

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日  
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

## ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。  
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）  
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

# ユーザーズ・マニュアル

## μSAP703000-B04

## μSAP70732-B04

### ADPCM ミドルウェア

---

#### 対象デバイス

μSAP703000-B04 : V850 ファミリ™

μSAP70732-B04 : V810 ファミリ™

〔メモ〕

# 目次要約

第1章	概 説	...	15
第2章	ライブラリ仕様	...	23
第3章	インストレーション	...	45
付 録	sample.c のソース・プログラム	...	51

V810 ファミリ , V850 ファミリ , V821 , V850E/IA1 , V850E/MA1 , V850E/MS1 , V850/SA1 , V850/SB1 , V850/SB2 , V850/SF1 , V850/SV1 , V853 は , 日本電気株式会社の商標です。

Green Hills Software は , 米国 Green Hills Software, Inc.の商標です。

PC DOS は , 米国 IBM Corp.の商標です。

Windows, MS-DOS は米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

UNIX は X/Open カンパニーリミテッドがライセンスしている米国ならびに他の国における登録商標です。

SUN4 は米国 Sun Microsystems, Inc.の商標です。

- **本資料の内容は予告なく変更することがありますので、最新のものであることをご確認の上ご使用ください。**
- 文書による当社の承諾なしに本資料の転載複製を禁じます。
- 本資料に記載された製品の使用もしくは本資料に記載の情報の使用に際して、当社は当社もしくは第三者の知的財産権その他の権利に対する保証または実施権の許諾を行うものではありません。上記使用に起因する第三者所有の権利にかかわる問題が発生した場合、当社はその責を負うものではありませんのでご了承ください。
- 本資料に記載された回路、ソフトウェア、及びこれらに付随する情報は、半導体製品の動作例、応用例を説明するためのものです。従って、これら回路・ソフトウェア・情報をお客様の機器に使用される場合には、お客様の責任において機器設計をしてください。これらの使用に起因するお客様もしくは第三者の損害に対して、当社は一切その責を負いません。

M7A 98.8

## 本版で改訂された主な箇所

箇所	内容
p.20	1.3.2 (3)サポート・ツールを変更
p.21	1.3.4 ディレクトリ構成を変更
p.45	3.1.1 V810 ファミリ版を変更
p.45	3.1.2 V850 ファミリ版を変更
p.46	3.1.3 ディレクトリとファイルを変更
p.47	3.2.1 UNIX 版を変更
p.48	3.2.2 MS-DOS/PC DOS 版を変更
p.49,50	3.3 サンプル・プログラムの作成を変更
p.51	付録 sample.c のソース・プログラムを変更

本文欄外の 印は、本版で改訂された主な箇所を示しています。

巻末にアンケート・コーナーを設けております。このドキュメントに対するご意見をお気軽にお寄せください。

# はじめに

**対象者** このマニュアルは、V810 ファミリ / V850 ファミリの応用システムを設計、開発するユーザを対象としています。

**目的** V810 ファミリ / V850 ファミリ用の応用、開発する際にサポートするミドルウェアを、ユーザに理解していただくことを目的としています。

**構成** このマニュアルは、大きく分けて次の内容で構成しています。

- ・概 説
- ・ライブラリ仕様
- ・インストレーション
- ・付 録

**読み方** このマニュアルの読者には、電気、論理回路、マイクロコンピュータおよび C 言語に関する一般知識を必要とします。

V810 ファミリ / V850 ファミリのハードウェア機能を知りたいとき

**各製品のユーザズ・マニュアル ハードウェア編**を参照してください。

V810 ファミリ / V850 ファミリの命令機能を知りたいとき

**各製品のユーザズ・マニュアル アーキテクチャ編**を参照してください。

**凡 例** データ表記の重み：左が上位桁，右が下位桁

アクティブ・ロウの表記： $\overline{\text{xxx}}$  (端子，信号名称に上線)

メモリ・マップのアドレス：上部 - 上位，下部 - 下位

注：本文中に付けた注の説明

注意：気を付けて読んでいただきたい内容

備考：本文の補足説明

数の表記：2 進数...xxxx または xxxxB

10 進数...xxxx

16 進数...xxxxH または 0x xxxx

2 のべき数を示す接頭語 (アドレス空間，メモリ容量)：

K(キ口)： $2^{10} = 1024$

M(メガ)： $2^{20} = 1024^2$

G(ギガ)： $2^{30} = 1024^3$

**関連資料** 関連資料は暫定版の場合がありますが、この資料では「暫定」の表示をしておりません。あらかじめご了承ください。

**V810 ファミリに関する資料**

製品名		データ・シート	ユーザーズ・マニュアル	
愛称	品名		ハードウェア編	アーキテクチャ編
V821™	μPD70741	U11678J	U10077J	

**V850 ファミリに関する資料**

品名	資料名	データ・シート	ユーザーズ・マニュアル	
			ハードウェア編	アーキテクチャ編
V853™	μ PD703003A,703004A, 703025A	U13188J	U10913J	U10243J
	μ PD70F3003A, 70F3025A	U13189J		
V850/SA1™	μ PD703014A,703014AY,703015A, 703015AY,703017A,703017AY	U14526J	U12768J	
	μ PD70F3017A,70F3017AY	U14527J		
V850/SB1™	μ PD703031A,703031AY, 703033A, 703033AY, 70F3033A,70F3033AY	U14734J	U13850J	
	μ PD703032A,703032AY,70F3032A,70F3032AY	U14893J		
V850/SB2™	μ PD703034A, 703034AY,703035A, 703035AY, 70F3035A,70F3035AY	U14780J		
	μ PD703037A,703037AY,70F3037A,70F3037AY	U14894J		
V850/SF1™	μ PD703078Y,703079Y,70F3079Y	作成予定	U14665J	
V850/SV1™	μ PD703039,703039Y,703040,703040Y, 703041,703041Y	U13953J	U14462J	
	μ PD70F3040,70F3040Y,	U14662J		
V850E/MS1™	μ PD703100-33,703100-40, 703101-33,703102-33	U13995J	U12688J	U12197J
	μ PD703100-A33,703100-A40, 703101-A33,703102-A33	U14168J		
	μ PD70F3102-33	U13844J		
	μ PD70F3102-A33	U13845J		
V850E/MA1™	μ PD703103,703105,703106,703107	作成予定	U14359J	U14559J
	μ PD70F3107	U14618J		
V850E/IA1™	μ PD703117,70F3116	作成予定	U14492J	

### V810 ファミリ開発ツールに関する資料 (ユーザーズ・マニュアル)

資料名		資料番号
CA732 (C コンパイラ)	操作編 (UNIX™ ベース)	U11013J
	操作編 (Windows™ ベース)	U11068J
	アセンブリ言語編	U11016J
	C 言語編	U11010J
RX732 (リアルタイム OS)	基礎編	U10346J
	ニュークリアス・インストレーション編	U10347J
	テクニカル編	U10490J

### V850 ファミリ開発ツールに関する資料 (ユーザーズ・マニュアル)

資料名		資料番号
IE-703002-MC (V853, V850/SA1, V850/SB1, V850/SB2, V850/SF1, V850/SV1, V850E/MS1 用インサーキット・エミュレータ)		U11595J
IE-V850E-MC (V850E/IA1 用インサーキット・エミュレータ), IE-V850E-MC-A (V850E/MA1 用インサーキット・エミュレータ)		U14487J
IE-703003-MC-EM1 (V853 用周辺 I/O ボード)		U11596J
IE-703017-MC-EM1 (V850/SA1 用周辺 I/O ボード)		U12898J
IE-703037-MC-EM1 (V850/SB1, V850/SB2 用周辺 I/O ボード)		U14151J
IE-703040-MC-EM1 (V850/SV1 用周辺 I/O ボード)		U14337J
IE-703102-MC (V850E/MS1 用周辺 I/O ボード)		U13875J
IE-703102-MC-EM1, IE-703102-MC-EM1-A (V850E/MS1 用周辺 I/O ボード)		U13876J
IE-703107-MC-EM1 (V850E/MA1 用周辺 I/O ボード)		U14481J
IE-703117-MC-EM1 (V850E/IA1 用周辺 I/O ボード)		U14770J
CA850 (Ver.2.30 以上) (C コンパイラ・パッケージ)	操作編	U14568J
	C 言語編	U14566J
	プロジェクト・マネージャ編	U14569J
	アセンブリ言語編	U14567J
ID850 (Ver.2.20 以上) (統合ディバッガ)	操作編 Windows ベース	U14580J
SM850 (Ver.2.20 以上) (システム・シミュレータ)	操作編 Windows ベース	U14782J
RX850 (リアルタイム OS)	基礎編	U13430J
	インストレーション編	U13410J
	テクニカル編	U13431J
RX850 Pro (リアルタイム OS)	基礎編	U13773J
	インストレーション編	U13774J
	テクニカル編	U13772J
RD850 (タスク・ディバッガ)		U13737J
RD850 Pro (タスク・ディバッガ)		U13916J
AZ850 (システム・パフォーマンス・アナライザ)		U14410J
PG-FP3 (フラッシュ・メモリ・プログラマ)		U13502J

### Green Hills Software™, Inc. (GHS 社) 製ツールに関する資料

GHS 社製ツールは、日本国内では下記で取り扱っております。各種製品とそれに関する資料については、下記へお問い合わせください。

(株)アドバンスド データ コントロールズ TEL (03)3576-5351

【メモ】

# 目 次

<b>第1章 概 説</b>	...	15
1.1 ミドルウェア	...	15
1.2 ADPCM	...	15
1.2.1 PCM リニア変換部	...	17
1.2.2 ADPCM 圧縮部	...	17
1.2.3 ADPCM 伸長部	...	18
1.2.4 リニア PCM 変換部	...	18
1.2.5 同期符号化補正部	...	18
1.3 製品概要	...	19
1.3.1 特 徴	...	19
1.3.2 動作環境	...	19
1.3.3 性 能	...	20
1.3.4 ディレクトリ構成	...	21
<b>第2章 ライブラリ仕様</b>	...	23
2.1 機 能	...	23
2.1.1 圧縮処理	...	23
2.1.2 伸長処理	...	24
2.2 RAM	...	25
2.3 データ型	...	25
2.3.1 int pcm_mlaw	...	25
2.3.2 int pcm_alaw	...	25
2.3.3 int linear_enc	...	26
2.3.4 int linear_dec	...	26
2.3.5 int adpcm_32kbps	...	26
2.3.6 int adpcm_16kbps	...	26
2.4 エラー処理	...	27
2.5 関数仕様	...	27
2.5.1 初期化関数	...	27
2.5.2 圧縮系関数	...	28
2.5.3 伸長系関数	...	34

