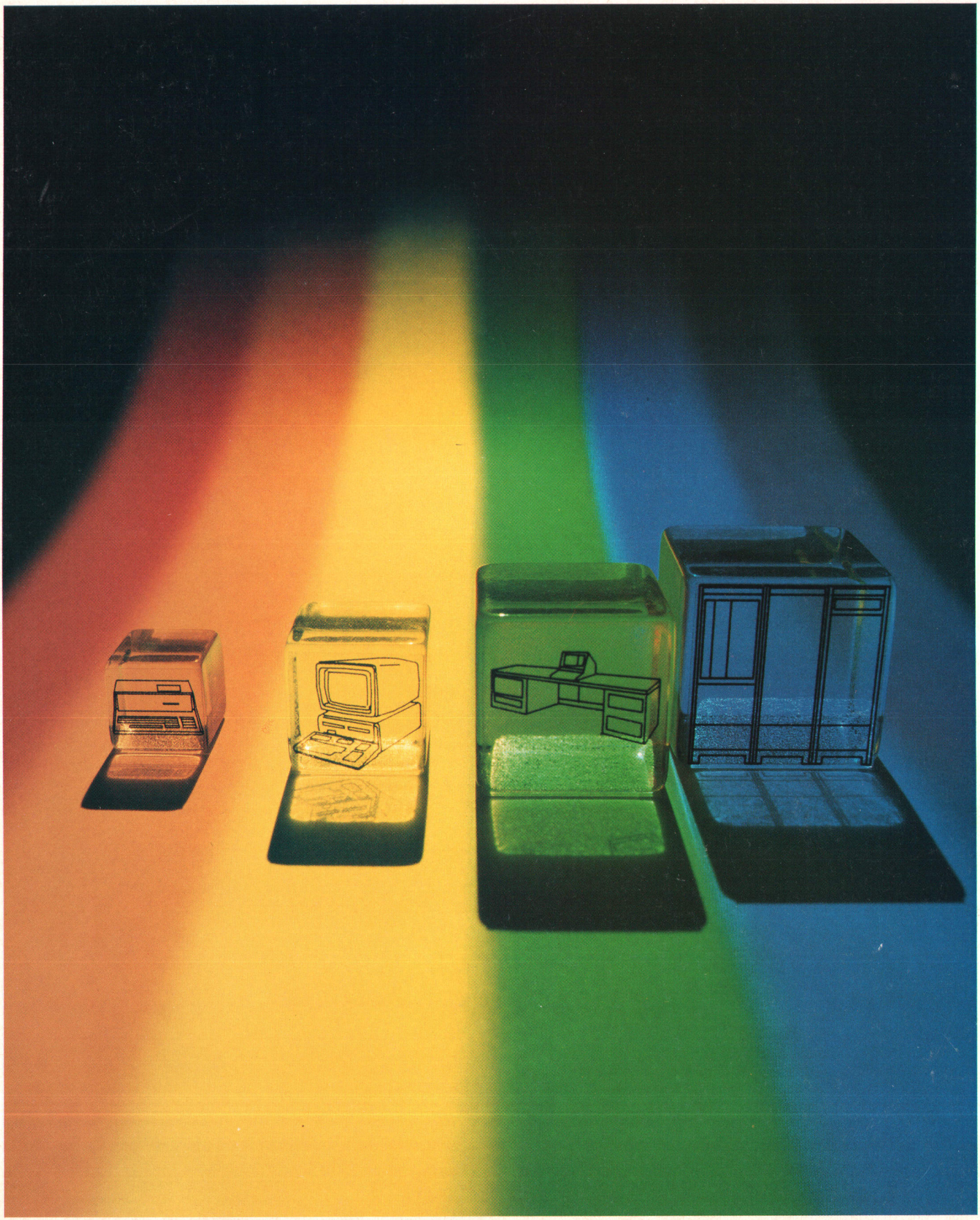
HEWLETT·PACKARD
JOURNAL

AUGUST 1985

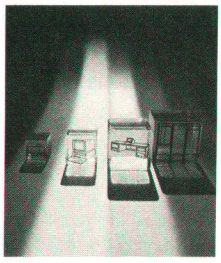


HEWLETT-PACKARD JOURNAL

August 1985 Volume 36 • Number 8

Articles

In this Issue



HP's next generation of computers is now under development, with first product introductions expected some time in 1986. The code name given to this development project is Spectrum, because the new computer line will include a spectrum of models ranging from desktop workstations to mainframe-class machines, all based on the same architecture. Our cover repeats the Spectrum theme, and in the article on page 4, Joel Birnbaum and Bill Worley introduce us to the basic principles of the new architecture and tell us what HP hopes to accomplish with it. Their article describes the new architecture's definition at HP Laboratories, a process that involved analyzing billions of instruction executions to determine the optimal instruction set for the new machines. Although the new architecture fits loosely within the class known as reduced instruction set computers, or RISCs, it also takes full advantage of VLSI (very large-scale integration) and new software technology. It does not, however, depend on any particular circuit technology, so instead of being rendered obsolete by the inevitable development of new circuit technologies in the future, it will allow designers to exploit new technologies for further performance gains. The first computers of the new generation will extend the high end of the present HP 3000 business computer and HP 1000 real-time computer lines. For HP customers, migration to the new models is expected to be simple. Existing HP 3000 software should run unmodified at about present speeds, or can be recompiled or further modified to improve performance. HP 1000 software will transport to the new machines using specially developed utilities and emulation capabilities.

Having glimpsed the future of HP computers, we turn to the state of the art in instruments. On pages 11 to 29, the designers of the HP 3326A Two-Channel Synthesizer describe its contributions to the measurement art. Sources of sine, square, and pulse waveforms are basic test instruments, and the HP 3326A provides two of them, both operating in the frequency range of dc to 13 megahertz. A major contribution is an internal phase calibrator that precisely defines the phase relationship between the two outputs, eliminating the need for extra equipment to do this. The same phase calibration capability can be used to set the phase relationships between the outputs of several HP 3326As to provide multiphase signals for testing guidance systems or phased array radars, with no extra equipment required. The two HP 3326A outputs can be used separately or added together, or one can modulate the other's amplitude or phase. An unusual feature is discrete sweep, up to 63 frequencies produced in any user-selected random order with independently programmable dwell times before switching to the next frequency. The HP 3326A uses HP's fractional-N frequency synthesis technique, as described in the article on page 11. The design requires down-conversion of a reference frequency source, and in the article on page 25, project manager Bill Spaulding describes how the challenge of designing a high-level active mixer for the down-converter was met successfully.

HP's Semiconductor Productivity Network is a family of products, primarily software, intended to automate and tie together the various steps in the making of integrated circuit chips. Last month's issue described the process control module, PC-10. On page 30, Reed White tells us about TC-10, the module that addresses the problem of acquiring data from equipment designed by many different manufacturers without detailed standards for data format and message content. The design of the module required a touch of prophecy to predict which way standards might go, and a willingness to risk taking a stand on which way they ought to go.

-R. P. Dolan

What's Ahead

In September, the spotlight will be on HP medical instruments and computers, as HP designers describe the HP 4760 family of intelligent cardiographs and the HP 3000 Series 37 Computer, the smallest and lowest-cost full-size HP 3000.

Beyond RISC: High-Precision Architecture

An introduction to scaling, complexity, and HP's new computer architecture.

by Joel S. Birnbaum and William S. Worley, Jr.

THE BRITISH GENETICIST J. B. S. Haldane once remarked that nature is not capricious about either the size or the complexity of its creations. For every creature there is a most convenient size, and a large change in size inevitably carries with it a change of form. Consider a typical small animal, say a microscopic worm, which breathes by diffusion through its smooth skin, absorbs food by osmosis through its straight gut and excretes by means of a simple kidney. If the worm's dimensions are increased tenfold in each direction, then through simple scaling laws, which relate surface area to the square of linear dimension and volume to the cube, the creature's volume (that is, its weight) is increased a thousand times. This means that if the new larger creature is to function as efficiently as its miniature counterpart, it will need a thousand times as much food and oxygen per day, and it will excrete a thousand times as much waste. However, if its shape is unaltered, the scaling laws dictate that the surface of the skin, intestine, and kidney will be increased only a hundredfold. Thus, a natural limit to the size of simple creatures is soon reached. For larger animals, the ratio of surface area to volume is altered by the creation of special, more complicated structures, such as lungs, gills, and coiled, rough-surfaced intestines. A man, for example, has the equivalent of 100 square yards of lung. We are led to the surprising conclusion that the higher animals are not larger than the lower because they are more complicated, but that they are more complex because they are larger. Nature does not complicate its creations unnecessarily, and it is a general rule of biological adaptation that function and form are closely related.

Man is not nearly so consistent as nature in controlling the complexity of his creations, and in the case of the digital computer may have complicated it unnecessarily. Albert Einstein once said that things should be made as simple as possible, but not simpler; this suggests that for a given size (i.e., performance) computer, there will be a commensurate level of complexity (i.e., concurrency). If a computer architecture is intended to define a scalable family of machines, then it follows that it must include provision for the hardware, software, and system organization compromises that are inevitably involved when the architecture is implemented.

Computer architecture and implementation are quite analogous to building architecture and construction. An architect designs a building to achieve a certain balance of aesthetics and function; it is up to the construction engineers to realize this concept subject to the structural integrity of materials and the laws of physics. Similarly, computer architects must make design decisions to achieve the goal of performing useful work within a set of constraints imposed by product size, range, cost, and usability. Hardware and software engineers must then translate this architecture into viable implementations, but unfortunately no comprehensive theory of computation guides their trade-offs. Modern computers are often the result of *ad hoc* decisions by these engineers, and tradition has frequently played too large a role.

In 1981, a group of architects and engineers at Hewlett-Packard Laboratories, with help from representatives of HP's computer divisions, began a series of precision measurements about computational behavior under a wide range of execution scenarios so that these trade-offs could be made more knowledgeably. The results of those studies led, through a process of iterative optimization, to the specification of an unconventional computer architecture, which defines a unified family of scalable computers offering significant cost/performance advantages over more traditional designs. Refinements by engineers in HP's product divisions followed, and ensuing implementations of this architecture have verified its potential over a broad range of size and function. This paper will discuss the design objectives and some of the basic principles of the architecture, emphasizing departures from orthodoxy. It will serve as an introduction to later papers which will present detailed treatments of the architecture, engineering level discussions of some implementations, and the results of performance analyses.

Design Objectives and Basic Principles

The program to develop implementations of the new architecture is code-named Spectrum. From the outset, the objective for the Spectrum program was to develop a microcomputer and minicomputer family with scalable cost and performance across the full range of product sizes and application areas then addressed by all existing Hewlett-Packard computer products. This task was further tempered by a very important overriding constraint: the new architecture must provide a smooth application migration path from all of these existing products. It is this albatross of compatibility that has usually prevented computer manufacturers from unifying their disparate product lines, for the cost of recoding applications is usually too great to be borne, and designs that compromise the specific requirements of technical, commercial, and real-time applications for the benefit of uniformity usually suffer degraded performance when compared with specifically optimized machines.

As a design objective, however, such a scalable unified architecture offers many incentives. If subsystem and peripheral interfaces can be consistent across family members and application domains, then the development effort for the family will be greatly reduced, as will the cost of maintenance and modification of both the hardware and

Architecture Genealogy

The first von Neumann computers (Fig. 1a) were simple, hardwired machines built from vacuum tubes. Their unreliability imposed strict physical limits on their complexity, and as a result they had very few registers in their data paths. Logic speed and memory speed were in approximate balance. When vacuum tubes gave way to solid-state designs, volume and power dropped by two orders of magnitude and logic speed increased by an order of magnitude. This led to a serious imbalance between logic and memory speeds. The response of designers was the microcoded machine (Fig. 1b). Such a machine exploits the speed imbalance between logic and main memory to reduce the real logic in the machine by substituting a microcoded interpreter, running out of a small, fast control store, for the missing hardware. More significantly, however, this design removes the physical constraint on architectural complexity, since the architecture is now a function of software (firmware) in the control store.

Since the microcoded machine did not address the central problem, which was slow memory access, the situation was ripe for the invention of an effective memory buffering mechanism known as a cache (Fig. 1c), which provides (almost) the access speed of a small memory with the capacity of a large, slower primary memory.

The success of cache designs and the insights provided by studying the instruction traces collected to aid in their design led to the proposed design shown in Fig. 1d, which provides all the data memory bandwidth of a cache design without the overhead of instruction interpretation.

The new HP architecture that is now the basis of the Spectrum program is an optimized case of Fig. 1d in which the data path complexity has been minimized to reduce cycle time and instruction coding has been tuned to prevent significant code size expansion.

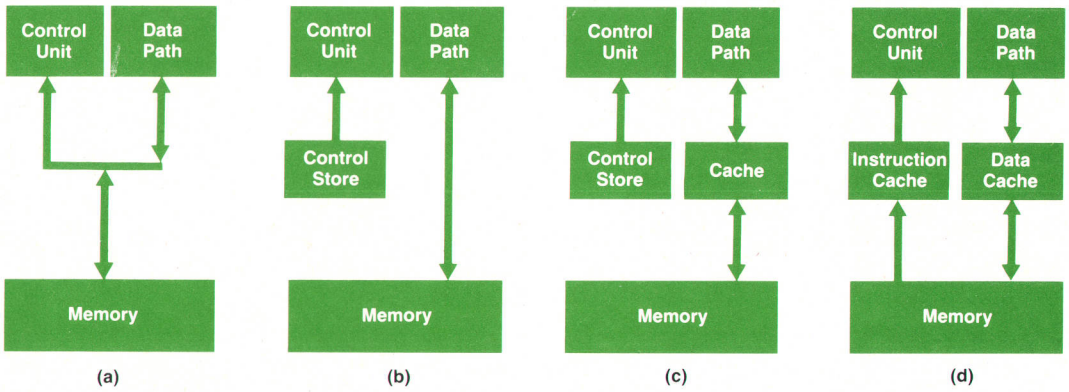


Fig. 1. *Computer architecture types: (a) von Neumann (b) microcoded (c) cache (d) proposed.*

the software. There are concomitant customer advantages in terms of cost of ownership, flexibility of configuration, and modularity, particularly in networks, where the same code can now run on workstations, network servers, and timeshared distributed mainframes. The challenge is to create a common, simple core, which can be incrementally enhanced by the addition of functionally specific structures for particular application execution environments. At the outset, we established and have adhered to three design doctrines:

- The architecture must support essentially all systems and applications programming being done in high-level languages.
- The architecture must scale in a technology-independent and implementation-independent way.
- It must be able to support the efficient emulation of or migration from previous architectures including their I/O subsystems.

In many ways, the new HP architecture is an architecture determined from the point of view of the software it must run. It was designed by a team of hardware and software engineers who sat side by side during all phases of its design. It falls roughly into the class of what have come to be known as reduced instruction set computers, or RISC machines. In many ways, as we shall see, the term RISC is unfortunate, since a reduced instruction set is not the goal, and is, in fact, only one part of the story. It does no good, for example, to build an engine capable of rapid execution if it cannot efficiently support a large virtual address space, or if it is idle much of the time waiting for information to arrive from the storage subsystem, or if execution is blocked by contention with the input/output mechanisms. It is crucially important to consider the storage hierarchy and the input/output subsystems as integral to any system design, for it is the taste with which these elements are combined that determines performance at the system throughput level and that further determines the effectiveness with which specific instantiations of the architecture can be optimized.

In many ways, RISC machines in general and HP's new architecture in particular appear to fly in the face of conventional wisdom about computer design. Many of the conventional axioms were formulated in the pre-VLSI era, and at a time when software technology, particularly that of control programs and compilers, was far less sophisticated than it is today. Rethinking of fundamental assumptions has often been curtailed by the need for compatibility with an installed product base. HP's new architecture is the result of a reexamination of these assumptions, aided by extensive and precise measurements, in the light of the capabilities of modern integrated circuits and systems software. To highlight differences, the major design principles

will be presented as a series of paradoxical assertions.

Paradoxical Assertions

An architecture based on a primitive hardwired instruction set, constrained to permit implementations that can execute most instructions in a single cycle, can lead to better cost/performance than a more complex microcode-based architecture.

For many years, the principal limitation to the growth of computing has been the extremely poor productivity of software application development. In fact, at least two thirds of the world's programmers are still involved in the maintenance and modification of old code. Therefore, it is not surprising that computer architects for many years have sought to raise the level of abstraction for the creation of both systems and applications programs, since programming in high-level languages has been shown to improve productivity in both the creation and the maintenance phases.

The principal architectural trend of the 1970s was to take advantage of the rapidly decreasing costs of hardware, brought about by increased chip densities, by implementing in hardware as high-level an interface for the instruction set as possible. The most common way of achieving this was through microcode interpretation of the architected instructions into the fundamental machine level. The justification for the microcode was often given in terms of greater performance, but in fact, this performance depended mostly on the microcode's being resident in high-performance control store.* The effectiveness of this scheme is then dependent upon the ability of the system architects to guess in advance which instructions can most benefit from being resident in the control store. In fact, a second advantage of microcode was probably more responsible for its pervasive use: system development modifications and tuning could now be done through microprogramming rather than hardware redesign, purportedly improving development costs and schedules because of the enhanced flexibility. For many machines in the intermediate (supermini) class, the microcode became large and complex because the architects relegated many complicated decisions relating to the specific execution environments to this interpreter.

During the mid-70s, far less intrusive and more precise performance measurements than had been possible before began to be made in a variety of university and industrial laboratories. Through the mechanism of instruction tracing across a wide variety of workloads, these measurements brought a new level of understanding of what was actually going on inside the computer execution units. It was discovered that the simple instructions, such as branch, load, store, and add, dominated the instruction execution frequencies and that the complex instructions were seldom used. The justification for executing such simple instructions interpretively in microcode is difficult, since a several-cycle performance penalty is usually exacted.

*The physical density of RAM in the 1970s was much lower than that of ROM, leading to lower speeds for RAM than for ROM. Similarly, initialization, refresh, and control were easier for ROM, and reliability was better.

These observations led to a new breed of machine which had a reduced number of primitive hardwired instructions—the ones that were executed most frequently—which could be constrained to execute in a single cycle (with the exception, of course, of instructions that must access storage, such as loads and branches).

The Spectrum program design team at HP Laboratories analyzed billions of instruction executions and used the results to create an instruction set in which the most frequently used instructions are implemented directly in hardware. While the early literature on so-called reduced instruction set computers tended to emphasize the distinction between microcoded and hardwired instructions, an equally important characteristic of the new HP architecture is that its designers have invested in high-speed general-purpose registers instead of microcode. A computer whose highest (fastest) level of the storage hierarchy is general-purpose registers has several advantages. The obvious one is that register-to-register instructions are intrinsically faster than those requiring storage access. Another is that modern compilers can analyze data and control flows and can thus allocate registers efficiently. The reuse of information held in registers is enhanced dramatically, in most instances producing shorter total path lengths and far fewer storage accesses. Yet another advantage of simple register-based instruction sets is that computations can often be broken into independent portions, which frequently permits greater overlap between the processing units and the storage hierarchy.

Modern computers typically have an execution pipeline which permits new instructions to be begun before prior ones are completed. HP's new architecture acknowledges this reality by making characteristics of the pipeline architecturally visible; delayed branches and loads are two consequences that will be discussed later. The effect is that an instruction can be executed almost every cycle. The pipeline can be relatively simple, because advances in compiler technology enable the clever scheduling of instructions to take advantage of cycles that would otherwise be idle.

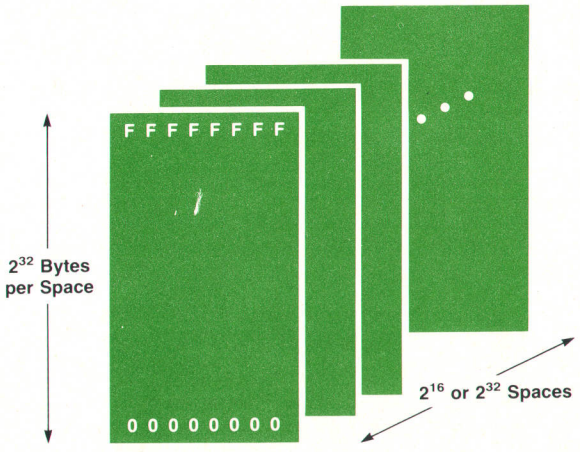


Fig. 1. *The new HP architecture provides for up to 2^{32} distinct virtual address spaces, each up to 2^{32} bytes long. Implementations may limit the number of virtual spaces to zero (real addressing only), 2^{16}, or the full 2^{32}.*

Some complex microcode-based machines have writable control store to permit different complex instruction execution environments to be created to match the task at hand. This has proven, in general, a very complicated task, and very few end users have availed themselves of this capability. In one sense, the new HP architecture can be thought of as having no microcode, but it is equally valid to consider it as having only microcode. The distinction is that the HP microstore is the normal dynamically managed storage hierarchy instead of a special control store. It is thus dynamically alterable and operates at a far higher level. This type of microcode can be written using high-level languages and standard development environments and can be paged in and out of the system in a straightforward fashion. In this way, further performance enhancements often result from the specialized construction of complicated instructions from the primitive hardwired ones, since these subroutines are frequently a better match for the specific run-time situation than generalized microcoded instruction sets.

