

PCI-X 2.0の 仕様概要と設計方針

後編

——I/Oセルの設計とパッケージング

Stillman Gates

PCI-X 2.0は従来のPCI-X 1.0に比べて大幅に性能が向上したが、これを実現するために設計者は多くの課題に取り組む必要がある。ここでは、I/Oセルやパッケージング、ECC(エラー訂正コード)についてその設計指針を解説する。(編集部)

PCI-X 2.0は、信号要件の異なる3種類のI/Oセルのカテゴリを定義しています^{注1}。そのため、設計者は信号I/Oセルの設計を見直したり、変更しなければならなくなる可能性が高いでしょう。

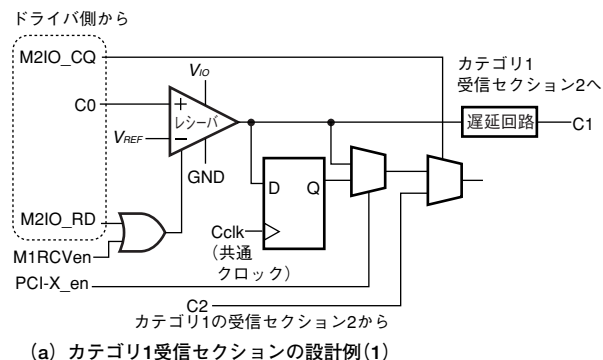
●動作モードによって使用するI/Oセルは異なる

以下にそれら3種類のI/Oセルについて説明します。

1) カテゴリ1 I/O セル

「カテゴリ1 I/Oセル」は、同期データ転送モードと共通クロック・データ転送モードの両方で動作する信号(アドレス/データ用のAD、C/BE#およびECC)に適用されます。

図1に、カテゴリ1 I/Oセルの受信セクション1の設計



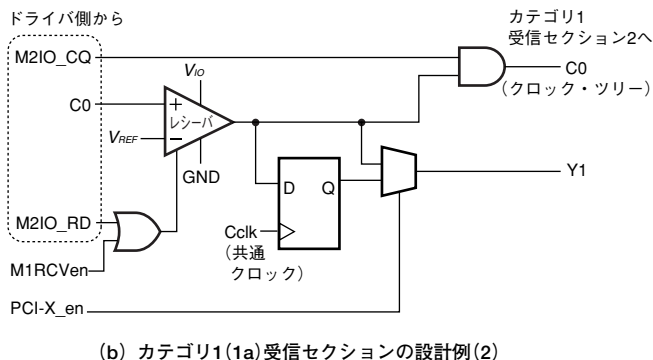
〔図1〕 カテゴリ1 I/Oセルの受信セクション1

カテゴリ1 I/Oセルを設計するうえでは、(a)と(b)の二つのタイプを用意しておくとうい。

例を二つ示します。片方の回路だけを使ってLSIを開発することもできますが、両方のタイプのI/Oセルを用意しておいたほうが、つごうが良いようです。

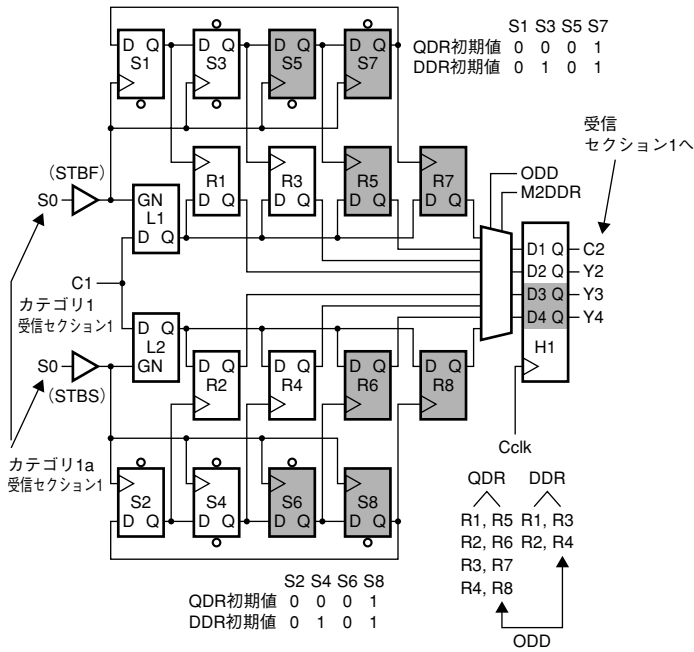
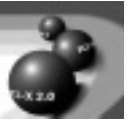
図2に、カテゴリ1 I/Oセルの受信セクション2の設計例を示します。図2(a)のS1～S8は、二つの循環型シフト・レジスタを構成しています。このシフト・レジスタに連動して、ラッチLnビットおよびレジスタRnビットのクロックが生成されます。図2(a)の上部のブロックではSTBF(第1ストロブ信号)によって、下部のブロックではSTBS(第2ストロブ信号)によってそれぞれクロックが生成されます。共通クロック・モードでは、受け取ったデータがH1に格納されます。Rnのレジスタには白いブロック(R1～R4)と灰色のブロック(R5～R8)がありますが、これらは交互に動作します。DDR(double data rate)しかサポートされていない場合、灰色のブロック(R5～R8)は使用しません。

