

リアルタイム音響処理を 成功に導く方法

著者: David Katz、インフォテインメント・システム・アーキテクチャ担当ディレクタ

多くの組み込みアプリケーションにおいて、リアルタイムの音響処理は非常に重要な要素となっています。具体的な例としては、音声の前処理、音声認識、ANC (Active Noise Cancellation) などが挙げられます。これらのアプリケーション領域では、リアルタイム性能に対する要求が着々と高まっています。そうしたニーズに適切に対応するためには、戦略的な考え方を採り入れなければなりません。大規模なSoC (System on Chip) が十分な性能を提供していることを考えると、追加で必要になった音響処理もSoCで対応したくなるかもしれません。しかし、その際には、遅延と時間確定性 (determinism) について慎重に検討する必要があります。これらは、リアルタイム・システムにおいて大きな問題につながりやすい重要な要素であることを理解しておかなければなりません。リアルタイム音響システムを開発する際には、SoCとオーディオ用デジタル・シグナル・プロセッサ (DSP) のうち、どちらで対応するのか慎重に検討を行う必要があります。本稿では、その選択で失敗しないために考察すべきいくつかの事柄について説明します。

遅延の小さい音響システムは、様々なアプリケーションで使用されています。車載分野を例にとると、パーソナル・オーディオ・ゾーン、RNC (Road Noise Cancellation)、車内通信システムなどで使われています。いずれのアプリケーションにおいても、遅延を抑えることが非常に重要です。

周知のとおり、自動車の分野には、車両の電動化という大きなトレンドが存在します。それに伴い、RNCの重要性はより増しています。電気自動車には、騒音を発する燃焼エンジンが存在しませ

ん。そのため、車両と道路の接触に伴って生じる音が非常に耳障りなものになっているのです。このような騒音を低減すれば、より快適な乗り心地が得られます。結果として、ドライバーの疲労を軽減することにもつながります。遅延の小さい音響システムをオーディオ用DSPではなくSoCに実装しようとする、数多くの課題に直面することになります。遅延、スケーラビリティ、アップグレード性、アルゴリズム、ハードウェア・アクセラレーション、カスタム・サポートなどが主な課題です。以下では、それぞれについて順に検討していきます。

遅延

リアルタイム音響処理を実現するシステムでは、遅延が重要な問題になります。プロセッサがリアルタイムのデータの移動や計算の要求に対応できない場合には、許容できないレベルで音声の途切れが生じる可能性があります。

通常、SoCに集積されるSRAMの容量は大きくありません。そのため、ローカル・メモリへのアクセスのほとんどは、キャッシュに依存することになります。その結果、コードやデータはデタミニスティックに使用されるわけではなく、処理に伴う遅延が増加することになります。ANCなどのリアルタイム・アプリケーションでは、それだけで必要な処理を実現できなくなる可能性があります。SoCは、マルチタスクの重い負荷を管理する非リアルタイムOSを実行します。このことから、システムの非デタミニスティックな動作が増加し、マルチタスク環境で比較的複雑な音響処理をサポートすることが困難になります。



図1は、SoCでリアルタイム・オーディオ処理を実行している場合の負荷の推移を表したものです。これを見ると、SoCのタスクの中でも優先順位の高い処理を実行している際には、CPUの負荷にスパイクが生じています。これらのスパイクは、メディア・レンダリング、ブラウジング、アプリケーションの実行など、SoCを中心としたアクティビティに伴って生じる可能性があります。CPUの負荷が100%を超えるレベルのスパイクが発生すると、SoCはリアルタイム動作を実現できなくなり、音声の途切れが生じます。

