HDL によるハードウェア設計

[目的]

ハードウェア記述言語(HDL: Hardware Description Language)を使用したハードウェア(ディジタル論理回路)の設計手法を学習する。また、学習用 FPGA ボードと HDL シミュレータによる回路の動作検証を通して、実際にハードウェアがどのように動作するのか理解する。

[概要]

HDL の一つとして広く利用されている Verilog-HDL を使用して、ディジタル論理回路、 すなわち組合せ回路や順序回路を記述する。記述した回路が正しく動作することを、学習用 FPGA ボードおよび Verilog-HDL シミュレータを用いて確認する。

HDLによるディジタル論理回路の設計は、一般的には図 1に示す設計フロー1で行われる。

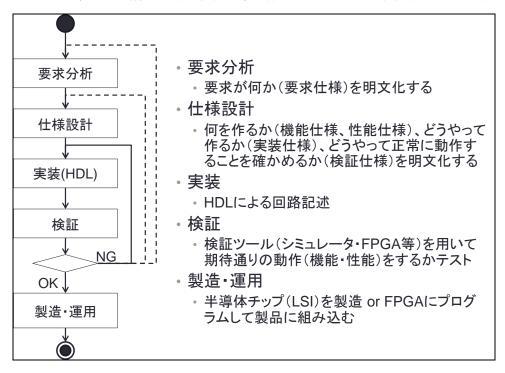


図 1 HDLによるディジタル論理回路設計フローの例(単純化したもの)

本実験では、仕様に基づいて HDL による記述を行い(実装)、期待通りの動作をすることをシミュレータ・FPGA²を用いてテストする(検証)という工程を、受講生が各自行う。一連の設計フローを経験し、それをレポートという形に明文化することを通じて、HDL によるハードウェア設計の実際を理解する。

1 この設計フローは、単純化して図示したものであり、実際の設計においては、実装と検証は様々なレベルで繰り返し行われる。また点線のフローは生じないことが望ましい。

² FPGA(Field Programmable Gate Array)は、開発者が任意のディジタル論理回路を、ソフトウェアの様にプログラムして実現することが可能な、半導体 LSI チップである。

「使用する教材」

- ・実験指導書(この資料)
- ・テキスト冊子: キットで学ぶ FPGA チャレンジャー Xilinx Artix-7 版 (WebClass 参照) *学習のためにいくつかのページを意図的に削除したり、編集を加えたりしている
- ·講義資料 (WebClass 参照)
- ・学習用 FPGA ボード Basys3 ボード
 - ・FPGA: Xilinx 社製 Artix-7 XC7A35T
 - ・ユーザ I/O デバイス :

LED 緑 16 個、スライドスイッチ 16 個、プッシュボタン 5 個、リセットボタン 7 セグメント LED 4 ケタ、VGA ポート、I/O ピン 6pin(Pmod) x 4 個、他

[実験内容]

以下の3項目について実験を行い、HDLで記述し合成した回路が期待通りに動作することを、FPGAボードとシミュレータを用いて確認する。

- 1. 組合せ回路を FPGA ボードで動かす
- 2. 順序回路を FPGA ボードで動かす・シミュレーションで何が起こっているか確認する
- 3. 少し複雑な回路を実装して FPGA ボードで動かす・シミュレーションで確認する

1. 組合せ回路を FPGA ボードで動かす

組合せ回路は、AND/OR/NOTから構成される論理関数に相当するディジタル論理回路である。すなわち、<u>出力はその時の入力によってのみ決まる</u>。入力の組合せで出力が決まるので、組合せ回路という。

