

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

μSAP77016-B04

G723.1 音声コーデック・ミドルウェア

対象デバイス

μPD77018

μPD77018A

μPD77019

μPD77110

μPD77111

μPD77112

μPD77113

μPD77114

μPD77116

[メ モ]

目次要約

第1章 概 説 ...	11
第2章 ライブラリ仕様 ...	15
第3章 インストレーション ...	35
付 録 サンプル・プログラム・ソース ...	37

[メ モ]

Windows 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。

- **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
全般	対象デバイスに μ PD77113, 77114 を追加
p.27	2.3.1 G.723.1 ANNEX A 用関数の引き渡しパラメータに説明を追加
p.28	2.3.2 G.723.1 ANNEX C 用関数の引き渡しパラメータに説明を追加
p.33	2.4.2 (2) SID フレームの圧縮フォーマットに説明を追加

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、 μ PD77016 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。

μ PD77016 ファミリは、 μ PD77015, 77016, 77017, 77018, 77018A, 77019, 77110, 77111, 77112, 77113, 77114, 77116^{*}の総称です。ただし、このマニュアルでは、 μ PD77018, 77018A, 77019, 77110, 77111, 77112, 77113, 77114, 77116 を対象デバイスにしています。

注 開発中

目的 μ PD77016 ファミリの応用システムを設計、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 このマニュアルでは、大きく分けて次の内容で構成しています。

第1章 概 説

第2章 ライブラリ仕様

第3章 インストレーション

付 録 サンプル・プログラム・ソース

読み方 このマニュアルの読者は、電気、論理回路、マイクロコンピュータおよびC言語に関する一般的知識が必要となります。

μ PD7701xファミリのハードウェア機能を知りたいとき

→ μ PD7701xファミリ ユーザーズ・マニュアル アーキテクチャ編を参照してください。

μ PD77111ファミリのハードウェア機能を知りたいとき

→ μ PD77111ファミリ ユーザーズ・マニュアル アーキテクチャ編を参照してください。

μ PD77016ファミリの命令機能を知りたいとき

→ μ PD77016ファミリ ユーザーズ・マニュアル 命令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, 信号の名称に上線)
注	: 本文中につけた注の説明
注意	: 気をつけて読んでいただきたい内容
備考	: 本文中の補足説明
数の表記	: 2進数... xxx または 0bxxx 10進数... xxx 16進数... 0xxx

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。

μ PD77016 ファミリに関する資料

資料名 品名	パンフレット	データ・シート	ユーザーズ・マニュアル		アプリケーション・ノート	
			アーキテクチャ編	命令編	基本ソフトウェア編	ライブラリ編
μ PD77016	U12395J	U10891J	U10503J	U13116J	U11958J	U12021J
μ PD77015		U10902J				
μ PD77017						
μ PD77018						
μ PD77018A		U11849J				
μ PD77019						
μ PD77019-013		U13053J				
μ PD77110		U12801J	U14623J			
μ PD77111						
μ PD77112						
μ PD77113		U14373J				
μ PD77114						
μ PD77116		-	U14624J	-	-	-

開発ツールに関する資料

資料名	資料番号	
SM77016 ユーザーズ・マニュアル	U11602J	
WB77016 ユーザーズ・マニュアル	言語編	U10078J
	操作編	U11506J
ID77016 ユーザーズ・マニュアル	U10118J	
IE-77016-98, IE-77016-PC ユーザーズ・マニュアル	ハードウェア編	U13044J
μ PD77016 スタータ・キット ユーザーズ・マニュアル	U13032J	
IE77016-CM-LC ユーザーズ・マニュアル	U14139J	
RX77016 ユーザーズ・マニュアル	機能編	U14397J
	コンフィギュレーション・ツール編	U14404J
RX77016 アプリケーション・ノート	HOST API 編	U14371J

注意 上記関連資料は、予告なしに内容を変更することがあります。設計などには、必ず最新の資料をご使用ください。

目 次

第1章 概 説 ... 11

- 1.1 ミドルウェア ... 11
- 1.2 G.723.1 音声コーデック ... 11
- 1.3 G.723.1 ANNEX A 音声コーデック ... 12
- 1.4 G.723.1 ANNEX C 音声コーデック ... 12
- 1.5 システム概要 ... 12
 - 1.5.1 特 徴 ... 12
 - 1.5.2 動作環境 ... 12
 - 1.5.3 性 能 ... 13
 - 1.5.4 ディレクトリ構成 ... 14

第2章 ライブラリ仕様 ... 15

- 2.1 G.723.1 音声コーデック処理フロー ... 15
- 2.2 関数仕様 ... 17
 - 2.2.1 エンコーダ初期化関数 ... 17
 - 2.2.2 デコーダ初期化関数 ... 20
 - 2.2.3 エンコーダ関数 ... 22
 - 2.2.4 デコーダ関数 ... 24
 - 2.2.5 バージョン取得関数 ... 26
- 2.3 外部インタフェース説明 ... 26
 - 2.3.1 G.723.1 ANNEX A 用関数の引き渡しパラメータ ... 27
 - 2.3.2 G.723.1 ANNEX C 用関数の引き渡しパラメータ ... 28
- 2.4 入出力データ・フォーマット ... 30
 - 2.4.1 エンコーダ入力/デコーダ出力データ・フォーマット ... 30
 - 2.4.2 エンコーダ出力/デコーダ入力データ・フォーマット ... 30

第3章 インストレーション ... 35

- 3.1 インストレーション手順 ... 35
- 3.2 サンプル作成手順 ... 35
- 3.3 シンボル命名規約 ... 36

付録 サンプル・プログラム・ソース ... 37

- 付.1 G.723.1 + ANNEX A 用 (sampleA.asm) ... 37
- 付.2 G.723.1 + ANNEX A + ANNEX C 用 (sampleAC.asm) ... 43

図の目次

図番号	タイトル, ページ
2 - 1	アプリケーション処理フロー (エンコーダ) ... 15
2 - 2	アプリケーション処理フロー (デコーダ) ... 16
2 - 3	圧縮データ・フレーム・フォーマット ... 30
3 - 1	サンプル・プログラム評価システム ... 35

表の目次

表番号	タイトル, ページ
1 - 1	必要メモリの容量 ... 13
2 - 1	G.723.1 ANNEX A 用関数の引き渡しパラメータ ... 27
2 - 2	G.723.1 ANNEX C 用関数の引き渡しパラメータ ... 28
2 - 3	G.723.1 ANNEX A 圧縮データのビット・レートとフレーム・バイト数 ... 30
2 - 4	G.723.1 ANNEX C 圧縮データのビット・レートとフレーム・バイト数 ... 30
2 - 5	6.3 Kbps の有音圧縮データ・フォーマット ... 31
2 - 6	5.3 Kbps の有音圧縮データ・フォーマット ... 32
2 - 7	無音圧縮データのビット割り当て ... 33
2 - 8	G.723.1 ANNEX C 圧縮データのビット割り当て ... 33
3 - 1	シンボル名 ... 36

第1章 概 説

1.1 ミドルウェア

ミドルウェアとは、プロセッサの性能をできるだけ引き出せるようにチューニングされたソフトウェア群で、従来ハードウェアが行っていた処理をソフトウェアで実現したものです。DSP という高性能プロセッサの出現、そして DSP が手軽にシステムに組み込める環境が整ってきたために、ミドルウェアという概念が現実のものとなりました。

NEC では、 μ PD77016 ファミリー用にマルチメディア・システムを実現する要素技術を提供しています。

たとえば音声コーデック、画像データの圧縮/伸長といったミドルウェアをタイムリに提供し、お客様のシステム開発を支援します。

μ SAP77016-B04 は、ITU-T 勧告 G.723.1 (付属勧告 ANNEX A , ANNEX C を含む) 音声圧縮伸長機能を提供するミドルウェアです。このマニュアルでは、特に指定のないかぎり、G.723.1 音声コーデックとして使用した場合を例にして説明しています。

注 International Telecommunication Union-Telecommunication Standardization Sector

1.2 G.723.1 音声コーデック

G.723.1 音声コーデックは、ITU-T で勧告された 5.3 Kbps , 6.3 Kbps の音声圧縮 / 伸長コーデックで、ACELP (代数的符号励振) と MP-MLQ (最尤量子化型マルチパルス) を使用した音声信号を符号化するためのアルゴリズムです。

G.723.1 音声コーデックは、電話帯域フィルタ (ITU-T 勧告 G.712) により帯域制限されたアナログ入力信号を 8 kHz で標本化し、次に 16 ビット・リニア PCM に変換することによって得られたデジタル信号を、符号器の入力として動作するように設計されています。同様に、復号器の出力もアナログ信号に戻される必要があります。

ITU-T 勧告 G.711 で規定される 64 Kbps PCM データなど、ほかの入出力形式の信号は、符号器の前で 16 ビット・リニア PCM へ、また復号器のあとで 16 ビット・リニア PCM から適当な形式へ変換する必要があります。符号器から復号器へ渡されるビット列は ITU-T 勧告 G.723.1 で規定されています。

備考 ACELP : Algebraic Code Excited Linear Prediction
MP-MLQ : Multi-Pulse Maximum Likelihood Quantization

1.3 G.723.1 ANNEX A 音声コーデック

ITU-T 勧告 G.723.1 の付属勧告 ANNEX A は、G.723.1 音声コーデックの無音圧縮機能であり、標準の G.723.1 音声コーデックに付加して使用します。また、無音圧縮機能を付加した音声コーデックは、付加していない音声コーデックとの相互接続はできません。この無音圧縮機能は、無音部分の圧縮率を上げることによって圧縮率を上げてビット・レートの軽減を目的としています。

