



VLIIO メモリ・インターフェイスを介した インテル® PXA27x プロセッサ・ファミリと ハードディスク・ドライブとの接続

アプリケーション・ノート

2005 年 6 月

本資料に掲載されている情報は、インテル製品の概要説明を目的としたものです。本資料は、明示されているか否かにかかわらず、また禁反言によらずにかかわらず、いかなる知的財産権のライセンスを許諾するためのものではありません。製品に付属の売買契約書『Intel's Terms and conditions of Sales』に規定されている場合を除き、インテルはいかなる責を負うものではなく、またインテル製品の販売や使用に関する明示または黙示の保証（特定目的への適合性、商品性に関する保証、第三者の特許権、著作権、その他、知的所有権を侵害していないことへの保証を含む）にも一切応じないものとします。

インテル製品は、医療、救命、延命措置などの目的への使用を前提としたものではありません。インテル製品は、予告なく仕様に変更される場合があります。

最新の仕様をご希望の場合や製品をご注文の場合は、お近くのインテルの営業所または販売代理店にお問い合わせください。

本書で紹介されている注文番号付きのドキュメントや、インテルのその他の資料を入手するには、1-800-548-4725（米国）にお問い合わせになるか、インテルのWebサイト <http://www.intel.com/> にアクセスしてください。

Intel、インテル、Intel ロゴ、MMX、Intel XScale は、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

* その他の社名、製品名などは、一般に各社の商標または登録商標です。

© 2005 Intel Corporation. 無断での引用、転載を禁じます。

目次

1.0	はじめに.....	5
2.0	ハードウェアの概要.....	5
2.1	インテル® PXA27x プロセッサのメモリ・インターフェイス	5
2.2	インテル® PXA27x プロセッサの VLIO インターフェイス	6
2.3	インテル® PXA27x プロセッサの DMA コントローラ・インターフェイス ...	7
2.4	CompactFlash のツール IDE 規格.....	8
3.0	インテル® PXA27x プロセッサの VLIO インターフェイスと ツール IDE CompactFlash インターフェイスとのマッピング	9
3.1	インテル® PXA27x プロセッサへの CompactFlash HDD の接続.....	10
3.2	ツール IDE マルチワード DMA モードのブロック図.....	11
3.3	タイミングに関する考慮事項	12
3.4	インテル® PXA27x プロセッサのレジスタ設定.....	13
3.4.1	MSC レジスタ設定.....	13
3.4.2	インテル® PXA27x プロセッサの GPIO レジスタ設定.....	14
4.0	HDD デバイスドライバ.....	14
4.1	HDD ドライバの概要	14
4.2	HDD 転送モードの設定.....	15
4.3	I/O アドレス・マッピング	15
4.4	PIO モードと割り込み処理.....	15
4.5	DMA モードと割り込み処理	16
4.6	高度な機能の設定	16
4.7	Linux ドライバに関する考慮事項	17
4.7.1	I/O アドレス・マッピング	17
4.7.2	DMA チャンネルの要求.....	18
4.7.3	IRQ の要求.....	18
4.7.4	DMA バッファおよびディスクリプタの割り当て.....	18
4.8	Microsoft* Windows* CE ドライバに関する考慮事項	18
4.8.1	I/O アドレス・マッピング	18
4.8.2	DMA チャンネルの要求.....	18
4.8.3	IRQ の要求.....	18
4.8.4	DMA バッファおよびディスクリプタの割り当て.....	18
5.0	まとめ	19

図

図 1.	インテル® PXA27x プロセッサのブロック図.....	6
図 2.	ツール IDE PIO モードのブロック図.....	10
図 3.	ツール IDE DMA モードのブロック図.....	12

表

表 1.	VLIO メモリ・インターフェイス信号.....	7
表 2.	DMA コントローラ・インターフェイス信号	8
表 3.	主要な CompactFlash ツール IDE 信号の一覧	9
表 4.	メモリ・マッピング	11
表 5.	略語の定義	19

改訂履歴

日付	改訂番号	説明
2005 年 6 月	001	アプリケーション・ノート『VLIO メモリ・インターフェイスを介したインテル® PXA27x プロセッサ・ファミリとハードディスク・ドライブとの接続』の初版

1.0 はじめに

ハンドヘルド機器の機能拡大に伴い、システム内に求められるストレージ容量が増加しています。一方、ハードディスク・ドライブ（HDD）では、容量の増加のほか、価格の低下と小型化が進んでいます。1.8インチや1.0インチなどの小型HDDは、一部の設計分野ではフラッシュ・ストレージの競合製品になっています。小型HDDの有望な用途としては、MP3プレーヤ、ポータブル・メディア・プレーヤ、ポータブル・ゲーム・システム、PDA、電話などがあります。

インテル® PXA27x プロセッサ・ファミリ（以後「PXA27x プロセッサ」）は、Intel XScale® マイクロアーキテクチャをベースにした統合型のシステム・オン・チップ設計を採用しています。624MHzのIntel XScale® コアと数多くのオンチップ周辺機器を組み合わせたPXA27x プロセッサは、ハンドヘルド市場と携帯電話市場でさまざまな製品を実現します。PXA27x プロセッサのメモリ・コントローラは、HDDインターフェイスを実装するためのソリューションを複数提供しています。このインターフェイスには、PCMCIA/CompactFlash* インターフェイスやVariable Latency Input/Output (VLIO) インターフェイスが含まれます。本書では、VLIOメモリ・インターフェイスを利用してCompactFlash、トゥルー IDE モード、パラレル ATA、HDDをPXA27x プロセッサに実装する方法について詳細に説明します。また、Programmed I/O (PIO) やフロースルー DMA を利用した設定についても紹介します。VLIOインターフェイス・ソリューションは、わずかなコンポーネントを追加するだけで、ローコストで効率的なDMAパフォーマンスを発揮します。

