Vol. 21 No. 3 情報処理学会論文誌 May 1980

制御用 16 ビットマイクロコンピュータのアーキテクチャ†

前島英雄†† 松本秀和††
大沼邦彦††† 喜田祐三††††

本論文では，制御用 16 ビットマイクロコンピュータに関し，特に制御用として重要な高速演算性能および高信頼性を得る LSI アーキテクチャについて述べる．本マイクロコンピュータは，既存のミニコンピュータと同一の命令体系を維持しながらNチャネル MOS デバイスによって製造する2種のカスタム LSI を基本要素として構成される．LSI の論理方式は，コンピュータ・アーキテクチャと同様にプロセッサの処理性能に大きな影響を与えるものである．そこで，高速演算の実現に対し，プロセッサ全体の論理構造を命令フェッチ用LSI (IFCU) と命令実行用 LSI (RALU) の2つの論理機能に分割するアーキテクチャを採った．また，LSI 内部では加算回路のキャリ高速伝播方法を提案し，マイクロ・サイクル・レベルでの高速化も示す．これによりカスタム LSI が，パイプライン制御（命令実行中に次の命令フェッチを並行して行う）機能を備え，300 nsのマシン・サイクル時間で動作するための技術を示す．結果として，このLSIを用いたマイクロコンピュータは，レジスタ・メモリ間加算命令 2.3 μs，ギブソン・ミックス 6.8 μs の高速性能を達成する．

更に，高信頼性の実現に対しては，カスタム LSI へ RAS 機能をオン・チップ化するアーキテクチャを提案する．これによりマイクロコンピュータの部品としての LSI が単なるデータ処理に止らず，主記憶保護やシステム監視などのエラー検出・処理をも行うことで制御用としての特長を活かし得ることを示す．

1. まえがき

最近の計算機制御では分散制御システムが広く受け入れられつつあり[1)~5)]，その特徴はホストに高性能メガミニコンピュータ，端末に多数の超小型マイクロコンピュータを配置することによって大規模かつ拡張性に富むシステムを容易に構成できる点にある．このような大規模システムにおいて，ホスト計算機には高いスループットが要求され，そのため汎用計算機に匹敵する性能を備えていなければならなくなっている．

これに対し，端末計算機はサブシステムごとに分散制御するため，既存のミニコンピュータ・アーキテクチャを維持しながら超小型化したマイクロコンピュータ（カスタム LSI）に置き換りつつある．この動きは汎用マイクロコンピュータ（標準 LSI）が8ビット系から 16 ビット系[6)~8)]へと進行し，性能的にミニコンピュータに接近しつつあること，半導体の高集積化技術が急速に発展していることといった周囲状況から着実に進展している．

† 16-bit Control Microcomputer Architecture by HIDEO MAEJIMA, HIDEKAZU MATSUMOTO(The 8 th Dept., Hitachi Research Laboratory, Hitachi, Ltd.), KUNIHIKO OHNUMA (Computer Control Hardware Dept., Omika Works, Hitachi, Ltd.), and YUZO KITA (The 5 th Dept., Hitachi Central Research Laboratory, Hitachi, Ltd.).

†† (株)日立製作所日立研究所第8部
††† (株)日立製作所大みか工場計算制御設計部
†††† (株)日立製作所中央研究所第5部

では，マイクロコンピュータにおける標準 LSI とカスタム LSI との差異はどこにあるのであろうか．前者が不特定多数のアプリケーションを対象として汎用的なアーキテクチャをとっているのに対し，後者が商用ミニコンピュータのアーキテクチャを踏襲している点に根本的な思想の違いがあろう．したがって，カスタムの場合，多くのアプリケーションをこなすことによって長い間に蓄積した豊富なソフトウェア財産や信頼性を含めた数々のノウハウを継承している．このため，システム固有の強みを集積したカスタム LSI によって構成した超小型マイクロコンピュータがミニコンピュータに置き換って分散制御システムにおけるキー・コンポーネントになり得るのである．

本論文では，このカスタム LSI のアーキテクチャに関し，特に制御用として重要な高速演算性能および高信頼性について論じている．高速演算性能に関してはパイプライン（命令実行中に次の命令フェッチを行う）制御方式と高速論理回路方式を，高信頼性に関してはプログラムの暴走を防止するシステム監視機能，プログラム破壊を防止する主記憶保護機能のオン・チップ化を提案している．

