

高位合成

4I24 中川寛之

目的

本実験の主たる目的は、高位合成（HLS: High-Level Synthesis）を用いたFPGAの設計フローを習得する。

高位合成とは

C/C++やSystemC、MATLABなどの抽象度の高いプログラミング言語で記述された動作仕様から、ASICやFPGAなどのハードウェア記述言語（HDL：Verilog/VHDL）を自動生成する技術

1. 高位合成の理解と実践:

- C++言語を用いて「10ビットの入力値に含まれる1（High）のビット数をカウントする回路（ビットカウンタ）」を記述し、**Vitis HLS**ツールを用いてRTL（Register Transfer Level）回路へ合成します。
- 合成レポート（遅延、リソース使用量）やスケジュールビューアを通じて、記述したコードがどのようにハードウェア化されるか（依存関係や並列性）を確認します。

2. 検証とIP化:

- CシミュレーションおよびC/RTL協調シミュレーション（Co-simulation）を行い、論理動作と生成されたRTLの動作が正しいことを波形レベルで検証します。
- 合成結果を出力ポートのハンドシェイク信号（**ap_vld**等）を含めた形で最適化し、**Vivado**で利用可能なIPコアとしてエクスポートします。

3. FPGAへの実装と課題解決:

- 作成したIPコアをVivado上で読み込み、評価ボード（Basys3）のスイッチやLEDに割り当てて実機動作を確認します。
- 発展課題として、ボタンスイッチ（BTNL）を押したタイミングで演算を開始するようにトップモジュール（**top.v**）の制御信号（**ap_start**等）を適切に設計・修正します。

使用ツール

• 高位合成ツール:

- **AMD Vitis HLS 2023.2**
- 用途：C++ソースコードの記述、Cシミュレーション、RTLへの合成、協調シミュレーション、IPコアの生成。

• 論理合成・配置配線ツール:

- **AMD Vivado 2023.2 ML Edition**
- 用途：IPコアの取り込み、RTLプロジェクトの作成、トップモジュール（Verilog HDL）の記述、ビットストリームの生成、実機への書き込み。

• 使用言語:

- **C++言語**: 高位合成用ソース (`bitcount.cpp`) およびテストベンチ (`test_bitcount.cpp`) の記述に使用。
- **Verilog HDL**: IPコアをFPGAのピンに接続するためのトップモジュール (`top.v`) の記述に使用。
- **ハードウェア (評価ボード)** :
 - **Digilent Basys3**,
 - 搭載FPGAデバイス : Artix-7 (`xc7a35tcpg236-1`)
 - 使用インターフェース : スライドスイッチ (入力 `SW0-9`)、LED (出力 `LD0-3`)、プッシュボタン (制御用 `BTNC`, `BTNL`) ,。

課題

1. 値をセットしておき、スイッチ BTNL を押したら、ビットをカウントせよ。

1. スイッチ START_BTN (BTNL) を追加
2. `ap_start`, `ap_done`, `ap_idle`を用意する
3. 検討事項

1. `ap_start`, `ap_done`, `ap_idle` の初期状態は ?

```
// top.v
reg ap_start = 1'b0;
```

2. `ap_start`をHにするのはどのようなときか ?

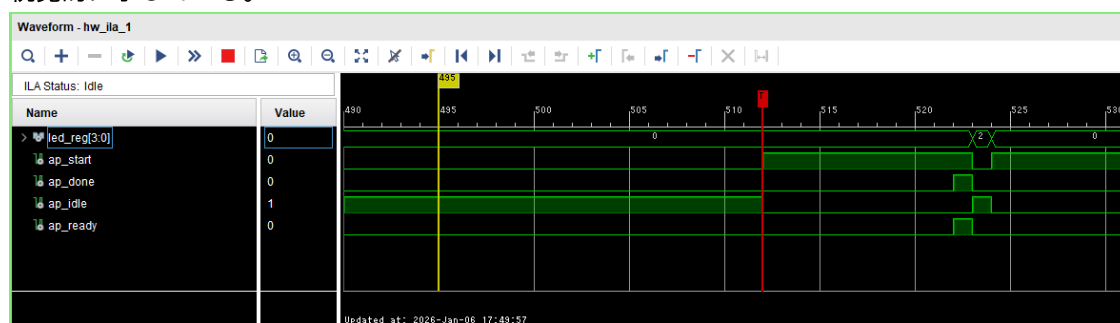
```
// top.v
if ((STARTBTN == 1'b1) && (startbtn_prev == 1'b0)) begin
    if (ap_idle == 1'b1) begin // アイドル状態の時のみ
        ap_start <= 1'b1;      // 計算開始
        led_reg <= 4'd0; // ★再計算開始の合図としてLEDを一旦消す
    (任意)
    end
end
```

3. `ap_start`をLにするのは (結果が求まったときは) どのようなときか ?

```
// top.v
if (ap_done == 1'b1) begin
    ap_start <= 1'b0;          // Startを下げる
    led_reg <= hbits_out_wire; // ★IPの結果をレジスタに焼き付ける
    (保持)
end
```

4. スイッチの影響（押）によって実行されたことを示すには？

計算開始時にLEDを一旦消灯し、計算完了時に結果を表示することで、ボタン押下による実行を視覚的に示している。



2. FPGAを学習した感想を書きなさい。

今回の実験を通じて、FPGAと高位合成技術の理解を深めることができました。特に印象的だったのは、C++で記述したアルゴリズムがVitis HLSによって自動的にハードウェア回路に変換される過程です。従来のVerilog HDLによる直接的な回路設計と比較して、高位合成は開発効率を大幅に向上させる可能性を持つことを実感しました。

一方で、ハンドシェイク信号（ap_start、ap_done、ap_idle）の制御や、ボタン入力のエッジ検出など、実際のハードウェア制御では細かな信号タイミングの理解が不可欠であることも学びました。CシミュレーションとC/RTL協調シミュレーションを通じて、ソフトウェア的思考とハードウェア的思考の橋渡しの重要性を認識できました。

FPGAは並列処理能力に優れ、リアルタイム処理が求められる用途に適していることを実機動作で確認でき、今後の応用可能性について期待が膨らみました。