**TUTORIAL NO. 1**

**VLSI from a User’s Perspective**

Or how to utilize VLSI without blowing hundreds of thousands of dollars and several years

**Instructor:** Rex Rice

**Audience:** Current and prospective VLSI users/planners/workers/watchers, including but not limited to managers, system planners, system architects, logic designers, design automation specialists, silicon designers, packaging engineers, and test engineers.

**Course Description:** This tutorial will design to provide a broad interdisciplinary perspective on the present and potential use of VLSI. It will emphasize economic considerations rather than cover detail processes. It will include a historical perspective and present examples of current VLSI programs. The main portion of the tutorial will trace a VLSI design through the complete process starting with system design and ending with a tested computer. Available alternatives and economic considerations will be discussed for each major step. Particular emphasis will be placed on hazards to be avoided.

**Rex Rice** is an independent consulting engineer specializing in digital systems, and has been actively engaged in the computer field since 1949. He has published over 50 papers and holds more than 20 computer patents. While at IBM he managed the development of the first transistorized computer.

He is best known for his work at Fairchild: the invention of the Dual In-Line Package (DIP), managing the first large silicon memory program, the ILLIAC IV main memory; and for his early perception of work in bus-oriented, functionally split array logic systems. The SYMBOL computer which he conceived and the program he managed (1966–70) to develop it have past sophisticated direct implementation of a high-level language in hardware presently in existence. Many of its features are just now appearing in new systems.

Rice, who received his BA in mechanical engineering at Stanford in 1940, has taught graduate computer courses at USC. He is an IEEE Fellow, a member of the Computer Society Governing Board, and founder of the COMPCONS.

**Course Outline:**

- **Historical Review and Attempted Forecast:** Steps from transistors to VLSI will be traced. Emphasis will be placed on economic motivations and changing job skills for workers in the field.
- **Current Examples:** One university and two commercial examples of successful VLSI programs will be reviewed. Two interesting but less successful projects will be covered for contrast.
- **Description of VLSI Design, Fabrication, and Testing Process:** Each step in the process will be discussed covering critical parameters and required decisions: choice of circuit/silicon technology; system architecture/logic; simulation; design automation; marketing, fabrication, and assembly; and testing.
- **Hazards Frequently Encountered**
- **Economic Review and Wrap-Up**

---

**TUTORIAL NO. 2**

**Microcomputer System Design and Techniques**

**Instructor:** Carol A. Ogdin

**Audience:** Computer programmers, systems analysts, and managers who have experience with large-scale computers, minicomputers, operating systems, and programming languages.

**Course Description:** Many of the techniques essential to the application of microcomputers are familiar to programmers and analysts, and some are vastly different. Writing software that will be permanently fixed in read-only memory, and in hundreds or thousands of copies, makes a difference in the strategies best used.

This one-day tutorial takes advantage of the participant’s experience with other kinds of computing systems to discuss the technology of micros. The significant differences from traditional computer system design are carefully explained. Some of these differences include: the knowledge of electronic circuitry, the subtle effects of software engineering discipline, and the notion that software is always more expensive than hardware!

The active participant will emerge from this tutorial adequately prepared to selectively read the technical literature and identify the appropriate application areas where micros excel.

**Carol Anne Ogdin** is technical director of Software Technique, Alexandria, Virginia. She is especially active in applying the latest software engineering and digital design strategies to the implementation of special-purpose computers. She is an acknowledged authority on micros and is in frequent demand as a consultant, lecturer, and author. Ogdin is consulting editor to both Mini-Micro Systems and EDN magazines, has three books in print on microcomputer design and management, and is an ACM National Lecturer.

**Course Outline:**

- **Evolution of the Micro:** evolution of computers and semiconductors • major micros • the 8- and 16-bit controversy • market sizes and technology trends
- **Microprocessor and Microcomputer Architecture:** the new jargon • basic components • system selection • applications — for one-chip micros, for CPUs, and for boards
- **Hardware/Software Tradeoffs:** buses and modules • memory technology • I/O ports and interfaces • systems on silicon • programming languages and techniques • operating systems
- **Management and Tools:** development systems • coordinating the hardware and software • project planning

---

**TUTORIAL NO. 3**

**Microprocessor Interfacing Techniques**

**Instructor:** G. Jack Lipovski

**Audience:** Microcomputer system designers, engineers who interconnect microcomputers with external equipment, programmers who write software for interface hardware. Some experience in logic design and in assembler language on some machine is required.

