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

Copyright (C) K.I. Technology CO., LTD. All Rights Reserved

## 変更履歴

初5 2010/01/18 新規作成 0000 (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	版数	日付	変更箇所	変更内容	作成
	初版	2010/01/18	新規作成		KIT



はじめに	4
1.概要	5
2.CL Getter IP 仕様について	6
2.1.CL Getter IP 概要	6
2.2.CL Getter IP プロック図	6
2.3.CL Getter IP 内部プロック説明	6
2.4.CL Getter IP で接続する信号について	7
2.5.CL Getter IP 性能について	10
2.6.CL Getter IP で接続する信号について	10
2.7. 接続信号仕様、制約について	11
3.CL Getter について	12
3.1.CL Getter 基本構成	12
3.2.入力 I/F モジュールについて	13
3.3.ユーザー回路について	15
3.4.ユーザー回路と CL Getter IP の接続について	17
3.5.ユーザーレジスタ、ユーザーメモリ設計例	
3.5.1.ユーザーレジスタ記述例	
3.5.2.ユーザーメモリ記述例	19
3.6.コンパイルについて	20
3.6.1.制約ファイル設定	20
3.6.2.IOFF 設定について	21
3.6.2.1.Synthesize による IOFF 設定	21
3.6.2.2.MAP による IOFF 設定	22
4.シミュレーションについて	23
4.1.ModelSim プロジェクト作成	23
4.1.1.ModelSim 起動	23
4.1.2.プロジェクト作成	24
4.1.3.テストベンチ構成	26
4.2.ファイルのコンパイル	27
4.3.シミュレーションの実行	
4.4.シミュレーションにおける注意事項	
4.5.シミュレーション入力データについて	
4.6.シミュレーションフロー	
5.取り扱い上の注意事項	



## 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。

本マニュアルで使用されている各会社名、各製品名は各社の商標または登録商標です。



## 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 ) 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。

- キャプチャーデータパッキング回路 外部メモリコントローラ ローカルバスコントロール
- カメラ用シリアル通信制御

## 2.2.CL Getter IP プロック図

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), .NLADS (NLADS), .nLBE (nLBE), .LHOLDA (LHOLDA), .nLREADY (nLREADY), .NCCS (NCCS),	.LA (LA), .LWNR (LWNR), .nLBLAST (nLBLAST), .nLBTERM (nLBTERM), .NLINTI (NLINTI), .NDREQ0 (NDREQ0),	
.NDREQ1 (NDREQ1), .NBIGEND (NBIGEND), .NWAIT (NWAIT), 	.BREQI (BREQI), .nLLOCKI (nLLOCKI),	
.DEV_IO_CS(DEV_IO_CS),	.CPU_RD(CPU_RD),	
.CLINK_CLK (CLINK_CLK), .CLINK_DVAL (USER_DVAL), .SERTFG (SERTFG), .SERTC (SERTC),	.CLINK_FVAL (USER_FVAL), .CLINK_IND (USER_IND),	//Camera Link I/F
.SDCLK_FPGA (SDCLK_FPGA_I), .BK0_BA (BK0_BA), .BK0_NCS (BK0_NCS), .BK0_NRAS (BK0_NRAS), .BK0_NWE (BK0_NWE), .BK0_DQMH0 (BK0_DQMH0), .BK0_DQML0 (BK0_DQML0), .BK0_DQ (BK0_DQ),	.BK0_A (BK0_A), .BK0_CKE (BK0_CKE), .BK0_NCAS (BK0_NCAS), .BK0_DQMH1 (BK0_DQMH1), .BK0_DQML1 (BK0_DQML1),	//SDRAM : bank0
.BK1_NCS (BK1_NCS), .BK1_BA (BK1_BA), .BK1_NRAS (BK1_NRAS), .BK1_NWE (BK1_NWE), .BK1_DQMH0 (BK1_DQMH0), .BK1_DQML0 (BK1_DQML0), .BK1_DQ (BK1_DQ),	.BK1_CKE (BK1_CKE), .BK1_A (BK1_A), .BK1_NCAS (BK1_NCAS), .BK1_DQMH1 (BK1_DQMH1), .BK1_DQML1 (BK1_DQML1),	//SDRAM : bank1
.mDPSW (mDPSW), .sLED (sLED),	.mLED (mLED),	//Reserved & debug pin
.USER_REG_RD (USER_REG_RD), .USER_FPGA_VER (USER_FPGA_VER),	.USER_MEM_RD (USER_MEM_RD),	//User I/O
.nmain_rese1 (nmain_rese1), .User_mem_cS (User_mem_cS), .CPU_WE (CPU_WE), .LA_REG (LA_REG),	.USER_REG_CS (USER_REG_CS), .CPU_IO_ENB (CPU_IO_ENB), .LD_DI (LD_DI), .LWnR_REG (LWnR_REG),	
.SDRAM_WR0(SDRAM_WR0), .MEM_MWD_2REG0(MEM_MWD_2REG0),	.SDRAM_WR_REG0(SDRAM_WR_REG0),	
.SDRAM_WR1(SDRAM_WR1), .MEM_MWD_2REG1(MEM_MWD_2REG1)	.SDRAM_WR_REG1(SDRAM_WR_REG1),	

);

