

C_003

FPGAを用いたN-Queens問題の解決について

On Solving the N-Queens Problem Using FPGA

中園 浩司*
Koji Nakazono

越村 三幸*
Miyuki Koshimura

藤田 博*
Hiroshi Fujita

長谷川 隆三*
Ryuzo Hasegawa

松崎 隆哲†
Takanori Matsuzaki

1 まえがき

N-Queens 問題とは、 $N \times N$ のチェス盤上に N 個のクイーンを互いに打ち合わないよう置き、というもので、単一解でよい場合には、 N の多項式時間の解法が知られている。しかし、解の総数を求める場合には、指数関数的な難度の問題となる。2004 年、電気通信大学のグループが、 $N = 24$ の解の総数を求めることに成功した [1]。これは、数 10 台の汎用パソコンで構成されるクラスタマシンを用い、C 言語による素朴なコードを並列実行させたものである。

一方、計算を高速化する目的に専用ハードウェアを用いる手法として、リコンフィギュラブルコンピューティング [2] を適用する研究が盛んになっている。実際、FPGA に代表される再構成可能なハードウェア技術の発達と、ハードウェア記述言語を受理する論理合成ツールの普及により、ソフトウェアの開発と大差ない時間で専用ハードウェアを簡単に実装できる環境が整ってきている。こうして、ソフトウェアでは手に負えない問題でもハードウェア並列性を最大限に引き出すことによって解決への道を拓く試みが容易に行えるようになった。

本稿では、バックトラック制御に基づく基本回路を示す。我々は、さしあたって現在未解決の $N = 25$ の解の総数を求めることを目指している。その現状と展望について述べる。

2 バックトラックに基づく基本回路

まず、求解のための基本デザインを示す。実際のチェス盤に対応した直観的で素朴な回路構成となっている。回路全体の概形を、 $N = 4$ の例で図 1 に示す。qq は探索行を表すレジスタで、注目行だけ 1 になる。q0~q3 は、クイーンの配置を記憶しておくレジスタで、高々 1 ビットが 1 になる。

図中の qq の値は、現在の注目行が上から 2 行目であることを示している。また、最上行の左から 2 番目のマスに既にクイーンが置かれていることが q0 の値からわかる。

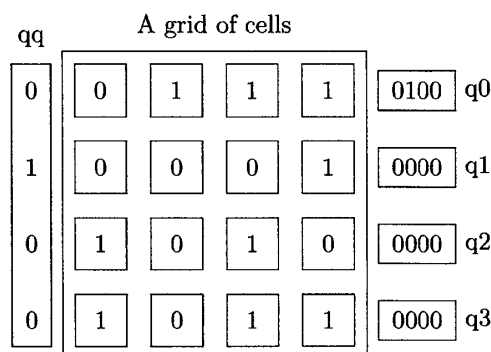


図 1: Outline of the 4-Queens circuit.

中央部 (A grid of cells) は、 4×4 のチェス盤の各マスに対応してセル (回路モジュール) を配置したものである。各セルは、効き筋を伝達するためのロジックと、バックトラックの戻り場所をマークするための 1 ビットのレジスタを含む。各セル内の数字は、基本的に 1 ならばクイーンを置くこと、0 ならば置けないことを表している。

ただし、行方向の効き筋については、“行内ではただ 1 マスにしかクイーンを置かない、” という制御に反映されており、制約伝播はしない。したがって、注目行より上の行 (図では最上行) においては、セルの中の数字の意味が異なる。すなわち、左から見たとき最初の 1 を示すセルは現在クイーンが置かれているマスに対応し、さらにその右側で 1 を示すセルは、その行内で未試行の候補であることを示す。

バックトラック動作状況を図 2 に示す。図中の $\sqrt{1}, \sqrt{2}$ は、それぞれ Q_1, Q_2 の効き筋にあたっていることを示す。また、●は、その行内ですでに置いたクイーンの代わりに置くことができる別のマスであるが、未試行であることを示す。図 2(a) で矢印が指している注目行において、クイーンがどのマスにも置けないことがわかる。そこで Q_2 までバックトラックし、図 2(b) に示すように試行済みの Q_2 をキャンセルすると同時に、残りの未試行マスの一つを選んで Q_2 を置きなおす。

*九州大学, Kyushu University

†近畿大学, Kinki University

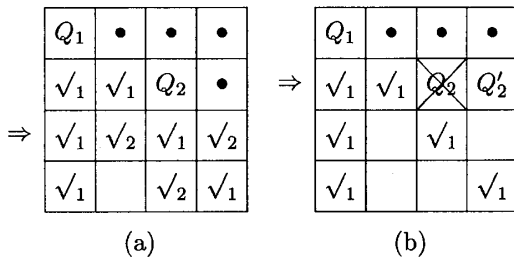


図 2: Backtracking.