本項目においてはFPGA上に組合せ回路を実現して期待通りに動作することを確認する。 *手順*

- 1-1. STEP01 を読み、学習用 FPGA ボードと FPGA について理解する。
- 1-2. STEP02 を読み、P.22 の Basys3 ボード動作チェックを行う。
- 1-3. 講義資料のスライド 14~32 を読み、Vivado のプロジェクトを作成する。最も単純な Verilog 記述を作成する。(P.32 ON_Circuit.v)。テキスト冊子の STEP03 は, 実験で使用 する Vivado よりバージョンが古く, 実際の手順とは異なる箇所があることに注意する。
- 1-4. STEP04 に従い、合成・デバイスの制約・ビットストリーム生成 (コンパイル)・デバイスへ書き込み(JTAG プログラミング)、を行うことで、FPGA ボードに作成した Verilog 記述に相当する回路をプログラムし動作を確認する。
- 1-5. STEP05 に従い、課題 05 (AND 回路) を作成し動作を確認する。
- 1-6. STEP06 に従い、課題 06 (OR 回路) を作成し動作を確認する。
- 1-7. レポート課題1「組合せ回路の実装・テスト」を行う。
 - → 仕様 (何を作ったか)、実装 (どうやって作ったか)、およびテスト結果について レポートに記載すること。
- 1-8. (オプション) STEP07 に従い、セレクタ回路を作成し動作を確認する。

2. 順序回路を FPGA ボードで動かす・シミュレーションで何が起こっているか確認する

順序回路は、メモリ要素 (フリップフロップなど) を含むディジタル論理回路である。すなわち、<u>出力はその時の入力と、以前の状態から決まる</u>。入力の順序により出力が決まるので、順序回路という。本項目においては、FPGA 上に順序回路を実現して、期待通りに動作することを確認する。また、回路の動作は非常に高速(100MH z で動作)であるため、シミュレーションにより動作の詳細な様子を観察して理解する。

手順

- 2-1. STEP08 に従い、**課題 08-1 (非同期リセット)** および**課題 08-2 (同期リセット)** を実施する。フリップフロップ(DFF)の動作を FPGA ボードにおいて確認する。
- 2-2. STEP09 に従い、**課題 08-1** と**課題 08-2** のフリップフロップ(DFF)の動作をシミュレーション(Vivado のシミュレータ)で確認する。
 - → 非同期リセットと同期リセットの違いを理解する。(班内でお互いに説明せよ)
- 2-3. STEP10 に従い、**課題 10-1 (カウンタ)、課題 10-2 (100MHz** クロック分周して **1Hz を作成)** および**課題 10-3 (1 秒ごとにカウントアップして LED 表示)** を実施する。 各回路の動作を確認する (FPGA ボード、および Vivado のシミュレータ)。
- 2-4. レポート課題2「順序回路の実装・テスト」を行う。
 - → 仕様(何を作ったか)、実装(どうやって作ったか)、およびテスト結果についてレポートに記載すること。
- 2-5. (オプション) STEP11 に従い、スイッチ入力を数えるカウンタ回路の動作を確認する。

3. 少し複雑な回路の実装・課題回路を作成して動作を検証する

本項目においては、これまで学習した Verilog-HDL での回路設計方法をもとに、前回学習したカウンタを応用した少し複雑な回路としてステートマシンを作成する。

手順

- 3-1. WebClass および講義資料ページに記載されている「第3回講義資料」「補足資料:ステートマシン設計の基本」 を参照し、簡単なステートマシンを実装する。FPGA ボードで動作を確認する。
- 3-2. レポート課題 **3**「ステートマシンの実装・テスト」を行う。
 - → 仕様(何を作ったか)、実装(どうやって作ったか)、およびテスト結果についてレポートに記載すること。
- 3-3. (オプション) STEP12 を読み、**課題 12-1**: 7 セグ LED デコーダ回路の動作を理解する。 (P.166 課題 12-2 の、7 セグ LED カウンタ回路はオプションとする)
- 3-4. (オプション) 任意の 4 ケタ数字を 7 セグメント LED に表示する回路を、講義資料 WEB サイトよりダウンロードし、FPGA ボードにおいて動作を確認する。
- 3-5. (オプション) レポート課題 1,2 が終わってない場合は取り組む。レポート課題 4 に取り組む。レポート作成を進める。

「課題]

ソースコードなどの参考資料は、講義資料 WEB サイト内の以下の URI を参照すること。 http://www.ced.is.utsunomiya-u.ac.jp/lecture/2022/jikkenb/hdl/examples/hdl.html

レポート課題1:組み合わせ回路の実装・テスト

積和演算回路を Verilog-HDL で実装・テストする。

◆ 機能仕様

入力: FPGA ボード上のスライドスイッチ(SW)の、SW15~8 を入力 A[7:0]、SW7~0 を入力 B[7:0]とする。

出力: A と B を乗算し、<u>本年度の数値(10 進数の 2022)</u>を加算した値を S (Sum の略)とし、FPGA ボード上の LED (LD0~LD15) に S[15:0]の値を出力する。

