

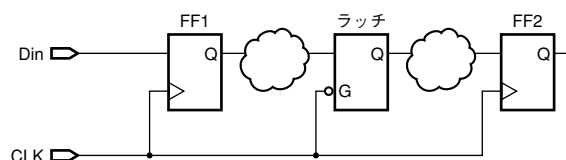
テスト容易化設計(DFT : design for testability)において、ラッチの置きかたによっては、DFTの対策が難しくなる。また、バスなどの双方向ラインを使用する際には3ステート・バッファが必要となるが、これはDFTの観点では不適切な回路となる。今回は、こうしたラッチと双方向バッファの取り扱いについて述べる。(編集部)

ラッチと双方向バッファは、テスト容易化設計(DFT : design for testability)にとって、もっともつごうの悪い回路です。現在の設計では、少なくともLSIの故障テストを考えた場合、双方向バッファを利用しないほうが賢明です。ラッチについては、限定した使用にとどめたほうがよいでしょう。

#### ●ラッチは反転動作のみで1段だけ使用

ラッチについては、反転動作のみで1段だけ使用する分には、これから紹介するような諸問題が発生せず、安全に利用できます。図1に示すように、反転動作のラッチでは、クロックが“L”のとき、データがすり抜けます。したがって、次段のFF(フリップフロップ)は、次のCLKの立ち上がりエッジでD1の値を取り込むことができます。クロックの立ち上がりエッジだけで見れば、反転動作のラッチが存在しなくても論理はまったく変化しません。そのため、

〔図1〕  
ラッチは反転動作のみで1段使用する  
だけなら問題にならない  
ラッチを反転動作で1段使用すると、  
CLKが立ち下がるまでは値を保持し、  
立ち下がり後に値が変化する。



(a) 反転動作のラッチを1段使用した回路

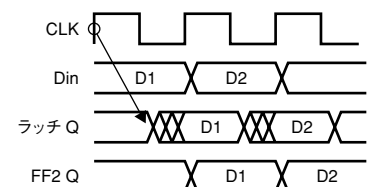
DFTのために特別な対策を施さなくても、FFとFFの間の組み合わせ回路の故障を検出できます。むしろ、反転動作のラッチは、積極的に利用したほうがよい面も多々あります。

#### ●同期RAMのホールド時間の保証

同期RAMで生じたホールド時間違反は、論理合成ツールでホールド時間を保証するコマンドを実行すれば問題を解決できます。ただし、同期RAMのホールド時間が1ns以上のこともあります。すると、ホールド時間を保証するために多くのバッファが挿入されてしまいます(図2)。0.18  $\mu\text{m}$  や0.13  $\mu\text{m}$  などのプロセスでは、バッファ1段分の遅延は、10psから20psくらいしかありません。もし、RAMのホールド時間が1nsだとすると、30個以上のバッファが

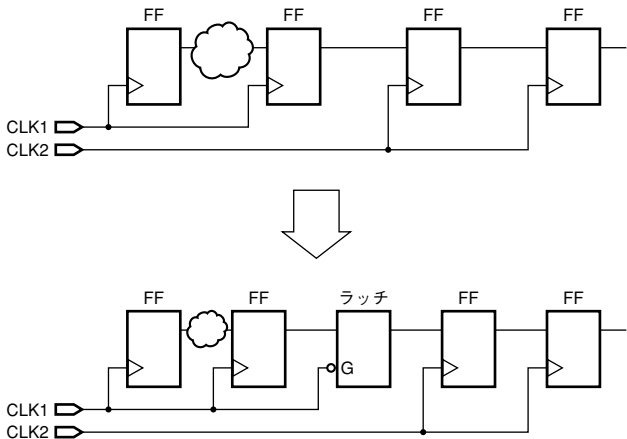
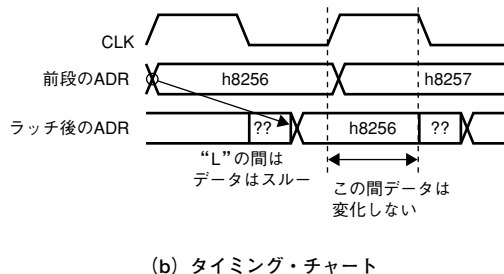
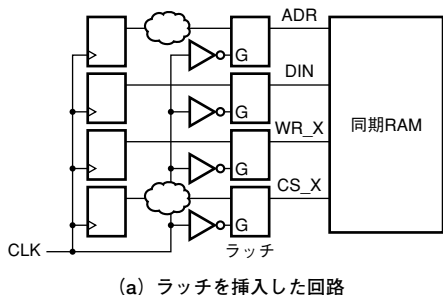


