

## **AN-6130PCle**

### **MIL-STD-1553 PCle 評価カード ユーザー・ガイド**

本文書は、Holt 社の『AN-6130PCle』の和訳になります。

分かりにくい表現や誤訳がある場合は、Holt 社発行の英語版文書を参照してください。



株式会社ナセル

## Introduction [はじめに]

Holt HI-6130 MIL-STD-1553 PCIe カードは、Holt HI-6130 MIL-STD-1553 マルチ・ターミナルを Windows7 が動作している PC スロットで、レーン x1、x4/x8 または x16 PCI Express (PCIe) 1.1 に接続するように設計されたス PC カード・インターフェイスです。HI-6130 は、MIL-STD-1553 二重冗長バス通信の単一電源 3.3V 動作の BC/MT/RT1/RT2 マルチターミナル・デバイスです。このカードには Holt ハイレベル・ソフトウェア・ライブラリと 2 つのデモ・プログラムがバンドルされています。付属の CD-ROM に収録されている 2 つのデモ・プロジェクトは、HI-6130 と Holt API ソフトウェア・ライブラリの基本的な機能を示しています。デモ・カードとソフトウェアは、新しいカスタム設計の出発点として使用できます。



## Evaluation Kit Contents [開発キット構成]

- 本アプリケーション・ノート、AN-6130PCIe ユーザー・ガイド
- Holt API ライブラリ・ソフトウェアおよび、ユーザー・マニュアル
- ANSI C HI-6130PCIe テスト・デモ・プロジェクト
- ANSI C HI-6130PCIe API デモ・プロジェクト
- 9 ピン Dsub から MIL-STD-1553 BNC ブレークアウト・ケーブル
- フル・サイズ PCI カード・オプション・ブラケット
- CPLD Verilog プロジェクト・ファイル
- サポート・ドキュメントとソフトウェアを含む CD-ROM

## Topics [目次]

- Introduction [概要]
- What's new in this release [変更履歴]
- Board setup [ボード設定]
- Quick Start Guide [クイック・スタート・ガイド]
- Hardware [ハードウェア]
- Programming Reference [プログラミング・リファレンス]
- Software [ソフトウェア]
- Customization [カスタマイズ]
- Summary [まとめ]
- Schematics and BOM [回路図、部品表]

## New in this release: [このリリースの新機能]

- PLX SDK、7.11 から 7.20 に更新
- Holt API ライブラリ、Rev.02-1-2 に更新
- Holt API デモ、Rev.2.0 に更新
- Holt API デモ、API デモのために更新

## Board Default Setup: [ボード・デフォルト設定]

SW2 のポジション 6 を OFF (上) の位置に設定します。デモが正常に動作するために必要な HI-6130 入力ピン AUTOEN を Low に設定します。下の図を参照してください。

JP6 と JP7 はオープンです。バス・トランス (nXBUS) の負側をテスト専用設置する場合は、必要に応じてこれらをジャンパします (オプション)。

JP2-JP5 は使用されていません。使用方法については、PLX PEX8311 PCI Express マニュアルを参照してください。

## Quick Start Guide [クイック・スタート・ガイド]

このボードは Avago Technologies (Avagotech.com) が現在所有する PLX Technology 製 PCIe ブリッジを使用して PC と通信します。

このガイドのソフトウェア・セクションの「PLX SDK のインストール」(P13) の手順に従って、PLX SDK をインストールします。あらかじめ作成された Windows 7 と互換性のある実行可能なデモ・プログラムは、CD-ROM のフォルダ「Holt HI-6130 PCIe Demo」に含まれています。このフォルダをデスクトップにコピーします。

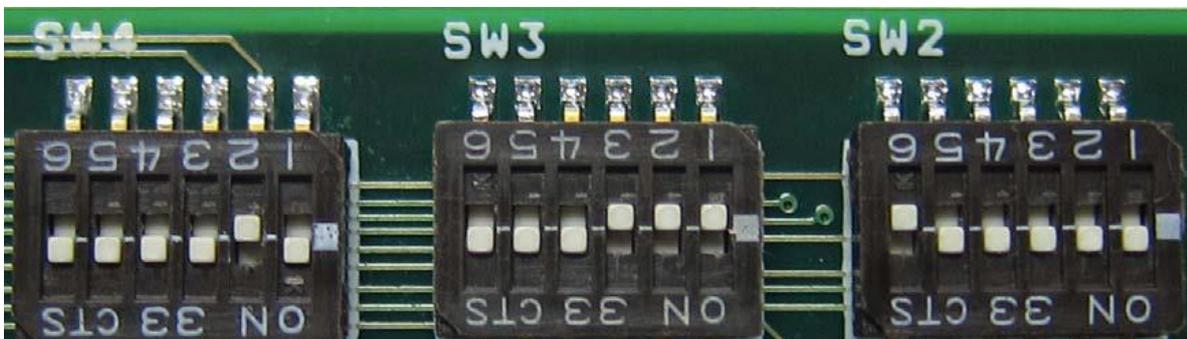
工場出荷時の DIP スイッチの設定を確認します (論理 0 は下、論理 1 は上)。

RT1 アドレス = 3、SW3 1-3 OFF、4-6 ON

RT2 アドレス = 1、SW4 1、3-6 ON、2 OFF

SW2 ポジション 1~5 はユーザー定義です。これらは将来リリースのデモ・プログラムで使用される可能性があります。

デフォルト DIP スイッチ設定

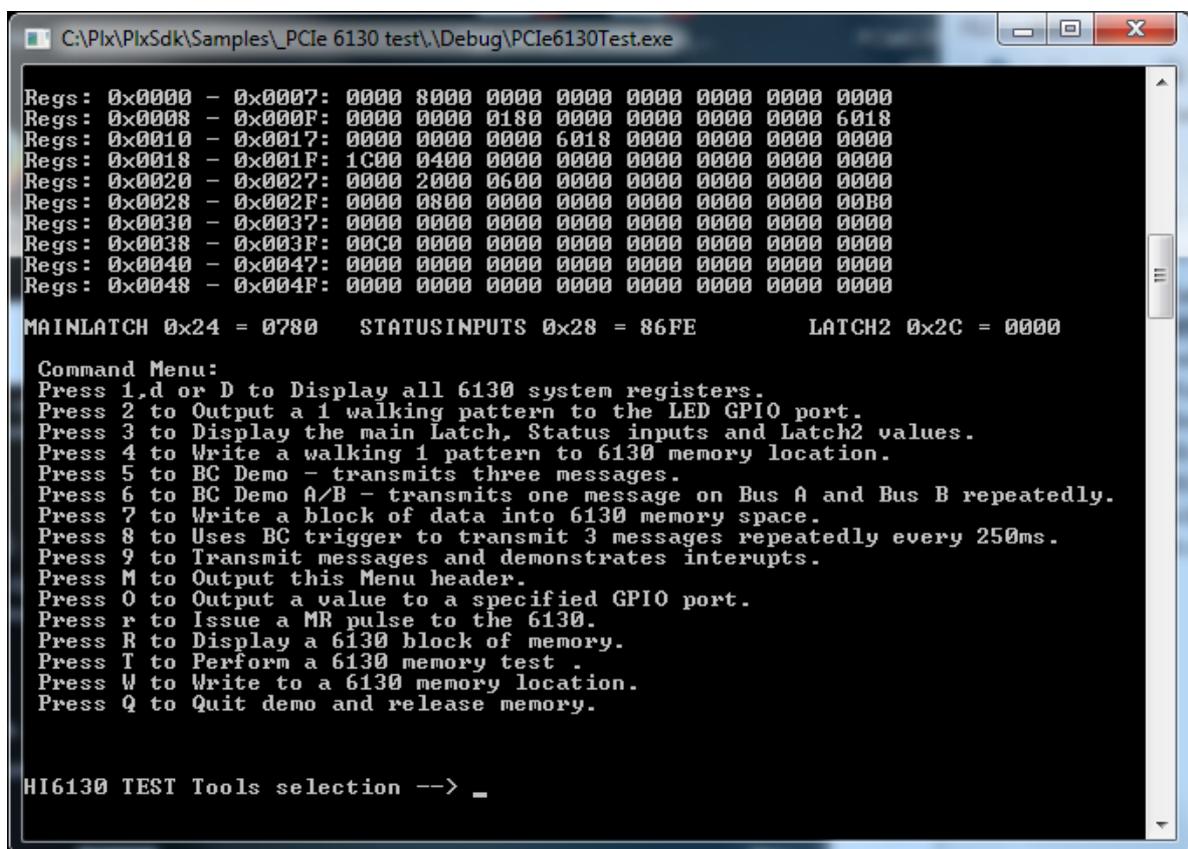


PCIe カードの金属製ブラケットは、フルハイトとロープロファイルの両方が用意されています。ご使用の PC スロットに合わせてブラケットを使用してください。

PC の電源が入っていない状態で、PCIe x1、x4/8 または x16 カードスロットにカードを差し込み、ガードが固定されるようにブラケット・ネジでカードを固定します。PC の電源を入れた後、Windows は新しいハードウェアを自動的に検出し、PLX SDK インストールによってインストールされたドライバを使用します。このデモで問題が発生した場合は、13 ページの「PLX SDK のインストール」に進み、PLX SDK のインストール後に Visual Studio を使用してプログラムを起動してください。

ドライバがインストールされたら、Holt HI-6130 Demo フォルダ（デスクトップ上にある）にあるアプリケーションをダブルクリックして、PCIe6130Test アプリケーション (.exe) を起動します。1 から x までの数字のセットを示すメニューが表示されています。Holt カードは、通常最初のアイテム #1 として “9056 10b5” [b:xx s:xx f:xx] と表示されます。「1」を入力して Enter を押します。

コンソール・ウィンドウに表示されるメインの命令・メニューを以下に示します。



```

C:\Plx\PlxSdk\Samples\_PCIe 6130 test\_Debug\PCIe6130Test.exe
Regs: 0x0000 - 0x0007: 0000 8000 0000 0000 0000 0000 0000 0000
Regs: 0x0008 - 0x000F: 0000 0000 0180 0000 0000 0000 0000 6018
Regs: 0x0010 - 0x0017: 0000 0000 0000 6018 0000 0000 0000 0000
Regs: 0x0018 - 0x001F: 1C00 0400 0000 0000 0000 0000 0000 0000
Regs: 0x0020 - 0x0027: 0000 2000 0600 0000 0000 0000 0000 0000
Regs: 0x0028 - 0x002F: 0000 0800 0000 0000 0000 0000 0000 0000
Regs: 0x0030 - 0x0037: 0000 0000 0000 0000 0000 0000 0000 0000
Regs: 0x0038 - 0x003F: 00C0 0000 0000 0000 0000 0000 0000 0000
Regs: 0x0040 - 0x0047: 0000 0000 0000 0000 0000 0000 0000 0000
Regs: 0x0048 - 0x004F: 0000 0000 0000 0000 0000 0000 0000 0000

MAINLATCH 0x24 = 0780   STATUSINPUTS 0x28 = 86FE   LATCH2 0x2C = 0000

Command Menu:
Press 1, d or D to Display all 6130 system registers.
Press 2 to Output a 1 walking pattern to the LED GPIO port.
Press 3 to Display the main Latch, Status inputs and Latch2 values.
Press 4 to Write a walking 1 pattern to 6130 memory location.
Press 5 to BC Demo - transmits three messages.
Press 6 to BC Demo A/B - transmits one message on Bus A and Bus B repeatedly.
Press 7 to Write a block of data into 6130 memory space.
Press 8 to Uses BC trigger to transmit 3 messages repeatedly every 250ms.
Press 9 to Transmit messages and demonstrates interrupts.
Press M to Output this Menu header.
Press 0 to Output a value to a specified GPIO port.
Press r to Issue a MR pulse to the 6130.
Press R to Display a 6130 block of memory.
Press I to Perform a 6130 memory test.
Press W to Write to a 6130 memory location.
Press Q to Quit demo and release memory.

HI6130 TEST Tools selection --> _

```

「1」を押すと、HI-6130 レジスタのメモリ・ダンプが実行されます。これによりマスター・リセット後に初期化される 0x0000~0x0047 のすべての HI-6130 システム・レジスタが表示されます。デフォルト値は、HI-6130 データシートで指定されています。アドレス 0x0000 のワードに 0x8000 が含まれていることに注目してください。これは、HI-6130 マスタ・ステータスとリセット・レジスタ (0x0001) です。MSB の Bit15 High は、HI-6130 READY が High であることを示します。これはホストがデバイスメモリとレジスタにアクセスする準備ができていることを意味します。レジスタとステータス・ビットの定義の詳細については HI-6130 のデータシートを参照してください。

---

HI-6130 BC は、いくつかの定義済みの MIL-STD-1553 メッセージで初期化されます。

「5」を押すと、BC が 3 つの (コマンド) メッセージを RT アドレス 3 に送信するように指示します。メッセージ 1: バス A/SA1/32 データワード/受信コマンド、メッセージ 2: バス B/SA1/32 データワード/送信コマンド、メッセージ 3: バス A/SA1/32 データワード/受信コマンドです。

別の RT ターミナルまたは、RT 他の RT 対応 MIL-STD-1553 試験装置が RT アドレス 3 に設定され、適切なバス結合器を介してバスに接続されている場合、メッセージ応答を監視することができます。このプログラムは、各送信後に、トランザクションされた BC データブロック、BC 命令リスト、および HI-6130 システム・レジスタをコンソールに出力します。オシロスコープで送信を表示するには、カードの左上の ACTIVE テスト・ポイントの立ち上がりをトリガします; 別のスコープ・プローブを ABBUS テスト・ポイントに取り付けます。小さなクリップ・リード線を使用して、この信号を持ち上げてスコープ・プローブにアクセスできるようにします。外部 RT またはテスト機器がバスに接続されていない場合は、ケーブル出力に 70Ω の終端抵抗を使用します。そうでない場合、オシロスコープで見ると信号が歪んでしまいます。

BC が 3 つのメッセージを RT アドレス=3 に繰り返し送信するように「8」を押します。メッセージ 1 と 3 はバス A を使用し、メッセージ 2 はバス B を使用します。

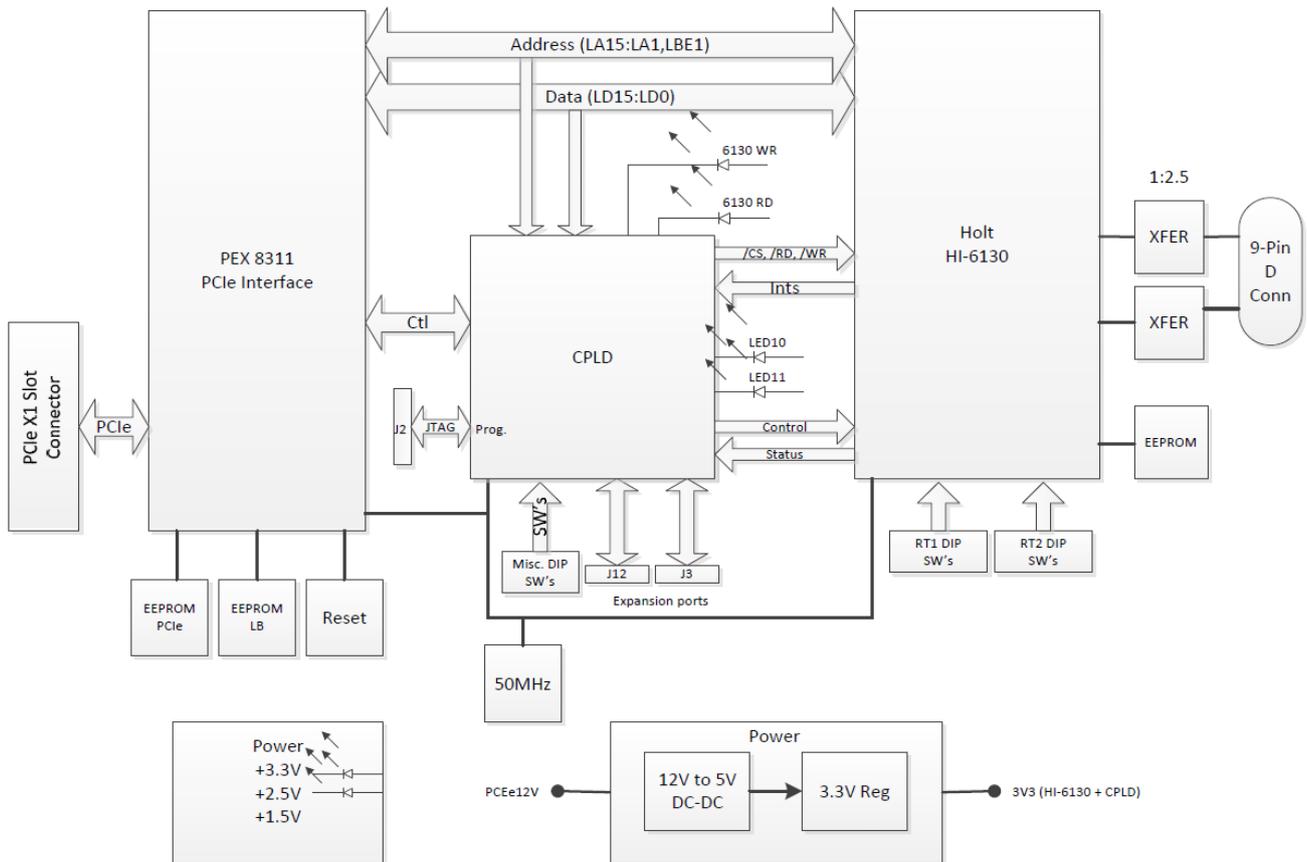
送信中、カードの上端にある LED10 および、11 は、送信されたメッセージに従ってバイナリ方式でカウントされます。ソフトウェア開発中に視覚的補助を提供するために、HI-6130 がプログラムによってリード/ライトされると、2 つの LED が点滅します。リードが発生すると LED8 が点滅し、ライトが発生すると LED9 が点滅します。

