計算機アーキテクチャ 126－19
（1997．10．29）

# 非同期式プロセッサを用いたコンピュータシステムの実現

上野洋一郎† 高村明裕† 小沢基一†
桑子雅史†† 南谷崇††

本稿では非同期式プロセッサを用いてコンピュータシステムを構成した場合と同期式プロセッサを用いた場合との違いを明らかにするとともに，相違点の大きいハードウェアに於いて既存のメモリやI/Oデバイスをどのように非同期式プロセッサに接続すれば良いかについて考察した．また実際に設計・製作した32-bit非同期式プロセッサの評価用システムを通して，これらの考察に基づいた既存のデバイスへのインタフェース方法の実証も行なった．

# Implementation of A Computer System with An Asynchronous Processor

YOICHIRO UENO,† AKIHIRO TAKAMURA,† MOTOKAZU OZAWA,†
MASASHI KUWAKO†† and TAKASHI NANYA ††

In this paper, we describe differences between computer systems with an asynchronous processor and computer systems with a synchronous processor. Since asynchronous hardware differs from synchronous one in many points, we present interfacing methods between asynchronous processors and synchronous peripherals, such as memory and I/O devices. These methods successfully works for the 32-bit asynchronous processor evaluation systems.

## 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論理素子が開発されている．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，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，非同期式回路と非同期式プロセッサの研究が盛んに行なわれている．

非同期式プロセッサの利点としては，

- 近傍とは高速に，大域ではそれに見合って低速にと，処理速度が柔軟に変化するので，殆どの処理を近傍で行なう様に設計すれば高速化が可能．
- より厳しい遅延に関する仮定を用いることで，非同期式回路は様々な原因による遅延の変動に対してより強くなり，遅延非依存性(delay insensitivity)の高いプロセッサが構成可能．
- ハンドシェイクによる制御を用いて演算に必要な回路のみ動作させることで，低消費電力化が可能．

等があげられる．そして非同期式プロセッサを実際に製作した報告もなされている[2)~5)]．南谷研究室でも実験的な8-bit非同期式プロセッサTITAC[6)]に続き，実用性を備えた32-bit非同期式プロセッサTITAC-2[7),8)]の設計・製作を行なった．

本稿では非同期式プロセッサを用いてコンピュータシステムを構成した場合と同期式プロセッサを用いた場合とで異なる点を明らかにした．そして特に相違点の大きいハードウェアについては，非同期式プロセッサと既存のメモリやI/Oデバイスとのインタフェースについて考察した．また実際に32-bit非同期式プロセッサTITAC-2の評価用システムで用いたインタフェースについても述べる．

## 2. 非同期式回路

### 2.1 制御方式

同期式回路の制御の基本は単純で，制御回路が演算回路に演算の種類を指示する信号を送るだけでよい．演算回路は次のクロックエッジでその指示に従った演算を開始するので，制御回路は何クロック後に演算結果が利用可能になるかもあらかじめ知ることが出来る．それに対し非同期式回路はクロック信号が無いため演

† 東京工業大学情報理工学研究科
Graduate School of Information Science and Engineering, Tokyo Institute of Technology
†† 東京大学先端科学技術研究センター
Research Center for Advanced Science and Technology, University of Tokyo

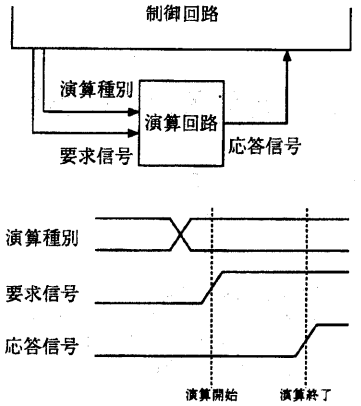


図1　要求応答方式
Fig. 1　request-acknowledge model

算回路はいつ演算を開始してよいのか，制御回路は演算結果がいつ利用可能になるかも分からない．そのため非同期式回路で同様の制御をするには，要求信号と応答信号という2つの信号を新たに導入する必要がある．図1の構成を例に要求信号と応答信号の機能を説明する．まず制御回路は動作させたい演算回路に対して演算種別の信号と共に要求信号を送る．演算回路は要求信号を受け取るとその時指示された演算種別に従って演算を開始し，演算が終了すると応答信号を返す．制御回路は応答信号が返ってきたことで演算結果が利用可能になったことが分かる．このように要求信号と応答信号を用いたハンドシェークによる回路の動作を制御する方式を要求応答方式と呼ぶ．

