

# 生体情報圧縮のための差分符号化命令 セット・プロセッサの提案

中塚 祥子<sup>1,a)</sup> 浜辺 崇<sup>1</sup> 坂主 圭史<sup>1</sup> 武内 良典<sup>1,b)</sup> 今井 正治<sup>1,c)</sup>

受付日 2012年11月5日, 採録日 2013年4月5日

**概要:** 本稿では, 前後の差分が小さいという特徴を持つ生体情報の圧縮のための符号化法を提案し, その符号化法を用いた低消費電力量で圧縮を行うプロセッサを提案する. 近年, 日常生活で長時間かけて行う検査のための生体情報センシング機器の開発が望まれている. 機器では通信時の消費電力量を抑えるために情報を圧縮する. 実験結果より, 提案する符号化法のための専用命令を持つ符号化プロセッサは, 生体情報の符号化処理を行った場合, (従来型の) RISC 命令のみを有するプロセッサと比較して, 消費電力量は最大 56.7%削減できることを確認できた.

**キーワード:** 低消費エネルギー技術, アーキテクチャ, プロセッサ

## Application Specific Instruction Set Processor for Delta Encoding of Biomedical Information

SHOKO NAKATSUKA<sup>1,a)</sup> TAKASHI HAMABE<sup>1</sup> KEISHI SAKANUSHI<sup>1</sup> YOSHINORI TAKEUCHI<sup>1,b)</sup>  
MASAHARU IMAI<sup>1,c)</sup>

Received: November 5, 2012, Accepted: April 5, 2013

**Abstract:** This paper proposes a coding method and an instruction set processor that utilizes characteristics of biomedical information. In recent years, biomedical information sensing system is developing for monitoring biomedical information in daily life. In these sensing systems, information is compressed for reducing energy consumption in communication. Proposed instruction set processor is designed for proposed delta encoding with low energy. Experimental results show that proposed processor achieves 56.7% energy reduction compared to a normal RISC implementation.

**Keywords:** low power, processor, architecture

### 1. はじめに

近年, 医療現場では高血圧等の慢性疾患の患者に対して, 病院で行う短時間の検査で得る生体情報だけでなく, 日常生活を送りながら長時間かけて取得しつづける生体情報も重要視されている. 病院の検査では異常が出ない場合でも, 長時間かけて取得しつづけた生体情報に異常が発見

される場合もあるからである. 日常生活での長時間での検査によってかかる患者の物理的・精神的負担軽減のため, 検査に使用する生体情報センシング機器は, 低侵襲・無拘束・無自覚である必要があり, また検査は長時間かつ実時間でできる必要がある [1]. このため, 検査機器は小型で軽量で低消費電力でなければならない. 生体情報センシング機器の開発の例としては, 膀胱内圧のデータや直腸内圧のデータ等の生体情報を測定し, 無線通信で体外の機器へ情報を送信するカプセル型の医療機器等がある [2]. 検査機器は長時間連続して生体情報を測定しつづけるので, 生体情報の量は多く, 情報を他の機器へ送信したり, センシング機器中に保存したりするためにかかる消費電力量は大き

<sup>1</sup> 大阪大学大学院情報科学研究科  
Graduate School of Information Science and Technology,  
Suita, Osaka 565-0871, Japan

a) n-shouko@ist.osaka-u.ac.jp

b) takeuchi@ist.osaka-u.ac.jp

c) imai@ist.osaka-u.ac.jp

くなる。このため、機器の電池のサイズを大きくしなければならなくなり、その場合、機器は小型化できなくなる。機器の小型化のために、通信時の消費電力量を削減することを考える [3]。そのため、送信する生体情報を圧縮して情報量を減らすことを提案する。精密な検査を行うためには、圧縮は1ビットの損失もなく復元できる可逆圧縮である必要がある。したがって生体情報の圧縮は、低消費電力量で行える可逆圧縮である必要がある。

本稿では、生体情報の圧縮のための符号化法を提案し、提案する符号化法を用いて低消費電力量で効率的に生体情報の圧縮を行うプロセッサを提案する。生体情報の中には前後の差分の値が小さいという特徴を持つものがあるので、その特徴を持つ生体情報の圧縮手法として可逆圧縮で差分の圧縮を行う符号化法を考える。また、圧縮時の消費電力量を抑えるために、符号化法のための専用命令を追加したプロセッサを提案する。専用命令を用いることで符号化が効率的に行われ、実行サイクル数が減少し、消費電力量を抑えることができる。

本稿の構成は以下のとおりである。2章では生体情報の圧縮に適した符号化法のアルゴリズムを提案する。3章では提案圧縮手法と他の圧縮手法の圧縮率を比較する。4章では、符号化処理のための専用命令を持つプロセッサを提案する。5章では、提案したプロセッサの評価実験について述べる。6章ではまとめと今後の課題について述べる。

## 2. 生体情報と可逆圧縮符号化法

本章では、生体情報と、提案する生体情報のための符号化法について述べる。

### 2.1 生体情報

生体情報は、検査機器で長時間連続して測定する。わずかな異常も逃さないように、短い間隔で測定を行う。たとえば、体内埋め込み型検査機器を用いた膀胱の内圧の測定では120 [msec] ごとに測定を行う。したがって、測定で得る生体情報の量は膨大になる。

本稿で扱う生体情報は膀胱の内圧の値である膀胱内圧のデータと直腸の内圧の値である直腸内圧のデータである。膀胱内圧のデータは膀胱の圧力値 [cmH<sub>2</sub>O] を120 [msec] ごとに測定したもので、直腸内圧のデータは直腸の圧力値 [cmH<sub>2</sub>O] を120 [msec] ごとに測定したものである。図1に膀胱内圧のデータの時間変化図を、図2に直腸内圧のデータの時間変化図を示す。

### 2.2 生体情報の符号化法

表1に、膀胱内圧と直腸内圧の前後のサンプルデータの値の差分の出現確率を示す。膀胱内圧と直腸内圧のデータは、差分の絶対値が小さいものほど高い出現確率となっている。前後の差分が小さいという特徴を持つ生体情報を圧

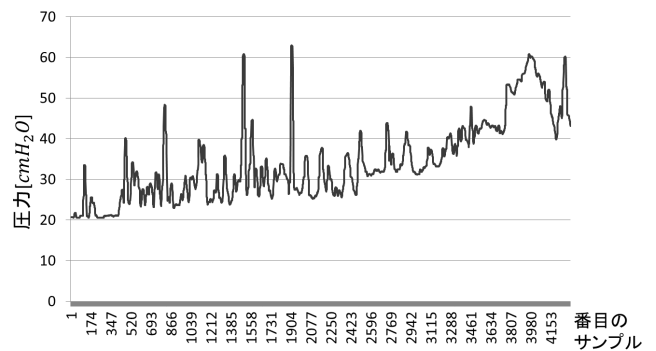


図1 膀胱内圧のデータ

Fig. 1 Data of intravesical pressure.

