

# Android アプリからの FPGA の利用による処理の高速化

電気通信大学 大学院 情報・通信工学専攻 成見研究室

1331051 塩谷 丈史

## 1. はじめに

近年では、携帯端末の普及と利用方法の多様化により、アプリケーションから要求されるスペックや機能が増加している。これは専用の周辺回路やモジュール、チップを搭載することで問題はある程度解消されるが、携帯端末のプロセッサには電力や放熱の問題によりそれらの要求に応えるには限度がある。FPGA を用いてアプリケーション毎に回路を書き換えられれば、個別の専用回路が無くても専用回路並みの性能が期待できる。

PC を利用する場合には、PC-FPGA Hybrid Cluster [1] に見られるように CPU と FPGA を搭載したシステムを使用することで、アプリケーションに最適化された回路を使用した例がある。近年、ARM コアと FPGA を組み合わせた Programmable SoC と呼ばれる製品が出現しており、本研究ではこの方法をモバイル向けに適用することを考えた。

本研究では、モバイル端末での利用を前提として、FPGA を用いた Android タブレットを試作し、いくつかの数値演算専用アプリを実装した(図 1)。Android から FPGA を利用する方法は Reconfigurable Android[2] で示されており、利用方法の改良と Zynq チップへの対応を行った。

## 2. 関連研究

尾崎らは FPGA を搭載した PC を FPGA 側のネットワーク回路を通して複数つなぐことで PC-FPGA の複合クラスタを構築している(図 1)。PC から FPGA の回路を書き換えることで、専用回路を使用した処理をネットワーク越しに行うことが可能となる。本研究では Programmable SoC を使用することでモバイル向けシステムを開発する点が異なる。

小池らは FPGA と CPU(x86 系 Intel Atom)を搭載した評価ボード上に Android を動作させ Android アプリの一部処理を FPGA で行うことで高速化する Reconfigurable

Android を提案している。このシステムは、DMA 転送を使用することで回路へデータを転送し FPGA で処理を肩代わりさせるという点で本研究と類似しているが、FPGA への再構成をあらかじめ行うことを前提としており OS 起動中にアプリを切り替えられない。本研究ではアプリ自身が回路を再構成するため OS を起動したまま異なる専用回路に切り替えることが可能である。

## 3. 提案システム

### 3.1. 概要

本システムは ZedBoard 上で構成される(図 2)。搭載された ARM CPU により Android OS を動作させ、アプリはドライバを通して FPGA の部分再構成と回路へのアクセスを行う。

### 3.2. デバイス

Xilinx が販売している Programmable SoC を搭載した ZedBoard をメインに使用する。また、OS を動作させるための周辺回路を FPGA 部分に構築する必要があり、Xilinx から提供されている ZedBoard\_OOB\_Design を使用した。また、タブレット型にするため、内蔵電源及び、無線ネットワークへの対応を行い、携帯可能にした。さらに、3D CAD を使用してフレームを設計し、3D プリンタによって出力し、タブレットの形にした。

### 3.3. パーシャルリコンフィギュレーション

PR 機能を使用することで、FPGA 内部の回路を部分的に動作中に書き換えることができる。これを利用してアプリごとに処理用の回路を切り替える。部分的に書き換えることで、書き換えの時間を短縮することが可能である[3]。PR で書き込むための回路データは Xilinx Wiki[4] を参考に作成した。

### 3.4. アプリ専用処理回路

処理回路は DMA Engine と接続され、DMA Engine は AXI バスを通して ARM コア及びメモリコントローラと接



図 1 試作した FPGA タブレット

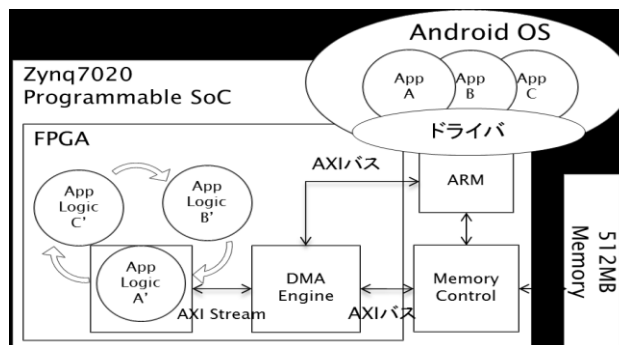


図 2 システムの構成

続される (図 2). 処理回路へのデータ転送は DMA によって行われ, バス幅は 64bit である. 処理回路自体はアプリによって異なり, 使用時に書き換えられる.