### 2.2　データ転送方式

同期式回路はデータ転送も単純で，制御回路が転送元への出力の指示と転送経路の指示を送るだけで良い．あとは制御回路がクロックを数えることで転送先へのデータの到着を知ることが出来る．非同期式のデータ転送では制御回路が転送元に要求信号を出し，転送元からデータが転送先に送られ，転送先が応答信号を返す，という手順で行なわれる．この時，転送先はデータが到着したことを認識できなければならない．そのためにはデータ自身に時間情報を付加する必要がある．

この時間情報を付加する方式の一つとして，データを2線式符号で符号化すると共に連続するデータを区別するために間にスペーサを挟む2線2相式がある (表1参照)．例えば図2では1-bitのデータ $x$ を $(x,\bar{x})$ という2線対で表し，$1,1,0,1$ という値を間にスペーサを挟みながら転送している．データの転送先では2線対が2線式符号になるとデータの到着を，$(0,0)$ になるとスペーサの到着を認識することが出来る．この2線2相式は信号線や信号線を駆動するゲートの遅延が未知であっても有限であれば正しくデータ転送をすることが可能である．

| 情報 | | 2線式表現 |
|---|---|---|
| 符号語 | 0 | 01 |
| | 1 | 10 |
| スペーサ | | 00 |
| 非符号語 | | 11 |

表1　2線2相式の符号
Table 1　2-rail 2-phase encoding

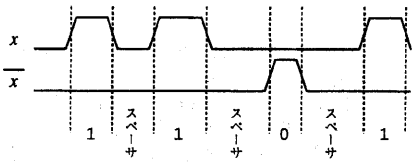


図2　2線2相式のデータ転送
Fig. 2　2-rail 2-phase data transfer

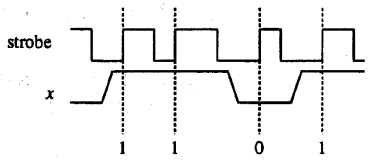


図3　束データ方式のデータ転送
Fig. 3　Bundled Data data transfer

この他に時間情報を付加する方式として，任意のビット数のデータ線に対して時間情報を表す信号線を1本だけ付加する束データ方式がある．例えば図3では，データ線 $x$ に新しいデータを出力した後に時間情報を表すstrobe信号を立ち上げることでデータ転送を実現している．データの転送先ではstrobe信号の立ち上がりでデータの到着を認識出来る．図2と図3を比べると分かるように，2線2相式に比べて束データ方式はスペーサが無い分高速である．しかし信号線や信号線を駆動するゲートの遅延及び遅延の変動が既知である必要がある．

他にもいくつかの非同期式データ転送方式がある．非同期式プロセッサのチップ外部とのデータ転送に限れば，ピン数の制限や既存のIC/LSIに2線式のものが無いことなどから束データ方式が主に用いられる．

## 3.　ソフトウェア上の相違点

非同期式プロセッサを用いてコンピュータシステムを構成した場合でもソフトウェア上の相違点は無い．ソフトウェアから見た時に現れる違いはアーキテクチャに基づくものであり，同じノイマン型のアーキテクチャを持つプロセッサであれば同期式プロセッサでも非同期式プロセッサでもプログラムは同じである．

特に違いを挙げるとすれば，「プロセッサに与える命令を処理するのに要する時間 (クロック数) が予測不

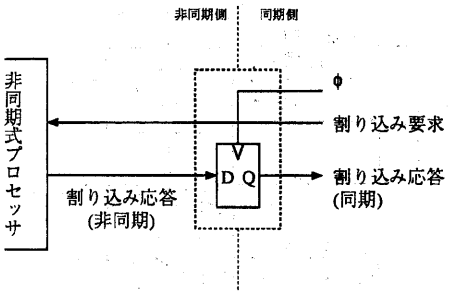


図4　割り込み信号のインタフェース
Fig. 4　interface of interrupt signal

能」という点がある．これは非同期式プロセッサが基準信号となるクロックを持たず自律的に動作するため，同じ演算であっても温度・電圧といった動作する環境や演算の対象となる値によって演算時間が変化するためである．しかしながらパイプライン化され，キャッシュメモリを持ち，並列実行や out of order 実行を行なっている現在のプロセッサでは，命令の処理に要する時間をプログラマが意識することは無意味であるので特に問題とはなり得ない．