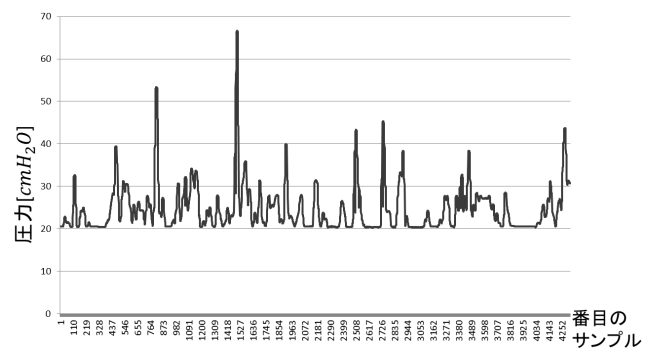


図2 直腸内圧のデータ

Fig. 2 Data of rectum pressure.

縮するため、差分を算出して符号化することを考える。生体情報を扱うため、手法は1ビットの損失もなく符号化・復号を行えなければならない。

図3に差分の符号化の概念図を示す。最初のサンプルデータはそのまゝの値を残し、2番目のサンプルデータからは1つ前のサンプルデータとの差を符号化していく。復号の場合は、最初のサンプルデータは取得したデータとし、2番目からは符号語を復号して得た差分と、1つ前に復号したサンプルデータを足してサンプルデータの値とする。

#### 2.2.1 生体情報に適した差分の符号化法

生体情報をより小さく圧縮するため、生体情報に適する差分の符号化の部分の方法を提案する。ゴロム符号 [4]、指数ゴロム符号 [5] は、データの偏りを活かした符号化法で、小さい値を短い符号語長で符号化する。これらの符号化法を基に、負数を扱えるように、そして生体情報の差分の偏りを活かして差分が小さい値に対しては短い符号語長を与えるようにして生体情報を圧縮できるような符号化法を提案する。

符号化の方法は以下のようになる。表現したい値を  $l$  とする。

(1) 以下の式を満たす  $i$  を求める。

$$-4 \leq l \leq 3 \text{ のとき}$$

$$i = 1$$

$$l < -4 \text{ のとき}$$

表 1 膀胱内圧と直腸内圧のデータの差分の出現確率 (%)

Table 1 Probability of the difference of the data of intravesical pressure and rectum pressure (%).

差分	-5	-4	-3	-2	-1	0	1	2	3	4	5	-6 以下, 6 以上
膀胱内圧 (%)	0.5	0.7	1.4	4.6	17.9	47.2	16.0	4.5	1.2	0.7	0.5	4.8
直腸内圧 (%)	0.4	0.9	1.3	4.6	15.4	50.6	14.2	4.7	1.0	0.8	0.4	5.7

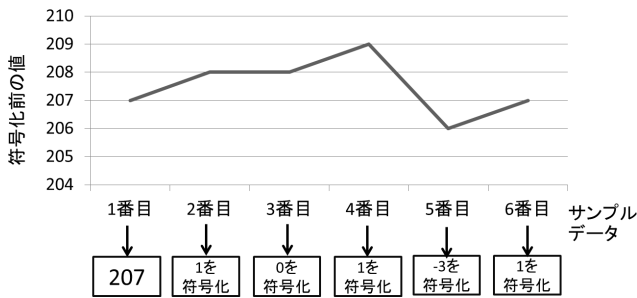


図 3 差分の符号化  
Fig. 3 Delta encoding.

表 2 提案する符号化法の符号語

Table 2 Codeword of the proposed coding method.

i	差分の範囲	表現法
1	-4~3	0***
2	-36~-5, 4~35	10*****
3	-292~-37, 36~291	110*****
4	-2340~-293, 292~2339	1110*****

$$-\sum_{j=1}^i 2^{3j-1} \leq l \leq -\sum_{j=1}^{i-1} 2^{3j-1} - 1$$

$l > 3$  のとき

$$\sum_{j=1}^{i-1} 2^{3j-1} \leq l \leq \sum_{j=1}^i 2^{3j-1} - 1$$

を満たす  $i$ .

- (2)  $(i-1)$  の数だけ 1 を並べ、次に 0 を入れて接頭語とする。
- (3) 末尾に  $3i$  ビットで  $l$  をバイナリ符号化したものを加える。

表 2 に提案する符号化法の符号語と値の関係を示す。また、復号は以下となる。復号したい符号語を  $m$  とする。

- (1) 接頭語を切り出し、1 の数を数えて、その数に 1 を足したものを  $i$  とする。
- (2) 末尾の  $3i$  ビットについて、最上位ビットが 1 のときは  $3i$  ビットで表される二進数の値に  $\sum_{j=1}^{i-1} 2^{3j-1}$  を引いた値、最上位ビットが 0 のときは  $3i$  ビットで表される二進数の値に  $\sum_{j=1}^{i-1} 2^{3j-1}$  を足した値が復号した値となる。

### 2.2.2 符号化の実装方法

本稿で扱う生体情報は 10 ビットで測定される膀胱内圧

表 3 符号語と差分の値

Table 3 Codeword and differential value.

i	差分の範囲	表現法
1	-4~3	0***
2	-36~-5, 4~35	1***0***
3	-292~-37, 36~291	1***1***0***
4	-2340~-293, 292~2339	1***1***1***0***