The critical question about reduced complexity architectures is whether or not the performance gains that result from direct execution will be lost through increased path length resulting from having to formulate complex instructions in terms of the simple ones. This valid question does not have a single simple answer. In many cases, an optimizing compiler will be able to produce path lengths no longer than conventional solutions. In other cases, it is judicious to add hardware to implement complex facilities; examples might be floating-point instructions, decimal arithmetic instructions, high-performance multiplications, and array operations. In general, the applications workload is the determining factor. More will be said later about the capabilities of the new architecture to permit flexible trade-offs between the hardware and software. The design philosophy used throughout is to preserve the performance of the primitive instructions and not to invest in hardware assists until both frequency of occurrence and functional need justify it.

A primitive instruction set is a better match to high-level languages than a complex one.

Although the dramatic advances in shrinking the size of computing devices with concomitant benefits in power, speed, and cost have garnered most of the attention over the last three decades, there has been steady and important progress in software technology as well. In particular, compilers have grown far more capable of analyzing and optimizing programs. A modern globally optimizing compiler can produce code that rivals handcode in all but low-level routines requiring direct control of machine resources. The new HP instruction set has been chosen to be the target for such a compiler. As a result, it is very regular; all instructions are 32 bits in length, opcodes and register fields always occur in the same locations, and there is great symmetry in the functions provided. The result is that with the exception of a few performance-critical routines such as the first-level interrupt handler, virtually all programming is done in high-level language. Furthermore, short, precise instructions serve as an excellent base for interpretation, so languages like LISP can be executed quite effectively, even without hardware assists.

Some insight into why a simple instruction set is a better match for a high-level language compiler can be gained by recognizing that human beings are quite good at formulating complex strategies and data structures to use powerful instructions. Compilers, however, are most effective at simple repetitive execution with a minimum of special cases. Compiler accuracy and code performance both suffer dramatically as the complexity increases. The compilers for the new HP architecture are designed to maximize register reuse and reschedule instruction sequences to maintain pipeline efficiency. By elimination of unused code and precalculation of many quantities at compile time, further performance gains are achieved.

A cache should not be hidden.

The efficiency of the storage hierarchy is crucial to any high-performance machine, since storage is intrinsically always slower than logic. The high-speed register set is the highest level of this hierarchy and has been discussed above; the next level is the cache or high-speed buffer memory, which helps to lessen the speed mismatch between the logic units and main memory. Its purpose is to achieve almost the bandwidth of the cache but with the capacity of the main store. The cache, as its name implies, is usually hidden from the perspective of the software, because it is generally not part of the architecture but an implementation afterthought. In the new HP architecture, the cache is exposed and the storage hierarchy is explicitly managed. This is made possible by the careful design of the operating system kernels, and by disallowing run-time instruction modification unless responsibility for it is taken in the software. The architecture allows implementations to buffer information in the high-speed storage to reduce the time required to translate virtual addresses, fetch instructions, and fetch and store data. The architecture supports up to 2^{32} virtual address spaces, each 2^{32} bytes long (Fig. 1). Virtual address translation is done by translation lookaside buffers, or TLBs. Separate TLBs can be used for instructions and data, or one can be used for both. The high-speed caches can also be separate for instructions and data, thus increasing the bandwidth from memory substantially, or one can be used for both. Splitting the cache effectively doubles its bandwidth since it allows us to fetch data and instructions simultaneously. This departure from the von Neumann concept of indistinguishability of instructions and data is achieved through the provision of instructions to permit the synchronization and management of the caches when modifications of instructions by stores are involved.

Several other strategies are employed to minimize the idle time of the processor while the storage hierarchy is being exercised. One example is the ability to execute delayed branch instructions, which differ from conventional branches in that the processor executes the instruction after the branch while the branch target is being fetched from storage, often using what would otherwise be an idle cycle (Fig. 2a). HP's new architecture also contains novel facilities for performing program logic functions without

requiring branch instructions. Similarly, code rearrangement can often overlap the execution of other instructions with instructions that access storage (Fig. 2b). The register-intensive computation model reduces the number of storage accesses, since only load and store instructions access main memory, all other computation being performed among registers or between a register and an immediate field contained in an instruction. The generous number of registers ensures that the register allocation scheme of the compiler can assign the most frequently used variables to them so as to incur a fraction of the memory references required by traditional architectures.

HP high-precision architecture provides a better base for a scalable range of specific product implementations than more customized microcoded architectures.

The new HP architecture is technology-independent in the sense that implementers can choose a technology based only on considerations of size, cost, and performance trade-offs with the assurance that code written on any member of the family will execute unchanged on any other member. This means, for example, that a development system on a large mainframe can be used to develop code that will run on a single-chip microprocessor in a personal computer or workstation, or that code normally executed in the workstation environment can run on a network server or timeshared mainframe when that is convenient or desirable. Within a given technology, performance will depend strongly on the trade-offs between hardware and software that have been made and on the size and number of caches that have been provided, since efficiency for a given application is quite sensitive to the number and locality of storage accesses.

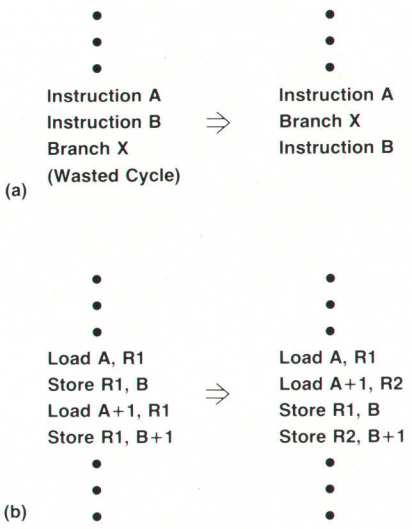


Fig. 2. *Branch and load scheduling. (a) One-cycle delayed branches can almost always be used to eliminate a cycle of cache delay by permuting instructions to occupy the branch's "delay slot" usefully. (b) Two ways of moving two words from A,A+1 to B,B+1. The straightforward code on the left incurs "load interlocks" between each load instruction and its adjacent store instruction, since the cache can be expected to require a full cycle to respond. The rescheduled code on the right uses an additional register to overlap the data fetch delays, resulting in a two-cycle saving.*

High-performance hardware in the form of assist processors can be added to any basic high-precision architecture system to enhance its performance or functionality (Fig. 3). Three categories of assist processors are differentiated by the level at which they interface to the memory hierarchy. *Special function units* interface to the memory hierarchy at the general register level and can be thought of as alternative processing units or as an alternative path through the execution unit of the existing processor. Examples include fixed-point binary multiply and divide units, emulation assists, and encryption or decryption hardware.

Coprocessors attach to the memory hierarchy at the level of the caches. They generally have their own internal registers and hardware evaluation mechanism. Examples of coprocessors are high-performance graphics or floating-point arithmetic engines. Note that the cache-to-coprocessor bandwidth can be different from the cache-to-main-processor bandwidth.

The third type of assist processor attaches to the main memory bus. High-precision architecture supports various types of multiprocessing with special instructions and control features. Multiprocessing can be homogeneous and tightly coupled, or the attached processor can be specialized (e.g., an array processor or an I/O controller). In all cases, such *attached processors* typically have their own registers and local storage.

High-precision architecture permits great flexibility in choosing cost/performance points based primarily on the choice of technology and secondarily on the configuration of the memory hierarchy and special VLSI hardware assists. In this way, implementations can be tuned for specific application domains and the same fundamental architecture can serve commercial, scientific, and real-time applications.

The I/O architecture of the computers being developed by the Spectrum program has been designed to permit complete control of I/O devices by privileged or nonprivileged code written in high-level language. It is based on a memory-mapped addressing structure (that is, all communication with external devices is done through registers that are addressed using normal load and store instructions). This conforms with the objective of making it possible to do all programming in high-level language, since the uniformity of addressing makes it possible to treat device drivers in a nonspecialized way. Since the I/O devices are addressed exactly as if they were memory locations, the same protection structure can be used for both, which greatly simplifies the overall system. We believe that this will protect the I/O better than many other more expensive schemes and can even be extended to protect individual I/O devices. The architecture is flexible enough to accommodate adapters that convert the new HP I/O protocol to other protocols and vice versa, so that foreign devices can be connected to new HP systems when appropriate. Just as in the storage hierarchy, the I/O architecture supports several types of interfaces ranging from sparse, direct control of simple devices to very rich functional controllers. The direct memory access of I/O is organized so as to present minimum interference in the storage hierarchy, which enables the processor to run at reasonable speed even when the full I/O bandwidth is being

used.

Since some members of the new HP computer family will be used as instrument and real-time process controllers, we have attempted to optimize the ability of the new architecture to respond to a wide range of external and internal interrupt conditions. Here again, the speed and uniformity of the instruction set and the simplicity and regularity of the control interfaces yield significant performance and functional advantages over more conventional hardware-dependent implementations. This structure has also enabled us to build an unusually rich set of hardware instrumentation into some of the early implementations. This has proven valuable during development, and we think it will help customers a great deal in tuning the performance of their configurations for particular workloads.

A reduced-complexity, high-precision architecture enables more graceful migration from other architectures than conventional approaches.

This is, of course, the overriding constraint for any new architecture proposed by a manufacturer with a large installed base of application programs. We believe that migration from existing HP products to the new computers will be among the least difficult that the industry has yet seen. There are many reasons for this, but perhaps the principal one is that the simplicity of the instruction set and control paths and the inclusion of versatile field-isolation facilities make the new machines extremely good interpreters, and so guarantee that software emulation of previous machines will be unusually effective where time dependencies are not critical, particularly since the system spends much of its time in the control program and software subsystems, which run in native mode. The emulated code can be freely combined with recoded native versions of the most critical routines, thus providing a continual incremental upgrade path. For many applications, simply recompiling the source code will yield significant improvements in performance. When source code is unavailable, an optimizing object-code compiler, which treats the object code of an earlier machine as input, has been found to produce important performance gains with acceptable code size expansions in a large percentage of cases. For those few cases where direct emulation or recompilation is not effective, migration tools have been developed.

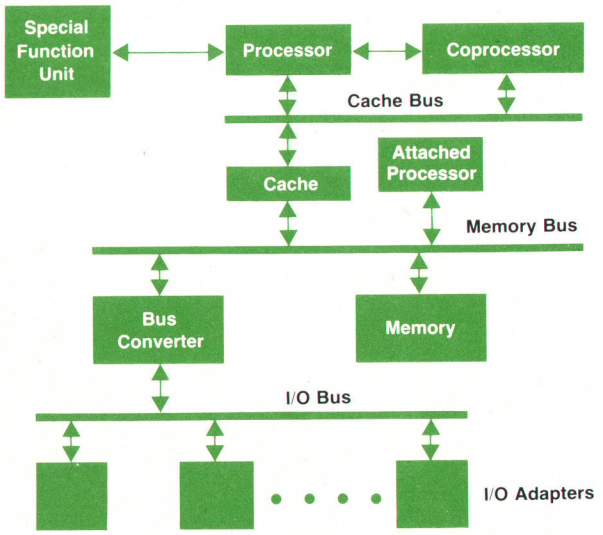


Fig. 3. *High-precision architecture system processing unit block diagram, showing high-performance assist processors added to enhance the performance or functionality of the basic system. Special function units connect directly to the processor, and can be thought of as alternative data paths associated with the processor's general registers. Coprocessors, on the other hand, have a control interface with the processor, but they have a private register set and a direct data interface with the cache. They are appropriate when special data types are supported that have little overlap with main processor data types, such as floating-point data. Attached processors are a third level of specialized assist. They have private registers, private data paths, and local storage. Attached processors execute separate instruction streams and are best suited for specialized operations on complex data types—for example, vector processing.*

Although the new HP high-precision architecture is very efficient at software-based emulation, hardware assists in the form of special function units, coprocessors, or attached processors can be provided where indicated. The I/O subsystem is designed to permit native and foreign mode device attachments, including adapters from selected previous HP I/O architectures. Future versions of the system will incorporate features for fault tolerance and high availability. Since the software is able to identify each module in a particular system, self-configuration without operator intervention will also be possible.

Conclusion

The high-precision architecture being developed by the Spectrum program provides the base for the next-generation family of HP computers. We think it will provide a cost/performance leadership position for commercial, scientific, and real-time applications and will scale across a wide range of function and performance. It will enable all HP computer systems to converge to a single architecture, with almost all programming done in a consistent way in a variety of high-level languages and operating system environments. Systems based on the new architecture will be able to provide compatible execution of application programs written for most earlier-generation HP systems. When required, attachment of older I/O devices will be possible.

The Spectrum program does not result from a single new idea, but rather is the result of the tasteful synthesis of the enormous progress in semiconductor technology with an increased understanding of the role of compilers and operating systems in optimizing an overall system design.

We have tried to learn from nature that simplicity is a relative term and have added complexity only where usage justifies it.

Later articles in this series will present greater architectural detail and will chronicle the specific implementations and their performance.

Acknowledgments

The Spectrum program is the result of the ideas and accomplishments of many people from many different

areas within HP. Space does not permit inclusion of a list of the names of all those whose contributions should be acknowledged. Future papers and articles on the Spectrum program will provide the opportunity for these individuals to describe their work and to be recognized for their accomplishments. The authors wish to thank Michael J. Mahon for materials that have been included in this article.

Authors

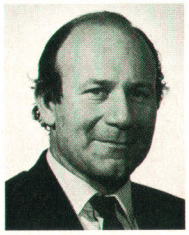
August 1985

4 Beyond RISC

William S. Worley, Jr.

After joining HP Laboratories in 1981, Bill Worley organized the original architecture and design team for the Spectrum program and directed the laboratory in which much of the early research and development for the program took place. He is now manager of HP's Information Software Operation. Bill was born in Denver, Colorado and earned MS degrees in both physics and information science from the University of Chicago in 1964 and 1967. His PhD in computer science was granted by Cornell University in 1971. Before joining HP he directed systems work at three university computer centers, taught computer science at the Illinois Institute of Technology and Cornell University, and worked in many areas of systems architecture and design for the IBM Corporation. Bill is married, has five children, lives in Saratoga, California, and enjoys reading and playing the piano. His two oldest sons are also R&D engineers at HP.

Joel S. Birnbaum



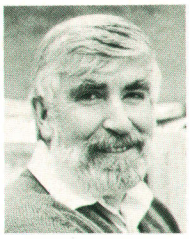
With HP since 1980, Joel Birnbaum is Vice President and Director of HP Laboratories and the originator of HP's Spectrum project. He was born in New York, New York and studied engineering physics at Cornell University (BS 1960) and experimental nuclear physics at Yale University (MS 1961 and PhD 1965). After joining the staff of the IBM Watson Research Center that same year, he worked on projects involving computer architecture, data acquisition and control systems, digital switching, and signal processing. One of his current interests is the future of computing and methods of making computers easier to use. Joel and his wife love to travel and to attend musical events. He was a college athlete and is still very interested in sports.

11 Frequency Synthesizer

19 Two-Channel Synthesizer

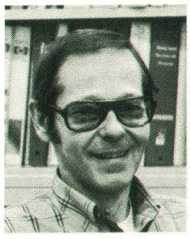
25 Active Mixer

Michael B. Aken



Born in Milwaukee, Wisconsin, Mike Aken received a BSEE degree from the University of Wisconsin in 1966 and came to HP the same year. He continued his education through the HP honors cooperative program, completing work for an MSEE degree from Colorado State University in 1969. He has worked on the development of a number of HP instruments, including the HP 3326A Two-Channel Synthesizer, and is now a project manager specializing in laboratory instrument support. He lives in Everett, Washington, is married, and has three children. He has served on the Lake Stevens, Washington vocational education board and enjoys hiking and camping in the Cascade Mountains.

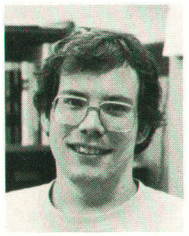
William M. Spaulding



A project manager at HP's Lake Stevens Instrument Division, Bill Spaulding has been at HP since 1969. He has contributed to the development of a number of HP instruments, including the HP 3326A Two-Channel Synthesizer, and his work in signal synthesis has resulted in two patent applications. His professional interests include frequency synthesis and RF design. Bill was born in Osceola, Iowa and received a BSEE degree from the University of Wyoming in 1967 and an MSEE degree from the University of New Mexico in 1969. He lives in Everett, Washington with his wife and three children and likes radio-controlled model airplanes, model railroading, and his home computer. He is an amateur radio operator (NA7Y) and can often be found working CW on the Extra Class bands from 10 to 80 meters.

21 Synthesizer Firmware

David A. Bartle



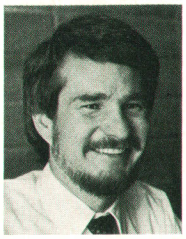
At HP since 1980, Dave Bartle wrote much of the software for the HP 3046 Selective Level Measuring System and contributed to the firmware design for the HP 3326A. He studied electrical engineering at Montana State University and was awarded a BSEE degree in 1978 and an MSEE degree in 1980. Dave was born in North Dakota, grew up in Montana, and is now a resident of Everett, Washington. He is married, has two children, and likes racquetball, golf, music, and photography.

Katherine F. Potter

Katie Potter was born in Oakland, California and studied mathematics, electrical engineering, and computer science at the University of Colorado (BA 1973 and BS 1979). She was a high school mathematics teacher before coming to HP in 1979. Her HP experience includes work on a digital filter design and on firmware for the HP 3326A Two-Channel Synthesizer. She is continuing to develop firmware for future products and also participates in an HP-sponsored career counseling program for secondary schools. Katie and her husband, who is also an HP R&D engineer, live in Lynnwood, Washington. They are the parents of a new son. Katie likes backpacking, sea kayaking, weaving, and skiing.

30 Test Data Collection

Reed I. White



Reed White was born in West Orange, New Jersey and completed a BSEE degree from the University of Wisconsin in 1970. He continued his studies at Wisconsin until 1972, when he earned an interdisciplinary master's degree in electrical engineering, computer science, filmmaking, and art. After managing his own company for six years and working on sawmill automation for another two years, he joined HP's Corvallis Division in 1978. He currently lives in Healdsburg, California and manages the Healdsburg office of HP's Manufacturing Productivity Division. Reed is a private airplane pilot and spends much of his outside time gathering information for a book he is writing, "The Western Flyer's Backpacking Guide."

Development of a Two-Channel Frequency Synthesizer

Combining two independent synthesizers, flexible modulation, and control circuits in a single package, this instrument can generate two-phase, two-tone, pulse, frequency hopping, and swept signals.

by Michael B. Aken and William M. Spaulding

AS THE COMPLEXITY of electronic circuits and systems has increased, the need for equally complex, high-quality test signals has grown. In response to the need for greater flexibility, Hewlett-Packard has developed the HP 3326A Two-Channel Synthesizer, Fig. 1. This instrument is based on the same fractional-N synthesis technique used in the HP 3325A Synthesizer/Function Generator,[1] which was introduced in 1978. The addition of a second synthesizer and output channel, all under the control of a common microprocessor, has resulted in a high-performance signal source that has operational features extending well beyond the direct combination of two independent signal sources.

Frequency coverage is from dc to 13 MHz with one-microhertz resolution below 100 kHz and one-millihertz resolution from 100 kHz to 13 MHz. Output levels are programmable from 10V p-p into 50Ω (23.98 dBm) to 1 mV p-p (−56.02 dBm) with 0.01-dB resolution. Independent preattenuator dc offset can be applied to the output signals. Functions available include dc-only, and there is independent control of the selected function on each output channel. The article on page 19 gives examples of the applications of the HP 3326A.

Functional Modes

The HP 3326A has four modes of operation: two-phase, two-channel, two-tone, and pulse.