1.4 G.723.1 ANNEX C 音声コーデック

ITU-T 勧告 G.723.1 の付属勧告 ANNEX C は、G.723.1 音声コーデックの可変ビット・レート・チャンネル・コーデックで、標準の G.723.1 または ANNEX A に付加して使用します。ANNEX C でサポートするビット・レートは、0.7 Kbps - 14.3 Kbps です。

ANNEX C は移動通信におけるマルチメディアに対する ITU-T H.324 ファミリの一部として設計されています。この規格によって、G.723.1 はあらゆる有線または無線伝送システムにも適応可能です。ANNEX C は、インタリーブやパースト・フォーマットのような伝送システムに依存する機能については規定していません。

1.5 システム概要

1.5.1 特 徴

- ・ 5.3 Kbps , 6.3 Kbps に圧縮符号化 (ただし、ANNEX C は可変ビット・レート)。
- ・ 高ビット・レート音声符号化。
- ・ サンプリング周波数 8 kHz で 240 サンプル/フレームの符号化、復号化を行う。
- ・ 音声入出力データはすべて 16 ビット・リニア・データ。

1.5.2 動作環境

(1) 動作対象 DSP

- ・ μ PD77018
- ・ μ PD77018A
- ・ μ PD77019
- ・ μ PD77110
- ・ μ PD77111
- ・ μ PD77112
- ・ μ PD77113
- ・ μ PD77114
- ・ μ PD77116 (開発中)

(2) 必要メモリ

表 1 - 1 必要メモリの容量

メモリ	種別	ANNEX A[ワード]	ANNEX C[ワード]
命令メモリ	-	5.6 K	7.9 K
Xメモリ	RAM ^注	2.1 K	2.2 K
	ROM	3.9 K	5.6 K
Yメモリ	RAM ^注	2.0 K	2.0 K
	ROM	5.6 K	6.5 K

注 RAM領域中スクラッチ領域は ANNEX A では X : 1024 ワード , Y : 1024 ワード , ANNEX C では X : 1108 ワード , Y : 284 ワードです。

(3) ソフトウェア・ツール (Windows™ 版)

DSP ツール : WB77016 (ワークベンチ)
 HSM77016 (ハイスピード・シミュレータ)

1.5.3 性 能

【条 件】 DSP : μ PD77016 ファミリ (33 MHz 動作時 33 MIPS)

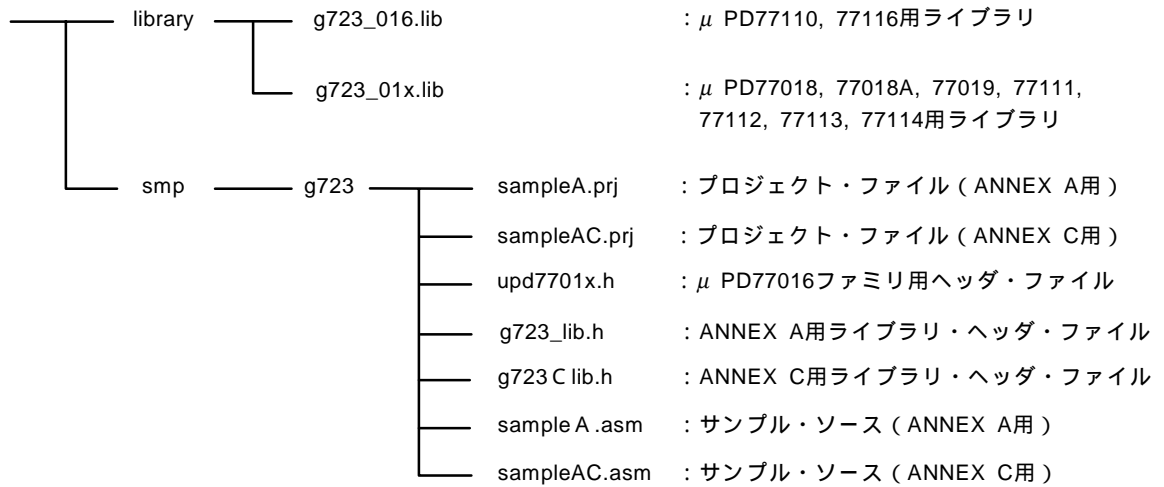
【1 フレームの処理をリアルタイム (30 ms) に実行するために必要な MIPS 値】

(G.723.1 ANNEX A) 圧縮処理 : 17.5 MIPS 伸長処理 : 1.5 MIPS

(G.723.1 ANNEX C) 圧縮処理 : +1 MIPS 伸長処理 : +1.5 MIPS

1.5.4 ディレクトリ構成

本モデルウェアのディレクトリ構成を次に示します。



ライブラリ・ファイル g723_016.lib, g723_01x.lib は、次のオブジェクト・ファイルを含んでいます。

ただし、G723_016.lib はμ PD77016 には使用できません。g723_01x.lib はμ PD77015, 77017 では使用できません。

- g723aenc.rel (G.723.1 ANNEX A エンコーダ)
- g723adec.rel (G.723.1 ANNEX A デコーダ)
- g723acom.rel (G.723.1 ANNEX A エンコーダ, デコーダ共有部分)
- g723cenc.rel (G.723.1 ANNEX C エンコーダ)
- g723cdec.rel (G.723.1 ANNEX C デコーダ)
- g723ccom.rel (G.723.1 ANNEX C エンコーダ, デコーダ共有部分)

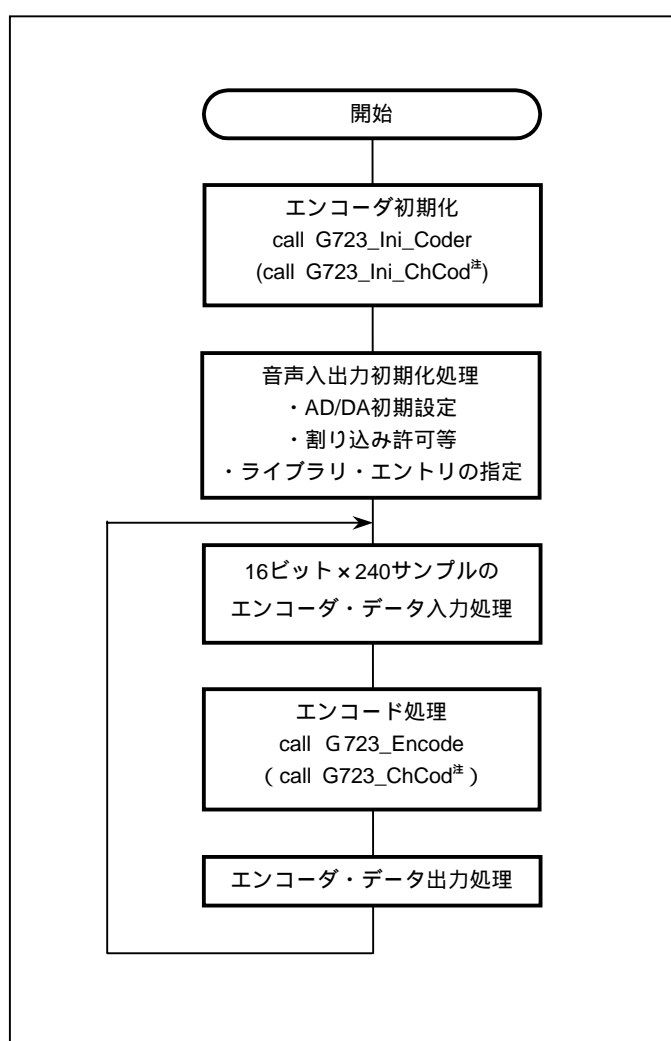
第 2 章 ライブラリ仕様

この章では本ミドルウェアの関数仕様と呼び出し規約について説明します。

2.1 G.723.1 音声コーデック処理フロー

本ミドルウェアを使用したアプリケーションの処理の流れを図 2 - 1 , 図 2 - 2 に示します。

図 2 - 1 アプリケーション処理フロー (エンコーダ)



注 “G723_Ini_ChCod”と“G723_ChCod”は ANNEX C 用の関数です。ANNEX C 使用時のみ“G723_Ini_Coder”および“G723_Encode”の後に call して使用します。

図2-2 アプリケーション処理フロー（デコーダ）



注 “G723_Ini_ChDec”と“G723_ChDec”は ANNEX C 用の関数です。ANNEX C 使用時のみ“G723_Ini_Decode”および“G723_Decode”の前に call して使用します。

2.2 関数仕様

2.2.1 エンコーダ初期化関数

エンコーダ初期化関数は、エンコーダ各定数の設定、係数テーブル、遅延バッファの初期化を行う関数です。

(1) スクラッチ領域初期化関数

【 分 類 】 スクラッチ領域初期化処理

【 関 数 名 】 G723_Start_Codec

【 機 能 概 要 】 G.723.1 ANNEX A で使用するスクラッチ領域を 0 クリアする。

【 形 式 】 call G723_Start_Codec

【 引 数 】 なし

【 返 却 値 】 なし

【 機 能 】 G.723.1 ANNEX A で使用するスクラッチ領域を 0 クリアします。

【使用レジスタ】 R0, DP0, DP4

【ハードウェア・リソースメント】

最大スタック・レベル	0
最大ループ・スタック・レベル	1
最大リピート回数	0
最大サイクル数	4000

(2) G.723.1 ANNEX A 用エンコーダ初期化関数

【分類】 エンコーダ初期化処理

【関数名】 G723_Ini_Coder

【機能概要】 G.723.1 ANNEX A エンコーダが使用する RAM 領域の初期化および各種パラメータを設定する。

【形式】 call G723_Ini_Coder

【引数】 X/Y Static 変数領域の先頭アドレス

```
例  r0l= StaticAreaX                ;
    *IOArea+G723_STATIC_X_PTR:x = r0l  ;
    r0l= StaticAreaY                ;
    *IOArea+G723_STATIC_Y_PTR:x = r0l  ;
    dp0=IOArea                      ;
```

