

最適化された低コスト FPGA アーキテクチャ内の高 性能 DSP の能力

ラティスセミコンダクター ホワイト・ペーパー

2004年6月

ラティスセミコンダクター株式会社

<http://www.latticesemi.co.jp>

はじめに

デジタル信号処理 (DSP) のアプリケーションは、広がり続けています。それは、ビデオと静止画の増加する用途や、ソフトウェア無線機 (SDR) などの再構成可能なシステムの需要などによってドライブされています。これらのアプリケーションの多くは、多大な DSP 機能の必要性和コストに対する敏感さを結びつけています。したがって、それは高性能かつ低価格の DSP ソリューションという需要を生み出します。

汎用の DSP チップと FPGA は DSP 機能を実装する 2 つの一般的な方法です。両アプローチにはそれぞれ利点があるので、アプリケーションの要件によって、最適な実装方法は異なるでしょう。このホワイト・ペーパーは、一般的な DSP 機能の概要を述べ、つづいて汎用 DSP と FPGA との違いについて探ります。そして、LatticeECP™-DSP (Economy Plus Digital Signal Processing) アーキテクチャの記述と、既存の FPGA ソリューションと LatticeECP-DSP との比較が続きます。

典型的な機能

有限インパルス応答 (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)、およびミキサーなどは多くのアプリケーションに良く見られます。これらの機能は、乗算要素に伴って加算、減算、および累積 (アキュムレート) の組み合わせをそれぞれ必要とします。このセクションは、これらの機能を実装するために用いられるアルゴリズムの概要を述べます。

有限インパルス応答 (FIR) フィルタ

FIR フィルタは連続する n データ要素を格納し、それぞれが 1 サイクルごとにシフトされます。これらの遅延要素は一般的にタップと呼ばれます。係数が各タップのデータに乘じられ、それらの結果を合算して出力を得ます。いくつかの実装ではすべての乗算を並列に実行します。より一般的な実装は N ステージに分割し、アキュムレータが 1 タップごとの結果を次々と加算していきます。この実装は、 N の計算ステージをとり、 n/N 個の乗算器を必要とするという、機能リソースと速度のトレードオフの例です。係数が固定であるか可変であるかによって、或いは係数の設計値によって、用いられる種々の最適設計手法がありますが、それについてはこのホワイト・ペーパーでは述べません。図 1 は典型的な FIR フィルタの実装を示します。

