

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。

AGX52010-1.0

はじめに

Arria™ GX デバイスは、高データ・スループットが要求される DSP アプリケーションに最適化された専用のデジタル信号処理 (DSP) ブロックを備えています。DSP ブロックとプログラマブル・ロジック・デバイス (PLD) の柔軟性を組み合わせることにより、様々な高性能 DSP ファンクションを簡単に実装することができます。CDMA2000、Voice over Internet Protocol (VoIP)、高精細テレビ (HDTV) などの複雑なシステムは、データを処理するために高性能 DSP ブロックを必要とします。これらのシステム・デザインは一般的に、有限インパルス応答 (FIR) フィルタ、複合 FIR 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)、および相関器として DSP ブロックを使用します。

Arria GX の DSP ブロックは、乗算、減算、累算、および加算演算を行う専用ブロックの組み合わせで構成されています。これらのブロックをコンフィギュレーションして、ほとんどの DSP ファンクションに必要な乗算器、乗算加算器および乗算アキュムレータなどの演算ファンクションを実装できます。

DSP ブロックに加え、TriMatrix™ メモリ構造も様々なソフト・マルチプライヤの実装をサポートしています。ソフト・マルチプライヤと専用 DSP ブロックの組み合わせによって、Arria GX デバイスではより多くの乗算器が使用でき、システム設計時に幅広い実装オプションと柔軟性が得られます。



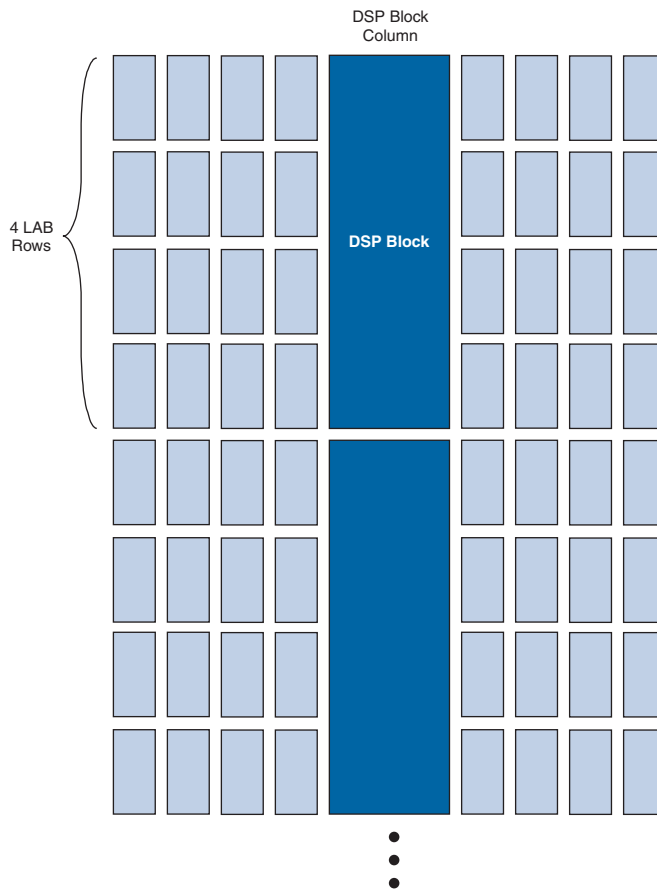
Arria GX デバイスについて詳しくは、「Arria GX デバイス・ハンドブック Volume 1」の「Arria GX デバイス・ファミリー・データシート」を参照してください。

DSP ブロックの概要

Arria GX デバイスは、乗算、乗算累積 (MAC) および積和ファンクションを効率的に実装する 2 ~ 4 カラムの DSP ブロックを備えています。図 10-1 に、DSP ブロック・カラムとその周りの LAB を示します。各 DSP ブロックは、以下をサポートするようにコンフィギュレーションできます。

- 8 個の 9×9 ビット乗算器
- 4 個の 18×18 ビット乗算器
- 1 個の 36×36 ビット乗算器

図 10-1. 縦列に配置された DSP ブロックと隣接する LAB



乗算器は、DSP ブロック内の加算器またはアキュムレータ・ブロックにデータを供給します。Arria GX デバイスの乗算器は、Q1.15 入力フォーマットでの丸めおよび飽和機能をサポートしています。また、DSP ブロックは、FIR フィルタなどのファンクションを効率的に実装するために、シフト・レジスタ・チェーンで動作するようにコンフィギュレーション可能な入力レジスタを備えています。DSP ブロック内のアキュムレータは任意の値に初期化でき、乗算器への Q1.15 入力フォーマットの丸めおよび飽和機能をサポートしています。シングル DSP ブロックは分割して、異なるコンフィギュレーション・モードで同時に動作させることができます。



Q1.15 フォーマットについて詳しくは、10-11 ページの「飽和および丸め機能」の項を参照してください。

1 カラムあたりの DSP ブロック数と使用可能なカラム数は、デバイスの集積度が大きくなるにつれて増加します。

表 10-1 に、Arria GX デバイスの DSP ブロック数と実装可能な乗算器数を示します。

デバイス	DSP ブロック数	9 × 9 乗算器	18 × 18 乗算器	36 × 36 乗算器
EP1AGX20C	10	80	40	10
EP1AGX35C/D	14	112	56	14
EP1AGX50C/D	26	208	104	26
EP1AGX60C/D/E	32	256	128	32
EP1AGX90E	44	352	176	44

表 10-1 の注:

- (1) 各デバイスには、記載した数の 9 × 9、18 × 18、または 36 × 36 ビット乗算専用回路があります。各デバイスの乗算器の総数はすべての乗算器の合計数ではありません。

DSP ブロック乗算器に加え、Arria GX デバイスの TriMatrix メモリ・ブロックをソフト・マルチプライヤとして使用できます。表 10-2 に、DSP ブロックおよびソフト・マルチプライヤを使用した Arria GX デバイスで利用可能な乗算器の総数を示します。

デバイス	DSP ブロック (18 × 18)	ソフト・マルチプライヤ (16 × 16) (1)、(2)	乗算器の総数 (3)、(4)
EP1AGX20C	40	102	142 (3.55)
EP1AGX35C/D	56	122	178 (3.18)
EP1AGX50C/D	104	202	306 (2.94)
EP1AGX60C/D/E	128	211	339 (2.65)

表 10-2. Arria GX デバイスの乗算器数 (2 / 2)

デバイス	DSP ブロック (18 × 18)	ソフト・マルチプライヤ (16 × 16) (1)、(2)	乗算器の総数 (3)、(4)
EP1AGX90E	176	324	500 (2.84)

表 10-2 の注:

- (1) ソフト・マルチプライヤは乗算モードの和で実装されています。RAM ブロックは、18 ビット・データ幅と最大 18 ビットの係数の和でコンフィギュレーションされます。
- (2) ソフト・マルチプライヤは、M4K および M512 TriMatrix メモリ・ブロックにのみ実装されます。M-RAM ブロックには実装できません。
- (3) 括弧内の数値は増加係数を示しています。これは、ソフト・マルチプライヤを使用した乗算器の総数を DSP ブロックでのみサポートされている 18 × 18 乗算器の数で除算した値です。
- (4) 乗算器の総数は、使用される乗算器モードによって異なります。



