

解説

システムの高信頼化技術†



井原 廣 一††

1. ま え が き

コンピュータシステムは、スーパーコンピュータからマイクロコンピュータまであらゆる分野に利用されてきていることは、あらためて述べるまでもない。これを可能にした最大の技術的発展が、LSI 技術であることは、これまた通説である。LSI 技術の未発達段階においては、ハードウェアの複雑さを避ける技術の開発に多くの努力が費やされた。次いで、LSI 技術により多量の標準的な素子が供給されると、汎用的に用いられる素子の標準化とこの LSI を組み合わせるシステムを構築することが研究開発の目標となってきた。しかし、この段階ではまだ標準素子間の結合は論理素子レベルのものであった。ところが、最近急速な変化が現われている。16ビットマイクロコンピュータおよび64kB 半導体メモリに先駆される超 LSI の出現、半導体 CAD 技術の発展と半導体製造技術の普及である。超 LSI の出現は、ハードウェアのコストと信頼性の呪縛からシステムを解放し、CAD および製造技術の普及は、システム構築上の制限を著しく軽減しつつある。コンピュータ技術者の多くは、ハードウェアよりもシステムやソフトウェアにより多くの関心を払わねばならなくなっている。ハードウェアにより制限された性能や容量の中で、システムを構成しプログラムを効率的に動かすのではなく、要求されるシステムの性能を発揮するハードウェアを構成することが可能になりつつある。パイプライン方式やアレイプロセッサ方式、ミニコンやマイクロコンによる分散コンピュータシステム、インテリジェントターミナルなどは主として高性能仕様を目的としたものである。

本稿の主題である、コンピュータの高信頼化面においても同様であり、素子や回路の高信頼化から、システムの観点からの高信頼化技術が主流になりつつあ

る。1つの方向は、汎用コンピュータにおいて、実時間利用が進むにつれて RAS 技術が重要となり、誤りの検知、訂正、再試行、誤り記録などの機能を次々と付加しており、現在では、システムの高信頼化に費やされるハードウェアコストは、20~30% に達しているといわれている^{1)~4)}。他の方法は、独立した複数の自律的なシステムの結合によってシステムを構築し超高信頼性、あるいはフォールトトレランス性を持たせるのである。

本稿においては、高信頼化の個々の技術を眺めることは他にゆずり^{5)~10)}、まず高信頼化コンピュータシステムの発展段階を述べ、最近、開発されている超高信頼化システムと呼ばれている^{2, 3}の例をシステムレベルで眺め、高信頼化技術の動向をさぐる。ここでいう超高信頼コンピュータシステム (Fault Tolerant Computing System) は、故障率が 10^{-10} /時間以下のものを、すなわち、MTBF が約 100 万年を目標としているものをいうことにする。

2. フォールトトレラントへの要求

ハードウェアのコストがシステムの相当な部分を占めていた初期のコンピュータシステムにおいては、コストパフォーマンスの向上がシステム構築時の目標であった。したがって、すべての処理を集中して大型コンピュータで行うトータルシステムが推奨され、コンピュータの障害が致命的な障害を発生する部分への導入は避けると同時に、障害が発生しないよう、あるいは、障害を早く発見するような素子レベル、回路レベルの高信頼化技術を追求めた。

その後、ハードウェアコストの低下とハードウェア信頼性の向上が進行するに従い、実時間システムが導入されてくる。情報処理が主体のシステムにおいては、データベースの喪失時における損失コストは膨大なものとなるので、一般に待機予備を持つ2重系システムの形態をとる。座席予約システム、銀行システムがこの例である。一方、生産システムにおけるコンピ

† A Survey of System Technology on Fault Tolerant and Reliable Computing by Hirokazu IHARA (Systems Development Laboratory of Hitachi Ltd.).

†† (株)日立製作所システム開発研究所

ユーシステムは主に階層システムの形態をとり、機能と実時間性を考えて、上位には計画統制などの経営的機能を、下位には実時間性の強い制御的機能を分散する。このシステム構成は、障害が発生しても部分的なバックアップシステムをとることによってシステム全体の停止を避けることが可能である。鉄鋼、電力、自動車などの生産制御システムがこの例である。

