

FROM THE UNIVERSITY OF ILLINOIS VIA JPL AND UCLA TO VYTAUTAS MAGNUS UNIVERSITY: 50 YEARS OF COMPUTER ENGINEERING BY ALGIRDAS AVIŽIENIS

David A. Rennels¹ and Miloš D. Ercegovic
Computer Science Department
University of California, Los Angeles
rennels@cs.ucla.edu, milos@cs.ucla.edu

¹*Also with the Jet Propulsion Laboratory*
Pasadena, California

Abstract This article surveys pioneering contributions of Algirdas Avizienis to the fields of fault-tolerance and dependable computing, digital arithmetic, and computer design, made during his 50-year long career in computer engineering.

Keywords: Dependable computing, fault-tolerance, arithmetic error codes, signed-digit arithmetic

1. Introduction

Algirdas Avizienis was born in Kaunas Lithuania in 1932, the son of a Lithuanian army officer. His family escaped the invasion of the Soviets during World War 2, and as a teenager, he and his family wound up in a displaced persons camp where he learned some of English reading Western novels. The family emigrated to Chicago, that then had a large Lithuanian community, and he worked his way through the University of Illinois. He has had a very successful career, becoming a senior professor, a leader of his research field, and a fellow of the IEEE; receiving numerous awards and an honorary doctorate, serving on many important advisory boards, and finally becoming the founding Rector of the re-established National Lithuanian University. A successful career is characterized by chance events, and a person who has to optimism to recognize opportunities in these events, and the creative abilities to take advantage of them and create something that has not existed before.

Al has done this and through a long career of teaching has made many friends and colleagues whose careers have been enhanced by his work.

2. Algirdas Avizienis and the Development of Fault-Tolerant Computing

Professor Algirdas Avizienis has been a leading figure in fault-tolerant computing for nearly a half-century, and his technical contributions have been seminal in its development. In addition, his leadership in founding the scholarly community of interest through IEEE and IFIP has shaped the growth of the field and greatly facilitated its development. This paper is a reflection on some of his major contributions, but it is not comprehensive – that would require a whole book.

The Beginning (1955-1962)

Upon finishing his MS at the University of Illinois (UIUC) in 1955, he took a job in sunny California with what was then a US Army Laboratory, Caltech's Jet Propulsion Laboratory (JPL), in Pasadena. There, he worked on developing the Corporal and Sergeant missiles - capable of carrying a nuclear warhead up to 100 miles. This was very early technology where target information was entered by turning dials (potentiometers) on the missile, and operator error under battlefield conditions would be catastrophic. This was Avizienis' first exposure to a problem of dependability for critical systems, and he came up with a design to enter the data using punched cards, automatically set the dials, and independently read out their settings. Although he claims that this, his first design was rather kludgy, he was told that parts of it were used in the subsequent Sergeant production models. The early JPL environment was very exciting, but he quickly realized that to take a leadership role, he would need a PhD. So after a year, he took leave of absence and went back to UIUC to finish a Ph.D. under the great computer arithmetic scholar, Jim Robertson. There he increased in wisdom and stature, and he also gained experience in logic design by designing parts of the ILLIAC II arithmetic unit.

While at UIUC new unexpected events and opportunities occurred that would propel him into the leadership of fault-tolerant computing. The USSR, launched Sputnik, the Army with JPL launched the first US satellite (Explorer 1) and an alarmed Congress passed the National Aeronautics and Space Act creating NASA. By the time that Avizienis returned to JPL in 1960, still in his twenties, JPL had been designated a NASA laboratory. Their mission of exploring space required computers that could last for up to 10 years in space without hope of external repair.

Furthermore the amount of electrical power from early solar arrays could only be a few tens of watts, and the state of integrated circuit technology that could be used in the space radiation environment (i.e., bipolar) was only a few gates or flip flops per chip.

Within a year or so (1961), Avizienis came up with the idea of creating a low-power, long life computer using self-repairing techniques by:

- partitioning a computer into modules of smaller size to obtain adequately low module failure rates;
- using codes to provide low cost concurrent error detection - allowing only one module of each type to be operating and thus saving power;
- providing a simple small hybrid redundant core (voting with un-powered spares) that would rollback and/or replace modules that detect errors.