**Course Description:** This tutorial emphasizes the principles of interface design and shows applications of interface design using the Motorola 6809 microprocessor and some compatible Motorola micros. Input-output chips. Topics covered include: (1) good programming practice for interface programs, subroutines, data structures and arithmetic, and the design of buses and address decoders; (2) the hardware design of software interfaces; (3) simple parallel I/O devices and interrupts; (4) a survey of conventional analog circuits used in microprocessors and discussion of frequency modulation analog circuitry; and (4) communications circuitry and programming.

**G. Jack Lipovski** is an associate professor of electrical engineering and computer science at the University of Texas at Austin, where he teaches microcomputer interfacing and computer architecture. While at the University of Texas, he designed the Texas Reconfigurable Array Computer. Earlier, at the University of Florida, he designed the CASSM data base computer. Lipovski is a past chairman of the Technical Committee on Computer Architecture of the Computer Society and ACM SIGARCH. He is a member of the Computer Society Governing Board and a Director of Euromicro.

**Course Outline:**

- **Introduction:** the M6809 instruction set • data structures • subroutines • computer arithmetic • bus timing • address decoding
- **Parallel I/O:** simple I/O • a traffic light controller • M6821 (PIA) • I/O • an IC tester • real-time and gradually synchronization interrupts
- **Analog Devices:** transducers • OP amps • hardware analog/digital and software tradeoffs • D-to-A converters • A-to-D converters • counters • M6840 • period measurement
- **Communications Devices:** levels of protocol • RS232 standard • modems • the UART • the M6850 (ACIA) • SDLC and IEEE-488 protocols
Interactive Computer Graphics

Instructor: Herbert Freeman

Audience: Programmers, engineers, designers, architects, and others who use or intend to use computer-generated graphics in their work, especially those for whom this is a new or recently entered area of activity.

Course Description: This tutorial presents an overview of the hardware, software, and application areas of interactive computer graphics. Major topics covered include graphics terminals, the pros and cons of different display devices, the architecture of a typical graphics processor, algorithms for line, character, and curve generation, graphics data structures, graphics languages and design languages, hidden-line and hidden-surface removal, vector and raster display systems, generation of shaded (tone) and color images, 3-D object modeling, and applications of computer graphics in engineering, science, architecture, cartography, and the graphics arts.

Herbert Freeman is professor of computer engineering and director of the Image Processing Laboratory at Rensselaer Polytechnic Institute, Troy, New York, where he is engaged in teaching and research in computer graphics and digital image processing. He has been in the field of computer graphics since the early 1960s, and is editor of the journal Computer Graphics and Image Processing. Prior to entering upon his academic career he spent 12 years in industry. He is the author of over 60 technical papers and the author/editor of three books. He is the past chairman of the IEEE Computer Society's Technical Committee on Machine Intelligence and Pattern Analysis, and is currently president of the International Association for Pattern Recognition.

Course Outline:

Introduction: definition and scope of interactive computer graphics • capabilities and limitations • definition of terms • historical summary • areas of application • related computer disciplines

Graphics Systems and Devices: display devices and interactive controls, characteristics and tradeoffs • graphics terminals, processor architecture, and hardware algorithms for line, character, and curve generation • cost/benefit tradeoffs • vector and raster systems, transformations, clipping, and scan conversion

Graphics Software: data structures • display handling procedures • low-level graphics languages • design languages • graphics systems

Interactive Graphics for 3-D Design: 3-D transformations and perspective • hidden-line and hidden-surface removal • 3-D object modeling • free-form surface representation • shading and half-toning

Applications

Future Trends

Computer-Assisted Design and Engineering

Instructor: Stephen R. Levine

Audience: Managers, users, and persons responsible for specifying computer-assisted design and engineering systems. Professionals who will use these systems need no background in computer science.

Course Description: This introduction to the use of computer and computer graphics in design and engineering describes the hardware and software of current and future systems. Other topics covered include system requirements, cost tradeoffs, system integration, facility management, and user training. Expectations for the near future will be presented.

Stephen R. Levine is president of Electronic Graphics Associates, a firm specializing in computer graphics consulting and systems integration. He also teaches an international short course on computer graphics, and has consulted widely on the subject. His previous experience includes eight years with Lawrence Livermore National Laboratory, where he directed the development of new computer graphics hardware and software, including the joint development of the Dicom D148 color film recorder; three years with Stanford Research Institute; and two years with Singer Simulation Products. Levine is a past president of the National Computer Graphics Association, a current director and a former vice chairman and Treasurer of ACM's SIGGRAPH. He received a BS and an MS in electrical engineering from the University of California, Berkeley, and a PhD in computer science from Stanford in 1975.

