

# VHDL 入門編トライアル・コース 演習マニュアル

## この演習を完了させるのに必要なアイテム

- ・ ModelSim®-Altera® Edition 10.4b (Quartus® Prime 15.1) または
- ModelSim-Altera Starter Edition 10.4b (Quartus Prime 15.1)
- ※ 上記ツールバージョン以外でも演習を行うことはできますが、メニューの位置や操作方 法など異なる場合があります。



2016年4月 Rev.1





## VHDL 入門編トライアル・コース 演習マニュアル

## <u>目次</u>

はじめに	3
演習 1	4
演習 2	9
演習 3	11
演習 /	
次日 +	13
	13
	17
改版履歴	19

## 🔥 ALTIMA

## <u>はじめに</u>

- 本マニュアルは、演習用データの保存先を C:¥lab¥vhdl\_lab として説明しています。もしあなたの保存した先が C:¥lab¥vhdl\_lab と異なる場合は、マニュアル内のパスを自分の環境に合わせ適応してください。
- この演習では、実際に記述した VHDL の回路が期待した動作をするかを確認するために、ファンクション・シミュレーションを実行します。その際に使用するソフトウェアは、ModelSim-Altera です。
- 演習で使用するためテストベンチ・ファイルは、あらかじめ作成済みのものを使用します。
- この演習におけるツールの操作および設定は、本演習に限定した内容です。
- 本コースは言語のトライアル・コースです。ツール操作を習得するコースではないため、操作の解説は省略 しています。ご了承ください。
- 演習データは、別途案内する Web ページからダウンロードすることができます。ダウンロードした ZIP ファイルを解凍すると、EXE ファイルが生成されます。その EXE ファイルを実行して、演習データの保存先を指定してください。(デフォルトの C:¥lab¥vhdl\_lab でも良いですが、お好みに応じて保存先を変更することもできます。

## 入ALTIMA 演習1

<目的>

- 記述の穴埋めをし、加算器を完成させます。
- ファンクション・シミュレーションを実行し、動作を確認します。



## ステップ1: シミュレーション用プロジェクトの作成

- 1. ModelSim-Altera を起動します。
- 2. File メニュー  $\Rightarrow$  New  $\Rightarrow$  Project を選択して、Create Project ダイアログ・ボックスを開きます。
- 3. Project Name 欄に、adder と記述します。Project Location 欄に、C:/lab/vhdl\_lab/lab1 (作業ディレクトリ) を選択します。それ以外はデフォルト状態のままで OK ボタンをクリックします。

Add items to the Project ウィンドウが表示されますが、今回は使用しません。Close ボタンで閉じます。

M Create Project
Project Name
adder
Project Location
C:/lab/vhdl_lab/labl Browse
Default Library Name
Copy Settings From
/modelsim_ae/modelsim.ini Browse
Copy Library Mappings C Reference Library Mappings
OKCancel

ステップ 2: デザインの作成とプロジェクトへの登録

- 1. File メニュー ⇒ Open を選択して、作業ディレクトリ内に保存されている adder.vhd を選択し開きます。
- ソースコードを確認します。エンティティ宣言部分が記述されていません。上記の回路仕様を参考にし、加算器のエンティティ名やポート名などの構成を記述して、デザインを完成させてください。
- 3. 記述終了後、File メニュー ⇒ Save にて、完成させた VHDL ファイルを adder.vhd として作業ディレクト リ内に上書き保存します。その後、addrer.vhd ファイルを閉じます。
- Project メニュー ⇒ Add to Project ⇒ Existing File を選択します。Add file to Project ウィンドウで Brose... ボタンをクリックして、adder.vhd と adder\_sim.vhd (テストベンチ)を選択して OK をクリックします。 Project ウィンドウには、下図のように指定した2ファイルが登録されます。

🔛 Project - C:/lab/vhdl_la	ab/lab1/adder		
▼Name	Status	Туре	Orde
adder.vhd	?	VHDL	1 (
adder_sim.vhd	?	VHDL	0 (
Library 🗶 🛗 Projec	t ×		

#### <u>ステップ 3: ソースコードをコンパイル</u>

- Compile メニュー ⇒ Compile Order を選択し、コンパイルの実行順序を設定します。Compile Order ウィンドウの上段に adder.vhd、下段に adder\_sim.vhd を配置後、OK ボタンをクリックして順番を確定します。
- Compile メニュー ⇒ Compile All により、ソースコードのコンパイルを実行します。エラーが出る場合には、 記述において文法的な間違いがあります。エラーメッセージをヒントに記述を修正してください。修正後は再 びコンパイルを実行し、エラーが無くなるまで記述のデバッグを行います。

