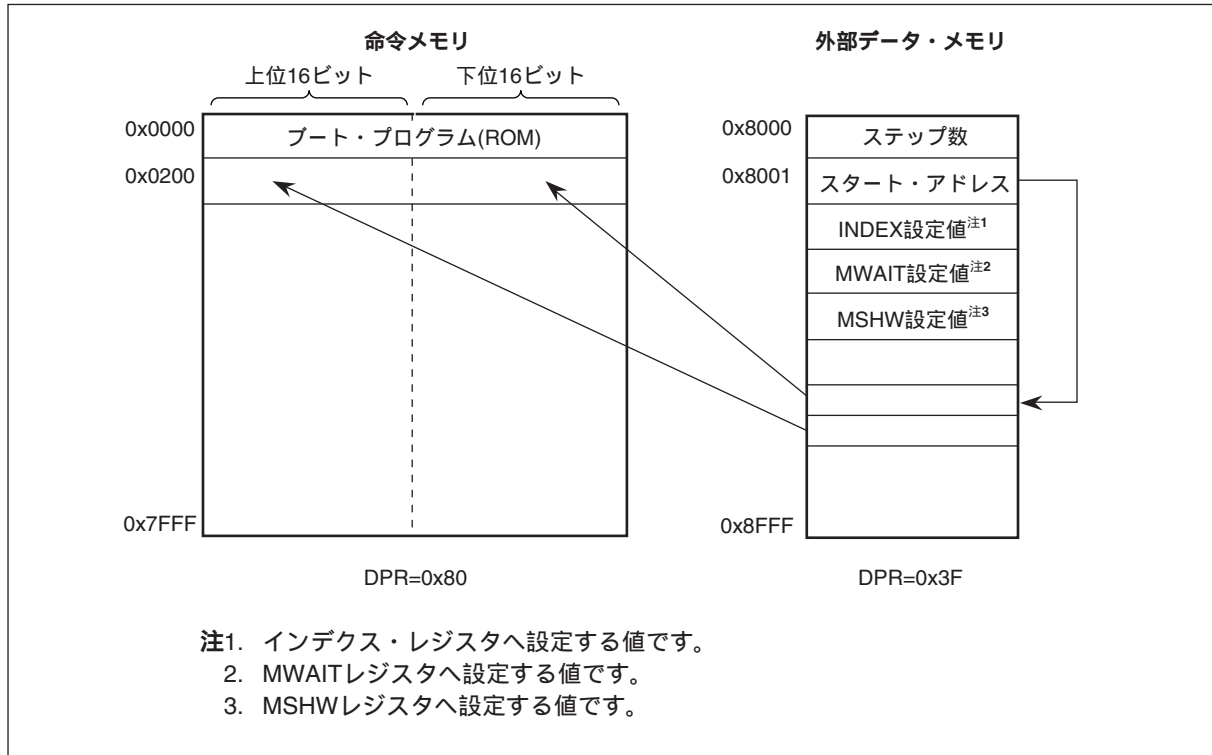
図 6-3 XメモリおよびYメモリからの並列ブートアップ



6.1.5 外部データ・メモリ・ブート

外部データ・メモリから命令コードをブートするモードです。ブートする命令ステップ数，ソース先頭アドレスのほか，ウェイト数などがパラメータとして必要です。最大ブート・ワード数は 0x4000 (16 Kワード) です。