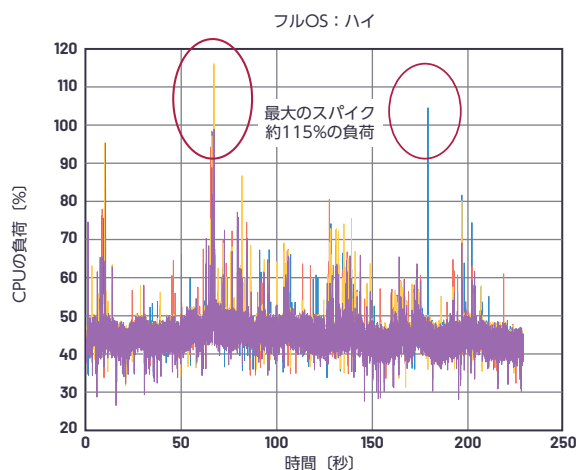


図1. CPUの瞬時負荷。代表的なSoCが、他のタスクに加えてオーディオ関連のメモリ処理を実行している場合の様子を表しています¹。

一方、オーディオ用DSPは、信号処理用のパスの全体にわたって遅延を小さく抑えられるように設計されています。信号処理用のパス全体とは、サンプリングされたオーディオ入力からコンポジット（オーディオ信号+アンチノイズ信号など）スピーカ出力までのすべてを指します。命令用のL1キャッシュとデータ用のSRAM（プロセッサ・コアに最も近いシングルサイクル・メモリ）は、中間データを外付けのメモリにオフロードすることなく、多くの処理アルゴリズムをサポートできるレベルの十分な容量を備えています。また、L1キャッシュ（SRAM）の容量を超過した場合には、オンチップのL2キャッシュ（コアから離れているものの、外付けのDRAMと比べればはるかに高速なアクセスが可能）により、中間データの処理用のバッファを提供することができます。加えて、オーディオ用DSPは、一般的にはリアルタイムOS（RTOS）を実行します。そのため、入力データは新たな入力データが到達するまでに処理されて、適切な宛先に送出されます。リアルタイム動作の最中にデータ・バッファがオーバーフローすることは決してありません。

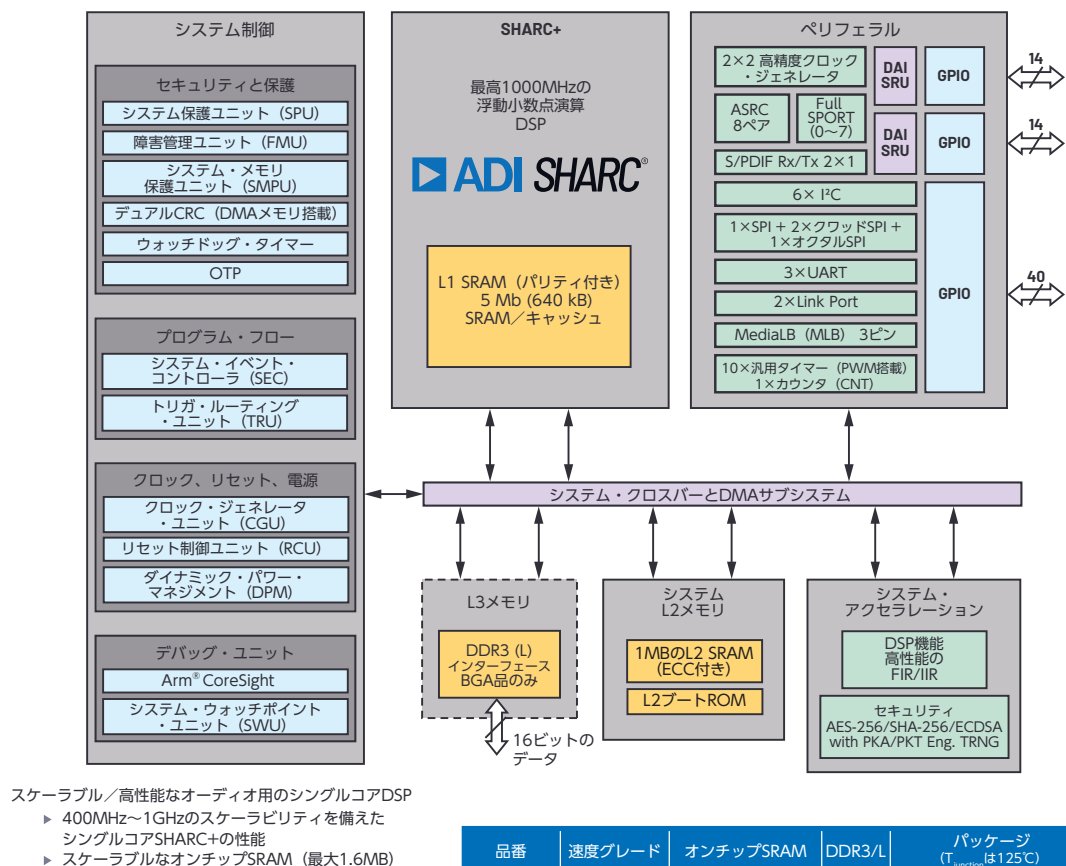
システムを起動する際の実際の遅延（多くの場合、オーディオが利用可能になるまでの時間を測定）も重要な指標です。特に車載システムでは、起動から一定の時間内に警告音を鳴らす必要があります。SoCの場合、通常はデバイス全体にわたりOSの起動を伴う非常に長い起動シーケンスが存在します。そのため、起動に関する要件を満足することが困難または不可能であるケースが少なくありません。一方、独自のRTOSを実行するスタンドアロンのオーディオ用DSPは、他のシステムの無関係な優先順位の影響を受けることなく、高速な起動に向けて最適化することができます。そのため、オーディオを利用できるようになるまでの時間については、容易に要件を満たすことが可能です。