Course Outline:

Introduction: hardware • software • state of the art

Applications: present • future • assessing needs

Systems: specification • evaluation • cost tradeoffs • utilization • training • managing

Future: near term expectations

An Overview of Networks and Distributed Processing

Instructor: Burt H. Liebowitz

Audience: Engineers and managers who are planning the development of distributed systems. Some background in computer technology would be highly desirable.

Course Description: This tutorial presents an overview of distributed processing, including a general introduction to computer networks. Three major areas of distributed processing are defined: point-of-use systems, resource-sharing networks, and multiprocessor systems. Characteristics, examples, benefits, and tradeoffs are presented for each area.

Technological issues are presented involving processors, communications, intercomputer coupling, executive software structures, system architectures, component selection, and allocation of functions and data files to multiple processors. Several case histories are discussed to provide insight into design issues, cost effectiveness, and management problems.

Burt H. Liebowitz is executive vice president of the International Computing Co. in Bethesda, Maryland. He has 20 years of experience in the computer field, the last five of which have been involved with distributed systems. He was responsible for the design and software implementation of the distributed minicomputer system used for off-track betting in New York City. He has taught at government installations and universities, including George Washington University, where he developed the course on distributed processing. An author of several articles published in leading technical journals, Liebowitz holds MS degrees in mathematics and electrical engineering.

Course Outline:

Computer Technology in Distributed Systems: minis • micros • mainframes • connecting multiple processors

Telecommunications for Geographically Distributed Systems: networks • protocols • commercial offerings

Resource-Sharing Networks: computer communication networks • computer networks

Intelligent Terminals: technology • applications

Point-of-Use Systems

Multiple-Processor Systems: architectures • software • comparisons with large systems • design issues

Multiprocessors

Distributed Data Bases: file-splitting • directory-splitting • design factors • problem areas

Case Histories

Management Issues

Design Issues

Future Trends

All tutorials start at 9 a.m.
TUTORIAL NO. 7

Computer Communications Protocols

Instructor: David C. Wood

Audience: Systems analysts and programmers, as well as technical designers and managers in the communications field. Assumes several years of experience in computing and some exposure to data communications.

Course Description: The objective of this tutorial is to provide an understanding of the protocols used in a computer communications network. Concepts of layering of protocols will be explained with reference to the Open Systems Architecture model, and emerging standards in the public and government arenas will be discussed. The emphasis is on the capabilities of each protocol and its principles of operation. The protocols encountered when interfacing to or using a general-purpose packet switching or local area network will be examined. The functions of each protocol will be defined, and examples drawn from both research and public networks will be explained, identifying their characteristics and limitations.

David C. Wood is an associate department head at The MITRE Corporation in McLean, Virginia. He is responsible for several projects involving computer networking and computer security, including the development of local area networks and their interconnection to long-haul packet networks. He has published numerous articles on packet switching networks, and has taught courses in protocols for The Johns Hopkins University, The George Washington University, and ACM. Wood is a member of the ACM Council and vice-chairman of the ACM Special Interest Group on Data Communication (SIGCOMM). He holds a Ph.D. in mathematics from the University of Manchester, England.

Course Outline:

Introduction: computer network examples, packet switching, local networks, protocols, layering, Open Systems Architecture

Link Control Protocols: functions, bi-level, high-level data link control (HDLC)

Contention Protocols: Aloha, carrier sense multiple access (CSMA), CSMA with collision detection

Packet Switching Interfaces: CCITT Recommendation X.25, features, implementations, 1980 revision

End-to-End Protocols: functions, Department of Defense standard Transmission Control Protocol (TCP) and Internet Protocol (IP)


TUTORIAL NO. 8

Computer Networks

Instructor: Ira W. Cotton

Audience: Managers, engineers, programmers, systems analysts, and students. Some experience in computing would be helpful.

Course Description: This introductory level tutorial addresses computer networks — the interconnection of computer systems by telecommunication facilities. A variety of different types of networks will be considered to illustrate the range of benefits from networking. Network technology will be examined in terms of switching technology, interfaces and protocols, and components. Non-technical issues relating to networks will also be discussed, including the regulatory environment and network management. The seminar concludes with a look at the future of computer networks.