Two-phase operation. In this mode, high-accuracy phase control is provided by Hewlett-Packard's implementation of fractional-N synthesis. Two synthesizers and two chains of output circuitry are combined to provide continuous phase control between the output channels over a range of ±720° with 0.01° resolution. An internal calibration system working through the instrument's controller provides fully calibrated phase between the output channels across the full frequency range of dc to 13 MHz. Waveforms may be sinusoidal, square, or mixed sine and square. The calibration system can be used to provide multichannel phase operation among several instruments for output frequencies greater than 1 kHz. Multiphase operation at lower frequencies can be achieved with the addition of a time-interval counter and a computer to act as system controller. Phase is not calibrated when using the HP 3326A's optional high-voltage amplifiers, since they are connected after the output attenuators outside the calibration path. High-voltage phase accuracy can be enhanced using an external time-interval counter such as the HP 5334A.

Fig. 1. *The HP 3326A Two-Channel Synthesizer can generate either two independent signals or a combination of the signals in its two channels. Each channel produces sine and square waves from 1 μHz to 13 MHz with separately controllable amplitude and dc offset. Pulse and flexible modulation capabilities are built-in.*

Two-channel phase performance is complemented by excellent sinusoidal signal purity. Harmonically related products are more than 80 dB below the carrier from 10 Hz to 50 kHz. Nonharmonically related spurious signals are greater than 80 dB down below 1 MHz and greater than 70 dB down from 1 MHz to 13 MHz. Phase performance specifications apply for equal and unequal levels between channels, for a sine/square waveform mix between channels, and for waveforms with dc offset.

Two-channel operation. In the two-channel mode, the HP 3326A functions as two independently programmable frequency synthesizers. Sweep time and sweep marker are the only parameters shared by both channels. Channels can be independently set to sinusoidal, square, or dc-only waveforms. The same harmonic and spurious specifications apply as in the two-phase mode.

Two-tone operation. In the two-tone mode, the frequency of Channel B tracks the frequency of Channel A with a programmable offset of up to ±100 kHz. Using the frequency sweep features of the instrument, tracking swept two-tone measurements may be made. Independent control of waveforms, amplitudes, and dc offsets is maintained in each channel.

Pulse generation. With two channels having precise phase control, it was a natural extension to provide pulse generation using the equivalent of a set/reset flip-flop triggered on the rising edges of the two phase-related sinusoids. As the phase of Channel B is varied with respect to Channel A over a range of 0 to 360°, pulses with precisely controlled duty cycles from 1 to 99% can be generated with a repetition frequency equal to the programmed frequency of Channel A. The complementary outputs of the square modulator are delivered to the output amplifiers to provide pulses 180 degrees out of phase on the two channels. Independently programmable dc offsets can be applied to the pulse waveforms.

Operating Features

Versatile modulation capabilities are built into the HP 3326A Two-Channel Synthesizer. Rear-panel inputs are provided for external amplitude and phase modulation of the two output channels. Internal modulation is provided by routing the signal from Channel B to the amplitude and/or phase modulation circuits of Channel A. Amplitude modulation rates up to 100 kHz and depths from 0 to 100% are allowed. Phase modulation is accomplished within the fractional-N phase-locked loops. Rates to 5 kHz and peak deviations to ±360° are possible. Modulation percentages and/or deviations are fully programmable in the internal modulation mode.

The linear frequency sweep capabilities of the HP 3325A Synthesizer/Function Generator are maintained in the HP 3326A. Except for sweep time and frequency marker, all sweep parameters are independently programmable for the two output channels. Sweep capabilities have been extended in the HP 3326A with the introduction of discrete frequency sweep (see box, page 15). In this sweep mode, up to 63 sequential sweep elements, consisting of frequency pairs for Channel A and Channel B and the dwell time before moving to the next element, can be programmed. All frequency changes are phase-continuous. Simulation of DTMF (dual-tone, multifrequency) signals, such as those used for telephone dialing systems, and FSK (frequency-shift keying) for modem testing can be implemented directly with the HP 3326A.

The power of the internal microprocessor has been used to provide extensive calibration and self-test capability. Internal automatic calibration of amplitude, phase, phase modulation angle, and dc offset is provided. Fifty tests of internal circuitry are provided in firmware to verify the instrument's condition. Many of these tests are performed at instrument turn-on. Others are accessed during service and troubleshooting procedures. In addition, calibration error constants for most parameters are stored internally and used to correct entered data to enhance the accuracy of the instrument.

Especially useful in the two-tone mode is an integral resistive signal combiner, which can be used to provide both tones on a single output connector. Low-output impedance high-voltage amplifiers for both channels are offered as an option. These amplifiers provide up to 40 volts peak to peak (into a 1000Ω, 200-pF-maximum load) at frequencies up to 1 MHz with voltage-source drive. A high-stability crystal reference oven option is also available.

Block Diagram

The functional blocks of a mix-down synthesized signal source appear twice in the block diagram of the HP 3326A Two-Channel Synthesizer, Fig. 2.

Synthesized square waves (ECL levels) from the main fractional-N synthesizer are routed to the Channel A output mixer and to the RF switch. The Channel A mixer heterodynes a level-controlled, fixed 20-MHz signal from the reference dividers with 20 to 33 MHz from the main fractional-N synthesizer in all functional modes. Block diagram flexibility stems from the RF switch, which configures the Channel B output mixer frequency scheme to provide the functional modes. Fig. 3 describes the Channel B mixing frequency schemes for the four operating modes (two-channel, two-phase, two-tone, and pulse).

In the two-channel mode, the mixing scheme of Channel B is configured like that of Channel A. Leveled, fixed-frequency 20-MHz signals are applied to the mixer low-level port, with the auxiliary fractional-N synthesizer applied to the high-level port. Channel B can then be programmed to operate as a separate synthesizer output channel over the full 13-MHz range. In the two-phase mode, the auxiliary fractional-N synthesizer is connected to the Channel B mixer low-level port. The auxiliary fractional-N synthesizer runs at a fixed frequency of 20 MHz, with a phase angle (with respect to the 20-MHz Channel A signal from the reference) that is programmable using the phase control features of the fractional-N synthesis technique. Both channels operate at the same frequency, which is controlled by the main fractional-N synthesizer across the 0-to-13-MHz output range.

Two-tone operation is provided by programming an offset frequency in the auxiliary fractional-N synthesizer. Signals of 20 MHz plus or minus zero to 100 kHz from the auxiliary fractional-N synthesizer are routed to the Channel B mixer low-level port. Tones separated by up to 100 kHz over the 13-MHz range are developed in the output channels. Tone phase can be controlled with the auxiliary frac-

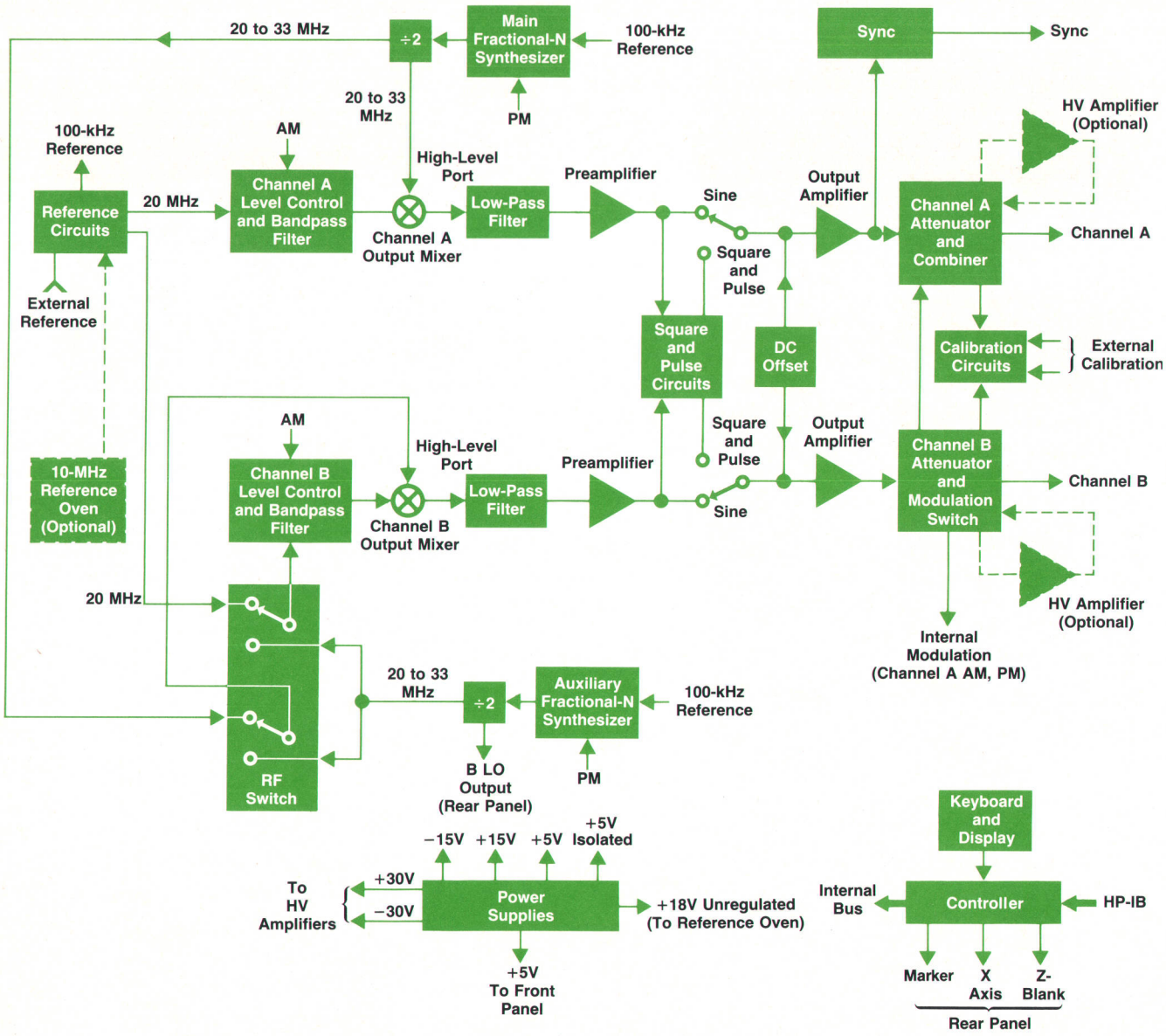


Fig. 2. *Simplified HP 3326A block diagram. The functional blocks of a mix-down synthesized signal source appear twice. Fractional-N synthesis provides high-accuracy phase control.*

tional-N synthesizer for harmonic relationships.

Pulse mode uses the same mixer configuration as two-phase mode. Preamplifier output sinusoids are shaped in the square and pulse circuits. A set/reset flip-flop, triggered by the edges of the two channels, generates the pulses, with duty cycle control established by varying the phase between the Channel A and Channel B signals. Level-controlled pulses 180° out of phase are then supplied to the output amplifiers.

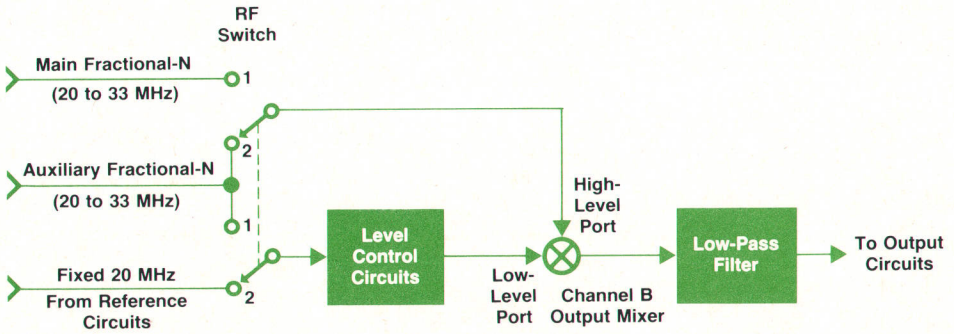
Square waves are generated by limiting the sinusoidal preamplifier outputs and applying them to level-control modulators. The modulator outputs are routed directly to the output amplifiers.

Sync circuitry subtracts any programmed dc offset from the Channel A output waveform. Fast comparators shape the signal to TTL levels, which are supplied to a front-panel connector.

Output attenuators provide 0 to 70 dB of programmable fixed attenuation for both channels. Fine amplitude control over a 10-dB range in 0.01-dB steps is accomplished with a 12-bit digital-to-analog converter in the level-control circuits for the individual channels. A resistive combiner is supplied on the Channel A output attenuator assembly to provide the convenience of two-tone outputs at a single front-panel connector. Combiner insertion loss is 6 dB and characteristic impedance is 50Ω.

External amplitude modulation signals from rear-panel connectors can be connected to the modulators for both output channels. Channel A can be amplitude and/or phase modulated by the signal from Channel B. Modulation signal switching is controlled by the microprocessor.

During calibration procedures, the attenuator outputs are disconnected from the front-panel connectors and routed to the calibration circuits.



Instrument Mode	RF Switch Position	Auxiliary Fractional-N Frequency Range
Two-Channel	2	20 to 33 MHz
Two-Phase	1	20 MHz
Pulse	1	20 MHz
Two-Tone	1	20 MHz ±(0 to 100 kHz)

Fig. 3. *HP 3326A Channel B output mixer RF signal switching.*

Fractional-N Synthesizers

Fractional-N frequency synthesis is an extremely powerful technique for signal generation with high-resolution frequency control.[1,2] It is used either directly or as an interpolation oscillator in many of the recent Hewlett-Packard swept-heterodyne analyzers and synthesized signal sources to implement local oscillators and/or output signals that are inherently stable, programmable, and easily swept. The HP 3326A is an excellent example of the versatility and adaptability of this synthesis technique.

Standard divide-by-N phase-locked loop synthesis, represented by the block diagram shown in Fig. 4, gives output frequencies that are integer multiples of the input reference frequency. High resolution in frequency demands the use of very large N numbers. However, for most applications the reference phase noise multiplication associated with large N is an unacceptable performance limitation.

Fractional-N techniques result in high-resolution frequency synthesis in a single phase-locked loop (Fig. 5, page 16). Here, the output frequency is N.F times the input frequency, where the fractional part (F) of the multiplier is composed of 12 BCD digits in the HP implementation. Phase noise multiplication is significant only for the integer part (N) of the multiplier, resulting in vastly improved performance over any other single-loop configuration with similar frequency resolution.

Since the counter output in the fractional-N loop is not an integer multiple of the reference frequency, the loop phase detector output contains a phase ramp with a period proportional to the fractional frequency offset. This ramping phase, if uncorrected, would result in very large phase modulation sidebands on the output frequency. To correct these sidebands, a digital phase accumulator in the control chip (an HP custom MOS chip) is updated with the fractional part of the frequency every reference cycle. The five most-significant digits of this accumulator feed a multiplying digital-to-analog converter (DAC), which provides a current ramp to the loop integrator to cancel the effect of ramping phase at the phase detector output. Sideband cancellation can be achieved to −130 dBc referred to the phase detector input.

Another benefit derived from the digital phase accumulator is precision control of phase in the phase-locked loop. By adding a constant to the contents of the accumulator in the control chip, the VCO phase with respect to the reference frequency can be varied with a theoretical resolution of 10 parts per million at the output frequency. This results in a phase resolution of 0.0036° using the five most-significant accumulator digits and the multiplying DAC. Phase control resolution in the HP 3326A is rounded to 0.01° for convenience.

Frequency sweep features have also been incorporated in the custom control chip. By adding a programmed frequency increment to the frequency register in the chip every reference cycle (10 μs), the loop output frequency can be swept. Flags generated on the control chip with digital comparators provide frequency marker and sweep limit indications (e.g., end of sweep).

Spectrally pure frequency systhesis is required for signals to be specified with accurate phase relationships. Spurious responses, harmonic distortion, and phase noise alter zero crossings. The HP 3326A implementation of fractional-N synthesis emphasizes spectral purity. Several circuit improvements are aimed directly at establishing performance consistency or improving performance.

Bias and API (automatic phase interpolation) current source temperature coefficients are stabilized to maximize

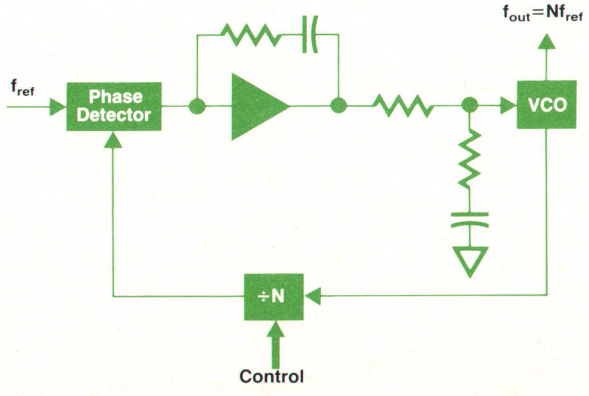


Fig. 4. *Conventional ÷N phase-locked loop block diagram.*

Discrete Sweep

Since the days of the first oscillator, user expectations have been increasing. The most important requirements are frequency accuracy, speed of frequency setting, and phase continuity. The synthesizer provided the frequency accuracy, while keyboards and the HP-IB brought local and remote programming. HP fractional-N frequency synthesis realized a glitch-free phase response. Now, the HP 3326A Two-Channel Synthesizer with its discrete sweep feature gives the user the ability to program a sequence of frequencies and the timing associated with each step of the sequence. Signals necessary for synchronization are also generated. Discrete sweep in the HP 3326A allows the user to program a series of 63 elements, each consisting of a frequency pair (Channel A and Channel B) and a dwell time. Once the frequencies and dwell times have been programmed, the user can recall or change each individual element, sweep continously through the programmed series of frequencies, or single-sweep the series. Dwell time is the interval between programming the individual local oscillators and the subsequent programming for the next element. It is limited to a range of values between 5 ms and 1000 s. Frequency settling time for the HP 3326A depends on the frequency step size. For large frequency steps the dwell time must be adjusted to allow for settling time. Although the technique is useful in the two-channel mode, its major contribution is in the other three HP 3326A modes. There, phase continuity is maintained between both of the local oscillators, phase repeatability is certain, and phase accuracy is exact at the particular frequency where the HP 3326A has been calibrated. A discrete sweep can be made in any of the four modes of operation. Since amplitude, offset, calibration, and function parameters are not stored with each discrete sweep element, the parameters used during the sweep are those in effect when the sweep is started. The operating mode chosen when the discrete parameters are programmed is remembered and cannot be changed without reentering the frequency and time parameters.

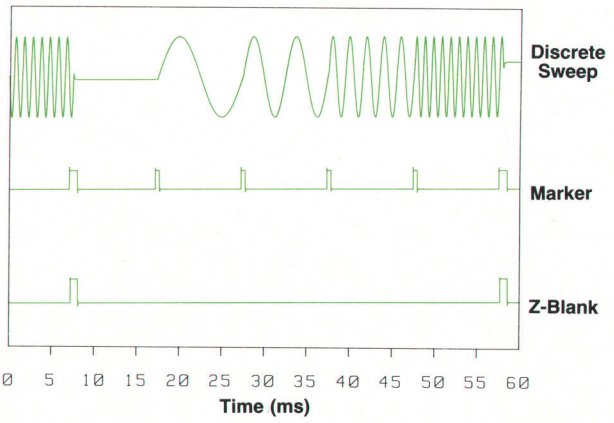


Fig. 1. *A discrete sweep with timing signals produced by the HP 3326A Two-Channel Synthesizer.*

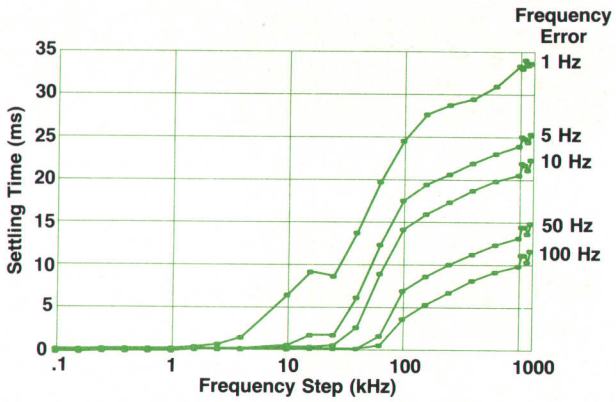


Fig. 2. *Typical settling time as a function of the size of the frequency change and allowable frequency error.*

Fig. 1 shows a discrete sweep with timing signals produced by the HP 3326A. The Z-blank signal occurs at the start of the sweep, and the marker occurs at the end of each timing cycle. The differences in marker pulse width for the first element and for the subsequent elements is caused by software timing. Fig. 2 shows typical settling times for various frequency changes.

Acknowledgments

Discrete sweep was first proposed by Dave Bartle. The idea was encouraged by Max Ramble and supported by Larry Smith.

