KJ-0836

CL Getter
IP 仕様/アプリケーション
マニュアル

株式会社　ケーアイテクノロジー

変更履歴

版数	日付	変更箇所	変更内容	作成
初版	2010/01/18	新規作成		KIT

目　次

はじめに

【輸出する際の注意事項】

本製品は日本国内での使用に限定しております。本製品を日本国外で使用された場合、弊社は一切責任を負いかねます。

また、弊社では本製品に関し、海外での保守及び技術サポートは行っておりません。

【ご注意】

1. 本マニュアルの内容の一部または全部を無断で転載することは禁止されています。
2. 本マニュアルの内容に関しては将来予告無しに変更することがあります。
3. 本マニュアルの内容について万全を期しておりますが、万一ご不審な点や誤り、記載もれ等お気付きの事がございましたらお手数ではございますが、弊社サポート窓口まで御連絡下さい。
4. 動作の影響に関しましては 3.項に関わらず責任を負いかねますのでご了承下さい。
5. 各ソフト(ISE、ModelSim 等)の操作については各々の操作説明書を参照して下さい。
   (本書で記載してます操作例等と操作説明書と異なる場合は(ソフトの)操作説明書を優先します。)
6. 医療機器や人命に直接的または間接的に関わるシステムなど、高い安全性が要求される用途には使用しないでください。

1.概要

本書は、CL Getter(KIT1140+KIT2010)による CL Getter 用の IP 仕様/アプリケーションマニュアルです。
構成としまして、

1 章：概要
2 章：CL Getter IP 仕様について(CL Getter IP についての説明)
3 章：CL Getter について
4 章：シミュレーション

となっております。

付属 CD 内にあります FPGA のサンプルソース及びテストベンチは JAI 製 CV-M71CL カメラ用のサンプルとなります。

・搭載 FPGA

Xilinx 社　Spartan3 シリーズ　XC3S5000-4FGG900C
開発環境：Xilinx 社　ISE10.1 Fundation 以降　(Web Pack は不可)　**※FPGA 変更の際に必要です。**
ダウンロードケーブル(Platform Cable USB　II)　**※FPGA 変更の際に必要です。**

2.CL Getter IP 仕様について

2.1.CL Getter IP 概要

CL Getter IP は基本機能としまして以下の機能を有します。

① キャプチャーデータパッキング回路

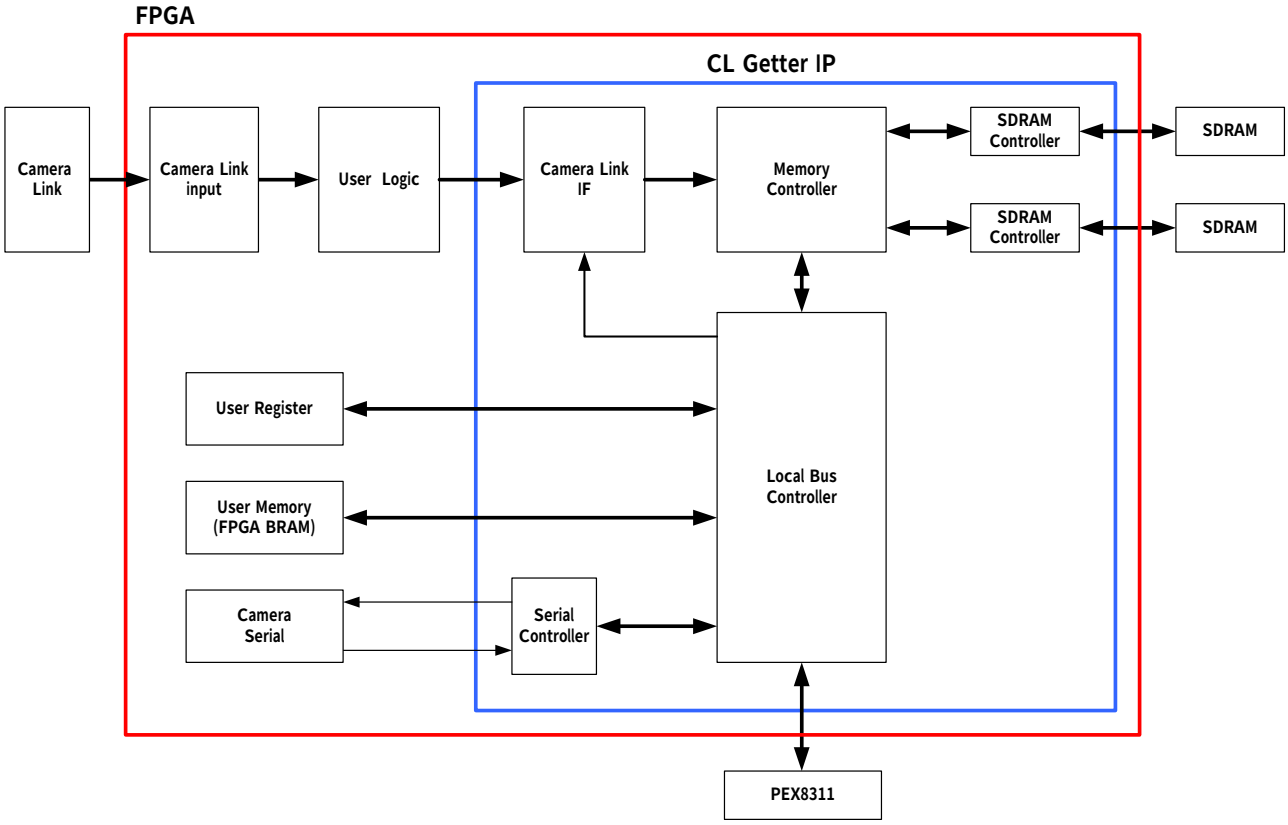
② 外部メモリコントローラ

③ ローカルバスコントロール

④ カメラ用シリアル通信制御

2.2.CL Getter IP ブロック図

CL Getter IP のブロック図を示します。青枠で囲まれた部分が CL Getter IP 部となります。



2.3.CL Getter IP 内部ブロック説明

CL Getter IP 内部のブロック毎機能を説明します。

・ IP IF Control部	**キャプチャーデータのパッキング等、ユーザー回路とIP回路の入力I/Fコントロールを行います。**
・ Memory Controller部	**IPへ入力されたデータとSDRAM Controller部へのデータ受け渡しや画像データをローカルバスへ転送コントロールを行います。**
・ SDRAM Controller部	**Memory Controller部よりデータ、R/Wリクエストを SDRAMコントロール信号として変換を行います。**
・ Local Bus Controller部	**ローカルバスのR/Wコントロールを行います。**
・ Serial Controller部	**カメラリンク用シリアル通信コントローラです。**

2.4.CL Getter IP で接続する信号について

トップモジュール上の camera_link_ip.v モジュールが CL Getter IP となります。

CL Getter IP モジュールの各信号ピンについて示します。

```
//+++ (5)  camera_link_ip module interface  +++//


(* box_type = "user_black_box" *)
camera_link_ip  camera_link_ip_0 (
        .FPGA_LCLK (FPGA_LCLK_I),           .NLRESET (NLRESET),             //PLX
        .LHOLD (LHOLD),                     .LA (LA),
        .NLADS (NLADS),                     .LWNR (LWNR),
        .nLBE (nLBE),                       .nLBLAST (nLBLAST),
        .LHOLDA (LHOLDA),                   .nLBTERM (nLBTERM),
        .nLREADY (nLREADY),                 .NLINTI (NLINTI),
        .NCCS (NCCS),                       .NDREQ0 (NDREQ0),
        .NDREQ1 (NDREQ1),                   .BREQI (BREQI),
        .NBIGEND (NBIGEND),                 .nLLOCKI (nLLOCKI),
        .NWAIT (NWAIT),
        .LD (LD),                           .DP (DP),

        .DEV_IO_CS(DEV_IO_CS),              .CPU_RD(CPU_RD),



        .CLINK_CLK (CLINK_CLK),             .CLINK_FVAL (USER_FVAL),        //Camera Link I/F
        .CLINK_DVAL (USER_DVAL),            .CLINK_IND (USER_IND),
        .SERTFG (SERTFG),
        .SERTC (SERTC),

        .SDCLK_FPGA (SDCLK_FPGA_I),                                         //SDRAM : bank0
        .BK0_BA (BK0_BA),                   .BK0_A (BK0_A),
        .BK0_NCS (BK0_NCS),                 .BK0_CKE (BK0_CKE),
        .BK0_NRAS (BK0_NRAS),               .BK0_NCAS (BK0_NCAS),
        .BK0_NWE (BK0_NWE),                 .BK0_DQMH1 (BK0_DQMH1),
        .BK0_DQMH0 (BK0_DQMH0),             .BK0_DQML1 (BK0_DQML1),
        .BK0_DQML0 (BK0_DQML0),
        .BK0_DQ (BK0_DQ),

        .BK1_NCS (BK1_NCS),                 .BK1_CKE (BK1_CKE),             //SDRAM : bank1
        .BK1_BA (BK1_BA),                   .BK1_A (BK1_A),
        .BK1_NRAS (BK1_NRAS),               .BK1_NCAS (BK1_NCAS),
        .BK1_NWE (BK1_NWE),                 .BK1_DQMH1 (BK1_DQMH1),
        .BK1_DQMH0 (BK1_DQMH0),             .BK1_DQML1 (BK1_DQML1),
        .BK1_DQML0 (BK1_DQML0),
        .BK1_DQ (BK1_DQ),

        .mDPSW (mDPSW),                     .mLED (mLED),                   //Reserved & debug pin
        .sLED (sLED),

        .USER_REG_RD (USER_REG_RD),         .USER_MEM_RD (USER_MEM_RD),     //User I/O
        .USER_FPGA_VER (USER_FPGA_VER),
        .nMAIN_RESET (nMAIN_RESET),         .USER_REG_CS (USER_REG_CS),
        .USER_MEM_CS (USER_MEM_CS),         .CPU_IO_ENB (CPU_IO_ENB),
        .CPU_WE (CPU_WE),                   .LD_DI (LD_DI),
        .LA_REG  (LA_REG),                  .LWnR_REG (LWnR_REG),

        .SDRAM_WR0(SDRAM_WR0),              .SDRAM_WR_REG0(SDRAM_WR_REG0),
        .MEM_MWD_2REG0(MEM_MWD_2REG0),

        .SDRAM_WR1(SDRAM_WR1),              .SDRAM_WR_REG1(SDRAM_WR_REG1),
        .MEM_MWD_2REG1(MEM_MWD_2REG1)

);
```

ローカルバスコントロール信号

ピン名	I/O	機能
FPGA_LCLK	I	ローカルバスクロック　66MHz固定
NLRESET	I	ローカルバスリセット信号(Low Active)
LHOLD	I	ローカルバスホールドリクエスト
LA[31:2]	I	ローカルアドレス
NLADS	I	ローカルバス　アドレスストローブ
LWNR	I	ローカルバス　ライト/リード
nLBE[3:0]	I	ローカルバス　バイトイネーブル
nLBLAST	I	ローカルバス　バーストラスト
LHOLDA	O	ローカルバス　LHOLD　ACK信号
nLBTERM	O	ローカルバス　バーストターミネート信号
nLREADY	O	ローカルバス　I/Oアクセス　レディー
NLINTI	O	ローカルバス　割込信号
NCCS	O	ローカルバス　コンフィギュレーションレジスタセレクト
NDREQ0	O	ローカルバス　DMA Demand Modeセレクト0
NDREQ1	O	ローカルバス　DMA Demand Modeセレクト1
BREQI	O	ローカルバス　バスリクエストIN信号
NBIGEND	O	ローカルバス　ビッグエンディアンセレクト
nLLOCKI	O	ローカルバス　ローカルロックIN
NWAIT	O	ローカルバス　WAIT　I/O
LD[31:0]	IO	ローカルデータ
DP[3:0]	IO	ローカルデータパリティ

※詳細は PLX 社：PEX8311 データシートを参照下さい。

カメラリンク　I/F信号

ピン名	I/O	機能
CLINK_CLK	I	カメラリンク　データピクセルクロック
CLINK_FVAL	I	カメラリンク　フレームバリッド
CLINK_DVAL	I	カメラリンク　ピクセルバリッド
CLINK_IND[31:0]	I	カメラリンク　入力データ
SERTFG	I	カメラリンク　シリアル入力
SERTC	O	カメラリンク　シリアル出力

