

SDx 開発環境

リリース ノート、インストールおよびライセンス ガイド

UG1238 (v2018.1) 2018 年 4 月 4 日

この資料は表記のバージョンの英語版を翻訳したもので、内容に相違が生じる場合には原文を優先します。資料によっては英語版の更新に対応していないものがあります。日本語版は参考用としてご使用の上、最新情報につきましては、必ず最新英語版をご参照ください。

改訂履歴

次の表に、この文書の改訂履歴を示します。

セクション	改訂のまとめ
2018 年 4 月 4 日 バージョン 2018.1	
2018.1 SDSoc および SDAccel 共通のアップデート	2018.1 の SDx™ のアップデートに関する情報を追加。
2018.1 SDSoc 開発環境の変更点	2018.1 の変更点に関する詳細を追加。
SDAccel 開発環境 2018.1 の変更点	2018.1 の変更点に関する詳細を追加。

目次

改訂履歴.....	2
第 1 章: リリース ノートおよびサポートされるハードウェア.....	4
SDSoC - SDAccel 開発環境共通機能.....	4
2018.1 SDSoC および SDAccel 共通のアップデート.....	4
SDSoC 開発環境の機能.....	5
SDAccel 開発環境の機能.....	6
第 2 章: SDx 環境の概要.....	16
SDSoC 概要.....	16
SDAccel の概要.....	17
ハードウェア要件.....	17
ソフトウェア要件.....	18
SDSoC インストールについて.....	18
SDAccel インストール.....	19
第 3 章: ザイリンクス ライセンス サイトでのライセンスの取得.....	22
第 4 章: SDx 環境のインストール.....	24
ツールのインストール準備.....	24
SDSoC および SDAccel のインストール.....	24
付録 A: その他のリソースおよび法的通知.....	32
参考資料.....	32
Documentation Navigator およびデザイン ハブ.....	33
お読みください: 重要な法的通知.....	33

リリース ノート および サポート される ハードウェア

SDSoC - SDAccel 開発環境共通機能

SDSoC™ および SDAccel™ は、Vivado® Design Suite と完全に統合されているので、別の Vivado Design Suite イメージは必要なくなりました。SDx™ Eclipse UI (SDx GUI) は、ユーザーの生産性を高めるために拡張されています。バックグラウンドのビルドにより GUI の操作がブロックされることはなくなり、New Project ウィザードが効率的になりました。

2017.4 以降のリリースでは、5.x DSA プラットフォームが使用されます。これらのプラットフォームは、プラットフォームのダイナミック コンフィギュレーションをイネーブルにします。

SDx コンパイラでは、リンク時に未使用の DDR に関連するインターフェイス ロジックが最適化で自動的に削除されるようになっています。プロファイリングおよびデバッグ情報を取り込むロジックは、コンパイル プロセス中にダイナミックに追加できます。以前は、このロジックが DSA 内に既に存在している必要がありました。ダイナミックプラットフォームでは、FPGA の使用可能なリソースをより効率的に使用できるので、コンパイル時間を最大で 45% 改善できます。

このリリースの新機能の多くに、5.x DSA を使用する必要があります。新規 5.x DSA および関連機能を使用できるようにするため既存のプロジェクトをアップデートする方法については、[SDAccel 移行サマリ](#)を参照してください。

2018.1 SDSoC および SDAccel 共通のアップデート

2018.1 では、SDSoC™ および SDAccel™ の深さ設定が変更されています。

2017.4 リリースまでは、`#pragma HLS stream` の `depth=N` 引数や `config_dataflow` の `-fifo_depth` 引数を使用した場合など、設計者のリクエストした FIFO 深さが RTL 生成段階で自動的に $N+1$ にインクリメントされていました。これにより、ユーザー リクエスト、GUI に表示される深さ、RTL での実際の深さが一致していませんでした。

2018.1 では、このような深さのインクリメントは発生しなくなっています。このため、以前 $N+1$ の深さにするために深さ N を指定していた場合は、デッドロック状態になるデザインもあります。こういったデザインは、FIFO 深さを指定するのに使用した TdI コマンドまたはプラグマのいずれかを変更して、リクエストの深さが $N+1$ になるようにする必要があります。

SDSoC 開発環境の機能

2018.1 SDSoC 開発環境の変更点

このリリースの SDSoC™ 開発環境に含まれるアップデートは、次のとおりです。

- ARM ツールチェーンを 6.2.1 から 7.2.1 に、MicroBlaze™ を 6.2.0 から 7.2.0 にアップデート
- Base Zynq® および Zynq UltraScale+ MPSoC プラットフォームをアップグレード
 - Vivado® 2018.1 IP のバージョン
 - スタンドアロン BSP、4.14 Linux カーネルをアップデート
- Simple DMA で最大 32 MB の転送をサポート (以前は 8 MB)
- C 呼び出し可能な IP ライブラリの作成 - sdx_pack ユーティリティが拡張され、出カスカラーおよび 512 ビット幅までの AXIMM インターフェイスをサポート
- ZCU104 および ZCU106 ベース プラットフォームが別のダウンロードから使用可能。
 - ZCU104 については、[アンサー 70710](#) を参照してください。
 - ZCU106 については、[アンサー 70859](#) を参照してください。
- バグ修正

2017.4 SDSoC 開発環境の新機能

2017.4 リリースの SDSoC™ 開発環境に含まれるアップデートは、次のとおりです。

- SDx™ GUI。
 - Vivado から DSA (デザイン サポート アーカイブ) をエクスポート後に SDSoC プラットフォームを作成するための新しいプラットフォーム プロジェクト タイプ。
 - プラットフォームをターゲットにするサブプロジェクトを複数含む新しいプラットフォーム プロジェクト タイプ。
 - OpenCV および画像処理アプリケーション用の XFAST ソース コード ライブラリからハードウェア関数を選択可能。
- RTL アクセラレータを C 呼び出し可能なライブラリにパッケージするための新しいユーティリティ sdx_pack。
- 別々のファイルにあるアクセラレータ関数をコンパイルするため、Vivado HLS を並列起動。
- その他の sds++ 改善点

- 。 データフロー解析、リソース共有、システム ポートのロード バランス、自動バリア同期 (wait 挿入) などを拡張して、アクセラレータ パイプラインのシステム パフォーマンスを改善。
- 。 struct/class 引数内で複数の配列メンバーをサポート。
- 。 ザイリンクス SDK ポート (freertos901_xilinx) を使用して FreeRTOS を v9.0.1 にアップデート。
- 。 Cortex-R5 コンパイラのサポートを gcc 6.2.1 にアップデート。
- 。 エラー処理およびメッセージを改善。
- 。 Zybo および MicroZed プラットフォームはボード ベンダーからのみ使用可能。
- 。 バグ修正。

SDAccel 開発環境の機能

SDAccel 開発環境 2018.1 の変更点

このリリースの SDAccel™ 開発環境では、既存の機能に次の変更が加えられています。

xilinx_vcu1525_dynamic_5_0 DSA は xilinx_vcu1525_dynamic_5_1 DSA に置換されています。

5.1 DSA では、5.0 DSA から次の点が更新されています。

- 。 バグ修正
 - 。 断続的に一時停止が発生する問題を修正するために、フリーランニング クロックのリセットを変更
 - 。 タイミング クロージャ改善のため、周波数カウンターをダイナミックに移動し、別の AXI Lite ポートを追加
- 。 改善点
 - 。 DDR4 bank1 の最大帯域幅を 20% 改善
 - 。 Memory Subsystem と AXI4-Lite IP の配置および SLR クロッシングを改善

2017.4 SDAccel 開発環境の新機能

2017.4 リリースの SDAccel™ 開発環境に含まれるアップデートは、次のとおりです。

5.x DSA および機能

注記: 2018.1 では、5.0 DSA が 5.1 に置き換わっています。 [SDAccel 開発環境 2018.1 の変更点](#) を参照してください。

表 1: xilinx_vcu1525_dynamic_5_0

エリア	SLR 0	SLR 1	SLR 2
一般情報			
SLR 記述	デバイスの下部、ダイナミック領域専用。	デバイスの中部、ダイナミック領域およびスタティック領域リソースで共有。	デバイスの上部、ダイナミック領域専用。

表 1: xilinx_vcu1525_dynamic_5_0 (続き)

エリア	SLR 0	SLR 1	SLR 2
ダイナミック領域の Pblock 名	pfm_top_i_dynamic_region_ pblock_dynamic_SLR0	pfm_top_i_dynamic_region_ pblock_dynamic_SLR1	pfm_top_i_dynamic_region_ pblock_dynamic_SLR2
計算ユニット配置構文 ¹	set_property CONFIG.SLR_ASSIGNMENTS SLR0 [get_bd_cells <cu_name>]	set_property CONFIG.SLR_ASSIGNMENTS SLR1 [get_bd_cells <cu_name>]	set_property CONFIG.SLR_ASSIGNMENTS SLR2 [get_bd_cells <cu_name>]
ダイナミック領域で使用可能なグローバル メモリ リソース ²			
メモリ チャンネル、システム ポート名	bank0 (16 GB DDR4)	bank1 (16 GB DDR4、スタティック領域) bank2 (16 GB DDR4、ダイナミック領域)	bank3 (16 GB DDR4)
ダイナミック領域で使用可能なファブリック リソースの概算			
CLB LUT	388K	199K	388K
CLB レジスタ	776K	399K	776K
ブロック RAM タイル	720	420	720
URAM	320	160	320
DSP	2280	1320	2280

1. ダイナミック プラットフォームは、デフォルトでカーネルをカーネルがアクセスするメモリ バンクと同じ SLR に配置します。この制御方法は、『SDAccel 環境ユーザー ガイド』(UG1023) の「カーネルのユーザー指定 SLR 割り当て」セクションを参照してください。
2. 各マップド メモリ チャンネルに約 20K 個の CLB LUT と 20K 個の CLB レジスタが必要です (スタティック領域の bank1 を除く)。SmartConnect ネットワークには最低限 12K 個の CLB LUT および 18K 個の CLB レジスタが必要で、各マップド メモリ チャンネルおよび各計算ユニットに追加リソースが必要です。

