

SoC 設計プラットフォームを用いたプロセッサ開発環境の評価
 1220079 四之宮 直輝 (集積システム研究室)
 (指導教員 密山 幸男 教授)

1. はじめに

チップにする集積トランジスタの数が増加するとともに、消費電力も増加している。しかし、スマートフォンや IoT 端末などの電力資源が限られた環境では、画像認識や音声認識など高度な処理能力と電力効率の向上の両立が求められている。本研究では、低消費電力と高い性能の両立を実現するアクセラレータ搭載高効率プロセッサを開発するため、プロセッサ開発環境について検討・評価する。さらに、コアプロセッサに RISC-V[1]を使用し、SoC 設計プラットフォームに ESP (Embedded Scalable Platforms)[2]を用いて設計した SoC の使用リソースと処理時間の評価を行う。

2. RISC-V

RISC-V はカリフォルニア大学バークレー校で開発された RISC (Reduced Instruction Set Computer) に基づく命令セットアーキテクチャ (ISA) である。RISC-V は基本命令と拡張命令をもっている。基本命令には、算術命令、論理演算命令を含んでいる。レジスタのビット長は 32、64、128 ビットから選択できる。拡張命令は、乗除算命令(M)、アトミック命令(A)、16bit 短縮命令(C)など 13 種類ある。必要な拡張命令を追加することで、高い性能を維持したまま、実装規模を小さくすることが可能である。

3. SoC 設計プラットフォーム

HERO[3]、Chipyard[4]、ESP、Tapasco[5]の 4 つの SoC 設計プラットフォームについて調査した。表 1 に各プラットフォームの特徴を示す。表 1 から本研究では ESP を採用する。

表 1 SoC 設計プラットフォームの比較

	プロセッサコア	アクセラレータ記述	通信方法	ドキュメント
HERO	ホストプロセッサ	VHDL, Verilog HDL	バス接続	△
Chipyard	Rocket/BOOM Ariane	Chisel	TileLink (AXI)	○
ESP	LEON3 Ariane/Ibex	VHDL, Chisel C/C++	NoC	○
Tapasco	ホストプロセッサ	VHDL, Chisel, C/C++	バス接続	△

4. ESP を用いた SoC 設計

ESP はコロンビア大学が開発したヘテロジニアス SoC 設計プラットフォームであり、プロセッサ、アクセラレータ、メモリなどを搭載する SoC のシステムレベルの設計が可能である。プロセッサコアには RISC-V の Ariane が使用できる。アクセラレータの設計は C/C++や SystemC による高位合成のほか、SystemVerilog や VHDL による RTL 設計が可能である。図 1 に ESP の設計フローを示す。

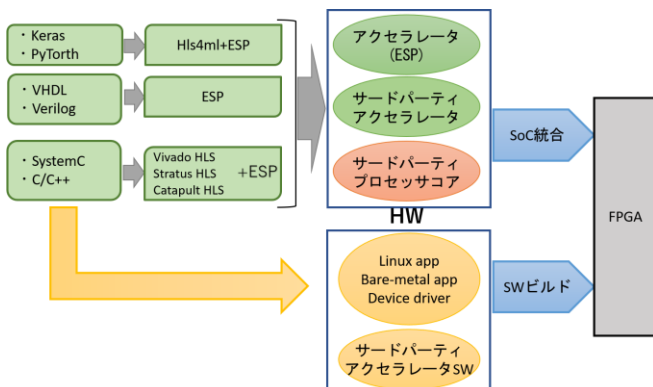


図 1 ESP を用いた SoC 設計フロー

ESP を用いた SoC 設計では、各モジュールをタイルとして配置する。タイルには、プロセッサ、アクセラレータ、メモリインタフェース、I/O の 4 種類がある[2]。ESP を用いたアクセラレータタイルの設計はインタフェース仕様の設定、アクセラレータの動作記述、高位合成の順で行う。インタフェース仕様の設定とアクセラレータの記述は ESP 上で行う。インタフェース仕様として、データビット幅やバッチ処理係数などを設定する。

5. 合成結果と処理結果

ターゲットボードを Xilinx 製 VC707 として SoC 合成を行った。アクセラレータは 32 ビット 64 オペランドの積和演算器を C 言語を用いて記述した。アクセラレータの設計完了後に、シングルコア SoC とアクセラレータ搭載シングルコア SoC、アクセラレータ搭載デュアルコア SoC の合成と RTL シミュレーションを行った。アクセラレータ非搭載 SoC と搭載した SoC の使用リソース数の比較を表 2 に、積和演算を 100 回行った演算時間と全処理時間を表 3 に示す。

表 2 使用したリソース数

	LUT	FF	BRAM	DSP
SingleCoreSoC	83,179	83,906	68	27
ACC 搭載 SingleCoreSoC	129,191	102,580	85	48
ACC 搭載 DualCoreSoC	230,737	180,213	248	75

表 3 処理時間と演算時間

	演算時間[ms]	全処理時間[ms]
SingleCoreSoC	5.9	16.2
ACC 搭載 SingleCoreSoC	0.27	10.6
ACC 搭載 DualCoreSoC	0.27	13.4

表 2 から、アクセラレータを搭載した場合、使用 LUT 数は約 46,000 個増加した。DSP は約 2 倍、LUT は 1.5 倍、FF・BRAM については約 1.2 倍となった。また、アクセラレータ搭載 SoC をデュアルコアにした場合、BRAM が約 3 倍増加した。表 3 から、アクセラレータを搭載することで、処理時間と演算時間も短縮した。しかし、アクセラレータ搭載デュアルコアプロセッサでは、逆に処理時間が長くなった。

6. まとめ

SoC 設計プラットフォーム ESP を用いてアクセラレータ搭載プロセッサを設計した。シングルコア SoC にアクセラレータを搭載することで、リソースは約 1.5~2 倍まで増加したが、処理時間を約 35%短縮した。また、デュアルコア SoC にアクセラレータを搭載することで、リソースは約 2.2~3 倍、BRAM は約 4 倍に増加したが、処理時間の削減は約 18%となり、期待より短縮されなかった。この要因は、2 つあるコアを適切に使用できていなかったことが考えられる。

参考文献

[1] <https://riscv.org/>
 [2] P. Mantovani, et al. "Agile SoC Development with Open ESP," in Proc. International Conference on Computer Aided Design (ICCAD), Sept. 2020
 [3] A. Kurth, et al. "HERO: Heterogeneous embedded research platform for exploring RISC-V manycore accelerators on FPGA". *arXiv preprint arXiv:1712.06497*. Dec 2017
 [4] A. Amid, et al. "Chipyard: Integrated design, simulation, and implementation framework for custom socs," IEEE Micro, vol40, no 4, pp10-21, Jul.2020
 [5] W. J. Dally, B. Towles, "Route packets, not wires: on-chip interconnection networks," In Proceedings of the 38th annual design automation conference, pp 684-689, Jun. 2001