2. 制御用マイクロコンピュータの必須条件

2.1 超小型

制御用マイクロコンピュータの具体的応用を**図1**に示す．この図は分散制御システムにおける端末計算機

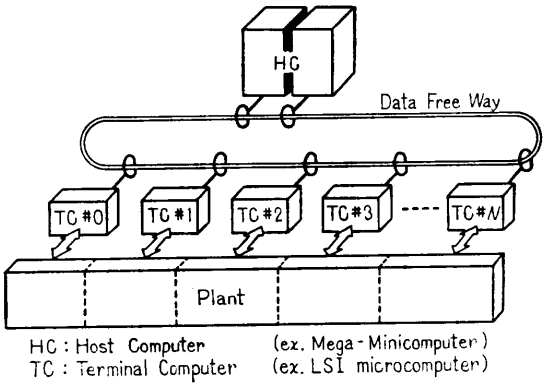


図 1 分散制御システム
Fig. 1 A distributed control system.

(TC : Terminal Computer) として使われた例を示したものである．大規模システムを多数のサブシステムに分散して制御するため，各々のコンピュータは超小型であることがシステムの最適化(性能・価格比向上)に不可欠である．

ところで，マイクロコンピュータに用いられている半導体プロセスは主に次の2つである．第1はビットスライス型 LSI に用いられている LS-TTL (Low power Schottky-Transistor Transistor Logic)，第2は1チップマイクロコンピュータに用いられている MOS (Metal Oxide Semiconductor) である．現時点では，集積度で MOS，デバイスのスピードで LS-TTL が優位にあるが，MOS における微細加工技術の進歩には著しいものがある．その波及効果は大きく，高密度かつ高速な LSI の実現に対し，将来にわたって最適なプロセスであると判断できる．

本論文にて述べるカスタム LSI も超小型マイクロコンピュータの実現に向けて N チャネル MOS 技術を用いて開発されたものである．

2.2 高 速 性

制御用コンピュータは図1にみられるように制御対象を直接制御するため，本質的に実時間処理を行うものであり，高速演算性能を必要としている．特に圧延機制御のような機械と直結する部分での DDC (Direct Digital Control) では機械の応答速度に見合った高性能が要求される．

前述したように現在の段階では，MOS デバイスのスピードが LS-TTL のそれよりも劣っており，これを克服する鍵は LSI アーキテクチャと内部論理回路方式にある．前者は命令の並列処理性，後者は MOS デバイスに適した高速論理の追求が課題である．

2.3 高 信 頼 性

制御用コンピュータの分野では RAS (Reliability Availability Serviceability) は重要な機能である．小さな誤りがシステム・ダウンに波及する致命的な障害に至ることが多々起り得るからである．この防止には先ず種々の段階でコンピュータの誤動作を検出することから始まる．一般的には主記憶に係るデータとアドレスの誤り，すなわちパリティ・エラー，アドレス・エラーの検出機能，プログラム領域（例えば，オペレーティング・システムのプロシージャ）の破壊を防止する主記憶保護機能が最低限要求される．また，ハードウェア，ソフトウェアの誤動作を一括して検知するシステム監視機能もシステム暴走の危険を回避する有力な手段となる．このようにマイクロコンピュータがデータ処理に止らず，エラー検出・処理をも行うことによって制御用コンピュータとしての特長を活かすことができるだけではなく，全体の部品数を低減する効果もある．

2.4 互 換 性

マイクロコンピュータが既存システムで開発したソフトウェア財産（コンパイラ，サブルーチン・ライブラリ等の各種ユティリティ）やハードウェア財産（主記憶，入出力装置等）を共用できる互換性はシステム構成，運用の面で利点が多い．例えば，分散制御システムにおいて，同一マクロ命令にてホスト計算機と端末計算機相互間の入出力装置，ファイルの共用（データ管理）やホスト計算機から端末計算機のプログラム起動／停止（タスク管理）といった「使い易さ」に魅力のあるシステムが得られる．また，ミニコンピュータの持つものと同等の命令体系，多様かつ高度なアドレッシング・モードは，半導体技術の進歩により数ミリ角のチップ上に集積可能となっている．