Here, his graduate work in computer arithmetic fit perfectly, because the key to implementing modules that checked themselves lay in the use of low-cost coding techniques (product codes, residue codes) that are invariant under arithmetic. He spent the next few years refining the architectural approach and the coding techniques that could be used in implementing the needed computer modules. This involved partitioning logic onto chips such that single chip failures would remain detectable, and analyzing the coverage of various codes. Here he developed the clever idea of organizing a system in an k -bit serial fashion, so that code checking of either product or residue codes based on a modulus of $2^k - 1$ could be done by simply performing a simple modulo $2^k - 1$ addition of digits as they passed by [4].

While remaining a member of the JPL technical staff, Avizienis joined the faculty of UCLA in 1962 teaching undergraduate and graduate courses on computer design and computer arithmetic.

1962-1972 The STAR Computer Decade

Technical Contributions By 1965-66 his conceptual design had matured to the point where an initial product-coded arithmetic processor was built and tested, and plans were made to implement the complete computer (with the other modules using inverse residue codes). The machine was designated the JPL Self-Testing and Repairing (STAR) computer. In developing the computer, many new research issues were raised, and Avizienis took on a team of graduate student advisees (also working as JPL engineers) who have remained his friends and colleagues for nearly forty years. Among the research issues addressed were:

- how to program such a machine - resulting in the design of an operating system, assembler, and some of the earliest work on program rollback (John Rohr);
- how to do hardware design and evaluation of such a machine - resulting in a design methodology and some of the earliest work on experimental fault-injection tests (David Rennels);
- how to provide reliability prediction of the STAR Computer - resulting in the development of new hybrid redundancy models based on recursive integral equations (Frank Mathur);
- how to use such a machine as an automated repairman for its host spacecraft - resulting in the first comprehensive study of using fault-tolerant design on a spacecraft (George Gilley).

There was an air of excitement as the machine progressed, since nothing like this had been attempted before, and by the date of the First International Symposium on Fault-Tolerant Computing (FTCS 1) the machine was working and ready to be displayed. (As an aside, it almost wasn't ready. A laboratory power supply unexpectedly jumped from 5 to 50 volts, and this author spent the weekend before FTCS 1 soldering new IC flat-packs onto the circuit boards of one of the modules.)

The primary papers on the JPL-STAR Computer were published in 1971 [10, 11], and having coined the term *Fault-Tolerant Computing* Avizienis wrote an article in *Computer Magazine* giving an overview of what it meant [12]. Related papers were published over the next several years [40, 14, 15, 32, 46, 51].

Perhaps Avizienis' shortest publication of this period, was a letter to the editor of a national news weekly. Being a champion table tennis player at UIUC, he wrote to complain that "if they wanted to refer to table tennis as *ping pong* then they should also refer to basketball as *dribble drabble* or golf as *put put*".

Creating the IEEE Technical Committee on Fault-Tolerant Computing

In initiating his STAR computer research, Avizienis visited and corresponded with others he could find who were doing advanced work in dependable systems, including the Saturn V at NASA Marshall, and the IBM Federal Systems Division. JPL subcontracts were awarded to the Stanford Research Institute (Now SRI) to design a voting power switch for the STAR computer, and the MIT Instrumentation Laboratory (now CS Draper Lab) for a read-only memory. Thus due to the STAR project, he had established contacts with many of the people who were to become the leading players in dependable computing.

Yet at that time there were few relevant conferences, and there were no regularly organized meetings or professional community dedicated to fault-tolerant computing. So Avizienis organized a Workshop on the Organization of Reliable Automata, sponsored by UCLA and the IEEE Technical Committee on Switching Circuit Theory and Logic Design in February 1966. The turnout and quality of work at this workshop demonstrated that a critical mass had been reached in this field, with representatives from MIT, MIT Instrumentation Laboratory, Stanford, SRI, UC Berkeley, Princeton, UIUC, IBM Research, University of Michigan, Bell Telephone Laboratories, Honeywell, Westinghouse, Universities of Kyoto and Osaka, Aerospace Corporation, et al.

In early 1969, Avizienis proposed to the IEEE that a new Technical Committee on Fault-Tolerant Computing be established, and it was approved in November 1969 with Avizienis as its first chair. The first order of business was to establish an annual conference, and the first International Symposium on Fault-Tolerant Computing took place in Pasadena, CA with Avizienis as general chair and Bill Carter as Program Chair. FTCS was international from the start, with papers from Japan, France and England as well as the US, and over the years, it has been hosted by research groups in Japan and in six European countries. Thirty four FTCS/DSN conferences later, one can say that the establishment of this community has had an enormous impact on dependable computing technology. It has influenced the careers and friendships of innumerable people involved in it.