【返却値】 なし

【機能】 G.723.1 ANNEX A エンコーダの初期化，各種パラメータの設定などを行う。

【使用レジスタ】 R0, R6, R7, DP0, DP4

【ハードウェア・リソースメント】

最大スタック・レベル	2
最大ループ・スタック・レベル	1
最大リピート回数	256
最大サイクル数	2000

(3) G.723.1 ANNEX C 用エンコーダ初期化関数

【分類】 エンコーダ初期化処理

【関数名】 G723_Ini_ChCod

【機能概要】 G.723.1 ANNEX C エンコーダが使用する RAM 領域の初期化および各種パラメータを設定する。

【形式】 G723_Ini_ChCod

【引数】 入出力バッファの先頭アドレスおよび各設定パラメータ

```

例      clr(r0);

        /*...*/ *EncLineBuff+12:x = r0h;
        /*...*/ *EncLineBuff+13:x = r0h;
        /*---*/ dp0 = LibEntry;
        r0l = C_StaticAreaX;
        /*...*/ *dp0++ = r0l;
        r0l = C_StaticAreaY;
        /*...*/ *dp0++ = r0l;
        r0l = EncLineBuff;
        /*...*/ *dp0++ = r0l;    // Input buffer
        r0l = EncChanBuff;
        /*...*/ *dp0++ = r0l;    // Output buffer
        r0l = 1;
        /*...*/ *dp0++ = r0l;    // EncodeSW (0:off 1:on)
        r0l = 1;
        /*...*/ *dp0++ = r0l;    // High/Low select (0:low 1:high)
        /*---*/ dp4 = BitrateConfig;
        /* */ *dp0++ = r0h;    // number of octets(dummy access)
        /* */ r0 = *dp4++;
        rep 6;
                /* */dp0++ = r0h          r0 = *dp4++;
        /*---*/ dp0 = LibEntry;
  
```

【返却値】 なし

【機能】 G.723.1 ANNEX C エンコーダの初期化，各種パラメータの設定などを行う。

【使用レジスタ】 R0, DP0, DP1, DP4

【ハードウェア・リソースメント】

最大スタック・レベル	0
最大ループ・スタック・レベル	0
最大リピート回数	15
最大サイクル数	34

2.2.2 デコーダ初期化関数

デコーダ初期化関数は、デコーダ各定数の設定、係数テーブル、遅延バッファの初期化を行う関数です。

(1) G.723.1 ANNEX A 用デコーダ初期化関数

【分類】 デコーダ初期化処理

【関数名】 G723_Ini_Decod

【機能概要】 G.723.1 ANNEX A デコーダの使用する RAM 領域の初期化および各種パラメータを設定する。

【形式】 call G723_Ini_Decod

【引数】 X/Y Static 変数領域の先頭アドレス

```
例  r0l= StaticAreaX                ;
    *IOArea+G723_STATIC_X_PTR:x = r0l ;
    r0l= StaticAreaY                ;
    *IOArea+G723_STATIC_Y_PTR:x = r0l ;
    dp0=IOArea                       ;
```

【返却値】 なし

【機能】 G.723.1 ANNEX A デコーダの初期化、各種パラメータの設定などを行う。

【使用レジスタ】 R0, R1, R6, R7, DP0, DP4, DP5

【ハードウェア・リソースメント】

最大スタック・レベル	2
最大ループ・スタック・レベル	1
最大リピート回数	145
最大サイクル数	1000

(2) G.723.1 ANNEX C 用デコーダ初期化関数

【分類】 デコーダ初期化処理

【関数名】 G723_Ini_ChDec

【機能概要】 G.723.1 ANNEX C デコーダの使用する RAM 領域の初期化および各種パラメータを設定する。

【形式】 call G723_Ini_ChDec

【引数】 入出力バッファの先頭アドレスおよび各設定パラメータ

```
例      clr(r0);

        /*---*/ dp0 = LibEntry + G723C_LIBENTRY_DECOFS;
        r0l = DecLineBuff;
        /*...*/ *dp0++ = r0l;    // Output buffer
        r0l = DecChanBuff;
        /*...*/ *dp0++ = r0l;    // Input buffer
        /* */ *dp0++ = r0h;    // clear error flag
        /*---*/ dp4 = BitrateConfig;
        /* */ *dp0++ = r0h;    // reserved
        /* */ r0 = *dp4++;

        rep 6;

        /* */ *dp0++ = r0h    r0 = *dp4++;

        /*---*/ dp0 = LibEntry;
```

【返却値】 なし

【機能】 G.723.1 ANNEX C デコーダの初期化，各種パラメータの設定などを行う。

【使用レジスタ】 R0, DP0, DP1, DP4

【ハードウェア・リソースメント】

最大スタック・レベル	0
最大ループ・スタック・レベル	0
最大リピート回数	15
最大サイクル数	34

2.2.3 エンコーダ関数

エンコーダ関数は、入力した 240 サンプルの音声信号を 189, 159 または 32 ビットに圧縮した信号を生成します。ただし、ANNEX C は可変レートです。

(1) G.723.1 ANNEX A 用エンコーダ関数

【 分 類 】 エンコード処理部

【 関 数 名 】 G723_Encode

【 機能概要 】 240 サンプル×16 ビットを 189,159 または 32 ビットに圧縮。

【 形 式 】 call G723_Encode

【 引 数 】 入出力バッファの先頭アドレスおよび各設定パラメータ

```
例  r01= EncPcmBuff                ; set IO parameter
    *IOArea+G723_ENC_BUFF_PTR:x = r01 ;
    r01= EncLineBuff              ;
    *IOArea+G723_VOUT_PTR:x= r01   ;
    *IOArea+G723_USE_VAD:x= ???    ; 0)VAD off, 1)VAD on
    *IOArea+G723_USE_HP:x= ???    ; 0)Hpf off, 1)Hpf on
    *IOArea+G723_WRK_RATE_E:x= ??? ; 0)6.3kbps, 1)5.3kbps
    dp0=IOArea                    ;
```

【 返 却 値 】 フレーム・タイプおよび Clipped サブフレーム数

```
例  ??? = *IOArea+G723_E_FRAME_TYPE:x ; 0)NoTx ,1)Active ,2)SID
    ??? = *IOArea+G723_COUNT_CLIP:x   ; clipped sub-frames 0..3
```

圧縮データ：引数で設定した出力バッファ

【 機 能 】 コーデックからの入力信号（240 サンプル×16 ビット）を 189, 159 または 32 ビットに圧縮します。

【使用レジスタ】 R0, R1, R2, R3, R4, R5, R6, R7, DP0, DP1, DP2, DP3, DP4, DP5, DP6, DP7, DN0, DN1, DN2, DN3, DN4, DN5, DN6, DN7, DMX, DMY

【ハードウェア・リソースメント】

最大スタック・レベル	7
最大ループ・スタック・レベル	4
最大リピート回数	256
最大 MIPS 値	17.5

(2) G.723.1 ANNEX C 用エンコーダ関数

【 分類 】 エンコード処理部

【 関数名 】 G723_ChCod

【 機能概要 】 G723_Encode 関数で圧縮したデータにチャンネル符号化を行う。

【 形式 】 call G723_ChCod

【 引数 】 入出力バッファの先頭アドレスおよび各設定パラメータ

```

例      clr(r0);

        /*...*/ *EncLineBuff+12:x = r0h;
        /*...*/ *EncLineBuff+13:x = r0h;
        /*---*/ dp0 = LibEntry;

        r0l = C_StaticAreaX;
        /*...*/ *dp0++ = r0l;

        r0l = C_StaticAreaY;
        /*...*/ *dp0++ = r0l;

        r0l = EncLineBuff;
        /*...*/ *dp0++ = r0l;    // Input buffer

        r0l = EncChanBuff;
        /*...*/ *dp0++ = r0l;    // Output buffer

        r0l = 1;
        /*...*/ *dp0++ = r0l;    // EncodeSW (0:off 1:on)

        r0l = 1;
        /*...*/ *dp0++ = r0l;    // High/Low select (0:low 1:high)
        /*---*/ dp4 = BitrateConfig;
        /* */   *dp0++ = r0h;    // number of octets(dummy access)
        /* */   r0 = *dp4++;

        rep 6;

                /* *//dp0++ = r0h          r0 = *dp4++;

        /*---*/ dp0 = LibEntry;

```

【 返却値 】 オクテット数: r2l

圧縮データ: 引数で設定した出力バッファ

【 機能 】 コーデックからの入力信号 (240 サンプル×16 ビット) を可変レートで圧縮する。

【使用レジスタ】 R0, R1, R2, R3, R4, R5, R6, R7, DP0, DP1, DP2, DP3, DP4, DP5, DP6, DP7, DN0, DN1, DN2, DN3, DN4, DN5, DN6, DN7, DMX, DMY

