

ストリームアプリケーションを対象としたスケーラブル FPGA システムの提案 Proposal of Scalable FPGA System for Stream Processing

久原 拓也[†] 吉見 真聡^{*} 三木 光範^{*}

Takuya Kuhara Masato Yoshimi Mitsunori Miki

1. はじめに

近年、科学計算に要するデータ量や演算量が膨大となり、大規模処理にはスーパーコンピュータや PC クラスタなどの高性能計算システムが用いられるようになった。これらのシステムにより、大規模処理を現実的な時間で実行可能になったが、その一方で、システムの導入に要する設備コスト、電力や空調、メンテナンスなどの運用コストの増大が問題となってきた。

そのような状況の中で、定型的な処理を専用ハードウェアで行うことにより、高い演算性能を維持したまま、システムのサイズや電力消費量の低減を可能にする手法が研究されている。この手法に適した処理として、動画処理やストリーム暗号処理のような、入力データを順に計算するストリームアプリケーション処理がある。ストリームアプリケーション処理に特化したハードウェアは、代表的なものでは GPU や Cell があるが、その他に FPGA(Field Programmable Gate Array) による専用ハードウェア構築も多く研究されている。FPGA は、柔軟に専用ハードウェアを再構成できるデバイスとして注目を集めており、実際に Splash2 や CUBE などの優れた性能を持つシステムの構築例もある [1]。FPGA は複数接続によるシステム拡大が可能であり、より優れた専用ハードウェアの構築が期待できるが、増設する FPGA のコストや、扱いが困難である点などいくつかの問題も持ちあわせている [2]。そこで、本研究では、ストリームアプリケーションを対象とした、安価で扱いやすいスケーラブルな FPGA システムの提案、構築を行う。

2. スケーラブル FPGA システムの提案

図 1 に提案するシステムの構成図を示す。

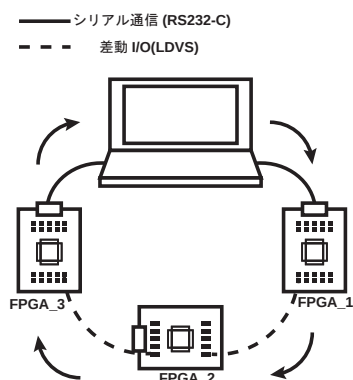


図 1 スケーラブル FPGA システムの構成

図 1 に示すように、PC からデータを受信する FPGA(図 1 の FPGA.1), FPGA 同士でデータの中継する FPGA(図 1 の FPGA.2), PC にデータを送信する FPGA(図 1 の FPGA.3) の 3 種類の FPGA がシステムに接続されており、PC からの入力信号が、複数の FPGA を介して処理され、PC に出力さ

れる構成を取る。FPGA は直列接続し、シストリックアレイでデータを処理する。シストリックアレイは制御信号でコアを同期させると共に、近接コア同士でデータを授受して、データをパイプライン処理する処理方式であり、ストリームアプリケーションに対する高速な処理が期待できる。

3 種類の FPGA に対し、ハードウェアを Verilog-HDL を用いて設計する。本システムでは、全てのハードウェアを、送信モジュール、受信モジュール、処理モジュールの 3 つに分離して設計する。図 2 にモジュール設計図を示す。

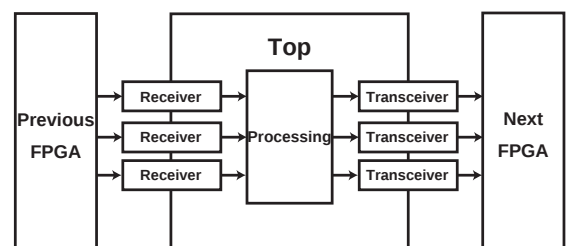


図 2 モジュール設計

図 2 のようにシステムをモジュラーに構成することで、処理モジュールを置き換えて、様々なアプリケーションをシステム上に実現可能になる。

3. スケーラブル FPGA システムの実装

提案したシステムについて、実装を行った。システムに組み込む FPGA ボードとして、50MHz 駆動の Xilinx Spartan-3AN を 3 台用い、PC と FPGA の接続にはシリアル通信 (RS232-C) を、FPGA 間の接続には差動 I/O(LDVS) を用いた。通信速度は、シリアル通信、差動 I/O 共に 9600bps となるように設計を行った。

また、動作確認の為に、本システムにアプリケーションとして 32bit 版 Mersenne Twister を実装した [3]。32bit 版 Mersenne Twister とは、1 クロックサイクルにつき 1 つの 32bit 乱数を生成するアプリケーションである。今回の動作確認では、Mersenne Twister を、FPGA_1 にのみ実装した。FPGA_1 は常に乱数を生成し続け、PC から ASCII 文字を受信したときに、1 つの乱数を残し 2 枚の FPGA を通して PC に送信するというような設計を行った。

4. 評価

以下の 3 つの構成で、乱数を 1000 個生成し、PC で取得するまでの時間を測定する実験を行った。図 3 に実験構成を示す。

- (1) FPGA1 枚を PC とシリアル接続したもの。
- (2) FPGA2 枚を PC とシリアル接続し、2 枚を差動 I/O で接続したもの。
- (3) FPGA2 枚を PC とシリアル接続し、2 枚の間に FPGA を 1 枚差動 I/O で接続したもの。