### 3.5. ソフトウェアの作成

#### 3.5.1. 使用したソフトウェア

Xilinx-ISE Design Suite 14.4 を使用して回路を作成した. また, ZedBoard 上で動作させるための OS として, Zynq Android 4.1 [5] を使用した. なお, Zynq Android 4.1 は起動ができず, さらに動作が不安定だったため, ソースコード等の修正を行った.

### 3.6. ドライバ

PR ドライバ, DMA ドライバ, メモリアクセスドライバの 3 つを使用する. このうち, PR ドライバについては Digilent が配布している Linux Kernel バージョン 3.6 にドライバ `xdevcfg(xilinx_devcfg.c)` のコードがあるのでこれを使用した. DMA ドライバは主に AXI によるメモリマップド I/O を読み書きすることで, DMA Engine に対し, 読み書きアドレスやパラメータのセット, 動作命令を行う. Xilinx のドキュメント `pg021_axi_dma` を参考に DMA ドライバを作成した. DMA はシンプルモードによる転送を行う. また, Android OS では `/dev/mem` へのアクセスが違反となるため, メモリアクセスのためのドライバを作成した. これは指定されたアドレスを `ioremap` し利用可能にするものである.

### 3.7. Android アプリ

前述のドライバを Android アプリから利用するために, Android NDK 及び SDK を使用し, ラッパライブラリを作成した. 処理回路のデータはアプリパッケージ (apk) に `asset` として登録した. このデータはアプリ起動時に読み出されて, ドライバを通して FPGA へ書き込まれる. 処理回路へのデータの転送やデータの回収は前述の DMA ドライバを通して行う.

## 4. 実装したアプリ

### 4.1. ビットパターンの検索

32M ビットのランダムなビット列の中から指定した 32 ビットの列が何回現れるかをカウントするものである. アプリ上での実行をそれぞれ 100 回行って平均の処理時間を測定した. この結果, FPGA を用いた方が約 180 倍高速であった (表 1).

### 4.2. アルゴンの分子動力学シミュレーション

力の計算を行う部分を専用回路によって処理した. 力の計算には式 1 を使用した. 各粒子に働く力を計算するパイプラインを 2 本実装した. この結果, 1024 粒子の場合, FPGA を用いた方が約 345 倍高速であった (表 2).

$$\frac{d}{dr}\phi(r) = 4 \left[ 12 \left( \frac{1}{r} \right)^{14} - 6 \left( \frac{1}{r} \right)^8 \right] \cdots \text{式 1}$$

表 1 ビットパターン検索 実行結果の比較

	CPU	FPGA
処理時間(ms)	2528.15	14.08
相対時間	179.56	1

表 2 アルゴンシミュレーション 実行結果の比較

	CPU	FPGA
処理時間(ms)	1968.180	5.701
相対時間	345.23	1

### 4.3. 外部 IO 操作

デバイスの外部 IO の出力をアプリから回路を通して操作する. 出力先には LED を取り付けて視覚的にわかりやすくした. 実行結果として, 実際にアプリから外部への出力が変わることが確認できた.

## 5. 課題

本研究で開発したデバイスは荒削りの部分が多い. Android OS の移植が不完全であるため, 利用できる機能に制限がある点や, カーネルにおいて無線化を実現したために, OS から無線 LAN の設定ができないなどの問題がある. また, 使用したモバイルバッテリーは通電していない状態でも回路に電流が流れ, 電力を消費してしまう問題がある. 現状では, アプリから回路へのデータ転送にはメモリ内でのデータコピーが発生してしまうため, 性能の低下につながっており, アプリから直接メモリを操作するための解決策が必要である. アプリから使用する回路の設計はアプリ開発者が行う必要があり, これには HDL による記述が必須であるため, 開発者の負担が大きい. このためこの負担を軽減する解決策が期待される.

### 参考文献

- [1] 尾崎亮, 上嶋明, 小畑正貴, "PC-FPGA 複合クラスタにおける部分再構成とその応用", CPSY, コンピュータシステム 111(398), 2012-01-18
- [2] 小池恵介, 太田涼, 大島浩太, 藤波香織, 郡信幸, 竹本正志, 中條拓伯, "Android における Java アプリケーションの FPGA アクセラレーション", 情報処理学会論文誌, Vol.53, No.12, 2740-2751, Dec. 2012.
- [3] J. P. Delahaye, G. Gogniat, C. Roland, and P. Bomel, "Software radio and dynamic reconfiguration on a DSP/FPGA platform", J. Frequenz, vol.58, no.5-6, pp.152-159, 2004.
- [4] Zynq 7000 Partial Reconfiguration Reference Design, <http://www.wiki.xilinx.com/Zynq+7000+Partial+Reconfiguration+Reference+Design>
- [5] Zynq Android 4.1 (cambridgehackers), <https://github.com/cambridgehackers/zynq-android4/wiki>