

独習アルテラ OpenCL 基礎演習 (Atlas-SoC ボード編)

ver.15



2016年7月 Rev.1

ELSENA,Inc.





独習アルテラ OpenCL 基礎演習 (Atlas-SoC ボード編)

<u>目次</u>

1. はじめに	3
1-1. 演習環境	
1-2. "Atlas-SoC" キット	
1-3. 付属ファイルの内容	5
2. デザイン・フロー	6
2-1. プログラミング・モデル	6
2-1-1. 一般的なプログラミング・モデル	6
2-1-2. アルテラ SDK for OpenCL での実装モデル	6
2-2. 開発フロー	7
3. 事前準備	8
3-1. Quartus Prime 開発ソフトウェアのインストール	
3-2. SoC EDS のインストール	
3-3. SDK for OpenCL のインストール	
3-4. v15.1 用 Atlas-SoC ボード・パッケージ(ボード・テンプレート)の追加	9
3-5. Windows 環境変数の確認・設定	9
3-6. 演習ファイルの抽出	
3-7. その他ツールのインストール	
3-8. Atlas-SoC と PC の接続	
3-9. PC 側ネットワークの設定	11
3-10. UART の接続確認と Linux の起動	
3-11. Ethernet の接続確認	
4. 演習	14
4-1. エンベデッド・コマンド・シェルの起動	
4-2. カーネルのコンパイル	
4-3. ホスト・アプリケーションのコンパイル	
4-4. ファイルの転送	
4-5. OpenCL の実行	
4-6. カーネルのコンパイル結果の確認	
改版履歴	

1. <u>はじめに</u>

この資料は、アルテラ SoC を利用して OpenCL[™] による FPGA ロジックの高位合成を行う方法を解説します。

アルテラが提供する SDK for OpenCL を使用した演習を体験することで、迅速かつ簡単にハードウェア・アクセ ラレータを構築する方法を理解することができます。演習では単純な配列加算の処理を OpenCL で実装します。

1-1. 演習環境

本演習では下記の開発環境を使用します。

項番	項目	内容
1	ホスト PC	Microsoft® Windows® 7 Professional sp1 (64 bit) 搭載の 64 bit マシン
2	アルテラ Quartus®	アルテラ SoC FPGA のハードウェアを開発するためのツールです。
	Prime 開発ソフトウェ ー	ソフトウェア開発に必要なハンドオフ・ファイルの生成も行います。
	P	本説明書では、Quartus Prime 開発ソフトウェア・ライト・エディション v15.1 を使用しています。
		■ アルテラ Quartus Prime 開発ソフトウェア・ライト・エディション v15.1
		http://dl.altera.com/15.1/?edition=lite
3	アルテラ SoC EDS	アルテラ SoC FPGA のソフトウェアを開発するためのツールです。
	$(ARM \otimes DS-5^{TM})$	ホスト・アプリケーション・ソフトウェアをコンパイルすることができます。
	Altera Edition)	本説明書では、アルテラ SoC EDS v15.1.1 を使用しています。
		■ アルテラ SoC エンベデッド・デザイン・スイート v15.1.1
		http://dl.altera.com/soceds/15.1.1/?edition=standard
4	アルテラ SDK for	アルテラ FPGA で OpenCL ソフトウェアを開発するためのツールです。
	OpenCL	OpenCL カーネルをコンパイル・実行することができます。
		本説明書では、アルテラ SDK for OpenCL v15.1 を使用しています。
		■ アルテラ SDK for OpenCL v15.1
		http://dl.altera.com/opencl/15.1/?edition=standard
5	Atlas-SoC ボード	本資料の説明でターゲット・ボードとして使用する、アルテラ Cyclone® V SoC を搭載し
		た Terasic Atlas-SoC ボードです。
		■ Atlas SoC ボード
		http://rocketboards.org/foswiki/view/Documentation/AtlasSoCDevelopmentPlatform
6	演習デザイン・ファイ ル	本資料の演習で使用する、Atlas-SoC 上で動作する、単純な配列加算の処理を OpenCL で実装します。
		ホスト・アプリケーションでは入力テーブルの確保と乱数の格納を行い、カーネル(FPGA ロジック)で演算された加算値が正しいかを比較します。
		付属ファイルの Lab フォルダ の下に、Altera-SoCFPGA-OpenCL-vectorAdd.tar.gz があります。
7	UART ターミナル・ソ フト	本演習では Tera Term を使用しています。 ダウンロードやインストールにつきまして は、別途、作成者または関連サイトの情報を参考にしてください。

