

ハードワイヤード制御論理のための除算器の設計と評価

指導教員：市川 周一

学籍番号：113708 宇山 和輝

1 はじめに

Programmable Logic Controller (PLC) は産業機械に用いられる制御用コントローラである。内部の制御論理を書き換えることにより生産現場で制御内容を変更できるが、大規模な制御では処理速度が不足することがある。これに対し市川 [1] らは高速化の観点から PLC 命令列を論理回路に変換するツールを作成し、FPGA を用いて評価した。FPGA は単一チップに数十〜数百万ゲートを集積でき、小面積で大規模な制御を実装できる。

市川 [1] らの作成したツールにおいて、四則演算は VHDL の算術演算パッケージで実装されている。これは処理の高速化の観点で有効であるが、回路の小型化や面積性能比 (AT 積) の低減に関しては未だ検討の余地がある。

2 除算器の設計

本研究では除算器の小型化、AT 積の改善を目的とする。AT 積は演算に要する面積 (Area) と時間 (Time) の積であり、単位性能あたりのコストを意味する。Altera [2] 社の提供する算術演算パッケージ (LPM package) を用い 32 bit 演算器を作成した場合、除算器は加減算器と比較して 42 倍の論理素子 (ALUT) を要する。一方、一般に PLC 命令列では、除算命令は他の演算命令より使用頻度が低い。従って小型除算器を用いることで除算の性能と引き換えに、回路規模の削減と低コスト化を実現できると考えられる。

除算器は、三菱電機の汎用シーケンサ MELSEC Q シリーズの除算命令の仕様に合わせて設計する。同シーケンサでは整数除算は 16 bit ないし 32 bit の符号付き演算に限定されているが、16 bit 除算は 32 bit 除算器を用いて演算可能であるため、本研究では 32 bit 除算器のみ評価する。

除算器には多くの設計方法が存在するが、本研究の目的は小型化であるため、処理が単純な回復除法と非回復除法を用いる。回復除法は被除数と除数を比較し、その差の値が負にならないよう減算、比較、シフトを繰り返すアルゴリズムである。非回復除法は一時的に被除数と除数の差が負になることを受容するが、次桁の処理で加算を行い演算結果を収束させるアルゴリズムである。非回復除法ではループ終了後に商と剰余の補正を必要とする。

回路は除算アルゴリズムの実装方法を変更し小型化できる。除算器の実装には、組合せ回路による配列型と、順序回路による減算シフト型の 2 種類がある。演算の高基数化により 1 回で 2 進数 2 桁以上の計算が可能であるが、1 回あたりの処理量は増加する。しかしサイクル数の減少によりレイテンシの減少、AT 積の改善が見込まれる。

3 評価

除算器の設計方法 2 通り (回復除法、非回復除法)、実装方法 2 通り (組合せ回路、順序回路) で予備評価を行った。一般的な回路設計において利用される LPM package とも比較する。

評価環境には論理合成ソフトウェアとして Altera 社 Quartus II 12.0 (64 bit Full Edition) を用いた。評価デバイスは Cyclone V, 5CG7FD7C5F23C6 を対象とし、ハードウェア記述言語は VHDL を用いた。論理合成には Intel Core i7-860, 8 GB RAM, Windows 7 64 bit を搭載した PC を用いた。

作成した除算器のレイテンシと ALUT について図 1 にまとめる。図 1 において Comb. は組合せ回路、Seq. は順序回路、Rest. は回復除法、NonRest. は非回復除法を意味する。また LPM. は LPM Package, r4 は基数 4, r16 は基数 16 を意味する。

論理回路においては一般に順序回路の方が小型かつ低速であり、本研究でも同様の結果が得られた。Comb. LPM. と比較して Seq. Rest. は 68.9%、Seq. NonRest. は 62.5% 小型である。回復除法と非回復除法を比較した場合、組合せ回路では非回復除法が 40.8% 小型であるが、順序回路では回復除法が 17.1% 小型であった。Seq. NonRest. の商と剰余の補正を省略すると、Seq.

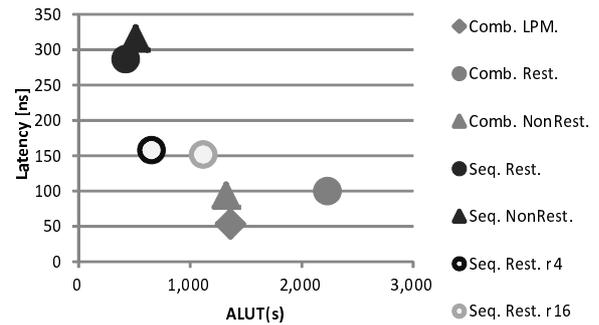


図 1: レイテンシ [ns] と論理規模 [ALUT]

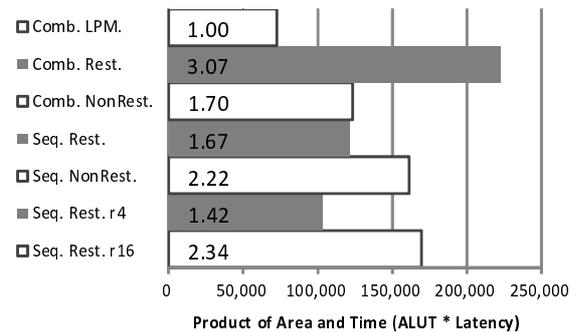


図 2: 各設計の AT 積

Rest. より 31.6% 小型になった。従って非回復除法による設計が回復除法より大型になる原因は、商と剰余の補正回路にあると考えられる。基数を変更した場合、基数の増加とともに回路規模も増加し、Seq. Rest. と比較して、Seq. Rest. r4 は 1.55 倍、Seq. Rest. r16 は 2.65 倍の回路規模になった。一方レイテンシは、Seq. Rest. r4 は 44.9%、Seq. Rest. r16 は 47.1% 減少した。

作成した除算器の AT 積の比較を図 2 に示す。全設計を比較すると、Comb. LPM. が 7.25×10^4 で最も小さい。順序回路では、Seq. Rest. は Comb. LPM. の 1.67 倍、Seq. NonRest は 2.22 倍であった。更に基数を変更した場合、Seq. Rest. r4 は 1.42 倍、Seq. Rest. r16 は 2.34 倍であった。よって今回設計した除算器では Seq. Rest. や Seq. Rest. r4 は小型かつ AT 積の小さい除算器である。これに Comb. LPM. を加えた除算器の中から、要求される小型化の程度や許容されるレイテンシを考慮し設計を選択することで、AT 積を大きく損なうことなく小型化が実現できる。

4 おわりに

除算器の設計、実装を変更し論理回路の小型化を実現した。除算器の回路規模とレイテンシはトレードオフの関係にあり、回路が小型化するにつれ実行時間、AT 積の増加が見られた。ただしデバイスや論理合成オプションによって生成される除算器が異なるため、一概に設計方法の優劣は結論づけられない。従ってそれぞれの応用システムで最適な設計を選択する必要がある。

参考文献

- [1] Ichikawa, S., Akinaka, M., Hata, H., Ikeda, R. and Yamamoto, H.: An FPGA Implementation of Hard-Wired Sequence Control System Based On PLC Software, *IEEJ Trans.*, Vol. 6, pp. 367 – 375 (2011).
- [2] Altera, : <http://www.altera.co.jp/>.