

この資料は英語版を翻訳したもので、内容に相違が生じる場合には原文を優先します。こちらの日本語版は参考用としてご利用ください。設計の際には、最新の英語版で内容をご確認ください。

SIII51005-1.1

はじめに

Stratix[®] III デバイス・ファミリは、DSP アプリケーションに最適化された専用の高性能デジタル信号処理 (DSP) ブロックを備えています。アルテラの Stratix デバイス・ファミリのこれらの DSP ブロックは、第三世代のハードワイヤード固定ファンクション・シリコン・ブロックで、信号処理能力の最大化、使いやすさ、シリコン・コスト削減に特化されています。

WiMAX、3GPP WCDMA、高性能コンピューティング (HPC)、Voice over Internet Protocol (VoIP)、H.264 ビデオ圧縮、医療画像、HDTV など多くの複雑なシステムでは、高度なデジタル信号処理技術が使用され、一般に大量の数学的計算が要求されます。Stratix III デバイスは、乗算、加算、減算、累算、総和、およびダイナミック・シフト演算を実行する専用エレメントの組み合わせで構成される DSP ブロックを備えており、高性能 Stratix III ソフト・ロジック・ファブリックと、TriMatrix[™] メモリ構造とを併用して、これらのブロックをコンフィギュレーションし、高度な固定小数点および浮動小数点演算ファンクションを構築することができます。これらのブロックを使用して、有限インパルス応答 (FIR) フィルタ、複素数 FIR フィルタ、無限インパルス応答 (IIR) フィルタ、高速フーリエ変換 (FFT) ファンクション、離散コサイン変換 (DCT) ファンクションなどの膨大な計算を要求する一般的なサブシステムを容易に実装できます。

DSP ブロックの概要

各 Stratix III デバイスには、乗算、積和、乗算累積 (MAC)、ダイナミック・シフトの各ファンクションを効率的に実行する DSP ブロックの 2 ~ 7 カラムがあります。Stratix III DSP ブロックの論理機能は、Stratix および Stratix II デバイスに見られる前世代の DSP ブロックのスーパーセットです。

Stratix III DSP ブロックのアーキテクチャ上の重要な機能として、次のものがあります。

- 高性能、消費電力最適化、完全レジスタ化およびパイプライン化された乗算演算
- ネイティブにサポートされる 9 ビット、12 ビット、18 ビット、36 ビット・ワード長
- ネイティブにサポートされる 18 ビット複素数乗算
- 効率的にサポートされる浮動小数点演算フォーマット (単精度で 24 ビット、倍数精度で 53 ビット)

- 符号付きおよび符号なし入力サポート
- 乗算結果を効率的に結合するビルトイン加算、減算、累算ユニット
- フィルタリング・アプリケーション用のタップ・ディレイ・ラインを形成するカスケード接続 18 ビット入力バス
- 出力結果を外部ロジックのサポートなしで、隣接するブロックに入力可能なカスケード接続 44 ビット出力バス
- 豊富でフレキシブルな演算丸めおよび飽和ユニット
- 効率的なバレル・シフタ・サポート
- 適応型フィルタリングをサポートするループバック機能

Stratix III デバイス・ファミリの DSP ブロック数を、表 5-1 に示します。

	デバイス	DSP ブロック	独立した入力および出力乗算演算子					Four Multiplier Adder モード
			9 × 9 乗算器	12 × 12 乗算器	18 × 18 乗算器	18 × 18 複素数	36 × 36 乗算器	18 × 18
Stratix III ロジック	EP3SL50	27	216	162	108	54	54	216
	EP3SL70	36	288	216	144	72	72	288
	EP3SL110	36	288	216	144	72	72	288
	EP3SL150	48	384	288	192	96	96	384
	EP3SL200	72	576	432	288	144	144	576
	EP3SE260	96	768	576	384	192	192	768
	EP3SL340	72	576	432	288	144	144	576
Stratix III エンハンスト	EP3SE50	48	384	288	192	96	96	384
	EP3SE80	84	672	504	336	168	168	672
	EP3SE110	112	896	672	448	224	224	896
	EP3SE260 (1)	96	768	576	384	192	192	768

表 5-1 の注:

- (1) EP3SE260 デバイスは、LE、メモリ、乗算器のリソースが豊富です。したがって、ロジック (L) とエンハンスト (E) バリエーションのいずれにも対応します。

表 5-1 に示すように、Stratix III の最大 DSP 用デバイス (EP3SE110) は、36 × 36、複素数 18 × 18、加算の各モードにおいて最大 896 個の 18 × 18 の乗算機能を提供することが可能になっています。

各 DSP ブロックは高さで 4 個の LAB ブロックを占有し、さらに 2 つのハーフ・ブロックに分割できます。これらのブロックはいくつかの共通クロック信号を共有し、同一の機能を持ちます。図 5-1 に各ブロックのレイアウトを示します。


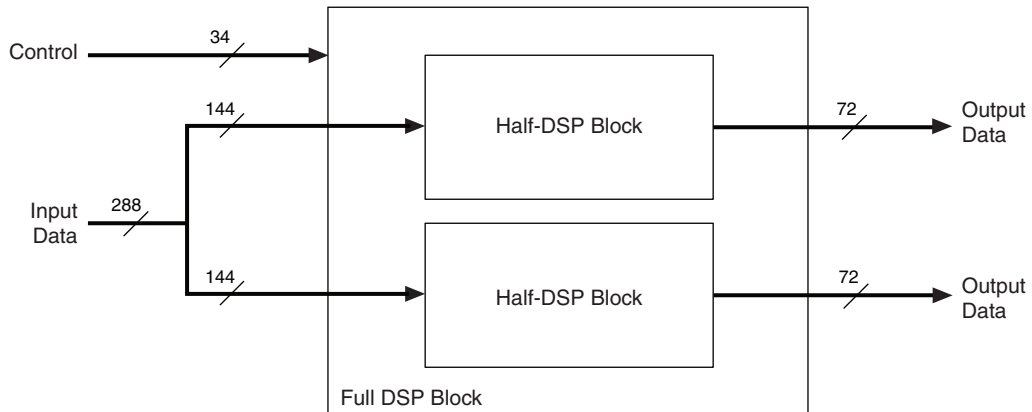
 Stratix III DSP ブロックの 288 ビット入力データ・ラインは、Stratix および Stratix II の 2 倍ですが、出力データ・ラインの本数は 144 ビットとなっており、従来と同一です。

図 5-1. DSP ブロック信号の概要



単純化された DSP 動作

Stratix および Stratix II デバイスでは、基本的なビルディング・ブロックは、2 個の 9 ビット × 9 ビット乗算器としても機能する 18 ビット × 18 ビット乗算器で構成されています。Stratix III の場合は、式 5-1 と図 5-2 に示すように、18 ビット × 18 ビット乗算器ペアとそれに続く第 1 ステージの 37 ビット加算 / 減算ユニットが基本的なビルディング・ブロックになります。すべての符号付き入力および出力データは、2 の補数形式で表現されます。

式 5-1. 乗算器の計算式

$$P[36..0] = A_0[17..0] \times B_0[17..0] \pm A_1[17..0] \times B_1[17..0]$$

図 5-2. 基本的な Two-Multiplier Adder のビルディング・ブロック

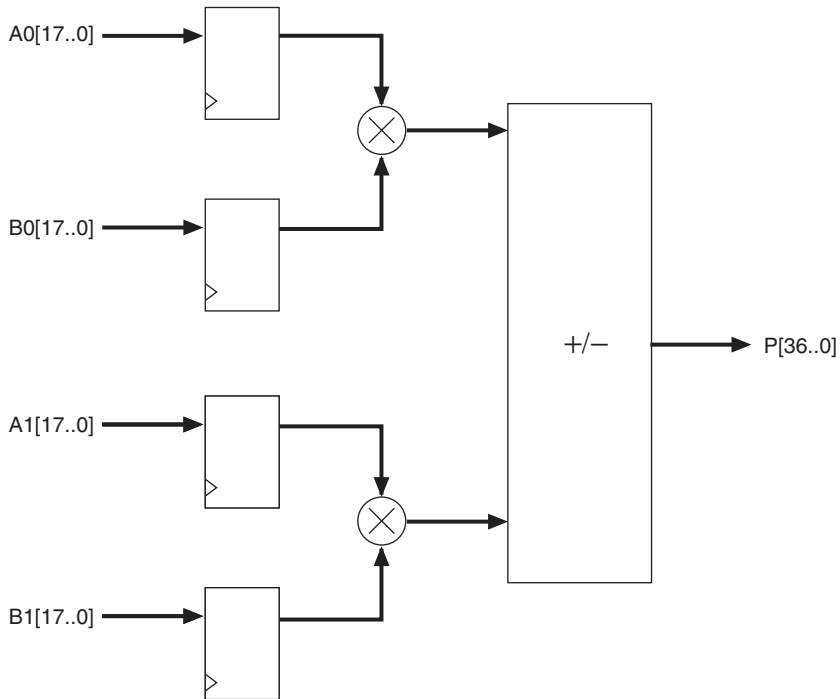


図 5-2 に示す構造は、以後の項で説明するとおり、複素数乗算器や 36×36 乗算器などのより複雑な構造を構築するのに非常に便利です。

Stratix III の各 DSP ブロックには、4 個の Two-Multiplier Adder ユニットがあります (ハーフ・ブロックあたり 2 個の Two-Multiplier Adder ユニット)。したがって、DSP ブロックあたり 8 個の 18×18 乗算器機能が存在します。

Two-Multiplier Adder ユニットに続いて、パイプライン・レジスタ、第 2 ステージ加算器、出力レジスタ・ステージがあります。第 2 ステージの加算器をコンフィギュレーションして、ハーフ・ブロックごとに次の代替ファンクションを提供することができます。

式 5-2. Four-Multiplier Adder の計算式

$$Z[37..0] = P_0[36..0] + P_1[36..0]$$

式 5-3. Four-Multiplier Adder の計算式 (44 ビット累算)

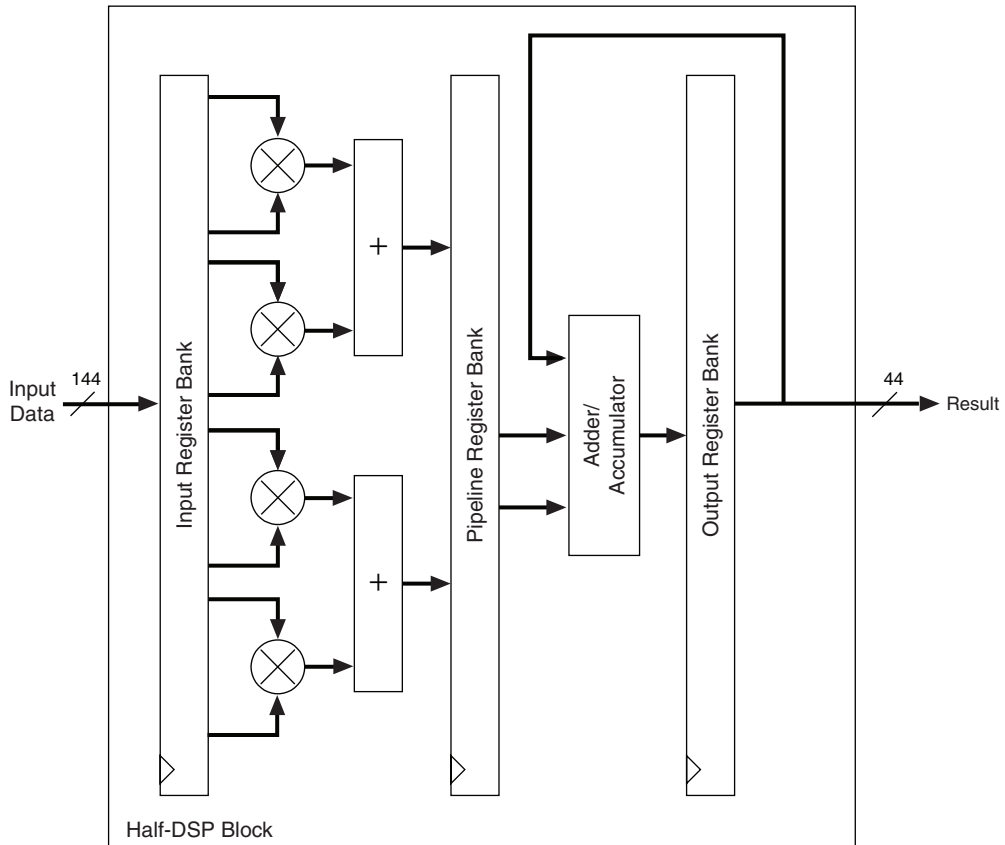
$$W_n[43..0] = W_{n-1}[43..0] \pm Z_n[37..0]$$

上記の式で、 n はサンプル時間を表し、 $P[36..0]$ は Two-Multiplier Adder ユニットからの結果です。

式 5-2 は 4 回の 18 ビット \times 18 ビット乗算演算 (Four-Multiplier Adder) の総和を生成し、式 5-3 は 4 回の 18 ビット \times 18 ビット乗算演算とこのユニットの出力をフィードバックすることによって最大 44 ビットの累算機能を提供します。これを図 5-3 に示します。

選択したモードに応じて、すべてのレジスタ・ステージをバイパスすることができます。

図 5-3. Four-Multiplier Adder および Accumulation 累算機能



一般的な FIR 構造を効率的にサポートするために、Stratix III の DSP ブロックへの重要な追加機能として、追加ソフト・ロジックのオーバーヘッドなしで、ハーフ・ブロックの結果を次のハーフ・ブロックに入力できる能力があります。これは専用の加算ユニットと 1 つ前のハーフ・ブロックの 44 ビット結果を、現在のブロックの 44 ビット結果に加算する配線を含めることによって実現しています。44 ビット結果は、次のハーフ・ブロックに送られるか、出力レジスタ・ステージを経て DSP ブロックから出力されます。これを図 5-4 に示します。詳細な例は、以降の項で説明します。

高速、低レイテンシの Four-Multiplier Adder ユニットと出力チェイニング加算器の「チェイン・カスケード」機能を組み合わせることで、最適な FIR およびベクトル乗算機能が得られます。

シングル・チャンネル型の FIR フィルタを効率的にサポートするには、乗算器入力のレジスタの1つをコンフィギュレーションして、タップ・ディレイ・ライン入力を形成し、リソースを節約すると共にシステム性能を向上させることができます。

