



Technical notes on using Analog Devices DSPs, processors and development tools  
 Visit our Web resources <http://www.analog.com/ee-notes> and <http://www.analog.com/processors> or  
 e-mail [processor.support@analog.com](mailto:processor.support@analog.com) or [processor.tools.support@analog.com](mailto:processor.tools.support@analog.com) for technical support.

## SPIによるSHARC®とBlackfin®間接続

寄稿者: Jeyanthi Jegadeesan, Andreas Pellkofer

Rev 1 – July 8, 2008

### はじめに

この EE ノートには、シリアル・ペリフェラル・インターフェース (SPI) を介してアナログ・デバイセズの Blackfin®プロセッサを SHARC®プロセッサに接続する方法を示しています。

主な内容:

- SPI の説明
- 物理層セットアップの説明
- SPI の設定
- Blackfin プロセッサ SPI インターフェースのプログラミング・モデル
- SHARC プロセッサ SPI インターフェースのプログラミング・モデル
- コード例 (別ファイルで提供)

### 動機

今日の組込みシステムは複数のプロセッサを必要とするケースが多々あり、それぞれが特定のアプリケーションを受け持ちます。アナログ・デバイセズの Blackfin と SHARC プロセッサの両方を用いる代表的なシステムとして、オーディオ環境に関連するものがあります。

SHARC プロセッサにとっての代表的なアプリケーションは、ダイナミックレンジの広い、高性能な浮動小数点オーディオ処理を必要とするものです。DSP とマイクロコントローラの両方を備えた Blackfin プロセッサは、たとえば次のように SHARC プロセッサを制御するホストとして使用できます。

- SHARC プロセッサのブート (SPI スレーブのブート)
- メッセージの送信 (低音、高音、音量、フェーダー)

図 1に、マルチプロセッサ・システムの信号フローの一例を示します。

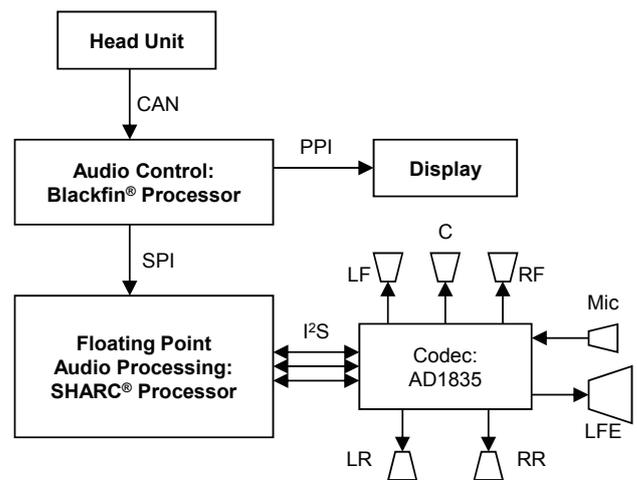


図 1. システム・セットアップ

### SPIについて

SPI®は、モトローラが命名した業界標準の同期シリアル・データ・リンクです。この標準は、全仕様が規定されてはいません。仕様にはハードウェアは含まれますが、ソフトウェア・プロトコルは含まれていません。

SPI は、複数の SPI 互換デバイスとの通信に対応しています。I<sup>2</sup>C のようなほかのシリアル・インターフェースとは違って、SPI ペリフェラルは 4 線式インターフェースであり (I<sup>2</sup>C は 2 線式)、2 本のデータ・ピン (MOSI、MISO) と 1 本のデバイス・セレクト・ピン (/SPISS)、1 本のゲーテッド・クロック・ピン (SCK) で構成されています。2 つのデータ・ピンで、ほかの SPI 互換デバイスとの全二重通信が可能となります。また、SPI にはプログラマブル・ボーレート、クロック位相、クロック極性もあります。デバイスはマスタ/スレーブ・モードで通信を行い、マスタ・デバイスがデータ・フレームを

開始します。複数のスレーブ・デバイスは、個別のスレーブ/チップ・セレクト・ラインを必要とします。

SPI の限定的機能を担うサブセットはナショナル・セミコンダクタ社の Microwire™ (μWire) であり、SPI をベースにしているため、これらは互換性があります。この EE ノート内の SPI はモトローラの SPI を指しています。

代表的な SPI 互換のペリフェラル・デバイス:

- マイクロコントローラ
- コーデック
- A/D コンバータ、D/A コンバータ
- センサ
- フラッシュ・メモリ・デバイス
- SP/DIF および AES/EBU デジタル・オーディオ・トランスミッタ/レシーバ
- LCD ディスプレイ

Blackfin および SHARC プロセッサ上の SPI インターフェースは以下の特長を備えています。

- 全二重同期シリアル・インターフェース
- 8 ビット/16 ビット・ワード・サイズ (Blackfin プロセッサ)
- 32 ビット・ワード・サイズ (SHARC プロセッサ)
- リトル・エンディアンまたはビッグ・エンディアン・フォーマット
- プログラマブル・ポーレート、クロック極性、位相
- マスタ・スレーブ、マルチマスタ・モード
- オープン・ドレイン出力: データ競合に起因するドライバ競合を回避し、マルチマスタ・シナリオに対応
- SPI デバイスからのマスタ/スレーブブート
- DMA 機能: コア・オーバーヘッドなしにデータ転送を実現