3. プロセッサの仕様

表1に制御用 16 ビットマイクロコンピュータの仕様概要を示す．この仕様は既存のミニコンピュータとソフトウェア，入出力装置共に完全な互換性を持っている．命令は S1，S2 の1語命令と L1，L2 の2語命令で構成されており，それぞれの形式を**図2**に示した．

本マイクロコンピュータの特長は，高速の固定小数点乗除算，浮動小数点演算を標準装備した強力な命令セット，割込みのハードウェア・ベクタリングによる高応答性，システムの信頼性を考慮した RAS 機能の

表 1 仕様概要
Table 1 Summary of specification.

Data length	16, 32 bits
Instruction length	16, 32 bits
Instruction formats	S1, S2, L1, L2
Instruction set	59 with floating-point arithmetics
Addressable memory	64 K words (1 word=16 bits)
Memory bus	16-bit data, 1-bit parity, and 16-bit address; DMA capability
Interrupt	Vectoring and Stacking, 3 levels
Register	Two 16-bit operation registers, six 16-bit base registers (3 of them are used as index registers), five internal working registers, and 32-bit program status word

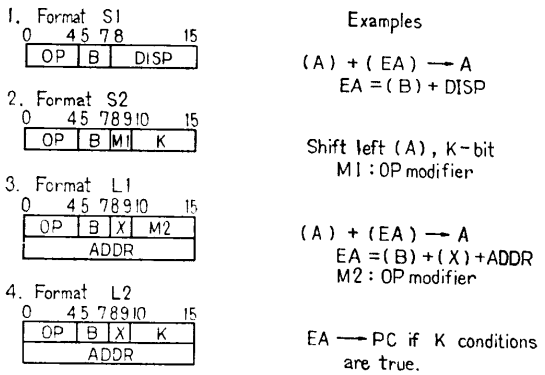


図 2 命令フォーマット
Fig. 2 Instruction formats.

充実にあり，これらは制御用コンピュータとして欠くことのできない重要なものである．

また，マイクロプログラム制御方式の採用により，シーケンス制御，DDC といった特殊なアプリケーションに対し，シーケンス命令，平方根・不感帯・飽和等の特殊命令を容易に追加することができ，適応性の高いマイクロコンピュータでもある．

4. 高速処理型のアーキテクチャ

Nチャネル MOS デバイスによりマイクロコンピュータを実現する場合，最大の課題はデバイス特性を見極め，いかに高速性を引き出すかにあるといえよう．このためには2段階の技術が必要であり，第1はシステム的に並列処理性を高める LSI 構造，第2は LSI 内部における信号遅延を最小とする回路構造である．

以下，それぞれについて説明する．

4.1 並列処理

命令処理の流れに注目すると，1つの命令は命令語の読出しとオペランドのアドレス計算までを行う「フェッチ」とこれに従属する「実行」の2つの直列処理に分解することができる．これらを独立したハードウェアで構成すればそれぞれの機能を並列に処理できるはずである．このような方式は「パイプライン制御」とか「先取り制御」とか呼ばれており，ミニコンピュータ以上の機種では常識的に用いられている技術である．しかし，LSI では集積度，パッケージのピン数に制限があり，その構成は難しい．

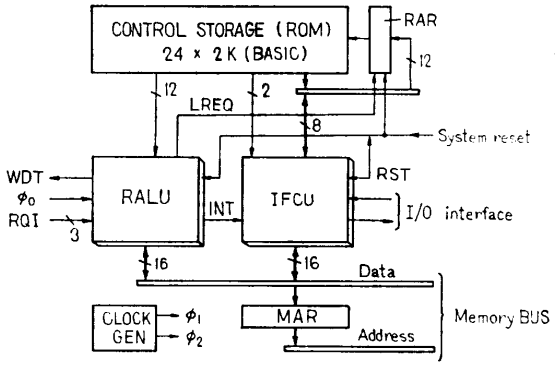


図 3 CPU ブロック図
Fig. 3 Block diagram of the CPU.