#### <u>ローカルバスコントロール信号</u>

ピン名	I/O	機能
FPGA_LCLK		ローカルバスクロック 66MHz固定
NLRESET		ローカルバスリセット信号(Low Active)
LHOLD	_	ローカルバスホールドリクエスト
LA[31:2]		ローカルアドレス
NLADS		ローカルバス アドレスストローブ
LWNR		ローカルバス ライト/リード
nLBE[3:0]		ローカルバス バイトイネーブル
nLBLAST		ローカルバス バーストラスト
LHOLDA	0	ローカルバス LHOLD ACK信号
nLBTERM	0	ローカルバス バーストターミネート信号
nLREADY	0	ローカルバス I/Oアクセス レディー
NLINTI	0	ローカルバス 割込信号
NCCS	0	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	0	ローカルバス DMA Demand Modeセレクト0
NDREQ1	0	ローカルバス DMA Demand Modeセレクト1
BREQI	0	ローカルバス バスリクエストIN信号
NBIGEND	0	ローカルバス ビッグエンディアンセレクト
nLLOCKI	0	ローカルバス ローカルロックIN
NWAIT	0	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		カメラリンク データピクセルクロック
CLINK_FVAL		カメラリンク フレームバリッド
CLINK_DVAL		カメラリンク ピクセルバリッド
CLINK_IND[31:0]		カメラリンク 入力データ
SERTFG		カメラリンク シリアル入力
SERTC	0	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		SDRAM_A 動作クロック 80MHz
BK0_NCS	0	SDRAM_A チップセレクト
BK0_CKE	0	SDRAM_A クロックイネーブル
BK0_BA[1:0]	0	SDRAM_A バンクアドレス
BK0_A[12:0]	0	SDRAM_A アドレス
BK0_NRAS	0	SDRAM_A Rowアドレスストローブ
BK0_NCAS	0	SDRAM_A Columnアドレスストローブ
BK0_NWE	0	SDRAM_A ライトイネーブル
BK0_DQMH1	0	SDRAM_A上位バイトマスクイネーブル1
BK0_DQMH0	0	SDRAM_A 上位バイトマスクイネーブル0
BK0_DQML1	0	SDRAM_A 下位バイトマスクイネーブル1
BK0_DQML0	0	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	0	SDRAM_B チップセレクト
BK1_CKE	0	SDRAM_B クロックイネーブル
BK1_BA[1:0]	0	SDRAM_B バンクアドレス
BK1_A[12:0]	0	SDRAM_B アドレス
BK1_NRAS	0	SDRAM_B Rowアドレスストローブ
BK1_NCAS	0	SDRAM_B Columnアドレスストローブ
BK1_NWE	0	SDRAM_B ライトイネーブル
BK1_DQMH1	0	SDRAM_B 上位バイトマスクイネーブル1
BK1_DQMH0	0	SDRAM_B 上位バイトマスクイネーブル0
BK1_DQML1	0	SDRAM_B 下位バイトマスクイネーブル1
BK1_DQML0	0	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]		KIT1140 DPSW			
mLED[3:0]	0	KIT1140 LED			
sLED[3:0]	0	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]		ユーザーメモリリードデータ
USER_FPGA_VER[15:0]		ユーザー設定FPGAバージョン
nMAIN_RESET	0	メインリセット(NLRESET or Software Reset)
USER_REG_CS	0	ユーザーレジスタチップセレクト
USER_MEM_CS	0	ユーザーメモリチップセレクト
CPU_IO_ENB	0	ローカルバス I/Oアクセルイネーブル
CPU_WE	0	ローカルバス ライトイネーブル(FPGA_LCLK1パルス幅)
LD_DI[31:0]	0	ローカルバス FPGA内部LD信号
LA_REG[16:2]	0	ローカルバス FPGA内部LA信号
LWnR REG	0	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 camera	type = "user_black_box" *) a_link_ip_camera_link_ip_0 ( .FPGA_LCLK (FPGA_LCLK_I), .LHOLD (LHOLD), .NLADS (NLADS), .nLBE (nLBE), .LHOLDA (LHOLDA), .nLREADY (nLREADY), .NCCS (NCCS), .NDREQ1 (NDREQ1), .NBIGEND (NBIGEND), .NWAIT (NWAIT), .LD (LD),	.NLRESET (NLRESET), I.A (LA), I.WNR (LWNR), .nLBLAST (nLBLAST), .nLBTERM (nLBTERM), .NLINTI (NLINTI), .NDREQ0 (NDREQ0), .BREQI (BREQI), .nLLOCKI (nLLOCKI), .DP (DP),	//PLX
	.CLINK_CLK (CLINK_CLK), .CLINK_DVAL (USER_DVAL), .SERTFG (SERTFG), .SERTC (SERTC),	.CLINK_FVAL (USER_FVAL), .CLINK_IND (USER_IND[31:0]),	//Camera Link I/F
	USER_REG_RD (USER_REG_RD), USER_FPGA_VER (USER_FPGA_VER), nMAIN_RESET (nMAIN_RESET), USER_MEM_CS (USER_MEM_CS), .CPU_WE (CPU_WE), I.A_REG (LA_REG),	.USER_MEM_RD (USER_MEM_RD), .USER_REG_CS (USER_REG_CS), .CPU_IO_ENB (CPU_IO_ENB), .LD_DI (LD_DI), .LWnR_REG (LWnR_REG),	//User I/O
);			

//+++ (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]

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。

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。

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。

CAPTURE\_HPSZ は 1 ライン周期以下として下さい。

CAPTURE\_VSZ は1フレームのライン数以下として下さい。

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。

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。





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) |----- user\_nem\_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。

