

やわらかいハードウェアを実現する STP(Stream Transpose)エンジン

古田 浩一郎・粟島 亨・藤井 太郎
戸川 勝巳・戸井 崇雄・本村 真人

要旨

近年、装置機能の増加・複雑化に伴うLSIへの非常に高度で複雑な処理に対し、CPUを搭載しSoC化することにより、やわらかさ（CPUによるC言語を用いたソフトウェア設計）と硬さ（専用設計による高性能ハードウェア）に切り分けて対応してきました。しかしCPUでの処理は性能が不足がちで、高性能なプログラマブルコアと使いやすい設計環境へのニーズがあり、その解としてSTPエンジンを開発しましたので本稿で紹介します。STPエンジンとその開発環境を用いることにより、やわらかさ（Cベース設計のプログラマブルコア）を保つつつ処理性能を向上できるため、様々な装置の高機能化・高性能化を実現できます。

キーワード

- プログラマブルコア
- プログラマブル演算器アレイ
- DMAコントローラ
- ストリームデータ
- Cベース設計
- 動作合成

1. はじめに

近年のシステムLSI開発においては、製造プロセスの微細化による集積度の向上や要求機能の増加により、開発期間の長期化、開発コストの高騰が顕著になっています。また、この様な課題に対して複数チップの共通化、機能の追加・修正、ハードとソフトの並行開発を目的としてプログラマブルコアを搭載するケースが増えています。しかしながら代表的なプログラマブルコアであるCPUでは、開発効率やデバッグ環境は優れているものの、要求に対して性能が十分でないことがよくあり、またCPUに対して高い性能を有するFPGAは、面積的にコア搭載が難しいことに加え、設計者がハードウェアを強く意識する必要がありました。そのためアルゴリズム開発者が設計できずにハードウェア設計者を介して実装していました。結果、アルゴリズム変更から実装までの時間が長く、性能最適化ループが回りづらいという課題がありました。

2. コンセプト

そこでNECエレクトロニクスでは高い性能を実現するハードウェア（STPエンジン）と、CPUのソフトウェア開発環境に近い設計ツール（C言語で設計できるSTPエンジン統合開発環境）を開発しました（図1）。開発したSTPエンジンとそ

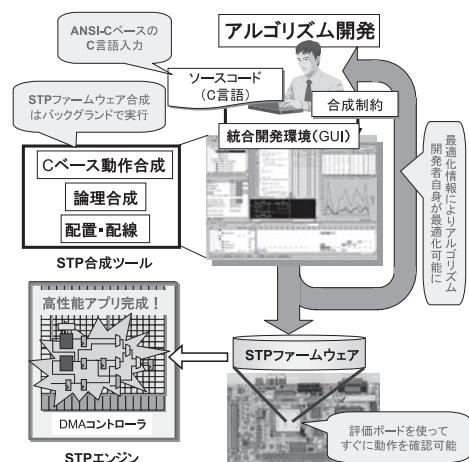


図1 STPエンジンのコンセプト

の設計ツールは以下のような特徴を有しています。

1) STPエンジン（ハードウェア）

- ・ プログラマブル演算器アレイ+DMAコントローラ構造を採用
- ・ 連続して入出力するデータ（ストリームデータ）を1つの処理単位として規格化し入出力インターフェースを単純化
- ・ DMAコントローラによりデータ転送効率を向上（プログラマブル演算器アレイとの同時動作、自律動作）

高性能低電力ロジック製品

やわらかいハードウェアを実現するSTP(Stream Transpose)エンジン

- ・プログラマブル演算器アレイによる超並列処理実行

2) STPエンジン統合開発環境（設計ツール）

- ・C言語で記述されたソースコードからSTPファームウェアを自動生成
- ・動作合成及びバックエンド合成ツールにより高い並列性抽出と演算最適化を実現
- ・高い開発効率を実現する統合設計環境（GUI）
(性能改善情報をCソースコードと関連付けして出力)

