

CAD技術から見た非同期回路設計

松永, 裕介
九州大学大学院システム情報科学研究所 情報工学部門

<http://hdl.handle.net/2324/3438>

出版情報：情報処理学会システムLSI設計技術研究会30周年記念研究会. 102, pp.23-26, 2001-11. 情報処理学会SLDM研究会

バージョン：

権利関係：ここに掲載した著作物の利用に関する注意 本著作物の著作権は（社）情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。



CAD 技術から見た非同期回路設計

松永 裕介

九州大学大学院システム情報科学研究院
情報工学部門

〒 816-8580 福岡県春日市春日公園 6-1
092-583-7621

matsunaga@c.csce.kyushu-u.ac.jp

[概要] 本稿では低消費電力を目指した非同期回路設計手法について、それを支援する CAD 技術の立場から論じる。同期回路の設計手法はすでに確立しており、さまざまな CAD ツールが実用化されている。一方、非同期回路の設計に関しては、まだ十分に整理されているとは言えず、システムティックな方法論は確立していない。特に、性能を最適化するための手法や消費電力を最適化する手法などが今後の研究課題になるものと思われる。

キーワード 非同期回路, 低消費電力設計, 論理合成, タイミング解析

On Asynchronous Circuits Design from A View Point of CAD technologies

Yusuke Matsunaga

Department of Computer Science and Communication Engineering
Graduate School of Information Science and Electrical Engineering
Kyushu University

6-1 Kasuga Koen, Kasuga, Fukuoka, 816-8580, JAPAN Phone +81-92-583-7621
matsunaga@c.csce.kyushu-u.ac.jp

[abstract] This paper describes the design methodology for asynchronous circuits aiming low power from a view point of CAD technologies for that. The design methodology for synchronous circuits has been established well, so that various CAD tools are used in practice. On the other hand, in the case of asynchronous circuits, there are almost no systematic methodologies yet, especially, for optimizing the performance of a circuit or, for optimizing the power consumption of a circuit. These will be the most significant research theme from now on.

Keywords asynchronous circuit, low power design, logic synthesis, timing analysis

1 はじめに

非同期回路とはある意味、「同期回路」にあらざるものであり、さまざまな種類の回路を包含した概念である。そのため、非同期回路のなかには、同期回路に比べて遥かに高速に動作したり、遥かに低消費電力で動作する回路があったとしても不思議ではない。可能性の議論で言えば、クロックに同期して動作するという大きな制約を課せられた同期回路よりも、高いポテンシャルを秘めた回路方式とすることができるだろう。問題は、どのようにしたらそのような非同期回路を設計できるのか、という方法論を見いだすことにある。今までも、いくつかの非同期回路の設計事例は紹介されてきているが、それらのほとんどが、非同期回路設計に習熟した設計者の手作業に負うところが多く、非同期回路設計をシステマティックに行うための方法論やそのためのCADアルゴリズム、CADツールに関する議論はほとんどなされていないと思われる。本稿では、非同期回路設計を同期回路設計との対比において、設計手法やCADの観点から述べる。

2 同期回路設計における計算機支援

ここでは同期回路とは、すべてのフリップフロップが単一のクロックに同期して動作する回路のことを指すものとする。「同期する」とは厳密に同じタイミングで動作することを意味するのではなく、相互に一定の関係を持ったタイミングで動作することを意味するものとする。多相クロックやスキューのあるクロックを持つ回路ももちろん、同期回路である。

同期回路は、Moore型順序機械もしくはMealy型順序機械として実現できる。Moore型とMealy型の違いは外部出力が内部状態のみに依存するのか(Moore型)、内部状態と外部入力に依存するのか(Mealy型)、ということである。以下ではより一般的なMealy型順序機械を取り上げる。Mealy型順序機械では回路は次のような6つ組で表される。

$$M = (S, S_0, I, O, \delta, \lambda) \quad (1)$$

ここで、

S : 状態の集合

S_0 : 初期状態の集合

I : 入力の集合

O : 出力の集合

$\delta: S \times I \rightarrow S$, 状態遷移関数

$\lambda: S \times I \rightarrow O$, 出力関数

である。Mealy型機械を図1に示す。 $|S|$ 個の状態を保持するためには $\log|S|$ 個(もしくはそれ以上)のフリップフロップを用意すれば良いので、Mealy型順序機械の本質的な設計は組み合わせ回路である δ (状態遷移関数)と λ (出力関数)の設計と見なすことができる。また、エッジトリガタイプのフリップフロップを用いることで、状態遷移関数の設計時にハザードの発生の有無を考慮する必要はない。このように順序回路を同期回路として実現することで、設計時に考慮する問題を軽減化したり、解消したりすることができる。そのいくつかを以下に示す。

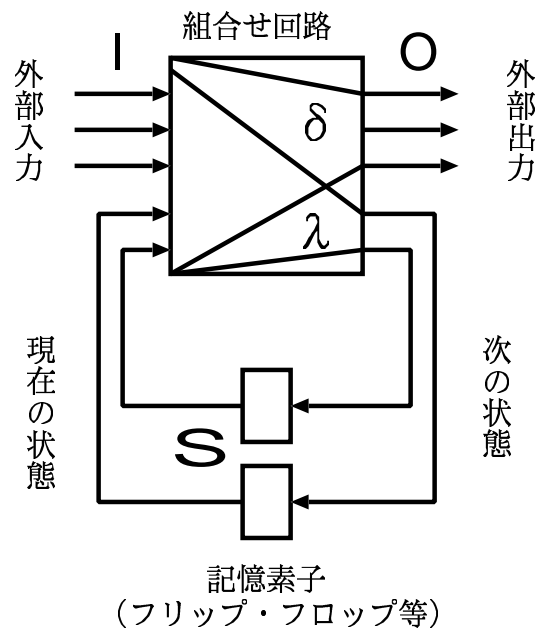


図1: Mealy型順序機械

- 順序回路として安全に動作する基準が明確になっている。
 - (前述のように) ハザードに対する考慮は要らない。
 - 組み合わせ回路部分の各バスの遅延時間がセットアップ条件および、ホールド条件を満たせばよい。
 - 詳細なタイミングを考慮しなくてもクロック単位で論理値の変化をトレースしておけば回路の動作シミュレーションが行える。
 - 回路の動作が決定的(deterministic)である。
- 動作を高速にするための基準、方法が明確になっている。

