

FPGAを用いたFFT処理ROSノードの初期評価

星野 美如[†] Pinheiro Leal Daniel[‡] 渡辺 晴美[‡] 佐藤 未来子[‡] 大川 猛[‡]
 東海大学大学院情報通信学専攻情報通信学研究科[†] トロント大学[‡]
 東海大学情報通信学部組込みソフトウェア工学科[‡]

1. はじめに

ロボットシステム開発では、制御のために用いられるデータ解析処理の低遅延化が求められている。複雑な制御を可能とするためには膨大なデータを扱う必要があるが、データ量の増加に伴い計算処理時間も増加してしまう問題がある。これを解決する為、データ解析処理にFPGAを用いることで低遅延な処理を行い、さらにROSノードとして実現することでロボットシステム開発への容易な導入を可能とするFFT処理ROSノードを提案する。なお本研究では、センサデータ・画像・音声など様々な分野でデータ解析に用いられるFFT処理に着目し、データ解析処理の一例として研究対象とした。

本稿では、提案した環境の設計・実装・初期評価について述べる。

2. FPGAを用いたFFT処理の実装

FPGA(Field-Programmable Gate Array)はユーザが実現したい機能に合わせて論理回路をプログラムすることができる論理デバイスである。FPGAではDSP(Digital Signal Processor)コア同士やDSPコアとユーザがプログラムした演算回路などを自由に接続できるため、固定された演算回路とそれらに接続されているハードウェアで予め決められた命令セットを実行するCPU・GPUと比較して遅延時間を安定的に短縮することができる特徴がある。

本研究で我々が提案する環境を実現するための予備実験として、FPGA上で16-1024点のFFTを行いその処理速度を計測した。開発環境としてVivado2019.1, Xilinx社製のFFT処理IPコアおよびサンプルデザイン[1]を使用した。また、FPGAの処理性能と比較するため、ARMとPCでも同様に16-2048点のFFTを行い、その処理速度を計測した。それぞれの実行環境詳細を表1, 実験結果を表1・図2に示す。

表1 実行環境詳細

	ボード	ライブラリ
FPGA	ZedBoard (Xilinx)	FFT IP (Xilinx) [1]
ARM	RaspberryPi3 Model B Plus Rev1.3	FFTSS (The SSI Project) [2]
PC	Intel (R) Xeon(R) CPU E3-1225 v6 @3.30GHz	FFTSS (The SSI Project) [2]

表2 FFT処理時間測定結果(単位: μs)

	FPGA	ARM	PC
16	10.50	2.00	0.08
32	10.84	3.90	0.18
64	10.57	9.16	0.40
128	13.81	20.33	0.81
256	20.51	41.64	1.70
512	33.82	92.23	3.84
1024	60.28	95.49	8.46
2048	113.92	275.17	24.22

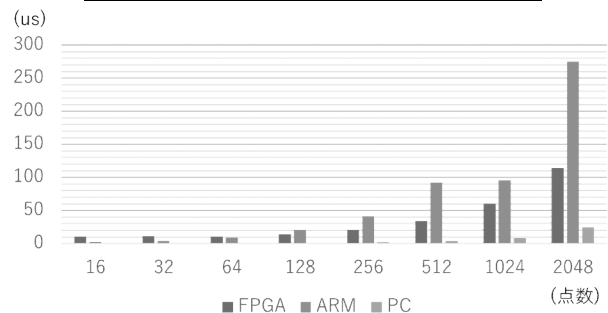


図1 FFT処理時間測定結果

2048点FFTの処理時間は、FPGAでは約114 μs , ARMでは約275 μs , PCでは約24 μs であった。PCはFPGAよりも高速で処理を行うことができるものの、消費電力が高くロボットシステムへの搭載は困難である。FPGAは最速でも10 μs 程度の処理時間を要し、計算量の少ない場合にはソフトウェアでの処理に劣るが、計算量が増えていくと結果が逆転し、128点以上のFFT処理においてはARMよりも高速に処理を行うことができた。特に、2048点FFTではARMの1/2以下の速度で処理を行っていることがわかる。

Preliminary Evaluation of FFT Processing with ROS Node on FPGA

[†]Mikoto Hoshino, Pinheiro Leal Daniel, Harumi Watanabe, Mikiko Sato, Takeshi Ohkawa

[‡]Tokai University, University of Toronto

3. F0rESTによるFPGAのROSノード化

本研究で用いるF0rEST (FPGA-Oriented Easy Synthesizer) [3][4]とは, Xilinx Zynq SoCを用いたHLSベースのFPGAモジュールをROS2システムに統合するオープンソースツールである. F0rESTはユーザが実現したい処理をVivado HLSで高位合成して生成したFPGAロジックをROS2ノード化するために必要なファイルを生成・構築し, ROS2とFPGA間で通信するPYNQドライバを自動生成することで, 他のROS2モジュールとFPGAハードウェアが容易に通信可能となる環境を提供する. 図2は, F0rESTによって生成されるROS2-FPGAコンポーネントの構造を表している.