◆ 実装仕様

・加算器・乗算器は Verilog-HDL の加算演算子・乗算演算子を使って記述する。

◆ 検証仕様

(1) FPGA ボードを用いて検証を行う。以下の入力値の組合せに対する出力値を記録し、 期待値と一致するかを確認する。結果は OK か NG かで示す。

| テスト | 入力 | | | 出力期待値 | 出力値 | 結果 |
|-----|--------|--------|-------|----------------------|---------|-------|
| 番号 | A[7:0] | B[7:0] | 10 進数 | S[15:0] | S[15:0] | OK/NG |
| 1 | 0 | 0 | 2022 | 16'b0000011111100110 | | |
| 2 | 1 | 0 | 2022 | 16'b0000011111100110 | | |
| 3 | 0 | 1 | 2022 | 16'b0000011111100110 | | |
| 4 | 1 | 1 | 2023 | | | |
| 5 | 2 | 3 | 2028 | | | |
| 6 | 100 | 100 | | | | |
| 7 | 250 | 250 | | | | |
| 8 | 255 | 255 | | | | |

(2) 自分の学籍番号をテスト入力値に設定し、出力値 (LED 状態) を、カメラで撮影してレポートに画像として貼る。撮影の際、加算器への入力値は、自分の学籍番号の下二桁の数字を入力 A、入力 B を 100 とし、LED の出力が期待値と同じになることを示す。例)学籍番号: 202199 入力 A=99, 入力 B=100 出力期待値 S=99*100+2022=11922=16'b0010111010010010 (10 進数の <math>11922 を 16bit の 2 進数で表した場合)

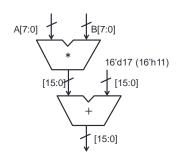


図 2 積和演算器のブロック図

レポート課題 2: 順序回路の実装・テスト

課題10-3で作成した1秒ごとにカウントアップしてLED表示する4ビットカウンタに、カウントダウンを行うdown信号を追加した回路をVerilog-HDLにて実装・テストする。

◆ 機能仕様

入力: FPGA ボード上のスライドスイッチ (SW) の SW0 を入力 down とする。すなわち, SW0 が ON になり, down 信号が 1 となっている間, カウントダウンが実行される。カウンタの最大値は 15 に変更し, 15 までカウントアップした後に出力が 0 になるようにすること。プッシュボタン BTNL (reset) を押すと初期状態 (すべての入力, 出力信号が 0 の状態) に戻ることとする。

出力: 1 秒ごとにカウントアップ/カウントダウンした数値を 4 ビットの 2 進数として FPGA ボード上の 4 つの LED に表示する。1 秒の時間は, FPGA ボードの 100MHz クロック信号を 100M 回カウントすることで計測する。

◆ 実装仕様

・以下の状態遷移図および状態遷移表に従った、4 ビットカウンタを作成する。

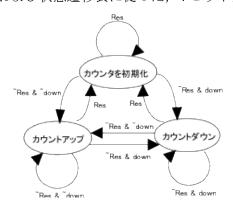


図3 アップダウンカウンタの状態遷移

| | | | 1 10000 = 12.000 | | |
|---------|---------|--------|------------------|---------|--|
| クロック | 入力 | | 出力 | | |
| 時刻 | BTNC | SW0 | LED[3:0] | 備考 | |
| H/1 ×/1 | (reset) | (down) | | | |
| 1 | 1 | X | X | - | |
| 2 | 0 | 0 | 0 | - | |
| 3 | 0 | 0 | 1 | カウントアップ | |
| 4 | 0 | 0 | 2 | カウントアップ | |
| 5 | 0 | 0 | 3 | カウントアップ | |
| 6 | 0 | 0 | 4 | カウントアップ | |
| 7 | 0 | 1 | 3 | カウントダウン | |
| 8 | 0 | 1 | 2 | カウントダウン | |
| 9 | 0 | 1 | 1 | カウントダウン | |
| 10 | 0 | 0 | 2 | カウントアップ | |
| 11 | 0 | 0 | 3 | カウントアップ | |
| 12 | 0 | 0 | 4 | カウントアップ | |