1972-1980 The shift to UCLA

At the conclusion of the JPL STAR effort, Avizienis turned his primary attention to research at UCLA. He obtained a five-year National Science Foundation Grant titled Fault-Tolerant Computing that enabled the establishment of the Dependable Computing and Fault-Tolerant Systems (DC-FTS) Laboratory at UCLA. The scope of its research has been very broad, encompassing many aspects of fault-tolerant system architecture, dependability modeling, and even formal specifications and program correctness. It is estimated that about 200 publications, 31 PhD dissertations and 20 M.S. theses have resulted from research started by Avizienis at JPL and in the DC-FTS laboratory. Of the original thirty, five of the PhD's have gone on to university faculty positions and the rest to responsible positions in government and industry.

It was Avizienis' style to initiate many graduate student projects of his own, but also to draw upon the strengths at UCLA by involving other

faculty as collaborators in applying fault-tolerance to new application areas. These areas and faculty collaborators are briefly described below:

- High Performance Numerical Processing (with Miloš Ercegovac and Tomás Lang) – Array processors and reconfigurable arrays for high speed numeric computation [16, 18, 20, 22, 29, 45, 54, 56] (3 PhDs)
- Memory systems (with Wesley Chu) – fault-tolerance of multiport memories [38]. (1 PhD)
- Database machines (with Alfonso Cardenas) – fault-tolerance issues in implementing associative processors [30]. (1 PhD)
- On-Line arithmetic and VLSI testing (with Miloš Ercegovac) – error coding algorithms for on-line arithmetic and design of testable CMOS chips [33, 42]. (2 PhDs)
- Computer networks (with Mario Gerla) – fault-tolerant ring networks and the use of Stochastic Petri nets to prove correctness and performance [34, 35].
- Modular systems composed of self-checking VLSI-based building blocks (with David Rennels and Miloš Ercegovac) [47].
- Formal specifications and program correctness (with David Martin) – compiler correctness and the syntheses of correct microprograms (2 PhDs)

Avižienis extended his work on error detecting codes used in the STAR computer with algorithms for two-dimensional residue codes which allowed error correction as well as detection [26]. Other studies were conducted to explore external monitoring and diagnosis.

He was especially interested in issues of fault-tolerant VLSI design and directed studies on self-checking VLSI design, yield enhancement, and techniques to enhance their testability. This resulted in techniques to use redundancy to improve chip yields [39] and to implement self-checking programmable logic arrays [57, 52].

He maintained his interest in reliability modeling and he directed the development of several new prediction models. A major advance in Markov modeling was contributed through the Ph.D. dissertation of Y. W. Ng, who devised an unified model that introduced transient faults degradability and repair [41]. The ARIES 76 reliability modeling system (written in APL) contained all these features and found wide acceptance for education, research, and in industry.

In about 1975 Avizienis turned his attention to the hard question of how to provide tolerance against software faults through Multi-Version Programming (MVP) alternatively called N-Version Programming. Here he proposed an approach akin to replication and voting in hardware, i.e., Multi-Version Software (MVS), that could be voted to override an error in any single version. All versions are partitioned into smaller modules whose results are individually voted. Thus all versions may have an error, but the system will continue correctly as long as there are not enough errors in the versions of any individual module to overwhelm the voting. An experiment was conducted using students in a software engineering class, and the results were satisfactory: faulty versions were outvoted and the redundant software continued to perform satisfactorily [19].

It was recognized however, that the dependability of MVS depended upon the availability of high quality specifications. It also depended on the various software versions not having related faults (i.e., maximizing diversity of the design). Furthermore it had yet to be applied to a critical real-world application. Thus the stage was set for continued investigation.

Design Diversity 1980-1990

During this period Avizienis organized and directed large scale experiments to explore the issues of multiversion software in depth.

Specification Languages To examine the effect of specification techniques on multi-version software, an experiment was designed in which three different specifications were used. The first was written in the formal specification language OBJ. The second specification language chosen was the non-formal PDL that was characteristic of current industry practice. English was employed as the third, or "control" specification language, since English had been used in the previous studies. Teams of programmers wrote MVP programs which were then run to determine the accuracy of the resulting code. Although there were errors in modules of some of the software versions, the MVP programs executed correctly when all versions were run and voted. It was determined that there were more code errors in software generated from the formal specification language than when using non-formal PDL [36, 23].