表 2: xilinx_kcu1500_dynamic_5_0

エリア	SLR 0	SLR 1
SLR 記述	デバイスの上部、ダイナミック領域およびスタティック領域リソースで共有。	デバイスの上部、ダイナミック領域専用。
ダイナミック領域の Pblock 名	pfm_top_i_dynamic_region_ pblock_dynamic_SLR0	pfm_top_i_dynamic_region_ pblock_dynamic_SLR1
計算ユニット配置構文 ¹	set_property CONFIG.SLR_ASSIGNMENTS SLR0 [get_bd_cells <cu_name>]	set_property CONFIG.SLR_ASSIGNMENTS SLR1 [get_bd_cells <cu_name>]
メモリ チャンネル、システム ポート名	bank0 (4 GB DDR4) bank1 (4 GB DDR4)	bank2 (4 GB DDR4) bank3 (4 GB DDR4)
CLB LUT	264K	325K
CLB レジスタ	529K	651K
ブロック RAM タイル	876	1080

表 2: xilinx_kcu1500_dynamic_5_0 (続き)

エリア	SLR 0	SLR 1
DSP	2217	2760

1. ダイナミック プラットフォームは、デフォルトでカーネルをカーネルがアクセスするメモリ バンクと同じ SLR に配置します。この制御方法は、『SDAccel 環境ユーザー ガイド』(UG1023) の「カーネルのユーザー指定 SLR 割り当て」セクションを参照してください。
2. 各マップド メモリ チャネルに約 20K 個の CLB LUT と 20K 個の CLB レジスタが必要です。SmartConnect ネットワークには最低限 12K 個の CLB LUT および 18K 個の CLB レジスタが必要で、各マップド メモリ チャネルおよび各計算ユニットに追加リソースが必要です。

2017.4 リリースでは、2017.2 と同じ `xocc` インターフェイスとの下位互換性のためで 4.X DSA がサポートされています。5.X DSA には、「SDAccel 移行サマリ」に示されているように `xocc` オプションを変更してください。

2017.4 で実行する DSA にかかわらず、既存の RTL カーネルは 2017.4 を使用して再パッケージする必要があります。

2018.2 以降は、`xocc` で 5.X DSA (および対応する `xocc` オプション) のみがサポートされます。4.X DSA は 2018.2 で廃止予定です。

SDx™ GUI

- Vivado® IDE を SDx から直接起動可能。
 - これにより、経験豊富なハードウェア設計者または Vivado 統合設計環境に精通した設計者が、ハードウェアにインプリメンテーションの変更を加え (詳細なタイミング クロージャなど)、結果を保存できます。
- また、合成済みまたはインプリメント済みの Vivado Design チェックポイント (.dcp) ファイルを Vivado セッションから取り込んで SDx セッションで直接使用して、最初から開始せずに残りのフローを完了できます。
- Vivado セッション中に加えた変更は、SDx にも後続の run で自動的に取り込まれます。
- RTL カーネル ウィザードを、コンパイル済みのカーネルおよびネットリスト (.dcp) に基づくカーネルを含む追加のタイプのパッケージ オプションをサポートするよう改善。
 - データフローは `xocc` の重要な機能 (C/C++ および OpenCL カーネル両方で)。データフローを使用する C/C++/OpenCL デザインに関連する DRC (および詳細な資料) を複数追加。
 - ユーザー C/C++/OpenCL コードでのキー DRC をハイライトする DRC ウィンドウを、SDx GUI の下部、[Console] タブの横に提供。

カーネル パフォーマンスの改善点

- データ転送レートを向上する次の方法を提供。
 - 自動メモリ結合および拡張。データ転送の自動拡張は、for-loop に `nounroll` プラグマを追加することによりディスエーブルにできます。
 - 新しいザイリンクス OpenCL 属性 `xcl_zero_global_work_offset` を使用してメモリ結合を手動で指定。この属性は、`clEnqueueNDRangeKernel` を `global_work_offset` なしで使用する場合に使用可能です。
 - 正しく指定した `global_work_offset` を使用することを強く推奨します。
- ワークグループ サイズを OpenCL セマンティクスに基づいて自動推論。

関数全体のベクター化を NDRange で `vec_type_hint` 属性を使用して指定可能。

- パフォーマンスを向上するため、`vec_type_hint` を使用することを強くお勧めします。
- 関数全体のベクター化を使用すると、ハードウェア インプリメンテーションのサイズが増加することがあります。

パフォーマンスを向上するため、サブ関数を自動的にインライン化。

これは、`noinline` プラグマおよび OpenCL 属性を使用してディスエーブルにできます。

ザイリンクス SDAccel ランタイム

- OpenCL API `clCreateSubDevices` をサポート。
 - 各計算ユニット (CU) にサブデバイスを作成でき、各 CU に複数の独立したコマンド キューを含めることが可能。
 - 各サブデバイスに含めることができるのは 1 つの CU のみ。
- Linux ドライバー サポートを改善
 - ドライバーで Linux DMA_BUF フレームワークが使用されるようになり、すべての Linux デバイスでデータを共有することが可能。
 - Linux SysFS フレームワークを介してデバイス データ (温度、電流など) をエクスポート可能。

RTL カーネルの改善点

- `xocc` コマンド ラインを使用した RTL カーネルのコンパイル時間パラメーター指定をサポート。
- 新しい `xocc` コマンド ライン オプションにより、1 つの RTL カーネル (`.xo` ファイル) を複数のカーネル インスタンスとしてインスタンス化可能。これらの個別のインスタンスは、お互いに独立させてキューに入れることが可能です。
- RTL カーネルをあらかじめコンパイルして SDx で合成を実行しないようにすることにより、SDx のコンパイル フロー時間を短縮。
- RTL カーネルをザイリンクス チェックポイント (`.dcp`) ファイルから作成可能。
- RTL カーネルの暗号化をサポート。

XOCC の改善点

- `--ini_file` オプションを使用して、アドバンス `--xp style` オプションのセットを 1 つのファイルで `xocc` に渡すことが可能 (`xocc.ini` ファイルを使用するのと同様)。
- `--report_dir` オプションにより、SDx run で生成されたレポート ファイルを別のディレクトリにコピー可能。
- `--log_dir` オプションにより、SDx run で生成されたログ ファイルを別のディレクトリにコピー可能。
- `--temp_dir` オプションにより、一時ファイルの生成にユーザー指定のディレクトリを使用可能。
- `--interactive` オプションにより、適切なプロジェクトを読み込んでいる場合に `xocc` 環境内から Vivado を起動可能。
- `--reuse_synth` オプションにより、合成済み Vivado デザイン チェックポイント (`.dcp`) ファイルを取り込み、SDx フローのディレクトリで使用してインプリメンテーションと `xclbin` の生成を実行可能。

- `--reuse_impl` オプションにより、インプリメント済み Vivado デザイン チェックポイント (.dcp) ファイルを取り込み、SDx フローのディレクトリで使用して `xclbin` を生成可能。
- `--remote_ip_cache` オプションにより、ユーザー指定の IP キャッシュ ロケーションを使用可能。これは、SDx フローの繰り返し実行時間を短縮できます。
- `--user_ip_repo_paths` オプションにより、SDx で追加の読み取り専用 IP キャッシュ ロケーションとカスタム IP 定義を使用可能。
- `--no_ip_cache` オプションを使用して IP キャッシュの使用をすべてオフにすることが可能。これは通常、デバッグ目的以外で使用することはお勧めしません。

プロファイル機能

- カーネルのプロファイル インストルメンテーションを `xocc` コンパイル オプション `-profile_kernel` を使用してイネーブルにすることが可能。
- `sdx_analyze` ユーティリティを使用してプロファイル レポートを生成。
- プロファイル サマリ レポートの改善点:
 - データ転送の表に、カーネル引数および DDR バンクを含む計算ユニット/ポートごとの情報を表示。
 - 計算ユニットの表に、計算ユニットごとのクロック周波数をレポート。

デバッグ機能

- カーネル デバッグをハードウェア エミュレーションの GDB および TCF を介してサポート。これにより、次が可能になります。
 - カーネルの実行の中間点で開始および終了。
 - カーネル引数とグローバル メモリの両方を確認。
- アプリケーション デバッグ: 次の新しい gdb 拡張により、デバッグ情報を向上。
 - `xprint kernel`: 保留中の NDRange イベントとその引数をすべて表示
 - `xprint all`: 有効な OpenCL オブジェクトをすべて表示
 - `xstatus all`: プラットフォームにインスタンス化した IP の内容を表示
 - 新しい `xocc` コマンド オプション `-dk` により、AXI プロトコル違反をデバッグするため Light Weight Protocol Checker IP を挿入。

エミュレーション機能

- `XCL_EMULATION_MODE`: `XCL_EMULATION_MODE` 環境変数の値に `sw_emu` または `hw_emu` が必要になりました。`XCL_EMULATION_MODE` 環境変数を `sw_emu` に設定すると、アプリケーション実行がソフトウェア エミュレーション モードに変更され、`hw_emu` に設定すると、ハードウェア エミュレーション モードがイネーブルになります。`XCL_EMULATION_MODE` 変数の設定をしなければ、エミュレーションをディスエーブルにできません。
- 出力アクセスおよび無効な読み出しまたは書き込み操作 (読み出し専用デバイスに書き込むなど) に対するメモリ チェックをソフトウェア エミュレーションに提供。

