



ADK-2130mPCIe
テクニカル・マニュアル



株式会社ナセル

Introduction [はじめに]

Holt のフルサイズ F2 Mini PCI カードのリファレンス・デザインは、1つの Mini PCI カードにトランスが統合された、1つ、または、2つの Holt HI-2130 MIL-STD-1553 マルチチャンネル・ターミナルを備えています。このカードは、Linux OS を搭載した PC または、シングル・ボード・コンピュータで動作するように設計されています（注記：Windows のサポートは近日中に利用可能になる予定です）。Demo ソフトウェアは、Holt API ライブラリ関数を使用して、ホスト・プログラミングを大幅に簡略化する抽象化レイヤーを提供します。このテクニカル・マニュアルでは、ハードウェア、ソフトウェアおよび、Holt Flash Drive (FD) ファイルをユーザーの PC に転送する方法について説明します。カードの概要、アプリケーション開発キット (ADK) の内容、および Holt ブータブル・フラッシュ・ドライブを使用して Demo ソフトウェアを実行する方法については、クイック・スタート・ガイド (QSG) を参照してください。このガイドの手順を使用して、ソフトウェア開発に使用する Holt ソフトウェアと Eclipse IDE ツールをユーザーの PC にインストールします。QSG のインストラクションを使用する場合 USB FD からブートして下さい。FPGA デザインについては、テクニカル・ガイド AN-PCIeFPGA を参照して下さい。



図 1 Mini PCIe カード EV-2130mPCIe-2F

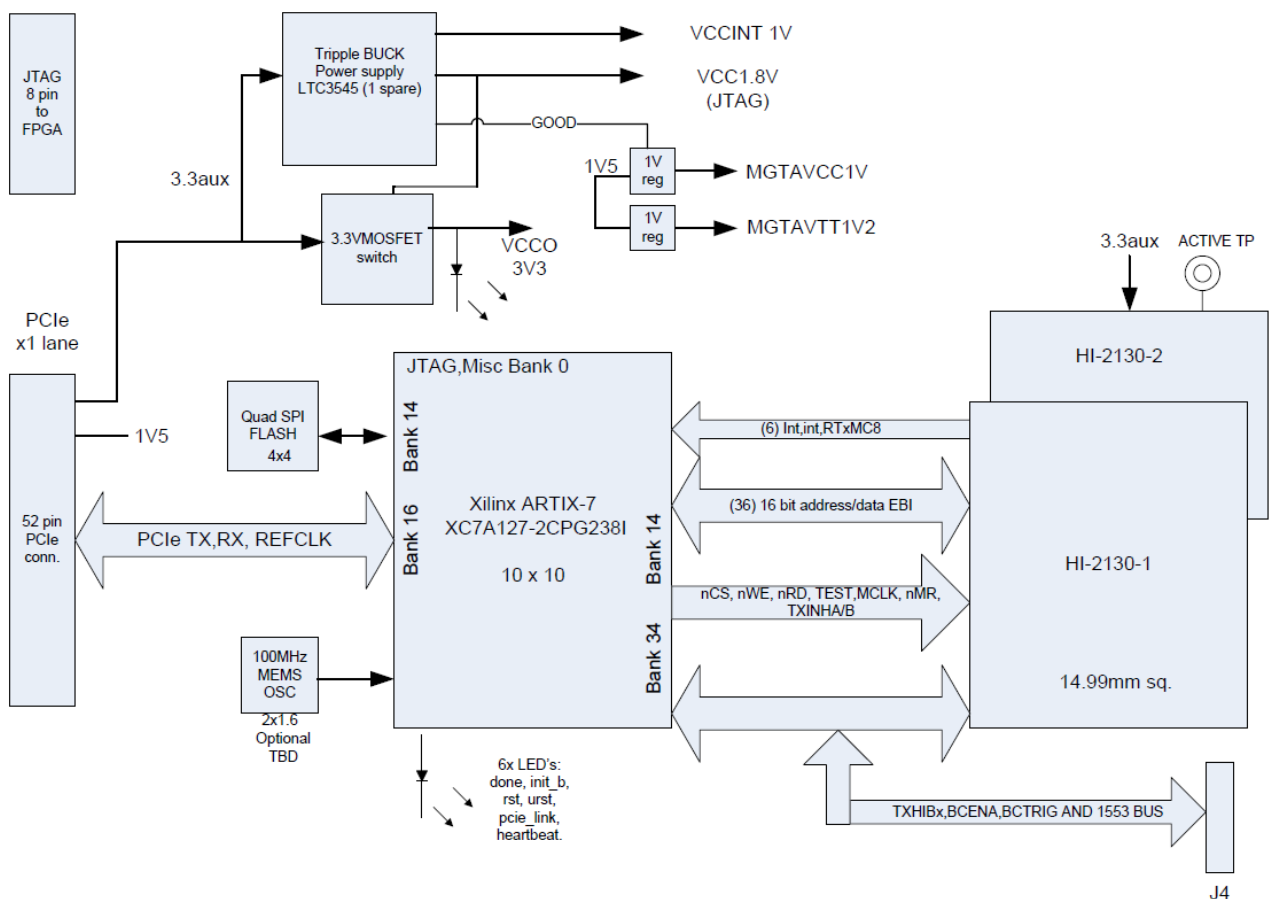
Features [特徴]

- Mini PCIe F2 フルサイズ、PCIe Gen2 シングル・レーン
- シングル・チャンネル ADK-2130mPCIe-1F または、デュアル・チャンネル ADK-2130mPCIe-2F
- Mini Card Electromechanical Specification Rev. 1.2
- 実績のある HI-2130 を使用した独立した 2×二重冗長 MIL-STD-1553 チャンネル
- BC、2×RT、MT の各チャンネル
- トランス結合 MIL-STD-1553 インターフェイス
- Holt API ライブラリのサポート
- Linux OS
- 動作温度：-40°C~+85°C
- カスタマイズ可能な FPGA
- BC、RT、RT2、MT の Demo ソフトウェア

PC requirements [PC 要件]

Linux OS :	Ubuntu : 16.04 LTS または、18.04 LTS (Holt FD インストール)
システム RAM :	Eclipse では 8GB、Vivado では 16GB 推奨
ハード・ドライブ容量 :	Eclipse プロジェクトでは 10GB 以上
Mini PCI スロット :	スタンダード・フルサイズ F2
USB 2.0 または、3.0 ポート :	Holt Flash Drive ファイルの転送用

Mini PCIe board block diagram [Mini PCI ボード・ブロック図]



Break Out Board [ブレイク・アウト・ボード]

ブレイク・アウト・ボードは、リボン・ケーブルを介して PCIe カードに接続し、両方のチャンネル (Dev0 と Dev1)からの BusA と BusB の両方の接続をブレイク・アウトするために使用されます。Transmit Inhibits 入力は High にプルアップされており、HI-2130 デバイスに Low を提供し、デフォルトで 1553 Bus 送信を有効にします。表 8 は、インターコネクト・リボン J4 コネクタの信号と説明のリストです。ボード・コンポーネントの位置、コネクタ、IC、およびテスト・ポイントについては、ドキュメントの最後にある図 3 ボード・リファレンスを参照してください。2 つのブレイク・アウト・ボードを使用できます。

シングル・チャンネル : mPCIe_breakout-1F

デュアル・チャンネル : mPCIe_breakout-2F

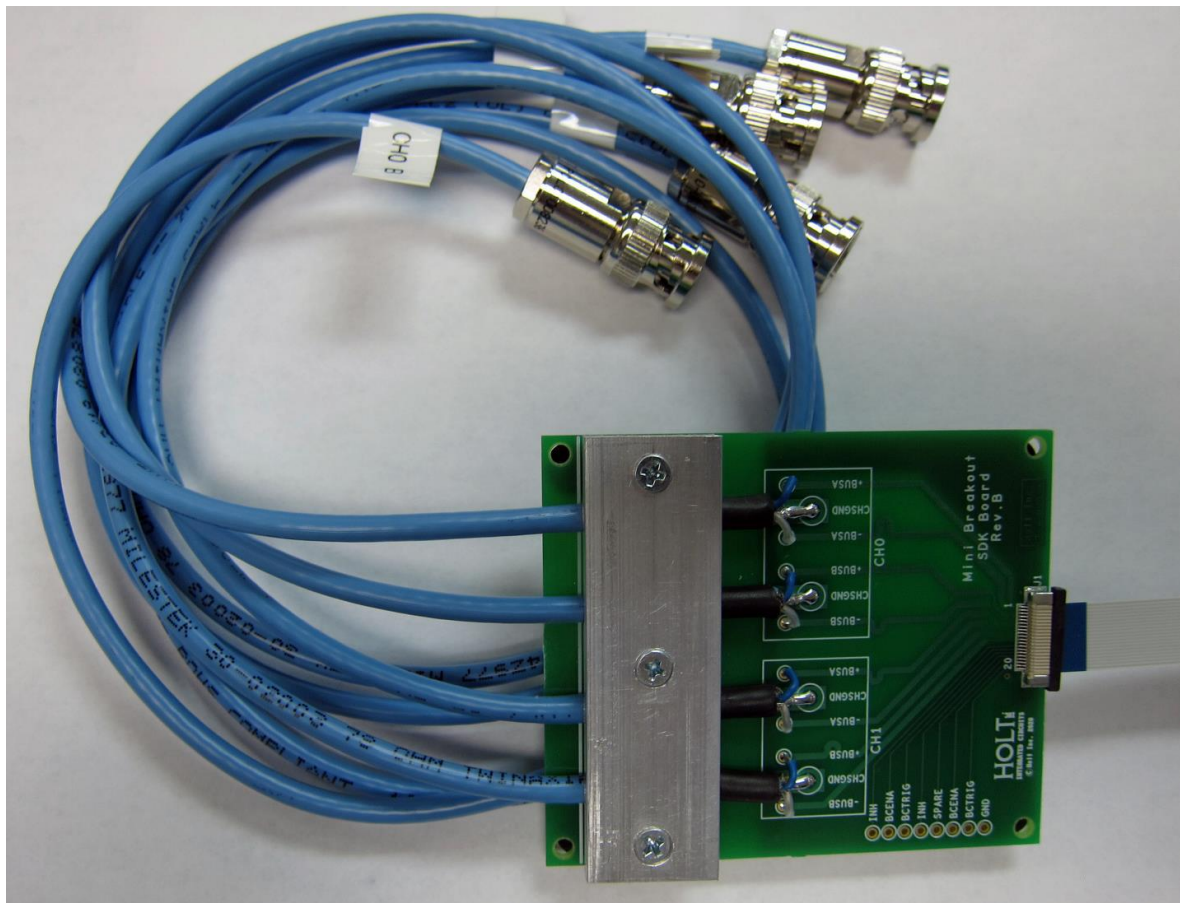


図 2 mPCIe_breakout-2F ブレイク・アウト・ボード

First Things First [重要なことから先に]

Holt のデモ・プログラムを正常に実行するには、PC が電源投入時にカードの PCIe インターフェイスを検出できる必要があります。

1. このドキュメントを読む前に「Holt QSG-2130mPCIe.pdf」を確認することを強くお勧めします。
2. Linux エクスペリエンスとコマンド・ラインを備えたターミナル・ウィンドウを推奨します。
3. PC の電源を切り、Holt Mini PCIe カードを取り付けます。ブレーク・アウト・ボードを接続することはできません。この手順には不必要なので、オプションです。
4. Holt FD の Ubuntu OS を使用して、ユーザーの PC に Ubuntu をインストールすることはできません。ユーザーが 18.04 LTS より古いバージョンの Ubuntu では、アップグレードできる可能性があります。Ubuntu のウェブサイトから Ubuntu 18.04 LTS をダウンロードし、提供されているインストール手順を使用することを推奨します。Holt は、他の Linux バージョンでは検証していません。Windows のサポートは将来的に予定しています。
<https://ubuntu.com/#download>
5. PC の電源を入れます。コンピュータの電源を入れた直後に、緑色の PCIe リンク LED 8 が ON になると、ハートビート LED 9 が点滅します。
6. LED 8 が点灯している場合、コンピュータの起動シーケンスでカード上の PCIe リンクが検出されています。これを確認する別の方法は、ターミナル・ウィンドウを開いて次のコマンドを実行することです

```
holt@holt-desktop:~/holt$ lspci | grep Xilinx  
"Memory controller: Xilinx Corporation Device 7011"
```

これが表示される場合は、PC は PCIe リンクを正常に検出しています。

“Xilinx Corporation Device 7011” が表示されない場合は、PC の電源を切り、カードを再度取り付けて、カードが完全に接続されていることを確認してください。LED が点灯しない場合は、Mini PCIe スロットの 1.5V が供給されていない可能性があります。PC マザーボードのマニュアルを参照して、構成または、ジャンパー設定で 1.5V を有効にする必要があるかどうかを確認して下さい。別の PC を使用する必要がある場合があります。

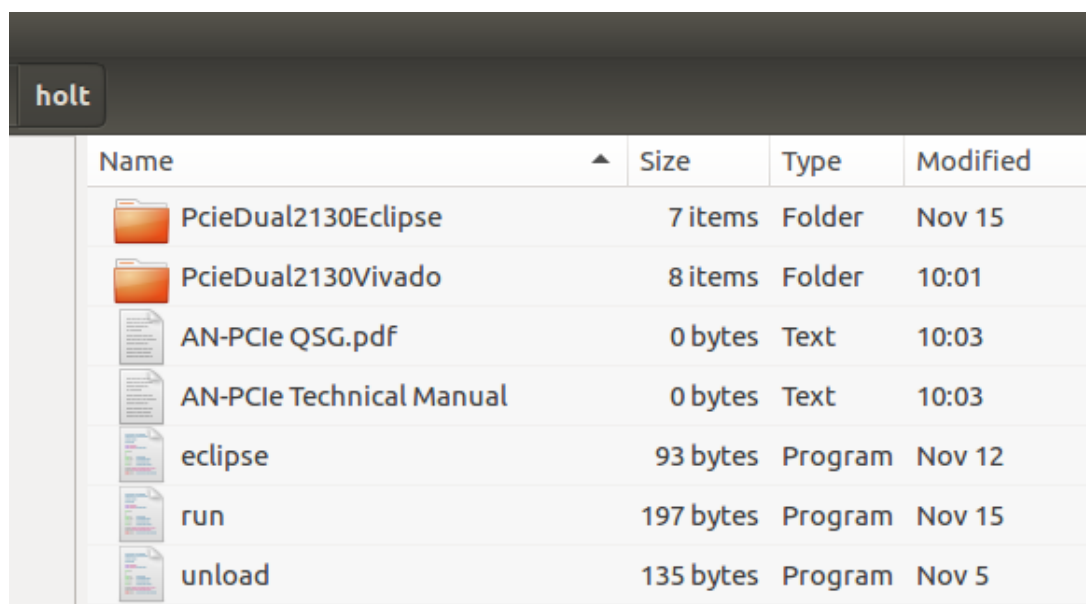
PC preparation for installing Eclipse and the Holt software [Eclipse と Holt ソフトウェアをインストールするための PC の準備]

