

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、誤りがな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

μPD77111 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

μPD77110

μPD77111

μPD77112

μPD77113A

μPD77114

μPD77115

[メ モ]

目 次 要 約

第 1 章	概 説	...	19
第 2 章	端子機能	...	23
第 3 章	アーキテクチャ	...	48
第 4 章	ブート機能	...	180
第 5 章	μ PD77115 の機能概要	...	198
第 6 章	μ PD77115 のペリフェラル	...	214
第 7 章	μ PD77115 のブート機能	...	253
第 8 章	開発ツール	...	257
付録 A	オーダ情報	...	261
付録 B	索 引	...	263

CMOSデバイスの一般的注意事項

静電気対策 (MOS全般)

注意 MOSデバイス取り扱いの際は静電気防止を心がけてください。

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

また、MOSデバイスを実装したボードについても同様の扱いをしてください。

未使用入力の処理 (CMOS特有)

注意 CMOSデバイスの入力レベルは固定してください。

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

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

初期化以前の状態 (MOS全般)

注意 電源投入時、MOSデバイスの初期状態は不定です。

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

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

Windows, Windows NT は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

PC/AT は米国 IBM 社の商標です。

本製品のうち、外国為替および外国貿易管理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 PD77110GC-9EU, μ PD77115F1-CN1, μ PD77115GK-9EU
ユーザ判定品 : μ PD77111GK- $x \times x$ -9EU, μ PD77111F1- $x \times x$ -CN1, μ PD77112GC- $x \times x$ -9EU, μ PD77113AF1- $x \times x$ -CN1, μ PD77114GC- $x \times x$ -9EU

- 本資料の内容は予告なく変更することがありますので、最新のものであることをご確認の上ご使用ください。
 - 文書による当社の承諾なしに本資料の転載複製を禁じます。
 - 本資料に記載された製品の使用もしくは本資料に記載の情報の使用に際して、当社は当社もしくは第三者の知的財産権その他の権利に対する保証または実施権の許諾を行うものではありません。上記使用に起因する第三者所有の権利にかかわる問題が発生した場合、当社はその責を負うものではありませんのでご了承ください。
 - 本資料に記載された回路、ソフトウェア、及びこれらに付随する情報は、半導体製品の動作例、応用例を説明するためのものです。従って、これら回路・ソフトウェア・情報をお客様の機器に使用される場合には、お客様の責任において機器設計をしてください。これらの使用に起因するお客様もしくは第三者の損害に対して、当社は一切その責を負いません。
 - 当社は品質、信頼性の向上に努めていますが、半導体製品はある確率で故障が発生します。当社半導体製品の故障により結果として、人身事故、火災事故、社会的な損害等を生じさせない冗長設計、延焼対策設計、誤動作防止設計等安全設計に十分ご注意願います。
 - 当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定して頂く「特定水準」に分類しております。また、各品質水準は以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認の上ご使用願います。
 - 標準水準：コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット
 - 特別水準：輸送機器（自動車、列車、船舶等）、交通用信号機器、防災/防犯装置、各種安全装置、生命維持を直接の目的としない医療機器
 - 特定水準：航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器、生命維持のための装置またはシステム等
- 当社製品のデータ・シート/データ・ブック等の資料で、特に品質水準の表示がない場合は標準水準製品であることを表します。当社製品を上記の「標準水準」の用途以外でご使用をお考えのお客様は、必ず事前に当社販売窓口までご相談頂きますようお願い致します。

M7 98.8

本版で改訂された主な箇所

箇所	内容
全般	μ PD77115 に関する記述を追加。
p.42	2.3.2 μ PD77115 の端子機能を追加。
p.190	4.5 シグネチャ演算の説明を追加。
p.191	4.6 命令 ROM 修正処理の説明を追加。
p.198	第 5 章 μ PD77115 の機能概要を追加。
p.214	第 6 章 μ PD77115 のペリフェラルを追加。
p.253	第 7 章 μ PD77115 のブート機能を追加。
p.257	8.1.2 ハイスピード・シミュレータ (HSM77016) のオーダ名称を変更。
p.266	B.3 レジスタ索引を追加。

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。

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

はじめに

対象者 このマニュアルは、 μ PD77111 ファミリの機能を理解し、それ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 PD77111 ファミリは、 μ PD77110, 77111, 77112, 77113A, 77114, 77115 の総称です。特に機能面に違いがない場合は、 μ PD77111 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。

目的 このマニュアルは、次の構成に示す μ PD77111 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 アーキテクチャ
- 第4章 ブート機能
- 第5章 μ PD77115 の機能概要
- 第6章 μ PD77115 のペリフェラル
- 第7章 μ PD77115 のブート機能
- 第8章 開発ツール
- 付録 A オーダ情報
- 付録 B 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 PD77110, 77111, 77112, 77113A, 77114 の基本機能を一通り理解しようとするとき

第1章 概説から第4章 ブート機能、第8章 開発ツールをお読みください。

μ PD77115 の機能を理解しようとするとき

第1章 概説、第2章 端子機能、第5章 μ PD77115 の機能概要から第8章 開発ツールをお読みください。

ソフトウェア・エンジニアの方へ

第1章 概説から第8章 開発ツールをお読みください。とくに、**第8章 開発ツール**では、ソフトウェアを開発するためのツール類、 μ PD77111 ファミリのツール類および評価システムを紹介しています。

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>x x x</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... x x x または 0b x x x 10進数... x x x x 16進数... 0x x x x x

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 PD77111 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 PD77110	U12395J	U12801J	このマニュアル	U13116J	U11958J	U12021J
μ PD77111						
μ PD77112						
μ PD77113A		U14373J				
μ PD77114						
μ PD77115						

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

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 説 ... 19

- 1.1 μ PD77111 ファミリの特徴 ... 19
- 1.2 μ PD77111 ファミリの主な応用分野 ... 21

第 2 章 端子機能 ... 23

- 2.1 端子接続図 ... 25
- 2.2 端子構成 ... 34
- 2.3 端子機能 ... 36
 - 2.3.1 μ PD77110, 77111, 77112, 77113A, 77114 の端子機能 ... 36
 - 2.3.2 μ PD77115 の端子機能 ... 42
- 2.4 未使用端子の処理について ... 46

第 3 章 アーキテクチャ ... 48

- 3.1 全体ブロック構成 ... 48
- 3.2 バ ス ... 50
 - 3.2.1 メイン・バス ... 50
 - 3.2.2 データ・バス ... 51
- 3.3 システム制御ユニット ... 54
 - 3.3.1 クロック・ジェネレータ ... 55
 - 3.3.2 リセット機能 ... 58
 - 3.3.3 パイプライン・アーキテクチャ ... 61
 - 3.3.4 スタンバイ機能 ... 63
- 3.4 プログラム制御ユニット ... 68
 - 3.4.1 ブロック構成 ... 68
 - 3.4.2 プログラム実行制御部 ... 69
 - 3.4.3 フロー制御部 ... 75
 - 3.4.4 割り込み ... 83
 - 3.4.5 エラー・ステータス・レジスタ (ESR : error status register) ... 97
- 3.5 データ・アドレッシング・ユニット ... 98
 - 3.5.1 ブロック構成 ... 98
 - 3.5.2 データ・メモリ空間 ... 98
 - 3.5.3 アドレッシング・モード ... 107
- 3.6 演算ユニット ... 118
 - 3.6.1 ブロック構成 ... 119
 - 3.6.2 汎用レジスタとデータ・フォーマット ... 119
 - 3.6.3 累乗算器 (MAC) と MAC 入力シフタ (MSFT) の演算機能 ... 122
 - 3.6.4 ALU 演算機能 ... 129
 - 3.6.5 BSFT シフト演算機能 ... 132
- 3.7 ペリフェラル・ユニット ... 134
 - 3.7.1 ブロック構成 ... 135

3.7.2	ペリフェラル・レジスタ	...	136
3.7.3	シリアル・インタフェース	...	137
3.7.4	ホスト・インタフェース	...	155
3.7.5	汎用入出力ポート	...	168
3.7.6	ウェイト・コントローラ	...	176
3.7.7	ディバグ・インタフェース (JTAG)	...	177

第4章 ブート機能 ... 180

4.1	概 要	...	180
4.2	ブート・モード	...	181
4.2.1	ブート・モードの分類	...	181
4.3	リセット時のブート	...	184
4.3.1	セルフ・ブートの場合	...	185
4.3.2	ホスト・ブートの場合	...	187
4.4	ブート・サブルーチン (リブート)	...	189
4.4.1	Xメモリ・ワード・リブート, Xメモリ・バイト・リブートのパラメータ	...	189
4.4.2	Yメモリ・ワード・リブート, Yメモリ・バイト・リブートのパラメータ	...	190
4.4.3	ホスト・リブートのパラメータ	...	190
4.5	シグネチャ演算	...	190
4.6	命令ROM修正処理	...	191
4.6.1	命令ROM修正処理の概要	...	191
4.6.2	命令ROM修正処理の使用方法	...	193
4.6.3	プログラム実行フロー	...	195
4.6.4	注意事項	...	196
4.7	ブート所要時間	...	197

第5章 μ PD77115の機能概要 ... 198

5.1	特 徴	...	198
5.2	内部ブロック図	...	200
5.3	リセット機能	...	201
5.3.1	リセット・タイミング	...	201
5.3.2	電源投入時の注意	...	202
5.3.3	電源投入時の推奨シーケンス	...	203
5.4	スタンバイ機能	...	203
5.4.1	ホールド・モード	...	203
5.4.2	ストップ・モード	...	205
5.5	クロック制御	...	206
5.5.1	概 要	...	206
5.5.2	クロック制御部の端子構成	...	207
5.5.3	CLKCNTLレジスタ	...	209
5.5.4	クロック状態遷移	...	210
5.5.5	PLLに関する留意事項	...	211
5.6	メモリ構成	...	212
5.7	割り込みベクタ・テーブル	...	213

第6章 μ PD77115のペリフェラル ... 214

6.1	ペリフェラル・レジスタ	...	215
6.2	シリアル・インタフェース	...	216
	6.2.1 オーディオ・シリアル・インタフェース	...	217
	6.2.2 スタンダード・シリアル・インタフェース	...	226
6.3	ホスト・インタフェース	...	227
	6.3.1 ホスト・インタフェースの端子	...	227
	6.3.2 ホスト・インタフェースのレジスタ	...	229
	6.3.3 ホスト・インタフェース用レジスタのアドレス	...	232
	6.3.4 ホスト・インタフェースのタイミング	...	233
6.4	汎用入出力ポート	...	235
	6.4.1 汎用入出力ポートの端子	...	235
	6.4.2 汎用入出力ポートのレジスタ	...	236
	6.4.3 汎用入出力ポートのタイミング	...	238
6.5	SDカード・インタフェース	...	240
	6.5.1 SDカード・インタフェースの端子	...	241
	6.5.2 SDカード・インタフェースのレジスタ	...	242
	6.5.3 CRC (Cyclic Redundancy Codes) 回路	...	246
	6.5.4 データ・フォーマット	...	246
6.6	ペリフェラル・バッファ	...	248
	6.6.1 ペリフェラル・バッファのレジスタ	...	248
	6.6.2 ペリフェラル・バッファ動作フロー	...	250
6.7	タイマ	...	251
	6.7.1 タイマのレジスタ	...	251

第7章 μ PD77115 のブート機能 ... 253

7.1	リセット時のブート	...	253
	7.1.1 バイト・ブートの場合	...	254
	7.1.2 ワード・ブートの場合	...	255
7.2	リブート	...	255
7.3	シグネチャ演算	...	256
7.4	ブート所要時間	...	256

第8章 開発ツール ... 257

8.1	ソフトウェア・ツール	...	257
	8.1.1 統合化開発環境ワークベンチ (WB77016)	...	257
	8.1.2 ハイスピード・シミュレータ (HSM77016)	...	257
	8.1.3 Cコンパイラ (CC77016)	...	258
	8.1.4 インサーキット・エミュレータ用システム・ソフトウェア (ID77016)	...	258
	8.1.5 DSP 開発ツール・ソフトウェア・パッケージ (SP77016)	...	258
8.2	ハードウェア・ツール	...	259
	8.2.1 インサーキット・エミュレータ	...	259
	8.2.2 その他のツール	...	260

付録 A オーダ情報 ... 261

A.1	オーダ情報	...	261
A.2	マスク・オプション	...	261

- A. 2. 1 クロック制御に関するオプション ... 261
- A. 2. 2 WAKEUP オプション ... 262
- A.3 マスク ROM 受注フォーマット ... 262

付録 B 索引 ... 263

- B. 1 アルファベットで始まる語句の索引 ... 263
- B. 2 五十音で始まる語句の索引 ... 264
- B. 3 レジスタ索引 ... 266
 - B. 3. 1 レジスタ名称順 ... 266
 - B. 3. 2 レジスタ略号順 ... 268

図の目次 (1/3)

図番号	タイトル, ページ
2 - 1	100 ピン・プラスチック TQFP (ファインピッチ)(14×14) ... 25
2 - 2	80 ピン・プラスチック TQFP (ファインピッチ)(12×12) ... 27
2 - 3	80 ピン・プラスチック TQFP (ファインピッチ)(12×12) ... 29
2 - 4	80 ピン・プラスチック・ファインピッチ BGA (9×9) ... 31
2 - 5	μPD77111 ファミリの端子構成図 ... 34
3 - 1	全体ブロック構成 ... 49
3 - 2	μPD77111 ファミリのクロック回路 ... 56
3 - 3	μPD77111 ファミリのクロック・タイミング ... 56
3 - 4	リセット・タイミング ... 60
3 - 5	PLL 設定タイミング ... 60
3 - 6	パイプライン・イメージ ... 61
3 - 7	ホールド・モード ... 64
3 - 8	プログラム制御ユニット ... 68
3 - 9	命令メモリ・マップ ... 69
3 - 10	PC の通常動作 ... 71
3 - 11	無条件イミューディエト・ジャンプのタイミング ... 73
3 - 12	無条件間接ジャンプのタイミング ... 73
3 - 13	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) ... 74
3 - 14	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) ... 74
3 - 15	RC のフォーマット ... 77
3 - 16	リピート命令の例 (2 回リピート) ... 78
3 - 17	リピート実行のタイミング (2 回リピート) ... 78
3 - 18	LC のフォーマット ... 79
3 - 19	ループ実行のタイミング (2 回ループ動作例) ... 81
3 - 20	外部割り込みタイミング ... 85
3 - 21	多重割り込みの概念 ... 91
3 - 22	割り込み受け付けタイミング ... 93
3 - 23	RETI 命令によるタイミング ... 94
3 - 24	割り込み遅延タイミング (1 サイクル遅延) ... 95
3 - 25	割り込み遅延タイミング (2 サイクル遅延) ... 96
3 - 26	データ・アドレッシング・ユニット ... 98
3 - 27	データ・メモリ・マップ ... 99
3 - 28	データ・メモリ・リード・サイクルのタイミング ... 103
3 - 29	データ・メモリ・ライト・サイクルのタイミング ... 103
3 - 30	データ・メモリ制御バンクと DWTR のフィールド構成 ... 104
3 - 31	バス・アービトレーションの手順 ... 106
3 - 32	DPn のビット反転 ... 111
3 - 33	DPn の分割 ... 113

図の目次 (2/3)

図番号	タイトル, ページ
3 - 34	通常のリモジュール演算の写像 ... 113
3 - 35	リモジュール調整の写像 ... 114
3 - 36	演算ユニットの詳細ブロック ... 119
3 - 37	汎用レジスタの分割フォーマット ... 120
3 - 38	汎用レジスタとデータ・メモリ間のデータ転送 ... 121
3 - 39	有符号値同士の乗算 ... 123
3 - 40	有符号値と無符号値の乗算 ... 124
3 - 41	無符号値同士の乗算 ... 125
3 - 42	累乗算 ... 127
3 - 43	1ビット・シフト累乗算 ... 128
3 - 44	16ビット・シフト累乗算 ... 129
3 - 45	BSFT 演算 ... 133
3 - 46	ペリフェラル・ユニット (μ PD77115 以外) ... 135
3 - 47	シリアル・インタフェース ... 137
3 - 48	シリアル・インタフェースの動作ブロック図 (1チャンネル) ... 140
3 - 49	シリアル・インタフェース出力タイミング ... 148
3 - 50	シリアル・インタフェース入力タイミング ... 150
3 - 51	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 ... 151
3 - 52	ホスト・インタフェース ... 155
3 - 53	ホスト・インタフェースの動作ブロック図 ... 157
3 - 54	ホスト・リード手順 (μ PD77111 ファミリー ホスト): HDT 書き込みウエイトなし ... 163
3 - 55	ホスト・ライト手順 (μ PD77111 ファミリー ホスト): HDT 読み出しウエイトなし ... 164
3 - 56	汎用入出力ポート ... 168
3 - 57	ウエイト・コントローラ ... 176
3 - 58	JTAG 端子外観図 ... 179
3 - 59	JTAG 端子処理 ... 179
4 - 1	セルフ・ブートのシステム構成例 ... 182
4 - 2	ホスト・ブートのシステム構成例 ... 183
4 - 3	ワード・ブートのイメージ ... 183
4 - 4	バイト・ブートのイメージ ... 184
4 - 5	ホスト・ブートの手順イメージ ... 187
4 - 6	命令 ROM 修正処理の概要 ... 192
4 - 7	命令 RAM にブートするプログラムのソース・イメージ例 ... 193
4 - 8	修正対応ルーチンのソース・イメージ例 ... 194
4 - 9	プログラム実行例 ... 195
5 - 1	μ PD77115 の内部ブロック図 ... 200
5 - 2	システム・リセット動作タイミング ... 201

図の目次 (3/3)

図番号	タイトル, ページ
5 - 3	ホールド・モード割り込み受け付けタイミング ... 203
5 - 4	クロック制御部ブロック ... 207
5 - 5	PLL ロック・アンロック状態遷移図 ... 210
5 - 6	μ PD77115 のメモリ・マップ ... 212
6 - 1	シリアル・インタフェース ... 217
6 - 2	オーディオ・シリアル出力タイミング ... 224
6 - 3	オーディオ・シリアル入力タイミング ... 225
6 - 4	ASIO 動作開始時の SO 動作 (マスタ・モードの場合) ...226
6 - 5	ホスト・インタフェース ... 227
6 - 6	ホスト・リード・タイミング ... 233
6 - 7	ホスト・ライト・タイミング ... 234
6 - 8	汎用入出力ポート ... 235
6 - 9	汎用ポートのデータ入力タイミング ... 238
6 - 10	汎用ポートのデータ出力タイミング ... 239
6 - 11	SD カード・インタフェース構成図 ... 241
6 - 12	PBU 動作フロー図 ... 250
6 - 13	タイマ構成図 ... 251
7 - 1	ホスト・バイト・ブートの手順 ... 254
7 - 2	ホスト・ワード・ブートの手順 ... 255

表の目次 (1/2)

表番号	タイトル, ページ
1 - 1	μ PD77111 ファミリの特徴 ... 20
2 - 1	機能端子の処理 ... 46
2 - 2	非機能端子の処理 ... 47
3 - 1	メイン・バス接続レジスタ ... 51
3 - 2	機能ブロックとバス ... 52
3 - 3	X データ・バス接続レジスタおよびメモリ ... 53
3 - 4	Y データ・バス接続レジスタおよびメモリ ... 54
3 - 5	PLL 逡倍率設定 ... 57
3 - 6	初期化対象 CPU レジスタと初期値 ... 58
3 - 7	初期化対象メモリ・マップト・レジスタと初期値 ... 59
3 - 8	初期化対象端子と初期状態 ... 59
3 - 9	ホールド中の端子状態 ... 63
3 - 10	ストップ中の端子状態 ... 66
3 - 11	ストップ・モード解除リセット期間中の出力端子の状態 ... 67
3 - 12	内部命令メモリの容量 ... 70
3 - 13	分岐命令の分類 ... 72
3 - 14	割り込み要因 ... 83
3 - 15	割り込みベクタ・テーブル ... 86
3 - 16	ROM と RAM の容量 ... 100
3 - 17	外部データ・メモリの容量 ... 100
3 - 18	端子の状態 ... 102
3 - 19	DWTR フィールド設定値とウエイト・サイクル数 ... 105
3 - 20	X, Y メモリ空間への同時アクセスの可否 ... 106
3 - 21	データ・ポインタのモディファイー覧 ... 112
3 - 22	汎用レジスタのフォーマット ... 120
3 - 23	累乗算機能 ... 126
3 - 24	ペリフェラル・レジスタのメモリ・マッピング ... 136
3 - 25	シリアル入出力インタフェースのステータス表示 ... 139
3 - 26	ハードウェア・リセット中とハードウェア・リセット後の端子の状態 ... 141
3 - 27	シリアル入出力エラー・フラグ設定条件 ... 142
3 - 28	SST (SST1 : 0x3801 : X/ : Y, SST2 : 0x3803 : X/ : Y) の機能 ... 144
3 - 29	SICM ビットと SIEF ビットの組み合わせ ... 146
3 - 30	ホスト入出力インタフェースのステータス表示 ... 157
3 - 31	ハードウェア・リセット中とハードウェア・リセット後の端子の状態 ... 158
3 - 32	HST (0x3807 : X/ : Y) の機能 ... 160
3 - 33	ホスト入出力エラー・フラグ設定条件 ... 161
3 - 34	ホスト・インタフェース用レジスタの選択 ... 162

表の目次 (2/2)

表番号	タイトル, ページ
3 - 35	ポート・コマンド・レジスタ (PCD : 0x3805 : X/ : Y) ... 171
3 - 36	テスト・インストラクション ... 178
4 - 1	P0, P1 のパターンとブート・モード ... 184
4 - 2	セルフ・ブートのパラメータ (0x4000 : Y) ... 185
4 - 3	ワード・ブートのパラメータ・メモリ・マップ ... 186
4 - 4	バイト・ブートのパラメータ・メモリ・マップ ... 186
4 - 5	ブート・サブルーチンのエントリ・ポイント ... 189
4 - 6	ブート所要時間 ... 197
5 - 1	初期化対象 CPU レジスタと初期値 ... 201
5 - 2	初期化対象ペリフェラル・レジスタと初期値 ... 202
5 - 3	初期化対象端子と初期状態 ... 202
5 - 4	ホールド・モード中の端子状態 ... 204
5 - 5	ストップ・モード中の端子状態 ... 205
5 - 6	PLL 逡倍率設定 ... 208
5 - 7	CLKCNTL の機能 ... 209
5 - 8	割り込みベクタ・テーブル ... 213
6 - 1	ペリフェラル・レジスタのメモリ・マッピング ... 215
6 - 2	ASST の機能 ... 221
6 - 3	SST のビット 6, 7 の機能 ... 222
6 - 4	各クロックと周波数の関係 ... 223
6 - 5	HST の機能 ... 230
6 - 6	ホスト・インタフェース用レジスタのバイト・アドレス・マップ ... 232
6 - 7	ホスト・インタフェース用レジスタのワード・アドレス・マップ ... 232
6 - 8	PCD の機能 ... 237
6 - 9	SDCMD_IDX の機能 ... 244
6 - 10	SDCTL の機能 ... 245
6 - 11	PCR の機能 ... 249
6 - 12	TCSR の機能 ... 252
6 - 13	TENR の機能 ... 252
7 - 1	ブート・モード ... 253

第1章 概 説

μ PD77111 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 といいます）である μ PD7701 \times ファミリの後継品種として位置づけられます。 μ PD77111 ファミリは、 μ PD77110, 77111, 77112, 77113A, 77114, 77115 の総称です。 μ PD77111 ファミリは、大容量メモリの搭載と高速化、低消費電力化により、幅広いアプリケーションに対応できます。

備考 μ PD7701 \times ファミリは、 μ PD77015, 77016, 77017, 77018A, 77019 の総称です。

1.1 μ PD77111 ファミリの特徴

- ★ 高速インストラクション・サイクル
 - 13.3 ns (μ PD77111, 77112, 77113A, 77114, 77115 : 75 MHz)
 - 15.4 ns (μ PD77110 : 65 MHz)
- バスネックを解消するハーバード・アーキテクチャ
- 3 ステージのパイプライン・アーキテクチャ
- 合理的に組み合わされた並列命令
- 3 オペランド命令（三項演算）が可能なマルチプライ・アキュムレータ
- 8 個のデータ・メモリ用ポインタ・レジスタ（XY メモリ空間に 4 個ずつ）
- 柔軟で高速なデータ転送を約束するデュアル・データ・メモリ空間
- 柔軟なメモリ・アクセスを可能とする多彩なアドレッシング・モード
- 演算オーバフローを解消するヘッドルーム・フォーマット
- 柔軟な取り扱いの内蔵ペリフェラル
- 多様な外部インタフェース
- 幅広いアプリケーションに応える割り込み機能（内部 6 レベル、外部 4 レベル）
- オーバヘッドを最小限にとどめるハードウェア・ループ機能
- プログラマブルな外部メモリ・アクセス・ウエイト
- ブート ROM 内蔵
- デバッグ機能内蔵（JTAG ポート）
- HALT 命令、STOP 命令による低消費電力化が可能
- μ PD7701 \times ファミリとソフトウェア・レベルで互換

備考 μ PD77115 についての詳しい説明は、第 5 章 μ PD77115 の機能概要から 第 7 章 μ PD77115 のブート機能を参照してください。

★

表1-1 μPD77111 ファミリの特徴

項 目		μ PD77110	μ PD77111	μ PD77112	μ PD77113A	μ PD77114	μ PD77115
インストラクション・サイクル (最高速度動作時)		15.4 ns	13.3 ns				
動作クロック周波数 (最大)		65 MHz	75 MHz				
クロック逡倍回路	PLL 逡倍率	×1~8の整数 (外部端子による選択)	×1~16の整数 (マスク・オプション)				×1~16の整数 (外部端子による選択)
	分周率	÷1 (固定)	÷1~16の整数 (マスク・オプション)				÷1~16
	HALT 分周率	÷8 (固定)	÷1~16の整数 (マスク・オプション)				÷1~16
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×16ビット+40ビット 40ビット (三項演算: R0=R0+R1H*R2L など)					
アキュムレータ		40ビット入力, 40ビット出力 (二項演算, 単項演算)					
汎用レジスタ		8個の40ビット・レジスタ (R0-R7)					
データ・メモリ・ポインタ		Xメモリ空間用4個 (DP0-DP3), Yメモリ空間用4個 (DP4-DP7)					
割り込み		内部6レベル (6要因), 外部4レベル (4要因)					
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	ブートROM	256ワード内蔵 (ブート機能用)					
	内部RAM	35.5Kワード	1Kワード	3.5Kワード		11.5Kワード	
	内部ROM	なし	31.75Kワード	48Kワード		なし	
	外部領域	なし					
データ・メモリ	X内部RAM	24Kワード	3Kワード	16Kワード		16Kワード	
	X内部ROM	なし	16Kワード	32Kワード		なし	
	X外部領域	32Kワード	なし	16Kワード	なし	8Kワード	なし
	Y内部RAM	24Kワード	3Kワード	16Kワード		16Kワード	
	Y内部ROM	なし	16Kワード	32Kワード		なし	
	Y外部領域	32Kワード	なし	16Kワード	なし	8Kワード	なし
シリアル・インタフェース		2チャンネル					1チャンネル
ホスト・インタフェース		8ビット					16ビット
汎用入出力ポート (それぞれ独立に入出力設定可能)		4本					8本
電源	DSP コア	2.3V~2.7V	1.8V~2.7V				
	外部 I/O	2.7V~3.6V	2.7V~3.6V				
スタンバイ機能		HALT 命令実行による					
パワーダウン機能		STOP 命令実行による					
パッケージ		100ピン TQFP	80ピンFBGA, 80ピンTQFP	100ピン TQFP	80ピンFBGA	100ピン TQFP	80ピンFBGA 80ピンLQFP
その他							SDカード I/F
		デバッグ機能内蔵 (JTAGポート)					
		CMOS プロセス					

1.2 μ PD77111 ファミリの主な応用分野

一般に DSP はその名前に示すとおり、デジタル信号処理を目的として開発されたデバイスといえますが、 μ PD77111 ファミリに代表される新世代テクノロジーの DSP は、メモリ・アクセス能力、割り込み機能など、汎用 CPU の機能も備えています。したがって、応用分野も信号処理という限られた領域にとどまらず、きわめて幅広く考えることができます。次に、そのような応用分野を分野別にまとめて紹介します。

一般信号処理

- デジタル・フィルタ (FIR フィルタ, BIQUAD フィルタ等)
- 高速フーリエ変換
- ヒルベルト変換
- 相関処理
- アダプティブ・フィルタ

通信分野

- 高速モデム (V.32 等)
- デジタル自動車電話 (音声コーデック, イコライザ等)
- MPEG
- エコー・キャンセラ
- アダプティブ・イコライザ
- デジタル PBX
- DTMF エンコーダ / デコーダ
- FAX
- スペクトル拡散通信
- 多重化通信

音声 / 音響

- 音声認識
- 音声符号化 / 複合化 (ADPCM, PARCOR 等)
- 音声合成 (音素合成, 規則合成等)
- シンセサイザ
- 電子楽器
- 音場抑制
- 効果音

画像処理 / グラフィクス

- アフィン変換
- 2次元直交変換 (フーリエ変換, アダマール変換, KL 変換等)
- フィルタリング (平滑化, メディアン・フィルタ等)
- 各種オペレータ (ラプラシアン, ゴーベル等)
- レイ・トレーシング, マンデルブロー
- CAD (3D グラフィクス等)

バーチャル・リアリティ
画像圧縮／伸長（DCT，ランレングス，可変長コーディング）
画像認識
コンピュータ・アニメーション

制御

ナビゲーション・システム
ディスク制御（CD，LD 等）
各種サーボ・システム（PID，AC サーボ等）
レーザ・プリンタ，複写機等の制御
ロボット制御
NC 制御
ファジィ制御

計測

スペクトル・アナライザ
ファンクション・ジェネレータ
パターン・マッチング
ロック・イン・アンプ
ボックス・カー・インテグレータ
各種解析装置（振動解析，過渡解析）

その他，一般数値処理

データ暗号化／復号化
ニューメリカル・プロセッサとして利用
ニューラル・システム

第 2 章 端子機能

この章では μ PD77111 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 PD77111 ファミリの端子名称は次のとおりです。

BCLK :	Serial Clock Input/Output
$\overline{\text{BSTB}}$:	Bus Strobe
CLKIN :	Clock Input
CLKOUT :	Clock Output
D0-D15 :	16-Bit Data Bus
DA0-DA14 :	External Data Memory Address Bus
EV _{DD} :	Power Supply for I/O Pins
GND :	Ground
HA0 , HA1 :	Host Data Access
$\overline{\text{HCS}}$:	Host Chip Select
HD0-HD15 :	Host Data Bus
$\overline{\text{HOLDAK}}$:	Data Bus Hold Acknowledge
$\overline{\text{HOLDRQ}}$:	Data Bus Hold Request
$\overline{\text{HRD}}$:	Host Read
$\overline{\text{HRE}}$:	Host Read Enable
$\overline{\text{HWE}}$:	Host Write Enable
$\overline{\text{HWR}}$:	Host Write
I.C. :	Internally Connected
$\overline{\text{INT1}} - \overline{\text{INT4}}$:	Interrupt
IV _{DD} :	Power Supply for DSP Core
LRCLK :	Left Right Clock Input/Output
MCLK :	Master Clock Input
$\overline{\text{MWR}}$:	Memory Write Output
NC :	Non-Connection
NU :	Not Used
P0-P7 :	Port
PLL0-PLL3 :	PLL Multiple Rate Mode
$\overline{\text{RESET}}$:	Reset
SCK, SCK1 , SCK2 :	Serial Clock Input
SDCLK :	SD Card Clock Output
SDCR :	SD Card Command Output Response Input
SDDAT :	SD Card Data Input/Output
SI, SI1, SI2 :	Serial Data Input
SIACK1 :	Serial Input Acknowledge

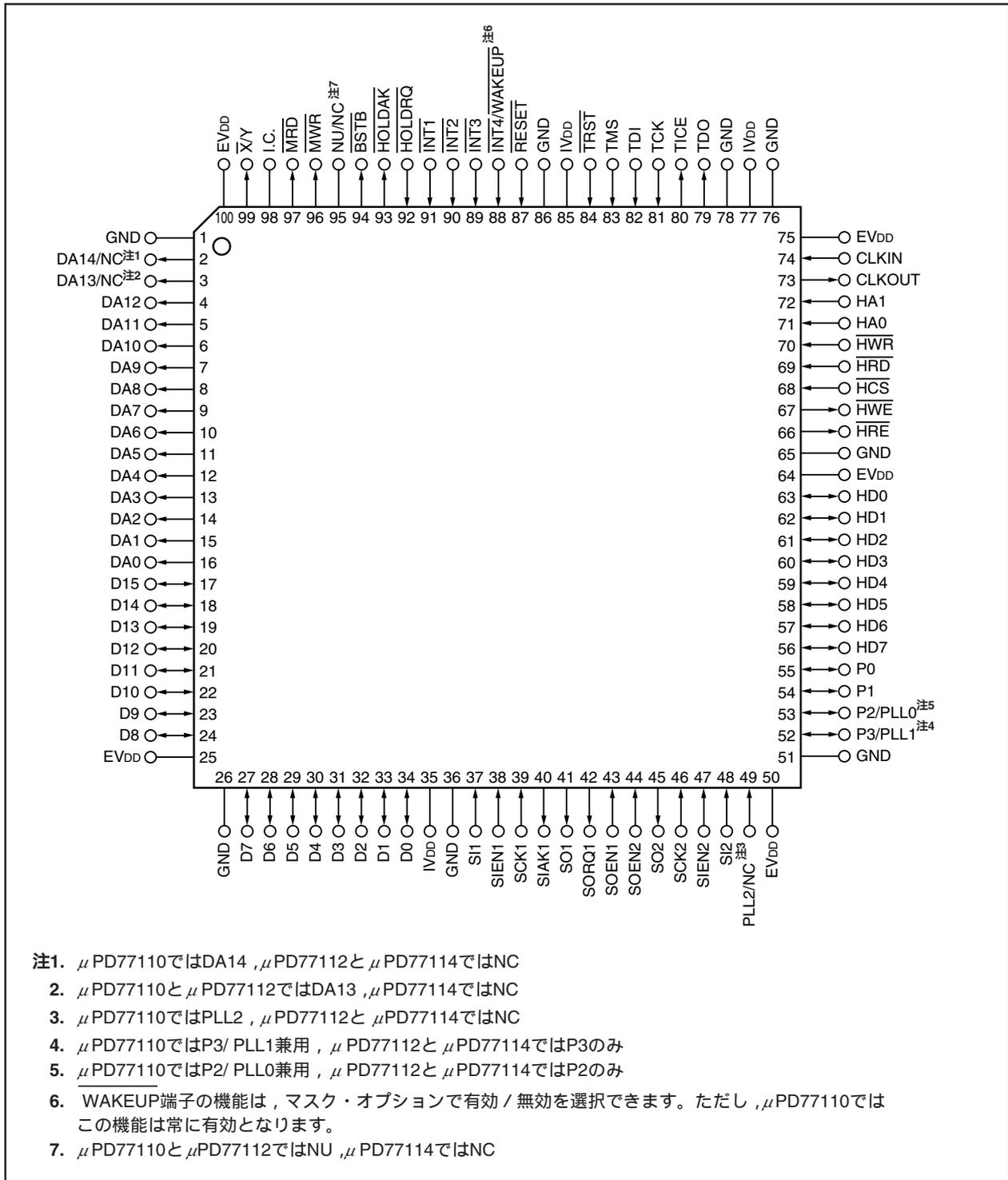
SIEN, SIEN1 , SIEN2 :	Serial Input Enable
SO, SO1 , SO2 :	Serial Data Output
SOEN, SOEN1 , SOEN2 :	Serial Output Enable
SORQ1 :	Serial Output Request
TCK :	Test Clock Input
TDI :	Test Data Input
TDO :	Test Data Output
TICE :	Test In-Circuit Emulator
TMS :	Test Mode Select
$\overline{\text{TRST}}$:	Test Reset
$\overline{\text{WAKEUP}}$:	Wakeup From STOP Mode
$\overline{\text{X}}/\text{Y}$:	X/Y Memory Select

2.1 端子接続図

(1) 100ピン・プラスチック TQFP (ファインピッチ)(14×14)

- ・ μ PD77110GC-9EU
- ・ μ PD77112GC- x x x -9EU
- ・ μ PD77114GC- x x x -9EU

図2-1 100ピン・プラスチック TQFP (ファインピッチ)(14×14)



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	26	GND	51	GND	76	GND
2	DA14/NC ^{注1}	27	D7	52	P3/PLL1 ^{注4}	77	IV _{DD}
3	DA13/NC ^{注2}	28	D6	53	P2/PLL0 ^{注5}	78	GND
4	DA12	29	D5	54	P1	79	TDO
5	DA11	30	D4	55	P0	80	TICE
6	DA10	31	D3	56	HD7	81	TCK
7	DA9	32	D2	57	HD6	82	TDI
8	DA8	33	D1	58	HD5	83	TMS
9	DA7	34	D0	59	HD4	84	$\overline{\text{TRST}}$
10	DA6	35	IV _{DD}	60	HD3	85	IV _{DD}
11	DA5	36	GND	61	HD2	86	GND
12	DA4	37	SI1	62	HD1	87	$\overline{\text{RESET}}$
13	DA3	38	SIEN1	63	HD0	88	$\overline{\text{INT4}} / \overline{\text{WAKEUP}}$ ^{注6}
14	DA2	39	SCK1	64	EV _{DD}	89	$\overline{\text{INT3}}$
15	DA1	40	SIK1	65	GND	90	$\overline{\text{INT2}}$
16	DA0	41	SO1	66	$\overline{\text{HRE}}$	91	$\overline{\text{INT1}}$
17	D15	42	SORQ1	67	$\overline{\text{HWE}}$	92	$\overline{\text{HOLDRQ}}$
18	D14	43	SOEN1	68	$\overline{\text{HCS}}$	93	$\overline{\text{HOLDAK}}$
19	D13	44	SOEN2	69	$\overline{\text{HRD}}$	94	$\overline{\text{BSTB}}$
20	D12	45	SO2	70	$\overline{\text{HWR}}$	95	NU/NC ^{注7}
21	D11	46	SCK2	71	HA0	96	$\overline{\text{MWR}}$
22	D10	47	SIEN2	72	HA1	97	$\overline{\text{MRD}}$
23	D9	48	SI2	73	CLKOUT	98	I.C.
24	D8	49	PLL2/NC ^{注3}	74	CLKIN	99	$\overline{\text{X}} / \overline{\text{Y}}$
25	EV _{DD}	50	EV _{DD}	75	EV _{DD}	100	EV _{DD}

注 1. μ PD77110 では DA14 , μ PD77112 と μ PD77114 では NC

2. μ PD77110 と μ PD77112 では DA13 , μ PD77114 では NC

3. μ PD77110 では PLL2 , μ PD77112 と μ PD77114 では NC

4. μ PD77110 では P3/PLL1 兼用 , μ PD77112 と μ PD77114 では P3 のみ

5. μ PD77110 では P2/PLL0 兼用 , μ PD77112 と μ PD77114 では P2 のみ

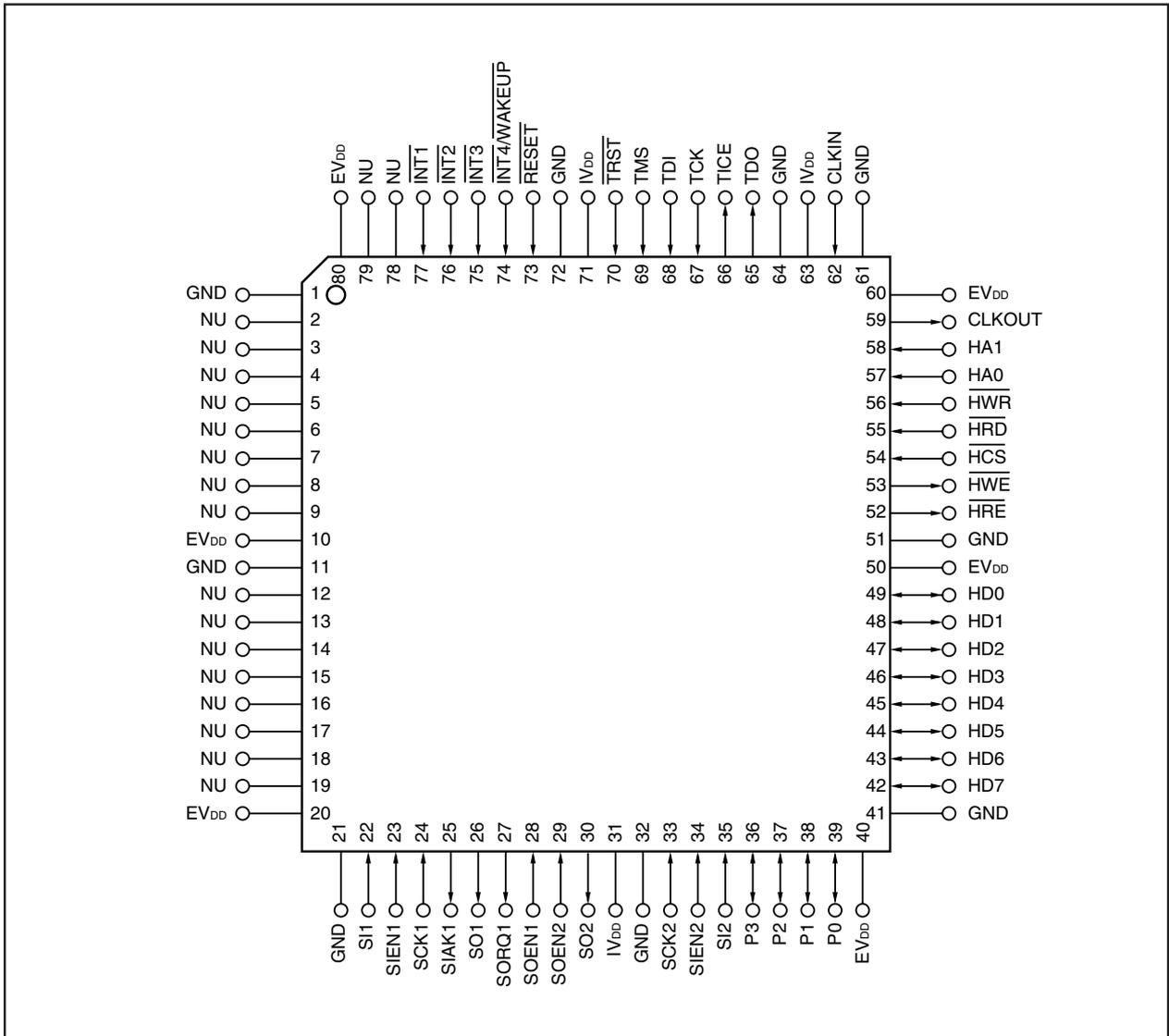
6. $\overline{\text{WAKEUP}}$ 端子の機能は、マスク・オプションで有効/無効を選択できます。ただし、 μ PD77110 では、この機能は常に有効となります。

7. μ PD77110 と μ PD77112 では NU , μ PD77114 では NC

(2) 80ピン・プラスチックTQFP (ファインピッチ)(12×12)

・μPD77111GK-xxx-9EU

図2-2 80ピン・プラスチックTQFP (ファインピッチ)(12×12)

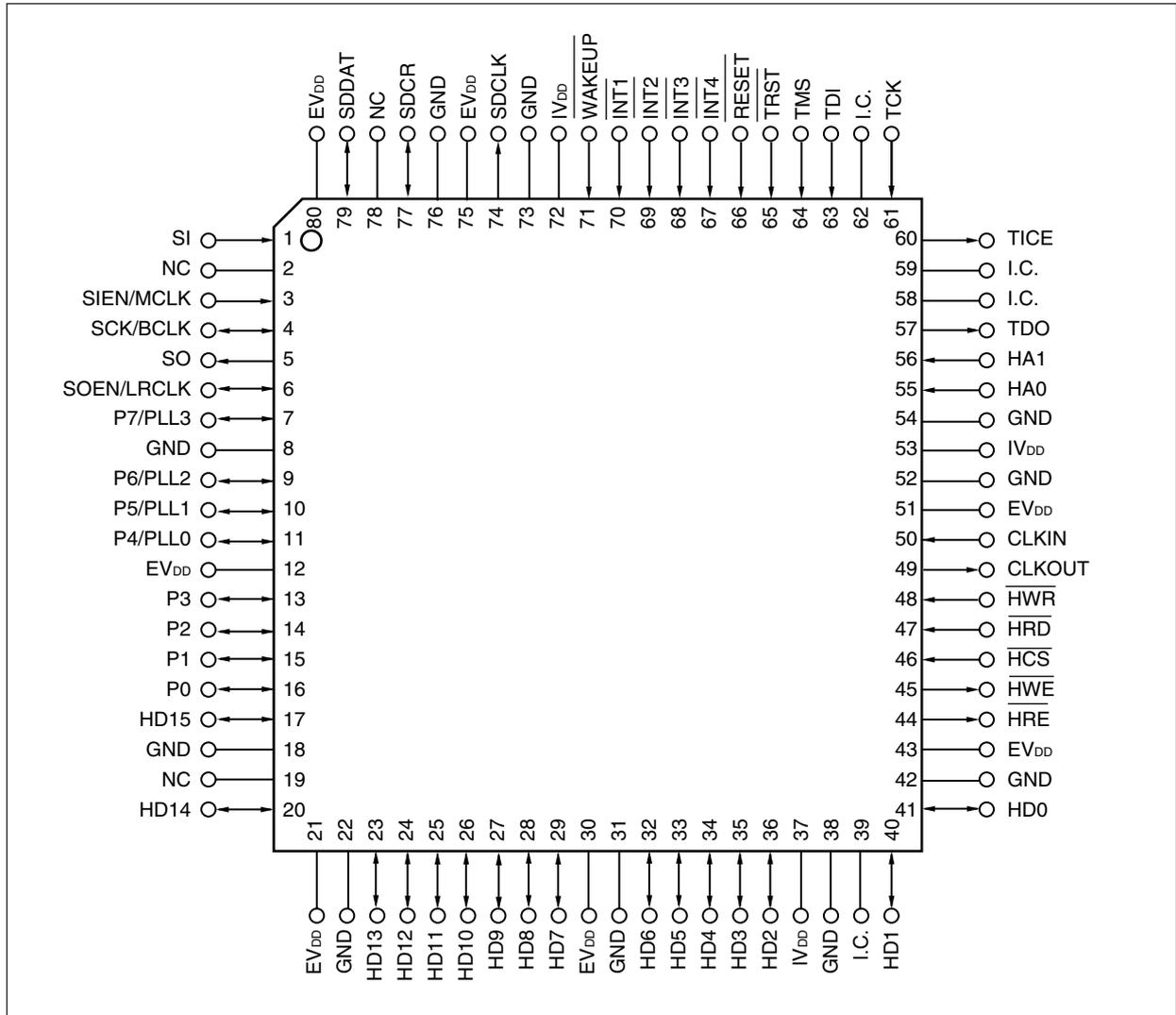


端子番号	端子名	端子番号	端子名	端子番号	端子名	端子番号	端子名
1	GND	21	GND	41	GND	61	GND
2	NU	22	SI1	42	HD7	62	CLKIN
3	NU	23	SIEN1	43	HD6	63	IV _{DD}
4	NU	24	SCK1	44	HD5	64	GND
5	NU	25	SIK1	45	HD4	65	TDO
6	NU	26	SO1	46	HD3	66	TICE
7	NU	27	SORQ1	47	HD2	67	TCK
8	NU	28	SOEN1	48	HD1	68	TDI
9	NU	29	SOEN2	49	HD0	69	TMS
10	EV _{DD}	30	SO2	50	EV _{DD}	70	$\overline{\text{TRST}}$
11	GND	31	IV _{DD}	51	GND	71	IV _{DD}
12	NU	32	GND	52	$\overline{\text{HRE}}$	72	GND
13	NU	33	SCK2	53	$\overline{\text{HWE}}$	73	$\overline{\text{RESET}}$
14	NU	34	SIEN2	54	$\overline{\text{HCS}}$	74	$\overline{\text{INT4}} / \overline{\text{WAKEUP}}$ 注
15	NU	35	SI2	55	$\overline{\text{HRD}}$	75	$\overline{\text{INT3}}$
16	NU	36	P3	56	$\overline{\text{HWR}}$	76	$\overline{\text{INT2}}$
17	NU	37	P2	57	HA0	77	$\overline{\text{INT1}}$
18	NU	38	P1	58	HA1	78	NU
19	NU	39	P0	59	CLKOUT	79	NU
20	EV _{DD}	40	EV _{DD}	60	EV _{DD}	80	EV _{DD}

注 $\overline{\text{WAKEUP}}$ 端子の機能はマスク・オプションで有効/無効を選択できます。

★ (3) 80ピン・プラスチック TQFP (ファインピッチ)(12×12)
 ・μ PD77115GK-9EU

図2-3 80ピン・プラスチック TQFP (ファインピッチ)(12×12)



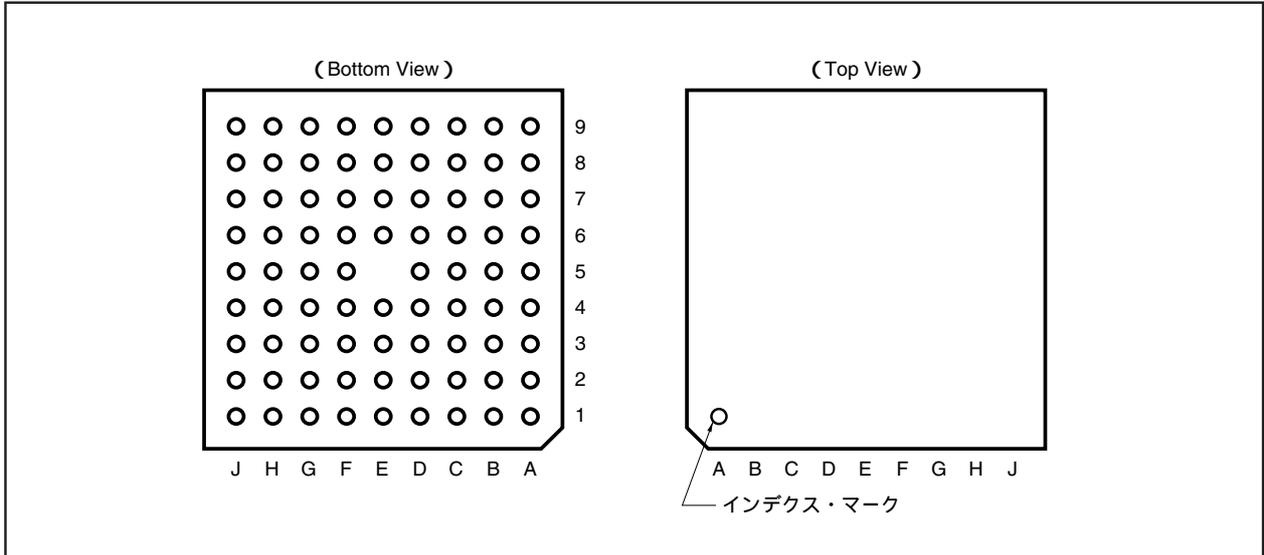
端子番号	端子名	端子番号	端子名	端子番号	端子名	端子番号	端子名
1	SI	21	EV _{DD}	41	HD0	61	TCK
2	NC	22	GND	42	GND	62	I.C.
3	SIEN/MCLK	23	HD13	43	EV _{DD}	63	TDI
4	SCK/BCLK	24	HD12	44	$\overline{\text{HRE}}$	64	TMS
5	SO	25	HD11	45	$\overline{\text{HWE}}$	65	$\overline{\text{TRST}}$
6	SOEN/LRCLK	26	HD10	46	$\overline{\text{HCS}}$	66	$\overline{\text{RESET}}$
7	P7/PLL3	27	HD9	47	$\overline{\text{HRD}}$	67	$\overline{\text{INT4}}$
8	GND	28	HD8	48	$\overline{\text{HWR}}$	68	$\overline{\text{INT3}}$
9	P6/PLL2	29	HD7	49	CLKOUT	69	$\overline{\text{INT2}}$
10	P5/PLL1	30	EV _{DD}	50	CLKIN	70	$\overline{\text{INT1}}$
11	P4/PLL0	31	GND	51	EV _{DD}	71	$\overline{\text{WAKEUP}}$
12	EV _{DD}	32	HD6	52	GND	72	IV _{DD}
13	P3	33	HD5	53	IV _{DD}	73	GND
14	P2	34	HD4	54	GND	74	SDCLK
15	P1	35	HD3	55	HA0	75	EV _{DD}
16	P0	36	HD2	56	HA1	76	GND
17	HD15	37	IV _{DD}	57	TDO	77	SDCR
18	GND	38	GND	58	I.C.	78	NC
19	NC	39	I.C.	59	I.C.	79	SDDAT
20	HD14	40	HD1	60	TICE	80	EV _{DD}

(4) 80ピン・プラスチック・ファインピッチ BGA (9×9)

・μ PD77111F1-x x x -CN1, μ PD77113AF1-x x x -CN1, μ PD77115F1-CN1

注意 これらの製品は、同じパッケージですが、端子配置が異なります。

図2-4 80ピン・プラスチック・ファインピッチ BGA (9×9)



μ PD77111F1-x x x -CN1 端子表

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	EV _{DD}	C3	NU	E6	$\overline{\text{HRE}}$	G8	HD4
A2	NU	C4	$\overline{\text{RESET}}$	E7	HD0	G9	HD5
A3	$\overline{\text{INT2}}$	C5	$\overline{\text{TRST}}$	E8	GND	H1	NU
A4	$\overline{\text{INT4}} / \overline{\text{WAKEUP}}$ 注	C6	TICE	E9	EV _{DD}	H2	NU
A5	IV _{DD}	C7	CLKIN	F1	NU	H3	SIEN1
A6	TCK	C8	HA0	F2	NU	H4	SOEN1
A7	IV _{DD}	C9	$\overline{\text{HWR}}$	F3	NU	H5	GND
A8	GND	D1	NU	F4	SIK1	H6	SI2
A9	EV _{DD}	D2	NU	F5	SOEN2	H7	P1
B1	NU	D3	NU	F6	P2	H8	GND
B2	GND	D4	$\overline{\text{INT1}}$	F7	HD1	H9	HD7
B3	NU	D5	TMS	F8	HD3	J1	EV _{DD}
B4	$\overline{\text{INT3}}$	D6	TDO	F9	HD2	J2	GND
B5	GND	D7	$\overline{\text{HCS}}$	G1	NU	J3	SCK1
B6	TDI	D8	$\overline{\text{HRD}}$	G2	NU	J4	SORQ1
B7	GND	D9	$\overline{\text{HWE}}$	G3	SI1	J5	IV _{DD}
B8	CLKOUT	E1	EV _{DD}	G4	SO1	J6	SCK2
B9	HA1	E2	GND	G5	SO2	J7	P3
C1	NU	E3	NU	G6	SIEN2	J8	P0
C2	NU	E4	NU	G7	HD6	J9	EV _{DD}

注 WAKEUP 端子の機能はマスク・オプションで有効 / 無効を選択できます。

μ PD77113AF1-x x x -CN1 端子表

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	-	C3	NU	E6	$\overline{\text{HCS}}$	G8	P1
A2	NU	C4	$\overline{\text{RESET}}$	E7	GND	G9	GND
A3	EV _{DD}	C5	TDI	E8	HD1	H1	NU
A4	$\overline{\text{INT3}}$	C6	TDO	E9	HD2	H2	NU
A5	GND	C7	CLKIN	F1	NU	H3	SCK1
A6	TMS	C8	HA0	F2	NU	H4	SOEN2
A7	GND	C9	EV _{DD}	F3	SOEN1	H5	SIEN2
A8	$\overline{\text{TRST}}$	D1	EV _{DD}	F4	GND	H6	P3
A9	-	D2	NU	F5	HD0	H7	P0
B1	NU	D3	$\overline{\text{INT2}}$	F6	SI2	H8	HD7
B2	NU	D4	NU	F7	HD3	H9	NU
B3	$\overline{\text{INT1}}$	D5	TCK	F8	HD6	J1	-
B4	$\overline{\text{INT4}} / \overline{\text{WAKEUP}}$ 注	D6	GND	F9	HD5	J2	NU
B5	IV _{DD}	D7	$\overline{\text{HWR}}$	G1	EV _{DD}	J3	SI1
B6	TICE	D8	$\overline{\text{HRD}}$	G2	GND	J4	SORQ1
B7	IV _{DD}	D9	EV _{DD}	G3	SIEN1	J5	SO2
B8	HA1	E1	NU	G4	SO1	J6	SCK2
B9	CLKOUT	E2	GND	G5	IV _{DD}	J7	EV _{DD}
C1	GND	E3	SIK1	G6	HD4	J8	NU
C2	NU	E4	NU	G7	P2	J9	-

注 $\overline{\text{WAKEUP}}$ 端子の機能はマスク・オプションで有効 / 無効を選択できます。

★ μ PD77115F1-CN1 端子表

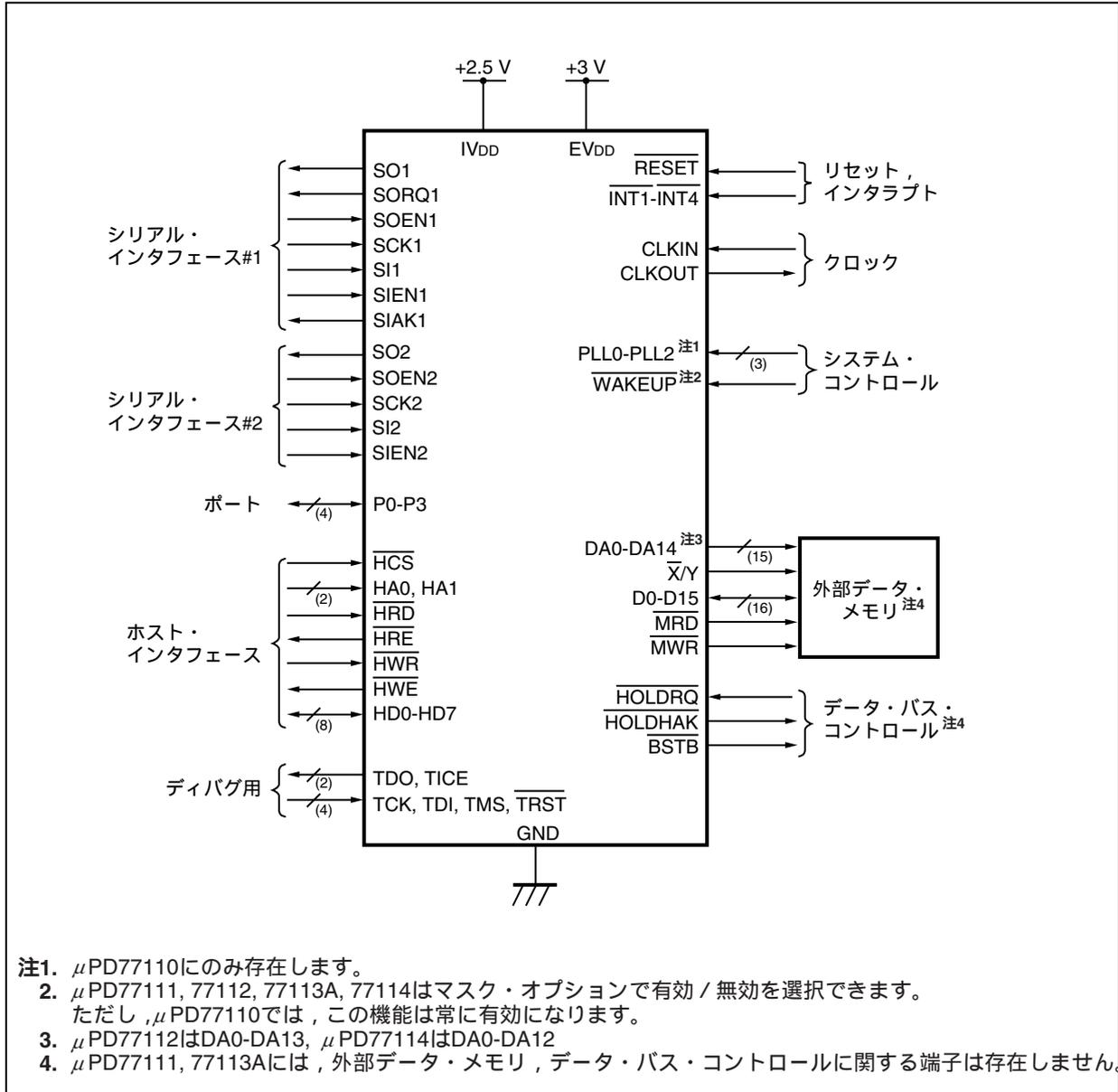
端子番号	端子名	端子番号	端子名	端子番号	端子名	端子番号	端子名
A1	EV _{DD}	C3	SDDAT	E6	GND	G8	$\overline{\text{HRE}}$
A2	NC	C4	GND	E7	$\overline{\text{HWR}}$	G9	EV _{DD}
A3	EV _{DD}	C5	$\overline{\text{INT3}}$	E8	EV _{DD}	H1	GND
A4	IV _{DD}	C6	$\overline{\text{TRST}}$	E9	CLKOUT	H2	EV _{DD}
A5	$\overline{\text{INT2}}$	C7	TICE	F1	EV _{DD}	H3	HD12
A6	$\overline{\text{RESET}}$	C8	TDO	F2	P0	H4	EV _{DD}
A7	TDI	C9	HA0	F3	P3	H5	GND
A8	I.C.	D1	SOEN/LRCLK	F4	HD9	H6	HD2
A9	I.C.	D2	P5/PLL1	F5	HD4	H7	IV _{DD}
B1	NC	D3	SO	F6	$\overline{\text{HRD}}$	H8	HD0
B2	SI	D4	P7/PLL3	F7	$\overline{\text{HWE}}$	H9	GND
B3	SDCR	D5	SDCLK	F8	CLKIN	J1	NC
B4	GND	D6	$\overline{\text{INT4}}$	F9	$\overline{\text{HCS}}$	J2	GND
B5	$\overline{\text{WAKEUP}}$	D7	IV _{DD}	G1	P1	J3	HD13
B6	INT1	D8	HA1	G2	HD15	J4	HD10
B7	TMS	D9	GND	G3	HD14	J5	HD7
B8	TCK	E1	P6/PLL2	G4	HD11	J6	HD6
B9	I.C.	E2	P4/PLL0	G5	HD8	J7	HD3
C1	SIEN/MCLK	E3	GND	G6	HD5	J8	GND
C2	SCK/BCLK	E4	P2	G7	HD1	J9	I.C.