SPVAL       CMIFVAL       CLINK (VAL         AM DVAL       CLINK (VAL       CLINK (VAL         CAM DVAL       CLINK (VAL       CLINK (VAL         PR(70)       CAM PA       CLINK (VAL         CLINK (LKL       CLINK (VAL       CLINK (VAL         CLINK (LKL       CLINK (VAL       CLINK (VAL         CLINK (LKL       CLINK (LKL       CLINK (LKL         CLINK (LKL       USR REG RD(S10)       CLINK (LKL         CLINK (LKL       USR REG RD(S10)       CLINK (LKL         USR REG RD(S10)       USR REG RD(S10)       USR REG RD(S10)         LIDD[101)       LARG(M21)       CLINK (LKL         LARG(M21)       USR REG RD(S10)       USR REG RD(S10)         LIDD[101)       LARG(M21)       USR REG RD(S10)         LINK REG       CD(S10)       CLINK (LKL         MAIN RESET       1       Active Low 02: minipility be 1: 667         LINK REG       1       D: 591/27/bis RD(21/07)       D: 591/27/bis RD(21/07)         MAIN RESET       1       D: 591/27/bis RD(21/07)       D: 591/27/bis RD(21/07)         CAM LVAL       1       D: 591/27/bis RD(21/07)       D: 591/27/bis RD(21/07)         USR REG RS CS       1       D: 791/27/bis RD(21/07)       D: 591/27/bis RD(21/07)     <	カメラより	comoro inpur	+ \/	ユーザー回路へ	Volling HEL
X0VAL       CMI DVAL       CLINK LVAL       CLINK LVAL         PATA3       CAM LVAL       CLINK LVAL       CLINK LVAL         PATA3       CLINK LVAL       CLINK LVAL       CLINK LVAL         PATA3       CLINK LVAL       CLINK LVAL       CLINK LVAL         CLINK CLK       USER REG RDGS       CLINK LVAL       CLINK LVAL         POTA3       CLINK LVAL       USER REG RDGS       CLINK LVAL         USER REG CS       CLINK LVAL       USER REG RDGS       FRALCUK         USER REG CS       CUNCLK       USER REG RDGS       TAJJST         USER REG CS       COUNE       CLINK LVAL       CLINK RUGA         USER REG CS       COUNE       CC(11)       CLINK RUGA         USER REG CS       COUNE       CC(11)       CLINK RUGA         USER REG CS       COUNE       CC(11)       CLINK RUGA         USER REG CS       COUNE       TAJJCA REG       CLINK RUGA         USER REG CS       COUNE       TAJKARES       CLINK RUGA         USER RE	XEVAL			CLINK FVAL	camera_input camera_input_0( .CAM_FVAL(XFVAL),
XIVAL       CAMIVAL       CAMIVAL       CAMIVAL       CAMIPA         CAMIPA       CLINKIND       CLINKIND       CLINKIND       CLINKIND       CAMIPA         CAMIPA       CAMIPA       CLINKIND       CLINKIND <td< td=""><td>XDVAL</td><td>CAM_DVAL</td><td>CLINK_DVAL</td><td></td><td>.CAM_DVAL(XDVAL), .CAM LVAL(XLVAL).</td></td<>	XDVAL	CAM_DVAL	CLINK_DVAL		.CAM_DVAL(XDVAL), .CAM LVAL(XLVAL).
PATO       CMUPA       CLINK (ND       CLINK (ND)       CLINK (ND)         PTOTO       CAMPA       USER REG ADD       CLINK (ND)       CLINK (ND)         CLINK CLK       USER REG ADD       USER REG ADD       USER REG ADD       CLINK (ND)         PTOTO       CLINK (CLK       USER REG ADD       USER REG ADD       USER REG ADD         USER REG CS       DJJ57       CLINK (ND)       CLINK (ND)       CLINK (ND)         USER REG CS       DJJ57       CLINK (ND)       CLINK (ND)       CLINK (ND)         UNR REG (ND)       CD (NS)       DJJ57       CLINK (ND)       CLINK (ND)         UNR REG (ND)       CD (NS)       DJJ57       CLINK (ND)       CLINK (ND)         UNR REG (ND)       CLINK (ND)       DJJ57       CLINK (ND)       CLINK (ND)         MAIN RESET       NAMAR RESET       DJJ57       CLINK (ND)       CLINK (ND)       CLINK (ND)       CLINK (ND)       CLINK (ND)       DJS6 REG (ND)       CLINK (ND)       DJS7 DS CD (ND)       DJS7 DS CD (ND)       DJS7 DS CD (ND)       DJS7 DS CD (ND)       <	XLVAL	CAM_LVAL			.CAM_PA(PA),
PC701       CMXPB         CLWPCH       CLWPCH         CLWCALK       USER REG ADDS         PC401       USER REG ADDS         PC401       USER REG ADDS         PC401       USER REG ADDS         CUW K       USER REG ADDS         CUW K       USER REG ADDS         CUW K       CCK         CUN K       CCK         LUBCIBIO       CCKK         LUBCIBI	PA[7:0]	CAM_PA	CLINK_IND	CLINK_IND[31:0]	.CAM_PB(PB), .CAM_PC(PC),
CAM PC     CAM	PB[7:0]	CAM_PB		CL Cottor IDA	CLINK CLK(CLINK CLK).
CLINK-CLK       UESR-REG.RD       USER-REG.RDCS	PC[7:0]	CAM_PC		CL Getter IPY	CLINK_FVAL(CLINK_FVAL),
CL Getter IPより FPGA,LCLK UBER,REG,CS CPU,WE LD (0)(310) LD (0)(310) LA,REG(142) LWR,REG         DUSC,REG,CS CPU,WE LD (0)(310) LNR,REG(142) LWR,REG         DNJ 37 CC (4:1]         PFGA,LCLK LWR,REG         PFGA,LCLK LWR,REG           MAIN,RESET	CLINK_CLK	CLINK_CLK	UESR_REG_RD	USER_REG_RD0[31:0]	.CLINK_LVAL(CLINK_LVAL),
PGALCLX         PGALCLX (PGALCLX)           UESR, REG.CS         D/X 57           CPU,WE         CC           CPU,WE         CC(1:1)           LA, REG(142)         LV, REG(142)           LWINR, REG         aMAIN, RESET           INAIN, RESET         1           PGALCLK, PEALCLA, UNIN, REG(142)         LVINR, REG           LWINR, REG         1           MAIN, RESET         1           PGALCK, PEALCLA, UNIN, REG(142)         LVINR, REG           LWINR, REG         1           MAIN, RESET         1           PGALCK, PEALCLA, UNIN, RESET         Active Lowの非同期Utery, REG, NUISER, REG, RDOS), LUSER, REG, RDOS, LUSER, REG, RDOS, LUSER, REG, RDOS, PEALCLA, REG, LUSER, REG, RDOS, PEALCLA, REG, LUSER, REG, RDOS, PEALCLA, REG, RUSER, REG, RDOS, LUSER, REG, RDOS, RUSER, REG, RDOS, LUSER, REG, RDOS, PEALCLA, REG, RUSER, REG, RDOS,	CL Getter IPより	l	JSER_REG_RDCS	USER_REG_RDCS	.CLINK_IND(CLINK_IND),
UESR.REG.CS         パメラペ         CCU WE (CPU, WE )         CCU UNE (CPU, WE )         CAM UNA (CPU )         CAM UNA	FPGA_LCLK	FPGA_LCLK			.FPGA_LCLK (FPGA_LCLK_I), .USER REG CS(USER REG CS).
CPUWE       CC       CC       DURULUL (CO)         LDD[[313]       LARE[142]       LMRERSET       DURULUL (CO)         IMAR RESET       IMAR RESET       CC (CO)       IMAR RESET         IMAR RESET       I       PCA44M0/1-F0227/120+782(1/7)-7.2.7/120+7         CLUK (LK       I       D570/27/05-002704/70-72         CAM VAL       I       D570/27/05-002704/70-72         CAM VAL       I       D570/27/05-002704/70-72         CAM VAL       I       D570/27/05-002104/70-72         CAM PG[70]       I       D570/27/05-002104/70-72         CAM PG[70]       I       D570/27/05-002104/70-72         ISER, REG CS       I       I = 0-1//1/2, 71/10-74         UNR, RESET       I       D = 0//1/2, 71/10-74         USER, REG CS       I       I = 0-1//1/2, 71/10-74         UNR, RESE       I       D = 0//1/2, 71/10-74         UNR, RESE       I       D = 0//1/2, 71/10-74	UESR_REG_CS	UESR_REG_CS		7727	.CPU_WE(CPU_WE),