図 5-4. FIR 構造の出力カスケード機能

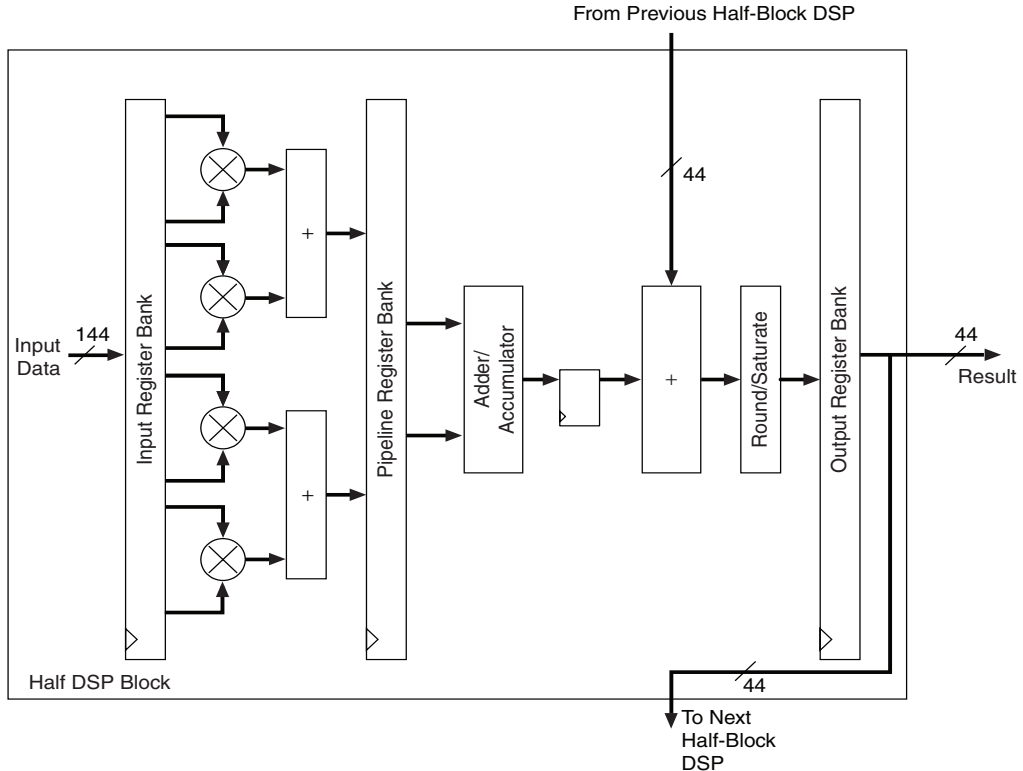
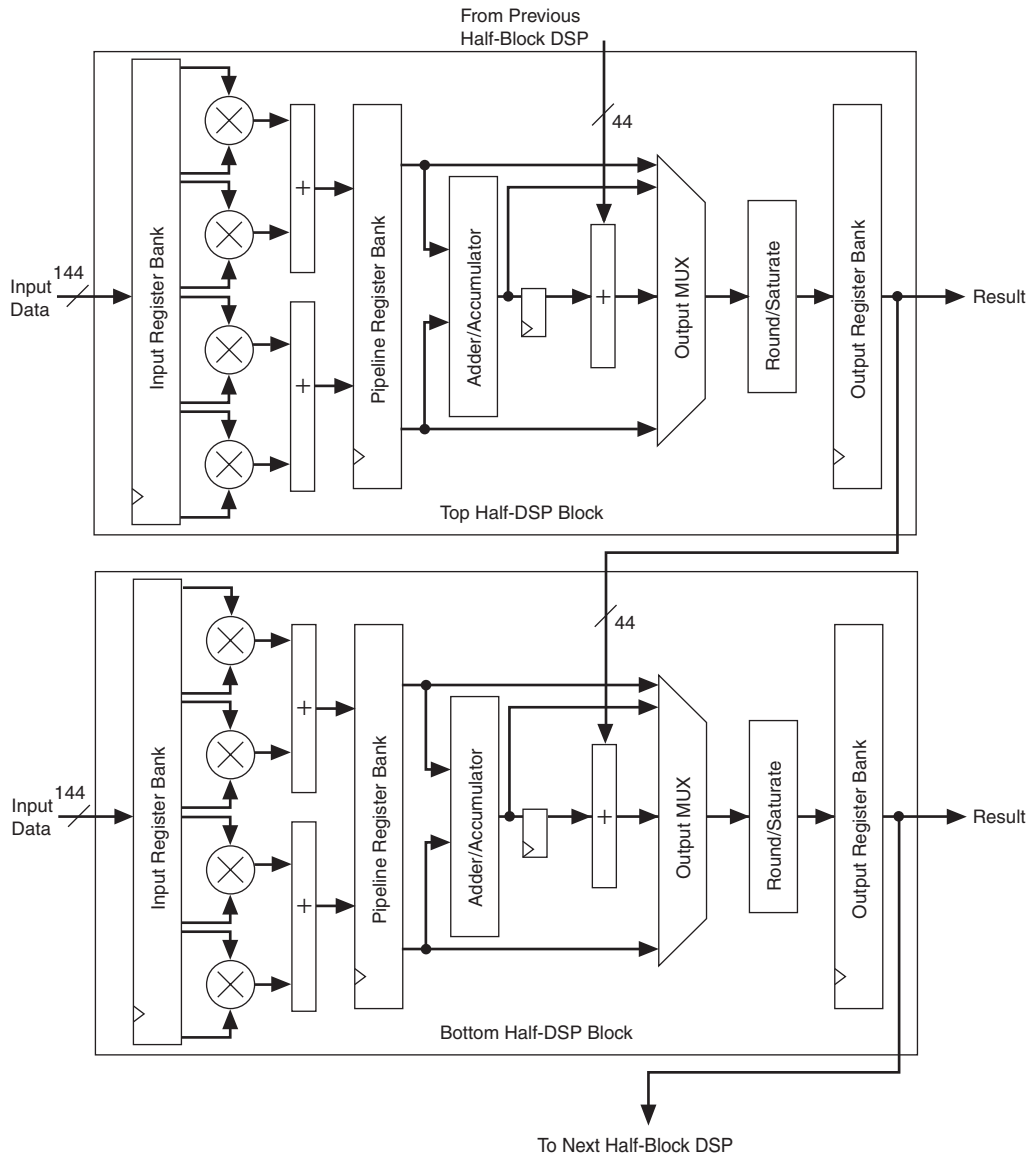


図 5-4 では、最適化された丸めおよび飽和ユニット (Rounding and Saturation Unit: RSU) も示しています。このユニットは、信号処理で使用される一般的な丸め演算および飽和機能の豊富なセットを備えています。

独立した乗算モードおよび総和モードに加えて、DSP ブロックをシフト演算の実行に使用できます。DSP ブロックは、1 クロック・サイクルで、論理左 / 右シフト、算術左 / 右シフト、およびローテイト操作をダイナミックに切り替えることができます。

Stratix III DSP ブロックのトップレベル図を図 5-5 に示します。ハーフ・ブロックについてのより詳細な内容を図 5-6 に示します。

図 5-5. Stratix III の完全な DSP ブロックの要約



動作モードの概要

Stratix III の各 DSP ブロックでは、5 つの基本動作モードのいずれかを使用できます。表 5-2 に、5 つの基本動作モードと、モードに応じて単一 DSP ブロック内に実装可能な乗算器数を示します。

モード	乗算器の幅	乗算器の数	ブロックあたりの数	符号付きまたは符号なし	丸め、飽和機能	シフト・レジスタ	チェインアウト加算器	第 1 ステージ加算 / 減算	第 2 ステージ加算 / 累算
独立した乗算器	9 ビット	1	8	両方あり	なし	なし	なし	—	—
	12 ビット	1	6	両方あり	なし	なし	なし	—	—
	18 ビット	1	4	両方あり	あり	あり	なし	—	—
	36 ビット	1	2	両方あり	なし	なし	なし	—	—
	ダブル	1	2	両方あり	なし	なし	なし	—	—
Two-Multiplier Adder (1)	18 ビット	2	4	符号付き (4)	あり	なし	なし	両方あり	N/A
Four-Multiplier Adder	18 ビット	4	2	両方あり	あり	あり	あり	両方あり	加算のみ
Multiply Accumulate	18 ビット	4	2	両方あり	あり	あり	あり	両方あり	両方あり
シフト (2)	36 ビット (3)	1	2	両方あり	なし	なし	—	—	—

表 5-2 の注：

- (1) このモードはループバック・モードもサポートします。ループバック・モードでは、DSP ブロックあたりのループバック乗算器の数は 2 であり、残りの乗算器は通常の Two-Multiplier Adder モードで使用できます。
- (2) ダイナミック・シフト・モードは、算術左シフト、算術右シフト、論理左シフト、論理右シフト、およびローテイト操作をサポートします。
- (3) ダイナミック・シフト・モードでは、32 ビット入力ベクタを操作しますが、乗算器の幅は 36 ビットでコンフィギュレーションされます。
- (4) 符号なしの値もサポートされますが、結果を 36 ビットに収容可能なことを確認する必要があります。

DSP ブロックは、2 個のハーフ・ブロック（上半分と下半分）で構成されます。各ハーフ・ブロックには 4 個の 18×18 乗算器があります。

Quartus® II ソフトウェアには、乗算器の動作モードをコントロールするのに使用するメガファンクションが含まれています。メガファンクションの MegaWizard® Plug-In Manager を使用して、適切なパラメータ設定を行った後、Quartus II ソフトウェアが自動的に DSP ブロックをコンフィギュレーションします。

Stratix III DSP ブロックは、異なるモードで同時に動作できます。各ハーフ・ブロックは 4 個の `clock`、`ena`、および `aclr` 信号を共有する以外は、完全に独立しています。例えば、1 つの DSP ブロックを分割して、1 つのハーフ・ブロックで 9×9 乗算器を、もう 1 つのハーフ・ブロックで 18×18 の Two-Multiplier Adder を操作することができます。これにより、DSP ブロックのリソース効率が向上し、Stratix III デバイスに実装できる乗算器を増やすことができます。Quartus II ソフトウェアは、同じブロック内で同じ DSP ブロック・リソースを共有可能な乗算器を自動的に配置します

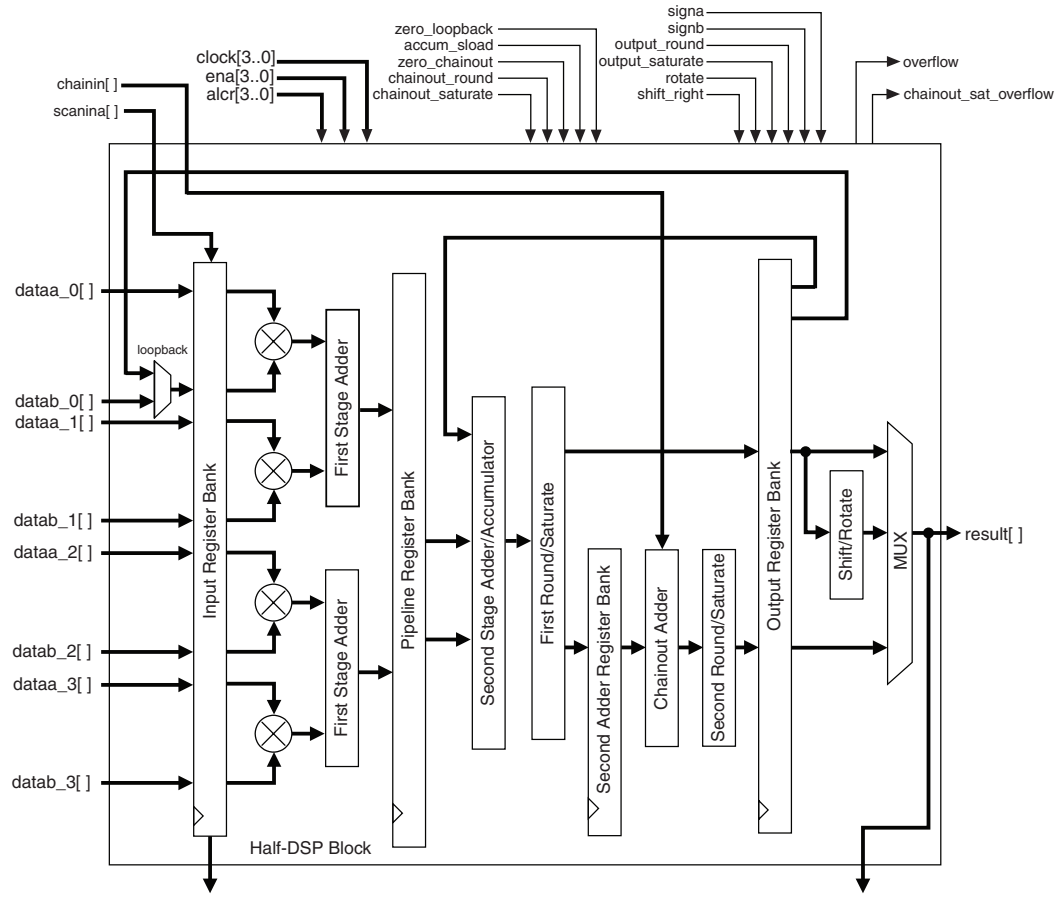
DSP ブロック のリソースの 説明

DSP ブロックは以下の要素で構成されます。

- 入力レジスタ・バンク
- 4 個の Two-Multiplier Adder
- パイプライン・レジスタ・バンク
- 2 個の第 2 ステージ加算器
- 4 個の丸めおよび飽和ロジック・ユニット
- 第 2 加算器レジスタおよび出力レジスタ・バンク