<b>第3章</b>	<b>インストール</b>	<b>...</b>	<b>45</b>
<b>3.1</b>	<b>提供形態</b>	<b>...</b>	<b>45</b>
3.1.1	V810 ファミリ版	...	45
3.1.2	V850 ファミリ版	...	45
3.1.3	ディレクトリとファイル	...	46
<b>3.2</b>	<b>ホスト・マシンへのファイル展開</b>	<b>...</b>	<b>47</b>
3.2.1	UNIX 版	...	47
3.2.2	MS-DOS/PC DOS 版	...	48
<b>3.3</b>	<b>サンプル・プログラムの作成</b>	<b>...</b>	<b>49</b>
<b>3.4</b>	<b>ロケーションの変更</b>	<b>...</b>	<b>50</b>
<b>3.5</b>	<b>シンボル名規約</b>	<b>...</b>	<b>50</b>
<b>付 録</b>	<b>sample.c のソース・プログラム</b>	<b>...</b>	<b>51</b>

# 図の目次

図番号	タイトル, ページ
1 - 1	勧告 G.726 ... 15
1 - 2	ADPCM 概略図 ... 16
1 - 3	ADPCM 処理構成 ... 16
1 - 4	ADPCM 圧縮部の構成 ... 17
1 - 5	ADPCM 伸長部の構成 ... 18
2 - 1	圧縮処理フロー ... 23
2 - 2	伸長処理フロー ... 24
3 - 1	サンプル・プログラム・データの構成 ... 49

# 表の目次

表番号	タイトル, ページ
1 - 1	V810 性能一覧 ... 20
1 - 2	V853 性能一覧 ... 20
2 - 1	圧縮系関数一覧 ... 23
2 - 2	伸長系関数一覧 ... 24

# 第 1 章 概 説

この章では、ミドルウェアと ADPCM について説明します。

## 1.1 ミドルウェア

ミドルウェアは、プロセッサの性能を最大限に引き出すようにチューニングされたソフトウェア群です。

現在、高性能 RISC プロセッサが比較的安く市場に投入され、従来専用ハードウェアに頼っていた処理を「高性能 RISC プロセッサ」+「ソフトウェア」というアプローチで実現できるようになりました。この「ソフトウェア」をミドルウェアと呼んでいます。

NEC では、ヒューマン・マシン・インタフェースおよび信号処理技術をミドルウェアの形で用意しています。さまざまなユーザのニーズに対応して、優れたシステム・ソリューションを提供しています。

ADPCM ミドルウェアは、音声コード（電話品質）の圧縮 / 伸長を提供するライブラリです。

**備考** RISC : Reduced Instruction Set Computer

## 1.2 ADPCM

このミドルウェアの ADPCM は、ITU-T の勧告 G.726 により定められたアナログ電話帯域（0.3 ~ 3.4kHz）の音声を符号化する方式です。

勧告 G.726 は、1988 年に電話品質符号化の国際標準規格として勧告された G.721（32 kbps ADPCM）に、40 kbps ADPCM、24 kbps ADPCM、16 kbps ADPCM を追加して、1990 年に再び勧告したものです。

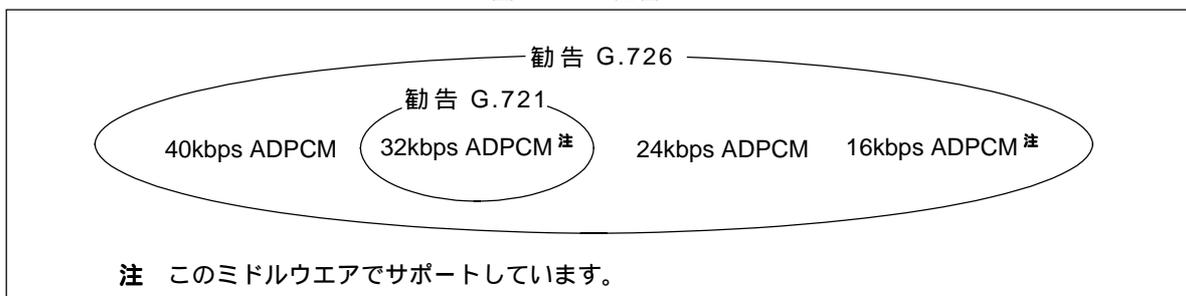
このミドルウェアでは、勧告 G.726 のうち、32 kbps ADPCM と 16 kbps ADPCM をサポートしています。

**備考** ADPCM : Adaptive Differential Pulse Code Modulation（適応差分パルス符号変調方式）

ITU-T : International Telecommunication Union-Telecommunication standardization sector

（国際電気通信連合-電気通信標準化部門）

図 1 - 1 勧告 G.726

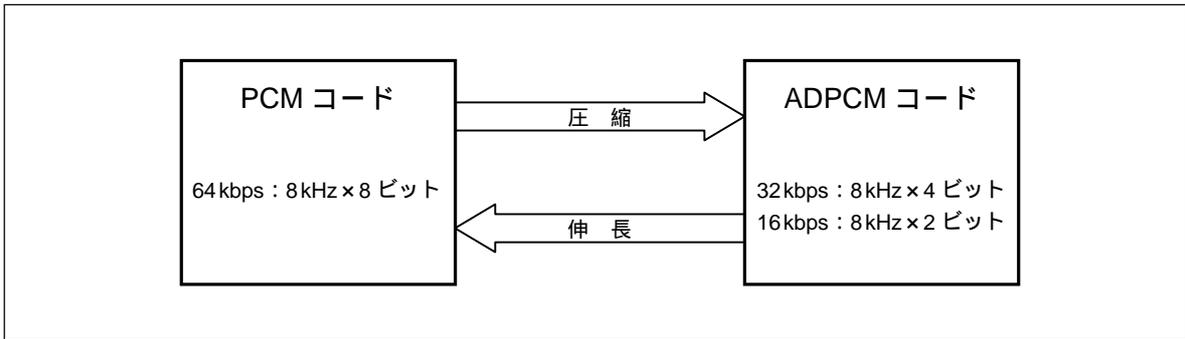


ADPCM は、ITU-T 勧告 G.711 で規定された 64kbps PCM コードがインタフェースの基本になっています。これは、電話品質の符号化方式として先に勧告 G.721 の 64kbps PCM が国際標準規格として広く適用されているためです。また、ADPCM がさらに圧縮率を高めるための標準規格として開発されたためです。

ITU-T 勧告 G.711 の 64kbps PCM は、帯域通過フィルタによって、0.3～3.4kHz に帯域制限したあと、8kHz でサンプリングしたアナログ音声信号を非線形量子化法により、8 ビットのデジタル信号に符号化します。64kbps PCM には、 $\mu$ -law方式（日本および北米で使用）と A-law方式（ヨーロッパで使用）の 2 つの方式があります。

ADPCM は、64kbps PCM で圧縮した音声コードをさらに圧縮するための標準化です。

図 1 - 2 ADPCM 概略図

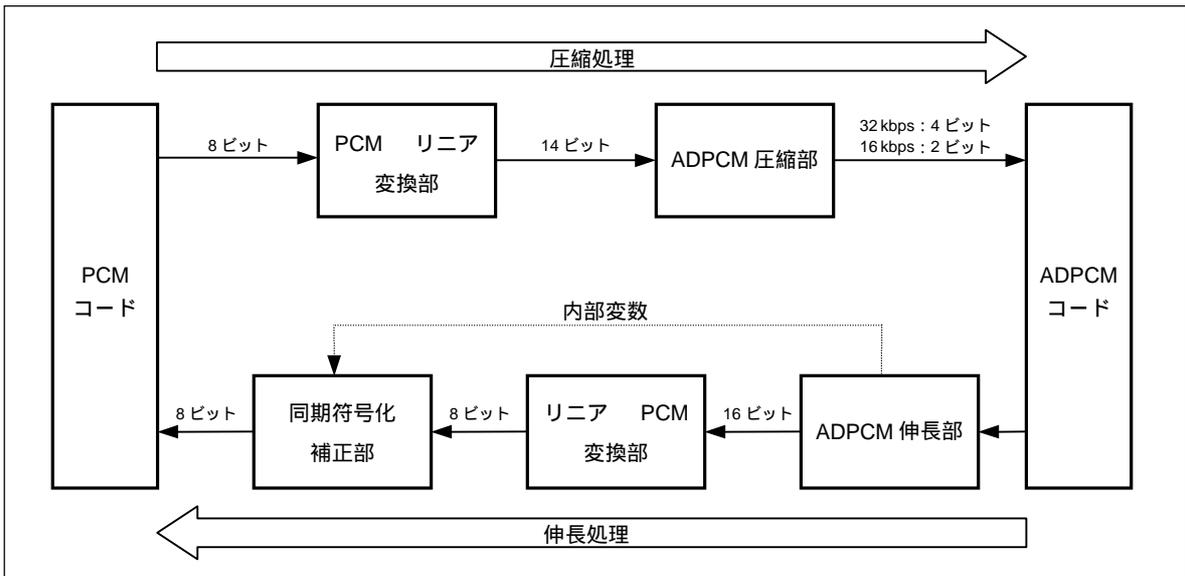


次に ADPCM の圧縮/伸長処理について説明します。

ADPCM の圧縮処理は、図 1-3 に示すように PCM リニア変換部と ADPCM 圧縮部で構成されています。

ADPCM の伸長処理は、図 1-3 に示すように ADPCM 伸長部、リニア PCM 変換部、同期符号化補正部で構成されています。伸長処理では、同期符号化補正部によって、既存のデジタル（64kbps PCM）通信網に組み込んだ場合の圧縮と伸長の繰り返しによる音質の劣化を防止します。

図 1 - 3 ADPCM 処理構成



### 1.2.1 PCM リニア変換部

64kbps PCM コード (8 ビット) をリニア・コード (14 ビット : 2 の補数表現) に変換します。

$\mu$ -law方式の PCM コードは、勧告 G.711 の PCM 復号器で変換します。

A-law方式の PCM コードは、勧告 G.711 の PCM 復号器で変換したあと、値を 2 倍して 14 ビットに調整します。

### 1.2.2 ADPCM 圧縮部

リニア・コード (14 ビット) を ADPCM コード (32kbps : 4 ビット, 16kbps : 2 ビット) に圧縮します。圧縮処理の手順を次に示します (図 1-4 参照)。

