

DDR2 SDRAM をフレームバッファに使用した CMOS カメラ表示回路の実装

小野 雅晃

筑波大学システム情報工学等技術室

〒305-8573 茨城県つくば市天王台 1-1-1

概要

CMOS (Complementary Metal Oxide Semiconductor) カメラの撮影画像を XGA (eXtended Graphics Array) サイズのディスプレイに表示する回路を Xilinx 社の FPGA (Field-Programmable Gate Array) に実装した。CMOS カメラは Aptina 社の 300 万画素のカメラを使用している。カメラのフレームレートは 12 fps (Frames Per Second)、ディスプレイのフレームレートは 60 fps であるため、フレームレート変換が必要となる。そのためのフレームバッファとして、SO-DIMM (Small Outline Dual In-line Memory Module) の DDR2 SDRAM (Double-Data-Rate2 Synchronous Dynamic Random Access Memory) を使用している。

キーワード: CMOS カメラ、ディスプレイ表示、DDR2 SDRAM

1. はじめに

筑波大学システム情報系、和田耕一教授から研究用に使用するカメラ表示装置を作って欲しいとの依頼を受けた。カメラ表示装置は研究用の回路を載せるベースとなるシステムである。

FPGA 基板を選定し、アカデミック価格があって安価な Digilent 社の XUPV5-LX110T ボード^[1]と Aptina 社の 300 万画素の CMOS カメラを購入して開発がスタートした。

当初、うまく回路が動作せず悩んだ期間があったが、CMOS カメラが撮影した映像を XGA のディスプレイに表示することができた。

2. CMOS カメラ表示装置の概要

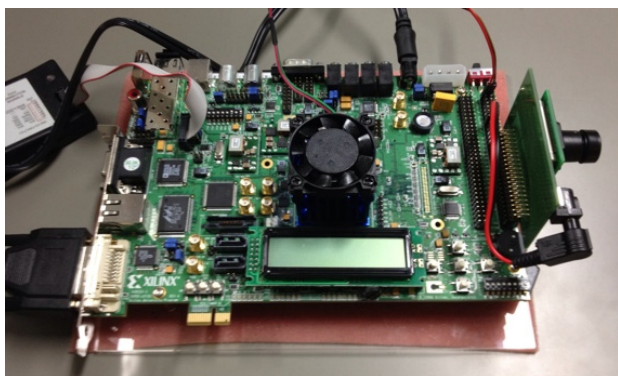


図 1. CMOS カメラ表示装置

CMOS カメラ表示装置を図 1 に、CMOS カメラ表示装置のブロック図を図 2 に示す。

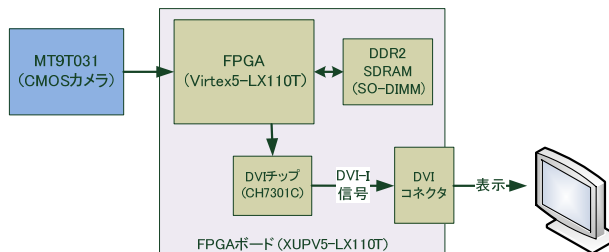


図 2. CMOS カメラ表示装置ブロック図

図 2 において、CMOS カメラ表示装置は、CMOS カメラと FPGA ボード (XUPV5-LX110T) の FPGA (Virtex5-LX110T)、DDR2 SDRAM (SO-DIMM)、DVI チップ (CH7301C)、DVI コネクタで構成されている。

CMOS カメラから取得された画像データは、FPGA 内で処理され、DDR2 SDRAM に書き込まれる。その画像データは、FPGA で読み出されて DVI チップに送られる。DVI チップで DVI 信号に変換され、DVI コネクタを通じてディスプレイに表示される。

CMOS カメラからの画像データを直接 DVI チップに送らないのは、主に画像のフレームレートが異なるからである。CMOS カメラは 12 fps で画像データを出力するが、ディスプレイは 60 fps の画像データを要求する。CMOS カメラからの画像データは 12 fps で DDR2 SDRAM に書き込み、ディスプレイに表示するために 60 fps で DDR2 SDRAM から読み出す。間に DDR2 SDRAM が入ることで、フレームレートの変換を行う。

