

キャッシュ機構を考慮した組み込み向け CPU の性能評価

川村 暁*・佐々木慶文*

Performance Evaluation of Embedded CPUs Based on the Cache Mechanism

Satoshi KAWAMURA* and Yoshifumi SASAKI*

**Dept. of Information Technology and Electronics, Faculty of Science and Engineering,
Ishinomaki Senshu University*

1 Shinmito Minamisakai Ishinomaki-shi MIYAGI. 986-8580 JAPAN

Abstract

This paper estimated the performance based on the cache mechanism of embedded CPUs. Embedded CPUs have limitation in the mechanism which improves performance. Therefore, revelation of performance is different from general CPU. As a result of fundamental operation and matrix operation estimating, it is suggested that the performance of CPU has influence of a cash mechanism.

1. はじめに

近年、スマートフォンや自動車など、身の回りの様々な製品は著しく高機能化、高性能化を遂げている。例えば、最新の自動車には衝突回避システムや自動駐車機能など、高度な処理を伴う機能が搭載されつつある。このような機能は、組み込み向け CPU を基本としたシステム上で、ソフトウェアにより実現されている。

このような高度な処理を実現するためには、システムの中核をなす組み込み向け CPU に高い処理性能が要求されることになる。一方で、システムは製品に組み込まれることが前提であるため、低コスト、低消費電力、取扱いの容易さなど、様々な制約を満たさなければならない^{(1)~(4)}。相反する制約条件を満たすためには、製品に最適な組み込み向け CPU を選択することが重要である。このためには、適切に組み込み向け CPU の性能を評価し、設計の指標とすることが重要である。

筆者らは、これまでに組み込み向け CPU の性能についていくつかの視点から評価を行い、製造元が公開している技術資料や動作周波数に代表されるような公称値だけでは、組み込み向け CPU を選択することが困難であることを明らかにしている^{(5)~(7)}。

本稿では、新たに組み込み向け CPU のキャッ

シュ機構に着目し、その実処理性能を評価した結果について報告する。具体的には、整数演算、浮動小数点演算および行列演算性能を様々な組み込み向け CPU で測定した。なお、行列演算はサイズを変えて演算時間を測定している。実験の結果、整数および浮動小数点演算の演算時間とサイズ別の行列演算時間の間には相関が見られないこと、動作周波数だけでは記述できないこと、処理系の影響が大きいことが示された。

2. 処理性能の計測方法

本稿では、組み込み向け CPU の性能として、整数演算および浮動小数点演算の2種類の数値演算性能と、メモリとのやりとりを必要とする行列演算の性能をベンチマークプログラムにより測定した。数値演算の性能は、古典的なベンチマークである Dhrystone (整数演算の性能を測る) と Whetstone (浮動小数点演算の性能を測る) を用いて測定した。ソースコードは、google の提供している UNIXbench に含まれるコードを用いた⁽⁸⁾。行列演算性能評価では、固有値行列の計算の前処理などで頻繁に用いられる Hessenberg 行列の演算を採用し、この演算時間を測定した。Hessenberg 行列の演算には、参考文献 (9) にあるソースコードを用いている。この処理は $O(n^2)$

*石巻専修大学理工学部情報電子工学科

キャッシュ機構を考慮した組み込み向け CPU の性能評価

表 1 比較に用いた計算機。網掛けのある計算機は、EABI (Embedded Application Binary Interface) でコンパイルし、実測を行った。

略称		KHGO0	KHGO2	A440O0	A440O2	A500O0	A500O2	A840O0	A840O2
Machine name		Kurobox HG		Armadillo-440(EABI)		Armadillo-500(EABI)		Armadillo840	
CPU	name	Motorola SPS MPC8241 PowerQUICC II (603e)		Freescale i.MX25(ARM926EJ-S) ARM926EJ-S rev 4 (v5l)		Freescale i.MX31(ARM1136JF-S) ARMv6-compatible processor rev 4 (v6l)		ルネサス-R-Mobile A1 (ARM Cortex-A9)	
	Frequency [MHz]	266		400		532		792	
	L1 cache [Kbytes]	16+16		16+16		16+16		64+64	
	L2 cache [Kbytes]	0		128		128		256	
	bogoMIPS	173.05		199.06		399.76		1576.53	
Memory	size [Mbytes]	128		128		128		512	
OS	distname	Linux ms-06s 2.6.20-kuroboxHG #2 Mon Jun 2 17:56:22 JST 2008 ppc GNU/Linux		Linux debian 2.6.26-at19 #1 PREEMPT Thu Apr 10 23:05:46 JST 2014 armv5tej GNU/Linux		Linux debian 2.6.26-at14 #1 PREEMPT Sat Apr 23 17:54:36 JST 2011 armv6l GNU/Linux		Linux armadillo840-0 3.4-at4 #1 PREEMPT Wed Jul 3 17:34:13 JST 2013 armv7l GNU/Linux	
Compiler	type	Self Compile		Cross Compile		Cross Compile		Cross Compile	
	name	gcc version 4.1.2 20061115 (prerelease) (Debian 4.1.1-21)		gcc version 4.3.2 (Debian 4.3.2-1.1)		gcc version 4.3.2 (Debian 4.3.2-1.1)		gcc (Debian 4.7.2-5) 4.7.2	
Optimization		O0	O2	O0	O2	O0	O2	O0	O2