SDAccel 移行サマリ

次の表に、2017.4 リリースを使用する際に必要な既存のフローおよびスクリプトへの変更を示します。

表 3: SDAccel 移行サマリ

エリア	2017.4 の動作	2017.4 で必要なアップデート
DSA	2017.4 では主に 5.x DSA がサポートされます。前のリリースからの既存の 4.X DSA は、2017.2 xocc オプションと共に使用できます。	ザイリンクスでは、5.x DSA プラットフォームにアップデートすることを強くお勧めします。プラットフォームは <code>--platform</code> オプションを使用して指定してください。
	区切り文字として、コロン (:) ではなくアンダースコア (_) が使用されます。	プラットフォーム名では、アンダースコア (_) が区切り文字として使用されます。これには、 <code>xocc</code> コマンドの <code>--platform</code> オプションを使用して DSA 名をアップデートする必要があります。
xocc オプション	シングル ステップ コンパイルはサポートされなくなっています。makefile フローを使用します。	カーネルは、2 つの <code>xocc</code> コマンドを個別に実行し、 <code>-compile (-c)</code> オプションを使用した後、 <code>-link (-l)</code> オプションを使用してリンクする必要があります。SDx™ GUI を使用する場合は、 <code>xocc</code> オプションをこの形式で使用した正しい makefile が自動的に生成されます。
	複数の DDR バンクを使用する場合、 <code>--xp map_connect</code> オプションは削除され、 <code>-sp</code> オプションを使用したシンプルな構文に置き換えられています。	システム ポートを指定するには、新しい <code>xocc</code> コマンド オプション <code>-sp</code> を使用します。 次に、既存の <code>map_connect</code> オプションを 2017.4 用に変更する例を示します。 (2017.2) <code>--xp</code> <pre>misc:map_connect=add.kernel.krnl_idct_1.M_AXI_GMEM.core.OCL-REGION_0.M00_AXI</pre> (2017.4) <code>--sp</code> <pre>krnl_idct_1.m_axi_gmem:bank0</pre>
	インクルード ファイルの場所が変更されています。	ザイリンクスの <code>include</code> ファイルを使用する場合は、 <code>include</code> パス名を <code>-I\$(XILINX_SDACCEL)/Vivado_HLS/include/</code> から <code>-I\$(XILINX_SDACCEL)/include</code> に変更します。

表 3: SDAccel 移行サマリ (続き)

エリア	2017.4 の動作	2017.4 で必要なアップデート
ホスト コード	Ubuntu OS 上でホスト コードをコンパイルします。	<p>Ubuntu OS 上でホスト コードをコンパイルする際は、<code>-std=c++14</code> を使用する必要があります。</p> <p>システム ヘッダー ファイル <code>sys/cdefs.h</code> は、<code>/usr/include</code> ディレクトリではなく <code>/usr/include/x86_64-linux-gnu</code> ディレクトリにあります。このため、<code>include</code> パスをアップデートする必要がある場合があります。</p>
	複数の <code>xclbin</code> ファイルを使用する場合は、2 つ目のファイルを読み込む前に最初のファイルを解放する必要があります。	<code>xclbin</code> ファイルが複数ある場合、次の <code>xclbin</code> ファイルを読み込む前に、 <code>clReleaseProgram()</code> および <code>clReleaseKernel()</code> を使用して現在の <code>xclbin</code> ファイルを解放する必要があります。
	複数のカーネルが同じ DDR にアクセスします。	<p>同じ DDR にアクセスするカーネルが複数あり、それらが循環依存関係にある場合、デッドロック状態が発生する可能性があります。</p> <p>異なる DDR にアクセスするようにするか、カーネル間が同期するようにして、カーネル間の依存性を切断する必要があります。</p> <p>たとえば 2 つのカーネルがあり、4 つの読み出しおよび 4 つの書き込み操作 (それぞれ 2 つの読み出しと 2 つの書き込み) でそれぞれが DDR バンクにアクセスする場合、2 つのカーネルの読み出しに 1 つの DDR を使用し、同じカーネルの書き込みに 1 つの DDR を使用する必要があります。</p>
	<code>xocc</code> でリンクされたサイズを超えるメモリにアクセスできた可能性があります。	ホスト コードはカーネルで使用可能なメモリ サイズにのみアクセス可能で、デザインでバンクが使用されないことに依存しています。 <code>-p</code> を使用して設定されていない DDR インターフェイスは <code>xocc</code> の最適化で削除されるので、割り当てられているサイズを超えるメモリにアクセスする既存のデザインは機能しません。
	共有ライブラリ	<p>SDx ホスト コードで HLS math ライブラリを使用する場合は、ダイナミック ライブラリを見つけるために、さらにリンケージ情報を <code>makefile</code> に追加する必要があります。次に例を示します。</p> <pre>\$(XILINX_VIVADO)/ lnx64/lib/csim -lhlsmc+ +-GCC46</pre>

表 3: SDAccel 移行サマリ (続き)

エリア	2017.4 の動作	2017.4 で必要なアップデート
C/C++ OpenCL カーネル コード	パイプ名には、すべて小文字を使用する必要があります。	パイプに使用する変数名には、すべて小文字を使用する必要があります。たとえば、次の例では「pipe int inFifo」は使用できず、すべてを小文字にする必要があります。 <pre>pipe int infifo _attribute__((xcl_reqd_pipe_depth(16)));</pre>
	デフォルトで OpenCL 2.0 画像関数がサポートされていました。現在では、これらにコンパイラ オプションが必要です。	OpenCL 2.0 画像関数を使用するとエラーが発生するので、xocc コマンド オプション --xp param:compiler.version=31 を使用して 3.1 コンパイラをイネーブルにする必要があります。
	メモリ結合を向上し、メモリ帯域幅を増加して、カーネル実行を高速化します。	メモリ帯域幅を増加するためにループのベクター化が自動的に実行され、リソース使用量が増加する場合があります。影響を受けるループに次を使用すると、この機能をオフにできます。 <pre>#pragma nounroll</pre>
	xcl_dependence 属性をする場合、3.1 バージョンのコンパイラが必要です。	xcl_dependence 属性を使用すると、エラーが発生します。この属性を削除するか、xocc コマンド オプション option--xp param:compiler.version=31 を使用して 3.1 コンパイラをイネーブルにする必要があります。
	OpenCL カーネル インターフェイスのデフォルトのデータ幅。	OpenCL カーネル AXIM インターフェイスのデフォルトのデータ幅は、512 ビットに変更されています。これによりカーネルのパフォーマンスは向上するはずですが、一部のデザインでタイミング クロージャに影響する可能性があります。
	インターフェイス バンドル オプションを使用する C および C++ カーネル。	バンドル オプションで指定するバンドル名には、次の AXI および AXI4-Lite インターフェイスの例で示すように、小文字を使用する必要があります。どちらの例でも、名前 bar は小文字です。 <pre>#pragma interface s_axilite port=foo bundle=bar #pragma interface m_axi port=foo bundle=bar</pre>

表 3: SDAccel 移行サマリ (続き)

エリア	2017.4 の動作	2017.4 で必要なアップデート
RTL カーネル	RTL カーネル ウィザードを使用してパッケージされた RTL カーネル XO ファイル	RTL カーネル ウィザードを再実行し、2017.4_sdx から同じオプションを選択します。Vivado プロジェクトでは、最上位の AXI インターフェイス ポートに適合した RTL を再度取り込み、カーネルを再パッケージして新しい XO ファイルを生成します。これで、2017.4 ツール フロー およびランタイムに必要な最新のメタ データが RTL カーネルに確実に含まれるようになります。
	RTL カーネル ウィザードを使用せずにユーザーがパッケージした RTL カーネル	<p>手動でパッケージした RTL カーネルは、2017.4 で機能するように再パッケージする必要があります。</p> <p>必要な新しいメタ データを挿入するため、次のコマンドが追加されています。</p> <pre>set_property sdx_kernel true [ipx::current_core] set_property sdx_kernel_type rtl [ipx::current_core]</pre> <p>各 IP に、IP と IP 上のすべての AXI インターフェイスを駆動する ap_clk という名前のプライマリ クロックが必要です。オプションでセカンダリ クロックを含めることもできますが、ap_clk_2 という名前にする必要があります。プライマリ クロックにはリセット ポート ap_rst_n が関連付けられており、セカンダリ クロックが存在する場合は、それに対応するリセット ポート ap_rst_n_2 が必要です。</p> <p>RTL カーネルを RTL カーネル ウィザードを使用せずに手動でパッケージする場合は、kernel.xml と component.xml (大文字/小文字が区別される) の間でポート インターフェイス名が一貫している必要があります。</p>
	パイプを含む RTL カーネル	<ul style="list-style-type: none"> kernel.xml 内のストリーム パイプ名には、すべて小文字を使用する必要があります。大文字を使用したり、大文字と小文字を混ぜることはサポートされません。 グローバル BRAM パイプ接続はサポートされません。
	手動で編集された RTL kernel.xml	宣言された各ポート (port/@name) は、カーネル属性 (arg/@port) に等しいものからマップする必要があります。

表 3: SDAccel 移行サマリ (続き)

エリア	2017.4 の動作	2017.4 で必要なアップデート
プロファイリング	プロファイリング ハードウェアを任意の DSA に動的に追加することが可能となり、プラットフォームであらかじめビルドしておく必要はなくなりました。これには、 <code>xocc</code> コマンド オプションをアップデートする必要があります。	プロファイリングは、プロファイリングデータの収集とデータの保存の 2 段階プロセスで実行されます。 カーネルを <code>-profile_kernel</code> オプションを使用してコンパイルすることにより、プロファイル インストールメンテーションをイネーブルにします。 <code>sdaccel.ini</code> ファイルで <code>profile=true</code> を設定し、プロファイルデータを収集します。
	<code>sdx_analyze profile <options></code> を使用してプロファイル サマリ レポートを生成します。	アプリケーション タイムライン トレース レポートを生成するのに使用されていた <code>sda2wdb</code> は <code>sdx_analyze</code> に置き換えられています。
	<code>sdx_analyze trace <options></code> を使用してアプリケーション タイムライン トレース レポートを生成します。	アプリケーション タイムライン トレース レポートを生成するのに使用されていた <code>sda2wdb</code> は <code>sdx_analyze</code> に置き換えられています。
	ザイリンクス PCIe プラットフォームのステータスをチェックする <code>sdxsyschk</code> コーティリティは使用できません。	廃止予定で、 <code>xbsak scan</code> コマンドに置き換えられています。
	プロファイル レポート (HTML) は自動的に生成されません。	HTML 形式のプロファイル サマリを生成するには、 <code>sdx_analyze profile --input sdaccel_profile_summary.csv --format html</code> を実行します。