LD (0)(10) LAREG[142] LWM.R.REG LWM.R.REG LWM.R.REG LWM.R.REG LWM.R.REG LWM.R.REG LWM.R.REG LWM.R.REG LWM.R.REG LWM.REGT MAIN.RESET MAIN.CLK.MAIN.CLK.MAIN.CLK.MAIN.RESET MAIN.RESET MAIN.CLK.MAIN.CLK.MAIN.CLK.MAIN.RESET MAIN.RESET MAIN.RESET MAIN.RESET MAIN.CLK.MAIN.CLK.MAIN.CLK.MAIN.CLK.MAIN.CLK.MAIN.CLK.MAIN.CLK.MAIN.CLK.MAIN.CLK.MAIN.CLK.MAIN.CLK.MAIN.CLK.MAIN.CLK.MAIN.CLK.MAIN.CLK.MAIN.CLK.MAIN.CLK.	CPU_WE	CPU_WE	сс —	CC[4:1]	.LD_DI(LD_DI), .LA_REG(LA_REG),
LAREG142] LWRR.REG         LV.REG142] LWR.REG         JUSER REG.RDUISER.REG.RDUIS. MAIN.RESET           MAIN.RESET         IO         MAIN.RESET         .CC(C). .MAIN.RESET           MAIN.RESET         I         PEOA@40.01~F0.22.0719v2.rV2.0719v2.rV2.vF PEOA@40.01~F0.22.0719v2.rV2.0710r2.rV2.0710r2.	LD_DI[31:0]	LD_DI[31:0]			.LWnR_REG(LWnR_REG),
LWR.REG       IMAIN RESET       IMAIN RESET       CICO, IMAIN RESET         IMAIN RESET       I       Active Lowの非同期"Destage PEGA全体のパードウェブリセット及りアウェフリセット CLINK CLK       I         NAME RESET       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Frame Valid信号         CAM, DVAL       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Forame Valid信号         CAM, DVAL       I       カメラリンクからのPort Aデーク         CAM, PA(7:0)       I       カメラリンクからのPort Aデーク         CAM, PC(7:0)       I       カメラリンクからのPort Bデーク         CAM, PC(7:0)       I       カメラリンクからのPort Cデーク         FPGA, LCLK       I       Dーカル/Xス 701/POA         USER, REG, CS       I       コーザーレジスクチップセレクト         LD, D(131:0)       I       Dーカル/Xス 701/POA         LD, D(131:0)       I       Dーカル/Xス FPGA内ABILVIR低号         LNR, REG       I       Dーカル/Xス FPGA内ABILVIR低号         LNR, REG       I       Dーカル/Xス FPGA内ABILVIR低号         LNR, REG       I       Dーカル/Xス PFGAPBILVTEATA </td <td>LA_REG[14:2]</td> <td>LA_REG[14:2]</td> <td></td> <td></td> <td>.USER_REG_RD(USER_REG_RD0),</td>	LA_REG[14:2]	LA_REG[14:2]			.USER_REG_RD(USER_REG_RD0),
	LWnR_REG	LWnR_REG			
Image: Numeric Structure		nMAIN_RESE	Т		.CC(CC),
Image: Section 1         Image: Section 1         Active Lowの非同期リセッド語号 FPGA全体のパードウェアリセット及びソフトウェアリセット CLINK CLK         I         Active Lowの非同期リセッド語号 FPGA全体のパ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ビケレルクロッ         CAM_FVAL         I         カメラリンクからのErame Valid信号           CAM_DVAL         I         カメラリンクからのData Valid信号         CAM_DVAL         I         カメラリンクからのData Valid信号           CAM_DVAL         I         カメラリンクからのPort Aデータ         CAM_EVAL         I         カメラリンクからのPort Aデータ           CAM_DVAL         I         カメラリンクからのPort Aデータ         CAM_EVAL         I         カメラリンクからのPort Aデータ           CAM_PG[7:0]         I         カメラリンクからのPort Aデータ         CAM_PG[7:0]         I         カメラリンクからのPort Aデータ           CAM_PG[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 Aデータ           CAM_PG[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 Aデータ           USER_REG_CS         I         I         I         Dーカルバス FPGA内部LD信号         I         I           LD_D[31:0]         I         I         Dーカルバス FPGA内部LD信号         I         D         D           LWR_REG         I         I         D         D         D					.nMAIN_RESET(nMAIN_RESET)
値号名         1/0         他でしいの3用同期142ッド留号 FPCA全体のバードウェアリセット及びソフトウェアリセット           NMAIN_RESET         1         カメラリンクからのどウセルクロック           CLINK_CLK         1         カメラリンクからのどウセルクロック           CAM_FVAL         1         カメラリンクからのData Valid信号           CAM_DVAL         1         カメラリンクからのData Valid信号           CAM_DVAL         1         カメラリンクからのData Valid信号           CAM_DVAL         1         カメラリンクからのData Valid信号           CAM_DPA(7:0)         1         カメラリンクからのData Valid信号           CAM_PB(7:0)         1         カメラリンクからのPort Aデータ           CAM_PB(7:0)         1         カメラリンクからのPort Bデータ           CAM_PC(7:0)         1         カメラリンクからのPort Bデータ           CAM_PC(7:0)         1         カメラリンクからのPort Bデータ           USER_REG_CS         1         ローカルレバスクロック           USER_REG_CS         1         ローカルレバス FPGA内部LD信号           LD_DI[31:0]         1         ローカルレバス FPGA内部LD信号           LA_REG[14:2]         1         ローカルレバス FPGA内部LD信号           LWnR_REG         1         ローカルレバス FPGA内部LD信号           LWnR_REG         1         ローカルレバス FPGA内部LD信号           LUM_R_REG         1         ローカルレバス FPGA内部LD信号           LUM_R_REG         1         ローカルレバス FP		nMAIN_RESEI			<i>]</i> ,
Image: Name         I         Free Resp. 400 (200 (200 (200 (200 (200 (200 (200		信号名	1/0		1
CLINK CLK         I         カメラリンクからのピケセルクロック           CAM_EVAL         I         カメラリンクからのPrame Valid信号           CAM_DVAL         I         カメラリンクからのData Valid信号           CAM_DVAL         I         カメラリンクからのData Valid信号           CAM_DVAL         I         カメラリンクからのData Valid信号           CAM_DA(T.O)         I         カメラリンクからのPort Aデータ           CAM_PE(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 Bデータ           CAM_PC[7.0]         I         カメラリンクからのPort Cデータ           PGA_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 FPGA内部LD信号           LD.D[131:0]         I         ローカルバス FPGA内部LM信号           LD.NR_REG         I         ローカルバス FPGA内部LM信号           LWnR_REG         I         ローカルバス FPGA内部LWnR信号           DAJUPOPOrtfractFracturera Linkratesmic TC Feit         The Statistical Stat		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	<del>。</del> ト及びソフトウェアリセット
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 Aデータ         CAM_PC[7:0]       I       カメラリンクからのPort Bデータ         CAM_PC[7:0]       I       D・カルバスクロック         USER_REG_CS       I       ローカルバス FPGA内部LDF         CPU_WE       I       ローカルバス FPGA内部LDF         L0_D[31:0]       I       ローカルバス FPGA内部LDF         LAREG[14:2]       I       ローカルバス FPGA内部LDF         LWR_REG       I       ローカルバス FPGA内部LDF         DUMR_TFYAL       O       camera_input vEジュールからのFrame Valid信号出力         CLINK_FVAL       O       camera_input vEジュールからのData Valid信号出力         CLINK_LVAL       O       camera_input vEジュールからのData Valid信号出力         CLINK_IND[31:0]       O       camera_input vEジュールからのData Valid信号出力         USER_REG_RD[31:0]		CLINK_CLK	I	カメラリンクからのピクセルクロック	