SDRAM Bank A信号

ピン名	I/O	機能
SDCLK_FPGA	I	SDRAM_A　動作クロック　80MHz
BK0_NCS	O	SDRAM_A チップセレクト
BK0_CKE	O	SDRAM_A クロックイネーブル
BK0_BA[1:0]	O	SDRAM_A　バンクアドレス
BK0_A[12:0]	O	SDRAM_A　アドレス
BK0_NRAS	O	SDRAM_A　Rowアドレスストローブ
BK0_NCAS	O	SDRAM_A　Columnアドレスストローブ
BK0_NWE	O	SDRAM_A　ライトイネーブル
BK0_DQMH1	O	SDRAM_A上位バイトマスクイネーブル1
BK0_DQMH0	O	SDRAM_A 上位バイトマスクイネーブル0
BK0_DQML1	O	SDRAM_A 下位バイトマスクイネーブル1
BK0_DQML0	O	SDRAM_A 下位バイトマスクイネーブル0
BK0_DQ[31:0]	IO	SDRAM_A　データバス

※詳細は Micron 社：MT48LC32M16A2P データシートを参照下さい

SDRAM Bank B信号

ピン名	I/O	機能
BK1_NCS	O	SDRAM_B チップセレクト
BK1_CKE	O	SDRAM_B クロックイネーブル
BK1_BA[1:0]	O	SDRAM_B バンクアドレス
BK1_A[12:0]	O	SDRAM_B アドレス
BK1_NRAS	O	SDRAM_B Rowアドレスストローブ
BK1_NCAS	O	SDRAM_B Columnアドレスストローブ
BK1_NWE	O	SDRAM_B ライトイネーブル
BK1_DQMH1	O	SDRAM_B 上位バイトマスクイネーブル1
BK1_DQMH0	O	SDRAM_B 上位バイトマスクイネーブル0
BK1_DQML1	O	SDRAM_B 下位バイトマスクイネーブル1
BK1_DQML0	O	SDRAM_B 下位バイトマスクイネーブル0
BK1_DQ[31:0]	IO	SDRAM_B データバス

※詳細は Micron 社：MT48LC32M16A2P データシートを参照下さい

LED / DPSW信号

ピン名	I/O	機能
mDPSW[3:0]	I	KIT1140 DPSW
mLED[3:0]	O	KIT1140 LED
sLED[3:0]	O	KIT2010 LED

FPGA内部ユーザー レジスタ/メモリ設定用信号

ピン名	I/O	機能
USER_REG_RD[31:0]	I	ユーザーレジスタリードデータ
USER_MEM_RD[31:0]	I	ユーザーメモリリードデータ
USER_FPGA_VER[15:0]	I	ユーザー設定FPGAバージョン
nMAIN_RESET	O	メインリセット (NLRESET or Software Reset)
USER_REG_CS	O	ユーザーレジスタチップセレクト
USER_MEM_CS	O	ユーザーメモリチップセレクト
CPU_IO_ENB	O	ローカルバス I/Oアクセルイネーブル
CPU_WE	O	ローカルバス ライトイネーブル(FPGA_LCLK1パルス幅)
LD_DI[31:0]	O	ローカルバス FPGA内部LD信号
LA_REG[16:2]	O	ローカルバス FPGA内部LA信号
LWnR_REG	O	ローカルバス FPGA内部LWNR信号

2.5.CL Getter IP 性能について

以下に CL Getter IP の性能について示します。

- CL Getter IP 性能

1. エリアセンサカメラ対応
2. 最大データ幅　32bit
3. 最大ピクセルレート　80MHz　　**※左記以上のレートの場合はお問い合わせ下さい。**
4. 4 つのキャプチャーモード機能(データ幅 8bit/16bit/24bit/32bit)
5. 最大ライン画素　2048 画素
6. カメラシリアル通信機能(デフォルト 9600baud　変更可)
7. ユーザー使用可能メモリ　最大 32Kx32bit (FPGA 内部 BRAM　出荷時　512x32bit)
8. PCI Express x1 (Rev1.0a 準拠)　転送レート　(100MB/sec 以上)　**※PC の使用環境により異なります。**

2.6.CL Getter IP で接続する信号について

下記にトップモジュール上の camera_link_ip モジュールのポート名を示します。赤字で示している信号とユーザー回路からの信号を接続する事により、画像のキャプチャーを行う事が出来ます。また、青字で示している信号は FPGA 内部ユーザーレジスタ、メモリ制御用信号となります。必要に応じて御使用下さい。

その他の信号につきましては変更しないで下さい。変更された場合、正しくキャプチャー出来なくなったり、基板に損傷を与える可能性があります。

「user.v」から出力される信号 USER_FVAL/USER_DVAL/USER_IND[31:0]を「camera_link_ip_0」モジュールへ接続して下さい。

```
(* box_type = "user_black_box" *)
camera_link_ip  camera_link_ip_0 (
        .FPGA_LCLK (FPGA_LCLK_I),           .NLRESET (NLRESET),          //PLX
        .LHOLD (LHOLD),                     .LA (LA),
        .NLADS (NLADS),                     .LWNR (LWNR),
        .nLBE (nLBE),                       .nLBLAST (nLBLAST),
        .LHOLDA (LHOLDA),                   .nLBTERM (nLBTERM),
        .nLREADY (nLREADY),                 .NLINTI (NLINTI),
        .NCCS (NCCS),                       .NDREQ0 (NDREQ0),
        .NDREQ1 (NDREQ1),                   .BREQI (BREQI),
        .NBIGEND (NBIGEND),                 .nLLOCKI (nLLOCKI),
        .NWAIT (NWAIT),
        .LD (LD),                           .DP (DP),

        .CLINK_CLK (CLINK_CLK),             .CLINK_FVAL (USER_FVAL),     //Camera Link I/F
        .CLINK_DVAL (USER_DVAL),            .CLINK_IND (USER_IND[31:0]),
        .SERTFG (SERTFG),
        .SERTC (SERTC),
                                    ,
                                    ,
        .USER_REG_RD (USER_REG_RD),         .USER_MEM_RD (USER_MEM_RD),  //User I/O
        .USER_FPGA_VER (USER_FPGA_VER),
        .nMAIN_RESET (nMAIN_RESET),         .USER_REG_CS (USER_REG_CS),
        .USER_MEM_CS (USER_MEM_CS),         .CPU_IO_ENB (CPU_IO_ENB),
        .CPU_WE (CPU_WE),                   .LD_DI (LD_DI),
        .LA_REG  (LA_REG),                  .LWnR_REG (LWnR_REG),
                                    ,
                                    ,
);

//+++ (5) end +++//
```

2.7. 接続信号仕様、制約について

ユーザー回路からカメラリンク IP へ接続される信号について以下にタイムチャート例を示します。IP へ接続する信号に付きまして、制約事項がありますので御注意下さい。

制約事項が満たせない場合、画像が正確にキャプチャー出来ません。

カメラからの画像データをキャプチャーする場合もユーザー回路にて画像処理を行う場合も IP へ渡す信号制約は変わりません。

CL Getter IP はエリアセンサカメラのキャプチャーを前提としておりますので、FVAL/DVAL/DATA をカメラ仕様と同等の信号の入力をして下さい。

[制約事項]

① USER_FVAL 信号は 1 フレーム中に 1 ハイアクティブとして下さい。

② USER_DVAL 信号は 1 ライン中に 1 ハイアクティブとして下さい。

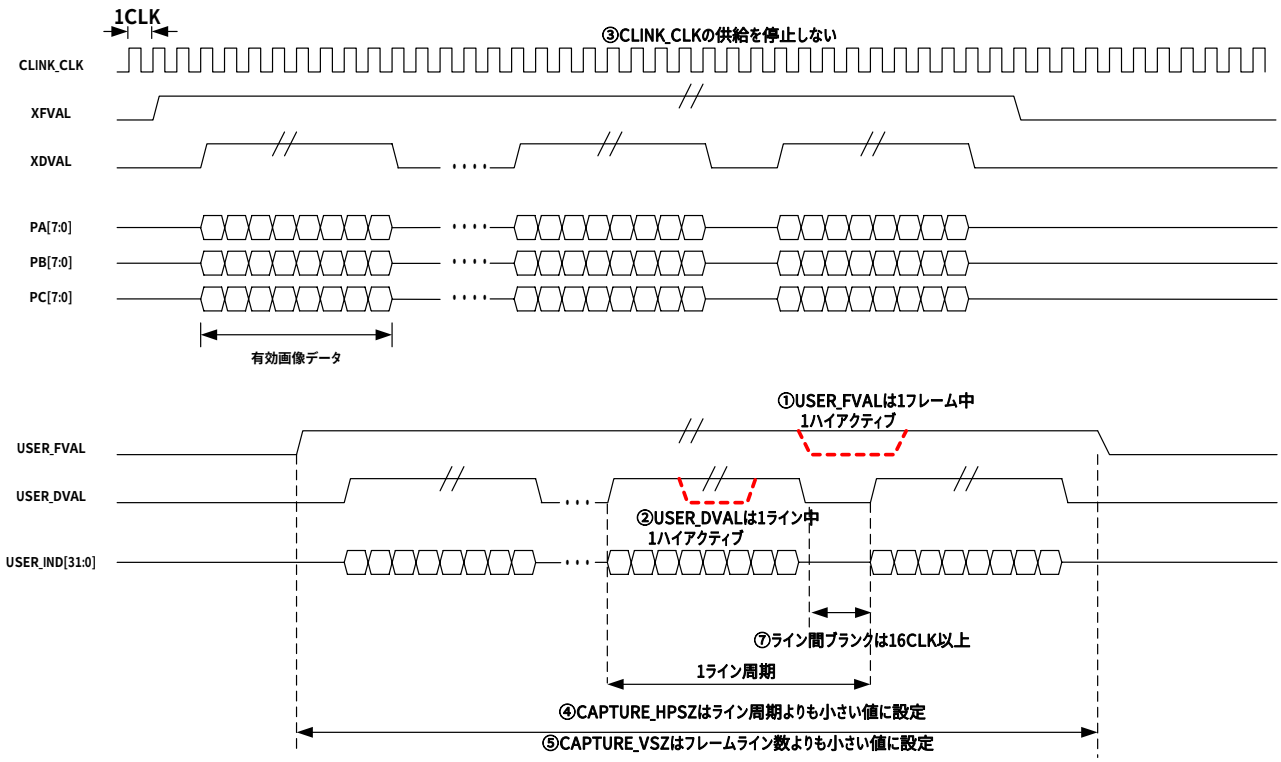
③ CLINK_CLK(カメラからの供給クロック)は必ず接続して下さい。また、途中で供給を停止しないで下さい。

④ CAPTURE_HPSZ は 1 ライン周期以下として下さい。

⑤ CAPTURE_VSZ は 1 フレームのライン数以下として下さい。

⑥ カメラからの XFVAL、XDVAL はハイアクティブの信号にして下さい。ロウアクティブの場合は入力時に論理反転して下さい。

⑦ ライン間ブランクはピクセルクロック(CLINK_CLK)65MHz以下時 16CLK 以上、80MHz時かつ 2048 画素/32bit 時は 80CLK 以上として下さい。



