

Spartan3A Starter Kit による DDR2 SDRAM コントローラの実装

小野 雅晃

筑波大学システム情報工学等技術室 (装置開発班)

〒305-8573 茨城県つくば市天王台 1-1-1

概要

Spartan3A Starter Kit に搭載されている FPGA (Field Programmable Gate Array) 上に DDR2 SDRAM (Double Data Rate 2 Synchronous DRAM) のコントローラを実装した。Spartan3A Starter Kit には Xilinx 社の Spartan3A FPGA と Micron Technology 社の DDR2 SDRAM が搭載されている。DDR2 SDRAM のコントローラとは、DDR2 SDRAM を使用するためのインターフェース回路である。DDR2 SDRAM コントローラの動作周波数は 150 MHz で、クロックの立ち上がりエッジと立ち下りエッジでデータがサンプルされる。この方式は一般的に DDR2-300 と呼ばれる。DDR2 SDRAM のデータ幅は 16 ビットであるので、最大データ転送レートは 600 MBytes/sec である。DQS (Data Strobe) 信号を DDR2 SDRAM の Read データの受信クロックとして使用した。これは Source-Synchronous clocking と呼ばれる方式である。

キーワード : FPGA, DDR2 SDRAM

1. はじめに

DDR2 SDRAM は現在のパーソナル・コンピュータに搭載されているメイン・メモリとして広く使用されている。DDR は Double Data Rate の略であり、クロックの立ち下りと立ち上がりの両方でデータを読み書きすることができる。例えば、DDR2-800 は 800 の半分の 400 MHz で動作する DDR2 SDRAM チップである。DDR2 SDRAM チップのデータ幅は 4, 8, 16 ビットのものがある。それぞれデータ幅が異なるのは、データ幅と目標とするメモリ容量によって DDR2 SDRAM チップを使い分けるためである。

DDR2 は 4n プリフェッチと呼ばれ、一度にその DDR2 SDRAM チップのデータ幅の 4 倍のデータを読み書きできる構造になっている。そのため、DDR2 SDRAM 本体の DRAM 素子は、DDR2 SDRAM コントローラとのインターフェース速度の 1/4 の速度で動作出来れば良い。DRAM 素子の速度の向上が難しいため、このような手段は DDR3 SDRAM でも使用されている。DDR3 SDRAM は 8n プリフェッチとなっており、一度に 8 倍の粒度でデータを読み書きする必要がある。

DDR2 SDRAM はデータ転送速度が高速であるため、FPGA で制御することが難しいデバイスであると認識されている。FPGA の理解を深め、FPGA の限界を極める方法を模索するため、DDR2 SDRAM コントローラを作製することにした。

Xilinx 社の開発ボードである Spartan3A Starter Kit は FPGA (Spartan3A-700, xc3s700a-4fg484) と 512 Mbit (32M X 16bits, MT47H32M16BN-3:D) の Micron

Technology 社の DDR2 SDRAM が搭載されている。この Spartan3A Starter Kit を利用して DDR2 SDRAM コントローラを実装することにした。

2. DRAM の動作^[1-4]

DDR2 SDRAM は DRAM の一種である。DRAM はスイッチ用トランジスタと電荷を貯めるコンデンサによって構成される。一般的な DRAM 素子の構造の

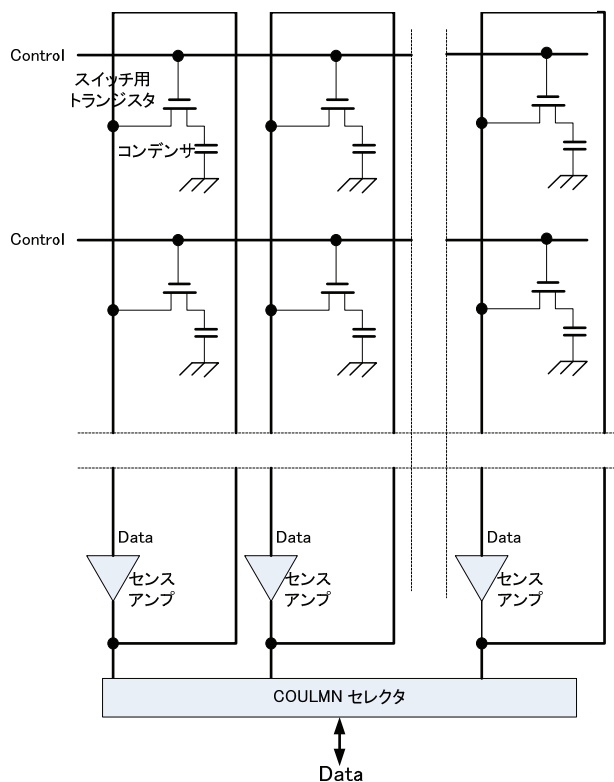


図 1. DRAM 素子の構造の概念図^[3,4]

概念図を図 1^[3,4]に示す。

図 1 のコンデンサに電荷がたまっている場合は論理 1、電荷がたまっていない場合は論理 0 となる。DRAM からデータを読みだす場合には、Control に電圧を印加して、その Control に接続されているスイッチ用トランジスタを ON にして、センスアンプにデータを読み込む。その時にコンデンサにたまっていた電荷は放電されて、DRAM 素子にはデータが無くなる (破壊読みだし)。この状態になると、COLUMN アドレスを入力してセンスアンプのデータを読み出したり、書き込んだりすることができる。ある一定の時間が過ぎるとセンスアンプに読みだしたデータを DRAM 素子に再書き込み (プリチャージ) する必要がある。これらの回路の集合体をバンクと呼んで

