TRON仕様チップ

東京大学　坂村　健

はじめに

TRON仕様チップの誕生は1980年代初頭という時代背景が関係している．1981年，時代としてはまだ16ビットマイクロプロセッサがやっと出始めたころであるが，この年のISSCC（International Solid State Circuits Conference）ではIntel，Bell Laboratories，HP，National Semiconductorの4社から新しい32ビットマイクロプロセッサが発表され大変な盛り上がりをみせた[1]．各々のアーキテクチャはユニーク．新しい時代を予見させるに十分なものであった．しかし，32ビットマイクロプロセッサはそのパワーに比べ応用となると当時はまだよく分かっていなかったというのが本当のところであろう．商用の32ビットマイクロプロセッサは1984年に初出荷され，エンジニアリングワークステーションはそれをいち早く採用していたが，他の需要は未知数であった．1984年はIntel iAPX286を採用したIBM PC/ATとMotorola MC68000を採用したApple Macintoshの代表的パーソナルコンピュータも発表されているが，どちらも16ビットプロセッサであった．このような背景のもと，新顔の32ビットプロセッサには量産に見合うだけの新しい応用が特に求められていたのである．

そんな当時，日本の半導体メーカは8ビット，16ビットプロセッサを海外半導体メーカのセカンドソースとして積極的に製造していたが，1980年代に入ってからは米国の半導体メーカがライセンスに消極的になり，関係はどちらかといえば悪化していたといえる．そのようなことがあって自社生産に移行しはじめていた．日立とMotorola，富士通とIntelなど提携はしていたが関係は疎遠になりつつあった．また米国のメーカとの係争も増えた．1982年にはIBM産業スパイ事件も起こった．当時日本の半導体メーカはまだどこも32ビットマイクロプロセッサを手がけておらず，方向を見定めていた．このような状況の中でTRONプロジェクトが1984年に始まった．

当時，ほとんどの組込み応用には8ビットマイクロプロセッサで十分でそれ以上はまず必要ないと思われていた．TRONプロジェクトが目指す[2]「身の回りのモノすべてにマイクロプロセッサが入り，それらはネットワークで結ばれる」という家電製品を中心とした生活用品にマイクロプロセッサが入るという将来構想はほとんどほら話か夢物語と思われていた．

TRONプロジェクトでは実は当初の開発の戦略として，下の階層はできるだけブラックボックスでいきたかった．組込み用リアルタイムカーネルは世界レベルで見て標準化の取組みが手薄だったのでITRONを最初に手がけ，他のOS（たとえばパーソナルコンピュータ用）やマイクロプロセッサの開発はできる限り延ばしたかった．

もちろんチップの開発には大いに興味があり，開発のための構想も発表していたが，開発パワーを考えると順序からいって実現はもっと後になるはずだと思っていた．BTRONも当初は既存OSの上でウインドウシステムなど上位レベルのものだけでも構わないと思っていたくらいである．アプリケーションなど上の階層の開発を先に行うべきで，それが成功すれば下の階層のOSやマイクロプロセッサの需要を自然に生むと考えていたのである．

TRON仕様チップはたまたま運良く機会が回ってきたため開発が早まった．きっかけを作ってくれたのは日立製作所で1985年のことである．筆者に32ビットマイクロプロセッサを共同開発しないかという誘いがあった．私もコンピュータアーキテクトとしてマイクロプロセッサの開発を行いたいと考えていたが，我が国には米国のように独立のファブリケーションや大学に開発のための設備が整備されているわけでもなく，実際開発するには障害が大きかった．この時国は第五世代コンピュータ，いわゆる人工知能の研究に大きな投資をしていて新しいマイクロプロセッサにはあまり興味を示していない．とにかくコンピュータは動くことが重要であり，ペーパーマシンで終わるのはつまらないと考えていたので，この共同開発の誘いは大きな魅力であった．このようなチャンスはそうあることではないと考えてこの誘いを引き受けることにした．すべてはここから始まった．