3) ハードウェアと設計ツールの相互最適化

- ・設計ツールに合わせたハードウェアアーキテクチャ
(データバス+FSM \leftrightarrow 演算器アレイ+制御回路、高対称性構造)
- ・ハードウェアアーキテクチャに最適化された設計ツール
(STPエンジンの演算器に合わせたスケジューラー、専用バックエンド合成ツール)

以下、開発した設計ツールとハードウェアについてそれぞれ説明します。

3. 設計ツール

STPエンジン統合開発環境には2つの必要条件がありました。1)STPエンジンの処理性能を十分に引き出せること、2)アルゴリズム開発者にとって使いやすくCPUに近い設計生産性の高さを維持できること、です。これらの条件を満たすために以下のような合成ツールと設計環境を開発しました。

1) 動作合成を発展させた合成ツール

NECで20年の実績がある動作合成ツールをコアとして使用し、STPエンジン向けに最適化しました（図2）。動作合成はもともとASIC向けの合成ツールでしたが、C言語入力であることや処理の並列性を引き出す方法などSTPエンジンのニーズや構造との共通点も多く、両者を組み合わせることで最大の効果が期待できました。同時にバックエンド合成ツールを開発することにより、演算器や配置配線レベルでの最適化も行いました。

更にこれらの合成ツールはSTPエンジンの特徴である演算器アレイのスケーラビリティに対して柔軟に適応します。例えば演算器数を変えたSTPエンジンのファームウェアは再合成によりソースコードを変更せずに生成可能です。

2) 統合開発環境

更に弊社ではアルゴリズム開発者に使いやすい環境を目指

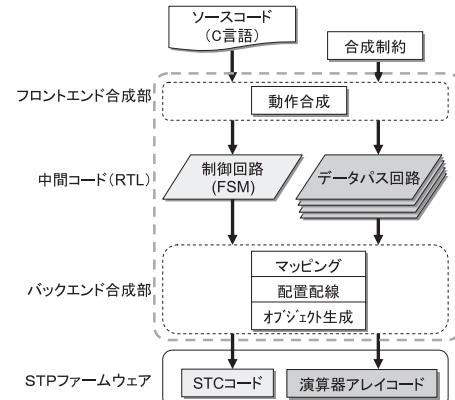


図2 設計ツールの合成フロー



図3 統合開発環境（GUI）

してGUIベースの統合開発環境を開発しました（図3）。この開発環境を用いればユーザは個別のツールを意識することなくボタン1つで合成を行うことができます。また、より高い性能を目指す場合にはオプションの設定やCソースコードの最適化を行います。統合開発環境では最適化に必要な情報をCソースコードと関連付けて確認することができます。具体的には処理のボトルネックになっている状態に要しているサイクル数や演算器の使用数、演算器・メモリへのアクセス情報をCソースコードに対応させて確認できるため、設計者はこれらのヒントを用いて最適化を行うことができます。また、オンチップのデバッグ環境やパラメタを振って合成を行う反復合成機能なども有しております、STPエンジンの合成と最適化、検証を全てGUI上で行えます。

4. アーキテクチャ (ハードウェア)

STPエンジンの動作イメージを図4に示します。STPエンジンは、プログラマブル演算器アレイ+DMAコントローラ構造を採用しています。また、連続したデータ（ストリームデータと呼ぶ）を1つの処理単位として入出力し、処理の投入はキューイング機構を用いて行います。これらは以下の様な理由に基づいています。

- ・データ転送 (DMA) と処理 (プログラマブル演算) を分割、専用化することにより性能及び面積効率を向上させる
 - ・DMAコントローラとプログラマブル演算器アレイを同時に動作させ、データ転送時間を隠蔽する
 - ・入出力データをストリームデータにすることでデータ転送時の初期（ヘッダ）オーバーヘッドを削減する
- また上記性能面以外にも本アーキテクチャの効果として、