の値と直腸内圧の値である。10 ビットは A/D 変換機の精度で決まるシステムの制約である。したがって、実装は  $i = 4$  までとなる。また、復号時にメモリからデータを読み出す回数を減らして高速に復号を行えるようにするために、接頭語の 1 と 0 を 4 ビットごとに配置した。この結果、データは 4 ビットずつ読み出すことができる。符号語長が固定長でないために 1 度に読み出すべきビット数を決めることができない。接頭語を 4 ビットごとに配置することで、データの読み出し単位を固定化することができ、読み出し回数を削減した。たとえば  $i = 3$  の場合、'1', '1', '0' を 1 ビットずつ読み出してから 9 ビット読み出すので 4 回の読み出しを行う必要があるが、4 ビットずつの 3 回の読み出しで済むこととなる。これは、4 ビット読み出した後にその最上位ビットが 1 であるか 0 であるかを判別するからである。

表 3 に符号語と差分の値の関係を示す。たとえば、1 を符号化する場合、符号語長は 4 ビットとなる。0 ビット目から 2 ビット目を 1 の 2 進数表現である '001' とし、3 ビット目を '0' とする。よって 1 の符号語は '0001' となる。また、4 を符号化する場合は、符号語長は 8 ビットとなる。0 ビット目から 2 ビット目、4 ビット目から 6 ビット目には、符号語長が 4 ビットで表現できる範囲を除いた値を 2 進数で表現した値が入る。4 ビットで表現できるのは -4 から 3 であるので、0 を 6 ビットの 2 進数で表現した '000000' を 3 ビットずつに区切ったものとなる。そして 3 ビット目を '0'、7 ビット目を '1' とする。よって 4 の符号語は '10000000' となる。

符号化する場合は最初のサンプルデータはそのままの値で符号化し、2 番目以降のサンプルデータは直前のサンプルデータとの差分を符号化していく。図 4 に作成した符号化プログラムの流れを示す。前処理では、使用する変数の初期化と最初のサンプルデータのロード、メモリへの保存

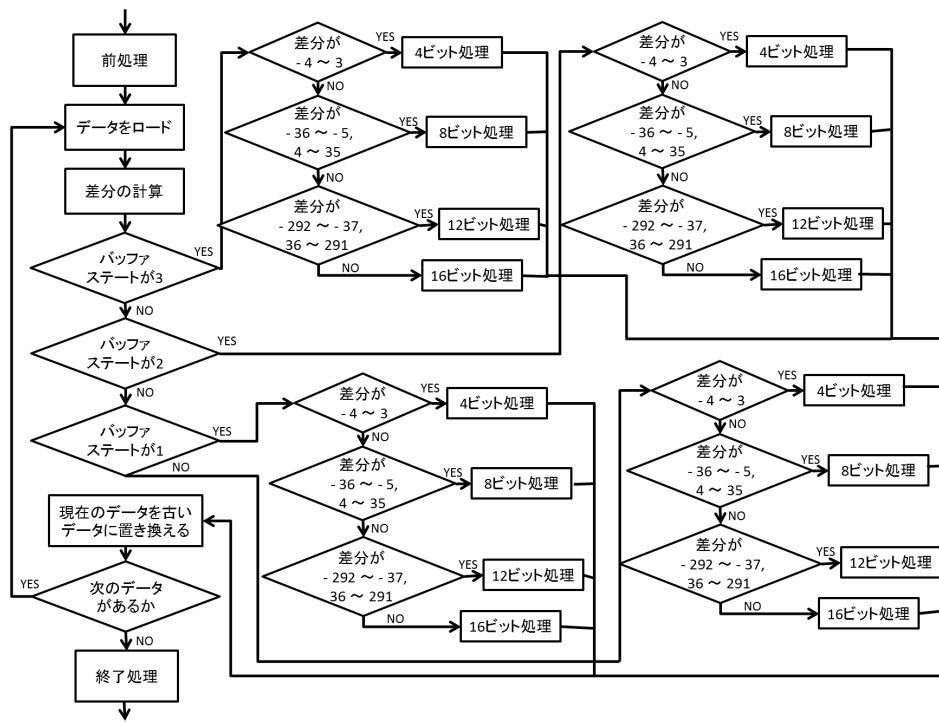


図 4 符号化の流れ

Fig. 4 Encoding flow.

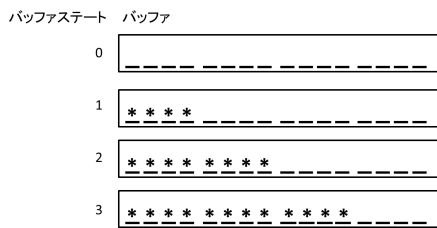


図 5 符号化時のバッファステートとバッファの状態

Fig. 5 The state of bufferstep and buffer at coding.

を行う。次に、2番目のサンプルデータをロードし、最初のサンプルデータとの差分を計算する。

本稿で扱う生体情報は 10 ビットなので、符号語長は 4 ビット、8 ビット、12 ビット、16 ビットのいずれかになる。メモリへ 16 ビットずつ保存する場合、符号語長の合計が 16 ビットになるまで符号語をためておく 16 ビットのバッファを使用する。また、バッファへ何ビット格納されているかの状態を示すバッファステートを用いる。図 5 にバッファステートとバッファ内のビット数との関係を示す。バッファステートは 4 種類のバッファの状態を示しており、バッファは符号語が合わせて 16 ビットになるまで符号語を保持しておく部分である。初期状態はバッファステートが 0 のときであり、このときバッファは空の状態である。バッファステートが 1 のときのバッファの状態は上位 4 ビットに符号語が入っている状態であり、2 のときは上位 8 ビット、3 のときは上位 12 ビット入った状態である。バッファが 16 ビット入り、満たされたときに符号語はメモリへ保存される。

バッファステートと差分をもとに必要なビット処理の場所へ分岐する。たとえば、バッファステートの値が 0 で差分の値が 2 のときはバッファステートが 0 の 4 ビット処理へ分岐する。必要なビット処理の場所へ分岐した後、差分を符号化する。

符号語の計算の後、バッファステートと符号語長に応じてバッファへの格納・メモリへの保存・バッファステートの変更を行い、ビット処理を終える。現在のサンプルデータを古いサンプルデータとし、次に読み込むべきサンプルデータがあるときはサンプルデータのロード部分へ戻り、サンプルデータがないときは終了処理へ移る。終了処理では、バッファステートが 0 以外のときにバッファに残っている符号語をメモリへ保存して終了する。

