

第4章

まばたき検知機能と顔検出機能の設計

eXCiteの活用事例

白石忠明

ここではC言語で記述されたソフトウェア・プログラムからハードウェアを合成した事例を取り上げる。アプリケーションは、まばたき検知と顔検出である。ハードウェアを意識せずに記述されたコードからハードウェアを合成するに当たっては、ほとんどの場合、ソース・コードを修正する必要がある。本稿では、ソフトウェア用コードに対して実施した修正事項を具体的に示す。
(編集部)

アルゴリズムを回路動作にて確認しようとする際、HDL (Hardware Description Language) による設計手法では、アルゴリズムの検討から実証確認までの作業量が多くなります。高い専門性を持った技術者であっても、数カ月以上の期間を必要とすることがあります。特に、ソフトウェアとハードウェアの最適な機能分割による性能とコスト、開発効率の最適なバランスを検討するのは容易ではなく、経験と勘に頼らざるを得ない部分も多いと思います。これら背景から、製品の開発期間と品質を確実に守るために、プラットフォームを大きく変える事は難しく、実績のあるプラットフォームに改良を重ねる傾向となります。

筆者らは、「最適構成となるプラットフォームの早期確認」を目標として、高位合成技術の確立を進めています。高位合成技術は、ソフトウェアとハードウェアの技術を必要とするために、5年以上の回路設計経験を有するハードウェア技術者が、デバイス・ドライバやDSP (Digital Signal Processor) ファームウェアのソフトウェア設計を行うことによる技術者の育成から開始しました。

高位合成技術は、次の二つの方向性を持って発展してい

るといえます。

- ① ソフトウェアとハードウェアの協調設計：ソフトウェアとハードウェアの機能分担の最適化を図るため、設計言語を共通化
- ② 回路設計の効率化：ソフトウェアかハードウェアかをあまり意識しないですむため、設計効率が向上
筆者らは、ソフトウェアとハードウェアの協調設計を主眼に置き、ANSI-Cからの高位合成を進めており、「まばたき検知機能」と「顔検出機能」に適用しました。



1. 高位合成ツールの適用

● 高位合成ツールとしてeXCiteを採用

高位合成ツールには、米国Y Explorations社の「eXCite Professional」を用いました。eXCite Professionalへ入力可能なCソース・コードは、ANSI-Cのサブセットです。回路化できない構文には対応していません。

ANSI-Cの構文に対して、メモリの初期化に使用するmemset()や平方根を導出する関数sqrt()などのC言語の標準関数は含まれません^{注1}。また、浮動小数点型については、float型のみです。double型はサポートしていません^{注2}。今回は、符号、指数、仮数部をすべて整数型の変数を使用して計算するユーザ関数を新規作成しました。float型はサポート対象ですが、eXCiteが用意する浮動小数点演算用ライブラリの浮動小数点用モジュールは、回路規模が大きく

注1：標準関数のサポート範囲は徐々に広がっている。

注2：double型への対応は計画されている。

Keyword

高位合成, eXCite, まばたき検知, 顔検出, 最適化, バイプライン, 記述スタイル

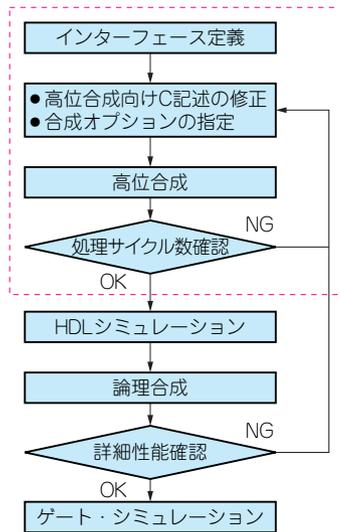


図1 Cソース・コードからハードウェア回路を生成するフロー

なるためです。

図1に、Cソース・コードからハードウェア回路を生成するフローを示します。高位合成により、処理サイクル数の情報を得て、所望のサイクル数に収まるかの確認を行います。また参考値として、動作周波数や回路規模が得られます。詳細値は論理合成の結果により判定し、問題があればCソース・コードや合成オプションの修正を行います。