【表 1-1.1】 この資料の説明で使用している主な開発環境



1-2. "Atlas-SoC" キット

このキットは、Cyclone V SoC デバイスが搭載されています。



【図 1-2.1】 "Atlas-SoC" キット

(1) Atlas-SoC ボードのディップ・スイッチ設定を確認します。



【図 1-2.2】SW10 設定

【表 1-2.1】SW10 設定表

ボード・リファレンス	信号名	内容	設定
SW10-1	MSEL 0		ON
SW10-2	MSEL 1	-	OFF
SW10-3	MSEL 2	これらのピンは、FPGA のコンフィギュレーション方 法の選択に使用します	ON
SW10-4	MSEL 3		OFF
SW10-5	MSEL 4		ON
SW10-6	N/A	N/A	N/A

🔥 ALTIMA

1-3. 付属ファイルの内容

独習アルテラ OpenCL 基礎演習 (Atlas-SoC ボード版)の付属ファイル Self-study_Altera_OpenCL_basic_v15.1 for Atlas-Soc.zip を解凍すると、以下のものが含まれています。

■ BSP_Atlas フォルダ (ボード・テンプレート)

board

c5soc_atlas

arm32 : ホスト・アプリケーション用ライブラリ等 c5soc_atlas_sharedonly : ボード・テンプレート・プロジェクト driver board_env.xml

share

- Lab フォルダ (演習ファイル)
 - Altera-SoCFPGA-OpenCL-vectorAdd.tar.gz
- SD_Image フォルダ (OpenCL 用 SD カード・イメージ)
 - atlas_opencl_sdimage_v1511_r3.tgz
- README.txt (アルテラ開発ツール 入手先 URL)

2. <u>デザイン・フロー</u>

アルテラの SDK for OpenCL は、エンベデッド ARM Cortex-A9 プロセッサ・コアをホストとして、ANSI C ベースの言語である OpenCL C で記述したソフトウェア関数(カーネル)を、FPGA デザイン(ハードウェア・アクセラレータ)に合成し、実行することができます。

この章では、OpenCL 開発における、ホストとカーネル、FPGA デザインへの合成などデザイン・フローを理解 することができます。

2-1. プログラミング・モデル

2-1-1. 一般的なプログラミング・モデル

OpenCL の一般的なプログラミング・モデルを下図に示します。 ホストでは、グローバル・メモリを確保し、プロセッサで処理するデータを準備し、プロセッサに制御を移します。 プロセッサはそのデータを処理した後、再びホストに制御を戻し、処理が完了したことを通知します。



【図 2-1-1.1】 一般的なプログラミング・モデル

2-1-2. アルテラ SDK for OpenCL での実装モデル

アルテラ SDK for OpenCL では、ホストを ARM Cortex-A9 プロセッサ・コアとして、グローバル・メモリを DDR3-SDRAM、プロセッサを FPGA ロジック、ローカル・メモリは FPGA 内の RAM ブロックに実装されます。





2-2. 開発フロー

まず、OpenCL プログラムを記述します。 OpenCL プログラムには FPGA に実装する部分を記述したカーネ ル・コードと、そのカーネルを制御するホスト・コードを記述します。

カーネル・コードは OpenCL C で記述しますので、そのままではプロセッサで動作させることはできません。最初にカーネル・コードを標準の C 言語で記述し、正しく動作することを確認した上で OpenCL C に変換することを 推奨します。