DSP ハーフ・ブロックの詳細な全体アーキテクチャを [図 5-6](#) に示します。

図 5-6. ハーフ DSP ブロック・アーキテクチャ



入力レジスタ

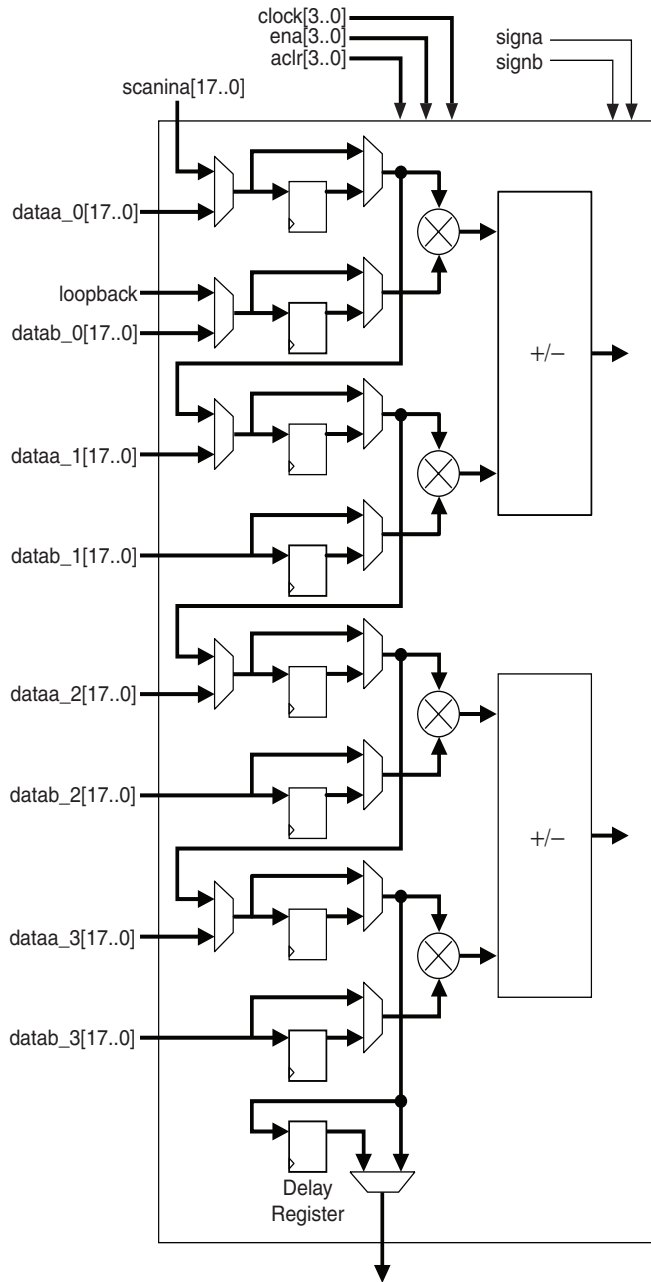
DSP ブロックのレジスタはすべてクロック信号のポジティブ・エッジでトリガされ、電源投入時にクリアされます。各乗算器オペランドは、入力レジスタに供給するか、入力レジスタをバイパスして直接乗算器に供給できます。(これはコンパイル時にコンフィギュレーションされます)。以下の DSP ブロック信号は、DSP ブロック内の入力レジスタを制御します。

- clock [3..0]
- ena [3..0]
- aclr [3..0]

それぞれの DSP ブロックには、ハーフ DSP ブロックあたり 9 個の 18 ビット・データ入力レジスタ・バンクが含まれています。ハーフ DSP ブロックごとに、4 個の乗算器の入力として 8 個のデータ・レジスタ・バンクを使用するオプションがあります。9 番目の特別なレジスタ・バンクは、DSP ブロックのカスケードとチェインアウトの両方の機能を使用するモードで要求されるディレイ・レジスタで、チェイン・カスケード機能を使用するときにレイテンシ要件をバランスさせるためのものです。

入力レジスタ・バンクの機能は、タップ・ディレイ・ラインをサポートすることです。したがって、乗算器の A 側の入力（上位レジスタ）は、[図 5-7](#) に示すように、一般的な配線またはカスケード・チェインからドライブできます。

図 5-7. ハーフ DSP ブロックの入力レジスタ



コンパイル時に A 入力⁽¹⁾が一般的な配線とカスケード・チェーンのどちらを選択するかを決める必要があります。カスケード・モードでは、1 つの乗算器ブロックからの専用シフト出力が、その下にある（同じハーフ DSP ブロック内の）隣接する乗算器、または次のハーフ DSP ブロックの最初の乗算器の入力レジスタに直接供給され、DSP ブロックごとに 1 個の 8 タップ・シフト・レジスタ・チェーンを形成します。DSP ブロックは、下位 DSP ブロックにカスケード接続することによって、シフト・レジスタ・チェーンの長さを伸ばすことができます。専用シフト・レジスタ・チェーンは 1 つのカラム内での接続となりますが、通常の FPGA 配線リソースを使用して、複数のカラムを接続し、より長いシフト・レジスタ・チェーンを実装できます。

シフト・レジスタは、FIR フィルタなどの DSP ファンクションで役立ちます。18×18 以下の幅の乗算器を実装する場合、入力シフト・レジスタは DSP ブロックの内部にあるため、シフト・レジスタ・チェーンを形成するための外部ロジックは必要ありません。この実装により、必要なロジック・エレメント (LE) リソースが大幅に減り、配線遅延も最小化できます。

Stratix III デバイスのハーフ DSP ブロック（上半分と下半分）の最初の乗算器には、[図 5-6](#) に示すように、一般配線とループバックのいずれかを選択するための第 1 乗算器 B 入力（下位レジスタ入力）レジスタ用のマルチプレックス回路があります。ループバック・モードでは、最上位 18 ビットのラッチされた出力が、それぞれのハーフ DSP ブロックの最初の乗算器の乗算器入力にフィードバックとして接続されます。ループバック・モードは、現在の出力の計算に 1 つ前の出力が必要となる再帰的フィルタによって使用されます。

ループバック・モードは、[5-26 ページ](#)の「[Two-Multiplier Adder 総和モード](#)」で詳細に説明しています。

[表 5-3](#) に、DSP ブロックの入力レジスタ・モードの概要を示します。

表 5-3. 入力レジスタ・モード					
レジスタ入力モード (1)	9 × 9	12 × 12	18 × 18	36 × 36	ダブル
パラレル入力	√	√	√	√	√
シフト・レジスタ入力 (2)	—	—	√	—	—
ループバック入力 (3)	—	—	√	—	—

表 5-3 の注：

- (1) 乗算器のオペランド入力のワード長は、コンパイル時にスタティックにコンフィギュレーションされます。
- (2) A オペランドでのみ使用可能。
- (3) ハーフ・ブロックごとに 1 ループバック入力のみ可能。詳細については、[図 5-15](#)を参照してください。


乗算器および第 1 ステージ加算器

乗算器ステージでは、 9×9 、 12×12 、 18×18 、または 36×36 の乗算器がネイティブにサポートされます。他のワード長の場合、直近の適切なネイティブ・ワード長までパディングされます。例えば、 16×16 はパディングされて、 18×18 が使用されます。詳細は、5-18 ページの「[独立乗算器モード](#)」を参照してください。乗算器のデータ幅に応じて、シングル DSP ブロックは複数の乗算を平行に実行できます。

乗算器の各オペランドは、符号付きまたは符号なし数値になります。2 つのダイナミック信号 `signa` と `signb` が、それぞれ各オペランドの表現を制御します。`signa/signb` 信号の logic 1 値は、`data A/data B` が符号付き数値であることを示します。logic 0 値は、符号なし数値を示します。表 5-4 に、各種オペランドの符号表現に対する乗算結果の符号を示します。乗算結果は、オペランドの 1 つが符号付き数値の場合は符号が付けられます。

データ A (<code>signa</code> 値)	データ B (<code>signb</code> 値)	結果
符号なし (ロジック 0)	符号なし (ロジック 0)	符号なし
符号なし (ロジック 0)	符号付き (ロジック 1)	符号付き
符号付き (ロジック 1)	符号なし (ロジック 0)	符号付き
符号付き (ロジック 1)	符号付き (ロジック 1)	符号付き

ハーフ・ブロックごとに、固有の `signa` および `signb` 信号があります。したがって、同じ DSP ハーフ・ブロックに供給される `data A` 入力はずべて、同じ符号表現でなければなりません。同様に、同じ DSP ハーフ・ブロックに供給される `data B` 入力はずべて、同じ符号表現でなければなりません。乗算器は、完全精度の 18×18 ループバック・モードおよび `Two-Multiplier Adder` モードを除いて、すべての動作モードで、符号表現には関係なく完全精度を提供します。詳細は、5-26 ページの「[Two-Multiplier Adder 総和モード](#)」を参照してください。

 `signa` 信号と `signb` 信号が使用されない場合、Quartus II ソフトウェアはデフォルトで符号なし乗算を実行するように乗算器を設定します。

乗算器の出力は、図 5-6 に示すように、第 1 ステージ加算器に供給可能な唯一の出力です。DSP ブロックには 4 個の第 1 ステージ加算器があります（ハーフ DSP ブロックごとに 2 個の加算器）。第 1 ステージ加算器ブロックには、加算と減算を実行する能力があります。加算または減算のコントロール信号はスタティックであり、コンパイル時にコンフィギュレーションしなければなりません。第 1 ステージ加算器は 18×18 複素数乗算の加算や、 36×36 乗算およびシフト演算の第 1 ステージを実行します。

ユーザの仕様に応じて、第 1 ステージ加算器の出力はオプションで、パイプライン・レジスタ、第 1 ステージ加算器、丸めおよび飽和ユニット、または出力レジスタのいずれかに供給されます。


パイプライン・レジスタ・ステージ

図 5-6 に示すように、第 1 ステージ加算器の出力はパイプライン・レジスタに供給されますが、パイプライン・レジスタをバイパスすることもできます。パイプライン・レジスタにより、特に後続の DSP ブロック・ステージを使用する場合に、DSP ブロックの動作周波数が向上します（ただし、追加のレイテンシのサイクルが必要）。パイプライン・レジスタは、信号パスを入力レジスタ / 乗算器 / 第 1 ステージ加算器と第 2 ステージ加算器 / 丸めおよび飽和 / 出力レジスタ間で分割して 2 つの短いパスにします。


第 2 ステージ加算器

DSP ブロックごとに 4 個の 44 ビット第 2 ステージ加算器があります（ハーフ DSP ブロックごとに 2 個の加算器）。第 2 ステージ加算器は、次のようにコンフィギュレーションできます。

- 36 ビット乗算器の最終ステージ
- 4 つ (18×18) の総和
- アキュムレータ (44 ビット最大)
- チェイン出力の総和 (44 ビット最大)

 チェイン出力加算器は、チェイン出力総和モードでは、第 2 レベル加算器と同時に使用できます。

第 2 ステージ加算器の出力は、丸めおよび飽和論理ユニットと出力レジスタのどちらに出力するかを選択できます。

 第 2 ステージ加算器は、乗算器および第 1 ステージ加算器と別々に使用することはできません。


丸めおよび飽和ステージ

丸めおよび飽和論理ユニットは、44 ビット第 2 ステージ加算器（丸め論理ユニットとそれに続く飽和論理ユニット）の出力に配置されます。ハード DSP ブロックごとに、2 個の丸めおよび飽和論理ユニットがあります。丸めおよび飽和論理ユニットの入力は、以下のステージの 1 つから送ることができます。

- 乗算器の出力（ 18×18 の独立した乗算モード）
- 第 1 ステージ加算器の出力（Two-Multiplier Adder）
- パイプライン・レジスタの出力
- 第 2 ステージ加算器の出力（Four-Multiplier Adder、 18×18 の Multiply-Accumulate モード）

これらのステージについては、5-18 ページの「動作モードの説明」で詳細に説明しています。

丸めおよび飽和論理ユニットは、それぞれダイナミック丸め信号および飽和信号によって制御されます。丸めおよび/または飽和の値が logic 1 の場合、それぞれ丸めおよび/または飽和論理ユニットが有効になります。

 丸めおよび飽和論理ユニットは、合体してまたは個別に使用できます。

第 2 加算器および出力レジスタ

第 2 加算器レジスタおよび出力レジスタ・バンクは、2 個の 44 ビット・レジスタのバンクであり、これらを結合することによって、 36×36 の出力結果をサポートする大規模な 72 ビット・バンクを形成できます。

Stratix III デバイスの異なるステージの出力は、出力選択ユニットを介して出力レジスタに配線されます。DSP ブロックの動作モードに応じて、出力選択ユニットは DSP ブロックの出力が、乗算器ブロック、第 1 ステージ加算器、パイプライン・レジスタ、第 2 ステージ加算器、または丸めおよび飽和論理ユニットの出力から送られるように選択します。出力選択ユニットは、指定された DSP ブロックの動作モードに基づいてソフトウェアで自動的に設定され、出力レジスタをドライブするかバイパスするかのオプションがあります。この例外はブロックがシフト・モードで使用される場合です。このケースでは、出力選択 MUX をユーザが直接ダイナミックに制御します。

DSP ブロックが「チェイン・カスケード」出力モードでコンフィギュレーションされている場合、両方の第2ステージ加算器が使用されます。最初の加算器は Four-Multiplier Adder の実行に、2 番目の加算器はチェインアウト加算器に使用されます。Four-Multiplier Adder の出力は、チェインアウト加算器に入る前に、第2ステージ加算器レジスタに配線されます。チェインアウト加算器の出力は、通常出力レジスタ・バンクに送られます。コンフィギュレーションに応じて、チェインアウトの結果は次のハーフ・ブロックのチェインアウト加算器入力、または汎用ファブリック（通常出力レジスタとして機能）の入力に配線できます。詳細は、5-18 ページの「動作モードの説明」を参照してください。

第2ステージおよび出力レジスタは、クロック信号のポジティブ・エッジでトリガされ、電源投入時にクリアされます。以下の DSP ブロック信号は、DSP ブロック内の出力レジスタを制御します。

- clock [3..0]
- ena [3..0]
- aclr [3..0]

動作モードの説明

独立乗算器モード

独立した入力および出力乗算器モードでは、DSP ブロックは汎用乗算器の個別乗算演算を実行します。

9、12、および 18 ビット乗算器

各 DSP ブロック乗算器を、9、12、18 ビット乗算にコンフィギュレーションすることができます。単一 DSP ブロックは、最大 8 個の個別 9×9 乗算器、6 個の 12×12 乗算器、または最大 4 個の個別 18×18 乗算器をサポートできます。9 ビット幅までのオペランドの場合、 9×9 乗算器が実装されます。10 ~ 12 ビット幅のオペランドでは、 12×12 乗算器が実装され、13 ~ 18 ビット幅のオペランドでは、 18×18 乗算器が実装されます。この実装は Quartus II ソフトウェアが LSB にゼロをパディングして行います。図 5-8、5-9、および 5-10 に、独立した乗算器演算モードの DSP ブロックを示します。

