# Improving on the Best: "Like a 1A, Only Better"

**Douglas C. Dowden**
**Robert E. McLear III**
**Alan H. Miller**
**Susan D. Schlough**
**David A. Welch**
**Ralph A. Wilson III**
**Stephen A. Zeile**

Since 1976 the 4ESS™ switch and its 1A processor had been the workhorse of the network. As technology advanced and customers required new services, the capacity of the 4ESS switch came close to being exhausted, making it necessary to develop a successor to the 1A processor. Developers of the 1B processor were presented with the challenge of designing and implementing a system that could accommodate the new network demands, that would be "like a 1A, only better." This paper describes several problems encountered during that project and how they were solved.

### Introduction

Developers of the 1B processor—members of the 1B Processor Development Department—had a significant challenge to meet in designing and implementing a system that would be "like a 1A, only better." Out of the box, in the first office, the 1B processor had to perform better than the 1A processor,[1] which has had thousands of office-years of field experience. This paper describes several of the more interesting technical challenges and the techniques used to solve them. It also discusses how the individual parts were simulated, analyzed, and verified before the processor was integrated into the 4ESS™ switch. Along the way, the developers learned lessons that others may find valuable in creating complex, highly reliable systems.

### The Challenge

The 1B processor's quality requirements—described by J. C. Hsu and L. A. Seese in the introduction to this issue—were stated simply: "Be like a 1A, but better." To the design community, "like a 1A" translated into creating object code compatibility with the 1A processor, maintaining existing interfaces to the rest of the 4ESS switch, and providing an equivalent utility system for the software development community. Being "better" meant improving the performance, increasing the available memory, positioning the 1B processor for future evolution, and delivering a product as reliable from its beginning as the 1A processor is after almost 20 years of evolution and refinement.

### Development Approach

The paper by C. E. Betta et al.[2] earlier in this issue details the 1B processor's high-level requirements. Above all else, the developers felt responsible for maximizing the reliability of the 1B processor. The short-term cost of designing hardware or software that would increase the long-term reliability was considered money well spent. All other requirements—compatibility, increased speed, and memory capacity—were less important than achieving maximum reliability.

To meet these requirements, developers divided their efforts among hardware design, diagnostic design, fault recovery software design, and project management. To ensure reliability throughout the various design disciplines, they emphasized both fault prevention and verification.

One area of fault prevention deserves special attention. Procedural errors are a major cause of switch downtime, regardless of the technology. Procedural errors can turn routine maintenance tasks into outages, and minor service interruptions into media events. Although the solutions to these potential problems lay outside the traditional hardware/software design paradigm, the developers made extensive efforts to solicit design input from installation, manufacturing, and customer documentation and training. They gathered information from AT&T Network Systems (NS) organizations and Network Services Division (NSD) supervisors with extensive field experience, literally making them part of the design and development team—for periods of a year or more.[3] (See Panel 1 for definitions of abbreviations, acronyms, and terms.) Their input was used to improve the basic design and to make the product easier to install, manufacture, and operate—and less subject to procedural errors. The Accelerated Life Testing Program, developed in partnership with the NSD customer, verified the developers' efforts to reduce procedural errors.[3]

Last, but not least, project members developed a Reliability Assurance Plan to crystallize the overall approach to build in, and then prove in, product reliability. The Reliability Assurance Team, described in the hardware design section of this paper, resolved open issues.

**Panel 1. Abbreviations, Acronyms, and Terms**

EMC—electromagnetic capability
EMI—electromagnetic interference
ESS—environmental stress screening
EST—environmental stress testing
ITN—Integrated Test Network
NS—AT&T Network Systems
NSD—AT&T Network Services Division
PFI—physical fault insertion
STRIFE—stress for life
stuck at one—the signal on a pin is forced to its logical 1 value
stuck at zero— the signal on a pin is forced to its logical 0 value
TLP—trouble-locating procedure

### Project Management

Besides making sure that the development team met its milestones, project managers coordinated efforts with functions in other program areas (such as documentation, deployment, and manufacturing) and actively managed development's interfaces to those areas. Concurrent engineering approaches minimized schedule delays that could result from serial handoffs. Other Best Current Practices were used where possible so that project members did not have to take the time to prove in methods and techniques.