Eclipse C/C++ IDE ツールは、www.eclipse.org から無償で、およびこのプロジェクトで使用できます。Eclipse が既にインストールされている場合は、この手順を省略できます。

1. Java JRE をインストールする必要があります。ターミナル・ウィンドウを開き「`sudo apt install default-jre`」と入力して Enter キーを押します。インストールを確認し、(Y) を押します。Java がインストールされたら、「Java -version」と入力して場 0 ジョンを示すメッセージを表示して、インストールを確認します。
2. Eclipse ダウンロード・ページにアクセスして、ソフトウェアをダウンロードします。Eclipse インストーラーを使用している場合は、C/C++パッケージを選択します。インストールが完了したら、オプションで Eclipse を起動して動作を確認し、プログラムを終了します。Eclipse がインストールされているターミナル・ウィンドウからプログラムのショートカットを作成し、リンクをデスクトップにドラッグすることができます。Eclipse がインストールされている場所に移動し、Eclipse プログラムのアイコンを右クリックして、「Make Link」を作成してリンクを作成します。リンクをデスクトップにドラッグします。

Holt デモ・ソフトウェアは、単一の Eclipse ワークスペース・フォルダにある 3 つのプロジェクト・サブフォルダで構成されています。

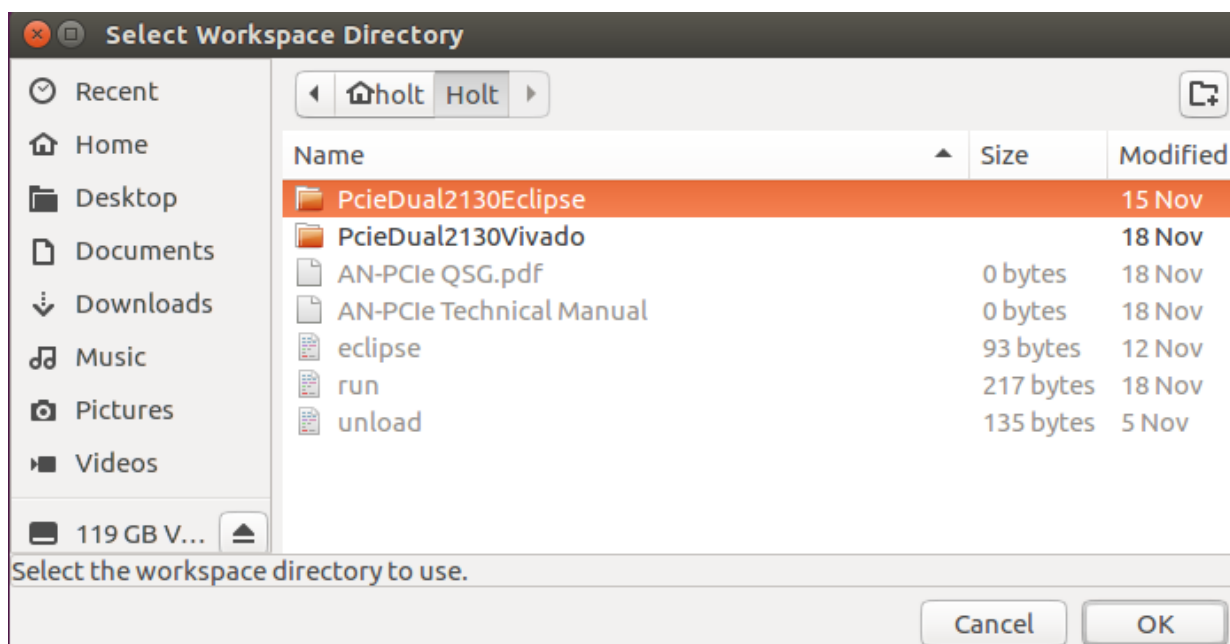
3. Ubuntu OS の Home フォルダに Holt フォルダを作成します。
全てのフォルダ、スクリプト・ファイル、その他の各種ドキュメントをフラッシュ・ドライブから PC に新しく作成された Holt フォルダにコピーします。フォルダとファイルは次のようになります。



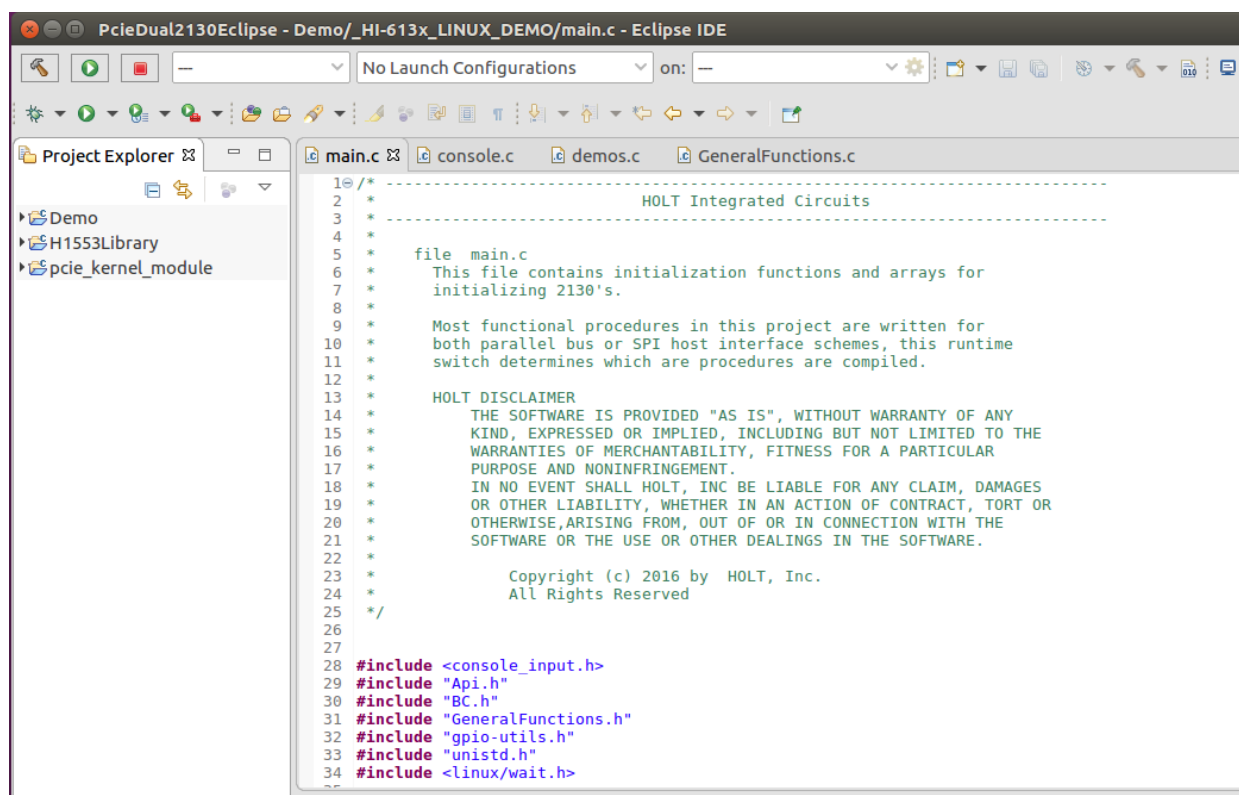
The screenshot shows a file manager window with a dark header bar containing the text 'holt'. Below the header is a table listing the contents of the folder. The table has five columns: Name, Size, Type, and Modified. The items listed are:

Name	Size	Type	Modified
PcieDual2130Eclipse	7 items	Folder	Nov 15
PcieDual2130Vivado	8 items	Folder	10:01
AN-PCIe QSG.pdf	0 bytes	Text	10:03
AN-PCIe Technical Manual	0 bytes	Text	10:03
eclipse	93 bytes	Program	Nov 12
run	197 bytes	Program	Nov 15
unload	135 bytes	Program	Nov 5

4. Eclipse を起動し、「Browse」ボタンを使用して以下に示すプロジェクト・フォルダを選択し、OK ボタンをクリックします。



- Eclipse が最初にプロジェクトを開いたときに、プロジェクトをインポートする必要があります。Eclipse プロジェクト・エクスプローラー (PE) ウィンドウは、「Project Explorer」で画面の右上隅にあるクイックアクセス・ウィンドウに表示されず、表示されている項目のリストから選択します。3つのプロジェクト・フォルダがPEに表示されます



6. 3つのプロジェクトの説明は、サブフォルダに含まれています。

Demo : デモ・ソフトウェアは、両方のデバイスをリセットし、Holt API ライブラリ関数を使用して両方のデバイスを初期化します。ビルド・コンフィグレーションには「Debug」と「Debug_precompiled_library」の2つがあります。Debug_precompiled_libraryには、H1553Library プロジェクトによって生成されたバイナリ・ライブラリファイルが必要です。下記参照。「Debug」コンフィグレーションをリビルドすると、Holt API ライブラリ・ファイルを含むすべてのデモ・ファイルがコンパイルされ、デモ実行可能ファイルが出力されます。

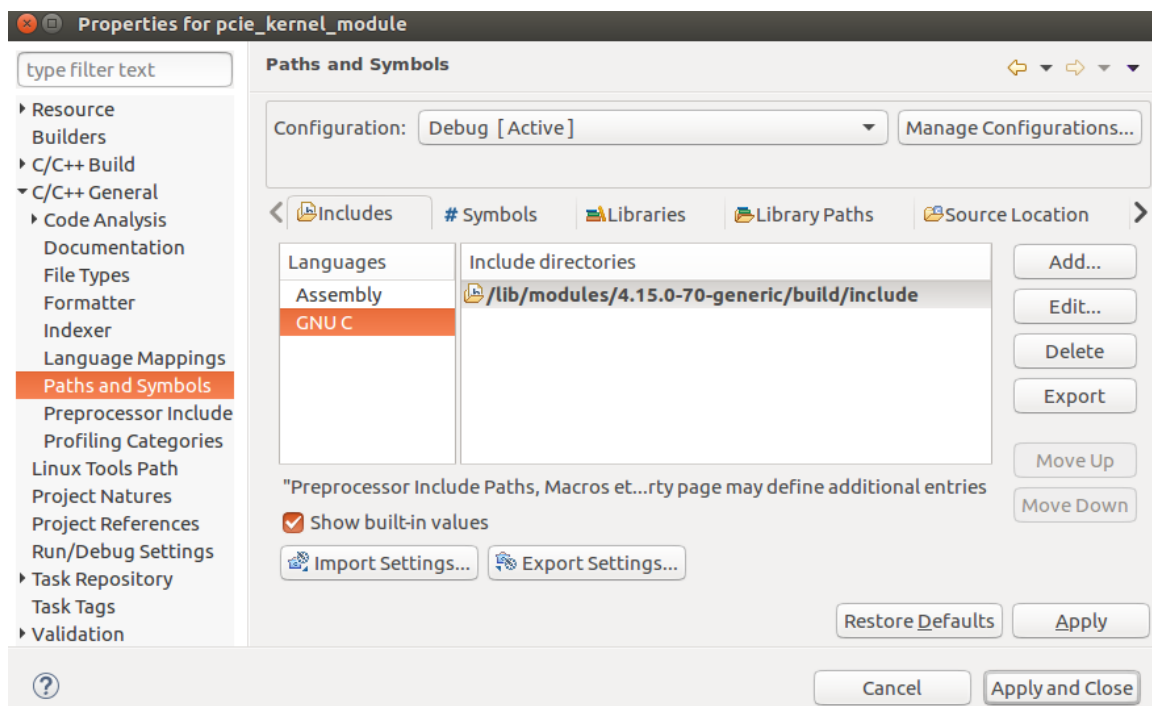
H1553Library : このプロジェクトは、上記の Debug_precompiled_library ターゲット・デモ・プロジェクトにリンクされている API ライブラリ出力ファイル libH1553Library.a をビルドします。

Pcie_kernel_module : このプロジェクトは、Linux ロード可能カーネル・モジュール「pcie_lkm.ko」をビルドします。このプロジェクトは、デモ・プロジェクトを実行する前に Linux OS にロードする必要があります。そうしないと、エラーが発生します。このモジュールは、Linux のデフォルト PCIe ドライバを使用して、HI-2130 デバイス・レジスタとメモリ空間の書き込みと読み込みをサポートします。

7. デモ実行可能ファイルを実行する前にプロジェクトをリビルドする手順

最初に実行してください！！

カーネル・モジュールは、PC の /lib/modules にある OS ヘッダー・ファイルのバージョンでリビルドする必要があります。通常 4.15.0-70-generic などの名前のフォルダがあります。このフォルダのパスは、次に示すように、カーネル・モジュール・プロジェクトの設定で設定する必要があります。これは、カーネル・モジュール・プロジェクト・フォルダを右クリックして「properties」を選択し、次に「paths and Symbols」の項目を選択して表示されます。エントリが PC のフォルダ名と一致しない場合は、エントリを編集して一致させます。



- カーネル・モジュール・プロジェクトをリビルドします。カーネル・モジュール・プロジェクトをリビルドする前にプロジェクトの「clean」を実行しておくといいです。PE 領域の「pcie_kernel_module」フォルダを右クリックしてプロジェクトのクリーンを実行し、コンテキスト・メニューから「clean」を選択します。クリーンアップが実行された後、コンテキスト・メニューからもプロジェクトをリビルドします。プロジェクトはエラーなしでビルドされます。通常、無視できる警告がいくつかあります。ビルドは新しい場合 0 ジョンの pcie_lkm.ko カーネル・モジュールを生成します。このモジュールを OS カーネルにロードできるようにするには、ターミナル・ウィンドウでこのコマンドを実行します。

```
sudo chmod 777 pcie_lkm.ko or sudo chmod +x pcie_lkm.ko
```