Ira W. Cotton is a senior associate with Booz, Allen and Hamilton, Inc., where he is conducting projects for a variety of clients in the areas of networking systems and office automation. He was formerly manager of the Local Networking and Office Systems Group at the National Bureau of Standards. He has written over 50 technical papers and has presented professional development courses internationally for many years. Cotton holds a master's degree in computer science from the University of Pennsylvania and a doctorate in business administration from The George Washington University. He is a Governor of the International Council on Computer Communications, a member of the ACM, IEEE Computer Society, and Sigma Xi, and serves on the editorial boards of several journals.

Course Outline:

Network Utilization: types of networks, components and configurations, applications, benefits, problems

Basic Communications: review of data communications fundamentals, circuits, modems, and modulation, multiplexing and concentration, interfaces and protocols

Network Technology: functions and components, communication switching, routing, control, network architecture, commercial networks

Network Management: regulation, network costs, evaluation and selection, operation and maintenance, user services

Future Developments: technology trends, new communications facilities, new applications, new opportunities, new problems

TUTORIAL NO. 9

A Pragmatic View of Distributed Processing Systems

Instructor: Harvey A. Freeman

Audience: Managers, engineers, programmers, systems analysts, designers, and students. Some experience in computing and some exposure to data communications would be helpful.

Course Description: This tutorial provides an overview of current concepts and problems associated with implementing a distributed processing system. The elements of a distributed system are defined and contemporary models of such systems are explored. This is followed by a survey of design issues associated with system software, interconnection systems, topology, and hardware concepts. Selected case studies of distributed systems are used to highlight the previously discussed concepts and to indicate the direction of distributed systems. The tutorial concludes with observations on future systems and supporting technologies.

Harvey A. Freeman is manager of the Data System Design Department at Sperry Univac. He currently manages an advanced development group investigating and developing concepts, techniques, and specifications for large-scale Sperry Univac computer systems in distributed processing environments, including local, back-end, and non-homogeneous networks. He is the author of numerous technical papers and co-author of the book, Data Base Computers. Freeman is also an adjunct professor in the computer science department at the University of Minnesota. He holds a Ph.D. in electrical engineering from the University of Illinois.

Course Outline:

Introduction: overview, nomenclature, reasons for distributed processing, networks, applications

Models: virtual circuits, datagrams, standards, hierarchical systems, common issues, system approaches, topology, networking

Design Issues: operating systems, protocol hierarchies, gateways, communication subsystems, contentions, channels, process partitioning

Case Studies: ETHERNET, SNA, selected others

Conclusions: future systems, concluding remarks

All tutorials start at 9 a.m.
### TUTORIAL NO. 10

**Local Computer Networks**

**Instructor:** Harvey A. Freeman  
**Audience:** Managers, engineers, programmers, system analysts, designers, and students. Some experience in computing and some exposure to data communications would be helpful.

**Course Description:** This introductory level tutorial addresses local computer networks, a class of systems which are generally owned by a single organization, cover distances on the order of a few miles, and use a communication subnetwork technology such as packet switching, circuit switching, or busing. First the overall definition and context of LCNs are described. Then important design issues and potential solutions currently identified for LCNs are discussed. This is followed by a number of case studies of important local computer networks. LCNs which evolved from distributed processing contexts and those which evolved from attempts to upgrade and enhance current large-scale systems are covered. Concludes with a view of the future directions of LCNs.

**Harvey A. Freeman** is manager of the Data Systems Design Department at Sperry Univac. He currently manages an advanced development group investigating and developing concepts, techniques, and specifications for large-scale Sperry Univac computer systems in distributed processing environments, including local, back-end, and non-homogeneous networks. He is the author of numerous technical papers and co-author of the book, *Data Base Computers*. Freeman is also an adjunct professor in the computer science department at the University of Minnesota. He holds a PhD in electrical engineering from the University of Illinois.

**Course Outline:**

- **Background:** Definition • Evolution • Taxonomy • Examples  
- **Design Issues:** Network entities • Network architecture • Network operation • Performance • User Considerations  
- **Case Studies:** HYPERSchannel • ETHERNET • Cluster/One Model A • selected others  
- **Future Directions:** Gateways • Commercial availability • Trends  
- **Conclusions**

### TUTORIAL NO. 11

**Models and Metrics for Software Management and Engineering**

**Instructor:** Victor R. Basili  
**Audience:** Software managers and engineers who are looking for quantitative assistance in managing, controlling, estimating, evaluating, and contracting software development and maintenance.

