

FPGA による大規模 CFD の実現へ向けた取り組みとその課題

Problems and strategies towards large-scale CFD acceleration on FPGAs

- 長名 保範, 成蹊大, 東京都武蔵野市吉祥寺北町 3-3-1, E-mail: osana@st.seikei.ac.jp
森下 博和, 田舎片 健太, 天野 英晴, 慶應義塾大, 神奈川県横浜市港北区日吉 3-14-1,
藤田 直行, JAXA, 東京都調布市深大寺東町 7-44-1

Yasunori Osana, Seikei University, 3-3-1 Kichijoji-Kitamachi, Musashino-shi, Tokyo Japan.

Hirokazu Morishita, Kenta Inakagata, Hideharu Amano,

Keio University, 3-14-1, Kouhoku-ku, Yokohama Japan.

Naoyuki Fujita, JAXA, 7-44-1 Jindaiji-Higashimachi, Chofu, Tokyo Japan.

FPGA is expected to be a hardware acceleration platform for CFD, because of its flexibility at hardware level. Although FPGA is a versatile hardware accelerator, it has some resource restrictions that requires consideration in system (board) design. This paper focuses on FPGA's logic and I/O resource restrictions in our FPGA CFD implementation.

1. はじめに

必要に応じて回路を再構成することのできる大規模集積回路 (VLSI) である FPGA は半導体製造技術の進歩とともに回路容量の大容量化・動作周波数の向上や、メモリブロック・乗算ブロックなどの機能ブロックの組み込みが行われてきた。数値計算への応用は比較的早く、Spash-2¹などの取り組みは1990年代前半に行われているが、回路面積や動作周波数の制約から、整数演算が中心であった。しかし、FPGAの性能向上に伴い、浮動小数点演算パイプラインをFPGA上に実装することが現実的になり、2000年代に入ってからFPGAをアクセラレータとして科学技術計算に用いる研究が盛んになっており、PROGRAPE-3²などが成功を取っている。

筆者らのグループの一部メンバーは以前よりFPGAを用いて生化学系の反応シミュレーションを高速に行うプロジェクト³を進めており、ここで得た知見を生かして計算流体力学ソフトウェアパッケージであるUPACSのFPGAによる高速化に関する検討・実装を行っている。FPGAでのCFD実装事例ではハードウェア規模に配慮して、演算精度などの諸条件に制約を与える場合が多いが、本研究では極力UPACSのFORTRANコードに忠実な実装を行い、ユーザがいままで使ってきたソフトウェアによる実装と同じ結果が得られるようなシステムを目指している。現在までに一部のメモリアクセス機構⁴や、主要な関数⁵をHDLで開発しており、既存の基板への実装を目指すとともに、これらのHDL記述から得られたハードウェア量などをもとにした、システム全体の構成に関する研究を進めている段階である。

UPACSをFPGAで動作させるためには、多数のFPGAを用いたシステムの構成になるが、この場合、I/Oやメモリと計算能力の配置・接続が実効性能を大きく左右し、ただ単純にFPGAやメモリを並べただけのシステム構成では十分な性能を発揮できないものと考えられる。本稿では、いくつかの商用FPGAのデータといままでに実装を終えたモジュールの諸元から、UPACSを動作させるためのシステム構成のありかたについて検討を行う。

2. FPGAを用いた大規模システムの事例

現在、UPACSのFPGA版を動作させるハードウェア基盤としてFLOPS-2Dボードを開発中であるが、これはCFDを実行するための理想形というよりも、実機での動作確認を行って問題点を明らかにするためのプロトタイプとしての意味合いが強い。

ここではFPGAを用いた大規模システムの例としてImperial College Londonで開発されている512個のFPGAを搭載したFPGAクラスタであるCube⁶とUniversity of California at BerkeleyやMicrosoft研究所が共同で開発したスケラブルなFPGAクラスタであるBEE3⁷の概観を説明する。

2.1 Cube

Cube⁶は、Xilinx社のSPARTAN-3 FPGA (XC3S4000)を用いたシステムで、図1に示すような構成になっている。各基板には16個のFPGAが1次元接続されたものが4つ載っており、その基板が8枚、やはり1次元接続されている。基板にはメモリを持たないため、シストリックアレイのような、連続してデータが流れる実装方法に適しており、暗号解読や文字列検索などのアプリケーションが開発されている。

