



インテル® FPGA SDK for OpenCL™

スタート・ガイド

インテル® Quartus® Prime 開発デザインスイートの更新情報: **17.1**



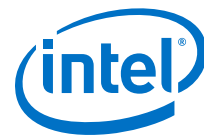
UG-OCL001 | 2017.12.08

最新版をウェブからダウンロード: [PDF](#) | [HTML](#)



目次

1. インテル® FPGA SDK for OpenCL™ スタート・ガイド	4
1.1. インテル FPGA SDK for OpenCL の使用条件.....	5
1.2. インテル FPGA SDK for OpenCL の内容.....	6
1.3. インテル FPGA SDK for OpenCL セットアップ・プロセスの概要.....	7
2. Windows 向け インテル FPGA SDK for OpenCL スタート・ガイド	10
2.1. インテル FPGA SDK for OpenCL のダウンロード	10
2.2. インテル FPGA SDK for OpenCL のインストール.....	11
2.3. インテル FPGA SDK for OpenCL ユーザー環境変数の設定	12
2.4. ソフトウェア・インストールの確認.....	13
2.5. FPGA ボード のインストール	14
2.6. エミュレーションによるホスト・ランタイムの機能性の検証	15
2.6.1. OpenCL デザイン例のダウンロード	16
2.6.2. エミュレーション用カーネルのコンパイル	16
2.6.3. ホスト・アプリケーションのビルド	16
2.6.4. OpenCL カーネルエミュレート.....	17
2.7. OpenCL カーネルの FPGA ハードウェア・コンフィグレーション・ファイルの作成	18
2.8. FPGA のハードウェア・イメージの更新.....	20
2.8.1. FPGA ボードのデバイス名の照会	20
2.8.2. FPGA のフラッシュメモリーのプログラミング	21
2.9. FPGA での OpenCL カーネルの実行	22
2.9.1. ホスト・アプリケーションの実行	22
2.9.2. 正常なカーネル実行からの出力.....	23
2.10. ソフトウェアのアンインストール	24
2.11. FPGA ボードのアンインストール	24
3. Linux 向け インテル FPGA SDK for OpenCL スタート・ガイド	25
3.1. インテル FPGA SDK for OpenCL のダウンロード	25
3.2. インテル FPGA SDK for OpenCL のインストール	26
3.3. インテル FPGA SDK for OpenCL ユーザー環境変数の設定	27
3.4. ソフトウェア・インストールの確認.....	28
3.5. FPGA ボードのインストール	28
3.6. エミュレーションによるホスト・ランタイムの機能性の検証	29
3.6.1. OpenCL デザイン例のダウンロード	30
3.6.2. エミュレーション用カーネルのコンパイル	30
3.6.3. ホスト・アプリケーションのビルド	31
3.6.4. OpenCL カーネルのエミュレーション.....	31
3.7. OpenCL カーネルの FPGA ハードウェア・コンフィグレーション・ファイルの作成	32
3.8. FPGA のハードウェア・イメージの更新.....	33
3.8.1. FPGA ボードのデバイス名の照会	34
3.8.2. FPGA のフラッシュメモリーのプログラミング	34
3.9. FPGA での OpenCL カーネルの実行	35
3.9.1. ホスト・アプリケーションの実行	36
3.9.2. 正常なカーネル実行からの出力.....	36
3.10. ソフトウェアのアンインストール	37



3.11. FPGA ボードのアンインストール	37
A. 改訂履歴.....	38



1. インテル® FPGA SDK for OpenCL™ スタート・ガイド

インテル® FPGA SDK for OpenCL™ スタート・ガイドでは、インテル FPGA Software Development Kit (SDK) for OpenCL をインストールする手順について説明します。(1) この資料には、インテル FPGA SDK for OpenCL を使用してサンプル OpenCL (2)アプリケーションをコンパイルする方法についての説明が含まれています。

OpenCL は、ヘテロジニアス並列デバイスのプログラミング向け C ベースのオープン・スタンダードです。OpenCL Specification version 1.0 について詳しくは [OpenCL 1.0 Reference Pages](#) を参照してください。OpenCL API (Application Programming Interface) およびプログラミング言語について詳しくは [OpenCL Specification version 1.0](#) を参照してください。

インテル FPGA SDK for OpenCL は、インテル FPGA をターゲットとする OpenCL アプリケーションをビルドし動作させるためのコンパイラとツールを提供します。また、インテル FPGA SDK for OpenCL は OpenCL Specification version 1.0 の Embedded Profile をサポートしています。

注意:

- インテル FPGA SDK for OpenCL のカーネル展開機能を必要とする場合、インテル FPGA Runtime Environment (RTE) for OpenCL をダウンロードしてインストールします。詳細は、[インテル FPGA RTE for OpenCL スタート・ガイド](#)を参照してください。

SDK と RTE を同一のホスト・システムにインストールすることはできません。SDK には既に RTE が含まれています。

- インテル FPGA SDK for OpenCL Intel Arria 10 GX FPGA 開発キットについては、[アプリケーション・ノート](#)を参照してください。詳細については、[インテル FPGA SDK for OpenCL 用 Intel Arria 10 GX FPGA 開発キットの設定](#)を参照してください。
- インテル FPGA SDK for OpenCL と Cyclone® V SoC 開発キットを併せて使用する必要がある場合には、[インテル FPGA SDK for OpenCL Cyclone V SoC Getting Started Guide](#) を参照してください。

関連情報

- [OpenCL のリファレンス・ページ](#)
- [OpenCL の仕様バージョン 1.0](#)
- [Intel FPGA RTE for OpenCL のスタート・ガイド](#)
- [インテル FPGA SDK for OpenCL Cyclone V SoC のスタート・ガイド](#)
- [インテル FPGA SDK for OpenCL 用 Intel Arria 10 GX FPGA 開発キットの設定](#)

(1) インテル FPGA SDK for OpenCL は Khronos 社公開のスペックに基づいており、Khronos 社のコンFORMANCE テストに合格しています。現在のコンFORMANCE 基準については www.khronos.org/conformance を参照してください。

(2) OpenCL™



1.1. インテル FPGA SDK for OpenCL の使用条件

インテル FPGA SDK for OpenCL のインストールおよびインテル FPGA Preferred Board for OpenCL で推奨されるアクセラレーター・ボード向け OpenCL アプリケーションの作成にあたり、使用するシステムがハードウェア、ターゲット・プラットフォーム、およびソフトウェアの要件を満たしている必要があります。

ハードウェア条件

アクセラレーター・ボード要件

- インテル のリファレンス・プラットフォーム、または インテル 推奨ボードベンダーのカスタム・プラットフォームを取得する。
詳しくは、インテル FPGA ウェブサイトで [インテル FPGA SDK for OpenCL FPGA Platforms](#) のプラットフォームのページを参照してください。

開発システム要件

- 必要なパッケージおよびドライバーをインストールするために、開発システムでの管理者権限を有している必要がある。
- ソフトウェアをインストールするために、開発システムが少なくとも 85 ギガバイト (GB) の空き領域を有している必要がある。
- 開発システムが少なくとも 24 GB の RAM を有している必要がある。

ヒント: 推奨されるシステム・ストレージ・サイズについては、ボードベンダーの資料を参照してください。

- PCI Express* (PCIe*)アクセラレータ・ボードでは、ホストマシンのマザーボードが少なくともボードと同じ幅(つまり PCIe レーンと同じ数)の使用可能な PCIe ポート・スロットを有している必要がある。

ホスト・システムは、以下のサポートされるオペレーティング・システムのいずれかで動作している必要があります。

- サポート対象の Windows および Linux オペレーティング・システムのリストについては、インテル ウェブサイトでの [オペレーティング・システムのサポートページ](#) を参照してください。

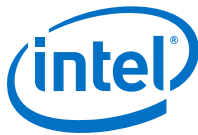
ソフトウェア要件

以下に示す、インテル FPGA SDK for OpenCL および インテル Quartus® Prime ソフトウェアと互換する C コンパイラーまたはソフトウェア開発環境のいずれかを使用してホスト・アプリケーションを開発します。

- Windows システムでは、Microsoft Visual Studio バージョン 2010 Professional を使用する
- Linux システムでは、GCC に含まれる C コンパイラーを使用する

Linux システムではバージョン 5 以降の Perl コマンドが必要です。システム環境変数設定で、*PATH* システムに Perl コマンドへのパスを含めます。

Intel Code Builder を含む インテル FPGA SDK for OpenCL パッケージには、Intel Code Builder は Java SE バージョン 1.8.71 以降が必要です。



関連情報

インテル FPGA SDK for OpenCL FPGA のプラットフォーム

1.2. インテル FPGA SDK for OpenCL の内容

インテル FPGA SDK for OpenCL にはロジック・コンポーネント、ドライバー、およびに特化したライブラリとファイルが付属しています。

ロジック・コンポーネント

- インテル FPGA SDK for OpenCL オフライン・コンパイラー は、OpenCL デバイス・コードをシステムが インテル FPGA にロードするハードウェア・コンフィグレーション・ファイルに変換する
- インテル FPGA SDK for OpenCL ユーティリティには診断テストの実行といった上位レベル・タスクを行うために呼び出す命令のセットが含まれている
- ホスト・ランタイム、これは OpenCL ホスト・アプリケーションに OpenCL ホスト・プラットフォーム API およびランタイム API を提供する

ホスト・ランタイムは、OpenCL API、ハードウェア抽象化、およびヘルパー・ライブラリを提供するライブラリで構成されています。

ドライバ、ライブラリとファイル

ソフトウェアのインストール・プロセスにより、ご使用のディレクトリに インテル FPGA SDK for OpenCL がインストールされます。*INTELFPGAOCSDKROOT* 環境変数は、このインストール・ディレクトリへのパスを参照します。

表 1. インテル FPGA SDK for OpenCL インストール・ディレクトリの選択コンテンツ

Windows フォルダー	Linux ディレクトリ	概要
bin	bin	SDK のユーザー・コマンドです。 <i>PATH</i> 環境変数設定にこのディレクトリを含めます。
board	board	ソフトウェアで使用可能な インテル FPGA SDK for OpenCL カスタム・プラットフォーム・ツールキットおよびリファレンス・プラットフォームです。 <ul style="list-style-type: none"> • カスタム・プラットフォーム・ツールキットへのパスを以下に示します。 <i>ALTERAOCLSDKROOT/board/custom_platform_toolkit</i> • 以下に s5_ref Reference Platform へのパスを示します。 <i>ALTERAOCLSDKROOT/board/s5_ref</i>
ip	ip	デバイス・カーネルのコンパイルに使用する IP (Intellectual Property) コアです。
host	host	ホスト・アプリケーションのコンパイルおよび動作に必要なファイルです。
host\include	host/include	ホスト・アプリケーションのコンパイルおよびリンクに必要な OpenCL Specification version 1.0 ヘッダ・ファイルおよびソフトウェア・インタフェース・ファイルです。 host/include/CL サブディレクトリには C++ヘッダ・ファイル <i>cl.hpp</i> も含まれています。このファイルは OpenCL version 1.1 C++ラッパー API を提供します。これらの C++バインディングにより、C++ホスト・プログラムがネイティブ C++クラスおよびメソッドを使用する OpenCL ランタイム API にアクセスできます。 重要: OpenCL version 1.1 C++バインディングは OpenCL Specification version 1.0 および 1.1 と互換します。 開発環境でこのパスを include ファイル検索パスに含めます。
host\nwindows64\lib	host/linux64/lib	OpenCL プラットフォームおよびランタイム API を提供する、OpenCL ホスト・ランタイム・ライブラリです。これらのライブラリはホスト・アプリケーションのリンクに必要です。

continued...



Windows フォルダー	Linux ディレクトリ	概要
		Linux で OpenCL アプリケーションを動作させるには、このディレクトリを <code>LD_LIBRARY_PATH</code> 環境変数設定に含めます。
host \windows64\bin	host/ linux64/bin	該当する場合に、ホスト・アプリケーションの動作に必要なランタイム・コマンドおよびライブラリです。64 ビット Windows システムでは、このディレクトリを <code>PATH</code> 環境変数設定に含めます。Windows システムでは、このフォルダーにランタイム・ライブラリが入ります。Linux システムでは、このディレクトリに <code>aocl</code> ユーティリティ・コマンド向けの、プラットフォームに特化したバイナリが入ります。
share	share	アーキテクチャに依存しないサポート・ファイルです。

