3x＋1問題に対するＦＰＧＡの利用

Application of FPGA to 3x+1 Problem

小島　航*　　角谷　浩享
Wataru OJIMA　　Hiroyuki KADOTANI

（平成18年９月13日受理）

FPGAは、問題に応じて内部回路構成が変更できるLSIである。本稿では、このFPGAを用いて、整数論の有名な問題、3x＋1問題を数値的に実験するための専用ボードを設計した。その結果、汎用型のCPU上でソフトウエア実験を行う場合に比べて約26倍の性能を得ることができた。

１．はじめに

3x＋1問題は、数学的な証明が得られていない整数論の問題の一つである。有名なオイラーの定理（整数nが２より大きいとき$X^n+Y^n=Z^n$を満足する整数　X, Y, Z の組は存在しない）は最近数学的に解決された。しかし、数学的な証明がなされるまでは、実験的にそのような数の組を見つける計算が試みられていた。（当然すべての試みは失敗に終わったのだが。）整数論の世界には、このような簡単に実験できるが、厳密な数学的証明が得られていない問題がある。他の有名な例に、ゴールドバッハの予想というのがある。これは、２より大きい任意の偶数は二つの素数の和で表せると言う予想である。（例えば８は３と５の和である。）この場合、大きな偶数にたいして二つの素数を実験的に見つける努力がなされる。

3x＋1問題（その内容は３節で述べるが）も同様な問題で、厳密な数学的証明は得られていないが、誰でも（小学生でも）簡単に実験して、その正しさを確かめることができる。以下に述べるようにxは、任意の整数で、xがある性質を持つことを証明すればよい。しかし、数学的な証明が得られていないため、実験的に大きなxに対して、簡単な演算（２で割る、３を掛ける、１を足す）を行い、整数xの性質を確かめる試みがなされている。

本論文では、3x＋1問題のための専用CPUをFPGAを使って設計する試みを述べる。

FPGAは、２節以下で簡単に説明するが、ある問題に対する専用CPUが設計できるチップ（LSI）である。通常CPUの構造やCPUを構成している要素の機能はあらかじめ決まっており、計算はソフトウエアにより行われる。これに対して、FPGAは計算の目的に応じて最適化されたCPUを設計する。一般的には、汎用型のCPUを用いてソフトウエアにより計算を行うより、専用のCPUを作成して計算を実行する方が高いパフォーマンスが得られる。

本稿では、3x＋1問題に対して作成したFPGAと汎用CPUを用いた場合の計算速度の比

＊　本学国際情報学部2002年度卒業生

較をおこない、FPGAの有効性を示す。

2．3x＋1問題とは[2]

3x＋1問題とは、コラッツの問題（角谷*の問題）といわれているものである。ある正の整数（x）を以下の条件で処理すると、1になる（ようである）という予想である。

xが奇数のときは、3倍して1を加える　→　(3x＋1)
xが偶数のときは、2で割る。　→　(x／2)
以上を繰り返し処理すると、1となる。

1つの例を挙げると、x＝5の場合は、

$$5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$$

となる。この問題は数学的には証明されていないと言われているが、コンピュータを用いて、ある一定の大きさの数まで、成り立つことが確かめられている。比較的最近の結果では、10×2^{58} (May 2006)[3]が報告されている。少なくとも現在まで、このような演算を繰り返して最終的に1とならない整数は、発見されていない。

本稿では、このような数値実験を効率よく行うためのボードを、FPGAを用いて作成することを試みた。

3．FPGA[1]とは

FPGA（Field Programmable Gate Array）とは再構成可能なLSIである。

FPGAはPLDの1種であり、通常SRAMベースの再構成可能で、揮発性のものをFPGA, PROMベースで不揮発性のものをPLDと呼ぶことが多い。前者は大容量、高速動作、実用上の書き換え回数に制限がないのが特徴であり、後者は不揮発、低消費電力が特徴である。FPGAの回路設計にはHDL（VHDL, Verilog HDL）言語によって回路を記述する方法が一般的である。これらはプログラミング言語ADAやPascal風の構造を持つ言語であり、ソースコードの記述で回路設計が可能である。また、近年C, C＋＋, Javaといった言語をベースに設計可能な開発環境も存在している。しかし、自由度の高いソフトフェアと違い、ハードウェアを記述するため、一定のハードウェアの知識と、それを意識した上での設計が必要である。