表 1: Configuration results.

N	Freq.	Run time	Size	C-time
14	21.05	0.09	21782	54:45
15	20.08	0.62	23715	1:13:27
16	19.13	4.03	27849	1:21:33
17	19.26	30.01	31740	1:42:00
18	17.84	225.02	34392	2:05:35
19	16.70	1984.40	38438	2:17:27
20	15.43	16519.83	44022	3:02:36

3 実験結果と評価

ここでは、探索の一部を並列化した回路の実験結果について述べる。実験に使用した FPGA は、ALTERA 社 APEX20KE, EP20K1500EBC652-3 を 1 個である。回路合成ツールは、同社の Quartus5.0 を用いた。

$N = 20$ までの実行結果を表 1 に示す。すべて並列度は 10, マスタの行う初期配置は $(N+1)/3$ 行分とした。表の項目 Freq. は合成ツールの算定による最大動作周波数 (MHz), Run time は最大動作周波数と Clock cycles から算定した理論実行時間 (秒), Size はここで用いた FPGA の最小論理要素を単位とした回路量, C-time は論理合成時間 (時:分:秒) である。

また、台数効果について調べるために、 $N = 10$ の場合について並列度を変えて実験を行った。この結果を表 2 に示す。P は並列度である。

N が 1 増える毎にクロック数が約 10 倍となっている。 N の増大とともに最大動作周波数は下がり、論理要素数も増加するが、 $N = 20$ までは、1 チップへの実装の許容範囲内である。 $N = 20$ で最大動作周波数で動作させれば、約 4 時間半で解が求まる。

並列数が 20 となるまでは、並列数に反比例してクロック数が減少している。20 から 40 の間では、クロック数はそれほど変化していない。

4 むすび

探索の一部を並列化した回路の最大動作周波数は、 $N = 20$ の実測値 15MHz をもとに推定すると、 $N = 25$ では 10MHz 程度に低下するものと考えられる。 $N = 20$ で

表 2: Parallelization results.

P	Cycles	Freq.	Size	C-time
2	12409	29.13	3079	6:06
4	6315	29.11	5382	10:28
6	4260	28.77	7742	15:11
8	3249	27.71	10033	21:53
10	2643	26.05	12348	28:50
20	1712	16.19	22926	1:24:38
40	1669	9.80	45513	4:22:23

10MHz の動作時間は、およそ 7 時間である。 N が 1 だけ大きくなる度に動作時間が 10 倍になると仮定すると、 $N = 25$ ではおよそ 80 年となる。現在の記録である $N=24$ の求解の追試ですら 8 年もの時間が必要となる。

これを 1ヶ月程度で解決できるようにするためには、 $N = 24$ で 1000 並列、 $N = 25$ では 10000 並列必要になる。

$N = 24$ の世界記録を達成したクラスタマシンでは、ハイパースレッティング対応のプロセッサを 68 個使っているので、論理的には 136 個のプロセッサで実行していることになる。現在使用しているボードでも、100 個強で 1000 並列は実現可能であるので、 $N = 24$ を解決することは不可能ではない。しかし、 $N = 25$ の場合にはボードが 1000 個も必要となってしまうため、これは現実的でない。むしろ、周波数を上げることを第一に考えるべきだろう。クラスタマシンの周波数が 2.8GHz であるのに対し、現在の回路の周波数は 10MHz 程度である。これが、仮に 100MHz で動作すると仮定すれば 1000 並列で解決することが可能となる。300MHz を達成するボードも現れてきており、期待できそうである。

現在、最大動作周波数の向上を目的とし、パイプライン方式で動作する回路を製作している。この回路は、これまで述べた盤面を回路化したものではなく、探索木を回路化したものである。本報告で主たる成果とした探索の一部並列化方式に比べ、問題規模に対するスケーラビリティで優り、動作周波数の向上と並列度の大幅な増大を見込めるものと考えている。

参考文献

- [1] 吉瀬 謙二, 片桐 孝洋, 本多 弘樹, 弓場 敏嗣, “PC クラスタを用いた N-Queens 問題の求解,” 電子情報通信学会論文誌, Vol.J87-D-I, No.12, pp.1145-1148, 2004.
- [2] K. Compton and S. Hauck, “Reconfigurable Computing: A Survey of Systems and Software,” *ACM Computing Surveys*, Vol.34, No.2. pp.171-210, 2002.
- [3] 梅永 明寛, “ハードウェアとソフトウェアによる問題解決の比較 —8 王妃問題の場合—,” 九州大学工学部電気情報工学科卒業論文, 2003.