**Course Description:** This tutorial presents a new quantitative approach to software management and software engineering. It focuses on attributes that can be managed quantitatively, covering both product-oriented and process-oriented attributes. A large set of models is utilized with emphasis on those areas where quantitative management can give the greatest payoff.

**Victor R. Basili** is an associate professor at the University of Maryland at College Park. A consultant to several industrial organizations and government agencies, he has been active in the design and development of several software projects and is currently involved in the measurement and evaluation of software development at NASA’s Goddard Space Flight Center. Basili received his PhD in computer science from the University of Texas at Austin in 1970.

**Course Outline:**

- **Introduction:** Quantitative software management • Modeling • Metrics  
- **Resource Models:** Classes of models • Empirical models of programming estimation and measurement • Model validation studies • An empirical meta-model • Theoretical dynamic modeling using the Rayleigh curve • Various uses of resource models  
- **Changes and Errors:** the use of changes and errors in the characterization, estimation, and evaluation of software development • the use of error analysis in improving the software development process • Error models for program evolution and reliability • Distributions by various classification schemes • Metrics based on error and change  
- **Product Measures:** Size • Control and data structure metrics • Cyclical complexity • Software science • Data binding • Spair • Metrics across time • Example uses and effects of software metrics on technology transfer • Quality assurance and contracting  
- **Data Collection:** Establishing the environment • Collecting the data • Data validation analysis and result reporting

### TUTORIAL NO. 12

**Modern Programming Language Design**

**Instructor:** Anthony I. Wasserman  
**Audience:** Programmers and system developers interested in an overview of current programming language design, and for others interested in the issues underlying the recent proliferation of programming languages.

**Course Description:** This tutorial presents issues and goals of recent programming language design efforts. Issues in control structures, data types, programming style, and other key topics are surveyed. Examples from Pascal, Ada, and other modern languages are incorporated, and a framework for understanding and using new programming languages is provided.

**Anthony I. Wasserman** is an associate professor of medical information science at the University of California, San Francisco, and a lecturer in the Computer Science Division at the University of California, Berkeley. His research interests include programming language design, programming methodology, data base management, computer science education, and information processing. He has been active in professional organizations, serving as chairman of ACM SIGSOFT, the special interest group on software engineering, an ACM National Lecturer, and a member of numerous technical program committees. He is the author of over 40 technical papers and co-editor of *Software Engineering Education — Needs and Objectives and Issues in Data Base Management*. Wasserman received his PhD in computer sciences from the University of Wisconsin, Madison, and his AB in mathematics and physics from the University of California, Berkeley.

**Course Outline:**

- **Issues in Programming Language Design:** Programming and problem solving • Goals • Synthesis of language features  
- **Overview of Pascal and Ada:** History and background • General language characteristics  
- **Control Structures — Sequential:** Issues (sequential and concurrent) • The “goto” debate • Iteration • Nondeterministic control • Parameter passing • Recursion  
- **Data Types and Variables:** Declarations • Type checking • Pascal and Ada data types • Abstract data types  
- **Control Structures — Concurrent:** Concept of a process • Concurrency mechanisms  
- **Programming Style:** Relationship to programming languages • Testing and verification issues • Readability of programs

**All tutorials start at 9 a.m.**
**TUTORIAL NO. 13**

**System Analysis and Requirements Specification**

**Instructor:** Raymond T. Yeh

**Audience:** Managers, analysts, and designers.

**Course Description:** This tutorial introduces a structured approach for the systematic analysis and derivation of software requirements. The methodology is based on the systematic construction of a conceptual model of the enterprise using a semantic network notation. An executable specification (or fast prototyping) can then be derived. The methodology is a synthesis of many approaches using a unified notation. Emphasis is on its practical applications with many examples.

Raymond T. Yeh is a professor and chairman of the Computer Science Department at the University of Maryland. He is also the director of the Center for Information Sciences Research. He has written and edited seven books and over 70 scientific articles. His current research interests during the past few years have been in the software methodology and database design areas. Yeh is a consultant to various industrial organizations and government agencies. He gives lectures worldwide.