3. CMOS カメラ^[2]

CMOS カメラは、Aptina 社の MT9T031 を使用した。MT9T031 は 300 万画素の CMOS デジタル・イメージセンサである。この CMOS カメラの有効画素数は 2,048×1,536 ピクセル、最大動作周波数は 48 MHz である。QXGA (2,048×1,536 ピクセル) の場合

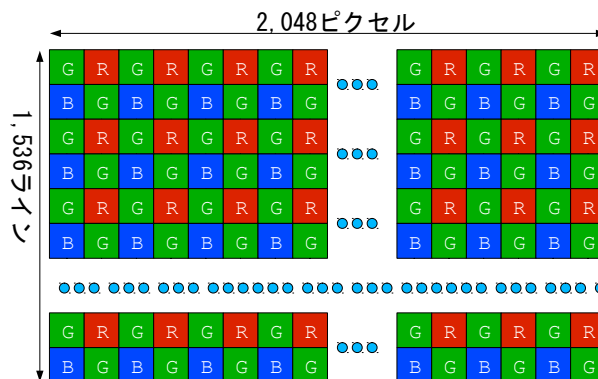


図 3. MT9T031 の画素の配置

に 48 MHz のクロックを入力すると 12 fps で画像をキャプチャすることができる。

MT9T031 の画素は Bayer カラー・パターンとなっている。赤 (R)、青 (B)、緑 (G) のいずれかの 1 色ずつの撮像素子の配列は Bayer カラー・パターンと呼ばれている。画素の配列の様子を図 3 に示す。図 3 に示すように、最初のラインの最初の画素は G である。その次は R となる。その後、G, R, G, R... と続く。最初のラインには B は存在しない。次のラインは B から始まり、その次は G となる。その後、B, G, B, G... と続く。2 番目のラインには R は存在しない。このように、R や B に対して G が 2 倍多くある。これは、人間の目の感度が緑付近をピークとなっていて、見かけ上の解像度を上げるためである^[3]。

通常は周辺の画素データを用いて補完を行うことで QXGA の画素データを作り出す。しかし、今回は画像処理に使用するベース回路として実装するので、2×2 の 4 画素のうちの 2 つの G の平均をとって新しい G とし、R と B はそのまま使用することにした。これで、元の画素の 1/4 の 1,024×768 ピクセル (XGA) の画像として使用している。(図 4 参照) これらの画像処理は FPGA ボード上の FPGA で行なっている。

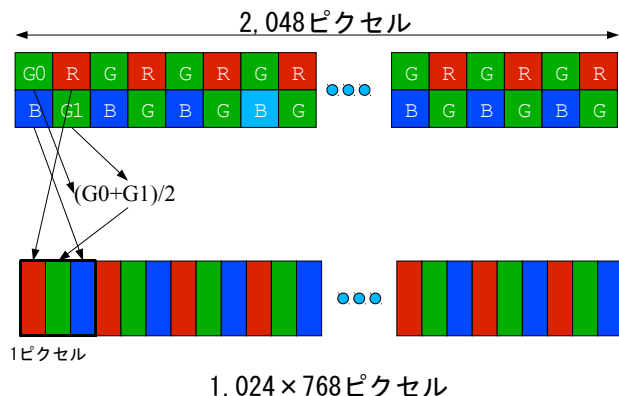


図 4. QXGA から XGA への画素の変換方法

MT9T031 には、動作モードの設定用レジスタが 256 個実装されている。これらのレジスタの設定方法は I2C (Inter-Integrated Circuit) シリアルインターフェースを介して行う。I2C シリアルインターフェースを FPGA に実装してあるので、初期化時にレジスタの値を設定することができる。設定できる項目としては、表示領域、ゲイン、シャッター状態等がある。現在は、カメラのゲインを設定している。

4. FPGA 内の回路ブロック

FPGA は、CMOS カメラからの画像データを受け取って、QXGA から XGA の変換を行った後で、DDR2 SDRAM へ書き込む。また、DDR2 SDRAM にバッファした画像データを読みだして、DVI チップのフォーマットに変換して DVI チップに書き込んでいく。

FPGA 内の回路ブロックを図 5 に示す。回路ブロックは、Camera Controller、Display Controller、Arbiter、DDR2 SDRAM Controller から構成される。