「q」を押すと終了します。

これはクイック・スタート・ガイドのセクションの最後です。

## Hardware [ハードウェア]

Holt HI-6130 PCIe インターフェイス・カードは、ブロック図のように以下の 3 つの主要 IC コンポーネントで構成されています。PEX 8311 は PCI Express から汎用バスへのブリッジで、PCIe スロットとローカル・バス間のインターフェイスを提供します。プログラムされた CPLD は、LB 信号を HI-6130 タイミング用の CSn、RDn、および WRn ストローブ信号に変換します。CPLD は他の GPIO グルー・ロジックも提供します。共有の単一 50MHz 発振器モジュールは、PEX8311 ローカル・バス、CPLD および HI-6130 用のクロックを提供します。HI-6130 のクロックは 50MHz でなければなりません。



HI-6130 PCIe カード・ブロック図

PEX 8300 PCIe ブリッジは、ホスト PC に対してスレーブモードで動作する「EndPoint」モードで使用されます。ドライバ、ドキュメント、デモ・ソフトウェアを含む SDK と RDK リファレンス・ボードの完全な設計は、PLX から入手できます。この PLX RDK は、Holt カードのベースとして使用されました。EEPROM (U6) には、ターゲット・ハードウェア・ベースアドレス空間、データバス幅、およびウェイト・ステート数のために LB を構成するために、電源投入時に PEX8311 がラッチする構成データが含まれています。データバス幅は、HI-6130 の幅に合わせて 16Bit に設定されています。2 番目の EEPROM も PEX8311 に接続され、PCIe 側を中心とした特別な設定を可能にするオプションのパラメータが用意されています。オプションの PCIe バス構成用に 2 番目の EEPROM (U2) が搭載されていますが、使用しても使用しなくてもかまいません。PLX は将来のニーズに備えてこの EEPROM をインストールすることを推奨します。これらの EEPROM は、PlxMon ユーティリティを使用して検査およびプログラミングすることができます。PLX ユーティリティを使用して、検査およびプログラミングすることができます。PLX ユーティリティ・プログラム「PlxMon.exe」は、SDK のインストール後に「¥Plx¥...¥Bin」フォルダにあります。このユーティリティの使用に関する情報は、4 章「PLX デバッグ・ユーティリティ」の PlxSdkUsersManual ドキュメントに記載されています。Holt はこのユーティリティを使用して 2 つの EEPROM をプログラムします。

HI-6130 は、リード/ライト中は 16Bit のデータバス、16Bit のアドレスバス、さらにバス制御入力信号を使用します。CS0<sub>n</sub>, WR<sub>n</sub>, RD<sub>n</sub> 信号は CPLD によってデコードされ、HI-6130 に送られます。LB の上位アドレスライン (LA31 : LA26) のみがチップ・セレクト用の CPLD によってデコードされます。LB アドレスとデータラインは PEX8311 から HI-6130 に直接接続します。これらのラインに加えて、CPLD から HI-6130 へと以下に列挙された、いくつかの専用入出力があります。HI-6130 の詳細な説明については、CD-ROM に収録されている Holt データシートおよび、アプリケーション・ノートを参照してください。

CPLD によって提供される機能 :

- HI-6130 タイミングを満たす HI-6130 バス・インターフェイス信号 CS<sub>n</sub>, RD<sub>n</sub>, WR<sub>n</sub>
- HI-6130 リード/ライト・アクセス LED
- HI-6130 および LED への制御信号のリードバック付きラッチ
- HI-6130 ステータスと DIP スイッチ。

CPLD は、HI-6130 IRQ、RT1MC8、および RT2MC8 信号を PLX8311 の割り込みピンに接続された単一信号に論理 OR します。PLX API が割り込みを処理する方法については、ソフトウェアのセクションを参照してください。PLX には、割り込みの処理方法を示すデモ・プロジェクト「LocalToPciInt」が含まれています。「LocalToPciInt」で使用されていた方法は、Holt デモで使用されています。

## Programming Reference (LB decoded addresses) [プログラミング・リファレンス]

HI-6130 chip select (R/W)—0x2000-0000-0x2000-07FFF (32K Words).

### Main Output Control Latches (R/W)—0x2400-0000

BIT	15	14	13	12	11	10	9	8
FIELD	TEST	EECOPYY	"K3"	LED2	LED1	RT2ENA	RT1ENA	BCENA
RESET	0	0	0	1	0	0	0	0

BIT	7	6	5	4	3	2	1	0
FIELD	MR	MTRUN	TXINHA	TXINHB	RAMDEC	TP31	ACKIRQ	BCTRIG
RESET	0	0	1	1	0	0	0	0

D0	BCTRIG	HI-6130 入力。立ち上がりエッジは、BC が次のオペコード命令を実行するようにトリガする。
D1	ACKIRQ	HI-6130 入力
D2	TP31	CPLD 予備ピン
D3	RAMEDC	HI-6130 エラー検出／訂正入力。このプログラムでは Low を設定します。
D4	TXINHB	HI-6130 バス B 入力禁止
D5	TXINHA	HI-6130 バス A 入力禁止
D6	MTRUN	HI-6130 MT 有効入力
D7	/MR	HI-6130 マスター・リセット入力
D8	BCENA	HI-6130 BC 有効入力
D9	RT1ENA	HI-6130 RT1 有効入力
D10	RT2ENA	HI-6130 RT2 有効入力
D11	/LED1	汎用 LED (オンボード LED10)。電源投入時に ON (Low)
D12	/LED2	汎用 LED (オンボード LED11)。電源投入時に ON (High)
D13	"K3"	未使用。CPLD から PCB 上のパッドに引き出されています。
D14	EECOPYY	HI-6130 EECOPYY 入力
D15	TEST	HI-6130 TEST 入力。通常動作の場合は Low に設定する必要があります。テスト・モードの詳細については、データシートを参照してください。

## Status Inputs (R only) – 0x2800-0000

BIT	15	14	13	12	11	10	9	8
FIELD	“A4”	SW2-5	SW2-4	SW2-3	SW2-2	SW2-1	“D5”	AUTOEN

BIT	7	6	5	4	3	2	1	0
FIELD	N/A	IRQ	RT2MC8	RT1MC8	WAIT	READY	MTPKTRDY	ACIVE

D0	ACTIVE	BC または、RT が 1553 メッセージを処理しているとき、HI-6130 は High を出力します。
D1	MTPKTRDY	HI-6130 出力。メッセージが完了するとモニタ・パケットをアクティブ High します。
D2	READY	HI-6130 出力。リセット後、ホストがデバイスを設定できるときに High になります。
D3	WAIT	HI-6130 出力。この設計では使用されないバス制御信号
D4	/RT1MC8	HI-6130 出力。RT1 がモードコード 8 を受信するとパルスを出力しリセットされます。
D5	/RT2MC8	HI-6130 出力。RT1 がモードコード 8 を受信するとパルスを出力しリセットされます。
D6	/IRQ (6130)	HI-6130 割り込み出力。
D7	N/A	未定義
D8	AUTOEN	SW2 DIP スイッチ 6 で設定します。EEPROM から自動初期化するための HI-6130 への入力。
D9	“D5”	「D5」から CPLD まで PCB 上のパッドに接続して使用しません。
D10	SW2-1	ユーザー定義 DIP スイッチ
D11	SW2-2	ユーザー定義 DIP スイッチ
D12	SW2-3	ユーザー定義 DIP スイッチ
D13	SW2-4	ユーザー定義 DIP スイッチ
D14	SW2-5	ユーザー定義 DIP スイッチ
D15	“A4”	使用されていませんが、「A4」から CPLD へのパッドに接続されています。

## Main Output Control Latches (R/W) – 0x2400-0000

BIT	15-5	4	3	2	1	0
FIELD	N/A	RT2LOCK	RT2SSF	MTSTOFF	RT1LOCK	RT1SSF
RESET	N/A	0	0	0	0	0

D0	RT1SSF	RT1 サブシステム・フェイル入力
D1	RT1LOCK	RT1 RT アドレス入力ロック入力
D2	MTSTOFF	HI-6130 メモリ・テスト無効。内部プルダウン抵抗で Low に設定してください。
D3	RT2SSF	RT2 サブシステム・フェイル入力
D4	RT2LOCK	RT2 RT アドレス入力ロック入力
D5:D15	N/A	未定義

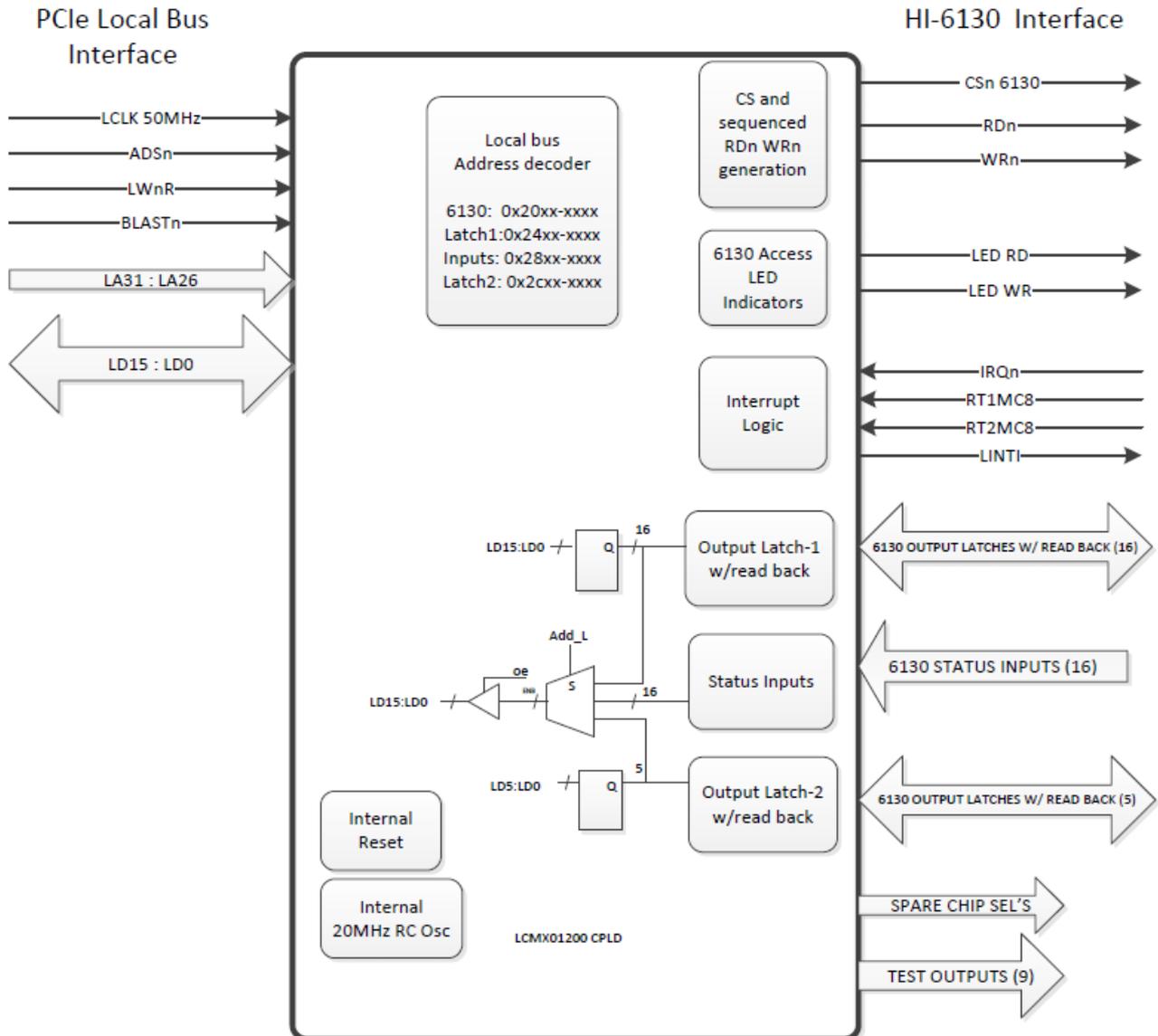
CPLD は Lattice LCMX01200 256 ピン BGA デバイスで、J2 の JTAG ポートを介してプログラムされています。Verilog のソース・コード・リストはこのドキュメントに含まれており、実際の Verilog ソース・ファイルと制約ファイルは CD-ROM に含まれている Lattice の「Diamond」ツール・プロジェクトに含まれています。CPLD のブロック図を以下に示します。CPLD には内部リセットと RC クロック・ジェネレータがあり、Verilog 設計で使用されます。

HI-6130 の PEX8311、CPLD の出カラムおよび、入力バッファで制御される LB タイミングでは、最大 16 個のウェイト・ステートをプログラムできます。14 (0xD) の値は LB EEPROM (U6) にプログラムされています。50MHz では、アクセス時間は  $1+14$  または  $1/50\text{MHz} \times 15 = 300\text{ns}$  です。これは、60ns のマージンを備えた非シーケンシャル・リード・サイクルでの HI-6130 の最悪の 240ns タイミング要件を満たします。CPLD は LB BLASTn 信号を使用して、CSn、RWn または、WRn 信号を HI-6130 および、内部ラッチと入力バッファにアサート解除するタイミングを設定します。LB からの ADSn 信号は、CPLD によってバス・サイクルを開始するために使用されます。より高速なアクセス時間が GPIO に使用されていたかもしれませんが、設計を単純化するために同じままにしました。LB 信号の説明とタイミングについては PLX8311 データシートを参照してください。

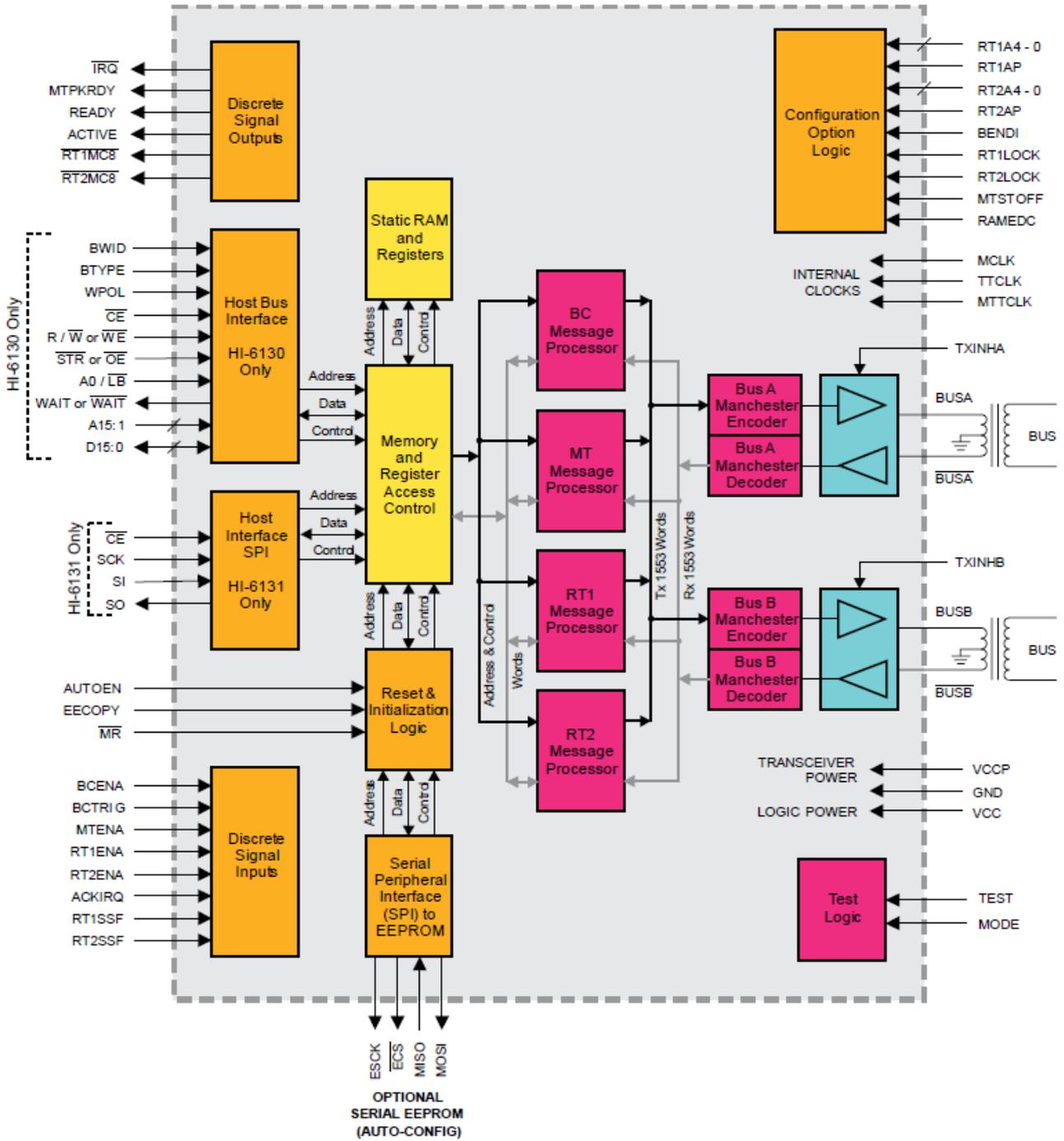
CPLD の大部分は、カスタム拡張のために十分な余裕を持って使用されていません。Lattice USB プログラミング・ケーブル PN : PN-USBN-2A は、CPLD を再プログラミングするのに必要ですが、Holt からは提供されていません。これは、エンドユーザーが Verilog コードを変更して CPLD を再プログラムしたい場合のみ必要です。Lattice Diamond CPLD 開発ソフトウェアの使用はこのドキュメントの範囲を超えています。多くのチュートリアルは Lattice Diamond ソフトウェアに組み込まれており、Web サイトからダウンロードできます。ボードがパワーアップすると、LED10 のみが ON になります。これは、CPLD がプログラムされているかどうかを判断する便利な方法です。

次ページに、CPLD のブロック図を示します。

## CPLD Functional Block Diagram [CPLD ファンクション・ブロック図]



HI-6130 Functional Block Diagram [HI-6130 ファンクション・ブロック図]



## Software [ソフトウェア]

2つの Holt デモ・プログラムが CD-ROM に収録されています。PCIe6130 テストには、ラッチされた I/O へのリード/ライト、HI-6130 メモリ空間へのリード/ライト、および BC 伝送用の HI-6130 の初期化を示すコマンド・メニューが含まれています。シンプルな BC デモでは、定期的に 3 種類のメッセージを送信します。このプログラムは、クイック・スタート・ガイドの章で説明されています。