2.0 ハードウェアの概要

ここでは、VLIOインターフェイスを介してPXA27xプロセッサをHDDに接続する際のハードウェア要件について説明します。

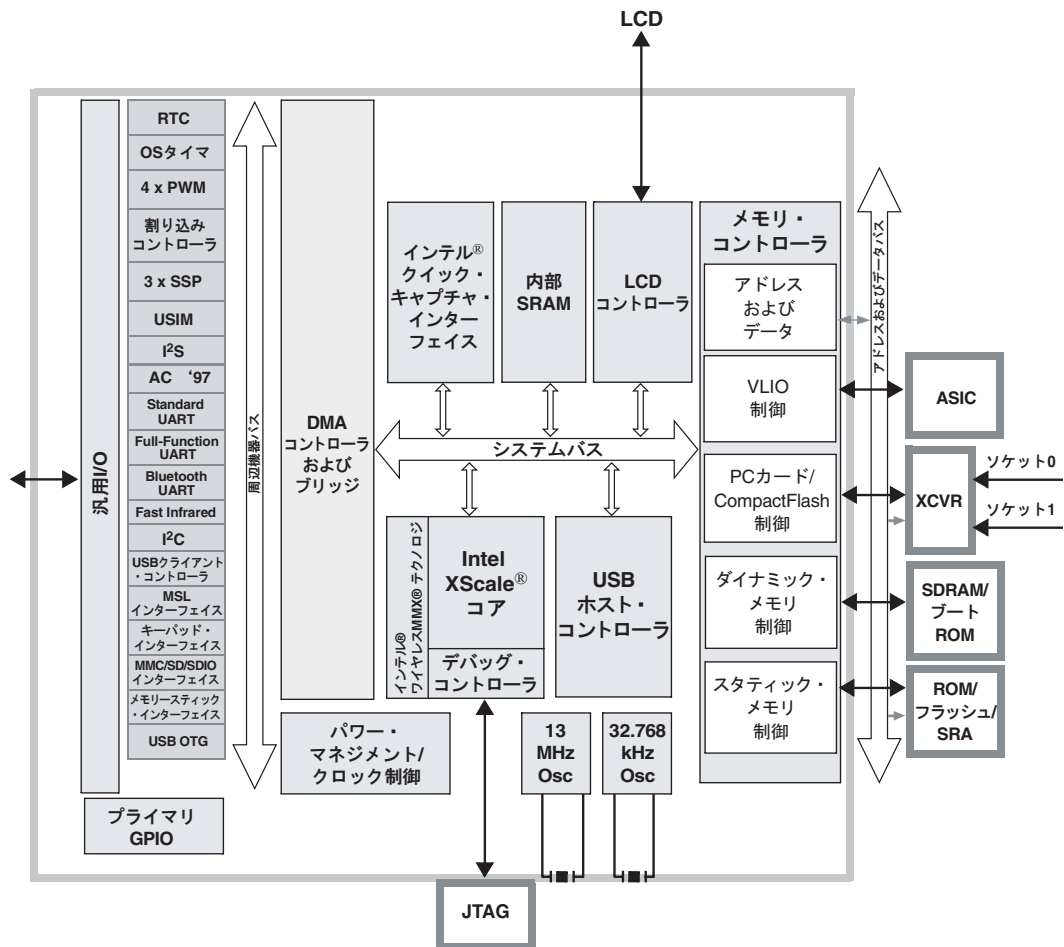
2.1 インテル® PXA27x プロセッサのメモリ・インターフェイス

PXA27xプロセッサのメモリ・コントローラは、以下のインターフェイスを介してさまざまな種類のメモリと処理をサポートします。

- SDRAMインターフェイス
- フラッシュ・メモリ・インターフェイス（非同期/同期）
- ROMインターフェイス
- VLIOインターフェイス
- PCカード（PCMCIA）インターフェイス
- 代替バス・マスタ・インターフェイス

図1に、PXA27xプロセッサとメモリ・インターフェイスを示します。

図 1. インテル® PXA27x プロセッサのブロック図



2.2 インテル® PXA27x プロセッサの VLIO インターフェイス

6つのスタティック・メモリ・ロケーション (nCS<5:0>) はいずれも、対応する MSCx[RTx] を 0b100 にプログラミングすれば、VLIO モードで動作するように設定することが可能です。静的なチップ選択を VLIO インターフェイスとして利用する場合、SRAM のときと同じように機能しますが、重要な違いがいくつかあります。

- VLIO では、RDY ピンを介してウェイトステートの変数を挿入できます。
- VLIO の読み出しアクセスは、nOE がバーストのビートごとにトグルするという点で、SRAM の読み出しアクセスと異なります。
- 最初の nOE アサーションは、チップ選択信号 (nCSx) がアサートされてから 2 CLK_MEM サイクル後に発生します。
- VLIO の書き込みでは、nWE の代わりに nPWE が使用されます。