2.2 端子構成

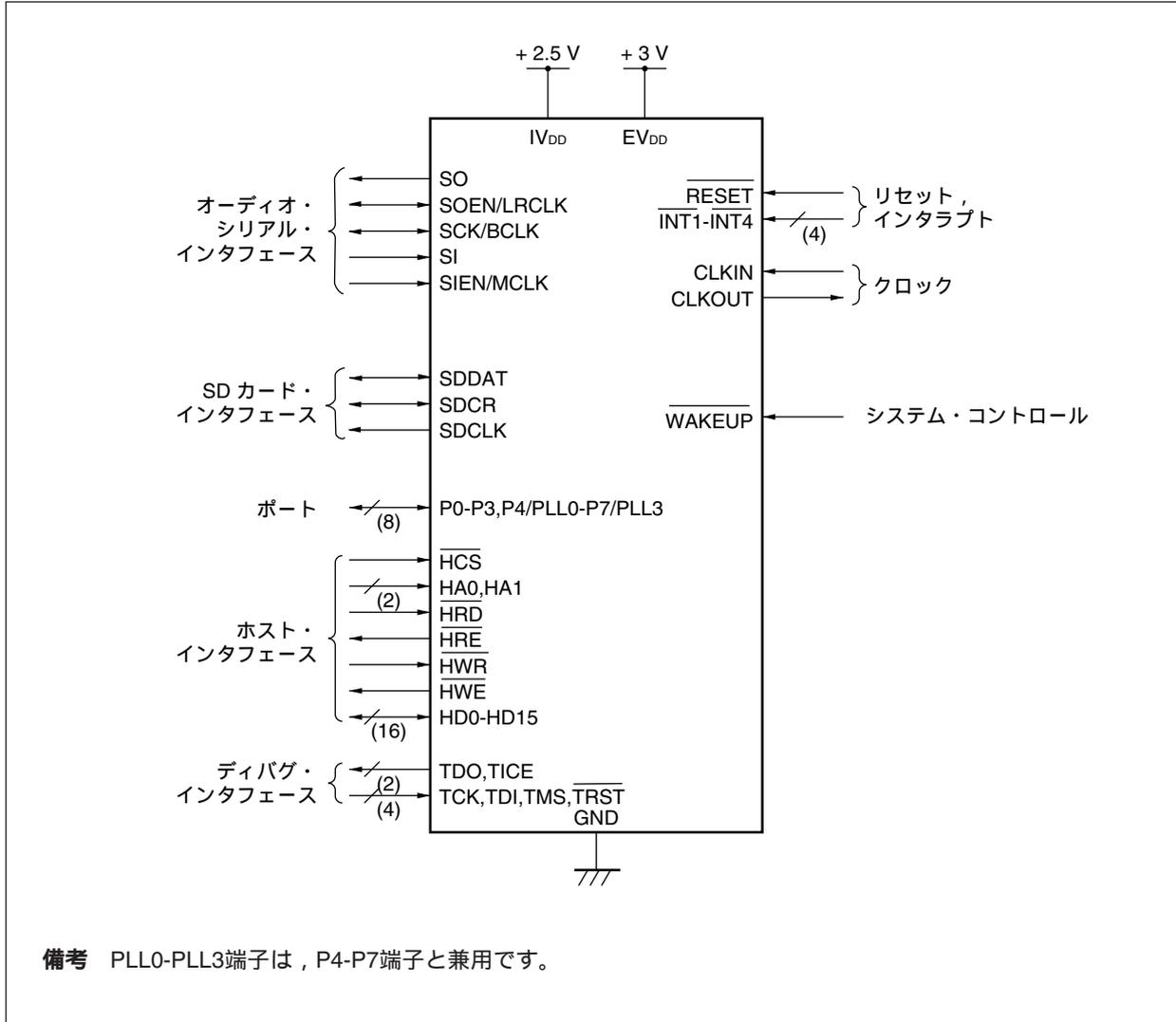
2.1 で示した端子接続を，機能別に分類して図 2 - 5 に示します。

図 2 - 5 μ PD77111 ファミリの端子構成図

(a) μ PD77110, 77111, 77112, 77113A, 77114



(b) μ PD77115



2.3 端子機能

2.3.1 μ PD77110, 77111, 77112, 77113A, 77114 の端子機能

パッケージによって端子番号が異なります。

また、同じ 80 ピン・ファインピッチ BGA であっても、 μ PD77111F1-x x x-CN1 と μ PD77113AF1-x x x-CN1 では、端子番号が異なります。ご使用になるパッケージ（品名）の欄をご覧ください。

(1) 電 源

端子名称	端子番号				入出力	機 能	兼用端子
	100 ピン TQFP	80 ピン TQFP	80 ピン FBGA				
			μ PD77111	μ PD77113A			
IV _{DD}	35, 77, 85	31, 63, 71	A5, A7, J5	B5,B7,G5	-	DSP コア用電源 (+2.5V) ・すべての端子を +2.5V 電源に接続します。	-
EV _{DD}	25, 50, 64, 75, 100	10,20, 40, 50, 60, 80	A1, A9, E1, E9, J1, J9	A3,C9,D1, D9,G1,J7	-	I/O 端子用電源 (+3.0V) ・すべての端子を +3.0V 電源に接続します。	-
GND	1,26,36, 51,65,76, 78, 86	1, 11, 21, 32,41,51, 61, 64, 72	A8, B2,B5, B7, E2, E8, H5, H8, J2	A5,A7,C1, D6,E2,E7, F4,G2,G9	-	接地 ・すべての端子を接地します。	-

備考 IV_{DD} と EV_{DD} の両電源は、同時に投入するようにしてください。

(2) 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
	100 ピン TQFP	80 ピン TQFP	80 ピン FBGA				
			μ PD77111	μPD77113A			
CLKIN	74	62	C7	C7	入力	クロック入力。 ・クロックは通常のデバイス動作では常に供給してください。ただし、ストップ・モードでは停止することもできます。	-
CLKOUT	73	59	B8	B9	出力	内部システム・クロック出力。 ・マスク・オプションにより CLKIN を逡倍、分周した信号を出力します。 詳細は 3.3.1 クロック・ジェネレータを参照してください。	-
$\overline{\text{RESET}}$	87	73	C4	C4	入力	内部システム・リセット信号入力。 ・デバイスをハードウェア的に初期化します。電源 ON 後必ず $\overline{\text{RESET}}$ 信号を入力してください。	-
PLL0	53	-	-	-	入力	PLL 逡倍率設定入力 (μ PD77110 の場合のみ) ・リセット入力時に次の設定に従って PLL の逡倍率を決定します。 PLL2:PLL1:PLL0 : 000 : PLL 逡倍率 × 1 を選択 001 : PLL 逡倍率 × 2 を選択 010 : PLL 逡倍率 × 3 を選択 : 111 : PLL 逡倍率 × 8 を選択 ・μ PD77111, 77112, 77113A, 77114 の場合は機能しません。	P2
PLL1	52	-	-	-	入力		P3
PLL2	49	-	-	-	入力		-
$\overline{\text{WAKEUP}}$	88	74	A4	B4	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。マスク・オプションによってこの端子の有効/無効を選択できます。 ・μ PD77110 の場合は常に有効のみの設定になります。	$\overline{\text{INT4}}$

(3) 割り込み

端子名称	端子番号				入出力	機 能	兼用端子
	100 ピン TQFP	80 ピン TQFP	80 ピン FBGA				
			μ PD77111	μPD77113A			
$\overline{\text{INT1}}$ - $\overline{\text{INT3}}$	91-89	77-75	D4,A3,B4	B3,D3,A4	入力	マスクブル外部割り込み入力。 ・立ち下がりエッジで検出されます。 ・割り込み競合した場合は各入力につき 1 レベルの記憶機能があります。	-
$\overline{\text{INT4}}$	88	74	A4	B4	入力		WAKEUP

(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
	100ピン TQFP	80ピン TQFP	80ピンFBGA				
			μ PD77111	μ PD77113A			
$\overline{X/Y}$	99	-	-	-	出力 (3S)	メモリ選択信号出力。 ・0: Xメモリを使用します。 ・1: Yメモリを使用します。 ・外部メモリは、同じにXメモリ空間およびYメモリ空間をアクセス対象とすることはできません。	-
DA0-DA14	16-2	-	-	-	入力 (3S)	外部データ・メモリのアドレス・バス。 ・外部メモリをアクセスします。 ・外部メモリをアクセスしないときは最後にアクセスした外部メモリのアドレスを出力し続けます。リセット後、外部メモリを一度もアクセスしていないときはロウ・レベル(0x0000)を出力し続けます。 ・ μ PD77112の場合はDA14、 μ PD77114の場合はDA14とDA13は機能しません。 NC(ノー・コネクション)となります。	-
D0-D15	34-27, 24-17	-	-	-	入出力 (3S)	16ビット・データ・バス。 ・外部メモリをアクセスします。	-
\overline{MRD}	97	-	-	-	出力 (3S)	リード出力。 ・外部メモリ・リード	-
\overline{MWR}	96	-	-	-	出力 (3S)	ライト出力。 ・外部メモリ・ライト	-
\overline{HOLDRQ}	92	-	-	-	入力	ホールド・リクエスト信号入力。 ・外部回路がデータ・メモリ・バスを使用したいときに、この信号をアクティブ(ロウ・レベル)にします。 ・0: ホールド・リクエスト ・1: ノー・リクエスト	-
\overline{BSTB}	94	-	-	-	出力	バス・ストロブ信号出力。 ・ μ PD77110,77112,77114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。 ・0: バスを使用している ・1: バスを使用していない	-
\overline{HOLDAK}	93	-	-	-	出力	ホールド・アクノリッジ信号出力。 ・外部回路が \overline{HOLDRQ} を要求した場合に、外部回路にデータ・メモリ・バスの使用を許可したときアクティブ(ロウ・レベル)になります。 ・0: バスの使用を許可する ・1: バスの使用を許可しない	-

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。

$\overline{X/Y}$, DA0-DA14, \overline{MRD} , \overline{MWR} : バス開放時 (\overline{HOLDAK} = ロウ・レベル)

D0-D15: 外部データ・メモリ非アクセス時, およびバス開放時 (\overline{HOLDAK} = ロウ・レベル)

(5) 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
	100 ピン TQFP	80 ピン TQFP	80 ピン FBGA				
			μ PD77111	μPD77113A			
SCK1	39	24	J3	H3	入力	シリアル 1 用クロック入力。 ・シリアル 1 の入力 / 出力関連信号は、この信号に同期してサンプリングされます。	-
SORQ1	42	27	J4	J4	出力	シリアル出力 1 リクエスト信号出力。 ・シリアル・データの出力に先立ちアクティブ (ハイ・レベル) になります。 ・0: シリアル・データ出力を受け付ける準備ができていない。 ・1: シリアル・データ出力を受け付ける準備ができた。	-
SOEN1	43	28	H4	F3	入力	シリアル出力 1 イネーブル信号入力。 ・外部回路がシリアル・データ出力を受け付ける準備ができたとき、アクティブ (ハイ・レベル) にして μ PD77111 ファミリに知らせます。	-
SO1	41	26	G4	G4	出力 (3S)	シリアル・データ出力 1。 ・SCK1 の立ち上がりに同期して出力されます。	-
SIEN1	38	23	H3	G3	入力	シリアル入力 1 イネーブル信号入力。 ・外部回路が、シリアル・データ入力を供給する準備ができたとき、アクティブ (ハイ・レベル) にして μ PD77111 ファミリに知らせます。	-
SI1	37	22	G3	J3	入力	シリアル・データ入力 1。 ・SCK1 の立ち下がりに同期して出力されます。	-
SI AK1	40	25	F4	E3	出力	シリアル入力 1 アクノリッジ信号出力。 ・シリアル・データ入力が可能であることを外部回路に知らせます。 ・0: シリアル・データ入力を受け付ける準備ができていない。 ・1: シリアル・データ入力を受け付ける準備ができた。	-
SCK2	46	33	J6	J6	入力	シリアル 2 用クロック入力。 ・シリアル 2 の入力 / 出力関連信号は、この信号に同期してサンプリングされます。	-
SOEN2	44	29	F5	H4	入力	シリアル出力 2 イネーブル信号入力。 ・外部回路がシリアル・データ出力を受け付ける準備ができたとき、アクティブ (ハイ・レベル) にして μ PD77111 ファミリに知らせます。	-
SO2	45	30	G5	J5	出力 (3S)	シリアル・データ出力 2。 ・SCK2 の立ち上がりに同期して出力されます。	-
SIEN2	47	34	G6	H5	入力	シリアル入力 2 イネーブル信号入力。 ・外部回路が、シリアル・データ入力を供給する準備ができたとき、アクティブ (ハイ・レベル) にして μ PD77111 ファミリに知らせます。	-
SI2	48	35	H6	F6	入力	シリアル・データ入力 2。 ・SCK2 の立ち下がりに同期して出力されます。	-

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 (RESET) 入力により、ハイ・インピーダンス状態になります。

(6) 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
	100ピン TQFP	80ピン TQFP	80ピンFBGA				
			μ PD77111	μ PD77113A			
HA1	72	58	B9	B8	入力	HD7-HD0 がアクセスするレジスタを指定します。 <ul style="list-style-type: none"> ・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	71	57	C8	C8	入力	HD7-HD0 がアクセスするレジスタを指定します。 <ul style="list-style-type: none"> ・1: HST, HDT (in), HDT (out) のビット 15-8 をアクセスします。 ・0: HST, HDT (in), HDT (out) のビット 7-0 をアクセスします。 	-
$\overline{\text{HCS}}$	68	54	D7	E6	入力	チップ・セレクト入力。	-
$\overline{\text{HRD}}$	69	55	D8	D8	入力	ホスト・リード入力。 <ul style="list-style-type: none"> ・ホストからのリード・パルスで, データはこの信号の立ち下がりに同期して出力されます。 	-
$\overline{\text{HWR}}$	70	56	C9	D7	入力	ホスト・ライト入力。 <ul style="list-style-type: none"> ・ホストからのライト・パルスで, データはこの信号の立ち下がりに同期して出力されます。 	-
$\overline{\text{HRE}}$	66	52	E6	-	出力	ホスト・リード・イネーブル出力。 <ul style="list-style-type: none"> ・この信号がアクティブ (ロウ・レベル) のとき, ホストからのリード・アクセスが可能であることを示します。 ・0: ホストからのリード・アクセス可能。 ・1: ホストからのリード・アクセス不可能。 	-
$\overline{\text{HWE}}$	67	53	D9	-	出力	ホスト・ライト・イネーブル出力。 <ul style="list-style-type: none"> ・この信号がアクティブ (ロウ・レベル) のとき, ホストからのライト・アクセスが可能であることを示します。 ・0: ホストからのライト・アクセス可能。 ・1: ホストからのライト・アクセス不可能。 	-
HD0-HD7	63-56	49-42	E7,F7,F9, F8,G8,G9, G7,H9	F5,E8,E9, F7,G6,F9, F8,H8	入出力 (3S)	8ビット・ホスト・データ・バス。	-

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。

(7) 入出力ポート

端子名称	端子番号				入出力	機 能	兼用端子
	100 ピン TQFP	80 ピン TQFP	80 ピン FBGA				
			μ PD77111	μPD77113A			
P0	55	39	J8	H7	入出力	汎用入出力ポート。 ・各端子は PCD(ポート・コマンド・レジスタ)によって独立に入出力設定可能です。入出力とも、PCD, PDT (ポート・データ・レジスタ) を介して実行します。	-
P1	54	38	H7	G8	入出力		-
P2	53	37	F6	G7	入出力		PLL0 ^注
P3	52	36	J7	H6	入出力		PLL1 ^注

注 μ PD77110 の場合のみ。μ PD77111, 77112, 77113A, 77114 の場合は兼用端子ではありません。

(8) デバッ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
	100 ピン TQFP	80 ピン TQFP	80 ピン FBGA				
			μ PD77111	μPD77113A			
TDO	79	65	D6	C6	出力	デバッグ用	-
TICE	80	66	C6	B6	出力		-
TCK	81	67	A6	D5	入力		-
TDI	82	68	B6	C5	入力		-
TMS	83	69	D5	A6	入力		-
TRST	84	70	C5	A8	入力		-

(9) その他

端子名称	端子番号				入出力	機 能	兼用端子
	100 ピン TQFP	80 ピン TQFP	80 ピン FBGA				
			μ PD77111	μPD77113A			
I.C.	98	-	-	-	-	内部接続端子です。オープンにしてください。	-
NU	(95)	2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 14, 15, 16, 17, 18, 19, 78, 79	A2,B1,B3, C1,C2,C3, D1,D2,D3, E3,E4,F1, F2,F3,G1, G2,H1,H2	A2,B1,B2, C2,C3,D2, D4,E1,E4, F1,F2,H1, H2,H9,J2, J8	-	未機能端子です。EV _{DD} に接続してください。 100 ピン TQFP の 95 ピン ・ μ PD77110GC : NU ・ μ PD77112GC : NU ・ μ PD77114GC : NC	-
NC	(2,3,49, 95)	-	-	-	-	未接続端子です。オープンにしてください。 ・ μ PD77110GC : NC なし ・ μ PD77112GC : 2, 49 ピン ・ μ PD77114GC : 2, 3, 49, 95 ピン	-
-	-	-	-	A1,A9,J1, J9	-	半田付け強化用の端子です。端子処理は不要です。	-

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 PD77110, 77111, 77112, 77113A, 77114 の正常な動作は保証されません。

★ 2.3.2 μ PD77115の端子機能

パッケージによって端子番号が異なりますので、ご使用になるパッケージの欄をご覧ください。

(1) 電 源

端子名称	端子番号		入出力	機 能	兼用端子
	80ピンFBGA	80ピンTQFP			
IV _{DD}	A4, D7, H7	37, 53, 72	-	DSPコア用電源(+2.5V)	-
EV _{DD}	A1, A3, E8, F1, G9, H2, H4	12, 21, 30, 43, 51, 75, 80	-	I/O端子用電源(+3V)	-
GND	B4, C4, D9, E3, E6, H1, H5, H9, J2, J8	8, 18, 22, 31, 38, 42, 52, 54, 73, 76	-	接地	-

(2) 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
	80ピンFBGA	80ピンTQFP			
CLKIN	F8	50	入力	クロック入力	-
CLKOUT	E9	49	出力	内部システム・クロック出力	-
$\overline{\text{RESET}}$	A6	66	入力	内部システム・リセット信号入力	
PLL0-PLL3	E2, D2, E1, D4	11-9, 7	入力	PLL 逡倍率設定端子 PLL3-PLL0: 0000 : ×16 0001 : ×1 0010 : ×2 0011 : ×3 0100 : ×4 0101 : ×5 0110 : ×6 0111 : ×7 1000 : ×8 1001 : ×9 1010 : ×10 1011 : ×11 1100 : ×12 1101 : ×13 1110 : ×14 1111 : ×15	P4-P7
$\overline{\text{WAKEUP}}$	B5	71	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) 割り込み

端子名称	端子番号		入出力	機 能	兼用端子
	80ピンFBGA	80ピンTQFP			
$\overline{\text{INT1}}$ - $\overline{\text{INT4}}$	B6, A5, C5, D6	70-67	入力	マスキブル外部割り込み入力 ・立ち下がりエッジで検出されます。	-

(4) 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
	80ピンFBGA	80ピンTQFP			
SCK/BCLK	C2	4	入出力	シリアル・クロック入出力 SCK: シリアル・クロック入力(入力) BCLK: シリアル・クロック(入出力)	-
SOEN/ LRCLK	D1	6	入出力	シリアル出力イネーブル/レフト・ライト・クロック入出力 SOEN: 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)	-
SO	D3	5	出力 (3S)	シリアル・データ出力	-
SIEN/MCLK	C1	3	入力	シリアル入力イネーブル/マスタ・クロック入力 SIEN: シリアル入力イネーブル MCLK: マスタ・クロック(マスタ・モード時)	-
SI	B2	1	入力	シリアル・データ入力	-

備考 表中入出力欄に“3S”を付記した端子は、データ送完了時、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。

(5) SDカード・インタフェース

端子名称	端子番号		入出力	機能	兼用端子
	80ピンFBGA	80ピンTQFP			
SDCLK	D5	74	出力	SDカード・クロック出力	-
SDCR	B3	77	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。	-
SDDAT	C3	79	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。	-

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。

(6) 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
	80ピンFBGA	80ピンTQFP			
HA1	D8	56	入力	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	C9	55	入力	HD15-HD0がアクセスするレジスタを指定します。 ・1: HST, HDT(in), HDT(out)のビット15-8をアクセスします。 ・0: HST, HDT(in), HDT(out)のビット7-0をアクセスします。 ・8ビット・モード時のみ有効です。16ビット・モード時は無効になります。	-
$\overline{\text{HCS}}$	F9	46	入力	チップ・セレクト入力	-
$\overline{\text{HRD}}$	F6	47	入力	ホスト・リード入力	-
$\overline{\text{HWR}}$	E7	48	入力	ホスト・ライト入力	-
$\overline{\text{HRE}}$	G8	44	出力	ホスト・リード・イネーブル出力	-
$\overline{\text{HWE}}$	F7	45	出力	ホスト・ライト・イネーブル出力	-
HD0-HD15	H8, G7, H6, J7, F5, G6, J6, J5, G5, F4, J4, G4, H3, J3, G3, G2	41, 40, 36-32, 29-23, 20, 17	入出力 (3S)	16ビット・ホスト・データ・バス	-

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。

(7) 入出力ポート

端子名称	端子番号		入出力	機能	兼用端子
	80ピンFBGA	80ピンTQFP			
P0	F2	16	入出力	汎用入出力ポート	-
P1	G1	15	入出力		-
P2	E4	14	入出力		-
P3	F3	13	入出力		-
P4	E2	11	入出力		PLL0
P5	D2	10	入出力		PLL1
P6	E1	9	入出力		PLL2
P7	D4	7	入出力		PLL3

(8) デバッ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
	80ピンFBGA	80ピンTQFP			
TDO	C8	57	出力	デバッグ用	-
TICE	C7	60	出力		-
TCK	B8	61	入力		-
TDI	A7	63	入力		-
TMS	B7	64	入力		-
$\overline{\text{TRST}}$	C6	65	入力		-

(9) その他

端子名称	端子番号		入出力	機能	兼用端子
	80ピンFBGA	80ピンTQFP			
I.C.	A8, A9, B9, J9	39, 58, 59, 62	-	内部接続端子です。オープンにしてください。	-
NC	A2, B1, J1	2, 19, 78	-	未接続端子です。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 PD77115 の正常な動作は保証されません。

2.4 未使用端子の処理について

実装時に未使用の端子は、次の表のとおりに取り扱ってください。

表 2 - 1 機能端子の処理

端子	入出力	推奨接続方法
$\overline{\text{INT1}} - \overline{\text{INT4}}$	入力	EV _{DD} に接続してください。
$\overline{\text{X}} / \text{Y}$	出力	オープンにしてください。
DA0-DA14	出力	
D0-D15 ^{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。
$\overline{\text{MRD}}$, $\overline{\text{MWR}}$	出力	オープンにしてください。
$\overline{\text{HOLDRQ}}$	入力	EV _{DD} に接続してください。 ・ μ PD77113A, 77114 は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)。
$\overline{\text{BSTB}}$, $\overline{\text{HOLDAK}}$	出力	オープンにしてください。
SCK1, SCK2	入力	EV _{DD} または GND に接続してください。
SCLK/BCLK	入出力	
SI, SI1, SI2	入力	
SIEN1, SIEN2	入力	GND に接続してください。
SIEN/MCLK	入力	
SOEN1, SOEN2	入出力	
SOEN/LRCLK	入出力	
SORQ1	出力	オープンにしてください。
SO, SO1, SO2	出力	
SDCLK	出力	
SIK1	出力	
HA0, HA1	入力	
$\overline{\text{HCS}}$, $\overline{\text{HRD}}$, $\overline{\text{HWR}}$	入力	EV _{DD} に接続してください。
$\overline{\text{HRE}}$, $\overline{\text{HWE}}$	出力	オープンにしてください。
HD0-HD15 ^{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。
SDCR	入出力	
SDDAT	入出力	
P0-P3	入出力	
TCK	入力	
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)。
$\overline{\text{TRST}}$	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)。
$\overline{\text{WAKEUP}}$	入力	EV _{DD} に接続してください。
CLKOUT	出力	オープンにしてください。

注1. 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。

2. $\overline{\text{HCS}}$, $\overline{\text{HRD}}$, $\overline{\text{HWR}}$ がハイ・レベル固定ならばオープン可能です。

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。

備考 プルアップ、プルダウン抵抗は 10 k ~ 100 k Ω を推奨します。

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。
NU	-	μ PD77110, 77111, 77112 は、EV _{DD} に接続してください。 μ PD7713A, 77114 は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。
NC	-	オープンにしてください。

第3章 アーキテクチャ

この章では、 μ PD77111 ファミリのアーキテクチャを物理的な機能ブロックに区分し、それらの機能ブロックをさらにトップダウン式に焦点を絞って解説します。最初に全体の構成について、後の節でさらに詳細な部分（ユニット）について説明します。 μ PD77115 は、従来の μ PD77111 ファミリーにペリフェラル機能が追加されています。 μ PD77115 固有のペリフェラル機能については、**第5章 μ PD77115 の概要**、**第6章 μ PD77115 のペリフェラル**を参照してください。

注意 特に指定のないかぎり、この章では従来の μ PD77111 ファミリー（ μ PD77110, 77111, 77112, 77113A, 77114）について説明しています。

μ PD77115 をご使用になる場合は、**第5章 μ PD77115 の機能概要**から**第6章 μ PD77115 のペリフェラル**をお読みください。

3.1 全体ブロック構成

ここでは、 μ PD77111 ファミリー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）

3.2 バスを参照してください。

システム制御ユニット

3.3 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

3.4 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

3.5 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

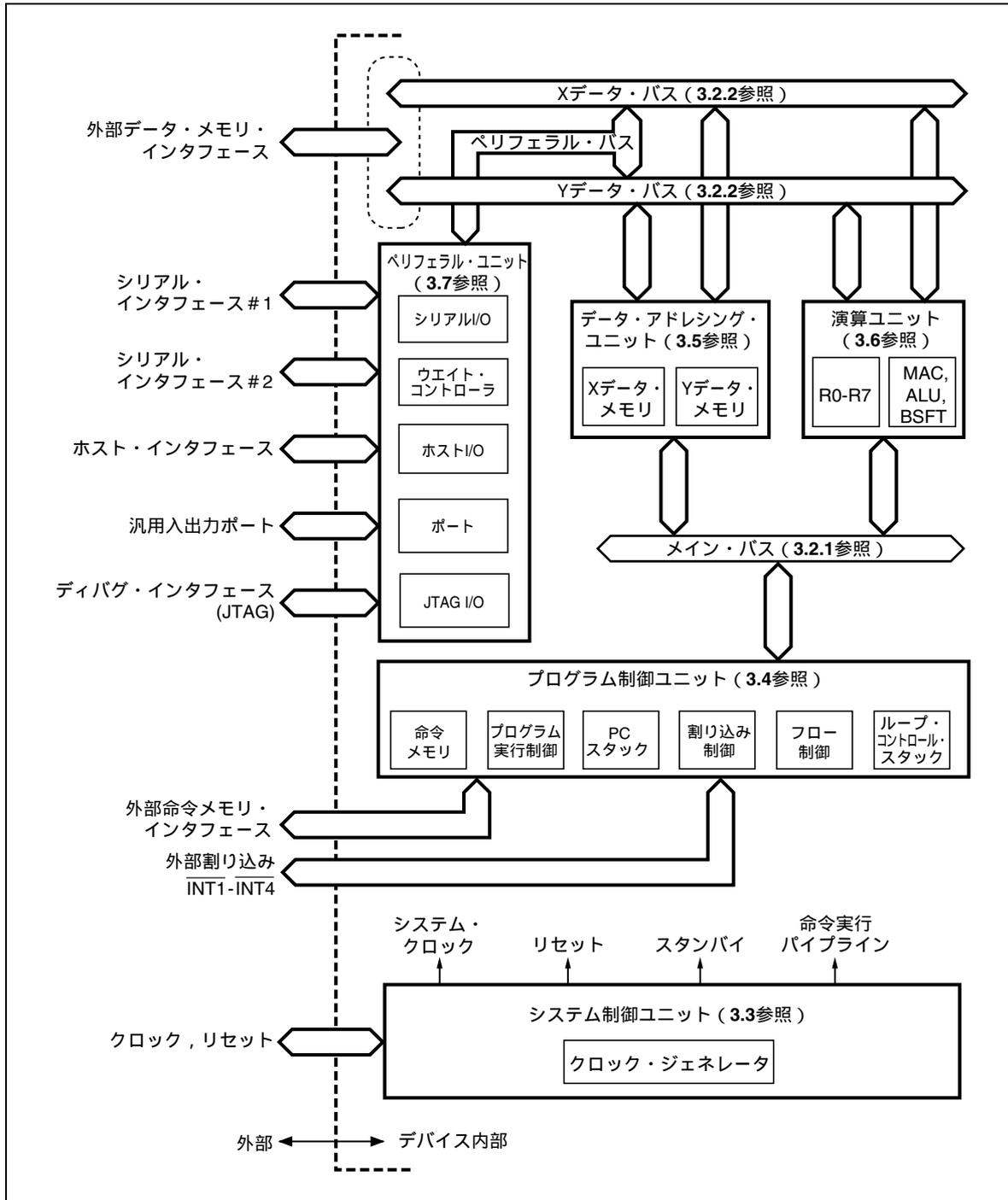
3.6 演算ユニットを参照してください。

ペリフェラル・ユニット

3.7 ペリフェラル・ユニットを参照してください。

全体ブロック構成を図3-1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3-1 全体ブロック構成



3.2 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 PD77111 ファミリには次の3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

3.2.1 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 3-1 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、3.6.2 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 3-1 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 3 - 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 3 - 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	

3.2.2 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 **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。

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次のようになります。

表3-2 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メモリ対象)
- 部分ロード/ストア命令 (X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メモリ対象)
- 即値インデクス・ロード/ストア命令 (X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メモリまたはY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3-3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 3 - 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 バス方向のみ
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 3.6.2 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 メモリ対象)
- 部分ロード/ストア命令 (Y 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 メモリ対象)
- 即値インデクス・ロード/ストア命令 (Y 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 メモリまたは Y 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 3 - 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 3 - 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 バス方向のみ
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 3.6.2 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) ペリフェラル・バス

内蔵ペリフェラル・レジスタと、X データ・バスおよび Y データ・バスを接続する 16 ビットのバスです。ペリフェラル・レジスタは X/Y メモリ空間に共通にマッピングされており、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 (ペリフェラル・レジスタ対象)
- 部分ロード/ストア命令 (ペリフェラル・レジスタ対象)
- ダイレクト・アドレッシング・ロード/ストア (ペリフェラル・レジスタ対象)
- 即値インデクス・ロード/ストア命令 (ペリフェラル・レジスタ対象)

ペリフェラル・バスの詳細については、3.7 ペリフェラル・ユニットを参照してください。

- 注意 1.** 内蔵ペリフェラルは、X メモリまたは Y 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。
- 2.** 1 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。

3.3 システム制御ユニット

μ PD77111 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

3.3.1 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。

(1) μ PD77111, 77112, 77113A, 77114 の場合

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図ることができます(この機能はマスク・オプションにより無効とすることもできます)。外部クロックは PLL によって逡倍され、出力分周器によって分周されます。また、ホールド分周器によってホールド・モード時の低クロック設定もできます。

逡倍率と分周率はマスク・オプションで設定できます。このとき内部システム・クロックとの周波数比は、次のように選択設定できます(PLL 逡倍率 m , 出力分周率 n , ホールド分周率を l とするとき)。

1 (外部): m/n (内部)

1 (外部): $m/n/l$ (内部): HALT モード時

PLL 制御回路では、入力されたクロックを 1 倍から 16 倍の整数逡倍します。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マスク・オプションで指定します。

出力分周器では PLL によって逡倍されたクロックを $1/1$ 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 DSP の動作周波数内に収まるように、出力分周率のマスク・オプションを指定します。

ホールド分周器はホールド・モード時のみ機能し、出力分周器のクロックを $1/1$ から $1/16$ の整数分の 1 に分周して内部に供給します。必要な分周を行えるように HALT 分周率のマスク・オプションを指定します。

DSP 内部に供給されるクロック(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”を選択できます。必要に応じてマスク・オプションを指定します。

また、出力分周率として奇数の値(1を除く)を指定した場合、CLKOUT 端子から出力されるクロックはハイ・レベル幅が通常動作時の 1 サイクル分のクロックとなります(デュティ比が 50%のクロックではありません)。

クロック回路を図 3-2 に、そのタイミングを図 3-3 に示します。

マスク・オプションの発注形式については、**A.2.1 クロック制御に関するオプション**を参照してください。

図3-2 μPD77111 ファミリのクロック回路

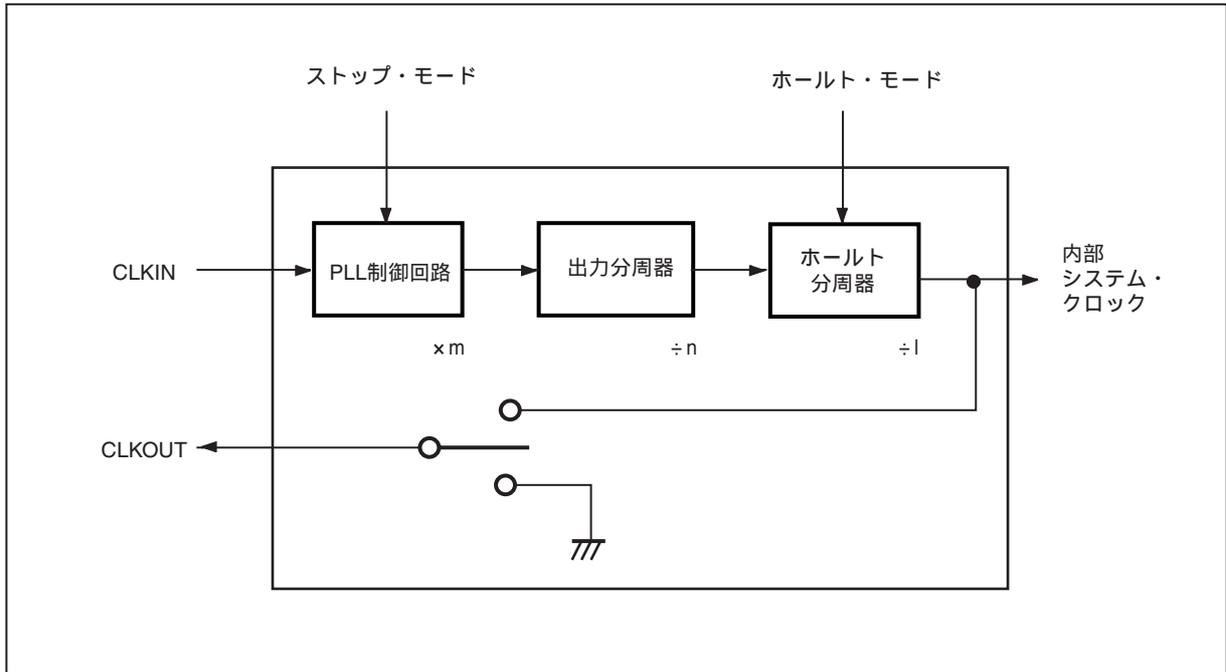
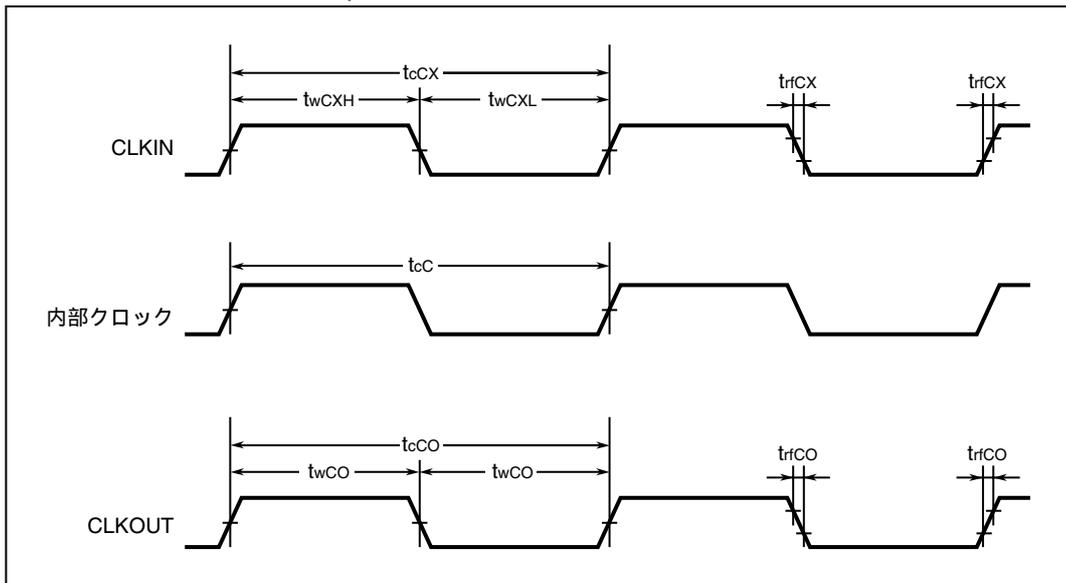


図3-3 μPD77111 ファミリのクロック・タイミング



★ (2) μ PD77110, 77115 の場合

回路構成はほかの μ PD77111 ファミリと同等ですが、 μ PD77110, 77115 の場合はマスク・オプションが存在しないため通倍率、分周率をマスク・オプションによって設定できません。

μ PD77110 は、PLL 通倍率 m の設定は外部端子 (PLL0-PLL2) によって行います。通倍率の設定値は 1 倍から 8 倍までの整数通倍になります。また、出力分周率 n は 1 固定に、ホールド分周率 l は 8 固定になっています。PLL の通倍率設定には外部端子 PLL0-PLL2 を使用します。PLL0, PLL1 は汎用入出力ポート P2, P3 と兼用になっており、設定時のみ PLL 設定端子として機能します。また、CLKOUT 端子出力を有効/無効とするオプションは“有効”固定となっています。

μ PD77115 は、PLL 通倍率 m の設定は外部端子 (PLL0-PLL3) によって行います。通倍率の設定値は 1 倍から 16 倍までの整数通倍になります。PLL の通倍率設定には外部端子 PLL0-PLL3 を使用します。PLL0-PLL3 は汎用入出力ポート P4-P7 と兼用になっており、設定時のみ PLL 設定端子として機能します。また、CLKOUT 端子出力を有効/無効とするオプションはレジスタによって設定します (5.5 クロック制御を参照)。

端子設定値と通倍率の関係は表 3 - 5 のようになります。

表 3 - 5 PLL 通倍率設定

★

(a) μ PD77110 の PLL 通倍率設定

端子設定値			通倍率 (m)
PLL2	PLL1	PLL0	
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

(b) μ PD77115 の PLL 通倍率設定

端子設定値				通倍率 (m)
PLL3	PLL2	PLL1	PLL0	
0	0	0	0	16
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
:	:	:	:	:
1	1	1	0	14
1	1	1	1	15

(3) PLL の初期化

電源投入時は PLL の初期化が必要です。3.3.2 リセット機能を参照してください。

(4) スタンバイ・モード時の動作クロック

ホールド・モードおよびストップ・モード時のシステム・クロックの動作状態は次のとおりです。

製品名	ストップ・モード	ホールド・モード
μ PD77110	停止	外部クロックの $m/8$ 倍
μ PD77111, 77112, 77113A, 77114,	停止	外部クロックの $m/n/l$ 倍
μ PD77115	停止	外部クロックの m/n 倍

★

備考 m : PLL 通倍率, n : 出力分周率, l : ホールド分周率

3.3.2 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・リセットします。リセットの目的は、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です。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 3 - 6 から表 3 - 8 に、また、図 3 - 4 にリセット・タイミングを示します。

それぞれのブート動作によって、各ピンおよび各レジスタの値がどう変化するかについては、第 4 章 **ブート機能** を参照してください。

電源投入時は、電源電圧が動作電圧に達してから、 $\overline{\text{RESET}}$ 端子がインアクティブ (ハイ・レベル) の状態で入力クロックの 4 クロック入力後にアクティブ (ロウ・レベル) にする必要があります。すなわち、パワーオン・リセット機能はありません。また、電源投入時は PLL の初期化が必要です。

表 3 - 6 初期化対象 CPU レジスタと初期値

レジスタ名	初期値	説 明
SR	0xF000	要因別割り込みはすべてイネーブルで、割り込み全体の許可状態は、現在と過去すべてのレベルに対してディスエーブルです。また、ループ命令実行中でないことを示します。
PC	0	0 番地はブート領域であり、ブート処理終了後 0x200 番地に分岐します。したがって、ユーザ領域としてのリセット・エントリは、0x200 番地となります。
SP	0	-
LC	0b1xxx xxxx xxxx xxxx	ループ命令実行中でないことを示します。カウント値そのものは不定です。
LSP	0	-
RC	0b1xxx xxxx xxxx xxxx	リピート命令実行中でないことを示します。カウント値そのものは不定です。
EIR	0xFFFF	割り込み全体への許可状態について、現在と過去すべてのレベルに対してディスエーブルであることを示します。
ESR	0	-

表 3 - 7 初期化対象メモリ・マップト・レジスタと初期値

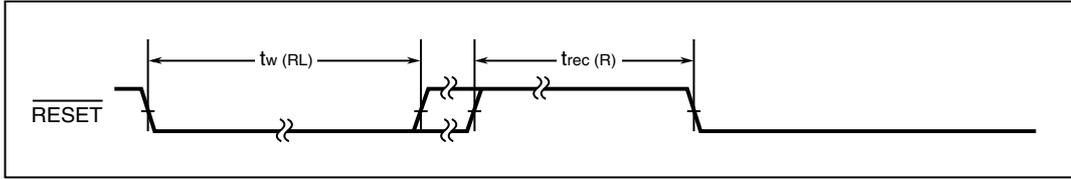
レジスタ名	初期値	説明
SST1, SST2	0x0002	シリアル・インタフェースは次のように初期化されます。 入力, 出力とも MSB ファースト 入力, 出力とも 16 ビット長 SDT のロード/ストアにウエイトを使用しない 状態遷移モード SDT ロード/ストアのエラー・フラグ・クリア SDT へのデータ・ストア許可 SDT からロードするデータなし
PCD	0x0000	I/O ポートは次のように初期化されます。 ビット操作しない モード設定しない
HST	0x0301	ホスト・インタフェースは次のように初期化されます。 HDT アクセスにウエイトを使用しない HRE 機能, HWE 機能とも禁止 UF0, UF1 ともゼロ・クリア ホスト・リード, ホスト・ライトのエラー・フラグ・クリア HDT ロード/ストアのエラー・フラグ・クリア ホストからのリード禁止 ホストからのライト許可

表 3 - 8 初期化対象端子と初期状態

端子名	初期状態
\overline{X}/Y	ロウ・レベル出力 [※]
DA0-DA14	
D0-D15	ハイ・インピーダンス
\overline{MRD} , \overline{MWR} , \overline{BSTB}	ハイ・レベル出力 [※]
SORQ1, SIAK1	ロウ・レベル出力
SO1, SO2	ハイ・インピーダンス
\overline{HRE} , \overline{HWE}	ハイ・レベル出力
P0-P3	入力状態
TICE	ロウ・レベル出力

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

図3-4 リセット・タイミング



(2) PLLの初期化

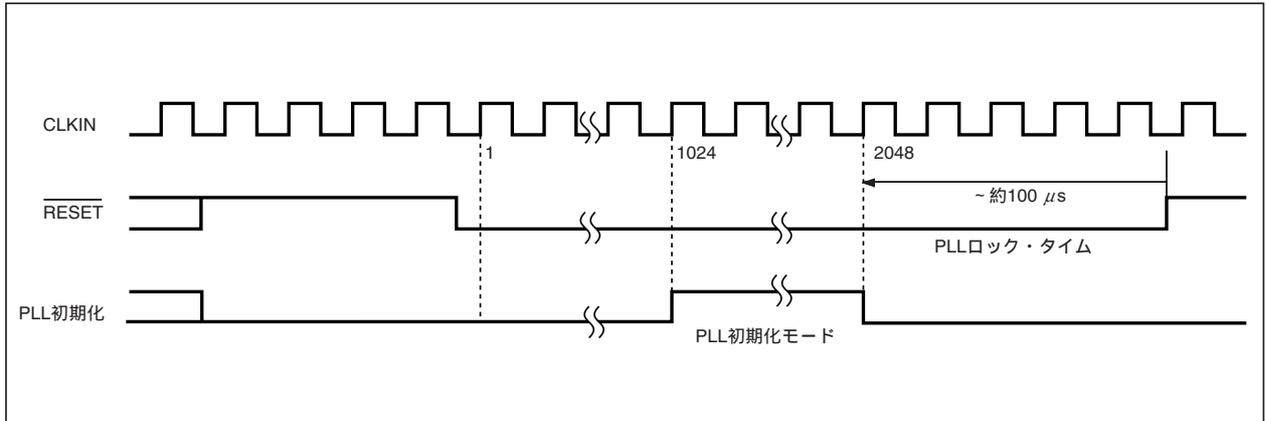
PLLは、 \overline{RESET} 端子がアクティブ(ロウ・レベル)になってから入力クロックの1024クロック目で初期化が開始されます。さらに初期化に1024クロックが必要となり、それからPLLがロックするまでに100 μs かかります。その後、 \overline{RESET} 端子をインアクティブ(ハイ・レベル)とすることで、マスク・オプション($\mu PD77111, 77112, 77113A, 77114$), あるいは外部端子設定($\mu PD77110$ に従ったPLLの設定値でDSPが動作します。図3-5にPLL設定タイミングを示します。

$\mu PD77110$ の場合、PLL0-PLL2の値はPLL初期化モードに入る前に固定し、PLL初期化モードの期間中保持しておく必要があります。

PLL初期化モードにはいる前にリセットをインアクティブにする場合は通常のリセットとなります(PLLの初期化は行われません)。

なお、PLLの初期化を行った場合は必ずブートアップ処理を行い、内部RAMの再初期化を行う必要があります。

図3-5 PLL設定タイミング



注意 PLL初期化モード中およびPLLロック期間中はリセットをインアクティブにしないでください。

3.3.3 パイプライン・アーキテクチャ

μ PD77111 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 PD77111 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読します。

E：実行サイクル

解読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 PD771110 が 65 MHz クロックで動作する場合、1インストラクション・サイクルは約 15.4 ns です。

μ PD77111, 77112, 77113A, 77114, 77115 が 75 MHz クロックで動作する場合、1インストラクション・サイクルは約 13.3 ns です。

次にパイプライン処理のイメージを示します。

図3-6(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3-6(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3-6 パイプライン・イメージ (1/2)

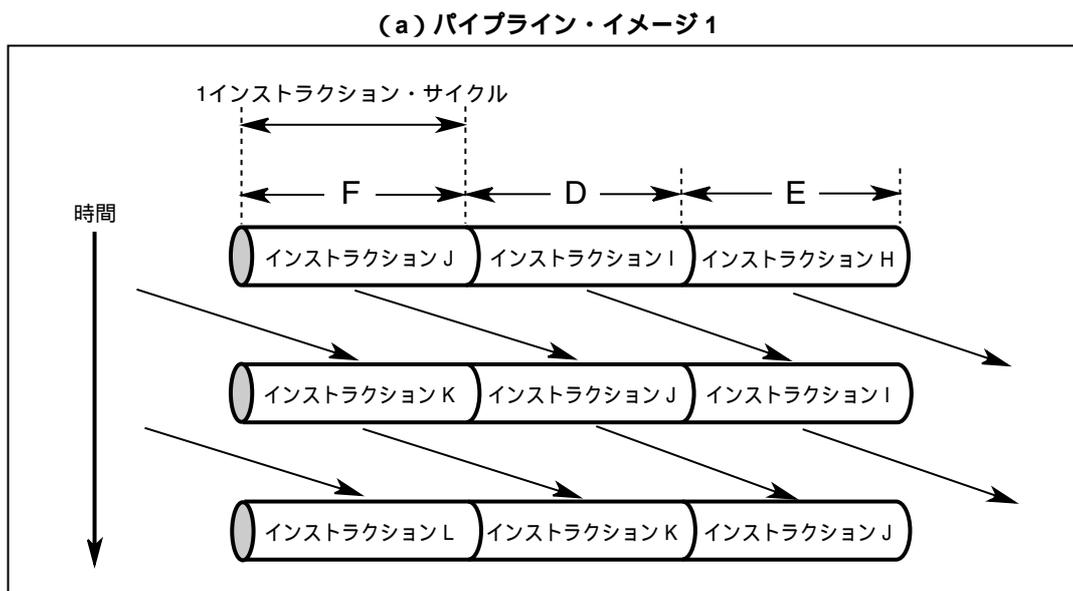
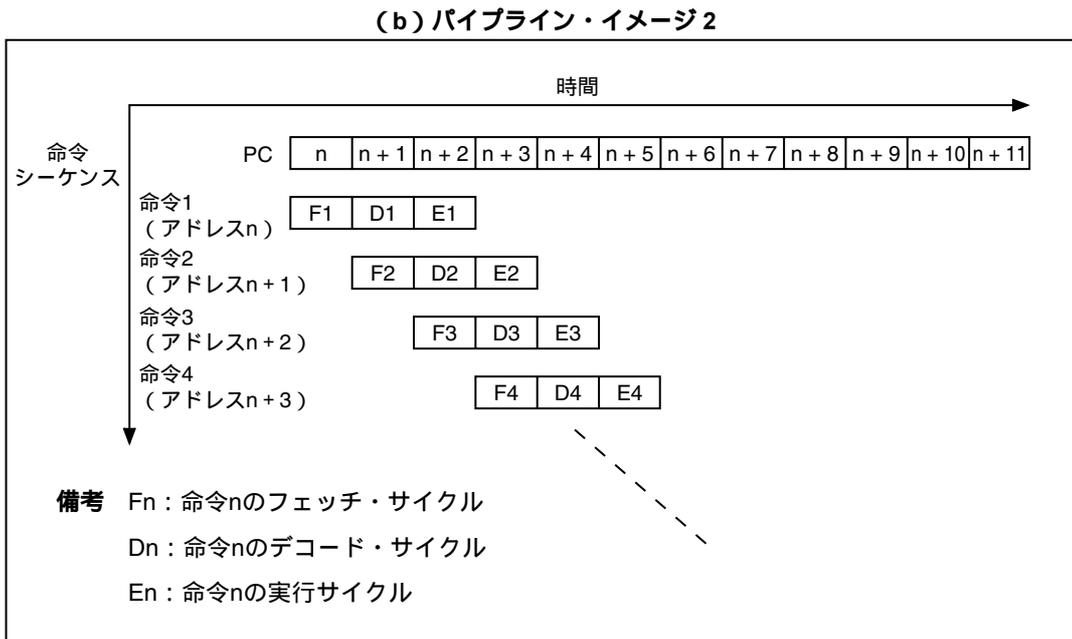


図3-6 パイプライン・イメージ (2/2)



(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、3.4.2 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 (3.4.4 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 (3.4.3 フロー制御部参照)。

3.3.4 スタンバイ機能

μ PD77111 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 命令を実行することによってホールド・モードに移行し、このときデバイスの消費電流は低下します。

ホールド・モードへ移行、ストップ・モードから復帰の過程は次のとおりです。

- (a) HALT 命令でホールド・モードに移行する。
- (b) このときレジスタ、および内部メモリはホールド直前の状態を保持し、消費電流は低下する。また、デバイスの端子の状態については表 3 - 9 を参照。
- (c) ホールド・モードからの復帰は、外部 / 内部の割り込み（ただし、マスクされていない要因による）、またはハードウェア・リセットによる（3.4.4 割り込み参照）。
- (d) 割り込みで復帰した場合、割り込み後のリターン・アドレスは HALT 命令直後の命令アドレスとなる。このとき復帰に先立ち、パワーダウン状態から回復するため、1 インストラクション・サイクルのヒートアップ・サイクル（NOP）が挿入される。

表 3 - 9 ホールド中の端子状態

端子名	HOLDRQ がアクティブ（ロウ・レベル）な場合	HOLDRQ がインアクティブ（ハイ・レベル）な場合
CLKOUT ^{注1}	(CLKIN × m/n) クロック出力 ^{注2}	
\overline{X}/Y	ハイ・インピーダンス	ロウ・レベル
DA14-DA0	ハイ・インピーダンス	直前の状態を保持
D15-D0	ハイ・インピーダンス	
$\overline{MRD}/\overline{MWR}$	ハイ・インピーダンス	ハイ・レベル
\overline{HOLDAK}	ロウ・レベル	ハイ・レベル
\overline{BSTB}	ハイ・レベル	
SORQ1, SIAK1, SO1, SO2	直前の状態を保持	
\overline{HRE} , \overline{HWE} , HD7-HD0	直前の状態を保持	
P3-P0	直前の状態を保持	
TDO, TICE	直前の状態を保持	

注意 ホールド中にハイ・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。

注 1. μ PD77111, 77112, 77113A, 77114 では、マスク・オプションで CLKOUT をロウ・レベル固定にしている場合は、ロウ・レベルのままです。

2. ホールド・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。

図3-7(a)に割り込みによってホールド・モードから復帰するイメージを示します。また、図3-7(b)にホールド・モードへの移行タイミングを、図3-7(c)にホールド・モードからの復帰タイミングを示します。

図3-7 ホールド・モード(1/2)

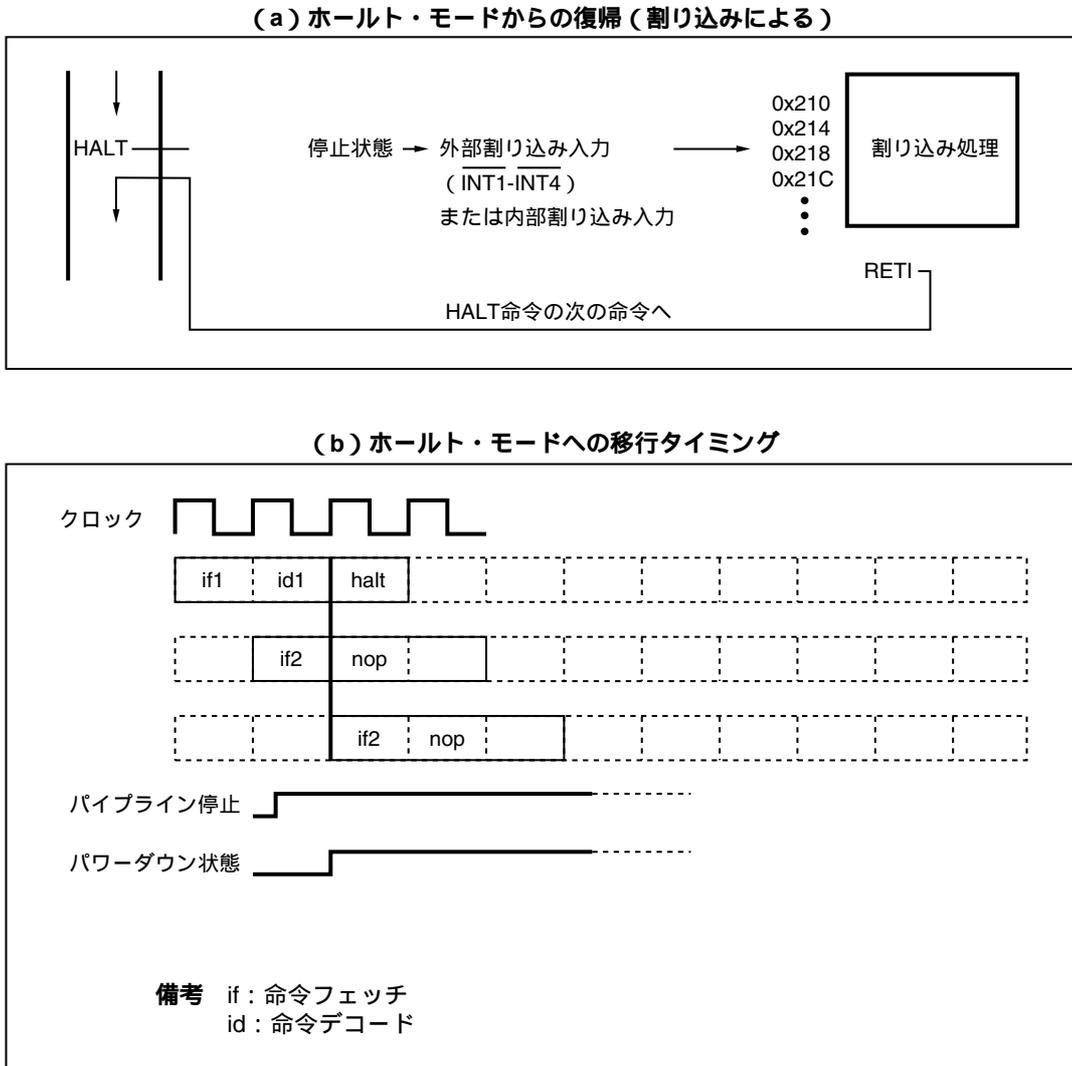
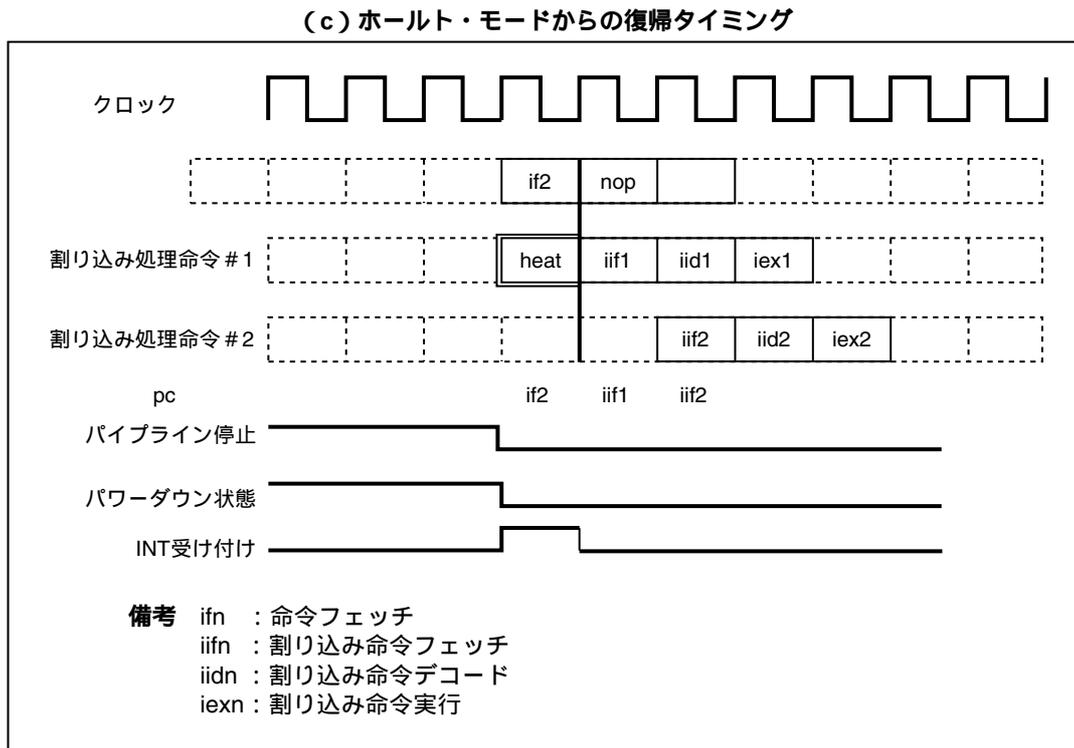


図3-7 ホールト・モード (2/2)



(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十 μ A 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) このときデバイスの端子の状態を表3-10に示す。
- (c) クロック回路とPLLは停止し、デバイスの消費電流は数十 μ A程度に低下する。
- (d) 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 $\overline{\text{WAKEUP}}$ 端子による。ハードウェア・リセットによる復帰にはPLLの引き込み時間がかかるため、100 μ s以上の時間、リセット・アクティブにしてください。

ストップ・モードからの復帰方法には次の2つがあります。

$\overline{\text{WAKEUP}}$ 端子による復帰

$\overline{\text{WAKEUP}}$ 端子によるストップ・モード解除では、PLLのロック時間(約100 μ s)のロウ・レベル入力を必要とし、立ち上がりによって実行を再開します。実行の再開はストップ・モードに入った命令の直後の命令からとなり、ストップ・モードに入る前のレジスタ状態、メモリ状態を保持しています。

$\overline{\text{WAKEUP}}$ 端子は $\overline{\text{INT4}}$ 端子と兼用になっており、ストップ・モード時にのみ $\overline{\text{WAKEUP}}$ 端子として機能します。ストップ・モード時以外は割り込み端子として機能します。また、 $\overline{\text{WAKEUP}}$ 端子によるストップ・モード解除機能はマスク・オプションによって無効にすることができます。マスク・オプションの存在しない μ PD77110の場合は、常に有効です。

注意1. $\overline{\text{WAKEUP}}$ 端子による復帰を行うためには、STOP 命令の直前に NOP 命令を必ず挿入してください。

```
例 :      inst#1 ANY      ;
          inst#2 NOP      ; 必ず挿入する必要があります。
          inst#3 STOP     ;
```

- ストップ・モード時もペリフェラル(シリアル・インタフェース, ホスト・インタフェース)に外部からアクセス信号が入力された場合は動作してしまいますので、注意してください。
 - SIAK , SORQ , $\overline{\text{HWE}}$, $\overline{\text{HRE}}$ がアクティブの場合、シリアル・イネーブル信号やホスト書き込み / 読み込みを行うと内部状態との不整合が生じます。
 - 割り込みを使用している場合、ストップ・モード中にシリアルあるいはホスト割り込みが発生しても、ストップ・モードからの復帰後にはこれらの割り込みは無効になります。ストップ・モード解除後、ダミー・アクセスを行い、割り込みによるペリフェラル・アクセスのシーケンスを再開する必要があります。

RESET 端子による復帰

ハードウェア・リセットでストップ・モードを解除すると、デバイス内部状態および出力端子は初期化されます。ただし、出力端子によっては、デバイスの PLL が安定するまでの間は端子の状態が不定となり、状態を保証できません。ストップ・モード解除リセット期間中の出力端子の状態を表 3 - 11 に示します。

表 3 - 10 ストップ中の端子状態

端子名	$\overline{\text{HOLDRQ}}$ がアクティブ(ロウ・レベル)な場合	$\overline{\text{HOLDRQ}}$ がインアクティブ(ハイ・レベル)な場合
CLKOUT ^注	ロウ・レベル	
$\overline{\text{X}}/\text{Y}$	ハイ・インピーダンス	ロウ・レベル
DA14-DA0	ハイ・インピーダンス	直前の状態を保持
D15-D0	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
HOLDAK	ロウ・レベル	ハイ・レベル
$\overline{\text{BSTB}}$	ハイ・レベル	
SORQ1, SIAK1, SO1, SO2	直前の状態を保持	
$\overline{\text{HRE}}$, $\overline{\text{HWE}}$, HD7-HD0	直前の状態を保持	
P3-P0	直前の状態を保持	
TDO, TICE	直前の状態を保持	

注 マスク・オプションで CLKOUT をロウ・レベルに固定している場合、初期化状態およびストップ・モード解除リセット期間中の状態とも、ロウ・レベルになります。

注意 ストップ中にハイ・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。

表3-11 ストップ・モード解除リセット期間中の出力端子の状態

端子名	初期化状態	ストップ・モード解除リセット期間中の状態
CLKOUT ^注	システム・クロック	不定
\overline{X}/Y	ロウ・レベル	不定
DA14-DA0	0x0000	
D15-D0	ハイ・インピーダンス	
\overline{MRD} , \overline{MWR}	ハイ・レベル	
\overline{BSTB} , \overline{HOLDAK}		
HD15- HD0	ハイ・インピーダンス	
\overline{HRE} , \overline{HWE}	ハイ・レベル	
SO1, SO2	ハイ・インピーダンス	
SIK1, SIRQ1	ロウ・レベル	
P3-P0	入力状態	

注 マスク・オプションで CLKOUT をロウ・レベルに固定している場合、初期化状態およびストップ・モード解除リセット期間中の状態とも、ロウ・レベルになります。

3.4 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

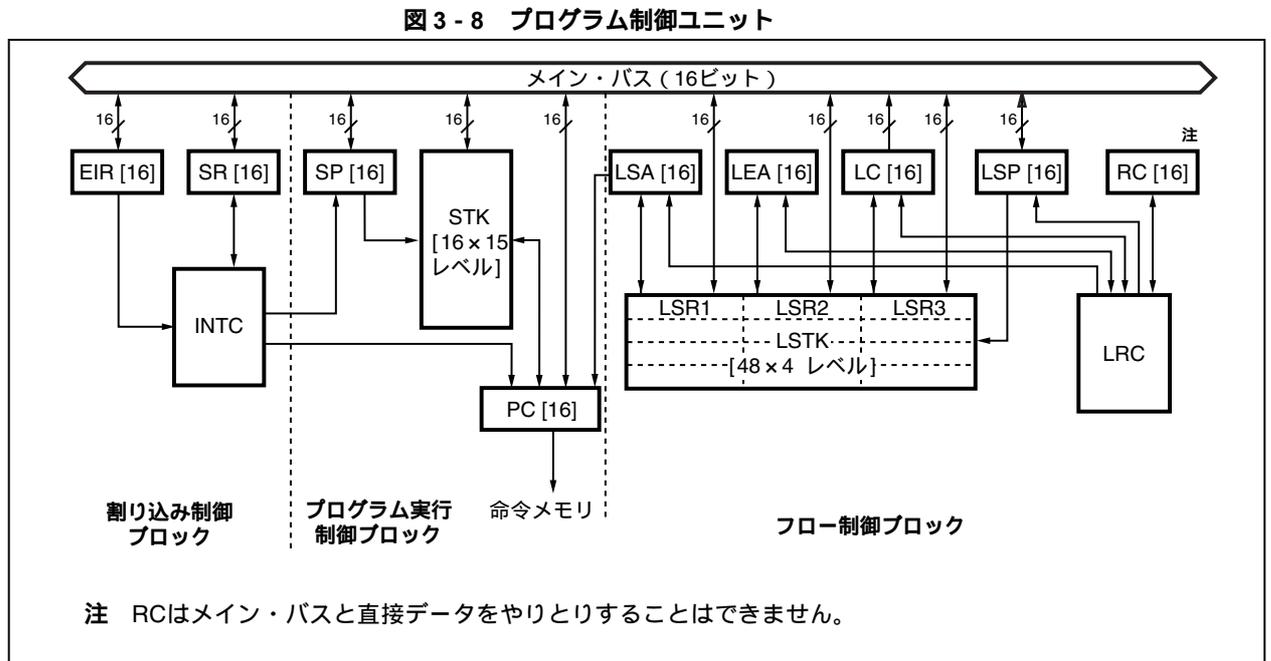
フロー制御部

割り込み制御部

3.4.1 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 3.4.2 プログラム実行制御部から 3.4.4 割り込みでそれぞれの機能の詳細を説明します。

3.4.1 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3-8に示します。



3.4.2 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) 命令メモリ

次に μ PD77111 ファミリの命令メモリ・マップを示します。

図 3-9 命令メモリ・マップ

	μ PD77110	μ PD77111,77112	μ PD77113A,77114	μ PD77115
0xFFFF	システム	システム	内部命令ROM (48 Kワード)	システム
0xC000 0xBFFF				
0x4000 0x3FFF	内部命令RAM (32 Kワード)	内部命令ROM (31.75 Kワード)	0xA000 0x9FFF	システム
0x1000 0x0FFF	システム	システム	0x8000 0x7FFF	システム
0x0240 0x023F	内部命令RAM (3.5 Kワード)	内部命令RAM (1 Kワード)	内部命令RAM (3.5 Kワード)	内部命令RAM (3.5 Kワード)
0x0200	ベクタ領域 (64ワード)	ベクタ領域 (64ワード)	ベクタ領域 (64ワード)	ベクタ領域 (64ワード)
0x01FF 0x0100	システム	システム	システム	ブートアップROM (512ワード)
0x00FF 0x0000	ブートアップROM (256ワード)	ブートアップROM (256ワード)	ブートアップROM (256ワード)	

注意 システム領域に割り当てられたアドレスには、プログラムやデータを置くことも、アクセスすることもできません。これらのアドレスをアクセスしたとき、 μ PD77111ファミリの正常な動作は保証されません。

★

(b) 内部命令メモリ

μ PD77111 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 3 - 12 に示します。

表 3 - 12 内部命令メモリの容量

品名	内部 ROM 容量	内部 RAM 容量
μ PD77110	なし	35.5K ワード
μ PD77111	31.75K ワード	1K ワード
μ PD77112		
μ PD77113A	48K ワード	3.5K ワード
μ PD77114		
μ PD77115	なし	11.5K ワード

★

(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 3.4.4 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。

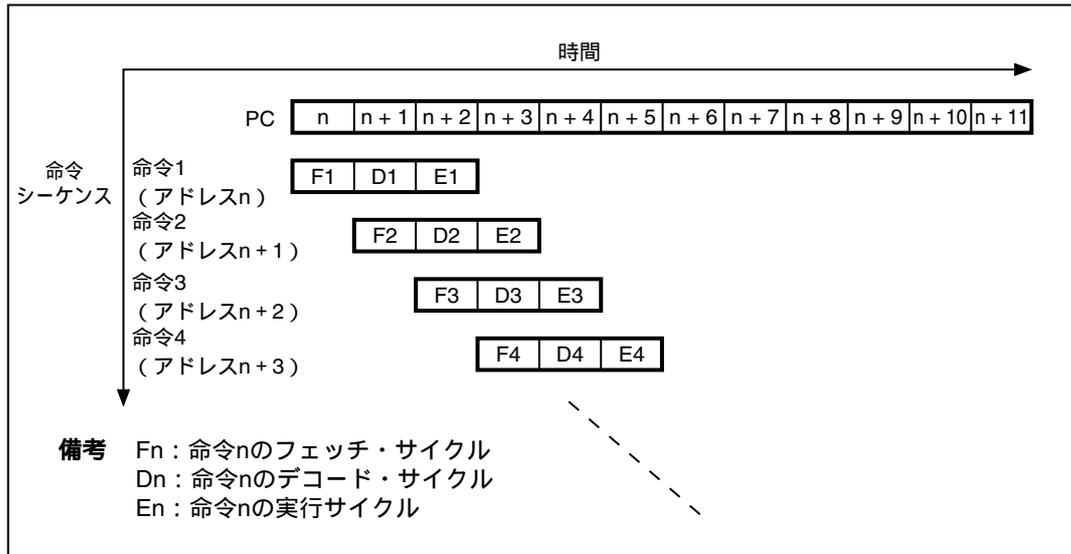
命令実行と PC の動作

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 3 - 10 に示します。

図3-10 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

μ PD77111 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 3 - 13 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 3 - 11 から図 3 - 14 を参照）。

表 3 - 13 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 3 - 11 から図 3 - 14 に示します。

無条件イミューディエト・ジャンプ

無条件間接ジャンプ

条件イミューディエト・ジャンプ（条件成立：分岐）

条件イミューディエト・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3-11 無条件イミディエイト・ジャンプのタイミング

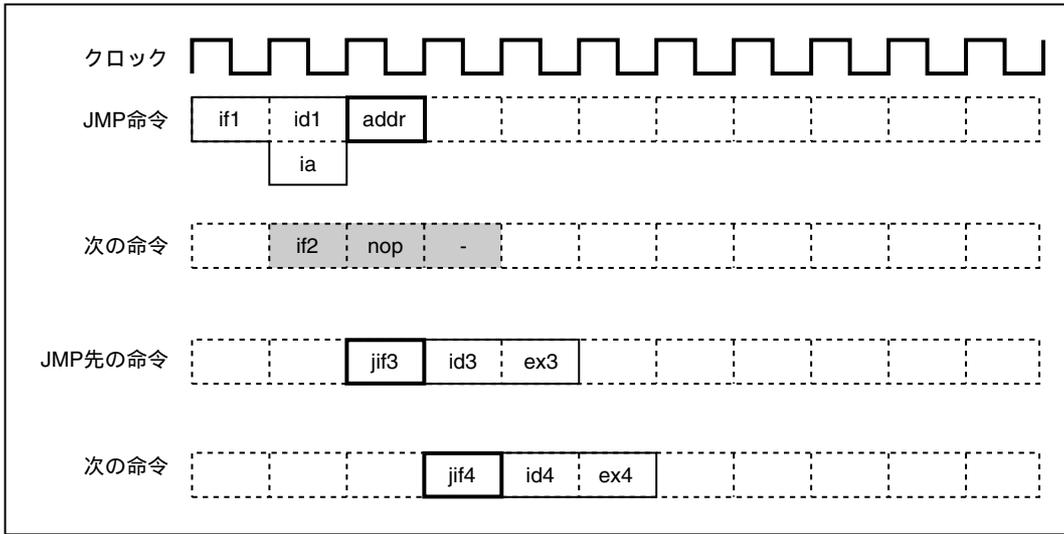


図3-12 無条件間接ジャンプのタイミング

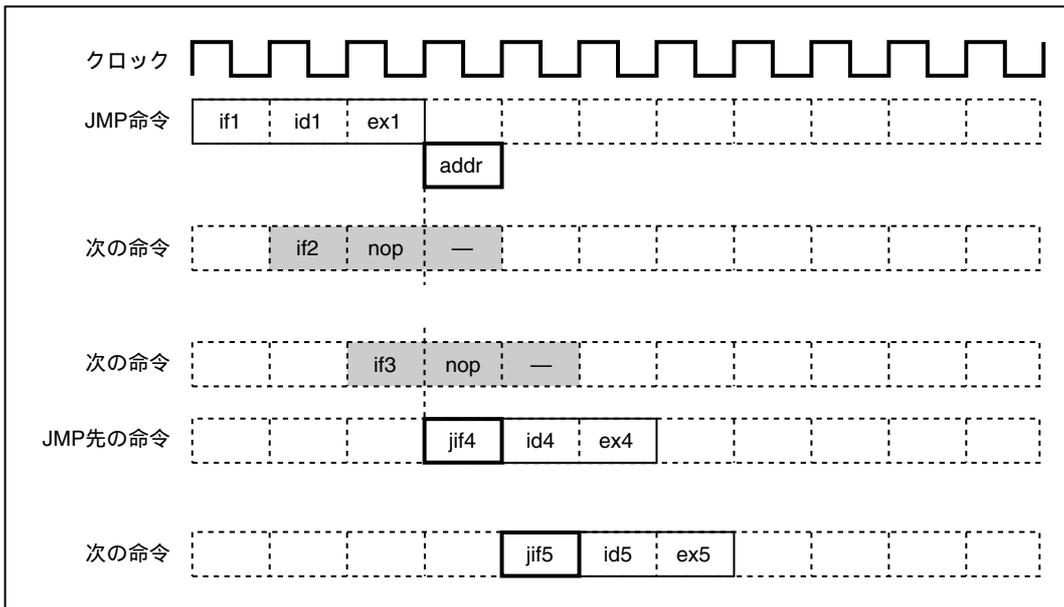


図3-13 条件イミューディエト・ジャンプのタイミング (条件成立: 分岐)