逆に、所望のスピードでは動作しない、とわかった時点でどのバスの遅延時間をどれだけ短くしなければならぬか、が明確になっている。

- 消費電力の見積りが比較的容易に行える。
各周期において組み合わせ回路の各部がどれくらいのスイッチング頻度でフリップするののかの見積りが行えれば、それと動作周波数から消費電力の見積りが行える。 *rightarrow* 遅延時間の場合と同様に、逆にどこをどうすれば消費電力が削減できるかの指標も明確になっている。

- 製造テストを容易にする仕組み (Design for Testability: DFT) を組み込むことが容易。
具体的には、フリップフロップをすべてスキャンタイプにすることで、順序回路に対するテスト問題を組み合わせ回路に対するテスト問題に置き換えることができる。

これらの利点を活かして同期回路設計をシステムティックに行う設計手法が確立しており、論理合成、スタティックタイミング解析、テスト生成、消費電力見積りなどの CAD ツールが実用化されている。確かにクロックの周波数を最悪の遅延時間で決めなければならないというデメリットはあるが、安定して動作するハードウェアを設計する、という観点からみれば同期回路のメリットは計り知れない。

3 非同期回路設計における計算機支援

一方、非同期回路に目を向けると、上記のような利点を挙げることは難しい。非同期回路の設計手法を考えるとときに障害となりうる点を以下に挙げる。

- 動作がイベントドリブンなので、非決定的な動作モデル (複数のイベントの生起順序が規定されていない動作モデル) を考える必要がある。 *rightarrow* 検証が難しい。フォーマルなメソッドとしては Petri net を用いたものが提案されているが、問題の複雑さは同期回路の状態探索問題よりも遥かに難しい。また、非決定的な動作モデルの場合、パタン (ステイミラス) を与えてシミュレーションしたところで、それは一つの動作にすぎず、すべての動作の検証をおこなうためには指数的なパタンを与えなければならない。

逆に、一般的な非同期回路ではなく、動作に非決定性を含まないモデルのみを扱うように限定する考え方もある。

- 同期クロックが存在しないので何らかの手段で動作の完了を表す必要がある。
 - － 状態遷移関数を単調変化回路として実現する。つまり、一回の遷移では必ず、 $0 \rightarrow 1$ もしくは

は $1 \rightarrow 0$ のどちらかの変化しか起こらないように回路を設計する。

この場合、ハザード (たとえば $0 \rightarrow 1 \rightarrow 0$ のような信号変化) が生じると誤動作するので、ハザードが生じないような回路を設計する必要がある。論理関数によってハザードが生じない回路では実現不可能な関数があるので、その場合には状態遷移を変更したり符号割り当てを変更する必要がある。

- － 2線式回路として論理関数を実現する。つまり、一つの論理値を2本の信号線を用いて表現する。たとえば、 $(0, 1)$ と $(1, 0)$ をそれぞれ 0 および 1 を表す定常状態として用いる。各々の論理関数は単調変化回路として実現する必要があるが、入力として肯定、否定両方の値が使用可能なので、任意の関数を単調変化回路として実現することは可能である。この手法の最大の欠点は通常の回路に比べて2倍の回路量を必要とすることである。
- － 最近の deep sub-micron 技術で問題になっているクロストークノイズなどに対する耐性が低い。

- 平均的な動作性能の見積りが難しい。同期式回路の場合、入力データによらずクロックで性能を見積もることができたが、非同期式回路の場合、入力データに依存して性能が変化するため、実際に動かしてみないと性能がわからない。

また、シミュレーションなどで性能見積りを行ったとしても、どの部分の遅延時間をどれだけ短くすると回路全体の性能がどれだけ向上するのか、の評価が難しい。これは同期回路のクリティカルパス解析と比べてもっとも異なる点である。

- 同様に消費電力の見積りが難しい。さらに消費電力を提言するために回路のどの部分をどのように修正すれば良いかがわからない。
- 製造テストの問題。テストをどのような仕組みで行うか、また、そのためのテストパタンをどのように作れば良いのか、について同期回路のテストのようなわかりやすいスキームは提案されていない。

歴史的にみて、非同期回路の設計手法に関しては、「どのようにしたら非同期回路が設計できるか」という点に関してはいくつかの提案がなされており、ハザードフリーな単調変化回路の合成手法や2線式回路の合成手法などの具体的な解決策が示されているものもあるが、「どのようにしたら高性能/低消費電力の非同期回路が設計できるか」という問いに答えるものは皆無であると思われる。

4 おわりに

現状では、誰もが簡単に高性能／低消費電力の非同期回路を設計できるわけではないが、これは必ずしも未来永劫この状況が続くと言っているわけでもない。前節で挙げた問題は今後の研究課題であり、これらをクリアしてゆくことで、非同期回路の設計環境を現在の同期回路の設計環境に近づけることは可能であると思われる。