図 6-4 外部データ・メモリからのブートアップ



外部メモリのパラメータ設定は， μ PD77210 ファミリから X 側ダイレクト・アクセスでブートアップを行います。次に例を示します。

例 1. 外部メモリ・アドレス (20 ビット空間) の 0x00005 から命令コードを配置する場合

source start address : 0x8005
index data : 0x0000

例 2. 外部メモリ・アドレス (20 ビット空間) の 0x10005 から命令コードを配置する場合

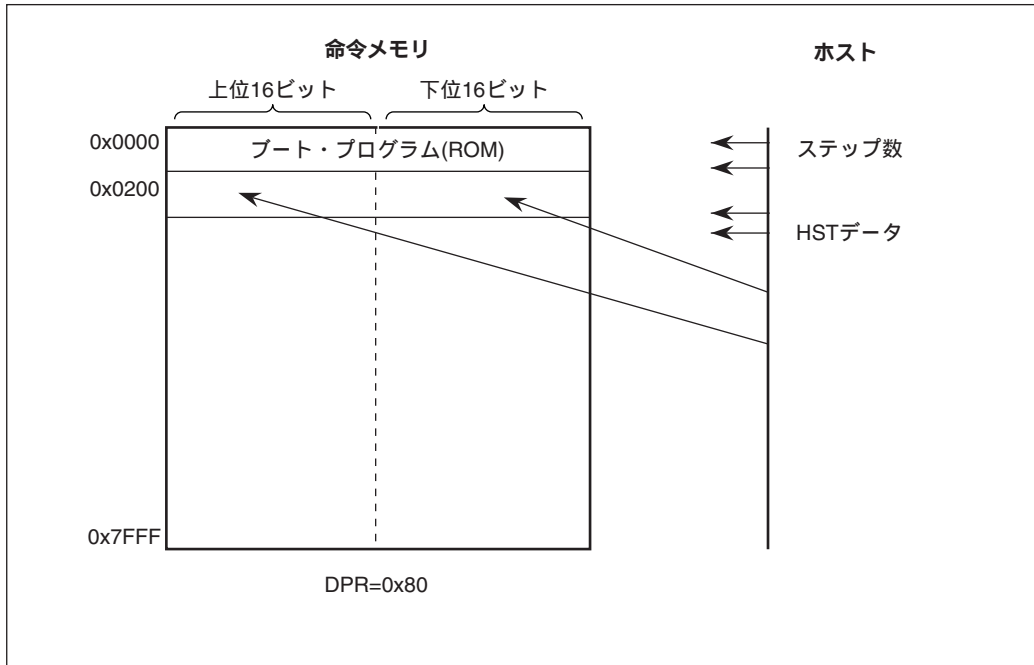
source start address : 0x8005
index data : 0x0010

注意 μ PD77213 の場合，メモリ・アドレス端子が SD カード・インタフェースと兼用になっているため，リセット直後の外部空間が 8 K ワードになります。したがってブート・ソースもこの空間に制限されます。

6.1.6 ホスト・ブート

ホスト・インタフェースを経由してホスト・デバイスより命令コードをブートするモードです。ブートする命令ステップ数，HST（ホスト・ステータス・レジスタ）がパラメータとして必要です。ホスト・インタフェースとのハンドシェークはウエイト動作で行います。最大ブート・ワード数は 0x7D00（31.5 Kワード）です。

図 6-5 ホスト・ブートアップ



注意 ホスト・インタフェースは、リセット解除後に 8 ビット・バス・モードとなっています。最初の 2 ワードのパラメータ（ステップ数，HST データ）は、8 ビット・バス・モードを用いてホスト・バスの下位 8 ビット・バスで、パラメータの下位 8 ビット，上位 8 ビットの順で転送してください。

ホスト・ブートを開始する場合は、ブート・プログラムは HST を 0x400 に設定します（ウエイト・モード， $\overline{\text{HWE}}$ 端子出力許可）。HST の設定値を転送後は、転送した値に書き換わりませんが、ウエイト・モードの指定を変更しないでください。

(1) ホスト・インタフェースの設定

ホスト・ブートに先立ち、ホスト・インタフェースは次のように設定されます。ただし、HAWE を除いてこれらの設定は、送られる HST 設定パラメータによって上書きされます。

HST = 0x0401

HAWE =1 : ウエイトを使用する。

HREM =0 : HRE をマスクしない。

HWEM =0 : HWE をマスクしない。

注意 HST レジスタの値は、HST 設定パラメータがブート・プロセス中で設定されたときに書き換えられます。

(2) ホスト・ブートのパラメータ

リセット時ホスト・ブートのパラメータを次に示します。

- ブート命令数 : ブートするプログラムの命令ステップ数 (ブート対象となる命令数) を表します。実際にデータとして転送されるデータ数は、命令ステップ数の2倍になります。
- HST 設定値 : HST に設定するデータです。HAWE (ビット 10) を除く、そのほかのビットを設定します。HAWE は設定値にかかわらず “1” です。
- 命令コード : 命令コード 32 ビットを、下位 16 ビット (ビット 15-0) を最初に、上位 16 ビット (ビット 31-16) を後からの順で転送します。したがって、8 ビット・バス・モードのときは、ビット 7-0, 15-8, 23-16, 31-24 のバイト順に転送することになります。また、16 ビット・バス・モードのときは、ビット 15-0, 31-16 のワード順に転送することになります。