本論文では，CPU(Central Processing Unit) を命令フェッチを行う LSI「IFCU」と命令実行を行う LSI「RALU」の2種カスタム LSI に分割した構成を提案する．その構成を**図3**に示したが，2種 LSI の他にメモリ・アドレス・レジスタ (MAR)，制御メモリ (ROM)，制御メモリ・アドレス・レジスタ(RAR) を基本構成要素としている．これをビットスライス型のバイポーラ LSI で構成したものと実装面で比較すると，プロセッサは約1/3に小形化される．この効果は，低消費電力化，低コスト化，さらに，高信頼度化へ波及する．以下に本 LSI の構造上の特徴を示す．

(1) IFCU の役割 (命令フェッチ)

命令語を主記憶から読出し，これを解読してオペランドのアドレスを計算するところまでを行う．さらに，命令実行を制御する制御メモリ (ROM) 中のマイクロプログラム先頭アドレスを作成し，これを RAR へ転送することにより RALU へ制御を渡す．以後，次の命令フェッチを開始できる．

(2) RALU の役割 (命令実行)

前記 RAR に置数されたアドレスに対応して ROM から読出されるマイクロ命令を順次実行することにより IFCU によって読出された命令の実行を行う．実行終了時に次の命令実行のためのマイクロプログラム先頭アドレスを IFCU に要求する．また，マイクロ

プログラムでの条件分岐制御も本 LSI が管理する.

(3) フェッチ制御方式

フェッチ動作は命令形式 (1語命令／2語命令), 命令ごとのアドレッシングの違いなど複雑な制御を必要としているため, マイクロ命令で制御するとその語長は長くなってしまう. そこでマイクロ命令制御はフェッチの起動のみとし, 以後は LSI 内部制御として同時に高速化も図った. この結果, IFCU 制御のマイクロ命令語長は 10 ビットとなり, S形式, L形式の命令フェッチ・サイクル数はそれぞれ 2, 3 サイクルとなった.

(4) パイプラインの同期化

前記したように命令フェッチが IFCU 内部で閉じた制御を行っているのに対し, 命令実行はマイクロ命令制御となっている. したがって, 命令フェッチと実行の間には一義的な関係は存在せず, パイプライン動作に「ずれ」が生ずることもある. この「ずれ」は次のような方式で解決した. すなわち, IFCU がマイクロプログラム先頭アドレスを提供し, RALU はこれを受け取るという基本的立場から, IFCU におけるフェッチ動作が終了していない場合には特定のアドレスを提供するようにしたものである. この特定アドレスに対応するマイクロ命令は無実行かつ命令実行終了を指示しており, RALU はそのマイクロ命令の実行を繰り返すことによって IFCU のフェッチ終了を待つ. このようにして両チップ間のパイプライン同期をとった.

(5) 割込みの受付け

割込みは, 先ず RALU においてレベル間の優先判定を行い, 同時に割込みの有無を IFCU に知らせる. IFCU ではフェッチ起動を指示するマイクロ命令が発行された時にこの判定を行う. もし, 割込みが有る場合にはフェッチ動作を抑制し, 前命令実行終了時に割込み処理用マイクロプログラム先頭アドレスを提供することで対処している.

以上のように, 2チップ間の制御の受渡しをマイクロプログラム・アドレスで達成したことにより2チップの独立性 (並列性), 高速性を高めると同時に, LSI 実現の際, 常に問題となるパッケージのピン数を 42 ピンに抑えることができた.

4.2 高速回路

プロセッサの高速処理を実現する今1つの方法は処理の最小単位であるマイクロ・サイクル時間を短縮すること, すなわち1マイクロ・サイクル中に行う論理

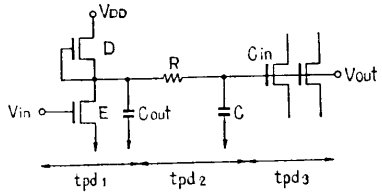


図 4 E/D MOS ゲート伝播遅延モデル

Fig. 4 Propagation delay model of the E/D MOS gate.

動作時間を最小とすることである. 1マイクロ・サイクルでは乗除算における繰り返し演算 (加減算とシフト) が最もクリティカルな論理動作である. このような論理動作において, 前節に示した IFCU におけるオペランドのアドレス計算および RALU における命令実行過程での各種データ演算を行う回路に最も時間的比重が大きい. いずれの回路においても桁上げ (キャリ) を発生し, 上位桁にこれを伝搬する加算の速度が問題である. 本節では加算回路を例として, NMOS に適した高速回路方式を述べることにする.