### 3. 提案符号化法と従来の符号化法との比較

提案する符号化法と、他の圧縮手法との比較を行った。可逆圧縮の必要性から、比較に用いたのはハフマン符号化 [6]、適応的ハフマン符号化 [7], [8]、算術符号化 [9]、辞書的符号化 [10] と、LZO 圧縮のサブセットである miniLZO 圧縮である。miniLZO 圧縮は生体情報センシングシステムでの情報圧縮に使われている [11]。これらはすべて可逆圧縮である。

圧縮率  $C$  [%] は、圧縮後のサイズ  $compsize$  と元のデータサイズ  $rawsize$  を用いて式 (1) で定義する。

$$C = (1 - compsize/rawsize) * 100 \quad (1)$$

圧縮率の値が大きいかほど小さく圧縮できたことを意味

表 4 膀胱内圧のデータの圧縮率 (%)

Table 4 Comparison of compression ratio of intravesical pressure data (%).

膀胱内圧 のデータ	提案手法	miniLZO	適応的			
			ハフマン	ハフマン	算術	辞書的
データ 1	57.1	16.1	-9.7	-8.0	-8.0	28.2
データ 2	54.9	-20.1	-24.7	-24.8	-24.5	-9.2
データ 3	58.2	22.9	-2.5	1.2	-0.4	35.2

表 5 直腸内圧のデータの圧縮率 (%)

Table 5 Comparison of compression ratio of rectum pressure data (%).

直腸内圧 のデータ	提案手法	miniLZO	適応的			
			ハフマン	ハフマン	算術	辞書的
データ 4	57.3	27.9	13.9	17.2	14.8	39.3
データ 5	51.6	-3.9	-9.1	-5.7	-7.3	4.9
データ 6	58.2	33.5	10.7	13.5	11.8	39.7

表 6 膀胱内圧の差分のデータの圧縮率 (%)

Table 6 Comparison of compression ratio of difference of intravesical pressure data (%).

膀胱内圧 のデータ	提案手法	miniLZO	適応的			
			ハフマン	ハフマン	算術	辞書的
データ 1	57.1	10.8	16.6	16.7	18.5	51.8
データ 2	54.9	-1.0	9.8	10.1	10.7	41.1
データ 3	58.2	22.9	20.0	20.2	21.6	55.9

表 7 直腸内圧の差分のデータの圧縮率 (%)

Table 7 Comparison of compression ratio of difference of rectum pressure data (%).

直腸内圧 のデータ	提案手法	miniLZO	適応的			
			ハフマン	ハフマン	算術	辞書的
データ 4	57.3	21.2	18.1	18.3	19.7	52.3
データ 5	51.6	-6.5	4.5	4.8	5.3	32.9
データ 6	58.2	30.0	22.0	22.2	23.5	56.3

し、圧縮率が負の値となったときは、元のデータのサイズよりも大きくなったことを意味する。

表 4 に膀胱内圧のデータを 6 種類の圧縮手法で圧縮した場合の圧縮率の比較結果を示す。どのデータにおいても、提案手法は圧縮率を 50%以上を保っている。

表 5 に直腸内圧のデータを 6 種類の圧縮手法で圧縮した場合の圧縮率の比較結果を示す。どのデータにおいても、提案手法は圧縮率を 50%以上を保っている。

miniLZO 圧縮、ハフマン符号化、適応的ハフマン符号化、算術符号化、辞書的符号化については、元のデータよりも符号化後のデータの方が大きくなっているものもある。これは、差分の出現確率は偏りが大きい、元の値そのものの出現確率は偏りが小さいからであると考えられる。以上より、提案する符号化法は差分が小さい生体情報のデータでは効果的に圧縮が可能であることが分かる。

表 6 に膀胱内圧の差分のデータに対して 6 種類の圧縮手法で圧縮した場合の圧縮率の比較結果を示す。表 4 では提案符号化のみについて差分のデータを用いたが、表 6 で

はすべての符号化について差分のデータを使用している。差分を符号化した場合、圧縮率は miniLZO 符号化では最大 22.9%、ハフマン符号化では 20.0%、適応的ハフマン符号化では 20.2%、算術符号化では 21.6%、辞書的符号化では 55.9%と向上している。どの符号化においても差分の符号化が適していることが分かる。生体情報の差分を符号化することで、データの偏りを利用してより小さく圧縮できることが分かる。

表 7 に直腸内圧の差分のデータを 6 種類の圧縮手法で圧縮した場合の圧縮率の比較結果を示す。差分を符号化した場合、miniLZO 符号化では圧縮率が小さくなったが、ハフマン符号化では圧縮率が最大で 22.0%、適応的ハフマン符号化では 22.2%、算術符号化では 23.5%、辞書的符号化では 56.3%と向上している。膀胱内圧の差分のデータの圧縮結果と同じく、どの符号化でも差分のデータを用いることで、データの偏りを利用してより小さく圧縮できることが分かる。

以上より、提案する符号は、処理対象の信号の性質であ

る前後の差分が小さいという特徴を活かした差分符号とそのデータの出現頻度の特性を利用したものであり、より小さく膀胱内圧と直腸内圧の差分のデータを圧縮できることが分かる。

#### 4. 生体情報圧縮処理向きプロセッサ・アーキテクチャ

本章では、符号化に適したプロセッサを提案する。本稿でターゲットとする生体情報センシングシステムには、全体を制御する RISC プロセッサが搭載されていることが一般的なので、その RISC プロセッサを利用して符号化を行うことを考える。しかしながら、図 4 を RISC 命令のみで実装すると、サイクル数が多くかかってしまい効率が悪い。したがって RISC プロセッサベースに、符号化のための専用命令を追加する。専用命令を用いることで符号化が効率的に行われ、実行サイクル数が減少し、消費電力量を抑えることができる。プロセッサの設計に使用したツールは ASIP Meister [12], [13] である。

##### 4.1 符号化用プロセッサの専用命令

RISC 命令に追加する符号化用の専用命令について述べる。図 6 に追加する符号化用専用命令を用いた場合の符号化の制御フローを示す。前処理で使用する変数の初期化と最初のサンプルのロード、メモリへの保存を行い、データロード命令である LDSAM 命令で値をロードし、差分を算出する。差分とバッファステートによって行われる分岐をまとめた命令である BR3 命令で符号化処理部分へ分岐する。差分の符号化を行う命令である COM4~COM16 命令で差分の符号化処理を行い、バッファへ符号語を格納する命令である RES0~RES3 命令で符号語を格納する。現