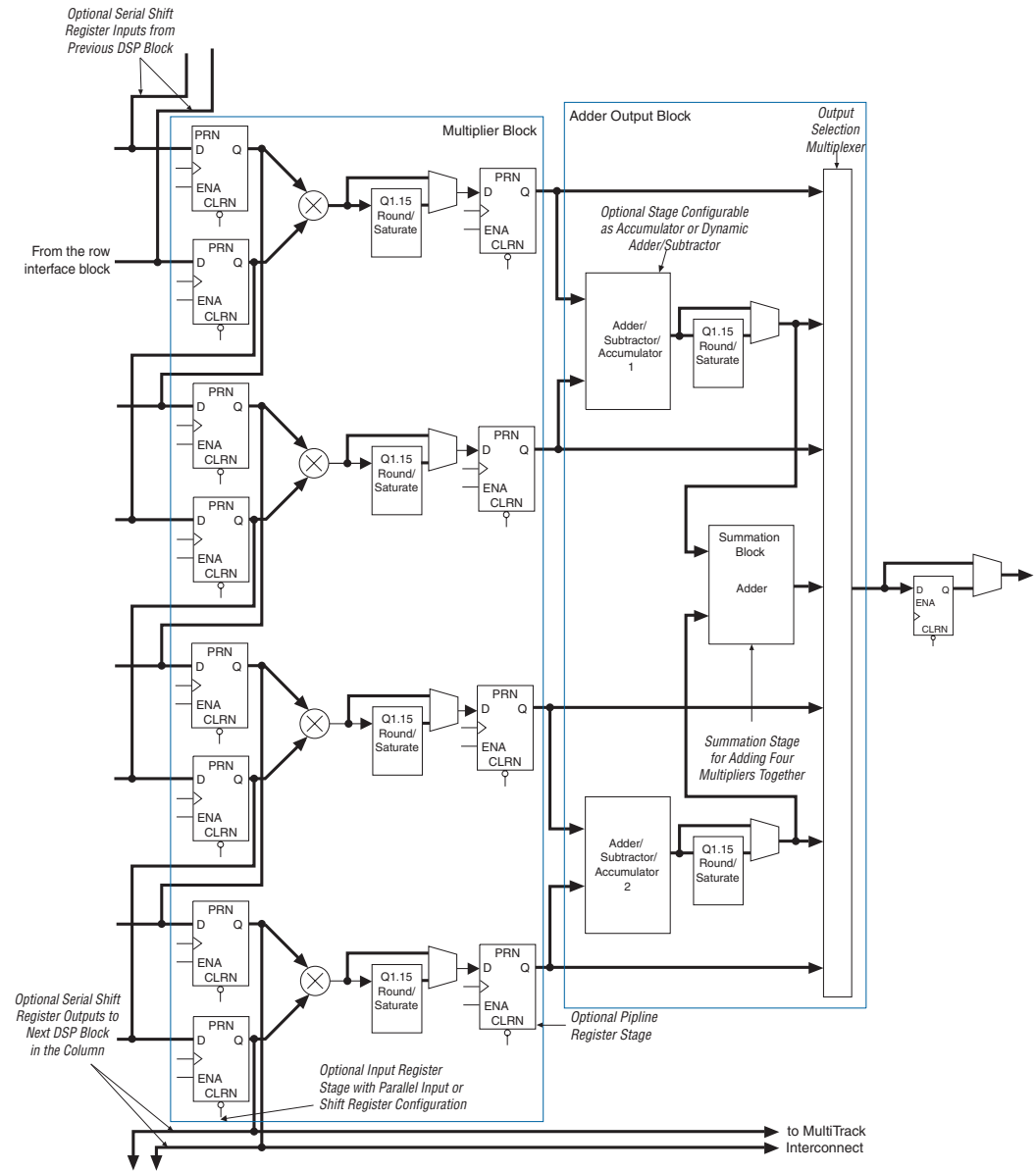
Arria GX の TriMatrix メモリ・ブロックについて詳しくは、「Arria GX デバイス・ハンドブック Volume 1」の「Arria GX デバイス・アーキテクチャ」の章を参照してください。



ソフト・マルチプライヤについて詳しくは、「AN306: Implementing Multipliers in FPGA Devices」を参照してください。

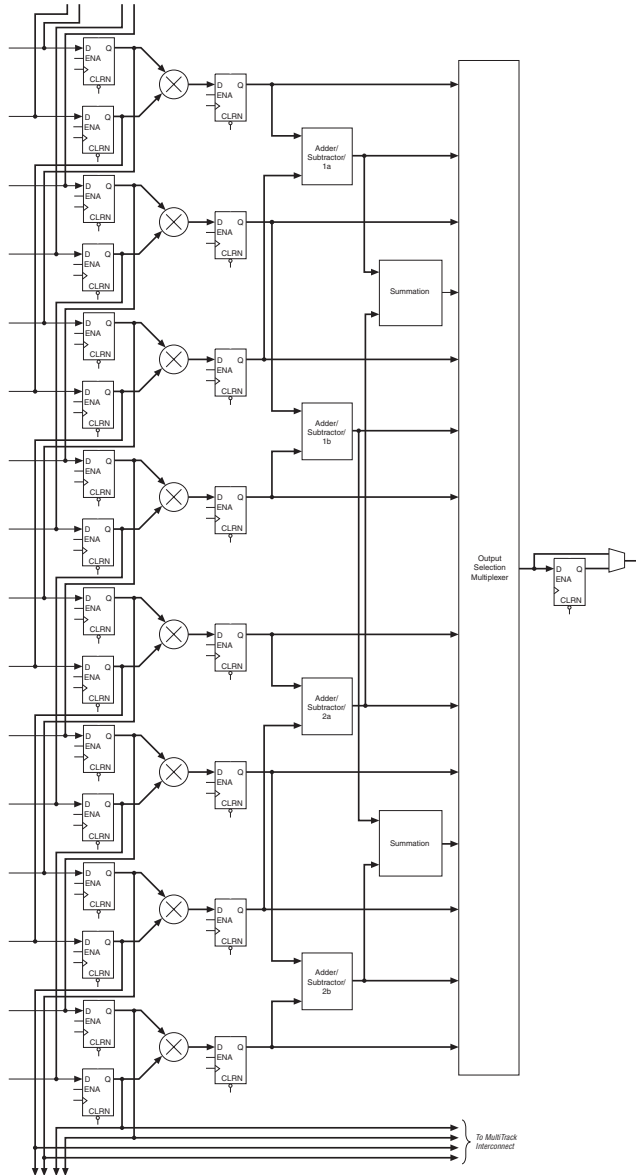
図 10-2 に、18 × 18 乗算器モードにコンフィギュレーションされた DSP ブロックを示します。

図 10-2. 18 × 18 モードの DSP ブロック



また、図 10-3 に、DSP ブロックの 9×9 乗算器コンフィギュレーションを示します。

図 10-3. 9×9 モードの DSP ブロック



アーキテクチャ

DSP ブロックは以下の要素で構成されます。

- 乗算器ブロック
- 加算器 / 減算器 / アキュムレータ・ブロック
- サメーション・ブロック
- 入力および出力インターフェース
- 入力および出力レジスタ

乗算器ブロック

各乗算器ブロックは、以下の要素で構成されています。

- 入力レジスタ
- 乗算器ブロック
- Q1.15 フォーマットの丸めおよび / または飽和ステージ
- パイプライン出力レジスタ

図 10-4 に、乗算器ブロック・アーキテクチャを示します。

図 10-4. 乗算器ブロック・アーキテクチャ

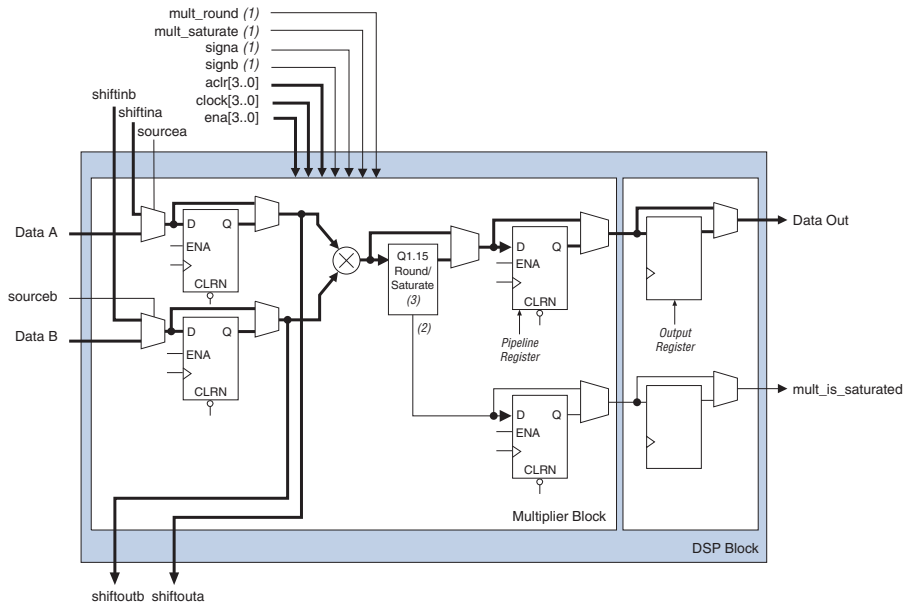


図 10-4 の注:

- (1) これらの信号は、データ・パス・パイプラインに合わせるために、ラッチされないか、または 1 回ラッチされます。
- (2) これらの信号は、1 個または 2 個のパイプライン・レジスタを通して送信できます。
- (3) 丸めおよび / または飽和機能は、Q1.15 入力用の 18×18 ビット符号付き乗算でのみサポートされています。

入力レジスタ

各乗算器オペランドは、入力レジスタまたは直接乗算器に供給できます。以下の DSP ブロック信号は、DSP ブロック内の各入力レジスタを制御します。

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

入力レジスタは乗算器に信号を供給し、shiftouta および shiftoutb の 2 本の専用シフト出力ラインをドライブします。図 10-5 に示すように、1 個の乗算器ブロックからの専用シフト出力は、同じ DSP ブロック内のその下に隣接する乗算器、または次の 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、シフト・レジスタ・チェーンを形成します。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。したがって、このシフト・レジスタ・チェーンは、Arria GX デバイス・ファミリの最大のデバイスでは 768 個のレジスタまで任意の長さで構成することができます。

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。9×9 および 18×18 乗算器を実装する場合、入力シフト・レジスタが 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 LE リソースが大幅に削減され、配線遅延も最小化できます。

Arria GX の DSP ブロックでは、sourcea および sourceb 信号により、乗算器オペランドを通常の入力とするかシフト・レジスタ入力とするかを動的に選択することができます。sourcea 信号のロジック 1 は、データ A が専用のスキャン・チェーンによって供給されることを示し、ロジック 0 は通常の入力によって供給されることを示します。この機能により、シフト・レジスタがスキャン・チェーンを使用して正常に動作する動的にロード可能なシフト・レジスタの実装が可能となり、データ入力値を使用してパラレルに動的にロードすることもできます。図 10-5 に、シフト・レジスタ・チェーンを示します。

