

非同期式 VLSI システム設計

南 谷 崇†

今日のデジタル VLSI システムはほとんどすべて同期式である。それは、同期式システムのほうが設計が簡単であり性能も良いとこれまで考えられてきたからだ。しかし、この数年、世界の各地で非同期式システム設計の研究が急速に盛んになってきた。また、いくつかの非同期式マイクロプロセッサも試作され、その実測性能が報告され始めた。本稿では、非同期式システムの必然性、その設計技術の現状を述べ、21 世紀における「VLSI 設計の新しい大地」への可能性を提示する。

Asynchronous VLSI Systems Design

TAKASHI NANYA†

With the wire-delay problem moving into dominance in VLSI chip design, asynchronous circuit/system design is enjoying a worldwide resurgence of interest, having been stimulating an increasing amount of work in this area for the last decade. Scientific American described this situation as "reviving a challenger to the modern microchip" in its issue of June 1995. In this note, we review recent developments of asynchronous microprocessor design and show a new perspective for VLSI design methodologies in the 21st century.

1. はじめに

デジタルシステムの動作方式は、クロックと呼ぶ特別な同期信号を使うか否か、で同期式と非同期式に分けられる。言うまでもなく、今日のデジタル VLSI システムはほとんどすべて同期式である。それは、同期式システムのほうが設計が簡単であり性能も良いとこれまで考えられてきたからだ。しかし、この数年、世界の各地で非同期式システム設計の研究が急速に盛んになってきた。また、いくつかの非同期式マイクロプロセッサも試作され、その実測性能が報告され始めた。本稿では、非同期式システムの必然性、その設計技術の現状を述べ、21 世紀における「VLSI 設計の新しい大地」を拓く可能性を示す。

2. 誰がクロックを発明したか?

クロックを用いない非同期式デジタルシステムの歴史は古い。コンピュータの黎明期である 1940 年代にはリレーあるいは真空管を用いていくつかの非同期式演算器が作られた。それらの開発経験を基にいくつかの古典的な非同期式回路の理論も 1950 年代には生まれている。そもそも順序回路の概念が登場する前の多くの電磁リレー式制御装置は非同期式だったはずだ。

非同期式システムがまず始めに存在した。その後、誰かがクロックを発明したのだ。

コンピュータがいわゆる第 2 世代 (トランジスタ時代) に移ってからは、素子品質が均一になりクロック同期による論理設計のほうが容易で性能も良いことからいわゆる CPU (中央処理装置) 部分はすべて同期式となった。それでも、IC 時代の初期まではまだ入出力/周辺部分などに設計者の名人芸的な非同期式設計が見られた。1970 年代以降の LSI/時代に入るとそれも姿を消し、今日では世の中の中すべてのデジタルシステムは同期式であると言って良い。クロックを使用すると信号変化に伴う過渡現象を隠すことができ、各時刻のシステム状態を明確に定義できるため論理設計やテストが簡単になる。従ってその自動化も容易である。実際、同期式設計スタイルをターゲットとした商用 CAD ツールの急速な進歩と普及が今日までの大規模デジタルシステムの設計を支えてきたと言える。

ところが、最近、非同期式デジタル回路/システム設計が、再び、多くの研究者、技術者の関心を集め始めている。特にこの数年、欧米を中心に企業でも研究・開発が急速に盛んになってきており、すでに一部の LSI 製品の内部設計に取り入れる動きが進んでいる。こうした状況を Scientific American 誌は 1995 年 6 月号で "reviving a challenger to the modern microchip" (邦訳: 日経サイエンス 1995 年 8 月号「非同期プロセッサの復活」) と表現した。

† 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

3. なぜ非同期式か？

なぜ、非同期式プロセッサが復活し始めたのか？ まず第一に、半導体・集積回路技術の驚異的な進歩により素子速度が向上する一方でシステム規模が大きくなった。このため、VLSIシステム設計において、素子遅延よりむしろ配線遅延が支配的要因になってきたことがある。将来の超高速素子をシステム性能の向上に活用しようとしたとき、システム全域にクロックを分配する同期式システムには明らかな性能限界が存在する。現在すでに性能向上の飽和領域に近づきつつあり、クロック分配などのタイミング制御に多大の設計コストがかかるようになってきている。これに対して非同期式システム設計にはクロックによる性能限界は存在せず、クロック・スキューの問題もない。そのシステム性能は、同期式のような最大遅延ではなく、局所計算の平均遅延だけで決まる。従って、非同期方式によれば、素子速度向上の効果をそのまま享受した高速計算の可能性に加えて、予測不能な環境変化(電源電圧、温度など)に起因する遅延変動の影響を受けない高信頼性システムの実現が期待できる。