略称		BBnA6O0	BBnA6O2	BBadC4O0	BBadC4O2	BBbkO0	BBbkO2	RasPiO0	RasPiO2
Machine name		BeagleBone RevA6		BeagleBoard RevC4		BeagleBone black RevA5B		RaspberryPi RevB	
CPU	name	TI AM3359, ARM Cortex-M3 ARMv7 Processor rev 2 (v7l)		TI OMAP3530, ARM Cortex-A8 CPU ARMv7 Processor rev 3 (v7l)		Texas Instruments Sitara AM3359AZCZ100 1GHz (Cortex-A8 + PowerVR SGX530)		Broadcom BCM2835 ARMv6-compatible processor rev 7 (v6l)	
	Frequency [MHz]	720		700		1000		700	
	L1 cache [Kbytes]	32+32		16+16		32+32		16+16	
	L2 cache [Kbytes]	256		256		256		128	
	bogoMIPS	498.89		698.47		297.4		464.48	
Memory	size [Mbytes]	256		256		512		512	
OS	distname	Linux beaglebone 3.2.34 #1 Wed Nov 21 14:17:11 CET 2012 armv7l GNU/Linux		Linux bb-1 2.6.32 #3 PREEMPT Mon Aug 22 02:07:19 JST 2011 armv7l GNU/Linux		Linux beaglebone 3.8.13 #1 SMP Mon May 20 17:07:58 CEST 2013 armv7l GNU/Linux		Linux raspberrypi 3.6.11+ #371 PREEMPT Thu Feb 7 16:31:35 GMT 2013 armv6l GNU/Linux	
Compiler	type	Self Compile		Cross Compile		Self Compile		Self Compile	
	name	gcc (GCC) 4.5.4 20120305 (prerelease)		gcc version 4.3.2 (Debian 4.3.2-1.1)		gcc (Linaro GCC 4.7-2013.02-01) 4.7.3 20130205 (prerelease)		gcc (Debian 4.6.3-14+rp1) 4.6.3	
Optimization		O0	O2	O0	O2	O0	O2	O0	O2

に従うことが知られている⁽⁹⁾。

表 1 に、性能評価に用いた 8 種類のシステムの仕様を示す。基本的には、組込み CPU 評価 (開発) ボードで提供されている環境を用いて実験を行った。このため、Linux の kernel バージョンやコンパイラ gcc のバージョンは異なっている。また、ARM 系 CPU の場合、OABI (Old application binary interface) ではなく EABI (Embedded application binary interface) を用いた方が、処理速度が高速になる場合がある^{(3), (7), (10)~(12)}。ため、いくつかの環境では、EABI 対応の環境を構築して実験を行っている (表 1 の網かけ部分)。

なお、組込み向け CPU を用いた評価 (開発) 環境では、いくつか、一般的な PC とは異なる点があることに留意が必要である。一つ目は、アプ

リケーション開発が実機で完結しているセルフ開発環境によるシステムだけではなく、母艦となる PC 上でアプリケーションを作成 (コンパイル) し、実機へ転送した上で実行するクロス開発環境も存在することである。二つ目は、実機は VGA などの画面出力がなく、シリアル通信 (RS-232C) やネットワーク (Ethernet) 経由で入出力を行う環境となることである。即ち、ROM イメージを作成して焼き付ける等としないかぎり、限りなくプレーンな環境は用意できないことを指す。これらの点は、一般的な PC とは異なる、組込み向け CPU ならではの環境要件である。