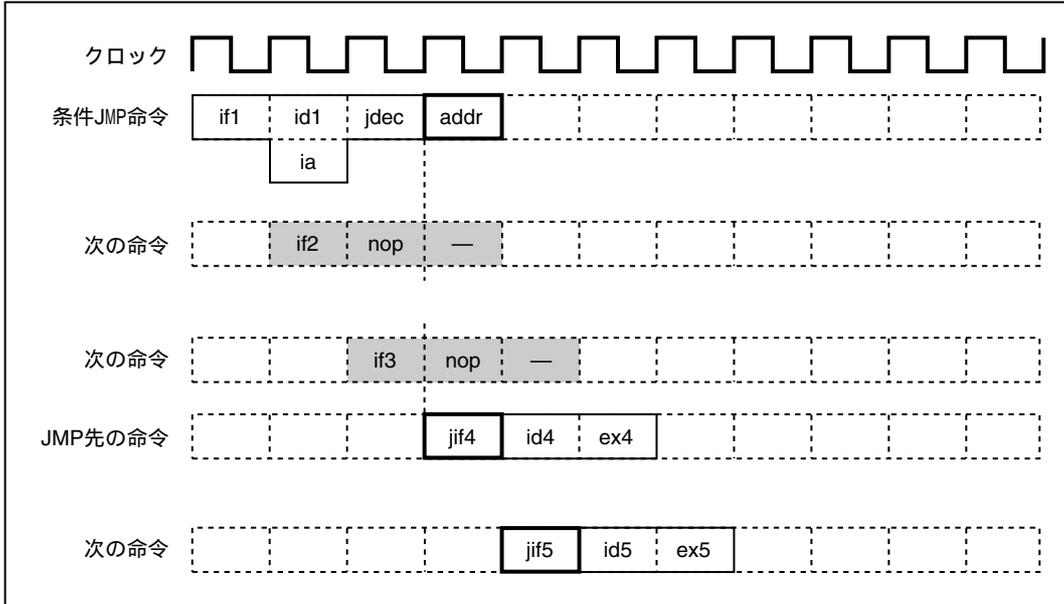
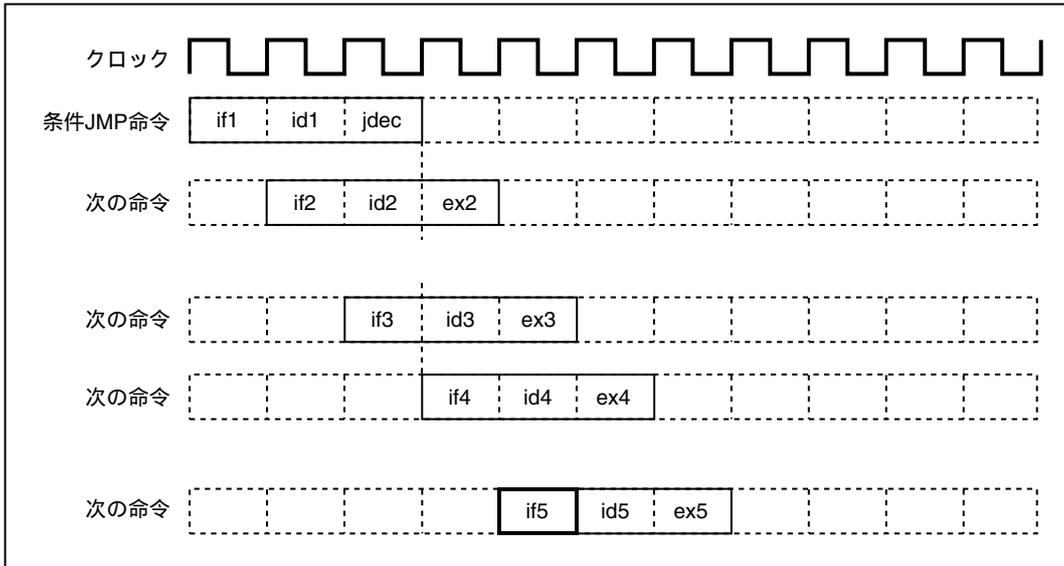


図3-14 条件イミューディエト・ジャンプのタイミング (条件不成立: 通過)



(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、**3.4.4 割り込み**を参照してください。

3.4.3 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)。μPD77111 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 以外のとき、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 3 - 15 にリピート・カウンタ (RC) のフォーマットを示します。

図 3 - 15 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 3.4.4 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*を参照してください。

(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。

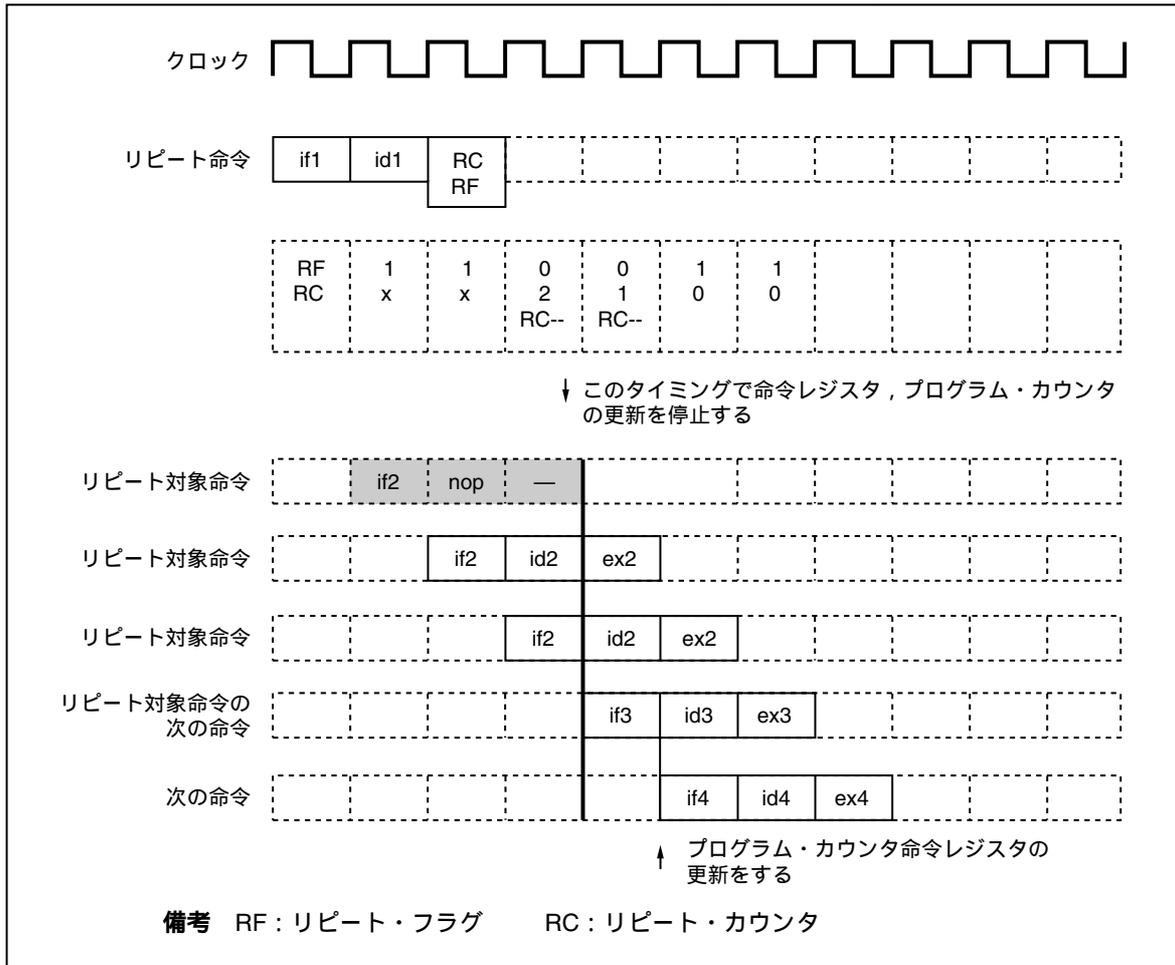
図 3 - 16 にアセンブリ・プログラムを図 3 - 17 にその実行タイミングを、それぞれ示します。

図 3 - 16 リピート命令の例 (2 回リピート)

```

REP    2 ;
R0    /= R1 ;
    
```

図 3 - 17 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。ループのネスタリングは 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 3 - 18 にループ・カウンタ (LC) のフォーマットを示します。

図 3 - 18 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 (3.4.4 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、3.4.4 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)。

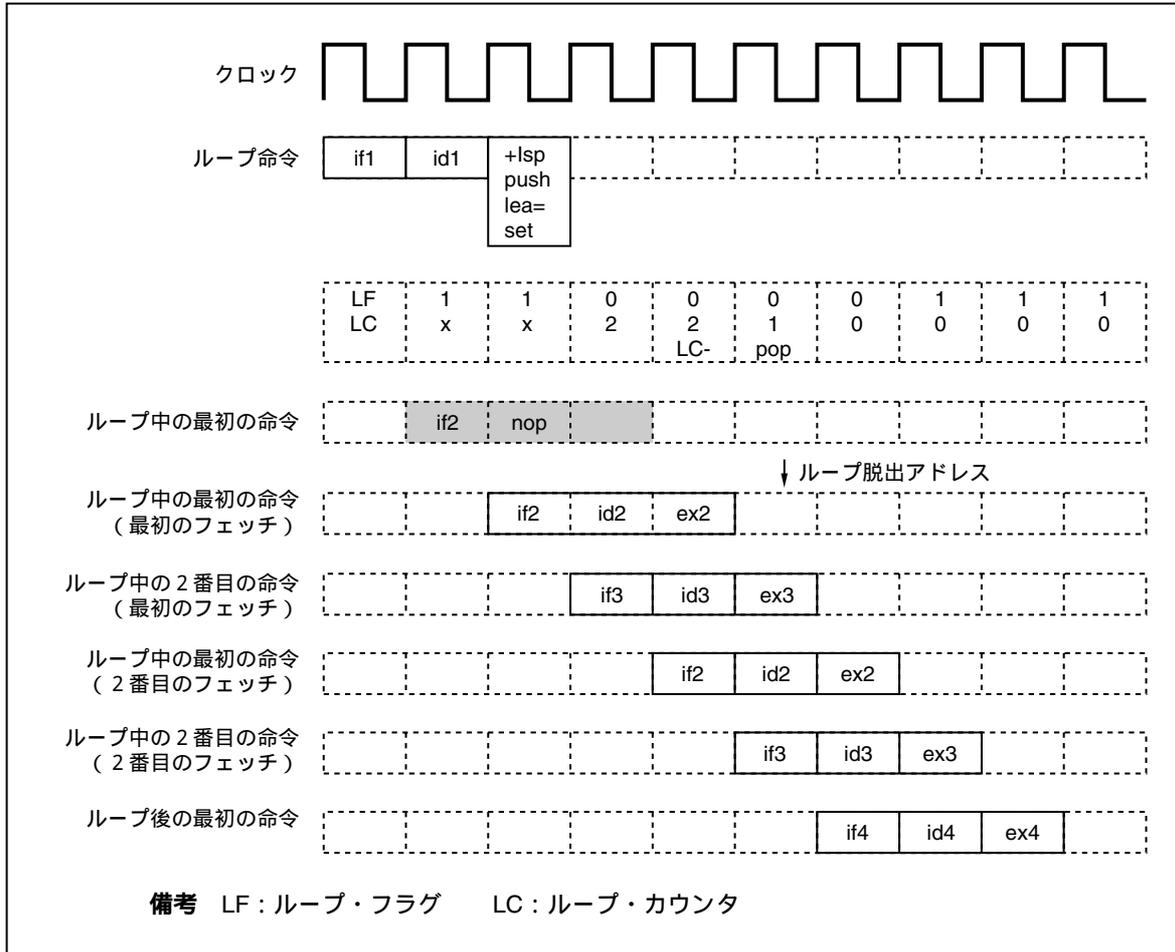
LOOP 命令, LPOP 命令については, **μ PD77016 ファミリ ユーザーズ・マニュアル 命令編**を参照してください。

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 3 - 19 に示します。

図 3 - 19 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) がオーバフローする前に、LSTKの内容をメモリに退避しておく、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
```

3.4.4 割り込み

μPD77111 ファミリには強力な割り込み機能があります。ここでは、次の機能について説明します。

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

外部割り込みのハードウェア条件

割り込みベクタ

(1) 割り込み要因

割り込み要因は、内部および外部割り込みで合計 10 個あります。

内部割り込み：内蔵ペリフェラルに設定された事象を要因とします。6 要因が設定されています。

外部割り込み：ハードウェア信号端子を介して、外部から起動されます。4 入力があります。

表 3 - 14 にすべての割り込み要因を示します。

表 3 - 14 割り込み要因

内部 / 外部	割り込み要因
内部	SI1 入力：SI1 (シリアル#1) の入力完了
	SO1 出力：SO1 (シリアル#1) の出力可能
	SI2 入力：SI2 (シリアル#2) の入力完了
	SO2 出力：SO2 (シリアル#2) の出力可能
	HI 入力：ホスト・インタフェースの入力完了
	HO 出力：ホスト・インタフェースの出力可能
外部	INT1：外部信号端子 $\overline{\text{INT1}}$ の立ち下がりエッジ
	INT2：外部信号端子 $\overline{\text{INT2}}$ の立ち下がりエッジ
	INT3：外部信号端子 $\overline{\text{INT3}}$ の立ち下がりエッジ
	INT4：外部信号端子 $\overline{\text{INT4}}$ の立ち下がりエッジ

(2) 割り込み制御機能

すべての割り込み要因は、内部、外部とも独立の事象、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。

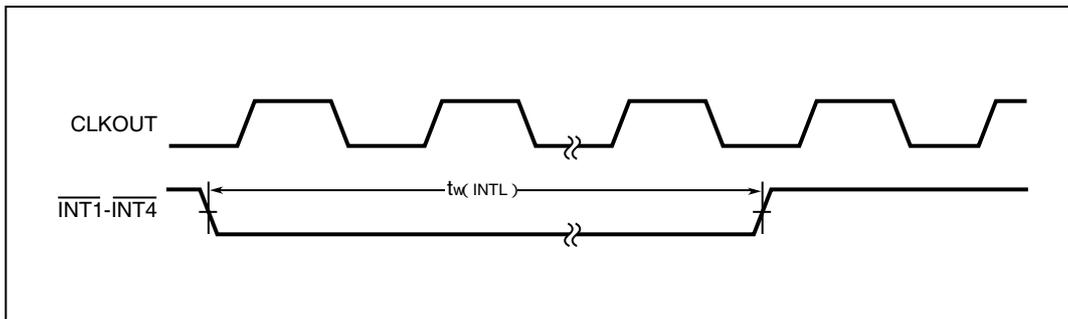
ジャンプ命令をフェッチ/デコード/実行中
リピート命令およびリピート対象命令をフェッチ/デコード/実行中
ループ命令をフェッチ/デコード/実行中
ループ終端命令をフェッチ中

(4) 外部割り込みのハードウェア条件

外部割り込み ($\overline{\text{INT1}} - \overline{\text{INT4}}$) 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3-20に外部割り込みタイミングをそれぞれ示します。

図3-20 外部割り込みタイミング



注意 ホールト・モード時の外部割り込み ($\overline{\text{INT1}} - \overline{\text{INT4}}$) のアクティブ時間は、通常動作時の $t_w(\text{INTL})$ に対して $l \times t_w(\text{INTL})$ (MIN.) に延長されます。これはホールト・モード時に CLKOUT の期間が通常動作時の l 倍に延長されるためです。ここで l はホールト分周率で、マスク・オプションによって設定します。 $\mu\text{PD77110}$ の場合、 l は 8 固定です。

(5) 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트리・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3-15に割り込みベクタ・テーブルを示します。 μ PD77115の割り込みベクタ・テーブルについては、5.7 割り込みベクタ・テーブルを参照してください。

表3-15 割り込みベクタ・テーブル

ベクタ	割り込み要因
	μ PD77110,77111,77112, 77113A, 77114
0x200	リセット
0x204	予約
0x208	
0x20C	
0x210	INT1
0x214	INT2
0x218	INT3
0x21C	INT4
0x220	SI1 入力
0x224	SO1 出力
0x228	SI2 入力
0x22C	SO2 出力
0x230	HI 入力
0x234	HO 出力
0x238	予約
0x23C	

- 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取り扱われます。
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分岐する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には RETI 命令を記述しないでください。記述した場合、実際にその割り込みが発生すると正常に動作しなくなります。
 4. マスク ROM 品においてもベクタ領域は内部 RAM 領域に存在するため、この領域のブートアップが必要です。また、リセット後のエントリが 0x200 番地になるため、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

; Interrupt vector table
int__vec imseg at 0x200 ; ベクタ・テーブル
:
:
org 0x220
; シリアル入力#1 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 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 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 ; start at 0x240 (すべてのベクタについて上記のように
; 記述をすればリンカにより、自動的に配置される)
:
:
; 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

```

(6) 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 3-8 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。μPD77115 のステータス・レジスタについては, 5.7 割り込みベクタ・テーブルを参照してください。

割り込み許可フラグ			ループ	予備		要因別割り込み許可フラグ									
EI	EP	EB	LF			オンチップ I/O デバイス						外部割り込みマスク			
						HO	HI	SO2	SI2	SO1	SI1	INT4	INT3	INT2	INT1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

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。要因別割り込み許可フラグには、次の種類があります。

外部割り込み 1-4 :

外部割り込み端子 ($\overline{\text{INT1}} - \overline{\text{INT4}}$) からの割り込み

SI1, SI2 :

シリアル入力完了し、シリアル・データ・レジスタ (SDT : 入力用) に受信が完了したときに発生する割り込み

SO1, SO2 :

シリアル出力完了し、シリアル・データ・レジスタ (SDT : 出力用) に送信データを書き込むことができるときに発生する割り込み

HI :

ホスト・インタフェース入力完了し、ホスト・データ・レジスタ (HDT : 受信用) に受信が完了したときに発生する割り込み

HO :

ホスト・インタフェース出力完了し、ホスト・データ・レジスタ (HDT : 送信用) に送信データを書き込むことができるときに発生する割り込み

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。

```

ROL = 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
ROL = SR      ; SR を用いて INT1 の割り込みをディスエーブルにする
R0 = R0 | 0x0001 ; INT1 = 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。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EI	EP	EB	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15

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。

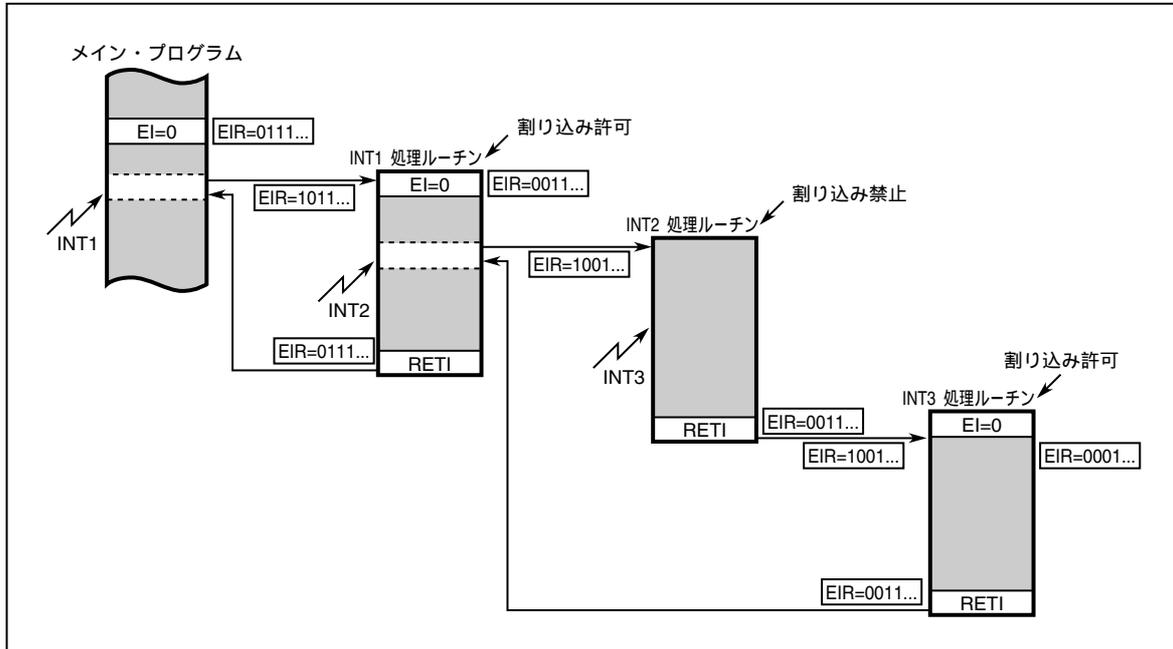
```

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 3 - 21 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 3 - 21 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。

INT1 が受け付けられ、INT1 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。

INT1 の処理ルーチン中で、EI = 0 (イネーブル) とする。

INT2 が受け付けられ、INT2 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。

INT2 の処理ルーチン中で INT3 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。

INT2 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 INT2 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 INT3 記憶が有効化され、INT3 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。

INT3 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 INT3 割り込み前の状態が回復する (INT1 処理中)。

INT1 の処理ルーチンを続行し、最後に RETI 命令で処理すると INT1 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 ; INT1 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。

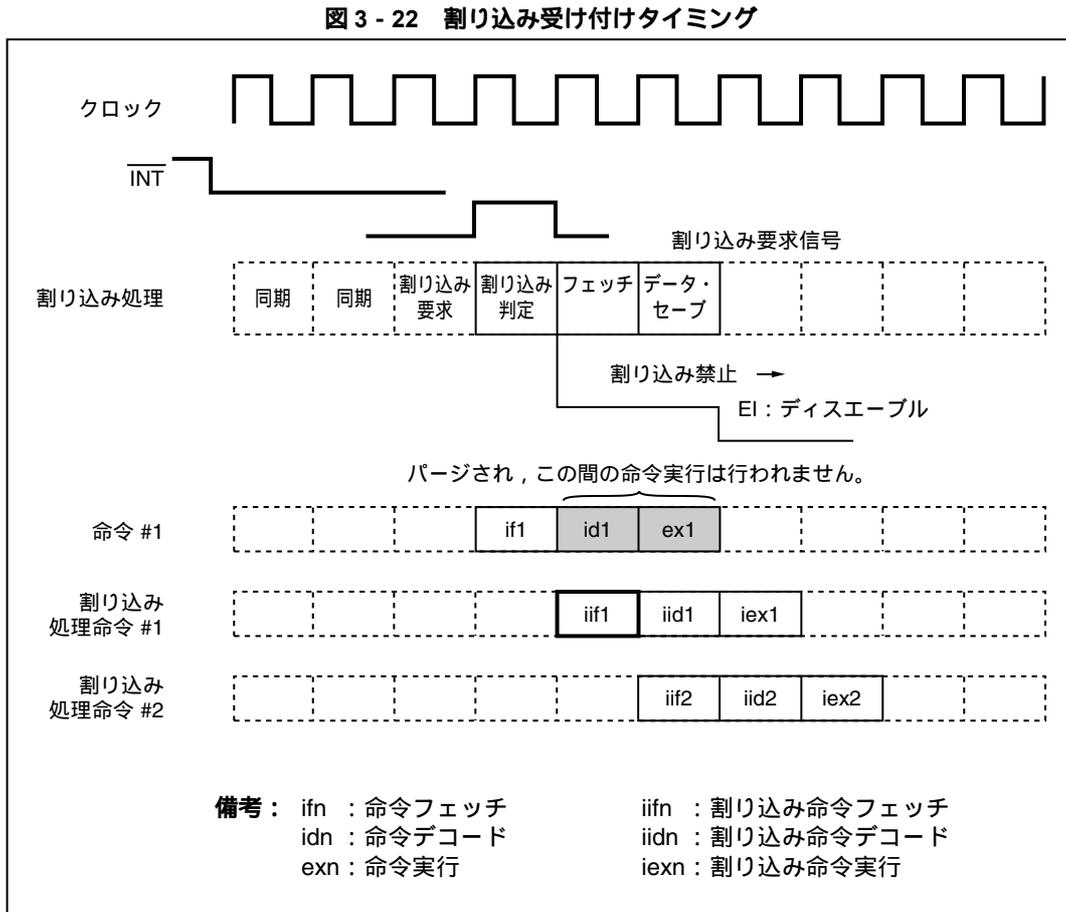
(7) 割り込みシーケンス

(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3-22に割り込み受け付けのタイミングを示します。



(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。

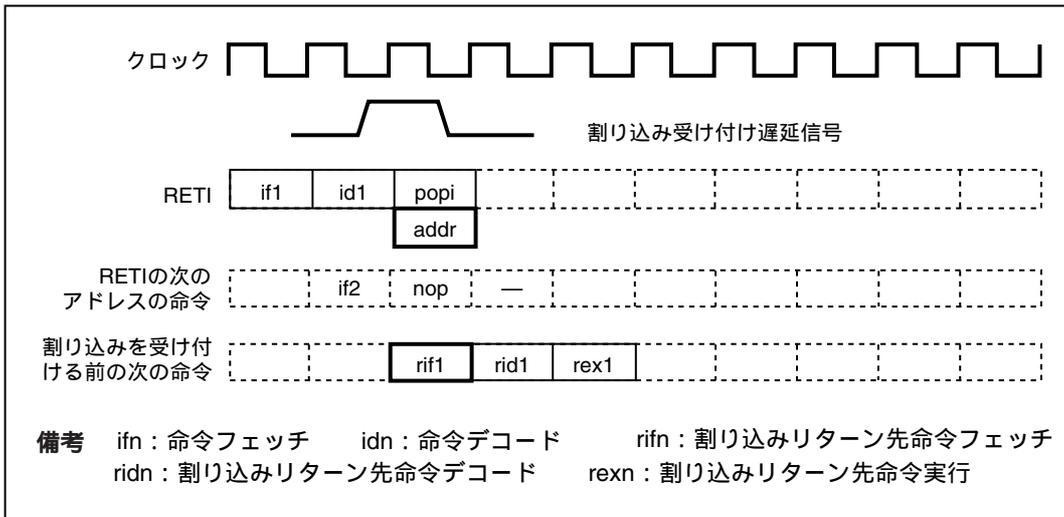
EIRレジスタを左シフトして割り込み許可フラグを回復する。

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。

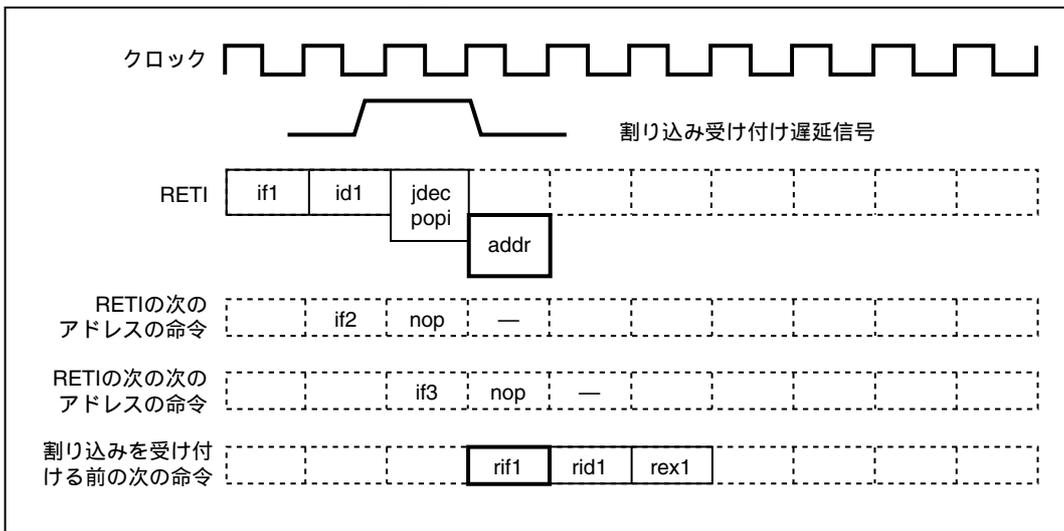
図3-23(a)に無条件RETI命令によってリターンするタイミングを、図3-23(b)に条件付きRETI命令で、条件成立によってリターンするタイミングをそれぞれ示します。

図3-23 RETI 命令によるタイミング

(a) 無条件



(b) 条件付き：条件成立



(8) 割り込みの遅延

割り込み受け付けの過程では、SP, STK, PC の各レジスタが自動的に管理されます。これらレジスタに働きかける命令との競合を避けるために、次に示す競合の発生する命令では割り込み受け付けが遅延されます。割り込みの受け付け自体(割り込み処理ルーチンへ分岐する)は1インストラクション・サイクル遅延されます。

注意 次の場合は、割り込みは受け付けられません。割り込み要求は、割り込み状態がイネーブルになるまで保持されます。

- ペリフェラル I/O がウエイト中
- 外部メモリ・アクセス・ウエイト・サイクル中
- リピート中

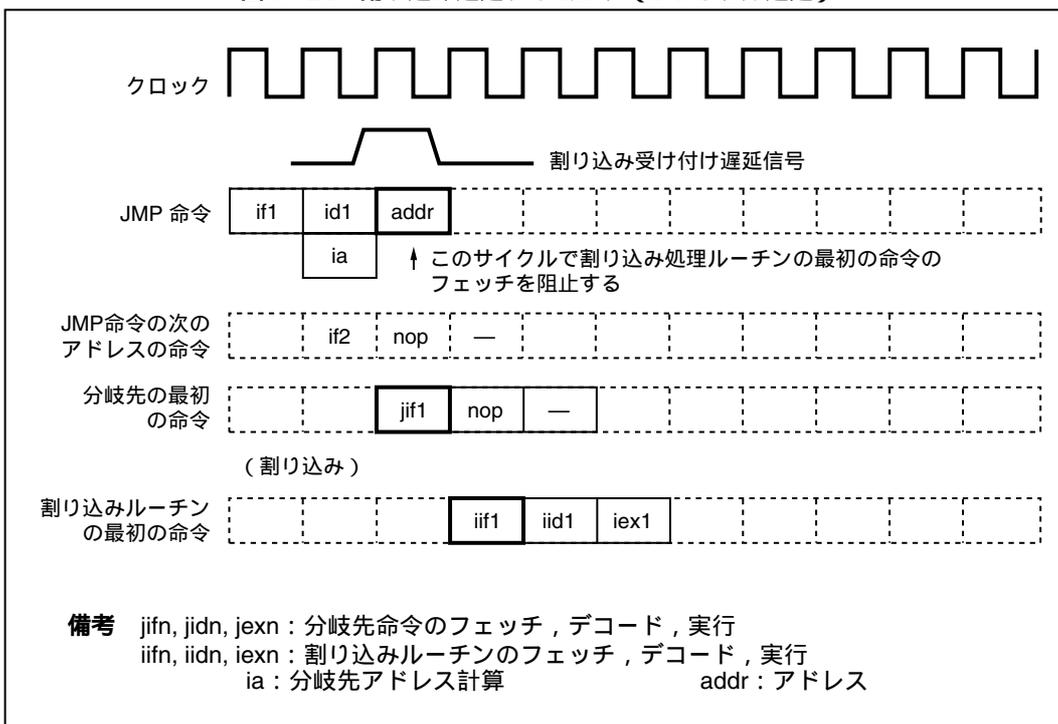
(a) 1 インストラクション・サイクルの遅延を生ずる命令

次に示す命令は、1 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。

- 無条件 JMP 命令 (即値データによる PC 相対ジャンプ) のデコード
- 無条件 CALL 命令 (即値データによる PC 相対ジャンプ) のデコード
- 無条件 RET 命令のデコード
- 無条件 RETI 命令のデコード
- FINT 命令のデコード
- ループ終端命令のフェッチ

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 3 - 24 に示します。

図 3 - 24 割り込み遅延タイミング (1 サイクル遅延)



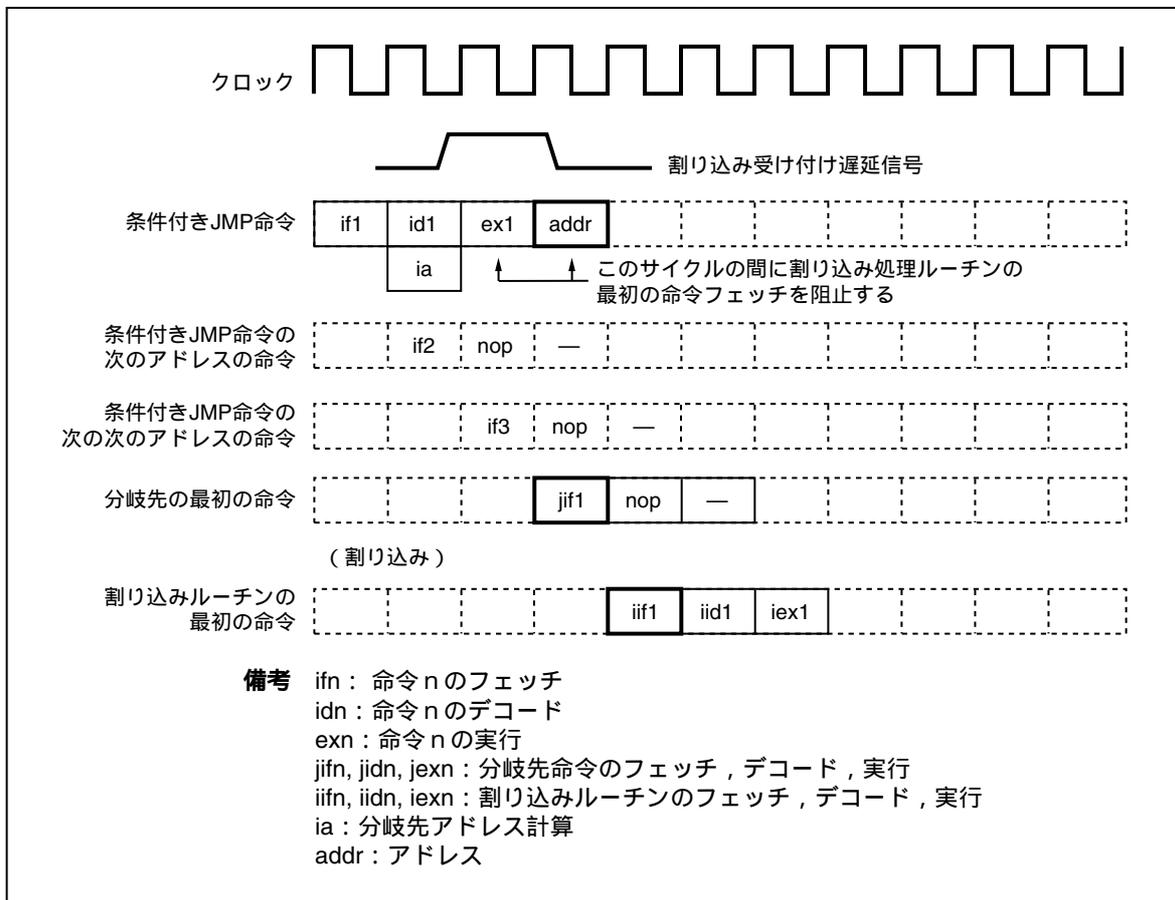
(b) 2 インストラクション・サイクルの遅延を生ずる命令

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 3 - 25 に示します。

図 3 - 25 割り込み遅延タイミング (2 サイクル遅延)



(9) 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 (フォーゲット・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 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。

(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。

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

DSP 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 PD7701x ファミリでは ESR のビット 0 は bac (パス・アクセス・エラー・フラグ) となっていますが、 μ PD77111 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。

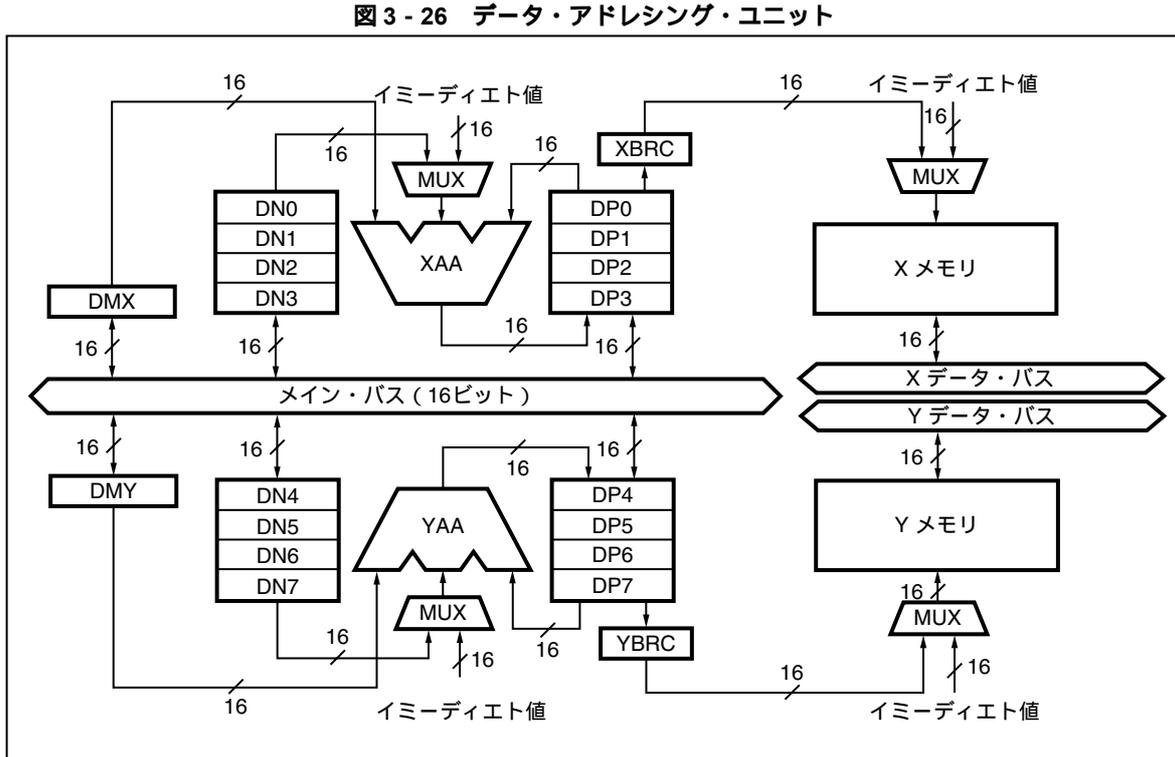
3.5 データ・アドレッシング・ユニット

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 PD77111 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。

3.5.1 ブロック構成

図 3 - 26 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。



3.5.2 データ・メモリ空間

μ PD77111 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

μ PD77111 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(3.2.2 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、μ PD77111ファミリの製品間では相違があります。図3-27に、各製品ごとのデータ・メモリ・マップを示します。

★

図3-27 データ・メモリ・マップ

	μPD77110	μPD77111,77112	μPD77113A,77114	μPD77115
0xFFFF	外部データ・メモリ (32Kワード)	外部データ・メモリ (16Kワード)注	データRAM(8Kワード)	システム
			0xE000 0xDFFF	
	0xC000 0xBFFF	システム		
0x8000 0x7FFF	データRAM (16Kワード)	データROM (16Kワード)	データROM (32Kワード)	0x6000 0x5FFF
				データRAM (8Kワード)
0x4000 0x3FFF 0x3840 0x383F 0x3800 0x37FF 0x3000 0x2FFF 0x2000	システム	システム	システム	システム
	ペリフェラル(64ワード)	ペリフェラル(64ワード)	ペリフェラル(64ワード)	ペリフェラル(64ワード)
	システム		システム	システム
	データRAM(4Kワード)	システム	データRAM(4Kワード)	データRAM(4Kワード)
0x1FFF 0x1000	システム		システム	システム
0x0FFF 0x0000	データRAM(4Kワード)	データRAM(3Kワード)	データRAM(4Kワード)	データRAM(4Kワード)
	0x0C00 0x0BFF		0x3000 0x2FFF 0x2000 0x1FFF 0x1000 0x0FFF	0x3000 0x2FFF 0x2000 0x1FFF 0x1000 0x0FFF

注意 システム領域に割り当てられたアドレスには、プログラムやデータを置くことも、アクセスすることもできません。これらのアドレスをアクセスしたとき、μPD77111ファミリの正常な動作は保証されません。

注 μPD77111,77113Aではシステム領域になります。

(2) 内部データ・メモリ

★ 図3-27に示されるように、μ PD77110は0番地から始まる32Kワードの領域、μ PD77111, 77112は0番地から始まる48Kワードの領域、μ PD77113A, 77114は0番地から始まる48Kワードの領域と0xE000番地から始まる8Kワードの領域、μ PD77115は0番地から64K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。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3.7.2 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3-16 ROMとRAMの容量

製品名	ROM		RAM	
	X	Y	X	Y
μ PD77110	なし		24Kワード	24Kワード
μ PD77111	16Kワード	16Kワード	3Kワード	3Kワード
μ PD77112				
μ PD77113A	32Kワード	32Kワード	16Kワード	16Kワード
μ PD77114				
μ PD77115	なし			

(3) 外部データ・メモリ・インタフェース

(a) 外部データ・メモリ容量

図3-27に示したように、μ PD77111ファミリには外部データ・メモリを拡張できる製品があります。また、拡張可能なメモリ容量は製品によって異なり、表3-17にそれらの相違を示します。

表3-17 外部データ・メモリの容量

製品名	X外部データ・メモリ	Y外部データ・メモリ
μ PD77110	32Kワード	32Kワード
μ PD77111	なし	なし
μ PD77112	16Kワード	16Kワード
μ PD77113A	なし	なし
μ PD77114	8Kワード	8Kワード
μ PD77115	なし	なし

(b) 外部データ・メモリのインタフェース

外部データ・メモリを取り扱う場合、次の点が内部データ・メモリと異なることに注意してください。

外部データ・バスはXデータ・バス、Yデータ・バスが共用しています(3.2.2 データ・バス参照)。したがって、プログラムからは論理的に分離しているX、Yメモリ空間も、 \overline{X}/Y 端子を1ビットのアドレスと見れば物理的には単一のメモリ空間とみなせることになります。

DWTR(データ・メモリ・ウエイト・サイクル・レジスタ)によるウエイト機能があります。

次に外部データ・メモリ・インタフェース端子を説明します。ただし、これらの端子は、外部データ・メモリを拡張できない μ PD77111, 77113A, 77115にはありません。

DA0-DA14(アドレス出力端子)

15ビットのアドレス出力端子です。

バス解放中はハイ・インピーダンスとなります。

リセット直後は、すべての端子がロウ・レベルを出力します。

外部データ・メモリをアクセスしないときは変化しません(アドレスを出力し続けます)。

注意 μ PD77112にはDA14端子がありません。 μ PD77114にはDA13, DA14端子がありません。

\overline{X}/Y (\overline{X}/Y 出力端子)

DA0-DA14から出力しているアドレスがXメモリ空間のときはロウ・レベルを出力し、Yメモリ空間のときはハイ・レベルを出力します。

バス解放中はハイ・インピーダンスとなります。

リセット直後はロウ・レベルを出力します。

D0-D15(データ入出力端子)

16ビットのデータ入出力端子です。

バス解放中はハイ・インピーダンスとなります。

外部データ・メモリをアクセスしないときにはハイ・インピーダンスとなります。

\overline{MRD} (メモリ・リード出力端子)

外部データ・メモリのリード・ストロブ出力端子です。

バス解放中はハイ・インピーダンスとなります。

CLKOUTの立ち上がりエッジでデータを読み出します。

\overline{MWR} (メモリ・ライト出力端子)

外部データ・メモリのライト・ストロブ出力端子です。

バス解放中はハイ・インピーダンスとなります。

CLKOUTの立ち上がりエッジでデータを出力します。

$\overline{\text{HOLDRQ}}$ (バス・ホールド・リクエスト入力端子)

バス占有要求信号入力端子です。

μ PD77111 ファミリ製品を含む複数の CPU 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。

$\overline{\text{HOLDRQ}}$ をロウ・レベルにすると、進行中のバス・サイクルが終了後、外部デバイスにバスを解放します。

$\overline{\text{BSTB}}$ (バス・ストロブ出力端子)

外部データ・バスの使用要求信号を出力する端子です。

バス・マスタとしてバスを掌握しているときはバスのストロブ信号で、バスに対してアクセスが行われていることを示しています。

バス・スレーブの状態にあるときは、バス・マスタの $\overline{\text{HOLDRQ}}$ 端子に入力する外部データ・バス要求信号となります。

$\overline{\text{HOLDAK}}$ (バス・ホールド・アクノリッジ出力端子)

外部デバイスに対し、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は、ロウ・レベルを出力します。

表 3 - 18 端子の状態

端子名	入出力	リセット中	リセット後の初期化値	外部メモリをアクセスしない場合	バス解放中
DA0-DA14 ^注	出力	ロウ・レベル	ロウ・レベル	前の状態を保持	ハイ・インピーダンス
$\overline{\text{X}}/\text{Y}$		ハイ・レベル			
D0-D15	入出力	ハイ・インピーダンス	ハイ・インピーダンス	ハイ・インピーダンス	ハイ・インピーダンス
$\overline{\text{MRD}}$	出力	ハイ・レベル	ハイ・レベル	ハイ・レベル	ハイ・インピーダンス
$\overline{\text{MWR}}$					
$\overline{\text{HOLDRQ}}$	入力	-	-	-	ロウ・レベル
$\overline{\text{BSTB}}$	出力	ハイ・レベル	ハイ・レベル	ハイ・レベル	ハイ・レベル/ ロウ・レベル
$\overline{\text{HOLDAK}}$		前の状態を保持	前の状態を保持	ハイ・レベル	ロウ・レベル

注 μ PD77112 は DA0-DA13 , μ PD77114 は DA0-DA12

(c) データ・メモリ・アクセス・タイミング

図3-28にリード・サイクルのタイミングを、図3-29にライト・サイクルのタイミングを示します。
 なお、データ・メモリ・リード・サイクルが連続して発生した場合、サイクル中の $\overline{\text{MRD}}$ はロウ・レベルになったままです。

図3-28 データ・メモリ・リード・サイクルのタイミング

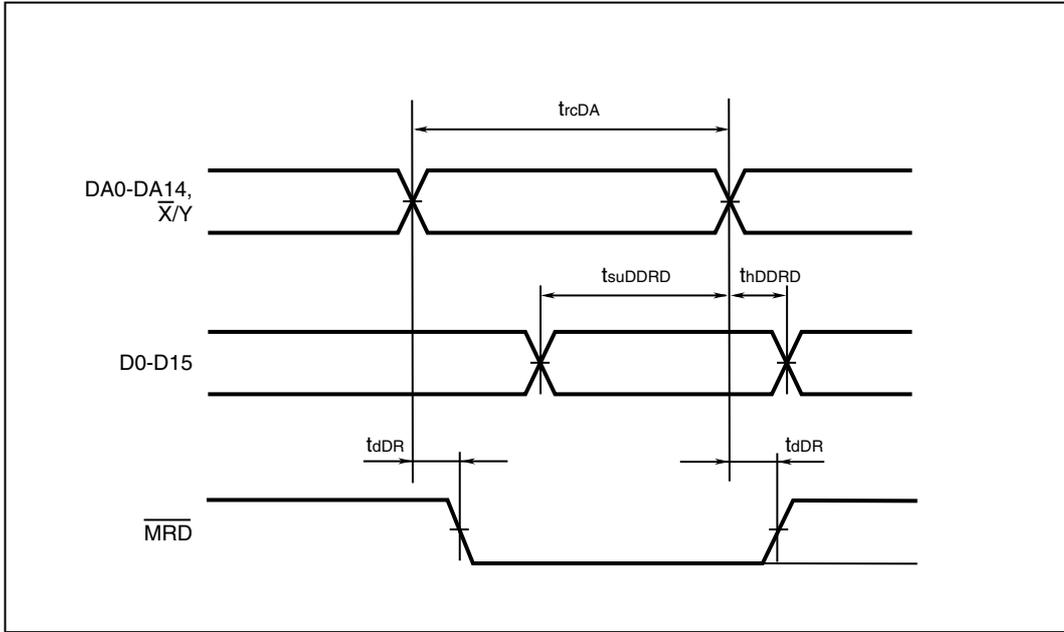
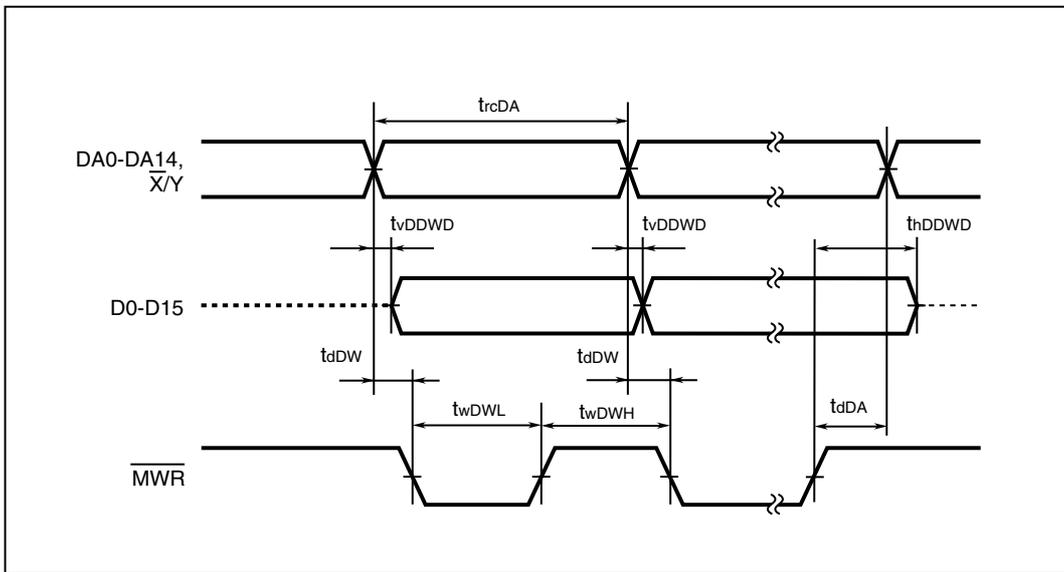


図3-29 データ・メモリ・ライト・サイクルのタイミング



(d) ウェイト・コントローラ

ウェイト回路は、アクセス・タイムの遅い外部データ・メモリのアクセスを可能にするもので、外部データ・メモリ・アクセス・サイクル中にウェイト・サイクルを挿入します。ウェイト機能は DWTR (データ・メモリ・ウェイト・サイクル・レジスタ) で制御します。

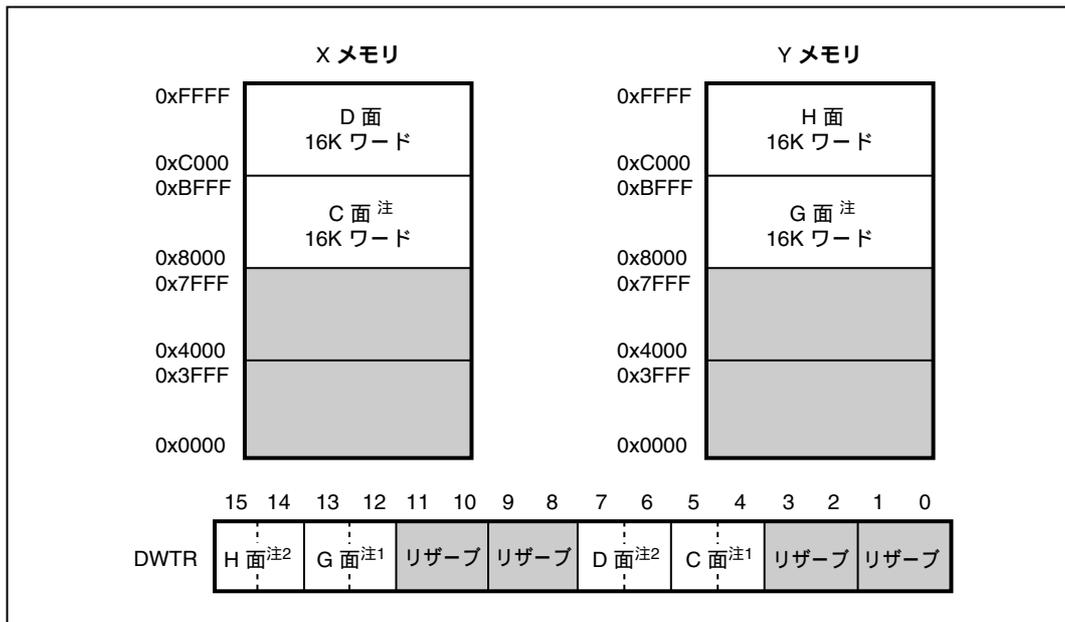
ペリフェラル・レジスタとしてメモリ空間にマッピングされている DWTR に対し、ソフトウェアであらかじめ定められたウェイト・サイクル数を指定します。

DWTR は 16 ビットのレジスタで、2 ビットずつ 2 フィールドに分割されます。X、Y メモリ空間 64 Kワードずつをそれぞれ 4 等分したうちの外部メモリに該当する 2 つの 16 Kワード・バンクに対して、それぞれ独立にウェイト・サイクル数を選択設定できるようになっています。

これらの制御イメージを、図 3 - 30 に、また、DWTR の各フィールドに設定される値とウェイト・サイクル数との関係を表 3 - 19 に示します。

備考 DWTR は、ソフトウェアで制御するプログラマブルなウェイト機能を実現する 16 ビットのレジスタです。ペリフェラル・レジスタの 1 つとして設けられ、アプリケーション・プログラムからこのレジスタに、あらかじめ定められたウェイト・サイクル数を選択して設定することができます。

図 3 - 30 データ・メモリ制御バンクと DWTR のフィールド構成



- 注意 1. μ PD77112, 77114 の場合、ビット 4, 5, 12, 13 もリザーブ・ビットとなります。
- 2. μ PD77111, 77113A の場合、すべてリザーブ・ビットとなります。
- 3. μ PD77113A, 77114 の場合 H, D 面の内部データ RAM 領域 (0xE000-0xFFFF) は設定にかかわらず 0 ウェイトに設定されます (外部メモリ領域は DWTR にて設定されます)。
- 4. リザーブ・ビットに対して 0 以外の値を設定しないでください。正常に動作しなくなる可能性があります。
- 5. ホスト・ブート時、DWTR は初期化されません。 μ PD77111, 77113A を含め、外部メモリ領域を使用しない場合でも、プログラム中で設定を行ってください。

表3-19 DWTR フィールド設定値とウエイト・サイクル数

ビット		ウエイト数	備考
15	14		
13	12		
7	6		
5	4		
0	0	0	外部データ領域を設定しない場合だけ設定可能 [※]
0	1	1	2サイクルでアクセスします
1	0	3	4サイクルでアクセスします
1	1	7	8サイクルでアクセスします

注 外部データ領域へのノー・ウエイト・アクセスはできません。外部データ領域を使用する場合は、必ず1ウエイト以上のウエイト・サイクルを設定してください（外部データ領域を使用しないときは任意の値でかまいません）。

注意 DWTRを設定した場合、設定されたウエイト・サイクル数はDWTRにデータを転送した命令の直後の命令から有効になります。

(e) バス・アービトレーション

μ PD77111 ファミリは、多重バス・マスタのメモリ構成をサポートするために、バス・アービトレーション機能を備えています。典型的な多重バス・マスタは、次の組み合わせでメモリを共有する場合として考えることができます。

μ PD77111 (マスタ) - μ PD77111 (スレーブ)

μ PD77111 - ホスト CPU

μ PD77111 - DMA コントローラ

注意 この項で述べる μ PD77111 とは、外部データ・バスを持つ、 μ PD77110, 77112, 77114 を示します。

ただし、これらの組み合わせでお互いの電源電圧が異なる場合、注意が必要です。典型的なバス・アービトレーションを図3-31に示し、動作の手順を次に説明します。

μ PD77111 が共有メモリにアクセスする命令を実行すると、 $\overline{\text{BSTB}}$ がアクティブ（ロウ・レベル）となる。

外部デバイスが $\overline{\text{HOLDRQ}}$ をアクティブ（ロウ・レベル）にする。

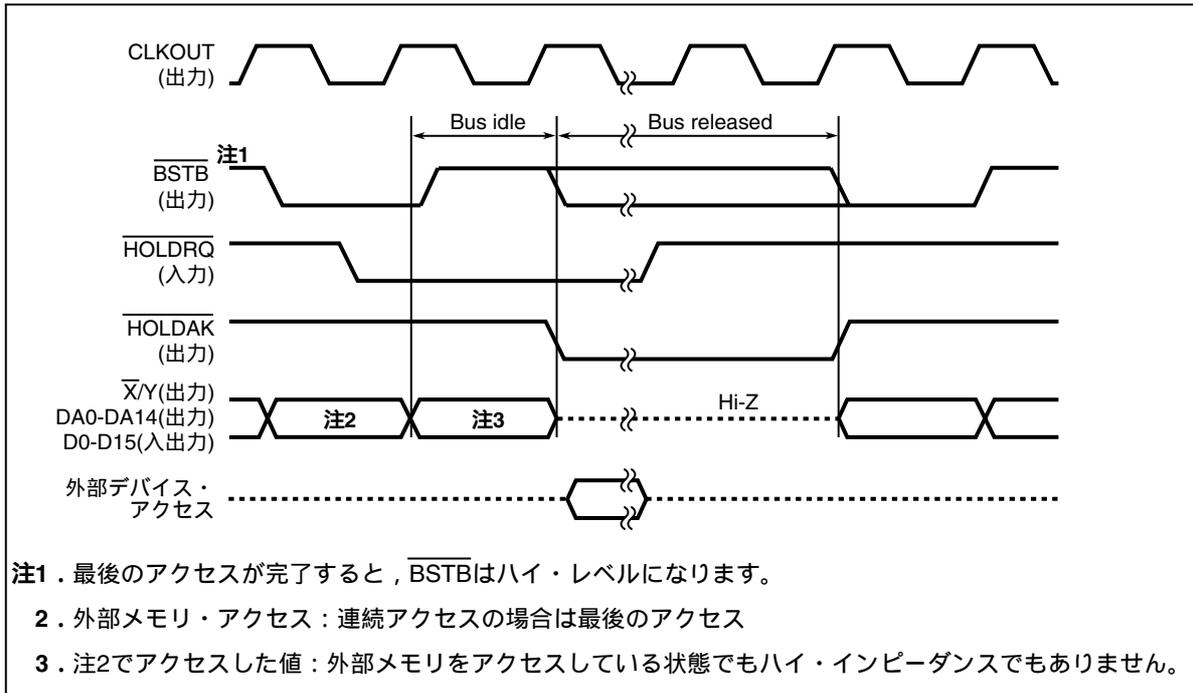
μ PD77111 はバス・サイクル終了後（連続アクセスの場合はアクセスが終了したあと）、 $\overline{\text{HOLDAK}}$ をアクティブ（ロウ・レベル）にする。

外部デバイスがバスを使用する。

外部デバイスは、アクセスを終了すると $\overline{\text{HOLDRQ}}$ をインアクティブ（ハイ・レベル）にする。

μ PD77111 は $\overline{\text{HOLDAK}}$ をインアクティブ（ハイ・レベル）にし、バス・アクセスを再開する。

図3-31 バス・アービトレーションの手順



(4) 同時アクセスの制限

μ PD77111 ファミリはさまざまな用途でメモリ空間を分割しています。一方で、 μ PD77111 ファミリは、パラレル・ロードなど X, Y メモリ空間に対して同時に 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。ここではそれらのメモリ区分のどのような組み合わせで同時アクセスが可能であるのかを説明します。表 3 - 20 に同時アクセスの可否を示します。X メモリ空間には、X データ・バスを通じて、Y メモリ空間には Y データ・バスを通じてアクセスします。

表 3 - 20 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	内部 ROM	内部 RAM	外部メモリ	ペリフェラル
内部 ROM	OK	OK	OK	OK
内部 RAM	OK	OK	OK	OK
外部メモリ	OK	OK	-	OK
ペリフェラル	OK	OK	OK	-

備考 OK : 同時アクセス可能

- : 同時アクセス不可

注意 並列アクセスが不可の場合のアクセスを行ったとき、 μ PD7701x ファミリでは ESR レジスタ中の bac フラグが 1 を示し、エラーを表示しましたが、 μ PD77111 ファミリではこの場合もエラーを表示しません。

3.5.3 アドレッシング・モード

μ PD77111 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 3 - 26 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
す。

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 (イミューディエト加算)

(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 **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。

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 3-32 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 にイミディエイ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（*μPD77016 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：

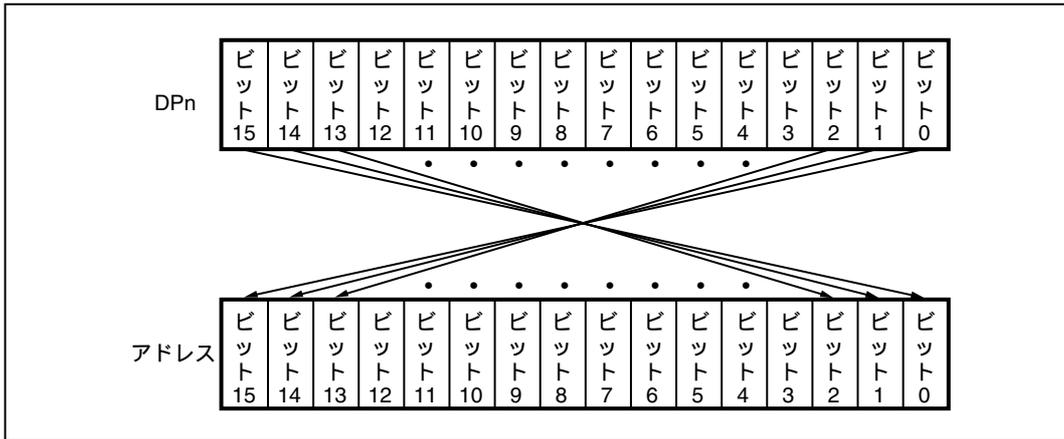
```
R5H=*DP3%%;
```

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 3 - 32 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3-32 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DN6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・イミューディエト加算)

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ューディエト値を加算します。

イミュー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 イミュー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3-21 データ・ポインタの Modifier 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)) + DP _H
	(n=4-7) DPn = ((DPL + DNn) mod (DMY + 1)) + DP _H
!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 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,

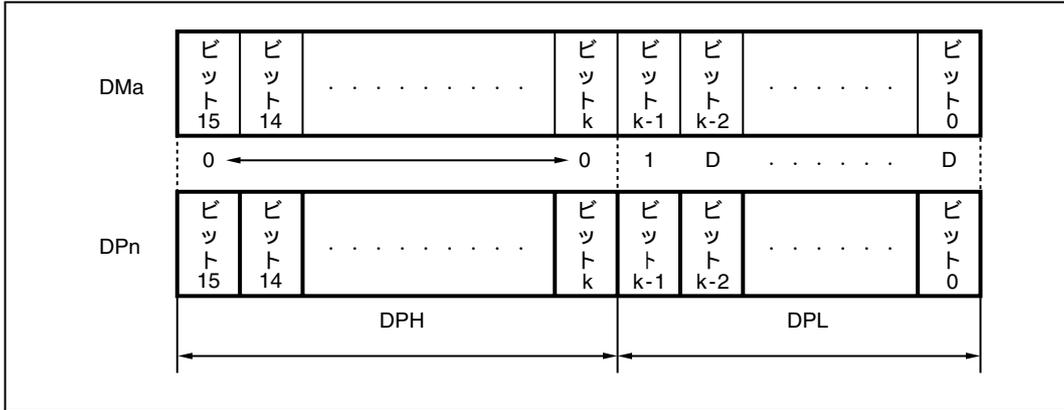
DP_H : DMA の値が [2k, 2(k - 1)] の範囲にある場合, DPn 初期値の下位 k ビットを 0 にした値 (図3-33 参照)

DP_L : 上記の場合で, DPn の下位 k ビットの値 (図3-33 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。

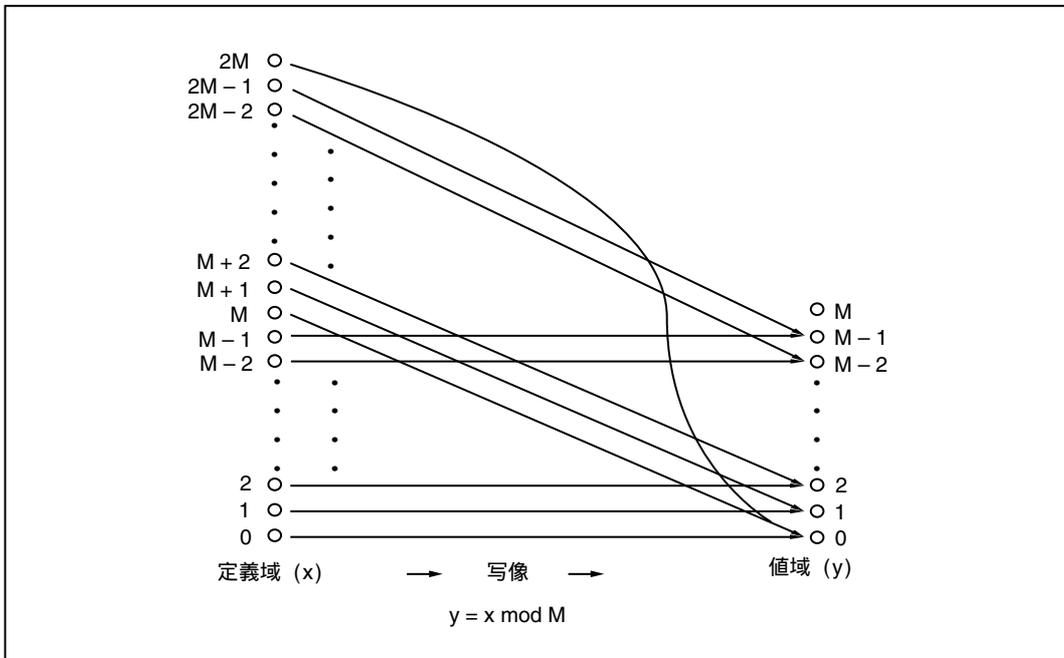
図3-33 DPnの分割



・意味

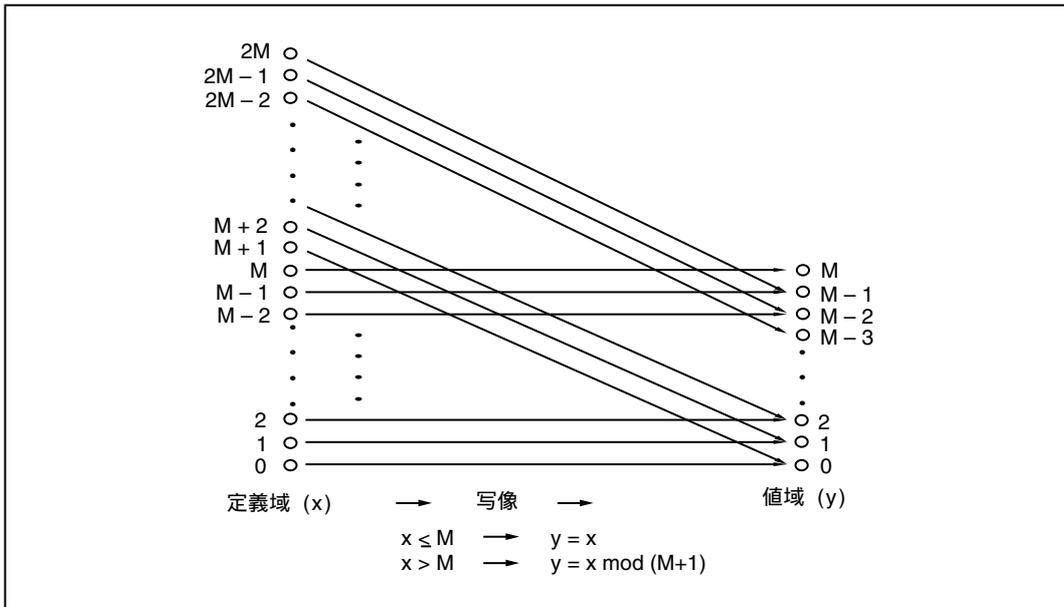
まず、通常のもジュロ演算は図3-34に表される写像と考えることができます。

図3-34 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3-35に表される写像と見るすることができます。