SDx 環境の概要

2017.4 SDx™ 環境ソフトウェア リリースには、Zynq® UltraScale+™ MPSoC および Zynq®-7000 AP SoC ファミリ用の SDSoC™ 開発環境と、データ センターおよび PCIe® ベースのアクセラレータ システム用の SDAccel™ 開発環境が含まれます。これらの環境は共通のインストーラーを使用しますが、ライセンスは別になっています。すべての SDx 環境には、ターゲット デバイスのプラグラムおよびカスタム ハードウェア プラットフォームの開発のために Vivado® Design Suite が含まれます。

SDSoC 概要

SDSoC™ (Software-Defined Development Environment for System-on-Chip) 環境は、Zynq®-7000 AP SoC および Zynq UltraScale+ MPSoC プラットフォームを使用してヘテロジニアス エンベデッド システムをインプリメントするための Eclipse ベースの統合設計環境 (IDE) です。SDSoC 環境では、ソフトウェア エンジニアおよびシステム アーキテクト用に、使いやすい Eclipse ベースの IDE を使用したエンベデッド C/C++ アプリケーション開発環境と、ヘテロジニアス Zynq-7000 AP SoC および Zynq UltraScale+ MPSoC 開発用の包括的なデザイン ツールが提供されています。

SDSoC 環境には、プログラマブル ロジックでの自動ソフトウェア アクセラレーションや、システム接続の自動生成などを実行する、フルシステム最適化 C/C++ コンパイラが含まれます。SDSoC 環境内のプログラミング モデルは、ソフトウェア エンジニアが簡単に理解できるように設計されています。アプリケーションは C/C++ コードで記述され、プログラマがターゲット プラットフォームとハードウェアにコンパイルするアプリケーション内の関数のサブセットを特定します。この後、SDSoC システム コンパイラによりアプリケーションがハードウェアとソフトウェアにコンパイルされ、ファームウェア、オペレーティング システム、アプリケーション実行ファイルを含むブート イメージを含めた完全なエンベデッド システムが Zynq デバイスにインプリメントされます。

SDSoC 環境では、C/C++ 関数の Zynq デバイス内の ARM CPU だけでなくプログラマブル ロジック ファブリックへのクロスコンパイルおよびリンクを含め、ソフトウェア抽象層を増加することによりハードウェアが抽象化されます。プログラマブル ハードウェアで実行するプログラム関数のユーザー仕様に基づいて、プログラム解析、タスク スケジューリング、プログラマブル ロジックおよびエンベデッド CPU へのバインディングが実行されるほか、ハードウェアおよびソフトウェア コード生成により、ハードウェアとソフトウェア コンポーネント間の通信および連携が調整されます。

また、SDSoC では、Zynq および Zynq UltraScale+ MPSoC デバイスのプログラマブル ロジックをターゲットにする OpenCL カーネルを使用し、ザイリンクスの提供するベース開発プラットフォームで OpenCL™ アプリケーションもサポートされます。

SDAccel の概要

SDAccel™ は、ザイリンクス FPGA ベースのアクセラレータ カードをターゲットにした OpenCL™ アプリケーション用の開発環境です。この環境では、高度な FPGA 設計の経験がなくても、インシステム プロセッサと FPGA ファブリックを同時にプログラムできます。アプリケーションは、OpenCL C で記述されたホスト プログラムと C、C++、OpenCL C、または RTL で記述された計算カーネルのセットとして取り込まれます。

ハードウェア要件

SDSoC ハードウェア要件

2017.4 SDSoC™ 環境リリースには、次の開発ボードのサポートが含まれます。

- Zynq®-7000 AP SoC を使った ZC702、ZC706、および ZedBoard 開発ボード
- Zynq® UltraScale+™ MPSoC を使った ZCU102 開発ボード。

その他のプラットフォームは、パートナーから入手できます。また、SDSoC プラットフォーム ユーティリティを使用すると、どの Zynq および Zynq UltraScale+ MPSoC カスタム ボードでもターゲットにできます。詳細は、SDSoC 開発者ゾーンのウェブ ページ <https://japan.xilinx.com/products/design-tools/software-zone/sdsoc.html> を参照してください。

ボードからの UART 出力を監視するには、mini-USB ケーブルも必要です。

SDAccel ハードウェア要件

SDAccel 環境には、次のハードウェアが必要です。

- アクセラレーション カード(次のいずれかを使用):
 - XCKU115-FLVB2104-2-E FPGA に基づいたザイリンクス Kintex UltraScale FPGA KCU1500 リコンフィギャラブル アクセラレーション カード。
 - XCVU9P-L2FSGD2104E FPGA に基づいたザイリンクス Virtex UltraScale FPGA VCU1525 リコンフィギャラブル アクセラレーション カード。
- ホスト コンピューター: アクセラレーション カードをホストするデスクトップ コンピューター。ホスト コンピューターには、次が含まれている必要があります。
 - PCIe Gen3 X8 スロットを含むマザーボード
 - 16 GB RAM
 - 100 GB のディスク空き容量
- プログラミング コンピューター: FPGA をプログラムするために提供されている Vivado Design Suite 2017.4 を実行するノート型またはデスクトップ コンピューター。

- プログラミング コンピューターをアクセラレーション カードに接続するためのザイリンクス プラットフォーム ケーブル USB 2、パーツ番号 HW-USB-II-G。詳細は、『プラットフォーム ケーブル USB II データシート』(DS593) を参照してください。
- その他のプラットフォームは、パートナーから入手できます。詳細は、SDAccel 開発者ゾーンのウェブ ページ <https://japan.xilinx.com/products/design-tools/software-zone/sdaccel.html> を参照してください。

ソフトウェア要件

SDx™ 開発環境は、Linux および Windows オペレーティング システムの両方で実行できます。サポートされるオペレーティング システムは、次のとおりです。

- Windows 7 および 7 SP1 Professional (64 ビット) (SDSoC のみ)
- Windows 10 Professional (64 ビット) (SDSoC のみ)
- Linux サポート
 - Red Hat Enterprise Workstation/Server 7.2-7.3 (64 ビット)
 - Red Hat Enterprise Workstation 6.7 および 6.8 (SDSoC のみ)
 - Red Hat Enterprise Workstation 6.9 (64 ビット)
 - CentOS 6.9、CentOS 7.3 (64 ビット) (SDAccel)
 - Ubuntu Linux 16.04.2 LTS (64 ビット)

SDSoC インストールについて

SDSoC™ のインストールには、次が含まれます。

- SDSoC 環境 (Eclipse/CDT ベースの GUI、高位システム コンパイラ、および ARM GNU ツールチェーンを含む)
- Vivado® Design Suite System Edition (Vivado 高位合成 (HLS) およびザイリンクス ソフトウェア開発キット (SDK) を含む)

SDSoC 環境には、ザイリンクス ソフトウェア開発キット (SDK) に含まれるのと同じ GNU ARM ツールチェーンが含まれるほか、SDSoC 環境で使用されるその他のツールも提供されています。SDSoC 環境のセットアップ スクリプトを使用すると、このツールチェーンを使用するように PATH 変数が設定されます。

SDSoC インストールの詳細は、次を参照してください。

- 提供されるツールチェーンには、32 ビットの実行ファイルが含まれているので、Linux ホスト OS のインストールには 32 ビットの互換ライブラリが含まれている必要があります。
- RHEL 6 および 7 64 ビット x86 Linux インストールには 32 ビット互換ライブラリが含まれおらず、別に追加する必要がある可能性があります。詳細は、<https://access.redhat.com/site/solutions/36238> を参照してください。
- RHEL、32 ビット互換ライブラリは、ルート アクセス権のあるスーパー ユーザー (またはルート管理者) になって、`yum install glibc.i686` コマンドを実行するとインストールできます。

- Ubuntu、32ビット互換ライブラリは、ルートアクセス権のあるスーパーユーザー(またはルート管理者)になって、次のコマンドを実行するとインストールできます。詳細は、[SDSoC 開発環境の機能](#)を参照してください。

```
sudo dpkg --add-architecture i386
sudo apt-get update
sudo apt-get install libc6:i386 libncurses5:i386 libstdc++6:i386
sudo apt-get install g++-multilib
sudo apt-get install libgtk2.0-0:i386 dpkg-dev:i386
sudo ln -s /usr/bin/make /usr/bin/gmake
```

- ツールチェーンのバージョンは、`arm-linux-gnueabi-g++ -v` コマンドを実行すると表示できます。
- シェルウィンドウの最後の行に、GCC version 4.9.2 20140904 (prerelease)(crosstool-NG linaro-1.13.1-4.9-2014.09 - Linaro GCC 4.9-2014.09) と表示されるはずですが、

SDAccel インストール

SDAccel™ 環境は Linux オペレーティングシステムでしか実行できず、Windows のサポートはありません。サポートされるのは、RedHat Enterprise Linux、CentOS 6.9 および 7.3 64 ビット、および Ubuntu 16.04 64 ビットです。

CentOS/RHEL 7.3 (6.9 もサポートあり) パッケージリスト

<https://fedoraproject.org/wiki/EPEL> の説明に従って EPEL リポジトリをインストールしてください。また、yum インストールコマンドを使用すると、次のパッケージがインストールされます。

- ocl-icd
- ocl-icd-devel
- opencl-headers
- kernel-headers
- kernel-devel
- gcc-c++
- gcc
- gdb
- make
- opencv
- libjpeg-turbo-devel
- libpng12-devel
- python(gdb capability in SDAccel works with python 2.7.5)
- git version 1.8.3.1 or higher
- unzip
- dmidecode
- redhat-lsb

- kernel-headers-\$(uname -r)
- compat-libtiff3
- libstdc++-static
- libtiff-devel
- pciutils
- strace
- perl
- boost-devel
- boost-filesystem
- gnuplot
- cmake
- lm_sensors