FPGA間やボード間の接続はすべて64bit, 100MHz (6.4Gbps)の平行接続である。

2.2 BEE3

BEE3⁷の構成を図2に示す。BEE3ではXilinx社のVirtex-5 FPGAを用いており、ピン互換性のある数種類のチップを搭載可能になっている。1枚の基板には4つのFPGAが搭載されており、隣接するFPGA間を64bit, 1GHz (500MHz DDR)で接続している。また、各FPGAは高速シリアルトランシーバを用いた10Gbpsのインタフェースを2チャンネルずつ持っており、これを用いてボード間を接続することで、多数のボードを結合することができる。ボード上にはDIMMスロットも備えられており、FPGAごとにDDR-SDRAMインタフェースが2チャンネル用意されている。

BEE3はFPGA間やボード間の接続が高速であるため比較的自由度の高い設計が可能で、many core processorのエミュレーションなどに利用されている。

3. ロジックリソース・計算能力に関する検討

本稿の主眼は計算能力、メモリ、I/Oのバランスをできるだけ定量的に検討することにあるが、評価を行うには具体的なモデルが必要であるので、ここではUPACSのFORTRANソースコードを基に実装した際のデータを用いる。これらの詳細は、文献⁴と文献⁵で述べられている。

3.1 評価用データセットの定義

まず最初に、FPGAに収容可能な演算器数を把握することで、回路の実装に必要なFPGAの数量や、性能見積もりが行えるようにするための基準になるようなデータを算出しておくことにする。

文献⁵では、いくつかの制限関数を適用したMUSCL法の実装を行っており、そこで必要とされる主要な浮動小数点演算器の数は表1のようになる¹。2次の場合には加減算器:乗算器:除算器の割合がおおよそ2:1:1であるのに対して、3次の場合には2:2:1に近い割合になる。この結果を見ると、CFDの実装に必要な演算器数の割合は諸条