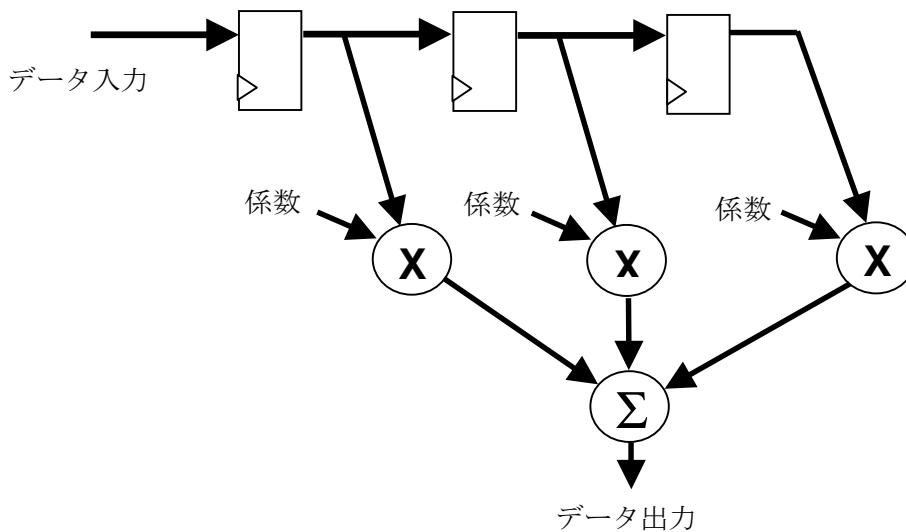


図1 典型的な FIR フィルタ

高速フーリエ変換(FFT)機能

FFT は、画像圧縮をはじめとしてサンプル・データのスペクトル成分を決定する種々アプリケーションに用いられ、実装する種々の方法があります。恐らく最も一般的な方法はクーリー・チュークの時間分割(DIT)アプローチを用いるもので、それはFFTを多くの小さなFFTに分解するものです。最も単純な実装は一般的に基底2 バタフライと呼ぶ要素(図2)を用い、入力データはその要素を複数回通過します。計算は図の左側に示されているように概念的には単純です。しかしながら、すべての乗算と加算は複素数で行われるので、図の右側に示されるように、実際に必要となる乗算と加算の回数は、少なからず挑戦的です。

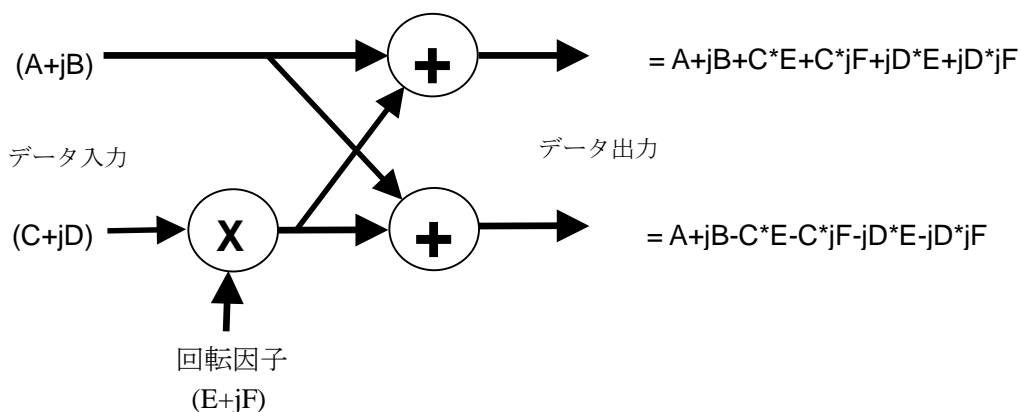


図2 FFTの実装に一般的に用いられる基底2 バタフライ

無限インパルス応答(IIR)フィルタ

フィードバック・パスがあることを除いて、IIR フィルタは FIR フィルタと似ていますが、このフィードバック・パスの存在が、IIR フィルタの設計と解析を FIR フィルタよりも複雑にします。しかしながら同じシリコンの規模を考慮した場合、IIR アプローチは FIR より効率の良いフィルタを提供することができます。いくつかの IIR アーキテクチャがありますが、一般的なアプローチの一つは図 3 に示す 2 次バイカッドを基本単位にして IIR フィルタを構成するものです。