**Course Outline:**

- The Requirements Problems
- What a Software Requirements Document Should Include: checklist
- Techniques for Contextual Analysis (Bounding the Solution Space): structured interviews, questionnaire design and analysis
- Model Construction: data modeling — data classification and organization; view modeling; view integration — process specification and organization; functional approach to embedded and real-time systems — modeling constraints
- Specifying the Requirements: deriving executable specification for prototyping
- A Review and Comparison with Other Approaches

---

**TUTORIAL NO. 14**

**Software Testing and Validation**

**Instructor:** Edward Miller

**Audience:** Those familiar with basic computer science topics, including elementary knowledge of programming languages and programming/debugging processes and outcomes. The main thrust of the seminar is the design methodology — defining what is and what is not known about software testing and validation technology. Thus, the audience is expected to be reasonably technically mature.

**Course Description:** This tutorial describes current methods of software testing in a framework that permits an attendee to select methods that work. It focuses on the philosophy of testing, the theoretical foundations of testing, tools, and techniques including static and dynamic testing, management and control issues, measurement and planning questions, and research and development directions.

Besides presenting detailed investigation into the motivation and methods of a variety of software testing techniques, the seminar provides in-depth access to technical literature pertaining to testing. Attendees should acquire an increased understanding of the techniques and results of software testing, how it can be employed in a software quality assurance activity, and how it fits into the entire software life cycle.

Edward Miller is technical director of Software Research Associates, San Francisco, California, and is a firm devoted to advanced technology and software application. His interests include software engineering management, software testing technology, automated (tool) technology, hierarchical design and implementation methods, and computer architecture. He was previously director of the Software Technology Center at Science Applications, Inc., in San Francisco, and director of the Program Validation Project at General Research Corporation in Santa Barbara, California.

Miller received a BSEE at Iowa State University in 1962, an MS in applied mathematics at the University of Colorado in 1964, and a PhD at the University of Maryland in 1968. He was editor of ACM's Computer Architecture News (1975 to 1976) and is currently associated technical editor of Computer Magazine.

**Course Outline:**

- Philosophy of Testing: motivating forces — testing principles — overview of methodology — organizing for testing
- Theoretical Foundations: general principles — graph theory — program modeling techniques — elements of reliable test theory — combinatorics of testing
- Tools and Techniques: general principles — static testing methods — dynamic testing methods — automated tools and software testing — systematic methodologies — test data generation
- Planning and Measurement: general principles — test coverage measures — test structuring
- Management and Control: economics of testing — test status monitoring
- Research and Development

---

**TUTORIAL NO. 15**

**Software System Design**

**Instructor:** Jack C. Wileden

**Audience:** Software designers, and project managers.

**Course Description:** This tutorial focuses on techniques applicable to the design activities that occur prior to the actual coding of a software system. A specific, generally applicable technique for design description and analysis is presented. When participants are part of a series of software development using this technique. Alternative approaches to software design are surveyed, and the prospects for automated software development environments are explored.

Jack C. Wileden is an assistant professor of computer and information science at the University of Massachusetts. His teaching experience spans all aspects of software.

His research interests are in the areas of software system design and analysis, with particular emphasis upon the development of techniques to aid in the design and analysis of concurrent software. He has been associated with the DREAM research team since 1976, and is presently involved in projects studying formal semantics of concurrent computation, software development environments, and issues in the development of cooperative distributed processing systems.

Wileden is a member of ACM, the IEEE Computer Society, and Sigma Xi, and is a former ACM National Lecturer. He received his PhD in computer and communication sciences from the University of Michigan in 1978.

**Course Outline:**

- Introduction: role of the design phase in software development
- The Design Phase: modeling of systems using a hierarchical description — data abstractions
- A Design Description Technique: state-based modeling of designs — generic component descriptions — component interaction — hierarchical component relationships
- Related Description Techniques: module interrelationship descriptions — abstract algorithm design — data flow design — concurrent and distributed system design
- A Design Analysis Technique: state model analysis — finite state testing — consistency checking
- Related Analysis Techniques: activity expression analysis — symbolic evaluation
- Software Development Environments: tools — organization

---

**ORLANDO Marriott, INN**

**Tutorial Week @ East 81**

6700 Sand Lake Rd., Orlando, FL 32807, (800) 226-9290

All tutorials start at 9 a.m.
Attendance limited to 100 per tutorial. There will be no registration at the door. Advance registration only: deadline — March 9, 1981. Select up to one tutorial per individual per day. Requests for refunds must be received in writing by March 23, 1981.