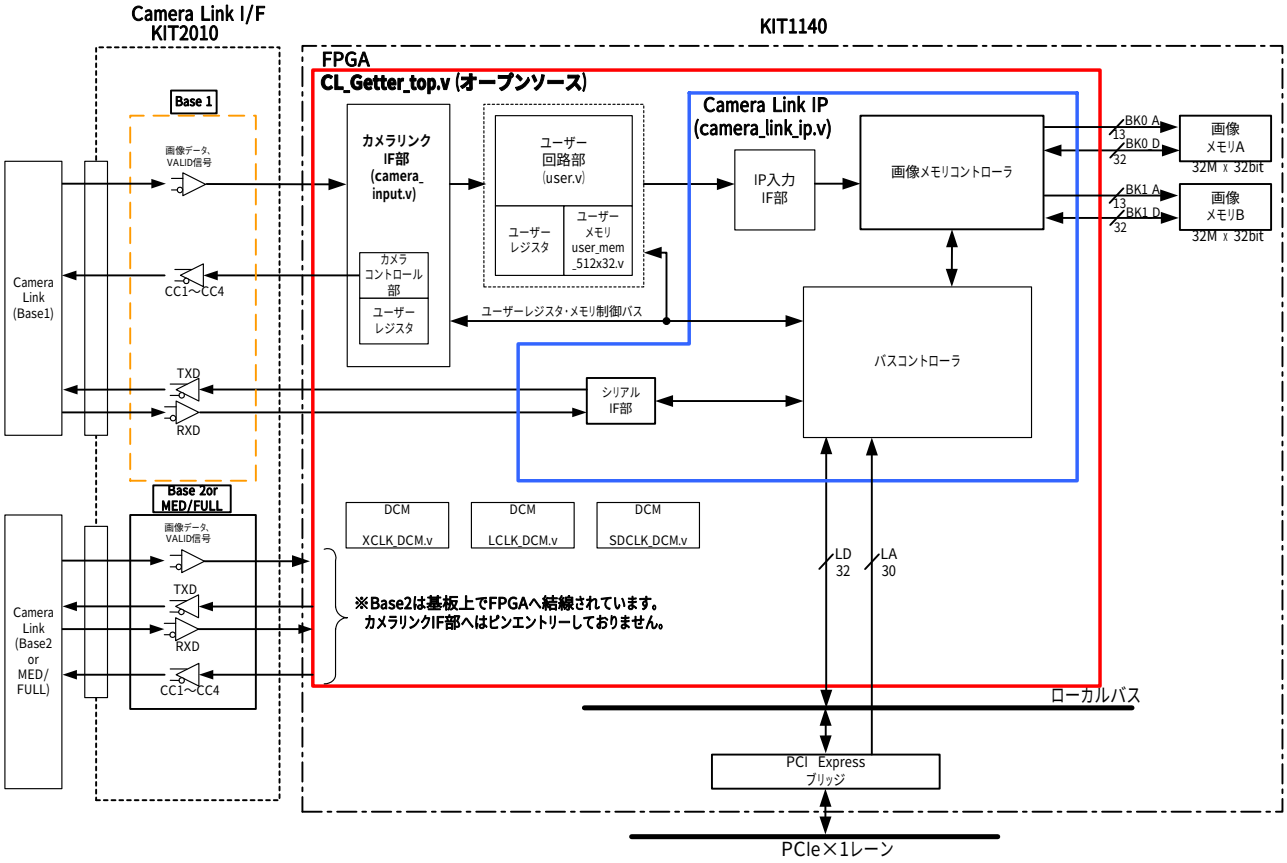
3.CL Getter について

3.1.CL Getter 基本構成

CL Getterの FPGA 内部は基本的にユーザー回路部(ユーザーによる変更可能な回路：赤枠部分)と Camera Link IP 部(変更出来ない回路：青枠部分)とで構成されています。

ユーザー回路へ画像処理回路を埋め込む事で画像処理基板として付加機能を有する事も可能となります。

付属 CD 内の FPGA プロジェクトは CL Getter IP を使用したサンプルです。



module tree

CL_Getter_top (オープンソース)

|----- XCLK_DCM.v
|----- LCLK_DCM.v
|----- SDCLK_DCM.v
|----- camera_input.v(オープンソース)
|----- user.v(オープンソース)
|----- camera_link_ip.v(IPモジュール：ブラックボックス)
|----- user_mem_512x32.v

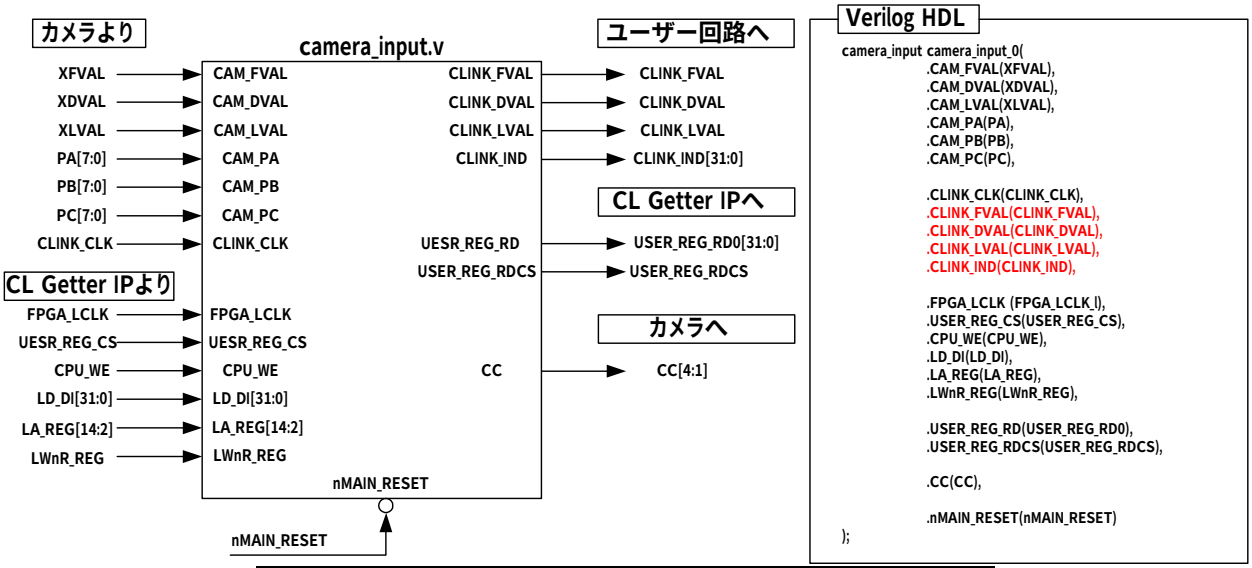
3.2.入力 I/F モジュールについて

入力 I/F モジュールは「camera_input.v」としてトップモジュールへエントリされています。

このモジュールはカメラから入力されます画素データの並べ替えを行います。

Verilog HDL コードの赤文字をユーザー回路モジュールへ接続します。

カメラ入力信号のサンプルについては付属 CD 内テストベンチファイルを参照下さい。



信号名	I/O	備考
nMAIN_RESET	I	Active Lowの非同期リセット信号 FPGA全体のハードウェアリセット及びソフトウェアリセット
CLINK_CLK	I	カメラリンクからのピクセルクロック
CAM_FVAL	I	カメラリンクからのFrame Valid信号
CAM_DVAL	I	カメラリンクからのData Valid信号
CAM_LVAL	I	カメラリンクからのLine Valid信号
CAM_PA[7:0]	I	カメラリンクからのPort Aデータ※
CAM_PB[7:0]	I	カメラリンクからのPort Bデータ※
CAM_PC[7:0]	I	カメラリンクからのPort Cデータ※
FPGA_LCLK	I	ローカルバスクロック
USER_REG_CS	I	ユーザーレジスタチップセレクト
CPU_WE	I	ローカルバス　ライトイネーブル(FPGA_LCLK1パルス幅)
LD_DI[31:0]	I	ローカルバス　FPGA内部LD信号
LA_REG[14:2]	I	ローカルバス　FPGA内部LA信号
LWnR_REG	I	ローカルバス　FPGA内部LWnR信号

※カメラリンクのPort情報に付きましてはCamera Link規格を参照して下さい。

信号名	I/O	備考
CLINK_FVAL	O	camera_input.vモジュールからのFrame Valid信号出力
CLINK_DVAL	O	camera_input.vモジュールからのData Valid信号出力
CLINK_LVAL	O	camera_input.vモジュールからのLine Valid信号出力
CLINK_IND[31:0]	O	camera_input.vモジュールからのDataバス出力
USER_REG_RD[31:0]	O	ユーザーレジスタリードデータ
USER_REG_RDCS	O	camera_inputモジュールユーザーレジスタチップセレクト
CC[4:1]	O	カメラコントロール1～4(オプション)

camera_input.v サンプルの Verilog HDL ソースではカメラリンクの各タップデータ CAM_PA/CAM_PB/CAM_PC を 32bit データとしてビットの並びを設定しています。
サンプルではカメラリンクからの PortA：R(赤)、PortB：G(緑)、PortC：B(青)が入力される場合の例となります。
使用されますカメラによりデータビットの並び替えが必要な場合がありますが、その際にこの部分を変更する事で対応する事が可能です。

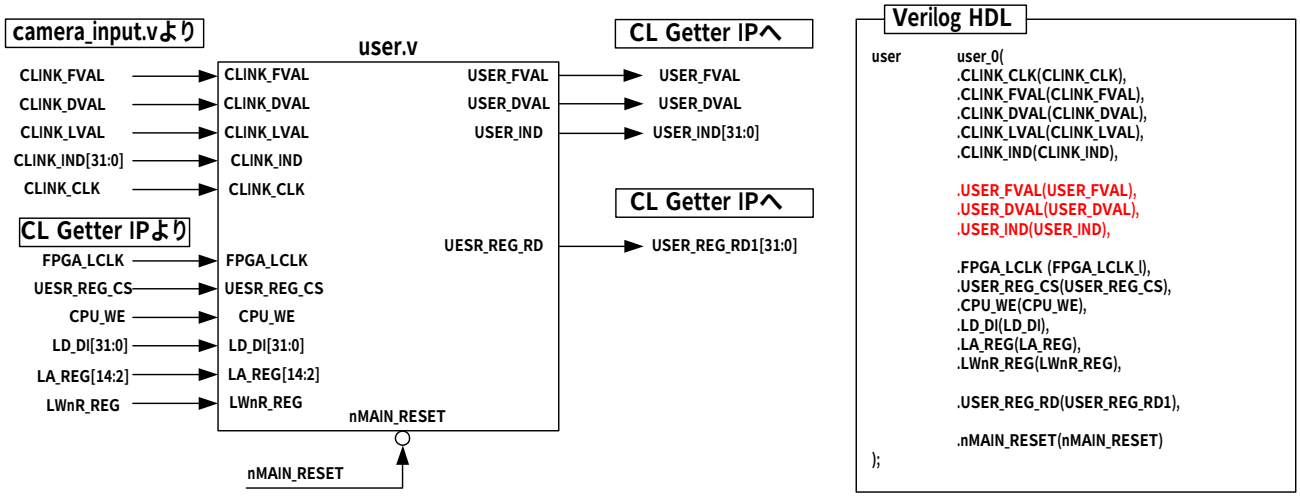
```
// Synchronized //
always@ (posedge CLINK_CLK or negedge nMAIN_RESET)
begin
        if (~nMAIN_RESET)
                begin
                        CLINK_FVAL  <=  1'b0;
                        CLINK_DVAL  <=  1'b0;
                        CLINK_LVAL  <=  1'b0;
                        CLINK_IND   <= 32'h00000000;
                end

        else
                begin
                        CLINK_FVAL      <=  CAM_FVAL;
                        CLINK_DVAL      <=  CAM_DVAL;
                        CLINK_LVAL      <=  CAM_LVAL;
                        CLINK_IND[31:24] <=  8'h00;
                        CLINK_IND[23:16] <=  CAM_PA[7:0];
                        CLINK_IND[15:8]  <=  CAM_PB[7:0];
                        CLINK_IND[7:0]   <=  CAM_PC[7:0];
                end
end
```

3.3.ユーザー回路について

ユーザー回路モジュールは「user.v」としてトップモジュールへエントリされています。

このモジュールはユーザーにてオリジナル画像処理機能を盛り込めます。

Verilog HDL コードの赤文字を IP モジュールへ接続します。



信号名	I/O	備考
nMAIN_RESET	I	Active Lowの非同期リセット信号 FPGA全体のハードウェアリセット及びソフトウェアリセット
CLINK_CLK	I	カメラリンクからのピクセルクロック
CLINK_FVAL	I	camera_input.vモジュールからのFrame Valid信号出力
CLINK_DVAL	I	camera_input.vモジュールからのData Valid信号出力
CLINK_LVAL	I	camera_input.vモジュールからのLine Valid信号出力
CLINK_IND[31:0]	I	camera_input.vモジュールからのDataバス出力
FPGA_LCLK	I	ローカルバスクロック
USER_REG_CS	I	ユーザーレジスタチップセレクト
CPU_WE	I	ローカルバス　ライトイネーブル(FPGA_LCLK1パルス幅)
LD_DI[31:0]	I	ローカルバス　FPGA内部LD信号
LA_REG[14:2]	I	ローカルバス　FPGA内部LA信号
LWnR_REG	I	ローカルバス　FPGA内部LWnR信号

