

3x + 1 予想検証回路の設計と評価

指導教官：市川周一

学籍番号：023710 小林尚大

1 はじめに

L. Collatz は、任意の自然数 n に対し、式 (1) の操作を繰り返すと必ず 1 になるという予想をたてた ($3x + 1$ 予想)。

$$H(n) = \begin{cases} 3n + 1, & \text{if } n \equiv 1 \pmod{2}, \\ n/2, & \text{if } n \equiv 0 \pmod{2}. \end{cases} \quad (1)$$

しかし、 $3x + 1$ 予想は未解決である [1]。そのためソフトウェアによる検証や、反例の探索が行われている。現在までに $n = 3 \times 2^{53}$ まで予想の成立が確認されている [2]。

本研究では Field Programmable Gate Array (FPGA) を用いて、 $3x + 1$ 予想を高速で計算させるための専用回路の設計と評価を行う。

2 3x+1 予想のアルゴリズム

k 以下で $3x + 1$ 予想が確認されているとき、 $n = k + 1$ の計算の途中結果が n より小さくなれば、1 まで計算を続けなくとも n について $3x + 1$ 予想が成り立つことがわかる。この他、 $3x + 1$ 予想を効率よく解くアルゴリズムが多く提案されている [3]。

- 入力値 n が偶数の場合はその値を無視する。
偶数は $n/2$ を行うので、入力値より小さくなる。
- 入力値 n が $n \pmod{4} = 1$ の場合はその値を無視する。
2 ステップ後に入力値より小さくなる。
- 計算値が偶数の場合、値が奇数になるまで右シフトする (make_odd)。
- 下位 n bit の値に応じて、複数ステップ分の計算を行う (n -bit composite polynomial)。2-bit composite polynomial では式 (2) のようになる。

$$f(n) = \begin{cases} 3\lfloor \frac{n}{4} \rfloor + 1, & \text{if } n \equiv 1 \pmod{4}, \\ 3\lfloor \frac{n}{4} \rfloor + 2, & \text{if } n \equiv 2 \pmod{4}, \\ 9\lfloor \frac{n}{4} \rfloor + 8, & \text{if } n \equiv 3 \pmod{4}, \\ \text{make_odd}(n), & \text{if } n \equiv 0 \pmod{4}. \end{cases} \quad (2)$$

C 言語でこれらのアルゴリズム (composite polynomial は 4 bit とする) を実装し実行時間を測定した。測定環境は Vine Linux 2.6r1, Athlon XP 2600+, コンパイラは gcc 2.95.3 でオプションは -O である。入力値の範囲を $3 \leq n < 2^{20}$ とし、1000 回の平均実行時間を測定すると 5.86×10^{-2} (sec.) であった。

3 設計

提案回路 1 式 (1) を実装した回路。演算幅は 140 bit とした。

提案回路 2 2 節で説明したアルゴリズム (composite polynomial は 2 bit とする) を実装した回路。また、シフタは LPM (Altera のモジュールジェネレータ) を使用した。

提案回路 3 提案回路 2 に次のような工夫を施した回路 (図 1)。make_odd を行うためのエンコーダとシフタのビット幅を 140 bit から 8 bit に削減した。式 (2) より、 $3\lfloor \frac{n}{4} \rfloor + 2 = (2\lfloor \frac{n}{4} \rfloor + \lfloor \frac{n}{4} \rfloor) + 2$ の計算で、cin を常に 1 にし足りない値をあらかじめ $2\lfloor \frac{n}{4} \rfloor$ に足しておくことでアダーを 1 つにした。140 bit マルチプレクサを 1 つ削減。

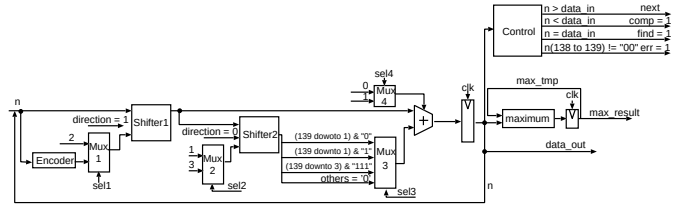


図 1: 提案回路 3

4 評価

Altera 社の Quartus II 3.0 を用いて、提案回路の Analysis & Synthesis, Fitter, Assembler, Timing Analyzer を行い、回路規模 (LE 数) と動作周波数を見積もった。デバイスは Stratix の EP1S10F780C7ES とした。各設計のサイクルシミュレータで計算終了までのサイクル数を計り、動作周波数とサイクル数から実行時間を見積もった。入力値の範囲を $3 \leq n < 2^{20}$ としたときの評価結果を表 1 にまとめる。

提案回路 1 はサイクル数が多いため実行時間が最も遅い。提案回路 2 は提案回路 1 に比べサイクル数は約 75% 減少する。しかし、回路規模は 2 倍増加し、動作周波数は 70% 下がる。提案回路 3 は提案回路 1 に比べ回路規模は 25% 増し、動作周波数は 20% 遅いが、サイクル数は提案回路 2 と同じであるため、実行時間が最も速い。また AT 積も最も小さい。AT 積は実行時間と回路規模の積で値が小さいほどコストパフォーマンスがよい。

表 1: 回路評価

	回路規模	動作周波数 (MHz)	サイクル数	実行時間 (sec.)	AT 積
提案回路 1	1489	62.75	14104634	2.25×10^{-1}	3.35×10^2
提案回路 2	3287	19.24	3614256	1.88×10^{-1}	6.18×10^2
提案回路 3	1758	49.19	3614256	7.35×10^{-2}	1.29×10^2

次に提案回路 3 を実機上で動作確認した。APEX20K シリーズの EP20K200EFC484-2X 用にコンパイルを行ったところ、動作周波数は 18 MHz であった。動作確認が目的のため、マージンをとって動作周波数を 8.3 MHz にした。サイクル数、実行時間、その他のデバッグ情報がシミュレーションと一致したため、この回路は正しく動作していると考えられる。

5 おわりに

提案回路 3 でも、実行時間はソフトウェアの実行時間より 25% 長い。しかし、今回選択したデバイスの LE 数は 10570 であるため、提案回路は 5 ユニットのせることが可能である。そのためソフトウェアより速く動作させられると思われる。

今後は複数ユニットをのせた時の測定結果を調べ、また 1 ユニットのでもソフトウェアより速く動作可能な回路を設計してゆきたい。

参考文献

- [1] R. K. Guy : “Unsolved Problems in Number Theory, second ed.,” Springer (1994).
- [2] T. Oliveira e Silva : “Maximum Excursion and Stopping Time Record-Holders for the Problem: Computational Results,” Math. Comput. vol. 68, no. 225, pp. 371–384 (1999).
- [3] G. T. Leavens, M. Vermeulen : “ $3x + 1$ Search Programs,” Computers Math. Applic. vol. 24, no. 11, pp. 79–99 (1992).