HI-6130 API デモはより複雑です。このデモの主な目的は、Holt の API ライブラリを実証することです。このデモ・プログラムでは、Holt の API ライブラリを使用している HI-6130 を BC、RT、または MT 動作に初期化する方法を示します。両方のデモ・プロジェクトは、Microsoft Visual Studio 2012 を使用して構築されています。HI-6130 および GPIO をサポートする全ての API および、ローレベル・ドライバは、ヘッダ・ファイル HI6130.h を伴うモジュール HI6130.c に含まれています。

このプロジェクトを再構築するには、以下の 3 つの項目が必要です：

- CD-ROM に収録されている Holt のデモ・プロジェクト
- Microsoft Visual Studio 2012（提供されません）
- PLX SDK 7.20（avagotech.com ウェブサイト）

デモ・ソフトの 4 つのレイヤを以下に示します。

アプリケーション層 デモ・ソフトウェア
HOLT API
PLX API
PLX デバイス・ドライバ

### PLX SDK installation [PLX SDK インストール]

デモ・プログラムを変更するには、Holt プロジェクトを SDK サンプル・フォルダに追加できるように、まず PLX SDK をインストールする必要があります。このカードのカスタム・ソフトウェア・アプリケーションを開発するには、PLX の SDK が必要です。以下のリンクから SDK をダウンロードしてください。PLX はオンライン登録を記入し、ログインとパスワードを取得することにより、SDK をダウンロードするためのユーザー登録が必要です。Holt は PLX SDK パッケージを直接顧客に提供することはできません。この文書が作成された時点では、Windows 7 がサポートされており、Linux を含む将来の OS バージョンが計画されています。SDK をインストールした後は、新しいドライバを有効にするために、PC は数分かかる場合があるか、SDK のインストーラによってインストールされる新しいドライバのために再起動が必要です。

<http://www.plxtech.com/mydata/downloads>（ハイパーリンクが動作しない場合は、avagotech.com に直接行ってください）

SDK は、いくつかのサブ・フォルダを持つルート・ドライブに「Plx」フォルダをインストールします。

「PlxSdkUserManual」は、「C:¥Plx¥PlxSdk¥Documentation」フォルダにあります。これは、PEX8311 PCIe インターフェイス用のインストール、ドライバ、ユーティリティ、および PLX API に関する重要なドキュメントが含まれます。

デモ・プロジェクトを PLX SDK Samples フォルダにインストールするには、次の手順に従います。

PLX SDK をインストールします。

CD-ROM に圧縮された Holt デモ・プロジェクト「PCIe 6130 Test.zip」を配置し、このプロジェクト・フォルダを以下の Samples フォルダに解凍します。

C:¥Plx¥PlxSdk¥Samples¥\_PCIe 6130 Test

これらのディレクトリ内の他のファイルを参照する PLX ファイルがあるため、このディレクトリ構造を使用してください。Holt API プロジェクトの詳細については、このドキュメントの後半で説明します。

Microsoft Visual Studio を起動し「ファイル」メニューの「プロジェクトを開く」を使用して、PCIe6130Test プロジェクトを開くか、または、スタート・ページに表示されるプロジェクトを開くショートカットから選択します。または、PCIe6130 テスト・プロジェクト・フォルダ内の PCIe6130Test プロジェクト・ファイルをダブルクリックします。

1. ソース・ファイルを含むソリューション・エクスプローラが左側に表示されます。これが表示されない場合は、最上部の「表示」メニューからソリューション・エクスプローラを開きます。
2. Visual Studio デバッガでプログラムを実行するには、上部「Debug」構成が選択されていることを確認します。ソリューションのビルドを選択するか、F6 を押してください。出力ウィンドウには、エラーが発生していないはずです。
3. 「Local Windows Debugger」という緑色の矢印を押してプログラムを実行します。コンソール出力は、クイック・スタート・ガイドの 4 ページのように表示されます。
4. デモ・コードの実行可能なバージョンをビルドするには「デバッグ」ではなく「リリース」の設定を選択し、プロジェクトを再ビルドします。実行可能ファイルはプロジェクト・フォルダの「Release」サブ・フォルダに置かれます。ソース・レベルのデバッグ機能が動作するように、ソフトウェア開発に「Debug」設定を使用します。

注意：SDK がインストールされると、PEX8311 用のドライバがインストールされます。

開発作業や修正のためのソフトウェアとカードの動作を理解するためには、以下の文書を読むことが重要です。

- PLX SDK ユーザー・マニュアル（SDK インストール時にインストールされます）
- PLX PEX8311 RDK ハードウェア・リファレンス・マニュアルおよび、PLX PEX8311 データシート。最新バージョンは、avagotech.com の Web サイトから入手できます。これらは SDK には含まれていません。
- Holt HI6130 データシート。CD-ROM に収録されています。
- Holt ハイレベル API ソフトウェア・ユーザー・マニュアル。CD-ROM に収録されています。

Holt の他の有用なアプリケーション・ノート：

- AN-6130\_x.pdf
- AN-6130DG\_x.pdf
- AN-550.pdf（IC コンデンサ・カップリングおよびトランス PCB 配線用）

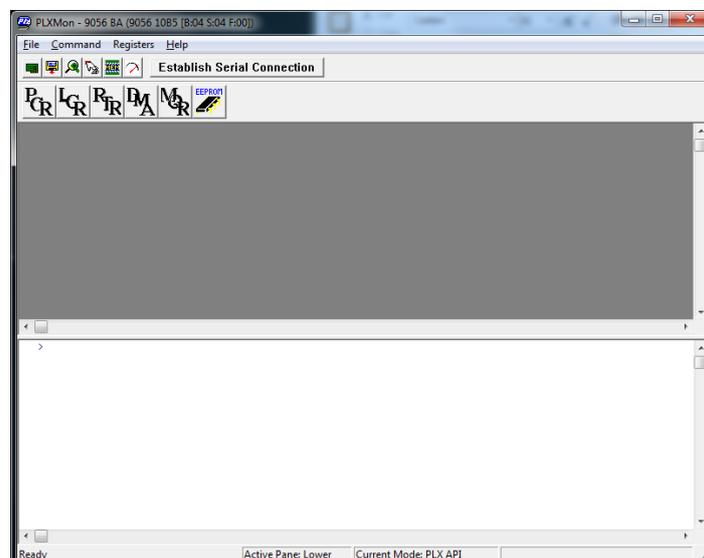
## PLX API's [PLX API]

PLX API はユニーク入力パラメータ (BAR) を使用して、リード/ライトする LB メモリ空間を指定します。BAR は、ベース・アドレス・レジスタの略です。BAR0 および、BAR1 は、アップストリーム PCIe 用に予約されています。BAR2 および、BAR3 は、LB 側に個別のメモリ空間を提供するために使用されます。BAR2 は空間 0、BAR3 は空間 1 を使用します。空間 0 および 1 の PEX8311 レジスタには、U6 EEPROM の内容から電源投入時に、必要な開始 BAR アドレスがロードされます。空間 0 は、HI-6130 64K メモリ空間専用で予約されており、開始アドレス 0x2000-0000 に設定されています。空間 1 は、アウトプット・ラッチへのリード/ライトと入力ステータス・バッファのリードを行う API 関数呼び出しで共有され、オンザフライで変更されます。ラッチは、HI-6130 へのさまざまな入力の状態を設定するために使用され、入力は HI-6130 ステータスおよび DIP スイッチの読み取りに使用されます。Holt のデモ・コードには、HI-1630 のメモリ空間をリード/ライトするためのいくつかの機能、ラッチ、および入力ステータス・バッファが含まれています。API 関数は HI6130.c にあります。これらの関数は、PLX API を使用して、「PlxPci\_PciBarSpaceRead(...)」または「PlxPci\_PciBarSpaceWrite(...)」のいずれかで LB にアクセスします。これらの API への入力パラメータの 1 つは「bOffsetAsLocalAddr」で、このパラメータは API が U32 オフセット・アドレスをどのように使用するかを制御します。「OffsetAsLocalAddr」が FALSE の場合、入力アドレスは空間 X からのオフセット・アドレスです。このパラメータが TRUE の場合、入力アドレスは完全なアドレスになります。PEX8311 および SDK API に関連する API、BAR スペースおよびその他の入力パラメータの説明および使用方法については、PLX API のユーザー・マニュアルを参照してください。

デバイス	BAR	空間	アドレス	注記
HI-6130	2	0	0x2000 - 0000	32K ワード幅
出力ラッチ	3	1	0x2400 - 0000	最初に使用した場所
入力	3	1	0x2800 - 0000	最初に使用した場所
出力ラッチ-2	3	1	0x2C00 - 0000	最初に使用した場所

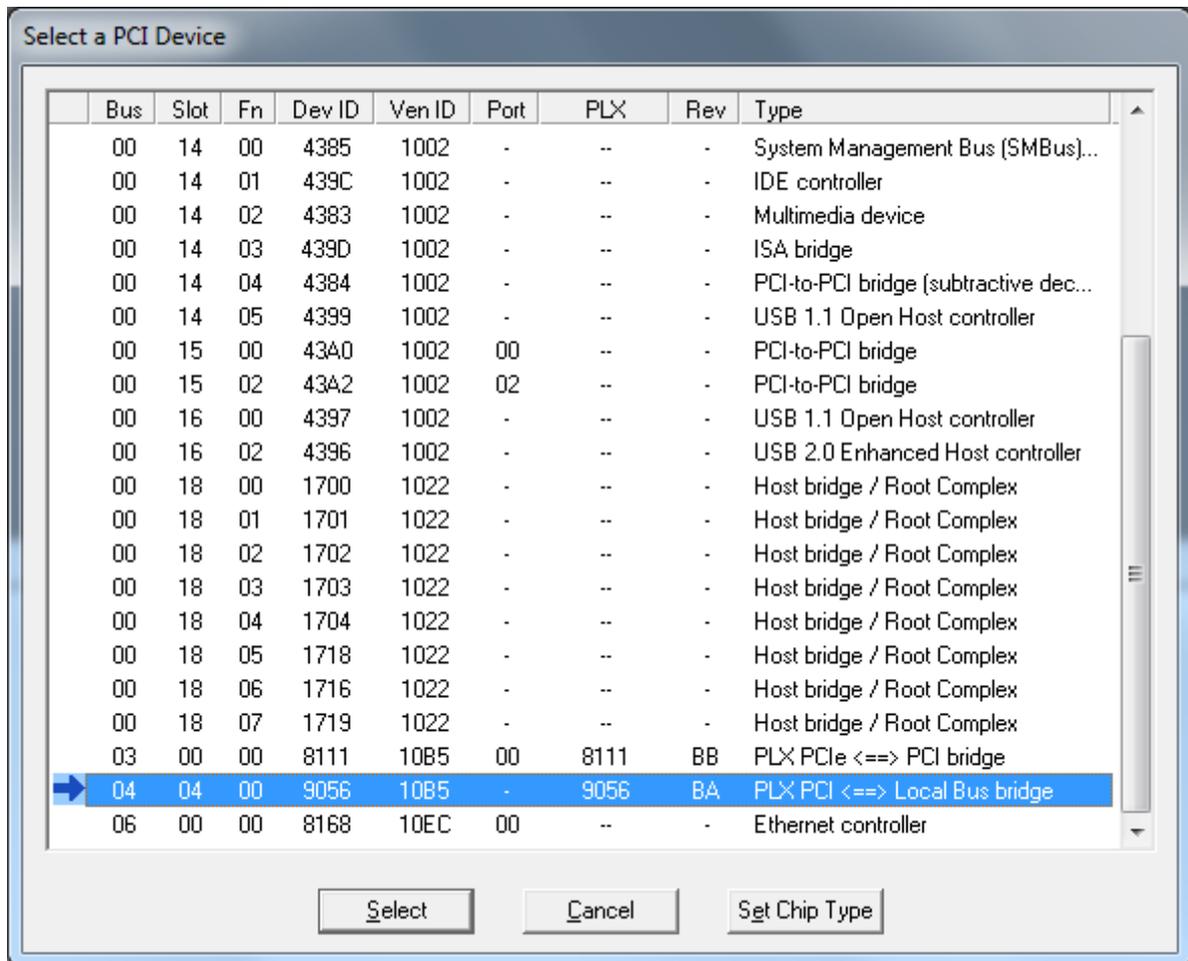
U6 EEPROM には、電源投入時に PEX8311 が LB 構成に使用する構成データが含まれています。LB は、16Bit のデータバス、14 のウェイト状態、空間 0 および空間 1 の開始アドレス/範囲、および TA/Ready 入力を無効にするように構成されています。PCI PLX サブ ID 3566 もこの EEPROM にプログラムされています。このサブ ID は、Holt デモ・ボード専用で PLX によって割り当てられます。

PlxMon ユーティリティは、Windows のスタート・メニューから起動するか、/Plx/.../Bin フォルダにあるアプリケーションをダブルクリックして起動します。PlxMon ユーティリティの開始画面を以下に示します。



特記事項：Avago は 2015 年に PLX を買収しましたが、SDK v7.2 はまだ「PLX」フォルダ名のままです。

コマンド・メニューから Holt PCIe カードを選択するか（または左側の緑色のアイコン・ボタンを押す）、Dev ID = 9056 および Ven ID = 10B5 デバイスを選択します。PEX8311 は内部に PEX8111 と PCI9056 で構成されています。



Holt カードを選択したら、LCR ボタン（左上）を選択してメイン・コントロール・レジスタを表示します。このメニューから新しい値を入力することにより、LB 構成への変更を動的に変更することができます。または、EEPROM ボタンを押して、以下に示す 9056 EEPROM の値を確認します。Ven ID が 10B5 であることに注意してください。これは PLX のベンダ ID であり、Holt に割り当てられたサブ ID 3566 で使用する必要があります。新しい最終製品には、PCI-SIG（PCI-SIG メンバになることによって得られる）または、PLX から取得されたサブ ID からの固有ベンダ ID が必要です。これらのパラメータと PLX API の詳細については、PEX8311 の PLX SDK ユーザー・ガイドとデータシートを参照してください。Holt カードの場合、これらの値を変更する必要はありません。

**9056 EEPROM Values**

**PCI Configuration Registers**

Device/Vendor ID (00)	905610B5	Class Code/Rev (04)	068000AA	Hot Swap Ctrl (54)	00004C06 -->
Subsystem ID (44)	356610B5	Max Lat/Int Pin & Line (08)	00000100	PM Capabilities (5C)	7A024801 -->
				PM Ctrl/Status (60)	00000000 -->

**Local Configuration Registers**

Space 0 Range (14)	FFFF0000 -->	VPD Boundary/Endian Desc (20)	20308500 -->
Space 0 Remap (18)	20000001	Direct Master -> PCI Range (30)	00000000
Expansion ROM Range (24)	00000000	Direct Master Memory Local Base Addr (34)	60000000
Expansion ROM Remap (28)	00000000	Direct Master I/O Local Base Addr (38)	50000000
Space 0/Exp ROM Descriptor (2C)	434300B5 -->	Direct Master -> PCI Memory Remap (3C)	00000000 -->
Space 1 Range (48)	FFFFFFFF0 -->	Direct Master -> PCI I/O PCI Configuration (40)	00000000 -->
Space 1 Remap (4C)	24000001	Mailbox 0 (0C)	00000000
Space 1 Descriptor (50)	000001B5 -->	Mailbox 1 (10)	00000000
Mode/DMA Arbitration (1C)	01200000 -->	PCI Arbiter Control (58)	00000000

Display Offsets from:  Serial EEPROM Base  
 PLX Chip Register Base

Buttons: Close, Refresh, Write, Load File, Save As...