続いて、カーネル・コードを FPGA デザインに合成、コンパイルし、.aocx (Altera Offline Compiler eXecutable file)を生成し、ホスト(Linux)に転送します。

ホスト・コードも、PC のクロス・コンパイラ環境で実行可能なホスト・アプリケーションにコンパイルし、ホストに 転送します。

ホスト・アプリケーションを実行すると、生成された .aocx ファイルで FPGA をコンフィギュレーションし、カーネル・コードに記述された機能と同等の処理が FPGA ロジックを使用して実行されます。



【図 2-2.1】 アルテラ SDK for OpenCL で開発フロー

3. <u>事前準備</u>

この章では、演習の実施に必要な機材を確認し、ツールをインストールします。

なお、後述のツールを既にインストール済みの場合は、新たにインストールする必要はありませんので読み飛ばしてください。

- 3-1. Quartus Prime 開発ソフトウェアのインストール
 - (1) 今回は、アルテラのサイトから、Quartus Prime v15.1 (Windows 版)をダウンロードしインストールします。
 (有償ライセンスが必要なスタンダード・エディションでも、無償のライト・エディションでも構いません。)
 - Windows の環境変数を開いて、LM_LICENSE_FILE を追加し、OpenCL 評価用ライセンス・ファイル のパスとファイル名を設定します。

[注意事項]

✓ OpenCL 評価用ライセンス・ファイル は、別途お客様担当の代理店にお問い合わせください。

システムのプロパティ X コンピューター名 ハードウェア 詳細設定 システムの保護 リモート ほゆかあ X	環境変数 環境変数(U)
	変数 値 ALTERAOCLSDKR C¥altera¥15.0¥hid Path C¥altera¥15.0¥hid 目 OSYS_ROOTDIR C¥altera¥15.0¥quartus¥sopc_builder¥bin TFMP 纵ISFRPOFIL S¥AnData¥1cad¥Term
変数(値(V): C¥ALTERA_License¥00A0B0B94F7F_144602983 OK キャンセル システム環境変数(S) システム環境変数(S)	新規(N) 編集(E) 削除(D) システム環境変数(S) 変数 値
変数 値 ACPath C型Program Files (x86)単Lenovo¥Access Conn. COMMPath C型Program Files¥Lenovo¥Communications Ut ComSpec C型Windows¥System32%cmd.exe configsetront C型Windows¥ConfigSetRont * 新規(W). 編集(D 削除(L)	I.M.LICENSE FILE C:¥ALTERA_Licence¥00A0B0B94F7F_1446829 NUMBER_OF_PROC 4 OS OS Windows_NT Path C:¥Prneram Files¥Common Files¥Microsoft Sh 新規(W) 編集(D 新規(W) 正常本1110
OK ##>>世ル	OK をクリック OK キャンセル

【図 3-1.1】LM_LICENSE_FILE 環境変数の設定

3-2. SoC EDS のインストール

アルテラのサイトから、SoC EDS v15.1.1 (Windows 版)をダウンロードしインストールします。

3-3. SDK for OpenCL のインストール

アルテラのサイトから、SDK for OpenCL v15.1 (Windows 版)をダウンロードしインストールします。

3-4. v15.1 用 Atlas-SoC ボード・パッケージ(ボード・テンプレート)の追加

付属ファイルの "BSP_Atlas" フォルダ の中身(board と share)を、SDK for OpenCL v15.1 インストール・フォ ルダ(例: C:¥altera¥15.1¥hld)の下に上書きコピーしてください。



【図 3-4.1】Atlas-SoC ボード・パッケージ(ボード・テンプレート)の追加

これにより board フォルダの下に、c5soc_atlas フォルダが追加され、share¥models¥dm フォルダの下に、 5csema4u23c6n_dm.xml ファイルが追加されます。

