

非同期式プロセッサ TITAC-2 の同期インタフェース

1G-3

桑子雅史

高村明裕

上野洋一郎

南谷 崇†

東京工業大学 情報理工学研究所

†東京大学 先端科学技術研究センター

1 はじめに

我々は 32 ビット非同期式プロセッサ TITAC-2 (Tokyo Institute of Technology Asynchronous Computer 2) の論理設計・レイアウトを行なった。TITAC-2 のメモリには、従来の同期式プロセッサで用いられているのと同じものを使用している。

クロックを用いない非同期式プロセッサにおいて同期式のデバイスを使用する場合、それらが要求する動作タイミングを保証し、非同期回路部分とのインタフェースを行なう方法が問題となる。

本稿では、TITAC-2 における同期/非同期インタフェースの実現方法を説明する。

2 TITAC-2 の遅延仮定

非同期回路の設計を考える上では遅延仮定が重要である。遅延仮定とは、素子や配線の遅延に関して設ける仮定のことである。遅延仮定にはいろいろなものが提案されている。設計された回路のタイミング・フォールト(素子・配線の遅延時間が回路の設計段階や実際の動作時において設計者の設けた仮定から外れること)に対する耐性、即ち、タイミング信頼性は、どのような遅延仮定に基づいて論理設計を行なうかに影響される。

一方、遅延仮定は速度性能にも影響を与える。一般に、タイミング信頼性の高くなるような厳しい遅延仮定に基づいた非同期回路の速度性能は低くなり、回路量は多くなる。

従って、非同期回路の設計においては速度性能とタイミング信頼性とのトレードオフによって遅延仮定を決定する必要がある [1]。

最も厳しい遅延仮定として知られているのは、DI(Delay Insensitive) モデル [2] である。これは「配線遅延及び素子遅延は有限であるが、その上限値は未知である」とするものである。この遅延仮定に基づいて設計した回路は、タイミング・フォールトを起こさない。

大きなファンアウトを持つ素子や、チップの端から端への配線は、テクノロジーマッピングやレイアウトによって遅延時間が大きく変動する可能性があるため、DI モデルに基づく設計を行なうのが適切である。しかし実際の論理設計において、このような素子や配線はシステム全体の中では非常に少ない。

多くの素子や配線は、テクノロジーマッピングやレイアウトの際に適切な制約を課すことで、遅延変動の範囲を予測することができると思われる。このような部分には、より緩い遅延仮定を用いることができ、それによって性能を向上させることができる。

以上の考察に基づき、TITAC-2 の設計では、回路全体を一定の面積を越えない複数のブロックに分割した後、ブロック間の配線には DI モデルを仮定し、ブロック内は以下のような比例遅延変動モデル [3] を仮定して論理設計を行なった。

比例遅延変動モデル ある配線遅延、または素子遅延の推定値が d_e であり、実際の遅延が d_a であるとき、 $v = d_a/d_e$ を遅延変動率という。このとき、「任意の二つの遅延の変動率 $v_1, v_2 (v_2 > v_1)$ の比 v_2/v_1 の上限値 k は既知である。」とする遅延仮定を比例遅延変動モデルと呼ぶ。

TITAC-2 では、 $k = 2$ として設計を行なっている。

3 2線2相式データ転送

クロックのない非同期回路でデータ転送を実現するためには、データの到着を知るためにデータに時間情報を付

Synchronous Interface of Asynchronous Processor TITAC-2
Masashi Kuwako, Akihiro Takamura, Youichiro Ueno, Takashi Nanya
Tokyo Institute of Technology, Graduate School of Information Science and Engineering

加する必要がある。TITAC-2 内の多くのデータ転送は 2 線 2 相式 [2] で行なわれている。

この方式では、1 ビットのデータ X は 2 本の信号線対 (x, \bar{x}) を用いて伝送される。この信号線対は有効符号語、無効符号語と呼ぶ 2 種の状態をとる。有効符号語はデータを 2 線符号化して表わしたものであり、 $(x, \bar{x}) = (0, 1)$ は $X = 0$ を、 $(x, \bar{x}) = (1, 0)$ は $X = 1$ を表わす。無効符号語は $(x, \bar{x}) = (0, 0)$ の状態である。

回路はその環境から有効符号語を入力されると、演算などの処理を行なって、結果の有効符号語を出力する。これが稼働相である。そして次に無効符号語が入力されると無効符号語を出力する。これが休止相である。この稼働相と休止相を交互に実行して演算を行なうことから、2 線 2 相式と呼ばれる。2 線 2 相式は DI モデルに基づくデータ転送を実現することができる。