Michael B. Aken
Development Engineer
Lake Stevens Instrument Division

loop phase stability. Drifts in bias currents translate to drifts in output phase. Drifts in API currents cause temperature dependence in phase ramp cancellation.

Monolithic operational amplifiers are used to implement the loop integrator amplifier and current summer amplifier to reduce complexity and parts count.

Reverse signal path isolation is important, particularly in the connection between the VCO and the counter circuitry, and in the counter-output-to-phase-detector connection. If injected into the VCO or phase detector, many of the counter and control chip signals can cause additional spurious responses.

An ECL flip-flop at the VCO output divides the synthesizer output frequency for a 6-dB improvement in phase noise and spurious perfromance, and reduces local oscillator even-order harmonic content to the output mixers to reduce generation of even-order mixer spurious products. The benefits derived from division of the fractional-N output were viewed as more important than the resulting frequency range restriction.

High-Dynamic-Range Output Circuitry

Mix-down signal sources generate output frequencies by heterodyning the RF from the synthesizer with a fixed frequency. To provide good harmonic distortion and spurious performance, mixing is usually done at very low levels (typically 2 to 10 millivolts) resulting in limited signal-to-noise ratios. Large amplifications are then required to provide output levels in the 10V-peak range. The HP 3326A takes advantage of a high-level active mixer which accepts 100-mV peak-to-peak input levels at the low-level port. Harmonic distortion and spurious products are typically

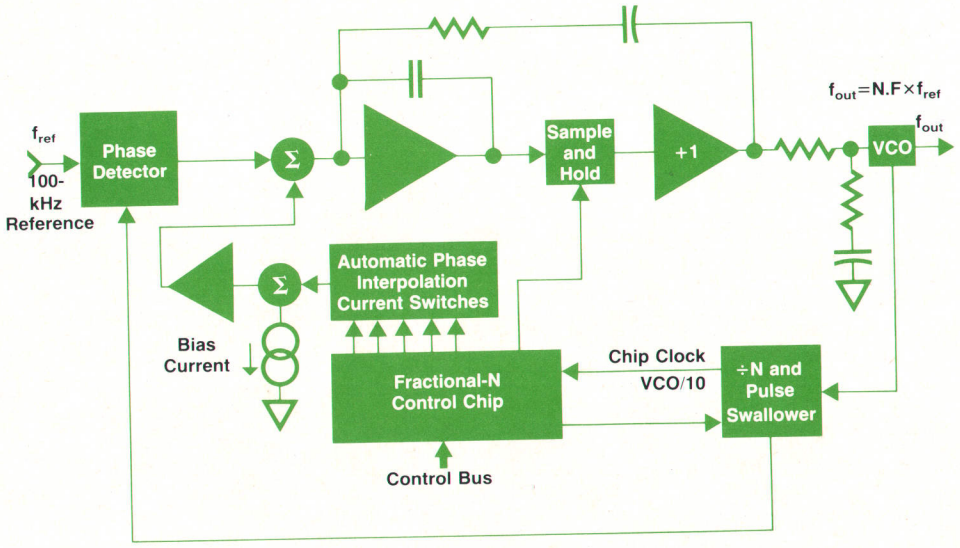


Fig. 5. *Fractional-N synthesizer block diagram.*

below −85 dBc across the 13-MHz band at the output of the mixer (see article, page 25). Active circuits also allow conversion gain, resulting in a mixer output level of approximately 100 millivolts. Subsequent amplification requirements are greatly reduced.

The active output mixer, based on a Gilbert cell multiplier configuration, is implemented with a discrete design to establish control of circuit parameters. Using this approach, even-order spurious mixing responses, which lie in-band for output frequencies greater than 7 MHz, are reduced to imperceptible levels. The active approach also helps maintain an acceptable signal-to-noise ratio.

The RF switches are implemented using current-controlled diode switches with ECL gates as buffer amplifiers. Typical switch isolations of 70 dB are achieved using standard printed circuit layout techniques. To reduce crosstalk and channel isolation further, unused signals in various modes are switched off at the source, again using ECL gate buffers.

Output amplification in the HP 3326A is implemented using high-performance, wideband operational amplifiers in parallel with feed-forward discrete ac amplifiers. Excellent distortion performance at low-frequencies is achieved, with good slew rate performance for higher-frequency rectangular waveforms. Care had to be exercised to tailor the amplifier frequency responses. Overall loop gain has to be high as the ac circuit gain increases to avoid aberrations in phase and amplitude responses of the overall amplifier. Slew rates better than 1300 volts per microsecond are achieved, and distortion is below −80 dBc up to 50 kHz with full scale signal levels of 20 volts peak-to-peak at the output amplifier (10V p-p into 50Ω, 50Ω back-matched voltage source).

Output Attenuation and Signal Combiner

Output attenuation is accomplished with relay-switched pi network pads. Steps of 10, 20, and 40 dB are provided for a total of 70 dB of fixed attenuation. With the 10 dB of high-resolution attenuation from the leveling circuitry, a total attenuation range of 80 dB is achieved.

Relay switches are provided on both attenuator assemblies to transfer signals from the output connector to the calibration circuits. By disconnecting the output signals from the front-panel connector, calibration is accomplished without being affected by user loads. To maintain source termination, the 40-dB pad is switched in during calibration.

The Channel B attenuator contains a switch to transfer Channel B signals from the front-panel connector to the internal modulation circuitry (PM on the main fractional-N circuit and AM on the Channel A level control circuits).

Instrument Controller and Power Supplies

The instrument microcomputer/controller is based on a 68B09 microprocessor. Memory is allocated as follows: 58K of ROM for program storage, 2K of nonvolatile RAM for saved instrument states, 2K of scratchpad RAM for stack space and program use, and 2K of memory mapped I/O for instrument control. The clock rate is 8 MHz. Support circuitry includes a programmable counter/timer to provide the one-millisecond interrupts for front-panel refresh and keyboard read, and a 9914 HP-IB interface chip to handle bus protocol and I/O. Nonvolatile RAM, backed up with a lithium battery, is included to provide nine saved instrument states. An X-drive DAC, also driven by the programmable counter/timer, provides an X-axis output proportional to sweep rate. Marker and Z-blank functions round out the sweep signal set.

Because of the interference generated by high-level logic signals, and the interference conducted as the internal bus picks up other signals in the instrument, the internal instrument bus is pulled low actively whenever the bus is not in use. This provides additional isolation by reducing bus impedance and the pickup of internal electric fields caused by higher-impedance phenomena.

Very clean, linearly regulated power supplies provide +5, +15 and −15 volts to the instrument. A separate +5V isolated supply is used to maintain HP-IB ground isolation. Auxiliary ±30V supplies power the optional high-voltage amplifiers, and an unregulated +18V provides standby power to a separate regulator on the optional reference oven assembly.

The power supplies are implemented using active linear regulation based on operational error amplifiers for low noise and high loop gain. The supply reference voltage is generated from a heavily decoupled, temperature compen-

(continued on page 18)

Two-Channel Synthesizer Phase Calibration

The calibrator of the HP 3326A Two-Channel Synthesizer provides not only the amplitude and offset accuracy enhancements that users have come to expect from Hewlett-Packard, but through the use of phase calibration, it also provides new levels of wideband phase accuracy. The combination of high phase resolution and phase correction techniques allows fast, repeatable programming of in-phase signals or phase offsets up to 720° in 0.01° steps between the output channels.

Fig. 1 shows phase accuracy as a function of frequency in the HP 3326A. The accuracy is limited on the low end by phase noise in the HP 3326A and on the high end by the residual time uncertainty limitations of the calibrator.

Phase calibration accuracy improves with identical waveforms (sine/sine or square/square), higher amplitudes, equal amplitudes, and midrange frequencies. At lower frequencies the results are more susceptible to noise. This noise disturbs the crossover points and makes it more difficult for the phase algorithm to converge. Phase compensation of the modulators and attenuator allows the instrument to calibrate phase in the areas of highest accuracy and then modify the phase for the frequency (attenuator compensation) and amplitude (modulator compensation) that are programmed. Phase matching of all cables, both internal and external, is necessary for high-frequency phase calibration.

The problem of calibrating phase accurately over four decades of frequency and 20 dB of dynamic range is complicated by several considerations. Among these are cable phase match, calibrator channel and offset match, phase null accuracy, indirect phase shifts in attenuators and modulators, ground loops and crosstalk, and noise, harmonic, and/or spurious content on the two waveforms.

The phase calibrator is designed to minimize the errors encountered in phase calibration through both hardware and software techniques. The basic calibrator block diagram is shown in Fig. 2. The technique used for phase comparison is not new, but it is applied here to wideband signals instead of the usual narrowband IF signals. The input switching connects the calibrator input to ground, or connects the Channel A and Channel B signals to measure the phase of Channel A with respect to Channel B, or vice versa. Zero-crossing detectors combine both ac and dc feedback to produce a square wave with hysteresis for both high-frequency and low-frequency input signals. The phase detector is alternately set and reset by the outputs of the zero crossing detectors. Outputs of the phase detector have a dc level that is proportional to the phase difference of the two signals. The low-pass filter/comparator combines differential filtering with a preset detection point to determine when the relative phase passes through 180°. The gain of the comparator allows

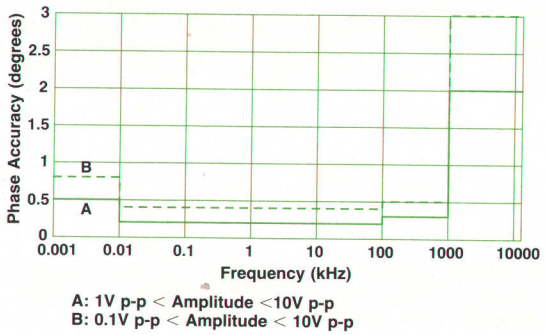


Fig. 1. *HP 3326A Two-Channel Synthesizer phase accuracy specifications.*

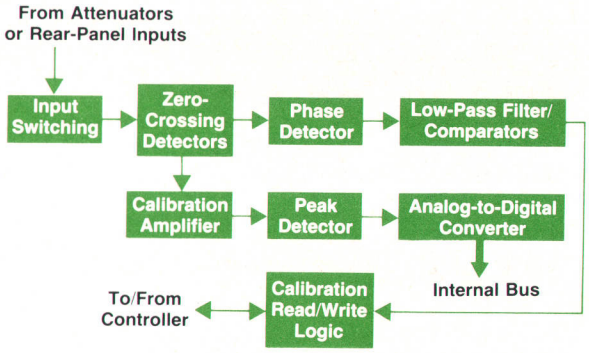


Fig. 2. *Phase calibrator basic block diagram.*

the calibrator to detect phase changes on the order of 0.003°. A comparator state vector signals phase crossover to the internal microprocessor. The HP 3326A firmware controls both the settling time of the filter/comparator and the phase step size. Once the comparator is set to a known value, the phase offset is stepped in smaller and smaller alternating increments until the exact 180° point is known. The signals to the calibrator are then reversed in the switching network and the procedure is repeated. One half of the difference between the two phase offset readings is the calibrated phase relationship. Many of the calibrator phase errors cancel in this procedure.

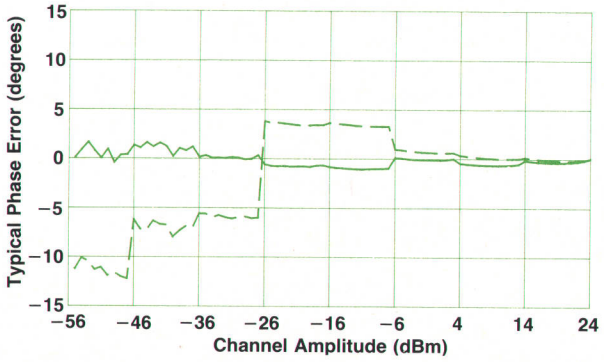


Fig. 3. *HP 3326A attenuator phase shift. Dashed line: uncalibrated. Solid line: corrected.*

Verification of the accuracy of the phase calibration system requires the ability to measure the calibrated outputs of the HP 3326A. This is done with the HP 3577A Network Analyzer and an HP 9000 Series 200 Computer. This testing requires the removal of systematic errors through matched cables and the substitution of a power splitter. Measured errors are cancelled by the controller. Resolution , stability, and accuracy are measured within 0.05°. A counter measures the phase relationships of the square wave so that the optimum values can be maintained in the presence of symmetry errors.

Fig. 3 shows the uncalibrated and corrected phase shifts of the attenuators.

Michael B. Aken
Development Engineer
Lake Stevens Instrument Division

(continued from page 16)
sated Zener diode, Load regulation for the 15V supplies is typically within 10 μV at the sense point for a 2A change in load current. Ripple at 120 Hz is typically no more than 10 μV. Requirements for power supply rejection in the critical circuits, particularly in the synthesizers and low-level portions of the output circuits, are greatly reduced.

Packaging Techniques

Packaging for an instrument with 80 dB of dynamic range presents a considerable challenge. Circuit isolation demands the use of tightly sealed individual circuit compartments for critical analog functional blocks, and for the digital interfaces to the controller. The HP 3326A uses a relatively inexpensive aluminum sand casting. Flash is cleaned from the casting, and the top and bottom surfaces are machined, drilled, and tapped for close seals to the six-layer motherboard and the compartment top covers. Circuits less susceptible to internal EMI, such as the power supply and controller, are packaged on standard printed circuit boards with edge-guide mounting and motherboard connectors. Sensitive RF signals are routed across the top of the cast card nest in double-shielded cables. Ribbon cables route digital signals to the front-panel display and keyboard, and to the rear-panel HP-IB connector.

An additional packaging constraint is imposed by the requirement for dc isolation between the inner chassis and the instrument side frames and outer case. Two separate ground systems are required for the motherboard, and insulators are installed between the card nest and the side frames. Separate grounds maintain HP-IB isolation and reduce multiple signal current return paths, which lead to amplitude errors at low levels (classic attenuator problem).

Acknowledgments

As with most complex design projects, the list of contributors to the success of the project is a long one. Middle project phases were managed by Larry Smith and Paul Thomas. Larry Smith continued as sole manager after the move to Washington as the Lake Stevens Instrument Division of Hewlett-Packard split from the parent Loveland Instrument Division in Colorado. A special note of thanks is extended to Larry for an outstanding job of management.

Lee Gregory and Kurt Rentel were key designers during the initial investigation and definition phases. Andy Cassino contributed to the investigation of design improvements to the fractional-N technique. Steve Reames contributed initial designs for the power supplies and reference frequency circuits. Tim Lock was the original product design engineer.

As design team changes resulting from the divisional move stabilized, a talented project team carried out the design and implementation of the instrument. Dave Bartle, Katie Potter, and Rich Wilson designed the instrument firmware. Al Lesko had responsibility for the fractional-N system until he left to return to school. Bill Ginder completed the fractional-N design. Grant Bower was responsible for power supply and controller circuit designs. Manfred Bartz contributed the output amplification and attenuator designs. Dave Rassmussen designed the modulator, the RF switches, and the output filtering, and contributed to the output mixer design along with Bill Ginder. Curt Allen was responsible for the development of the calibration and square/pulse circuits. Andy Purcell had design responsibility for the AM and offset assemblies, and contributed to instrument evaluation and testing. Paul Gallagher and Ann Testroet were responsible for the package development in its final form. Debbie Fromholzer contributed the industrial design support for the instrument. Clark Nicholson, Joe Diederichs, and Larry Bennet developed the instrument turn-on and test system for manufacturing. It has been our personal pleasure to work with all these people during the course of the project.

References

1. D.D. Danielson and S.E. Froseth, "A Synthesized Signal Source with Function Generator Capabilities," *Hewlett-Packard Journal*, Vol. 30, no.1, January 1979.

2. Ulrich L. Rhode, *Digital PLL Frequency Synthesizers: Theory and Design*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983, Chapter 3, pp. 124-141.

Applications of a Two-Channel Synthesizer

by Michael B. Aken

THE HP 3326A TWO-CHANNEL SYNTHESIZER offers the user exceptional versatility with its combination of two frequency synthesizers, phase calibration, synchronized sweeping, and a frequency agile discrete sweep. A few examples of its applications are:

HP 3326A Feature	Applications
Multiphase Testing	Three-phase control circuit design and testing
Discrete Sweep	Sonar testing
	Two-tone rejection testing
	Dual-tone multifrequency generation
Two-Tone Mode	Communications scrambling
	Illegal tone combination generation
	Tone decoder testing
	Mixer testing
	Intermodulation distortion
Two-Phase Mode	Transducer testing
	Two-tone jitter generation
	Phasemeter testing
	Phase locking

Multiphase Testing

The ability to control and set the relative phases of three or more signals easily is important in the development and testing of control systems that demand flexible phase requirements on the inputs. The HP 3326A can calibrate, equalize, or offset its phase with respect to any number of other HP 3326As in a minimum number of steps. If power splitters are used, the system can be calibrated without removing any cables. For more than two HP 3326As, the user has the choice of either daisy chaining the phase reference or connecting the phase reference from a master instrument to all of the remaining instruments. The daisy chain method introduces a cumulative phase error, while the parallel method decreases the reference amplitude by 6 dB for each additional instrument.

The sources of phase error in multiphase calibration (or in external calibration) are dissimilar waveforms (square or sine), the lack of phase matching of cables used in the system, the amplitude ratio of the two signals used in external calibration, and the absolute amplitude of these signals.

Maximally accurate phase calibration using the external calibration ports requires that both the slew rates (dv/dt) and the harmonic content of the two input signals be closely matched. At frequencies below 1 kHz it is extremely difficult to provide both dc blocking in a 50Ω system and sufficient squaring-circuit hysteresis in the calibrator without affecting higher-frequency phase performance. In multiphase applications below 1 kHz, additional instrumentation is needed to provide calibrated phase performance.

Fig. 1 shows the normal connections for multiphase operation. Fig. 2 shows the additional equipment needed for operation below 1 kHz.

Logic Signal Simulation

At first, the use of a precision two-channel function generator to drive logic circuits may seem like overkill. Closer inspection reveals that the capabilities for providing worst-case analog signals or real simulation of spurious conditions are unmatched. The following are some brief examples of using the HP 3326A to exercise logic circuits:

- The absolute phase calibration between the two square outputs suggests an application for determination of the phase sensitivity of two clock systems to errors in overlap. This feature can also be used to determine setup times in edge-triggered systems.
- Its amplitude resolution and stability along with its dc offset resolution and accuracy suit the HP 3326A for worst-case testing of external TTL-compatible ports for margin requirements.
- The ability to modulate channel A precisely with channel B allows another form of worst-case testing. The use of the built-in combiner allows precise phase jitter of different values and bandwidths to be introduced at a logic input. The phase jitter in degrees peak is the arc tangent of the ratio of the two amplitudes.
- The ability to multiphase calibrate two HP 3326As in pulse or square mode allows generation of either four

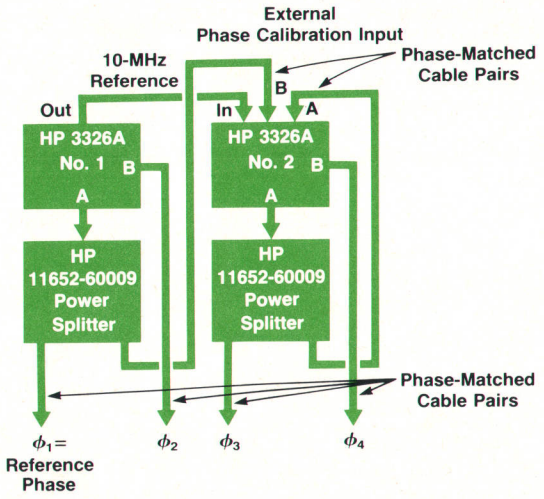


Fig. 1. *The HP 3326A Two-Channel Synthesizer can calibrate its phase with respect to any number of other HP 3326As in a minimum number of steps. The normal connections for such multiphase calibration at frequencies above 1 kHz are shown here for four-phase calibration.*

Measuring Intermodulation Distortion with a Two-Channel Synthesizer

A good example of the capabilities of the HP 3326A Two-Channel Synthesizer is a swept measurement of intermodulation distortion (IMD). Analyzing intermodulation distortion provides insight into the nonlinear characteristics of a circuit or system. Common applications are in audio or communications.

The test stimulus or driving signal is usually a composite of two closely-spaced, high-level sine waves. Fig. 1 shows the spectrum of the signal as reproduced by the circuit under test. Note the resulting distortion products. The distortion products to be measured are offset in frequency from the stimulus signals.