The DEDIX Testbed In order to provide a long-term research facility for design diversity experiments his research group implemented the DEDIX (DEsign DIversity eXperiment) system, a distributed supervisor and testbed for multiple-version software. It provided tools to

supervise and monitor the execution of N diverse versions of an application program functioning as a fault-tolerant N -version software unit. DEDIX also provided Input/Output services and a transparent interface to the users writing individual versions, so that they need not be aware of the existence of multiple versions and the implementation of recovery algorithms. It began operation in 1965 [25]. The design team that implemented this system, in addition to Avizienis and John Kelly, included a remarkable international group. They included Per Gunningberg (Uppsala), Lorenzo Strigini (Pisa), Pascal Traverse (Toulouse) and Udo Voges (Karlsruhe).

To apply this methodology to a real-application, Avizienis collaborated with Honeywell/Sperry, who provided a problem of pitch control for automated landing systems [27]. A MVS experiment was conducted with the programming teams using different languages to further increase the diversity of their versions (C, Pascal, Modula2, Prolog, and T). Very few errors were found after acceptance testing, and the redundancy of the voted programs provided correct operation. Their paper concluded that the technology was ready for use in commercial applications. It was also concluded, however, that the specification and programming methodology developed over extensive experiments must be followed very carefully to preserve diversity [37].

As a result of Avizienis original MVS research, NASA initiated a Multi-University N -Version Programming Experiment in which UCLA and three universities were involved in writing N -version programs for control of a Redundant Strapped Down Inertial Measurement Unit. The results provided a large amount of data on preserving diversity in MVP programming, effectiveness of MVP, and methodologies for code testing.

Multi-version programming was somewhat controversial when it was started because it is a hard and open-ended problem without simple answers. It required building tools, conducting patient experiments involving teams of students, and its practical applications were seen to be limited and not provable. Only when one looks deeper into this does one recognize its major contributions to the *science* of dependable computing.

- First it asks one of the hardest of questions. How can we get correct operation when people make design mistakes? Then it explores the best ways we know how to do this – providing qualitative and quantitative results and insights.
- Second, the experiments provided extensive data on how programming errors are made and what procedures can be used to minimize these errors – thus providing invaluable data in software engineer-

ing. In the course of doing these experiments, diverse design has proven itself as an effective technique for program debugging and even finding errors in specifications.

- Third, there are highly critical applications where diverse design has been selected as the only way to meet stringent reliability requirements.

Following the MVP research, Avizienis turned his attention back to hardware architecture. He and a student have done an extensive study of design errors in modern microprocessors and, since retiring from UCLA he has developed a new distributed architecture for COTS components [28].

Creation of the IFIP Working Group 10.4 on Dependable Computing and Fault Tolerance

As the research community in fault-tolerant and dependable computing (that Al Avizienis had such a large part in building) became large, highly international and well established, it became clear that it should be represented in the International Federation of Information Processing. Al Avizienis was the primary person in establishing this IFIP working group in October 1980 and served as its first chairman. It has become a mechanism where leading international researchers regularly gather and address new issues that arise in this important technical area.

3. Contributions to Digital Arithmetic

In his landmark 1960 doctoral dissertation [2, 3] on signed-digit number systems and arithmetic algorithms, Avizienis provided the foundation and formalisms enabling their systematic use in the design of arithmetic units. Signed-digit number systems had been previously recognized as useful. They were used in the Booth multiplier recoding in radix 2 of the set $\{0,1\}$ to $\{-1,0,1\}$, and similarly in higher radices $r > 2$. Likewise, the use of redundancy in quotient digit sets allowed faster division algorithms such as the SRT division [50] and led to generalized high-radix non-restoring division investigated in detail by Robertson [48] and others at the University of Illinois in the 60s and 70s [50, 49]. Interestingly, the potential advantages of signed-digit numbers were noticed by many early mathematicians including Augustine Cauchy who discussed signed-digit numbers as early as 1840.

Avizienis was first to propose and develop general algorithms for arithmetic operations on signed-digit representations to achieve a "closed" arithmetic system. His "totally parallel" addition algorithm elegantly

eliminates the notorious carry problem by allowing redundancy in each digit position. This led to a two-step addition algorithm that can be executed in constant time, that is, independently of the number of digits. In other words, redundant arithmetic algorithms retain weighted representations while "decoupling" digit positions. As a consequence, one can also perform addition and other operations in the most-significant-digit-first manner. Thus Avizienis' work ultimately led to online arithmetic where all operations can be performed most-significant-digit-first (MSDF), generating the result digits while consuming the input digits [31]. As discussed by Avizienis in [5, 7], the MSDF class of arithmetic algorithms has inherent capabilities to perform variable-precision operations and to keep track of significance of generated digits .