表 1 レポート課題2の状態遷移表

表中のXは、任意の値を示す(Don't care という)。

◆ 検証仕様

- ・シミュレータを用いて検証を行う。実時間(数秒間=数千万クロック)のシミュレーション結果を確認するのは時間がかかるので、シミュレータを用いた検証の際は、2クロックに1桁ずつLEDへの出力信号を変化させることとする。
- ・入力値の組合せに対する出力値を記録し、期待値と一致するかを確認する。結果は OK か NG かで示す。以下に検証に用いる表の例を示す。この表は例であるため、適宜 変更して構わない。ただし、表に記載する入出力信号の値と、レポートに貼り付けるシミュレーション結果が同じになるようにすること。

| クロック | 入力 | ħ | 出力期待値 | 出力値 | 結果 |
|------|---------|--------|----------|----------|-------|
| 時刻 | BTNC | SW0 | LED[3:0] | LED[3:0] | OK/NG |
| | (reset) | (down) | | | |
| 0 | 1 | 0 | 0 | | |
| 1 | 0 | | 1 | | |
| 2 | | | 1 | | |
| 3 | | | 2 | | |
| 4 | | | 2 | | |
| 5 | | | 3 | | |
| 6 | | | 3 | | |
| 7 | | | 4 | | |
| 8 | | | 4 | | |
| 9 | | 1 | 3 | | |
| 10 | | | 3 | | |
| 11 | | | 2 | | |
| 12 | | | 2 | | |
| 13 | | | 1 | | |
| 14 | | | 1 | | |
| 15 | 1 | 0 | 0 | | |
| 16 | | | 0 | | |

表 2 レポート課題 2 の動作検証表の例

・信号の値は、シミュレーション波形の画面キャプチャを行い、レポートに貼り付ける こと。シミュレーション波形は以下の様子を含むものとする。複数個に分けてシミュ レーション波形をレポートに記載しても構わない。

- 1. 2クロック毎のカウントアップ
- 2. RESET 信号の入力による初期化
- 3. down 信号の入力による 2 クロック毎のカウントダウン
- 4. down 信号が 1 から 0 になった場合の 2 クロック毎のカウントアップ

レポート課題3: ステートマシンの実装・テスト

自分の学籍番号の数字を、順番に 1 秒間に 1 桁ずつ LED に表示する順序回路を Verilog-HDL にて実装・テストする。

◆ 機能仕様

入力: プッシュボタン BTND (start) を押すことで学籍番号の表示を開始する合図と する。プッシュボタン BTNC (reset) を押すと初期状態に戻ることとする。

出力: **FPGA** ボード上の 4 つの **LED** に自分の学籍番号を順番に 1 秒間に 1 桁ずつ、4 ビットの 2 進数として表示する。1 秒間の時間は、**FPGA** ボードの 100MHz クロック信号を 100M 回カウントすることで計測する。

◆ 実装仕様

・以下の状態遷移図および状態遷移表に従った、ステートマシン3を作成する。

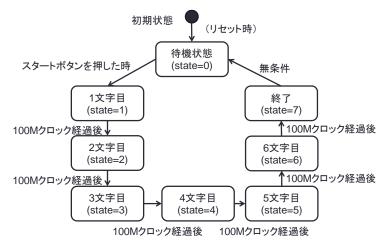


図 4 レポート課題3の状態遷移図

| 状態 (state) | 入力 | | 次状態 | 出力 | |
|---------------|--------------|-----------------|----------|----------|--------------|
| | BTNC (reset) | BTND (start) | (state') | LED[3:0] | 備考 |
| X | 1 | X | 0 | X | - |
| 0 | 0 | 0 | 0 | 0 | - |
| U | 0 | 1 | 1 | 0 | - |
| 1 | 0 | X | 2 | 2 | 100M クロック経過後 |
| 2 | 0 | X | 3 | 0 | 100M クロック経過後 |
| 3 | 0 | X | 4 | 2 | 100M クロック経過後 |
| 4 | 0 | X | 5 | 1 | 100M クロック経過後 |
| 5 | 0 | X | 6 | 9 | 100M クロック経過後 |
| 6 | 0 | X | 7 | 9 | 100M クロック経過後 |
| 7 | 0 | X | 0 | 0 | - |