高性能マイクロプロセッサに対する市場の需要はパーソナルコンピュータを中心にせいぜい16ビットであったが32ビットに向けたビジネスはすでに始まっていた．日本のメーカの判断としては，米国のセカンドソースは不可能になりつつあり，といって単独一社オリジナルアーキテクチャでは特にソフトウェア開発のリスクが大き過ぎる．本心は一社で立ち向かいたいがリスク分散のた

	GMICRO/100	GMICRO/200	GMICRO/300	GMICRO/400	GMICRO/500	TX1	TX2	O32	MN10400
開発メーカ名	三菱	日立	富士通	三菱	日立	東芝	東芝	沖	松下
クロック MHz	25	25	33	40	66	16	25	33	20
MIPS (ピーク) (平均)	12.5 10.1	12.5 12.0	33 32	80 45	130	8 4	25 12	15 10	20 8
ドライストーン V1.1 (回/秒)	16K	21K	58K	78.9K	228K		20K		19K
命令数	92	123	135	100	126	93	87	103	93
MMU	No	Yes	Yes	No	Yes	No	No	Yes	No
キャッシュ I: 命令ｷｬｯｼｭ D: ﾃﾞｰﾀｷｬｯｼｭ S: ｽﾀｯｸｷｬｯｼｭ	256B (I)	1KB (I) 128B (S)	1KB (I) 1KB (D)	4KB (I) 4KB (D)	8KB (I) 8KB (D)	No	No	1KB (I) 1KB (D)	1KB (I)
パイプライン	5 stage	6 stage	5 stage	5 stage	5 stage	5 stage	4 stage	6 stage	4 stage + storebuffer
トランジスタ数	340K	730K	900K	1,485K	1,650K	450K	350K	700K	400K
製造技術 (μm)	1.0	1.0	0.8	0.5	0.6	1.0	1.0	0.8	0.8
パッケージ	135pin PGA 152pin QFP	135pin PGA	179pin PGA	160pin QFP	256pin QFP	155pin PGA 144pin QFP	184pin QFP	208pin PGA	144pin PGA 148pin QFP

表-1　TRON仕様に基づいて作られたマイクロプロセッサー覧

めできれば何社かで連合したいといったところが本音であろう．そこで仕様がオープンなTRON仕様チップなら，リスクも分散できるであろうと考え，後にはほとんどの日本の半導体メーカがTRON仕様チップ[3), 4), 16)]に大変な興味を示すようになった．1985年春から仕様の開発はスタートし，1986年秋には日立と富士通がTRON仕様に基づく32ビットマイクロプロセッサHF32の開発計画を発表，1987年3月には東芝もTX3の開発計画を発表，5月には三菱が日立・富士通グループ（GMICROグループ）に加わり，1988年12月には松下，沖もTRON仕様チップの開発を発表した[5) ～16)]（一覧は文献13）表-1参照）．

ところが開発リスク分散のための戦略が成功し開発に参加するメーカが増えるに従い組込み応用だけでなく，パソコンはまだしも，ミニコンピュータさらには大型コンピュータの下位機種にまで使おうという要望すら出てきた．スポンサは増えたが，要求も多種多様となり，スポンサ側の船頭が多く，すべてを満たす解は難しくなる．新しいアーキテクチャを提案しても，メーカが現在持っている資産は必ずサポートしてほしいという要望が出る．まずこちらの希望どおりにはいかなくなるのである．

仕様の変遷

1つはendian（エンディアン）の問題☆1である．big-endianとlittle-endianのどちらにするかという問題なのだが，製造技術的にはいずれを採用しても差があるわけではない．しかし，問題は大きい．IBMの大型コンピュータの流れを汲むものやMotorolaのマイクロプロセッサはbig-endianであるし，可変長サイズ命令で当時評判の高かったDEC VAX11シリーズやintelのマイクロプロセッサはlittle-endianであった．

TRON仕様チップは可変長命令をとっていたのでlittle-endianで考えるのが自然である．ところがスポンサである半導体メーカ連合はbig-endianを主張した．これは日立や富士通はIBMアーキテクチャの汎用コンピュータを販売していて，この新しいチップを廉価なミニコンやオフコンに利用しホストコンピュータと連携をとることを考えていたからである．