図 5-8. ハーフ DSP ブロックの 18 ビット独立乗算器モード

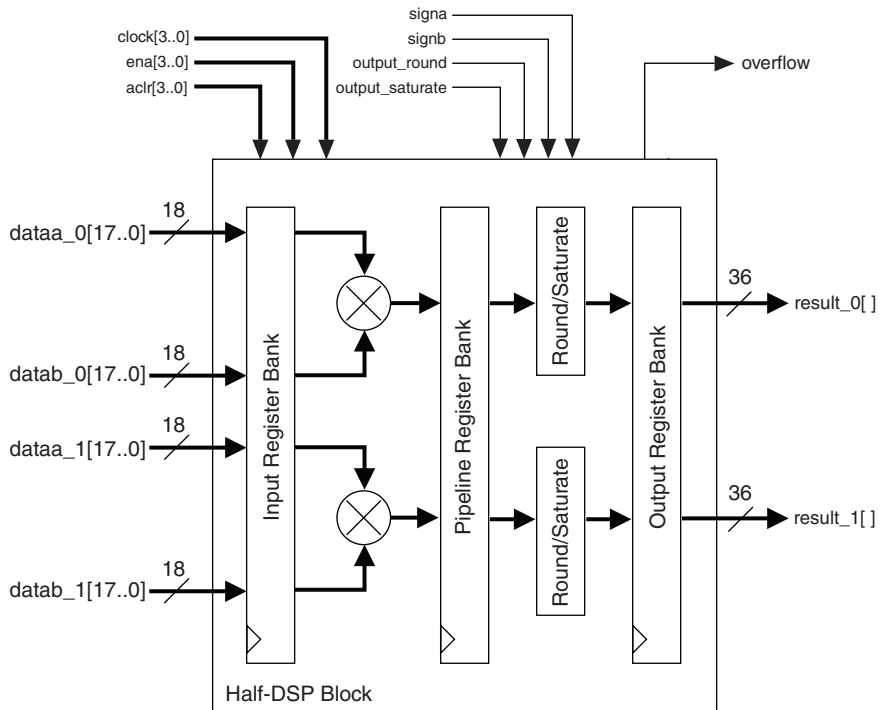


図 5-9. ハーフ DSP ブロックの 12 ビット独立乗算器モード

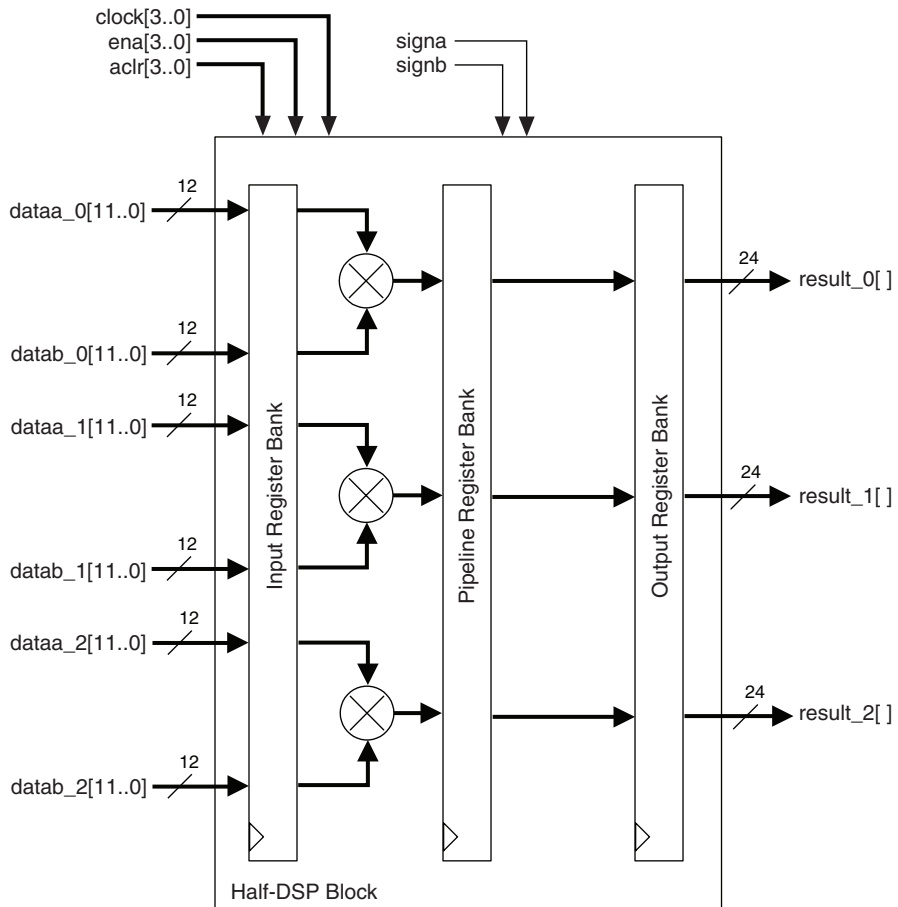
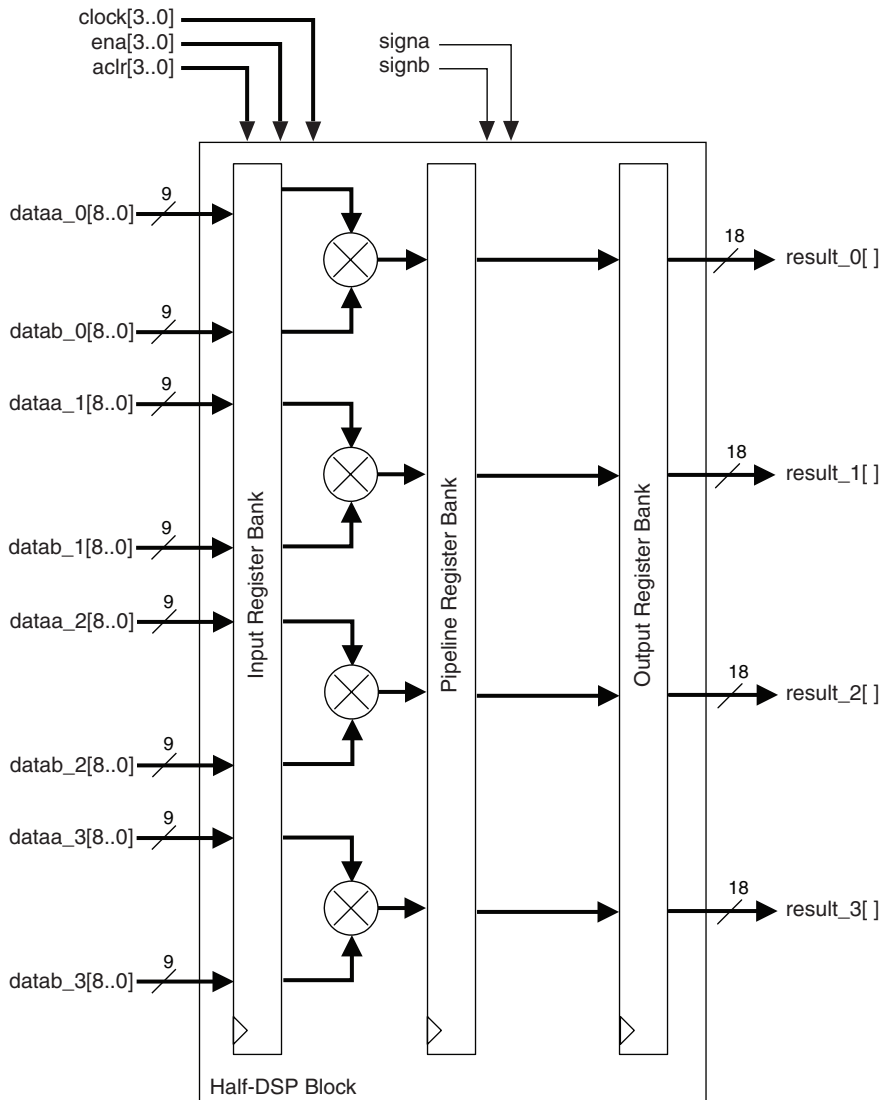



図 5-10. ハーフ・ブロックの 9 ビット独立乗算器モード



乗算器オペランドは、符号付き整数、符号なし整数、または両方の組み合わせを受け入れることができます。signa 信号と signb 信号をダイナミックに変更して、DSP ブロックにラッチすることができます。さら

に、乗算器の入力と結果は個別にラッチ可能です。DSP ブロック内でパイプライン・レジスタを使用して乗算器の結果をパイプラインできるため、DSP ブロックの性能が向上します。

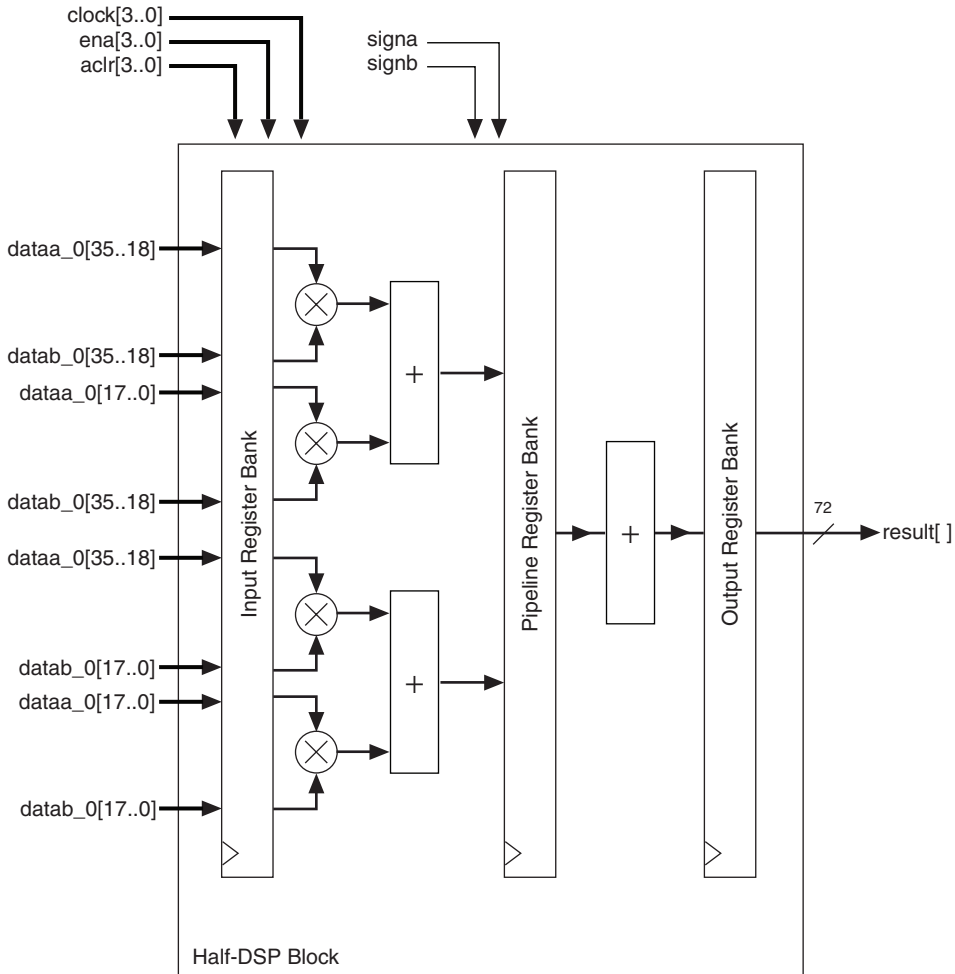
 丸めおよび飽和論理ユニットは、18 ビット独立乗算器モードでのみサポートされます。

36 ビット乗算器

4 個の 18×18 乗算器を使用して、 36×36 乗算器を効率的に構築できます。このような簡素化は、1 つのハーフ DSP ブロックに都合よくフィットし、 36×36 モードを選択することによって自動的に DSP ブロックに実装されます。Stratix III デバイスは、DSP ブロックあたり最大 2 個の 36 ビット乗算器を配置できます（ハーフ DSP ブロックに 1 つの 36 ビット乗算器）。36 ビット乗算器は、独立した乗算器モードでも使用されますが、ハーフ DSP ブロック全体を使用します。これには、 36×36 ビット乗算演算を実装するためのパイプライン・レジスタ後の専用ハードウェア・ロジックが含まれます。これを [図 5-11](#) に示します。

36 ビット乗算器は、単精度および拡張単精度浮動点演算アプリケーションの仮数の乗算など、18 ビット精度を超える要求するアプリケーションに有効です。

図 5-11. ハーフ DSP ブロックの 36 ビット独立乗算器モード



ダブル乗算器

Stratix III DSP ブロックは、IEEE 倍精度浮動小数点乗算の仮数部分を計算するのに必要な符号なし 54×54 ビット乗算器を効率よくサポートするようにコンフィギュレーションできます。 54×54 ビット乗算器は、基本的な 18×18 乗算器、シフタ、および加算器を使用して構築できます。シフタおよび加算器で構築された Stratix III DSP ブロックを効率的に利

用するために、基本的な 36×36 乗算器モードをわずかに変更した、特殊なダブル・モード（部分的 54×54 乗算器）を使用できます。これを図 5-12 および図 5-13 に示します。

図 5-12. ハーフ DSP ブロックのダブル・モード

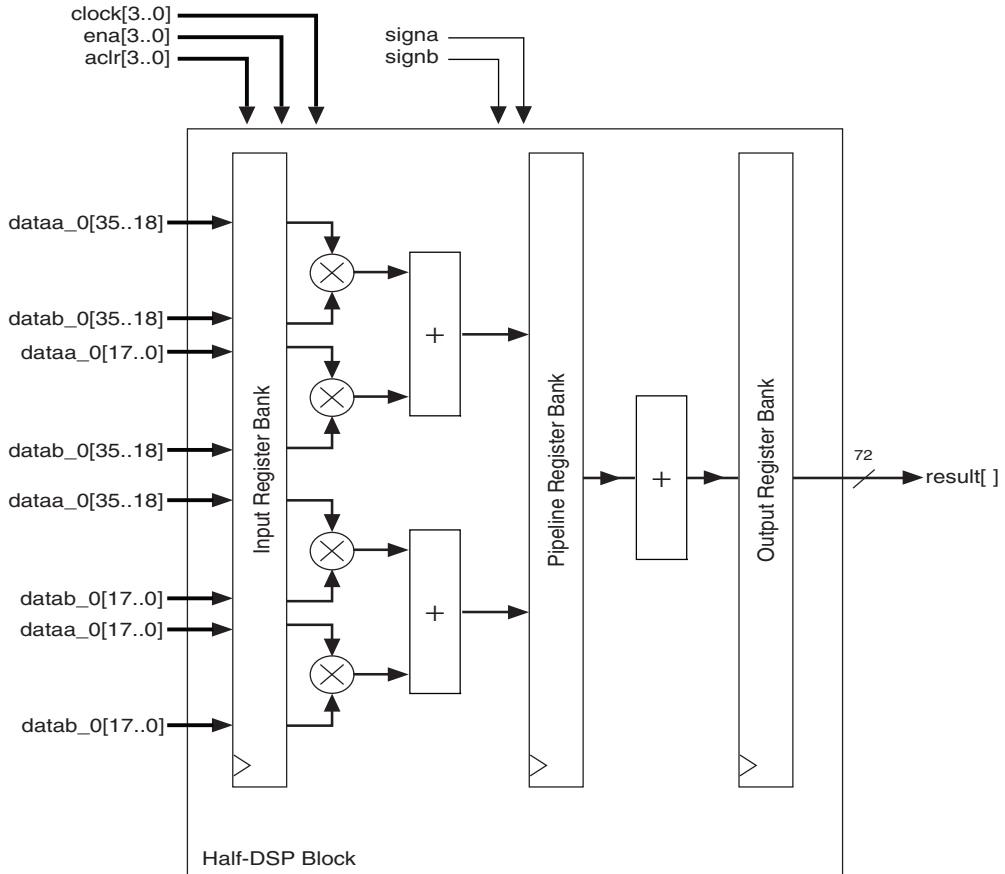
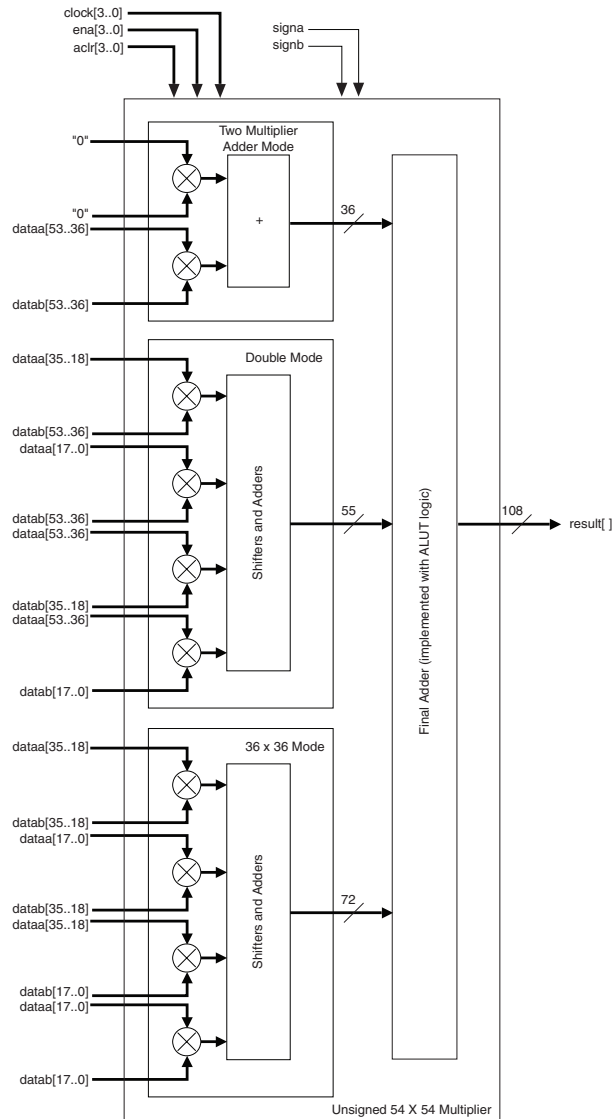