これらのパラメータを、ホストから見た転送順序に従って並べると次のようになります。

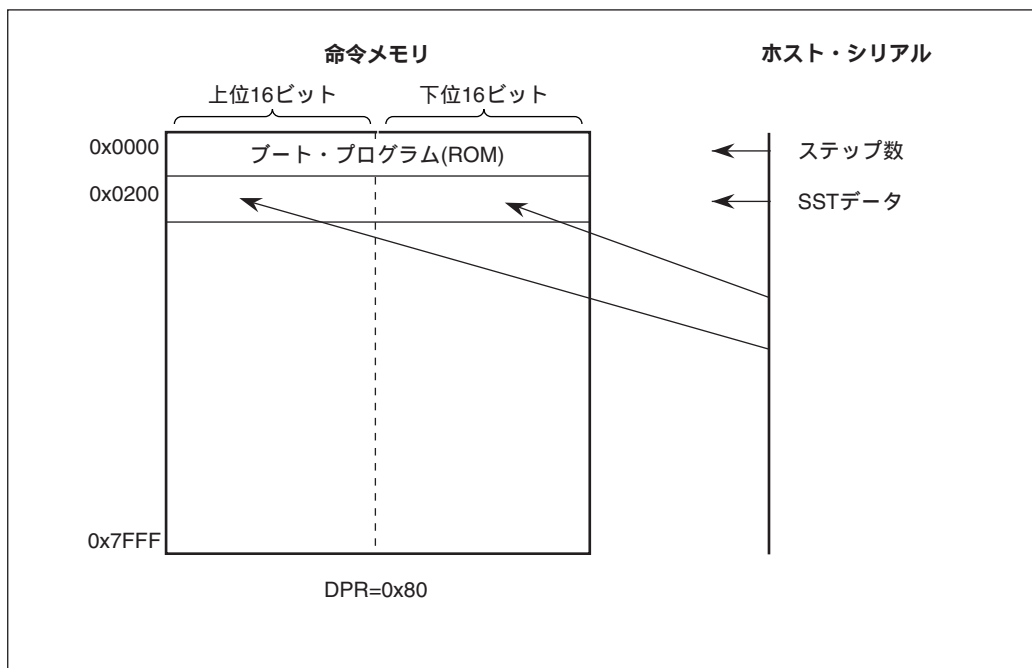
順序	8 ビット・バス・モード	16 ビット・バス・モード	
	HD7-HD0	HD15-HD8	HD7-HD0
1	ステップ数の下位 8 ビット	ダミー	ステップ数の下位 8 ビット
	ステップ数の上位 8 ビット	μ PD77210 ファミリが HDT(in) からデータをロードする間ウエイト	
	μ PD77210 ファミリが HDT(in) からデータをロードする間ウエイト	ダミー	ステップ数の上位 8 ビット
2	HST の下位 8 ビット	ダミー	HST の下位 8 ビット [※]
	HST の上位 8 ビット	μ PD77210 ファミリが HDT(in) からデータをロードする間ウエイト	
	μ PD77210 ファミリが HDT(in) からデータをロードする間ウエイト	ダミー	HST の上位 8 ビット [※]
3	1 番目の命令コード (ビット 7-0)	1 番目の命令コード (ビット 15-8)	1 番目の命令コード (ビット 7-0)
	1 番目の命令コード (ビット 15-8)	μ PD77210 ファミリが HDT(in) からデータをロードする間ウエイト	
	μ PD77210 ファミリが HDT(in) からデータをロードする間ウエイト	1 番目の命令コード (ビット 31-24)	1 番目の命令コード (ビット 23-16)
4	1 番目の命令コード (ビット 31-24)	2 番目の命令コード (ビット 15-8)	2 番目の命令コード (ビット 7-0)
	1 番目の命令コード (ビット 23-16)	μ PD77210 ファミリが HDT(in) からデータをロードする間ウエイト	
	μ PD77210 ファミリが HDT(in) からデータをロードする間ウエイト	2 番目の命令コード (ビット 31-24)	2 番目の命令コード (ビット 23-16)
5	2 番目の命令コード (ビット 7-0)	3 番目の命令コード (ビット 15-8)	3 番目の命令コード (ビット 7-0)
	2 番目の命令コード (ビット 15-8)	μ PD77210 ファミリが HDT(in) からデータをロードする間ウエイト	
	μ PD77210 ファミリが HDT(in) からデータをロードする間ウエイト	3 番目の命令コード (ビット 31-24)	3 番目の命令コード (ビット 23-16)
	⋮		

注 16 ビット・バス・モードを指定します。

6.1.7 シリアル・ブート

時分割シリアル・インタフェースを経由したシリアル・デバイスより命令コードをブートするモードです。ブートする命令ステップ数, SST (シリアル・ステータス・レジスタ) がパラメータとして必要です。シリアル・インタフェースとのハンドシェークはウエイト動作で行います。最大ブート・ワード数は 0x7D00 (31.5 Kワード) です。

図 6-6 シリアル・ブートアップ



リセット解除後, SIAK 信号がアクティブとなったあと, 順次パラメータおよびプログラム・データを TSI 経由で入力します。

初期でのシリアル・ステータス・レジスタ (SST) は, 0x0EC0 であり, MSB ファースト, 16 ビット, アクセス・ウエイト許可, シリアル入力連続モードになっています。