スケーラビリティ

SoCを使用した場合の遅延は、ノイズ制御などのアプリケーションでも問題になります。ただ、音響処理に関するSoCの重要な欠点はもう1つあります。それはスケーラビリティです。SoCによって、多くの異なるサブシステムを伴う大規模なシステム（車載ヘッドエンド・ユニットやクラスタ）を制御するケースを考えます。その場合、SoCについては、ローエンドのオーディオからハイエンドのオーディオまでの異なるニーズに対し、スケーリングによって簡単に応じることはできません。なぜなら、各サブシステムのコンポーネントにおいて、スケーラビリティに対するニーズは常に競合し、SoC全体の利用率についてのトレードオフが必要になるからです。例えば、ヘッドエンドに適用されるSoCがリモートのチューナに接続されていたとします。そのチューナは、車種に応じて数チャンネルから多数のチャンネルまでスケーリングしなければならないとしましょう。その場合、チャンネルの構成ごとに先述したリアルタイム性に関する懸念が増大することになります。なぜなら、SoCの制御下に機能が追加されるたびに、SoCのリアルタイム動作と、複数の機能に使用されるアーキテクチャの主要なコンポーネントのリソースの利用可能性が変化するからです。そうしたリソースの例としては、メモリ帯域幅、プロセッサ・コアのサイクル、システムのバス・ファブリックにおけるアービトレーション・スロットなどが挙げられます。

上述したように、マルチタスクに対応するSoCに他のサブシステムを接続することについては懸念があります。それ以外に、音響サブシステム自体にも固有のスケーラビリティの問題があります。例えば、実際のアプリケーションでは、ローエンドからハイエンドまでのスケーリング（例えば、ANCのアプリケーションでマイクやスピーカのチャンネル数を増やす場合など）が行われるケースがあります。また、基本的なオーディオ信号のデコードから、ステレオ再生、3Dの仮想化といった高度な機能まで、オーディオに対するユーザ体験のスケーリングが行われることもあります。そうした要件は、ANCに対応するシステムにおいてリアルタイム処理に関する制約を共有することはありません。しかし、システムで使用するオーディオ用プロセッサの選択には直接影響を及ぼします。