図 5-13. 符号なし 54 × 54 乗算器



Two-Multiplier Adder 総和モード

Two-Multiplier Adder コンフィギュレーションでは、DSP ブロックは 4 個の 18 ビット Two-Multiplier Adder（ハーフ DSP ブロックあたり 2 個の Two-Multiplier Adder）を実装できます。2 個の乗算器の出力を加算または減算するように加算器をコンフィギュレーションできます。加算または減算は、コンパイル時に選択する必要があります。Two-Multiplier Adder ファンクションは FFT、複素数 FIR、IIR フィルタなどのアプリケーションに有効です。図 5-14 に、Two-Multiplier Adder モードでコンフィギュレーションされた DSP ブロックを示します。

ループバック・モードは、Two-Multiplier Adder モードの別のサブ機能です。図 5-15 に、ループバック・モードでコンフィギュレーションされた DSP ブロックを示します。このモードは、2 個の乗算器の 36 ビット加算結果を取り込んで、最上位 18 ビットを入力にフィードバックします。下位 18 ビットは破棄されます。ダイナミック zero_loopback 信号を使用して、ループバック・データを無効にする、すなわちゼロとするオプションがあります。zero_loopback 信号の値が logic 1 の場合、zeroed データが選択され、ループバックが無効になりますが、logic 0 ではループバックされたデータが選択されます。

 ループバック・モードを使うか、汎用の Two-Multiplier Adder モードを使うかのオプションは、コンパイル時に選択する必要があります。

Two-Multiplier Adder モードでは、すべての入力が必要な 18 ビットおよび符号なしの場合、結果には 37 ビットが必要です。Two-Multiplier Adder モードの出力データ幅は 36 ビットに制限されているため、この 37 ビット出力は実現できません。最大結果が 36 ビットに収まるその他の組み合わせは許可されます。例えば、2 個の 16×16 符号付き Two-Multiplier Adder は実現可能です。

Two-Multiplier Adder モードは、丸めおよび飽和論理ユニットをサポートします。DSP ブロック内でパイプライン・レジスタと出力レジスタを使用すると、乗算器・加算器の結果をパイプラインでできるため、DSP ブロックの性能が向上します。

図 5-14. ハーフ DSP ブロックの Two-Multiplier Adder モード

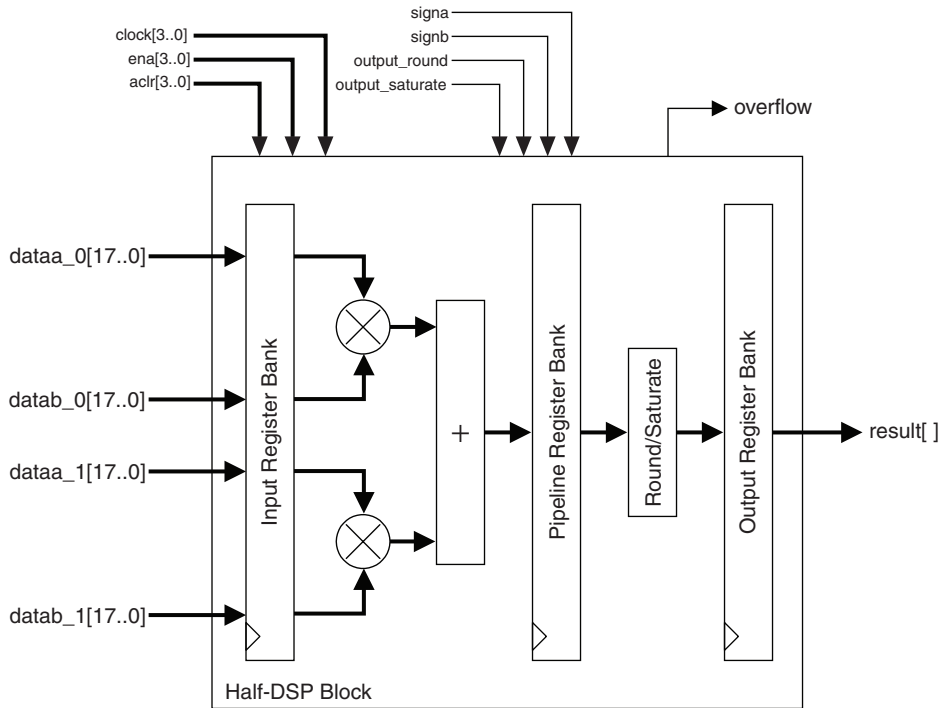
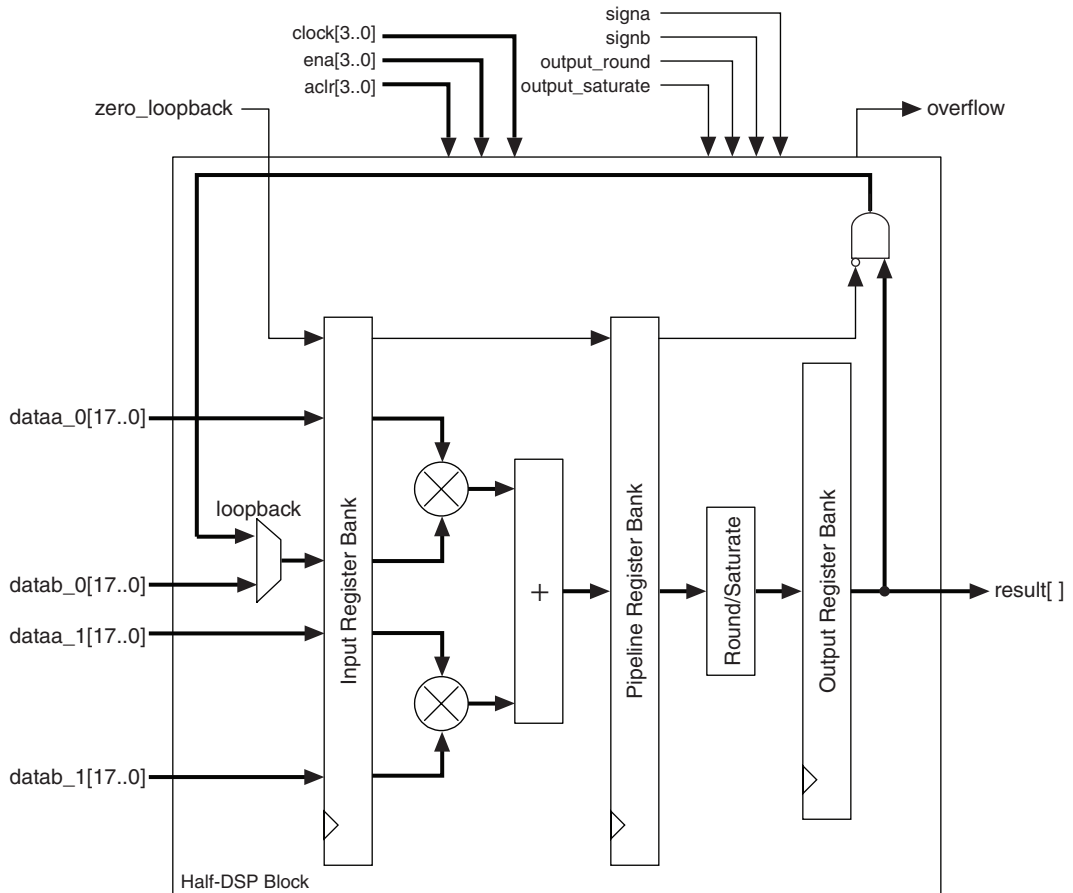


図 5-15. ハーフ DSP ブロックのループバック・モード



18 × 18 複素数乗算

Two-Multiplier Adder モードで使用する場合、Two-Multiplier Adder モードを使用して複素数乗算器を実装するように、DSP ブロックをコンフィギュレーションできます。単一のハーフ DSP ブロックは、1 個の 18 ビット複素数乗算器を実装できます。

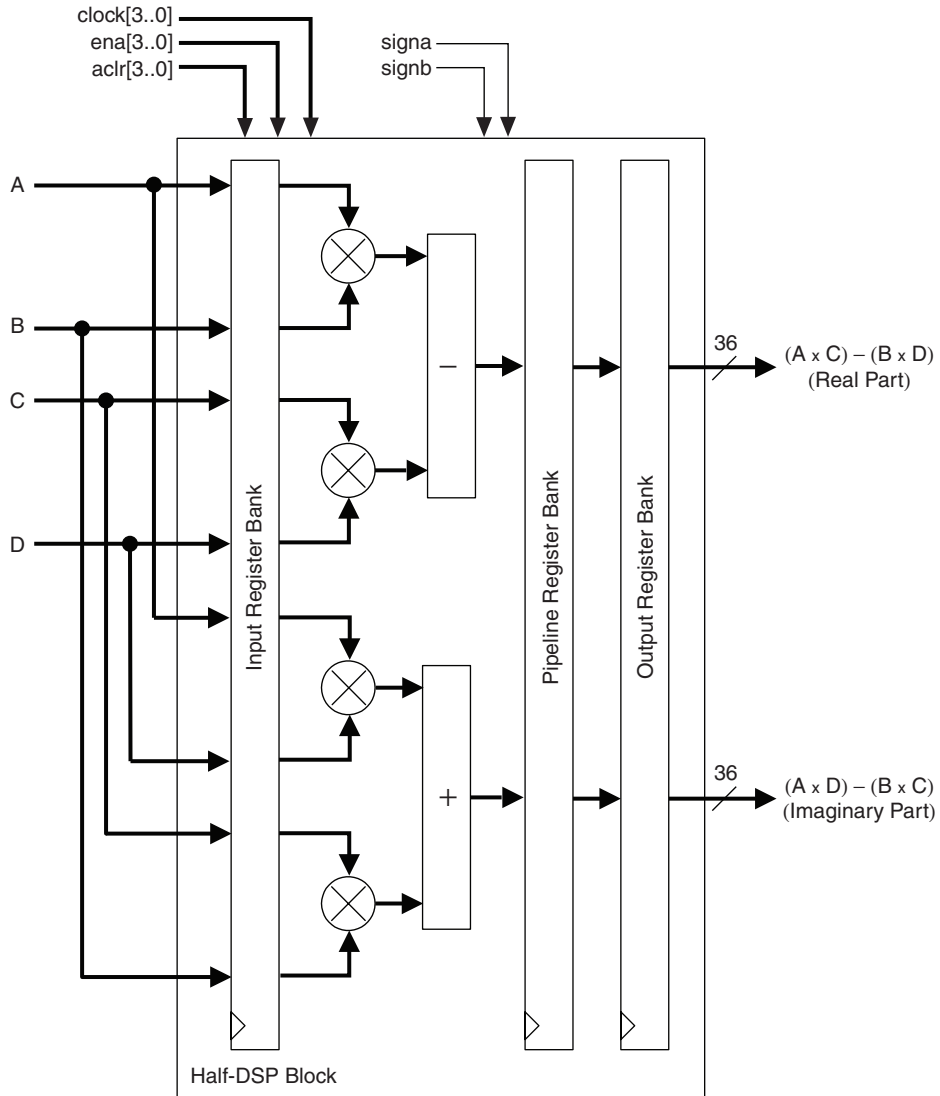
複素数乗算は、式 5-4 に示すように記述できます。

式 5-4. 複素数乗算式

$$(a + jb) \times (c + jd) = ((a \times c) - (b \times d)) + j((a \times d) + (b \times c))$$

DSP ブロック内でこの複素数乗算を実装する場合、実数部 $((a \times c) - (b \times d))$ は 1 つの減算器ブロックに供給する 2 個の乗算器を使用して実装され、仮数部 $((a \times d) + (b \times c))$ は、1 つの加算器ブロックに供給する別の 2 個の乗算器を使って実装されます。図 5-16 に、18 ビット複素数乗算を示します。このモードでは、自動的にすべての入力符号付き数値を使用していると仮定します。