OpenCL アプリケーション例

OpenCL デザイン例のページから OpenCL アプリケーション例をダウンロードできます。

関連情報

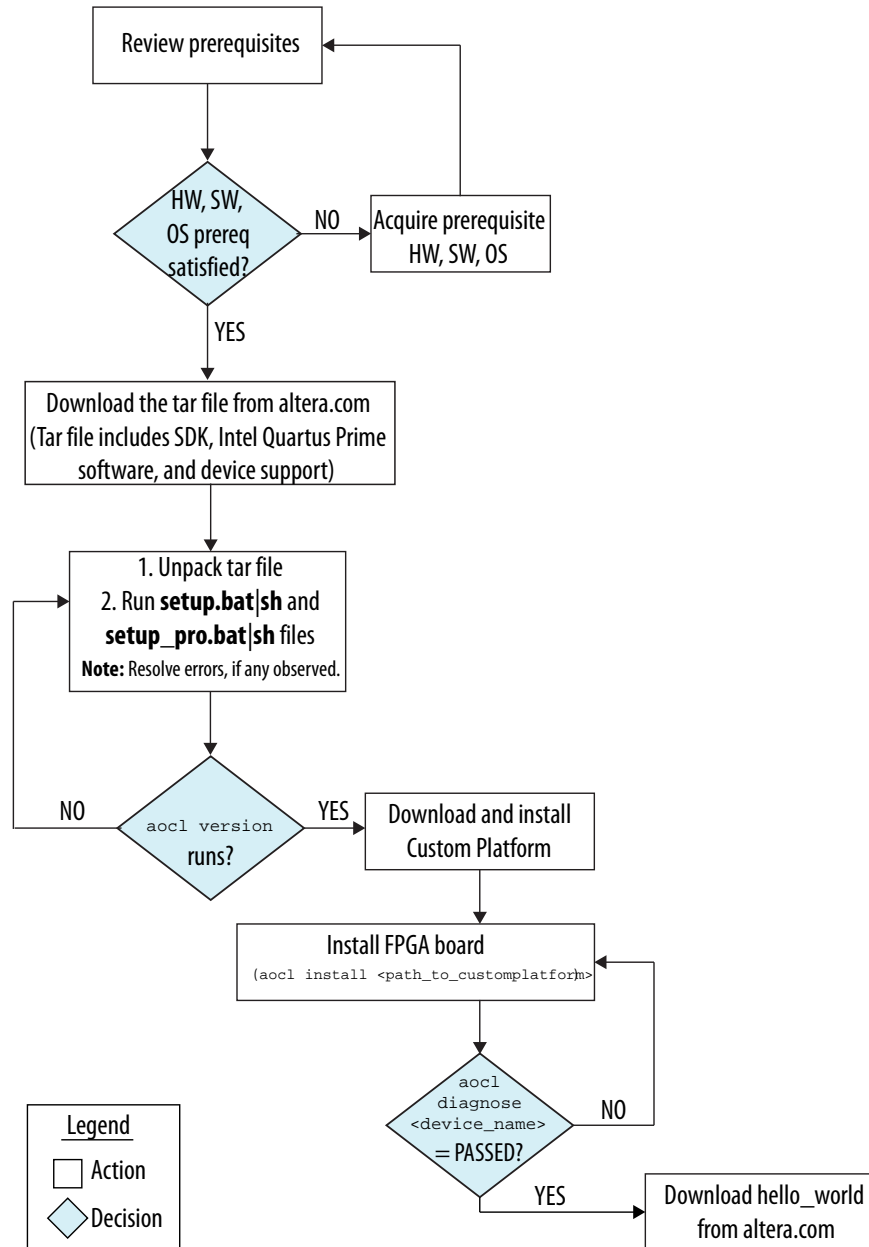
[OpenCL デザイン例の生成](#)

1.3. インテル FPGA SDK for OpenCL セットアップ・プロセスの概要

この *インテル FPGA SDK for OpenCL スタート・ガイド* では、インテル FPGA SDK for OpenCL のインストールおよび FPGA プログラミングの手順を紹介します。

以下の図は、必要なソフトウェアの設定および FPGA ボードのインストール手順をまとめています。

図 -1: インテル FPGA SDK for OpenCL インストール手順の概要



注意: aocl diagnose ユーティリティを実装した後に起こりうるエラーについては、[diagnose ユーティリティの実行後に発生する可能性のあるエラー](#)のセクションを参照してください。

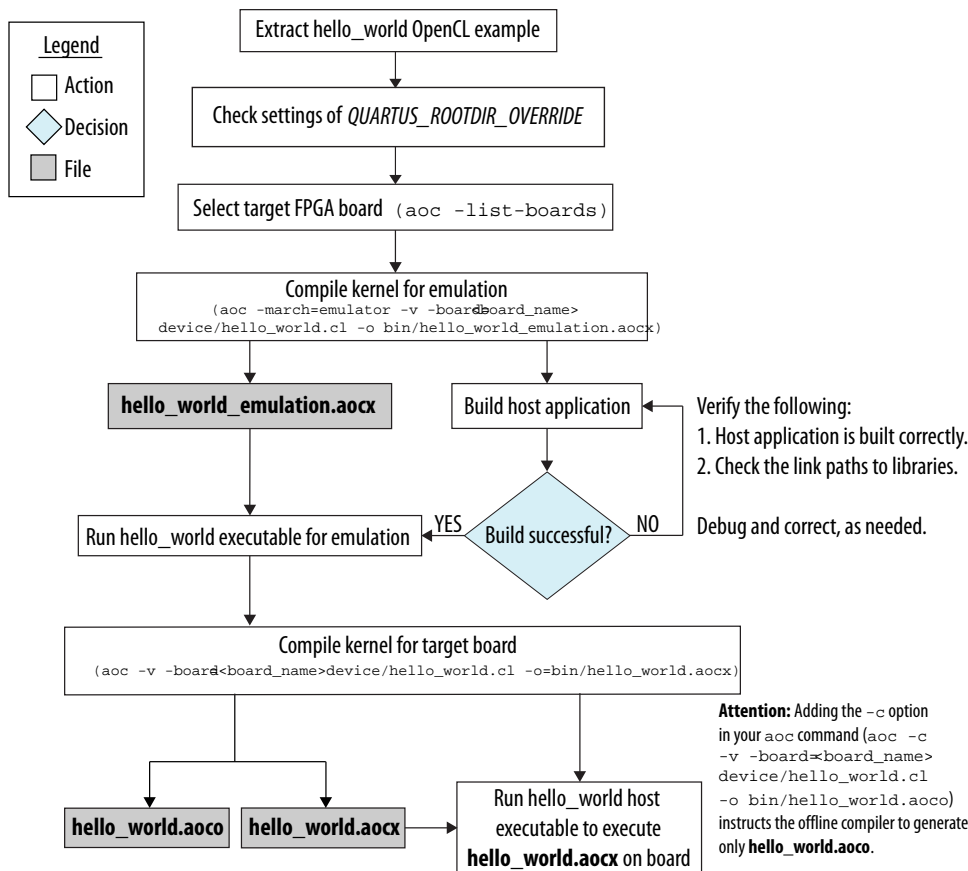
ソフトウェアとハードウェアの初期設定完了後に、*hello_world* OpenCL デザイン例からハードウェア・イメージを作成できます。

注意: FPGA をハードウェア・イメージでプログラミングする前に、FPGA が最新の インテル FPGA SDK for OpenCL を用いて作成されたイメージを有することを確認してください。詳しくは、*FPGA のハードウェア・イメージの更新*を参照してください。



以下の図は、FPGA をプログラミングするための手順をまとめています。

図 -2: FPGA プログラミングの概要



関連情報

FPGA のハードウェア・イメージの更新 (20 ページ)



2. Windows 向け インテル FPGA SDK for OpenCL スタート・ガイド

インテル FPGA SDK for OpenCL セットアップ・プロセスには、ソフトウェアのダウンロードとインストール、FPGA ボードのインストール、ならびに FPGA での OpenCL カーネルの実行までが含まれます。

1. [インテル FPGA SDK for OpenCL のダウンロード](#) (10 ページ)
2. [インテル FPGA SDK for OpenCL のインストール](#) (11 ページ)
3. [インテル FPGA SDK for OpenCL ユーザー環境変数の設定](#) (12 ページ)
4. [ソフトウェア・インストールの確認](#) (13 ページ)
5. [FPGA ボードのインストール](#) (14 ページ)
6. [エミュレーションによるホスト・ランタイムの機能性の検証](#) (15 ページ)
7. [OpenCL カーネルの FPGA ハードウェア・コンフィグレーション・ファイルの作成](#) (18 ページ)
8. [FPGA のハードウェア・イメージの更新](#) (20 ページ)
9. [FPGA での OpenCL カーネルの実行](#) (22 ページ)
10. [ソフトウェアのアンインストール](#) (24 ページ)
11. [FPGA ボードのアンインストール](#) (24 ページ)

2.1. インテル FPGA SDK for OpenCL のダウンロード

インテル FPGA SDK for OpenCL のダウンロード・センターから、Windows 向け インテル FPGA SDK for OpenCL および関連するすべてのソフトウェアが含まれているインストール・パッケージをダウンロードします。

OpenCL ダウンロード・センターの インテル FPGA SDK では、以下のすべてのソフトウェアおよびファイルを含む tar ファイルを提供しています。

- インテル FPGA SDK for OpenCL
 - インテル Arria® 10 および インテル Stratix® 10 デバイス以外のデバイス向けに インテル Quartus Prime スタンダード・エディションソフトウェア
 - インテル Arria 10 および インテル Stratix 10 デバイス向けに インテル Quartus Prime 開発ソフトウェア・プロ・エディションのソフトウェア
 - デバイス・サポート
1. インテル FPGA SDK for OpenCL ダウンロード・センターは次の URL にあります。
<http://dl.altera.com/opencl/>
 2. エディションを選択します。
 3. ソフトウェアのバージョンを選択します。デフォルトでは最新バージョンが選択されています。
 4. 以下のギア・レートのいずれかを選択します。

Intel Corporation. 無断での引用、転載を禁じます。Intel、インテル、Intel ロゴ、Altera、ARRIA、CYCLONE、ENPIRION、MAX、NIOS、QUARTUS および STRATIX の名称およびロゴは、アメリカ合衆国および/またはその他の国における Intel Corporation の商標です。インテルは FPGA 製品および半導体製品の性能がインテルの標準保証に準拠することを保証しますが、インテル製品およびサービスは、予告なく変更される場合があります。インテルが書面にて明示的に同意する場合を除き、インテルはここに記載されたアプリケーション、または、いかなる情報、製品、またはサービスの使用によって生じるいっさいの責任を負いません。インテル製品の顧客は、製品またはサービスを購入する前、および、公開済みの情報を信頼する前には、デバイスの仕様を最新のバージョンにしておくことをお勧めします。

*その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

ISO
9001:2008
登録済



- Akamai DLM3 ダウンロード・マネージャ
- ダイレクト・ダウンロード

5. **Windows SDK** タブで、ダウンロードするパッケージの下にある**ダウンロード**をクリックします。選択したエディションに応じて、次のパッケージをダウンロードできます。

Pro エディション Pro エディションを選択した場合は、次のパッケージをダウンロードできます。

- **インテル FPGA SDK for OpenCL(Quartus Prime Pro Edition ソフトウェアとデバイスを含む)**

このパッケージは、インテル FPGA SDK for OpenCL オフライン・コンパイラと インテル FPGA SDK for OpenCL ユーティリティを含むインテル FPGA SDK for OpenCL のコア機能を提供します。

- **インテル FPGA SDK for OpenCL(Quartus Prime Pro Edition ソフトウェアおよびインテル・コード・ビルダー搭載デバイスを含む)**

このパッケージは、OpenCL カーネルを開発するための IDE を提供するために、インテル FPGA SDK for OpenCL のすべての機能と Intel Code Builder との統合を提供します。

Standard エディション Standard エディションを選択した場合、次のパッケージをダウンロードできません。

- **インテル FPGA SDK for OpenCL(Quartus Prime ソフトウェアおよびデバイスを含む)**

このパッケージは、インテル FPGA SDK for OpenCL オフライン・コンパイラと インテル FPGA SDK for OpenCL ユーティリティを含むインテル FPGA SDK for OpenCL のコア機能を提供します。

ダウンロードおよびインストール方法の横の**もっと表示**をクリックすると、ダウンロードとインストールの手順が表示されます。

6. ダウンロード・ページのダウンロードおよびインストール方法で表示された手順を実行します。

関連情報

[Intel FPGA\(旧アルテラ\)ウェブサイト](#)

2.2. インテル FPGA SDK for OpenCL のインストール

このセクションでは、ダウンロードした tar ファイルを解凍し、インストレーション・ファイルを実行して全てのソフトウェアとファイルをインストールします。

Windows バージョンの インテル FPGA SDK for OpenCL をフォルダーにインストールします。

これらの手順を実行するには、管理者権限が必要です。

SDK、インテル Quartus Prime ソフトウェア、デバイス・サポート・ファイルを同時にインストールするには、以下の手順を行います。

1. ダウンロードした AOCL-<version>-<build>-windows.tar ファイルを一時的なフォルダーに解凍します。
2. インストーラを実行します。インテル は、インテル Quartus Prime ソフトウェアの両方のエディションをインストールすることを推奨します。



- a. setup.bat ファイルを実行し、インテル Quartus Prime スタンダード・エディションのソフトウェアを含む AOCL をインストールします。
 - b. setup_pro.bat ファイルを実行し、インテル Quartus Prime 開発ソフトウェア・プロ・エディションのソフトウェアを含む SDKL をインストールします。
3. **注意:** インストーラが、ソフトウェア・インストール・パスを指定するユーザー環境変数 `INTELFPGAOCSDKROOT` をセットします。

`INTELFPGAOCSDKROOT` が最新バージョンのソフトウェアを指定していることを確認してください。Windows コマンド・ウィンドウを開き、コマンド・プロンプトに `echo %INTELFPGAOCSDKROOT%` と入力します。

SDK インストールの返されるパスはの場所を指していない場合、`INTELFPGAOCSDKROOT` の設定を編集します。

環境変数設定の変更の手順について詳しくは、[インテル FPGA SDK for OpenCL ユーザー環境変数の設定](#)を参照してください。

関連情報

[インテル FPGA SDK for OpenCL ユーザー環境変数の設定 \(12 ページ\)](#)

2.3. インテル FPGA SDK for OpenCL ユーザー環境変数の設定

インテル FPGA SDK for OpenCL Windows ユーザー環境変数を恒常的にセットするか一時的にするかの選択が可能です。環境変数設定は、ソフトウェアに対して FPGA ボードとホスト・ランタイムを記述します。

注意: 環境変数を恒常的にセットする場合は、インストール時に一度、設定の適用を行います。環境変数を一時的にセットする場合には、インストール時およびそれ以降に実行するすべてのセッションで、設定の適用を行う必要があります。

表 2. インテル FPGA SDK for OpenCL Windows ユーザー環境変数設定

環境変数	含めるパス
<code>PATH</code>	<ol style="list-style-type: none">1. <code>\$(INTELFPGAOCSDKROOT)/bin</code>2. <code>%INTELFPGAOCSDKROOT%\host\windows64\bin</code>3. <code>%INTELFPGAOCSDKROOT%\host\windows64\bin</code> <p><code>INTELFPGAOCSDKROOT</code> がソフトウェア・インストールのパスを指定している場合</p>



