



# 続・量産時でも不具合もなく動く DDRメモリ・コントローラを設計する

井倉将実

実機デバッグ編

前回(本誌2003年11月号, pp.124-135)は、机上の設計時における問題点を説明した。今回は実機でDDR-333(1ピン当たりの最大データ転送速度が333MbpsのDDR SDRAM)の検証を行う。DDR-333のデータ転送速度を実現するには、クロックの制約や、適切なI/Oインターフェースの定義、基板の設計などに注意する必要がある。(編集部)

前回は、FPGA(米国Xilinx社のVirtex-II Pro)を使ってDDR(double data rate)SDRAMコントローラの回路を設計する際に、高速な動作に対応するためにはどのような手法や概念が必要になるのかを説明しました。また、シミュレーションを通じて筆者が実際に設計した際に苦慮した点(特に、受信DQS信号の扱い)と、それを解消するための手法も紹介しました。

前回設計したDDR SDRAMコントローラをPCI-X基板に実装したので、今回は実機動作における苦労話を紹介してみたいと思います(写真1)。

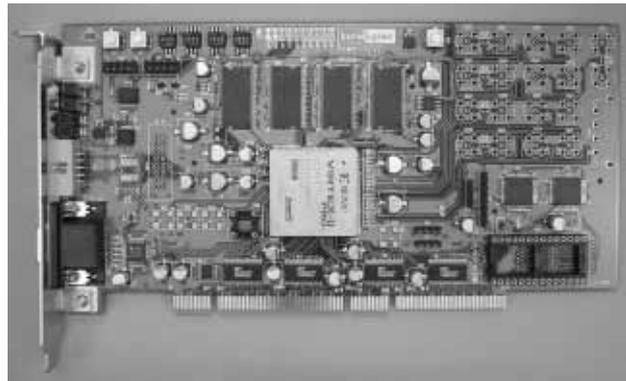
前回の記事では、DDR SDRAMの動作を理解しやすく

するため、必要最低限の回路やクロック駆動回路しか取り入れませんでした。今回は、外部にCPUコア「PowerPC 405」やPCI-Xバス・コントローラを搭載するために、DDR SDRAMにかかわる部分の回路構成を若干変更しています。

## ●システムの中の共有メモリとして使用する

表1に、前回紹介したDDR SDRAMコントローラを搭載したPCI-X基板の仕様を示します。表2と表3には、DDR SDRAMの主な信号名とコマンド・テーブルを示します。

前回はDDR SDRAMコントローラ単体の設計についてお話ししましたが、このときはシステム全体での使い勝手は考慮されておらず、単にDDR-333を駆動させることに全力を注いでいました。しかし、今回のDDR SDRAMは、共有メモリとしてPCI-XバスとCPUの両方からアクセスされることとなります。このため、PCI-XバスとCPUという二つのバス・システムに対して適切に動作するDDR



〔写真1〕DDR-333を搭載したPCI-Xボード

〔表1〕PCI-X基板の仕様

|             |  |
|-------------|--|
| 採用メモリ       | DDR-333(バス・クロック周波数は166MHz)   |
| メモリ容量など     | 256Mバイト, 64ビット幅, CASレイテンシ(CL)=2.5, バースト長=8<br>リフレッシュ周期=約7.8 $\mu$ s/1回     |
| DDRアクセス・マスタ | 2系統<br>●PowerPC405-PLB(プロセッサ・ローカル・バス)<br>●64ビットPCI-Xバス・ターゲット・イニシエータ・コントローラ |
| アービトレーション方式 | DDR SDRAMコントローラ内にバス・アービタを内蔵  |
| CPUバス仕様     | 64ビットPLBバス<br>最大64ビット4ワード・バースト, リード・プリフェッチ機能付き                             |
| PCI-X側バス仕様  | 64ビット独自バス<br>最大64ビット8ワード・バースト, ライトバック・バッファ/リード・プリフェッチ機能付き                  |
| 動作クロック      | DDR SDRAM系は166MHz(両エッジを使用)<br>CPU系は120MHz<br>PCI-X系は66MHz~133MHz(可変)       |

## コラム：拡張モード・レジスタへのアクセスについて

前回の記事の中で、「拡張モード・レジスタは使わないからアクセスしなくてもよい」という1文を入れました。しかし、そのあと米国 Micron Technology社のDDR-400を購入してテストした際、拡張モード・レジスタへのアクセスによってDLLをイネーブルしなければならぬことがわかりました。

拡張モード・レジスタにアクセスせず、モード・レジスタの中の「Reset-DLL」(メモリ・アドレス[8])を‘1’にセットして、LMR(ロード・モード・レジスタ)コマンドを発行した場合、DDR-200などではデータが出力されており、筆者は動いていると判断していました。