4 TITAC-2 のメモリと束データ方式

TITAC-2 のチップ内部のキャッシュメモリと、チップ外部のメインメモリには、従来の同期式プロセッサで用いられているメモリデバイスを使用している。内部メモリはスタンダードセルの RAM マクロ、外部メモリはスタティック RAM 及び EPROM である。

メモリはその動作タイミングが規定されており遅延時間が予測可能なため、その読み出し/書き込み動作には、 n ビットの信号線に対して 1 本のストロブ信号線を用いる束データ方式 [2] を用いている。

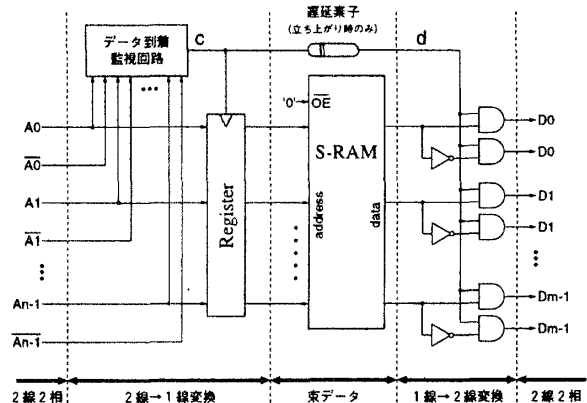


図 1: 外部メモリの読み出し動作の概念図

束データ方式の回路と 2 線 2 相式の回路のインタフェースは図 1 のように実現している。これはメインメモリの読み出し動作を例にとった概念図である。

この動作は以下ようになる。入力にアドレスの有効符号語が到着すると、データの到着を監視する回路の出力 c が 1 となってアドレスがレジスタに読み込まれる。レジスタの読み込み遅延時間+メモリの読み出し遅延時間+インバータの遅延時間が経過した後、遅延素子の出力 d が 1 となり 2 線符号化されたデータが出力される。入力に無効符号語が到着すると、 $c = 0, d = 0$ となってデータ出力は無効符号語となる。

TITAC-2 では、束データ方式で動作させる必要がある回路間のデータ転送は、そのデータパスの途中にある回路も束データ方式の回路とする設計を行なっている。

例えば、命令キャッシュにおけるラインフェッチでは、メインメモリ→ラッチ→キャッシュメモリのデータ転送が行なわれる [4]。これを行なう回路においては、ラッチも同期式回路で用いられているものと同じものを使用している。メインメモリ→ラッチ→キャッシュメモリのデータパス全体を束データ方式とすることで、2 線 2 相式⇔束データ方式の変換を減らして速度性能の向上と回路量の削減をしている。

5 可変遅延回路の使用

束データ方式によるデータ転送を行なう箇所での遅延素子の遅延時間をいくらにするかは問題である。 $k=2$ の比例遅延変動モデルに基づくとする、メモリの最大遅延時間の2倍の大きさの遅延素子を用いることになり、システムの動作速度を大きく低下させる。

TITAC-2では、束データ方式で用いる遅延素子にプロセッサ動作時にチップ外部から遅延時間を可変できる回路を使用することでこの問題を解決した。

キャッシュメモリのようなチップ内部にある回路の動作時間を決定するための遅延素子は、図2のように実現している。適当な遅延時間を持つゲートを継続接続して、各ゲートの出力をマルチプレクサに入力する。チップ外部から直接書き込みを行なえるラッチの出力によってマルチプレクサの選択が行なわれる。

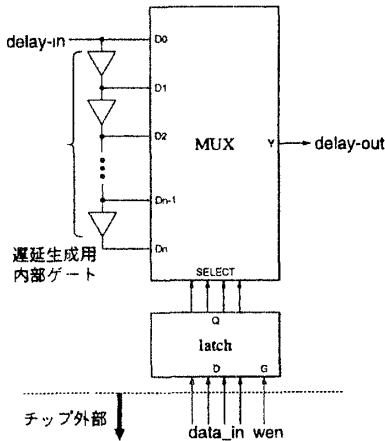


図2: チップ内部デバイス用の可変遅延回路

メインメモリのようなチップ外部にある回路の動作時間を決定するための遅延素子は、図3のように実現している。図2との違いは、チップ内部の遅延設定用ゲート群によって生成した遅延に、チップ外部に接続する任意の遅延素子による遅延を加算することも可能としてあることである。

