

実行頻度の偏りを利用した命令コード割り当てによる命令ROMの低消費電力化

井上, 弘士
福岡大学工学部電子情報工学科

モシニヤガ, ワシリー
福岡大学工学部電子情報工学科

村上, 和彰
九州大学大学院システム情報科学研究所

<https://hdl.handle.net/2324/3623>

出版情報：第15回 回路とシステム（軽井沢）ワークショップ，pp.565-570，2002-04．電子情報通信学会
バージョン：
権利関係：

実行頻度の偏りを利用した命令コード割当てによる 命令ROMの低消費電力化

Power Reduction Technique for I-ROMs Exploiting Instruction Frequency

井上弘士[†]

モシニヤガ・ワシリー[†]

村上和彰^{††}

[†]福岡大学 工学部 電子情報工学科

^{††}九州大学大学院 システム情報科学研究科

Koji INOUE[†]

Vasily MOSHNYAGA[†]

Kazuaki MURAKAMI^{††}

[†]Dept. of Electronics Engineering and Computer Science, Fukuoka University

^{††}Dept. of Informatics, Kyushu University

1 はじめに

携帯電子機器市場の拡大を背景に、組み込みシステムにおける低消費エネルギー化が重要視されるようになった。特に、命令ROMやデータRAM、CPUなどが1個のLSIに集積されるシステムLSIの低消費エネルギー化は、今後の携帯電子機器システムにおいて必要不可欠な要素技術である。このようなシステムLSIの全消費エネルギーは、1)CPUコアにおける消費エネルギーと、2)メモリ・システムにおける消費エネルギーに大別できるが、本稿では後者のメモリ・システムに焦点を当てる。

メモリに関する消費エネルギー (E_{MEMORY}) は、プログラム実行における総メモリ・アクセス回数 (N_{ACC}) と、メモリ・アクセス当たりの平均消費エネルギー ($AMAE$) の積で近似できる。ここで、 N_{ACC} は実行プログラムの特徴に依存し、 $AMAE$ は主にメモリ・サイズに依存する。組み込みシステムにおいて、命令キャッシュを搭載していない場合には命令フェッチ毎に命令ROMアクセスが発生する。また、アプリケーションの高機能化に伴い、命令ROMに格納されるプログラムのサイズは年々増加傾向にある。よって、組み込みシステムの低消費エネルギー化を実現するためには、命令ROMの低消費エネルギー化が重要課題となる。

そこで本稿では、命令ROMの低消費エネルギー化を目的とした命令コード割当て手法を提案する。ROMアクセス当たりの消費エネルギーは読出し対象データのビットパターンに依存する。メモリ内に格納される情報は、ビットラインに対するメモリセルの接続情報で表されるためである(つまり、読出すべき情報が1か0かによって、プリチャージされたビットラインがディスチャージされるか否かが決定される)。通常、命令コードはデコードの容易性を考慮して決定される。これに対し、提案手

法では、命令ROMのビットライン・スイッチング回数削減を目的として命令コードを決定する。つまり、事前にアプリケーション・プログラムの解析を行い、実行頻度の高い命令を検出する。そして、これらの命令に対し、ビットラインのスイッチング回数が最小となるよう、命令コードの割当てを行う。これにより、プログラム実行における総ビットライン・スイッチング回数を削減でき、命令ROMの低消費エネルギー化を達成できる。

以下、第2章では、ROMアクセスにおける消費エネルギーに関して説明する。また、これまでに提案された低消費エネルギー化技術を紹介する。次に、第3章では提案する命令コード割当て手法を提案し、第4章でベンチマーク・プログラムを用いた定量的評価を行う。最後に、第5章で簡単にまとめる。

2 命令ROMの低消費エネルギー

本節では、ROMアクセス動作と、それにより消費されるエネルギーに関して説明する。ここで、図1に示すようにマスクROMを例にとるが、本稿で提案する手法はフラッシュROM等にも適用可能である。

図1に示すように、マスクROMでは、ビットラインに対してメモリセルを接続するか否かによってビット情報を記憶する。ここで、図1において、ビットラインに接続されているメモリセルはビット情報「0(ロー・レベル)」を、ビットラインに接続されていないメモリセルはビット情報「1(ハイ・レベル)」を記憶していると仮定する。一般に、メモリ・アクセス時間を短縮するため、全てのビットラインは V_{PRE} ($V_{dd}/2$ 程度) にプリチャージされる。そして、ROMに対する読出し要求が発生した際、アドレスのデコードを行い、対応するワードラインを活性化する。もし、読出すビット・データが