コンピュータシステムの導入がより一般的になると、保守、拡張、変更、取り換えなどのコストも考えねばならなくなり、また、公共システムへの導入により、システム信頼性の期待は益々高いものとなる。また、これまでの人間機能の置換から人間能力では不可能な環境における機能を行わせる要求も発生してくる。すなわち、システムのライフサイクルコストと、耐故障性からシステムが評価されてくる。人命に関するシステム、公共システム、宇宙システム、軍用システムなどがこの例である。図-1 に性能と障害の関係性をまとめてみた。

図-1 における第3象限に入るコンピュータシステムは、分散システムと多重系システム技術により実現されるもので、この内容を D. B. Rennel は次のように分類している¹¹⁾。

(1) 障害が人命や高価な装置に損害を与える恐れのある実時間制御システムとして、工場、病院、交通システムなどに適用されており、演算結果が正しいことと共に MTTR の小さい(秒のオーダー)ことが必要である。

(2) 保守不可能な無人宇宙機器に搭載されるコンピュータシステムで5年以上の長寿命を必要とする。システム自身が必要なスペアを持ち、地上からの診断や自動障害復旧を行う必要がある。

(3) 大規模なリソースシェアシステムで、部分的なユーザ向きの能力の喪失は許されるが、共通データ

の破壊は絶対許されない。電話交換や、商用タイムシェアリングシステムがこの例である。

(4) 信号処理を行う 500~1,000 FLOPS の性能が要求される超高性能コンピュータシステムの複雑さとクロックマージンの減少は、システムにトランジェントエラー、パーマネントエラーを増大させるが、繰り返し演算を利用した高信頼システムが考えられる。

(5) 保守コストが高価で、フォールトトレラント技術のコストのほうが安価な場合の例としては、軍用システムがある。交換モジュールが到着するまで機能を果し、モジュール到着時に交換することにより、保守員や測定機器を減少することが可能となる。

著者は、以上の他に(6)鉄鋼や加工組立工場における長大な連続ラインの制御システムの場合を追加する。一部の障害は結局全ラインの停止を意味し、その間の損害が高信頼化コストを越えるからである。

(1), (3), (6)については、拡張性も重要な要素である。すなわち、実時間稼動中にシステムの規模の拡大、機能の拡大、変更が常に行われるのであり、これは人為的障害の1つと考えておく必要がある。

3. 高信頼性コンピュータシステム開発の経過

コンピュータの歴史を振り返ってみると、その出現時より信頼性についての問題点が指摘されている。それが、トランジスタ、IC、LSI と素子の変遷と共に飛躍的に素子の不良率は減少してきた。一方、これに伴って、より高信頼化を要求される分野への導入が促進されている。

高信頼化システムは、軍用や宇宙開発用、すなわち、比較的成本制限の少ない分野において、1950年代から始まっている。SAPO、SAGE がそれである。SAGE¹²⁾は、防空システムとして真空管を用いた AN/FSQ-7 というコンピュータでデュプレックス系を構成している。1950年代後半には、ベル研究所の電子交換機システム No.1 ESS¹³⁾がある。このシステムは、ユニットごとの2重待機系構成をとっており、同期処理のチェックポイントでつき合せを行う。演算素子は個別部品である。IBM が担当した RTCC (Real Time Computer Complex)¹⁴⁾システムは、有人飛行計画の1950年代の Vanguard から Appollo を経て最近の Space Shuttle までの地上におけるコマンドアンドコントロールシステムとして汎用の計算機による待機2重系である。要求される信頼性は 0.9995 であり、こ

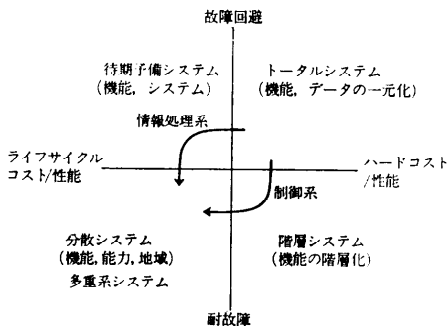


図-1 コンピュータシステムの評価と変化

れを満足するために、打上げや帰還時には MOC (Mission Operational Computer) を主コンピュータ、DSC (Dynamic Stand by Computer) を待機系とし、瞬時切換を可能にしている。軌道飛行に入ると、MOC がチェックポイントごとに磁気テープにシステムのデータを蓄積し、1重系として動作し、異常時には MOC の再起動、あるいは DSC へ移行する。