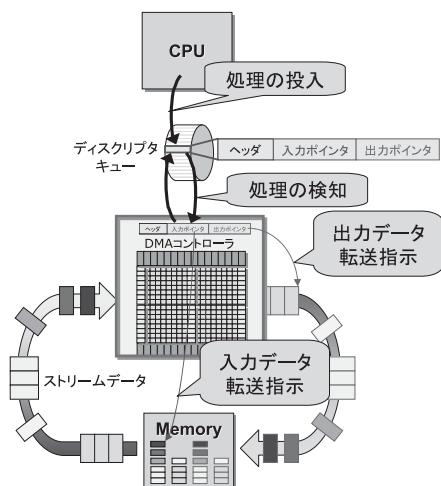


図4 STPエンジンの動作イメージ

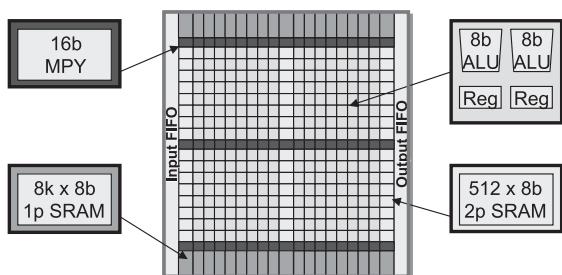


図5 プログラマブル演算器アレイの構成イメージ

入出力が規格化されたためプログラミングモデルを単純化でき、I/Oタイミングなどハードウェアを意識させない開発環境を提供できることも、本構成を採用した理由の1つです。

次にプログラマブル演算器アレイの構成を図5に示します。図のようにプログラマブル演算器アレイは演算器とメモリをアレイ状に配置することにより構成されています。アレイの中心は8bit演算器（ALU）で、その周辺をメモリや乗算器が取り囲む構成になっています。STPではこれらの演算器やメモリを用いて任意の処理を実現し、多数の演算器やメモリを用いて処理を並列に実行することにより高い性能を達成します。

このようにSTPエンジンでは性能のボトルネックになりやすいデータの転送と処理に対して最適なハードウェア構成を用いることにより、高い性能と柔軟性を実現します。

5. XBridge (クロスブリッジ)

STPエンジンを搭載した汎用チップとして開発したXBridgeについて紹介します。XBridgeは複数の標準インターフェースと2つのプログラマブルコア（CPUとSTPエンジン）から構成されるブリッジチップです（図6）。メモリコントローラに接続されるSDRAMを介して異なるインターフェース間でデータの載せ換えを行うことができます。また搭載されているCPUとSTPエンジンを用いることにより、処理全体（CPU部分とSTP部分の両方）をC言語で設計しながら、CPUでは性能が達成できない部分をSTPエンジンで加速することができます。