The use of redundant number systems has become pervasive and represents one of the most important developments in the field of digital arithmetic. Avizienis' work on redundant representation systems provided much of the underlying framework, and it also provided a framework for studying other important techniques such as digit-set recoding, carry-save addition, and, as mentioned, online arithmetic. In [6] he developed arithmetic microsystems suitable for IC implementations based on signed-digit arithmetic. Further work on a universal arithmetic building element (ABE) and design methods for arithmetic processors was described in [8]. These ideas were applied in the design of combinational arithmetic systems for approximation of functions [55]. Complexity issues of redundant arithmetic were investigated in [17].

The first extensive use of signed-digit number system and algorithms was in the Illiac III computer as described in [1]. A quick look at the literature indicates that signed-digit number system have been frequently used in both general-purpose processors and in application-specific processors, in particular, in digital signal processors.

Avizienis is also widely known for his novel work on low-cost arithmetic error codes. In particular, he made original contributions to low-cost arithmetic error-detection and correction codes [13] and developed efficient algorithms for error-coded operands [15]. These algorithms were implemented in the radix-16 processor constructed for the Jet Propulsion Laboratory STAR computer mentioned above. The 1971 paper is a classic which was selected as the representative paper on the topic in the 1982 text on "Reliable System Design" by D. Siewiorek and R.S. Swarz. His ideas on error codes led to research in their applications in mass memories [43, 44]. In [22] the low-cost error codes have been investigated in the context of large high-performance computers, and they were extended to signed-digit operands in [21]. His two-dimensional low-

cost residue and inverse residue codes are novel and appear to be very useful for checking of memories as well as of byte-oriented processors [24].

Besides mentoring a large number of PhD and MS students, Avizienis established at UCLA one of the first graduate courses in the USA dedicated to computer arithmetic algorithms and processors. He developed extensive notes and contributed a widely-known unified algorithmic specification [9].

Avizienis has been an active participant in the IEEE Symposia on Computer Arithmetic (ARITH) since the first workshop in 1969. For his seminal contributions to digital arithmetic he has been honored twice: he was invited as the keynote speaker to the 8th IEEE Symposium on Computer Arithmetic in 1978, Lake Como, Italy, and the proceedings of the 12th IEEE Symposium on Computer Arithmetic in 1995, Bath, England, were dedicated to him.

4. On to Even Bigger Things

Over the decades, Al Avizienis maintained close ties to both the Los Angeles and the international Lithuanian community – inviting visiting Lithuanian scholars to his home and frequently visiting Lithuania. One of his proudest accomplishments was working to build a local Lithuanian Boy Scout Camp in the mountains of Southern California. Due to this, fate intervened again when Lithuania achieved independence. His home town of Kaunas Lithuania was historically the Lithuanian academic center, and Avizienis and others saw the opportunity to re-open the National University of Lithuania, Vytautas Magnus University, previously closed by the Soviets, and establish western-style research and PhD programs. From 1990-1993 he served as the founding rector in re-starting this university.

This is probably his crowning accomplishment that will have a major impact on the development of Lithuania, and on the careers of many students. Starting with 180 first-year students, VMU currently has an enrolment of 7000, including about 800 Master's and 200 Doctorate students. After retiring as rector, Prof. Avizienis has served at VMU as a Research Professor and Professor Honoris Causa since 1994, working on fundamental concepts of dependable computing and on an Immune System Paradigm for design of fault-tolerant systems. He has served as a member of the Kaunas city council and will doubtlessly remain involved in community service in other ways in the future.

In his long career, Professor Avizienis has repeatedly demonstrated the ability to seize opportunities, offer innovative solutions to the new situations that present themselves, and in the process enrich the tech-

nology and the lives of those with whom he works. We will be expecting more new and interesting results from his efforts.