図 2 と 図 3 に SPI ブロック図を示します。

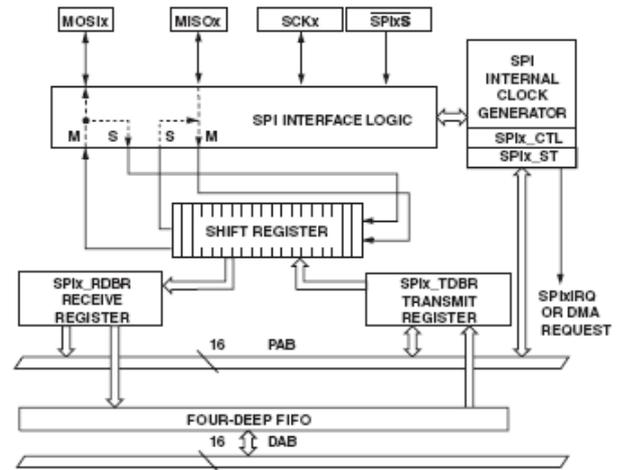


図 2. Blackfin SPI のブロック図

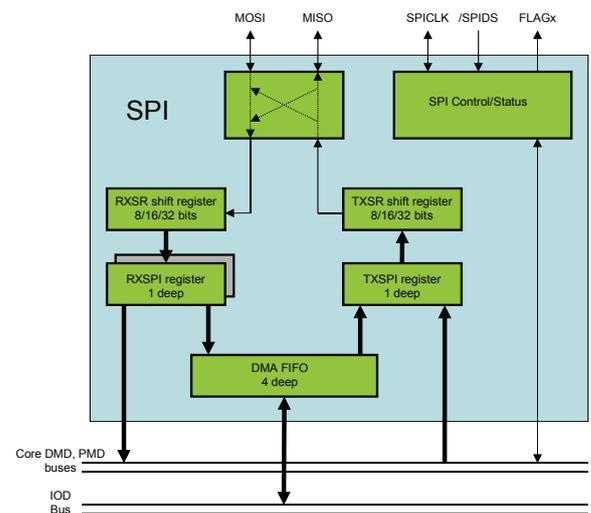


図 3. SHARC SPI のブロック図

詳細については、Blackfin プロセッサ派生製品および SHARC プロセッサ派生製品の各『Hardware Reference』マニュアル (HRM) の「SPI」を参照してください。

## 物理層のセットアップ

図 4に、SHARC SPIスレーブ・デバイスとBlackfin SPI マスタ・デバイスに関する信号の接続と方向を示します。

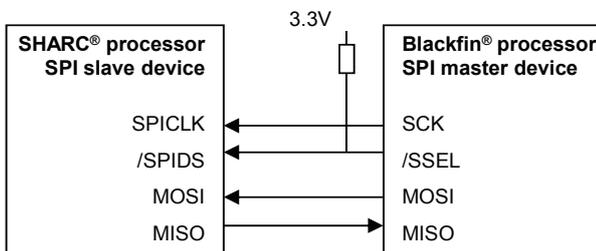


図 4. マスタとスレーブの接続

SPI マスタ・デバイスは、必ずクロックとスレーブ・セレクト・ライン信号の両方を駆動します。SPI スレーブ・デバイスは、クロックとアクティブ・スレーブ・セレクト・ライン信号の両方が存在するときだけアクティブになります。

この2つのプロセッサの SPI マスタ・デバイスと SPI スレーブ・デバイスで異なる点は、SPI 制御レジスタの1ビット (MSTR) です。

マスタ SPI デバイスの場合、/SPISS ピンはマルチマスタ環境においてエラー信号入力として使用できます。1個のプルアップ抵抗の接続を推奨します。

SPI スレーブ・セレクト・イネーブル出力信号 (/SSEL) は、SPI プロトコルで必ずアクティブ・ローになります。各ピンはリセット中に駆動されないため、抵抗でプルアップすることを推奨します。



Blackfin プロセッサは、SCK ライン対応のプルアップ抵抗が必要です。SHARC プロセッサの DPI は SPICLK ピンに内部プルアップを提供するので、ここでは外部プルアップは不要です。

## Blackfin プロセッサのプログラミング・モデル

Blackfin プロセッサの SPI インターフェースのプログラミング・モデルについては、Blackfin 派生製品の『Hardware Reference』マニュアルで説明しています。ADSP-BF537 HRM に示した SPI のフローチャートは、SPI 転送のセットアップ用のリファレンスとして利用できます。

このセクションは、プログラミング・モデルについて特に詳しく説明しており、現行文書の付録および要約として活用してください。



この EE ノートは、シングル・マスタ/スレーブ環境に焦点を当てており、ADSP-BF537 プロセッサは SPI マスタとして機能し、ADSP-21369 プロセッサは SPI スレーブ・デバイスとして機能します。ここに示す例は、この2つのプロセッサをベースにした他のセットアップにも適用できます。

## 準備

ほとんどの Blackfin プロセッサは、ピン数を減らすためにピン多重化を利用しています。ADSP-BF537 プロセッサでは、SPI 信号のほとんどがポート F 経由でアクセスできます。最も重要な5つの信号 (SCK、MISO、MOSI、/SPISS、SPISEL1) はほかのペリフェラルと多重化されません。

ADSP-BF537 プロセッサの場合は、PORTF\_FER に書き込んで SPI 信号をイネーブルにする必要があります。デフォルトでは、GPIO 機能がイネーブルとなります。

SPISEL2~SPISEL7 の使用については、HRM の「SPI」と「汎用ポート」を参照してください。これらの信号は (必要な場合)、イネーブルにしてください。

ペリフェラルから、またはペリフェラルに対して転送終了を通知する場合は、割り込みを使用します。そうしない場合には、プロセッサから定期的にステータス・ビットをポーリングして転送終了を検出する必要があります。コアとシステム/ペリフェラル間のクロック比が大きいと (たとえば、SPI クロック = 1 MHz、コア・クロック = 500 MHz)、ポーリングはおそらく問題にはなりません。この間は、ほかの動作を実行することができます。

コア・イベント・コントローラとシステム割り込みコントローラ (SIC) も共に設定する必要があります。詳細については、『Blackfin Processor Programming Reference』マニュアル (PRM) の「プログラム・シーケンサ」と HRM の「システム割り込み」を参照してください。