入力信号 (リニア・コード) から予測信号を引いて差分信号を算出します。

差分信号を適応量子化器で符号化します。

32kbps ADPCM では 15 レベルの適応量子化器を使用して、4 ビットに符号化します。

16kbps ADPCM では 4 レベルの適応量子化器を使用して、2 ビットに符号化します。

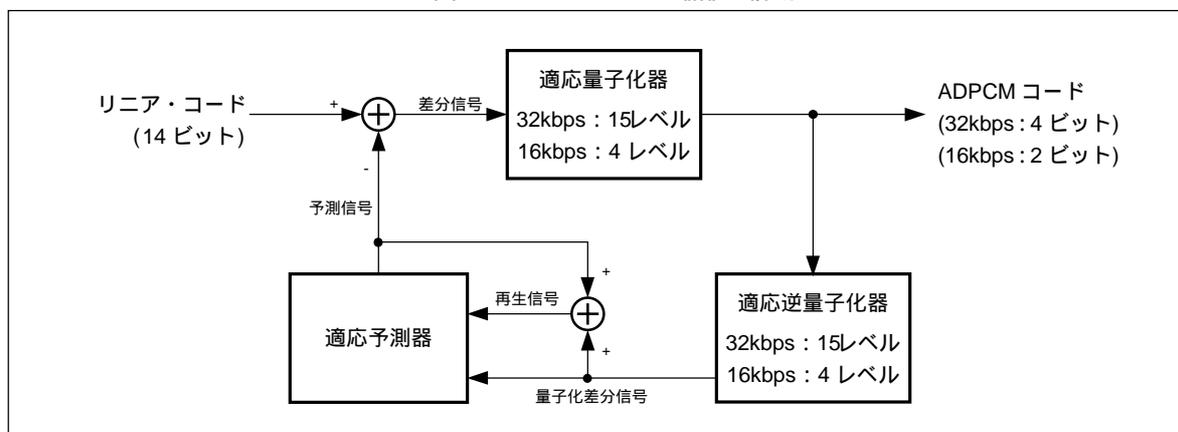
この符号化した信号が ADPCM コードで圧縮処理の出力信号になります。

適応逆量子化器で、ADPCM コードから量子化差分信号を生成します。

予測信号に量子化差分信号を加えて、入力信号を再生します。

適応予測器で、再生信号と量子化差分信号から次の入力信号の予測値を算出します。

図 1 - 4 ADPCM 圧縮部の構成



### 1.2.3 ADPCM 伸長部

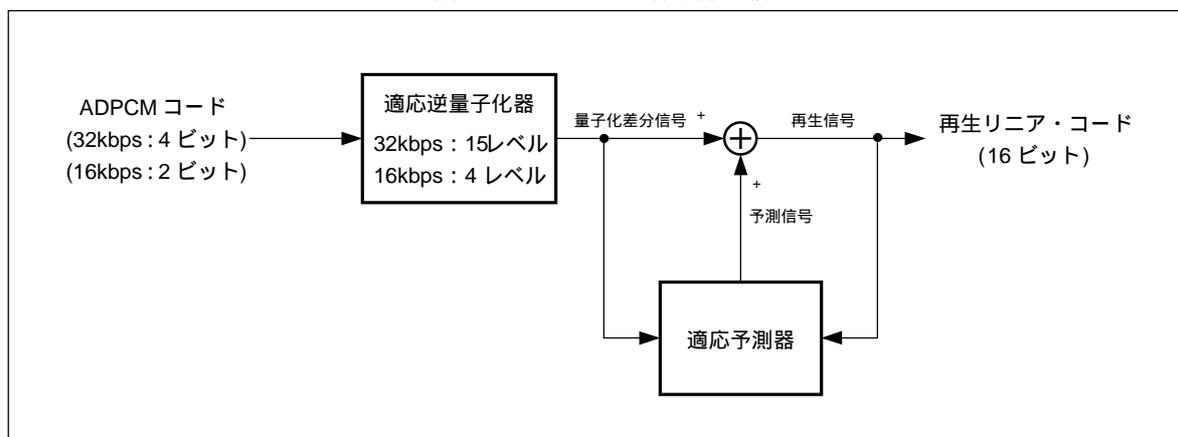
ADPCM コード (32kbps : 4 ビット, 16kbps : 2 ビット) を再生リニア・コード (16 ビット) に伸長します。伸長処理の手順を次に示します (図 1-5 参照)。

適応逆量子化器で, ADPCM コードから量子化差分信号を生成します。

予測信号に量子化差分信号を加えて, 入力信号を再生します。この信号が ADPCM 伸長処理の出力信号になります。

適応予測器で, 再生信号と量子化差分信号から次の入力信号の予測値を算出します。

図 1 - 5 ADPCM 伸長部の構成



### 1.2.4 リニア PCM 変換部

再生リニア・コード (16 ビット) を 64kbps PCM コード (8 ビット) に変換します。

$\mu$ -law方式の PCM コードは, 勧告 G.711 の PCM 符号器で変換します。

A-law方式の PCM コードは, 再生リニア・コードを調整 (1/2) してから, 勧告 G.711 の PCM 符号器で変換します。

### 1.2.5 同期符号化補正部

同期符号化補正は, PCM から ADPCM への圧縮と ADPCM から PCM への伸長を繰り返す (同期タンデム符号化) ときに発生する累積的なゆがみを補正します。

入力信号である再生リニア・コードを, 1.2.1 PCM リニア変換部と 1.2.2 ADPCM 圧縮部の , の手順で, 次に圧縮する場合の ADPCM コード値をシミュレートして, 現在の ADPCM コードと同じになるように再生リニア・コードの値を補正します。

## 1.3 製品概要

### 1.3.1 特 徴

ADPCM の国際標準規格 (ITU-T 勧告 G.726) に準拠しています。

#### (1) ビット・レート

次の 2 つのビット・レートをサポートしています。

- ・ 32kbps ADPCM (1 サンプル・データを 4 ビットに圧縮)
- ・ 16kbps ADPCM (1 サンプル・データを 2 ビットに圧縮)

#### (2) 64kbps PCM コード・インタフェース

64kbps PCM の国際標準規格 (ITU-T 勧告 G.711) に準拠した次の 2 つの方式をサポートしています。

- ・  $\mu$ -law 方式
- ・ A-law 方式

また、伸長で同期タンデム符号化をしない場合は、ITU-T 勧告 G.726 から同期符号化補正の機能を削除して実行速度を速くしたのもサポートしています。

#### (3) リニア・コード・インタフェース

ADPCM の国際標準規格 (ITU-T 勧告 G.726) から、64kbps PCM コード・インタフェースにかかわる省略可能な機能 (64kbps PCM コードとリニア・コードの相互変換、同期符号化補正) を削除したリニア・コード・インタフェース (14 ビット精度) をサポートしています。

**備考** リニア・コード・インタフェースは、同期タンデム符号化に対応していません。

### 1.3.2 動作環境

#### (1) 対象 CPU

- ・ V810 ファミリ
- ・ V850 ファミリ

#### (2) 必要メモリ

- ・ ROM : 総容量 9K バイト以下 (リンクする関数により容量は変化します)
- ・ RAM : 80 バイト以下

( 3 ) サポート・ツール

( a ) V810 ファミリ

- ・ NEC 製 C コンパイラ・パッケージ : CA732 ( Windows3.1 版以降 , SUN4™ 版 )
- ・ GHS 社製 C コンパイラ/アセンブラ : C-V810 ( Windows3.1 版以降 , SUN4 版 )

( b ) V850 ファミリ

- ・ NEC 製 C コンパイラ・パッケージ : CA850 ( Windows3.1 版以降 , SUN4 版 )
- ・ GHS 社製 C コンパイラ/アセンブラ : C-V850 ( Windows3.1 版以降 , SUN4 版 )

1.3.3 性 能

1 サンプルング・データ当たりの処理時間を次に示します。

( 1 ) V810

【条件】 25MHz , 32 ビット・バス , キャッシュ OFF , 0 ウェイト ROM/RAM

表 1 - 1 V810 性能一覧

圧縮 / 伸長		処理時間
圧縮	リニア・コード・インタフェース	62 μs 以下
	PCM コード・インタフェース	63 μs 以下
伸長	リニア・コード・インタフェース ( 同期タンデム符号化非対応 )	59 μs 以下
	PCM コード・インタフェース ( 同期タンデム符号化非対応 )	61 μs 以下
	PCM コード・インタフェース	66 μs 以下

( 2 ) V853

【条件】 33MHz , 内部 ROM/RAM , 0 ウェイト外部 RAM ( 音声データ )

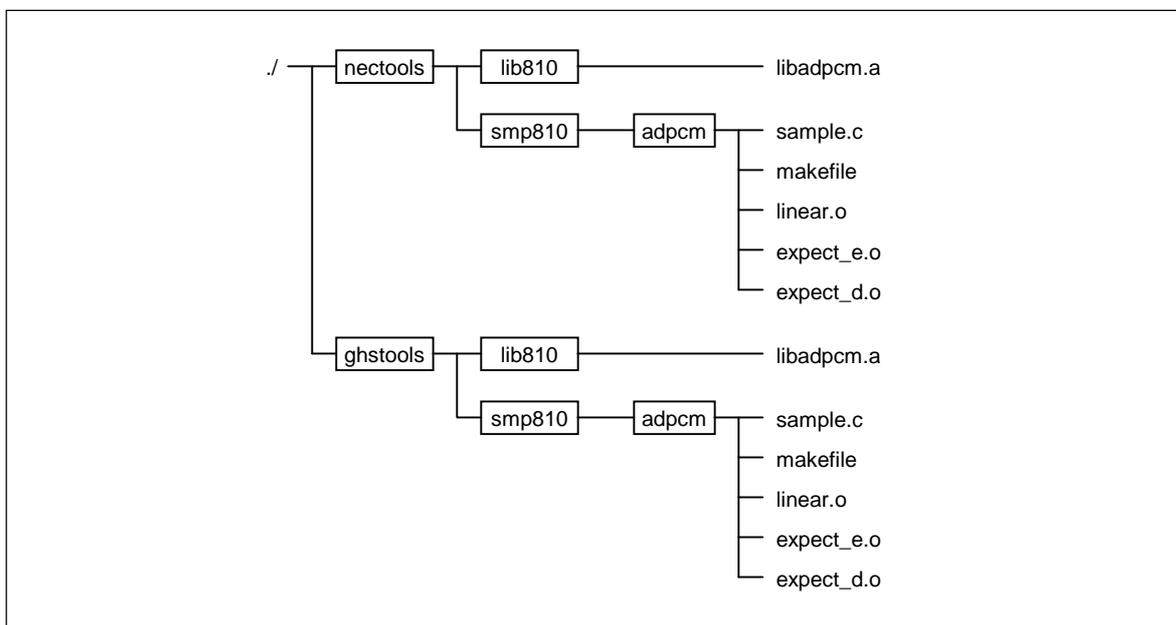
表 1 - 2 V853 性能一覧

圧縮 / 伸長		処理時間
圧縮	リニア・コード・インタフェース	29 μs 以下
	PCM コード・インタフェース	30 μs 以下
伸長	リニア・コード・インタフェース ( 同期タンデム符号化非対応 )	28 μs 以下
	PCM コード・インタフェース ( 同期タンデム符号化非対応 )	29 μs 以下
	PCM コード・インタフェース	31 μs 以下

