

二相式非同期プロセッサの構成に関する一考察

4D-1

上野洋一郎

本間一樹

南谷 崇

東京工業大学 工学部

1 はじめに

近年のデバイス技術の進歩により次々と高速な半導体が開発され、実験室レベルではスイッチング遅延が数ピコ秒の論理素子が実現されたという報告もある。しかし、半導体の高速化に伴って、遅延全体の中で配線遅延の占める割合が増加してきた。この結果、回路全体にクロックを分配する必要のある同期式回路では素子の高速性を十分に生かすことが困難になりつつある [1]。

非常に高速な素子を生かす方法の一つとしてクロック信号の無い非同期式回路を用いる方法があり、非同期式回路を用いてプロセッサを構成することで高速なプロセッサが得られると考えられる。

本稿では、Muller モデル [2] と要求応答モデルに基づく 2 相式非同期プロセッサの一構成法を示す。

2 Muller モデル

Muller モデルの仮定を以下に示す。

1. 入力変化は回路自身の特別な出力 (完了信号) が出された時のみ許される。
2. 配線遅延は存在せず、素子遅延の大きさは有限であるが上限値は未知である。

Muller モデルにおいて「配線遅延は存在しない」としているが、配線遅延を素子の出力の遅延に含めたり、疑似素子を挿入することで配線遅延に対応することが可能である。

3 2 相式

本稿で示す 2 相式非同期プロセッサでは要求応答モデルで配線と回路の「稼働相」と「休止相」の 2 相を制御し、プロセッサとして統一された動作を実現する [3][4]。配線は「1」又は 2 線 2 相式の「有効符号語」で要求や応答を出している稼働相と、「0」又は 2 線 2 相式の「無効符号語」で要求や応答を取り下げている休止相を繰り返し、回路は要求が来ると稼働相に移行して処理後応答を出し、要求が取り下げられると休止相に移行して応答を取り下げる。

4 プロセッサの演算部の構成

4.1 レジスタ間転送モデルと構成要素

プロセッサを構成する上で基本となるレジスタ間転送モデルを 2 相式非同期プロセッサに当てはめたモデルを図 1 に示し、図中の構成要素の動作を以下で説明する。

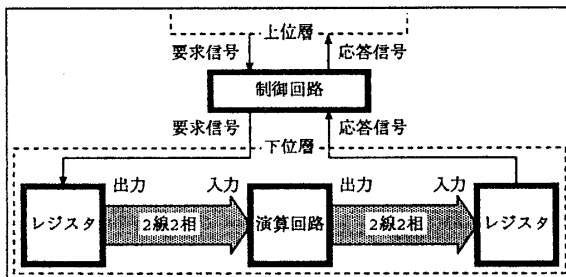


図 1: 2 相式非同期レジスタ間転送モデル

制御回路

上位層からの要求信号に応じて下位層の要求・応答動作や上位層への応答信号を制御する。

レジスタ

要求信号を受け取ると記憶している情報を出力に有効符号語で出し、要求信号が取り下げられると出力に無効符号語を出す。また、入力に有効符号語が来るとその情報を記憶し応答信号を出し、入力に無効符号語が来ると応答信号を取り下げる。

Note on 2-Phase Asynchronous Processor Organization
Yoichiro Ueno, Ichiki Honma, Takashi Nanya
Faculty of Engineering, Tokyo Institute of Technology

演算回路

入力に有効符号語が来ると演算を行なって出力に有効符号語を出し、入力に無効符号語が来ると演算を終了し出力に無効符号語を出す。

4.2 機能モジュール間の接続方法

プロセッサの内部には「Register File」や「ALU」といった幾つかの機能モジュールが存在する。これらの機能モジュール相互の接続方法として最も一般的な方法がバスを用いた接続である。

バスは複数の機能モジュールに同時に少ない配線量でデータを送信できるが、動作する必要のない機能モジュールにまでデータを送ってしまうため、データの到着でモジュールが動作を開始する図 1 のレジスタ間転送モデルには不適当である。又、2 相式動作を確実にするためにバスの状態を完全に監視するように構成すると、実際にデータを転送したい配線の長さよりも長い配線を監視することになるので高速性が失われる。

本稿の構成法では、「データ転送のある機能モジュール間を 1:1 でつなぐ」方法を用いた。この方法は図 1 のモデルに適するが、配線の共有が無い配線量が増加する欠点もある。