🛅 Project - C:/lab/vhdl_	jab/lab1/adder d		
Name	⊽ Status	Туре	Order
📄 adder_sim.vhd	1	VHDL	1
👕 adder.vhd	×	VHDL	0

コンパイルに成功すると、ステータスが 🗸 マークに変わり、成功したという内容のメッセージも表示されます。

******					
**	Project - C:/lab/vhdl_la	ab/lab1/adder			
₹N	ame	⊽ Status	Туре	Order	
	📄 adder_sim.vhd		VHDL	1	
	adder.vhd	1	VHDL	0	
		-			エラーなし
#	Commile of adder	ubd was	3110083	ful	7
17.	compile of dddei		Jacobb		
#	Compile of addei	:_sim.vhd v	yas suc	cessful.	•
#	2 compiles, O fa	ailed with	no err	ors.	

## <u>ステップ 4: デザインのロード</u>

- 1. Simulator メニュー ⇒ Start Simulation を選択し、Start Simulation ダイアログ・ボックスを起動します。
- 2. Design タブより work ディレクトリを展開して、adder\_sim の下位にある adder を選択し OK ボタンをク リックします。デザインのロードが開始されます。

Design VHDL Verilog Libraries SDF Others	4
* Name	•
🔁 🗄 adder	
E adder_sim	
H-JII 220model ver	
The second seco	
+altera_Insim_ver	
· → /h altera_mf	•
	►
Design Unit(s)	Resolution
work.adder_sim(adder)	default 💌

3. ロードが問題なく終了すると、sim タブ(sim ウィンドウ)が追加され、関連ウィンドウが起動します。

🙀 sim - Default		💫 Objects 💷		;;=		
▼ Instance	Design unit	▼ Name			Value	16 🗖
adder_sim	adder_sim(,	💽			00000000.	Sigr
🕂 🗐 u1	adder(logic)	💶 🔶 adder_b			00000000.	Sigr
line29	adder_sim(.	🛨 🔶 sum			UUUUUUU	Sigr
line38	adder_sim(,	🖅 🔶 inveca			00000000	Sigr
line46	adder_sim(.	🖅 🔶 invecb			00000000.	Sigr
L line47	adder_sim(,	A pariod a			50000 pc	Cor
📕 standard	standard					
🗾 textio	textio	🚳 Processes (Active) 💳				
std_logic_1164	std_logic_1.	Name	Type (filtered)	State	Order	Pare
std_logic_arith	std_logic_a.	line 47	VHDL Process	Active	1	ladd.
sta_logic_unsigned	sta_logic_u.	line 46	VHDL Process	Ready		ladd
		line 38	VHDL Process	Ready		Jadd
		line 29	VHDL Process	Ready		Jadd
		🎒 line _15	VHDL Process	Ready		/add
•	•					
👖 Library 🗶 🛗 Project 🗙 🌄 sim 🗙	< >	•				

## \Lambda ALTIMA 🗕

ステップ 5: ファンクション・シミュレーションの実行

1. sim ウィンドウ内の adder\_sim の下位にある u1 を選択し、右クリック ⇒ Add to ⇒ Wave ⇒ All items in region を選択します。

💊 Objects 🗉 🛺 sim - Default 😑 **= +** ₫ × - + a × 🔢 Wave - Default Instance ▼ Name 😢 🗖 Now 🌶 🕨 🔶 Design unit <u></u> Msas adder\_sim(. View Declaration 🥘 line\_ adder\_sim(, 🥘 line\_ View Instantiation adder\_sim(. line\_ adder\_sim(. UVM adder\_sim(. 🗾 standarc • ► standard UPF ⊩ 🗾 textio textio nocesses (Active) - + a × 🗾 std\_logic std logic 1. Add Wave Ctrl+W ▼ Name Type (filtered) 🗾 std\_logic std\_logic\_a. Add Wave To 🗾 std\_logid std\_logic\_u. 3 Add Dataflow Ctrl+D Add to Wave All items in region All items in region and below List Сору Ctrl+C Log All items in design Find... Ctrl+F Now AL 🖸 🕤 0 ps Dataflow Save Selected... Watch ۲