- pcie_kernel_module サブフォルダでターミナル・ウィンドウを開き、この bash スクリプトを実行して、新しくビルドされたカーネル・モジュールをロードします。

```
sudo sh holt_pcie_load
```

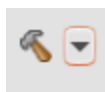
カーネル・モジュールのロードは、PC の電源投入後に一度だけ必要です。カーネル・モジュール・プロジェクトを変更してリビルドする場合、新しいカーネル・モジュールをリロードする前に、既存のカーネル・モジュールをアンロードする必要があります。

```
sudo sh unload_pcie_load
sudo sh load_pcie_load
```

これらのスクリプト・コマンドは、QSG 命令で使用される「run」スクリプトに埋め込まれています。

- 以前に提供されたものと同じ手順を使用して、H1553Library プロジェクトをクリーンアップしてリビルドします。PE でナビゲートし libH1553Library.a ファイルを見つけます。このファイルをコピーして、PE の Demo プロジェクト・フォルダに貼り付けます。
- Demo プロジェクトをクリーンアップしてリビルドします。2 つの Demo プロジェクト・ターゲットがあります。「debug」は、Holt API ソース・ファイルと Demo ファイルをリビルドして、デモ実行可能ファイルを生成します。これは、デモ・プロジェクトをリビルドするもっとも簡単で最速の方法です。Holt API ソース・ファイルが無い場合は、Debug_precompiled_library ターゲットを使用する必要があります。このプロジェクトは libH1553Library.a ファイルを使用します。このファイルは、事前にビルドされ、Holt によって提供された Demo フォルダにすでに存在しているはずですが。

ヒント：プロジェクトをリビルドする別の方法は、最初に PE ウィンドウ領域で対応するプロジェクト・フォルダを選択してから、Eclipse 画面の上部に表示されるビルド「hammer」アイコンをクリックすることです。プルダウン・メニューを使用してビルドターゲットを選択します。



12. Demo を実行するには 2 つの方法があります (カーネル・モジュールを読み込んで Demo プロジェクトをリビルドした後)。最初の方法は、Holt ディレクトリでターミナル・ウィンドウを開き、「run」bash スクリプトを実行することです。これにより、カーネル・モジュールが読み込まれ、Demo プログラムが自動的に実行されます。この方法は QSG ドキュメントで使用されています。

13. Eclipse 統合デバッガで Demo を実行します

- a. はじめに、カーネル・モジュールが以前にロードされていることを確認します
- b. Eclipse で、画面の左上または、右上にある緑色 (虫) デバッグ・アイコンをクリックします。C/C++ Controller Application または、Local C/C++ Application のいずれかを選択するように求める単純なウィンドウが表示されます。Local C/C++ Application を選択します。
- c. デバッガは、main.c をウィンドウに表示し、コードの最初の行を緑色に強調表示します。ユーザーは、Resume アイコンを押してプログラムを実行するか F8 を押します。



- d. デバッガでは、ブレークポイント、シングル・ステップ、およびハイエンドの統合ソース・レベル・デバッガに典型的なその他のデバッガ機能を使用できます。Eclipse デバッガの使用方法をよりよく理解するには、Eclipse のヘルプを参照してください。
- e. Demo コンソール・メニューがコンソール・ウィンドウに表示されます。コンソール・タブをダブルクリックしてウィンドウ・サイズを拡大します。完全なメニューが表示されます。コードを変更する場合は、Eclipse デバッガを使用してください。ターミナル・ウィンドウから (Eclipse を使用せずに) デモを実行すると、表示とエクスペリエンスが少し向上します。

PcieDual2130Eclipse - Demo/_HI-613x_LINUX_DEMO/main.c - Eclipse IDE

Debug Demo

main.c console.c demos.c GeneralFunctions.c

```

71  */
72
73
74  int main (
75      int argc, /* number of arguments */
76      char * argv[] /* array of arguments */
77  )
78  {
79      int ret = 0;
80      S16BIT channel = 0;
81
82      for (channel = DEV0 ; channel < 2; channel++)
83      {
84
85          if(HoltInitialize(channel, // also performs HW reset to both Dev0 and Dev1
86                          0,
87                          HOLT_MODE_BC | HOLT_MODE_RT | HOLT_MODE_RT2 | HOLT_MODE_SMT | HOLT_MODE_RTMT,

```

Console Registers Problems Executables Debugger Console Memory

Demo [C/C++ Application] Demo

```

*****
Holt Integrated Circuits
Mini PCIe Dual HI-2130 API Demo
Demo Rev: 1.0    Compiled: Nov 26 2019 10:29:55
API Lib Rev: 03-5-0
*****

BC On  SMT On  RT1 On  RT2 On

Press 'a' or 'A' to run Dev0 or Dev1 BC Async demo.
Press 'b' or 'B' to run Dev0 or Dev1 RT demo.
Press 'c' or 'C' to run Dev0 or Dev1 RT2 demo.
Press 'k' or 'K' to Enable Dev0 or Dev1 RTMT.
Press 'H' to send high priority BC message.
Press 'L' to send low priority BC message.
Press 'n' or 'N' to run Dev0 or Dev1 BC Major Minor Frame demo.

```

Demo project description [Demo プロジェクトの説明]

Demo ソフトウェアは、Linux ドライバを初期化し、ホストが PCIe インターフェイスを介して HI-2130 デバイスと通信できるようにします。これは、main.c の `HoltInitialize()` で実行されます。Xilinx FPGA は、PCIe バスと両方の HI-2130 間のハードウェア・インターフェイスとして機能します。

メニューがコンソールに表示され、プログラムはユーザーがコンソール・コマンドを押すのを待ちます。コマンドは `chk_key_input()` 関数によって受け入れられます。ユーティリティのコードの一部は `console.c` に含まれています。

モジュール `Demo.c` に含まれるすべての 1553 BC、RT、および SMT デモは、Holt API ライブラリ関数を使用して、ターミナルを初期化および、制御します。BC はメッセージを送信するように初期化され、RT と SMT は 1553 データ・ワードを読み書きできるように初期化されます。Holt HI-613x マニュアルは、変更を加える前にコードがどのように機能するかを全体的に理解するために、デモコードとともに検討する必要があります。

Demo Preparations [Demo の準備]

デモを実行する前に、PCIe カードとブレイク・アウト・ボードの間にリボン・ケーブルを接続します。

小さなリボン・ケーブルを Mini PCIe カード J4 コネクタに慎重に挿入し、もう一方の端をブレイク・アウト・ボードに挿入します。ケーブルのピッチは細かく、コネクタのプラスチック製ファスナーはデリケートなので、コネクタの損傷を防ぐために注意が必要です。両手でプラスチックのファスナーを慎重に引き出し、ケーブルを挿入して、両側のファスナーを押し込みます。

ブレイク・アウト・ボードを接続せずにデモを実行することは可能ですが、これは推奨されておらず、1553 バス終端がないため必ずしも機能しない場合があります。

Demo のレビューについては、いくつかの例とコンソール画面が表示された QSG を参照してください。QSG ですでにカバーされているコンソール・コマンドの一部は、ここではカバーされません。

QSG で述べたように、両方のデバイスの 2 つの RT アドレスは同じです。RT1 = 3 および RT2 = 1。BC デモはこれら 2 つの RT アドレスにのみメッセージ・コマンドを送信するため、これによりデモが簡素化されます。そして RT は同じ RT アドレスでプログラムされます。ユーザーはコマンド「9」を使用して、他の目的で RT アドレスを変更できます。

Demos [デモ]

Holt フォルダで「run」スクリプトを実行して、Demo プログラムを実行します。QSG は「run」スクリプトを実行し、ほとんどのデモを実行する方法を提供します。

「run」スクリプトには、8つのターミナル・コマンドが含まれています。

```
cd PcieDual2130Eclipse/pcie_kernel_module
sudo sh holt_pcie_unload
sudo sh holt_pcie_load
cd ..
cd Demo
cd Debug
ls
sudo ./Demo
```

```
holt@holt-desktop:~/holt$ sh run
06:00.0 Memory controller: Xilinx Corporation Device 7011
[sudo] password for holt:
Holt Linux driver unloaded
Holt Linux driver loaded
1553LibrarySrc_linkedfolder Demo_HI-613x_LINUX_DEMO makefile
objects.mk sources.mk
Setting nMR chan 0 LOW Setting nMR chan 0 HIGH READY asserted
Setting nMR chan 1 LOW Setting nMR chan 1 HIGH READY asserted
Number of Devices found: 2

Initial default RT addresses:
DEVO:RT1=3 DEVO:RT2=1 DEV1:RT1=3 DEV1:RT1
Optionally use console command '9' to change these RT addresses
BEFORE RUNNING RT

*****
Holt Integrated Circuits
Mini PCIe Dual HI-2130 API Demo
Demo Rev: 1.0 Compiled: Dec 11 2019 08:41:16
API Lib Rev: 03-5-0
*****

BC On SMT On RT1 On RT2 On

Press 'a' or 'A' to run Dev0 or Dev1 BC Async demo.
Press 'b' or 'B' to run Dev0 or Dev1 RT demo.
Press 'c' or 'C' to run Dev0 or Dev1 RT2 demo.
Press 'k' or 'K' to Enable Dev0 or Dev1 RTMT.
Press 'l' or 'L' to send high priority BC message.
Press 'h' or 'H' to send low priority BC message.
Press 'n' or 'N' to run Dev0 or Dev1 BC Major Minor Frame demo.
Press 'x' or 'X' to stop Dev0 or Dev1 BC transmissions.
Press 'S' to run SMT demo.
Press 't' to display RT Traffic Toggle.
```

```
----- Utilities -----
Press 'r' or 'R' to Display Dev0 or Dev1 HI-2130 Registers.
Press 'w' for Memory Watch window
Press 'f' Reads J4 connector and FPGA control signals
Press '1' for Register Write
Press '2' for Memory Write
Press '3' RT Mode Code data word reads
Press '4' Master Reset and reinitializes terminals
Press '5' Toggle Dev0 BCENA on/off
Press '6' Toggle Dev1 BCENA on/off
Press '9' Set RT addresses
Press '0' Toggle between User and Demo(default) modes

Press 'M' for menu, or press any valid menu key. >>
```

注記：「-1F」カード（HI-2130×1つ）の場合、大文字のコマンドはコンソール・メニューに表示されません。

以下の説明では、「r」などの一部のコマンドには、2番目のデバイスに対して大文字の「R」と同等のコマンドがあります。Dev0の例をいくつか示します。

コマンド「r」：デバイスのシステム・レジスタを名前と値で表示します。レジスタ設定の確認に便利です。
コマンド「w」：メモリ・ウォッチ・ウィンドウは、各デバイスの位置 0x0000 から始まる 256 ワードを読み取ります。これは、すべてのシステム・レジスタを一目で確認するのに役立ちます。サブコマンドを使用すると、メモリ空間を上下に移動できます。これは RT 制御ブロック、BC メッセージ・リスト、割り込みログ・テーブルなど、メモリの広い領域を確認するのに役立ちます。

コマンド「1」：デバイスによってシステム・レジスタの 0x0000～0x004F に書込みます。

コマンド「2」：デバイスによって任意のレジスタ／メモリ 0x0000～0x7FFF に書込みます。

コマンド「n」：BCに15メッセージを送信するように命令します。

コマンド「n」または、「N」を使用して、BCに15メッセージを送信するように命令します。外部RTがブレーク・アウト・ボードに接続され、アドレスが3の場合、外部RTは15メッセージを受信する必要があります。コンソール・メニューには何も表示されません。BCが送信していることを確認するには、外部RTの接続を解除し、コマンド「b」を使用して内部RTを有効にし、さらに「k」および「t」を押します。これらのコマンドはDev0のRT3を有効にするため、RTトラフィック・データはメッセージ受信後にコンソールに表示されます。コマンド「n」を押すと、以下に示すようなRTトラフィック・データが表示されます。いったん有効化された内部RTを無効化するメニュー・コマンドはありません。Demoプログラムが最初に起動されたときに状態と同じように、コマンド「4」を使用してカードとプログラムをリセットして開始できます。

>n
>

Dev0 MSG #0000. TIME = 00040628us BUS A TYPE0: BC to RT
CMD1 1BC0 --> 03-R-30-00
DATA 0101 0202 0303 0404 0505 0606 0707 0808
0909 1010 1111 1212 1313 1414 1515 1616
1717 1818 1919 2020 2121 2222 2323 2424
2525 2626 2727 2828 2929 3030 3131 3232

STA1 1800

Dev0 MSG #0001. TIME = 00041324us BUS B TYPE1: RT to BC
CMD1 1FC0 --> 03-T-30-00
STA1 1800
DATA 0101 0202 0303 0404 0505 0606 0707 0808
0909 1010 1111 1212 1313 1414 1515 1616
1717 1818 1919 2020 2121 2222 2323 2424
2525 2626 2727 2828 2929 3030 3131 3232

Dev0 MSG #0002. TIME = 00042020us BUS A TYPE1: RT to BC
CMD1 1FC0 --> 03-T-30-00
STA1 1800
DATA 0101 0202 0303 0404 0505 0606 0707 0808
0909 1010 1111 1212 1313 1414 1515 1616
1717 1818 1919 2020 2121 2222 2323 2424
2525 2626 2727 2828 2929 3030 3131 3232

Dev0 MSG #0003. TIME = 00042720us BUS A TYPE0: BC to RT
CMD1 1BC0 --> 03-R-30-00
DATA 0101 0202 0303 0404 0505 0606 0707 0808
0909 1010 1111 1212 1313 1414 1515 1616
1717 1818 1919 2020 2121 2222 2323 2424
2525 2626 2727 2828 2929 3030 3131 3232