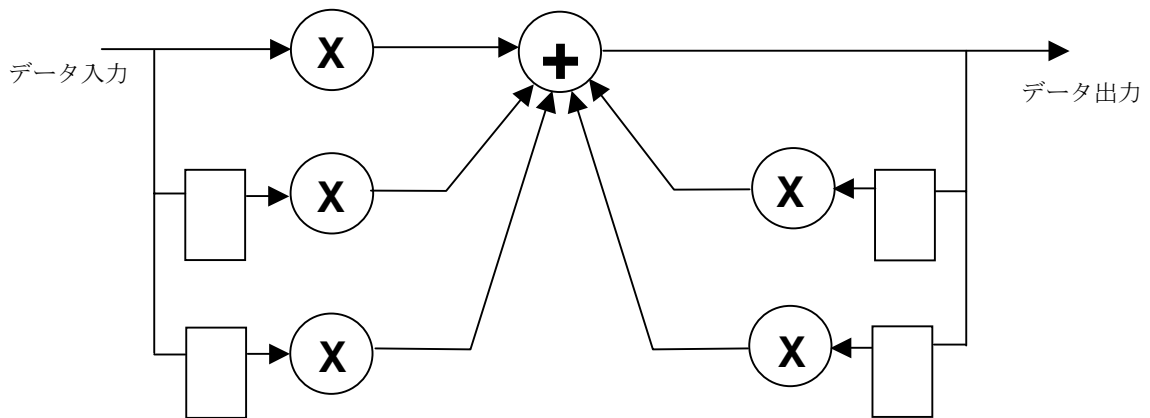


図 3 2 次バイカッド IIR

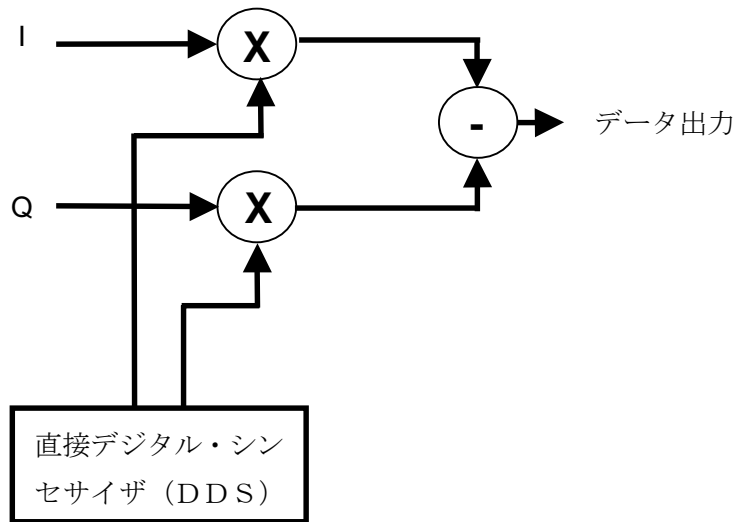


図 4 複素演算を用いる典型的なアップ・コンバータ・ミキサー

ミキサー機能

多くのアプリケーションで、信号の周波数をシフトさせるためにミキサーを用いま

す。概念的にはただ一度の乗算を行います。デジタル・アプリケーションでは複素形式で数を表す（通常 “I/Q 成分 “と呼ぶ）と多くの利点があることから、実際は複素乗算となります。図 4 はデジタル・アップ・コンバージョンで用いられるミキサーを示します。

汎用 DSP ソリューション 対 FPGA 実装

一般的な機能の記述で図示されるように、乗算器は DSP アプリケーションの中心的なものですが、加算、減算または累積(アキュムレート)があとに続きます。汎用 DSP チップは、これら機能の効率的な実装を汎用のマイクロプロセッサ技術に組み合わせています。演算器 (アキュムレータ) 数は一般に 1~4 の範囲にあつて、プロセッサがデータをシーケンス処理し、乗算もしくは他の処理機能を介して、メモリあるいはアキュムレータに中間結果を格納します。性能は主として演算に用いるクロック・スピードを増加させることによって、向上させます。典型的なクロック・スピードは 10MHz から 1GHz までです。性能は” 百万 MAC (MMAC) 毎秒 ” によって表され、通常 10~4000 の範囲です。より高性能を必要とする機能は複数の DSP エンジンに処理を分割しなければなりません。これらチップの価格は、性能の低いデバイスの数ドルから、上位の何百ドルまで及びます。このアプローチの主要な利点は、(サポートするデバイスによっては) C などの高水準プログラミング言語で書かれたアルゴリズムを直接実装できる能力です。

Device	Clock Speed	Number of Multipliers	MMAC/s	1K Unit Cost*	Cost per MMAC/s
TI DSP	1GHz	4	4000	\$256	\$0.064
TI DSP	300MHz	4	1200	\$40	\$0.033
ECP-DSP20	250MHz	28	7000	\$59	\$0.008

北米代理店を通しての 1K 当りの概略値

表 1 汎用 DSP と FPGA アプローチの比較

DSP 指向の FPGA は、1 個のチップに多くの並列機能を実装する能力を提供します。汎用配線、ロジック、およびメモリ・リソースは、機能を相互接続し、追加機能やシーケンスを実行し、必要に応じてデータを格納するのに用いられます。いくつかの基本的なデバイスは、乗算器だけのサポートを提供するため、ユーザが他のすべての機能をロジックで構成する必要があります。より洗練されたデバイスは、DSP ビルディング・ブロックの一部として加算、減算、およびアキュムレータ機能を提供します。FPGA は何十もの乗算器要素を通常持っており、数百 MHz のクロック・スピードで動作することができます。例えば、LatticeECP-DSP20 FPGA には、最大 250MHz の速度で走ることがで

