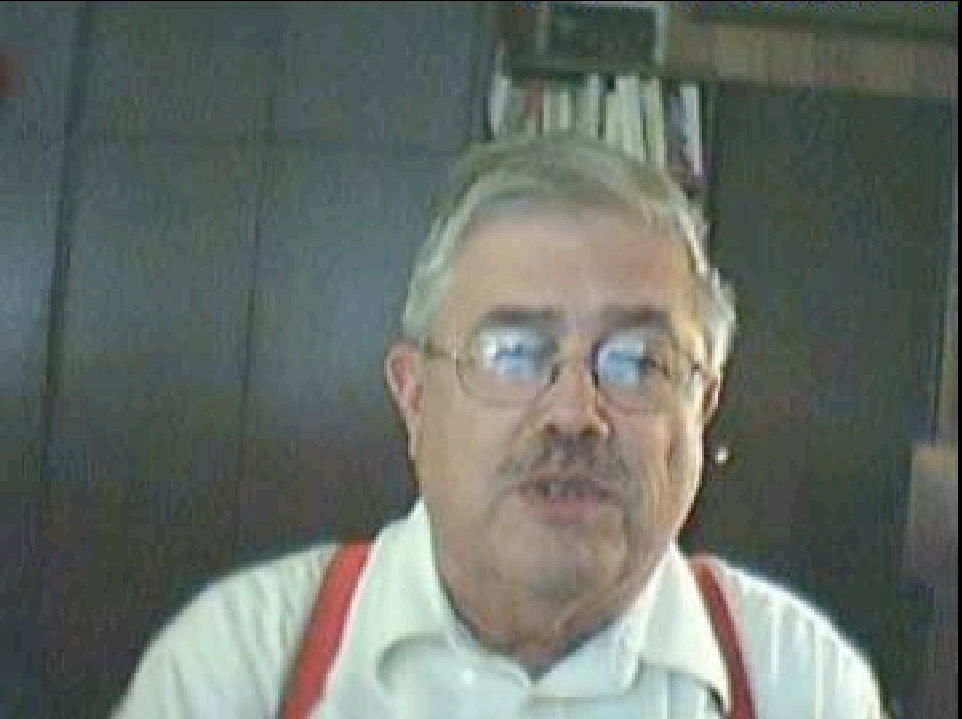
FROM THE UNIVERSITY OF ILLINOIS VIA JPL AND UCLA TO VYTAUTAS MAGNUS UNIVERSITY: 50 YEARS OF COMPUTER ENGINEERING BY ALGIRDAS AVIZIENIS