Ubuntu 16.04 パッケージ リスト

apt-get install コマンドを使用すると、次のパッケージがインストールされます。

- ocl-icd-libopencl1
- opencl-headers
- ocl-icd-opencl-dev
- linux-headers
- linux-libc-dev
- g++
- gcc
- gdb
- make
- libopencv-core
- libjpeg-dev
- libpng-dev
- python
- git version 1.8.3.1 or higher
- dmidecode
- lsb
- unzip
- linux-headers-\$(uname -r)
- libstdc++ -static
- opencv
- libtiff5-dev
- pciutils
- strace
- perl

- libboost-dev
- libboost-filesystem-dev
- gnuplot
- cmake
- lm-sensors

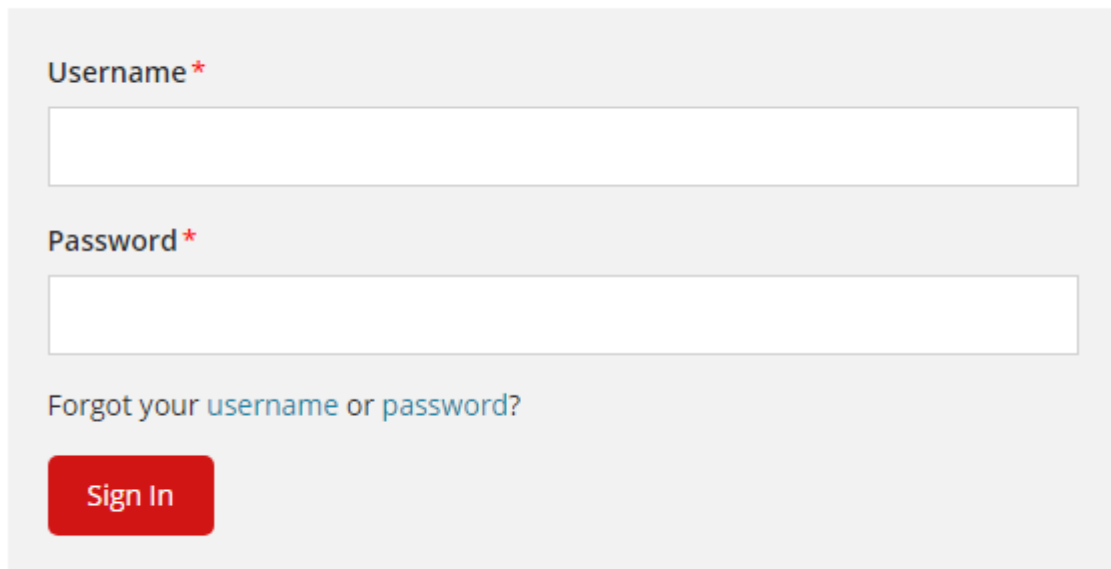
ザイリンクス ライセンス サイトでのライセンスの取得

SDx™ 開発環境のライセンスを取得する方法は、次のとおりです。

1. ザイリンクス ライセンス ウェブサイト <https://japan.xilinx.com/getproduct> にサインインします。次の図を参照してください。

図 1: ザイリンクス ライセンス サイト - サインイン ページ

Sign In



Username *

Password *

[Forgot your username or password?](#)

Sign In

[New to Xilinx? Create your account](#)

SDAccel™ - ザイリンクス OpenCL™ 設計環境のライセンスを初めて取得する場合は、ザイリンクスの販売代理店に連絡し、SDAccel ライセンス ウェブサイトにアクセスできるようにしておく必要があります。

SDSoC には 60 日間の評価ライセンスが付いており、使用可能なライセンスのリストに表示されるはずですが、

2. アカウント ドロップダウン リストで [XILINX - SDSoC Environment] または [SDAccel Environment] をクリックします。

注記: これは、SDSoC™ または SDAccel ライセンスを購入または取得した場合にのみ表示されます。

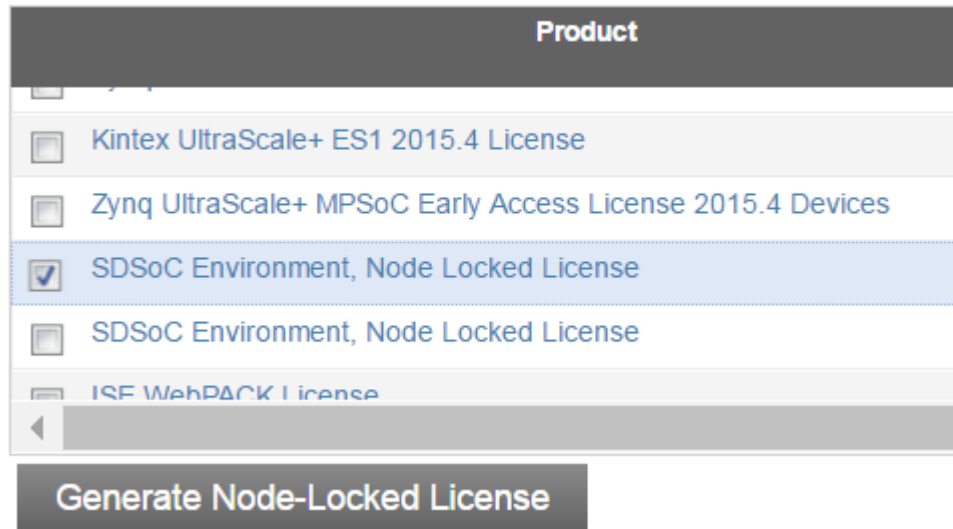


重要: SDSoC を購入前でも、最初に使用するとき SDSoC の 60 日間評価ライセンスが表示されるはずですが。

3. [Certificate Based Licenses] メニューから [SDSoC Environment, Node-Locked License] または [SDAccel Environment, Node-Locked License] をクリックします。

図 2: [Certificate Based Licenses] メニュー

Certificate Based Licenses



4. [Generate node-locked license] をクリックします。
5. [License Generation] 画面でホスト ID を入力して [Next] をクリックします。
6. ライセンスのホスト ID が正しいことを確認したら、[Next] をクリックします。
7. 使用許諾契約を確認したら [Accept] をクリックします。

xilinx.notification@entitlenow.com からライセンス ファイルが添付された電子メールが送付されます。

8. XILINXD_LICENSE_FILE 環境変数をシステムのライセンス ファイルのディレクトリに指定します。

SDx 環境のインストール

この章では、SDSoC™ 環境および SDAccel™ 環境のインストール プロセスについて説明します。

ツールのインストール準備

注記: インストールを開始する前に、次の手順を終了しておく必要があります。

1. システムが次の必要条件を満たしているかどうか確認してください。
 - [ソフトウェア要件](#)
 - [SDSoC ハードウェア要件](#)
 - [SDAccel ハードウェア要件](#)
2. インストール時間を削減するため、アンチウイルス ソフトウェアをオフにします。
3. インストールを開始する前に、すべてのプログラムを終了します。

SDSoC および SDAccel のインストール

SDSoC™ および SDAccel™ のインストールには、次の 2 つの方法があります。どちらのインストール方法も[ザイリンクス ダウンロード ウェブサイト](#)から実行できます。

注記: SDSoC と SDAccel のインストーラーは別々になっています。使用する製品のインストーラーを起動すると、デバイスが前もって選択されます。

注記: インストールには、スタンドアロンと SDx インストールしかなく、Vivado® バージョンが別にあるわけではありません。別の xsetup インストーラー (SDx™ 用と Vivado 用がある) を使用すると、インストールの内容が異なります。SDx xsetup インストーラーを実行すると、2017.4 Vivado xsetup インストーラーを実行した場合と同じ Vivado がインストールされ、提供されるオプションは Vivado System Edition と同じになります。SDx xsetup インストーラーを使用すると、Vivado System Edition バージョンが含まれます。

ウェブ インストーラーの使用

ザイリンクスでは、ウェブ インストーラーをお勧めしています。

ウェブ インストーラーを使用すると、インストールするものだけを選んで必要なデータのみがダウンロードされるようにできます。途中でネットワーク エラーになった場合は、最初からやり直すのではなく、最後に停止したところから再開されます。

注記: インストーラーによってあらかじめ選択されるデバイスは異なります。SDAccel™ 専用ウェブ インストーラーの場合は 7 シリーズ、UltraScale™、UltraScale+™、SDSoC 専用インストーラーの場合は Zynq®-7000、UltraScale+ MPSoC が選択されており、共通 SDx SFD (シングルファイル ダウンロード) イメージの場合はデバイスは何も選択されていません。

フル インストール ファイルのダウンロードおよびインストール

フル製品インストールをダウンロードした場合は、ファイルを解凍して、xsetup (Linux の場合) または xsetup.exe (Windows の場合、ただし SDAccel™ には使用不可) を実行します。

ウェブ インストーラー クライアントをダウンロードした場合は、ダウンロードしたファイルを実行します。インストール タイプの選択画面が表示されたら、通常のザイリンクス ログイン ID を使用してログインします。

ダウンロードした tar.gz ファイルを解凍するには、7-zip または WinZip (v.15.0 以降) の使用をお勧めします。

- [Download and Install Now] を選択すると、次の画面で特定のツールおよびデバイス ファミリーを選択でき、その選択に関連するファイルのみがダウンロードされ、インストールされます。ログイン ID を入れたら、従来のウェブベースのインストールか、フル イメージのダウンロードかを選択します。
- [Download Full Image] を選択した場合は、ダウンロード ディレクトリをユーザーが指定して、Windows のみか Linux のみか、両方のオペレーティング システムをサポートするインストールかを選択する必要があります。[Download Full Image] を選択した場合は、これ以上オプションを選択する必要はなく、ダウンロード ディレクトリから xsetup アプリケーションを実行して別々にインストールを実行する必要があります。

バッチ モード インストール フロー

インストーラーは、バッチ プロセスで実行できます。標準的なエディションを実行するには、インストール ディレクトリを指定するか、インストーラーにインストール ディレクトリと、どのツール、デバイス、オプションをインストールするか伝えるコンフィギュレーション ファイルを取得する必要があります。インストーラーには、よく使用されるコンフィギュレーションに基づいてリファレンス オプション ファイルを生成できるモードがあるので、インストールをさらに編集してカスタマイズできます。

