

ASIC 設計経験者が陥りやすい 誤った FPGA ローパワー対策と正しい手法

Junichi Chonan, Calypto Div., Siemens EDA, Japan, junichi_chonan@mentor.com

アブストラクト—国際的な SDGs に見られるように、持続可能な開発目標が求められる情勢において、半導体における消費電力対策は喫緊の課題である。ローパワー対策というとクロックゲーティングや、それを基本とした対策が広く知られているが、同じ対策を FPGA に適用しても、その異なる構造によってむしろ消費電力増となることが多い。ASIC から FPGA に置き換えるプロジェクトや、戦略的に FPGA を使用するプロジェクトでは、FPGA 固有のローパワー対策を適用することで、世界的に求められるエネルギー要件への対応となり得る。

キーワード— *Low Power; ASIC; FPGA; Clock Gating, Redundant Reset,*

I. はじめに

近年、地球規模で見られる温暖化などの環境変化への対策として、脱炭素化に伴うエネルギー政策の見直しや、国連サミットで採択された SDGs (持続可能な開発目標) に掲げられるゴール達成が叫ばれている。半導体デバイスの開発も例外ではなく、IoT や Edge デバイスとクラウド、つながる自動車や自動運転など、かつてない規模であらゆる分野で使われており、本来達成しなくてはならない機能と性能の実現に加えて省電力化は必須要件である。また消費電力は熱の問題と密接に関係している。半導体デバイスがある一定時間以上に熱を帯びてくると、その面積の極小性から冷却が急激に困難になる。また積層に伴ってパッケージ内部に熱がこもってしまうことも問題として顕在化している。熱対策としてヒートシンクを装備すれば小型化を阻害し、ファンを装備すれば新たな電力を消費する。このことから、ある一定時間以上に熱を上げないように消費電力対策を考慮しながら設計する事 = ローパワー設計が急務となっている。本稿では半導体におけるローパワー化の機会全体を俯瞰し、その中から RTL におけるローパワー設計のメソッドロジについて議論する。

II. 求められるローパワー設計の再考

図 1. に示す逆三角形は、High Level 設計、RTL 設計、Gate Level 設計のそれぞれの設計フェーズにおいて、どれくらいパワー削減の機会や効果が期待できるかを示したものである。[1] この図は決して新しいものではなく、過去 20 年にわたりさまざまな論文やカンファレンス、書籍、インターネット上のメディア記事などで使用されてきた。逆ピラミッド型になっていることから分かるように、High Level 設計の段階では C/C++ 高位合成などの際に、アーキテクチャの選択やリソース共有などの大きな選択肢が検討できるため、ローパワー対策としてはより効果的である。

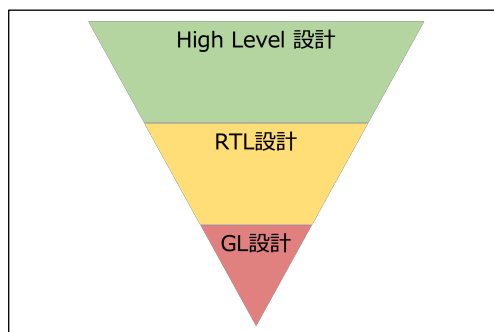


図 1. ローパワー対策によって期待される消費電力削減効果

それ続く RTL 設計や GL 設計などの抽象度におけるローパワー対策は、High Level 設計に比較すればその効果は下がるものの、それでもある一定の消費電力削減効果が期待できる。そして前出のとおり SDGs など掲げられる目標を達成するためには、High Level 設計、RTL 設計、GL 設計のすべてのフェーズにおけるローパワー対策が必要である。また高位合成を採用していないプロジェクトや、過去の RTL 設計資産を多く再利用するプロジェクトでは、RTL 設計フェーズにおけるローパワー対策が最も大きな効果が期待される領域となる。ここで表 1 に、それぞれの設計フェーズにおけるローパワー対策の特徴を、そのポイントと期待される効果によって示す。