【ハードウェア・リソースメント】

最大スタック・レベル	3
最大ループ・スタック・レベル	2
最大リピート回数	636
最大 MIPS 値	1.0

2.2.4 デコーダ関数

デコーダ関数は、189, 159 または 32 ビットに圧縮された信号を 240 サンプル×16 ビットの音声データに伸長します。

(1) G.723.1 ANNEX A 用デコーダ関数

【 分 類 】 デコード処理部

【 関 数 名 】 G723_Decode

【 機 能 概 要 】 189, 159 または 32 ビットに圧縮されたデータを 240 サンプル×16 ビットに伸長。

【 形 式 】 call G723_Decode

【 引 数 】 入出力バッファの先頭アドレスおよび各設定パラメータ

```
例 r0l= DecPcmBuff                ; set IO parameter
    *IOArea+G723_DEC_BUFF_PTR:x = r0l    ;
    r0l= DecLineBuff              ;
    *IOArea+G723_VINP_PTR:x= r0l        ;
    *IOArea+G723_USE_PF:x= ???         ; 0)Psf off, 1)Psf on
    *IOArea+G723_CRC_RESULT:x = ???     ;CRC_result,0)Normal,else)Err

    dp0=IOArea                    ;
```

【 返 却 値 】 フレーム・タイプ, ワーク・レートおよびエラー・フレーム数

```
例 ???= *IOArea+G723_D_FRAME_TYPE:x    ; 0)NoTx ,1)Active ,2)SID
    ???= *IOArea+G723_WRK_RATE_D:x     ; 0)6.3kbps, 1)5.3kbps
    ???= *IOArea+ERR_FRM_COUNT:x      ; Num of Error frame 0..0x7ff
```

伸長データ：引数で設定した出力バッファ

【 機 能 】 189, 159 または 32 ビットに圧縮されたデータを音声信号 (240 サンプル×16 ビット) に伸長する。

【使用レジスタ】 R0, R1, R2, R3, R4, R5, R6, R7, DP0, DP1, DP2, DP3, DP4, DP5, DP6, DP7, DN0, DN1, DN2, DN3, DN4, DN5, DN6, DN7, DMX

【ハードウェア・リソースメント】

最大スタック・レベル	4
最大ループ・スタック・レベル	3
最大リピート回数	240
最大 MIPS 値	1.5

(2) G.723.1 ANNEX C 用デコーダ関数

【分類】 デコード処理部

【関数名】 G723_ChDec

【機能概要】 チャネル復号化を行い, G723_Decode 関数に引き渡す符号を作成する。

【形式】 call G723_ChDec

【引き数】 入出力バッファの先頭アドレスおよび各設定パラメータ

```

例      clr(r0);

        /*---*/ dp0 = LibEntry + G723C_LIBENTRY_DECOFS;
        r0l = DecLineBuff;
        /*...*/ *dp0++ = r0l;           // Output buffer
        r0l = DecChanBuff;
        /*...*/ *dp0++ = r0l;           // Input buffer
        /* */ *dp0++ = r0h;           // clear error flag
        /*---*/ dp4 = BitrateConfig;
        /* */ *dp0++ = r0h;           // reserved
        /* */ r0 = *dp4++;

        rep 6;

        /* */ *dp0++ = r0h    r0 = *dp4++;
        /*---*/ dp0 = LibEntry;

```

【返却値】 mode (0: 6.3 K, 1: 5.3 K, 2: SID): r0l

ビット・ストリームの最終アドレス+1: r1l

伸長データ: 引数で設定した出力バッファ

【機能】 可変レートで圧縮されたデータを音声信号(16ビット×240サンプル)に伸長する。

【使用レジスタ】 R0, R1, R2, R3, R4, R5, R6, R7, DP0, DP1, DP2, DP3, DP4, DP5, DP6, DP7, DN0, DN1, DN2, DN3, DN4, DN5, DN6, DN7, DMX

【ハードウェア・リソースメント】

最大スタック・レベル	4
最大ループ・スタック・レベル	3
最大リピート回数	15
最大 MIPS 値	1.5

2.2.5 バージョン取得関数

バージョン取得関数は、ライブラリのバージョンを返します。

- 【 分 類 】 バージョン情報取得
- 【 関 数 名 】 G723_GetVersion
- 【 機 能 概 要 】 ライブラリのバージョンを返す。
- 【 形 式 】 call G723_GetVersion
- 【 引 数 】 なし
- 【 返 却 値 】 R0H メジャー・バージョン番号
 R0L マイナ・バージョン番号
- 【 機 能 】 本ミドルウェアのバージョン番号を 32 ビットの値で返します。
 例 R0 = 0x00'0x0001'0x0100 の場合
 バージョン : V1.01
- 【使用レジスタ】 R0

2.3 外部インタフェース説明

ここでは、G.723.1 ANNEX A 用関数 G723_Encode , G723_Decode と G.723.1 ANNEX C 用関数 G723_ChCod , G723_ChDec の各関数コール時に引き渡すパラメータを定義します。

2.3.1 G.723.1 ANNEX A 用関数の引き渡しパラメータ

スクラッチ領域の確保は次のとおりに行ってください。スクラッチ領域 X, Y には “align at 0x20” が必要です。

- ・スクラッチ領域 X : X メモリ上に “lib_Scratch_x” というレベル名で 1024 ワードの大きさ
- ・スクラッチ領域 Y : Y メモリ上に “lib_Scratch_y” というレベル名で 1024 ワードの大きさ

スタティック領域の確保は次のとおりに行ってください。スタティック領域 X には “align at 0x20”, スタティック領域 Y には “align at 0x200” が必要です。

- ・ “lib_Scratch_x” と “lib_Scratch_y” は同じアドレスになるようにしてください。

表 2 - 1 G.723.1 ANNEX A 用関数の引き渡しパラメータ

機能	分類	シンボル ^{注6} (オフセット・アドレス)	設定値
エンコーダ/デコーダ共通パラメータ			
スタティック X メモリ領域	Control input ^{注1}	G723_STATIC_X_PTR	X メモリ・アドレス
スタティック Y メモリ領域	Control input ^{注1}	G723_STATIC_Y_PTR	Y メモリ・アドレス
エンコーダ・パラメータ			
エンコーダ入力バッファ	Control input	G723_ENC_BUFF_PTR	X メモリ・アドレス
エンコーダ出力バッファ	Control input	G723_VOUT_PTR	X メモリ・アドレス
コーディング・ビット・レート	Control input	G723_WRK_RATE_E	0 : 6.3 Kbps, 1 : 5.3 Kbps
VAD 制御	Control input ^{注2}	G723_USE_VAD	0 : VAD オフ, 1 : VAD オン
Preprocessing HPF 制御	Control input ^{注2}	G723_USE_HP	0 : HPF オフ, 1 : HPF オン
エンコード・フレーム・タイプ	Status output	G723_E_FRAME_TYPE	0 : 非伝送, 1 : 有音, 2 : SID
クリップ・フレーム数	Status output	G723_COUNT_CLIP	0-3
デコーダ・パラメータ			
デコーダ出力バッファ	Control input	G723_DEC_BUFF_PTR	X メモリ・アドレス
デコーダ入力バッファ	Control input	G723_VINP_PTR	X メモリ・アドレス
Postprocessing filter 制御	Control input ^{注3}	G723_USE_PF	0 : フィルタ・オフ, 1 : フィルタ・オン
CRC フラグ	Control input	G723_CRC_RESULT	0 : ノーマル, else : エラー
デコード・フレーム・タイプ	Status output	G723_D_FRAME_TYPE	0 : 非伝送, 1 : 有音, 2 : SID
デコード・ビット・レート	Status output ^{注4}	G723_WRK_RATE_D	0 : 6.3 Kbps, 1 : 5.3 Kbps
フレーム・エラー・カウント	Status output ^{注5}	G723_ERR_FRM_COUNT	0-0x7FFF

注 1. G723_Start_Codec 以降は変更禁止です。

2. エンコード中は変更禁止です。
3. デコード中は変更禁止です。
4. 有音フレーム時のみ有効です。
5. 正常フレームを受信した場合に 0 にリセットされます。
6. シンボルは各関数 call 時の DP0 の値からのオフセットを示します。

2.3.2 G.723.1 ANNEX C 用関数の引き渡しパラメータ

スクラッチ領域の確保は次のとおりに行ってください。スクラッチ領域 X, Y には “align at 0x20” が必要です。

- ・スクラッチ領域 X: X メモリ上に “lib_Scratch_x” というラベル名で 1108 ワードの大きさ
- ・スクラッチ領域 Y: Y メモリ上に “lib_Scratch_y” というラベル名で 1024 ワードの大きさ

スクラッチ領域は, ANNEX A と ANNEX C のうち大きい方 (x は ANNEX C, y は ANNEX A を示します) の大きさを確保してください。

表 2 - 2 G.723.1 ANNEX C 用関数の引き渡しパラメータ

(1/2)

機能	分類	シンボル ^{※4} (オフセット・アドレス)	設定値
エンコーダ/デコーダ共通パラメータ			
スタティック X メモリ・エリア	Control input ^{※1}	G723C_LibStatic_X	X メモリ・アドレス
スタティック Y メモリ・エリア	Control input ^{※1}	G723C_LibStatic_Y	Y メモリ・アドレス
エンコーダ・パラメータ			
エンコーダ入力バッファ	Control input	G723C_LibEncInBuffer	X メモリ・アドレス
エンコーダ出力バッファ	Control input	G723C_LibEncOutBuffer	X メモリ・アドレス
符号化用動作制御	Control input	G723C_LibEncoderSW	0: オフ, 1: オン
ハイ・レベル/ロウ・レベル切り替え	Control input ^{※2}	G723C_LibEncHighLow	0: ロウ・レベル, 1: ハイ・レベル
符号化結果のデータ・サイズ	Control input	G723C_LibEncOutSize	ビット数
6.3 Kbps で符号化時のチャンネル・レートの上限	Control input ^{※3}	G723C_LibEncULim63	bps 数
6.3 bps で符号化時のチャンネル・レートの下限	Control input ^{※3}	G723C_LibEncLLim63	bps 数
5.3 Kbps で符号化時のチャンネル・レートの上限	Control input ^{※3}	G723C_LibEncULim53	bps 数
5.3 Kbps で符号化時のチャンネル・レートの下限	Control input ^{※3}	G723C_LibEncLLim53	bps 数
SID で符号化時のチャンネル・レートの上限	Control input ^{※3}	G723C_LibEncULimSID	bps 数
SID で符号化時のチャンネル・レートの下限	Control input ^{※3}	G723C_LibEncLLimSID	bps 数