表 1: 命令実行頻度 (mpeg_decode using “mei16v2.m2v” input file)

Ranking	Op-Field			Func-Field			Reg-Field		
	Inst.	Freq.	Op-code (#of UCBs)	Inst.	Freq.	Op-code (#of UCBs)	Reg.	Freq.	Op-code (#of UCBs)
1	Special	17.9%	000000(6)	ADD	9.0%	100000(5)	R1	26.8%	00001(4)
2	ADDI	16.5%	001000(5)	MOVI2FP	1.9%	110101(2)	R29	14.8%	11101(1)
3	LW	15.0%	100011(3)	SLT	1.9%	101010(3)	R2	12.3%	00010(4)
4	SW	12.8%	101011(2)	SUB	1.5%	100010(4)	R3	10.7%	00011(3)
5	BNEZ	4.2%	000101(4)	OR	1.3%	100101(3)	R31	6.4%	11111(0)

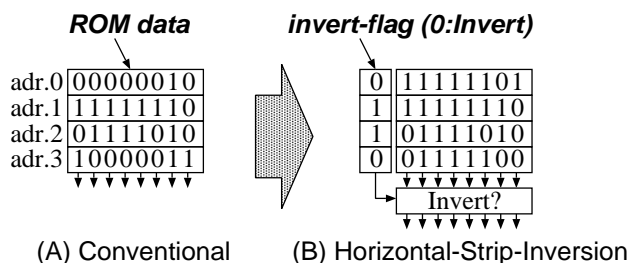


図 2: HSIによる低消費エネルギー化

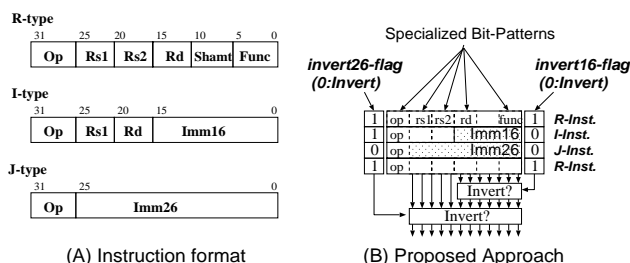


図 3: 提案手法による低消費エネルギー化

4 実行頻度の偏りを利用した命令コード割当て

4.1 従来の低消費エネルギー技術

第2節で説明したように、命令ROMのビットラインにおける消費エネルギー E_{BL} は、読出しデータのビット・パターンに依存する。つまり、読出しデータに含まれるUCB(UnConforming Bit: ビットラインに接続されているメモセル)の数だけビットライン・スイッチングが発生する。そこで、読出されるUCB数を削減するため、HSI(Horizontal-Strip-Inversion)手法が提案された[3]。

HSI手法の基本概念を図2に示す。HSI手法では、ROMに格納すべきデータに関して、ワード毎にUCB数をカウントする。もし、総UCB数がワード・サイズの半分より大きい場合、ワードデータ内の全てのビットを反転し、ROMに格納する。それと同時に、対応するワードデータが反転されているか否かを示す1ビットのフラグをセットする。これにより、読出しデータにおいてUCB数を削減し、引いては、ROMのビットライン・スイッチング回数を削減する。

4.2 提案手法

プログラム実行時間の90%は命令コードの10%によって費やされるという「90/10の法則」が一般に知られている。この法則から、命令セット・アーキテクチャでは多くの命令が定義されているにも関わらず、頻繁に使用される(実行される)命令は数種類であると考えられることができる。

表1は、MipsベースのプロセッサにおいてMPEG2デコーダを実行した際、最も頻繁に実行された命令の種類(上位5命令)、および、最も頻

繁に参照されたレジスタ番号を示す。また、命令コードのビットパターン、ならびに、それに含まれるUCB(UnConforming Bit)の数も表している。本実験において仮定した命令フォーマットを図3(A)に示す。なお、詳細な実験環境は第5.1で述べる。表1より、実行される命令の種類に大きな偏りがあることが分かる。しかしながら、通常、命令コードは命令デコードの容易性を考慮して決定される。そのため、頻繁に実行される命令のコードに対して、多くのUCBが割り当てられているといった結果を招く場合がある。