表 3 レポート課題 3 の状態遷移表 (学籍番号: 202199 の場合)

表中のXは、任意の値を示す(Don't care という)。すなわち 1 行目は任意の状態において BTNC の値が 1 だったら、BTND の値に関わらず状態 0 に遷移することを示す。

³ カウンタを用いることで、任意の状態遷移を持つ順序回路を構成することができる。回路 の制御部となる順序回路をステートマシンと呼ぶ。

◆ 検証仕様

- ・シミュレータを用いて検証を行う。実時間(数秒間=数千万クロック)のシミュレーション結果を確認するのは時間がかかるので、シミュレータを用いた検証の際は、2クロックに1桁ずつLEDへの出力信号を変化させることとする。
- ・入力値の組合せに対する出力値を記録し、期待値と一致するかを確認する。結果は OK か NG かで示す。以下に検証に用いる表の例を示す。

表 4 レポート課題3の動作検証表の例

| クロック | 入 | 力 | 出力期待値 | 出力値 | 結果 |
|------|---------|---------|----------|----------|-------|
| 時刻 | BTNC | BTND | LED[3:0] | LED[3:0] | OK/NG |
| | (reset) | (start) | | | |
| 0 | 1 | 0 | 0 | | |
| 1 | 0 | | 0 | | |
| 2 | | 1 | 0 | | |
| 3 | | 0 | 2 | | |
| 4 | | | 2 | | |
| 5 | | | 0 | | |
| 6 | | | 0 | | |
| 7 | | | 2 | | |
| 8 | | | 2 | | |
| 9 | | | 1 | | |
| 10 | | | 1 | | |
| 11 | | | 9 | | |
| 12 | | | 9 | | |
| 13 | | | 9 | · | |
| 14 | | | 9 | | |
| 15 | | | 0 | | |
| 16 | | | 0 | | |

・信号の値は、シミュレーション波形の画面キャプチャをレポートに貼り付けること。

レポート課題 4: 少し複雑なディジタルシステムの実装・テスト

図 3に示す4ビット**算術論理演算装置**(ALU)を設計・実装・テストする。

【必須の実装仕様】

この ALU は、4 ビットデータ A,B、及びキャリーCin を入力とし、op で指定される演算を行い、演算結果 S、キャリーCout を出力する。また、S がゼロの時、Z に1を出力する(すなわち,Z はゼロフラグ)。キャリーCout は演算結果により桁上がりする,もしくは演算結果が負の値となる場合に1を出力する。作成した算術論理演算装置の動作検証には,次ページの表 5 に記載の入力値を用い,レポートにはシミュレーションの波形を記載すること。

【オプションの実装仕様】

FPGA ボードで動作検証を行う場合、入力は、スライドスイッチを使って A[3:0], B[3:0], op[2:0]を設定し、プッシュスイッチで Cin を設定する。出力は、S[3:0]を 1 ビットずつ"0" か"1"を 7 セグメント LED の 1 桁ごとに表示し、Cout と Z を LED に表示する。 7 セグメント LED 表示の参考プロジェクトは、講義資料ページにアップロードされている。 オプションの実装仕様を満たした場合は、レポートに FPGA ボードの画像を貼り付けること。

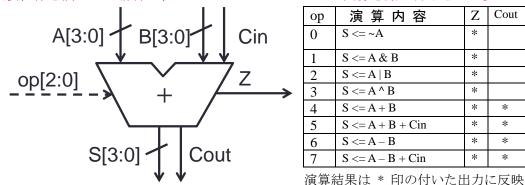


図 3 4ビット算術論理演算装置

表 5 レポート課題 5 の動作検証表

| op[3:0] | 入力 | | | 出力 | | |
|---------|--------|--------|-----|--------|------|---|
| | A[3:0] | B[3:0] | Cin | S[3:0] | Cout | Z |
| X | XXXX | XXXX | X | XXXX | X | X |
| 0 | 0000 | 0001 | 0 | | | |
| 1 | 0001 | 0000 | 0 | | | |
| 1 | 0001 | 0001 | 0 | | | |
| 2 | 0000 | 0001 | 0 | | | |
| 2 | 0000 | 0000 | 0 | | | |
| 3 | 0001 | 0001 | 0 | | | |
| 3 | 0001 | 0000 | 0 | | | |
| 4 | 1101 | 0010 | 0 | | | |
| 5 | 1111 | 0001 | 0 | | | |
| 6 | 1111 | 1010 | 0 | | | |
| 7 | 0101 | 0110 | 0 | | | |
| 5 | 1111 | 0001 | 1 | | | |
| 7 | 0101 | 0110 | 1 | | _ | |