STA1 1800

Dev0 MSG #0004. TIME = 00043416us BUS B TYPE1: RT to BC
CMD1 1FC0 --> 03-T-30-00
STA1 1800
DATA 0101 0202 0303 0404 0505 0606 0707 0808
0909 1010 1111 1212 1313 1414 1515 1616
1717 1818 1919 2020 2121 2222 2323 2424
2525 2626 2727 2828 2929 3030 3131 3232

Dev0 MSG #0005. TIME = 00044112us BUS A TYPE1: RT to BC
CMD1 1FC0 --> 03-T-30-00
STA1 1800
DATA 0101 0202 0303 0404 0505 0606 0707 0808
0909 1010 1111 1212 1313 1414 1515 1616
1717 1818 1919 2020 2121 2222 2323 2424
2525 2626 2727 2828 2929 3030 3131 3232

Dev0 MSG #0006. TIME = 00044814us BUS A TYPE0: BC to RT
CMD1 1BC0 --> 03-R-30-00
DATA 0101 0202 0303 0404 0505 0606 0707 0808
0909 1010 1111 1212 1313 1414 1515 1616

1717 1818 1919 2020 2121 2222 2323 2424
2525 2626 2727 2828 2929 3030 3131 3232

STA1 1800

Dev0 MSG #0007. TIME = 00045510us BUS B TYPE1: RT to BC

CMD1 1FC0 --> 03-T-30-00

STA1 1800

DATA 0101 0202 0303 0404 0505 0606 0707 0808

0909 1010 1111 1212 1313 1414 1515 1616

1717 1818 1919 2020 2121 2222 2323 2424

2525 2626 2727 2828 2929 3030 3131 3232

Dev0 MSG #0008. TIME = 00046206us BUS A TYPE1: RT to BC

CMD1 1FC0 --> 03-T-30-00

STA1 1800

DATA 0101 0202 0303 0404 0505 0606 0707 0808

0909 1010 1111 1212 1313 1414 1515 1616

1717 1818 1919 2020 2121 2222 2323 2424

2525 2626 2727 2828 2929 3030 3131 3232

Dev0 MSG #0009. TIME = 00046906us BUS A TYPE0: BC to RT

CMD1 1BC0 --> 03-R-30-00

DATA 0101 0202 0303 0404 0505 0606 0707 0808

0909 1010 1111 1212 1313 1414 1515 1616

1717 1818 1919 2020 2121 2222 2323 2424

2525 2626 2727 2828 2929 3030 3131 3232

STA1 1800

Dev0 MSG #0010. TIME = 00047602us BUS B TYPE1: RT to BC

CMD1 1FC0 --> 03-T-30-00

STA1 1800

DATA 0101 0202 0303 0404 0505 0606 0707 0808

0909 1010 1111 1212 1313 1414 1515 1616

1717 1818 1919 2020 2121 2222 2323 2424

2525 2626 2727 2828 2929 3030 3131 3232

Dev0 MSG #0011. TIME = 00048298us BUS A TYPE1: RT to BC

CMD1 1FC0 --> 03-T-30-00

STA1 1800

DATA 0101 0202 0303 0404 0505 0606 0707 0808

0909 1010 1111 1212 1313 1414 1515 1616

1717 1818 1919 2020 2121 2222 2323 2424

2525 2626 2727 2828 2929 3030 3131 3232

Dev0 MSG #0012. TIME = 00049000us BUS A TYPE0: BC to RT

CMD1 1BC0 --> 03-R-30-00

DATA 0101 0202 0303 0404 0505 0606 0707 0808

0909 1010 1111 1212 1313 1414 1515 1616

1717 1818 1919 2020 2121 2222 2323 2424

2525 2626 2727 2828 2929 3030 3131 3232

STA1 1800


```
Dev0 MSG #0013. TIME = 00049696us BUS B TYPE1: RT to BC
CMD1 1FC0 --> 03-T-30-00
STA1 1800
DATA 0101 0202 0303 0404 0505 0606 0707 0808
    0909 1010 1111 1212 1313 1414 1515 1616
    1717 1818 1919 2020 2121 2222 2323 2424
    2525 2626 2727 2828 2929 3030 3131 3232
```

```
Dev0 MSG #0014. TIME = 00050392us BUS A TYPE1: RT to BC
CMD1 1FC0 --> 03-T-30-00
STA1 1800
DATA 0101 0202 0303 0404 0505 0606 0707 0808
    0909 1010 1111 1212 1313 1414 1515 1616
    1717 1818 1919 2020 2121 2222 2323 2424
    2525 2626 2727 2828 2929 3030 3131 3232
```

コマンド「a」: 100ms ごとに繰り返しメッセージのセットを継続的に送信するように BC に命令します。部分的なリストについては、QSG を参照してください。コマンド「x」: メッセージを停止します。

コマンド「l」: BC Async デモ (コマンド「a」) の実行中に、Bus B 上に 3 つ事前定義された優先度の低い BC メッセージをメッセージ・シーケンスに挿入します。これは、1 度だけ行われます。

```
Dev0 MSG #1694. TIME = 00070698us BUS B TYPE0: BC to RT
CMD1 0822 --> 01-R-01-02
DATA DEAD BEEF
STA1 0800
```

```
Dev0 MSG #1695. TIME = 00071054us BUS B TYPE1: RT to BC
CMD1 0C2F --> 01-T-01-15
STA1 0800
DATA BBBB 0202 1414 0404 0505 0606 0707 0808
    0909 1010 1111 1212 1313 1414 1515
```

```
Dev0 MSG #1696. TIME = 00071210us BUS B TYPE0: BC to RT
CMD1 0825 --> 01-R-01-05
DATA CAFE CODE 0303 0404 0505
STA1 0800
```

コマンド「h」: コマンド「l」と同様の BC Async デモの実行中に、事前定義された高優先度 BC メッセージをメッセージ・シーケンスに挿入しますが、繰り返し可能です。これは、他の 2 つのメッセージの間に挿入されたメッセージ#1553 を示しています。

```
Dev0 MSG #1532. TIME = 00129598us BUS A TYPE2: RT to RT
CMD1 182A --> 03-R-01-10
CMD2 0C2A --> 01-T-01-10
STA1 0800
DATA BBBB 0202 1414 0404 0505 0606 0707 0808
    0909 1010
STA2 1800
```

HoltBCSendAsyncMsgHP returns 0

>

```
Dev0 MSG #1533. TIME = 00042566us BUS B TYPE0: BC to RT
```

```
CMD1 0822 --> 01-R-01-02
DATA DEAD BEEF
STA1 0800
```

```
Dev0 MSG #1534. TIME = 00098222us BUS A TYPE0: BC to RT
CMD1 1822 --> 03-R-01-02
DATA 0005 0002
STA1 1800
```

コマンド「S」: 現在の SMT コマンドとデータ・バッファ・アドレスを示します。この例は、両方のデバイスで BC と両方の RT を有効にし、コマンド「a」と「A」を使用して両方の BC から送信を開始するために必要な手順を示しています。どちらのデバイスも-2F カードでのみ使用できるため、-1F カードでは大文字のコマンドを入力する必要がなく、以下の出力に Dev1 メッセージは表示されません。一部のメッセージは Dev0 からのものであり、一部は Dev1 からのものです。

```
>> b
>c
>k (RTMT Demo)
>t (Traffic Enabled)
>B
>C
>K
RTMT Demo
>a
>A (this will be seen much later inter-mixed in the messages below)
```

With a -2F card a mixed of messages from Dev0 and Dev1 are displayed.

```
Dev0 MSG #0283. TIME = 00130428us BUS A TYPE2: RT to RT
CMD1 182A --> 03-R-01-10
CMD2 0C2A --> 01-T-01-10
STA1 0800
DATA BBBB 0202 1414 0404 0505 0606 0707 0808
0909 1010
STA2 1800
```

```
Dev1 MSG #0284. TIME = 00090538us BUS A TYPE0: BC to RT
CMD1 1822 --> 03-R-01-02
DATA 0005 0002
STA1 1800
```

```
Dev1 MSG #0285. TIME = 00090842us BUS A TYPE2: RT to RT
CMD1 182A --> 03-R-01-10
CMD2 0C2A --> 01-T-01-10
STA1 0800
DATA BBBB 0202 1414 0404 0505 0606 0707 0808
0909 1010
STA2 1800
```

```
Dev0 MSG #0286. TIME = 00099052us BUS A TYPE0: BC to RT
CMD1 1822 --> 03-R-01-02
DATA 0005 0002
STA1 1800
```

```
Dev0 MSG #0287. TIME = 00099356us BUS A TYPE2: RT to RT
CMD1 182A --> 03-R-01-10
CMD2 0C2A --> 01-T-01-10
STA1 0800
DATA BBBB 0202 1414 0404 0505 0606 0707 0808
0909 1010
STA2 1800
```

注記：コマンド「9」（リセット後のみ）を使用して、4つのRTのRTアドレスを変更できます。BCデモ「n」および「a」は、RTアドレス1および、3にメッセージを送信します。BCメッセージRTアドレスは、BCメッセージ関数で設定されます。デモ・コード・ファイル「demo.c」のbcAsync()およびMajorMinorframe()。BCコマンド・ワードでこれらのRTアドレスを変更するには、コードでアドレスを変更する必要があります。

コマンド「0」：Demoモードとユーザー・モードを切り換えます。デフォルトは「Demo」モードです。コマンド「f」：Dev0とDev1の両方に送られるFPGA制御信号の状態を読み取って表示します。信号は、次のセクションの表1にリストされています。Demoモードとユーザー・モードの説明については、次のセクションを参照してください。

```
Press 'M' for menu, or press any valid menu key. >> f
Dev 0 values:
P_BC_TRIG = 0
P_NMR = 1
P_TEST = 0
P_BCENA = 1
P_NTRUM = 1
P_RT1ENA = 1
P_RT2ENA = 1
P_INHIBIT = 0
P_INPUTCONTROL demo/user mode = 0 0=demo, 1=user
P_SPAREINPUT = 0
P_MTPKTRDY = 1
P_ACTIVE = 0
P_RT1MC8 = 1
P_RT2MC8 = 1
P_READY = 1
Dev 1 values:
P_BC_TRIG = 0
P_NMR = 1
P_TEST = 0
P_BCENA = 1
P_NTRUM = 1
P_RT1ENA = 1
P_RT2ENA = 1
P_INHIBIT = 0
P_INPUTCONTROL demo/user mode= 0 0=demo, 1=user
P_SPAREINPUT = 0
P_MTPKTRDY = 1
P_ACTIVE = 0
P_RT1MC8 = 1
P_RT2MC8 = 1
P_READY = 1
>
```

Hardware Description [ハードウェアの説明]

Holt Mini PCIe カードは、PCI Express Mini Card Electromechanical Specification に準拠した 50.95mm × 30mm フォームファクタに 2 つの Holt HI-2130 二重冗長マルチターミナルを備えています。Xilinx XC7A12T FPGA は、2 つの HI-2130 と PCIe Gen2 バス間のインターフェイスとして機能します。電源投入後に PCIe リンクが確立されると、ホストは Linux OS の組み込み PCIe ドライバを使用して、カードと通信します。

FPGA HI-2130 Dev0 and Dev1 control signals [FPGA HI-2130 Dev0 および Dev1 制御信号]

2 つの HI-2130 デバイス (Dev0/Dev1) と通信するために、共通の平行 16Bit アドレスおよび、データ・バスが FPGA に実装されています。これらの信号は回路図の 2 ページに記載されています。

両方のデバイスのターミナル制御信号 (HI-2130) は、I/O レジスタ 0x8000~0x8013 にアクセスするためのメモリマップソフトウェア書き込み機能を使用してホストによって制御される FPGA I/O に接続されます。書き込み可能な信号は、1 を書き込んで High、または、0 を書き込んで Low に設定することができます。例えば、Dev0 の BCENA を High に設定する場合: 「Chan」パラメータは、パラメータを 1 または、0 に設定することにより、Dev0 または、Dev1 を指定します。

```
HoltRegWrite(Chan, P_BCENA, 1); // Chan=0 for Dev0.
```

Demo Mode vs. User Mode [Demo モードと User モード]

ソフトウェア・デモンストレーションを簡単にするために、INPUT_CONTROL (0x800D) が Low に設定されている場合は、Demo モードが選択され、High の場合は User モードが選択されます。電源投入時のデフォルトは Demo モードです。

Demo モードは、一部の制御信号はデフォルトで有効な状態にプリセットされているため、ターミナルはすべて有効になっています (BCENA=1、RTENA=1、MTENA=1 など)。

ユーザー・モードでは、ユーザーは BC、RT、または、MT のイネーブル制御信号の任意の組み合わせを設定して、各デバイスのこれらのターミナルを有効または、無効にできます。User モードでは、2 つの制御信号、BCENA および BCTRIG が J4 コネクタピンから供給されます。

Demo モードまたは、User モードを選択するには、Write 関数を使用します。

```
HoltRegWrite(Chan, INPUT_CONTROL, 0); // 0=demo mode (default)
```

```
HoltRegWrite(Chan, INPUT_CONTROL, 1); // 1=user mode
```

デモ・ソフトウェアが「Demo mode -default」に設定されている場合、FPGA からデバイスへの BCENA0 および BCENA1 ピンはソフトウェアで設定できます。「User」モードでは、両方の BC が J4 コネクタの状態に従います。デフォルトでは、両方の BC の有効化が無効になっています。BC を有効にするには、ユーザーはコネクタの BEEnable ピンを接地する必要があります。BCTRIG0 および BCTRIG1 入力提供されていますが、現在デモでは使用できません。

コンソール・メニュー・コマンド「0」を使用して、User モードと Demo モードを選択できます。コンソール・メニュー・コマンド「f」は、各デバイスからのすべての信号を読み取り、各信号の状態を表示します。