本研究では, F0rEST用いてFPGAをROS2ノード化することで, FPGAの設計コストを解決し, ロボットシステムへの容易な実装を可能にした.

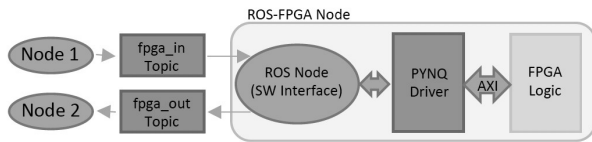


図2 ROS2-FPGAコンポーネント構造

4. ROS2ノード通信遅延時間の評価

次に, ROS2ノード通信遅延時間を評価するため, 単純な処理をHLS記述し, F0rESTで生成したFPGA処理ROS2ノードによる処理時間を計測した. 実験用の処理内容は, 512-2048個のデータの平均値をそれぞれのデータから引いた値を出力するものとした. 計測対象は, ROS2の通信遅延時間, FPGA処理ROS2ノードがPythonのstruct_pack / struct_unpackを用いた浮動小数点の変換を行いFPGAロジックにデータを読み書きする時間, およびFPGAロジック内での処理時間とした. 計測結果を表3に示す.

ROS2の通信遅延時間は約1.3-1.5ms, FPGA処理時間は2048点でも0.5ms以下と高速であったが, 浮動小数点の変換およびMMIOを用いたFPGAロジックへのデータの読み書きでの遅延が大きく, 2048個のデータを扱う場合では約80ms前後もの処理時間を要した.

5. 評価

これらの実験をもとに, FPGAを用いたFFT処理ROS2ノードの性能を評価するため, 前項で計測したFPGA処理ROS2ノードの通信遅延時間と, 予備実験で計測したFPGA上でのFFT処理時間を合わせ, FPGAを用いたFFT処理ROS2ノードの予測処理時間を算出した. 結果を表4に示す.

結果から, FPGA上でのFFT処理時間よりも入出

力データをPythonで処理している時間が大きく, 遅延の原因となっていることがわかる.

表3 ROS2通信遅延およびFPGA処理時間(単位: μs)

FFT 点数	512	1024	2048
通信遅延 (入力)	1,349.15	1,493.43	1,361.30
入力データ変換・入力	19,426.78	39,643.81	78,237.69
FPGA 処理時間	250.11	339.90	460.76
処理結果変換・出力	20,350.30	40,621.64	81,048.13
通信遅延 (出力)	1,347.77	1225.38	1,115.12

表4 FFT処理ROS2ノード予測処理時間(単位: μs)

FFT点数	512	1024	2048
FPGA処理外遅延時間	42,474.00	80,533.50	161,762.24
FFT処理時間 (FPGA)	33.82	60.28	113.92
予測時間 (合計)	42,507.82	80,593.78	161,876.16

6. おわりに

本研究では, ロボットシステム開発でのデータ解析処理の低遅延化を目的として, FFT処理をFPGA上で行い, さらに同システムの汎用化を目的として, FPGA処理をF0rESTによってROS2ノード化し, 初期評価を行った.

結果, 計算量が多い場合においてFPGAは優れた速度での処理が可能であることがわかった. しかし, FPGA上でのFFT処理をROS2ノードとして実現した場合, Pythonでの浮動小数点の変換およびFPGAロジックへのデータの読み書きに時間を要してしまうため, データの入力から処理結果の出力までにかかる時間は大幅に遅延してしまうことが示された. 上記遅延時間を短縮する方法の検討が今後の課題である.

謝辞

本研究は, JST, CREST, JPMJCR19K1 の支援を受けたものです.

参考文献

- [1] <https://xilinx.com/products/intellectual-property/fft.html>
- [2] <https://www.ssisc.org/fftss/>
- [3] <https://github.com/ros2-forest/forest>
- [4] Daniel Pinheiro Leal, et al., "Automated Integration of High-Level Synthesis FPGA Modules with ROS2 Systems," Proc. Intl. Conf. on FPT20 Dec. 09-11, 2pages, 2020.