References

- [1] Atkins, D. E., Design of the Arithmetic Units of ILLIAC III: Use of Redundancy and Higher Radix Methods", *IEEE Transactions on Computers*, C-19(8):720-733, August 1970.
- [2] Avizienis, A., *A Study of Redundant Number Representations for Parallel Digital Computers*, Ph.D. Thesis, University of Illinois, Urbana, 1960.
- [3] Avizienis, A., Signed Digit Number Representations for Fast Parallel Arithmetic, *IRE Transactions on Electronic Computers*, EC-10(9): 389-400, September 1961.
- [4] Avizienis, A., A Study of the Effectiveness of Fault-Detecting Codes for Binary Arithmetic, Jet Propulsion Laboratory, Pasadena, California, Technical Report 32-711, September 1, 1965.
- [5] Avizienis, A., On flexible implementation of digital computer arithmetic, *Proc. IFIP Congress*, pp. 664-670, 1962.
- [6] Avizienis, A., Arithmetic microsystems for the synthesis of function generators, *Proc. of the IEEE*, 54(12):1910-1919, December 1966.
- [7] Avizienis, A., Binary-compatible signed-digit arithmetic, *Proc. Fall Joint Computer Conference*, pp. 663-672, 1964.
- [8] Avizienis, A. and C. Tung, A Universal Arithmetic Building Element (ABE) and Design Methods for Arithmetic Processors, *IEEE Transactions on Computers*, C-19(8):733-745, August 1970.
- [9] Avizienis, A., Digital Computer Arithmetic: A Unified Algorithmic Specification, *Proceedings Symposium on Computers and Automata*, pp. 509-525, April 1971.
- [10] Avizienis, A., Gilley, G. C., Mathur, F. P., Rennels, D. A., Rohr, J. A., Rubin, D. K., The STAR (Self-Testing-and-Repairing)Computer: An Investigation of the Theory and Practice of Fault-Tolerant Computer Design, *IEEE Trans. on Computers*, Vol. C-20, No. II, November 1971, pp. 1312-1321; also in *Digest of the 1971 International Symposium on Fault-Tolerant Computing*, Pasadena, CA, March 1971, pp. 92-96.
- [11] Avizienis, A., Arithmetic Error Codes: Cost and Effectiveness Studies for Application in Digital System Design, *IEEE Trans. on Computers*, Vol. C-20, No. II, November 1971, pp. 1322-1330; also in *Digest of the 1971 International Symposium on Fault-Tolerant Computing*, Pasadena, CA, March 1971, pp. 118-121.
- [12] Avizienis, A., Fault-Tolerant Computing An Overview, *IEEE Computer*, Vol. 4, No. I, February 1971, pp. 5-8.
- [13] Avizienis, A., Arithmetic Error Codes: Cost and Effectiveness Studies for Application in Digital System Design, *IEEE Transactions on Computers*, C-20(11):1322-1331, November 1971.
- [14] Avizienis, A., Rennels, D. A., Fault-Tolerance Experiments With The JPL STAR Computer, in *Digest of COMPCON '72* (Sixth Annual IEEE Computer Society Int. Conf.), San Francisco, California, 1972, pp. 321-324.
- [15] Avizienis, A., Arithmetic Algorithms for Error-Coded Operands, *IEEE Trans. on Computers*, C-22(6):567-572, June 1973; also in *Digest of FTCS-2, the 2nd*