3. 計算機実験結果

実機による計算機実験結果を示す。それぞれの項目は 10 回計測したときの平均値 (処理回数ま

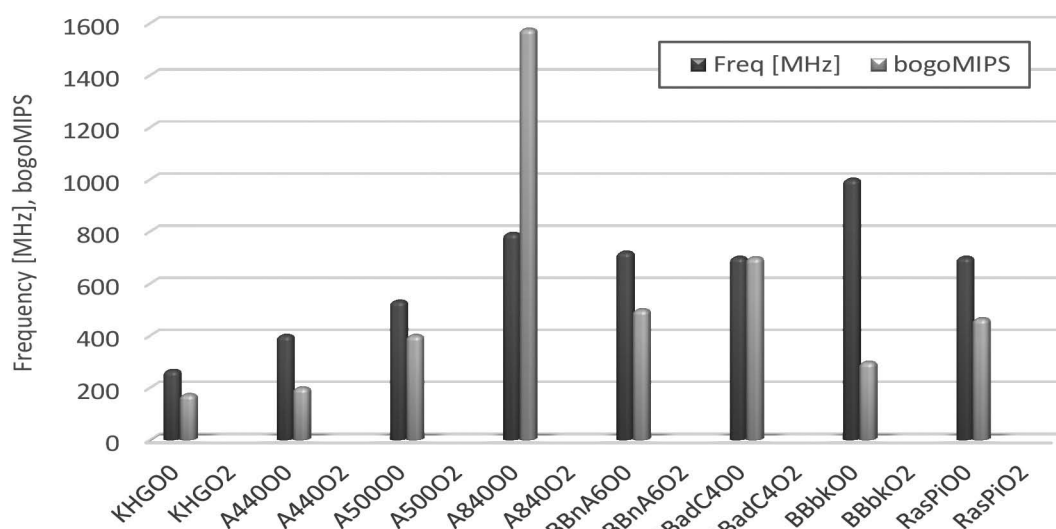


図1 CPUの動作周波数 [MHz] と bogoMIPS 値。bogoMIPS 値と動作周波数の間には相関が見られない。なお、O0 だけで示しているが、最適化別の数値とはならない（動作周波数はハードウェア的な意味で一定であり、bogoMIPS 値は OS 起動時に測定される）ため、O2 でも同値となる。

たは処理時間）を示している。ここで、Linux 上の実験では、値が大きく振れる場合があるため、振れ幅が大きすぎる場合はその値は採用せず、計測し直した場合がある（マルチタスク OS であるため）。

はじめに、基本的な情報である、各 CPU の動作周波数と bogoMIPS 値を示す（図 1）。なお、表 1 では、O0 と O2 別に計測するように示しているが、動作周波数および bogoMIPS は同値となるため、O0 にだけ値を記している。動作周波数と bogoMIPS に、相関は全く見られない。bogo と冠しているように、この値は当てにならないことが分かる。

3.1 Dhrystone および Whetstone による性能評価結果

Dhrystone の結果を図 2 に、Whetstone の結果を図 3 に示す。図 2 より、動作周波数が最も高い Beagle Bone Black (bbblk) より、EABI+アーキテクチャを最適化している Beagle Board C4 (BBadC4) の方が高速であることがわかる。このように、動作周波数だけでは実際の演算性能を記述しきれていないことがわかる。

図 3 の Whetstone の結果から、動作周波数が最も高速な Beagle Bone Black (bbblk) が高速

なわけではなく、Armadillo 840 (A840) が高速であることがわかる。特筆すべきは、-O2 とした場合の A840 の高速性である。Whetstone は浮動小数点演算に対するベンチマークであるから、FPU が優れていることが示唆される。

動作周波数と性能の関係について考察するため、Dhrystone 値 [MDIPS] を動作周波数 [MHz] で除した結果を図 4 に、Whetstone [MWIPS] を動作周波数 [MHz] で除した結果を図 5 に示す。

図 4 の動作周波数あたりの Dhrystone 値でみると、PowerPC に基づく KuroBoxHG と、EABI+CPU 環境のもの (Armadillo440・500、BeagleBoard C4) が動作周波数あたりの処理効率がよい事がわかる。同様に、図 5 の動作周波数あたりの Whetstone 値でみると、PowerPC に基づく KuroBoxHG と、Armadillo 840・RaspberryPi O2 の効率がよいことがわかる。搭載している FPU の性能+ライブラリの性能によると考えられる。

また、図 4 で効率がよかったもの (KuroBoxHG、Armadillo440・500、BeagleBoard C4) と、図 5 で効率がよかったもの (KuroBoxHG、Armadillo440、RaspberryPi O2) が完全には一致しないことから、それぞれの処理環境毎の得手

Dhrystone [MDIPS]