VHDL デザインの入出カピンが登録された状態で Wave ウィンドウが起動します。

2. Transcript ウィンドウに以下の実行コマンドを入力し、Enter キーでシミュレーション実行を開始します。 run\_1us

※\_\_ は半角スペース

Wave ウィンドウにシミュレーション結果が波形表示されます。作成した回路が正しく動作しているか確認しましょう。Wave メニュー  $\Rightarrow$  Zoom  $\Rightarrow$  ···、またはツールバーのアイコンにて波形を明確に確認できます。 信号の Radix を変更する場合は、信号選択後に右クリック  $\Rightarrow$  Radix  $\Rightarrow$  目的の表示を選択してください。

💶 Wave - Default 💷 💷		
<b>≈</b> •	Msgs	
🕞 < /adder_sim/u1/a	21	20
🕀 🔶 /adder_sim/u1/b	15	14
🕞 🕁 /adder_sim/u1/sum_	36	34
¢	Object Declaration	
	Add 🕨	
	Edit 🕨	
	View 🕨	
	UPF 🕨	
	Radix 🔸	Global Signal Radix
	Format 🕨 🕨	Sumbolic
	Cast to 🔹 🕨	Biserry
	Carebia a Circala	Octobel Control Control
	Compline Signals	Uccal
	Group	Decimal
	Ungroup	✓ Unsigned
	<b>F</b>	Hexadecimal



以下の例のように、期待する動作(加算)をしていますか?

📕 Wave - Default 🚞		;;;;;;;;											
\$1.	Msgs												
🖅 🎝 /adder_sim/u1/a	15	(1	(2	3	(4	5	<u>(</u> 6	7	(8	9	(10	11 )	
🖅 🎝 /adder_sim/u1/b	12	(5		6		7		8		9		10	
💶 👆 /adder_sim/u1/sum	27	6	7	9	(10	12	13	15	<u>(16</u>	18	<u>)</u> 19	21 )	

様々なツールバーを使って波形を見やすくしてください。主なものを紹介します。

- 🔤 : ピン名の表示方法の切り替え
- 日:指定した範囲を拡大して表示
- 🔍 : 波形の全体表示
- 🕙 : 拡大表示
- 🔍 : 縮小表示

#### ステップ 6: シミュレーションおよびプロジェクトの終了

1. Simulate メニュー⇒ End Simulation よりシミュレーションを終了します。メッセージが表示されますので "はい(Y)"を選択します。



2. メイン・ウィンドウの Project タブをアクティブにし、File メニュー ⇒ Close Project を選択します。メッセージが表示されますので "はい(Y)"を選択し、プロジェクトを終了します。

## これで、演習1は終了です。

## <u>演習2</u>

#### <目的>

- when-else 文で比較器を作成します。
- ファンクション・シミュレーションを実行し、動作を確認します。



#### <動作条件>

入力信号 da と db を比較します。

- da と db が等しい場合は、equ は High (1) を出力し、それ以外のときは Low (0) を出力
- ▶ da が大きい場合は、agb は High(1)を出力し、それ以外のときは Low(0)を出力
- ➢ db が大きい場合は、alb は High (1) を出力し、それ以外のときは Low (0) を出力

#### ステップ1: シミュレーション用プロジェクトの作成

- 1. ModelSim-Altera が起動していない場合は、ModelSim-Altera を起動します。
- 2. File メニュー  $\Rightarrow$  New  $\Rightarrow$  Project を選択して、Create Project ダイアログ・ボックスを開きます。
- 3. Project Name 欄に、adder と記述します。Project Location 欄に、C:/lab/vhdl\_lab/lab2(作業ディレクトリ) を選択します。それ以外はデフォルト状態のままで OK ボタンをクリックします。

Add items to the Project ウィンドウが表示されますが、今回は使用しません。Close ボタンで閉じます。

#### ステップ 2: デザインの作成とプロジェクトへの登録

- 1. File メニュー  $\Rightarrow$  New  $\Rightarrow$  Source  $\Rightarrow$  VHDL よりテキスト・エディタを開きます。
- 2. ソースコードを記述します。上記の回路仕様を参考にし、比較器を完成させてください。
- 3. 記述終了後、File メニュー ⇒ Save As より、完成させた VHDL ファイルを compare.vhd として作業ディ レクトリ内に保存します。その後、compare.vhd ファイルを閉じます。
- 4. Project メニュー ⇒ Add to Project ⇒ Existing File を選択し、compare.vhd と compare\_sim.vhd(テスト ベンチ)を指定します。Project ウィンドウには、指定した 2 ファイルが登録されます。

