

## HDL によるハードウェア設計 レポート課題

### [レポート作成ガイドライン] 必読!!

#### ◆レポート全体について

- ・実験指導書・テキスト冊子・講義資料の内容のコピーはしない。
- ・**実験指導書やテキスト冊子、講義資料に記載されている図表を無断で転載しない** (著作権侵害)。
- ・他の IT 技術者が読んで理解できるよう、この実験固有の最低限の情報のみを記載する。
- ・実験時の注意事項をレポートに書かない。
- ・やったことを現在形で書かない、完了形で書く。(普遍的な事実については現在形)

#### ◆レポートに何を書くか？

- ・本レポートにおいては、システムの設計・実装・検証結果をそれぞれ書くこと。
  - ✓ 設計とは、作るものが何であるか (仕様)
  - ✓ 実装とは、作ったもの (ソースコード)
  - ✓ 検証とは、「正しいことを調べる」という意味。つまり仕様通りに正しく実装したかを確かめること
  - ✓ 検証結果とは、どのように確かめたか、確かめた結果が OK か NG か
- ・課題によってシミュレーションのみでの検証や FPGA ボードを使った検証があることに注意

#### ◆結果と考察について

- ・実験結果を見て即座にわかることは、結果の図面とともに文章で示す。これが「結果」。
- ・結果を比較してわかることや、文献調査を併せたものが「考察」。

#### ◆課題の考察

- ・実験レポートにおける考察とは、実験データを基に議論すること。
- ・対象の HDL 記述やデータをソースコード行や図番等で示し、そこから引用して議論すること。  
(元の記述やデータが分からないと議論にならない)

#### ◆まとめ→目的

- ・まとめ (結論) は、目的に対応させて書くこと。
- ・目的は、「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。」こと。
- ・目的に対して単に「理解した」というのはあいまい。目的を分解し、サブ目的にする必要がある
- ・サブ目的を達成することで、全体の目的を達成した、という意味合いのまとめの文章を自分の言葉で作文すること。

#### ◆考察課題

- ・「論ずる」とは、必要に応じて文献等を調べ、必要に応じて引用しながら自分の考えの道筋を文章で著すこと、調べた内容を示すだけでは不足。
- ・**Web サイトからのコピー禁止**。参考にするのは構わないが、レポートには**自分の言葉で書くこと**。

[課題]

レポート課題 1 から 5 はすべて必須課題である。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/2024/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、本年度の数値 (10 進数の 2024) を加算した値を S (Sum の略)とし、FPGA ボード上の LED (LD15~LD0) 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

以下の(1)と(2)の 2 つの方法で検証を行うこと。

(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。

表 1 レポート課題 1 の動作検証表

テスト番号	入力		出力期待値		出力値 S[15:0]	結果 OK/NG
	A[7:0]	B[7:0]	10 進数	S[15:0]		
1	0	0	2024	16'b0000011111101000		
2	1	0	2024	16'b0000011111101000		
3	0	1	2024	16'b0000011111101000		
4	1	1	2025			
5	2	3	2030			
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 : 202599 入力 A=99、入力 B=100 出力期待値  $S = 99 * 100 + 2024 = 11924 = 16'b0010111010010100$  (10 進数の 11924 を 16bit の 2 進数で表した場合)

## 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出力が 0 になるようにすること。プッシュボタン BTNC (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。

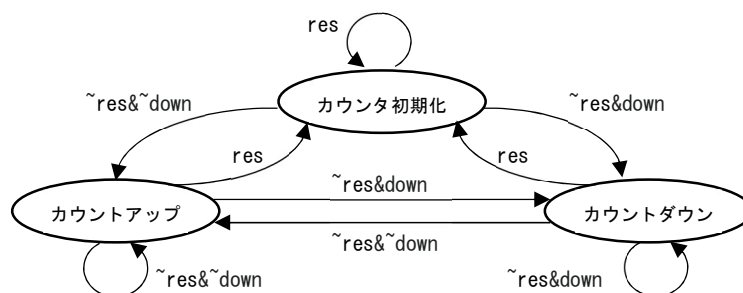


図 1 アップダウンカウンタの状態遷移図

表 2 レポート課題 2 の状態遷移表

クロック時刻	入力		出力	備考
	BTNC (reset)	SW0 (down)	LED[3:0]	
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、クロック時刻などは適宜変更して構わない。また、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。