※カメラリンクのPort情報に付きましてはCamera Link規格を参照して下さい。

信号名	I/O	備考
USER_FVAL	O	user.vモジュールからのFrame Valid信号出力
UESR_DVAL	O	user.vモジュールからのData Valid信号出力
USER_IND[31:0]	O	user.vモジュールからのDataバス出力
USER_REG_RD[31:0]	O	ユーザーレジスタリードデータ

user.v サンプルの Verilog HDL ソースでは CLINK_FVAL/CLINK_DVAL/CLINK_IND をフリップフロップを 1 段介して出力しています。

この部分を変更する事によりユーザーオリジナル画像処理回路を盛り込めます。

```
// Synchronized //
always@ (posedge CLINK_CLK or negedge nMAIN_RESET)
begin
        if (~nMAIN_RESET)
                begin
                        USER_FVAL  <=  1'b0;
                        USER_DVAL  <=  1'b0;
                        USER_IND   <= 32'h00000000;
                end

        else
                begin
                        USER_FVAL      <=  CLINK_FVAL;
                        USER_DVAL      <=  CLINK_DVAL;
                        USER_IND[31:0] <=  CLINK_IND[31:0];
                end
end
```

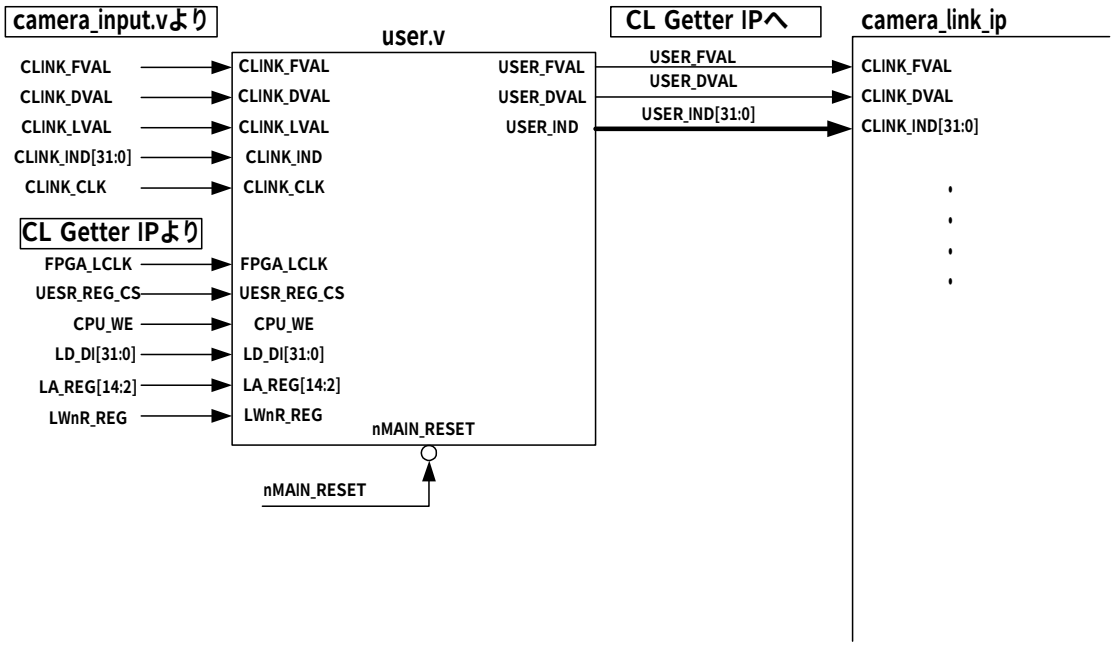
3.4.ユーザー回路と CL Getter IP の接続について

3.3.項で示しました「user.v」から出力される信号を「camera_link_ip_0」モジュールへ接続して下さい。

2.7.項の制約を満たした信号を入力した場合、その後はユーザーが意識する事無く、外付け SDRAM へデータが格納され、PC へ画像データを転送する事が出来ます。

SDRAM へのデータ格納状況や PC へのデータ転送につきましてはシミュレーションモデルにて確認頂けます。

```
//+++ (5)  camera_link_ip module interface  +++//

(* box_type = "user_black_box" *)
camera_link_ip  camera_link_ip_0 (
        .FPGA_LCLK (FPGA_LCLK_I),          .NLRESET (NLRESET),          //PLX
        .LHOLD (LHOLD),                    .LA (LA),
        .NLADS (NLADS),                    .LWNR (LWNR),
        .nLBE (nLBE),                      .nLBLAST (nLBLAST),
        .LHOLDA (LHOLDA),                  .nLBTERM (nLBTERM),
        .nLREADY (nLREADY),                .NLINTI (NLINTI),
        .NCCS (NCCS),                      .NDREQ0 (NDREQ0),
        .NDREQ1 (NDREQ1),                  .BREQI (BREQI),
        .NBIGEND (NBIGEND),                .nLLOCKI (nLLOCKI),
        .NWAIT (NWAIT),
        .LD (LD),                          .DP (DP),

        .CLINK_CLK (CLINK_CLK),            .USER_FVAL (USER_FVAL),      //Camera Link I/F
        .USER_DVAL (USER_DVAL),            .USER_IND (USER_IND[31:0]),
        .SERTFG (SERTFG),
        .SERTC (SERTC),
                                    .
                                    .
                                    .
                                    .
);

//+++  (5)  end  +++//
```



※ CLINK_IND[31：0]/USER_IND [31:0] は wire 宣言でバス設定しないと 1bit のデータ線として接続されますので注意して下さい。その他の多ビット信号は wire 宣言をして下さい。

3.5.ユーザーレジスタ、ユーザーメモリ設計例

camera_input.v 及び user.v のモジュールではローカルバスよりユーザーが設定できるレジスタ、メモリが用意されています。
camera_iunput.v では CC1～CC4 をエントリしており、カメラの外部同期用信号生成パラメータを設定する事も可能です。
また、user.v では画像処理に関するパラメータ設定、制御信号、その他信号を PC より設定可能です。
Verilog HDL 上にサンプル回路を記述しておりますので、参考程度にお使い下さい。

3.5.1.ユーザーレジスタ記述例

トップモジュールにユーザーが設定出来るレジスタの記述例を示しています。
ローカルバスからの Write/Read についてアプリケーションソフトよりアクセスする事が容易となります。ローカルバスコントロールは IP 内部で行っています。

(1) ローカルバス ユーザーレジスタ Write 制御部分

下記に camera_input.v モジュール内のユーザーレジスタ Write 制御部分を示します。
ローカルアドレス 8000hex～800Chex に Write アクセスが来た場合に FPGA 内部レジスタへデータを格納します。

```
// set to user control register //
always@ (posedge FPGA_LCLK or negedge nMAIN_RESET)
begin
        if (~nMAIN_RESET)
                begin
                        USER_REG0_D[31:0]   <=  32'h00000000;
                        USER_REG1_D[31:0]   <=  32'h00000000;
                        USER_REG2_D[31:0]   <=  32'h00000000;
                        USER_REG3_D[31:0]   <=  32'h00000000;
                        end

        else if (USER_REG_CS & CPU_WE & (LA_REG[14:2] == 13'h000))     // User Reg. Address ; 0000 8000 hex ; user reg0
                USER_REG0_D[31:0]   <=  LD_DI[31:0];

        else if (USER_REG_CS & CPU_WE & (LA_REG[14:2] == 13'h001))     // User Reg. Address ; 0000 8004 hex ; user reg1
                USER_REG1_D[31:0]   <=  LD_DI[31:0];

        else if (USER_REG_CS & CPU_WE & (LA_REG[14:2] == 13'h002))     // User Reg. Address ; 0000 8008 hex ; user reg2
                USER_REG2_D[31:0]   <=  LD_DI[31:0];

        else if (USER_REG_CS & CPU_WE & (LA_REG[14:2] == 13'h003))     // User Reg. Address ; 0000 800c hex ; user reg3
                USER_REG3_D[31:0]   <=  LD_DI[31:0];

        else
                begin
                        USER_REG0_D[31:0]   <=  USER_REG0_D[31:0];
                        USER_REG1_D[31:0]   <=  USER_REG1_D[31:0];
                        USER_REG2_D[31:0]   <=  USER_REG2_D[31:0];
                        USER_REG3_D[31:0]   <=  USER_REG3_D[31:0];
                end
end
```

(2) ローカルバス ユーザーレジスタ Read 制御部分

下記に camera_input.v モジュール内のユーザーレジスタ Read 制御部分を示します。

ローカルアドレス 8000hex～800Chex に Read アクセスが来た場合に FPGA 内部レジスタへ格納されているデータを CL_Getter IP へ転送します。

```
// CPU Read from user control register //
always@ (posedge FPGA_LCLK or negedge nMAIN_RESET)
begin
        if (~nMAIN_RESET)
                USER_REG_RD[31:0]  <=  32'h00000000;

        else if (USER_REG_CS & ~LWnR_REG & (LA_REG[14:2] == 13'h000)) // User Reg. Address ; 0000 8000 hex ; user reg0
                USER_REG_RD[31:0]  <=  USER_REG0_D[31:0];

        else if (USER_REG_CS & ~LWnR_REG & (LA_REG[14:2] == 13'h001)) // User Reg. Address ; 0000 8004 hex ; user reg1
                USER_REG_RD[31:0]  <=  USER_REG1_D[31:0];

        else if (USER_REG_CS & ~LWnR_REG & (LA_REG[14:2] == 13'h002)) // User Reg. Address ; 0000 8008 hex ; user reg2
                USER_REG_RD[31:0]  <=  USER_REG2_D[31:0];

        else if (USER_REG_CS & ~LWnR_REG & (LA_REG[14:2] == 13'h003)) // User Reg. Address ; 0000 800c hex ; user reg3
                USER_REG_RD[31:0]  <=  USER_REG3_D[31:0];
        else
                USER_REG_RD[31:0]  <=  32'h00000000;
end
```

3.5.2.ユーザーメモリ記述例

トップモジュールにユーザーが使用出来るメモリ(FPGA の BRAM)の記述例を示しています。

ローカルバスからの Write/Read についてアプリケーションソフトよりアクセスする事が容易となります。ローカルバスコントロールはレジスタ同様、IP 内部で行っています。

```
// user internal memory control //

assign USER_MEM_WE = USER_MEM_CS & (LA_REG[16:11] == 6'h00) & CPU_WE;

user_mem_512x32 user_mem_512x32_0 (
        .clka (FPGA_LCLK),              .dina (LD_DI),                  .addra (LA_REG[10:2]),
        .wea (USER_MEM_WE),             .douta (USER_MEM_RD)
);
```

3.6.コンパイルについて

ここでは Xilinx 社 ISE Foundation の基本的な XST 及び Implement のオプション設定について述べます。オプション設定によりタイミング制約を満たす事が出来る様になる場合もあります。

また、CL_Getter IP はトップでエントリーされていますが、Verilog HDL のソースとしては公開しておりません。

IP ソースとしましては NGC ファイル(camera_link_ip.ngc)として提供致しますので、論理合成時にプロジェクトディレクトリ内に保存して下さい。同様に camera_link_ip.ngc 内で使用してます camera_link_lfifo_2kx32.ngc/fifo_2kx8.ngc/plx_dp_ram_2kx32.ngc もプロジェクトディレクトリ内に保存して下さい。

保存されていない場合は ISE 上でエラーとなりますので御注意下さい。

FPGA サンプルソースでの ISE の設定は参考程度にお使い下さい。変更されますとタイミングが満たせなくなる場合があります。

3.6.1.制約ファイル設定

タイミング制約として UCF ファイル(.ucf)が設定出来ます。

本基板を使用される場合は FPGA サンプルソースに付いてます「CL_Getter.ucf」を必ず御使用し、変更しないで下さい。

「CL_Getter.ucf」は Base 1ch での制約ファイル及びピン情報となっております。Base 2ch 及び Camera Link Medium /Full Configuration 時は UCF ファイルの変更が必要となります。但し、UCF ファイルの変更は CL Getter としての保証対象外となります。