## SPIの初期化

SPI インターフェースは、設定を行うためのシステム MMR (メモリ・マップド・レジスタ) の一連のレジスタ (SPI\_FLG、SPI\_BAUD、SPI\_CTL) で構成されています。

SPI ステータスは、SPI\_STAT レジスタから読み出すことができます。

コア駆動の転送の場合は、SPI 送受信データ・バッファ・レジスタ (SPI\_TDBR/SPI\_RDBR) に加えて SPI\_SHADOW が必要です。

### SPI\_FLGレジスタ

SPI マスタ・デバイスの場合、スレーブを選択するには、SPI\_FLG に値を書き込んで該当するスレーブ・セレクト・イネーブル (FLSx) ビットを設定する必要があります。

スレーブ・セレクト値 (FLGx) ビットで、スレーブ・セレクト・ラインに駆動される値が決まります。

CPHA = 1 (SPI\_CTL 内の CPHA ビット設定) の場合、出力値は FLGx ビットのソフトウェア制御によって設定されます。

CPHA = 0 の場合は、SPI ハードウェアが出力値を設定し、FLGx ビットは無視されます。つまり、ハードウェアが 1 ワードごとにスレーブの選択と選択解除 (/SSEL) を行います (図 5 と図 7 を参照)。

次の図は、SPI 経由の 16 ビット・ワードの転送を示しています。ここには最初の 2 ワードを示しています。Blackfin プロセッサはマスタであり、SPI クロックとスレーブ・セレクト・ラインを駆動します。SHARC プロセッサはスレーブとして機能し、マスタ SPI デバイスからデータを受信します。

図 7 と図 8 は送信のアクティブ・ロー SCK 版を示しています。この場合、クロックの極性は反転されています。

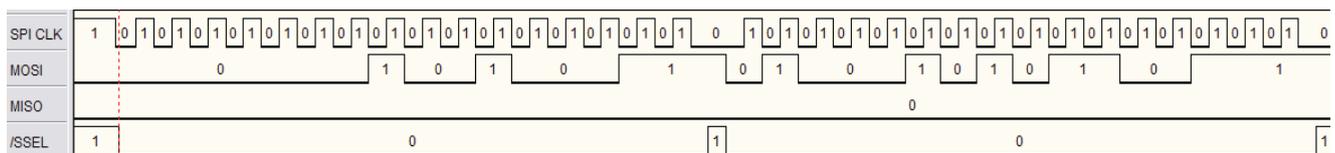


図 5. CPHA = 0、アクティブ・ハイ SCK (CPOL = 0)

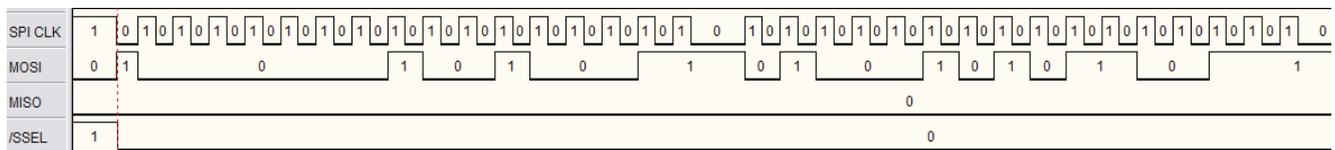


図 6. CPHA = 1、アクティブ・ハイ SCK (CPOL = 0)

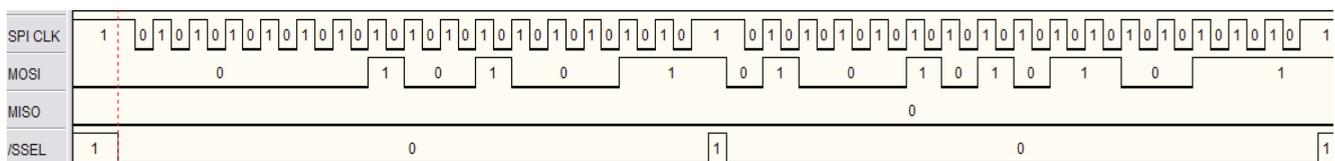


図 7. CPHA = 0、アクティブ・ロー SCK (CPOL = 1)

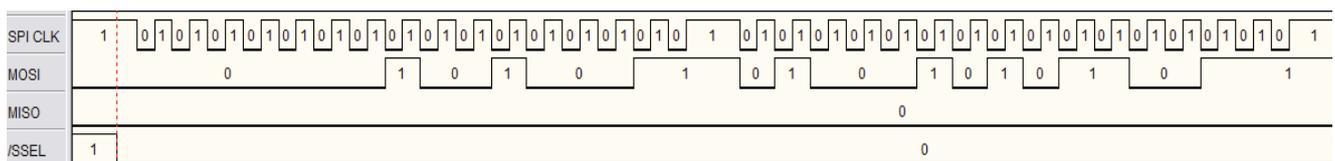


図 8. CPHA = 1、アクティブ・ロー SCK (CPOL = 1)

### SPI\_BAUD

SPI\_BAUD レジスタでは、SPI クロック周波数を設定できます。これは、Blackfin プロセッサのシステム・クロック (SCK) から生成されます。SCK を最大の 133 MHz にすると、これは仕様上、最大の SPI クロック周波数である約 33 MHz (SCK/4) に変換されます。これより高い周波数が必要な場合は、Blackfin プロセッサの SPORT インターフェースを使って SPI をエミュレートすることができます。SPORT クロックの最大周波数は SCK/2 なので、最高で約 66 MHz のクロック速度を達成できます。これについては EE-304 を参照してください<sup>[4]</sup>。