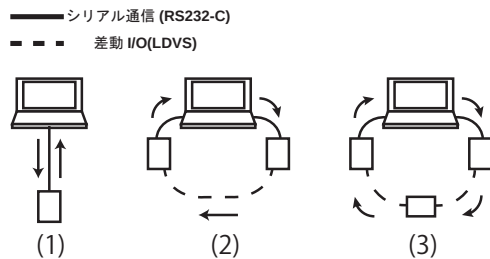


図3 実験構成

各構成において10回ずつ実験を行い、時間の平均、受信した有用なデータ量 (net), 受信した総データ量 (gross) を算出した。結果を表1に示す。

表1 通信性能 (理論性能 9600bps)

構成	時間 [s]	net[bps]	gross[bps](理論値との比率)
(1)	5.85	6830.35	8537.81(88%)
(2)	7.17	5576.07	8364.11(87%)
(3)	8.22	4861.50	8507.63(88%)

本システムの Mersenne Twister では 32bit の乱数データを生成するが、これをシリアル通信で PC に送信するために、32bit の乱数データを 8bit ずつに区切り、4 回に分けて送信している。そのため、スタートビット、ストップビットを含めた1回の乱数送信辺りの送信データ量は 40bit となる。以上より、構成 (1)(2)(3) で PC が受信した有用なデータ量を、表1の net 項のように求められる。

このデータ量は、理論的な通信量である 9600bps と比較すると、(1) では 2800bps 程度、(2) では 4000bps 程度、(3) では 4800bps 程度少ない値となっている。今回、乱数送信のトリガである ASCII 文字の送信を、PC が一つの乱数を受信してから行うようプログラム側で設定した。これにより、FPGA が PC に乱数データを送信しない時間帯が発生し、データに有用でないデータが混じったと考えられる。図4に、PC から ASCII 文字が送信され、FPGA_1 が FPGA_2 に、FPGA_2 が FPGA_3 に、FPGA_3 が PC にデータを送信するまでのタイムチャートを示す。

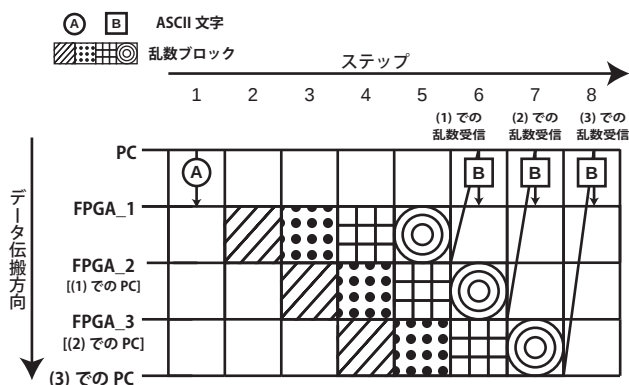


図4 データ送信のタイムチャート

図4に示したように、(1) では送信サイクル5ステップ中1ステップで、(2) では送信サイクル6ステップ中2ステップで、(3) では送信サイクル7ステップ中3ステップで FPGA が PC に乱数データを送信できない時間帯が発生している。これらを含めると、構成 (1)(2)(3) で PC が受信した実際のデータ総量は、表1の gross 項のように求められる。

net 項の値に比べ、gross 項の値と理論値との差が小さいものになっていることがわかる。gross 項の値と理論値との 1400bps 程度の差は、PC が FPGA から乱数を受け取ってから次の ASCII 文字を送る際の遅延や、乱数生成の時間が積み重なって生まれたものと考えられる。

以上の結果から、本システムは、通信処理がボトルネックとなっていないが、アプリケーション処理、信号の送受信が正常に動作していることが確認できた。今回の実験では FPGA1 枚にアプリケーション処理を実装したが、複数枚の FPGA に処理を分散することで、高速なアプリケーション処理が可能となる。また、本システムは FPGA を増設することが可能であり、より高速なシステムを構築も可能である。今回用いたプログラムでは、FPGA の増設に伴いスループットが低下していくが、プログラムを、乱数データ受信途中で次の ASCII 文字を送信できるように変更することで、高スループットな通信が可能になると考えられる。

5. まとめと今後の展望

本研究では、ストリームアプリケーション処理を対象とした、安価で扱いやすくスケラブルな FPGA システムを提案した。また、実際に3台のFPGAを用いてシステムを構築し、Mersenne Twister を実装して動作確認実験を行った。

今回、構築したシステムでは、通信処理が大きなボトルネックとなっているが、シリアル通信でより高速なデータレートを用いたり、差動通信で用いるピン数を増やし、パラレルで送受信を行うことで、通信速度を高速化できると考えられる。今後の展望としては、これら通信処理の改善や、実際に複数枚のFPGAでストリームアプリケーションの評価を行う他、システムをより扱いやすいものにするため、最小構成のFPGAボードの作成などを行いたい。

参考文献

- [1] 吉見 真聡 and 西川 由理 and 天野 英晴 and 三木 光範 and 廣安 知之 and オスカーメンサー. ストリームアプリケーション向け大規模 FPGA アレイ CUBE の性能評価. 情報処理学会論文誌. コンピューティングシステム, Vol. 3, No. 3, pp. 209–220, 2010-09-17.
- [2] 高前田伸也, 渡邊伸平, 姜軒, 藤枝直輝, 植原昂, 三好健文, 古瀬謙二. メニーコアアーキテクチャ研究のためのスケラブルな HW 評価環境 ScalableCore システム. 情報処理学会研究報告. 計算機アーキテクチャ研究会報告, Vol. 2009, No. 3, pp. 1–10, 2009-10-19.
- [3] 渡部信吾, 阿部公輝. 疑似乱数生成器 Mersenne Twister の VLSI 設計. 情報処理学会研究報告. CSEC, [コンピュータセキュリティ], Vol. 2005, No. 41, pp. 13–18, 2005-05-19.