4．FPGAの構成について

今回2種類のFPGAボードを用意し、それぞれに計算回路を組み込んだ。
それぞれの構成について説明をする。

1つ目は、Altera社Cyclone FPGAを搭載した、プライムシステムズ製CX-USBボー

＊　蛇足であるが、本論文の著者の一人との名前の一致は単なる偶然である。

ドを使用したものである。このボードの特徴は、ホストとなるPC（Win）からUSBを使い接続でき、FPGAの構成、内部レジスタの読み出し、書き込みが簡単にできることである。このボードでは、3x＋1問題処理用回路を内部レジスタと接続し、ホストPCから計算結果の取得は、FPGA内のレジスタを監視することで計算処理をした。（図１を参照）

２つ目は、Xilinx社Virtex-II-Pro FPGAを搭載した、Xilinx製ML-310ボードを使用したものである。このボードの特徴は、PowerPCプロセッサを内蔵した高性能FPGAを搭載し、ボード上にPC等と同等な標準インターフェースを搭載していることである。シリアルポート、パラレルポート、PCIバススロット、DDR－SDRAMスロット、IDEインターフェイス等、PC標準の物を使用しており、寸法等もPCのメインボードと同様の仕様となっている。また、PowerPCプロセッサ上で、Linuxや、組み込み用途向けOSが動作するため、PCに近い装置を開発することが可能である。特に、Linuxでは、標準的なハード、ソフトが動作するため、X-WindowSystem上でGUI環境の動作も可能である。このボードでは、3x＋1問題処理用回路を内部バス（OPBバスインターフェイス）に接続し、PowerPCプロセッサから、処理回路のレジスタを監視することで計算処理をした。（図２を参照）

５．計算回路について

3x＋1問題については、２節で示した処理手順が必要であるが、ソフトウェアまたはハードウェアで処理する際は、次のように処理した。

① 初期値xを設定
② xが奇数のときは、３倍して１加える。　$x=x\times3+1$
③ xが偶数のときは、２で割る。　$x=x/2$
④ xが終了条件のとき、終了し、そうでないときは、②にもどる。

このときの初期値xは３以上であり、終了条件は$x=1$のときである。
終了条件については、計算機上で連続して検証するため、初期値xよりも小さいという条件も設定できるようにした。また、計算過程において、xの最大値と、計算に要したステップ数をカウントし、計算の正確さと、性能評価の参考とした。また、このときの処理は、

初期値は必ず奇数を取る、
３倍して１加えるときは、１ビット左シフトしたものに、値と１を加える、
２で割るときは１ビット右シフトする、

などの処理に変更した。
計算回路内部のレジスタの構成は、以下のようにした。

内部レジスタ群の構成
① 初期値xを保持するレジスタ……（R1）
② 計算に使用するレジスタ……（R2）
③ 最大数を保持するレジスタ……（R3）
④ 計算ステップ数を保持するレジスタ……（R4）
⑤ 計算回路の状態変移を設定、表示するレジスタ‥（R5）
⑥ エラー表示用レジスタ……（R6）

このときのレジスタの幅は、R1のビット幅に対してR2、R3は2倍の幅を用意した。例えばR1が32bitsのときはR2、R3は64bitsの幅を持つことになる。これまでの計算結果から、初期値の2倍の幅の数までは大きくならないことが、経験的にわかっているので、処理の都合上一定の大きさとした。当然であるが、オーバーフローの監視をしつつ、計算をすすめるため、2倍以上の幅になればエラーで停止するようにしてある。この回路をFPGA内に回路規模、または動作周波数の制限内で複数用意し、並列処理を行った。