### SPI\_CTL

SPI\_CTL レジスタについては、プロセッサの HRM でも説明されています。

「SPIについて」でも触れたように、モトローラは SPI の (タイミング) プロトコルを明確に規定していないため、一般には 4 種類のモードが使用されています。これらのモードは、CPOL (クロック極性) と CPHA (クロック位相) の 2 ビットで設定できます。

現在の Blackfin プロセッサの SPI インターフェースは、SHARC SPI と異なり、32 ビット・データ・ワードをサポートしていません。

### SPI\_STAT

SPI ステータス・レジスタ (SPI\_STAT) は 2 つのビット・グループに分けることができます。最初のグループは、バッファ・アンダーラン、バッファ・オーバーフロー、マルチマスタ環境下の競合といった、エラーの検出用です。もう 1 つのビット・グループは、送信バッファまたは受信バッファのステータスと SPI ステータス (SPI の終了) を示します。

詳細については、これらのステータス・ビットの使用法を記した『SPI DMA 転送と割込みサービス』を参照してください。

### SPI DMA の初期化

SPI は、1 度に 1 送信動作または 1 受信動作を行う 1 本の専用 DMA チャンネルを備えています。

詳細については、HRM の「ダイレクト・メモリ・アクセス」を参照してください。

### SPI 転送開始

SPI マスタ・デバイスの場合、スレーブを選択するには、SPI\_FLG に書き込んで該当するスレーブ・セレクト・イネーブル (FLSx) ビットを設定する必要があります。

SPI DMA モードでは、初期化中に SPI インターフェースと DMA をイネーブルにはなりません。この両方を正しく設定したら、DMA (DMAEN=1)、SPI (SPEN=1) と順に開始します。

SPI コア・モードの場合、TIMOD=00 で実行していれば、初期化中に SPI をイネーブルにできます。TIMOD=01 であれば、SPI ビットを設定するとすぐに SPI が開始されます。SPI\_TDBR に書き込み、次に SPI をイネーブルにするか、または SPI 送信動作の TIMOD=00 も使用してください。最後のケースでは、SPI\_RDBR へのダミー読出しアクセスで転送が開始されます。

### SPI DMA 転送と割込みサービス

	DMA Tx	DMA Rx	Core Tx	Core Rx
DMA_RUN	1x	-	-	-
TXS	2x	-	1x	-
RXS	-	-	-	≥1x
SPIF	1x	-	1x	1x

表 1. ポーリングされるステータス・ビット

SPI DMA 動作の場合は、DMA 割込みの発生を待つ必要があります。それが発生すると、DMAx\_IRQ\_STATUS レジスタの DMA\_DONE ビットが設定されます。



DMA\_DONE 割込みは、最後のメモリ・アクセス (読出し/書込み) が完了するとアサートされます。パリティへの送信のために (メモリ読出し)、割込み発生時には、依然としてチャンネルの DMA FIFO 内に最大 4 データ・ワードのデータが存在する可能性があります。

SPI 受信または SPI 送信用のサービス・ルーチンは、最初に割込みソースをクリアする必要があります。この場合は、DMA\_DONE ビットに 1 を書き込みます。すなわち、WIC (1 を書き込んでクリア) を行います。そうしないと、割込みが再度ラッチされます。



アプリケーションは最後のデータがペリフェラルにいつ転送されたか知るために、DMA\_RUN ビットをテストまたはポーリングすることができます。FIFO 内に保留中のデータがある限り、DMA\_RUN ビットは 1 のままです。

DMAx\_IRQ\_STATUS レジスタの DMA\_RUN ビット (DMA\_RUN = 1) がロー (DMA\_RUN = 0) になるまでポーリングを続けます。



SPI 転送に DMA を使用するとき、DMA\_DONE 割込みは DMA FIFO が空であることを意味しますが、この時点では、まだ送信待ちのデータが PERIPHERAL (SPI) DMA FIFO 内に存在する可能性があります。したがって、ソフトウェアは SPI\_STAT レジスタの TXS が 2 連続読出しでローになるまで (その時点で SPI DMA FIFO は空になる)、TXS をポーリングする必要があります。

TXS ビットは転送バッファへの書込みが可能なときに定義されますが、SPI シフト・レジスタにはデータが残っている可能性があります。

SPIF ビットは、最後の SCK エッジ後の 1/2 SCK 期間で設定されます。そのときに、ワードの最終ビットがシフト出力され、SPI 転送が終了します。

RXS ビットは、受信バッファの読出しが可能なときに定義されます。1 ワード転送は RXS ビットの設定により終了し、この時点で新しいワードが受信されて受信バッファ (SPI\_RDBR) にラッチされていることがわかります。マスタ SPI の場合は、RXS が SCK の最後のサンプリング・エッジのすぐあとに設定されます。スレーブ SPI の場合は、CPHA や CPOL に関係なく RXS が最後の SCK エッジのすぐあとに設定されます。遅延は一般に数クロック (SCLK) サイクルであり、TIMOD やポーレートに左右されません。SPI\_RDBR が満杯のときに (TIMOD=00) 割込みを生成するよう設定している場合は、割込みが RXS 設定から 1SCLK サイクルの間アクティブになります。この割込みに依存していないときは、RXS ビットをポーリングして転送終了を検出できます。

割込みは DMA の最終メモリ書込み動作のあとに発生するので、DMA 受信動作にポーリングは不要です。DMA 送信動作の場合は、DMA FIFO と SPI DMA FIFO 間の遅延に注意する必要があります。

## SPI コア 転送

SPI コア 転送の場合も割込みを使用できます。しかし、SPI クロックとコア・クロックの比が 1 に近い場合は、割込みの使用はあまり意味がありません。割込みを 1 ワードごとに行うと、MIPS 性能が大幅に低下します。