3-5. Windows 環境変数の確認・設定

Windows の環境変数を開いて下記の変数を追加してください。

- (1) 以下の環境変数が存在し正しい値が設定されていることを確認します。
 - 変数: ALTERAOCLSDKROOT
 - 值: (例: C:¥altera¥15.1¥hld)
 - 変数: QUARTUS_ROOTDIR
 - 值: (例: C:¥altera¥15.1¥quartus)
- (2) 以下の環境変数を追加します。
 - 変数: AOCL_BOARD_PACKAGE_ROOT
 - 值: %ALTERAOCLSDKR00T%¥board¥c5soc_atlas
 - 変数: PATH
 - 值: %QUARTUS_ROOTDIR%¥bin64;%ALTERAOCLSDKROOT%¥windows64¥bin

3-6. 演習ファイルの抽出

付属ファイル内の "Lab" フォルダ の中にある Altera-SoCFPGA-OpenCL-vectorAdd.tar.gz ファイルを PC の任意のワーク・フォルダ (この例では、C:¥Work) にコピーして解凍します。

解凍は、エンベデッド・コマンド・シェルからコマンドラインで実行できます。

Windows エクスプローラで、SoC EDS のインストール・フォルダ(この例では、C:¥altera¥15.1¥embedded)を開き、 その下にある Embedded_Command_Shell.bat をダブルクリックしてエンベデッド・コマンド・シェルを起動し、以下 のコマンドで Altera-SoCFPGA-OpenCL-vectorAdd.tar.gz ファイルを解凍します。

\$ cd	″C:¥Wo	rk″ 🚽
\$ tar	-xzf	Altera-SoCFPGA-OpenCL-vectorAdd.tar.gz 🛛

[注意事項]

✓ コピー&解凍先のフォルダ・パス名にスペースや日本語(全角文字)が無いよう注意してください。

3-7. その他ツールのインストール

UART ターミナル・ソフトをインストールします。本資料では Tera Term を使用していますが、機能的に同等で あれば別のツールを使っても構いません。

なお、ツールのダウンロードやインストールにつきましては、別途、作成者または関連サイトの情報を参考にしてください。

3-8. Atlas-SoC と PC の接続

PC と Atlas-SoC を接続し、microSD カードをカード・スロットに挿入し、電源を入れてください。





[注意事項]

✓ 付属ファイル内の "SD_Image" フォルダ の下に

atlas_opencl_sdimage_v1511_r3.tgz

ファイルがありますので、これを エンベデッド・コマンド・シェル から tar -xzf コマンドで解凍し、フリー・ソ フト Win32DiskImager を使用して microSD カードに書き込みます。

3-9. PC 側ネットワークの設定

Atlas-SoC と PC を Ethernet で接続するための設定を行います。

- (1) Windows のスタート・メニューから「コントロールパネル」を選択し、「ネットワークと共有センター」をクリ ックします。
- (2)「**アダプタの設定の変更**」を開いて、「ローカルエリア接続」を選択・右クリックして「プロパティ」を選択します。
- (3) プロパティ・ダイアログが表示されたら、「インターネットプロトコルバージョン 4 (TCP/IPv4) 」を選択し、 「プロパティ」ボタンをクリックします。
- (4) 「次の IP アドレスを使う」にチェックを入れ、"IP アドレス"と"サブネット マスク"に任意のアドレスを入力します。この例では、IP アドレス: 192.168.1.31、サブネットマスク: 255.255.255.0 に設定します。

インターネット プロトコル バージョン 4 (1	TCP/IPv4)のプロパティ 🛛 💽					
全般						
ネットワークでこの機能がサポートされている場合は、IP 設定を自動的に取得することがで きます。サポートされていない場合は、ネットワーク管理者に適切な IP 設定を問い合わせ てください。						
◎ IP アドレスを自動的に取得する(O)						
③次の IP アドレスを使う(S):						
IP アドレス(I):	192 . 168 . 1 . 31					
サブネット マスク(U):	255 . 255 . 255 . 0					
デフォルト ゲートウェイ(D):	· · ·					
◯ DNS サーバーのアドレスを自動的に取得	する(B)					
 ③ 次の DNS サーバーのアドレスを使う(E): 						
優先 DNS サーバー(P):	· · · .					
代替 DNS サーバー(A):						
🔲 終了時に設定を検証する(L)	詳細設定(V)					
	OK *+>>セル					