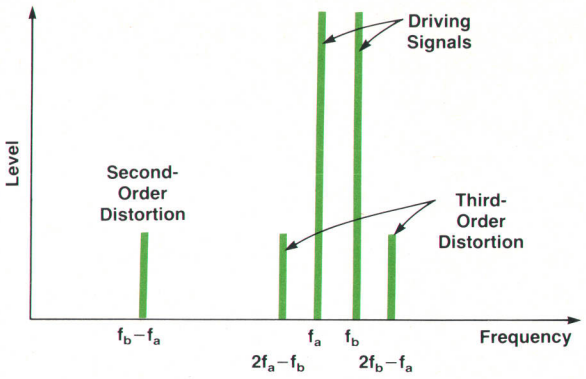


Fig. 1. *Second-order and third-order intermodulation distortion products.*

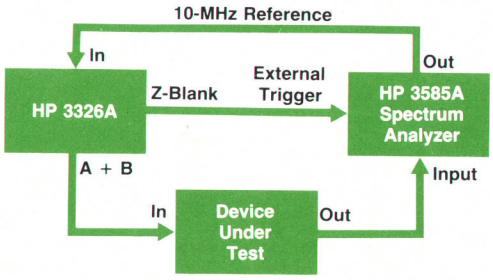


Fig. 2. *Setup for swept intermodulation distortion measurements using the HP 3326A Two-Channel Synthesizer.*

In the past, these measurements were made with two separate sources, an external signal combiner, and a spectrum or wave analyzer to select and measure the desired distortion product.

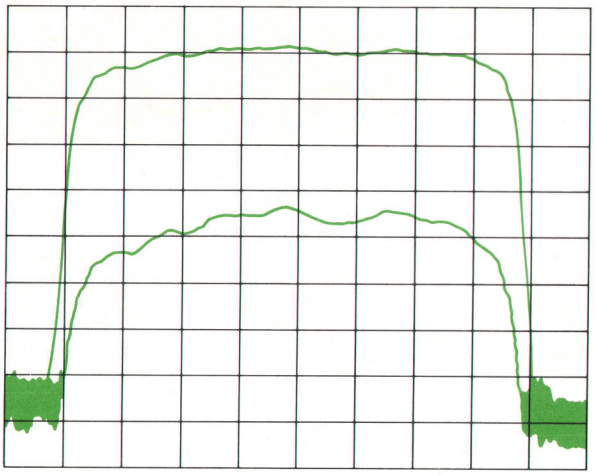


Fig. 3. *Swept third-order distortion measurement of an IF amplifier and filter. Distortion level is the difference between the reference trace (upper) and the measurement trace (lower). Vertical scale: 10 dB/division. Horizontal scale: 5 kHz/division.*

To characterize a circuit over a frequency range of interest, a series of single-frequency measurements were made by successively setting both sources and the analyzer to the appropriate frequencies. Even with programmable instruments under computer control, these measurements were rather slow.

Using the HP 3326A to produce the stimulus signal and the HP 3585A Spectrum Analyzer to measure the response, the measurements are simpler and faster. The HP 3326A has a two-tone mode and an built-in signal combiner to provide a sweeping two-tone signal with IMD as low as −80 dBc.

As shown in Fig. 2, the source and analyzer share a common frequency reference.

The Z-blank signal from the HP 3326A causes the instruments to start their sweeps simultaneously, with the desired frequency offsets. These offsets are maintained while the instruments sweep synchronously to produce a display like that shown in Fig. 3.

Ben Zarlingo
Product Marketing Engineer
Lake Systems Instrument Division

well-defined square waves or two pulse trains.

- Discrete sweep allows a sequence of repeatable square waves or pulses to be used as inputs where repetitive well-known frequencies are required. An example of this is the generation of a pseudorandom set of frequencies to excite a logic system.

The HP 3326A is not limited to square waves in logic testing. Sine waves can be used to test comparators and Schmitt triggers. The HP 3326A can be used to test for amplitude-modulation-to-phase-modulation conversion and hysteresis. Discrete sweep can be used to characterize the delay and response time of comparators.

Communications Testing

The combination of two sources and the ability to calibrate amplitude, phase, and modulation allow the testing and simulation of a large variety of communication devices.

In the world of low-cost instrumentation, the user's expectation for modulation capability of instruments demands that the modulation port be available, and that some nominal full-scale input sensitivity and bandwidth be specified. The HP 3326A, with its calibrated amplitude and phase modulation, offers modulation functions whose specifications are exceeded only by extremely expensive test equipment. Not only is the accuracy exceptional, but the resolution, frequency flexibility, and remote control capability of modulation in the HP 3326A make this instru-

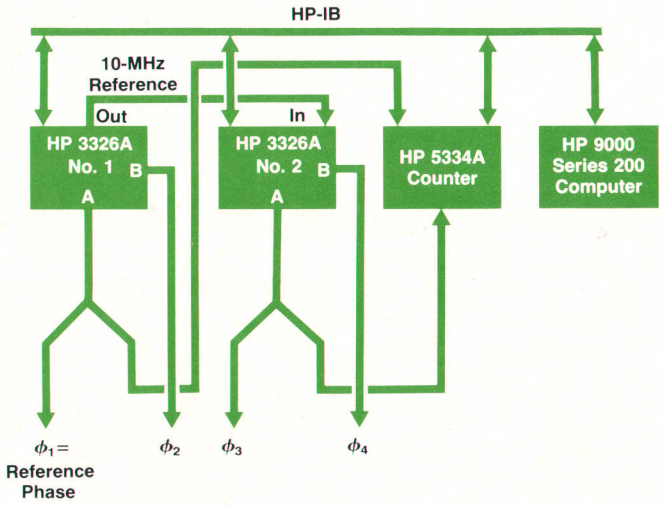


Fig. 2. *Connections for four-phase calibration below 1 kHz, showing the additional equipment needed.*

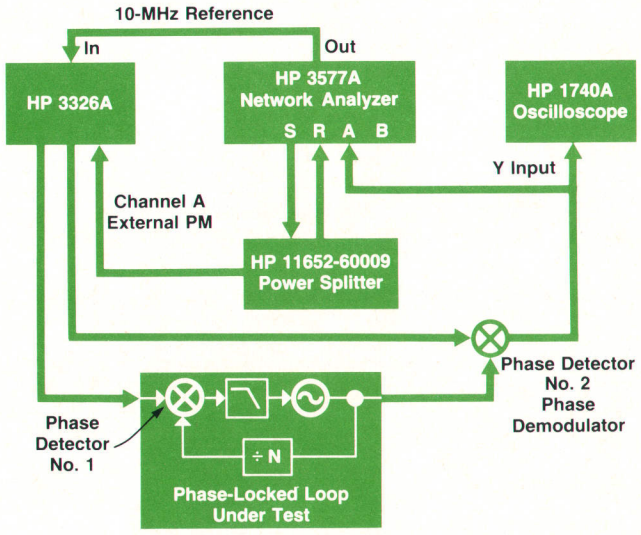


Fig. 3. *HP 3326A simplifies the testing of phase-locked loops and control loops. Shown here is a setup for frequency and phase response measurements.*

ment stand out in high-quality systems applications.

The combiner provides the stimulus for intermodulation tests, and when one or both channels are sweeping , it also provides unequaled ability to look at real-time intermodulation distortion products (see box, page 20). For phase jitter testing through the use of two tones, the HP 3326A can provide calibrated sine waves with a known amount of phase jitter. In this case (two combined tones) the signal also contains a modulation component, unlike the constant amplitude of the phase modulated signal.

The combination of linked sweep, phase calibration, and various output functions can allow previously unavailable mixer and phase detector testing in noncontroller environments. When both channel synthesizers are sweeping in the same direction, phase repeatability and single-point accuracy are maintained. Standard sweep with marker is available for conventional filter testing. Discrete sweep allows real-time settling-time measurements of filters.

Control Loop Testing

Although applications in testing of control loops and phase-locked loops have existed a long time, the HP 3326A brings simplicity of operation to these measurements. This is because the HP 3326A can generate two different outputs whose phase can be calibrated and offset. Referring to Fig. 3, this allows the demodulator #2 signal to be generated and set up with a mininum of effort. The use of the HP 3577A Network Analyzer[1] as a detector allows both the amplitude and the phase of the loop to be measured.

Reference

1. R.A. Witte and J.W. Daniels, "An Advanced 5-Hz-to-200-MHz Network Analyzer," *Hewlett-Packard Journal*, Vol. 35, no. 11, November 1984.

Synthesizer Firmware for User Interface and Instrument Control

by David A. Bartle and Katherine F. Potter

INSTRUMENT FIRMWARE (software in read-only memory) performs three major functions in the HP 3326A Two-Channel Synthesizer. First, it implements the user interface presented by the front panel and the HP-IB (IEEE 488) remote control bus. Second, it provides all of the control algorithms and signals for the internal circuitry. Third, it enhances the instrument's performance with extensive self-calibration and self-test capabilities.

Running this firmware is a 68B09 8-bit microprocessor with 64K of address space apportioned as follows: 58K in ROM, 4K in RAM (including 2K of nonvolatile RAM), and 2K for I/O. The primary language used for the firmware is Pascal adapted for use in an instrument control environment. Approximately 92% of the firmware (11,475 lines of

source code) is written in Pascal and the remaining 8% is written in assembly language.

Design Philosophy

The most important objective for development of the firmware was to implement a friendly and reliable user interface at the front panel and the interface bus (HP-IB). Second, during the product development, it was crucial to support the hardware development effort with firmware for control of individual circuit boards, system integration, and testing. Third, completing the firmware in a timely fashion with no known errors (bugs) was important.

The firmware design task was broken into a sequence of design steps, using the concepts of structured software design.[1] The initial task was the generation of a document detailing the proposed instrument feature set and operation from a user's point of view. This document was used as a design guide for the firmware development. It clarified the instrument's proposed operation to everyone involved in the instrument development effort. This document was carefully maintained to ensure an accurate, current picture of the intended instrument operation.

Next, a software architecture was proposed and expressed graphically as a hierarchy chart.[2] The modules in the hierarchy chart were defined by their logical function and data use. Architecture walkthroughs were conducted to refine the architecture and to ensure that it fit the needs of the definition.

With the architecture designed, the internal structures of individual modules in the hierarchy were defined and designed in a top-down order: the highest-level module was designed first, then the next-highest-level module, and so on. Each module was further decomposed into procedures (and/or functions) supporting the defined function of the module. The internal logic of each procedure was designed using the pseudocode technique.[3] Complete modules were subjected to a design review.

Finally, Pascal source code was generated for the procedures contained in each module, again in a top-down order. If possible, modules were individually tested as the source code generation was completed. Module integration and testing began with the completion of the core of the architecture, which included the EXECUTIVE, KEY, HP-IB, COMMAND BUILDER, DISPLAY, and EXECUTE modules (see firmware block diagram, Fig. 1). Other modules were initially implemented as dummy or do-nothing modules.

Firmware testing was performed both manually and by software designed for testing the HP 3326A firmware.[4] This software package is written in BASIC and runs on an HP Series 200 Computer, which controls the HP 3326A via the HP-IB. It is designed to test the entire user interface as well as parameter limits, command syntax, error conditions, and bus reliability.

Compiler

In addition to a viable design philosophy and plan for a large firmware development effort, it is important to have appropriate tools to support the design effort. A Pascal compiler was developed for the 68B09 microprocessor. This compiler runs on an HP 9000 Series 200 Computer. Several special compiler features are included to expedite the development task. Compilation of individual modules allows designers to work in parallel and reduces compiler overhead time. Floating-point math capability simplifies amplitude control and calibration algorithms. Binary-coded decimal (BCD) math supports the necessary dynamic range for frequency entry and control (1 microhertz to 13 MHz is 14 decades). A first in, first out (FIFO) buffer data structure, referred to as the message queue, allows easier implementation of communication between the EXECUTIVE, the real-time processes, and various modules.

Design Implementation

The main program, the EXECUTIVE, is responsible for initializing the instrument at power-on, processing inputs and events from other processes, responding to programming errors, checking the hardware fault register for hardware errors, and ensuring orderly state changes in the hardware.

The INITIALIZE module is used by the EXECUTIVE to perform all power-on hardware and software initialization and to set up the instrument state (frequency, amplitudes, etc.). After programming the initial instrument state, INITIALIZE performs a self-test and an instrument calibration.

The remaining portion of the EXECUTIVE is a loop which checks through an ordered list of potential activities for the one that has requested processing. This ordering causes each input to be completely processed before the next input

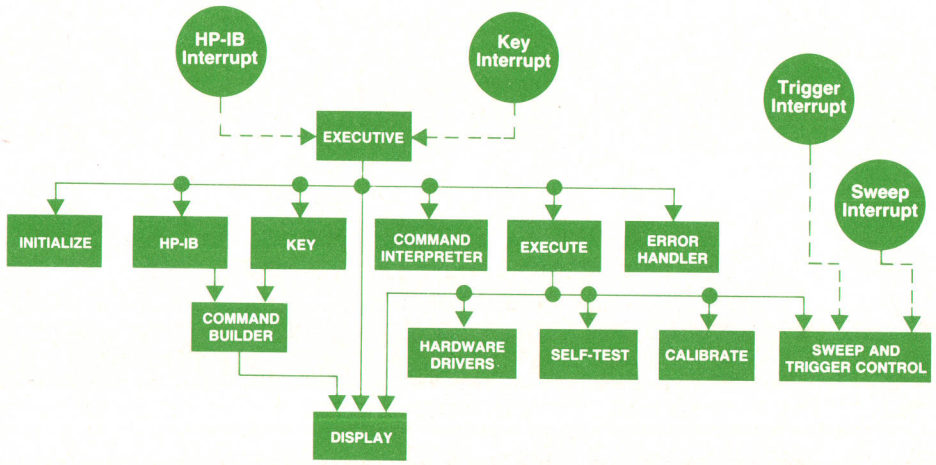


Fig. 1. *Firmware architecture of the HP 3326A Two-Channel Synthesizer.*

is handled.

Several interrupt processes interact with the EXECUTIVE. These are the HP-IB input process, the keyboard input process, and indirectly, the sweep process and the external trigger process. The communication between each of these and the EXECUTIVE is controlled by a special buffer, called a message queue. These queues provide synchronized exchange of data between interrupt modules and the EXECUTIVE to prevent interaction problems.

When a key is pressed or an HP-IB character is sent, the input is placed in a message queue. The EXECUTIVE recognizes the input and sends it, with the necessary input history, to the KEY or HP-IB module. These modules use the BUILD COMMAND module to return the updated input history and a complete or partial command. An entry state machine keeps track of the status of the input process.

When a complete command is generated, the EXECUTIVE sends it to the COMMAND INTERPRETER where the command is checked for programming errors. If there are no errors, a list of tasks used to implement the command is placed in a task message queue. Otherwise, an error is reported. When the EXECUTIVE finds a task in the task message queue, it calls on EXECUTE to perform that task.

If there is an error, the EXECUTIVE sends it to the ERROR HANDLER, which reports the error via the HP-IB status register and/or displays it and starts an error timer. When the timeout occurs, the ERROR HANDLER is called again by the EXECUTIVE to replace the error message with the previous display.

Calibration Firmware

HP 3326A performance is enhanced by extensive self-calibration. The output signal dc offset, amplitude, and phase are measured close to the output connectors. These measurements are used by the calibration firmware to modify the algorithms controlling those parameters to improve their accuracy. The HP 3326A calibrates signal amplitude, dc offset, phase, internal amplitude modulation, and internal phase modulation. Undesirable phase shift caused by level control circuitry and attenuators is also compensated. Altogether, the HP 3326A uses 24 calibration correction factors.

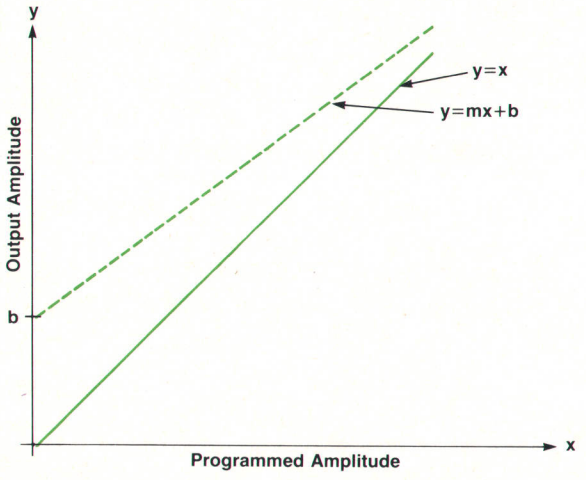


Fig. 2. *For an ideal amplitude control system, the output amplitude equals the programmed amplitude (solid line). For a real system, the response (dashed line) differs from the ideal. The HP 3326A firmware measures two points on the actual response curve, computes the slope and intercept, and corrects for system errors.*

Amplitude Calibration

The amplitude control for the HP 3326A consists of a DAC-controlled dc signal modulating the level of a 20-MHz signal. This gives a 10-dB range of amplitude control with 0.01-dB resolution. Fixed attenuation of up to 70 dB can be programmed by combining 10, 20, and 40-dB pads.

For an ideal amplitude control system, the output amplitude equals the programmed output amplitude, as shown by the solid line in Fig. 2. For an actual system, gain and loss variations, DAC offset, gain errors, and other factors yield an amplitude response described by the dashed curve in Fig. 2. The calibration firmware measures the actual response and corrects the input to the DAC (digital-to-analog converter) so that the actual response is mapped more closely into an ideal response. Two points on the actual response (dashed line in Fig. 2) are measured and used to compute the slope m and the Y-axis intercept b. These are used to compute corrections for any programmed amplitude.

The HP 3326A calibrates the peak amplitude of each channel for the sine and square functions. This includes computation of slope and offset corrections for each function, for a total of four calibration correction factors for each channel.

DC Offset Calibration

The dc offset of each channel is calibrated by computing a gain error correcton for each channel. Unlike ac amplitude calibration, there is only one dc gain correction to be computed for each channel because there is only one signal path from the dc control circuitry to the output. However, the ac signal paths contribute unwanted dc offsets that must be compensated by the dc control circuitry, and therefore several offset correction values are computed, for a total of six dc offset calibration correction factors per channel.

Phase Calibration

The HP 3326A performs three types of phase calibration: internal, external, and multiphase. Internal phase calibration calibrates the phase of Channel B with respect to Channel A at the output connectors on the front panel. External phase calibration calibrates the phase of Channel B with respect to Channel A at the external calibration connectors on the rear panel. Multiphase calibration calibrates the phase of Channel A with respect to another HP 3326A reference signal at the external phase calibration connectors on the rear panel.

Internal Amplitude Modulation Calibration

Internal AM (Channel B modulating Channel A) is calibrated by measuring the peak of the modulation envelope and comparing it to the expected value. A calibration correction factor, which is the ratio of the expected value to the measured value, is then applied to the programmed amplitude of the modulating signal, Channel B.

Internal Phase Modulation Calibration

A phase modulation constant in degrees per volt is calculated for Channel A by measuring its phase shift when a dc phase modulation reference is applied. This constant is then used in programming the amplitude of Channel B when it is used as the internal phase modulation source for Channel A.

Modulator Phase Correction

The sine modulator exhibits a side effect of a phase change when the modulator level is changed. The phase shift of each modulator is measured at 10 dB below full scale. This measured phase shift is used as an end point of a predetermined modulator phase-versus-level response curve. The phase error at various modulator levels is corrected by applying an opposing phase shift whose magnitude is determined from this curve. The modulator phase shift is corrected as amplitude is changed. No attempt is made to compensate for the square modulator.

Attenuator Phase Shift Correction

The attenuators used for level control exhibit a fairly linear phase shift as a function of frequency. Compensation for this phase error is achieved by applying an opposing phase shift whose magnitude is determined by linearly interpolating average attenuator phase shift values measured at 13 MHz. A value is permanently stored for each range (0 to 70 dB of attenuation). Each value represents the combination of one or more attenuators. Whenever an attenuator is changed for either channel or a phase calibration is performed, a new phase correction value is computed and applied to Channel B. This correction is only applied in the two-phase and pulse modes.

Sweep Control Firmware

The sweep control firmware is responsible for starting, controlling, and stopping sweeps, for generating markers, Z-blank and X-drive outputs, and for responding to external triggers. Once the sweep has started, the firmware is interrupt-driven. The sweep firmware can write commands to both of the HP 3326A's fractional-N frequency synthesizers simultaneously to provide synchronous sweep starts and to increase discrete sweep dwell time accuracy.