## 4. ハードウェア上の相違点

同期式と非同期式のハードウェア上の大きな違いは，クロック信号の有無の他に2節で述べた要求応答方式とデータ転送方式にある．

このため非同期式プロセッサとメモリや I/O デバイスを接続するには非同期式と同期式を接続する特別なインタフェースが必要となる．本節では非同期式プロセッサと周辺とを繋ぐ非同期・同期インタフェースを幾つかに分類し，各々について考察する．

### 4.1 プロセッサ外部からの要求信号

同期式のプロセッサに於いても割り込み要求やバス要求といった信号については要求信号と応答信号によって何らかのハンドシェークを行なっている．

クロックに同期した割り込み要求応答信号と非同期式プロセッサのインタフェースの例を図4に示す．同期側の割り込み要求信号はそのまま非同期式のプロセッサに入力される．非同期側の割り込み応答信号はクロック信号でサンプリングしてから同期側に返される．

この時注意しなければならないのは，同期側の割り込みを要求した回路は割り込み応答が返ってくるまで待つ必要があることである．クロックのある同期式プロセッサであればある一定のクロック数の間に割り込み応答が返ることを保証できるが，クロックのない非同期式プロセッサではいつ応答が返ってくるか知る方法がないためである．またパルス信号によって割り込み要求を発生する場合はパルス信号をレベル信号に変換する必要がある．

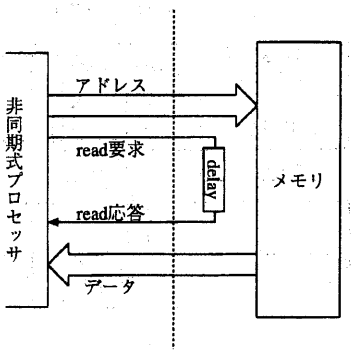


図5　非同期メモリの読み込み
Fig. 5　read operation for asynchronous memory

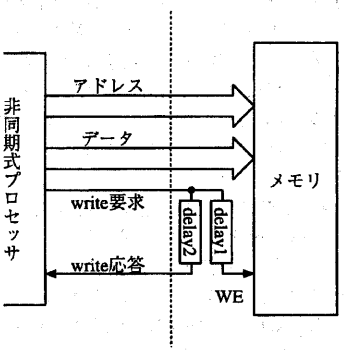


図6　非同期メモリの書き込み
Fig. 6　write operation for asynchronous memory

### 4.2 クロック信号を持たない回路とのデータ転送

クロック信号を持たない回路とのデータ転送の代表例として非同期のメモリアクセスが挙げられる．

非同期の Static Memory は，読み込み時にはアドレスが確定してから一定の時間後にデータが確定し，書き込み時にはアドレスとデータが確定してから一定時間後に書き込みが終了する．この動作を非同期式プロセッサの要求応答方式による制御と束データ方式に当てはめると，読み込みの場合が図5に，書き込みの場合が図6となる．

図5では読み込みたいアドレスと共に read 要求を出力する．メモリはアドレス確定後一定の時間が経つとデータを確定させる．read 要求の信号はメモリがデータを確定させるまでの時間を遅延素子 (delay) で測ってから，read 応答を返す．非同期式プロセッサは read 応答を受け取ってからメモリから来たデータを記憶することで正しいデータの読み込みを実現する．

図6では書き込みたいアドレスとデータと共に write 要求を出力する．write 要求の出力 (アドレスとデータの確定) から遅延素子 (delay1) でセットアップ時間を測って WE 信号を生成する．またメモリへの書き込