したがって、データ転送のためにハードウェア・ループ (Blackfin のシーケンサ機能) をセットアップするのが賢いやり方です。このループ内では、各転送間でワードごとにポーリングする必要があります。

## SPI 転送 終了

SPI の終了手順は開始手順に似ています。SPI DMA 転送の場合は、逆順でコンポーネントを終了します。該当するスレーブ・セレクト・イネーブル (FLSx) ビットをクリアしてスレーブの選択を解除します。次に DMA を終了し、少なくとも SPI をディスエーブルにします。

また、エラー・ビットをクリアして (W1C)、SPI\_STATUS レジスタをリセットすることができます。RXS ビットをクリアするには、SPI\_RDBR からダミー読出しを行う必要があります。

## 要約: DMA と コア SPI の 転送

Blackfin SPI コントローラは 2 つの転送方法を提供します。DMA ベースの転送をセットアップするか、あるいはプロセッサ・コアを使って SPI\_TDBR レジスタまたは SPI\_RDBR レジスタにアクセスし、転送を駆動して、2 ビットの TIMOD (転送開始モード) でセットアップすることができます。

DMA ベースの転送では、プロセッサの負荷を低減できます (ほかの計算またはタスクは並列に実行できます)。転送 (DMA) 終了の確認が必要な場合は、割込みをイネーブルにします。そうしない場合は、DMA ステータス・ビットをプロセッサでポーリングする必要があります。

DMA ベースの転送は、特に大規模転送に適しています。

しかし、Blackfin プロセッサが DMA 受信モード (TIMOD=10) でマスタ (MSTR=1) のときは、DAM ベースの転送に1つのデメリットが生じます。この場合、最後のワードが受信されて転送を終了した場合でも SPI が SPI クロックとスレーブ・セレクト・ラインを駆動し、カレント・インナー・ループ・カウンタ・レジスタの値はゼロのままです (DMAx\_CURR\_X\_XOUNT=0)。こうなるのは、SPI および DMA が、ワード・カウントと同期していないためです。SPI\_RDBR への読出しアクセスがあると (DMA とコアのいずれかがこれを行っているかにかかわらず)、SPI は必ず実行終了 (SPE=0) となるまで MISO ライン上の新しいデータをフェッチします。

接続先のスレーブ側で転送されるワードがわからないか、スレーブ終了のためにカウンタが適宜動作していないか、あるいはその両方の場合、スレーブは必要以上のワードを転送するよう駆動される可能性もあり得ます。

**i** DMA での受信モード (TIMOD=10) では、SPI DMA FIFO にデータがある限り (FIFO が空ではないとき)、SPI はメモリへの DMA 書き込みを要求し続けます。DMA エンジンが、SPI DMA ワード・カウント・レジスタの値が 1 から 0 に変わるまで、SPI DMA FIFO からのワード読出しとメモリへの書き込みを続けます。そして、SPI は SPI DMA モードがディスエーブルになるまでワードを受信し続けます。

**i** SPI DMA 転送の全 FIFO の深さ = 6: DMA ペリフェラル (SPI) FIFO (4 個の 16/8 ビット・ワード) + SPI\_TDBR/SPI\_RDBR + SPI インターフェース・シフト・レジスタ。図 10 を参照してください。

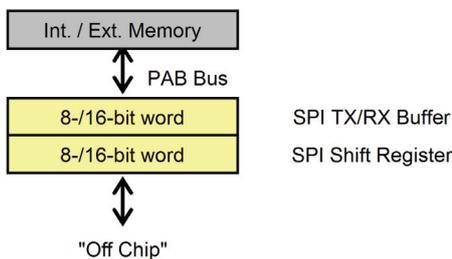


図 9. SPI コア FIFO/バスの構造

この場合や、ほんの数ワード (短い制御メッセージなど) しか送信する必要がない場合は、コア駆動転送のほうが適正な方法です。転送開始モードには、SPI\_RDBR (TIMOD=00) の読出しで開始する方法と SPI\_TDBR (TIMOD=01) への書き込みで開始する方法の 2 つがあります。SPI がイネーブルのときは (SPE=1)、転送が即座に開始されるので、2 番目のモードは使用しないでください。

コア・ベースの SPI 送受信は、TIMOD ビットを 00 に設定して行う必要があります。SPI 受信の場合は、SPI\_RDBR データ・バッファを単に読み出すだけです。SPI 送信の場合は、最初に SPI\_TDBR データ・バッファを充てんし、次に SPI\_RDBR への読出しアクセスを行って転送を開始します。

デバッグに関連しますが、コア SPI 転送の監視は SPI DMA 転送の場合より簡単です。たとえば、すべてのシングル・ワードをシングルステップ・モードで転送できます。実行中の DMA は停止できません。コアが「停止」ステータスの場合、DMA はバックグラウンドで実行されます。

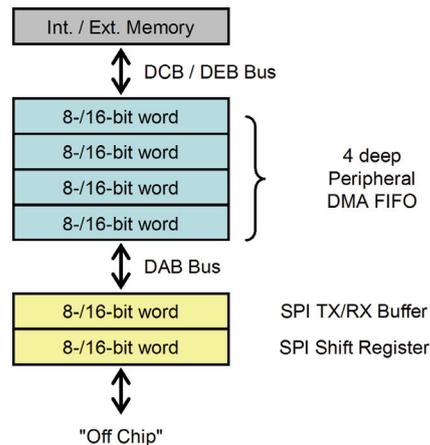


図 10. SPI DMA FIFO/バスの構造



SPI\_RDBR への超高速アクセスはダミー・アクセスであり、最初のワードは無効です。SPI\_RDBR へのアクセスがあるたびに、SPI インターフェースは駆動され次のデータ・ワードをフェッチします。