- 恒常的な環境変数設定を適用するには、以下の手順を行います。
 - a. **Windows Start menu > Control Panel** をクリックします。(Windows 8.1 では Control Panel アプリケーションを開きます)
 - b. **System and Security > System** をクリックします。
 - c. **System** ウィンドウで **Advanced system settings** をクリックします。
 - d. **System Properties** ダイアログボックスの **Advanced** タブをクリックします。
 - e. **Environment Variables** をクリックします。
Environment Variables ダイアログボックスが表示されます。
 - f. 既存の環境変数設定を変更するには **User variables for <user_name>** の下で変数を選択してから **Edit** をクリックします。**Edit User Variable** ダイアログボックスで **Variable value** フィールドに環境変数設定を入力します。
 - g. 新しい環境変数を追加する場合には、**User variables for <user_name>** の下で **New** をクリックします。**New User Variable** ダイアログボックス・ボックスで **Variable name** フィールドと **Variable value** フィールドに環境変数名と設定をそれぞれ入力します。

環境変数に複数の設定がある場合には、それぞれの設定を区切るためにセミコロンを用います。

- 一時的な環境変数設定を適用する場合には、コマンド・ウィンドウを開き `%INTELFPGAOCCLSDKROOT%\init_opencl.bat` スクリプトを実行します。

以下にスクリプトの出力例を示します。

```
Adding %INTELFPGAOCCLSDKROOT%\bin to PATH
Adding %INTELFPGAOCCLSDKROOT%\host\windows64\bin to PATH
```

`init_opencl.bat` スクリプトの実行は、現在のコマンド・ウィンドウにのみ影響します。このスクリプトは以下のタスクを行います。

- インテル Quartus Prime インストレーションの検出
- Microsoft Visual Studio インストレーションの検出
- `LIB` 環境変数を適切にセットするために Microsoft Visual Studio 環境をインポート
- `PATH` 環境変数が Microsoft `LINK.EXE` ファイルおよび `aoc.exe` ファイルへのパスを含むことの確認

2.4. ソフトウェア・インストレーションの確認

`version` ユーティリティ・コマンドを呼び出すよにより、正しいバージョンの OpenCL がインストールされていることを確認できます。

- コマンド・プロンプトで `aocl version` ユーティリティ・コマンドを呼び出します。以下に示すような出力によりインストールの成功を通知します。

```
aocl <version>.<build> (Intel(R) FPGA SDK for OpenCL(TM), Version <version>
Build <build>, Copyright (C) <year>
    インテル
    Corporation)
```

- インストールが失敗した場合には、ソフトウェアを再インストールしてください。詳しくは、*Intel® FPGA ソフトウェアのインストールおよびライセンスのマニュアル* および *Intel FPGA (旧アルテラ) ナレッジベース* を参照してください。

関連情報

- [Intel FPGA のソフトウェアのインストールおよびライセンス](#)
- [Intel FPGA\(旧アルテラ\)ナレッジベース](#)

2.5. FPGA ボード のインストール

FPGA ボード向けに OpenCL アプリケーションを作成する前に、ボードベンダーからカスタム・プラットフォームをダウンロードしてインストールする必要があります。ほとんどのカスタム・プラットフォーム・インストーラーは管理者権限が必要です。ボードを Windows のホストシステムにインストールには、`install` ユーティリティ・コマンドを呼び出します。

ボードのインストールの手順を以下に示します。いくつかのカスタム・プラットフォームは追加のインストール作業が必要です。ボードのインストールに関する詳細は、ボードベンダーの資料を参照してください。

1. ボードベンダーの手順に従って、FPGA ボードをシステムに接続します。
2. ボードベンダーのウェブサイトから FPGA ボード向けのカスタム・プラットフォームをダウンロードします。インテル FPGA SDK for OpenCL Reference Platform (例: Stratix V Network Reference Platform (s5_net)) をダウンロードするには、ウェブサイトの [インテル FPGA SDK for OpenCL FPGA Platforms \(www.altera.co.jp\)](#) のページを参照してください。
3. カスタム・プラットフォームを任意のフォルダー (つまり、システムフォルダー以外) にインストールします。

複数のカスタム・プラットフォームを有する `aocl diagnose` などの SDK ユーティリティを使用する同じシステムで、同時に複数のカスタム・プラットフォームをインストールすることができます。カスタム・プラットフォームのサブディレクトリは `board_env.xml` ファイルを含みます。

複数のカスタム・プラットフォームを有するシステムでは、ホストプログラムは、カスタム・プラットフォームの MMD (メモリーマップド・デバイス) ライブラリー・ディレクトリにリンクするのではなく、ボードの検出に FPGA Client Driver (FCD)、従来の Altera Client Driver (ACD) を使用する必要があります。FCD がカスタム・プラットフォーム向けに正しく設定されている限り、FCD はインストールされたすべてのボードをランタイムで検出します。

4. `QUARTUS_ROOTDIR_OVERRIDE` ユーザー環境変数が正しい インテル Quartus Prime ソフトウェア・インストール・ディレクトリを指すように設定します。
インテル Arria 10 または インテル Stratix 10 デバイスを有する場合、`QUARTUS_ROOTDIR_OVERRIDE` を インテル Quartus Prime 開発ソフトウェア・プロ・エディションのインストール・ディレクトリを指すように設定します。それ以外の場合は、`QUARTUS_ROOTDIR_OVERRIDE` を インテル Quartus Prime スタンダード・エディション開発ソフトウェアを指すように設定します。
5. カスタム・プラットフォーム・ライブラリー (例: MMD (メモリーマップド) ライブラリー) へのパスを `PATH` 環境変数設定に追加します。

ユーザー環境変数の設定および `init_openc1` スクリプトの実行について詳しくは [インテル FPGA SDK for OpenCL ユーザー環境変数の設定](#) を参照してください。

6. コマンドプロンプトで `aocl install` を呼び出します。

`aocl install <path_to_customplatform>` を呼び出すと、ホストとハードウェア・カーネル・プログラム間の通信ができるようにボードドライバーもインストールされます。



注 ボードのインストールには管理者権限が必要です。管理者として Windows のコマンドプロンプトを実行するには、**Start > All Programs > Accessories** をクリックします。**Accessories** の下の **Command Prompt** を右クリックし、表示された **Run as Administrator** をクリックします。

- マシンにインストールされた FPGA デバイスのリストを照会するには、`aocl diagnose` コマンドを呼び出します。ソフトウェアが、`acl0` から `acl31` までの `acl` 番号で表示される `<device_name>` を含む出力を生成します。

注 `aocl diagnose` ユーティリティ実装後の可能性のあるエラーについては、[Possible Errors After Running the diagnose ユーティリティ](#)を参照してください。アクセラレーター・ボードの`<device_name>` の照会について詳しくは、「*FPGA ボードのデバイス名の照会*」の項を参照してください。

- FPGA ボードが正常にインストールされていることを確認するには、`aocl diagnose <device_name>` コマンドを呼び出し、ボードベンダーが推奨する任意の診断テストを実行します。

関連情報

- [Cyclone V SoC 開発キットのインストール](#)
- [FPGA ボードのデバイス名の照会 \(20 ページ\)](#)
- [インテル FPGA SDK for OpenCL FPGA のプラットフォーム](#)
- [インテル FPGA SDK for OpenCL ユーザー環境変数の設定 \(12 ページ\)](#)

2.6. エミュレーションによるホスト・ランタイムの機能性の検証

インテル FPGA SDK for OpenCL Emulator を用いて OpenCL デザイン例をエミュレーションすることにより、ホスト・ランタイムの機能性をテストします。

- カスタムまたはリファレンス・プラットフォームをインストールします。これは、エミュレーションが特定の FPGA ボードを対象とするためです。
- `QUARTUS_ROOTDIR_OVERRIDE` 環境変数が正しい インテル Quartus Prime ソフトウェアのエディションを指定していることを確認します。Windows コマンド・ウィンドウを開き、コマンド・プロンプトに `echo %QUARTUS_ROOTDIR_OVERRIDE%` と入力します。

インテル Quartus Prime ソフトウェアのインストール・フォルダーへのパスが返されない場合には、これを `QUARTUS_ROOTDIR_OVERRIDE` 設定に加えます。

- インテル Arria 10 以外また インテル Stratix 10 以外のデバイス向けには、`QUARTUS_ROOTDIR_OVERRIDE` は インテル Quartus Prime スタンダード・エディションソフトウェアのインストール・フォルダーを指定します。
 - インテル Arria 10 また インテル Stratix 10 デバイス向けには `QUARTUS_ROOTDIR_OVERRIDE` は、インテル Quartus Prime 開発ソフトウェア・プロ・エディションソフトウェアのインストール・フォルダーを指定します。
- `PATH` ユーザー環境変数設定に、Microsoft Visual Studio の `LINK.exe` ファイルへのパスを追加します。
 - `LIB` ユーザー環境変数設定に、Microsoft Visual Studio の Microsoft compilation time ライブラリへのパスを追加します。
- [OpenCL デザイン例のダウンロード \(16 ページ\)](#)



2. エミュレーション用カーネルのコンパイル (16 ページ)
3. ホスト・アプリケーションのビルド (16 ページ)
4. OpenCL カーネルエミュレート (17 ページ)

2.6.1. OpenCL デザイン例のダウンロード

ウェブサイトの OpenCL Design Examples ページでは、ダウンロードして FPGA で動作させることができる、多様なコンプレキシティのサンプル・アプリケーションを提供しています。

以下に Hello World デザインをダウンロードする場合の手順を示します。

1. OpenCL Design Examples ページで Basic Examples の下の **Hello World** をクリックします。
2. Hello World Design Example ページで Downloads の下の **<version> x64 Windows package (.zip)** をクリックし、使用するプラットフォーム用の圧縮ファイルをダウンロードします。
3. `exm_opencl_hello_world_x64_windows_<version>.zip` ファイルを、書き込みアクセス可能なフォルダー内に解凍および保存します。
重要: フォルダ名にスペースが含まれていないことを確認します。

関連情報

[OpenCL デザイン例の生成](#)

2.6.2. エミュレーション用カーネルのコンパイル

エミュレーション用 OpenCL カーネルをコンパイルするために、`aoc` コマンドに `-march=emulator` オプションを含めます。

- ターゲット FPGA ボード用にカーネルをコンパイルするには、コマンドプロンプトで `hello_world` デザインに移動し、次のコマンドを呼び出します。

```
aoc -march=emulator -v -board=<board_name> device/  
hello_world.cl -o=bin/hello_world_emulation.aocx
```

インテル FPGA SDK for OpenCL オフライン・コンパイラーは `hello_world.cl` カーネルソース・ファイルをコンパイルし、`bin` サブディレクトリーに `hello_world_emulation.aocx` エミュレーション固有の実行可能ファイルを作成します。

2.6.3. ホスト・アプリケーションのビルド

`<path_to_exm_opencl_hello_world_x64_windows_<version>>`
`\hello_world\hello_world.sln` ファイルにはホスト・ソリューションが含まれています。Microsoft Visual Studio の `.sln` ファイルでこれを開いてから、`main.cpp` ファイル内に OpenCL ホスト・アプリケーションをビルドします。

Microsoft Visual Studio 2015 を使用している場合、ACD とクロノスから Installable Client Driver (ICD) が必要です。ACD と ICD との Visual Studio 2015 をセットアップするには、事前のホストアプリケーションを構築するには、次のタスクを実行します。

1. FCD と ICD が正しく設定されていることを確認します。`aocl install<path_to_customplatform>` ユーティリティ・コマンドを起動すると、それらを設定するために失敗した場合は、手動で AFCD や ICD を設定する必要があります。



1. 手順については、*インテル FPGA SDK for OpenCL Programming Guide* の *Accessing Custom Platform-Specific Functions* と *Linking to the ICD Loader Library on Windows* のセクションを参照してください。
1. ホスト・アプリケーションを `OpenCL.lib` ライブラリにリンクします。
1. a. ソリューション・プロパティの下で、**Configuration Properties** > **Linker** > **Input** を選択します。
1. a. **Additional Dependencies** フィールドに、`OpenCL.lib` を入力します。
1. **注意:** ACD と ICD を使用しているため、ホスト・プログラムを `alteracl.lib` またはカスタム・プラットフォームの MMD ライブラリに直接リンクしないでください。

ホスト・アプリケーションをビルドするには、以下のステップを行います。

1. Microsoft Visual Studio で
`<path_to_exm_opencl_hello_world_x64_windows_<version>>`
`\hello_world\hello_world.sln` ファイルを開きます。
2. ビルド設定を確認します。デフォルトのビルド設定は **Debug** ですが、**Release** も使用できます。ソリューション・プラットフォームとして適切なオプションを選択する(たとえば、x64 アーキテクチャでは **x64** を選択)必要があります。
3. **Build** > **Build Solution** メニュー・オプションを選択するか、F7 キーを押すことにより、ソリューションをビルドします。
`hello_world.exe` 実行可能ファイルが
`<path_to_exm_opencl_hello_world_x64_windows_<version>>`
`\hello_world\bin` フォルダ内に作成されます。
4. 正常にビルドされたことを確認します。出力の末尾が以下に示すようなメッセージで終わっていれば、ビルドが成功しています。

```
1> Build succeeded.  
1>  
1> Time Elapsed 00:00:03:29  
===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
```

注 警告 LNK4009: PDB 'vc90.pdb' was not found with...は、ビルドに影響を与えないので無視することができます。以前のバージョンの Microsoft Visual Studio を使用して Windows ライブラリをビルドした場合に、コンパイラがこの種の警告を表示することがあります。

関連情報

- [カスタム・プラットフォーム固有の機能へのアクセス](#)
- [Windows 上の ICD ローダー・ライブラリへのリンク](#)