SoCのコプロセッサとしてオーディオ用DSPを利用するのは、オーディオに関するスケーラビリティの問題を解決するための完璧な策です。モジュール式のシステム設計を行えるようになり、最適なコストでソリューションを実現可能になります。ほとんどの場合、SoCは、より大規模なシステムのリアルタイム音響処理に関するニーズに対して重点を置くことはできません。ただ、その処理を遅延の小さいオーディオ用DSPにオフロードする役割は果たせます。またオーディオ用DSPについては、コードの互換性とピンの互換性に関連する包括的なロードマップにおいて、様々な価格/性能/メモリの製品が提供されています。つまり、システム設計者が所定の製品のレベルに応じてオーディオ性能を最適化することができるだけの高い柔軟性を備えています。



品番	速度グレード	オンチップSRAM	DDR3/L	パッケージ (T _{junction} は125℃)
ADSP-21569W	1GHz	640kB L1	あり	400ボールのBGA 17mm×17mm (0.8mmピッチ)
ADSP-21567W	800MHz			
ADSP-21566W	600MHz			
ADSP-21565W	800MHz	640kB L1	なし	120ピンの LQFP-EP 14mm×14mm
ADSP-21563W	800MHz			
ADSP-21562W	400MHz			

図2. スケラビリティの高いオーディオ・プロセッサ「ADSP-2156x」

アップグレード性

今日の自動車では、無線によるファームウェアのアップデートが一般的になっています。アップグレード性は、重要なパッチを発行したり、新たな機能を提供したりする手段としてますます重要になっています。ただ、SoCでは、サブシステムの間の依存性が高まっていることから、アップグレードによって大きな問題が生じる可能性があります。そもそも、SoCでは、処理用の複数のスレッドとデータの移動用のスレッドがリソースについて競合することになります。新たな機能が追加された場合、特にアクティビティのピークが突発的に発生している間は、プロセッサの処理能力やメモリをより多く奪い合うこととなります。音響処理につい

て言えば、SoCに他の制御領域の機能が追加されると、リアルタイム性能に予期せぬ影響が及ぶ可能性があります。また、このような状況の副次的な影響として、すべての動作プレーンにまたがり、新たな機能のクロステストを実施しなければなりません。そのため、競合するサブシステムの様々な動作モードの間で無数のパーミュテーションが発生します。結果として、ソフトウェアの検証量は、アップグレードのパッケージごとに指数関数的に増加します。

見方を変えると、SoCにおけるオーディオ性能の向上は、SoCが制御する他のサブシステムの機能のロードマップに加えて、SoCの利用可能な処理能力に依存するとも言えます。

アルゴリズムの開発、性能

リアルタイム音響処理に使用するアルゴリズムの開発については、SoCよりもオーディオ用DSPの方が適していることは明らかです。なぜなら、オーディオ用DSPは、そのタスク向けに特化した製品だからです。スタンドアロンのオーディオ用DSPには、SoCとは大きく異なる点があります。それは、オーディオ用DSPではグラフィカルな開発環境を利用できることです。つまり、DSP向けのコーディング経験が浅い技術者でも、質の高い音響処理を設計に組み込むことができるのです。この種のツールを利用することにより、品質や性能を犠牲にすることなく開発時間を短縮することが可能になります。結果として、開発コストを削減できます。

ここでは、グラフィカルなオーディオ開発環境の一例として、アナログ・デバイセズのSigmaStudio[®]を紹介します。同ツールには、直感的に使用できるGUI (Graphical User Interface) が用意されています (図3)。また、そのGUIに統合された形で信号処理用の様々なアルゴリズムが提供されます。そのため、複雑なオーディオ信号のフローを容易に作成することができます。加えて、オーディオ伝送用のA²B (Automotive Audio Bus) の構成もグラフィカルに行えます。このように、SigmaStudioはリアルタイム音響処理向けのシステム開発に大いに役立ちます。