このリファレンスは各四半期リリースごとに生成して、新しいデバイス、オプションまたはその他の変更点がユーザーのオプション ファイルに含まれるようにすることをお勧めします。

バッチ モードの使用を開始するには、まずコマンド シェルを開いて、抽出したインストーラーを保存したディレクトリに移動します。

Windows の場合、管理者権限でコマンド ウィンドウを開き、次のオプションを使用した xsetup.exe ではなく、\bin ディレクトリの xsetup.bat ファイルを実行します。

コンフィギュレーション ファイルの生成

1. xsetup -b ConfigGen を実行します。

これで次のメニューが表示されるインタラクティブ モードになります。SDSoC™ および SDAccel™ 開発環境エディションの場合は、SDx™ IDE を選択します。

2. 選択すると、コンフィギュレーション ファイルのディレクトリとファイル名が表示され、インタラクティブ モードが終了します。

次は、コンフィギュレーション ファイルの例です。

```
Edition=SDx Development Environments

# Path where Xilinx software will be installed.
Destination=/opt/Xilinx

# Choose the Products/Devices the you would like to install.
Modules=Built-in Platforms and associated devices for SDSoC:1,Zynq
UltraScale+ MPSoC:1,Virtex UltraScale+ HBM ES:0,Zynq-7000:1,Kintex
UltraScale+ ES:0,Kintex UltraScale+:1,ARM Cortex-A53:1,Spartan-7:1,ARM
Cortex-A9:1,ARM Cortex R5:1,Virtex UltraScale+ ES:0,System Generator
for DSP:0,Artix-7:1,Built-in Platforms and associated devices for
SDAccel:1,DocNav:1,Kintex-7:1,Virtex UltraScale+:1,Model
Composer:0,Zynq UltraScale+ RFSoc ES:0,Kintex UltraScale:1,Engineering
Sample Devices for Custom Platforms:0,Virtex UltraScale:1,Zynq
UltraScale+ MPSoC ES:0,MicroBlaze:1,Virtex-7:1

# Choose the post install scripts you'd like to run as part of the
finalization step. Please note that some of these scripts may require
user interaction during runtime.
InstallOptions=Acquire or Manage a License Key:0,Enable WebTalk for SDK
to send usage statistics to Xilinx:1,Enable WebTalk for Vivado to send
usage statistics to Xilinx (Always enabled for WebPACK license):1

## Shortcuts and File associations ##
# Choose whether Start menu/Application menu shortcuts will be created
or not.
CreateProgramGroupShortcuts=1

# Choose the name of the Start menu/Application menu shortcut. This
setting will be ignored if you choose NOT to create shortcuts.
ProgramGroupFolder=Xilinx Design Tools

# Choose whether shortcuts will be created for All users or just the
Current user. Shortcuts can be created for all users only if you run
the installer as administrator.
CreateShortcutsForAllUsers=0

# Choose whether shortcuts will be created on the desktop or not.
CreateDesktopShortcuts=1

# Choose whether file associations will be created or not.
CreateFileAssociation=1
```

コンフィギュレーション ファイルの各オプションは GUI のオプションと同じで、値 1 はそのオプションが選択されていることを、値 0 はそのオプションが選択されていないことを示します。

注記: このコンフィギュレーション ファイルでは、デフォルトではインストールされるデバイスは選択されていません (すべてのデバイスの値が 0)。デバイスをインストールするには、そのデバイスの値を 1 に変更する必要があります。

インストーラーの実行

ここまででユーザーのインストール プリファレンスを反映するコンフィギュレーション ファイルを編集したので、次はインストーラーを実行します。インストーラー コマンド ラインの一部として、[ザイリンクスの使用許諾契約](#)および[サードパーティの使用許諾契約](#)を承認し、WebTalk の契約条件を理解していることを示す必要があります。

WebTalk 契約条件

WebTalk 契約条件の内容は次のとおりで、インストーラーを実行中に許諾する必要があります。

[I agree] をクリックすることで、上記の WebTalk に関する契約条件のセクション 13 を読み、<https://japan.xilinx.com/webtalk> の WebTalk FAQ を読む機会が与えられたことを確認します。セクション 13(c) に記述される特定の条件が当てはまる場合は、WebTalk をオフにできることを理解します。条件が該当しない場合は、ソフトウェアをアンインストールするか、インターネットに接続されていないマシンでソフトウェアを使用することで、WebTalk をディスエーブルにできます。該当条件を満たすことができない場合、またはこのような情報の伝達を回避するための適切な手順を踏めない場合は、セクション 13(b) に記述された目的でセクション 13(a) で記述された情報をザイリンクスが収集することに同意します。

コマンドラインを使用する場合に上記のそれぞれの項目の同意を示すには、コマンドライン オプションの `-a` または `--agree` を使用します。上記の 1 つがリストに含まれない場合、または `agree` オプションが指定されない場合、エラーメッセージが表示されてインストーラーが停止して、インストールができません。

バッチ モード インストール

これは、コンフィギュレーション ファイルを使用した典型的な新規インストールのコマンドライン例です。

```
xsetup --agree XilinxEULA,3rdPartyEULA,WebTalkTerms
--batch Install --config install_config.txt
```

ザイリンクスのデフォルトの Edition コンフィギュレーションの 1 つを使用する場合、`--config` オプションを指定する必要はありませんが、デスティネーション ディレクトリはコンフィギュレーション ファイルに含まれるので、コマンドラインでこれを指定する必要があります。

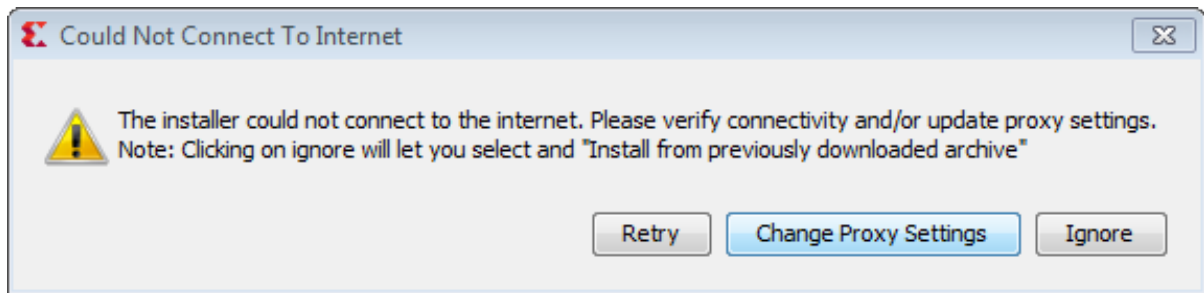
```
xsetup --agree 3rdPartyEULA,WebTalkTerms,XilinxEULA
--batch Install --edition "SDx Development Environments" --location
"C:\Xilinx"
```

上記のコマンドは、指定した Edition のデフォルトのコンフィギュレーション オプションを使用しています。デフォルト コンフィギュレーション オプションを確認するには、上記のように `-b ConfigGen` モードを使用してください。SDx インストーラーのバッチ モードでは、アンインストールおよびアップグレード (ツールおよびデバイスの追加) を実行することもできます。 `xsetup -h` or `xsetup --help` を実行すると、インストーラー バッチ モードですべてのオプションをリストできます。

接続の確認

インストーラーでは、Windows のシステム プロキシ設定を使用してインターネットに接続されます。接続には、[コントロール パネル] → [Internet Options] での設定が使用されます。Linux ユーザーの場合は、Firefox ブラウザーのプロキシ設定が接続に使用されます。

図 3: Vivado Design Suite のインストール - 接続



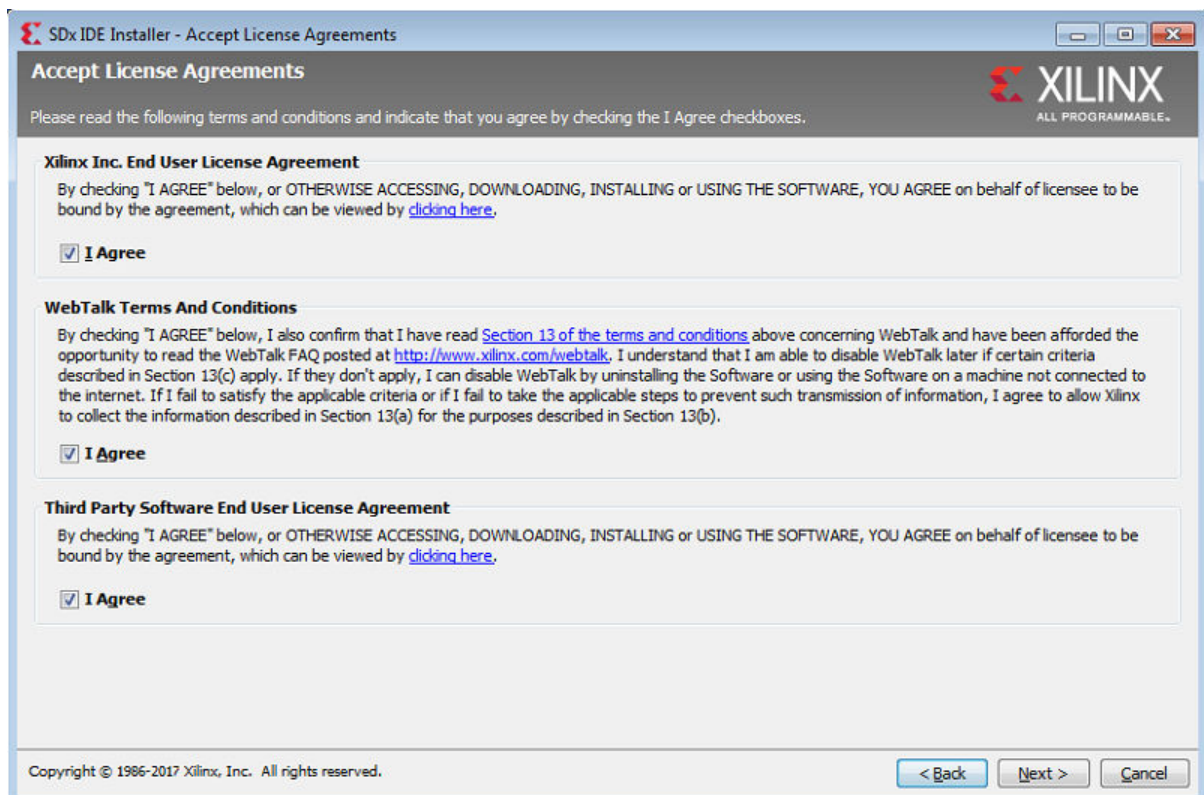
接続に問題がある場合は、次を確認してください。

1. 別のプロキシ設定を使用する場合は、[Manual Proxy Configuration] オプションで設定を指定します。
2. 会社のファイアウォールにユーザー名とパスワードを使用したプロキシ認証が必要かどうかを確認してください。必要であれば、上記のダイアログ ボックスで [Manual Proxy Configuration] で設定します。
3. Linux ユーザーが Firefox ブラウザーで [Use system settings] または [Auto-detect settings] のいずれかを選択した場合は、インストーラーでプロキシを手動で設定する必要があります。

