Vol. 36 No. 12　　情　報　処　理　　Dec. 1995

解　説

フォールトトレランス技術の展望と今後の課題†

当　麻　喜　弘††

1. まえがき

新しい装置やシステムが登場すると，その故障にどう対処するかということが常に問題になる．最近は，コンピュータの使用がネットワーク化し，その故障が広域に影響するようになるにつれ，故障が存在しても正しく機能するとか，安全に動作するといったコンピュータシステムのフォールトトレランス（耐故障性）に関心が高まり，我が国でもフォールトトレランスをキーコンセプトの1つとしたコンピュータシステムが製造販売されるようになった．

ここでは，フォールトトレラントコンピューティング（以下FTCと略記する）技術とフォールトトレラントコンピュータ（以下FTコンピュータと略記する）の発展の経過を概観し，あわせて今後の課題を考察する．

2. 年　　表

個々の技術に触れる前に，コンピュータの進歩に重要なエポックを成した諸技術とともに，全般的な発展の様子を**図-1**に見ておこう．各年代をおおまかに総括すれば，1950年代：FTC技術の理論的整備期，1960年代：FTコンピュータの揺籃期，テスト技術の発達，Totally Self-Checking（以下TSCと略記する）の概念の提唱，1970年代：FTコンピュータの本格的な開発期，フォールトトレラントソフトウェア（FTSW）の提唱，1980年代：商用FTコンピュータの出現，1990年代：日本における商用FTコンピュータの出現，ということになろう．このほか，

（1） チェックポイントリスタート，多重化，パリティチェックといった基本的概念がコンピュータの開発初期の頃からすでに提唱されている，

（2） FTコンピュータの多数派はstand-by方式で，多数決方式は少数派である．特に商用FTコンピュータではstand-by方式が圧倒的に多い，

（3） microprocessorの開発を端緒としたPC/WSの実用化の後に，商用FTコンピュータが実用化された，

（4） 我が国の基盤技術に関する貢献は10年ほど遅れて立ち上がったが，最近は，いろいろな分野で貢献している，

（5） 我が国が電子式コンピュータの時代に入ったのはコンピュータの出現からおよそ10年ほど遅れたが，FTコンピュータの開発もアメリカに比べ10年ほど遅れている，

といった点を見てとれる．

3. FTC基盤技術

3.1 誤り検出・訂正符号

まず誤りを検出しそれに応じて対応するというのがFTCの基本であるから，誤り検出・訂正符号技術はFTCの基盤技術の1つである．

2重比較による誤り検出，また，リトライによる誤り回復などはすでに1940年代のリレーコンピュータで実用化されている[2]．我が国で開発されたリレーコンピュータでも2-out-of-5符号などが用いられた．ENIACでも，その改良機にチェックポイント・リスタートの手法が試みられている[57]．1951年にパリティチェック（UNIVAC-1 Whirlwind）[3]，1960年には剰余符号が商用コンピュータ（UNIVAC III[21]）で実用化された．誤り訂正符号の主メモリへの適用は1960年から始まっている[8]（IBM Stretch）．バースト誤り訂正符号（Fire符号）は1970年にIBMのディスク3330で実用化された[47]．

† Evolution of Fault Tolerance Techniques—From the Past to the Future by Yoshihiro TOHMA (Department of Information and Communication Engineering, Faculty of Engineering, Tokyo Denki University).

†† 東京電機大学工学部情報通信工学科

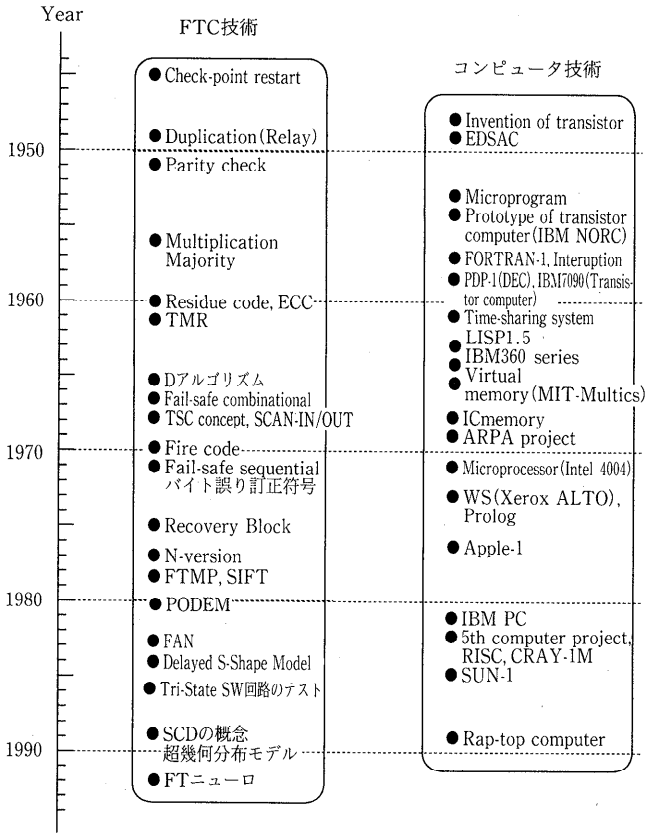


図-1　FTC 技術

1968 年に TSC の概念が初めて示された[19]．IBM-S/360 の TSC バージョンが設計されたが，実用化に至らなかった．TSC 回路を構成する場合，self-testing 性の実現が容易でないことが多い．このため最近では「TSC goal」を目指し，strongly fault-secure (SFS) な回路構成に関心が向けられている[40),62)]．