基本的に, SST としてのパラメータは 0x0EC0 固定です。この設定はリブート時も同様です。

6.1.8 ノン・ブート

リセット解除後にブートアップを行わず, 0x200 番地からの命令を実行します。μ PD77210 ファミリは 0x200 番地からの領域が RAM であるため, 電源投入後はこの領域が不定になります。したがって, 電源投入時のブート・モードとしてこのモードを指定することができません。

一度ブートアップを行ったあとにリセットを入力する場合, このモードを設定することによってブートアップを省略することができます。

ただし, スタンバイ・モード (ホールドおよびストップ) からの復帰の場合は, ノン・ブートは使用できません。これは, リセット時にクロック回路が分周器出力から CLKIN 端子入力クロック・ダイレクトへと切り替わると, システム・クロックにハザードが発生する可能性があり, その結果, 内蔵 RAM およびレジスタの内容を保証できないからです。

6.2 イニシャル・リセット・ブートとPLL

イニシャル・リセット・ブートによってPLLの動作が異なります。

リセットによって強制的に外部入力クロックによる動作になります。ブート・モードによってブート・ルーチン中でPLLが起動されます。ブート・モードとPLLの起動の関係を表6-3に示します。

PLLが起動しないブート・モードの場合、ブートは外部入力クロックによって行われます。PLLを使用する場合、ブート後、ユーザ・プログラム中でPLLの起動とクロック・ソースの切り替えが必要になります。

PLLが起動しクロック・ソースをPLLの出力に切り替えるブート・モードの場合、ブートはPLLの出力クロックで行われます。

いずれの場合もブート・ルーチン中で、分周器の起動が行われます(分周器側への切り替えは行いません。また、分周率は1/1で設定します)。

表 6-3 ブート・モードとPLL

ブート・モード	PLL
ノン・ブート	起動しません。
Xメモリ・ブート	起動,およびクロック・ソースとしてPLLを選択します。
Yメモリ・ブート	起動,およびクロック・ソースとしてPLLを選択します。
XYメモリ・ブート	起動,およびクロック・ソースとしてPLLを選択します。
外部データ・メモリ・ブート	起動しません。
ホスト・ブート	起動,およびクロック・ソースとしてPLLを選択します。
シリアル・ブート	起動,およびクロック・ソースとしてPLLを選択します。

6.3 リポート

ユーザのアプリケーション・プログラムからブート ROM 中のリポート・ルーチンをコールすることにより、新たに命令 RAM に命令データを設定することが可能です。

表 6-4に各リポート・エントリ・アドレスおよび設定パラメータを示します。

リポートでは、転送元、転送先のページを指定することが可能です。リポートでは転送対象として命令とデータの区別を行いません。データ・ページ・レジスタ(DPR)を 0x00-0x3F にすることにより、データ・メモリへのブートも可能です。ただし、データ・メモリをブートする場合も、命令ブートと同じルーチンを使用するため、32 ビット単位のステップ数管理であり、また、X、Y の一律ブートとなります。最大ブート・ワード数はブート・モードによって異なり、0x4000 (16 K ワード) - 1-0x7D00 (31.5 K ワード) であり、ページ単位で行います。

表 6-4 リポート・エントリ・アドレスとパラメータ

リポート・モード	エントリ・アドレス	パラメータ					
		命令ステップ数	転送元開始アドレス	転送元	転送先開始アドレス	転送先ページ(DPR)	その他の使用レジスタ
Xリポート	0x1	R7L	DP3	R6L ^{注1}	DP2	R5L	DP6
Yリポート	0x2	R7L	DP7	R6L ^{注1}	DP6	R5L	DP2
XYリポート	0x3	R7L	DP3, DP7	R6L ^{注1}	DP2	R5L	DP6
外部リポート	0x4	R7L	DP3	R6L ^{注2}	DP2	R5L	DP6
ホスト・リポート	0x5	R7L	-	R6L ^{注3}	DP2	R5L	DP6
シリアル・リポート	0x6	R7L	-	R6L ^{注4}	DP2	R5L	DP6

注1. page : 転送元のデータ・ページを指定します。

2. index : 転送元のインデクス・レジスタを指定します。

3. hst : HST の設定値を指定します。HAWE は 1 としてください。

4. sst : SST の設定値を指定します。HAWE は 0xEC0 としてください。

リポートを行う場合の注意点は次のとおりです。

注意1. リポート処理では、使用するレジスタの値が破壊され、プログラム・スタックを 1 レベル(エントリ時)、ループ・スタックを 1 レベル使用します。必要なレジスタは、退避/復帰を行ってください。リポートは、すべての割り込みを禁止した状態で行ってください。