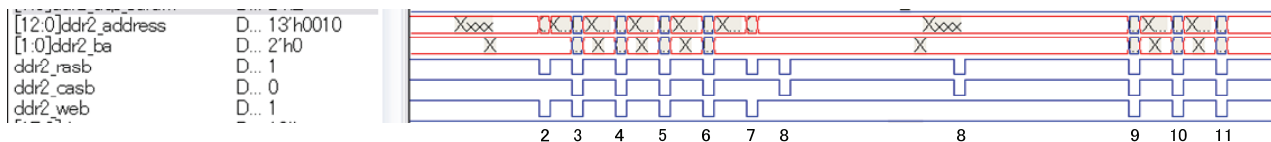


図 2. DDR2 SDRAM の初期化手順タイミングチャート

いる。通常の DDR2 SDRAM にはバンクが 4 つ入っている。全部のバンクをアクティブにすれば、COLUMN アドレスで表される数の 4 倍のメモリ領域に連続的（バースト）にアクセスすることができる。

DRAM 素子が論理 1 の場合にコンデンサに蓄えられている電荷は、リーク電流により時間とともに減少し、何もしないと論理 0 となってしまふ。そのため、DRAM では、リフレッシュと呼ばれるデータの書き込みが欠かせない。リフレッシュには集中リフレッシュ方式と分散リフレッシュ方式があるが、DDR2 SDRAM のメーカーによっては集中リフレッシュ方式を使用できない場合がある。今回製作した DDR2 SDRAM コントローラは分散リフレッシュ方式を採用し、7.8 us に 1 回のリフレッシュを行う。リフレッシュを行う場合は、全バンクのプリチャージを行ってから、リフレッシュを行う。

3. EDO DRAM から DDR2 SDRAM までの特徴

DRAM は” 2.1 DRAM の動作” で述べてきたとおり、内部は完全にアナログ動作である。EDO DRAM までは、入出力タイミングが非同期動作だったが、SDRAM になるとクロックに同期して入出力するための回路を DRAM に付加した構造となった。DDR SDRAM は SDRAM のプリフェッチ数を 2n とし、2 倍の粒度でデータを入出力することができるようにした SDRAM である。そのため DRAM 素子の速さを変更することなしに、2 倍の速度で入出力を行うことができる。DDR SDRAM では、DLL (Delay-Locked Loop) を内蔵して、クロック入力からデータ出力までの遅延をキャンセルすることができるようになった。

DDR2 SDRAM では、プリフェッチ数が 4n となつて、4 倍の速度で動作が可能となった。チップ内部に On Die Termination を内蔵し、任意のタイミングでターミネーターを入れることで、波形を改善することができる。更に Posted CAS をサポートして、より早いタイミングで Write または Read コマンドを入れることが可能となった。

4. DDR2 SDRAM の動作^[5]

4.1 DDR2 SDRAM の初期化

DDR2 SDRAM は、最初に使用するとき初期化が必要となる。図 2 に DDR2 SDRAM のシミュレーション時の初期化手順を示す。図 2 のタイミングチャートの下に書いてある番号は、下記の初期化手順の番号を示している。図 2 左端の ddr2_address はアドレス、ddr2_ba はバンク・アドレス、ddr2_rasb は /RAS (Row Address Strobe)、ddr2_casb は /CAS (Column Address Strobe)、ddr2_web は /WE (Write Enable) を表す。例えば、図 2 のタイミングチャートの 2 は全バンク・プリチャージ・コマンド (ddr2_address[10]='1', ddr2_rasb='0', ddr2_casb='1', ddr2_web='0') を示している。

1. 電源とクロックが安定した（リセットが解除されてから）200 us 後に、CKE を 1 にして NOP または DESELECT コマンドを入力する。
2. 400 us 待つて全バンク・プリチャージ・コマンドを入力する。
3. EMR(2)レジスタに値をセットする。
4. EMR(3)レジスタに値をセットする。
5. EMR レジスタに値をセットする（DLL をイネーブル）。
6. MR レジスタに値をセットする（DLL をリセット）。
7. 全バンク・プリチャージ・コマンドを発行する。
8. リフレッシュ・コマンドを 2 回発行する。
9. MR レジスタに値をセットする（DLL リセットなし）。
10. EMR レジスタに値をセットする（OCD Default）。
11. EMR レジスタに値をセットする（OCD exit）。

上記のように複雑な初期化シーケンスを踏む必要がある。6. で DLL をリセット後、200 クロックしてから ACT (bank ACTivate)、WRIT (Write)、READ (Read)、PALL (Precharge of ALL banks)、REF (Refresh) 等のコマンドを入力して、正常の操作ができるようになる。

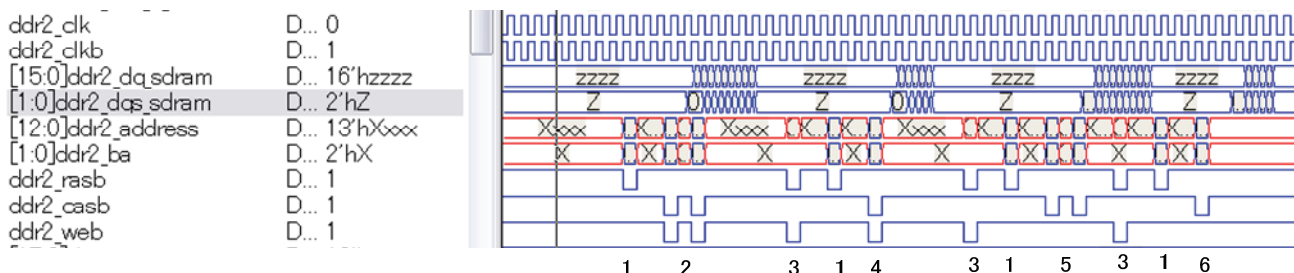


図 3. DDR2 SDRAM の Write, Read 動作タイミングチャート

4.2 DDR2 SDRAM の Write, Read 動作

DDR2 SDRAM の Write, Read 動作の一例を図 3 に示す。図 3 の左端の信号名の `ddr2_clk` は DDR2 SDRAM へ入力されるクロック、`ddr2_clkb` は `ddr2_clk` の反転クロック、`ddr2_dq_sdram` は DDR SDRAM に入出力するデータ、`ddr2_dqs_sdram` は DDR2 SDRAM に入出力するデータ・ストロブを示す。以下の信号は図 2 と同様となる。図 3 のタイミングチャートの下の番号が、下に示す番号の動作を表している。