【図 3-9.1】 PC 側ネットワークの設定

3-10.UART の接続確認と Linux の起動

- (1) Atlas-SoC と PC が UART ケーブルで接続されていることを確認します。
- (2) Atlas-SoC に接続されている COM ポートを選択し「ボーレート」を "115200" に設定します。

Tera Term: シリアルポー	ート 設定	×
ポート(P):	COM8 -	ок
ボー•レート(B):	115200 🔹	
データ(D):	8 bit 🔹	キャンセル
バリティ(A):	none 🔹	
ストップ(S):	1 bit 🔹	ヘルプ(H)
フロー制御(F):	none 🔹	
送信遅延	115200 を選択	
₹لار ا	リ/字(c) 0 ミ	ジ秒/行(L)

【図 3-10.1】Tera Term でのシリアル・ポート設定

(3) Atlas-SoC の WARM RESET ボタンを押下します。



【図 3-10.2】Atlas-SoC の WARM RESET ボタンを押下

- (4) Linux コンソールに Linux の起動ログが表示されます。ログインするには、login プロンプトに root と 入力します。
- (5) passwd J と入力して root のパスワードを設定します。
- (6) 新規パスワードが要求された場合は、この例では altera. 123 を入力した後、もう一度 altera. 123 を入力します(入力の際、パスワードは表示されません)。

COM6:115200baud - Tera Term VT	
Elle Edit Setup Control Window Help	
[OK] Started Update UTMP about System Runlevel Changes. [OK] Started WPA supplicant.	*
The Angstrom Distribution cyclone5 ttyS0	
Angstrom v2014.12 - Kernel 3.10.31-Itsi-05151-gc8b20a4-dirty	
cyclone5 login: libphy: stmmac-1:01 - Link is Up - 1000/Full root でログイン root@cyclone5: # passwod と入力 Enter new UNIX password: パスワードを入力 Retype new UNIX password: パスワードを入力 passwd: password updated successfully root@cyclone5: # ■	
Toolecyclones. #	-

【図 3-10.3】 ログインとパスワードの設定



3-11.Ethernet の接続確認

Atlas-SoC と PC が Ethernet ケーブルで接続されていることを確認します。

- (1) Linux コンソールから 任意の IP アドレスを設定します。この例では、192.168.1.30 を設定しますので、以下のように入力してください。
 - # ifconfig eth0 192.168.1.30 🚽
- (2) 続いて、Linux および PC 双方から Ping を送り、正しく接続できていることを確認します。その際、どちら かの Ping が通らない場合は Windows のファイア・ウォール設定を解除するなどしてください。



【図 3-11.1】 Ethernet の接続確認

4. 演習

演習には、前述「3-6. 演習ファイルの抽出」で解凍した Altera-SoCFPGA-OpenCL-vectorAdd を使用します。
 単純な配列加算の処理を OpenCL で実装します。

ホスト・アプリケーションでは入力テーブルの確保と乱数の格納を行い、カーネル(FPGA ロジック)で演算され た加算値が正しいかを比較します。

4-1. エンベデッド・コマンド・シェルの起動

後ほど実行する、「カーネルのコンパイル」と「ホスト・アプリケーションのコンパイル」は、このエンベデッド・コマンド・シェルからコマンドラインで実行します。

- Windows エクスプローラで、SoC EDS のインストール・フォルダ(この例では、C:¥altera¥15.1¥embedded)を 開き、その下にある Embedded_Command_Shell.bat をダブルクリックしてエンベデッド・コマンド・シェルを 起動します。
- (2) エンベデッド・コマンド・シェルから 以下を実行します。