表 1 に、VLIO メモリデバイスとのやりとりに利用可能な信号を示します。

表 1. VLIO メモリ・インターフェイス信号

信号名	方向	極性	説明
nCS<5:0>	出力	アクティブ・ロー	スタティック・メモリのチップ選択
MA<25:0>	出力	該当せず	すべてのメモリタイプへの出力アドレス 注意：すべての VLIO デバイスは PXA27x プロセッサのメモリ・コントローラとやりとりする際に16ビット以上のバス幅を必要とするため、MA0 をバイトアドレス指定に使用しないでください。MA0 を使って 128MB パーティション内の上位64MBメモリをアドレス指定するには、オプションのアドレス指定モードを利用できます。
MD<31:0>	双方向	該当せず	すべてのメモリタイプの双方向データ
DQM<3:0>	出力	アクティブ・ハイ	データ・バイト・マスク制御 DQM<0> は MD<7:0> に対応 DQM<1> は MD<15:8> に対応 DQM<2> は MD<23:16> に対応 DQM<3> は MD<31:24> に対応 0 = 対応するバイトをマスクしない 1 = 対応するバイトをマスクする
nWE	出力	アクティブ・ロー	VLIO メモリの書き込みを有効化
nOE	出力	アクティブ・ロー	スタティック・メモリの出力を有効化
RDY	入力	アクティブ・ハイ	ウェイトステートを挿入するための VLIO 信号 0 = 待機 1 = VLIO の準備が完了
その他の I/O 信号			
RDnWR	出力	アクティブ・ハイ	出力トランシーバによって使用されるデータ方向の信号 0 = MD<31:0> は PXA27x プロセッサによって制御される 1 = MD<31:0> は PXA27x プロセッサによって制御されない

2.3 インテル® PXA27x プロセッサの DMA コントローラ・インターフェイス

PXA27x プロセッサの DMA コントローラは、周辺機器やコンパニオン・チップによって生成された要求に応じて、メモリシステムとの間でデータ転送を行います。こうした周辺機器やコンパニオン・チップは、アドレスおよびコマンドをメモリシステムに直接提供することができません。代わりに、アドレスおよびコマンドは、ソフトウェアによって設定され、PXA27x プロセッサの DMA コントローラ内にある 32 本の DMA チャンネルによって維持されます。PXA27x プロセッサの DMA コントローラは、フロースルー転送とフライバイ転送の両方をサポートしています。フロースルー転送とフライバイ転送の使用法および制約や、PXA27x プロセッサの DMA コントローラの詳細については、『Intel® PXA27x Processor Family Design Guide』のセクション 5 と『Intel® PXA27x Processor Family Developer's Manual』を参照してください。本書で扱われている HDD インターフェイスでは、

フロースルーDMA を利用することによって、データ転送処理を行う PXA27x プロセッサの DMA コントローラに起因したソフトウェアのオーバーヘッドを削減しながら、高いスループットを実現しています。

表 2 に、DMA コントローラとのやりとりに利用可能な信号を示します。

表 2. DMA コントローラ・インターフェイス信号

信号名	種類	説明
DREQ<2:0>	入力	<p>外部コンパニオン・チップ要求</p> <p>外部コンパニオン・チップは、DMA 転送要求が必要なときに、DREQ 信号をアサートします。DMA コントローラは、ローからハイへの移行を登録して、新しい要求を識別します。DMA コントローラがローからハイへの移行を認識できるように、信号は 4 MEM_CLK サイクルの間アサート状態が維持されなければなりません。アサートが解除された後は、少なくとも 4 MEM_CLK サイクルの間アサート解除の状態が維持されなければなりません。</p> <p>外部コンパニオン・チップは、データ転送の完了まで待機しなくても、次の要求をアサートできます。このコンパニオン・チップでは、各 DREQ<2:0> ピンに最大で 31 の実行待ち要求を保持することが可能です。実行待ち要求の件数は、専用ステータス・レジスタ DRQSRx に記録されます。</p> <p>DREQ<1:0> ピン上の要求は、フライバイ・モードとフロースルー・モードのいずれにおいてもデータ転送に使用されます。DREQ<2> ピン上の要求は、フロースルー・モードでのみデータ転送に使用されます。</p>
DVAL<1:0>	出力	<p>有効外部コンパニオン・チップ</p> <p>メモリ・コントローラは、DVAL をアサートして、コンパニオン・チップに通知します。データを制御する必要があるか、または出力データが有効です。</p>

2.4 CompactFlash のツール IDE 規格

「CF+ and CompactFlash Specification, Revision 3.0」¹ では、HDD への接続に利用可能な 3 種類の動作モードが定義されています。

- PC カード ATA 使用 I/O モード
- PC カード ATA 使用メモリモード
- ツール IDE モード

エンベデッド製品で 사용되는パラレル ATA HDD のほとんどは、ツール IDE モードで動作します。このモードには、ローコスト、ATA HDD の互換性、優れたパフォーマンスといった利点があります。表 3 に、ツール IDE モードで 사용되는 CompactFlash 信号を示します。

1. CompactFlash およびツール IDE 規格の詳細については、「CF+ and CompactFlash Specification Revision 3.0」を参照してください。CompactFlash Association の Web サイト <http://www.compactflash.org/> (英語) からダウンロードできます。