3.2 多 重 化

1950 年代半ばに，冗長性を持ち込むことで信頼性が向上できることを von Neumann や Moore-Shannon が示した[4),6)]．1965 年に Pierce は冗長設計に関する技術をいちはやく著書にまとめている[11)]．

von Neumann の多数決方式は 1960 年に入り TMR (Triple Modular Redundancy) 方式として結実した[7)]．この方式は，OAO プロジェクト☆のメモリ系[9)]や打ち上げロケット Saturn I およびV の誘導制御系[15)]に応用された．

Moore-Shannon 流の部品レベルの単純な多重化は OAO プロジェクトでも用いられたが，その後，機能ユニットレベルの多重化が行われるようになった．その代表例は 1960 年代にベル研究所で開発された電子交換機である[10)]．1970 年代以降，多数の FT コンピュータが開発されたが，その多くはこの方式に基づいている．

3.3 ソフトウェアのフォールトトレランス

高信頼性のコンピュータを実現する場合，ソフトウェアの信頼性が最近ではむしろ問題になっている[58),61)]．プログラム中に多少フォールトが含まれていても正しい結果を与えるフォールトトレラントソフトウェアの考え方が 1970 年代後半に生まれた．ハードウェアの予備切替え方式と同様な Recovery Block 方式[31)]と，多数決方式と同様な N-Version Programming[39)]が提案されている．Recovery Block 方式では Acceptance Test がキーとなるが，これがはっきりしていないのでまだ

☆ Orbiting Astronomical Observatory. 1965 年に打ち上げたが失敗した．

実用化されていないようである．N-Version Programmingは2-Version形式のものが民間航空機や鉄道の制御系で実用化されている[68]．

コンピュータシステムの開発費のうち，ソフトウェアの開発費が50%を超えるともいわれる状況で，複数のバージョンを用意しなければならない上記の手法は，特別のアプリケーションで利用されると思われる．当面は，残存フォールト数推定モデル[77]の開発やテスト手法の合理化などによって，デバッグ過程を徹底することに努力が向けられている．

3.4 我が国のFTC基礎研究

我が国では主として鉄道関係の機器をフェイルセイフ化☆することに古くから強い関心が持たれ，1966年のフェイルセイフ回路[12]の発表以来，1970年代中ごろまでにわたって，各種のフェイルセイフ論理系[13),20),23)]，論理関数の単調性に着目したフェイルセイフ組合せ回路[5),25)]，フェイルセイフ順序回路の構成理論[14),24),30)]などの研究が行われた．これらの成果の一部は実用化されている．また，フェイルセイフシステムの安全側出力を非符号語に選べばフェイルセイフの定義はFault-Secureと同じであること[36]，また，これまで提案された非順序符号を用いたフェイルセイフ順序回路がSFS順序回路であること[60]などが指摘されている．

メモリチップの集積技術の進歩にともない，隣接（バイト）誤り検出・訂正符号が実用上重要になってきた．1977年に，単一バイト誤りを訂正し，一部の2重バイトの誤りを検出する符号が提案されて以来，現在に至るまで，隣接誤り検出・訂正符号の研究が引き続き行われている[37),87),90)]．最近では，semi-distance符号や[67]，error-tolerant符号[91]などの新しい提案が行われている．

テスト容易化設計として1977年に発表されたIBMのLSSD (Level Sensitive Scan Design)が有名であるが，1968年にすでに我が国において同様なアイディアが発表されていることを強調しておきたい[18]．

テスト生成については，RothのDアルゴリズム，GoelのPODEMと続いたが，これらの無駄を排したFANが1983年に発表され[49]，一時，世界最速のアルゴリズムといわれた．我が国のテスト生成に関する研究は継続的にきわめて盛んで[55),56),63),64),66),84),85)]，1986年にはtri-state switchを含む回路のテスト生成が初めて論じられた[53]．

TSCに関連した研究としては，SFSに続きSCD (Strongly Code-Disjoint)の概念が提案され，SFSとSCDの性質を具備した回路の設計法などが発表されている[75),83)]．

我が国では，ソフトウェアのフォールトトレランスの研究はほとんど行われていず，まだソフトウェア中のフォールト（バグ）をできるだけ除去する，いわゆる，フォールトアボイダンスに力点が置かれているようである．ソフトウェアのデバッグ過程で残存するソフトウェアフォールト数を推定するモデルとして，非斉次ポアソン過程(NHPP)に基づくもの[48),51)]がよく知られているが，このほか，構成モジュールのテスト時期のずれに着目して指数関数を合成するもの[65]，テスト期間における時間の取扱いの曖昧さを避けて超幾何分布を利用するもの[69),80)]などが発表されている☆．これらは，バグを取り除くときに新たにバグは入らないと仮定しているが☆☆この可能性を考慮にいれた研究も行われている[71]．

しかし，ソフトウェアフォールトを完全に除去することは期待できず，きわめて高い信頼性が要求される場合，ソフトウェアフォールトの存在を許容するFTSWのアプローチが必要となろう．

新しいFTCのパラダイムとしてニューラルネットワークのフォールトトレランスの研究が最近始まっている[82),86),92)]．また，設計誤りを防ぐための設計検証に興味を持つ人も増えている[73]．

4. FTコンピュータの開発

4.1 世界のFTコンピュータ

1970年代になると，スペースシャトルの飛行制御用コンピュータ[33]に多数決方式が用いられるようになった．またNASAのAircraft Energy Efficiency計画の一貫として開発されたDraper研究所のFTMP[42]，SRI (Stanford