1960 年代前半においての宇宙分野では、無人宇宙観測衛星 OAO (Orbiting Astronomical Observatory)^{15),16)}計画に採用された IBM による PPDS (Primary Processor and Data Storage) がある。1年間の完璧な稼働を実現するために、回路レベルに4重の冗長性をもたせ、電磁遅延線を3重系として多数決方式とし、さらに、データメモリは2重、命令メモリは4重のメモリを用いている。アポロ計画、スカイラブ計画のロケット Saturn 制御に用いられたコンピュータ¹⁵⁾は MTBF 25,000 時間の目標で ULD (Unit Logical Device)、マイクロミニチュア回路とフェライトコアを用いた汎用コンピュータであるが、7つのモジュールより成っており、それぞれ3重系の多数決方式を採用、メモリは2重系とし、パリティチェックにより異常を発見すると、正常系からのデータでその部分を修正する Self-correcting duplex redundancy を開発している。民間用としては、SAGE の流れを汲んだ SABRE システム¹⁷⁾と呼ばれる航空座席予約システムが、IBM 7098 によるデュプレックス系として誕生している。

続いて 1960 年中頃には、集積回路を用いた No. 2 ESS, No. 3 ESS¹³⁾がコンピュータレベルの待機系として開発され、マイクロプログラム制御方式が導入された。No. 1 ESS および No. 2 ESS は処理の同期を行い、特別に両系の処理内容のつき合せユニットを持っているが、No. 3 ESS は非同期で、特につき合せユニットを持たず各系で相互にチェックを行っている。

1969 年、JPL (Jet Propulsion Lab.) の Avizienis 教授等が、JPL-STAR (Self-Testing-And Repairing)¹⁸⁾として開発した深宇宙船用コンピュータは、その後の SAMSO など宇宙船コンピュータに影響を与えたものとして有名である。このシステムの信頼性は、100,000 時間の稼働を目指しており、7個のモジュールからなる。このうちハードコアモジュール TARP (Test-And-Repair-Processor) で、バス上の信号をチェックし、機能モジュールからの状態情報を監視する。TARP モジュールは、2 out of 3 構成をとって

おり、さらに、2つの予備モジュールを持っている。機能モジュール内では、データも命令もすべてエラー検出コードで表現されており、命令、実行と同時に障害は検知され、TARP により待機予備系に切り換えられる。切り換えは、消費電力節約のため電源の入り切で行う。

1970 年代には MSI から LSI の時代になり、実装密度、消費電力、信頼性などが各段と改善された。スペースシャトルの宇宙船旅行用コンピュータ^{15),16),19)}としての信頼性は、一重障害に対してはシステム性能に無影響であり、二重以上の障害に対しても安全復帰をすること(fail operational/fail safe (FO/FS))が要求されている。これを満たすため、5台の独立した汎用コンピュータを24系統のデータベースで結合している。打上げや帰還時の誘導・航行・制御などの重要なタスクを行う時には、4台のコンピュータがソフトウェアにより同期して独立に計算し、データのつき合せを行う。残りの1台は、ソフトウェアによる障害を防ぐバックアッププログラムを実行する。また、軌道上においては、1台のコンピュータが誘導、航行制御を他の1台がシステムの管理を行い、残りの3台は搭載機器の管理または待機系となる。コンピュータは、中央処理装置(CPU)と入出力制御装置(IOP)より成っており、マイクロプログラム制御である。

またこの年代には、ミニコンピュータおよびマイクロコンピュータ方式による分散処理システムの開発が行われた。システムの目的は主として処理性の向上であったが、同時に高信頼化システムとしても注目され検討がなされている。Carnegie-Mellon 大学の C.mmp (Multi-mini-processor), Cm*(a modular multimicro processor) C. vmp (voted multiprocessor)²⁰⁾が有名である。

C. vmp は C. mmp, Cm* と異なりフォールトトレランスシステムとして工業用に研究開発されたものである。LSI-11 を3台用いて、プロセッサとメモリ間を結ぶバスに2 out of 3 の双方向 Voting 回路を入れている。3つのプロセッサは、それぞれが単独に異なったタスク処理を行うモードと同一のタスクを行って、バス上のデータの多数決をとるモードがあり、この切換えはダイナミックにプログラムで行えるようになってい