FPGA_LCLK クロックのトグルレートを 66MHz Duty 比 50%とする。

SDCLK_FPGA クロックのトグルレートを 80MHz Duty 比 50%とする。

XCLK クロックのトグルレートを 80MHz Duty 比 50%とする。

下記に UCF ファイルの記述例を示します。

ISE 上で project → add source → ucf ファイル選択 → OK ボタンを押下する事で設定出来ます。

＜制約ファイル記述例 UCF ファイル タイミング制約部分＞

```
NET "FPGA_LCLK" TNM_NET = "FPGA_LCLK";
TIMESPEC TS_FPGA_LCLK = PERIOD "FPGA_LCLK" 66 MHz HIGH 50%;
NET "SDCLK_FPGA" TNM_NET = "SDCLK_FPGA";
TIMESPEC TS_SDCLK_FPGA = PERIOD "SDCLK_FPGA" 80 MHz HIGH 50%;
NET "XCLK" TNM_NET = "XCLK";
TIMESPEC TS_XCLK = PERIOD "XCLK" 80 MHz HIGH 50%;

OFFSET = IN 6.5 ns BEFORE "XCLK";
OFFSET = IN 6.5 ns BEFORE "SDCLK_FPGA";
OFFSET = IN 8.5 ns BEFORE "FPGA_LCLK";
```

3.6.2.IOFF 設定について

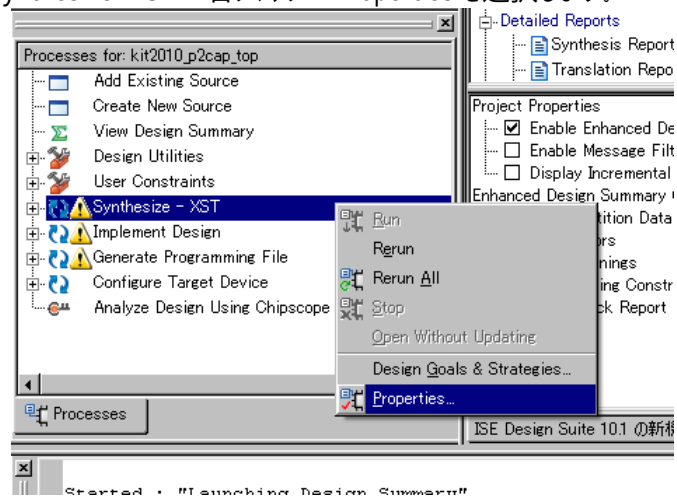
IOFF の設定については Synthesize と MAP の両方で設定出来ます。

タイミング特性(tsu,tco,Fmax)が改善されますので是非設定して下さい。

設定値に差異があった場合は Synthesize の設定が優先されますので注意して下さい。

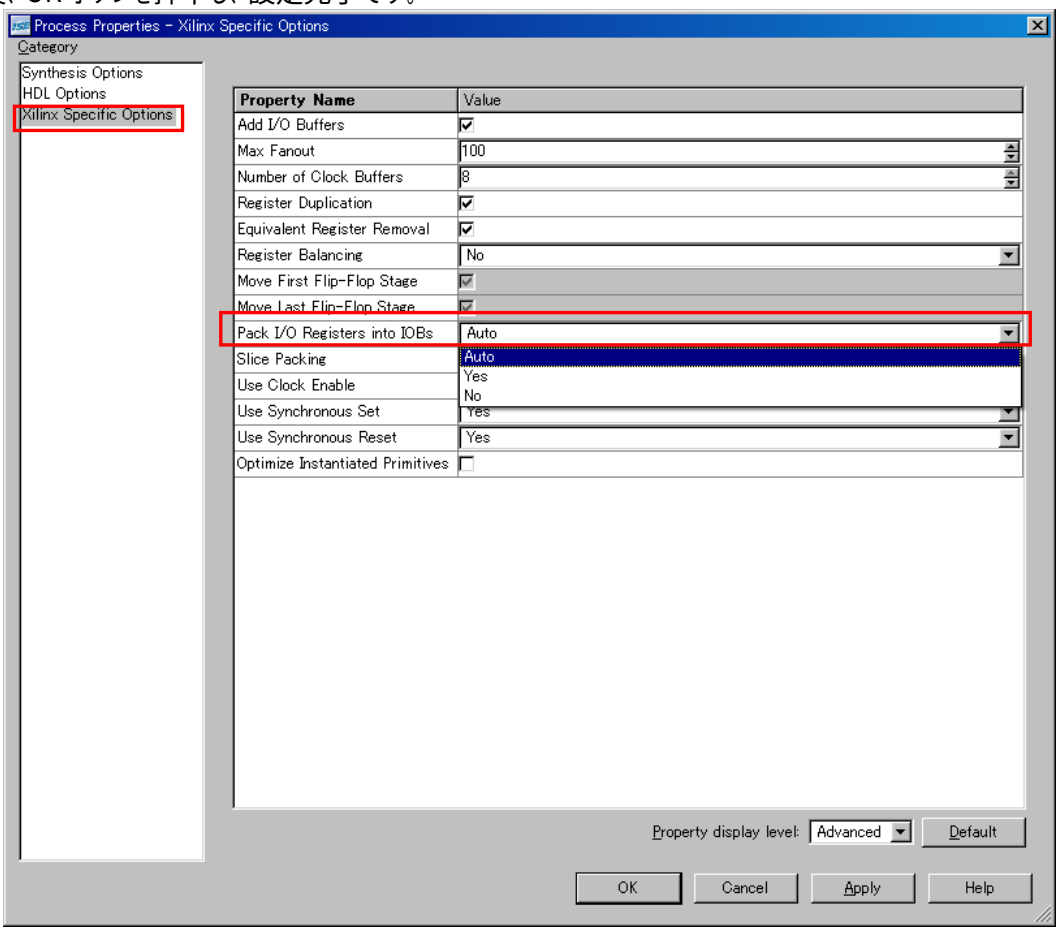
3.6.2.1.Synthesize による IOFF 設定

① ISE トップ画面で Synthesize-XST⇒右クリック⇒Properties を選択します。



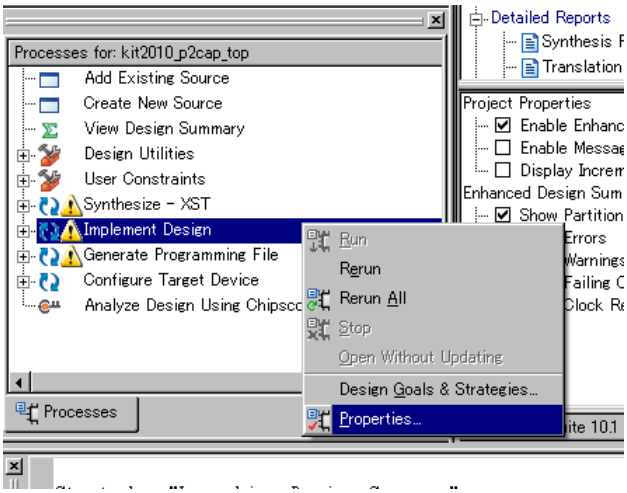
②Category を「Xlinx Specific Options」にし、「Pack I/O Registers into IOBs」を“Yes”に設定します。

設定後、OK ボタンを押下し、設定完了です。

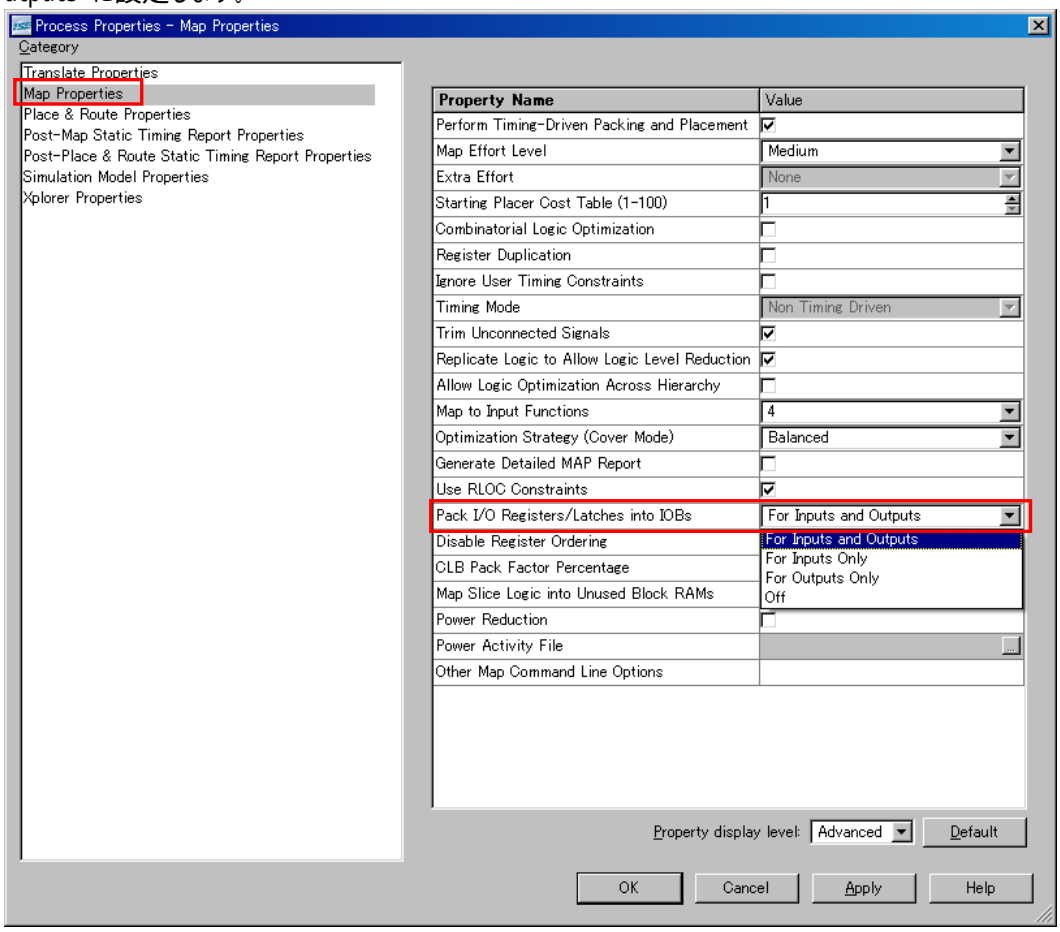


3.6.2.2.MAP による IOFF 設定

① ISE トップ画面で Implement Design⇒右クリック⇒Properties を選択します。



② Category を「Map Properties」にし、「Pack I/O Registers / Latches into IOBs」を“For Inputs and Outputs”に設定します。



設定は 100%反映される訳ではありませんので、IOB レポートにより確認して下さい。
また、入力ピン直後、出力ピン直前にフリップフロップで記述されていない信号は IOFF へマッピングされませんので御注意下さい。