1. ACT コマンドで、`ddr2_address` に Row アドレス、`ddr2_ba` にバンク・アドレスを指定してアクティベートする。
2. WRIT コマンドを 2 つ入力して、DDR2 SDRAM へ 2 バースト転送でデータを Write する。`ddr2_dq_sdram` と `ddr2_dqs_sdram` には DDR2 SDRAM コントローラからデータとデータ・ストロブが出力される。
3. 次に書き込むデータはバンクかまたは Row アドレスが異なるため、一旦、PALL コマンドで全バンクをプリチャージする。
4. WRIT コマンドを 1 つ入れて、単一転送でデータを書き込む。
5. READ コマンドを 2 つ入れて、2 バースト転送でデータを Read する。Read 時には、DQ と DQS は DDR2 SDRAM から出力される。
6. 単一転送の Read の例である。Read コマンドを 1 回発行している。

4.3 DDR2 SDRAM の Write, Read 動作のタイミング

DDR2 SDRAM の Write 動作のタイミングチャートを図 4^[5]に示す。上のブロックが FPGA 内部の信号を示し、下のブロックが DDR2 SDRAM での信号を示している。FPGA から DDR2 SDRAM に到達する信号は、IOB のバッファの遅延や配線遅延の分、FPGA 内部信号より遅延 (FPGAtoDDR2_delay) している。Write 動作時に DQS、DQS#、DQ と DM を入れるタイミングには tDQSS (NOM)、tDQSS (MIN)、tDQSS (MAX)の 3 通りのタイミングがある。今回の

DDR2 SDRAM コントローラでは tDQSS (NOM) のタイミングを使用している。tDQSS (NOM) のタイミングは WRIT コマンドを入れてから 2 クロック目のクロックの立ち上がりと同時に DQS が立ち上がるタイミングとなる。なお、図 4 の WL (Write Latency) は 2 クロックである。Write コマンドから 1 クロック後に、それぞれハイインピーダンス状態から DQS が 0、DQS#が 1 にアサートされる。その後、CK 及び CK#と同様な動作を行う。DQ は DQS や DQS#よりも位相が 90 度進んでいる。DQ は S2 の 1/4 クロック前からデータが有効になり、半クロックごとにデータを出力する。

DDR2 SDRAM の Read 動作のタイミングチャートを図 5^[5]に示す。Write 動作と同様に、上のブロックが FPGA 内部の信号を示し、下のブロックが DDR2 SDRAM での信号を示している。今、FPGA から READ コマンドが発行され、FPGA2DDR2_delay だけ遅延して DDR2 SDRAM に到達する。READ コマンドからデータが出てくるまでのレイテンシ(CL)は 3 クロックであるので、DDR2 SDRAM に READ コマンドが到達してから 3 クロック後に DQ が始まる。その DQ や DQS が DDR2 SDRAM から FPGA まで到達する遅延時間を DDR2toFPGA_delay とする。FPGA が Read データを受けるときの場合は、FPGAtoDDR2_delay+ DDR2toFPGA_delay の遅延が発生することになるので、Read データを受け取る FIFO (First In First Out) の Write イネーブルのタイミングを決定することが困難となる。そのため、あらかじめ FPGA 内部の READ_timing を SD_LOOP_OUT から出力し、FPGA から DDR2 SDRAM までの配線を行って戻って SD_LOOP_IN に帰ってきた READ_timing を使用して、FIFO の Write イネーブルとすればタイミングの問題は発生しない。

5. DDR2 SDRAM コントローラ

DDR2 SDRAM コントローラとは、DDR2 SDRAM を使うためのインターフェース回路である。DDR2 SDRAM コントローラは FPGA 内に搭載されて、DDR2 SDRAM との間で、初期化処理や書き込み (Write)、読み出し (Read) 動作を行う。Spartan3A Starter Kit には Xilinx 社の Spartan3A-700