オーディオ処理に適したハードウェア

オーディオ用DSPのプロセッサ・コアは、効率的な並列浮動小数点演算とデータ・アクセス向けに特化して設計されています。そのアーキテクチャに加え、多くの場合、マルチチャンネルの専用アクセラレータを搭載しています。そのアクセラレータは、FFT (Fast Fourier Transform)、FIR/IIR (Finite/Infinite Impulse Response) フィルタ、ASRC (Asynchronous Sample Rate Converter) など、オーディオ分野で使用される一般的な処理に対応しています。これらを利用すれば、オーディオ用のフィルタ処理、サンプリング、周波数領域の変換をCPUコアの外部でリアルタイムに実行できます。結果として、コアの実効性能を高めることが可能になります。また、最適化されたアーキテクチャとデータフロー管理機能により、柔軟性が高くユーザフレンドリなプログラミング・モデルが提供されます。

オーディオの分野では、チャンネルの数、フィルタ・ストリーム、サンプル・レートなどが急増しています。そのため、プロセッサには、最大限に構成が可能なピン・インターフェースを備えていることが求められます。つまり、インラインのサンプル・レート変換、高精度のクロック供給、同期型の高速シリアル・ポートなどに対応できるようになっていなければなりません。



図3. SigmaStudioのGUI

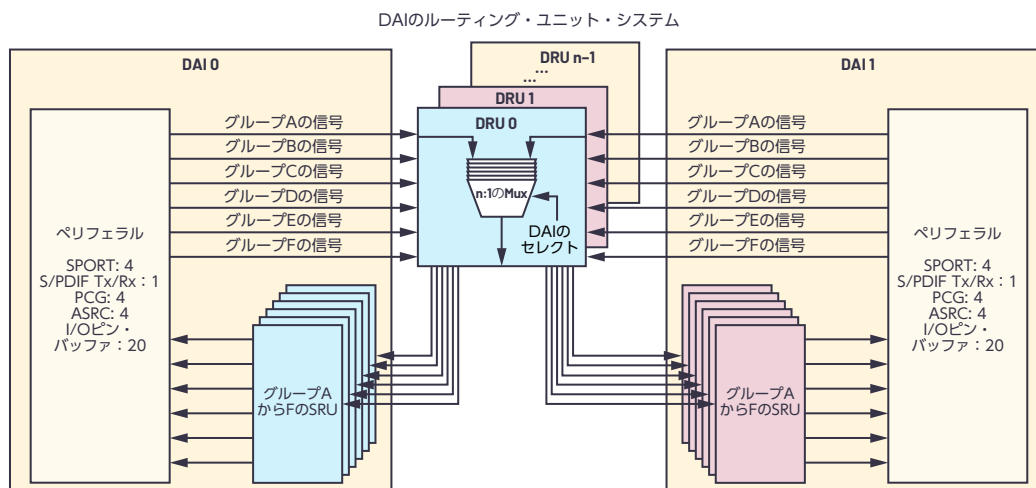


図4. DAIのブロック図

それにより、データを効率的にルーティングし、遅延の増加を防ぐことができます。また、外付けのインターフェース用ロジック回路が不要になります。アナログ・デバイセズのSHARC®ファミリの場合、図4に示すデジタル・オーディオ・インターフェース (DAI : Digital Audio Interface) を利用できます。

カスタマ・サポート

組み込みプロセッサを採用して開発を行う場合には見落としがちなことがあります。それはカスタマ・サポートです。

SoCのベンダーは、自社のDSP統合製品で音響アルゴリズムを実行することを推奨しています。しかし、それにはいくつかマイナスの側面があります。まず、SoCの通常のアプリケーション開発では、音響に関する専門知識は必要とされません。そのため、ベンダーからサポートを得ようとしても、一筋縄ではいかないことがあります。つまり、SoCが内蔵するDSPを使用して独自の音響アルゴリズムを開発しようとした場合、期待するほどのカスタマ・サポートは得られにくいということです。ベンダーは標準的なアルゴリズムしか提供していないこともあれば、音響アルゴリズムをSoCの1つまたは複数のコアに移植するだけで多額のNRE (Non-Recurring Engineering) を請求することもあります。ベンダーが完成度の高い低遅延のフレームワーク・ソフトウェアを提供していない場合、成功裏に開発を完了できる保証はありません。また、SoCをベースとして音響処理を開発する場合には、サードパーティのエコシステムの脆弱さに悩まされることになるでしょう。SoCの世界では、音響処理は中心的なテーマではあり