表 3 レポート課題2の動作検証表の例

クロック時刻	入力		出力期待値	出力値	結果
	BTNC (reset)	SW0 (down)	LED[3:0]	LED[3:0]	OK/NG
0	1	0	0		
1	0	0	0		
2	0	0	1		
3	0	0	1		
4	0	0	2		
5	0	0	2		
6	0	0	3		
7	0	0	3		
8	0	1	2		
9	0	1	2		
10	0	1	1		
11	0	1	1		
12	0	0	2		
13	0	0	2		
14	0	0	3		
15	0	0	3		
16	1	0	0		
17	1	0	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。

1. 2クロックごとのカウントアップ
2. RESET信号の入力による初期化
3. down信号の入力による2クロックごとのカウントダウン  
(down信号を数クロックの間1とし、その間カウントダウンがされ続ける様子)
4. down信号が1から0になった際の2クロックごとのカウントアップ
5. 15までカウントアップしたあとに、出力が0に戻る様子

いずれもカウントアップ/カウントダウンが開始されたクロックだけでなく、その後もカウントアップ/カウントダウンが行われている様子が分かるように数クロック分のシミュレーション波形を記載すること。なお、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。

### 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にする。

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<sup>1</sup>を作成する。

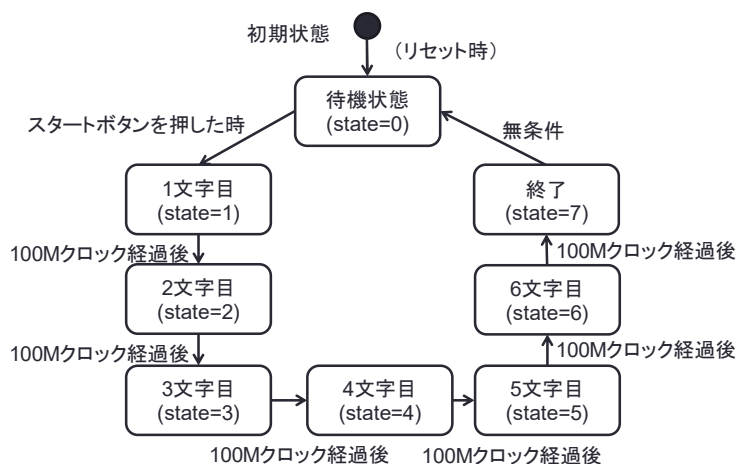


図 2 レポート課題 3 の状態遷移図

表 4 レポート課題 3 の状態遷移表 (学籍番号 : 202599 の場合)

状態 (state)	入力		次状態 (state')	出力		備考
	BTNC (reset)	BTND (start)		LED[3:0]		
X	1	X	0	X		-
0	0	0	0	0		-
	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	5		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。

<sup>1</sup> 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 5 レポート課題3の動作検証表の例（学籍番号：202599の場合）

クロック時刻	入力		出力期待値	出力値	結果	
	BTNC (reset)	BTND (start)	LED[3:0]	LED[3:0]	OK/NG	
0	1	0	0			
1	0	0	0			
2			1	0		
3			0	2		
4			2			
5			0			
6			0			
7			2			
8			2			
9			5			
10			5			
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。なお、表 6 に示すとおり op が 0,1,2,3 の場合、キャリーCout の出力は任意の値でよい。作成した算術論理演算装置の動作検証には、表 7 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] および 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。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。

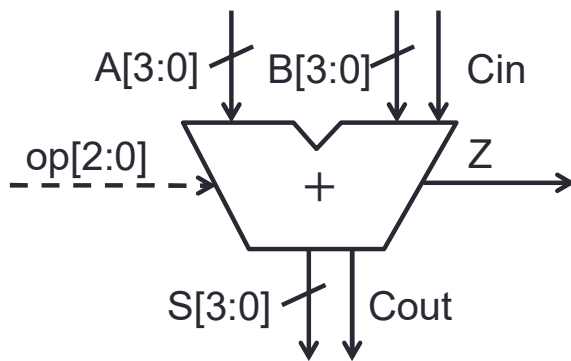


図 3 4 ビット算術論理演算装置

表 6 満たすべき出力

op	演算内容	Z	Cout
0	$S \leq \sim A$	*	X
1	$S \leq A \& B$	*	X
2	$S \leq A   B$	*	X
3	$S \leq A \wedge B$	*	X
4	$S \leq A + B$	*	*
5	$S \leq A + B + Cin$	*	*
6	$S \leq A - B$	*	*
7	$S \leq A - B + Cin$	*	*

