

Javaが書ければFPGAが開発できる!

PC上でもすぐに動き、並列処理も記述可能なプログラミング言語 Java でハードウェアを開発

Synthesijerを使ったFPGA上でのグラフィックス・プログラミング

三好 健文 Takefumi Miyoshi

SynthesijerはJavaをベースに筆者が開発を進めている高位合成処理系です。オープン・ソースで公開しており、無償で使えます。ここでは開発事例として、簡単なグラフィックス・プログラミング(FPGAハードウェア開発)を紹介します。ソフトウェアとしてPCで動作を確認した上でハードウェア化できる特徴を活用すると、非常に楽に開発を進めることができます。

1 グラフィックス・システムの概要

- **VGA解像度で8色表示**
今回作るシステムの概略を図1に示します。画像描画のバックエンド(BitMapDisplay)は640ドット×480ラインのフレーム・バッファ0と1を持ち、バッファにデータを書き込むと画像表示できるグラフィックス表示システムです。フレーム・バッファ0を1に優先(上書き)して表示することとします。実装の説明を簡単にするために、色数は8色とします。また画像のスクロールができるように、フレーム・バッファ・メモリのx, y座標にオフセットを与えることで、表示位置を指定できるようにしましょう。
- **グラフィックス表示プログラムの例**
このシステム上でのプログラムの例をリスト1に示

します。この例では、フレーム・バッファ0の座標(100, 100)に20ピクセルの正方形を表示し、また、フレーム・バッファ1上には、色を変えながら40ピクセルの正方形を敷き詰めます。
さらに、フレーム・バッファ1の表示オフセットを定期的に移動させることで、表示画像をスクロールさせます。

- **BitMapCanvasの実装**
ユーザには描画APIを提供します(BitMapCanvas)。点を打つメソッド(pset)に加えて、線を描画するメソッド(line)と、四角を描画するメソッド(rect)を用意することにします。
BitMapCanvasのコードをリスト2に示します。lineメソッドの実装は、Bresenham's Algorithm⁽¹⁾の例を、ほぼそのままJavaに記述し直したものです。

2 ソフトウェアとしての実行

- **Javaグラフィックス・ライブラリSwingを使う**
実装したAPIが想定通り動作するか、ソフトウェアとして実行することで確認してみましょう。ソフトウェアとして動作させるために、描画バックエンドであるBitMapDisplayクラスおよび描画エンジンに相当する部分を、Javaのグラフィックス・ライブラリであるSwingを使って実装しました(リスト3)。このバックエンドでは、640×480のウィンドウ(JFrame)を描画し、そのウィンドウに座標を指定して点を描画できるようにしています。

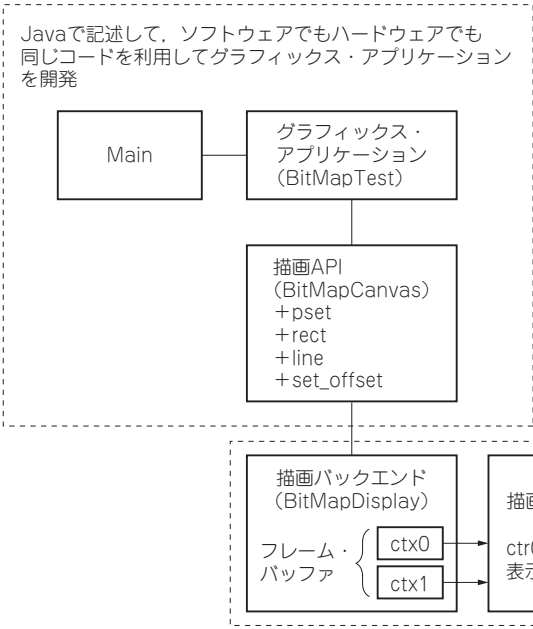


図1 実装するシステムのブロック図