図3-35 モジュロ調整の写像



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 3 - 34 で利用可能なバッファ・サイズは M であるのに対し、図 3 - 35 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 を利用できることとなります。

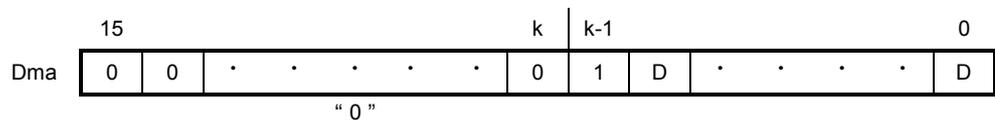
・リング・カウンタの動作範囲

モジュロ・インデクス加算によりリング・カウンタ動作する範囲の先頭アドレスは、データ・ポインタおよびモジュロ・レジスタの値によって決まります。

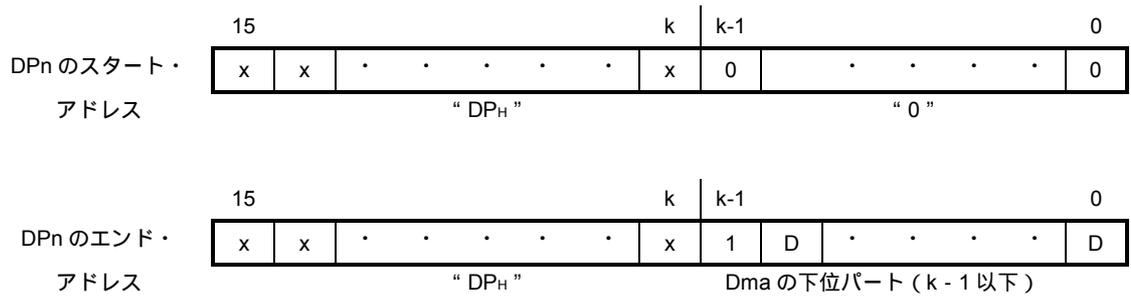
モジュロ・レジスタの値が $2k - 1$ $DMX/Y < 2k$ の場合、データ・ポインタの下位 k ビットが 0 のアドレスが先頭アドレスになります。また、データ・ポインタの下位 k ビットがモジュロ・レジスタと同じアドレスが終端アドレスになります。データ・ポインタの上位 16 - k ビットは変化しません。

リング・カウンタ動作する範囲

Dma を次のように設定します。



リング・カウントのスタート・アドレスとエンド・アドレスは次のとおりです。



・制限

モジュロ・アドレッシングでは、次の制限事項を守ってください。

DMa の範囲は、1-0x7FFF にしてください。

DNn の値の絶対値は DMa を越えないようにしてください。

注意 DMa に 0 を設定することはできませんから、バッファ・サイズ=1 の巡回バッファを構成することはできません。

・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

:

3.6 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 : $R0 = R2 + R3$

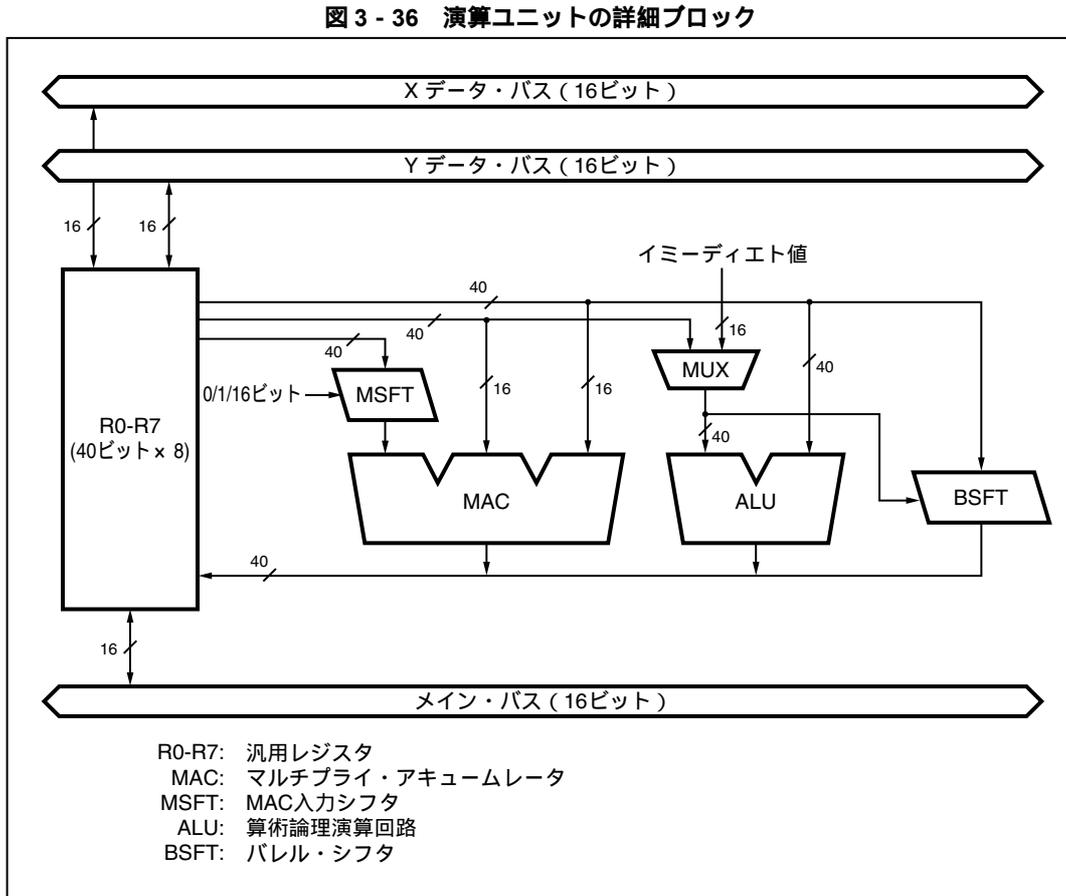
単項演算命令：1 オペランド入力が必要とするすべての操作

例： NEG : $R0 = -R1$

ここでは、**3.6.1 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。

3.6.1 ブロック構成

図3-36に演算ユニットの詳細ブロックを示します。



3.6.2 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部分：ビット 15-0（下位 16 ビット）

H部分：ビット 31-16（中間 16 ビット）

E部分：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部分は数値演算/論理演算の種類によって異なります（表 3 - 22 参照）。
図 3 - 37 に R0HL-R7HL を除いた 5 種類のフォーマットを示します。

表 3 - 22 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 パス転送					-	
レジスタ間転送	-		-	-	-	-

図 3 - 37 汎用レジスタの分割フォーマット

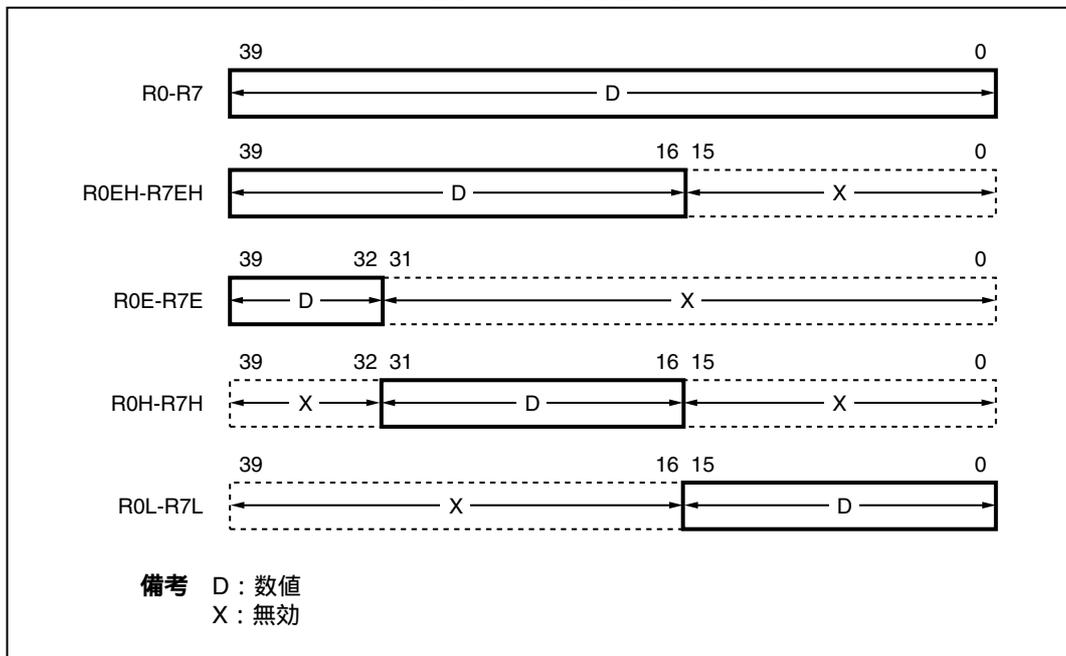
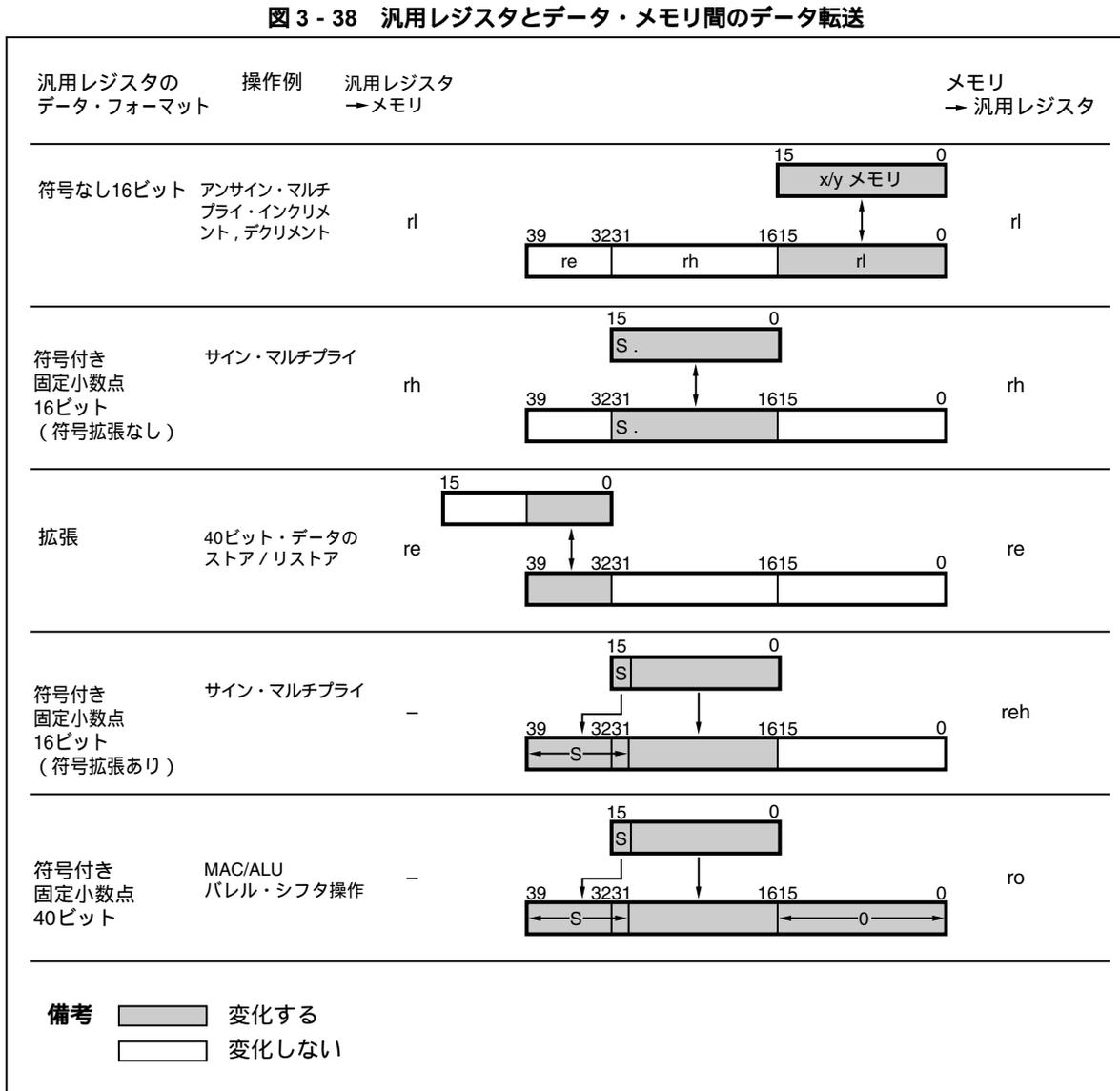


図3-38に汎用レジスタとデータ・メモリ間のデータ転送を示します。



(2) 数値フォーマット

μPD77111 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。

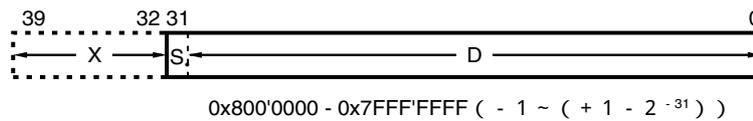
(a) 固定小数点フォーマット

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。

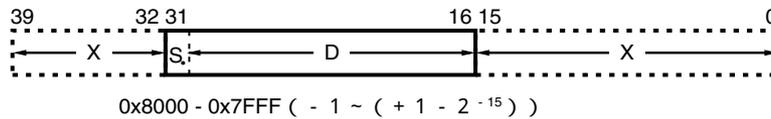
40ビット・データ・フォーマット(加減算などの入力)



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)

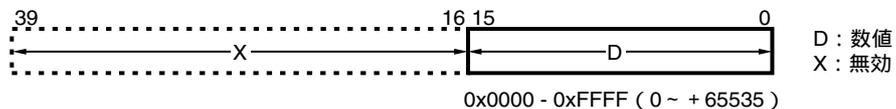


備考 32 ビット固定小数点フォーマットおよび 16 ビット固定小数点フォーマットで、データの絶対値は 1 を越えることはありません。したがって、それらのフォーマットをパラメータとした累算を汎用レジスタ上で実行するかぎりでは、E パートはオーバーフロー吸収領域(ヘッド・ルーム)として機能します。これは 1 累算で 1LSB (E パート換算) のオーバーフローの可能性のあるものと仮定しても、256 回の連続累算に対してオーバーフロー判定を省略することができます。

(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)



3.6.3 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

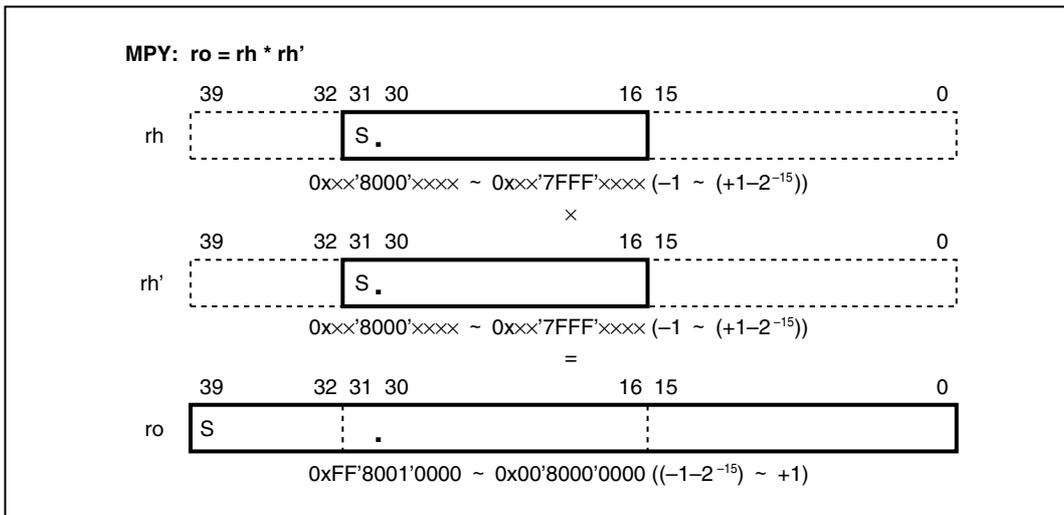
有符号値と無符号値の乗算

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 3 - 39 に示します。

図 3 - 39 有符号値同士の乗算



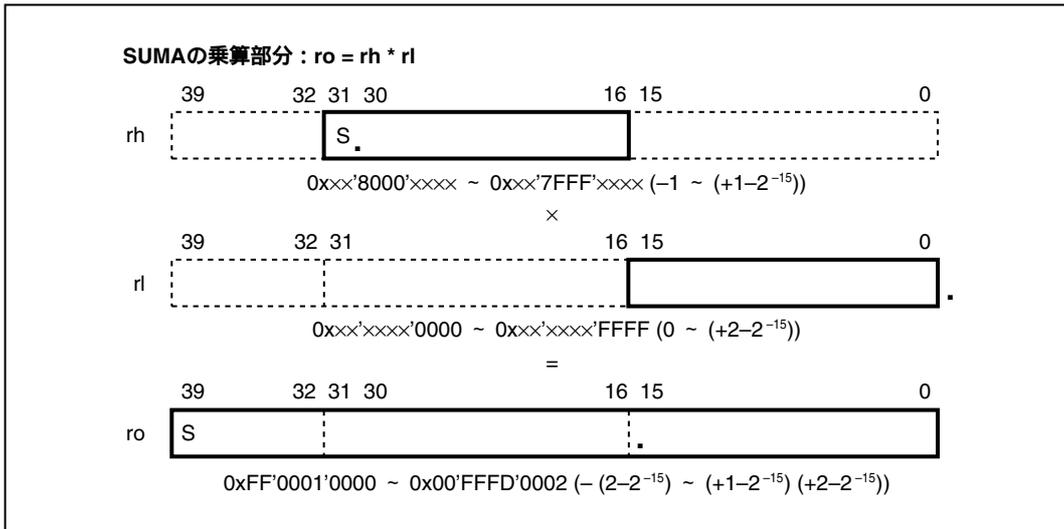
- 備考 1.** $0x8000 (-1)$ 同士の乗算を行うと、結果は $0x00'8000'0000$ となりますが、32 ビット固定小数点フォーマットの範囲では $+1$ の表現ができないため、オーバーフローになります (拡張ビット $re = 0x00$ はこの場合 32 ビット・フォーマットの符号ビットとは異なります)。ただしこの場合でも、40 ビット・フォーマットとして見た場合の値は正確です ($0x00'8000'0000 = +1$)。
- 2.** 16 ビット値同士を乗算すると、有効となるのは最大で 31 ビットです。したがって、乗算結果を示すレジスタの LSB は常に 0 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3-40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3-40 有符号値と無符号値の乗算

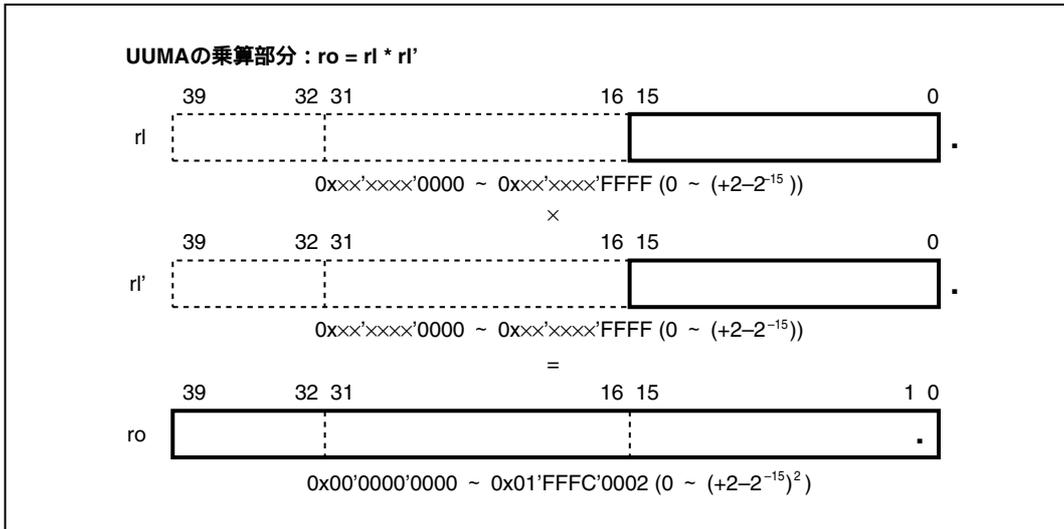


(c) 無符号値同士の乗算

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3-41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3-41 無符号値同士の乗算



(2) 累乗算機能 (三項演算)

μ PD77111 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3-23にこれらの組み合わせを示します。

表3-23 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

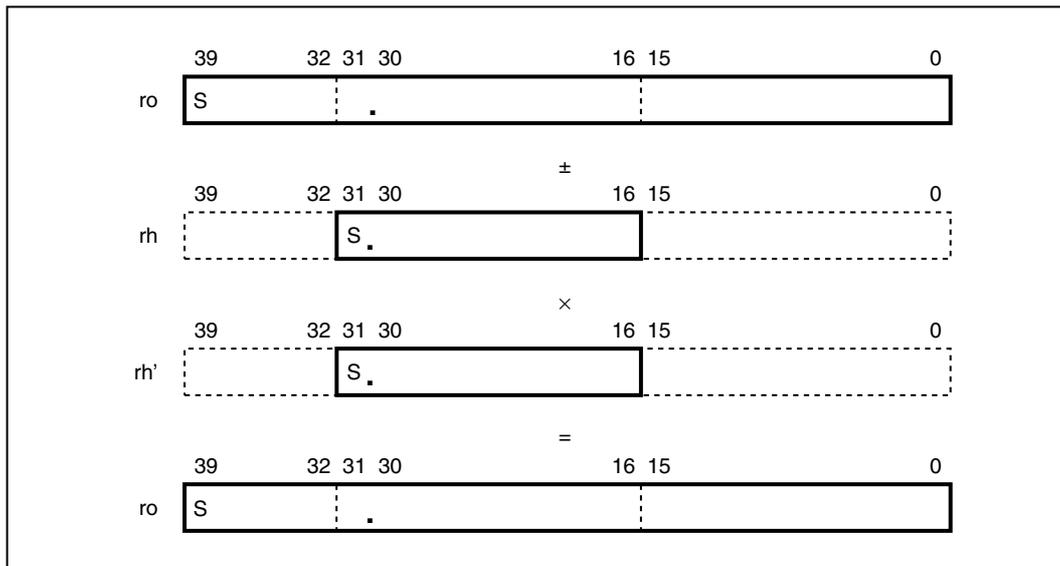
2つの乗算の入力は(符号付き)16ビット固定小数点型です。これらの乗算結果と累算用の40ビット固定小数点データを加算します。関連する命令は次のとおりです。

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

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

図3-42にこの操作のイメージを示します。

図3-42 累乗算



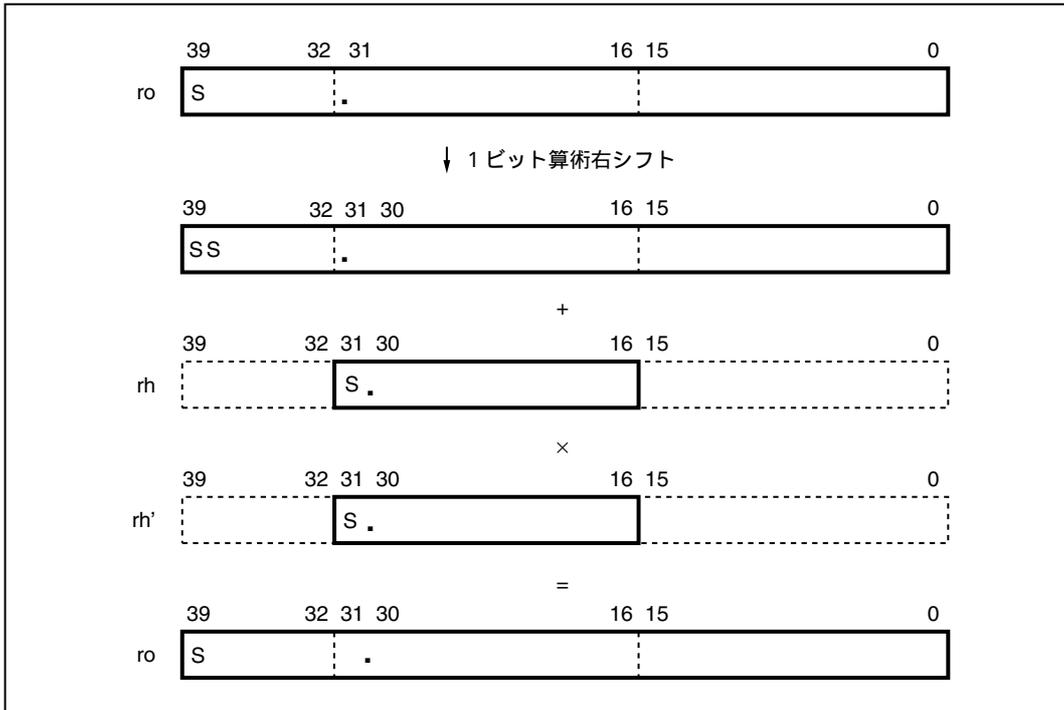
(b) 1ビット・シフト累乗算

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3-43にこの操作のイメージを示します。

図3-43 1ビット・シフト累乗算



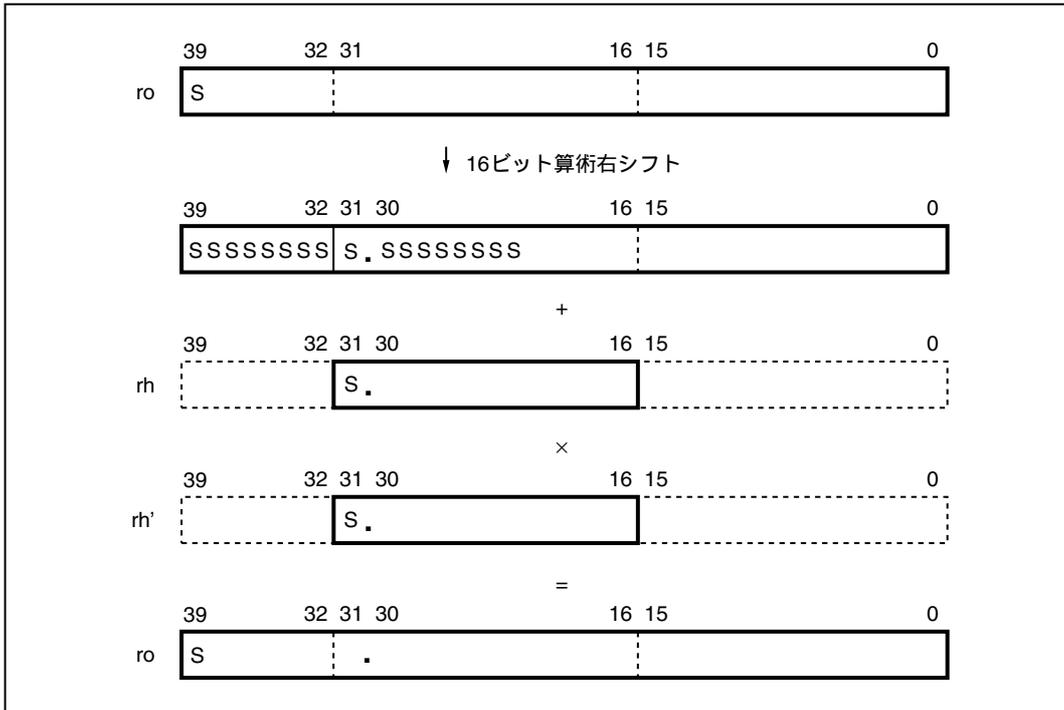
(c) 16ビット・シフト累乗算

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。

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

図3-44にこの操作のイメージを示します。

図3-44 16ビット・シフト累乗算



3.6.4 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)

イミューディエト・アンド命令 (IAND)

オア命令 (OR)

イミューディエト・オア命令 (IOR)

イクスクルーシブ・オア命令 (XOR)

イミューディエト・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 イミュ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 ビットのイミュ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。

3.6.5 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 **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。

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)

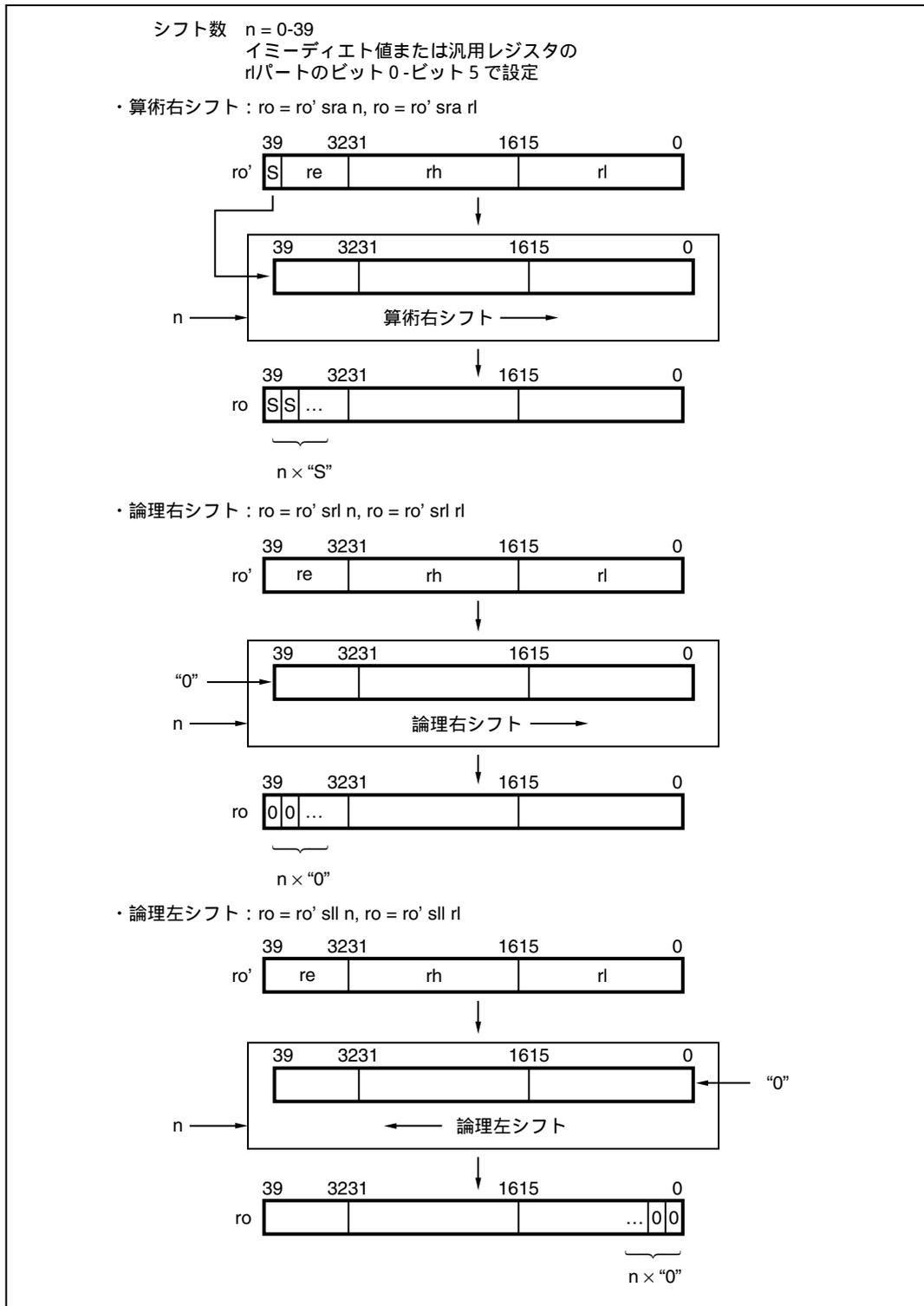
イミディエト論理左シフト命令 (ISLL)

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3-45にBSFTの各機能を示します。

図3-45 BSFT 演算



3.7 ペリフェラル・ユニット

μ PD77111 ファミリは、次に示す基本的なペリフェラル・インタフェース機能を内蔵しています。

ユーザ・アプリケーションからこれらペリフェラルを取り扱うときは、内部メモリ領域にマッピングされたペリフェラル・レジスタにアクセスします。

備考 μ PD77115 のペリフェラルについては、**第6章 μ PD77115 のペリフェラル**を参照してください。

シリアル・インタフェース (スタンダード・シリアル・インタフェース^{注1})

ホスト・インタフェース

汎用入出力ポート

ウェイト制御機能^{注2}

ディバグ・インタフェース^{注3}

オーディオ・シリアル・インタフェース^{注4}

SDカード・インタフェース^{注4}

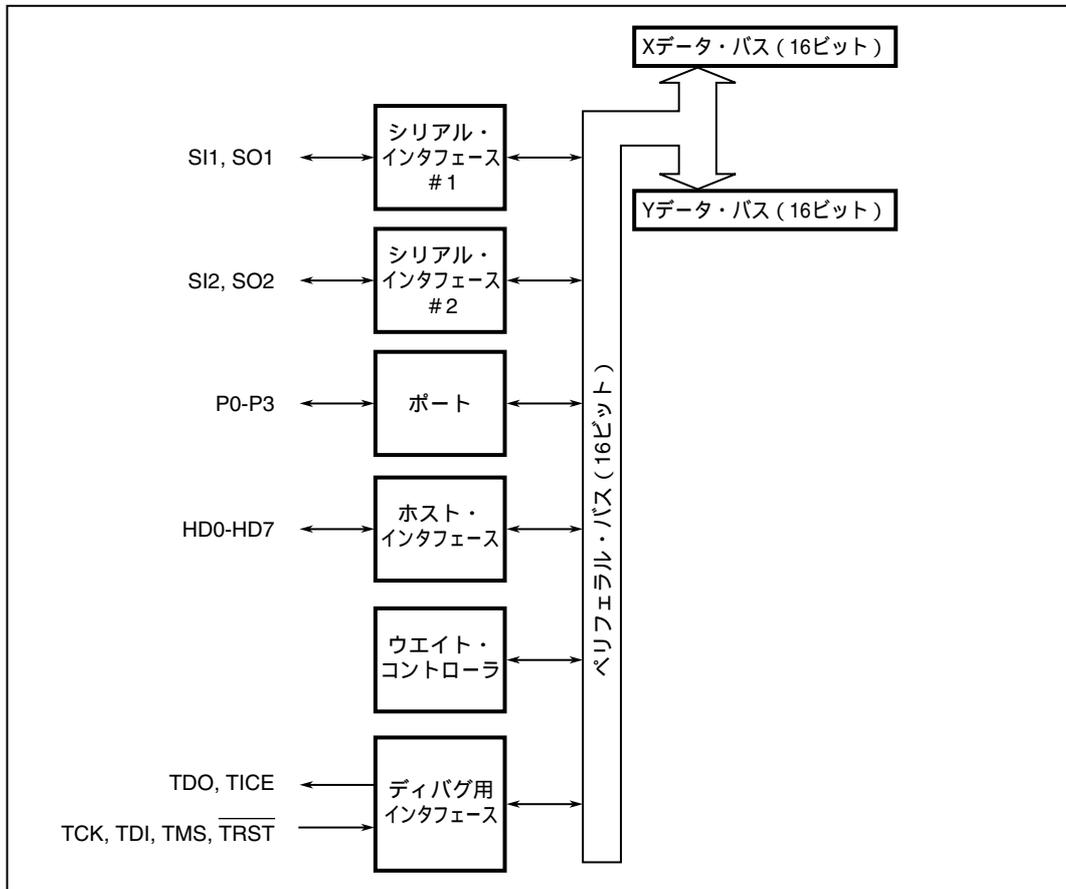
タイマ^{注4}

- 注 1.** μ PD77115 のスタンダード・シリアル・インタフェースは、従来の μ PD77111 ファミリのシリアル・インタフェースと同じです。
- 2.** 一般的な意味でのペリフェラルではありませんが、 μ PD77111 ファミリのなかでペリフェラル制御と同様な取り扱い方式をとっています。
- 3.** ユーザ・プログラムから取り扱うことはできません。
- 4.** μ PD77115 のみの機能です。詳細については、**第6章 μ PD77115 のペリフェラル**を参照してください。

3.7.1 ブロック構成

図3-46にペリフェラル・ユニットのブロック構成を示します。

図3-46 ペリフェラル・ユニット (μ PD77115 以外)



3.7.2 ペリフェラル・レジスタ

内蔵ペリフェラルの取り扱いは、内部データ・メモリ空間にマッピングされたペリフェラル・レジスタにアクセスすることで実現されます。表 3 - 24 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、それらの概要を説明します。

- 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、ユーザによる明示的な定義付けが必要です。
2. これらのレジスタは、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。

表 3 - 24 ペリフェラル・レジスタのメモリ・マッピング

XY メモリ・アドレス	レジスタ名	機 能	ペリフェラル名	ロード/ストア (L/S)
0x3800	SDT1	第 1 シリアル・データ・レジスタ	SIO	L/S
0x3801	SST1	第 1 シリアル・ステータス・レジスタ		
0x3802	SDT2	第 2 シリアル・データ・レジスタ		
0x3803	SST2	第 2 シリアル・ステータス・レジスタ		
0x3804	PDT	ポート・データ・レジスタ	PIO	
0x3805	PCD	ポート・コマンド・レジスタ		
0x3806	HDT	ホスト・データ・レジスタ	HIO	
0x3807	HST	ホスト・ステータス・レジスタ		
0x3808	DWTR	データ・メモリ・ウエイト・サイクル・レジスタ	WTR	
0x3809-0x383F	予約領域	注意 この領域にはアクセスしないでください。	-	-

備考 μ PD77115 のペリフェラル・レジスタについては、6.1 ペリフェラル・レジスタを参照してください。

3.7.3 シリアル・インタフェース

μ PD77111 ファミリは同一構造のシリアル・インタフェースを2チャンネル内蔵しています。
 主な特徴を次に示します。

備考 μ PD77115 は、スタンダード・シリアル・インタフェースとオーディオ・シリアル・インタフェースを各1チャンネル内蔵しています。スタンダード・シリアルインタフェースの構造は、シリアル・インタフェースと同じです。

クロック供給

各チャンネルごとに外部から供給可能

シリアル入力/出力でクロック共用

ビット長

チャンネルごとおよび入出力ごとに8ビットまたは16ビットを設定可能

MSB ファースト, LSB ファーストのデータ・フォーマットを選択可能

内部データ・バス接続

Xデータ・バス, Yデータ・バスと接続

内部ハンドシェーク

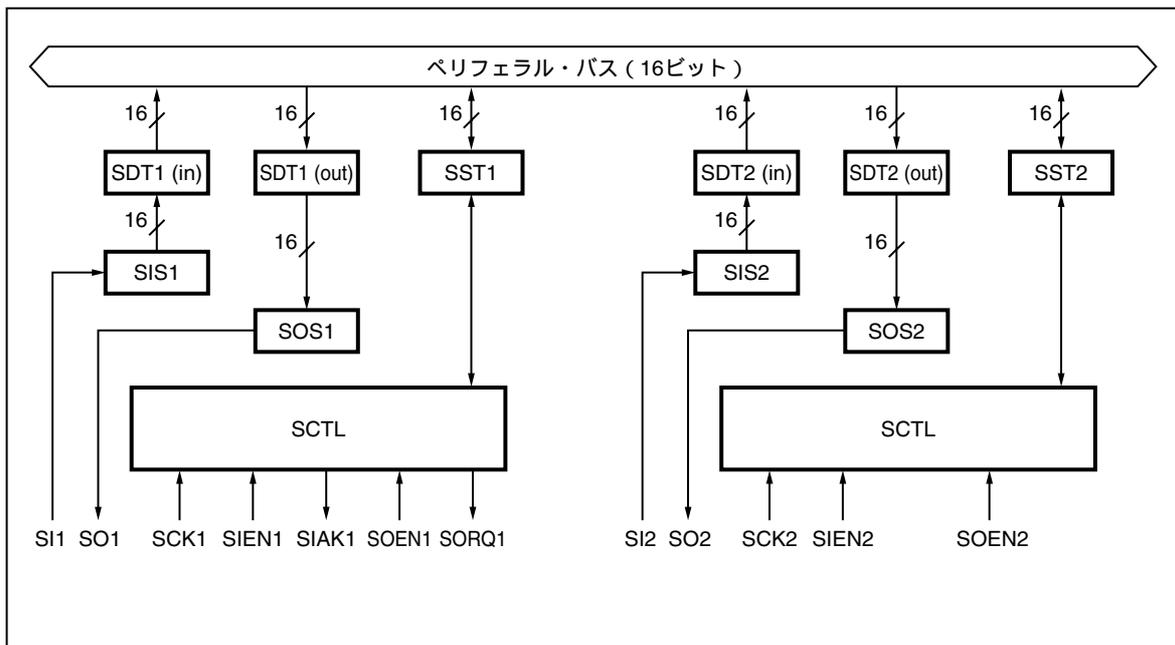
ポーリング, ウェイトまたは割り込みによる

外部ハンドシェーク

専用ステータス信号による

シリアル・インタフェース制御回路 (SCTL) が端子, レジスタの制御をします。シリアル・インタフェースのブロック図を図3-47に示します。

図3-47 シリアル・インタフェース



[シリアル・インタフェースの動作概要]

ここでは、シリアル・インタフェースの機能について、 μ PD77111 ファミリ内部の論理的な動作を説明します (詳細なタイミングについては図 3 - 49, 3 - 50 参照)。

シリアル・インタフェースによるデータ転送のために、入力/出力ともにダブル・バッファを備えています。シリアル入力は、次のレジスタで構成されています。

- ・ SIS レジスタ (serial input shift register) : SI ピンより 1 ビットずつのシリアル・データを入力し、16 ビットのパラレル・データとして SDT (in) に出力します。
- ・ SDT (in) レジスタ (serial data input register) : SIS レジスタより 16 ビットのパラレル・データを入力し、ペリフェラル・バスに 16 ビットのパラレル・データを出力します。

シリアル出力は、次のレジスタで構成されています。

- ・ SDT (out) レジスタ (serial data output register) : ペリフェラル・バスから 16 ビットのパラレル・データを書き込み、SOS に 16 ビットのパラレル・データを出力します。
- ・ SOS レジスタ (serial output shift register) : SDT (out) より 16 ビットのパラレル・データが入力され、SO ピンより 1 ビットずつのシリアル・データを出力します。

外部からは、1 ビットのシリアル・データ入力ピン (SI), 出力ピン (SO) を用いてアクセスします。 μ PD77111 ファミリ内部では、8/16 ビット・パラレルの入力データ・レジスタ SDT (in), 出力データ・レジスタ SDT (out) を用いることによりシリアル入出力を行います。SIS SDT (in) 間および SDT (out) SOS 間のデータ転送は自動的に行われるため、プログラムで直接制御する必要はありません。

シリアル・データ転送において同期をとるために、専用外部ピンおよび各レジスタの状態を監視するための内部フラグが用意されています。

- ・ SIAK (serial input acknowledge) : SIS の状態を監視するための外部ピンです。
 SIAK = ハイ・レベル (SIS が空) : 新たなシリアル・データの入力を開始可能
 SIAK = ロウ・レベル (SIS が空ではない) : SIS にはまだ有効なデータが存在しているため、新たなシリアル・データの入力は不可
- ・ SORQ (serial output request) : SOS の状態を監視するための外部ピンです。
 SORQ = ハイ・レベル (SOS が空ではない) : SOS に出力待ちのデータが存在 (SOEN をハイ・レベルにすることによりデータを出力可能)
 SORQ = ロウ・レベル (SOS が空) : SOS には出力するデータが存在しない。
- ・ SLEF フラグ (serial load enable flag) : SDT (in) の状態を監視するフラグです (シリアル・ステータス・レジスタ (SST) 中のフラグです) 。
 SLEF = 1 (SDT (in) が空ではない) : SDT (in) には有効な入力データが存在
 SLEF = 0 (SDT (in) が空) : SDT (in) からロードできる入力データが存在しない
- ・ SSEF フラグ (serial store enable flag) : SDT (out) の状態を監視するフラグです (シリアル・ステータス・レジスタ (SST) 中のフラグです) 。
 SSEF = 1 (SDT (out) が空) : 新たな出力データを SDT (out) にストア可能
 SSEF = 0 (SDT (out) が空ではない) : SDT (out) にはまだ有効な出力データが存在

適切な制御信号やシリアル・クロック入力を加えたうえで、実際にシリアル入力ピン (SI) およびシリアル出力ピン (SO) によってデータ転送を行えるかどうかは、内部のハードウェアによって自動的に判断されます。

- ・ SOS レジスタにデータがないとき (SORQ = ロウ・レベル) に SO ピンからシリアル・データを出力しようとした場合、SO ピンは、ハイ・インピーダンスになります。
- ・ SIS レジスタに入っているデータが SDT (in) レジスタに転送されるまで (SIS レジスタにデータが入っている (SI AK = ロウ・レベル) ときに、新たにシリアル・データを入力しようとした場合、SIS レジスタに入っているデータは、上書きされません。

これらのハードウェア制御とは別に、SDT (in) レジスタからのロードおよび SDT (out) へのストアは完全にプログラムによって制御されます。有効なデータを上書きしないように、また同じデータを二度引き取らないように、SLEF/SSEF をチェックするか、割り込みを利用するなどの方法を用いて、データを正しくロードもしくはストアするようにしてください。

シリアル・データを連続で入力/出力する場合は、次のことに注意してください。

ステータス・フラグでポーリングする場合

SSEF フラグ (SDT (in) レジスタのステータス・フラグ) または、SSEF フラグ (SDT (out) レジスタのステータス・フラグ) の状態に常に気を付け、データ転送が途切れないようにしてください。

シリアル入力割り込みを使用する場合

割り込みが発生したら、速やかにシリアル・データをロードしてください。

シリアル出力割り込みを使用する場合

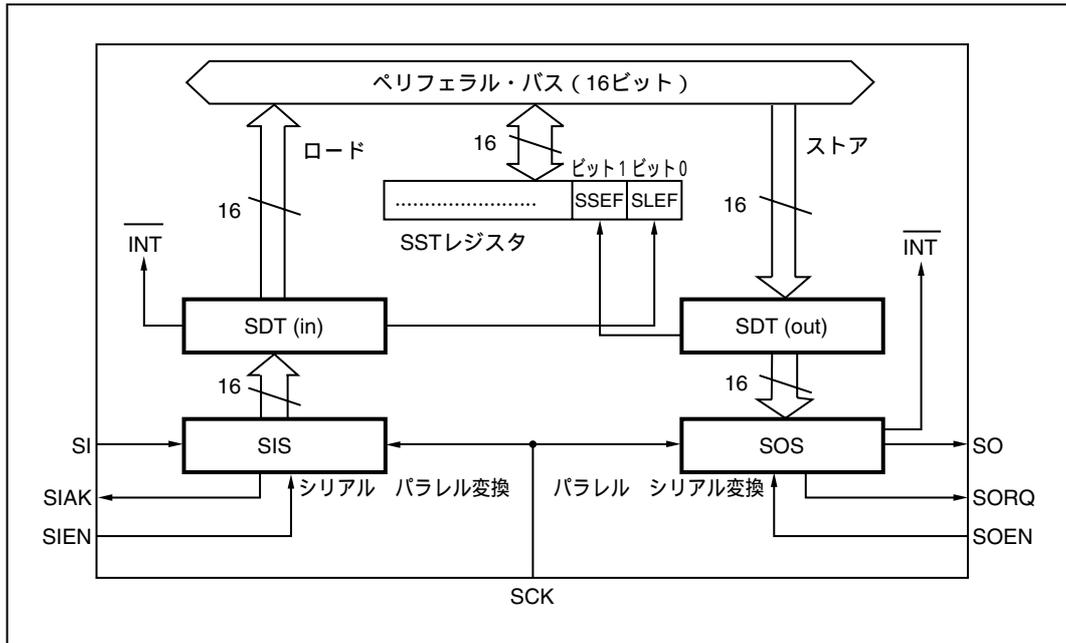
最初のデータ転送 (単発出力時も同様) においては、割り込みが使用できないので、割り込み以外を使用して転送します。次のデータからは割り込みが発生したら、速やかにシリアル・データをストアしてください。また、最後のデータをストアする前に、次の割り込みが発生しないように割り込みを使用禁止にしてください。

次に、シリアル入出力インタフェースのステータスと動作ブロック図を示します。

表 3 - 25 シリアル入出力インタフェースのステータス表示

レジスタ		ステータス表示	ステータス	備考
シリアル入力	SIS	SI AK ピン	ハイ・レベル : 空	シリアル・データ入力可
			ロウ・レベル : データ保持中	シリアル・データ入力不可
	SDT (in)	SLEF フラグ	1 : データ保持中	SDT (in) からデータをロード可
			0 : 空	SDT (in) からデータをロード不可
シリアル出力	SOS	SORQ ピン	ハイ・レベル : データ保持中	シリアル・データ出力可
			ロウ・レベル : 空	シリアル・データ出力不可
	SDT (out)	SSEF フラグ	1 : 空	SDT (out) にデータをストア可
			0 : データ保持中	SDT (out) にデータをストア不可

図3-48 シリアル・インタフェースの動作ブロック図(1チャンネル)



(1) シリアル・インタフェースの端子

μ PD77111 ファミリーはシリアル・インタフェースを2チャンネル内蔵しています。シリアル・インタフェースの端子の後ろにつく数字は各チャンネルの信号に対応します。

シリアル信号はクロックとデータを除き、すべてアクティブ・ハイです。

(a) SCK1, SCK2 (シリアル・クロック-入力)

シリアル・データ入出力用のクロック入力端子です。

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,SCKに同期して行います。

(b) SORQ1 (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。

SCKの立ち上がりに同期して出力が変化します。

シリアル・データをシリアル・データ出力レジスタに書き込むと,アクティブ(ハイ・レベル)になります。

SOEN および SORQ がアクティブになると,シリアル出力を開始します。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。

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。

(c) SOEN1, SOEN2 (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。

SCK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アクティブ(ハイ・レベル)にします。SOEN および SORQ がアクティブになると,シリアル出力を開始します。

(d) SO1, SO2 (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。
SCKの立ち上がりに同期して出力が変化します。
出力が終了するとハイ・インピーダンスになります。

(e) SIEN1, SIEN2 (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。
SCK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アクティブ(ハイ・レベル)にします。SIENおよびSIAKがアクティブになると、シリアル入力を開始します。

(f) SIAK1 (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。
SCK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ったときアクティブ(ハイ・レベル)になります。SIENおよびSIAKがアクティブになると、シリアル入力を開始します。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。
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。

(g) SI1, SI2 (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。
SCKの立ち下がりに同期して入力データをサンプリングします。

表3-26 ハードウェア・リセット中とハードウェア・リセット後の端子の状態

端子名	入出力	リセット中	リセット後
SCK1, SCK2	入力	-	-
SORQ1	出力	ロウ・レベル	ロウ・レベル
SOEN1, SOEN2	入力	-	-
SO1, SO2	出力	ハイ・インピーダンス	ハイ・インピーダンス
SIEN1, SIEN2	入力	-	-
SIAK1	出力	ロウ・レベル	ロウ・レベル
SI1, SI2	入力	-	-

(2) シリアル・インタフェースのレジスタ

μPD77111ファミリはシリアル・インタフェースを2チャンネル内蔵しています。シリアル・インタフェースのレジスタの後ろにつく数字は各チャンネルの信号に対応します。

(a) SDT1, SDT2 (シリアル・データ・レジスタ : 0x3800 : XI : Y, 0x3802 : XI : Y)

シリアル・データ・レジスタ(SDT)は、シリアル・データ入出力するための16ビット・レジスタです。

SDTの値は、レジスタ間転送命令によって入出力ができます。

8ビットで入出力する場合は、SDTの上位8ビットにシリアル・データが入出力されます。

シリアル・データ出力レジスタ (SDT (out))

シリアル出力するデータを設定する 16 ビット・レジスタです。

SDT に対するストア命令を実行すると、このレジスタにペリフェラル・バスからデータを入力します。

SO の出力は、MSB ファーストと LSB ファーストの 2 種類から選択できます。

シリアル出力シフト・レジスタ (SOS) が空くと、このレジスタの値を SOS に設定します。

シリアル・データ入力レジスタ (SDT (in))

シリアル入力されたデータを読み出す 16 ビット・レジスタです。

SDT からのロード命令を実行すると、このレジスタのデータをペリフェラル・バスに出力します。

入力時に、MSB ファーストで出力するか、LSB ファーストで出力するかを選択できます。

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 の値をこのレジスタに設定します。

(b) SST1, SST2 (シリアル・ステータス・レジスタ : 0x3801 : X/ : Y, 0x3803 : X/ : Y)

シリアル・ステータス・レジスタ (SST) は、シリアル入出力のモード設定と、ステータスを表す 16 ビット・レジスタです。

MSB/LSB ファースト、16/8 ビットの指定、 μ PD77111 ファミリ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。

リセット時の値は、0x0002 です。

SST の各ビットの機能を表 3 - 28 に示します。

表 3 - 27 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。

(c) SOS1 , SOS2 (シリアル出力シフト・レジスタ)

シリアル出力シフト・レジスタ (SOS) は、シリアル・クロック SCK の立ち上がりエッジに同期してシリアル・データを SO 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、シリアル・データ出力レジスタ SDT (out) から新しいデータを入力します。

(d) SIS1 , SIS2 (シリアル入力シフト・レジスタ)

シリアル入力シフト・レジスタ (SIS) は、シリアル・クロック SCK の立ち下がりエッジに同期して SI から入力されるデータを受け取りながらシフトする、16 ビットのシフト・レジスタです。

指定したビット数の入力が終わると、シリアル・データ入力レジスタ SDT (in) にデータを出力します。

表 3 - 28 SST (SST1 : 0x3801 : X/ : Y , SST2 : 0x3803 : X/ : Y) の機能 (1/2)

ビット	名 称	ロード/ストア (L/S)	ビットの機能
15	SOTF	L/S	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 1 : LSB ファーストでシリアル出力
14	SITF	L/S	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 1 : LSB ファーストでシリアル入力
13	SOBL	L/S	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 1 : 8 ビットでシリアル出力
12	SIBL	L/S	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 1 : 8 ビットでシリアル入力
11	SSWE	L/S	SDT 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 1 : ストア・ウェイトを使用する SSEF が 0 のときに、 μ PD77111 ファミリが SDT (out) にデータをストアしようとした場合に、ウェイトを挿入する。
10	SLWE	L/S	SDT 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 1 : ロード・ウェイトを使用する SLEF が 0 のときに、 μ PD77111 ファミリが SDT (in) にデータをロードしようとした場合に、ウェイトを挿入する。
9	SICM [※]	L/S	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。
8	SIEF [※]	L/S	シリアル単発入力許可フラグ 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 フラグは次の命令サイクルで自動的にリセットされる。
7-4	予約	-	予約ビット このビットには値を設定できない。 読み出し時は不定。

注 SICM, SIEF の組み合わせ使用例は表 3 - 29 のとおりです。なお、スピーチ・データなどの連続データが入力される場合は、2 の状態 (SICM = 1, SIEF = 0) で使用してください。

備考 ハードウェア・リセット後の SST の設定 : 0x0002 ;

入出力ワード長 : 16 ビット

入出力 : MSB ファースト

ウェイト機能 : 使用しない

シリアル入力連続モード : 単発モード

シリアル入力 : 行わない

表3-28 SST (SST1 : 0x3801 : X/ : Y , SST2 : 0x3803 : X/ : Y) の機能 (2/2)

ビット	名称	ロード/ストア (L/S)	ビットの機能
3	SSER	L/S	SDT ストア・エラー・フラグ 0 : エラーなし 1 : エラー SSEF が 0 のときに, μ PD77111 ファミリが SDT (out) にデータをストアした場合に, 1 になる。 一度セットされると, μ PD77111 ファミリが 0 を書き込むまで変化しない。
2	SLER	L/S	SDT ロード・エラー・フラグ 0 : エラーなし 1 : エラー SLEF が 0 のときに, μ PD77111 ファミリが SDT (in) からデータをロードした場合に, 1 になる。 一度セットされると, μ PD77111 ファミリが 0 を書き込むまで変化しない。
1	SSEF	L	SDT ストア許可フラグ SDT (out) 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 PD77111 ファミリが SDT (out) にデータをストアした場合 0 になる。
0	SLEF	L	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 (in) に転送した場合に 1 になる。 μ PD77111 ファミリが SDT (in) からデータをロードした場合 0 になる。

備考 ハードウェア・リセット後の SST の設定 : 0x0002 ;

入出力ワード長 : 16 ビット

入出力 : MSB ファースト

ウェイト機能 : 使用しない

シリアル入力連続モード : 単発モード

シリアル入力 : 行わない

表3-29 SICM ビットと SIEF ビットの組み合わせ

組み合わせ例	ビット9 SICM	ビット8 SIEF	機能
1	0	0	<ul style="list-style-type: none"> ・状態遷移モードです。また、シリアル入力を行わないときもこのモードになります。 ・SIAC がロウ・レベルになります。ただし、SIAC がハイ・レベルのときにこのモードにセットしても、シリアル入力を開始するまではハイ・レベルのままです。
2	1	0	<ul style="list-style-type: none"> ・シリアル連続入力モードです。 ・シリアル入力可能な場合、SIAC はハイ・レベルを出力します。シリアル入力開始後、SIAC はロウ・レベルになります。SDT (in) からデータがロードされると再びシリアル入力可能になり、SIAC はハイ・レベルを出力します。完全なデータ・ワードがシフトされたときに SDT (in) が空の場合、SIS の内容はただちに (SCK に同期して) SDT (in) に転送され、SIAC は、ハイ・レベルになります (図3-50 (a) 参照)。
3	0	1	<ul style="list-style-type: none"> ・シリアル単発入力モードです。 ・シリアル入力可能な場合、SIAC はハイ・レベルを出力します。シリアル入力開始後 SIAC はロウ・レベルになります。SDT (in) からデータがロードされても SIAC はロウ・レベルのままです。 ・1 にセットされた SIEF フラグは、次の命令サイクルで自動的にリセットされます (図3-50 (b) 参照)。
4	1	1	この組み合わせは設定しないでください。

(3) シリアル・インタフェースのタイミング**(a) 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 3 - 49 (a) と (b) 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、SORQ (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 SOEN (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 SCK (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)、SCK の続く立ち上がりエッジで SORQ がロウ・レベルになり (c)、同じ立ち上がりエッジから SO (シリアル・データ出力端子) にデータ出力が開始される (d)。

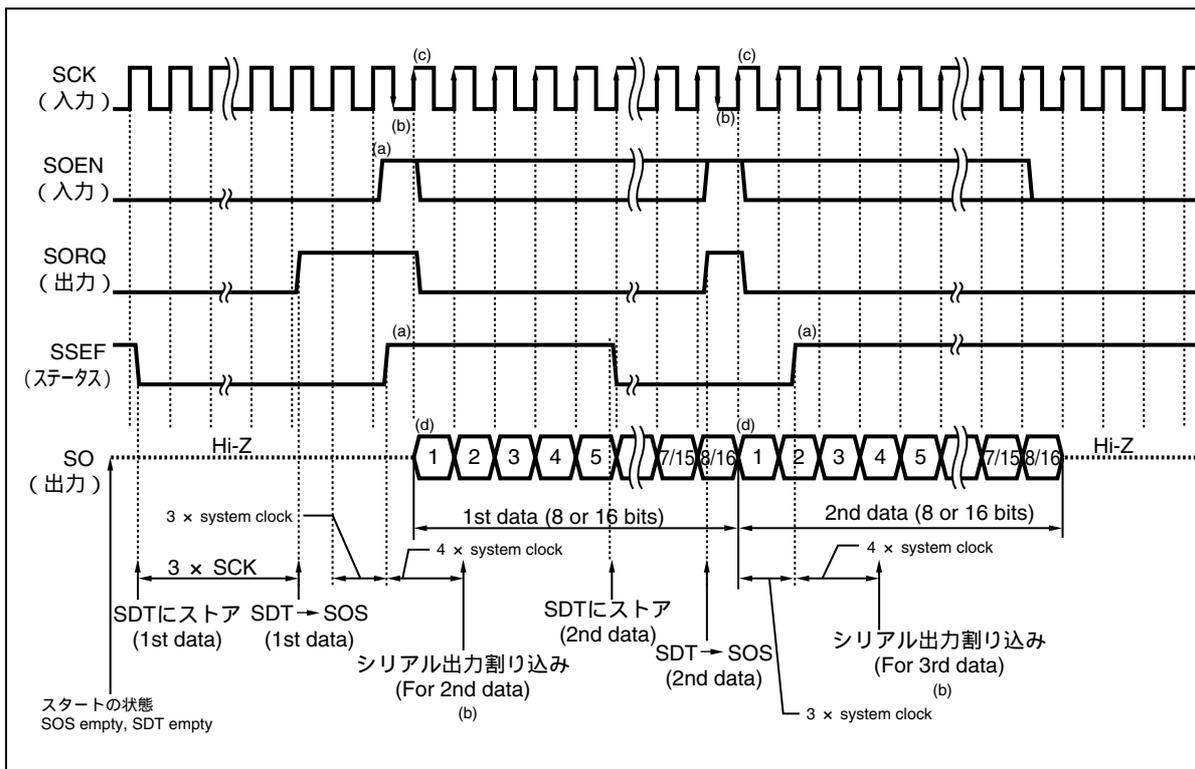
SDT が空になったあと、SEEF は 1 にセットされて、次のデータ書き込みが可能であることをアプリケーションに知らせ (a)、SSWE = 1 のもとで有効化されていた SDT ストア・ウエイト機能は解除される。またこのとき SO による割り込み要求を発生する (b) が割り込みの要因別許可フラグや EI の状態により、有効な割り込みとして処理されたり、または記憶されることになる (3.4.4 割り込み参照)。

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ければ、続く SCK の立ち上がりエッジで、SO はハイ・インピーダンスとなる^{注2}。

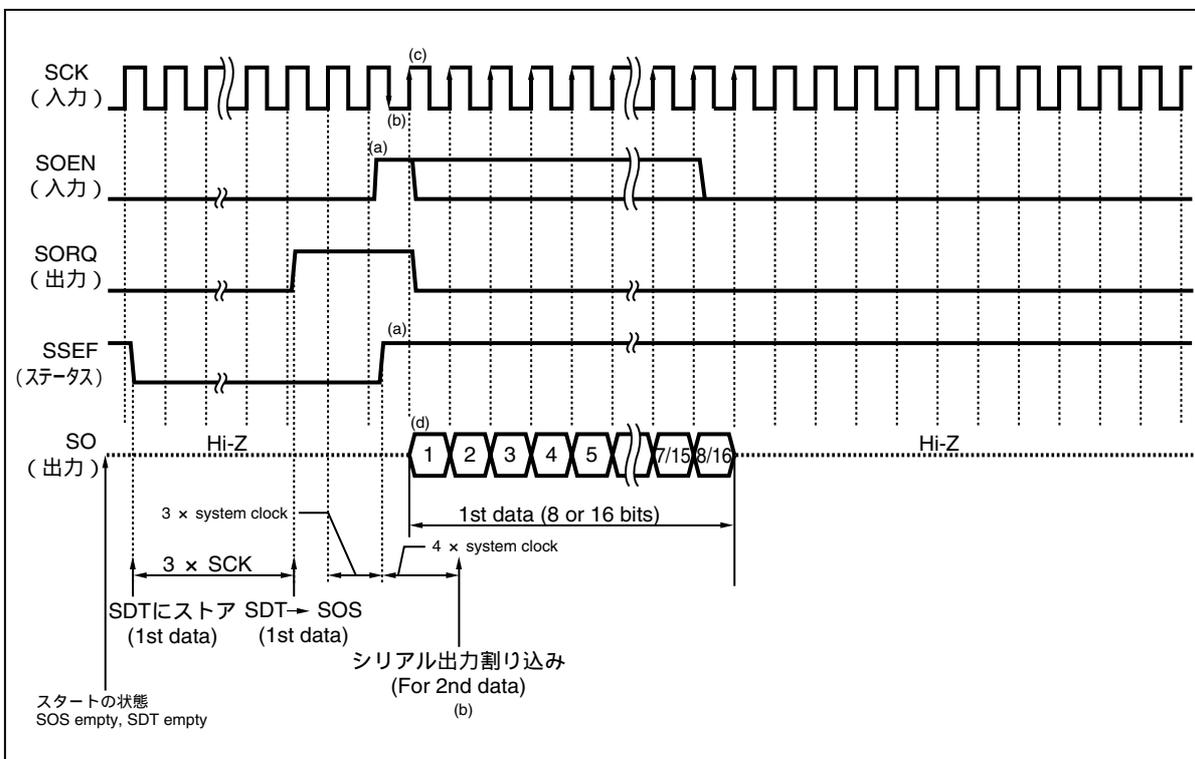
- 注 1.** SOEN がアクティブになるまえに、少なくとも 3 回の SCK 立ち上がりエッジが経過していなければなりません。データの入出力時のみにクロックをパースト的に供給するシステム構成ではこのことに十分注意してください。
- 2.** 厳密には、最終ビット出力までにすでに次のデータが供給されており、かつ最終ビット出力サイクル中、SCK の立ち下がりまでに SOEN がアクティブとなり、有効にサンプルされた場合、SO はハイ・インピーダンスとならずに連続して次のデータを出力します (図 3 - 49 (a) 参照)。
したがって、最終ビットを出力したあと、少なくとも 1 回の SCK 立ち上がりエッジを供給しなければなりません。

図3-49 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



(b) 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 3 - 50 (a) と (b) 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。

SI AK (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 SI EN (シリアル入力キネープ端子) 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 SI EN の変化は、直後の SCK の立ち下がりエッジでサンプリングされ^{※1} (a), SCK の続く立ち上がりエッジで SI AK がロウ・レベルになり (b), 同じ SCK サイクルの立ち下がりエッジから SI (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、SI から SIS (シリアル入力シフト・レジスタ) へ 1 ビットずつ SCK 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 SCK 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 SI AK 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。