最後の転送が終わったら、最終ワードを SPI\_SHADOW データ・バッファから読み出すか（別の読み出し転送を駆動しません）、最後に SPI\_RDBR にアクセスする前に最初の SPI をディスエーブルにします (SPE = 0)。SPI\_RDBR へのアクセスのみで、RXS（リード・データ・バッファ・フル）ビットはクリアされます。

## Blackfin プロセッサ SPI の例

関連する .ZIP 内の例は、ADSP-BF537 Blackfin プロセッサの SPI マスタと SPI スレーブの実装に関するものです。詳細については、README.txt ファイルを参照してください。

## SHARC プロセッサのプログラミング・モデル

SPI ポートは、ADSP-21161N、ADSP-2126x、ADSP-2136x、ADSP-2137x SHARC プロセッサ上で使用できます。以下に示すプログラミング・モデルは、主に ADSP-2126x、ADSP-2136x、ADSP-2137x プロセッサに適用できます。ADSP-21161N プロセッサの場合、SPI プログラミング・モデルは異なります。このセクションは、プログラミング・モデルについてある程度具体的かつ詳細に説明しており、HRM における現行文書の付録および要約として活用できます。

### 準備

SHARC プロセッサの各 SPI ポートには以下の信号があります。

- SPICLK
- SPIDS#
- MOSI
- MISO
- SPI スレーブ・セレクト信号 (SPIFLG3-0)

ADSP-2116x および ADSP-2126x プロセッサは 1 個の SPI ポートを備えており、専用ハードウェア・ピンで SPI 信号を使用できます。フラグ 3~0 ピンは、SPI がマスタのときスレーブ・セレクト信号用として機能します。

ADSP-21362/3/4/5/6 プロセッサは 2 個の SPI ポートを備えています。このため、プライマリ SPI 信号を専用のハードウェア・ピンで使用でき、セカンダリ SPI 信号を信号ルーティング・ユニット (SRU) 経由でプロセッサのデジタル・アプリケーション・インターフェース (DAI) ピンに接続できます。プライマリ SPI ポートはフラグ 3~0 ピンをスレーブ・セレクト信号として使用しますが、セカンダリ SPI ポートは任意の DAI ピンに接続できる 4 つの専用 SPIFLG3~0 信号を備えています。

ADSP-21367/8/9、ADSP-2137x プロセッサの場合は、両方の SPI ポートをシグナル・ルーティング・ユニット 2 (SRU2) 経由でプロセッサのデジタル・ペリフェラル・インターフェース (DPI) ピンに接続できます。ADSP-21367/8/9 と ADSP-2137x の SPI ポートは両方とも、SPI をマスタとして使用するときには専用の SPIFLG3~0 信号をスレーブ・セレクト信号とします。

SHARC プロセッサの DAI ピンと DPI ピンは内部プルアップ抵抗を備えています。デフォルトではこのプルアップ抵抗がイネーブルになりますが、対応するプルアップ・イネーブル・レジスタへの書き込みでディスエーブルにすることができます。SPI 信号を DAI または DPI ピンに接続するとき、外付けのプルアップを追加する必要はありません。

ADSP-21367/8/9、ADSP-2137x プロセッサの場合、SPI をマスタとして使用するときには、SRU2 経由で SPICLK 信号を接続しているときにアプリケーションでは下記の点に注意する必要があります。CLKPL = 0 のとき、SPICLK の接続先である DPI ピンのピン・イネーブル信号は以下のように設定する必要があります。

```
SRU(SPI_CLK_O, DPI_PB03_I);
```

```
SRU(HIGH, DPI_PBEN03_I);
```

CLKPL = 1 のとき、SPICLK の接続先である DPI ピンのピン・イネーブル信号は以下のように設定する必要があります。

```
SRU(SPI_CLK_O, DPI_PB03_I);
```

```
SRU(SPI_CLK_PBEN_O, DPI_PBEN03_I);
```

## SPIの初期化

SHARCプロセッサのSPIポートは、SPIを設定するために以下の制御レジスタとステータス・レジスタを使用します。

- SPICTL
- SPIFLG
- SPIBAUD
- SPISTAT

### SPICTLレジスタ

SPICTLレジスタは、SPIの設定を選択しSPIをイネーブルにできる制御ビットを備えています。たとえば、以下のような制御ビットもあります。

- ワード長の選択
- マスタ/スレーブ動作
- データ・フォーマット (リトル・エンディアン/ビッグ・エンディアン)
- クロック位相とクロック極性
- データ・ピンのオープン・ドレイン出力
- パッキング・イネーブル

SPICTLレジスタの詳細については、プロセッサのHRMを参照してください。

### SPIFLGレジスタ

SPIFLGレジスタは、SPIをマスタとして設定するときだけ使用します。このレジスタは、スレーブ・モード動作には使用しません。SPIFLGでは、SPIFLG3~0信号の1つまたは全部をスレーブ・セレクト信号として選択できます。SPIFLGレジスタのDSxENビットを設定すると、それに対応するSPIFLGx信号がスレーブ・セレクト信号として選択されます。

CPHASE=0のとき、スレーブ・セレクト信号はハードウェアによって自動的に生成されます。SPIFLGx信号は各データ転送の前にローレベルにアサートされ、連続する各転送間でハイレベルにアサートされます。

CPHASE = 1のときは、ユーザ・アプリケーションでスレーブ・セレクト信号を手動で生成する必要があります。SPIFLGレジスタのSPIFLGxビットは、この目的のために使用できます。このモードでは、SPIFLGレジスタのSPIFLGxビットを設定または

クリアすると、それがSPIFLGx信号に反映されません。SPIがイネーブルになる前に、SPIFLGレジスタのSPIFLGxビットを設定してこのピンがハイレベルであることを確認してください。転送が起動されたらすぐにそれが実行される前に、SPIFLGレジスタのSPIFLGxビットをクリアします。