David A. Rennels[1]
and Milos D. Ercegovac

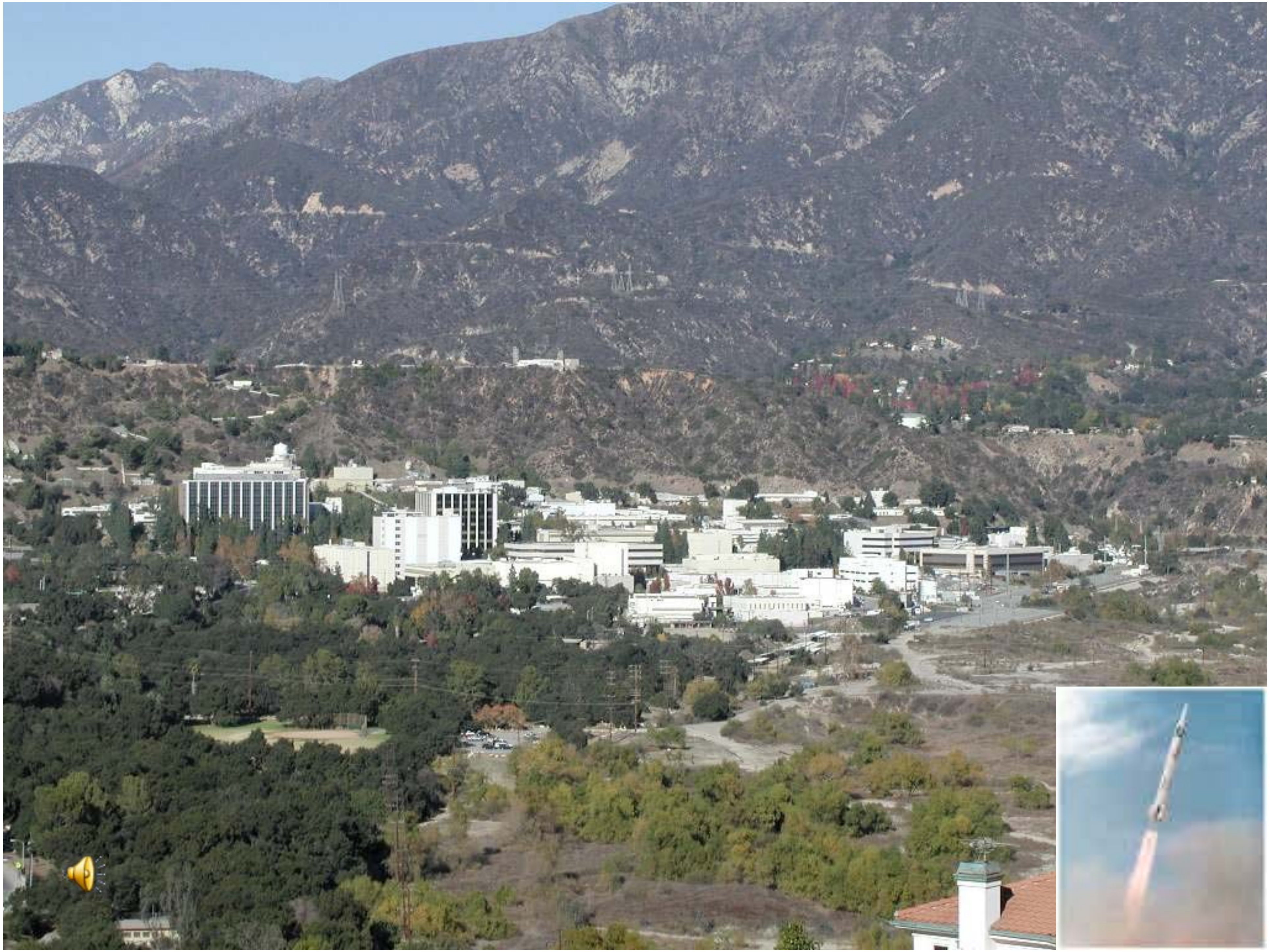
Computer Science Department
University of California, Los Angeles

[1] Also with the Jet Propulsion Laboratory Pasadena, California

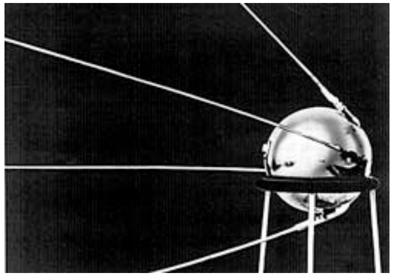
Kaunas

Dresden

Chicago



While Al was finishing his PhD



1. Sputnik was Launched

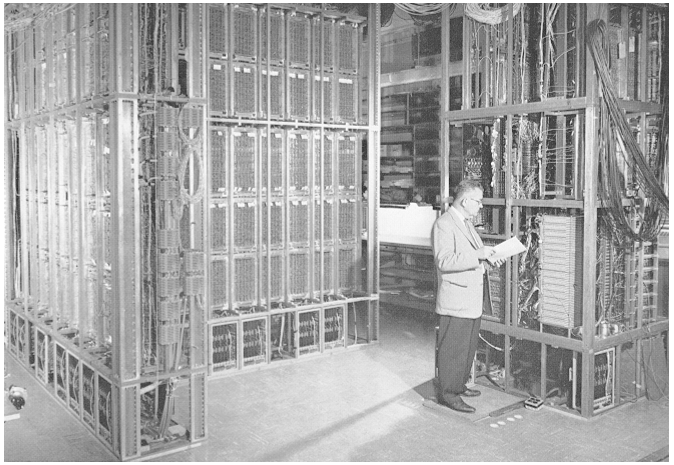
2. JPL and Huntsville Army Labs Launched Explorer