きる 28 個の 18×18 乗算器があり、7,000MMAC 毎秒までの性能を提供します。表 1 は FPGA と汎用アプローチを比較します。

LatticeECP-DSP アーキテクチャ

LatticeECP-DSP デバイスは、低コスト FPGA ファブリックと、4~10 個の sysDSP™ ブロックとの組み合わせから構成されています。図 5 は ECP デバイスの総合的なブロック図を示します。LatticeECP™ ファミリーにおける sysDSP™ ブロックは、4 種の機能要素をサポートし、データパス幅は 9、18、または 36 ビットです。ユーザは、DSP ブロックの機能要素を選択し、次にデータパス幅とタイプ(符号あり/なし)を選びます。これらデータパス幅と、符号あり/符号なしの指定は、複数要素を構成可能な場合でも同一ブロック内では混在できません。sysDSP™ ブロックのリソースで構成することができる 4 つの要素は次の通りです:

- MULT(乗算、図 6)
- MAC(積和、図 7)
- MULTADD(乗算と加算/減算、図 8)
- MULTADDSUM(乗算と加算/減算および足し算、図 9)

各ブロックで有効な要素の数は、選択されたデータパス幅に依存します。これらの複数の要素は、高度に並列度のある DSP 機能の実装のために連結することができます。表 2 はブロックの能力を示します。

乗算器ビット幅 →	X9	X18	X36
MULT	8	4	1
MAC	4	2	--
MULTADD	4	2	--
MULTADDSUM	2	1	--

表 2 sysDSP™ ブロック各要素の最大数

sysDSP™ ブロックは入力、中間、および出力段にオプションとしてのパイプライン用レジスタを内蔵しています。さらに、必要に応じて入力を並列にロード、またはアレイを介してシフトすることロードすることができます。また演算、減算、および加算を動的に符号あり/符号なしで切り換えるオプションがあります。