加算回路は, 従来, リップル・キャリ方式, キャリ・ルックアヘッド方式が知られているが, いずれの方式をとっても1マイクロ・サイクルでの論理段数全体の 40～50% を占めている. したがって, この部分に高速回路を取り入れることが最も効果的な高速化手段となる.

16 ビット加算回路の信号遅延時間の算出は, 図4のモデルに従い, 次式を用いて行った.

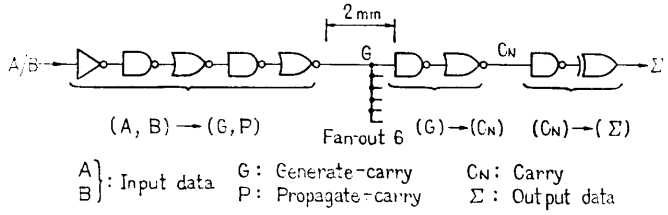
$$t_{pd}=\sum_{i=1}^{N}(t_{pd1i}+t_{pd2i}+t_{pd3i}). \qquad (1)$$

ここで, N は加算回路の論理段数, t_{pd1} は各ゲートの出力容量 C_{out} による遅延時間, t_{pd2} は各ゲート間の配線容量 C および配線抵抗 R による遅延時間, t_{pd3} は次段ゲートの入力容量 C_{in} の総和, すなわちファンアウトによる遅延時間を表す.

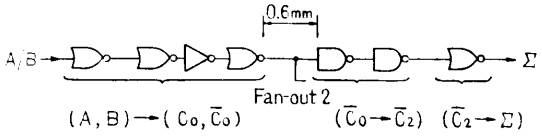
MOS 論理回路の信号遅延時間は論理段数 N と3種の遅延要素から決定されるが, それぞれの要素は互いに独立でないため, 各要素間の干渉をきめ細かく分析しなければならない. 以下, その過程を述べることにする.

(1) 論理段数 N の短縮

論理段数だけに注目すると従来方式ではキャリ・ルックアヘッド方式が最も少なく, 16 ビット・データ加算におけるクリティカル・パスは図5に示したように 10 段が方式上の限界である. これをさらに減少さ



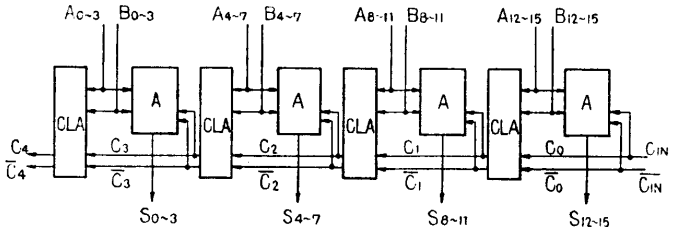
(a) Carry-Lookahead method.



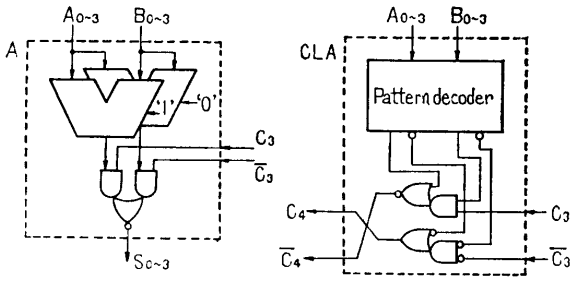
(b) New method.

図 5 2種加算回路のクリティカル・パス

Fig. 5 Critical paths of two adder circuits.



(a) Configuration of 16-bit adder.



(b) 4-bit dual adder. (c) Carry-lookahead circuit.

図 6 高速加算回路

Fig. 6 High-speed adder circuit.