図2(b)は、カテゴリ1 I/Oセルの受信セクション2に要



(b) カテゴリ1(1a)受信セクションの設計例(2)

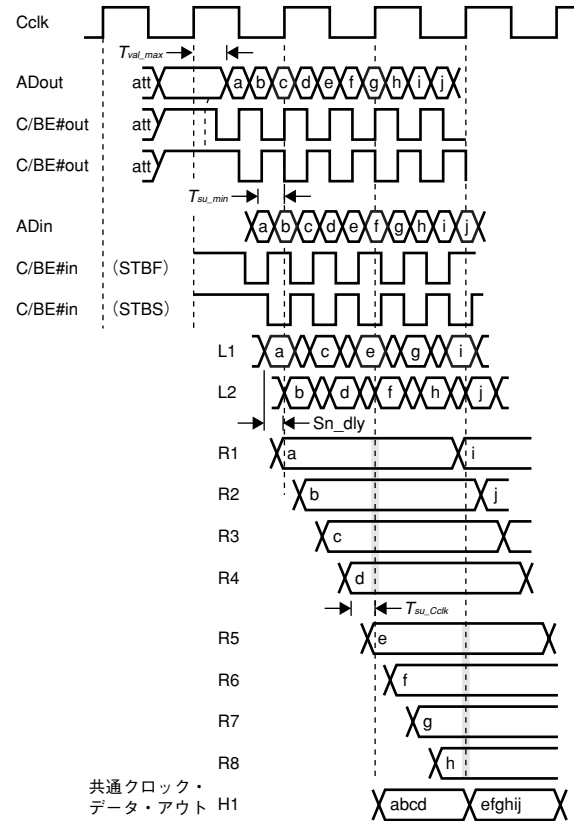
注1：2002年7月、PCI-SIGによりPCI-X 2.0の最終仕様が発表された。なお、本稿はPCI-X 2.0ドラフト案に基づいて執筆されたものであるため、設計するには必ず最終仕様を参照してほしい。



出典：PCI-X Work Group, Nov.15, 2001, Face-to-Face, mode2 QDR receiver slides

〔図2〕 カテゴリ1 I/O セルの受信セクション2

S1～S8は、二つの循環型シフト・レジスタを構成している。このシフト・レジスタに連動して、LnビットおよびRnビットのクロックが生成される。Rnのレジスタは、R1～R4のブロックとR5～R7のブロックに分かれ、それぞれが交互に動作する。ただし、DDRしかサポートしていない場合は、R5～R7は使用しない。



求されるタイミングを表しています。PCI-X 2.0の仕様では、このタイミングを実現するためにクロックを追加し、受信セクションにバリエーションを持たせている点に注意してください。

2) カテゴリ2 I/O セル

「カテゴリ2 I/Oセル」は、共通クロック・データ転送モードでのみ動作する信号（プロトコル制御用信号FRAME#、IRDY#など）に適用されます。カテゴリ2 I/Oセルの例として、クロック・レシーバが挙げられます。クロック・レシーバは単なる受信回路であり、駆動回路は含まれてい

せん。また、クロック・レシーバのI/Oセルには、ノイズ除去のための特別な処理が必要です。

図3(a)に、PLLのジッタを最小にするクロック・レシーバ・セルの設計例を示します。クロック・レシーバ・セルでは、外部のクロック信号を内部のコアに送る際にジッタが発生しないようにしなければなりません。ジッタは、セルの電源やグラウンドのノイズが原因となって発生します。この対策としては、電源とグラウンドの外部との接続を独立させ、ノイズを抑制するフィルタを入れます。

また、目的に応じて「to PLL」と「bypass PLL」の二つの

〔図3〕
クロック・レシーバ・セル

PLLのジッタを最小にするクロック・レシーバ・セルの設計例を示す。(a)には、目的に応じて「to PLL」と「bypass PLL」を使い分ける二つの出力を示している。

