Abstracts

from recent papers by IBM authors

An Efficient Algorithm for Exploiting Multiple Arithmetic Units, R. M. Tomasulo, *IBM Journal of Research and Development* **11,** No. 1, 25–33 (January 1967). This paper describes the methods employed in the floating-point area of the SYSTEM/360 MODEL 91 to exploit the existence of multiple execution units. Basic to these techniques is a simple common data bussing and register tagging scheme which permits simultaneous execution of independent instructions while preserving the essential precedences inherent in the instruction stream. The common data bus improves performance by efficiently utilizing the execution units without requiring specially optimized code. Instead, the hardware, by "looking ahead" about eight instructions, automatically optimizes the program execution on a local basis. The application of these techniques is not limited to floating-point arithmetic or SYSTEM/360 architecture. It may be used in almost any computer having multiple execution units and one or more "accumulators." Both of the execution units, as well as the associated storage buffers, multiple accumulators and input/output buses, are extensively checked.

ASLT Circuit Design, R. F. Sechler, A. R. Strube, and J. R. Turnbull, *IBM Journal of Research and Development* **11,** No. 1, 74–85 (January 1967). The full switching-speed potential of high performance transistors is difficult to realize in a current switch configuration because of the instability which exists when many circuits are interconnected in a system. With a phase compensating network in the emitter current source, however, it has been possible to design a stable circuit using 1 Gc/sec transistors. Design techniques and engineering aspects of the circuit which result in a 5-nsec. in-the-environment propagation delay are described. Particular attention is given to the dc design, stability analysis, switching performance, evaluation, and specification of the circuit.

ASLT: An Extension of Hybrid Miniaturization Techniques, R. H. F. Lloyd, *IBM Journal of Research and Development* **11,** No. 1, 86–92 (January 1967). The hybrid miniaturization technique of applying active and passive components separately to a packaging substrate enables independent customization of components and minimizes stray capacitance between components. Manufacturing advantages result because all components can be pretested, and the critical component joining operation can be performed as a continuous operation. This paper describes IBM's Advanced Solid Logic Technology (ASLT) wherein the capabilities of hybrid circuit technology have been extended to produce a high-speed, high-density digital logic module compatible with automated production. Design considerations and assembly processes are detailed. A development of the power handling capability reviews the various thermal paths within the module.

Analysis of the Time-Dependence of Multi-Freedom Mechanical Systems in Relative Coordinates, M. A. Chace, *ASME Second Conference on Mechanisms,* Paper No. 66-Mech-23 (October 10–12, 1966). A theoretical model and a computational procedure are proposed for the analysis of the time-dependence of highly constrained, multi-degree-of-freedom mechanical systems. Problems are formulated as a simultaneous set of Lagrange differential equations in the system relative coordinates, accompanied by second-order constraints. The equations are linear in the second-order terms and are solved by a numerical integration procedure.

Computer-Aided Kinematic Synthesis of a General Planar Mechanism, I. S. Kiss, *ASME Second Conference on Mechanisms,* Paper No. 66-Mech-40 (October 10–12, 1966). This paper describes a computer-aided kinematic synthesis method. The method is restricted to the synthesis of a general planar mechanism with a single degree of freedom. The program solves a large class of synthesis problem combinations and is capable of optimizing many of the important mechanism and motion characteristics. The range of the program, together with the method of solution, is discussed and a sample computer input and output listing is included. The program is part of a computer-aided design system called COMMEND.

Computer Applications of Lasers, W. V. Smith, *Applied Optics* **5,** No. 10, 1533–1538 (October 1966). The applicability of lasers to the logic, memory, input/output, and data transmission linkage portions of computers are analyzed qualitatively. The more promising potential applications are considered to lie in the memory, interconnection, and input/output areas, and the least promising in the fast logic area. Coherent optical data preprocessing is cited as an example of an already clearly useful laser application.

Computer Graphics—Where Are We?, F. D. Skinner, *Datamation* **12,** No. 5, 28–31 (May 1966). Computer-linked graphic devices promise a breakthrough comparable to the introduction of data processing itself. This article reports on the use of display consoles—cathode-ray tube displays with versatile entry devices—now and in the near future. The many display devices now available range from small alphameric units to fairly sophisticated terminals which permit the display of graphic images with points or lines.