図 10-5. シフト・レジスタ・チェーン 注 (1)

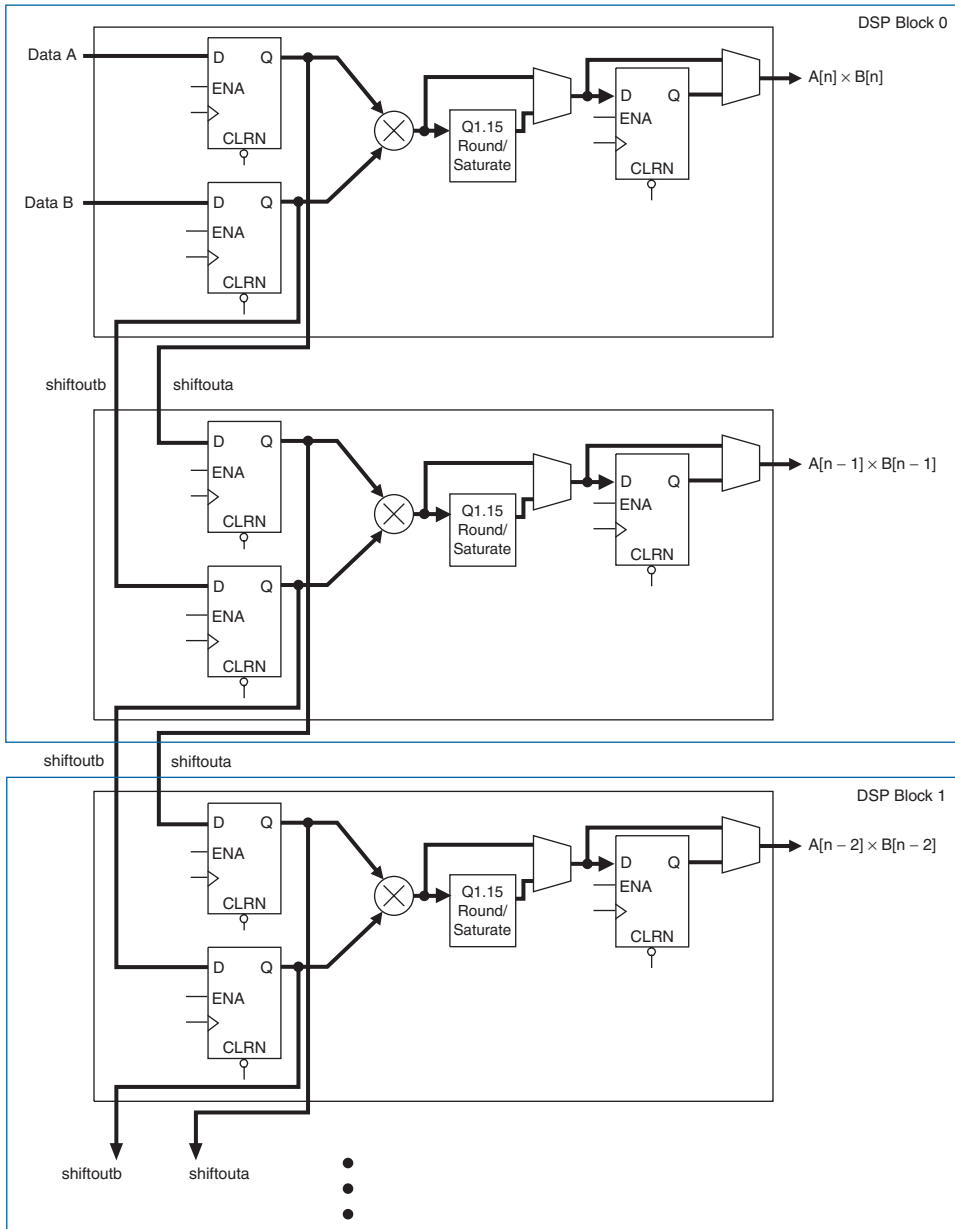


図 10-5 の注:

(1) Data A または Data B 入力のいずれかを定数係数乗算用の平行入力に設定できます。

表 10-3 に、DSP ブロックの入力レジスタ・モードの概要を示します。

レジスタ入力モード	9 × 9	18 × 18	36 × 36
パラレル入力	√	√	√
シフト・レジスタ入力	√	√	-


乗算器ステージ

乗算器ステージは、9 × 9、18 × 18、または 36 × 36 乗算器、そしてこれらのコンフィギュレーション間において他のより小さな乗算器をサポートします。詳細については、10-19 ページの「動作モード」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パラレルに実行できます。

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。signa および signb の 2 つの信号は、それぞれ各オペランドの表現を制御します。signa 信号のロジック 1 はデータ A が符号付き数値であることを示し、ロジック 0 は符号なし数値であることを示します。表 10-4 は、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ています。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 (signa 値)	データ B (signb 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


各 DSP ブロックには、1 つの signa と 1 つの signb 信号しかありません。したがって、同じ DSP ブロックに供給されるデータ 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 ブロックに供給されるすべてのデータ B 入力も同じ符号表現でなければなりません。乗算器は、符号表現に関係なく完全精度を提供します。

 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。

飽和および丸め機能

DSP ブロックには、Q1.15 入力フォーマット用の各 18×18 乗算器の後に、オプションの飽和および丸め機能の実行をサポートするハードウェアがあります。

 Q1.15 入力フォーマットは 16 ビット入力幅を必要とするため、デザインでは飽和および丸め機能オプションに 18×18 乗算器を使用する必要があります。

 Q1.15 入力フォーマットの乗算には、符号付き乗算器が必要です。Q1.15 入力フォーマットの最上位ビット (MSB) は、値の符号ビットを表します。乗算中に符号拡張が正しく処理されるようにするには、符号付き乗算器を使用します。

Q1.15 フォーマットは、16 ビットを使用して各固定小数点入力を表現します。MSB は符号ビットで、残りの 15 ビットは小数点 (または小数值) の後の値を表現するために使用されます。この Q1.15 の値は、以下の計算式に示すように、 2^{15} で除算される 16 ビットの整数値表現に相当します。

$$-\frac{1}{2} = 1\ 100\ 0000\ 0000\ 0000 = -\frac{0x4000}{2^{15}}$$

$$\frac{1}{8} = 0\ 001\ 0000\ 0000\ 0000 = \frac{0x1000}{2^{15}}$$

Q1.15 の数値はすべて -1 と 1 の間にあります。

乗算を実行する際、Q1.15 入力は 18 の乗算器入力のうち 16 しか使用しないにもかかわらず、乗算器には 18 ビット入力バス全体が送信されます。これは 2 つの最下位ビット (LSB) が常に 0 になる 1.17 入力に似ています。

乗算器の出力は、丸めまたは飽和機能の実行前は 2.34 の値 (合計 36 ビット) になります。2 つの MSB は符号ビットです。出力には 1 つの符号ビットしか必要ないため、2 つの MSB のうち 1 つを無視でき、結果として丸め機能または飽和機能の実行前には Q1.34 の値となります。

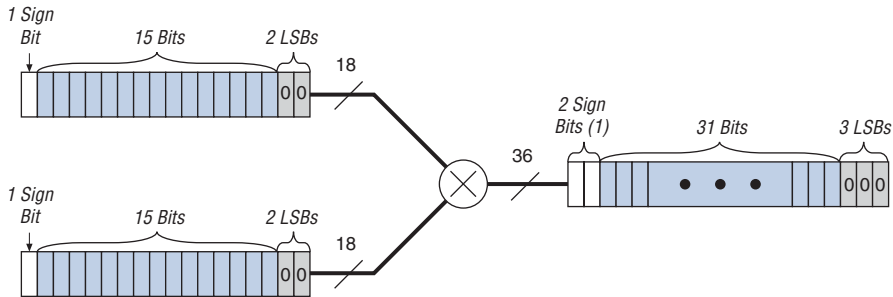
デザインで飽和機能を実行すると、乗算器出力は 1.31 フォーマットで 0x7FFFFFFF に飽和します。これは全 36 ビット乗算器出力のうちビット [34..3] を使用します。3 つの LSB は 0 に設定されます。

DSP ブロックは、乗算器またはアキュムレータの出力の LSB から `mult_is_saturated` または `accum_is_saturated` オーバフロー信号値を取得します。したがって、飽和が発生するたびに、乗算器またはアキュムレータの出力の LSB から `mult_is_saturated` または `accum_is_saturated` オーバフロー信号に 1 を送ります。それ以外の場合、飽和機能がイネーブルされるかまたは乗算器またはアキュムレータの出力の LSB の値を反映するときには、このオーバフロー信号は 0 になります。

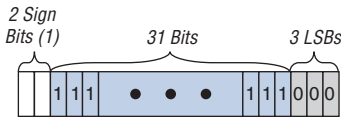
デザインで丸め機能を実行するとき、1.31 フォーマットの `0x00008000` を乗算器の出力に加算し、全 36 ビット乗算器出力のビット [34..15] のみを使用します。1.31 フォーマットの `0x00008000` を 36 ビット乗算器の結果に加算することは、2.34 フォーマットの `0x0 0004 0000` を加算するのと同様です。16 の LSB は 0 に設定されます。図 10-6 に、デザインが乗算で丸めおよび飽和機能を実行するときに使用されるビットを示します。