表 3. 主要な CompactFlash トゥルー IDE 信号の一覧

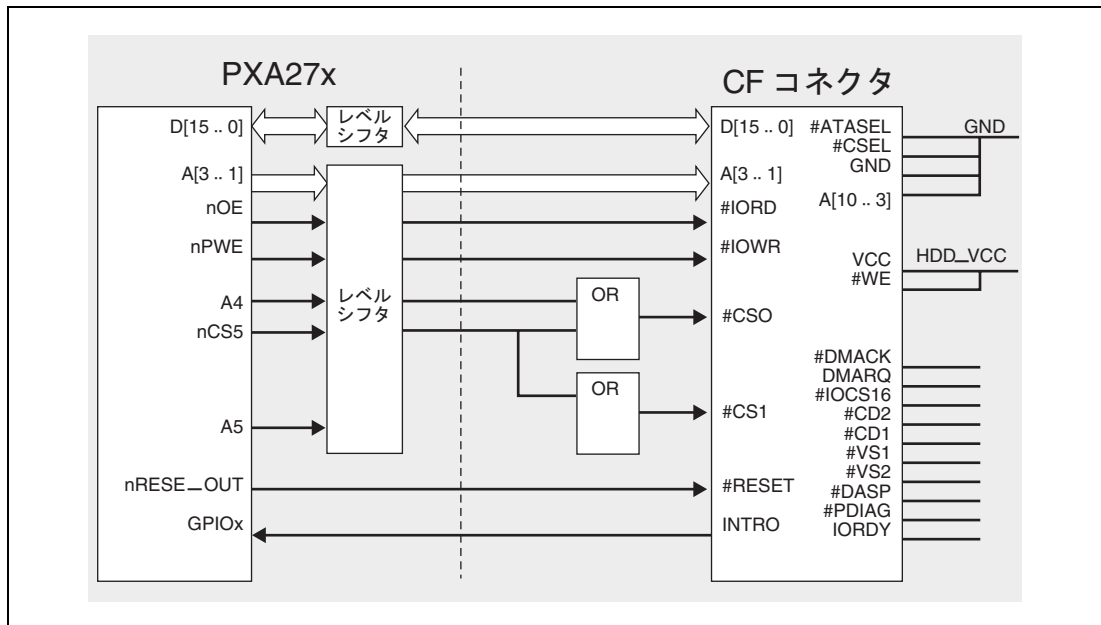
信号名	方向	説明
D<0:15>	双方向	データライン
A<0:2>	入力	HDD アドレスライン <0:2>
IORD#	入力	I/O 読み出しストロブを制御
IOWR#	入力	I/O 書き込みストロブを制御
ATA SEL#	入力	トゥルー IDE モードを有効化
IOCS16#	出力	16 ビット転送を指定
DMARQ#	出力	DMA 要求
DMACK	入力	DMA 肯定応答
INTRQ	出力	割り込み要求
Reset#	入力	リセット
CSEL#	入力	ケーブル選択
CS#<0:1>	入力	チップ選択信号
IORDY	出力	入出力が準備完了

3.0 インテル® PXA27x プロセッサの VLIO インターフェイスと トゥルー IDE CompactFlash インターフェイスとのマッピング

このセクションでは、PXA27x プロセッサの VLIO インターフェイスに対する、トゥルー IDE モードでの CompactFlash カードの物理/論理インターフェイスについて説明します。本書に掲載されているデバイスドライバのサンプルコードでは、CompactFlash HDD デバイスがトゥルー IDE モードで接続されているという想定です。電源投入シーケンス中に ATASEL# (OE#) 入力信号がホストによって接地された場合のみ、CompactFlash カードはトゥルー IDE モードに設定されます。

CompactFlash のトゥルー IDE モードの場合、本書で扱われている設計は、PIO モード 4 および DMA モード 2 で動作するように設計されていますが、Ultra DMA やその他の動作モードでは動作しません。図 2 に、トゥルー IDE PIO モードにおける PXA27x プロセッサの VLIO インターフェイスと CompactFlash との接続のブロック図を示します。

図 2. トゥルー IDE PIO モードのブロック図



3.1 インテル® PXA27x プロセッサへの CompactFlash HDD の接続

本書では、HDD は、3.3V のロジックを使用していると想定しています。一方、PXA27x プロセッサは 1.8V のメモリバスを使用しているため、HDD インターフェイス・ロジックとの間でロジックをバッファするのにレベルシフタが使用されていると想定しています。

以下の各項目では、PXA27x プロセッサに対する CompactFlash トゥルー IDE 信号のマッピングについて説明します。

- **HDD D15～D0**: HDD のデータラインは、PXA27x プロセッサの D0～D15 ラインに接続されます。
- **HDD A2～A0**: すべての VLIO デバイスは PXA27x プロセッサのメモリ・コントローラとやりとりする際に 16 ビット以上のバス幅を必要とするため、A[2:0] は PXA27x プロセッサの A[3:1] に接続されます。PXA27x プロセッサの A0 によるバイト選択は必要ありません。
- **HDD A10～A3**: HDD のこれらのアドレスラインは接地されます。
- **HDD IORD#**: CompactFlash カードからの I/O 読み出しストロブ・ラインは、PXA27x プロセッサのメモリ・コントローラの nOE に接続されます。
- **HDD IOWR#**: CompactFlash カードからの I/O 書き込みストロブ・ラインは、PXA27x プロセッサのメモリ・コントローラの nPWE に接続されます。
- **HDD RESET#**: CompactFlash カードからのリセットラインは、PXA27x プロセッサの nReset_Out またはパワーオン・リセット回路に直接接続され、CompactFlash デバイスをリセットします。
- **HDD ATASEL#**: この信号を接地すると、トゥルー IDE モードが指定されます。
- **HDD CSEL#**: この信号を接地すると、接続可能なドライブ（マスタドライブ）が 1 つしかないことを示せます。
- **HDD WE#**: この信号は VCC に接続されます。

- **HDD DMARQ、DMACK#**：これらの信号は、PXA27x プロセッサと HDD の間の DMA 転送に固有のものです。トゥルー IDE PIO モードでは使用されません。PIO モードの場合、HDD DMACK# は接続しないでおくか、VCC にプルアップします。これらの信号の詳細と、未使用の信号の処理については、「CF+ and CompactFlash Specification」を参照してください。トゥルー IDE マルチワード DMA モードの場合、信号のロジックと接続の詳細については、本書のセクション 3.2 を参照してください。
- **HDD IOCS16#、CD2#、CD1#、IORDY、DASP#、PDIAG#、VS1#、VS2#**：これらの信号は使用されません。
- **HDD CS0#、CS1#**：本書で扱われている設計では、PXA27x の 2 本のアドレスラインと nCS5 を利用して、HDD に必要な 2 つのチップ選択信号を取得します。HDD の CS0# 信号および CS1# 信号に対する、PXA27x プロセッサの nCS5 のメモリ・マッピングについては、表 4 を参照してください。