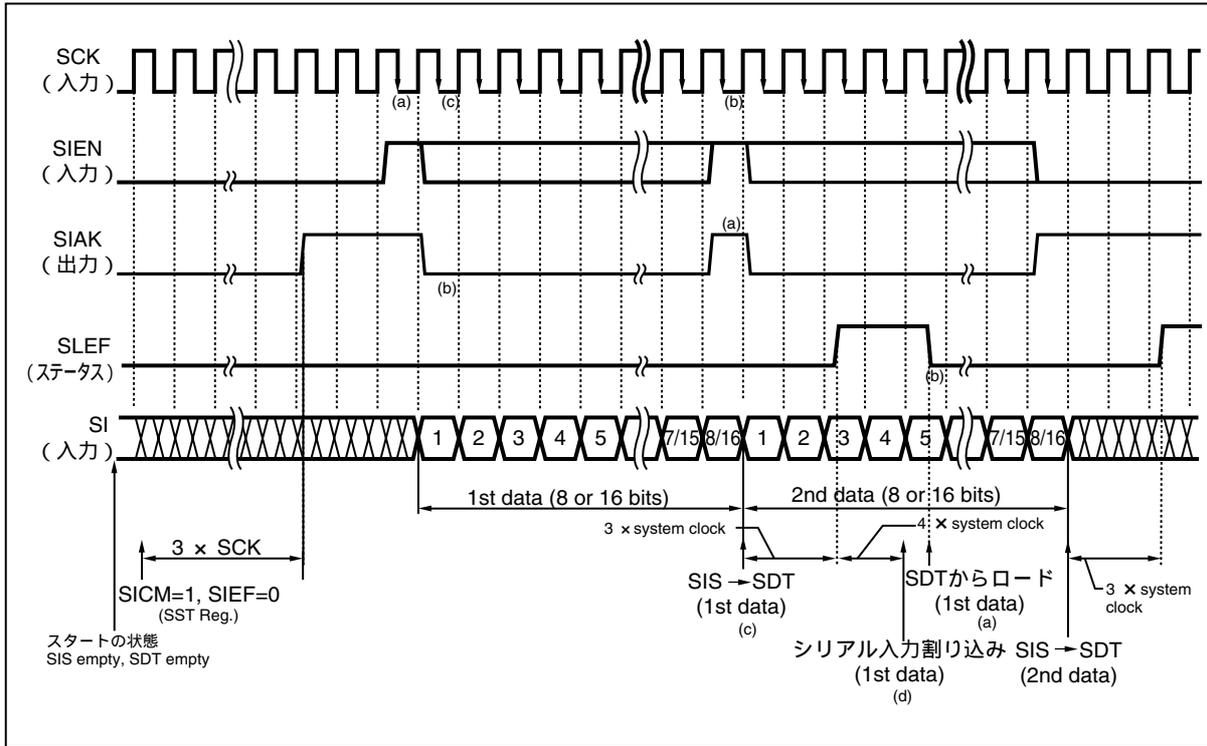
その後、SLEF = 1 になってアプリケーション・プログラムにシリアル入力データ・ワードの成立を知らせる (c)。このとき SLWE (SDT : ロード・ウエイト許可ビット) = 1 でデータ・ウエイト状態にあった場合、ウエイトは解除される。またこのとき、SI による割り込み要求を発生するが (d), 割り込みの要因許可フラグや EI の状態により、有効な割り込みとして処理されたり、または記憶されることになる (3.4.4 割り込み参照)。

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。

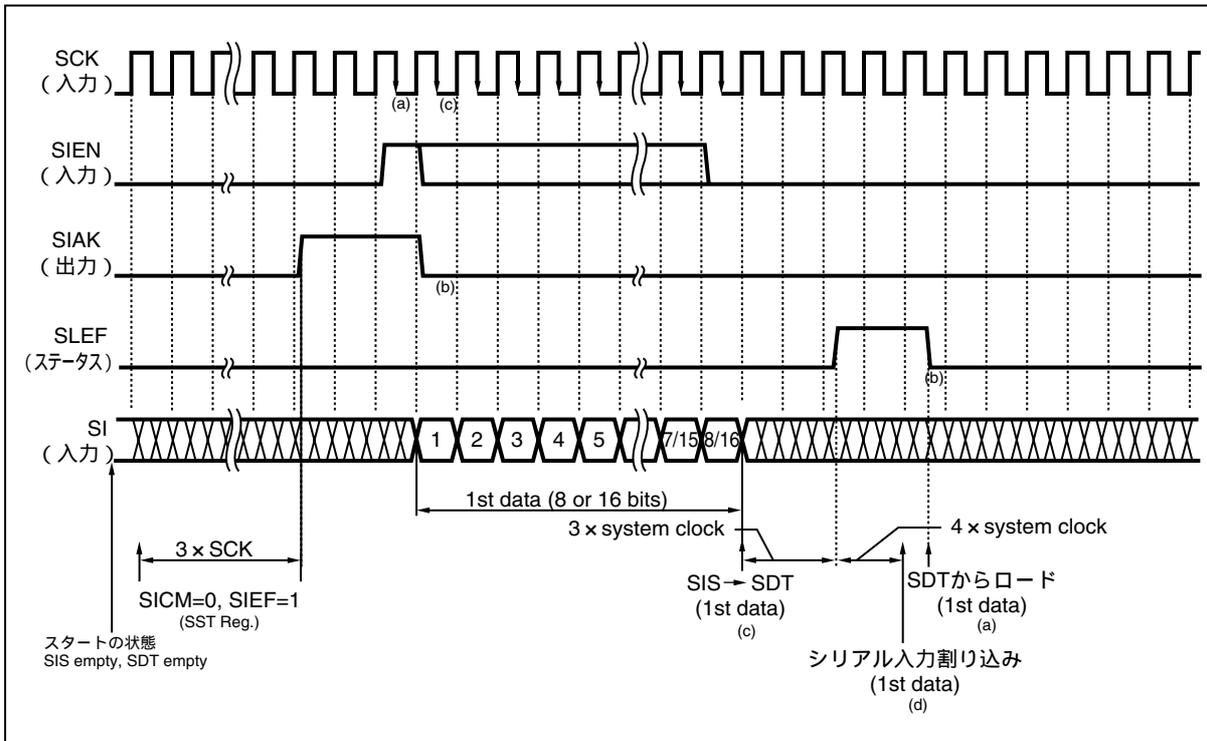
- 注 1. SI EN がアクティブになる前に、少なくとも 3 回の SCK 立ち上がりエッジが経過していなければなりません。シリアル入出力部のハードウェアは、SCK をタイミング・クロックとしてパイプライン状に動作します。データの入出力時のみにクロックをバースト的に供給するシステム構成では、このことに十分注意してください。連続してクロックを供給するようなシステム構成の場合は、リセット後の最初のデータについてだけ注意してください。
- また、このような理由から、最終ビットが取り込まれたあと、少なくとも 2 回の SCK 立ち上がりエッジを供給しなければなりません。
2. 最終ビット入力サイクル中、SCK の立ち下がりまでに SI EN がアクティブとなって有効にサンプルされた場合、連続する次の SCK サイクルから次のデータを取り込みます (図 3 - 50 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3-50 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

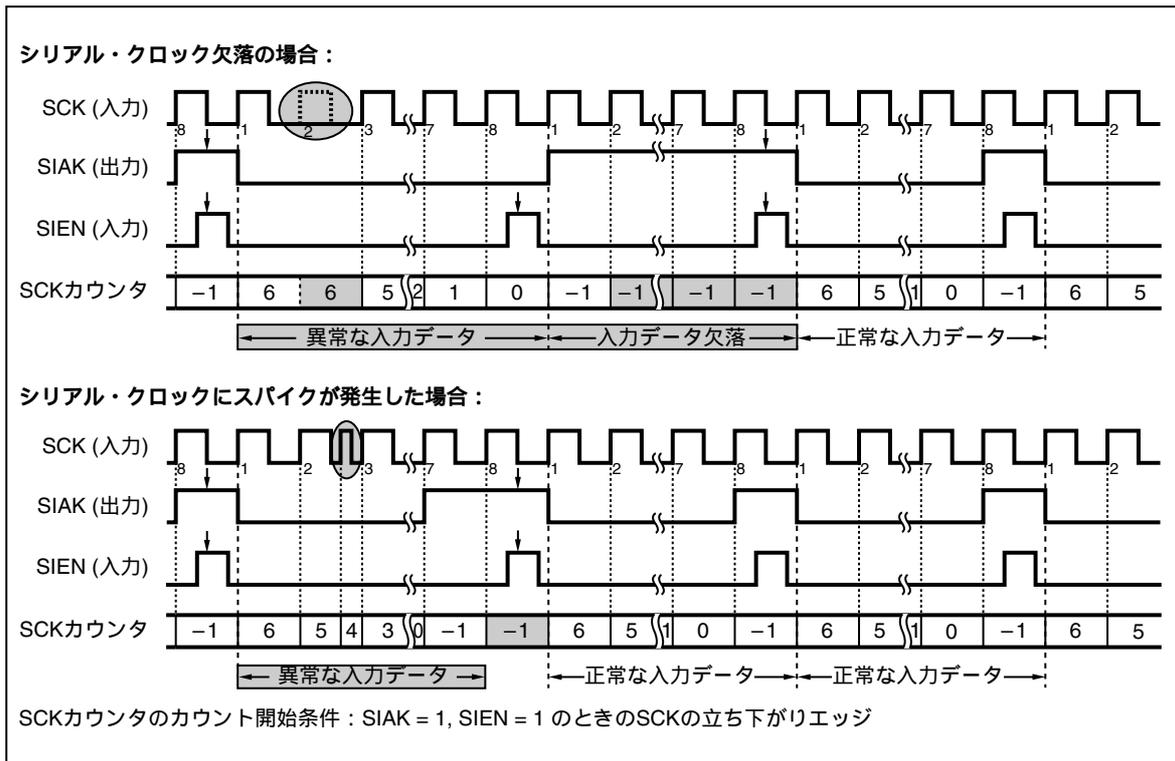


(c) 規格外のシリアル・クロックによる入出力タイミング

図3-51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。

SIEN, SOEN 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3-51のように、SIEN, SOEN 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、SCK をカウントすることによって、SIEN を入力している例を示しましたが、SIAK の状態をみて SIEN を入力する方が確実です。

図3-51 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

μPD77111 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、ハンドシェー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ー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 */
ROL = SR          ;
R0 = R0 | 0x0030  ;
SR = ROL          ;

ROL = 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、DSP 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 (3.4.4 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 (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、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、あるいは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) パーストでデータを転送するときに、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 SDT に転送する直前の命令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。

```

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              ;
    
```

3.7.4 ホスト・インタフェース

μ PD77111 ファミリは、外部のホストCPUおよびDMAコントローラなどのデータ転送を行うためのホスト・インタフェースを内蔵しています。

主な特徴を次に示します。

8 ビット・パラレル・ポート

16 ビット・パラレル・ポート^注

データ範囲

上位 8 ビットまたは下位 8 ビットをアドレスで選択

内部データ・バスの接続

X データ・バスおよび Y データ・バスに接続

内部 16 ビット，外部 8 ビット構成

外部デバイスとは，8 ビット・データ・バスでインタフェース

内部ハンドシェーク

ポーリング，ウエイトまたは割り込みによる

外部ハンドシェーク

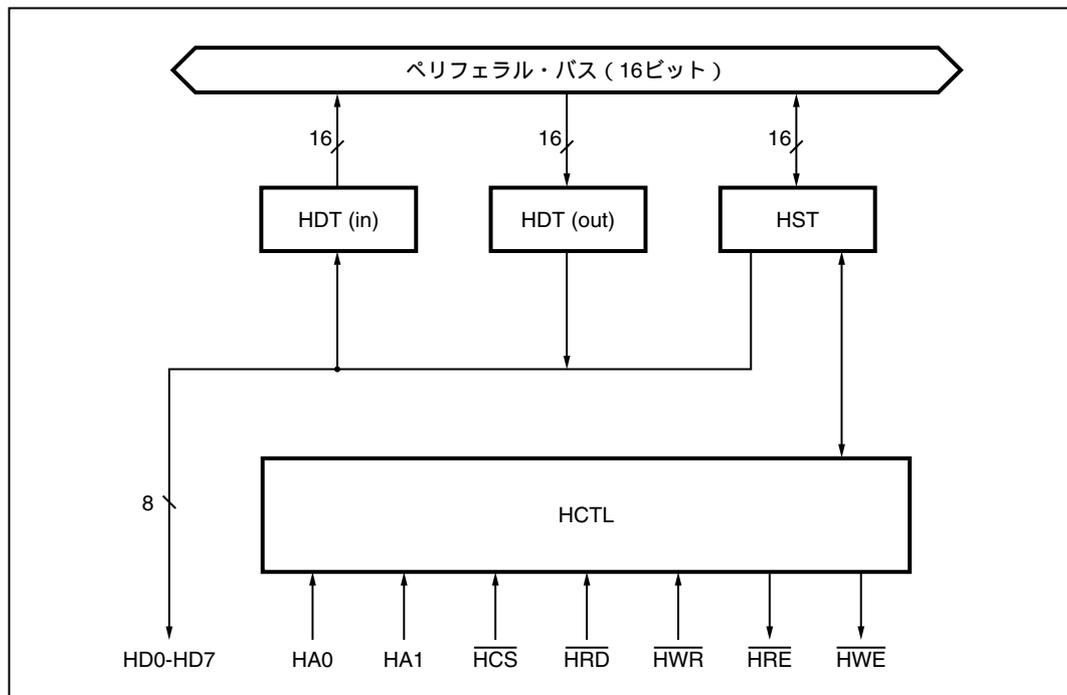
専用ステータス信号による

注 μ PD77115 のみの機能です。6.3 ホスト・インタフェースもあわせてご覧ください。

ホスト・インタフェース制御回路 (HCTL) が端子，レジスタの制御をします。

ホスト・インタフェースのブロック図を図 3 - 52 に示します。

図 3 - 52 ホスト・インタフェース



[ホスト・インタフェースの動作概要]

ここでは、ホスト・インタフェースの機能について μ PD77111 ファミリ内部の論理的な動作を説明します（詳細なタイミングについては図 3 - 54, 3 - 55 参照）。

ホスト・インタフェースによるデータ転送のために、入力/出力ともにバッファを一段備えています。

ホスト入力、ホスト出力は次のレジスタで構成されています。

- ・ HDT (in) レジスタ (host data input register) : HD0-HD7 ピンより各 8 ビットのパラレル・データ (上位バイトと下位バイト) を入力し、16 ビットのパラレル・データとして、ペリフェラル・バスに 16 ビットのパラレル・データを出力します。
- ・ HDT (out) レジスタ (host data output register) : ペリフェラル・バスから 16 ビットのパラレル・データを書き込み、各 8 ビットのパラレル・データ (上位バイトと下位バイト) を HD0-HD7 ピンより出力します。

外部とは 8 ビットのホスト・データ入出力ピンを用いてアクセス可能です。 μ PD77111 ファミリ内部ではパラレル入力レジスタ HDT (in), 出力レジスタ HDT (out) を用いることによりアクセスが可能です。

ホスト・データ転送において同期をとるために、専用外部ピンおよび各レジスタの状態を監視するための内部フラグが用意されています。

- ・ $\overline{\text{HWE}}$ (host write enable), HWEF (host write enable flag) : HDT (in) の状態を監視するための外部ピンとフラグ (ホスト・ステータス・レジスタのフラグ) です。

$\overline{\text{HWE}}$ = ハイ・レベル, HWEF = 0 (HDT (in) が空ではない) HDT (in) にはまだ有効なデータが存在。

ホストは HDT (in) に新たなデータの書き込みが不可。 μ PD77111 ファミリは HDT (in) からのロードが可能。

$\overline{\text{HWE}}$ = ロウ・レベル, HWEF = 1 (HDT (in) が空)

ホストは HDT (in) に新たなデータの書き込みが可能。 μ PD77111 ファミリは HDT (in) からのロードが不可。

- ・ $\overline{\text{HRE}}$ (host read enable), HREF (host read enable flag) : HDT (out) の状態を監視するための外部ピンとフラグ (ホスト・ステータス・レジスタのフラグ) です。

$\overline{\text{HRE}}$ = ハイ・レベル, HREF = 0 (HDT (out) が空)

HDT (out) には有効なデータが存在しない。ホストは HDT (out) からの読み出しが不可。 μ PD77111 ファミリは HDT (out) へ出力データをストア可能。

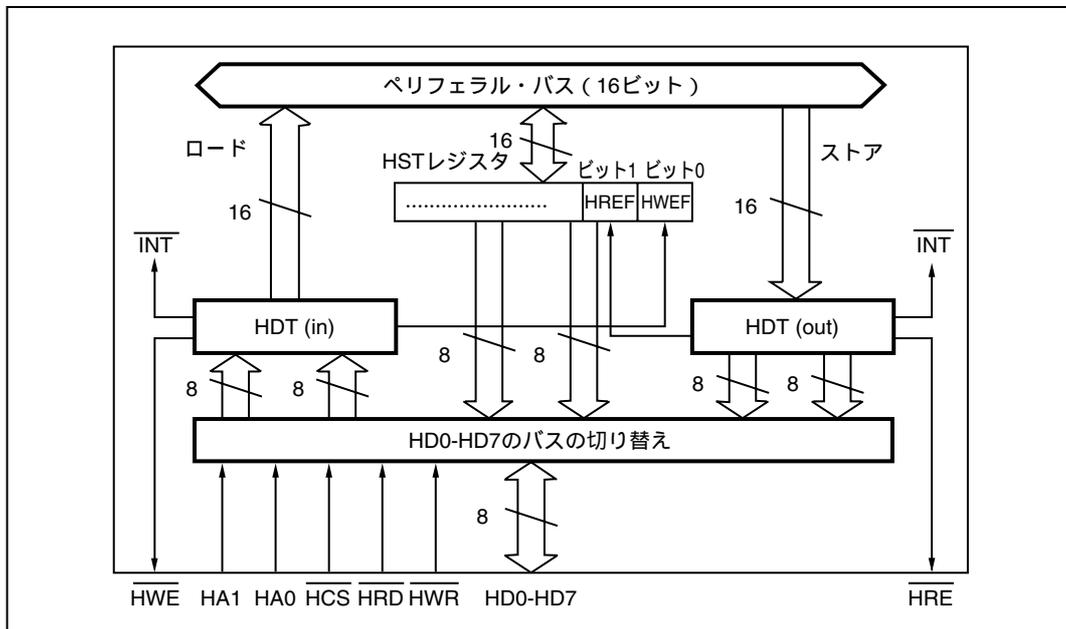
$\overline{\text{HRE}}$ = ロウ・レベル, HREF = 1 (HDT (out) が空ではない)

ホストは HDT (out) から新たなデータの読み出しが可能。 μ PD77111 ファミリは HDT (out) へ出力データのストア不可。

表 3 - 30 ホスト入出力インタフェースのステータス表示

レジスタ	ステータス表示	ステータス	備 考	
ホスト・ライト	HDT(in)	HWE ピン	ハイ・レベル：データ保持中	ホストは HDT(in)に書き込み不可
		ロウ・レベル：空	ホストは HDT(in)に書き込み可	
	HWEF フラグ	1：空	HDT(in)からデータをロード不可	
		0：データ保持中	HDT(in)からデータをロード可	
ホスト・リード	HDT(out)	HRE ピン	ハイ・レベル：空	ホストが読むべき新たなデータは HDT(out)に存在しない
		ロウ・レベル：データ保持中	ホストは HDT(out)を読みだし可	
	HREF フラグ	1：データ保持中	データを HDT(out)にストア不可	
		0：空	データを HDT(out)にストア可	

図 3 - 53 ホスト・インタフェースの動作ブロック図



(1) ホスト・インタフェースの端子

ホスト・インタフェースの端子はすべてアクティブ・ロウです。

(a) HCS (ホスト・チップ・セレクト-入力)

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 CPU が、ホスト・インタフェースのレジスタをアクセスする間はアクティブ(ロウ・レベル)になります。

(b) HA1, HA0 (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 CPU 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期間は変化させないでください。

(c) $\overline{\text{HRD}}$ (ホスト・リード・ストロブ-入力)

ホスト・インタフェースのリード・ストロブ信号入力端子です。

ホスト CPU 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しないでください。

(d) $\overline{\text{HWR}}$ (ホスト・ライト・ストロブ-入力)

ホスト・インタフェースのライト・ストロブ信号入力端子です。

ホスト CPU 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しないでください。

(e) HD0-HD7 (ホスト・データ-入力/出力)

ホスト・インタフェースのデータ入出力端子です。

ホスト CPU が、ホスト・インタフェースのレジスタにアクセスする場合、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・インピーダンスになります。

(f) $\overline{\text{HRE}}$ (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 読み出し可能な場合アクティブ (ロウ・レベル) になり、HDT の上位バイトのデータを読み出すとき、 $\overline{\text{HRD}}$ 端子の入力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。HDT の下位バイトをアクセスしても変化しません。

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。

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

HDT に対する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はアクティブ (ロウ・レベル) になり、HDT の上位バイトにデータを書き込むとき、 $\overline{\text{HWR}}$ 端子入力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。

HDT の下位バイトのデータをアクセスしても変化しません。

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。

表 3 - 31 ハードウェア・リセット中とハードウェア・リセット後の端子の状態

端子名	入出力	リセット中	リセット後
$\overline{\text{HCS}}$	入力	-	-
HA0, HA1			
$\overline{\text{HRD}}$			
$\overline{\text{HWR}}$			
HD0-HD7	入出力	ハイ・インピーダンス ($\overline{\text{HCS}}$ 端子インアクティブ時)	
$\overline{\text{HRE}}$	出力	ハイ・レベル	
$\overline{\text{HWE}}$			

(2) ホスト・インタフェースのレジスタ

(a) ホスト・データ・レジスタ (HDT-0x3806 : X/Y)

ホスト・データ・レジスタ (HDT) は、ホスト・インタフェースからデータを入出力するための 16 ビット・レジスタです。

HDT の値は、ロード/ストア命令を使ってデータの入出力ができます。

ホスト・データ出力レジスタ (HDT (out))

ホスト・インタフェースから出力するデータを設定する 16 ビット・レジスタです。

HDT に対するストア命令を実行すると、このレジスタにペリフェラル・バスからデータを入力します。

外部デバイスが読み出す場合、HA0 で上位 8 ビットまたは下位 8 ビットを指定します。

ホスト・データ入力レジスタ (HDT (in))

ホスト・インタフェースから入力するデータを設定する 16 ビット・レジスタです。

HDT からのロード命令を実行すると、このレジスタのデータをペリフェラル・バスに出力します。

外部デバイスが書き込む場合、HA0 で上位 8 ビットまたは下位 8 ビットを指定します。

(b) ホスト・ステータス・レジスタ (HST-0x3807 : X/Y)

ホスト・ステータス・レジスタ HST は、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・レジスタです。

ホスト CPU とホスト・インタフェース間およびホスト・インタフェースと μ PD77111 ファミリの間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 の値は、ロード/ストア命令を使って、データの入出力ができます。

外部から読み出すときは、HA0 で上位 8 ビットか下位 8 ビットを指定します。

リセットにより HST の値は 0x0301 になります。

表 3 - 32 に HST の各ビットの機能を、表 3 - 33 にホスト入出力エラー・フラグの設定条件を示します。

表 3 - 32 HST (0x3807 : X/ : Y) の機能 (1/2)

ビット	名 称	ホストからの リード/ライト (R/W)	μ PD77111 ファミリからの ロード/ストア (L/S)	ビットの機能
15 11	予約	-	-	予約ビット このビットには値を設定できない。 読み出し時は不定。
10	HAWE	R	L/S	HDT アクセス・ウエイト許可ビット 0: ウエイトを使用しない 1: ウエイトを使用する HREF が 1 のときに、ユーザが HDT (out) にデータをストアしようとした場合、ウエイトが挿入される。 HWEF が 1 のときに、ユーザが HDT (in) からデータをロードしようとした場合、ウエイトが挿入される。
9	HREM	R	L/S	HRE マスク・ビット 0: マスクしない HREF の状態 (ビット 1 参照) に応じて、 $\overline{\text{HRE}}$ が変化する。 1: マスクする $\overline{\text{HRE}}$ はインアクティブ (ハイ・レベル) となる。
8	HWEM	R	L/S	HWE マスク・ビット 0: マスクしない HWEF の状態 (ビット 0 参照) に応じて、 $\overline{\text{HWE}}$ が変化する。 1: マスクする $\overline{\text{HWE}}$ はインアクティブ (ハイ・レベル) となる。
7 6	UF1 UF0	R	L/S	ユーザ・フラグ
5	HRER	R	L/S	ホスト・リード・エラー・フラグ 0: エラーなし 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プログラムで 0 を書き込むまで変化しない。
4	HWER	R	L/S	ホスト・ライト・エラー・フラグ 0: エラーなし 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プログラムで 0 を書き込むまで変化しない。

備考 ハードウェア・リセット後の HST の設定 : 0x0301

- ・ウエイト機能 : 使用しない
- ・HRE/HWE マスク : マスクする
- ・ホスト・ライト可
- ・ホスト・リード不可

表 3 - 32 HST (0x3807 : X/ : Y) の機能 (2/2)

ビット	名 称	ホストからの リード/ライト (R/W)	μ PD77111 ファミリからの ロード/ストア (L/S)	ビットの機能
3	HSER	R	L/S	HDT ストア・エラー・フラグ 0: エラーなし 1: エラー HREF が 1 のときに, μ PD77111 ファミリが HDT にストアした場合に 1 になる。 一度セットされるとプログラムで 0 を書き込むまで変化しない。
2	HLER	R	L/S	HDT ロード・エラー・フラグ 0: エラーなし 1: エラー HWEF が 1 のときに, μ PD77111 ファミリが HDT からロードした場合に 1 になる。 一度セットされるとプログラムで 0 を書き込むまで変化しない。
1	HREF	R	L	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 PD77111 ファミリが HDT にストアした場合 1 になる。ホスト CPU が HDT の上位バイトを読み出した場合 0 になる。 書き込み時は無視される。
0	HWEF	R	L	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 PD77111 ファミリが HDT からロードした場合 1 になる。ホスト CPU が HDT の上位バイトに書き込んだ場合 0 になる。 書き込み時は無視される。

備考 ハードウェア・リセット後の HST の設定 : 0x0301

- ・ウエイト機能 : 使用しない
- ・HRE/HWE マスク : マスクする
- ・ホスト・ライト可
- ・ホスト・リード不可

表 3 - 33 ホスト入出力エラー・フラグ設定条件

エラー・フラグ名	要 因	解除条件
HRER	HREF = 0 においてホスト・リード	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
HWER	HWEF = 0 においてホスト・ライト	
HSER	HREF = 1 において HDT にストア	
HLER	HWEF = 1 において HDT からロード	

(3) ホストから見たホスト・インタフェースのレジスタ

ホスト CPU は DSP の HA0, HA1 端子を用いることにより, アクセス対象としてホスト・ステータス・レジスタ HST またはホスト・データ・レジスタ HDT の上位バイトまたは下位バイトを設定可能です。

表 3 - 34 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 3 - 34 ホスト・インタフェース用レジスタの選択

HCS	HRD	HWR	HA1	HA0	転送対象のレジスタ	バイト
0	0	0	x	x	設定禁止	-
0	0	1	0	0	HDT(out)	下位 8 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
0	0	1	1	1		上位 8 ビット
0	1	0	0	0	HDT(in)	下位 8 ビット
0	1	0	0	1		上位 8 ビット
0	1	0	1	x	設定禁止	-
0	1	1	x	x	対象なし	-
1	x	x	x	x		-

(4) ホスト・インタフェースのタイミング

(a) ホスト・リードの動作 (μ PD77111 ファミリ ホスト)

μ PD77111 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 3 - 54 に示します)。

μ PD77111 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。

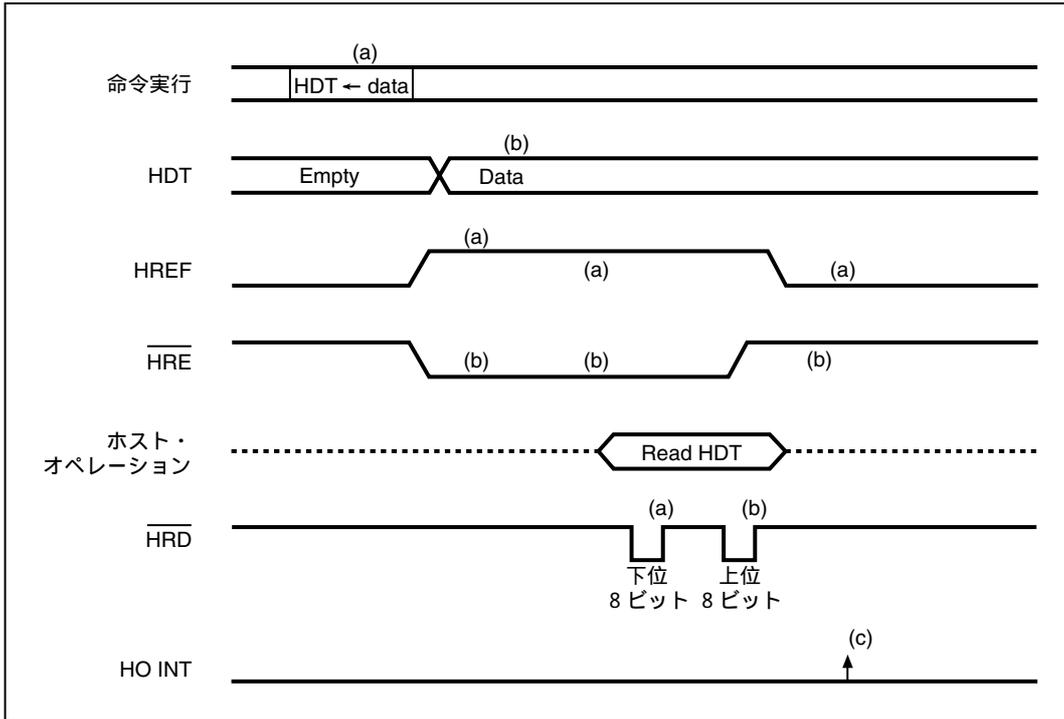
(1) HST を読み出し, ソフトウェア的に HREF = 1 を検出する (a)。

(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 (3.4.4 割り込みを参照)。

図3-54 ホスト・リード手順 (μ PD77111 ファミリ ホスト): HDT 書き込みウエイトなし



(b) ホスト・ライトの動作 (μ PD77111 ファミリ ホスト)

ホストからμ PD77111 ファミリ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 3 - 55 に示します)。

ホストはμ PD77111 ファミリの 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)。

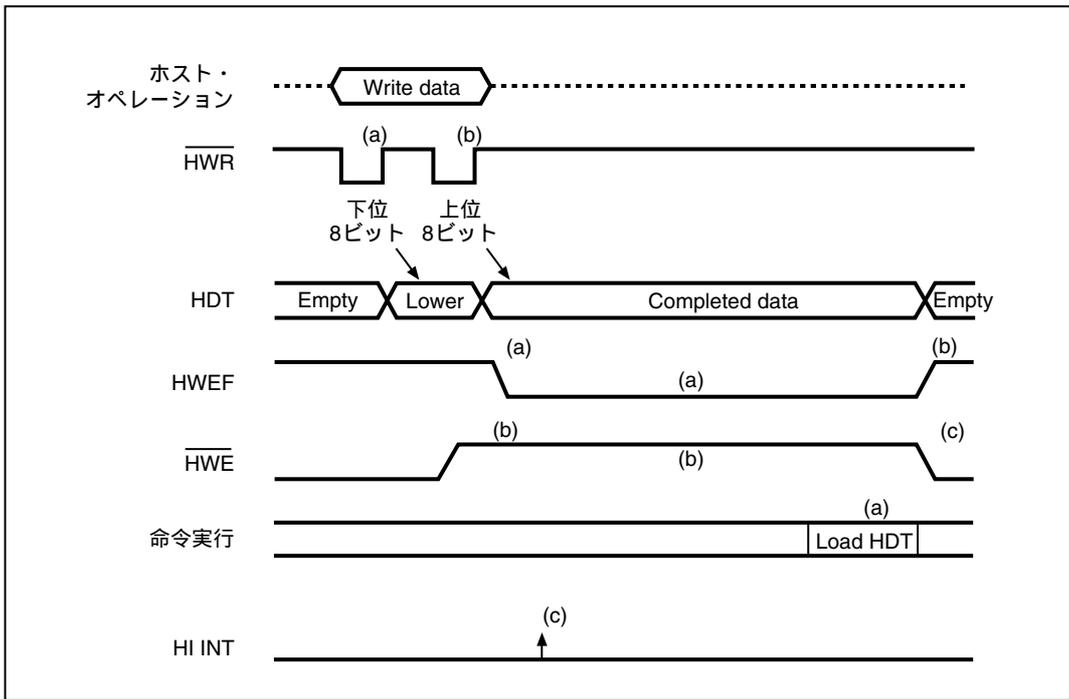
により HST の HWEF = 0 となってμ PD77111 ファミリのアプリケーション・プログラムに対し, HDT にデータが提示されたことを知らせる (a)。同時に $\overline{\text{HWE}}$ 端子はインアクティブ (ハイ・レベル) になり, HDT がビジーであることを外部に知らせる (b)。

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 (3.4.4 割り込み参照)。

μ PD77111 ファミリ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 3 - 55 ホスト・ライト手順 (μ PD77111 ファミリ ホスト): HDT 読み出しウエイトなし



(5) ハンドシェーク

μ PD77111 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 PD77111 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、ハンドシェー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ー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 */
ROL = 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、DSP 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 (3.4.4 割り込み参照)。

(c) 割り込み

μ PD77111 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 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、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、あるいは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) バーストで転送するときに、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 HDT に転送する直前の割り込み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。

3.7.5 汎用入出力ポート

μPD77111 ファミリは4端子の汎用入出力ポートを持っています。その概要を次に示します。

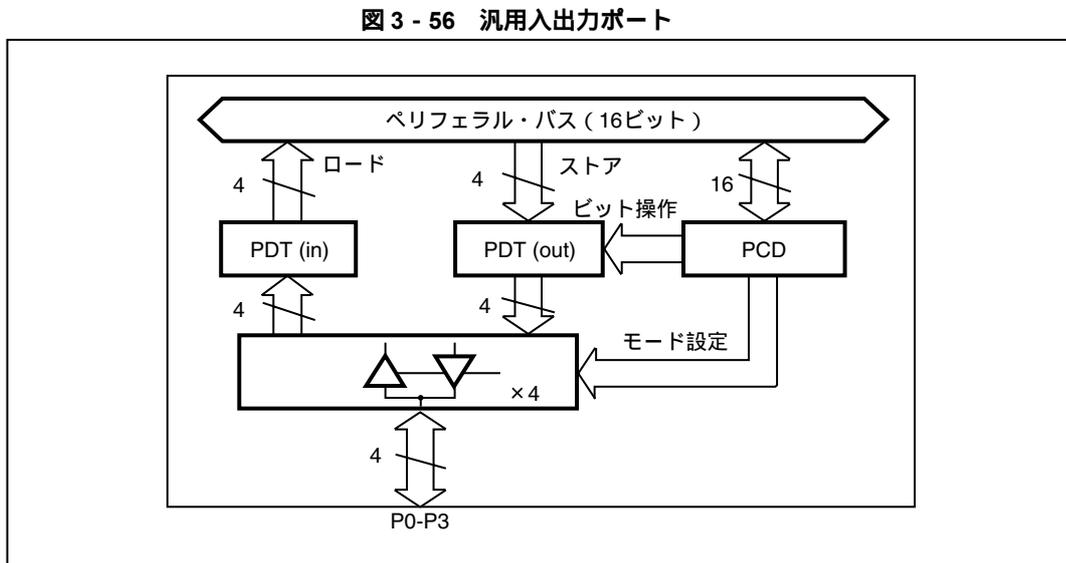
ハードウェア・リセットで入力モードにセットされ、リセット後のP0とP1への入力値がブート・モードを決定する。

アプリケーション・プログラムのもとでは、各端子ごとに任意に入出力を指定できる。

出力として指定された端子は、端子ごと独立に出力値を制御できる。

備考 μPD77115の汎用入出力端子は、P0-P7の8本（4ビットの汎用出力ポートが2セットあります）です。ここでは、P0-P3を例にあげて説明しますが、構成は同じです。μPD77115の機能については、6.4 汎用入出力ポートを参照ください。

図3 56に、汎用入出力ポートのブロック図を示します。



(1) 汎用ポートの使い方

汎用ポートを使うには次の3通りの方法があります。

(a) モード変更（入力 出力，出力 入力）

PCD レジスタにデータをストアすることによって、P0-P3 ピンを入力または出力モードに設定することができます。

PCD にストア …… 各ビットのモードを設定する（入力または出力）

(b) データ入力（P0-P3 μPD77111 ファミリ）

入力データは PDT (in) レジスタからロードされます。

PDT からロード …… 4ビットのデータ入力を入力する

(c) データ出力 (μ PD77111 ファミリ P0-P3)

ある値を出力ピンに出力するには、次の2通りの方法があります。

(i) PDT を使う方法

PDT (out) に出力データをストアします。

PDT にストア …… 4 ビットのデータを出力する

(ii) PCD を使う方法

1つの出力ピンに対してのみ、PCD レジスタで出力する値をビット操作します。

なお、モード設定とビット操作は同時に行うことができます。

PCD にストア …… 各ビットのモードを設定する (入力または出力)
 ・1 ビット出力データを操作する (ロウ・レベルまたはハイ・レベル)

(2) ポート・インタフェースの端子**(a) P0-P3 (汎用入出力ポート)**

汎用入出力端子です。次の機能があります。

出力端子は、CLKOUT の立ち上がりエッジに同期して出力が変化する。

入力端子は、CLKOUT の立ち上がりエッジに同期してサンプリングされる。

(3) ポート関連レジスタ**(a) ポート・データ・レジスタ (PDT-0x3804 : X / : Y)**

汎用入出力ポートを使ってデータをやりとりする 16 ビット・レジスタです。汎用入出力ポートからデータを入力するときは、PDT を読み出します。データを出力するときは、PDT に出力データを書き込むと、その値が P0-P3 へ設定されます。これらの端子は、PDT のビット 0-ビット 3 に対応しています。データはロード/ストア命令を使って、PDT と入出力できます。

PDT からのロード命令を実行すると、このレジスタのデータをペリフェラル・バスに出力します。Pn 端子を入力に指定した場合、入力端子 Pn にハイ・レベルを入力しているときにビット n が 1 に、入力端子 Pn にロウ・レベルを入力しているときにビット n が 0 になります。端子 Pn' が出力端子の場合、ビット n' の値は不定です。

PDT に対するストア命令を実行すると、このレジスタにペリフェラル・バスからデータを入力します。Pn' 端子を出力に指定した場合、ビット n' が 1 のときに出力端子 Pn' がハイ・レベルを出力し、ビット n' が 0 のときに、出力端子 Pn' がロウ・レベルを出力します。端子 Pn が入力端子の場合、ビット n の値は無効です。

(b) ポート・コマンド・レジスタ (PCD-0x3805 : X/Y)

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・レジスタです。

PCD の値は、ロード/ストア命令を使ってデータの入出力ができます。

PCD レジスタのすべてのビットが汎用レジスタにロードできるわけではありませんので注意してください (表 3 - 35 参照)。

リセットにより、PCD の値は 0 になります。

PCD の各ビットの機能を表 3 - 35 に示します。

表3-35 ポート・コマンド・レジスタ (PCD : 0x3805 : X/ : Y) (1/2)

ビット	名称	種類	ロード/ストア (L/S)	ビットの機能
15	BE	ビット操作	S	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
14	PSR	ビット操作	S	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
13	ME	モード設定	S	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
12	IO	モード設定	S	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
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	S	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
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

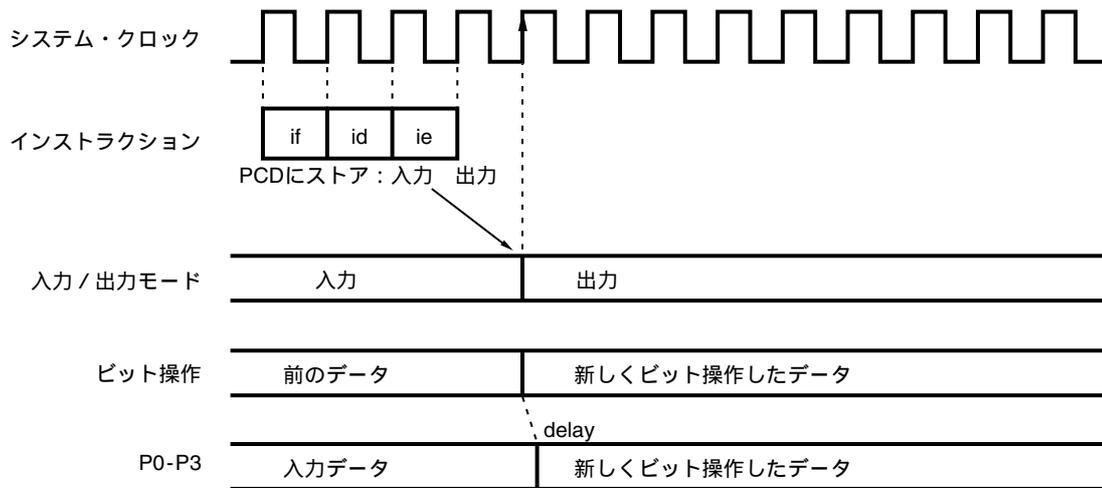
表3-35 ポート・コマンド・レジスタ (PCD : 0x3805 : X / Y) (2/2)

ビット	名称	種類	ロード/ストア (L/S)	ビットの機能
3 2 1 0	M3 M2 M1 M0	モード設定	S	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
		モード・ステータス	L	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

(4) ポート・インタフェースのタイミング

汎用入出力ポートは、本来は同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。

(a) 入力からの出力へのモード変更



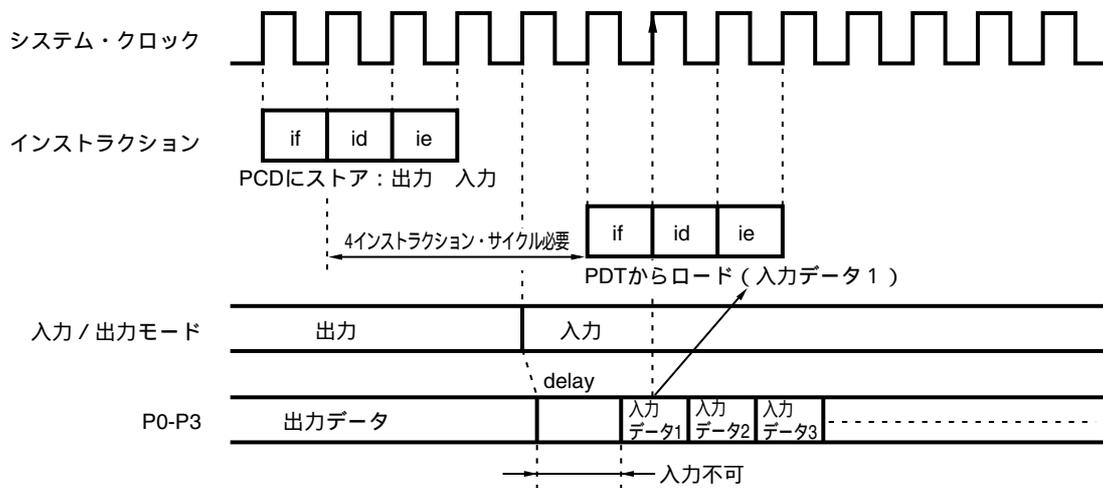
各ピンのモードは、PCD レジスタにストアする命令の実行サイクルから、2 システム・クロック後に入力から出力に変更されます。

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。

(b) 出力から入力へのモード変更



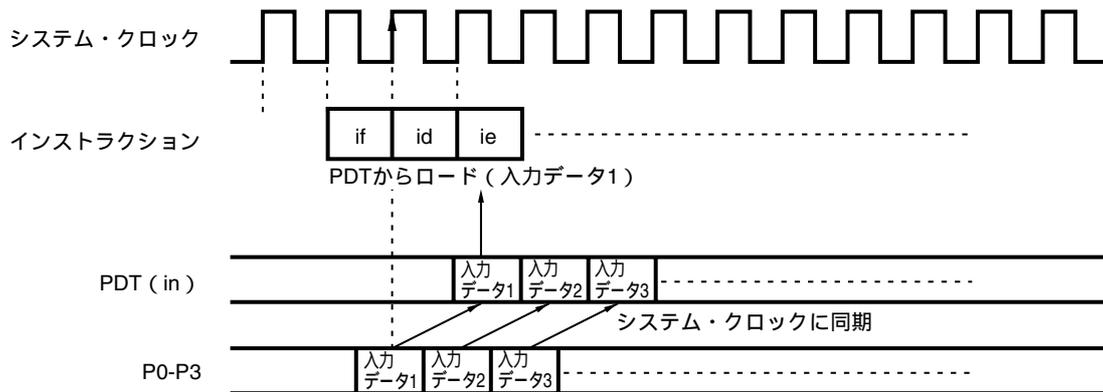
各ピンのモードは、PCD レジスタにストアする命令の実行サイクルから、2 システム・クロック後に、出力から入力に変更されます。しかしμ PD77111 ファミリは、入力に変更されてから、2 システム・クロックの間は入力を受け付けません。よって、PCD レジスタにストアする命令から PDT レジスタからロードする命令の間は、最低 4 システム・クロックの時間を空けてください。

プログラム例:

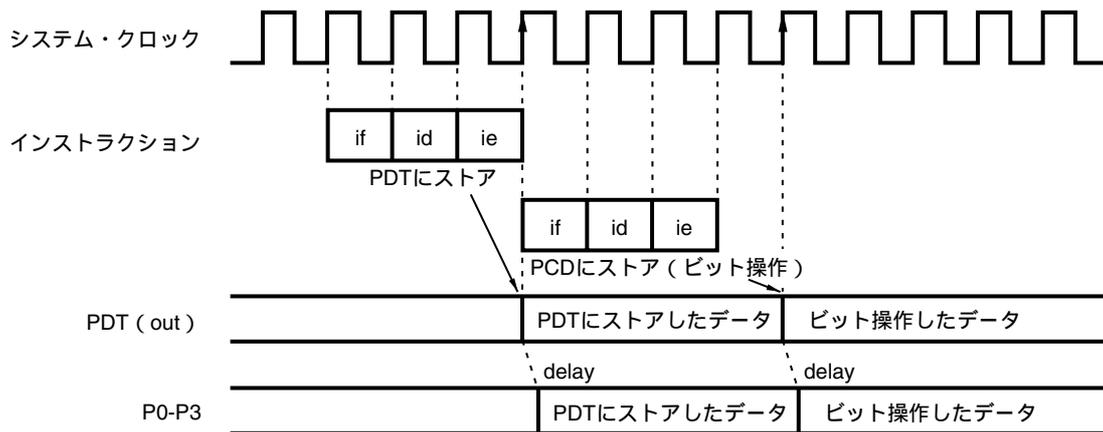
```
#define PCD = 0x3805
#define PDT = 0x3804
R0L = 0x200f ;
* PCD : x = R0L ; P0-P3 出力 入力
この間は最低 4 システム・クロック空けてください
R1L = * PDT : x ; PDT からロード
```

(c) 入力ポートのタイミング

ピン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。



(d) 出力ポートのタイミング



(PDT レジスタにストアする命令の場合)

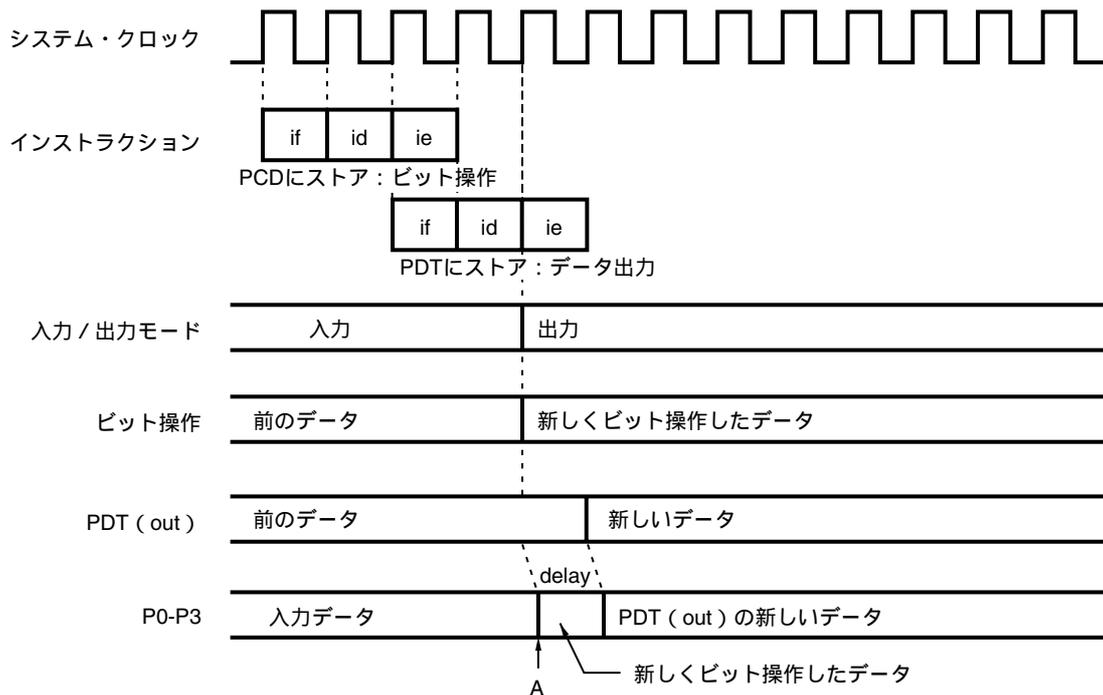
出力されるデータは、PDT レジスタにストアする命令の実行サイクルから、1 システム・クロック後に出力されます。

(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せん。

(e) 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) 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