4.3 マスタ・スレーブレジスタ

図 1 のレジスタ間転送モデルでは、データの転送元レジスタと転送先レジスタとは分離しているが、多くの場合この二つのレジスタは図 2 に示すように同一のレジスタである。もしレジスタが記憶部を一つしか持たないとレジスタに入力されたデータを記憶した時点でレジスタの出力が変化してしまい、情報の循環が発生して因果関係が保証できない。

因果関係を保証するためには、レジスタに記憶部を二つ設けて情報を出力する動作 (slave) と新しい情報を受け取る動作 (master) を二つの記憶部に振り分ける、「マスタ・スレーブ化」が必要である (図 3)。また、マスタ・スレーブ化によって生じる演算結果の master から slave への転送は、演算の休止相で行なうことで、次のレジスタ間転送による演算で即座に使用可能とする。

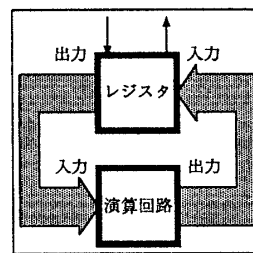


図 2: 一致する場合

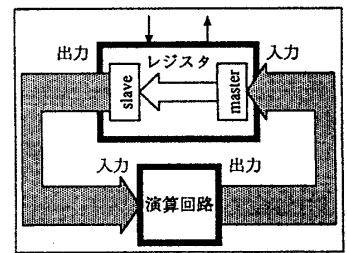


図 3: マスタスレーブによる実現

4.4 演算部の構成

以上のレジスタ間転送モデル、機能モジュール間の接続方法、さらにマスタ・スレーブレジスタを用いて構成したプロセッサの演算部の一例を図 4 に示す。

本構成の各機能モジュールの機能と動作を次に示す。

- プログラムの解析結果を記憶するデコードレジスタ (DR)
 - DR から受け取った演算種別と出力先に従って演算を行ない演算結果をレジスタファイルか主記憶に出力する ALU
 - DR から受け取った出力レジスタ番号と出力先、入力レジスタ番号に従って被演算情報を出力、演算結果を記憶するレジスタファイル
 - DR から受け取った書き込み・読み出し指示と、ALU から受け取ったアドレスに従って書き込み・読み出しを行なう主記憶
- また本構成で可能な処理は次の通り。
- レジスタ～レジスタ間での演算
 - レジスタ～即値間での演算

- ・一つのレジスタでの演算
- ・レジスタから主記憶への書き込み
- ・主記憶からレジスタへの読み出し

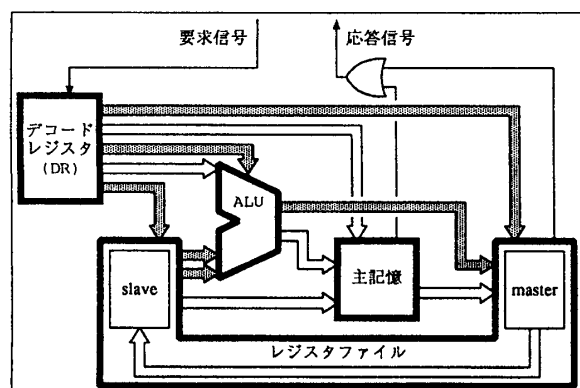


図4: 非同期プロセッサの演算部の構成

図4の色の付いた矢印はデコードレジスタに記憶されている情報を解析した結果、レジスタ～レジスタ間の演算が実行される際に、情報がどのような経路を流れるかを示したものである。

5 プロセッサの制御部の構成

5.1 休止相の制御

要求応答モデルでは複数の処理を順次に行なう逐次制御を、ある処理が終了して応答信号を返すとその信号を次に開始する処理への要求信号とすることで実現する。繰り返し制御は、その処理の応答信号を反転して要求信号に戻す(応答が来ると要求を取り下げ、応答が取り下げられれば要求を出す)ことで実現する。フェッチ、デコード、実行を繰り返す様なプロセッサの制御回路は、逐次制御と繰り返し制御を組み合わせると図5のようになる。

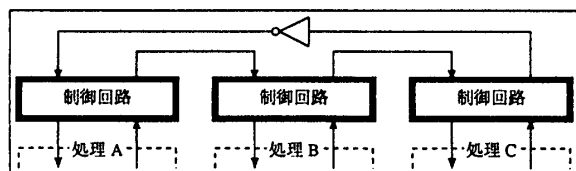


図5: 逐次制御と繰り返し制御を組み合わせた制御回路

制御回路に「上位からの要求が来ると下位の稼働相と休止相を終了させてから上位に応答を返す」という動作の「正縁起動回路[3]」を用いると、処理の流れが図6のようになり、全体の処理時間の約半分を休止相が消費して処理効率は半分程度に低下する。