Computer Program Aids Circuit Analysis, E. T. Johnson, *Electronic Design* **14,** No. 23, 56–61 (October 11, 1966). S-plane or Laplace transform techniques for analysis and design have formed the basis for much of the linear circuit theory in recent years. Knowledge of the pole and zero locations can be an important aid in circuit design and network syntheses by providing insight into circuit stability, frequency and transient responses. However, tedious manual calculations required by these techniques limit practical application. This limitation can be removed by shifting the calculating task to a computer through an experimental program called LISA (LInear System Analysis). A free-format, applications-oriented input language makes the program easy to learn and use. Input to the system is entered as nodal circuit descriptions or as transfer functions. Outputs may be poles and zeros, frequency and transient response, root locus or sensitivity. Data is returned to the engineer in the form of listings or plots. Features such as the ability of the program to reflect parameter and topological changes allow the user to experiment, study and solve his design problems without getting lost in algebraic or programming details.

Computerized Design Information Systems and Their Impact on Draftsmen, V. L. Hoberecht, *Journal of Industrial Arts Education* **25,** No. 5, 15–17 (May–June 1966). Design information systems differ from other data processing systems primarily in the equipment used for gating information into and out of the system. Special input/output equipment has been developed for design information systems which allow a draftsman/designer and a computer to communicate with each other more quickly and effectively. They permit the handling of designs and other graphic data as well as printed data. Communication can be carried on in a "conversational" mode with questioning, interruptions, and decisions by both the draftsman/designer and the computer.

Design of a High-Speed Transistor for the ASLT Current Switch, J. Langdon and E. J. Van Derveer, *IBM Journal of Research and Development* **11,** No. 1, 69–73 (January 1967). The evolution of a high-speed current switch transistor design is described from initial design considerations through final optimization of horizontal geometry. It was found that a very narrow geometry was desirable, in order to produce the desired low base resistance (~40Ω). Other characteristics of this design include low capacitance, well-controlled emitter forward voltage, and high-frequency cutoff. Compatibility with the SLT form factor assures manufacturability. This transistor when used in ASLT circuits yields circuit delays of 1.8 nsec.

The Electrostatic Storage Display Tube: A New Projection Display System, J. M. Engel, *Eleventh Technical Symposium of the Avionics Panel of AGARD,* (November 7, 1966). The Electrostatic Storage Display Tube (ESDT) is a special tube which combines the intrinsic charge storage property of a dielectric membrane with the high-speed, high-resolution capability of a newly developed

electron gun and an independent optical system to provide bright, high-contrast, large-screen displays. The writing and recording functions of this tube have been completely separated from the actual display function, thus providing several advantages over conventional storage display tubes. A single tube may be used for theater-size displays or for hard-copy outputs on photosensitive media. Several different tube structures are described, all of which offer resolutions of about 2000 TV lines per useful field of view, good contrast, and erasure times of under two seconds. Other advantages include writing times varying from one frame per hour to commercial TV speeds, and long storage of the developed image (requiring no electric power) without flicker. Brightness of the display is determined entirely by external light sources.

Graphics in Design Engineering, W. H. Sass, *Paper Summaries of the SAE Advanced Launch Vehicles Conference,* Paper No. 660459, 1–24 (June 1966). Men and computers can work together in the design of complex components and systems using graphic input/output devices. Man-oriented I/O devices, such as light pens, keyboards, and displays, extend the capabilities of men and computers in evolving a design. Close man-machine rapport allows the designer to aid the computer in performing tasks beyond its programmed capability. This paper describes, in detail, one such graphic application that has been implemented for the design of digital computer circuit cards. References are made to an extended program for the design of complete computing systems.

IBM Graphic Display System, R. J. Houldin, *Information Display* **3,** No. 5, 34–40 (September/October 1966). Computer displays with graphic capabilities have added a new dimension to data processing by making information more accessible and more easily assimilated. Summaries, graphs, and charts can be quickly retrieved and viewed on many screens simultaneously. Several of the more advanced displays are beginning to link the creativity of man and the calculating power of the computer to design applications.