\$ export ALTERAOCLSDKROOT="C:¥altera¥15.1¥hld" \$ export PATH=\$ {ALTERAOCLSDKROOT} /bin:\$PATH \$ export AOCL_BOARD_PACKAGE_ROOT=\$ {ALTERAOCLSDKROOT} /board/c5soc_atlas

(3) 現在認識されているボードを確認します。

\$ cd "C:¥altera¥15.1¥hld¥bin" 🚽 \$ aoc —list-boards ↓ Board list: c5soc atlas sharedonly

🔥 ALTIMA

4-2. カーネルのコンパイル

カーネルのコンパイルでは、バック・グランドで Quartus Prime 開発ソフトウェアが実行されますので、Quartus Prime 開発ソフトウェアが起動中であれば終了させてください。

(1) エンベデッド・コマンド・シェルから 以下のコマンドを入力します。

<pre>\$ cd "C:\Work\Altera-SoCFPGA-OpenCL-vectorAdd\device" \$ aoc -v -report vectorAdd.cl -o vectorAdd.aocx -board c5soc_atlas_sharedonly aoc: Environment checks are completed successfully. You are now compiling the full flow!! aoc: Selected target board c5soc_atlas_sharedonly aoc: Running OpenCL parser c:/Work/Altera-SoCFPGA-OpenCL-vectorAdd/device/vectorAdd.cl:23:47: warning: decl aring kernel argument with no 'restrict' may lead to low kernel performance kernel void vectorAdd(_global const float *x,</pre>					
c:/Work/Altera-SoCFPGA-OpenCL-vectorA aring kernel argument with no 'restri global cons	c:/Work/Altera-SoCFPGA-OpenCL-vectorAdd/device/vectorAdd.cl:24:47: warning: decl aring kernel argument with no 'restrict' may lead to low kernel performance global const float *y,				
2 warnings generated. aoc: OpenCL parser completed successfully. aoc: Compiling aoc: Linking with IP library					
; Estimated Resource Usage Summary					
; Resource	+ Usage ;				
: Logic utilization : 65% :					
; Dedicated logic registers ; 27% ;					
; Memory blocks ; 24% ;					
; DSP blocks ; 0% ;					
aoc: First stage compilation completed successfully. aoc: Hardware generation completed successfully.					

- (2) ご使用の PC スペックにもよりますが、カーネルのコンパイルには 13 分程度かかります。エンベデッド・コ マンド・シェルのプロンプトが表示されるまで待ちます。
- (3) コンパイルが成功すると、以下のように vectorAdd.aocx ファイルが生成されます。



4-3. ホスト・アプリケーションのコンパイル

(1) エンベデッド・コマンド・シェルから 以下のコマンドを入力します。

\$ cd "C:\Work\Altera-SoCFPGA-OpenCL-vectorAdd" \$ make all a arm-linux-gnueabihf-g++ host/src/main.cpp common/src/AOCL_Utils.cpp -o vector_a dd -IC:/altera/15.1/hld/host/include -Icommon/inc -LC:\altera\15.1\altera\15

(2) コンパイルが成功すると、以下のように vector_add ファイルが生成されます。

```
$ Is -1 2
合計 52
drwx-----+ 1 ???????? mkpasswd 0 5月 28 18:03 common
drwx-----+ 1 ??????? mkpasswd 0 5月 28 18:34 device
drwx-----+ 1 ??????? mkpasswd 0 5月 28 18:03 host
-rwx-----+ 1 ??????? mkpasswd 1406 12月 10 19:59 Makefile
------+ 1 elsfae mkpasswd 43296 5月 28 18:37 vector_add
```

4-4. ファイルの転送

ホスト・アプリケーションとコンパイル済みのカーネルを Atlas-SoC の microSD カードに転送します。

(1) SCP(セキュア・コピー)を使用してファイルを転送することができます。

SCP を経由してホスト PC から Atlas-SoC の microSD カードにファイルを転送するには、ホスト PC から scp <source_filename> root@<board_ip_address>:<target_filename> コマンドを使用します