表 4. メモリ・マッピング

nCS55	A4	A5	CS0#	CS1#	R/W フィールド	アドレス
0	0	1	0	1	タスク・ファイル・レジスタ	0x1400_0020
0	1	0	1	0	CTL レジスタ	0x1400_0010
0	1	1	1	1	DMA	0x1400_0030
1	X	X	1	1	該当せず	該当せず

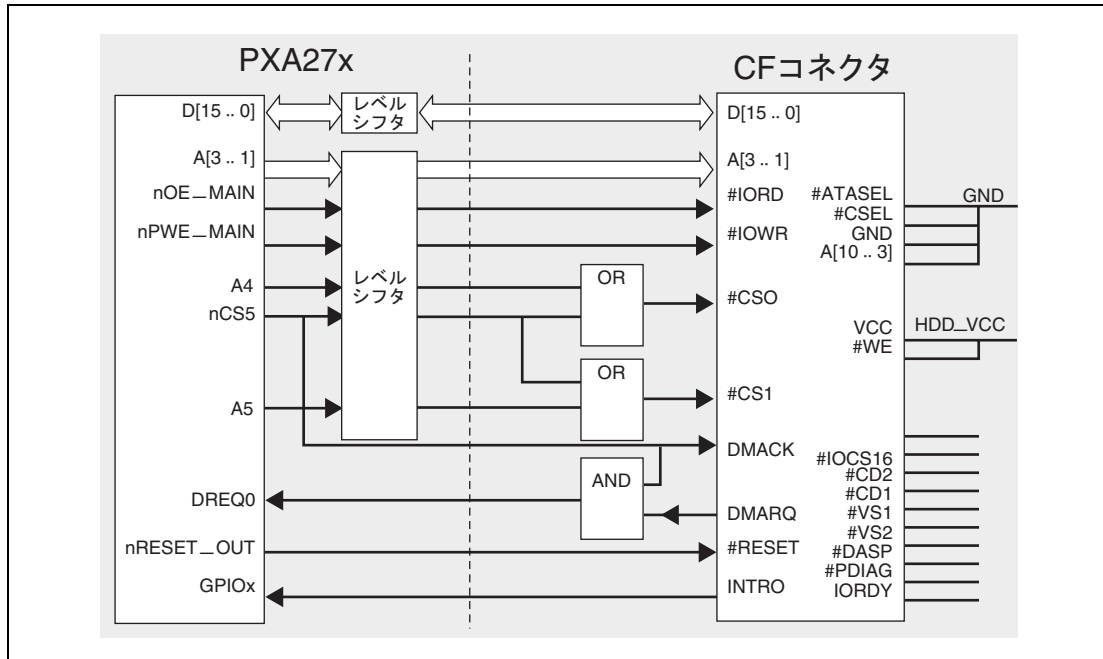
3.2 トゥルー IDE マルチワード DMA モードのブロック図

PXA27x DMA コントローラのフロースルー・モードは、PXA27x プロセッサと HDD の間のデータ転送に利用できます。ただし、DMA サポートの実装に当たっては、以下の 2 つの課題を解決しなければなりません。

1. PXA27x プロセッサの DMA 信号はエッジトリガされるが、HDD の DMA 信号はレベルトリガされます。この課題を解決するために、ブロック図では AND ゲートを追加して、PXA27x プロセッサの DREQ0 信号に向かう DMARQ ライン信号と、nCS5 信号とを組み合わせています。
2. PXA27x プロセッサの DMA コントローラがフロースルー・モードで動作する場合、DMA 肯定応答信号が生成されません。この課題を解決するために、HDD に必要な DMACK 信号として nCS5 ピンを使用しています。

図 3 に、トゥルー IDE DMA モードにおける PXA27x プロセッサの VLIO インターフェイスと CompactFlash との接続のブロック図を示します。

図 3. トゥルー IDE DMA モードのブロック図



3.3 タイミングに関する考慮事項

CompactFlash の規格では、CompactFlash トゥルー IDE の PIO モード 4 および DMA モード 2 に関するタイミング要件が定義されています。詳細については、「CF+ and CompactFlash Specification」の第 4 章を参照してください。タイミングの詳細な比較は、『Intel® PXA27x Processor Family Developer's Manual』、『Intel® PXA27x Processor Electrical, Mechanical, and Thermal Specification』、HDD のドキュメントを参考にして行ってください。

PXA27x VLIO インターフェイスは、フラッシュや SRAM などの高速メモリデバイスとやりとりするように設計されています。そのため、一部のタイミング・パラメータについては特別な考慮が必要です。いくつかの HDD ベンダのデバイスを分析した結果によると、PXA27x プロセッサで指定された制限には「CS Valid to -IORD/-IOWR」と「Write Data Hold」が近いことが判明しています。システム設計者は、それぞれ独自のタイミング分析とコンポーネント選択を行ってください。場合によっては、グルーロジックのタイミングを調整する必要があります。または、タイミング要件が異なる別の HDD を選択したり、わずかなタイミングの差が許容されるかどうか HDD メーカーに確認する必要があります。

