

汎用高位合成系を用いた 64 ビット RISC-V 機械語からのバイナリ合成

Binary Synthesis from 64bit RISC-V Executable Codes

Using General-Purpose High-Level Synthesizer

由良 駿
Suguru Yura

石浦 菜岐佐
Nagisa Ishiura

関西学院大学 工学部
School of Engineering, Kwansai Gakuin University

1 はじめに

バイナリ合成 [1] は機械語プログラムからレジスタ転送レベルのハードウェア設計を生成する技術であり、一部または全部がアセンブリやインラインアセンブリで書かれたプログラムからもハードウェア合成が可能である。文献 [2] は汎用高位合成系を利用したバイナリ合成の容易な実装手法を提案し、32 ビットの RISC-V に対する処理系を実装している。本研究では、64 ビット RISC-V 用のバイナリ合成系を実装する。

2 汎用高位合成系を用いたバイナリ合成

文献 [2] のバイナリ合成法は、機械語プログラムを等価な C プログラムに変換し、これを汎用の高位合成系でハードウェア化するというものである。機械語プログラムを逆アセンブルし、得られたアセンブリを C プログラムに変換する。アセンブリコードから C プログラムへの変換処理系を実装するだけで容易にバイナリ合成系を開発できる。32 ビットの RISC-V であれば、レジスタは 32 ビットの符号なし整数として宣言し、例えば “addi a3, a4, 12” という命令であれば “a3 = a4 + 12;” という C の文に変換する。

3 64 ビットの RISC-V に対応したバイナリ合成系

本稿では、32 ビット用の処理系を拡張することにより 64 ビットの命令セットに対応させる。アセンブリプログラムの高位合成可能な C プログラムへの変換例を図 1 に示す。汎用レジスタは 64 ビットの符号なし整数 (uint64_t 型) として宣言し、64 ビットの加減算やシフト命令は uint64_t 型の演算に変換する。64 ビット型整数乗算の下位 64 ビットを結果とする mul 命令は、uint64_t 型の乗算で実現できる。64 ビット符号付き整数の乗算の上位 64 ビットを計算する mulh 命令は、レジスタに格納されている値を 128 ビットの符号付き整数型に変換してから乗算を行い、上位 64 ビットをシフト演算により抽出する C の文に変換する。図中の SINT64 は、64 ビットの符号なし整数を符号付き整数に変換するマクロである。addw 命令等のように汎用レジスタの下位 32 ビットに対する演算命令は、演算結果を uint32_t にキャストして下位 32 ビットだけを残し、符号付き演算の場合にはその結果を 64 ビットに符号拡張する C の文に変換する。

4 実験結果

実装したバイナリ合成系により Xilinx Artix-7 をターゲットに、いくつかのプログラムに対してハードウェア

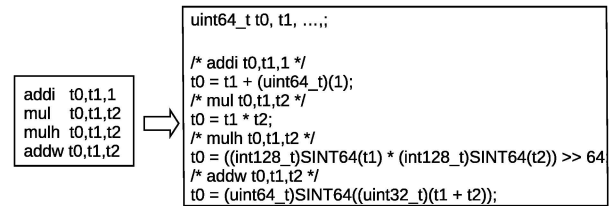


図 1: プログラム変換例

生成を行った。結果を表 1 に示す。高位合成系には Xilinx Vitis を使用した。“insertion_sort” は挿入ソート、“linear_search” は線形探索、“heap_sort” はヒープソート、“lcm” は最小公倍数の計算であり、いずれも 64 ビットデータを扱うものである。機械語プログラムはプログラムから GCC により生成した。“32 ビット版”と“64 ビット版”は、それぞれ RV32IM と RV64IM の命令セットにコンパイルした機械語からの合成結果である。“LUT” は回路規模、“Cycle” は実行サイクル数を示す。いずれのプログラムにおいても、回路規模、サイクル数は 64 ビット版の方が小さい値となっている。

表 1: 合成結果

	32 ビット版 [2]		64 ビット版	
	LUT	Cycle	LUT	Cycle
insertion_sort	1,362	8,246	504	3,752
linear_search	144	465	126	472
heap_sort	2,198	21,292	1,115	13,897
lcm	84,494	874	7,397	455

High-Level Synthesizer: Xilinx Vitis (2022.2)
Target: Xilinx Artix-7 (xc7a100tcsq324-1)

5 むすび

本稿では、文献 [2] の方法に基づき 64 ビット RISC-V 機械語からのバイナリ合成系を実装した。今後の課題としては浮動小数点演算命令セットへの拡張が挙げられる。謝辞 本研究は一部 JSPS 科研費 24K14885 の助成による。

参考文献

- [1] G. Stitt and F. Vahid: “Binary synthesis,” *ACM TO-DAES*, vol. 12, pp. 1–30 (Aug. 2007).
- [2] R. Nakamichi, S. Kishimoto, N. Ishiura, and T. Kondo: “Binary Synthesis Using High-Level Synthesizer as its Back-End,” in *Proc. SASIMI 2022*, pp. 121–126 (Oct. 2022).