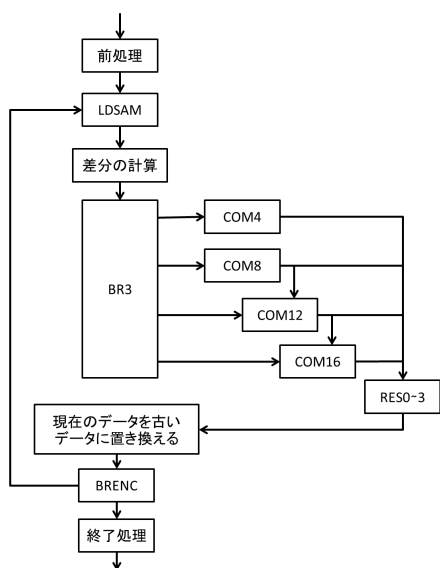


図 6 専用命令を用いた場合の符号化の制御フロー

Fig. 6 Controll flow of encoding in the case of using special instructions.

在のデータを古いデータに置き換えた後、分岐命令である BRENC 命令で次のループへ移るか、終了処理へ移る。

以下で用いている記号について、&はビットごとの論理積を、|はビットごとの論理和を、<<は指定したビット数分の左シフト演算を意味している。

##### 4.1.1 データロード命令

- LDSAM rs (LoaD SAMple)

次に符号化するサンプルデータを GPR(rs) へロードする。

##### 4.1.2 符号化用分岐命令

符号化用分岐命令は符号化に必要な分岐をまとめた命令である。

- BRENC rs (BRanch ENCode)

GPR(rs) に入る符号化すべきサンプルデータの数を基に、符号化すべきサンプルデータがあるときには前処理の直後の状態に戻り、サンプルデータがないときには終了処理部分へ移る。

- BR3 rs (BRanch 3bufferstep)

GPR(rs) に入っているバッファステートの値と、レジスタ GPR7 に格納されている差分の値に応じて 16 カ所へ分岐する。図 7 に BR3 分岐命令の動作を示す。たとえば、バッファステートの値が 3 で差分の値が 2 のときは、バッファステートが 3 のときの 4 ビット処理部分へ分岐する。

##### 4.1.3 差分符号化命令

差分符号化命令は差分を符号化するとき使用する命令である。

- COM4 rd rs (COMpress 4bit)

GPR(rs) を 4 ビットに符号化する命令。

$$GPR(rd) = GPR(rs) \& 0x0007 \text{ の計算を行う。}$$

- COM8 rd rs (COMpress 8bit)

GPR(rs) を 8 ビットに符号化するとき使用する命令。

$$GPR(rd) = (GPR(rs) \ll 1) \& 0x0070 |$$

$$GPR(rs) \& 0x0007 | 0x0080 \text{ の計算を行う。}$$

- COM12 rd rs (COMpress 12bit)

GPR(rs) を 12 ビットに符号化するとき使用する命令。

$$GPR(rd) = (GPR(rs) \ll 2) \& 0x0700 |$$

$$GPR(rs) | 0x0880 \text{ の計算を行う。}$$

- COM16 rd rs (COMpress 16bit)

GPR(rs) を 16 ビットに符号化するとき使用する命令。

$$GPR(rd) = (GPR(rs) \ll 3) \& 0x7000 |$$

$$GPR(rs) | 0x8880 \text{ の計算を行う。}$$

レジスタ GPR7 に格納されている差分を 12 ビットに符号化して得た符号語をレジスタ GPR10 に格納する動作を、専用命令を使って記述すると以下のようなになる。

```
COM8 %GPR10,%GPR7
```

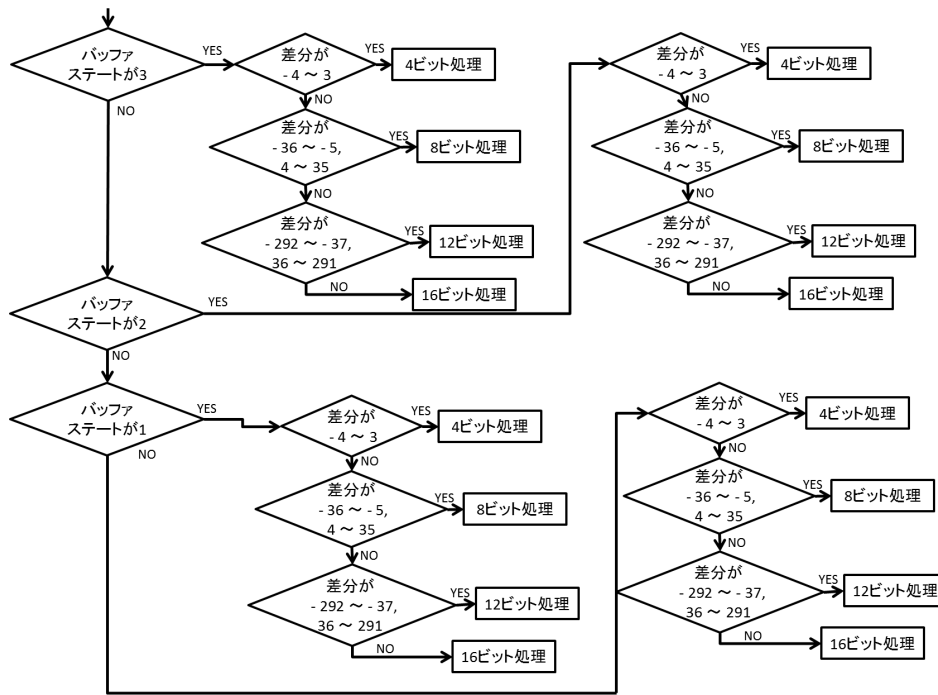


図 7 BR3 分岐命令の動作

Fig. 7 Behavior of instruction of BR3.