6．比較対象と結果について

今回用意したFPGAは130nmプロセスで生産されているものである。比較対象としては、同世代のプロセスで製造された標準的なマイクロプロセッサが適当であるため、130nmプロセスのIntel Pentium 4との比較とした。130nmプロセス版のPentium 4プロセッサは1.5～3.4Ghz版まであるが、標準的なグレードの2.4Ghz版とした。比較用に、Compaq VisualFortran6.6上で計算用ソフトウェアを作成し、性能と動作の正確さを検証した。このときの処理は、初期値に対してそれ以下の数値になった時点で打ち切るようにした。また、比較対象のFPGAはCX-USBボード上のCycloneFPGAのみとした。ML-310ボード上のFPGAでは、周辺回路用の領域が大きく、前者と比べて、計算回路の数が少なくなってしまったためである。

7．比較結果と考察

$3x+1$問題の探索において、約1000億までの数では、FPGA内に計算処理回路を9個、動作周波数を72MHzとしたとき、Pentium4 2.4GHz上のソフトウェア処理と比較し約26倍の処理速度がえられた。これは並列処理により、得られたものである。並列処理とは次の2点である。

① 計算処理回路を複数持つこと
② ハードウェア処理による内部処理の平行実行

1つめは、複数の計算回路で並列処理を行うということであるが、ソフトフェア処理でも同様のことは可能である。マルチプロセッサ上のマルチスレッド処理や、クラスタリングシステム上の並列処理などである。この問題に関しては、それぞれの処理は独立してお

り、処理間の通信などあまり必要ではないため、これらの並列処理でもかなりの効率で高速化が期待できる。２つめは、ソフトウェア処理とハードウェア処理の違いからくるものであり、FPGAを用いた処理の特徴でもある。ソフトウェア処理では、原則的に逐次実行処理をしており、前段の処理結果がでるまで次の処理はできない。ソフトウェア処理では、これらは逐次処理の結果を待ち、その後分岐処理等で行うため、処理時間の増加原因になる。今回の計算回路では、分岐する部分は両方実行、計算処理とは関係の薄い、最大値や計算に要するステップ数の取得、エラー処理等は、独立して平行処理するため、処理時間の増加の要因にはならない。また、ハードウェア処理であるため、各処理は１クロックサイクルで効率良く処理できるが、CPUでは分岐処理や、基本的な演算処理も64、96ビット処理など多倍長演算となるため、複数クロックサイクルを要する。（図４参照）

また、計算過程で得られた結果については、約１兆までの数についてはソフトウェア処理で得られた結果とFPGAによるハードウェア処理によって得られた結果が一致した。それ以上の数については、特徴のある数付近の計算において同様の値が観測できた。
（表１、表２、参考文献３参照）

８．まとめ

FPGAを用いた3x＋1問題において、同世代のマイクロプロセッサよりも高速処理できることを確認できた。今回使用したFPGAは標準的な容量、速度のものであり、同世代のFPGAでもより大容量、高速動作のものが存在する。これらの高性能FPGAを用いればより高速な処理も可能である。今回のFPGAのプログラミングの難易度はそれほど高くはない。この問題向けの回路の基本的な部分は、筆者がFPGAを使いはじめたころに、FPGAプログラミングの練習のために作成したものである。しかし、複雑で難易度の高い処理をFPGA上でハードウェア処理に置き換えるには、かなりの時間と知識が必要となるため困難を極めることも多い。FPGAでは、マイクロプロセッサをはじめ、ほとんどの種類のLSIの機能を実装することが可能であるため、性能や実装機能に制限はあるものの、ほとんどの処理を実装することは可能である。しかし、それらを自前で実装するのは、現実的には難しいし、非常に効率の悪い作業でもある。有償、無償、オープン、クローズド、など条件はさまざまであるが、汎用の回路は多くの種類で完成されたものが用意されている。これらを使いつつ、効率よく回路設計が可能であれば、ある程度の規模の処理までは実装できるはずである。一方でこれらの汎用回路を使用できないようなものは、規模が大きくなると実装困難になってくる。