る。C. mmp と同様な構成をとっているものに電通研の PPS-R^{21),22)}がある。

一方、コンピュータネットワーク (ARPA) は、益

益拡大し、その IMP としてロッキードの SUE ミニコンピュータによる複合システム Pluribus²³⁾が開発されている。このシステムは、複数のプロセッサモジュール、メモリモジュール、I/O モジュール間を1対1に対称的にバスカプラで密結合した構成をとっている。高信頼化のための特別なハードウェアを持たず、STAGE と呼ばれる OS により、IO レベルの階層の構成制御処理を行っている。このシステムは、少々の処理中断を許容するので、ソフトウェアに障害処理をゆだねることができるのであろう。カリフォルニア大学の PRIME²⁴⁾ は ARPA 用として複数のマイクロコンピュータにメモリユニットを共用させた構成をとっており、障害時も処理の中断はせず、若干の性能低下となる。

1975 年、わが国においてもコマンドアンドコントロールシステムとして新幹線運行管理用に COMTRAC²⁵⁾が開発されている。これは、99.99% 以上の稼働率と演算出力のフェイルセーフを目標としたもので、汎用制御用計算機を3台対称に接続し、DSC と呼ばれる特別なハードウェアでタスクの同期と処理結果のつき合せを行い、残りの1台を待機系とし、1台のコンピュータが障害を起こしても瞬時自動的に2重系運転を行うようにしたものである。この方式は1982年開業の東北上越新幹線用にも採用されている。

リアルタイムトランザクション処理汎用コンピュータとして2重系を基本構成とした Non-Stop Computer と名づけた Tandem 16²⁶⁾もこの頃開発されている。このシステムは、Dynabus という2重の高速プロセッサ間バスを有しており、ソフトウェア切換えを自動的に行う。また、Dualport コントローラにより、2重系構成で避けることができない共通メモリバススイッチなど信頼性上の弱点を除いたとしている。

4. 超高信頼コンピュータシステム

石油ショックによるエネルギー問題は、機体の軽量化による省エネルギー航空機の計画を発足させた。1977年より NASA の主導で故障率 $10^{-10}/\text{hr}$ 以下の超高信頼コンピュータの開発が行われている。すなわち、1990年代の高エネルギー航空機制御用コンピュータとしてのスタンフォード研究所 (SRI) の SIFT (Software Implemented Fault Tolerant) と Draper 研究所 (CSDL) の FTMP (Fault Tolerant Multiprocessor) である。

また、各種のマイクロプロセッサを結合したシステムが、JPL によって開発されつつある。

SCCM (Self Checking Computer Module) と呼ばれる自律性を持つモジュールを結合して、広い範囲の要求に応じるシステムを構成しようとするもので FTBBC (Fault Tolerant Building Block Computer) である。

4.1 SIFT^{27)~31)}

このシステムは、標準的なコンピュータハードウェアを複数台結合し同一プログラムを走らせることにより、ハードウェアを可能なかぎり単純にし、システムのフォールトトレラント性をソフトウェアで持たせる。処理の結果のチェックとタスクの同期をソフトウェアにより行って、ハードウェア障害の検出とマスクを行う。継続する障害は、予備のハードウェアによりシステム再構成を行う。図-2 にハードウェア構成を示す。

プロセッサ間の結合手段のブロードキャストインタフェースはプロセッサからは独立して動作しており、ドライバとレシーバおよびデータファイルを持っている。データファイルは8つのデータバッファに分けられ、7つのバッファは他の7台のプロセッサよりの情報を格納し、8番目のバッファは送信すべき情報を格納する。各プロセッサは、この受信バッファの内容を取り出し、多数決タスクにより3ないし5の多数決論理処理を行う。図-3 に処理の概念図を示す。