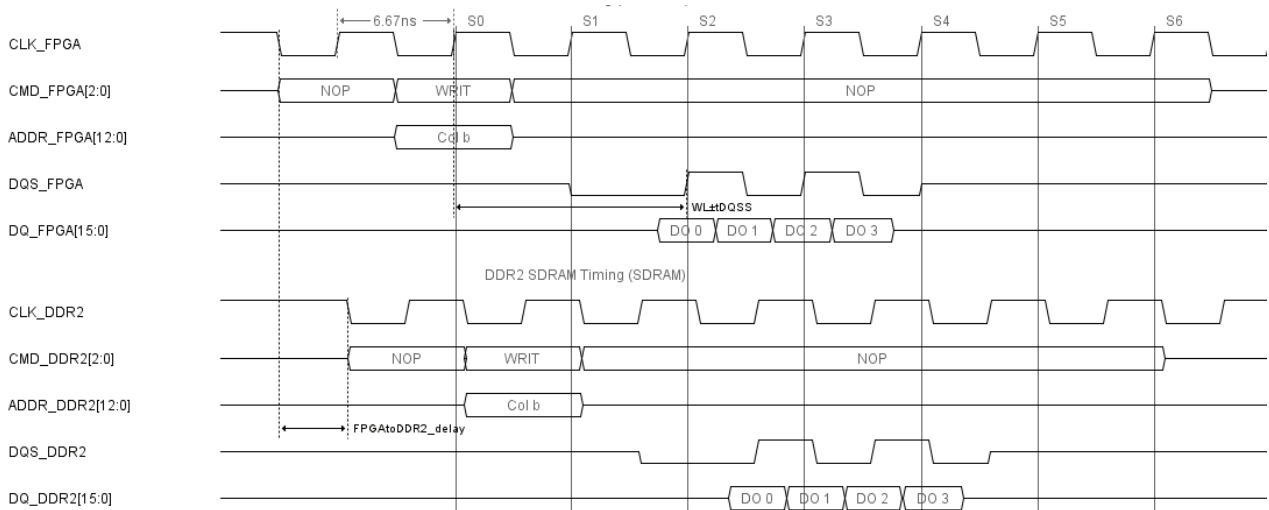


図 4. DDR2 SDRAM の Write 動作のタイミングチャート^[5]

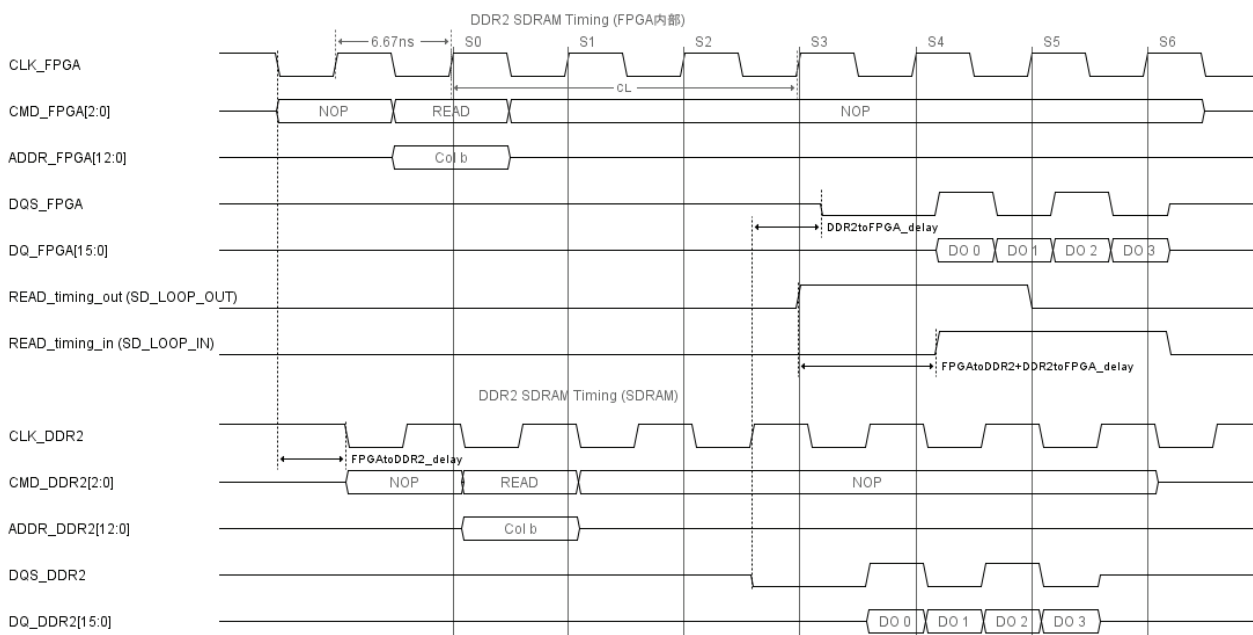


図 5. DDR2 SDRAM の Read 動作のタイミングチャート^[5]

(XC3S700A) という FPGA と Micron Technology 社の DDR2 SDRAM が搭載されている。今回作製した DDR2 SDRAM コントローラは FPGA に実装されて、Micron Technology 社の DDR2 SDRAM との間でデータの転送を行う。

5.1 DDR2 SDRAM コントローラの開発についての問題点

DDR2 SDRAM コントローラの開発は難しい。理由は高速のデータ転送速度にある。DDR2 SDRAM は供給されたクロックの立ち上がりエッジと立ち下りエッジに同期して、データを読み書きする。例えば、今回作製した DDR2-300 のクロック周波数は 150 MHz である。このクロックの両エッジでデータをサンプルまたは出力する。そのデータ・サンプル・ウィンドウ(データをサンプルすることができる幅)は、 $(1/150 \text{ MHz})/2 = 3.33 \text{ nsec}$ となる。更にクロックジッタ、データジッタやデータサンプル用の FF (Flip Flop) のデータ・セットアップ時間、データ・ホールド時間で削られて、いくらも残らなくなってしまふ。さらに、DDR2 SDRAM コントローラと DDR2 SDRAM 間のプリント基板上の距離も問題になってくる。一般的に FR-4 のプリント基板のマイクロストリップラインの伝搬遅延値は約 70 psec/cm ^[6]とされている。例えば、FPGA から DDR2 SDRAM までの配線長が 10 cm とすると、FPGA が DDR2 SDRAM からのデータを Read する場合のデータの遅れは $10 \text{ (cm)} \times 2 \times 70 \text{ (psec/cm)} = 1.4 \text{ nsec}$ となる。よって、配線による影響も大きく、配線長も考慮する必要がある。さらに、データとクロックのプリント基板上での配線長の差も問題となる。プリント基板上ではデータとクロックの配線を等長に配線する必要がある。Spartan3A Starter Kit では、図 6 に示すように、配線長をそろえる工夫がされている(等長配線)。等長配線がなされているとすると、Read データを受けるクロックを DQS にすれば、データと同