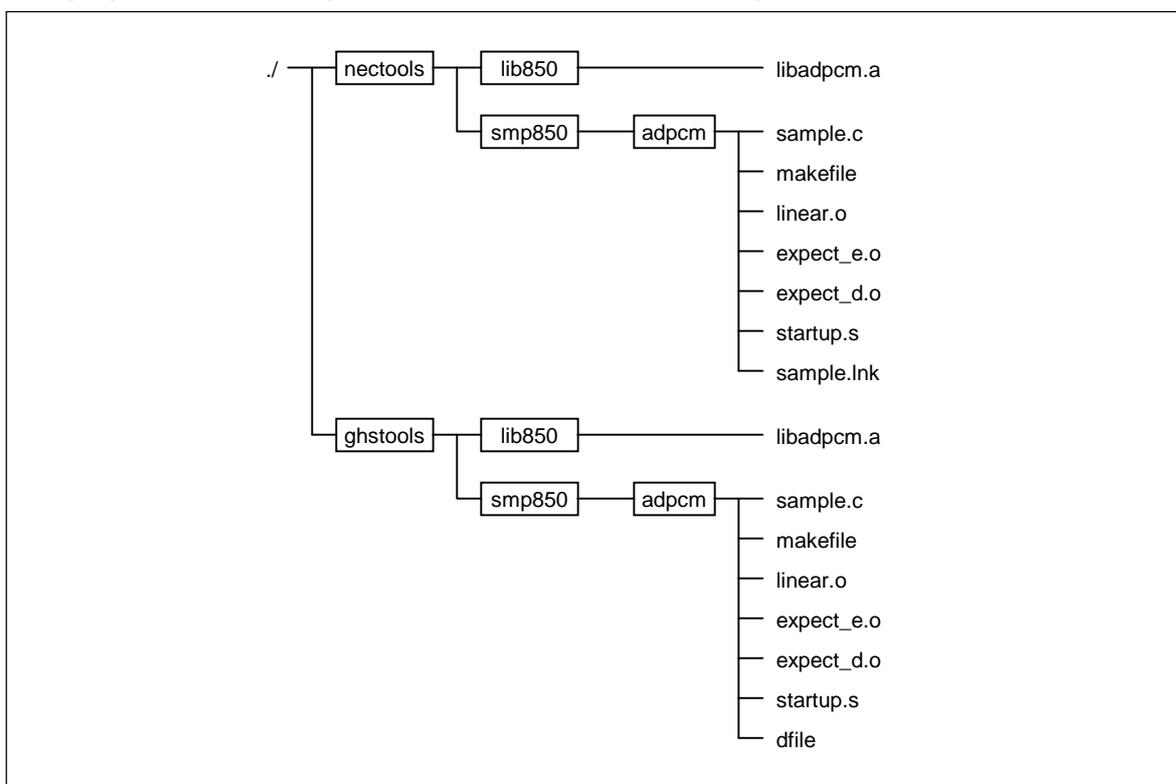
### 1.3.4 ディレクトリ構成

ミドルウェア・ライブラリ ADPCM のディレクトリ構成を次に示します。

#### ( 1 ) V810 ファミリ版 (ソフトウェア・バージョン : Ver.1.00)



#### ( 2 ) V850 ファミリ版 (ソフトウェア・バージョン : Ver.1.01)



(メ モ)

## 第2章 ライブラリ仕様

### 2.1 機能

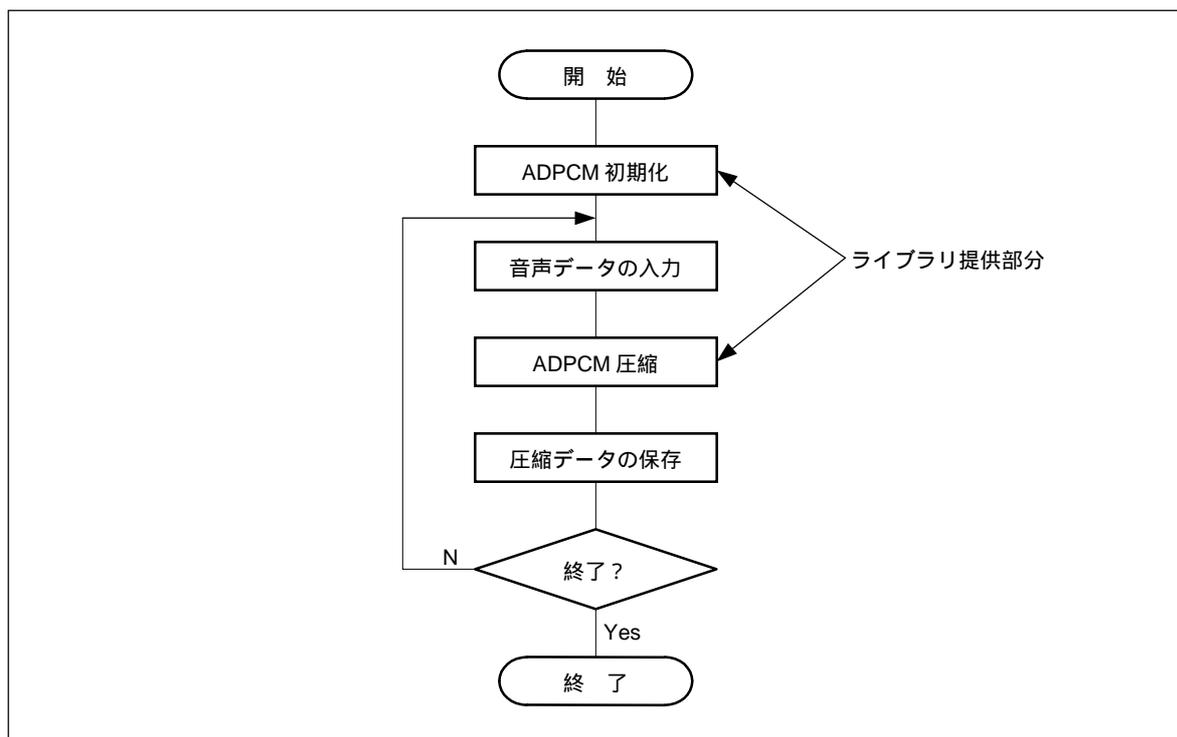
#### 2.1.1 圧縮処理

音声データを 32kbps, 16kbps の ADPCM コードに圧縮します。入力する音声データは, リニア・コード, 64kbps PCM コード ( $\mu$ -law, A-law) に対応して, それぞれの関数に分類されています。

表 2 - 1 圧縮系関数一覧

関数分類	入力データ・タイプ	圧縮データ・タイプ
adpcm_l32_enc()	リニア・コード	32kbps ADPCM
adpcm_pm32_enc()	64kbps PCM コード( $\mu$ -law)	32kbps ADPCM
adpcm_pa32_enc()	64kbps PCM コード(A-law)	32kbps ADPCM
adpcm_l16_enc()	リニア・コード	16kbps ADPCM
adpcm_pm16_enc()	64kbps PCM コード( $\mu$ -law)	16kbps ADPCM
adpcm_pa16_enc()	64kbps PCM コード(A-law)	16kbps ADPCM

図 2 - 1 圧縮処理フロー



### 2.1.2 伸長処理

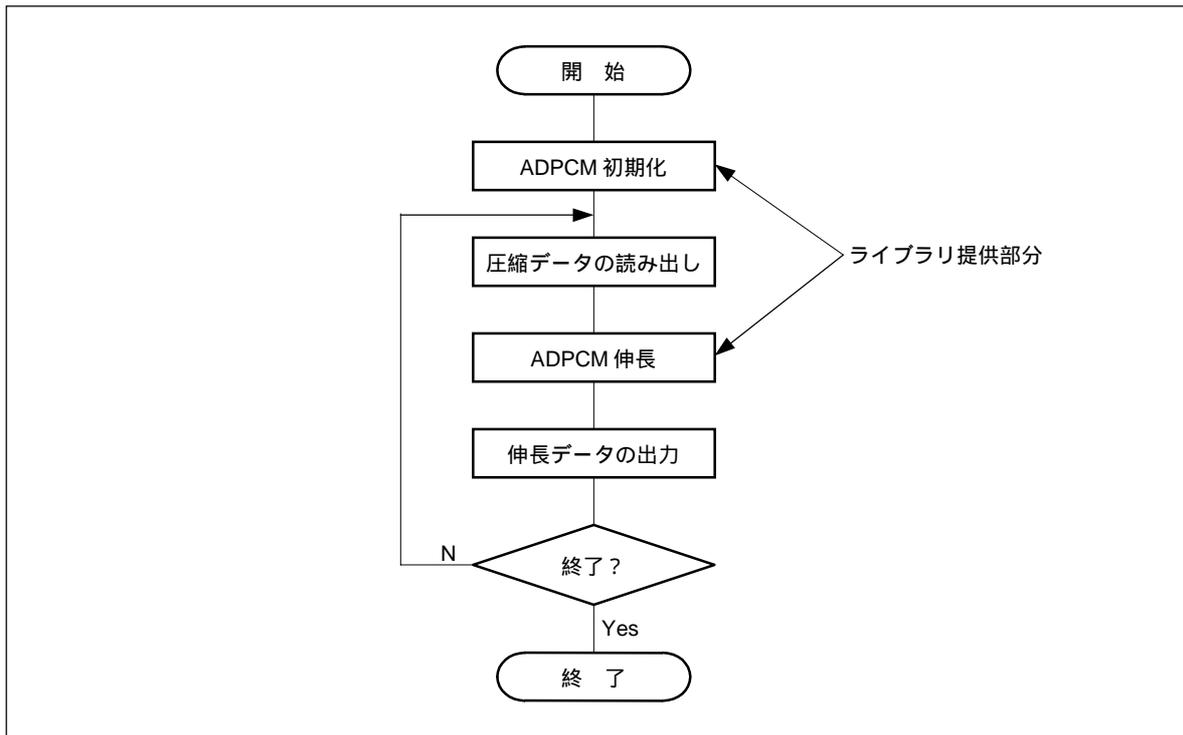
圧縮データ (32kbps, 16kbps の ADPCM コード) を音声データに伸長します。出力する音声データは、リニア・コード、64kbps PCM コード( $\mu$ -law, A-law)に対応して、それぞれの関数に分類されています。

表 2 - 2 伸長系関数一覧

関数分類	圧縮データ・タイプ	出力データ・タイプ
adpcm_l32_dec()	32kbps ADPCM	リニア・コード
adpcm_pm32_dec()	32kbps ADPCM	64kbps PCM コード( $\mu$ -law)
adpcm_pa32_dec()	32kbps ADPCM	64kbps PCM コード(A-law)
adpcm_tpm32_dec() 注	32kbps ADPCM	64kbps PCM コード( $\mu$ -law)
adpcm_tpa32_dec() 注	32kbps ADPCM	64kbps PCM コード(A-law)
adpcm_l16_dec()	16kbps ADPCM	リニア・コード
adpcm_pm16_dec()	16kbps ADPCM	64kbps PCM コード( $\mu$ -law)
adpcm_pa16_dec()	16kbps ADPCM	64kbps PCM コード(A-law)
adpcm_tpm16_dec() 注	16kbps ADPCM	64kbps PCM コード( $\mu$ -law)
adpcm_tpa16_dec() 注	16kbps ADPCM	64kbps PCM コード(A-law)

注 タンデム同期符号化に対応している関数です。

図 2 - 2 伸長処理フロー



## 2.2 RAM

このライブラリでは、圧縮／伸長のそれぞれの処理に対して、決められたメモリ容量（常駐領域）を必ず確保してください。常駐領域の確保は、ユーザ・アプリケーション上で次のように行ってください。

```
int work[16]
```

ユーザ・アプリケーション側で常駐領域を管理して、圧縮／伸長系関数の入力パラメータでポインタ（先頭アドレス）を指定してください。一連の圧縮／伸長処理が終了するまで、常駐領域の内容を壊さないでください。常駐領域が壊された場合の動作は保証できません。また、このライブラリでは、圧縮／伸長系関数は次に示すスタック領域を使用します。