2. イニシャル・リセット・ブートでは、転送先のページ(DPR = 0x80)が固定であるため、0x8000-0xFFFF の命令メモリのブートアップはリポート・ルーチンにて行ってください。

3. シリアル・リポート時の SST パラメータは 0x0EC0 にしてください。

4. 命令 0x0200-0x7FFF へのブートでは、DPR を 0x80 とし、destination は、0x8200-0xFFFF とし 0x8000 のオフセットが必要となります。

5. リポート処理により DPR は、0x00 に書き換えられます。

★ 6. 内部命令 RAM へのリポート・クラスを作成したときに生成される転送先開始アドレスが 0x8000 番地にオフセットされたアドレスにはなりません。ブート・クラスの場合はオフセットされたアドレスが生成されず。内部命令 RAM へのリポート・クラスを作成したときに生成される転送先開始アドレスに、0x8000 番地のオフセットをユーザ側で設定してください。

付録 A 索引

A.1 五十音で始まる語句の索引

[あ]

アーキテクチャ ... 43
アドレッシング・モード ... 96
アドレス ALU ... 97
イニシャル・リセット・ブート ... 241, 248
インサーキット・エミュレータ機能 ... 227
ウエイト ... 158, 170
演算ユニット ... 107
オーダ情報 ... 23
オーディオ・シリアル・インタフェース ... 136
オーバフロー・エラー・フラグ ... 89

[か]

外部データ・メモリ・インタフェース ... 92, 172
外部データ・メモリ・ブート ... 244
外部データ・メモリ・マップ ... 95
外部データ・メモリ容量 ... 92
外部割り込みのハードウェア条件 ... 204
拡張インタフェース ... 228
間接アドレッシング ... 98
クロック・コントローラ ... 212
クロック・ジェネレータ ... 51
クロック・モニタ出力 ... 219
クロックの切り替えタイミング ... 215
クロック制御の注意点 ... 216
固定小数点フォーマット ... 110

[さ]

算術演算 ... 118
システム制御ユニット ... 50
使用方法概要 ... 39
シリアル・ブート ... 247
乗算機能 ... 112
時分割 (TDM) シリアル・インタフェース ... 127
数値フォーマット ... 110
整数フォーマット ... 111

スタック・エラー・フラグ ... 89
スタンダード・シリアル・インタフェース ... 145
スタンバイ・モード時のクロック ... 219
スタンバイ機能 ... 58
スタンプ・モード ... 208
スロット ... 134
スロット拡張 ... 134
全体ブロック構成 ... 43
ソフトウェア・ループ・スタック ... 75

[た]

タイマ ... 208
ダイレクト・アクセス ... 180
多重割り込み ... 82
端子機能 ... 24, 30
端子構成 ... 29
端子接続 ... 25
直接アドレッシング ... 98
ディバグ・インタフェース ... 224
ディバグ機能 ... 227
データ・アドレッシング・ユニット ... 90
データ・バス ... 47
データ・ポインタのモディファイー一覧 ... 101
データ・メモリ空間 ... 90
同時アクセスの制限 ... 93
特 徴 ... 20

[な]

内部データ・メモリ ... 92
内部命令メモリ ... 61
ノー・チェンジ ... 99
ノン・ブート ... 247

[は]