3. NASA was established

4. JPL became a NASA Lab with responsibility For Planetary Exploration

5. Long-life, fault tolerant computers were needed

Fortuna



6. Al learned computer design working on the Illiac computer at UIUC

7. He became an expert in computer arithmetic working under Jim Robertson.

This was just the kind of knowledge needed to develop the JPL STAR Computer

The JPL-Self-Testing and Repairing (STAR) Computer

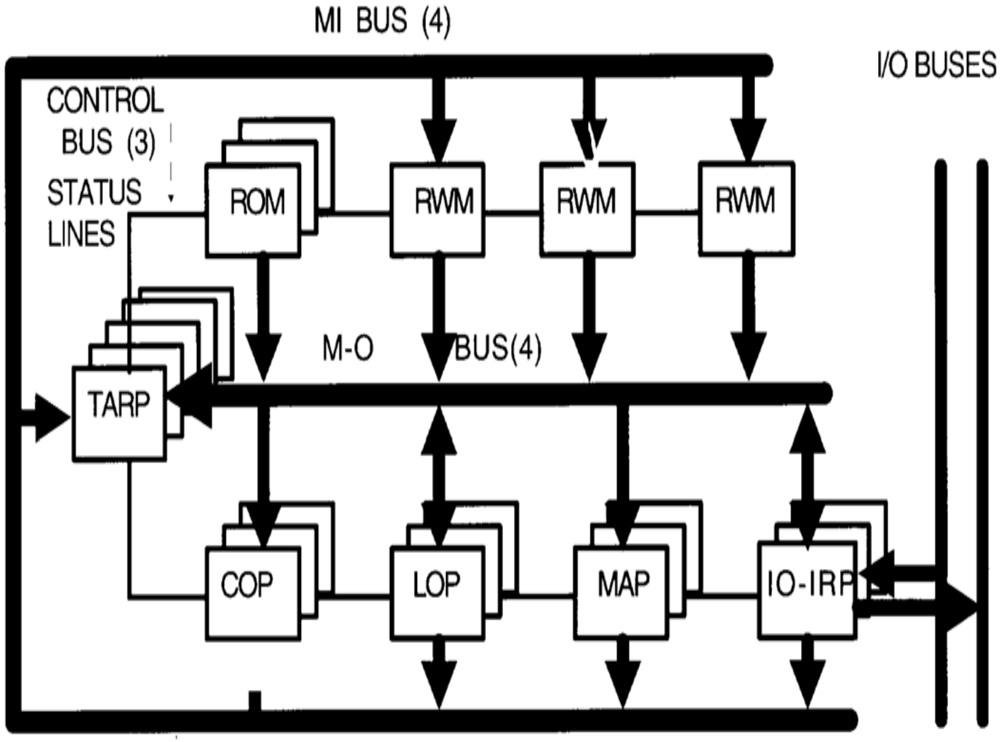


Figure 2: The JPL STAR Computer

RWM - Read Write Memories (Core)
COP - Control Processor – (for addressing – residue codes)
MAP - Main Arithmetic Processor (residue/arithmetic codes)
LOP - Logic Processor (Duplicated for Logic Operations)
IO-IRP – I/O and Interrupt Processor
TARP - Test and Repair Processor (Simple, 3-voted with spares)