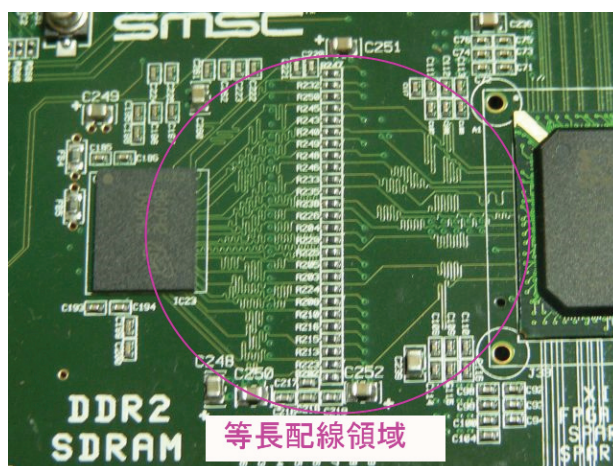


図 6. Spartan3A Starter Kit における等長配線

じ配線遅延があるはずである。よって DQS をデータ (DQ) よりも 1/4 周期遅延させることができれば、Read データを受けることができる(図 5 参照)。もう 1 つ問題になるのは、Read データが来るタイミングである。これは、FPGA の DDR2 SDRAM コントローラから Read コマンドを DDR2 SDRAM に発行して、Read 動作を行わせる場合である。何も工夫をしないと、配線による遅延時間や FPGA からのクロックの出力遅延時間、Read データが入ってくる FPGA の入力用 FF の遅延時間などで、DDR2 SDRAM コントローラはどのタイミングで Read データが取れるかがわからない。Spartan3A Starter Kit には、Read タイミングを伝送するための SD_LOOP_IN と SD_LOOP_OUT がある。DDR2 SDRAM コントローラは自分で Read のタイミング信号を SD_LOOP_OUT から出力して、DDR2 SDRAM までの配線遅延を追加した信号を再度 SD_LOOP_IN から入力することによって、Read データの来るタイミングを知ることができる。