2.6.4. OpenCL カーネルエミュレート

OpenCL カーネルをエミュレーションするには、使用するカーネルにビルドしたプラットフォーム上で、`.aocx` エミュレーション・ファイルを実行します。

カーネルをエミュレーションするには、以下のステップを行います。

1. ユーティリティ・コマンド `aocl linkflags` を実行し、ホスト・アプリケーションをビルドするためにどのライブラリが必要かを調べます。ソフトウェアが、エミュレーションおよび通常のカーネル・コンパイル・フロー両方のライブラリをリストします。
2. ホスト・アプリケーションを、`aocl linkflags` ユーティリティ・コマンドにより返されたライブラリにリンクします。
3. `hello_world_emulation.aocx` ファイルを現在使用している作業ディレクトリーに移し、ホストがこれを容易に見つけることができるようにします。
4. Windows の場合、最初に `set`
`CL_CONTEXT_EMULATOR_DEVICE_INTELFPGA=<number_of_devices>` コマンドを呼び出してエミュレートされたデバイスの数を定義し、ホスト・アプリケーションを実行します。このコマンドは、エミュレータが提供する必要がある同一のエミュレーション・デバイスの数を特定します。
注 環境変数の `CL_CONTEXT_EMULATOR_DEVICE_INTELFPGA` が設定されていると、**意** エミュレートされたデバイスのみが使用可能になります。つまり、すべての物理ボードへのアクセスが無効になります。
5. ホスト・アプリケーションを実行した後、`set`
`CL_CONTEXT_EMULATOR_DEVICE_INTELFPGA=` コマンドを呼び出して
`CL_CONTEXT_EMULATOR_DEVICE_INTELFPGA` 変数の設定を解除します。

エミュレートされたカーネルを呼び出す度に、共有ライブラリのコピーである `<process_ID>-libkernel.so` がデフォルトの一時的ディレクトリーに作成されます。この `<process_ID>` は、各エミュレーションの実行に対して割り当てられた固有の数値です。デフォルトのディレクトリーは、Windows では `TMP` または `TEMP` 環境変数の設定により、Linux では `TMPDIR` の設定により変更することができます。

関連情報

[ホスト・アプリケーションの実行 \(22 ページ\)](#)

2.7. OpenCL カーネルの FPGA ハードウェア・コンフィグレーション・ファイルの作成

Windows バージョンの `hello_world` OpenCL デザイン例を、OpenCL Design Examples ページからダウンロードします。`hello_world.aocx` AOC 実行可能ファイルを作成するために、インテル FPGA SDK for OpenCL Offline Compiler を使用して `hello_world` カーネル・プログラムをコンパイルします。`.aocx` ファイルは、FPGA ハードウェア・コンフィグレーション・ファイルです。

FPGA ボードの正常なインストール、およびカーネルのエミュレート後に、デバイスで動作させる `.aocx` を作成します。以下に、`hello_world` デザイン例から `.aocx` ファイルを作成する手順を示します。OpenCL デザイン例について詳しくは、OpenCL Design Examples ページを参照してください。

注意:

インテル FPGA SDK for OpenCL の Offline Compiler は使用する FPGA ボード向けに OpenCL カーネルをコンパイルします。ハードウェア・コンフィグレーション・ファイルを作成する前に、FPGA ボードをインストールしておく必要があります。まだ FPGA ボードをインストールしていない場合には、FPGA ボードのインストールで手順を確認してください。

1. `QUARTUS_ROOTDIR_OVERRIDE` 環境変数が正しい インテル Quartus Prime ソフトウェアのエディションを指定していることを確認します。Windows コマンド・ウィンドウを開き、コマンド・プロンプトに `echo %QUARTUS_ROOTDIR_OVERRIDE%` と入力します。



インテル Quartus Prime ソフトウェアのインストール・フォルダーへのパスが返されない場合には、`QUARTUS_ROOTDIR_OVERRIDE` 設定にこれを加えます。

- インテル Arria 10 以外および インテル Stratix 10 以外のデバイス向けには、`QUARTUS_ROOTDIR_OVERRIDE` は インテル Quartus Prime スタンダード・エディションソフトウェアのインストール・フォルダーを指定します。
 - インテル Arria 10 および インテル Stratix 10 デバイス向けには、`QUARTUS_ROOTDIR_OVERRIDE` は インテル Quartus Prime 開発ソフトウェア・プロ・エディションソフトウェアのインストール・フォルダーを指定します。
2. 使用する FPGA ボードを選択します。カスタム・プラットフォームで使用可能な FPGA ボードのリストを表示するには、コマンド・プロンプトで `aoc --list-boards` コマンドを呼び出します。

`aoc` コマンドの `--list-boards` オプションについて詳しくは、*インテル FPGA SDK for OpenCL のプログラミングガイドの使用可能な FPGA ボード(--list-boards)* をカスタム・プラットフォームで表示の項を参照してください。

3. コマンド・プロンプトで、エミュレーションに使用した `hello_world.cl` ファイルを含む `hello_world` デザイン・フォルダーに移動します。
4. 使用する FPGA ボード向けカーネルをコンパイルするために、以下のコマンドを呼び出します。

```
aoc -v --board <board_name> device/hello_world.cl -o bin/hello_world.aocx
```

このコマンドは、以下のタスクを実行します。

- OpenCL ソース・コードから インテル Quartus Prime デザイン・プロジェクト・ファイルの生成
- 初期シンタックス・エラーのチェック
- 基本的な最適化の実行
- 必要な中間ファイルを含む `hello_world` フォルダーの作成
- *インテル FPGA SDK for OpenCL Offline Compiler Object File (.aoco)* の作成
- `.aocx` ファイルの作成

注 カーネルの複雑さによっては、`.aocx` ファイルのビルドに数時間かかる場合があります。コンパイルの進行状況を画面上に表示するには、`aoc` コマンドに `-v` フラグを含めます。以下に出力の一例を示します。

```
aoc: Environment checks are completed successfully.
You are now compiling the full flow!!
aoc: Selected target board <board_name>
aoc: Running OpenCL parser...
aoc: OpenCL parser completed successfully.
aoc: Compiling...
aoc: Linking with IP library ...
aoc: First stage compilation completed successfully.
aoc: Setting up project for CvP revision flow...
aoc: Hardware generation completed successfully.
```

コンパイル・プロセスが完了すると、AOC がライン `aoc: Hardware generation completed successfully.` を表示します。



aoc コマンドの `-board=<board_name>` オプションについて、詳しくは *Intel FPGA SDK for OpenCL のプログラミング・ガイドの特定のFPGA ボード(--board <board_name>)用のカーネルのコンパイルの項*を参照してください。

aoc コマンドの `-v` オプションについて詳しくは、*Intel FPGA SDK for OpenCL のプログラミング・ガイドのコンパイル進捗レポート(-v)の生成の項*を参照してください。

aoc コマンドの `-o=<filename>` オプションについて詳しくは、*Intel FPGA SDK for OpenCL のプログラミング・ガイドの Intel FPGA SDK for OpenCL オフライン・コンパイラ出力ファイル(-o=<filename>)の名前の指定の項*を参照してください。

関連情報

- [カスタム・プラットフォーム\(--list-ボード\)で使用可能な FPGA ボードリスト](#)
- [特定の FPGA ボード\(-board=<board_name>\)用のカーネルのコンパイル](#)
- [コンパイル進捗報告\(-v\)の生成](#)
- [AOC 出力ファイル名\(-o <filename>\)の指定](#)
- [OpenCL デザイン例の生成](#)
- [Intel FPGA SDK for OpenCL ユーザー環境変数の設定 \(12 ページ\)](#)
- [FPGA ボード のインストール \(14 ページ\)](#)

2.8. FPGA のハードウェア・イメージの更新

該当する場合には、FPGA で OpenCL カーネル・プログラムを実行する前に、FPGA のフラッシュメモリーが最新バージョンの OpenCL ソフトウェアを使用して作成したハードウェア・イメージを格納していることを確認します。

注意:

- カスタム・プラットフォームが有効な OpenCL イメージのフラッシュメモリーへのプリロードを必要とする場合には、Intel Quartus Prime デザイン・スイートのメジャー・リリースごとに、最新バージョンのソフトウェアと互換するハードウェア・イメージを FPGA のフラッシュメモリーにプログラミングします。
- Intel Arria 10 GX FPGA 開発キット付きの Intel FPGA SDK for OpenCL については、アプリケーション・ノートを参照してください。詳細については、*Intel FPGA SDK for OpenCL 用 Intel Arria 10 GX FPGA 開発キットの設定*を参照してください。

関連情報

[Intel FPGA SDK for OpenCL 用 Intel Arria 10 GX FPGA 開発キットの設定](#)

2.8.1. FPGA ボードのデバイス名の照会

一部の OpenCL ユーティリティ・コマンドでは、デバイス名 (<device_name>) を指定する必要があります。<device_name> は、FPGA デバイスに対応する acl 番号 (例 :acl0 ~ acl31) を参照します。アクセラレーター・ボードのリストを照会すると、OpenCL ソフトウェアはデバイス名の順にマシンにインストールされたリストを生成します。



- マシんにインストールされたデバイスのリストを照会するには、コマンドプロンプトで `aocl diagnose` と入力します。
ソフトウェアは次のような出力例を生成します。

```
aocl diagnose: Running diagnostic from ALTERAOCLSDKROOT/board/<board_name>/
<platform>/libexec

Verified that the kernel mode driver is installed on the host machine.

Using board package from vendor: <board_vendor_name>
Querying information for all supported devices that are installed on the
host machine ...

device_name  Status  Information
-----
acl0         Passed  <descriptive_board_name>
           PCIe dev_id = <device_ID>, bus:slot.func = 02:00.00,
           at Gen 2 with 8 lanes.
           FPGA temperature=43.0 degrees C.

acl1         Passed  <descriptive_board_name>
           PCIe dev_id = <device_ID>, bus:slot.func = 03:00.00,
           at Gen 2 with 8 lanes.
           FPGA temperature = 35.0 degrees C.

Found 2 active device(s) installed on the host machine, to perform a full
diagnostic on a specific device, please run aocl diagnose <device_name>

DIAGNOSTIC_PASSED
```

2.8.2. FPGA のフラッシュメモリーのプログラミング

インテル FPGA SDK for OpenCL デザイン例のハードウェア・イメージをデバイスのフラッシュメモリーにロードすることにより、FPGA をコンフィグレーションします。FPGA は、通電していない間もフラッシュメモリー内のハードウェア・コンフィグレーション・ファイルを保持します。システムに電源を投入した際に、FPGA 回路は、このフラッシュメモリーのハードウェア・イメージに基づいてコンフィグレーションされます。したがって、OpenCL と互換するハードウェア・コンフィグレーション・ファイルを、必ず FPGA のフラッシュメモリーにロードしておく必要があります。

また、多くのカスタム・プラットフォームでは、適切に機能するために OpenCL イメージをフラッシュメモリーへプリロードしておく必要があります。たとえば、ほとんどの PCIe ベースのボードがフラッシュメモリー内に有効な OpenCL イメージを必要とします。これにより、ホストシステムが最初に電源投入された際に、ボード上のハードウェアがイメージを使用して FPGA デバイスをコンフィグレーションします。FPGA が有効な OpenCL イメージでコンフィグレーションされなければ、システムが PCIe エンドポイントの列挙に失敗するか、あるいはドライバが機能しなくなります。

いずれのデザインを動作させる前にも、ボードのフラッシュメモリーが最新バージョンの OpenCL ソフトウェアと互換する有効な OpenCL イメージを有していることを確認します。ボード固有の要件については、使用するボードのベンダーの資料を参照してください。

注意:

ハードウェア・コンフィグレーション・ファイルを FPGA のフラッシュメモリーにロードする際には、ロード処理が完了するまでシステムの通電を保ってください。ロードには数分を要することがあります。また、OpenCL カーネルを呼び出す、あるいは FPGA ボードと通信するホスト・コードを起動しないでください。

ハードウェア・コンフィグレーション・ファイルを FPGA ボードのフラッシュメモリーにロードするには、以下のステップを行います。

1. カスタム・プラットフォームに必要なすべてのドライバとユーティリティをインストールします。

たとえば、一部のカスタム・プラットフォームでは、フラッシュメモリーにハードウェア・コンフィグレーション・ファイルをロードするために インテル FPGA ダウンロード・ケーブルのドライバーをインストールする必要があります。インストール手順については インテル FPGA ダウンロード・ケーブル ユーザーガイドを参照してください。

2. 使用するカスタム・プラットフォーム向けのデザイン例をダウンロードします。

注意: OpenCL Design Examples ページからデザイン例をダウンロードし、書き込みアクセスが可能な位置にデザイン例を解凍します。

3. フラッシュメモリーにハードウェア・コンフィグレーション・ファイルをロードするために、`aocl flash <device_name> <example_design_filename>.aocx` コマンドを呼び出します。このコマンドの<device_name>には使用する FPGA デバイスに対応する acl 番号 (例: `acl0~acl31`)が入ります。また、<example_design_filename>.aocx は、デザイン例パッケージの <example_design_filename>.cl ファイルから作成するハードウェア・コンフィグレーション・ファイルです。
4. デバイスあるいはコンピュータの電源を落としてから再投入します。

電源の再投入により、FPGA コンフィグレーション・デバイスがフラッシュメモリーからハードウェア・コンフィグレーション・ファイルを取得し、FPGA をコンフィグレーションすることを確実にします。

警告: 一部のカスタム・プラットフォームでは、フラッシュメモリーのプログラミング後にホスト・システム全体に電源の再投入を行う必要があります。たとえば、PCIe ベースのカスタム・プラットフォームでは、PCIe エンドポイントを再列挙するためにホスト・システムの再起動が必要な場合があります。インテル は、フラッシュメモリーのプログラミング後にホスト・システム全体の電源を再投入することを推奨します。

関連情報

- [Intel FPGA ダウンロード・ケーブル II \(以前は USB-Blaster II ダウンロードケーブル\) ユーザーガイド](#)
- [OpenCL デザイン例](#)
- [インテル FPGA SDK for OpenCL ユーザー環境変数の設定 \(12 ページ\)](#)

2.9. FPGA での OpenCL カーネルの実行

Microsoft Visual Studio バージョン 2010 Professional で OpenCL ホスト・アプリケーションをビルドし、`hello_world.exe` 実行可能ファイルを呼び出すことによりアプリケーションを動作させます。インテル FPGA SDK for OpenCL は、64 ビット・ホスト・バイナリとのみ互換しています。

関連情報

[ホスト・アプリケーションのビルド \(16 ページ\)](#)