表1 メモリ・マップ I/O デバイス制御信号

デバイス信号	ソース	機能	アドレス	ソフトウェア 読込/書込み	Use
BC_TRIG	FPGA->Device	BC トリガ・パルス	0x8000	Read/Write	High、Low の順にセット (1/0)
NMR	FPGA	デバイス・リセット	0x8002	Read/Write	Low、High の順にセット
TEST	FPGA	Test ピン状態	0x8003	Read/Write	High : テスト・モード テスト・モードについては 6130 データシート参照
BCENA	FPGA	BC 有効	0x8004	Read/Write	BC 有効にするには High にセット
MTRUN	FPGA -> Device	MT 有効	0x8009	Read/Write	MT 有効にするには High にセット
RT1ENA	FPGA -> Device	RT1 有効	0x800A	Read/Write	RT1 有効にするには High にセット
RT2ENA	FPGA -> Device	RT2 有効	0x800B	Read/Write	RT2 有効にするには High にセット
TXINHA/TXINHB	J4 Connector	バスの禁止	0x800C	Read only	J4 へのハード・ワイヤ
INPUT_CONTROL	FPGA	Demo モードまたは、ユーザー・モードの設定	0x800E	Read/Write	0=Demo モード(デフォルト) 1=ユーザー・モード
SPARE_INPUT	J4 Connector	Spare 入力	0x800E	Read only	リード可能な Spare 入力
MTPKTRDY	Device -> FPGA	Monitor Packet 準備完了デバイス出力	0x800F	Read only	オプション・リード
ACTIVE	Device -> FPGA	アクティブ状態	0x8010	Read only	オプション・リード
RT1MC8	Device->FPGA	RT1 モード・コード 8	0x8011	Read only	オプション・リード
RT2MC8	Device->FPGA	RT2 モード・コード 8	0x8012	Read only	オプション・リード
READY	Device->FPGA	Ready 状態を示す	0x8013	Read only	オプション・リード デモ・ソフトウェアで、MR 後にデバイスの準備ができているかどうかを判断するために使用されます

J-TAG & FPGA Boot [J-TAG & FPGA ブート]

ブート SPI フラッシュ・メモリは、出荷時にプログラムされていますが、Xilinx Vivado ツールを使用して、DIGILENT JTAG-HS2 USB プログラミング・ケーブル（付属しません）でユーザーが再プログラム可能です。

FPGA が事前にプログラムされた SPI フラッシュから正常に起動すると、D2 LED が点灯します。再プログラミング後に FPGA コンフィグレーションを再起動するには、SW1 ボタンを押します。通常の動作では、このスイッチは無視できます。

FPGA は 3 つの信号を使用します：コンフィグレーション・オプションの M0、M1 および M2 入力。M0 と M1 はハードワイヤードなので、この設計では M2 のみ変更できます。しかし、デフォルトでは、プルダウン抵抗が M2 を Low に保持し、FPGA が SPI フラッシュから起動するようにコンフィグレーションします。再プログラミングするには、J2 ピン 8（図 3 ボード・リファレンスを参照）を High にプルアップし、M2 を High に駆動する必要があります。通常の使用では、ユーザーが Verilog 設計をカスタマイズし、SPI ブートフラッシュを再プログラミングしたい場合以外は、これらの接続をユーザーが変更する必要はありません。再プログラミングの手順については、このドキュメントの後半で説明します。

表 2 FPGA ブート・オプション

機能	M2 - M1 - M0	注記
ダイレクト FPGA プログラミング	0 - 0 - 1	FPGA をプログラムする（揮発性）
SPI ブートフラッシュのプログラム (デフォルト構成)	1 - 0 - 1	フラッシュから SPI ブートフラッシュ（不揮発性）FPGA ブートをプログラムします

初期化ステータスを提供するために、FPGA によって駆動される 2 つの緑色の LED があります。FPGA の初期化シーケンスが完了すると LED D2 が点灯し、DONE_0 信号を High にアサートします。FPGA がコンフィグレーションリセット状態のときに LED D10 が点灯し、初期化が完了すると消灯します。Xilinx は、FPGA コンフィグレーション・シーケンスに関する完全なテクニカル・ガイドを提供しています。Xilinx UG470 を参照してください。

HI-2130 Parallel Interfaces [HI-2130 パラレル・インターフェイス]

FPGA に実装された 16Bit パラレル・インターフェイスは、両方の HI-2130 デバイスとのインターフェイスに使用されます。アドレスおよび、データ・バスといくつかの他の信号は、両方のデバイス間で共通です。固有のチップセレクトライン nCE0 と nCE1 を使用してそれらの間で選択します。

表 3 FPGA への HI-2130 共通インターフェイス信号

HI-2130	FPGA バンク	プライマリコネクタ	コメント
Address pins: [A15:A0]	14	-	アドレス・バス
Data bus pins: [D15:D0]	14	-	双方向データ・バス
nRE	14	-	リード・ストロブ(Intel mode)
nWE	14	-	ライト・ストロブ(Intel mode)
MCLK50	34	-	FPGA からのマスター50MHz 入力クロック
MTCLK	34	-	オプション・クロック、TBD
TTCLK	34	-	オプション・クロック、TBD

表 4 HI-2130 Device 0

HI-2130	FPGA バンク	J4コネクタ	コメント
nCS0	14		チップ・セレクト 0
nMRO	34		マスター・リセット 0
nIRQ0	34		割り込み出力
TEST0	34		High にアサートされたときのテスト・モード
RT1ENAO	34		FPGA によって RT1 有効に制御
RT2ENAO	34		FPGA によって RT2 有効に制御
RT1MC80	34		RT1MC8 出力、FPGA への入力
RT2MC80	34		RT2MC8 出力、FPGA への入力
CHOBGENABO (BCENA)	34	YES	デフォルトは、プルアップ抵抗により有効です。コネクタによって Low に設定するか、FPGA で上書きします。
CHOBCTRIG (BCTRIG)	34	YES	通常 Low。コネクタでパルスを入力するか、FPGA で上書きします。
MTRUNO	34		FPGA によって制御される入力
READY0, ACTIVE0	34		FPGA への出力、入力。機能は未定
MTPKTRDY0	34		出力、未使用または未定
TXINHA, TXINHB		YES	どちらも、コネクタの CH0INHIBIT0 に接続された入力を禁止します。デフォルトは有効です。両方のトランスミッタを無効にするには、Low をアサートします

表 5 HI-2130 Device 1

HI-2130	FPGA バンク	J4 コネクタ	コメント
nCS1	14		チップ・セレクト 1
nMR1	34		マスター・リセット 1
nIRQ1	34		割り込み出力
TEST1	34		High にアサートされたときのテスト・モード
RT1ENA1	34		FPGA によって RT1 有効に制御
RT2ENA1	34		FPGA によって RT2 有効に制御
RT1MC81	34		RT1MC8 出力、FPGA への入力
RT2MC81	34		RT2MC8 出力、FPGA への入力
CHOBCEENAB1 (BCENA)	34	YES	デフォルトは、プルアップ抵抗により有効です。コネクタによって Low に設定するか、FPGA で上書きします。
CHOBCTRIG (BCTRIG)	34	YES	通常 Low。コネクタでパルスを入力するか、FPGA で上書きします。
MTRUN1	34		FPGA によって制御される入力
READY1, ACTIVE1	34		FPGA への出力、入力。機能は未定
MTPKTRDY1	34		出力、未使用または未定
TXINHA, TXINHB		YES	どちらも、コネクタの CH1INHIBIT1 に接続された入力を禁止します。デフォルトは有効です。両方のトランスミッタを無効にするには、Low をアサートします

J4 コネクタ入力ピン : Transmit Inhibit、BCENAx、BCTRIG は、ESF 保護されていますが、DC 定常状態入力電圧は 3.6V を超えてはなりません。

Power supply [電源]

PCIe コネクタからの 3.3aux 電力は、FPGA の 1V および、1V8 電源レールを生成するダブルバック・コンバータに電力を供給します。「power good1」信号を使用して、FPGA GBT PCIe トランシーバ・レールに 1V および、1V2 を供給する 2 つのリニア電圧レギュレータを有効にします。1V8 は、SPI フラッシュ、MEMS オシレータ、および FPGA バンク 14、15、34 レールに電力を共有する Xilinx の電源シーケンスの推奨を満たすために別の 3V3 電源をオンにするために使用される MOSFET スイッチをオンにするために使用されます。

表 6 電源供給

供給電圧	回路図名	機能	最大電流	消費電流
1V	VCCINT_1V	FPGA ロジック	800mA	200mA
1.8V	VCCAUX_1V8	FPGA 補助機能 (JTAG)	800mA	100mA
3.3V	VCC03V3	FPGA I/O レール、フラッシュ、OSC、その他	1000mA	200mA
3.3V (PCIe conn.)	3V3aux	2×HI-2130	1600mA	1553 メッセージのデューティ・サイクルに依存
1.5V (PCIe conn.)	MGTAVCC1V MGTAVCTT1V2	GBT (PCIe bus) にクリーンな電力を供給します	150mA 150mA	<100mA <100mA

表 7 69Ω に送信される電力とサーマル・ベンチ測定

アクティブチャンネル 0=U7 1=U8	TX デューティ・サイクル %	ICG 電流 (3.3V) 69Ω 負荷	U7 (2130) °C 温度。 自然空冷	U8 (2130) °C 自然空冷
0	100	1.07	70	56
0 & 1	100	1.86	82	78
0 & 1	100	1.86	50 ファンあり	45 ファンあり
0 & 1	100	1.78	81	78
0 & 1	50	1.02	65	63
0	50	.655	59	52
0	10	.34	49	45
0 & 1	10	.42		
0 & 1	IDLE	.26	46	44

注記：ICG は、通常の 75Ω バスカプラインインピーダンスでは低くなります。

表 8 J4 コネクタ・ピン

ピン	名称	説明
1	CHANNEL 0 APOS	MIL-STD-1553 CHO A+ (BUS A)
2	CHANNEL 0 ANEG	MIL-STD-1553 CHO A- (nBUS A)
3	Chassis GND	取付ネジ- 他の接続はありません
4	CHANNEL 0 BPOS	MIL-STD-1553 CHO B+ (BUS B)
5	CHANNEL 0 BNEG	MIL-STD-1553 CHO B- (nBUS B)
6	Chassis GND	取付ネジ- 他の接続はありません
7	CHANNEL 1 APOS	MIL-STD-1553 CH1 A+
8	CHANNEL 1 ANEG	MIL-STD-1553 CH1 A-
9	Chassis GND	取付ネジ- 他の接続はありません
10	CHANNEL 1 BPOS	MIL-STD-1553 CH1 B+
11	CHANNEL 1 BNEG	MIL-STD-1553 CH1 B-
12	Chassis GND	取付ネジ- 他の接続はありません
13	CHOINHBIT0	チャンネル 0 送信禁止。10K プルアップ。2130 Inhibit ピンへの反転直接接続であるインバータに接続します。
14	CHO BCENAB	チャンネル 0 BC 有効。10K プルアップ
15	CHO BCTRIG	チャンネル 0 BC トリガ。10K プルアップ
16	SPARE INPUT	未使用
17	CH1INHBIT1	チャンネル 1 送信禁止。10K プルアップ。2130 Inhibit ピンへの反転直接接続であるインバータに接続します。
18	CH1 BCENAB	チャンネル 1 BC 有効。10K プルアップ
19	CH1 BCTRIG	チャンネル 1 BC トリガ。10K プルアップ
20	Logic GND	

表9 カードテスト・ポイント

テストポイント	信号	回路図ページ	機能
TP1	ACTIVE0, Device0 HI-2130	3	BC/RT/SMT コマンド中に High をアサートします
TP2	Device 0 APOS	3	APOS (BUS A)
TP3	Device 0 ANEG	3	ANEG (nBUS A)
TP4	Device 0 BPOS	3	BPOS (BUS B)
TP5	Device B BNEG	3	BNEG (nBUS B)
TP6	ACTIVE1, Device1 HI-2130	4	BC/RT/SMT コマンド中に High をアサートします
TP7	Device 1 APOS	4	APOS (BUS A)
TP8	Device 1 ANEG	4	ANEG (nBUS A)
TP9	Device 1 BPOS	4	BPOS (BUS B)
TP10	Device 1 BNEG	4	BNEG (nBUS B)
TP11	FPGA GPIO	5	ユーザー実装に利用可能
TP12	ADC 12-bit	5	オプションの ADC、ユーザー実装

TP1 と TP6 (フィードスルーのみ) は、HI-2130 のアクティブ出力で、新しいソフトウェアやその他の設計の変更をチェックするときに役立ちます。ACTIVE は、BC、RT、または、SMT コマンドの実行中に High をアサーとし、ソフトウェアまたは、FPGA 設計の変更をデバッグするときにチェックするための良い出発点として機能します。テスト・ポイント・ボードの配置についてはボード・リファレンスを参照してください。

System Clocks [システム・クロック]

100MHz PCIe システム・クロックは、PCIe コネクタから FPGA に入力されます。このクロック信号は、PCIe に必要な GPT クロック入力ペアに入力され、FPGA ロジックの汎用クロックとしても機能します。オプションの 100MHz クロックは、U13 MEMS 発信器モジュールによってオンボードで提供されます。子のクロックは現在使用されていません。

GPIO

プライマリコネクタから単一の GPIO ピンは、抵抗で絶縁され、ESD 保護され、FPGA ピンにルーティングされます。これは入力または、出力であり、ユーザー定義です。

ADC 12Bit converter [ADC 12 Bit コンバータ]

ADC 入力はテスト・ポイントで提供されますが、この機能は現在の設計では実装されていません。要求がある場合、ユーザーはこれを実装するオプションがあります。

FPGA Spare Pins [FPGA スペア・ピン]