¹実際には比較器をいくつか含むものがあるが、ここでは省略する。

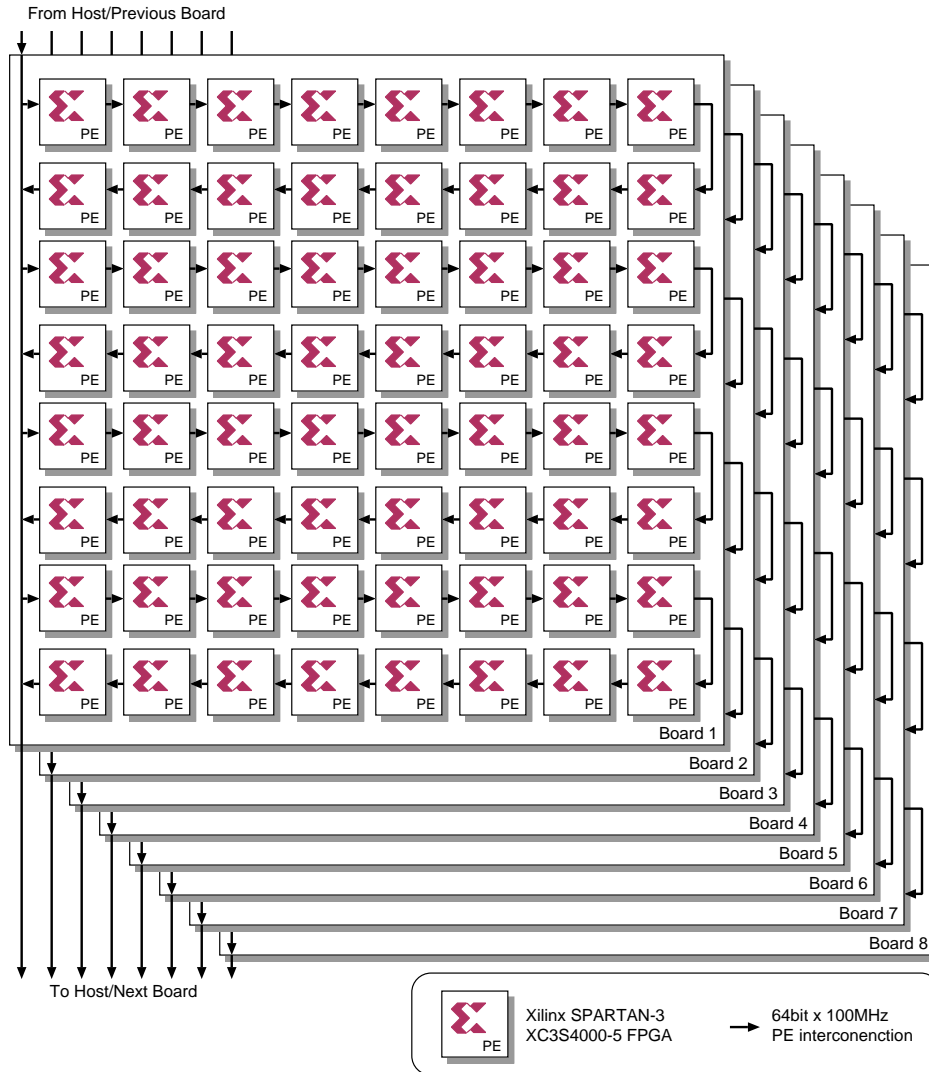


Fig. 1: Cube の構成

件によってさまざまに異なることが予想されるが、本稿では簡単な目安を提示することが目的なので、この 2 種類の概算割合を用いて、以降

- 演算器セット “2nd”: 加減算器 2, 乗算器 1, 除算器 1
- 演算器セット “3rd”: 加減算器 2, 乗算器 2, 除算器 1

Tab. 1: MUSCL 法の実装における演算器数

Order	Limiter function	Add	Mult	Div
2nd	No limiter	10	5	6
	van Leer	12	4	7
	van Alabada	13	5	6
	Min-mod	9	4	6
	Superbee	9	4	6
	Average	11	4	6
3rd	No limiter	10	8	5
	Min-mod	16	14	9
	Hemker-Koren	17	12	7
	Average	14	11	7

のふたつの演算器セットを評価用に用いることとする。なお、いずれの演算器も IEEE-754 に基づく 64bit の倍精度浮動小数点形式であり、Xilinx 社の最新版の IP コアとして提供されるものを用いている。

3.2 FPGA に収容可能な演算器数

表 2 は、Xilinx 社の各世代の FPGA 製品上に各種の浮動小数点演算器をマッピング・配置配線した結果の回路リソース使用量と動作周波数を示したものである。さらに、これを基に FPGA のスライス数および組み込み DSP(乗算) ブロックの数から算出した演算器セットの収容可能数は、表 3 のようになる。FPGA は、各製品シリーズのミッドレンジに位置するものと、最大容量のものを選択し、組み込み乗算器を十分な数持つ FPGA ではそれを使う場合、使わない場合についての結果を挙げた。この結果から、FPGA の持つ論理ブロック数や乗算ブロック数の配分などによって、どちらの演算器セットの

ほうが多くの演算器を FPGA 上に実装できるかが変化する
ことや、型番に現れるようないわゆるゲート数換算の
回路容量には演算器の数が必ずしも比例しないことがわ
かる。したがって、FPGA を用いた計算システムを構成
する場合、どのチップを選択するか、という点は非常に
重要であることがわかる。

また、参考のために、表 3 に示す演算器のパイプライン
稼働率が 100% で、全体はもっとも最大動作周波数が遅
い種類の演算器の動作周波数に合わせて動作すると仮定
した場合のピーク性能値を表 4 に示す。

4. I/O リソースに関する検討

CFD では、各格子点が圧力や密度、速度といった複
数の変数をもっており、さらに各格子点の次の時刻での状
態を計算するには数カ所の近傍点の値も必要であり、計
算を行うパイプラインには多くのデータをメモリから読
み出して投入しなければならない。

実際にこれをハードウェアで行うに当たって問題とな
るのは、

- メモリと FPGA の間の通信バンド幅に制約があり、
メモリアクセスがボトルネックとなる危険がある
- メモリアクセスのバンド幅の問題を解決したとして
も、パイプラインが複数の FPGA にまたがる場合、
FPGA 間の通信バンド幅が新たな問題となる

の 2 点である。前者は比較的定式化しやすい問題であり、
一度メモリから読み出したデータを隣接格子点の計算時
に再利用するなどの方法が有効である。これに関しては以
前から研究がなされている(文献⁸など)と共に、UPACS
をベースとした実装における効率化についても検討を
行っている⁴。

