

2 J-1

多相クロックを用いた高速デジタル回路の機能設計アルゴリズム

林 世紀

田中 譲

北海道大学 大型計算機センター

北海道大学 工学部

1. はじめに

実時間処理や大規模高速計算などを目的とする専用処理装置の研究開発が盛んである。高いスループットを実現するデジタル回路では、多相(N相)クロック方式がよく用いられる。この方式ではN個の相信号を使用して、構成素子の動作タイミングを制御する。単相クロック方式に比べ、多相クロック方式では精細なタイミング設計が可能である。そのため高い処理効率が期待できる。反面、設計面では最適化に多くの時間を要するという問題がある。

本稿では、多相クロックを用いたデジタル回路の機能レベルでの設計手法について述べる。各相信号は共通の周期Tで繰り返されるパルス列であり、位相はT/Nずつ異なる(図1参照)。高いスループットを得るには、ラッチ配置、および各ラッチの制御タイミングが問題となる<sup>(1)</sup>。これらを決定するアルゴリズムを示す。

ビルディングブロック方式によるレイアウト設計が普及している。本手法でもレイアウト設計にはこの方式を仮定した。したがって、機能ブロック内へのラッチ挿入は行わない。ただし、パイプライン方式の機能ブロックを使う場合はこの限りでない。

このアルゴリズムは機能素子間のラッチ挿入に基礎を置いている。そのためステージ分割法<sup>(2)</sup>では困難なループを含む回路の設計が可能である。

2. 設計手法

設計者はプログラムの形で動作仕様を記述する(図2参照)。これは一つの無限ループから構成される。ループの中は単一代入規則に従う代入文の列である。本手法では、与えられた動作仕様から機能レベルの回路(図3参照)を自動合成し、同時にスループットの向上を図る。物理的な制約から通常、クロックの相数Nは多くても10程度である。

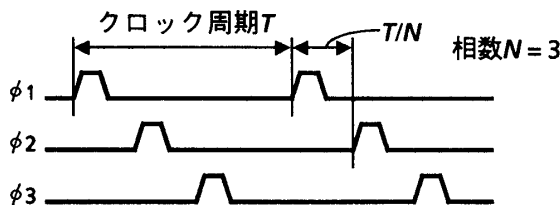


図1 多相(3相)クロック

```

input in;          (* 入出力変数の宣言 *)
output out;

do forever;
begin
  v3 ← f(v2);
  v2 ← g(v1);
  v1 ← in - c · u3;
  u3 ← u2;
  u2 ← u1;
  u1 ← out;
  out ← a1 · v1 + a2 · v2 + a3 · v3 + b
end;
    
```

図2 動作仕様

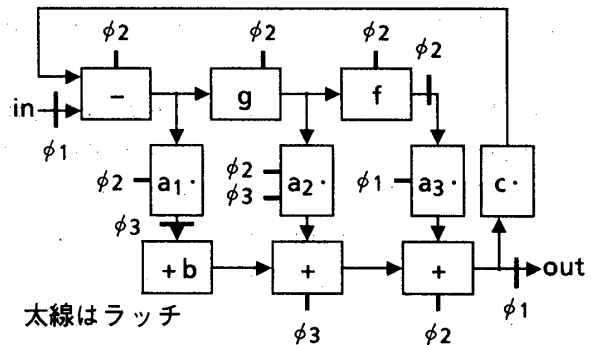


図3 多相クロックを用いたデジタル回路

相数は設計者が適当に定めるか、さもなくば、1から順に最適な相数を捜せばよい。

以下、次のような回路設計を考える。

- (1) 回路の動作仕様は与えられる。
- (2) クロック信号の相数Nも与えられる。
- (3) クロック周期Tをできる限り短くする。

3. 設計アルゴリズム

設計アルゴリズムは2つのステップから成る。(ステップ1) 動作仕様中に関数が現れるたびに、その関数を実現する機能素子を用意する。関数の依存関係に基づき機能素子間の接続を決定する。(ステップ2) 機能素子間に必要に応じてラッチを挿入する。最短のクロック周期と各ラッチの制御タイミングを決める。

ステップ1では、機能素子とそれらの接続関係が決定される。これをグラフ表示したものを機能グラフと呼ぶ(図4参照)。ただし、パイプライン方式の機能素子を使う場合は、その段数分の頂点を用意する。例えば、引き算-は2段のパイプラインを用いて実現する。

A Functional Design Algorithm for a High-Speed Digital Circuit using Multiphase Clocking

Toshinori HAYASHI

Computing Center, Hokkaido Univ.

Yuzuru TANAKA

Faculty of Engineering, Hokkaido Univ.

次に、ステップ2で用いるアルゴリズムについて説明する。動作仕様中、 $n$ 回前の繰り返しにおける関数 $f$ の値が関数 $g$ の引数になっているとしよう。このとき、機能グラフの $f$ から $g$ への辺上に重み $n$ を定義する(図4参照)。経路 $p$ 上の重みの総和を $w(p)$ により表す。各頂点には、対応する素子の遅れ時間が定まる(図4中に斜体数字で示した)。経路 $p$ 上の両端を除く頂点について、それらの遅れ時間の総和を取る。これを $d(p)$ により表す。2つの辺 $e_1, e_2$ に対し、 $W(e_1, e_2), D(e_1, e_2)$ を次のように定義する。

$$W(e_1, e_2) = \min\{w(p) \mid p \text{ は } e_1 \text{ から } e_2 \text{ への経路}\}$$

$$D(e_1, e_2) = \max\{d(p) \mid p \text{ は } w(p) = W(e_1, e_2) \text{ を満たす } e_1 \text{ から } e_2 \text{ への経路}\}$$