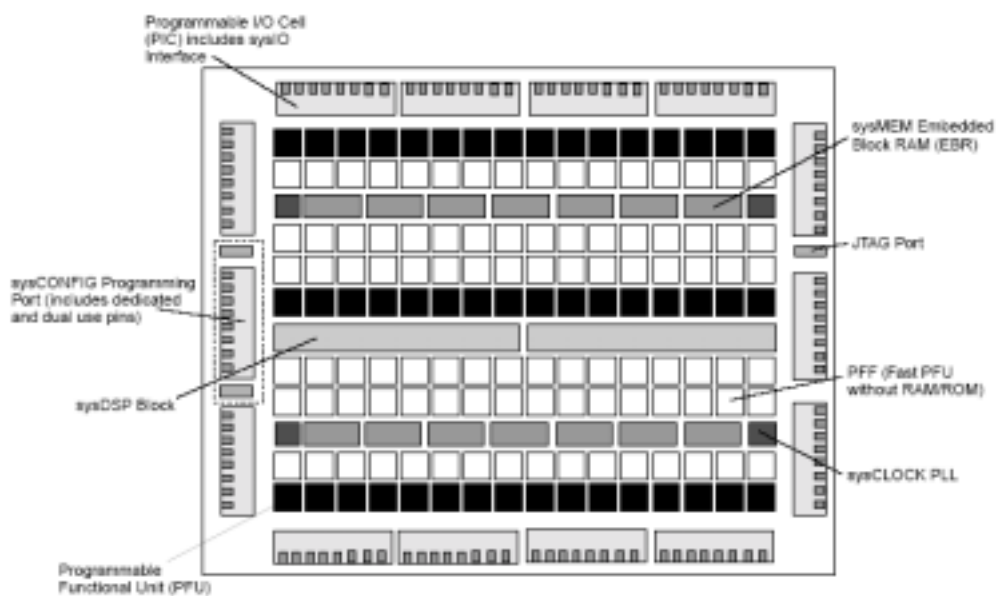


図 5 LatticeECP-DSP ブロック図

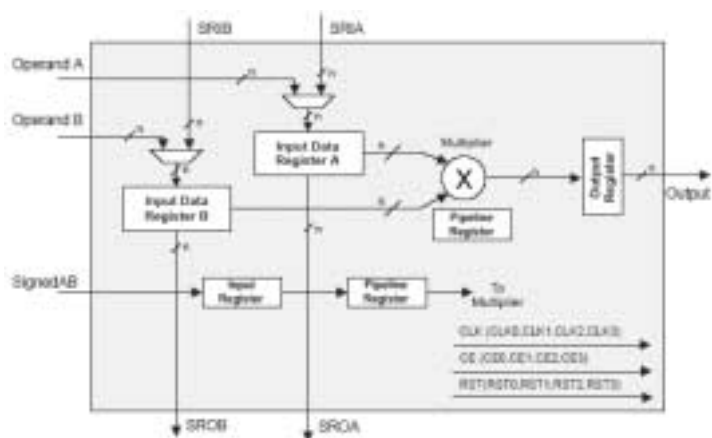


図 6 MULT (乗算器) 要素

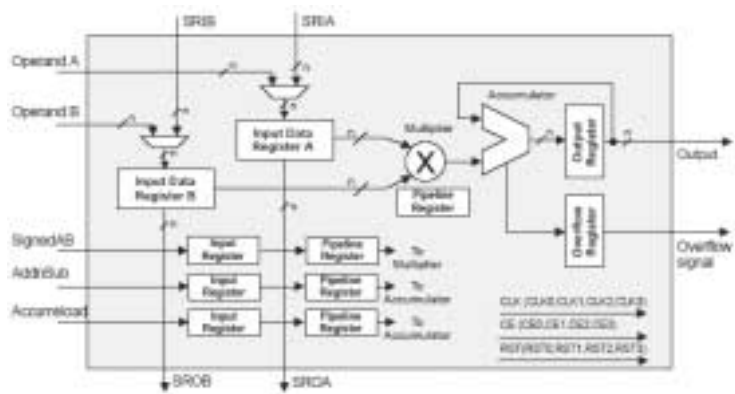


図 7 MAC (積和) 要素

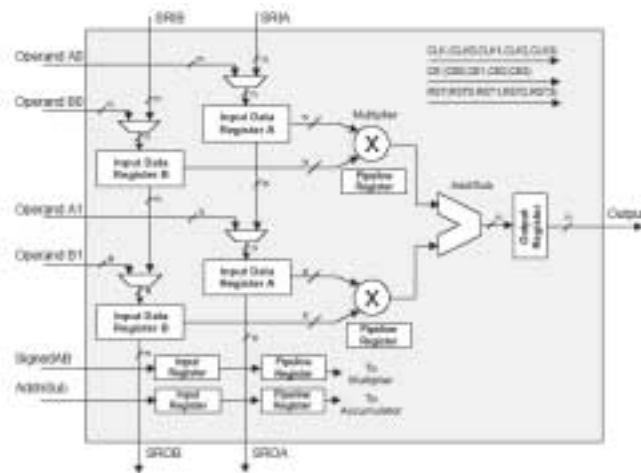


図 8 MULTADD(乗算・加算/減算)