- International Symposium on Fault-Tolerant Computing*, Newton, MA, June 1972, pp. 25-29.
- [16] Avizienis, A., Parhami, B., A Fault-Tolerant Parallel Computer System for Signal Processing, *Digest of FTCS-4, the 4th International Symposium on Fault-Tolerant Computing*, Champaign, IL., June 1974, pp. 2-8 - 2-13.
- [17] A. Avizienis, Redundancy in number representations as an aspect of computational complexity of arithmetic functions, *Proc. 3rd IEEE Symposium on Computer Arithmetic*, pp. 87-89, 1975.
- [18] Avizienis, A., Ercegovic, M., Lang, T., Sylvain, P., Thomasian, A., An Investigation of Fault-Tolerant Architectures for Large-Scale Numerical Computing, *Proceedings of the Symposium on High-Speed Computer & Algorithm Organization*, University of Illinois at Urbana-Champaign, April 1977, Academic Press, pp. 173-178.
- [19] Avizienis, A., Chen, L., On the Implementation of N-version Programming for Software Fault Tolerance During Execution, *Proceedings COMPSAC 77*, (First IEEE-CS International Computer Software and Applications Conference), Chicago, November 1977, pp. 149-155.
- [20] Avizienis, A., Bond, J. W. III, Fault Tolerance in Large Computing Systems, *Proceedings of the 3rd Jerusalem Conference on Information Technology*, Jerusalem, August 1978, pp. 9-16.
- [21] A. Avizienis, Low-cost residue and inverse residue error-detecting codes for signed-digit arithmetic, *Proc. 5th IEEE Symposium on Computer Arithmetic*, pp. 165-168, 1981.
- [22] A. Avizienis and C. S. Raghavendra, Applications for arithmetic error codes in large, high-performance computers, *Proc. 6th IEEE Symposium on Computer Arithmetic*, pp. 169-173, 1983.
- [23] Avizienis, A., Kelly, J. P. J., Fault Tolerance by Design Diversity: Concepts and Experiments, *Computer*, Vol. 17, No. 8, August 1984, pp. 67-80.
- [24] Avizienis, A., Arithmetic algorithms for operands encoded in two-dimensional low-cost arithmetic codes, *Proc. 7th IEEE Symposium on Computer Arithmetic*, pp. 285-292, 1985.
- [25] Avizienis, A., Gunningberg, P., Kelly, J. P. J., Strigini, L., Traverse, P. J., Tso, K. S., Voges, U., The UCLA DEDIX system: a Distributed Testbed for Multiple-Version Software, *Digest of FTCS-15, the 15th International Symposium on Fault-Tolerant Computing*, Ann Arbor, Michigan, June 1985, pp. 126-134.
- [26] Avizienis, A., Two-Dimensional Low-Cost Arithmetic Residue Codes: Effectiveness and Arithmetic Algorithms, *Digest of FTCS-16, the 16th International Symposium on Fault-Tolerant Computing*, Vienna, Austria, July 1986, pp. 330-336.
- [27] Avizienis, A., Ball, D. E., On the Achievement of a Highly Dependable and Fault-Tolerant Air Traffic Control System, *Computer*, Vol. 20, No. 2, February 1987, pp. 84-90.
- [28] Avizienis, A., A FaultTolerance Infrastructure for Dependable Computing with High-Performance Components, *Digest International Conference on Dependable Systems and Networks*, DSN 2000, New York, June 25-28, 2000, pp. 81-86.
- [29] Baqai, I., Lang, T., Reliability Aspects of the ILLIAC IV Computer, *Proceedings of the 1976 International Conference on Parallel Processing*, August 1976, pp. 123-131.

- [30] Cardenas, A. F., Alavian, F., Avizienis, A., Performance of Recovery Architectures in Parallel Associative Database Processors, *ACM Transactions on Database Systems*, Vol. 8, No. 3, September 1983, pp. 291-323.
- [31] Ercegovac, M.D. and T. Lang, *Digital Arithmetic*, Morgan Kaufmann Publishers, San Francisco, 2004.
- [32] Gilley, G. C., A Fault-Tolerant Spacecraft, *Digest of FTCS-2, the 2nd International Symposium on Fault-Tolerant Computing*, Newton, MA, June 1972, pp. 105-109.
- [33] Gorji-Sinaki, A., *Error-Coded Algorithms for On-Line Arithmetic*, PhD Thesis, University of California, Los Angeles.
- [34] Grnarov, A., Kleinrock, L., Gerla, M., A Highly Reliable Distributed Loop Architecture, *Digest of FTCS-10, the 10th International Symposium on Fault-Tolerant Computing*, Kyoto, Japan, October 1980, pp. 319-324.
- [35] Grnarov, A., Arlat, J., Avizienis, A., Modeling of Software Fault-Tolerance Strategies, *Proceedings of the 11th Annual Pittsburgh Modeling & Simulation Conference*, University of Pittsburgh, Pennsylvania, Vol. II, Part 2, May 1980, pp. 571- 578.
- [36] Kelly, J. P. J., Avizienis. A., A Specification-oriented Multi-version Software Experiment, *Digest of FTCS-13, the 13th International Symposium on Fault-Tolerant Computing*, Milano, Italy, June 1983, pp. 120-126.
- [37] Kelly, J. P.J. et al., A Large Scale Second Generation Experiment in Multi-Version Software: Description and Early Results, *Digest of FTCS 28, International Symposium on Fault-Tolerant Computing*, Tokyo, Japan, June 27-30, pp. 9-14.
- [38] Korff, P., *A Multiaccess Memory*, Ph.D. thesis, UCLA Computer Science Department, University of California, Los Angeles, June 1976; also Technical Report UCLA-ENG-7607, July 1976.
- [39] Mangir, T. E., Avizienis, A., Fault-Tolerant Design for VLSI: Effect of Interconnect Requirements on Yield Improvement of VLSI Designs, *IEEE Transactions on Computers*, Vol. C-31, No. 7, July 1982, pp. 609-616.
- [40] Mathur, F. P., On Reliability Modeling and Analysis of Ultra-Reliable Fault-Tolerant Digital Systems, *IEEE Transactions on Computers*, Vol. C-20, No. II, November 1971, pp. 1376-1382.
- [41] Ng, Y. W., Avizienis, A., A Unified Reliability Model for Fault-Tolerant Computers, *IEEE Transactions on Computers*, Vol. C-29, No. II, pp. 1002-1011, November 1980.
- [42] Oklobdzija, V. G., Ercegovac, M. D., Testability Enhancement of VLSI Using Circuit Structures, *Proceedings of IEEE 1982 International Conference on Circuits and Computers*, New York, 1982.
- [43] Parhami, B. and A. Avizienis, Application of arithmetic error codes for checking mass memories, *Digest of FTCS-3, the 3rd Int. Symposium on Fault-Tolerant Computing*, pp. 47-51, June 1973.
- [44] Parhami, B. and A. Avizienis, Detection of storage errors in mass memories using low-cost arithmetic codes, *IEEE Transactions on Computers*, C-27(4):302-308, April 1978.