2.9.1. ホスト・アプリケーションの実行

FPGA で OpenCL カーネルを実行するために、Windows で .sln ファイルからビルドしたホスト・アプリケーションを動作させます。



1. PATH 環境変数にパス%INTELFPGAOCSDKROOT%\host\windows64\bin を加えます。
2. コマンド・プロンプトで
`<path_to_exm_opencl_hello_world_x64_windows_<version>>`
\`hello_world\bin` フォルダ内のホスト実行可能ファイルに移動します。
3. `hello_world.exe` 実行可能ファイルを呼び出します。
`hello_world` 実行可能ファイルが、FPGA でカーネル・コードを実行します。

2.9.2. 正常なカーネル実行からの出力

使用する FPGA で OpenCL カーネルを実行するためにホスト・アプリケーションを動作させた際に、OpenCL は正常なカーネル実行を通知します。

以下に出力の例を示します。

```
Reprogramming device [0] with handle 1
Querying platform for info:
=====
CL_PLATFORM_NAME                = Intel(R) FPGA SDK for OpenCL(TM)
CL_PLATFORM_VENDOR              = Intel Corporation
CL_PLATFORM_VERSION = OpenCL 1.0 Intel(R) FPGA SDK for OpenCL(TM), Version
<version>

Querying device for info:
=====
CL_DEVICE_NAME                  = <board name> : <descriptive board
name>
CL_DEVICE_VENDOR                = <board vendor name>
CL_DEVICE_VENDOR_ID            = <board vendor ID>
CL_DEVICE_VERSION = OpenCL 1.0 Intel(R) FPGA SDK for OpenCL(TM), Version
<version>
CL_DRIVER_VERSION               = <version>
CL_DEVICE_ADDRESS_BITS         = 64
CL_DEVICE_AVAILABLE            = true
CL_DEVICE_ENDIAN_LITTLE        = true
CL_DEVICE_GLOBAL_MEM_CACHE_SIZE = 32768
CL_DEVICE_GLOBAL_MEM_CACHELINE_SIZE = 0
CL_DEVICE_GLOBAL_MEM_SIZE      = 8589934592
CL_DEVICE_IMAGE_SUPPORT        = true
CL_DEVICE_LOCAL_MEM_SIZE       = 16384
CL_DEVICE_MAX_CLOCK_FREQUENCY = 1000
CL_DEVICE_MAX_COMPUTE_UNITS    = 1
CL_DEVICE_MAX_CONSTANT_ARGS    = 8
CL_DEVICE_MAX_CONSTANT_BUFFER_SIZE = 2147483648
CL_DEVICE_MAX_WORK_ITEM_DIMENSIONS = 3
CL_DEVICE_MEM_BASE_ADDR_ALIGN  = 8192
CL_DEVICE_MIN_DATA_TYPE_ALIGN_SIZE = 1024
CL_DEVICE_PREFERRED_VECTOR_WIDTH_CHAR = 4
CL_DEVICE_PREFERRED_VECTOR_WIDTH_SHORT = 2
CL_DEVICE_PREFERRED_VECTOR_WIDTH_INT = 1
CL_DEVICE_PREFERRED_VECTOR_WIDTH_LONG = 1
CL_DEVICE_PREFERRED_VECTOR_WIDTH_FLOAT = 1
CL_DEVICE_PREFERRED_VECTOR_WIDTH_DOUBLE = 0
Command queue out of order?    = false
Command queue profiling enabled? = true
Using AOXC: hello_world.aocx

Kernel initialization is complete.
Launching the kernel...

Thread #2: Hello from the Intel(R) FPGA OpenCL(TM) compiler!

Kernel execution is complete.
```

2.10. ソフトウェアのアンインストール

Windows で インテル FPGA SDK for OpenCL をアンインストールするには、アンインストーラを実行し、これまでの設定向けに変更したすべての環境変数を元に戻します。

1. Windows の Start メニューのショートカットから、インテル Quartus Prime デザイン・スイートでインストール・フォルダーに移動します。
2. **Uninstall インテル Quartus Prime スタンダード・エディション** を選択します。
アンインストール・ウィザードが表示されます。
3. アンインストール・ウィザードで、以下のステップを行います。
 - a. **Individual components** を選択して **Next** をクリックします。
 - b. SDK を選択して **Next** をクリックします。
アンインストーラが インテル FPGA SDK for OpenCL をアンインストールします。
4. インテル Quartus Prime 開発ソフトウェア・プロ・エディションソフトウェアと関連付けられている インテル FPGA SDK for OpenCL をアンインストールするには、Altera<version> Pro Edition フォルダで **Uninstall インテル Quartus Prime 開発ソフトウェア・プロ・エディション** を選択し、アンインストール・ウィザードを開きます。
 - a. 3 (24 ページ)を繰り返します。
5. `PATH` 環境変数から以下のパスを削除します。
 - a. `$INTELFPGAOCLSDKROOT/bin`
 - b. `%INTELFPGAOCLSDKROOT%\host\windows64\bin`
6. `INTELFPGAOCLSDKROOT` 環境変数を削除します。
7. `QUARTUS_ROOTDIR_OVERRIDE` 環境変数を削除します。

2.11. FPGA ボードのアンインストール

Windows で FPGA ボードをアンインストールするには、`uninstall` ユーティリティ・コマンドを呼び出し、カスタム・プラットフォームをアンインストールし、関係する環境変数の設定を元に戻します。OpenCL アプリケーションを異なるカスタム・プラットフォームに属する FPGA ボードへ移行する場合には、必ず現在の FPGA ボードをアンインストールする必要があります。

FPGA ボードをアンインストールするには、以下のステップを行います。

1. ボード・ベンダが案内する手順に従って、使用しているマシンからボードを取り外します。
2. `aocl uninstall` ユーティリティ・コマンドを呼び出して、現在のホスト・コンピュータ・ドライバを削除します(例:PCIe ドライバ)。インテル FPGA SDK for OpenCL は、FPGA ボードと通信するためにこれらのドライバを使用しています。
3. カスタム・プラットフォームをアンインストールします。
4. `PATH` 環境変数の設定を元に戻します。



3. Linux 向け インテル FPGA SDK for OpenCL スタート・ガイド

インテル FPGA SDK for OpenCL セットアップ・プロセスには、ソフトウェアのダウンロードとインストール、FPGA ボードのインストール、ならびに FPGA での OpenCL カーネルの実行までが含まれます。

1. [インテル FPGA SDK for OpenCL のダウンロード](#) (25 ページ)
2. [インテル FPGA SDK for OpenCL のインストール](#) (26 ページ)
3. [インテル FPGA SDK for OpenCL ユーザー環境変数の設定](#) (27 ページ)
4. [ソフトウェア・インストールの確認](#) (28 ページ)
5. [FPGA ボードのインストール](#) (28 ページ)
6. [エミュレーションによるホスト・ランタイムの機能性の検証](#) (29 ページ)
7. [OpenCL カーネルの FPGA ハードウェア・コンフィグレーション・ファイルの作成](#) (32 ページ)
8. [FPGA のハードウェア・イメージの更新](#) (33 ページ)
9. [FPGA での OpenCL カーネルの実行](#) (35 ページ)
10. [ソフトウェアのアンインストール](#) (37 ページ)
11. [FPGA ボードのアンインストール](#) (37 ページ)

3.1. インテル FPGA SDK for OpenCL のダウンロード

ウェブサイトのダウンロード・センターから、Windows 向け インテル FPGA SDK for OpenCL および関連するすべてのソフトウェアが含まれているインストール・パッケージをダウンロードします。

ダウンロード・センターの FPGA SDK では、以下のすべてのソフトウェアおよびファイルを含む tar ファイルを提供しています。

- インテル FPGA SDK for OpenCL
 - インテル Quartus Prime スタンダード・エディション以外のデバイス用のソフトウェア インテル Stratix 10 デバイス
 - インテル Quartus Prime 開発ソフトウェア・プロ・エディション以下のためのソフトウェア インテル Arria 10 デバイスのデバイスサポート
1. 次の URL での インテル FPGA SDK for OpenCL のダウンロード・センターへ進みます。
<http://dl.altera.com/opencl/>
 2. エディションを選択します。
 3. ソフトウェアのバージョンを選択します。デフォルトでは最新バージョンが選択されています。
 4. 以下のギア・レートのいずれかを選択します。
 - **Akamai DLM3** ダウンロード・マネージャ



— ダイレクト・ダウンロード

5. **Linux SDK** タブで、ダウンロードするパッケージの下の**ダウンロード**をクリックします。
選択したエディションに応じて、次のパッケージをダウンロードできます。

Pro エディション Pro エディションを選択した場合は、次のパッケージをダウンロードできます。

- **インテル FPGA SDK for OpenCL(Quartus Prime Pro Edition ソフトウェアとデバイスを含む)**

このパッケージは、インテル FPGA SDK for OpenCL オフラインコンパイラと インテル FPGA SDK for OpenCL ユーティリティを含むインテル FPGA SDK for OpenCL のコア機能を提供します。

- **インテル FPGA SDK for OpenCL(Quartus Prime Pro Edition ソフトウェアおよびインテル・コード・ビルダー搭載デバイスを含む)**

このパッケージは、OpenCL カーネルを開発するための IDE を提供するために、インテル FPGA SDK for OpenCL のすべての機能と Intel Code Builder との統合を提供します。

Standard エディション Standard エディションを選択した場合、次のパッケージをダウンロードできません。

- **インテル FPGA SDK for OpenCL(Quartus Prime ソフトウェアおよびデバイスを含む)**

このパッケージは、インテル FPGA SDK for OpenCL オフラインコンパイラと インテル FPGA SDK for OpenCL ユーティリティを含むインテル FPGA SDK for OpenCL のコア機能を提供します。

ダウンロードおよびインストール方法の横の**もっと表示**をクリックすると、ダウンロードとインストールの手順が表示されます。

6. ダウンロード・ページのダウンロードおよびインストール方法で表示された手順を実行します。

関連情報

[Intel FPGA\(旧アルテラ\)ウェブサイト](#)

3.2. インテル FPGA SDK for OpenCL のインストール

ダウンロードした tar ファイルを解凍し、インストール・ファイルを実行して全てのソフトウェアとファイルをインストールします。

Linux バージョンの インテル FPGA SDK for OpenCL をディレクトリにインストールします。

- `sudo` または `root` 権限を有している必要があります。
- Linux OS カーネル・ソースとヘッダ(例:`kernel-devel.x86_64`と`kernel-headers.x86_64`)および、GCC(GNU Compiler Collection)(`gcc.x86_64`)をインストールする必要があります。
- Intel Code Builder を含むパッケージをインストールする場合、Intel Code Builder を実行するには、Java SE 1.8.71 以降がインストールされている必要があります。以前のバージョンの Java SE がインストールされている場合でも、Intel Code Builder のインストールを完了できます。ただし、Intel Code Builder を実行するには、Java バージョンの前提条件を満たす必要があります。



注意: ソフトウェアを、ディレクトリにいずれの `.cshrc` (C Shell Run Commands file) も `.bashrc` (Bash Run Commands file) も含んでいないシステムにインストールする場合には、環境変数 `INTELFPGAOCSDKROOT` と `PATH` を手動でセットする必要があります。または、`.cshrc` ファイルと `.bashrc` ファイルを作成してから環境変数をこれらに追加します。環境変数をセットした後に、使用する端末をリスタートして更新が反映されていることを確認します。

インテル FPGA SDK for OpenCL、インテル Quartus Prime ソフトウェア、デバイス・サポート・ファイルを同時にインストールするには、以下の手順を行います。

1. 一時ディレクトリにダウンロードした tar ファイルを解凍します。
2. インストーラを実行します。インテル は、デバイスのサポートに必要な インテル Quartus Prime ソフトウェアのエディションの 1 つをインストールすることを推奨します。
 - a. `setup.sh` ファイルを実行し、インテル Quartus Prime スタンダード・エディションソフトウェアを含む SDK をインストールします。
 - b. `setup_pro.sh` ファイルを実行し、インテル Quartus Prime 開発ソフトウェア・プロ・エディションソフトウェアを含む SDK をインストールします。
3. **注意:** インストーラが、ソフトウェア・インストール・パスを指定する環境変数 `INTELFPGAOCSDKROOT` をセットします。

`INTELFPGAOCSDKROOT` が最新バージョンのソフトウェアを指定していることを確認してください。シェルを開き、コマンド・プロンプトに `echo $INTELFPGAOCSDKROOT` と入力します。

インテル FPGA SDK for OpenCL インストールの返されるパスはの場所を指していない場合、`INTELFPGAOCSDKROOT` の設定を編集します。

環境変数設定の変更の手順について、詳しくは [インテル FPGA SDK for OpenCL ユーザー環境変数の設定](#) を参照してください。

関連情報

[インテル FPGA SDK for OpenCL ユーザー環境変数の設定 \(27 ページ\)](#)

3.3. インテル FPGA SDK for OpenCL ユーザー環境変数の設定

インテル FPGA SDK for OpenCL Linux ユーザー環境変数を恒常的にセットするか一時的にするかの選択が可能です。環境変数設定は、ソフトウェアに対して FPGA ボードとホスト・ランタイムを記述します。

注意: 環境変数を恒常的にセットする場合は、インストール時に一度、設定の適用を行います。環境変数を一時的にセットする場合には、インストール時およびそれ以降に実行するすべてのセッションで、設定の適用を行う必要があります。

表 3. インテル FPGA SDK for OpenCL Linux ユーザー環境変数設定

環境変数	含めるパス
<code>PATH</code>	<code>\$INTELFPGAOCSDKROOT/bin</code> <code>INTELFPGAOCSDKROOT</code> がソフトウェア・インストールのパスを指定している場合
<code>LD_LIBRARY_PATH</code>	<code>\$INTELFPGAOCSDKROOT/host/linux64/lib</code>

- 恒常的な環境変数設定を適用するには、シェルを開き、`export <variable_name>=<variable_setting>`、`<variable_setting>`、`:$<variable_name>` コマンドを入力します。

たとえば、コマンド `export PATH=「$INTELFPGAOCSDKROOT/bin」`、`INTELFPGAOCSDKROOT/bin」:$PATH` は、`PATH` 設定のリストに `$INTELFPGAOCSDKROOT/bin` を追加します。