しかし、上に挙げたようにバンド幅の問題が発生する
のはメモリ-FPGA 間だけでなく、パイプラインが複数
の FPGA にまたがる場合には FPGA-FPGA 間の通信バ
ンド幅も問題となる。メモリ-FPGA 間では、メモリ
から読み出された値を近隣の格子点において再利用する
ことが比較的容易であるが、FPGA-FPGA 間ではデー
タの再利用による圧縮は困難であり、メモリ-FPGA 間
では圧縮されていたデータ量が FPGA-FPGA 間では圧縮
されない状態で転送されなければならなかったり、解く
数式の形によってはデータグロウグラフの幅が途中で拡
がり、パイプラインの途中では入力よりも多くのデー
タが流れる可能性がある、といったことが考えられる。こ
のような場合、ひとつのパイプラインが複数の FPGA に
またがる場合、FPGA-FPGA 間の転送バンド幅はメモ
リ-FPGA 間の転送バンド幅の数倍必要になると考えら
れる。

4.1 パラレル入出力

まず、FPGA の通常のユーザ I/O ピンを利用したパラ
レル入出力能力について調査を行った。

メモリとして DDR2-SDRAM SO-DIMM を 1 チャ
ネルあるいは 2 チャネル FPGA に接続すると仮定し、表 3
などに示した評価で用いた FPGA のユーザ I/O ピン数
から、残りのピンで FPGA 間の配線をどれだけ確保で
きるかを計算したのが表 5 である。この表を作成するに
あたっては、SO-DIMM は 1 チャネルあたり 130 ピン
(DDR2-SDRAM SO-DIMM は 200 ピンのモジュールで
あり、このうち約 1/3 のピンは電源ピンである)、FPGA
間通信は 1 チャネルあたり 72 ピン(データ 64bit + 制御
8bit)、その他の信号線 80 ピンと仮定して計算を行った。
また、表に挙げた FPGA のうち、Virtex-4 を除く FPGA
は高速シリアルトランシーバを装備しているが、ここで
はこれらは使わないものとしている。また、実際にはピン
配置の決定にあたっては基板のレイアウト上のさまざ
まな制約を受けるため、この表の値より小さくなる。

パラレル入出力については、64bit のパラレル信号線
を用いた場合、演算器と同クロック周波数で駆動すれば
演算器ひとつ分の出力を毎クロック転送することができ、
DDR ならばふたつ分の出力を転送することができる。
SDRAM が同じく 64bit 幅の DDR で動作することを考

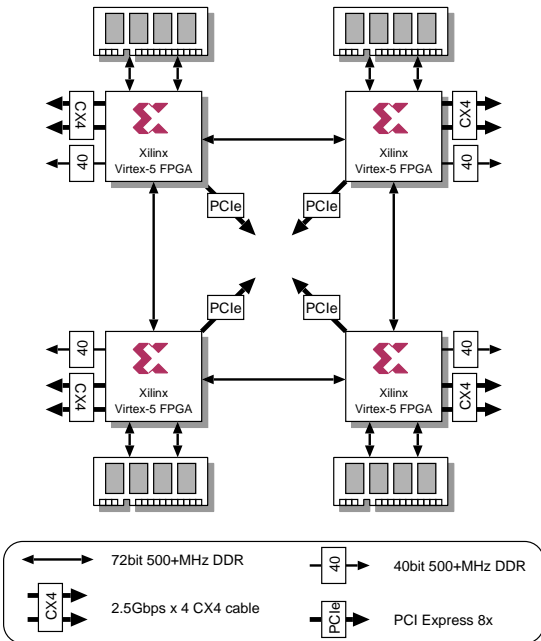


Fig. 2: BEE3 の構成

Tab. 2: 各種演算器の面積・動作周波数

	Add	Mult	Mult	Div
DSP block	なし	なし	使用	なし
VirtexII Pro-6				
Slices	968	1379	738	3689
DSPs	0	0	16	0
Frequency	227.58	145.79	150.63	135.48
Virtex4-11				
Slices	961	1379	566	3691
DSPs	0	0	17	0
Frequency	318.17	210.04	396.83	231.00
Virtex5-2				
Slices	340	693	169	1535
DSPs	0	0	11	0
Frequency	384.32	271.89	436.30	309.89
Virtex6-2				
Slices	310	663	132	12
40 DSPs	0	0	11	0
Frequency	455.58	238.89	437.83	377.93