表 I. 設計フェーズごとのローパワー対策の特徴[図表番号+段落前]

| 設計フェーズ | 考慮すべきローパワー対策 | 期待される効果 |
|---------------|--------------|--|
| High Level 設計 | 使用リソースの考慮 | 適切なリソース配分や共有によりローパワー化を実現するハードウェアの最小構成を得る |
| RTL 設計 | 無駄な動作の抑制 | 不要なトグルを適切なリソースによって抑制し最小のダイナミックパワーで回路を動作させる |
| GL 設計 | 電源分離 | 必要最低限の電力で回路を動作させる |

次にローパワー対策の鉄則とも言うべき対策方法について、設計フェーズと関連づけて考察する。

A. 不要な論理演算や論理回路の削除

論理演算器の削除や集約、共有などによって不要な演算器を削除したり共有する手法は、High Level 設計が最も検討や評価がしやすいフェーズとなる。高位合成に先駆けて行われる実装探索では制約によって回路の性能、消費電力、面積の選択肢が示されるため、ローパワーの観点からも回路検討がなされる。

しかし高位合成を使用しないプロジェクトであっても、論理演算器を削除できるかどうか、他の論理演算器と共有できるかどうかを検討することは重要である。また再利用する RTL 設計資産によっては、オリジナルの開発者がいない、明確な仕様書が存在しないなどの理由で、使用することのないモードまで含めて再利用してしまう可能性も捨てきれない。このような不要な論理回路は、RTL 設計フェーズにおけるコードカバレッジ測定で活性化されないコードとして特定されることがある。またフォーマル検証エンジンによるリーチャビリティ解析によって到達しないコードの特定から発覚することもある。

B. 信号の不要なトグルの削除

RTL 設計のフェーズにおいて、信号ごとに不要となるトグルをゲーティングによって削除する対策である。トグルにより電位が上下することで電流が引き込まれるものの、不要なトグルとして論理的な影響につながらないものである。一見地味ではあるが、ポイントを絞り込むことで有効な対策となる。

C. 高周波数かつ高ファンアウトの信号の確認およびグリッチ発生箇所の確認

これは上記 B. の補足的項目ではあるが、RTL 設計フェーズや GL フェーズにおいて高周波数信号と、そのファンアウトを確認することは重要である。またグリッチを確認することも重要である。このような信号におけるトグルをゲーティングする際に、パワー最適化を目的とした回路変更によって問題が発生しないことを確認することも重要である。

III. RTL 設計におけるクロックゲーティング

高周波数かつ高ファンアウトの信号として共通するのはクロック信号である。クロックが一番速いデータの切り替え周波数に合わせて高周波数で動作する必要がある。ただしデータ側は常に同じ間隔で切り替わるわけではなく、より高い周波数で一定の周波数で動作するクロックには無駄なトグルも当然多い。これをゲーティングするローパワー対策の手法は一般的にはクロックゲーティングと呼ばれる。また RTL 設計を論理合成し GL ネットリストを生成しても、レジスタの位置や数は変わらないため、より抽象度の高い RTL で積極的にクロックゲーティング対策を行うのが王道となる。

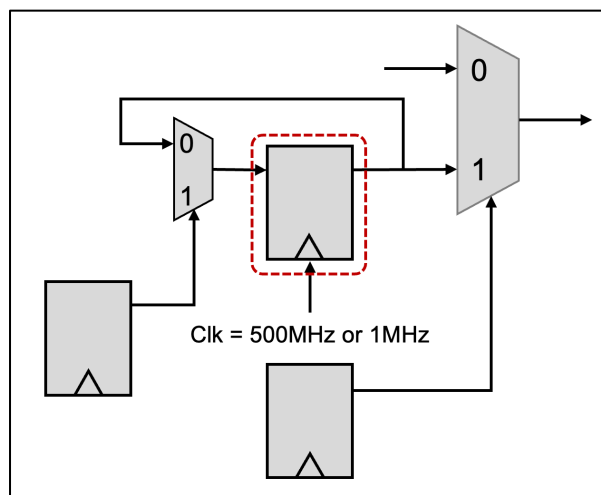


図 2. クロックゲーティングの対象候補

図 2.において赤い点線で囲まれたレジスタが最適化の対象候補である。もちろんすべてのレジスタを最適化できれば理想的だが、優先順位をつけて対策しなくてはならない場合には、レジスタを駆動するクロック周波数で判断することになる。