ここで強調しておきたいことは，仕様を作るのもオープンにしていたということである．この当時から仕様のレビューにプロジェクト参加者は意見を出すことができた．元となるオリジナルの案はメインアーキテクトである私がすべて作ったが，レビューによりそれを修正して最終案としていたのである．

1986年9月のメーカ側の議事録．litte-endianにするかbig-endianにするかは迷っている．すでにチップはbig-endianで設計を進めている．プロジェクトとして出す仕様書は最初に出たチップに合わせてくれるか？littleとbigの互換性について要検討．

10月のTRONの資料．この資料では仮にlittle-endianによる記述をしているが，little-endianに決定したわけではない．endianについては別に議論する．

11月のTRONの資料．endian変換に対するオーバヘッドを少しでも削減するためにバイトを逆順にするバイトリバース命令RVBYを導入する．これを実装できるならbig-endianにする．

結局この命令が実装されることになりTRON仕様チップはbig-endianとなった．TRONプロジェクトにおけるデータ形式の規格であるTAD（TRON Application Databus）☆2もbig-endianベースで作られた．ところが，後にスーパー301条の候補として上

☆1 複数ビットあるいは複数バイトにまたがるデータをアドレス付けしたり，メモリに入れたりする際，上位の桁に小さいアドレス（番号）を対応させるのがbig-endianである．これとは反対に下位の桁に小さいアドレス（番号）を対応させるのがlittle-endianである．すなわち問題なのは，メモリイメージのアドレスの進む向きに対して，上位桁のバイトから下位桁のバイトに向かって配置されるのか，その逆かである．複数バイトにまたがるデータでは，big-endianとlittle-endianの間でバイナリレベルのデータ互換性がない．

げられることになるBTRONがこの時ちょうど並行してintel iAPX286を使ったPCをプラットフォームとして開発を進行させていた．endian問題はチップをとるかBTRONをとるかの問題だったのである．すなわちintel系のマイクロプロセッサはlittle-endianであるのでbig-endianのTADをそのまま持ってくるとエンディアンのスワップ処理をしなければならない．これを非力なPC側で毎回行うのは大変だ．結局，準TADとしてlittle-endian形式のTADも作られることになった．

産業界から強く出された要求のもう1つの例が符号付きのBCD形式まで考慮した10進命令である．TRON仕様チップの命令には一般的なBCD形式の演算をする10進命令は含まれていた．しかし半導体メーカはオフコンに応用するときにCOBOLを使うので，もっと強力な10進命令が欲しいと主張した．そこで10進演算強化命令が仕様として追加されることになった．具体的には，最下位に符号（16進でa，c，e，fが正，b，dが負をあらわす）を持つ絶対値表現のBCD形式を扱う加減算命令が，拡張命令として仕様に追加された．正にCOBOL用である．

1986年9月には，素案をベースにメーカのCOBOLを使う事務処理部門で評価を行い，基本命令だけに比べ命令実行数が7割削減されるという大きな効果があることが分かった．また，そこから次のような機能追加の意見が出された．

(1) 加算減算をオペランドの符号により自動選択をして演算する．たとえばA＋(－B) という演算は実際には，A－Bとして演算する．
(2) 補数命令，比較命令の追加
(3) 10進データのチェック
(4) 4桁のPACK/UNPACK

そして，(1) (2) はDCで始まるDCADD，DCSUB，DCADDU，DCSUBU，DCX，DCADJ，DCADJU，DCADJX，DCCMP，DCCMPU，DCCMPX[16] という命令となった．(3) はこれらの命令の実行時にデータのチェックが行われ，各桁4ビットの数値が0～9であり，符号部はa～fでないと10進不正オペランド・トラップが働くようにした．また，PACK/UNPACKはそのままの名称で (4) を満たすようにした．TRONではレベル分けといって，仕様のサブセットを許していた☆3が，どのレベルまでを満足しているかを明示するシステムをとってユーザに互換性の範囲が分かるようにしていた．10進演算強化命令は〈〈L2〉〉という水準以上で実装すべき命令として入れたわけである．GMICRO/300にはこの命令すべてが入っていた．しかし，残念ながらTRON仕様チップを使ったオフコンは作られることはなかった．

