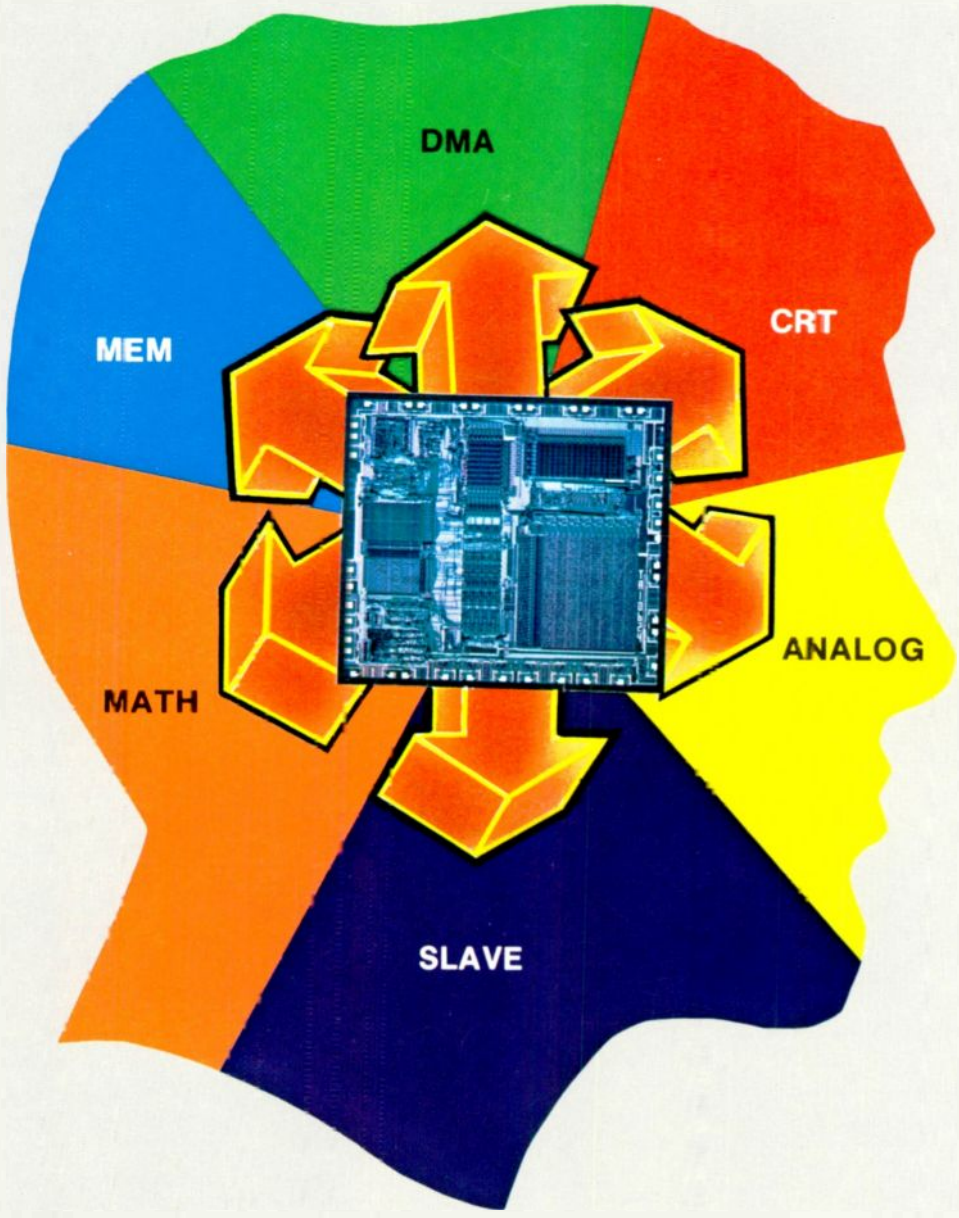
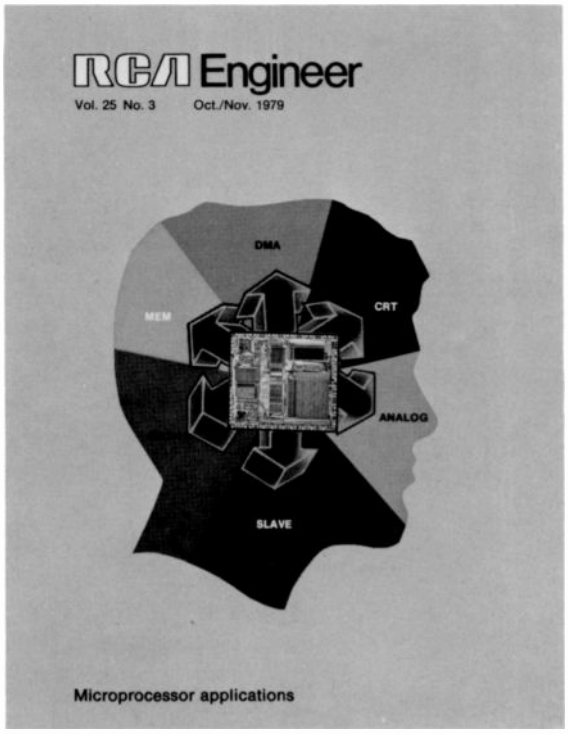
RCA Engineer

Vol. 25 No. 3 Oct./Nov. 1979



Microprocessor applications



The cover is an artist's interpretation of an "intelligent" product with the microprocessor as its "brain." (In this case, we show the 1804 microcomputer, soon to be released on the market.) Each of the surrounding segments represents the types of devices which make the microprocessor work effectively in a product.

The ANALOG segment represents those devices which convert the incoming and outgoing signals from analog to digital and vice versa. CRT (cathode ray tube) refers to a display device for microcomputer to operator information. DMA (direct memory access) devices permit high speed access to data stored in the computer's memory without disrupting normal computer operation. The memory (MEM) segment represents the stored information of the system. MATH: Because microprocessors are usually not designed to perform arithmetic operations efficiently, special chips have been designed to perform these operations while interfacing with the microprocessor. Most multi-microprocessor systems consist of a master processor along with SLAVE processors that run in parallel with the master but take their direction from it.

The importance of choosing the appropriate microprocessor and peripheral devices for each application is stressed in Don Latham's cover message on the next page.

RCA Engineer

A technical journal published by
RCA Research and Engineering
Bldg. 204-2
Cherry Hill, N.J. 08358
Tel. 222-4254 (609-338-4254)

•To disseminate to RCA engineers technical information of professional value • To publish in an appropriate manner important technical developments at RCA, and the role of the engineer • To serve as a medium of interchange of technical information between various groups at RCA • To create a community of engineering interest within the company by stressing the interrelated nature of all technical contributions • To help publicize engineering achievements in a manner that will promote the interests and reputation of RCA in the engineering field • To provide a convenient means by which the RCA engineer may review his professional work before associates and engineering management • To announce outstanding and unusual achievements of RCA engineers in a manner most likely to enhance their prestige and professional status.

Microprocessors:

their impact on system designs and systems designers

Donald C. Latham

Microprocessors are a milestone achievement in technology which will very likely have an impact on our lives comparable to that of the printing press, the internal combustion engine, and the transistor. Fulfillment of a technological promise can be so long in coming that the impact of new products is not immediately recognized, but microprocessors have had an impact while still in their infancy. Significantly, the microprocessor potential continues to grow and promises to be overwhelming. On-chip density and speed improvements in the next three to five years may reach 100,000 gates per chip with internal clock rates over 100 MHz. With this level of technology, extremely complex computer and signal processing systems could be integrated on a single chip. Application designs will become even more software intensive, and hardware costs in both computer and memory will be driven down to a small fraction of the overall costs.

Full realization of microprocessor potential requires new and perhaps radical approaches and discipline in system design, software and firmware development, test and evaluation, and field service. Imagination and resourcefulness will be as necessary in this process as they were 25 years ago in the initial attempts to apply the transistor.

For many years after the invention of the transistor and its availability, we had to wean electronic equipment designers from the notion of replacing vacuum tubes with transistors on a one-for-one basis. Once this elementary notion was dismissed, the full potential of transistor circuits was realized, and an entirely new generation of electronic products appeared. An analogous situation exists with microprocessor applications. Many early applications simply mimicked hardware features in software, basically, doing the same thing in a different way. More recently, however, with the maturing of microprocessor architecture and software design and testing, we are beginning to realize applications which were initially considered impossible or unreasonable.

The necessity for radical departures in the way of doing business places new demands on design and systems engineers. Each engineer must become familiar with the new technology. Selecting the "right" microprocessor for an application is not a trivial chore: 60 to 80 different microprocessor architectures are readily available on the commercial market, and an additional three to ten times that number are available in supporting functional LSI arrays. Many key characteristics must be evaluated, such as flexibility of instruction set, data type manipulation, memory address modes, and flexibility of the I/O and of the arithmetic and logic operations.

Typically less than ten percent of a microprocessor-based project involves pure hardware design issues, while the remaining 90 percent is devoted to microprocessor selection, systems architecture planning, software design, system integration, and testing. As a result, today's microprocessor systems designers must make extensive hardware/software tradeoffs at the front end of the program, develop innovative programmable designs, create test philosophies, and employ key testing tools such as logic analyzers, pattern generators, and in-circuit emulators.

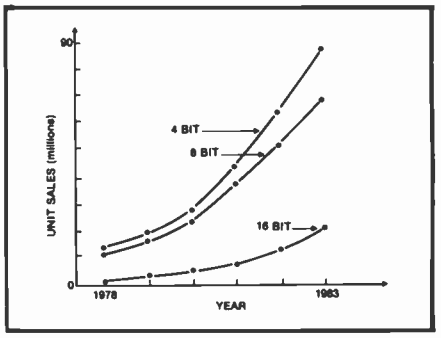
In this issue, we have a sampling of RCA microprocessor applications. A host of exciting, and even revolutionary, new microprocessor-based products and systems are soon to follow.

Donald C. Latham
Division Vice President, Engineering
Government Systems Division
Moorestown, N.J.

Highlights

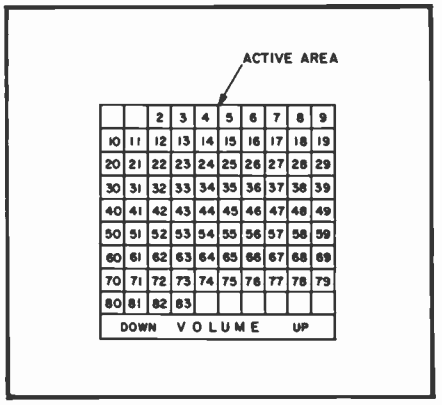
introduction

why is the microprocessor industry growing at such a rapid pace and what will be its effect on the application's concept?



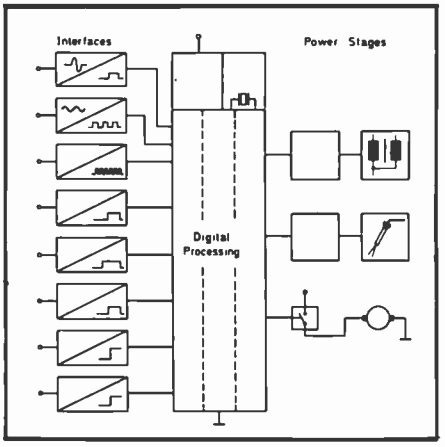
microprocessor-based products

RCA is making intelligent TV receivers and cameras.



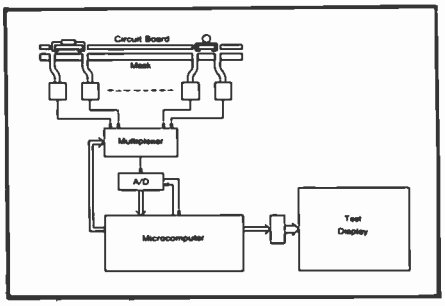
microprocessors in systems

microprocessors aid in energy conservation for automotive, temperature-regulation and communication systems.



microprocessors in manufacturing

introducing intelligence into automatic testing saves money and reduces errors in electrical parts inspection.

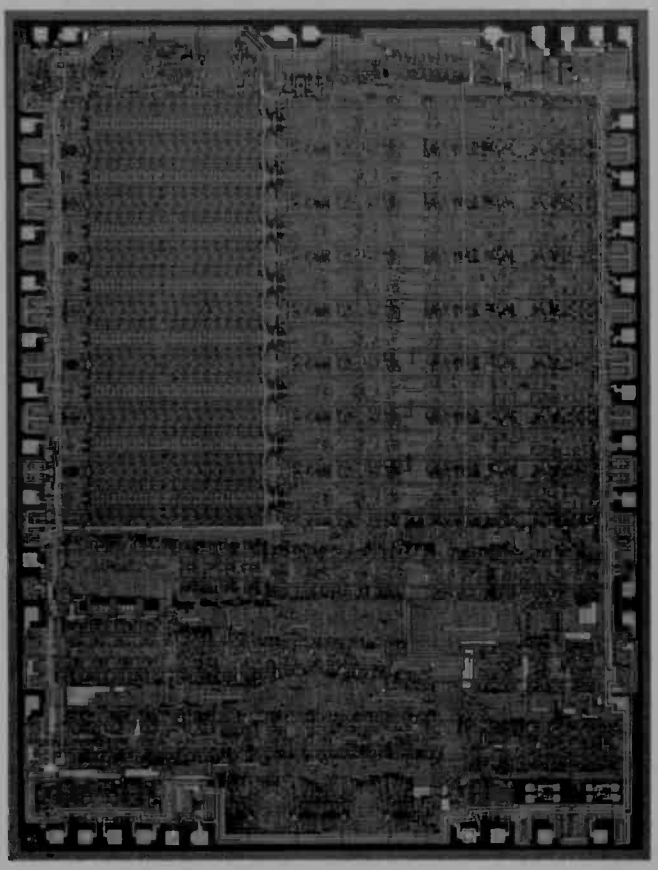


In future issues

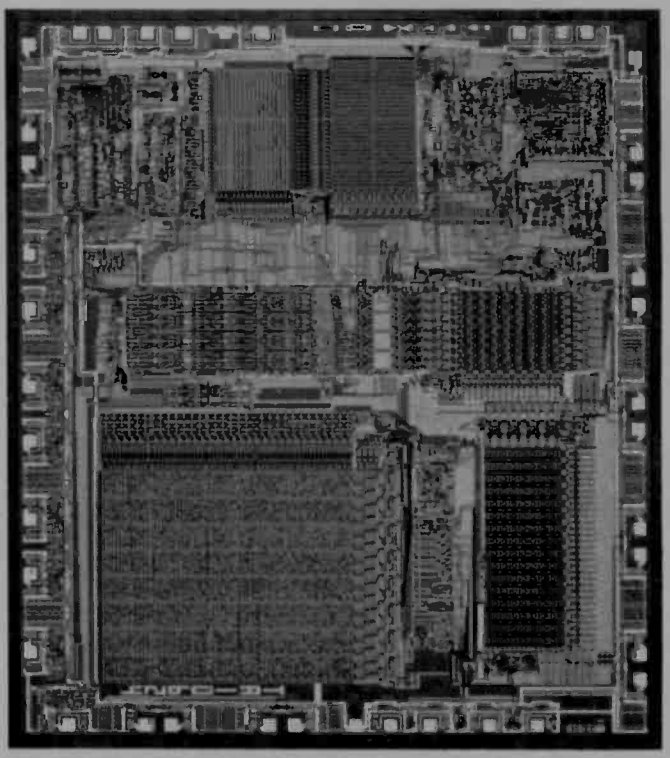
quality & reliability, RCA worldwide, color TV receivers, anniversary issue

RCA Engineer

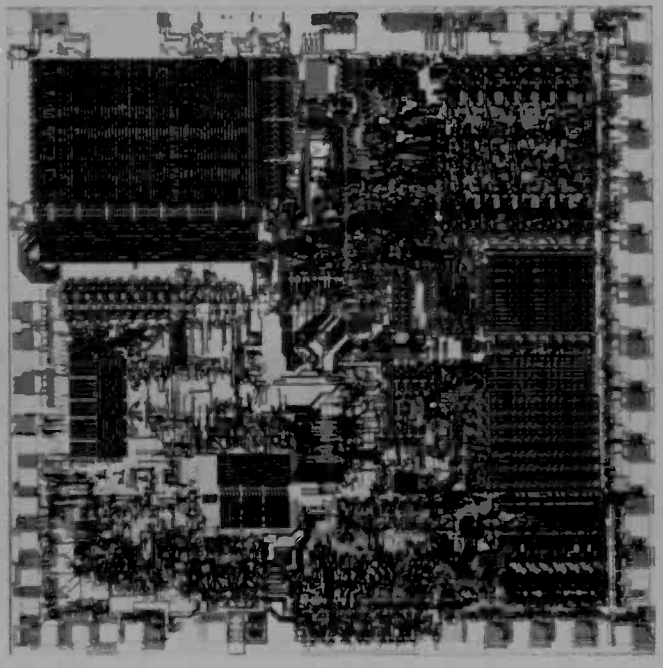
Vol 25|No. 3 Oct.|Nov. 79



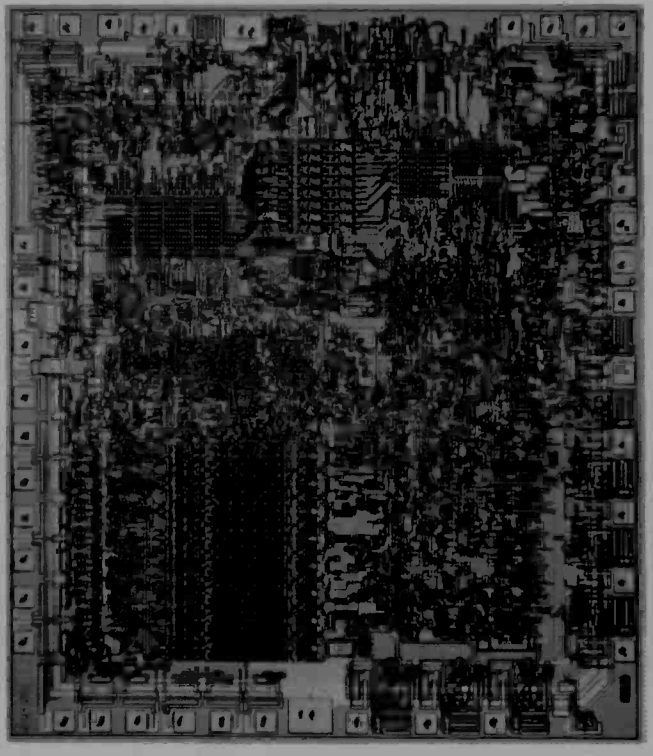
RCA 1802, introduced in 1975, 5400 transistors, 40,000 mil² area, CMOS.



RCA 1804, introduced in 1979, 32,000 transistors, 54,500 mil² area, CMOS SOS.



Intel 8086, introduced in 1978, 29,000 transistors, 51,000 mil² area, HMOS†.



Intel 8080, introduced in 1974, 4500 transistors, 32,400 mil² area, NMOS†.

†Courtesy of INTEL corporation

P.M. Russo

The microprocessor industry and the emerging applications revolution

Microprocessor and microcomputer applications have a profound impact on industry as well as the individual consumer — yesterday, a computer; today, a microprocessor; tomorrow, personalized microcomputer systems.

Abstract: *The capability of large scale integration and its interaction with rapidly evolving computer and microprocessor technologies are reviewed. The microprocessor industry is discussed with special emphasis on categorizing microprocessors and projecting unit and dollar sales volumes. Industrial, commercial and consumer applications of microprocessors are reviewed, focusing primarily on new products made possible by this new technology. Finally, a brief look is taken at future trends in microsystem evolution and their impact on the emerging applications revolution.*

The rapidly unfolding digital LSI revolution, which began in the late 1960s, is leaving a permanent imprint on many aspects of our lives. The first computer revolution began with Von Neumann's work in 1945,[1] and rapidly evolved to today's highly computerized and data-oriented society. The second computer revolution began with the commercialization of the microprocessor in 1971, when Intel introduced the 4004, 4-bit single chip central processing unit (CPU).[2] It took a few years for the business impact of microprocessors to be understood, and a few more years for the potential effect on the consumer to be visualized.[3]

The rapid progress in computer and digital electronics technology is illustrated in Fig. 1. It took a decade from the 1945 invention of the stored program computer (instructions and data stored in same memory) to the development of the first all-transistor machine. A decade later, standard digital logic families such as Texas Instrument's TTL and RCA's COS/MOS began to emerge. This is the era when Large Scale Integration (LSI) capability, which allowed one to interconnect large numbers of transistors on a single piece of silicon substrate, began to evolve.

As stated above, the first microprocessor (μP), the Intel 4004, was commercialized in 1971. The 4004 concept arose from attempting to satisfy several custom calculator chip designs with a single family of LSI parts. Though Intel is often granted the honor of inventing the μP, it is fair to point out that many computer architects with an interest in LSI were postulating, and working towards microprocessors prior to this date. The early work on the RCA 1802, for example, predates the Intel

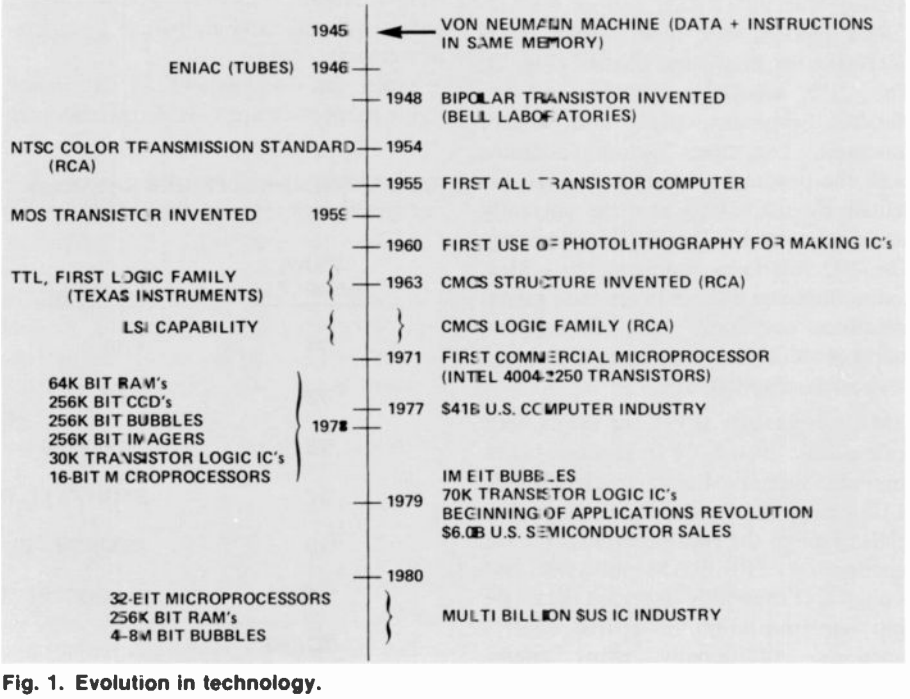


Fig. 1. Evolution in technology.

Reprint RE-25-3-1
Final manuscript received Aug. 29, 1979.

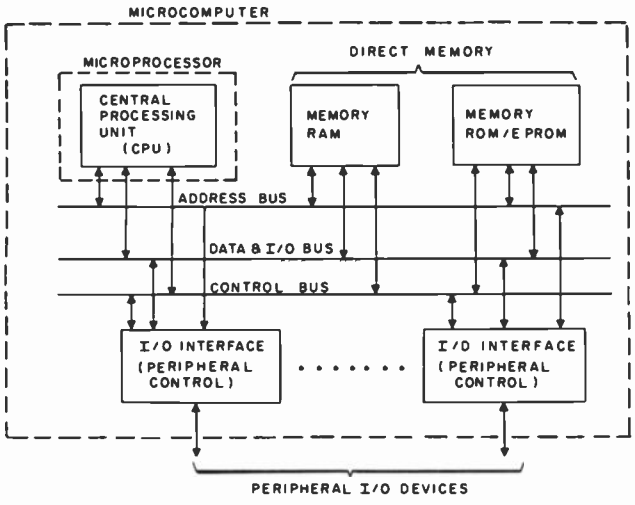


Fig. 2. The fundamental sub-systems of any single-processor system.

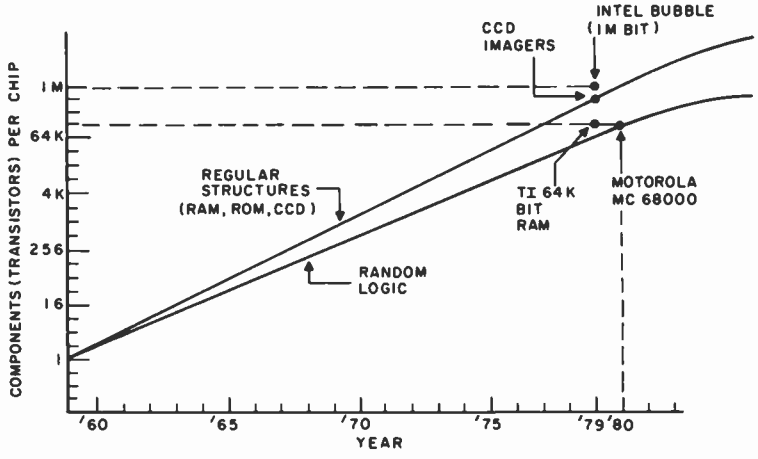


Fig. 3. Evolution in digital semiconductor IC complexity.

4004 announcement. Less than a decade after the Intel 4004, million-device integrated circuits (ICs) are being sampled, powerful 16-bit microprocessors are a reality, and the applications revolution marches on to the accelerating rhythm of the LSI drum.

Microprocessors and microcomputers

Any single-processor computer system, from the simplest microwave oven controller to a sophisticated data processing system, consists of three classes of subsystems: the central processing unit (CPU); direct memory, which can be READ/WRITE (RAM) and/or READ-ONLY (ROM); and input/output (I/O) interfaces for peripheral control (Fig. 2). The CPU subsystem performs all the classical arithmetic, logic, and control functions. The direct memory contains both the program (instructions to be executed by the CPU) and the currently active data on which the CPU is operating. The I/O interfaces represent the critical communication links between the internal computer operations and the external world of I/O devices such as mass memory, keyboards, displays, etc.

Microprocessors (μPs) are single chip realizations of the CPU function of a computer system. Microcomputers (μCs), in the context of this paper, are single chip realizations of the entire computer system function, i.e., CPU, RAM and ROM, and enough I/O capability to permit the single chip implementation of desired control functions. Additionally, many microcomputers sport an option of having either mask-programmed ROM (low-cost for high-volume applications) or EPROM (UV light erasable, reprogrammable for low-volume applications). From the above, it appears that minimal configurations for microprocessor-based systems require three chips. Other system architectures may require the use of multiple CPUs to achieve desired objectives.[4,5,6]

Trends in LSI/ VLSI technology

The history of LSI and VLSI (Very Large Scale Integration) is brief but explosive. Figure 3 illustrates the rapid evolution in IC capacity, with the upper boundary representing regular structures (e.g., RAMs, ROMs, etc.) while the lower boundary represents random logic (e.g., microprocessors).

Since the development of the metal oxide semiconductor (MOS) transistor in the late 1950s, device complexity has doubled every year. The primary contributing elements to this growth rate are larger die size, higher density (finer microstructure), and advances in device design. This rate of progress has not been uniform for all technologies — for example, EPROM technology has been evolving at a rate where doubling in complexity has occurred about every two years (Table I).

Beyond 1979, the growth rate in IC complexity will probably slow to doubling every 1.5 to 2 years. The reasons for this slowing are due partially to technological limits being approached, but primarily to the fact that it is not yet clear how designers can effectively make use of these complex ICs containing hundreds of thousands of transistors.[7,8]

The growth in complexity has a direct effect on the cost of integrated circuits. The cost of LSI integrated circuits consists of two major components: silicon chip cost and assembly/test cost. Based on the

Table I. Evolution of EPROM technology.

YEAR (SAMPLED)	EXAMPLE	BITS	BYTES
'71	1702	2K	256
'73	2704	4K	512
'75	2708/2758	8K	1K
'77	2716 INTEL,TI	16K	2K
'79	2532/2732 TI, INTEL	32K	4K
'80/'81	TI, MOSTEK, INTEL	64K	8K
'82/'85	–	128K	16K

available design techniques and process technology, silicon chip cost is roughly an exponential function of complexity. On the other hand, the assembly/test cost, as a first approximation, can be treated as independent of complexity. This leads to an inverse relationship between assembly/test cost per function and complexity. By combining these two components, the cost per function will have a minimum corresponding to the optimum complexity for the current state of the art design technique and process technology (Fig. 4). As time goes on, the silicon chip cost will decrease due to improving density, processing technology, and material. This will result in the optimum cost per function decreasing with time as shown in Fig. 5.

From Table I and Figs. 4 and 5, it is apparent that, as time progresses, more and more of the system will fit on a single integrated circuit, and the cost per computation function will decrease exponentially. This will make possible distributed intelligence in the broadest sense, where every sub-system will have local processing capability and the ability, where needed, to interact with other intelligent sub-systems or systems.

Microprocessor impact on LSI/VLSI technology

The rapidly increasing use of LSI and VLSI in non-memory applications would not have occurred if the microprocessor had not come to the rescue. Figure 6 illustrates this phenomenon. Prior to the invention of the microprocessor, ever more complex ICs were developed to perform ever more specialized applications. Hence, except for a few large volume market segments (e.g., data communications), IC volume decreased rapidly with complexity.

With the invention of the microprocessor, it became economically viable

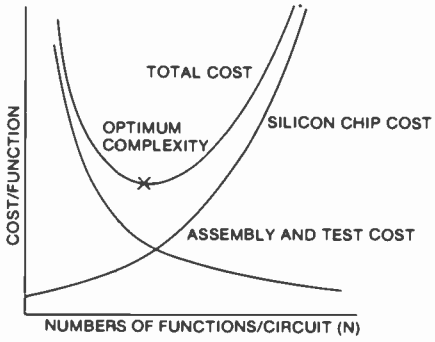


Fig. 4. LSI cost contributing factors showing optimum complexity for lowest cost per function.

to develop large complex random logic ICs and then specialize their application in software. In fact, it is the invention of the stored-program computer itself (of which μPs and μCs are embodiments) that has spurred the use of RAM and ROM and has generated the volumes needed to attain the exponentially decreasing per-bit costs. Figure 7 illustrates the evolution of dynamic NMOS (N Channel MOS) RAM cost during the past decade, projected through the early 1980s. It is our best guess that costs of 0.012 cents/bit will be reached in the early 1980s with 64 K-bit dynamic RAM chips selling for about $8 each in volume. These costs are possible because NMOS RAMs are the dominant memory chips used in computer systems giving rise to extremely large volume production with the attendant learning curve benefits. For example, Mostek Corporation, which controls about 30% of the 16K NMOS dynamic RAM market, is expected to double production to about 30 million units in 1980 — a very large volume indeed. Density improvements in NMOS RAMs are yielding average areas under 0.50 mils^2 (1 mil = 0.001 in.) per transistor (e.g., National NCM 4164 64K-bit RAM chip has an area of 3400 mils^2).

Categorizing microprocessors and microcomputers

In categorizing μPs and μCs, discussion will be restricted to MOS devices. Bipolar devices, to date, have been extensively used only in specialized applications requiring high throughput. The bulk of bipolar μPs (there are no bipolar μCs) are of the bit-slice variety requiring multi-chip CPU implementations (the Fairchild 9440) which emulates the Data General NOVA instruction set is a noteworthy exception). Finally, 4-bit μPs *will not* be discussed (4-bit μCs *will* be discussed), since they are obsolete and their days are numbered — in fact, Intel Corporation is no longer taking orders for its 4040 family of 4-bit μPs.

Microcomputers range from relatively simple 4-bit devices (Texas Instruments TMS 1000 series, National COPS 400 series, etc.) with volume costs in the $1-3 range, to more powerful 8-bit devices (Mostek 3870 series, Intel 8048 series, etc.) with volume costs in the $3-10 range.

Microprocessors, on the other hand, vary from mid-range machines (Intel 8085, RCA 1802, Zilog Z80, etc.) with volume CPU costs in the $2-10 range, through

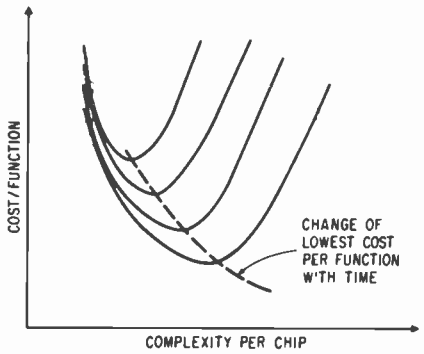


Fig. 5. Cost per function decreases with time.

higher performance 16-bit CPUs (Motorola 68000, Zilog Z8000, Intel 8086, TI 9900, etc.) whose capability approaches that of mid-1970s minicomputers, and whose chip costs may range from tens to hundreds of dollars. These latter devices are relatively new to the market (8086 sampled in 1978, Z8000 and MC68000 sampled in late 1979), and hence their costs will tumble as we move down the learning curve.

Figure 8 presents a rough view of processor performance ranges vs computer category. Note that the dollar figures shown represent mature product, i.e., those in volume production. Note also that even though many 8-bit μPs cost less than 8-bit μCs, the latter have on-board RAM, ROM and I/O, which usually results in far lower system cost.

Microprocessors and μCs exist in a wide variety of technologies. *p*-channel MOS (PMOS) is fading as a viable technology. *n*-channel MOS is the dominant technology and is the standard for all existing 16-bit machines. In the 4- and 8-bit categories, CMOS parts are available for low power applications (RCA 1802, CMOS TMS 1000, etc.), at nominal cost premiums along with bipolar devices (Fairchild 9440, etc.) for high performance. Silicon-on-sapphire (SOS) parts are not yet commercially available.

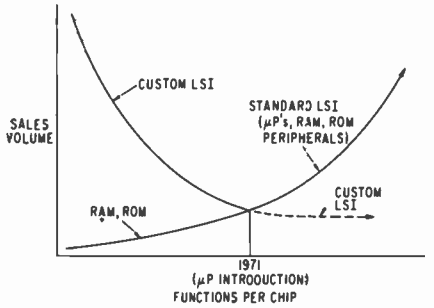


Fig. 6. Custom vs. standard LSI sales volume vs. chip complexity.

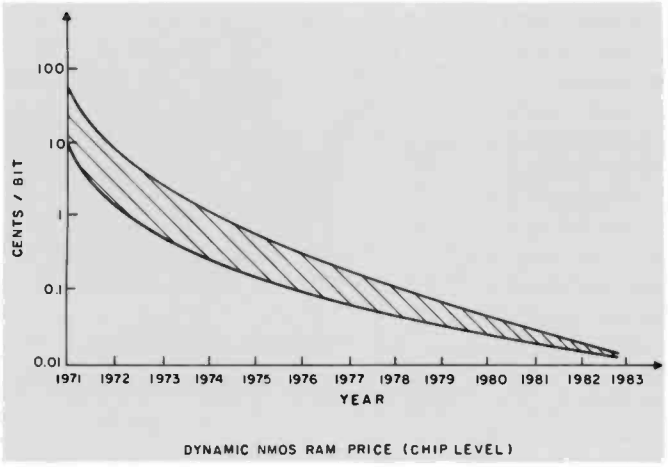


Fig. 7. Evolution in the chip level price per bit of NMOS RAM.

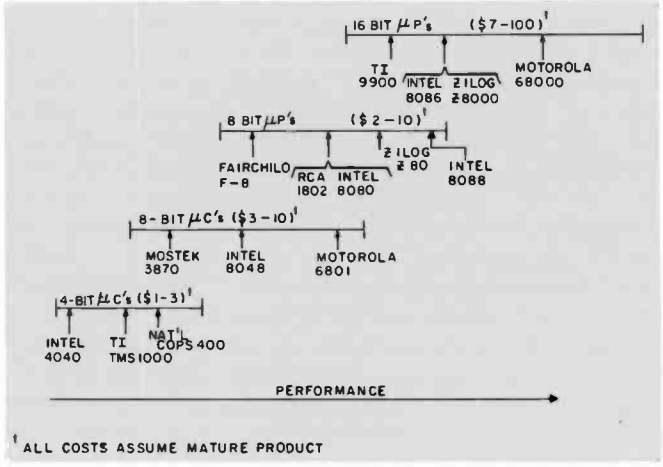


Fig. 8. Categorizing μPs and μCs.

The unquestionable technological leader in the microprocessor industry has been Intel Corporation. Figure 9 illustrates the evolution of its μP and μC families. Note the rapid improvements in technology (average transistor area) which went from 8.6 mil^2/transistor for the 8008 in 1972 to 1.8 mil^2/transistor for the 8086 in 1978. RCA's only commercially available μP, the 1802, is included for completion, along with the soon-to-be-sampled SOS 1804 μC. The average transistor area for the RCA 1804 is slightly lower than that of the Intel 8086. This is probably due to the 1804 containing a significant amount of on-chip memory (2 K bytes ROM, 64 bytes RAM), whose regular structures yield higher packing densities. Photomicrographs of the Intel 8086 and of the RCA 1804 are presented on page 4.

The microprocessor industry

The μP has changed the early trends of LSI development and has reoriented those efforts to the design of CPUs, memories, and ever more complex peripheral ICs which permit the rapid development of low chip count dedicated computer systems. As the industry matures, it is rapidly moving away from merely making ICs towards a full-system-capability industry. Intel, the leader in IC technology, is also the leader in the development of support circuits, single board computer systems, and development systems. More recently, Intel has begun to market preprogrammed devices which satisfy desired system level functions. The manufacturing cost of these devices is the same as that of unprogrammed devices, but, since there is added value in the eyes of the customer, margins can be improved.

Projecting trends

Major microprocessor and microcomputer application areas will be discussed in the following section. Here, we will attempt to summarize business trends and to project them a few years into the future. Before doing so, it is noteworthy to point out that most published sales figures and forecasts relate to non-captive markets. Captive markets (such as TI, IBM, DEC and ATT production for internal use) account for a significant fraction (close to 50%) of world production.

The market for microprocessors will have a compound growth rate of about 25% through the early 1980s, approaching an annual volume of 200 million units by 1983, with a market value approaching 500 million dollars. When one adds sales of supporting ICs such as peripherals, RAMs, ROMs, and EPROMs, the market swells to about 1.5 billion dollars per year. Figure 10 presents, roughly, how unit sales for 4-bit μCs, 8-bit μPs and μCs, and 16-bit μPs are expected to grow.[9-12]

The numbers for 8- and 16-bit devices are similar to the ones generated in a recent Creative Strategies report.[11] Figure 10, however, projects larger unit sales for 4-bit devices than does Creative Strategies (19.5 million units in 1983). It should be noted, however, that the Creative Strategies numbers are not consistent with current industry production trends. Texas Instruments, in their sales presentations, claims that it has a capacity to produce "several million" TMS 1000 machines per month. Additionally, the real volume applications for 4-bit devices are in the consumer area, and this area is just beginning to explode. The trend in this area will be to pull more system capability onto the 4-bit chip to lower system cost rather than to go for unneeded computing power represented by 8-bit architectures.

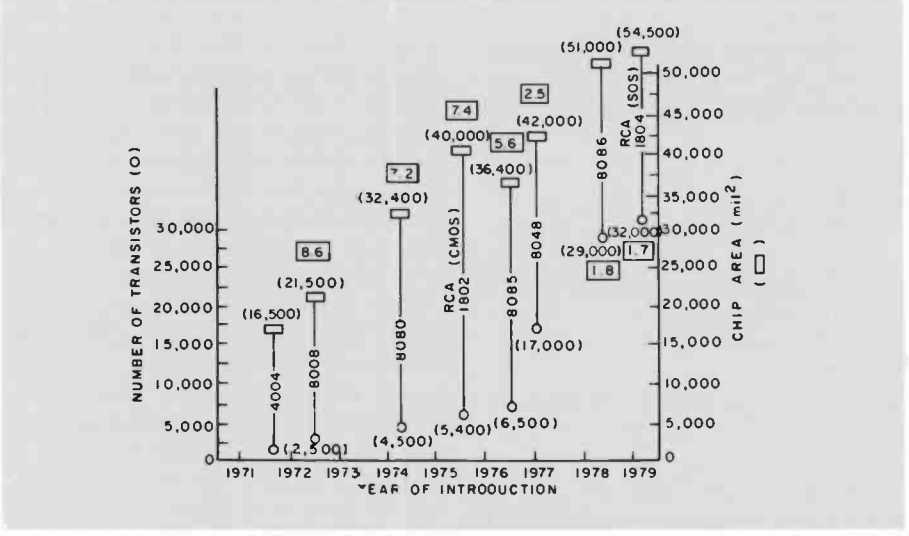


Fig. 9. INTEL and RCA microprocessor and microcomputers. (Products showing chip area and complexity evolution. □ denotes average transistor area in mil² (□/o).

Support items count in the cost

Key ingredients in the success of a microprocessor manufacturer in addition to price and delivery, are system development support and the availability of support chips and of single board computer families. The microprocessor itself usually represents a small fraction of the system LSI cost, usually less than 1% for industrial and commercial applications and perhaps 5-25% for consumer products (note that for μCs, the CPU portion usually takes up much less than half the IC area). Because of the above, microprocessor applications are *rarely* designed to maximize CPU utilization but rather to minimize system cost (development and production). Thus the ease with which one can develop a system (development support) and the availability of peripheral ICs (low system chip count) usually dominate the CPU selection process.[13]

Applications overview

The μP revolution has truly become an applications revolution. By the year 2000, 5 to 10 *billion* μPs and μCs will be in service — about one for each living person on earth. In the majority of cases, the person interacting with the μP or μC will not know that a computer is involved. One will simply have improved service or functionality for the task at hand.

The applications of μPs and μCs break down into three broad categories — industrial, commercial, and consumer.

Industrial applications are those that employ microprocessors in the design or manufacturing processes of the industry under consideration. These include process control, testing, data acquisition, numerical control, instrumentation, and robotics.

Commercial applications are those used in providing new or improved services and/or ways of doing business. These include communications (data and telephone), point-of-sale systems, intelligent cash registers and business uses (word processing, accounting, etc.).

Consumer applications are those that either add features or improve the cost/performance of existing consumer products (ignition control in automobiles, timing control in microwave ovens, etc.), or those that give rise to entirely new classes of consumer products (personal computers, programmable video games, etc.).

The following three sections briefly overview industrial, commercial, and consumer applications of microprocessors, and provide references to permit the interested reader to pursue the subjects in depth. One measure of the extent of the ongoing microprocessor application revolution is the February 1978 issue of the *IEEE Proceedings*, which was dedicated to that very subject. This special issue of the *RCA Engineer* that you are reading gives further testimony to this rapidly evolving technology.

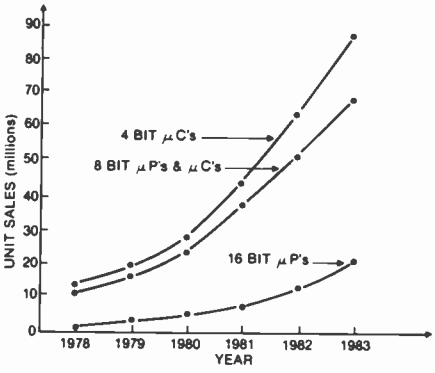


Fig. 10. Forecast of annual unit μP and μC sales through 1983.

Industrial applications

Microprocessors are having a major impact on industrial applications, including the areas of testing, control, instrumentation, data acquisition, numerical machine control, and even robotics. In all these areas, equipment offering expanded functions, better human interfaces, improved reliability, and lower cost are emerging.

Testing. Automatic Test Systems (ATS), which can be used to automatically identify component, subsystem and system faults, are becoming increasingly important from the view of reliability and maintainability. Reliability is improved with ATS since marginal components and subsystems are replaced prior to equipment shipment, resulting in a lower probability of field failure. Ease of maintenance is also improved since system-level tests can identify the probable subsystem failures, and so give rapid repair turnaround and, hence, reduced downtime.

ATS has historically been used in large military applications and in automatic testing of complex parts, where expensive test fixtures can be economically justified. The advent of microprocessors has opened many avenues for low-cost dedicated testers which means that industries can now economically justify automated and more complete testing of low-cost/low-complexity subsystems right on down to the component level.

Reference 16 describes a very successful application of RCA's COSMAC microprocessor to 100% testing of populated TV convergence boards (Fig. 11). Another example is an RCA-developed potentiometer tester.[17] Further examples of μP testing applications are presented in this issue. Many other examples can be found in the *Proceedings* of the IECI '79.[18]

Process Control and Data Acquisition. The process control and data acquisition areas differ from testing in that they must operate in real time.[19-21] This implies that the microcomputer must have time responses rapid enough to accommodate the process control or data acquisition system under consideration. Whether we are talking about controlling a power generation system or a complex chemical plant, the system must sample critical system outputs and generate suitable system inputs often enough to achieve the desired level of control.

The advent of low cost μPs means that dedicated, intelligent subsystems are now viable. For example, a μP system can be dedicated to data acquisition alone. A dedicated microprocessor can perform statistical analysis on the job, format data for more efficient off-line processing, and perform periodic self-checking and autocalibration. However, its main advantage may, in fact, be the flexibility resulting from distributed software control. Simple software changes can alter the sampling rate, specify new data formats, or alter the self-checking algorithms. This results in better human engineered outputs, modularity in hardware (improved maintenance) and slower system obsolescence.

Before the use of μPs became

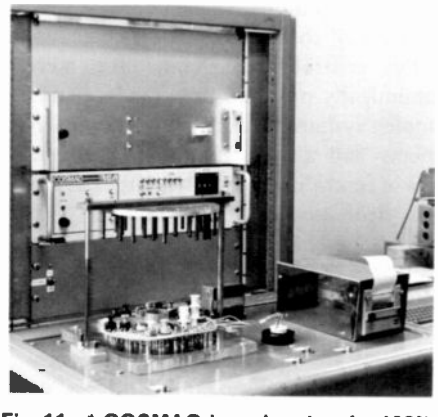


Fig. 11. A COSMAC-based system for 100% testing of TV convergence boards (on-line in Bloomington).

Careful choice of system development support items is a must.

The importance of quality system development aids cannot be overemphasized for any μP or μC to be selected by a significant fraction of users. On the software side, editors, file management systems, assemblers and high level language translators are essential. These may be time-sharing based or resident in a stand-alone development system (e.g., COSMAC CDS, or Intel MDS). On the hardware side, In-Circuit Emulators (ICE pioneered by Intel), are necessary for efficient system debugging[14], along with standard facilities such as mass memory (usually floppy disc), hard copy output, terminal input, EPROM burning facility, and standard I/O ports (e.g., IEEE 488 or RS-232).

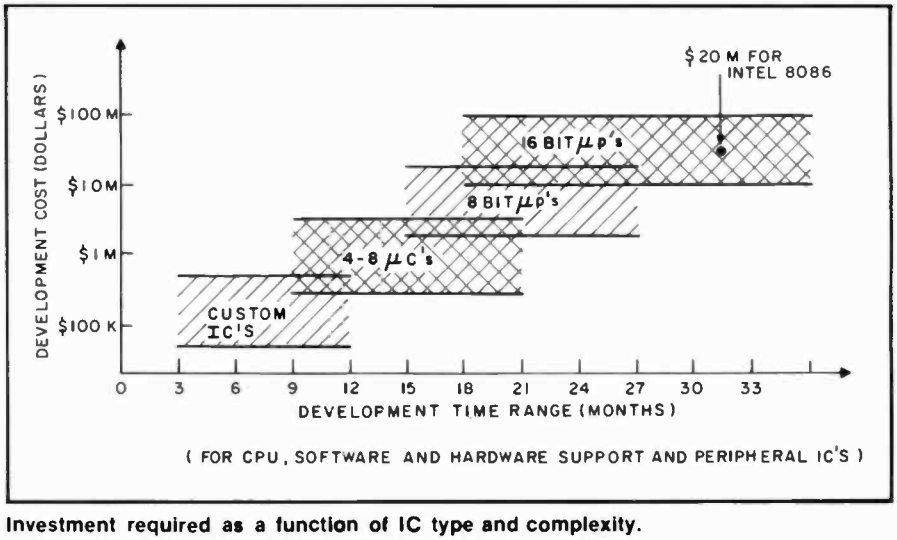
Each manufacturer of μPs or μCs provides the user with a variety of hardware and software development aids. Additionally, several independent companies are marketing general purpose development systems (Tektronix, Futuredata, Hewlett-Packard). In these systems, the user need only plug-in the appropriate characterization module, and insert the appropriate software diskette to create an environment customized to the μP/μC at hand.

Support systems options

Advantages of using the vendor's own development systems are their timeliness (they can usually support a new device soon after it is sampled), and lower cost. The key advantage of the more universal systems is that one capital investment permits the development of products using different microprocessor families. For example, the Tektronix 8002 system supports the Intel 8080/8085, the TI 9900, the Motorola 6800, the Zilog Z80, and the RCA 1802.

One may be tempted to conclude from the above that μP/μC vendors may be able to relegate the development of support systems to companies such as Tektronix or Hewlett-Packard. That conclusion, however, is a fallacy. Good development support by the μP/μC vendor will encourage the proliferation of that device family. This *latter* development is what spurs the general development system manufacturers to invest in support of a specific device family. In fact, the order in which a company like Tektronix will develop support systems for μP/μC families is closely correlated to their popularity (e.g., support on the Tektronix 8002 for the INTEL 8080 was available in 1977 whereas support for the RCA 1802 became available in October, 1979.)

Investments required to support μP and μC lines are a strong function of the device sophistication (shown above). It takes a few hundred thousand dollars to develop a custom IC. A few million are enough to support a μC family. For general purpose μPs,



Investment required as a function of IC type and complexity.

widespread, mini or large computers were the hearts of complex controllers. Because of this centralization, controllers were economically practical only for large and complex systems. Furthermore, these early systems had all the attendant disadvantages of centralized control — large cabling costs, decreased reliability, and less flexibility. Smaller systems were designed with relay logic, sequencers, or, perhaps, analog feedback control.

Through the use of low-cost microprocessors, it became possible to decentralize the control function and use intelligent controllers dedicated to specific tasks. Many advantages result from local-loop and distributed control, including lower cabling costs, reduced noise pickup, improved overall reliability (if one small system fails, the remainder keep operating), simpler maintenance, and more flexibility.

An example of a μP-based process control system is a prototype wafer silicing machine developed at RCA Laboratories. Keeping the cutting force constant, via μP control, resulted in fewer wafers being damaged.[22] Numerous other examples are presented in reference 18.

Instrumentation and IEEE Standard 488. The impact microprocessors are having on instrumentation is just beginning to be felt.[23-26] From simply adding new features to instruments (digital read outs, averaging, etc.) to auto-calibration, instruments will never be the same again.

A major factor in the increasing sophistication and flexibility of instruments has been the development of the IEEE Standard 488 Instrumentation Bus pioneered by Hewlett-Packard. The IEEE 488 Interface Bus provides a versatile effective communication link[27,28] for exchanging digital information in an unambiguous manner. It can accommodate a wide range of devices with similar protocols and is being proposed as an international standard. The number of instruments supporting the bus grows almost daily, and currently exceeds 300.

Other more specialized industrial

this number is much larger. Intel has already spent about $20 million on the Intel 8086 support. These large investments merely reflect the levels of support needed by customers to effectively use these new parts. Increasingly, the level of support and service available is given high priority in selecting a μP or μC vendor.

The single board computer

Besides system development aids, the availability of single board computer (SBC) families significantly influences processor selection. For low volume applications, development costs greatly exceed hardware costs and hence the emphasis is on minimizing design time. The availability of a standard rack with power supply, standard CPU and memory boards, and standard I/O boards (A/D, D/A, IEEE 488 bus, etc.) greatly reduces system design, testing, maintenance, and documentation costs. Examples of such families include the Intel SBC 80 line and the newer RCA Microboard family. For manufacturers such as Intel system level sales (including development systems and SBC parts) represent a significant and increasing portion of their revenue base.

applications of microprocessors abound. From energy-subsystem control, to motor-speed control, to numerical-machine control, to robotics, all are increasingly applying μPs to obtain new levels of performance at economically viable cost levels. References 29-31, along with the *Proceedings* of the IECI '79,[18] contain a wealth of information on these subjects.

Commercial applications

Commercial applications of μPs include their uses in communications (telephony and data), in medical applications and in business applications (from word processing to intelligent supermarket weighing devices).

The most significant emerging trend in communications is the switch from predominantly analog (voice) traffic to a more balanced mix between voice and data. By 1985, communications traffic will be evenly divided between voice and data. This will most certainly spur the development of all-digital terestrial channels. This, in turn, will greatly accelerate the use of distributed intelligence within the network. Intelligent multiplexers, concentrators, PABXs, modems, etc., incorporating the ubiquitous μP, will emerge. More and more, computers will take over the traditional role of operators. As all-digital networks emerge, there will be an increasing need for efficient techniques for transmitting voice in digital form. References 32-36 discuss a variety of μP applications to communications.

Microprocessors are being increasingly used in medical applications. Because of the technical and legal complexities of using new technologies in diagnosing, monitoring or treating humans, μPs today are being widely used on an experimental basis. Their routine use is at the embryonic stage of development. Reference 37 overviews the entire subject and contains a bibliography of 129 papers.

Word processing and intelligent terminals are emerging as major users of μP technology. For example, approximately 13 million Americans have a hearing or speech defect. This represents a major potential market for a portable telephone terminal. Except for the lowest end "dumb" terminal, most terminals sport some μP control. Many typewriters and other word processing machines employ μP control — not only to add capability but also for internal use to reduce cost.

One interesting example is the intelligent typewriter from the Qyx Division of Exxon Enterprises, Inc., introduced in 1978. It contains three μPs, a Zilot Z80 for word processing and master control, and a pair of Fairchild F8s for drive motor control of the carriage and the rotary print head. It offers ultra-accurate print head positioning to permit automatic erase backspace. In addition, it incorporates intelligent features such as the ability to store and recall stock phrases, to center a line of type, and to automatically line up decimal points in columns of numbers. All this capability costs under $1700.

Other commercial applications of μPs include their use in security systems, environmental control systems, and intelligent weighing devices. In the latter, for example, the store clerk need only type in "price per pound" or "price per quarter-pound," and the price will be computed automatically and displayed. A multi-microcomputer energy management system is described in this issue. Other examples include taxi meters, automated gasoline dispensing pumps, and traffic controllers. In the latter, μPs combined with sensors, can adaptively vary light sequences to maximize traffic throughput, while minimizing waiting time for individual drivers.

Consumer applications

Microprocessors and associated LSI technology are having a major impact on consumer products. From simply adding new features to standard products to making possible entirely new product categories, the consumer computer revolution is just beginning.