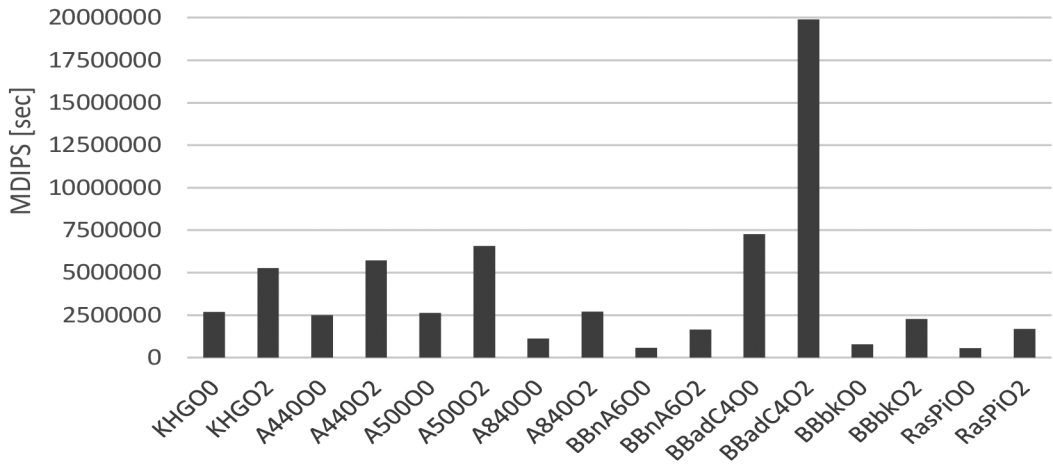


図2 Dhrystone ベンチマークの結果。1秒あたりの Dhrystone 値、ただし M (100 万) Dhrystone で示している。

Whetstone [MWIPS]

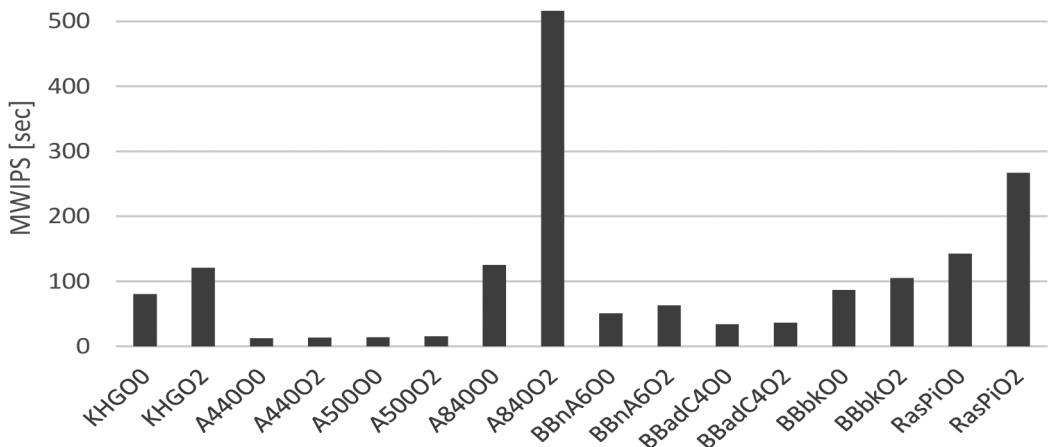


図3 Whetstone ベンチマークの結果。1秒あたりの Whetstone 値、ただし M (100 万) Whetstone で示している。

不得手があることがわかる。

3.2 Hessenberg 行列

サイズを変化させて行列演算を行った場合の測定結果を図6に示す。行列演算としては、Hessenberg 行列を用いている⁽⁹⁾。

図6では、行列サイズが大きくなると処理時間が増加しているが、途中から傾き（増加率）が変わっているものがいくつかある。これは、L1

データキャッシュ（一次データキャッシュ）に収まりきらなくなったためと推察される（表1で示したが、L1 データキャッシュのサイズは、16Kbytes と 32Kbytes に分かれる。このため、行列のサイズが大きくなると、データキャッシュからあふれるため、処理効率が悪くなるためである）。