使用許諾契約の受諾

インストールを続行する前に、使用許諾契約をお読みください。契約条件を許諾できない場合は、インストールをキャンセルしてザイリンクスまでお問い合わせください。

図 4: 使用許諾契約

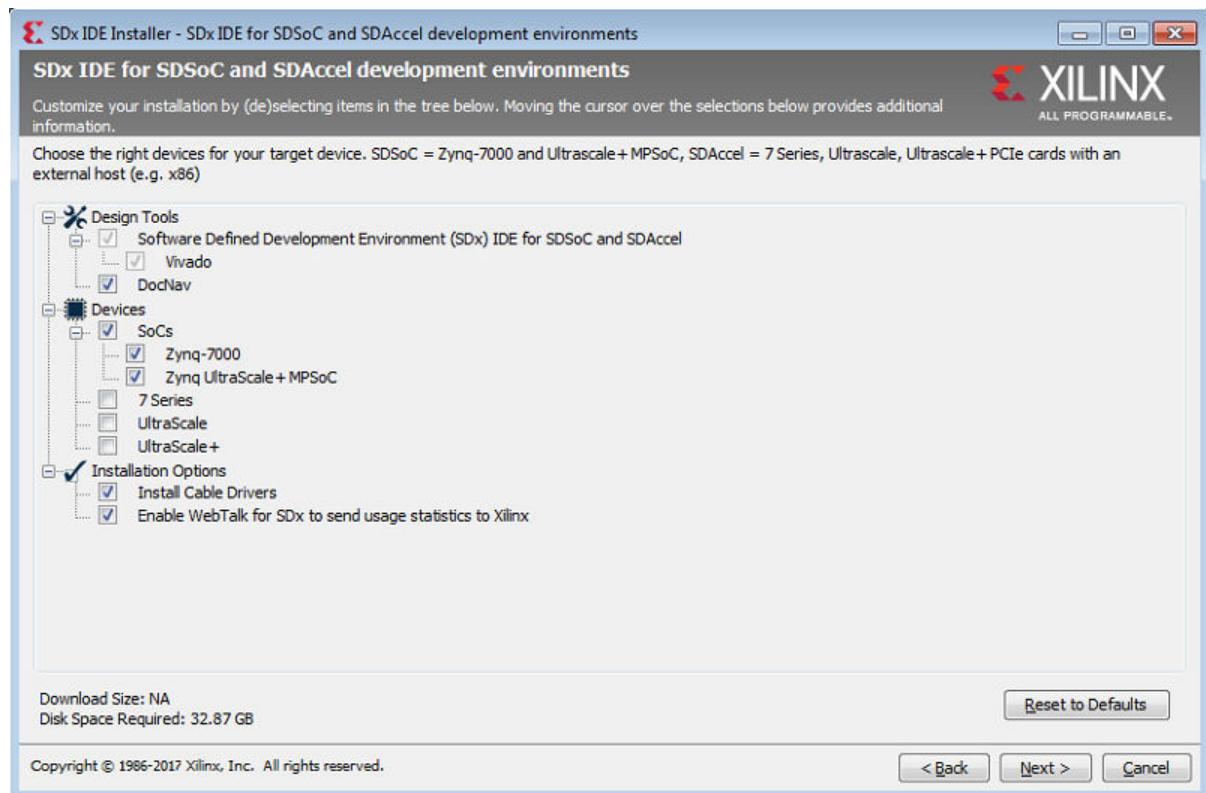


ツールおよびデバイス オプションの選択

デザイン ツール、デバイス ファミリーおよびインストール オプションを選択してインストールをカスタマイズします。必要なものだけを選択することで、製品のダウンロードおよびインストールにかかる時間を削減できます。後から追加するには、OS の [スタート] メニューか Vivado の [Help] メニューのいずれかで [Add Design Tools or Devices] をクリックします。

使用する製品のインストーラーを起動すると、デバイスが前もって選択されます。

図 5: デザイン ツールおよびデバイス オプション



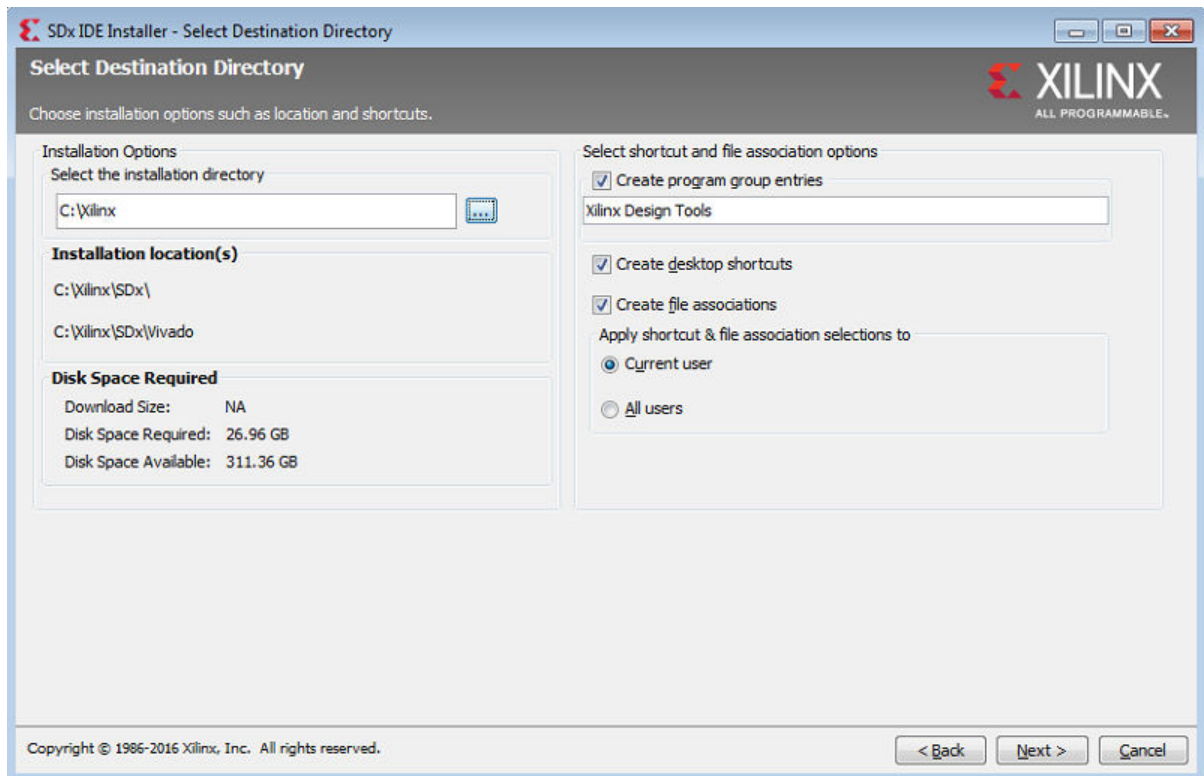
インストール ディレクトリとオプションの設定

ツールのインストール ディレクトリは次の図のように定義します。

注記: インストール ディレクトリ名には、スペースが含まれないようにしてください。

プログラム グループ入力 ([スタート] メニュー) とデスクトップ ショートカットの作成はカスタマイズできます。ショートカット作成およびファイルの関連付けオプションは、現在のユーザーまたはすべてのユーザーに適用できます。