そこで、本稿では、頻繁に実行される命令コードには可能な限り多くのCBを割り当てることで、ビットライン・スイッチング回数を削減する手法を提案する。命令コード割当てアルゴリズムは以下に示すように極めて単純である。まず、入力サンプルを用いて実行対象となるアプリケーション・プログラムを事前に実行し、命令出現頻度ならびにレジスタ参照頻度を解析する。そして、表1に示すように、命令フォーマットにおける各フィールドに関して実行頻度リストを作成する。その後、作成した実行頻度リストに従って、最も多く実行された命令(参照されたレジスタ)から順番に、多くのCBを有するビットパターンを割り当てる。例えば、表1のop-fieldに関して、SPECIAL命令には“111111”，ADDIには“111110”，LWには“111101”を割り当てる。MPEGデコーダに関して、本アルゴリズムに基づき命令コードのビット割り当て結果を図4に示す。

提案手法は、命令フォーマット中のオペレーション・コード(op-field, func-field), ならびに、レジ

スタ・フィールド (register-field) に関してのみ適用可能である。そこで、即値フィールド (immediate-field) に関しては、第 4.1 節で説明した HSI 手法を適用可能である。図 3(B) に示すように、16 ビット即値用ならびに 26 ビット即値用のインバート・フラグを設ける。そして、I 型および J 型の各命令において、即値フィールド内の UCB 数をカウントする。もし、総 UCB 数が即値サイズの半分より多い場合には、即値フィールド内の全ビットを反転する。また、対応する反転フラグをセットする。

5 評価

5.1 実験環境

提案手法の有効性を明らかにするために定量的評価を行った。具体的には、DLX ベース・プロセッサ・シミュレータである *fast* を利用し、提案手法を適用した際の命令 ROM ビットライン・スイッチング回数を測定した。なお、本評価では、ビットライン消費エネルギーによって命令 ROM 消費エネルギーを近似する (つまり、ビットライン以外での消費エネルギーは無視する)。ベンチマーク・プログラムとしては、SPEC95 ベンチマーク・サイト [8] より 2 つの整数プログラム (*099.go*, *129.compress*)、ならびに、Mediabench ベンチマーク・サイト [9] より 2 つのプログラム (*ADPCM decoder* and *MPEG2 decoder*) を用いた。以下、評価対象モデルを示す。

BASE: 本評価における基本モデルであり、特に低消費エネルギー化手法は適用していない。命令コードは *dlxsim* シミュレータ [7] で定義されたものを使用 (前述したように、*dlx-sim* の命令コードは、MIPS プロセッサのそれを基本として定義されている) する。ただし、従来型 ROM において、格納される全てのビット・データを反転可能である (HSI のようにワード単位ではない)。この場合、メモリセルより読出されたデータは常にセンスアンプにて反転され、出力される。実際、全てのシミュレーションにおいて、全ビット・データを反転しない場合と比較して、反転した場合の方が低い消費エネルギーであった。そこで、本モデルでは、命令 ROM に格納されている全てのビット・データは反転して格納されると仮定する。

HSI (Horizontal Strip Inversion): 第 4.1 節で説明した HSI 手法を適用した命令 ROM である。

ENCsp: 第 4.2 節で説明した提案手法。各プログラムの解析結果に基づき、それぞれ個別に命令コードが決定される。なお、I 型および J 型命令の即値フィールドに関しては従来手法と同様であり、即値のビットパターンがそのまま命令 ROM に格納される。

ENCcall: 第 4.2 節で説明した提案手法。ただし、全プログラムの解析結果に基づき 1 つの実行頻度

表を作成し、全プログラム共通の命令コードを決定する。ENCsp と同様、即値フィールドに関しては、即値のビットパターンがそのまま命令 ROM に格納される。

COMBsp: ENCsp において、即値フィールドには HSI 手法を適用したモデル (図 3(B))

COMBall: ENCcall において、即値フィールドには HSI 手法を適用したモデル (図 3(B))

なお、本実験において HSI 手法を適用した場合には、反転フラグ読出しのためのビットライン・スイッチングも考慮する。また、命令コード中の使用されないフィールド (例えば、シフト命令を除く R 型命令の *shumt-field*) は CB で埋められており、ビットライン・スイッチングは発生しないと仮定する。

5.2 実験結果

各ベンチマーク・プログラムにおける命令 ROM 消費エネルギー (E_{BL}) とその内訳を図 5 に示す。全ての結果は、BASE モデルの結果で正規化している。