図 10-6. 丸めおよび飽和ビット

18 × 18 Multiplication



Saturated Output Result



Rounded Output Result

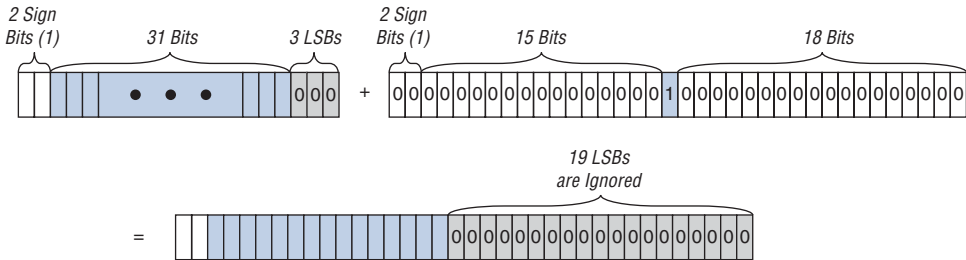


図 10-6 の注:

(1) 両方の符号ビットは同一です。デザインは1つの符号ビットのみ使用し、もう1つは無視されます。

デザインが multiply_accumulate または multiply_add 演算を実行する場合、乗算器の出力は、2,31 フォーマットの値として加算器 / 減算器 / アキュムレータ・ブロックに入力され、3つの LSB は 0 になります。

パイプライン・レジスタ

乗算器からの出力はパイプライン・レジスタに供給できますが、このレジスタはバイパスできます。パイプライン・レジスタはどの乗算器サイズに対しても実装でき、DSP ブロックの最大性能を向上させることができます。特に後続の DSP ブロックの加算器ステージを使用するときには有効です。パイプライン・レジスタは加算器 / 減算器 / アキュムレータ・ブロックと加算器 / 出力ブロック間の長い信号パスを分割して、2 つの短いパスを作成します。

加算器 / 出力ブロック


加算器 / 出力ブロックは、以下の要素で構成されています。

- 加算器 / 減算器 / アキュムレータ・ブロック
- サメーション・ブロック
- 出力選択マルチプレクサ
- 出力レジスタ

加算器 / 出力ブロックは、以下のようにコンフィギュレーションできます。

- 出力インタフェース
- オプションでロード可能なアキュムレータ
- 1 レベル加算器
- 第1レベル加算器にダイナミック加算/減算コントロールを備えた2 レベル加算器
- 36 ビット乗算器、9×9 複素数乗算器、または 18×18 複素数乗算器の最終ステージ。

出力選択マルチプレクサは DSP ブロックの出力コンフィギュレーションを設定します。出力レジスタは加算器 / 出力ブロックの出力をラッチするのに使用できます。

 加算器 / 出力ブロックを乗算器と別々に使用することはできません。


 図 10-7 に、加算器 / 出力ブロック・アーキテクチャを示します。

図 10-7. 加算器 / 出力ブロック・アーキテクチャ

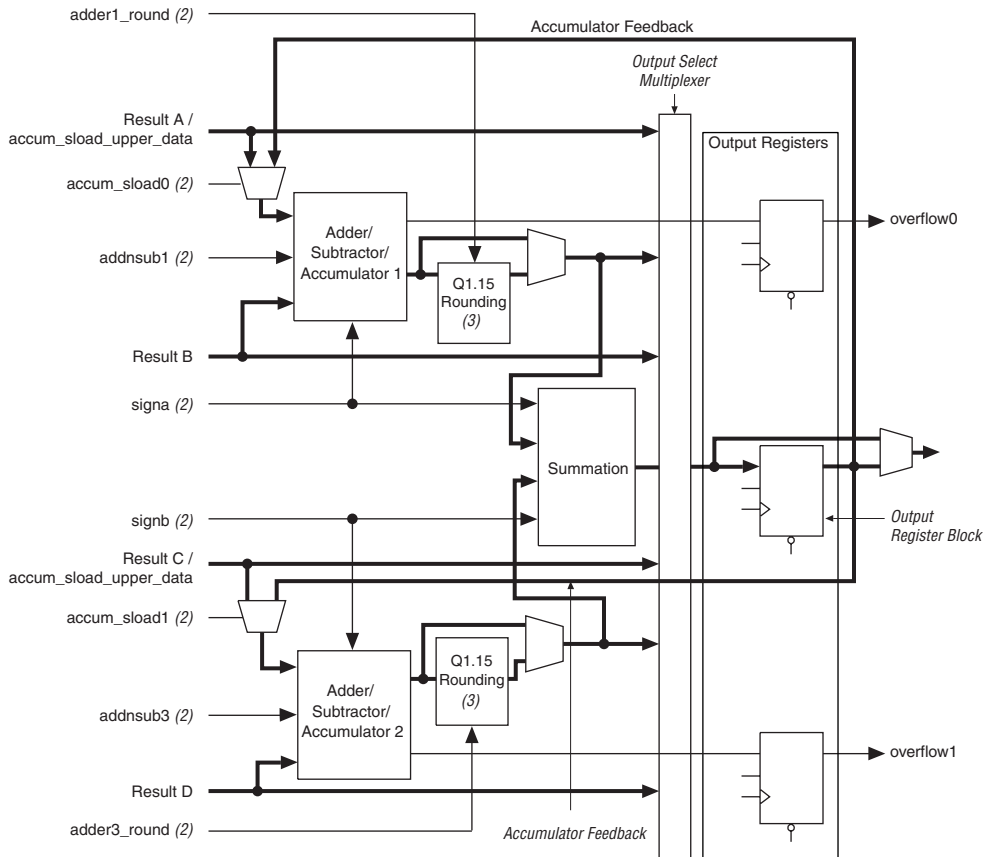


図 10-7 の注:

- (1) 加算器 / 出力ブロックは、 18×18 モードです。 9×9 モードでは、4つの加算器 / 減算器ブロックと2つのサマーション・ブロックがあります。
- (2) これらの信号はパイプライン・レジスタを通して送信できます。パイプラインの長さは1または2に設定できます。
- (3) Q1.15 入力は、 9×9 または 36×36 モードでは使用できません。

加算器 / 減算器 / アキュムレータ・ブロック

加算器 / 減算器 / アキュムレータ・ブロックは、加算器 / 出力ブロックの第1レベルの加算器ステージです。このブロックはアキュムレータまたは加算器 / 減算器としてコンフィギュレーション可能です。

アキュムレータ

図 10-7 のように、加算器 / 減算器 / アキュムレータがアキュムレータとしてコンフィギュレーションされると、加算器 / 出力ブロックの出力はアキュムレータにフィードバックされます。アキュムレータは、加算のみまたは減算のみ実行するようにセットアップできます。あるいは `addnsub` 信号を使用して累算方向をダイナミックに制御することも可能です。`addnsub` 信号のロジック 1 はアキュムレータが加算を実行していることを示し、ロジック 0 は減算を示します。

各アキュムレータは DSP ブロック出力レジスタをクリアするか、`accum_sload` 信号を使用してクリアできます。`accum_sload` 信号を使用したアキュムレータのクリアは、出力レジスタのリセットとは関係がないため、クロック・サイクルを失うことなく累算をクリアして、新しい累算を開始できます。`accum_sload` 信号は、フィードバック・マルチプレクサをコントロールして、乗算器の出力をアキュムレータ・フィードバック・パスではなく 0 と合計するように指定します。

アキュムレータは、`accum_sload` 信号および 1 クロック・サイクル・レイテンシを伴う `accum_sload_upper_data` バスを使用して、0 以外の値で初期化 / プリロードすることも可能です。アキュムレータのプリロードは、乗算器の結果に `accum_sload_upper_data` バス上の指定された値を加算することによって行われます。アキュムレータをクリアする場合と同様に、`accum_sload` 信号はフィードバック・マルチプレクサに対して、アキュムレータ・フィードバック信号の代わりに、`accum_sload_upper_data` 信号でアキュムレータにデータを供給するよう指定します。`accum_sload_upper_data` 信号は、アキュムレータの上位 36 ビットのみロードします。アキュムレータ全体にロードするには、アキュムレータにデータを供給している乗算器を 1 の乗算を実行するように設定し、その乗算器を通して下位 16 ビットの値を送信する必要があります。

アキュムレータがオーバフローまたはアンダーフローを検出すると、クロックのポジティブ・エッジでオーバフロー信号が **High** になります。オーバフロー信号は、オーバフローまたはアンダーフローが検出された後、オーバフローまたはアンダーフロー状態が持続している場合でも、1 クロック・サイクルの間のみ **High** になります。オーバフロー信号を無期限に、またはラッチがクリアされるまで保持するには、DSP ブロック外部のラッチを使用する必要があります。

DSP ブロックは各アキュムレータで Q1.15 入力フォーマットの飽和および丸め機能をサポートします。飽和機能または丸め機能、あるいはその両方がアキュムレータの出力に対して実行される場合は、以下の信号が使用できます。