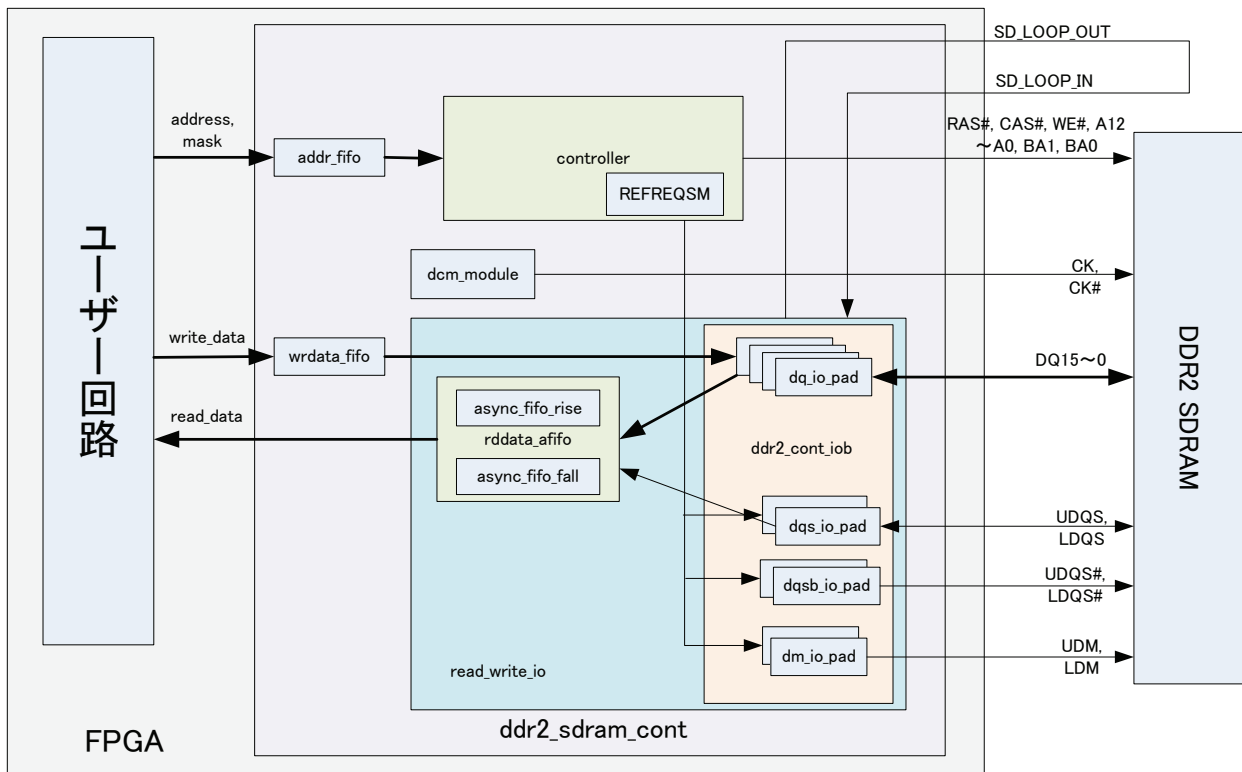


図 7. DDR2 SDRAM コントローラ・ブロック図

5.2 DDR2 SDRAM コントローラの特徴

今回作成した DDR2 SDRAM コントローラの特徴を以下に述べる。

1. Xilinx 社の IP の MIG (Memory Interface Generator)を使用しないオリジナルの DDR2 SDRAM コントローラである。
2. 150 MHz クロック動作の DDR2-300 の転送レートを持ち、データ幅が 16 ビットなので、データ転送レートは 600 MBytes/sec である。
3. DDR2 SDRAM の Read データは分散 RAM (LUT(Look Up Table))を使用した FIFO で受けられる。今回は IOB (Input Output Block)の入力用 FF (Flip-Flop) を使用していない。
4. Read データのタイミングを計るために、プリント基板の配線を遅延線として使用している。
5. Read 時には、DQS を IOB の遅延素子で遅延して、位相を 90 度ずらしたクロックとして使用している。
6. ACT コマンドでアクティベートしたバンクは、プリチャージしないで保持し、同一 ROW アドレス、同一バンクの時には ACT コマンドを発行せずに直接 Read、Write コマンドを発行することができる。それ以外の場合は一旦、プリチャージして、もう一度対応する ACT コマンドを自動的に入力することができる。

5.3 DDR2 SDRAM コントローラの構成

DDR2 SDRAM コントローラは、Xilinx 社のアプリケーションノート XAPP253^[7]を参考にして作製した。XAPP253 を参考にしてはいるが、完全にオリジナル

の設計となっている。使用言語は Verilog2001 である。DDR2 SDRAM コントローラの各ブロックを図 7 に示す。DDR2 SDRAM コントローラは、controller, addr_fifo, wrdata_fifo, rddata_afifo, read_write_io, dcm_module, そしてトップモジュールの ddr2_sram_cont で構成されている。各モジュールにはサブモジュールを持つモジュールもある。controller は REF_REQ_SM サブモジュール、rddata_afifo は async_fifo_rise async_fifo_fall サブモジュール、read_write_io は ddr2_cont_job サブモジュールを持つ。更に ddr2_cont_job サブモジュールは、その下に dq_io_pad, dqs_io_pad, dqsb_io_pad, dm_io_pad のサブモジュールを持っている。dq_io_pad, dqs_io_pad, dqsb_io_pad, dm_io_pad サブモジュールは、DDR2 SDRAM の各 IO に対応している。このようにファイルを細かく分けることで、フロアプラン時にも上手にフロアプランできるように考えて、モジュールを決定した。

controller は、全体の制御を行う。DDR2 SDRAM の初期化手順や Read, Write のタイミングなど、総合的な制御を受け持つ。addr_fifo はアドレスを入力する FIFO で、ユーザー回路からの DDR2 SDRAM のアドレスを入力する。アドレスを書く場合には同時に R/W に Read か Write かを書き込む。

wrdata_fifo はユーザー回路が write データを書き込む FIFO である。DDR2 SDRAM にデータを Write する場合には、addr_fifo に書かれるアドレスと同時に Write データを書くことになる。

read_write_io は Read, Write に関する DDR2 SDRAM への IO を行うモジュールである。DQ, DQS, DM などの IO とインターフェースを行う。

rddata_fifo は Read データを受ける非同期 FIFO (async_fifo_rise, async_fifo_fall) を持っている。DQS

をクロックとして DQ のデータを入力し、出力は内部クロックを基準としてデータを出力する非同期 FIFO を使って、Read データを内部クロックに同期して読み出す。このあたりの非同期 FIFO の取り扱いがもっとも苦労した部分である。

6. DDR2 SDRAM コントローラの作製過程 [8]

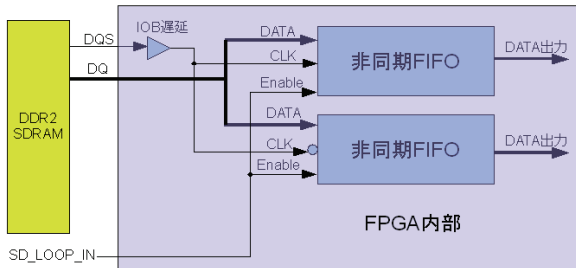


図 8. SD_LOOP_IN を使用した DDR2 SDRAM の READ データの受信回路ブロック図

最初に一番難しい Read データをどう受けるかというところを考えた。その結果、図 8 に示すように LUT の分散 RAM を FIFO 記憶素子として用いた非同期 FIFO を使用して、DQ を記憶することにした。DQS は IOB 遅延素子や FPGA の内部配線を利用して、DQ よりも 1/4 周期だけ遅延させることとした。さらに、Read タイミング信号を SD_LOOP_OUT に出力し、DDR2 SDRAM までの往復の配線遅延を含んだ信号を SD_LOOP_IN に入力する。その信号を非同期 FIFO の Write イネーブルとして使用することで、Read データのタイミングを検出することにした。さらに Write イネーブルの FPGA 内の配線は、低スキュー、低ディレイで供給する必要があったため、制