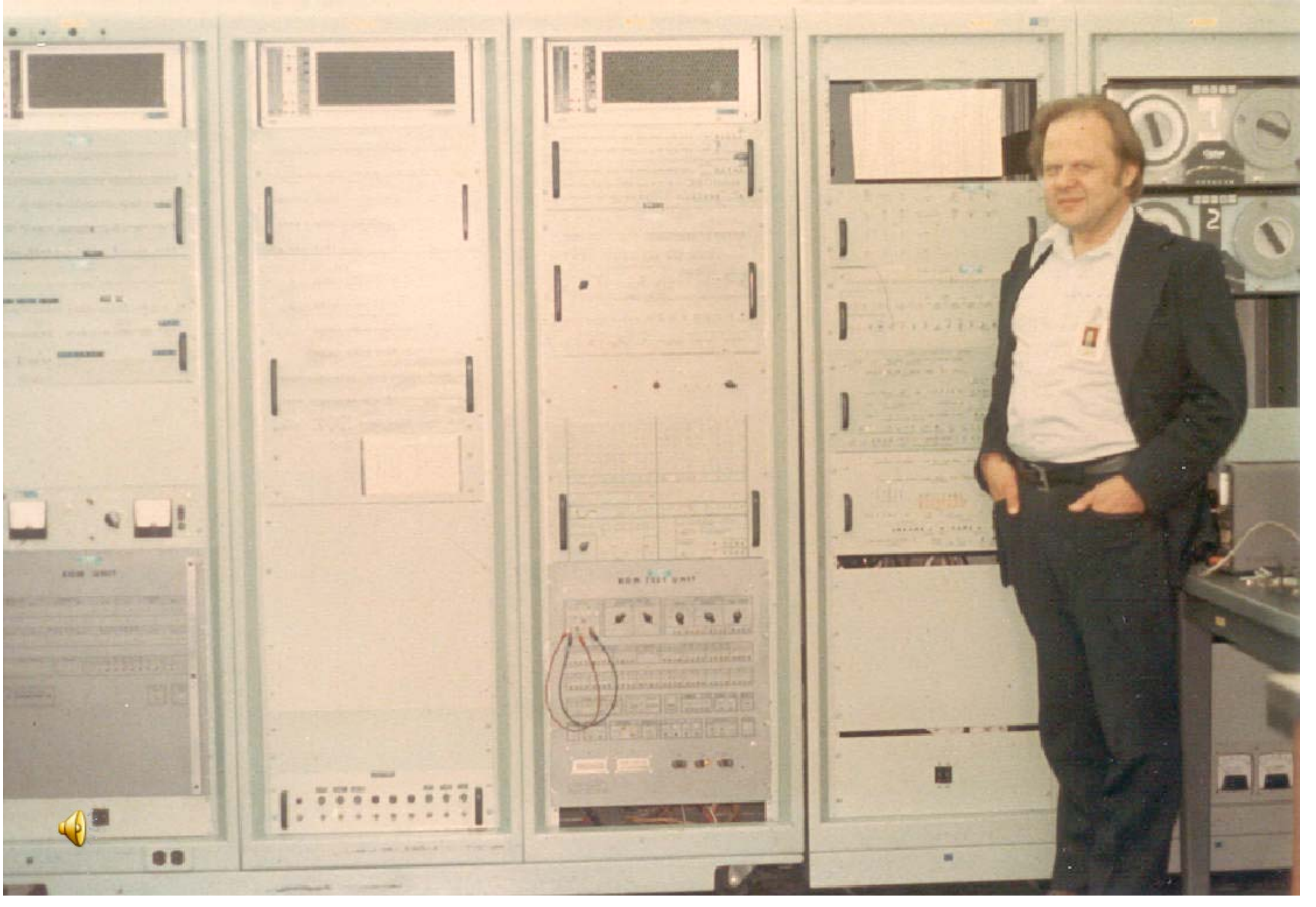
- Low level of Integration required Processor and Memory to be Divided into modules to obtain adequately low failure rates

- To keep power as low as possible
  i) 4-bit serial design
  ii) single copies of modules were powered with coding used to detect errors.
  iii) Coded information was transferred over the buses, with bus checking done by TARP