3.4 インテル® PXA27x プロセッサのレジスタ設定

3.4.1 MSC レジスタ設定

PXA27X nCS5 を IDE インターフェイスに接続するには、MSC2 レジスタを設定してトゥルー IDE インターフェイスのタイミング要件を満たす必要があります。MSC2 レジスタは nCS5 および nCS4 を対象にしており、nCS5 を設定するには、同レジスタの上位半分のワードのみ修正が必要です。

- ビット 31[RBUFF] : 低速のメモリデバイスを指定するには、このビットを設定します。
- ビット 30 ~ 28 [RRR] : 読み出し/書き込み後に nCS がアサート解除されたときから、nCS がアサートされるまでの間隔を定義します。メモリクロックが 104MHz の場合はこのフィールドを 0b0001 に設定し、208MHz の場合は 0b0010 に設定します。
- ビット 27 ~ 24 [RDN] : MSCx[RDN] フィールドは、以降の読み出しアクセス時間を制御し、nPWE/nOE でのアドレスホールドのためにフラッシュ・メモリをバーストします。メモリクロックが 104MHz の場合はこのフィールドを 0b0010 に設定し、208MHz の場合は 0b0100 に設定します。
- ビット 23 ~ 20 [RDF] : MD/DQM の場合、書き込み間の nPWE のアサート解除期間と nPWE/nOE のアサート期間に設定します。メモリクロックが 104MHz の場合はこのフィールドを 0b0111 に設定し、208MHz の場合は 0b1100 に設定します。
ビット 19 : 16 ビットのデータバス幅を指定するには、このビットを 1 に設定します。
- ビット 16 ~ 18 : VLIO デバイスを指定するには、このフィールドを 0b100 に設定します。

例 1. MSC2 レジスタを設定するコードの例

MSC2 レジスタを設定するには :

```
unsigned long msc2=MSC2;
msc2&=0x0000ffff;

#ifdef VLIO_MEM208M
    msc2|=0 << 31 |
        1 << 28 |
        2 << 24 |
        7 << 20 |
        1 << 19 |
        4 << 16;
#else
    msc2|=0 << 31 |
        2 << 28 |
        4 << 24 |
        14 << 20 |
        1 << 19 |
        4 << 16;
#endif
MSC2= msc2;
```

3.4.2 インテル® PXA27x プロセッサの GPIO レジスタ設定

VLIO 信号および DMA 信号の要件を有効にするには、以下の GPIO を設定する必要があります。

- DREQ0 GPIO 20 : DMA 要求信号として設定する必要があります。このピンを入力および代替機能 1 に設定します。
- RDY GPIO 18 : 入力および代替機能 1 に設定します。
- nCS5 GPIO 33 : 出力および代替機能 2 に設定します。
- nPWE GPIO 49 : 出力および代替機能 2 に設定します。

例 2. GPIO レジスタを設定するコードの例

GPIO レジスタを設定するには :

```
// pin 18 19
pins[0]=2;fn[0]=2;
pins[1]=XLLP_GPIO_RDY;
pins[2]=XLLP_GPIO_MBREQ;
fn[1]=XLLP_GPIO_ALT_FN_1;
fn[2]=XLLP_GPIO_ALT_FN_1;
XllpGpioSetDirectionIn((P_XLLP_GPIO_T)v_pGPIOReg, pins);
XllpGpioSetAlternateFn((P_XLLP_GPIO_T)v_pGPIOReg, pins, fn);
// pin 33 49
pins[0]=2;fn[0]=2;
pins[1]=XLLP_GPIO_nPWE;
pins[2]=XLLP_GPIO_nCS5;
fn[1]=XLLP_GPIO_ALT_FN_2;
fn[2]=XLLP_GPIO_ALT_FN_2;

XllpGpioSetOutputState1((P_XLLP_GPIO_T)v_pGPIOReg, pins);
XllpGpioSetDirectionOut((P_XLLP_GPIO_T)v_pGPIOReg, pins);
XllpGpioSetAlternateFn((P_XLLP_GPIO_T)v_pGPIOReg, pins, fn);
```

4.0 HDD デバイスドライバ

4.1 HDD ドライバの概要

トゥルー IDE HDD ドライバは、Linux* および Microsoft* Windows* CE 向けのブロックドライバとして OS に登録されます。いずれの OS でも、ドライバは読み出し/書き込みと IOCTL 処理を実装する必要があります。このドライバでは、PIO モードと DMA モードの両方がサポートされており、上位層アプリケーションへの読み出し/書き込みインターフェイスは同一です。PIO モードの場合、HDD インターフェイスから読み出されるデータと、HDD インターフェイスに書き込まれるデータはすべて、CPU コアが直接処理します。そのため、HDD のデータにアクセスする際は、CPU の使用率がきわめて高くなります。DMA モードの場合、CPU コアは DMA バッファとディスクリプタの準備にしか関与しません。DMA コントローラが SDRAM と HDD の間のデータ転送を担当するため、CPU コアを解放してその他のシステムタスクの処理に割り当てられます。

4.2 HDD 転送モードの設定

ドライバ初期化の最初の段階で、HDD 転送モードを PIO モード 4 や DMA モード 2 などに設定します。例 3 に、SET_FEATURE(0xEF) コマンドを使用して HDD 転送モードを設定する方法を示します。