Sweep programming is table-driven. When the sweep is reset, all the critical internal frequencies (start, stop, rate, and markers) are calculated for both the sweep and the retrace. These values are placed in a table indexed by the sweep state. The main fractional-N synthesizer generates an interrupt at each critical frequency to reactivate the sweep process. At each such interrupt, the sweep state machine reprograms the fractional-N frequency synthesizers with values from the table and advances to the next state.

Each sweep type—ramp, triangle, and discrete—has a different state machine. Ramp sweep, for example, cycles through the following states: sweep from start to marker, sweep from marker to stop, retrace from stop to start. If there is no marker, then ramp sweep alternates between sweep from start to stop and retrace from stop to start.

Synchronous sweep start of both the Channel A and Channel B outputs is possible with a combination of hardware and firmware. The hardware provides the capability to send instructions to both fractional-N synthesizers simultaneously. To start the sweep synchronously, the firmware enables this synchronous loading, preloads the start sweep instruction, and at the appropriate time, loads the instruction to both channels. Because both synthesizers are operating with the same reference clock, their responses are synchronous.

The discrete sweep firmware also takes advantage of synchronous loading. After programming each frequency step, the firmware prepares for the next step by preloading the Channel A and Channel B frequency values. Then the synchronous instruction loading is enabled and the frequency-to-output instruction is preloaded. When the dwell time has passed, the frequency-to-output instruction is immediately executed by both the Channel A and Channel B fractional-N frequency synthesizers. This overlapping of programming and simultaneous frequency changing allows the duration of discrete sweep elements to be very close to the user-programmed dwell time.

HP-IB Operation

The HP 3326A has two different modes of HP-IB data transfer, buffered and nonbuffered. In the buffered mode, each character transferred over the bus is placed in a buffer, and another character can be transferred immediately. Control is not returned to the EXECUTIVE until the buffer is full or no more data transfer is attempted by the controller. In the nonbuffered mode, after each character is transferred, control is returned to the EXECUTIVE and further processing may occur. The buffered transfer has the advantage of minimizing the data transfer time between a controller and the HP 3326A; this is useful if several instruments are being controlled simultaneously, as in an automatic test system environment.

Acknowledgments

Larry Smith provided the leadership to get the firmware development organized and under way. Ken Snyder was responsible for the 68B09 compiler and assembler development and support. Rich Wilson contributed many good ideas to the overall software architecture and implementation, as well as writing the HP-IB, EXECUTE, and SELF-TEST modules. Many thanks to Andy Purcell for his contributions to the development of the firmware test software and to Linda McGee who contributed many hours of firmware testing.

References

1. W.P. Stevens, *Using Structured Design*, John Wiley and Sons, New York, 1981.
2. M.L. Gembarowski, "Software Methodology Preserves Consistency and Creativity," *Hewlett-Packard Journal*, Vol. 36, no. 3, March 1985.
3. R.S. Pressman, *Software Engineering: A Practitioner's Approach*, McGraw-Hill, 1982.
4. G.J. Meyers, *Software Reliability*, Wiley-Interscience, 1976.

A High-Level Active Mixer

by William M. Spaulding

TO PROVIDE LOW-BAND COVERAGE, most synthesized signal sources require the use of an output mixer to heterodyne RF signals down to baseband. The HP 3326A Two-Channel Synthesizer accomplishes this mixing function with a high-level, active mixer design based on a standard Gilbert cell configuration.[1]

Mixer designs based on the Gilbert cell configuration shown in Fig. 1 have been used for many years in both discrete and monolithic forms. Like diode mixers, these circuits have performance limitations, particularly in the area of balance. Active approaches also introduce excess noise, primarily because of base circuit resistances.[2,3] However, when noise considerations are treated appropriately, active designs have some very distinct advantages.

One advantage is conversion gain. Mixers distribute the input power among several frequencies. As a result, the level of the desired sideband (usually the lower sideband in a source) is 7 to 10 dB below the RF input port level; this is conversion loss. Active devices amplify these signals during the mixing process. Conversion gain in an active mixer is limited by the impedance levels (noise and filtering constraints) at the collectors of the output transistors, and by the amount of local feedback used for linearization of the low-level differential pair. Gains of 0 dB are readily achieved, resulting in an overall sideband level improvement of as much as 16 dB over a diode mixer.

Another advantage of active designs is that LO port levels can be lower than those in a diode mixer. Active bipolar differential pairs can be fully switched with signal levels of approximately 100 millivolts peak-to-peak across the input.[4] This corresponds to a level of approximately −10 dBm. Even if the bases are overdriven by 6 dB, LO amplification requirements are eased by as much as 20 dB. For a fixed port balance, feedthrough of LO power to the output is similarly reduced.

A third advantage is that active mixers isolate reflected power, which is always a problem in diode mixers. With a current-source-driven resistive structure at the IF port of an active mixer, the source match can be very good. The filter reflections are terminated in these resistors, and little energy is reflected back into the active devices or into the filter to be rereflected.

For these reasons, and because the frequency scheme of the HP 3326A Two-Channel Synthesizer is well-suited to an active approach, an active mixer was selected for the instrument.

Theoretical Considerations

In the generalized Gilbert cell multiplier shown in Fig. 1, mixing action results from periodically reversing the currents in the output collectors by switching the cross-coupled differential pairs at the LO rate, so the RF signal is multiplied by ±1 at the LO rate. Fig. 2 shows a circuit diagram and a simplified representation of a doubly balanced diode mixer. The action of the LO-driven diodes is an alternate connection of ground (from the center tap of the LO transformer) to opposite ends of the RF transformer T2. Current in the RF transformer is reversed at the LO rate, and the RF signal is multiplied by ±1.[5,6]

In the idealized case for either mixer, for a perfect LO square wave and no dc offset on the RF signal, the following mathematical analysis applies. For an LO square wave of unity amplitude and no dc offset:

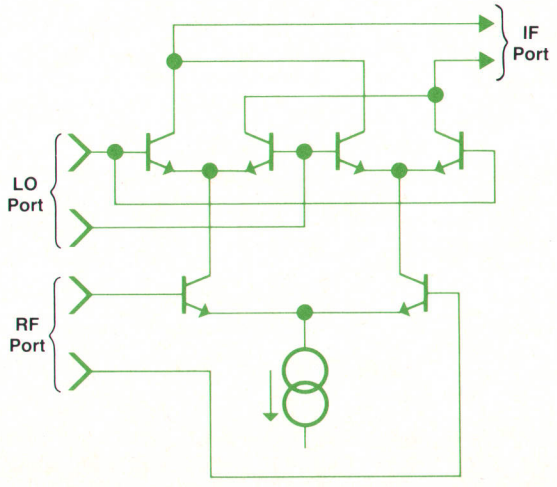


Fig. 1. *Basic Gilbert cell multiplier. Bias networks are not shown.*

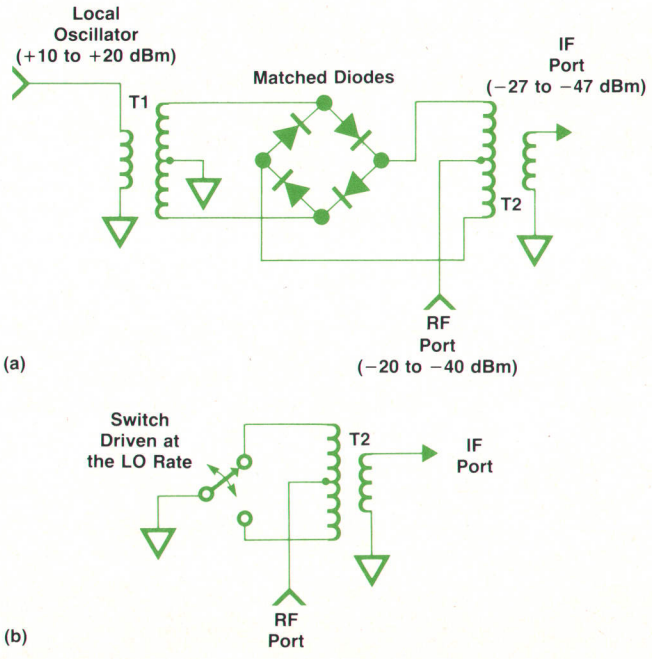


Fig. 2. *The classic diode ring mixer (a) and its equivalent circuit (b).*

$$f_{LO}(t) = \sum_{n=-\infty}^{\infty} \frac{2j}{n\pi} \sin^2(n\pi/2)\, e^{-j2\pi n f_{LO} t} \quad (1)$$

Noting that the function is odd (time axis asymmetry) and that the dc term is zero:

$$f_{LO}(t) = \sum_{n=1}^{\infty} \frac{4}{n\pi} \sin n\omega_{LO} t, \quad n \text{ odd} \quad (2)$$
$$= 0, \quad n \text{ even}$$

Multiplying by a sinusoidal function of amplitude A_{RF},

$$f_{IF}(t) = \sum_{n=1}^{\infty} \frac{4A_{RF}}{n\pi} \sin n\omega_{LO} t \cos \omega_{RF} t, \quad n \text{ odd} \quad (3)$$
$$= 0, \quad n \text{ even}$$

Since $\sin x \cos y = \frac{1}{2}[\sin(x+y) + \sin(x-y)]$,

$$f_{IF}(t) = \sum_{n=1}^{\infty} \frac{2A_{RF}}{n\pi} [\sin(n\omega_{LO} + \omega_{RF})t + \sin(n\omega_{LO} - \omega_{RF})t], \quad (4)$$
$$n \text{ odd}$$
$$= 0, \quad n \text{ even}$$

Equation 4 is the time function for the mixer IF output. Two observations can be made from this equation. First, the IF signal contains no components at the LO or RF frequencies or their harmonics for the idealized case. These signals have been balanced out. Second, the frequency spectrum consists of sum and difference products distributed symmetrically about the LO frequency and its harmonics (Fig. 3).

Now suppose that, as is usually the case in real mixers, the LO square wave is slightly asymmetrical and can be represented as the sum of the pulse and square waves shown in Fig. 4. For small asymmetry, the Fourier series for the pulse waveform is:

$$f_p(t) \approx -\frac{2\tau}{T} - \sum_{n=1}^{N} \frac{4\tau}{T}\left[\cos n\omega_{LO} t + \frac{n\pi\tau}{T} \sin n\omega_{LO} t\right], \quad (5)$$
$$\frac{N\pi\tau}{T} << 1$$

For N harmonics, the modified LO function is the linear sum of equations 1 and 5:

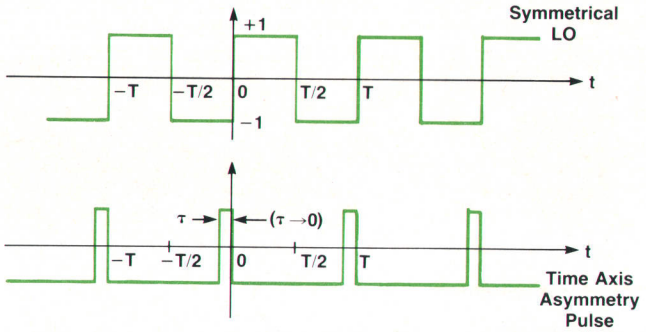


Fig. 4. *An asymmetrical LO waveform can be represented as the sum of the square wave and pulse train shown here.*

$$f_{LO}(t) \approx -\frac{2\tau}{T} + \sum_{n=1}^{N} \left[\left(\frac{4}{n\pi}\right)\left(\sin^2\left(\frac{n\pi}{2}\right) - \left(\frac{n\pi\tau}{T}\right)^2\right) \sin n\omega_{LO} t - \frac{4\tau}{T} \cos n\omega_{LO} t\right] \quad (6)$$

At the IF port:

$$f_{IF}(t) \approx -\frac{2\tau A_{RF}}{T} \cos n\omega_{LO} t + \sum_{n=1}^{N} \left[\frac{2A_{RF}}{n\pi}\left(\sin^2\left(\frac{n\pi}{2}\right) - \left(\frac{n\pi\tau}{T}\right)^2\right)\{\sin(n\omega_{LO} + \omega_{RF})t + \sin(n\omega_{LO} - \omega_{RF})t\} - \frac{2\tau A_{RF}}{T}\{\cos(n\omega_{LO} + \omega_{RF})t + \cos(n\omega_{LO} - \omega_{RF})t\}\right] \quad (7)$$

Considering the odd terms from the series:

$$f_{IF}(t)\Big|_{n \text{ odd}} \approx \sum_{n=1}^{N} \left[\frac{2A_{RF}}{n\pi}\left(1 - \left(\frac{n\pi\tau}{T}\right)^2\right)\{\sin(n\omega_{LO} + \omega_{RF})t + \sin(n\omega_{LO} - \omega_{RF})t\} - \frac{2\tau A_{RF}}{T}\{\cos(n\omega_{LO} + \omega_{RF})t + \cos(n\omega_{LO} - \omega_{RF})t\}\right] \quad (8)$$

Combining quadrature terms and dropping negligible terms since $n\pi\tau/T << 1$:

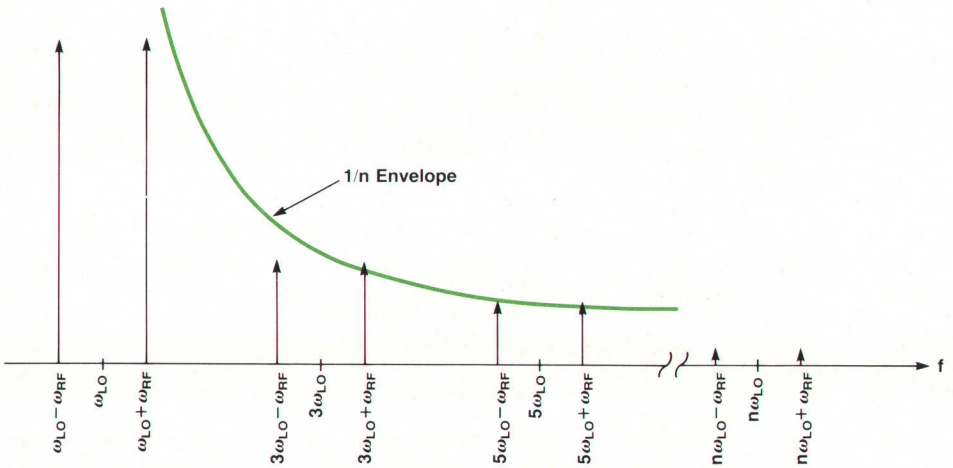


Fig. 3. *Spectrum of the mixer IF signal in the ideal case (linear scales).*

$$f_{IF}(t)\Big|_{n\,odd} \approx \sum_{n=1}^{N} \left[\frac{2A_{RF}}{n\pi}\sqrt{1+\left(\frac{n\pi\tau}{T}\right)^2}\{\sin\left((n\omega_{LO}+\omega_{RF})t+\theta_n\right) + \sin\left((n\omega_{LO}-\omega_{RF})t+\theta_n\right)\}\right] \quad (9)$$

where:

$$\theta_n \approx \tan^{-1}\frac{n\pi\tau}{T} \quad \text{for} \quad \frac{n\pi\tau}{T} << 1 \quad (10)$$

Recombining all terms (with even terms from the pulse series):

$$f_{IF}(t) \approx -\frac{2\tau A_{RF}}{T}\cos\omega_{RF}t + \sum_{\substack{n=1,\\ n\,odd}}^{N}\left[\frac{2A_{RF}}{n\pi}\sqrt{1+\left(\frac{n\pi\tau}{T}\right)^2}\{\sin\left((n\omega_{LO}+\omega_{RF})t+\theta_n\right) + \sin\left((n\omega_{LO}-\omega_{RF})t+\theta_n\right)\}\right] - \sum_{\substack{n=2,\\ n\,even}}^{N}\frac{2\tau A_{RF}}{T}\{\cos(n\omega_{LO}+\omega_{RF})t + \cos(n\omega_{LO}-\omega_{RF})t\} \quad (11)$$

Three results are of interest:

1. RF signal feedthrough appears in the mixer output. The RF signal is transferred to the output attenuated by the factor $2\tau/T$.
2. The phase and amplitude of the odd product terms are modified. This is a particularly critical factor in an instrument in which the phase of the IF signal is important, such as the HP 3326A or a vector analyzer. Local oscillator symmetry is critical to maintaining performance even in ideal mixers. When the LO signal is swept over a range of frequencies (as in a source or swept analyzer), variations in asymmetry can cause ripple in the IF signal phase and amplitude response. Asymmetry in fixed LO applications results in an error that can be calibrated out of the system.
3. Quadrature sum and difference products appear around even-order LO harmonic frequencies, attenuated by the factor $2\tau/T$. For many applications, the only importance of the additional product terms is their modification of the desired sideband, or the additional filter requirements imposed to remove them.

Another complication in real mixers is that the RF signal usually has a dc offset. This corresponds to transistor base-emitter offset voltage for an active mixer. In the diode mixer, dc offset can result from either diode mismatch or from the output of a dc-coupled amplifier on the RF port. Such dc-coupled ports are often used in instruments that upconvert from baseband because of the physical size limitations of coupling capacitors, and because of transient settling times associated with large coupling capacitors.

The RF signal can be represented by:

$$f_{RF}(t) = V_{dc} + A_{RF}\cos\omega_{RF}t \quad (12)$$

Multiplying by equation 2 and simplifying:

$$f_{IF}(t) = \frac{4V_{dc}}{n\pi}\sum_{\substack{n=1,\\ n\,odd}}^{\infty}\sin n\omega_{LO}t + \frac{2A_{RF}}{n\pi}\sum_{\substack{n=1,\\ n\,odd}}^{\infty}[\sin(n\omega_{LO}+\omega_{RF})t + \sin(n\omega_{LO}-\omega_{RF})t] \quad (13)$$

Thus, when the RF signal has dc offset, LO feedthrough terms appear in the mixer output signal. The magnitude of the feedthrough terms can easily be (and often is) larger than the desired sideband term.

As a result of these considerations, careful attention to LO symmetry and to residual dc offsets is required in any high-performance mixer design.

Practical Design Considerations for the Active Mixer

Generation of typical source output levels (+25 to +27 dBm) using a diode ring mixer requires a great deal of amplification between the mixer, running with an IF level on the order of −20 to −40 dBm, and the output attenuator. With as much as 60 dB of gain, signal-to-noise ratios at the output become severely degraded unless extreme care is used to design amplification with very low noise figures. When mixing frequencies lie well below the cutoff frequencies (f_T) of the devices, well-designed active mixers can help ease output circuit gain requirements.

Drive configuration. Referring to Fig. 1, there are two obvious choices for driving the LO and RF ports of the active mixer. Single-ended drive could be used by (ac) grounding one side of the respective differential pairs and connecting the other to the signal. Balanced drive generally requires additional circuitry (i.e., transformers or differential amplifiers).

If single-ended signals are used, a voltage approximately equal to half the input level appears at the emitter junctions. Finite and nonlinear current source output impedances result in potentially large and harmonic-rich common mode signals. It has been our experience that any extra circuitry required for balanced drive pays for itself by keeping these emitter points at virtual ground, thereby minimizing common mode signal generation.

Gain control and linearization. Parameter match in active

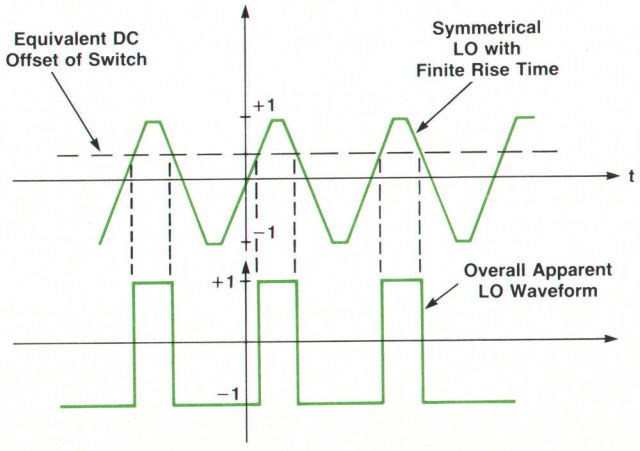


Fig. 5. *Effects of LO rise time and system offset on apparent LO symmetry.*

devices is always less than perfect. The use of local feedback in the form of emitter degeneration in the RF pair accomplishes both linearization of the devices and, once IF port collector impedances have been selected, provides control of overall conversion gain.

Any harmonic distortion generated by the RF pair is translated directly to harmonic distortion in the IF signal. Linearization of the RF pair is extremely important in the HP 3326A Two-Channel Synthesizer because of its specification of −80 dBc for harmonic distortion through audio frequencies.