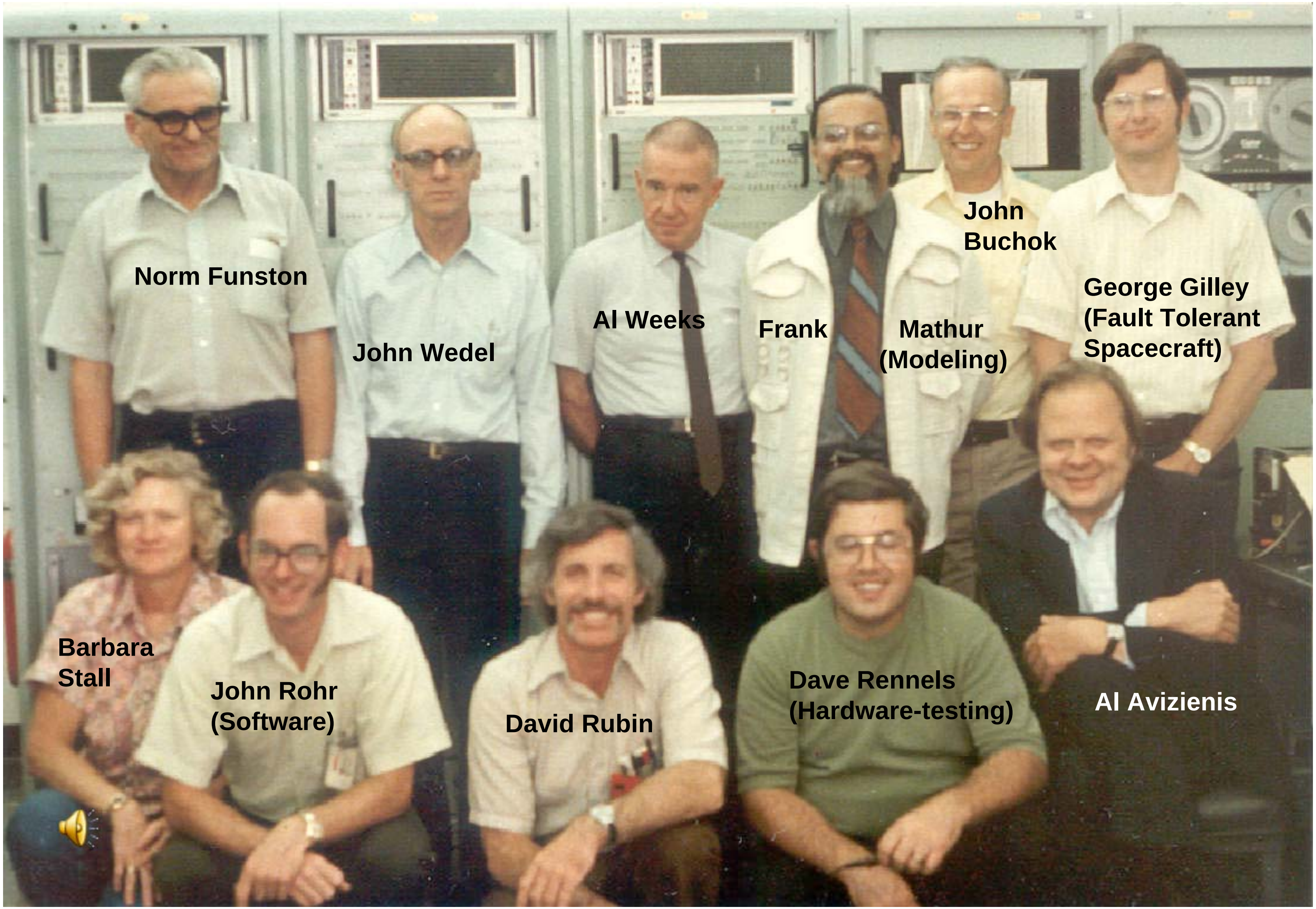
- Duplication with comparison used in a few cases (COP) where coding not applicable (logical operations)

- Hybrid redundant TARP (Test and Repair Proc.) effected recovery from errors:
  - program rollback for transient error recovery
  - replacement of modules as necessary for permanent fault recovery

Algirdas Avizienis and his STAR Computer (circa 1972)



STAR Group Picture (with PhD topics of members who were Al's grad students)



Implementing the STAR Machine opened up many new research issues

- 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**).

Building the Fault-Tolerance Community

In initiating his STAR computer research, Al 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 International) to design a voting power switch for the STAR computer, and the **MIT Instrumentation Laboratory** (now CS Draper Lab) for a read-only memory.

Thu,s 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*.

Founding the IEEE TC and FTCS

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 Al 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.

On to UCLA

As early as 1962, Avizienis began teaching classes in computer design and computer arithmetic at UCLA

By the end of the STAR computer project at JPL, his primary focus shifted to UCLA (although he remained at JPL until ~1980 on a part time basis).

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

Al's work at JPL and in the DC-FTS Lab has resulted in about **200 publications**, **31 PhD dissertations** and **20 M.S**.
Five of the PhD's have gone on to university faculty positions and the rest to responsible positions in government and industry.

The JPL Office

The Dependable Computing and Fault-Tolerant Systems (DC-FTS) Laboratory

Cooperative research with many of the UCLA Faculty in a wide variety of projects:

- **High Performance Numerical Processing** (with Milos Ercegovac and Tomas 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 Milos 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 Milos Ercegovac) [47].
- **Formal specifications and program correctness** (with David Martin) – compiler correctness and the syntheses of correct microprograms (2 PhDs)

The Dependable Computing and Fault-Tolerant Systems (DC-FTS) Laboratory

Among Avizienis' individual projects were:

(1) He 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.