バ ス ... 46
ハードウェアの初期化 ... 52

- パイプライン・アーキテクチャ ... 56
 バウンダリ・スキャン・テスト機能 ... 226
 ハンドシェーク ... 156,169
 汎用入出力ポート ... 193
 汎用レジスタ ... 46,108
 汎用レジスタとデータ・フォーマット ... 108
 ビット・リバース回路 ... 97
 ブート・モード指定 ... 241
 ブートアップ機能 ... 241
 プレ・ビット・リバース・アンド・ポスト・インデクス
 加算 ... 100
 フロー制御部 ... 68
 プログラム実行制御部 ... 61
 プログラム制御ユニット ... 60
 ページング機能 ... 221
 ペリフェラル ... 122
 ペリフェラル・レジスタ ... 124
 ペリフェラル STOP モード ... 223
 ペリフェラル - メモリ間転送 ... 187
 ペリフェラル - メモリ間転送バス ... 50
 ボーリング ... 157,169
 ポスト・イミディエイト加算 ... 101
 ポスト・インクリメント ... 99
 ホスト・インタフェース ... 161
 ポスト・インデクス加算 ... 99
 ポスト・デクリメント ... 99
 ホスト・ブート ... 245
 ホスト・ブートのパラメータ ... 246
 ポスト・モジュロ・インデクス加算 ... 100
- [ま]
- マスク・モード ... 208
 マルチプレクサ ... 97
 未使用端子の処理について ... 37
 命令訂正機能の動作 ... 221
 命令メモリ ... 62
 命令メモリ訂正機能 ... 220
 命令メモリのエイリアス ... 94
 メイン・バス ... 46
 メイン・バス接続レジスタ ... 47
 モジュロ・インデクス加算と巡回バッファ ... 102
- [や]
- 要因別割り込み許可フラグ ... 80
 予備フラグ ... 80
- [ら]
- リセット機能 ... 52
 リピート機能 ... 70
 リブート ... 249
 リブート後の割り込み ... 207
 リング・カウンタの動作範囲 ... 104
 累乗算器 ... 111
 累乗算機能 ... 114
 ループ機能 ... 71
 ループ・スタック・エラー・フラグ ... 89
 ループ・フラグ ... 80
 ループ/リピート制御回路 ... 69
 論理演算 ... 119
- [わ]
- 割り込み ... 76,159,171
 割り込みからの復帰 ... 84
 割り込み許可と禁止の設定順序 ... 207
 割り込み許可フラグ ... 79
 割り込みコントローラ ... 200
 割り込みシーケンス ... 83
 割り込み制御機能 ... 76
 割り込み制御のソフトウェア ... 79
 割り込みテーブル ... 203
 割り込みに関する注意点 ... 205
 割り込みの受け付け条件 ... 76
 割り込みの競合と記憶 ... 88
 割り込みの制限事項 ... 88
 割り込みの遅延 ... 86
 割り込みベクタ ... 77
 割り込みベクタ・テーブル ... 77
 割り込み要因 ... 76

A. 2 アルファベットで始まる語句の索引

[A]

Additional IO ... 228
 ALU 演算機能 ... 117
 ASIO ... 136

[B]

BE ... 194
 BSFT シフト演算機能 ... 120

[C]

CLKC ... 212
 CRC 回路 ... 237

[D]

DMA アクセス ... 181
 DSP コア・カーネル ... 45

[E]

EB ... 79
 EI ... 79
 EP ... 79

[F]

fint 命令 ... 206

[H]

HALT 命令によるスタンバイ・モード ... 58
 HIO ... 161

[I]

IEIO ... 224
 IMC ... 220
 INTC ... 200
 IO ... 195

[J]

JTAG ポート ... 224

[L]

LF ... 80

lse ... 89

[M]

MAC ... 111
 MAC 入力シフト ... 111
 ME ... 194
 MIO ... 172
 MSFT ... 111
 MUX ... 97

[O]

ovf ... 89

[P]

PIO ... 193
 PLL ロック・レンジ設定 ... 52
 PLL 通倍率設定 ... 51
 PMT ... 187
 PMT 転送手順 ... 192

[S]

SDCIF ... 230
 SD カード・インタフェース ... 230
 SIO ... 145
 ste ... 89
 STOP 命令によるスタンバイ・モード ... 59

[T]

TIM ... 208
 TSIO ... 127

[X]

XAA ... 97
 XBRC ... 97
 XY メモリ・ブート .. 243
 X データ・バス ... 48
 X データ・バス接続レジスタおよびメモリ ... 49
 X データ・メモリ空間 ... 91
 X メモリ・ブート ... 242

[Y]	Y データ・バス接続レジスタおよびメモリ ...	50
YAA ...	Y データ・メモリ ...	97
YBRC ...	Y メモリ・ブート ...	97
Y データ・バス ...		49

A. 3 レジスタ索引

A. 3. 1 レジスタ名称順

[あ]

インデクス・レジスタ : DN0-DN7 ...	96
エラー・ステータス・レジスタ : ESR ...	89
オーディオ・シリアル・ステータス・レジスタ : ASST ...	139
オーディオ・シリアル・データ・レジスタ : ASDT ...	139

[か]

拡張インタフェース制御レジスタ : APCR ...	229
クロック制御レジスタ : CLKC ...	213

[さ]

シリアル出力シフト・レジスタ : SOS ...	130,139,148
シリアル・ステータス・レジスタ : SST ...	129,131,140,147
シリアル・データ・レジスタ : SDT ...	147
シリアル入力シフト・レジスタ : SIS ...	130,139,148
スタック : STK ...	63
スタック・ポインタ : SP ...	63
ステータス・レジスタ : SR ...	79, 205

[た]