図7～図9および表2は、行列のサイズ8×8の処理速度を基準にし、行列サイズの増加と処理

Dhry / Freq

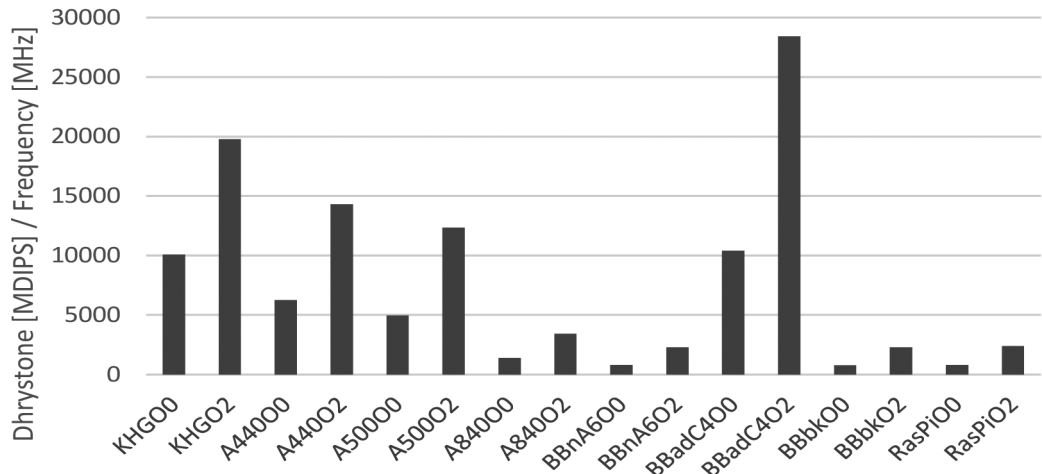


図4 CPUの動作周波数 [MHz] で規格化した Dhrystone 値 [MDIPS]。

Whet / Freq

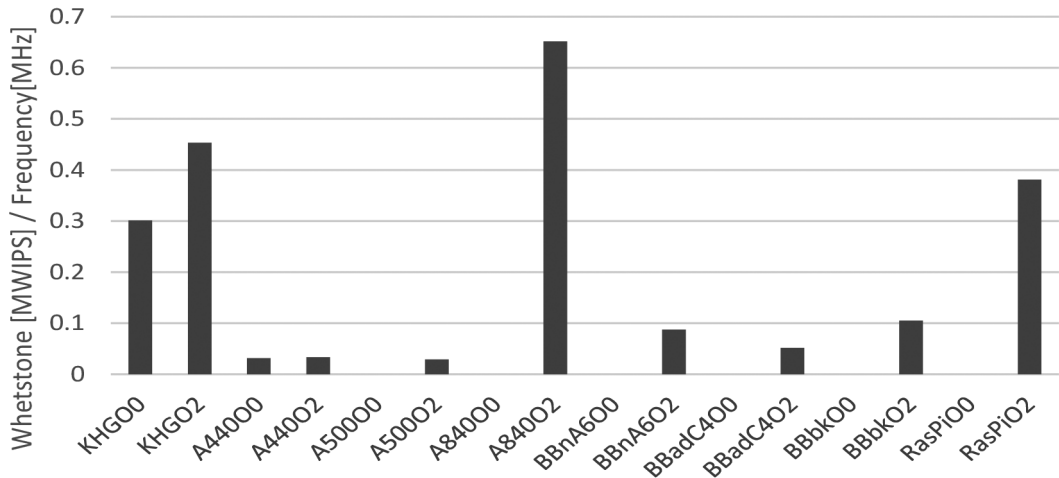


図5 CPUの動作周波数 [MHz] で規格化した Whetstone 値 [MWIPS]。

時間の関係を示している。図7は、 8×8 行列の処理時間を基準（1.0）とし、各サイズの行列演算が基準に対して何倍の時間を要したかを示しており、図8は図7を、行列の実データサイズで書き直したグラフである。実データサイズは、 $x \times x \times \text{float}$ で導出できる（ x ：行列のサイズ、float：変数のサイズ。floatは4bytes）。

これらから、サイズ毎に効率の上下があることがわかる。KuroboxHG と RaspberryPi の効率

の悪化が目立っている。一般的に、O2 が若干、速度低下が少ない。データサイズ 16384bytes（ $=64 \times 64$ ）の場合も、36864bytes（ $=96 \times 96$ ）の場合も、大きな乖離はない。RaspberryPi は 36864 において、O2 の劣化がある。データサイズが大きい場合（65536（ $=128 \times 128$ ））、L1 データキャッシュからあふれる状態での動作では、O2 の性能が劣化しがちである。KuroboxHG と RaspberryPi では、この傾向が著しい。