クロックゲーティングはその名が指し示すとおり、クロックをゲーティングする対策方法である。図 3.に ASIC などで行われるクロックゲーティング対策の概念を示す。

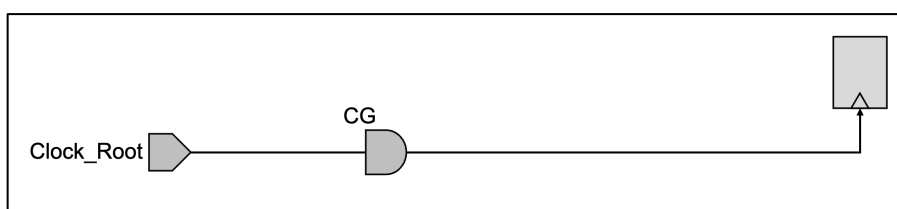


図 3. ASIC におけるクロックゲーティングの概念

ASIC では配置配線の段階で Clock Tree Synthesis などのツールによりクロック設計を行い、RTL 記述に挿入されたクロックゲーティングセル (CG セル) も回路構成に合わせて配置することができる。この際に Clock Root から CG セルまでの配線は制御することができないため、この部分をできるだけ短くする。そして CG セルが駆動する配線を長くすることで、大きな配線容量を持つ配線部分のトグルを抑え、ローパワー対策の効果を大きくしようとする。

一方で FPGA では配置配線については色々なリソースが領域として割り当てられているため、クロックゲーティングを行おうとすると図 4.に示すようにコンフィギュレーション・ブロック内でクロックイネーブルが付いたレジスタを使用することになる。ASIC のように大きな配線容量を持つ配線上のトグルを制御することにはならない。

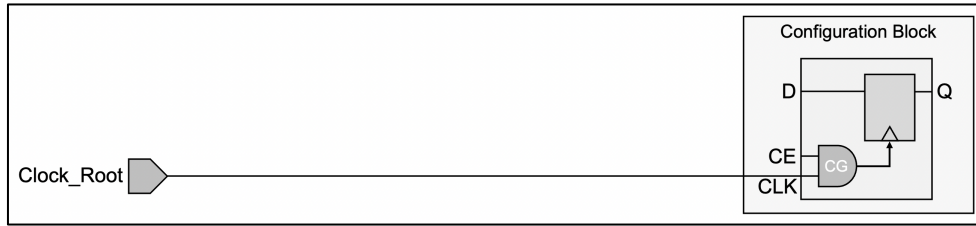


図 4. FPGA におけるクロックゲーティングの概念

一般的には RTL 記述の段階でクロックゲーティング対策を行うが、ASIC 対象として一般的なローパワー対策を FPGA に適用した場合に、果たして効果があるのだろうか。次のセクションで2つの具体的な対策例について考察する。

IV. FPGA におけるクロックゲーティング対策の考察

まず ASIC では良く使われるローパワー対策の1つである XOR ゲーティングという手法を FPGA に適用するケースについて考察する。XOR ゲーティングはクロック周波数とデータ周波数の差分が大きい際に有効な最適化手法である。