- `accum_round`
- `accum_saturation`
- `accum_is_saturated` 出力

各DSPブロックには、2組のaccum_round信号とaccum_saturation信号があり、それぞれ（各アキュムレータに対して1組の信号）アキュムレータ出力で丸めまたは飽和機能を実行するかどうかを制御します。アキュムレータ出力の丸めおよび飽和機能は、Q1.15 入力フォーマットの計算に必要なビット幅に適合する 16×16 乗算アキュムレータを実装する場合にのみ使用可能です。accum_round 信号および accum_saturation信号のロジック1は、丸めおよび飽和機能が実行されたことを示し、ロジック0は丸めおよび飽和機能が実行されていないことを示します。accum_is_saturated 出力信号のロジック 1 は、アキュムレータの結果に飽和が発生したことを知らせます。

図 10-10 に、乗算アキュムレータ動作を実行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。

加算器 / 減算器

addnsub1 または addnsub3 信号は、加算または減算のいずれかを実行しているかを指定します。addnsub1 信号または addnsub3 信号のロジック 1 は、加算器 / 減算器が加算を実行していることを示し、ロジック 0 は減算を示します。これらの信号は、DSP ブロックの外部ロジックを使用してダイナミックに制御できます。最初のステージが減算器としてコンフィギュレーションされた場合、出力はA-BおよびC-Dとなります。

加算器 / 減算器ブロックは、乗算器ブロックと同じ signa 信号と signb 信号を共有します。signa および signb 信号は、1 または 2 クロック・サイクルのレイテンシでパイプライン化すること、またはパイプライン化しないことが可能です。

DSP ブロックは、各加算器 / 減算器の後で Q1.15 入力フォーマットの丸め機能（飽和機能ではなく）をサポートします。addnsub1_round 信号と addnsub3_round 信号は、加算器 / 減算器の出力に丸め機能が実行されているかどうかを判断します。

addnsub1_round信号はトップ加算器/減算器の丸め機能をコントロールし、addnsub3_round 信号はボトム加算器 / 減算器の丸め機能をコントロールします。加算器出力の丸め機能は、Q1.15 入力フォーマットの演算に必要なビット幅に適合する 16×16 乗算器・加算器を実装する場合にのみ使用可能です。addnsub_round 信号のロジック 1 は、丸め機能が実行されたことを示し、ロジック 0 は丸め機能が実行されていないことを示します。

サメーション・ブロック

加算器 / 減算器ブロックの出力は、加算器 / 減算器ブロックの両方の出力を合計する加算器ブロックであるサメーション・ブロックに供給されます。サメーション・ブロックは3つ以上の乗算器の結果を合計するときに使用されます。これは FIR フィルタリングなどのアプリケーションに役立ちます。

出力選択マルチプレクサ


加算器 / 出力ブロックの各種エレメントの出力は、出力選択マルチプレクサによって送信されます。出力マルチプレクサは、DSP ブロックの動作モードに基づいて、乗算器ブロックの出力、加算器 / 減算器 / アキュムレータの出力、またはサメーション・ブロックの出力のいずれかを DSP ブロックの出力に送るかを選択します。出力選択マルチプレクサのコンフィギュレーションは、指定した 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ます。

出力レジスタ

出力レジスタを使用して DSP ブロック出力をラッチすることができます。以下の信号は、DSP ブロック内の各出力レジスタをコントロールできます。

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

出力レジスタはどの DSP ブロック動作モードでも使用できます。

 出力レジスタは、Multiply-Accumulate モードではアキュムレータの一部を形成します。



Arria GX の DSP ブロックの配線およびインタフェースについて詳しくは、「Arria GX デバイス・ハンドブック Volume 1」の「Arria GX デバイス・アーキテクチャ」の章を参照してください。

動作モード

DSP ブロックは、アプリケーションのニーズに応じて、4つの基本動作モードの1つ、または2つのモードの組み合わせで使用できます。表 10-5 に、4つの基本動作モードとモードに応じて1つの DSP ブロック内に実装可能な乗算器数を示します。

モード	乗算器数		
	9 × 9	18 × 18	36 × 36
Simple Multiplier	8 個の乗算器と 8 つの積の出力	4 個の乗算器と 4 つの積の出力	1 つの乗算器
Multiply-Accumulate	-	2 つの 52 ビット乗算累積ブロック	-
Two-Multiplier Adder	4 個の 2 乗算器・加算器 (2 つの 9 × 9 複素数乗算)	2 個の 2 乗算器・加算器 (1 つの 18 × 18 複素数乗算)	-
Four-Multiplier Adder	2 個の 4 乗算器・加算器	1 個の 4 乗算器・加算器	-

Quartus II ソフトウェアには、乗算器の動作モードを制御するのに使用され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。

Arria GX DSP ブロックは、異なるモードで同時に動作できます。例えば、1つの DSP ブロックを分割して、両方の乗算器の入力 a および入力 b が同じ符号表現を持つ 9 × 9 乗算器や 18 × 18 乗算器・加算器を動作させることができます。これにより、DSP ブロックのリソース効率が向上し、Arria GX デバイス内により多くの乗算器を実装できます。Quartus II ソフトウェアは、同じブロック内で同じ DSP ブロック・リソースを共有できる乗算器を自動的に配置します。

さらに、Arria GX の各 DSP ブロックを、以下の 3 つのモードでダイナミックに切り換わるようにセットアップできます。

- 最大 4 個の独立した 18 ビット乗算器
- 最大 2 個の 18 ビット乗算アキュムレータ
- 1 個の 36 ビット乗算器

Arria GX の DSP ブロックのハーフ・ブロックには、それぞれ独立したモード・コントロール信号があり、同じ DSP ブロック内に複数の 18 ビット乗算器、または乗算アキュムレータを実装して、(DSP ブロックのハーフ・ブロックに別々に存在する場合) それぞれダイナミックに切り換えることができます。デザインに 36 ビット乗算器が必要な場合、乗算器には DSP ブロック全体が必要なため、それに対応して DSP ブロック全体を切り換える必要があります。ダイナミック・モード切り換えをサポートする最小入力ビット幅は 18 ビットです。

Simple Multiplier モード

Simple Multiplier モードでは、DSP ブロックは、汎用乗算器や多数の個別乗算演算を必要とするアプリケーションに対して、個別の乗算演算を実行します。

9 ビットおよび 18 ビット乗算器

各 DSP ブロック乗算器は、9 ビットまたは 18 ビット乗算器用にコンフィギュレーションされます。1 つの DSP ブロックは最大 8 個の個別の 9×9 乗算器、または最大 4 個の個別の 18×18 乗算器をサポートできます。最大 9 ビットのオペランド幅には 9×9 乗算器が実装され、10 ~ 18 ビットのオペランド幅には 18×18 乗算器が実装されます。図 10-8 に、Simple Multiplier 動作モードの DSP ブロック図を示します。

図 10-8. Simple Multiplier モード

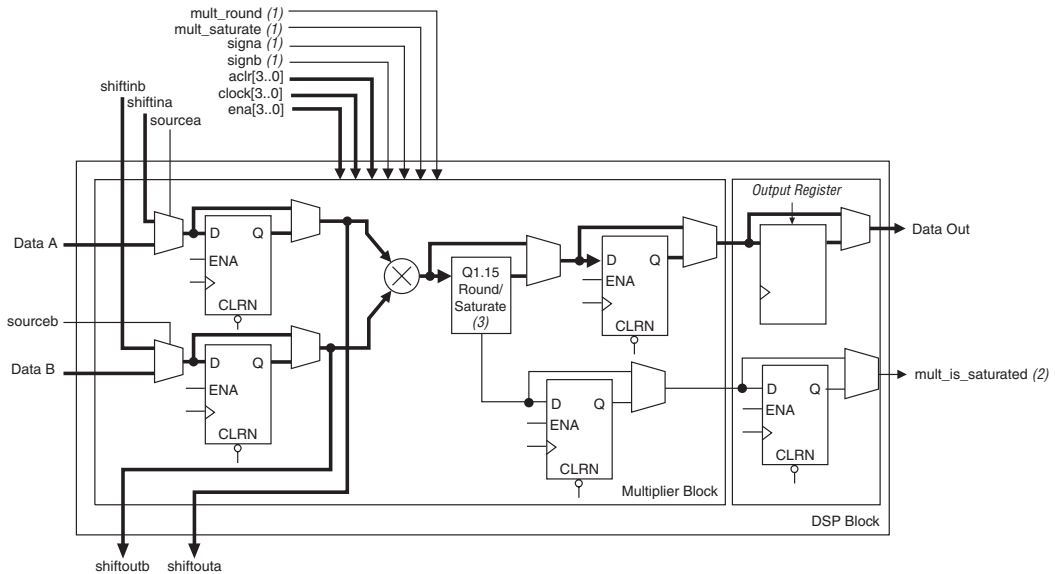


図 10-8 の注:

- (1) これらの信号は、データ・パス・パイプラインに合わせるために、ラッチされないか、または 1 回ラッチされます。
- (2) この信号はデータ・パスと同じレイテンシを持っています。
- (3) 丸めおよび飽和機能は、Q1.15 入力用の 18×18 ビットの符号付き乗算でのみサポートされています。