Estimation of distortion levels. A practical approach to estimation of distortion levels has been developed over the years. The technique is based on peak variations in the value of r_e of the transistor. It is based on the familiar equation:

$$r_e \approx \frac{26}{I_E} \text{ for } I_E \text{ in milliamperes.}$$

For a bias current level of 10 mA, there is approximately 2.6Ω of dynamic emitter resistance. Assuming that the maximum signal current of one milliampere peak is used, the variation of r_e is:

$$\Delta r_e \approx \frac{26}{9} - \frac{26}{11} \approx 0.5\Omega$$

Each side of the differential pair experiences an excursion of 0.5Ω under signal conditions (one side goes up 0.25Ω, and one goes down 0.25Ω). Assume that the devices that make up the differential pair are matched within 1%, that is, the maximum difference in the resistance variation between devices ($\Delta r_e'$) is 0.005Ω. The change in total resistance in the emitter circuit under signal conditions is, therefore, 0.005Ω. The distortion can be estimated as follows:

$$|D| = 20 \log \frac{\Delta r_e'}{R_E}$$

where R_E is the total emitter resistance.

If a value for R_E of 100Ω is used, one might expect to see distortion products at approximately −86 dBc. No insight is provided as to the distribution of the harmonic energy. Although this method probably does not satisfy the purist, we have found it to be a reasonably effective predictor for small nonlinearities.

Common mode suppression and overall balance. Try as one might to avoid them, common mode currents seem to show up in every design. Common mode products would be of no concern except for mechanisms that perform common-mode-to-differential conversions. These conversion mechanisms are usually highly nonlinear, resulting in reinsertion of harmonic energy. Several measures can be in-

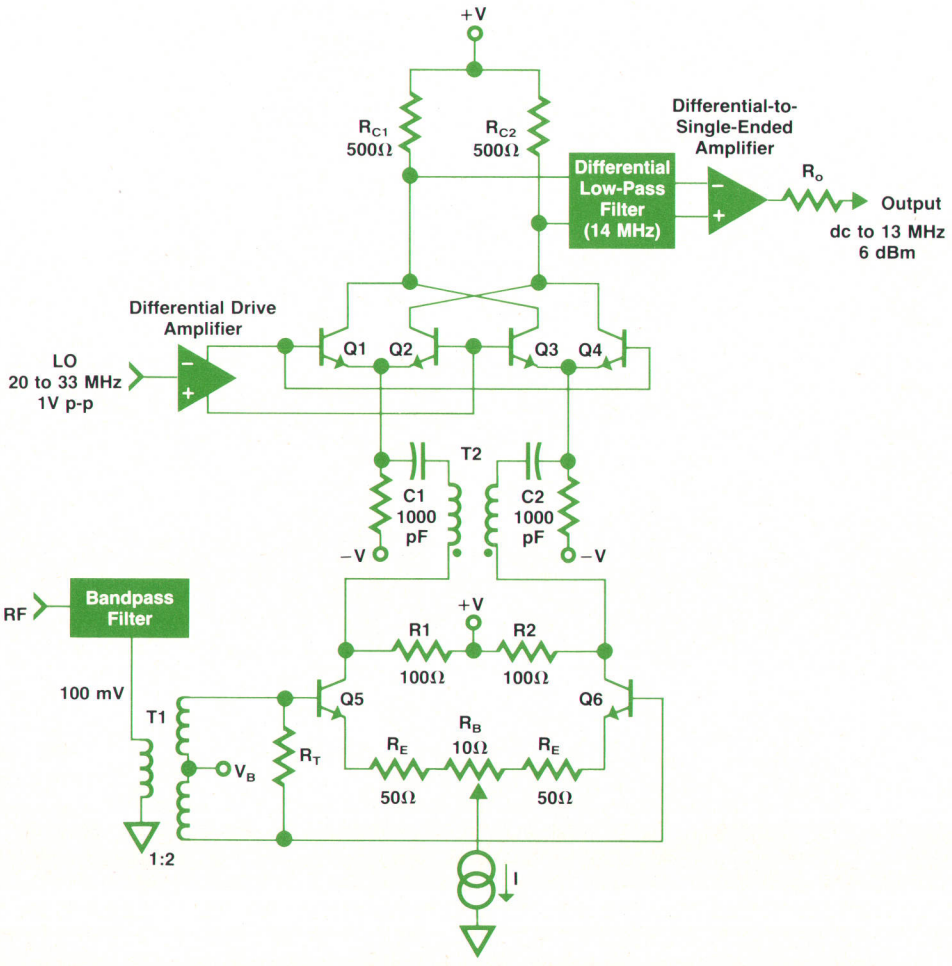


Fig. 6. *Simplified schematic diagram of the active mixer in the HP 3326A Two-Channel Synthesizer.*

voked to reduce the effect of common mode signals (see "Mixer Implementation" below).

Local oscillator rise time. Fig. 5 illustrates the effect that dc offset in switches (diode match in the ring mixer or V_{BE} match in the active mixer) has on apparent LO symmetry. For purposes of illustration it is assumed that the switch changes state at the dc offset level shown by the dashed line. Even though the zero crossings are symmetrical in the LO signal, switch toggling at the dc offset level results in an equivalent LO waveform that is highly asymmetrical. From inspection it is obvious that as the LO rise time approaches zero, the dc offset no longer alters the times at which the switches change state, and symmetry is preserved.

Switch devices are often nonlinear as they change state.[7] If rise times are short, the influence of these nonlinear transitions is reduced, since less time is spent in transition with respect to the period.

For frequencies at which limiter amplifiers can be implemented effectively, rise time reduction should be pursued. At higher frequencies, where the LO port is driven with a sinusoid, increasing LO power has the effect of reducing the time from LO zero crossing to switch state change.

Mixer Implementation

Fig. 6 shows the schematic diagram of the active mixer in the HP 3326A. Balanced drive is used on both ports as the first measure to ensure balance and reduction of common mode problems.

Variable resistor R_B is used to adjust the balance of the linear pair (Q5 and Q6). Because overall circuit balance, as indicated by minimum RF feedthrough in the IF signal, does not necessarily occur at equal dc collector currents in Q5 and Q6, capacitors C1 and C2 are introduced to prevent a corresponding unbalance in the bias currents of the LO switches (Q1-Q4). By not disturbing the switch bias, a reduction in $2f_{RF} - f_{LO}$ spurious products of as much as 20 dB is achieved. Noting the frequency scheme, this spurious product lies in-band for LO frequencies greater than 27 MHz (output frequencies greater than 7 MHz). The final measure against common mode signals is the inclusion of balun T2.

Differential gain is set (approximately) by the ratio of the sum of the output collector resistors ($R_{C1} + R_{C2}$) to the emitter resistance ($2r_e + 2R_E + R_B$). The 500Ω resistors represent a reasonable maximum value for the combination of gain, noise, and filter realizability.

Transistors Q1 and Q2, Q3 and Q4, and Q5 and Q6 are matched RF devices with typical f_T of 1.5 GHz. They are provided with maximum V_{BE} specifications and guaranteed β match of 80%.

Performance

The following performance is achieved in the HP 3326A's active mixer.

Frequency Scheme
- LO input: 20 MHz to 33 MHz
- RF input: 20 MHz fixed-frequency, level-controlled
- IF output: dc to 13 MHz, differential signal

Harmonic Distortion (typical)
- All harmonics below −90 dBc, dc to 1 MHz at mixer collectors
- Harmonics below −76 dBc to 13 MHz
- Ultimate performance limited by differential-to-single-ended converter

Spurious Responses (typical)
- All spurious products (notably $2f_{RF} - f_{LO}$) better than −90 dBc
- Ultimate overall performance limited by output frequency reinsertion into the mixer, primarily output amplifier current pulses on power supplies

Acknowledgments

Several people contributed to the development of the active mixer. Kurt Rentel was the original design engineer responsible for development of the circuit. Dave Rasmussen and Bill Ginder subsequently contributed to the design as the instrument progressed through the various prototype stages. My personal thanks to all.

References

1. P.R. Gray and R.G. Meyer, *Analysis and Design of Analog Integrated Circuits*, John Wiley and Sons, New York, 1977, pp. 563-575.
2. M. Schwartz, *Information Transmission, Modulation and Noise*, McGraw-Hill, New York, 1959, pp. 249-259.
3. C.D. Motchenbacher and F.C. Fitchen, *Low-Noise Electronic Design*, John Wiley and Sons, New York, 1973, Chapter 4.
4. P.R. Gray and R.G. Meyer, *Analysis and Design of Analog Integrated Circuits, Second Edition*, John Wiley and Sons, New York, 1984, pp. 194-197.
5. D.G. Tucker, "Some Aspects of the Design of Balanced Rectifier Modulators for Precision Applications," *Journal of the Institute of Electrical Engineers*, Vol. 95, Part 3 (Radio and Communication Engineering), no. 35, May 1948, pp. 161-172.
6. H.L. Krauss, C.W. Bostain, and F.H. Raab, *Solid-State Radio Engineering*, John Wiley and Sons, New York, 1980, pp. 194-201.
7. H.P. Walker, "Sources of Intermodulation in Diode-Ring Mixers," *The Radio and Electronic Engineer*, Vol. 46, no. 5, May 1976, pp. 247-255.

Automated Test Data Collection for IC Manufacturing

Collecting, storing, and analyzing data from a variety of test equipment and CPUs that use different formats, languages, and protocols is made possible by this software product for HP's Semiconductor Productivity Network.

by Reed I. White

UNLIKE MOST OTHER manufacturing operations, where the quality of a product can be checked at each step, the ultimate success of a manufacturing run of integrated circuits can only be determined by testing each die on a wafer at the end of the fabrication process. Consequently, IC fabrication areas devote a major proportion of their resources to keeping each parameter of the process at its specified value. If this objective is met, the process is said to be stable, or in control. If the appropriate parameters are always controlled within properly selected limits, a predictable percentage (or yield) of good ICs will be fabricated.

Unfortunately, this is not a simple task. Determining which parameters to monitor and setting cost-effective control limits are major tasks. Once the control parameters have been established, measurements and tests can create data volumes in the order of one to ten million values per week. Process engineers cannot effectively analyze data deluges of this magnitude without the help of computers.

The need for global (process through test) analysis caused IC manufacturers to be among the first users to require integration of varied software systems. In some cases, an increase in yield of only a few percent can increase yearly profit substantially. With this kind of leverage, IC process engineers easily justified the purchase of computers to facilitate global analysis.

Soon, manufacturers discovered the difficulty of coupling different computer systems and software packages. The application programs on their new computers would not talk to one another! Incompatibilities existed with hardware, protocol, CPU data representation, and application program data formats. Even with newly purchased computer systems, IC manufacturers still found it impossible to do comprehensive global data analysis.

HP's Semiconductor Productivity Network (SPN) products solve many of these problems (see Fig. 1). SPN is a large system of integrated software modules for the management of an IC manufacturing environment. Since its introduction several years ago, it has become widely used by IC manufacturers. The core module, IC-10, tracks wafers through the manufacturing process and provides status and performance reports. CA-10 is the accounting package. PL-10 provides a means for advance production planning. EN-10 allows engineering data to be collected manually. EA-10 provides a data base for manufacturing and engineering data and a number of sophisticated data analysis tools (see box on page 32). All of these modules operate on an HP 3000 Computer System.

The first SPN product to operate on two different CPUs was PC-10, the process automation module discussed in last month's issue.[1] PC-10 is tightly coupled to IC-10 on the HP 3000. It uses HP 1000 Computers to provide real-time interfaces to automatic processing equipment.

The TC-10 Tester Collection System described here couples tester data to SPN's EA-10 global engineering data analysis system and provides hooks for connecting to other analysis systems (Fig. 2). Since most IC testers are currently based on non-HP equipment, the TC-10 development team had to develop a spectrum of communications-related tools to allow portions of the product to be ported to any CPU. This generic set of tools is called NEXUS.

History of TC-10

Work on TC-10 began at Hewlett-Packard's largest integrated circuit manufacturing facility, one of the world's most automated. Numerous types of computers and application systems were being coupled. Analysis showed that only 10% of the programmers' time was being spent on solving the data processing problem, because 90% of their

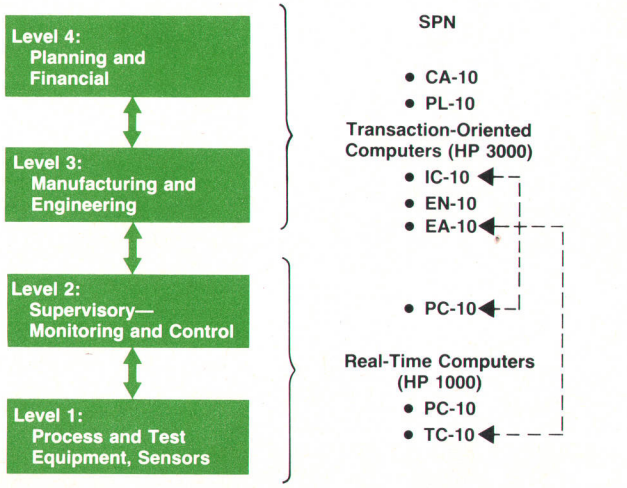


Fig. 1. *HP's Semiconductor Productivity Network (SPN) is a group of software products designed to simplify and automate complex microelectronic manufacturing areas.*

time was spent on devising ways to couple dissimilar systems. It became clear that programmer productivity could increase significantly if a generic solution to coupling could be found. The result was a prototype version of HP's Data Transport Standard (DTS), which defines a data language and its software.

The concept of a common data language seemed obvious to the designers at that time. Less obvious was the discovery that a carefully designed data language allows large complex networks of application programs to be relatively tolerant to change. Since the extreme cost of change has always been one of data processing's major problems, this has proven to be a most valuable discovery. The benefit is not unlike the same advantage discovered years ago with standard high-level programming languages, and more recently with standard hardware buses such as IEEE 802 and IEEE 488.

Later, HP's new Manufacturing Productivity Division brought together engineers representing seven of HP's IC manufacturing facilities. The fabrication areas ranged in diversity from HP's highest-volume shop in Corvallis, Oregon, to the R&D laboratories in Palo Alto, California. The engineers' task was to define the ultimate test area network: not just to define a local area test network, but to specify a fully integrated solution. Since data acquisition and analysis were considered to be of highest priority, this area received immediate attention.

About the same time, Hewlett-Packard purchased Software Management Corporation, the creators of an integrated software package for use in IC manufacturing environments. With this acquisition, HP gained a foundation on which to build. The package included an engineering analysis system, EA-10. Shortly thereafter, work began on TC-10, the software that would move test results from different types of test equipment to EA-10.

The problem was complex. Numerous types of HP and non-HP computers could be members of the same network. At the hardware and system levels, data communications between CPUs of different manufacture is, at best, difficult. Numbers and text are represented differently on different machines. File systems are often incompatible. At the application level, each tester uses a different scheme for encoding data. To be useful, many TC-10 utilities would need to function on any testing system, and be callable from any of the following languages: assembly language, Ada*, BASIC, C, COBOL, Fortran, and Pascal.

To keep the project manageable, and to avoid the mistake of intermingling application solutions with datacom solutions, the project was divided into two parts: the NEXUS software tools and TC-10.

The software tools portion dealt with generic communications issues, such as "How does one enable different application programs, coded in different programming languages, on different machines, with different operating systems, to communicate?" It soon became clear that the design and implementation of a data language would be the design team's first contribution. Fortunately, a general-purpose data language had been in use since 1980 at HP's Corvallis IC manufacturing facility. The NEXUS team was able to use the experience gained from this early prototype to design a commercially viable product. Meanwhile, engineers on the TC-10 half of the project were developing the application-specific portion of the solution, dealing with such issues as transaction design and data base architecture. Like the NEXUS team, TC-10 designers also benefited from experience. Project members had previously designed and coded custom interfaces for applications ranging from test areas to facility-wide engineering data bases. Armed with their practical experience and wish lists from the seven HP IC plants, they worked with the NEXUS engineers and HP's Semiconductor Productivity Network groups to design a unified solution.

Product Development Strategy

TC-10 was being designed to be interfaced easily to a tester of any manufacture. It became clear that this difficult objective could be achieved by the development of suitable data standards, and by total commitment to an open system philosophy. Data languages of sufficient versatility did not exist at the time, and no industry-wide test result standards were in use. Since these were needed for the foundation of the product, it was decided that an investment in developing a high-quality solution would be justified.

In keeping with modern design methodology, and to maintain compatibility with the ISO open system interconnect (OSI) model (Fig. 3),[2] software architecture should be cleanly layered. The software design project also was split

*Ada is a registered trademark of the U.S. government (Ada Joint Program Office).

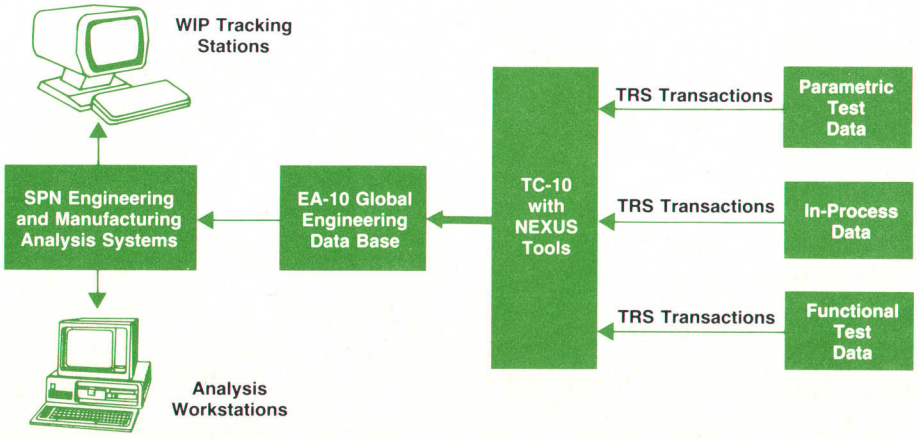


Fig. 2. *The TC-10 Tester Collection System for SPN couples data from test equipment and process monitors into the EA-10 global data base for access by other SPN modules. The NEXUS software tools included with TC-10 simplify the task of connecting and programming new equipment to communicate with SPN. TRS is the SPN Test Result Standard.*

EA-10 Data Analysis System

EA-10 consists of two parts: a data base for storing engineering and manufacturing information, and tools for extracting and analyzing the information.

The data base receives information from the IC production areas via EN-10. It receives data from IC test areas via TC-10. The design of EA-10 allows data from the IC fabrication and test areas to be correlated.

Data is extracted from the EA-10 data base with the aid of LEA, the Language for Engineering Analysis. Using LEA, a user can extract data of interest and move it to a flat-file. The flat-file facilitates the formatting of the data for input to a package of engineering analysis tools.

The analysis tools can report results in tabular form, perform statistical analysis, and produce graphic results. When used together, these tools enable engineers to concentrate their attention on process problem solving, rather than on the routine mechanics of data manipulation and statistics.

into two parts: NEXUS and TC-10. The NEXUS portion covered generic functions such as the DTS data language and communication tools designed to aid users in connecting and programming new equipment for use in SPN. TC-10 covered the application-specific tester-related functions.

The NEXUS software tools were built with a serious commitment to having no bugs. (After 16 months of use, at the time of this writing, these tools still have zero bugs.) This was achieved by built-in self-tests, and by automated testing methodology. (This care also minimized the bug count in the TC-10 application code.)

Since testers are based upon a wide variety of CPUs, the code is written in ANSI Standard Pascal and structured for portability. To facilitate portability, source code is provided, along with a suite of validation programs to verify proper functionality. Adherence to documented programming standards and extensive documentation also streamline the porting process.

NEXUS Architecture

NEXUS, the set of general-purpose tools, can be divided into two major parts: the Data Transport Standard and a high-level, guaranteed delivery network.

DTS defines a high-level, object-oriented language for representing numerous data types. The language is designed to permit flexible and efficient movement or storage of any type of data across any type of communications link, through any type of file, on any computer. DTS includes formal format specifications and the software to perform encoding, changing, and decoding of DTS data records.

Although the NEXUS networking software is in an earlier stage of development than DTS, its architecture may be of interest. It is a transaction-flow architecture, with guaranteed transaction delivery. Special spool files guarantee recoverability in the event of a system crash. Since transactions can be safely sent without waiting for acknowledgment, inoperative links or computers will not disable other links and computers that continue to operate. Data is routed according to transaction content, rather than by conventional node address. NEXUS can coexist with and use conventional networking solutions. It is well suited for loosely coupled applications operating in real-time manufacturing environments that require 100% uptime.