Microcomputer controls. Low-cost μCs are already changing a host of consumer products. From exercise and coffee machines to electric ranges, from sewing machines to microwave ovens, from white goods to home security systems, a whole new generation of intelligent consumer products is emerging. Most home appliances typically need only relatively simple controls. For that reason 4-bit devices, with their low costs, are being shipped by the millions. In fact, it can be said that the 1979-1980 timeframe represents the transition period during which it is becoming cheaper to implement timing functions with a 4-bit μC than with traditional methods. This will result in the across-the-board use of these devices in appliance control rather than only in the higher feature-oriented models.[38] The RCA Programmable TV, described in this issue, is an excellent example of how a μC can add value to a big ticket consumer product. The products discussed above were predominantly examples where electronics have replaced mechanical controls in consumer applications. More interesting, however, are the host of new consumer products spawned by the ongoing revolution in LSI.

Microcomputer-based games. Non-video microcomputer-based games are just beginning to appear, but already most of the major toy manufacturers are getting heavily involved.[39,40] Judging from the success of action games such as Parker Brothers' "Code Name: Sector," a one player submarine chase game, Milton Bradley's "Battleship," a missile-firing two-

player naval warfare game, and more cerebral games such as Mattel, Inc.'s "Football" — toys will never be the same again. A more complex example is the Chess Challenger from Fidelity Electronics (Fig. 12). The Texas Instruments' "Speak and Spell," which contains voice synthesis circuitry, is currently being shipped at a rate of about 60 K per month. The latter two products would not have been possible without μP technology.

The video game revolution began in the early 1970s on two fronts. Atari pioneered the development of video arcade games with the 1972 introduction of "Pong." That same year, Magnavox introduced "Odyssey," a consumer ball-and-paddle game based on circuitry patented by Sanders Associates.

More recently, the consumer market has begun to evolve away from dedicated games and towards microprocessor-based programmable games. Examples include the Atari VCS and the recently announced Mattel Intelvision.

Personalized microprocessors for auto and home. Automotive applications of μPs will have a major impact on the car of the future. From "under-the-hood" functions such as engine control (spark timing, fuel metering, etc.) and braking to "dashboard" functions (e.g., digital display of MPG), new levels of performance, economy, and information display will be achieved.[41-43]

Another area of major current interest is that of the home computer. There are two divergent paths in this development. One is towards the more specialized "hobby" market (e.g., RCA VIP), and the other is towards the "programmer" market (e.g., Radio Shack TRS-80). The true consumer computer product is yet to be announced although Atari (models 400 and 800) and TI 9914 are steps in the right direction. The consumer computer area is of major interest to both computer- and consumer-oriented corporations. Related developments in home data services such as Viewdata and Teletext offer intriguing potential. References 45-48 discuss the ongoing revolution in personal computing and television-related data information systems. Reference 15 is an entire issue of the *IEEE Transactions on Consumer Electronics* dedicated to consumer text display systems.

Fig. 12. Chess challenger from Fidelity Electronics.

The intelligent (programmable) thermostat is also emerging as a major consumer product. Selling in the $100-200 range to a market of 70 million U.S. homes, annual volumes of 3-5 million units will be reached in a few years. This is but one of many examples of products that were not economically feasible prior to the invention of the microprocessor.

Other consumer applications of μPs are under development, e.g., simple home control systems, electronic calendar/reminders, and wireless security systems, although it is not yet clear which of these developments will evolve into viable volume markets. Reference 44 presents an overview of the impact microprocessors are having on the consumer market.

Conclusion

The computer and LSI/VLSI revolutions are moving ahead hand in hand, and are becoming an integral part of our society. The computer industry, born only 32 years ago, has passed $50 billion/year in sales and is streaking towards the $100 billion/year mark. IBM received orders for over 32,000 System/38s, small business systems which range in price from $100,000 to $600,000 during the first year following product announcement. By 1983, the minicomputer business will grow to an annual volume of over 500,000 units per year. Marketing Development, Concord, Mass., predicts that the market for computers under $2,000 will grow from $442 million in 1979 to $1.5 billion by 1984. All of the above attests to the fact that computer technology has become inextricably entwined both in our lives and in the world economy. Reference 49 documents the pervasiveness of the computer and presents a comprehensive bibliography of its impacts.

Lower costs will impact consumer products. The semiconductor business, which is growing 15-20% annually, will reach $6 billion in sales by 1979. Explosive developments in digital LSI and VLSI are the fuel of the computer revolution. These developments led to the μP and the beginning of the second computer revolution where distributed intelligence systems will be used by the millions. Each year, the complexity of ICs doubles and the cost per function falls by about 25%. This is resulting in powerful computing elements costing but a few dollars. These costs are coming into the area required to impact volume consumer products — and the existing consumer applications are but the tip of the applications iceberg.

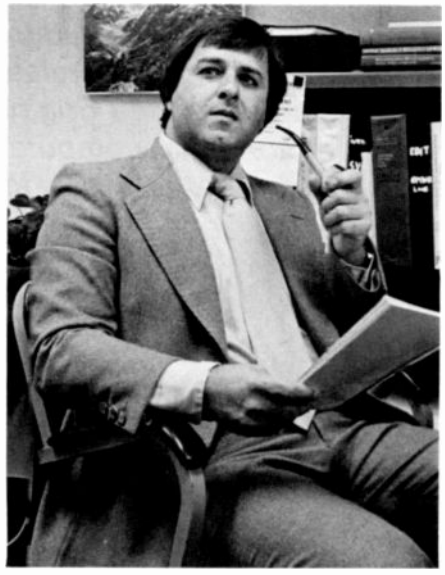
Entire LSI/VLSI systems on one chip. As LSI/VLSI technology continues to evolve, μPs and μCs will evolve in two directions. At the low end, ICs will be designed with standard cells representing entire sub-systems (e.g., RAM, ROM, CPU, A/D, D/A, etc.). Entire systems, including analog interfaces, will be put on the same chip to reduce system cost. At the high end, ever more powerful CPUs, peripheral ICs and co-processors along with ever larger and faster memories will yield exponentially increasing computing power with but a handfull of chips. The INTEL 8089 co-processor is one such example. It is built with HMOS II technology which is 30% faster and 10% denser than the HMOS technology used in the 8086 CPU.

Portability plus lower cost. Increasingly, especially in μC-based consumer applications, low power dissipation and product portability will become realistic design goals. The potent mix of liquid crystal displays (LCDs) with CMOS circuitry will yield an array of portable game, computer and other intelligent products. No longer will you have to leave your computer chess game behind when you board the train or hit the beach! Though CMOS technology trails NMOS in density by about two years, the price differential for equivalent parts will decrease to small levels as we move down the learning curve. For this reason, the current exploding usage of 4-bit NMOS μCs will be duplicated by CMOS devices in the next few years. Furthermore, since the bulk of true volume applications does not need 8-bit CPU throughput, the trend will evolve towards pulling more and more I/O onto the chip resulting in lower system cost.

From microprocessor-based automobile ignition systems to intelligent thermostats, from hand-held computer games to consumer computers, from programmable TVs and VCRs to microwave oven controllers, we are witnessing the beginning of a consumer applications revolution. With over 150 μPs and μCs to choose from, the real challenge is in the development of new products and uses that fully exploit the potential of LSI and VLSI technology.

References

1. von Neumann, J., "First Draft of a Report on the EDVAC," University of Pennsylvania, Philadelphia (June, 1964).
2. "Here Comes the Second Computer Revolution," *Fortune* (Nov., 1975).
3. "The Computer Society," *TIME* (Feb., 20, 1978).
4. Russo, P.M., "Architectural Trade-Offs in Designing Microcomputer Systems," *RCA Engineer* (Feb./March, 1977).
5. Peatman, J.B., *Microcomputer-Based Design*, McGraw-Hill, New York (1977).
6. Russo, P.M., "Interprocessor Communication for Multi-Microcomputer Systems," *IEEE Computer*, pp. 67-75 (April, 1977).
7. Noyce, R.D., "From Relays to MPUs, *IEEE Computer*, pp. 22-29 (Dec., 1976).
8. Moore, G "VLSI: Some Fundamental Challenges," *IEEE Spectrum* (April, 1979).
9. *Electronics*, p. 90 (May 10, 1979).
10. "U.S. Markets Forecast 1978," *Electronics*, p. 134 (Jan. 5, 1978).
11. "The Microprocessor Industry," *Creative Strategies International* (May, 1979).
12. "U.S. Markets Forecast 1979," *Electronics*, p. 114 (Jan. 4, 1979).
13. "Peripheral Chips Shift Microprocessor Systems into High Gear," *Electronics*, pp. 93-106 (Aug. 16, 1979).
14. Kelley, J.M., "Cut Hardware, Software Development Costs — Take Advantage of In-Circuit Emulators," *Electronic Design*, pp. 66-71 (Aug. 16, 1979).
15. "Consumer Text Display Systems," Special Issue on, *IEEE Transactions on Consumer Electronics* (July, 1979).
16. Marcantonio, A.R., "Microprocessor-Based Printed Circuit Board Tester," *Proceedings of IECI* '78, Philadelphia, Pa. (March 20-22, 1978).
17. Marcantonio, A.R., "Microcomputer-Controlled Potentiometer Test System," *Proceedings of IECI* '79, Philadelphia, Pa., pp 308-311 (March 19-21, 1979).
18. *Proceedings of IECI* '79, the IEEE Industrial Electronics and Control Instrumentation Society's Annual Conference on Industrial and Control Applications of Microprocessors, Philadelphia, Pa. (March 19-21, 1979).
19. Wang, C.C., Wu, C.T., Russo, P.M., Abramovich, A., and Marcantonio, A.R., "Microprocessors in Manufacturing and Control," *RCA Engineer*, special issue on microprocessor technology (Feb./March, 1977).
20. Wittke, J., and Simpson, T.F., "Microcomputers in Picture Tube Manufacturing," *RCA Engineer* (April/May, 1978).
21. Abramovich, A., "An Interpolating Algorithm for Control Applications on Microprocessors," *Proceedings of IECI* '78, Philadelphia, Pa (March 20-22, 1978).
22. Van Le, D., Demers, R., Etzold, K.F., and Firester, A.H., "Wafer Slicing with Microprocessor-Based Controller," *Proceedings of IECI* '79, Philadelphia, Pa. (March 19-21, 1979).
23. Randle, W.C., and Keith, N., "Microprocessors in Instrumentation," IEEE Proceedings, special issue on microprocessor applications (Feb., 1978).
24. Wu, C.T., "CVM — A Microprocessor-Based Intelligent Instrument," *RCA Engineer*, special issue on microprocessor technology (Feb./March, 1977).
25. Oliver, B.M., "The Role of Microelectronics in Instrumentation and Control," *Scientific American* (Sept., 1977).
26. "Instruments '79," Special Report, *Electronic Design* (Nov. 22, 1978).
27. Loughry, D.C., and Allen, M.S., "IEEE Standard 488 Microprocessor Synergism," *IEEE Proceedings*, special issue on microprocessor applications (Feb., 1978).
28. "IEEE Standard Digital Interface for Programmable Instrumentation," *IEEE*, IEEE Std. 488, (1975).
29. Sohrabji, N., "Microprocessors Extend Scope of Automated Manufacturing," *EDN* (March 5, 1978).
30. Doi, Y., "Robots Get Smarter and More Versatile," *IEEE Spectrum* (Sept., 1977).
31. Goksel, K. and Parrish, E.A., Jr., "The Role of Microcomputers in Robotics," *Computer Design* (Oct., 1975).
32. Lippman, M.D., "Microprocessors in Data Communications," *RCA Engineer*, special issue on microprocessor technology (Feb./March, 1977).
33. Stanzione, D.C., "Microprocessors in Telecommunications Systems," *IEEE Proceedings*, special issue on microprocessor applications (Feb., 1978).
34. Gundlach, R., "Large-Scale Integration is Ready to Answer the Call of Telecommunications," *Electronics* (April, 1977).
35. Melvin, D.K., "Microcomputer Applications in Telephony," *IEEE Proceedings*, special issue on microprocessor applications (Feb., 1978).
36. Mayo, J.S., "The Role of Microelectronics in Communication" *Scientific American* (Sept., 1977).
37. Klig, V., "Biomedical Applications of Microprocessors," *IEEE Proceedings*, special issue on microprocessor applications (Feb., 1978).
38. Walker, G.M., "LSI Controls Gaining in Home Appliances," *Electronics* (April 14, 1977).
39. Rosenblatt, A., "Electronic Games No Longer Need TV," *Electronics* (Aug., 18, 1977).
40. Mennie, D., "Self-Contained Electronic Games," *IEEE Spectrum* (Dec., 1977).
41. Robbi, A.D., and Tuska, J.W., "A Microcomputer-Controlled Spark Advance System," *RCA Engineer*, special issue on microprocessor technology (Feb./March, 1977).
42. Lile, W.R., and Tuska, J.W., "Electronic Control of Vehicle Brakes," *RCA Engineer*, special issue on microprocessor technology (Feb./March, 1977).
43. Marley, J., "Evolving Microprocessors Which Better Meet Needs of Automotive Electronics, *IEEE Proceedings* (Feb., 1978).
44. Russo, P.M., Wang, C.C., Baltzer, P.K. and Weisbecker, J.A., "Microprocessors in Consumer Products," *IEEE Proceedings* (Feb., 1978).
45. Key, A.C., "Microelectronics and the Personal Computer," *Scientific American* (Sept., 1977).
46. Doerr, J., "Low-Cost Microcomputing: The Personal Computer and Single-Board Computer Revolutions," *IEEE Proceedings* (Feb., 1978).
47. McKee, K.M., "Data Television — A Bright Star on the Color TV Horizon, *RCA Engineer* (April/May, 1979).
48. Lipoff, S.J., "Mass Market Potential for Home Terminals," IEEE Chicago Fall Conference on Consumer Electronics (Nov. 6-7, 1978).
49. Russo, P.M. and Wang, C.C., "The Pervasive Computer," *RCA Engineer* (April/May, 1978).



Paul M. Russo is Head of Microsystems Research, RCA Laboratories. During the 1969-70 academic year he was on the faculty at Berkeley, California, where he developed and taught courses in circuit theory and computer-aided-design. He joined RCA Laboratories in 1970 as a member of the technical staff, was appointed Head, TV Microsystems Research in 1977. He has been involved with microprocessors since their inception, and has contributed to the early efforts that led to the RCA 1802 product and to the use of microprocessors in a variety of current and future RCA products and systems. Dr. Russo was a guest co-editor of the February, 1978 issue of the *IEEE Proceedings* special issue on microprocessor applications, and is the General Chairman, of IECI '80, an annual conference on industrial, control and instrumentation applications of mini and microcomputers to be held in Philadelphia, Pa., March 17-20, 1980.

Contact him at:
RCA Laboratories
Princton, N.J.
Ext. 3234

P.P. Fasang

A microcomputer glossary

Computer technology is creating a new language that not only incorporates new terms, but often redefines standard English. Words, such as address, bit and stack, assume extended meanings that relate to computer technology, while maintaining their original connotations. A few of the most common microcomputer terms are defined for the layman and, as a refresher, for the engineer.

access time: the time between a request for information from a storage medium and the time the information is available.

accumulator: register(s) which contain results of the arithmetic/logic unit operations.

address: a computer word used for designating a specific location in memory.

A/D converter: (analog-to-digital converter) an electronic device for changing a DC voltage to a binary-coded value. Computers cannot process a continuous voltage waveform but can process binary value.

addressing mode: techniques for specifying memory locations for the purpose of storing and/or retrieving information.

ALU: (arithmetic/logic unit) the hardware portion of a computer which performs arithmetic functions, such as addition and subtraction and logic functions such as shift, AND and OR.

architecture: the logical organization of the hardware portion of the CPU.

Reprint RE-25-3-2
Final manuscript received Sept. 27, 1979.

assembler: a program which translates application programs written in English-like symbolic language (assembly language) into machine language (binary).

assembly listing: a listing which shows the assembly-language program and the translated machine-language program.

ASCII: (American standard code for information interchange) a 7-bit code for representing the English alphabet, numbers, and special symbols, such as $, ↑, and ←.

baud: a data transmission-rate unit. For most applications, a baud is equal to one bit per second.

BCD: (binary coded decimal) a coding scheme for representing the ten decimal numbers.

binary number: a number whose digits are either zero or one. Computers can "read" and "write" only binary numbers.

bit: a binary digit.

bit-slice microprocessor: an *n*-bit wide processing element usually connected in parallel to implement a microcomputer of *n*-bit word length. The instruction set is customer defined (by a micro-code).

buffer: a register for holding temporary data.

bus: a set of conductors which carries all necessary computer signals, such as data, address, and control signals.

byte: a computer word of 8 bits.

cell: a memory bit.

CMOS: (complementary metal oxide silicon) a very low power logic technology.

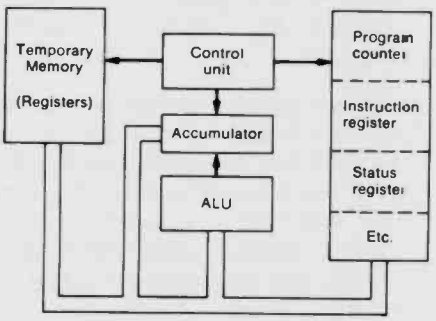
CPU: (central processing unit) the arithmetic/logic unit (ALU), registers, and control circuits of a computer. The CPU decodes instructions, issues timing signals, and performs all control functions.

cross assembler: an assembler for use in a computer with an instruction set other than the one which the application program is written for.

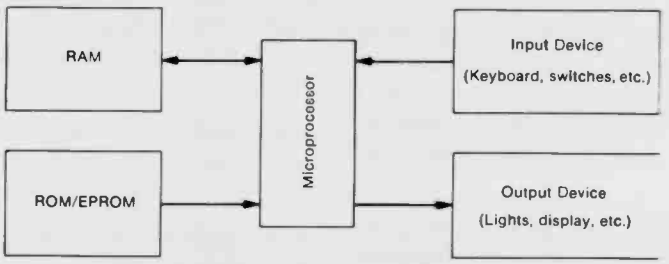
cycle time: the shortest period of time at the end of which a sequence of events repeats itself. Some of the events may be retrieving a word from a memory cell, interpreting the meaning of the word, and executing the instruction.

The microprocessor is the "brain" for the microcomputer.

A basic microprocessor contains an ALU which performs arithmetic and logic functions, an accumulator which contains the results of the ALU operations, a control unit which generates timing and control signals, a program counter which holds the address of the next instruction to be executed, and other registers. Although a very complex integrated circuit, by itself, the microprocessor is not useful. To realize its potentials, the microprocessor must be augmented with other necessary support circuits. A microcomputer system makes possible the realization of the potentials of a microprocessor.



A basic microcomputer system



Block diagram of a basic microprocessor

In the basic microcomputer system, the microprocessor serves as the "brain" of the system, RAM allows information to be stored as well as to be retrieved. ROM/EPROM allows information to be retrieved (read) only, the input device allows information to be entered into the system, and the output device allows information from the computer to be displayed or printed.

Information stored in the ROM/EPROM is information that requires no alteration by the computer program and generally this information is a program. One advantage of storing information in a ROM/EPROM is that unauthorized people cannot tamper with the program.

Hardware such as A/D converter, D/A converter, drivers, and other circuits may also be needed in a real-world application. These and other hardware are incorporated into the system depending on the specific application at hand. Because of their low costs, microprocessors are ideal for dedicated systems which require "intelligence."

D/A converter: (**d**igital-to-**a**nalog converter) a device which changes a binary signal into a dc voltage level.

decode: to translate or interpret a computer word into something familiar or useful for performing tasks.

display: a device which shows a letter or a number or simply emits light; as a miniature light bulb.

DMA: (**d**irect **m**emory **a**ssess) a technique for receiving information from or transferring information to the main memory of a computing system without having the CPU involved.

drivers: circuits which increase the driving capability of an output circuit.

encoder: a device for translating a one-bit signal into a multibit signal.

editor: a computer program which aids a programmer in his source program creation. An editor helps to perform the following functions: typing in program, making corrections, assigning line numbers to all lines, resequencing of line numbers, locating selected characters, and listing of partial or entire program.

EPROM: (**e**rasable **p**rogrammable **r**ead **o**nly **m**emory) a type of non-volatile random-access memory chip which can be programmed in the field.

execute cycle: the amount of time for an instruction, that has been fetched, to be executed.

fetch cycle: the amount of time for an instruction or data to be retrieved from a memory location.

flag: a bit in a register which is used to keep tract of the state of an input/output device, the state of a register, or the state of the microprocessor. For example, a flag can be used to keep tract of the state of the interrupt feature by setting the flag to 0 for an uninterrupted state and to 1 to denote an interrupted state.

firmware: software that is stored in random-access read-only memory.

handshaking: a predefined procedure for sending information from a terminal (sender) to a computer (receiver) or vice versa that informs the sender that the receiver is ready to receive.

hardware: computer circuits and peripheral devices.

hexadecimal: a number with a digit value ranging from 0 to 15. The 16 hexadecimal numbers are: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F.

high-level language: a computer language such as FORTRAN or BASIC which resembles English. One high-level language instruction is translated by a compiler or interpreter program into several machine-language instructions.

The author wishes to acknowledge the assistance of K.R. Wurtzel who read the manuscript and made many valuable contributions.

IC: (Integrated circuit) very small electronic circuits contained in a single package which perform sophisticated functions.

interface: as a noun, a physical circuit or subsystem which allows two different types of circuits or systems to be connected together. As a verb, to perform the above-mentioned function.

index register: a register whose function is to store a number which is used as a pointer to reference a parameter.

instruction: the machine language words in a computer program that tell the computer what to do.

I/O device: (Input/output device) an input device or an output device: Examples of input devices are keyboards, sensors, and switches. Examples of output devices are displays, audio indicators, and X-Y plotters.

interrupt: a request to the computer to service an external device. The external device can get the attention and the service of the computer by sending an interrupt signal to the computer. The computer will first finish the execution of the current instruction and then it will service the external device.

keyboard: portion of a terminal that is used to input information to the computer. A terminal has two separate logical entities; one is the keyboard which is used to input information to the computer. The other is the printer or display.

LED: (light emitting diode) an electronic device which sends out light when it is turned on.

micro-code: the logic-level definition of the instruction set of a bit-slice microcomputer or similar type of machine. Some microprocessors have a micro-coded architecture.

microcomputer: a computer whose CPU is a microprocessor.

microprocessor: a micro-electronic chip which contains an ALU, registers, input/output ports, and control and timing circuits. It is capable of performing arithmetic and logical operations.

mnemonic: the short-hand symbolic names or abbreviations which have pre-defined meanings, and which represent instructions in assembly language.

multiplexer: a circuit which performs path selection function so that a computer can "talk" to several external devices one at a time.

object codes: the binary codes which are obtained as a result of the translation done by an assembler.

octal number: a number with a value ranging from 0 to 7.

operand: the number which follows an instruction. Example: *LDA 5. LDA* is a mnemonic which represents an instruction that tells the computer to load the operand into the accumulator. The operand in this case is 5. When this instruction is executed, the number 5 is loaded into the accumulator.

page: a number of consecutive memory locations, nominally 256. If the memory of a computer has 4096 locations, the computer has 4096 ÷ 256 = 16 pages of memory.

parity bit: a bit which is used to detect a transmission error.

pointer: an address which corresponds to the beginning address of a program or table.

PROM: (programmable read-only memory) a random-access type of memory which can be programmed once only with a PROM programming machine.

RAM: (random access memory) memory devices which can store and retrieve information in any location in an amount of time which is independent of the memory location selected.

ROM: (read only memory) memory devices which allow only retrieval of information. The information in the ROM is stored during manufacture before the ROM is put in operation. The computer can only read information out of the memory device, therefore, altering the information by the program is not possible.

register: an electronic device for storing information. In addition, a register may perform other functions such as shifting of bits, selectively clearing some bits, and selectively setting some bits.

resident assembler: an assembler that runs in a system that executes the object codes which are generated by that assembler.

sequential memory: memory in which information storage and retrieval is done in a sequential manner. For example, to read the contents of location 50, the computer must search through locations 00 through 49. It cannot jump from location 00 to location 50.

software: computer programs which are written using instructions, operands, and labels in a logical fashion to accomplish the required task.

source code: the program as it appears in high-level language or in assembly language (as opposed to machine language).

stack: storage and retrieval of information in a first-in-last-out fashion, i.e., the word that is stored first will be the word which is read out last. The term also refers to a collection of sequential memory cells which store and retrieve information in a first-in-last-out fashion.

stack pointer: a register whose contents correspond to the address of the top of the stack.

table-driven: a technique of branching to different subroutines. The starting address of a subroutine is contained in a table and in order to branch to a subroutine, the computer must first go to the appropriate location in the table.

terminal: an input/output device through which a person can "talk" to a computer.

tristate: a type of logic circuit whose output can assume three states: a logical zero, a logical one, or a high impedance state. In the high impedance state, the operation of other tri-state devices connected to the same point is uneffected.

word: a group of bits (the most common are 8, 16, 32, or 64, depending on the computer) which comprises a single unit of information.

Pat Fasang's biographical sketch and photograph appear with his article on microprocessor education programs on page 71.

B. Astle|M. Lurie|K. Schroeder

The "berry picking" approach to computer systems development

A successful development strategy for automating the TK-47 television camera.

Abstract: *Development of computer systems requires extensive interaction among those involved, unlike picking berries. Berry picking can be accelerated in proportion to added manpower, but computer systems development cannot. It is also difficult to expand the scope of a project while maintaining the original time schedule. If these goals must be accomplished, then innovative strategies are required. The strategy illustrated here is an example of how the bricklaying method worked in solving an automation problem.*

The example which will be used to demonstrate one useful technique to accomplish this is the history of the development of the RCA TK-47 professional color TV camera. This project was expanded early in its development to include automation of the camera set-up adjustments, while retaining the original schedule. The strategy used to accomplish this expansion was to partition the automation feature into an autonomous processing element (microcomputer) and to interface it to the existing design by means of a simple interface. This approach successfully met the time schedule, demonstrating its feasibility.

Projects can be divided into several categories. A perfectly partitionable project is one which can be partitioned into tasks requiring little or no communication among the workers. Progress in such projects is essentially proportional to manpower. Examples, given in reference 1, are reaping corn and picking cotton. At the other extreme is the unpartitionable project which takes a certain time no matter how many people work at it. An example of such a project is tuning a piano. It is not likely that two piano tuners could cooperate effectively on one piano and finish in less time. In fact, the opposite is more probable. Most projects fall somewhere between these two extremes. Some number of people can work together effectively, as shown by the minimum in the middle curve in Fig. 1, but there is a level of manpower beyond which further increases do not help. In fact, it is possible that these additional people will serve only to delay the completion of the project, as indicated by the rising portion of the curve.

Unlike either berry picking or piano tuning, computer systems development is a complex set of interdependent software and hardware tasks requiring manpower to perform the tasks, a plan to coordinate them, and the tools to support their execution. The planning of such an effort can itself be a complex process. It involves the enumeration of the system development tasks, specifying both the time and resources required to accomplish each of the tasks and their appropriate sequencing. This plan often is represented on charts such as a PERT chart that specifies the magnitude and skills of the manpower required and identifies the serial or parallel relationship between the tasks.

Nonlinear effects of increased manpower in computer systems

The use of planning tools like a PERT chart is prone to several common misconceptions. First, there is the belief that all the tasks will actually follow the predicted schedule. Maybe this misconception exists because of the tendency to believe what we read. Certainly it is because we believe that all will go well, or that tasks will only take as long as they ought to take.[1] Anyone familiar with Murphy's Law knows this is certainly not true.

A second common misconception is the belief that effort is equivalent to progress. Although cost is usually proportional to effort, progress often is not. This is partially due to the tremendous differences in individual productivity and to the level of success of the organization of the effort. The larger the work force the more critical its orchestration.

Another common misconception is the belief that people and months of effort are interchangeable in achieving progress; e.g., if one person can do a project in six months, then six could do it in one month. This belief is not even approximately true for software development where addition of manpower may not decrease the completion time at all, as indicated by the middle curve in Fig. 1.

There are several reasons for the lack of interchangeability of people and months in a computer systems development project. Perhaps the most important reason is the extensive interaction required among the participants. The amount of communication among them is almost proportional to the square of their number (N). More explicity, the number of interfaces is $N(N-1)/2$. If each interface requires a time t per month of each of two

Reprint RE-25-3-3
Final manuscript received May 25, 1979.

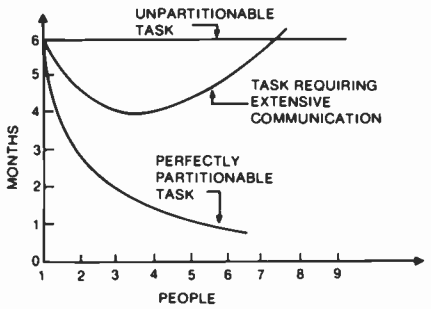


Fig. 1. There is a level of manpower, in a system, beyond which further increases do not help.

participants, and each contributes a total time T per month, then the effective work time per month is $NT-N(N-1)t$. This reaches a maximum when $N=(T+t)/2t$. At this point each worker is spending almost half his time communicating. Of course, this simple model, in which the interface time is independent of the total number of workers, does not normally apply, but it indicates the problems inherent in some types of project organizations.

Another reason for the nonlinear effects encountered when using increased manpower on such projects is increased overhead due to: more people potentially leaving during the life of the project, bottlenecks in the access to a limited number of development tools, difficulty in communication due to the size of project meetings, general increases in management overhead requirements, and delay due to people already working on the project having to train the newcomers.

A consequence of the erroneous belief in the man-month equivalency myth is the tendency to address either slippages in schedule or increases in project scope by adding manpower. This rarely works in conventional project approaches. In fact, Brooks states that adding manpower to a late software project makes it later.[1] It is clear that an unconventional method is required if additional manpower is to be effectively used to enhance or speed up development projects.

The "berry picking" approach, combining autonomy with interaction

We would like to report a method by which one can avoid the communication dilemma and effectively add manpower to an ongoing systems development project. For want of a better name we shall call the method "the berry picking approach," since its objective is to limit the required communication overhead, approximating that activity's success at accelerating progress in proportion to added manpower. To achieve this objective usually requires organization of the project so that the additional system functions can be isolated into autonomous subsystems reducing the communication overhead.

The berry picking approach can be used to shorten the development time normally required for a system. It can also be used to increase the scope of an ongoing systems development without significantly impacting the completion schedule. In both cases, more manpower will be required than would normally be considered optimal when using conventional techniques, but the berry picking technique succeeds where conventional ones often fail.

To use this technique, the system must be structured in a top-down fashion where the functions that are to be implemented autonomously are only loosely coupled to the other functions within the system. Coupling is a measure of the strength of interconnection between the functions, or software modules, within the system.[2] If two tasks within the system are loosely coupled, then one can design, implement and debug one function without knowing very much about any other function within the system. Also, making a change to one function within such a system has little probability of requiring a change to any other. Loosely coupled tasks tend to have simple interfaces which contain little data and almost no control information. This, however, is an ideal which cannot always be attained.

Thus, our technique partitions the system into loosely coupled pieces which require little communication to implement. We distribute these pieces among a larger work force (maybe even in different locations) without encountering the potentially massive overhead predicted by Brooks.[1]

This technique requires extra hardware, but has been made economical in many cases by the availability of inexpensive microcomputers.

A case history

Let us now examine a case history which illustrates the effectiveness of the berry picking approach. This is the development of the RCA TK-47 color TV camera with automatic setup, i.e., automatic adjustment of the many controls that are needed to produce a good color picture.

The TK-47 control functions

The camera head contains three pickup tubes upon which are focused the red, green and blue images. The video signals from the three tubes are processed and combined to produce a composite color signal. In order that the final image, as viewed on a monitor, resemble the original scene as closely as possible, it is necessary that the three scanned images be accurate in an absolute sense (no distortion). They should also correspond to one another as closely as possible at every point with respect to their geometry and signal levels. Geometrical errors would produce spurious colors at the edges of objects and, in extreme amounts, would give visible distortion. Level errors would produce color errors that shift as the level of illumination changed.

To minimize these errors and to produce the highest quality video output, approximately 100 setup control functions are provided on the camera. These setup functions control such things as the shape and size of the raster scans, and the levels and gains in different parts of the images. Many of the controls have nonlinear effects, and many of them interact with each other.

The TK-47 is digitally controlled. The values of the control settings are stored digitally in the camera processor, which can be physically remote from the camera head, but is connected to it via a cable as indicated in Fig. 2. To change the values of the stored control settings, a setup control unit (SCU) is connected to the camera processor. The original version of this SCU had provision to alter any of the stored digital control values manually. The automatic version was conceived to be basically a manual unit with an added option to make the setup process completely automatic.

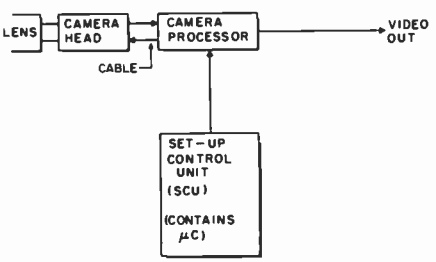


Fig. 2. The camera processor can be physically remote from the camera head, but is connected to it via a cable.

Manual control

In the manual version, the control functions are presented in groups of from one to four to the user who must be an experienced camera technician. He can select the controls he is adjusting by pressing buttons on the panel of the SCU. The microcomputer inside the unit then configures the system accordingly and displays the functions of the control knobs above each active knob. In addition, various helpful messages can be displayed.

The setup procedure consists of selecting a series of control functions and adjusting them to optimize the image. Unfortunately, some functions that are performed at different times interact and so have to be repeated, often many times. Other functions interact directly, but the interactions are so complex that it is not obvious, even to the skilled technician, how to attain the optimum setting of the control. For example, there are ten controls for each color which affect the registration of the images: horizontal and vertical size, centering, skew, bow and linearity. It is not obvious how to adjust them to reduce an error in one corner. Thus, the manual setup procedure is quite costly since it requires experienced personnel and is inherently time consuming.

Automatic control

The advantages of automating this procedure are to save time, to increase the uniformity of the setup from day to day and from camera to camera, to increase its accuracy, and to reduce the requirement for skilled personnel.

The automation can be accomplished by adding a video detector to the system as indicated in Fig. 3. This video detector is under the control of the SCU and measures different image parameters when the camera is optically focused on a special chart, or alternatively on an illuminated test slide within the lens. The SCU uses these measurements to calculate the various control errors.

Extensive software must be added to the manual SCU to convert it to an automatic version. In order to assess the extent of this additional software, let us examine the automatic set-up procedure in more detail. When the operator initiates the automatic set-up procedure, the SCU initializes the hardware detector and puts it into the mode for detecting the appropriate information from the chart image. The detector may be required to detect the video levels at various points on the chart, or the horizontal and vertical positions of various reference marks on the chart, or the amplitude of the grating patterns to determine focus. This information is then processed by the microcomputer and the various image errors calculated. These errors are then converted to the camera control adjustments required to correct the errors and are transmitted to the camera head. The process is repeated until the image errors are minimized, and then the next function is tackled. The process continues until all types of image errors have been minimized. Then control is returned to the operator. The SCU and automatic unit may now be disconnected from the camera processor and the camera operated in its normal mode.

Segregation of manual and automatic controls

As you can tell from this necessarily brief description, there is extensive interaction between the automation of the setup functions and the rest of the system. The automation functions must parallel the manual ones. They must be designed in conjunction with the detector hardware and must interface intimately with the manual setup program. The methodology of the adjustments must be closely linked to the camera design and should reflect, to some degree, the expectations of experienced camera technicians.

When the automation project was begun in earnest, the manual setup system was still being developed. To have added manpower to the project to develop the automatic system, in a conventional way, would have delayed the completion until long after its scheduled date, for the reasons previously stated. Furthermore, additional manpower was available only at a different location.

The solution to the problem of developing the automation function without delaying the project was to use the berry picking approach. This was accomplished by severing the automation of the setup in its entirety from the manual camera system, and then adding as much duplicate power to the automatic system as was necessary to minimize the communications interface between the two subsystems. This meant adding an additional microcomputer and building two interfaces together with a communications link, as indicated in Fig. 4. The advantages were that development of the two subsystems could proceed almost independently, and that most of the

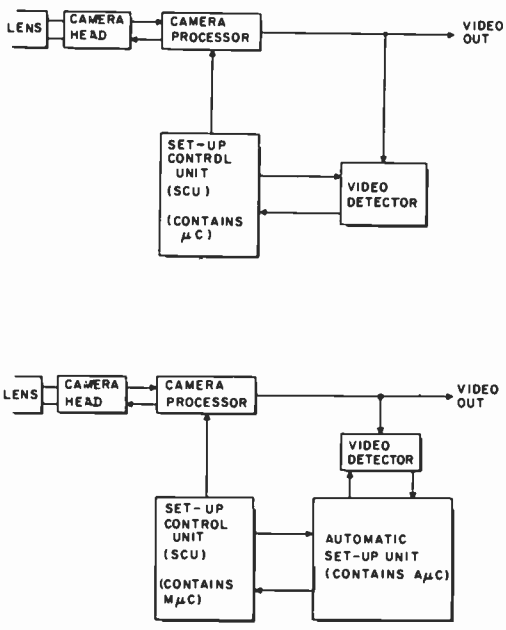


Fig. 4. To segregate the automation system, an additional microcomputer and two interfaces with a communications link were added.

problems associated with the development of the video detector hardware and automation software could be solved without affecting the ongoing manual system development.

The crucial point of the berry picking approach is to subdivide the project into tasks so as to minimize coupling. The first step in partitioning the manual and automatic operations into two loosely coupled subsystems was to separate those functions that were logically and physically related to the camera itself from those that were solely for automation. The manual microcomputer (MμC) became the controller for communication between the camera and the automation microcomputer (AμC). The automation functions for the MμC were placed in a separate module of its program, reached only when the camera is in its automatic mode to minimize coupling to the existing manual system. The AμC and the video detector perform all the measuring and calculating functions, and determine the required camera adjustments independently of the MμC.

For this approach to be successful, the communication interface must be very carefully and clearly defined. We established immediately that the MμC is in control, with the AμC responding to commands. Communication between the two microcomputers is on a schedule synchronous with the TV field rate (60 Hz). At the beginning of each field, the MμC interrupts the AμC and sends a command

Why was the tower of Babel an engineering fiasco?

"Now the whole earth used only one language, with few words. On the occasion of a migration from the east, men discovered a plain in the land of Shinar, and settled there. Then they said to one another, 'Come, let us make bricks, burning them well.' So they used bricks for stone, and bitument for mortar. Then they said, 'Come, let us build ourselves a city with a tower whose top shall reach the heavens (thus making a name for ourselves), so that we may not be scattered all over the earth.' Then the Lord came down to look at the city and tower which human beings had built. The Lord said, 'They are just one people, and they all have the same language. If this is what they can do as a beginning, then nothing that they resolve to do will be impossible for them. Come, let us go down, and there make such a babble of their language that they will not understand one another's speech.' Thus the Lord dispersed them from there all over the earth, so that they had to stop building the city.

Genesis 11:1-8

Frederick P. Brooks, Jr., in *The Mythical Man-Month* tells us that the tower of Babel, which according to Genesis was man's second engineering undertaking after Noah's ark, failed for two reasons—communication and organization. Lack of communication led to poor coordination and, hence, failure in the engineering project. Although the people had a clear mission, plenty of manpower, sufficient materials, abundant time, and adequate technology, they were unable to complete their project due to poor organization.

Brooks states that, in a large programming project, "if there are n workers on a project, there are $(n^2-n)/2$ interfaces across which there may be communication, and there are potentially almost 2^n teams within which coordination must occur." Since the purpose of organization is to reduce the amount of communication and coordination necessary, poor organization must lead to project failure and good organization will determine its success.

Communication reaches a low overhead with a division of labor and a specialization of function. Using a tree-like structure of organization that incorporates division and specialization of labor will result in a diminishing need for detailed communication.

Once communication has been reduced to a minimum, the need for effective communication must be recognized. Three essential communication approaches for the programming project are: (1) informal exchange of information; (2) regularly scheduled project meetings; and (3) the use of a formal project workbook.

Although the engineers of the tower of Babel failed to complete their project when they lost the ability to communicate, the efficiency of a software system development can be increased by reducing communication needs to a low level through division of labor and specialization of function by task partitioning coupled with loose interfacing.

B.L.S.

telling it what to do next. This command might be to begin a particular function or to continue a calculation in progress. At the end of each field, the MμC inputs data via direct memory access (DMA) from the AμC memory. The data includes a status word and calculated camera adjustments. Using DMA and interrupt in this way, programming of the extensive calculations in the AμC proceeded without concern for interaction with the camera. The RCA COSMAC 1802 microprocessor is well suited to that particular mode of operation because of its onboard handling of DMA.

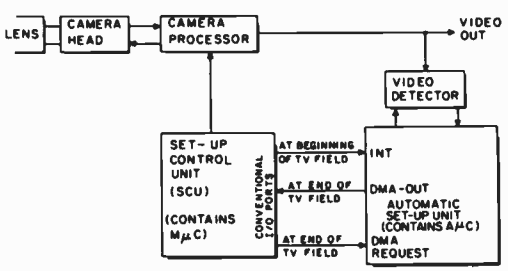


Fig. 5. After segregation of the automatic and manual systems, the systems were loosely coupled to allow communication across the partitions.

When the AμC receives a command to make a measurement, it outputs a conditioning command to the video detector and then inputs the results of the measurement. Any required conditioning of the camera head or processor is performed by the MμC maintaining the established communication discipline. Figure 5 illustrates the communications paths.

Conclusion

Faced with the problem of expanding a system development program with a fixed deadline, we took the approach of partitioning the project into two groups of tasks with minimum coupling, using separate microprocessors for each, and establishing simple, well-defined interfaces. This allowed additional people to work on the new parts of the system without the pitfall of reducing everyone's output because of interaction, communications and additional management requirements.

This article is a revision of a paper presented at the Fifth Annual Conference on Industrial and Control Applications of Microcomputers.[4]

Once the interface between the camera (and its associated manual controls) and the automation system was defined, development of the two parts proceeded independently. Furthermore, debugging was carried out independently for the two parts, using some hardware simulators and, for the software, a microprocessor simulator on the RCA CMS timesharing computer.[3] In addition to the obvious benefits of the top-down approach, this allowed the automation group to fix most of its problems with few demands on the camera group or its equipment, which was itself a center of great activity as deadlines approached. When problems were discovered by either group, modifications could be tried without interacting with other parts of the system.

The value of the berry picking approach was proven when the individual systems were connected. Few new problems were found. In spite of our efforts, most were related to the interface. These problems were not serious (once discovered), and were roughly equally divided between hardware and software. The automation was completed on schedule and demonstrated at the National Association of Broadcasters Show in April, 1978.

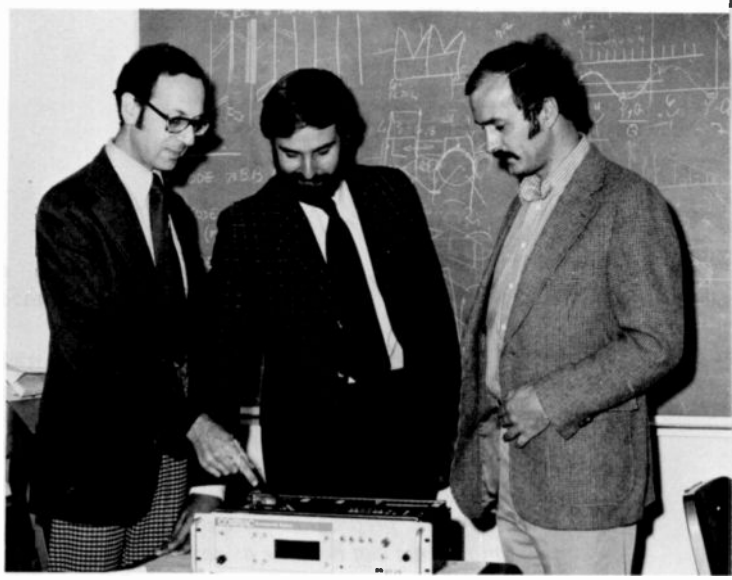
It is only fair to point out that the additional work did indeed require additional manpower and cost. But without the approach of partitioning the project to allow parallel development, the additional people and money would have been used less efficiently, and the time schedule would not have been met.

Acknowledgments

Although the emphasis of this article is on the separation of development tasks, cooperation — not always the same as communication — between the two groups was essential, at the planning stage, throughout the development, and particularly at the difficult moment of connection of the complete system. The authors acknowledge the help and cooperation of the RCA Broadcast Systems Division, who developed everything in the camera other than our small part, and have final responsibility for the success of the project. At our Laboratory, Robert Flory and Bernard Hurley made contributions as significant as the authors'. The Leader of the Laboratory group, Charles Oakley, first suggested the idea of using two microprocessors to divide the tasks.

References

1. Brooks, F.P., Jr., *The Mythical Man-Month*, pp. 13-26, Addison-Wesley (1975).
2. Yourdon, E., Constantine, E., *Structured Design*, pp. 95-126, Yourdon Press (1978).
3. *Timesharing Manual for the RCA CDP1802 COSMAC Microprocessor*, RCA Solid State Div., Somerville, N.J. (1978).
4. Fifth Annual Conference on Industrial and Control Applications of Microprocessors, *Conference Proceedings*, pp. 161-164 (March 1979).



Mike Lurie is shown on the left, Ken Schroeder, center, and Brian Astle on the right.

Brian Astle originally joined RCA in 1967 at the CE Division in Indianapolis from Mullard Research Laboratories in England where he had worked on applications of soft ferrites. While with the CE Division he worked on television-related products doing analog circuit and system design. After leaving RCA in 1970, he rejoined RCA, in 1974, at the DSRC to work on television devices and circuits. Recently, he has concentrated on software for new products.

Contact him at:
RCA Laboratories
Princeton, N.J.
Ext. 3230

Michael Lurie joined the RCA David Sarnoff Research Center in 1969 doing research on noise in holographic systems, and applications of holography to video recording. He was Team Leader for the development of optics for the Holo-Tape holographic video recording system. His present interests are electro-optic system, with emphasis on optical video recording, and microprocessor control for electro-optic systems. Dr. Lurie is the author of many technical papers, and several patents.

Contact him at:
RCA Laboratories
Princeton, N.J.
Ext. 3110

Ken Schroeder has ten years of software experience in both minicomputer and microcomputer systems. Now working on microcomputer-based consumer products, he has also worked with software for medical instrumentation, laboratory automation, and navigation satellites. His emphasis has been in the area of hardware/software systems tradeoff and software engineering.

Contact him at:
RCA Laboratories
Princeton, N.J.
Ext. 2890

K. Karstad

Intelligent controllers for color-television receivers

The microprocessor brings new and powerful features to TV-receiver control including a flexibility that previously was not practical.

Abstract: *Microprocessors now make possible programmable TV-receiver controllers with intelligence. Electronic tuning, implemented with phase-locked loop frequency synthesis, allows automatic compensation for unavoidable differences in tuner characteristics. With a phase-locked loop system and a microcomputer under user control, preprogramming of channels and events can be done at any time. A battery back—up system is implemented to avoid losing the prestored information. The software and hardware interaction that makes all this possible is described.*

Until recently, the number of user-controllable features on TV sets was quite limited, both for technical and economical reasons. Channel selection and indication were by mechanical means and the remaining features were restricted to sound and picture control. With the advent of the microprocessor, electronic tuning became common. Electronic control opened up the way for channel selection from a keyboard. However, there is no clear industry trend when it comes to applying electronics to control functions.

In most cases the control electronics have been standard MSI circuits or custom MSI/LSI tailored for specific functions. The availability of the microprocessor is about to change all this. A sprinkling of TV models with microprocessor control is already on the market,[1,2] and opens up promises of numerous useful and sophisticated features not previously possible.

This article describes a microprocessor-controlled, color-TV receiver with a large number of features and options. Not all the features will be commonly found in the average receiver for a long time to come. Nevertheless this design, based on the CDP1802 microprocessor, illustrates the possibilities of applying the stored program concept to implement a variety of ideas. Microprocessor control offers a modular design approach. Basic and essential functions such as tuning and station selection are implemented first and other features can be added or existing ones modified for various models or markets. Important new features can quickly be brought to the marketplace simply by program changes or additions without a long range custom LSI development phase.

The design also highlights the close coupling between software and hardware, and how several real-time events are processed by the CDP1802, orchestrating all three modes of I/O techniques: programmed, interrupt and DMA.[3]

Tuning control

An all-electronic tuning system is implemented with frequency synthesis. A phase-locked loop (PLL) selects the correct local oscillator (L.O.) frequency for a specific channel and keeps the receiver tuned regardless of changes in the tuner due to time and temperature (Fig. 1).

The PLL system differs fundamentally from the open loop analog system commonly employed. In the latter case a predetermined voltage is applied to the tuner with the expectation that the generated frequency will be correct. With no feedback, as in an open loop system, changes in components with time and temperature affect the frequency. Automatic compensation for unavoidable differences in tuner characteristics is not achieved in an open loop analog type circuit.

Any number on the keyboard, from 1 to 99, is translated into a bit code which sets the PLL to regulate on a constant L.O. frequency. The data format is a 16-bit word; the two most significant bits are decoded to select 1 of 4 bands (VHF1, VHFIII, VHF, Extra). The remaining 14 bits are the binary representation of the *N*-number for a specific channel. Fourteen bits give sufficient resolution to place the L.O. frequencies as little as 25 kHz apart for VHF. This makes it possible to select any channel in the world, as well as unassigned channels used in cable TV and master antenna system (MATV) installations. Additional fine tuning is implemented with the same resolution.

The tuner's L.O. frequency is measured and the control voltage necessary to force the frequency to be correct for a selected channel is generated by the loop. The divide-by-*N* counter, under software control by the user, therefore, selects the channel.

A ROM table translates the key depression into the correct bit code for the PLL according to international channel allocations. With a 14-bit code, the PLL design places the L.O. on the exact

Reprint RE-25-3-4
Final manuscript received Oct. 3, 1979.

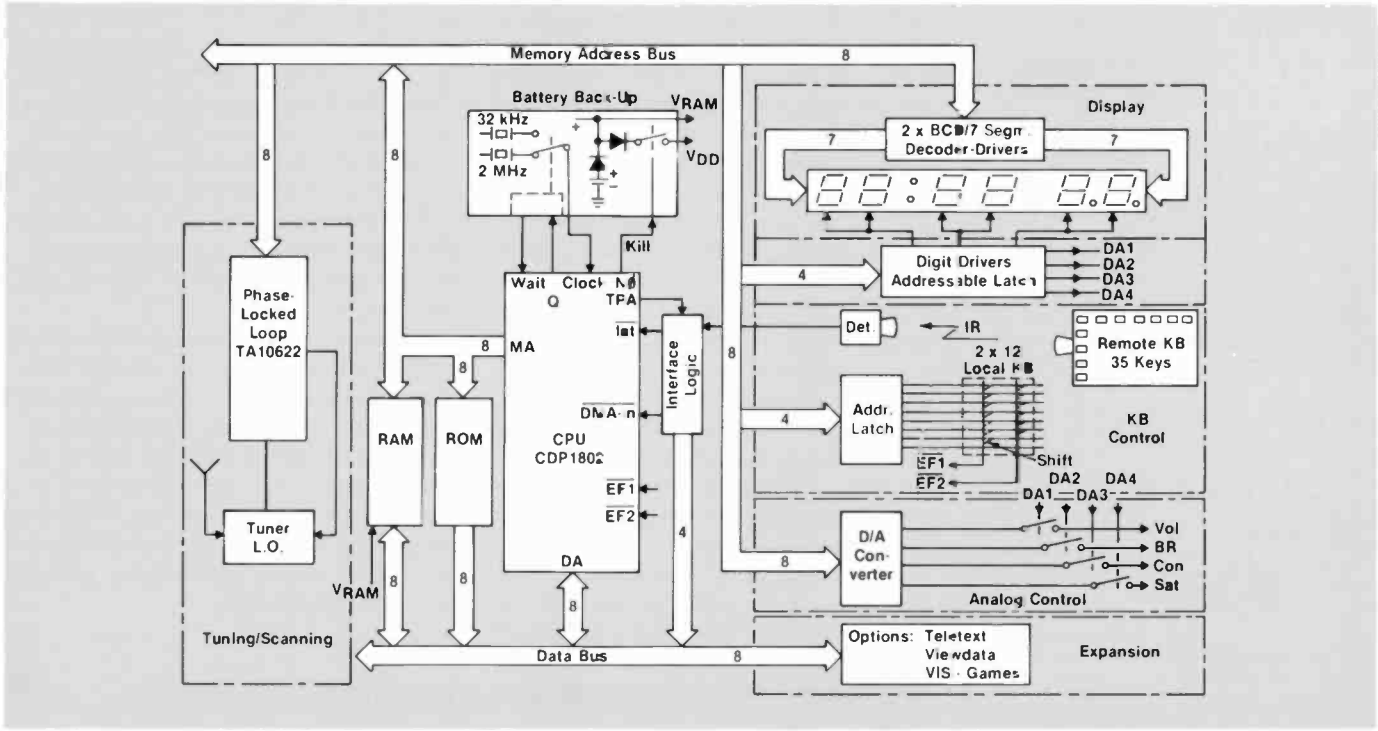


Fig. 1. The user interacts with this CTV controller via a local 2 x 12-keyset or a 35-key remote unit sending IR pulses. Display and feedback are on a 6-digit display showing clock and program.Exact tuning is through a PLL.

allocated frequency within any channel and additionally permits direct fine tuning, over the whole channel. While the traditional reason for fine tuning (AFT) is eliminated by the exact frequency synthesis approach, some users desire it in fringe areas for improved picture quality. There is also, in some areas, a need to tune in non-allocated channels for cable TV and master antenna systems (MATV). Fine tuning is implemented in 25-kHz steps for VHF and 100-kHz steps for UHF.

Preprogramming

An important feature of the controller design is preprogramming of channels. A program is the set of instructions that the viewer uses, by depressing the keyboard keys, to control tuning, channel changes, and viewing times. Each program may be assigned to any arbitrarily chosen channel. In this design, 16 storage locations are allocated for 16 different programs. Into any of these locations, any program can be stored and assigned to any channel. Tuning now amounts to dialing a number between 1 and 16.