ですが、DDR-333などの高クロックにしたところ、バス・クロックのエッジにほとんど同期したDQSが出てくるはずなのに、その位相がずれてデータの取り込みが行えないことがわかりました。

この原因を追究していくと、DLLが動いていないということがわかりました。DDR-200のバス・クロックは100MHz、つまり1サイクルは10nsなので、5nsの遅延でDQSが出ており、このため実際にはCL=2.0なのですがCL=2.5で動いているように見えるのです。それがDDR-333やDDR-400になると、1サイクルが6nsなので、ほと

んど1サイクル分DQSの出力が遅延するため、データが取り込めなかったというわけです。

この問題について、DLLがうまく機能せず、バス・クロックだけで動いているものと想定して、ために拡張モード・レジスタをアクセスしてみました。その結果、今度はバス・クロックと同期してDQS信号が出てくるようになりました。

当初からモード・レジスタの「Reset-DLL」を行わなければデータは出てこなかったのですが、筆者は「DLLはもともと有効になっているものだ」と勘違いしていました。拡張モード・レジスタへのアクセスがなくても、ほかのDDR SDRAMでは特に不つごうはなかったもので、無視していたわけです。

これによって、モード・レジスタ設定では、「プリチャージ」、「拡張モード・レジスタ設定」、「モード・レジスタ設定」、「プリチャージ」と四つのコマンドを発行するように変更しました。

読者のみなさんに対して混乱を招くような記事であったことをおわびし、ここに訂正いたします。

SDRAMコントローラを設計する必要があります。

以下に、今回改良した点について説明します。

### ●FPGAの内部メモリ構成を利用した同期化を行う

CPUとPCI-Xバスから共通して見えるDDR SDRAMコントローラを構築するため、シェアード(共有)アクセスを可能にする回路を入れました。この回路の主な機能は、非同期でアクセスが来る外部要求に対してDDR SDRAMコントローラ側で同期化することです。

DDR SDRAMコントローラは166MHzクロックで駆動

されます。これに対してCPUやPCI-Xバスは120MHzや66MHzなどという別クロックで駆動されます。つまり、DDR SDRAMコントローラと外部システムは非同期で動くこととなります。そのため、同期化の手段が必要となるわけです。

非同期で駆動するコントローラの間を接続する方法として、通常、アクセスの開始や終了などの制御信号系ではフリップフロップを使用します。また、データ・バスやアドレス・バスも同期化するのが一般的です。

しかし、ここではFPGAならではの手法を採用しました。これは、書き込み側でいったんバッファにデータを取り込

〔表2〕DDR SDRAMの重要な信号

| 信号名             | 機能   |
|-----------------|--|
| CKE             | クロック・イネーブル。DDR SDRAM内部の回路に対してクロックが有効であることを指示する                 |
| CK/CK#          | 基準クロック。2相の差動クロックになっている   |
| A <sub>n</sub>  | アドレス・バス。メモリの読み書き時に行アドレスと列アドレスを与える                              |
| BA <sub>n</sub> | バンク・アドレス。要求されたオペレーションがどのメモリ・ブロックに対して発行されるかを指定する                |
| CS#             | チップ・セレクト。DDR SDRAMのセレクト信号                                      |
| RAS#            | 行アドレス・ストロープおよびバンク・セレクト   |
| CAS#            | 列アドレス・ストロープおよびコマンド・セレクト  |
| WE#             | ライト・イネーブル。CAS#とこの信号がアサートされていれば書き込みサイクル。この信号がネゲートされていれば読み出しサイクル |
| DQM             | データ・マスク制御。データを書き込む際に使用される                                      |
| DQ <sub>s</sub> | データ・バス。データの入出力を行う  |
| DQS             | データ・ストロープ信号。入出力データのタイミングを指示する                                  |

〔表3〕DDR SDRAMコマンド・テーブル

| コマンド名                               | CS# | RAS# | CAS# | WE# | アドレス  |
|-------------------------------------|-----|------|------|-----|-------|
| NOP(選択なし)                           | H   | X    | X    | X   | X     |
| NOP(操作なし)                           | L   | H    | H    | H   | X     |
| アクティブ(バンクの選択と行アドレスへのアクセス)           | L   | L    | H    | H   | バンク/行 |
| 読み出し(バンクの選択と列アドレスの選択、および読み出しバースト開始) | L   | H    | L    | H   | バンク/列 |
| 書き込み(バンクの選択と列アドレスの選択、および書き込みバースト開始) | L   | H    | L    | L   | バンク/列 |
| バースト終了                              | L   | H    | H    | L   | X     |
| プリチャージ(バンク内の行アドレスをクローズ)             | L   | L    | H    | L   | コード   |
| オート・リフレッシュまたはセルフ・リフレッシュ             | L   | L    | L    | H   | X     |
| モード・レジスタ設定                          | L   | L    | L    | L   | OPコード |