DTS

HP's SPN Data Transport Standard, or DTS, defines a high-level data language that functionally falls within Layer 6 of the ISO open system interconnect model. DTS provides a standardized means for encoding data between application programs written in different programming lan-

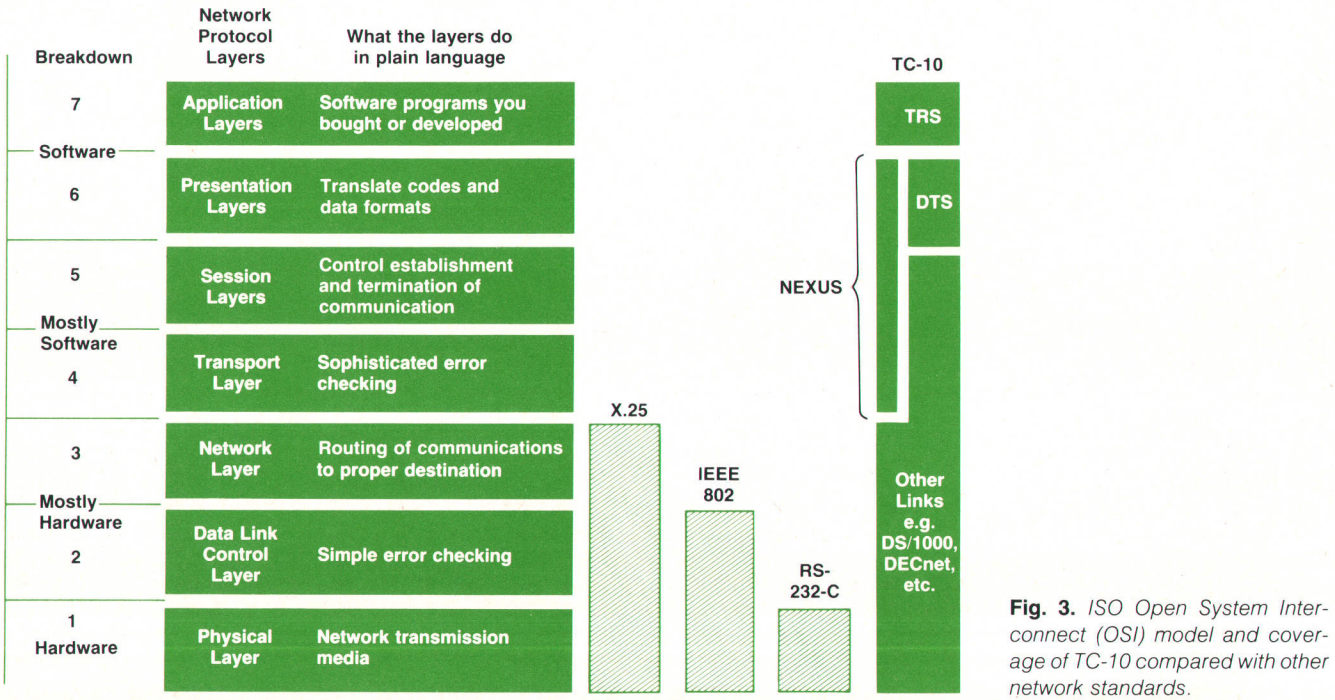


Fig. 3. *ISO Open System Interconnect (OSI) model and coverage of TC-10 compared with other network standards.*

guages, on different operating systems, and on different machines. DTS can be thought of as a software bus for coupling application modules, analogous to the IEEE-488 hardware bus (HP-IB) used to couple CPUs and peripherals.

Primary DTS design objectives were flexibility (extensibility), reliability, and speed. This resulted in a relatively sophisticated solution that would be difficult for different users to implement without creating variations or dialects. Hence, HP supplies DTS source code.

In the DTS language, all data is encapsulated in data transport records, or DTRs. They are similar to Pascal records, in that they have an identifiable type and a number of data fields. However, the DTR structure does not lock DTS to any particular programming language. Since the data transport records must be able to survive outside the protective envelope of a programming language, they are designed to be robust and flexible. DTRs are variable-length logical records that can be converted to a number of styles, as appropriate for the environment. For example:

- ASCII DTRs may be used on a link that can transfer only printable ASCII characters.
- Binary DTRs may be used where number conversion overhead is to be avoided.
- External DTRs are in transportable form, allowing CPU-to-CPU communication.
- Internal DTRs contain data in the form most efficient for the CPU on which they reside.
- Friendly DTRs carry field identification information with each data field, and require no format tables.
- Fast DTRs have fixed format, and require the use of format tables. They are used where high-speed random field access is a requirement.

The above styles can be combined—for example, a friendly ASCII DTR or a friendly binary DTR.

The general structure of a DTR is illustrated in Fig. 4a. This general structure illustrates a DTR's four possible components: header, format table, data, and checksum. The presence of each component depends upon system needs and user requirements. A header is always required, but format tables are usually not present. A typical DTR may look like the one shown in Fig. 4b.

The DTR header contains two kinds of information: data identifiers and style identifiers. The data identifiers uniquely identify the data and its format. The IDs are hierarchical so that a DTR will never get confused with DTRs in other subsystems.

The style identifiers specify the style, or structural characteristics, of the DTR. Individual bits in the style bytes indicate structural characteristics and processing options: ASCII or binary, internal or external, with or without checksum, friendly or fast, trace, security, encryption, write protect, etc.

A small percentage of DTRs may contain a format table. The format table describes the data fields of a fixed-format, fast DTR. If sent, the format table will be sent with the first DTR in a series of DTRs.

The data component contains user data in one of two significantly different styles: friendly or fast. In both cases, the data component contains fields of data. The identity of each field is implied by its relative position. This method of field identification is used because it is considerably more efficient than using a separate ID for each field.

The optional checksum may be used to ensure data and system integrity. The checksum algorithm always generates two printable ASCII characters.

Since DTRs are logical records, a means must be provided to keep them distinguishable from one another. For reasons of efficiency and flexibility, users are encouraged to use special DTS length descriptors to delimit DTRs. Physical variable-length records or text files may carry DTRs, with or without length descriptors. A bundle is defined as two or more DTRs that are glued together in such a way that they will not get separated in processing. A bundle should be able to stand alone without the need for other DTRs to help define its purpose. For example, it may represent an update to a data base, a change in status, or a general ledger entry. In Fig. 5, the structure DTRs, the left and right parentheses at the ends, glue together a number of DTRs containing related data to form a transaction bundle.

TRS

The SPN Test Result Standard, or TRS, defines semantics for general encoding of test results in an IC manufacturing environment. TRS provides a standardized means for data flow between different types of test equipment and data analysis systems. TRS uses the data language or data bus defined by DTS.

Data is sent in bundles of up to 2000 bytes. The intertask communication software is designed such that a bundle will never get fragmented in the course of normal operation, communications link failure, program abort, or system crash. TRS bundles are designed to be decoded easily, even when mixed with bundles from other testers or when resubmitted in duplicate after a disaster. The DTS data transport record structure and bundling scheme enables test data to be sent across any communications link, through transaction-flow architecture, and through any type of file in a file-oriented architecture.

The TRS bundles carry two major categories of test equipment information: test program parameter descriptions and data created during an IC test. The simplest approach would be to send the program information and test results together each time a die is tested. However, this is often impractical since the program information requires two to ten times as many bytes as the test result itself. Such inefficiency can be avoided if the program information is sent once before testing the first die. When the program information is received, it must be saved in a data base associated with the data analysis system. Since the analysis system

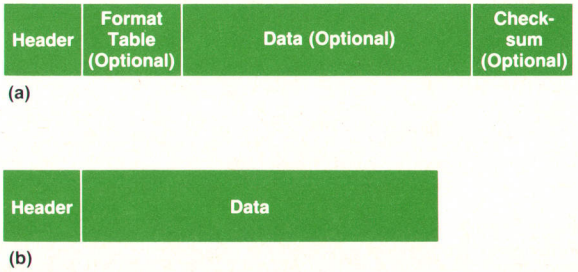


Fig. 4. *Data transport records: (a) general structure and (b) typical structure (format tables are usually not present).*

must also avoid the overhead of storing multiple copies of the same information, it too will keep a copy of the program information on file.

To guarantee that data is correctly stuffed in the analysis data base, program and result data bundles must be logically linked to one another. A header DTR at the beginning of each bundle provides the required linkage. The following fields guarantee that the data base will always be correctly updated: AREA, WIP_LOCATION, LOT, TEST_COUNT, TEST_PROGRAM, and TEST_PGM_REVISION. These fields are defined in the Test Result Standard specification available from HP.[3]

Program information bundles contain program detail DTRs after the bundle header DTR. The detail DTRs define the parameters for each test result. At least two fields are required: TEST_NUMBER and TEST_NAME. TEST_NUMBER identifies a test result's position within an array and is the key that eliminates the need to retransmit all this data with each test result. TEST_NAME is the alphanumeric name that corresponds to TEST_NUMBER. A number of other optional fields, such as unit of measure and limits, may be present in the detail DTR.

Test result bundles carry the actual test results. A bundle header DTR is required at the beginning to identify and link the data to its previously sent program information. One or more test result DTRs follow. These contain the IC wafer identification and the test location on the wafer, followed by arrays of test result values or bin counts. The test result's position in the array corresponds to the TEST_NUMBER previously sent with the program information. Wafer map and composite map DTRs are also defined in the TRS specification.

Formatting Data

One of the first questions a TC-10 user may ask is, "Where should we format the data? At the tester, at the data reduction nodes, or at the data analysis systems?" The answer is that the data formatting can be done anywhere. At first, the user may choose to do all conversions on the HP 3000 Computer. However, for the following reasons, most users eventually choose to create formats at or closer to the tester node:

- Tester results are not yet standardized. Every manufacturer uses a different technique. TRS is a general solution to this problem.
- Test equipment log data is generally not optimized for communications or for updating global analysis data bases. TRS and DTS are.
- TRS and DTS routines are designed to be portable. In most cases, no portable code exists for decoding test equipment log files.
- The software for the conversion only has to be written once, at the test system. Otherwise, conversions would have to occur all over the network.
- Programmers can do a better job on machines with which they are familiar. When a test engineer adds a new test system to the network, the engineer is able to do a better job in a more timely manner at the test system.
- If more than one data standard is being used, it is sometimes difficult to identify which standard is being received.
- Analysis systems are bottlenecks for data flow. It is not wise to burden data reduction nodes and analysis systems with the additional overhead of data conversions.
- Existing test equipment log files lack vital work-in-process (WIP) tracking information required for global analysis of in-process and test data. It is easier to get this information at the test system.
- Although the network may start as several lines to an HP 3000 Computer, it will likely evolve to include data reduction nodes and numerous work stations. One data standard is the only manageable solution in a complex environment.

Having made the decision as to where the standard TRS formats will be created, the next step is to port the software. The average time required for an engineer to port and test the formatting software using the subroutines included in the NEXUS tool set has been about one week. Writing the user program that converts the test equipment log files to TRS transactions takes somewhat longer. This may involve simple array operations, additional user screens, and eventually passing the data to a TC-10 formatting routine. Although time can vary depending upon the complexity of the problem, documentation on the test equipment, and knowledge of the engineer, one month of development time is typical.

Transporting Data

The DTS portion of the NEXUS tool set is designed to encode data so that it can be handled by many different CPUs or communication links. DTS routines can cope with real numbers represented differently, swapped bytes, ASCII-only links and files, fixed-length records, and variable-length records, to name a few common problem areas. Although DTS formatting provides a form that is easy to communicate reliably, DTS routines do not actively move data from one place to another.

Other portions of the tool set deal more directly with movement of data. The architecture is optimized for moving transactions through loosely coupled systems in real-time manufacturing environments. NEXUS modules are concerned with guaranteed delivery, spooling data to prevent bottlenecks, routing transactions according to content, and reliable disaster recovery. The tool set philosophy encourages the use of available communications links wherever possible. Still at the early stages of evolution, NEXUS datacom solutions are primarily a methodology.

DTS is the first NEXUS module to be released as an HP product. For the present, it is the user's responsibility to move the formatted data from the test equipment to the HP 3000 Computer containing the TC-10 data base stuffer. So that users can benefit from future releases, they are encouraged to build a test system network that is compatible with the NEXUS architecture.

Updating the Analysis Data Base

The TC-10 data base stuffer verifies the integrity of in-

Fig. 5. *A DTR transaction bundle.*

coming data and updates the EA-10 engineering analysis data base. The stuffer architecture is designed to provide this service without degrading the response time to on-line computer users. Architectural components that help meet this objective are:

- Transaction flow processing using message files. This avoids the high overhead of file handling (repetitive creates, opens, closes, and purges).
- Stuffer modules operating at a lower priority than on-line, terminal-oriented tasks.
- Spooling with message files. This helps eliminate bottleneck problems. It also increases reliability and simplifies disaster recovery.
- An error correction facility that allows errors to be corrected when convenient.

Until fully supported SPN links are available, it is the customer's responsibility to move the formatted data from the test equipment to the HP 3000 Computer System. Current TC-10 customers are using the following links: tape, DS-1000/3000, SECS-9836/1000, and terminal emulator.

Once data has arrived at the HP 3000, the bundles of data are fed (via the TCRECEIVE process) to message file TCMSGFL, the input to the TC-10 stuffer (see Fig. 6). Since each individual bundle contains sufficient information for decoding, bundles from different testers can be mixed and merged with one another. Consequently, the stuffer's input message file can be fed from a number of sources simultaneously.

The heart of the system is the transaction processor TCP100E. It reads the bundles being sent to it via the message file TCMSGFL. First, TCP100E checks the validity of each bundle. The syntax and checksums (if present) must be valid, or the bundle will be sent to the error file. Some data fields are checked for existence on the data base. For example, depending upon the situation, the following fields may be checked for existence and validity: AREA, LOT, TESTCOUNT, etc.

Bundles with potential problems are divided into two categories: warnings and errors. Bundles without errors are decoded and stuffed into the EA-10 data base. Warnings and errors are sent to the error status file TCSTATFL. Bundles that have errors (and consequently have not been stuffed) are time-stamped and sent to the TCERRTX file. At the user's convenience, bundles in the TCERRTX file can be corrected by the Error Correction Facility, ECF. ECF enables the user to perform on-line corrections. For example, an operator may have entered an incorrect lot number. ECF allows mistakes such as this to be corrected easily. ECF also provides reports that aid in the error correction process. After bundles have been corrected, the operator may release them for resubmission to the transaction processor.

Dealing with Unsynchronized Software Updates

TC-10 must continue to evolve to provide an efficient solution in a rapidly expanding technological environment. Key areas of growth are data base technology, data communications, new testers, and new techniques for analysis of test data. The older, tightly coupled, integrated architectures have restricted product evolution, and have led to unacceptably high maintenance costs. For example, a number of tightly coupled products must all be tested and released in carefully synchronized cycles. Since TC-10 will be operating on a multiplicity of CPUs and testers, such synchronization of update cycles is not practical.

TC-10 uses loosely coupled design and the NEXUS methodology to avoid the requirement for synchronized software updates. Each module is interfaced using fully defined transactions. Direct remote data base updates are not permitted. Although TC-10 is currently used primarily with the EA-10 data base, its design allows it to be used with any test system or data base. The NEXUS DTS tools allow variable-size data fields and the addition of new fields and new records. DTS allows the coexistence of old and new formats, eliminating the need for troublesome file-conversion programs. These features allows TC-10 modules to be developed relatively independently of one another.

Growth Strategy

TC-10 architecture has been designed far in advance of its current implementation. The design is based upon modular, building-block methodology so that the product can be immediately useful, yet evolve to remain a versatile, comprehensive solution over the years. This strategy allows customers to begin automation with released modules, while enabling them to incorporate new features as future modules are released. Risk is low since key concepts in the architecture have been prototyped and implemented in various HP manufacturing facilities.

The DTS module was coded first since it defines the product's data language, the foundation of an evolutionary

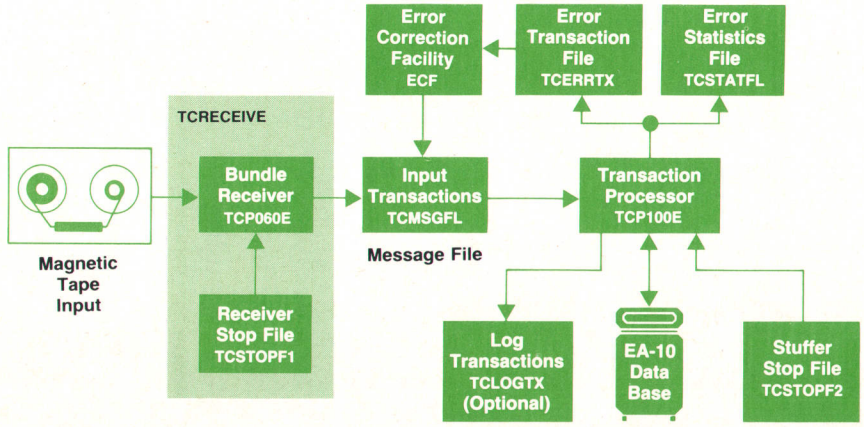


Fig. 6. *TC-10 data base stuffer processing flow.*

integrated architecture. The next logical step was to write software to encode, decode, and stuff TRS transactions. Future modules will concentrate on communications, guaranteed delivery (spoolers), routing according to content, data reduction, and WIP tracking.

Open System Philosophy

An open system product is designed and documented to enable users and other manufacturers to integrate their applications. The manufacturer that commits to providing an open system is effectively telling users and other manufacturers that the integration of equipment of any manufacture into the system is encouraged.

TC-10 is an open system product. The key interface mechanism for coupling test equipment to data analysis packages is a layered set of data format standards. Documents that fully specify TRS[3] and DTS[4] have been submitted to the IEEE and the Semiconductor Equipment Manufacturer's Institute (SEMI) for consideration as standards, and are available to the public.

In TC-10, HP has taken the open system philosophy several steps beyond the normal level. First, fundamental TRS and DTS source code can be purchased from Hewlett-Packard, including the right to copy (HP Part No. 33931A). Second, the source code is written in ANSI Standard Pascal, designed for portability. Third, software validation programs and test files to test the ported software are provided as part of the product. This minimizes risk and maximizes programmer productivity, since customers do not have to write their own TRS or DTS software. Perhaps more important, it helps reduce the possibility of different users spawning incompatible dialects.

Some testers do not support Pascal, but this does not preclude the use of DTS and TRS. If syntax and semantics are followed according to specification, DTS and TRS code can be created directly by the user's program.

Acknowledgments

Since the number of persons who have contributed in one way or another to TC-10 is too great to list here, acknowledgments are limited to engineers who have invested a significant part of their career in the project. Early conceptional design and coding were done by Rob Lucke and Dave Vomocil at HP's Corvallis Division. The concepts were converted into a product at HP's facility in Healdsburg, California. Rusten Hogness and Tom Whitestine wrote most of the zero-bug DTS software. Cindi von Mueller led the TC-10 portion of the project, and with Steve Chiapella's help, designed and coded the TC-10 software.

The engineers from HP's IC facilities deserve substantial credit for their essential professional contributions and guidance. Without their help, TC-10 would have been considerably less practical.

References

1. W.H. Higaki, "Remote Monitoring and Control of Semiconductor Processing," *Hewlett-Packard Journal*, Vol. 36, no. 7, July 1985.
2. *Reference Model of Open Systems Interconnection*, International Standards Organization, ISO/TC97/SC16, Draft International Standard ISO/DIS/7498, 1982.
3. *Test Result Standard Specification*, available from Hewlett-Packard, 137 Lincoln Street, Healdsburg, CA 95448, U.S.A.
4. *Data Transport Standard Specifications*, ibid.

Address Correction Requested
Hewlett-Packard Company, 3000 Hanover Street, Palo Alto, California 94304

Bulk Rate
U.S. Postage
Paid
Hewlett-Packard Company

HEWLETT-PACKARD JOURNAL

August 1985 Volume 36 • Number 8

Technical Information from the Laboratories of Hewlett-Packard Company

Hewlett-Packard Company, 3000 Hanover Street
Palo Alto, California 94304 U.S.A.
Hewlett-Packard Central Mailing Department
Van Heuven Goedhartlaan 121
1181 KK Amstelveen, The Netherlands
Yokogawa-Hewlett-Packard Ltd., Suginami-Ku Tokyo 168 Japan
Hewlett-Packard (Canada) Ltd.
6877 Goreway Drive, Mississauga, Ontario L4V 1M8 Canada

CHANGE OF ADDRESS: To subscribe, change your address, or delete your name from our mailing list, send your request to Hewlett-Packard Journal, 3000 Hanover Street, Palo Alto, CA 94304 U.S.A. Include your old address label, if any. Allow 60 days.

5953-8538

HP Archive