This preprogramming technique requires the channels to be known. For the more general case where this is not true, a scanning mode is available. After pressing the scan key, the search, starting at the current channel, stops at the first higher channel available on the air. If the scan mode is terminated, the frequency channel is automatically assigned to the currently

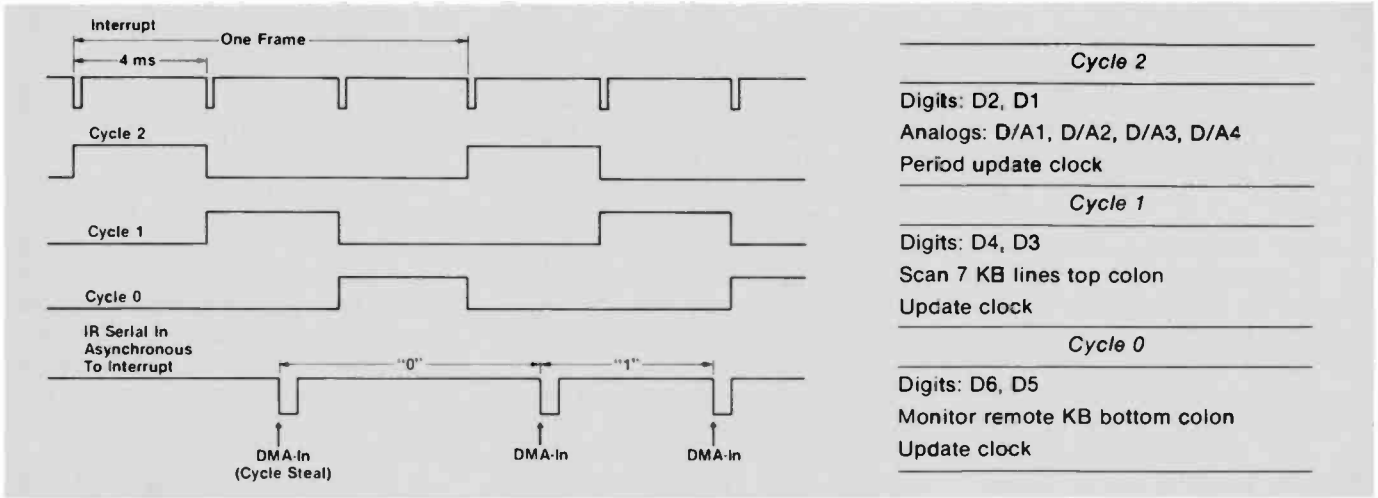


Fig. 2. The system is interrupt driven, with the interrupt service routine split and time multiplexed. Remote IR control pulses are received in real time via DMA without interference with ongoing processing.

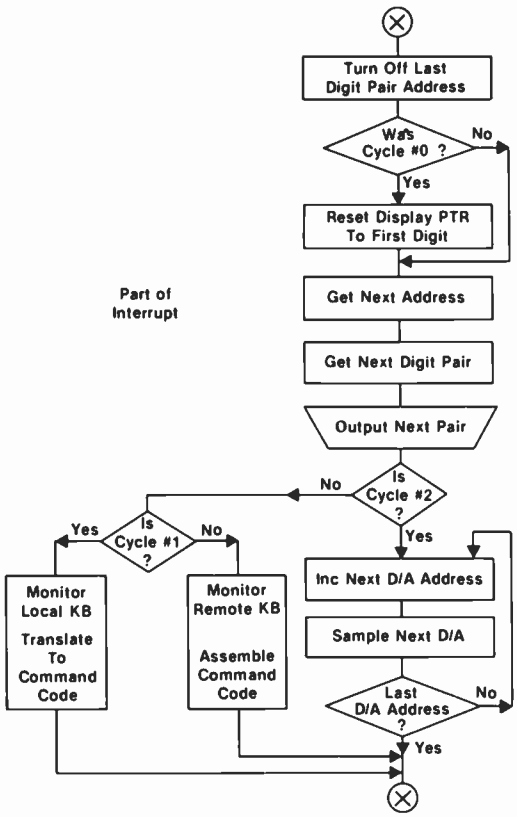


Fig. 3. This detail of the interrupt service routine shows time division multiplexing in three cycles in order to distribute the real-time load.

chosen program number. Each push of the scan button makes the tuning system search for the nearest higher available carrier. If no station is found, the scan wraps around (goes back to the current channel) and stops on the frequency from where the search started.

In addition to the channel scan discussed above, a program scan is also part of the tuning system. This scan searches the 16 prestored programs. The scan starts at the current program and stops at the nearest higher program number. Off-the-air stations are quickly bypassed while valid stations are given enough time to tune in. A wrap-around feature is part of this scan mode also.

A 6-digit display and a 24-hour clock make it possible to preprogram and monitor events. This applies for a period from the current time to 24 hours into the future. An arbitrary number of memory locations, eight in this design, are reserved for event programs. The user, while watching the display, simply keys in the time for the set to turn on (or switch if already on) and the program number to tune to. This can be repeated until the list is full, which is indicated by the display flashing 9s. There are no restrictions on the sequence of entering events. The software orders the list in time sequence so that thhe event nearest in time always is on the top of the list. The software checks every minute to see if an event is to be activated; if so, the executed event is erased from the list and the next in time is moved up. One of the digit's decimal points, if lit, indicates an event is pending. When it is executed, the period starts blinking until the receiver acknowledges by pushing any key. If no acknowledgment is received within 5 minutes, the set turns itself off.

What is described above may be termed a "once" event. It is just as easy to program an event which is automatically repeated every 24 hours or one that turns a program off at a specified time.

The programmed information (time, program number) can be listed on the display by a LIST button on the keyboard to indicate if it is a "once," "off" or "repeat" event. If the list is empty or exhausted, the display flashes 8s.

Notice that, with a PLL system, preprogramming of channels and events can be done at any time. No stations have to be on the air. The set can even be off. It is now possible for the dealer to preprogram sets before they are delivered to customers.

Display and keyboard control

Channel and program numbers are shown on a 6-digit LED off-screen display. For a period of time, on-screen display was popular. It seemed, however, that a dedicated off-screen display was the simplest solution. It can be on permanently, and does not distract the viewer's attention from the screen.

Six digits are the optimum design permitting continuous display of clock and program number. This is the normal or priority display mode. Upon request from the keyboard, both channel and program numbers are shown, the channel number displacing the clock display. Also, in the scanning mode, the channel frequencies assigned to program numbers are automatically displayed. A 6-digit display is sufficient for easy programming of events as well as listing what events already are programmed.

The local keyboard, with 12 keys and shift-key, can handle 24 tasks. The basic ones are program selection, scanning, analog controls (up/down), mute, on/off and setting the clock.

The master keyboard is remote with 35 keys and communicates with the receiver over an infra-red link. Each depressed key generates a 7-bit unique code as a pulse position modulated (PPM) pulse train.[4] This pulse train, an asynchronous event with respect to any other task, is processed by the DMA channel. Hence, the random real-time event is given required priority.

The presence of a real-time clock permits event programming, i.e., a certain program can be automatically turned on or off at a specific time.

The analog values, volume, brightness, contrast and saturation, are changed continuously up or down while holding the appropriate key. A 6-bit word in memory, providing 64-step resolution, is output at a predetermined rate and incremented or decremented by one each time. With the sample-and-hold multiplexing technique, one output port is sufficient for serving four different channels.

Expansion

A modular structure of the software makes it possible to add options by adding additional ports to the data bus and additional segments of code to the software. Two options of considerable future interest are Teletext[5] and Viewdata.[6] If, for instance, a key marked Teletext is depressed, the software checks for the existence of this option. If it is present, certain keys are, from this moment, redefined and execute different tasks from the normal TV mode. The latter mode is recalled by another key, NTV.

A Video Interface System (VIS), which also is Teletext compatible, is another option. This system, built around two LSI CMOS chips,[7] offers a variety of formats for displaying and modifying data under software control with either NTSC or PAL compatible output signals.

Battery back-up

A battery back-up system is implemented to avoid losing prestored channel programs and event programs during power failure. The outage may last from seconds to hours and be indefinite if the user simply pulls the ac cord, accidentally or planned. Due to an all CMOS system, continuous operation is economically possible with four small nickel-cadmium (NiCd) rechargeable cells.

A two-level battery mode is implemented. The central processing unit (CPU) senses if the power fails; the NiCd cells automatically take over and run the

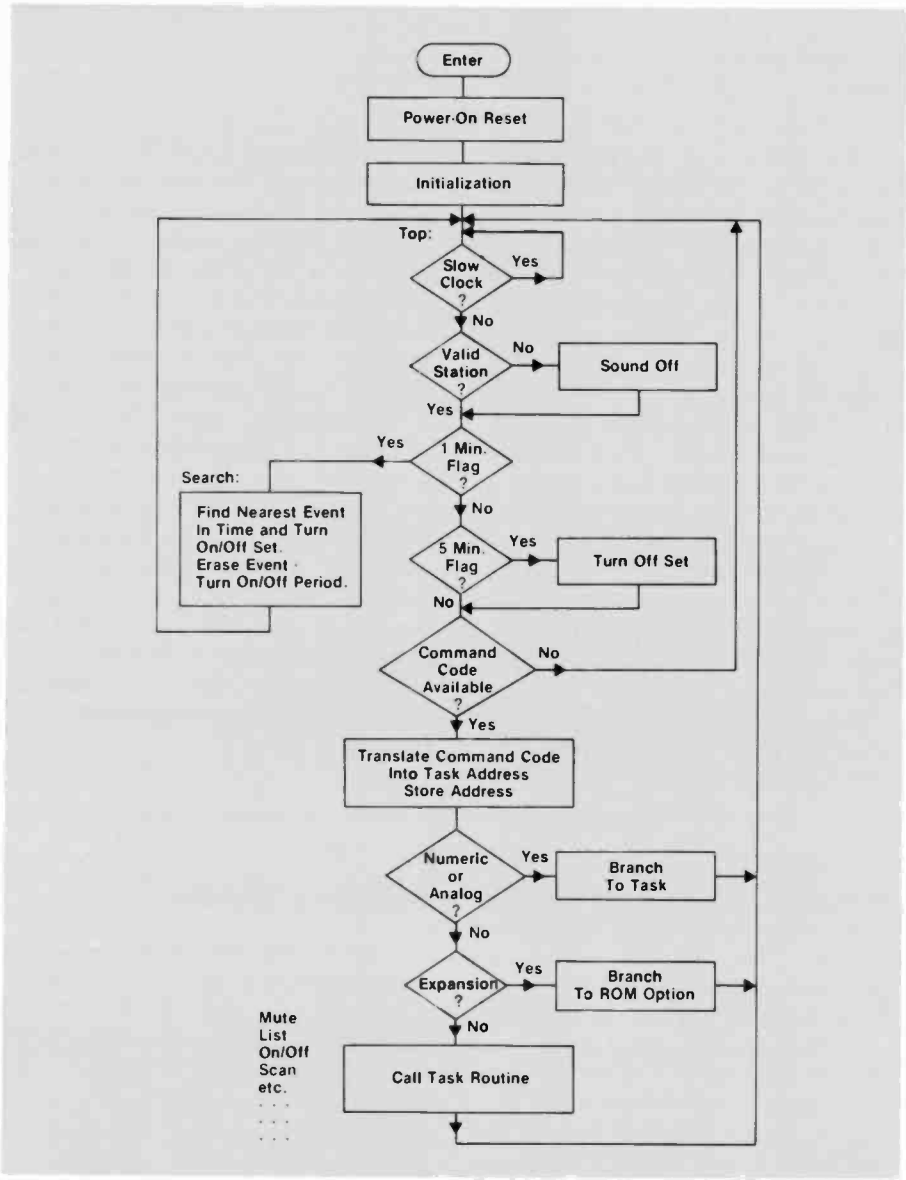


Fig. 4. The main program loops in a short section, testing to see if a keyboard command was received. If it is, the appropriate task is called and executed.

whole system for a predetermined period: one week, refer to the flowchart (Fig. 4).

In order to conserve power, the CPU, upon detecting power failure, automatically switches from the 2-MHz crystal to a low frequency crystal at 32 kHz. The switching takes place during a few milliseconds WAIT state, which the CPU also enters upon detecting power failure. If power is not restored within the predetermined number of seven days, the CPU turns off after shutting down the whole system, with the exception of the RAM, which stores user-programmed information. The remaining battery power is sufficient for approximately three months of storage. If power returns before three months, operation resumes as normal with stored information intact.

Software and hardware interaction

There are many tasks to be performed in this real-time controller application, with strict requirements. The software requirements are: ability to multiplex the 6-digit display, to multiplex four analog channels, to update the clock, to monitor the remote and local keyboards for action, and to execute the keyboard-defined tasks.

The clock must be updated regularly and frequently enough so as not to lose time. The display must be refreshed and at a rate high enough to avoid flicker, i.e., at least 50-60 Hz. The analog values must also be refreshed steadily at a rate high enough to maintain a dc signal and without excessive large filter constants.

The keyboards must be checked frequently enough to catch any random key depressions. Finally, any command received (fine tuning, scan, mute, etc.), must be processed and executed.

These overlapping and partially conflicting specifications are resolved by an interrupt driven system together with a DMA-channel for the remote keyboard. A pulse train, derived from the CPU clock, interrupts the software program every 4 ms. (Fig. 2). A one-second counter in the interrupt routine is incremented, and later in the same cycle checked, to determine whether one second has elapsed. If it has, an update routine (still part of the interrupt service routine) updates 4 RAM locations of the clock display. The load is distributed during interrupt by dividing it into 3 cycles. The processing in these cycles is done according to the flowchart of Fig. 3, which shows part of the interrupt service routine between the incrementing of the one-second counter and the update routine. The sequence of events is as follows: an interrupt occurs. The last digit pair is turned off and the next pair is turned on. Then, the four analog channels are multiplexed and refreshed and the clock is updated. Control now returns to the main program. At the next interrupt, the last digit pair is turned off and the next one on. The clock is updated again, but in this cycle, the local keyboard is also scanned for key closure. At the third interrupt, the last digit pair is handled and the service routine monitors whether or not a character is being received from the remote keyboard and processes the command. Once more the clock is updated, and the main program resumes execution. The clock is updated every 4 ms, and every 12 ms the display and the D/A channels are refreshed. Both keyboards are monitored, and, if a key closure took place, the required processing is done for the main program to act upon later.

A key depression on the local keyboard is translated into a command code and stored in memory for the main program to process. The software monitors the presence of a bit stream arriving from the remote keyboard over the infrared link, measures the time between pulses, and assembles the information into a character which represents the actual command code. Finally the program tags the received command code as to what type it is; a single command (ON/OFF) or a repeat command (VOLUME UP) for the duration of the key depression.

When control returns to the main program in each interrupt time slot, the

A unique hardware and software interplay for the clock, display and keyboard

The local keyboard with 12 keys and shift key can handle 24 tasks. These are the basic tasks such as program selection, scanning, analog controls (up/down), mute, on/off and clock setting. The master keyboard is remote with 35 keys and communicates with the receiver over an infrared link. Each depressed key generates a 7-bit unique code as a pulse position modulated (PPM) pulse train. This pulse train, an asynchronous event with respect to any other task, is processed by the direct memory access (DMA) channel. Hence, the random real-time event is given required priority.

Four locations in random access memory (RAM) are assigned to hold the four clock digits. Another buffer in RAM, the display buffer, contains binary coded data (BCD) for the six display digits. The update routine transfers data for the four clock digits from the update buffer.

The configuration of the data in the display buffer is closely related to the hardware scheme. Segment data for a digit pair are fed from two separate latch/decoder circuits. An 8-bit addressable latch selects the digit pair or the D/A channel to be sampled. Another addressable latch provides scan address for testing key closures on the local keyboard. The required 16 bits of data are assembled in one 16-bit central processing unit (CPU) register and output over the memory address bus with just one instruction. Four RAM locations store the 6-bit D/A data.

The software section of the interrupt routine interacts with this hardware. The program turns off the last select line for a digit pair and tests if it was cycle #0. If the answer is yes, a new frame of events is about to be repeated and the display pointer is reset to the top of the display buffer. Segment data for the next digit pair is output and the pair is selected. In this example, the pair is output and selected. In this example, the answer to testing cycle #2 is yes. The four D/A channels are now sampled in a burst mode after which the routine exits to the update part. At next interrupt, since it was not cycle #0, the program turns on the next digit pair. The program finds itself in cycle #1 and the local keyboard routine is activated. Finally, at the third interrupt (cycle #0), the remote keyboard routine is sequenced.

program simply checks to determine if a command was received from one of the keyboards. If so, that task, i.e., change brightness, is executed. If no command is received, the CPU is essentially idle until the next interrupt occurs. In most cases, the free time in one of the three cycles is ample time for processing any task in the main program; if it is not, the next interrupt simply postpones the background processing into one or a few more time slots. This is not apparent to the user.

The flowchart for the main program (Fig. 4) gives additional information. An interrupt can occur any time in the main program. Most of the time, it loops at the top (goes back to the beginning of the program), when no keyboard command is received.

Summary

The microprocessor brings new and powerful features to TV-receiver control including a flexibility that previously was not practical to implement. Incorporating a real-time clock allows event-programming; the TV set turns itself on or off or switches stations in a preplanned sequence. User interaction is via a local keyboard on the set or a remote 35-key unit. A PLL loop with its technical advantages forms a major system block and has its own LSI controller. With the exception of the ECL prescaler, the low-pass filter and operational amplifier, all circuitry for the PLL loop is integrated on one CMOS chip.

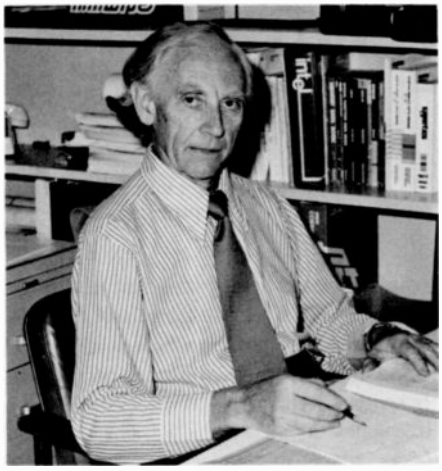
A more compact system is possible, in a future design, with more of the MIS logic integrated on larger but fewer chips. Also, if the CDP1804 microprocessor is substituted, 2-K bytes of ROM and 64-bytes of RAM would be absorbed inside the CPU chip. The modular design approach will permit other features to be added at later dates. Once intelligent control is implemented, the features are only limited by the designer's imagination and the economical facts of the marketplace.

References

1. Kleiman, Art, "Programmable Color TV," *Radio-Electronics* (May, 1977).
2. Baum, Wolfgang, "Farbfernsehgerat mit Microprozessor — Steuerung," *Funkschau*, Heft 17 (1977).
3. *User Manual for the CDP1802 Microprocessor, MPM-201B* and *COSMAC Microprocessor Product Guide, MPM-180B*, RCA Corp., Solid State Div., Somerville, N.J.
4. Valvo: Data Sheet SAB 3011.
5. "Teletext — The LSI Solution," Mullard Technical Information, TP1606.
6. "Viewdata," *Wireless World* (Feb., 1977).
7. Data Sheets CDP1869 and CDP1870, RCA Corp., Solid State Div., Somerville, N.J.

Acknowledgment

The author gratefully acknowledges the help and contributions of Jerry Fogarty.



Kaare Karstad has worked in the microprocessor field since the product group was formed in the Solid State Division. His work has covered applications, engineering, and marketing support. He is presently doing microcomputer system designs in the areas of consumer applications and support systems. His earlier background includes CMOS applications and computer-oriented research at the RCA Laboratories.

Contact him at:
MOS Microcomputer Systems Engineering
Solid State Division
Somerville, N.J.
Ext. 6041

B.W. Beyers|A.J. Suchko

The AutoProgrammer

The AutoProgrammer, which gives the viewer program sequence control, is a new function spin-off of the 8-bit microcomputer.

Abstract: *The 8-bit microcomputer provided the option of designing a programmer for TV-viewing control. How this new product, the AutoProgrammer, was defined during the design phase, and how optimum performance vs. low cost influenced microcomputer selection is discussed. The software and hardware that were used in the AutoProgrammer are described. The future promises a continuing evolution of microcomputer features.*

The evolution of a new component provides the product designer with two choices: either to improve the cost performance ratio of an existing function or to provide a completely new function. Primary emphasis in the design of mature products is put on the first option; however, occasionally a technology evolves which generates a component whose attributes encourage the second option. The origin of the 8-bit microcomputer provided the option of designing a programmer for TV viewing. Of course, such a product could have been built before the advent of the microcomputer, but the cost/performance ratio did not appear satisfactory until the 8-bit microcomputer was available at a consumer product price.

Product definition

It was desirable to minimize the development expenditure of the programmer, especially tooling costs, and to have as little impact on the rest of the TV receiver as possible. It was also felt that a digital clock built into the receiver was marketable. So, our approach was to use essentially the same front-panel space as a normal keyboard-tuned receiver. This was accomplished by using a 12-key keyboard (3x4 keys) instead of the present 10-key (3x3 + 1 key) keyboard, by controlling receiver power and volume from the keyboard, and by replacing the 2-digit channel display by a 4-digit display which displays both time and channel.

The next major step in the product definition was to define algorithms that allowed the user to enter and review viewing schedules using only the 12 keys and the 4-digit display. (Use of the CRT for display was considered, but because of the desirability of a continuous clock display and higher cost of a CRT display, the product was restricted to a 4-digit, 7-segment display). In order to interact with the user, the keyboard was double labelled and the 7-segment display was used to create various alphanumeric abbreviations to prompt the user (Ed — for edit, etc.) and to display time and channel information. To simplify operation of the receiver in the "normal" viewing mode, all programming input functions are active only when the TV set is off; thus, in the normal viewing mode the receiver function is similar to an earlier model (Direct-Address model). With the set off, the programming input sequence is: Start, Day, Time of Day, Channel, End. Using the Monday through Friday button for the day selection allows a single input command to select a channel each weekday. Other functions such as Clock Set and Edit rely on similar input sequences.

Microcomputer selection

Four factors were of major importance in selecting the microcomputer for this task, namely: price, amount of Read/Write, random-access memory (RAM), number of I/O ports, and amount of read-only memory (ROM). At the time our decision was made, the Mostek 3870 microcomputer was easily the best choice. The nearest competition did not have sufficient I/O capability, had only one-half the ROM of the 3870, and cost more. We were not overly concerned about the machine organization or instruction set because we felt the auto-programmer software would not involve extended calculations or impose serious time constraints. The RAM size, of course, limits the number of user commands that can be stored for channel selection. The emphasis on latched I/O results from our desire to drive the display without multiplexing the digits. In TV receivers, the multiplexing currents frequently radiate into the "rabbit ear" antenna and interfere with the TV picture. It was uncertain how much ROM would be needed to realize the desired algorithm, but we felt the 2048 (two kilobytes), 8-bit words in the 3870 would be sufficient, whereas 1024, 8-bit words (1 kilobyte) might not be enough. (Of course we needed all of the 2 kilobytes.) The price of the 3870 has, unfortunately, increased over the initial quote. Apparently, manufacturers overestimated the yield and underestimated the market.

Software

The time and channel change information is stored in two, 8-bit words: three bits for day of the week and five bits for hour in the first word, one bit for half hour and seven bits for channel in the second word (Fig. 1). The system is restricted to operating on the half hour. The seven bits for channel

Reprint RE-25-3-5
Final manuscript received Sept. 19, 1979.

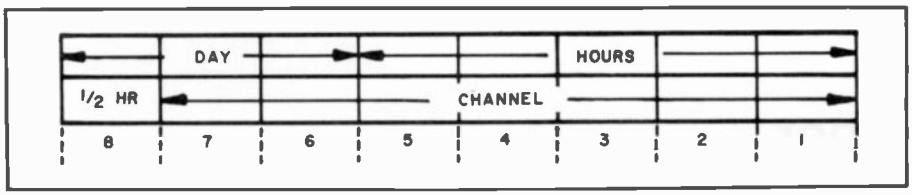


Fig. 1. The time and channel change information is stored in two 8-bit words.

Fig. 2. Each programming command begins by pushing the start button and ends by pushing the end button.

information are kept in a binary-coded decimal (BCD)-like format with channels above 79 folded into unused codes (i.e., channel 80_{10} can be coded as $7A_{16}$, 81 as $7B_{16}$, 82 as $7C_{16}$) since hex codes above 9 are unused in BCD representation. The eighth day code is used for the Monday-Friday option. The routine that interprets channel change commands searches the command list in such a manner that any individual day command programmed for the same time as a Monday-Friday command takes precedence. This allows an "except" feature for easy implementation of a Tuesday, Thursday, and Monday, Wednesday, Friday schedule, if desired. Channel 00 is interpreted as an off command.

One of the problems with the 3870 is that nesting subroutines is difficult. As a result, we restricted our software to one level of subroutines.

Interrupts

The interrupt facility of the microcomputer is used to drive the time-of-day clock from the 60-Hz power line. In order to save RAM, no machine states are saved during an interrupt. (That is, when an interrupt occurs, the program counter, accumulator and other information normally used to reenter a program are not saved). This is possible because the tasks to be performed can be done between interrupts (with but one exception) and the programs do not need to be reentered. The idle loop, which scans the keyboard, also manipulates the interrupt enable and allows the 60-Hz clock to interrupt at a time when the machine state is known. Another advantage of this scheme is that subroutines associated with the 60-Hz clock can also be used by the main program without writing reentrant code. The one task that exceeds 16 milliseconds (the period of the 60-Hz interrupt) is the channel change command which requires a long delay (~ 500 ms) when the TV set power is turned on, and 33 milliseconds between digit outputs to the synthesizer. This is handled by setting a flag which directs the program flow to the channel-change subroutine after a 60-Hertz interrupt instead of accepting keyboard inputs.

Fortunately, it is only necessary to maintain the volume level (via a dc voltage) while the receiver is on and tuned to a station. In this state, most of the processor time is spent in the keyboard scan loop with short interruptions to serve the 60-Hz clock and occasional loops through the "search" routine to see if a channel-change command should be executed. Since none of these functions is very time demanding, it was possible to add the 40-Hz variable-duty cycle rectangular-waveform generator in the keyboard scan loop. The processor clock (4 MHz) is divided by an internal prescaler and counter by a factor of 10^5. The content of the volume register is compared to the continuously counting modulo 256 counter to determine the duty cycle. A "1" is outputted as long as the content of the volume register is greater than the content of the counter.

Internal delays are derived from the counter associated with the volume (100-μs increments), or from a counter associated with the 60-Hz clock (16 millisecond increments).

Controls for the human factor

"People Proofing" is a major problem in programming a computer for a consumer product. The algorithms for the AutoProgrammer were constrained to satisfy three major human factor criteria: (1) symmetrical data entry, (2) no illegal entries, (3) no penalty for the experienced user. By "symmetrical data entry", we mean that all commands have the same format. Each programming command begins by pushing the start button and ends by pushing the end button (Fig. 2). The

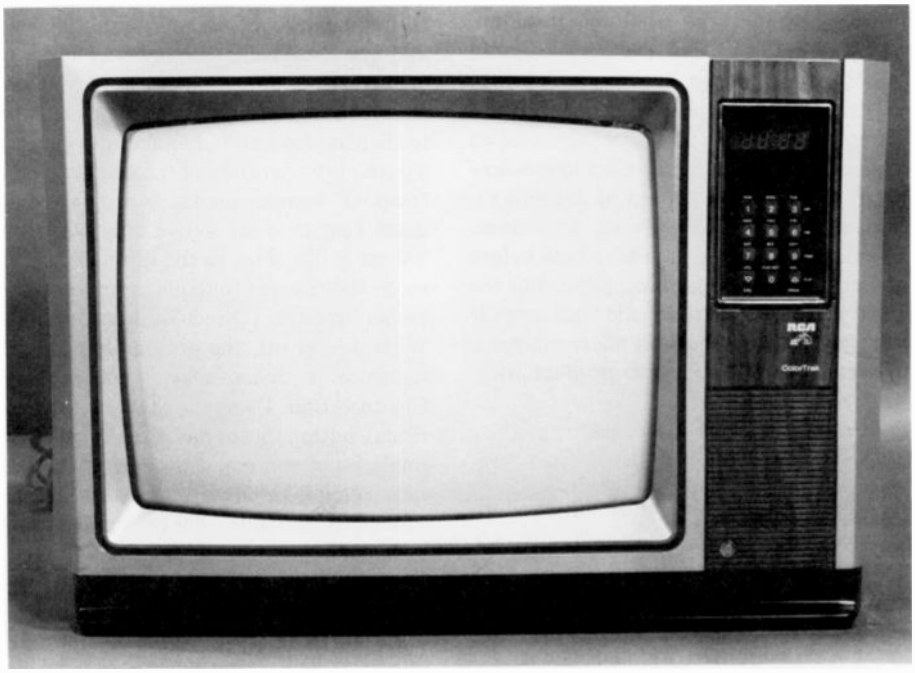


Fig. 3. The 12-key encoded keyboard and a 4-digit phospherescent display are located adjacent to the tube on the receiver.

start button is followed by a mode entry (Program, Edit or Clockset) then a day entry (e.g., Monday). In the Program- and Clockset-modes, time is entered next and followed by channel number, in the program mode. In the Edit mode, a programmed time is displayed first, followed by the channel number to maintain symmetry with the programming sequence. The display provides visual response to keyboard inputs.

Illegal entries, that is, entries outside the meaningful range, e.g. 13 o'clock, are not accepted. Since there is no response to keyboard entries outside the desired range, a partial entry is not affected by an obviously incorrect key closure. Programming is restricted to half-hour increments and entries are rounded to the half hour and displayed as such as soon as the last digit is entered. To avoid penalizing the user who does not need the visual prompts, the software allows the user to override the prompt timing by simply inputting the correct sequence at the user's preferred speed. However, the beginning user can retain the visual prompts as long as desired by holding the key down. When the key is released, the display is maintained for 2 seconds and then changes to prompt the next entry.

Software testing

Testing the software is another major problem for a consumer product. The problems one can think of are usually not *the* problem. To test for this other set of problems, a varied set of commands was programmed by a number of persons with different educational backgrounds. Further test procedures were written for testing the Auto-Programmer instruments in our life-test facilities. In this case, careful attention was paid to details such as correct timing, execution of commands at correct time and retention of programmed information. The 700-hour test should have revealed any small cumulative errors.

Hardware

Besides the 3870, the other major components in the system are a 12-key encoded keyboard and a 4-digit phosphorescent display (Fig. 3). Interfacing the keyboard is straightforward, but the display is more difficult because of the large number (33) of leads involved. In order to reduce the sensitivity to kine-arc induced voltages (discharges of its 30-kV acceleration voltage in the CRT) and to eliminate additional cable and mounting costs, it was decided to mount the display directly on the module printed circuit board. Because of this, the location and size of the module were constrained. These constraints made layout difficult and induced some concern about maximum temperature. Open-collector hex inverters (7406s) were used to provide the voltage gain necessary to drive the display and, because of the size constraints mentioned above, ceramic-resistor networks were used to provide the pull-up resistors for the inverters.

The power supply was designed to provide 85 mA to the 3870 at rated voltage for at least 400 milliseconds after power line dropout. This was accomplished by using a half-wave rectifier which charged a 6600-μf capacitor to 15 volts from the basic-power source and by cutting off the power to the 7406s during a power dropout.

With 32 input/output lines, it was still necessary to share five lines. The ON/OFF-control line is shared with the strobe line to the synthesizer tuner and four display lines are shared with the four data lines to the synthesizer. In order to do this a display digit is blanked during the strobe output and a delay is added to the On/Off circuit to keep the receiver on during the 33-millisecond strobe pulse. The delay was realized as a one shot with discrete components.

Hardware testing

The 3870 has special provisions for testing. Basically, these allow the internal bus of the computer to be accessed from two 8-bit ports. A commercial tester is available which tests the machine by reading the ROM and comparing the contents to a reference ROM. The instruction set is tested by comparison with another machine, and various parametric tests are also made.

The testing of the AutoProgrammer module, however, required a custom tester. To test the completed module, the microcomputer is forced to an idle state with all of its outputs high. The microcomputer then pulls each output low and checks for the appropriate level at the module extremity. For instance, the display outputs are checked by optical sensors that read the display. Power-supply voltages and other module parameters are also checked. This test has proven satisfactory. We designed and built a tester using a minicomputer. The basic philosophy of the tester is to test the module manufactured, assuming the microcomputer works correctly. To accomplish this test, the microcomputer is forced to an idle state with all of its output ports high. The microcomputer than pulls each port low and checks for the appropriate level at the module extremity. Display action is checked by optical sensors. Various power supply voltages are also tested and the microcomputer is exercised briefly. This test has proven satisfactory.

Conclusion

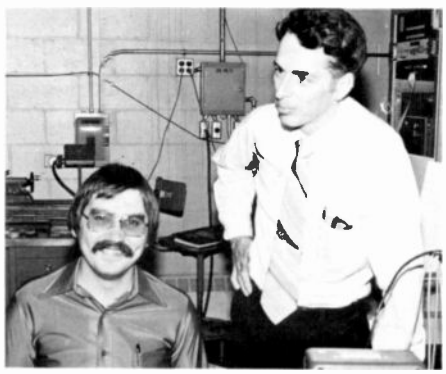
In the future, we expect to see a reduction in the cost per function of semiconductors that will allow scheduling features to appear in a larger part of the product line. We also anticipate a much broader use of microcomputers for television controls and a continued evolution of new features.

Adam Suchko has worked in all sections of television design and microprocessor development. He came to RCA in 1974 and worked on deflection systems and video test systems. He has worked on microprocessor-based systems since 1976, including the Autoprogrammer.

Contact him at:
RCA Consumer Electronics
Indianapolis, Ind.
Ext. 5908

B.W. Beyers joined RCA Consumer Electronics in 1965 where he worked as an advanced development engineer until 1974. In 1971 he was awarded the RCA Laboratories Outstanding Achievement Award for digital design related to TV. In 1974 he joined the TV tuning systems group.

Contact him at:
RCA Consumer Electronics
Indianapolis, Ind.
Ext. 5906



Suchko is shown seated beside Beyers in the laboratory.

D. Sprague

82-channel, single-touch TV tuning system

Intelligent tuning and volume-control interfaces for television viewers are now possible with analog touchplates and 8-bit microcomputers.

Abstract: *The invention of 8-bit microcomputers supplied the potential for an intelligent interface between the user and the TV: the single-touch TV tuning system. The analog-touchplate design is discussed. How microcomputers are used to give intelligence to the interface and provide ideal performance is described. The problems encountered in finding the best algorithm are demonstrated. A description of the advantages of a touchplate control system provides the conclusion.*

The interface between user and device is a very important part of any consumer product. A good interface allows easy and efficient communication while being safe, reliable, and rugged. There are definite advantages to be gained from giving the interface some intelligence through the use of a microprocessor or microcomputer. An interface that is intelligent is even more of an asset to the user and enhances the product.

The difference between microprocessor and microcomputer

Before continuing, let's discuss the distinction between a microprocessor and a microcomputer. A microprocessor is a single-chip processor which can execute instructions and manipulate data according to those instructions. In order to be considered a computer, a system must contain not only a microprocessor but also memory and input/output channels. A microcomputer is a single-chip computer which means that it consists of a microprocessor, some read-only memory (ROM), some Read/Write random-access memory (referred to as RAM) and input/output channels all on a single integrated circuit.

The microcomputer is generally more limited in computing power and expandability than a microprocessor, however, its conservative size makes it ideal for small-scale control applications such as an intelligent interface.

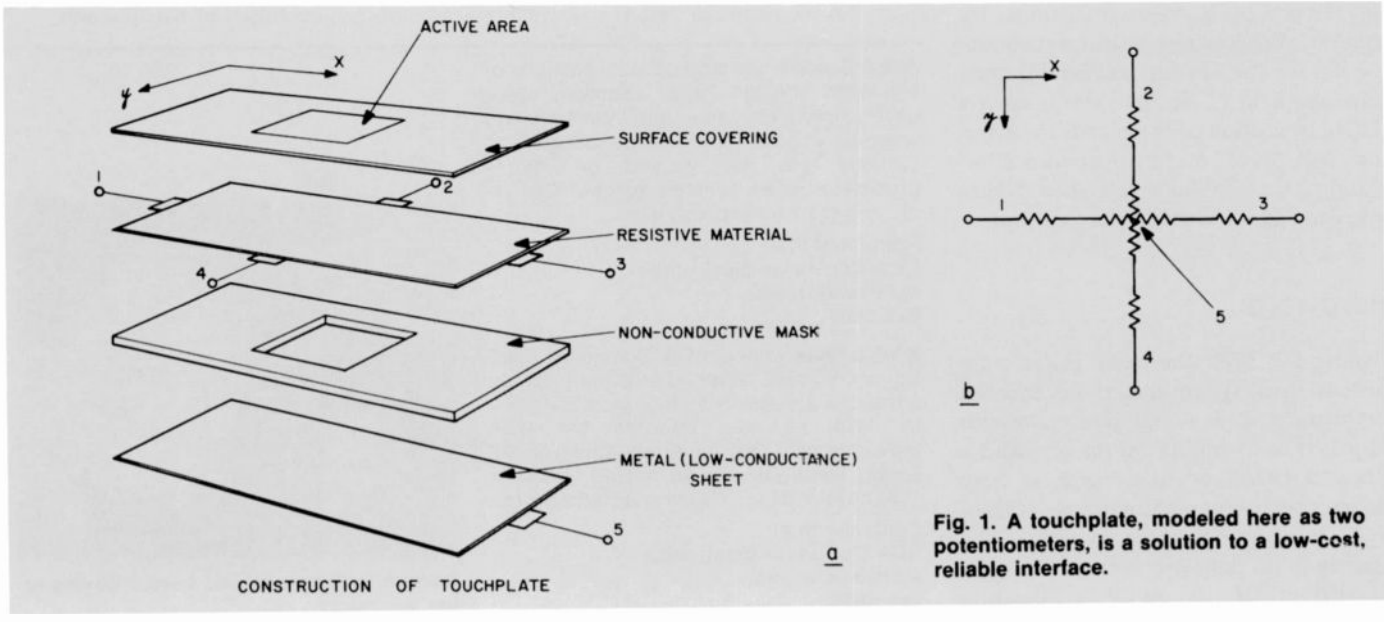


Fig. 1. A touchplate, modeled here as two potentiometers, is a solution to a low-cost, reliable interface.

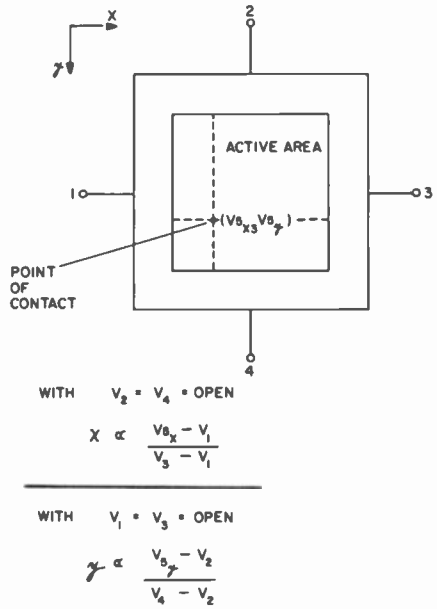


Fig. 2. The touchplate is a square-shaped surface from which the *x* and *y* coordinates of the point-of-touch may be determined.

Touchplate tuning and volume control

A project was established at the labs that would provide an evaluation of an idea for a TV input device. The device would give the user single-button access to any one of the 82 TV channels. The interface device should also feature volume control anticipating use in remote control systems. The final criteria for such a product, which is present in all consumer products, were low-cost and acceptable reliability.

One solution to the questions of cost and reliability was found in the use of an analog touchplate. This device is a square-shaped surface from which the *x* and *y* coordinates of the point at which it is touched may be determined. Since the device is analog, i.e., continuous over the surface, we can divide it, through software, into an array of cells of any order we choose (4x4, 10x10, or 5x6, for example).

There are several advantages to using this touchplate rather than a matrix of 82 calculator-type keys. The touchplate is both less expensive and more reliable. Surface coverings such as teflon or mylar may be specified to the manufacturer of the touchplate (I Incorporated of San Francisco, California) depending on the ruggedness desired.

A construction diagram and a simplified circuit model are shown in Fig. 1a and 1b.

The *x* and *y* coordinates of the point of contact are measured as follows. When the active area of the touchplate is pressed, the resistive material contacts the conducting plate at the point of contact. By applying a voltage difference between pins 1 and 3, while leaving pins 2 and 4 open, a voltage may be measured from pin 5, which is proportional to the *x* coordinate of the point of contact. Likewise, to measure the *y* coordinate, pins 1 and 3 are open circuited while a voltage is applied between pins 2 and 4. Now the voltage on pin 5 is proportional to the *y* coordinate of the point of contact (Fig. 2).

Microcomputer control

A microcomputer could be used to switch the voltages to pins 1 and 3 or pins 2 and 4, measure the voltage at pin 5 using an analog-to-digital converter (ADC), and scale these measured voltages (one for *x*, one for *y*) as necessary. Since there must be at least 82 possible selections from the touchplate, software will be used to scale the *x* and *y* values to range between 0 and 9. This gives us the equivalent of a 10x10 array of keys. Some of the extra keys will be used to control the volume.

A layout of the touchplate surface showing the channel numbers and volume control area is illustrated in Fig. 3.

In addition to decoding the touchplate, the microcomputer must also send the appropriate signals to the tuner to change channels and output a voltage level for volume control. The Intel 8746 microcomputer was chosen as the controller. Some of the features which influenced this decision were the 27 input/output ports, 1-k bytes of erasable programmable read-only memory (EPROM) (for program memory), and 64 bytes of RAM (for the storage of variables) all on the microcomputer chip. This proved to be more than enough memory and I/O, allowing us to get away with using only one chip (the 8748) for the brains of the interface device.

The connection to the tuner requires five output lines. Four of the lines are used to send binary-coded-decimal digits (4-bit binary words of values 0-9) and the fifth is a strobe. The volume control requires a voltage level output ranging from 0 volts (minimum volume) to 6 volts (maximum volume).

Analog-to-digital conversion

If the microcomputer is to control the touchplate and communicate with the TV, both an analog-to-digital converter (to decode the touchplate) and a digital-to-analog converter (to control the volume) will be required. However, since volume control isn't needed while decoding the

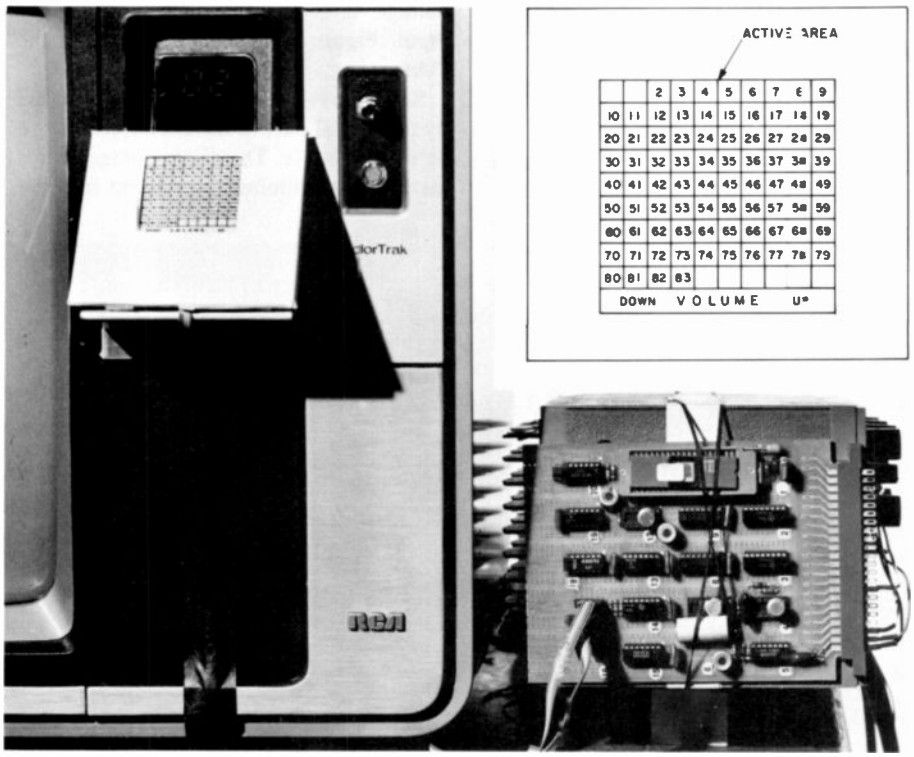


Fig. 3. There are 82 possible selections from the touchplate. Some of the extra keys are used to control volume.

Reprint RE-25-3-6
Final manuscript received Oct. 3, 1979.

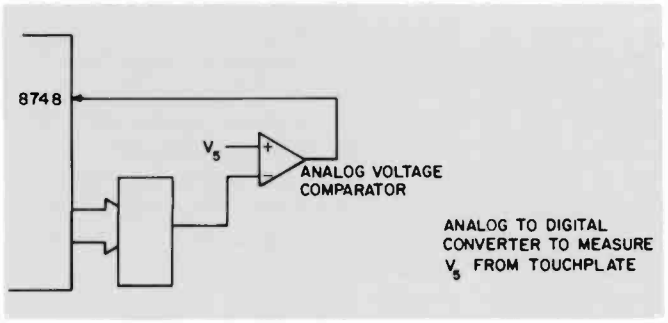


Fig. 4. The analog-to-digital conversion can be accomplished using the ADC, a voltage comparator, and a routine implemented in software.

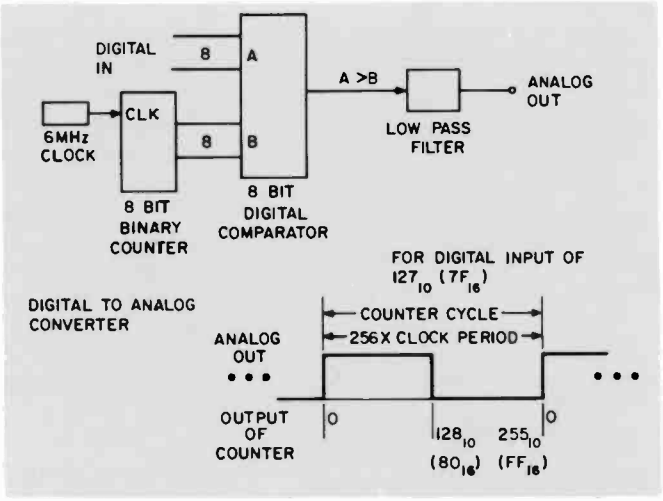


Fig. 5. Digital-to-analog conversion is implemented by a variable pulse-width method.

touchplate, the analog-to-digital conversion could be done using the digital-to-analog converter (DAC), a voltage comparator, and a successive-approximations routine implemented in software. This is illustrated in Fig. 4.

The successive-approximations algorithm works by putting each bit high, starting with the most significant one. A change in the output of the comparator means that the output of the DAC is greater than the voltage we are trying to measure. In this case, the bit is reset to the low state. If, when the bit is placed high, the output of the comparator doesn't change, then the bit is kept high. This procedure is repeated for each bit, in order of decreasing numerical significance and the output of the DAC zeros in on the unknown voltage. Once all eight bits have been tested, the resulting binary word being output to the DAC is proportional to this unknown voltage.

Digital-to-analog conversion

Now all that's left to be specified is the digital-to-analog converter. Because this DAC will be used in decoding position on the touchplate, accuracy will be an important consideration while the speed will be of lesser importance. For this reason, the DAC was implemented by a variable pulse-width method.

As the name implies, the mechanism of this type of conversion is to vary the pulse width (or duty cycle) of a periodic square wave in proportion to a binary input. Then a low-pass filter is used to remove the fundamental frequency and all harmonics. This leaves only the dc content or the average of the waveform. Thus, we obtain a voltage level proportional to a binary input. Figure 5 is a block diagram of the DAC.

Since the relative accuracy is determined by the digital part of the circuit, it tends to be very accurate. The disadvantage to this method is the limited conversion rate due to the time constant of the low-pass filter. As stated before, however, the accuracy, not the speed, is the important factor in this application.

Creating linearity with a microcomputer

After prototyping the circuit, which included the microcomputer, a DAC, a voltage comparator, and the necessary circuitry for switching voltages to the touchplate, software experimentation was started. At this point a major problem was encountered: the touchplate was much more nonlinear than expected. To illustrate the problem, equipotential lines for the touchplate are shown in Fig. 6. The effect is much greater near the edge than near the center, but, due to the small size of

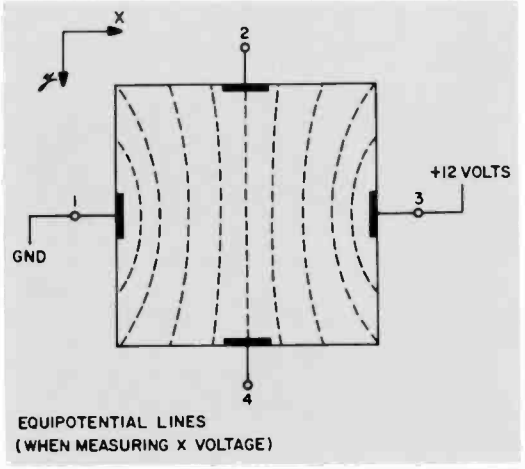


Fig. 6. Expotential lines show the nonlinearity of the touchplate.

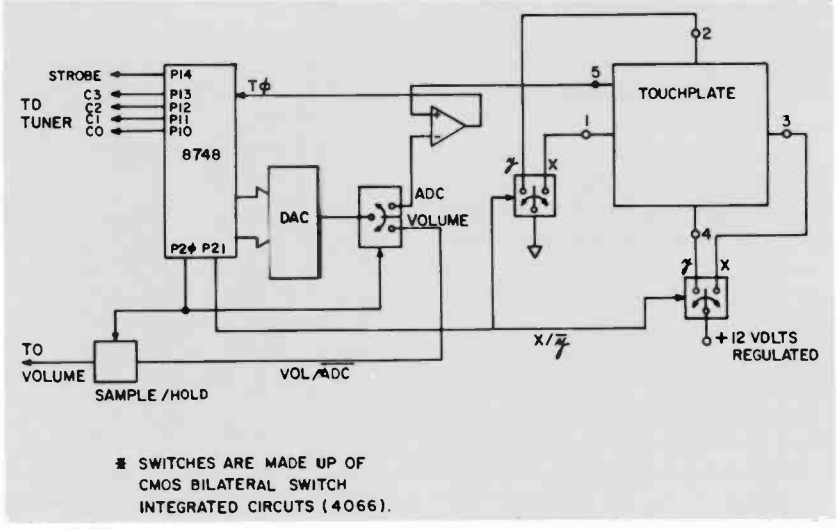


Fig. 7. The final design incorporates CMOS bilateral switches for voltage and volume-level output control.

the touchplate, the whole area had to be used. This nonlinearity greatly increased the complexity of the routine which uses the measured values of x and y to determine which cell (or key) is being pressed because the x and y values are no longer independent.

The problem could have been solved by changing the physical design of the touchplate, but for this project we chose to use the available computing power of the 8748, to make this nonlinear device appear to the user as essentially linear, though discrete. The essence of the interface design is the use of a microcomputer's computational power to alter the apparent characteristics of a device, to make the device appear more ideal.

Finding the best algorithm

Several search algorithms were considered in trying to find the best method to obtain the cell number (00-99), given the binary values of the x and y voltages. Most of the algorithms considered were unreliable because the column being selected depended on both x and y as did the row being selected.

The final solution, a rather brute-force one, was determined to be the best choice in terms of repeatability and computing time. This algorithm consists of checking the x and y voltages at the point of contact, in their binary form, against the x and y values of the center of each cell (these latter values were previously measured and stored in a look-up table). The correct cell (the one being pressed) was the one for which the distance between the x and y coordinates of the point of contact and the x and y coordinates of the center of the cell was a minimum.

A block diagram of the final design is shown in Fig. 7. CMOS bilateral switches are used to switch voltages to the touchplate and to switch the DAC between the comparator and the volume-level output. Another comparator, not shown in the diagram, sends an interrupt signal to the microcomputer when the touchplate is pressed.

When the touchplate is pressed, the x and y voltages are measured, converted to a cell number, and the new channel number is sent to the tuner. If the cell number were greater than 89, meaning that the bottom row was selected, the volume would be increased or decreased.

From the user point of view, the operation is very simple. To select a channel, the area of the touchplate in the immediate vicinity of that number is pressed. The channel selected is tuned within 0.1 seconds. To increase or decrease the volume, the lowest row is pressed on the right or left side.

The final solutions

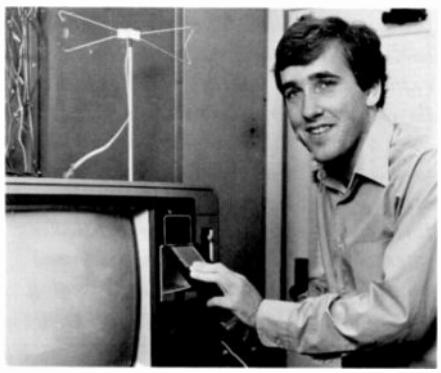
The system was found to function better, when we evaluated it, than expected, considering that the size of the touchplate's active area is only 4x4 cm, giving a cell size of 4x4 mm. One problem we did notice was that, near the edge of the active area, it was harder to make the desired selection because of the thickness of the mask. The solution to this problem would involve simply using a larger touchplate. This would make the device as easy to operate as a matrix of keys.

Another change to the system, for production purposes, would be to use another type of digital-to-analog converter. This would have little, if any, effect on the operation of the device, but would reduce the number of integrated circuits. There are two routes that could be taken: (1) do the digital-to-analog conversion entirely in software, letting the microcomputer take the place of the counter and digital comparator; or (2) use a single-chip DAC. At first glance, the former method may seem much more desirable, but a higher order filter, requiring more components, would be necessary to reduce the conversion time. This would probably offset most of the reduction in cost obtained by eliminating the counter and comparator. Thus, the latter method, utilizing a DAC chip would probably be the better one.

Conclusion

As a channel input device, this method has questionable advantages over the memory type tuning systems (a list of active channels is scanned using up and down buttons). However, since the touchplate may be manufactured in a variety of sizes with specified impedances and tolerances, it has potential uses in other areas. Some of the characteristics that make the microcomputer/touchplate combination a desirable input device are: (1) through software the number and configuration of the keys may be altered quite easily; and (2) the front cover on which the labels are placed could be replaceable allowing the keys to be relabeled. These two characteristics together would result in a very flexible keyboard for such applications as video games, portable data terminals, and point-of-sales terminals. Also, since the device is continuous, it could be used as a digitizer or as an inexpensive replacement for a joystick.

We have looked at the design of a microcomputer-controlled input device and discussed some of the advantages gained by giving the device some intelligence. The important point is that, because of the present technology and more specifically, because of the decreasing price of these microcomputers, it is becoming possible to place microcomputers right in small devices such as the one described to reduce the amount of circuitry (to do decoding for example) and to make the device appear more ideal.