### SPIBAUDレジスタ

SPIBAUDレジスタは、SPIをマスタとして設定するときだけ使用します。このレジスタは、スレーブ・モード動作には使用しません。SPIBAUDでは、SPIマスタのボーレートを選択できます。SPIボーレートはBAUDRビット(15~1)で設定できます。このレートはプロセッサのコア・クロックから得られません。ADSP-2136x、ADSP-2137xプロセッサの場合、SPIボーレートは次式で求められます。

$$\text{SPIボーレート} = \text{CCLK} / (8 * (\text{BAUDR} - 1))$$

ADSP-2126xプロセッサの場合、SPIボーレートは次式で求められます。

$$\text{SPIボーレート} = \text{CCLK} / (4 * (\text{BAUDR} - 1))$$

### SPISTATレジスタ

SPISTATレジスタは送受信バッファFIFOのステータスと現送信の完了を知らせる読出し専用レジスタです。このレジスタには、送信/受信エラーとマルチマスタ・エラーを示すビットもあります。このレジスタのビットはWICビットです。

## SPI DMAの初期化

SPIには、DMA動作モードを初期化する以下のレジスタがあります。

- SPIDMAC
- IISPI
- IMSPI
- CSPI
- CPSPI

IISPI、IMSPI、CSPI、CPSPIレジスタはDMAパラメータ・レジスタです。IISPI、IMSPI、CSPIレジスタは、データ転送に必要な内部メモリ・アドレスの値、修飾子の値、カウント値をそれぞれ保持します。CPSPIレジスタは、DMAチェイニングがイネーブルのときに次のシーケンスの値を保持します。

SPIDMAC レジスタには、送信 DMA または受信 DMA の設定、割り込みイネーブル、DMA イネーブル、および DMA チェイニング・イネーブルを行うための制御ビットがあります。また、現行の DMA 転送ステータスと特定のエラー状態を示すステータス・ビットもあります。SPI DMA を初期化する場合は、SPIDMAC レジスタの SPIDEN ビットを設定します。

## SPI 転送開始

SPI 転送は、コア・モードまたは DMA モードで実行できます。SPI 制御レジスタの TIMOD1~0 ビットでは、データ転送モードを選択します。コア・モードでは、コアが送信バッファ (TXSPI) レジスタと受信バッファ (RXSPI) レジスタに直接アクセスします。DMA モードでは、DMA コントローラが受信バッファまたは送信バッファにアクセスしてデータ転送を行います。

TIMOD1~0 ビット値が‘00’のとき、SPI マスタのイネーブル後に RXSPI バッファの読出しを行うとデータ転送が開始されます。これらのビット値が‘01’のとき、SPI マスタのイネーブル後に TXSPI バッファに書き込みを行うとデータ転送が開始されます。DMA 動作モードの場合、TIMOD1~0 ビットの設定値は‘10’です。DMA の処理方向 (送信または受信) に応じて、DMA コントローラは送信バッファへの書き込みまたは受信バッファの読出しを行い、データ転送を開始します。



DMA コントローラは、DMA データ転送に使用する深さ 4 の FIFO を備えています。内部 DMA 要求は、FIFO 上の 4 データ・ワード (1 グループ) に対して生成されます。DMA カウントが 4 の倍数でない場合は、4 ワードの各グループに対して要求が生成され、最後にワードの残りに対して 1 つの要求が生成されます。たとえば、DMA カウントが 7 の場合、DMA コントローラは 2 つの DMA 要求を生成します。1 つは 4 の DMA カウント、もう 1 つはデータ・ワードの残りに対するものです。

どの場合も、マスタはスレーブ・デバイスへの SPI クロックを生成します。スレーブ・セレクト信号は、CPHASE ビットの設定に応じて自動的に生成されるか、手動で生成します。

## SPI 転送 / 割り込みサービス

コア駆動の転送は、割り込み駆動モードまたはポーリング・モードで実行できます。ポーリング・モードでは、ユーザ・アプリケーションで SPISTAT レジスタの読出しを行って送信 / 受信バッファのステータスを継続的にポーリングする必要があります。送信バッファが空の状態か、受信バッファ内にデータあるときは、アプリケーションで送信バッファにデータを書き込むか、受信バッファ内のデータを読み出すことができます。

割り込み駆動モードでは、送信バッファが空の状態か、受信バッファ内に完全な 1 ワードがあるときは、TIMOD1~0 ビットの設定に応じて、割り込みが自動的に生成されます。割り込みサービス・ルーチン内では、ユーザ・アプリケーションが送信バッファに書き込みを行うか、受信バッファの読出しを行う必要があります。

一般に、ユーザ・アプリケーションは全二重通信が必要なポーリング・モードと割り込み駆動モードの組合せを使用できます。

DMA 駆動転送では、ブロック転送終了時に割り込みが生成されます。DMA チェイニング・モードの場合は、各シーケンスの終わりか全シーケンスの終わりに割り込みを生成できます。DMA 割り込みは、SPI 経由で受信したデータ・ブロックの処理に使用するか、新しい転送を開始するために使用できます。



コア・モードで受信バッファの読出しを行ってデータ転送を開始するよう SPI マスタが設定されているときは、最初にマスタがダミー読出しを行う必要があります。このダミー読出しにより、スレーブ・デバイスから最初のデータを送信するためのクロックが生成されます。N ワードのデータを受信するには、マスタはコア・モードで N+1 データ転送を行う必要があります。SPI 受信 DMA 動作の場合、DMA コントローラはこのダミー読出しを管理します。