(2) エンベデッド・コマンド・シェルから 次のように入力し vector_add ホスト・アプリケーションを Atlas-SoC に 転送します。接続を続けるかどうかを聞かれた場合は、yes を入力します。この例ではパスワードは altera. 123 を使用します。

\$ scp vector_add root@192.168.1.30:/home/root 🚽				
Could not create directory '/home/11149/.ssh'.				
The authenticity of host '192.168.1.30 (192.168.1.30)' can't be established.				
ECDSA key fingerprint is SHA256:SZhkriFhnzX/arvfUwpBVVAG57AUnFXGLCSwRCnHAOE.				
Are you sure you want to continue connecting (yes/no)? yes J 🛛 🗾 🗾				
Failed to add the host to the list of known hosts (/home/11149/.ssh/known_hosts)				
Password: パスワードを入力				
vector_add 100% 42KB 42.3KB/s 00:00				

(3) 次のように入力し vectorAdd.aocx ファイルを Atlas-SoC に転送します。再度、パスワード altera. 123 を使用します。

\$ scp ./device/vectorAdd.aocx root@192.168.1.30:/home/root 🚽					
Could not create directory '/home/11149/.ssh'.					
The authenticity of host '192.168.1.30 (192.168.1.30)' can't be established.					
ECDSA key fingerprint is SHA256:SZhkriFhnzX/arvfUwpBVVAG57AUnFXGLCSwRCnHAOE.					
Are you sure you want to continue connecting (yes/no)? yes 🎣 🛛 🗾 yes を入力					
Failed to add the host to the list of known hosts (/home/11149/.ssh/known_hosts)					
Password: パスワードを入力					
vectorAdd. aocx 100% 1663KB 1.6MB/s 00:00					

これで、Atlas-SoC の microSD カードにファイルが転送できました。

4-5. OpenCL の実行

転送されたカーネルで FPGA をコンフィギュレーションし、ホスト・アプリケーションを実行することで OpenCL の動作を確認することができます。

(1) Linux コンソールから 以下のように入力し、ターゲットの Linux 上で ドライバ・モジュールをロードしま す。

source ./init_opencl.sh ↓

(2) 次のように入力して、ボードが正常に動作していることを確認します。

```
# aocl diagnose a
aocl diagnose: Running diagnostic from /home/root/aocl-rte-15.1.0-1.arm32/board/c5soc/arm32/bin
Verified that the kernel mode driver is installed on the host machine.
Using platform: Altera SDK for OpenCL
Board vendor name: Altera Corporation
Board name: DEOnanoSoc : Cyclone V SoC Development Kit
Buffer read/write test passed.
DIAGNOSTIC_PASSED
```

(3) 次のように入力して、転送したファイルを実行可能にします。

```
# chmod 777 vector_add 
# chmod 777 vectorAdd.aocx
```

(4) ターゲットの Linux 上で OpenCL カーネルプログラムのプログラミングを行います。リコンフィギュレーショ ンが成功すると、LED の点滅パターンが変化します。

```
# aocl program /dev/acl0 vectorAdd.aocx accl program: Running reprogram from /home/root/aocl-rte-15.1.0-1.arm32/board/c5soc/arm32/bin Reprogramming was successful!
```

(5) ターゲットの Linux 上で ホスト(ARM)プログラムを実行します。

```
# ./vector_add I
Initializing OpenCL
Platform: Altera SDK for OpenCL
Using 1 device(s)
c5soc_atlas_sharedonly : Cyclone V SoC Development Kit
Using AOCX: vectorAdd.aocx
Reprogramming device with handle 1
Launching for device 0 (1000000 elements)
Time: 318.877 ms
Kernel time (device 0): 177.687 ms
Verification: PASS
```