Tab. 3: FPGA 上に実装可能な演算器セット数

VirtexII Pro-6												
	2VP70*		2VP100*									
2nd	10	(40)	15	(60)								
3rd	9	(45)	13	(65)								
Virtex4-11												
	4VLX100	4VLX200	4VSX35*	4VSX55*	4VFX60*	4VFX140*						
2nd	7	(28)	2	(8)	3	(12)	4	(16)	10	(40)		
3rd	5	(25)	10	(50)	2	(10)	3	(15)	3	(15)	5	(25)
Virtex5-2												
	5VLX110T	5VLX330T	5VLX110T*	5VLX330T*	5VSX95T*	5VSX240T*						
2nd	5	(20)	17	(68)	5	(20)	17	(68)	6	(24)	15	(60)
3rd	4	(20)	14	(70)	2	(10)	8	(40)	5	(25)	14	(70)
Virtex6-2												
	6VLX240T	6VLX760	6VLX240T*	6VLX760*	6VSX315T*	6VSX475T*						
2nd	14	(56)	46	(184)	18	(72)	59	(236)	24	(96)	37	(148)
3rd	11	(55)	37	(185)	17	(85)	39	(195)	23	(115)	35	(175)

注 1) カッコ内は演算器数

注 2) チップ型番に* は、DSP ブロック使用

Tab. 4: 表 3 から算出したピーク性能 (GFLOPS)

VirtexII Pro-6							
	2VP70	2VP100					
2nd	5.42	8.13					
3rd	6.10	8.81					
Virtex4-11							
	4VLX100	4VLX200	4VSX35	4VSX55	4VFX60	4VFX140	
2nd	5.88	10.08	1.85	2.77	3.70	9.24	
3rd	5.25	10.50	2.31	3.47	3.47	5.78	
Virtex5-2							
	5VLX110T	5VLX330T	5VSX95T	5VSX240T	5VLX110T*	5VLX330T*	
2nd	5.44	18.49	7.44	18.59	6.20	21.07	
3rd	5.44	19.03	7.75	21.69	3.10	12.40	
Virtex6-2							
	6VLX240T	6VLX760	6VSX315T*	6VSX475T*	6VLX240T*	6VLX760*	
2nd	13.32	43.77	36.28	55.93	27.21	89.19	
3rd	13.08	44.01	43.46	66.14	32.12	73.70	

Tab. 5: ユーザ I/O ピン数と、DRAM チャンネル数・FPGA 間通信チャンネル数

	Maximum User I/O	DRAM None	DRAM 1ch	DRAM 2ch
2VP70	996	12	10	9
2VP100	1164	15	13	11
4VLX100	960	12	10	8
4VLX200	960	12	10	8
4VSX35	448	5	3	1
4VSX55	640	7	5	4
4VFX60	576	6	5	3
4VFX140	896	11	9	7
5VLX110T	680	8	6	4
5VLX330T	960	12	10	8
5VSX95T	640	7	5	4
5VSX240T	960	12	10	8
6VLX240T	720	8	7	5
6VLX760	1200	15	13	11
6VX315T	720	8	7	5
6VSX475T	840	10	8	6

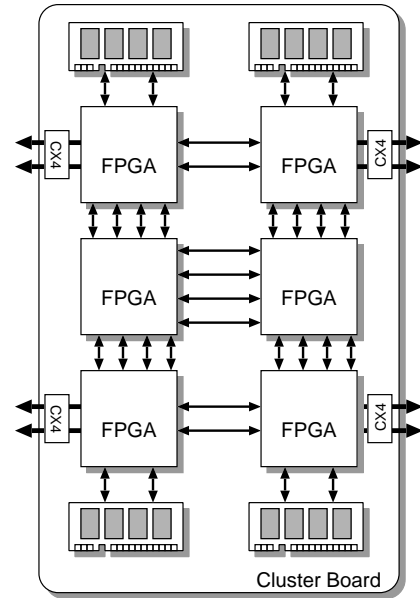


Fig. 3: 提案システムの構成

えると、DRAM を 2 チャンネル接続した FPGA から引き出せる配線が多い FPGA で 64bit×4~7 チャンネル程度、DRAM なしでも 5~8 チャンネル程度、という状況は基板設計の時点で FPGA 間の通信量について十分に検討が必要であることを意味している。

4.2 シリアル入出力

表 5 に挙げた FPGA の多くは、最大 2.5Gbps で動作する高速シリアルトランシーバを 16 チャンネル以上備えている。これらは 4 チャンネルを 1 リンクとして、10Gbps の伝送路として用いることが多い。これは、8b10b コーディングした場合には実効速度が最大で 8Gbps となるので、64bit のデータを 125MHz 程度で送ることができる。

演算器やメモリが 200MHz 以上で動作することを考慮すると、この速度は前節の平行入出力と比べて若干遅いことになり、充分注意して使わないとボトルネックになる可能性がある。