- 一時的な環境変数設定を適用する場合には、コマンド・ライン端末を開き `source $INTELFPGAOCSDKROOT/init_opencl.sh` コマンドを実行します。

以下にスクリプトの出力例を示します。

```
Adding $INTELFPGAOCSDKROOT/bin to PATH
Adding $INTELFPGAOCSDKROOT/host/linux64/lib to LD_LIBRARY_PATH
```

3.4. ソフトウェア・インストールの確認

`version` コーティリティ・コマンドを呼び出すことにより、正しいバージョンの OpenCL がインストールされていることを確認できます。

- コマンド・プロンプトで `aocl version` コーティリティ・コマンドを呼び出します。以下に示すような出力によりインストールの成功を通知します。

```
aocl <version>.<build> (Intel(R) FPGA SDK for OpenCL(TM), Version <version>
Build <build>, Copyright (C) <year> Corporation)
```

- インストールが失敗した場合には、ソフトウェアを再インストールしてください。詳しくは、*Intel FPGA ソフトウェアのインストールおよびライセンスのマニュアル*および *Intel FPGA (旧アルテラ) ナレッジベース*を参照してください。

3.5. FPGA ボードのインストール

FPGA ボード向けに Linux で OpenCL アプリケーションを作成する前に、ボードベンダーからカスタム・プラットフォームをダウンロードおよびインストールする必要があります。ほとんどのカスタム・プラットフォーム・インストーラは、管理者権限を必要とします。 `install` コーティリティ・コマンドを呼び出して、ボードを Linux ホスト・システムにインストールします。

以下の手順は、ボードのインストール手順の概要を説明します。カスタム・プラットフォームの中には、追加のインストール作業が必要なものがあります。ボードのインストールの詳細については、使用するボードベンダーの資料を参照してください。

1. ボードベンダーが案内する手順に従って、FPGA ボードをシステムに接続します。
2. ボードベンダーのウェブサイトから、使用する FPGA ボード向けのカスタム・プラットフォームをダウンロードします。インテル FPGA SDK for OpenCL FPGA SDK for OpenCL Reference Platform (例: Stratix V Network Reference Platform)をダウンロードするには、ウェブサイトでの *インテル FPGA SDK for OpenCL FPGA SDK for OpenCL FPGA Platforms* を参照してください。
3. カスタム・プラットフォームを任意のディレクトリ(システム・ディレクトリではなく)にインストールします。



複数のカスタムプラットフォームを持つ `aocl diagnose` などのユーティリティを使用して、同じシステムに複数のカスタム・プラットフォームを同時にインストールできます。カスタム・プラットフォームのサブディレクトリには、`board_env.xml` ファイルが含まれています。

複数のカスタム・プラットフォームを持つシステムでは、ホスト・プログラムではなく、直接カスタム・プラットフォーム「メモリーマップされたデバイス(MMD)のライブラリにリンクするよりも、ボードを発見するために、アルテラのクライアントドライバ(ACD)を使用することを確認してください。

4. `QUARTUS_ROOTDIR_OVERRIDE` ユーザー環境変数が、使用する インテル Quartus Prime ソフトウェアのインストール・ディレクトリを指定するようにセットします。シェルを開き、コマンド・プロンプトで `echo $QUARTUS_ROOTDIR_OVERRIDE` と入力します。

インテル Arria 10 および インテル Stratix 10 デバイスを使用する場合、`QUARTUS_ROOTDIR_OVERRIDE` が インテル Quartus Prime 開発ソフトウェア・プロ・エディションソフトウェアのインストール・ディレクトリを指定するようにセットします。それ以外の場合には、`QUARTUS_ROOTDIR_OVERRIDE` が インテル Quartus Prime スタンダード・エディションソフトウェアのインストール・ディレクトリを指定するようにセットします。

5. `LD_LIBRARY_PATH` 環境変数設定に、カスタム・プラットフォーム・ライブラリ(例:MMD (memory-mapped)ライブラリ)へのパスを追加します。

Linux ユーザー環境変数の設定および `init_openc1` スクリプトの実行について、詳しくは [インテル FPGA SDK for OpenCL ユーザー環境変数の設定の項](#)を参照してください。

6. 注意: ボードをインストールするには、`sudo` または `root` 権限が必要です。

コマンド・プロンプトで `aocl install` を呼び出します。

`aocl install` を呼び出すことにより、ホスト・アプリケーションとハードウェア・カーネル・プログラムが通信するためのボードドライバもインストールされます。

7. 使用するマシンにインストールされた FPGA デバイスのリストを照会するには、`aocl diagnose` コマンドを呼び出します。

ソフトウェアが、`acl0` から `acl31` までの `acl` 番号で表示される `<device_name>` を含む出力を生成します。

注 `aocl diagnose` ユーティリティを実装した後に起こりうるエラーについては、[diagnose ユーティリティの実行後に発生する可能性のあるエラーのセクション](#)を参照してください。アクセラレータ・ボードの `<device_name>` の照会について、[FPGA ボードのデバイス名の照会の項](#)を参照してください。

8. FPGA ボードが正常にインストールされていることを確認するには、`aocl diagnose <device_name>` コマンドを呼び出し、ボードベンダーが推奨する任意の診断テストを実行します。

3.6. エミュレーションによるホスト・ランタイムの機能性の検証

インテル FPGA SDK for OpenCL Emulator を用いて OpenCL デザイン例をエミュレーションすることにより、ホスト・ランタイムの機能性をテストします。

1. カスタムまたはリファレンス・プラットフォームをインストールします。これは、エミュレーションが特定の FPGA ボードを対象とするためです。
2. `QUARTUS_ROOTDIR_OVERRIDE` 環境変数が正しい インテル Quartus Prime ソフトウェアのエディションを指定していることを確認します。シェルを開き、コマンド・プロンプトに `echo $QUARTUS_ROOTDIR_OVERRIDE` と入力します。

インテル Quartus Prime ソフトウェアのインストール・ディレクトリへのパスが返されない場合には、これを `QUARTUS_ROOTDIR_OVERRIDE` 設定に加えます。

- インテル Arria 10 以外および インテル Stratix 10 以外のデバイス向けには、`QUARTUS_ROOTDIR_OVERRIDE` は インテル Quartus Prime スタンダード・エディションソフトウェアのインストール・ディレクトリを指定します。
- インテル Arria 10 および インテル Stratix 10 デバイス向けには `QUARTUS_ROOTDIR_OVERRIDE` は、インテル Quartus Prime 開発ソフトウェア・プロ・エディションソフトウェアのインストール・ディレクトリを指定します。

3. `LD_LIBRARY_PATH` 環境変数設定に、インテル FPGA SDK for OpenCL ユーザー環境変数の設定で示されたパスが含まれていることを確認します。シェルを開き、コマンド・プロンプトで `echo $LD_LIBRARY_PATH` を入力します。

返されたパスに `$INTELFPGAOCSDKROOT/host/linux64/lib` が含まれていなければ、これらを `LD_LIBRARY_PATH` 設定に追加します。

エミュレートされたカーネルを呼び出す度に、共有ライブラリのコピーである `<process_ID>-libkernel.so` がデフォルトの一時的ディレクトリに作成されます。この `<process_ID>` は、各エミュレーションの実行に対して割り当てられた固有の数値です。デフォルトのディレクトリは、`TMPDIR` 環境変数の設定により変更することができます。

1. [OpenCL デザイン例のダウンロード](#) (30 ページ)
2. [エミュレーション用カーネルのコンパイル](#) (30 ページ)
3. [ホスト・アプリケーションのビルド](#) (31 ページ)
4. [OpenCL カーネルのエミュレーション](#) (31 ページ)

3.6.1. OpenCL デザイン例のダウンロード

ウェブサイトの OpenCL Design Examples ページでは、ダウンロードして FPGA で動作させることができる、多様なコンプレキシティのサンプル・アプリケーションを提供しています。

以下に Hello World デザインをダウンロードする場合の手順を示します。

1. OpenCL Design Examples ページで Basic Examples の下の **Hello World** をクリックします。
2. Hello World Design Example ページで Downloads の下の **<version> x64 Linux package (.tgz)** をクリックし、使用するプラットフォーム用の圧縮ファイルをダウンロードします。
3. `.tgz` ファイルを、書き込みアクセス可能なディレクトリ内に解凍および保存します。
重要: ディレクトリ名にスペースが含まれていないことを確認します。

関連情報

[OpenCL デザイン例の生成](#)

3.6.2. エミュレーション用カーネルのコンパイル

エミュレーション用 OpenCL カーネルをコンパイルするために、`aoc` コマンドに `-march=emulator` オプションを含めます。



- ターゲット FPGA ボード用にカーネルをコンパイルするには、コマンドプロンプトで hello_world デザインに移動し、次のコマンドを呼び出します。

```
aoc -march=emulator -v -board=<board_name> device/  
hello_world.cl -o=bin/hello_world_emulation.aocx
```

は hello_world.cl カーネルソース・ファイルをコンパイルし、bin サブディレクトリーに hello_world_emulation.aocx エミュレーション固有の実行可能ファイルを作成します。

3.6.3. ホスト・アプリケーションのビルド

<path_to_exm_opencl_hello_world_x64_linux_<version>>/hello_world/
Makefile ファイルにより、ホスト実行可能ファイルをビルドします。

ホスト・アプリケーションをビルドするには、以下のステップを行います。

- hello_world ディレクトリーに移動します。
- \$ make -f Makefile コマンドを呼び出します。あるいは、単に make コマンドを呼び出します。
hello_world 実行可能ファイルが
<path_to_exm_opencl_hello_world_x64_linux_<version>>/
hello_world/bin ディレクトリー内に作成されます。

3.6.4. OpenCL カーネルのエミュレーション

OpenCL カーネルをエミュレーションするには、カーネルを構築するプラットフォームで .aocx ファイルのエミュレーションを実行します。

カーネルをエミュレーションするには、次のステップを実行します。

- ユーティリティー・コマンド aocl linkflags を実行し、ホスト・アプリケーションを構築するためにどのライブラリーが必要かを調べます。ソフトウェアはエミュレーションと通常のカーネル・コンパイル・フロー両方のライブラリーをリストします。
- ホスト・アプリケーションは aocl linkflags ユティリティー・コマンドで返されたライブラリーとリンクします。
- hello_world_emulation.aocx ファイルを現在使用している作業ディレクトリーに移し、ホストがこれを容易に検索できるようにします。
- エミュレーション用のホスト・アプリケーションを実行するには、
envCL_CONTEXT_EMULATOR_DEVICE_ALTERA=<number_of_devices>
<host_application_filename> コマンドを呼び出します。
このコマンドは、Emulator が提供する必要がある同一のエミュレーション・デバイスの数を特定します。

エミュレーションされたカーネルの呼び出しごとに、デフォルトの一時ディレクトリー内に、<process_ID>が各エミュレーションの実行に割り当てられた固有の数値である <process_ID>-libkernel.so という共有ライブラリーのコピーを作成します。デフォルトのディレクトリーは、Windows では TMP または TEMP 環境変数の設定により、Linux では TMPDIR の設定により、オーバーライドできます。

関連情報

[ホスト・アプリケーションの実行 \(36 ページ\)](#)

3.7. OpenCL カーネルの FPGA ハードウェア・コンフィグレーション・ファイルの作成

Linux バージョンの hello_world OpenCL デザイン例を、OpenCL Design Examples ページからダウンロードします。hello_world.aocx AOC 実行可能ファイルを作成するために、インテル FPGA SDK for OpenCL の Offline Compiler を使用して hello_world カーネル・プログラムをコンパイルします。.aocx ファイルは、FPGA ハードウェア・コンフィグレーション・ファイルです。

FPGA ボードの正常なインストール、およびカーネルのエミュレート後に、デバイスで動作させる.aocx を作成します。以下に、hello_world デザイン例から.aocx ファイルを作成する手順を示します。OpenCL デザイン例について、詳しくは OpenCL Design Examples ページを参照してください。

注意:

インテル FPGA SDK for OpenCL の Offline Compiler は使用する FPGA ボード向けに OpenCL カーネルをコンパイルします。ハードウェア・コンフィグレーション・ファイルを作成する前に、FPGA ボードをインストールしておく必要があります。まだ FPGA ボードをインストールしていない場合には、FPGA ボードのインストールで手順を確認してください。

1. `QUARTUS_ROOTDIR_OVERRIDE` 環境変数が正しいエディションの インテル Quartus Prime ソフトウェアを指定していることを確認します。シェルを開き、コマンド・プロンプトに `echo $QUARTUS_ROOTDIR_OVERRIDE` と入力します

インテル Quartus Prime ソフトウェアのインストール・ディレクトリへのパスが返されない場合には、`QUARTUS_ROOTDIR_OVERRIDE` 設定にこれを加えます。

- インテル Arria 10 以外や インテル Stratix 10 以外のデバイス向けには、`QUARTUS_ROOTDIR_OVERRIDE` は インテル Quartus Prime スタンダード・エディションソフトウェアのインストール・フォルダーを指定します。
- インテル Arria 10 および インテル Stratix 10 デバイス向けには、`QUARTUS_ROOTDIR_OVERRIDE` は インテル Quartus Prime 開発ソフトウェア・プロ・エディションソフトウェアのインストール・フォルダーを指定します。

2. 使用する FPGA ボードを選択します。カスタム・プラットフォームで使用可能な FPGA ボードのリストを表示するには、コマンド・プロンプトで `aoc --list-boards` コマンドを呼び出します。

`aoc` コマンドの `-list-boards` オプションについて、詳しくは *インテル FPGA SDK for OpenCL のプログラミング・ガイドの使用可能な FPGA ボード(--list-boards)* をカスタム・プラットフォームで表示の項を参照してください。

3. コマンド・プロンプトで、エミュレーションに使用した `hello_world.cl` ファイルを含む `hello_world` デザイン・ディレクトリに移動します。
4. 使用する FPGA ボード向けカーネルをコンパイルするために、以下のコマンドを呼び出します。

```
aoc -v --board <board_name> device/hello_world.cl -o bin/hello_world.aocx
```