例 3. HDD 転送モードを設定するコードの例

HDD 転送モードを設定するには：

```
#define ATA_CMD_SETFEATURES 0xEF
#define SETFEATURES_XFER 0x03
#define XFER_MW_DMA_2 0x0c
#define XFER_PIO_4 0x22
//set transfer modes
    VLIO_WRITE_PORT_UCHAR(TRUE_IDE_ATA_REG_DRV_HEAD, 0xa0);
#ifdef  VLIODISK_PIOMODE4
    VLIO_WRITE_PORT_UCHAR(TRUE_IDE_ATA_REG_SECT_CNT, XFER_PIO_4);
#else
    VLIO_WRITE_PORT_UCHAR(TRUE_IDE_ATA_REG_SECT_CNT, XFER_MW_DMA_2);
#endif
    VLIO_WRITE_PORT_UCHAR(TRUE_IDE_ATA_REG_FEATURE, SETFEATURES_XFER);
    VLIO_WRITE_PORT_UCHAR(TRUE_IDE_ATA_REG_COMMAND, ATA_CMD_SETFEATURES);
```

4.3 I/O アドレス・マッピング

nCS5 の場合、物理アドレスは、0x14000000 から始まる 64MB スロットです。この物理 I/O 領域にフル機能でアクセスするには、この領域を仮想アドレス空間にマッピングする必要があります。

4.4 PIO モードと割り込み処理

トゥルー IDE HDD 割り込みラインでは、高度なトリガ処理が行われます。PIO モードの場合、割り込みは以下のようにして発生します。

読み出し：

1. まずホストがパラメータを HDD タスク・ファイル・レジスタにセットしてから、読み出しコマンドを送信します。
2. ホストは HDD 割り込みの発生を待機します。
3. HDD は、内部バッファ内にデータを準備できると、割り込みラインをプルアップして割り込みをトリガします。
4. ホストは HDD から通知を受け取ると、割り込みモードにトラップし、メモリバスを介して HDD からデータを読み出します。
5. すべてのデータの読み出しが終わると、HDD は割り込みとステータスをクリアします。

書き込み：

1. まずホストがパラメータを HDD タスク・ファイル・レジスタにセットしてから、書き込みコマンドを送信します。
2. HDD は、データを受け取る準備が整うと、DRQ ステータスをセットします。
3. ホストがデータを HDD に書き込みます。
4. HDD は、必要なデータをすべて受け取ると、割り込みをトリガします。

5. ホストは HDD から通知を受け取ると、割り込みモードにトラップし、ステータスをクリアします。

4.5 DMA モードと割り込み処理

DMA モードの場合、割り込みは以下のようにして発生します。

読み出し：

1. まずホストがパラメータを HDD タスク・ファイル・レジスタにセットし、DMA とそのディスクリプタを準備してから、読み出しコマンドを送信します。
2. HDD は、nDMARQ をアサートすることによって DMA を要求します。
3. ホスト DMA コントローラは、nDMACK 応答を返して、メモリバスからデータを読み出します。
4. すべてのデータの読み出しが終わると、HDD は割り込みをトリガします。
5. ホストは HDD から通知を受け取ると、割り込みモードにトラップし、割り込みとステータスをクリアします。

書き込み：

1. まずホストがパラメータを HDD ファイルレジスタにセットし、DMA とそのディスクリプタを準備してから、書き込みコマンドを送信します。
2. HDD は、nDMARQ をアサートすることによって DMA を要求します。
3. ホスト DMA コントローラは、nDMACK 応答を返して、データを HDD に書き込みます。
4. すべてのデータを受け取ると、HDD は割り込みをトリガします。
5. ホストは HDD から通知を受け取ると、割り込みモードにトラップし、割り込みとステータスをクリアします。

4.6 高度な機能の設定

書き込みキャッシュの有効化

ほぼすべての HDD は内部書き込みバッファを備えています。デフォルトではバッファが常に無効になっています。書き込みキャッシュを有効にするには、機能パラメータ 2 を指定して SET_FEATURE(0xEF) コマンドを実行します。

例 4. 書き込みキャッシュを有効化するコードの例

書き込みキャッシュを有効にするには：

```
#define ATA_CMD_SETFEATURES 0xEF
#define SETFEATURES_EN_WCACHE 0x02

VLIO_WRITE_PORT_UCHAR(TRUE_IDE_ATA_REG_DRV_HEAD, 0xa0);
VLIO_WRITE_PORT_UCHAR(TRUE_IDE_ATA_REG_FEATURE, SETFEATURES_EN_WCACHE);
VLIO_WRITE_PORT_UCHAR(TRUE_IDE_ATA_REG_COMMAND, ATA_CMD_SETFEATURES);
```

パワー・マネジメント (PM) の有効化

パワー・マネジメント (PM) 機能を利用すると、HDD は、一定秒数アイドル状態が続いた場合に、自動的にスタンバイ・ステータスに移行します。

PM 機能を有効にするには、Sector Count レジスタを指定して ATA_CMD_SET_IDLE コマンドを実行し、スタンバイ・ステータスに移行するまでのアイドル秒数 (*5) を設定します。

例 5. パワー・マネジメントを有効化するコードの例

```
#define ATA_CMD_SET_IDLE 0xE3
VLIO_WRITE_PORT_UCHAR(TRUE_IDE_ATA_REG_DRV_HEAD, 0xa0);
VLIO_WRITE_PORT_UCHAR(TRUE_IDE_ATA_REG_SECT_CNT, 2); // set 2*5 seconds to idle mode
VLIO_WRITE_PORT_UCHAR(TRUE_IDE_ATA_REG_COMMAND, ATA_CMD_SET_IDLE);
```

複数セクタの設定

PIO モードの場合、デフォルトでは、各セクタの読み出し/書き込みのたびに割り込みが 1 回トリガされますが、このような割り込みは必要ありません。ATA 仕様では、複数セクタを定義して効率化を図っています。このコマンドを利用すると、複数のセクタが単一の割り込みをトリガするように設定できます。