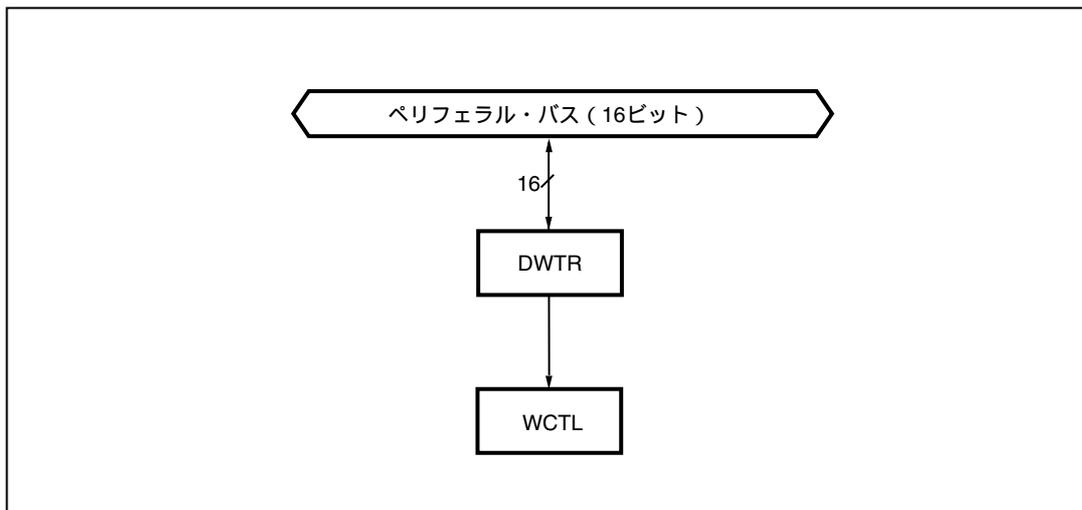
R0L = All_In_mode           ; P3-P0 入力端子
*PCD : x = R0L              ;
R0L = P0_Out_mode + Out_P0_Low ; P0 出力端子 (ロウ・レベル)
*PCD : x = R0L              ;
R0L = P1_Out_mode + Out_P1_High ; P1 出力端子 (ハイ・レベル)
*PCD : x = R0L              ;
```

3.7.6 ウェイト・コントローラ

μ PD77110, 77112, 77114 は, 外部メモリ領域をアクセスするとき, 挿入するウェイト数を 0, 1, 3 および 7 ウェイトのいずれかに, あらかじめレジスタで指定することができます。

ウェイト・コントローラのブロック図を図 3 - 57 に示します。

図 3 - 57 ウェイト・コントローラ



(1) データ・メモリ・ウェイト・サイクル・レジスタ (DWTR-0x3808 : X/ : Y)

3.5.2 (3)(d) ウェイト・コントローラを参照してください。

3.7.7 ディバグ・インタフェース (JTAG)

μPD77111 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)

(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。

μPD77111 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。

[ディバグ用端子 (TAP : test access port)]

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。

・ TCK (入力)

テスト・クロック入力。

未使用時は 0 を入力します (勧告準拠)。

注意 TCK をハイ・レベルで停止しないでください。

・ TMS (入力): テスト・モード・セレクト入力。

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。

・ TDI (入力): テスト・データ入力。

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。

・ TDO (出力): テスト・データ出力。

TCK の立ち下がりエッジに同期して出力が変化します。

・ TICE (出力): インサーキット・エミュレーションのブレーク・モードを構築するための出力です。

・ $\overline{\text{TRST}}$ (入力): テスト・リセット入力。

通常動作時は 0 (アクティブ・ロウ) に、ディバグが使用時、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。

(2) 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 PD77111 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。

(a) テスト・インストラクション・レジスタ (Test Instruction Register)

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3-36に示します。

表3-36 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

(b) テスト・バイパス・レジスタ (Test Bypass Register)

TDI から入力したデータを、TDO に出力するレジスタです。

(3) ディバグ機能 (インサーキット・エミュレータ機能)

μ PD77111 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. ディバグ機能は、 μ PD7701xファミリ、 μ PD77111ファミリ共通のインサーキット・エミュレータが使用します。インサーキット・エミュレータを使用する場合のJTAG端子処理を図3-59に示します。ユーザ・システムでインサーキット・エミュレータ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3-58 JTAG 端子外観図

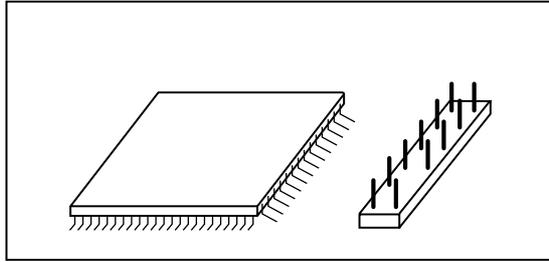
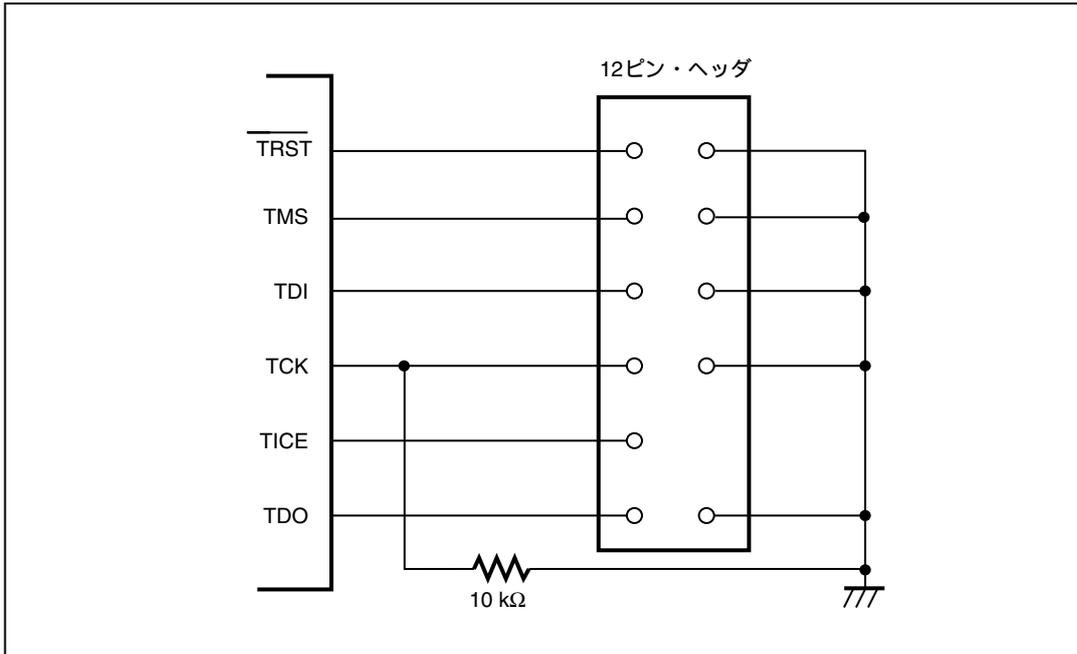


図3-59 JTAG 端子処理



第4章 ブート機能

この章では、 μ PD77111 ファミリのブート機能を説明します。最初に概要を説明し、次にブート機能の種類（ブート・モード）、リセット時のブート、ブート・サブルーチン、ブート所要時間の順に説明します。

注意 特に指定のないかぎり、この章では、 μ PD77115 を除いた μ PD77111 ファミリーについて説明しています。
 μ PD77115 をご使用になる場合は、第7章 μ PD77115 のブート機能をお読みください。

4.1 概 要

μ PD77111 ファミリーは従来の μ PD7701x ファミリー製品と同等のブート機能をサポートしています。ただし、 μ PD77110 はブート機能の一部に制限があります。各項目を参照してください。 μ PD77115 のブート機能については、第7章 μ PD77115 のブート機能を参照してください。

内部命令 RAM にブート・アップするためのプログラムを、命令メモリ空間の 0-0xFF 番地に ROM として内蔵しています。

ハードウェア・リセットで PC（プログラム・カウンタ）が“0”になり、リセット後、ただちにブートを実行するリセット・ブートのほか、サブルーチン・サービスとして、いくつかのエントリ・ポイントが公開されています。したがって、それらを利用すれば、アプリケーション・プログラムから命令メモリ空間中の任意の RAM 部分を書き換えることが可能となります（リブート機能）。

この章では、ブート機能を整理するためのモードの分類（リセット・ブートとリブート、セルフ・ブートとホスト・ブート、ワード・ブートとバイト・ブート）、各モードにおけるブートの機能、ブートのパラメータ、呼び出し方法などについて解説します。

ブート実行にあたって、次のレジスタが関わってきます。

[ブート実行後、内容が変化するレジスタ]

R7

DP3

DP7

HDT（ホスト・データ・レジスタ）

[ブート実行の前に、設定されるレジスタ]

IWTR（命令メモリ・ウェイト・サイクル・レジスタ）

DWTR（データ・メモリ・ウェイト・サイクル・レジスタ）

HST（ホスト・インタフェース・ステータス・レジスタ）

注意 これらのレジスタが常にすべて使用されるわけではありません。詳しくは各ブート・モードの説明を参照してください。

4.2 ブート・モード

4.2.1 ブート・モードの分類

ブート・モードを次の3つの視点から分類します。

ブート起動形式による分類

ブート・ソースによる分類

転送ワード・サイズによる分類

(1) ブート起動形式による分類

ブートは起動形式から見て、次の2つの形式に分類できます。

- (a) リセット・ブート
- (b) リブート

(a) リセット・ブート

ハードウェア・リセット時、PC (プログラム・カウンタ) は“0”にクリアされますが、命令メモリの0番地は内部ブートROM領域のリセット・ブート・エン트리・ポイントであり、 μ PD77111ファミリは自動的にリセット・ブートを実行します。このときブート対象となる命令メモリ領域は、内部命令RAM (0x200番地から開始) に限定されています。

(b) リブート

ブートROM領域には、ブート・サービスのためのサブルーチン・エン트리・ポイントが設定されています。ユーザはアプリケーション・プログラムからこれらリブート・サブルーチン呼び出すことによって、任意の時点で任意部分の命令RAMを書き換えることができます。

(2) ブート・ソースによる分類

ブートとは何らかの形式により命令を書き換えることですが、書き換えるべき命令コード・データ、および書き換えのためのパラメータを、どこから獲得するかにより次の2つの形式があります。

- (a) セルフ・ブート
- (b) ホスト・ブート

(a) セルフ・ブート

セルフ・ブートでは、データ・メモリに用意されたプログラム・コード・データを命令メモリに転送します。ブート・パラメータは、リセット・ブートではデータ・メモリに (Yメモリ: 0x4000番地から)、リブートではレジスタに設定します。

ROMバージョンの製品 (μ PD77111, 77112, 77113A, 77114) では、ROM領域にブート・パラメータとプログラム・データを固定し、リセット・ブートをセルフ・ブートとすることで、スタンドアロン・システムを構築できます。ブート・パラメータは、常に内部YROMに格納されます。プログラム・データは内部または外部のXYROMのどちらにも格納することができます。

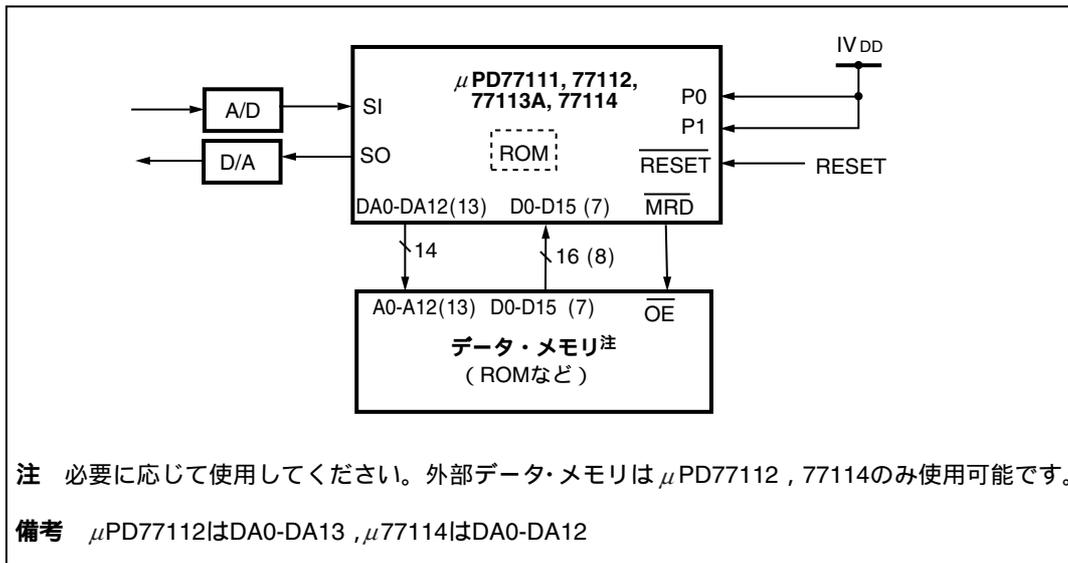
μ PD77110 では、次のような点が異なります。

セルフ・ブートを行う場合、ブート・プログラムは 0x4000 : Y のブート・パラメータを読み込みますが、μ PD77110 の 0x4000 : Y は RAM のため、電源投入時の値は不定になります。このため、μ PD77110 では、電源投入時のブート・モードとして、セルフ・ブートを選択することはできません。ホスト・ブートを行う必要があります。

ただし、μ PD77110 の場合でも、電源投入時以外のリセットによるブートとして、あらかじめ 0x4000 : Y 以降にブート・パラメータを記述しておくことによってセルフ・ブートを行うことができます。この場合のリセットは、通常のリセットを行ってください（これは、μ PD77110 を使って、μ PD77111 , 77112 , 77113A , 77114 のブートアップで動作のエミュレーションをする場合に行います）。

図 4 - 1 に、セルフ・ブートのシステム構成例を示します。

図 4 - 1 セルフ・ブートのシステム構成例



(b) ホスト・ブート

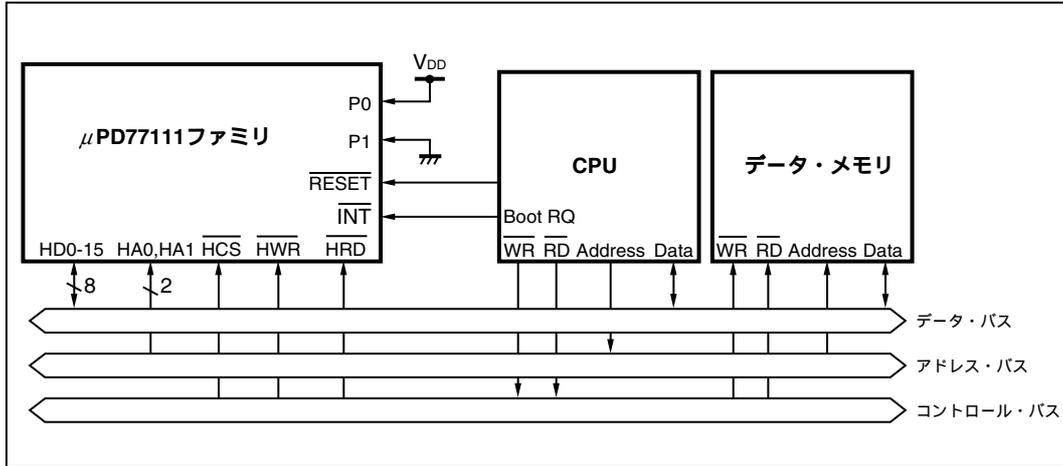
ホスト・ブートでμ PD77111 ファミリは、プログラム・コード・データをホスト・インタフェースから読み込んで命令メモリに転送します。ブート・パラメータは、リセット・ブートではプログラム・コード・データに先立ってホスト・インタフェースから読み込みます。リブートの場合は、ブート・パラメータは、アプリケーションプログラムによって、レジスタに設定されなければなりません。

ホスト・ブートは、ホスト・インタフェースに別の CPU を接続し、その CPU からμ PD77111 ファミリにプログラムをダウン・ロードするようなシステム構成で利用されます。

特にμ PD77110 の場合はホスト・ブートが標準のブート方法となります。

図 4 - 2 にホスト・ブートのシステム構成例を示します。

図4-2 ホスト・ブートのシステム構成例



(3) 転送ワード・サイズによる分類

この分類は、セルフ・ブートでのみ意味があります。セルフ・ブートのパラメータで、データ・メモリ側読み出しサイズを次のように指示できます。

16ビット・ワード / 1アドレス (ワード・ブート) : 図4-3 参照

8ビット・バイト / 1アドレス (バイト・ブート) : 図4-4 参照

したがってワード・ブートの場合、2データ・メモリ・アドレスが1命令ステップに対応し、バイト・ブートでは、4データ・メモリ・アドレスが1命令ステップに対応することになります。

リセット時のセルフ・ブートにおいては通常、ブート・パラメータとプログラム・コード・データをROMとして固定します。

図4-3 ワード・ブートのイメージ

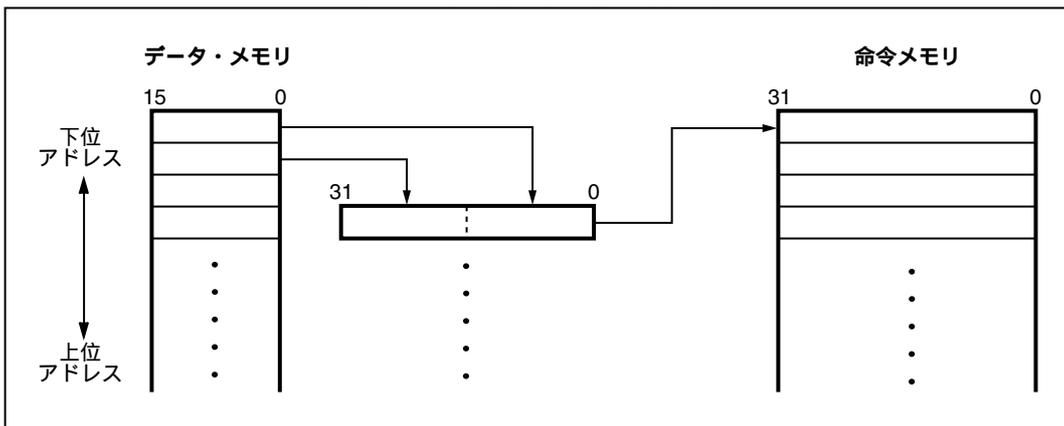
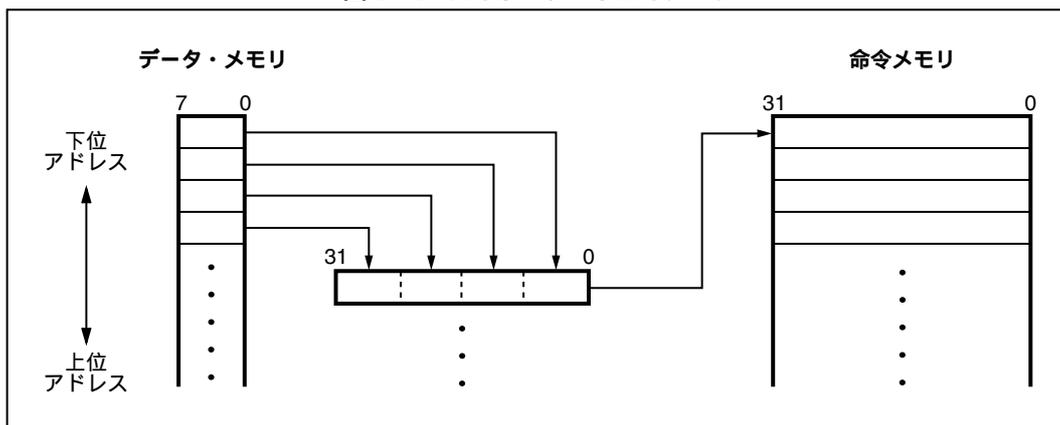


図4-4 バイト・ブートのイメージ



4.3 リセット時のブート

μ PD77111 ファミリは、ハードウェア・リセット解除後 0 番地に配置されたブート・プログラムを実行します。ブート・プログラムは、最初に汎用入出力ポート P0, P1 を読み込み、そのビット・パターンによってブート・モード（セルフ・ブート、ホスト・ブート）を決定します。リセット時に P0, P1 に提示されるパターンとブート・モードの関係を表 4-1 に示します。μ PD77113A, 77114 の場合、ホスト・ブートあるいはセルフ・ブートが指定されたときに、ブート処理と同時に内部データ RAM のセルフ・チェックを行います。

表 4-1 P0, P1 のパターンとブート・モード

P1	P0	ブート・モード
0	0	ブートを実行せず、0x200 番地に分岐する ^注
0	1	ホスト・ブートを実行後、0x200 番地に分岐する
1	1	セルフ・ブートを実行後、0x200 番地に分岐する
1	0	設定禁止

注 この設定は、パワーオン時のリセット・ブートや、その後パワーダウン・モードからの復帰などのために、DSP がリセットを必要とするときに使用します。

リセット時のブートでは、命令メモリへのローディング・アドレスを示すパラメータはありません。命令メモリへは次のようなローディング・アドレスが決められています。

内部 RAM 領域の先頭（0x200 番地）からロードされる。

注意 μ PD77110 の場合、リセット・ブートの対象は 0x0200-0x0FFF の命令 RAM だけです。0x4000-0xBFFF の命令 RAM へのブートアップにはホスト・リブートを利用します。また、電源投入時のブート・モードとしてセルフ・ブートを選択することはできないので、ホスト・ブートを行う必要があります。PLL の初期化を行ったときも同様です。あらかじめ、Y メモリの 0x4000 番地以降にブート・パラメータを記述しておくことによって、その後のリセット入力によるセルフ・ブートを行うことは可能です（PLL の初期化を行うリセットを除きます）。ただし、この場合のブートの対象は 0x0200-0x0FFF の命令 RAM のみとなります。

4.3.1 セルフ・ブートの場合

(1) セルフ・ブートのパラメータ

最初に次のパラメータが、Yメモリの0x4000番地から読み込まれます。

メモリ空間指示

ワード・ブート/バイト・ブート指示

表4-2にパラメータ内容を示します。

表4-2 セルフ・ブートのパラメータ (0x4000 : Y)

ビット	値	意味
0	0	Yメモリ・ブート プログラム・コードをYメモリ空間から読み込む。
	1	Xメモリ・ブート プログラム・コードをXメモリ空間から読み込む ^{注1} 。
1	0	ワード・ブート データ・メモリを16ビットで読み込む。 したがって、命令メモリ1アドレスがデータ・メモリ2アドレス分に該当する ^{注2} 。
	1	バイト・ブート データ・メモリを8ビットで読み込む。 したがって、命令メモリ1アドレスがデータ・メモリ4アドレス分に該当する ^{注3} 。
2-7	任意	バイト・ブート時
2-15	任意	ワード・ブート時

注1. Xメモリ・ブートでも、ブート・パラメータはすべてYメモリ空間から読み込みます。このときデータ・メモリ・ウエイトは、7ウエイトが設定されています。パラメータ・アドレスは次のとおりです。

0x4000 : Y-0x4004 : Y (ワード・ブート時)

0x4000 : Y-0x4009 : Y (バイト・ブート時)

2. 図4-3を参照してください。
3. 図4-4を参照してください。

注意 DWTR, IWTR, R7, DP3 および DP7 レジスタは、ブート・ルーチンによって変更されます。

(a) ワード・ブートのパラメータ

ワード・ブート時のパラメータ・メモリ・マップを表4-3に示します。

表4-3 ワード・ブートのパラメータ・メモリ・マップ

アドレス	メモリの値
0x4000 : Y	16/8 ビット, X/Y
0x4001 : Y	DWTR に設定する値
0x4002 : Y	ダミー・データ
0x4003 : Y	読み込むプログラムが置かれているデータ・メモリの先頭アドレス
0x4004 : Y	プログラムのステップ数 ^注

注 プログラムのステップ数とは、32 ビット=1ワードとしたときのワード数であり、外部データ・メモリ中に占めるワード数ではありません。

(b) バイト・ブートのパラメータ

バイト・ブート時のパラメータ・メモリ・マップを表4-4に示します。

表4-4 バイト・ブートのパラメータ・メモリ・マップ

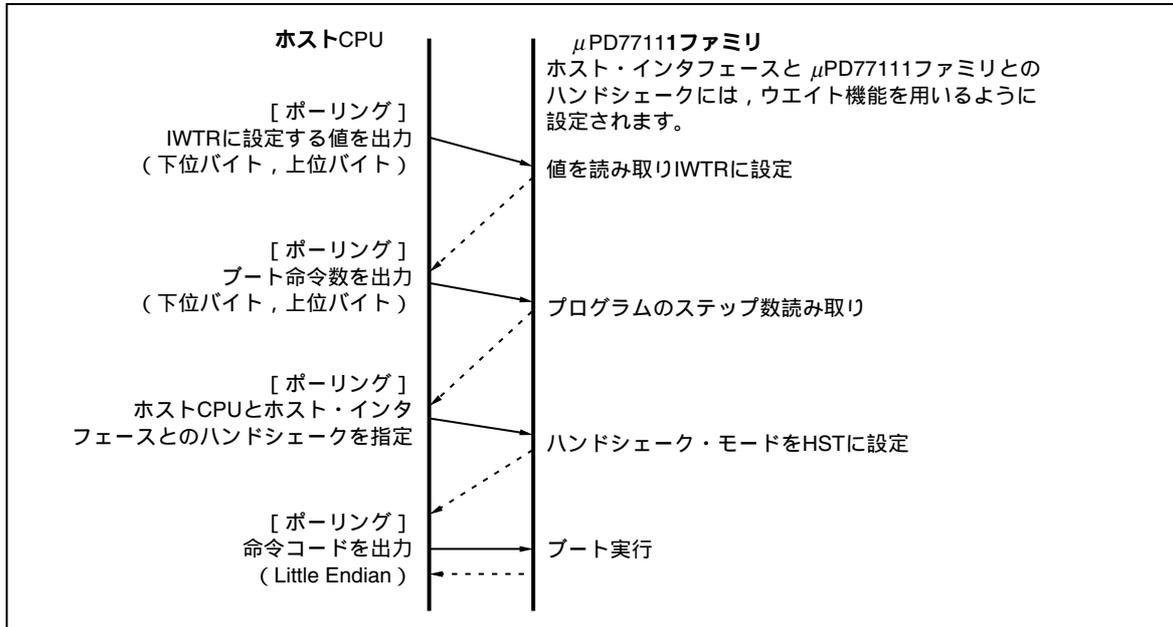
アドレス	メモリの値
0x4000 : Y	16/8 ビット, X/Y
0x4001 : Y	-
0x4002 : Y	DWTR に設定する値 (下位バイト)
0x4003 : Y	DWTR に設定する値 (上位バイト)
0x4004 : Y	ダミー・データ
0x4005 : Y	ダミー・データ
0x4006 : Y	読み込むプログラムが置かれているデータ・メモリの先頭アドレス (下位バイト)
0x4007 : Y	読み込むプログラムが置かれているデータ・メモリの先頭アドレス (上位バイト)
0x4008 : Y	プログラムのステップ数 ^注 (下位バイト)
0x4009 : Y	プログラムのステップ数 ^注 (上位バイト)

注 プログラムのステップ数とは、32 ビット=1ワードとしたときのワード数であり、外部データ・メモリ中に占めるワード数ではありません。

4.3.2 ホスト・ブートの場合

ホスト・ブートは、ブート・パラメータと命令コードをホスト・インタフェースから獲得します。図4-5にその手順イメージを示します。

図4-5 ホスト・ブートの手順イメージ



注意 μPD77110の場合, 電源投入時のブートはホスト・ブートを用いることになります。また, ブートアップの対象となる命令領域は 0x0200-0x0FFF の命令 RAM だけとなります。0x4000-0xBFFF への命令 RAM へのブートアップにはホスト・リブートを使用します。

(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 と 16 ビット・モード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 : ウェイトを使用する。

HREM =0 : HRE をマスクしない。

HWEM =0 : HWE をマスクしない。

16 ビット・モード : ホストは, 指定のパラメータを下位 8 ビット, 上位 8 ビットの順に HDT に書き込みます (これは HST への設定値ではなく, ホスト・ブートの規約です)。

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。

ダミー・データ	: 0x0000 を転送します。
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。HAWC (ビット 10) を除く、そのほかのビットを設定します。HAWC 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、ホスト・インタフェースが 8 ビット幅であることを考慮すると、ビット 7-0, 15-8, 23-16, 31-24 の部分バイト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。

1 番目の転送	: ダミー・データの下位 8 ビット
2 番目の転送	: ダミー・データの上位 8 ビット
ポーリング	: μ PD77111 ファミリが HDT (in) からデータをロードする間ウエイト
3 番目の転送	: ブート命令数の下位 8 ビット
4 番目の転送	: ブート命令数の上位 8 ビット
ポーリング	: μ PD77111 ファミリが HDT (in) からデータをロードする間ウエイト
5 番目の転送	: HST 設定値の下位 8 ビット
6 番目の転送	: HST 設定値の上位 8 ビット
ポーリング	: μ PD77111 ファミリが HDT (in) からデータをロードする間ウエイト
7 番目の転送	: 1 番目の命令コード (ビット 7-0)
8 番目の転送	: 1 番目の命令コード (ビット 15-8)
ポーリング	: μ PD77111 ファミリが HDT (in) からデータをロードする間ウエイト
9 番目の転送	: 1 番目の命令コード (ビット 23-16)
10 番目の転送	: 1 番目の命令コード (ビット 31-24)
ポーリング	: μ PD77111 ファミリが HDT (in) からデータをロードする間ウエイト
11 番目の転送	: 2 番目の命令コード (ビット 7-0)
:	:
:	:
(4n+6) 番目の転送	: n 番目の命令コード (ビット 31-24) ^注
ポーリング	: μ PD77111 ファミリが HDT (in) からデータをロードする間ウエイト

注 転送する命令コード数を n とすれば、総転送バイト数は 4n+6 となります。

4.4 ブート・サブルーチン（リポート）

通常、アプリケーション・プログラムは命令メモリを書き換えることはできませんが、ブート ROM のなかで公開されているブート・サブルーチン（リポート）を利用すれば命令メモリに新たな命令を設定できます。特に μ PD77110 の場合、0x4000-0xBFFF の領域へのブートアップはリポートを用いることになります。リポートのエントリ・ポイントを、表 4-5 に示します。リポートを実行する場合は、所定のパラメータをレジスタに設定したあと、表 4-5 に示されたアドレスをコール（CALL 命令による）します。

リポートに関連しないレジスタおよび端子は、リポート中およびリポート後は、ブート・サブルーチンをコールしたときの状態を保持します。リポートによって初期化されることはありません。

表 4-5 ブート・サブルーチンのエントリ・ポイント

リポート・モード			エントリ・アドレス
セルフ・ブート	Xメモリ	ワード・リポート	0x2
		バイト・リポート	0x4
	Yメモリ	ワード・リポート	0x1
		バイト・リポート	0x3
ホスト・ブート	ホスト・リポート		0x5 (μ PD77110 以外)
			0x6 (μ PD77110)

注意 1. リポートを実行するときは、次のことに注意してください。

使用するレジスタの値は保存されません。

プログラム・スタックを 1 レベル消費します（エントリ時）。

ループ・スタックを 1 レベル消費します。

リポートは、すべての割り込みを禁止した状態で実行してください（リポート中に割り込みを受け付けた場合の正常な動作は保証できません）。

リポートを終了したあと、サブルーチンを呼び出した命令（CALL 命令）の次のアドレスにリターンします。

2. DWTR, R7, DP3, DP7, HST および HDT は、ブート・ルーチンによって書き換えられます。

4.4.1 Xメモリ・ワード・リポート、Xメモリ・バイト・リポートのパラメータ

Xメモリ上に命令コードを配置するXリポートには、1データ・メモリ・アドレスに16ビット・ワードを置くXメモリ・ワード・リポートと、8ビット・バイトを置くXメモリ・バイト・リポートがあります。どちらの場合でも、次に示すパラメータを所定のレジスタに設定したうえで、表4-5に示すそれぞれのエントリ・ポイントをコールします。

R7L：リポートする命令ステップ数

DP3：命令コードの格納されているXメモリの先頭アドレス

DP7：ロードする命令メモリの先頭アドレス

注意 1. パラメータ・レジスタの値は保存されません。

2. 必要に応じてDWTRを設定してください。

4.4.2 Yメモリ・ワード・リポート, Yメモリ・バイト・リポートのパラメータ

Yメモリ上に命令コードを配置するYリポートには, 1データ・メモリ・アドレスに16ビット・ワードを置くYメモリ・ワード・リポートと, 8ビット・バイトを置くYメモリ・バイト・リポートがあります。どちらの場合でも, 次に示すパラメータを所定のレジスタに設定したうえで, 表4-5に示すそれぞれのエントリ・ポイントをコールします。

R7L: リポートする命令ステップ数

DP3: ロードする命令メモリの先頭アドレス

DP7: 命令コードの格納されているYメモリの先頭アドレス

注意 1. パラメータ・レジスタの値は保存されません。

2. 必要に応じてDWTRを設定してください。

4.4.3 ホスト・リポートのパラメータ

ホスト・インタフェースからリポートする場合は, 次に示すパラメータを所定のレジスタに設定したうえで, 表4-5に示すエントリ・ポイントをコールします。

R7L: リポートする命令ステップ数

DP3: ロードする命令メモリの先頭アドレス

注意 1. パラメータ・レジスタの値は保存されません。

2. 必要に応じてHST, IWTRを設定してください。ただし, リポート・ルーチンを読み出す前に必ずHSTのHAWEは“1”(ウエイトを使用)に設定してください。

3. リポート開始時点で, HDTは空(読み出す前のデータが残っていない状態)でなければなりません。

4. μ PD77110では上記レジスタ以外にDP6も書き換えられます。

★ 4.5 シグネチャ演算

μ PD77110, 77113A, 77114は, 大容量の命令RAMを内蔵するため, 命令RAMデータの内容をベリファイする機能がサポートされています。シグネチャ演算処理は, ブートアップされた命令RAMのデータをチェックするため, IEEE802.3で勧告している次に示す生成多項式の演算処理を行い, 結果をレジスタに返すものです。シグネチャ演算処理はブート・プログラム中にサブルーチンとして用意されています。エントリ・アドレス0x9のサブルーチン・コールで実行され, 結果をR7に返します。

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{21} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^6 + x^4 + x^2 + x^1 + 1$$

シグネチャ演算処理をサブルーチン・コールすると, 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)使用します。必要なレジスタは, 退避/復帰を行う必要があります。

シグネチャ演算処理に使用するレジスタは, R1-R7, DP3, DP6です。

注意 シグネチャ演算処理は、すべての割り込みを禁止した状態で行います。演算処理中に割り込みを受け付けた場合には DSP の正常な動作は保証できません。パラメータの設定は次のとおりです。

- ・ R7L : 演算する命令ワード数 (必須)
- ・ DP3 : 演算する命令メモリの先頭アドレス (必須)

演算対象として内部命令 RAM (0x0200-0x0FFF および 0x4000-0xBFFF) 以外を選択しないでください。
最大ワード数は、0x8000 です。

★ 4.6 命令 ROM 修正処理

μ PD77113A, 77114 は、マスク ROM 化した命令に誤りがある場合、最大 4 アドレス分の命令に対して修正を行うことができます。

エントリ・アドレスは 0x10D で、次のパラメータを設定してからこのアドレスをコールすることによって、修正処理を実行します。

R7L : 修正処理をする命令 ROM のアドレス
R6H, R6L : 命令コード (32 ビット)

注意 1. 命令 ROM 修正処理のエントリをコールする場合は次のことに注意してください。

- ・使用するレジスタの値は保存されません。
- ・プログラム・スタックを 1 レベル消費します (エントリ時)。
- ・すべての割り込みを禁止した状態でエントリ・コールしてください。

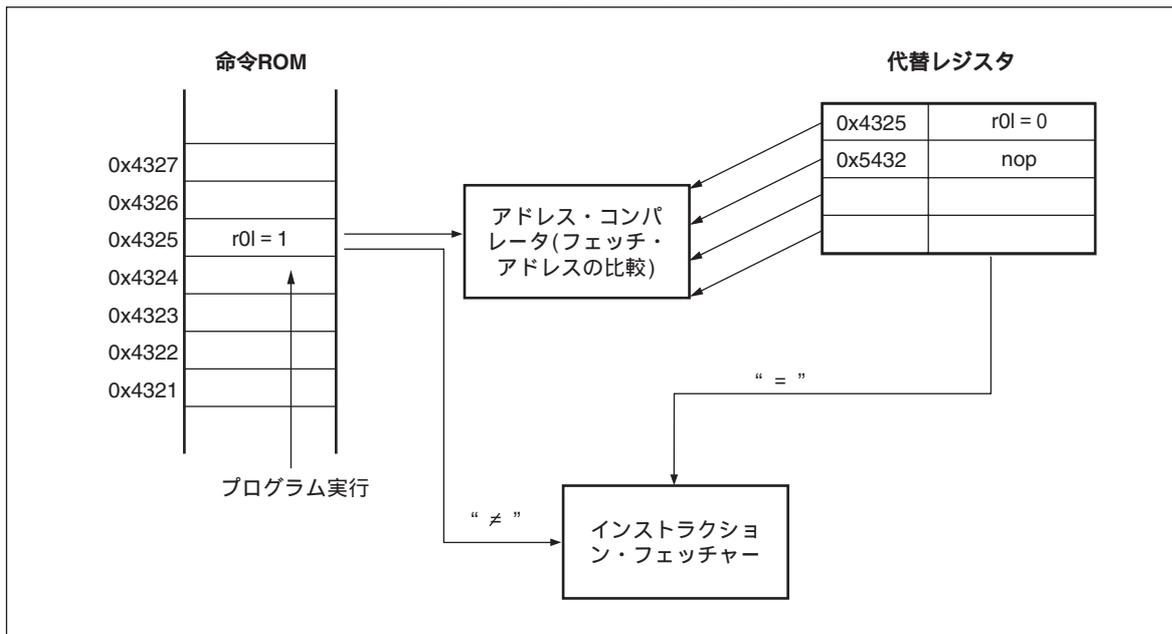
2. R4, R5, R6, R7, DP3 および DP7 は、修正処理ルーチン中で書き換わります。

4.6.1 命令 ROM 修正処理の概要

μ PD77113A, 77114 は、命令フェッチ・アドレスのアドレス・コンパレータと 4 セットの命令メモリ用代替レジスタを持っています。

命令のフェッチ時にフェッチ・アドレスと代替レジスタに設定された修正対象の命令アドレスとをコンパレータで比較し、一致した場合に代替レジスタに設定された命令をフェッチして実行します。

図4-6 命令ROM修正処理の概要



4.6.2 命令 ROM 修正処理の使用方法

(1) イニシャル・リセット・ブートを利用する方法

原則としてリセット時のブート・モードをホスト・ブートあるいは外部データ領域からのセルフ・ブートとして使用します。

内部命令 RAM はリセット時のブート対象領域です。このため、ホスト側あるいは外部データ領域に接続される不揮発性メモリ（フラッシュ・メモリなど）に、修正を行うプログラムをブート・コードとして格納することにより本機能を実現することができます。

図 4 - 7 命令 RAM にブートするプログラムのソース・イメージ例

修正前	修正後
Reset_vector:	Reset_vector:
jmp Main;	jmp Main;
:	:
Interrupt_vector:	Interrupt_vector:
nop;	nop;
:	:
Main:	Main:
Instruction #1;	call Rom_correction;
Instruction #2;	Instruction #1;
:	Instruction #2;
	:
	Rom_correction:
	r7l = (修正対象の命令 ROM アドレス#1);
	r6h = (修正後の命令コード#1 の上位 16 ビット);
	r6l = (修正後の命令コード#1 の下位 16 ビット);
	call 0x10D;
	r7l = (修正対象の命令 ROM アドレス#2);
	r6h = (修正後の命令コード#2 の上位 16 ビット);
	r6l = (修正後の命令コード#2 の下位 16 ビット);
	call 0x10D;
	:
	ret;
	:

(2) あらかじめ修正対応ルーチンを作成する方法

命令 ROM あるいは命令 RAM にあらかじめ修正対応のルーチンを組み込み、外部インタフェースからのデータ入力により本機能を実現します。

図4-8 修正対応ルーチンのソース・イメージ例

```

Reset_vector:
    jmp Main;
    :

Interrupt_vector:
    nop;
    :

Main:
    if ( ROM 修正あり ) call Rom_correction;注1
    Instruction #1;
    Instruction #2;
    :

Rom_correction:
    r0l = ( 修正の個数 );注2

    loop r0l{
        r7l = ( 修正対象の命令 ROM アドレス );注2
        r6h = ( 修正後の命令コードの上位 16 ビット );注2
        r6l = ( 修正後の命令コードの下位 16 ビット );注2
        call 0x10D;
    }
    ret;
    :

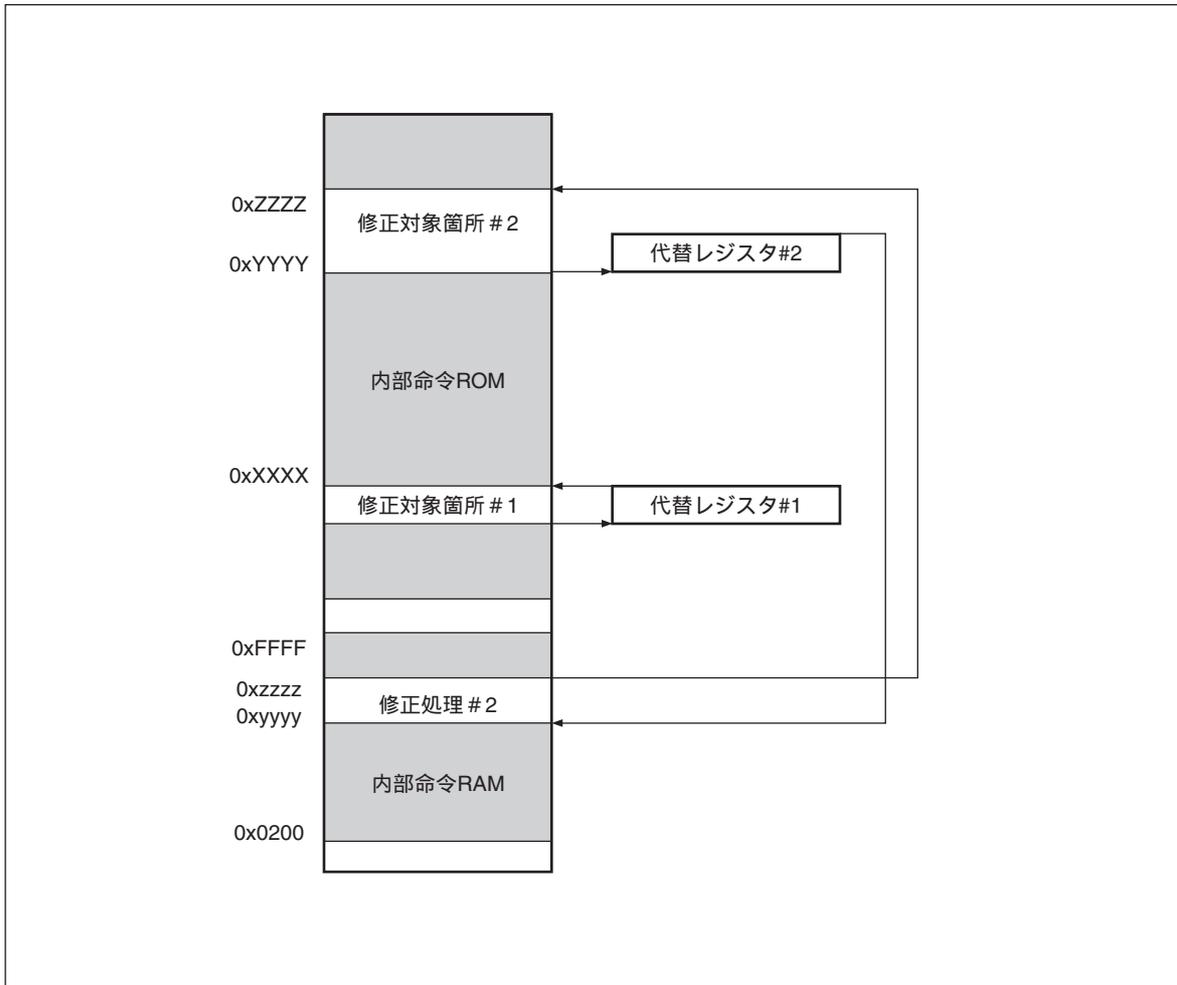
```

- 注 1. ポート端子の入力値、ホストあるいはシリアル・インタフェースからのコマンド入力により、命令 ROM 修正を行うか判断をします。各インタフェースの設定をあらかじめ行う必要があります。また、コマンド入力のシーケンスはユーザにおいて考慮する必要があります。
2. メモリ、ホストあるいはシリアル・インタフェースからのデータ入力により、命令メモリの修正個数、アドレス、命令コードを取得します。各インタフェースの設定をあらかじめ行う必要があります。また、データ入力のシーケンスはユーザにおいて考慮する必要があります。

4.6.3 プログラム実行フロー

図4-9に命令ROM修正処理機能を実施したプログラムの実行例を示します。

図4-9 プログラム実行例



#1 1命令の修正

コンパレータでアドレス比較し、修正対象箇所に一致したら、そのアドレスに割り当てられた代替レジスタに格納されている命令コードをフェッチし実行する。

#2 連続した複数命令の修正例（あるいは1命令を複数命令で修正する修正例）

コンパレータでアドレス比較し、修正対象箇所の先頭に一致したら、そのアドレスに割り当てられた代替レジスタに格納されている命令コードをフェッチし実行する。代替レジスタには内部命令RAMへの分岐命令を記述する。

あらかじめ内部命令RAMに修正対象箇所#2の修正モジュールを記述しておく（ホスト・ブート、外部データ領域からのセルフ・ブートで対応）。置き換えた分岐命令からこの領域に分岐し、実行する。

修正モジュールの最後に正常なフローとなる位置（修正対象箇所#2の最後の命令の次の実行アドレス）への分岐命令を記述しておくことによって元のフローに復帰する。

4.6.4 注意事項

命令 ROM 修正処理機能の使用に関して、次の点に注意してください。

- ・修正できる命令アドレスは最大 4 箇所です。
5 回目以降のエントリ・アドレス (0x10D) のコールでは、その修正内容は無視されます。
- ・イニシャル・リセット・ブートによって修正内容は無効になります。
リセット入力においてイニシャル・リセット・ブート (ノン・ブートを除く) を行うと代替レジスタの内容は初期化されます。
- ・開発ツールのソフトウェア・シミュレータ (HSM77016) での本機能のサポートはありません。
ブート (リブート) および命令 ROM 修正処理機能については HSM77016 でサポートをしていません。
- ・命令コードを取得するには HSM77016 を利用すると便利です。
HSM77016 の命令ウインドウにおいて、実際に修正する命令アドレスにおいて、修正する命令を記述します。
命令の表示形式を「二モニック」から「ヘキサ」に変更することによって、命令コードを取得できます。
特に分岐命令は相対ジャンプとなっていますので、実際に修正するアドレスで命令コードを取得しないと、分岐先が正しく得られないことがあります。

4.7 ブート所要時間

ブートに要する時間を、表4-6に示します。

表4-6 ブート所要時間

ブート・モード		時間(サイクル数)	
ブート	セルフ・ブート	ワード・ブート時 ^{※1}	$50 + (4 + 2D) \times W$
		バイト・ブート時 ^{※1}	$60 + (8 + 4D) \times W$
	ホスト・ブート ^{※1}		$40 + 6 \times W$ ^{※2}
リブート	セルフ・リブート	ワード・ブート時	$20 + (4 + 2D) \times W$
		バイト・ブート時	$20 + (8 + 4D) \times W$
	ホスト・リブート	μ PD77110 以外	$20 + 6 \times W$ ^{※2}
		μ PD77110	$20 + 20 \times W$ ^{※2}
	シグネチャ演算 (μ PD77110, 77113A, 77114)		$30 + 25 \times W$
	命令ROM修正処理 (μ PD77113A, 77114)		25 ~ 50
	内部データRAMのセルフ・チェック (μ PD77113A, 77114) ^{※1}		約 755000

注1. μ PD77113A, 77114 の場合、リセット・ブートを行うときに必ず内部データRAMのセルフ・チェックが行われます。

- ブートROMコードによる理論値です。ホストCPUのアクセス速度および μ PD77111ファミリのホスト・インタフェース仕様に依存します。

備考 W: ブート命令ワード数, D: データ・メモリ・ウエイト・サイクル

第 5 章 μ PD77115 の機能概要

この章では、 μ PD77115 の概要について説明します。

μ PD77115 は、デジタル・オーディオに特化した DSP で、MP3 (MPEG-1 Audio Layer3) AAC (MPEG-2 Advanced Audio Coding) などのオーディオ符号化方式のデコードを 1 チップで実現する低消費電力 DSP です。

複数のオーディオ符号化方式に対応するため命令 RAM を内蔵し、ソフトウェアの変更を可能にしています。

μ PD77111 ファミリのほかの製品との違いは、次のとおりです。

- ・ SD カード・インタフェース追加
- ・ オーディオ・シリアル・インタフェース追加
- ・ 16 ビット・タイマ追加
- ・ SD カード・レジスタから内蔵データ RAM への DMA 転送機能
- ・ シリアル・インタフェース 1 チャンネルのみ
- ・ 外部データ・メモリ空間なし
- ・ パラレル・ポートの 8 ビット化
- ・ ホスト・インタフェースの 16 ビット化

命令セット、演算機能、制限事項などはほかの μ PD77111 ファミリの製品と同じです。詳細については、**第 3 章 アーキテクチャ**、および μ PD77016 ファミリの **ユーザズ・マニュアル 命令編**を参照してください。

5.1 特 徴

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
- ・ PLL の通倍率を PLL0-PLL3 端子でリセット時に設定
- ・ アドレス・マッピングされたレジスタにより PLL バイパス・モードを選択可能
- ・ アドレス・マッピングされたレジスタによりクロック出力端子 CLKOUT のロウ・レベル固定出力を選択可能
- ・ JTAG 対応機能内蔵

動作周波数

- ・ 50 MHz (内部電圧 2.0 ~ 2.7 V 時)
- ・ 75 MHz (内部電圧 2.3 ~ 2.7 V 時)

メモリ空間

- ・ 内部命令メモリ RAM : 11.5K ワード × 32 ビット
- ・ 内部データ・メモリ RAM : 16K ワード × 16 ビット × 2 面 (X メモリ , Y メモリ)
- ・ 外部データ・メモリ なし

プログラミング

- ・ 16 ビット×16 ビット+40 ビット 40 ビット・マルチプライ・アキュムレータ
- ・ 8 個×40 ビットの汎用レジスタ
- ・ 8 個のデータ・メモリ・ポインタ・レジスタ (メモリ空間 2 つに 4 個ずつ)
- ・ 10 レベル割り込み (外部 4 レベル, 内部 6 レベル)
- ・ 3 オペランド命令 ($R0 = R0 + R1 * R2$)
- ・ 実行ステージはパイプラインなし

内蔵ペリフェラル

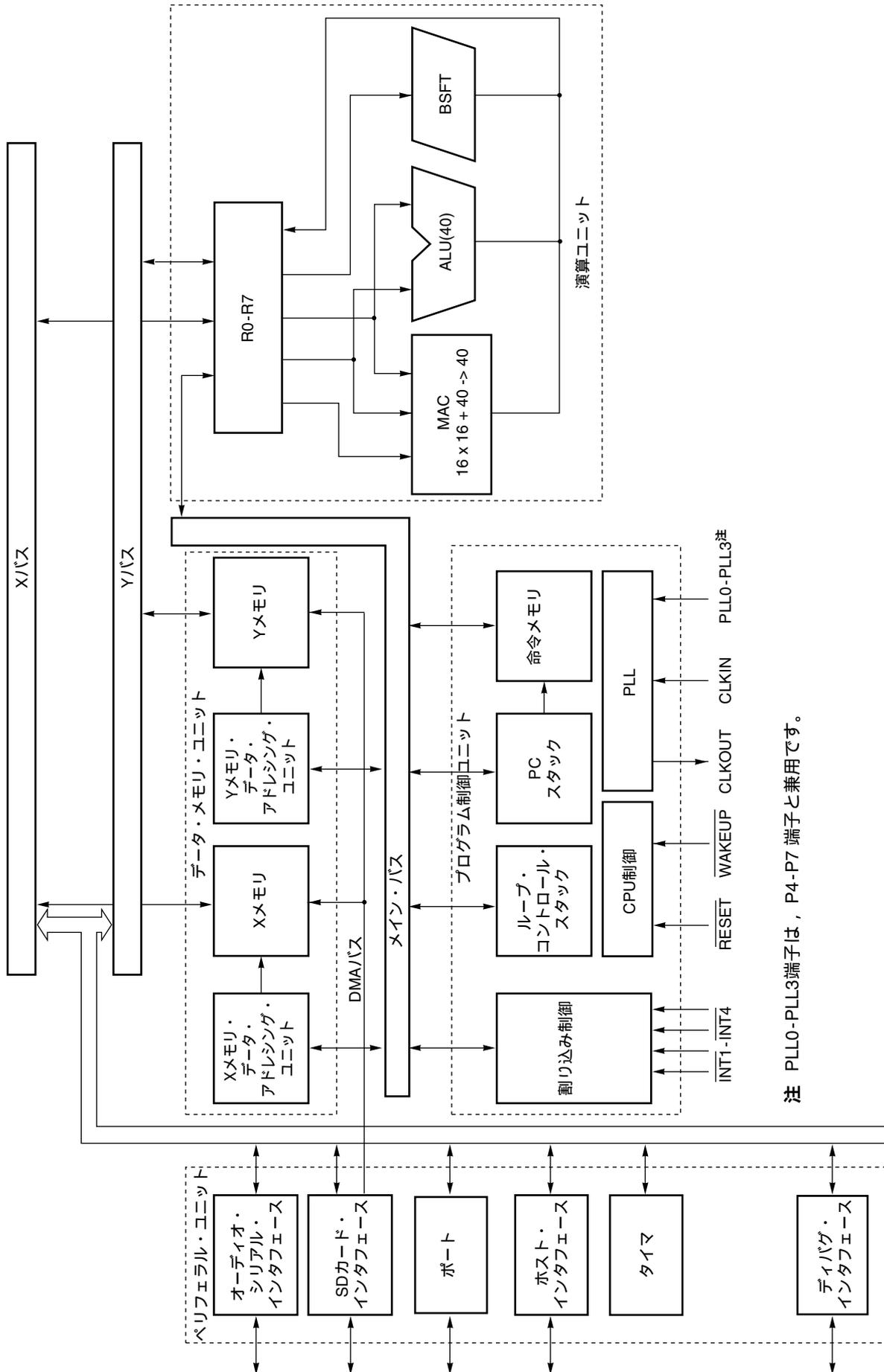
- ・ 8 本の I/O ポート
- ・ 1 セットの 32 ビット/64 ビット・オーディオ・シリアル・インタフェース
16 ビット汎用シリアル・インタフェースとしても使用可能。
- ・ 8 ビット/16 ビット・ホスト・インタフェース
- ・ SD カード・インタフェースおよび DMA 転送機能
- ・ 16 ビット・タイマ
- ・ クロック・コントロール

その他

- ・ CMOS プロセス
- ・ DSP コア用電源 2.0 ~ 2.7V, 外部 I/O 用電源 2.7 ~ 3.6 V

5.2 内部ブロック図

図5-1 μPD77115の内部ブロック図



注 PLL0-PLL3端子は、P4-P7端子と兼用です。

5.3 リセット機能

5.3.1 リセット・タイミング

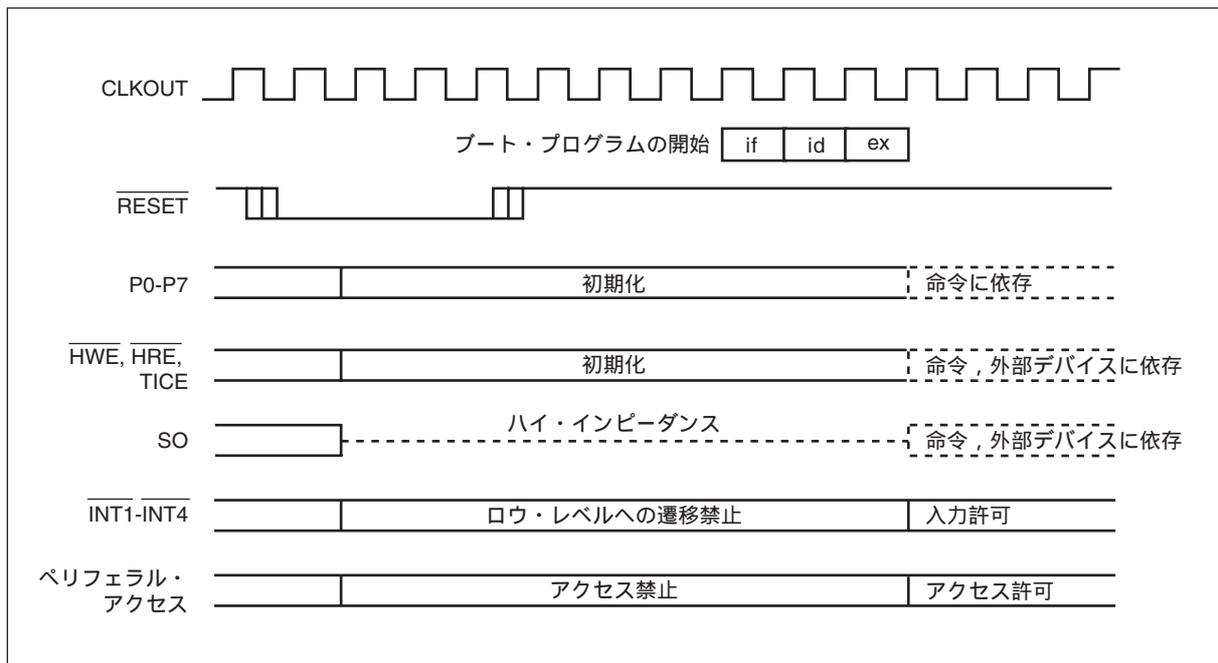
リセット信号を規定の期間(内部クロックの6サイクル分),アクティブ(ロウ・レベル)にすることで,リセットが実行されます。

リセット時には P0, P1 の値によってブート・モードを, PLL0-PLL3 の値によって PLL の逡倍率を設定します。P0, P1 および PLL0-PLL3 の値は, リセットが解除される 3 クロック前からリセットが解除されてから 50 クロック後までの間, 安定している必要があります。

電源投入時は, $\overline{\text{RESET}}$ 端子をアクティブ(ロウ・レベル)にする必要があります。パワーオン・リセット機能はありません。

リセット動作タイミングを図 5 - 2 に示します。

図 5 - 2 システム・リセット動作タイミング



リセットによって初期化される内部レジスタと初期値は次のとおりです。

表 5 - 1 初期化対象 CPU レジスタと初期値

レジスタ	初期値
SR	0xF000
PC	0
SP	0
EIR	0xFFFF
ESR	0
LC	0b1xxx xxxx xxxx xxxx (MSB = 1)
LSP	0
RC	0b1xxx xxxx xxxx xxxx (MSB = 1)

備考 “x” は不定を表します。

リセットによって初期化されるペリフェラル・レジスタと初期値は次のとおりです。

表 5 - 2 初期化対象ペリフェラル・レジスタと初期値

レジスタ	初期値
SST	0x0002
PCD	0x0000
HST	0x0301
ASST	0x8012
TIR	0xFFFF
TCR	0xFFFF
TCSR	0x0000
TENR	0x0000
CLKCNTL	0x0000
SDCTL	0x0202
SDCMD_IDX	0x4800
PSAR	0x0000
PSR	0x0000
PPR	0x0000
PCR	0x0000

リセットによって初期化される端子と初期値は表 5 - 3 のとおりです。

表 5 - 3 初期化対象端子と初期状態

対象端子	初期状態
CLKOUT	内部クロック出力（クロック出力禁止時はロウ・レベル）
SO	ハイ・インピーダンス
$\overline{\text{HRE}}$, $\overline{\text{HWE}}$	ハイ・レベル出力
HD0-HD15	ハイ・インピーダンス
P0-P7	ハイ・インピーダンス（入力端子）
TICE	ロウ・レベル出力
SDCLK	内部クロック出力（クロック出力禁止時はロウ・レベル）
SDCR	ハイ・インピーダンス（入力端子）
SDDAT	ハイ・インピーダンス（入力端子）

5.3.2 電源投入時の注意

電源投入時は、 $\overline{\text{RESET}}$ 端子をアクティブ（ロウ・レベル）にする必要があります。パワーオン・リセット機能はありませんので、 μ PD77115の端子状態、内部状態を初期化するために必ずリセット信号を入力してください。

また、 μ PD77115を完全に初期化するために、リセット信号のアクティブ（ロウ・レベル）期間中は最低6クロックのCLKINからシステム・クロックを入力してください。

電源投入時にリセット信号が入力されない場合は、端子状態が定まらず、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ことがあります。

5.3.3 電源投入時の推奨シーケンス

次のシーケンスでリセット信号を入力することによって、5.3.2 でおきる問題点を回避することができます。

- (1) リセット信号にロウ・レベルを入力してアクティブにします。
- (2) 電源を投入します。

このとき、DSP コア用電源と外部 I/O 電源の投入順はとくに規定はありませんが、同時または、DSP コア用電源→外部 I/O 電源の順を推奨しています。

- (3) クロック入力, P0, P1, PLL0-PLL3 の入力値を設定します (電源投入と同時にかまいません)。
- (4) 6 クロック以上, リセット信号をアクティブにします。
- (5) リセット信号を解除 (ハイ・レベル) します。
- (6) リセット解除から約 50 クロック後に, P0, P1, PLL0-PLL3 (P4-P7) をオープンにします。

5.4 スタンバイ機能

μ PD77115 には、低消費電流モードとしてホールト・モードとストップ・モードがあります。2 つのモードは対応する命令を実行することで対応するモードに遷移します。

5.4.1 ホールト・モード

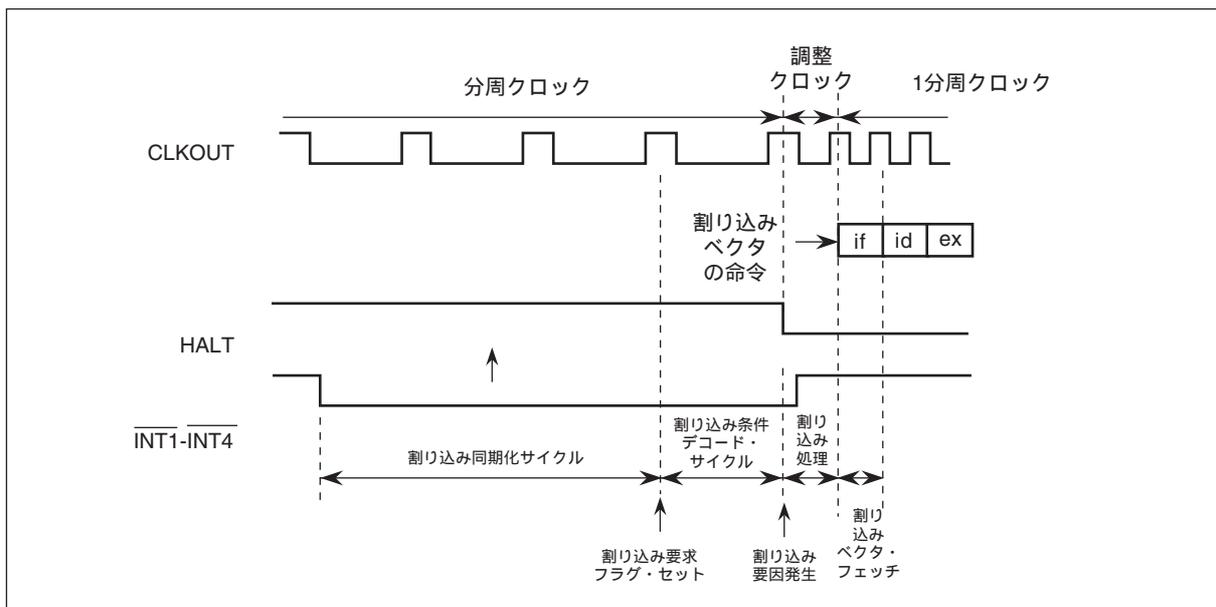
HALT 命令の実行により μ PD77115 の動作を停止します。

割り込みを受け付けるとホールト・モードを解除します。割り込み処理からの復帰先は、HALT 命令の直後の命令です。

μ PD77115 は HALT 命令の実行によりクロック制御部で設定した内部システム・クロックの 1/2 ~ 1/16 に分周した固定クロックに切り替えます。ホールト・モードでの端子割り込み受け付けタイミングを図 5-3 に示します。

ホールト・モードでの分周クロックは、デューティ 50% ではなく、ハイ・レベル幅が 1 サイクルのクロックとなります。

図 5-3 ホールト・モード割り込み受け付けタイミング



μ PD77115 の内蔵ペリフェラルは、ホールド・モードから影響を受けません。シリアル・インタフェース、ホスト・インタフェース、SD カード・インタフェースおよび汎用ポートはデータの入出力ができます。

シリアル・インタフェース、ホスト・インタフェース、SD カード・インタフェース、タイマからの割り込みを許可しておくことにより、ペリフェラルからのアクセス要求によりホールド・モードを解除することができます。

ペリフェラル用の端子、CLKOUT 端子および JTAG 用の端子を除く μ PD77115 の出力端子は、ホールド・モード中は HALT 命令実行の直前の状態を保持します。CLKOUT 端子は分周したクロックを出力し、JTAG コントローラはホールド・モードに影響を受けません。

ホールド・モード中の μ PD77115 の端子状態を表 5 - 4 に示します。

表 5 - 4 ホールド・モード中の端子状態

端子名	ホールド中の状態
CLKOUT	分周クロック出力
SO	影響を受けません。
$\overline{\text{HRE}}$, $\overline{\text{HWE}}$, HD0-HD15	
P0-P7	
SDCLK, SDCR, SDDAT	
TDO, TICE	

5.4.2 ストップ・モード

ストップ命令の実行によりプロセッサを停止します。

ホールド・モードとの違いを次に示します。

(1) ホールド・モードの場合

- ・ PLL を停止しないため、高速復帰が可能です。
- ・ 割り込みで復帰します。

(2) ストップ・モードの場合

- ・ PLL を停止することができます。ソフトウェアでレジスタ (CLKCNTL) に PLL オフ状態を設定すると、PLL を停止し、リーク電流のみの低消費電流となります。
- ・ ストップ・モード期間中は外部クロック供給を停止 (ロウ・レベル) することができます。
- ・ システム・リセット、もしくは $\overline{\text{WAKEUP}}$ 端子で復帰します。

ほかの μ PD77111 ファミリでは、ストップ・モード時は自動的に PLL がオフする機能になっていましたが、 μ PD77115 では、ストップ・モード時には自動的に PLL がオフしないため、ソフトウェアによって PLL を ON/OFF する必要があります。これによって内部 RAM 状態を保持したままで、リセットによる復帰も可能となります。

ストップ・モードはシステム・リセットで解除した場合、復帰先はブート ROM の 0 番地となります。 $\overline{\text{WAKEUP}}$ 端子で解除した場合は、次命令を実行します。

ストップ・モード中の μ PD77115 の端子状態を表 5 - 5 に示します。

表 5 - 5 ストップ・モード中の端子状態

端子名	ストップ中の状態
CLKOUT	ロウ・レベル出力
SO	影響を受けません。
$\overline{\text{HRE}}$, $\overline{\text{HWE}}$, HD0-HD15	
P0-P7	
SDCLK, SDCR, SDDAT	
TDO, TICE	

ストップ・モードは μ PD77115 の内蔵ペリフェラルには影響を与えません。ただし、 μ PD77115 の CPU は動作クロックを含めて停止しており、ペリフェラルが出力するアクセス要求フラグ、割り込みなどを CPU 側で検出することはできません。

スタンバイ・モードの詳細は 3.3.4 **スタンバイ機能**も参照してください。

5.5 クロック制御

5.5.1 概要

外部クロック入力 (CLKIN) から入力したクロック信号を PLL で逡倍して、システム・クロックを生成します。逡倍率は、PLL0-PLL3 端子を経由して読み込み、設定します。ユーザの逡倍設定は、ブート中に行うだけで、アプリケーション動作中に逡倍設定を直接変更することはできません。

CLKCNTL レジスタは PLL 動作の ON/OFF、クロック・ソース (外部クロックと逡倍クロック、非分周と分周出力) 選択、出力用の分周比設定、CLKOUT イネーブルを行います。

リセット直後は、外部クロックが直接システム・クロックに供給されています。ユーザ・プログラムの先頭で、PLL ロック時間 (約 100 μ s) のあと PLL のクロックに切り替えを行う必要があります。

内部に供給するシステム・クロックは、ハードウェアの状態によってクロック・ソース (外部、PLL 出力または分周出力) が変更できる構成になっています。

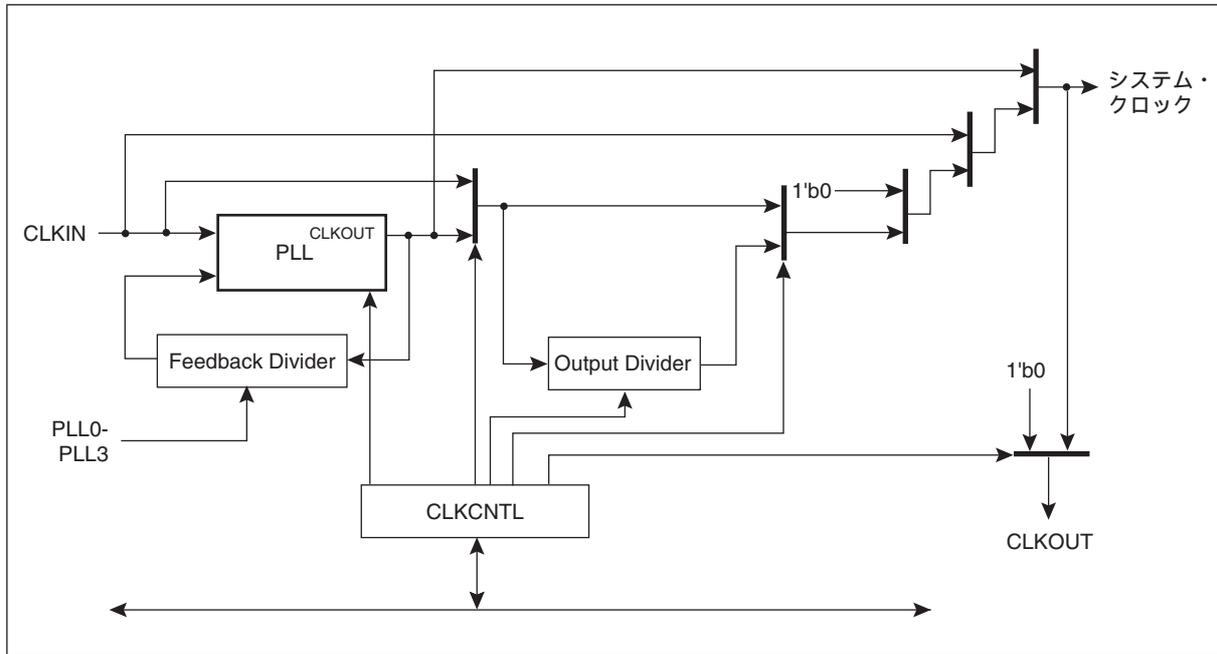
- ・ HALT モード時は自動的にクロック・ソースが分周回路からの出力を選択します。このときは CLKCNTL レジスタで設定した分周比のため、あらかじめ分周クロックで動作しているときの HALT モードはクロック不変です (分周クロックのデューティは 50%ではありません)
- ・ STOP モード時はシステム・クロックをマスクします。ただし、PLL は ON 状態です。STOP モードで PLL を停止したい場合は、システム・クロック供給を外部クロックに切り替えたあと、CLKCNTL レジスタの PLEN ビットを "0" にします。

クロック制御の主な特徴を次に示します。

- ・ PLL0-PLL3 端子で PLL 逡倍率の設定 (1-16 逡倍)
- ・ レジスタで、次の設定を行います。
 - PLL の ON/OFF 制御
 - 内部システム・クロックを外部供給クロック (PLL 入力クロック) か PLL 出力クロックの選択
 - CLKOUT 端子の出力許可 / 不許可の選択
 - 低速動作：分周回路の停止 / 動作の制御
 - 内部システム・クロックを分周クロックに切り替え制御
 - 低速動作：分周比設定 (1-16 分周)

クロック制御部のブロック図を図 5 - 4 に示します。

図5-4 クロック制御部ブロック



5.5.2 クロック制御部の端子構成

(1) CLKIN (クロック入力)

PLL の基準クロックおよび μ PD77115 内部のシステム・クロックを入力する端子です。

- ・ PLL 通倍が 1 の場合：最大周波数：75 MHz@2.3 V, 50 MHz@2.0 V
- ・ PLL 通倍が 16 の場合：最大周波数：4.68 MHz@2.3 V, 3.125 MHz@2.0 V
- ・ μ PD77115 内部のシステム・クロックとして使用する場合：0 ~ 75 MHz@2.3 V, 0 ~ 50 MHz@2.0 V

(2) CLKOUT (システム・クロック出力)

μ PD77115 内部のシステム・クロックを出力する端子です。

CLKCNTL レジスタの CKOEN ビットを 0 にすることで、CLKOUT 端子を 0 に固定できます。

CLKCNTL レジスタの PLLSEL ビットが 0 のときは、CLKIN の周期が CLKOUT に出力し、PLLSEL ビットが 1 のときは、PLL クロック (通倍クロック) の周期が出力されます。ホールド・モードおよび CLKCNTL レジスタの ODIVEN ビットが 1 のときは、ODIV ビットで設定した分周クロックの周期が出力されます。

(3) PLL0-PLL3 (PLL 入力)

PLL の逓倍を設定する端子で、汎用入出力ポートの P4-P7 と兼用です。

リセット解除後、PLL0-PLL3 で設定した値で逓倍率を設定します。PLL0-PLL3 と逓倍率の設定を表 5 - 6 に示します。

表 5 - 6 PLL 逓倍率設定

端子設定値				逓倍率
PLL3	PLL2	PLL1	PLL0	
0	0	0	0	16
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

5.5.3 CLKCNTL レジスタ

CLKCNTL レジスタは PLL 動作の ON/OFF, クロック・ソース選択 (外部クロックと逡倍クロック, 非分周と分周出力), 出力用の分周比設定, CLKOUT イネーブルを行うレジスタです。CLKCNTL の初期値 0x0000 で, XバスおよびYバスと入出力できます。CLKCNTL レジスタの各ビットの機能を表5-7に示します。

表5-7 CLKCNTL の機能

ビット	名称	ロード/ ストア	機能
15-9	リザーブ	-	リザーブ・ビット ・このビットには値を設定できない。読み出し時は不定。
8	PLLEN	L/S	PLL オン/オフ設定ビット 0: PLL オフ (デフォルト) 1: PLL オン PLLSEL が 1 (PLL 選択) のときに, PLLEN を 0 にすることを禁止する (デッドロックするため)
7	PLLSEL	L/S	システム・クロック・ソース選択ビット 0: 外部クロック供給 (デフォルト) 1: PLL 出力供給
6	CKOEN	L/S	CLKOUT 端子出力許可ビット CLKOUT 端子からシステム・クロックを出力する許可を行うビット 0: CLKOUT 出力不許可 (デフォルト, 0 固定) 1: CLKOUT 出力許可 (システム・クロック)
5	ODIVEN	L/S	低速動作用分周回路のオン/オフ設定ビット 0: 分周回路オフ (デフォルト) 1: 分周回路オン HALT モードおよび分周クロック選択を使用する場合は事前に ODIVEN ビットを 1 に設定して分周動作をさせておくこと。
4	ODIVSEL	L/S	分周クロック選択ビット 0: 外部クロック (PLLSEL : 0) 0: PLL 出力 (PLLSEL : 1) 1: 外部分周クロック (PLLSEL : 0) 1: PLL 分周クロック (PLLSEL : 1)
3-0	ODIV	L/S	分周クロック率設定ビット 0000 : 1/16 分周 (デフォルト) 0001 : 1/1 分周 0010 : 1/2 分周 0011 : 1/3 分周 0100 : 1/4 分周 0101 : 1/5 分周 0110 : 1/6 分周 0111 : 1/7 分周 1000 : 1/8 分周 1001 : 1/9 分周 1010 : 1/10 分周 1011 : 1/11 分周 1100 : 1/12 分周 1101 : 1/13 分周 1110 : 1/14 分周 1111 : 1/15 分周

5.5.4 クロック状態遷移

(1) PLL ロック・アンロック状態遷移

リセット直後、システム・クロックは外部クロック入力に設定され、PLL 動作は停止 (パワーダウン) 状態になります。ブート中に外部端子 PLL0-PLL3 の 4 ビットを読み込み、逡倍率が設定されます。

ブート・プログラム中の初期化設定で、PLLEN ビットが自動的に 1 になり PLL が動作を開始します。ブート・プログラム中で PLL 動作は開始しますが、PLLSEL ビットは 0 のままなのでシステム・クロックは外部クロックで動作しています。PLL クロックをシステム・クロックとして使用する場合、ブート処理が完了した後、PLL ロック時間 (ユーザ・プログラム中で約 100 μ s を計測) 待ってから、PLLSEL ビットを 1 に設定します。

PLL クロックを使用しない場合は、ホスト・ブート完了後、PLLEN ビットを 0 に設定し PLL を停止させ、PLL をパワーダウン状態にできます。

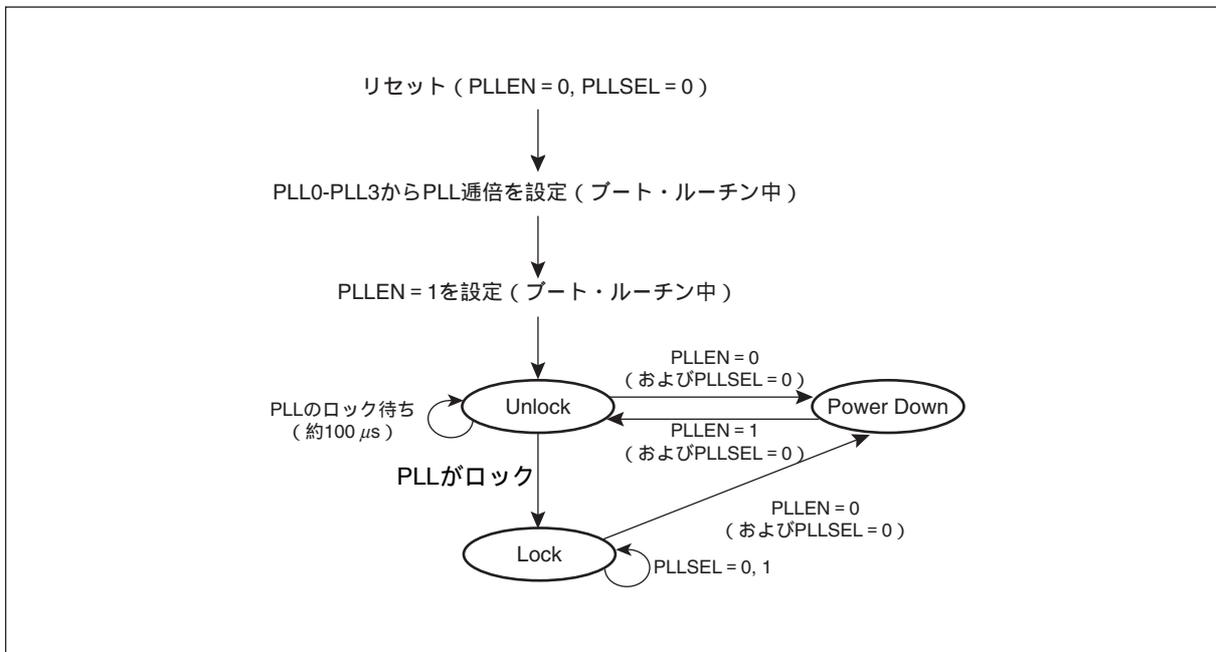
ノンブート指定でもブート・プログラム中の初期化設定は行われ、PLLEN ビットは自動的に 1 になり PLL は動作を開始します。

PLL パワーダウン状態で PLLEN ビットを 1 に設定すると PLL 動作を開始します。

注意 PLL 停止 (PLLEN = 0) に設定する場合は、あらかじめシステム・クロックを外部クロック入力にしておいてください。

PLL ロック・アンロック状態遷移図を図 5 - 5 に示します。

図 5 - 5 PLL ロック・アンロック状態遷移図



(2) システム・クロック状態遷移

分周クロックの選択はレジスタによる設定と HALT モードの 2 種類があります。レジスタ設定の場合は ODIVSEL ビットが 0 で非分周クロックを、1 で分周クロックを選択します。HALT モードでは自動的に分周クロックが選択されます。

非分周クロックへ戻す場合は、必ず分周クロックにしたときと同じクロック・ソース（外部クロック / PLL）にしなければなりません。

分周回路は消費電力を抑えるために ON/OFF の切り替えが可能です。したがって、分周クロックを選択する場合は必ず ODIVEN ビットを 1 に設定して分周回路を動作させる必要があります（HALT モードを使用する場合も必要です）。

レジスタによる分周クロック動作の場合は分周動作（ODIVEN = 1, ODIVSEL = 1）、非分周動作へ復帰（ODIVEN = 0, ODIVSEL = 0）と同時に設定可能ですが、HALT モードで使用する場合はあらかじめ ODIVEN = 1 に設定しておく必要があります。

5.5.5 PLL に関する留意事項

(1) 電源投入時、リセット時の PLL

μ PD77115 は電源投入直後、あるいはリセット直後、クロック・ソースが PLL になっていません（CLKIN 端子からの直接の入力になっています）。PLL を使用する場合は、ユーザ・プログラム中で設定する必要があります。PLL そのものはブート・ルーチン中で起動されています。ユーザ・プログラムからは CLKCNTL レジスタを用いて PLL の起動および停止、クロック・ソースの選択を行うことができます。

[PLL 起動から PLL 選択までの流れ]

電源投入、リセット

↓

PLL 起動（ブート・ルーチン中で自動的に起動^注）

↓

ブート・プロセス^注

↓

ユーザ・プログラム起動後、ユーザ・プログラム中で 100 μ s 待ちます（PLL のロックを待つ）

↓

ユーザ・プログラム中でクロック・ソースを PLL に設定

注 リセット時のノン・ブートを含みます

したがって、ブート終了後に実際にユーザ・プログラムの中で上記を実現するには、次のプログラムを実行することになります。