第二に、携帯用電子機器などを始めとして、電力消費低減を要求するデジタルシステムの応用分野が急拡大していることである。またハイエンドMPUの電力消費低減も必須である。現在の主流であるCMOSゲートの電力消費は、近似的には、回路中の信号遷移の量、すなわち、単位時間当たりに充電/放電の起きる配線数、に比例する。同期式システムでは、システム全域に分配されるクロック配線が引き起こす信号遷移による電力消費が大きな割合を占めるのに対して、非同期式システムでは与えられた計算の実行に必要な時と順序で必要な場所だけ信号遷移は起きない。動作が要求されない部分回路では信号遷移が起きず、その間の電力消費はごくわずかな漏れ電流によるものだけである。従って、性能を犠牲にすることなく電力消費を低減できる可能性があり、多くの企業が注目しているのは主にこの点である。

4. 性能？ それともタイミング信頼性？

デジタルシステムを設計する場合、素子及び配線の遅延に関する情報が必要である。遅延は様々な要因で変動する。論理設計手法、レイアウト、プロセス技術、実装技術など、設計製造段階での変動要因に加えて、電源電圧変動、温度などの稼働環境によって大きく変動する。さらに処理されるデータにも依存する。システムの生涯を通じて実際に起こり得る遅延の大きさを推定し、その振るまいを表す遅延モデル(遅延仮定)に基づいて設計を進めることになる。このとき、どの遅延モデルを前提に設計するかで出来上がるシステムの性能とタイミング信頼性は大きく左右される。

もし、同期式システムや束データ方式のデータバスのように、設計時に推定した最大遅延はシステムの生涯を通じて不変であると楽観的に仮定して設計を行うと、回路構成の効率性は良くなるが、遅延変動に対する信頼性を保証できない。一方、DIモデル¹⁾やQDIモデル²⁾のように、素子遅延や配線遅延はシステムの生涯を通じて上限値の予測も不可能なほど自由に変動し得ると悲観的に仮定して設計すると、遅延変動に対する信頼性は完全に保証するが、回路量が増えたり、性能低下が生じる。

配線遅延が支配要因となる今後のVLSI技術を考えると、システムの設計/製造段階、稼働環境において予測不能な遅延変動を招く可能性は大きくなる。従って楽観的モデルは非現実的だろう。一方で、悲観的モデルのように、各要素遅延が全くばらばらに自由に増減すると考えるのもまた非現実的である。

従って、設計の前提となる遅延モデルは素子技術、システム実装技術、動作環境に対して十分なレベルの性能とタイミング信頼性を達成するものでなければならない¹⁾³⁾。

5. 非同期式データバス

同期式と非同期式の違いは、2値信号をレジスタ、メモリなどの記憶素子に書き込む際に共通クロックを用いるかどうか、という点にある。デジタルシステムの基本動作はレジスタ間データ転送である。同期式では、転送元レジスタのデータはデータ変換部(ALUやメモリなど)を経て加工され、クロックパルスの到着に同期して転送先レジスタへ書き込まれる。(同じクロックパルスで転送元レジスタも書き換えられ、次のデータが送出される。)すなわち、クロックパルスがデータを書込むタイミングを定めている。このため、クロックパルスの到着する前に有効なデータが転送先レジスタの入力端子へ到着していなければならない。システム全体の動作も共通クロックに合わせた周期で進む。

これに対して非同期式では、データ転送が必要(かつ可能)になったら転送元レジスタへ要求信号を出す。その結果、転送元レジスタから送出されたデータは変換回路を経て転送先レジスタへ書き込まれ、その完了を示す応答信号が生成される。この応答信号が次のデータ転送を要求するトリガーになる。