## <u>ステップ 3: コンパイル</u>

\Lambda ALTIMA

- Compile メニュー ⇒ Compile Order にて、コンパイルの実行順序を設定します。Compile Order ウィンド ウの上段に compare.vhd、下段に compare\_sim.vhd を配置後、OK ボタンをクリックして順番を確定しま す。
- Compile メニュー ⇒ Compile All により、コンパイルを実行します。エラーが出る場合には、記述において文法的な間違いがあります。エラーメッセージをヒントに記述を修正してください。修正後は再びコンパイルを実行し、エラーが無くなるまで記述のデバッグを行います。

## <u>ステップ 4: ロード</u>

- 1. Simulate  $\lambda = \neg \rightarrow$  Start Simulation を選択し、Start Simulation ダイアログ・ボックスを起動します。
- 2. Design タブより work ディレクトリを展開し、compare\_sim の下位にある compare を指定後 OK ボタン をクリックします。デザインのロードが開始されます。
- 3. ロードが問題なく終了すると、sim タブ(sim ウィンドウ)が追加され、関連ウィンドウが起動します。

#### ステップ 5: ファンクション・シミュレーションの実行

- 1. Sim ウィンドウ内の compare\_sim の下位にある u1 を選択し、右クリック ⇒ Add to ⇒ Wave ⇒ All items in region を選択します。すると VHDL デザインの入出カピンが登録された状態で Wave ウィンドウ が起動します。
- 2. Transcript ウィンドウに以下の実行コマンドを入力し、Enter キーでシミュレーション実行を開始します。

run\_lus ※\_は半角スペース

 Wave ウィンドウにシミュレーション結果が波形表示されます。作成した回路が正しく動作しているか確認しましょう。Wave メニュー ⇒ Zoom ⇒ …、またはツールバーのアイコンにて波形を明確に確認できます。 信号の Radix を変更する場合は、信号選択後に右クリック ⇒ Radix ⇒ 目的の表示を選択してください。

😦 Wave - Default 💷 💷	Wave - Default														
💫 🗸	Msgs														
🖅 🎝 /compare_sim/u1/da	-No Data-	1	(2	3	(4	15	<u></u> ί6	7	(8	9	(10	11	12	13	14
	-No Data- -No Data-	9	(8	7	(6	15	<u>)</u> 4	3	<u></u> ζ2	1	<u>χο</u>	255	<u>)</u> 254	253	252
/compare_sim/u1/agb	-No Data-							$\vdash$		$\vdash$					
/compare_sim/u1/alb	-No Data-					┣									

## ステップ 6: シミュレーションおよびプロジェクトの終了

- 1. Simulate メニュー⇒ End Simulation よりシミュレーションを終了します。メッセージが表示されますので "はい(Y)"を選択します。
- 2. メイン・ウィンドウの Project タブをアクティブにし、File メニュー ⇒ Close Project を選択します。メッセージが表示されますので "はい(Y)"を選択し、プロジェクトを終了します。

## これで、演習2は終了です。



## <u>演習3</u>

<目的>

- process 文で乗算器を作成します。
- ファンクション・シミュレーションを実行し、動作を確認します。



## <u>ステップ1:シミュレーション用プロジェクトの作成</u>

- 1. ModelSim-Altera が起動していない場合は、ModelSim-Altera を起動します。
- 2. File メニュー  $\Rightarrow$  New  $\Rightarrow$  Project を選択して、Create Project ダイアログ・ボックスを開きます。
- Project Name 欄に、Project Name 欄に、mult4x4 と記述します。Project Location 欄に C:/lab/vhdl\_lab/lab3 (作業ディレクトリ)を選択します。それ以外はデフォルト状態のままで OK ボタンを クリックします。

Add items to the Project ウィンドウが表示されますが、今回は使用しません。Close ボタンで閉じます。

## <u>ステップ 2: デザインの作成とプロジェクトへの登録</u>

- 1. File メニュー  $\Rightarrow$  New  $\Rightarrow$  Source  $\Rightarrow$  VHDL よりテキスト・エディタを開きます。
- 2. ソースコードを記述します。上記の回路仕様を参考にし、乗算器を完成させてください。
- 3. 記述終了後、File メニュー ⇒ Save As より、完成させた VHDL ファイルを mult4x4.vhd として作業ディ レクトリ内に保存します。その後、mult4x4.vhd ファイルを閉じます。
- 4. Project メニュー ⇒ Add to Project ⇒ Existing File を選択し、mult4x4.vhd と mult4x4\_sim.vhd(テスト ベンチ)を指定します。Project ウィンドウには、指定した 2 ファイルが登録されます。