タイマ・カウント・レジスタ : TCR ...	210
タイマ初期値設定レジスタ : TIR ...	209
タイマ制御レジスタ : TCSR ...	210
ダイレクト・アクセス・インデクス・レジスタ : MIDX ...	175
データ・ページ・レジスタ : DPR ...	222
データ・ポインタ : DP0-DP7 ...	96
訂正アドレス・レジスタ : CAR ...	220
訂正イネーブル・フラグ・レジスタ : CEFR ...	220
訂正ページ・レジスタ : CPR ...	220
訂正命令コード・レジスタ : CUIR , CLIR ...	221
テスト・インストラクション・レジスタ ...	226
テスト・バイパス・レジスタ ...	226

[は]

パワー制御レジスタ : POWC ... 223
汎用レジスタ : R0-R7 ... 46
プログラム・カウンタ : PC ... 61
ポート・コマンド・レジスタ : PCD ... 194
ポート・データ・レジスタ : PDT ... 194
ホスト・ステータス・レジスタ : HST ... 163
ホスト・データ・レジスタ : HDT ... 163

[ま]

命令ページ・レジスタ : IPR ... 222
メモリ・インタフェース・ウエイト・レジスタ : MWAIT ... 175
メモリ・インタフェース・スタート・アドレス・レジスタ : MADR ... 175
メモリ・インタフェース・セットアップ/ホールド幅レジスタ : MSHW ... 174
メモリ・インタフェース・ライン・オフセット・レジスタ : MOFS ... 175
メモリ・インタフェース・ライン・レンジ・レジスタ : MLEN ... 176
メモリ・インタフェース制御/ステータス・レジスタ : MCST ... 175
メモリ・データ・レジスタ : MDT ... 174
モジュロ・レジスタ : DMX, DMY ... 97

[ら]

リピート・カウンタ : RC ... 68, 70
ループ開始アドレス・レジスタ : LSA ... 68
ループ・カウンタ : LC ... 69, 72
ループ終端アドレス・レジスタ : LEA ... 68
ループ・スタック : LSTK ... 69
ループ・スタック・ポインタ : LSP ... 69

[わ]

割り込み許可フラグ・スタック・レジスタ : EIR ... 81
割り込み制御レジスタ : ICR ... 201

[A~Z]

DMA アクセス・データ・アドレス・レジスタ : MADP ... 176
DMA アクセス・データ残りレンジ・レジスタ : MALP ... 176
PMT アドレス・ポインタ : PMP ... 189
PMT サイズ・レジスタ : PMS ... 189
PMT スタート・アドレス・レジスタ : PMSA ... 189
PMT 制御レジスタ : PMC ... 189
PMT ワード・カウンタ : PMWC ... 189
SD カード・インタフェース・コマンド・シフト・レジスタ : SDCSR ... 233
SD カード・インタフェース・データ・シフト・レジスタ : SDDSR ... 233
SD カード・インタフェース・レスポンス・レジスタ : SDRPR ... 233

SD カード・インタフェース CRC ステータス・レジスタ : SDSBR	...	233
SD カード・インタフェース CRC レジスタ : SDCRCR	...	233
SD カード・インタフェース制御レジスタ : SDCTL	...	233
SD カード・コマンド・アーギュメント・ハイ側レジスタ : SDCMD_AGH	...	232
SD カード・コマンド・アーギュメント・ロウ側レジスタ : SDCMD_AGL	...	232
SD カード・コマンド・インデクス・レジスタ : SDCMD_IDX	...	232
SD カード・コマンド・レジスタ : SDCMD	...	232
SD カード・データ・レジスタ : SDDR	...	232
TDM シリアル・ステータス・レジスタ : TSST	...	129
TDM シリアル・データ・レジスタ : TSDT	...	129
TDM フレーム・フォーマット・レジスタ : TFMT	...	129
TDM 受信スロット・レジスタ : TRXH	...	129
TDM 受信スロット・レジスタ : TRXL	...	129
TDM 送信スロット・レジスタ (上位) : TTXH	...	129
TDM 送信スロット・レジスタ (下位) : TTXL	...	129

A. 3.2 レジスタ略号順

[A]

APCR : 拡張インタフェース制御レジスタ	...	229
ASDT : オーディオ・シリアル・データ・レジスタ	...	139
ASST : オーディオ・シリアル・ステータス・レジスタ	...	139

[C]

CAR : 訂正アドレス・レジスタ	...	220
CEFR : 訂正イネーブル・フラグ・レジスタ	...	220
CLKC : クロック制御レジスタ	...	213
CPR : 訂正ページ・レジスタ	...	220
CUIR , CLIR : 訂正命令コード・レジスタ	...	221

[D]

DMX , DMY : モジュロ・レジスタ	...	97
DN0-DN7 : インデクス・レジスタ	...	96
DP0-DP7 : データ・ポインタ	...	96
DPR : データ・ページ・レジスタ	...	222

[E]

EIR : 割り込み許可フラグ・スタック・レジスタ	...	81
ESR : エラー・ステータス・レジスタ	...	89

[H]

