第三世代携帯電話に採用されているCODECを試してみる

音声圧縮伸張アルゴリズムAMRのTMS320C5000への実装

高橋 徹

携帯電話に採用されている音声圧縮伸張アルゴリズムをDSP搭載ボードに実装する例を紹介する．音声圧縮伸張アルゴリズムとしてAMRを，DSP搭載ボードとしてTMS320C5000シリーズ（TI社）のDSPが搭載されているTMS320VC5416DSKとTMS320C5510DSKを使用する．（筆者）

音声圧縮伸張規格の動向とDSPでの実現

● 音声圧縮伸張規格とその応用機器

まず，音声圧縮伸張規格の動向について簡単に紹介します（**図1**）．高サンプリング周波数（16kHz以上）の規格もありますが，ここでは電話帯域の音声信号（サンプリング周波数8kHz）用の規格に限定しています．

32kbpsなどの比較的高いビット・レートのアルゴリズムとしては，波形符号化処理であるADPCMがよく用いられており，日本のPHSでも32kbpsのADPCMが採用されています．

16kbps以下の低ビット・レートになると，ここ10年ぐらいではCELP（Code Excited Linear Prediction：符号励振型線形予測符号化）と呼ばれるアルゴリズムを基本としたものが多く，図に示している規格の大部分はCELP系のアルゴリズムです．応用例としては携帯電話が多く，日本では，第2世代方式（PDC）のフル・レートにVSELP，ハーフ・レートにPSI-CELPというアルゴリズム，またPDCのエンハンスト・フル・レート（NTTドコモではハイパートークという名称）にCS-ACELPというアルゴリズムが採用されています．

GSM系や米国のCDMA，TDMAにもCELP系のアルゴリズムが採用されています．ここで『G.』で始まっているアルゴリズムは，ITU-T（International Telecommunication Union - Telecommunication Standardization Sector）という国連の電気通信関連の標準化機関で標準化されたアルゴリズムです．

しかし最近は，CELP系のアルゴリズムも音質の向上ということに関してはかなり限界に近づいてきている感があります．最近開発されたアルゴリズムについては，さらなる高音質化というよりも多機能化に重点をおかれたアルゴリズムが多いと思われます．後ほど紹介するAMRも基本的には既存のアルゴリズムを組み合わせて多ビット・レートに対応した規格になっています．

● 音声圧縮アルゴリズムと使用されるDSP

これらのアルゴリズム（以下音声CODEC）がどのような形で各種製品に組み込まれているかですが，特定の音声CODECの専用LSIというのは少なく，汎用DSP＋ソフトウェアで実現する例が一般的です．

最近の携帯電話では，アプリケーション系の大規模LSI（DSPコア＋CPUコア＋周辺回路）のDSPコアで音声CODECソフトウェアを動作させる例も多くあります．DSPの種類として固定小数点DSPと浮動小数点DSPがありますが，多くの音声

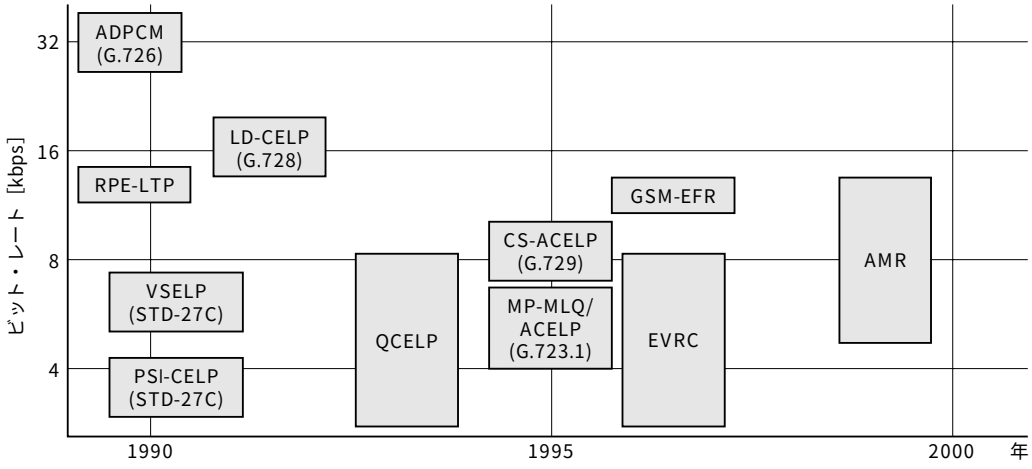

図1
音声圧縮伸張規格の動向

KEYWORD——音声圧縮伸張アルゴリズム，DSP，CELP，ITU-T，AMR，第三世代携帯電話，FOMA，TMS320UC5416DSK，TMS320C5510DSK，LCS，Code Composer Studio

CODECは開発元あるいは規格策定元から固定小数点のCソースが提供されていること，および携帯電話などの低消費電力を要求されるアプリケーションで使用されることが多いことなどから，固定小数点DSPが使用されているケースがかなり多いと思われます．