```
rep 8000 ; ←CLKIN 入力クロックに対して 100  $\mu$ s 以上になるようにリピート回数を指定
nop ;
r0l=*0x382e:y ; ←CLKCNTL レジスタの読み込み
r0=r0 | 0x0080 ; ←クロック・ソースに PLL を選択
*0x382e:y=r0l ; ←クロック・ソースの切り替え
```

PLLのロック待ちは、通常はユーザ・プログラム中で行うことになります。リピート回数は、nop実行の回数を指定するものですが、これはPLLがロックするのを待つものとなります。ロックしていないPLLのクロックを入力すると、誤動作の恐れがあります。PLLのロックには最大約100 μs程度かかりますので、CLKIN入力クロックに対して100 μs以上となれば構いません。

(2) STOPモードとPLL

STOPモード時にPLLを停止する場合は、いったんクロック・ソースを直接入力に切り替えてからPLLを停止する必要があります。

特に注意しなければならないのは、WAKEUP端子によるSTOPモードからの復帰を行う場合です。レジスタやメモリの内容を保持したまま復帰するには、必ずクロック・ソースを直接入力に設定する必要があります。リセットによる復帰の場合は、前述のとおりです。

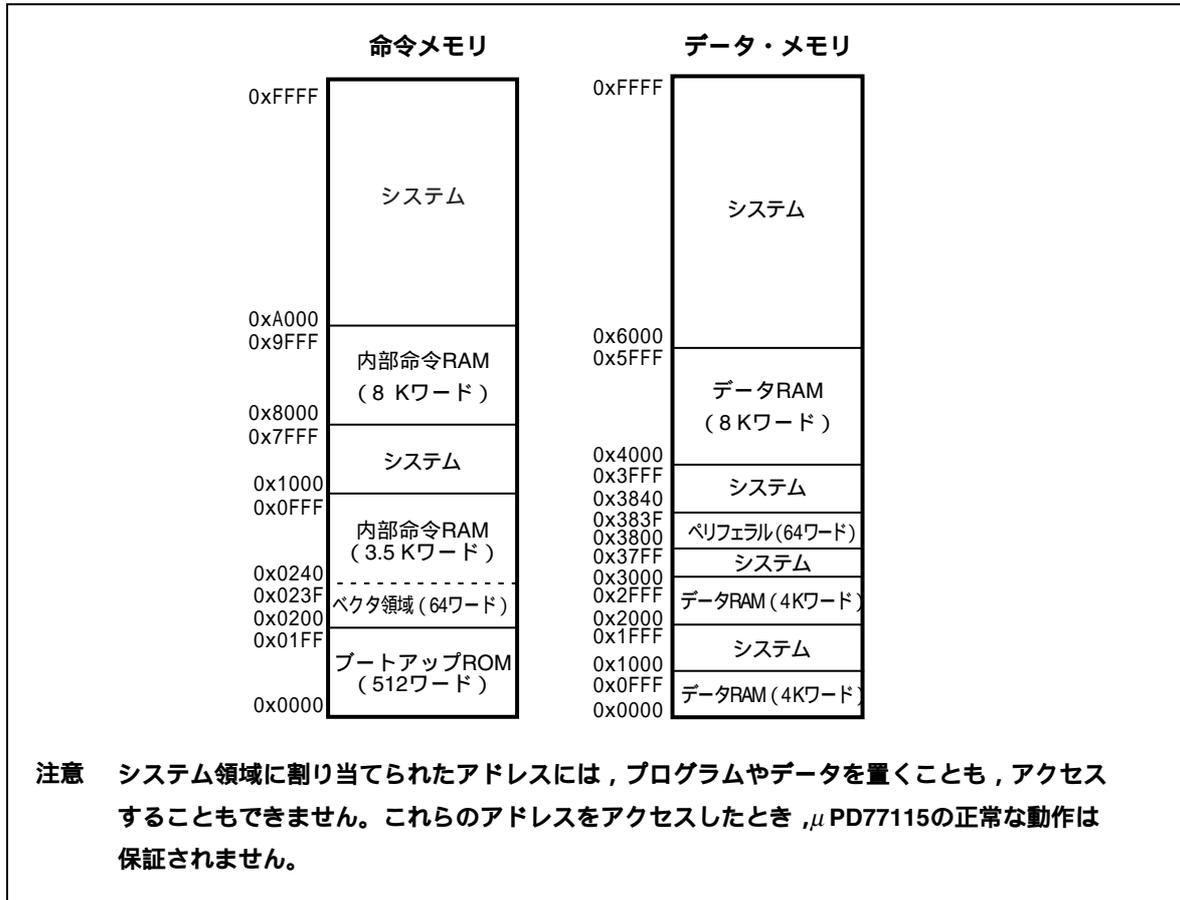
5.6 メモリ構成

μPD77115は1ワードが32ビットの命令メモリ空間を1面と、1ワードが16ビットのデータ・メモリ空間を2面持っています(命令メモリとデータ・メモリ空間を分離したハーバード・アーキテクチャを採用しています)

μPD77115は、11.5Kワード×32ビットの命令RAMを内蔵しています。μPD77115は、16Kワード×16ビット×2面のデータRAMを内蔵します。

μPD77115は、データ・メモリ空間に64ワードの内蔵ペリフェラル領域を持っています。

図5-6 μPD77115のメモリ・マップ



5.7 割り込みベクタ・テーブル

μ PD77115 ではペリフェラル・マクロからの割り込みおよび端子からの割り込みを 12 本持ちます。割り込みベクタ・テーブルを表 5 - 8 に示します。

表 5 - 8 割り込みベクタ・テーブル

ベクタ	割り込み要因
0x200	リセット
0x204	予約
0x208	
0x20C	
0x210	INT1
0x214	INT2
0x218	INT3
0x21C	INT4
0x220	SI 入力
0x224	SO 出力
0x228	SDDAT 入力 / PBU
0x22C	SDDAT 出力
0x230	HI 入力
0x234	HO 出力
0x238	SDCR 入力
0x23C	タイマ

注意 割り込みベクタ 0x0228 は、SD カード・インタフェースからの割り込みと PBU からの割り込みが兼用になっており、PBU の PCR レジスタ内にある EN ビットによって切り替わります。EN = 1 の場合、PBU による DMA 転送完了後の割り込みが 0x0228 の割り込みとなります。EN = 0 の場合、SDDAT 端子からのデータ入力割り込みが 0x0228 の割り込みとなります。

SR レジスタの要因別割り込み許可フラグの対応は次のようになります。

ビット	割り込み要因
11	Timer
10	SDCR
9	HO
8	HI
7	SDDAT (出力)
6	SDDAT (入力) / PBU
5	SO
4	SI
3	INT4
2	INT3
1	INT2
0	INT1

第6章 μ PD77115 のペリフェラル

この章では μ PD77115 のペリフェラルについて説明します。ほかの μ PD77111 ファミリと共通の機能については **第3章 アーキテクチャ**の該当箇所を参照してください。

μ PD77115 ではペリフェラル・レジスタを X メモリ空間および Y メモリ空間にメモリ・マップしています。ペリフェラル・レジスタはメモリと同様なアクセスを行います。 μ PD77115 の内蔵ペリフェラルを次に示します。

- ・シリアル・インタフェース
- ・ホスト・インタフェース
- ・汎用入出力ポート
- ・SD メモリ・カード・インタフェース
- ・ペリフェラル・バッファ
- ・タイマ

6.1 ペリフェラル・レジスタ

ペリフェラル関係のレジスタはXメモリ空間およびYメモリ空間をマルチプレクスしたペリフェラル・バスに接続されています。そのためXメモリおよびYメモリ空間上にメモリ・マップして見えます。内蔵ペリフェラル・レジスタを、Xメモリ空間およびYメモリ空間の両方から同時にアクセスすることを禁止します。各レジスタのメモリ・マッピングを表6-1に示します。

表6-1 ペリフェラル・レジスタのメモリ・マッピング

X/Yメモリ・アドレス	レジスタ名	機能	ペリフェラル名
0x3800	SDT/ASDT	シリアル・データ・レジスタ	ASIO
0x3801	SST	シリアル・ステータス・レジスタ	
0x3802	ASST	オーディオ・シリアル・ステータス・レジスタ	
0x3803	予約領域	注意 この領域にはアクセスしないでください。	-
0x3804	PDT	ポート・データ・レジスタ	PIO
0x3805	PCD	ポート・コマンド・レジスタ	
0x3806	HDT	ホスト・データ・レジスタ	HIO
0x3807	HST	ホスト・ステータス・レジスタ	
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	SDDR	SDカード・データ・レジスタ	SDCIF
0x3811	SDCMD_IDX	SDカード・コマンド・レジスタ・インデクス	
0x3812	SDCMD_AGH	SDカード・コマンド・レジスタ・アークギュメント(上位)	
0x3813	SDCMD_AGL	SDカード・コマンド・レジスタ・アークギュメント(下位)	
0x3814	SDCTL	SDカード・コントロール・レジスタ	
0x3815	SDRPR	SDカード・レスポンス・レジスタ	
0x3816	SDSBR	SDカードCRCステータス・ビジィ・レジスタ	
0x3817-0x381F	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	TIR	タイマ初期値レジスタ	TIMER
0x3821	TCR	タイマ・カウント・レジスタ	
0x3822	TCSR	タイマ・コントロール/ステータス・レジスタ	
0x3823	TENR	タイマ・カウント・イネーブル・レジスタ	
0x3824-0x382D	予約領域	注意 この領域にはアクセスしないでください。	
0x382E	CLKCNTL	クロック・コントロール・レジスタ	PLL
0x382F	予約領域	注意 この領域にはアクセスしないでください。	-
0x3830	PSAR	DMAスタート・アドレス・レジスタ	PBU
0x3831	PSR	DMAサイズ・レジスタ	
0x3832	PRR	DMAポインタ・レジスタ	
0x3833	PCR	DMAコントロール・レジスタ	
0x3834-0x383F	予約領域	注意 この領域にはアクセスしないでください。	-

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。

6.2 シリアル・インタフェース

μ PD77115は、シリアル・インタフェースを1チャンネル内蔵しており、従来の μ PD77111ファミリと同様のスタンダード・シリアル・インタフェースの機能に加えて、一般的なオーディオ・コーデックのLSIが備えている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・インタフェースを内蔵しています。スタンダード・シリアル・インタフェースとオーディオ・シリアル・インタフェースの機能を同時に動作させることはできません。レジスタによりこれらの機能を切り替えます。

オーディオ・シリアル・インタフェースは2つのモードがあり、 μ PD77115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 μ PD77115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。

主な特徴を次に示します。

(1) オーディオ・シリアル・インタフェース

- ・クロック供給
 - マスタ・モード
 - MCLK：マスタ・クロック（入力）
 - BCLK：オーディオ・シリアル・クロック（出力）
 - LRCLK：レフト/ライト・クロック（出力）
 - スレーブ・モード
 - MCLK：未使用
 - BCLK：オーディオ・シリアル・クロック（入力）
 - LRCLK：レフト/ライト・クロック（入力）
- ・ビット長
 - 32ビットまたは64ビットを選択
- ・内部データ・バスとの接続
 - MSBファーストで入出力
- ・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

(2) スタンダード・シリアル・インタフェース

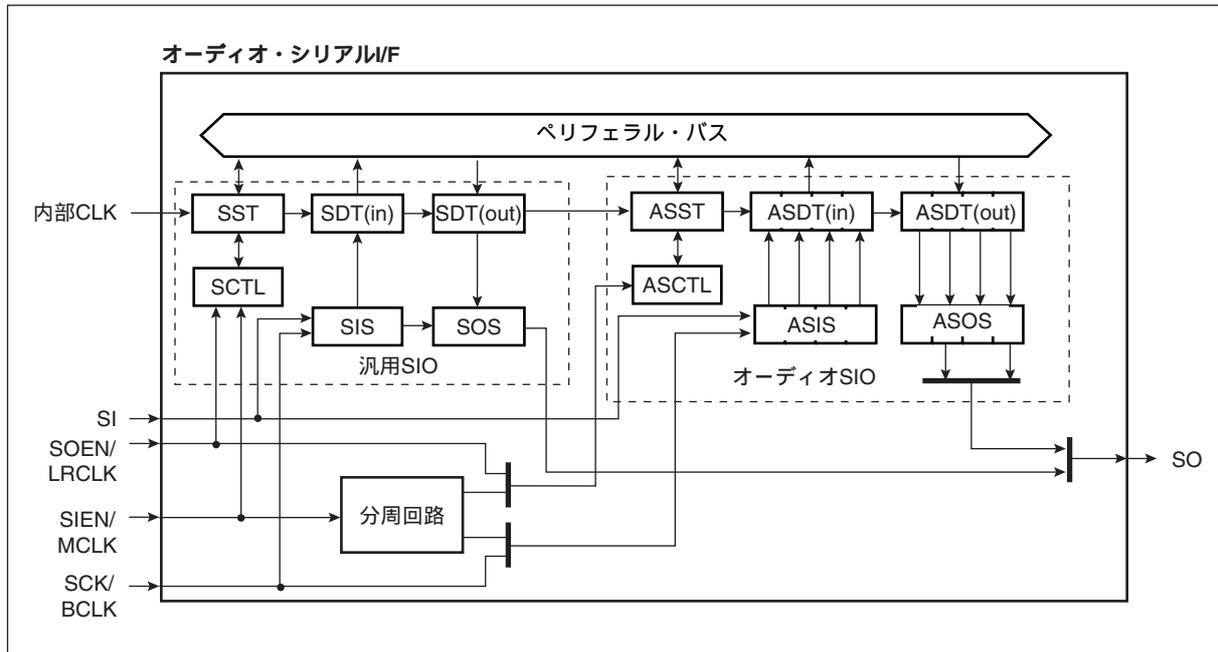
- ・クロック供給
 - SCK端子：外部クロック入力（SI/SO共用）
- ・ビット長
 - SI/SO独立に、16ビットまたは8ビットを選択
- ・内部データ・バスとの接続
 - SI/SO独立に、MSBファーストまたはLSBファーストを選択
- ・内部ハンドシェイク
 - ポーリング、ウエイトまたは割り込みによるハンドシェイク

シリアル・インタフェース・ブロック内は、スタンダード・シリアル・インタフェース（汎用SIO）とオーディオ・シリアル・インタフェース（オーディオSIO）があります。

汎用SIOの制御回路（SCTL）が、汎用SIOの機能として使用する際に端子およびレジスタの制御を行い、オーディオSIOの制御回路（ASCTL）が、オーディオSIOの機能として使用する際に、端子およびレジスタの制御を行

います。シリアル・インタフェースのブロック図を図6-1に示します。

図6-1 シリアル・インタフェース



6.2.1 オーディオ・シリアル・インタフェース

(1) オーディオ・シリアル・インタフェースの端子

オーディオ・シリアル・インタフェース端子は、スタンダード・シリアル・インタフェース端子と兼用です。

(a) MCLK (マスタ・クロック入力)

マスタ・モード時は入力、スレーブ・モード時は未使用（外部でGNDと接続してください）になります。スタンダード・シリアル・インタフェースのSIEN端子と兼用です。

マスタ・モード時に、シリアル動作のクロックであるBCLKとLRCLKのマスタ・クロックとなる入力端子です。MCLKから入力したクロックを分周して、BCLKクロックとLRCLKクロックをシリアル・インタフェース内部で生成します。BCLKクロックはオーディオ・シリアル・インタフェースのシフト動作に使用すると同時にBCLK端子から出力し、 μ PD77115外部のオーディオ・コーデックのBCLKクロックになります。また、LRCLKクロックはオーディオ・シリアル・インタフェースの制御動作に使用すると同時にLRCLK端子から出力し、 μ PD77115外部のオーディオ・コーデックのLRCLKクロックになります。

(b) BCLK (シリアル・クロック入出力)

マスタ・モード時は出力、スレーブ・モード時は入力（デフォルト）になります。

リセット直後はスレーブ・モードであるため入力になっています。スタンダード・シリアル・インタフェースのSCK端子と兼用です。

オーディオ・シリアル・インタフェースのシリアル・クロックでシリアル・データ入出力用のクロック入出力端子です。

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BCLK端子から出力し、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、BCLK 端子から入力したクロックを直接シリアル・クロックとして使用しません。

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 BCLK に同期して行います。

マスタ・モード時はレジスタ設定によって出力を“0”固定にすることができます(デフォルト)。

(c) LRCLK (レフト/ライト・クロック入出力)

マスタ・モード時は出力、スレーブ・モード時は入力(デフォルト)になります。

リセット直後はスレーブ・モード(入力)になっています。スタンダード・シリアル・インタフェースの SOEN 端子と兼用です。

オーディオ・シリアル・データの 32 ビット/64 ビットの同期する信号で、32 ビットの場合は、BCLK の 32 周期が LRCLK の 1 周期になり、64 ビットの場合は、BCLK の 64 周期が LRCLK の 1 周期になります。

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 LRCLK 端子から出力し、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、LRCLK 端子から入力したクロックを直接フレーム・クロックとして使用します。

LRCLK の立ち下がりに同期してオーディオ・シリアル入力シフト・レジスタ (ASIS) からオーディオ・シリアル・データ入力レジスタ (ASDT (in)) へシリアル - パラレル変換され、オーディオ・シリアル・データ出力レジスタ (ASDT (out)) からオーディオ・シリアル出力シフト・レジスタ (ASOS) へパラレル - シリアル変換されます。

LRCLK 端子の信号の極性とフレーム・データの L/R はレジスタにより Low-High で LR 出力、High-Low で LR 出力のどちらにも設定できます。使用する DAC またはコーデックの仕様に合わせて変更する必要があります。

マスタ・モード時はレジスタ設定によって出力を“0”固定にすることができます(デフォルト)。

(d) SO (シリアル・データ出力)

シリアル・データの出力端子です。スタンダード・シリアル・インタフェースの SO 端子と兼用です。BCLK の立ち下がりに同期して出力が変化します。

マスタ・モード時はレジスタ設定によって出力を“0”固定にすることができます(デフォルト)。

(e) SI (シリアル・データ入力)

シリアル・データの入力端子です。スタンダード・シリアル・インタフェースの SI 端子と兼用です。BCLK の立ち上がりに同期してサンプリングします。

(2) オーディオ・シリアル・インタフェースのレジスタ

(a) オーディオ・シリアル・データ・レジスタ (ASDT)

オーディオ・シリアル・データ・レジスタ (ASDT) はシリアル・データを入出力するための 64 ビット・レジスタです。シリアル出力用とシリアル入力用のレジスタを別々に持っています。ASDT の値は X バスおよび Y バスと入出力できます。

・オーディオ・シリアル・データ出力レジスタ (ASDT (out))

シリアル出力するデータを設定する 64 ビットのレジスタです。シリアル出力用の ASDT に対するストア命令を実行すると ASDT にペリフェラル・バスからデータを入力します。SO から MSB ファーストで出力します。

シリアル出力シフト・レジスタ (ASOS) が空になると ASDT の値を ASOS に設定します。ASSEF が「0」のときに ASDT に対するストア命令を実行すると、ASSER 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、ペリフェラル・バスから ASDT (out) へ 16 ビット・データを 4 回に分けてストアします。 μ PD77115 から見える ASDT は 1 つのレジスタとしてアドレス・マッピング (0x3800) しているため、同じアドレスに対して上位ビット側 (ビット 63-48) から順番にビット 47-32, ビット 31-16, ビット 15-0 の 4 回ストアすることで 64 ビットのデータが ASDT (out) にストアできます。

32 ビットのデータ転送を行うときは、ペリフェラル・バスから ASDT (out) へ 16 ビット・データを 2 回に分けてストアします。64 ビット・データ転送と同様に、同じアドレスに対して上位ビット側 (ビット 31-16) から順番にビット 15-0 を 2 回ストアすることで 32 ビットのデータが ASDT (out) にストアできます。

ASSEF ビットは 4 回 (64 ビット転送) または 2 回 (32 ビット転送) 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。

・オーディオ・シリアル・データ入力レジスタ (ASDT (in))

シリアル入力されたデータを読み出す 64 ビットのレジスタです。

シリアル入力用の 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 ASDT のデータを入力します。SI から MSB ファーストで入力します。

シリアル入力シフト・レジスタ (ASIS) に最後のビットが入力されると、ASIS の値を ASDT に設定します。ASLEF が「0」のときに ASDT からのロード命令を実行すると、ASLER 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、ASDT (in) からペリフェラル・バスへ 16 ビット・データを 4 回に分けてロードします。 μ PD77115 から見える ASDT は 1 つのレジスタとしてアドレス・マッピング (0x3800) しているため、同じアドレスに対して上位ビット側 (ビット 63-48) から順番にビット 47-32, ビット 31-16, ビット 15-0 の 4 回ロードすることで 64 ビットのデータを ASDT (in) からロードできます。

32 ビットのデータ転送を行うときは、ASDT (in) からペリフェラル・バスへ 16 ビット・データを 2 回に分けてロードします。64 ビット・データ転送と同様に同じアドレスに対して上位ビット側 (ビット 31-16) から順番にビット 15-0 の 2 回ロードすることで 32 ビットのデータを ASDT (in) からロードできます。

ASLEF ビットは 4 回 (64 ビット転送) または 2 回 (32 ビット転送) 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。

(b) オーディオ・シリアル・ステータス・レジスタ (ASST)

オーディオ・シリアル・ステータス・レジスタ (ASST) は、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・シリアル切り替え、分周比設定など、 μ PD77115 の CPU コア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 の値は X バスおよび Y バスと入出力できます。リセット時の値は、0x8012 です。

ASST のビット構成については表 6 - 2 に示します。

(c) オーディオ・シリアル出力シフト・レジスタ (ASOS)

オーディオ・シリアル出力シフト・レジスタ (ASOS) は、シリアル・データを SO 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 (シリアル・データ出力レジスタ) から新しいデータを入力します。ASOS はペリフェラル・バスと接続していません。

(d) オーディオ・シリアル入力シフト・レジスタ (ASIS)

オーディオ・シリアル入力シフト・レジスタ (ASIS) は、SI 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 (シリアル・データ入力レジスタ) にデータを出力します。ASIS は、ペリフェラル・バスと接続していません。

(e) シリアル・ステータス・レジスタ (SST)

このレジスタはスタンダード・シリアル・モードのときのステータス・レジスタですが、ビット 6 およびビット 7 はオーディオ・シリアル・インタフェース用の設定ビットとして使用します。SST のビット 6, 7 については表 6 - 3 に示します。

表6-2 ASSTの機能(1/2)

ビット	名称	ロード/ ストア	各ビットの機能
15	SOAD	L/S	スタンダード/オーディオ切り替えビット ・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)
14	ASOEN	L/S	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
13	ASIEN	L/S	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
12-10	BDIV	L/S	マスタ・クロックから BCLK/LRCLK を生成するとき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
9	ADRST	L/S	オーディオ・シリアル・リセット許可ビット 入出力回路およびフラグがリセット(ASSE=0, ASSEF=1, ASLER=0, ASLEF=0)される。リセット後, ADRST は自動的に0にクリアされる。 ・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
8	ASSWE	L/S	ASDT 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。 ASDT(out)にデータが残っている状況でストアした場合に, μ PD77115にウエイトを挿入する。
7	ASLWE	L/S	ASDT 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。 ASDT(in)にデータがない状況でロードした場合に, μ PD77115にウエイトを挿入する。
6	ADOBL	L/S	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
5	ADIBL	L/S	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
4	MSSEL	L/S	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) リセット後,一度マスタ・モードへ切り替えたあとスレーブ・モードへ切り替えることはできない。

表6-2 ASSTの機能(2/2)

ビット	名称	ロード/ ストア	各ビットの機能
3	ASSER	L/S	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 PD77115がASDTに書き込んだ場合に1なる。 一度セットされると, μ PD77115が0を書き込むまで変化しない。
2	ASLER	L/S	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 PD77115がASDTを読み出した場合に1なる。 一度セットされると, μ PD77115が0を書き込むまで変化しない。
1	ASSEF	L	ASDT ストア許可フラグ ASDT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なる(デフォルト)。 μ PD77115がASDTに書き込むと0になる。
0	ASLEF	L	ASDT 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ASDTに転送した場合に1なる。 μ PD77115がASDTを読み出すと0なる(デフォルト)。

備考 L: 読み出し専用, L/S: 読み出し/書き込み可能

表6-3 SSTのビット6,7の機能

ビット	名称	ロード/ ストア	各ビットの機能
7	INVLR	L/S	LRCLK 極性反転ビット ・0: LRCLKがロウ・レベル→ハイ・レベルの順番で1サンプリングのデータを入出力(デフォルト) ・1: LRCLKがハイ・レベル→ロウ・レベルの順番で1サンプリングのデータを入出力
6	DACPC	L/S	シリアル系の端子をロウ・レベル固定にするためのビット マスタ・モード時 ・0: SO, LRCLK, BCLKは0固定出力。SIは0固定入力(デフォルト) ・1: 通常動作 スレーブ・モード時 ・0: SOは0固定出力。SIは0固定入力(デフォルト) ・1: 通常動作 初期状態ではオーディオ出力が有効になっていないので注意してください。

備考 L/S: 読み出し/書き込み可能

(3) 各クロックの周波数の関係

LRCLK は、オーディオ信号のサンプリング周波数で動作するクロックであり、MCLK の周波数に対して $1/256$ 、 $1/384$ 、 $1/512$ の3種類があります。また、1フレームで扱う転送ビットは32ビットと64ビットの2種類があります。

MCLK、BCLK、LRCLK の周波数の関係を表6-4に示します。

表6-4 各クロックと周波数の関係

MCLK	256 fs		384 fs		512 fs	
転送ビット	32 ビット	64 ビット	32 ビット	64 ビット	32 ビット	64 ビット
BCLK	$1/8 * \text{MCLK}$	$1/4 * \text{MCLK}$	$1/12 * \text{MCLK}$	$1/6 * \text{MCLK}$	$1/16 * \text{MCLK}$	$1/8 * \text{MCLK}$
LRCLK	$1/256 * \text{MCLK}$	$1/256 * \text{MCLK}$	$1/384 * \text{MCLK}$	$1/384 * \text{MCLK}$	$1/512 * \text{MCLK}$	$1/512 * \text{MCLK}$

(4) オーディオ・シリアル・インタフェースのタイミング

(a) オーディオ・シリアル出力タイミング

ASDT に対するストア命令を実行すると、ASST の ASSEF が 0 になります。ASST の ASSWE が 1 で ASSEF が 0 のときに ASDT にストア命令を実行すると、ウエイトが発生します。

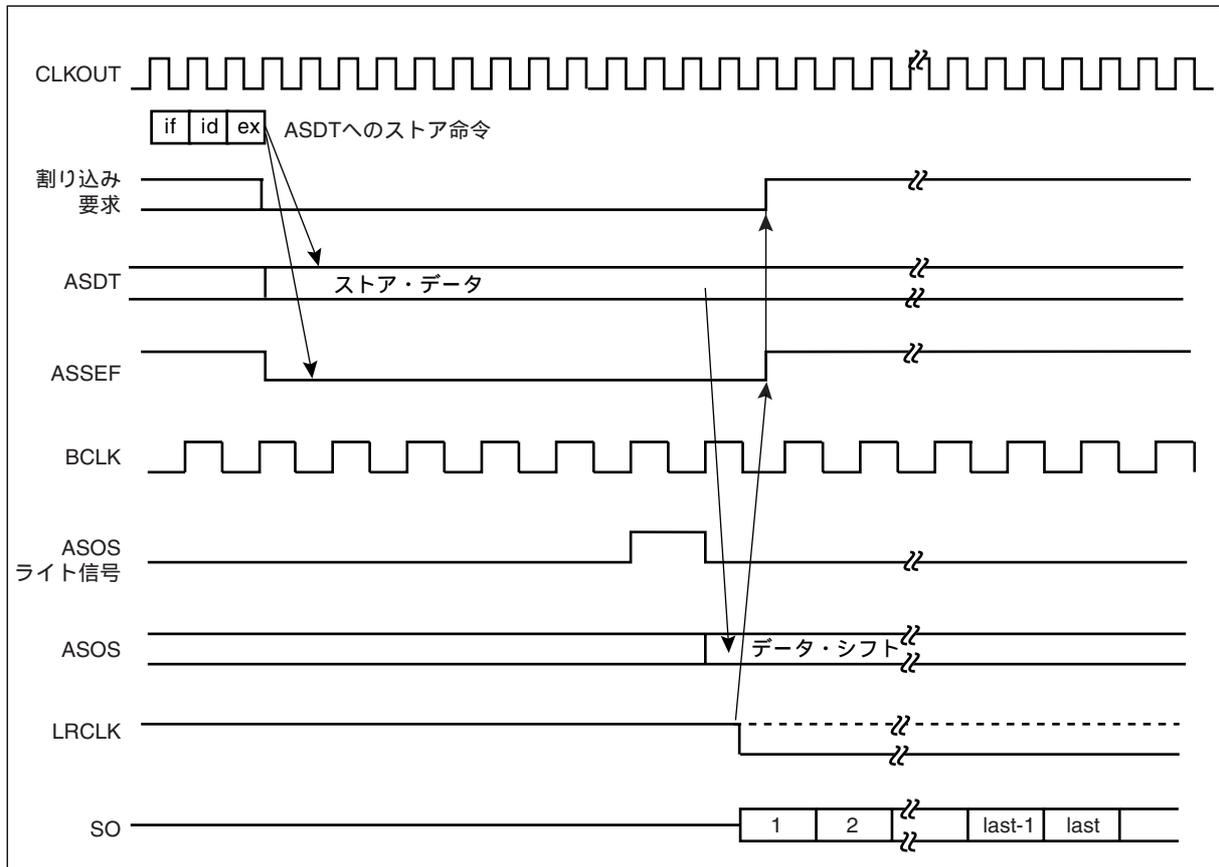
オーディオ・シリアル出力シフト・レジスタ (ASOS) が空の場合は、ASDT (シリアル・データ出力レジスタ) の値を ASOS に設定します。同時に ASSEF のセット要求を出力します (ASSEF はシステム・クロックに同期したセット要求によりセットします)。また、セットした ASSEF によって μPD77115 に対し SO 割り込み要求を出力し、ASSWE が 1 でウエイト状態にあるときには、ウエイトを解除します。

LRCLK をロウ・レベルにすると、BCLK の立ち下がりに同期して SO からシリアル・データを出力します。

最後のビットの出力を終了すると、SO はハイ・インピーダンスになります。

オーディオ・シリアル出力タイミングを図 6 - 2 に示します。

図 6 - 2 オーディオ・シリアル出力タイミング



(b) オーディオ・シリアル入力タイミング

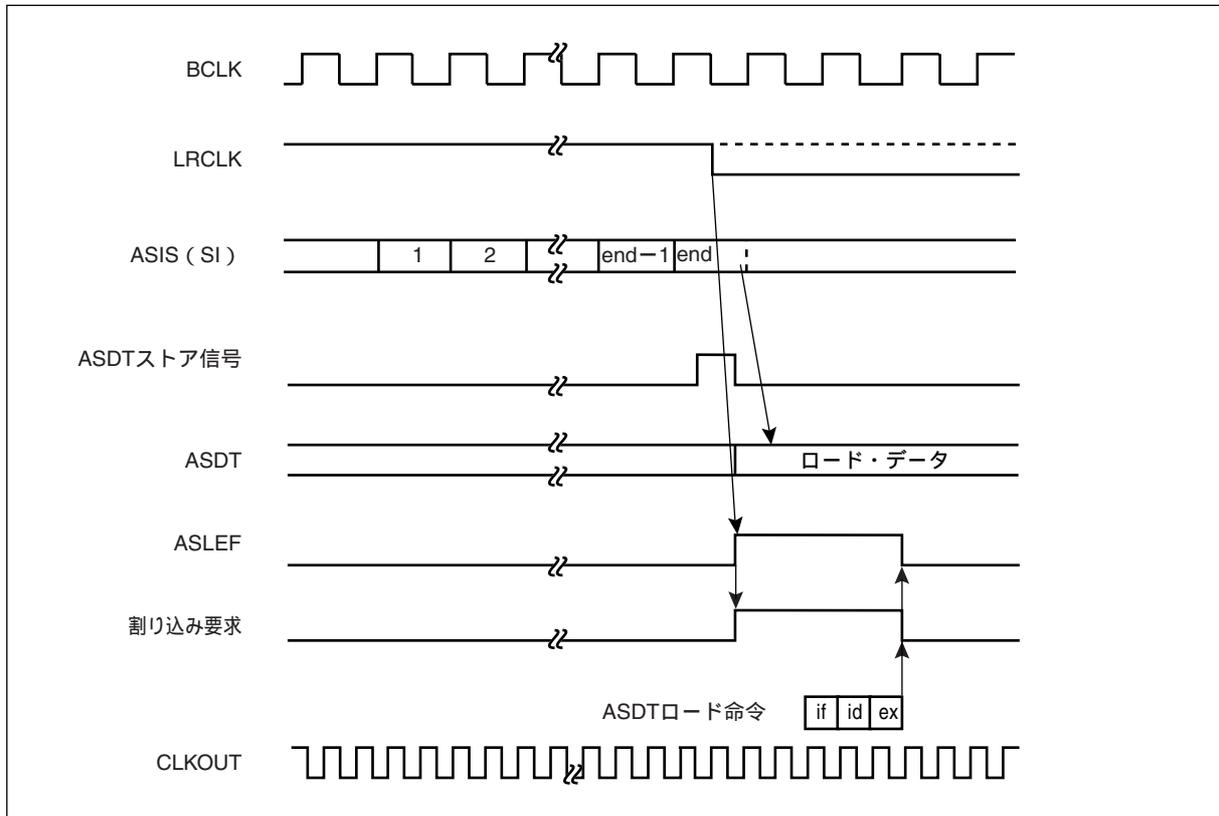
オーディオ・シリアル入力シフト・レジスタ (ASIS) が空である場合に LRCLK がロウ・レベルであるとき、BCLK 立ち上がりで同期して SI からシリアル・データを入力します。

最終 - 1 ビットのデータを、オーディオ・シリアル入力シフト・レジスタ (ASIS) に入力すると、ASST の ASLEF が 0 の場合には ASLEF セット要求を出力します (ASLEF はシステム・クロックに同期したセット要求によりセットします)。次に、最終ビットのデータを ASIS に入力すると同時に ASIS の値を ASDT (シリアル・データ入力レジスタ) に設定します。また、セットした ASLEF によって μPD77115 に対して SI 割り込み要求を出力し、ASLWE が 1 でウエイト状態にあるときには、ウエイトを解除します。

ASDT からのロード命令を実行すると ASLEF が 0 になります。ASST の ASLWE が 1 で ASLEF が 0 のときに、ASDT からのロード命令を実行するとウエイトが発生します。

オーディオ・シリアル入力タイミングを図 6 - 3 に示します。

図 6 - 3 オーディオ・シリアル入力タイミング



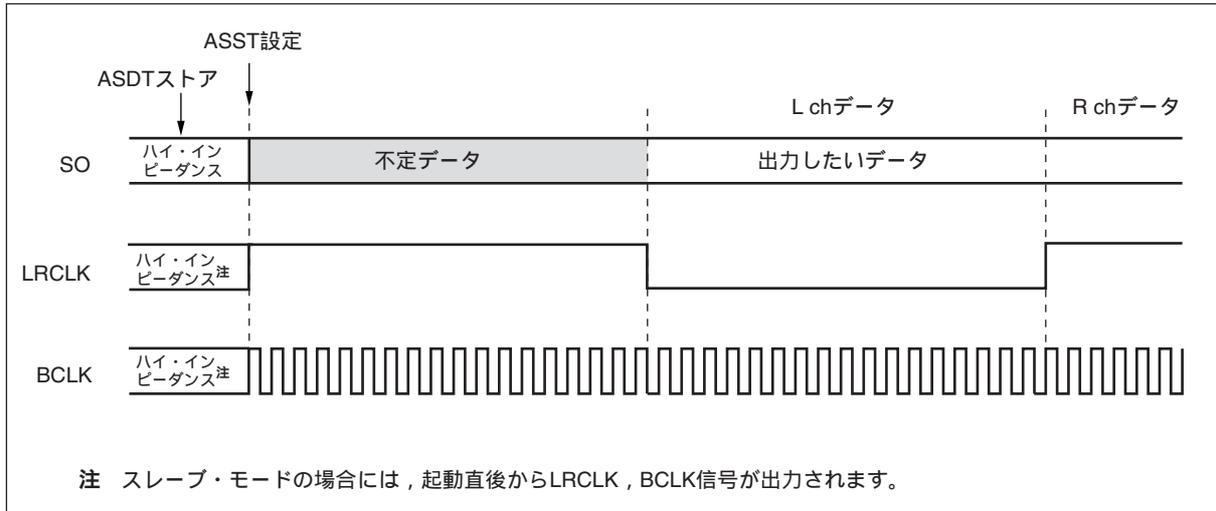
(5) 起動時のオーディオ・シリアル・インタフェース (ASIO) に関する注意事項

μ PD77115を起動後、オーディオ・シリアル・インタフェース動作開始時に、SO 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。SO を使用するときには、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 6-4 ASIO 動作開始時の SO 動作 (マスタ・モードの場合)



6.2.2 スタンダード・シリアル・インタフェース

スタンダード・シリアル・インタフェースは μ PD77111ファミリのほかの製品と同等の機能です。詳しい説明は3.7.3 シリアル・インタフェースを参照してください。

また、スタンダード・シリアル・インタフェース端子および内部シリアル・レジスタ (SDT) は、オーディオ・シリアル・インタフェースと兼用になっています。

6.3 ホスト・インタフェース

μ PD77115は、外部のホストCPUやDMAコントローラなどのデータ転送を行うためのホスト・インタフェースを内蔵しています。従来の μ PD77111ファミリは8ビット・データ・バスのみでしたが、 μ PD77115は16ビット・データ・バスをサポートしています。

基本的な機能は μ PD77111ファミリのほかの製品と同じです。3.7.4 ホスト・インタフェースも参照してください。

主な特徴を次に示します。

- ・16ビット・パラレル・ポート

外部デバイスとは、8ビット/16ビット・データ・バスでインタフェース

- ・データ範囲

バイト（8ビット）モード： μ PD77115内部のホスト・インタフェース用レジスタ（HDT）上位8ビットまたは下位8ビットをアドレスで選択。

ワード（16ビット）モード： μ PD77115内部のホスト・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・アクセス可能。

- ・内部データ・バスの接続

XバスおよびYバスに接続

内部は16ビット

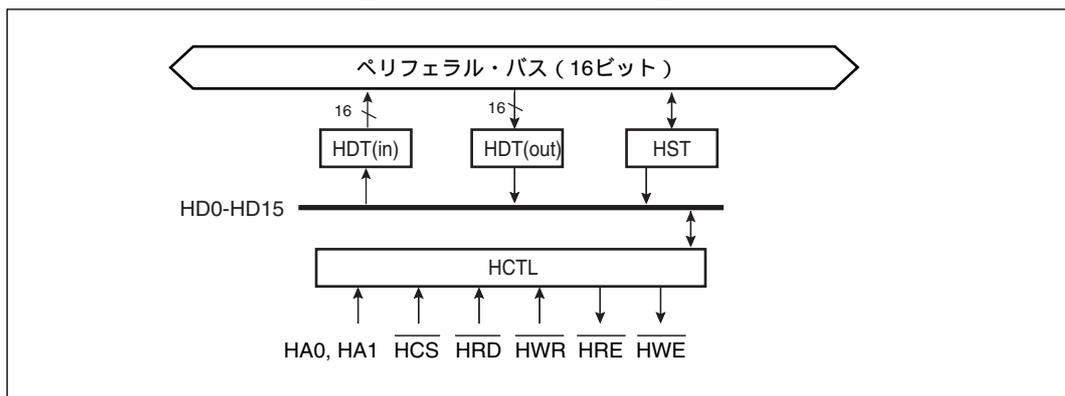
- ・内部ハンドシェイク

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

ホスト・インタフェース制御回路（HCTL）が端子、レジスタを制御します。

ホスト・インタフェースのブロック図を図6-5に示します。

図6-5 ホスト・インタフェース



6.3.1 ホスト・インタフェースの端子

(1) $\overline{\text{HCS}}$ （ホスト・チップ・セレクト入力）

ホスト・インタフェース・セレクト信号の入力端子です。この端子はロウ・アクティブです。ホストCPUがホスト・インタフェースのレジスタをアクセスする場合、アクセス期間中アクティブにします。

(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 CPU がホスト・インタフェースのレジスタをアクセスする場合、アクセス期間中は確定していなければなりません。

(3) $\overline{\text{HRD}}$ (ホスト・リード・ストロブ入力)

ホスト・インタフェースのリード・ストロブ信号入力端子です。この端子はロウ・アクティブです。ホスト CPU がホスト・インタフェースのレジスタをリードする場合にアクティブにします。 $\overline{\text{HWR}}$ 端子と同時にアクティブにしないでください。

(4) $\overline{\text{HWR}}$ (ホスト・ライト・ストロブ入力)

ホスト・インタフェースのライト・ストロブ信号入力端子です。この端子はロウ・アクティブです。ホスト CPU がホスト・インタフェースのレジスタにライトする場合にアクティブにします。 $\overline{\text{HRD}}$ 端子と同時にアクティブにしないでください。

(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。ホスト CPU がホスト・インタフェースのレジスタにアクセスする場合、データの入出力を行います。 $\overline{\text{HCS}}$ 端子がインアクティブの場合、ハイ・インピーダンスになります。

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

HDT に対するリード処理許可信号の出力端子です。この端子はロウ・アクティブです。

CLKOUT の立ち上がりに同期してアクティブになり、 $\overline{\text{HRD}}$ 端子の立ち下がりに同期してインアクティブになります。

HDT をリード可能な (μ PD77115 が HDT にデータをストアした) 場合にアクティブとなり、HDT の上位バイトのデータをリードした場合 (バイト・モード) またはワード・リードした場合にインアクティブになります。HDT の下位バイトのアクセスでは変化しません (バイト・モード)。HST の HREM を 1 とすることで常にインアクティブとなります。システム・リセットによりインアクティブになります。

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

HDT に対するライト処理許可信号の出力端子です。この端子はロウ・アクティブです。

CLKOUT の立ち上がりに同期してアクティブになり、 $\overline{\text{HWR}}$ 端子の立ち下がりに同期してインアクティブになります。

HDT にライト可能な (μ PD77115 が HDT からデータをロードした) 場合にアクティブになり、HDT の上位バイトのデータをライトした場合 (バイト・モード) またはワード・ライトした場合にインアクティブになります。HDT の下位バイトのアクセスでは変化しません (バイト・モード)。HST の HWEM を 1 とすることで常にインアクティブとなります。システム・リセットによりインアクティブになります。

6.3.2 ホスト・インタフェースのレジスタ

(1) ホスト・データ・レジスタ (HDT)

ホスト・データ・レジスタ (HDT) はホスト・インタフェースからデータを入出力するための 16 ビットのレジスタです。HDT の値は、X バスおよび Y バスと入出力できます。

(a) ホスト・データ出力レジスタ (HDT (out))

ホスト・インタフェースから出力するデータを設定する 16 ビット・レジスタです。

HDT に対するストア命令を実行すると、このレジスタにペリフェラル・バスからデータを入力します。外部デバイスがバイト (8 ビット) リードする場合に、HA0 で上位 8 ビットまたは下位 8 ビットを指定します。外部デバイスがワード (16 ビット) リードする場合は HA0 の値は影響しません。HREF が「1」のときに HDT に対するストア命令を実行すると、HSER が「1」になります (ストア・エラー)。

(b) ホスト・データ入力レジスタ (HDT (in))

ホスト・インタフェースから入力するデータを設定する 16 ビット・レジスタです。

HDT に対するロード命令を実行すると、このレジスタのデータをペリフェラル・バスに出力します。外部デバイスがバイト (8 ビット) ライトする場合に、HA0 で上位 8 ビットまたは下位 8 ビットを指定します。外部デバイスがワード (16 ビット) ライトする場合は HA0 の値は影響しません。HWEF が「1」のときに HDT に対するロード命令を実行すると、HLER が「1」になります (ロード・エラー)。

(2) ホスト・インタフェース・ステータス・レジスタ (HST)

ホスト・インタフェース・ステータス・レジスタ (HST) はホスト・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。ホスト CPU とホスト・インタフェース間、およびホスト・インタフェースと μ PD77115 間とのインタフェースの指定や書き込み、読み出しエラー表示を行います。HST の値は、X バスおよび Y バスと入出力できます。バイト (8 ビット) で外部に読み出すときは、HA0 で上位 8 ビットまたは下位 8 ビットを指定します。リセット時の値は、0x0301 です。

表 6 - 5 にビット構成を示します。

表6-5 HSTの機能(1/2)

ビット	名称	リード/ライト (ホストから)	ロード/ストア (DSP から)	機 能
15-12	リザーブ	-	-	リザーブ・ビット ・このビットには値を設定できない。読み出し時は不定。
11	HBM	R	L/S	ホスト・アクセス・ビット・モード ・0: バイト(8ビット)モード ・1: ワード(16ビット)モード
10	HAWE	R	L/S	HDT アクセス・ウエイト許可ビット ・0: ウエイトを使用しない ・1: ウエイトを使用する HREF が 1 のときに μ PD77115 が HDT に書き込もうとした場合にウエイトを発生する。 HWEF が 1 のときに μ PD77115 が HDT を読み出そうとした場合にウエイトを発生する。
9	HREM	R	L/S	ホスト CPU リード・イネーブル・マスク・フラグ ・0: マスクしない(ホスト CPU の HDT に対するリード可否に応じて, \overline{HRE} が変化する) ・1: マスクする(\overline{HRE} はインアクティブとなる)
8	HWEM	R	L/S	ホスト CPU ライト・イネーブル・マスク・フラグ ・0: マスクしない(ホスト CPU の HDT に対するライト可否に応じて, \overline{HWE} が変化する) ・1: マスクする(\overline{HWE} はインアクティブとなる)
7	UF1	R	L/S	ユーザ・フラグ
6	UF0			
5	HRER	R	L/S	ホスト・リード・エラー・フラグ ・0: エラーなし ・1: エラー HREF が 0 のときにホスト CPU が HDT をリードした場合に 1 になる。 一度セットされると μ PD77115 が 0 を書き込むまで変化しない。
4	HWER	R	L/S	ホスト・ライト・エラー・フラグ ・0: エラーなし ・1: エラー HWEF が 0 のときにホスト CPU が HDT にライトした場合に 1 になる。 一度セットされると μ PD77115 が 0 を書き込むまで変化しない。
3	HSER	R	L/S	HDT ストア・エラー・フラグ ・0: エラーなし ・1: エラー HREF が 1 のときに μ PD77115 が HDT にストアした場合に 1 になる。 一度セットされると μ PD77115 が 0 を書き込むまで変化しない。

表6-5 HSTの機能(2/2)

ビット	名称	リード/ライト (ホストから)	ロード/ストア (DSP から)	機 能
2	HLER	R	L/S	HDTロード・エラー・フラグ ・0:エラーなし ・1:エラー HWEFが1のとき μ PD77115がHDTからロードした場合に1になる。一度セットされると, μ PD77115が0を書き込むまで変化しない。
1	HREF	R	L	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 PD77115がHDTにストアした場合1になる。 ホストがHDTの上位バイトをリードした場合0になる。 書き込み(ストア)時は無視。
0	HWEF	R	L	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 PD77115がHDTからロードした場合1になる。 ホストがHDTの上位バイトにライトした場合0になる。 書き込み(ストア)時は無視。

6.3.3 ホスト・インタフェース用レジスタのアドレス

ホスト CPU が、ホスト・インタフェースのレジスタをアクセスする場合、HA1 および HA0 でレジスタを指定します。

表 6 - 6 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のバイト・アドレス・マップ、表 6 - 7 にワード・アドレス・マップを示します。

表 6 - 6 ホスト・インタフェース用レジスタのバイト・アドレス・マップ

$\overline{\text{HCS}}$	$\overline{\text{HRD}}$	$\overline{\text{HWR}}$	HA1	HA0	転送対象のレジスタ/バイト
0	0	0	×	×	禁止
0	0	1	0	0	HDT (出力) 下位 8 ビット
0	0	1	0	1	HDT (出力) 上位 8 ビット
0	0	1	1	0	HST 下位 8 ビット
0	0	1	1	1	HST 上位 8 ビット
0	1	0	0	0	HDT (入力) 下位 8 ビット
0	1	0	0	1	HDT (入力) 上位 8 ビット
0	1	0	1	×	禁止
0	1	1	×	×	対象なし
1	×	×	×	×	対象なし

表 6 - 7 ホスト・インタフェース用レジスタのワード・アドレス・マップ

$\overline{\text{HCS}}$	$\overline{\text{HRD}}$	$\overline{\text{HWR}}$	HA1	HA0	転送対象のレジスタ/ワード
0	0	0	×	×	禁止
0	0	1	0	×	HDT (出力) 16 ビット
0	0	1	1	×	HST 16 ビット
0	1	0	0	×	HDT (入力) 16 ビット
0	1	0	1	×	禁止
0	1	1	×	×	対象なし
1	×	×	×	×	対象なし

6.3.4 ホスト・インタフェースのタイミング

(1) ホスト・リード・タイミング

HDT に対するストア命令を実行すると、ホスト・インタフェース・ステータス・レジスタ HST の HREF が 1 になり、HST の HREM が 0 の場合には $\overline{\text{HRE}}$ がアクティブになります。

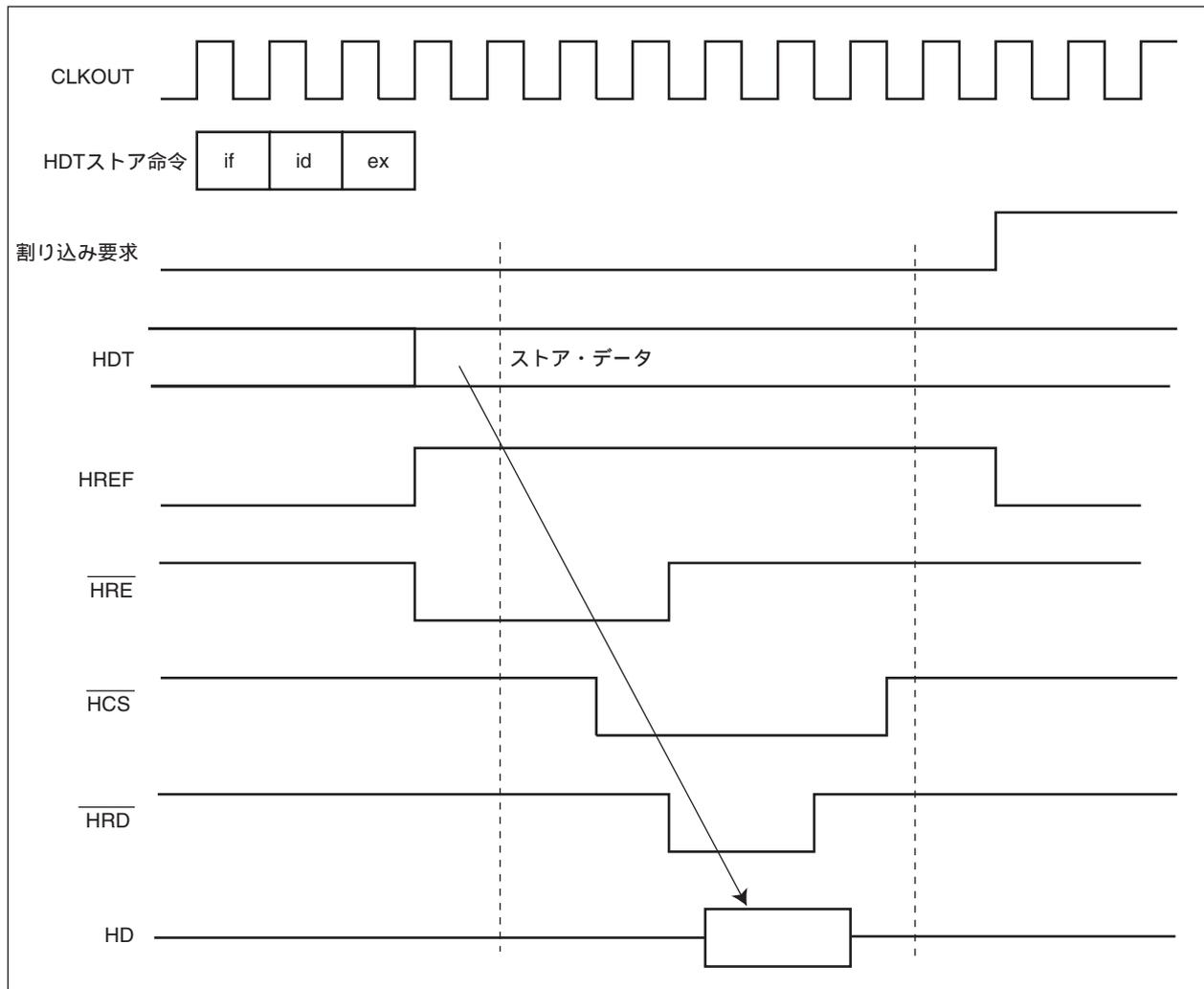
HST の HAWE が 1 で、HREF が 1 のときに HDT にストア命令を実行すると、ウエイトが発生します。

バイト・モード時、ホスト CPU が HDT の上位バイトをリードすると、HREF のリセット要求を出力し $\overline{\text{HRE}}$ がインアクティブになります (HREF はシステム・クロックに同期したリセット要求によりリセットします)。

ワード・モード時、ホスト CPU が HDT のワードをリードすると、HREF のリセット要求を出力し $\overline{\text{HRE}}$ がインアクティブになります (HREF はシステム・クロックに同期したリセット要求によりリセットします)。また、 μ PD77115 内部に対し HO 割り込み要求を出力し、HST の HAWE が 1 でウエイト状態のときはウエイトを解除します。

ホスト・リード・タイミングを図 6-6 に示します。

図 6-6 ホスト・リード・タイミング



(2) ホスト・ライト・タイミング

バイト・モード時, ホスト CPU が HDT の上位バイトにライトすると, ホスト・インタフェース・ステータス・レジスタ HST の HWEF のリセット要求を出力し, $\overline{\text{HWE}}$ がインアクティブになります (HWEF は, システム・クロックに同期したリセット要求によってリセットします)。

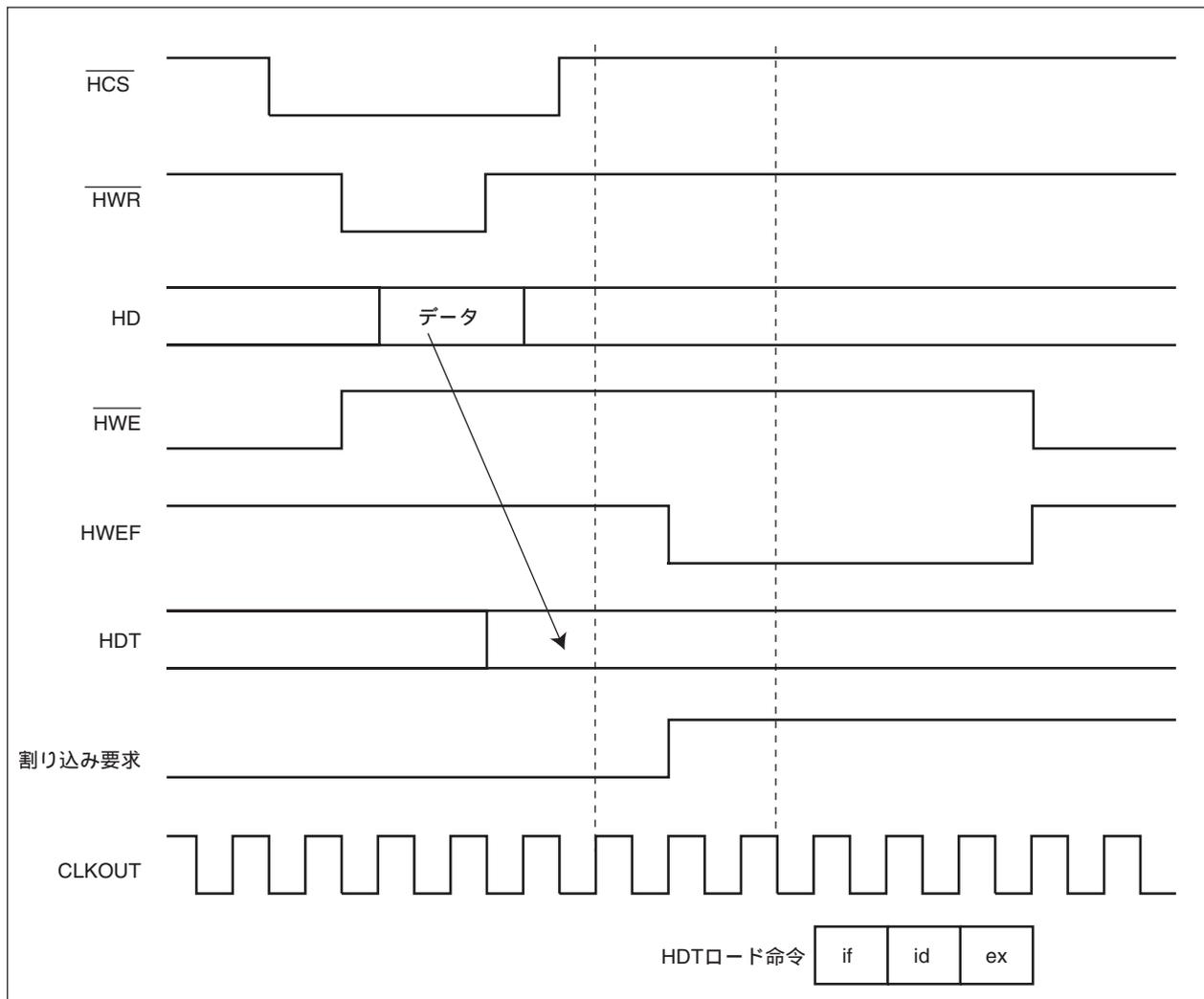
ワード・モード時, ホスト CPU が HDT にライトすると, ホスト・インタフェース・ステータス・レジスタ HST の HWEF のリセット要求を出力し, $\overline{\text{HWE}}$ がインアクティブになります (HWEF は, システム・クロックに同期したリセット要求によってリセットします)。

また, μ PD77115 内部に対し HI 割り込み要求を出力し, HST の HAWE が 1 でウェイト状態のときはウェイトを解除します。

HDT へのロード命令を実行すると HWEF が 1 になり, HST の HWEM が 0 ならば $\overline{\text{HWE}}$ がアクティブになります。HST の HAWE が 1 で HWEF が 1 のときに HDT に対するロード命令を実行すると, ウェイトが発生します。

ホスト・ライト・タイミングを図 6 - 7 に示します。

図 6 - 7 ホスト・ライト・タイミング



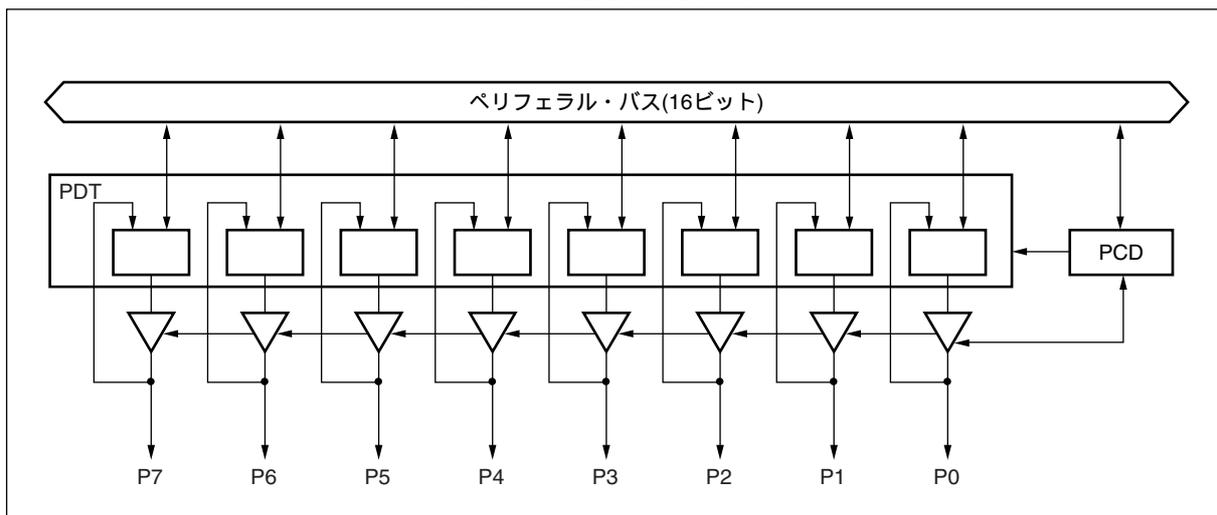
6.4 汎用入出力ポート

μ PD77115にはシステム・リセット直後のブート・モード指定やPLL 通倍，外部デバイスの制御などのための汎用入出力ポートがあります。従来の μ PD77111ファミリは4ビット・ポートでしたが， μ PD77115は8ビット・ポートをサポートしています。主な特徴を以下に示します。

- ・ 8 端子
- ・ 入出力指定
 - 8 本独立に入出力を選択
- ・ 内部データ・バスの接続
 - XバスおよびYバスに接続
- ・ コマンド制御
 - 出力端子のセット/リセットを独立に操作(たとえば，P7-P1を変化させずにP0のみモード/出力データを変更できます)

汎用入出力ポートのブロック図を図6-8に示します。

図6-8 汎用入出力ポート



6.4.1 汎用入出力ポートの端子

(1) P7-P0 (ポート入出力)

汎用ポート入出力端子です。

出力端子は，CLKOUTの立ち上がりに同期して出力が変化します。

入力端子を，CLKOUTの立ち上がりに同期してサンプリングします。

6.4.2 汎用入出力ポートのレジスタ

(1) ポート・データ・レジスタ (PDT)

汎用入出力ポートから入力されたデータを読み出す、または汎用入出力ポートへ出力するデータを設定する、16ビットのレジスタです。

μ PD77115がPDTに対しロード命令を実行すると、PDTのデータをペリフェラル・バスに出力します。

Pi端子を入力モードとした場合、入力端子Piのハイ・レベル入力によりPDTのビットiが1となり、入力端子Piへのロウ・レベル入力によりPDTのビットiが0となります。

Pj端子が出力端子の場合はビットjの値は不定です。

注意 出力モードとしたPDTのロードでは出力端子の状態をロードします。

μ PD77115がPDTに対しストア命令を実行するとペリフェラル・バスからPDTにデータを入力します。

Pj端子を出力モードとした場合、PDTのビットjが1のときに出力端子Pjがハイ・レベルを出力し、PDTのビットjが0のときに出力端子Pjがロウ・レベルを出力します。

Pi端子が入力端子の場合は、ビットiの値は無効です。

PDTの値は、XバスおよびYバスと入出力できます。

(2) ポート・コマンド・レジスタ (PCD)

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。ポートの入出力モードの変更はPCDレジスタに対するデータ設定に対し遅延時間があります。

PCDの値は、XバスおよびYバスと入出力できます。リセット時の値は0です。PCDの各ビット機能を表6-8に示します。

表 6 - 8 PCD の機能

ビット	名称	機 能
15	BE	書き込み時： ビット操作処理実行許可ビット。 B2-B0, PSR の内容によるビット操作命令実行の可否を指定する。 1：設定許可，0：設定禁止 読み出し時： 未使用（0 を出力する）
14	PSR	書き込み時： セット/リセット・ビット指定ビット。 B2-B0 で指定した番号のポートをセットするかリセットするかを指定する。 1：ビット・セット，0：ビット・リセット 読み出し時： 未使用（0 を出力する）
13	ME	書き込み時： モード設定命令実行許可ビット。 M7-M0, IO の内容によるモード設定実行の可否を指定する。 1：設定許可，0：設定禁止 読み出し時： 未使用（0 を出力する）
12	IO	書き込み時： 設定モード指定ビット。 M7-M0 で選択した番号のポートを入力とするか，出力とするかを指定する。 1：出力ポート，0：入力ポート 読み出し時： 未使用（0 を出力する）
11	-	書き込み時： 将来の拡張用リザーブ・ビット。 書き込んだ値は，ビット操作，モード変更処理に影響を与えない。 読み出し時： 未使用（0 を出力する）
10-8	B2-B0	書き込み時： ビット操作ポート指定ビット。 ビット操作を実施する出力ポートを B2-B0 で表す 2 進数で指定する。 B2 B1 B0：選択ポート 000：P0 を選択 001：P1 を選択 010：P2 を選択 011：P3 を選択 100：P4 を選択 101：P5 を選択 110：P6 を選択 111：P7 を選択
7-0	M7-M0	書き込み時： モード設定ポート指定ビット。 入出力モード設定を行うポートを 1 ビットごとに指定する。 M7：1 で P7 を選択，0 で非選択 M6：1 で P6 を選択，0 で非選択 M5：1 で P5 を選択，0 で非選択 M4：1 で P4 を選択，0 で非選択 M3：1 で P3 を選択，0 で非選択 M2：1 で P2 を選択，0 で非選択 M1：1 で P1 を選択，0 で非選択 M0：1 で P0 を選択，0 で非選択 読み出し時： 設定モード表示ビット。 入出力ポート（P7-P0）の入出力モード設定状態（1：出力，0：入力モード）

6.4.3 汎用入出力ポートのタイミング

(1) 入力端子のタイミング

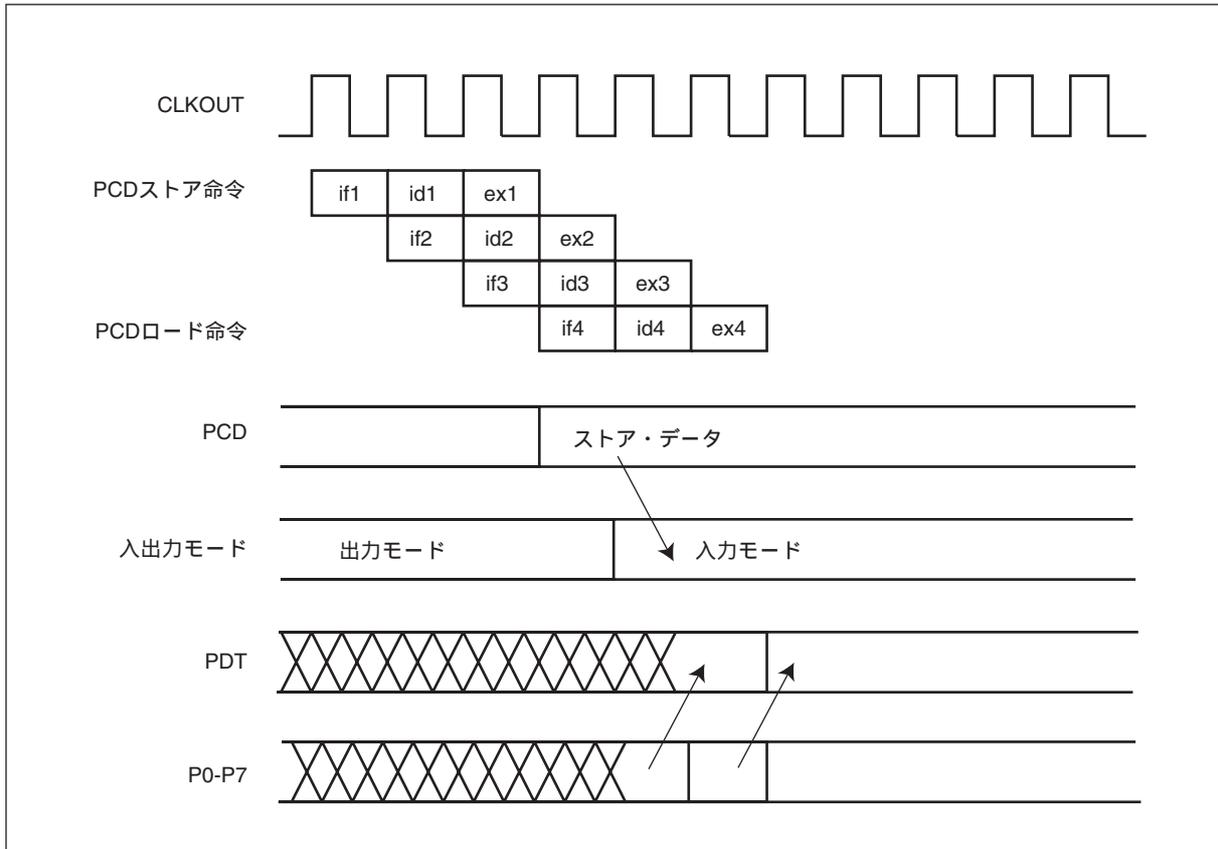
入力端子は、CLKOUTの立ち下がりに同期してサンプリングします。

PDTからのロード命令で読み込めるのは、同命令のデコード・サイクル間の端子状態です。

Pi端子を出力端子から入力端子に変更する命令を実行してから、Pi端子への入力データを読み込めるのは3命令後以降です。

データ入力タイミングを図6-9に示します。

図6-9 汎用ポートのデータ入力タイミング



(2) 出力端子のタイミング

出力端子は、CLKOUT の立ち上がり同期して出力が変化します。

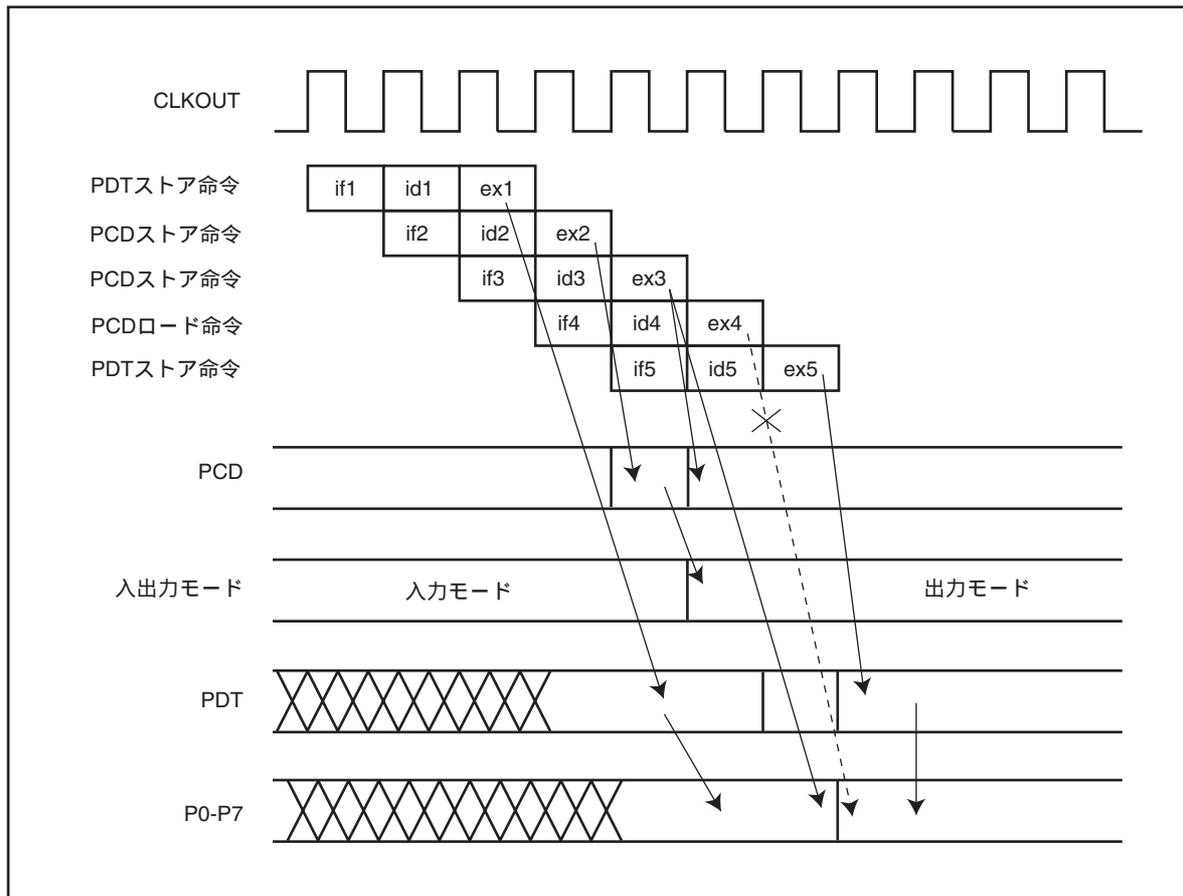
PDT に対するストア命令を実行すると、実行サイクルの直後のサイクルで P0-P7 が変化します。PCD に対するストア命令でビット操作を実行すると、実行サイクルの 2 命令後のサイクルで P0-P7 が変化します。

PCD に対するストア命令で Pj 端子を出力端子に設定すると同時に、Pj 端子のビット操作を同時に指定できます。

データ出力タイミングを図 6 - 10 に示します。

- 注意**
- ・ PCD に対するストア命令でビット操作を実行した直後の命令で、PDT に対するストア命令は禁止します。
 - ・ PDT に対するストア命令の同時記述命令がウエイトを発生させた場合、PCD に対するストア命令実行後、PDT に対するストア命令の実行前に割り込みベクタに分岐した場合に出力ポートの動作が変わります。
 - ・ 図 6 - 10 では、PDT ストア命令の直前の PCD ストア命令による出力ポートの変更は、次の PDT ストア命令による出力ポート変更により無効となりますが、PDT ストア命令の実行をウエイトした場合や PCD ストア命令実行後に割り込みベクタに分岐した場合、PCD ストア命令による出力ポート設定が有効となり、ウエイト解除もしくは割り込み復帰後の PDT ストア命令で出力ポートが変更されます。

図 6 - 10 汎用ポートのデータ出力タイミング



6.5 SDカード・インタフェース

μ PD77115は、SD (Secure Digital) メモリ・カードとデータ転送を行うためのSDカード・インタフェースを内蔵しています。

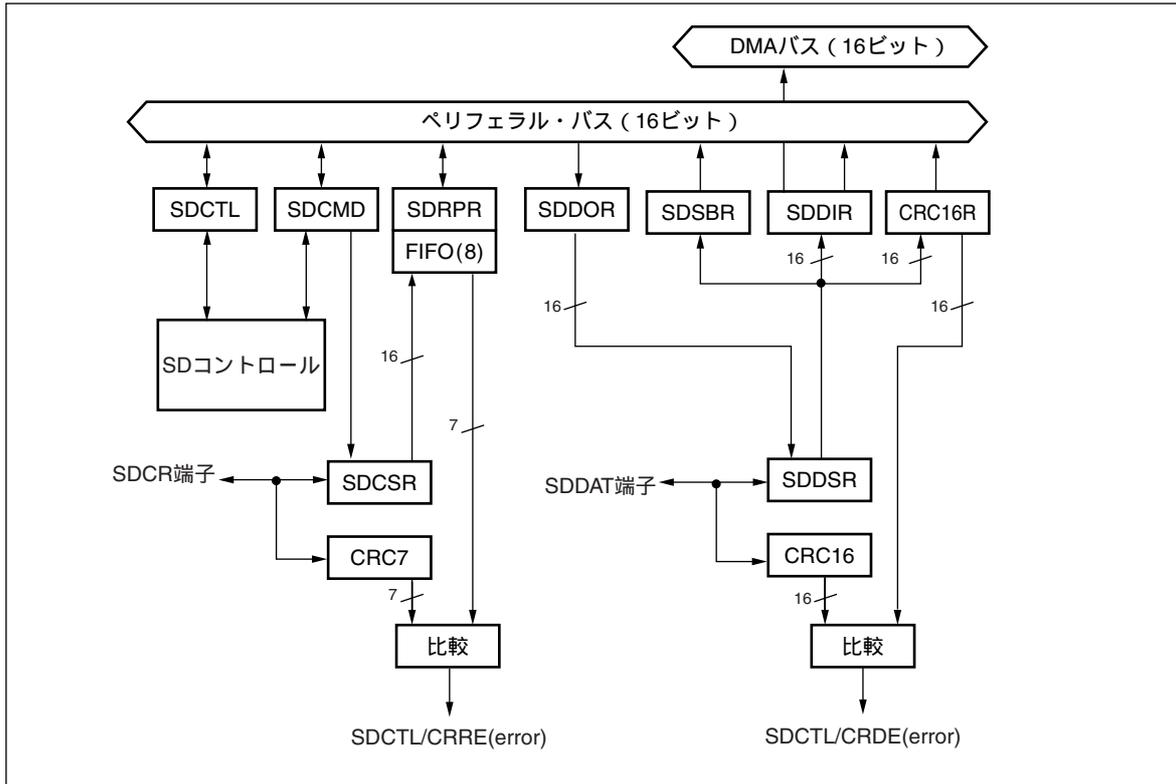
SD メモリ・カード・インタフェース仕様は、SD Memory Card Specifications Part 1 PHYSICAL LAYER SPECIFICATION Version 1.0 March 2000 に基づいて設計されています。主な特徴を次に示します。

- ・3 端子
 - クロック (SDCLK)
 - コマンド/レスポンス (SDCR)
 - データ (SDDAT)
- ・SDバスのみに対応
- ・カード挿抜、ライト・プロテクト信号は μ PD77115上ではサポートしていません。
- ・内部データ・バスの接続
 - XバスおよびYバスに接続、データ入力レジスタ (SDDIR) のみDMAバスに接続
- ・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6-11に示します。

注意 基本的にSDカード・インタフェースを介したSDカードのセキュア・エリアのリード/ライトは μ PD77115に内蔵される非公開の認証プログラム、暗号プログラムとNECから提供されるそれらプログラムを使用するためのインタフェース・ソフトウェアによって行います。これはSDの認証システム、暗号システムを保護するためにこのような構成になっています。また、SDカードのユーザ・エリアのリード/ライトも同様にインタフェース・ソフトウェアによって行われます。ユーザにはインタフェース・ソフトウェアの仕様が提示されますが、認証および暗号プログラム自体を直接見ることはできません。また、この認証および暗号プログラムは4C Entityと契約されたお客様のみカスタム製品として非公開エリアに格納いたします。したがって、標準品として販売されている μ PD77115には内蔵されていません。

図6-11 SDカード・インタフェース構成図



6.5.1 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25MHzです(ただし、この周波数を越えないようにSDカード・インタフェース・コントロール・レジスタ(SDCTL)のDIVビットの設定を行う必要があります)。

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。SDカード・インタフェース・コントロール・レジスタ(SDCTL)のSDSTPビットを“1”に設定することによりSDCLK出力をロウ・レベルとすることができます。

SDCLKのハイ・レベル幅はデューティ50%ではなく、 μ PD77115の内部クロック1クロック分の長さになっています。

(2) SDCR (SDカード・コマンド出力/レスポンス入力)

SDCR端子は、コマンド出力とレスポンス入力を行う端子でSDカードCMD端子に接続します。

μ PD77115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、端子状態はハイ・インピーダンスとなるため、デバイス外部のボード上のプルアップにより、 V_{DD} レベル(“1”)にする必要があります。

(3) SDDAT (SD カード・データ入出力)

ライト・データ：出力

リード・データ：入力

SDDAT 端子は、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 PD77115 から SD カードに対してリード・コマンド送信後、データをリードします。ライト・コマンド送信後、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、端子状態はハイ・インピーダンスとなるため、チップ外部のボード上のプルアップにより、V_{DD} レベル (“1”) にする必要があります。

6.5.2 SD カード・インタフェースのレジスタ

(1) SD カード・データ出力レジスタ (SDDOR)

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。SDCTL レジスタの SDS ビットが「0」のときに SDDOR 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)。SDDOR は、ペリフェラル・バスへデータをロードすることはできません。

(2) SD カード・データ入力レジスタ (SDDIR)

SD カードから入力されたデータを保持する 16 ビットのレジスタです。通常のリードのほか、ペリフェラル・バッファからのリード要求により、SDDIR から DMA バスにデータを出力します。ペリフェラル・バスのロードとペリフェラル・バッファのリードが競合した場合、 μ PD77115 にウエイトを挿入し、ペリフェラル・バッファのリードを優先します。SDCTL レジスタの SDL ビットが「0」のときに SDDIR 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 (ストア・エラー)。SDDIR は、ペリフェラル・バスからデータをストアすることはできません。

(3) SD カード・インタフェース・レスポンス・レジスタ (SDRPR)

SD カード・インタフェース・レスポンス・レジスタ (SDRPR) は 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 FIFO (8×16 ビット) で構成しています。

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、SDRPR7-SDRPR3 には意味のないデータが入っています。

SDRPR から 3 回 (48 ビット・レスポンス)、8 回 (136 ビット・レスポンス) のロードを繰り返すことで SD カードのレスポンス・データをリードできます。

なお、136 ビット・レスポンスの場合、ビット 135-128 の先頭 8 ビットはこのレジスタに格納されず破棄されます。これらのビットはスタート・ビットおよびリザーブ・ビットであり、破棄しても問題ありません。

(4) SD カード・インタフェース CRC ステータス・ビジィ・レジスタ (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 (SDSBR) は SD カードヘ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られる 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 ビットで構成しており、ビット 3 がビジィ・フラグ、ビット 2-0 が CRC ステータスの機能になっています。SDSBR からロードできますがストアはできません。

ビジィ・フラグが“1”のとき、ビジィ状態であることを示し、“0”のとき、ビジィ状態ではないことを示します。

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 カードヘータを送信したあとは、かならず 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。

(5) SD カード・コマンド・レジスタ (SDCMD)

SD カード・コマンド・レジスタ (SDCMD) は 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・レジスタで構成されています。コマンド解釈+コマンド・インデクス出力用 (SDCMD_IDX) とコマンド・アークユメント・ハイ側用 (SDCMD_AGH) とコマンド・アークユメント・ロウ側用 (SDCMD_AGL) のレジスタを別々に持っています。SDCMD はロード/ストアができます。