例 6. 複数セクタを設定するコードの例

複数セクタを設定するには：

```
VLIO_WRITE_PORT_UCHAR(TRUE_IDE_ATA_REG_SECT_CNT, pDisk->d_MaxMultiSectors);
VLIO_WRITE_PORT_UCHAR(TRUE_IDE_ATA_REG_COMMAND, ATA_CMD_SETMULTI);
```

パラメータ MaxMultiSectors は、HDD のアイデンティティ・データにあります。

4.7 Linux ドライバに関する考慮事項

Linux ドライバを作成する際は、以下の課題を解決する必要があります。

- I/O アドレスのマッピング
- DMA チャンネルの要求
- 割り込みの要求
- DMA バッファおよびディスクリプタの割り当て

4.7.1 I/O アドレス・マッピング

Linux では、mmap api を利用して、物理アドレスを仮想空間にマッピングできます。または、VLIO アドレス空間を、arch/arm/mach-pxa/generic-bvd.c ファイルにあるシステムの静的マッピング・テーブルに追加するという選択肢もあります。

例 7. 静的マッピング・テーブルの例

VLIO アドレス空間が追加された静的マッピング・テーブルの例を以下に示します。

```
static struct map_desc standard_io_desc[] __initdata = {
    /* virtual    physical    length    domain    r w c b */
    { 0xf5000000, 0x14000000, 0x01000000, DOMAIN_IO, 0, 1, 0, 0 }, /* VLIO IO */
    { 0xf6000000, 0x20000000, 0x01000000, DOMAIN_IO, 0, 1, 0, 0 }, /* PCMCIA0 IO */
    { 0xf7000000, 0x30000000, 0x01000000, DOMAIN_IO, 0, 1, 0, 0 }, /* PCMCIA1 IO */
    { 0xf8000000, 0x40000000, 0x01800000, DOMAIN_IO, 0, 1, 0, 0 }, /* Devs */
    { 0xfa000000, 0x44000000, 0x00100000, DOMAIN_IO, 0, 1, 0, 0 }, /* LCD */
    { 0xfc000000, 0x48000000, 0x00100000, DOMAIN_IO, 0, 1, 0, 0 }, /* Mem Ctl */
    { 0xfe000000, 0x4c000000, 0x00100000, DOMAIN_IO, 0, 1, 0, 0 }, /* USB host */
    { 0xff000000, 0x00000000, 0x00100000, DOMAIN_IO, 0, 1, 0, 0 }, /* UNCACHED_PHYS_0 */
    LAST_DESC
};
```

この例の 1 行目には、新しく追加された VLIO デバイスの MMU 属性が記載されています。

4.7.2 DMA チャンネルの要求

例 8. DMA チャンネル要求の例

```
vlio_dma_channel=pxa_request_dma("VLIODMA",DMA_PRIO_HIGH, vlio_dma_finished,
NULL);
```

4.7.3 IRQ の要求

例 9. IRQ 要求の例

```
request_irq(hwif->irq,&ide_intr,sa,hwif->name,hwgroup)
```

4.7.4 DMA バッファおよびディスクリプタの割り当て

例 10. DMA バッファおよびディスクリプタの割り当ての例

```
consistent_alloc( GFP_KERNEL, VLIO_DMA_DESC_NUM * sizeof(pxa_dma_desc), (void *)&vlio_
dma_descriptors_physical, 0);
```

4.8 Microsoft* Windows* CE ドライバに関する考慮事項

Linux ドライバの場合と同様に、Microsoft* Windows* CE ドライバを作成する際も、以下の課題を解決する必要があります。

- I/O アドレスのマッピング
- DMA チャンネルの要求
- 割り込みの要求
- DMA バッファおよびディスクリプタの割り当て

4.8.1 I/O アドレス・マッピング

例 11. VLIO アドレス空間のマッピング

Windows* CE では、VLIO アドレス空間を以下のようにマッピングできます。

```
v_VlioBase = (char *)MmMapIoSpace(ioPhysicalBaseVlio,0x100,FALSE);
```

4.8.2 DMA チャンネルの要求

例 12. DMA チャンネル要求の例

```
XllpDmacAllocChannel(&v_DMACHannel, XLLP_DMACHANNEL_PRIORITY_HIGH)
```

4.8.3 IRQ の要求

例 13. IRQ 要求の例

```
InterruptInitialize(SYSINTR_EXPBD, pDisk->d_IRQEvent, NULL, 0)
```

4.8.4 DMA バッファおよびディスクリプタの割り当て

例 14. DMA バッファおよびディスクリプタの割り当ての例

```
v_VlioDMADescVirtual = (char *) HalAllocateCommonBuffer(&Adapter, TOTAL_BUF_SIZE, &PA,
FALSE);
```

5.0 まとめ

本書では、PXA27x プロセッサを CompactFlash トゥルー IDE モードの HDD に接続する方法についての詳細と推奨事項を紹介しました。PIO モードおよび DMA モードでの HDD 接続のほか、Linux および Windows* CE での HDD ドライバの作成についても説明しています。

付録 A：略語

表 5. 略語の定義

略語	説明
ATA	Advanced Technology Attachment
CF	CompactFlash
CS	チップ選択
DMA	Direct Memory Access
DRQ	データ要求
GPIO	汎用 I/O
HDD	ハードディスク・ドライブ
IDE	Integrated Device Electronics
PDA	Personal Digital Assistant
PIO	Programmed I/O
PM	パワー・マネジメント
VLIO	Variable Latency Input/Output



インテル株式会社

〒300-2635 茨城県つくば市東光台 5-6
<http://www.intel.co.jp/>

© 2005, Intel Corporation. 無断での引用、転載を禁じます。