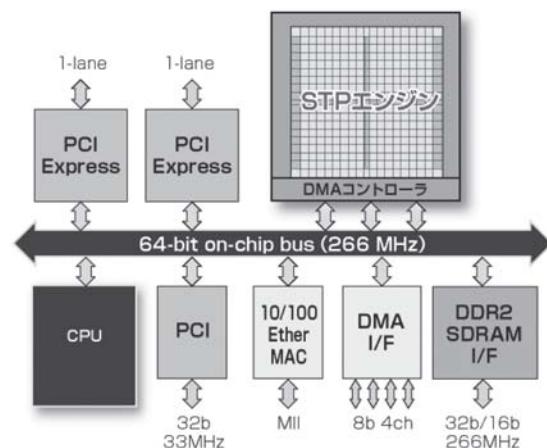


図6 XBridgeブロック図

高性能低電力ロジック製品

やわらかいハードウェアを実現するSTP(Stream Transpose)エンジン

例えばPCI-Expressを用いて既存のシステムにXBridgeを接続すれば、既存システムの機能追加や性能向上を行うオプションボードとして使うことができます。

6. 評価キット

上記XBridge及びSTPエンジンの評価キット（評価ボードと評価環境）を開発しました（図7）。開発した評価ボードはXBridgeとその動作に必要なデバイス（SDRAM, FlashRAMなど）を搭載しており、XBridgeの多彩なインターフェースを用いて既存システムと接続できます。また、評価ボードはPCと接続（PCI-Expressまたはイーサネット経由）することによりSTPエンジン評価環境としても使用できます。STPエンジン評価環境では、コーディング、デバッグなどを体感できると同時に実際のデータを用いて性能評価も行えます。

7. アプリケーション評価例

STPエンジン評価環境を用いて画像処理アプリケーションを評価した結果を示します（図8）。対象とした処理は画像

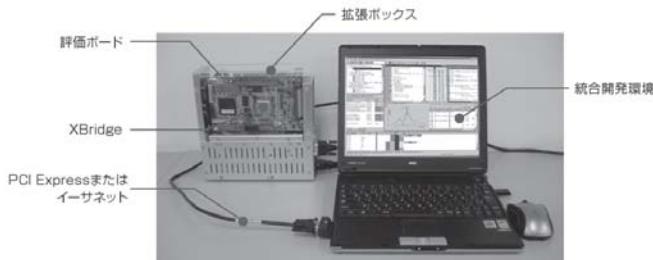


図7 XBridge・STP評価キット

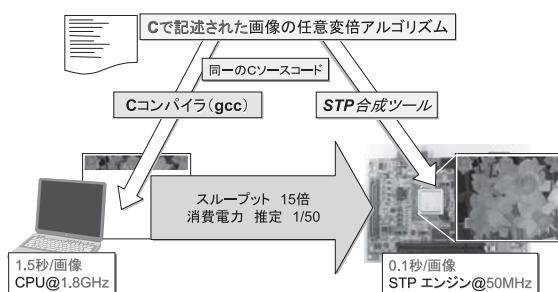


図8 同一CソースコードからのCPUとの性能比

の変倍処理でPC上のCPUとの性能比較を行いました。結果、Cソースライン数は700行、STPエンジンの出力スループットは50Mバイト/秒（200%拡大時）となりました。同じ入力CコードをCPUで処理した場合と比べると15倍程度の性能向上になります。またSTPエンジンは50MHzで動作しており、電力は1/50（推定）になります。これはSTPの演算器及びメモリの並列性が抽出されたこと、データ転送とデータ処理を同時実行したことなどの効果によるものです。

8. むすび

本稿では、入力言語としてC言語を用い、ソフトウェア設計に近い開発環境と、CPUより高性能な処理を実現するSTPエンジンについて説明しました。またSTPエンジンを搭載したXBridgeについても紹介しました。

今後、セット開発の複雑化に伴いC言語設計と高性能プログラマブルコアへのニーズはますます強くなることが予想されますが、弊社ではSTPエンジンの技術と開発環境を通じてお客様セット開発の性能向上、柔軟性向上、短TAT化などに貢献してまいります。

参考文献

- 1) 本村 真人;CベースプログラマブルHWコア「STPエンジン」の現状と展望,信学技報, Vol.108, No.300, pp.51-56, RECONF2008-48

執筆者プロフィール

古田 浩一朗
NECエレクトロニクス
第1SoC事業本部
システムASIC事業部
チームマネージャー

栗島 亨
ビジネスイノベーションセンター
EDA開発センター
主任研究員

藤井 太郎
NECエレクトロニクス
第1SoC事業本部
システムASIC事業部
主任

戸川 勝巳
NECエレクトロニクス
第1SoC事業本部
システムASIC事業部
主任

戸井 崇雄
ビジネスイノベーションセンター
EDA開発センター
主任

本村 真人
NECエレクトロニクス
第1SoC事業本部
システムASIC事業部
プロジェクトマネージャー