

学籍番号 氏名	V17093 中村 光希	指導教員	藤枝 直輝
題目	PYNQ プラットフォームを用いた可視化システムのプロトタイプ		

1 背景と目的

データやその解釈を受け手に分かりやすく伝えるには、データの可視化が重要である。得られたデータを適切な図や表で表すことは、工学系の文章作成に求められるスキルの 1 つである [1]。さらに切り口を変えて可視化を行うことで、受け手に新しい視点を提供できることもある。

物理現象から乱数を生成する真性乱数生成器 (TRNG) では、手法によっては測定値の分布を見ることで、生成される乱数の質をある程度推測できる。例えば Fujieda らが提案した FPGA (Field Programmable Gate Array) 向けの TRNG [2] では、測定によって得られたカウンタの値の分布をヒストグラムやヒートマップといった形で可視化している。可視化については事後的なデータ処理によって行われているが、リアルタイムで行うことができれば、動作状況をその場でモニタリングできる可能性がある。

本研究では可視化のプラットフォームとして PYNQ[3] に注目する。PYNQ とは FPGA とプロセッサをワンチップに搭載した Zynq と呼ばれるチップ向けに構成されたプラットフォームである。Python 言語とライブラリを使用することで FPGA の機能を Python から利用できる。今回使用するボードは HDMI 端子を 2 つ備え、ビデオ入出力を FPGA から制御することができる PYNQ-Z1 ボードである。これに対する可視化システムのプロトタイプを作成し、問題点を明らかにすることで、今後の具体的な可視化システム設計への足がかりとする。

2 ソフトウェア処理によるプロトタイプ

第一のプロトタイプとして、ソフトウェア処理による可視化のプロトタイプを作成した。PYNQ の Web サイト [3] からダウンロードできるバージョン 2.6 の SD カードイメージを使用したデフォルトデザインとそこから HDMI 出力に関係しないハードウェアを取り除いたデザインを用意する。これらのそれぞれに対して画像を描写し、HDMI 出力した際にかかる時間を比較する。Python でのコード記述は同じものを使用する。画像は虹を模したものとし、pillow と呼ばれる画像処理ライブラリを使用した。性能評価を行うにあたっては time モジュールを使用し、PYNQ デザインの読み込み、画像の作成にかかる処理時間の測定した。

測定の結果、PYNQ デザインの読み込みに関して、デフォルトデザインでは 6.89 秒、自作したデザインでは 1.72 秒となり 4 倍の差が生じた。このことから、不要なハードウェアをデザインから取り除くことで、初期化にかかる時間を短くできることが分かった。画像の作成にかかる処理時間に関しては、デフォルトデザインでは 0.08 秒、自作したデザインでは 0.07 秒と差は大きく生じない結果となった。

3 ハードウェア処理によるプロトタイプ

ソフトウェア処理では画像を HDMI 出力したが、リアルタイムで動作状況をその場でモニタリングするためには画像ではなく、映像として表示し続けることが必要である。しかし、ソフトウェア処理だけではフレームレートの高い滑らかな画像を表示するのは難しい。そのため第二のプロトタイプとしてグラフィックパターン送信回路を用いた、ハードウェア処理による可視化のプロトタイプを作成した。

パターン送信回路はヒートマップを用いた可視化を念頭に置き、配列の要素に対応した色をスクロールしながら表示するものとした。パターン送信回路や HDMI 出力を含んだデザインを作成し、パターン送信回路を定期的に動作させるプログラムを Python で作成した。実機実験の結果、想定したとおりの映像パターンが HDMI で出力されることを確認した。

4 結論

本研究では、PYNQ プラットフォームによる可視化システム設計のために、ソフトウェア処理とハードウェア処理による 2 種類のプロトタイプを検討した。これらのプロトタイプから得られた知見をもとに、具体的な可視化システムを実現していくことが、今後の課題である。

参考文献

- [1] 西出利一, 理系のための文章術入門, 化学同人, 2015.
- [2] N. Fujieda and S. Takashima, An MMCM-based high-speed true random number generator for Xilinx FPGA, International Journal of Networking and Computing, vol. 11, no. 2, pp. 154-171, 2021.
- [3] PYNQ: Python Productivity, <http://www.pynq.io/>, 2022-02-01 参照.