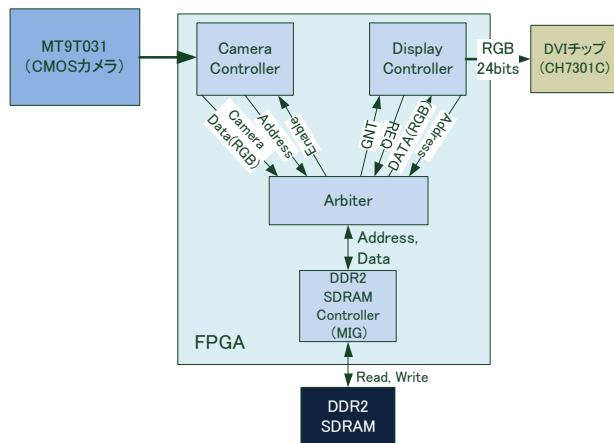


図 5. FPGA 内の回路ブロック

4.1 Camera Controller

Camera Controller のブロック図を図 6 に示す。

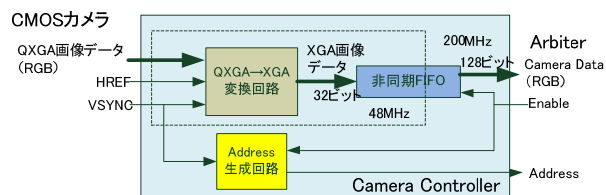


図 6. Camera Controller ブロック図

図 6 に示すように、CMOS カメラから入力された QXGA の画像データは、Camera Controller 内の QXGA→XGA 変換回路に入力され、XGA の画像データに変換される。変換された XGA 画像データは非同期 FIFO に 32 ビット長のデータとして入力される。ここまでの回路は CMOS カメラの動作周波数の 48 MHz クロックで動作する。Address 生成回路はフレームバッファ用の DDR2 SDRAM の番地を決定する。非同期 FIFO から画像データを出力するごとに Address をインクリメントして、画像フレームが終了したらリセットされる。非同期 FIFO は、Enable がアサートされたら 128 ビット長の画像データを Arbiter に出力する。32 ビットから 128 ビットへのビット長の変換は FIFO で行なっている。Address 生成回路と非同期 FIFO の出力は 200 MHz で動作する。

4.2 Display Controller

Display Controller のブロック図を図 7 に示す。

図 7 に示すように、Display Controller は、DDR2 SDRAM から入力した画像データ (DATA(RGB)) を DVI チップに送る働きをする。非同期 FIFO から DVI チップに送るデータが無くならないように画像データ管理回路が、非同期 FIFO 内の画像データを管理する。DVI チップに送る画像データが途切れると画像の乱れに直結する。

画像データの管理手順は、まず、初期化時に画像データ管理回路は Arbiter に REQ を送って画像データを要求する。GNT が帰ってきたらハンドシェイク信号を使用して、非同期 FIFO を画像データで FULL にする。その後、DVI チップに画像データを送り始める。

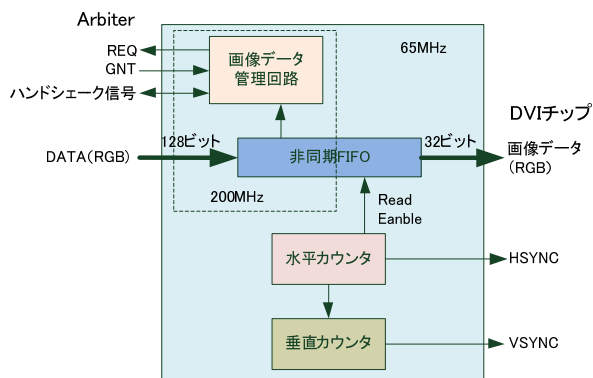


図 7. Display Controller ブロック図

画像データ管理回路は非同期 FIFO の容量が半分になったら、再度 Arbiter に REQ を送る。GNT が帰ってきたら、非同期 FIFO の半分の容量分の画像データを要求し、画像データを DDR2 SDRAM から受け取る。画像データを受け取っている間も、非同期 FIFO は画像データを DVI チップに送っているが、Arbiter のスループットが DVI チップの要求するスループットの約 12 倍であるため、非同期 FIFO に画像データが半分以上貯まることになる。

その他、Display Controller は、ディスプレイの表示機能を持っていて、画面のピクセルをカウントする水平カウンタ、画面のラインをカウントする垂直カウンタを持っている。水平カウンタからは水平同期信号 (HSYNC)、垂直カウンタからは垂直同期信号 (VSYNC) を出力している。Display Controller は、水平、垂直同期信号送出と共に DVI チップの入力フォーマットに合わせた画像データの送出を行う。