回路図の Spare pin を参照してください。

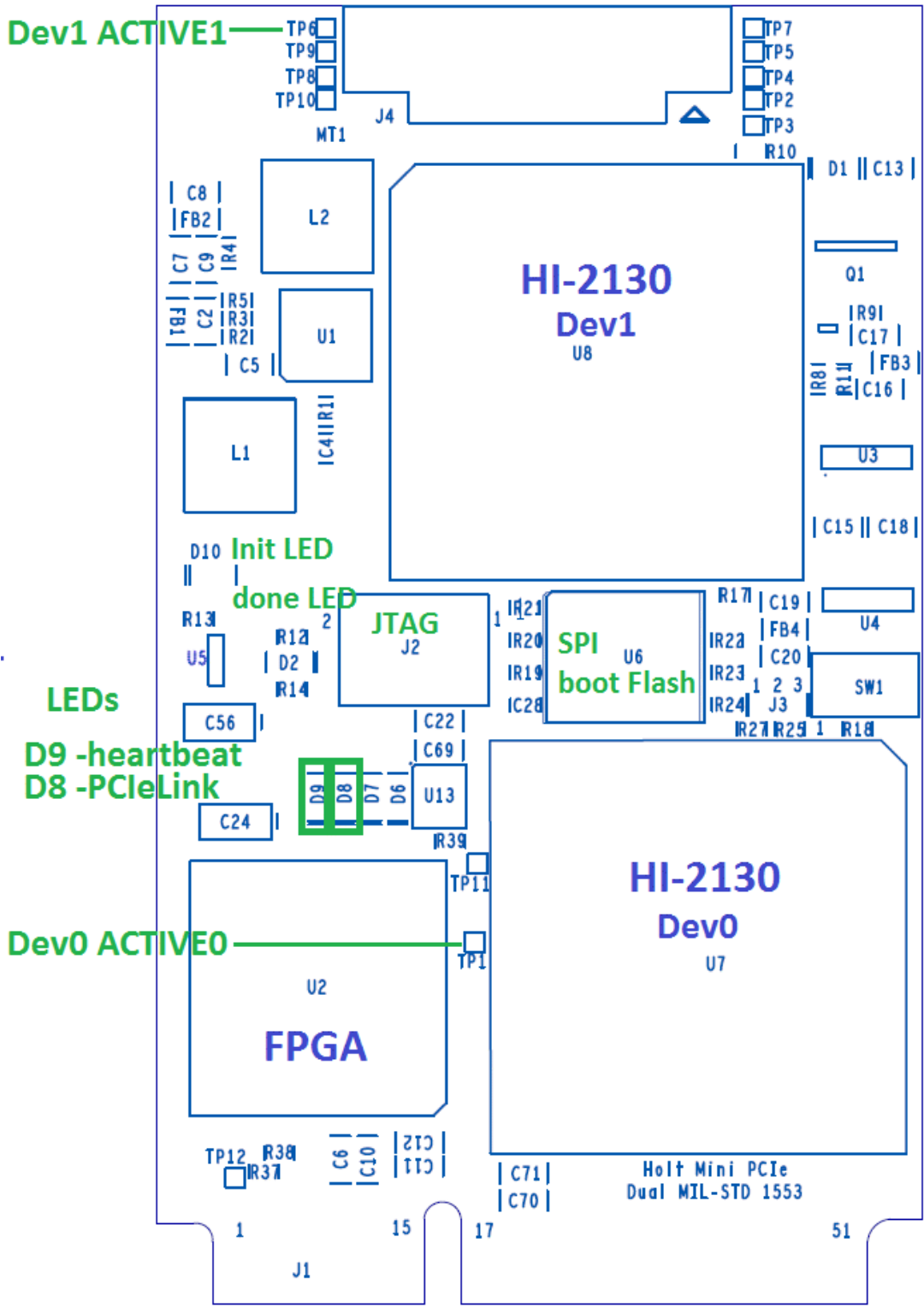


図3 ボード・リファレンス

Summary [まとめ]

このテクニカル・ガイドでは、Holt Mini PCIe Eclipse プロジェクトをインストールして Eclipse にインストールし、プロジェクトをビルドして再実行できるようにするために必要な手順をまとめました。Xilinx FPGA 設計の詳細については、テクニカル・ガイド「AN-MPCIeVivado テクニカル・ガイド」を参照してください。

Item	Qty	Description	Reference	Digikey P/N	Mfg P/N
1	1	PCB, Bare, Evaluation Board	N/A		N/A
2	2	Cap Cer 10pF 10V COG/NPO 0402 SMD	C5,C9	399-8939-1-ND	Kemet C0402C100C8GACTU
3	2	Cap Cer 0.01uF 16V X7R 0402 SMD	C6,C10	399-6297-1-ND	Kemet C0402T103K4RACTU
4	22	Cap Cer 0.1uF 10V X7S 0201 SMD	C4,C14,C21,C23,C28,C35,C37,C39,C40,C41,C42,C44,C46,C47,C48,C49,C50,C54,C60,C65,C66,C67	490-14450-1-ND	Murata GRM033C71A104KE14D
5	1	Cap Cer 0.1uF 10V X7R 0402 SMD	C22	478-7891-1-ND	AVX 0402ZC104KAT2A
6	4	Cap Cer 0.22uF 10V X5R 0402 SMD	C11,C12,C70,C71	490-3910-1-ND	Murata GRM155R61A224KE19D
7	12	Cap Cer 0.47uF 25V X5R 0402 SMD	C27,C29,C30,C31,C33,C34,C52,C59,C62,C63,C64,C69	490-12270-1-ND	Murata GRT155R61E474ME01D
8	4	Cap Cer 1uF 16V X5R 0402 SMD	C15,C16,C18,C19	490-12255-1-ND	Murata GRT155R61C105KE01D
9	4	Cap Cer 2.2uF 16V X5R 0402 SMD	C3,C8,C17,C20	445-9076-1-ND	TDK C1005X5R1C225M050BC
10	6	Cap Cer 4.7uF 10V X5R 0402 SMD	C26,C32,C55,C58,C61,C68	445-13820-1-ND	TDK C1005X5R1A475K050BC
11	4	Cap Cer 10uF 10V X5R 0402 SMD	C1,C2,C7,C13	490-13211-1-ND	Murata GRJ155R60J106ME11D
12	10	Cap Tant 47uF 20% 6.3V 300mOhm 0603	C24,C25,C36,C38,C43,C45,C51,C53,C56,C57	478-9699-1-ND	AVX F380J476MMAAXEH3
13	1	Res 100, 1%, 1/20W 0201 SMD	R7	P122654CT-ND	Panasonic ERJ-1GNF1000C
14	1	Res 226K 1%, 1/20W 0201 SMD	R2	P122842CT-ND	Panasonic ERJ-1GNF2263C
15	1	Res 255K 1%, 1/20W 0201 SMD	R5	P122874CT-ND	Panasonic ERJ-1GNF2553C
16	1	ReS 340K 1%, 1/20W 0201 SMD	R3	P122949CT-ND	Panasonic ERJ-1GNF3403C
17	2	Res 511K 1%,1/20W 0201 SMD	R1,R4	P123057CT-ND	Panasonic ERJ-1GNF5113C
18	1	Res 0, 5%, 1/20W 0201 SMD	R40	P15979CT-ND	Panasonic ERJ-1GN0R00C
19	1	Res 10, 5%, 1/20W 0201 SMD	R39	P123219CT-ND	Panasonic ERJ-1GNJ100C
20	1	Res 15, 5%, 1/20W 0201 SMD	R6	P123244CT-ND	Panasonic ERJ-1GNJ150C
21	1	Res 220, 5%, 1/20W 0201 SMD	R14	P123276CT-ND	Panasonic ERJ-1GNJ221C
22	3	Res 330, 5%,1/20W 0201 SMD	R8,R12,R28	P123304CT-ND	Panasonic ERJ-1GNJ331C
23	5	Res 470, 5%, 1/20W 0201 SMD	R10,R41,R42,R43,R44	P123332CT-ND	Panasonic ERJ-1GNJ471C
24	5	ReS 1K, 5%, 1/20W 0201 SMD	R15,R25,R26,R27,R38	P123221CT-ND	Panasonic ERJ-1GNJ102C
25	1	Res 3.3K, 5%, 1/20W 0201 SMD	R9	P123305CT-ND	Panasonic ERJ-1GNJ332C
26	8	Res 4.7K, 5%, 1/20W 0201 SMD	R13,R18,R19,R20,R21,R22,R23,R24	P4.7AECT-ND	Panasonic ERJ-1GNJ472C
27	1	Res 9.1K, 5%, 1/20W 0201 SMD	R37	P123382CT-ND	Panasonic ERJ-1GNJ912C
28	11	Res 10K, 5%, 1/20W 0201 SMD	R11,R16,R17,R29,R30,R31,R32,R33,R34,R35,R36	P122414CT-ND	Panasonic ERJ-1GNF1002C
29	12	Test Points	TP1-TP12 (DNI)	None	None
30	6	Ferrite 120 Ohm@100Mhz,0402 SMD	FB1,FB2,FB3,FB4,FB5,FB6	490-5192-1-ND	Murata BLM15BD121SN1D
31	2	Inductor fixed 1.5uH,1.75A .047 mOhm	L1,L2	732-1005-1-ND	Würth EL 744031001

Holt Integrated Circuits, Inc.

PCB P/N: HV047

Rev.A

Bill of Materials

EV-2130mPCle-1F

Evaluation Board

32	7	Green Led, Clear 0402 SMD	D1,D2,D6,D7,D8,D9,D10	1497-1219-1-ND	SunLED XZVG68W-2
33	1	Conn 8-Pin 0.8mm Micro Socket	J2	CLE-104-01-G-DV	Samtec CLE-104-01-G-DV
34	1	Through Hole 3-Pin Connector	J3 (DNI)	None	None
35	1	Conn Zero Insertion 20-pos,0.5mm FFC	J4	SAM14910CT-ND	Samtec ZF5S-20-01-T-WT-TR
36	1	Trans Mosfet P-Ch 30V 5A SOT23	Q1	SI2347DS-T1-GE3CT-ND	Vishay SI2347DS-T1-GE3
37	1	Trans NPN 40V 0.2A SOT416	Q2	MMBT3904TT1GOSCT-ND	On MMBT3904TT1G
38	2	TVS Diode 3.6V 8.8V 10USON	D4,D5	296-43875-1-ND	TI TPD4E02B04DQAR
39	1	IC Reg Linear 1V 150mA SOT23-5	U3	497-6871-1-ND	ST LD39015M10R
40	1	IC Reg Linear 1.2V 150mA SOT23-5	U4	497-6872-1-ND	ST LD39015M12R
41	2	IC Inverter 1CH 1-INP SC70-5	U5,U11	296-11600-1-ND	TI SN74LVC1G04DCKR
42	3	IC Inverter 2CH 2-INP SC70-6	U9,U10,U12	296-13262-1-ND	TI SN74LVC2G04DCKR
43	1	Embedded FPGA Artix-7 238-Pin .5mm	U2	XC7A12T-2CPG238I-ND	Xilinx XC7A12T-2CPG238I
44	1	IC Flash 64M SPI 108Mhz 8-WSON	U6	S25FL064LABNFM010-ND	Cypress S25FL064LABNFM010
45	1	IC Reg Buck Adj 0.8A TRPL 16-WFQFN	U1	LTC3545IUD#PBF-ND	Linear LTC3545IUD#PBF
46	1	HI-2130 121-BGA LBxx Low Profile	U7	HI-2130 LBxx	Holt HI-2130 LBxx
47	1	Mems Osc XO 100Mhz LVCMOS	U13	1473-30420-1-ND	SiTime SIT8918BE-73-33E-100G
48	1	SW SPST Momentary	SW1	SW1020CT-ND	Omron B3U-1000P

<i>Item</i>	<i>Qty</i>	<i>Description</i>	<i>Reference</i>	<i>DigiKey</i>	<i>Mfr P/N</i>
1	1	PCB, Bare, Eval Board	N/A	-----	NewTek # 13989
2	1	Conn FFC Bottom 5mm R/A	J2A	SAM14910CT-ND	Samtec ZF5S-20-01-T-WT-TR
3	1	Cable FFC 20 Pos 0.5mm 5" Long	J2B	WM11409-ND	Molex 0152660213
4	1	24-Inch Triax Plug Cable	CH0(AB)	None	MilesTek CA-2014-48
5	1	Aluminum Block 0.75"x2.5"x0.25"	Tie Block	None	OnlineMetals # 1142
6	3	Hex Nut 3/16" Steel 4-40	None	36-4694-ND	Keystone 4694
7	3	Washer Split Lock #4	None	36-4693-ND	Keystone 4693
8	3	Machine Screw Flat #4-40	None	36-9502-ND	Keystone 9502
9	4	Rubber Foot, Bumpon Black , .312 X.200 H	Four corners	SJ5746-0-ND	3M SJ61A1