注 1. G723_Start_Codec 以降は変更禁止です。

2. ハイ・レベル/ロウ・レベルの切り替えスイッチは 6.3 Kbps/5.3 Kbps/SID に共通です。
3. チャンネル・レートの上限, 下限に設定する値は符号化時/復号化時に次の範囲内でなければなりません。

6.3 Kbps の場合 7034 以上 20634 以下

5.3 Kbps の場合 6000 以上 17534 以下

SID の場合 1600 以上 4334 以下

4. シンボルは各関数 call 時の DP0 の値からのオフセットを示します。

表 2 - 2 G.723.1 ANNEX C 用関数の引き渡しパラメータ

(2/2)

機能	分類	シンボル ^{注3} (オフセット・アドレス)	設定値
デコーダ・パラメータ			
デコーダ出力バッファ	Control input	G723C_LibDecOutBuffer	Xメモリ・アドレス
デコーダ入力バッファ	Control input	G723C_LibDecInBuffer	Xメモリ・アドレス
復号結果エラー表示	Status output ^{注1}	G723C_LibDecErrorInfo	ビット2: FII, ビット1: EFI ビット0: BFI
予約	-	G723C_LibDecReserved	-
6.3 Kbps で復号時のチャンネル・レートの上限	Control input ^{注2}	G723C_LibDecULim63	bps 数
6.3 Kbps で復号時のチャンネル・レートの下限	Control input ^{注2}	G723C_LibDecLLim63	bps 数
5.3 Kbps で復号時のチャンネル・レートの上限	Control input ^{注2}	G723C_LibDecULim53	bps 数
5.3 Kbps で復号時のチャンネル・レートの下限	Control input ^{注2}	G723C_LibDecLLim53	bps 数
SID で復号時のチャンネル・レートの上限	Control input ^{注2}	G723C_LibDecULimSID	bps 数
SID で復号時のチャンネル・レートの上限	Control input ^{注2}	G723C_LibDecLLimSID	bps 数

注 1. 復号時は、復号結果エラー表示の値を、復号化用符号データの CRC 値として記入します。

なお、このエラー表示により該当するフレーム分の音声復号処理はフレーム補償処理となります。

2. チャンネル・レートの上限、下限に設定する値は符号化時 / 復号化時に次の範囲内であればなりません。

6.3 Kbps の場合 7034 以上 20634 以下

5.3 Kbps の場合 6000 以上 17534 以下

SID の場合 1600 以上 4334 以下

3. シンボルは各関数 call 時の DP0 の値からのオフセットを示します。

2.4 入出力データ・フォーマット

ここでは ITU-T 勧告 G.723.1 の入出力データのフォーマットを示します。

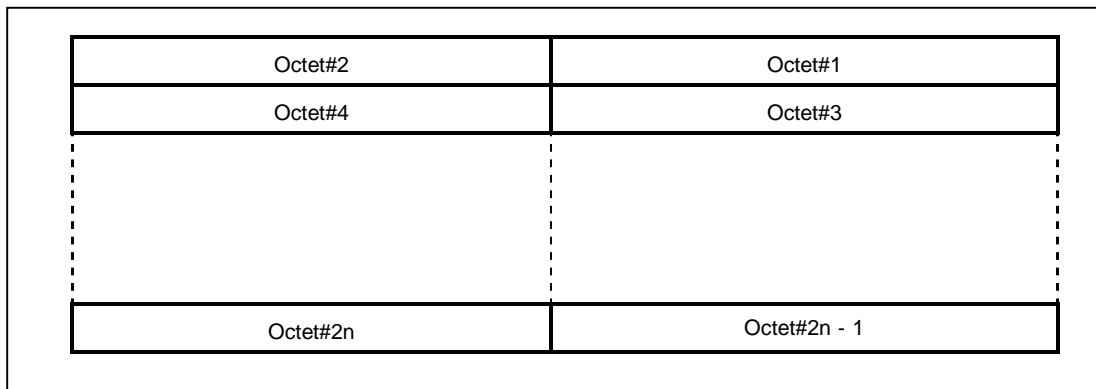
2.4.1 エンコーダ入力/デコーダ出力データ・フォーマット

エンコーダ入力データは、8 kHz でサンプリングした PCM データで、16 ビット×240 サンプルの形で入力してください。また、デコーダ出力データは 8 kHz でサンプリングした 16 ビット×240 サンプルの PCM データの形で出力されます。

2.4.2 エンコーダ出力/デコーダ入力データ・フォーマット

エンコーダ出力およびデコーダ入力用圧縮データのフォーマットを次に示します。

図 2 - 3 圧縮データ・フレーム・フォーマット



備考 2n : フレーム・バイト数

表 2 - 3 G.723.1 ANNEX A 圧縮データのビット・レートとフレーム・バイト数

ビット・レート	フレーム・バイト数
6.3 Kbps	24
5.3 Kbps	20
SID	4

表 2 - 4 G.723.1 ANNEX C 圧縮データのビット・レートとフレーム・バイト数

ビット・レート	フレーム・バイト数
6.3 Kbps	27-78
5.3 Kbps	23-66
SID	6-17

(1) 有音データの圧縮フォーマット

エンコーダ関数の出力, およびデコーダ関数の入力となる, 6.3 Kbps の有音圧縮データ 189 ビット・データ, および 5.3 Kbps の有音圧縮データ 159 ビット・データのフォーマットを表 2 - 5, 2 - 6 に示します。圧縮データについての詳細は, ITU-T 勧告を参照してください。

表 2 - 5 6.3 Kbps の有音圧縮データ・フォーマット

TRANSMITTED OCTETS	PARx_By, ..., ...
1	LPC_B5 - LPC_B0, VADFLAG_B0, RATEFLAG_B0
2	LPC_B13 - LPC_B6
3	LPC_B21 - LPC_B14
4	ACL0_B5 - ACL0_B0, LPC_B23, LPC_B22
5	ACL2_B4 - ACL2_B0, ACL1_B1, ACL1_B0, ACL0_B6
6	GAIN0_B3 - GAIN0_B0, ACL3_B1, ACL3_B0, ACL2_B6, ACL2_B5
7	GAIN0_B11 - GAIN0_B4
8	GAIN1_B7 - GAIN1_B0
9	GAIN2_B3 - GAIN2_B0, GAIN1_B11 - GAIN1_B8
10	GAIN2_B11 - GAIN2_B4
11	GAIN3_B7 - GAIN3_B0
12	GRID3_B0, GRID2_B0, GRID1_B0, GRID0_B0, GAIN3_B11 - GAIN3_B8
13	MSBPOS_B6 - MSBPOS_B0, UB [*]
14	POS0_B1, POS0_B0, MSBPOS_B12 - MSBPOS_B7
15	POS0_B9 - POS0_B2
16	POS1_B2, POS1_B0, POS0_B15 - POS0_B10
17	POS1_B10, POS1_B3
18	POS2_B3 - POS2_B0, POS1_B13 - POS1_B11
19	POS2_B11 - POS2_B4
20	POS3_B3 - POS3_B0, POS2_B15 - POS2_B12
21	POS3_B11 - POS3_B4
22	PSIG0_B5 - PSIG0_B0, POS3_B13, POS3_B12
23	PSIG2_B2 - PSIG2_B0, PSIG1_B4 - PSIG1_B0
24	PSIG3_B4 - PSIG3_B0, PSIG2_B5 - PSIG2_B3

注 UB は未使用ビット (値は 0) を示します。

表2-6 5.3 Kbpsの有音圧縮データ・フォーマット

TRANSMITTED OCTETS	PARx_By, ..., ...
1	LPC_B5 - LPC_B0, VADFLAG_B0, RATEFLAG_B0
2	LPC_B13 - LPC_B6
3	LPC_B21 - LPC_B14
4	ACL0_B5 - ACL0_B0, LPC_B23, LPC_B22
5	ACL2_B4 - ACL2_B0, ACL1_B1, ACL1_B0, ACL0_B6
6	GAIN0_B3 - GAIN0_B0, ACL3_B1, ACL3_B0, ACL2_B6, ACL2_B5
7	GAIN0_B11 - GAIN0_B4
8	GAIN1_B7 - GAIN1_B0
9	GAIN2_B3 - GAIN2_B0, GAIN1_B11 - GAIN1_B8
10	GAIN2_B11 - GAIN2_B4
11	GAIN3_B7 - GAIN3_B0
12	GRID3_B0, GRID2_B0, GRID1_B0, GRID0_B0, GAIN3_B11 - GAIN3_B8
13	POS0_B7 - POS0_B0
14	POS1_B3 - POS1_B0, POS0_B11 - POS0_B8
15	POS1_B11 - POS1_B4
16	POS2_B7 - POS2_B0
17	POS3_B3 - POS3_B0, POS2_B11 - POS2_B8
18	POS3_B11 - POS3_B4
19	PSIG1_B3 - PSIG1_B0, PSG0_B3 - PSG0_B0
20	PSIG3_B3 - PSIG3_B0, PSIG2_B3 - PSIG2_B0