4-6. カーネルのコンパイル結果の確認

コンパイルされたカーネルのデザインを確認します。

(1) Quartus Prime 開発ソフトウェアを起動して、File メニュー ⇒ Open Project から、

"C:¥Work¥Altera-SoCFPGA-OpenCL-vectorAdd¥device¥vectorAdd¥top.qpf"を開きます。

				Open Project			×
				😋 🖉 🖉 « device 🕨 vectorAdd 🕨		✓ 4 vectorAdd0	D検索 👂
) (Quartus Prime Lite Edi	ition		整理 ▼ 新しいフォルダー			i≕ - - •
File	Edit View Project	Assignments	Processing	Altera-SoCFPGA-OpenCL-vectorAdd	^ 名前 ^	更新日時	種類
	No	01111) common	🔒 db	2016/03/31 9:11	ファイル フォル
	New	Ctri+N		device	hps_isw_handoff	2016/03/31 9:10	ファイル フォル…
~	Open	Ctrl+O	×	🍌 vectorAdd	🌗 iface	2016/03/31 8:52	ファイル フォル
	Close	Ctrl+F4) host	🍌 incremental_db	2016/03/31 8:58	ファイル フォル…
				EmbOpenCL_Lab	scripts	2016/03/31 8:52	ファイル フォル…
-	New Project Wizard			\mu ユーザー	system	2016/03/31 8:55	ファイル フォル…
X	Open Project	Ctrl+J		👝 HDPC-UT (D:)	E top.qpf	2016/03/31 8:52	QPF ファイル
	Save Project			😡 PUBLIC (¥¥file01) (M:)			
	Close Project			B Lenovo_Recovery (Q:)		m	۰.
				ファイル名(N): top.qpf		✓ Quartus Prim	ne Project File (* 👻
						開<(0)	キャンセル

【図 4-6.1】Quartus Prime 開発ソフトウェアを起動して、File メニュー ⇒ Open Project

(2) Processing メニュー ⇒ Compile Report を選択します。

			\	Compilation Rep	ort - to	ip 🗵	
			Table of Contents	(₽₽	Flow Summary	
			Flow Summary			Flow Status	Successful - Thu Mar 31 01:10:43 2016
Quartus Brima Lita Edition _ Cu/War	dd/douioo/upotorⅆ	d, # Clobal Sattiaga		Quartus Prime Version	15.1.0 Build 185 10/21/2015 SJ Lite Edition		
Quartus Prime Lite Edition - C./ Wor	ad/device/vectorAdc			Revision Name	top		
File Edit View Project Assignments	Proc	cessing Tools Window Help		it Gibbal Settings		Top-level Entity Name	top
	STOP	Stop Processing	Ctrl+Shift+C	ne ry		Family	Cyclone V
	_					Device	5CSEMA4U23C6
Project Navigator 🔥 Hierarchy 🔹 🗐		Start Compilation	Ctrl+L			Timing Models	Final
Entity/Instance Al Ms need	5	Analyze Current File		tesis		Logic utilization (in ALMs)	3,687 / 15,880 (23 %)
	-	Start				Total registers	7301
Cyclone V: 5CSEMA4U23C6		Start	· · ·			Total pins	121 / 314 (39 %)
▷ 100 top 100 3686.5 (1.0)		Update Memory Initialization File				Total virtual pins	0
	\mathbf{r}	Compilation Report	Ctrl+R			Total block memory bits	120,112/2,764,800 (4%)
	-	Dynamic Synthesis Report				Total PLLs	2/5(40%)
	· ·			d Messages		Total DLLs	1/4(25%)
			Assembler			Total DSP Blocks	0/84(0%)
			🕨 🕨 TimeQuest Timi	ng Analyzer		Total HSSI RX PCSs	0
						Total HSSI PMA RX Deserializers	0
						Total HSSI TX PCSs	0
						Total HSSI PMA TX Serializers	0



<u> 改版履歴</u>

Revision	年月	概要
1	2016年7月	初版

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
- 4. 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。