PlxMon 画面の下部には、コマンド・メニューで LB メモリ空間のリード／ライトを可能にするコンソール・ウィンドウがあります。これらは、初期のハードウェア・チェックアウト中に LB 構成を確認するのに便利です。このウィンドウの Return キーに続いて、「？」を押すと、コマンドのヘルプ・ページが表示されます。これらのコマンドは、PLX API のユーザー・マニュアルにも記載されています。いくつかの例は、PLX PEX8311 RDK ハードウェア・リファレンス・マニュアルに記載されています。

---

## Holt HI-6130 PCIe API Demo using Holt's high-level API library. [Holt のハイレベル API ライブラリを h 使用した Holt HI-6130 PCIe API デモ]

HI-6130 PCIe API デモ・プログラムでは、Holt のハイレベル API ソフトウェア・ライブラリを示しています。このバージョンの API デモでは、API ライブラリ・ファイルは別のフォルダ「\_HI - 613x API LIB」に含まれており、解凍して、「\_HI - 6130 PCIe API Demo」プロジェクト・フォルダが存在するのと同じディレクトリ・フォルダにインストールする必要があります。両方のプロジェクト・フォルダを PLX SDK サンプル・フォルダにコピーする前に、zip 形式のプロジェクト・フォルダを CD-ROM からデスクトップにコピーして解凍します：

```
C:\%Plx%\PlxSdk\Samples\_HI-613x API LIB
```

```
C:\%Plx%\PlxSdk\Samples\_HI-6130 PCIe API Demo
```

Holt API ライブラリ・ファイル「apiLib.lib」はあらかじめ作成されており、\_HI - 613x API LIB/lib フォルダ内にあります。特定の C ソース・ファイルがこのバージョンに含まれていないため、Visual Studio を使用してこのプロジェクトを再構築することはできません。Holt の販売団利点にソフトウェア・ライセンス契約 (SLA) を問い合わせしてください。

このデモは、BC、RT1、RT2、SMT および IMT のデモを行うためにコンソール上のコマンド・プロンプト・メニューで構成されています。

Dslave と LocalToPciInt で使用されている PLX API の例の多くは、Holt プロジェクトで使用されていました。この 2 つのプロジェクトを最初に PLX API ライブラリに慣れ親しむときには、このアイデアを参考にするをお勧めします。これらの PLX プロジェクトは、LB メモリ空間が異なるように定義されているため、Holt PCIe カード上で実行されません。

Visual Studio プロジェクトの「Release」サブ・フォルダに含まれている実行可能アプリケーション「PCleHolt」をダブルクリックするか、Visual Studio を起動してプロジェクトを開いてプログラムをデバッグ・モードで実行して、プログラムを起動します。デバッグ・モードでプロジェクトを実行するには、デバッグ設定が最上部で選択されていることを確認します。場合によっては、ビルド・プルダウン・メニューから「クリーン PCIe6130Holt」を選択して、プロジェクトをクリーンにする必要が生じることがあります。プロジェクトを再構築した後、無視することができるいくつかの警告が出力ウィンドウに表示されることがありますが、デバッガがプロジェクトを実行できないような重大なエラーは発生しません。

1~x までの数字のセットを示すメニューが表示されます。Holt カードは通常、最初の項目として表示され、#1 with "9056 10b5" [b:xx s:xx f:xx]と表示されます。「1」と入力して Enter を押します。

次のコンソール・メイン・メニューが表示されます。以下に示すプログラムの改訂は異なる場合があります。

```

C:\Plix\PlixSdk\Samples\HI-6130_2 API Demo\Debug\PCIE6130Holt.exe
*****
Holt Integrated Circuits Hi-6130-PCIe Project Rev-2.0
Holt API Ver: 02-1-2
*****

Press 1 to Display 6130 memory 0-0xFFFF.
Press D to Display all 6130 system registers.
Press A to run Async Demo.
Press H to send high priority BC message.
Press L to send low priority BC message.
Press T to display RT Traffic Toggle.
Press K to Enable RIMT.
Press F to set GP2 flag for Async demo.
Press M to display this Menu.
Press N to run BC Major Minor Frame demo.
Press S to run SMT demo.
Press I to run IMT demo.
Press B to run RI demo.
Press C to run RT2 demo.
Press O to Output a value to a specified GPIO port.
Press W to Write to a specified memory location.
Press R to Pulse Master Reset line.
Press Q to quit demo.
Press X to quit BC.

=====
Press M to Output this Menu header.

Waiting for Ready to go HIGH
Ready is HIGH
Regs: 0x0000 - 0x0007: 00C0 8000 0000 0000 0000 0000 0000 0000
Regs: 0x0008 - 0x000F: 0000 0000 0180 0000 0000 0000 0000 6018
Regs: 0x0010 - 0x0017: 0000 0000 0408 6018 0000 0000 0408 0080
Regs: 0x0018 - 0x001F: 1C00 0400 0000 0000 0000 0000 0000 0000
Regs: 0x0020 - 0x0027: 0080 2000 0600 0000 0000 0000 0000 0000
Regs: 0x0028 - 0x002F: 0000 0000 0000 0000 0000 0000 0000 00B0
Regs: 0x0030 - 0x0037: 0000 0000 0000 0000 0000 0000 0000 0000
Regs: 0x0038 - 0x003F: 00C0 0022 094A 0000 0000 0000 0000 0000
Regs: 0x0040 - 0x0047: 0000 0000 0000 1C50 0000 0000 0000 0000
Regs: 0x0048 - 0x004F: 0000 1C57 0000 1C5A 0000 0000 0000 0000

HI6130 Demo selection --> _

```

D を押すと、HI-6130 システム・レジスタにラベルが表示され、続いて先頭一末尾でフォーマットされた同じレジスタ値の後に 8 つのレジスタ値が続きます。

```

HI6130 Demo selection --> d
Regs: 0x0000 - 0x0007: 10C0 8100 0000 0000 0000 0000 0000 0000
Regs: 0x0008 - 0x000F: 0000 0000 0180 0000 0000 0000 0000 6018
Regs: 0x0010 - 0x0017: 0000 0000 0408 6018 0000 0000 0408 0080
Regs: 0x0018 - 0x001F: 1C00 0400 0000 0000 0000 0000 0000 0000
Regs: 0x0020 - 0x0027: 0080 2000 0600 0000 0000 0000 0000 0000
Regs: 0x0028 - 0x002F: 0000 0000 0000 0000 0000 0000 0000 00B0
Regs: 0x0030 - 0x0037: 0000 0000 1A00 3E2E 3E10 FD7E 0000 C101
Regs: 0x0038 - 0x003F: 00CA 0022 FD97 0801 0000 0000 0000 0000
Regs: 0x0040 - 0x0047: 0000 0000 0000 0935 0000 0000 0000 0000
Regs: 0x0048 - 0x004F: 0000 093C 0000 093E 0000 0000 0000 3070

```

レジスタ 0 (0000) は、マスタ・コントロール・レジスタで、レジスタ 1 はマスタ・ステータスおよびリセット・レジスタです。ワードアドレス 0x0001 の MSB が High (8) であることに注意してください。これは、HI-6130 がホスト・アクセスの準備ができていることを示す READY ビットです。TeraTerm ウィンドウを十分に広げて、全部の 16 個のレジスタを見渡せる最適な表示にします。

## General structure of demo functions [デモ機能の一般的な機能]

Holt API デモ・プログラムは、モジュール「demos.c」に含まれています。API ライブラリは、実行可能オブジェクト・コードとしてライブラリ・ファイル「apiLib.lib」に含まれています。「Demos.c」には、BC、RT、および MT ターミナルを初期化するためにコンソール・メニューから実行されるデモをサポートするデモ初期化 API 関数呼び出しが含まれています。キー押下は「console.c」で検出され、「main.c」メインループから呼び出され、「demos.c」のデモ機能を実行します。

コマンド「A」と「N」は、BC コマンドを送信し、オシロスコープで表示することができます。オプションで、「K」と「T」コマンド・シーケンスを使用して、デコードされたメッセージ・トラフィック・データをコンソールに表示することもできます。これらのデモでは、Holt API ライブラリを使用して、BC 非同期メッセージ、メジャー／マイナー・フレーム、低優先度、高優先度の非同期（スケジュールされていない）メッセージを生成する方法を示します。外部の MIL-STD-1553 テスト装置でこれらのメッセージを表示するか、オシロスコープで表示してください。

この演習では、内部 BC を使用してメッセージを送信するので、メッセージ・トラフィック・データがコンソールに表示されます。内部 BC、RT、および SMT は同じバスピンを共有するため、RT および SMT モニタ・ターミナルは BC メッセージを受信します。外部 BC がすでに 9 ピン／MIL-STD-1553 トライアキシャル・ケーブルにバスカプラを介して接続されている場合、それを接続したままにしておき、オンチップ BC 送信と競合する外部送信をすべて無効にしてください。外部 BC が接続されていない場合は、外部 75Ω 1/2W 抵抗を BUSA および nBUSA テスト・ポイントに接続してバスを終端します。これらは 9 ピン D-sub コネクタとトランスに隣接する赤と黒のテスト・ポイントです。

### コンソール上での RT トラフィックによる内部 BC デモ :

1. 両方の RT を有効にするには、コマンド「B」と「C」を押します
2. コマンド「K」を押して、RTMT デモを有効にします
3. コマンド「T」（または、スペース）を押して、コンソールに RT トラフィックを表示します。コマンド「T」は交互にオンとオフを切り換え、コンソールに表示される RT トラフィックを有効または無効にします。コマンド「T」の使用は、コマンド「K」の事前実行に依存します。

```

HI6130 Demo selection --> b
HI6130 Demo selection --> c
HI6130 Demo selection --> k
RTMT Demo
RTMT Demo
HI6130 Demo selection --> t
Traffic Enabled
HI6130 Demo selection --> _

```

4. コマンド「A」を押して、BC 送信メッセージを開始します

メッセージ画面が急速に表示されます。「T」を押すとコンソール出力が停止します。コンソールは停止し、以下の画面のように表示されます。もう一度「T」を押すと、コンソール出力が再開されます。コマンド「T」を使用しても BC の送信は停止せず、RT または MT メッセージの受信も継続されます；コンソール出力のみを停止します。

```

C:\Plix\PlixSdk\Samples\HI-6130_2 API Demo\Debug\PCIE6130Holt.exe
MSG #0273.  TIME = 00015740us    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
            DATA 0909 1010
            STA2 1800

MSG #0274.  TIME = 00114640us    BUS A    TYPE0: BC to RT
            CMD1 1822 --> 03-R-01-02
            DATA 0005 0002
            STA1 1800

MSG #0275.  TIME = 00114942us    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
            DATA 0909 1010
            STA2 1800

MSG #0276.  TIME = 00115038us    BUS A    TYPE0: BC to RT
            CMD1 1822 --> 03-R-01-02
            DATA 0005 0002
            STA1 1800

t
Traffic Disabled
HI6130 Demo selection --> _

```

次のスクリーン・ショット（次ページ）は、BC 非同期デモの最中にキャプチャされたメッセージの一部を表示しています。次のページに示す送信は、Ballard USB 1553 モニタを使用してキャプチャされました。

5. プログラム起動から、メッセージ・トラフィックを表示するために「K」および「T」シーケンスが有効になる前に BC が開始されると、最初のメッセージにエラーが含まれる可能性があります。これは正常です；これは、処理中メッセージの途中で RT と MT が有効になっているために発生しています。

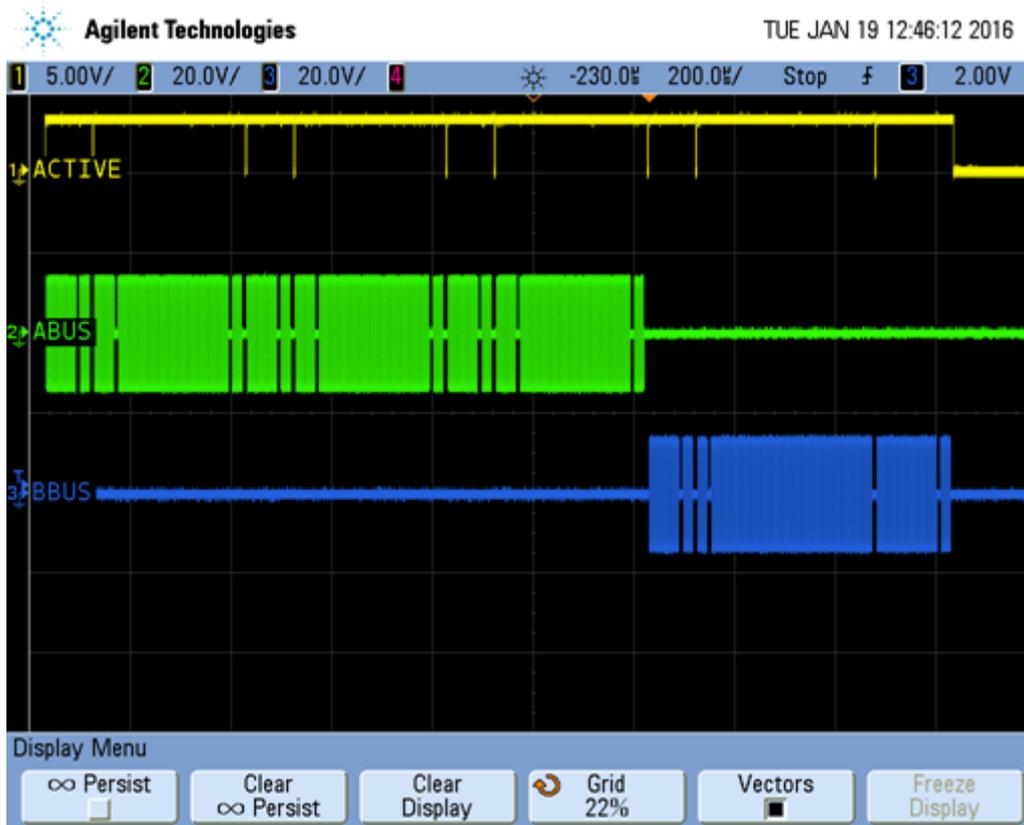
Rec #	Time	Msg Name	Message	Bus	Error	Data 4x8	Chan	Swd Bits	Warning
0	T=000:00:00:15.346929 dT=000:00:00:00.000000	RT:RT03	Cwd1=1822 (03,R,01,02) <DATA WORDS> Swd1=1800	A		01: 0005 0002	0		
1	T=000:00:00:15.346942 dT=000:00:00:00.000013	RT:RT03	Cwd1=182A (03,R,01,10) Cwd2=0C2A (01,T,01,10) Swd2=0800 <DATA WORDS> Swd1=1800	A		01: BBBB 0202 1414 0404 05: 0505 0606 0707 0808 09: 0909 1010	0		
2	T=000:00:00:15.3461056 dT=000:00:00:00.000113	RT:RT03	Cwd1=1822 (03,R,01,02) <DATA WORDS> Swd1=1800	A		01: 0005 0002	0		
3	T=000:00:00:15.3461129 dT=000:00:00:00.000073	RT:RT03	Cwd1=182A (03,R,01,10) Cwd2=0C2A (01,T,01,10) Swd2=0800 <DATA WORDS> Swd1=1800	A		01: BBBB 0202 1414 0404 05: 0505 0606 0707 0808 09: 0909 1010	0		
4	T=000:00:00:15.3461243 dT=000:00:00:00.000113	RT:RT03	Cwd1=1822 (03,R,01,02) <DATA WORDS> Swd1=1800	A		01: 0005 0002	0		
5	T=000:00:00:15.3461256 dT=000:00:00:00.000012	RT:RT03	Cwd1=182A (03,R,01,10) Cwd2=0C2A (01,T,01,10) Swd2=0800 <DATA WORDS> Swd1=1800	A		01: BBBB 0202 1414 0404 05: 0505 0606 0707 0808 09: 0909 1010	0		
6	T=000:00:00:15.3461374 dT=000:00:00:00.000118	RT:RT03	Cwd1=1822 (03,R,01,02) <DATA WORDS> Swd1=1800	A		01: 0005 0002	0		
7	T=000:00:00:15.3461447 dT=000:00:00:00.000073	RT:RT03	Cwd1=182A (03,R,01,10) Cwd2=0C2A (01,T,01,10) Swd2=0800 <DATA WORDS> Swd1=1800	A		01: BBBB 0202 1414 0404 05: 0505 0606 0707 0808 09: 0909 1010	0		
8	T=000:00:00:15.3461501 dT=000:00:00:00.000053	RT:RT03	Cwd1=1822 (03,R,01,02) <DATA WORDS> Swd1=1800	A		01: 0005 0002	0		
9	T=000:00:00:15.3461574 dT=000:00:00:00.000073	RT:RT03	Cwd1=182A (03,R,01,10) Cwd2=0C2A (01,T,01,10) Swd2=0800 <DATA WORDS> Swd1=1800	A		01: BBBB 0202 1414 0404 05: 0505 0606 0707 0808 09: 0909 1010	0		
10	T=000:00:00:15.347094 dT=-000:00:00:00.000479	RT:RT03	Cwd1=1822 (03,R,01,02) <DATA WORDS> Swd1=1800	A		01: 0005 0002	0		
11	T=000:00:00:15.347101 dT=000:00:00:00.000007	RT:RT03	Cwd1=182A (03,R,01,10) Cwd2=0C2A (01,T,01,10) Swd2=0800 <DATA WORDS> Swd1=1800	A		01: BBBB 0202 1414 0404 05: 0505 0606 0707 0808 09: 0909 1010	0		

## 6. BC 低優先度非同期メッセージ挿入

コマンド「L」は、フレームの最後に低優先度メッセージを BC 送信に挿入します。まず、「B」と「C」を押して両方の RT を有効にし、「A」を押して BC の送信を有効にします。バス B で 3 つの追加メッセージを送信するには「L」を押します。バス B はスコープを見やすくするために使用しています。RT が有効になっていない場合、バス B に再試行メッセージが表示され、挿入された 3 つのメッセージを確認することが難しくなります。電源投入またはリセット後に 1 回だけ機能します。

Rec #	Time	Message	Bus	Error	Data 4x8	Chan	Swd Bits	Warning
0	T=000:00:00:0... dT=000:00:00:0...	Cwd1=0822 (01,R,01,02) <DATA WORDS> Swd1=0800	B		01: DEAD BEEF	1		
1	T=000:00:00:0... dT=000:00:00:0...	Cwd1=0C2F (01,T,01,15) <DATA WORDS> Swd1=0800	B		01: BBBB 0202 1414 ... 05: 0505 0606 0707 ... 09: 0909 1010 1111 ... 13: 1313 1414 1515	1		
2	T=000:00:00:0... dT=000:00:00:0...	Cwd1=0825 (01,R,01,05) <DATA WORDS> Swd1=0800	B		01: CAFE CODE 0303 ... 05: 0505	1		

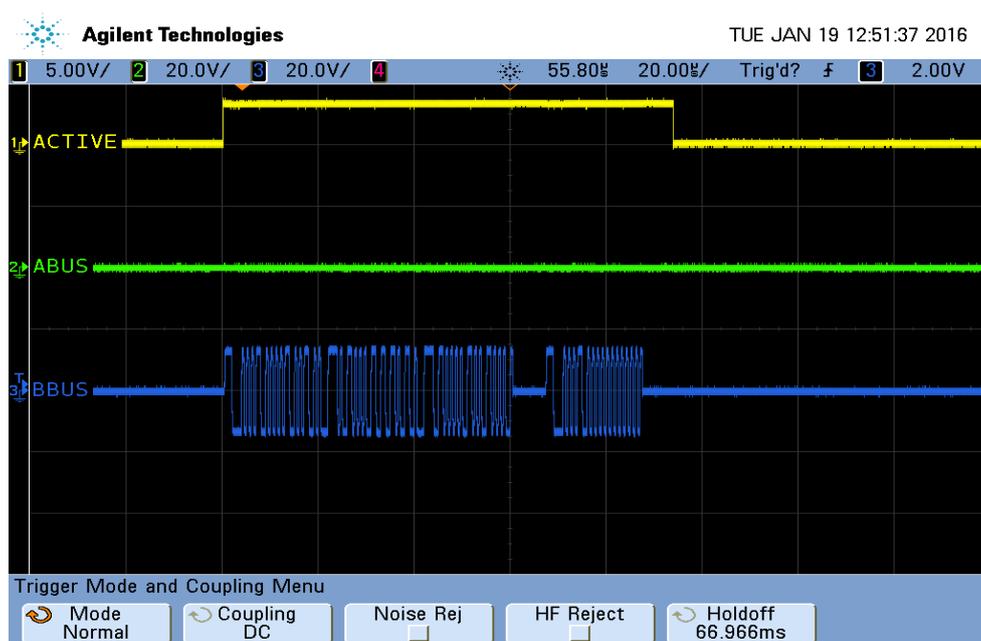
これは、低優先度メッセージが青色で送信された瞬間のバス A およびバス B の信号を示します。