5. まとめと考察

5.1 まとめ

本稿では Xilinx 社の FPGA の諸元と、UPACS のソースコード中のいくつかのサブルーチンを FPGA に実装した結果から、FPGA の回路・入出力の量リソースに関して見積もりを行った結果を示した。その結果、

- FPGA 内の論理ブロック・乗算ブロックなどのリソース配分によって、最適となる演算器セットの割合が異なるため、FPGA の選定には慎重な検討が必要
- 現在市販されている FPGA では、CFD のアプリケーション全体を回路容量が不足するため、複数の FPGA に分割せざるを得ない
- 複数の FPGA に分割する場合、外部メモリへの転送バンド幅と FPGA 間の転送バンド幅を充分にとる必要があるが、パッケージのピン数制約から多くの困難を伴うことが予想される

などの点が明らかになった。

5.2 CFD 向けシステム構成についての提案

上記の点を克服していくには、BEE3 のようなクラスタ構成をとり、平行通信による十分な通信バンド幅を持つ数個の FPGA を搭載した基板 (クラスタボード) にひとつの機能ユニット (一連の演算パイプライン) を納めて、高速シリアルトランシーバを用いて基板を接続し、全体のシステムを構成するのが望ましいと考えられる。

BEE3 ではクラスタ内の結合がリングになっており、4 つの FPGA は対称であるが、深いパイプラインを構成するような場合には、図 3 のようにパイプラインの方向に十分なバンド幅を用意し、横方向のバンド幅は抑えるような構成が使いやすいのではないかと考えられる。

5.3 今後の課題

メモリアクセスの効率化を図ったり、パイプラインを複数の FPGA に分割する作業は非常に時間のかかる作業であり、これを支援、あるいは半自動化するような記述言語や、ツールの開発が必要であると考えており、今後の重要な課題としたい。

クラスタボードの構成は現在のところ例としての提案しかできない段階であるが、これは実際のデータフローグラフを元にプロファイリングを行い、FPGA の規模、数、配置などを見積もる必要がある。また、並列計算機の場合と同様に、相互接続網の形態や性能がシステム全体の性能に影響することは明白である。これらは今後、FPGA 版 UPACS の開発と並行してデータ収集を行い、検討していく。

参考文献

1. Jeffrey M. Arnold, Duncan A. Buell, and Elaine G. Davis. Splash 2. In *Proceedings of the fourth annual ACM symposium on Parallel algorithms and architectures*, pp. 316–322. ACM Press, 1992.
2. Tsuyoshi Hamada, Naohito Nakasato, and Toshikazu Ebisuzaki. A 236 Gflops astrophysical simulation on a reconfigurable super-computer. In *Proceedings of IEEE/ACM SC 2005 Conference*, Nov. 2005.
3. 長名保範, 吉見真聡, 岩岡洋, 小嶋西川 由理, 舟橋啓, 広井賀子, 柴田裕一郎, 岩永直樹, 北野宏明, 天野英晴. FPGA を用いた汎用生化学シミュレータ ReCSiP. 電子情報通信学会論文誌 D, Vol. J89-D, No. 6, pp. 1163–1172, Jun. 2005.
4. Hirokazu Morishita, Yasunori Osana, Naoyuki Fujita, and Hideharu Amano. Exploiting memory hierarchy for a computational fluid dynamics accelerator on FPGAs. In *Proceedings of the International Con-*

ference on Field-Programmable Technology 2008, pp. 193–200, Dec. 2008.

5. Kenta Inakagata, Hirokazu Morishita, Yasunori Osana, Naoyuki Fujita, and Hideharu Amano. Modularizing flux limiter functions for a computational fluid dynamics accelerator on FPGAs. In *Proceedings of the 10th International Conference on Field Programmable Logic and Applications*, pp. 654–657, Sep. 2009.
6. Oskar Mencer, Kuen Hung Tsoi, Stephen Cramer, Timothy Todman, Wayne Luk, Ming Yee Wong, and Philip Heng Wai Leong. CUBE: A 512-FPGA cluster. In *The 5th Southern Programmable Logic Conference (SPL)*, Apr. 2009.
7. John D. Davis, Charles P. Thacker, and Chen Chang. BEE3: revitalizing computer architecture research. MSR Technical Report MSR-TR-2009-45, Microsoft Research, 2009.
8. William D. Smith and Austars R. Schnore. Towards an rcc-based accelerator for computational fluid dynamics. *The Journal of Supercomputing*, Vol. 30, No. 3, pp. 239–261, Dec 2004.