4.シミュレーションについて

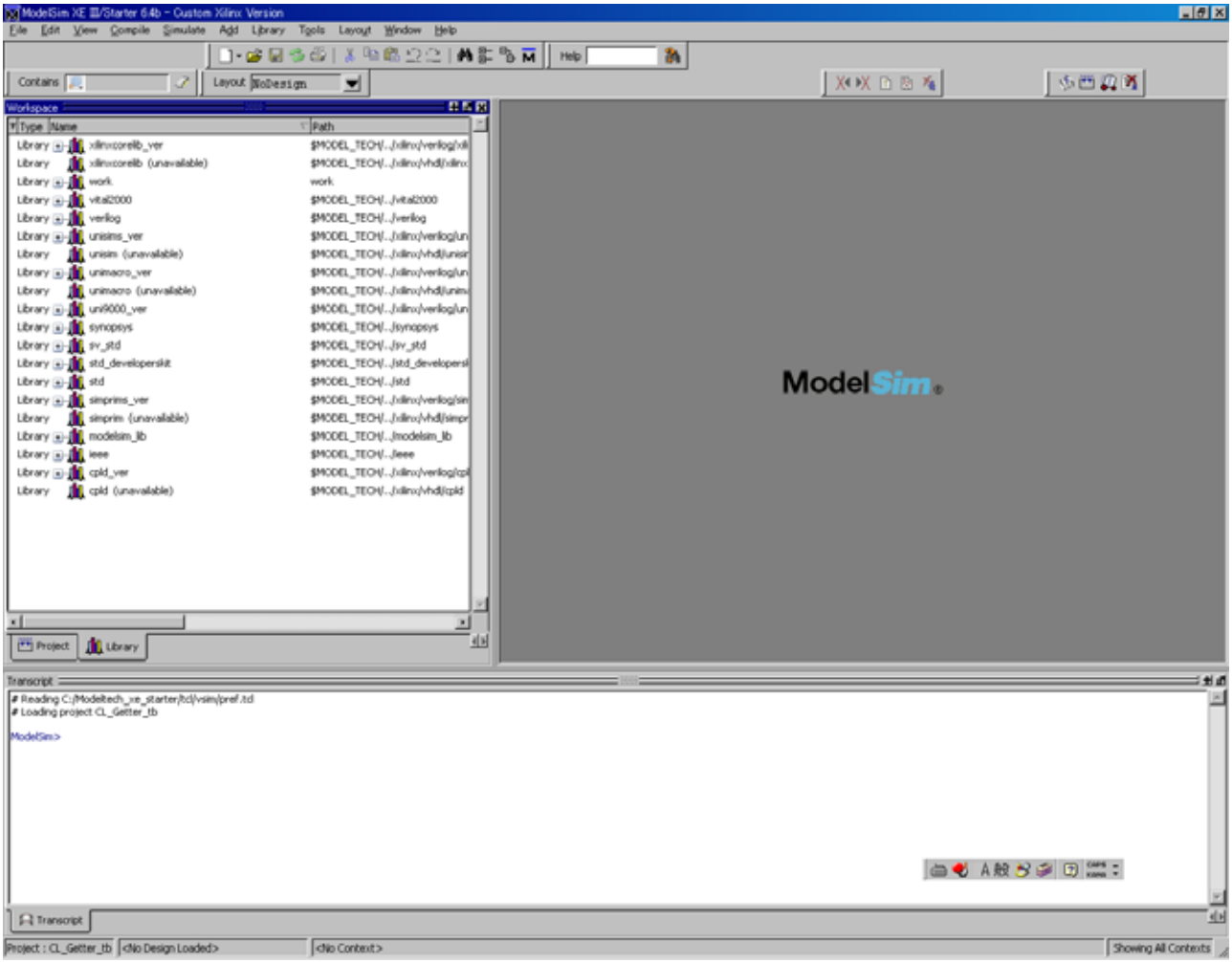
4.1.ModelSim プロジェクト作成

この章では Xilinx 社の無償 ModelSim である ModelSim XE III Starter 6.4b を使用して IP のシミュレーション方法を示します。ユーザー回路から SDRAM への格納、レジスタ設定等のシミュレーションが行えるサンプルテストベンチも付属しております。あくまで参考程度にお使い下さい。
ユーザー回路のシミュレーションでキャプチャー動作の確認の際に御使用下さい。

4.1.1.ModelSim 起動

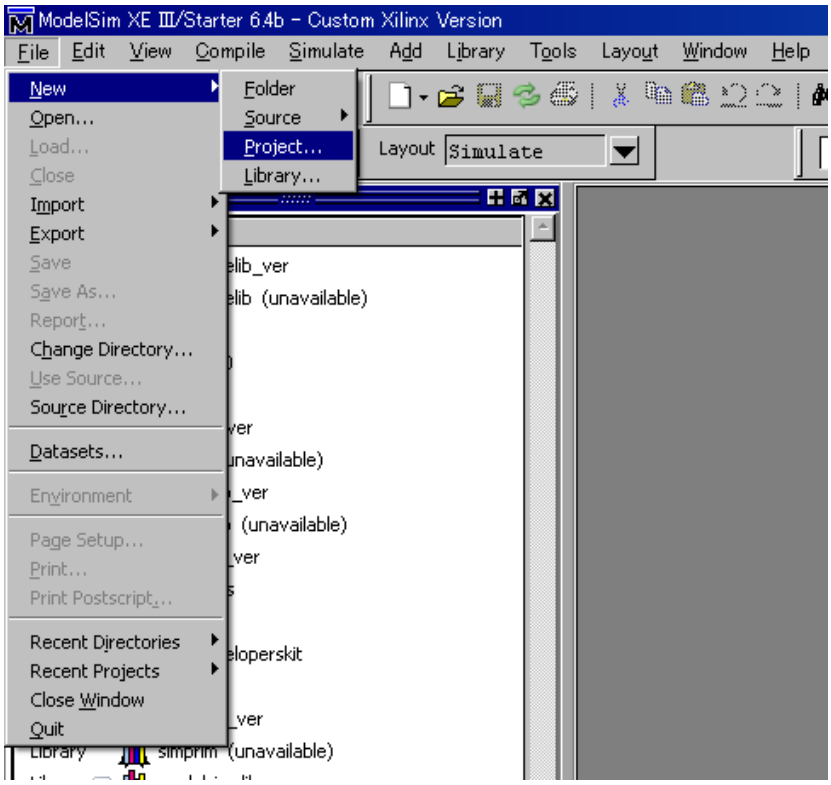
下図アイコンをダブルクリックし、ModelSim XE III Starter 6.4b を起動します。

起動すると下記ウィンドウが開きます。

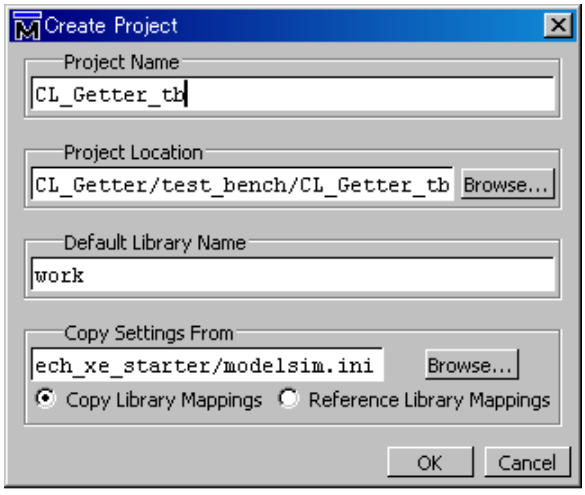


4.1.2.プロジェクト作成

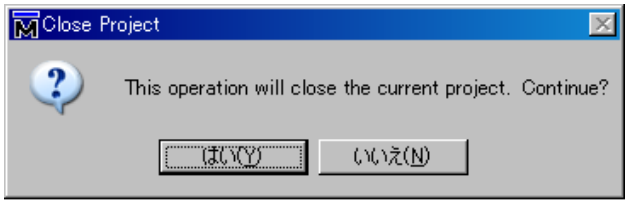
File → New → Project にてファイルを指定します。



「Project Location」にてテストベンチを作成するディレクトリを設定し、「Project Name」を記入するとフォルダが指定した所に作成されます。



前のプロジェクトが Close されていない場合、下記メッセージが表示されますが、閉じて新規プロジェクトを作成する場合は OK を押します。



【確認】

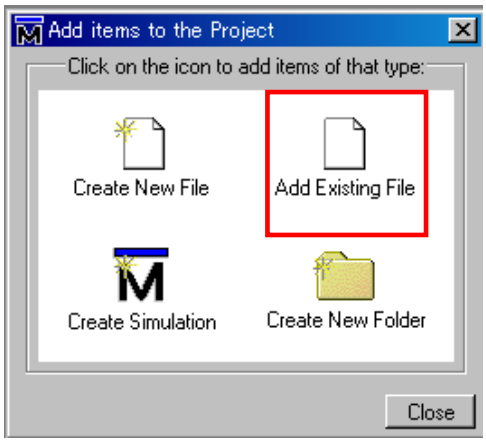
Project Name： 任意の名称をつけてください。日本語は禁止です。

Project Location：どこにプロジェクトを作成するかパスを指定します。

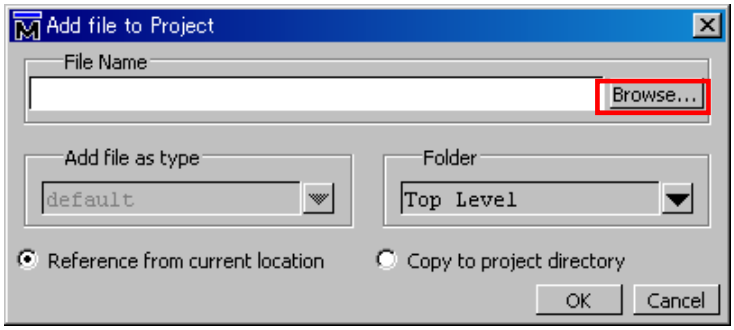
Default Library Name：変更しないでください。

これでプロジェクトが作成されます。

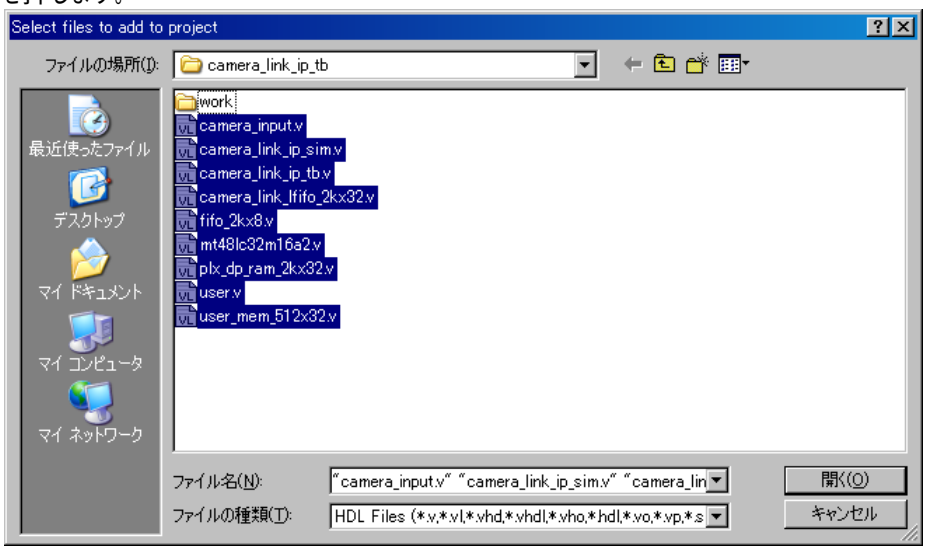
Add Existing File（下図赤枠部分）にて検証対象ファイル(今回は VerilogHDL ファイルを選択)を選択します。



「Browse. . .」ボタンを押すと次ページの画面が出ます。



検証対象となる V ファイルを全て選択します。この時、テストベンチファイルも同時に選択します。その後「開く」ボタンを押します。



「開く」ボタンを押した後、ModelSim 画面上に V ファイルが読み込まれます。

4.1.3.テストベンチ構成

ファイル構成を先に示します。

camera_link_ip_tb.v	----	camera_input_.v	カメラリンク入力 I/F モジュール
(Testbench file)	----	user.v	ユーザー回路モジュール
	----	camera_link_ip_sim.v	カメラリンク IP シミュレーションモデル
	----	user_mem_512x32.v	BRAM ラッパーファイル
	----	mt48lc32m16a2.v	SDRAM シミュレーションモデル※
	----	plx_dp_ram_2kx32.v	IP 内 BRAM ラッパーファイル
	----	camera_link_lfifo_2kx32.v	IP 内 BRAM ラッパーファイル
	----	fifo_2kx8.v	IP 内 BRAM ラッパーファイル