☆ 一部のジャーナリズムではフォールトトレランスとフェイルセイフを混同しているように思われるが，我々の立場ではこれら2者を明確に区別している[79]．

☆ 超幾何分布モデルは，ハードウェア設計過程における誤りの混入に対しても適用でき[93]，人間の思考過程における誤りの発生に一般的に適用できる可能性を示すものとして興味深い．

☆☆ この点について現場の批判は大きい．

Research Institute) のSIFT[28]，さらにカーネギーメロン大学のC. vmp[35]などのコンピュータも3重化多数決方式に基づいている．

機能ユニットレベルの多重化方式を用いた本格的なFTコンピュータのパイオニア機は1960年代に開発されたSTAR (Self-Test-And-Repair) コンピュータであろう[17]．これは，故障ユニットを予備ユニットに切り替えるシステムであるが，制御ユニットには3重化多数決方式を用いている．1970年代には，このほか，ARPAネットワークのメッセージ中継用コンピュータPluribus[29]，タイムシェアリングコンピュータとしてのPrime[26]，さらにカーネギーメロン大学の実験機C. mmp[27]，C. m*[38]などが開発された．メインフレームをベースにしたものとしては，航空管制用のIBM-9020[16]，汎用のUNIVAC-1100/60[45]が有名である．

最近では，VLSI技術の進歩にともない，セルフチェッキング機能を備えた計算モジュールがジェット推進研究所 (JPL) とUCLAによって開発されている[43]．また，アメリカ連邦航空局 (FAA) の音頭で進められている航空交通管制用Advanced Automation System (AAS)[54]の開発が関心を集めている．

ヨーロッパでは，きわめて早い時期にTMR方式のコンピュータSAPO[2]がチェコスロバキヤで，また1970年代に，STARコンピュータに似たMECRA[22]，2重比較方式のコンピュータCOPRA[34]などがフランスで開発されている．

1980年の前後から，TANDEM，STRATUS，その他のメーカがオンライントランザクション処理 (OLTP) を目的としたFTコンピュータを商品化した．これらの商用FTコンピュータはほとんどが予備切替え方式である．リアルタイム制御を狙った多数決方式のコンピュータも商品化されたがあまりパッとしなかった．しかし，1990年初頭に発表されたTANDEM社のIntegrity S2は多数決方式を用いており[76]，興味深い．

4.2 我が国のFTコンピュータ

我が国がいわゆる電子式コンピュータ☆，☆☆の時代に入った1960年の前後では，信頼性といえばもっぱらフォールトアボイダンスであり，フォールトトレランスはほとんど注目されなかった．

本格的なFTコンピュータが我が国で最初に開発されたのは，それからおよそ10年ほど後で，1972年にESSの日本版D-10が開発された．このときの手法はDual/Duplexである．電子交換機はその後D-70になって (1982年) マルチプロセッサのアーキテクチャを取り入れている[74]．

Real-time用FTコンピュータの代表的な実用例は新幹線制御用のCOMTRAC (COMputer-aided TRAffic Control system) であろう．COMTRACは最初dual方式で1972年に運転を始めたが，1975年には対称形の3プロセッサ方式に拡張された[32),41]．しかし動作の基本は2重比較である．残る1つは予備として待機し，不一致が検出されたときそれぞれのプロセッサに組み込まれたチェッカによって故障と診断されたものと替わる．

日立は1983年にADS (Autonomous Decentralized System) という概念に基づいた地下鉄の制御システムを開発している[59]．各構成ユニットは同一の構造をしており，各ユニット間では上下の関係はない．各ユニットは自分の判断で自律的に動作する．この方式によれば，誤りは故障ユニットに局限され，他のユニットが汚染されることはない．

2重比較チェックと回復をサポートする若干のハードウェアを組み込んだマイクロプロセッサ (GMICRO/100 FTSと呼ばれている) の開発プロジェクトが1987年に始まっている．

3重化多数決方式のFTコンピュータの開発は1980年頃から大学や鉄道総合技術研究所で行われている．最初は，東北大学でTMRに基づいたマイクロコンピュータシステムが試みられた[44]．ここでは，複数のコンピュータで同時に故障が生じる場合を想定し，各コンピュータで実行する同じタスクに時間のスキューを与えている．

1985年からJRはSMILEという電子連動システムを実用し始めたが，その原形は1980年に発表されている[46]．この設計の特色は，3重化したvoterの入出力間にさらに比較器を設けた点であ

☆ 我が国ではリレーコンピュータの後にただちにトランジスタコンピュータが実用化され，真空管式コンピュータの時代はなかったといってよい．これは世界のコンピュータの歴史上きわめて特異な点である．

☆☆ ENIACをもってコンピュータの最初の発明とする向きもあるが，いわゆる，プログラム内蔵式のコンピュータとしてはEDSAC (1949年) が最初である．

ろう．JRはフェイルセイフ性を特に重視しており，この比較器もフェイルセイフな性質を備えている．最近のJRは，3重化多数決方式よりフェイルセイフ比較器に依存した2重比較方式(BSS-Bus-level Synchronized Duplex System)に傾斜しているように見える[70),88)]．

東工大でも1980年からいわゆるSAFE (Softwarc Assisted Fault-tolerant Experimental) システムの開発を始め，1984年に実動した．4重化したプロセッサ/メモリユニット (PMUと呼んでいる) をやはり4重化したバスユニット (BU) で結合し，各PMUでのタスクの進行はソフトウェア的に同期をとっている[50)]．最初3重系で多数決をとりながら動作を進めるが，ある時期故障診断タスクを各PMUが実行しその結果を交換し合い，全体の情報で総合的に故障PMUもしくはBUを判断する．故障ユニットはシステムから除外され，予備として待機していたものが組み込まれる．