TRON仕様チップのアーキテクチャ

マイクロプロセッサはとかく命令セットを含むアーキテクチャ（ISPアーキテクチャ）が強調されるが，普及のためにはISPアーキテクチャだけでは話にならない．しかも32ビットと大規模になってシステム指向がないと開発できなくなっていた．またISPアーキテクチャとは別に，周辺LSIや開発システムを充実しなければならない．

16ビットマイクロプロセッサのIntelとMotorolaとの競争でアーキテクチャ的にはMotorolaの方がリッチなのにIntelを採用する例が多いのは周辺LSIの充実が理由ということはよく知られており，マイクロプロセッサのみを開発したのではシステム開発で苦労すると思われたので，Intel x86シリーズの周辺LSIがそのまま利用できる案を提案した．だがこれは受け入れられなかった．

米国では本気で日本のメーカが連合軍を作って32ビットマイクロプロセッサ市場に攻めてくると恐れられた．またそうなった場合には米国のメーカですらTRON仕様チップの互換プロセッサを手がけなければならないだろうと考え，一部プロジェクトをスタートさせた米国の大手半導体メーカもあった．が心配したようなことは起こらなかった．日本のどのメーカもみな横並びにまさにオープンアーキテクチャの基に少しずつ性格の違うCPUを開発した．結局32ビットプロセッサ開発のテストベッドになり，開発人員育成が主になってしまった．各社が分担してCPUでなく周辺LSIを開発して充実させれば話は違ったものになったと思う．GMICRO/500などの当時の性能をみても現在のパーソナルコンピュータの主力チップの1つとなっていたのは間違いない．

命令セット（ISP）アーキテクチャは決定する立場に立つと一般の認識とは逆にそう重要とは思えなかった．RISC/CISC論争は盛んであったが，研究レベルは別として商用プロセッサになるとそう意味のあるものとも思えなかった．TRON仕様チップでRISCを採用しなかっ

☆2 TADはTRON Application Databusの略で，TRONアーキテクチャのデータ交換規約を指す．たとえばワードプロセッサの種類が違うとデータ交換ができなかったり修飾情報をうまく渡せないという，ようなことをなくすために，できるだけ高い水準でデータ交換できるように定めている．この中には多国語を扱う文字コード，図形などを扱う二次元データからはじまり，リアルタイムに進行する複数の音データの同期までを含む表現，住宅やビルなどの環境制御用のデータなどを含む．

☆3 TRON仕様チップでは，マイクロプロセッサ全般に亙るアーキテクチャを設計し，その上でサブセット化あるいは拡張ができるように仕様レベルを設けた．具体的に作られるプロセッサにはこのレベルを表示することにより，利用者はソフトウェア互換性について知ることができるようになっている．
〈〈L1R〉〉仕様：メモリマネージメントなしの標準仕様すなわち，実記憶のみを使うTRON仕様チップでサポートされるべき必要最小限の仕様
〈〈L1〉〉仕様：メモリマネージメント付きの標準仕様
〈〈L2〉〉仕様：拡張仕様．高機能命令および将来導入されたる特殊な命令

たのはオブジェクト効率の悪さが最大の理由であるが，RISC陣営の主張が誇大であったことに多くの人が気がつくのは10年以上経ってからである☆4.

TRON仕様チップのISPアーキテクチャの特長[16]には次のようなものがある.

(1) 命令がリッチで対称性の高いアーキテクチャ

32ビットマイクロプロセッサは大型コンピュータ並みのソフトウェアで苦しむと思われた．そこでアーキテクチャの構築にあたっては，まずソフトウェアが作りやすいことを念頭に置くべきであると考えた. 具体的には，コンパイラが作りやすいISPにしたかった. というのは，日本はソフトウェア特にコンパイラやOSなどの開発は得意ではない．RISCアーキテクチャに対して十分性能を出すようなオプティマイズコンパイラを作るのはなかなか大変である．そこでソフトウェアを作りやすいアーキテクチャを目指した.