※シミュレーションモデルダウンロード

サンプルテストベンチプロジェクト内に SDRAM のシミュレーションモデルが含まれておりません。

本書においては、Micron 社製 SDR-SDRAM のシミュレーションモデルを用いる為、

- **mt48lc32m16a2.v**

を Micron 社ホームページよりダウンロードしておく必要があります。

本資料中で使用したモデルは 2004.5.21、Version：2.1 を使用しております。

テストベンチ作成

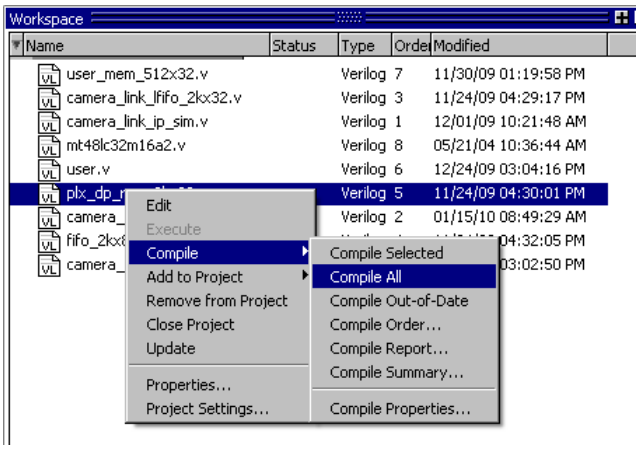
本書においては、

- **camera_link_ip_tb.v　：シミュレーション用テストベンチ**

をサンプルソースとして用いています。こちらのファイルは Verilog HDL で記述されています。

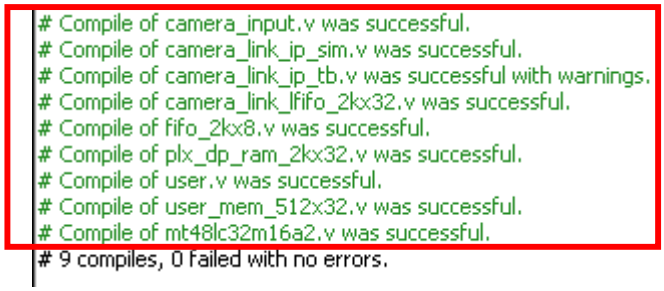
4.2.ファイルのコンパイル

ツールバー上の「Comile」→「Compile　All」を選択すると 4.1.2.項で選択した V ファイルが ModelSim 上で Compile されます。



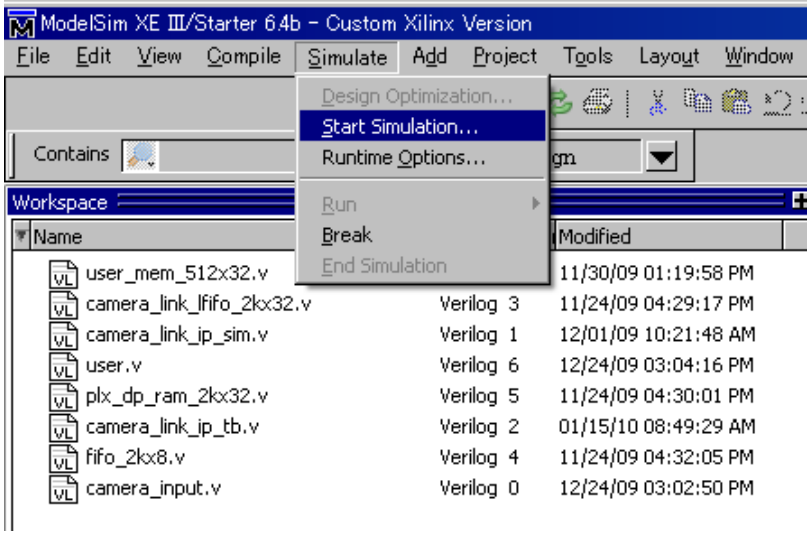
コンパイルが正常に完了した場合、赤枠で囲んだ部分の様に「＃Compile of ○○.v was successful.」と表示されます。

コンパイルエラーが表示されましたら V ファイルを修正して下さい。



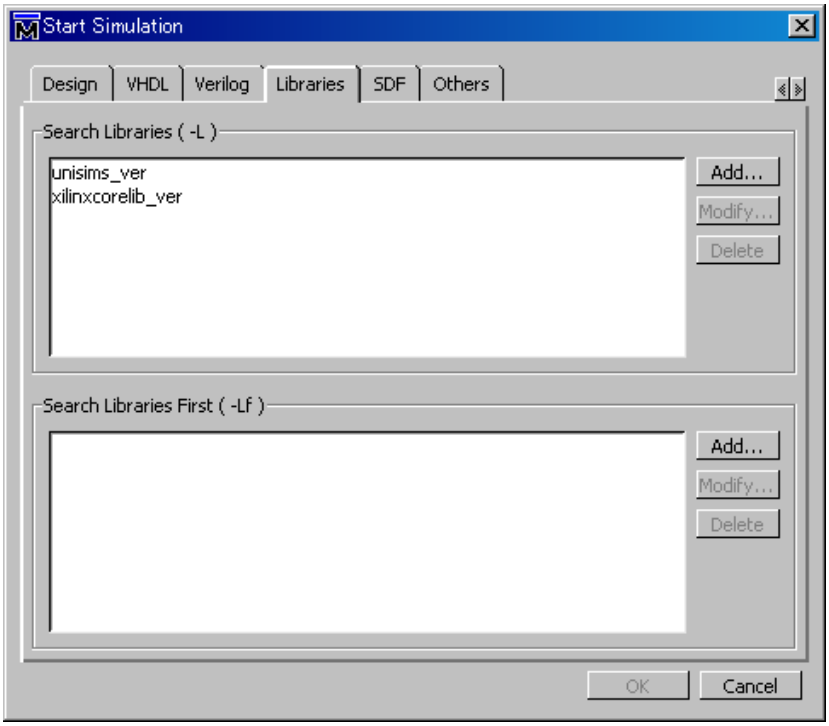
4.3.シミュレーションの実行

コンパイル完了後、ツールバー上の「Simulation」→「Simulate…」を押します。



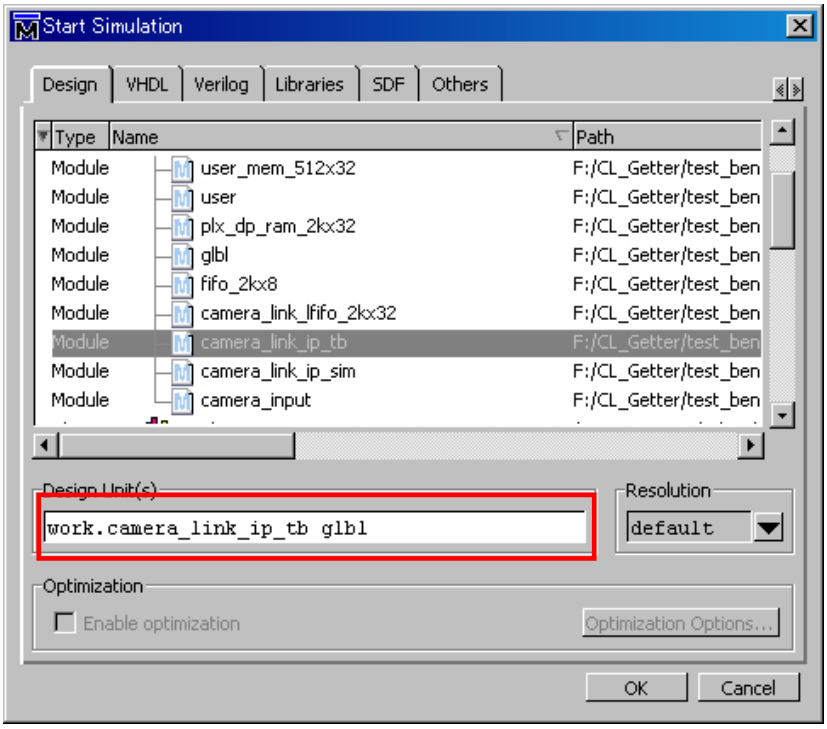
下記ウィンドウが開きましたら、「Libraries」タグに設定し、「unisim_ver」、「xilinxcorelib_ver」を選択し、Add ボタンを押します。

unisims と unisims_ver、xilinxcorelib と xilinxcorelib_ver ありますが、Verilog HDL の場合は○○_ver のライブラリを選択して下さい。

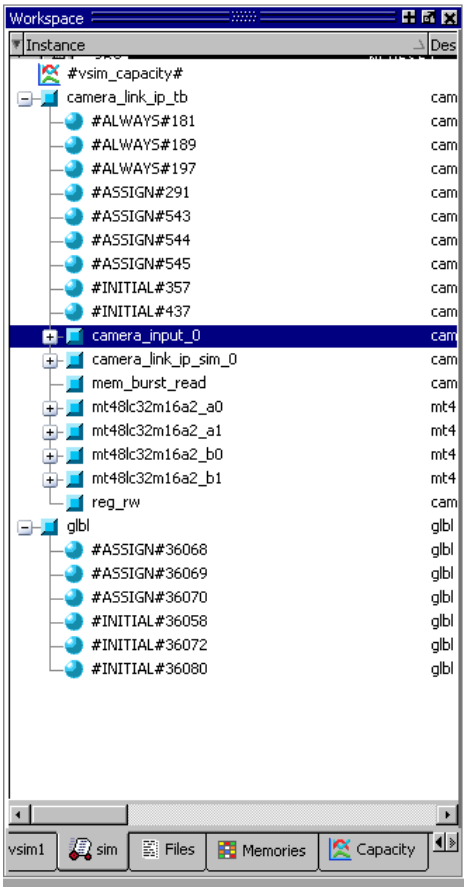


読み出すライブラリファイルを設定後、「Design」タグに設定し、テストベンチファイルを選択します。選択した後、スペースを入れ、「glbl」を入力してください。(下図赤枠部分)

今回「work」フォルダ内に入ってるのでフォルダを開き、「camera_link_ip_tb」を選択しています。ファイル選択後、OK ボタンを押します。(拡張子は省略してあります)