COM12 %GPR10,%GPR7

COM8 命令を使って差分から 12 ビットの符号の下位 8 ビットを符号化し、レジスタ GPR10 へ格納する。COM12 命令を使って、差分から 12 ビットの符号の上位 4 ビットを符号化し、レジスタ GPR10 に格納されている COM8 命令で符号化した下位 8 ビットと連結して 12 ビットの符号語を作り、レジスタ GPR10 へ格納する。

レジスタ GPR7 に入っている差分を 16 ビットに符号化して得た符号語をレジスタ GPR10 に入れる動作を、専用命令を使って記述すると以下ようになる。

COM8 %GPR10,%GPR7

COM12 %GPR10,%GPR7

COM16 %GPR10,%GPR7

4.1.4 符号格納命令

符号化したデータが 16 ビットに満たないときにバッファへ格納する命令である。rs を符号語、rd をバッファとする。

- RES3 rd rs (RE-Stuff bufferstep3)  
バッファステートが 3 のときの 4 ビット格納処理を行う。  
 $GPR(rd) = (GPR(rs) \ll 12) \& 0xf000$  の計算を行う。
- RES2 rd rs (RE-Stuff bufferstep2)  
バッファステートが 2 のときの 4 ビット格納処理を行う。  
 $GPR(rd) = ((GPR(rs) \ll 8) \& 0xf00) | GPR(rs)$  の計算を行う。
- RES1 rd rs (RE-Stuff bufferstep1)

バッファステートが 1 のときの 4 ビット格納処理を行う。

$$GPR(rd) = (GPR(rs) \ll 4) \& 0x00f0 | GPR(rs)$$

$GPR(rd)$  の計算を行う。

- RES0 rd rs (RE-Stuff bufferstep0)  
バッファステートが 0 のときの 4 ビット格納処理を行う。  
 $GPR(rd) = (GPR(rs) \& 0x000f) | GPR(rs)$  の計算を行う。

5. 評価実験

本章では、専用命令を実装したプロセッサと、ベースとなった命令拡張を行っていない RISC プロセッサ (Brownie Micro 16) との比較実験とその考察について述べる。

5.1 実験方法

符号化専用命令を実装した符号化用プロセッサとベースプロセッサ Brownie Micro 16 について、6 種類のテストデータに対して符号化を行い、実行サイクル数、消費電力量を比較した。またプロセッサの面積と消費電力と最大動作周波数の比較を行った。

Brownie Micro 16 は、ASIP Solutions 社の提供する 16 ビットの汎用 RISC プロセッサである。表 8 に Brownie Micro 16 に実装されている命令セットを示す。

プロセッサの面積と最大動作周波数は Synopsys 社の Design Compiler [14] を用いて見積もった値である。論理合成で使用したライブラリは 0.18 μm CMOS ライブラリ

表 8 Brownie Micro 16 の命令セット  
Table 8 Instruction set of Brownie Micro 16.

命令の種類	実装されている命令
算術演算	加算, 減算, 算術シフト
論理演算	論理和, 論理積, 排他的論理和, 否定, 論理シフト
比較演算	等号比較, 不等号比較, 符号無し比較
即値演算	即値加算, 即値論理シフト, 即値ロード
ビット演算	ビットセット, ビットクリア, ビットテスト
ロード, ストア	ロード, ストア
分岐演算	条件分岐
ジャンプ演算	即値ジャンプ, レジスタ間接ジャンプ
割り込み演算	割り込み発行, 割り込み復帰
特殊処理	NOP, スリープ

表 9 膀胱内圧のデータと直腸内圧のデータの圧縮率 (%)  
Table 9 Compression ratio of intravesical pressure and rectum pressure (%).

	膀胱内圧のデータ			直腸内圧のデータ		
	データ 1	データ 2	データ 3	データ 4	データ 5	データ 6
圧縮率 (%)	55.9	57.4	58.2	57.1	57.0	58.9

である。論理合成は、面積が最小になるように条件を与えて行った。実験時の動作電圧は 1.8 [V], 動作周波数は 100 [MHz] である。面積を最小にして作成したプロセッサについて、最大動作周波数についても調べた。論理合成して得た回路で符号化プログラムを Mentor 社の ModelSim [15] を用いてシミュレーションして実行サイクル数を測定し、さらに Power Compiler [14] を用いて消費電力を測定した。消費電力量は、実行サイクル数  $N$  [サイクル], 消費電力  $P$  [ $\mu$ W], 動作周波数  $f$  [MHz] を用いて式 (2) で求められる。

$$E = PN \frac{1}{f} \quad (2)$$

### 5.2 テストデータ

テストデータは、膀胱内圧のデータと直腸内圧のデータである。膀胱内圧のデータはデータ 1, データ 2, データ 3 で、サンプルデータが 1,000 個のデータである。直腸内圧のデータはデータ 4, データ 5, データ 6 で、サンプルデータが 1,000 個のデータである。表 9 に膀胱内圧と直腸内圧の各データにおける符号化の圧縮率を示す。

### 5.3 実験結果

#### 5.3.1 追加した専用命令の実行サイクル数に関する評価

提案命令の効果を示すために、表 10 に追加した専用命令を使用せずベースプロセッサにある命令のみで動作を記述した場合の命令数を示した。各専用命令を、追加命令を持たないプロセッサで実行するときの命令数は、各専用命令の効果を示すことになる。BR3 命令を除く専用命令では、従来約 4 命令で記述していた動作を 1 命令で行うことができるようになった。BR3 命令は分岐を含む命令であり、符号化を行う値やバッファステートの状態によって命

表 10 専用命令の効果

Table 10 Effect of specific instructions.

追加した専用命令	ベースプロセッサで実行するときの命令数
LDSAM	3
BRENC	3
BR3	8~79
COM4	2
COM8	8
COM12	4
COM16	4
RES3	5
RES2	5
RES1	4
RES0	3

表 11 プロセッサの設計指標

Table 11 Design qualities of processors.

プロセッサ	面積 [ $\mu$ m <sup>2</sup> ]	消費電力 [ $\mu$ W/MHz]
ベース プロセッサ	61,449	30.51
提案 プロセッサ	80,382 (+30.8%)	30.76 (+0.8%)

令数が異なる。バッファステートが 3 で差分の値が -4 から 3 の間のとき、8 命令分の動作を 1 命令で実行できる。バッファステートが 0 で差分の値が -2340 から -293, または 292 から 2339 の間のとき、79 命令分の動作を 1 命令で実行できる。

#### 5.3.2 プロセッサの面積, 消費電力, 最大動作周波数

表 11 に Brownie Micro 16 と提案プロセッサの面積と消費電力を示す。表 11 中の括弧内の数値は、Brownie Micro



表 12 膀胱内圧のデータと直腸内圧のデータの符号化処理時の実行サイクル数  
**Table 12** Execution cycles of coding of data of intravesical pressure and rectum pressure.