Dave Sprague is a member of the Microsystem Research Group, whose main emphasis is microprocessor application, at the David Sarnoff Research Center. He started full-time employment with RCA in June of this year, having worked the previous summer at the David Sarnoff Research Center through a co-op program. It was during this summer employment period that the work described in this article was done. Dave's present work is also in the area of microprocessor applications to TV features.

Contact him at:
RCA Laboratories
Princeton, N.J.
Ext. 3178

H.A. Goldstand

Software techniques for a microprocessor-based data acquisition device

The microprocessor-controlled vehicle monitoring system is potentially applicable to all classes of vehicles — land, marine, and air — for both commercial and military applications.

Abstract: *A Vehicle Monitoring System (VMS), based on the RCA 1802 microprocessor, was developed for recording the use, condition, and maintenance action experienced by military vehicles in their day-to-day operation. Data are collected over a period of up to one month by an unattended on-board monitoring unit and are stored in a highly compacted digital form for later analysis. A multi-tasking operating system enables the real-time monitoring of 18 digital and 25 analog parameters. Data predefined as "non-significant" are discarded while significant data are saved in a 15-K byte storage area in the form of detailed profiles, summary indicators and histograms. Retrieval of data is accomplished through the use of an off-board cassette unit which can also be used to modify the control software of the monitoring unit. The data can then be analyzed by an off-line computer system to extract information relating to vehicle design and performance, effectiveness of maintenance programs and causes or precursors of vehicle component failures.*

Incorporating microprocessor control into a data acquisition system provides extremely flexible collection schemes. A record can be kept of an individual parameter when it contains some special significance, or of a result summarizing a critical relationship among several parameters. The Vehicle Monitoring System (VMS),* developed for the Tank Automotive Research and Development Command (TARADCOM), was designed by RCA on the basis of this principle.

The VMS is a research instrument currently installed in two military vehicles, the M35 2½-ton truck and the M113 armored personnel carrier, for the purpose of collecting data to investigate the high cost of operating and maintaining the land vehicle fleet. Data, collected by the VMS, describe the use, condition and maintenance exposure of its host vehicle. The hardware consists of two on-vehicle and two off-vehicle units shown in Fig. 1.

The VMS electronics assembly (VMSEA) and harness assembly are installed on the vehicle. The harness contains the transducers and switches installed throughout the vehicle and is specific to the vehicle type. The VMSEA consists of an RCA 1802 CMOS microcomputer system and signal conditioning hardware. The functions performed by the VMSEA are determined by the software and are redefined for different vehicles by changing the program. This is achieved by connecting the data retrieval equipment (DRE) to the VMSEA. The DRE contains two digiswitches and a hexadecimal display to allow operator communication with the VMSEA. A digital-cassette unit is used both to reprogram the VMSEA and to retrieve the collected data from the VMSEA memory. Other functions performed by the DRE include determining the health of the on-vehicle VMS assemblies and isolating faults, if required, and recording results of special tests [1,3] requiring unusual operator/vehicle interaction that would not occur during normal operation. The DRE is intended to be used by a special contact team. The second off-board unit, the maintenance-action input (MAI), allows a mechanic to enter data corresponding to any maintenance performed on the vehicle. The MAI unit would typically be kept at the vehicle maintenance shop.

Data storage techniques

The VMS monitors a selected set of vehicle parameters and records significant events. The recorded data can then be processed by an off-line processing center to evaluate a set of vehicle indicators that describe the operating history of the vehicle. The parameters monitored consist of analog, digital and pulse-train-type signals. All

*Under contract DAAK30-77-C-000k, sponsored by the Defense Advanced Projects Research Agency

Reprint RE-25-3-7
Final manuscript received Sept. 25, 1979.

Fig. 1. The hardware consists of two on-vehicle and two off-vehicle units.

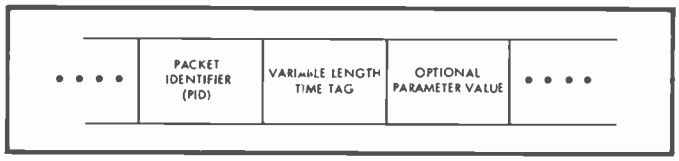


Fig. 2. Data are stored as compressed packets.

data, however, are encoded in digital format for on-board processing and storage. Fifteen kilobytes of memory are dedicated to storing processed data prior to transfer to the DRE cassette unit. In order to compress all of the vehicle data generated during a nominal thirty-day period into a memory of this capacity, a scheme was devised in which a portion of the data, particularly that surrounding a significant event, is retained in high resolution format while a summary indicator of all of the data is retained in a "binned" format.

High-resolution storage

The high-resolution format consists of time-tagged data samples entered into storage at the sampling rate and under the conditional requirements peculiar to each parameter. Engine speed, for example, is sampled at a 0.1-second rate and, in the absence of a uniform acceleration, is stored whenever it crosses one of a set of thresholds which differ by 200 RPM. Should a uniform acceleration be detected, only the end points are recorded.

Data are stored as compressed packets, as shown in Fig. 2. These packets consist of a fixed-length packet identifier (PID), a variable-length time tag, and an optional parameter value. The PID indicates the particular parameter and the conditions under which the data were taken. The time tag indicates the time at which the measurement occurred, and varies in length according to changes in a major increment (hours, minutes, or seconds) of the time since the previous packet. If the exact value of the parameter is of interest, it is also included in the packet and varies in length according to the degree of resolution required.

Bin storage

Since most of a vehicle's operating profile can be reconstructed adequately with lower temporal resolution, data storage capacity is saved by "binning." A bin is a reserved memory segment which records three types of measurements: accumulated time, event counts, and parameter values. The numbers in each bin normally accumulate for one week, although this period can be adjusted downward by software. At the end of the binning period, new bins are established and the old bins are compressed to minimize storage requirements.

Overwrite

All sampled data are actually entered into both high resolution storage and processed bin storage. High resolution data packets are organized as a circular buffer. When the memory capacity allocated to high resolution storage is filled, the memory contents are overwritten, starting again at the beginning of the buffer. Bin data for the first four binning periods (one month maximum total duration) are never destroyed. When more than four binning periods occur, an additional fifth bin area will contain data for the most recent (current) binning period.

The overwrite feature of high resolution storage is automatically inhibited upon the occurrence of any of a set of predefined events. Should an inhibiting event occur, a predetermined amount of high resolution data surrounding that event will be protected. In the absence of any inhibiting events, data memory will contain the most recent high resolution data plus all the binned data.

Maintenance-action data

Allocation of data memory for the different forms of storage is shown in Fig. 3. In addition to the current and packed binning and high-resolution areas, there are areas for maintenance-action data, trip data and special test data. Maintenance-action data are composed of entries made by the mechanic using the MAI, and include the hour the entry was made, a component identification code, a maintenance-action code and number of man-hours taken to perform the maintenance. A separate storage area for MAI data insures that the high-resolution data overwrite mechanism will not destroy maintenance data. In the trip-data area, trip-summary information is recorded. A trip is defined as an engine running with road speed present for at least two minutes. Data stored per trip include: the time of trip completion; duration of the trip; average and maximum road speed and exhaust gas temperature; and accumulated road shock and vibration level data. Special test results share storage space with trip data.

Many of the functions relating to the

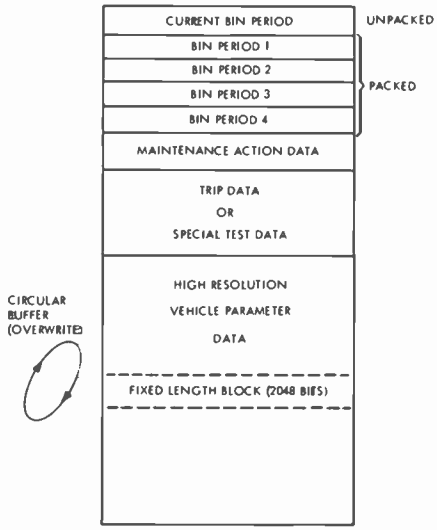


Fig. 3. Data memory is allocated for different forms of storage.

data storage techniques described are readily adjustable through software changes.

VMSEA hardware overview

The VMSEA hardware was designed to function without modification on any of its intended host vehicles outfitted with the proper harness. The hardware is tailored to a particular vehicle through software reconfiguration. The VMSEA contains an RCA 1802 microprocessor, 32 kilobytes of CMOS random-access memory (RAM) and a 256-byte programmable read-only memory (PROM) used to perform the initial program load via the DRE cassette unit. The VMSEA weighs less than 10 kg and is smaller than 13,000 cm.[3] It operates from vehicle power (10 to 32 volts), and consumes 15 watts while the vehicle is running. An internal battery maintains the real-time clock and memory for at least two weeks in the event that the host vehicle battery is removed. No vehicle parameters are monitored, however, in that condition.

Signals from the vehicle harness enter the VMSEA through 40 input channels. Dependent upon the nature of the signals, e.g., switch position, analog voltage, or pulse train, they will pass through status register, analog multiplexed A/D converter, counters or interrupt circuitry.

Software approach

The complexity of the data acquisition function would not be possible without the high level of control provided by the microprocessor and the associated software system. The large number of input channels and high-synchronous sampling rates require an approach more sophisticated than sequencing through the vehicle parameters one at a time, performing tests on them and storing the result. The software was, therefore, structured as six application tasks operating under the control of a real-time multitasking operating system.

After passing through the VMSEA hardware, incoming vehicle data go through four stages of software before finally being transferred to the DRE cassette unit. These stages consist of: (1) acquisition, (2) processing, (3) packing, and (4) retrieval. These stages are assigned to four application tasks which communicate with each other and with the incoming data stream via the operating system. This arrangement allows the acquisition phase to run in real-time, collecting up to four seconds of buffered data.

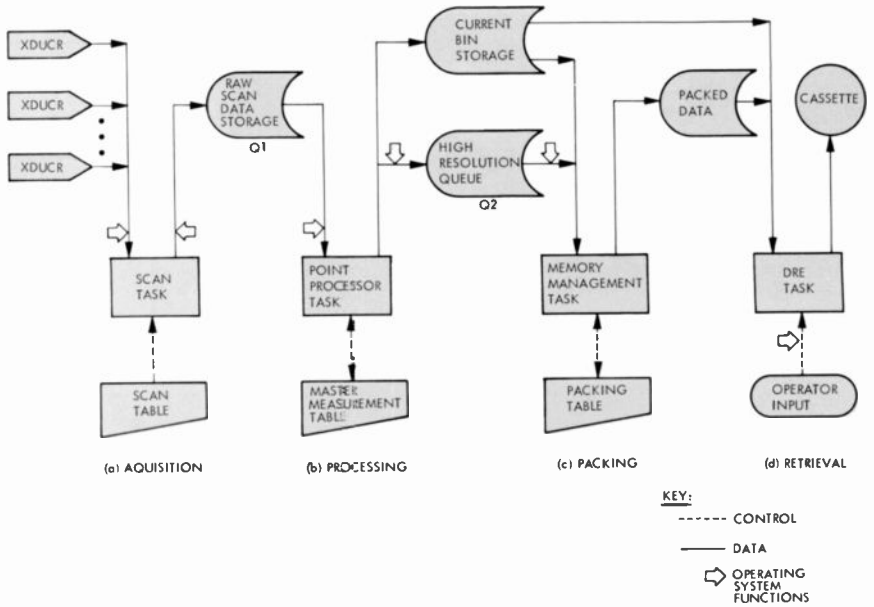


Fig. 4. Software handles data through acquisition, processing, compression, and transferral to cassette.

Figure 4 shows the four stages of data handling by the software. Data are input from the hardware by the acquisition stage, and selected for storage in data memory by the processing stage. Processed data may then be compressed by the packing stage and are finally transferred to cassette by the retrieval stage. The *scan* task reads raw data from the hardware and stores it in the raw data system buffer. Data are removed from the buffer by the *point processor* task which decides which information must be saved either in bins or high-resolution storage. Binning is done directly by the point processing task, while data to be packed in the high-resolution area are passed to the *memory management* task via the high-resolution queue. Data from the queue as well as the current bin area are packed by this task and stored in the appropriate packed-data area. Upon command from the DRE device, the DRE task moves the data from the data area to the cassette where they can then be transferred to an off-board computer system for analysis.

The remaining two application tasks are *peak detect* and *mode control*. Peak detect finds the peak-battery current and cor-

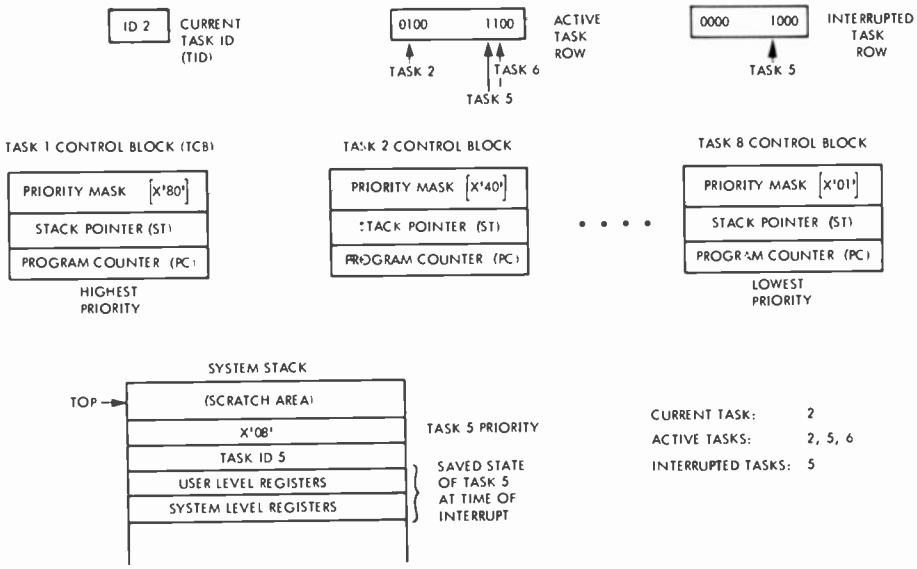


Fig. 5. Functions, executing at the lower level, can only be pre-empted by level I interrupt processing.

responding battery voltage when the starter is engaged. A separate task with a high priority was assigned to this function so that an immediate and virtually uninterrupted response could be made to this event. Mode control monitors several key vehicle parameters in order to set a summary vehicle state indicator, used by other tasks in interpreting vehicle parameters.

Operating system

Application tasks are selected and executed under supervision of the operating system. The operating system manages all system resources (e.g., A/D converter), intertask communication (e.g., raw data buffer), and priority control (scheduling). These functions are performed by system level subroutines, the scheduler, the interrupt handler, and the extended interrupt handler.

Scheduling

A key element of the operating system is the scheduler or priority-control program which, in conjunction with the interrupt handler, decides the next function to be executed. There are ten levels of priority in the VMS: two operating system levels and eight levels of application tasks. The operating system has a higher priority than all application tasks. The two operating system levels correspond to: (1) hardware interrupts disabled (highest priority), and (2) hardware interrupts enabled. Functions executing at the lower level can only be pre-empted by level I interrupt processing (see Fig. 5). These functions include extended interrupt processing, system subroutine calls, and the task scheduler itself. Removal of items from the interrupt queue, however, executes at the higher operating system priority level (i.e., with interrupts disabled).

The task scheduler is entered whenever the interrupt queue is empty. It operates on two lists or rows – the interrupt row and the active row. Any task which is a candidate for being run, including the task currently executing, is considered active. The task actually running is also called the current task. All active tasks have a bit, corresponding to their priority task, set in the active row. If a task has been interrupted and pre-empted by a higher priority task, it is placed in the interrupt row in addition to the active row. Dynamic information needed for each non-current task is maintained in either the task control

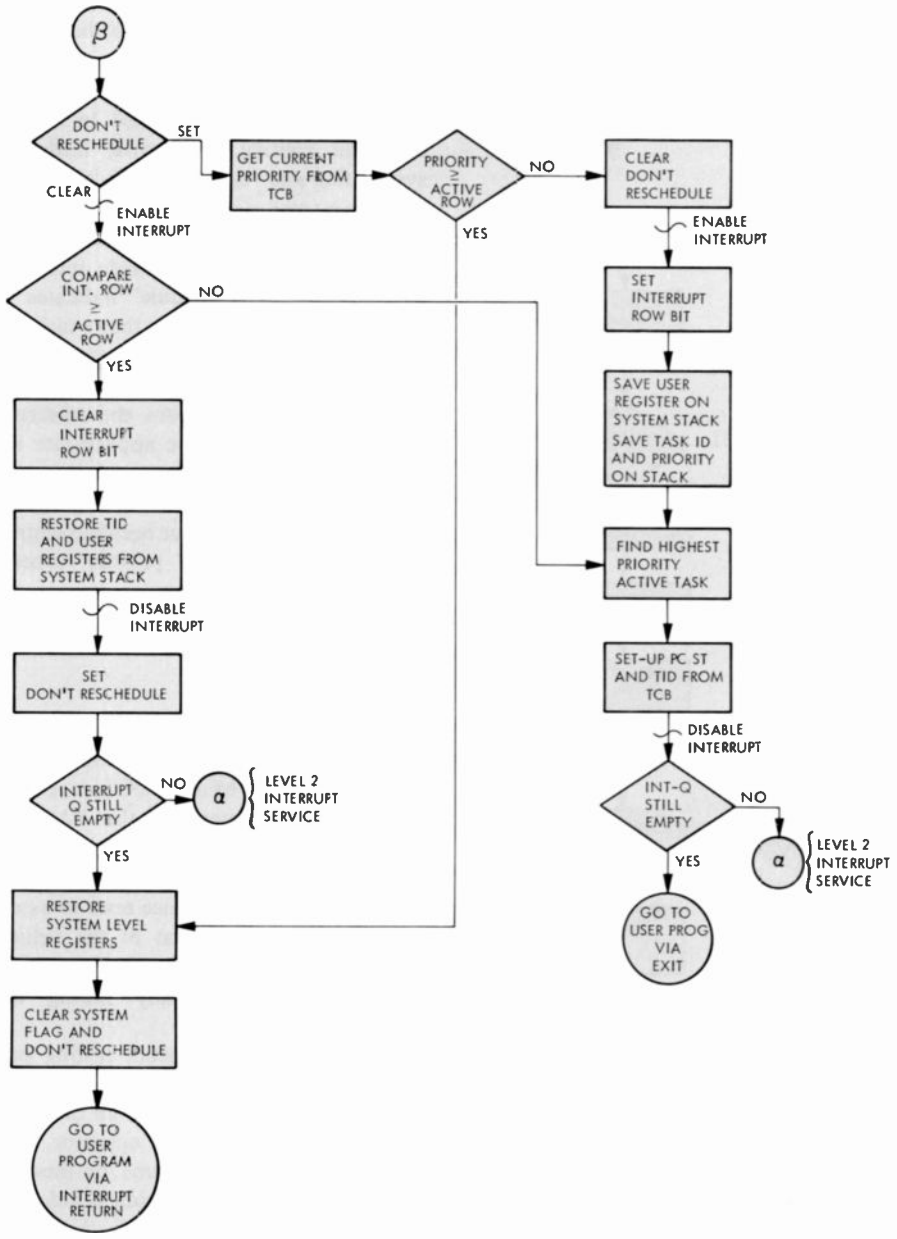


Fig. 6. The current task is task 2, given by the current task identifier, which points to the task control block of the current task.

block (TCB) for the individual task or in the system stack. The latter applies only to interrupted tasks.

Figure 6 shows a possible snapshot of the task status. The current or running task is task 2, given by the current task identifier (TID), which points to the task control block of the current task. The active tasks are 2, 5, and 6, as indicated by the appropriate bits being set in the active row. Task 5 had been interrupted during execution, and is, therefore, also listed in the interrupt row. Its status at the time of the interrupt is stored in the system stack.

The scheduling algorithm selects the highest priority task from a combination of the active and interrupt rows. Figure 7 describes the algorithm in detail. The scheduler first checks to see if the current task had been interrupted and partially saved on the stack. If so, "don't reschedule" has been set and a check is made to ensure that the priority of the current task is still higher than all other active tasks. Usually, this is the case and the current task is restored from the stack and continues execution. In the event that some other active task does have a higher priority than the current task, the current task must be fully saved on the stack and placed in the interrupt row. The scheduler then selects the highest priority task from the active row, restores its registers from the task's TCB, and makes it the new current task.

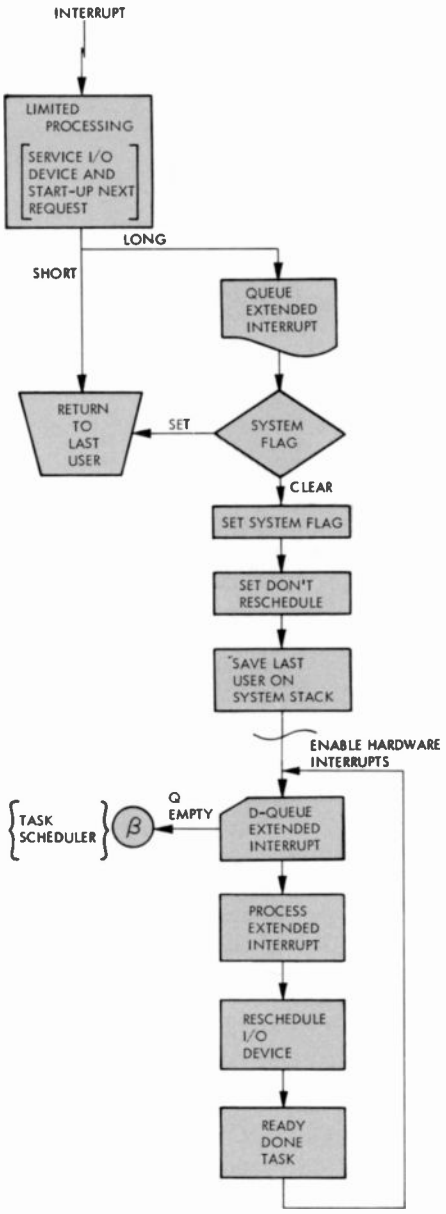


Fig. 7. The scheduling algorithm selects the highest priority task from a combination of the active and interrupt rows.

Interrupts

The operating system responds to events occurring in real-time through a system of interrupts. Possible interrupt sources include A/D output available, vehicle battery disconnected, and 40-Hz timing pulses. Figure 7 shows how an interrupt is processed. The response may be either "short" or "long." A "short" response has no effect on priority control (i.e., task scheduling). After processing the interrupt, the handler returns control to the task (or extended interrupt) in progress at the time the interrupt occurred. If, however, the interrupt could cause a waiting task to become active, a "long" response places a request for extended processing on the first in, first out interrupt queue. If the system flag is set, indicating that the operating system was in control at the time of the interrupt, control is returned to the interrupted routine. Otherwise, both the system flag and "don't reschedule" are set, the current task status is partially saved on the system stack, and control passes to the beginning of the extended interrupt handler. "Don't reschedule" indicates to the priority control routine that a task has been partially saved on the stack. The extended handler removes a request for extended processing from the interrupt queue and performs the appropriate action. This could include scheduling any waiting tasks which were readied by the interrupt. When the queue becomes empty, control passes to the priority control program.

Self-test features

It is not reasonable to be able to repair a complex electronics system, such as the VMS, in the field, by a service team, except by gross replacement of the VMSEA, backup battery, transducers or harness assembly. The VMSEA, however, will conduct routine confidence tests to assess its own health and that of the vehicle harness assembly. Test results are acquired and stored automatically during unattended operations.

The results of confidence testing performed during the most recent binning period can also be displayed on the DRE unit at the request of the operator. This enables him to make some on-the-spot corrections such as replacing a defective transducer or repairing a cable. A dummy harness can be substituted for the actual harness assembly to determine if the fault persists even while the dummy harness supplies a known test signal. A self-test routine used in conjuction with the dummy harness assists in isolating faults to a specific analog or digital channel of the harness.

Conclusion

The Vehicle Monitoring System has been in operation since 1978 collecting data to establish a data base for statistical analysis. It is expected to furnish data impacting the design, operation and maintenance policy of military vehicles. As a result of the analysis, it may be desirable to modify the nature of the exact vehicle data required and the conditions under which they are stored. The ease of reprogramming the VMSEA assembly makes this a feasible and reasonable approach. The general purpose hardware structure linked with microprocessor-controlled tailoring to achieve specific goals makes the VMS a powerful and useful tool. The exploitation of microprocessor technology provides adaptability and flexibility through software rather than hardware redesign. Although presently implemented on only two vehicles, the VMS is potentially applicable to all classes of vehicles — land, marine, and air — for both commercial and military applications.

References

1. Fineman, H.E., Fitzpatrick, T.E., and Fortin, A.H., "Simplified Automotive Test Equipment Through the Use of Advanced Electronics," *NEREM 74 Record* (Oct., 1974).
2. Hadden, S.C., Hanson, R.E., and Stewich, M.W., "Vehicle Monitoring System," The 26th Meeting of the Mechanical Failure Prevention Group, Chicago, Ill. (May, 1977).
3. Hanson, R.E. and Fischer, H.L., "New Techniques for Automated Engine Diagnostics," *Autotestcon '77*.
4. Yourdan, E., *Design of On-line Computer Systems*, Prentice Hall, Englewood Cliffs, N.J. (1972).

Harvey Goldstand, a Senior Member at RCA, Automated Systems, has worked in software development of security and building management systems as well as non-electronic test equipment. He headed the programming team and designed the executive for the Vehicle Monitoring System and is currently involved with microprocessor-based automated test equipment for the XM-1 tank.

Contact him at:
Automated Systems
Burlington, Mass.
Ext. 2176

A. Kaplan|D.L. Sherwood

MASS — a modular ESM signal processor

Plug-in modules provide flexibility and efficiency for design evolution during the long development cycle and for future requirements of military systems.

Abstract: *The long development cycle for military systems, combined with the highly dynamic environment in which they must operate, present a major challenge to the system designer—to design a system which will not be obsolete before it is fielded and which can evolve after deployment to meet future operational requirements. This paper presents the architecture of an Electronic Support Measures (ESM) signal processor designed to meet this challenge. The MASS processor (Modular Adaptive Signal Sorter) features a completely modular organization in which the types and number of modules can be selected to match the specific application. Growth capability is provided by adding modules for increased throughput or developing new modules for increased functional capability. The internal control of the processor adaptively monitors the status of all processing modules as part of a dynamic resource allocation function. This, in turn, provides an inherent fail soft operation.*

In addition to the demanding performance requirements that are imposed on ESM systems by the very dense and complex signal environment that exists today and that can only get worse in the future, the Modular Adaptive Signal Sorter (MASS) has some unique requirements which address the realities of the military system development process. Military systems must undergo a very long research, development, test and evaluation cycle before production begins and equipment reaches the field. For Electronic Support Measures (ESM) systems, in particular — in which the operational problem is constantly changing — this long development cycle creates a problem: Systems can be obsolete by the time they are fielded. This poses a challenge to the system designer, to conceive system organizations which are inherently capable of being upgraded to meet a constantly changing threat environment. This is one of the primary objectives of the MASS processor, a state-of-the-art generic ESM signal processor which RCA is developing under the sponsorship of the Electronic Warfare Laboratory of the U.S. Army Electronics Research and Development Command at Fort Monmouth, N.J.

Reprint RE-25-3-8
Final manuscript received Sept. 17, 1979.

MASS design objectives

Ten years ago it appeared that the required flexibility could be provided by software control of ESM systems implemented with general purpose computers. Unfortunately, history has proved us wrong. Indeed, the evolution of technology has reversed this view. Today software has proven to be very inflexible and very expensive. Hardware is becoming less costly, and we are learning how to use hardware modules in organizations which provide extreme flexibility.

The MASS design is a processor organization featuring plug-in expandability. Throughput can be increased by inserting more of existing modules. Functional capability can be enhanced by developing new modules which are compatible with the basic processor organization without any redesign. This type of organization provides an inherent fail-soft operation. It fosters a variable configuration processor which can be used in many different applications while retaining spares commonality. Functional objectives are: (1) real-time operation in the very dense signal environment that exists today, and will get worse in the future; (b) a comprehensive capability of handling both time and frequency diversity signals which will become more common in the future; and (3) special emphasis on acquiring short duration pop-up signals. A final objective in the MASS design is to demonstrate its capability for technology insertion. To do this, the processor is being designed for implementation with custom VLSI devices to minimize the time required for advanced technology to be incorporated in field equipment.

Processor organization

To place the MASS processor in the context of a total system, Fig. 1 outlines a conceptual ESM system with an antenna set serving a wideband receiver which measures for each pulse received: monopulse frequency, azimuth, pulse width, time-of-arrival, and pulse amplitude. This stream of monopulse data is the input to the MASS processor. MASS reports processed data to the user, receives

user control and priority information, controls the system front end, and in ECM applications provides real-time data to a jammer.

The MASS processor is organized as a distributed architecture in which a set of parallel microprocessor-based modules operates between two data busses. One carries high speed pulse data, the other carries control data. Figure 2 presents this architecture.

Signal sorting and resource allocation

The first two modules in MASS are the frequency-azimuth histogram and the acquisition-control modules. These units serve two functions in the processor. One is to provide first-level frequency-azimuth sorting of the signal data, and the other is to provide a resource allocation function for the remaining modules. With a distributed processing system such as this, there must be some rationale for assigning signals to the various modules. In MASS, the distribution of signals in frequency-azimuth space is used as a basis for making this assignment.

With this method of control, the processor is adaptive to a highly variable signal environment. The acquisition-control module monitors the histogram as it builds. It assigns frequency-azimuth cells or groups of cells to individual acquisition processors, based on the pattern which develops, or, if necessary, to the exotic processors. In the histogram that is shown, the group of cells on the left is typical of what would be perceived for single signals that are separable using only frequency and azimuth. The group of cells at the bottom is a pattern that would be seen for multiple signals that are close in frequency and azimuth and which must be deinterleaved in the time domain. The pattern on the right is what would be perceived for a frequency hopper. Based on these patterns, the acquisition-control module will assign sorting tags to cells or groups of cells and pass these sorting tags to the appropriate acquisition processor or exotic processor.

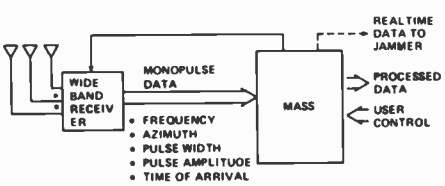


Fig. 1. MASS in a conceptual ESM system.

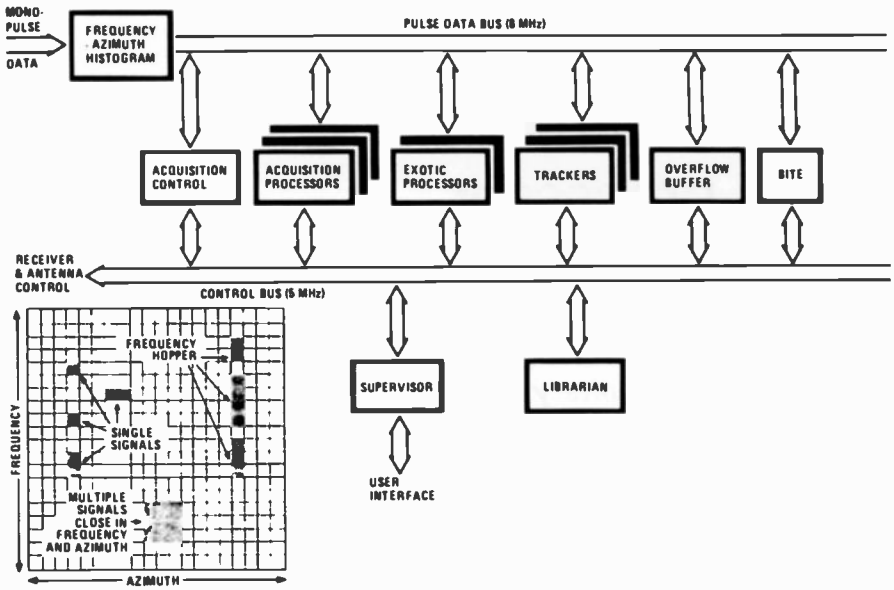


Fig. 2. MASS organization.

Once this assignment has been made, the frequency-azimuth histogram, which has been storing the data as the histogram builds, will transmit the data along the pulse data buss with the appropriate sorting tag appended to the data block. Only the module which has been assigned that sorting tag would accept the data off the buss and will accept all subsequent pulses with that sorting tag. The frequency-azimuth-histogram module also has a blocking mode to exclude data from cells that were previously processed and to prevent high pulse rate signals such as pulse doppler from saturating the processor.

Signal time-domain analysis and tracking

The acquisition and exotic processors perform time-domain analysis of the signals which have already been sorted in frequency-azimuth space. Their function is to separate multiple interleaved signals, if necessary, and to characterize each one in terms of its time-domain characteristics. Once pulse trains are separated, the monopulse measurements of frequency and azimuth can then be combined to improve the accuracy of these parameters when they are reported. The algorithms selected for these functions are discussed later in the paper.

The trackers perform several functions in the processor. A common element of all of these functions is the separation of individual signals from the composite signal received, based on signal characteristics provided by the acquisition processors. Once separated, these data can be used for scan analysis, or for detecting changes in emitter characteristics that occurred after acquisition (i.e., PRF or scan pattern). They can also be used to detect new signals that come up within the designated frequency-azimuth cell after the acquisition processor has completed its function on that cell. The tracker can also be preset to the parameters of specific signal types for warning receiver applications.

Support functions

The overflow buffer and BITE are two modules that do not contribute to the functional capability of the processor but that are essential to its proper operation. The overflow buffer serves to even out the data rate to the various processors. A radar signal environment is highly variable in its short term data rate. There can be as much as 4:1 ratio between short-term, peak-data rate and the average-data rate. The processor is designed for the average-data rate and during these short term peaks, pulses will be captured in the overflow buffer. When the peak is over, they will be retransmitted to the modules. Built in test is an essential feature for any system. In this architecture, BITE is a continuous function activated whenever the processor is less than 50% loaded. The BITE module generates test-signal data and transmits them over the pulse-data bus to the module under test. By monitoring the signal reports out of that module, the BITE module can rapidly complete a rigorous functional go/no go test.

The librarian provides file handling

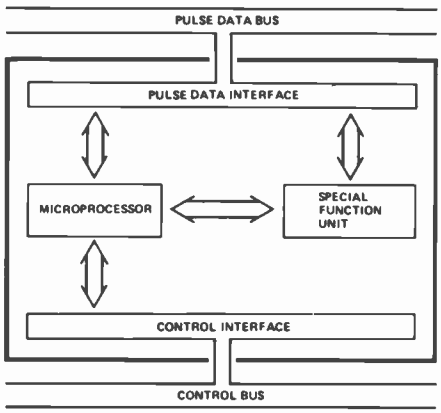


Fig. 3. Common module organization.

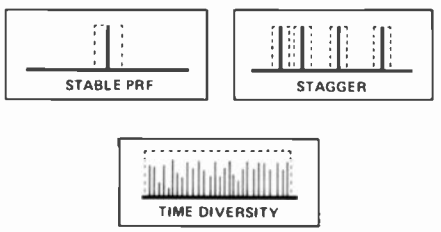


Fig. 4. Histograms of time difference between successive pulses.

capability for the processor. It typically has a file of signal characteristics to use in signal recognition and also maintains a file of the current active signal environment. The supervisor performs high-level control of the processor, communicates finished reports to the user, and controls the front end of the system. It also monitors processor status. When power is turned on, the supervisor polls all of the physical card slots to determine which modules are installed and operative. This status information is continuously updated by the BITE results. If a failure is detected, the processor continues to operate with the remaining modules, providing a graceful degradation of system capability.

The final two modules in the system are the two busses themselves. To achieve true modularity in the system, the buss structures have to be capable of being upgraded and evolving to meet future requirements. The pulse-data buss at the top of the diagram carries high speed pulse data with a 32-bit word length and 8-MHz clocking rate to provide a total data rate of 256 megabits per second. The control buss at the bottom operates with a narrower word length and lower clocking speed.

Module hardware organization

Most of the modules have a common hardware organization. They vary only in the particular ROM firmware which is used with the module and in the amount of memory employed. Figure 3 presents this organization. These modules consist of four main elements. They are the pulse data interface, the control buss interface, a common microprocessor, and a special function unit configured to optimize performance for the types of pulse data processing required for MASS. The pulse data interface and control interface will be common to all modules that are used in MASS. This is a fundamental requirement of modularity. All of them use the same microprocessor; however, the amount and type of memory varies from one to another. The librarian has a large but relatively slow memory. The tracker has a smaller, but much faster memory. The acquisition modules utilize the special function unit.

Processing algorithms

In evaluating the various algorithms that are available for signal acquisition and measurement, it was concluded that the extreme flexibility in the hardware organization must be complemented by flexibility in processing algorithms. The most powerful and flexible technique available and the one which will provide the most growth capability in the future is based on histogramming.

Time- and frequency-diversity signals

To illustrate the power of this technique, Fig. 4 presents some typical histogram patterns of the time difference (delta T) between successive pulses of various radar signals A signal with a stable PRF produces a single line in the histogram. A staggered signal has multiple lines in the histogram. A signal with a high degree of time diversity produces a random, but bounded pattern.

Note that jitter on the stable signal or the stagger signal broadens the lines, but they are still automatically recognizable. Figure 5 shows additional analysis that can be done on the time-diversity pattern. By calculating the derivative of delta T and generating a histogram of these values, unique recognizable patterns for triangular, and sinusoidally modulated PRI can be detected. For signals which exhibit frequency diversity, this same type of processing can be applied to the frequency data. An important feature of this method is that it uses the stability of the histogram to govern its operation. It will not report a measurement until the pattern satisfies a consistency test. With a simple stable PRF, this occurs with only a small number of pulses. A time-diversity signal requires a larger number. This feature provides several important benefits. For stable signals, which comprise the majority of the signal environment today, the algorithm is very fast and efficient. For complex signals, the algorithm adapts to the complexity of the signal and continues processing until it can report a reliable measurement. Both frequency and time-diversity signals will increase in importance in the future. This type of histogram processing provides the flexibility for dealing with new signals which will enter the environment in the future.

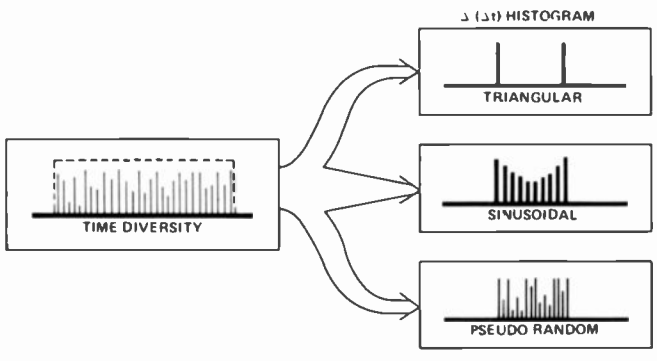


Fig. 5. Time diversity histogram analysis.

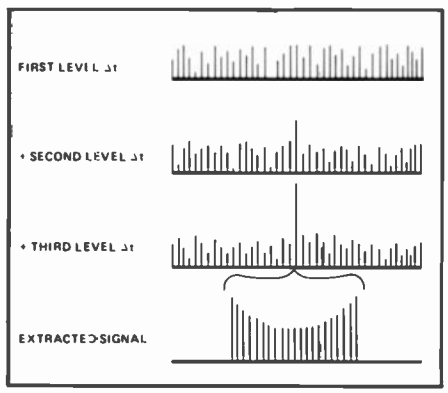


Fig. 6. Histogram deinterleaver.

Can *any* problem be solved by a computer?

A popular conception is that with enough speed and memory any problem can be solved by a computer. Like any other technology, there are theoretical limitations on a computer's capabilities. Before any problem can be solved by a computer it must first be translated into an algorithm. Algorithms are the necessary procedures that are used in programming a computer. They are the set of instructions that tell the computer how to perform an operation.

An algorithm is a set of rules that precisely defines a sequence of operations in such a way that each rule is effective and definite and that the sequence terminates in a finite time. Many attempts have been made independently to mathematically characterize the class of functions computable by algorithm. There are problems where mathematicians have proved that it is impossible to construct an algorithm for their solution.

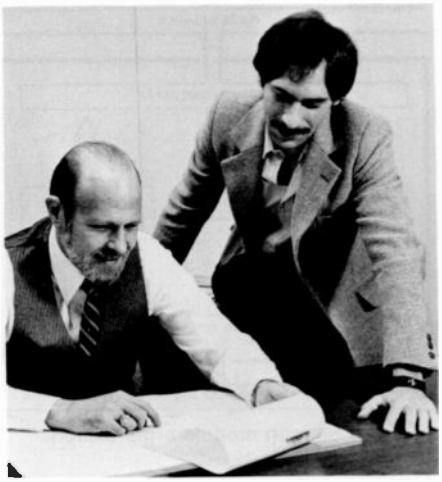
The early descriptions of algorithm given by researchers can be categorized in three ways. The first is abstract computing machines such as the Turing machine. The second is a formal construction of computing procedures such as the Thue system. The third is a formal construction that yields classes of functions such as recursive functions.

The Turing "machine" is an abstract mathematical model that gets its name from A.M. Turing, who introduced the idea in 1936 in a paper on the theory of computation. The Turing machine algorithm is composed of rules of a restricted set that are obeyed by a device called a Turing machine. Since the Turing machine is operated by a human being who carries out an algorithm, the Turing machine may be conceptualized as having an infinite paper tape that is divided into squares that contain one symbol each from a finite alphabet of symbols.

The Turing machine can read or write a symbol onto a square of the tape by means of a head that is directly over the square. The head may be moved right or left, one square at a time, to place the head over any square. Aside from the tape memory, the Turing machine contains one other cell that holds a symbol called the state of the machine. This symbol is taken from a finite alphabet that is different from the tape alphabet. The algorithm for the Turing machine consists of a set of rules that define tape operations, head movement, and change of state. The state of the machine tells which instruction the Turing machine is carrying out at any given time. The Turing machine actions are determined completely by rules that specify the actions to be taken for each possible observed symbol and each possible state.

The Turing machine is a tool for evaluating whether a problem is computable. The limitations imposed by noncomputable problems have not yet hindered the widespread application of computers, but like all technologies, as greater levels of performance are sought, these theoretical limits will ultimately have to be faced.

B.L.S.



Al Kaplan, seated, and Dave Sherwood review a MASS printout.

Al Kaplan is Manager, Advanced Programs in GCS Special Systems. Throughout his career, he has focussed on the development of state of the art real time systems using advanced signal processing technology. These systems have included weapon guidance, ballistic missile defense, air traffic control and command and control systems as well as his current activities in ESM, ECM and intelligence systems.

Contact him at:
Special Systems
Government Systems Div.
Camden, N.J.
Ext. 2540

Dave Sherwood is a Senior Engineer in Advanced Programs, GCS Special Systems, responsible for the detailed system design of MASS. In the past, he has been responsible for the design and integration of airborne, ground, and naval ELINT and ECM Systems. Recently, he has focussed his activities primarily on digital system architectures and signal processing techniques for distributed microprocessor systems.

Contact him at:
Special Systems
Government Systems Div.
Camden, N.J.
Ext. 3907

Signal deinterleaving

This same type of histogram processing can also be used for signal deinterleaving as illustrated in Fig. 6. The first line shows the noise-like appearance of the histogram of the delta Ts of multiple interleaved signals. However, by calculating the second level difference (the time difference between every other pulse) and adding it in, then calculating the third level difference (between every third pulse) and adding it in, etc., individual lines emerge in the histogram which correspond to the most common delta Ts that are seen. These are used as trial values for examining the data and evaluating single-signal histograms. These histograms can be built in parallel for maximum throughput. An extensive analysis of these algorithms has been performed. They provide both extreme flexibility in deinterleaving multiple time-diversity signals and efficiency in handling stable signals.

Summary

MASS is a modular design specifically configured to support system evolution in the field. It is a distributed microprocessor architecture, designed to handle a very dense signal environment in real time with particular focus on exotic signal handling. It features extreme flexibility in both hardware organization and in processing algorithms.

The plug-in modularity of MASS is an effective approach to assuring that our field forces are equipped with the most modern processing equipment available, despite the long development cycle for military systems.

D.R. Carley

Microprocessors in automotive electronics

Automatic Placement and Routing allows the customer to design engine-control logic based on standard cells while reducing development time.

Abstract: *Emission and fuel efficiency requirements have accelerated the use of electronics in engine control. Analog vs. digital, and NMOS vs. CMOS devices are compared for use in engine-control systems. The RCA integrated-circuit layout, Automatic Placement and Routing (APAR), which interconnects standard logic cells by a computer to provide a custom design, is discussed. The use of microprocessors in two system examples is demonstrated. The conclusion points out the future expanded usage of microprocessors in automotive electronics.*

Electronic-based systems are becoming increasingly important in automobiles. The requirements for better fuel efficiency, mandated emission standards, and the demand for more sophisticated driver information and display systems have all led to a rapid increase in solid state device usage. The most powerful systems are based on microprocessors.

Solid state devices were first successfully used in radios, followed by power control applications such as breakerless ignition, alternator diodes and electronic voltage regulators. Integrated circuits got their start in seat belt interlock circuits and automotive clocks. These early applications showed that solid state devices are reliable and cost-effective solutions to automotive problems. The emission and fuel efficiency requirements in effect in the United States have accelerated the use of electronics in engine control. All of the major manufacturers in the United States will have microcomputers in at least some models in the 1980 model year. Much more extensive use is planned in the 1981 model year.

Reprint RE-25-3-9
Final manuscript received Aug. 16, 1979.

Engine control systems — analog or digital

Many of the earlier solid state designs for engine control used analog techniques. This was a natural starting place due to the successful experience in entertainment applications and the analog nature of many of the signals needed for the engine to control the combustion process. Some analog-based systems for spark timing control were put into production; however, most newer systems use digital techniques. This evolution toward digital is due to a number of factors:

- Digital LSI technology is evolving rapidly;
- Digital systems are less sensitive to supply voltage and temperature variation;
- Precision adjustment and timing are not required; and
- Broader system design tolerances are possible.

This trend to digital has led to two system approaches: standard, or more usually custom, integrated circuits for a specific well-defined function; and microprocessor-based systems for more complex systems.

The basis for use of a microprocessor is its flexibility. The system capability is easy to expand by increasing the memory and occasionally adding additional input/output capability. Different engine needs can be met by software and Read-Only Memory (ROM) changes; hardware changes are not usually needed. In some systems, the required modifications for different engine parameters can be done with Programmable Read-Only Memory (PROM).

NMOS vs. CMOS

There are a number of different applications for microprocessors in the automobile, each with its own special requirements. The largest usage in the United States will be in engine control applications. To the usual microprocessor selection criteria of architecture, support parts and developmental aids, must be added the difficult environmental and reliability needs of automotive applications. Today's automotive processors are built using either NMOS or CMOS technology. Figure 1 shows the relative LSI capability of the two technologies; note that NMOS has had a significant lead in transistor count but the advent of SOS technology has narrowed the gap.

The advantages of CMOS

CMOS has several important performance advantages over NMOS. Table I summarizes the differences. Of particular significance is that it has the best noise immunity of any LSI technology, low power consumption and wide operating voltage range as well as the ability to operate from −65 to +125°C. These factors contribute to improved system reliability. The noise immunity and operating voltage range make CMOS more tolerant of

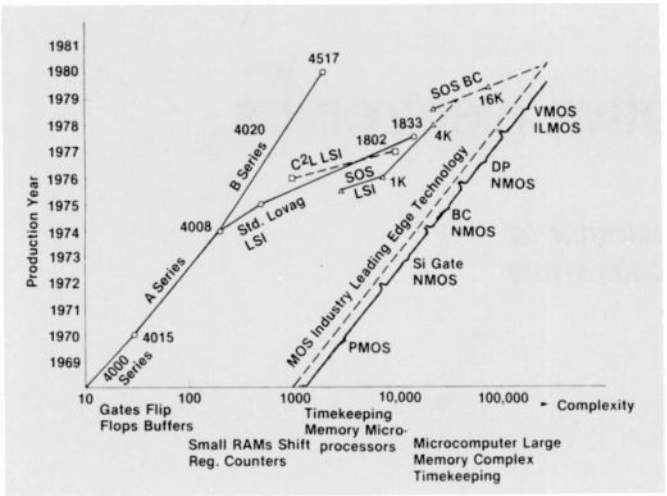


Fig. 1. NMOS leads in transistor count, but SOS technology narrows the gap.

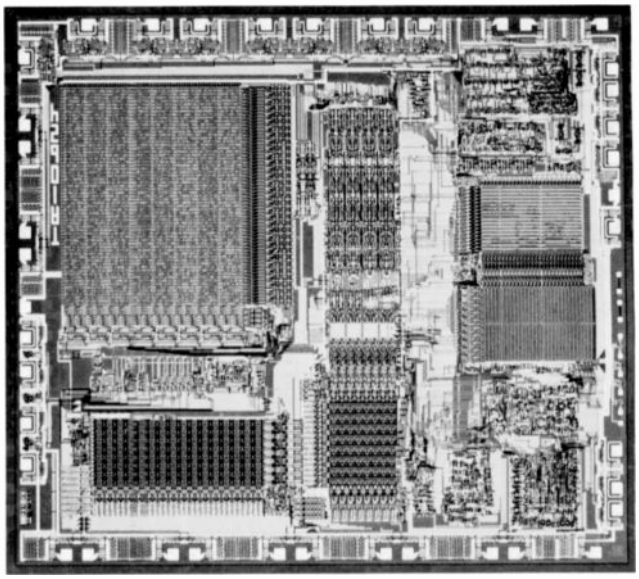


Fig. 2. The CDP 1804 is the most advanced CMOS microcomputer developed by RCA.

automotive voltage transients and low voltage during cold start and make it resistant to noise pulses. The low power consumption and high maximum operating temperature make CMOS devices cooler and typical operation is further away from the maximum permissible temperature for any ambient temperature. This reduction in maximum junction temperature contributes directly to improved long term reliability; it also allows location of CMOS systems in the engine compartment rather than the passenger compartment or somewhere in a body cavity. This alleviates problems with ground loops and simplifies connector construction as well as minimizing lead lengths. Mounting the electronic packages in the engine compartment also minimizes packing redesign due to body styling changes.

A second order advantage of the wide temperature and voltage range is the capability to operate devices during life testing and burn-in at accelerated conditions. Acceleration factors of 50-100 times can be obtained by the combination of increased voltage and temperatures. This acceleration allows quick reliability evaluation of new processes and designs.

A CMOS microcomputer

The most advanced CMOS microcomputer developed by RCA is the 1804 microcomputer. It is an 8-bit CMOS/SOS device that has a powerful instruction set, 2-K bytes of ROM and 64-bytes of RAM; the total transistor count is approximately 32,000 (Fig. 2). For most engine control systems only two LSI circuits will be needed, the 1804 microcomputer and a custom I/O circuit. Expansion is easily accomplished since additional memory may be added with no interface circuits (Fig. 3).

Custom I/O development

For most engine control systems it will be necessary to develop a special purpose interface circuit for added logic capability and to provide adequate real-time response and an interface to the various sensors and outputs. In general, this interface circuit is different for each customer and incorporates special features and circuitry that are proprietary to each user. RCA has developed an automated integrated circuit layout capability that has proven to be very successful in producing custom LSI circuits.

APAR layout

This technique, called Automatic Placement and Routing (APAR), is based on a large number of standard logic cells that are placed and interconnected by a computer. The advantages of this technique are: (1) the customer can design the logic based on the standard cells much the way he would produce a breadboard with packaged ICs; and (2) the total development time is significantly reduced with a high probability of working samples on the first pass.

The present standard cell library consists of over 60 cells of logic functions, layout geometries such as interconnecting tunnels, bond pads, high current drivers and some special cells for automotive applications such as voltage comparators and analog switches. Each logic cell has a

Table I. Comparison of NMOS and CMOS performance.

Characteristic	*CMOS*	*NMOS*
1. Quiescent power	1 — 100 μwatts SSI to LSI	100 — 1500 milliwatts MSI — LSI
2. Operating power @ 1 MHz	1 — 100 milliwatts SSI — LSI	100 — 1500 milliwatts MSI — LSI
3. Noise immunity	30% of supply voltage	10% of supply voltage
4. Supply voltage range	5 ± 20% 8 ± 50% 3 — 20	5 ± 5% 5 ± 10%
5. Temperature range	−55°C to 125°C	0 — 70°C −55° to 125°C special selection

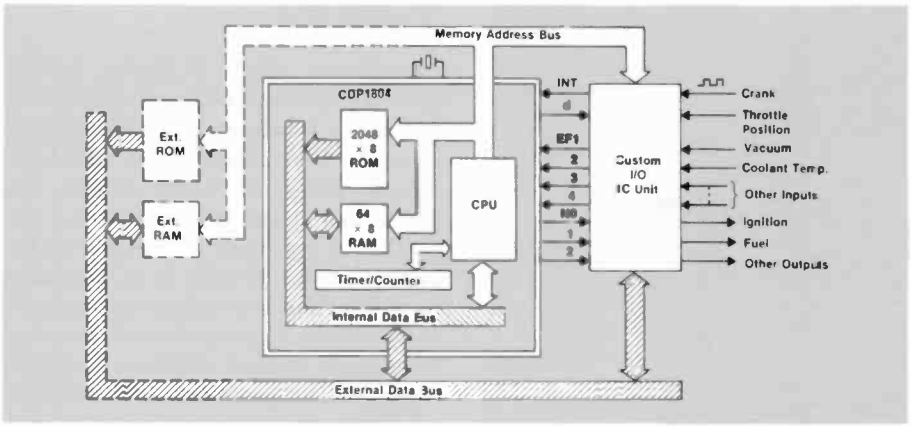


Fig. 3. Expansion of the 1804 microcomputer-based engine control system is accomplished easily.

fixed height of 315 microns, the width of the cell depends on the complexity of the logic function and varies from 60 to 400 microns. The connections to the cell are made at the top and bottom of each cell. Figure 4 shows a typical cell layout.

APAR design process

The design process, using the library, is straightforward. There is a data sheet for each cell (Fig. 5) that gives the logic function, cell interconnection data, and basic performance. The integrated circuit is then configured using the cell library by the logic designer. The next step in the process is to enter information defining the cells and their interconnections into the computer data base. This common data base is used for logic simulation, cell placement and interconnection and generation of the test program.

The output of the APAR run is checked for design rule violations and a check plot is prepared. There are generally some manual modifications made to reduce the overall die size using interactive graphic equipment. The final file is used to drive electron beam mask-making equipment to produce the required photomasks. Following conventional wafer processing, the test program generated via the data base is used to check functionality.

Since the process is fully computerized after generation of the logic diagram, the process is fast and error-free. The throughput time is largely determined by queuing and is approximately 1/4 of the time required for a layout using conventional drafting procedures.