緩い同期の3重系の割り込み処理はかなり複雑で，SAFEシステムでは実質的な割り込み処理に入る前に，各PMUの状態のconsistencyを確保する前処理を行っている[52)]．

1990年1月24日に打ち上げられた人工衛星「飛天」の中に日立によって開発されたFTコンピュータが搭載されている．このコンピュータは一応3重化多数決系であるが，各ユニットに組み込まれたチェッカと，ユニット間で交換された出力を参照してvoterに加えるべき入力を適当に選ぶ．これにより，単一の故障に耐えるだけではなく，2重比較まで冗長性を効率よく利用している[72),81)]．

これまでは，いわば特種用途向きに開発されたFTコンピュータであったが，1990年代に入り商用FTコンピュータが発売されるようになった．1990年に富士通とPFUから，それぞれ，OLTP (On-Line Transaction Processing) やリアルタイム処理を目指したSURE 2000とCompact-Aが発表された．1992年には，日立がFT 6000を発表している．

SURE 2000の設計思想はメモリ共有のprimary-secondaryのペア方式である．興味あるのは，OSの機能をkernelとsubfunctionsに分け，kernelはすべてのプロセシングユニットのローカルメモリに置き，subfunctionsはサーバとして適当なプロセシングユニットに配備している点である．この結果，フォールトトレランスのほかに，OSの機能変更も無停止で行える[78)]．

Compact - AはSURE 2000の，FT 6000はCOMTRACの簡易版といった印象を受ける．前者ではユニット間の相互監視機能を強化し，後者では故障ユニットの同定を第3ユニットをまじえた少数派としている．

5. 学会活動

おそらく，FTC関連の組織だった学会活動は1962年，Washington, D. C.で開催されたSymposium on Redundancy Techniques for Computing Systemsが最初ではないかと思われる．その後，当時のIEEE Computer SocietyのPresidentであったスタンフォード大学のMcCluskey教授にUCLAのAvizienis教授が働きかけ，1970年にComputer Societyの下にTechnical Committee on Fault-Tolerant Computingができ，1971年から毎年International Symposium on Fault-Tolerant Computing (通称，FTCS) が開催されるようになったという．

1980年にはIFIP - TC 10の下にWG 10.4 (Reliable Computing and Fault Tolerance)☆というワーキンググループが設けられた．これは年2回研究会を開催し，さらに，Dependable Computing for Critical Applications (DCCA) 国際会議を主催している．

その他にも，関連したConference, Symposium, Workshopが随時開催されいる．また，3年に1度開催されるIFIP Congressにおいても，FTC関連の話題が取りあげられるようになった．

我が国では，FTCSの日本開催に備えて，1980年にFTCS日本国内委員会が電子通信学会(当時) に置かれた．1984年からは同学会にフォールトトレラントシステム (FTS) 研究専門委員会が設けられ，研究会を主催している．このほか，大阪大学の樹下教授の主催で，インフォーマルなFTC研究会が夏，冬行われている．三根先生，駒宮先生☆☆の指導の下，各位のご協力によ

☆ 最近はDependable Computing and Fault Toleranceといっている．
☆☆ 故人．

って1980年，1988年にFTCS-10，FTCS-18がそれぞれ，京都と東京で開催された．1991年には第1回のPacific Rim Fault-Tolerant Computing国際シンポジウムが川崎で開催された．

なお，1996年に仙台で再びFTCS-26を催すよう準備を進めており，皆様のご支援を再度お願いしたい．

6. 新しい動向と今後の課題

6.1 ISSRE

1980年代は，ソフトウェアの信頼性が問題であると繰り返し指摘された時期のように思われる．

ソフトウェアの信頼度なるものが評価できるかといった批判もあったが，1990年代に入ると，ソフトウェア信頼性技術（Software Reliability Engineering）というSub-CommitteeがIEEE Computer SocietyのTechnical Committee on Software Engineeringの下に設けられ，1991年からISSREという国際シンポジウムを開催するようになった．デバッグ過程やサービス動作時のモデル化と評価，ソフトウェアのテスト法，各種ツールの開発などの研究発表が行われている．

6.2 フォールト/誤りモデルとフォールト注入技術に関するワークショップ

信頼性の要求水準が高くなると，実測によってそれを検証することは事実上不可能になる．このため，故障を意図的に挿入し，フォールトトレランスが意図したとおりに機能するか，あるいは，ソフトウェアのテストが十分であるか，といった検証を行う手法が提唱されている．我が国では，この種の研究はあまり行われていないように見える．

6.3 ニューラルネットワークのフォールトトレランス

ニューラルネットワークのフォールトトレラント設計法も新たな課題であるが，ここではニューラルネットワークの学習機能を利用した新しいフォールトトレランスの可能性について触れてみたい[89]．

たとえば，EX-ORを学習によって実現した階層型のニューラルネットワークの一部の結線が切れると，もちろん，EX-ORの機能は失われる．しかし，切れたままの状態で再び学習を行うと，他の結線の値が変わり，EX-ORの機能が復元される．

これはフォールトトレランスの観点から大変興味深い．たとえば断線によってただちに誤動作しないように冗長性を持たせておきながら定期的に学習させれば，より長い間フォールトトレランス機能を保持させることが可能ではないかと考えられる．通常の予備を設けておく方式では，故障検出と同時に故障診断を行い，さらには，システム再構成が必要になるが，上の場合これらの面倒は一切不要である．