4.3 Arbiter

Arbiter は、Camera Controller の DDR2 SDRAM 書き込みと Display Controller の DDR2 SDRAM 読み出しの調停を行う。

Display Controller の非同期 FIFO にデータが無くなると画面が乱れてしまうので、Display Controller の要求を優先して DDR2 SDRAM の使用权を与える。

Display Controller が DDR2 SDRAM を読んでいる間に CMOS カメラから出力された画像データは、Camera Controller の非同期 FIFO に貯められる。Camera Controller の非同期 FIFO は十分な容量が確保されている。しかも、データがある程度溜まった方が、バースト長が長くなるので、データ書き込み帯域が増加してスループットが向上する。従って、CMOS カメラの画像データを書き損じることは無い。

図 8 に Arbiter のシミュレーション結果を示す。Camera Controller Signals の下には、Camera Controller からの信号を表示している。Display Controller Signals の下には、Display Controller からの信号を表示し、DDR2 SDRAM Controller Signals の下には、DDR2 SDRAM コントローラへの信号を表示している。

Display Controller のバス要求信号 `dispc_req` が 1 になると、Arbiter からバス許可信号 `dispc_gnt` を返す。Display Controller は `dispc_gnt` が帰ってきたら、`dispc_addr` に読み出す DDR2 SDRAM のアドレスをセットして、`dipc_addr_we` を 1 にアサートした後で 0 にする。`dipc_addr_we` の作業を Display Controller に実装された非同期 FIFO の半分の容量になるまで繰り返す。これらの動作は Read コマンドとして DDR2 SDRAM コントローラに投入される。DDR2 SDRAM Controller Signals の `ddr2_read_write` (1 = Read)、`ddr2_addr_fifo_wren`、`ddr2_address` を見れば同様の信号が投入されているのがわかる。`dipc_req` の立ち上がりから次の立ち上がりまでは、985nsec だった。実質的な Display Controller の DDR2 SDRAM の Read コマンド投入時間は 80 nsec だったので、80 nsec / 985 nsec x 100 ≒ 8 % の帯域を使用している。

Camera Controller の画像データの書き込みは、非同期 FIFO のカウント数 (`cmfifo_rd_data_count`) が 2 以上になったときに行われる。その場合は、Arbiter の回路から `camc_addr_ena` と `camc_data_ena` が出力される。`camc_addr_ena` が 1 クロック分アサートされ、`camc_data_ena` は 2 クロック分アサートされる。これらの動作は Write コマンドとして、DDR2 SDRAM コントローラに投入される。`ddr2_read_write` (1 =