クロックなしで、 N ビット幅のデータを転送先レジスタへ書き込むタイミングをどう決めるか？これは非同期式論理設計における重要な選択の一つである。一般にデータバスの遅延は、実行される命令、データによって、またビット毎にも、ばらつきがある。そこでおおまかに次の二つの極端な場合に分ける。

(1) 遅延のばらつきが非常に大きい場合:

1ビットデータを2本の信号線で表す2線

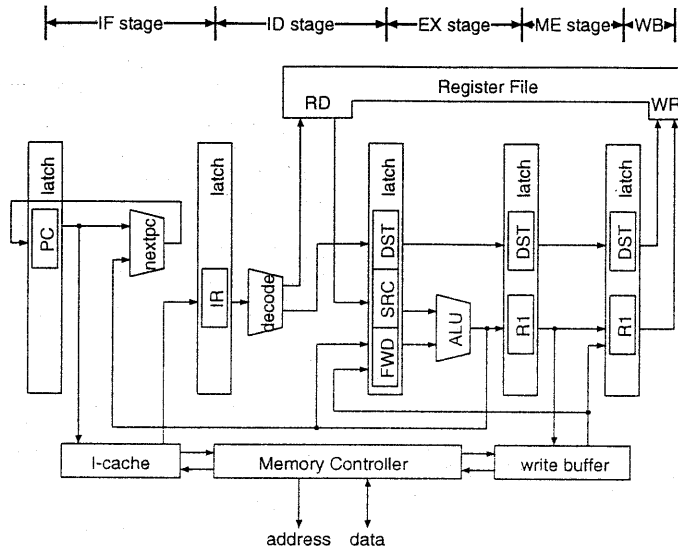


図1 TITAC-2のプロセッサ構成
Fig. 1 TITAC-2 Pipeline Structure

式データ表現を用いる。従って、データ幅は $2N$ になる。ビット毎に初期状態を $(0,0)$ として、有効なデータの到着 (書き込みタイミング) を次のような遷移で表現する。

$(0,0) \rightarrow (0,1)$ '0' が到着
 $(0,0) \rightarrow (1,0)$ '1' が到着

$(0,1), (1,0) \rightarrow (0,0)$ 初期状態へ復帰
データバス上のどのインターフェースでも、データ転送を実行するために初期状態 $(0,0)$ から有効データ $(0,1)$ または $(1,0)$ へ遷移する期間 (稼働相と呼ぶ) と次のデータ転送に備えるために初期状態へ復帰する期間 (休止相と呼ぶ) を交互に繰り返す。これを 2 線 2 相式と呼ぶ。

(2) 遅延のばらつきが非常に小さい場合:

N ビット・データの遅延がほぼ一定で、その上限値 D が既知の場合、値 D を考慮した適当な遅延素子を挿入した 1 本の ready 信号を N ビットデータに付加する。ready 信号の $0 \rightarrow 1$ 遷移が転送先レジスタへの有効データの到着を示す。これを束データ方式と呼ぶ。

すなわち、非同期システムでは、共通クロックを使わない代わりに、 N ビットのデータバスを実現するのに、最も楽観的な場合で $(N+1)$ ビット、最も悲観的な場合には $2N$ ビットのデータ幅が必要になる。実際には、この範囲内のデータ幅で性能と信頼性のトレードオフを考慮した種々の設計があり得る。

6. 非同期式プロセッサ TITAC-2

TITAC-2 は 32 ビットのマイクロプロセッサである⁴⁾⁵⁾⁶⁾。アーキテクチャとして MIPS-R2000 を採用した。その理由は、

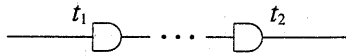
- (1) アーキテクチャ本が出版されている。
- (2) 実用の (同期式) チップが存在する。
- (3) C コンパイラが簡単に入手できる