せる方策は処理の並列性を増すことに尽きる．その具体的手段として次の3項目を提案する．

（i） 加算部(A)と桁上げ発生部 (CLA) を分離する（並列動作）．

（ii） 下位からの桁上げ発生の起る前に先行して2組の加算結果を準備する（加算部の2重化）．

（iii） 正負両論理1対の桁上げ伝播論理を設ける（桁上げ伝播の高速化）．

以上の3点を基本に構成した 16 ビット高速加算回路を**図 6** に示す．本方式のクリティカル・パスは図5に示したように7段となって実用的な加算回路としては最少である．また，この回路方式は論理段数が少ないばかりでなく，配線領域をも少なくするのでLSI 化に向いた方式である．この点に関しては(3)項にて詳しく説明する．

（2） t_{pd1} の短縮

各ゲートの出力容量 C_{out} は MOS トランジスタの W と L に依存し，t_{pd1} はこの C_{out} および W/L 値によって決まる．W/L 値を大きくとるとトランジスタの駆動能力が増加し，スイッチングが速くなるが，C_{out} および入力容量 C_{in} は増加するため，適当な W/L 値を選ばなければならない．本回路では回路全体のバランスを考え，これを 0.5～2.0 の範囲とした．

（3） t_{pd2}, t_{pd3} の短縮

各ゲート間の配線長およびファンアウトは回路方式によって大きく異なる．図5に示したように，キャリ・ルックアヘッド方式では最大配線長が約 2 mm，ファンアウト数6となる大きな負荷が存在した．このため，配線による信号遅延，回路サイズへの影響が大きくなり，高速な MOS 回路には適さない．これに対し，本論文の新しい方式では，4つの CLA 回路によって分散化した結果，最大配線長を 0.6 mm 程度に抑え，かつファンアウト2とすることができた．したがって，配線容量C，次段ゲート入力容量 C_{in} の総和それぞれをともに 1/3 に減少し，t_{pd2}, t_{pd3} の短縮を達成した．

以上の結果，計算機による回路解析では 16 ビットのデータ加算遅延時間を約 76 ns と推定し，これは実測データによっても確認された．

5. 高信頼型のアーキテクチャ

ミニコンピュータに匹敵する性能を備え，制御用として用いられるマイクロコンピュータにとって RAS は重要な機能である．これが CPU の LSI 内に集積してゆくことは丁度 DMA や HDLC 機能といった周辺 LSI が CPU に包含され，1チップ化してゆく方向と同様に進んでゆくものと思われる．本論文では「制御用」を強く意識し，RAS を高速処理性と同等の重要さとしてとらえている．次に，RAS として不可欠な項目を述べる．

プロセッサの誤動作を分析すると，その大半の原因は主記憶に直接あるいは間接的に関係している．した

がって，この誤りを検出し，処理することが RAS の大きな役割である．本論文にてマイクロコンピュータの RAS として最低限必要であると考えた誤り検出を挙げ，その処理方式を示す．

5.1 エラー検出

（1） パリティ・エラー（データ誤り）

主記憶のハード障害が原因である．主記憶制御回路にて検出し，IFCU に報告され処理される（次節参照）．

（2） アドレス・エラー（アドレス誤り）

ハード，ソフト障害が原因であり，（1）と同様に検出，処理される．

（3） インバリッド・エラー（無定義命令）

本エラーは IFCU において命令解読後に検出される．その結果はマイクロプログラム先頭アドレスに反映し，無定義命令があたかも 1 つの命令であるかのごとくマイクロプログラム処理する．

（4） プロテクト・エラー（主記憶保護）

主記憶の書込み禁止領域（例えばオペレーティング・システムのプロシージャ部）への書込み指示誤り．IFCU 内のアドレス情報（IMAR）と保護すべき領域を定めたプロテクト・レジスタ（PRT）の内容を比較・検出し，（1）と同様の手順で処理される．

（5） システム暴走

ソフト，ハードにかかわらず原因不明のエラーによるシステムの暴走を検出するにはプロセッサの動作を何らかの手段で常に監視しなければならない．そこで，本マイクロコンピュータではウォッチドッグ・タイマ（WDT）と呼ぶハード・タイマを RALU 内に他の論理と切り放して配置した．本タイマは一定時間内にソフトウェアによるイニシャライズが成されることを前提に使われ，もしハードあるいはソフトの誤りによってイニシャライズされない時にシステム異常とみなすものである．この結果は直ちに RALU 外部に報告され，緊急停止等の処理が成される．