- [45] Raghavendra, C. S., Avizienis, A., Ercegovac, M. D., Fault Tolerance in Binary Tree Architectures, *IEEE Transactions on Computers*, Vol. C-33, No. 6, June 1984, pp. 568-572.
- [46] Rennels, D. A., Avizienis, A., RMS: A Reliability Modeling System for Self-Repairing Computers, *Digest of FTCS-3, the 3rd International Symposium on Fault-Tolerant Computing*, June 1973, pp. 131-135.
- [47] Rennels, D. A., Avizienis, A., Ercegovac, M., A Study of Standard Building Blocks for the Design of Fault-Tolerant Distributed Computer Systems, *Digest of FTCS-8, the 8th International Symposium on Fault-Tolerant Computing*, Toulouse, France, June 1978, pp. 144-149.
- [48] J. E. Robertson, A New Class of Digital Division Methods, *IRE Transactions Electronic Computers*, EC-7(3):88-92, September 1958.
- [49] Robertson, J. E., *Introduction to Digital Computer Arithmetic*, Department of Computer Science, University of Illinois at Urbana-Champaign, File No. 599, 1964.
- [50] Robertson, J. E., Arithmetic unit (Chapter 8) in On the design of very high-speed computer, Computer Science Department, University of Illinois at Urbana-Champaign, Technical Report No. 80, 1957
- [51] Rohr, J. A., STAREX Self-Repair Routines: Software Recovery in The JPL-STAR Computer, *Digest of FTCS-3, the 3rd International Symposium on Fault-Tolerant Computing*, Palo Alto, California, June 1973, pp. 11-16.
- [52] Sievers, M., Avizienis, A., Analysis of a Class of Totally Self-Checking Functions Implemented in a MOS LSI General Logic Structure, *Digest of FTCS-11, the 11th International Symposium on Fault-Tolerant Computing*, Portland, Maine, June 1981, pp. 256-261.
- [53] Sleky, A. G. and A. Avizienis, A modified bi-imaginary number system, *Proc. 4th IEEE Symposium on Computer Arithmetic*, pp. 48-55, 1978.
- [54] Sylvain, P., Vineberg, M., The Design and Evaluation of the Array Machine: A High-Level Language Processor, *Proceedings of Second Annual Symposium on Computer Architecture*, Houston, TX, January 1975, pp. 119-125.
- [55] Tung, C. and Avizienis, A., Combinational arithmetic systems for the approximation of functions, *AFIPS Conference Proceedings 1970 Spring Joint Computer Conference*, pp. 95-107, 1970.
- [56] Vineberg, M., Avizienis, A., Implementation of a Higher- Level Language on an Array Machine, *Proceedings of the International Workshop on Computer Architecture*, Grenoble, France, June 1973.
- [57] Wang, S. L., Avizienis, A., The Design of Totally Self-Checking Circuits Using Programmable Logic Arrays, *Digest of FTCS-9, the 9th International Symposium on Fault-Tolerant Computing*, Madison, WI, June 1979, pp. 173-180.