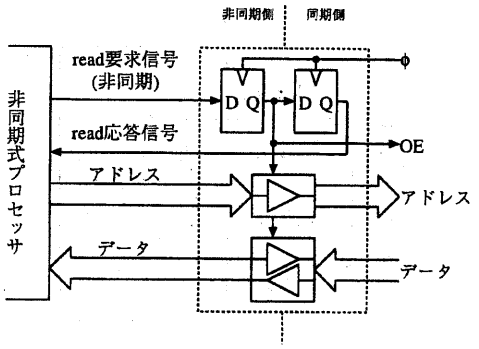


図7　クロックのサンプリングによる読み込み
Fig. 7　read operation with clock sampling

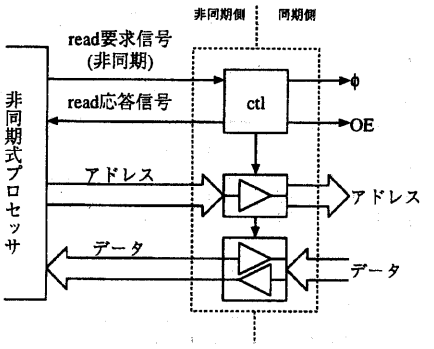


図8　read 要求信号からのクロック発生
Fig. 8　clock generation from read-request

みが終了するまでの待ち時間を WE 信号とは別の遅延素子 (delay2) で測って write 応答信号を生成する．非同期式プロセッサは write 応答を受け取るとメモリへの書き込みが終了したものとして次の処理に移る．

メモリ以外の I/O 関係のデバイスでも同様の回路で非同期式プロセッサとインタフェースをとることが可能である．

### 4.3　クロック信号を必要とする回路とのデータ転送

クロック信号を必要とする回路とのデータ転送の主なものとしては，同期バスに接続されるデバイスやクロックに同期してデータを転送する Synchronous RAM や RAMBUS DRAM が挙げられる．

これらの同期式の回路と非同期式プロセッサの間でデータ転送を行なう場合，クロックの周波数やその変動の許容量から 3 種類の方法が考えられる．

(1)　非同期式プロセッサの動作周期の最悪値がクロック周期がよりも短い場合

この場合，クロックを用いる回路からのデータ転送を非同期式プロセッサが取りこぼすことは無い．図7に示すように読み込み処理は read 要求信号をクロックでサンプリングして Output Enable(OE) 信号を生成する．read 応答信号は読み込みに要する時間に合わせて数クロック遅らせてから非同期式プロセッサに返す．

書き込みの場合も同様で，OE の替わりに Write Enable(WE) 信号を生成する．

同期バスに接続されるメモリや I/O デバイスとのデータ転送がこれにあたる．

(2)　非同期式プロセッサの動作周期とクロック周期は同程度だが，クロックは常時必要とせず周期が変動しても構わない場合

この場合，非同期式プロセッサの動作周期が最悪値になると，クロックを用いている回路からの転送を非同期式プロセッサが取りこぼす可能性がある．しかしクロックが常時必要でなく周期の変動が許容される場合には，非同期式プロセッサが追従可能な周期のクロックを非同期式プロセッサ自体が発生することで対応出来る．

読み込みの場合は図8のように読み込み要求信号から，クロックや OE，応答信号を生成する．この時クロック信号のデューティ比やクロックが Low の期間，Hi の期間の最小値を保証する付加回路 (ctl) が必要となる．書き込みの場合も同様である．

記憶をリフレッシュする必要のない同期式の RAM がこれにあたる．

(3)　非同期式プロセッサの動作周期とクロック周期は同程度で且つ，クロック周期の変動も許容されない場合

この場合，非同期式プロセッサの動作周期が最悪値になると，クロックを用いている回路からの転送を非同期式プロセッサが取りこぼす可能性がある．クロックの周期を変動させられないので非同期式プロセッサが独自にクロックを生成することも不可能である．

そこで図9に示すように非同期式プロセッサと一定のクロックを必要とする回路の間に (2) に分類されるメモリ (2) を挟む．

図9の読も込み動作は次の通り．read 要求信号が回路 ctl–1 に来ると，クロック周期の変動を許容しない同期 (3) 側とクロックの変動を許容する同期 (2) 側の間のデータ転送を開始する．このデータ転送には正確なクロック$\phi 1$ を用いる．この転送が終ると回路 ctl–1 から回路 ctl–2 に転送要求信号が送られる．転送要求信号が回路 ctl–2 に来ると，クロックを変動を許容する同期 (2) 側と非同期式プロセッサの間のデータ転送を開始する．このデータ転送には転送要求信号から生成したクロック$\phi 2$ を用いる．この転送が終了すると read 応答信号が非同期式プロセッサに返されて読み込み操作が終了する．書き込みの場合も同様である．