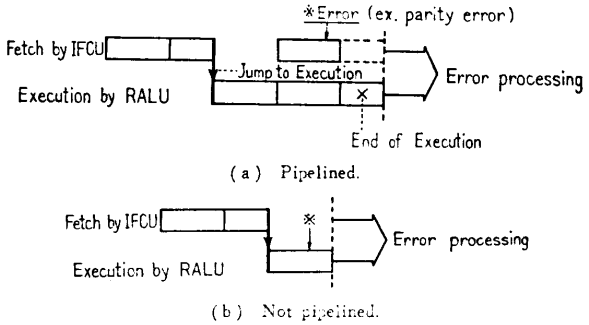


図 7 パイプライン制御の処理装置におけるエラー処理
Fig. 7 Error processing in a pipelined processor.

5.2 エラー処理方式

パイプライン制御のプロセッサでは前節におけるパリティ，アドレス・エラーの発生時期が重要なポイントである．すなわち，パイプライン状態か否かでエラー検出後の処理方法が分れることになる．これは命令フェッチ中（パイプライン中）に発生するエラーによって実行中の前命令への外乱をなくすこととエラー発生時の LSI 内部状態保存が目的である．

エラー検出後の 2 種の処理方法を図 7 に示し，以下

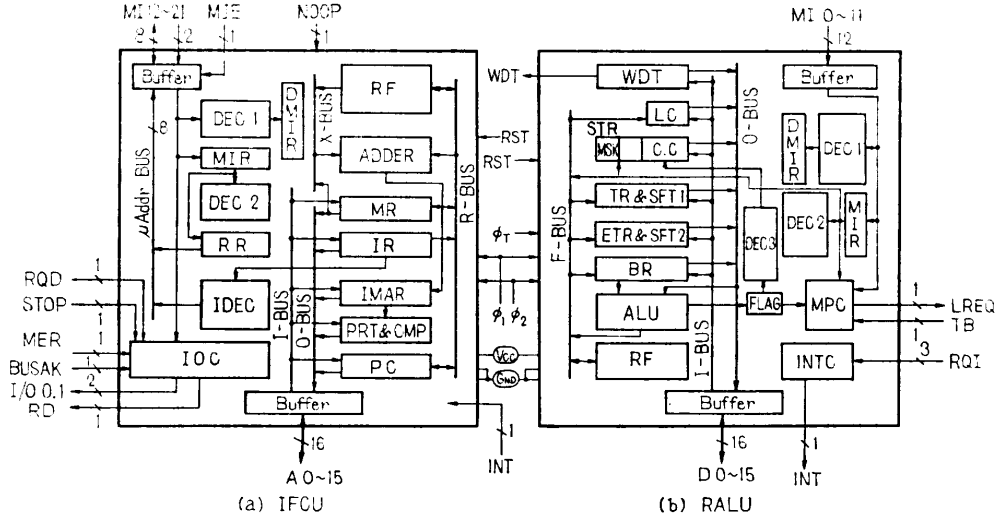


図 8 LSI ブロック図
Fig. 8 Block diagram of the LSI chips.

表 2 命令実行時間
Table 2 Execution times
(For memory cycle of 525 ns).

	Addition (μs)	Multiplication (μs)	Division (μs)
Fixed-points	2.3	11.4	15.3
Floating-points	22.8	44.4	66.8
Scientific Gibson Mix		6.8 μs	
System Mix		3.2 μs	

にその内容を述べる.

(a) パイプライン中のエラー

命令処理は, IFCU での命令フェッチ (Fetch) 後 RALU へ制御が渡され, 命令実行 (Execution) が開始される. 命令実行中に並行する次の命令フェッチでエラーが発生した場合, 前命令の実行終了を待って図3に示した IFCU におけるリセット端子 (RST) からシステム・リセット信号を送出する. これにより, エラー処理用マイクロプログラムを起動し, 処理する.

(b) 非パイプライン中のエラー

命令の実行終了を待つこと無しに, 直ちにシステム・リセットを行い, (a)と同様に処理する.