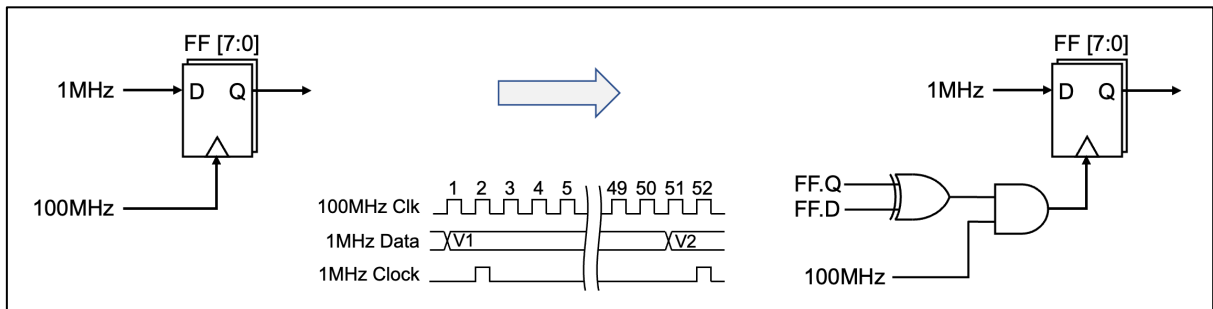


図 5. XOR ゲーティング

図 5.の左のレジスタのように 100MHz クロックで駆動され、そのデータ入力信号が 1MHz である場合について考えてみる。これはデータ対クロックの周波数比率が 1/100 であり、クロック 50 サイクル中に 1 度しかデータが変化せず、残る 49 サイクルは同じデータを無駄打ちしていることになる。ここで消費される電力のほとんどは、配線長に伴う配線容量があるクロック上のトグルであり、レジスタへの入力データが変化していないため内部でトグルが発生して電力が消費されることはない。

そこでデータの変化を検出し、その変化に合わせた遅いクロックを作り出すためにレジスタ入力信号とレジスタ出力信号の XOR を取り、それをゲーティング信号としてクロック供給するような回路を挿入している。これによりデータが変化していない状態でのトグルを抑えることができる。なおレジスタが図 5.のように 8 ビット幅のワードとして RTL 記述されている場合、合成後のネットリストでは XOR 回路はワード単位で変化点を検出するために、8 ビット分の XOR ツリー回路が推定される。

このローパワー対策の効果が得られるのは、データ周波数がクロック周波数に対して著しく低いケースである。仮にデータ周波数が 20MHz で、クロック周波数に対して 1/5 の関係であるとする、以下のような 2 つの懸念点が浮かび上がる。

1. 1 つ目の懸念は XOR を含むクロックゲーティング回路が 20MHz で動作し、挿入した回路そのものの消費電力が増えるというものである。

- 2つ目の懸念は挿入した XOR 回路の出力に発生するグリッチの悪影響である。XOR 回路はグリッチの発生確率は高く、そのグリッチを加味した周波数は 4 倍となることもある。XOR の動作周波数が 1MHz ならば 4MHz の周波数となったとしてもクロック 100MHz に対しては大幅なクロックゲーティング効果が期待できるが、20MHz で動作する XOR のグリッチを加味した周波数が 80MHz にも達すると、クロックゲーティングによる効果は期待できない。また XOR のゲーティング回路が XOR ツリーとなって実装されるため、さらに多くのグリッチが発生する可能性もある。デメリットの上記 1. と合わせると、かえって消費電力の増加につながる懸念がある。

ここまでは ASIC で良く使用される XOR ゲーティングによるローパワー対策の例であるが、FPGA の場合にはどうなるだろうか。FPGA では CG セルに相当する機能がコンフィギュレーションブロックのレジスタ内部に組込まれているため、クロックの配線長による消費電力の削減は極めて限定的である。むしろ XOR や XOR ツリーで構成されるゲーティング回路の挿入に伴って面積が増え、かつ配線の引き回しによる配線容量が増えるため、データ入力信号の周波数が十分に低い場合であったとしても、消費電力が増えてしまう傾向にある。

ASIC で良く使われるローパワー対策の 2 つ目の例はサーキュラーバッファへの置換えである。シフトレジスタなどのように構造的にレジスタのクロックをゲーティングすることができないブロックをサーキュラーバッファ回路に置換える対策である。この置換えの仕組みを図 6. に示す。