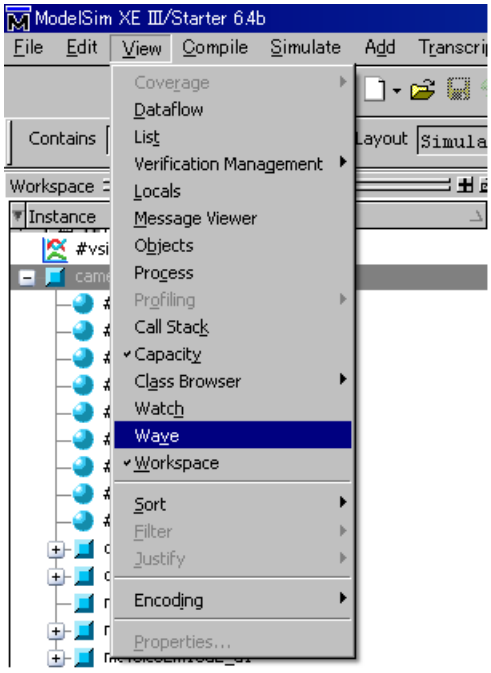


「camera_link_ip_tb」を OK しますと下図ウィンドウになります。今回のシミュレーションで見る事の出来るモジュール名(信号名)となります。



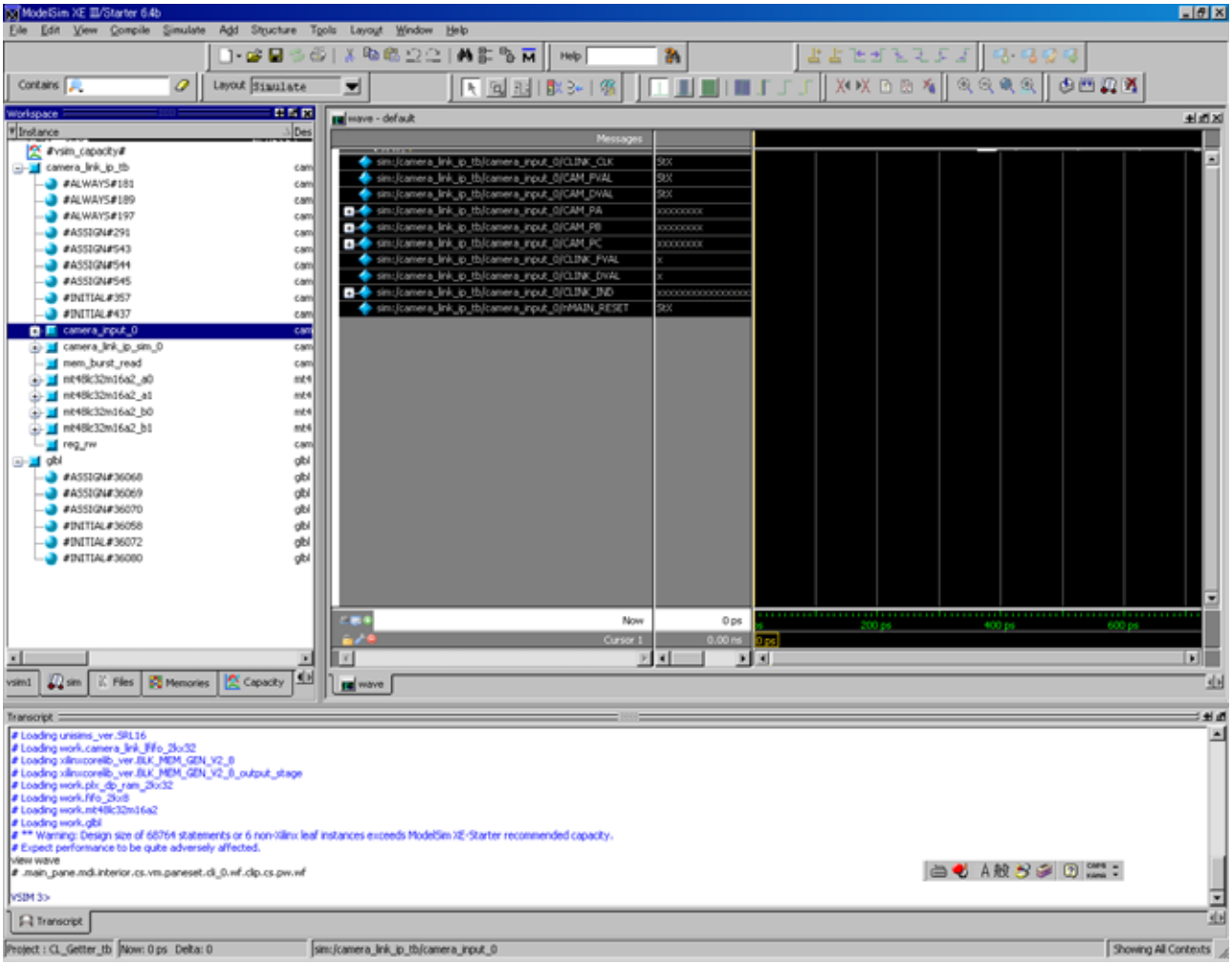
信号を波形で見る為に Wave ウィンドウを開きます。

「View」→「Wave」を選択します。

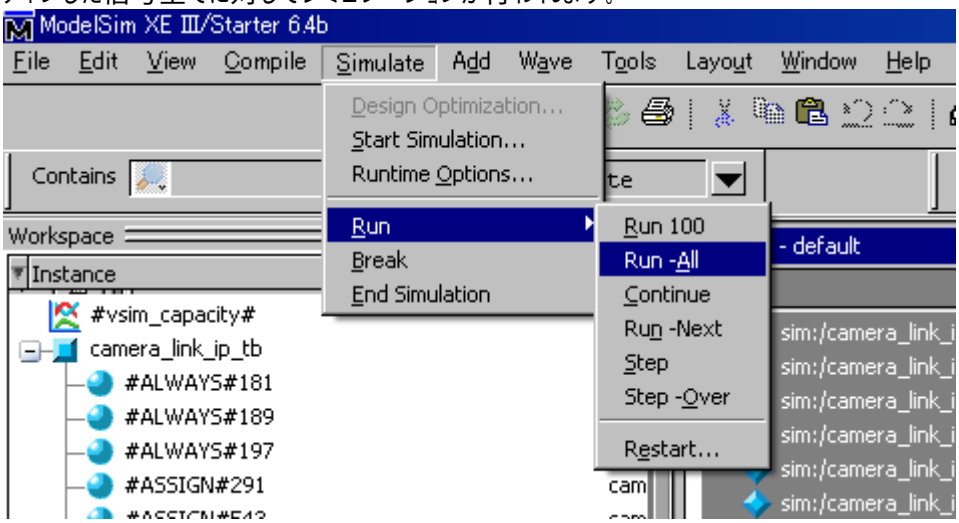


次に先程開いた wave ウィンドウに見たい信号を取り込みます。

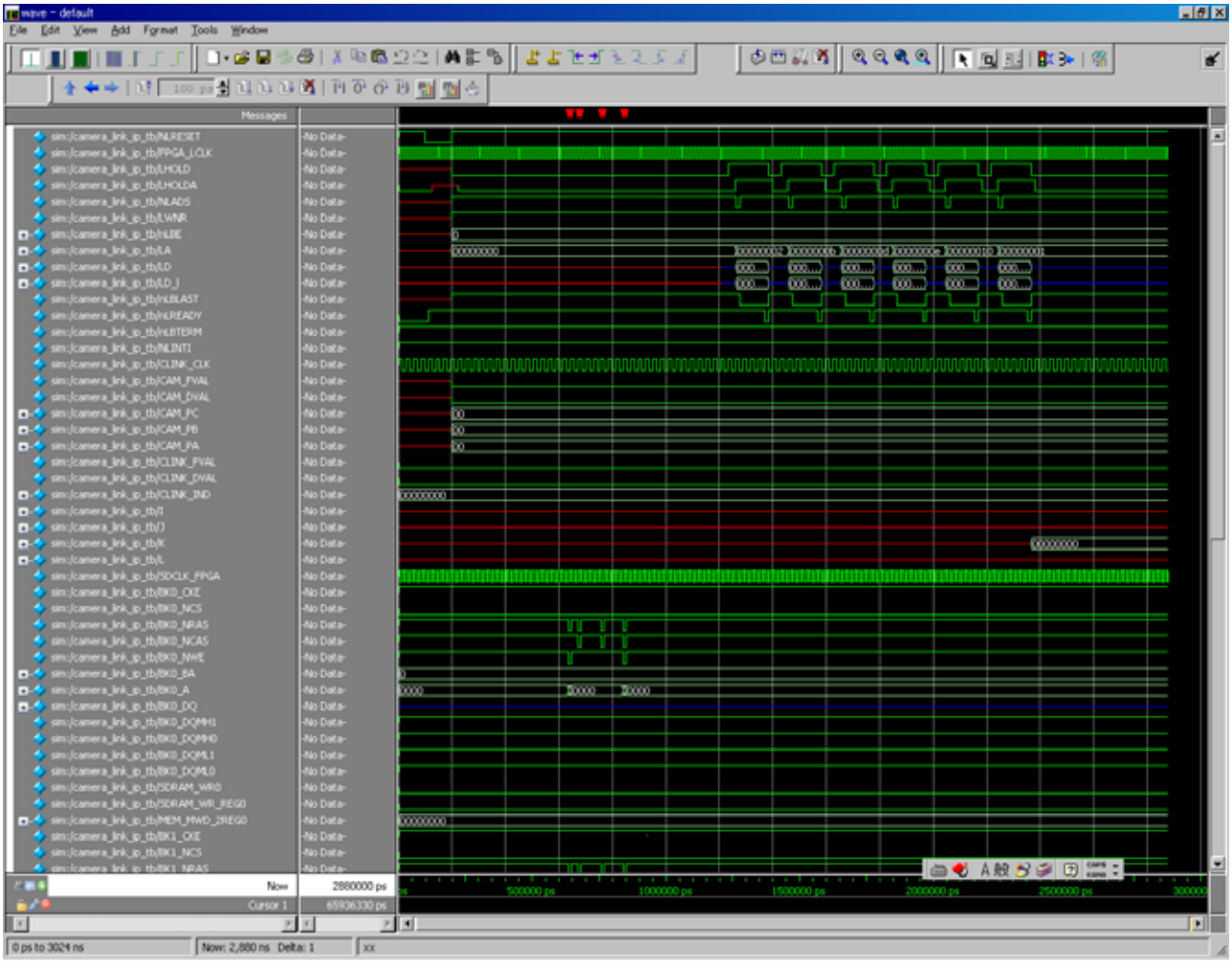
見たい信号上で右クリックし「Add」→「Add to Wave」をクリックします。選択したモジュールの信号が wave ウィンドウへ追加されます。



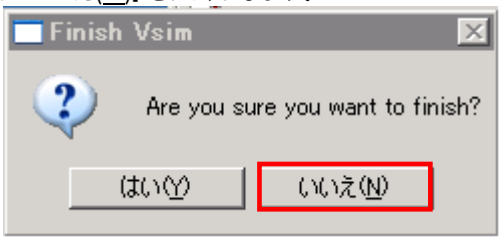
シミュレーションを開始します。「Simulate」→「Run」→「Run　All」を選択しますと先程「wave－default」画面にアサインした信号全てに対してシミュレーションが行われます。



シミュレーション中の波形が「wave－default」ウィンドウに出力されます。



テストベンチファイルのシステム・タスク $finish の所までシミュレーションが進みますと、自動的に終了します。「はい(Y)」をクリックすると ModelSim 自体が終了しますので、引き続きシミュレーション結果を見たい場合（波形データを見たい場合）は「いいえ(N)」をクリックします。



シミュレーション終了後、「wave－default」ウィンドウを見る事によってハードデバッグに使用する事ができます。

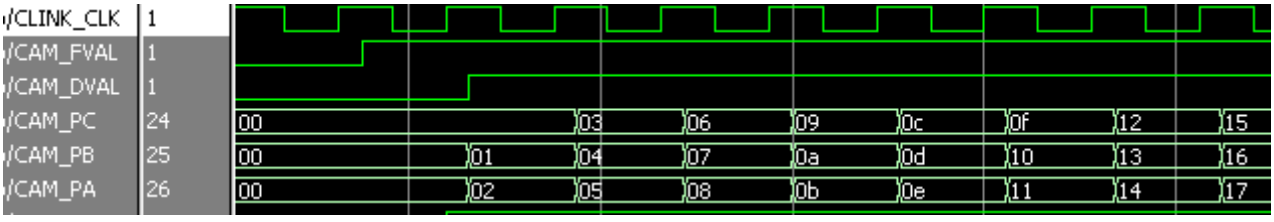
4.4.シミュレーションにおける注意事項

- 本資料で使用している SDRAM のシミュレーションモデルではイニシャル時間が実際に必要な時間より短くなっております。シミュレーション時間の短縮の為、IP のシミュレーションモデル(camera_link_ip_sim.v)は SDRAM のイニシャルを短くしております。(シミュレーション用イニシャル時間 400ns 程度)実機動作と異なりますのでご注意下さい。
- IP のシミュレーションモデル(camera_link_ip_sim.v)は論理合成出来ません。
- サンプルテストベンチソースはあくまで参考程度にご使用下さい。
- 今回使用してますテストベンチはファンクションシミュレーションとなっております。遅延時間が含まれておりませんのでシミュレーション結果と実機での動作には違いが出る場合があります。

4.5.シミュレーション入力データについて

以下にシミュレーション用カメラ入力データについて示します。

1 ライン 767 画素、1 画素 RGB24bit、PortC/PortB/PortA によるインクリメントパターンとなります。(下図参照下さい)
また、1 フレームは 5 ラインとなります。



4.6.シミュレーションフロー

5.取り扱い上の注意事項

【重要】本製品を正しく使用する為、下記の注意事項をお守り下さい。
これらの注意事項を守らなかった場合は、全て製品保証及びサポートの対象外となります。

- 基板保護の為、ほこりや湿気の多い場所では使用・保管しないで下さい。
- 本製品を直射日光の当る場所、火気や暖房器具の近くで使用・保管しないで下さい。
- 本製品に水などの液体を掛けないで下さい。感電及び故障の原因となります。
- 本製品を磁気や電波の発生する機器の近くでは、使用・保管しないで下さい。
- 基板に強い衝撃や静電気を与えないよう、丁寧に取り扱いして下さい。故障の原因となります。
- CL Getter は PCI Express(Gen1)スロットへ挿入する事を前提としております。他の I/F や改造基板への接続はしないで下さい。
- 電源電圧は使用の範囲内で正しく使用して下さい。仕様を満足していない電源や不安定な電源を使用した場合、基板が故障もしくは誤動作する事があります。
- 専用ケーブルを使用する事で、Camera Link 対応のカメラへ接続可能です。
- 本製品を装置へ装着する際は必ず装置の電源を OFF にして行ってください。また、取り外す際も装置の電源を OFF にして下さい。感電及び故障の原因となります。
- 本基板への一切の加工行為を行わないで下さい。
- カメラリンクケーブルや PCI Express スロットからの活線挿抜をしないで下さい。
- Vender ID/Device ID は不正に使用しないで下さい。

お問い合わせ先

・販売に関するお問い合わせ

販売用メールアドレス：sales@kitech.co.jp

・技術に関するお問い合わせ

サポート用 URL　http://www.kitech.co.jp/company/contact_index.html

サポート用メールアドレス：support@kitech.co.jp