6.4 分散環境のフォールトトレランス

最近，コンピュータシステムについては，クライアント・サーバの概念や分散データベースなどのように，処理資源（機能およびデータ）を分散化しこれらをネットワークで結合する，といった傾向が強まっている．これらを見渡したとき，次のような点が問題となろう．

● Consistency

分散処理環境での複数のプロセス/プロセッサによる協調的処理が行われる場合，プロセス/プロセッサ間でグローバルな状態に関する共通的な認識を保つ必要が生じる．故障時でもこのプロセス/プロセッサ間のconsistencyを保持することは結構面倒である．

●ネットワーク

ローカルなネットワークにはまだ現実的な問題がありそうである．フォールトが生じた場所を判定する故障診断機能を組み込んでおかないとフォールトを修復することが非常に困難になる．

●タイミング

リアルタイム処理ではタイミングの制約を満たすことが重要となる．起動に対する無反応，あるいは，不適切な時間の応答，起動なしでも生じるスプリアスな出力といったタイミング誤りに対するフォールトトレランスが今後大きな問題となるのではなかろうか．タイミング誤りについては新しいアプローチが必要と思われる．

●評価

FTコンピュータシステムを実現する場合，コストの妥当な配分のためにも適正な設計を行わなければならない．このために，評価の手法を充実する必要がある．

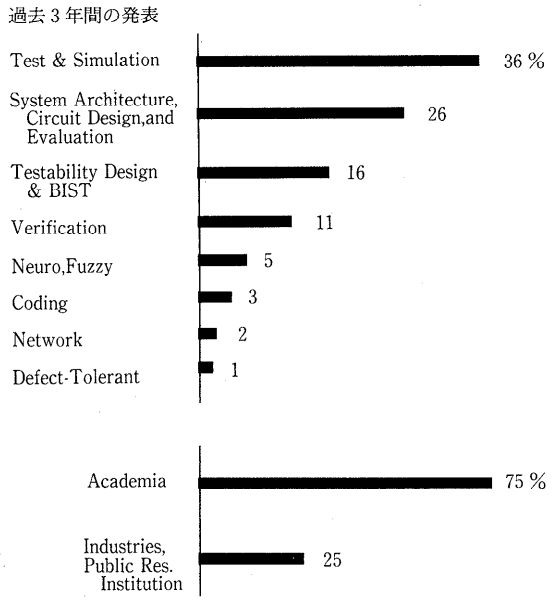


図-2 電子情報通信学会 FTS 研究会における発表論文

すでに述べたように，fault-injection technique が注目されている．我が国でもこのような実験ベッドを用意すべきではなかろうか．

ソフトウェアについては，その信頼性評価がまだ十分に行われているとは言い難い．フィールドサービス時の信頼性評価，たとえば，ソフトウェアが原因の MTBF を推定する方法をまだわれわれは手にしていない．

図-2 に最近3年間に電子情報通信学会 FTS 研究会で発表された論文の分野を示す．これより，我が国の研究がかたよっている様子が窺われる．もうすこしバランスよく，幅広い研究が行われることが重要であろう．

情報処理の手本・目標は人間が行っている高度な知的処理である．これらに留意した知的フォールトトレランスの新しいパラダイムが開かれることを期待したい．

参 考 文 献

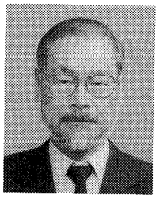
1) Oblonsky, J.: Some Features of the Czeckoslovak Relay Computer SAPO, Nachrichtentechnische Fachberichte, Vol. 4, pp. 73-75 (1956).
2) Harvard Proc., Proc. 2nd Symposium on Large Scale Digital Calculating Machinery, Annals Comp. Lab., Vol. XXII (1949).
3) Weik, M. H.: A Survey of Domestic Electronic Digital Computing Systems, Report No. 971, Commerce Department, PB-111996, Aberdeen Proving Ground, Ballistic Research Laboratories (1955).
4) von Neumann, J.: Probabilistic Logic and the Synthesis of Reliable Organisms from Unreliable Components, Automata Studies ed. by C. E. Shannon and J. McCarthy, pp. 43 98, Princeton University Press (1956).
5) Mine, H. and Koga, Y.: Basic Properties and a Construction Method for Fail-Safe Logical System, IEEE Trans. Comput., Vol. EC-10, No. 6, pp. 282-289 (June 1967).
6) Moore, E. F. and Shannon, C. E.: Reliable Circuits using Less Reliable Relays, Jour. Franklin Institute, pp. 191-208, 281-297 (Sep. 1956).
7) Brown, W. G., Tierney, J. and Wasserman, R.: Improvement of Electronic Computer Reliability through the Use of Redundancy, IRE Trans. Electron. Comput., pp. 407-417 (Sep. 1961).
8) Buchholz, W. (ed): Planning a Computer System, Project Stretch, McGraw-Hill Book Co. Inc. (1962).
9) Lewis, T. B.: Primary Processor and Data Storage Equipment for the Orbiting Astronomical Observatory, IEEE Trans. Electron. Comut. (Dec. 1963).
10) No. 1 ESS Issue, Bell Systems Technical Journal, Vol. 43, No. 5 (1964).
11) Pierce, W. H.: Failure Tolerant Computer Design, Academic Press (1965).
12) 駒宮安男, 森沢一栄, 土屋誠治: フェイルセイフ基本論理回路, 電気4学会連合大会講演論文集, No. 1987 (1966).
13) 渡辺昭治, 高橋泰司: Fail Safe 論理系と誤り訂正機能のある二重系の一構成法, 電気通信学会電子計算機研究会資料 (Jan. 1966).
14) 都倉信樹, 嵩 忠雄, 尾崎 弘: フェイルセイフ順序回路について, 電気通信学会オートマトンと自動制御研究会資料 (June 1966).
15) Anderson, J. E. and Macri, F. J.: Multiple Redundancy Applications in a Computer, Proc. 1967 Annual Symposium on Reliability, pp. 553-562 (Jan. 1967).
16) IBM Systems Journal, Vol. 6, No. 2 (1967).
17) Avizienis, A.: Design of Fault-Tolerant Computers, AFIPS Proc. Fall Joint Computer Conference (FJCC), pp. 733-742 (1967).
18) 小林, 松江, 柴: FLT に適したフリップフロップ回路, 電気通信学会全国大会, No. 892 (1968).
19) Carter, W. C. and Schneider, P. R.: Design of Dynamically Checked Computer Systems, Proc. IFIP-68, pp. 878-883 (1968).
20) 平山 博, 渡辺昭治, 浦野義頼: Fail Safe 論理系の構成理論, 電子通信学会論文誌, Vol. 52-C, No. 1, pp. 33-40 (Jan. 1969).
21) Hsiao, M. Y. and Tou, J. T.: Application of Error-Correcting Codes in Computer Reliabil-