#### 7. BC 高優先度非同期メッセージの挿入

以前の BC 低優先度メッセージの例と同じ手順を実行しますが、今回は、現在のスケジュール済みメッセージの完了後に単一の高優先度を挿入するには、「H」を押します。このコマンドは繰り返し可能です。

T=000:00:0...	Cwd1=0822 (01,R,01,02)				
dT=000:00:0...	<DATA WORDS>	B	01: DEAD BEEF	1	
	Swd1=0800				



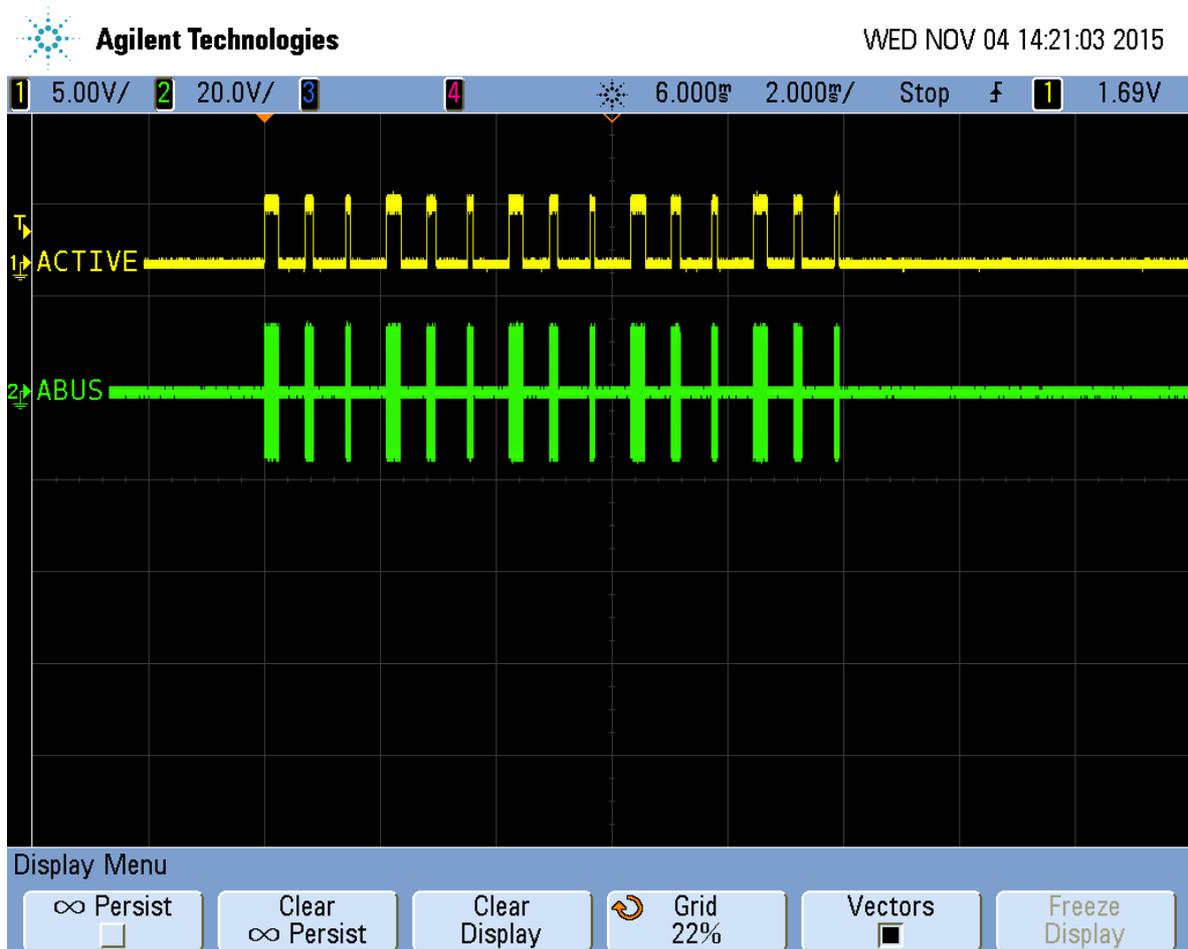
8. 「N」コマンドは、RT アドレス 3 に 15 個のコマンドを送信します。「B」を押すと RT が有効になります。

RT の命名は混乱する可能性があり、さらに説明が必要です：HI-6130 には 2 つの RT ターミナルがあります。2 つの RT ターミナルにはそれぞれ RT アドレスを設定できるようにボード上に独立した RT アドレス DIP スイッチがあります。

最初の RT (RT または RT1 と呼ばれる) は、RT アドレス 3 に設定

第 2 の RT (RT2) は RT アドレス 1 に設定

「N」を押すと、以下のようにバスに表示される BC 送信 (15 個のメッセージが送信されます) が実行されます。オプションでコンソール上のメッセージ・トラフィックを表示するには、まだ有効になっていない場合は「T」を押して RT メッセージ・トラフィックを有効にします。



コマンド「N」(15メッセージ)トラフィックを、Ballard USB 1553 モニタを使用してキャプチャした画面です。コマンド「T」を使用して、コンソールで同じメッセージを表示します。

Rec #	Time	Message	Bus	Error	Data 4x8	Chan	Swd Bits	Warning
0	T=000:00:00:00.1583950 dT=000:00:00:00.000000	Cwd1=1C2A (03,T,01,10) <DATA WORDS> Swd1=1800	A		01: 1000 1001 1002 1003 05: 1004 1005 1006 1007 09: 1008 1009	0		
1	T=000:00:00:00.15831208 dT=000:00:00:00.000258	Cwd1=1825 (03,R,01,05) <DATA WORDS> Swd1=1800	A		01: AAAA 0202 0303 0404 05: 0505	0		
2	T=000:00:00:00.15831466 dT=000:00:00:00.000257	Cwd1=1822 (03,R,01,02) <DATA WORDS> Swd1=1800	A		01: BBBB 0202	0		
3	T=000:00:00:00.1584128 dT=-000:00:00:00.000337	Cwd1=1C2A (03,T,01,10) <DATA WORDS> Swd1=1800	A		01: 1000 1001 1002 1003 05: 1004 1005 1006 1007 09: 1008 1009	0		
4	T=000:00:00:00.1584446 dT=000:00:00:00.000318	Cwd1=1825 (03,R,01,05) <DATA WORDS> Swd1=1800	A		01: AAAA 0202 0303 0404 05: 0505	0		
5	T=000:00:00:00.1584704 dT=000:00:00:00.000258	Cwd1=1822 (03,R,01,02) <DATA WORDS> Swd1=1800	A		01: BBBB 0202	0		
6	T=000:00:00:00.1584967 dT=000:00:00:00.000262	Cwd1=1C2A (03,T,01,10) <DATA WORDS> Swd1=1800	A		01: 1000 1001 1002 1003 05: 1004 1005 1006 1007 09: 1008 1009	0		
7	T=000:00:00:00.15841224 dT=000:00:00:00.000256	Cwd1=1825 (03,R,01,05) <DATA WORDS> Swd1=1800	A		01: AAAA 0202 0303 0404 05: 0505	0		
8	T=000:00:00:00.15841542 dT=000:00:00:00.000318	Cwd1=1822 (03,R,01,02) <DATA WORDS> Swd1=1800	A		01: BBBB 0202	0		
9	T=000:00:00:00.1585205 dT=-000:00:00:00.000337	Cwd1=1C2A (03,T,01,10) <DATA WORDS> Swd1=1800	A		01: 1000 1001 1002 1003 05: 1004 1005 1006 1007 09: 1008 1009	0		
10	T=000:00:00:00.1585463 dT=000:00:00:00.000258	Cwd1=1825 (03,R,01,05) <DATA WORDS> Swd1=1800	A		01: AAAA 0202 0303 0404 05: 0505	0		
11	T=000:00:00:00.1585721 dT=000:00:00:00.000257	Cwd1=1822 (03,R,01,02) <DATA WORDS> Swd1=1800	A		01: BBBB 0202	0		
12	T=000:00:00:00.15851043 dT=000:00:00:00.000321	Cwd1=1C2A (03,T,01,10) <DATA WORDS> Swd1=1800	A		01: 1000 1001 1002 1003 05: 1004 1005 1006 1007 09: 1008 1009	0		
13	T=000:00:00:00.15851301 dT=000:00:00:00.000258	Cwd1=1825 (03,R,01,05) <DATA WORDS> Swd1=1800	A		01: AAAA 0202 0303 0404 05: 0505	0		
14	T=000:00:00:00.15851565 dT=000:00:00:00.000264	Cwd1=1822 (03,R,01,02) <DATA WORDS> Swd1=1800	A		01: BBBB 0202	0		

有効にされていない RT に BC メッセージが送信されると「RT no response」(NORES) エラーが表示されます。

```
MSG #0170.    TIME = 00086918us   BUS A   TYPE2: RT to RT
              CMD1 182A --> 03-R-01-10
              CMD2 0C2A --> 01-T-01-10
```

**ERROR: NORES**

## コマンド「S」と「I」

これらのデモ・コマンドは、HI-6130 の SMT または IMT モニタ機能を初期化してメイン・メニューに戻りますが、コンソールには何も表示されません。SMT も初期化され、「T」 RT トラフィック・コマンドで使用されます。

## Using an external BC (such as Ballard tester) to transmit messages to the demo board.

[デモ・ボードにメッセージを送信するために外部 BC (Ballard テスタなど) を使用する]

適切に終端された外部バスを使用して接続された BC を使用して RT または SMT にコマンドを送信する場合は、外部の 75Ω 抵抗 (存在する場合) を取り外し、MIL-STD-1553 ケーブルおよび、カプラを使用して BC テスト装置をデモ・ボードの丸型トライアキシャル・バスジャックに接続します。バスカプラをすぐに使用できない場合は、BC テスタをデモ・ボードのトライアキシャル・ジャックに直接接続しますが、この場合は 75Ω の終端抵抗を使用してください。

プログラムを再起動し、「B」または「C」を押して両方の RT を有効にし、「K」と「T」を押してコンソールの RT トラフィックをアクティブにします。以下に示すメッセージに類似した SA=1 および 32 のデータワードを有する BC->RT へのメッセージを構成します。コンソールは、トランザクションの後、BC によって送信されたメッセージを表示する必要があります。

```

C:\Plix\PlixSdk\Samples\HI-6130_2 API Demo\Debug\PCIE6130Holt.exe
HI6130 Demo selection --> b
HI6130 Demo selection --> c
HI6130 Demo selection --> k
RTMT Demo
RTMT Demo
HI6130 Demo selection --> t
Traffic Enabled
HI6130 Demo selection -->
MSG #0000.  TIME = 00040626us  BUS A  TYPE0: BC to RT
            CMD1 1820 --> 03-R-01-00
            DATA 0000 0001 0002 0003 0004 0005 0006 0007
                  0008 0009 000A 000B 000C 000D 000E 000F
                  0010 0011 0012 0013 0014 0015 0016 0017
                  0018 0019 001A 001B 001C 001D 001E 001F
            STA1 1800

```

MIL-STD-1553 の典型的な高速レートで繰り返しメッセージを送信する場合、コンソールに表示される RT トラフィックは、Windows PC OS の制限によりペースを維持できない可能性があります。メッセージ内容とリピート周期によっては、コンソールにメッセージが表示されないことがあります。すべてのメッセージは適切に処理され、有効な RT と MT によってキャプチャされますが、一部のメッセージはコンソールに表示されません。Holt API 関数呼び出しについては、ADK CD-ROM に収録されている Holt API ユーザーズガイド HI-6130-API\_xx.pdf で詳しく説明されています。

**Appendix- A CPLD Verilog Source [付録 A CPLD Verilog ソース]**

**A current copy of this source file is included in the Verilog source file on the CD-ROM. [このソース・ファイルの現在のコピーは、CD-ROM の Verilog ソース・ファイルに含まれています]**

```
// Holt PCI_6130 interface
module count_osc (rstn, osc_clk, LED, clk,           // clk and reset pins
                 testpoint,                       // test output
                 add, decoderOutput,              // address decoder
                 lclk,blast,blast_q,ads,lwr,RDn,WRn, // Read Write strobes
                 LEDRD, LEDWR,                   // LED flashers
                 Latch,                           // 16 latched outputs
                 DataBus,                         // Data bus In/Out (16)
                 InputBuffer,                    // Inputs (16)
                 nLINTi,                          // Interrupt output pin
                 Latch2,
                 TP13
                 );

// Misc. Signals
input rstn ;
output osc_clk ;
output [7:0]LED ;
output clk ;
output testpoint ;           // Test OE
output TP13;                 // Test ADSn

// Inputs
input lclk;                  // LCLK 50 MHZ input
input blast;                // blast input
input [5:0]add;              // inputs LA31-LA26 (6) for decoder
input [15:0]InputBuffer;    // 16 status input pins
input ads;                  // ADS input
input lwr;                  // LW/R input
//input RT1MC8n, RT2MC8n, IRQn; // Interrupt inputs

// InOuts
inout [15:0]DataBus;        // 16 In/out Data Bus pins

// Outputs
output blast_q;             // blast output
output [4:0]decoderOutput;  // address decoder outputs
output reg RDn;             // /RD output strobe
output reg WRn;             // /WR output strobe
output reg LEDRD, LEDWR;    // LED flashers for 6130 indication
output reg [15:0]Latch;     // 16 latched outputs
output reg [15:0]Latch2;    // 2nd set of latches
output nLINTi;              // Interrupt output to PLX

// Wires

// Constants
`define H6130                8'b11111110 // H6130 address decode address
`define LatchAddress         8'b11111101 // Latches output decode address
`define InputsAddress        8'b11111011 // Inputs address decode address
`define Latch2Address        8'b11110111 // Inputs address decode address

// Registers
reg [26:0]c_delay ;
reg [7:0] add_L;
reg [0:0] blastq;
reg [3:0] counter;
reg [3:0] counter2;
reg oe;                     // output enable for buffer reads
reg [15:0] mux;             // internal bus for muxing readback bus'
```

```

// Get IC clk and reset
GSR GSR_INST (.GSR(rstn));           // Reset occurs when argument is active low.
OSCC OSCC_1 (.OSC(osc_clk));

// Logic
assign LED[3:0] = add_L;              // copy decoder outputs to LEDs for test
assign LED[4] = blast_q;             // copy of blast output
assign LED[5] = WRn;                // copy WRn to output
assign LED[6] = RDn;                // copy WRn to output
assign LED[7] = lclk;               // copy clock to output
// End of test

assign decoderOutput = add_L;
assign TP13 = ads;

/* 3-8 Address Decoders */
always@(*)
    case (add)
        6'b001000 : add_L = `H6130;           // led1 CS20 - 6130
        6'b001001 : add_L = `LatchAddress;    // led2 CS24 - Output latches
        6'b001010 : add_L = `InputsAddress;   // led3 CS28 - Inputs
        6'b001011 : add_L = `Latch2Address;   // led4 CS2C - Latch2 output latches
        6'b001100 : add_L = 8'b11101111;     // led5 Spare CS
        default : add_L = 8'b11111111;       // default all OFF
    endcase

/* 16 bit 3 to 1 multiplexer */
always@(*)
    begin
        case(add_L)
            `LatchAddress : mux = Latch;       // First latches routed to mux
            `InputsAddress : mux = InputBuffer; // Status inputs (6130) routed to mux
            `Latch2Address : mux = Latch2;     // Second set of latches routed to mux
            default : mux = 16'hFFFF;         // default
        endcase
    end

/* RD, RW, 6130CS Strobe generation */
// generate delayed blast signal used to clear RD, WR.
always @(posedge lclk) begin
    if (blast)
        blastq <= 1'b1;                     // set high during reset
    else
        blastq <= 1'b0;                     // set high during reset
    end
assign blast_q = blastq;                   // leave as output pin for possible debugging later

// generate bracketed RDn stobe
always @(posedge lclk or negedge blastq)
    begin
        if(~blastq)
            RDn <= 1'b1;                     // set rd high
        else begin
            if(~lwr && ~ads)
                RDn <= 1'b0;                 // set rd low
            end
        end

// generate bracketed WRn stobe
always @(posedge lclk or negedge blastq)
    begin
        if (~blastq)
            WRn <= 1'b1;                     // set wr high if blast_q =0
        else begin

```

```

        if(lwr && ~ads)
            WRn <= 1'b0; // set wr low
        end
    end

// ----- 6130 RD & WR LED flashers -----
always @(posedge c_delay[18] or negedge RDn)
    begin
        if (~RDn)
            begin
                if(add_L == `H6130) // only if 6130
                    begin
                        LEDRD <= 1'b0; // Turn on led
                        counter = 6'b0; // reset timer
                    end
                end
            else
                if (counter==3)
                    LEDRD <= 1'b1; // turn off led
                else counter = counter +1;
            end
    end

always @(posedge c_delay[18] or negedge WRn)
    begin
        if (~WRn)
            begin
                if(add_L == `H6130) // only if 6130
                    begin
                        LEDWR <= 1'b0; // Turn on led
                        counter2 = 6'b0; // reset timer
                    end
                end
            else
                if (counter2==3)
                    LEDWR <= 1'b1; // turn off led
                else counter2 = counter2 +1;
            end
    end

/* ----- Latched 16 GPIO's -----*/
// For Latch outputs
always @(posedge WRn or negedge rstn)
    begin
        if (~rstn)
            Latch <= 16'b0001000000110000; // Defaults: LED1 On (low), TXINHA, TXINHB off
        else if (add_L == `LatchAddress) // latches decode
            Latch <= DataBus; //dbus;
        end
    // For Latch2 outputs
    always @(posedge WRn or negedge rstn)
        begin
            if (~rstn)
                Latch2 <= 16'h002A; // Defaults: all low (00A5 for testing only)
            else if (add_L == `Latch2Address) // latches decode
                Latch2 <= DataBus; //dbus;
            end
        /* ----- Read Input Buffers, or Latches 16 GPIO's -----*/
        always @(*)
            begin
                if(~RDn & (add_L==`InputsAddress))
                    oe=1'b1; // turn on buffer
                else if(~RDn & (add_L==`LatchAddress))
                    oe=1'b1; // turn on buffer
                else if(~RDn & (add_L==`Latch2Address))
                    oe=1'b1; // turn on buffer
            end
    end