## <u>ステップ3: コンパイル</u>

- Compile メニュー ⇒ Compile Order にて、コンパイルの実行順序を設定します。Compile Order ウィンド ウの上段に mult4x4.vhd、下段に mult4x4\_sim.vhd を配置後、OK ボタンをクリックして順番を確定しま す。
- Compile メニュー ⇒ Compile All により、コンパイルを実行します。エラーが出る場合には、記述において文法的な間違いがあります。エラーメッセージをヒントに記述を修正してください。修正後は再びコンパイルを実行し、エラーが無くなるまで記述のデバッグを行います。

## ステップ 4: ロード

- 1. Simulate メニュー ⇒ Start Simulation を選択し、Start Simulation ダイアログ・ボックスを起動します。
- 2. Design タブより work ディレクトリを展開し、mult4x4\_sim の下位にある mult4x4 を指定後 OK ボタン をクリックします。
- 3. ロードが開始され問題なく終了すると、sim タブ(sim ウィンドウ)が追加され、関連ウィンドウが起動しま す。

#### ステップ 5: ファンクション・シミュレーションの実行

- sim ウィンドウ内の mult4x4\_sim の下位にある u1 を選択し、右クリック ⇒ Add to ⇒ Wave ⇒ All items in region を選択します。すると VHDL デザインの入出カピンが登録された状態で Wave ウィンドウ が起動します。
- 2. Transcript ウィンドウに以下の実行コマンドを入力し、Enter キーでシミュレーション実行を開始します。

run\_lus ※\_は半角スペース

 Wave ウィンドウにシミュレーション結果が波形表示されます。作成した回路が正しく動作しているか確認しましょう。Wave メニュー ⇒ Zoom ⇒ …、またはツールバーのアイコンにて波形を明確に確認できます。 信号の Radix を変更する場合は、信号選択後に右クリック ⇒ Radix ⇒ 目的の表示を選択してください。

🝙 Wave - Default 🔤 🔤		±										
<b>*</b>	Msgs											
	5	1	2	3	4	5	6	7	8	9	10	
💶 🤣 /mult4×4_sim/mult_b	15	5		<b>1</b> 6		7		8		9	L I	
=	75	5	10	18	24	35	42	56	64	81	(90 )	
r												

## ステップ 6: シミュレーションおよびプロジェクトの終了

- 1. Simulate メニュー ⇒ End Simulation よりシミュレーションを終了します。メッセージが表示されますので "はい(Y)"を選択します。
- 2. Workspace ウィンドウの Project タブをアクティブにし、File メニュー ⇒ Close Project を選択します。メッセ ージが表示されますので "OK" を選択し、プロジェクトを終了します。

## これで、演習3は終了です。

## 演習 4

<目的>

- 演習 2 の比較気を if-else 文で作成します。
- ファンクション・シミュレーションを実行し、動作を確認します。



<動作条件>

入力信号 da と db を比較します。

- da と db が等しい場合は、equ は High (1) を出力し、それ以外のときは Low (0) を出力
- ▶ da が大きい場合は、agb は High(1)を出力し、それ以外のときは Low(0)を出力
- ▶ db が大きい場合は、alb は High (1) を出力し、それ以外のときは Low (0) を出力

## <u>ステップ1: プロジェクトの作成</u>

- 1. ModelSim-Altera が起動していない場合は、ModelSim-Altera を起動します。
- 2. File メニュー  $\Rightarrow$  New  $\Rightarrow$  Project を選択して、Create Project ダイアログ・ボックスを開きます。
- 3. Project Name 欄に、Project Name 欄に、compare\_if と記述します。Project Location 欄に C:/lab/vhdl\_lab/lab4 (作業ディレクトリ)を選択します。それ以外はデフォルト状態のままで OK ボタンを クリックします。

Add items to the Project ウィンドウが表示されますが、今回は使用しません。Close ボタンで閉じます。

#### ステップ 2: デザインの作成とプロジェクトへの登録

- 1. File メニュー  $\Rightarrow$  New  $\Rightarrow$  Source  $\Rightarrow$  VHDL よりテキスト・エディタを開きます。
- 2. ソースコードを記述します。上記の回路仕様を参考にし、比較器を完成させてください。
- 3. 記述終了後、File メニュー ⇒ Save As より、完成させた VHDL ファイルを compare\_if.vhd として作業 ディレクトリ内に保存します。その後、compare\_if.vhd ファイルを閉じます。
- 4. Project メニュー ⇒ Add to Project ⇒ Existing File を選択し、compare\_if.vhd と compare\_if\_sim.vhd (テストベンチ)を指定します。Project ウィンドウには、指定した 2 ファイルが登録されます。