The IBM System/360 Model 91: Floating-Point Execution Unit, S. F. Anderson, J. G. Earle, R. E. Goldschmidt, and D. M. Powers, *IBM Journal of Research and Development* **11,** No. 1, 34–53 (January 1967). The principal requirement for the MODEL 91 floating-point execution unit was that it be designed to support the instruction-issuing rate of the processor. The chosen solution was to develop separate, instruction-oriented algorithms for the add, multiply, and divide functions. Linked together by the floating-point instruction unit, the multiple execution units provide concurrent instruction execution at the burst rate of one instruction per cycle.

The IBM System/360 Model 91: Machine Philosophy and Instruction-Handling, D. W. Anderson, F. J. Sparacio, and R. M. Tomasulo, *IBM Journal of Research and Development* **11,** No. 1, 8–24 (January 1967). The SYSTEM/360 MODEL 91 central processing unit provides internal computational performance one to two orders of magnitude greater than that of the IBM 7090 Data Processing System through a combination of advancements in machine organization, circuit design, and hardware packaging. The circuits employed will switch at speeds of less than 3 nsec., and the circuit environment is such that delay is approximately 5 nsec. per circuit level. Organizationally, primary emphasis is placed on (1) alleviating the disparity between storage time and circuit speed, and (2) the development of high speed floating-point arithmetic algorithms. This paper deals mainly with item (1) of the organization. A design is described which improves the ratio of storage bandwidth and access time to cycle time through the use of storage interleaving and CPU buffer registers. It is shown that history recording (the retention of complete instruction loops in the CPU) reduces the need to exercise storage, and that sophisticated employment of buffering techniques has reduced the effective access time. The system is organized so that execution hardware is separated from the instruction unit; the resulting smaller, semiautonomous "packages" improve intra-area communication.

The IBM System/360 Model 91: Some Remarks on System Development, M. J. Flynn and P. R. Low, *IBM Journal of Research and Development* **11,** No. 1, 2–7 (January 1967). This paper is intended to introduce the reader to IBM'S SYSTEM/360 MODEL 91 and to the technical articles in this issue that describe its development. We review the basis in system requirements from which the design objectives were drawn, describe the overall system briefly, and present a summary of the engineering topics discussed in the other papers.

The IBM System/360 Model 91: Storage System, L. J. Boland, G. D. Granito, A. U. Marcotte, B. U. Messina, and J. W. Smith, *IBM Journal of Research and Development* **11,** No. 1, 54–68 (January 1967). This paper discusses the design concepts employed in the development of the IBM SYSTEM/360 MODEL 91 storage system. Particular attention is paid to the exploitation of SYSTEM/360 capabilities in the areas of large storage capacity, concurrent operation, and flexibility, as they apply to the highly overlapped MODEL 91 system. An interleaved set of main storage modules is used with the MODEL 91 to help mask the difference between machine cycle time and storage access time. The set is connected to the central processor, peripheral storage control element and maintenance console by three time-shared buses—one for addresses, one for data-in, and one for data-out. The main storage control element (MSCE) controls these buses to maximize the storage access rate. To achieve minimum access time, requests are normally sent directly to the storage modules. The proper module is selected by the MSCE, the address gated in and the storage cycle started. If the module is busy from a previous request, the request is stored in a request stack for a later attempt. If the request is accepted, it is stored in an accept stack. This stack controls the data-out gating of the storage modules, and notifies the CPU of the destination of returning data. It also furnishes module busy information which controls the recycling of rejected requests. An important feature is the ability of the MSCE to logically sequence store/fetch requests, by interlocking the rejected requests with the current request without any degradation of minimum access time. Additionally, each address sent to the MSCE is compared with the addresses of waiting and in-process requests. This allows serial fetching of two adjacent single words of a double-word storage cycle. Fetches following stores to the same location can be executed without waiting for a fetch storage cycle. Peripheral storage is provided in the system for both block transfers of data and individual word fetches and stores. All requests to peripheral storage are sent via the peripheral storage control element. The MSCE is synchronized with the CPU and uses the same machine cycle. Ideally, a request can be honored each machine cycle, but the actual rate is determined by storage module conflicts. The storage system performance is measured in access rate and access time. The MSCE has been simulated to measure the effects of storage speeds, degree of interleaving, and changes in MSCE controls.