ity Studies, IEEE Trans. Reliability, Vol. R-18, No. 3, pp. 108-118 (Aug. 1969).
22) Maison, F. P.: The MECRA: A Self-Reconfigurable Computer for Highly Reliable Process, IEEE Trans. Comput., Vol. C-20, No. 11, pp. 1382-1388 (Nov. 1971).
23) 向殿政男: C形Fail-Safe論理の数学的構造について, 電子通信学会論文誌, Vol. 55-C, No. 12, pp. 812-819 (Dec. 1969).
24) Tohma, Y., Ohyama, Y. and Sakai, R.: Realization of Fail-Safe Sequential Machines using k-Out-Of-n Codes, IEEE Trans. Comput., Vol. C-20, No. 11, pp. 1270-1275 (Nov. 1971).
25) 山本英雄: フェイルセイフ論理回路を実現する素子の条件について, 電子通信学会論文誌, Vol. 55-D, No. 1, pp. 68-69 (Jan. 1972).
26) Baskin, H. B., Borgerson, B. R. and Roberts, R.: Prime—A Modular Architecture for Terminal Oriented Systems, AFIPS Proc. Spring Joint Computer Conference (SJCC), pp. 431-437 (May 1972).
27) Wulf, W. A. and Bell, C. G.: C. mmp—A Multi-Mini-Processor, AFIPS Proc. FJCC, pp. 765-777 (1972).
28) Wensley, J. H.: SIFT—Software Implemented Fault Tolerance, 同上, pp. 243-253.
29) Heart, F. E., Ornstein, S. M., Crowther, W. R. and Barker, W. B.: A New Minicomputer/Multiprocessor for the ARPA Network, Proc. National Computer Conference, pp. 529-537 (1973).
30) Tohma, Y.: Design Technique of Fail-Safe Sequential Circuits using Flip-Flops for Internal Memory, IEEE Trans. Comput., Vol. C-23, No. 12, pp. 1149-1154 (Dec. 1974).
31) Randell, B.: System Structure for Software Fault Tolerance, IEEE Trans. Software Engineering, Vol. SE-1, No. 1, pp. 220-232 (June 1975).
32) Ihara, H. et al.: Computer Aided Traffic Control System 'COMTRAC' for Hakata Shinkansen, Hitachi Review, Vol. 24, No. 4, pp. 181-188 (1975).
33) Sklaroff, J. R.: Redundancy Management Technique for Space Shuttle Computers, IBM Jour. Res. Dev., pp. 20-28 (Jan. 1976).
34) Meraud, C. and Browaey, F.: Automatic Rollback Techniques of the COPRA Computer, Digest of Papers, 6-th International Symposium on Fault-Tolerant Computing (FTCS-6), pp. 23-29 (June 1976).
35) Siewiorek, D. P., Canepa, M. and Clark, S.: C. vmp: The Analysis, Architecture and Implementation of a Fault Tolerant Multiprocessor, Department of Electrical Engineering and Computer Sciencc, Carnegie-Mellon University (Dec. 1976).
36) 当麻喜弘: フェイルセイフシステムの理論, 電子通信学会信頼性研究会資料, R77-6 (1977).
37) Fujiwara, E. and Kawakami, T.: Modularized b-Adjacent Error Correction, Digest of Papers, FTCS-7, pp. 199 (June 1977).
38) Swan, R. J., Fuller, S. H. and Siewiorek, D. P.: Cm*: A Modular Multi-Microprocessor, Proc. AFIP Conference, Vol. 46, pp. 637-644, 645-655 (1977).
39) Avizienis, A. and Chen, L.: On the Implementation of N-Version Programming for Software Fault Tolerance during Execution, Proc. COMPSAC, pp. 149-155 (Nov. 1977).
40) Smith, J. E. and Metze, G.: Strongly Fault Secure Logic Networks, IEEE Trans. Comput., Vol. C-27, No. 6, pp. 491-499 (June 1978).
41) Ihara, H, Fukuoka, K., Kubo, Y. and Yokota, S.: Fault-Tolerant Computer System with Three Symmetric Computers, Proc. IEEE, Vol. 66, No. 10, pp. 1160-1177 (Oct. 1978).
42) Hopkins, A. L., Smith, T. B. and Lala, J. H.: FTMP—A Highly Reliable Fault-Tolerant Multiprocessor for Aircraft, Proc. IEEE, Vol. 66, No. 10, pp. 1221-1239 (Oct. 1978).
43) Rennels, D. A., Avizienis, A. and Ercegovac, M.: A Study of Standard Building Blocks for the Design of Fault-Tolerant Distributed Computer Systems, Digest of Papers, FTCS-8, pp. 144-149 (June 1978).
44) 亀山充隆, 樋口龍雄: TMRによるフォールトトレラントマイクロコンピュータシステムの一構成法, 信学技報, EMCJ78-57, pp. 11-16 (Jan. 1979).
45) Boone, L. A., Liebergot, H. L. and Sedmak, R. M.: Availability, Reliability and Maintenabilty Aspects of the Sperry-Univac 1100/60, Digest of Papers, FTCS-10, pp. 3-8 (Oct. 1980).
46) Kawakubo, K., Nakamura, H. and Okumura, I.: The Architecture of Fail-Safe and Fault-Tolerant Computer for Railway Signaling Device, Digest of Papers, FTCS-10, pp. 372-374 (Oct. 1980).
47) Hsiao, M. Y., Carter, W. C., Thomas, J. W. and Stringfellow, W. R.: Reliability, Availability, and Serviceability of IBM Computer Systems: A Quarter Century of Progress, IBM Jour. Res. Dev., Vol. 25, No. 5, pp. 453-465 (Sep. 1981).
48) Ohba, M. and Kajiyama, M.: Inflection S-Shaped Software Reliability Growth Model, Proc. WGSE Meeting, Vol. 28, Information Processing Society (IPS) of Japan (1983).
49) Fujiwara, H. and Shimono, T.: On the Acceleration of Test Generation Algorithms, IEEE Trans. Computers, Vol C-33, No. 12, pp. 1137-1144 (Dec. 1983).
50) 米田友洋, 河村俊明, 古屋 清, 当麻喜弘: 多数決に基づく耐故障システムの故障診断とシステ