レポート課題5: 考察課題

以下の 2 つについて考察し、述べよ。必ず参考文献を記載すること。Web サイトからのコピペは禁止!!コピペが発覚した場合、レポートは不合格とします。

- 1. C 言語と Verilog-HDL の主な類似点と相違点について論ぜよ。
- 2. 1 で述べた類似点と相違点を踏まえて、HDL ならびにシミュレーションを用いて論理回路設計を行う利点について述べよ。

オプション課題 (任意)

実施は必須ではないが、レポートに記載がある場合は評価の際に加点する。取り組むのは どちらか1つだけでもよい。

- ・オプション課題 1: 8 ビット簡易電卓(加算もしくは乗算)
- ・オプション課題 2: スロットマシンゲーム

・オプション課題 1: 8 ビット簡易電卓(加算もしくは乗算)

8 ビットの数字を加算もしくは乗算可能な、簡易電卓を設計・実装・テストする。

【必須】

計算結果は、16 進数の数字として 7 セグメント LED に表示する。ユーザは、値を設定するときはスライドスイッチ(8 ビット)を設定する。操作方法は以下の通りである。

- ・BTNLを押すと値1として記憶し、7セグメントLEDに表示する。
- ・BTNCを押すと値2として記憶し、7セグメントLEDに表示する。
- ・BTNR を押すと、2つの値の和を7セグメントLED に表示する。

【オプション】

・BTND を押すと、2つの値の積を7セグメント LED に表示する。

・オプション課題 2: スロットマシンゲーム

以下のゲームを設計・実装・テストする

【必須】

7 セグメント LED に数字 3 ケタを表示し、時間と共に高速に変化する。ユーザがプッシュボタンを押すと、数字が停止する。

【オプション】

システムは各桁の数字が全て一致している場合、「当たり」と判定して、ユーザに示す。 (例:LEDを賑やかに変化させる。)

「補足資料]

補足 1. Verilog-HDL シミュレーションにおいて信号値を出力する方法

レポート作成時、動作の様子を示すためには、画面写真をキャプチャして貼りつけること。信号値の変化をテキストとして表示したい場合、シミュレーションの際に、テストベンチの Verilog のモジュール内に図 4に示す記述を加えると、信号値の変化をテキストで出力することが出来る。(%b: 2 進数、%d:10 進数、%o:8 進数、%h:16 進数)

initial \$monitor(\$stime, "inO=%b in1=%b out=%b", inO, in1, out);

図 4 シミュレーションにおける信号値を出力するための記述例

補足 2. FPGA のコンパイル結果・リソース使用量・性能を知る方法

FPGA のコンパイル(合成・配置配線)を行った後、Vivado の Design Summary を見ることで、リソース使用量や性能を知ることが出来る。(画面右の Project Summary ウィンドウ内で、下にスクロールするとグラフが現れる)

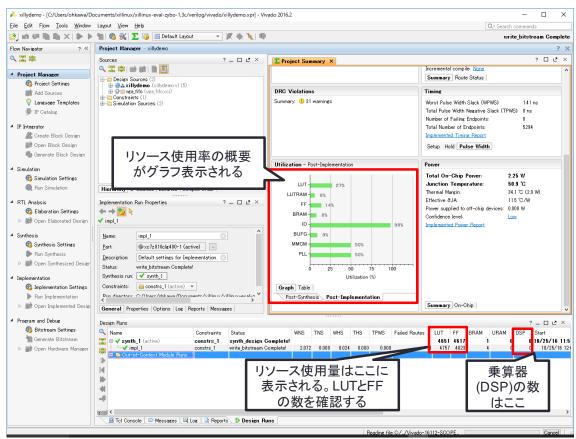


図 5 Vivado の Project Summary の読み方