このようにクロック周期の変動を許容するメモリをクロックを切り替えて使うことで動作周期の不一致

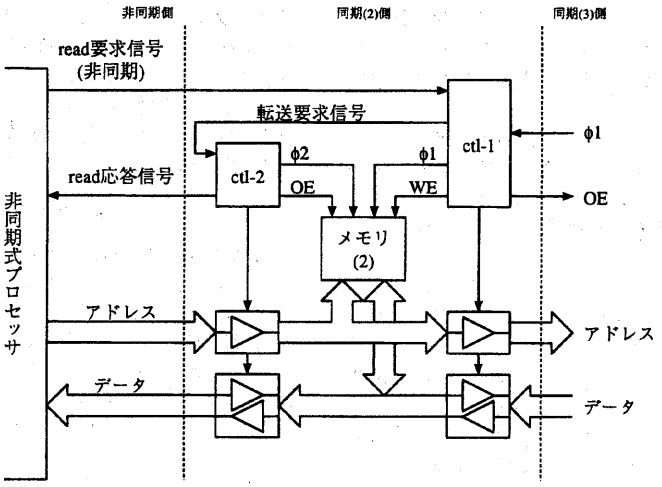


図 9　高速な同期転送の場合
Fig. 9　data transfer from high-speed synchronous transfer

を吸収する.
RAMBUS DRAM のように非常に高速な同期バスを用いてデータ転送を行なう場合が該当する. RAMBUS DRAM からの転送ではメモリ (2) には高速なキャッシュメモリになると考えられる.

## 5. TITAC-2 評価用システム

### 5.1 TITAC-2

非同期式プロセッサ TITAC-2 は南谷研究室において設計・製作された 32-bit 非同期式プロセッサで以下のようなアーキテクチャを持つ.

- アドレス・データともに 32-bit 幅
- MIPS R2000 準拠の命令セット
- 5 ステージのパイプライン構成
- 8 KByte の命令キャッシュ
- 31 本の汎用レジスタ
- 例外処理機能
- 外部割り込み機能
- メモリ保護機能

TITAC-2 チップの製造は $0.5\mu m$, 3 層メタルの 3.3V CMOS Process で行ない, $12.15mm \times 12.15mm$ の die に 496,367 個のトランジスタと 8.6KByte のメモリマクロを搭載した.

製造後のチップの評価によって以下の結果を得た.

- 54 VAX MIPS(dhrystone ver 2.1, 電源電圧 3.3V)
- 電源電圧が 1.5V ～ 6.0V の間で正常に動作
- チップ外部の温度が $-196°C$ ～ $+100°C$で正常に動作

### 5.2 ソフトウェア

TITAC-2 評価用システムのためのプログラム開発に

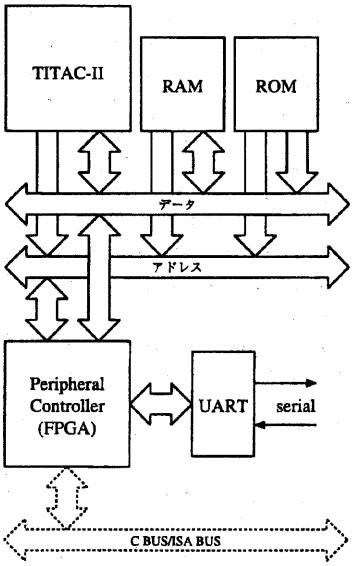


図 10　TITAC-2 評価用システム
Fig. 10　the evaluation system of TITAC-2

は C 言語を使用している. コンパイラには TITAC-2 用に変更した GNU-C を用いている. dhrystone ver 2.1 や簡単な PCM 音声の再生プログラム, マンデルブローの計算, BIOS の開発などを行なっているが, 同期式のシステムと何ら変わる所はない.

### 5.3 ハードウェア

TITAC-2 評価用システムの大まかな構成を図 10 に示す.

この評価用システムで用いたハードウェアの非同期・同期インタフェースをまとめると以下のようになる.

これらの非同期・同期インタフェースは正常に動作していることが確認されている．また 10MHz 程度のクロックで動作する同期バスとの接続も現在検討中である．

- RAM/ROM アクセスに 4.2節のインタフェース
  TITAC-2 評価用システムではクロック入力を持たない非同期のメモリを用いている．
- FPGA からの割り込み要求/応答に 4.1節のインタフェース
  FPGA の動作クロックでサンプリングされる．
- FPGA を通した UART や外部 I/O のアクセスに 4.3節の (1) のインタフェース
  FPGA の動作クロックは 10MHz で TITAC-2 チップの動作速度に比して十分遅い．