ム再構成, 電子通信学会論文誌, Vol. J67-D, No. 7, pp. 738-744 (July 1984).
51) Yamada, S., Ohba, M. and Osaki, S.: S-Shaped Software Reliability Growth Models and Their Applications, IEEE Trans. Reliability, Vol. R-33, No. 4, pp. 289-292 (Oct. 1984).
52) Yoneda, T., Suzuoka, T. and Tohma, Y.: Implementation of Interrupt Handler for Loosely-Synchronized TMR Systems, Proc. FTCS-15, Ann Arbor, Michigan, pp. 246-251 (June 1985).
53) Itazaki, N. and Kinoshita, K.: Algorithmic Generation of Test Patterns for Circuits with Tri-State Modules, Proc. FTCS-16, Vienna, Austria, pp. 64 69 (July 1986).
54) Avizienis, A. and Ball, D. E.: On the Achievement of a Highly Dependable and Fault Tolerant Air Traffic Control System, Computer, Vol. 20, No. 2, pp. 84-90 (Feb. 1987).
55) Takamatsu, Y. and Kinoshita, K.: CONT: A Concurrent Test Generation Algorithm, Proc. FTCS-17, Pittsburgh, Pennsylvania, pp. 22-27 (June 1987).
56) Furuya, K.: A Probabilistic Approach to Locally Exhaustive Testing, 同上, pp. 62-65.
57) Carter, W. C.: Experiences in Fault-Tolerant Computing, 1947-1971, The Evolution of Fault-Tolerant Computing, ed. by A. Avizienis, et al., pp. 1-36, Springer-Verlag/Wien (1987).
58) Clement, G. F. and Giloth, P. K.: Evolution of Fault-Tolerant Switching Systems in AT & T, 同上, pp. 37-54.
59) Ihara, H., Mori, K., Miyamoto, S.: Evolution of Reliable Computing in Hitachi and Autonomous Decentralized System, 同上, pp. 77-99.
60) Nanya, T. and Kawamura, T.: A Note on Strongly Fault-Secure Sequential Circuits, IEEE Trans. Comput., Vol. C-36, No. 9, pp. 1121-1123 (Sep. 1987).
61) Hsueh, M. C. and Iyer, R. K.: A Measurement-Based Model of Software Reliability in a Production Environment, Proc. COMPSAC87, Tokyo, pp. 354-360 (Oct. 1987).
62) Nanya, T. and Kawamura, T.: Error Secure/Propagating Concept and its Application to the Design of Strongly Fault Secure Processors, IEEE Trans. Comput., Vol. C-37, No. 1, pp. 14-24 (Jan. 1988).
63) Fujiwara, H.: Computational Complexity of Controllability/Observability Problems for Combinational Circuits, Proc. FTCS-18, Tokyo, Japan, pp. 64-69 (June 1988).
64) Teshima, S., Chujo, N., Sano, N., Nagase, H. and Takigawa, M.: Accelerated Fault Simulation by Propagating Disjoint Fault-Sets, 同上, pp. 116-121.
65) Matsumoto, K., Inoue, K., Kikuno, T. and Torii, K.: Experimental Evaluation of Software Reliability Growth Models, 同上, pp. 148-153.
66) Nanya, T., Mourad, K. and McCluskey, E. J.: Multiple Stuck-at Fault Testability of Self-Testing Checkers, 同上, pp. 381-386.
67) Matsuzawa, K. and Fujiwara, E.: Masking Asymmetric Line Faults Using Semi-Distance Codes, 同上, pp. 354-359.
68) Voges, U. ed.: Software Diversity in Computerized Control Systems, Chapter 2, pp. 7-21, Springer-Verlag/Wien (1988).
69) Tohma, Y., Tokunaga, K., Nagase, S. and Murata, Y.: Structural Approach to the Estimation of the Number of Residual Software Faults based on the Hyper-Geometric Distribution, IEEE Trans. Software Eng., Vol. 15, No. 3, pp. 345-355 (Mar. 1989).
70) Hasegawa, Y. et al.: A New Train Control System by Radio, Proc. IEEE Vehicular Technology Conference (May 1989).
71) Ohba, M. and Chou, X. M.: Does Imperfect Debugging Affect Software Reliability Growth?, Proc. 11-th International Conference on Software Engineering, pp. 237-244 (1989).
72) Kanekawa, N., Maejima, H., Kato, H. and Ihara, H.: Dependable Onboard Computer Systems with a New Method—Stepwise Negotiating Voting, Proc. FTCS-19, Chicago, Illinois, pp. 13-19 (June 1989).
73) Yoneda, T., Nakade, K. and Tohma, Y.: A Fast Timing Verification Method Based on the Independence of Units, 同上, pp. 134-141.
74) Yamada, T. and Ogawa, S.: Fault-Tolerant Multi-Processor for Digital Switching Systems, 同上, pp. 245-252.
75) Nanya, T. and Uchida, M.: A Strongly Fault-Secure and Strongly Code-Disjoint Realization of Combinational Circuits, 同上, pp. 390-397.
76) FT Systems, No. 89/90 (Jan./Feb. 1990).
77) 当麻喜弘: ソフトウェアの信頼性, 情報処理, Vol. 31, No. 4, pp. 508-517 (Apr. 1990).
78) 村松 洋, 伊達政広, 吉田 浩, 北岡正治, 黒羽法男, 岩田勝行: システムを止めずに保守・運用が可能なOSを開発, 日経エレクトロニクス, No. 520, pp. 209-223 (Feb. 1991).
79) 当麻喜弘, 南谷 崇, 藤原秀雄: フォールトトレラントシステムの構成と設計, 槇書店 (Mar. 1991).
80) Tohma, Y., Yamano, H., Ohba, M. and Jacoby, R.: The Estimation of Parameters of the Hyper-Geometric Distribution and its Application to the Software Reliability Growth Model, IEEE Trans. Software Eng., Vol. 17, No. 5, pp. 483-489 (May 1991).
81) Takano, T., Yamada, T., Shutoh, K. and Kanekawa, N.: Fault Tolerance Experiments