- ・V810 ファミリ版：3ワード（12バイト）
- ・V850 ファミリ版：2ワード（8バイト）

## 2.3 データ型

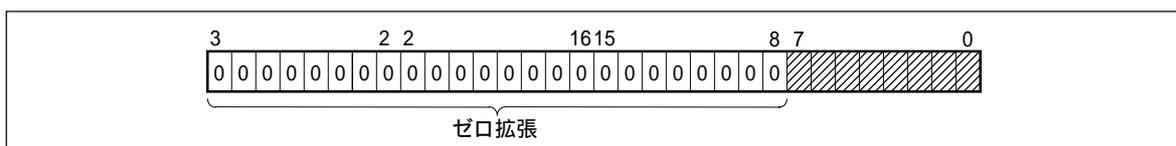
このライブラリで使用するデータについて説明します。各データ型は処理を高速化するために、int 幅（32ビット）に統一しています。また、各関数内では有効範囲についてチェックしないので注意してください。

### 2.3.1 int pcm\_mlaw

【内 容】64kbps PCM コード（ $\mu$ -law）

【表記方法】8ビット・データ（ビット31までゼロ拡張してください。）

【有効範囲】0～255

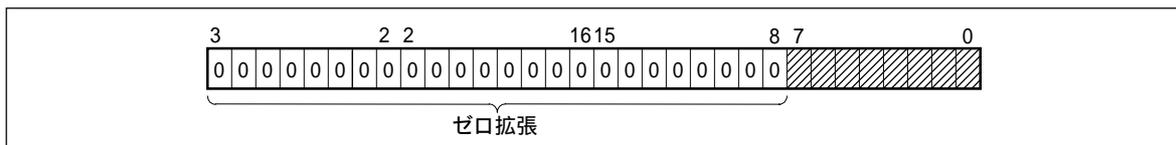


### 2.3.2 int pcm\_alaw

【内 容】64kbps PCM コード（A-law）

【表記方法】8ビット・データ（ビット31までゼロ拡張してください。）

【有効範囲】0～255

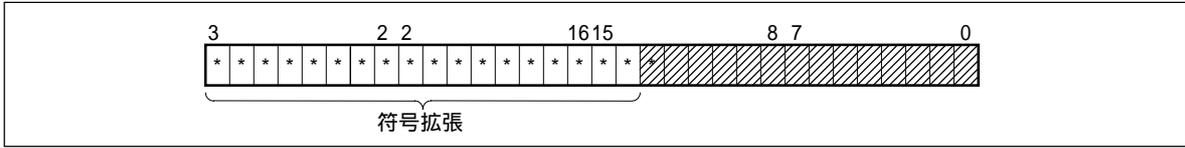


### 2.3.3 int linear\_enc

【内 容】リニア・コード（圧縮処理引き数）

【表記方法】14ビット・データ（2の補数表現，ビット31まで符号拡張してください。）

【有効範囲】 - 8192 ~ 8191

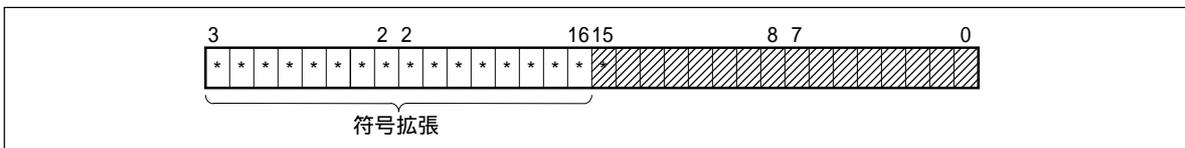


### 2.3.4 int linear\_dec

【内 容】リニア・コード（伸長処理戻り値）

【表記方法】16ビット・データ（2の補数表現，ビット31まで符号拡張されています。）

【有効範囲】 - 32768 ~ 32767

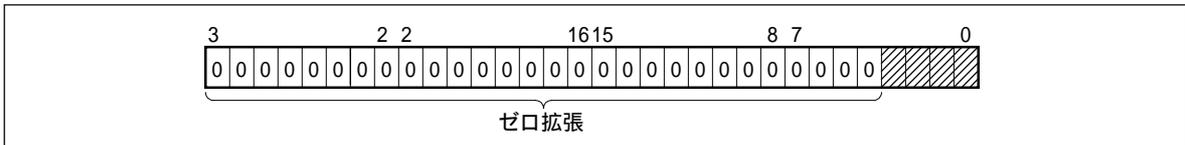


### 2.3.5 int adpcm\_32kbps

【内 容】32kbps ADPCM コード

【表記方法】4ビット・データ（ビット31までゼロ拡張してください。）

【有効範囲】 1 ~ 15

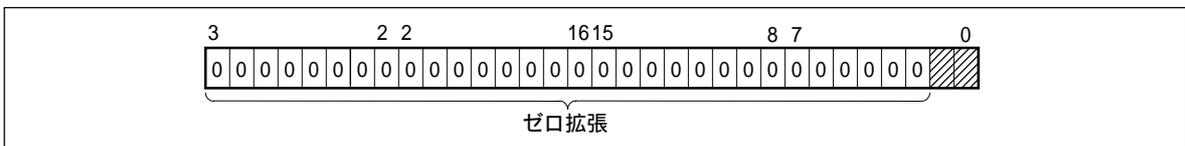


### 2.3.6 int adpcm\_16kbps

【内 容】16kbps ADPCM コード

【表記方法】2ビット・データ（ビット31までゼロ拡張してください。）

【有効範囲】 0 ~ 3



## 2.4 エラー処理

ADPCM ライブラリの各関数には、戻り値によるエラー・コードの出力などのエラー処理機能がありません。これは、勧告で計算途中の有効範囲外の値を制限（飽和）しているため、エラーが存在しないからです。エラーについては、NMI などがあります。これらのエラーが発生した場合は、関数コールの前後でフラグを管理するなどして、ユーザ・アプリケーション側で管理してください。

## 2.5 関数仕様

ADPCM ライブラリでは、初期化関数、圧縮（エンコーダ）系関数を6種類、伸長（デコーダ）系関数を10種類の合計17種類の関数を提供しています。次にこれらの関数について説明します。

### 2.5.1 初期化関数

#### (1) adpcm\_init 関数

エンコーダ、デコーダを初期化する関数です。

アプリケーション・プログラムで、エンコーダとデコーダの処理用常駐領域（64 バイト）を確保して、そのアドレスを引き数とします。

adpcm\_init 関数は、一連のエンコード/デコード処理をする前に必ず呼び出してください。

【分類】 ADPCM 初期化処理（エンコーダ、デコーダ共通）

【関数名】 adpcm\_init

【機能概要】 ADPCM エンコーダ、デコーダの常駐 RAM 領域を初期化します。

【形式】 void adpcm\_init(int \*work);

【引き数】

型	引き数	説明
int	work [16]	常駐領域（64 バイト）の先頭アドレス

【戻り値】 なし

【機能】 エンコード、デコード処理用常駐領域を初期化します。

**注意** adpcm\_init 関数を呼び出したあとは、一連のエンコード、デコードが終了するまで常駐 RAM 領域を壊さないでください。壊れた場合の動作は保証できません。

## 2.5.2 圧縮系関数

(1) `adpcm_l32_enc` 関数

リニア・コード・インタフェースの 32kbps ADPCM エンコード処理を行う関数です。

入力パラメータで与えられたリニア・コードをエンコードして、戻り値として 32kbps ADPCM コードを返します。

`adpcm_l32_enc` 関数に渡すパラメータのうち、エンコード処理用常駐領域は、`adpcm_init` 関数に渡したアドレスと同じアドレスを指定します。

【分類】 32kbps ADPCM エンコード処理 (リニア・コード・インタフェース)

【関数名】 `adpcm_l32_enc`

【機能概要】 指定したリニア・コードを 32kbps ADPCM コードにエンコードします。

【形式】 `int adpcm_l32_enc(int linear_enc, int *work)`

【引き数】

型	引き数	説明
int	<code>work [16]</code>	常駐領域 (64 バイト) の先頭アドレス
int	<code>linear_enc</code>	リニア・コード

【戻り値】

型	戻り値	説明
int	<code>adpcm_32</code>	32kbps ADPCM コード

【機能】 リニア・コードで与えられた 1 つのサンプリング・データを、32kbps ADPCM コードにエンコードします。

【備考】 ITU-T 勧告 G.726 の 32kbps ADPCM エンコーダから、次の機能を削除しています。  
・ 64kbps PCM コードからリニア・コードへの変換

( 2 ) `adpcm_pm32_enc` 関数

$\mu$ -law PCM コード・インタフェースの 32kbps ADPCM エンコード処理を行う関数です。

入力パラメータで与えられた 64kbps  $\mu$ -law PCM コードをエンコードして、戻り値として 32kbps ADPCM コードを返します。

`adpcm_pm32_enc` 関数に渡すパラメータのうち、エンコード処理用常駐領域は、`adpcm_init` 関数に渡したアドレスと同じアドレスを指定します。

【分 類】 32kbps ADPCM エンコード処理 (  $\mu$ -law PCM コード・インタフェース )

【関 数 名】 `adpcm_pm32_enc`

【機能概要】 指定した 64kbps  $\mu$ -law PCM コードを 32kbps ADPCM コードにエンコードします。

【形 式】 `int adpcm_pm32_enc(int pcm_mlaw,int *work)`

【引 き 数】

型	引き数	説 明
int	<code>work [16]</code>	常駐領域 ( 64 バイト ) の先頭アドレス
int	<code>pcm_mlaw</code>	64kbps $\mu$ -law PCM コード

【返 り 値】

型	返り値	説 明
int	<code>adpcm_32</code>	32kbps ADPCM コード

【機 能】 64kbps  $\mu$ -law PCM コードで与えられた 1 つのサンプリング・データを、32kbps ADPCM コードにエンコードします。

( 3 ) **adpcm\_pa32\_enc** 関数

A-law PCM コード・インタフェースの 32kbps ADPCM エンコード処理を行う関数です。

入力パラメータで与えられた 64kbps A-law PCM コードをエンコードして、戻り値として 32kbps ADPCM コードを返します。

adpcm\_pa32\_enc 関数に渡すパラメータのうち、エンコード処理用常駐領域は、adpcm\_init 関数に渡したアドレスと同じアドレスを指定します。

【分 類】 32kbps ADPCM エンコード処理 ( A-law PCM コード・インタフェース )

【関 数 名】 adpcm\_pa32\_enc

【機能概要】 指定した 64kbps A-law PCM コードを 32kbps ADPCM コードにエンコードします。

【形 式】 int adpcm\_pa32\_enc(int pcm\_alaw,int \*work)

【引 き 数】

型	引き数	説 明
int	work [16]	常駐領域 ( 64 バイト ) の先頭アドレス
int	pcm_alaw	64kbps A-law PCM コード

【返 り 値】

型	返り値	説 明
int	adpcm_32	32kbps ADPCM コード

【機 能】 64kbps A-law PCM コードで与えられた 1 つのサンプリング・データを、32kbps ADPCM コードにエンコードします。

(4) `adpcm_l16_enc` 関数

リニア・コード・インタフェースの 16kbps ADPCM エンコード処理を行う関数です。

入力パラメータで与えられたリニア・コードをエンコードして、戻り値として 16kbps ADPCM コードを返します。

`adpcm_l16_enc` 関数に渡すパラメータのうち、エンコード処理用常駐領域は、`adpcm_init` 関数に渡したアドレスと同じアドレスを指定します。