```

```
        else
            oe=1'b0; // turn off buffer
        end

assign testpoint = oe; // leave as pin for debugging, later change to reg
//assign DataBus = oe ? InputBuffer : 16'hZ; // Read the 16 inputs
assign DataBus = oe ? mux : 16'hZ; // Read the 16 inputs

/* Misc Logic, Interrupt, MR */
assign nLINTi = InputBuffer[4] & InputBuffer[5] & InputBuffer[6]; // Interrupt pins

// End of Misc.

// The c_delay counter is used to slow down the internal oscillator (OSC) output
// to a rate of approximately 0.5 Hz
always @(posedge osc_clk or negedge rstn)
    begin
        if (~rstn)
            c_delay <= 32'h0000 ;
        else
            c_delay <= c_delay + 1;
    end

assign clk = c_delay[18] ;

endmodule
```

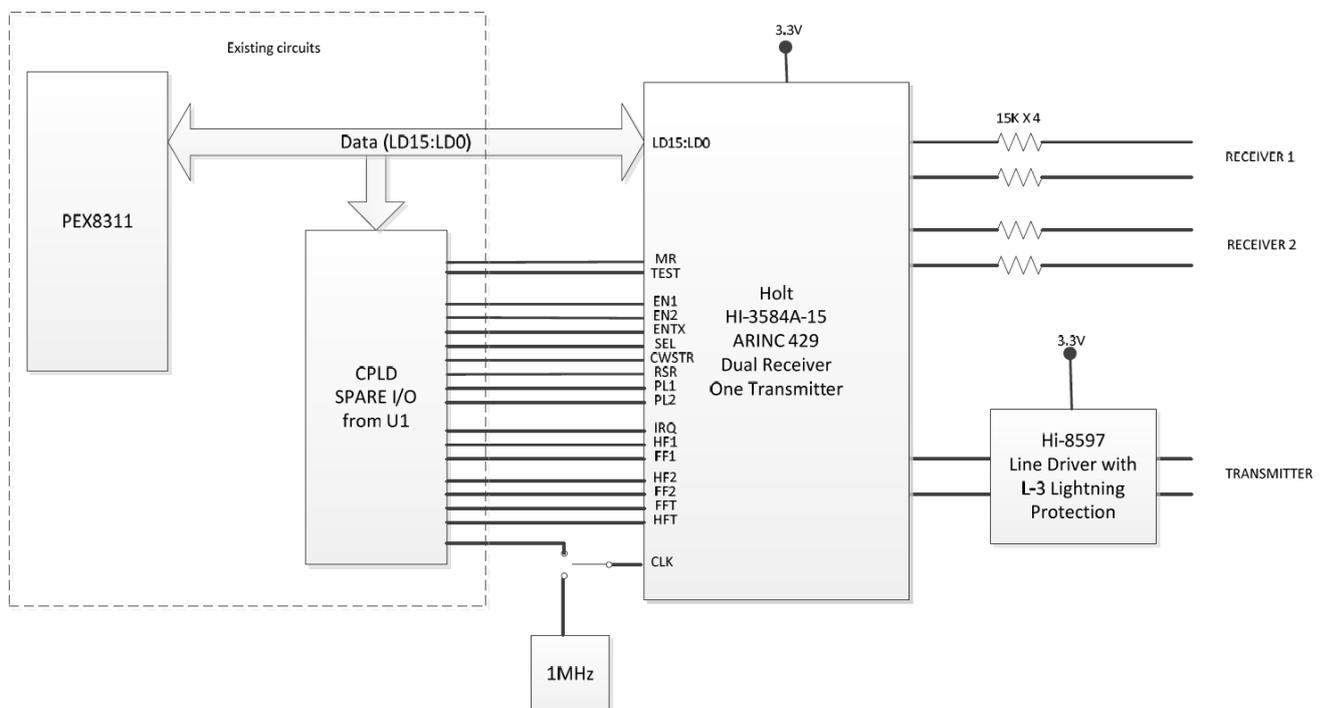
## Customization [カスタマイズ]

この章では、設計を向上させる方法のガイドラインを示します。これは、新しい設計を必要とする可能性が最も高いですが、J12 および J3 ヘッダー・コネクタで 14 個の CPLD I/O を使用する小型のアドオンボードでは、限られた試作が可能です。PCIe 12V、3.3V の電圧とグランド接続がこれらのコネクタに供給されています。これらの接続を使用して回路を追加する場合、特に MIL-STD-1553 を使用する場合は、電源の適切性を慎重に再評価する必要があります。

### ARINC 429

Holt には、この設計で PCIe ローカルバス (LB) と CPLD にインターフェイスできる ARINC 429 プロトコル・レシーバとトランスミッタがいくつかあります。同じハードウェアとソフトウェアの技術がすべて適用されます。パラレル・インターフェイスを持つデバイスは、LB 上でインターフェイスするのが最も簡単です。推奨される ARINC 429 16Bit パラレル製品の中には、HI-3582A、HI-3583A、HI-3584A があります。3.3V の唯一の解決策を得るには、HI-8584 または HI-8597 (レベル 3 雷保護を内蔵) などの 3.3V 単一電源レールライン・ドライバで HI-3584A を使用してください。

次のブロック図は、2 つの ARINC 429 レシーバと DO-160 レベル 3 雷保護を満たす単一 3.3V 電源で動作する一つのトランスミッタのインターフェイスを示しています。HI-3584A への 1MHz CLK 入力は、CPLD またはオプションで外部発振器モジュールから直接供給することができます。これらの ARINC 429 デバイスの詳細については、HI-3584A、HI8597 データシートおよび AN-3582A アプリケーション・ノートを参照してください。他の推奨事項については Holt 代理店にお問い合わせください。



### Discrete-to-Digital I/O Sensing

Holt HI-8425、HI-8426、HI-8430 または HI-8431 製品を使用して、この設計に Discrete to Digital 検出機能を簡単に追加できます。これらには、ハイサイドまたはローサイドの 4 つの出力ドライバも含まれます。これらのデバイスは、入力に調整可能な閾値と wetting currents および、出力に過電流フォルト保護を備えています。これらのデバイスは、入力および出力信号にパラレル I/O を使用し、スペアの CPLD ピンのいくつかによって駆動することができます。HI-8425 および HI-8430 は Discrete to Digital ファミリーの最新のメンバで他のものも使用できます。他の推奨事項については、Holt 販売店にお問い合わせください。

### EBR-1553 BC/MT/RT Terminals [EBR-1553 BC/MT/RT ターミナル]

HI-6140 は、HI-6130 を置き換えて 10Mbit/s の拡張ビット・レート MIL-STD-1553B ターミナル通信を実現することができます。HI-6140 は、HI-6130 と同じ 16Bit パラレル・バスと 50MHz クロック入力を使用し、レジスタ・セットとアーキテクチャが非常に似ています。Holt HI-4853 スルーレート制御 RS-485 トランシーバを使用して EBR バスを駆動します。この変更にはアドオンボードではなく、ボードの改訂が必要です。データシートおよびアプリケーション・ノートについては、Holt の Web サイトの HI-6140 製品ページを参照するか、推奨事項については Holt の販売店にお問い合わせください。

### MIL-STD-1553 Terminal Options [MIL-STD-1553 ターミナル・オプション]

HI-6130 に加えて、HI-6120 は単一の RT が必要な場合に適しています。HI-6120 は同じ 16Bit パラレル・バス・インターフェイスと 50MHz クロック入力を HI-6130 と同様のピン配列で共有しています。HI-6120 の RT 動作は HI-6130 とほぼ同じですが、レジスタとピン配列には多少の違いがあります。HI-6120 は、使用が簡単で、HI-6130 よりも安価です。この変更にはアドオンボードではなくボードの改訂が必要です。

### Power supplies [電源供給]

PCIe コンピュータ・アドイン・カード・スロットには、カスタムカードを設計するときに考慮すべき制限があります。PC アドイン・カード・ソケットでは、PCI Express 仕様によると、3.3V 電源は±9%、12V 電源は±8%の変動が可能です。HI-6130 は±5%の電圧許容範囲内で動作するように設計されているため、3.3V 電源から直接デバイスに電力を共有すると送信中に MIL-STD-1553 の性能特性を満たさなくなります。電源のレギュレーションが不十分で、過渡応答が適切ではない可能性が高いです。実際の電源レール特性は、コンピュータによって大きく異なります。HI-6130 や HI-6120 のような MIL-STD-1553 デバイスは、1553 送信中に 0.9A まで上がります。2 段階の電力変換が必要です。Holt カードは、12V~5V の DC-DC コンバータを使用して 3.3V のリニア・レギュレータに供給します。3.3V レギュレータにより厳密に調整された出力は、HI-6130 と CPLD に電力を供給します。

12V レールからの PCIe 電流消費量は、x1 カードの場合は 0.5A、x4/x8 カードの場合は 2.1A、x16 カードの場合は 4.4A を超えることはできません。PCIe x1 カード (Holt の例のような) の場合、スロット 12V レールからの最悪ケースの使用可能な電力は  $(12V \cdot 8\%) \times 0.5A = 5.52W$  です。DC-DC 変換効率は 90%で、使用可能な負荷容量は 5.52 から 4.97W に減少します。最終的な 5V~3.3V の段階ではリニア・レギュレータを使用するため、出力電流はレギュレータ入力電流に等しくなります。したがって、最大出力不可電流は  $4.97W / 5V = 0.99A$  です。3.3V レギュレータは 3.27W を供給できます。より小さい設計には x1 スロットを使用し、より大きい電力を必要とする設計には x4 / x8 または x16 スロットを使用します。

---

新しいカスタム設計では、カードのサイズ（フルハイトとロープロファイル）および使用されるスロット・タイプ（x1～x16）に応じて変化する最大許容電流および消費電力に関する PCIe に制限事項に注意する必要があります。

### **Clocks [クロック]**

単一の 50MHz 発振器モジュールが PEX8311 LB、CPLD、HI-6130 の間で共有されています。LB は、より複雑な設計では望ましい高速バス性能を達成するために、最大 66MHz までクロックすることができます。HI-6130 は常に 50MHz でクロックされなければならないので、追加のオシレータが必要になります。LB を 66MHz で動作させる場合、PLX LB EPROM レジスタのウェイト数を増やす必要があります。LB と CPLD へ、CPLD 内部の同期ストローク信号と HI-6130 への CSn、RDn、WRn ストローク信号を生成するために同じクロックを使用する必要があります。より高速な LB の場合、PEX8311 と CPLD の LB クロック入力には最大 66MHz を使用し、HI-6130 には別の 50MHz クロックを使用します。

### **Additional memory [追加メモリ]**

適切な設計により、オンボード・メモリを追加することが可能です。アドレスとデータバスを PEX8311 の LB 信号に直接接続し、スペアの CPLD ピンを使用して必要な数の制御信号をメモリに割り当て、それに応じて Verilog を変更します。目的のメモリ空間、ウェイト・ステート数、およびバス幅タイプの LB EEPROM スペース・レジスタを設定します。

### **PCB layout considerations [PCB レイアウトの考慮事項]**

PCIe 高速バス信号については、PLX のデータブックとハードウェアのチェックリストに従ってください。HI-6120、HI-6130 および HI-6140 トランスおよびデカップリング・コンデンサの PCB レイアウト・ガイドラインについては、Holt AN-550 を参照してください。

## まとめ

Holt HI-6130 PCIe ロープロファイル・カード・リファレンス設計は、HI-6130 MIL-STD-1553 マルチ・ターミナルを 1 レーン (x1) PCIe バスに接続する方法を示しています。デモ・ソフトウェアでは、Holt のハイレベル API ソフトウェア・ライブラリが提供されデモされています。すべての設計ファイルは、OrCAD 回路図、2 つのソフトウェア・デモ・プロジェクト、CPLD 用の Verilog ソース、迅速なカスタム開発を可能にするその他の関連ドキュメントを含む C-ROM に含まれています。追加の MIL-STD ターミナル、ARINC 429 プロトコル IC、Discrete to Digital とメモリを使用して設計を行い、カスタマイズする方法についていくつか指定しました。この設計に関するご質問は、Holt に連絡してください。PLX PEX8311 および PLX API ソフトウェアおよびドライバのサポートについては、Web サイトのテクニカルサポートページを使用して Avago に直接お問い合わせください。

参照：

<http://www.holtic.com/category/351-mil-std-1553.aspx>

<http://www.holtic.com/category/352-arinc-429.aspx>

[www.holtic.com/category/420-discrete-to-digital-components.aspx](http://www.holtic.com/category/420-discrete-to-digital-components.aspx)