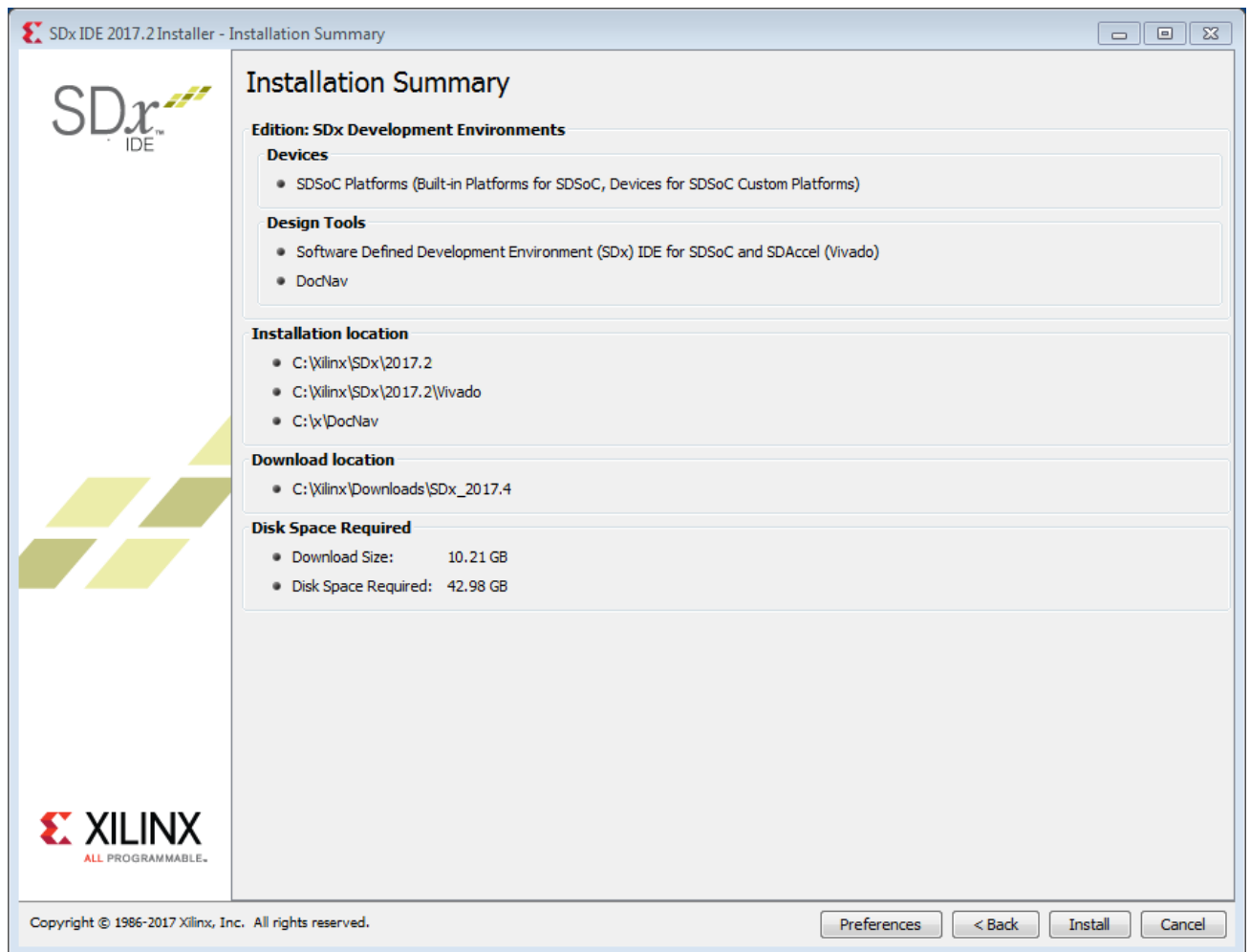
図 6: インストール ディレクトリとオプション



インストールの詳細の確認

インストールの詳細をサマリ ページで確認します。

図 7: インストール サマリ



[Install] をクリックすると、インストールが始まります。インストールが終了するには、数分かかります。

SDx を実行するための環境の設定

1. SDx を実行する環境を設定するには、次のファイルを読み込んで `sdx` コマンドが `PATH` に含まれるようにします。

```
C Shell: source <SDX_INSTALL_DIR>/settings64.csh
Bash: source <SDX_INSTALL_DIR>/settings64.sh
```

その他のリソースおよび法的通知

ザイリンクス リソース

アンサー、資料、ダウンロード、フォーラムなどのサポート リソースは、[ザイリンクス サポート](#) サイトを参照してください。

ソリューションセンター

デバイス、ツール、IP のサポートについては、[ザイリンクス ソリューションセンター](#)を参照してください。デザイン アシスタント、デザイン アドバイザリ、トラブルシューティングのヒントなどが含まれます。

参考資料

このガイドの補足情報は、次の資料を参照してください。

日本語版のバージョンは、英語版より古い場合があります。

SDAccel の資料

1. 『SDAccel 環境ユーザー ガイド』 ([UG1023](#))
2. 『SDAccel 環境最適化ガイド』 ([UG1207](#))
3. 『SDAccel 環境チュートリアル: 概要』 ([UG1021](#))
4. 『SDSoC 環境プラットフォーム開発ガイド』 ([UG1164](#))

SDSoC の資料

1. 『SDSoC 環境ユーザー ガイド』 ([UG1027](#))
2. 『SDSoC 環境最適化ガイド』 ([UG1235](#))
3. 『SDSoC 環境チュートリアル: 概要』 ([UG1028](#))
4. 『SDSoC 環境プラットフォーム開発ガイド』 ([UG1146](#))

その他の資料

1. 『SDx プラグマ リファレンス ガイド』 ([UG1253](#))
2. 『ザイリンクス OpenCV ユーザー ガイド』 ([UG1233](#))
3. 『プラットフォーム ケーブル USB II データシート』 ([DS593](#))

その他のリソース

1. ザイリンクス ライセンス ウェブサイト: <https://japan.xilinx.com/getproduct>
2. SDSoC 開発者ゾーン: <https://japan.xilinx.com/products/design-tools/software-zone/sdsoc.html>
3. SDAccel 開発者ゾーン: <https://japan.xilinx.com/products/design-tools/software-zone/sdaccel.html>
4. 『ザイリンクス エンドユーザー使用許諾契約』 (UG763)
5. 『サードパーティ エンドユーザー使用許諾契約』 (UG1254)

Documentation Navigator およびデザイン ハブ

Xilinx® Documentation Navigator (DocNav) を使用するとザイリンクスの資料、ビデオ、およびサポート リソースにアクセスして情報をフィルター、検索できます。Xilinx Documentation Navigator を開くには、次のいずれかを実行します。

- Vivado® IDE で [Help] → [Documentation and Tutorials] をクリックします。
- Windows で [スタート] → [すべてのプログラム] → [Xilinx Design Tools] → [DocNav] をクリックします。
- Linux コマンド プロンプトに「docnav」と入力します。

ザイリンクス デザイン ハブには、資料やビデオへのリンクがデザイン タスクおよびトピックごとにまとめられており、これらを参照することでキー コンセプトを学び、よくある質問を解決できます。デザイン ハブにアクセスするには、次のいずれかを実行します。

- Xilinx Documentation Navigator で [Design Hubs View] タブをクリックします。
- ザイリンクス ウェブサイトで [デザイン ハブ](#) ページを参照します。

Documentation Navigator の詳細は、ザイリンクス ウェブサイトの [Documentation Navigator](#) ページを参照してください。Xilinx Documentation Navigator からは日本語版は参照できません。ウェブサイトのデザイン ハブ ページの一部は翻訳されており、日本語版が提供されている場合はそのリンクも追加されています。

お読みください: 重要な法的通知

本通知に基づいて貴殿または貴社 (本通知の被通知者が個人の場合には「貴殿」、法人その他の団体の場合には「貴社」。以下同じ) に開示される情報 (以下「本情報」といいます) は、ザイリンクスの製品を選択および使用することのためにのみ提供されます。適用される法律が許容する最大限の範囲で、(1) 本情報は「現状有姿」、およびすべて受領者の責任で (with all faults) という状態で提供され、ザイリンクスは、本通知をもって、明示、黙示、法定を問わず (商品性、非侵害、特定目的適合性の保証を含みますがこれらに限られません)、すべての保証および条件を負わない (否認する) ものとし、また、(2) ザイリンクスは、本情報 (貴殿または貴社による本情報の使用を含む) に関係し、起因し、関連する、いかなる種類・性質の損失または損害についても、責任を負わない (契約上、不法行為上 (過失の場合を含む)、その他のいかなる責任の法理によるかを問わない) ものとし、当該損失または損害には、直接、間接、特別、付随的、結果的な損失または損害 (第三者が起こした行為の結果被った、データ、利益、業務上の信用の損失、その他あらゆる種類の損失や損害を含みます) が含まれるものとし、それは、たとえ当該損害や損失が合理的に予見可能であったり、ザイリンクスがそれらの可能性について助言を受けていた場合であったとしても同様です。ザイリンクスは、本情報に含まれるいかなる誤りも訂正する義務を負わず、本情報または製品仕様のアップデートを貴殿または

貴社に知らせる義務も負いません。事前の書面による同意のない限り、貴殿または貴社は本情報を再生産、変更、頒布、または公に展示してはなりません。一定の製品は、ザイリンクスの限定的保証の諸条件に従うこととなるので、<https://japan.xilinx.com/legal.htm#tos> で見られるザイリンクスの販売条件を参照してください。IP コアは、ザイリンクスが貴殿または貴社に付与したライセンスに含まれる保証と補助的条件に従うこととなります。ザイリンクスの製品は、フェイルセーフとして、または、フェイルセーフの動作を要求するアプリケーションに使用するために、設計されたり意図されたりしていません。そのような重大なアプリケーションにザイリンクスの製品を使用する場合のリスクと責任は、貴殿または貴社が単独で負うものです。<https://japan.xilinx.com/legal.htm#tos> で見られるザイリンクスの販売条件を参照してください。

自動車用のアプリケーションの免責条項

オートモーティブ製品 (製品番号に「XA」が含まれる) は、ISO 26262 自動車用機能安全規格に従った安全コンセプトまたは余剰性の機能 (「セーフティ設計」) がない限り、エアバッグの展開における使用または車両の制御に影響するアプリケーション (「セーフティ アプリケーション」) における使用は保証されていません。顧客は、製品を組み込むすべてのシステムについて、その使用前または提供前に安全を目的として十分なテストを行うものとします。セーフティ設計なしにセーフティ アプリケーションで製品を使用するリスクはすべて顧客が負い、製品の責任の制限を規定する適用法令および規則にのみ従うものとします。

Copyright

© Copyright 2016-2018 Xilinx, Inc. Xilinx、Xilinx のロゴ、Artix、ISE、Kintex、Spartan、Virtex、Vivado、Zynq、およびこの文書に含まれるその他の指定されたブランドは、米国およびその他のザイリンクス社の商標です。OpenCL および OpenCL のロゴは Apple Inc. の商標であり、Khronos による許可を受けて使用されています。すべてのその他の商標は、それぞれの所有者に帰属します。

この資料に関するフィードバックおよびリンクなどの問題につきましては、jpn_trans_feedback@xilinx.com まで、または各ページの右下にある [フィードバック送信] ボタンをクリックすると表示されるフォームからお知らせください。フィードバックは日本語で入力可能です。いただきましたご意見を参考に早急に対応させていただきます。なお、このメール アドレスへのお問い合わせは受け付けておりません。あらかじめご了承ください。