頂点 $f$ から頂点 $g$ への辺を $[f, g]$ と表すと、例えば、 $W([1, 2], [23, 24]) = 0, D([1, 2], [23, 24]) = 58$ となる。 $W(e_1, e_2), D(e_1, e_2)$ の値を求めるアルゴリズムが知られている<sup>(3)</sup>。

時間の集合 $Q$ を次のように定義する。

$$Q = \{N \cdot D(e_1, e_2) / i \mid e_1, e_2 \text{ は辺で、} i \text{ は相数 } N \text{ 以下の自然数}\}$$

例えば、 $N=3$ とすると、 $3 \times 58, 3 \times 58/2, 58$ は $Q$ の要素である。最短のクロック周期は $Q$ の要素中に必ず存在する。 $Q$ の中から二分探索により、次の条件を満たす最小の要素 $T$ を見つける。それが最短のクロック周期である。

(実現条件) 適当なラッチ配置と、各ラッチを制御する相信号が存在し、クロック周期 $T$ で動作可能である。

実現条件を満たす回路が存在するかどうかを求める問題を実現問題と呼ぼう。次にこの問題を解くアルゴリズムについて述べる。機能グラフの各辺を次のように色付けする。ラッチを挿入することに決めた辺を青、しないことに決めた辺を赤、どちらにも決めていない辺を無色とする。初期状態では全て無色である。辺 $e$ と非負整数 $i$ に対し、ラベル値 $L_i(e)$ を次のように定義する。

$$L_i(e) = \begin{cases} 0 & e \text{ の始点が入力} \\ \perp & i = 0 \\ \max\{L_{i-1}(e), m_i(e, e)\} & i > 0 \end{cases}$$

ここで、辺 $e, e_0$ に対し $m_i(e, e_0)$ を次のように定義する。

$$m_i(e, e_0) = \max\{\ell_i(e', e_0) \mid e' \text{ の終点は } e \text{ の始点と一致し、} D(e', e_0) \leq T \text{ を満たす}\}$$

さらに、 $\ell_i(e, e_0)$ を次のように定義する。

$$\ell_i(e, e_0) = \begin{cases} \lceil L_i(e) \rceil + N \cdot D(e, e_0) / T - N \cdot W(e, e_0) & e \text{ は青色} \\ m_i(e, e_0) & e \text{ は赤色} \\ \min\{m_i(e, e_0), \lceil L_i(e) \rceil + N \cdot D(e, e_0) / T - N \cdot W(e, e_0)\} & e \text{ は無色} \end{cases}$$

$\min\{\perp, a\} = \perp, \max\{\perp, a\} = a$ とする。 $N$ は相数を表す。例えば、 $N=3, T=30$ 、全ての辺を無色とすると、 $L_4([4, 5]) = 0.5, L_4([11, 12]) = 4$ である。辺 $[1, 2]$ が赤なら、 $L_4([11, 12]) = 4.3$ である。

グラフ中の全辺の数を $n$ とする。ある辺 $e$ が存在し、 $L_n(e) \neq L_{n+1}(e)$ を満たすとしよう。この場合、無色の辺をどのように色付けしてもクロック周期 $T$ では実現できない。このような色付きグラフはラベル付けに失敗すると言う。そうでないときはラベル付けに成功すると言う。

グラフ中の各辺の向きを逆にすることにより新たなグラフが得られる。これに対し、同様にラベル値 $M_i(e)$ を定義する。

発見的手法を用い次のように実現問題を解く。

- (1)  $\min\{M_n(e) - \lceil L_n(e) \rceil, L M_n(e) - L_n(e)\}$ が最小になるような辺 $e$ を選ぶ。
- (2) 辺 $e$ を赤に色付けしたグラフがラベル付けに成功するならば、それを新たなグラフとする。
- (3) (2)のラベル付けに失敗し、さらに $e$ を青に色付けしたグラフもラベル付けに失敗するならば、実現不可能と判断し、終了する。
- (4) さもなくば、青に色付けしたグラフを新たなグラフとする。
- (5) 得られたグラフの全ての辺が青か赤に色付けされたならば、実現可能と判断し、終了する。
- (6) さもなくば、得られた色付きグラフに対し、(1)から(6)を適用する。

このアルゴリズムを図4の機能グラフに適用すると、最短のクロック周期は30となり、太線で示した辺が青色となる。この結果から図3の回路が得られる。

#### 4. おわりに

多相クロックを用いたデジタル回路の機能レベルでの設計手法について述べた。さらに、高いスループットを実現するための設計アルゴリズムを示した。

#### 参考文献

- (1) 林、田中:多相クロックを用いたデジタル回路の設計、情処第38回全大4S-3
- (2) N. Park, A. C. Parker, "Theory of Clocking for Maximum Execution Overlap of High-Speed Digital Systems", IEEE Transaction on Computers, Vol. 37, No. 6, 1988, pp. 678-690.
- (3) C. E. Leiserson, et al., "Optimizing Synchronous Circuitry by Retiming", Proc. Third Caltech Conf. VLSI, Computer Science Press, 1983, pp. 87-116.

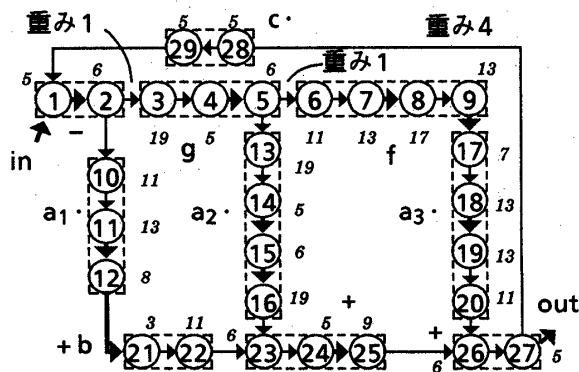


図4 機能グラフ