このコマンドは、以下のタスクを実行します。

- OpenCL ソース・コードから インテル Quartus Prime デザイン・プロジェクト・ファイルの生成
- 初期シンタックス・エラーのチェック
- 基本的な最適化の実行



- 必要な中間ファイルを含む hello_world ディレクトリの作成
- インテル FPGA SDK for OpenCL Offline Compiler Object File (.aoco)の作成
- .aocx ファイルの作成

注 カーネルの複雑さによっては .aocx ファイルのビルドに数時間かかる場合があります。コンパイルの進行状況を画面上に表示するには、aoc コマンドに `-v` フラグを含めます。以下に出力の一例を示します。

```
aoc: Environment checks are completed successfully.
You are now compiling the full flow!!
aoc: Selected target board <board_name>
aoc: Running OpenCL parser...
aoc: OpenCL parser completed successfully.
aoc: Compiling...
aoc: Linking with IP library ...
aoc: First stage compilation completed successfully.
aoc: Setting up project for CvP revision flow...
aoc: Hardware generation completed successfully.
```

コンパイル・プロセスが完了すると、AOC がライン `aoc: Hardware generation completed successfully.` を表示します。

aoc コマンドの `-board=<board_name>` オプションについて、詳しくは *インテル FPGA SDK for OpenCL のプログラミング・ガイドの特定のFPGA ボード(--board <board_name>)用のカーネルのコンパイルの項を参照してください。*

aoc コマンドの `-v` オプションについて、詳しくは *インテル FPGA SDK for OpenCL のプログラミング・ガイドのコンパイル進捗レポート(-v)の生成の項を参照してください。*

aoc コマンドの `-o=<filename>` オプションについて、詳しくは *インテル FPGA SDK for OpenCL のプログラミング・ガイドの インテル FPGA SDK for OpenCL オフライン・コンパイラ出力ファイル(-o=<filename>)の名前の指定の項を参照してください。*

関連情報

- [FPGA ボードのインストール \(28 ページ\)](#)
- [OpenCL デザイン例](#)
- [カスタム・プラットフォーム\(--list-ボード\)で使用可能な FPGA ボードリスト](#)
- [特定の FPGA ボード\(-board=<board_name> \)用のカーネルのコンパイル](#)
- [コンパイル進捗報告\(-v\)の生成](#)
- [AOC 出力ファイル名\(-o <filename> \)の指定](#)

3.8. FPGA のハードウェア・イメージの更新

該当する場合には、FPGA で OpenCL カーネル・プログラムを実行する前に、FPGA のフラッシュメモリーが最新バージョンの OpenCL ソフトウェアを使用して作成したハードウェア・イメージを格納していることを確認します。

注意:

- カスタム・プラットフォームが有効な OpenCL イメージのフラッシュメモリへのプリロードを必要とする場合には、Intel Quartus Prime デザイン・スイートのメジャー・リリースごとに、最新バージョンのソフトウェアと互換するハードウェア・イメージを FPGA のフラッシュメモリにプログラミングします。
- Intel Arria 10 GX FPGA 開発キット付きのについては、アプリケーション・ノートを参照してください。詳細については、*インテル FPGA SDK for OpenCL 用 Intel Arria 10 GX FPGA 開発キットの設定*を参照してください。

3.8.1. FPGA ボードのデバイス名の照会

一部の OpenCL コマンドでは、デバイス名 (<device_name>) を指定する必要があります。<device_name> は、FPGA デバイスに対応する acl 番号 (例 :acl0 ~ acl31) を参照します。アクセラレーター・ボードのリストを照会すると、OpenCL ソフトウェアはデバイス名の順にマシンにインストールされたリストを生成します。

- マシンにインストールされたデバイスのリストを照会するには、コマンドプロンプトで `aocl diagnose` と入力します。
ソフトウェアは次のような出力例を生成します。

```
aocl diagnose: Running diagnostic from ALTERAOCLSDKROOT/board/<board_name>/
<platform>/libexec

Verified that the kernel mode driver is installed on the host machine.

Using board package from vendor: <board_vendor_name>
Querying information for all supported devices that are installed on the
host machine ...

device_name  Status  Information
-----
acl0         Passed  <descriptive_board_name>
           PCIe dev_id = <device_ID>, bus:slot.func = 02:00.00,
           at Gen 2 with 8 lanes.
           FPGA temperature=43.0 degrees C.

acl1         Passed  <descriptive_board_name>
           PCIe dev_id = <device_ID>, bus:slot.func = 03:00.00,
           at Gen 2 with 8 lanes.
           FPGA temperature = 35.0 degrees C.

Found 2 active device(s) installed on the host machine, to perform a full
diagnostic on a specific device, please run aocl diagnose <device_name>

DIAGNOSTIC_PASSED
```

3.8.2. FPGA のフラッシュメモリのプログラミング

インテル FPGA SDK for OpenCL デザイン例のハードウェア・イメージをデバイスのフラッシュメモリにロードすることにより、FPGA をコンフィグレーションします。FPGA は、通電していない間もフラッシュメモリ内のハードウェア・コンフィグレーション・ファイルを保持します。システムに電源を投入した際に、FPGA 回路は、このフラッシュメモリのハードウェア・イメージに基づいてコンフィグレーションされます。したがって、OpenCL と互換するハードウェア・コンフィグレーション・ファイルを、必ず FPGA のフラッシュメモリにロードしておく必要があります。

また、多くのカスタム・プラットフォームでは、適切に機能するために OpenCL イメージをフラッシュメモリへプリロードしておく必要があります。たとえば、ほとんどの PCIe ベースのボードが、フラッシュメモリ内に有効な OpenCL イメージを必要とします。これにより、ホスト・システムが最初に電源投入さ



れた際に、ボード上のハードウェアがイメージを使用して FPGA デバイスをコンフィグレーションします。FPGA が有効な OpenCL イメージでコンフィグレーションされなければ、システムが PCIe エンドポイントの列挙に失敗するか、あるいはドライバが機能しなくなります。

いずれのデザインを動作させる前にも、ボードのフラッシュメモリが最新バージョンの OpenCL ソフトウェアと互換する有効な OpenCL イメージを有していることを確認します。ボード固有の要件については、使用するボードのベンダーの資料を参照してください。

注意:

ハードウェア・コンフィグレーション・ファイルを FPGA のフラッシュメモリにロードする際には、ロード処理が完了するまでシステムの通電を保ってください。ロードには数分を要することがあります。また、OpenCL カーネルを呼び出す、あるいは、それ以外でも FPGA ボードと通信するホスト・コードを起動しないでください。

ハードウェア・コンフィグレーション・ファイルを FPGA ボードのフラッシュメモリにロードするには、以下のステップを行います。

1. カスタム・プラットフォームに必要なすべてのドライバとユーティリティをインストールします。
2. 使用するカスタム・プラットフォーム向けのデザイン例をダウンロードします。
注意: OpenCL Design Examples ページからデザイン例をダウンロードし、書き込みアクセスが可能な位置にデザイン例を解凍します。
3. フラッシュメモリにハードウェア・コンフィグレーション・ファイルをロードするために、`aocl flash <device_name> <example_design_filename>.aocx` コマンドを呼び出します。このコマンドの<device_name>には使用する FPGA デバイスに対応する aocl 番号 (例: aocl0 ~ aocl31) が入ります。また、<example_design_filename>.aocx は、デザイン例パッケージの <example_design_filename>.cl ファイルから作成するハードウェア・コンフィグレーション・ファイルです。
4. デバイスあるいはコンピュータの電源を落としてから再投入します。

電源の再投入により、FPGA コンフィグレーション・デバイスがフラッシュメモリからハードウェア・コンフィグレーション・ファイルを取得し、これで FPGA をコンフィグレーションすることを確実にします。

警告: 一部のカスタム・プラットフォームでは、フラッシュメモリのプログラミング後にホスト・システム全体に電源の再投入を行う必要があります。たとえば、PCIe ベースのカスタム・プラットフォームでは、PCIe エンドポイントを再列挙するためにホスト・システムの再起動が必要な場合があります。インテル は、フラッシュメモリのプログラミング後にホスト・システム全体の電源を再投入することを推奨します。

関連情報

- [OpenCL デザイン例の生成](#)
- [インテル FPGA SDK for OpenCL ユーザー環境変数の設定 \(27 ページ\)](#)

3.9. FPGA での OpenCL カーネルの実行

OpenCL ホスト・アプリケーションを Makefile ファイルによりビルドする必要があり、このアプリケーションを `hello_world` 実行可能ファイルを呼び出すことにより動作させます。OpenCL アプリケーションをビルドするには、`gcc`™ や `make` といった GNU 開発ツールが必要です。

関連情報

[ホスト・アプリケーションのビルド \(31 ページ\)](#)

3.9.1. ホスト・アプリケーションの実行

FPGA で OpenCL カーネルを実行するために、Linux で .sln ファイルからビルドしたホスト・アプリケーションを動作させます。

1. `LD_LIBRARY_PATH` 環境変数にパス `$INTELFPGAOCCLSDKROOT/host/linux64/lib` を加えます。
2. コマンド・プロンプトで
`<path_to_exm_opencl_hello_world_x64_linux_<version>>/hello_world/bin` ディレクトリ内のホスト実行可能ファイルに移動します。
3. `hello_world` 実行可能ファイルを呼び出します。
`hello_world` 実行可能ファイルが、FPGA でカーネル・コードを実行します。

3.9.2. 正常なカーネル実行からの出力

使用する FPGA で OpenCL カーネルを実行するためにホスト・アプリケーションを動作させた際に、OpenCL は正常なカーネル実行を通知します。

以下に出力の例を示します。

```
Reprogramming device [0] with handle 1
Querying platform for info:
=====
CL_PLATFORM_NAME           = Intel(R) FPGA SDK for OpenCL(TM)
CL_PLATFORM_VENDOR        = Intel Corporation
CL_PLATFORM_VERSION = OpenCL 1.0 Intel(R) FPGA SDK for OpenCL(TM), Version
<version>

Querying device for info:
=====
CL_DEVICE_NAME             = <board name> : <descriptive board
name>
CL_DEVICE_VENDOR          = <board vendor name>
CL_DEVICE_VENDOR_ID       = <board vendor ID>
CL_DEVICE_VERSION = OpenCL 1.0 Intel(R) FPGA SDK for OpenCL(TM), Version
<version>
CL_DRIVER_VERSION         = <version>
CL_DEVICE_ADDRESS_BITS   = 64
CL_DEVICE_AVAILABLE      = true
CL_DEVICE_ENDIAN_LITTLE  = true
CL_DEVICE_GLOBAL_MEM_CACHE_SIZE = 32768
CL_DEVICE_GLOBAL_MEM_CACHELINE_SIZE = 0
CL_DEVICE_GLOBAL_MEM_SIZE = 8589934592
CL_DEVICE_IMAGE_SUPPORT  = true
CL_DEVICE_LOCAL_MEM_SIZE = 16384
CL_DEVICE_MAX_CLOCK_FREQUENCY = 1000
CL_DEVICE_MAX_COMPUTE_UNITS = 1
CL_DEVICE_MAX_CONSTANT_ARGS = 8
CL_DEVICE_MAX_CONSTANT_BUFFER_SIZE = 2147483648
CL_DEVICE_MAX_WORK_ITEM_DIMENSIONS = 3
CL_DEVICE_MEM_BASE_ADDR_ALIGN = 8192
CL_DEVICE_MIN_DATA_TYPE_ALIGN_SIZE = 1024
CL_DEVICE_PREFERRED_VECTOR_WIDTH_CHAR = 4
CL_DEVICE_PREFERRED_VECTOR_WIDTH_SHORT = 2
CL_DEVICE_PREFERRED_VECTOR_WIDTH_INT = 1
CL_DEVICE_PREFERRED_VECTOR_WIDTH_LONG = 1
CL_DEVICE_PREFERRED_VECTOR_WIDTH_FLOAT = 1
CL_DEVICE_PREFERRED_VECTOR_WIDTH_DOUBLE = 0
Command queue out of order? = false
Command queue profiling enabled? = true
Using AOCX: hello_world.aocx

Kernel initialization is complete.
Launching the kernel...
```



```
Thread #2: Hello from the Intel(R) FPGA OpenCL(TM) compiler!  
Kernel execution is complete.
```

3.10. ソフトウェアのアンインストール

Linux で インテル FPGA SDK for OpenCL をアンインストールするには、GUI アンインストーラを介してソフトウェア・パッケージを削除してから、ソフトウェア・ディレクトリを削除し、これまでの設定向けに変更したすべての環境変数を元に戻します。

1. 以下のステップを行い、ソフトウェア・パッケージを削除します。
 - a. インテル Quartus Prime スタンダード・エディションソフトウェアならびに インテル Quartus Prime 開発ソフトウェア・プロ・エディションソフトウェア向けをアンインストールするために、`<install directory>/uninstall` ディレクトリにある `aocl-<version>-uninstall.run` プログラムを実行します。
2. `PATH` 環境変数から `$INTELFPGAOCCLSDKROOT/bin` を削除します。
3. `LD_LIBRARY_PATH` 環境変数から `$INTELFPGAOCCLSDKROOT/host/linux64/lib` を削除します。
4. `INTELFPGAOCCLSDKROOT` 環境変数を削除します。
5. `QUARTUS_ROOTDIR_OVERRIDE` 環境変数を削除します。

3.11. FPGA ボードのアンインストール

Linux で FPGA ボードをアンインストールするには、`uninstall` ユーティリティー・コマンドを呼び出し、カスタム・プラットフォームをアンインストールして、関連のある環境変数の設定を元に戻します。OpenCL アプリケーションを異なるカスタム・プラットフォームに属する FPGA ボードへ移行する場合には、必ず現在の FPGA ボードをアンインストールする必要があります。