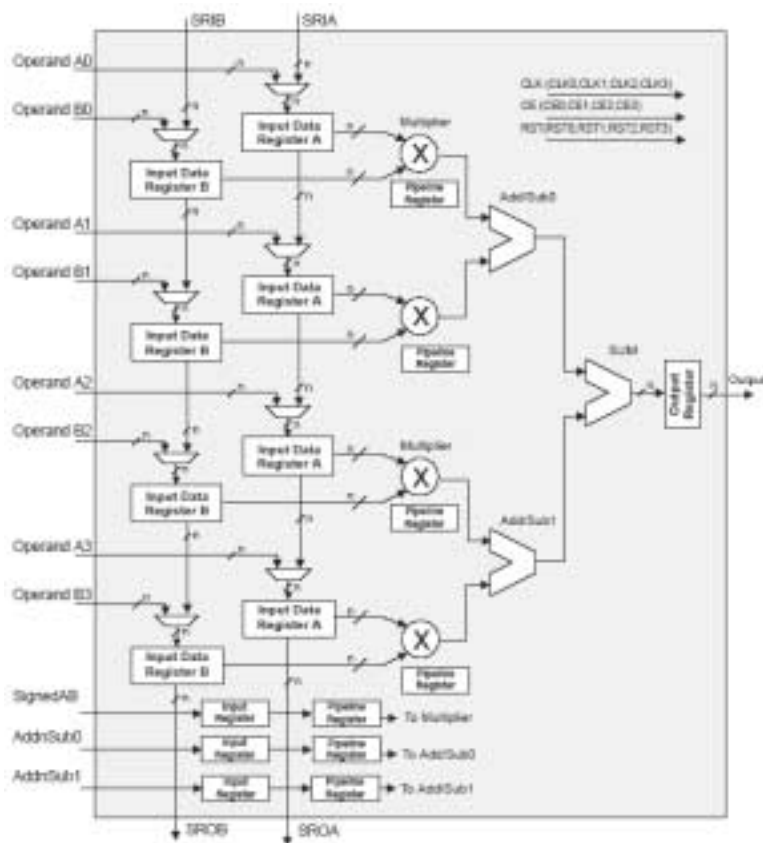


図 9 MULTADDSUM(乗算・加算/減算・足し算)要素

性能とデバイス利用効率の改善

sysDSP™ ブロック内のパイプライン・レジスタ、足し算、減算、および累積(アキュ

ミュレート)機能の提供は、それらの実用性を増加させます。既に述べたように、典型的な機能では、乗算を加算、足し算、または累積(アキュムレート)に組み合わせる必要性があります。パイプライン・レジスタは、広いデータパスで実装されるほど、リソースの消費は急速に顕著になります。これらの機能をすでに実装している sysDSP™ ブロックは、汎用 FPGA 領域のリソースの消費を抑え、かつより高い性能をもたらします。これらは従って低コスト化と直結します。多くのケースで、設計者による遅いスピード・グレードの、より小規模デバイスの選択が可能になるためです。

LatticeECP-DSP デザイン・フロー

ラティスは sysDSP™ ブロックの能力にアクセスする 4 つの簡単な方法を設計者に提供します。

- ispLEVER® ツールに提供されている Module/IP Manager グラフィカル・インタフェースで、DSP 要素を実装するモジュールを迅速に生成すること。HDL ソース内でこれらのモジュールを適宜インスタンスして用いることができる。
- デザインの HDL ソース内に特定の機能をコードで記述することで、論理合成ツールが DSP ブロックの使用を推論できるようにすること。
- MathWorks 社の Simulink ツールで、ラティス・ブロック・セットを用いてデザインを実装すること。ispLEVER® ツールの ispLeverDSP 部が適宜これらのブロックを HDL に変換する。
- ソースコードの DSP プリミティブを直接インスタンス化すること。