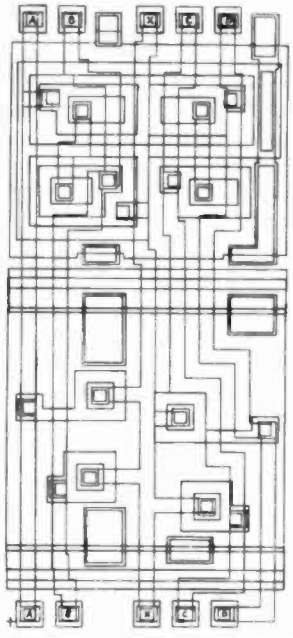


Fig. 4. The connections to the APAR logic cell are made at the top and bottom.

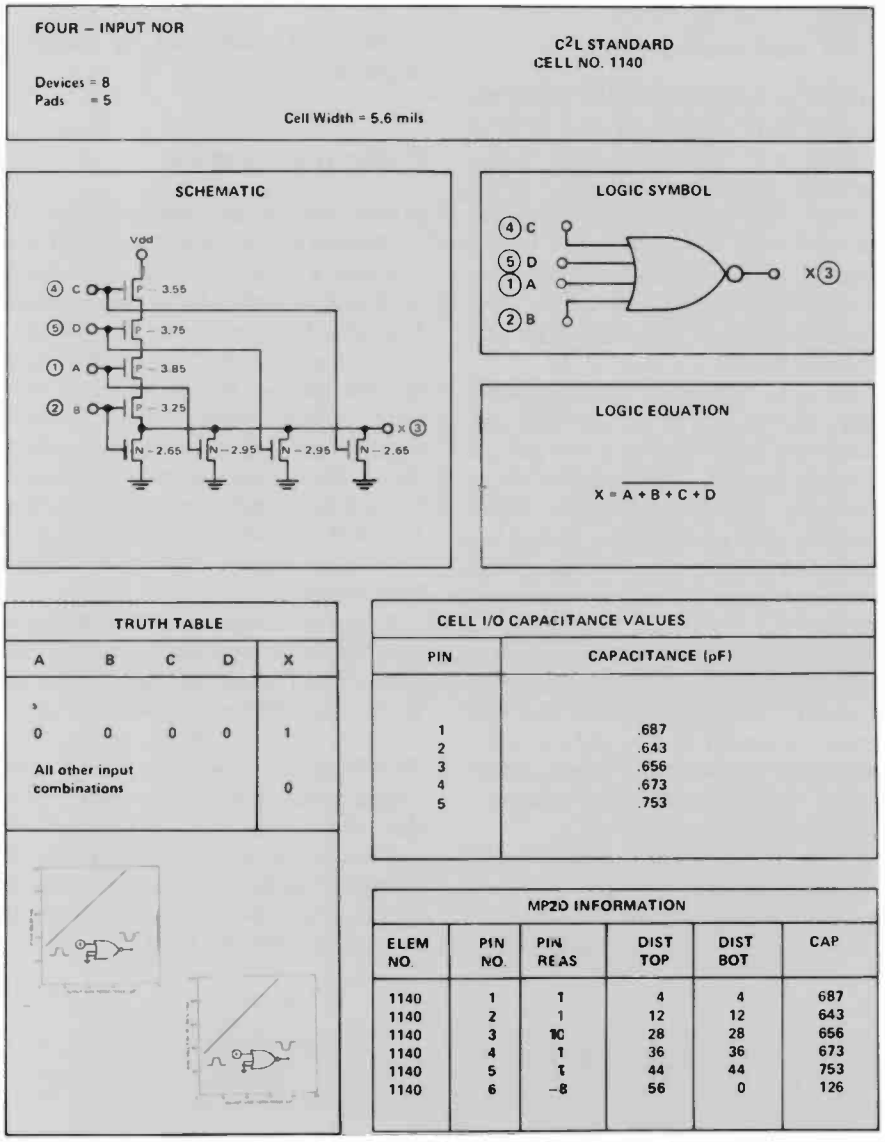


Fig. 5. A data sheet, for each APAR cell, gives the logic function, cell interconnection, and basic performance.

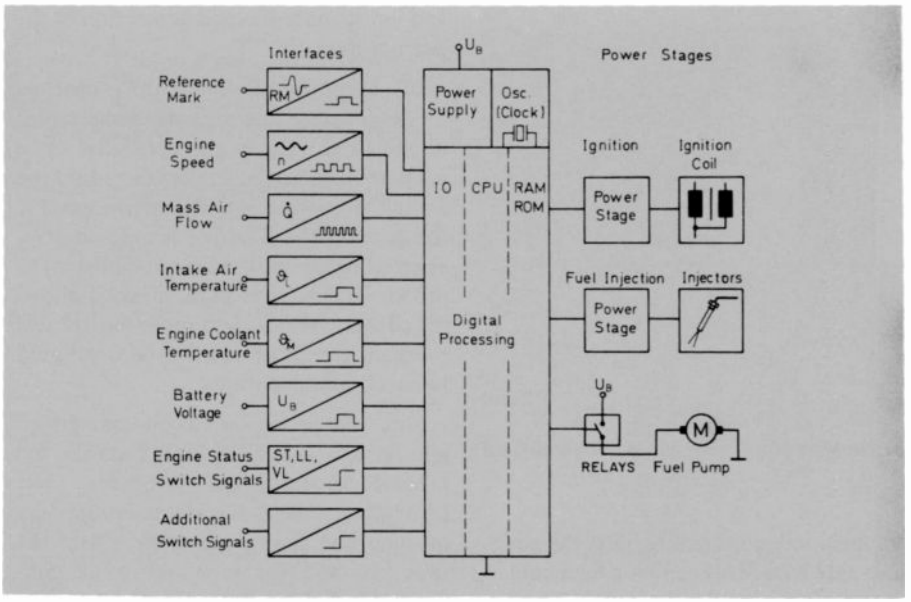


Fig. 6. Block diagram of Bosch engine-control system.

APAR device simplicity

CMOS is an ideal technology for computer layout. All static circuitry is used; dynamic clocking and multiphase logic design procedures usually used in NMOS are not necessary. The current levels are low, and race conditions are avoided so that cell placement and node loading do not affect the logic functionality. Maximum clock frequencies of approximately 5 MHz may be used at 5 volts.

The upper limit of device complexity using this technique is currently approximately 3000 transistors in a random logic configuration. The size consumed per transistor is approximately 10,000 microns, about 1.6 times that needed in a hand-packed layout in the same technology. Thus, the die cost is somewhat higher.

Completed circuits are then used to check out the design in the automobile. Generally, there is substantial time between design verification and the start of production; however, if there is not, APAR circuits can be produced in substantial quantities for production.

If time allows, the fully proven APAR circuit may be redone using manual techniques to achieve smaller die size. This manual layout is done using the APAR circuit as a guide and is, therefore, simplified.

System examples

Microprocessors are now used in several production engine control systems. Figure 6 shows a block diagram of a system developed by Robert Bosch GmbH used by Bayrische Motoren Werke AG (BMW).

This system controls both fuel injection and the ignition. It is based on the RCA 1802 microprocessor with standard program and read/write memories, and an APAR input/output chip designed jointly by RCA and Bosch. The precise amount of fuel and the correct spark timing are determined by the information received from the sensors.

A second example is a system developed by Chrysler Corporation to control spark timing based on six different inputs. The system uses only four LSI parts, the 1802, a standard RAM and ROM and a custom APAR chip. An interesting feature of this system is that it contains a small PROM that can handle 4, 6 and 8 cylinder applications as well as allowing for some change in engine control constants. The system is mounted in the air intake to the air cleaner under the hood.

Conclusion

The present needs of engine control systems vary from 25,000 to 50,000 devices which are usually contained in several packages. As more functions are added, for instance anti-knock, cruise control, idle speed and transmission control, the number of transistors needed will be increased toward 100,000. In such systems most of the transistor count is concentrated in the memory with one transistor per bit in the program memory and five or six per bit in the read/write memory. As an example, with program memory of 7-K words and 256 bytes of read/write memory, a total memory transistor count of about 70-K transistors would be needed.

The continual improvement in LSI technology will be used to increase the reliability of the system. Automotive users will not continue to be at the leading edge of LSI complexity in terms of number of transistors.

Don Carley joined RCA in 1957 and initially had responsibility in rf power transistor development. In 1971, he moved into integrated-circuit operations and subsequently held a number of positions in that area, including Manager, CMOS Engineering and Manager, Microprocessor Applications. He is presently involved in automotive IC development.

Contact him at:
Solid State Division
Somerville, N.J.
Ext. 6810

W.D. Henn

Alarm status monitoring and reporting

Globcom solves communication alarm status monitoring by incorporating an easy to maintain microprocessor-controlled system.

Abstract: *Globcom needed to improve and update its alarm status monitoring system. This paper describes how this problem was solved by using a μP-controlled system. It points out the types of tasks and requirements that were necessary for improvement and updating and why a μP was the logical solution to the problem. The features and capabilities of the system that was implemented are described. The simplicity of software and hardware maintenance is revealed.*

This design of a μP-controlled Alarm Status Monitoring System is based on the knowledge gained through the implementation of two prototype systems presently in use at Globcom.

The first system, a basic Alarm Point Scanner without operator interactive software, proved the feasibility and usefulness of such an approach and has been in operation for almost two years in the main operating area in New York.

The second system, designed with built-in sensors for alarm testing of government circuits and including operator interactivity, was installed at the company's operating center at Piscataway, New Jersey, in August of 1977.

The system design to be discussed incorporates concepts from both of these systems and will be installed at both locations in October of this year.

The problem

In order to maintain its leadership as an international record carrier, Globcom must of necessity maintain its many communications circuits at a high level of operating efficiency. These circuits may comprise many individual links, consisting of the communications channel and the necessary data transmission equipment such as time and frequency division multiplexers, high-speed modems, etc. This equipment contains a certain amount of circuitry dedicated to monitoring the operating status of the equipment itself or the quality of the data being received from the distant end. The outputs from these circuits are brought out to front-panel indicators called alarms.

The major portion of the data-transmission equipment is located in an operating area called a "Technical Operating Control Center" or TOCC. These TOCCs are located throughout the Globcom system at such places as San Francisco, Washington, Miami and Piscataway, with the largest presently in New York.

In an early effort to centralize these alarms, the larger TOCCs extended the alarms to an indicator panel where the operating staff could monitor them.

This arrangement was viable in the early days when the volume of traffic was low and the operating staff could take the time necessary to monitor this display panel and log in the equipment problems as they occurred. Unfortunately, the pressure of today's communications world does not allow for all the the time necessary to properly observe the information presented in these tabular displays. To further complicate the situation, the circuit designators associated with each alarm might be in error due to the frequent changes in circuit configurations which are commonplace today as our customers upgrade or reconfigure their own data communications networks. The resulting confusion only served to further reduce the circuit operating efficiency by virtue of increased down time. There had to be a better way.

The solution

The solution to the problem is well within the realm of μP capability since, in an overall view, the task definition seems relatively simple from a functional standpoint.

The requirements for a system that will be able to efficiently handle the reporting of alarm status information both for the present Globcom operating system and the forseeable future are as follows:

1. The system must be capable of monitoring the alarm outputs of the present equipment and report changes of alarm status to one or more locations on a selective basis.

2. The report must include the new status, the identification of the alarm by circuit identifier; the time and date of status change and perhaps the physical location of the equipment being monitored.

3. In order to assist in overall management of the network, certain alarm points can be grouped together to form a major status message which would be routed to a circuit management group to indicate the status of a major communications facility such as a submarine cable or a satellite channel.

4. The system should be able, at some later date, to be interconnected via a large computer network to similar systems in

Reprint RE-25-3-10
Final manuscript received Aug. 16, 1979.

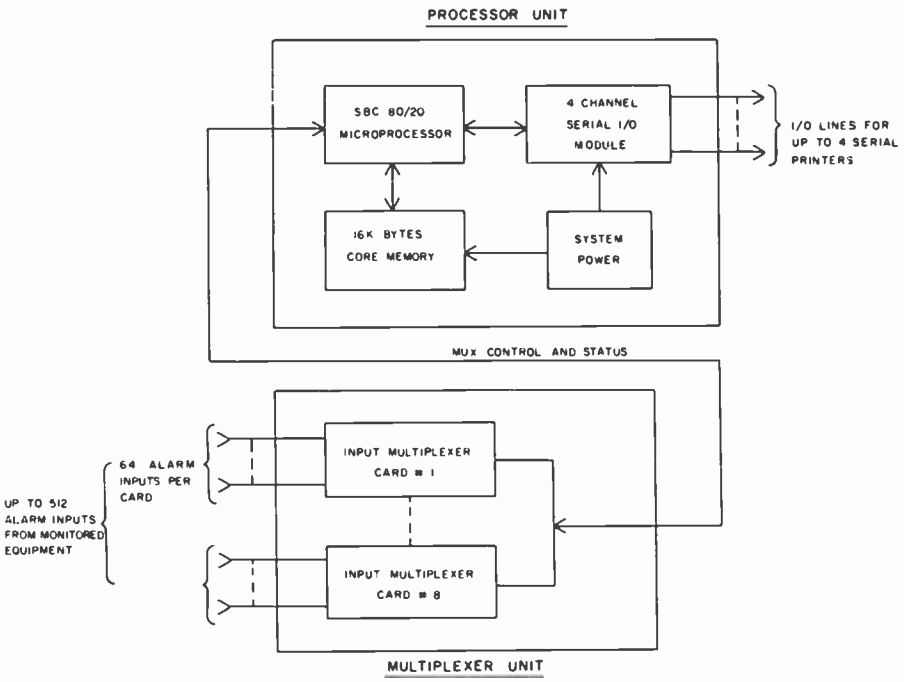


Fig. 1. Alarm points are extended to a common point where they are sampled by a digital multiplexer.

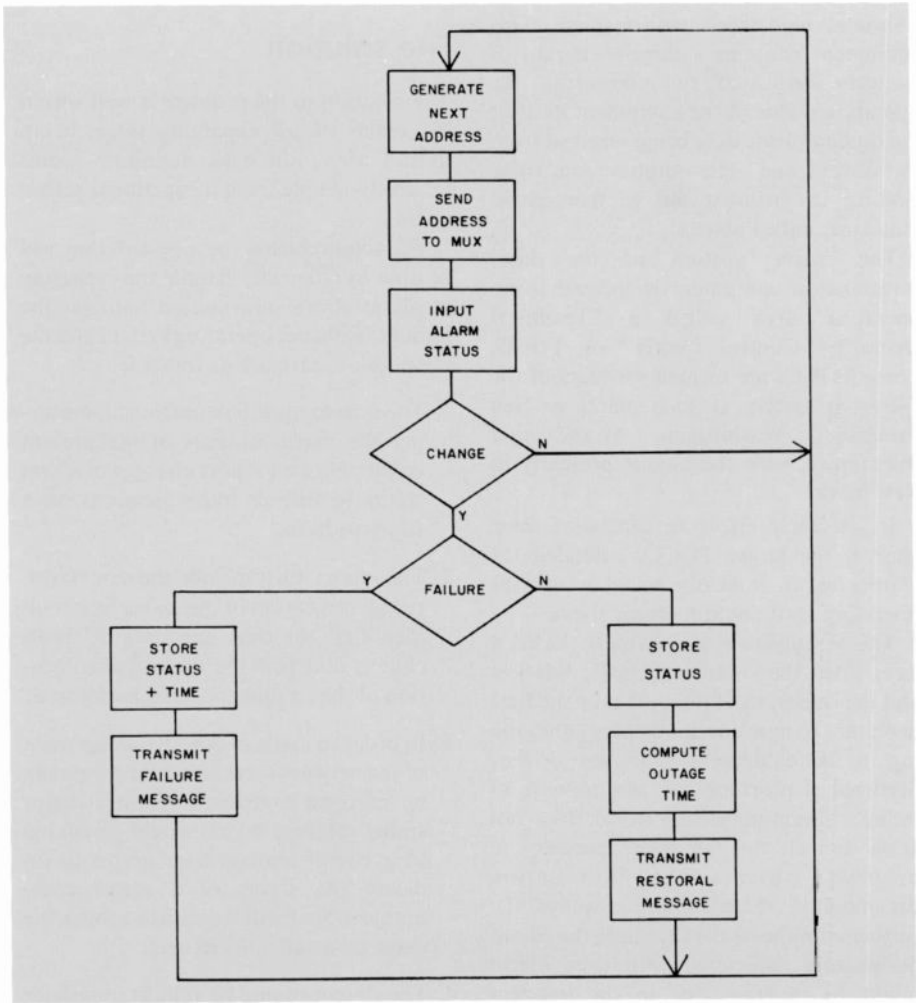


Fig. 2. The system program provides the "intelligence" for determining the status of the alarm points.

other TOCCs. This would enable an overall survey of the operating system status to be made on a real-time basis and correlation of circuit faults between TOCCs to assist in fault-finding operations.

Since the requirements include a high degree of flexibility in configuration in order to maintain compatibility with future alarm monitoring systems, a hard-wired logic approach was not taken. The decision to base the development of the system on μP technology was made as a solution.

System capability and features

This alarm status monitoring and reporting system has the capacity to monitor up to 512 alarm points in expansion increments of 64 points. The expansion is accomplished by simply adding input multiplexer cards, providing the external alarm point wiring and entering the additional alarm point identifiers via the system console.

The file containing the alarm point identifiers is located in core memory in order that no data will be lost in the event of a power outage. In addition to this ID file, a power down software routine saves all the system registers, status tables and ram scratch pad areas in core to insure data integrity upon restoration of power.

As a double safety feature, provision is also made to save the ID file on magnetic tape cassette by means of a service routine available to the operator and a cassette transport which is a part of the control console. By this means, the ID file is saved for later reloading should a system malfunction cause a complete or partial loss of data.

The alarm status reports can be output to any one or combination of four data stations including the control console. Each data station consists of a hard-copy printer and keyboard. Information is 8-bit ASCII-encoded data, serially transmitted at a 300-bit per second rate over a 20 ma loop.

System functional description

The Data Transmission equipment alarm points for the area to be monitored are extended to a common point where they are scanned or sampled in sequence by a digital multiplexer under software control

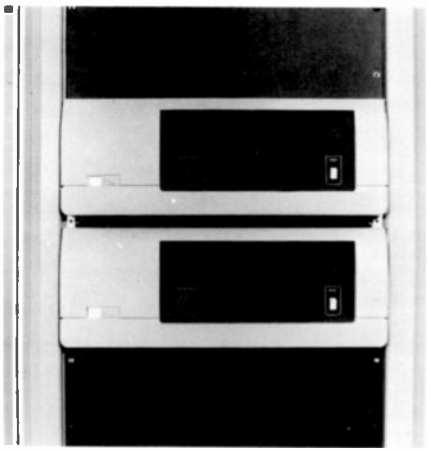


Fig. 3. The system is contained in two rack-mounted chassis.

(Fig. 1). The system program (Fig. 2) provides the "intelligence" necessary to determine if the status of each particular alarm point has changed since the last sample. If the status is the same, the next point is sampled. If the status is now determined to have changed, the direction of change is noted and the appropriate message is output to the proper data terminal or terminals as the case may be. For example, if the alarm point now indicates a failure, when on the previous sample it had displayed a normal state, a programmed wait cycle is entered, during which time the alarm point is sampled on a regular basis. If upon exit from this cycle, the alarm point still indicates a failure, a message consisting of the word 'failure,' the date and time, the alarm point identifier and the equipment location is sent to the appropriate data terminal.

At the time when an alarm point indicates a return to normal, a similar sequence is initiated resulting in a message indicating a 'restoral' for a particular alarm point together with the time and date of restoral and the number of minutes elapsed since the failure occurred.

For those alarm points that are also members of a group for circuit management purposes, a separate record is kept of their status and if all members of that group indicate a failed status, a message so indicating is output to a data terminal in the circuit management area.

In addition to these spontaneous reports, an overview of the system status is available at any time upon request from any of the data terminals. This report lists all of those alarm points presently in a failed state together with the number of minutes that have elapsed since detection of a failed state.

System implementation

Because of the low-volume nature of this application and the need for ease of system expansion and readily available spare parts, the system was implemented on a modular basis using, for the most part, off-the-shelf hardware.

The system uses only four module types: CPU, I/O, Memory and an in-house designed input multiplexer, housed in two rack-mounted chassis (Fig. 3).

These modules are interconnected via a data bus for which many manufacturers are presently supplying cards to support various functions such as serial and parallel data I/O, disc controllers, A/D and D/A interfaces, etc.

Software maintenance

It was previously mentioned that an alarm point, once given an identifier, does not necessarily keep that identifier forever. Due to changing customer requirements, which necessitate the reconfiguring of the data communications links, it becomes necessary to re-identify the alarm points.

The prime factor in determining the usefullness of a system once it has been put into operation is the ease with which the operating personnel can interact with the system so that it can be modified to keep pace with changes in the operating configuration.

For this system, we have chosen to implement this man/machine interface through the use of an English-language interactive operating system. This system leads the operator through the necessary steps to update the alarm point identifier file, to set the real-time logging clock, to regroup alarm points and to determine the destination of the status messages. Even the uninitiated can call up a synopsis of the entire system operating procedure by entering the letters "H E L P" from the keyboard of the control console.

Hardware maintenance

The hardware may be maintained at least down to the card level in the field by means of low-cost, commercially available automatic test equipment. Diagnostic software for this equipment is readily written at the engineering level allowing many aspects of the system to be tested via a single interconnection through the CPU socket of the system under test.

Summary

In retrospect, this problem lent itself to solution by a μP-controlled system because a large amount of flexibility was required in system operation by virtue of necessary changes in the alarm point ID file.

A spin-off benefit of the system data bus architecture is the avoidance of system obsolescence. The system hardware can readily be used as part of a future expansion program or supplemented with other available module types for an entirely new configuration.

Equipment alarm systems that were previously impractical to design because they were either too complicated to be cost-effective, or so rudimentary in their reporting capability as to be almost worthless are now possible for the design engineer to implement due to the availability of μP technology.

William D. Henn, is a Principal Member of the Engineering Staff, Advanced Development & Design Group, RCA Global Communications, Inc., New York, N.Y. From 1966 to 1968, he was at the RCA Advanced Communications Laboratory in New York City, when he joined the Advanced Design Group at Globcom and was assigned to tasks involving digital tape buffer design and alternate voice/data controllers. In 1972, he was given the television camera design responsiblity for the Videovoice communications system and is presently working on applying microprocessor technology to data communications.

Contact him at:
Global Communications
New York, N.Y.
Ext. 3568

A. Abramovich|A.K. Patel

Microcomputer system design for remote process control

Microcomputer-controlled energy management provides efficient use of energy-related resources, reduced downtime for maintenance and unlimited flexibility in terms of functional implementation, interface requirements and a simpler design.

Abstract: *A microcomputer-based Energy Management System has been incorporated at the RCA Service Company facility in Cherry Hill. The operating requirements for such a system and the hardware that is used for its implementation are described. A central microcomputer is the heart of the system and is based on an Intel SBC 80/20 single-board computer. The remote processor, which communicates climate-control data to the central microcomputer and implements its commands, uses an Intel 8748, single-chip microcontroller. Work is being continued to replicate the remote processors for computer control of the remaining buildings in the Service Company facility.*

A microcomputer network for energy management now controls building 203 at the RCA Service Company facility in Cherry Hill, New Jersey. The Service Company's facility consists of a seven-building complex, each one having its own heating, ventilation, and air conditioning (HVAC) equipment. Prior to automation, each building's environment was regulated by a pneumatic control system. This system provided zonal-temperature regulation based solely on feedback from local thermostats. Other equipment, such as air supply and exhaust fans, as well as dampers, are controlled by a number of electromechanical timers as well as the facilities operators.

Reprint RE-25-3-11
Final manuscript received Sept. 25, 1979.

Automation of such an HVAC system would supply RCA with useful and important information as to the benefits which can be realized. These benefits could manifest themselves in terms of improved operating efficiency and reduced operating costs. In addition, a vehicle will be provided to conduct long-term studies in the areas of energy management, algorithm optimization and hardware development.

The system to accomplish these goals is configured as a network of microcomputer-based controllers which interact with a high-capability central microcomputer. In this article, a total system overview will be given, followed by a description of the hardware implementation.

System description

The computer energy management system (CEMS) (Fig. 1) is configured with a central processor located in the boiler room in Bldg. 205. The operator communicates with the central processor via a keyboard and printer. Remote slave processors are located in each building's HVAC equipment room, known as the Penthouse. The remote processor interfaces with the HVAC equipment solenoids

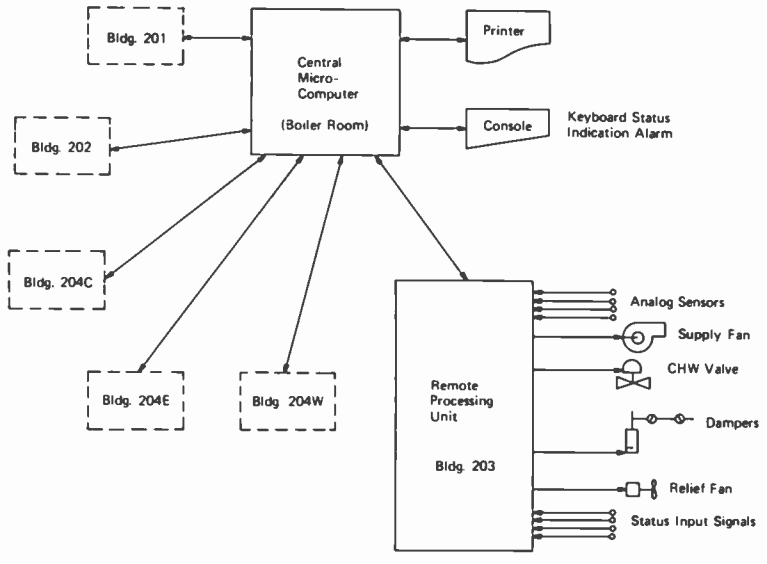


Fig. 1. The Computer Energy Management System is configured with a central processor.

and relays. The equipment used in a typical HVAC system, shown in Fig. 2, consists of a number of high-powered fans, and various dampers to control air flow through the building, bring in fresh outside air, and vent stale building air. In addition, devices, such as chilled-water valves, control cooling levels in the summer, and reheat coils operate in the winter. Solid-state temperature and humidity sensors are used to characterize the building's environment.

The CEMS system is intended to operate in conjunction with, and control, the existing pneumatic system rather than totally replace it. This is done to minimize the impact of system installation, reduce installation costs and provide a natural fail-safe system.

The CEMS is designed to implement the following operating requirements:

- Ability to control remotely located equipment automatically or as a result of manual operator keyboard entries.
- Implement control algorithms for efficient management of energy resources.
- Implement system scheduling such as:
  System power-up and power-down times
  Stagger the turn-on sequence of heavy duty equipment to even out electrical loading.
- Execute operator commands via a keyboard to display system status, change system operating parameters and override decisions made by the computer.
- Monitor and report alarm conditions.
- Act as a security system to detect fire, smoke, and breach conditions, and to remotely control door locks.
- Provide fail-safe operation. In case of a failure, revert back to use of the existing pneumatic system.

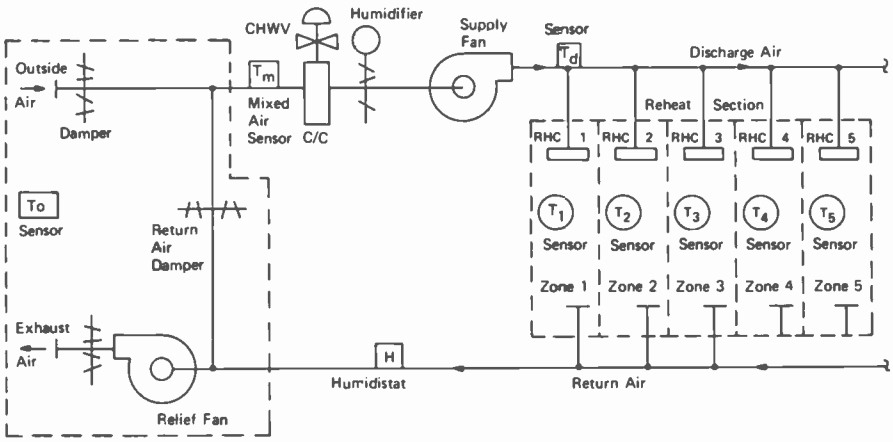


Fig. 2. In a typical HVAC system the equipment consists of high-powered fans, dampers and sensors.

mands from a dedicated-function keyboard. In addition, the CMC executes the energy control algorithms, and system scheduling functions, and it generates status logs and environmental reports.

The hardware complement, which comprises the central microcomputer (Fig. 4), is based on an Intel SBC 80/20 single-board computer. This board contains an 8080A Microprocessor, 4-k bytes of RAM, sockets for up to 8-k bytes of ROM, an eight-level interrupt controller, RS-232 channel, timers, and, of course, 48 parallel I/O lines. A ROM expander board expands the system's program capacity by an additional 32-k bytes. The SBC 80/20s RS-232 channel is dedicated to the printer. Communication with the remote processors is generated on an SBC 534 four-channel communication board. System channel capacity is increased by adding additional SBC 534 boards. A special interface board holds the keyboard encoder, the real-time clock, power-fail detection circuits and battery charger. The communication links' differential line drivers and receivers are also located on this board. The use of differential signals for the communication link permits essentially noise-free operation with dedicated lines up to 2000 feet long.

The NMOS RAM supplied on the SBC 80/20 board was replaced with CMOS RAM. Data retention, under a power failure situation, is maintained for periods of up to four hours. The real-time clock, the Intel 8741 single-chip microcomputer, also continues its operation under battery back-up. It maintains correct time of day and date information during power failure periods.

The dedicated-function keyboard is designed for ease of use and clarity. Its 48 keys are grouped as three keypads. Keyboard layout is shown in Fig. 5. As seen from the layout, each keypad is assigned related sets of functions or commands. Key entry and selection is from left to right. More than one key may be selected from each keypad. A typical entry sequence would be as follows:

SET, Bldg. 203, SUPPLY FAN, ON, ENTER

The terminology used on the individual keys is selected for operator familiarity. The keyboard console also contains a numeric display for use as a digital clock. A number of status indicators reveal infor-

Master processor architecture

The central microcomputer (CMC, see Fig. 3) can be viewed as the heart of the system. Its functions include communication with the remote processor units which gather status information from their respective buildings, and also dissemination of appropriate control information to the remote processors. This is accomplished through a dedicated multichannel serial communication link. Other functions of the CMC include interfacing with the operator via a printer, and receiving com-

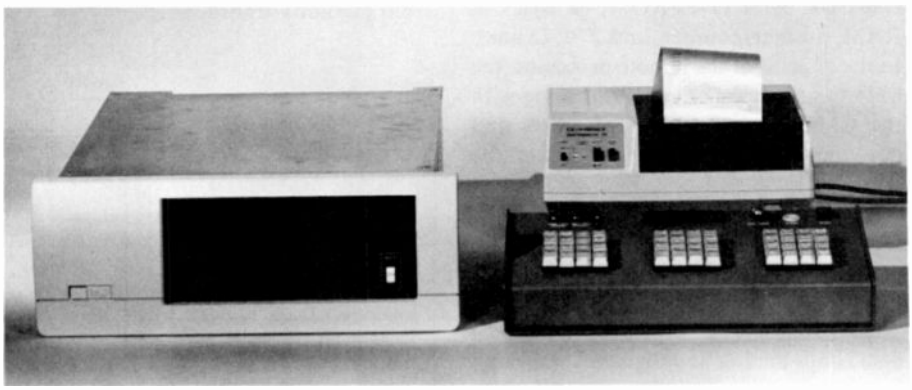


Fig. 3. The CMC is comprised of a processing unit, an operator's console and a printer.

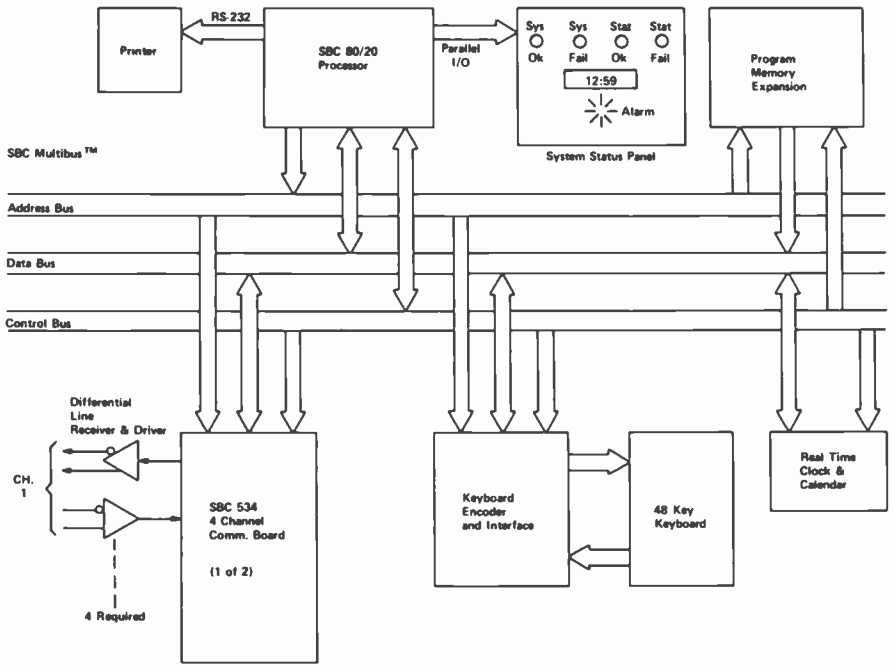


Fig. 4. Hardware for the central microcomputer is based on an Intel SBC 80/20 single-board computer.

mation such as: system RUN, system FAIL, system OK (no alarms), and alarm condition. An audible alarm is included to attract attention, and an alarm reset push button. The printer, selected for this application, is an electrostatic printer which is software programmable to print in 20-, 40-, or 80-column formats. The bold 20-column format is used for alarm and special messages, the smaller 40-column format is used for normal printing. Software for the CMC was written entirely in PLM and occupies 34-k bytes of memory.

Remote processor architecture

The Intel 8748, single-chip microcontroller, is ideally suited for use in the remote processor. The 8748 contains on-chip, 1-k bytes of EPROM, 64 bytes of RAM, a timer/counter, and 27 I/O lines. In the context of the remote processor, the 8748 has the tasks of communicating with the CMC, and checking for errors and decoding commands received from the CMC.

Commands received from the CMC fall into two categories. The first type is a command to the remote processor to initiate its data acquisition process and transmit to the CMC all the corresponding data, both analog and digital. The second command type includes control data generated by the CMC, both analog and digital, which the 8748 transfers to the appropriate outputs.

Figure 6, the remote processor block diagram, shows that the 8748 controls a 24-channel analog data acquisition subsystem. Sixteen of these channels are dedicated to inputs from temperature sensors. These sensors are two-wire constant-current sources whose current varies proportionally and linearly with temperature. The device responds to temperature changes at a rate of 1 microamp per degree K. The advantage of using constant current devices is that they may be located many thousands of feet away from the processor, as they are independent of wiring resistance. These devices offer high noise immunity because voltage variations across the device do not change the current through it. The remaining eight analog inputs are of general purpose nature and are used for inputs from humidity transducers.

The 8748 also reads sixteen digital status inputs, and outputs to eight digital control lines. All status inputs and control outputs are isolated from the outside world through optically coupled input and output modules. These modules translate ac or dc signals to "clean" logic levels, and conversly allow logic inputs to switch ac or dc loads. In Fig. 7, the internal layout of the remote processor, the logic boards, as well as the interface modules are shown. The output buffer is gated with the output of a "watch-dog" timer whose function is to provide fail-safe operation. Each time a successful communication link transaction is completed, regardless of type, the 8748 retriggers the "watch-dog" timer. This enables the buffer outputs for normal operation. If communication errors are detected by either the CMC or the 8748 remote processor, the communication process is interrupted, or if the 8748 processor ceases to operate properly, the "watch-dog" timer will not be refreshed. After a 30-second time-out period the timer disables the output buffer, and the system reverts to its fail-safe mode of operation. The fail-safe mode is the pre-automation pneumatic control system. A communication-interface chip and differential-line drivers and receivers interface the 8748 to the communication link. Software for the remote processors was written in assembly language and is approximately 750 bytes long.

Communication protocol

The function of the communication protocol is twofold. The first, and most obvious, is to carry out the exchange of information between the CMC and a remote processor. The second, and just as important, is to be an integral part of the fail-safe mechanism. Transmission of data between the CMC and a remote processor is bi-directional, but only the CMC initiates a communication exchange. The

Keyboard = 1

Display	Set	Cancel	Enter
All BLDS	BLD 201	BLD 202	BLD 203
Schedule	BLD 204E	BLD 204W	BLD 204C
Power Up	Power Down	Log Time	ALARM History

Keyboard = 2

TO BLD In Use	TM Computer Control	TO Fan Messages	TR Temp Corr
1 Season	2 Suppy Fan	3 MIN OAD	4 DOOR
5 Set Point	6 Error Signal	7 Cycle Time	8 Delay Time
9 Temp/ Humidity	10 Relief Fan	11 Smoke Detector	12 Override

Keyboard = 3

Oct 7	Nov 8	Dec 9	Open On Winter
Jul 4 Sun	Aug 5	Sep 6	Closed Off Summer
Apr 1 Thu	May 2 Fri	Jun 3 Sat	• Time Of Day
Jan Uptime Mon	Feb 0 Tue	Mar Day Wed	- Date

Fig. 5. The dedicated-function keyboard is designed for ease of use and clarity.

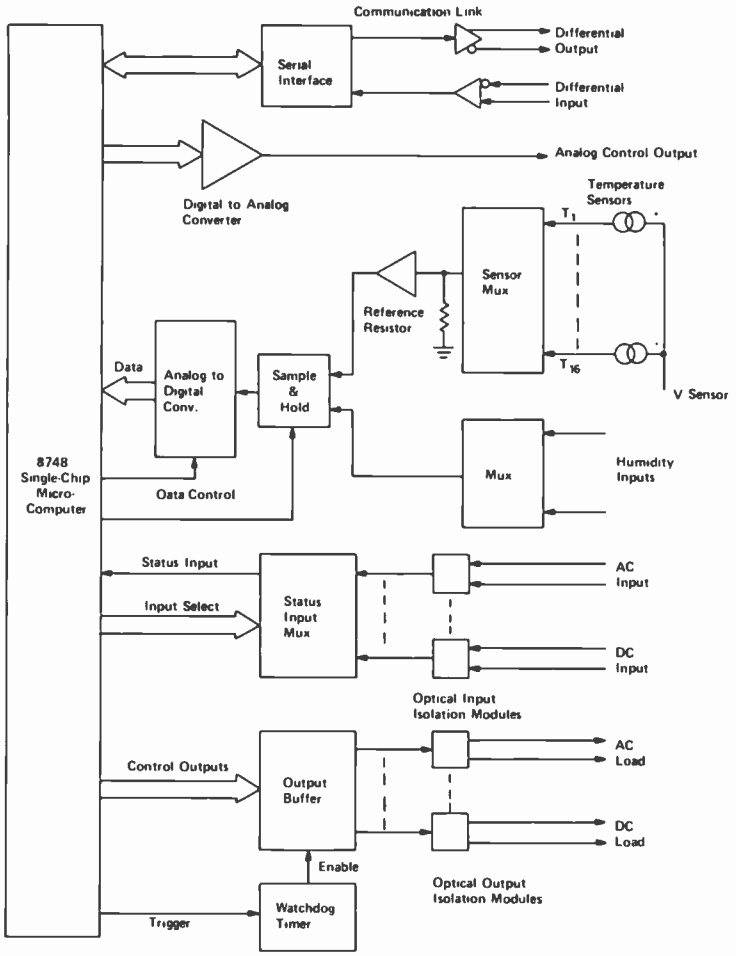


Fig. 6. The Intel 8748 controls a 24-channel analog data acquisition subsystem and digital I/O.

remote processor's function is to respond, but not initiate communication. A fixed length preamble of seven bytes initiates and defines a certain procedure, and is formatted in the following manner:

1st byte	total message byte count
2nd byte	complement of 1st byte
3rd byte	remote unit I.D. number
4th byte	command code
5th byte	error code
6th byte	checksum (exclusive-or of all other bytes in message $+55_{16}$)
7th byte	data byte count
8th byte	through n^{th} byte are data, if included

A number of tests are performed by both processors to verify data integrity; some are obvious and some are implied. The tests performed include: verfication that the received byte count equals the stated byte count, checking unit ID number, checking for valid commands and error codes, verification of checksum value, and checking that the message byte count, less the data byte count equals seven. A time-out function detects when either the last byte was received, or when transmission has been prematurely interrupted. A check of the message byte count determines which situation has occurred. This ensures that neither processor waits for data that will never arrive.

Every transmission is always acknowledged by the receiving unit by the return of data, or an acknowledge (ACK) to indicate that a control command was accepted, or a no acknowledge (NACK) to indicate that an error was detected in the message just received. Only data received during an error-free exchange is used; otherwise, it is discarded. In the remote processor, every error-free communication exchange is used to re-trigger the unit's "watch-dog" timer.

In the event that a communication link failure occurs, the CMC notifies the operator by printing an error message. If the CMC fails to properly communicate with a remote unit for five consecutive tries, it considers the remote processor as being out of order. Nevertheless, the CMC attempts to re-establish communication with the remote processor after a temporary problem, such as a power failure.

Performance

At this time, this prototype computerized energy management system has been installed and is operating at the Cherry Hill location of the RCA Service Co. Not enough time has passed to accurately assess its contributions to energy savings. But there is no doubt that substantial savings will be attained. Every minute saved in operation of heavy electrical loads, such as air supply fans and water-chillers, is directly translatable to dollar savings. The use of fresh outside air instead of "processed" air is optimized by the CEMS system, with additional savings of our valuable energy resources.

More immediately obvious is the improved control capability and the operator's awareness of conditions in the building under control. As such, the CEMS has proven to be invaluable.

Work is being continued at the present time to replicate the remote processors so that computer control may be added to remaining buildings as soon as possible.

Conclusion

This system, although specifically designed to operate as an energy management system, can be viewed as a general-purpose microcomputer network which is well suited for remote process monitoring and control. It has the ability to reliably control activities and gather information from locations many thousands of feet away. The use of modems could extend the range substantially. The dedicated keyboard approach simplifies the requirements placed on the operator, and this is an important contributing factor to the overall system reliability. The remote processing units are designed for unattended operation and their operational reliability is enhanced by the error-checking mechanisms built into the communication protocol.

The use of such a system for energy management will realize the efficient use of energy-related resources through automation and algorithms programmed into the central microcomputer. Other benefits are realized in the areas of reduced down-time for maintenance, because marginal operating conditions can be detected and reported before they turn into problems.

The use of microcomputers in such a

Building control with brains

All of us are currently involved in a nationwide program to conserve energy. Logical steps such as reducing the temperature and shutting off lights have been implemented to take out the fat. Another step is to improve efficiency by adding insulation or improving maintenance. A third step is to optimize operations through improved controls.

The computer has long been known for its ability to collect data, compile it and perform certain routine and preset functions. A computerized control system utilizing a microprocessor has been applied to optimize energy utilization and building operations at Cherry Hill.

Optimization can occur simply by revising current operating schedules to reflect the actual needs of the operation, i.e., ten hours per day vs. 24 hours. A second improvement could occur with the use of time clocks to improve operation. These steps were all implemented at Cherry Hill with the HVAC system, resulting in significant savings of energy. Seeing the savings in oil and electricity, the question arose: "How can we fine tune the system"? The method chosen was computerized control and monitoring system that provides for centralized reporting, analysis and action.

In order to intelligently control HVAC operations, a certain amount of information is needed, such as: outside conditions, zone temperatures, relative humidity, time of day and occupancy schedule.

In addition, certain key controls have to be under the control of the operator through the computer, such as: zone temperature set points, damper position and chilled water temperature. With this information, the computer can logically control and shed steam, chilled water and electrical consumption.

In addition to optimizing operations, the system helps eliminate waste resulting from equipment over-design. What is more, the equipment is set up to warn of fire, danger to life and breach of security. The potential for other applications such as boiler control and chiller operation are many.

In an attempt not to fall victim to the "over-kill" syndrome of the purchasing equipment with the ability to perform more tasks, obtaining more information and monitoring more conditions than is necessary or that can be efficiently used, the Energy Systems Analysis Group at the Laboratories was contacted for assistance. The "off the shelf components" microprocessor system with unique software methodologies has resulted in a flexible, smaller, cost-effective building operation aid than is currently commercially available.

While the microcomputer remote process control system has only been recently installed, early indications are that the resulting savings will be substantial, with the ever-increasing energy rates.

Bob Weaver, Service Company
Cherry Hill, N.J., Ext. 5721

Patel, standing, and Abramovich check some CPU results.

Abe Abramovich is a Member of the Technical Staff at the RCA Laboratories in Princeton. He joined RCA in 1976 and is currently a member of the Microsystems Research Group. Abe was a member of the team responsible for the development of an automated board testing instrument which currently is used by the Consumer Electronics Division. For that work, he was awarded the 1978 RCA Laboratories Outstanding Achievement Award. He was also responsible for research work in the application of microcomputers to TV remote control systems. His most recent assignment was as project coordinator for the Computer Energy Management System.

Abe has published a number of articles on the applications of Integer Arithmetic and was the recipient of the 1978 Outstanding Paper Award from the Industrial and Control Instrumentation Society of the IEEE.

Contact him at:
RCA Laboratories
Princeton, N.J.
Ext. 2545

Anil Patel is a Member of Technical Staff at RCA Laboratories in Princeton. He joined Microsystems Research in 1978 and has been involved with microprocessors and their applications to energy management system. Currently, he is involved in microprocessor applications for new consumer products.

Contact him at:
RCA Laboratories
Princeton, N.J.
Ext. 2812

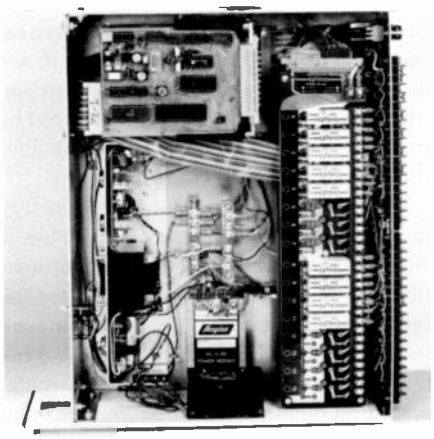


Fig. 7. The remote processor with logic and interface elements are shown.

system permits unlimited flexibility in terms of functional implementation, interface requirements and a simpler design. This directly leads to reduced system cost and improved reliability.

Acknowledgment

The Energy Management System, described in this report, is the culmination of a cooperative effort between theRCA Service Co. and the RCA Laboratories. At the Service Co., Mr. Bob Weaver, Director of Facilities, appreciated the potential and benefits such as a system would provide, and eagerly supported the project.

Here at the Labs, three groups supplied the expertise required to implement this project. First, Dr. B. Hershenov's Energy System Analysis Group, in Dr. B.F. Williams Laboratory, held the overall responsibility for the project. In that group, Dr. K.S. Vanguri supplied the knowhow required for the implementation of the HVAC interface and the energy related algorithms. Dr. P.M. Russo's Microsystems Research Group and Dr. R.H. Roth's Systems Architecture Research Group, both in A.H. Teger's Laboratory, supplied the technical expertise. Mr. R.J. Poulo, in the SAR group, was responsible for the software development effort.

R. Poulo

Software design methodologies applied to remote process control

The high-level tools of data-flow diagrams and top-down design and implementation provide crucial development techniques for microprocessor systems.

Abstract: *Microprocessors are being used today in applications so complex that they run an entire software system. The choice of software tools, i.e., data flow diagrams vs. flowcharts, and the steps to be taken in building a system are discussed. Comprehensive data flow diagrams, how they evolved and their function in the implementation of the Computer Energy Management System are presented. The design of the modules to implement the system organization, using a top-down design implementation, is described. The result of applying these software techniques has been a coherent, highly modular and easily modifiable software system.*

In the past, the development of software for microprocessors was years behind the development of software for large computers. Limited memory size and inadequate software tools had always dictated that system designers program in assembly language and use debugging tools that were scarcely worth the name. It seemed to follow, as a corollary, that inadequate design tools would also be used. However, with microprocessor hardware following the growth of large computers (at a faster rate) it is time to apply more sophisticated tools to the design and implementation of software for microprocessors. This article will attempt to provide some motivation for the software tools actually used in designing the Computer Energy Management System (CEMS; see "Microprocessor System Design for Remote Process Control" in this issue).

Reprint RE-25-3-12
Final manuscript received Sept. 12, 1979.

Choice of software tools

As in CEMS, microprocessors are now being used in applications so complex that they run not a program but a software system. The emphasis here is on the word "system" where the components of a system are themselves programs or sets of programs. System design, therefore, consists of defining the function of each component and the interactions between components. The design of the internal mechanism of each component program is a separate problem which should not be tackled until the system design is complete.

In designing a system, the designer does not, in general, know much about the time sequencing of the system components. In some systems, the time sequence may not be unique over several iterations or, worse, may not even be definable. A simple example of this, though not the only example, is any system which is known in advance to have some parallel processing. The tools that will be useful to the system designer will be those that do not make any explicit mention of a sequence of operations. While tools that do explicitly mention the sequence of operations could be used, by the very process of using them, the designer makes decisions regarding the time sequence. Not only should these decisions not be made at this stage but the wrong decisions might be made. The right decisions may not be clear until much later.

Unfortunately, the most widely used design tool is the flowchart. Flowcharts are ideal tools for separating out and graphically displaying just the flow of control, or time sequence of a set of operations, in a system. But this is exactly what is not desired in systems design. A mathematician might say that flowcharts are orthogonal to the kind of tools that are most useful. While a good argument can also be made against flowcharts at the level of program design, it should be clear that one should never use a flowchart in systems design.

If flowcharts are bad then what is good? To answer that, one must look at the result of a system design. This result is a plan describing the functions of the major components and the data that are passed between them. An important point is that the data are really the *raison d'être* of the system. In particular, given the input data the purpose of a system is to produce, by any means necessary, the output data. With this in mind, another way to describe the result of a system design is that it is a description of the flow of data through the system, the transformations suffered by the data and a set of programs to accomplish the transformations.

The first software tool is the data flow diagram. This is a graphical illustration of the flow of data in a system and the transformations applied to the data. The energy management system will be presented, first, as a sequence of data flow diagrams showing how the system evolved. A description of the programs that will achieve the transformations will be derived by another software tool. When the design

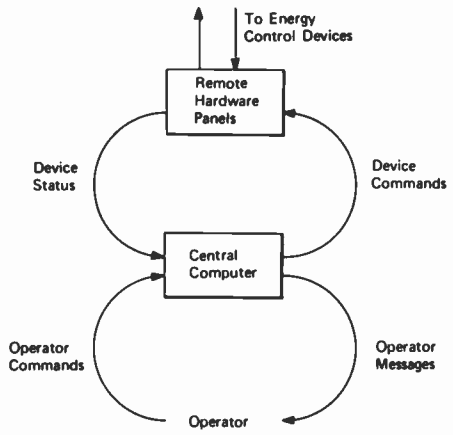


Fig. 1. The central computer is the heart of the Computer Energy Management System environment.

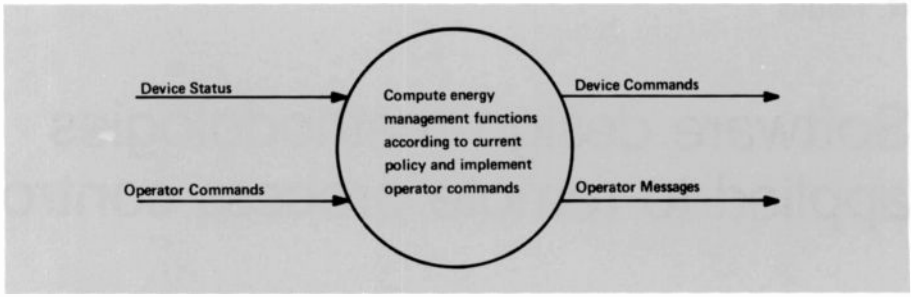


Fig. 2. The function of the central computer is to take, as input, the building-environment status and operator commands, and to output commands to remote hardware and messages to the operators.

is complete, one can plan the implementation of the system.

We now have these three distinct steps in building a system: deriving the data flow diagrams and supporting information, deriving the set of programs to implement the data transformations, and implementing these programs to produce the system. The word "design" will be given a restricted meaning. The three steps will be respectively called analysis, design and implementation. By segregating the construction of the system in this way, each major step involves the use of one major software tool.

Evolution of the CEMS software

A high-level block diagram of CEMS is shown in Fig. 1. This figure displays the system environment as seen by the central computer. Abstracting from the figure we get the simple data flow diagram shown in Fig. 2. This data flow diagram shows that the function of the central computer in CEMS is to take, as input, the actual status of the building environment and the operator commands and apply appropriate data transformations to produce as output the commands to the remote hardware panels and status messages to the operator.

The process of analysis now consists of refining the diagram of Fig. 2 in stages to achieve the same overall data transformation by a set of smaller, interconnected transformations. Such smaller transformations are more specific, better defined and closer to the ultimate implementation than large transformations. Figure 3 shows the next level of refinement for CEMS. Each circle, or bubble, represents some data transformation. The flows of data into and out of a bubble are shown by the arrows connecting the bubbles. One must remember that there is no implication whatever regarding a time ordering of the data transformations nor of how any transformation is to be realized.

Only the gross detail of the central computer is shown in Fig. 3. The functional decomposition is roughly into transformations accomplishing the operations of input, computation and output. The first bubble, in Fig. 3, gets the actual status from the remote hardware panel and compares it to the commands that were sent to the panel. If there is a discrepancy, an operator message must be generated describing the error. Thus, the inputs to the bubble are the actual status and the device commands while the outputs are the actual status and operator messages. The second bubble computes the new set of logical commands according to the energy management policy and implements commands entered by the operator. The last bubble converts the logical commands to physical device commands and sends them to the remote hardware panels. At this point there is little detail visible, yet, some insight into the proper design of the system has been gained. Input and output and their associated functions have been separated from the remainder of the system.

The central bubble of Fig. 3 represents a complex data transformation. Since the energy management policy and the operator commands will operate independently of each other, the next refinement is to separate this bubble into two bubbles. At first glance one might conclude that the two new bubbles are themselves independent, but a closer examination reveals that, in their output, both have commands to the remote energy control devices. This is clear for the energy management bubble. The operator, who can assume direct control over the system, can also issue device commands. With no restriction on the operator, it is inevitable that these two sets of commands will often be contradictory. The solution is a third new bubble whose function is to arbitrate between the two sets of device commands and produce the set of commands that will really be sent to the remote panels.