## SPI転送停止

SPI 転送は SPI がディスエーブルになると終了します。SPI は、割込みサービス・ルーチン内でディスエーブルにできます。コア・モード動作の場合、必要な数のワードの送信または受信が終了した後に、SPICTL レジスタの値をクリアして SPI をディスエーブルにすることができます。SPISTAT レジスタの SPIFE ビットは、SPI をディスエーブルにする前にポーリングする必要があります。このビットは、現在の転送が完了したかどうかを示します。SPI がディスエーブルになる前に、まず現在のデータ転送が完了していることを確認してください。

DMA モード動作の場合、ISR 内で SPIDMAC レジスタの SPIDMAS ビットをポーリングする必要があります。このビットは DMA 完了ステータスを示し、DMA 転送の実行中に設定されます（転送が完了するとクリアされます）。SPISTAT レジスタの SPIFE ビットは、最終データの転送が完了したか確認するためにポーリングする必要があります。次に、SPICTL レジスタと SPIDMAC レジスタの値をクリアして SPI と SPIDMA レジスタをディスエーブルにできます。

DMA チェイニング・モードの場合、チェイン・ポインタ・レジスタへのゼロ書き込みによって SPI 転送を終了できます。ゼロを書き込んだら、SPIDMAC レジスタの SPIDMAS ビットと SPICTL レジスタの SPIFE ビットをポーリングします。現在のデータ転送が完了したら、SPICTL レジスタと SPIDMAC レジスタの値をクリアして SPI と SPIDMA レジスタをディスエーブルにできます。

## SHARC プロセッサ SPI のコード

EE ノートに示したコード例は、ADSP-21369 プロセッサ用に実装されています。SPI マスタ・コードは、ADSP-21369 プロセッサのプライマリ SPI を DMA モードでマスタとして設定します。SPIBAUD レジスタの除数値は、コア・クロックと選択された SPI ボーレートに基づいて実行時に計算されます。SPI スレーブ・コードは、ADSP-21369 プロセッサのプライマリ SPI を DMA モードでスレーブとして設定します。この 2 つのコード例では、マクロを使って DMA 転送の方向を選択できます。また、マクロを使ってクロック位相とクロック極性の設定を選択することもできます。

このコードは、変更を加えずに ADSP-21367/8 および ADSP-2137x プロセッサに使用できます。ADSP-21362/3/4/5/6 および ADSP-2126x プロセッサでは、SPI 設定コードをそのまま使えますが、InitsRU 関数を呼び出す必要はありません。ADSP-2126x プロセッサでは、SPIBAUD レートの計算を変更する必要があります。

## 結論

この EE ノートでは、SHARC、Blackfin の各プロセッサにおける SPI プログラミング・モデルについて解説しています。また、SHARC と Blackfin 間で行う SPI 経由のシリアル通信に関して、コーディングの例も掲載しています。EE ノートで公開した今回の事例は、ADSP-21369 SHARC と ADSP-BF537 Blackfin を用いて、SPI ポート間でのテストを行っています。

## 付録

本書と関わりのある .ZIP ファイルには以下のコード例が含まれています。

- [1] Blackfin プロセッサの SPI マスタ・デバイスのサンプル・コード
- [2] Blackfin プロセッサの SPI スレーブ・デバイスのサンプル・コード
- [3] SHARC プロセッサの SPI マスタ・デバイスのサンプル・コード
- [4] SHARC プロセッサの SPI スレーブ・デバイスのサンプル・コード

## 参考資料

- [1] *ADSP-BF53x/BF56x Blackfin Processor Programming Reference*. Rev. 1.0, June 2005. Analog Devices, Inc.
- [2] *ADSP-BF537 Blackfin Processor Hardware Reference*. Rev. 2.0, December 2005. Analog Devices, Inc.
- [3] *ADSP-BF534/ADSP-BF536/ADSP-BF537 Blackfin Embedded Processor Data Sheet*. Rev. C, February 2007. Analog Devices, Inc.
- [4] *Using the Blackfin Processor SPORT to Emulate a SPI Interface (EE-304)*. Rev. 1, November 10, 2006. Analog Devices, Inc.
- [5] *ADSP-21368 SHARC Processor Hardware Reference*. Rev. 1.0, September 2006. Analog Devices, Inc.
- [6] *ADSP-2136x SHARC Processor Hardware Reference for ADSP-21362/3/4/5/6 Processors*. Rev. 1.0, October 2005. Analog Devices, Inc.
- [7] *ADSP-2126x SHARC Processor Peripherals Manual*. Rev 3.0, December 2005. Analog Devices, Inc.
- [8] *ADSP-21371: SHARC Processor Data Sheet*. Rev. 0, July 2007. Analog Devices, Inc.
- [9] *ADSP-21375: 266 MHz High Performance SHARC Processor Preliminary Data Sheet*. Rev. PrB, December 2005. Analog Devices, Inc.
- [10] *ADSP-21367/ADSP-21368/ADSP-21369 SHARC Processors Data Sheet*. Rev. A, August 2006. Analog Devices, Inc.
- [11] *ADSP-21362/ADSP-21363/ADSP-21364/ADSP-21365/ADSP-21366 SHARC Processors Data Sheet*. Rev. B, June 2007. Analog Devices, Inc.
- [12] *ADSP-21261: 3rd Generation, Low-Cost, 150 MHz SHARC Processor Data Sheet*. Rev 0, April 2006. Analog Devices, Inc.
- [13] *ADSP-21262: 3rd Generation Low Cost 32-Bit Floating-Point SHARC Processor Data Sheet*. Rev. B, October 2005. Analog Devices, Inc.
- [14] *ADSP-21266: High Performance SHARC Audio Processor Data Sheet*. Rev. B, May 2005. Analog Devices, Inc.
- [15] *SHARC SPI Slave Booting (EE-177)*. Rev. 3, January 19, 2007. Analog Devices, Inc.

## 文書の履歴

Revision	Description
Rev 1 – July 8, 2008 by Jeyanthi Jegadeesan and Andreas Pellkofer	Initial release.