(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・データを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。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 6-9 に示します。

表 6 - 9 SDCMD_IDX の機能

ビット	名称	ロード/ ストア	機 能
15	リザーブ	-	リザーブ・ビット ・このビットには値を設定できない。読み出し時は0。
14-11	DBL	L/S	データ・ブロックの範囲を設定するビット 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
10	RES	L/S	SD カードからのレスポンスの有無を設定するビット 0 : レスポンスなし (ノー・レスポンスのオペレーションとして扱う) 1 : レスポンスあり
9-8	RW	L/S	データのリード・ライトおよびデータ転送ストップを設定するビット 00 : リード・ライトなし (ノー・データのオペレーションとして扱う) 01 : リードあり (リード・オペレーション) 10 : ライトあり (ライト・オペレーション) 11 : リード・ライトなし
7	リザーブ	L/S	リザーブ・ビット ・このビットには値を設定できない。読み出し時は不定。
6	BSL	L/S	受信したレスポンスのビット数を設定するビット 0 : 48 ビット 1 : 136 ビット
5-0	IDX	L/S	SD カードへ送信するコマンド・インデッ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。

(b) SD カード・コマンド・アーギュメント・ハイ側レジスタ (SDCMD_AGH)

SD カードへ出力するコマンド・アーギュメントのハイ側を設定する 16 ビットのレジスタです。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・アーギュメントのロウ側を設定する 16 ビットのレジスタです。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。

(6) SD カード・インタフェース・コントロール・レジスタ (SDCTL)

SD カード・インタフェース・コントロール・コマンド・レジスタ (SDCTL) は 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・レジスタです。リセット初期値は、0x0202 です。SDCTL の各ビットの機能を表 6 - 10 に示します。

表6-10 SDCTLの機能(1/2)

ビット	名称	ロード/ ストア	機 能
15	ALLE	L/S	SD カード I/F 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 ALLE がセットされる。 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 をストアする)
14	CEN	L/S	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
13-10	DIV	L/S	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: 1/2 分周 0010: 1/3 分周 0011: 1/4 分周 0100: 1/5 分周 0101: 1/7 分周 0110: 1/8 分周 0111: 1/9 分周 1000: 1/10 分周 1001: 1/12 分周 1010: 1/16 分周 1011: 1/32 分周 1100: 1/64 分周 1101: 1/128 分周 1110: 1/256 分周 1111: 1/512 分周
9	SDSTP	L/S	SD カード I/F ストップ・ビット。 SD カード I/F 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 SDSTP ビットに 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, SD カード I/F 内は状態を保持する。 0: SD カード I/F 動作 1: SD カード I/F 停止 (デフォルト)
8	SDRST	L/S	SD カード I/F リセット許可ビット。 SDRST ビットを 1 に設定することで, SD カード I/F 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
7	リザーブ	-	リザーブ・ビット。 このビットには値を設定できない。 読み出し (ロード) 時は不定。
6	CRRE	L/S	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 μPD77115 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 (0 にクリアするには, 0 をストアする)
5	CRDE	L/S	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 μPD77115 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 (0 にクリアするには, 0 をストアする)

表 6 - 10 SDCTL の機能 (2/2)

ビット	名称	ロード/ ストア	機 能
4	SDSE	L/S	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エラーと判断され、1をセットする。 0: エラーなし 1: エラーあり (0 にクリアするには、0 をストアする)
3	SDLE	L/S	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エラーと判断され、1をセットする。 0: エラーなし (デフォルト) 1: エラーあり (0 にクリアするには、0 をストアする)
2	SDRL	L	SDRPR ロード許可フラグ。 SDRPR は、16 ビット×8 のレジスタで構成している。 0: SDRPR7- SDRPR0 にデータが入っていないことを示す (デフォルト) 1: SDRPR7- SDRPR0 にデータが入っていることを示す。
1	SDS	L	SDDR ストア許可フラグ。 0: SDDOR にデータが入っていることを示す。 1: SDDOR にデータが入っていないことを示す (デフォルト)
0	SDL	L	SDDR ロード許可フラグ。 0: SDDIR にデータが入っていないことを示す (デフォルト) 1: SDDIR にデータが入っていることを示す。

6. 5. 3 CRC (Cyclic Redundancy Codes) 回路

CRC 回路は、μPD77115 と SD カード間のデータ・エラーが生じたときにエラーを検出するものです。

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、μPD77115 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・データに付加した CRC データと、μPD77115 内でレスポンスやリード・データの CRC を計算して付加した CRC データとを比較し、不一致であればエラーとしてフラグを立てます。

6. 5. 4 データ・フォーマット

SDCR 端子から入出力するコマンド、レスポンス、SDDAT 端子から入出力するデータのフォーマットに関して次に示します。

(1) コマンド・フォーマット

ビット・ポジション	47	46	45-40	39-8	7-1	0
Width (ビット)	1	1	6	32	7	1
Value	'0'	'1'	X	X	X	'1'
Description	start bit	transmission bit	command index	argument	CRC7	end bit

ビット 45-40 の command index value は、SDCMD_IDX レジスタのビット 5-0 に設定した値が代入され、ビット 39-8 の argument value には、SDCMD_AGH、SDCMD_AGL レジスタで設定した値が代入されます。また、ビット 7-1 の CRC7 には、ビット 47-8 の CRC 計算結果が代入されます。

(2) レスポンス・フォーマット

(a) R1 : Normal Response (48 ビット・レスポンス)

ビット・ポジション	47	46	45-40	39-8	7-1	0
Width (ビット)	1	1	6	32	7	1
Value	'0'	'0'	X	X	X	'1'
Description	start bit	transmission bit	command index	card status	CRC7	end bit

ビット 45-40 の command index value およびビット 39-8 の card status value は、SD カードから出力した値が代入され、SDRPR レジスタに保持します。また、ビット 7-1 の CRC7 には、SD カード側でビット 47-8 の CRC 計算結果を付加します。

(b) R2 : CID, CSD register (136 ビット・レスポンス)

ビット・ポジション	135	134	133-128	127-1	0
Width (ビット)	1	1	6	127	1
Value	'0'	'0'	'111111'	X	'1'
Description	start bit	transmission bit	reserved	CID or CSD register includes internal CRC7	end bit

ビット 127-1 の CID/CSD register value は、SD カードから出力した値が代入され、SDRPR レジスタに保持します。また、ビット 7-1 の CRC7 には、SD カード側でビット 127-8 の CRC 計算結果を付加します。

(c) R3 : OCR register (48 ビット・レスポンス)

ビット・ポジション	47	46	45-40	39-8	7-1	0
Width (ビット)	1	1	6	32	7	1
Value	'0'	'0'	'111111'	X	'111111'	'1'
Description	start bit	transmission bit	reserved	OCR register	reserved	end bit

ビット 39-8 の OCR register value は、SD カードから出力した値が代入され、SDRPR レジスタに保持します。

(d) R6 : RCA response (48 ビット・レスポンス)

ビット・ポジション	47	46	45-40	39-8	7-1	0
Width (ビット)	1	1	6	32	7	1
Value	'0'	'0'	X	X	'111111'	'1'
Description	start bit	transmission bit	command index '000011'	39-24 : RCA 23-8 : card states	CRC7	end bit

ビット 45-40 の command index value およびビット 39-8 の RCA, card status value は、SD カードから出力した値が代入され、SDRPR レジスタに保持します。また、ビット 7-1 の CRC7 には、SD カード側でビット 47-8 の CRC 計算結果を付加します。

6.6 ペリフェラル・バッファ

PBUは、SDカード・インタフェース内のレジスタから内部データRAMへデータ転送を行うペリフェラル・バッファ・ユニットです。

この機能はSDアクセス用のインタフェース・ソフトウェアの中で扱われるため、ユーザ側で使用することは原則としてありません。

主な特徴を次に示します。

- ・SDカード・インタフェースのSDDIRからXデータRAMまたはYデータRAMへのDMA転送を行うことが可能です。
- ・DMA転送は専用バスを用いて行われるためデータRAMのバンクが競合しないかぎり、プログラムの実行を妨げません。
- ・SDカード・インタフェースとPBUとのインタフェースは割り込みで行います。
- ・DMA転送完了の通知は割り込みおよびフラグにより行います。
- ・一度に転送可能なデータはMAX: 4Kワードで0x5000-0x5FFFにマッピングされたRAMのバンクとします。
- ・バッファリングは巡回バッファではなく、レジスタで設定したバッファリング・サイズにデータが転送完了した時点でバッファリングは終了し、PBUから割り込みを発生します。
- ・PBUのアクセスとプログラムによるアクセスがデータRAMの同一バンクで競合した場合、またはSDカード・インタフェースのSDDIRレジスタへのロードで競合した場合、PBUからのアクセスを優先し、プログラムはPBUのアクセスが完了するまでウェイトします。

6.6.1 ペリフェラル・バッファのレジスタ

(1) ペリフェラル・バッファ・スタート・アドレス・レジスタ (PSAR)

ペリフェラル・バッファ・スタート・アドレス・レジスタ (PSAR) は、転送先であるデータRAMの先頭アドレスを設定するレジスタです。設定するアドレス幅は12ビット(4Kワード)で、0x000-0xFFFFの範囲で行います。0x000を設定するとデータRAMの0x5000からバッファリングを開始し、0x002を設定するとデータRAMの0x5002からバッファリングを開始します。PSARの初期値は0x0000で、PSARの値はXバスおよびYバスと入出力できます。ビット15-12にストアしたデータは無効で、ロード時は0をバスに出力します。

(2) ペリフェラル・バッファ・サイズ・レジスタ (PSR)

ペリフェラル・バッファ・サイズ・レジスタ (PSR) は、バッファリングのワード・サイズを設定するレジスタです。設定するサイズ範囲は13ビットで、0x0001-0x1000の範囲で行います。PSRの初期値は0x0000で、PSRの値はXバスおよびYバスと入出力できます。ビット15-13にストアしたデータは無効で、ロード時は0をバスに出力します。

(3) ペリフェラル・バッファ・サイズ・カウント・レジスタ (PSCR)

ペリフェラル・バッファ・サイズ・カウント・レジスタ (PSCR) は、サイズのダウンカウント値を保持する 13 ビット・レジスタです。PCR の EN ビットを 1 にすると、PSR で設定したサイズ値を取り込み、SD カード・インタフェースからの割り込みを受け付けてデクリメントを繰り返します。PSCR が 0x0001 になった時点でμPD77115 に対して割り込み要求を出力します。PSCR の初期値は 0x0000 で、X バスおよび Y バスから入出力できません。

(4) ペリフェラル・バッファ・アドレス・ポインタ・レジスタ (PPR)

ペリフェラル・バッファ・アドレス・ポインタ・レジスタ (PPR) は、実際にデータ RAM へアクセスするアドレスを保持するレジスタです。PSAR で設定した先頭アドレスを保持し、SD カード・インタフェースからの割り込みを受け付けてインクリメント動作を行います。次にアクセスするアドレスを保持しているため、バッファリングのアクセス状況をμPD77115 から確認できます。PPR の初期値は 0x0000 で、PPR の値は X バスおよび Y バスと出力でき、値を設定することはできません。ロード時のビット 15-12 は 0 をバスに出力します。

(5) ペリフェラル・バッファ・コントロール・レジスタ (PCR)

ペリフェラル・バッファ・コントロール・レジスタ (PCR) は、バッファリングのコントロールとステータス認識を行うレジスタです。PCR の初期値は 0x0000 で、PCR の値は X バスおよび Y バスと入出力できます。ビット 15-3 にストアしたデータは無効で、ロード時は 0 をバスに出力します。PCR の各ビットの機能を表 6 - 11 に示します。

表 6 - 11 PCR の機能

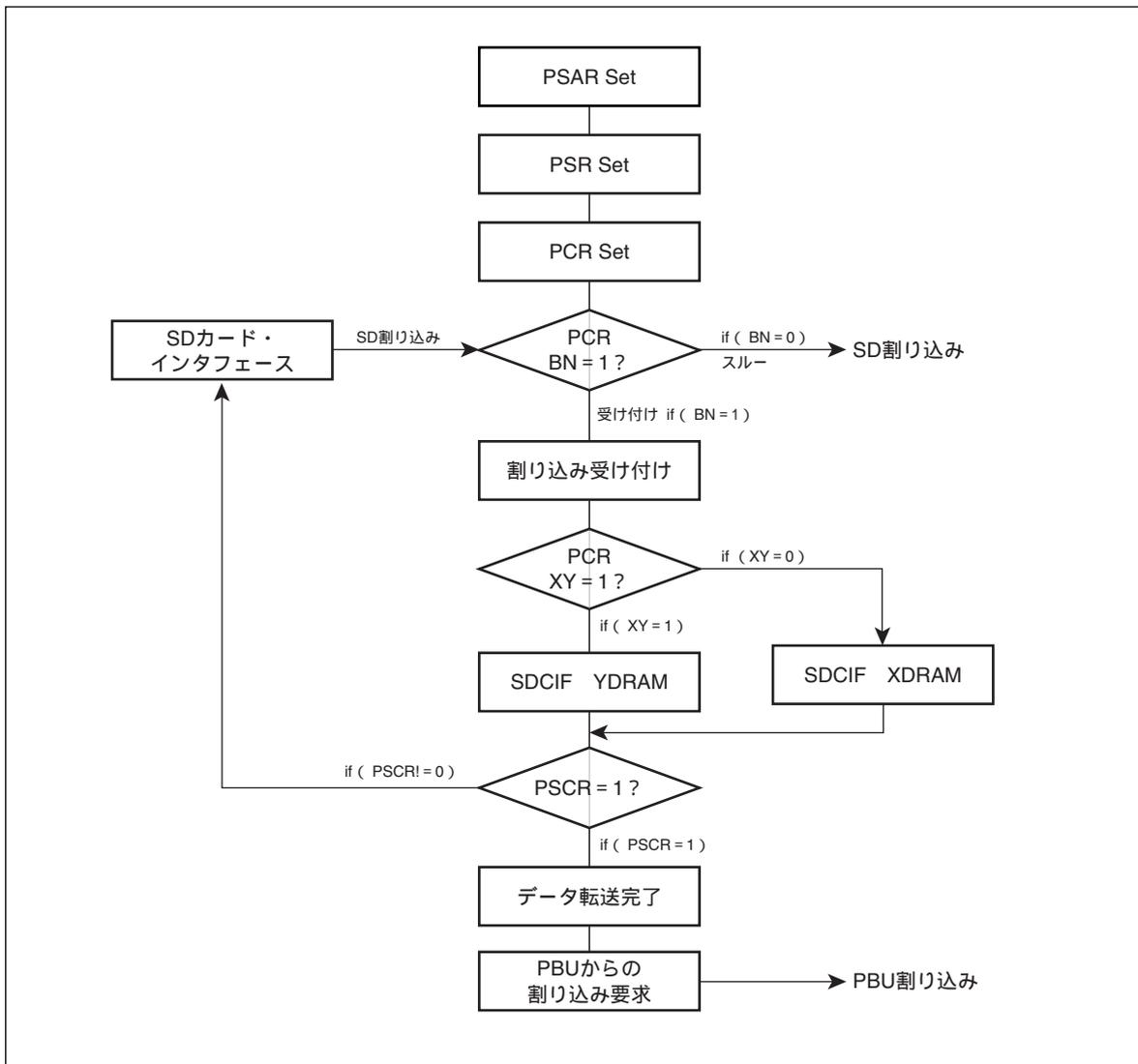
ビット	名称	ロード / ストア	機能
15-3	リザーブ	-	リザーブ・ビット。 ・このビットには値を設定できない。読み出し時は不定。
2	ERR	L/S	PCR ストア・エラー・フラグ。 バッファリング動作中に PCR ヘストア動作が行われた場合、“1” がセットされる。 0: エラーなし (デフォルト) 1: エラーあり (クリアするには、バッファリング停止中に 0 をストアする)
1	EN	L/S	バッファリング実行イネーブル。 バッファリングを開始するためのビット。1 を設定することにより、PBU が動作を開始し、SD カード・インタフェースからの割り込みをトリガにして DMA 転送を行う。 0: DMA 転送停止状態 (デフォルト) 1: DMA 転送状態 (転送終了後、自動的に EN は 0 にクリアされる)
0	XY	L/S	メモリ選択ビット。 バッファ・メモリを X メモリか Y メモリを選択するビット。 0: X メモリ (デフォルト) 1: Y メモリ

6.6.2 ペリフェラル・バッファ動作フロー

- (1) PSAR レジスタにスタート・アドレスを設定します。
- (2) PSR レジスタにバッファ・サイズを設定します。
- (3) PCR レジスタの EN ビットに 1 を設定します。
- (4) SD カードから割り込み要求があった場合,SD カード・インタフェースの SDDIR レジスタのデータを,PCR の XY ビットで設定されたメモリへ転送します。
- (5) PPR レジスタが 0x0001 になるまで,割り込み要求と転送を繰り返し,PPR レジスタが 0x0001 になった時点で転送を完了し,PBU 割り込み要求を出力します。

PBU 動作フロー図を図 6-12 に示します。

図 6-12 PBU 動作フロー図



6.7 タイマ

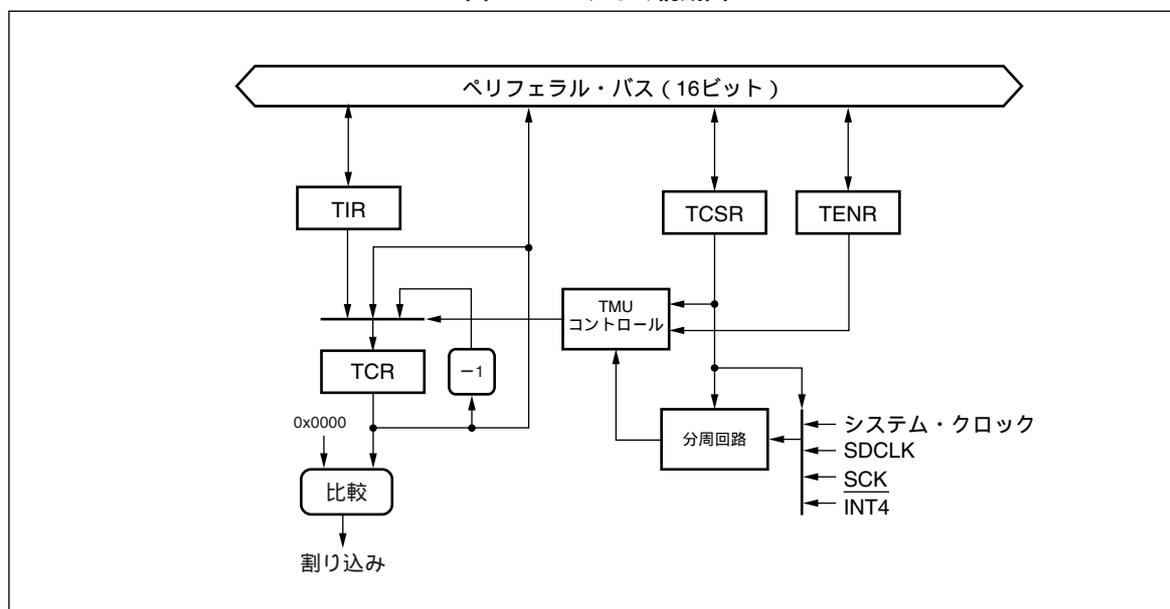
タイマは、16ビットのカウントを行うタイマで、カウントするクロックは、4種類あり、分周比は8通りを選択できます。

主な特徴を次に示します。

- ・タイマ・クロックは、システム・クロック、SDカード・クロック、シリアル・インタフェース・クロック、 $\overline{\text{INT4}}$ 端子からの割り込みの4種類から選択可能です。
- ・分周比は、1, 2, 4, 8, 16, 32, 64, 128の8通りから選択可能です。
- ・タイマ・カウントは、1回またはリピート動作を選択可能です。
- ・タイマとμPD77115のインタフェースは割り込みで行います。

タイマ構成図を図6-13に示します。

図6-13 タイマ構成図



6.7.1 タイマのレジスタ

(1) タイマ初期値レジスタ (TIR)

タイマ初期値レジスタ (TIR) は、カウント初期値 (周期) を設定する 16 ビット・レジスタです。タイマ動作では設定した初期値は変わりません。実際のカウント回数は、TIR に設定した初期値 + 1 になります。たとえば、TIR : 0x0008 であった場合、カウント回数は 9 回になります。TIR の初期値は 0xFFFF で、X バスおよび Y バスと入出力できます。

(2) タイマ・カウント・レジスタ (TCR)

タイマ・カウント・レジスタ (TCR) は、ダウンカウント値を保持する 16 ビット・レジスタです。TENR の TEN ビットを 1 にすると、TIR で設定したカウント初期値を取り込み、カウントダウンを繰り返します。TCR が 0x0000 になった時点で μPD77115 に対して割り込み要求を出力します。TCR の初期値は 0xFFFF で、X バスおよび Y バスと入出力できます。

(3) タイマ・コントロール・ステータス・レジスタ (TCSR)

タイマ・コントロール・ステータス・レジスタ (TCSR) は、タイマの動作設定、ステータスを保持する 16 ビット・レジスタです。TCSR の初期値は 0x0000 で、X バスおよび Y バスと入出力できます。タイマ動作中はバスからのストアはできません。ストアした場合、TCSR はアップデートされずにタイマ動作には影響しませんが、エラー・フラグを立てます。TCSR の各ビットの機能を表 6 - 12 に示します。

表 6 - 12 TCSR の機能

ビット	名称	ロード/ ストア	機 能
15-11	リザーブ	-	リザーブ・ビット ・このビットには値を設定できない。読み出し時は不定。
10	TFNC	L/S	タイマ動作方法の指定ビット 0: リピート動作 (デフォルト) 1: ワンタイム動作
9	TER	L/S	タイマ・エラー・フラグ タイマ動作中に TCSR に対して μPD77115 からストアした場合、“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、タイマ停止中に 0 をストアする)
8-6	リザーブ	-	リザーブ・ビット ・このビットには値を設定できない。読み出し時は不定。
5-4	TCLKSEL	L/S	タイマ・クロック・ソース選択ビット 00: システム・クロック (デフォルト) 01: SD カード・クロック (SDCLK) 10: シリアル・クロック (SCK) 11: 割り込み端子 ($\overline{\text{INT4}}$)
3	リザーブ	-	リザーブ・ビット ・このビットには値を設定できない。読み出し時は不定。
2-0	TCLKPS	L/S	タイマ・クロック・ソース分周選択ビット 000: 1 / 1 分周 (デフォルト) 001: 1 / 2 分周 010: 1 / 4 分周 011: 1 / 8 分周 100: 1 / 16 分周 101: 1 / 32 分周 110: 1 / 64 分周 111: 1 / 128 分周

(4) タイマ・イネーブル・レジスタ (TENR)

タイマ・イネーブル・レジスタ (TENR) は、タイマ動作のオン / オフを設定するレジスタです。オン / オフするビットは LSB の 1 ビットで行います。TENR の初期値は 0x0000 で、X バスおよび Y バスと入出力できます。TENR の各ビットの機能を表 6 - 13 に示します。

表 6 - 13 TENR の機能

ビット	名称	ロード/ ストア	機 能
15-1	リザーブ	-	リザーブ・ビット ・このビットには値を設定できない。読み出し時は不定。
0	TEN	L/S	タイマ・オン / オフ指定ビット 0: オフ (デフォルト) 1: オン

第7章 μ PD77115 のブート機能

μ PD77115 は内部命令 RAM にブートアップするためのブート用プログラム（ブート ROM）を内蔵しています。

内部命令 RAM（0x0200-0x0FFF）はリセット・ブートアップおよびホスト・リブート処理によってブートアップを行います。内部命令 RAM（0x8000-0x9FFF）はホスト・リブート処理によってブートアップを行います。電源投入時には必ずリセット・ブートによって内部命令 RAM に μ PD77115 のプログラムをブートアップする必要があります。

7.1 リセット時のブート

リセット解除後にブート・モード（汎用ポートで設定）をブート ROM でチェックします。汎用ポートの設定とブート・モードの関係を表 7-1 に示します。

表 7-1 ブート・モード

P1	P0	ブート・モード
0	0	ノンブート
0	1	ホスト・バイト・ブート
1	1	ホスト・ワード・ブート
1	0	禁止

注意 μ PD77115 では、外部データ・メモリ空間がないため、セルフ・ブートはサポートしません。

ブートアップ中に割り込みを受け付けた場合には μ PD77115 の正常な動作は保証しません。

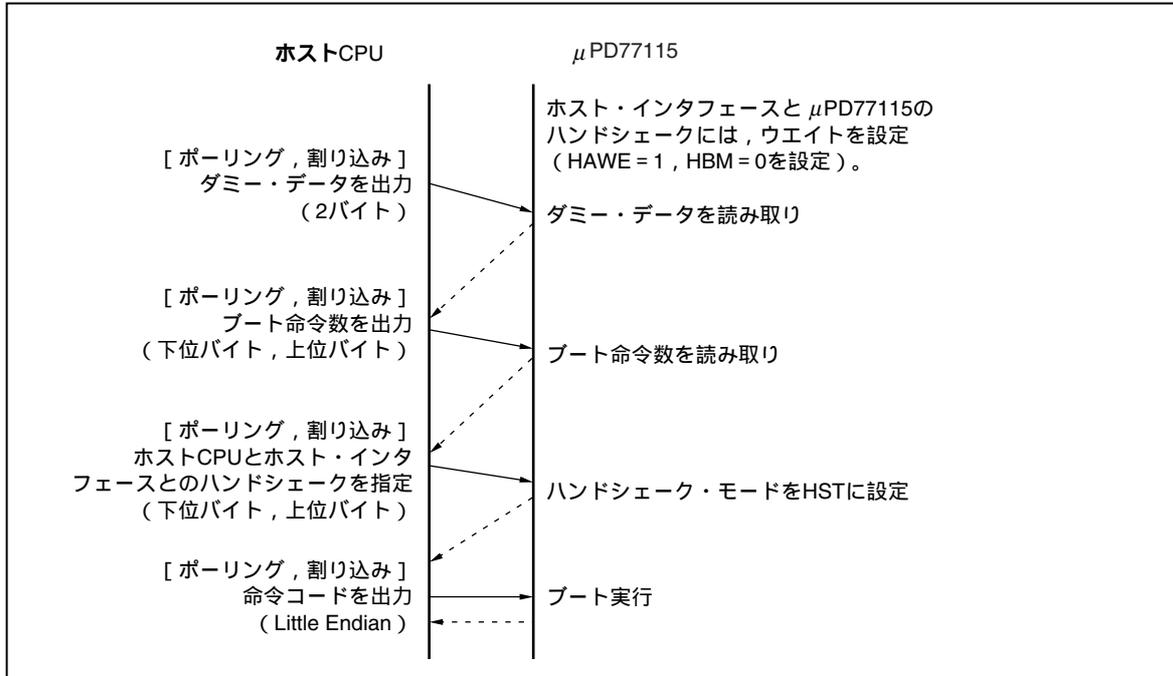
リセット・ブートアップはホスト・ブートのみをサポートし、バイト・ブートとワード・ブートがあります。

7.1.1 バイト・ブートの場合

ホスト・バイト・ブートの手順を図7-1に示します。命令コードは、リトル・エンディアンで読み取ります。

ホスト・バイト・ブートを行う場合、ハンドシェイク指定でステータス・レジスタ (HST) の HAWE (HDT アクセス・ウエイト許可ビット) を 1, HBM (HDT アクセス・ビット・モード) を 0 に設定する必要があります。

図7-1 ホスト・バイト・ブートの手順

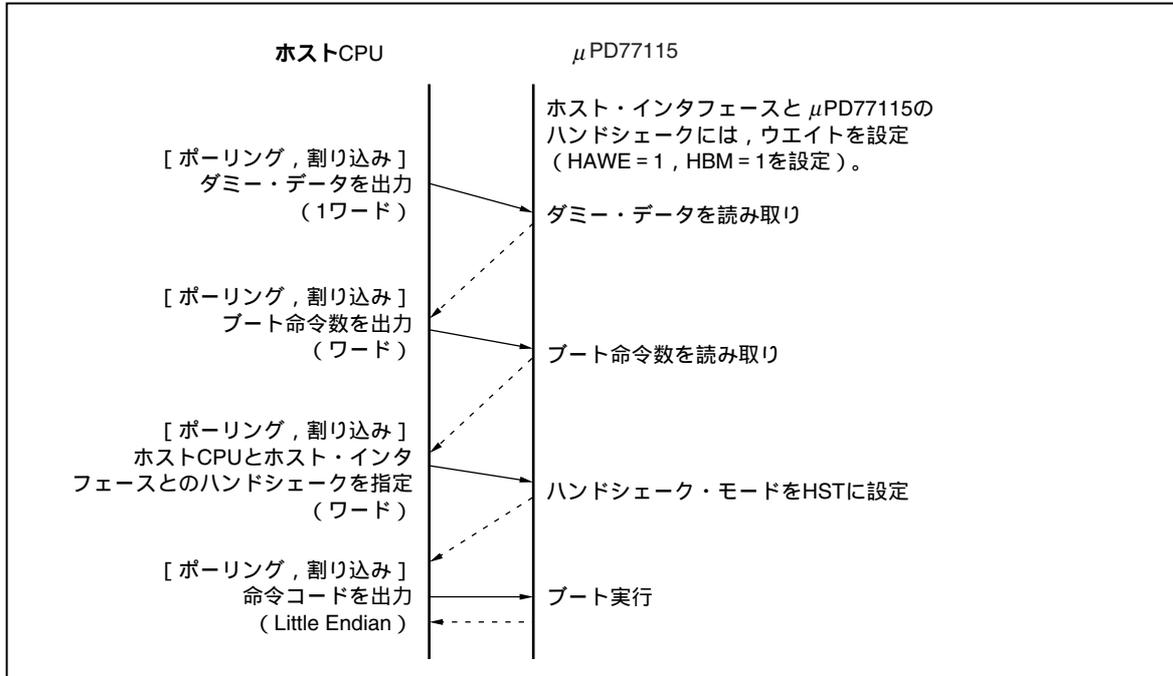


7.1.2 ワード・ブートの場合

ホスト・ワード・ブートの手順を図7-2に示します。命令コードは、リトル・エンディアンで読み取ります。

ホスト・ワード・ブートを行う場合、ハンドシェイク指定でステータス・レジスタ (HST) の HAWE (HDT アクセス・ウエイト許可ビット) を 1, HBM (HDT アクセス・ビット・モード) を 1 にする必要があります。

図7-2 ホスト・ワード・ブートの手順



7.2 リポート

命令 RAM の内容を変更 (リポート) するには、ユーザ・プログラム上でパラメータを設定し、ブート ROM に用意したサブルーチンを使用します。

リポートをサブルーチン・コールすると、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 / 復帰を行う必要があります。

リポートは、すべての割り込みを禁止した状態で行います。リポート中に割り込みを受け付けた場合には μ PD77115 の正常な動作は保証できません。

リポートはホスト・ブートのみをサポートしています。

ホスト・リポートは内部命令 RAM (0x0200-0x0FFF) の 3.5 K ワード空間および内部命令 RAM (0x8000-0x9FFF) の 8K ワード空間をリポートするためのルーチンです。パラメータの設定は次のとおりです。また、HDT に読み出す前のデータを残さないでください。ホスト・リポートのエントリ・アドレスは 0x6 です。

- ・ R7L : ブートする命令ワード数 (必須)
- ・ DP3 : リポートする命令メモリの先頭アドレス (必須)
- ・ HST : μ PD77115CPU とペリフェラルのハンドシェイク指定 (必須)
ホスト CPU と μ PD77115 のハンドシェイク指定 (選択)

ブート先として内部命令 RAM (0x0200-0x0FFF および 0x8000-0x9FFF) 以外を選択しないでください。最大ブート・ワード数は、それぞれ 0x0E00 および 0x2000 です。

注意 ホスト・リブートの実行では、 μ PD77115 内部と内蔵 HIO とのインタフェースはウエイトにより行うため、HST 内の HAWE ビットのセット (HAWE = 1) が必須となります。

ホスト・リブートをバイト・データで行う場合は、HST 内の HBM ビットを 0 にセットする必要があり、ワード・データで行う場合は、HBM ビットを 1 にセットする必要があります。

7.3 シグネチャ演算

μ PD77115 は、大容量の命令 RAM を内蔵するため、命令 RAM データの内容をベリファイする機能がサポートされています。シグネチャ演算処理は、ブートアップされた命令 RAM のデータをチェックするため、IEEE802.3 で勧告している次に示す生成多項式の演算処理を行い、結果をレジスタに返すものです。シグネチャ演算処理はブート・プログラム中にサブルーチンとして用意されています。エントリ・アドレス 0x9 のサブルーチン・コールで実行され、結果を R7 に返します。

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{21} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^6 + x^4 + x^2 + x^1 + 1$$

シグネチャ演算処理をサブルーチン・コールすると、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) 使用します。必要なレジスタは、退避 / 復帰を行なう必要があります。

シグネチャ演算処理に使用するレジスタは、R1-R7, DP3, DP6 です。

注意 シグネチャ演算処理は、すべての割り込みを禁止した状態で行います。演算処理中に割り込みを受け付けた場合には μ PD77115 の正常な動作は保証できません。パラメータの設定は次のとおりです。

- ・ R7L : 演算する命令ワード数 (必須)
- ・ DP3 : 演算する命令メモリの先頭アドレス (必須)

演算対象として内部命令 RAM (0x0200-0x0FFF および 0x8000-0x9FFF) 以外を選択しないでください。最大ワード数は、0x0E00 および 0x2000 です。

7.4 ブート所要時間

モード	時間 (サイクル数)
ホスト・バイト・ブート (8 ビット)	$(W \times 2 \times 2 \times 7) + 32$
ホスト・ワード・ブート (16 ビット)	$(W \times 2 \times 7) + 32$
ホスト・バイト・リブート (8 ビット)	$(W \times 2 \times 2 \times 7)$
ホスト・ワード・リブート (16 ビット)	$(W \times 2 \times 7)$
シグネチャ演算	$(W \times 24) + 30$

W : ブート命令ワード数

注意 リブートの所要時間は、リブート・コール実行サイクルの次のサイクルから、リブート・コールの次の命令実行サイクル前までの命令サイクル数を表します。

第8章 開発ツール

ここでは、 μ PD77111 ファミリの開発ツールを紹介します。

注意 この章はどのような開発ツールがあるかを述べたものです。開発ツールの詳細な解説については、各ツールのマニュアルを参照してください。

8.1 ソフトウェア・ツール

μ PD77111 ファミリのソフトウェア・ツールは、Windows[®] 95, 98, Windows NT[®] 4.0 ベースで用意されています。

また、これらのツールは μ PD77016 ファミリー共通です。

8.1.1 統合化開発環境ワークベンチ (WB77016)

リロケータブル・アセンブラ、リンカ、エディタ、Makeユーティリティを一体化したプログラム開発環境です。プログラムの編集からオブジェクト・プログラムの作成、ソフトウェア・シミュレータの起動までを一貫して効率よく行うことができます。

ホスト・マシン	媒体	オーダ名称
PC-9800 シリーズ, IBM PC/AT [™] 共通	CD-ROM	μ SAB17WB77016

8.1.2 ハイスピード・シミュレータ (HSM77016)

μ PD77111 ファミリー, および μ PD77016 ファミリーの動作をシミュレートできます。

シミュレーションの対象は、プログラム制御ユニット、外部メモリ、インストラクション・メモリ、ホスト・インタフェース、シリアル・インタフェースおよび I/O ポートです。

ホスト・マシン	媒体	オーダ名称
PC-9800 シリーズ, IBM PC/AT 共通	CD-ROM	μ SAB17SM77016

8.1.3 C コンパイラ (CC77016)

ANSI 規格に準拠した高性能 C クロス・コンパイラにより、 μ PD77111 ファミリ、および μ PD77016 ファミリのソフトウェア・アプリケーションを作成します。

C ソース・プログラムを WB77016 のアセンブリ言語へコンパイルします。アセンブル、リンクにはワークベンチ (WB77016) を、コード検証にはソフトウェア・シミュレータ (HSM77016) をご利用ください。

ホスト・マシン	媒体	オーダ名称
PC-9800 シリーズ, IBM PC/AT 共通	CD-ROM	μ SAB17CC77016

8.1.4 インサーキット・エミュレータ用システム・ソフトウェア (ID77016)

インサーキット・エミュレータを制御するためのソフトウェアです。ソフトウェア・シミュレータと同一のユーザ・インタフェースでインサーキット・エミュレータを操作することができます。

ホスト・マシン	媒体	オーダ名称
PC-9800 シリーズ, IBM PC/AT 共通	CD-ROM	μ SAB17ID77016

8.1.5 DSP 開発ツール・ソフトウェア・パッケージ (SP77016)

WB77016, HSM77016, ID77016 を 1 枚の CD-ROM に収録した製品です。

ホスト・マシン	媒体	オーダ名称
PC-9800 シリーズ, IBM PC/AT 共通	CD-ROM	μ SAB17SP77016

8.2 ハードウェア・ツール

8.2.1 インサーキット・エミュレータ

デバイス開発のインサーキット・エミュレーション機能とホスト・マシンとの間のインタフェースとなる製品です。インサーキット・エミュレータを使用するには、システム・ソフトウェア (ID77016) と、ID77016 が動作する環境が必要です。8.1.4 インサーキット・エミュレータ用システム・ソフトウェアを参照してください。

(1) USB インタフェース版インサーキット・エミュレータ (NDSP_ICE001)

ホスト・マシン	オーダ名称
PC-9800 シリーズ, IBM PC/AT 共通	-

備考1. ホスト・マシン本体に USB インタフェースが内蔵されている必要があります。

動作対象 OS については、販売、サポート先までお問い合わせください。

2. μ PD77111, 77112, 77113A, 77114, 77115 を評価する場合には、エバリエーション・チップとして μ PD77110 を使用することをお勧めします。
3. この製品の販売、サポートはアンドール システム サポート株式会社 (〒140-0004 東京都品川区南品川 2-15-8 TEL 03 (3450) 8101) が行います。

(2) PCMCIA インタフェース版インサーキット・エミュレータ (NDSP_ICE002)

ホスト・マシン	オーダ名称
PC-9800 シリーズ, IBM PC/AT 共通	-

備考1. ホスト・マシン本体に PCMCIA インタフェースが内蔵されている必要があります。

動作対象 OS については、販売、サポート先までお問い合わせください。

2. μ PD77111, 77112, 77113A, 77114, 77115 を評価する場合には、エバリエーション・チップとして μ PD77110 を使用することをお勧めします。
3. この製品の販売、サポートはアンドール システム サポート株式会社 (〒140-0004 東京都品川区南品川 2-15-8 TEL 03 (3450) 8101) が行います。

8.2.2 その他のツール

(1) アプリケーション・ボード・ミニ

簡易的な評価ボードです。DSP とステレオ・サウンド・コーデックが搭載されています。搭載されている DSP は、 μ PD77110 と μ PD77115 の 2 種類があります。

ホスト・マシン	オーダ名称
PC-9800 シリーズ, IBM PC/AT 共通	-

備考1. インサーキット・エミュレータおよびソフトウェア・ツールは付属していません。

2. この製品の販売, サポートはアンドール システム サポート株式会社 (〒140-0004 東京都品川区南品川 2-15-8 TEL 03 (3450) 8101) が行います。

付録 A オーダ情報

A.1 オーダ情報

オーダ名称	パッケージ
μ PD77110GC-9EU	100 ピン・プラスチック TQFP (ファインピッチ)(14×14)
μ PD77111GK-x x x-9EU	80 ピン・プラスチック TQFP (ファインピッチ)(12×12)
μ PD77111F1-x x x-CN1	80 ピン・プラスチック・ファインピッチ BGA (9×9)
μ PD77112GC-x x x-9EU	100 ピン・プラスチック TQFP (ファインピッチ)(14×14)
μ PD77113AF1-x x x-CN1	80 ピン・プラスチック・ファインピッチ BGA (9×9)
μ PD77114GC-x x x-9EU	100 ピン・プラスチック TQFP (14×14)
μ PD77115F1-CN1	80 ピン・プラスチック・ファインピッチ BGA (9×9)
μ PD77115GK-9EU	80 ピン・プラスチック TQFP (ファインピッチ)(12×12)

備考 x x x は ROM コード番号です。

A.2 マスク・オプション

μ PD77111, 77112, 77113A, 77114 には, マスク ROM 発注時に設定しなければならないマスク・オプションがあります。

A.2.1 クロック制御に関するオプション

クロックに関するオプションとして, 次の4点を設定する必要があります。

PLL 通倍率

出力分周率

HALT 分周率

CLKOUT 端子出力の有効 / 無効

PLL の通倍率を m , 出力分周率を n , HALT 分周率を l とすると, 各動作モードとクロックの関係は次のとおりです。詳細は 3.3.1 クロック・ジェネレータを参照してください。

動作モード	DSP 内部に供給されるクロック
通常動作モード	外部入力クロックの m/n 倍
HALT モード	外部入力クロックの $m/n/l$ 倍
STOP モード	停止

A.2.2 WAKEUP オプション

STOP モードから復帰する方法としてハードウェア・リセットのほかに $\overline{\text{WAKEUP}}$ 端子を用いる機能をマスク・オプションで設定することができます。詳細は 3.3.4 (2) STOP 命令によるスタンバイ・モードを参照してください。

A.3 マスク ROM 受注フォーマット

ROM コードの発注方法 (C10302J), または WB77016 ユーザーズ・マニュアルを参照してください。

付録 B 索引

B.1 アルファベットで始まる語句の索引

[A]

ALU 演算機能 ... 129

[B]

BSFT シフト演算機能 ... 132

[C]

CC77016 ... 258

CRC 回路 ... 246

C コンパイラ ... 258

[D]

DSP 開発ツール・ソフトウェア・パッケージ ... 258

[G]

HALT 命令によるスタンバイ・モード ... 63

HSM77016 ... 257

[I]

ID77016 ... 258

[J]

JTAG ポート ... 177

[L]

LRC ... 76

[M]

MAC ... 122

MAC 入力シフト ... 122

MSFT ... 122

MUX ... 107

[N]

NDSP_ICE001 ... 259

NDSP_ICE002 ... 259

[P]

PCMCIA インタフェース版インサートキット・エミュレータ ... 259

PLL 設定タイミング ... 60

PLL 通倍率設定 ... 57, 208

PLL に関する留意事項 ... 211

PLL の初期化 ... 60

PLL ロック・アンロック状態遷移 ... 210

[S]

SD カード・インタフェース ... 240

SP77016 ... 258

STOP 命令によるスタンバイ・モード ... 65

[U]

USB インタフェース版インサートキット・エミュレータ ... 259

[W]

WAKEUP オプション ... 262

WB77016 ... 257

[X]

XAA ... 107

XBRC ... 107

X データ・バス ... 52

X データ・メモリ空間 ... 99

X メモリ・バイト・リポート ... 189

X メモリ・ワード・リポート ... 189

[Y]

YAA ... 107

YBRC ... 107

Y データ・バス ... 53

Y データ・メモリ ... 99

Y メモリ・バイト・リポート ... 190

Y メモリ・ワード・リポート ... 190

B.2 五十音で始まる語句の索引

[あ]

アーキテクチャ ... 48
 アドレッシング・モード ... 107
 アドレス ALU ... 107
 アプリケーション・ボード・ミニ ... 260
 インサーキット・エミュレータ ... 259
 インサーキット・エミュレータ用システム・ソフトウェア ... 258
 ウェイト ... 153, 166
 ウェイト・コントローラ ... 104, 176
 演算ユニット ... 118
 オーダ情報 ... 261
 オーディオ・シリアル・インタフェース ... 217

[か]

開発ツール ... 257
 外部データ・メモリ・インタフェース ... 100
 外部割り込みのハードウェア条件 ... 85
 間接アドレッシング ... 109
 機能端子の処理 ... 46
 クロック・ジェネレータ ... 55
 クロック・タイミング ... 56
 クロック回路 ... 56
 クロック状態遷移 ... 210
 クロック制御 ... 206
 クロック制御に関するオプション ... 261
 クロック制御部の端子構成 ... 207
 固定小数点フォーマット ... 121
 コマンド・フォーマット ... 246

[さ]

シグネチャ演算 ... 190, 256
 システム・クロック状態遷移 ... 211
 システム制御ユニット ... 54
 初期化対象端子と初期状態 ... 59, 202
 シリアル・インタフェース ... 137, 216
 数値フォーマット ... 121
 スタンダード・シリアル・インタフェース ... 216, 226
 スタンバイ・モード時の動作クロック ... 57
 スタンバイ機能 ... 63, 203
 整数フォーマット ... 122

セルフ・ブート ... 181
 セルフ・ブートのパラメータ ... 185
 全体ブロック構成 ... 48
 ソフトウェア・ツール ... 257
 ソフトウェア・ループ・スタック ... 82

[た]

タイマ ... 251
 多重割り込み ... 91
 端子機能 ... 23, 36
 端子構成 ... 34
 端子接続図 ... 25
 直接アドレッシング ... 108
 デバッグ・インタフェース ... 177
 デバッグ機能 ... 178
 デバッグ用端子 ... 177
 データ・アドレッシング・ユニット ... 98
 データ・バス ... 51
 データ・フォーマット ... 246
 データ・ポインタのモディファイア一覧 ... 111
 データ・メモリ ... 98, 212
 データ・メモリ・アクセス・タイミング ... 103
 電源投入時の推奨シーケンス ... 203
 電源投入時の注意 ... 202

[な]

内部データ・メモリ ... 100
 内部命令メモリ ... 70
 ノー・チェンジ ... 109

[は]

バス ... 50
 ハードウェア・ツール ... 259
 ハードウェアの初期化 ... 58
 ハイスピード・シミュレータ ... 257
 バイト・ブート ... 186, 254
 パイプライン・アーキテクチャ ... 61
 バウンダリ・スキャン・テスト機能 ... 178
 バス・アービトレーション ... 105
 ハンドシェーク ... 151, 165

- 汎用入出力ポート ... 168, 235
- 汎用レジスタとデータ・フォーマット ... 119
- 非機能端子の処理 ... 47
- ビット・リバース回路 ... 107
- ブート・サブルーチン ... 189
- ブート・モード ... 181
- ブート機能 ... 180, 253
- ブート所要時間 ... 197, 256
- プレ・ビット・リバース・アンド・ポスト・インデクス
加算 ... 110
- フロー制御部 ... 75
- プログラム実行制御部 ... 69
- プログラム実行フロー ... 195
- プログラム制御ユニット ... 68
- ペリフェラル ... 134, 214
- ペリフェラル・バス ... 54
- ペリフェラル・バッファ ... 248
- ペリフェラル・バッファ動作フロー ... 250
- ペリフェラル・レジスタ ... 136, 215
- ポート・インタフェースの端子 ... 169
- ポート関連レジスタ ... 169
- ポーリング ... 152, 165
- ポスト・イミューディエト加算 ... 111
- ポスト・インクリメント ... 109
- ホスト・インタフェース ... 155, 227
- ポスト・インデクス加算 ... 110
- ポスト・デクリメント ... 110
- ホスト・ブート ... 182
- ホスト・ブートのパラメータ ... 188
- ポスト・モジュロ・インデクス加算 ... 110
- ホスト・リブートのパラメータ ... 190
- [ま]
- マスク・オプション ... 261
- マスク ROM 受注フォーマット ... 262
- マルチプレクサ ... 107
- 未使用端子の処理 ... 46
- 命令 ROM 修正処理 ... 191
- 命令メモリ ... 69, 212
- メイン・バス ... 50
- メイン・バス接続レジスタ ... 51
- モジュロ・インデクス加算と巡回バッファ ... 112
- [ら]
- リセット・タイミング ... 60, 201
- リセット・ブート ... 181
- リセット・ブートアップ ... 253
- リセット機能 ... 58, 201
- リセット時のブート ... 184, 253
- リピート・カウンタ (RC) のフォーマット ... 77
- リピート機能 ... 77
- リブート ... 181, 189, 255
- リング・カウンタの動作範囲 ... 114
- ループ/リピート制御回路 ... 76
- ループ・カウンタ (LC) のフォーマット ... 79
- ループ機能 ... 78
- 累乗算器 ... 122
- レスポンス・フォーマット ... 247
- [わ]
- ワード・ブート ... 186, 255
- ワ - クベンチ ... 257
- 割り込み ... 83, 153, 167
- 割り込みシーケンス ... 92
- 割り込み制御機能 ... 84
- 割り込みの受け付け条件 ... 84
- 割り込みの競合と記憶 ... 96
- 割り込みの遅延 ... 95
- 割り込みベクタ ... 85
- 割り込みベクタ・テーブル ... 86, 213
- 割り込み要因 ... 83

B.3 レジスタ索引

B.3.1 レジスタ名称順

[A~Z]

SD カード・インタフェース・コントロール・レジスタ : SDCTL ... 244
 SD カード・インタフェース・レスポンス・レジスタ : SDRPR ... 242
 SD カード・インタフェース CRC ステータス・ビジィ・レジスタ : SDSBR ... 243
 SD カード・コマンド・アーギュメント・ハイ側レジスタ : SDCMD_AGH ... 244
 SD カード・コマンド・アーギュメント・ロウ側レジスタ : SDCMD_AGL ... 244
 SD カード・コマンド・インデクス・レジスタ : SDCMD_IDX ... 243
 SD カード・コマンド・レジスタ : SDCMD ... 243
 SD カード・データ出力レジスタ : SDDOR ... 242
 SD カード・データ入力レジスタ : SDDIR ... 242

[あ]

インデクス・レジスタ : DN0-DN7 ... 107
 エラー・ステータス・レジスタ : ESR ... 97
 オーディオ・シリアル・ステータス・レジスタ : ASST ... 220
 オーディオ・シリアル・データ・レジスタ : ASDT ... 219
 オーディオ・シリアル・データ出力レジスタ : ASDT (out) ... 219
 オーディオ・シリアル・データ入力レジスタ : ASDT (in) ... 219
 オーディオ・シリアル出力シフト・レジスタ : ASOS ... 220
 オーバフロー・エラー・フラグ : ovf ... 97

[か]

クロック・コントロール・レジスタ : CLKCNTL ... 209

[さ]

シリアル・ステータス・レジスタ : SST,SST1,SST2 ... 136, 142, 220
 シリアル・ストア・イネーブル・フラグ : SSEF ... 138
 シリアル・データ・レジスタ : SDT1, SDT2 ... 136, 141
 シリアル・データ出力レジスタ : SDT (out) ... 138, 142
 シリアル・データ入力レジスタ : SDT (in) ... 138, 142
 シリアル・ロード・イネーブル・フラグ : SLEF ... 138
 シリアル入力シフト・レジスタ : SIS1,SIS2 ... 138, 143
 シリアル出力シフト・レジスタ : SOS1,SOS2 ... 138, 143
 スタック : STK ... 70
 スタック・エラー・フラグ : ste ... 97
 スタック・ポインタ : SP ... 70
 ステータス・レジスタ : SR ... 88

[た]

タイマ・イネーブル・レジスタ : TENR ... 252
タイマ・カウント・レジスタ : TCR ... 251
タイマ・コントロール・ステータス・レジスタ : TCSR ... 252
タイマ初期値レジスタ : TIR ... 251
データ・ポインタ : DP0-DP7 ... 107
データ・メモリ・ウエイト・サイクル・レジスタ : DWTR ... 104, 136, 176
テスト・インストラクション・レジスタ : Test Instruction Register ... 178
テスト・バイパス・レジスタ : Test Bypass Register ... 178

[は]

プログラム・カウンタ : PC ... 69
ペリフェラル・バッファ・アドレス・ポインタ・レジスタ : PPR ... 249
ペリフェラル・バッファ・コントロール・レジスタ : PCR ... 249
ペリフェラル・バッファ・サイズ・カウント・レジスタ : PSCR ... 249
ペリフェラル・バッファ・サイズ・レジスタ : PSR ... 248
ペリフェラル・バッファ・スタート・アドレス・レジスタ : PSAR ... 248
ポート・コマンド・レジスタ : PCD ... 136, 170, 236
ポート・データ・レジスタ : PDT ... 136, 169, 236
ホスト・ステータス・レジスタ : HST ... 136, 159
ホスト・データ・レジスタ : HDT ... 136, 159, 229
ホスト・データ出力レジスタ : HDT(out) ... 156, 159, 229
ホスト・データ入力レジスタ : HDT(in) ... 156, 159, 229

[ま]

モジュロ・レジスタ : DMX, DMY ... 107

[ら]

リピート・カウンタ : RC ... 75
ループ・カウンタ : LC ... 76
ループ・スタック : LSTK ... 76
ループ・スタック・エラー・フラグ : lse ... 97
ループ・スタック・ポインタ : LSP ... 76
ループ・フラグ : LF ... 89
ループ開始アドレス・レジスタ : LSA ... 75
ループ終端アドレス・レジスタ : LEA ... 75

[わ]

割り込み許可フラグ : EB ... 88
割り込み許可フラグ : EI ... 88
割り込み許可フラグ : EP ... 88
割り込み許可フラグ・スタック・レジスタ : EIR ... 90

B.3.2 レジスタ略号順

[A]

ASDT : オーディオ・シリアル・データ・レジスタ ... 219
 ASDT (in) : オーディオ・シリアル・データ入力レジスタ ... 219
 ASDT (out) : オーディオ・シリアル・データ出力レジスタ ... 219
 ASOS : オーディオ・シリアル出力シフト・レジスタ ... 220
 ASST : オーディオ・シリアル・ステータス・レジスタ ... 220

[C]

CLKCNTL : クロック・コントロール・レジスタ ... 209

[D]

DMX , DMY : モジュロ・レジスタ ... 107
 DN0-DN7 : インデクス・レジスタ ... 107
 DP0-DP7 : データ・ポインタ ... 107
 DWTR : データ・メモリ・ウエイト・サイクル・レジスタ ... 104, 136, 176

[E]

EB : 割り込み許可フラグ ... 88
 EI : 割り込み許可フラグ ... 88
 EIR : 割り込み許可フラグ・スタック・レジスタ ... 90
 EP : 割り込み許可フラグ ... 88
 ESR : エラー・ステータス・レジスタ ... 97

[H]

HDT : ホスト・データ・レジスタ ... 136, 159, 229
 HDT(in) : ホスト・データ入力レジスタ ... 156, 159, 229
 HDT(out) : ホスト・データ出力レジスタ ... 156, 159, 229
 HST : ホスト・ステータス・レジスタ ... 136, 159

[L]

LC : ループ・カウンタ ... 76
 LEA : ループ終端アドレス・レジスタ ... 75
 LF : ループ・フラグ ... 89
 LSA : ループ開始アドレス・レジスタ ... 75
 lse : ループ・スタック・エラー・フラグ ... 97
 LSP : ループ・スタック・ポインタ ... 76
 LSTK : ループ・スタック ... 76

[O]

ovf : オーバフロー・エラー・フラグ ... 97

[P]

PC : プログラム・カウンタ ... 69
 PCD : ポート・コマンド・レジスタ ... 136, 170, 236
 PCR : ペリフェラル・バッファ・コントロール・レジスタ ... 249
 PDT : ポート・データ・レジスタ ... 136, 169, 236
 PPR : ペリフェラル・バッファ・アドレス・ポインタ・レジスタ ... 249
 PSAR : ペリフェラル・バッファ・スタート・アドレス・レジスタ ... 248
 PSCR : ペリフェラル・バッファ・サイズ・カウント・レジスタ ... 249
 PSR : ペリフェラル・バッファ・サイズ・レジスタ ... 248

[R]

RC : リピート・カウンタ ... 75

[S]

SDCMD : SD カード・コマンド・レジスタ ... 243
 SDCMD_AGH : SD カード・コマンド・アーギュメント・ハイ側レジスタ ... 244
 SDCMD_AGL : SD カード・コマンド・アーギュメント・ロウ側レジスタ ... 244
 SDCMD_IDX : SD カード・コマンド・インデクス・レジスタ ... 243
 SDCTL : SD カード・インタフェース・コントロール・レジスタ ... 244
 SDDIR : SD カード・データ入力レジスタ ... 242
 SDDOR : SD カード・データ出力レジスタ ... 242
 SDRPR : SD カード・インタフェース・レスポンス・レジスタ ... 242
 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 ... 243
 SDT (in) : シリアル・データ入力レジスタ ... 138, 142
 SDT (out) : シリアル・データ出力レジスタ ... 138, 142
 SDT1, SDT2 : シリアル・データ・レジスタ ... 136, 141
 SIS1, SIS2 : シリアル入力シフト・レジスタ ... 138, 143
 SLEF : シリアル・ロード・イネーブル・フラグ ... 138
 SOS1, SOS2 : シリアル出力シフト・レジスタ ... 138, 143
 SP : スタック・ポインタ ... 70
 SR : ステータス・レジスタ ... 88
 SSEF : シリアル・ストア・イネーブル・フラグ ... 138
 SST, SST1, SST2 : シリアル・ステータス・レジスタ ... 136, 142, 220
 ste : スタック・エラー・フラグ ... 97
 STK : スタック ... 70

[T]

TCR : タイマ・カウント・レジスタ ... 251
 TCSR : タイマ・コントロール・ステータス・レジスタ ... 252
 TENR : タイマ・イネーブル・レジスタ ... 252
 Test Bypass Register : テスト・バイパス・レジスタ ... 178
 Test Instruction Register : テスト・インストラクション・レジスタ ... 178
 TIR : タイマ初期値レジスタ ... 251

— お問い合わせ先 —

【技術的なお問い合わせ先】

NEC半導体テクニカルホットライン
(電話：午前 9:00～12:00，午後 1:00～5:00)

電話 : 044-435-9494
FAX : 044-435-9608
E-mail : info@lsi.nec.co.jp

【営業関係お問い合わせ先】

システムLSI第一営業事業部

東京 (03)3798-6106, 6107, 6108, 6155
大阪 (06)6945-3178, 3200, 3208
名古屋 (052)222-2375
仙台 (022)267-8740
水戸 (029)226-1702
広島 (082)242-5504
鳥取 (0857)27-5313
松山 (089)945-4149

システムLSI第二営業事業部

東京 (03)3798-6110, 6111, 6112, 6151, 6156
名古屋 (052)222-2170, 2190
松本 (0263)35-1662
前橋 (027)243-6060
立川 (042)526-5981
静岡 (054)254-4794
金沢 (076)232-7303
福岡 (092)261-2806

【資料の請求先】

上記営業関係お問い合わせ先またはNEC特約店へお申しつけください。

【NECエレクトロニクス デバイス 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.ic.nec.co.jp/>

アンケート記入のお願い

お手数ですが、このドキュメントに対するご意見をお寄せください。今後のドキュメント作成の参考にさせていただきます。

[ドキュメント名] μPD77111 ファミリ ユーザーズ・マニュアル アーキテクチャ編

(U14623JJ3V0UM00 (第3版))

[お名前など](さしつかえのない範囲で)

御社名(学校名, その他) ()
ご住所 ()
お電話番号 ()
お仕事の内容 ()
お名前 ()

1. ご評価(各欄に をご記入ください)

項 目	大変良い	良 い	普 通	悪 い	大変悪い
全体の構成					
説明内容					
用語解説					
調べやすさ					
デザイン, 字の大きさなど					
その他()					
()					

2. わかりやすい所(第 章, 第 章, 第 章, 第 章, その他)

理由 []

3. わかりにくい所(第 章, 第 章, 第 章, 第 章, その他)

理由 []

4. ご意見, ご要望

5. このドキュメントをお届けしたのは

NEC 販売員, 特約店販売員, その他 ()

ご協力ありがとうございました。

下記あてに FAX で送信いただくか, 最寄りの販売員にコピーをお渡しください。

日本電気(株) NEC エレクトロニクス
半導体テクニカルホットライン
FAX : (044) 435-9608

2000.6