

Listed are abstracts from recent papers by IBM authors. Inquiries should be directed to the publications cited.

A cyclic-queue model of system overhead in multiprogrammed computer systems, P. A. Lewis and G. S. Shedler, *Journal of the Association for Computing Machinery* **18**, No. 2, 199–220 (April 1971). A probabilistic model is presented of a multiprogrammed computer system operating under demand paging. The model contains an explicit representation of system overhead, the CPU requirements and paging characteristics of the program load being described statistically. Expressions for steady-state CPU problem program time, CPU overhead time, and channel utilization are obtained. Some numerical results are given which quantify the gains in CPU utilization obtained from multiprogramming. It is also pointed out heuristically and demonstrated numerically that an actual decrease in CPU utilization results if there is too much overhead associated with multiprogramming and if the average time between page exceptions decreases too rapidly with increasing number of multiprogrammed jobs.

Probability models for buffer storage allocation problems, Donald P. Gaver, Jr. and Peter A. W. Lewis, *Journal of the Association for Computing Machinery* **18**, No. 2, 186–198 (April 1971). This paper considers some of the issues that arise when messages or jobs in-bound to a computer facility are buffered prior to being processed. Models are developed that describe (a) the results of blocking a single memory unit for the use of diverse messages, (b) the occupancy behavior of a buffer that is tied to a single message source, and (c) the occupancy of a buffer dynamically shared among many independent sources.

Sensor based computers applied to the laboratory, D. A. Daniels and D. A. Opp, *Materials Research and Standards* **11**, No. 2, 12–18 (February 1971). The sensor based computer is distinguished from a conventional data processing machine by its capability to collect data from external sources, to analyze those data, and as a result, to close the loop back to the external device by performing a control function. The Engineering Data Integrated Test (EDIT) Center is a laboratory automation project at the SDD Laboratory, IBM Endicott. Since its inception in 1967, EDIT personnel have been developing a test center concept that employs a central CPU (1800 System) for data collection and engineering tests. The paper discusses EDIT system concepts, features required by laboratory personnel, and outlines how laboratory applications may be interfaced to a single time-shared computer test center. Several specific laboratory experiments and tests are described, with emphasis placed on the user operating techniques.

Parallelism, pipelining, and computer efficiency, Tien Chi Chen, *Computer Design* **10**, No. 1, 69–76 (January 1971). Despite the impressive advance in modern computer designs, there will always be need for machines more powerful than those available. In the quest for performance above and beyond that deliverable by hardware componentry, two alternative multiprocessing approaches to computer design can be taken. One alternative is to subdivide each oncoming job among many identically constructed mechanisms, commonly referred to as parallelism; the other is to develop a collection of specialized mechanisms capable of working simultaneously to form a general-purpose organization, commonly called overlap, or pipelining in its extreme form. This article is a study of these two approaches based on a simple geometric analysis.

Abstracts

Key-to-address transform techniques: a fundamental performance study on large existing formatted files, V. Y. Lum, P. S. T. Yuen, and M. Dodd, *Communications of the ACM* **14**, No. 4, 228–239 (April 1971). The results of a study of eight different key-to-address transformation methods applied to a set of existing files are presented. As each method is applied to a particular file, load factor and bucket size are varied over a wide range. In addition, appropriate variables pertinent only to a specific method take on different values. The performance of each method is summarized in terms of the number of accesses required to get to a record and the number of overflow records created by a transformation. Peculiarities of each method are discussed. Practical guidelines, obtained from the results are stated. Finally, a proposal for further quantitative fundamental study is outlined.

The terminal in the terminal-oriented system, M. G. Smith, *The Australian Computer Journal*, **2**, No. 4, 160–165 (November 1970). The terminal of the 1970's is pictured as an increasingly important component in data processing systems, both economically and functionally. Technological and economic factors influencing future terminal design are presented, including rationales supporting the emergence of the A/N display terminal as the "workhorse" terminal of the future.

Deterministic pushdown store machines and real-time computation, Stephen N. Cole, *Journal of the Association for Computing Machinery* **18**, No. 2, 306–328 (April 1971). A comparison is made of the computing capabilities of deterministic pushdown store machines and real-time iterative arrays of finite-state machines. The main result is that every deterministic pushdown store computation can be performed by some (multidimensional) iterative array in real-time. The latter are strictly more powerful since they can recognize the set of palindromes in real-time, which deterministic pushdown store machines cannot do even if permitted unlimited computing time. During the development of the main result, variants of pushdown store machines: the tabulator machines and the n-dimensional pushdown store machines, are introduced. By imposing a real-time constraint and letting the number of tabs and the number of dimensions vary, an infinite hierarchy of pushdown store (deterministic context-free) languages is obtained.