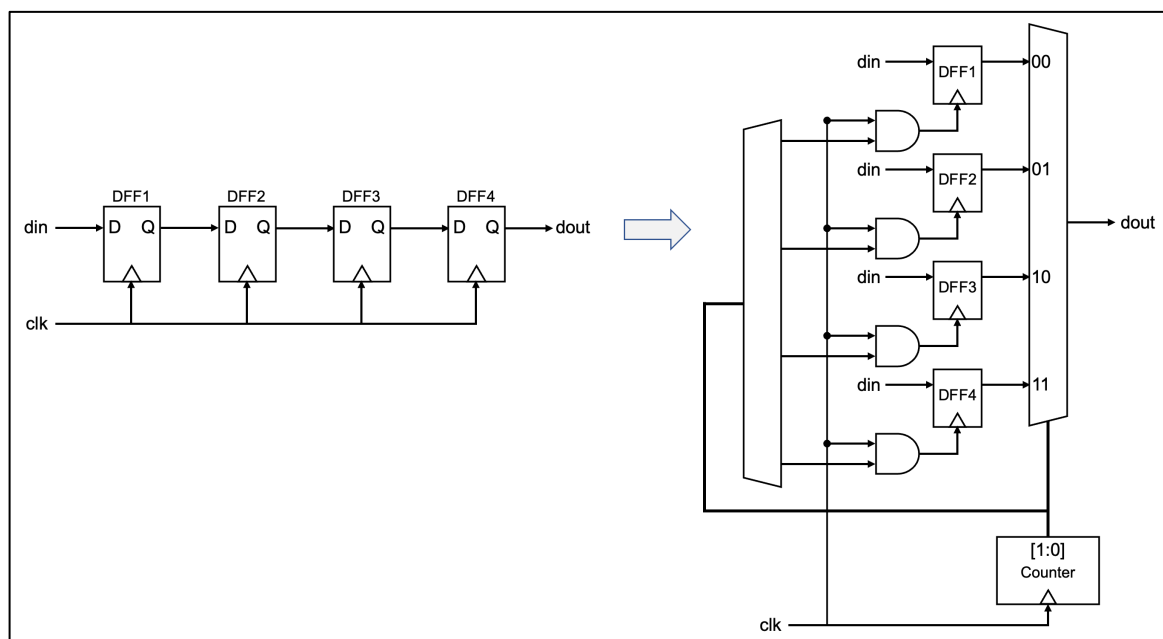


図 6. シフトレジスタからサーキュラーバッファへの置換え

シフトレジスタのクロックはゲーティングによって止めることはできないが、サーキュラーバッファに置換えることで、同時に動作させるレジスタの数を減らす事が可能となる。このサーキュラーバッファはシフト段数分のアドレスを設け、アドレスに該当のレジスタを動作させる。図 6. の例では 4 シフトのシフトレジスタを、0 から 3 まで 4 つのアドレスを持つサーキュラーバッファに置換えることになる。そしてその 1 つ 1 つのアドレスにアサインされているレジスタは、1/4 回分のみ動作すれば良い事になる。このためシフトする段数が深ければ深いだけ、大きなパワーの削減効果が見込める。

FPGA に対して、このようなローパワー対策は有効となるだろうか。FPGA には コンフィギュレーションブロックや演算スライスの中にラインバッファとして適応することが可能な小さな RAM が存

在している。シフトレジスタを RTL 記述すると、FPGA 実装においてはコンフィギュレーションブロック内のレジスタにはマッピングされず、ラインバッファ用の小さな RAM が割当てられる。つまりはシフトレジスタをトランジスタレベルで構成したようになる。

一方でサーキュラーバッファを RTL 記述すると、前出の小さな RAM には割当てられない。その理由は、各レジスタの直後に MUX が存在するためである。その結果、複数のコンフィギュレーションブロックのレジスタが使用されるようになり、かえって大きな構成となってしまう、ローパワー対策を行ったことで、かえってより多くの消費電力が必要となってしまう。

この2つの例から学ぶべきことは、デバイスの構成を考慮することなく、単に周波数削減を目的としたローパワー対策を講じても、かえって消費電力増につながる可能性があるということである。