プロセッサ	膀胱内圧のデータ [Cycles]			直腸内圧のデータ [Cycles]		
	データ 1	データ 2	データ 3	データ 4	データ 5	データ 6
ベースプロセッサ	573,580	557,184	550,045	560,485	562,101	545,033
提案 プロセッサ	231,955 (-59.6%)	224,737 (-61.1%)	221,238 (-59.8%)	226,008 (-59.7%)	226,585 (-59.7%)	219,473 (-59.7%)

表 13 膀胱内圧のデータと直腸内圧のデータの符号化処理時の消費電力量  
**Table 13** Energy consumption of coding of data of intravesical pressure and rectum pressure.

プロセッサ	膀胱内圧のデータ [nJ]			直腸内圧のデータ [nJ]		
	データ 1	データ 2	データ 3	データ 4	データ 5	データ 6
ベースプロセッサ	23,631	23,012	22,717	23,092	23,271	22,401
提案 プロセッサ	10,345 (-56.2%)	10,046 (-56.3%)	9,867 (-56.6%)	10,057 (-56.5%)	10,128 (-56.5%)	9,701 (-56.7%)

表 14 専用命令の実行回数  
**Table 14** Frequency of specific instructions.

追加した専用命令	データ 1	データ 2	データ 3	データ 4	データ 5	データ 6
LDSAM	999 (100%)	999 (100%)	999 (100%)	999 (100%)	999 (100%)	999 (100%)
BRENC	999 (100%)	999 (100%)	999 (100%)	999 (100%)	999 (100%)	999 (100%)
BR3	999 (100%)	999 (100%)	999 (100%)	999 (100%)	999 (100%)	999 (100%)
COM4	903 (90.4%)	945 (94.6%)	969 (97.0%)	940 (94.1%)	937 (93.8%)	974 (97.5%)
COM8	94 (9.4%)	48 (4.8%)	20 (2.0%)	49 (4.9%)	51 (5.1%)	25 (2.5%)
COM12	2 (0.2%)	6 (0.6%)	10 (1.0%)	10 (1.0%)	11 (1.0%)	0 (0.0%)
COM16	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
RES3	225 (22.5%)	239 (23.9%)	245 (24.5%)	240 (24.0%)	238 (23.8%)	241 (24.1%)
RES2	229 (22.9%)	237 (23.7%)	244 (24.4%)	234 (23.4%)	238 (23.8%)	245 (24.5%)
RES1	227 (22.7%)	235 (23.5%)	240 (24.0%)	228 (22.8%)	230 (23.0%)	246 (24.6%)
RES0	222 (22.2%)	234 (23.4%)	240 (24.0%)	238 (23.8%)	231 (23.1%)	242 (24.2%)

16 を基準としたときの増減の割合を表している。面積について、Brownie Micro 16 と比べて符号化用プロセッサは 30.8%増加している。消費電力について、Brownie Micro 16 と比べて符号化用プロセッサは 0.8%増加している。また、提案プロセッサの最大動作周波数は 143 [MHz] となった。ベースプロセッサの最大動作周波数は 190 [MHz] であるので、24.7%減少している。

5.3.3 符号化処理時の実行サイクル数と消費電力量

符号化処理時の実行サイクル数と消費電力量について、

Brownie Micro 16 と符号化用プロセッサについて比較を行った。表 12 に膀胱内圧のデータと直腸内圧のデータを符号化したときの実行サイクル数を示す。表 12 中の括弧内の数値は、Brownie Micro 16 に対する増減の割合を表している。符号化用プロセッサは Brownie Micro 16 と比べて最大 61.1%実行サイクル数を削減した。

表 13 に膀胱内圧のデータと直腸内圧のデータを符号化したときの消費電力量を示す。表 13 中の括弧内の数値は、Brownie Micro 16 に対する増減の割合を表している。符号

化用プロセッサは Brownie Micro 16 と比べて最大 56.7%消費電力量を削減した。

## 5.4 考察

### 5.4.1 追加命令のサイクル数に関する評価

表 14 に、追加した専用命令が符号化プログラム実行中に何回使用されているかを示す。表 14 の括弧内の数値は、999 回圧縮を行う中で専用命令が使用される回数の割合を表している。実験で使用したデータは 1 データあたりサンプル数は 1,000 個であり、最初の値は差分符号化されないので 999 個のサンプルが差分符号化される。値をロードする LDSAM 命令、終了判定を行い分岐をする BRENC 命令、各ビットに符号化を行うための COM4~COM16 命令、バッファステートと符号語長によってバッファへの格納を行う RES3~RES0 命令は毎回実行されている。今回使用したデータでは、16 ビットに符号化される値が入っていなかったために、16 ビットに符号化するための命令である COM16 が 1 度も使用されていない。差分の偏りを利用した符号化を行っているために、一番短い 4 ビットに符号化する COM4 が他の符号化命令と比べて一番多く実行されている。COM16 以外のどの命令も、比較的多く呼び出されており、実行サイクル数の削減が効果的に行われていることが分かる。

### 5.4.2 プロセッサの面積、消費電力、最大動作周波数

プロセッサの面積は Brownie Micro 16 と比べ、30.8%増加した。消費電力においても増加した。面積と消費電力の増加は、専用命令を追加したことによるものである。また、命令を追加したことによりクリティカルパスの長さが伸びたために遅延が大きくなり、符号化用プロセッサは Brownie Micro 16 よりも最大動作周波数が減少した。

### 5.4.3 実行サイクル数と消費電力量

符号化処理時では、Brownie Micro 16 と比べ、符号化用プロセッサの実行サイクル数・消費電力量ともに大幅に削減することができた。専用命令を用いることで、実行サイクル数を削減できた。消費電力量について、消費電力は符号化用プロセッサよりも Brownie Micro 16 の方が小さいが、消費電力の差よりも実行サイクル数の差が大きいため消費電力量は実行サイクル数の影響を強く受けるので、消費電力量も大幅に削減することができた。