今後の課題としては、

- 3x＋1問題において、今回のものを改良し、どの程度の数まで探索できるか。
  より実用的な処理をFPGA上で実装すること。
- 設計の難易度とFPGAに実装することで得られる利点のバランスの良い問題を探すこと。
- FPGAの再書き換え可能な特性を生かしたソフトウェアとFPGA上のハードウェアとバランス良く組み合わせて処理できるような問題を探すこと。

がさしあたり重要であると考えている。

参考文献

（1）入門書として、山際 真一著　「FPGAボードで学ぶ論理回路設計」、CQ出版、がある。
（2）例えば、http://www.tuat.ac.jp/~kotani/3xplus1.htm
（3）http://www.ieeta.pt/~tos/3x+1.html

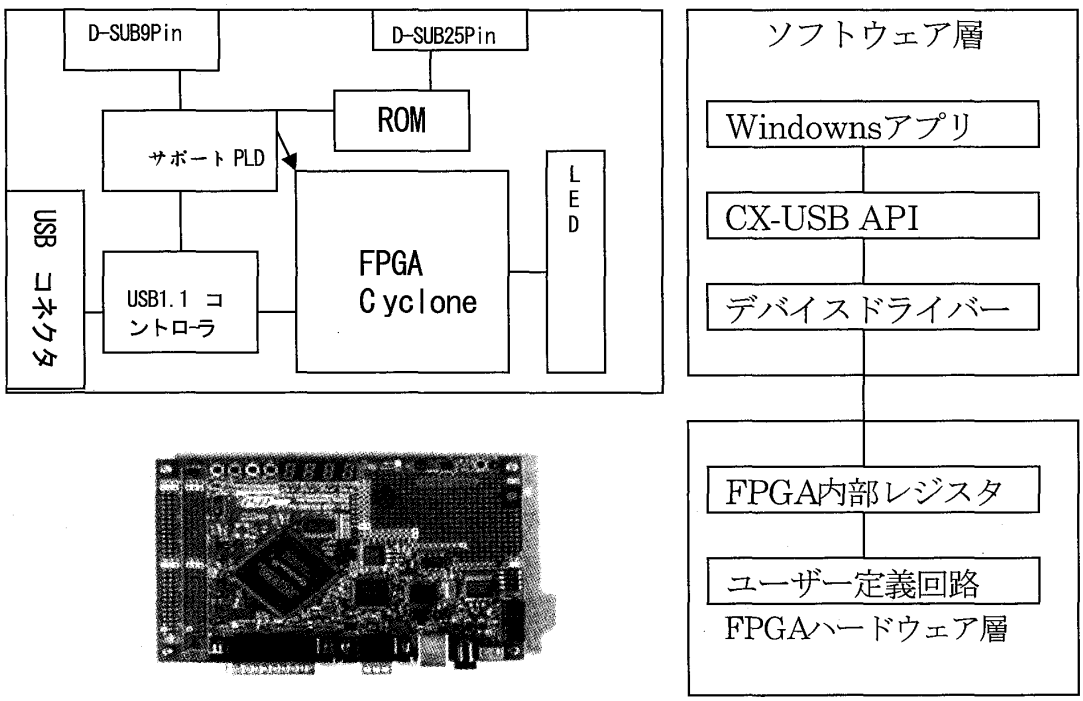

図1　CX-USBボードの主要な構成図と外観写真

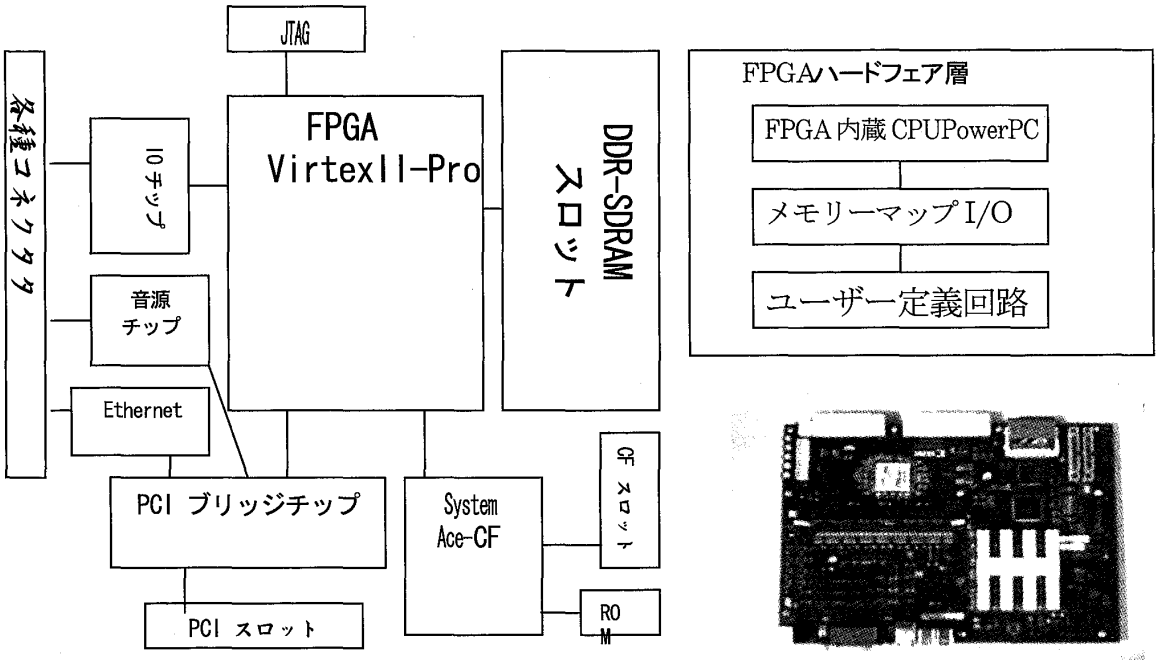

図2　ML-310ボードの主要なブロック図と外見写真