より高速な非同期式プロセッサを用いて同期式と同等な高性能コンピュータシステムを構成する場合の非同期・同期インタフェースの適用は以下のようにまとめることが出来る．

- 1MByte 程度の容量の高速非同期 SRAM を用いて 2 次キャッシュを構成．4.2節のインタフェースを備えた周辺制御チップを用意すると共にプロセッサ自体もこのインタフェースを持つ．
- 主記憶として Synchronous DRAM か RAMBUS DRAM を用い，これらの RAM へのインタフェース (4.3節の (2) か (3)) は周辺制御チップが備える．
- 主記憶の一部を構成する ROM としては高速な非同期 ROM を 4.2節のインタフェースで用いる．
- 様々な I/O デバイスを接続する同期バスと非同期式プロセッサの接続には，4.3節の (1) のインタフェースを周辺制御チップが備える．
- 割り込みの制御は周辺制御チップで行ない，非同期式プロセッサとは 4.1のインタフェースで接続する．

## 6. ま と め

非同期式プロセッサを用いてコンピュータシステムを構成した場合と同期式プロセッサを用いた場合との違いを明らかにすると共に，32-bit 非同期式プロセッサ TITAC-2 の評価用システムにおいて用いたメモリや I/O デバイスのインタフェースから，既存のメモリや I/O デバイスと非同期式プロセッサを接続するインタフェースについて考察した．この結果から適切な非同期・同期インタフェースを用いれば殆んどのメモリや I/O デバイスは接続可能であると考えられる．

しかし，既存の同期式プロセッサを非同期式プロセッサに差し替えただけで動作させるためには，非同期式プロセッサ内部に非同期・同期インタフェースをも内蔵させる必要があるので無駄が多く最良の方法ではない．同期式のプロセッサが専用の周辺制御チップを用いるのと同様に，非同期式プロセッサでも非同期・同期インタフェースを備えた専用の周辺制御チップを用いて同期バスと様々な I/O デバイスを接続する方法が望ましいと考えられる．

なお，本研究の一部は科研費補助金基盤研究 (B)09480049，及び（株）半導体理工学研究センターとの共同研究によるものである．

## 参 考 文 献

1) 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 (1993).

2) Martin, A. J., Burns, S. M., Lee, T. K., Borkovic, D. and Hazewindus, P.J.: The Design of an Asynchronous Microprocessor, *Advanced Research in VLSI: Proceedings of the Decennial Caltech Conference on VLSI* (Seitz, C.L.(ed.)), MIT Press, pp. 351–373 (1989).

3) Tierno, J. A., Martin, A. J., Borkovic, D. and Lee, T. K.: A 100-MIPS GaAs Asynchronous Microprocessor, *IEEE Design & Test of Computers*, Vol. 11, No. 2, pp. 43–49 (1994).

4) Furber, S. B., Day, P., Garside, J. D., Paver, N.C. and Woods, J.V.: A Micropipelined ARM, *Proceedings of VLSI 93* (Yanagawa, T. and Ivey, P. A.(eds.)), pp. 5.4.1–5.4.10 (1993).

5) Furber, S. B., Day, P., Garside, J. D., Paver, N. C. and Temple, S.: AMULET2e, *Embedded Microprocessor Systems* (Muller-Schloer, C., Geerinckx, F., Stanford-Smith, B. and van Riet, R.(eds.)) (1996). Proceedings of EMSYS'96 - OMI Sixth Annual Conference.

6) Nanya, T., Ueno, Y., Kagotani, H., Kuwako, M. and Takamura, A.: TITAC: Design of a Quasi-Delay-Insensitive Microprocessor, *IEEE Design & Test of Computers*, Vol. 11, No. 2, pp. 50–63 (1994).

7) Nanya, T., Takamura, A., Kuwako, M., Imai, M., Fujii, T., Ozawa, M., Fukasaku, I., Ueno, Y., Okamoto, F., Fujimoto, H., Fujita, O., Yamashina, M. and Fukuma, M.: TITAC-2 : A 32-bit Scalable-Delay-Insensitive Microprocessor, *HOT CHIPS IX*, Stanford, pp. 19–32 (1997).

8) Takamura, A., Kuwako, M., Imai, M., Fujii, T., Ozawa, M., Fukasaku, I., Ueno, Y. and Nanya, T.: TITAC-2 : An asynchronous 32-bit microprocessor based on Scalable-Delay-Insensitive model, *Proc. International Conf. Computer Design (ICCD'97)* (1997).