演算結果は \* 印の付いた出力に反映  
X の場合は任意の出力値でよい

表 7 レポート課題 5 の動作検証表

op[2:0]	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、指定されたキーワードを含めて述べよ。必ず参考文献を記載すること。なお、講義資料およびテキスト冊子は参考文献にならないので注意する。**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する。**

1. C 言語と Verilog-HDL の主な類似点と相違点について以下のキーワードを含めて論ぜよ。

キーワード: FPGA ボード, シミュレーション, ハードウェア, ソフトウェア

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以下のキーワードを含めて述べよ。

キーワード: ハードウェア, コスト, 実装, 検証

### **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, "in0=%b in1=%b out=%b", in0, 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)

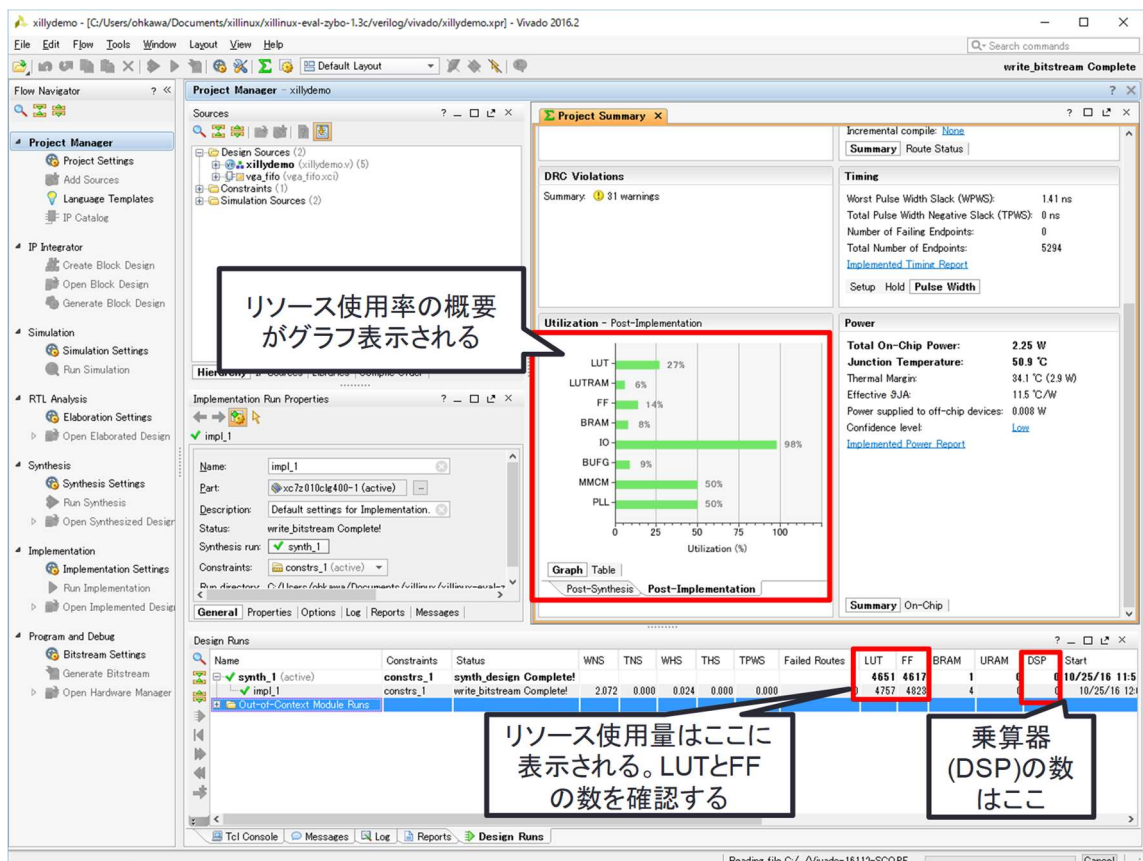


図 5 Vivado の Project Summary の読み方