などである。また、研究室メンバーで手に負える程度の設計規模になりそうだとの見通しも重要な決め手になった。実現した命令セットは MIPS-R2000 とほぼ同じである。異なる点は、multiply 命令、divide 命令の結果を下位 32 ビットだけとしたこと、分岐命令の遅延スロットを 2 としたこと、特権命令を変更したこと、などである。プロセッサ構成 (図 1) は、R2000 と同様、命令フェッチ (IF)、デコード (ID)、実行 (EX)、メモリアクセス (ME)、ライトバック (WB) から成る 5 段パイプライン構成で、40 本の 32 ビットレジスタ (内、8 本はカーネルモードのみ)、8KB 命令キャッシュ、例外処理、外部割り込み、記憶保護機構を備えている。

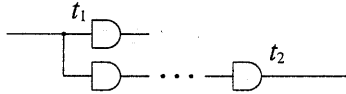
TITAC-2 では、楽観的過ぎも悲観的過ぎもしない、現実的なモデルとして、Scalable-Delay-Insensitive (SDI) モデルと呼ぶ次のような遅延仮定を導入し、それを基に論理設計を行った。

「任意の二つの回路要素 (ゲートまたは配線) $C1, C2$ の遅延を d_1, d_2 とすると、 $D =$

Specification: Transition t_1 precedes transition t_2
 (a) DI implementation: t_2 must be caused by t_1



(b) QDI implementation: t_2 may be caused by other fanout branch that shares its stem with t_1



(c) SDI implementation: t_2 may be caused by common stem t that also causes t_1 , if $K \cdot d_1 < d_2$

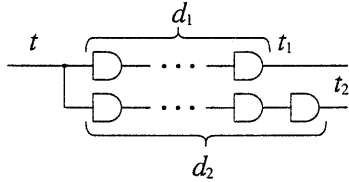


図2 SDIモデルによる回路実現
 Fig. 2 How SDI model works

d_1/d_2 をC2に対するC1の相対遅延と呼ぶ。回路の設計段階で推定される相対遅延を D_e で表わし、システムの生涯を通じて起こり得る実際の相対遅延を D_a で表わすと、 $R = D_a/D_e$ は相対遅延変動率を表わす。このとき、上限値(定数) K が存在して、 $1/K < R < K$ が常に成り立つ。」

すなわち、SDIモデルは、回路要素の絶対的な遅延変動には上限値を仮定しないが、相対的な変動には限りがある、と仮定する。

SDIモデルを適用する効果を説明しよう。今、二つの信号遷移 t_1 と t_2 に関する仕様で、 t_1 が t_2 より必ず先に起きなければならないとする。これを実現するために、DIモデルのもとでは、遅延に上限値がないので図2(a)に示すように、 t_1 を原因としなければ t_2 は決して起きないように回路を構成する必要がある。QDIモデルのもとでは、分岐配線には遅延差はないとするため(b)に示すように、原因となる幹を t_1 と共有する枝によって t_2 が起きてよい。SDIモデルのもとでは、(c)に示すように、 t_1 と t_2 の共通原因となる幹からの経路の遅延をそれぞれ d_1, d_2 としたとき、 $K \cdot d_1 < d_2$ が成り立つような回路構成であればよい。従って、遅延変動に対する信頼性を確保しつつ、速度性能を向上させることができる。

SDIモデルを適用する設計は以下の指針に従う。

- (1) システム全体を複数の機能ブロックに分割する。
- (2) 各ブロックおよびその接続をQDIモデルに基づいて設計する。

表1 MIPS-R2000との比較

Table 1 Comparison of TITAC-2 and MIPS-R2000

| | TITAC-2 | MIPS-R2000 |
|------------------------------------|------------------------|-------------------|
| Process | 0.5 μ m CMOS | 2 μ m CMOS |
| Routing layers | 3-metal | 2-metal, 1-poly |
| Designing method | Standard-cell | Full-custom |
| Die size | 12.15 \times 12.15mm | 8.5 \times 10mm |
| On-chip memory | 8.6KB SRAM | none |
| Transistor count | 496K | 100K |
| Clock frequency | — | 16.7MHz |
| Supply voltage | 3.3V | 5V |
| Power dissipation (cache disabled) | 2.11W | — |
| | 1.02W | < 2W |
| Throughput | 54.1 VAX MIPS | — |
| (cache disabled) | 26.5 VAX MIPS | 12 VAX MIPS |

- (3) 使用するプロセス技術およびブロックサイズから K の値を決定する。
- (4) 各ブロックで $K \cdot d_1 < d_2$ が成り立つような局所変換を行う。