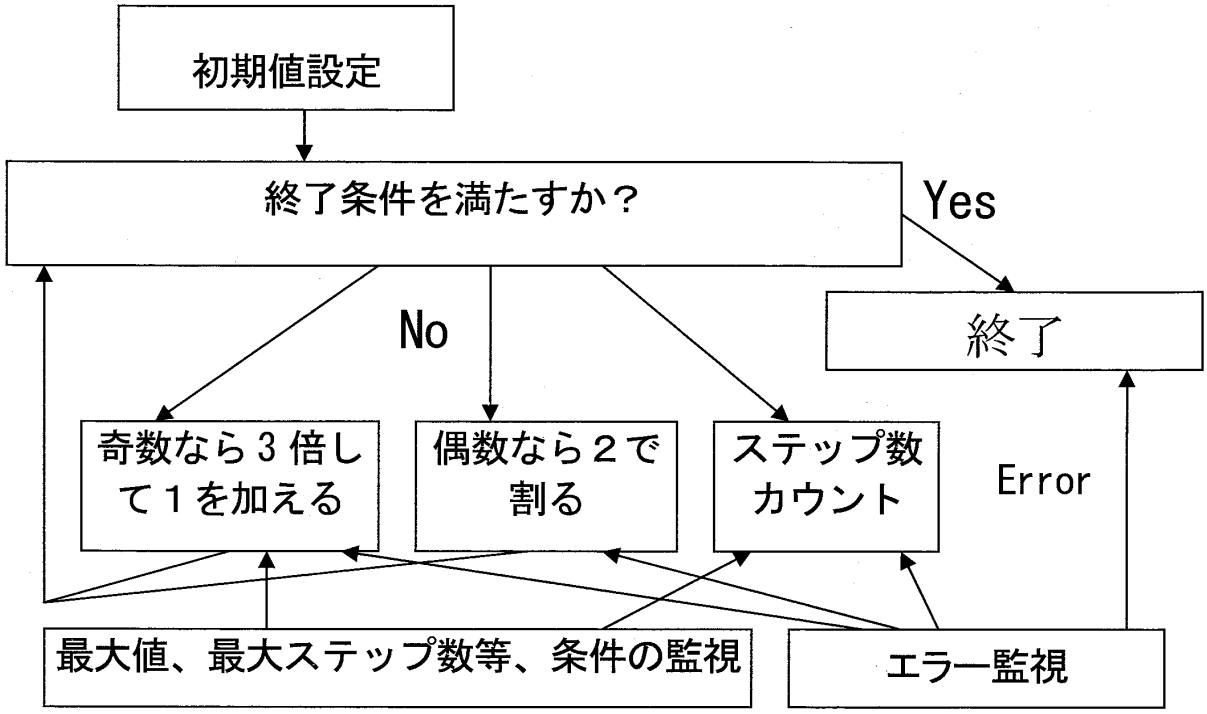

図３　FPGA内の計算処理の流れ図

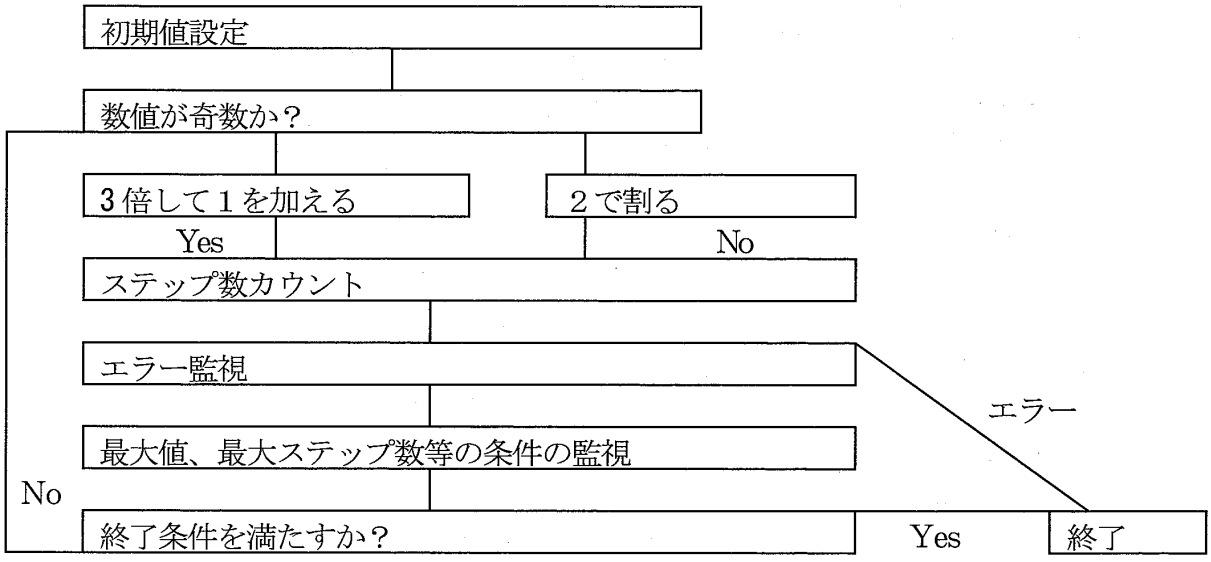

図４　ソフトウェアによる処理の流れ図

表1　最大ステップ数更新の表

数　X	ステップ数	
1008932249296230	1445	FPGAにて値を確認
739448869367967	1187	
31835572457967	1177	
13179928405231	1122	
2081751768559	988	ここまでFPGAで計算
898696369947	897	ここまでソフトウェアで計算
12235060455	892	
2788008987	729	
1827397567	706	
1200991791	649	
217740015	644	
63728127	613	
56924955	502	
26716671	486	
20638335	476	
13421671	468	
以下の値は省略		

表2　最大の数更新の表

数　X	最大の数	
1980976057694848447	64024667322193133530165877294264738020	FPGAにて値を確認
1038743969413717663	319391343969356241864419199325107352	
中間の数は省略		未計算
3716509988199	207936463344549940448754 64	ここまでFPGAで計算
2674309547647	770419949849742373052272	
871673828443	400558740821250122033728	ここまでソフトウェアで計算
567839682631	100540173225585986235988	
446559217279	39533276910778060381072	
272025660543	2194848365670417963748	
231913730799	2190343823882874513556	
204430613247	1415260793009654991088	
110243094271	1372453649566268380360	
以下の値は省略		