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Dort Aデータ         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 Aデータ         CAM_PC[7:0]       I       カメラリンクからのPort Cデータ         CAM_PC[7:0]       I       カメラリンクからのPort C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       ユーザーレジスクチッブセレクト         CPU_WE       I       ローカルバス ライトイネーブル(FPGA_LCLK1/\Thu又幅)         LD_DI[31:0]       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LA信号         LWnR_REG       I       ローカルバス FPGA内部LA信号         CLINK_FVAL       O       camera_input.vモジュールからのFrame Valid信号出力         CLINK_FVAL       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DataValid信号出力         CLINK_IND[31:0]       O       camera_input.vモジュールからのDataValid信号出力         USER_REG_RD[31:0]       O       コーザーレジスタリードデータ		CAM_FVAL	I	カメラリンクからのFrame Valid(	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         CAM_PC[7:0]       I       カメラリンクからのPort Cデータ         CAM_PC[7:0]       I       カメラリンクからのPort Cデータ         PGA_LCLK       I       Dーカルバスクロック         USER_REG_CS       I       コーザーレジスタチップセレクト         CPU_WE       I       Dーカルバス 576A内部LD信号         LD_D[0]1:0]       I       Dーカルバス FPGA内部LD信号         LAREG[14:2]       I       Dーカルバス FPGA内部LM信号         LWnR_REG       I       Dーカルバス FPGA内部LMR信号         LWnR_REG       I       Dーカルバス FPGA内部LMR信号         CLINK_FVAL       O       camera_input.vモジュールからのFrame Valid信号出力         CLINK_FVAL       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Data Valid信号出力         CLINK_LVAL       O       camera_input.vモジュールからのData/CAU力         USER_REG_RD[31:0]       O       コーザーレジスタリードデータ		CAM_DVAL	I	カメラリンクからのData 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         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ブセレクト         CPU_WE       I       ローカルバス ライトイネーブル(FPGA_LCLK1) (プルス輸)         LD_D[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LM信号         LWn R, REG       I       ローカルバス FPGA内部LWn R信号         DAJJYOOPOrtfff & LOTE XUTEXCHART SULT F3UL       MATHEMAN SULT F3UL         Image: Chi State Sult Sult Sult Sult Sult Sult Sult Sult		CAM_LVAL	I	カメラリンクからのLine Valid信号	
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         PPGA_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         LAREG[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LMR信号         LWnR_REG       I       ローカルバス FPGA内部LMR信号         LVMR_REG       I       ローカルバス FPGA内部LMR信号         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Line Valid信号出力         CLINK_IVAL       O       camera_input.vモジュールからのData Valid信号出力         UNK_LVAL       O       camera_input.vモジュールからのData/ズの出力         USER_REG_RD[31:0]       O       ユーザーレジスタリードデータ		CAM_PA[7:0]	I	カメラリンクからのPort Aデータ	
CAM_PC[7:0]       I       カメラリンクからのPot 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ブセレクト         CPU_WE       I       ローカルバス ライトイネーブル(FPGA_LCLK1/バルス幅)         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LMR信号         DJUJODPOrtff報に付きましてはCamera Link規格を参照して下さい。           Image: 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_IVAL       O       camera_input.vモジュールからのData/パス出力         USER_REG_RD[31:0]       O       ユーザーレジスタリードデータ		CAM_PB[7:0]	I	カメラリンクからのPort Bデータ	
FPGA_LCLK       I       ローカルパスクロック         USER_REG_CS       I       ユーザーレジスタチップセレクト         CPU_WE       I       ローカルパス ライトイネーブル(FPGA_LCLK1パルス幅)         LD_DI[31:0]       I       ローカルパス FPGA内部LD信号         LA_REG[14:2]       I       ローカルパス FPGA内部LA信号         LWnR_REG       I       ローカルパス FPGA内部LWnR信号 <b>かメラリンクのPort情報に付きよしてはCamera Link類格を参照して下さい</b> (信号名         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_IVAL       O         camera_input.vモジュールからのData/ズ出力         USER_REG_RD[31:0]       O         camera_input.vモジュールからのData/ズ出力		CAM_PC[7:0]	I	カメラリンクからのPort Cデータ	
USER_REG_CSIユーザーレジスタチップセレクトCPU_WEIローカルパス ライトイネーブル(FPGA_LCLK1)パルス幅)LD_DI[31:0]Iローカルパス FPGA内部LD信号LA.REG[14:2]Iローカルパス FPGA内部LA信号LWnR_REGIローカルパス FPGA内部LWnR信号 <b>カオリンクのPort情報に付きましてはCamera Link規格を参照して下さい</b> 信号名I/O催考CLINK_FVALOcamera_input.vモジュールからのData Valid信号出力CLINK_DVALOcamera_input.vモジュールからのData Valid信号出力CLINK_IND[31:0]Ocamera_input.vモジュールからのDataパス出力USER_REG_RD[31:0]Oユーザーレジスタリードデータ		FPGA_LCLK	I	ローカルバスクロック	
CPU_WEIローカルパス ライトイネーブル(FPGA_LCLK1/パルス幅)LD_DI[31:0]Iローカルパス FPGA内部LD信号LA,REG[14:2]Iローカルパス FPGA内部LA信号LWnR_REGIローカルパス FPGA内部LWnR信号 <b>かメラリンクのPort情報に付きましてはCamera Link規格を参照して下さい</b> 作号名I/O備考CLINK_FVALOcamera_input.vモジュールからのFrame Valid信号出力CLINK_DVALOcamera_input.vモジュールからのData Valid信号出力CLINK_LVALOcamera_input.vモジュールからのData Valid信号出力CLINK_IND[31:0]Ocamera_input.vモジュールからのData/パス出力USER_REG_RD[31:0]Oコーザーレジスタリードデータ		USER_REG_CS	I	ユーザーレジスタチップセレクト	
LD_DI[31:0]Iローカルパス FPGA内部LD信号LA_REG[14:2]Iローカルパス FPGA内部LMn保信号LWnR_REGIローカルパス 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現格を参照して下さい。信号名I/O催考CLINK_FVALOCLINK_DVALOcamera_input.vモジュールからのData Valid信号出力CLINK_LVALOcamera_input.vモジュールからのData Valid信号出力CLINK_IND[31:0]Ocamera_input.vモジュールからのData/(ス出力)USER_REG_RD[31:0]Oユーザーレジスタリードデータ		CPU_WE	I	ローカルバス ライトイネーブル	(FPGA_LCLK1パルス幅)
LA_REG[14:2]       I       ローカルパス FPGA内部LA信号         LWnR_REG       I       ローカルパス 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。		LD_DI[31:0]	I	ローカルバス FPGA内部LD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現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/(ス出力)         USER_REG_RD[31:0]       O       ユーザーレジスタリードデータ		LA_REG[14:2]	I	ローカルバス FPGA内部LA信	<u>₽</u>
カメラリンクのPort情報に付きましてはCamera Link現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パス出力         USER_REG_RD[31:0]       O       ユーザーレジスタリードデータ		LWnR_REG	I	ローカルバス FPGA内部LWnF	信号
信号名I/O備考CLINK_FVALOcamera_input.vモジュールからのFrame Valid信号出力CLINK_DVALOcamera_input.vモジュールからのData Valid信号出力CLINK_LVALOcamera_input.vモジュールからのLine Valid信号出力CLINK_IND[31:0]Ocamera_input.vモジュールからのDataパス出力USER_REG_RD[31:0]Oユーザーレジスタリードデータ		カメラリンクのPort	青報に付きましては	Camera Link規格を参照して下さ	, <b>I</b> ,
信号名I/O備考CLINK_FVALOcamera_input.vモジュールからのFrame Valid信号出力CLINK_DVALOcamera_input.vモジュールからのData Valid信号出力CLINK_LVALOcamera_input.vモジュールからのLine Valid信号出力CLINK_IND[31:0]Ocamera_input.vモジュールからのDataパス出力USER_REG_RD[31:0]Oユーザーレジスタリードデータ					
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パス出力         USER_REG_RD[31:0]       O       ユーザーレジスタリードデータ		信号名	1/0	備考	
CLINK_DVALOcamera_input.vモジュールからのData Valid信号出力CLINK_LVALOcamera_input.vモジュールからのLine Valid信号出力CLINK_IND[31:0]Ocamera_input.vモジュールからのDataパス出力USER_REG_RD[31:0]Oユーザーレジスタリードデータ		CLINK_FVAL	0	camera_input.vモジュールからの	DFrame Valid信号出力
CLINK_LVALOcamera_input.vモジュールからのLine Valid信号出力CLINK_IND[31:0]Ocamera_input.vモジュールからのDataパス出力USER_REG_RD[31:0]Oユーザーレジスタリードデータ		CLINK_DVAL	0	camera_input.vモジュールからの	DData Valid信号出力
CLINK_IND[31:0]Ocamera_input.vモジュールからのDataパス出力USER_REG_RD[31:0]Oユーザーレジスタリードデータ		CLINK_LVAL	0	camera_input.vモジュールからの	 DLine Valid信号出力
USER_REG_RD[31:0] O ユーザーレジスタリードデータ		CLINK_IND[31:0]	0	camera_input.vモジュールからの	DData/「ス出力
		USER_REG_RD[31:0]	0	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		USER_REG_RDCS	0	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)		CC[4:1]	0	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'h0000000;
              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	1/0	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 <b>規格を参照して下さい。</b>