Figure 4 shows this refinement. It should be clear that a heretofore unknown function of the system, arbitration of two sets of inconsistent commands, has been revealed by a simple refinement of a data flow diagram. In fact, the refinement was so simple and obvious that one might have been tempted to not bother with it, thus, ignoring an important aspect of the system until too late. The data flows from the actual status to the compute-new-energy-system-commands and implement-

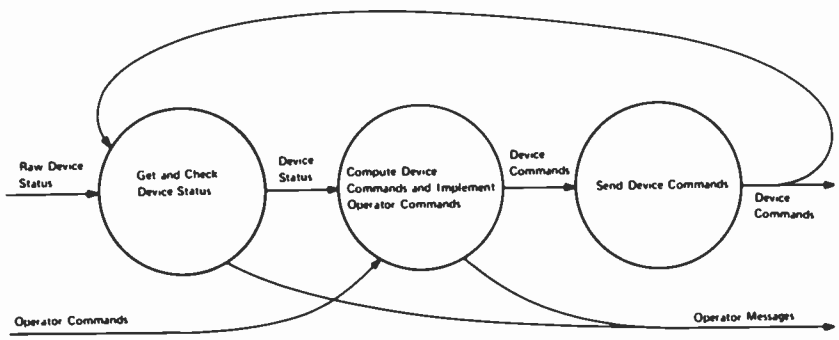


Fig. 3. The system is refined into transformations for input, processing and output. The need for send-device commands is not obvious. During early system development, its independent existence was deemed useful and later work confirmed this.

operator-commands bubbles are included for use in the energy algorithms and operator messages, respectively.

It is instructive to pause and consider what the system might have looked like at this stage had flowcharts been used. First of all, it is possible, and perhaps probable, that the energy management functions would have been considered of luminary importance from the beginning. The function of handling the operator commands that override the energy management algorithms and directly control the environment would be relegated to being an ancillary function. Verifying the actual status would certainly be considered a detail to be added later and would probably end up as the first part of the program that computed the new set of device commands. Yet, in the actual system design, there is still no detail as to the working of the energy management algorithms. Instead, the high-level function performed by these algorithms has been integrated with the other high-level functions of the system which is the essence of top-down design as applied to whole systems.

The next step is to decompose the compute-new-energy-system commands bubble. Figure 5 shows the result. One of the new bubbles represents the execution of all algorithms that are to be evaluated on a continuous basis, i.e., as often as possible. The other represents the execution of algorithms that are controlled by a schedule. The schedule itself is a global data repository and is also shown in Fig. 5. The compute-scheduled-energy-functions bubble reads the schedule to decide when to execute the various algorithms. Since some algorithms control when other algorithms are executed, they make changes in the schedule resulting in the data flow to the schedule. In addition, the operator can both change and examine the schedule so there are data flows between the schedule and implement-operator-commands bubble.

The last major refinement of the data flow diagrams is shown in Fig. 6. Because the fans that supply air to the buildings require special treatment, it is useful to separate them from the other device commands. The restriction on the fans is that no two fans can be turned on within two minutes of each other. Since the algorithms generate fan turn on commands without regard to this restriction, a fan command queue has been introduced. The choose-energy-system-commands bubble takes commands off the queue when enough time has passed.

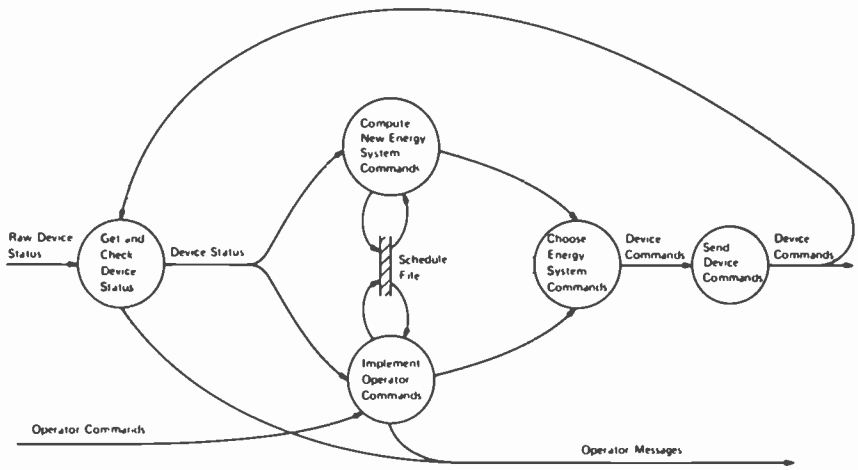


Fig. 4. Arbitration of two sets of inconsistent commands was revealed by a refinement of the data flow diagram which shows how the energy management functions fit in with the other functions of the central computer.

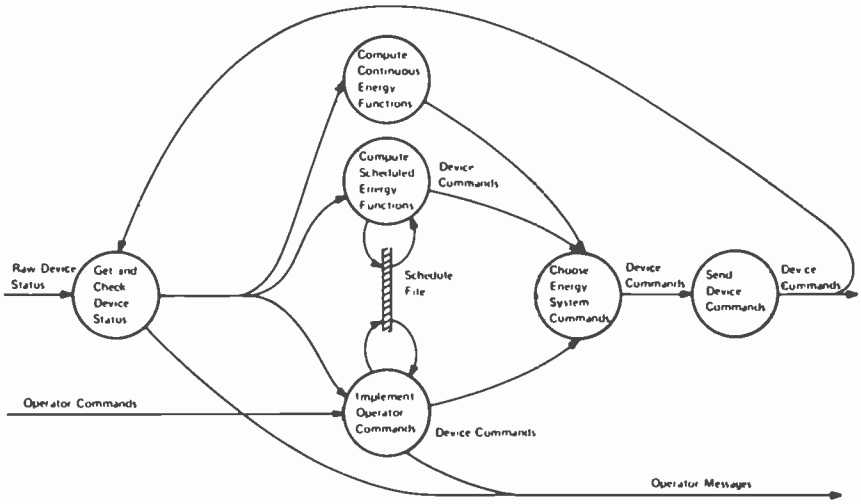


Fig. 5. The energy management functions are separated according to how function execution is determined.

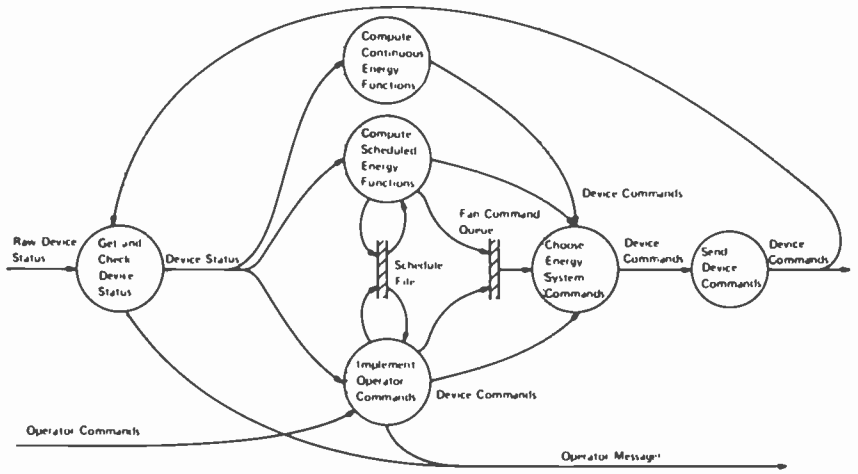


Fig. 6. The final refinement of the CEMS data flow diagram separates the commands to turn on the fans for special treatment.

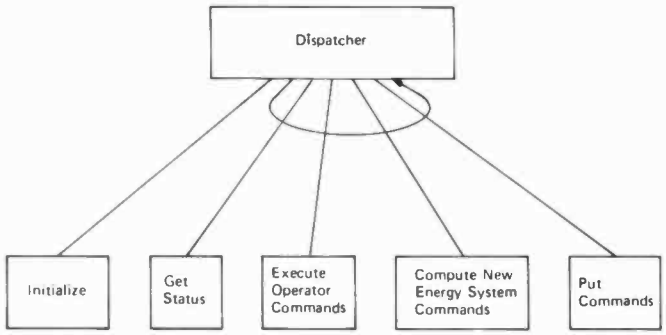


Fig. 7. The top-level structure chart shows how the dispatcher successively calls routines to implement high-level functions of the system.

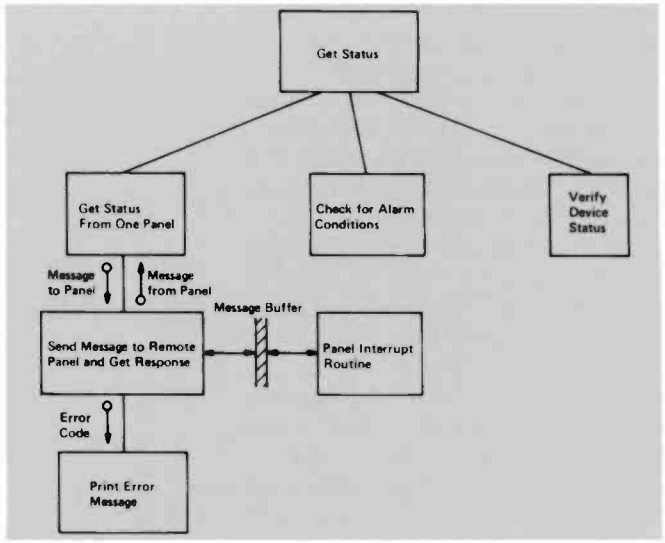


Fig. 8. Subroutines of the get-status routine are used to get and check the status of the various devices.

Design and implementation

The system organization, as shown in Fig. 6, completes the analysis phase and brings us to the design phase. The task is to design modules to implement the system organization. The software tool used is the structure chart, which shows each routine and the calling relations between routines. Each box of a structure chart represents one routine. Lines between boxes show what routines call other routines. Information passed in routine arguments is shown on a chart next to the connecting line. Again, the design is top-down. The first step is shown in Fig. 7, where a dispatcher successively calls routines to implement the high-level functions of the system. This dispatcher is the main routine of the system and calls its subroutines repeatedly to continuously execute all of the system operations. How each subroutine executes its particular operation is not the concern of the dispatcher.

Each second-level routine is now broken up in the same way. For example, the "get-status" routine of Fig. 7 is shown with its subroutines in Fig. 8. The operation of obtaining status partially consists of getting the status from each remote hardware panel. Routine "get-status-from-one-panel" gets the status from one hardware panel. It will be called several times for each get-status operation but its workings are of no concern to the get-status routine. Note the routine "verify-device-status" which performs the check of the device status against the expected status. This is the routine that implements the function of comparing the raw-device-status data flow with the device-commands data flow carried out by the get-and-check-device-status bubble in Fig. 6.

Real-time systems, such as CEMS, have special problems which are also amenable to treatment with the software tools. For example, the problem of communicating with the processors on the remote hardware panels requires that the design phase include how the interrupt routines fit in with the routines of the main process. Figures 8 and 9 show this interrupt routine. Note that, as an interrupt routine, it is never explicitly called by another routine. This is shown in the figure by the absence of a connecting line to a higher level routine. In the design phase, the details of processor communication were planned and completely understood before any implementation had begun, including the details of how the main routine and interrupt routine shared and passed information. Similar treatment was given to the interrupt routines for the keyboard, printer and external clock.

The process of refinement, by specifying

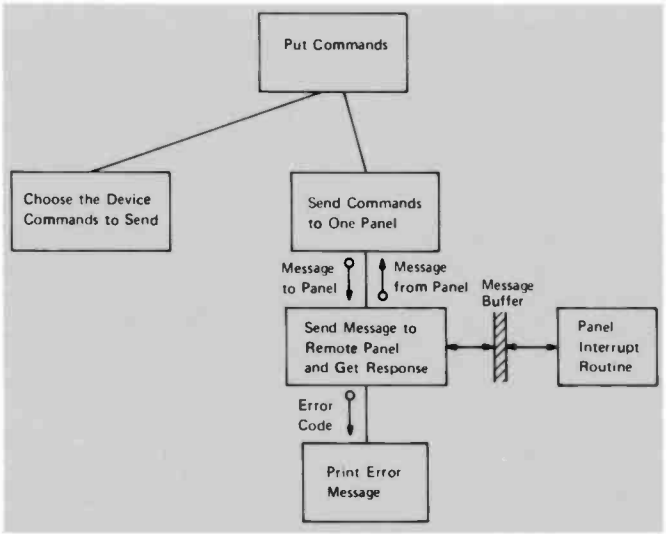


Fig. 9. These routines are used to select and send commands to the remote panels.

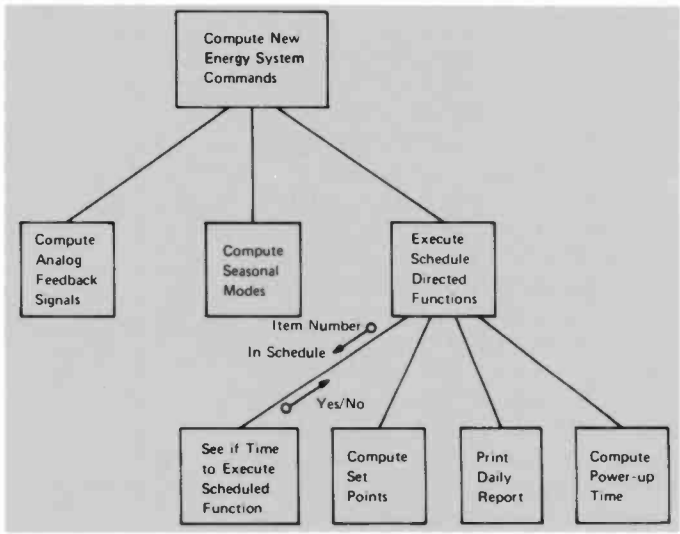


Fig. 10. These routines implement the energy-management policy.

routines at lower levels, continues until the function of all routines can be specified by roughly a one-page English description. Given a good functional decomposition, in a data flow diagram, the specification of the routines will proceed quickly. In designing CEMS, about six weeks of trial and error were dedicated to analysis culminating in Fig. 6, while only one additional week yielded all the structure charts, the major ones appearing as Figs. 7-10.

A top-down implementation was used in writing the code of the routines. The routines, shown in Fig. 7, were written and tested first, to be followed by the routines in Fig. 8, only when they seemed to work correctly. Top-down implementation often seems to be backwards at first examination, the usual approach being to design top-down but implement bottom-up. Yet, the arguments in favor of applying the top-down methodology apply to implementation methods as well as to analysis and design methods. The ramifications of a change in a high-level routine will propagate deeply into the lower regions of the structure charts while the converse is false.

All routines, including the interrupt routines, were written in PL/M, a PL/1-like language designed for microprocessors. There is not a single line of assembly code in the system. As a result, the software consists of independent modules which can be individually replaced if the internal functions must be changed. The collective size of the programs is 34 kilobytes, which the author feels is not large considering the functions of the system and the low-level abilities of the host machine language. More to the point, it is difficult, approaching impossible, to write assembly code which is as bug-free as high-level code. It is better to have a correct, moderately sized program than an incorrect, small program. In fact, the program size could not be reduced greatly. The code generated by the PL/M compiler has been examined and found to be quite efficient.

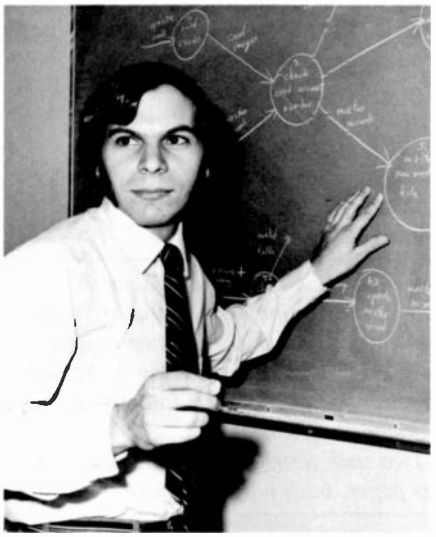
Summary

The result of applying these software techniques has been a coherent, highly modular and easily modifiable software system. Whenever changes are required, the data flow diagrams and structure charts provide a ready means of analyzing how the changes fit into the system and just what the implementation should be. Furthermore, the routines that implement the energy management algorithms and operator commands constitute less than half of the overall code. The remainder is the framework of a general control system for a set of satellite processors.

The scope of CEMS, together with its success, show the value of approaching microprocessor systems with high-level tools and techniques. As an example, the project had been underway for a year before any thought was given to software, yet the software development proceeded quickly once it was begun. The value of data flow analysis, in particular, is immense. It is estimated that, if more common techniques were used, the project would have taken months longer to complete. As an added benefit, the documentation of the internal workings of the system is automatically complete, for it consists mainly of the set of data flow diagrams, structure charts and supporting information (e.g., details of data flow) that were made throughout the project life cycle.

Bibliography

1. DeMarco, T., *Structured Analysis and System Specification*, Yourdon, Inc.
2. Yourdon, E., and Constantine, L.L., *Structured Design*, Yourdon, Inc.



Richard Poulo joined RCA in 1978 and soon began designing the software system for CEMS. Previously he had worked on designing an operating system for a Cray 1 computer, and on various other projects including simulation and graphics. He is currently investigating general software testing techniques and is testing a PASCAL compiler.

Contact him at:
RCA Laboratories
Princeton, N.J.
Ext. 3568

System design, program structure design and programming

A common example of a function that requires system design, structure design and programming is the conversion of a manually operated data processing system into an automatic system. The parameters of the new system are determined by the system to be converted which imposes certain restrictions on the new system.

The system designer must first analyze the manual system and define the problem. Data flow is analyzed and data flow diagrams devised for implementing automatic data processing.

The second step, and one which is fundamental to the success of the system, is the design of the structure of the programs that will process the data. It is at this stage that the solution to the problem defined in the first stage of development is composed. The structure of the programs is determined by the parameters of the manual system to be converted. The programs are structured by the priorities of the tasks to be performed.

The last step in the sequence of steps to be performed in the development of a system is the actual programming. Programs must be written to translate the data flow information provided in the first step into instructions that tell the computer how to process the data. The programs that are to be written have been determined by the overall design structure. Programming is the implementation of the solution that was composed in the second phase of the system development.

B.L.S.

M. Herskowitz|N.J. Fedele

Dedicated microprocessor-based testers for improved fault isolation

Microcomputer-based testers that are easily updatable and expandable, through reprogramming and modular hardware, bring objectivity to manufacturing testing.

Abstract: *As with any manufacturing step, testing must be made to be cost effective. One way to reduce the hardware costs involved in production testing is through the use of small, dedicated microcomputer-based testers.*

Two such testers are described in this paper, both based on Intel's SBC 80/20 single board microcomputer. The first system insures that BCD keyboards meet required specifications. The second verifies the presence of components on printed circuit boards prior to wave soldering. Both of these systems reduce manufacturing costs by finding failures early in the production cycle.

Due to the ever increasing complexity of consumer products, production testing is becoming an integral part of the manufacturing process. With the impact of microprocessor technologies, low cost dedicated test systems are becoming extremely popular for production testing. Microprocessor-based systems have the advantage of being able to be easily reprogrammed to satisfy a new test sequence using different parameter values. With modular hardware, most functions can be provided by simply plugging in the required interface and modifying the software. The only custom circuitry required is used for sensors, detectors, and signal conditioning between the computer and the unit under test. This level of standardization provides for easier maintenance and reduced tester costs, since the basic hardware may be reproduced many times.[1] For these reasons, many microprocessor-based systems are turning up in the factory environment.

Two such systems, both designed to identify component level faults, are described in this article. The first system, about to be introduced on-line, verifies the proper operation of keyboards on RCA Televisions. The second system, still undergoing feasibility study, hopes to monitor the performance of automatic component inserting (ACI) equipment by identifying missing components on printed circuit boards prior to soldering. Both systems are based on Intel's SBC 80/20-4 Single Board Microcomputer. The control programs for these systems reside in erasable programmable read only memory (EPROM) which facilitates program changes during the prototyping stage.

Reprint RE-25-3-13
Final manuscript received Aug. 24, 1979.

FSKT system

The frequency synthesis keyboard tester (FSKT) is a go/no-go system that ensures

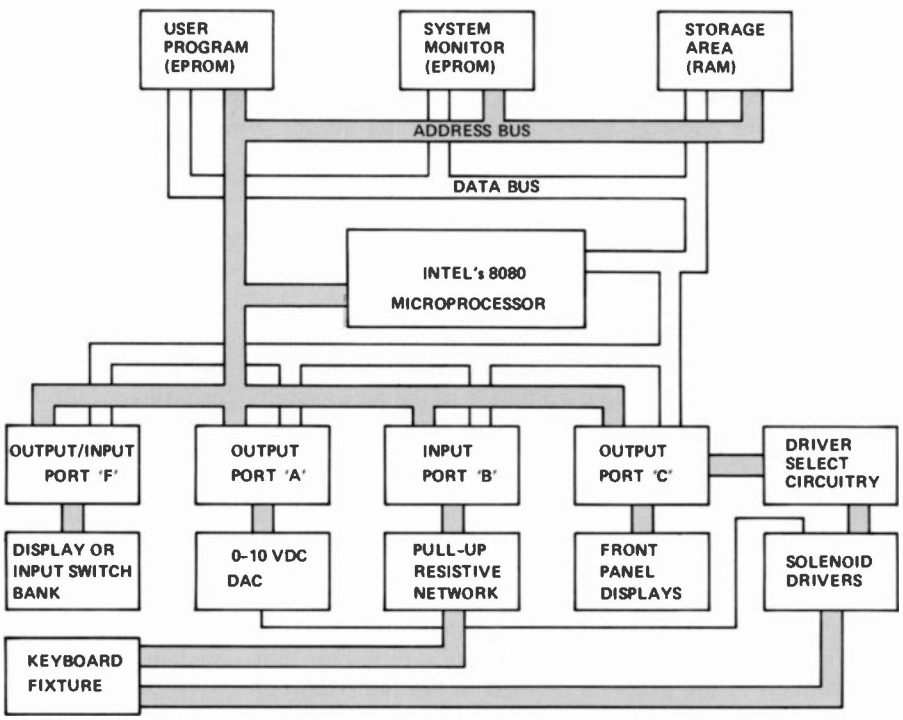


Fig. 1. Block diagram of the FSKT system showing the system organization.

proper operation of the channel-select keyboard used on TV sets equipped with the frequency-synthesis tuner. A system diagram is shown in Fig. 1. This system verifies minimum and maximum force specifications for each key, and ensures that the correct data lines are enabled prior to activation of the strobe flag. Solenoids are used to actuate the keys, since testing time is an important consideration in overall system design. A mechanical fixture positions the keyboard for testing. Operation of the keyboard by the fixture offers significant advantages over manual methods including: a more objective test procedure, reduced operator interaction, and reduced testing time.

System organization

A photograph of the FSKT is presented in Fig. 2. The interface from the central processing unit (CPU) to the keyboard under test is composed of the mechanical fixture and the driver/select board, which generates the stimulus and feeds the response back to the microprocessor. The driver select card interprets commands from the computer, selects the solenoid and applies the proper force to the key precisely positioned under the solenoid actuator.

The fixture, shown in Fig. 3, houses the twelve actuator solenoids. It automatically starts the test sequence, once the keyboard is properly positioned, and ejects the part upon completion of the test. The fixture includes a safety interlocking network that eliminates the possibility of injury to the operator.

Calibration and measurement

To use a solenoid to provide an accurate force stimulus, its force-voltage characteristic must be known. The force-voltage relationship for a solenoid is linear as expressed by Ampere's law. The relationship between force and displacement, however, is rather complex and will vary with different plunger and winding configurations. Typically, force values decrease significantly with increasing plunger displacement. For the given application, the plunger remains stationary until the solenoid force has overcome the impedance of the key. Thus, the displacement will remain constant while the force can be varied with voltage.

Since the voltage-force relationship is basically linear in nature, simple experimentation will reveal the correspondence for a particular solenoid. Figure 4 shows the mechanism setup that will allow a calibration table to be formed. In use, the cantilevered beam is displaced a given amount by applying a constant force against the plunger. The voltage on the coil is lowered until the plunger is displaced. At that voltage, the force is equal to or less than the force exerted by the deflected beam. The force due to the beam is measured by a strain gauge, while the voltage is recorded. The values can now be calculated and stored in program memory.

As mentioned, the use of solenoids in the FSKT system enables rapid key actuation. The time to complete a single pass through a twelve-key keyboard is approximately a second. Most of the time is consumed in the delay routine, waiting for the solenoid to become stable. The stabilizing time for the solenoids used is about 20 ms, as shown in Fig. 5.

The photograph, presented in Fig. 6, shows the input voltage (stimulus) applied to the solenoid (the upper portion), and the voltage (response) produced by the strain gauge as determined by the calibration procedure. The voltage produced by the strain gauge corresponds to a specific force value. Notice that the force always rises to the end level and never overshoots the specific value.

System operation

The FSKT performs a simple go/no-go test displaying a green light if a keyboard passes, a red light if it fails. The testing sequence proceeds as follows: after the operator places the keyboard in the fixture, a microswitch, located inside the mechanical fixture, signals the microcomputer that the keyboard is seated correctly and ready to be tested. Once the sequence is initiated, the running lamp is turned on and the program directs the D/A converter to output a voltage value corresponding to a force of 3 ounces. The program then checks to see if any of the signals from the keyboard are activated. If any are, the key has triggered below the lower limit, and the part fails and testing stops at that point. If signals remain deactivated, a larger force is applied and the lines retested. This procedure is repeated until the key triggers or the force required to trigger the key is greater than the upper limit, and the part fails. The strobe line is repeatedly examined at the appropriate place to ensure that it does not preceed any data line. This procedure verifies that, in actual keyboard operation, the data lines are not being read when they are in an indeterminate state. Once the program exits from the loop, the strobe line is again tested for closure, and the binary-coded decimal (BCD) code from the keyboard is verified. The zero key is a special case, because the BCD code for zero does not affect any data line. A counter, controlling the number of times the overall cycle is executed, is initialized with its value stored in the data table.

Upon completion of testing, the program signals the operator to eject the part, turns off the running lamp, and indicates whether the part passed or failed.

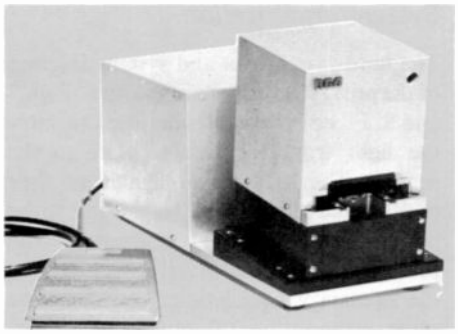


Fig. 2. Photograph showing the mechanical fixture (small box) next to the electronic package.

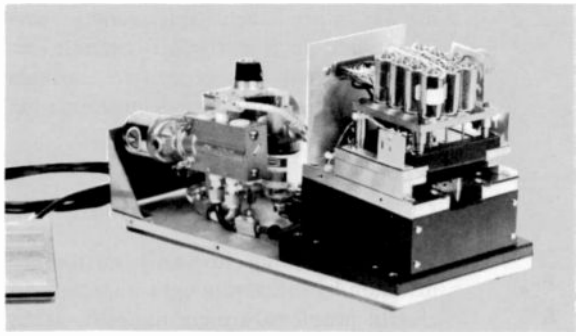


Fig. 3. The fixture, which contains the twelve actuator solenoids, automatically starts the test sequence and ejects the part upon test completion.

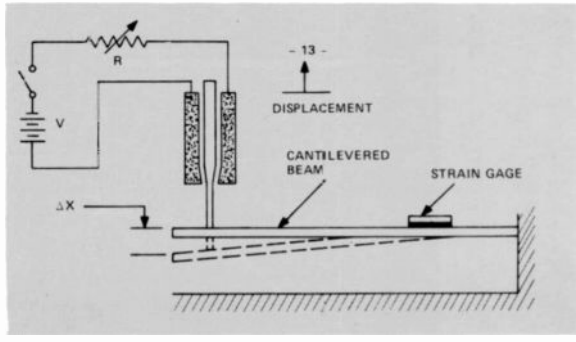


Fig. 4. The cantilevered beam is displaced a given amount (Δx) by applying a constant force against the plunger.

Post ACI system

The post automatic component inserting (ACI) verifier insures the presence of all passive components on production circuit boards immediately following their machine insertion. This is accomplished by comparing the light passing through the holes of a populated board to the light which passes through the holes of an unpopulated board. The system allows for variations in hole size and lead diameter, which adds flexibility and improves reliability.

System organization

The hardware layout and system diagram of the post ACI verifier are shown in Figs. 7 and 8. Fiber optic cables are used to carry the light from the fixture plate to the sensors. The 1024 sensor signals are then multiplexed down to a single line to be handled by the A/D converter. The program interrogates the sensors, performs the necessary calculations, and presents error information (if any) to the light-emitting diode (LED) display. The display consists of an unpopulated circuit board with the holes drilled out to accept small LEDs. The LEDs are wired on a 32 x 32 matrix with each intersection corresponding to a particular sensor. This display format was chosen to facilitate rapid identification of errors by production personnel.

Calibration and measurement

When attempting to verify component insertion by measuring light transmission, certain problems are encountered. Many different sized holes exist on the board, and even holes of the same size will contain parts with varying lead diameters. Furthermore, even with proper care, variations in light intensity across the face of the board are difficult to avoid, as are variations in fiber-optic and photocell characteristics. Because of this, no single decision point could be chosen that would be suitable for all holes on the board. Rather than calculate and store all different references in memory, a more flexible approach was chosen. Before testing begins, a separate calibrate routine is executed in which readings are collected for an unpopulated board. During the test mode, each sensor reading is compared with its corresponding calibration value. If sufficient* reduction in value occurred during the test, the hole is considered filled, otherwise it is not. The calibration need only be done once each day when the system is first powered up, but it may be desirable to calibrate to each lot of boards, since each one may be slightly different.

This calibration method has an added benefit. It allows spare sensors to exist in the system for handling engineering change notifications (ECN). During calibration, all 1024 sensors are interrogated without regard for their status. If the spare sensors are kept covered, their readings will be zero or near zero. During the test mode, each calibration reading is checked prior to testing of that sensor. If the reading is below some preset limit, the sensor is skipped. If a sensor needs to be added, a fiber optic link from the sensor to the fixture plate can be supplied (a hole must be drilled for it), and a corresponding LED wired into the display matrix. If a sensor is to be deleted, the fiber optic link can be removed and the sensor covered. In both cases, a recalibration will be required, which takes about one or two seconds.

* Currently, a 25 percent reduction is used as the cutoff, which represents the anticipated decrease resulting from a lead whose diameter is half that of the hole. This is viewed as a reasonable worst case.

System operation

The control panel consists of four lamps (PASS, FAIL, TESTING, and ERROR) and two pushbutton switches (CALIBRATE and TEST). Immediately following application of power, or after a reset, the system is in the calibrate mode, as indicated by a flashing ERROR lamp. At this time, the operator places an unpopulated board onto the fixture and presses the calibrate switch. After the system calibrates, the PASS lamp will be on, unless one or more sensors saturates, in which case, the FAIL lamp will be on and the ERROR lamp will continue to flash. The calibrate switch is still active during this condition, allowing for the problem to be corrected. This feature is added as a precautionary measure only, as sensor saturation is not expected to be a problem.

After calibration is complete, the operator places a populated board onto the fixture and presses the "test" switch. All currently used sensors are compared with their calibration readings and the first four errors encountered (if any) will be sent to the LED display. The test status (PASS or FAIL) is sent to the control panel, and an audible alarm sounds on failure. The test procedure can then be repeated.

In addition to the confidence check which exists for the sensors, there are two checks for displays. Following a reset, the four status lamps all come on for ap-

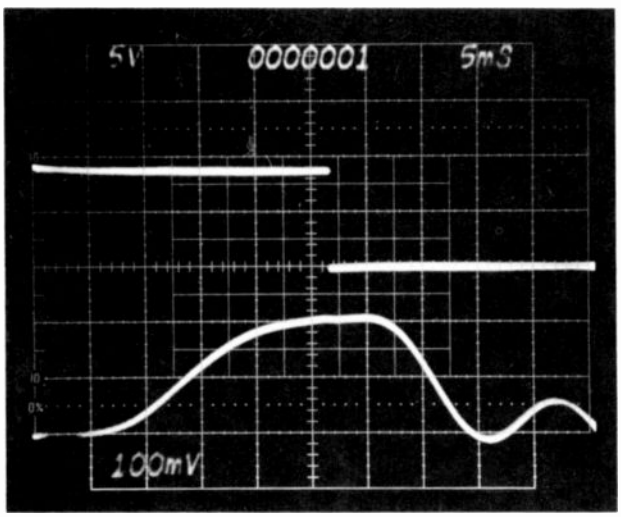


Fig. 5. The stabilizing time for the solenoids is about 20 ms.

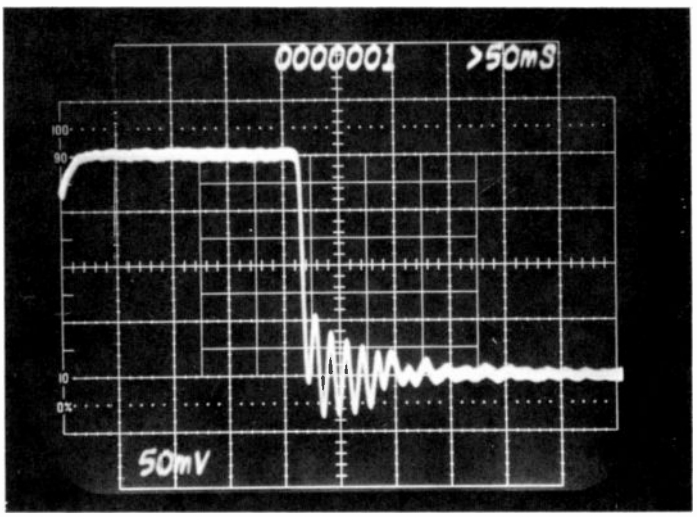


Fig. 6. Solenoid response waveform as measured by the strain gage.

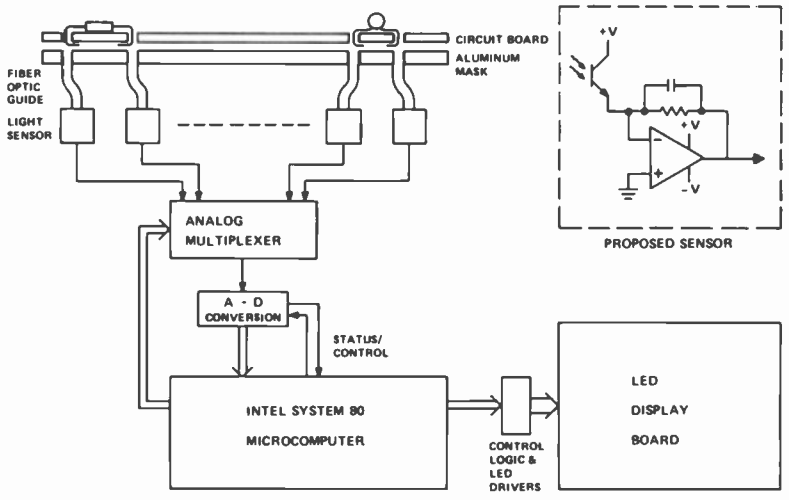


Fig. 7. The hardware layout is shown in this diagram of the system architecture for the ACI verfication project.

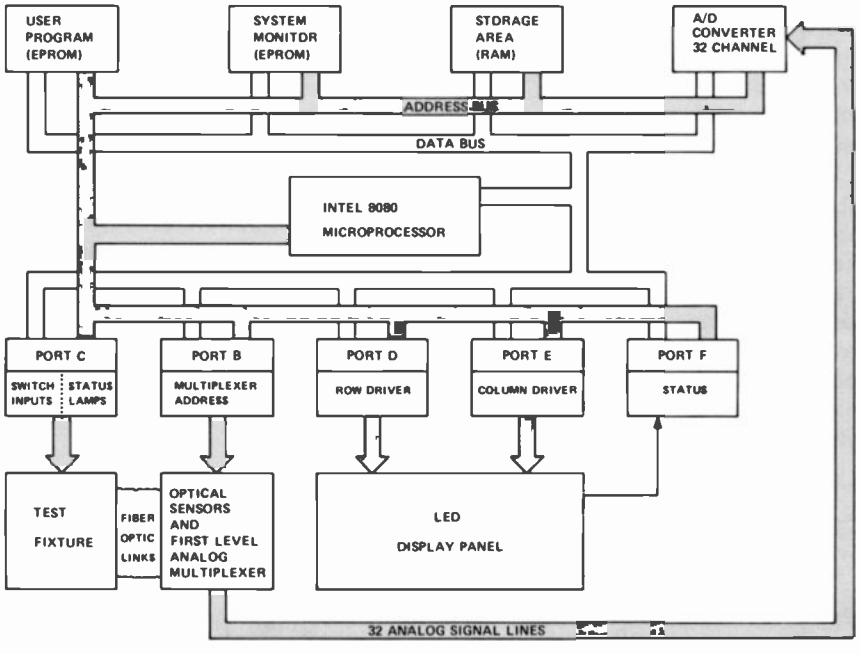


Fig. 8. The program interrogates the sensors, performs the necessary calculations, and presents error information to the LED display.

proximately a second allowing for a visual verification. The LED display has a built-in sensing circuit which checks for proper current flow each time a LED is accessed. If one fails to come on when accessed, the system halts with the LED address latched to an internal display. This error check for the LED display was chosen due to the high current demand which would result from simultaneously illuminating 1024 LEDs.

Conclusion

The two flexible microcomputer-based testers, that have been discussed, offer more effective performance than the current manual test approach. The present keyboard test method consists of connecting the keyboard to a working television receiver and observing keyboard operation on a two-digit display. This approach is subjective and time consuming. Furthermore, it does not examine the keyboard sufficiently to identify all its possible failure modes.

The television circuit boards are currently being visually inspected by an operator or are not inspected at all prior to soldering. Failure to identify missing components prior to soldering increases the cost of replacement. Furthermore, if the missing part is not identified prior to the time that power is applied to the circuit board, the possibility exists that damage may occur to other components.

Both of the test systems described are capable of performing the required tasks more objectively than they are now being performed and can also provide increased throughput. Each can be reprogrammed to accommodate new test specifications and each makes use of modular hardware. These features give microcomputer-based testers longer usable lives with reduced development cost and time.

Acknowledgment

The authors are extremely grateful for Angelo Marcantonio's contributions to both projects.

Reference

1. "Microcomputer for Test and Control Applications," *RCA Engineer*, Vol. 22, p. 64 (Feb./March, 1977).

Herskowitz works at the terminal while Fedele watches.

Martin Herskowitz joined RCA Laboratories in 1974. After receiving a B.S. in Applied Science Engineering in 1979, he joined the Microsystems Research group where he now designs automated test systems for CE manufacturing.

Contact him at:
Microsystems Research
RCA Laboratories
Princeton, N.J.
Ext. 2194

Nicola John Fedele joined RCA Laboratories in 1978 and, since then, has been involved with microprocessor research and development. Mr. Fedele is presently engaged in the design and development of hardware and software systems for microprocessor-based automatic test and control equipment.

Contact him at:
Advanced Systems Research Laboratory
RCA Laboratories
Princeton, N.J.
Ext. 2622

on the job/off the job

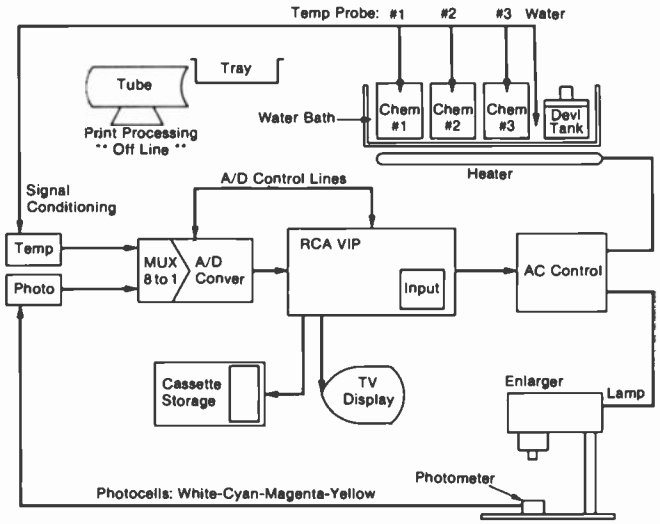
A quick survey of how RCA people use microprocessors at home and at work

Automated darkroom

Darkroom work involves all processes needed to convert a roll of film into a print. Basically, these processes include film development, print exposure, and print development. Development requires processing the film or paper in a series of chemical baths in which duration and temperature must be controlled. Exposure requires analyzing the quantity of light received at the unexposed print. This amount of light is used to compute a time of exposure. If a color print is being exposed, the quality of the light must also be analyzed; that is, the amount of each primary color must be known so that correction filters can be used to correct any imbalance of the light, which will cause an incorrect rendering of the picture.

Currently, I am using my RCA-VIP home computer as a timer. I input a series of times into the VIP which correlate with the processes I wish to monitor. For each process, the computer times down and provides a 'beep' at completion. All other factors such as temperature and light quality and quantity are monitored by other means.

In order to use the computer to its fullest extent, I am constructing an automated darkroom. Temperature control is provided by means of a heater and temperature probes. The heater is used to keep the water bath at a constant temperature. The temperature probes are used to monitor the water bath and chemicals. Exposure control is provided by an enlarger lamp controller and a photometer. The photometer is used to monitor the quantity of the light, the white photocell, and the quality of the light of the color photocells. In this system a TV monitor is used to display the output of the photometer and the temperature probes along with any status information that can be presented for processes which are conducted in normal room light. The cassette unit is used to store the programs and data for use by the computer.



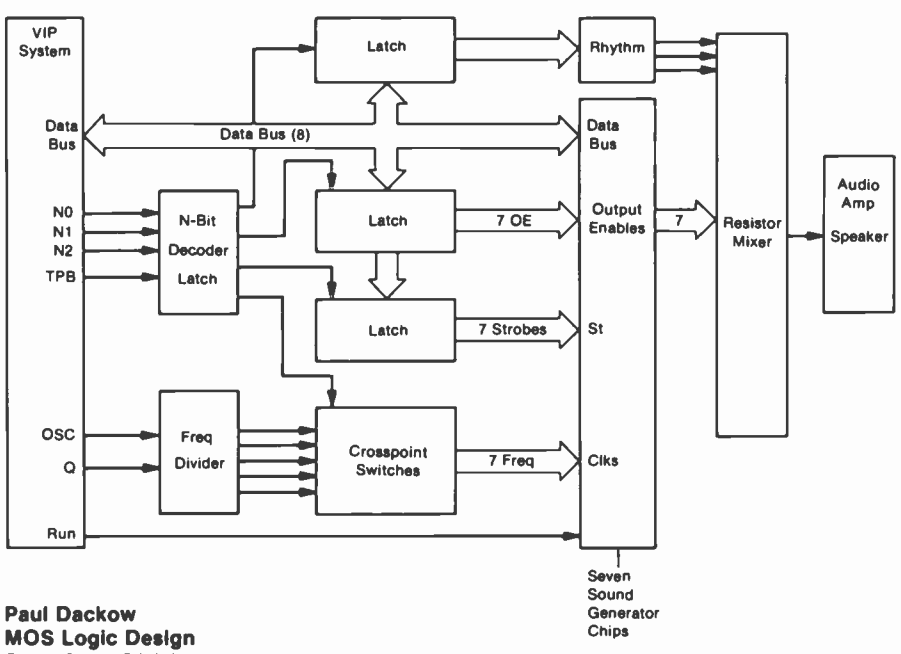
Stanley Gollaszewski
Missile and Surface Radar
Moorestown, N.J.
Ext. 3764

COSMACDAC music generator

This VIP microprocessor controls a music generator system. It features seven separate melody channels and three rhythm channels (bass drum, snare drum, wood block) all mixed into one or two outputs. Although similar in some respects to the VIP sound boards, there is less emphasis on frequency synthesis and envelope generation, and more emphasis on the software management and control of all the channels.

The six octave range is controlled in software, as are the speeds, repeats, relocations, and other control functions. Each note for a channel is stored in one memory byte, containing frequency and duration information.

There are seven pieces presently in the song library, ranging from one to eight minutes in length.



Paul Dackow
MOS Logic Design
Solid State Division
Somerville, N.J.
Ext. 6836

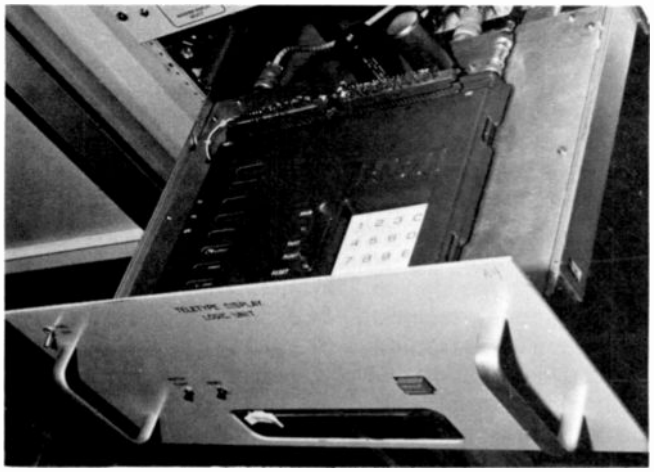
Use of the RCA COSMAC VIP in a teletype display system

An outmoded teletype display system at the TRADEX Site, Kwajalein, was recently upgraded to a more flexible and versatile system through the use of the RCA COSMAC VIP Microcomputer. The new system replaces a noisy and bulky TTY Televiewer with a system consisting of a logic unit, the VIP, an existing TV display controller, and several TV display monitors. The heart of the system is the VIP, and through its use the capability of the display system has been considerably enhanced.

Incoming TTY data are converted from Baudot to ASCII code in the logic unit and then input to the VIP via its byte I/O port. CDP1802 machine language software in VIP ROM processes each character to determine validity, remove parity, and to format and output the data to the display controller. Data are transferred to the controller a character at a time for upper case numbers and symbols, and a word at a time for lower case letters. Messages are displayed beginning on the bottom line of the display and are scrolled upward as a line of data is accumulated. Words are checked for length and, if too long for the current line, are displayed on the next line. The TTY displays are time-shared with tracking sensor track file status displays during TRADEX mission support.

P.G. Vise
Service Company
Cherry Hill, N.J.

J.B. Galpin
Missile and Surface Radar
Moorestown, N.J.



VIP intrusion alarm

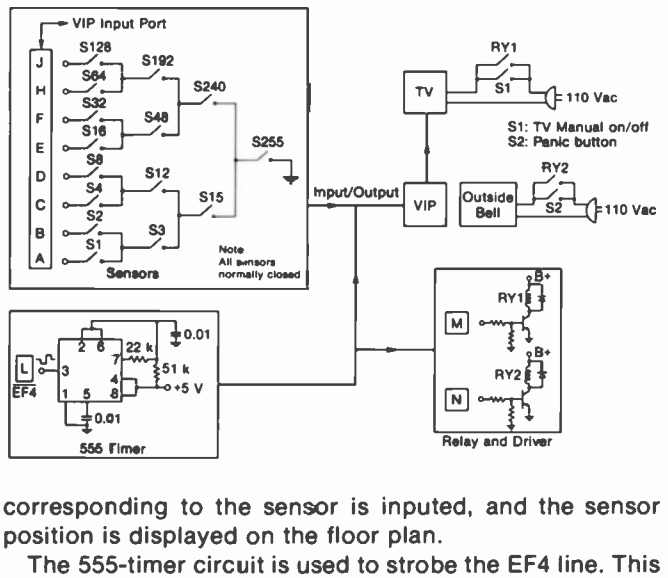
My intrusion alarm was designed to run in a basic 2-k VIP system. The software includes a modified Chip 8 interpreter and Chip 8 program. The hardware consists of fifteen sensors, two relays with drivers, a 555-timer circuit, and an external power supply.

The 15 sensors are constantly being monitored for a violation. If a sensor is activated, the TV is turned on and house floor plan is displayed on the TV. A dot is displayed, on the floor plan, to indicate the room in which an intrusion occurred. An internal alarm is then sounded for one minute. This internal alarm is to awaken me if an intrusion occurs at night when I am at home. The outside alarm is sounded for one minute. The system automatically resets after the outside alarm is turned off. A panic button is also provided to turn on the outside alarm. Time delays are incorporated to allow me to exit and enter the house without sounding the outside alarm.

With some software modifications and appropriate sensors, the system can be adapted for monitoring fire sensors, water overflow sensors, etc.

The floor plan was constructed by using the VIP Video Display Drawing Game in the *RCA COSMAC VIP CDP18S711 Instruction Manual.*

The sensors are connected to the VIP input/output port. When all sensors are closed (normal position) the eight input lines are all zero. If a sensor is activated, a number



corresponding to the sensor is inputed, and the sensor position is displayed on the floor plan.

The 555-timer circuit is used to strobe the EF4 line. This circuit controls the inputing of sensor conditions.

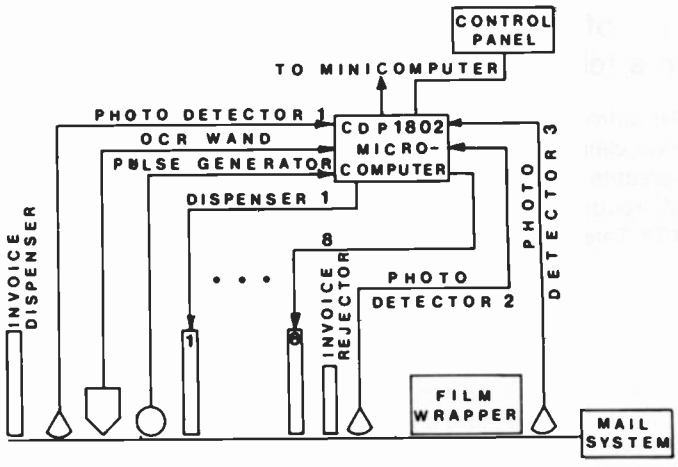
The relay and driver circuits control the TV and outside bell. A byte is outputed to turn these on and off.

Clifton Criswell, Jr.
Consumer Electronics Division
Indianapolis, Ind.
Ext. 5936

Microcomputer application in mail order warehousing

Microcomputers play a major role in the automatic order fulfillment system (AFS) currently under installation at RCA Records Indianapolis operation. The system consists of four automatic packaging lines each supported by a microcomputer. In each line, customer invoices are read by an OCR wand, following which product is automatically packaged with the invoice. The microcomputers work in a real-time mode, receiving data from the wand, and simultaneously transmitting them to the machine for operations monitoring and control and to a minicomputer for data processing.

Each microcomputer is configured around a RCA CDP-1802 microprocessor with the appropriate interfacing for inputs and outputs for the minicomputer and the packaging machine.

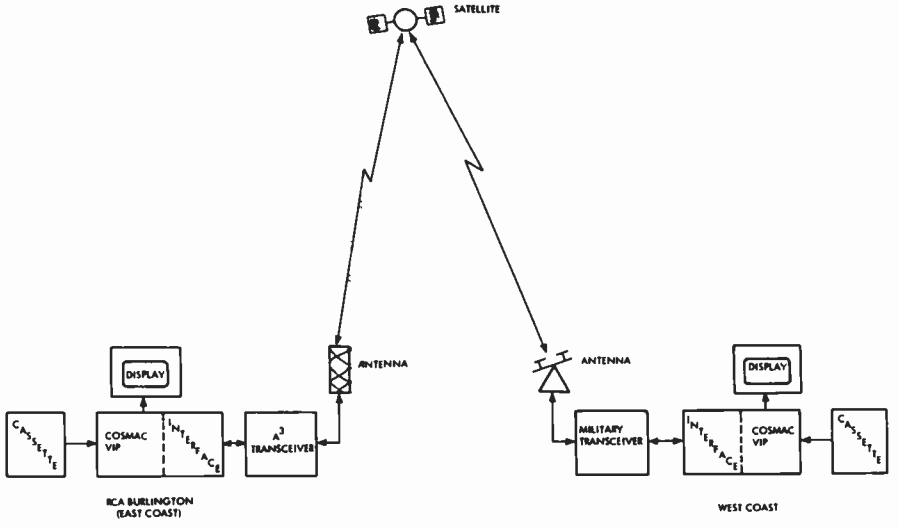


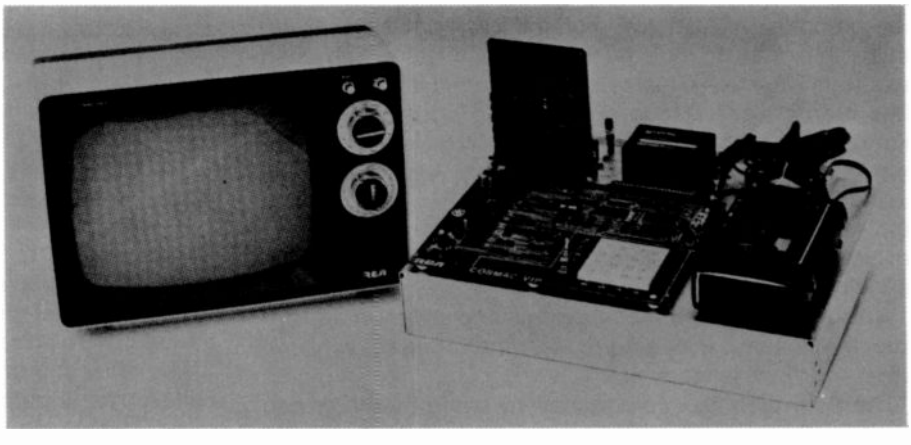
H. Nanda|K. Ringeman|H. Shreder
RCA Records
Indianapolis, Ind.
Ext. 6484

Satellite communications link evaluation using COSMAC VIPs

One of the major elements of the advanced autonomous array (A^3) surveillance buoy system, which is under development at Automated Systems in Burlington, Mass., is a bidirectional digital UHF satellite communications link. A pair of COSMAC VIPs are employed to automatically conduct a sequence of tests to evaluate the reliability of this link.

Each VIP system is equipped with a special interface board which contains control circuitry to interface the VIP with one of two types of satellite transceivers, as well as additional memory. The test software is written both in the CHIP-8 interpretive language, for operator interface, and in COSMAC assembly language for conduction of the the tests. All aspects of the tests, from transceiver control and message generation to bit error detection and display, are performed by the VIPs. During the tests, the two VIPs alternate their functions. One unit acts initially as a transmitter controller and then as a receiver controller while the second unit performs the complementary function. When both units have transmitted and received a message, they perform a bit error detection operation, display the results for the test operator, and then resume the next test cycle.