(2) SID フレームの圧縮フォーマット

エンコーダ関数の出力，およびデコーダ関数の入力となる，32 ビット無音データのフォーマットを表 2 - 7 に示します。圧縮データについての詳細は，ITU-T 勧告を参照してください。

ただし，非伝送の場合は“ TRANSMITTED OCTETS=1 ”に“ 0x3 ”を，“ TRANSMITTED OCTETS=2-4 ”に“ 0x0 ”を返します。

表 2 - 7 無音圧縮データのビット割り当て

TRANSMITTED OCTETS	PARx_By, ...
1	LPC_B5 - LPC_B0, VADFLAG_B0, RATEFLAG_B0
2	LPC_B13 - LPC_B6
3	LPC_B21 - LPC_B14
4	GAIN_B5 - GAIN_B0, LPC_B23, LPC_B22

(3) G.723.1 ANNEX C (チャンネル・コーデック) 付加圧縮データ・フォーマット

エンコーダ関数の出力，およびデコーダ関数の入力となる，ANNEX C のデータ・フォーマットを表 2 - 8 に示します。圧縮データについての詳細は，ITU-T 勧告を参照してください。

表 2 - 8 G.723.1 ANNEX C 圧縮データのビット割り当て

TRANSMITTED OCTETS	チャンネル・ビット
1	UCB[7], UCB[6], UCB[5], UCB[4], UCB[3], UCB[2], UCB[1], UCB[0]
2	U[2], U[1], U[0], UCB[12], UCB[11], UCB[10], UCB[9], UCB[8]
3	U[10], U[9], U[8], U[7], U[6], U[5], U[4], U[3]
Mp / 8 + 2	..., ..., U[Mp], U[Mp - 1], U[Mp - 2], ..., ..., ...
:	:
MAll / 8 + 2	UB, UB ,UB, U[MAll - 1], U[MAll - 2], U[MAll - 3], U[MAll - 4], U[MAll - 5]

備考 UB は未使用ビット (値は 0) を示します。

保護されたビット列のビット・レートは (MAll + 13) × 1000 / 30 bps です。

[メ モ]

第3章 インストール

3.1 インストール手順

本ミドルウェアの提供媒体は、3.5 インチ・フロッピー・ディスク (1.44 MB) のみです。ホスト・マシンへのインストールの手順を次に示します。

(1) 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。ソフトウェア・ツールが使用しているディレクトリ (例 : C:\DSPTools) の下にファイルをコピーします。ここでは、AドライブからCドライブへコピーした場合を示します。

```
a:>xcopy /s *.* c:\DSPTools <CR>
```

(2) ファイルがコピーされたことを確認します。各ディレクトリについては、1.5.4 ディレクトリ構成を参照してください。

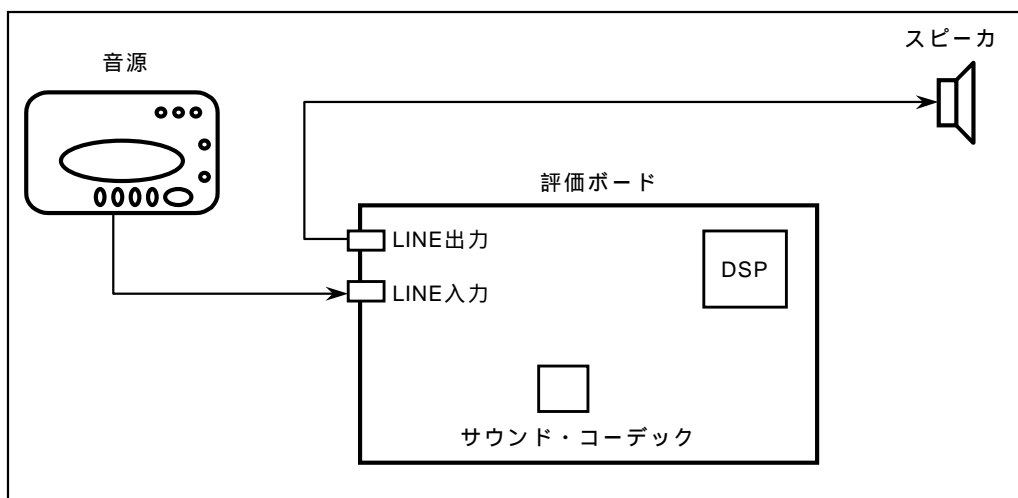
```
a:>dir c:\DSPTools <CR>
```

3.2 サンプル作成手順

sample のディレクトリに、サンプル・プログラムを格納しています (sampleA.asm, sampleAC.asm のソース・プログラムについては、付録 サンプル・プログラム・ソース参照)。

サンプル・プログラムは、実際に CD や DAT などの音源、スピーカなどを接続して、リアルタイムで音源の圧縮伸長を行うことができます。

図 3 - 1 サンプル・プログラム評価システム



次に本ミドルウェアのサンプル・プログラムのビルド方法について例を示します。

(1) WB77016 (ワークベンチ) を起動します。

(2) sampleA.prj (sampleAC.prj) プロジェクトを開きます。

例 Project メニューの Open Project コマンドで sampleA.prj (sampleAC.prj) を指定します。

(3) ビルドを実行し, sampleA.Ink (sampleAC.Ink) が生成されたことを確認します。

例 Make メニューの Build All コマンドを選択すると, sampleA.Ink (sampleAC.Ink) ファイルが生成されます。

(4) 評価ボードへプログラムをダウンロードします。

3.3 シンボル命名規約

本ライブラリ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を組み合わせで使用するときは, 重複しないようにしてください。

表 3 - 1 シンボル名

分類	規約	
	G.723.1 ANNEX A	G.723.1 ANNEX C
関数名	G723_XXXX	G723_ChXXXX
マクロ, 定数名	G723_XXXX	
セクション名	__G723_XXXX	__G723_CXXXX

付録 サンプル・プログラム・ソース

付.1 G.723.1 + ANNEX A 用 (sampleA.asm)

```
/*-----*/
/* File Information */
/*-----*/
/* Name      : sampleA.asm */
/* Type      : ASM Programming Language source code */
/* Version   : V1.00 */
/* Date      : 1998 June 10 */
/* CPU       : uPD7701x Family */
/* Assembler : WB77016 Ver2.21 */
/* About     : NEC uPD7701x Family G723.1 Speech CODEC Middle-Ware Library */
/*           : Sample program [1ch, OFF-LINE mode] */
/*-----*/
/* Copyright (c) NEC Corporation 1995,1996,1997,1998 */
/* NEC CONFIDENTIAL AND PROPRIETAR */
/* All rights reserved by NEC Coporation */
/* Use of copyright notice does not evidence publication */
/*-----*/

/*---- include files ----*/
#include "uPD77016.h"
#include "g723_lib.h"

/*---- Global Variables ----*/
public lib_Scratch_x      /* Scrach area X */
public lib_Scratch_y     /* Scrach area Y */

/*---- Global Functions ----*/
extrn G723_GetVersion    /* Get Version function */
extrn G723_Start_Codec  /* Start G723 functions */
extrn G723_Ini_Coder    /* Initialize G723 Coder variables */
extrn G723_Ini_Decod    /* Initialize G723 Decoder variables */
extrn G723_Encode       /* Encode 1 Frame */
extrn G723_Decode       /* Decode 1 Frame */

$EJECT
/**/
/*****
/* Vector Table */
/*****
Vct_Ix  imseg at 0x0200
        call Init_Proc          ; reset
        jmp G723_Proc          ;
        nop                    ;
        nop                    ;
        reti                    ; not used
        nop                    ;
        nop                    ;
        nop                    ;
        reti                    ; not used
        nop                    ;
        nop                    ;
```

```

nop                                ;
reti                               ; not used
nop                                ;
nop                                ;
nop                                ;
reti                               ; INT0
nop                                ;
nop                                ;
nop                                ;
reti                               ; INT1
nop                                ;
nop                                ;
nop                                ;
reti                               ; INT2
nop                                ;
nop                                ;
nop                                ;
reti                               ; INT3
nop                                ;
nop                                ;
nop                                ;
reti                               ; SI#1
nop                                ;
nop                                ;
nop                                ;
reti                               ; SO#1
nop                                ;
nop                                ;
nop                                ;
reti                               ; SI#2
nop                                ;
nop                                ;
nop                                ;
reti                               ; SO#2
nop                                ;
nop                                ;
nop                                ;
reti                               ; HOST IN
nop                                ;
nop                                ;
nop                                ;
reti                               ; HOST OUT
nop                                ;
nop                                ;
nop                                ;
reti                               ; not used
nop                                ;
nop                                ;
nop                                ;
reti                               ; not used
nop                                ;
nop                                ;
nop                                ;

/*****
/* G723 OFF-Line Process                                     */
/*****
Smp_Ix  imseg
Init_Proc:
        clr(r0)                                ; No Wait
        *DWTR:x = r01                          ;