乗算器オペランドは、符号付き整数、符号なし整数、またはそれらの組み合わせを受け入れ可能です。signa 信号および signb 信号はダイナミックに変更でき、DP ブロックにラッチできます。さらに、乗算器の入力と結果は個別にラッチ可能です。DSP ブロック内のパイプライン・レジスタは乗算器の結果をパイプライン化するのに使用でき、DSP ブロックの性能を向上させます。

36 ビット乗算器

36 ビット乗算器も単純な乗算器モードですが、36×36 ビット乗算演算を実装する加算器 / 出力ブロックを含む DSP ブロック全体を使用します。このデバイスは 36 ビット入力のうち 18 ビット・セクションを 4 個の 18 ビット乗算器に入力します。加算器 / 出力ブロックは、サマーション・ブロックを使用して乗算器から取得した部分積を追加します。パイプライン・レジスタは乗算器ステージとサマーション・ブロックの間で使用でき、乗算時間を短縮します。36×36 ビット乗算器は、符号付き、符号なし、および符号が混合した乗算をサポートしています。図 10-9 に、36 ビット乗算器を実装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 10-9. 36 ビット乗算器

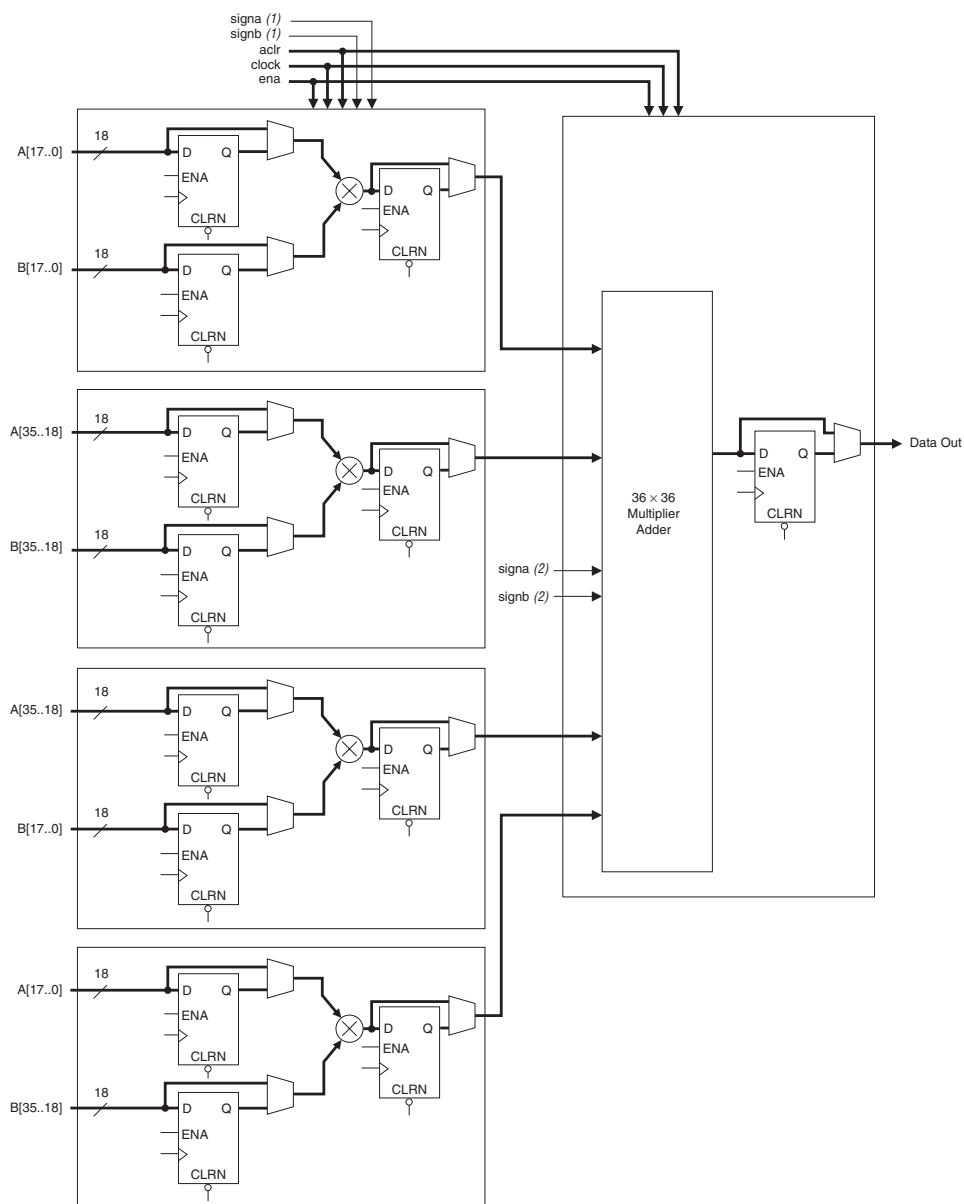


図 10-9 の注:

- (1) これらの信号は、パイプラインに合わせるために、ラッチされないか、または1回ラッチされます。
- (2) これらの信号は、データ・パス・パイプラインに合わせるために、ラッチされないか、1回または2回ラッチされます。

36 ビット乗算器は、高精度浮動小数点の仮数の乗算を行う演算アプリケーションなど、18 ビットを超える精度を必要とするアプリケーションに有効です。

Multiply-Accumulate モード

Multiply-Accumulate モードでは、乗算器ステージの出力は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た加算器 / 出力ブロックに供給されます。図 10-10 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 10-10. Multiply-Accumulate モード

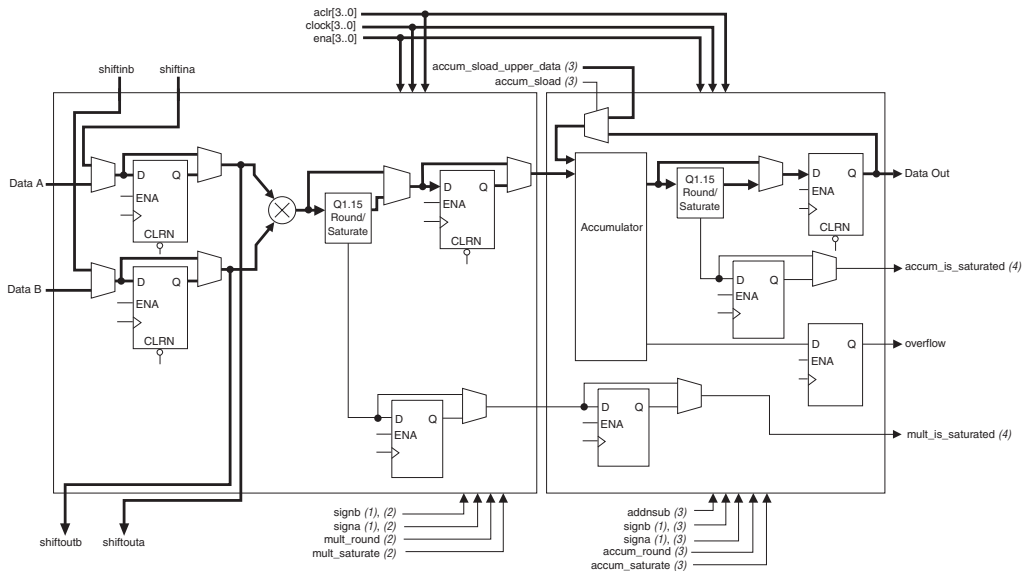



図 10-10 の注:

- (1) signa 信号および signb 信号は、乗算器ステージと加算器 / 出力ブロックでは同じです。
- (2) これらの信号は、データ・パス・パイプラインに合わせるために、ラッチされないか、または 1 回ラッチされます。
- (3) これらの信号は、1 個または 2 個のパイプライン・レジスタを通して送信できます。
- (4) これらの信号はデータ・パスのレイテンシと一致します。

1 つの DSP ブロックは、最大 2 個の独立した 18 ビット乗算アキュムレータを実装できます。18 ビットより少ないビット幅の乗算アキュムレータは、18 ビット乗算器の未使用の下位ビットをグラウンドに接続することにより実装されます。


乗算アキュムレータの出力は、36 ビット乗算器の結果の累積加算に対応するために、52 ビット幅となっています。このモードでは、DSP ブロックは出力レジスタと accum_sload 信号およびオーバーフロー信号を使用します。accum_sload 信号は、クロック・サイクルを失わずに新しい累算演算を開始できるようアキュムレータをクリアするために使用されます。この信号は、ラッチなし、あるいは1回または2回ラッチすることができます。accum_sload 信号は、1 クロック・サイクルのペナルティで、accum_sload_upper_data 信号に指定された値をアキュムレータにプリロードするのにも使用できます。accum_sload_upper_data 信号は、アキュムレータの上位 36 ビット（アキュムレータの [51..16] ビット）のみロードします。アキュムレータ全体にロードするには、アキュムレータにデータを供給している乗算器に 1 の乗算を実行するよう設定し、その乗算結果を使って下位 16 ビット（ビット [15..0]）の値を送信する必要があります。ビット [17..16] は、accum_sload_upper_data 信号と乗算器出力の両方でオーバーラップされます。これらの信号のいずれかにより、ビット [17..16] をロードするのに使用できます。