命令の種類によらず利用できるオペランドの種類，アドレシングモードの対称性が高く，ほとんどの命令でメモリーメモリ間演算が可能である．しかも利用頻度の高い命令は短縮型という命令長の短い形式を用意していて，高速実行とプログラムサイズの短縮を実現している. 短縮型においても制限されるのはアドレシングモード，オペランドの即値，オペランドサイズだけで，レジスタ間の対称性は確保されている．異種サイズ間演算命令，サブルーチン命令などもソフトウェア開発を効率よく行うことだけでなく，高級言語によるアプリケーションの高速実行には欠かせないものである.

対称型命令セットの採用ということは，特に当時すでにシステム記述言語として主力のC言語に着目し，Cコンパイラが作りやすくなるような命令体系を採用し，汎用マイクロプロセッサの中ではCマシンに近い性格を持ち合わせていたといえよう．これらの特長は，後にTRON仕様チップのコンパイラを開発した米国コンパイラメーカGreen Hills Software社やMicrotec Research社がTRON仕様チップのコンパイラは大変作りやすかったと報告していた[22]ことからも実証されている.

(2) OS用命令

並行して開発しているTRON仕様のOSを効率よく動かすための命令群を持つこと．もちろん他の汎用OSにも適用は可能であった. 高速コンテクストスイッチ命令，キュー操作命令，可変長ビットフィールド命令，遅延割込み機構，遅延コンテクスト例外発生機能などである.

(3) 64ビットへの拡張性

64ビットデータ，64ビットアドレシングへの拡張を最初から考慮してある．命令語中のオペランドサイズフィールドはすべて64ビットが指定可能なビット割当てがしてある．また64ビット用命令のオペレーションコードもリザーブされている.

☆4 RISCの提唱者の1人David Pattersonは，「CISCにはVAX，iAPX432があたりx86はエレガントではないがシンプルなので違う」と考えていた．しかし業界紙が商用コンピュータで「RISCでないものはCISC」と扱ったことから混乱が始まったとしている[21].

マイクロプロセッサ普及の条件

一言でいえばTRON仕様チップの仕様に基づく各社の実装は初めての独自32ビットマイクロプロセッサにもかかわらずよくできたといえるであろう．GMICRO/200は1986年の開発発表の後，1988年夏にサンプル出荷，1989年秋に量産となった．しかし，出荷時期が大きく遅れたことが普及には障害となった．これは開発側から見れば種々の理由，開発バグ，資金，はじめて取り組むことなど多くのことより，今となっては仕方がないことかもしれない．なるようにしかならなかったといえるだろう（その後開発されたGMICRO/500を見れば日立の実装に対しての実力が大きいことは分かる）．しかし，当時TRON仕様チップを使おうとする人々からしてみればモノがこないことにはどうにもならないということになる．もたついているうちにより高い性能のマイクロプロセッサが求められるようになっていった．ただ，いくつかのRISCプロセッサに見られるように高い性能のプロセッサを予定どおりに出荷しても成功するとは限らず，大量に使われる具体的応用があるかどうかの方が本当は重要なのだろうが．開発されたマイクロプロセッサにはMMU（Memory Management Unit）も搭載されて組込み応用だけでは明らかに持て余してしまいワークステーションやパーソナルコンピュータなどに利用しないととても合わなかった.

私はTRON仕様チップの一番の用途はスーパーパーソナルコンピュータだと考えていた．これは，当時米国で勢いをつけてきていたワークステーションの能力を備え，パーソナルでも使えるようなローコストなマシンである．当時はパーソナルコンピュータというとまだ性能は低くゲームや自分でBASICを使ってプログラムを書く，ビジネスでは表計算とワードプロセッサとして利用するというのが中心だった．ちょうどそのころ米国では新しいワークステーションメーカがいくつも生まれている．Sun，Three Rivers PERQ，APOLLO DOMAINといったワークステーションを送りだしている．PERQ，DOMAINは独自OS，SunはUNIXをOSとしていた．また，ミニコンメーカとして老舗のDEC，DG，HPなどもUNIXのサポートをしていたという時代である.