Project managers constantly monitored the project status by conducting biweekly internal reviews and monthly executive-level customer reviews. The program manager, who had the ultimate responsibility to deliver a reliable 1B processor on schedule, attended each review and helped resolve issues between program areas as needed. Having a high-level manager with intimate working-level knowledge of the problems and successes was a valuable aspect of the program manager's role. He never failed to remind the project members—by word and action—of the strong executive-level interest in, and support for, the success of this project.

In addition to the periodic reviews already mentioned, outside organizations were asked to conduct special project audits, which were definitely worth the time and effort. A Switching Architecture Review Board was convened to evaluate the technical merits and plans of the 1B processor. Likewise, a project management audit was held to evaluate the nontechnical and management aspects of the project. All issues, whether raised during the periodic reviews or by the special audit teams, were tracked until they were satisfactorily resolved.

### Hardware Design, Development, and Testing

The hardware designers were as obsessed with the 1B processor's reliability as they were with the entire system design. This is an area of the overall system design that has great potential to improve system reliability. If the hardware never fails, then no opportunity exists for outages caused by software during system recovery or procedural errors made during repairs. This section describes the processes used to design, verify, and control the changes needed to maximize the hardware's reliability.

The architecture of the 1B processor is described by C. E. Betta et al.[2] earlier in this issue. The processor is physically large, and its design is somewhat complex. A duplex 1B processor cabinet contains 100 circuit packs divided between two 7-foot-tall frames (similar to those used in the 5ESS® switch). Most of its circuit packs measure 13″ x 16″, with 600-pin connectors. Figure 1 shows the four major logic units: the bus driver/receiver unit, the interface bus unit, the central