## ステップ 3: コンパイル

\Lambda ALTIMA

- 1. Compile メニュー ⇒ Compile Order にて、コンパイルの実行順序を設定します。Compile Order ウィンド ウの上段に compare\_if.vhd、下段に compare\_if\_sim.vhd を配置後、OK ボタンをクリックして順番を確 定します。
- Compile メニュー ⇒ Compile All により、コンパイルを実行します。エラーが出る場合には、記述において文法的な間違いがあります。エラーメッセージをヒントに記述を修正してください。修正後は再びコンパイルを実行し、エラーが無くなるまで記述のデバッグを行います。

#### <u>ステップ 4: ロード</u>

- 1. Simulate メニュー ⇒ Start Simulation を選択し、Start Simulation ダイアログ・ボックスを起動します。
- 2. Design タブより work ディレクトリを展開し、compare\_if\_sim の下位にある compare\_if を指定後 OK ボタンをクリックします。
- 3. ロードが開始され問題なく終了すると、sim タブ(sim ウィンドウ)が追加され、関連ウィンドウが起動しま す。

#### ステップ 5: ファンクション・シミュレーションの実行

- sim ウィンドウ内の compare\_if\_sim の下位にある u1 を選択し、右クリック ⇒ Add to ⇒ Wave ⇒ All items in region を選択します。すると VHDL デザインの入出カピンが登録された状態で Wave ウィンドウ が起動します。
- 2. Transcript ウィンドウに以下の実行コマンドを入力し、Enter キーでシミュレーション実行を開始します。

run\_1us ※\_\_*は半角スペース* 

 Wave ウィンドウにシミュレーション結果が波形表示されます。作成した回路が正しく動作しているか確認しましょう。Wave メニュー ⇒ Zoom ⇒ …、またはツールバーのアイコンにて波形を明確に確認できます。 信号の Radix を変更する場合は、信号選択後に右クリック ⇒ Radix ⇒ 目的の表示を選択してください。

	😰 Wave - Default 🚃																
ľ	<u></u>	Msg	s														
ſ	💶 🧄 /compare_if_sim/da	21		1	2	3	<u>(</u> 4	5	<u>(</u> 6	7	<u>(8</u>	9	<u>(10</u>	11	12	13	14
	💶 🧄 /compare_if_sim/db	245	U	9	(8	7	<u>(</u> 6	5	<u>(</u> 4	3	2	1	<u>(</u> 0	255	254	253	252
	🔶 /compare_if_sim/equ	0	U						1								
	🔷 /compare_if_sim/agb	0	U											1			
	<pre>/compare_if_sim/alb</pre>	1															

## ステップ 6: シミュレーションおよびプロジェクトの終了

- 1. Simulate メニュー ⇒ End Simulation よりシミュレーションを終了します。メッセージが表示されますので "はい(Y)"を選択します。
- 2. Workspace ウィンドウの Project タブをアクティブにし、File メニュー ⇒ Close Project を選択します。メッセ ージが表示されますので "OK" を選択し、プロジェクトを終了します。

## <u>これで、演習4は終了です。</u>

## 入ALTIMA 演習 5

<目的>

- Case 文を使用し、マルチプレクサを作成します。
- ファンクション・シミュレーションを実行し、動作を確認します。



<動作条件>

- ▶ セレクト・コントロール信号 (sel) が Low (0) ならば a[3..0] を出力
- ▶ セレクト・コントロール信号 (sel) が High (1) ならば b[3..0] を出力

## <u>ステップ1: プロジェクトの作成</u>

- 1. ModelSim-Altera が起動していない場合は、ModelSim-Altera を起動します。
- 2. File メニュー  $\Rightarrow$  New  $\Rightarrow$  Project を選択して、Create Project ダイアログ・ボックスを開きます。
- Project Name 欄に、Project Name 欄に、mux4 と記述します。Project Location 欄に C:/lab/vhdl\_lab/lab5 (作業ディレクトリ)を選択します。それ以外はデフォルト状態のままで OK ボタンをクリックします。

Add items to the Project ウィンドウが表示されますが、今回は使用しません。Close ボタンで閉じます。

## <u>ステップ 2: デザインの作成とプロジェクトへの登録</u>

- 1. File メニュー  $\Rightarrow$  New  $\Rightarrow$  Source  $\Rightarrow$  VHDL よりテキスト・エディタを開きます。
- 2. ソースコードを記述します。上記の回路仕様を参考にし、マルチプレクサを完成させてください。
- 3. 記述終了後、File メニュー ⇒ Save As より、完成させた VHDL ファイルを mux4.vhd として作業ディレ クトリ内に保存します。その後、mux4.vhd ファイルを閉じます。
- 4. Project メニュー ⇒ Add to Project ⇒ Existing File を選択し、mux4.vhd と mux4\_sim.vhd(テストベンチ) を指定します。Project ウィンドウには、指定した 2 ファイルが登録されます。