これに対して本構成法では、「上位からの要求が来ると下位の稼働相を実行後、上位に応答を返すと共に休止相を実行する」という動作の「自掃モジュール[5]」を用いることで、処理の流れが図7のようになり、休止相による処理効率の低下を無くしている。



図6: 効率の悪い2相処理

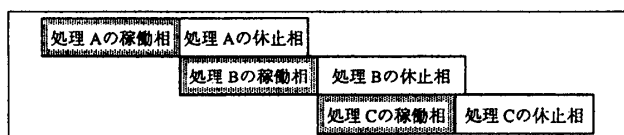


図7: 効率の改善された2相処理

5.2 割り込みの実現

割り込みはプロセッサにおける重要な機能の一つであるが、割り込みと通常の処理の間には因果関係が存在しない。しかも、割り込みによって発生するプロセッサの状態退避の処理と通常の処理は、互いにプロセッサ自身を共有しているので排他的に実行する必要がある。二つの動作を調停するために「アービタ(arbiter)」を導入する必要がある。

通常割り込みによる状態退避の処理は命令実行の終了後、次の命令のフェッチの前に行なわれる。従って、命令実行からの応答信号とフェッチへの要求信号の間にアービタを挿入してプロセッサの通常処理と割り込みとの間に因果関係をつける。

5.3 制御部の構成

以上の休止相制御と割り込みの実現方法を用いて構成したプロセッサの制御部の一例を図8に示す。この図8ではアービタで割り込みの要求があったことを記憶する「割り込みフラグ」へのアクセスに順序関係を与えている。

本構成の制御回路は以下の処理を繰り返す。

退避処理 割り込みフラグをチェックしてセットされていると状態退避の処理を行ない、セットされていないと何もせず即座に応答を返す

フェッチ 命令のフェッチ

デコード 命令のデコード(図では省略)

実行 命令の実行

割り込み要求があると「割り込み受け付け処理」で割り込みフラグをセットし、通常の処理は「退避処理」で割り込みフラグのセットを確認してプロセッサの状態退避を行なった後、割り込み処理ルーチンの実行に移行する。

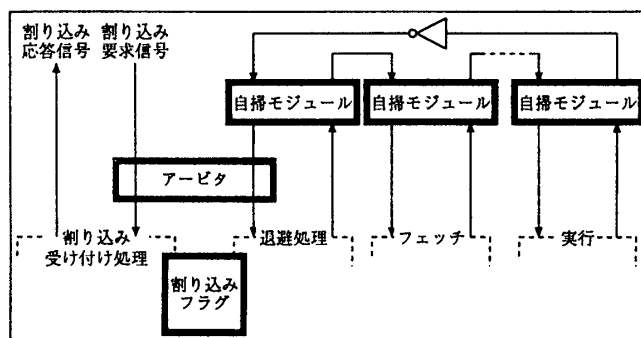


図8: 非同期プロセッサの制御部の構成

5.4 まとめ

本稿では2相式非同期プロセッサの基本構成要素と一構成法を示した。プロセッサ全体の構成は、図8の「実行」の部分に図4が、「割り込み受け付け処理」、「退避処理」、「フェッチ」、「デコード」の部分には本稿では割愛した各々のデータ転送路が繋がる。

また、2相式の本構成法での処理能力低下を無くする方法として、自掃モジュールとマスター・スレーブ間の転送を休止相に行なう方法を示した。又、アービタを用いた非同期プロセッサにおける割り込みの一構成法を示した。

今後は、同期式プロセッサで用いられている高速化手法(パイプライン等)の非同期化や、非同期プロセッサ独自の高速な構成法の研究を行なう必要がある。

本研究の一部は(財)大川情報通信基金1991年度研究助成によるものである。

参考文献

- [1] 南谷崇. 同期式プロセッサの限界と非同期式プロセッサの課題. 信学技報, December 1990. FTS90-45.
- [2] D.E.Muller and W.S.Bartky. "A theory of asynchronous circuits". In *Proc.Int.Symp.Theory of Switching*, pp. 204-243, 1959.
- [3] 山村良恵, 南谷崇. 非同期式制御モジュールと組合せ回路の一構成法. 信学技報, May 1991. FTS91-17.
- [4] 上野洋一郎, 南谷崇. 2線2相2系方式による非同期式レジスタ間転送. 信学技報, July 1991. FTS91-23.
- [5] 籠谷裕人, 南谷崇. プロセス記述による非同期式制御回路合成の一手法. 情処研報, December 1991. DA60-10 (信学技報, VLD91-98 と同じ).