障害発生時のシステム再構成のプログラムの関係を図-4 に示す。

タスクの起動はスケジュールテーブルに従うが、タスクを割りあてるフレーム方式を採用している。

各プロセッサで発生する2ミリ秒ごとのクロックはタスク同期のため、50マイクロ秒の誤差を許してお

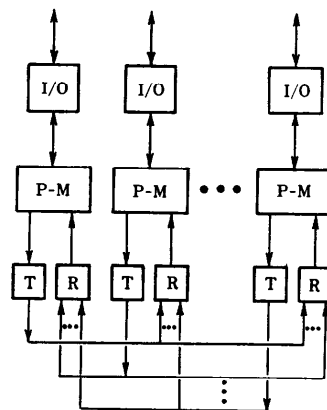


図-2 SIFT システム

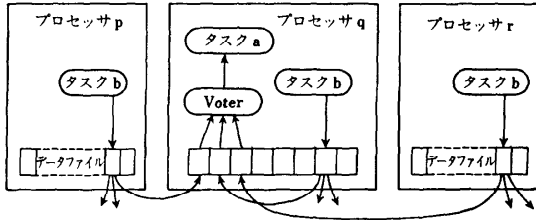


図-3 SIFT のデータ転送と多数決方式

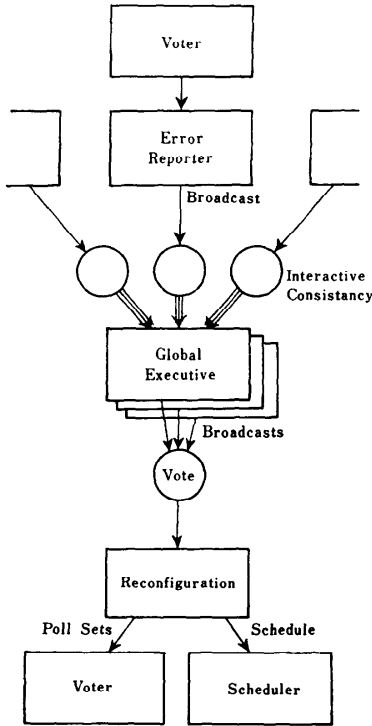


図-4 SIFT の障害処理フロー

り1つのクロックの障害をマスクするために4つのクロックを用いている。各プロセッサは他のプロセッサのクロックを取り込みクロック間のスキューを調べ、自己のスキューが他のクロックと比較して一定値を越えていれば修正する。

4.2 FTMP^{32)~34)}

実験システムとしては、LRU (live replacable unit) と呼ばれる単位で実装されており構成を 図-5 に示す。

SIFT と同様に多数決方式をとっているが、各モジュールごとに持っているバス Guardian ユニットによ

って、すべてのトランザクションをビットごとにハードウェアで多数決をとる。

システムを構成するプロセッサ、メモリ、およびこれらを結合するシリアルバスは、それぞれ予備を持った3重系として動作し、プロセッサ Triad、メモリ Triad、バス Triad と呼ばれる。

プロセッサ Triad とメモリ Triad の間のデータの転送方法を 図-6 に示す。

図-7 に示す Guardian バスユニットは、各モジュールに2組付属させ2重系を構成し、バスを選択し多数決理論で障害を検知する。

障害検知は、voter の不一致により発見され、障害が永久的なものの場合、プロセッサは使用しているバスを切りかえ、バスか、他のモジュールかの区別を行う。一時的な障害の場合疑わしいモジュールについて記録を残し、バスを切りかえ蓄積された記録から障害モジュールを探し出す。復旧処理は、正常なプロセッサが、新しく Triad を構成するユニットを決め、そのバス Guardian に対してコマンドを送ることにより行われる。

このシステムはすべてのトランザクションがハードウェアレベルでビットごとに同期しているため、ソフトウェアから見たシステムの構成は、3つのプロセッサと IO ポート、および、シェアードメモリより成っている仮想システムである。クロックシステムにおいて、各プロセッサモジュールに付属する発振器は他の3つの発振器の 2 out of 3 信号により位相同期を行う。

4.3 FTBBC^{35)~37)}

VLSI 技術、分散技術そして高信頼化技術をふんだんに採り入れたものである。4種の VLSI を用いて、市販の各種マイクロプロセッサやメモリを結合し、外部バスあるいは外部 IO と接続する。これが 図-8 に示す SCCM である。

後述の4種の BB (Building Block) は、並列バスにより結合され、それぞれ Memory-Mapped IO として動作する。したがって、各 BB は内部のプロセッサあるいは、外部バス経由で他の SCCM によりアクセスされるので、他の SCCM のメモリの読み書き、障害情報の収集、処理への介入などが可能である。