信号名	1/0	備考
USER_FVAL	0	user.vモジュールからのFrame Valid信号出力
UESR_DVAL	0	user.vモジュールからのData Valid信号出力
USER_IND[31:0]	0	user.vモジュールからのDataバス出力
USER_REG_RD[31:0]	0	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'h0000000; 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〈、外付け SDRAM ヘデータが格納され、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), .LHOLD (LHOLD), .NLADS (NLADS), .nLBE (nLBE), .LHOLDA (LHOLDA), .nLREADY (nLREADY), .NCCS (NCCS), .NDREQ1 (NDREQ1), .NBIGEND (NBIGEND), .NWAIT (NWAIT),	.NLRESET (NLRESET), .LA (LA), .LWNR (LWNR), .nLBLAST (nLBLAST), .nLBTERM (nLBTERM), .NLINTI (NLINTI), .NDREQ0 (NDREQ0), .BREQI (BREQI), .nLLOCKI (nLLOCKI),	//PLX
.LD (LD),	.DP (DP),	
.CLINK_CLK (CLINK_CLK), .USER_DVAL (USER_DVAL), .SERTFG (SERTFG), .SERTC (SERTC),	.USER_FVAL (USER_FVAL), .USER_IND (USER_IND[31:0]),	//Camera Link I/F