(2) 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 (Tulin Mangir) and to implement self-checking programmable logic arrays (M. Sievers, S. Wang).

(3) 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. The **ARIES 76** reliability modeling system (written in APL) contained all these features and found wide acceptance for education, research, and in industry.

Design Diversity and Multi Version Programming 1980 - 1990

Having established a major record of accomplishment in hardware fault-tolerance, Avizienis decided to tackle the difficult open ended problems of **ultra reliable systems and fault-tolerance in software**.

He recognized before many of us, that most of the basic techniques for hardware fault-tolerance were understood and that design errors (especially software errors) would be crucial in achieving future dependability.

His approach was **Multi-Version programming** – writing and running at least three independent copies of code and voting the results to correct errors in individual versions (and running versions on different hardware if there was concern about hardware design errors.)

The approach is akin to replication and voting in hardware, *i.e.*, **Multi-Version Software (MVS)**, is 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.

N-Version Programming Experiments

- An initial experiment was conducted in the 1970's using students in a software engineering class, and the results were satisfactory: **faulty versions were outvoted and the redundant software continued to perform satisfactorily**.

- 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.

- Experiments were conducted writing **specifications in different languages** to maximize to learn which languages were least ambiguous and thus resulted in more error free code by real programmers. (PDL beat formal languages)

N-Version Programming Experiments (cont'd)

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.

In addition to Avizienis and John Kelly, the design team was a remarkably 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.

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.

Contributions of Design Diversity and N-Version Programming Experiments to the *science* of dependable computing.

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 this work 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 engineering. 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**.

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 and Alain Costes were the primary movers in establishing this **IFIP Working Group** in October 1980 and Al 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.

Some of the important planning for this was done in 1981

(Al, Bill Carter, John Meyer)

Contributions to Digital Arithmetic

In his landmark 1960 doctoral dissertation on **signed-digit number systems and arithmetic algorithms**, Avizenis provided the foundation and formalisms enabling their systematic use design of arithmetic units.

He 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** 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**, *i.e.*, independently of the number of digits. As a consequence, one can also perform addition and other operations in the most-significant-digit-first (MSDF) manner which ultimately led to online arithmetic where all operations can be performed MSDF, generating the result digits while consuming the input digits.

Avizienis' work on redundant representation systems provided also a framework for studying other important techniques such as digit-set recoding, carry-save addition, and, as mentioned, online arithmetic. He developed **arithmetic microsystems and universal arithmetic building element (ABE)** suitable for IC implementations based on signed-digit arithmetic.

Contributions to Digital Arithmetic

The use of redundant number systems has become pervasive and represents one of the most important developments in the field of digital arithmetic. The first extensive use of signed-digit number system and algorithms was in the **Illiac III computer**. A quick look at the literature indicates that **signed-digit number system has been frequently used in both the general-purpose processors and in application-specific processors**, in particular, in digital signal processors.

Avizenis is also widely known for his original contributions to **low-cost arithmetic error-detection correction codes** where he developed efficient algorithms for error-coded operands. These algorithms were implemented in the radix-16 processor constructed for the **Jet Propulsion Laboratory STAR computer**. 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 Siewiorek and Swarz. 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.

Contributions to Digital Arithmetic

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.

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**,
- the proceedings of the **12th IEEE Symposium on Computer Arithmetic in 1995, Bath, England, were dedicated to him**.

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.

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.

Fortuna

Gaudeamus igitur

Vivat academia	*Long live the academy!*	**Vivant omnes virgines**	*Long live all maidens*
Vivant professores	*Long live the teachers!*	**Faciles, formosae.**	*Easy and beautiful!*
Vivat membrum quodlibet	*Long live each male student!*	**Vivant et mulieres**	*Long live mature women also,*
Vivat membra quaelibet	*Long live each female student!*	**Tenerae amabiles**	*Tender and loveable*
Semper sint in flore.	*May they always flourish!*	**Bonae laboriosae.**	*And full of good labor.*

Conclusion

Restarting **Vitautas Magnus University** is probably Al Avizienis 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 Doctoral students.

After retiring as rector, he 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 technology and the lives of those with whom he works. We will be expecting more new and interesting results from his efforts.