まず、HSI 手法 (HSI) と提案手法 (ENCsp) を比較する。HSI では、最善ケースで約 10% の低消費エネルギー化を達成している。もし、反転フラグによるオーバーヘッドを完全に無視した場合でも、約 6% ~ 18% の低消費エネルギー化である。これに対し、プログラム毎に最適化した命令コード割当てを行う提案手法では、全てのベンチマーク・プログラムにおいて 30% 程度の低消費エネルギー化を達成している。これは、第 4.2 で述べたように、全てのプログラムにおいて命令の出現頻度 (ならびにレジスタ参照頻度) に偏りがあり、その結果、読出される UCB 数を大幅に削減できたためである。

次に、提案手法 (ENCsp) と組み合わせ (COMBsp) 手法に関して考察する。ENCsp では、即値フィールドには即値のビットパターンがそのまま現れる。そのため、即値フィールドに関しては、消費エネルギーを削減できない。これに対し、即値フィールドに対して HSI 手法を適用した COMBsp では大幅な消費エネルギー削減を実現している。通常、1 や 2 といった小さな値が即値として使用される場合が多い。この場合、即値のビットパターンには多くの UCB (つまり「0」) が出現する。よって、即値フィールドに対する HSI 手法の適用が大きな効果をもたらしたものと考える。COMBsp では、ワード当たり 2 ビットの反転フラグが実装される。それにも関わらず、全てのプログラムにおいて、COMBsp は 40% 以上の消費エネルギー削減を達成している。

最後に、プログラム専用命令コードを有する ENCsp と、全プログラム共通の命令コードを有する ENCcall を比較する。*adpcm_dec* を除く全てのプログラムに関して、ENCcall は ENCsp と同程度の

		Op-Field								Register-Field							
lower	upper	000	001	010	011	100	101	110	111	11111	R31	R30	R29	R28	R27	R26	R25
	000	SP	FP	J	JAL	BEQZ	BNEZ	BFPT	BFPT	BFPF	SLTUI						
	001	ADDI	MULI	SUBI	BFPF	LHU	ANDI	SF	ORI	DIVPI	XORI	LHI	NOF				
	010	RFE	TRAP	BREAK	JR	SGEUI	JALR	MODU	MODI	MULUI	MULI	SGEUI	SNEI				
	011	SEQUI	SNEI	SLEUI	SLTI	ORI	SGTI	SLEI	ANDI	SGEUI	JAL	DIVPI	SLEI	DIVPUI	LH		
	100	LB	LH	RFE	SD	LW	DIVPUI	LBU	SNEUI	LHU	JALR	LF	SLTI	LD	FP		
	101	SB	LD	SH	TRAP	SGTI	SW	SRAI	SEQI	JR	SF	SB	SD	LBU			
	110		LF		SUBI	SGTUI	BREAK	SUBUI	MODI	J	MODUI	LB	SRLI	SH	SRAI	BNEZ	
	111	SEQUI	SRLI	SNEUI	BEQZ	SLTUI	ADDUI	SGTUI	SW	SLEUI	LHI	SGEUI	LW	SLLI	ADDI	NOP	SP

11111	R31	R1
11110	R30	R29
11101	R29	R2
11011	R27	R3
10111	R23	R31
01111	R15	R30
11100	R28	R4
00000	R0	R25

Base
Code for MPEG

図 4: MPEG デコーダ用命令コード

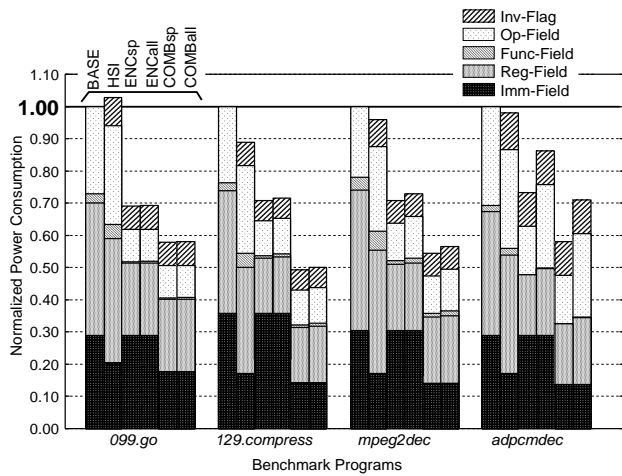


図 5: 消費エネルギー

消費エネルギー削減効果を達成している。本実験では、全てのベンチマーク・プログラムは DLX 用コンパイラ (*gcc-dlx*) でコンパイルした。したがって、全てのプログラムは、基本的に同じアルゴリズムに基づき最適化される。よって、対象プログラムが異なる場合においても出現頻度の高い命令に関して偏りが存在し、その結果、共通命令コードを用いた場合でもある程度の消費エネルギー削減効果を得ることができる。