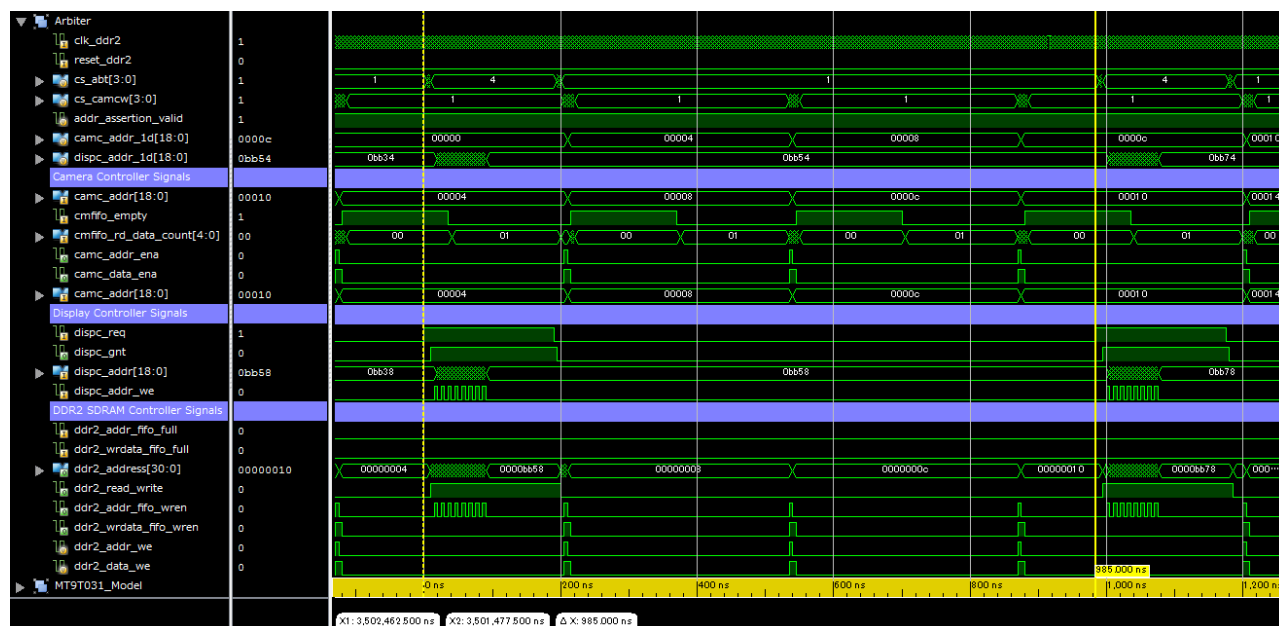


図 8. Arbiter のシミュレーション結果

Write)、 ddr2_address、 ddr2_addr_fifo_wren、 ddr2_data_ffio_wren が動いているのが見える。なお、Arbiter はアドレス、コマンドを処理するモジュールなので、データは処理されていない。データは他のモジュールを介して接続されている。Camera Controller のアクセスには 10 nsec 必要だった。

Display Controller の dispc_req の立ち上がりから次の立ち上がりまでに 985 nsec 必要で、その間に 10 nsec の Camera Controller のアクセスが 3 個入っている。 $(10 \text{ nsec} \times 3 + 80 \text{ nsec}) / 985 \text{ nsec} \times 100 \approx 11 \%$ となる。Camera Controller と Display Controller で全体域の 11 %を使用していて、残りの 89 %の帯域は空いている。これが全部使用できるわけではないが、十分に他の処理を並列に行うことができる。

5. DDR2 SDRAM コントローラ

DDR2 SDRAM コントローラは、Xilinx 社の標準の MIG (Memory Interface Generator) ^[4]を使用している。現在の動作周波数は 200 MHz で、Dual Data Rate で動作するので、DDR2 - 400 となり、400 MHz で動作するのと等価である。DDR2 SDRAM コントローラのデータの粒度は 256 ビットである。これは、4 (プリフェッチ数) \times 64 ビット (SO-DIMM のデータ幅) = 256 ビットのためである。MIG の Camera Controller や Display Controller へのデータ幅は 128 ビット長であるので、1 回のアクセスのために 2 回のデータ転送が必要となる。

図 9 に DDR2 SDRAM コントローラのシミュレーション結果を示す。これは図 8 に示したと同じ時刻の DDR2 SDRAM コントローラの波形となる。

図 9 において、app_ が付加されている信号は Arbiter 等との入出力信号となる。なお、Arbiter 以外にもデータバスが DDR2 SDRAM に接続されている。 ddr2_ が付加されている信号は DDR2 SDRAM への制御信号、データバス、クロックである。

これから DDR2 SDRAM へのコマンドの例を示す。 ddr2_ras_n = 1、 ddr2_cas_n = 0、 ddr2_we_n = 1 の信号は DDR2 SDRAM への Read コマンドとなる。Read

コマンドは Display Controller からの要求となるので、16 クロック期間に渡ってアサートされる。

ddr2_ras_n = 1、 ddr2_cas_n = 0、 ddr2_we_n = 0 の信号は DDR2 SDRAM への Write コマンドとなる。 Write コマンドは Camera Controller からの要求となるので、通常は 2 クロック期間アサートされる。

DDR2 SDRAM の Read コマンドと Write コマンドの期間を合わせると、“4.3 Arbiter”で導出した帯域と同じになる。これは、ACTIVATE コマンド (ddr2_ras_n = 0、 ddr2_cas_n = 1、 ddr2_we_n = 1) や PRECHARGE コマンド (ddr2_ras_n = 0、 ddr2_cas_n = 1、 ddr2_we_n = 0) やそのリカバリ時間を含んでいないので、それらを考慮すると帯域の使用率が上がることがわかる。