V. FPGA デバイスをターゲットとした場合のローパワー対策

それでは FPGA デバイスをターゲットとした場合には、どのようなローパワー対策があるのだろうか。XILINX 社の資料を参照すると、レジスタやメモリブロックのイネーブルピンを制御することがローパワー対策となることが分かる。[2] 例えばレジスタセル-FDRE や Block RAM には Clock Enable ポート (CE) が存在している。ただしローパワー対策の考え方においては、いかにクロックラインのトグルを抑えるかではなく、クロック供給を抑制することでレジスタや Block Ram の動きをいかに長く止めることができるかが着眼点となる。

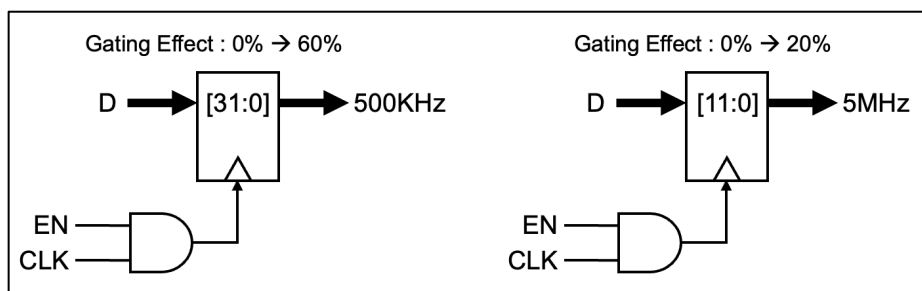


図 7. レジスタ出力ポイントにおけるゲーティング効果

図 7. にレジスタ出力ポイントにおける周波数、ゲーティング効果、レジスタのビット数が異なる 2 つの例を示している。左側のレジスタではゲーティング効果が 60%だとすると 500KHz が 200KHz となり削減効果は 300KHz 分あることが容易に計算できる。さらにレジスタのビット数は 32 ビットであるため、トータルの削減効果としては $300\text{KHz} \times 32\text{bit} = 9.6\text{MHzBit}$ という考え方が成立する。一方で右側のレジスタのゲーティング効果は 20%であるので 5MHz が 4MHz となり削減効果は 1MHz 分である。そしてレジスタのビット数は 12 ビットであるため、トータルの削減効果は $1\text{MHz} \times 12\text{bit} = 12\text{MHzBit}$ となる。このように ASIC における周波数削減はクロックに着目してゲーティングを行うが、FPGA の場合は出力ポートのデータ周波数に着目する。

レジスタやメモリの出力が、回路動作に直接関与しないタイミングで CE ポートをより長くディスエーブル化することができれば、レジスタ自身だけでなくレジスタ間の組合せ論理回路においてもローパワー化が見込める。例えば対象レジスタの後段に MUX 回路があるかどうかを確認し、その MUX がレジスタ以外の信号を選択している場合、その MUX 制御信号を用いてレジスタの CE ポートをディスエーブルにするなどの対策が考えられる。これは対象がメモリの場合でも同じである。

実際の FPGA 設計例や実験用の比較的小さなテストケースに対して、このローパワー対策を行う前と対策後の比較の一部を表 1. に示す。

表 1. 実 FPGA デザインにおける MUX ローパワー対策例

| Power(W) | Clocks | CLB Logic | MUX | | | Signals |
|---------------|--------|-----------|--------------|------------------|----------|---------|
| | | | LUT as Logic | LUT as Shift Reg | Register | |
| Original RTL | 0.206 | 0.161 | 0.114 | 0.027 | 0.015 | 0.237 |
| Optimized RTL | 0.211 | 0.160 | 0.113 | 0.027 | 0.015 | 0.214 |

このデータでは Register 部分の割合がそれほど多くはないため、Register における削減効果を確認することはできないが、LUT as Logic で示されるレジスタ間の組合せ論理部分についてはレジスタの出力ポートの周波数削減の効果が確認できる。