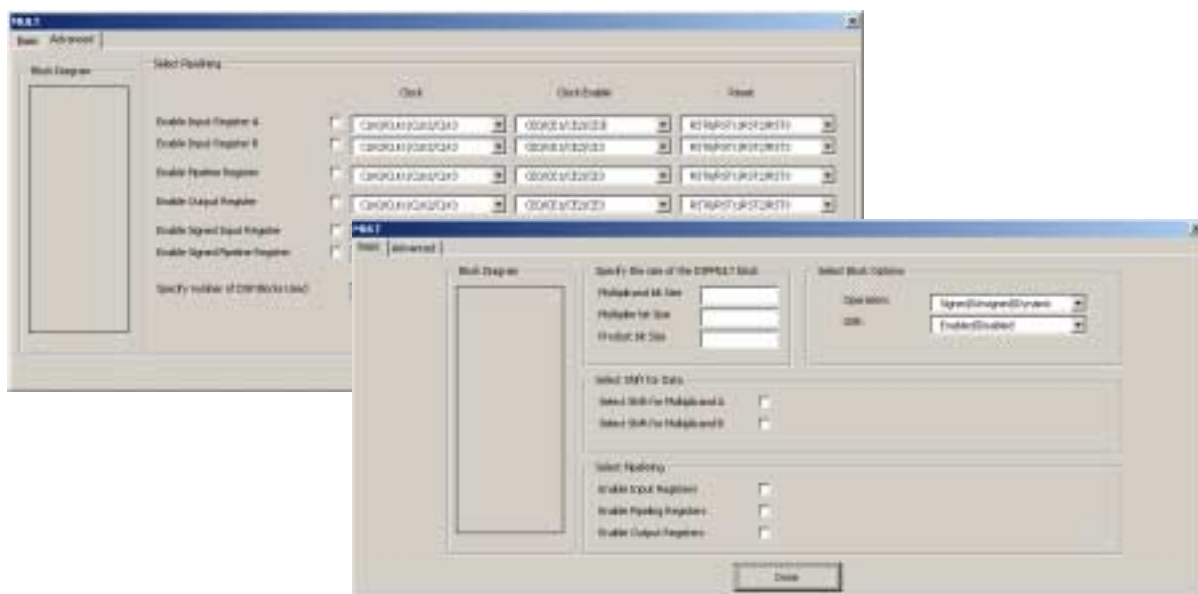


図 10 モジュール内の乗算器要素の構成

実際の設計で選ばれる方法は、DSP アルゴリズムの設計手法と、物理的な実装上必要なコントロールの度合いに依存するでしょう。図 10 は、Module Manager を用いた MULT 要素の仕様を示します。図 11 は MathWorks 社の Simulink ツールにおけるラティス・ブロック・セットの使用例を示します。

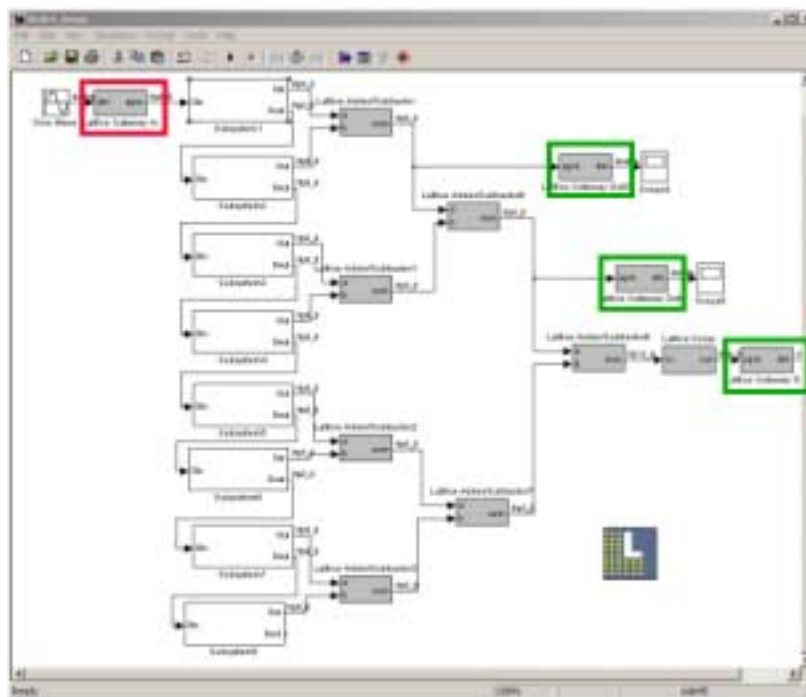


図 11 MathWorks 社 Simulink ツールによるラティス・ブロック・セットの使用例

低コスト FPGA の実装

LatticeECP/EC デバイスの参入で、ユーザは今、現世代の低コスト FPGA 3 種の中から選択することができます。すなわち、Xilinx 社の Spartan III、Altera 社の Cyclone、および LatticeECP/EC です。