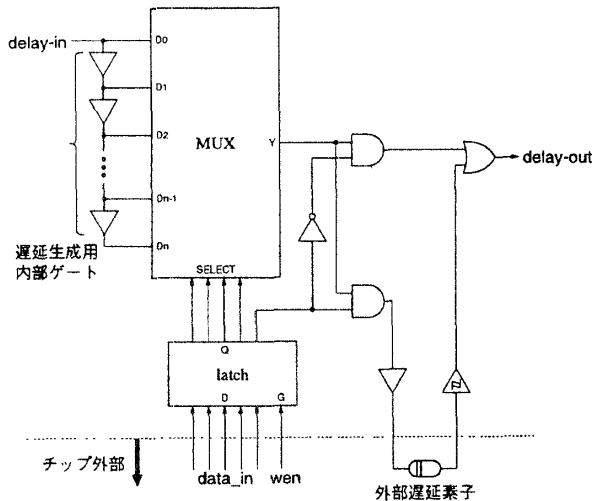


図3: チップ外部デバイス用の可変遅延回路

TITAC-2で使用したスタンダードセルにおいて、チップ内部に遅延素子を実現しようとする、その遅延時間の絶対誤差は-30%~+50%程度とかなり大きなものとなる。

TITAC-2の設計において必要となった遅延素子は、使用する箇所によって異なるが、およそ2ns~10nsの遅延時間を持つものであった。チップ外部に遅延素子を接続すると、信号線をチップ内部と外部との間を往復させるだけで約2nsを要するため、あまり小さな遅延時間を設定することがで

きない。また、一般に外部に接続する遅延素子は遅延の可変ステップも数ns単位と大きくなってしまい、遅延時間を細かく調整できない。

従って、図3及び図2のような回路を使用した。

また、図1の例のように、立ち上がり時のみに遅延があり、立ち下がり時には遅延がゼロであって欲しい遅延素子がTITAC-2の設計においていくつか必要であった。この実現には、Qモジュール [2] と呼ばれる回路を可変遅延回路に付加した図4を用いている。

この回路では delay-in が 0→1 となると、 x が 0→1、 y が 0→1、 x が 1→0、 y が 1→0 と遷移してから delay-out が 0→1 となる。delay-in が 1→0 となると、 x, y は遷移せずに delay-out が 1→0 となる。従って、図4によって、可変遅延回路の遅延時間の約2倍の大きさを持つ立ち上がり方向のみの遅延を実現できる。

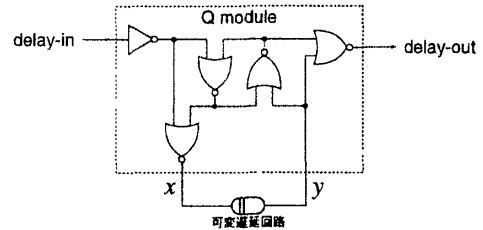


図4: チップ外部デバイス用の可変遅延回路

6 まとめ

本稿では、非同期式プロセッサ TITAC-2 において従来の同期式回路に用いられているメモリデバイスと非同期式回路とのインターフェースを実現する方法を示した。

これは、プロセッサの動作時にチップ外部から調整することが可能な可変遅延回路を使用するものである。TITAC-2 全体では、このような可変遅延回路を11箇所で使用しており、従来の同期式デバイスを用いている部分においても十分なタイミング信頼性と高い速度性能を両立させている。

TITAC-2ではプロセッサの動作を観測するために、各可変遅延回路の入出力も含む要所の信号線がマルチプレクサを通してチップ外部に引き出されている。実際にTITAC-2を動作させるときには、これらの信号遷移を観測することによって可変遅延回路の遅延時間を決定することになる。この方法で遅延設定をどの程度正確に行なえるかは本稿執筆時点では不明である。また、チップの温度変動に対してどの程度の耐性を持つのかも不明である。以上の課題は、チップが完成次第、実際のチップを用いて検証する予定である。

また、可変遅延による同期インターフェースは、比較的大きなチップ面積を占有する。可変遅延を用いずにタイミング信頼性と速度性能を両立できるような同期インターフェースの研究も今後の課題である。

なお、本研究の一部は新エネルギー・産業技術総合開発機構 (NEDO) 提案公募型・最先端分野研究開発事業受託研究 C-026、並びに科学研究費補助金 (試験研究 B) 07558036 によって行われたものである。

参考文献

- [1] Masashi Kuwako and Takashi Nanya. Timing-reliability evaluation of asynchronous circuits based on different delay models. In *International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pp. 22-31, Salt Lake City, November 1994. IEEE Computer Society.
- [2] 南谷崇. 非同期式プロセッサ — 超高速 VLSI システムを目指して —. 情報処理, Vol. 34, No. 1, pp. 72-80, January 1993.
- [3] 高村明裕, 桑子雅史, 南谷崇. 非同期式プロセッサ TITAC-2 の論理設計における高速化手法. 信学論 (D-I), Vol. J80-D-I, No. 3, March 1997. (掲載予定).
- [4] 石川誠, 桑子雅史, 山崎淳, 上野洋一郎, 南谷崇. 非同期式プロセッサ TITAC-2 のキャッシュ構成. 情報処理学会第 54 回全国大会, March 1997.