## <u>ステップ 3: コンパイル</u>

- 1. Compile メニュー ⇒ Compile Order にて、コンパイルの実行順序を設定します。Compile Order ウィンド ウの上段に mux4.vhd、下段に mux4\_sim.vhd を配置後、OK ボタンをクリックして順番を確定します。
- Compile メニュー ⇒ Compile All により、コンパイルを実行します。エラーが出る場合には、記述において文法的な間違いがあります。エラーメッセージをヒントに記述を修正してください。修正後は再びコンパイルを実行し、エラーが無くなるまで記述のデバッグを行います。

## ステップ 4: ロード

- 1. Simulate メニュー ⇒ Start Simulation を選択し、Start Simulation ダイアログ・ボックスを起動します。
- 2. Design タブより work ディレクトリを展開し、mux4\_sim の下位にある mux4 を指定後 OK ボタンをクリ ックします。
- 3. ロードが開始され問題なく終了すると、sim タブ(sim ウィンドウ)が追加され、関連ウィンドウが起動しま す。

#### ステップ 5: ファンクション・シミュレーションの実行

- sim ウィンドウ内の mux4\_sim の下位にある u1 を選択し、右クリック ⇒ Add to ⇒ Wave ⇒ All items in region を選択します。すると VHDL デザインの入出カピンが登録された状態で Wave ウィンドウが起 動します。
- 2. Transcript ウィンドウに以下の実行コマンドを入力し、Enter キーでシミュレーション実行を開始します。

run\_lus ※\_*は半角スペース* 

 Wave ウィンドウにシミュレーション結果が波形表示されます。作成した回路が正しく動作しているか確認しましょう。Wave メニュー ⇒ Zoom ⇒ …、またはツールバーのアイコンにて波形を明確に確認できます。 信号の Radix を変更する場合は、信号選択後に右クリック ⇒ Radix ⇒ 目的の表示を選択してください。

wave - Default											
<u></u>	Msgs										
/mux4_sim/sel	0										
💶 🔶 /mux4_sim/a	9	9									
💶 🔶 /mux4_sim/b	7	7									
💶 🛶 /mux4_sim/y	9	9	7	9	(7	9	χ7	9	7	9	(7
<b>₽</b>											

## ステップ 6: シミュレーションおよびプロジェクトの終了

- 1. Simulate メニュー ⇒ End Simulation よりシミュレーションを終了します。メッセージが表示されますので "はい(Y)"を選択します。
- 2. Workspace ウィンドウの Project タブをアクティブにし、File メニュー ⇒ Close Project を選択します。メッセ ージが表示されますので "OK" を選択し、プロジェクトを終了します。

## <u>これで、演習5は終了です。</u>

## 演習6

ALTIMA

<目的>

- 非同期クリア & クロック・イネーブル付きフリップフロップを作成します。
- ファンクション・シミュレーションを実行し、動作を確認します。



<動作条件>

- ▷ クリア信号(aclr) が Low (0) のとき、フリップフロップは Low (0) を出力する(クリアされる)
- クリア信号(aclr) が High (1)、且つクロック・イネーブル信号 (clken) が High (1) のとき、出力 q は 入力 d を出力する

#### <u>ステップ1: プロジェクトの作成</u>

- 1. ModelSim-Altera が起動していない場合は、ModelSim-Altera を起動します。
- 2. File メニュー  $\Rightarrow$  New  $\Rightarrow$  Project を選択して、Create Project ダイアログ・ボックスを開きます。
- 3. Project Name 欄に、Project Name 欄に、ff と記述します。Project Location 欄に C:/lab/vhdl\_lab/lab6 (作 業ディレクトリ)を選択します。それ以外はデフォルト状態のままで OK ボタンをクリックします。