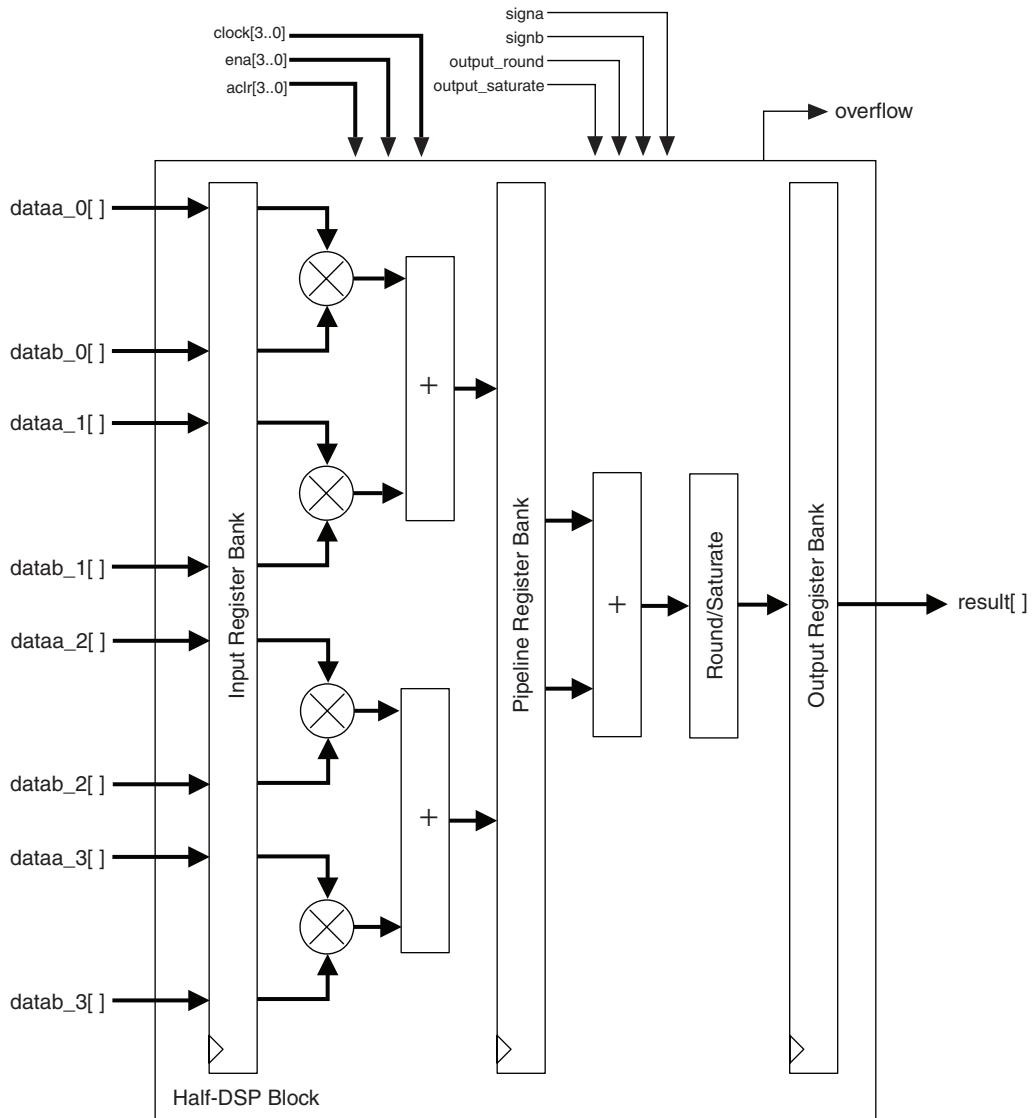
図 5-16. Two-Multiplier Adder モードを使用した複素数乗算器



Four-Multiplier Adder

図 5-17 に示す Four-Multiplier Adder コンフィギュレーションでは、DSP ブロックは 2 個の Four-Multiplier Adder (ハーフ DSP ブロックあたり 1 個の Four-Multiplier Adder) を実装できます。これらのモードは、1 次元および 2 次元フィルタリング・アプリケーションの実装に便利です。Four-Multiplier Adder は、2 つの加算ステージで実行されます。最初に、4 個の乗算器の 2 つの出力が、2 個の第 1 ステージ加算器ブロックで総和されます。式 5-2 と式 5-3 に示されるように、これらの 2 個の加算器ブロックの結果は、第 2 ステージ加算器ブロックで総和され、最終的な Four-Multiplier Adder の結果を生成します。

図 5-17. ハーフ DSP ブロックの Four-Multiplier Adder モード

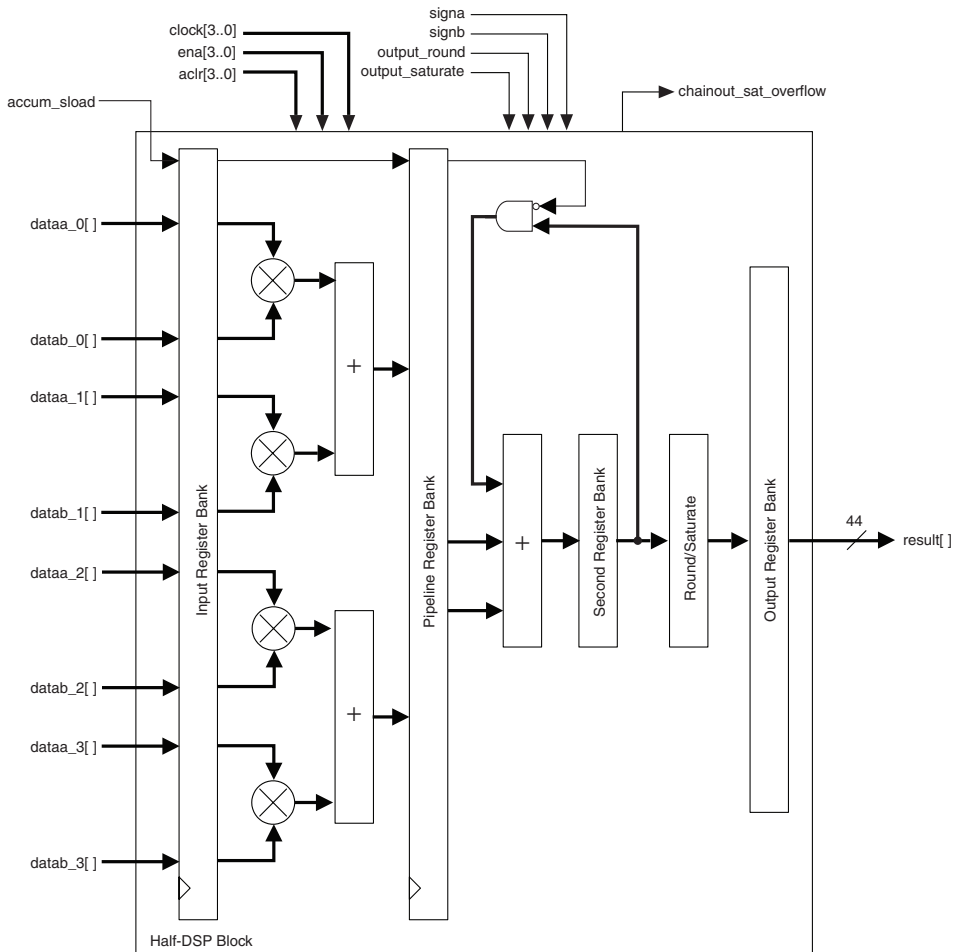


Four-Multiplier Adder モードは、丸めおよび飽和論理ユニットをサポートします。DSP ブロック内でパイプライン・レジスタと出力レジスタを使用すると、乗算器・加算器の結果をパイプラインできるため、DSP ブロックの性能が向上します。

Multiply Accumulate モード


Multiply Accumulate モードでは、第 2 ステージの加算器は 44 ビット・アキュムレータまたは減算器としてコンフィギュレーションされます。DSP ブロックの出力は、第 2 ステージの加算器にループバックされ、式 5-3 に従って第 1 ステージの加算器ブロックの 2 つの出力と加算または減算されます。図 5-18 に、Multiply Accumulate モードで動作するようにコンフィギュレーションされた DSP ブロックを示します。

図 5-18. ハーフ DSP ブロックの Multiply Accumulate モード



単一 DSP ブロックは、最大 2 個の独立した 44 ビット・アキュムレータを実装できます。

累積のクリアには、ダイナミック `accum_sload` コントロール信号が使用されます。`accum_sload` 信号に `logic 1` 値がある場合は、アキュムレータに乗算器の結果のみ同期的にロードされ、`logic 0` がある場合は、DSP ブロックの出力が乗算器と第 1 ステージ加算器の出力に加算または減算され(アキュムレータ・フィードバック)累積が有効になります。


 アキュムレータおよび減算器のコントロール信号はスタティックなので、コンパイル時にコンフィギュレーションする必要があります。

このモードは、18 ビット乗算アキュムレータとしてコンフィギュレーションされているため、丸めおよび飽和論理ユニットをサポートします。DSP ブロック内でパイプライン・レジスタと出力レジスタを使用すると、DSP ブロックの性能が向上します。

シフト・モード

Stratix III デバイスは、32 ビット入力に対してのみ以下のシフト・モードをサポートします。

- 算術左シフト、ASL[N]
- 算術右シフト、ASR[32-N]
- 論理左シフト、LSL[N]
- 論理右シフト、LSR[32-N]
- 32 ビット・ローテータまたはバレル・シフタ、ROT[N]

 ダイナミック・ローテイト信号およびシフト・コントロール信号を使用すると、これらのモード間でシフト・モードを切り替えることができます。

Nios® II などのソフト・エンベデッド・プロセッサは、Stratix III デバイスのシフト・モードを使って、ダイナミック・シフトおよびローテイト操作を簡単に実行できます。図 5-19 に、シフト・モード・コンフィギュレーションを示します。

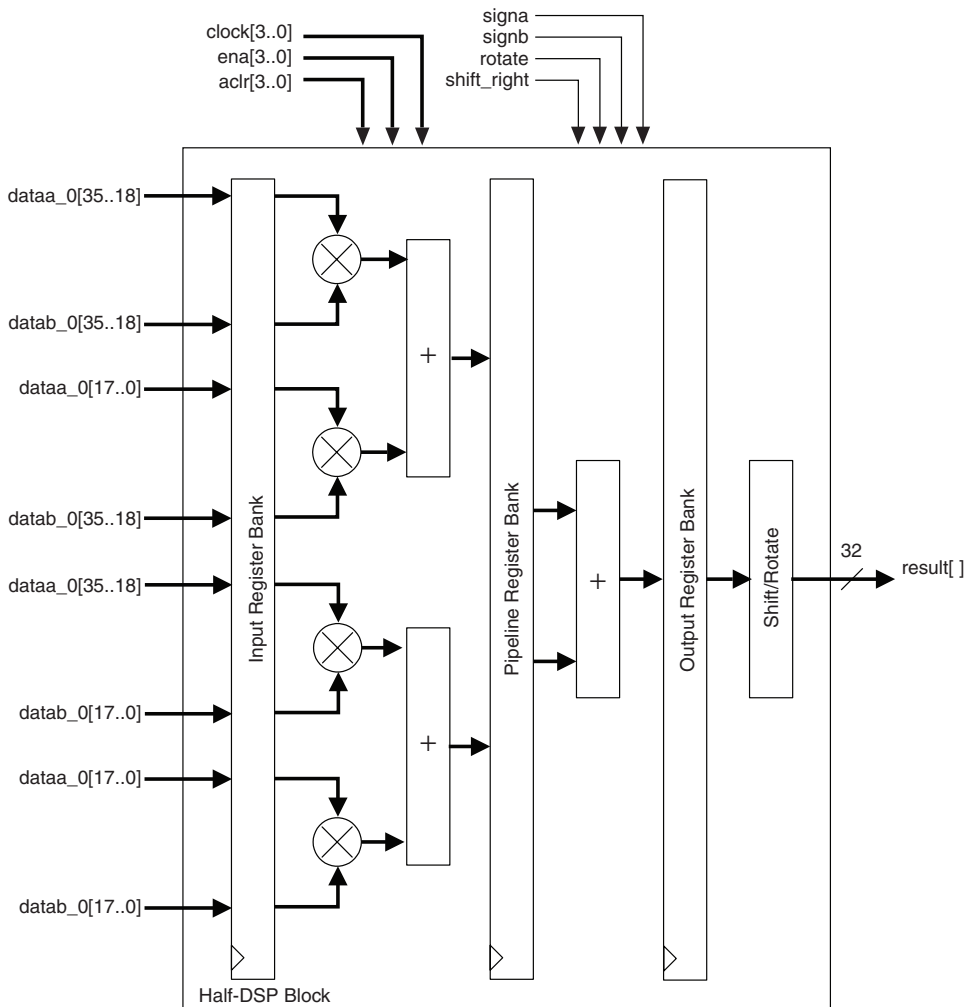
シフト・モードでは、利用可能な乗算器を使用して、必要な 32 ビット・データの論理または算術左、右シフト、またはローテイト操作を実行できます。DSP ブロックは、独立した 36 ビット乗算器モードと同様に、シフト・モード操作を実行するようにコンフィギュレーションされます。

算術右シフトには、符号付き入力ベクタが必要です。算術右シフト中に、32 ビット・ベクタの MSB まで符号拡張されます。論理右シフトでは、符号なし入力ベクタが使用されます。論理右シフト中、最上位ビットに

ゼロがパディングされ、32 ビット・ベクタが右にシフトします。パレル・シフトは、符号なし入力ベクタを使用し、32 ビット・ワード長で回転ファンクションを実行します。

2つのコントロール信号 rotate および shift_right が、signa および signb 信号と共にシフト操作を決定します。シフト操作の例を表 5-5 に示します。

図 5-19. ハーフ DSP ブロックのシフト操作モード



例	Signa	Signb	シフト	ロー テイト	A 入力	B 入力	結果
論理左シフト LSL [N]	符号なし	符号なし	0	0	0xAABBCCDD	0x0000100	0xBBCCDD00
論理右シフト LSR [32-N]	符号なし	符号なし	1	0	0xAABBCCDD	0x0000100	0x000000AA
算術左シフト ASL [N]	符号付き	符号なし	0	0	0xAABBCCDD	0x0000100	0xBBCCDD00
算術右シフト ASR [32-N]	符号付き	符号なし	1	0	0xAABBCCDD	0x0000100	0xFFFFF0AA
ローテイト ROT [N]	符号なし	符号なし	0	1	0xAABBCCDD	0x0000100	0xBBCCDDAA

丸めおよび飽和モード

丸めおよび飽和ファンクションは、DSP 演算で頻繁に要求されます。丸め処理は、ビット増加とその副作用を制限するために使用され、飽和処理はオーバフローとアンダフローの副作用を抑えるために使用されます。

Stratix III デバイスでは、次の2つの丸めモードがサポートされています。

- Round-to-Nearest-Integer モード
- Round-to-Nearest-Even モード

この2つのオプションのいずれかをコンパイル時に選択する必要があります。

Round-to-Nearest-Integer は、バイアスされた丸め処理をサポートし、DSP 演算で一般に使用される丸め処理の最も簡単な形式です。Round-to-Nearest-Even 方式は、バイアスのない丸め処理をサポートし、DC オフセットが問題になる場合に使用されます。表 5-6 に、Round-to-Nearest-Even の動作を示します。この2つのモードの違いの例を表 5-7 に示します。この例では、6 ビット入力が4 ビットに丸められます。表 5-7 から、2つの丸め処理オプションの主な違いは、残余ビットが直近の2つの整数のちょうど中間にあること、および LSB がゼロ(偶数)であることが分かります。