キャッシュ機構を考慮した組み込み向け CPU の性能評価

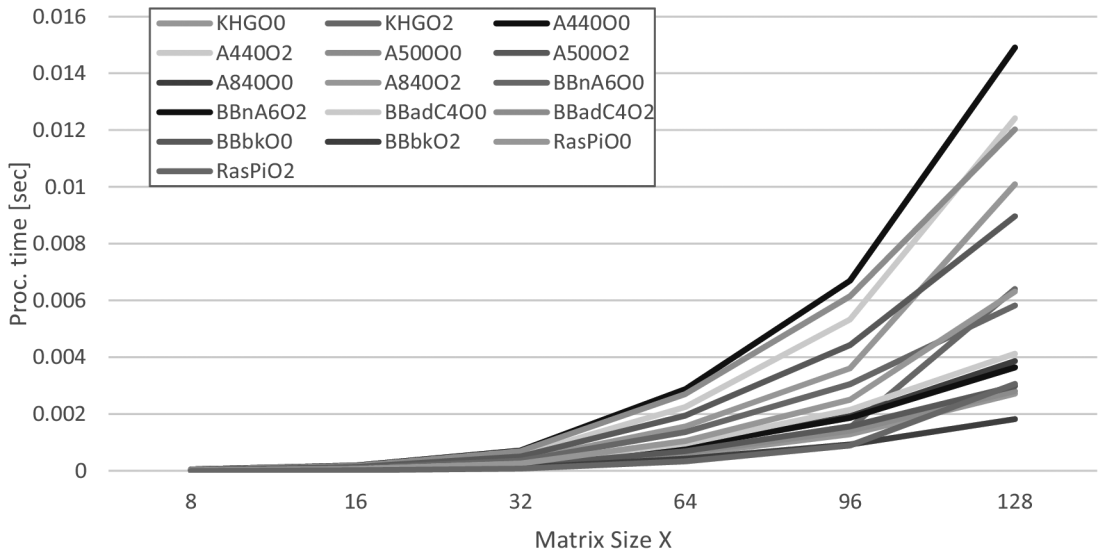


図6 Hessenberg 行列のサイズと演算時間 [sec]。

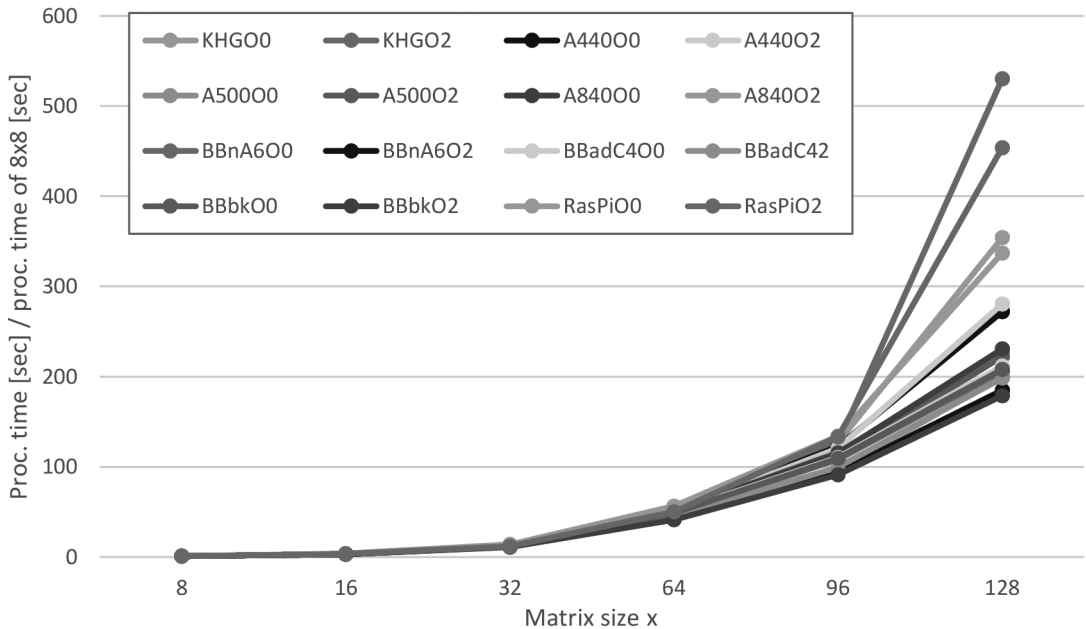


図7 Hessenberg 行列の最小サイズ (8×8) を基準とした処理時間。

表 2 に、Hessenberg 行列演算の、8×8 行列演算を基準 (1.0) とした場合の相対的な速度を数値で示した。8×8 行列の演算速度から、どれだけ処理効率が落ちているかが示される。差が大きいほど、サイズ依存性が著しいことになる。KuroboxHG と RaspberryPi の増加が目立つ。

実験結果について考察する。本稿の実験のよう

な同一の処理で評価すると、CPU の得手不得手や特質が現れることがわかる。PowerPC コアの PowerQUICC は、数値演算は高速であること、同じ ARM 系の CPU でも特に浮動小数点演算性能に大きな違いがあること、ABI が異なると性能が異なることが明らかとなったが、この要因としては FPU の有無や FPU の作り込みの差が影