5.3 バス消費エネルギーに与える影響

提案手法では、命令実行頻度の偏りに基づき命令コードのビットパターン割当てを行う。これにより、読出し時にビットライン・ディスチャージを発生する UCB(UnConforming Bit) の総数を削減し、低消費エネルギー化を実現できる。これは、第 2 節で述べたように、命令 ROM のビットライン・スイッチングは、読出し対象データのビット・パターンにのみ依存するためである。これに対し、命令 ROM から CPU に対するデータ・バスの消費エネルギーは、連続して読出される命令間でのビット・スイッチング回数に依存する。したがって、提案手

法により命令コードの再割当てを行った場合、従来方式と比較してデータ・バスでのスイッチング回数が増加する可能性がある。

そこで、各評価モデルにおいて、プログラム実行時におけるバス・スイッチング回数を測定した。その結果、BASE モデルと比較して、実行頻度に基づく命令コードの再割当てを行った場合、バス・スイッチング回数を 8~18% 程度削減できた。提案手法では、頻繁に実行される命令に対して CB(Conforming Bit) を多く含むビット・パターンを割当てる。その結果、CB を多く含む命令が連続してアクセスされる確率が高くなるためである。

5.4 命令デコーダ消費エネルギーに与える影響

一般に、命令コードのビットパターンは、命令デコーダの複雑さを低減するように決定される。これに対し、提案手法では、命令 ROM の低消費電力化を目的に命令コード・ビットパターンを決定する。そのため、命令デコーダが複雑になり、消費エネルギー・オーバーヘッドが生じる可能性がある。そこで、MIPS ベース RISC プロセッサを設計し、命令デコーダにおける消費電力オーバーヘッドを評価した。本評価では、ハードウェア記述言語を用いて命令デコーダを設計し、論理合成を行った後、回路中の全ノード数(配線数)を求めた。そして、各ノードにおける負荷容量は一定(ある定数)と仮定し、従来方式における命令デコーダとの消費電力比を求めた。なお、スイッチング確率に関しては、第 5.3 で求めたバス・スイッチング確率を使用している。

実験結果を図 6 に示す。オペ・コードならびにレジスタ・フィールドに関するデコードを全てハードウェアで行った場合(図中の ENCsp と ENCall)、従来方式と比較して命令デコーダの消費電力が 15%~25% 増加する。命令デコーダは毎クロック・サイクル活性化されるため、本オーバーヘッドは無視できない。しかしながら、レジスタ・フィールドに関しては、コンパイラによるレジスタ割当アルゴリズムを拡張する事で、このようなオーバーヘッドを完全に

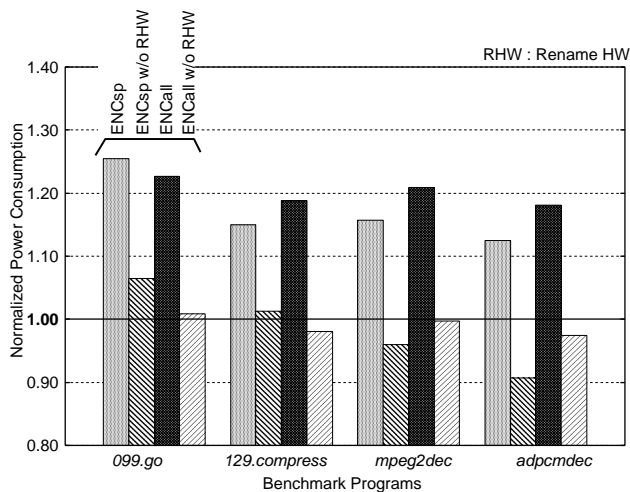


図 6: 命令デコーダにおける消費電力オーバーヘッド

隠蔽できる(つまり,参照頻度の高い変数を R31 のよに番号の大きなレジスタに割当てて)。この場合(w/o RHW と記された棒グラフ),最悪ケースでも消費電力オーバーヘッドは 8%程度(099.go)であり,その他の場合には逆に低消費電力化を達成している。これは,デコード回路の複雑さに伴う消費電力オーバーヘッドと比較して,第 5.3 節で示したように命令デコーダ入力のスイッチング回数削減による消費電力削減効果が顕著に現れたためである。