実はUNIXがまだ開発途上であった1970年代に，AT&Tのベル研究所に行き当時開発中だったUNIXを知り，日本のメーカにもUNIXをOSとして採用したらどうかと薦めたことがあった．当時日本は大型コンピュータはバッチ処理から一部先端的なところでTSSを使っているという時代であり，ミニコンはようやくDisk

Operating Systemが動いているという状況である．残念ながら，どちらかというとパーソナルで使うのに適したUNIXには誰も興味を示さなかった．どうしてもIBMに追い付け追い越せという考えがコンピュータ部門にあったのだと思う．約10年後，TRON仕様チップを開発する時点においてもスーパーパーソナルコンピュータというコンセプトに産業界は興味を示してくれなかった．もう少し言えば，半導体部門は興味を示したのであるが，コンピュータ部門はマイクロプロセッサをおもちゃとしか見ていなかったのである．同一の社内にコンピュータ部門も持つ日本の半導体メーカは半導体部門で勝手にコンピュータを作ることはできなかった．このため，後に作られたTRON仕様チップによるコンピュータシステムは，あくまでもソフトウェア開発装置あるいはCPUの評価基板として作られた．

TRON仕様チップを使ったパーソナルコンピュータが早期に登場しなかった理由の1つはここにある．では半導体メーカでないメーカは作れなかったか．もちろん理論的には作れる．しかし，intelのチップに比べれば作りにくいという事情があった．というのはintelの戦略はCPUを作るだけではなく，コンピュータシステムとしてつくるための周辺チップを徹底してそろえていたからである．それに対し，前述したようにTRON仕様チップは各社がみなCPUを作ることを第一義とした．このため，TRON仕様チップの応用製品を作るのはなかなか敷居が高かったのである．

intelは1989年にGMICRO/200の性能を上回るiAPX486の生産を開始し，1990年1月にはMotorolaがMC68040を発表した．またRISC陣営も見かけ上のMIPS値の高いチップをリリースしてきた．日本連合のTRON仕様チップに危機感を覚えた米国陣営の反撃であった．TRON側もGMICRO/300で対抗したが，パーソナルコンピュータという応用を実現できなかったために，一般の市場に見られる製品に組み込まれることができなかった．

GMICRO/300を搭載したパーソナルコンピュータはBTRON仕様OSの開発メーカであるパーソナルメディアで1993年に作られた[17),18)]が，当時のintel iAPX486を搭載したWindows3.1と比較して，非常に軽快に稼働した．TRON仕様チップがパーソナルコンピュータを作るのに向いていなかったわけではなかったのである．

また，GMICRO/300の後継であるGMICRO/500はintelのPentiumと同じ年1993年にリリースされている．GMICRO/500はPentiumに対して性能面でも製造技術面でもまったく遜色がない（文献14）表-3参照）．

しかしこれもパーソナルコンピュータに使われることはなかった．チップのパワーから考えると，このことはチップを普及させるには最悪だ．

1つの仕様に基づく複数のインプリメンテーション

1つの仕様に基づいて，これだけ多くのメーカが独自にインプリメントを行うという例はそれまでまったくなかったことだ．このMLDD（Multi-Layered Design Diversity）Architectureと呼ぶ設計開発手法[19)]は新しい考え方であり，もしTRON仕様チップがうまく普及していたら，メーカにもユーザにも多大なメリットをもたらしたであろう．仕様書に基づき開発する場合，どうしても仕様書の不備や誤解からエラーが混入する．もし複数のインプリメンターが開発を行えば相互の動作テストからエラーの混入を検出したり，その要因が実装にあるのか仕様書（仕様自体か表現かも含め）にあるのか検出することもできる．結局トータルのエラーを早期に減少させることができ，しかもユーザは複数メーカからの供給を得ることができる．

具体的な例がある．サンプルのチップができ上がり評価をしている最中である．ユーザーズマニュアルの原稿もできあがり，チップも動いている．あるメーカがTRON仕様チップの開発をしている他社のチップと仕様確認を行っていたところ，仕様を勘違いして作っていることを発見した．CHK命令という主に配列の範囲をチェックする命令である．この命令はメモリ上に書かれた上限と下限をインデックスの値と比較する．これらの値は命令語中のsxで指定されるオペランドサイズで解釈，演算される．そしてその結果のフラグが立てられると共に，デスティネーションレジスタにインデックスとの下限値との差分を格納する．格納されるこのサイズをワード固定と勘違いしていた．つまりこうしていた．