FPGA ボードをアンインストールするには、次の手順を行います。

1. ボードベンダーの手順に従って、マシンからボードを取り外します。
2. `aocl uninstall <path_to_customplatform>` ユーティリティー・コマンドを呼び出して、現在のホスト・コンピューター・ドライバー（例：PCIe ドライバー）を削除します。インテル FPGA SDK for OpenCL は、FPGA ボードと通信するためにこれらのドライバーを使用しています。
3. カスタム・プラットフォームをアンインストールします。
4. `LD_LIBRARY_PATH` 環境変数の設定を解除します。



A. 改訂履歴

表 4. インテル FPGA SDK for OpenCL のスタート・ガイドの改訂履歴

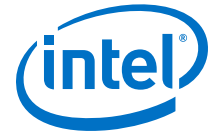
日付	バージョン	変更内容
2017 年 12 月	2017.12.08	<ul style="list-style-type: none"> Intel Code Builder を含む インテル FPGA SDK for OpenCL のインストール・パッケージの利用の可否を含む次のセクションを更新します。 <ul style="list-style-type: none"> インテル FPGA SDK for OpenCL のダウンロード (10 ページ) インテル FPGA SDK for OpenCL のダウンロード (25 ページ) Intel Code Builder の実行に必要な Java SE バージョンの前提条件を含むように、以下のセクションを更新しました。 <ul style="list-style-type: none"> インテル FPGA SDK for OpenCL の使用条件 (5 ページ) インテル FPGA SDK for OpenCL のインストール (26 ページ)
2017 年 11 月	2017.11.06	<ul style="list-style-type: none"> 以下の変更を実行します。 <ul style="list-style-type: none"> ALTERAOCLSDKROOT から INTELFPGAOCCLSDKROOT に変更 CL_CONTEXT_EMULATOR_DEVICE_ALTERA から CL_CONTEXT_EMULATOR_DEVICE_INTELFPGA に変更 Quartus Prime から インテル Quartus Prime に変更 Arria 10 から インテル Arria 10 に変更 USB Blaster から インテル FPGA ダウンロード・ケーブルに変更 インテル® FPGA SDK for OpenCL™ スタート・ガイド (4 ページ)では、明快さを向上させるために OpenCL リファレンス・ページを OpenCL 1.0 リファレンスページに変更。また Intel Arria 10 GX アプリケーション・ノートへのリファレンスを追加。 インテル FPGA SDK for OpenCL の前提条件では: <ul style="list-style-type: none"> インテルの優先アクセラレータ・ボードから Intel FPGA Preferred Board for OpenCL へ変更。 Microsoft Visual Studio 2010 Professional から Microsoft Visual Studio Professional 2010 以上へ変更。 インテル FPGA SDK for OpenCL の内容では: <ul style="list-style-type: none"> 「インテル FPGA SDK for OpenCL は、論理コンポーネント、ドライバ、および SDK 固有のライブラリとファイルを提供します。」では、論理コンポーネントをプログラムに変更。 Logic Components セクションの下で、「ホスト・プラットフォーム API とランタイム API」を「ホストとランタイム API」に変更。 ホスト・アプリケーションのビルド (16 ページ)では、Microsoft Visual Studio として Microsoft Visual Studio 2015 へのリファレンスを更新。 FPGA での OpenCL カーネルの実行 (22 ページ)では、Microsoft Visual Studio 2010 Professional を Microsoft Visual Studio として更新。 Linux と Windows の両方のセクションで、インテル FPGA SDK for OpenCL のライセンスに関するトピックを削除。

continued...

Intel Corporation. 無断での引用、転載を禁じます。Intel、インテル、Intel ロゴ、Altera、ARRIA、CYCLONE、ENPIRION、MAX、NIOS、QUARTUS および STRATIX の名称およびロゴは、アメリカ合衆国および/またはその他の国における Intel Corporation の商標です。インテルは FPGA 製品および半導体製品の性能がインテルの標準保証に準拠することを保証しますが、インテル製品およびサービスは、予告なく変更される場合があります。インテルが書面にて明示的に同意する場合を除き、インテルはここに記載されたアプリケーション、または、いかなる情報、製品、またはサービスの使用によって生じるいっさいの責任を負いません。インテル製品の顧客は、製品またはサービスを購入する前、および、公開済みの情報を信頼する前には、デバイスの仕様を最新のバージョンにしておくことをお勧めします。

*その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

ISO
9001:2008
登録済



日付	バージョン	変更内容
		<ul style="list-style-type: none"> • 次のトピック インテル Stratix 10 でデバイスのサポートを追加。 <ul style="list-style-type: none"> – インテル FPGA SDK for OpenCL のダウンロード (10 ページ) – FPGA ボード のインストール (14 ページ) – エミュレーションによるホストランタイムの機能性の検証 (15 ページ) – OpenCL カーネルの FPGA ハードウェア・コンフィグレーション・ファイルの作成 (18 ページ) – インテル FPGA SDK for OpenCL のダウンロード (25 ページ) – FPGA ボードのインストール (28 ページ) – エミュレーションによるホストランタイムの機能性の検証 (29 ページ) – OpenCL カーネルの FPGA ハードウェア・コンフィグレーション・ファイルの作成 (32 ページ) • :シングル・ダッシュと-option=<value>の表記規則を次のトピックに実装。 <ul style="list-style-type: none"> – インテル FPGA SDK for OpenCL セットアップ・プロセスの概要 (7 ページ) – OpenCL カーネルの FPGA ハードウェア・コンフィグレーション・ファイルの作成 (18 ページ) – エミュレーション用カーネルのコンパイル (16 ページ) • ガイドの中で AOCL_BOARD_PACKAGE_ROOT への参照が廃止されたので削除。 • aocl install のインスタンスを aocl install <path_to_customplatform>に更新。 • aocl uninstall のインスタンスを aocl uninstall <path_to_customplatform>に更新。 • インテル FPGA SDK for OpenCL セットアップ・プロセスの概要 (7 ページ)では、aocl diagnose を実装した後の考えられるエラーについて、インストール・プロセス概要図の後にノートを追加。 • FPGA のハードウェア・イメージの更新 (20 ページ)では、インテル FPGA SDK for OpenCL のインテル Arria 10 GX FPGA 開発キットのコンフィグレーションのアプリケーション・ノートの設定に関するノートと関連リンクを追加。
2017年5月	2017.05.05	<ul style="list-style-type: none"> • Altera Client Driver(ACD)を FPGA Client Driver(FCD)に変更。 • Windows および Linux 用の インテル FPGA SDK for OpenCL のダウンロードでダウンロードの手順を更新。 • ダウンロードした OpenCL のデザイン例の解凍先のフォルダー名にスペースを含めない注意を追加。
2016年10月	2016.10.31	<ul style="list-style-type: none"> • Altera SDK for OpenCL を インテル FPGA SDK for OpenCL に変更。 • Altera Offline Compiler を インテル FPGA SDK for OpenCL オフライン・コンパイラに変更。 • Windows および Linux 用の FPGA ボードのインストールでは、次の更新プログラムを提供。 <ul style="list-style-type: none"> – SDK がマルチ・カスタム・プラットフォームのインストールをサポートしていることに注意してください。マルチ・カスタム・プラットフォームのインストールで各ボード上の SDK ユーティリティを使用するには、AOCL_BOARD_PACKAGE_ROOT 環境変数の設定は関連するボードのカスタム・プラットフォーム・サブディレクトリに対応している必要があります。 – マルチ・カスタム・プラットフォーム・システムでは、ホストプログラムは MMD ライブラリに直接リンクするのではなく、ACD を使用してボードを検出する必要があります。 • Windows 向けホスト・アプリケーションの構築では、ホスト・アプリケーションを構築する前に、Microsoft Visual Studio 2015 で使用するための ACD と ICD を設定するための前提条件タスクの概要を説明。
2016年5月	2016.05.02	<ul style="list-style-type: none"> • アルテラ・ウェブサイト上のオペレーティング・システムのサポートページへのリンクをサポートされている Windows 版と Linux 版のリストを交換。 • %ALTERAOCLSDKROOT%\windows64\bin 設定を Windows 環境変数のリストに追加。 • 複数のデバイスをエミュレートするための CL_CONTEXT_EMULATOR_DEVICE_ALTERA 変数の設定に関する Windows の指示を修正。
continued...		



日付	バージョン	変更内容
2015年11月	2015.11.02	<ul style="list-style-type: none">• <i>Quartus II</i> を <i>Quartus Prime</i> に変更。• サポートされる Windows のバージョンに Windows 8.1 を追加。• tar ファイルのダウンロードおよびインストール手順を修正し、AOCL、Quartus Prime ソフトウェア、およびデバイス・サポートを付加。• AOCL のみのインストール手順は現在のバージョンでは使用できないため、これを削除。• hello_world デザイン例をエミュレーションすることによりホスト・ランタイムの機能性を確認するための手順を追加。• 図 <i>FPGA プログラミングの概要</i> を修正し、プログラミング・フローにエミュレーションを追加。• アンインストール手順を更新。
2015年5月	15.0.0	<ul style="list-style-type: none">• 手順を以下の2つの項に再編：<ul style="list-style-type: none">– Windows 向け AOCL スタート・ガイド– Linux 向け AOCL スタート・ガイド
2014年12月	14.1.0	<ul style="list-style-type: none">• インフォメーション・フローの再編。• RHEL (Red Hat Enterprise Linux) バージョン・サポートの更新。• AOCL の内容の項を追加。• 新しい AOCL (Altera Software Development Kit (SDK) for OpenCL) のシングル・ライセンス向けにライセンス取得の手順を更新。• ボードのアンインストール手順を更新し、aocl uninstall コマンドを追加。• 環境変数の設定のための init_openc1 スクリプトに関する情報を追加。• ソフトウェアおよびボードのアンインストール手順をソフトウェアのアンインストールおよび FPGA ボードのアンインストールの項として再編。
2014年6月	14.0.0	<ul style="list-style-type: none">• 使用条件 s の項を更新。• 図 AOCL インストール手順の概要を更新。• ソフトウェア・ダウンロード手順を更新。• Windows 向けの AOCL インストールおよびアンインストールの手順を更新。• Linux システム向けに：<ul style="list-style-type: none">– Installing the AOCL on Linux Using RPM および Uninstalling the AOCL on Linux Using RPM の項を追加。– Installing the AOCL on Linux Using the GUI Installer の項を更新。• ソフトウェア・ライセンスの取得の項を追加• FPGA ボードのインストール d の項でデバイスの照会および診断テストの実行の手順を更新• OpenCL カーネルの FPGA ハードウェア・コンフィギュレーション・ファイルの作成の項を更新します。<ul style="list-style-type: none">– AOCL_BOARD_PACKAGE_ROOT 向けにセットするパスを更新– hello_world.cl をコンパイルする際の -v オプションによる AOC の出力例を更新– AOC 出力を更新• IFPGA のデバイス名の特定の項の追加• hello_world のディレクトリおよびファイル名の更新に伴い、ホスト・アプリケーションのビルドおよび実行のための手順を修正• FPGA ボードのアンインストールの項を追加
2013年12月	13.1.1	<ul style="list-style-type: none">• 使用条件の項を更新し、Linux カーネル・ソース、ヘッダ、および GCC が必要であることを付加
2013年11月	13.1.0	<ul style="list-style-type: none">• インフォメーション・フローの再編。• 使用条件の項を更新• ボード・インストール手順を更新• ボード・インストール手順を更新• AOCL と Quartus II ソフトウェアのインストールの項を追加• ソフトウェアのインストールおよびアンインストール手順を更新。

continued...



日付	バージョン	変更内容
		<ul style="list-style-type: none"> 以下の図を更新 <ul style="list-style-type: none"> AOCL インストール手順の概要 FPGA プログラミングの概要 ライセンスの項を削除 board-specific の手順およびコンフィグレーション手順を全て削除 カーネル設定および FPGA プログラミングについて説明するための OpenCL アプリケーション例を moving_average から hello_world に変更 更新されたフラッシュ・プログラミング手順を含む FPGA のハードウェア・イメージの更新の項を追加 Windows 向け USB-Blaster ドライバーのインストールの項を削除 Windows および Linux システム向けの、FPGA での正常な hello_world カーネル実行の出力を更新 図 移動平均の例の内容を削除 図 Visual Studio で host.sln を開くを削除
2013年6月	13.0 SP1.0	<ul style="list-style-type: none"> 必要な Quartus II および AOCL ソフトウェアのバージョンを 13.0 から 13.0 SP1 に更新 図 A Correct Windows Device Manager After Complete Board Driver Installation for a BittWare Board を追加 Windows 向け Verifying the Functionality of the BittWare Board の項でベンダーおよびデバイス ID を更新 ディレクトリに .cshrc ファイルも .bashrc ファイルも含まない Linux システムに AOCL をインストールする手順を更新 AOCL デザイン例へのパスを更新 図 Contents of the Moving Average Example を更新 図 Contents of the Moving Average Example を更新
2013年5月	13.0.1	<ul style="list-style-type: none"> OpenCL_SDK フォルダあるいはディレクトリを AOCL に改名 Linux システムで AOCL のインストールが .cshrc および .bashrc ファイルに依存することについての注意を追加 Linux システムで AOCL のインストールが .cshrc および .bashrc ファイルに依存することについての注意を追加 Windows 7 を使用するシステムで Jungo WinDriver のインストールが成功しない可能性があることについての注意を追加。再インストール手順を追加 BittWare FPGA ボードに実装されていない aocl コマンドについてのエラー・メッセージが表示されることについての注意を追加 環境変数 AOCL_BOARD_PACKAGE_ROOT の設定についての注意メッセージを追加 環境変数 AOCL_BOARD_PACKAGE_ROOT の設定についての注意メッセージを追加 Windows および Linux システム向けにパスを AOCL のデフォルト位置に修正 Linux システムで AOCL をインストールする際に PATH 環境変数に追加するパス名を修正。パス名は \$QUARTUS_ROOTDIR/bin64 ではなく \$QUARTUS_ROOTDIR/bin
2013年5月	13.0.0	<ul style="list-style-type: none"> インストールおよびコンパイル手順を更新 インストールおよびコンパイル手順を更新 フラッシュ・プログラミング手順を更新し、これを Appendix A に移動 ソフトウェアおよび資料ダウンロードのページへのリンクを更新
2012年11月	12.1.0	初版。