## 6. おわりに

本研究では、差分が小さいという特徴を持つ生体情報の圧縮手法として差分の符号化法を提案した。そして、提案した符号化法を効果的に行える専用命令を持ったプロセッサを提案した。提案したプロセッサを用いて、生体情報センシング機器と外部機器との通信時の消費電力量を削減するために、送信する生体情報に対して低消費電力での圧縮を行った。

符号化では膀胱内圧のデータ、直腸内圧のデータに対し 50%から 60%程度の圧縮ができたことを確認した。符号化用プロセッサでは、専用命令追加前のプロセッサと比較して実行サイクル数・消費電力量を 50%程度削減することができた。今後は復号処理についてのプロセッサの検討を行う。

謝辞 本研究の一部は、文部科学省「医・工・情報連携によるハイブリッド医工学産学連携拠点整備事業」および地域イノベーション戦略支援プログラム（国際競争力強化地域）「けいはんな学研都市ヘルスケア開発地域」による。

## 参考文献

- [1] 今井正治, 武内良典: 医療・ヘルスケア用生体情報センシング・システムでの無線通信の高信頼度化と低消費電力化, 2012 年電子情報通信学会総合大会, pp.SS-43-46 (Mar. 2012).
- [2] Iwato, H., Sakanushi, K., Takeuchi, Y. and Imai, M.: A Small-area and Low-power SoC for Less-invasive Pressure Sensing Capsules in Ambulatory Urodynamic Monitoring, *IEICE Trans. Electronics*, Vol.E95-C, No.4, pp.487-494 (2012).
- [3] Yu, Y., Krishnamachari, B. and Prasanna, V.K.: Data Gathering with Tunable Compression in Sensor Networks, *IEEE Trans. Parallel And Distributed Systems*, Vol.19, No.2, pp.276-287 (2008).
- [4] Golomb, S.W.: Run-Length Encodings, *IEEE Trans. Information Theory*, IT-12(3), pp.339-401 (1966).
- [5] Li, L. and Chakrabarty, K.: On Using Exponential-Golomb Codes and Subexponential Codes for System-on-a-Chip Test Data Compression, *Journal of Electronic Testing: Theory and Applications*, Vol.20, pp.667-670 (2004).
- [6] Huffman, D.A.: A Method for the Construction of Minimum-Redundancy Codes, *Proc. I.R.E.*, pp.1098-1101 (Sep. 1952).
- [7] Marcelloni, F. and Vecchio, M.: A Simple Algorithm for Data Compression in Wireless Sensor Networks, *IEEE Communications Letters*, Vol.12, No.6, pp.411-413 (2008).
- [8] Vitter, J.S.: Design and Analysis of Dynamic Huffman Codes, *Journal of the Association for Computing Machinery*, Vol.34, No.4, pp.825-845 (1987).
- [9] Witten, I.H., Neal, R.M. and Cleary, J.G.: Arithmetic Coding For Data Compression, *Computing Practices, Comm. ACM*, Vol.30, No.6, pp.520-540 (1987).
- [10] Kraus, J., Tobiska, T. and Bubla, V.: Looseless Encodings And Compression Algorithms Applied On Power Quality Datasets, *CIREC 20th International Conference on Electricity Distribution*, Paper 0987, Prague, pp.8-11 (June 2009).
- [11] Oberhumer.com GmbH, LZ0 Professional data compression library (online) (Nov. 2012), available from (<http://www.oberhumer.com/>).
- [12] 今井正治, 武内良典: コンフィギュラブル・プロセッサ生成技術の現状と今後の展望, 第 17 回 回路とシステム軽井沢ワークショップ, pp.541-548 (2004).
- [13] 今井正治, 武内良典, 塩見彰睦, 佐藤 淳, 北嶋 暁: 特定用途向きプロセッサ開発システム ASIP Meister, 電子情報通信学会技術研究報告 ICD, Vol.102, No.401, pp.39-44 (2002).
- [14] Synopsys, Inc., available from

[http://www.synopsys.co.jp/products/dc\\_ultra/detail.html](http://www.synopsys.co.jp/products/dc_ultra/detail.html).

- [15] Mentor Graphics Japan Co. Ltd., available from <http://www.mentorg.co.jp/>.



中塚 祥子

2012年大阪大学基礎工学部情報科学科卒業。現在、同大学院修士課程在学中。生体情報センシングシステムに興味を持つ。



浜辺 崇 (正会員)

2004年大阪大学基礎工学部情報科学科卒業。2006年同大学院博士前期課程修了。2012年同大学院博士後期課程修了。現在、マイクロニクス株式会社勤務。システムレベル設計手法、VLSI設計自動化に関する研究に従事。



坂主 圭史 (正会員)

1997年東京工業大学工学部電気・電子工学科卒業。2002年同大学院理工学研究科電気・電子工学専攻博士後期課程修了。博士(工学)。2002年大阪大学大学院情報科学研究科助手。2007年同大学院助教。2012年より、株式会社東芝にて半導体リソグラフィのCAD技術開発に従事。



武内 良典 (正会員)

1987年東京工業大学工学部電気・電子工学科卒業。1992年同大学院博士後期課程修了。博士(工学)。1996年大阪大学大学院基礎工学研究科情報数理学専攻講師。現在、同大学院情報科学研究科准教授。VLSI設計およびVLSI CADの研究に従事。IEEE、電子情報通信学会各会員。



今井 正治 (正会員)

1974年名古屋大学工学部電気工学科卒業。1979年同大学院博士後期課程修了(工学博士)。同年豊橋技術科学大学奉職。1994年同教授。1996年大阪大学大学院基礎工学研究科情報数理学専攻教授。その間、1984年から1985年にかけて米国サウスカロライナ大学工学部電気計算機工学科客員助教授(文部省在外研究員)。これまで組合せ最適化アルゴリズム、ハードウェア/ソフトウェア協調設計等の研究に従事。1991年より日本電子機械工業会およびIEEE/DASCにおいてEDA標準化作業に従事。現在、情報処理学会設計自動化研究会主査。IEEE、ACM、電子情報通信学会、人工知能学会各会員。