【分類】 16kbps ADPCM エンコード処理 (リニア・コード・インタフェース)

【関数名】 `adpcm_l16_enc`

【機能概要】 指定したリニア・コードを 32kbps ADPCM コードにエンコードします。

【形式】 `int adpcm_l16_enc(int linear_enc,int *work)`

【引き数】

型	引き数	説明
int	<code>work [16]</code>	常駐領域 (64 バイト) の先頭アドレス
int	<code>linear_enc</code>	リニア・コード

【戻り値】

型	戻り値	説明
int	<code>adpcm_16</code>	16kbps ADPCM コード

【機能】 リニア・コードで与えられた 1 つのサンプリング・データを、16kbps ADPCM コードにエンコードします。

【備考】 ITU-T 勧告 G.726 の 16kbps ADPCM エンコーダから、次の機能を削除しています。  
 ・ 64kbps PCM コードからリニア・コードへの変換

( 5 ) `adpcm_pm16_enc` 関数

$\mu$ -law PCM コード・インタフェースの 16kbps ADPCM エンコード処理を行う関数です。

入力パラメータで与えられた 64kbps  $\mu$ -law PCM コードをエンコードして、戻り値として 16kbps ADPCM コードを返します。

`adpcm_pm16_enc` 関数に渡すパラメータのうち、エンコード処理用常駐領域は、`adpcm_init` 関数に渡したアドレスと同じアドレスを指定します。

【分 類】 16kbps ADPCM エンコード処理 (  $\mu$ -law PCM コード・インタフェース )

【関 数 名】 `adpcm_pm16_enc`

【機能概要】 指定した 64kbps  $\mu$ -law PCM コードを 16kbps ADPCM コードにエンコードします。

【形 式】 `int adpcm_pm16_enc(int pcm_mlaw,int *work)`

【引 き 数】

型	引き数	説 明
int	<code>work [16]</code>	常駐領域 ( 64 バイト ) の先頭アドレス
int	<code>pcm_mlaw</code>	64kbps $\mu$ -law PCM コード

【返 り 値】

型	返り値	説 明
int	<code>adpcm_16</code>	16kbps ADPCM コード

【機 能】 64kbps  $\mu$ -law PCM コードで与えられた 1 つのサンプリング・データを、16kbps ADPCM コードにエンコードします。

( 6 ) **adpcm\_pa16\_enc** 関数

A-law PCM コード・インタフェースの 16kbps ADPCM エンコード処理を行う関数です。

入力パラメータで与えられた 64kbps A-law PCM コードをエンコードして、戻り値として 16kbps ADPCM コードを返します。

adpcm\_pa16\_enc 関数に渡すパラメータのうち、エンコード処理用常駐領域は、adpcm\_init 関数に渡したアドレスと同じアドレスを指定します。

【分 類】 16kbps ADPCM エンコード処理 ( A-law PCM コード・インタフェース )

【関 数 名】 adpcm\_pa16\_enc

【機能概要】 指定した 64kbps A-law PCM コードを 16kbps ADPCM コードにエンコードします。

【形 式】 int adpcm\_pa16\_enc(int pcm\_alaw,int \*work)

【引 き 数】

型	引き数	説 明
int	work [16]	常駐領域 ( 64 バイト ) の先頭アドレス
int	pcm_alaw	64kbps A-law PCM コード

【返 り 値】

型	返り値	説 明
int	adpcm_16	16kbps ADPCM コード

【機 能】 64kbps A-law PCM コードで与えられた 1 つのサンプリング・データを , 16kbps ADPCM コードにエンコードします。

## 2.5.3 伸長系関数

## (1) adpcm\_l32\_dec 関数

リニア・コード・インタフェースの 32kbps ADPCM デコード処理を行う関数です。

入力パラメータで与えられた 32kbps ADPCM コードをデコードして、返り値としてリニア・コードを返します。

adpcm\_l32\_dec 関数に渡すパラメータのうち、デコード処理用常駐領域は、adpcm\_init 関数に渡したアドレスと同じアドレスを指定します。

【分類】 32kbps ADPCM デコード処理 (リニア・コード・インタフェース)

【関数名】 adpcm\_l32\_dec

【機能概要】 指定した 32kbps ADPCM コードをリニア・コードにデコードします。

【形式】 int adpcm\_l32\_dec(int adpcm\_32,int \*work)

【引き数】

型	引き数	説明
int	work [16]	常駐領域 (64 バイト) の先頭アドレス
int	adpcm_32	32kbps ADPCM コード

【返り値】

型	返り値	説明
int	linear_dec	リニア・コード

【機能】 32kbps ADPCM コードで与えられた 1 つのサンプリング・データを、リニア・コードにデコードします。

【備考】 ITU-T 勧告 G.726 の 32kbps ADPCM デコーダから、次の機能を削除しています。

- ・リニア・コードから 64kbps PCM コードへの変換
- ・同期符号化補正

( 2 ) `adpcm_pm32_dec` 関数

$\mu$ -law PCM コード・インタフェースの 32kbps ADPCM デコード処理を行う関数です。

入力パラメータで与えられた 32kbps ADPCM コードをデコードして、返り値として 64kbps  $\mu$ -law PCM コードを返します。

`adpcm_pm32_dec` 関数に渡すパラメータのうち、デコード処理用常駐領域は、`adpcm_init` 関数に渡したアドレスと同じアドレスを指定します。

【分 類】 32kbps ADPCM デコード処理 (  $\mu$ -law PCM コード・インタフェース )

【関 数 名】 `adpcm_pm32_dec`

【機能概要】 指定した 32kbps ADPCM コードを 64kbps  $\mu$ -law PCM コードにデコードします。

【形 式】 `int adpcm_pm32_dec(int adpcm_32,int *work)`

【引 き 数】

型	引き数	説 明
int	<code>work [16]</code>	常駐領域 ( 64 バイト ) の先頭アドレス
int	<code>adpcm_32</code>	32kbps ADPCM コード

【返 り 値】

型	返り値	説 明
int	<code>pcm_mlaw</code>	64kbps $\mu$ -law PCM コード

【機 能】 32kbps ADPCM コード で与えられた 1 つのサンプリング・データを、64kbps  $\mu$ -law PCM コードにデコードします。

【備 考】 ITU-T 勧告 G.726 の 32kbps ADPCM デコーダから、次の機能を削除しています。  
・同期符号化補正

( 3 ) **adpcm\_pa32\_dec** 関数

A-law PCM コード・インタフェースの 32kbps ADPCM デコード処理を行う関数です。

入力パラメータで与えられた 32kbps ADPCM コードをデコードして、戻り値として 64kbps A-law PCM コードを返します。

adpcm\_pa32\_dec 関数に渡すパラメータのうち、デコード処理用常駐領域は、adpcm\_init 関数に渡したアドレスと同じアドレスを指定します。

【分 類】 32kbps ADPCM デコード処理 ( A-law PCM コード・インタフェース )

【関 数 名】 adpcm\_pa32\_dec

【機能概要】 指定した 32kbps ADPCM コードを 64kbps A-law PCM コードにデコードします。

【形 式】 int adpcm\_pa32\_dec(int adpcm\_32,int \*work)

【引 き 数】

型	引き数	説 明
int	work [16]	常駐領域 ( 64 バイト ) の先頭アドレス
int	adpcm_32	32kbps ADPCM コード

【返 り 値】

型	返り値	説 明
int	pcm_alaw	64kbps A-law PCM コード

【機 能】 32kbps ADPCM コード で与えられた 1 つのサンプリング・データを、64kbps A-law PCM コードにデコードします。

【備 考】 ITU-T 勧告 G.726 の 32kbps ADPCM デコーダから、次の機能を削除しています。  
・同期符号化補正

( 4 ) **adpcm\_tpm32\_dec** 関数

μ-law PCM コード・インタフェースの 32kbps ADPCM デコード処理を行う関数です。

入力パラメータで与えられた 32kbps ADPCM コードをデコードして、返り値として 64kbps μ-law PCM コードを返します。

adpcm\_tpm32\_dec 関数に渡すパラメータのうち、デコード処理用常駐領域は、adpcm\_init 関数に渡したアドレスと同じアドレスを指定します。

【分 類】 32kbps ADPCM デコード処理 ( μ-law PCM コード・インタフェース )

【関 数 名】 adpcm\_tpm32\_dec

【機能概要】 指定した 32kbps ADPCM コードを 64kbps μ-law PCM コードにデコードします。

【形 式】 int adpcm\_tpm32\_dec(int adpcm\_32,int \*work)

【引 き 数】

型	引き数	説 明
int	work [16]	常駐領域 ( 64 バイト ) の先頭アドレス
int	adpcm_32	32kbps ADPCM コード

【返 り 値】

型	返り値	説 明
int	pcm_mlaw	64kbps μ-law PCM コード

【機 能】 32kbps ADPCM コードで与えられた 1 つのサンプリング・データを、64kbps μ-law PCM コードにデコードします。

( 5 ) `adpcm_tpa32_dec` 関数

A-law PCM コード・インタフェースの 32kbps ADPCM デコード処理を行う関数です。

入力パラメータで与えられた 32kbps ADPCM コードをデコードして、戻り値として 64kbps A-law PCM コードを返します。

`adpcm_tpa32_dec` 関数に渡すパラメータのうち、デコード処理用常駐領域は、`adpcm_init` 関数に渡したアドレスと同じアドレスを指定します。

【分 類】 32kbps ADPCM デコード処理 ( A-law PCM コード・インタフェース )

【関 数 名】 `adpcm_tpa32_dec`

【機能概要】 指定した 32kbps ADPCM コードを 64kbps A-law PCM コードにデコードします。

【形 式】 `int adpcm_tpa32_dec(int adpcm_32,int *work)`

【引 き 数】

型	引き数	説 明
int	<code>work [16]</code>	常駐領域 ( 64 バイト ) の先頭アドレス
int	<code>adpcm_32</code>	32kbps ADPCM コード

【返 り 値】

型	返り値	説 明
int	<code>pcm_alaw</code>	64kbps A-law PCM コード

【機 能】 32kbps ADPCM コードで与えられた 1 つのサンプリング・データを、64kbps A-law PCM コードにデコードします。

( 6 ) **adpcm\_l16\_dec** 関数

リニア・コード・インタフェースの 16kbps ADPCM デコード処理を行う関数です。

入力パラメータで与えられた 16kbps ADPCM コードをデコードして、戻り値としてリニア・コードを返します。

adpcm\_l16\_dec 関数に渡すパラメータのうち、デコード処理用常駐領域は、adpcm\_init 関数に渡したアドレスと同じアドレスを指定します。

【分 類】 16kbps ADPCM デコード処理 (リニア・コード・インタフェース)

【関 数 名】 adpcm\_l16\_dec

【機能概要】 指定した 16kbps ADPCM コードをリニア・コードにデコードします。

【形 式】 int adpcm\_l16\_dec(int adpcm\_16,int \*work)