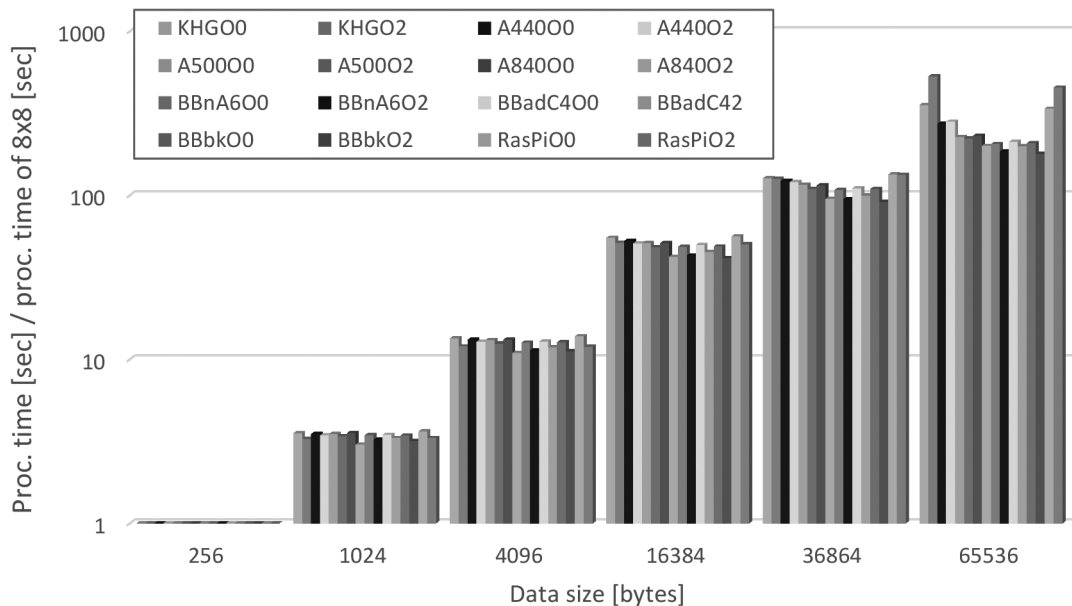


図8 Hessenberg 行列の最小サイズ (8×8) を基準とした実データサイズに対する処理時間の相対値 (処理時間増加率)

表2 処理時間増加率一覧

Data Size [bytes]	KHGO0	KHGO2	A440O0	A440O2	A500O0	A500O2	A840O0	A840O2
256	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1024	3.534	3.268	3.501	3.435	3.500	3.397	3.548	3.025
4096	13.414	11.995	13.168	12.772	13.087	12.471	13.174	10.944
16384	55.003	51.239	52.500	50.778	51.019	48.180	51.007	42.187
36864	126.887	126.029	122.178	120.438	115.876	109.604	114.933	94.893
65536	354.176	530.072	271.955	280.454	226.677	222.194	230.605	200.451

Data Size [bytes]	BBnA6O0	BBnA6O2	BBadC4O0	BBadC42	BBbkO0	BBbkO2	RasPiO0	RasPiO2
256	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1024	3.447	3.235	3.462	3.313	3.431	3.174	3.630	3.313
4096	12.626	11.363	12.819	11.844	12.703	11.196	13.799	11.932
16384	48.406	42.965	49.664	45.043	48.711	41.429	56.266	50.235
36864	107.922	94.448	110.393	99.523	109.334	91.227	133.839	133.113
65536	205.917	184.620	211.880	198.418	207.969	178.922	337.086	453.799

1～50 51～100 101～200 201～300 301～400 401～500 500～

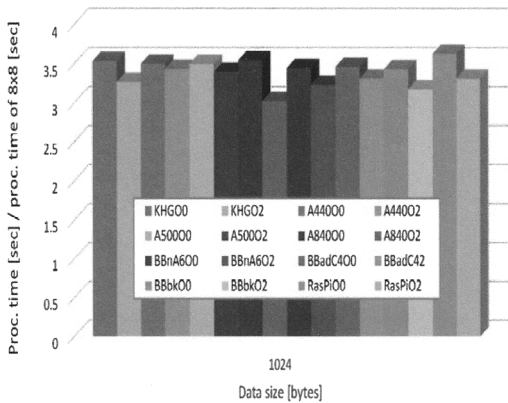
響していると考えられる。また、単純な演算 (Dhrystone、Whetstone) が高速でも、行列演算のような実応用処理が高速とは限らない (例: BeagleBoard C4) 場合があることが明らかと

なった。

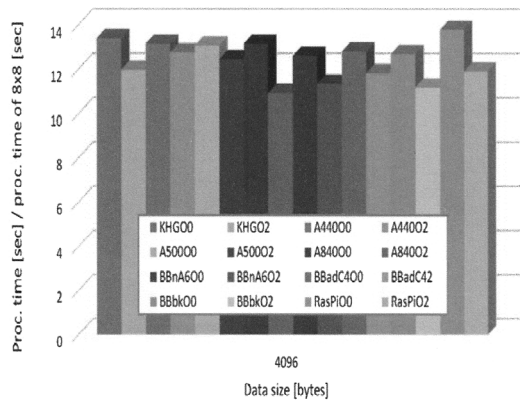
このように、実時間評価と、相対的な評価 (処理時間の増加率) は、結果が大きく異なる上に、公称値だけをシステム設計の指針とすることは不

キャッシュ機構を考慮した組み込み向け CPU の性能評価

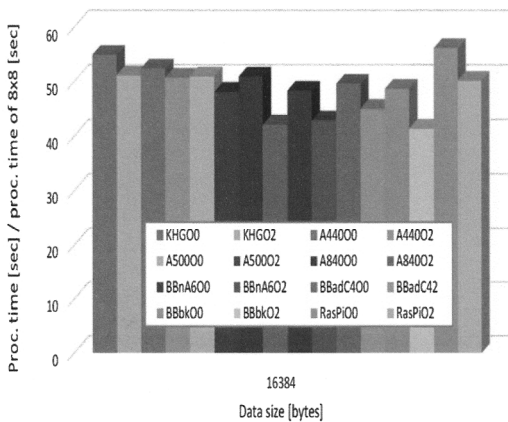
(a) data size 1024[bytes]