```



```

        *IWTR:x = r01                                ;

        r01 = HST_WAIT                               ; Wait mode
        *HST:x = r01                                ;
        ret                                          ;

$EJECT
/**/
/*-----*/
/* Scratch_Area / Static_Area / IO_Buffer          */
/*      !!! 'lib_Scratch_x' must be equal to 'lib_Scratch_y' !!!      */
/*-----*/
Scratch_X  xramseg align at 0x020
lib_Scratch_x:
        ds      G723_SCRATCH_X_BUFSIZE

Scratch_Y  yramseg align at 0x020
lib_Scratch_y:
        ds      G723_SCRATCH_Y_BUFSIZE

Static_X   xramseg align at 0x020
StaticAreaX:
        ds      G723_STATIC_X_BUFSIZE

Static_Y   yramseg align at 0x200
StaticAreaY:
        ds      G723_STATIC_Y_BUFSIZE

IO_Area_X  xramseg
IOArea:
        ds      G723_IOTABLE_SIZE

IO_Buff_X  xramseg
EncPcmBuff:
        ds      G723_FRAME
DecPcmBuff:
        ds      G723_FRAME
EncLineBuff:
        ds      24/2
DecLineBuff:
        ds      24/2

/*-----*/
/* Variables                                     */
/*-----*/
Smp_Xe1    xramseg
WrkMode:
        ds      1

/*****
/* G723 OFF-Line Process                          */
*****/
Smp_Ix     imseg
G723_Proc:
        call G723_Start_Codec                    ; Need only for Simulation test

        r01 = *HDT:x                             ; WrkMode 1)Cod ,2)Dec
        *WrkMode:x = r01                         ;

;;        call G723_GetVersion                    ; Library Version

```

```

;;      *HDT:x = r0h                ; comment_out for
;;      *HDT:x = r0l                ;      Simulator Test

r0l = StaticAreaX                  ;
*IOArea+G723_STATIC_X_PTR:x = r0l ; Static Area X
r0l = StaticAreaY                  ;
*IOArea+G723_STATIC_Y_PTR:x = r0l ; Static Area Y

r0l = EncPcmBuff                   ;
*IOArea+G723_ENC_BUFF_PTR:x = r0l ; PCM Buffer for Encode
r0l = EncLineBuff                  ;
*IOArea+G723_VOUT_PTR:x= r0l      ; Code Buffer for Encode
r0l = *HDT:x                       ;
*IOArea+G723_USE_HP:x= r0l        ; UseHp 0)No , 1)UseHp
r0l = *HDT:x                       ;
*IOArea+G723_USE_VAD:x= r0l      ; UseVAD 0)No , 1)UseHp

r0l = DecPcmBuff                   ;
*IOArea+G723_DEC_BUFF_PTR:x = r0l ; PCM buffer for Decode
r0l = DecLineBuff                  ;
*IOArea+G723_VINP_PTR:x = r0l    ; Code buffer for Decode
r0l = *HDT:x                       ;
*IOArea+G723_USE_PF:x= r0l      ; UsePf 0)No , 1)UsePf

r0l = *WrkMode:x                   ;
r0 = r0 & 0x0001                   ;
if( r0==0 ) jmp INI_LBC_DEC        ;
    /*---*/      dp0 = IOArea      ;
    call G723_Ini_Coder            ;
    jmp LBC_LOOP                  ;

INI_LBC_DEC:
    /*---*/      dp0 = IOArea      ;
    call G723_Ini_Decod           ;

LBC_LOOP:
    r0 = *HDT:x                   ; Control 0)Stop, 1)Continue
    if( r0==0 ) jmp LBC_END       ;

    r0l = *WrkMode:x              ;
    r0 = r0 & 0x0001              ;
    if( r0==0 ) jmp LBC_DEC       ;
        call GetPcmData           ;
        r0l = *HDT:x              ;
        *IOArea+G723_WRK_RATE_E:x = r0l ; WrkRate 0)6.3, 1)5.3 [kbps]

        /*---*/      dp0 = IOArea ;
        call G723_Encode          ;

        call PutLineData          ;
;;      r0l = *IOArea+G723_E_FRAME_TYPE:x ; comment_out for
;;      *HDT:x = r0l              ;      Simulator Test
        jmp LBC_NEXT              ;

LBC_DEC:
    r0l = *WrkMode:x              ;
    r0 = r0 & 0x0002              ;
    if( r0==0 ) jmp LBC_NEXT     ;
        call GetLineData          ;
        r0l = *HDT:x              ;
        *IOArea+G723_CRC_RESULT:x = r0l ; CRC_result 0)Normal, else)Err

```

```

        /*---*/      dp0 = IOArea          ;
        call G723_Decode                    ;

        call PutPcmData                    ;
;;      r01 = *IOArea+G723_D_FRAME_TYPE:x ; comment_out for
;;      *HDT:x = r01                        ; Simulator Test
;;      r01 = *IOArea+G723_WRK_RATE_D:x    ;
;;      *HDT:x = r01                        ;
;;      r01 = *IOArea+G723_ERR_FRM_COUNT:x;
;;      *HDT:x = r01                        ;

LBC_NEXT:
        jmp LBC_LOOP                      ;

LBC_END:
        jmp $                              ;

$EJECT
/**/
/*****/
/* Input PCM data for Encode process */
/*****/
Smp_Ix  imseg
GetPcmData:
        /*---*/      dp0 = EncPcmBuff      ;
        loop G723_FRAME {                  ;
                r01 = *HDT:x                ;
                /* */      *dp0++ = r01     ;
        }
        ret                                ;

/*****/
/* Output decoded PCM data */
/*****/
Smp_Ix  imseg
PutPcmData:
        /*---*/      dp0 = DecPcmBuff      ;
        loop G723_FRAME {                  ;
                /* */      r01 = *dp0++    ;
                *HDT: = r01                ;
        }
        ret                                ;

/*****/
/* Input Code data for Decode process */
/*****/
Smp_Ix  imseg
GetLineData:
        /*---*/      dp0 = DecLineBuff     ;
        loop 24/2 {                          ;
                r01 = *HDT:x                ;
                /* */      *dp0++ = r01     ;
        }
        ret                                ;

/*****/
/* Output Encoded Code data */
/*****/
Smp_Ix  imseg
PutLineData:

```

```

/*---*/    dp0 = EncLineBuff          ;
loop 24/2 {                                ;
    /* */    r01 = *dp0++              ;
    *HDT:x = r01                        ;
}
ret                                          ;
end

```



```

nop                ;
nop                ;
nop                ;
reti               ; INT1
nop                ;
nop                ;
nop                ;
reti               ; INT2
nop                ;
nop                ;
nop                ;
reti               ; INT3
nop                ;
nop                ;
nop                ;
reti               ; SI#1
nop                ;
nop                ;
nop                ;
reti               ; SO#1
nop                ;
nop                ;
nop                ;
reti               ; SI#2
nop                ;
nop                ;
nop                ;
reti               ; SO#2
nop                ;
nop                ;
nop                ;
reti               ; HOST IN
nop                ;
nop                ;
nop                ;
reti               ; HOST OUT
nop                ;
nop                ;
nop                ;
reti               ; not used
nop                ;
nop                ;
nop                ;
reti               ; not used
nop                ;
nop                ;
nop                ;

/*****
/* G723 OFF-Line Process                                     */
/*****

Smp_Ix  imseg
Init_Proc:
    clr(r0)                ; No Wait
    *DWTR:x = r01          ;
    *IWTR:x = r01          ;

    r01 = HST_WAIT         ; Wait mode
    *HST:x = r01          ;
    ret                    ;

```

```

$EJECT
/**/
/*-----*/
/* Scratch_Area / Static_Area / IO_Buffer */
/*      !!! 'lib_Scratch_x' must be equal to 'lib_Scratch_y' !!! */
/*-----*/
Scratch_X xramseg align at 0x020
lib_Scratch_x:
;      ds      G723_SCRATCH_X_BUFSIZE
      ds      G723C_SCRATCH_X_BUFSIZE          ; is bigger than G723_SCRATCH_X_BUFSIZE

Scratch_Y yramseg align at 0x020
lib_Scratch_y:
      ds      G723_SCRATCH_Y_BUFSIZE

Static_X xramseg align at 0x020
StaticAreaX: ds      G723_STATIC_X_BUFSIZE
C_StaticAreaX: ds      G723C_STATIC_X_BUFSIZE          ; Annex C

Static_Y yramseg align at 0x200
StaticAreaY: ds      G723_STATIC_Y_BUFSIZE
C_StaticAreaY: ds      G723C_STATIC_Y_BUFSIZE          ; Annex C

IO_Area_X xramseg
IOArea:
      ds      G723_IOTABLE_SIZE
LibEntry:
      ds      G723C_LIBENTRY_SIZE          ; Annex C

IO_Buff_X xramseg
EncPcmBuff:
      ds      G723_FRAME
DecPcmBuff:
      ds      G723_FRAME
EncLineBuff:
      ds      24/2 + 2
DecLineBuff:
      ds      24/2 + 1

EncChanBuff: ds      48          ; Annex C
DecChanBuff: ds      48          ; Annex C

/*-----*/
/* Variables */
/*-----*/
Smp_Xe1 xramseg
WrkMode:
      ds      1

/*-----*/
/* Annex C initialization */
/*-----*/
AnnexC_Const yramseg
BitrateConfig:
      dw      77, 27, 65, 23, 16, 6          /* Channel rate limits */

InitC imseg
InitAnnexC:
      clr(r0);
      /*...*/      *EncLineBuff+12:x = r0h;
      /*...*/      *EncLineBuff+13:x = r0h;