CHK/c bounds.sx, index.sx, destreg.w

ドット以下はサイズを示すが，最初の2つのオペランドのサイズはsxだが，destregのサイズがwとなっている．おそらくレジスタであるからサイズ固定と考えてしまったのであろう．TRONの仕様ではこのサイズもオペランドサイズsxに従うようになっている．

CHK/c bounds.sx, src.sx, destreg.sx

この発見によりマニュアルとチップを変更することになる．一般にこの手の小さな誤りは量産版としてリリースされたあと発見される．ハードメーカがテストをしていてどうもおかしいとチップメーカに指摘をすることもよくある．供給を受けたハードメーカに後から送られる多数の正誤表（Errata）の中に示され，次のマスクで変更予定などと書かれているものだ．うっかり正誤表を見過ごしてしまうと，新しいバージョンのチップに変わった時にソフトウェアが誤動作するというようなことが起こる．

おわりに

TRON仕様チップは1つの仕様により作られたにも

かかわらずそれぞれを見ると面白く，個性もあった．1981年のISSCCで米国のメーカが個性的で興味深い32ビットマイクロプロセッサを出してきたのに匹敵する独創的なアーキテクチャで日本の技術を誇示するのに十分であった．たとえば沖電気のO32は電子交換機や通信制御装置に使うことを目的としてチップバスの監視機構を持ち，二重化システムや多数決システムが容易に構成できるようになっている．三菱のGMICRO/100，東芝のTX1は実記憶版として作られ，高速ファクシミリなどOA機器やFA機器など組込み用途に特化した．これらのCPUはASICのプロセッサコアとして利用できるようになっていて，実装密度の高いシステムを実現することができる．今ではVHDLやVerilog-HDLなどのハードウェア記述言語（HDL，Hardware Description Language）によりマイクロプロセッサコアを記述しておき，望む機能を付け加えることにより，システムLSIを半導体プロセスに依存せずに開発することが容易になってきている．また，高速化技術も多段パイプライン，分岐予測，大容量TLB，キャッシュ内蔵，スーパースカラなどそれぞれのチップが目的性能に応じて採用している．歴史的にみると日本のメーカが32ビットを手がけられるジャンプ台となり組込み用32ビットの誕生を促した．TRON仕様チップはパーソナルコンピュータという目立つところで活躍することはできなかった．しかし，GMICRO/200は宇宙線対策が施され，宇宙用の世界初の32ビットマイクロプロセッサとして現在技術試験衛星「きく7号」に搭載され地球の周りを回っているし，GMICRO/300，500はNTTの電子交換機として通信の基幹を担っている．またGMICRO/100は半導体製造装置の制御装置や衛星放送の制御装置として日本の産業の基幹を支えている．

振り返ると1980年代初頭の32ビットマイクロプロセッサの誕生は1971年のi4004誕生にも比較できるほど大きい．32ビット機種の登場と共にマイクロプロセッサは32ビット演算，32ビットアドレス空間，仮想記憶サポート，きめ細かい例外処理など大型コンピュータ並みの機能を持ち，ほとんどのコンピュータのCPUがマイクロプロセッサになるまで時間はあまりかからなかった．

最近TOXBUS☆5が国際標準としてISOに採択されたという知らせを受けた．ISO/IEC DIS 14576 (STbus) として提案していたもので，この分野で初めて日本からの提案で国際標準ができた．TOXBUSはマルチプロセッサ，フォールトトレラント機能をサポートするTRON仕様チップのプロセッサシステムバスとして仕様を開発していたものである．TRON仕様チッププロジェクトは我が国の半導体技術者をエンカレッジする役割を担った[20]こと，派手ではないが，インフラストラクチャの基幹部で活躍していること，その一部が国際規格に採択されたことなどの貢献もした．

本稿をおこすにあたり，日立製作所金原和夫氏，稲吉秀夫氏，富士通伊野弘行氏，故伊藤又夫氏，森昭助氏，三菱電機岡久雄氏，北岡隆氏，榎本龍彌氏，東芝故江川英晴氏，浪本敬二氏，沖電気工業上原一矩氏，市浦登氏，松下電器産業坂尾隆氏，出口雅士氏をはじめとする共にTRON仕様チップを創り上げてきた多くの人々に感謝したい．