【引 き 数】

型	引き数	説 明
int	work [16]	常駐領域 (64 バイト) の先頭アドレス
int	adpcm_16	16kbps ADPCM コード

【返 り 値】

型	返り値	説 明
int	linear_dec	リニア・コード

【機 能】 16kbps ADPCM コードで与えられた 1 つのサンプリング・データを、リニア・コードにデコードします。

【備 考】 ITU-T 勧告 G.726 の 32kbps ADPCM デコーダから、次の機能を削除しています。

- ・リニア・コードから 64kbps PCM コードへの変換
- ・同期符号化補正

(7) `adpcm_pm16_dec` 関数

$\mu$ -law PCM コード・インタフェースの 16kbps ADPCM デコード処理を行う関数です。

入力パラメータで与えられた 16kbps ADPCM コードをデコードして、返り値として 64kbps  $\mu$ -law PCM コードを返します。

`adpcm_pm16_dec` 関数に渡すパラメータのうち、デコード処理用常駐領域は、`adpcm_init` 関数に渡したアドレスと同じアドレスを指定します。

【分類】 16kbps ADPCM デコード処理 ( $\mu$ -law PCM コード・インタフェース)

【関数名】 `adpcm_pm16_dec`

【機能概要】 指定した 16kbps ADPCM コードを 64kbps  $\mu$ -law PCM コードにデコードします。

【形式】 `int adpcm_pm16_dec(int adpcm_16,int *work)`

【引き数】

型	引き数	説明
int	<code>work [16]</code>	常駐領域 (64 バイト) の先頭アドレス
int	<code>adpcm_16</code>	16kbps ADPCM コード

【返り値】

型	返り値	説明
int	<code>pcm_mlaw</code>	64kbps $\mu$ -law PCM コード

【機能】 16kbps ADPCM コードで与えられた 1 つのサンプリング・データを、64kbps  $\mu$ -law PCM コードにデコードします。

【備考】 ITU-T 勧告 G.726 の 16kbps ADPCM デコーダから、次の機能を削除しています。

- ・同期符号化補正

( 8 ) `adpcm_pa16_dec` 関数

A-law PCM コード・インタフェースの 16kbps ADPCM デコード処理を行う関数です。

入力パラメータで与えられた 16kbps ADPCM コードをデコードして、戻り値として 64kbps A-law PCM コードを返します。

`adpcm_pa16_dec` 関数に渡すパラメータのうち、デコード処理用常駐領域は、`adpcm_init` 関数に渡したアドレスと同じアドレスを指定します。

【分 類】 16kbps ADPCM デコード処理 ( A-law PCM コード・インタフェース )

【関 数 名】 `adpcm_pa16_dec`

【機能概要】 指定した 16kbps ADPCM コードを 64kbps A-law PCM コードにデコードします。

【形 式】 `int adpcm_pa16_dec(int adpcm_16,int *work)`

【引 き 数】

型	引き数	説 明
int	<code>work [16]</code>	常駐領域 ( 64 バイト ) の先頭アドレス
int	<code>adpcm_16</code>	16kbps ADPCM コード

【返 り 値】

型	返り値	説 明
int	<code>pcm_alaw</code>	64kbps A-law PCM コード

【機 能】 16kbps ADPCM コードで与えられた 1 つのサンプリング・データを、64kbps A-law PCM コードにデコードします。

【備 考】 ITU-T 勧告 G.726 の 16kbps ADPCM デコーダから、次の機能を削除しています。  
・同期符号化補正

( 9 ) **adpcm\_tpm16\_dec** 関数

μ-law PCM コード・インタフェースの 16kbps ADPCM デコード処理を行う関数です。

入力パラメータで与えられた 16kbps ADPCM コードをデコードして、返り値として 64kbps μ-law PCM コードを返します。

adpcm\_tpm16\_dec 関数に渡すパラメータのうち、デコード処理用常駐領域は、adpcm\_init 関数に渡したアドレスと同じアドレスを指定します。

【分 類】 16kbps ADPCM デコード処理 ( μ-law PCM コード・インタフェース )

【関 数 名】 adpcm\_tpm16\_dec

【機能概要】 指定した 16kbps ADPCM コードを 64kbps μ-law PCM コードにデコードします。

【形 式】 int adpcm\_tpm16\_dec(int adpcm\_16,int \*work)

【引 き 数】

型	引き数	説 明
int	work [16]	常駐領域 ( 64 バイト ) の先頭アドレス
int	adpcm_16	16kbps ADPCM コード

【返 り 値】

型	返り値	説 明
int	pcm_mlaw	64kbps μ-law PCM コード

【機 能】 16kbps ADPCM コードで与えられた 1 つのサンプリング・データを、64kbps μ-law PCM コードにデコードします。

(10) `adpcm_tpa16_dec` 関数

A-law PCM コード・インタフェースの 16kbps ADPCM デコード処理を行う関数です。

入力パラメータで与えられた 16kbps ADPCM コードをデコードして、戻り値として 64kbps A-law PCM コードを返します。

`adpcm_tpa16_dec` 関数に渡すパラメータのうち、デコード処理用常駐領域は、`adpcm_init` 関数に渡したアドレスと同じアドレスを指定します。

【分類】 16kbps ADPCM デコード処理 (A-law PCM コード・インタフェース)

【関数名】 `adpcm_tpa16_dec`

【機能概要】 指定した 16kbps ADPCM コードを 64kbps A-law PCM コードにデコードします。

【形式】 `int adpcm_tpa16_dec(int adpcm_16,int *work)`

【引き数】

型	引き数	説明
int	<code>work [16]</code>	常駐領域 (64 バイト) の先頭アドレス
int	<code>adpcm_16</code>	16kbps ADPCM コード

【戻り値】

型	戻り値	説明
int	<code>pcm_alaw</code>	64kbps A-law PCM コード

【機能】 16kbps ADPCM コードで与えられた 1 つのサンプリング・データを、64kbps A-law PCM コードにデコードします。

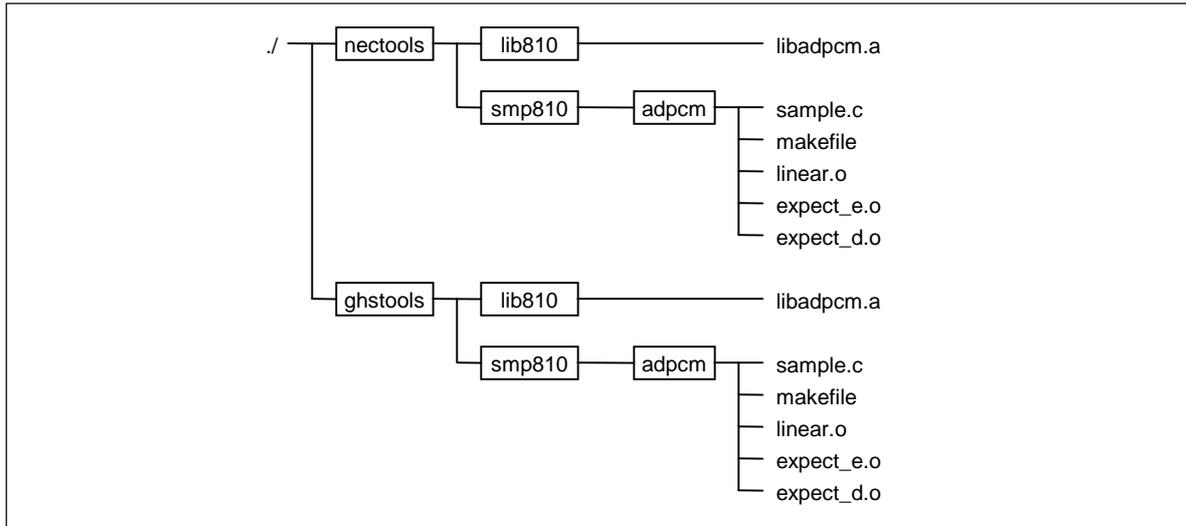
[メモ]

## 第3章 インストール

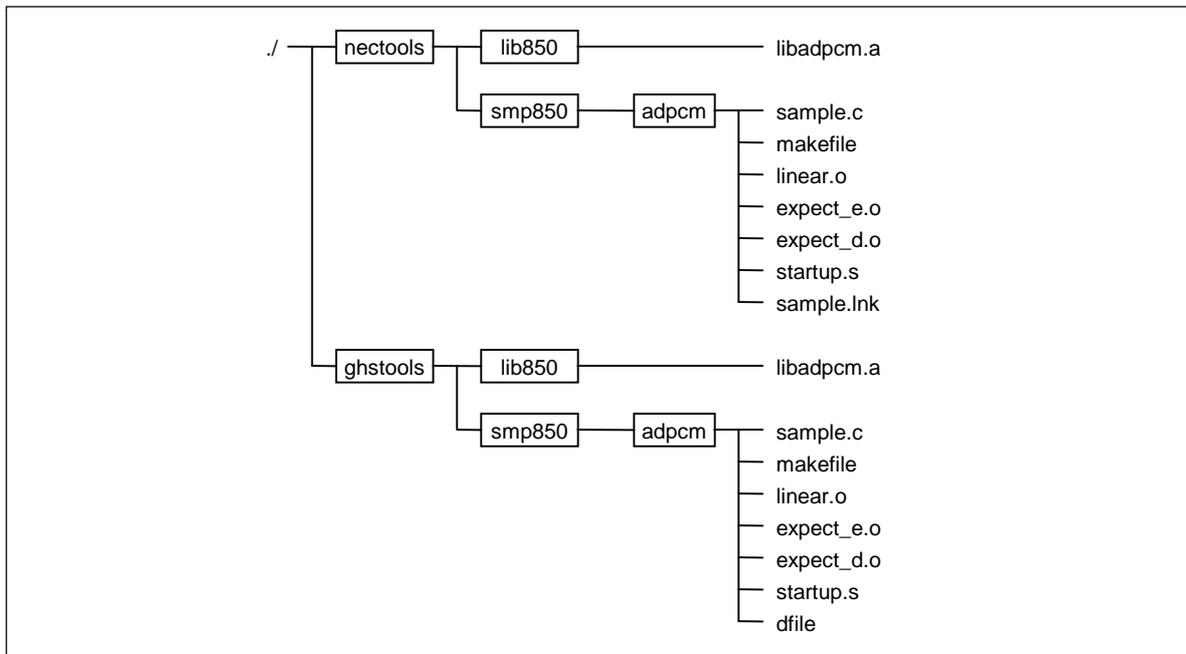
### 3.1 提供形態

ミドルウェア・ライブラリ ADPCM は、NEC 製または GHS 社製ツールを使用してアプリケーションを開発するためのライブラリで、NEC 製と GHS 社製の2種類を提供しています。次に提供媒体の内容を示します。