Paul Berrett
Automated Systems
Burlington, Mass.
Ext. 3793

An automatic ROM tester

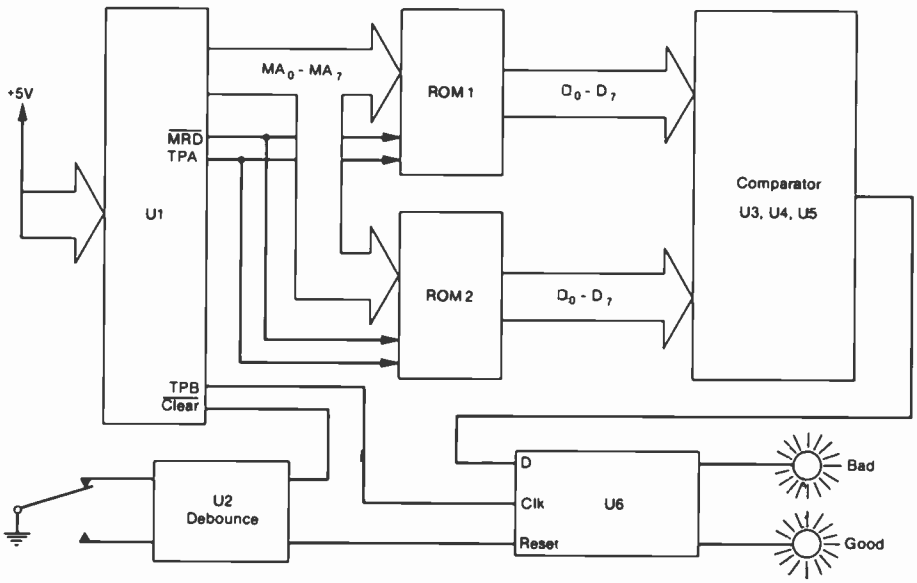
In the mass production of systems using read only memory (ROMs), the factory test of the chips is often a problem. At Automated Systems, this difficulty was resolved by using a very simple 1802 microprocessor circuit, with no memory.

The advantage of using an 1802 microprocessor in testing of ROMs is quite easy to see—it eliminates the need for no less than 14 special circuits. With this computer-controlled automatic equipment, 64,000 tests are executed in one-fourth of a second.

The simple circuit described below performs an automatic functional test of CMOS ROMs. It works by comparing each address location on a known-good chip with the corresponding location on a chip under test, and lighting an error light if there is any discrepancy.

These ROMs are not standard memory components, but use a multiplexed address bus peculair to the 1802 micro-processor. The one instruction, subtract memory immediate (SMI), is wired in. When the test button is pushed, the program computer resets to 0 where it sees the instruction SMI wired in. The subtrahend is loaded from *P + 1*, subtracted from the accumulator, and the *P* counter is indexed to repeat the process. The test sockets are addressed through 64,000 locations in 0.25 second and compared in real-time. If a single address or data error occurs, it is permanently registered, and an error light is lit.

The 1802 presented the easiest way to generate the multiplexed address and the proper timing signals. Ordinarily, designing a microprocessor into a device this simple would require more hardware design than all the rest of the unit, and would also require at least some software to be written. However, the 1802 requires very little support hardware in a minimum configuration (in this case a debounced pushbutton for the reset, a crystal for the clock and a handful of pullup resistors to tie off unused inputs), and this lent attraction to the idea.



Moreover, there is no real need for a program. If we use hardware comparators, all that is needed is a ploy to cycle the program counter through all possible memory locations. Any instruction which does not change the program counter will work. We do not put this instruction in memory, though—we wire it directly to the data bus with resistors. At the standard clock rate of 2.0 megahertz, the 1802 executes each instruction in 8.0 microseconds and will cycle the program counter through all 65,536 memory address locations in approximately 0.5 second.

As an added bonus, if the instruction is an immediate type, the program counter will go twice as fast, completing an entire test cycle in only one-fourth second. On the 1802, the operation code FF (SMI) serves this purpose.

Once the address bus is taken care of, the outputs of the two memory chips must be compared. Exclusive OR gates U3 and U4 compare the outputs, and U5 ORs these together to produce an error signal (see diagram). This signal, however, is only valid after the lower address is on the bus and before the upper address for the next location appears. The 1802 provides a signal, TCB, during this period. This signal is used to clock the error signal into U6A, where it is used to set a latch and light an error light.

Mike Gilbert
Automated Systems
Burlington, Mass.
Ext. 2872

Proud of your hobby?

Why not share your hobby with others? Perhaps their interest will make your hobby more satisfying. Or maybe you'll find others who already share your hobby and who can help make your own efforts more rewarding.

The *RCA Engineer* likes to give credit to engineers who use their technical knowledge away from the job. We've published articles about subjects as diverse as a satellite weather station, model aircraft and railroading, solar heating, an an electronic fish finder.

For more information on how you can participate in this feature of the *RCA Engineer,* call your local EdRep (listed on the inside back cover of the *Engineer*) or contact Frank Strobl (222-4220) or Betty Stotts (222-4255) at the *RCA Engineer.*

P.P. Fasang

Microprocessor/software-oriented educational program at RCA

Corporate Engineering Education offers a program consisting of ten microprocessor/software courses with broad audience appeal that will help technical personnel stay up-to-date.

Abstract: *This paper briefly describes the history of microprocessors and their applications. The computer/software engineering curriculum, identified and developed for RCA's technical personnel, is presented, along with the educational philosophy behind this curriculum. Current and future course offerings are enumerated.*

Brief history of microprocessors

The understanding and market acceptance of any revolutionary new product are slow to come, particularly for a product as complex as the microprocessor. It took several years before systems designers fully understood the versatility of microprocessors. Also, before managers could justify using this device in new product lines, manufacturing processes had to be established and refined.

The first 4-bit commercial microprocessor, the Intel 4004, appeared on the market in 1971. This device, developed primarily for calculator-oriented applications, is a monolithic integrated circuit (IC) employing large-scale integration (LSI) in metal-oxide semiconductor (MOS) technology. The 4004 was soon followed by a variety of microprocessors, with most of the major semiconductor manufacturers producing one or more types. The Intel 8008 was the first general purpose 8-bit microprocessor. It became widely available in 1972. This was followed by the Intel 8080 and then the Intel 8085, each "generation" being better than the previous one because of improvement in the design and in the manufacturing processes.

RCA introduced the 1801 two-chip microprocessor in 1975. This was quickly replaced by the single-chip 1802 microprocessor which employed silicon gate CMOS technology. By 1976, there were over two dozen microprocessors available from over a dozen manufacturers.

Microprocessors today

Although less than ten years old, already microprocessors have changed the way many electronic systems are designed and implemented. The microprocessor industry is probably the fastest growing one today. Its effect on the economy has been qualified as the "second industrial revolution." A complete microcomputer can now be implemented on a single chip for a few dollars in large quantities. It brings "programmed intelligence" to all the processes that surround us.

In a sense, we can say that the digital computer is disappearing: it is being incorporated into other equipment and systems. As more microprocessors are incorporated into equipment, the demand formerly placed on the central main-frame computer becomes less. In some instances as more microcomputers move into equipment, we approach having a "distributed-intelligence" system. An example is a grocery store check-out cash register system in which each cash register has a certain amount of "intelligence."

Microprocessors today find applications ranging from grocery-weighing scales to intelligent computer terminals. The microprocessor has not only made system design simpler (in terms of IC chip count), but also more powerful in terms of computational and logic capability.

At RCA, microprocessors have been used in different applications such as testing of video discs, vehicle engine testing, spacecraft command and control systems, radar systems, telecommunication equipment, and computer-aided manufacturing.

The program and philosophy

Recognizing the importance of the microprocessor, Corporate Engineering Education (CEE) began the development of a computer/software curriculum in 1978 (Fig. 1). The objectives of this curriculum are twofold. One is to help RCA technical personnel who have no background in microprocessors learn from ground zero the essence of microprocessors so that they can make the transition in their design work from using random logic to using

Reprint RE-25-3-15
Final manuscript received July 20, 1979

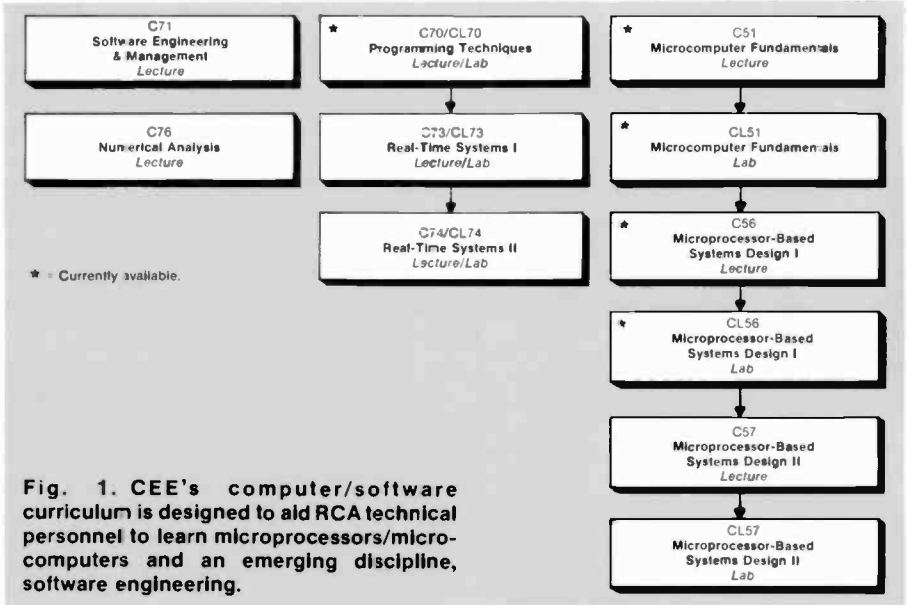


Fig. 1. CEE's computer/software curriculum is designed to aid RCA technical personnel to learn microprocessors/microcomputers and an emerging discipline, software engineering.

programmable "intelligent" chips. The other objective of the curriculum is to provide instructional material in an emerging discipline known as software engineering (Fig. 2). This discipline encompasses the process of specification, definition, analysis, algorithm formulation, coding, verification, and measurement of performance of computer programs at the application as well as the systems levels.

CEE's philosophy in this curriculum, as in others, is to provide educational materials to the broadest audience within the Corporation in such a way that the materials are up to date, relevant and applications-oriented, while still maintaining a high degree of theoretical framework. Emphasis is on understanding of basic concepts; thus, what the student learns is portable.

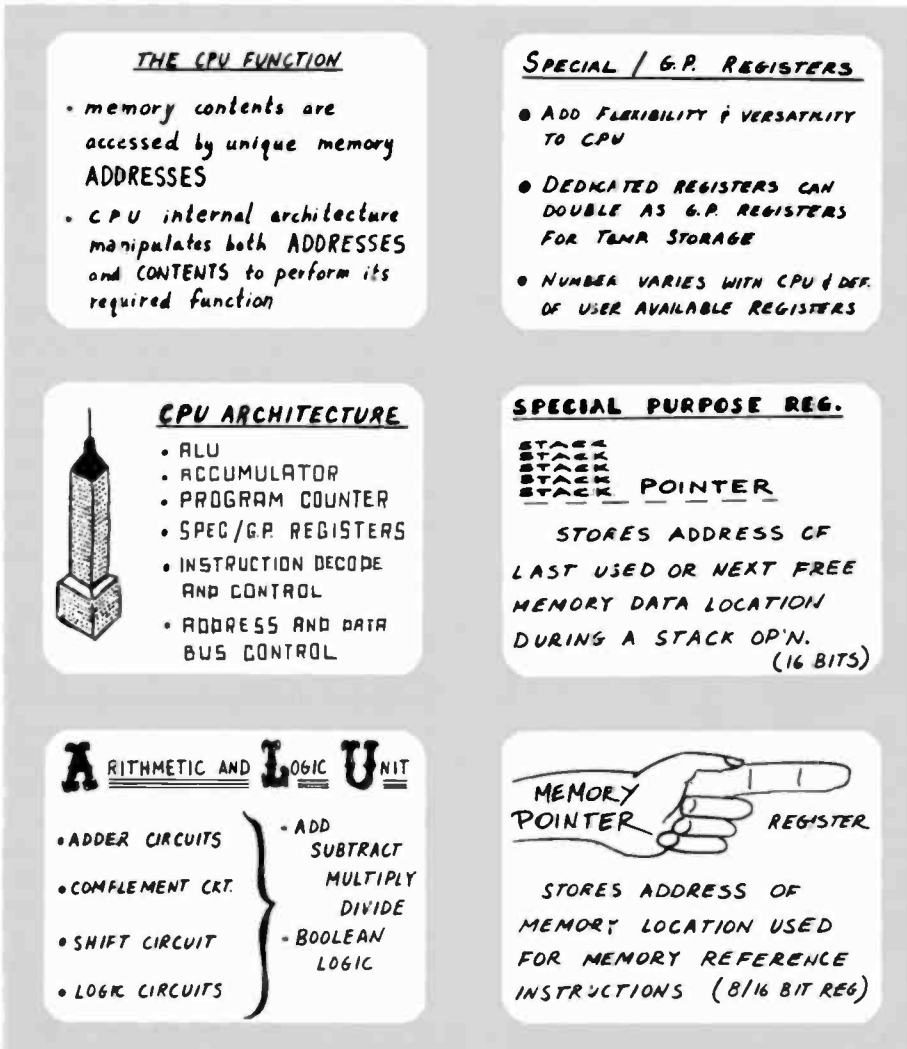


Fig. 2. Each student receives extensive course material such as these sample visuals from the course, *Microprocessor-Based Systems Design.* Course handouts, including copies of all visuals, illustrations and tabular material used in the videotape lectures, eliminate the need for copious note taking during the lecture. The author (and artist) for this course is **Jospeh P. Paradise** of the Solid State Division, Somerville, N.J.

Course contents

C51: Microcomputer Fundamentals (Lecture) is the first course in the microprocessor course series. It is an introductory course, covering such topics as binary codes, logic gates, flip-flops, registers, counters, binary addition and subtraction, VIP (Video Interface Processor) programming, subroutines, and input/output (I/O) structure of the RCA 1802 microprocessor.

CL51: Microcomputer Fundamentals Laboratory Exercises consists of a series of ten laboratory exercises, each to be programmed, tested, and debugged at an RCA COSMAC VIP™ computer system. Programming is done at the interpretive CHIP-8 (Computer Hobbyist Interpreter Program) language level. Topics covered include the use of the audio cassette recorder as an I/O device, development of programs to obtain various video configurations by using programming techniques such as looping, branching, moving images, and subroutines.

C56: Microprocessor-Based Systems Design I (Lecture) is the second course in the microprocessor course series. This course lays the theoretical framework upon which applications can be created. Topics covered include microprocessor organization, types of instructions, timing consideration, addressing techniques, memory and programmed I/O, interrupt and DMA (Direct Memory Access) I/O, I/O communication, peripherals and peripheral interfaces, microprocessor system development aids, microprocessor family survey, and system design considerations.

In addition to microprocessor/-microcomputer and software engineering courses, CEE offers other courses to RCA personnel, such as communication theory, control systems, phaselock techniques, digital signal processing, mathematics, and optoelectronics. For additional information or a copy of the CEE catalog, contact your local training office or call Margaret Gilfillan, TACNET 222-5255.

Fig. 3. Top: Filling part of an ever-present need for associate instructors, Jim Gentry, GCS, Camden, holds a discussion at a pause point in a video tape in C51, *Microcomputer Fundamentals.* Bottom: Jim is shown talking to a student in the laboratory course of CL51, *Microcomputer Fundamentals Exercises.* Associate instructors are valuable assets to the success of the CEE program.

CL56: Microprocessor-Based Systems Design I Laboratory Exercises consists of a series of eight laboratory exercises and a four-week team project. Topics covered include timing loops, keyboard interfacing, display system memory mapping, serial data transfer, parallel I/O with examples of handshaking protocols, interrupts and the VIP display, and time and space optimization. The team project may be a combination of hardware and software or software only.

C57: Microprocessor-Based Systems Design II (Lecture) is currently under development. This course is largely devoted to applications of microprocessors in a wide range of uses. Case studies of microprocessor-based products, subsystems, and systems which have been developed and designed by RCA engineers constitute the major part of the course. Also included in the plan is a session on the demonstration of the utility of a general microprocessor-based design development system.

CL57: Microprocessor-Based Systems Design II Laboratory Exercises will supplement the lecture portion (C57), and will emphasize concepts underlying the development of a system or subsystem which includes both hardware and software components.

C70/CL70: Programming Techniques (Lecture/Lab) covers programming styles, methodology of development, structure, ease of testing and modification, documentation, and related considerations. Topics discussed include searching and sorting techniques, binary trees, data structures, linked lists, stacks, queues, structured programming, and top-down programming.

C73/CL73 and C74/CL74: Real Time Systems I (Lecture/Lab) and II (Lecture/Lab) have not yet been developed. Although the contents of these courses have yet to be defined, the aim here is to identify and convey a set of software tools which can be used in real-time computer operating systems.

C71: Software Engineering and Management is designed to give an overall perspective of software design. Topics covered include characterization of software, concept of data flow, control flow, software testing, software development processes, phases of development, required specifications and considerations on software reliability, and structured programming. The overall software development process is stressed from a management point of view.

C76: *Numerical Analysis* is the last course in the present curriculum. The envisioned objective of the C76 course is to introduce the course participants to the general subject of numerical techniques and to give them an appreciation of the different approximation techniques for computation made possible with the advent of modern computational techniques and capabilities. The subject is sufficiently general in nature so that participants who are engaged in diversified areas of technical work which require computational results can benefit from the course.

Table I. Current course offerings in the computer/software curriculum (as of July 1979).

Course	*Released*	*Enrollment-to-date*
Microcomputer Fundamentals Lab (CL51)	June 1978	163
Microcomputer Fundamentals Lecture/Lab (C51/CL51)	Sept. 1978	733
Microprocessor-Based Systems Design I Lecture (C56)	Feb. 1979	198
Software Engineering and Management (C71)	May 1979	26
Two additional courses which became available in the fall are:		
Microprocessor-Based Systems Design I Laboratory Exercises (CL56)	Sept. 1979	*
Programming Techniques (C70/CL70)	Sept. 1979	*

* Data not available at time of writing.

Current and future course offerings

Six courses in the computer/software curriculm are now available. Over 1100 engineers, technicians, and managers have enrolled in the four courses available during the 1978-79 academic year (Table I). The remaining five courses are in the early planning stages of development. They will very likely be developed in the order given here:

- Microprocessor-Based Systems Design II Lecture (C57)
- Microprocessor-Based Systems Design II Lab (CL57)
- Real-Time Systems I Lecture/Lab (C73/CL73)
- Real-Time Systems II Lecture/Lab (C74/CL74)
- Numerical Analysis Lecture (C76)

Lecture/lab combination

Whenever it is feasible, a laboratory course is offered to supplement the lecture course (Fig. 3). These labs have been very well received. They help the students obtain hands-on experience with what they are learning, or already have learned, in the lecture class.

As part of the laboratory course offering and through the generous cooperation of the Solid State Division, participants in a microprocessor laboratory course may purchase an RCA VIP microcomputer at a price substantially lower than the list price. Thus far, through this offering, many participants have purchased VIPs (Fig. 4). With a VIP at home, a participant can "play" with the microcomputer at leisure and learn programming and interfacing techniques at the same time. This is very desirable because it makes learning fun, and with this in mind, CEE encourages more participants to have their own VIPs.

Conclusion

To meet the needs of rapidly changing technology and to help the technical personnel to stay current, CEE is developing a computer/software curriculum which has two components: microprocessors and software engineering. CEE is now past the midpoint of this development effort. Judging from the number of students enrolled in the courses and the feedback received from them, the curriculum has been extremely well received.

References

1. *RCA Engineering Education Catalog*, 1979-1980, published by RCA Corporate Engineering Education, Cherry Hill, NJ 08358, pp 10-19.
2. Shapiro, Louis, Ph.D., "Resources Available for Learning Computer Science," *RCA Engineer*, Vol. 23, No. 6, pp 12-19 (Apr./May, 1978).
3. *Intel MCS-85 User's Manual*, Sept. 1978, Intel Corporation, 3065 Bowers Avenue, Santa Clara, CA 95051, pp 1-1.
4. *RCA Microprocessor Products: Program Development Guide for the COSMAC Microprocessor*, MPM-102, RCA Solid State Division, Somerville, NJ 08876, p 2 (1975).
5. *RCA Microprocessor Technology: Twenty-Four Articles by RCA Engineers and Scientists, Microprocessors—Design in Transformation*, A special issue published by the *RCA Engineer* Staff, pp 76-89 (1977).
6. Lewis, D., and Siena, W.R., "Microprocessors or Random Logic," (three-part series), *Electronic Design*, Vol. 21, No. 18 (Sept. 1, 1973).

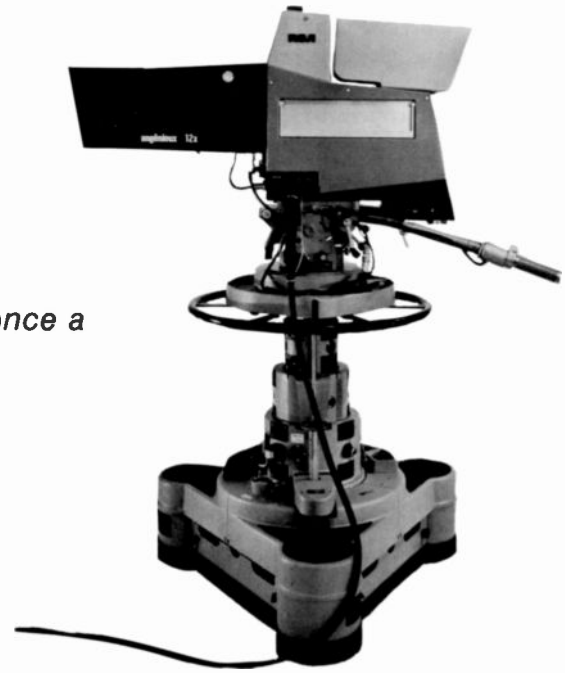
Pat Fasang is Administrator, Engineering Education Programs, Corporate Research and Engineering. He is engaged in corporate education need assessment, course design and implementation, education consultation and is a co-developer of CL56, *Microprocessor-Based Systems Design Laboratory Exercises*. At present, he is developing C57, *Microprocessor Applications*.

Contact him at:
Engineering Education
Cherry Hill, N.J.
Ext. 5020

L.J. Thorpe|R.A. Dischert

The TK-47 Autocam television camera

A built-in microcomputer makes camera alignment, once a time-consuming chore, a fifteen-second task.



Abstract: *The new TK-47 camera greatly simplifies the chore of camera alignment, both for "major" or component-replacement alignment and the "daily" pre-use alignment. It does this by using detectors to analyze video information from the camera and it employs a microcomputer to compute corrections which are then transmitted back to the camera via a digital data bus. Alignment is very fast, taking only fifteen seconds for the daily check, and multiple alignments for different operating conditions can be stored.*

Despite the considerable simplification afforded by the Plumbicon tube itself and the incredible circuit versatility made possible by the new integrated electronics, color television cameras are today complex and sophisticated systems. The diversity in camera designs centers principally about the optical splitting mechanisms and the attendant video-processing and picture-enhancement schemes employed. The art of color-camera design, however, stubbornly remains one of careful extraction of tiny electrical signals from photo-sensitive pickup tubes, followed by extensive electronic correction and processing to compensate for the ever-present aberrations of the camera's basic components, namely, optics, pickup tubes, and yokes.

The actual electrical corrections made in any camera are unique to its particular complement of components, a fact which still negates the dream of producing the true "black box" camera. An optimum final picture requires many compensating controls and an alignment procedure that requires careful iteration and attention to many interacting parameters. The vagaries in color-camera design have heavily hinged upon the placement of these controls, that is, whether they are located in the camera head itself, in a remote camera control unit (CCU), or on a video operator's control panel.

Reprint RE-25-3-16
Final manuscript received July 26, 1979.

The problem — maintaining alignment

Keeping a color camera aligned optimally — be it studio camera, outside broadcast camera, or handheld portable camera — remains a task involving disciplined realignment of pertinent controls. Historically, this has been an undesirable investment in time, effort, and finally, cost.

Today's small portable self-contained cameras, made possible by smaller pickup tubes and more powerful integrated circuits, heralded the debut of the unattended color camera. Now, cameras could operate satisfactorily for extended periods of time without the encumbrance of conventional sophisticated technical control panels. However, these cameras remain unique in that their technical setup is based on a stable pre-alignment stored in dozens of potentiometers contained within the camera body itself. Periodic checkout remains a necessity and the changing of a major component, such as a pickup tube, still requires a careful step-by-step realignment, made more tedious by the inevitable compactness of such a camera's electronic packaging.

The solution — a digital memory

However, a precedent had been set in camera design. The logical evolutionary advance from this point was to use a modern digital memory to store all of the camera alignment data that had been kept in the bank of potentiometers, along with convenient electronic access to this memory for periodic inspection, and when necessary, readjustment or updating. This concept is the basis of RCA's TK-47 "Autocam" design.

The development effort on the new TK-47 high-quality outside broadcast and studio camera was approached on three fronts:

1. The design of a basic camera that coupled the best of prior art and experience with the innovations developed in the electronic news gathering (ENG) type cameras to produce a highly stable, reliable, and simple camera chain.
2. The development of a control system based on the concept of a digitally stored setup, resident within each camera chain; a remote control terminal giving rapid access to these memories;

and contemporary data-bus control for streamlined interconnection of multi-camera systems. This step is decisive, in that it completely eliminates potentiometer panels from each camera chain, and does not preserve them as secondary levels of control, as is the current practice for automatic cameras. Only by placing total reliance on the now-proven superiority of digital components over their analog counterparts, such as potentiometers, could we buttress the claim of superior achievements in stability, reliability, and performance and the preservation of an inherent simplicity in each camera chain.

3. The refinement of this digital control system to embody a powerful automatic capability for full technical alignment of a color camera, with absolutely no attendant complication of the camera chain itself.

The latter phase of the design program offered an exciting possibility to circumvent the dilemma formerly posed by earlier attempts at automated setup. The concept of an updating control system which is truly supervisory, and not an integral part of each camera chain, overcame the previously formidable limitations to the successful development of an automatic camera.

The TK-47 camera chain

The camera head

The camera chain itself (Fig. 1) centers about a rugged camera head designed carefully to the exacting demands of outside broadcast applications. Innovations such as a new precision yoke design, isothermal cooling techniques, shock-mounted optics coupled with a 30-mm tube system, RGB splitting optics, and a cast body supporting the large high-quality lenses, deliver superior picture quality under all dynamic scene conditions. The camera head connects via a simple lightweight flexible camera cable to a single-frame camera processor unit (CPU), which processes and finally encodes the video signals generated in the head. This unit also serves as the central system interface point for the camera chain.

The camera processor unit

The CPU contains a simple rudimentary microcomputer, contained on a single plug-in module, which continually reads

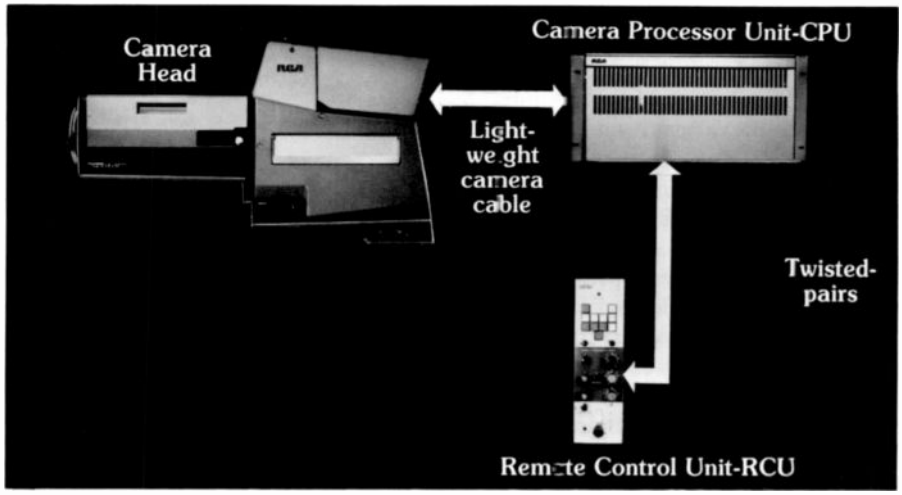


Fig. 1. The TK-47 camera achieves and maintains optimum operational conditions automatically.

the camera's central digital memory and so sustains the analog setup adjustments(Fig. 2). The microcomputer also supervises the updating of this memory during camera alignment with a setup terminal. The memory is a low-power CMOS RAM made nonvolatile by battery backup. The RCA CDP1802 8-bit microprocessor forms the nerve center of this microcomputer; its inbuilt direct memory access (DMA) is used elegantly for the continuous memory readout. One complete digital umber, corresponding to one control function in the camera, is read out during every television line at a rate synchronous with the camera's synchronizing system. This rate permits the complete sequence of camera adjustments (totaling more than one hundred) to be scanned every tv field.

A single-wire control signal is transmitted to the camera head, where it is separately demultiplexed to actuate the controls contained in the tube, scanning, and video circuits. This approach offers all the advantages of digital storage while preserving the simplicity of direct analog control, using a surprisingly small amount of simple stable hardware. Where a multiplicity of analog waveforms must be remotely controlled and assembled to produce registration and shading corrections, a custom-built analog LSI chip reduces circuitry substantially and enhances circuit reliability and temperature stability.

The TK-47 automatic system

The basic manual controlling loop of the TK-47 is formed by an operator's interpolation of the many technical parameters of the camera. It consists of using conventional remote visual inspection of two camera monitoring video outputs (one for a monochrome picture monitor and another for a separate display on a waveform monitor and vectorscope), and

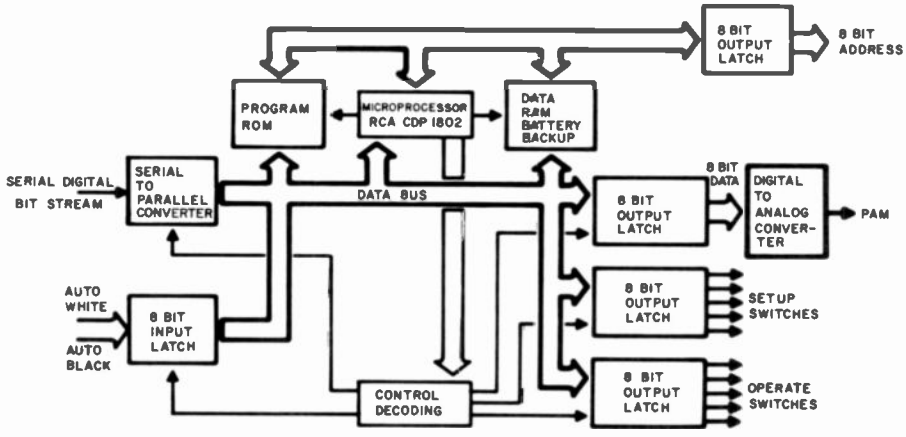


Fig. 2. This block diagram shows the simple rudimentary microcomputer used by the TK-47.

in turn sending back to the camera chain appropriate correcting or updating signals from a specially designed digital terminal mounted intimately with these monitors (Fig. 3). When the camera is satisfactorily aligned, the monitors, the control setup terminal, and the operator can work elsewhere — they're needed no longer. That is, the supervisory elements can be detached, letting the camera operate for an extended period as a stable black box. This scheme permits, for the first time, the entire camera chain to be totally controlled from a remote location (as far as 300 meters from the camera CPU). In fact, an entire system of camera chains can be supervised, monitored, and completely technically aligned from this single remote location — a tremendous advantage in time, cost, and convenience. Multi-camera systems can now have, for the first time, high technical performance maintained with precision, consistancy, and discipline.

If electronic sensors were developed which could extract from these same two video monitoring signals all absolute and comparative status information relevant to the camera's technical alignment — and, if these sensors could be followed by a decision making system which could activate the digital updating mechanism — then all the elements for an automatic control loop would be realized. This fully automatic level of control has been successfully implemented in the TK-47 Autocam (Fig. 4).

The key elements required for a reliable and precise approach to the automatic alignment of an electronic system as complex as a color tv camera can be listed as follows.

First, the system must have a carefully designed *optical source*, which can inject as much detailed information into the camera system as necessary, pertinent to the many camera parameters required to align a chain.

Second, it must have a *detector system* of high precision, reliability, and simplicity. The TK-47 uses a remarkably simple approach using only two detectors to sample every function. The total approach is based upon the reduction of all measurements to time-interval measurements, which are then performed digitally with a high degree of precision.

Third, there must be a *processing system* capable of identifying, processing, and analyzing video signal data, and then transmitting the digital correction data back to the camera for the necessary corrections.

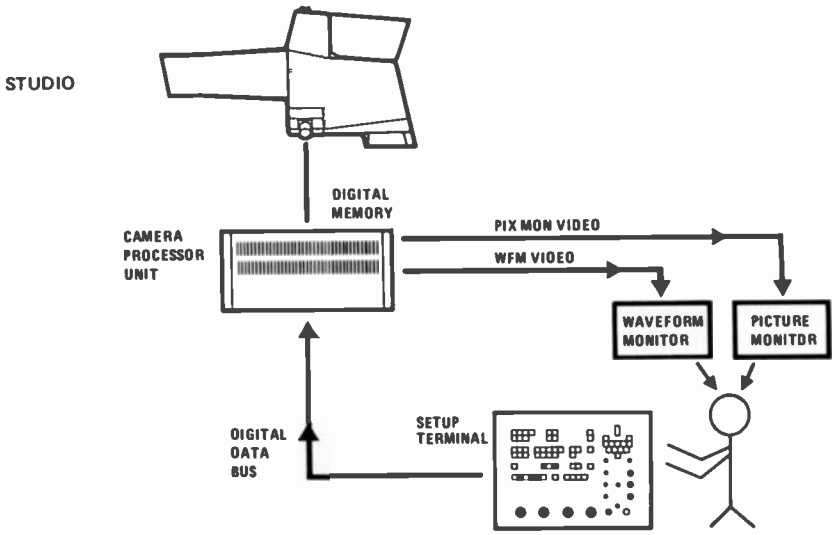


Fig. 3. Total supervision and control of the entire camera chain itself can be accomplished from a remote location which may be as far away as 900 feet from the camera processing unit.

Generating the signals

The setup terminal developed for the manual adjustment of the camera is a self-contained microcomputer with extensive capability. It can have any parameter of the camera functions examined by signalling any needed combination of video signals on the two monitoring outputs. In addition, this terminal can, under software control, dictate the sequence of signals to be examined, and allow access to the appropriate memory locations in the camera so that the camera function being examined can be adjusted. The addition of two plug-in modules to this terminal converts it to full automatic capability. One module houses the detector system, the second an address generator, which takes digital signals derived from the video information on the diascope image and then generates the data addresses directly.

Detecting and correcting

The detectors provide information that is converted into digital numbers defining absolute conditions, such as the black level of one video channel, or differential conditions, such as relative positions of the red and green images in a particular portion of the raster. After these digital numbers are temporarily stored, the software assumes a

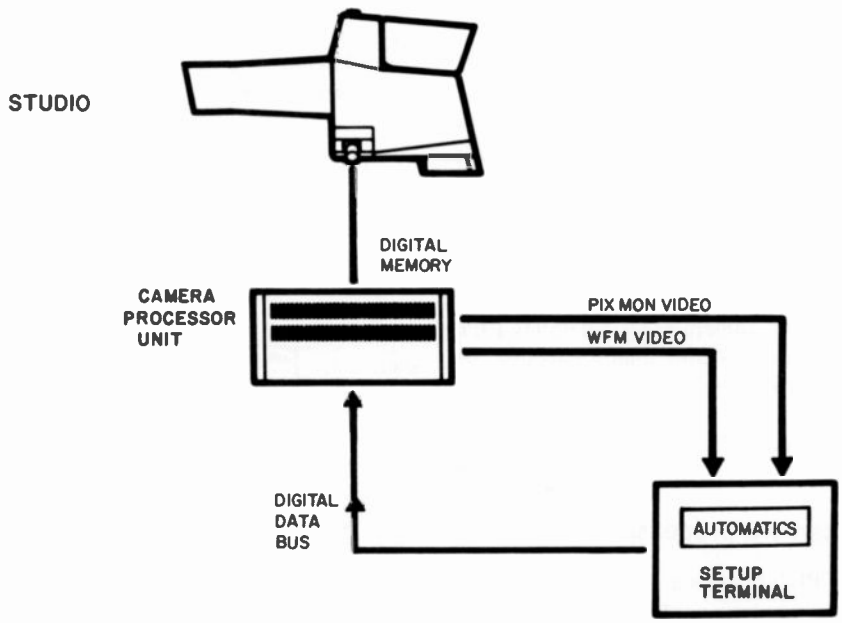


Fig. 4. A fully automatic control system has been implemented in the TK-47.

leading role. Powerful algorithms make decisions on which controls should be incremented; calculations are made when necessary; correction data is computed, synchronized, multiplexed, and transmitted to the camera memory, making the first update. The resulting alterations in the camera functions are in turn rescrutinized by the detectors and the software uses the newly presented digital numbers to compute the next appropriate adjustments. This process continues until the output data converges inside a prechosen measurement window.

Up to twenty controls are simultaneously manuevered when two images are being registered. All the algorithms employed use multiple measurements and averaging techniques to minimize errors caused by video noise or positional uncertainties.

Major realignment

The automatic alignment system has been programmed to operate in two distinct modes for practical color camera use — major realignment and daily checkout. In the first instance, the changing of major camera components, such as pickup tubes, dictates a complete realignment of the camera chain; the TK-47 system can perform this extensive alignment automatically. The procedure is initiated from the setup terminal's control panel and starts with the diascope being automatically brought into the optical path of the camera. The alignment itself is in two phases. The initial series of gross adjustments consists of reconstructing sensible images, setting the master size and aspect ratio of the green image, and making the coarse registration correction and precision setting of the beam. The final phase includes refocusing the images, registering them prcisely, entering the RGB video processing sequence, and finally performing correction and gray-scale color balance. The above alignment is obviously not a frequent requirement, but the automatic system performs this conventionally time-consuming procedure rapidly and accurately.

Daily alignment

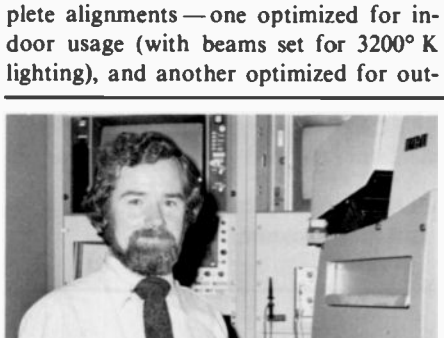
The automatic system also offers a very attractive alternative to the daily "preflight" checkout normally performed on cameras before they go on the air. The automatic daily alignment can be initiated from the vision control area via the remote operate panel or by the cameraman from the rear of the camera. First, the electrical focus is examined (and readjusted only if necessary), then the red and blue images registration is checked relative to green, followed by black and white levels, shading, and finally a gray-scale balance. Under normal conditions, this auto checkout takes only ten to fifteen seconds to scrutinize and readjust (if necessary) some 48 controls. This means a typical studio's complement of cameras can be checked out automatically with a high degree of consistency within a minute or two, effectively eliminating what is conventionally a lengthy and often stressful daily task.

Additional pre-set alignments

The digital memory in each camera chain, while physically small (two chips), has extensive storage capacity that endows each camera with some useful operational features. This memory is separated into discrete banks, each having a capacity to store a total technical alignment of the camera. Thus, the TK-47 has the unprecedented ability of containing two complete alignments — one optimized for indoor usage (with beams set for 3200° K lighting), and another optimized for outdoor lighting. Normally, these two conditions would require different beam settings, registration, etc., and so a complete readjustment. A simple recall mechanism on the video operator's panel permits instantaneous conversion of the camera to either setup. Conversion between these two setups is possible indefinitely. The camera has upward of half a dozen such storage banks, which allow further flexibility in that other alignments, for special effects, special lighting conditions, or simple emergency backup, can be stored. These other banks are controlled from the setup terminal, or from the video operator's remote control unit.

Conclusion

The TK-47 Autocam fulfills a need expressed during our 1976 worldwide market survey — a camera producing high picture performance without the troublesome technical attendance accompanying conventional color camera systems. In addition, the decisive step into microprocessor technology opens an avenue to forthcoming developments of considerable import to tv program production. In the world of broadcast, the times, indeed, "are a-changing."



Larry Thorpe began his career in 1961 at the BBC Designs Dept. designing a variety of studio broadcast equipment. He came to RCA Broadcast Systems in Camden in 1966 where, as Senior Design Engineer, he worked on the design of various camera equipment. As Unit Manager, Thorpe is responsible for the development of the TK-47 Line Camera.

Contact him at:
Broadcast Systems
Camden, N.J.
Ext. 4768

Bob Dischert joined RCA in 1954. When he first joined RCA, he was assigned to the Television Camera activity where he worked on the first production color camera of the industry, the TK-41. He then moved to the TV Terminal activity. He also made contributions to the transistorization of other TV Terminal products. He returned to the Camera activity and worked there successively on a series of five TV cameras. He was recently transferred to the David Sarnoff Laboratories where he is continuing his work on Broadcast equipment.

Contact him at:
RCA Laboratories
Princeton, N.J.
Ext. 2330

F.A. Eble|E.W. Richards, Jr.

Rapid RMA assessment—the painless plot

Here are new software tools to quickly and easily transform your conceptual models for product reliability, maintainability, and availability into publication-quality output.

Abstract: *The RECAP interactive desktop computer programs now in use at RCA/Moorestown speed reliability, maintainability, and availability engineers from preliminary modeling to final plotted output. The analysis program, with 18 mathematical models and six operating modes, handles most systems that can be represented by serially connected subsystems. The plotting program produces captioned block diagrams for these systems analyses. Tape files hold completed analyses for plotting or editing. RECAP offers versatile options and conversational displays for user convenience.*

Reliability, maintainability, and availability (RMA) engineers must perform many tasks in the analysis of systems. These tasks include allocations, modeling, trade-offs, predictions for tracking and evaluating reliability growth, and preparation of reports and proposals. In the process of performing this work, the engineers must rely on many interfaces, such as consultants, computers, typists, and illustrators.

This paper describes RMA engineering computation and plot (RECAP), a computerized tool, developed at RCA, to turn early conceptual models of reliability, maintainability, and availability into publication-quality output.

RECAP is a family of interactive BASIC language desktop computer programs including an RMA analysis program, a block diagram plotting program, and ten data storage files, all contained on one digital tape cassette. RECAP also can be stored on a high-speed disk with 50 or more data files.

Completed system analyses are transferred between the two programs and in and out of storage by means of a COMMON data instruction. Analyses may be recalled from storage by the analysis program for reprinting or editing, or by the plotting program for block diagram generation.

RECAP operates on an HP9830 calculator equipped with these accessories:

1. RAM: 16 k (analysis program), 8 k (plotting program)
2. ROMs: Advanced Programming I or II, matrix operations, string variables, plotter control
3. Printer, plotter

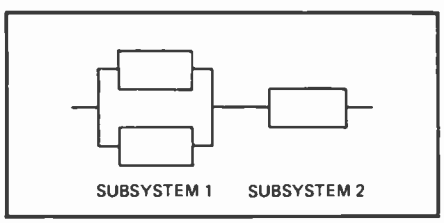


Fig. 1. RECAP can model this simple system in one step.

The analysis program

RECAP's analysis program accepts failure rates from parts (or computer program) reliability predictions, and repair times from maintainability analyses. It models most hardware (or software) systems that can be represented by successive levels of serially linked subsystems. Figure 1 illustrates a simple system that RECAP can model in one step. The more complex system, shown in Figs. 2 and 3, requires two steps.

For each subsystem, RECAP computes effective and gross failure rates, availability, mean time between failures (MTBF), and either mean time to repair (MTTR) for inherent availability calculations, or mean corrective down time (MCDT) for operational availability.

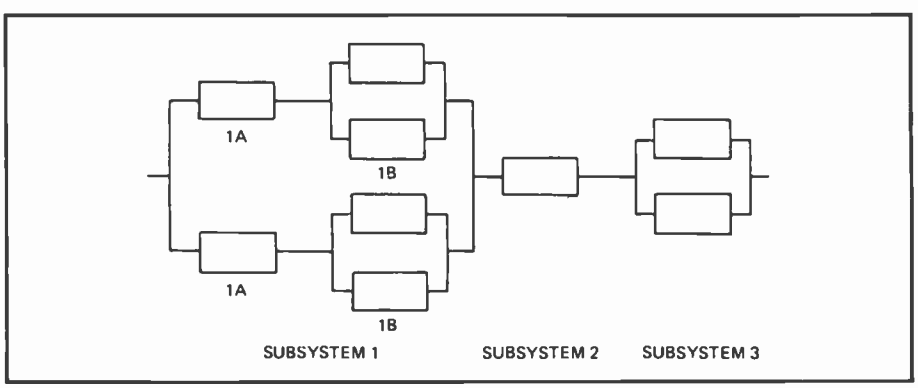


Fig. 2. This system requires two steps for modeling.

Reprinted with permission from *Proceedings* 1979 *Annual Reliability and Maintainability Symposium*, pp. 398-402, January 1979.

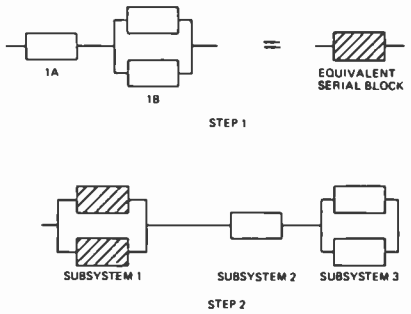


Fig. 3. Two-step RECAP analysis of complex system.

Mathematical models

The analysis program currently contains a repertory of 18 mathematical reliability models (see Table I). It offers flexibility for refinement of these models and addition of new ones as desired.

Subsystem input data not fitting one of the models in Table I will be rejected, either immediately when the improper entry is made, or by an error printout after all subsystem data has been entered. In this way the operator is protected from errors that could produce mathematically meaningless results.

Displays

The analysis program requests inputs by displaying messages on the HP9830's 32-character LED readout. Operator responses are integer codes (to specify courses of action) or input parameters. Display messages have been phrased to provide maximum information in minimum space without sacrificing readability.

The program sounds an attention-getting "BEEP" to alert the operator whenever an entry is required. An extensive checking system causes rejection of out-of-range or illogical entries. When an entry is rejected, the calculator "BEEPS" again and repeats the question.

Program operation

A streamlined flow chart of analysis program operation (Fig. 4) shows various operating modes. Refer to the illustrative RECAP printouts in Figs. 5 through 7 while reading the descriptions that follow.

"NEED TAPE INDEX (1 = Y, 0 = N)" is asked first. If 1 is entered, all ten tape files are searched and identifying data for each (name of stored system analysis, number of subsystems, MTBF, and availability) is printed.

Next question: "DELETE (-1), REPRINT (0), OTHER (1)?" If the "delete" or "reprint" option is selected, the operator must designate the tape file to be addressed. An entry of 1 alerts the program to receive new input data, beginning with a system name and selection of inherent or operational availability mode.

Input data is entered subsystem by subsystem. The mathematical model and numerical parameters for each subsystem are determined by the operator's responses to the following sequence of questions. Questions identified with an asterisk (*) are asked only when applicable.

1. Number of operating units? (Entering 0 ends the system analysis.)
2. Number of failures allowed?
3. Active or standby redundancy?*
4. Repair policy — deferred or immediate?*
5. Redundancy switch reliability?*
6. Active unit failure rate?
7. Standby unit failure rate?*
8. Unit repair time?*
9. Subsystem repair (overhaul) time?*
10. Subsystem overhaul cycle?*

For instance, once the operator has defined a serial reliability model by specifying that no failures are allowed, the only other parameters requested will be active unit failure rate and unit repair time.

Input data plus computed failure rate and availability are printed for every subsystem (Figs. 5 through 7). The current subsystem may be canceled at any time during data entry by entering −2. System results are printed when the analysis is ended.

New input data can be processed in four ways. After the data for the first subsystem of a new analysis has been entered, the display gives the operator a choice of these calculation modes:

- *Enter* 0 *for permanent analysis mode.* This mode formats the data for storage, plotting, and editing. Once permanent analysis mode has been selected, the program is locked into that mode for all subsystems until the system analysis ends.

  In this mode, a name is requested for every subsystem. Serial-model subsystems can be designated noncritical. Entering −1 for number of operating units cancels the last subsystem analyzed. A system analysis is limited to 30 subsystems in permanent analysis mode.
- *Enter* 1 *for preliminary analysis mode.* This mode provides a flexible method for trial analyses, sensitivity studies, and trade-offs. Subsystems can be added or subtracted as desired. The left column shows the number of subsystems analyzed at every step. Running totals of system failure rate and availability are printed to the right of every subsystem analysis.

  During preliminary analysis, a new mode request is displayed after data entry for each successive subsystem. Entering 1 adds the subsystem; entering −1 subtracts it and backdates system failure rate and availability.

  Systems analyzed in this mode are not retained in the calculator's memory. A preliminary analysis can be performed without disturbing the permanent analysis that may be concurrently stored in memory. There is no limit to the number of subsystems in a preliminary RECAP analysis.
- *Enter* 2 *for revision mode.* This mode updates a previous permanent analysis by replacing any designated subsystem with the subsystem that has just been entered.
- *Enter* 3 *for addition mode.* This mode updates a previous permanent analysis by

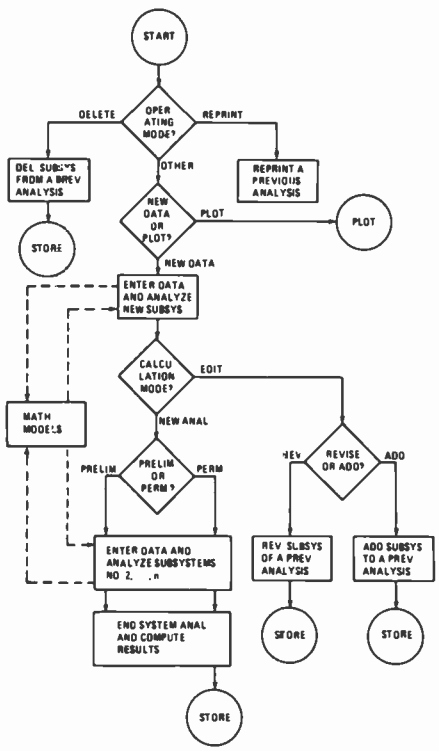


Fig. 4. Various operating models are shown in this program flow chart for the RECAP analysis program.

Reprint RE-25-3-17
Final manuscript received March 1, 1979.

adding the subsystem that has just been entered. It may be inserted at any point, or at the end.

The plotting program

RECAP's plotting program generates liberally captioned reliability block diagrams, suitable for publication, for previously analyzed systems. Analysis data can be transferred directly between the programs or stored in a tape file for future plotting.

Displays

The conversational display format of the analysis program is repeated here, although the operator-machine interface is less demanding. After a few initial inputs, the operator's only tasks are changing the paper and initiating the next page as each plotted page is completed.

Program operation

Before plotting begins, the operator is given an opportunity to change the name of the system to be plotted, and to store the renamed system.

Block diagrams are plotted five subsystems to a page (see Fig. 8). Most of the input and output data contained in the analysis printout is duplicated on the plotted page, making the plot self-sufficient for formal proposals and reports.

Six basic subsystem formats are provided, representing six classes of models:

1. Serial
2. Serial, noncritical
3. Active redundancy, two active units
4. Active redundancy, three or more active units
5. Standby redundancy, two units
6. Standby redundancy, three or more units

Up to six pages of block diagrams, representing as many as 30 subsystems, can be plotted to document a system analysis. A title page summarizing the results (MTBF, MTTR or MCDT, availability) is produced to complete the package.

Sample analysis

This example traces a hypothetical radar system through four phases of analysis and plotting. Our system consists of (1) 100-element phased array antenna, (2) twin transmitter modules, (3) dual-channel receiver, and (4) signal processor with serial and parallel portions. Each subsystem might in turn represent a lower level system analysis. Similarly, the radar system could be a subsystem in a higher level system analysis.

Phase 1: Preliminary analysis with design changes to meet an allocation. An

```
                ***RECAP=RMA ENGINEERING COMPUTATION AND PLOT***

                      *****SYSTEM RMA ANALYSIS*****

                                TIMES ARE IN HRS
                      FAILURE RATES ARE IN FAIL/MILL HRS

                      CODE  REDUNDANCY   REPAIR

                      0     NONE         OVERHAUL CYCLE
                      1     ACTIVE       IMMED SGL UNIT
                      2     STANDBY      IMMED MULT UNITS

SAMPLE RADAR SYSTEM

    *INHERENT AVAILABILITY*

 SUB  NO OF ALLW  CODES   ACT/SBY  MTIME/ OHAUL      SUBSYS      TOTAL
 SYS  UNITS FAIL RED REP  UNIT FR  MTTR   CYCLE      FR AVL      FR/AVL

   1   100   10   1   0    18.00   4.00   4380       33.46       33.46
                                    4.00              0.9999      0.9999
                                                   R= 0.8637

   2     2    1   1   1  4500.00   3.50      0       137.42      170.88
                                    3.50              0.9995      0.9994
 Receiver Subsystem
   3     2    0   0   1   212.60   2.25      0       425.20      596.08
                                    2.25              0.9990      0.9984

   4     1    0   0   1   800.00   1.63      0       800.00     1396.08
                                    1.63              0.9987      0.9971
                                                 Exceeds allocation
   5     2    1   1   2   560.00   1.48      0         0.93     (1397.0)
                                    0.74              1.0000      0.9971
 Delete Receiver
   4     2    0   0   1   212.60   2.25      0      -425.20      971.81
                                    2.25             -0.9990      0.9981
 New Receiver                                    Meets allocation
   5     2    1   2   0   232.60   2.25     72         9.87     (981.68)
       SW REL= 0.9950       0.00   2.25              1.0000      0.9981
                                                   R= 0.9993

                                    SYSTEM FAILURE RATE =       981.68
                                    GROSS FAILURE RATE =      13185.20

 SYS MTBF =    1018.66   MTTR =   1.96

                                    SYSTEM AVAIL =              0.9981
```

Fig. 5. Preliminary analysis printout.

```
SAMPLE RADAR SYSTEM

    *INHERENT AVAILABILITY*

 SUB  NO OF ALLW  CODES   ACT/SBY  MTIME  OHAUL      SUBSYS      TOTAL
 SYS  UNITS FAIL RED REP  UNIT FR  MTTR   CYCLE      FR AVL      FR/AVL

ANT ELEMENTS
  -1   100   10   1   0    18.00   4.00   4380       33.46
                                    4.00              0.9999
                                                   R= 0.8637

TRANSMITTERS
  -2     2    1   1   1  4500.00   3.50      0       137.42
                                    3.50              0.9995

RECEIVER
  -3     2    1   2   0   232.60   2.25     72         9.87
       SW REL= 0.9950       0.00   2.25              1.0000
                                                   R= 0.9993

SIG PROC (SER)
  -4     1    0   0   1   800.00   1.63      0       800.00
                                    1.63              0.9987

SIG PROC (RED)
  -5     2    1   1   2   560.00   1.48      0         0.93
                                    0.74              1.0000

                                    SYSTEM FAILURE RATE =       981.68
                                    GROSS FAILURE RATE =      13185.20

 SYS MTBF =    1018.66   MTTR =   1.98

                                    SYSTEM AVAIL =              0.9981

  DATA STORED IN FILE  5
```

Fig. 6. Permanent analysis printout.

allocated failure rate of 1000 failures per million hours (FPMH) must be met. The trial analysis shown in Fig. 5 initially predicts 1397 FPMH. A new receiver configuration employing a standby unit is substituted, and the requirement is met; receiver unit failure rate increases slightly due to fault-sensing circuitry needed to trigger redundancy switching. In practice, the decision to add subsystem redundancy would involve technical and cost tradeoffs.