☆5 TOXBUSは高性能なマルチプロセッサシステムを構築することを目的として，分散して置かれるキャッシュのコヒーレンシー制御をサポート，フォールトトレラントシステム機能強化するなど，高性能システムバスとして設計された．主な特長としては，(1) バスの使用効率の高い同期転送方式 (2) アドレス／データの多重転送による信号数の削減 (3) バス信号線，バスインタフェース制御部の障害検出 (4) 障害発生ユニットをシステムバスから切り離すアイソレーション機能 (5) キャッシュコヒーレンシー制御プロトコルなどである．

参考文献

1) International Solid-State Circuits Conference Digest of Technical Papers, IEEE (1981).
2) Sakamura, K. : The Objective of the TRON Project: TRON Project 1987, Springer-Verlag (1987).
3) Sakamura, K.: TRON VLSI CPU: Concepts and Architecture: TRON Project 1987, Springer-Verlag (1987).
4) Sakamura, K.: Architecture of the TRON VLSI CPU, IEEE MICRO, Vol.7, No.2 (Apr. 1987).
5) Tomisawa, O. et al.: Design Considerations of the GMICRO/100: TRON Project 1987, Springer-Verlag (1987).
6) Takagi, K. et al.: Outline of GMICRO/200 and Memory Management Mechanism: TRON Project 1987, Springer-Verlag (1987).
7) Itho, M.: Architecture Characteristics of GMICRO/300: TRON Project 1987, Springer-Verlag (1987).
8) Kiyohara, T. et al.: Pipeline Structure of Matsushita 32-bit Microprocessor: TRON Project 1987, Springer-Verlag (1987).
9) Namimoto, K. et al.: TX Series Based on TRONCHIP Architecture: TRON Project 1987, Springer-Verlag (1987).
10) Ito, N. et al.: Architectural Features of OKI 32-Bit Microprocessor: TRON Project 1988, Springer-Verlag (1988).
11) Uchiyama, K. et al.: The GMICRO/500 Superscalar Microprocessor with Brahcn Buffers: Special East Asia Issue, IEEE MICRO, Vol.13, No.5 (Oct. 1993).
12) Special Issue on The 32-bit Microprocessor in Japan, IEEE MICRO, Vol.8, No.2 (Apr. 1988).
13) 森　昭助: 小特集: TRONプロジェクトの現状と展望: CHIPサブプロジェクトの現状と展望, 情報処理, Vol.36, No.10 (Oct. 1994).
14) 坂村　健: 解説: 日本のマイクロプロセッサ技術: 1. 日本のマイクロプロセッサ技術, 情報処理, Vol.39, No.3 (Mar. 1997).
15) 新世代マイクロプロセッサ: RISC, CISC, TRON: 4. TRON仕様マイクロプロセッサ, 日経エレクトロニクス・ブックス (Sep. 1989).
16) 坂村　健監修: トロン仕様チップ標準ハンドブック, パーソナルメディア (Sep. 1991).
17) SIGBTRON基本ボード: TRONWARE, Vol.22, パーソナルメディア (Aug. 1993).
18) ピュアTRONマシンMCUBE: TRONWARE, Vol.32, パーソナルメディア (Apr. 1995).
19) Watanabe, A. et al.: The Multi-Layered Design Diversity Architecture: Application of the Design Diversity Approach to Multiple System Layers : Proceedings of the Ninth TRON Project Symposium, IEEE Computer Society Press (Dec. 1992).
20) 伊藤元昭: 21世紀を拓くための人材を育てた「TRON」プロジェクトが残したもの, 日経マイクロデバイス (Feb. 1997).
21) Patterson, D. A. and Sequin, C. H.: RISC I: A Reduced Instruction Set Computer, 25 Years of the International Symposia on Computer Architecture Selected Papers, pp.24-26, ACM (1998).
22) Agarwal, D. et al.: Generation and Debugging of Optimized Code for the TRON Architecture, TRON Project 1989, Sakamura, K. (ed.) , Springer-Verlag, pp.297-320 (1989).

(平成11年2月9日受付)