Item	Qty	Description	Reference	Digikey P/N	Mfg P/N
1	1	PCB, Bare, Evaluation Board	N/A		N/A
2	2	Cap Cer 10pF 10V COG/NPO 0402 SMD	C5,C9	399-8939-1-ND	Kemet C0402C100C8GACTU
3	2	Cap Cer 0.01uF 16V X7R 0402 SMD	C6,C10	399-6297-1-ND	Kemet C0402T103K4RACTU
4	22	Cap Cer 0.1uF 10V X7S 0201 SMD	C4,C14,C21,C23,C28,C35,C37,C39,C40,C41,C42,C44,C46,C47,C48,C49,C50,C54,C60,C65,C66,C67	490-14450-1-ND	Murata GRM033C71A104KE14D
5	1	Cap Cer 0.1uF 10V X7R 0402 SMD	C22	478-7891-1-ND	AVX 0402ZC104KAT2A
6	4	Cap Cer 0.22uF 10V X5R 0402 SMD	C11,C12,C70,C71	490-3910-1-ND	Murata GRM155R61A224KE19D
7	12	Cap Cer 0.47uF 25V X5R 0402 SMD	C27,C29,C30,C31,C33,C34,C52,C59,C62,C63,C64,C69	490-12270-1-ND	Murata GRT155R61E474ME01D
8	4	Cap Cer 1uF 16V X5R 0402 SMD	C15,C16,C18,C19	490-12255-1-ND	Murata GRT155R61C105KE01D
9	4	Cap Cer 2.2uF 16V X5R 0402 SMD	C3,C8,C17,C20	445-9076-1-ND	TDK C1005X5R1C225M050BC
10	6	Cap Cer 4.7uF 10V X5R 0402 SMD	C26,C32,C55,C58,C61,C68	445-13820-1-ND	TDK C1005X5R1A475K050BC
11	4	Cap Cer 10uF 10V X5R 0402 SMD	C1,C2,C7,C13	490-13211-1-ND	Murata GRJ155R60J106ME11D
12	10	Cap Tant 47uF 20% 6.3V 300mOhm 0603	C24,C25,C36,C38,C43,C45,C51,C53,C56,C57	478-9699-1-ND	AVX F380J476MMAAXEH3
13	1	Res 100, 1%, 1/20W 0201 SMD	R7	P122654CT-ND	Panasonic ERJ-1GNF1000C
14	1	Res 226K 1%, 1/20W 0201 SMD	R2	P122842CT-ND	Panasonic ERJ-1GNF2263C
15	1	Res 255K 1%, 1/20W 0201 SMD	R5	P122874CT-ND	Panasonic ERJ-1GNF2553C
16	1	ReS 340K 1%, 1/20W 0201 SMD	R3	P122949CT-ND	Panasonic ERJ-1GNF3403C
17	2	Res 511K 1%,1/20W 0201 SMD	R1,R4	P123057CT-ND	Panasonic ERJ-1GNF5113C
18	1	Res 0, 5%, 1/20W 0201 SMD	R40	P15979CT-ND	Panasonic ERJ-1GN0R00C
19	1	Res 10, 5%, 1/20W 0201 SMD	R39	P123219CT-ND	Panasonic ERJ-1GNJ100C
20	1	Res 15, 5%, 1/20W 0201 SMD	R6	P123244CT-ND	Panasonic ERJ-1GNJ150C
21	1	Res 220, 5%, 1/20W 0201 SMD	R14	P123276CT-ND	Panasonic ERJ-1GNJ221C
22	3	Res 330, 5%,1/20W 0201 SMD	R8,R12,R28	P123304CT-ND	Panasonic ERJ-1GNJ331C
23	5	Res 470, 5%, 1/20W 0201 SMD	R10,R41,R42,R43,R44	P123332CT-ND	Panasonic ERJ-1GNJ471C
24	5	ReS 1K, 5%, 1/20W 0201 SMD	R15,R25,R26,R27,R38	P123221CT-ND	Panasonic ERJ-1GNJ102C
25	1	Res 3.3K, 5%, 1/20W 0201 SMD	R9	P123305CT-ND	Panasonic ERJ-1GNJ332C
26	8	Res 4.7K, 5%, 1/20W 0201 SMD	R13,R18,R19,R20,R21,R22,R23,R24	P4.7AECT-ND	Panasonic ERJ-1GNJ472C
27	1	Res 9.1K, 5%, 1/20W 0201 SMD	R37	P123382CT-ND	Panasonic ERJ-1GNJ912C
28	11	Res 10K, 5%, 1/20W 0201 SMD	R11,R16,R17,R29,R30,R31,R32,R33,R34,R35,R36	P122414CT-ND	Panasonic ERJ-1GNF1002C
29	12	Test Points	TP1-TP12 (DNI)	None	None
30	6	Ferrite 120 Ohm@100Mhz,0402 SMD	FB1,FB2,FB3,FB4,FB5,FB6	490-5192-1-ND	Murata BLM15BD121SN1D
31	2	Inductor fixed 1.5uH,1.75A .047 mOhm	L1,L2	732-1005-1-ND	Würth EL 744031001

Holt Integrated Circuits, Inc.

PCB P/N: HV047

Rev.A

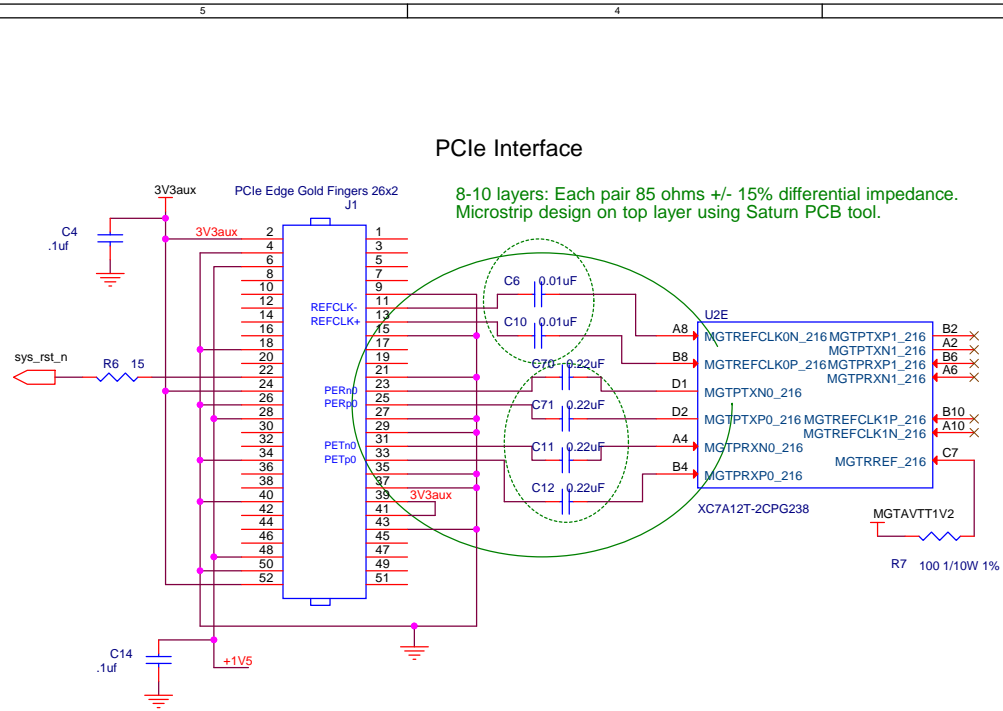
Bill of Materials

EV-2130mPCle-2F

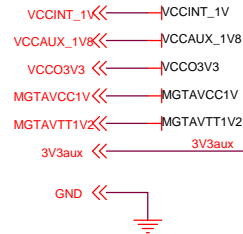
Evaluation Board

32	7	Green Led, Clear 0402 SMD	D1,D2,D6,D7,D8,D9,D10	1497-1219-1-ND	SunLED XZVG68W-2
33	1	Conn 8-Pin 0.8mm Micro Socket	J2	CLE-104-01-G-DV	Samtec CLE-104-01-G-DV
34	1	Through Hole 3-Pin Connector	J3 (DNI)	None	None
35	1	Conn Zero Insertion 20-pos,0.5mm FFC	J4	SAM14910CT-ND	Samtec ZF5S-20-01-T-WT-TR
36	1	Trans Mosfet P-Ch 30V 5A SOT23	Q1	SI2347DS-T1-GE3CT-ND	Vishay SI2347DS-T1-GE3
37	1	Trans NPN 40V 0.2A SOT416	Q2	MMBT3904TT1GOSCT-ND	On MMBT3904TT1G
38	2	TVS Diode 3.6V 8.8V 10USON	D4,D5	296-43875-1-ND	TI TPD4E02B04DQAR
39	1	IC Reg Linear 1V 150mA SOT23-5	U3	497-6871-1-ND	ST LD39015M10R
40	1	IC Reg Linear 1.2V 150mA SOT23-5	U4	497-6872-1-ND	ST LD39015M12R
41	2	IC Inverter 1CH 1-INP SC70-5	U5,U11	296-11600-1-ND	TI SN74LVC1G04DCKR
42	3	IC Inverter 2CH 2-INP SC70-6	U9,U10,U12	296-13262-1-ND	TI SN74LVC2G04DCKR
43	1	Embedded FPGA Artix-7 238-Pin .5mm	U2	XC7A12T-2CPG238I-ND	Xilinx XC7A12T-2CPG238I
44	1	IC Flash 64M SPI 108Mhz 8-WSON	U6	S25FL064LABNFM010-ND	Cypress S25FL064LABNFM010
45	1	IC Reg Buck Adj 0.8A TRPL 16-WFQFN	U1	LTC3545IUD#PBF-ND	Linear LTC3545IUD#PBF
46	2	HI-2130 121-BGA LBxx Low Profile	U7,U8	HI-2130 LBxx	Holt HI-2130 LBxx
47	1	Mems Osc XO 100Mhz LVCMOS	U13	1473-30420-1-ND	SiTime SIT8918BE-73-33E-100G
48	1	SW SPST Momentary	SW1	SW1020CT-ND	Omron B3U-1000P

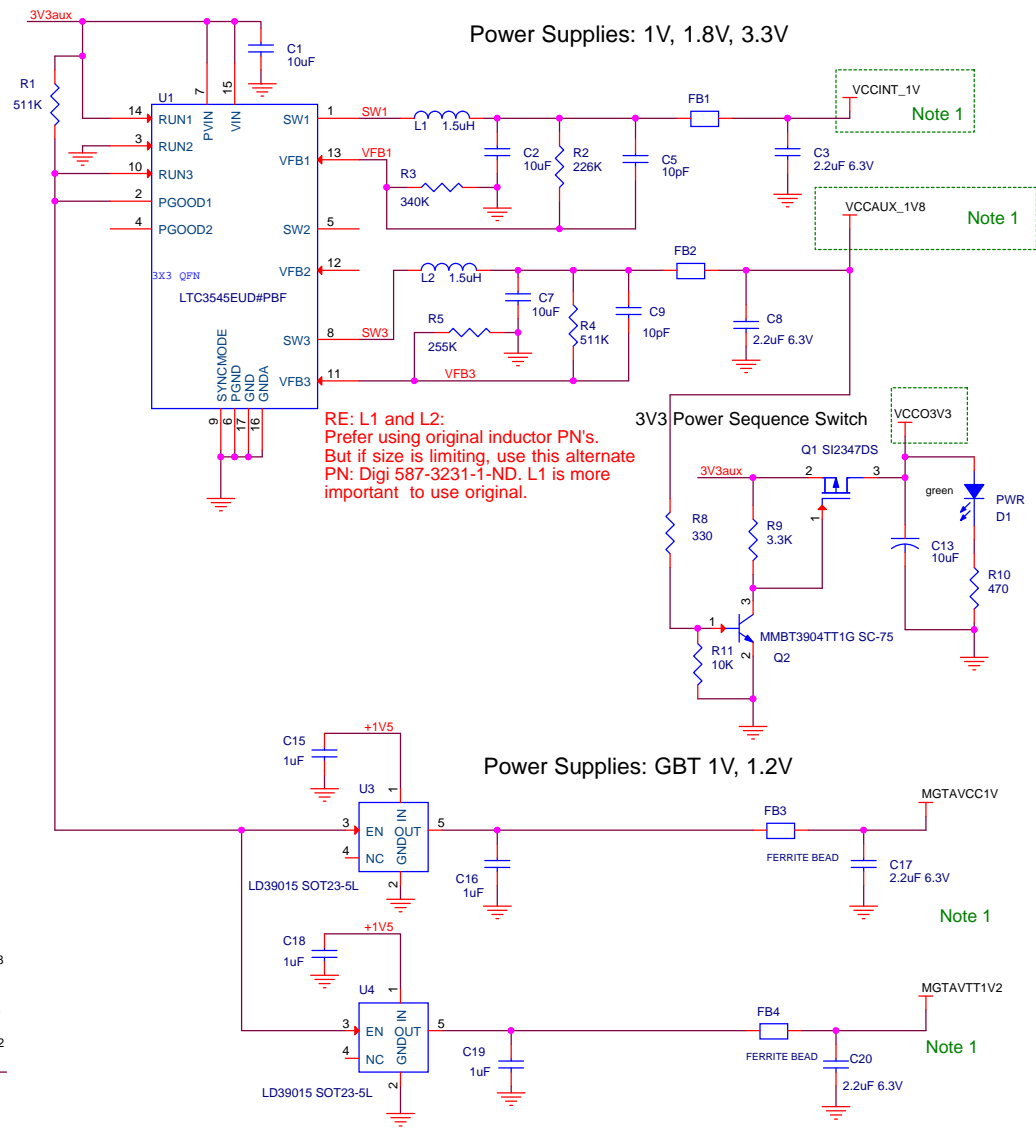
<i>Item</i>	<i>Qty</i>	<i>Description</i>	<i>Reference</i>	<i>DigiKey</i>	<i>Mfr P/N</i>
1	1	PCB, Bare, Eval Board	N/A	-----	NewTek # 13989
2	1	Conn FFC Bottom 5mm R/A	J2A	SAM14910CT-ND	Samtec ZF5S-20-01-T-WT-TR
3	1	Cable FFC 20 Pos 0.5mm 5" Long	J2B	WM11409-ND	Molex 0152660213
4	2	24-Inch Triax Plug Cable	CH0(AB),CH1(AB)	None	MilesTek CA-2014-48
5	1	Aluminum Block 0.75"x2.5"x0.25"	Tie Block	None	OnlineMetals # 1142
6	3	Hex Nut 3/16" Steel 4-40	None	36-4694-ND	Keystone 4694
7	3	Washer Split Lock #4	None	36-4693-ND	Keystone 4693
8	3	Machine Screw Flat #4-40	None	36-9502-ND	Keystone 9502
9	4	Rubber Foot, Bumpon Black , .312	Four corners	SJ5746-0-ND	3M SJ61A1



Where possible parts were selected for bottom height requirement < 1.35mm to allow back side placement.

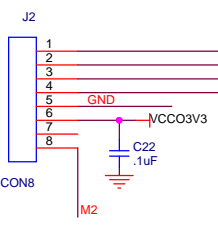


Note 1: Distribute power over planes to where they need to go under the FPGA as much as possible.