表 5-6. Round-to-Nearest-Even モードの例

6 ビットから 4 ビットの丸め	奇数 / 偶数 (整数)	小数	整数への 加算	結果
010111	x	> 0.5 (11)	1	0110
001101	x	< 0.5 (01)	0	0011
001010	偶数 (0010)	= 0.5 (10)	0	0010
001110	奇数 (0011)	= 0.5 (10)	1	0100
110111	x	> 0.5 (11)	1	1110
101101	x	< 0.5 (01)	0	1011
110110	奇数 (1101)	= 0.5 (10)	1	1110
110010	偶数 (1100)	= 0.5 (10)	0	1100

表 5-7. Round-to-Nearest-Integer と Round-to-Nearest-Even の比較

Round-to-Nearest-Integer	Round-to-Nearest-Even
010111 ⇒ 0110	010111 ⇒ 0110
001101 ⇒ 0011	001101 ⇒ 0011
001010 ⇒ 0011	001010 ⇒ 0010
001110 ⇒ 0100	001110 ⇒ 0100
110111 ⇒ 1110	110111 ⇒ 1110
101101 ⇒ 1011	101101 ⇒ 1011
110110 ⇒ 1110	110110 ⇒ 1110
110010 ⇒ 1101	110010 ⇒ 1100

Stratix III では、2つの飽和モードがサポートされます。

- 非対称飽和モード
- 対称飽和モード

この2つのオプションのいずれかをコンパイル時に選択する必要があります。

2 の補数形式では、表現可能な最大負数は $-2^{(n-1)}$ 、最大正数は $2^{(n-1)}-1$ です。対称飽和は、最大負数を $-2^{(n-1)} + 1$ に制限します。例えば、32 ビットの場合：

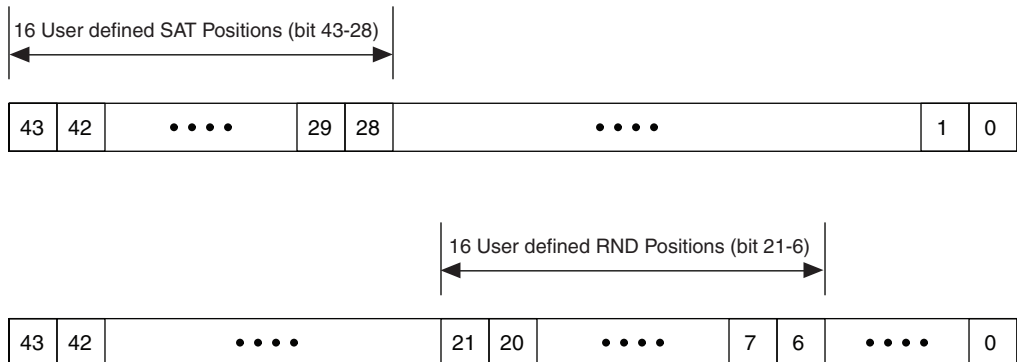
- 非対称 32 ビット飽和：最大 = 0x7FFFFFFF、最小 = 0x80000000
- 対称 32 ビット飽和：最大 = 0x7FFFFFFF、最小 = 0x80000001


表 5-8 に、飽和の動作を示します。この例では、44 ビット入力が 36 ビットに飽和されます。

表 5-8. 飽和の例		
44 ビットから 36 ビットへの飽和	対称 SAT の結果	非対称 SAT の結果
5926AC01342h	7FFFFFFFh	7FFFFFFFh
ADA38D2210h	80000001h	80000000h

Stratix III デバイスは、丸めおよび飽和論理ユニットで柔軟性を高めるには、44 ビット・バス ([43:0]) のうち最大 16 ビット位置をコンフィギュレーションできます。16 のコンフィギュレーション可能なビット位置は、コンパイル時に選択する必要があります。これらの 16 ビット位置は、[図 5-20](#) に示すように、丸め処理ではビット [21:6]、飽和处理では [43:28] を占めます。

図 5-20. 丸めおよび飽和の位置



 対称飽和の場合、飽和したデータの LSP の配置を決定するために、RND ビット位置も使用されます。

上記の丸めおよび飽和ファンクションは、表 5-2 で規定されるとおり、通常サポートされる乗算演算で使用できます。ただし、累積タイプの演算では、以下の規約が使用されます。

丸め論理ユニットの機能は、次の形式をとります。

結果 = $\text{RND}[\sum(A \times B)]$ 、累積タイプの演算に使用される場合

同様に、飽和論理ユニットの機能は、次の形式をとります。

結果 = $\text{SAT}[\sum(A \times B)]$ 、累積タイプの演算に使用される場合

丸めおよび飽和論理ユニットの両方が累積タイプの演算に使用される場合は、次のような形式になります。

結果 = $\text{SAT}[\text{RND}[\sum(A \times B)]]$

DSP ブロックのコントロール信号

Stratix III DSP ブロックは、一連のスタティック信号とダイナミック信号を使用してコンフィギュレーションされます。DSP ブロックのダイナミック信号は、ユーザがコンフィギュレーション可能で、実行時にトグルの有無を設定できます。DSP ブロックのダイナミック信号のリストを、表 5-9 に示します。

信号名	機能	数
<ul style="list-style-type: none"> ● signa ● signb 	すべての乗算器および加算器に対する符号付き / 符号なしコントロール signa: 各乗算器への dataa[17:0] 入力用 signb: 各乗算器への datab[17:0] 入力用 signa = 1、signb = 1、符号付 - 符号付乗算 signa = 1、signb = 0、符号付 - 符号なし乗算 signa = 0、signb = 1、符号なし - 符号付乗算 signa = 0、signb = 0、符号なし - 符号なし乗算	2
output_round	第 1 ステージの丸め / 飽和ブロックに対する丸めコントロール。 output_round = 1、乗算出力の丸め処理 output_round = 0、通常の乗算出力	1
chainout_round	第 2 ステージの丸め / 飽和ブロックの丸めコントロール。 chainout_round = 1、乗算出力の丸め処理。 chainout_round = 0、通常の乗算出力	1

表 5-9. DSP ブロックのダイナミック信号 (2 / 2)		
信号名	機能	数
output_saturate	Q 形式乗算に対する第 1 ステージの丸め / 飽和ブロックの飽和コントロール。丸めと飽和の両方がイネーブルされている場合、丸め処理の結果で飽和が実行されます。 output_saturate = 1、飽和サポート output_saturate = 0、飽和サポートなし	1
chainout_saturate	Q 形式乗算に対する第 2 ステージの丸め / 飽和ブロックの飽和コントロール。丸めと飽和の両方がイネーブルされている場合、丸め処理の結果で飽和が実行されます。 chainout_saturate = 1、飽和サポート chainout_saturate = 0、飽和サポートなし	1
accum_sload	アキュムレータ値をゼロにするかどうかをダイナミックに指定します。accum_sload = 0、累積入力を出力レジスタの値とする accum_sload = 1、累積入力をゼロに設定	1
zero_chainout	チェーンアウト値をゼロにするかどうかをダイナミックに指定します。	1
zero_loopback	ループバック値をゼロにするかどうかをダイナミックに指定します。	1
rotate	rotation = 1、ローテイト機能がイネーブル	1
shift_right	shift_right = 1、右シフト機能がイネーブル	1
	ハーフ・ブロックあたりの信号数	11
clock0 clock1 clock2 clock3	DSP-block-wide クロック信号	4
ena0 ena1 ena2 ena3	入力およびパイプライン・レジスタ・イネーブル信号	4
aclr0 aclr1 aclr2 aclr3	DSP block-wide 非同期クリア信号 (アクティブ Low)	4
	フル・ブロックあたりのトータル・カウント	34

アプリケーション例

FIR の例

有限インパルス応答フィルタは、多くのシステムでスペクトル乗算の実行に使用される一般的なファンクションです。基本的な形式を式 5-5 に示します。

式 5-5. 有限インパルス応答フィルタの計算式

$$y(n) = \sum_{k=0}^{N-1} x(n-k) \times c(k)$$

この式では、 $x(n)$ はフィルタへの入力サンプル、 $c(k)$ はフィルタ係数、 $y(n)$ はフィルタされた出力サンプルです。通常、DDC (Digital Down Converter) などのほとんどのアプリケーションで、この係数は時間が経過しても変化しません。FIR フィルタは、多くの形式で実装できますが、最も簡単な形式はタップ・ディレイ・ライン方式です。

Stratix III DSP ブロックは、各種タイプの FIR フィルタを非常に効率的に実装できます。タップ・ディレイ・ラインを形成するには、DSP ブロックの入力レジスタ・ステージには入力を 18 ビット幅形式によるチェイン方式でカスケードする機能があります。2 個のビルトイン・パラレル入力スキャン・パスを備えた Stratix II DSP ブロックとは異なり、Stratix III は 288 データ入力に対して 1 個のビルトイン 18 ビット・パラレル入力レジスタ・スキャン・パスのみサポートします。

1 組の 18 ビット入力バスに対して、最初の 18 ビット・バスの A 入力が入力フィードバックされて、2 番目 (下位) の入力ペアの入力時に再度ラッチされます。詳細は、図 5-21 を参照してください。

乗算器の B 入力は、汎用配線から供給されます。データを 18 ビット・パラレル形式でスキャン・インし、各サイクルで汎用配線からの 18 ビット入力バスで乗算することができます。

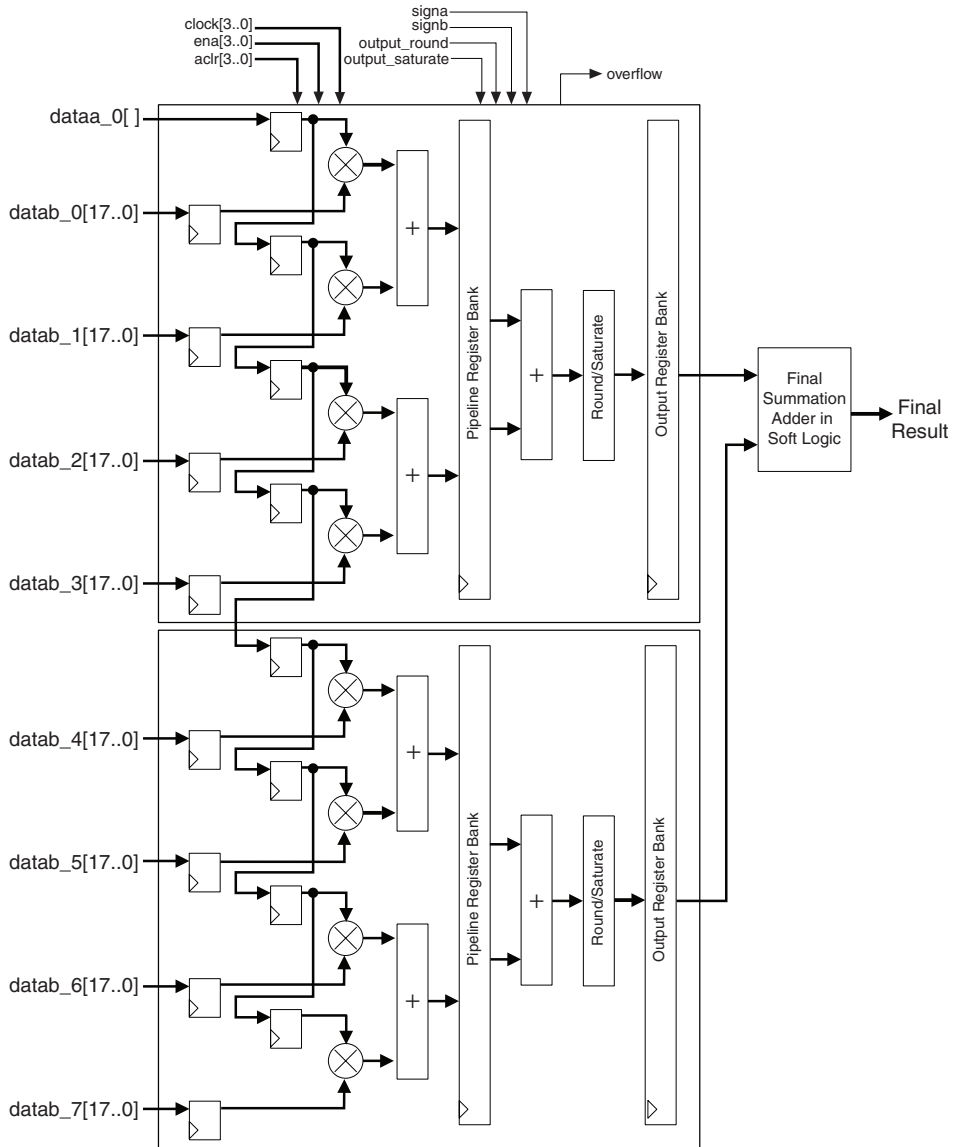
通常、FIR フィルタでは、(カスケードではなく汎用配線からの) 固定データ入力は、カスケード入力による乗算を必要とする定数になります。18 ビット・モードでは、DSP ブロックには汎用配線信号とカスケード信号バスをラッチしてから乗算するのに十分な数の入力レジスタがあります。このため、18 ビット・カスケード・モードでは 8 タップを持つことが可能になります。各タップは 1 個の乗算器と見なすことができます。完全な DSP ブロックの 8 つの乗算器入力が、パラレル・スキャン・チェインでカスケードされる場合は、図 5-21 に示すように、8 タップの FIR フィルタが作成されます。

次の（下位）DSP ブロックにパラレル・スキャン・チェーンを出力するオプションをイネーブルすることによって、DSP ブロックを連結して、9 つ以上のタップを持たせることができます。同様に、前の（上の）カスケード・チェーンの出力は、現在のブロックの入力として使用されます。各ハーフ・ブロックの最初（先頭）の乗算器には、通常の配線から、または前の（上の）カスケード・チェーンから 18 ビット・カスケード・チェーン入力を選択するオプションがあります。また、各ハーフ DSP ブロックの最後のカスケード・チェーンは、最後（先頭から 4 番目）の入力レジスタ以降のカスケード・チェーンを出力配線チャンネルに配線し、パイプライン・レジスタと出力レジスタの両方をバイパスすることによって DSP ブロックを抜けることができます。この連結により、ユーザーは必要な長さのフィルタを簡単に構築できます。

Four-Multiplier Adder モードは、各乗算器への入力の 1 つを前の（上の）レジスタからのチェーン・カスケード入力の形にして使用できます。この方法は、通常の Four-Multiplier Adder と非常によく似ていますが、汎用配線以外からの入力がある点が異なります。

完全な FIR では、個々の Four-Multiplier Adder ごとの結果をツリーまたはチェーン・カスケード方式で結合することができます。図 5-21 に示すように、外部ロジックと加算器を使用すると、ツリー加算を非常に簡単に実装できます。