);

//+++ (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 内部レジスタヘデータを格納しま

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'h0000000;
       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。



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

ISE トップ画面で Synthesize-XST 右クリック	Properties	を選択します	0
Processes for: kit2010_p2cap_top         Add Existing Source         Create New Source         ∑         View Design Summary         ∑         User Constraints         Q         Maplement Design         Q         Generate Programming File         Q         Configure Target Device         Analyze Design Using Chipscop	Rerun Rerun Bur Resun St Rerun <u>A</u> ll	Project Properties     Project Properties     Project Properties     Display In     Display In     Enhanced Design	rts sis Report ition Repo hanced De issage Fill cremental Summary ition Data rs nings ning Constr k Report
Processes	Open Witho Design Goa Con Properties.	uut Updating als & Strategies ISE Design Suite	10.1 の新礼

Category 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。

Property Name	Value
Add I/O Buffers	<u>v</u>
Max Fanout	100
Number of Clock Buffers	8
Register Duplication	<b>v</b>
Equivalent Register Removal	<b>N</b>
Register Balancing	No
Move First Flip-Flop Stage	
Move Last Flip-Flop Stage	
Pack I/O Registers into IOBs	Auto
Slice Packing	Auto
Use Clock Enable	Yes
Use Synchronous Set	Yes
Use Synchronous Reset	Yes
Optimize Instantiated Primitives	
	Property display level: Advanced 💌 Defa



#### 3.6.2.2.MAP による IOFF 設定

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



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

🚅 Process Properties - Map Properties		
<u>C</u> ategory		
Translate Properties		
Place & Boute Properties	Property Name	Value
Post-Map Static Timing Report Properties	Perform Timing-Driven Packing and Placement	<b>v</b>
Post-Place & Route Static Timing Report Properties	Map Effort Level	Medium 💌
Simulation Model Properties	Extra Effort	None
plorer Properties	Starting Placer Cost Table (1-100)	1
	Combinatorial Logic Optimization	
	Register Duplication	
	Ignore User Timing Constraints	
	Timing Mode	Non Timing Driven 💌
	Trim Unconnected Signals	ঘ
	Replicate Logic to Allow Logic Level Reduction	ঘ
	Allow Logic Optimization Across Hierarchy	
	Map to Input Functions	4
	Optimization Strategy (Cover Mode)	Balanced
	Generate Detailed MAP Report	
	Use RLOC Constraints	ব
	Pack I/O Registers/Latches into IOBs	For Inputs and Outputs
	Disable Register Ordering	For Inputs and Outputs
	CLB Pack Factor Percentage	For Inputs Only
	Map Slice Logic into Unused Block RAMs	Off
	Power Reduction	
	Power Activity File	
	Other Map Command Line Options	
	<u>P</u> roperty display	y level: Advanced 💌 Default
	OK Cano	el <u>A</u> pply Help

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。

M Create Project
Project Name
CL_Getter_tb
Project Location
CL_Getter/test_bench/CL_Getter_tb Browse
Default Library Name
work
Copy Settings From
ech_xe_starter/modelsim.ini Browse
● Copy Library Mappings ○ Reference Library Mappings
OK Cancel

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。

🙀 Add items to the Proj	ect 🗵
Click on the icon to a	add items of that type:
Create New File	Add Existing File
Create Simulation	Create New Folder
	Close

「Browse...」ボタンを押すと次ページの画面が出ます。

M Add file to Project	×
File Name	
	Browse
Add file as type	Folder
default 🖤	Top Level 🔻
Reference from current location	Copy to project directory
	OK Cancel



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。

Select files to add to	project				? ×
ファイルの場所(1):	Camera_link_ip_	tb	•	+ 🗈 💣 🎟	
最近使ったファイル で デスクトップ マイドキュメント マイ コンピュータ マイ ネットワーク	Camera_input.v camera_ink_ip_s camera_ink_ip_s camera_ink_ip_st camera_ink_ifito fiffo_2kx8.v mt48lc32m16a2; mt48lc32m16a2; muser_mem_512x3 user_mem_512x3	im.v ov _2kx32.v 2 2 2 2			
	ファイル名( <u>N</u> ): ファイルの種類( <u>T</u> ):	"camera_input.v" "came HDL Files (*v,*.vl,*.vhc	era_link_ip_sim.v i,*.vhdl,*.vho,*.ho	″ ″camera_lin dl,*.vo,*.vp,*.s ▼	開(( <u>0</u> ) キャンセル