5.5 入力依存性に関する考察

これまでの評価では,プログラムに対する入力データが予め与えられており,それに基づき命令コードのビットパターンを決定した。そこで,入力データの違いが提案手法の有効性に与える影響を明らかにするため,複数の入力データを用いた実験評価を行った。

3つのベンチマーク・プログラムに関して,異なる入力データを使用した場合の実験結果を表 2 に示す。099.go ならびに 129.compress に関しては training 入力データ,また, mpeg_decode に関しては, "mei16v2.m2v" 入力データに基づき命令コード・ビットパターンを決定している。全ての結果は, BASE モデルの結果で正規化している。表 2 より,異なる入力データを用いた場合でも,40~60%のビットライン・スイッチングを削減している。これは,実行頻度の偏りが,入力データの種類ではなく,プログラムが有する特徴に大きく依存するためと考える。

6 おわりに

本稿では,命令 ROM の低消費エネルギー化を目的として,実行頻度の偏りを利用して命令コード・ビットパターン割当て手法を提案した。頻繁に実行される命令に対し,読出し時にビットライン・ディ

表 2: 異なる入力データにおける消費エネルギー

Input	HSI	CODEsp	CODEall
<i>099.go</i>			
SPEC test-input	5.03 %	51.51 %	51.43 %
<i>129.compress</i>			
SPEC test-input	39.83 %	60.79 %	60.77 %
<i>mpeg2dec</i>			
MPEG2 Verify	4.72 %	46.03 %	43.69 %
tennis	3.59 %	45.11 %	42.45 %

スタージを必要としない CB(Conforming Bit) を多く割当てることによって,低消費エネルギー化を達成できる。

ベンチマーク・プログラムを用いて評価を行った結果,命令 ROM のビットライン・スイッチング回数を最大 30%程度削減した。また,過去に提案された HSI 手法と組み合わせることで,最大 50%程度の削減効果を得ることができた。今後,命令デコーダにおける性能/消費エネルギー・オーバーヘッドを明らかにする必要がある。また,実設計に基づき,デコーダやセンスアンプ等を含んだ総合的な消費エネルギー削減効果を評価する予定である。

謝辞

日頃から御討論頂く,九州大学 大学院システム情報科学研究科 安浦寛人 教授に感謝します。なお,本研究は一部,文部省科学研究費補助金(課題番号:09358005,11308011,12358002,13308015)による。

参考文献

- [1] L. Benini, G. De Micheli, E. Macii, D. Sciuto, and C. Silvano, "Asymptotic Zero-Transition Activity Encoding for Address Buses in Low-Power Microprocessor-Based Systems," *Proc. in the Grate Lakes Symposium on VLSI*, pp. 77-82, Mar. 1997.
- [2] L. Benini, G. De Micheli, A. Macii, Enrico Macii, and M. Poncino, "Reducing Power Consumption of Dedicated Processors Through Instruction Set Encoding," *Proc. in the Grate Lakes Symposium on VLSI*, Feb. 1998.
- [3] Y. Chang, B. Park and C. Kyung, "Conforming Inverted Data Store for Low Power Memory," *Proc. of International Symposium on Low Power Electronics and Design*, pp.91-93, Aug. 1999.
- [4] Tohru Ishihara and Hiroto Yasuura, "A Power Reduction Technique with Object Code Merging for Application Specific Embedded Processors," *Proc. of Design, Automation and Test in Europe Conference 2000*, pp. 617-623, Mar. 2000.
- [5] M. R. Stan and W.P. Burleson, "Bus-Invert Coding for Low-Power I/O," *IEEE Tran. on Very Large Scale Integration (VLSI) Systems*, vol. 3., no. 1, Mar. 1995.
- [6] University of Minnesota Computer Engineering Research Group, URL: <http://www-mount.ee.umn.edu/~okeefe/mcerg/fast-dlx/>
- [7] "DLXsim - A Simulator for DLX," URL: http://hardy.ocs.mq.edu.au/mpce_courses/dlxsim/report.html
- [8] SPEC (Standard Performance Evaluation Corporation), URL: <http://www.specbench.org/osg/cpu95>.
- [9] Mediabench, URL: <http://www.cs.ucla.edu/~leec/mediabench/>.