第3世代携帯電話の音声CODEC AMR

Adaptive Multi-Rate（以下AMR）は，3GPP（3rd Generation Partnership Project）という第3世代移動体通信システムの規格を制定する標準化機関で標準音声CODECとして採用された規格です．NTTドコモのFOMAにも標準音声CODECとして採用されています．

AMR-WB（Adaptive Multi-Rate Wideband：16kHzサンプリング）という高サンプリング周波数の規格と区別するために正式にはAdaptive Multi-Rate Narrowband（AMR-NB）という名称になっていますが，一般的にはAMR-NBではなく単にAMRと呼ばれることが多いようです．したがって，ここでも単にAMRと記述します．

AMRはマルチレート符号化機能，有音無音検出機能，背景雑音生成機能，フレーム誤り隠ぺい機能などから構成されています．マルチレート符号化機能では，4.75kbpsから12.2kbpsまでの8種類のレートの符号化モードをもっており，符号化モードはフレームごと（20ms）に変更可能です．したがって，基地局からの距離や干渉状態，トラフィック量などに応じてフレキシブルにビット・レートを変更でき，W-CDMA方式などの移動体通信システムに適しています．

AMRのリファレンスCソース・コードは3GPPのWebサイトで公開されています（http://www.3gpp.org/）．3GPPのAMR規格に準拠していることを謳うためには，このCソースの実行結果と完全一致（bit exact）させる必要があります．この評価のためのテスト・ベクタも同じサイトで入手可能です．

AMRのTMS320VC5416DSKボードへの実装

● DSPボードへの実装の手順

音声CODECをDSP搭載ボードや機器に実装するための手順はいろいろあると思います．ここでは実験的に**図2**のような手順で進めることを前提として，各項目について説明します．

● リファレンスCソース・コードの入手

上記のように，3GPPのWebサイトでAMRのリファレンスCソースを入手できます．ここでは，リリース5と呼ばれている規格のCソースを使用することにします．具体的には，

```
http://www.3gpp.org/ftp/Specs/latest/
                    Rel-5/26_series/26073-530.zip
```

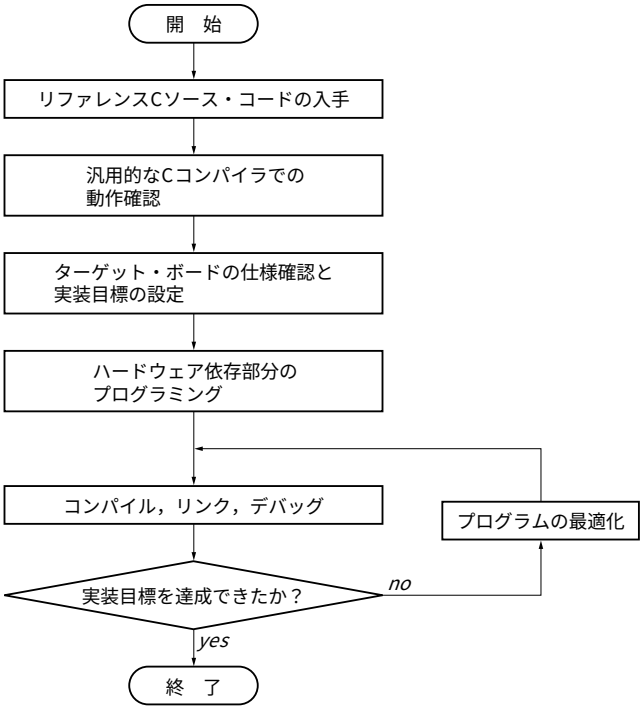

図2　実装手順の一例

をダウンロードします（2004年6月時点）．

これに対応するテスト・ベクタは，

```
http://www.3gpp.org/ftp/Specs/latest/
                    Rel-5/26_series/26074-500.zip
```

になります．

● 汎用的なCコンパイラでの動作確認

DSPへ実装して動作確認する前に，使い慣れた汎用的なC言語開発環境で動作確認します[注1]．

DSPへの実装時にオリジナルのCソースをそのまま使用した場合は不具合の発生率は少ないと思いますが，処理量の最適化のためにCソースを部分的に書き換えたり，一部の関数をアセンブリ言語で記述したりすると，不具合の発生率が増加します．そのとき，DSPソフトウェア開発用PC，あるいは別のPC上で汎用Cコンパイラのデバッガ環境を同時に立ち上げておけば，リファレンス・ソフトウェアとの比較が容易になり，デバッグの効率がよくなります．

AMRのCソースはマルチ・プラットホーム対応になっており，デフォルトで__MSDOS__，__sun__，linuxなどの定義が用意されているので，比較的簡単に実行ファイルを生成して動作確認できると思います．

● ターゲット・ボードの仕様確認と実装目標の設定

▶DSPスターター・キットTMS320VC5416DSK

TMS320VC5416DSK（Texas Instruments社）には，TMS320

注1：AMRをDSPに実装しようとするような技術者であれば，PCなどで動作するC言語の開発環境をすでにもっていると思うので，それを前提に記述する．