「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。

V	orkspace					
r	Name	Status	Туре	Order	Modified	
	vi user_mem_512x32.v		Verilog	7	11/30/09 01:19:58 PM	
	🕀 camera_link_lfifo_2kx32.v		Verilog	3	11/24/09 04:29:17 PM	
	🔁 camera_link_ip_sim.v		Verilog	1	12/01/09 10:21:48 AM	
	🔁 mt48lc32m16a2.v		Verilog	8	05/21/04 10:36:44 AM	
	VL user.v		Verilog	6	12/24/09 03:04:16 PM	
	🔁 plx_dp_ram_2kx32.v		Verilog	5	11/24/09 04:30:01 PM	
	🔂 camera_link_ip_tb.v		Verilog	2	01/15/10 08:49:29 AM	
	🔂 fifo_2kx8.v		Verilog	4	11/24/09 04:32:05 PM	
	🔂 camera_input.v		Verilog	0	12/24/09 03:02:50 PM	

#### 4.1.3.テストペンチ構成

ファイル構成を先に示します。

camera_link_ip_tb.v	 camera_inputv	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。

Verilog         7         11/30/09 01           Verilog         7         11/30/09 01		
with camera_link_lifo_2lxx32.v         Verilog 3         11/24/09 04           with camera_link_jp_sim.v         Verilog 1         12/01/09 10           with d8lc32m16a2.v         Verilog 8         05/21/04 10           with user.v         Verilog 6         12/24/09 03           with pk_dp         Edit         Verilog 2         01/15/10 08           with ffo_2lxxt         Compile         Verilog 2         01/15/10 08           with camera         Edit         Compile Selected         04           with camera         Compile         Compile Compile All         04           with camera         Compile Compile Compile Order         Compile Order         07           With camera         Properties         Properties         Compile Summary         Compile Summary	1:19:58 PM 4:29:17 PM 0:21:48 AM 0:36:44 AM 3:04:16 PM 4:30:01 PM 8:49:29 AM 4:32:05 PM 3:02:50 PM	

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 vwas 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。 ModelSim XE III/Starter 6.4b - Custom Xilinx Version <u>File E</u>dit <u>V</u>iew Compile Simulate Add Project Tools Layo<u>u</u>t <u>W</u>indow 6 🕌 🛍 🏙 🎦 Start Simulation... Contains ▼ 2. Runtime Options... m Workspace <u>R</u>un <u>B</u>reak 🔻 Name Modified End Simulation 11/30/09 01:19:58 PM user\_mem\_512x32.v VĽ 11/24/09 04:29:17 PM camera\_link\_lfifo\_2kx32.v Verilog 3 νĽ camera\_link\_ip\_sim.v Verilog 1 12/01/09 10:21:48 AM VĽ Verilog 6 12/24/09 03:04:16 PM user.v νĽ 📄 plx\_dp\_ram\_2kx32.v 11/24/09 04:30:01 PM Verilog 5 camera\_link\_ip\_tb.v Verilog 2 01/15/10 08:49:29 AM 핸 fifo\_2kx8.v Verilog 4 11/24/09 04:32:05 PM νĽ camera\_input.v Verilog 0 12/24/09 03:02:50 PM VĽ

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。

M Start Simulation	×
Design VHDL Verilog Libraries SDF Others	<u>«</u> »
unisims_ver xilinxcorelib_ver	Add Modify Delete
-Search Libraries First ( -LF )	Add Modify Delete
OK	Cancel



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、「glbI」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)

🕅 Start Simulation	×
Design VHDL Verilog Libraries SDF Oth	ners
Type Name	🗸 Path
Module - M user_mem_512x32	F:/CL_Getter/test_ben
Module 🔣 Module	F:/CL_Getter/test_ben
Module plx_dp_ram_2kx32	F:/CL_Getter/test_ben
Module — 🕅 gibi	F:/CL_Getter/test_ben 🚽
Module — Mifo_2kx8	F:/CL_Getter/test_ben
Module — Module — Module — Module — Module	F:/CL_Getter/test_ben
Module – 🕅 camera_link_ip_tb	F:/CL_Getter/test_ben
Module — ink_ip_sim	F:/CL_Getter/test_ben
Module camera_input	F:/CL_Getter/test_ben
	· · · · · · · · · · · · · · · · · · ·
-Design Unit(s)	
work camera link in th albl	default
work.cdmcrd_rink_rp_cb_grbr	
Optimization	
	Optimization Options
	optimization options
	OK Cancel



「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 ウィンドウヘ





	~ ~ / _ / =	12 13 1 J	121 00 3	<b>7</b> 0				
ModelSim XE III/Starter 6.4	Ь							
$\underline{F}ile  \underline{E}dit  \underline{V}iew  \underline{C}ompile$	<u>S</u> imulate	A <u>d</u> d	W <u>a</u> ve	T <u>o</u> ols	Layo <u>u</u> t	<u>W</u> indow	<u>H</u> elp	
	Design Optimization Start Simulation			\$ <b>6</b>   ¥ 🖻 🛍 🗅 🔷				
Contains 🔎	Runtime Options			te				
Workspace	<u>R</u> un Break End Simulation			<u>R</u> un	100	dofoult		
Tinstance				Run	- <u>A</u> ll	- deradic		
				⊆ont	inue			
#vsim_capacity#				Ru <u>n</u> ·	-Next	sim:/cam	era link i	
□- <u>−</u> camera_link_ip_tb				Step		sim:/cam	era link i	
ALWAYS#181				Step	- <u>O</u> ver	sim:/cam	era link i	
						sim: /cam	era link i	
				R <u>e</u> st	art	sim:/cam	era link i	
— 🕘 #ASSIGN#291						sim, /camera_link_i		
						sim:/cam	era_IINK_I	

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。

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。

🔲 Finisl	h Vsim	×
2	Are you su	re you want to finish?
	はい M	いいえ(N)

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。

/CLINK_CLK	1										
/CAM_FVAL	1										
/CAM_DVAL	1										
/CAM_PC	24	00		)03	)06	09	<u>)0c</u>	X	Of (	12	(15
/CAM_PB	25	00	)01	)04	)07	į0a	)0d		10 )	13	(16
/CAM_PA	26	00	)02	)05	)08	Ĵ0b	<u>)</u> 0e		11	14	(17



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〈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 <u>http://www.kitech.co.jp/company/contact\_index.html</u>

サポート用メールアドレス:support@kitech.co.jp