● 高位合成を段階的に適用

高位合成化に向けて、ハードウェアをまったく意識せずに記述されたCソース・コードを回路化し、ソフトウェアとの動作比較を行うことを目標としました。

対象とした「まばたき検知機能」や「顔検出機能」は、元々は、動画像の中から人の顔を検出する際に、目を開いた顔画像の抽出を目的としたC言語によるソフトウェア・プログラムでした。まず、まばたきを検知する機能だけを抜き出し、比較的小さな規模の高位合成を成功させ、その次に、規模の大きい顔検出機能の高位合成を行うステップを経ました。こうしてパソコンのミドルウェアのレベルから、DSPへの移植を経て、高位合成による回路化を図りました。

(1) まばたき検知機能

まばたき検知では、あらかじめ顔検出を行うと目の位置が絞り込めるため、目の検出は比較的容易です。しかし、今回のまばたき検知では、顔検出の処理を含んでいません。

そこで目の位置の絞り込みの代わりに、検知対象者に目の位置を合わせてもらう仕様になりました(図2)。これにより、まばたき検知の本来の機能は、何も変更しなくてもよ

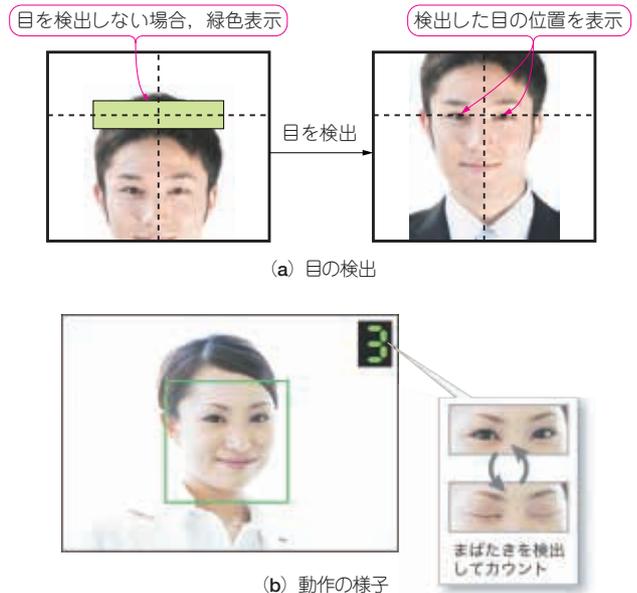


図2 まばたき検知機能

顔検出機能は持たない小規模なものである。検知対象者に目の位置を合わせてもらう。

くなります。

(2) 顔検出機能

顔検出処理では、顔の特徴を持つ検出フィルタを用いて、入力画像上を顔検出サイズを変更させながらスキャンを行います。処理時間は、入力画像サイズ、顔検出最小サイズ、サーチ・ステップ画素数、顔検出サイズのスケール変更量、顔の検出率、誤検出率により異なります。

顔検出処理は、画像の輝度情報に対し顔検出器を走査させ、検出の有無を判定します。一つの顔検出器は、顔の特徴を持つフィルタ・テーブルを用いて複数の矩形フィルタを多段に接続し、結果がしきい値以上であった場合に結果を有効とします。図3にその構成を示します。

顔検出の判定は、複数の顔検出器を多段直列に接続し、すべての顔検出器の結果が有効であった場合に、最終の顔検出結果として「検出あり」とします。検出途中で顔検出器が無効であった場合は、以降の顔検出処理を打ち切り、次の顔検出範囲に検出位置を移動し、検出処理を顔検出器1から再度実施します。

2. 高位合成のための最適化

高位合成の実施にあたっては、Cソース・コードの分析を行い、回路化できない構文を置き換える必要があります。次に処理やデータの流れを整理し、回路化に向けてイン