オーバーフロー信号は、アキュムレータのオーバーフローまたはアンダーフローを示します。この信号は、サイクルごとに新しい累算操作が実行されるため、クロック・サイクルごとにアップデートされます。この信号を保存するには、外部ラッチを使用できます。addnsub 信号は、累算または減算がダイナミックに実行されるかどうかを指定するのに使用できます。

 乗算器ステージで 1 の乗算を指定することにより、アキュムレータのみ(乗算なし)を DSP ブロック内に実装することができます。

乗算・加算モード

乗算・加算モードでは、乗算器ステージの出力は、2 個以上の乗算器の出力を合計または減算する加算器または減算器としてコンフィギュレーションされた加算器 / 出力ブロックに供給されます。DSP ブロックは、2 つの乗算・加算（2 個の乗算器の出力を加算 / 減算）または 4 つの乗算・加算ファンクション（4 個の乗算器の出力を加算 / 減算）を実装するようにコンフィギュレーションできます。

 DSP ブロック内の加算器ブロックは、乗算演算に続く場合にのみ使用できます。

Two-Multiplier Adder

2 乗算器・加算器コンフィギュレーションでは、DSP ブロックは 4 個の 9 ビットまたはそれより以下の乗算器・加算器、または 2 個の 18 ビット乗算器・加算器を実装します。加算器は両方の乗算器出力の和、または両方の乗算器出力の差をとるようにコンフィギュレーションできます。加算 / 減算演算をダイナミックに変更するオプションがあります。これらの乗算・加算ファンクションは FFT や複素数 FIR フィルタなどのアプリケーションに役立ちます。図 10-11 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。

図 10-11. Two-Multiplier Adder モード

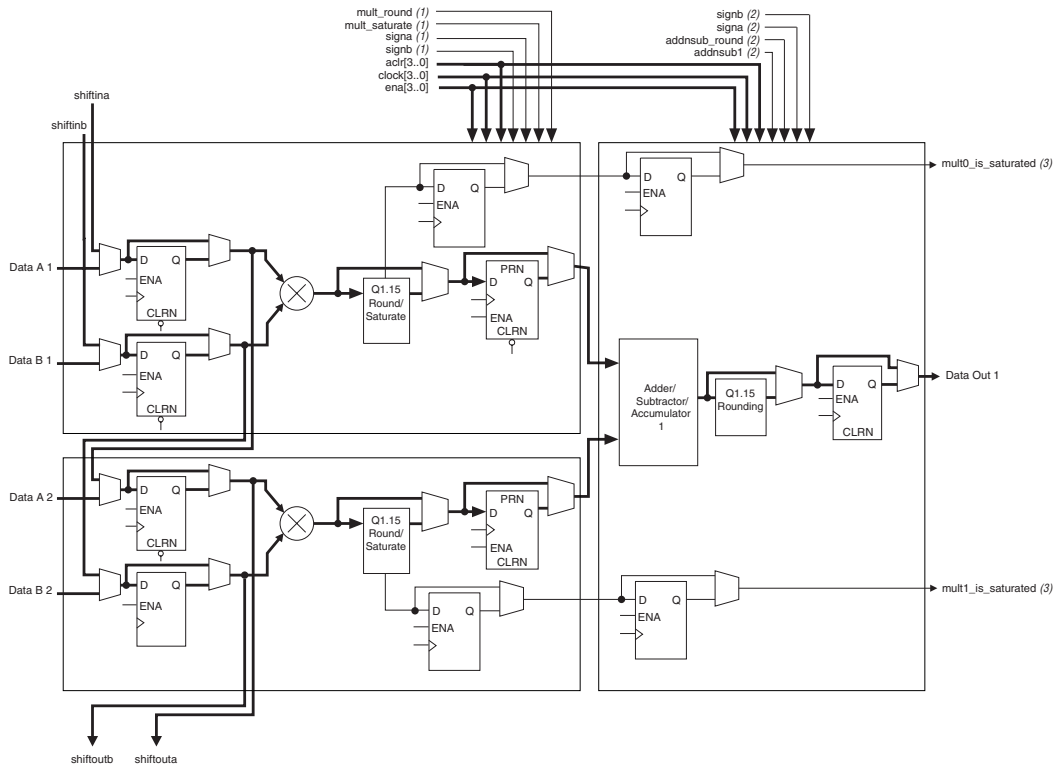


図 10-11 の注:

- (1) これらの信号は、データ・パス・パイプラインに合わせるために、ラッチされないか、または 1 回ラッチされます。
- (2) これらの信号はパイプライン・レジスタを通して送信できます。パイプラインの長さは 1 または 2 に設定できます。
- (3) これらの信号はデータ・パスのレイテンシと一致します。

複素数乗算器

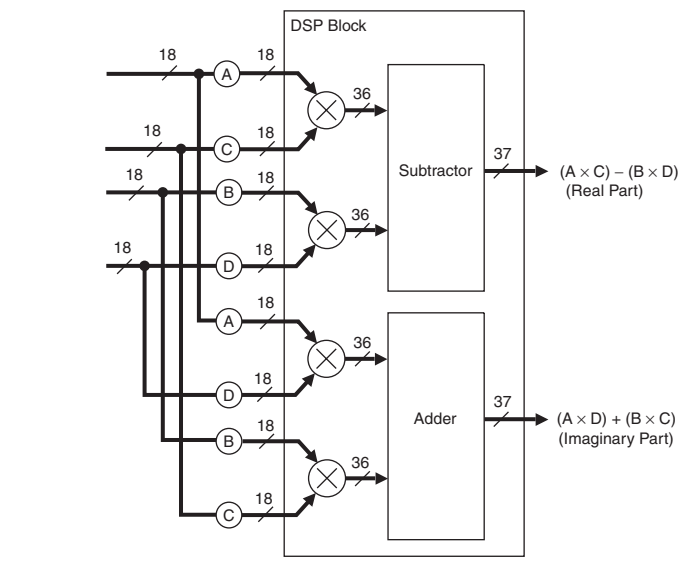
DSP ブロックは Two-Multiplier Adder モードを使用して、複素数乗算器を実装するようにコンフィギュレーションできます。シングル DSP ブロックは、1 個の 18×18 ビットの複素数乗算器、または 2 個の 9×9 ビットの複素数乗算器を実装できます。

複素数乗算は、以下に示すように記述できます。

$$(a + jb)(c + jd) = ((ac) - (bd)) + j((ad) + (bc))$$

この複素数乗算を DSP ブロック内に実装するには、18 ビットまでのデータに対して、1 個の減算器ブロックに供給される 2 個の乗算器を使用して実数部 $((ac) - (bd))$ が実装され、加算ブロックに供給される他の 2 個の乗算器を使用して仮数部 $((ad) + (bc))$ が実装されます。図 10-12 に、18 ビット複素数乗算を示します。9 ビットまでのデータ幅に対しては、DSP ブロックは 4 つの加算器 / 減算器 / アキュムレータ・ブロックに供給される 8 個の 9 ビット乗算器を使用して、2 つの別々の複素数乗算演算を実行できます。DSP ブロックの外部リソースを使用して、正しい実数部と虚数部の入力成分を適切な乗算器入力に送って、正しい複素数乗算演算を実行する必要があります。

図 10-12. Two-Multiplier Adder モードを使用した複素数乗算器



Four-Multiplier Adder

4 乗算器・加算器コンフィギュレーションでは、DSP ブロックは 1 組の 18×18 乗算器・加算器か、2 組の独立した 9×9 乗算器・加算器を実装できます。これらのモードは、1 次元または 2 次元フィルタリング・アプリケーションの実装に役立ちます。4 乗算器・加算器は、2 つの加算ステージで実行されます。4 個の乗算器の 2 つの出力は、2 つの第 1 ステージの加算器 / 減算器 / アキュムレータ・ブロックにおいて内部で合計されます。これら 2 つの加算器 / 減算器 / アキュムレータ・ブロックは、最終ステージのサマーション・ブロックで合計され、最終的な 4 乗算器・加算器の結果を生成します。図 10-13 に、Four-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。