Core-BB (Core-BB) は、独立した2つのプロセッサの処理の制御、同期および照合、内部インタフェースの制御と SCCM 内の障害処理を行う。

MI-BB (Memory Interface-BB) は、メモリ制御を行い、ハミングコードによる1ビットエラーの訂正、

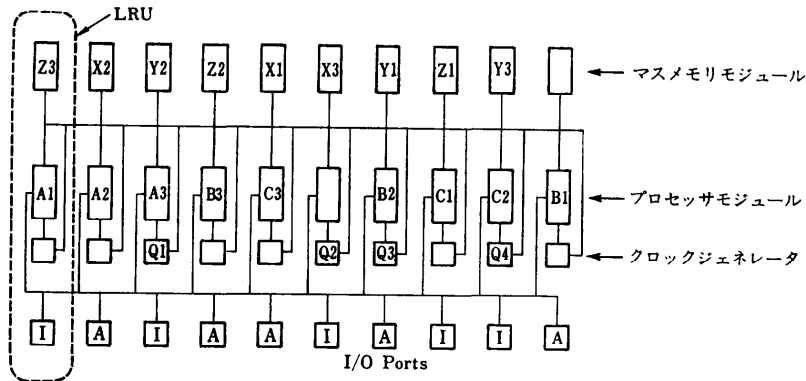


図-5 FTMP システム

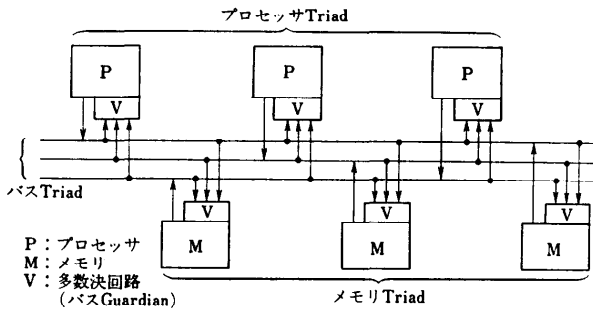


図-6 FTMP のデータ転送

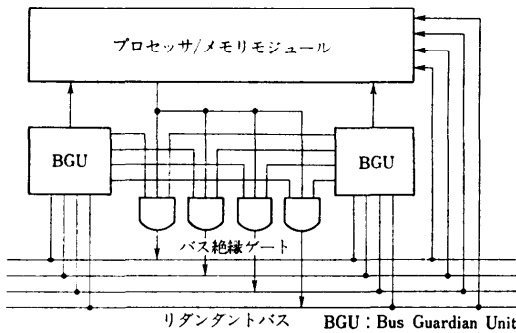


図-7 バス guardian ユニット

2ビットエラーの検出およびエラービットの置き換えを行う。

BI-BB (Bus Interface-BB) は、マイクロプログラム制御方式である。他の SCCM と結ばれた3系統の冗長バスと SCCM の内部バスを接続する。BC は3つの BA のうち1つを選んで、他の SCCM との通信を行う。

IO-BB (Input-Output-BB) は、周辺機器との結合

を行う。

FTBBC としてのシステム構成はシステムに対する要求にしたがって種々考えられる。ここでは、実験システムとしての UDS (United Data System) とよばれる階層システムの概要について紹介する。

UDS は 図-9 に示す HLM (High Level Module) としてシステムの管理制御を行う SCCM 群と、TM (Terminal Module) として被制御機器に対応して制御やデータ交換を行う SCCM 群の2階層構成である。

HLM は、10~100 ミリ秒ごとのポーリングにより TM における障害を認識して、予備系への切り換えや初期設定に必要なプログラムロードあるいは構成コマンドを SCCM の BB に発行する。

TM 内に切り換えるべき予備系がなくなってしまうと、HLM は BA、内部バス、IOBB が正常である限り、IOBB に対して直接コマンドを送り、TM の代行を行う。

中断が許されぬ管理制御用 HLM は同期2重系とし、主系の主導権により、2.5 ミリ秒ごとに外部より与えられる RTI (Real Time Interrupt) により相互監視を行う。TM はそれぞれの要求信頼度に応じて多重系、待期系をとる。

各 SCCM の制御プログラムは RTI ごとにバックグラウンドプログラムを退避し、メモリの内容を読み出し、フォアグラウンドプログラムをスケジュールテーブルにしたがって起動する。