6. LSI 仕 様

パイプライン制御と RAS 機能を包含する2種のカスタム LSI の内部回路構成を**図8**に示す. これらの LSI はNチャネル MOS プロセスによって製造され, IFCU, RALU それぞれのチップ・サイズは 6 mm×5.9 mm, 5.86 mm×6 mm であり, 単一+5 V 電源で消費電力約 0.7 W, TTL コンパチブルの入出力レベルをもつ. また, 0℃ から 70℃ の全領域で 3.3 MHz のクロック周波数で動作する.

この結果, 命令処理時間ではアクセス時間約 525 ns の主記憶を用いた場合, レジスタ・メモリ間加算命令が 2.3 μs, その他の処理時間は**表2**に示したごとく極めて高速なものが得られた. これを総合的に評価すれば, 科学計算における平均命令実行時間 (Scientific Gibson Mix) が 6.8 μs, オペレーティング・システムにおける平均命令実行時間 (System Mix) が 3.2 μs となる. これによって本 LSI を基本構成要素としたプロセッサの性能の高さを示すことができる. これらの数値は素子スピードの数倍速いバイポーラ素子で構成したプロセッサの性能を約 20% 向上している. 現段階の MOS プロセスでも LSI アーキテクチャおよび回路技術の追求によってミニコンピュータの性能・機能をマイクロコンピュータに集約することが十分可能であることを示している.

7. む す び

制御用 16 ビットマイクロコンピュータのアーキテクチャについて報告した. LSI のアーキテクチャはコンピュータ・アーキテクチャ同様, プロセッサの処理性能に与える影響が大きい. 高速性能を得るため, 命令処理の流れに注目し, 命令フェッチ用 LSI (IFCU) と命令実行用 LSI (RALU) の2種 LSI に分割してパイプライン制御し, 処理の並列性を増すことで実現した. また, LSI 内部では加算回路において, キャリの高速伝播方式を示し, マイクロ・レベルでの高速化を図った. この結果, カスタム LSI は 300 ns のマシン・サイクルで動作し, レジスタ・メモリ間加算 2.3 μs, ギブソン・ミックス 6.8 μs の高速性能を達成した. さらに, 制御用マイクロコンピュータとして重要な高信頼度を得る諸機能 (RAS) のオン・チップ化を示した.

2種カスタム LSI を含むマイクロコンピュータは, 既存のミニコンピュータと同一の命令セットを持ちながら1ボードで構成でき, 高級言語, アセンブラ, 各種ユティリティのソフトウェア・サポート, 入出力装置のレパートリもすでに準備されている. さらに, ファームウェア技術により標準のコンピュータとして動作中に問題向きプロセッサへの切り換えが可能であり, 多彩なアプリケーションに適応するものである.

なお, 本論文で述べた LSI は, 制御用 16 ビットマイクロコンピュータ HIDIC 08-L のカスタム LSI として開発されたものである. 本稿の研究に当たり, 有益なご助言をいただいた日立製作所・大みか工場桑原洋氏, 保田勲氏, 同武蔵工場城聖一氏に謝意を表します.

参 考 文 献

1) 平子, ほか: 制御用分散処理システム, pp. 346～349: 情報処理, Vol. 20, No. 4 (1979).
2) 平井, ほか: 制御用計算機における分散処理ネットワーク技術の動向, 日立評論, Vol. 60, No. 7, pp. 1～6 (1978).
3) 桑原, ほか: 制御用コンピュータのネットワーク・システム, 電気学会雑誌, Vol. 98, No. 3, pp. 199～203 (1978).
4) Jensen E. D.: The Honeywell Experimental Distributed Processor-An overview, Computer, Vol. 11, No. 1, pp. 28～38 (1978).
5) Sebern, M. J.: A minicomputer compatible micro-computer system, The DEC LSI-11,

proceeding of IEEE, Vol. 65, No. 6 (1976).
6) Morse, S. P. et al.: The Intel 8086 Micro processor: A 16-bit Evolution of the 8080, Computer, Vol. 11, No. 6, pp. 18～27 (1978).
7) Shima, M.: Two versions of 16-bit chip span microprocessor, minicomputer needs, Electronics, Vol. 51, No. 26 (1978).
8) Stritter, E. et al.: A Microprocessor Architecture for a Changing World; The Motorola 68000, Computer, Vol. 12, No. 2 (1979).

(昭和54年10月9日受付)
(昭和55年2月8日採録)