#### 3.1.1 V810 ファミリ版



#### 3.1.2 V850 ファミリ版



### 3.1.3 ディレクトリとファイル

各ディレクトリおよびファイルについて説明します。

#### ( 1 ) nectools

NEC 製のツールを使用して、アプリケーションを開発する場合の ADPCM プログラムを格納しています。

#### ( 2 ) ghstools

GHS 社製のツールを使用して、アプリケーションを開発する場合の ADPCM プログラムを格納しています。

#### ( 3 ) lib810 , lib850

ADPCM のライブラリを格納しています。

#### ( 4 ) smp810/adpcm , smp850/adpcm

ADPCM を使用したサンプル・プログラムおよびサンプル音声データを格納しています。

## 3.2 ホスト・マシンへのファイル展開

提供媒体からホスト・マシン上にファイル群を転送する手順を、UNIX 版(SUN4)と MS-DOS™/PC DOS™ 版に分けて説明します。

### 3.2.1 UNIX 版

UNIX 版の提供媒体は、CGMT と 3.5 インチ FD の 2 種類があります。この 2 種類の媒体には、tar 形式で ADPCM のファイル群を格納しています。ホスト・マシンへのインストールの手順を次に示します。

ADPCM をインストールするためのディレクトリを作成します。ここでは mw\_adpcm という名前のディレクトリを作成します。

```
% mkdir mw_adpcm <CR>
```

作成したディレクトリに移動します。

```
% cd mw_adpcm <CR>
```

提供媒体をセットします。

- ・CGMT の場合、磁気テープ装置にセットします。
- ・3.5 インチ FD の場合、フロッピー・ディスク装置にセットします。

tar コマンドを実行して、ファイル群をディスク上に展開します。なお、ホスト・マシンにより指定するスペシャル・ファイル名は異なります。

/dev/rst8 であるとして、NEC 製ツール用のファイル群を展開する場合を次に示します。

```
% tar -xvof /dev/rst8 nectools <CR>
```

同様に GHS 社製ツール用のファイル群を展開する場合を次に示します。

```
% tar -xvof /dev/rst8 ghstools <CR>
```

ファイルがインストールされたことを確認します。各ディレクトリについては **3.1 提供形態** を参照してください。

```
% ls -CFR <CR>
```

### 3.2.2 MS-DOS/PC DOS 版

MS-DOS/PC DOS 版の提供媒体は、3.5 インチ FD で提供してします。ホスト・マシンへのインストールの手順を次に示します。

ADPCM をインストールするためのディレクトリを作成します。ここでは A ドライブに mw\_adpcm という名前のディレクトリを作成します。

```
A> md mw_adpcm <CR>
```

作成したディレクトリに移動します。

```
A> cd mw_adpcm <CR>
```

提供媒体をフロッピー・ディスク・ドライブにセットします。ここでは C ドライブにセットとします。

xcopy コマンドを実行して、ファイル群を展開します。

NEC 製ツール用のファイル群を展開する場合を次に示します。

```
A> xcopy c:\nectools . /s /e /v <CR>
```

同様に GHS 社製ツール用のファイル群を展開する場合を次に示します。

```
A> xcopy c:\ghstools . /s /e /v <CR>
```

ファイルがインストールされたことを確認します。各ディレクトリについては **3.1 提供形態**を参照してください。

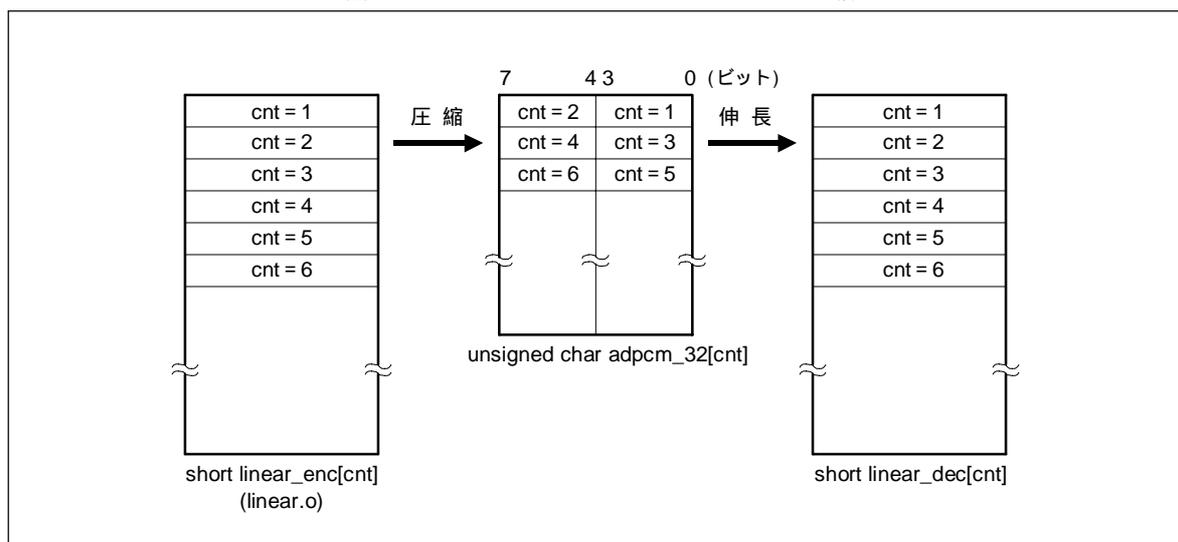
```
A> dir a:\nectools <CR>
```

### 3.3 サンプル・プログラムの作成

smp810 (V810 ファミリ版) または smp850 (V850 ファミリ版) のディレクトリには、32kbps ADPCM (リニア・コード・インタフェース) の圧縮関数と伸長関数を使用したサンプル・プログラムと音声データを格納しています (sample.c のソース・プログラムについては付録を参照してください)。

サンプル・プログラムの内容は、linear.o のファイルに格納された linear\_enc という配列のデータを圧縮して、adpcm\_32 という配列に結果を格納します。次に、この圧縮データを伸長して、linear\_dec という配列にデコード結果を格納します。

図3-1 サンプル・プログラム・データの構成



次に、V810 ファミリ版のサンプル・プログラムの make について例を示します (NEC 製ツールを使用して、UNIX 版を使用した場合)。

サンプル・プログラムを格納しているディレクトリに移動します。

```
% cd mw_adpcm/nectools/smp810/adpcm <CR>
```

エディタを使用して、startup.c をターゲットにあわせて変更します。サンプルで提供している startup.c は、スタック・ポインタの初期化だけを記述しています。

make コマンドを実行して、sample.elf を生成します。

```
% make sample.elf <CR>
```

sample.elf が生成されたことを確認します。

```
% ls -l sample.elf <CR>
```

インサーキット・エミュレータなどを使用して、ターゲットにダウンロードして実行します。

### 3.4 ロケーションの変更

ADPCM ライブラリは、次に示すセクション名が付けられています。ユーザのターゲットにあわせて、ロケーションを変更できます。

セクション名	タイプ	内 容
.ad_text	.text	テキスト(命令コード)
.ad_data	.data	テーブル・データ(定数)

### 3.5 シンボル名規約

ADPCM ライブラリ内で使用するシンボル名などは、次に示す規約に従っています。ほかのアプリケーションと組み合わせて使用するときは、重複しないように注意してください。

分 類	規 約
関数/シンボル	先頭に "adpcm_" を付けています。

## 付 録 sample.c のソース・プログラム

```
--- sample.c ---
extern short linear_enc[0x4000];
unsigned char adpcm_32[0x4000/2];
short linear_dec[0x4000];

int work[16];

void
encoder()
{
    unsigned register int cnt;
    adpcm_init( work );
    for( cnt = 0; cnt<0x4000; cnt++ ){
        if(cnt & 1){
            adpcm_32[cnt>>1]|=((unsigned char)adpcm_132_enc((int)linear_enc[cnt],
                work)) << 4;
        }
        else{
            adpcm_32[cnt>>1]=((unsigned char)adpcm_132_enc((int)linear_enc[cnt],
                work));
        }
    }
}

void
decoder()
{
    unsigned register int cnt;
    adpcm_init( work );
    for( cnt = 0; cnt<0x4000; cnt++ ){
        if(cnt & 1){
            linear_dec[cnt] = (short)adpcm_132_dec( ((int)adpcm_32[cnt>>1]) >> 4,
                work );
        }
        else{
            linear_dec[cnt] = (short)adpcm_132_dec( ((int)adpcm_32[cnt>>1]) & 0xf ,
                work );
        }
    }
}

main ()
{
    encoder();
    decoder();
}
```

(メ モ)

[ × 毛 ]

## — お問い合わせ先 —

### 【技術的なお問い合わせ先】

NEC半導体テクニカルホットライン  
(電話：午前 9:00～12:00，午後 1:00～5:00)

電話 : 044-435-9494  
FAX : 044-435-9608  
E-mail : s-info@saed.tmg.nec.co.jp

### 【営業関係お問い合わせ先】

#### 第一販売事業部

東京 (03)3798-6106, 6107,  
6108

名古屋 (052)222-2375

大阪 (06)6945-3178, 3200,  
3208, 3212

仙台 (022)267-8740

郡山 (024)923-5591

千葉 (043)238-8116

#### 第二販売事業部

東京 (03)3798-6110, 6111,  
6112

立川 (042)526-5981, 6167

松本 (0263)35-1662

静岡 (054)254-4794

金沢 (076)232-7303

松山 (089)945-4149

#### 第三販売事業部

東京 (03)3798-6151, 6155, 6586,  
1622, 1623, 6156

水戸 (029)226-1702

広島 (082)242-5504

高崎 (027)326-1303

鳥取 (0857)27-5313

太田 (0276)46-4014

名古屋 (052)222-2170, 2190

福岡 (092)261-2806

### 【資料の請求先】

上記営業関係お問い合わせ先またはNEC特約店へお申しつけください。

### 【インターネット電子デバイス・ニュース】

NECエレクトロニクスデバイスの情報がインターネットでご覧になれます。

URL(アドレス)

<http://www.ic.nec.co.jp/>

## アンケート記入のお願い

お手数ですが、このドキュメントに対するご意見をお寄せください。今後のドキュメント作成の参考にさせていただきます。

[ドキュメント名] μSAP703000-B04, μSAP70732-B04 ユーザーズ・マニュアル  
(U11381JJ2V1UMJ1 (第2版))

[お名前など] (さしつかえない範囲で)

御社名(学校名, その他) ( )  
ご住所 ( )  
お電話番号 ( )  
お仕事の内容 ( )  
お名前 ( )

1. ご評価(各欄に をご記入ください)

項 目	大変良い	良い	普通	悪い	大変悪い
全体の構成					
説明内容					
用語解説					
調べやすさ					
デザイン, 字の大きさなど					
その他 ( )					
( )					

2. わかりやすい所(第 章, 第 章, 第 章, 第 章, その他 )  
理由 [ ]

3. わかりにくい所(第 章, 第 章, 第 章, 第 章, その他 )  
理由 [ ]

4. ご意見, ご要望

5. このドキュメントをお届けしたのは  
NEC 販売員, 特約店販売員, その他 ( )

ご協力ありがとうございました。

下記あてに FAX で送信いただくか, 最寄りの販売員にコピーをお渡しください。

日本電気(株) NEC エレクトロニクス  
半導体テクニカルホットライン

FAX : (044) 435-9608

2000.6