5. む す び

高信頼コンピュータシステムの研究開発過程を特に

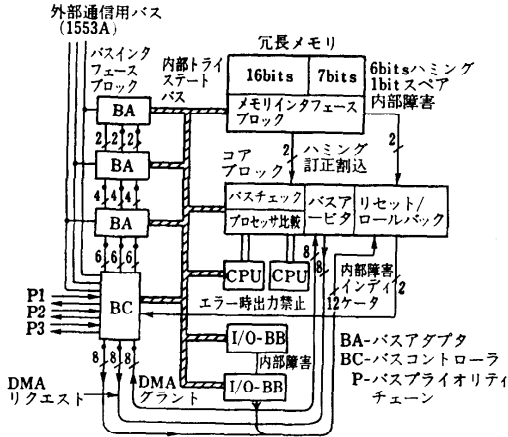


図-8 FTBCCのSCCM

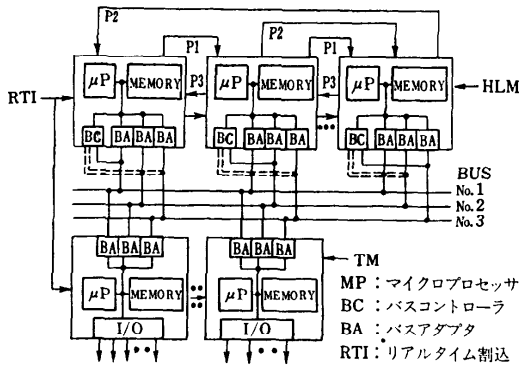


図-9 UDSの構成

その時代を代表すると思われる例を挙げて、システムレベルで眺めてみた。本稿に記載できなかったシステムも多々あるがお許しをねがい参考文献^{38)~42)}を参照願いたい。具体的な超高信頼システムとして SIFT, FTMP, FTBCC について簡単に紹介した。いずれのシステムも汎用コンピュータではないが、システムレベルでの超高信頼化への参考になるものと思われる。事実、産業用として昨年末 SIFT の流れをくむ Series 300 が発表された⁴³⁾システムレベルで超高信頼化が進むと、その信頼度を予測することが困難になり、逆に信頼度予測モデルに記述できるシステム構築の方策の研究が重要になりつつある^{30), 44)}。

参考文献

1) Sedmak, R. M. et al.: Fault Tolerant of a General Purpose Computer Implemented by Very Large Scale Integration, IEEE Trans.

comput, Vol. C-29, No. 6, p. 492 (1980).
 2) Boone, L. A. et al.: Availability, Reliability and Maintainability Aspects of the Sperry Univac 1100/60, IEEE proc. FTCS-10, p. 3 (1980).
 3) Tomita, K. et al.: A Highly Reliable Computer System-Its implementation and result, IEEE proc. FTCS-10, p. 15 (1980).
 4) Kamiuchi, T. et al.: H-80-A Load Sharing, N: 1 Back up Multisystem, Compeon Spring '78, p. 261 (1978).
 5) Avizienis, A.: Fault-Tolerance; The Survived Attribute of Digital Systems, IEEE proc. Vol. 66, No. 10, p. 1109 (1978).
 6) Avizienis, A.: The Methodology of Fault-Tolerant computing, 1st USA-JAPAN comput, conf. p. 405 (1972).
 7) 当麻: フォルトトレラントとその関連技術, 信学誌, Vol. 59, No. 4, p. 359 (1976).
 8) 当麻他: フォルトトレラントシステム, 信学誌, Vol. 63, No. 10, p. 1031 (1980).
 9) 小特集, コンピュータシステムの信頼性, 電学誌, Vol. 100, No. 8.
 10) 小特集, 高信頼化システム技術, 電学誌, Vol. 101, No. 7.
 11) Rennels, D. A.: Distributed Fault-Tolerant Computer System, IEEE Computer, Vol. 13, No. 3, p. 55 (1980).
 12) Hsiao, M. Y. et al.: Reliability, Availability, and Serviceability of IBM Computer Systems, A Quarter Century of progress IBM J. Res. Dev, Vol. 25, No. 5, p. 453 (1981).
 13) Toy, W. N.: Fault-Tolerant Design of Local Ess processor, IEEE proc. Vol. 66, No. 10, p. 1126 (1978).
 14) James, S. E.: Evolution of Real-Time Computer System for Manned Space Freight, IBM J. Res. Dev, Vol. 25, No. 5, p. 417 (1981).
 15) Cooper, A. E.: Development of on-board Space Computer Systems, IBM, J. Res. Dev. Vol. 20, No. 1, p. 5 (1976).
 16) Olson, P. F. et al.: Real time Systems for Federal Applications, A Review of Significant Technological Developments, IBM, J. Res. Dev, Vol. 25, No. 5, p. 405 (1981).
 17) Perry, M. N. et al.: American Airlines "SABRE", Electronic Reservations System proc. Western Joint Computer Conference, p. 593 (1961).
 18) Avizienis et al.: The STAR (Soft-Testing-And-Repairing) Computer, An Investigation of the Theory and Practice of Fault Tolerant Computer Design, IEEE Trans. Comput, Vol. c-20, p. 1312 (1971).
 19) Sklaroff, J. R.: Redundancy Management