6. DVI チップ^[5]

DVI チップは Display Controller から出力された画像出力を DVI のフォーマットに変換する。DVI チップには Chrontel 社の CH7301C を使用している。この DVI チップは DVI-I 用で、デジタル出力とアナログ出力の両方を出力することができる。また、I2C インターフェースで設定できる設定レジスタを持っていて、テスト用カラーバーの設定や DAC の設定などのいろいろな設定を行うことができる。FPGA に設定レジスタ設定用の回路を搭載していて、XGA モードで表示できるように 3 つのレジスタを設定している。

7. まとめ

研究のベースとなるカメラ表示装置を開発した。CMOS カメラからの QXGA 画像の 2 \times 2 画素を 1 画素に変換して、XGA 画像とした。また、XGA 画像を DVI チップによって DVI に変換し、ディスプレイに出力して目で確認することができた。

今回のカメラ表示装置は画像のボケを復元するハードウェアの研究用として使用されている。

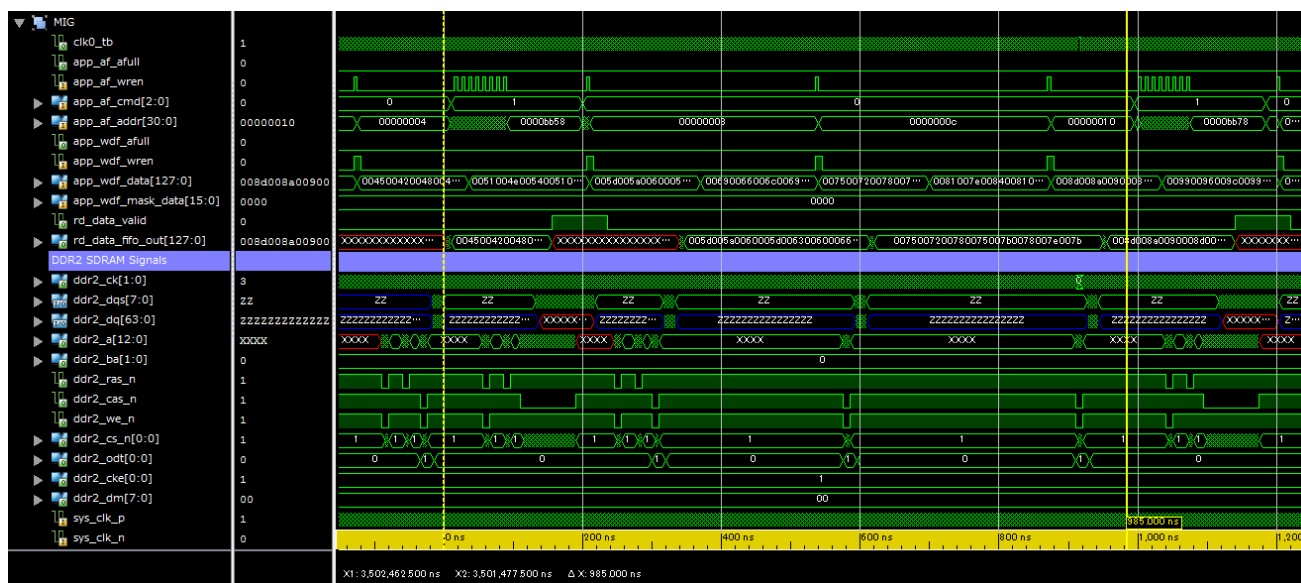


図 9. DDR2 SDRAM コントローラのシミュレーション結果

8. 謝辞

“DDR2 SDRAM をフレームバッファに使用した CMOS カメラ表示回路”を製作する機会を与えて頂いた、筑波大学システム情報系の和田耕一教授に深く感謝いたします。

参考文献

- [1] ML505/ML506/ML507 Evaluation Platform User Guide, Xilinx, UG347 (v3.1.1) October 7, 2009
- [2] 1/2-Inch 3-Megapixel CMOS Digital Image Sensor MT9T031, Aptina Imaging Corporation, MT9T031_DS - Rev.D 6/10 EN
- [3] CMOS イメージセンサ, ウィキペディア, <http://ja.wikipedia.org/wiki/CMOS%E3%82%A4%E3%83%A1%E3%83%BC%E3%82%B8%E3%82%BB%E3%83%B3%E3%82%B5>
- [4] Memory Interface Solutions User Guide, Xilinx, UG086 (v3.6) September 21, 201
- [5] CH7301C DVI Transmitter Device, Chronitel, 201-0000-056 Rev. 1.5, 3/17/2010