SDIモデルを正当化するために、レイアウトには一定の制約が課される。TITAC-2では $K = 2$ とした。これを正当化するために、使用した0.5 μ m CMOSプロセスに対して、各ブロックは1.93mm \times 1.93mmの範囲に収まるようなレイアウト設計を行った。

7. TITAC-2 チップ

TITAC-2チップは、0.5 μ m、3層メタル、3.3V電源CMOSプロセスのスタンダード・セル・ライブラリ、及び新規に作成した専用マイクロ・セル(C素子、アービタ、2線式全加算器など)を用いて設計した。図3にそのチップ写真を示す。12.15mm(コアサイズは10.55mm)四方のチップ上に約50万トランジスタと8.6Kバイトのメモリマクロが集積されている。Dhrystone V2.1ベンチマークによる実測性能は、室温、電源電圧3.3Vで54.1VAX MIPS、消費電力2.11Wである。表1に、同じアーキテクチャのMIPS R2000とTITAC-2の比較結果を示す。このように同期式と非同期式を直接比較した例は少ないため、資料的価値はあるが、プロセス技術、設計環境、設計者の能力など、諸条件が全く異なるので、この比較結果から何らかの意味ある結論を導き出すのは必ずしも容易ではない。

TITAC-2の特徴はそのDelay-Insensitivityにある。遅延変動を生じさせるような環境の変動、例えば、電源電圧やチップ温度の変動に対しても正常に動作する。但し、当然のことながら、速度性能と消費電力は変化する。図4に電源電圧を1.5Vから4.0Vまで連続的に変化させた場合のTITAC-2の速度性能(MIPS)と消費電力(W)の実測値を示す。

8. 世界の動き

TITAC-2と同様な非同期式プロセッサ開発プロ

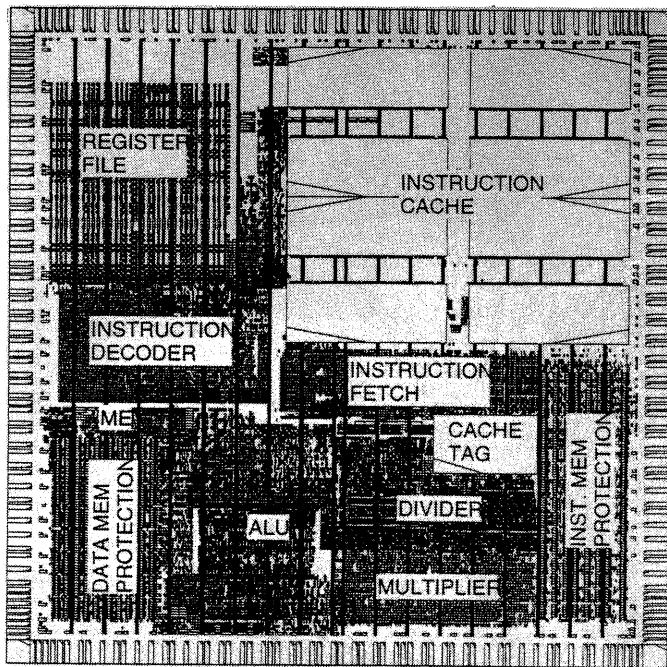


図3 TITAC-2のチップ写真
Fig. 3 Die Photo

プロジェクトとしては、英国 Univ. of Manchester の AMULET2e⁷⁾と米国 CalTech の MiniMIPS⁸⁾がある。AMULET2e は ARM 社の 32 ビット RISC プロセッサの非同期版であり、0.5 μ m、3層メタル、CMOS プロセスのフルカスタム設計で 1996 年 10 月にチップとして実現されている。電源電圧 3.3V での Dhrystone V2.1 ベンチマークで 40MIPS の速度性能であるが、その消費電力は 0.15W と少ない。一方、カリフォルニア工科大学の MiniMIPS は、MIPS-R3000 の命令セットを実現するが一部機能 (TLB、外部割り込みなど) を省略した簡易型非同期式 MIPS である。まだ 0.6 μ m CMOS プロセスによる設計段階にあり、チップはできていないが、シミュレーションによれば、3.3V 電源、電力消費 7W で 280MIPS の速度性能を達成している。非同期式システムの高速度性に関して勇気づけられる結果である。