HDT : ホスト・データ・レジスタ	...	163
HST : ホスト・ステータス・レジスタ	...	163

[I]

ICR : 割り込み制御レジスタ ... 201

IPR : 命令ページ・レジスタ ... 222

[L]

LC : ループ・カウンタ ... 69, 72

LEA : ループ終端アドレス・レジスタ ... 68

LSA : ループ開始アドレス・レジスタ ... 68

LSP : ループ・スタック・ポインタ ... 69

LSTK : ループ・スタック ... 69

[M]

MADP : DMA アクセス・データ・アドレス・レジスタ ... 176

MADR : メモリ・インタフェース・スタート・アドレス・レジスタ ... 175

MALP : DMA アクセス・データ残りレングス・レジスタ ... 176

MCST : メモリ・インタフェース制御 / ステータス・レジスタ ... 175

MDT : メモリ・データ・レジスタ ... 174

MIDX : ダイレクト・アクセス・インデクス・レジスタ ... 175

MLEN : メモリ・インタフェース・ライン・レングス・レジスタ ... 176

MOFS : メモリ・インタフェース・ライン・オフセット・レジスタ ... 175

MSHW : メモリ・インタフェース・セットアップ / ホールド幅レジスタ ... 174

MWAIT : メモリ・インタフェース・ウエイト・レジスタ ... 175

[P]

PC : プログラム・カウンタ ... 61

PCD : ポート・コマンド・レジスタ ... 194

PDT : ポート・データ・レジスタ ... 194

PMC : PMT 制御レジスタ ... 189

PMP : PMT アドレス・ポインタ ... 189

PMS : PMT サイズ・レジスタ ... 189

PMSA : PMT スタート・アドレス・レジスタ ... 189

PMWC : PMT ワード・カウンタ ... 189

POWC : パワー制御レジスタ ... 223

[R]

R0-R7 : 汎用レジスタ ... 46

RC : リピート・カウンタ ... 68, 70

[S]

SDCMD : SD カード・コマンド・レジスタ ... 232

SDCMD_IDX : SD カード・コマンド・インデクス・レジスタ ... 232

SDCMD_AGH : SD カード・コマンド・アーギュメント・ハイ側レジスタ ... 232

SDCMD_AGL : SD カード・コマンド・アーギュメント・ロウ側レジスタ ... 232

SDCRCR : SD カード・インタフェース CRC レジスタ ... 233
SDCSR : SD カード・インタフェース・コマンド・シフト・レジスタ ... 233
SDCTL : SD カード・インタフェース制御レジスタ ... 233
SDDR : SD カード・データ・レジスタ ... 232
SDDSR : SD カード・インタフェース・データ・シフト・レジスタ ... 233
SDRPR : SD カード・インタフェース・レスポンス・レジスタ ... 233
SDSBR : SD カード・インタフェース CRC ステータス・ビジィ・レジスタ ... 233
SDT : シリアル・データ・レジスタ ... 147
SIS : シリアル入力シフト・レジスタ ... 130,139,148
SOS : シリアル出力シフト・レジスタ ... 130,139,148
SP : スタック・ポインタ ... 63
SR : ステータス・レジスタ ... 79, 205
SST : シリアル・ステータス・レジスタ ... 129,131,140,147
STK : スタック ... 63

[T]

TCR : タイマ・カウント・レジスタ ... 210
TCSR : タイマ制御レジスタ ... 210
TFMT : TDM フレーム・フォーマット・レジスタ ... 129
TIR : タイマ初期値設定レジスタ ... 209
TRXH : TDM 受信スロット・レジスタ (上位) ... 129
TRXL : TDM 受信スロット・レジスタ (下位) ... 129
TSDT : TDM シリアル・データ・レジスタ ... 129
TSST : TDM シリアル・ステータス・レジスタ ... 129
TTXH : TDM 送信スロット・レジスタ (上位) ... 129
TTXL : TDM 送信スロット・レジスタ (下位) ... 129

【発 行】

NECエレクトロニクス株式会社

〒211-8668 神奈川県川崎市中原区下沼部1753

電話（代表）：044(435)5111

—— お問い合わせ先 ——

【ホームページ】

NECエレクトロニクスの情報がインターネットでご覧になれます。

URL(アドレス) <http://www.necel.co.jp/>

【営業関係，技術関係お問い合わせ先】

半導体ホットライン

(電話：午前 9:00～12:00，午後 1:00～5:00)

電 話 : 044-435-9494

E-mail : info@necel.com

【資料請求先】

NECエレクトロニクスのホームページよりダウンロードいただくか，NECエレクトロニクスの販売特約店へお申し付けください。