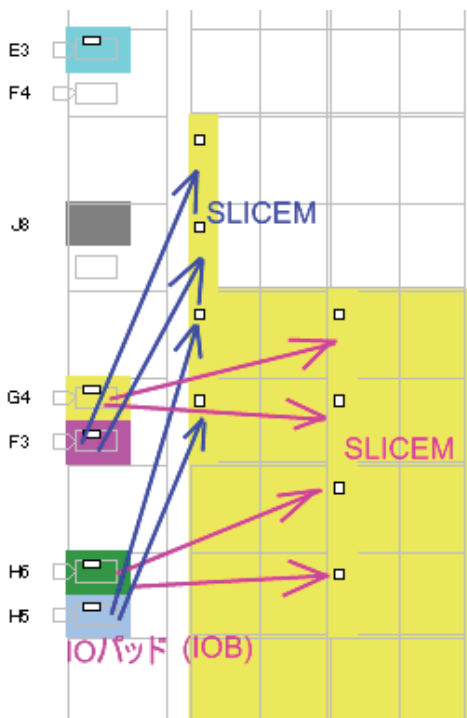


図 9. 非同期 FIFO 分散 RAM 素子のフロアプラン

約ファイルに MAXDELAY 制約、MAXSKEW 制約を付加して、一定のスキュー値、ディレイ値に抑えた。

動作周波数は 150 MHz と厳しく、入力のセットアップ時間のタイミングもかなり厳しいため、最初のインプリメントは、コントローラ部分を含まない IO 部分だけで実現の可能性があるかどうかを確かめてみた。タイミング制約をかけて試してみたが、非同期 FIFO の最初の分散 RAM 素子までの遅延が 2.2 nsec 程度ばらついてしまった。これは問題なので、Floorplanner ツールによるフロアプランを実行し、微調整を試みた。そうすると、この部分の最大遅延差は 142 psec に収めることができた。そのフロアプランの様子を図 9 に示す。図 9 で、IO パッドから非同期 FIFO の分散 RAM 素子となる SLICEM の位置を、どこにすれば遅延が最小になるかを Timing Analyzer ツールで確認しながらフロアプランを行った。

当初、標準 IP の非同期 FIFO を使用していたが、シミュレーションの結果により、同期リセットであることがわかった。DQS は常時クロックが出ていないので、同期リセットはシミュレーションが困難である。そこで、自分で非同期リセットの非同期 FIFO^[7]を作成した。これでシミュレーションがうまくできるようになり、インプリメントにも成功して、IO テストモデルでの作業は終了した。Advanced

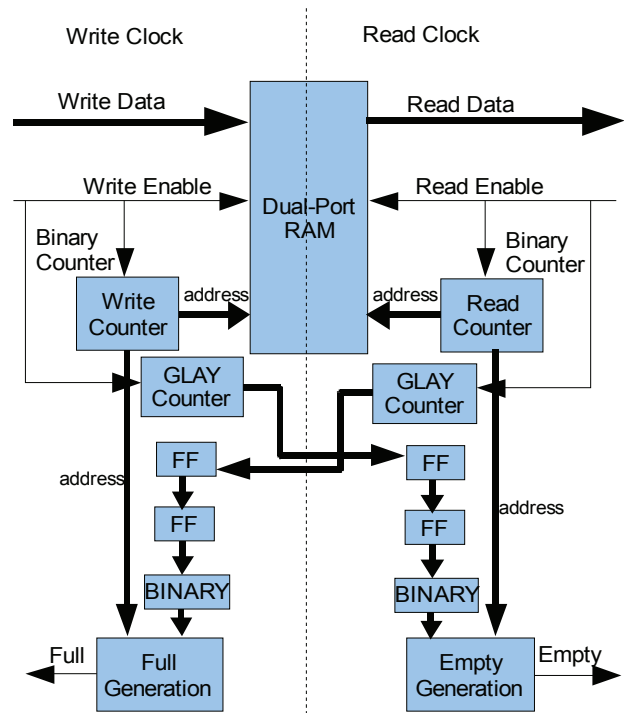


図 10. 非同期 FIFO ブロック図^[9]

FPGA Design^[9]を参考にして作製した非同期 FIFO の回路を図 10^[9]に示す。非同期 FIFO は Write data と Read Data のクロックが異なり、Write と Read が競合する場合があるためカウンタにグレイコード・カウンタを使用している。グレイコード・カウンタを使用することで、1 つカウンタが進むごとにカウンタ値が 1 ビットのみ変化する。このグレイコード値を相手側のクロックドメインに渡してバイナリデータ

に変換することによって、FIFO の Full や empty を誤りなく検出することができた。相手側のクロックドメインに渡されるグレイコード値は FF がメタステーブル状態に陥るのを防ぐために、相手側のクロックで2段に同期化した。

次のステップとして、IO テストモデルでの作業後にコントローラ全体を作成し、インプリメントすることにした。最初は動作周波数を 200 MHz としたかったが、Spartan3A-4 スピード・グレードでは無理だということがわかった。それで動作周波数のターゲットを 150 MHz に変更した。それでもかなり厳しいので、動作タイミングの変更などの処置を行った。その結果、何とかタイミングエラーがない状態まで持っていくことができ、FF のツリー上の複製は使用しないでもタイミング制約を満足することができた。

FF のツリー状の複製とは、FF が何段かつながっている状態で、最終段の FF から複数の出力が出ているときに、途中の FF をツリー状に複製することを言う。例えば、FPGA の中で3段の FF を介した後に4つの IO パッドに出力しているとする。FF のツ

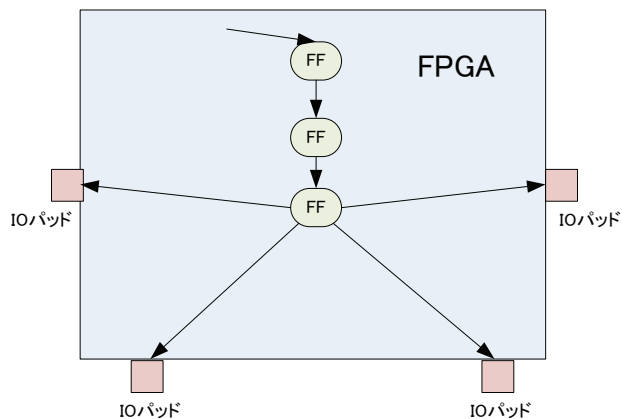


図 11. FF のツリー状の複製をしていない場合

リー状の複製をしていない場合を図 11 に示す。この状態では3段目の FF から4つの IO パッドすべてに対して配線が伸びている。この状態では3段目の FF をどの位置に配置しても、配線が長くなってしまふ。一方 FF のツリー状の複製をした場合を図 12 に示す。この場合には、FF がツリー状に複製