9. むすび

大規模な非同期式システムの実現可能性について論じる時期は過ぎた。実用レベルのマイクロプロセッサをクロックなしで実現する設計技術はすでに用意されている。今必要なことは、素子性能の向上をシステム

性能の向上に直線的に反映し得る非同期式システムの特徴を十分に引き出すアーキテクチャの開発と、それを実現する非同期式設計技術を現在の成熟した同期式設計と並ぶ選択肢の一つとして設計者に提供できる環境を構築することである。

高性能マイクロプロセッサは、ソフトウェアからプロセス技術まであらゆるレベルで最先端の技術を総合して実現される。現在の同期式マイクロプロセッサ開発に投じられている資金、人、時間を考えると、大学の一研究室で開発された TITAC-2、AMULET2e、MiniMIPS などでも示された非同期式マイクロプロセッサの性能は、まだ配線遅延がそれほど支配的とは言えない現在の技術水準でも、同期式プロセッサと十分比較し得るレベルに達していると言うことができる。今後、配線遅延が支配要因となる傾向が進めば、クロックを用いない非同期式システム設計の優位性が一層顕著になるとと思われる。

しかし、将来非同期式が同期式に取って代わる、と考えるのは非現実的である。21 世紀における「VLSI 設計の新しい大地」を展望するとき、最もあり得るシナリオは同期式と非同期式の融合であろう。応用分野によって性能、消費電力、信頼性のトレードオフは異なる。そのトレードオフに応じた同期/非同期融合

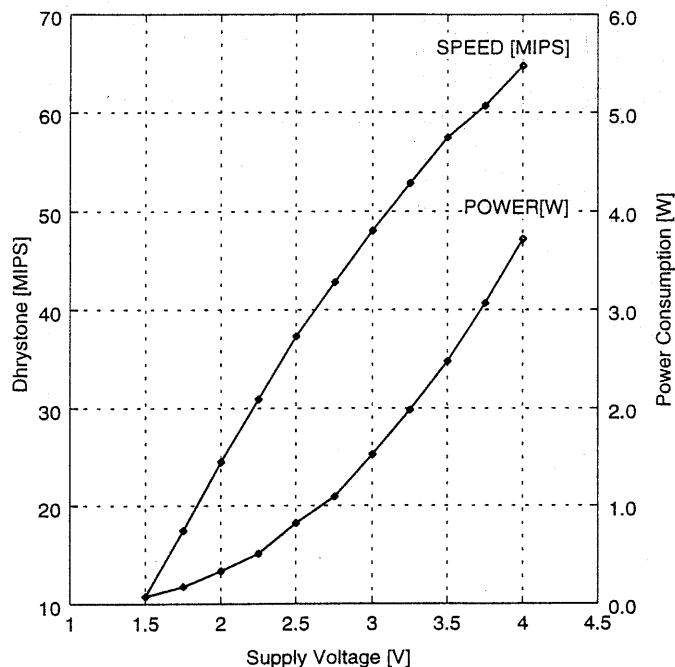


図4 TITAC-2の速度と消費電力
Fig. 4 Speed Performance and Power Consumption
for Variable Supply Voltage

アーキテクチャの開発が今後の課題である。

謝辞

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) 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).
- 3) Nanya, T.: Challenges to dependable asynchronous processor design, *Proc. Int. Symp. on Logic Synthesis and Microprocessor Architecture*, pp. 132-139 (1992).
- 4) 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, pp. 189-196 (1997).
- 5) 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).
- 6) 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).
- 7) S.B.Furber, J.D.Garside, S.Temple, J.Liu, P.Day and N.C.Paver: AMULET2e: An Asynchronous Embedded controller, *Proc. of ASYNC'97*, pp. pp.290-299 (1997).
- 8) A.J.Martin, A.Lines, R.Manohar, M.Nystrom, P.Penzes, R.Southworth, mings, U. and T.K.Lee: The Design of an Asynchronous MIPS R3000 Microprocessor, *Proc. ARVLSI'97* (1997).