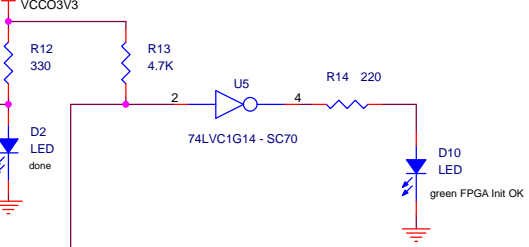
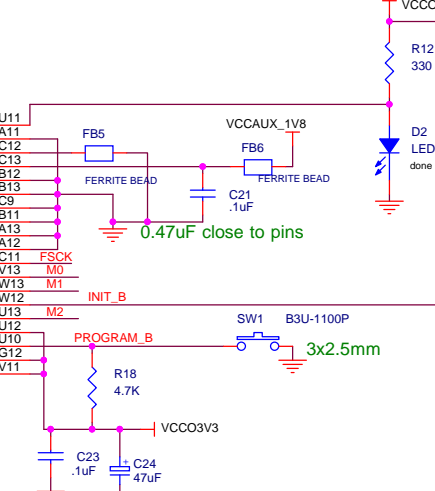
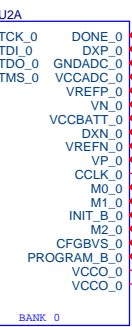
Title		
Holt PCIe-Mini dual MIL-STD 1553		
Size	Document Number	Rev
B	<Doc>	A
Date:	Thursday, January 16, 2020	Sheet 1 of 5

JTAG



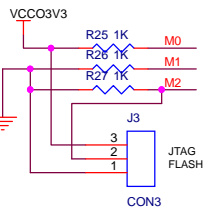
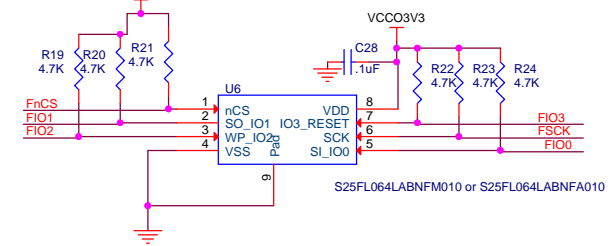
CONFIG. BANK

XC7A12T-2CPG238

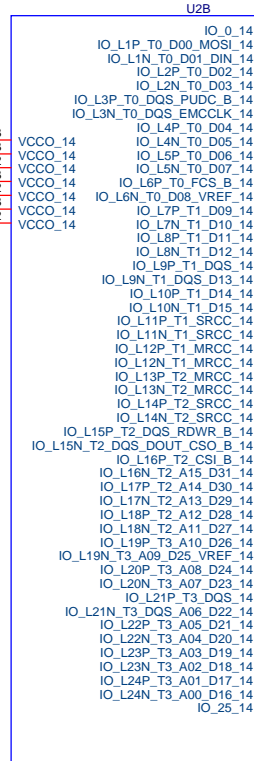
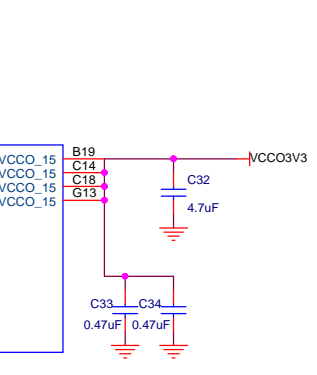
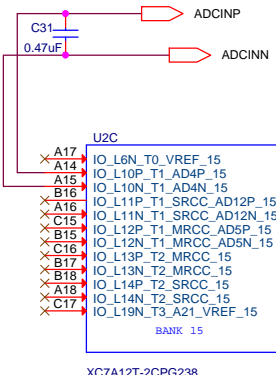


FPGA BOOT FLASH

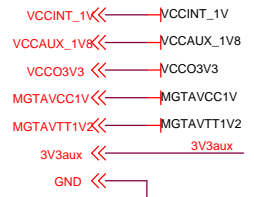
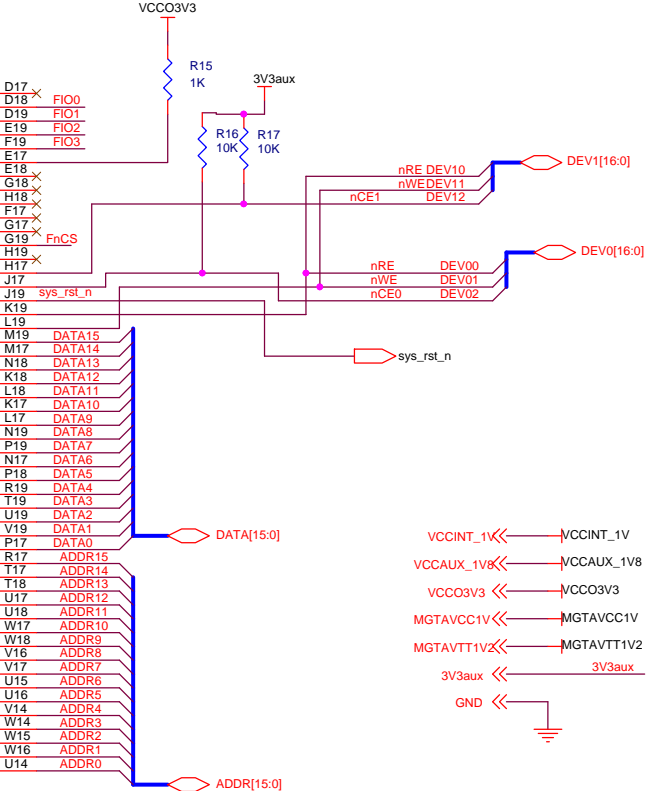
S25FL064LABNFM10 or S25FL064LABNFA010



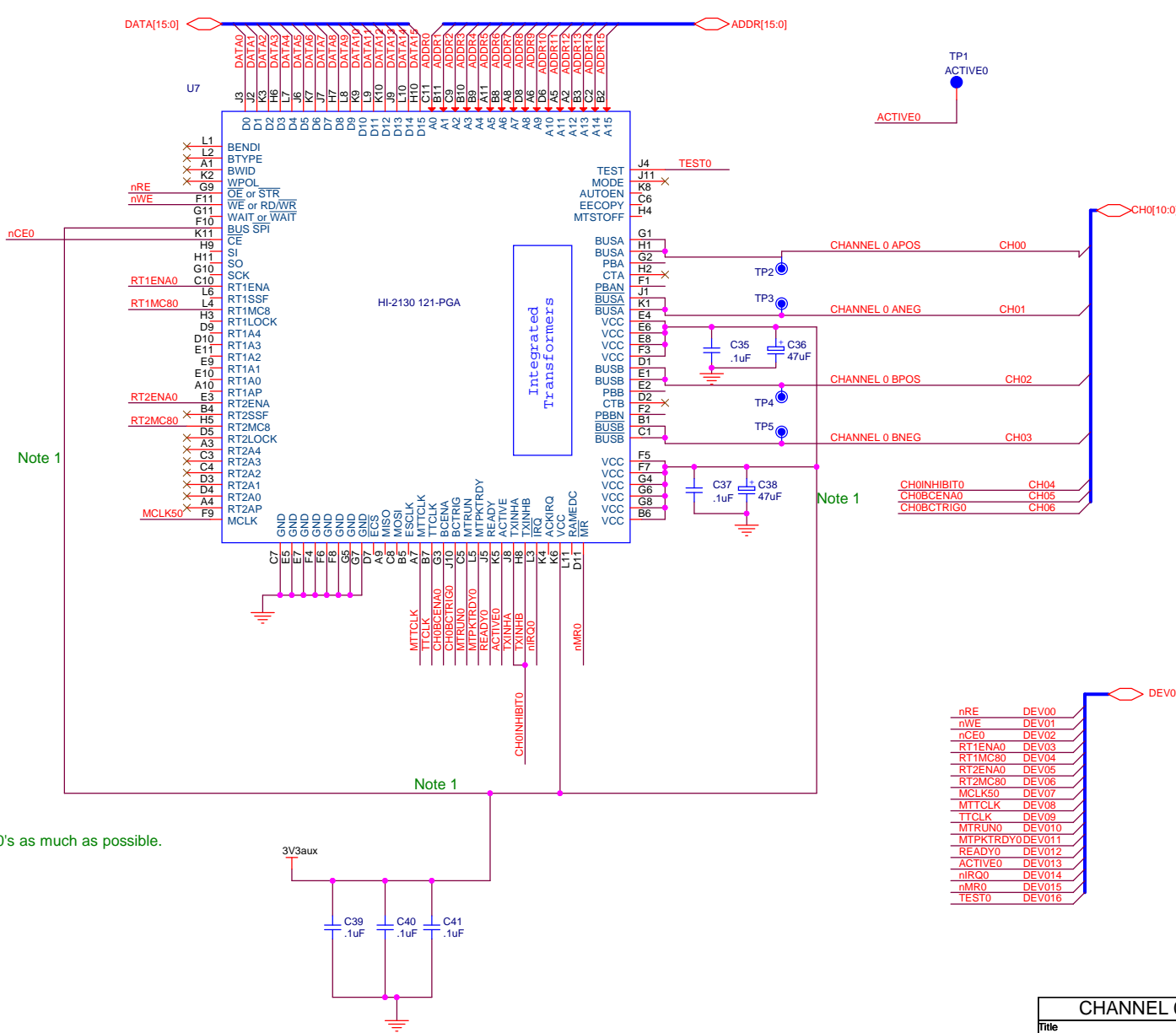
	M2	M1	M0
FLASH	0	0	1
JTAG	1	0	1



XC7A12T-2CPG238

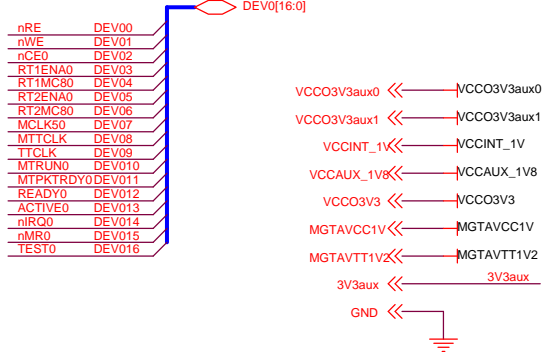


Title		
Holt PCIe-Mini dual MIL-STD 1553		
Size	Document Number	Rev
B	<Doc>	A
Date:	Thursday, January 16, 2020	Sheet 2 of 5



Note 1:
Distribute power using planes to 2130's as much as possible.

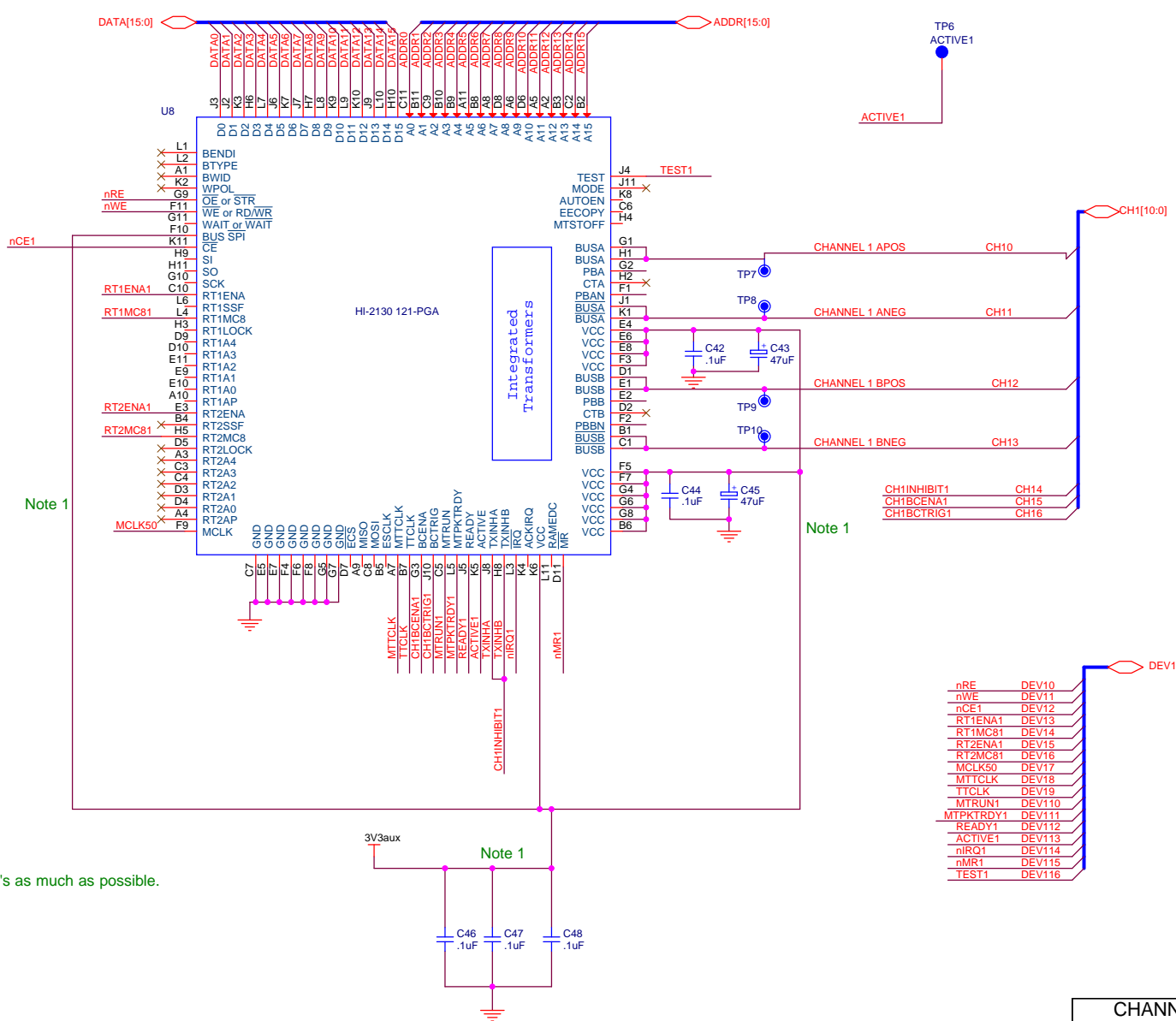
Note 1



CHANNEL 0		
Title Holt PCIe-Mini dual MIL-STD 1553		
Size B	Document Number <Doc>	Rev A
Date: Monday, September 30, 2019	Sheet 3	of 5

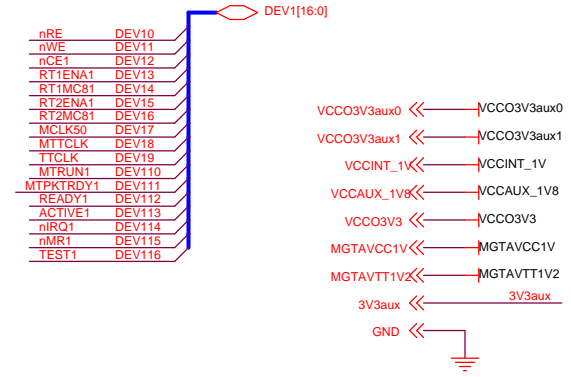
5 4 3 2 1

E D C B A



Note 1: Distribute power using planes to 2130's as much as possible.

Note 1



CHANNEL 1		
Title Holt PCIe-Mini dual MIL-STD 1553		
Size B	Document Number <Doc>	Rev A
Date:	Monday, September 30, 2019	Sheet 4 of 5