図 5-21. タップ・ディレイ・ライン入力を使用した FIR フィルタと最終結果のツリー加算

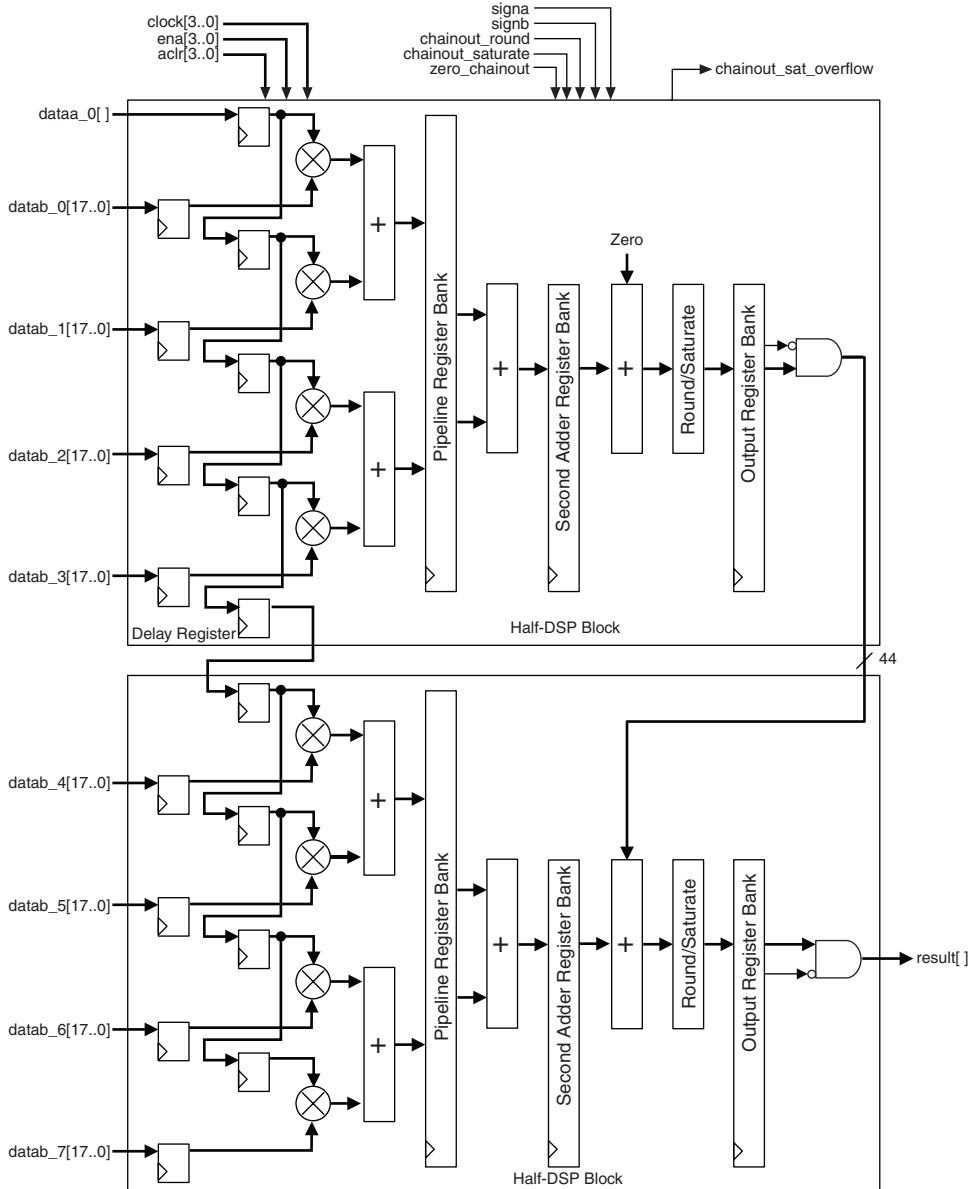


チェイン・カスケード加算を高速化および効率化するために、DSP ブロックはカスケード・モードでチェインアウト・ファンクションを実装できます。このモードは、[図 5-22](#) に示すように、第 2 ステージの 44 ビット加算器を使用して、-half DSP ブロックの現在の Four-Multiplier Adder を隣接する-half DSP ブロックの Four-Multiplier Adder に加算します。

この手法が可能になるのは、-half DSP ブロックそれぞれに 2 個の第 2 ステージ加算器があるためです。2 個の第 2 ステージ加算器の 1 つを使用して、現在の Four-Multiplier Adder を加算します。2 番目の第 2 ステージ加算器は、最初の第 2 ステージ加算器の出力を取り込み、それを隣接する-half DSP ブロックの Four-Multiplier Adder の結果に加算します。

[図 5-22](#) では、隣接する-half DSP ブロックを現在の Four-Multiplier Adder に加算する加算器がチェインアウト加算器であることを明確に示しています。この手法は複数の DSP ブロックをまとめてチェインおよび加算するのに使用されます。チェインアウト加算器の出力はラッチできます。ラッチされたチェインアウト出力は、チェインアウト加算器のために下方に隣接する DSP ブロックに供給するか、または汎用 FPGA 配線に供給できます。チェインアウト結果は、ダイナミック zerochainout 信号に logic 1 を印加することによってゼロアウトできます。zerochainout 信号はラッチすることも可能です。

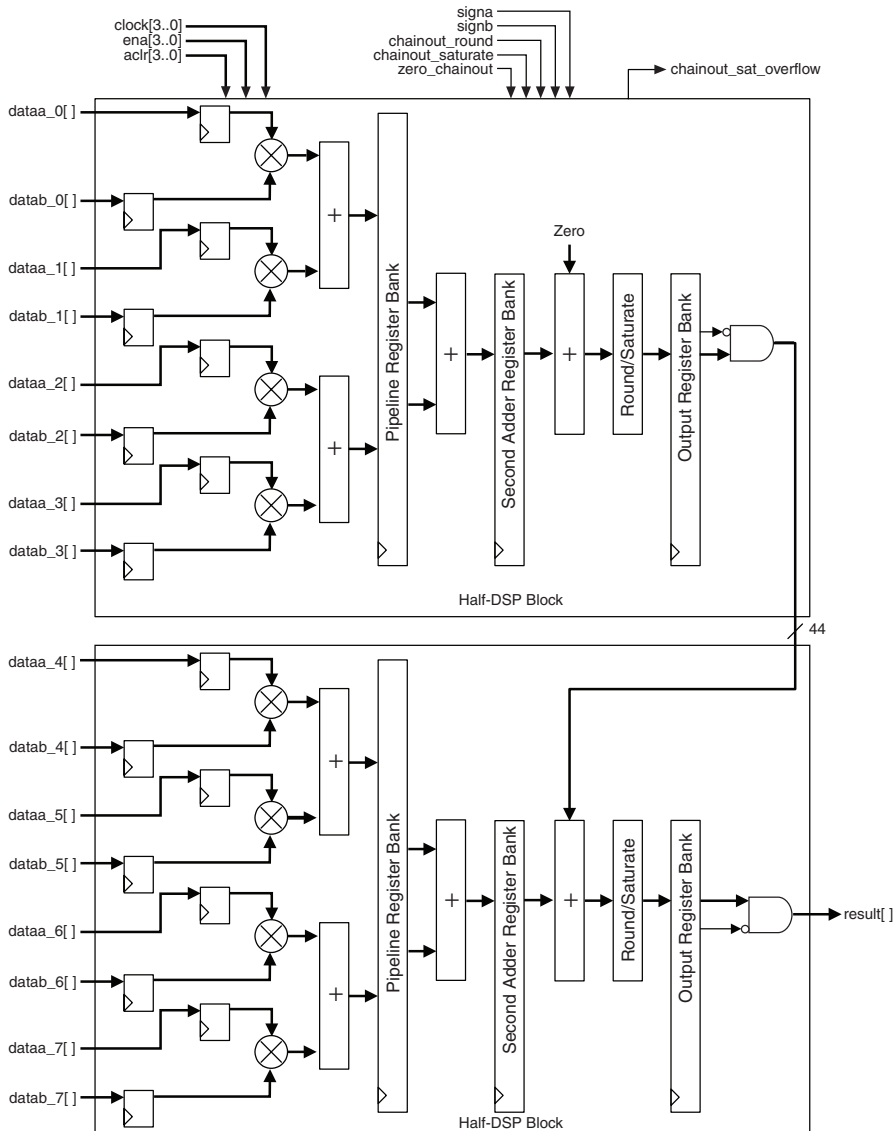
図 5-22. タップ・ディレイ・ライン入力を使用した FIR フィルタと最終結果のチェイン・カスケード加算



入力カスケードとチェインアウト機能の両方を使用する場合、DSP ブロックは各ハーフ DSP ブロックの境界で、またはブロック間の境界で 18 ビット・ディレイ・レジスタを使用して、入力スキャン・チェイン・データをチェインアウト・データに同期させます。上のハーフ・ブロックは積の総和を計算し、出力レジスタでラッチした後、次のブロックにチェインします。ディレイ・レジスタは、カスケード入力を 1 クロック・サイクル遅らせて、チェイン出力でのレイテンシを補正します。

システム・クロックが DSP ブロックの速度よりも遅いアプリケーションでは、効率を改善するために乗算器を時分割することができます。これにより、マルチチャンネルおよびセミ・パラレル FIR 構造が可能になります。これを達成するための構造は、[図 5-21](#) および [5-22](#) に似ています。主な違いは、入力カスケード・チェインが使用されず、ハーフ DSP ブロックがそれぞれ独立した入力を備えた Four-Multiplier モードで使用される点です。[図 5-23](#) に、チェイン・カスケード加算の例を示します。

図 5-23. チェイン・カスケード総和を使用したセミ・パラレル FIR 構造

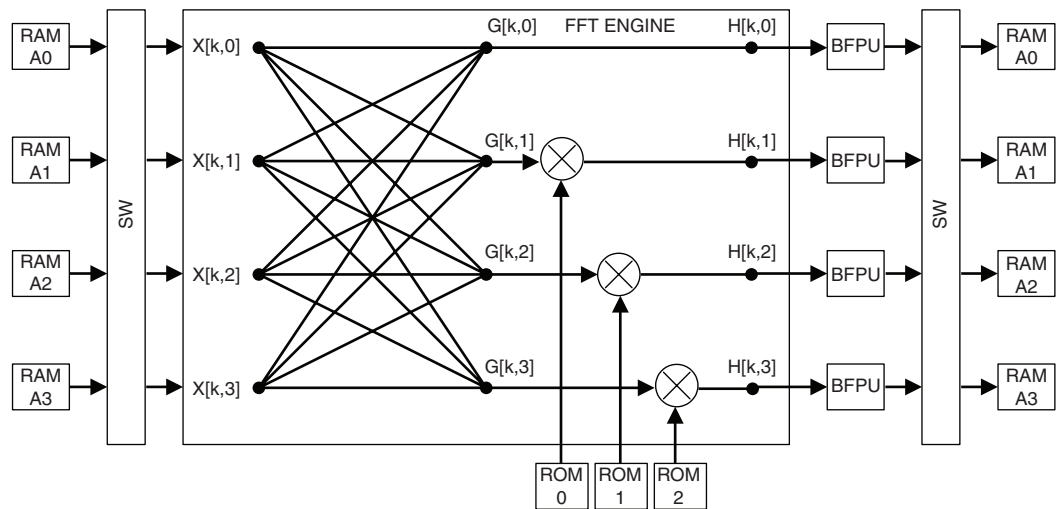


FFT の例

高速フーリエ変換（FFT）は、タイム・ドメインのサンプルを周波数ドメインに、またはその逆に変換するのに使用される非常に一般的な DSP ファンクションです。FFT の基本ビルディング・ブロックは FFT バタフライです。本質的に複素信号を扱う演算ですが、Stratix III DSP ブロックを使用すると、複素 FFT バタフライのコアを非常に効率的に形成できます。

図 5-24 に、基数 4 のバタフライを示します。各バタフライは 3 個の複素乗算器を必要とします。これは、Stratix III では、データおよび回転子 (Twiddle) ワード長が 18 ビット以下と仮定すると、3 個のハーフ DSP ブロックを使用して実装できます。

図 5-24. 基数 4 のバタフライ



ソフトウェア・サポート

アルテラは、DSP ブロックの各種モードをデザインに実装するための 2 つの主な方法として、インスタンス化と推論を提供しています。両方の方法とも、次の Quartus II メガファンクションを使用します。

- lpm_mult
- altmult_add
- altmult_accum
- altfp_mult

Quartus II ソフトウェアのメガファンクションのインスタンス化により、DSP ブロックを使用することができます。あるいは、推論を使用する場合は、HDL デザインを作成し、サードパーティ製の合成ツール (LeonardoSpectrum、Synplify など) もしくは Quartus II を使用して合成します。この合成ツールは、乗算器、乗算器・加算器、乗算アキュムレータ、シフト・ファンクションを認識することによって、適切なメガファンクションを推論します。Quartus II ソフトウェアは、いずれかの方法を使用して、コンパイル時に DSP ブロックに機能をマップします。



メガファンクションと「MegaWizard Plug-In Manager」の使用方法については、「Quartus II ソフトウェア・ヘルプ」を参照してください。



詳しくは、「Quartus II ソフトウェア・ハンドブック」Volume 1 の「合成」セクションを参照してください。

まとめ

Stratix III デバイスの DSP ブロックは、FIR フィルタ、IIR フィルタ、FFT ファンクション、エンコーダなどの高データ・スループットを要求する DSP アプリケーションをサポートするように最適化されています。これらの DSP ブロックは柔軟性があり、特定のアプリケーションに合わせて複数の動作モードの 1 つを実装するようにコンフィギュレーションできます。ビルトイン・シフト・レジスタ・チェーン、乗算器、加算器 / 減算器は、これらのファンクションを実装するのに必要な外部ロジック量を少なくして、DSP アプリケーションでのリソース利用を効率化し、性能とデータ・スループットを改善します。Quartus II ソフトウェアは、LeonardoSpectrum や Synplify ソフトウェアと併用すれば、これらの乗算器ファンクションを DSP ブロックで実装するための完全で使いやすいフローを提供します。

改訂履歴

表 5-10 に、本資料の改訂履歴を示します。

表 5-10. 改訂履歴		
日付 & ドキュメント・バージョン	変更内容	概要
2007 年 5 月 v1.1	図 1 ~ 21 の信号名を更新。2 点の新しい図、 図 5-12 および 図 5-13 を追加。 図 5-18 を更新。 表 5-5 および 表 5-9 を更新。 表 5-10 を削除。5-23 ページの「 ダブル乗算器 」を追加。 5-34 ページ の「 シフト・モード 」に説明を追加。	—
2006 年 11 月 v1.0	初版	—