Large-Scale Logic Arrays: Testing for the Millions, J. W. Lind, *Electronics* **39,** No. 21, 98–101 (October 17, 1966). Computer logic circuits are attaining a high level of integration in advanced computer technology. Consequently, many circuit tests per circuit block are required as a result of this integration trend. DC functional testing of logic circuits is one type of test required. In this type of testing, logic inputs ("1" and "0"), DC loads and logic power supplies must be applied to a large number of circuit pins at an accuracy of a few tenths of a percent. In addition, the application and measurement of "1" and "0" logic levels must be accomplished at a high rate of speed to insure a reasonable test time per circuit block. This paper describes the analog portion of an advanced logic test system designed and built by an IBM test equipment group to meet the rigorous test requirements dictated by high density, low level integrated logic circuits.

Meeting the Metric Problem, A. Saliby, *Engineering Digest* **12,** No. 10, 54–56 (October 1966). When machines developed in one country are to be manufactured in other countries that use a different method of measurement and different methods of orthographic projection, difficulties arise. A method of overcoming these problems has been found and is being used now by IBM engineers. It involves a dual dimensioning system where all dimensions are given in both inches and millimetres.

A Modified Bode Criterion for Stability of Feedback Systems, P. J. Granata and H. B. Aasnaes, *EEE* **14,** No. 10, 97–99 (October 1966). A convenient method has been developed for the design of a stable single-loop feed-back system. This method involves superimposing the curve of the inverse of the feedback transfer function on the Bode plot of the forward transfer function. If pre-established restrictions on the slope difference between these two plots are fulfilled at their intersection, system stability is obtained. The method permits the proper compensation network to be determined, while satisfying closed-loop gain and stability requirements through separate manipulation of the forward and feedback transfer functions.

Ueber die Notwendigkeit einer Fehlerschranken-Arithmetik fuer Rechenautomaten*, (On the Necessity of Error-Bound Arithmetic for Automatic Computation), K. Nickel, *Numerische Mathematik* **9,** No. 1, 69–79 (1966). In extension of Moore's "interval-arithmetic," a new "error-bound arithmetic" has been suggested. Each "number" in this new arithmetic consists of a couple of real numbers: the approximate value and an error bound for the given number. It is shown in four examples that for arithmetical and logical reasons such a new arithmetic is necessary for most algorithms. With the aid of this arithmetic it is possible, for example, to find approximate roots of numeric equations and to give error bounds for them. It is also possible to give a criterion for the number of iteration steps stopped *without any additional a priori knowledge about the solution* (such as: stop, if the difference between two consecutive approximations is less than 10^{-15}). In the paper it is suggested that the new algorithm be written in FORTRAN, ALGOL, or PL/I.

* Work performed while the author was at Notre Dame University as visiting professor.

Die numerische Berechnung der Wurzeln eines Polynoms*, (Numerical Computation of the Roots of Polynomials), K. Nickel, *Numerische Mathematik* **9,** No. 1, 80–98 (1966). The paper deals with a new method for the computation of all (complex) roots of a polynomial of the degree n with given (complex) coefficients. The result is an approximate value plus an error bound for each root. The algorithm does not need initial values. It never fails, not even in the case of multiple roots or of clusters of roots. Experiments have been taken with random-generated polynomials with many more than 100,000 roots. In the range of $n = 1$ to $n = 30$ the number of iterations necessary for the evaluation of one root never exceeds six in average and 16 in the worst case. The experimental results also give much information about the dependence on n for error and error bound in average and in the most disadvantageous case, respectively. It is essential for the algorithm given that an "error bound arithmetic" as mentioned in the previous paper is available and can be used.

* Work performed while the author was at Notre Dame University as visiting professor.

Performance Deterioration of Optimum Systems, J. J. Rissanen, *IEEE Transactions on Automatic Control* **AC-11,** No. 3, 530–534 (July 1966). This paper deals with the problem of evaluating the extent to which system parameters may be changed and still guarantee that system performance will remain within a specified limit. Particular attention is given to linear systems with quadratic performance indices. A sampled data system is analyzed by way of example.