of the "Hiten" Onboard Space Computer, Proc. FTCS-21, Montreal, Canada, pp. 26-33 (June 1991).
82) 丹 康雄: フォールトトレランスを獲得する階層形ニューラルネットワークとその性質に関する研究, 東京工業大学学位論文 (Mar. 1992).
83) Nanya, T., Hatakeyama, S. and Onoo, R.: Design of Fully Exercised SFS/SCD Logic Networks, Proc. FTCS-22, Boston, Massachusetts, pp. 96-103 (July 1992).
84) Fujino, T. and Fujiwara, H.: An Efficient Test Generation Algorithm Based on Search State Dominance, 同上, pp. 246-253.
85) Kajihara, S., Shiba, H. and Kinoshita, K.: Removal of Redundancy in Logic Circuits under Classification of Undetectable Faults, 同上, pp. 263-270.
86) Koyanagi, Y. and Tohma, Y.: Fault Tolerant Neural Networks in Optimization Problems, 同上, pp. 412-418.
87) Fujiwara, E. and Hamada, M.: Single b-Bit Byte Error Correcting and Double Error Detecting Codes for High-Speed Memory Systems, 同上, pp. 494-501.
88) Nakamura, H. and Takeshi, K.: Fault-Tolerant Microcomputer Design and Application for Railway Train Control, Information Processing 92, Vol. 1, Elsevier Science Publishers B. V., pp. 652-658 (Sep. 1992).
89) 当麻喜弘: 知的情報処理の自己修復・再生, 人工知能とニューロコンピュータ, 「大学と科学」公開シンポジウム組織委員会編, クバプロ, pp. 204-208 (Sep. 1992).
90) Fujiwara, E. and Kitakami, M.: A Class of Error Locating Codes for Byte-Organized Memory Systems, Proc. FTCS-23, Toulouse, France, pp. 110-119 (June 1993).
91) Matsubara, T. and Koga, Y.: A Proposal for Error-Tolerating Codes, 同上, pp. 130-136.
92) Tohma, Y. and Koyanagi, Y.: Design of Neural Networks to Tolerate the Mixture of Two Types of Faults, 同上, pp. 268-277.
93) Tohma, Y. and Matsunaga, Y.: Application of Hyper-Geometric Distribution Model to Hardware Debugging Process, The 1993 Pacific Rim International Symposium on Fault-Tolerant Computing, Melbourne, Australia (Dec. 1993).

(平成7年2月13日受付)



当麻 喜弘 (正会員)

1933年生. 1956年東京工業大学工学部電気工学科卒業. 1961年同大学院博士課程修了. 工学博士. 同大学に33年間勤務後, 1994年定年退官, 同大学名誉教授. 現在, 東京電機大学情報通信工学科教授. スイッチング回路理論, フォールトトレラントシステムなどの研究に従事. 著書「順序回路論」,「スイッチング回路理論」,「フォールトトレラントシステムの構成と設計」(共著)など. 1982-83年本会理事. 電子情報通信学会, ソフトウェア科学会, 人工知能学会各会員. IEEECS-TCFTC の Advisory Board Member, IFIPWG 10.4 の Vice-Chairman. 1960年電気通信学会稲田賞, 1965年同岡部賞, 1976年電子通信学会著述賞, 1980年 IEEE Fellow, 1992年 IFIP Silver Core 賞, 1994年大川出版賞.