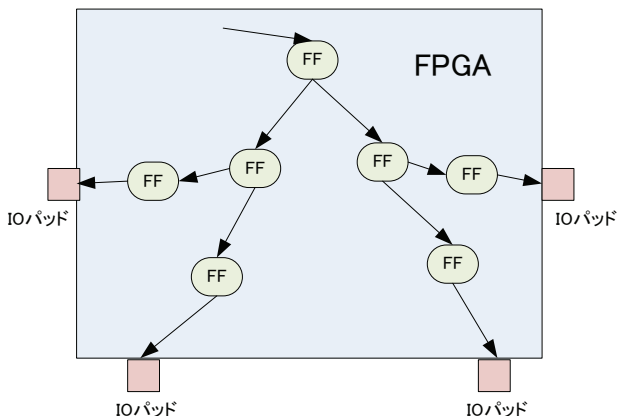


図 12. FF のツリー状の複製をした場合

され、それぞれの配線が図 10 の FF の3段目から IO パッドまでの配線より短くなる。すべての配線が短くなるとクリティカルパスの遅延が減少し、最大動作周波数を高くすることができる。このテクニックは、以前 Virtex-4 の DDR2-400 の DDR2 SDRAM コントローラを作る際に使用した。これは、同じ信号をそれぞれの位置が離れた IO パッドに出力する場合に有効なテクニックであるが、今回は出力する IO パッドが近距離でまとまっていたので、このテクニックを使わないでもタイミング制約を満足することができた。

いよいよ、シミュレーションの動作も問題なしとなって、実機による動作を確認するところまで持っていくことができた。最初に動作を確認してみると動作しなかった。いろいろトラブルシュートしてみたが、Read タイミングを出力する SD_LOOP_OUT に出力する部分のタイミング制約がなく、最終段の FF が IOB にマップされていないことがわかった。このトラブルを修正するとコントローラは動作した。

7. まとめ

Spartan3A Starter Kit に搭載されている FPGA 上に DDR2 SDRAM のコントローラを実装した。DDR2 SDRAM コントローラの動作周波数は 150 MHz で、DDR2-300 動作となっている。バーストテスト回路において 30 分間ランダムなデータを Write して、Read したデータを元のデータと比較したところエラーは発生しなかった。これは、データ転送の効率を 50% とすると、1.44 TBytes のデータを読み書きしてエラーを調べたことになる。

今後は DDR3 SDRAM のコントローラの実現にも挑戦していきたい。

8. 謝辞

シミュレーション用 DDR SDRAM バーストテストコードを提供いただいた菅原システムズの菅原孝幸様に深く感謝いたします。DDR SDRAM バーストテストコードを DDR2 SDRAM バーストテスト回路に変更して、DDR2 SDRAM コントローラをテストすることができました。

参考文献

- [1] 小野 雅晃, EDO-DRAM 制御モジュールの実現, 筑波大学技術報告 No.19 (1999) 23-30
- [2] フリー百科事典『ウィキペディア (Wikipedia)』
Dynamic Random Access Memory
http://ja.wikipedia.org/wiki/Dynamic_Random_Access_Memory
- [3] マイコミジャーナル、メモリ技術解説(1) メモリの基本、SRAM/DRAM
<http://journal.mycom.co.jp/news/2002/09/05/09.html>
- [4] SDRAM の使い方, ユーザーズマニュアル, Document No.J0123N10 (Ver. 1.0), Elpida Memory, Inc. (2001)
- [5] DDR2 SDRAM データシート Rev.N 1/09 EN, Micron Technology, Inc.
- [6] 高速ボード・レイアウト・ガイド 2003 年 9 月 ver. 1.1, Altera Corporation
http://www.altera.co.jp/literature/an/an224_j.pdf
- [7] Synthesizable 400 Mb/s DDR SDRAM Controller, XAPP253 (v2.3) June 1 2004, Xilinx Inc.
- [8] FPGA の部屋まとめサイト、Spartan3A Starter Kit
http://marsee101.web.fc2.com/spartan3a_starter_kit.html
- [9] Steve Kilts, Advanced FPGA Design: Architecture, Implementation, and Optimization, Wiley-IEEE Press (2007) 92-97

Implementation of a DDR2 SDRAM controller using the Spartan 3A Starter Kit

Masaaki Ono

Academic Service Office for Systems and Information Engineering, University of Tsukuba,
1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573 Japan

A double data rate 2 (DDR2) synchronous DRAM (SDRAM) controller was implemented on field-programmable gate arrays (FPGAs) provided by the Spartan 3A Starter Kit. The Spartan 3A Starter Kit includes the Spartan 3A FPGA device from Xilinx and DDR2 SDRAM from Micron Technology. A DDR2 SDRAM controller is an interface circuit for use of DDR2 SDRAM. The DDR2 SDRAM controller had an operating frequency of 150 MHz, and data were sampled at both the rising and falling edges of the clock. This format is typically known as DDR2-300. The DDR2 SDRAM used had a data width of 16 bits, so the maximum data transfer rate was 600 MBytes/sec. The DQS (Data Strobe) signal served as the DDR2 SDRAM's receive clock for Read data. This technique is known as source-synchronous clocking.

Keywords: FPGA; DDR2 SDRAM