〔図2〕同期RAMは、ホールド時間の保証のため多量のバッファが挿入される



(b) タイミング・チャート

〔図3〕  
同期RAMのホールド時間を保証するために反転動作のラッチを挿入  
同期RAMの前段に反転動作のラッチを挿入することで、ホールド時間の保証の心配がなくなる。



〔図4〕非同期クロック間のメタステーブル対策  
非同期クロックの間でも、スキャン回路の挿入方法によってはテスト・モード時に同一クロックで動作させることがある。この場合、反転動作のラッチを挿入しておけばホールド時間の保証の心配はなくなる。

追加されることとなります。もちろん、回路規模の増加が気にならないのであれば、論理合成ツールでホールド時間の保証を実現したほうが手間がかりません。

もし、回路規模の増加が気になるのであれば、反転クロックのラッチを使用する方法もあります。図3のように同期RAMの前に反転クロックのラッチを使用すると、クロックが“H”の期間、この信号の値が変化することはありません。同期RAMのホールド時間がクロック周期の半分以下であれば、ホールド時間の問題はなくなります。

クロックが“H”の期間、反転クロックのラッチのゲートは開いています。もし、ラッチの前段に多くの論理ゲートが存在したとしても、データはラッチをすり抜けます。このため、このパスの遅延は、1周期分からラッチの遅延分を引いた分だけですみます。すなわち、ラッチを挿入することで、同期RAMに到達するまでの遅延値をあまり意識することなく、ホールド時間の保証の問題を解決できます。

同期RAMに限らず、ハード・マクロのインターフェー

ス部にラッチを利用することもあります。組み合わせ回路の段数が多くなければ、このような反転動作のラッチを反転動作のFFに置き換えることも可能です。しかし、前回説明したように反転動作のFFは、レイアウト時にスキャン回路のリオーダを行うとつづごうが生じます。ラッチを利用した場合、このような問題は発生しません。

●非同期クロック間の転送では反転ラッチを挿入

非同期クロック間は非同期動作なので、通常動作時にはホールド時間を保証する必要がありません。しかし、テスト・モード時に同一クロックで動作させることもあります。

最新の自動テスト・パターン生成(ATPG: automatic test pattern generation)ツールでは、非同期クロック間を自動で認識し、それぞれのクロックの立ち上がりのタイミングをずらすことで、上記の動作に対応するものもあります。この場合も、特に非同期クロック間のホールド時間の保証を心配する必要はありません。100万ゲート規模のLSIでは、通常4~32本のスキャン・チェーン(スキャン・パス)を作成することになります。もし、クロック・ラインの数が少なければ、クロック・ラインごとに別のスキャン・チェーンを作成する方法を取ります。このとき、非同期クロックに対応したATPGツールを使用すれば、DFT上の問題は発生しません。しかし、クロック・ラインの数が多く大規模LSIの設計では、クロック・ラインごとにスキャン・チェーンを接続していくことが困難になります。この場合、テスト・モード時にクロックを切り替えて、非同期クロックの間を同一クロックで動作させることもあります。

もし、非同期クロックを同一のクロックとして動作させるのならば、二つのクロック間のホールド時間の保証を行う必要が生じます。非同期クロックに対して、それぞれ別のクロック・ツリー・シンセシスを実行すると、非同期ク