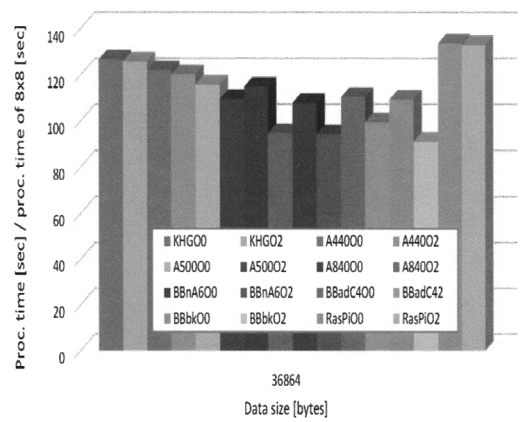
(b) data size 4096[bytes]



(c) data size 16384[bytes]



(d) data size 36864[bytes]



(e) data size 65536[bytes]

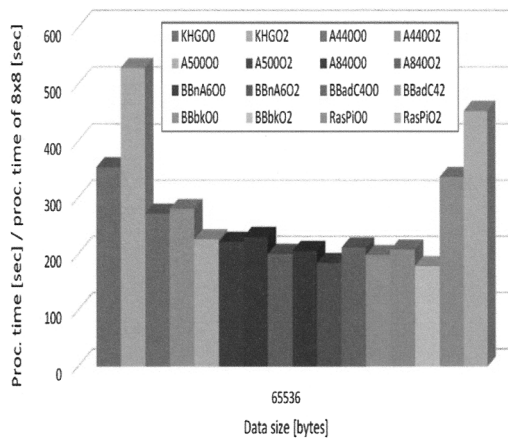


図9 行列データサイズ別の処理時間増加率。処理時間の増加の様子が、処理系によって大きく異なることがわかる。とくに (e) において、KHGO2 および RasPIO2 の、処理時間の増加率（劣化率）が大きいことがわかる。

適切であることが示された。

4. まとめ

技術資料から推定することが困難な組み向け CPU の実処理性能をはかる一つの方法として、整数演算性能と浮動小数点演算性能およびサイズを変えながらの Hessenberg 行列演算を用いて検討を行った。本稿ではキャッシュ機構観点からの性能評価を検討したが、実験の結果、整数演算および浮動小数点演算時間とサイズ別の行列演算時間の間には相関が見られないこと、クロック周波数だけでは性能を記述できないこと、処理系の影響も大きいことが示された。特に、整数演算、浮動小数点演算の性能から Hessenberg 行列の演算速度を推測しにくいこと、キャッシュの影響、FPU の影響だけではなく、ABI による影響もあることが示された。

参考文献

- (1) Heath Steve (2002) Embedded systems design Second Edition, Newnes.
- (2) Peter Marwedel (2003) Embedded System Design, Springer.
- (3) Rob Toulson, Tim Wilmshurst (2012) Fast and Effective Embedded Systems Design: Applying the ARM mbed, Newnes.
- (4) Andrew N. Sloss ら著、アーム (翻訳) (2007) ARM 組み込みソフトウェア入門—記述例で学ぶ組み込み機器設計のためのシステム開発 (Design Wave Advance シリーズ)、CQ 出版
- (5) 川村、佐々木、佐藤、丸岡 (2010) 組み向け CPU の性能評価試み—最適化の影響—、石巻専修大学紀要第 21 号、pp.19-25.
- (6) 川村、佐々木、佐藤、丸岡 (2011) 代表的な数値計算を用いた組み向け CPU の性能評価、石巻専修大学紀要第 22 号、pp.19-25.
- (7) 川村、小山、佐々木 (2010) ARM 系 CPU の性能評価—Application Binary Interface の及ぼす影響—、石巻専修大学研究紀要第 24 号、pp.87-94.
- (8) UNIX bench (URL), <https://code.google.com/p/byte-unixbench/>
- (9) William H. Press ら著 (2007) Numerical Recipes 3rd Edition: The Art of Scientific Computing, Cambridge University Press
- (10) ARM ltd. (2001, 2002) ARM926EJ-S/ (Rev:0 S (Rev:0) テクニカルリファレンスマニュアル、ARM ltd.
- (11) ARM ltd. (2002, 2003) ARM1136JF-S/ ARM1136J-S (Rev:r0p2) テクニカルリファレンスマニュアル、ARM ltd.
- (12) Arm ltd. (2009) Application Binary Interface for the ARM Architecture v2.08 (PDF), http://infocenter.arm.com/help/topic/com.arm.doc.ih0036b/IHI0036B_bsabi.pdf