続いて Register と Block RAM を含む比較的小さなテストケースについてローパワー対策の前と後で比較したものが表 2.である。

表 2.小さな FPGA テストケースにおける MUX ローパワー対策例

| Type | Gated FF | Gated Mem | Number of | | | Power (W) | | | | | | |
|---------------|----------|-----------|-----------|------|------|-----------|--------|--------|--------|-------|-------|--------|
| | | | SRL16 | FDRE | BRAM | CLK | Reg | SRL | Comb | BRAM | Total | Static |
| Original RTL | 50 | 1 | 26 | 55 | 1 | 0.012 | <0.001 | <0.001 | <0.001 | 0.006 | 0.019 | 0.467 |
| Optimized RTL | 122 | 1 | 0 | 133 | 1 | 0.010 | <0.001 | <0.001 | <0.001 | 0.001 | 0.012 | 0.467 |

この比較でも Reg や SRL などの消費電力がもともと小さいため比較はできないが、1つ使われている BRAM、つまり Block RAM 部分においてローパワー対策の効果を確認することができる。

ここでクロックについて見てみると、表 1.のケースではクロックによる消費電力は増加しており、表 2.のケースでは減少している。そこでクロックによる消費電力が何に依存しているかを調べるために、非常に小さなテストケースにより確認したところ、表 3.のような結果が得られた。

表 3.クロック消費電力を調査するためのテストケース比較

| Type | User Gated | Tool Gated | # of FDRE | Clock Power | Register Power | Comb Power | Total Power | Static Power |
|-----------------|------------|------------|-----------|-------------|----------------|------------|-------------|--------------|
| Original RTL 1 | 0 | 0 | 132 | 0.015 | <0.001 | <0.001 | 0.488 | 0.467 |
| Optimized RTL 1 | 128 | 0 | 132 | 0.015 | <0.001 | <0.001 | 0.488 | 0.467 |
| Original RTL 2 | 0 | 0 | 132 | 0.015 | <0.001 | <0.001 | 0.459 | 0.467 |
| Optimized RTL 2 | 128 | 0 | 132 | 0.011 | <0.001 | <0.001 | 0.484 | 0.467 |

ここでは最適化による効果を見るために配置配線の制約を変えているが、それが何故クロックの消費電力に影響を及ぼしているかは別の Clock Regions というレポートを確認する必要がある。[3]にその Clock Regions レポートを 2 例、図 8.に示す。

図 8. Clock Regions のレポート比較

図 8.の左側では、132 個ある FF のうち 131 個は 1 つのクロックリージョンに、残る 1 個の FF は異なるクロックリージョンに配置されている。対して図 8.の右側では 132 個すべての FF が 1 つのクロックリージョンに収まっている。表 3.における Original RTL 1、Optimized RTL 1、Original RTL 2 の 3 つのケースのレポートは図 8.の左側、Optimized RTL 2 のレポートは図 8.の右側にあたる。つまりすべてのレジスタが複数のクロックリージョンに配置される場合にクロックラインの消費電力が上がり、逆にすべてのレジスタが 1 つのクロックリージョンに配置されれば、クロックラインの無駄な消費電力が抑えられる。これは配置配線の制約にも大きく依存するため、思ったようにコントロールできないが、少なくとも消費電力を分析する際の 1 つの価値ある指標である。

このようなことを踏まえて、ぜひお奨めしたいローパワー対策は、リダンダント・リセットの削除である。図 9.にリダンダントなリセットロジックを削除した対策例を示す。ここで各レジスタは ff1[7:0]のように 8 ビット幅を持つ。