Phase 2: Permanent analysis, stored for plotting. The preliminary printout in Fig. 5 is untidy, does not name the subsystems, and was not retained in memory. Fig. 6 shows the permanent analysis of the system configuration which evolved in Phase 1. The printout confirms storage in tape file 5.

Table I. RECAP mathematical models.

Model	Oper. Units	Allow. Failures	Redundancy	Repair	Simul. Unit Repair Capability	λ_{eff} = Subsystem Failure Rate
1	M	0	None	Immed.	Single	$M\lambda$
2	M	M	None	Immed.	Single	0
3	M	N	Active	Defer	N/A	$-\frac{\ln R(T)}{T}$ where $R(T) = \sum_{k=0}^{N} \frac{M!}{k!(M-k)!} e^{-(M-k)\lambda T} (1-e^{-\lambda T})^k$
4	2	1	Active	Immed.	Single	$2\lambda^2/(\mu + 2\lambda)$
5	3	1	Active	Immed.	Single	$6\lambda^2/(\mu + 3\lambda)$
6	3	2	Active	Immed.	Single	$\mu(1-A)/A$ where $A = \frac{\mu^3 + 3\mu^2\lambda + 6\mu\lambda^2}{\mu^3 + 3\mu^2\lambda + 6\mu\lambda^2 + 6\lambda^3}$
7	M	N	Active	Immed.	Single	$\mu(1-A)/A$ where $A = 1 - \left(\frac{M\lambda}{\mu}\right)^{N+1}$
8	2	1	Active	Immed.	Multiple	$2\lambda^2/(\mu + 2\lambda)$
9	3	1	Active	Immed.	Multiple	$6\lambda^2/(\mu + 3\lambda)$
10	3	2	Active	Immed.	Multiple	$3\mu(1-A)/A$ where $A = \frac{\mu^3 + 3\mu^2\lambda + 3\mu\lambda^2}{\mu^3 + 3\mu^2\lambda + 3\mu\lambda^2 + \lambda^3}$
11	M	N	Active	Immed.	Multiple	$(N+1)\mu(1-A)/A$ where $A = e^{-\frac{M\lambda}{\mu}} \sum_{k=0}^{N} \frac{\left(\frac{M\lambda}{\mu}\right)^k}{k!}$
12	M	N	Standby	Defer	N/A	$-\frac{\ln R(T)}{T}$ where $R(T) = e^{-M\lambda T} \sum_{k=0}^{N} \frac{(R_s M\lambda T)^k}{k!}$ or (see Note 2) $R(T) = e^{-M\lambda T} \sum_{k=0}^{N} \frac{(1-e^{-\lambda_s T})^k \Gamma(\frac{M\lambda}{\lambda_s}+k)}{\Gamma(k+1)\Gamma(\frac{M\lambda}{\lambda_s})}$
13	1	1	Standby	Immed.	Single	$\lambda^2/(\mu + \lambda)$
14	2	1	Standby	Immed.	Single	$4\lambda^2/(\mu + 2\lambda)$
15	1	2	Standby	Immed.	Single	$\mu(1-A)/A$ where $A = \frac{\mu^3 + \mu^2\lambda + \mu\lambda^2}{\mu^3 + \mu^2\lambda + \mu\lambda^2 + \lambda^3}$
16	1	1	Standby	Immed.	Multiple	$\lambda^2/(\mu + \lambda)$
17	2	1	Standby	Immed.	Multiple	$4\lambda^2/(\mu + 2\lambda)$
18	1	2	Standby	Immed.	Multiple	$3\mu(1-A)/A$ where $A = \frac{6\mu^3 + 6\mu^2\lambda + 3\mu\lambda^2}{6\mu^3 + 6\mu^2\lambda + 3\mu\lambda^2 + \lambda^3}$

λ = Active Unit Failure Rate
λ_s = Standby Unit Failure Rate
μ = Unit Repair Rate
T = Overhaul Cycle
R_s = Standby Switch Reliability

NOTES:
1. Immediate Repair Models – see Acknowledgment
2. Formula for Standby Redundancy with Standby Unit Failure Rate – see Reference
3. Models 3, 7, 11, and 12 involve approximations
4. Large T values for Models 3 and 12 may cause inaccuracies in system results when serially combining subsystems

```
SAMPLE RADAR SYSTEM (REVISED)
   (INHERENT AVAILABILITY)

 NUM NO OF ALLW MODES   ACT/SBY  MTIME  OHAUL     SUBSYS      TOTAL
 SYS UNITS FAIL RED REP UNIT FR  MTTR   CYCLE     FR AVL      FR AVL

UPDATE:          SAMPLE RADAR SYSTEM
                                                FROM FILE  5

REVI  ON TO     TRANSMITTERS
 -2     2    1   1   1   4500.00    3.50      0        137.42
                                    3.50               0.9995

TRANSMITTERS
 -2     2    1   1   1   3000.00    3.50      0        91.70
                                    3.50               0.9992

  NO. OF SUBSYSTEMS =  5

                                   SYSTEM FAILURE RATE -       905.96
                                   GROSS FAILURE RATE =       10185.20

   SYS MTBF =     110...0    MTTR =    1.85
                                   SYSTEM AVAIL =               0.9983

  DATA STORED IN FILE  5
```

Fig. 7. Revision mode printout.

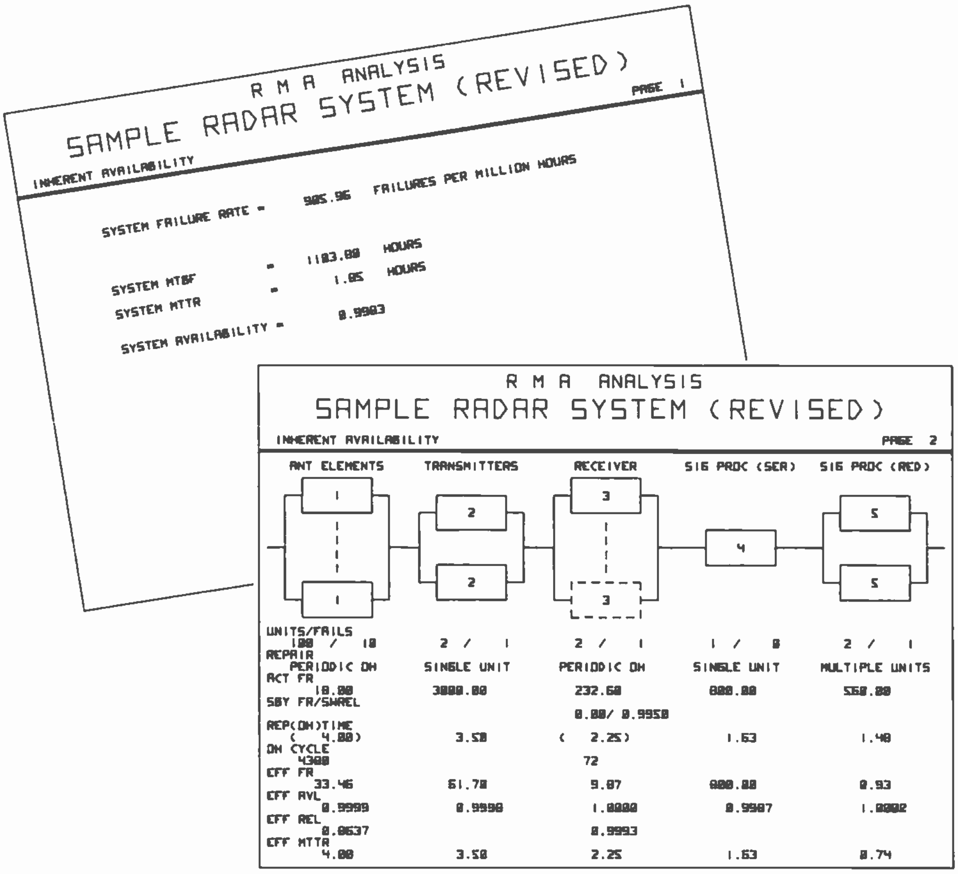


Fig. 8. Block diagrams are plotted five subsystems to a page.

Phase 3: Update to reflect reliability growth. Development of a higher reliability transmitting tube reduces transmitter module failure rate. This change is reflected in the revision documented in Fig. 7. The RECAP analysis program prints only the edited subsystem plus the new system totals when operating in an update mode; however, the entire analysis can be reprinted at any time.

Phase 4: Plotting of updated analysis for publication. The plotting program goes to work; its output is reproduced in Fig. 8. System and subsystem names are centered automatically.

Looking ahead

RECAP is adaptable to execution on the later generation HP9845 computer system, using the integral printer-plotter to produce block diagrams. Models can be added to expand its versatility. A space-saving plot option — ten subsystems per page with minimal printed area — may be incorporated in the plotting program.

The authors hope that this article and the mini-analysis above have communicated the central message of RECAP — computers are not forbidding and remote; they can be as personal, as productive, and as painless as you wish to make them in your everyday work.

Acknowledgment

Equations for immediate-repair models in Table I were provided by William J. O'Leary of RCA/MSR.

Reference

1. Thompson, R.J., "A Generalized Expression for the Reliability of a System Utilizing Standby Redundance," Aerospace Corporation Report TOR-1001(2303)-1 (Aug. 5, 1966).

Frank Eble joined RCA in 1960. He is currently responsible for logistics engineering of MSR-designed equipment for the AEGIS Weapon System.

Contact him at:
Missile & Surface Radar
Moorestown, N.J.
Ext. 2555

Earl Richards is currently working in the area of reliability prediction in MSR's Technical Assurance activity. He joined RCA in 1959.

Contact him at:
Missile & Surface Radar
Moorestown, N.J.
Ext. 3768

Eble is seated at the terminal while Richards watches as data is being input for plotting.

Dates and Deadlines

Upcoming meetings

Ed. Note: Meetings are listed chronologically. Listed after the meeting title (in bold type) are the sponsor(s), the location, and the person to contact for more information.

DEC 10-15, 1979—**Infrared and Near Millimeter Waves** (MTT) Americana of Bal Harbour, Miami Beach, FL **Prog Info:** K.J. Button, MIT National Magnet Lab., Cambridge, MA 02139 (617-253-5561)

DEC 12, 1979—**Computer Networking Symposium** (C, NBS) Gaithersburg, MD **Prog Info:** Harry Hayman, P.O. Box 639, Silver Spring, MD 20901 (301-439-7007)

DEC 12-14, 1979—**18th Conference on Decision and Control** (CS) Galt Ocean Mile Hotel, Fort Lauderdale, FL **Prog Info:** Prof. Stephen Kahne, Dept. of Systems Engineering, Case Institute of Technology, Case Western Reserve Univ., Cleveland, OH 44106 (216-368-4076)

JAN 7-9, 1980—**Pacific Telecommunications Conference** (COM) Ilikai Hotel, Honolulu, HI **Prog Info:** Richard J. Barber, 2424 Maile Way, #704, Honolulu, HI 96822 (808-948-7879)

JAN 7-10, 1980—**Photovoltaic Specialists Conference** (ED) Town & Country Hotel, San Diego, CA **Prog Info:** Prof. C. Backus, Arizona State Univ., Coll. of Engr. & Applied Science, Tempe, AZ 85281 (602-965-3857)

JAN 22-24, 1980—**Reliability & Maintainability Symposium** (R) San Francisco Hilton and Tower, San Francisco, CA **Prog Info:** A.M. Smith, General Electric Co., Fast Breeder Reactor Dept., 310 De Guigne Dr., M/C/S-65, Sunnyvale, CA 94086 (408-297-3000, Ext. 627)

JAN 28-30, 1980—**Integrated & Guided Wave Optics** (QEA, OSA) Hyatt Lake Tahoe, NV **Prog Info:** Joan Connor, Optical Society of America, 2000 L St., NW, Suite 620, Washington DC 20036 (202-293-1420)

JAN 29-31, 1980—**Aerospace and Electronic Systems Winter Convention (WINCON)** (AES, Los Angeles Council) Sheraton Universal Hotel, N. Hollywood, CA **Prog Info:** Phillip G. Halamandaris, Gould Inc.,—NAVCOM Systems Div., 4323 Arden Drive, El Monte, CA 91731 (213-442-0123, Ext. 602)

FEB 13-15, 1980—**Intl. Solid State Circuits Conference** (SSC, San Francisco Sec.) Hilton Hotel, San Francisco, CA **Prog Info:** Lewis Winner, 301 Almeria Ave., Coral Gables, FL 33134 (305-446-8193)

FEB 25-28, 1980—**COMPCON Spring '80** (C) Jack Tar Hotel, San Francisco, CA **Prog Info:** Harry Hayman, P.O. Box 639, Silver Spring, MD 20901 (301-439-7007)

FEB 26-28, 1980—**Laser and Electro-Optical Systems/Inertial Confinement Fusion** (QEA, OSA) Town & Country Hotel, San Diego, CA **Prog Info:** Joan Connor, Optical Society of America, 2000 L Street N.W. (Suite 620), Washington, DC 20036 (202-293-1420)

MAR 3-5, 1980—**NCC Office Automation Congress** (C) **Prog Info:** Harry Hayman, P.O. Box 639, Silver Spring, MD 20901 (301-439-7007)

MAR 4-6, 1980—**International Zurich Seminar on Digital Communications** (Switzerland Sec. COM (Cooperating)) Swiss Federal Institute of Tech. on Digital Communications, Zurich, Switzerland **Prog Info:** Prof. P.E. Leuthold, Eidgenossische Technische Hochschule Zurich Institut Fur Hochfrequenztechnik, Sternwartstrasse 7, Zurich, Switzerland (Tele.: T41-1-326211)

MAR 11-14, 1980—**Computer Architecture for Non-numeric Processing 5th Workshop** (C) Pacific Grove, CA **Prog Info:** Harry Hayman, P.O. Box 639, Silver Spring, MD 20901 (301-439-7007)

MAR 16-18, 1980—**Particle Accelerator Conference** (NPS) **Prog Info:** Dr. Louis Castrell, NPS Meetings Coordinator, National Bureau of Standards, C333 Radiation Physics, Washington, DC 20234

MAR 17-20, 1980—**Industrial Control & Instrumentation Applications of Mini & Microcomputers** (IECI) Sponsors: IECI, Sheraton Hotel, JFK Blvd., Phila., PA **Prog Info:** Patrick P. Fasang, RCA Corp., Route 38, Cherry Hill, NJ 08358 (609-338-5020)

MAR 24-25, 1980—**Radio Transmitters and Modulation Techniques** (IEE, IERE) IEE, Savoy Place, London, WC2 **Prog Info:** Conference Dept., IEE, Savoy Place, London WC2R OBL, England

MAR 24-27, 1980—**Magnetic Fluids 2nd Intl. Conf.** (MAG) Marriott Inn, Orlando, FL **Prog Info:** Markus Zahn, Dept of EE, Univ. of Florida, Gainesville, FL 32611 (904-392-4964 Ofc., 904-392-4960 Sect.)

APR 7-11, 1980—**Optical Computing Int'l. Conference** (C) Hyatt Regency, Washington, DC **Prog Info:** Sam Horvitz, P.O. Box 274, Waterford, CT 06385 (Office: 203-447-4270, Home: 203-442-0829)

APR 8-10, 1980—**Reliability Physics Symposium** (R, ED) Caesar's Palace, Las Vegas, NV **Prog Info:** Glen T. Cheney, Bell Laboratories, 555 Union Blvd., Allentown, PA 18103 (215-439-7628)

APR 9-11, 1980—**Intl. Conf. on Acoustics, Speech and Signal Processing** (ASSP, IEEE), Fairmount Hotel, Denver, CO **Prog Info:** J. Robert Ashley, Univ. of Colorado, Coll. of Engr. & Appl. Sci., Dept. of Elec. & Comp. Engr., 1100 14th Street, Denver, CO 80202 (303-629-2554 or 2872)

APR 13-16, 1980—**Southeastcon '80,** Opryland Hotel, Nashville, TN **Prog Info:** Larry K. Wilson, Box 1687, Station B, Nashville, TN 37235 (615) 322-2771)

APR 21-23, 1980—**American Power Conference** Ill. Inst. Tech., PES & 8 other engr. societies, Palmer House, Chicago, IL **Prog Info:** R.A. Budenholzer, 246 E-1, IIT, Chicago, IL 60616 (312-567-3196)

APR 21-24, 1980—**Intl Magnetics Conf.** (INTERMAG) (MAG), Boston Sheraton Hotel, Boston, MA **Prog Info:** D.I. Gordon, Naval Surface Weapons Center, White Oak Lab., Silver Spring, MD 20910 (202-394-2167)

APR 28-30, 1980—**Internation Radar Conference** (IEEE, AES, IEEE Wash. section), Stouffer's National Center Hotel, Arlington, VA **Prog Info:** R.T. Hill/J. Kalitta, c/o Conference Office, 777 14th St., NW Suite 917, Washington, DC 20005 (202-637-4217)

APR 28-30, 1980—**30th Electronic Components Conference** (CHMT, EIA) Hyatt Regency San Francisco, San Francisco, CA **Prog Info:** Dr. H.J. Gisler, Electro Scientific Industries, 13900 N.W. Science Park Dr., Portland, OR 97229 (503-641-4141)

APR 28-30, 1980—**Circuits and Systems Int'l. Symposium** (CAS) Shamrock Hilton Hotel, Houston, TX **Prog Info:** Prof. R.J.P. DeFigueiredo, General Chairman, Dept. of Electrical Engineering, Rice University, P.O. Box 1892, Houston, TX 77001 (713-527-8101, ext. 3568)

MAY 12-15, 1980—**Industrial and Commercial Power Systems Conference** (IA, IEEE, Houston Section), Stouffer's Greenway Plaza, Houston, TX **Prog Info:** Baldwin Bridger, Powell Elect. Mfg. Co., P.O. Box 12818, Houston, TX 77017 (713-944-6900)

MAY 13-15, 1980—**Electro** (IEEE sponsors) Reg 1, New Eng. Council, METSAC, (ERA, New Eng. & N.Y. chapters), Boston-Sheraton Hynes Auditorium, Boston, MA **Prog Info:** Dale Litherland, Electronic Con-

ventions, Inc., 999 N. Sepulveda Blvd., El Segundo, CA 90245 (213-772-2965)

MAY 19-20, 1980—**Southeast Symp. on System Theory** (C), Old Dominion University, Cavalier Hotel, Virginia Beach, VA **Prog Info:** Harry Hayman, P.O. Box 639, Silver Spring, MD 20901 (301-439-7007)

MAY 28-30, 1980—**Intl. Microwave Symp.** (MTT) Shoreham Americana Hotel, Washington, D.C. **Prog. Info:** Lawrence R. Whicker, Naval Research Lab Code 5250, Washington, DC 20375 (202-767-3312)

JUNE 8-11, 1980—**Intl. Conference on Communications,** Red Lion Inn, Seattle, WA **Prog Info:** W.W. Keltner, Room 1402, 1600 Bell Plaza, Seattle, WA 98191 (206-345-3999) & (206-655-3601)

JUNE 9-11, 1980—**Int'l. Symposium on Electrical Insulation** (IEEE) (EI) 57 Park Plaza Hotel, Boston, MA **Prog Info:** Dr. H. St. Onge, IREQ-Hydro Quebec Institute of Research, P.O. Box 1000, Varennes, PQ, Canada, JOL 2 PO (514-652-8420)

Calls for papers

Ed. Note: Calls are listed chronologically by meeting date. Listed after the meeting (in bold type) are the sponsor(s), the location, and deadline information for submittals.

JAN 7-10, 1980—**14th IEEE Photovoltaic Specialists Conf.,** San Diego, CA **Deadline Info:** 7/15/79 300-word abstract to: Charles E. Backus, College of Engineering and Applied Sciences, Arizona State University, Tempe AZ 85281

Pen and Podium

Recent RCA technical papers and presentations

To obtain copies of papers, check your library or contact the author or his divisional Technical Publications Administrator (listed on back cover) for a reprint.

Americom

J. Christopher|J.E. Keigler
The RCA Americom satellite communications system—30th International Astronautical Congress, Munich, Germany (9/17-22/79)

Astro-Electronics

J. Blankenship|Col. J. McGlinchey
Evolution of the Block 5 spacecraft to meet changing military needs—Military Electronics Defense Exp (MEDE) 1979, Wiesbaden, West Germany (9/25-27/79)

F. Chu
A turbidity current model—Oceans IV Conference, San Francisco, Ca. (9/13/79)

F. Drago|H. Hawley|H. Huang|B. Dornan
C-Band FET power amplifier for TWTA replacement—30th International Astronautical Congress, Munich, Germany (9/79)

J. Keigler|J. Christopher
The RCA Americom satellite communications system—30th International Astronautical Congress, Munich, Germany (9/79)

K. Phillips|G. Schmidt
Post launch design & implementation of a S/C momentum bias attitude control system—AIAA Guidance & Control Conference, Palo Alto, Calif. (8/6/79)

Automated Systems

M.J. Cantella|B. Capone (RADC)
W. Ewing(RADC)|J.J. Klein
L. Skolnick (RADC)|R. Taylor (RADC)
A TV compatible Schottky barrier monolithic IRCCD focal plane—SPIE 23rd International Technical Symposium & Instrumentation Display, San Diego, Calif. (8/79)

R.T. Cowley|R.E. Hanson
New generation of combat vehicle field test systems—Military Electronics Defence Expo '79, Wiesbaden, Germany (9/79)

S.H. Eames|E.L. Naas
Automation of decision making logistic support—14th International Logistics Symposium, Clearwater, Fla. (8/79)

A. Eiranova|J.E. Fay
Can software portability alleviate proliferation headaches?—AUTOTESTCON '79, Minneapolis, Minn. (9/20/79)

R.E. Hanson|H.L. Resnick
STE-T, a forward area automated test systems for combat vehicles—AUTOTESTCON '79, Minneapolis, Minn. (9/79)

D.M. Kulig|R.C. Plaisted|T.J. Plunkett
Equipment specification for ATE testability—AUTOTESTCON '79, Minneapolis, Minn. (9/19/79)

N.L. Laschever
Where we stand—Editorial for *IEEE Reflector* (9/79)

F.F. Martin
The GVS-5, a versatile handheld laser rangefinder for tactical use—Military Electronics Defence Expo '79, Wiesbaden, Germany (9/25/79)

J.P. Mergler
A natural command language for C³ applications—Military Electronics Defence Expo '79, Wiesbaden, Germany (9/25/79)

J.F. Weiss
Fault isolation speed-up for LSI boards—IEEE Test Conference '79, Cherry Hill, N.J. (10/79)

Broadcast Systems

S.L. Bendell|C.A. Johnson
Matching a new tube to the TK-47 camera—SMPTE (10/22/79)

R.N. Hurst
Automatic television camera with microprocessor control—IEEE (9/20/79)

W.S. Sepich
EDICS—Engineering Drawing Information and Control System—American Inst. of Ind. Engineers (9/27/79)

R.M. Unetich
New opportunities in television transmitter design—IEEE Fall Symposium, Washington, D.C. (9/21/79)

Government Communications Systems

D. Hampel
Application of VLSI to smart sensors—MEDE, Wiesbaden, Germany, *Proceedings* (9/26/79)

A. Kaplan|D. Sherwood
MASS: A modular ESM signal processor—MEDE, Wiesbaden, Germany; *Proceedings* (9/9-12/79)

D. Sheby
On the tracked vs. wheeled vehicle discrimination problem: a vehicle-ground coupling feature—Sensor Technology Symposium (US Army Waterways Experimental Station, Vicksburg, Miss.) *Proceedings* (9/25-27/79)

D.B. Wolfe
Natural frequencies and mode shapes of multi-degrees of freedom systems on a programmable calculator—Vibrations Meetings '79 ASME Design Conference, *Proceedings* (9/9-12/79)

Laboratories

D. Botez|M. Ettenberg
Comparison of surface- and edge-emitting LEDs for use in fiber-optical communications—*IEEE Transactions on Electron Devices*, Vol. ED-26, No. 8 (8/79)

R.S. Crandall|R. Williams|B.E. Tompkins
Collection efficiency measurements on a-Si:H solar cells—*J. Appl. Phys.*, 50(8) (8/79)

D.A. de Wolf
Depolarization term of the wave equation—*J. Opt. Soc. Am.*, Vol. 69, No. 9 (9/79)

S. Freeman
Optimum fault isolation by statistical inference—*IEEE Transactions on Circuits and Systems*, Vol. CAS-26, No. 7 (7/79)

J.J. Hanak
Monolithic solar cell panel of amorphous silicon—*Solar Energy*, Vol. 23, pp. 145-147 (1979)

H. Kiess
Electrophotographic processes in ZnO layers—*Progress in Surface Science*, Vol. 9, pp. 113-142 (1979)

H.P. Kleinknecht
Diffraction gratings as keys for automatic alignment in proximity and projection printing—*SPIE Developments in Semiconductor Microlithography IV*,Vol. 174 (1979)

K. Knop|H.W. Lehmann|R. Widmer
Microfabrication and evaluation of diffractive optical filters prepared by reactive sputter etching—*J. Appl. Phys.* 50(6) (6/79)

H. Kressel
Semiconductor light sources for fiber optical communication—*Fiber Optics: Advances in Research and Development* (1979)

G.H. Olsen|C.J. Nuese|M. Ettenberg
Reliability of vapor-grown InGaAs and InGaAsP heterojunction laser structures—*IEEE J. of Quantum Electronics*, Vol. QE-15, No. 8 (8/79)

J.I. Pankove
Field-indured population inversion of distant donor-acceptor pairs—*IEEE Transactions on Electron Devices*, Vol. ED-26, No. 8 (8/79)

J.I. Pankove|W.V. Hough
Optical properties of boron hydride BH_x—*J. Appl. Phys.* 50(9) (9/79)

R.S. Stepleman
Monotone convergence and effective stopping criteria for numerical processes—*BIT*, pp. 278-281 (1979) (Sweden)

Missile and Surface Radar

S. Abbott|C.T. Schilsky
The interface between reliability and systems design—RCA Reliability and Quality Symposium, RCA Princeton Labs (10/18-19/79)

K. Abend|J.R. Platt
Extrapolating bandlimited signals with noise and quantization—RADC Spectrum Estimation Workshop, Griffith AFB, N.Y. (10/3-5/79) *Proceedings*

B.F. Bogner
Vehicular traffic radar—IEEE Phila. Section, Phila., Pa. (9/25/79)

F.J. Buckley
Management of RT programming—Lecturer; 2-day continuing professional education seminar sponsored by Drexel University, Phila., Pa. (10/9-10/79); San Francisco, Calif. (10/17-18/79); Phila., Pa. (10/23-25/79)

F.J. Buckley
A standard for software quality assurance plans—*Computer*, Vol. 12, No. 8, pp. 43-50 (8/79)

F.J. Buckley
Software quality assurance standards—COMPCON Fall '79, Microprocessor Technical Standards Panel (Session 10), Washington, D.C. (9/5/79)

L.L. Coulter
Automated test system for phased array antenna—Tri-Service Manufacturing Technology Conference, Phoenix, Ariz., *Proceedings* (10/22-25/79)

R.C. Durham
EMC test procedure for ABC system operations room upgrade equipment—*Communicating Professional Ideas*, M.D. Morris writing-course booklet

I.E. Goldstein
Introduction; Topside design, other; Combat system integration; Computer/digital technology; Room arrangements—Summer Course: Combat Systems Engineering and Ship Design, MIT, Cambridge, Mass. (8/79)

W.A. Harmening
Implementing a near field antenna test facility—*Microwave Journal*, Vol. 22, No. 9, pp. 44-55 (9/79)

P.R. Kalata
Linear prediction, filtering, and smoothing: an information theoretic approach—*International Journal of Information Sciences*, Vol. 17, pp. 1-14 (2/79)

R.E. Killion
Achieving reliability in unattended systems—RCA Reliability and Quality Symposium, RCA Princeton Labs (10/18-19/79)

E.G. Lurcott
System functional analysis and functional allocation—Summer Course: Combat Systems Engineering and Ship Design, MIT, Cambridge, Mass. (8/79)

F.E. Oliveto
The role of science and technology in the future—U.S. Association of the Club of Rome, Arlington, Va. (10/8-13/79)

M.H. Plofker
System Disciplines—Summer Course: Combat Systems Engineering and Ship Design, MIT, Cambridge, Mass. (8-79)

R.J. Renfrow
Topside design, radar; Topside design, weapons—Summer Course: Combat Systems Engineering and Ship Design, MIT, Cambridge, Mass. (8/79)

E.E. Roberts
Error budgeting; Alignment and ship's flexure—Summer Course: Combat Systems Engineering and Ship Design, MIT, Cambridge, Mass. (8/79)

G.J. Rogers
Support systems, mechanical—Summer Course: Combat Systems Engineering and Ship Design, MIT, Cambridge, Mass. (8/79)

R.M. Scudder|L.H. Yorinks
Low sidelobe reflector antenna for tactical radars—Antenna Applications Symposium, Univ. of Illinois, *Digest* (9/26-28/79)

D. Shore
Survivable C^3—Military Electronics Defence Expo '79, Wiesbaden, W. Germany, *Proceedings* (9/25-27/79)

D. Staiman
Automated near field antenna test set for phased array production—Antenna Applications Symposium, Univ. of Illinois, *Digest* (9/26-28/79)

D. Staiman
Automated near-field antenna test set for phased array production—Antenna Measurements Symposium, Atlanta, Ga., *Digest* (10/17-18/79)

S.A. Steele
Software modularity and its impact on flexibility in military systems—Military Electronics Defence Expo '79, Wiesbaden, W. Germany, *Proceedings* (9/25-27/79)

J.T. Threston
Combat systems performance tradeoffs: analysis; detection; control; missile; AEGIS—Summer Course: Combat Systems Engineering and Ship Design, MIT, Cambridge, Mass. (8/79)

R.B. Webb
Support systems, electrical—Summer Course: Combat Systems Engineering and Ship Design, MIT, Cambridge, Mass. (8/79)

Patents

Commercial Communications Systems

Dischert, R.A.|Bendell, S.L.
Video image highlight suppression circuit with delayed compensation—4166281

Dischert, R.A.|Thorpe, L.J.
Setup control unit for television cameras—4167022

Katagi, K.
Reduction of target shift in coordinate converter—4164739

Consumer Electronics

Steckler, S.A.|Balaban, A.R.
Tuning system including a memory for storing tuning information with user controls arranged to facilitate its programming—4164711

Stewart, M.C.
Locking mechanism for record package—4164782

Peer, J.C.|Dietz, W.F.|Gries, R.J.|Nero, L.W.
Overvoltage protected de-boost regulator—4169241

Willis, D.H.
Switching regulator for a television apparatus—4163926

Willis, D.H.
Automatic peak beam current limiter—4167025

Government Communications Systems

Bessette, O.E.|Griffin, J.S.
Rotating head recorder with different recording and playback speeds—4167023

Corsover, S.L.
Film guide for optical scanners—4168506

Laboratories

Carlson, D.E.|Wronski, C.R.
Schottky barrier amorphous silicon solar cell with thin doped region adjacent metal Schottky barrier—4163677

Carlson, D.E.
Amorphous silicon solar cell allowing infrared transmission—4166919 (assigned to U.S. government)

Catanese, C.A.
Image display device with ion feedback control and method of operating the same—4164681

Curtice, W.R.
Time interval measurement—4165459

Curtice, W.R.
Threshold gate—4166965

Gange, R.A.
Cathode and method of operating the same—4167690

Hanak, J.J.
Cermet layer for amorphous silicon solar cells—4167015

Holmes, D.D.
Suppression of luminance signal contamination of chrominance signals in a video signal processing system—4167020

Holmes, D.D.
Suppression of chrominance signal contamination of the luminance signal in a video signal processing system—4167021

Kaganowicz, G.
Method of depositing a silicon oxide layer—4168330

Matsumoto, Y.
Video disc pickup apparatus—4164755

Matsumoto, Y.
Method for manufacturing a diamond stylus for video disc players—4165560

Nostrand, G.E.|Hanak, J.J.
Method of removing the effects of electrical shorts and shunts created during the fabrication process of a solar cell—4166918

Stanley, T.O.
Phosphor screen for flat panel color display—4166233

Toda, M.|Matsumoto, Y.|Osaka, S.
Disc record groove skipper—4164756

Williams, R.|Woods, M.H.
Method of testing radiation hardness of a semiconductor device—4168432

Wolkstein, H.J.|Goel, J.|Rosen, A.
Microwave power limiter comprising a dual-gate FET—4167681

Missile and Surface Radar

Profera, C.E.
Antenna feed system—4163974 (assigned to U.S. government)

Woodward, O.M.
Rotary joint—4163961

Picture Tube Division

Calamari, J.A., Jr.
Method for salvaging the light-absorbing matrix and support of a luminescent screen—4165396

Nubani, J.I.|Muenkel, R.L.
Stem-sealing method for assembling electron tubes including improved cullet collection—4165227

RCA Records

Khanna, S.K.
PVC molding composition—4168256

SelectaVision

Burrus, T.W.
Noise reduction apparatus—4167749

Solid State Division

Ahmed, A.A.
Television kinescope protection circuit—4164687 (assigned to U.S. government)

Ahmed, A.A.
Inverting buffer circuit—4166964

Nyul, P.
Electroluminescent semiconductor device having optical fiber window—4167744

Petrizio, C.J.
AC voltage regulator—4168476

Engineering News and Highlights

J. Edgar Hill appointed Division Vice President

J. Edgar Hill was appointed Division Vice President and General Manager, Commercial Communications Systems Division, on June 12, 1979. Before assuming his present position, Mr. Hill was Division Vice President and General Manager, Broadcast System, a position he had held since January 1977. For the preceding five years he was Division Vice President, Broadcast Marketing.

Mr. Hill joined RCA in 1935 and served in manufacturing and design engineering until transferring to sales in 1947. He was a sales engineer for RCA broadcast equipment in New England for seven years, when, in 1954, he was reassigned to the Camden home office and promoted to area sales manager. In July 1969 he was appointed to the post of Manager, International Sales, for the division.

Wright Heads Astro-Electronics

Paul E. Wright has been appointed Division Vice President and General Manager, Astro-Electronics, Princeton, N.J. Mr. Wright came to Astro-Electronics late last year as Division Vice President, Operations, after serving for more than a year as Division Vice President Engineering, Government Systems Division. Previously, Mr. Wright had been Director of the Division's Advanced Technology Laboratories (ATL), Camden, N.J. He joined RCA in 1958 and had held a number of engineering posts at ATL, including Manager, Applied Physics and Mechanics, and Manager, Advanced Mechanical Technology. Mr. Wright holds several patents and is the author of a number of technical papers in the fields of electronics, thermodynamics, and mechanics.

Duffy joins Corporate Engineering Education

Edward Duffy joined RCA Corporate Engineering Education as Administrator, Technical Education Programs, in October 1979. Previously, he was a Principal Instructor for Process Control Division of Honeywell at Fort Washington, Pa. His responsibilities, besides teaching, were the development of course programs which included microprocessor-based systems and electric control system analysis. He also taught a bio-medical system course and was the International Technical Support Advisor for the bio-medical instrumentation division of Smith Kline Corporation. Mr. Duffy will be involved with the identification of RCA's technical educational needs and with the implementation of courses to meet those needs.

Staff announcements

Americom

John Christopher, Vice President, announces the organization of Technical Operations as follows: **Walter H. Braun,** Director, Systems and Advanced Technology Engineering; **Paul W. DeBaylo,** Manager, Reliability and Quality Assurance; **Murray Fruchter,** Director, Terrestrial Systems Technical Operations; **Peter H. Plush,** Manager, Programs and Technical Operations Services; **Joseph J. Schwarze,** Manager, Space Systems Technical Operations.

Commercial Communications Systems

Jerry L. Copeland is appointed Division Vice President and General Manager, Mobile Communications Systems. He will report to **J. Edgar Hill,** Division Vice President and General Manager, Commercial Communications Systems Division.

Consumer Electronics

Eugene E. Janson is appointed Manager, Technical Support and Product Reliability. He will report to **J. Paul Belanger,** Director, Product Assurance.

J. Peter Bingham, Division Vice President, Engineering, announces the organization of New Products Laboratory as follows: **Eugene Lemke,** Chief Engineer, Television Advanced Development; **Robert M. Rast,** Manager, Digital Systems.

James A. McDonald is appointed Manager, Display Systems. He will continue to report to **Eugene Lemke,** Acting Manager, Engineering Development.

John C. Peer is appointed Manager, Television Systems Development. He will report to **Eugene Lemke,** Chief Engineer, Television Advanced Development.

Dal F. Griepentrog is appointed Manager, Engineering. He will report to **Loren R. Wolter,** President and General Manager, RCA Taiwan, Limited.

Judy Yeast is new Ed Rep

Judy Yeast has been appointed an Editorial Representative for the Solid State Technology Center. She succeeds Leslie Adams who has taken an educational leave.

Ms. Yeast joined RCA in August as Technical Programs Administrator. Previously she has worked in Data Management for Texas Instruments, and was responsible for preparing environmental impact statements for Southern California Edison Company.

As EdRep, Judy will assist SSTC engineers with papers for the *RCA Engineer* and inform the editors of new developments, professional activities, awards, publications and promotions in her area.

Staff announcements, continued.

Global Communications

Dean R. Ferguson is appointed to the newly created position of Manager, Operations, West Coast. In his capacity, Mr. Ferguson will be responsible for coordinating all installations and maintenance of data terminals and circuits and supervising all Technical personnel in the Western Region. He will report to **David Mer,** Manager, Operations and Engineering.

Laboratories

Brown F. Williams, Director, announces the organization of Display and Energy Systems Research Laboratory as follows: **David E. Carlson,** Head, Photovoltaic Devices Research; **Richard Williams,** Fellow, Technical Staff; **Joseph J. Hanak,** Fellow, Technical Staff; **Arthur H. Firester,** Head, Process and Applications Research; **John P. Russell,** Head, Display and Device Concepts Research; **Robert W. Shisler,** Manager, Advanced Development - Yokes (Lancaster); **Frans Van Hekken,** Manager, Advanced Development - Electron Guns (Lancaster); **Peter J. Wojtowicz,** Head, Electron Optics and Deflection Research; **P. Niel Yocom,** Head, Display Materials and Process Research; **Karl G. Hernqvist,** Fellow, Technical Staff; **Simon Larach,** Fellow, Technical Staff.

Bernard Hershenov, formerly Head, Energy Systems Analysis, has been named Director of the newly formed Solid-State Device Laboratory. He reports to **Henry Kressel,** Staff Vice President, Solid State Technology.

Bernard Hershenov, Director, announces the organization of the Solid State Devices Laboratory as follows: **Richard Denning,** Manager, Advanced Power Engineering - Somerville; **Michael Ettenberg,** Head, Optoelectronics Devices and Systems Research; **Henry S. Sommers, Jr.,** Fellow, Technical Staff; **Bernard Hershenov,** Acting, Control and Energy Management Systems; **Charles J. Nuese,** Head, Silicon Device Research; **Jacques I. Pankove,** Fellow, Technical Staff.

Robert D. Lohman, Director, announces the organization of VideoDisc Systems Research Laboratory as follows: **Marvin Blecker,** Head, Systems Evaluation Research; **Jon K. Clemens,** Head, Signal Systems Research; **James J. Gibson,** Fellow, Technical Staff; **Robert D. Lohman,** Acting, Player Research; **John A. van Raalte,** Head, Video Recording Research.

Daniel A. Walters, Director, announces the organization of the Communication Systems Research Laboratory as follows: **Emilie M. Lengel,** Manager, Data Communications and Computer Applications; **Leonard Schiff,** Head, Communications Analysis; **Paul Schnitzler,** Head, Transmission Technology; **Harold Staras,** Staff Scientist, Satellite Programs.

Edward C. Douglas is appointed Manager, Advanced Process Technology. He will report to **Israel H. Kalish,** Manager, Integrated Circuit Design and Process Development.

Solid State Division

Donald W. Laird, Jr. is appointed Administrator, Integrated Circuits Administration. He will report to **Richard C. Pinto,** Division Vice President, Integrated Circuits.

Larry J. Gallace is appointed Director, Quality and Reliability Assurance. He will report to **Carl R. Turner,** Division Vice President, Product Assurance and Planning.

Promotions

Consumer Electronics

Dal F. Griepentrog from Manager, Project Engineering, to Manager, Resident Engineering.

Alfred J. Schick from Member, Engineering Staff, to Manager, Technical Support/VCR.

Picture Tube Division

Charles W. Meredith from Associate Member, Technical Staff, to Manager, Instrumentation Engineering.

Solid State Division

Al A. Key from Leader, Technical Staff, to Manager, Product Engineering.

Walter F. Lawrence from Senior Member, Technical Staff, to Manager, Tooling and Product Engineering.

Joseph P. Paradise from Member, Technical Staff, to Leader, Technical Staff.

Randy Rhodes from Member, Technical Staff, to Leader, Technical Staff.

Nicholas Kucharewski from Leader, Technical Staff, to Manager, Design Engineering.

Professional Activities

RCA Engineers speak at the American Institute of Industrial Engineers Region VIII Annual Conference

John W. Freeman, Gerry L. Teague and **Dave Mishra** spoke at the Region VIII Annual Conference on "Proven Techniques for Reducing Indirect Labor Costs," which was held on October 26, 1979, at the Hilton Inn in Indianapolis, Indiana. Some of the aims of the conference were to provide useful techniques for the measurement and improvement of productivity in the industrial environment. The design and development of a universally-applicable, computerized system for labor and activity control of distribution and warehousing operations were discussed.

Dave Mishra, Chief Engineer, RCA Records, was the conference chairman and gave the welcoming speech. John W. Freeman, Leader of Systems and Industrial Engineering at RCA Records, and Gerry L. Teague, Senior Industrial Engineer at RCA Records, spoke on "Computerized Multi-Attribute Labor Control System for Warehousing." Some of the items discussed included characteristics of warehousing and order picking operations, requirements for performance measurement, design and development of computerized labor control, and the cost-benefit evaluation.


Conn


Johnson


Ketcik


Lutz


Mehling


Sampson


Strip

technical excellence

Moorestown announces seven second-quarter 1979 Technical Excellence Award Winners

Seven Technical Excellence Awards were presented to Missile and Surface Radar personnel during the second quarter 1979. The award winners and brief summaries of the citations are given below.

A.A. Conn — for directing the computer program functional integration task leading to the successful Navy OT-IIIB test at the CSED Site in May. The integration involved programs aggregating more than 700 thousand words, in 19 computers. His outstanding leadership in this area was a key contribution to RCA exceeding its planned goals for OT-IIIB.

R.S. Johnson — for his contribution to the planned updating of BMEWS (Balistic Missile Early Warning System). His comprehensive analyses led to simulations that firmly established sizing of the main computers required for upgraded BMEWS forward site operation. His tradeoff studies were largely responsible for the decision to upgrade, instead of replace, the existing BMEWS radars.

J.M. Ketcik — for technical direction of the integration and test of the AN/SPY-1A computer program Element Test Function (ETF) and the AN/SPY-1A ETF/ORTS fault detection system. He was a principal driving force in the effort that led to the integrated AN/SPY-1A ETF/ORTS system successfully tested by the Navy during Operational Test OT-IIIB at the AEGIS CSED Site.

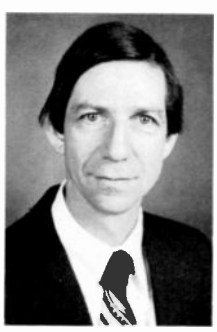
M.J. Lutz — for his leadership of a major definition effort involving emplacement and alignment requirements for the AN/SPY-1A array antennas, off-array waveguides, and transmitters. His knowledge of AEGIS equipment, ship design, and alignment techniques has expedited the AEGIS shipbuilder's design task and will simplify the AN/SPY-1A installation.

T.H. Mehling — for technical contributions and personal dedication exhibited in the integration of the AEGIS tactical software at the CSED Site during Navy test OT-IIIB. His intimate knowledge of AEGIS software and AN/UYK-7 computer hardware were important elements in ensuring the timely integration of five major AEGIS elements and two large simulator systems.

G.P. Sampson — for design and implementation of computerized database systems used to produce the AEGIS Ship Combat System cable detail design and installation drawings and equipment configuration definition documents. His two systems were vital in supporting the on-time delivery of the first in a series of cable drawings and configuration definition documents.

J. Strip — for outstanding contributions to AEGIS Weapon System development and testing. As coordinator of the DDG 47 Baseline Review Committee he led a team that achieved significant simplifications in the computer programs for AEGIS C&D and WCS. In addition, he provided a systems-level approach to coordinating and simplifying the OT-IIIB integration and test approach, enabling successful, on-schedule operational tests at CSEDS.

GCS presents Technical Excellence Awards


Daniel


Levin

The Technical Excellence Committee at Government Communications Systems has presented two awards for outstanding and innovative accomplishments. The award winners and brief summaries of their citations are given below.

Jim Daniel — for his highly innovative work in the conceptual design and detailed circuit development for the Small Terminal Modem. Jim's task was to develop a modem with state-of-the-art performance exceeding that of competitive equipment design, in a package a fraction the size of the competitive equipment. His accomplishment in meeting this difficult objective was a highly professional and commendable effort.

Stephen E. Levin — for his outstanding work on the Maroon Archer Program. (Since information regarding this program is confidential, we are unable to give the full citation for this award at this time.)

Licensed engineers

When you receive a professional license, send your name, PE number and state in which registered, RCA division, location, and telephone number to *RCA Engineer*, Bldg. 204-2, RCA Cherry Hill, N.J. New listings (and corrections or changes to previous listings) will be published in each issue.

Missile and Surface Radar

P.R. Kalata, NJ; 26010

Obituary eratum. Dr. George Robert Shaw's obituary in the Aug./Sept. 1979 issue of the *RCA Engineer* should have also stated that he received his Ph.D. degree from the University of Wisconsin in 1920.

Corporate Quality & Reliability Symposium Held at DSRC

A Corporate symposium on Quality and Reliability was held at the David Sarnoff Research Center in Princeton on October 18 and 19. The major theme of the symposium was "Quality & Reliability — Impact on Profitability." The comprehensive program covered reliability systems and techniques used in both commercial and government units of RCA.

Ed Schecter of Astro-Electronics served as chairman for the symposium which was attended by 100 people.

Three invited executives presented their views of RCA's approach to quality. **William C. Hittinger,** Executive Vice President, Research and Engineering delivered the opening address.

Mr. Hittinger noted that in today's environment of intense international competition and increasing consumer expectations and demands, outstanding quality/reliability and life must be designed into a product starting at the day of product introduction. He pointed out that the discipline required to do the job right the first time includes scrupulously establishing and following effective design rules without accommodative deviations and the need for disciplined teamwork between marketing, engineering and manufacturing.

W.J. Willoughby, Deputy Chief of Naval Materials, RM&Q, was the invited luncheon speaker of the first day. Mr. Willoughby stated that semiconductors represented one of the two major quality problems in military equipment. He described how screening represented a cost-effective approach and discussed some of the methods preferred by the Navy.

During the second day **Roy Pollack,** Executive Vice President, noted that good quality is good business. He stated that the word "quality" also covered reliability and safety. The mission of the quality experts is to develop a true quality climate throughout the Corporation even if it involves entering the confrontation zone between quality and immediate P&L objectives. He then backed up his belief that good quality is good business by recounting the horrendous recall costs incurred by industry due to faulty product, that quality determines our competitive position and market standing and that quality represents the best integrating report card to grade business.

All the talks presented at the symposium were videotaped. The videotape is being made available to RCA units through Engineering Education in Cherry Hill. For information on getting this videotape, contact Margaret Gilfillan on TACNET 222, ext. 5255.

Most of the papers presented will be published in the *RCA Engineer*, Dec./Jan., Vol. 25, No. 4.

First-Day Session: —
Quality & Reliability — Impact on Profitibility

"Quick Analysis of Test Rejects and Repairs — the DARTS System"	**H.S. Baird** RCA Laboratories
"Design for High Reliability and Low Cost"	**D.I. Troxel** Government Communications Systems
"Reliability of Unattended Equipment"	**R.E. Killion** Missile and Surface Radar
"Low-Arc Picture Tube Development Using Statistically Designed Experiments"	**C.W. Thierfelder** **F.J. Hinnenkamp|D. Shainin** Picture Tube Division
"Reliability Growth Testing"	**E.B. Gamble** Avionics Systems
"The Interface Between Reliability and Systems Design"	**C.T. Schilsky** Missile and Surface Radar
"Using Standards for Reliability Results"	**W.S. Sepich** Broadcast Systems
"Software Reliability"	**D.C. Bowen** Government Communications Systems
"Do MIL Spec Quality Systems Pay Off in the Commercial World?"	**W.E. Bradley|G.J. Buchko** Electro-Optics and Devices
"Data Reporting via PMC"	**J.W. Gaylord** Solid State Technology Center
"Process Control Feedback Utilizing Manufacturing Attributes Planning System	**G.J. Giering** Electro-Optics and Devices

Second-Day Session:
New Technology Impact on Reliability

"High Reliability vs. Chip Complexity"	**J. Hillbrand|K.R. Anderson** Government Systems Division
"Design to Unit Cost Production Problems"	**F.P. McGurk** Automated Systems
"The Reliability Engineering Laboratory Organization and Functions in the Solid State Division (Including Step Stress Testing and Real-Time Indicators)"	**L.H. Gibbons** Solid State Division
"RCA Americom Quality Assurance: Measurement and Action"	**P. DeBaylo** Americom
"Measuring Quality of Service"	**J.E. Steoger** Service Company
"IC Reliability Assurance Plan"	**H.E. Elrod** Consumer Electronics

Editorial Representatives

Contact your Editorial Representative, at the extensions listed here, to schedule technical papers and announce your professional activities.

Commercial Communications Systems Division

Broadcast Systems
BILL SEPICH* Camden, N.J. Ext. 2156
KRISHNA PRABA Gibbsboro, N.J. Ext. 3605
ANDREW BILLIE Meadow Lands, Pa. Ext. 6231

Mobile Communications Systems
KARL NEUMANN* Meadow Lands, Pa. Ext. 6444

Avionics Systems
STEWART METCHETTE* Van Nuys, Cal. Ext. 3806
JOHN McDONOUGH Van Nuys, Cal. Ext. 3353

Cablevision Systems
JOHN OVNICK* N. Hollywood, Cal. Ext. 241

Government Systems Division

Astro-Electronics
ED GOLDBERG* Princeton, N.J. Ext. 2544

Automated Systems
KEN PALM* Burlington, Mass. Ext. 3797
PETER DIACONOFF Burlington, Mass. Ext. 2975

Government Communications Systems
DAN TANNENBAUM* Camden, N.J. Ext. 3081
HARRY KETCHAM Camden, N.J. Ext. 3913

Government Engineering
MERLE PIETZ* Camden, N.J. Ext. 2161

GSD Staff
ED MOORE* Cherry Hill, N.J. Ext. 5833

Missile and Surface Radar
DON HIGGS* Moorestown, N.J. Ext. 2836
JACK FRIEDMAN Moorestown, N.J. Ext. 2112

Solid State Division

JOHN SCHOEN* Somerville, N.J. Ext. 6467

Power Devices
HAROLD RONAN Mountaintop, Pa. Ext. 633 or 27
SY SILVERSTEIN Somerville, N.J. Ext. 6168

Integrated Circuits
FRED FOERSTER Somerville, N.J. Ext. 7452
JOHN YOUNG Findlay, Ohio Ext. 307

Electro-Optics and Devices
RALPH ENGSTROM Lancaster, Pa. Ext. 2503

*Technical Publications Administrator, responsible for review and approval of papers and presentations.

Consumer Electronics

CLYDE HOYT*Indianapolis, Ind. Ext. 5208
FRANCIS HOLT Indianapolis, Ind. Ext. 5217
PAUL CROOKSHANKS Indianapolis, Ind. Ext. 5080
STEVE RACE Indianapolis, Ind. Ext. 5636
DON WILLIS Indianapolis, Ind. Ext. 5883

SelectaVision VideoDisc Operations

W.M. WORKMAN* Indianapolis, Ind. Ext. 3235

RCA Service Company

JOE STEOGER* Cherry Hill, N.J. Ext. 5547
RAY MacWILLIAMS Cherry Hill, N.J. Ext. 5986
DICK DOMBROSKY Cherry Hill, N.J. Ext. 4414

Distributor and Special Products Division

CHARLES REARICK* Deptford, N.J. Ext. 2299

Picture Tube Division

ED MADENFORD* Lancaster, Pa. Ext. 3657
NICK MEENA Circleville, Ohio Ext. 228
JACK NUBANI Scranton, Pa. Ext. 499
J.R. REECE Marion, Ind. Ext. 5566

Americom

MURRAY ROSENTHAL* Piscataway, N.J. Ext. 4363

Globcom

WALT LEIS* New York, N.Y. Ext. 7711

RCA Records

DAVE DEVARAJAN* Indianapolis, Ind. Ext. 6109

NBC

BOB MAUSLER* New York, N.Y. Ext. 4385

Patent Operations

JOSEPH TRIPOLI Princeton, N.J. Ext. 2992

Research and Engineering

Corporate Engineering
HANS JENNY* Cherry Hill, N.J. Ext. 4251

Laboratories
MAUCIE MILLER Princeton, N.J. Ext. 2322
JUDY YEAST Somerville, N.J. Ext. 7357

RCA Engineer

A technical journal published by
Corporate Research and Engineering
"by and for the RCA engineer"

Printed in the U.S.A. Form No. RE-25-3