図 10-13. Four-Multiplier Adder モード

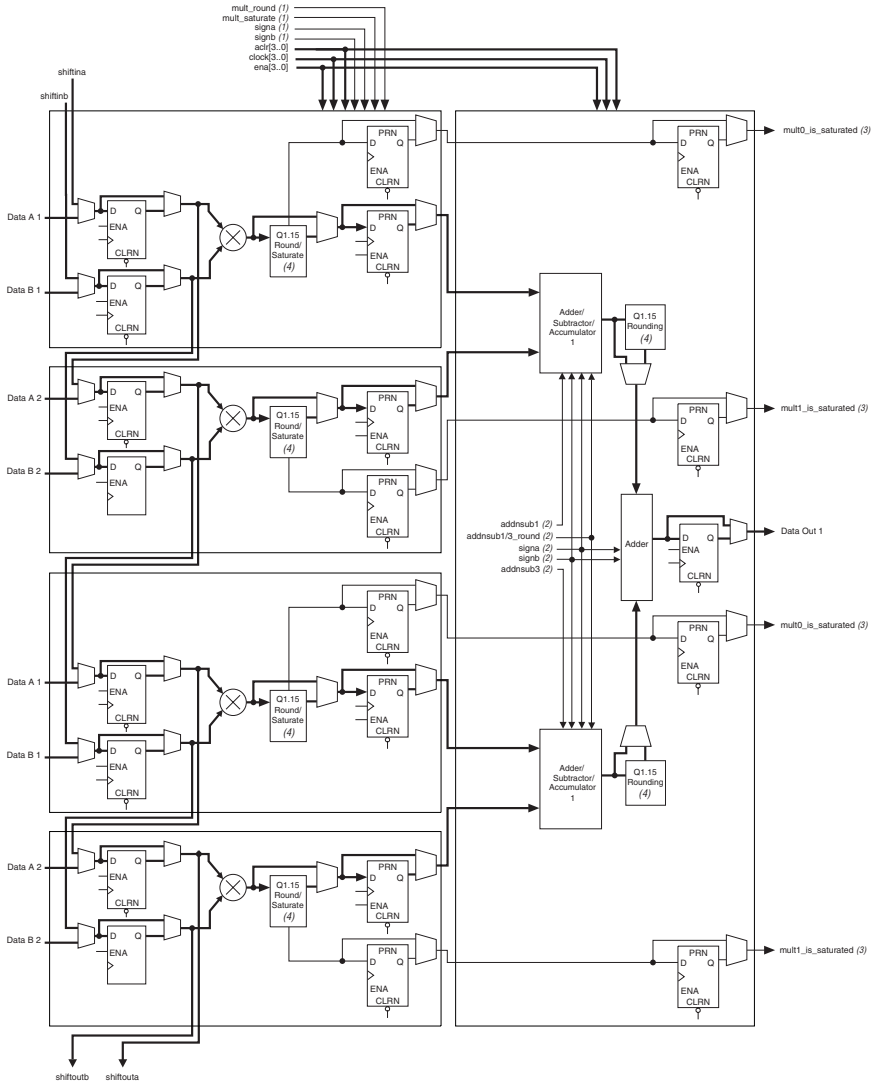


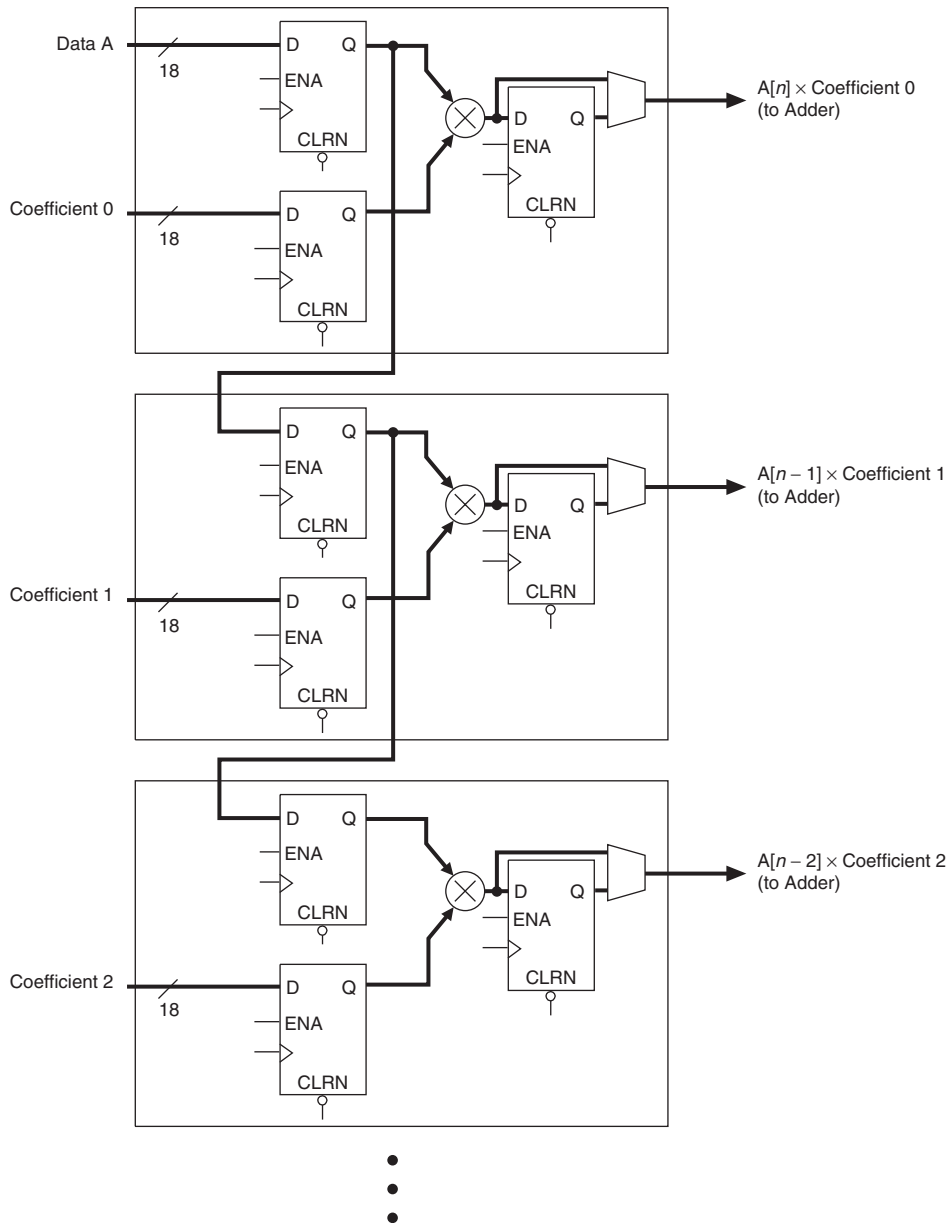
図 10-13 の注:

- (1) これらの信号は、データ・パス・パイプラインに合わせるために、ラッチされないか、または 1 回ラッチされます。
- (2) これらの信号は、データ・パスのレイテンシに合わせるため、パイプライン・レジスタを通して送信する必要があります。
- (3) これらの信号はデータ・パスのレイテンシと一致します。
- (4) 丸めおよび飽和機能は、Q1.15 入力用の 18×18 ビットの符号付き乗算でのみサポートされています。


FIR フィルタ

Four-Multiplier Adder モードは、FIR フィルタおよび複素数 FIR フィルタのアプリケーションの実装に使用できます。これを行うために、DSP ブロックは、1 組の入力レジスタを専用シフト・レジスタ・チェーンを使用してシフト・レジスタとしてコンフィギュレーションした Four-Multiplier Adder モードでセットアップされます。シフト・レジスタとしてコンフィギュレーションされた入力レジスタ・セットには入力データが含まれ、通常入力としてコンフィギュレーションされた入力にはフィルタ係数が保持されます。図 10-14 に、入力レジスタを使用して Four-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。

図 10-14. 入力シフト・レジスタにより Four-Multiplier Adder モードを使用して実装された FIR フィルタ



DSP ブロック内のビルトイン入力シフト・レジスタ・チェーンによって、ロジック・エレメント (LE) 内の DSP ブロック外部のシフト・レジスタは不要になります。このアーキテクチャ機能では、フィルタ回路はすべて DSP ブロック内でローカライズされるため、フィルタ・デザインが簡略化されフィルタ性能が向上します。

 36 ビットの Simple Multiplier モード用の入力シフト・レジスタは、DSP ブロックの外部レジスタを使用して実装する必要があります。

1つの DSP ブロックは、4 タップの 18 ビット FIR フィルタを実装できます。4 タップを超えるフィルタを実装する場合、LE を使用して実装された追加の加算器ステージによって、DSP ブロックをカスケード接続できます。

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ユーザ・デザインに実装するための方法として、インスタンス化と推論を提供しています。どちらの方法も以下の 3 つの Quartus II メガファンクションを使用します。

- lpm_mult
- altmult_add
- altmult_accum

Quartus II ソフトウェアで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、LeonardoSpectrum™ や Synplify などのサードパーティ製の合成ツール、または乗算器、乗算器・加算器、および乗算アキュムレータ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定する Quartus II Native Synthesis を使用して、それを合成することができます。いずれの方法を使用する場合も、Quartus II ソフトウェアはコンパイル時に機能を DSP ブロックにマップします。



メガファンクションおよび MegaWizard Plug-In Manager の使用方法については、Quartus II オンライン・ヘルプを参照してください。



詳しくは、「Quartus II 開発ソフトウェア・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

Arria GX デバイスの DSP ブロックは、FIR フィルタ、FFT ファンクション、および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・チェイン、加算器 / 減算器 / アキュムレータ・ブロック、およびサマーション・ブロックによって、これらのファンクションの実装に必要な外部ロジック量が削減され、リソース利用が効率的になり、DSP アプリケーションの性能とデータ・スループットが向上します。Quartus II ソフトウェアは、LeonardoSpectrum™ や Synplify ソフトウェアと併用することにより、これらの乗算器ファンクションをDSPブロックに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 10-6 に、本資料の改訂履歴を示します。

表 10-6. 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 v1.0	初版	—