Add items to the Project ウィンドウが表示されますが、今回は使用しません。Close ボタンで閉じます。

## ステップ 2: デザインの作成とプロジェクトへの登録

- 1. File メニュー  $\Rightarrow$  New  $\Rightarrow$  Source  $\Rightarrow$  VHDL よりテキスト・エディタを開きます。
- 2. ソースコードを記述します。上記の回路仕様を参考にし、フリップフロップを完成させてください。
- 3. 記述終了後、File メニュー ⇒ Save As より、完成させた VHDL ファイルを ff.vhd として作業ディレクト リ内に保存します。その後、ff.vhd ファイルを閉じます。
- 4. Project メニュー ⇒ Add to Project ⇒ Existing File を選択し、ff.vhd と ff\_sim.vhd(テストベンチ)を指定 します。Project ウィンドウには、指定した 2 ファイルが登録されます。

#### <u>ステップ3: コンパイル</u>

 Compile メニュー ⇒ Compile Order にて、コンパイルの実行順序を設定します。Compile Order ウィンド ウの上段に ff.vhd、下段に ff\_sim.vhd を配置後、OK ボタンをクリックして順番を確定します。

## 🔥 ALTIMA

Compile メニュー ⇒ Compile All により、コンパイルを実行します。エラーが出る場合には、記述において文法的な間違いがあります。エラーメッセージをヒントに記述を修正してください。修正後は再びコンパイルを実行し、エラーが無くなるまで記述のデバッグを行います。

#### <u>ステップ 4: ロード</u>

- 1. Simulate メニュー ⇒ Start Simulation を選択し、Start Simulation ダイアログ・ボックスを起動します。
- 2. Design タブより work ディレクトリを展開し、ff\_sim の下位にある ff を指定後 OK ボタンをクリックしま す。
- 3. ロードが開始され問題なく終了すると、sim タブ(sim ウィンドウ)が追加され、関連ウィンドウが起動しま す。

#### ステップ 5: ファンクション・シミュレーションの実行

- sim ウィンドウ内の ff\_sim の下位にある ff を選択し、右クリック ⇒ Add to ⇒ Wave ⇒ All items in region を選択します。すると VHDL デザインの入出カピンが登録された状態で Wave ウィンドウが起動 します。
- Transcript ウィンドウに以下の実行コマンドを入力し、Enter キーでシミュレーション実行を開始します。
   run lus ※」は半角スペース
- Wave ウィンドウにシミュレーション結果が波形表示されます。作成した回路が正しく動作しているか確認しましょう。Wave メニュー ⇒ Zoom ⇒ …、またはツールバーのアイコンにて波形を明確に確認できます。 信号の Radix を変更する場合は、信号選択後に右クリック ⇒ Radix ⇒ 目的の表示を選択してください。

	💶 Wave - Default 🚃							
	<b>\$</b> -•	Msgs						
	✓ /ff_sim/clk	0	າທາທາກ	wwwww	wwwww	www.ww	wwwww	տուսուսու
l	<pre>/ff_sim/clken</pre>	0						
l	<pre>//f_sim/clr //cc_sim/clr</pre>	0						
	//r_sini/a /ff_sim/a	0					n	
l								

## ステップ 6: シミュレーションおよびプロジェクトの終了

- 1. Simulate メニュー ⇒ End Simulation よりシミュレーションを終了します。メッセージが表示されますので "はい(Y)"を選択します。
- 2. Workspace ウィンドウの Project タブをアクティブにし、File メニュー ⇒ Close Project を選択します。メッセ ージが表示されますので "OK" を選択し、プロジェクトを終了します。

## <u>これで、演習5は終了です。</u>

## 以上で本コースの演習はすべて終了です。お疲れ様でした。

## <u> 改版履歴</u>

Revision	年月	概要
1	2016年4月	· 初版(Web 対応)

#### 免責およびご利用上の注意

弊社より資料を入手されましたお客様におかれましては、下記の使用上の注意を一読いただいた上でご使用ください。

- 1. 本資料は非売品です。許可無く転売することや無断複製することを禁じます。
- 2. 本資料は予告なく変更することがあります。
- 本資料の作成には万全を期していますが、万一ご不明な点や誤り、記載漏れなどお気づきの点がありましたら、本資料を入手されました下記代理店までご一報いただければ幸いです。
   株式会社アルティマ ホームページ: http://www.altima.co.jp
   技術情報サイト EDISON: http://www.altima.jp/members/index.cfm
   株式会社エルセナ ホームページ: http://www.elsena.co.jp
   技術情報サイト ETS : http://www.elsena.co.jp/elspear/members/index.cfm

本資料で取り扱っている回路、技術、プログラムに関して運用した結果の影響については、責任を負いかねますのであらかじめご了承ください。

5. 本資料は製品を利用する際の補助的な資料です。製品をご使用になる際は、各メーカ発行の英語版の資料もあわせてご利用ください。