Practical Bang-Bang Design, T. R. Fredriksen, *Control Engineering* **13,** No. 10, 78–84 (October 1966). An experimental control system illustrates a typical variation of bang-bang control where a detent provides the required second mode. Exclusive use of solid-state switching logic and power control gives performance not possible with previous relay controllers. The control concepts are described in terms of modern control theory.

A Real-Time Operating System for the Saturn V Launch Computer Complex, F. R. Palm, *Proceedings of* IBM *Real-Time Systems Seminar,* Paper No. 4, 1–8 (November 1966). As missile systems become increasingly larger and more complex, the automation of checkout and launch becomes a necessity. For the Saturn V system, the National Aeronautics and Space Administration has selected a three-computer complex and special input/output and interface equipment to assist in the checkout and launch phases. This complex of equipment is required to do three basic tasks in parallel during these phases. These tasks include: (1) intercommunication between man and vehicle, (2) monitoring of system parameters, and (3) performance of vehicle tests. In order to make these computers function as a system and to allow efficient accomplishment of the above tasks, IBM has implemented an operating system. This operating system, while providing for these tasks, is programmed so that loss of most types of input/output equipment does not keep the system from providing services for the main tasks due to hardware malfunctions.

Real-Time Optimal Guidance, K. R. Brown and G. W. Johnson, *Proceedings of* IBM *Real-Time Systems Seminar,* Paper No. 3, 1–11 (November 1966). This paper describes an algorithm for repetitive inflight calculation of optimal rocket steering laws for orbital injection and rendezvous missions. The algorithm is based on a general "indirect" method of solving the optimal trajectory problem as a boundary value problem in ordinary differential equations, and it avoids assuming artificial physical simplifications or specialized mission definitions. At each guidance cycle, the solution to the boundary value problem is updated by a single iteration of a modified Newton's method using partial derivatives obtained by numerical integration of variational differential equations. Simulation results show that a precalculated linear steering law provides adequate initialization to assure that this iterative guidance scheme recovers from worst-case inflight perturbations and achieves desired end conditions with near-minimum cost.

A Study of Computer-Assisted Instruction in Industrial Training, H. A. Schwartz and R. J. Haskell, Jr., *Journal of Applied Psychology* **50,** No. 5, 360–363 (October 1966). The study was undertaken to test the feasibility of remote computer-assisted instruction as an industrial training technique. 79 newly hired electronic technicians received their required training in basic data-processing principles through programmed texts, the standard method used for this presentation. 25 equivalent students received the same training through a keyboard-operated terminal device linked remotely to an IBM 1440 computer system. No significant differences in examination scores were obtained; however, there was a significant saving (approximately 10%) in the time required to complete the course. In an attitude questionnaire administered subsequent to the courses, both groups rated their respective method of instruction as approximately equal to regular classroom techniques in terms of effectiveness and desirability.

A Statistical Analysis of Telephone Circuit Error Data, P. A. W. Lewis and D. R. Cox, *IEEE Transactions on Communication Technology* **COM-14,** No. 4, 382–389 (August 1966). Berger and Mandelbrot in 1963 proposed a particular renewal process as a model for the occurrence of errors in data transmitted over telephone circuits. Besides the assumed independence between the successive intervals between errors, they assume that the intervals have a Pareto distribution. Their graphical analyses of large amounts of data indicated departures from the model which Mandelbrot proposed in 1964 to account for in an extended model. Some simple formal statistical procedures are given for analyzing this sort of data, procedures that are not affected by the possibility that the population mean-interval-between-errors is infinite. The departures from independence of intervals noted by Berger and Mandelbrot are formally verified from the analysis of data from a single data transmission test. A separate analysis of another set of data is also made and the results are compared to see what features of the error patterns are general, and what features are particular to different transmission conditions. In both sets of data analyzed, the outstanding feature detected is the strong positive correlation between successive long intervals between errors. Evidence is also found which indicates that the upper tail of the marginal distribution of intervals between errors does not follow a hyperbolic law.