<http://www.pcisig.com/home>

<http://www.avagotech.com/products/pcie-switches-bridges/pcie-bridges/pex8311>

<http://www.avagotech.com/support/request-tech-support>

<http://www.microsoft.com/visualstudio/eng/downloads>

注：インターネット・リンクが時々古くなるので、メインの Web サイトをナビゲートし、必要な情報を探す必要があるかもしれません。

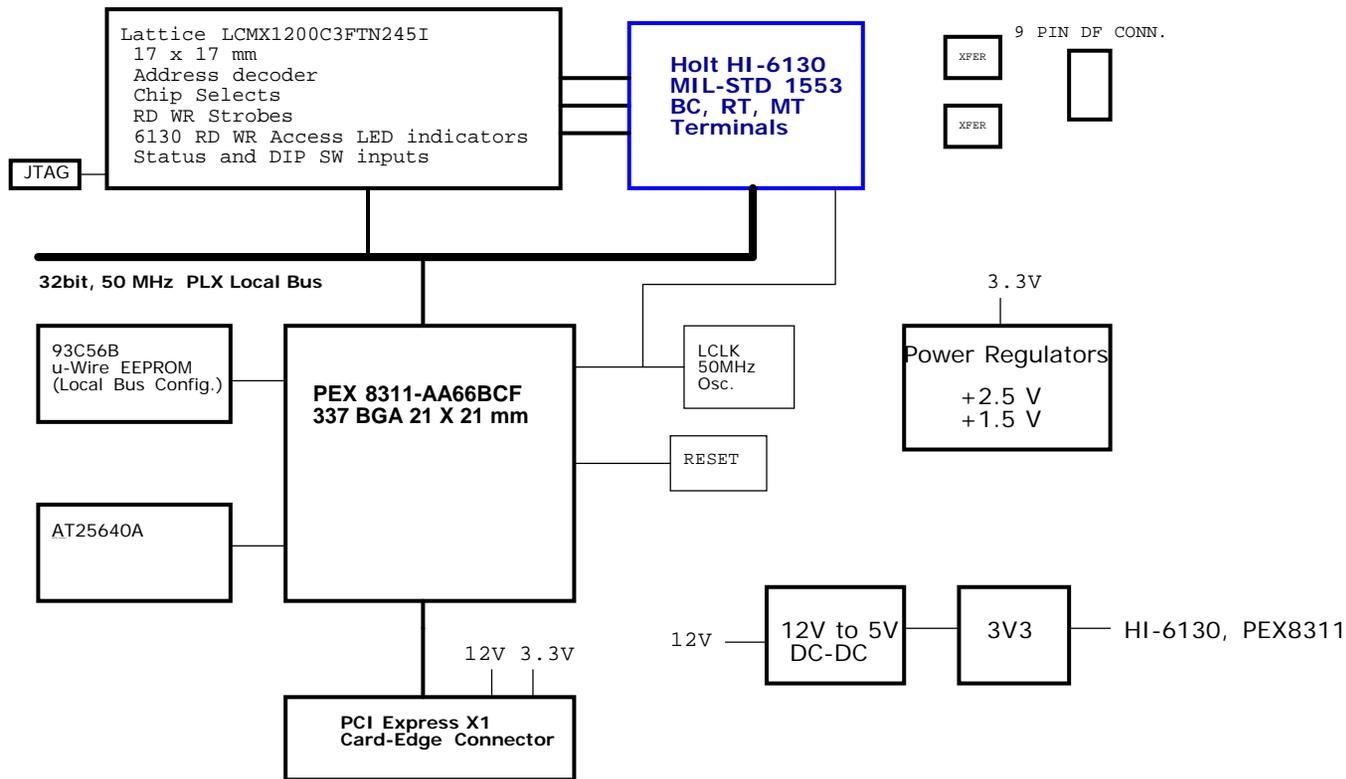
**Table Of Contents**

**Block Diagram**

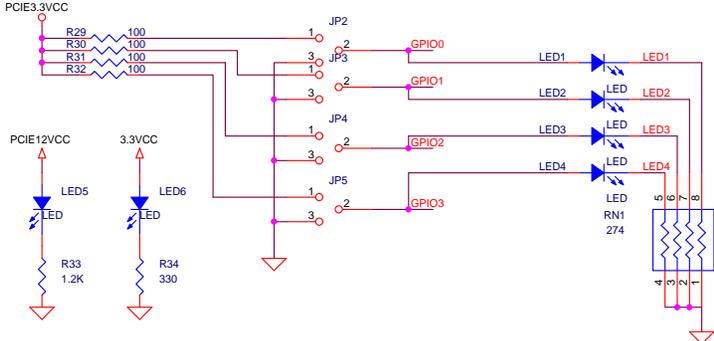
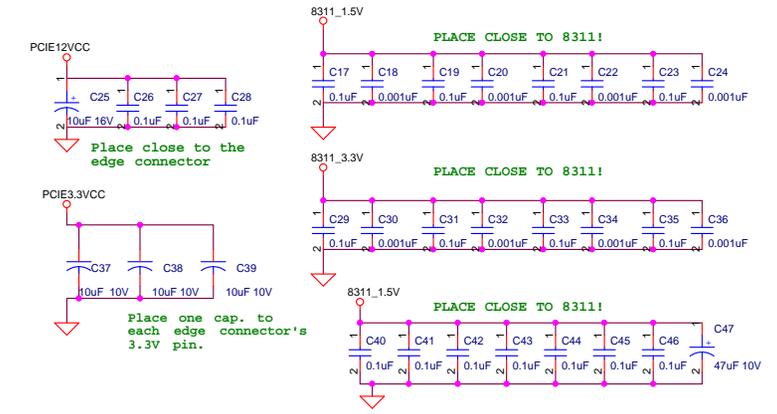
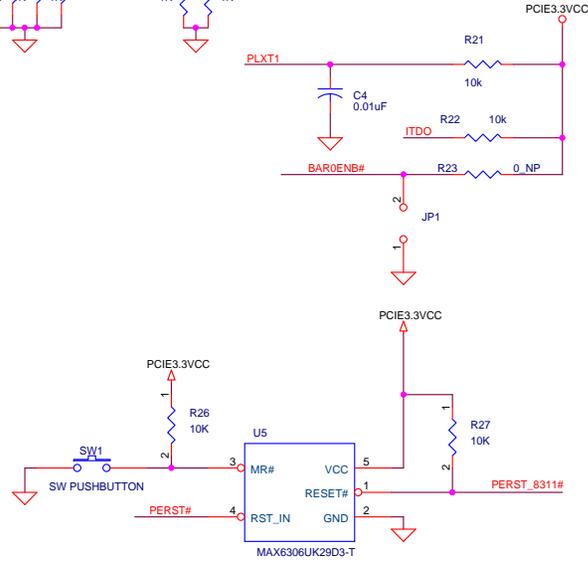
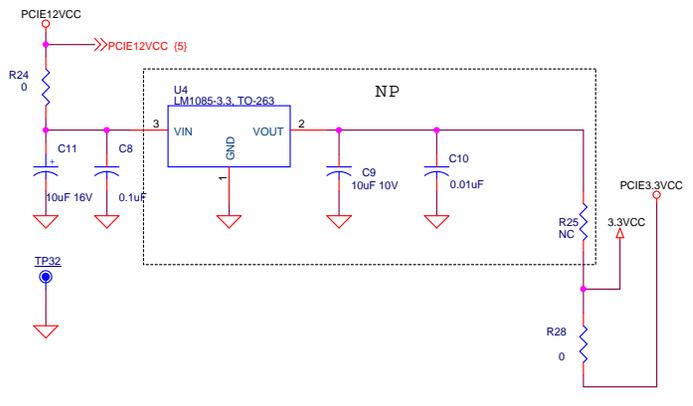
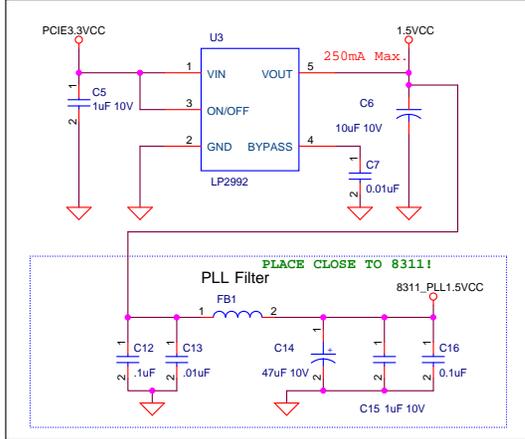
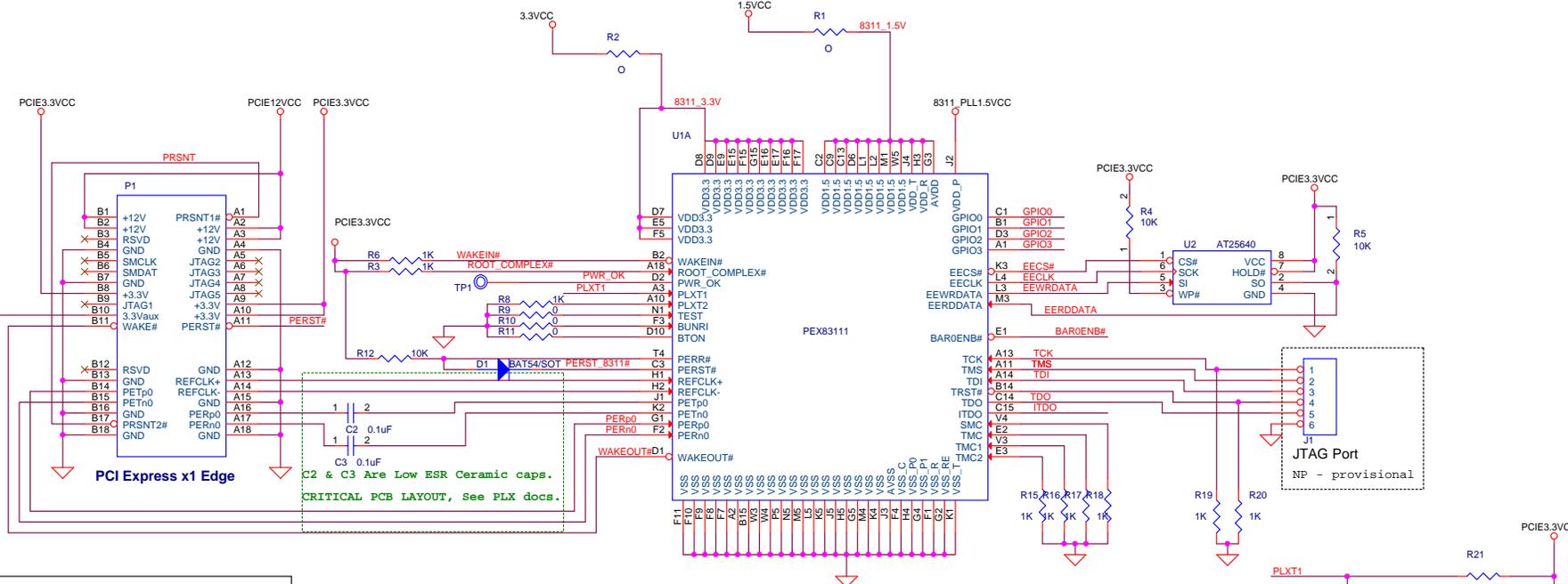
Date	Changes
------	---------

9/19/2013 Rev A

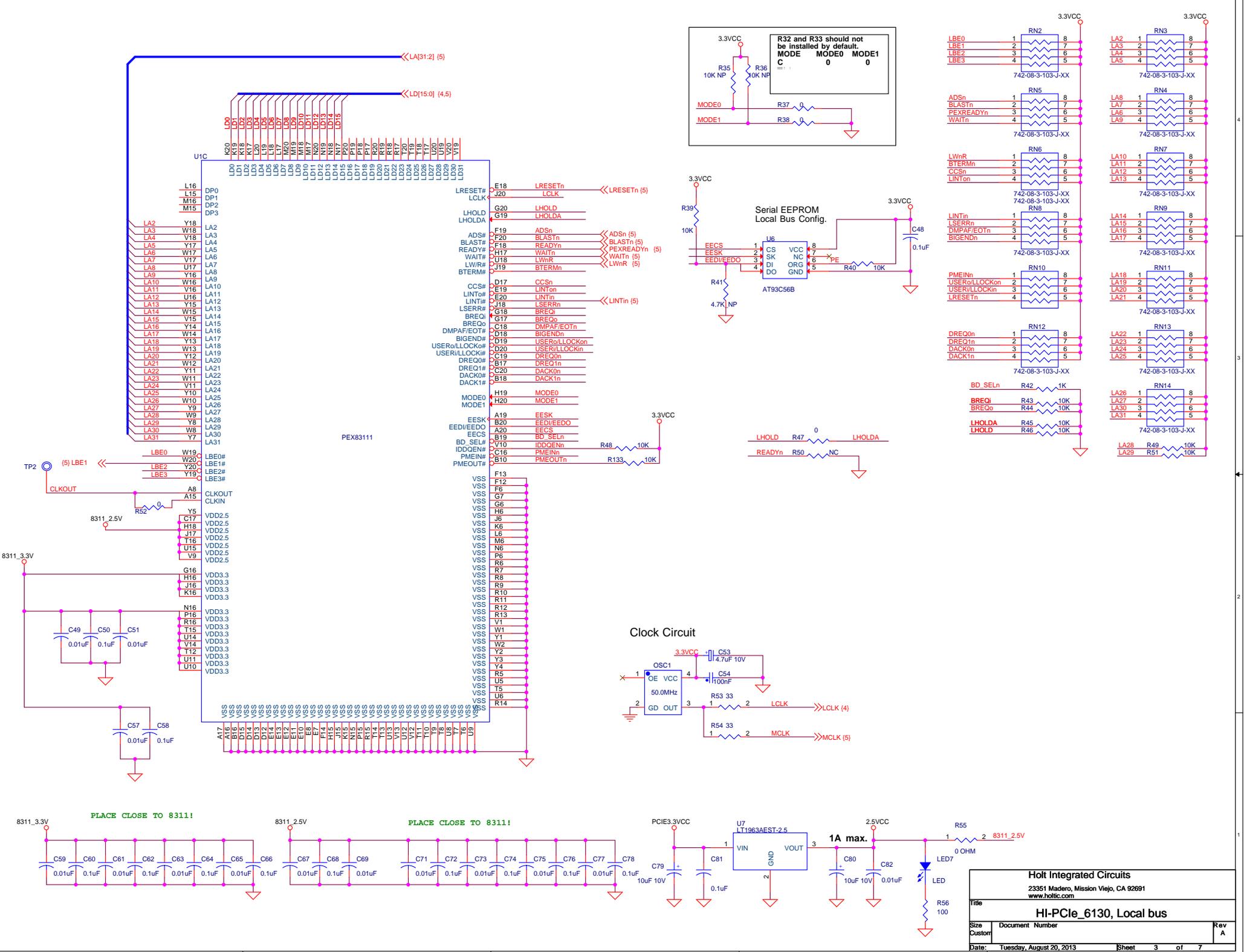
- 1: Cover Page
- 2: PEX8311 PCI Express Bus
- 3: PLX8311 Local Bus
- 4: CPLD JTAG/6130 Inputs
- 5: HI -6130
- 6: CPLD - POWER
- 7: PEX8311 NC Balls



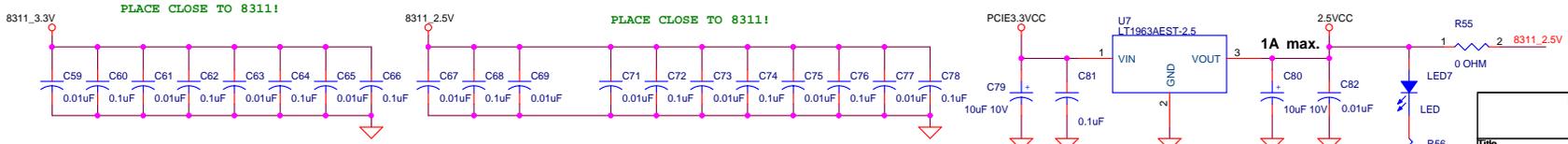
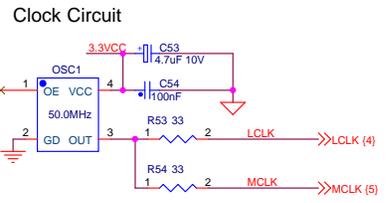
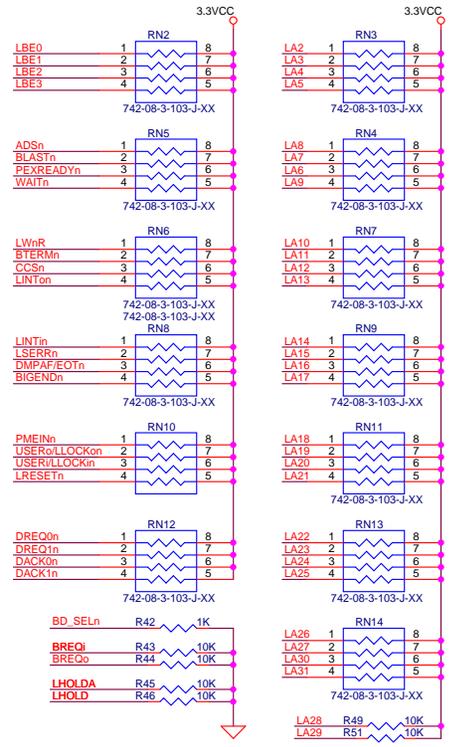
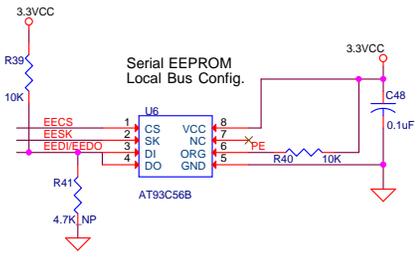
<b>Holt Integrated Circuits</b> 23351 Madero, Mission Viejo, CA 92691 <a href="http://www.hltic.com">www.hltic.com</a>		
<b>Title</b> HI-PCIe_6130		
<b>Size</b> B	<b>Document Number</b>	<b>Rev</b> A
<b>Date:</b> Thursday, September 19, 2013		
<b>Sheet</b> 1		<b>of</b> 7



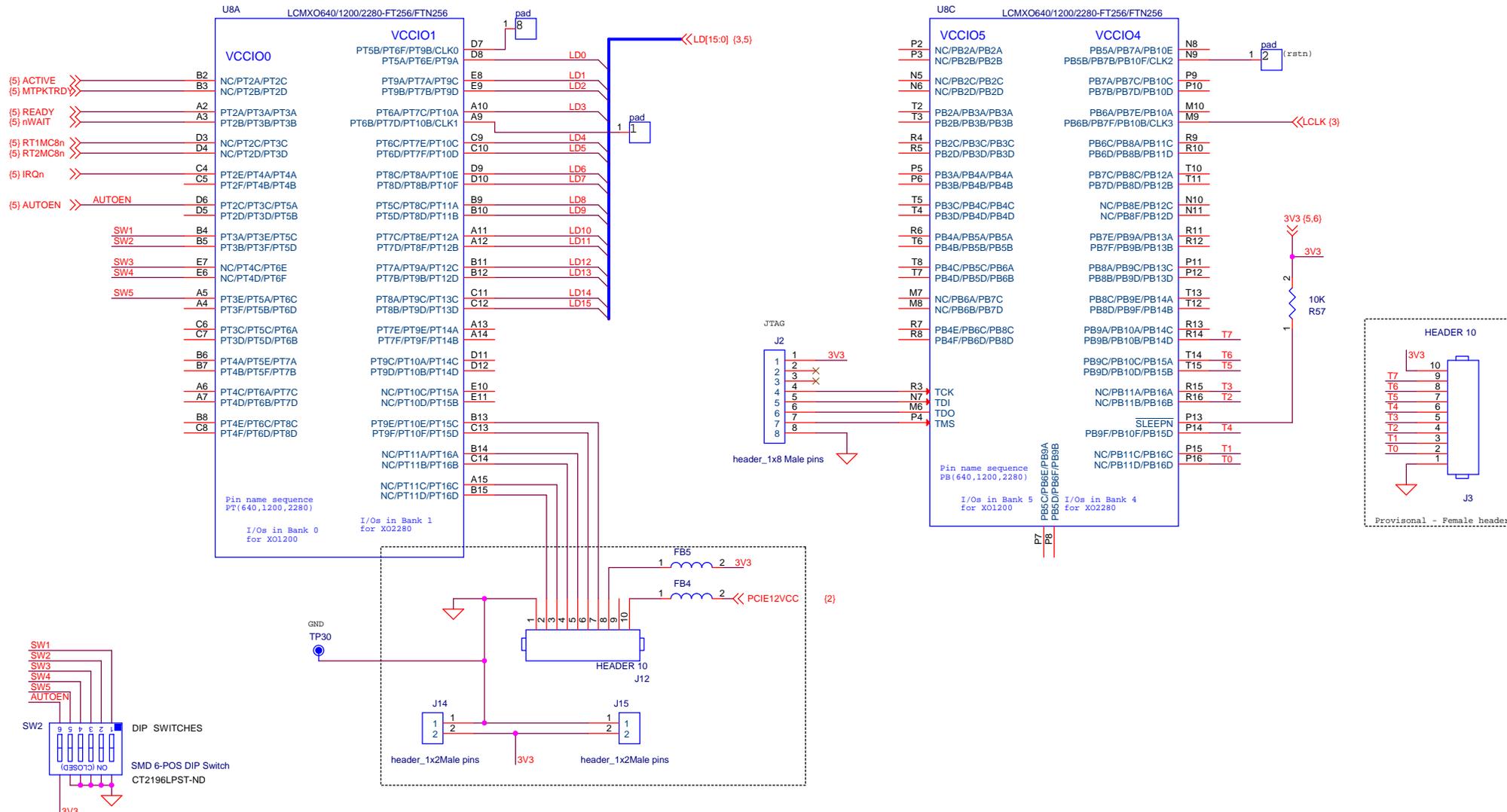
Holt Integrated Circuits 23351 Madero, Mission Viejo, CA 92691 www.hltic.com		
Title	HI-PCIe_6130, PCIe bus	
Size	Document Number	Rev A
Customer		
Date:	Thursday, September 19, 2013	Sheet 2 of 7



R32 and R33 should not be installed by default.  
 MODE MODE0 MODE1  
 C 0 0



Holt Integrated Circuits		
23351 Madero, Mission Viejo, CA 92691		
www.holtec.com		
Title	HI-PCIe_6130, Local bus	
Size	Document Number	Rev
Date:	Tuesday, August 20, 2013	Sheet 3 of 7



Default: SW2 position 6 (AUTOEN) should be Open (UP).