Mail to: Tutorial Week East '81, P.O. Box 639, Silver Spring, MD 20901, (301) 589-3386

I wish to register for the following tutorials (circle numbers):

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

☐ Check enclosed  ☐ Visa/BankAmericard  ☐ Master Charge

Card No. _______________ Exp. Date __________ Signature _______________________

Name _____________________ Membership No. ______________

Street Address ____________________ Tel. ________

City/State/Zip ____________

Includes bound tutorial and luncheon. There will be a no-host reception on Monday night.

FEES

<table>
<thead>
<tr>
<th></th>
<th>Per Tutorial</th>
<th>All Week</th>
</tr>
</thead>
<tbody>
<tr>
<td>Members</td>
<td>$85</td>
<td>$400</td>
</tr>
<tr>
<td>Nonmembers</td>
<td>$100</td>
<td>$475</td>
</tr>
</tbody>
</table>

Room Reservation Request

IEEE COMPUTER SOCIETY, April 6–10, 1981

Name ____________________________

Address __________________________

City/State/Zip _____________________

Arrival Date ___________ Time ______

Departure Date ________________

Please check room preference:

☐ single or double ($50.00)
☐ with kitchen ($60.00)

NOTES:

• Please include ONE night's advance room DEPOSIT to confirm the room reservation.
• Prevailing city occupancy tax per room per day will be added to room rate.
• Reservations must be received by MARCH 5, 1981, in order to fulfill your room reservation request.

Mail to:
Reservations, IEEE-CS Tutorial Week
ORLANDO MARRIOTT INN
6700 Sand Lake Rd., Orlando, FL 32807, (800) 228-9290
If you cannot attend Tutorial Week East 81, use this order form to purchase any of the tutorial texts listed below.

<table>
<thead>
<tr>
<th>ORDER NO.*</th>
<th>TUTORIAL TITLES</th>
<th>QTY.</th>
<th>MEMBER PRICE**</th>
<th>NON-MEMBER PRICE</th>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>288</td>
<td>VLSI: The Coming Revolution in Applications and Design</td>
<td></td>
<td>$16.50</td>
<td>$22.00</td>
<td></td>
</tr>
<tr>
<td>259</td>
<td>Microcomputer System Design and Techniques</td>
<td>12.00</td>
<td>16.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>266</td>
<td>Tutorial and Selected Readings in Interactive Computer Graphics</td>
<td>18.75</td>
<td>25.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>209</td>
<td>Distributed Processing</td>
<td>12.00</td>
<td>16.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>201</td>
<td>A Practical View of Computer Communications Protocols</td>
<td>12.00</td>
<td>16.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>297</td>
<td>Computer Networks</td>
<td>12.00</td>
<td>16.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>299</td>
<td>A Pragmatic View of Distributed Processing Systems</td>
<td>21.00</td>
<td>28.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>304</td>
<td>Local Computer Networks</td>
<td>12.00</td>
<td>16.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>310</td>
<td>Models and Metrics for Software Management and Engineering</td>
<td>15.00</td>
<td>20.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>312</td>
<td>Programming Language Design</td>
<td>15.00</td>
<td>20.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>211</td>
<td>Software Methodology</td>
<td>12.00</td>
<td>16.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>207</td>
<td>Software Testing and Validation</td>
<td>12.00</td>
<td>16.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>311</td>
<td>Software System Design: Description and Analysis</td>
<td>15.00</td>
<td>20.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

** IMPORTANT INFORMATION: **
* To assure prompt processing of your order, be sure to enter the 3-digit number that appears ahead of each publication title.
** Remember, member rates apply on the first copy for personal use only. No returns or refunds after 60 days from date of shipment (90 days overseas).
*** All unit prices include postage for 4th class book rate (U.S., Canada and Mexico). For priority shipping, add $2.00 per book. Overseas mail shipped sea mail (7-8 weeks delivery). For airmail (1 week delivery), please add $10.00 per book.

Charge Card No.          Exp. Date   | Signature   | California residents add 6% tax |
Name (please print)      | Member No.  | Optional Shipping Charge***    |
Address                  |             | Total                         |
City/State/Zip           | Country     |                             |

**SEND TO:**
IEEE Computer Society
10662 Los Vaqueros Circle
Los Alamitos, CA 90720

**PAYMENT (check one):**
☐ Check Enclosed
☐ Bill Visa/BankAmericard
☐ Bill Master Charge

Foreign checks must be in US dollars drawn on a US bank.