- Technique for Space Shuttle Computers, IBM J. Res. Dev, Vol. 20, No. 1, p. 20 (1976).
- 20) Siewiorek, D. P. et al.: A Case Study of C.mmp, Cm*, and C. vmp, part I-Experiences with Fault Tolerance in Multi-processor Systems, IEEE proc. Vol. 66, No. 10, p. 1178 (1978).
 - 21) Murakami, K. et al.: Poly Processor System analysis and design, proc. of 4th Annual Symposium on Computer Architecture (1977).
 - 22) 村上他: コンピュータアーキテクチャの最近の進歩, 信学誌, Vol. 60, No. 6, p. 669 (1977).
 - 23) Katsuki, D. et al.: Pluribus-An operational Fault-Tolerant Multiprocessor, IEEE proc. Vol. 66, No. 10, p. 1146 (1978).
 - 24) Baskin, H. B. et al.: PRIME A modular architecture for terminal-oriented systems, proc. AFIPS SJCC Vol. 40, p. 431 (1972).
 - 25) Ihara, H. et al.: Fault Tolerant Computer System with Three Symmetric Computers, IEEE, proc. Vol. 66, No. 10, p. 1160 (1978).
 - 26) John A. Despotakis: A Fault-Tolerant Multiple Computer System, 11th Hawaii int. conf. on system science, p. 78 (1978).
 - 27) Wensley, J. H. et al.: SIFT Design and Analysis of a Fault-Tolerant Computer for Aircraft Control proc. IEEE, Vol. 66, No. 10, p. 1240 (1978).
 - 28) Weinstock, C. B.: SIFT System Design and Implementation, proc. FTCS-10, p. 75 (1980).
 - 29) Goldberg, J.: SIFT, A Provable Fault-Tolerant Computer for Aircraft Freight Control, INFORMATION PROCESSING 80 p. 151 (1980).
 - 30) Melliar-Smith, P. M.: Hierarchical Specification of the SIFT Fault Tolerant Freight Control System, SRI Technical Report CSL-123 (Mar. 1981).
 - 31) Mayer, J. F. et al.: Performability Evaluation of the SIFT Computer, IEEE Trans. Comput. Vol. C-29, No. 6, p. 501 (1980).
 - 32) Hopkins, A. L. et al.: The Architectual Elements of a Symmetric Fault-Tolerant Multi-processor, IEEE Trans. comput. Vol. C-24, No. 5, p. 498 (1975).
 - 33) Hopkins, A. L. et al.: FTMP-A Highly Reliable Fault Tolerant Multiprocessor for Aircraft, IEEE proc. Vol. 66, No. 10, p. 1221 (1978).
 - 34) Smith, T. B. et al.: Architectural Description of a Fault-Tolerant Multi-processor Engineering Proto Type, IEEE, FTCS-8, p. 194.
 - 35) Rennels, D. A.: Architectures for Fault-Tolerant Space Craft Computers, IEEE proc. Vol. 66, No. 10, p. 1255 (1978).
 - 36) Rennels, D. A. et al.: A Study of Standard Building Blocks for the Design of Fault Tolerant Distributed Computer Systems, proc. FTCS-8, p. 144 (1978).
 - 37) Carter, W. C. et al.: Computer error control by testable morphic Boolean Functions-A Way of Removing handcore, FTCS-2, p. 154 (1972).
 - 38) Megna, V. A. et al.: Development of a Multicomputer Fault-Tolerant Digital Fly-by-Wire system, proc. 3rd USA-JAPAN comput. conf. p. 336 (1978).
 - 39) Deswarte, Y. et al.: A Fault Tolerant Multi-microprocessor Architecture for SARGOS, proc. FTCS-11, p. 251 (1981).
 - 40) Sauer, A. M. et al.: The Fault-Tolerant Micro-computer system BFS, proc. FTCS-11, p. 252 (1981).
 - 41) Whiteside, A. et al.: Fault-Tolerant Multi-computer System for Control Applications, proc. FTCS-11, p. 286.
 - 42) Cioffi, G.: MuTEAM: Architectural Insights of a Distributed Multimicro-processor System, proc. FTCS-11, p. 17 (1981).
 - 43) Bernhand, R.: Minis and Mainframes, IEEE Spectrum. Vol. 19, No. 1, p. 34 (1982).
 - 44) Costes, A. et al.: SURF, A program for Dependability Evaluation of Complex Fault-Tolerant Computing Systems, IEEE FTCS-11, p. 72.

(昭和57年1月13日受付)