Altera 社の Cyclone FPGA は DSP の指向の要素を全く持っていないので、多くの内部リソースを消費しないで大きな DSP 機能を実装するのはとても大変です。当然、これらの実装で高性能を達成することは同じように大変です。Xilinx 社の Spartan III FPGA ファミリはある程度基本的な乗算能力を提供します。確かに、これは DSP 能力を全く持っていないよりは望ましいですが、典型的な設計に見られる加算器と減算器、アキュムレータ、およびパイプライン・レジスタなどを実装するためには、まだ顕著なリソースを消費しなければなりません。

これらリソースを提供することの効果を測定するために、ラティスは FIR フィルタと IIR フィルタの性能と使用効率をベンチマークしました。用いた FIR は 18 ビット幅

データで 64 タップのフィルタで、IIR フィルタは、2 個のバイカッドとしてアレンジした 4 次、18 ビットのデータ・パスです。図 12 に Spartan III と LatticeECP-DSP の両方の結果を示します。

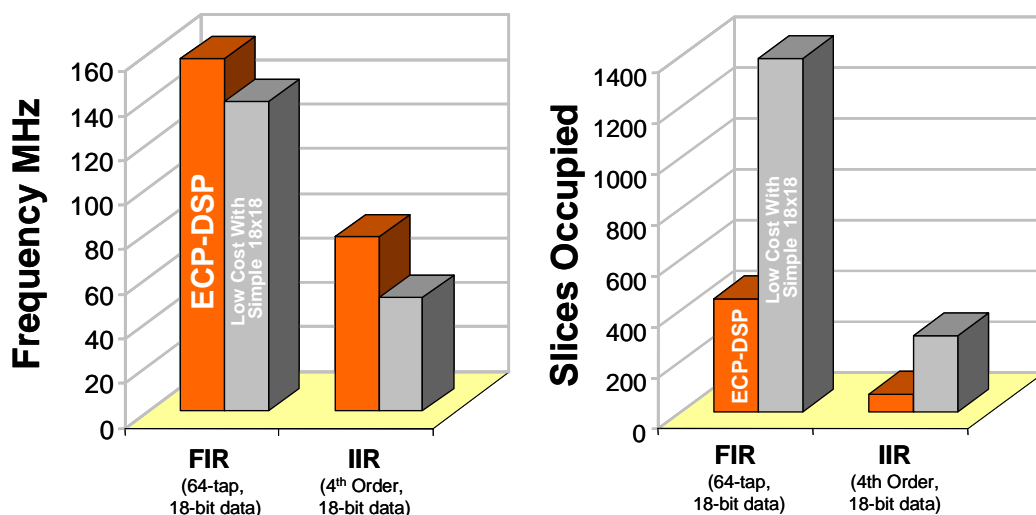


図 12 LatticeECP-DSP と Spartan III の FIR と IIR 実装

まとめ

DSP 技法の使用は、アナログ実装を犠牲にしても成長し続けるでしょう。DSP アプリケーションで通常用いられる機能の分析は、乗算器、加算、減算、および累積(アキュムレート)要素の組み合わせが必要であることを示しています。LatticeECP™ デバイスは低コスト FPGA ファブリックに組み合わせた高性能 DSP ブロックを提供します。sysDSP™ ブロックの中に加算、減算、累積(アキュムレート)、およびパイプライン・レジスタの実装によって、性能と LUT の使用効率は基本的な乗算器能力だけを 제공하는他社の低コスト FPGA ソリューションよりも極めて高くなっています。sysDSP™ ブロックのスピードと使用効率の利点は、ユーザがより小規模で遅いスピード・グレード・デバイスを選択できることによってコストを低減する手助けとなります。

###

(注) LatticeECP-DSP デバイスに用いられた FPGA ファブリックに関するその他の詳細については、ラティス・ホワイト・ペーパー「多量生産アプリケーション用 FPGA の最適化」を参照してください。