ません。むしろ、その場しのぎでサポートされている機能だと言う方が適切でしょう。

オーディオ用DSPを採用した場合、複雑な音響システムを開発するための非常に強力なエコシステムを活用できます。例えば、最適化されたアルゴリズムのライブラリや、デバイス・ドライバ、RTOS、使いやすい開発ツールなどを利用可能です。また、製品を市場に投入するまでの時間を短縮するためのオーディオ用リファレンス・プラットフォームなども提供されています。図5に示したのは、アナログ・デバイセズのSHARCオーディオ・モジュールです。このようなプラットフォームは、SoCの分野では珍しいかもしれませんが、スタンドアロンのオーディオ用DSPの分野では、ごく一般的に提供されています。

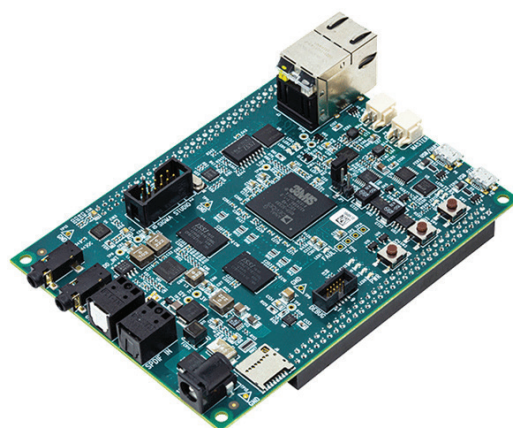


図5. SHARCオーディオ・モジュールの外観

リアルタイム音響処理を実現するシステムを設計する際には、リソースについて慎重に検討を行い、戦略的な計画を立てる必要があります。マルチタスクに対応するSoCの残った処理能力を割り当てただけでは不十分であることは明らかです。ぜひ、遅延の小さい処理を実現できるよう最適化されたスタンドアロンのオーディオ用DSPの採用をご検討ください。そうすれば、堅牢性を高め、開発時間を短縮できることが可能になります。また、将来のニーズと性能レベルに対応できる最適なスケーラビリティが得られるはずです。

参考資料

¹ Paul Beckmann 「[Multicore SOC Processors: Performance, Analysis, and Optimization](#) (マルチコアのSoCプロセッサ：性能、分析、最適化)」 2017 AES International Conference on Automotive Audio、2017年8月

著者について

David Katz (david.katz@analog.com) は、アナログ・デバイセズで車載向けインフォテインメント・システムのアーキテクチャを担当するディレクタです。30年にわたり、アナログ、デジタル、組み込みシステムの設計に携わってきました。世界に向けて、組み込み処理に関する記事を100本近く執筆。同分野については、いくつかの学会論文も発表しています。以前は、Motorolaのシニア設計エンジニアとしてケーブル・モデムやFAIに関する業務に携わっていました。コーネル大学で電気工学の学士号と修士号を取得しています。

EngineerZone[®]

オンライン・サポート・コミュニティ

アナログ・デバイセズのオンライン・サポート・コミュニティに参加すれば、各種の分野を専門とする技術者との連携を図ることができます。難易度の高い設計上の問題について問い合わせを行ったり、FAQを参照したり、ディスカッションに参加したりすることが可能です。



Visit ez.analog.com

*英語版技術記事は[こちら](#)よりご覧いただけます。