```

```

/*---*/    dp0 = LibEntry;
r0l = C_StaticAreaX;
/*...*/    *dp0++ = r0l;
r0l = C_StaticAreaY;
/*...*/    *dp0++ = r0l;
r0l = EncLineBuff;
/*...*/    *dp0++ = r0l;                // Input buffer
r0l = EncChanBuff;
/*...*/    *dp0++ = r0l;                // Output buffer
r0l = 1;
/*...*/    *dp0++ = r0l;                // EncodeSW (0:off 1:on)
r0l = 1;
/*...*/    *dp0++ = r0l;                // High/Low select (0:low 1:high)
/*---*/    dp4 = BitrateConfig;
/* */      *dp0++ = r0h;                // number of octets(dummy access)
/* */      r0 = *dp4++;
rep 6;
    /* */      *dp0++ = r0h    r0 = *dp4++;

/*---*/    dp0 = LibEntry;
call G723_Ini_ChCod;

clr(r0);
/*---*/    dp0 = LibEntry + G723C_LIBENTRY_DECOFS;
r0l = DecLineBuff;
/*...*/    *dp0++ = r0l;                // Output buffer
r0l = DecChanBuff;
/*...*/    *dp0++ = r0l;                // Input buffer
/* */      *dp0++ = r0h;                // clear error flag
/*---*/    dp4 = BitrateConfig;
/* */      *dp0++ = r0h;                // reserved
/* */      r0 = *dp4++;
rep 6;
    /* */      *dp0++ = r0h    r0 = *dp4++;

/*---*/    dp0 = LibEntry;
call G723_Ini_ChDec;

ret;

/*****
/* G723 OFF-Line Process */
*****/
Smp_Ix    imseg
G723_Proc:
    call G723_Start_Codec                ; Need only for Simulation test

    r0l = *HDT:x                          ; WrkMode 1)Cod ,2)Dec
    *WrkMode:x = r0l                      ;

;;    call G723_GetVersion                ; Library Version
;;    *HDT:x = r0h                        ; comment_out for
;;    *HDT:x = r0l                        ; Simulator Test

    r0l = StaticAreaX                      ;
    *IOArea+G723_STATIC_X_PTR:x = r0l    ; Static Area X
    r0l = StaticAreaY                      ;
    *IOArea+G723_STATIC_Y_PTR:x = r0l    ; Static Area Y

```



```

r01 = EncPcmBuff                ;
*IOArea+G723_ENC_BUFF_PTR:x = r01 ; PCM Buffer for Encode
r01 = EncLineBuff                ;
*IOArea+G723_VOVT_PTR:x = r01   ; Code Buffer for Encode
r01 = *HDT:x                      ;
*IOArea+G723_USE_HP:x = r01     ; UseHp 0)No , 1)UseHp
r01 = *HDT:x                      ;
*IOArea+G723_USE_VAD:x= r01     ; UseVAD 0)No , 1)UseHp

r01 = DecPcmBuff                ;
*IOArea+G723_DEC_BUFF_PTR:x = r01 ; PCM buffer for Decode
r01 = DecLineBuff                ;
*IOArea+G723_VINP_PTR:x = r01   ; Code buffer for Decode
r01 = *HDT:x                      ;
*IOArea+G723_USE_PF:x = r01     ; UsePf 0)No , 1)UsePf

;; Setup Lib-Entry for Annex C
call InitAnnexC;

r01 = *WrkMode:x                ;
r0 = r0 & 0x0001                ;
if( r0==0 ) jmp INI_LBC_DEC     ;
/*---*/ dp0 = IOArea            ;
call G723_Ini_Coder             ;
jmp LBC_LOOP                    ;

INI_LBC_DEC:
/*---*/ dp0 = IOArea            ;
call G723_Ini_Decod            ;
LBC_LOOP:
r0 = *HDT:x                      ; Control 0)Stop, 1)Continue
if( r0==0 ) jmp LBC_END        ;

r01 = *WrkMode:x                ;
r0 = r0 & 0x0001                ;
if( r0==0 ) jmp LBC_DEC        ;
call GetPcmData                 ;
r01 = *HDT:x                      ;
*IOArea+G723_WRK_RATE_E:x = r01 ; WrkRate 0)6.3, 1)5.3 [kbps]

/*---*/ dp0 = IOArea            ;
call G723_Encode                ;

/*---*/ dp0 = LibEntry          ; Annex C
call G723_ChCod                 ;

; call PutLineData                ; comment for Annex C
call PutChannelData;

;; r01 = *IOArea+G723_E_FRAME_TYPE:x ; comment_out for
;; *HDT:x = r01                      ; Simulator Test
jmp LBC_NEXT                    ;

LBC_DEC:
r01 = *WrkMode:x                ;
r0 = r0 & 0x0002                ;
if( r0==0 ) jmp LBC_NEXT       ;
; call GetLineData                 ;
call GetChannelData             ;

```

```

/*---*/      dp0 = LibEntry          ; Annex C
call G723_ChDec          ;

;
r01 = *HDT:x            ;
r01 = *(LibEntry + G723C_LIBENTRY_ERRINFO):x;
*IOArea+G723_CRC_RESULT:x = r01    ; CRC_result 0)Normal, else)Err

/*---*/      dp0 = IOArea          ;
call G723_Decode        ;

call PutPcmData        ;

;;
r01 = *IOArea+G723_D_FRAME_TYPE:x ; comment_out for
;;
*HDT:x = r01           ; Simulator Test
;;
r01 = *IOArea+G723_WRK_RATE_D:x   ;
;;
*HDT:x = r01           ;
;;
r01 = *IOArea+G723_ERR_FRM_COUNT:x;
;;
*HDT:x = r01           ;

LBC_NEXT:
    jmp LBC_LOOP        ;

LBC_END:
    nop                 ;
    nop                 ;
    nop                 ;
    nop                 ;
    nop                 ;
    nop                 ;
    nop                 ;
    halt                ;

$EJECT
/**/
/*****/
/* Input PCM data for Encode process */
/*****/
Smp_Ix  imseg
GetPcmData:
    /*---*/      p0 = EncPcmBuff          ;
    loop G723_FRAME {                    ;
        r01 = *HDT:x                      ;
        /* */      *dp0++ = r01          ;
    }
    ret                                           ;

/*****/
/* Output decoded PCM data */
/*****/
Smp_Ix  imseg
PutPcmData:
    /*---*/      dp0 = DecPcmBuff          ;
    loop G723_FRAME {                    ;
        /* */      r01 = *dp0++          ;
        *HDT:x = r01                      ;
    }
    ret                                           ;

/*****/
/* Input Code data for Decode process */
/*****/
Smp_Ix  imseg

```

```

GetLineData:
    /*---*/      dp0 = DecLineBuff      ;
    loop 24/2 {      ;
        r01 = *HDT:x      ;
        /* */      *dp0++ = r01      ;
    }
    ret      ;

/*****
/* Output Encoded Code data      */
/*****

Smp_Ix  imseg
PutLineData:
    /*---*/      dp0 = EncLineBuff      ;
    loop 24/2 {      ;
        /* */      r01 = *dp0++      ;
        *HDT:x = r01      ;
    }
    ret      ;

/*=====**
*      Annex C
*=====*/

Smp_Ix  imseg
PutChannelData:
    /*---*/      p0 = EncChanBuff;
    clr(r0);
    /*...*/      r01 = *(LibEntry + G723C_LIBENTRY_OUTSIZE):x;
    r0 = r0 + 1;
    r0 = r0 srl 1;
    loop r01 {
        /*...*/      r01 = *dp0++;
        /*...*/      *HDT:x = r01;
    }
    ret;

GetChannelData:
    /*---*/      dp0 = DecChanBuff;
    clr(r0);
    r01 = *HDT:x;      /* r01 = the number of words */
    loop r01 {
        /*...*/      r01 = *HDT:x;
        /*...*/      *dp0++ = r01;
    }
    ret;

end

```

— お問い合わせ先 —

【技術的なお問い合わせ先】

N E C 半導体テクニカルホットライン（インフォメーションセンター）
（電話：午前 9:00～12:00，午後 1:00～5:00）

電話：044-548-8899
FAX：044-548-7900
E-mail：s-info@saed.tmg.nec.co.jp

【営業関係お問い合わせ先】

半導体第一販売事業部								
半導体第二販売事業部	〒108-8001	東京都港区芝 5 - 7 - 1	（日本電気本社ビル）				(03)3454-1111	
半導体第三販売事業部								
中部支社 半導体第一販売部	〒460-8525	愛知県名古屋市中区錦 1 - 17 - 1	（日本電気中部ビル）				(052)222-2170	
中部支社 半導体第二販売部							(052)222-2190	
関西支社 半導体第一販売部	〒540-8551	大阪府大阪市中央区城見 1 - 4 - 24	（日本電気関西ビル）				(06)6945-3178	
関西支社 半導体第二販売部							(06)6945-3200	
関西支社 半導体第三販売部							(06)6945-3208	
北海道支社	札幌	(011)231-0163	甲府支店	甲府	(055)224-4141	京都支社	京都	(075)344-7824
東北支社	仙台	(022)267-8740	長野支店	松本	(0263)35-1662	神戸支社	神戸	(078)333-3854
岩手支店	盛岡	(019)651-4344	静岡支店	静岡	(054)254-4794	中国支社	広島	(082)242-5504
郡山支店	郡山	(024)923-5511	立川支店	立川	(042)526-5981,6167	鳥取支店	鳥取	(0857)27-5311
長岡支店	長岡	(0258)36-2155	埼玉支店	大宮	(048)649-1415	岡山支店	岡山	(086)225-4455
水戸支店	水戸	(029)226-1717	千葉支店	千葉	(043)238-8116	四国支社	岡山	(089)945-4149
群馬支店	高崎	(027)326-1255	神奈川支店	横浜	(045)682-4524	九州支社	福岡	(092)261-2806
太田支店	太田	(0276)46-4011	三重支店	津	(059)225-7341			
宇都宮支店	宇都宮	(028)621-2281	北陸支店	金沢	(076)232-7303			

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] μSAP77016-B04 ユーザーズ・マニュアル
(U13955JJ2V0UM00 (第2版))

[お名前など](さしつかえのない範囲で)

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, NEC 半導体ソリューション技術本部員,
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。

NEC 半導体テクニカルホットライン
FAX : (044) 548-7900