<b>Holt Integrated Circuits</b>		www.holtic.com
23351 Madero, Mission Viejo, CA 92691		
<b>HI-6130 Status Inputs, JTAG</b>		
Title	Document Number	
Size 3		Rev A
Date: Thursday, September 19, 2013	Sheet 4 of 7	

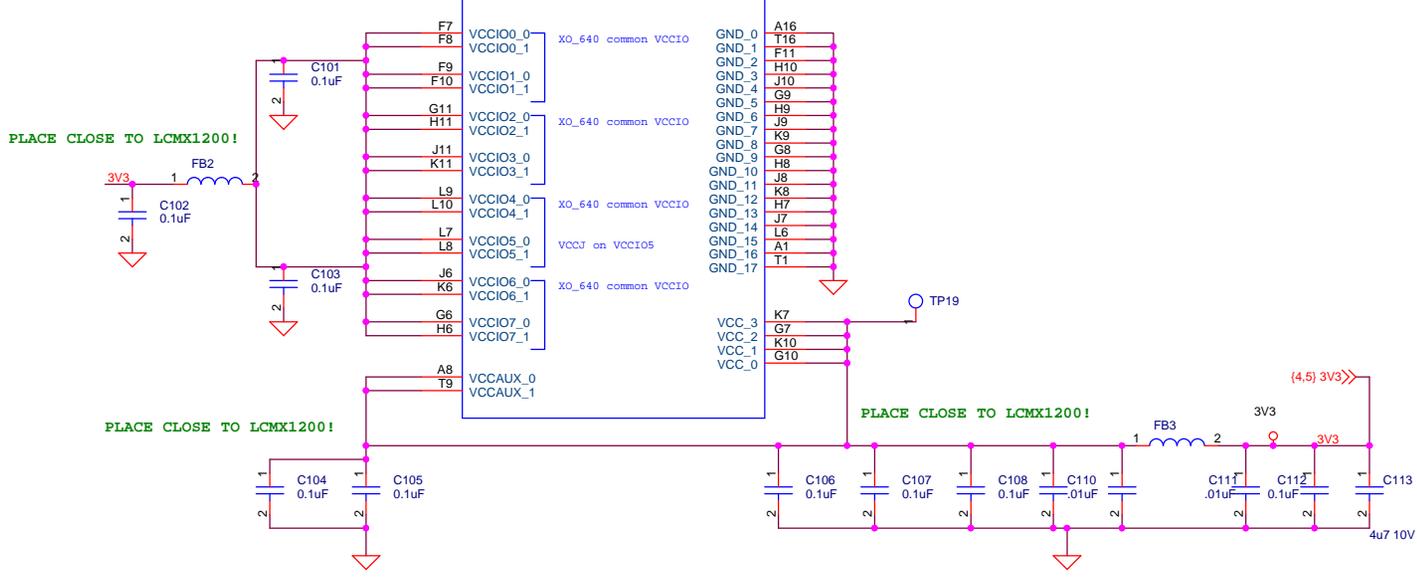


U8D LCMXO640/1200/2280-FT256/FTN256

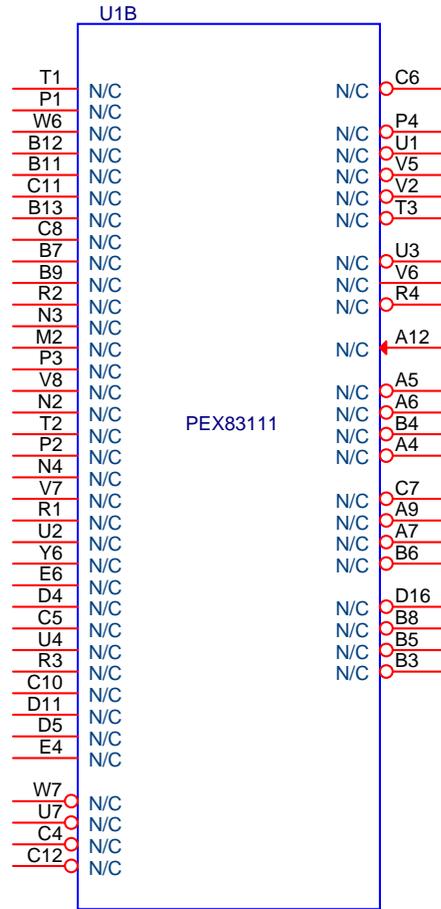
VCCIO2		VCCIO3		
D14	NC/PR2A/PR3A/LV_T	PR7A/PR9A/PR11A/LV_T	H16	
D13	NC/PR2B/PR3B/LV_C	PR7B/PR9B/PR11B/LV_C	J16	
E13	NC/PR3A/PR4A/LV_T	NC/PR9C/PR11C	J12	
E12	NC/PR3B/PR4B/LV_C	NC/PR9D/PR11D	K12	
F13	NC/PR3C/PR4C	PR7C/PR10A/PR13A/LV_T	J15	
F12	NC/PR3D/PR4D	PR7D/PR10B/PR13B/LV_C	K15	
E14	PR3C/PR4A/PR5A/LV_T	PR8A/PR10C/PR13C	J14	
F14	PR3D/PR4B/PR5B/LV_C	PR8B/PR10D/PR13D	K14	
B16	PR2A/PR4C/PR5C	PR8C/PR11A/PR14A/LV_T	J13	
C16	PR2B/PR4D/PR5D	PR8D/PR11B/PR14B/LV_C	K13	
C15	PR2C/PR5A/PR6A/LV_T	PR9A/PR11C/PR14C	K16	
D15	PR2D/PR5B/PR6B/LV_C	PR9B/PR11D/PR14D	L16	
D16	PR3A/PR5C/PR6C	PR9C/PR12A/PR15A/LV_T	L15	
E16	PR3B/PR5D/PR6D	PR9D/PR12B/PR15B/LV_C	M15	
E15	PR4A/PR6A/PR7A/LV_T	PR10C/PR12C/PR15C	M16	
F15	PR4B/PR6B/PR7B/LV_C	PR10D/PR12D/PR15D	N16	
F16	PR5A/PR6C/PR7C	PR10A/PR13A/PR16A/LV_T	L14	
G16	PR5B/PR6D/PR7D	PR10B/PR13B/PR16B/LV_C	M14	
G12	PR4C/PR7A/PR9A/LV_T	PR11A/PR13C/PR16C	L12	
G13	PR4D/PR7B/PR9B/LV_C	PR11B/PR13D/PR16D	L13	
H12	PR6C/PR7C/PR9C	PR11C/PR14A/PR17A/LV_T	N15	
H13	PR6D/PR7D/PR9D	PR11D/PR14B/PR17B/LV_C	N14	
G14	PR5C/PR8A/PR10A/LV_T	NC/PR14C/PR17C	M12	
H14	PR5D/PR8B/PR10B/LV_C	NC/PR14D/PR17D	M13	
G15	PR6A/PR8C/PR10C	NC/PR15A/PR18A/LV_T	N13	
H15	PR6B/PR8D/PR10D	NC/PR15B/PR18B/LV_C	N12	
		NC/PR16A/PR20A	L11	
		NC/PR16B/PR20B	M11	

Pin name sequence  
PR (640, 1200, 2280)

U8E LCMXO640/1200/2280-FT256/FTN256

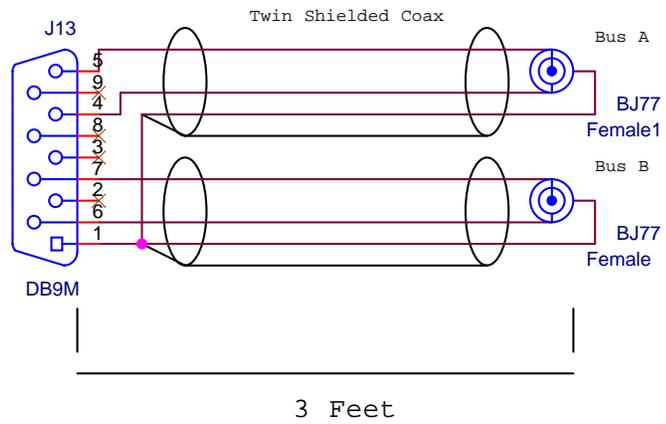


Holt Integrated Circuits		
23351 Madero, Mission Viejo, CA 92691		www.holtic.com
Title		
CPLD power		
Size	Document Number	Rev
B		
Date:	Wednesday, August 14, 2013	Sheet 6 of 7



Holt Integrated Circuits 23351 Madero, Mission Viejo, CA 92691			www.holtic.com
Title <b>PEX 8311, NC BALLS</b>			
Size A	Document Number <Doc>		Rev
Date: Tuesday, August 13, 2013			
Sheet 7 of 7			

Bill of Materials		PCIe HI-6130 Interface Board Rev. A			19-Sep-13
Item	Qty	Description	Reference	DigiKey	Mfr P/N
1	1	PCB, Bare, Evaluation Board	N/A	-----	Advanced Circuits 3568129-MQ
2	2	Ferrite Bead, 330 Ohm @ 100MHz 0805	FB4,FB5	490-5988-1-ND	BLM21PG331SN1D
3	3	Ferrite Bead, 390 Ohm @ 100MHz 500mA DC 0603	FB1,FB2,FB3	490-5980-1-ND	BLM18EG391TN1D
4	8	Capacitor, Ceramic 0.001uF 50V 10% X7R 0603	C18,C20,C22,C24,C30,C32,C34,C36	399-1082-1-ND	Kemet C0603C102K5RACTU
5	20	Capacitor, Ceramic 10nF 10% 50V X7R 0603	C4,C7,C13,C49,C51,C57,C59,C61,C63,C65,C67,C69,C71,C73,C75,C77,C82,C100,C110,C111	399-1091-1-ND	Kemet C0603C103K5RACTU
6	58	Capacitor, Ceramic 100nF 10% 50V X7R 0603	C1,C2,C3,C8,C12,C16,C17,C19,C21,C23,C26,C27,C28,C29,C31,C33,C35,C40,C41,C42,C43,C44,C45,C46,C48,C50,C54,C58,C60,C62,C64,C66,C68,C72,C74,C76,C78,C81,C83,C85,C87,C88,C89,C91,C92,C94,C95,C98,C101,C102,C103,C104,C105,C106,	399-5089-1-ND	Kemet C0603C104K5RACTU
7	2	Capacitor, Ceramic 1uF 10V 10% X7R 0603	C5,C15	399-9449-1-ND	Kemet T491A105K010AT
8	1	Capacitor, Ceramic 4.7uF 10V 10% X5R 0603	C113	399-5503-1-ND	Kemet C0603C475K8PACTU
9	2	Capacitor, Tantalum 4.7uF 10% 10V Size A	C53,C86	399-3699-1-ND	Kemet T491A475M016AT
10	12	Capacitor, Tantalum 10uF 10V 10% Size A	C6,C10,C14,C37,C38,C39,C79,C80,C90,C93,C96,C99	399-3684-1-ND	Kemet T491A106K010AT
11	3	Capacitor, Tantalum 10uF 16V 10% Size A	C11,C25,C97	399-3687-1-ND	Kemet T491A106M016AT
12	2	Capacitor 47uF 10% 6.3V Tantalum Low ESR SMD C	C14,C47	495-1543-1-ND	Kemet T495C476K016ZTE300
13	1	Capacitor 68uF 10% 6.3V Tantalum Low ESR SMD C	C84	495-1507-1-ND	Kemet T495C107K006ZTE150
14	1	Resistor Array 270 Ohm 4 Res 1206	RN1	742C083271JPCT-ND	CTS 742C083271JP
15	13	Resistor Array 10K Ohm 4 Res 1206	RN2,RN3,RN4,RN5,RN6,RN7,RN8,RN9,RN10,RN11,RN12,RN13,RN14	742C083103JPCT-ND	CTS 742C083103JP
16	1	Resistor, 0.0 Ohm 1/10W 0603	R23	DNP	Panasonic ERJ-3GEY0R00V
17	1	Resistor, Provision, 5% 1/10W 0603	R50	DNP	
18	1	Resistor, 4.7K ohm 5% 1/10W 0603	R41	DNP	Panasonic ERJ-3GEYJ472V
19	2	Resistor, 10K Ohm 5% 1/10W 0603	R35,R36	DNP	Panasonic ERJ-3GEYJ103V
20	2	Resistor 0.0 Ohms, 1/8W, 0805	R25,R71	DNP	Panasonic ERJ-6GEY0R00V
21	5	Resistor 0.0 Ohms, 1/8W, 0805	R2,R24,R28,R55,R67	P0.0ACT-ND	Panasonic ERJ-6GEY0R00V
22	8	Resistor, 0.0 Ohm, 1/10W, 0603	R1,R9,R10,R11,R37,R38,R47,R52	P0.0GCT-ND	Panasonic ERJ-3GEY0R00V
23	3	Resistor, 33 Ohm 5% 1/10W 0603	R53,R54,R70	P33GCT-ND	Panasonic ERJ-3GEYJ330V
24	5	Resistor, 100 Ohm 5% 1/10W 0603	R29,R30,R31,R32,R56	P100GCT-ND	Panasonic ERJ-3GEYJ101V
25	5	Resistor, 330 ohm 5% 1/10W 0603	R34,R64,R66,R68,R69	P330GCT-ND	Panasonic ERJ-3GEYJ331V
26	10	Resistor, 1K Ohm 5% 1/10W 0603	R3,R6,R8,R15,R16,R17,R18,R19,R20,R42	P1.0KGCT-ND	Panasonic ERJ-3GEYJ102V
27	1	Resistor, 1.2K Ohm 5% 1/10W 0603	R33	P1.2KGCT-ND	Panasonic ERJ-3GEYJ122V
28	21	Resistor, 10K Ohm 5% 1/10W 0603	R4,R5,R12,R21,R22,R26,R27,R39,R40,R43,R44,R45,R46,R48,R49,R51,R57,R63,R65,R72,R133	P10KGCT-ND	Panasonic ERJ-3GEYJ103V
29	4	Header 1x3, Male, 0.1" Pitch	JP2,JP3,JP4,JP5	S1012E-03-ND	Sullins PEC36SAAN
30	4	Shunt 1x2, 0.1" Pitch	JP2,JP3,JP4,JP5	A26227-ND	TE 382811-6
31	3	Header 1x2, Male, 0.1" Pitch	JP1,J14,J15	S1012E-02-ND	Sullins PEC36SAAN
32	1	Header 1x6, Male, 0.1" Pitch	J1	S1012E-06-ND	Sullins PEC36SAAN
33	2	Header 1x8, Male, 0.1" Pitch,	J2,J5	S1012E-08-ND	Sullins PEC36SAAN
34	2	Header 1x10, Male, 0.1" Pitch,	J3,J12	S1012E-10-ND	Sullins PEC36SAAN
35	1	Osc 50MHz, 25ppm, 3.3V SMD 5x7mm	Osc1	535-10087-1-ND	Abracon ASV-50.000MHZ-E-T
36	11	LED Green 0805	LED1-LED11	160-1179-1-ND	Lite-On LTST-C170GKT
37	1	Diode Schottky 40V 0.3A SOT23	D1	497-12131-1-ND	ST BAT54FILMY
38	1	Switch Tactile SPST-NO 0.02A 15V	SW1	P12943SCT-ND	Panasonic EVQ-Q2K03W
39	3	DIP Switch 6-Pos Half Pitch SMD	SW2,SW3,SW4	CT2186LPST-ND	CTS 218-6LPST
40	2	Test Point, Red, 0.062" hole	TP8,TP11	5010K-ND	Keystone 5010
41	3	Test Point, Black, 0.062" hole	TP9,TP12,TP32	5011K-ND	Keystone 5011
42	1	Test Point, White, 0.062" hole	TP20 - Active	5012K-ND	Keystone 5012
43	4	Test Point, White, 0.040" hole	TP10,TP13,TP14,TP31	5002K-ND	Keystone 5002
44	2	Test Point, 0.040" hole	TP2,TP19	N/A	
45	7	Test Point Pad	1,2,3,4,5,6,7	N/A	
46	1	Connector DB9F, R/A PCB Short Body, Board Lock	J4	A35107-ND	TE 1734354-1
47	1	PEX 8311 PCIe IC - 337BGA	U1	862-PEX8311-AA66BC-F (Mouser PN)	PLX PEX-8311-AA66BCF
48	1	IC PLD 1200LUTs 211 I/O 256FTBGA	U8	220-1064-ND	Lattice LCMXO1200C-3FTN256I
49	1	IC EEPROM 512Kbit 20 MHz 8-SOIC	U11	25LC512T-I/SNCT-ND	Micro Chip 25LC512T-I/SN
50	1	IC, EEPROM 64K 20MHz 8-SOIC	U2	AT25640B-SSHL-T-ND	Atmel AT25640B-SSHL-T
51	1	IC HI-6130 100-PQFP	U9	HI-6130	Holt IC
52	2	Transformer PM-DB2791S 2.5T	T1,T2	PM-DB2791S	Holt IC
53	1	Regulator LDO 1.5V 0.25A SOT23-5	U3	LP2992IM5-1.5/NOPBCT-ND	LP2992IM5-1.5/NOPB
54	1	Regulator 3.3V 3A, DDPK TO-263-4	U10	LM1085IS-3.3/NOPB-ND	TI LM1085ISX-3.3/NOPB
55	1	Converter DC/DC 1A 5V Out SIP Vertical	U4	945-1038-ND	Recom Power R-785-0-1.0
56	1	IC Reg LDO 2.5V 1.5A SOT223	U7	LT1963AEST-2.5#PBF-ND	Linear LT1963AEST-2.5#PBF
57	1	Reset IC, 2.93V, SOT-23-5L	U5	MAX6306UK29D3+TCT-ND	Maxim MAX6306UK29D3+T
58	1	IC, 3-wire EEPROM 2Kb, 8-SOIC	U6	AT93C56B-SSHM-TCT-ND	Atmel AT93C56B-SSHM-T
59	1	Low Profile PCIe Bracket w/ DB 9 Opening	N/A	Low Bracket Star Mfg	Star Manufacturing
60	1	High Profile PCIe Bracket w/ DB 9 Opening	N/A	High Bracket Star Mfg	Star Manufacturing
61	2	Triax Connector w/ Bend Relief Spring Plug Crimp	N/A	Mouser 530-PL75-29	Trompeter/Emerson PL75-29
63	2	Two 3ft in Length Cable	N/A	Mouser: 530-TWC-78-1/500ft	Twinax TWC-78-1 - 500ft Reel
64	1	Conn DB9 Male Solder Cup Nickel w/ Gold Plated Pins	N/A	209ME-ND	Norcomp 171-009-103L001
65	1	Backshell DB9 Die Cast Black Chrm	N/A	970-09BCA-ND	Norcomp 970-009-040R011



<b>Holt Integrated Circuits</b> 23351 Madero, Mission Viejo, CA 92691 <a href="http://www.holtic.com">www.holtic.com</a>		
<b>Title</b> HI-6130 PCIe MIL-STD-1553 Cable		
<b>Size</b> A	<b>Document Number</b>	<b>Rev</b> NEW
<b>Date:</b> September 2013		<b>Sheet</b> 1 of 1

## REVISION HISTORY

---

P/N	Rev	Date	Description of Change
AN-6130PCle	NEW	09/27/2013	Release
AN-6130PCle	A	06/30/2014	Revise for API demo program changes.
AN-6130PCle	B	03/04/2015	Update board photo on page 1.
AN-6130PCle	C	02/02/2016	Updates for Holt API 2.x and PLX SDK 7.2.

---