

デジタルトレーニング FPGA編

飯塚研究室 B4 福島幸弥

| はじめに

- 100桁求めるアルゴリズムは実装できなかった
- 試みたことを紹介する

| Binary-Splitting 法

- $e = \sum_{n=0}^{\infty} \frac{1}{n!}$

このような級数を再帰的に二分して求めていく方法

- Pythonで仮実装したら100桁以上の精度で求められた
(参考： <https://qiita.com/kyamaz/items/0061710cdadc11e3a644>
ChatGPT)

- Verilogでの再帰の方法がわからず断念

| 代わりに用いたアルゴリズム

- $e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n$
- nが十分大きければ、eに近似できる

$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n$ の短所

- 収束が非常に遅い
 - Pythonで確かめると $n = 2^{32}$ でも10桁程度しか一致しない
2.71828182814259...
- 多倍長演算が事実上できない
 - 主に2乗で求めるため、上位の桁がどうしても必要となる
 - データ長が n に比例して増加

$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n$ の長所

- 2乗器しか必要ない
 - $n = 2^N$ とすると非常に都合がよく
 - $1 + \frac{1}{n} = 1.00 \dots 1_{(2)}$
 - n 乗の部分は2乗を N 回行えば良いと、かなり簡単に実装できる
- 計算量は増えづらい
 - 2乗の計算しか行わないため、計算量は $O(\log_2 n)$

シミュレーション波形

\buffer[0][15:0] = 0001	0001
\buffer[1][15:0] = 0080	0001 0002 0004 0008 0010 0020 0040 0080
\buffer[2][15:0] = 1FC0	0000 0001 0006 001C 0078 01F0 07E0 1FC0
\buffer[3][15:0] = 3580	0000 0004 0038 0230 1360 A2C0 3580
\buffer[4][15:0] = C7E0	0000 0001 0046 071C 8C78 B1F0 C7E0
\buffer[5][15:0] = F680	0000 0038 1110 12A0 5740 F680

- 2乗器は正しく動作した
- 10進への変換器は未実装