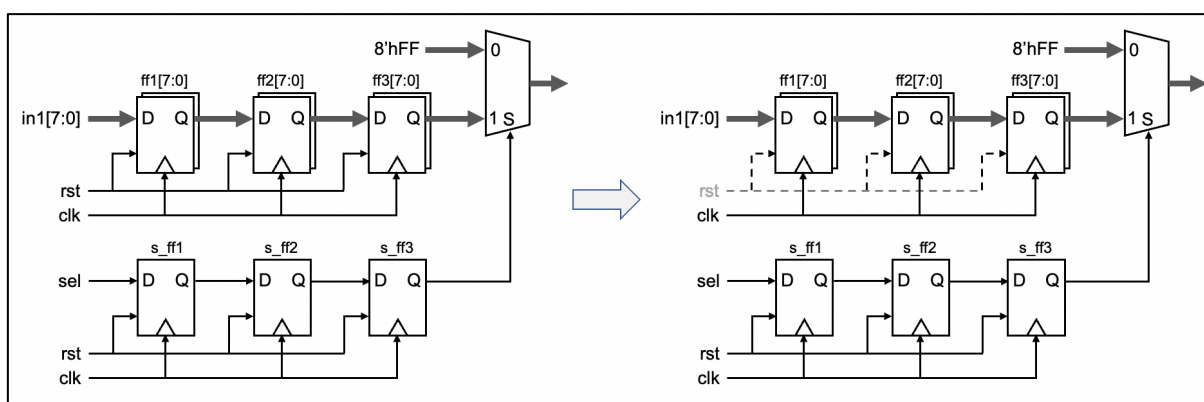


図 9. リダンダント・リセットの削除によるローパワー対策の例

過去に ASIC プロジェクトで設計したことがあれば、レジスタは必ずリセット付きを使用すること、といったようなリントチェック項目を常に有効にはしていないだろうか。あるいはレジスタに実装されているリセットロジックが本当に意味ある機能を果たしているかどうかを確認せずに使っていないだろうか。図 9.の左側では上段の in1[7:0]信号と下段の sel 信号を伝えるすべてのレジスタに同じリセットである rst 信号が使われている。リセット中はマルチプレクサの 0 側が選択されるが、リセットを解除して 3 サイクルすると初めてマルチプレクサに意味ある sel 信号が消費される。その間の同じ 3 サイクル分は、in1[7:0]信号の内容が何であろうと関係ない。これがリダンダント・リセットと呼ばれる所以であり、つまり不要なリセットロジックとなる。であれば図 9.の右側にあるようにリセットロジックを持たない面積の小さなレジスタに置き換えることで、ローパワー化が期待できる。

そして忘れてはいけないのが、クロックリージョンである。ローパワー対策によってレジスタの面積が小さくなり、そしてリセット信号の引き回しが無くなることで、同じクロックリージョンに配置される可能性が高くなる。それによってクロックラインの消費電力も下がることが期待される。

なお、このセクションで紹介したレジスタ出力やメモリ出力の後段に MUX がある場合には、レジスタ出力が使われていない場合にレジスタやメモリの動作をできるだけ長く抑える対策や、リダンダント・レジスタの削除などの手法は、ASIC においても効果が期待できる対策である。

VI. まとめ

ASIC をターゲットとした RTL 設計では、とにかくクロックゲーティングという手法が一般化し、さらに XOR ゲーティングやサーキュラーバッファと呼ばれる手法が編み出され、活用されてきた。ASIC 設計者であっても量産が見込めないプロジェクトでは FPGA を利用したり、あるいは初期ロットは FPGA で代替したりする場合もある。このような状況において ASIC と同じローパワー対策を適用していたのでは、むしろ消費電力が増加してしまうことにもなりかねない。異なるデバイスに対しては、その構造に見合ったローパワー対策が求められる。

そして忘れてならないことは、ローパワー対策は High Level 設計、RTL 設計、GL 設計のすべてのフェーズにおいて、それぞれの特徴に応じた対策を講じることが大切であり、これからの電子デバイスの開発においては欠かすことができない。

参考文献

- [1] Affaq Qamara, Fahad Bin Muslimb, Javed Iqbalb, Luciano Lavagnob, “LP-HLS: Automatic power-intent generation for high-level synthesis based hardware implementation flow”, MICROPROCESSORS and MICROSYSTEMS, Volume 50, pp 26-38, May 2017
- [2] AMD/Xilinx “Vivado Design Suite User Guide: Power Analysis and Optimization” UG907 V2020.1 June 3, 2020
<https://docs.xilinx.com/v/u/2020.1-English/ug907-vivado-power-analysis-optimization>
- [3] AMD/Xilinx Vivado Tool 2018.3 “Clock Regions” report