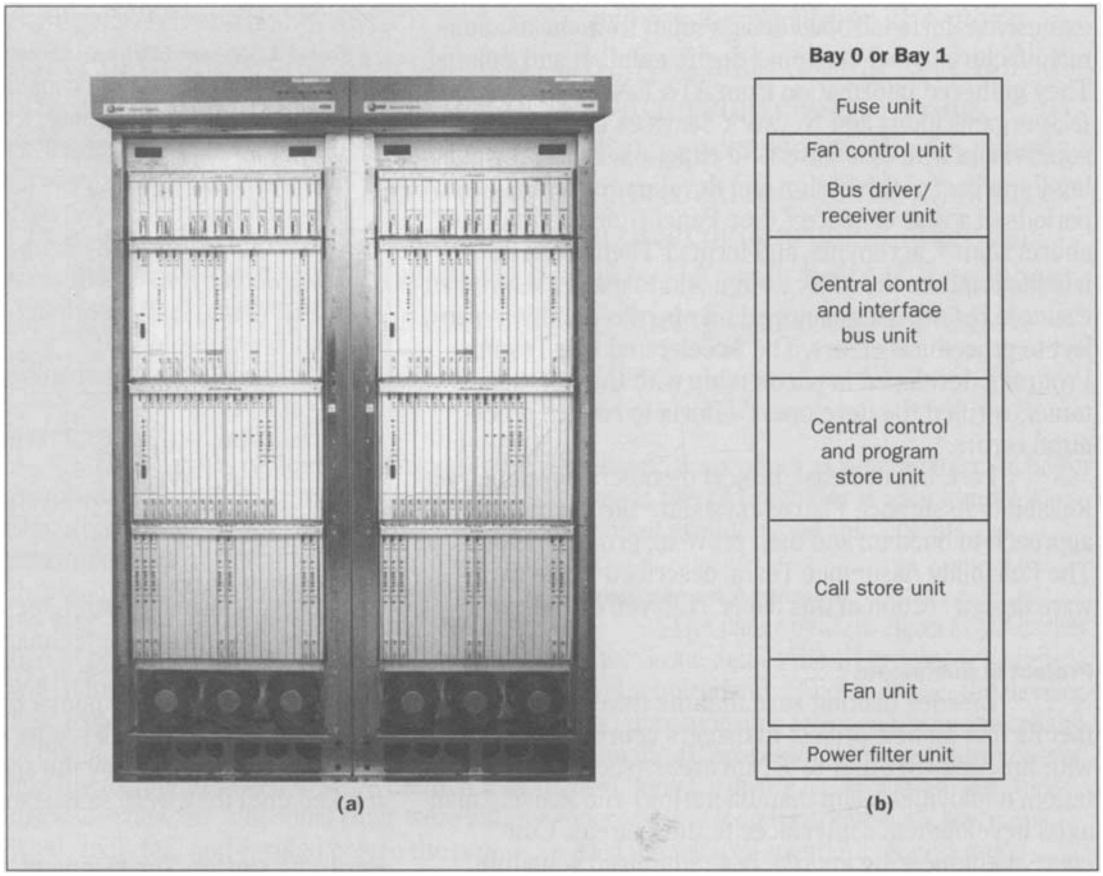


**Figure 1. (a) The 1B processor, contained in two bays, and (b) the physical layout of a sample 1B processor cabinet.**

control unit, and the store units. Three other units—the fuse unit, the fan units, and the power filter unit—contain little logic. The central control unit consists of 16 circuit packs per bay, each populated with some 250 devices, of which roughly 50 percent are programmable logic devices. Interconnecting all the 1B processor circuit packs are 8 backplanes (4 codes, each used twice), and nearly 800 cables that carry more than 13,000 signals.

**Hardware Design.** As described earlier, the major focus during the 1B processor development was reliability—both on the day of cutover and throughout the 1B processor's lifetime. After reliability, the next concern was making the processor's operation transparent to the user, while improving performance and expanding the memory spectrum.

The solution chosen to address the performance issue was twofold:

- To speed up the clock rate of the processor, and
- To convert it into a three-stage execution pipeline architecture.

Increasing the clock speed was made possible by the evolution of technology since the design of the 1A processor, including an increase in the speed and integration of components, and a smaller overall size. The pipeline divided the operation of each 1B processor into indexing, instruction execution, and memory store. (A prefetch queue exists in both the 1A and the 1B processors.)

To expand the memory spectrum, the 1B processor changed the use of one of the 1A processor's bits. Although the 1A processor is a 24-bit machine, it uses only 22 bits for addressing memory, and the remaining two as software flag bits. The 1B processor reclaimed one of these remaining bits, devoting 23 bits to addressing memory. To position the 1B processor for future evolution, the hardware throughout was designed as a 32-bit architecture, although only 24 bits are currently being used. A new interface bus was also added, providing locations for adjunct processors or bridges to adjunct processors to be placed, each having direct memory access to and from the 1B processor's memory system.

Like the 1A processor, the 1B processor has a duplex architecture, with two sides running in lock step and matched on a clock-cycle basis. Any discrepancy triggers an interrupt into fault recovery code. The hardware was designed with a history stack that enables the recovery software to determine when and why things went bad and which piece of hardware to implicate. Given the pipelined nature of the machine, this stack was absolutely necessary.

Developers of the 4ESS switch application depend on being able to see into and control the operation of the 1B processor as they design and test their code. To support this effort, the 1B processor utility system was designed and integrated into the hardware design from the beginning. Comprising roughly 25 percent of the overall 1B processor hardware, the utility system allows the developer to monitor internal registers; control processor execution, such as the ability to stop, start, or execute a single instruction; and to trigger action based on addresses and/or data patterns. Utility system software, developed along with this hardware, gives users a linear view of their code execution, as if the pipeline operation were non-existent.

**Design Efforts.** Because of the extremely aggressive rollout planned for the 1B processor, and because the project required the delicate skill of a brain transplant, extraordinary measures were taken throughout the hardware design interval. To ensure that the product being delivered would be reliable, the developers undertook a number of preventive measures and verification steps that went beyond the typical development.

The first preventive measure, more a management decision than a technical one, was made early in the development cycle. The hardware design and manufacturing teams were directed to avoid any custom chip development. Although this decision caused considerable effort during the development, in hindsight it proved to be a particularly wise choice. In the early stages of 1B processor development, an extraordinary effort went into 1A processor archeology. The 1A processor had been designed some 20 years earlier. Since its release, little attention had been paid to documenting the ongoing maintenance of the architecture and design. The 1A processor's 800,000 opcode variants, and often incorrect documentation, made our effort to understand its functionality considerable—and error prone. Each opcode had to be mapped into the pipelined architecture of the 1B processor. Because of the pipeline, the effect of potential combinations of opcodes had to be considered to avoid resource contention between consecutive operations. Had the developers decided early in the design process to use very large-scale integrated implementation, the design changes that occurred during the development would have been unmanageable.

In retrospect, perhaps the most significant contributor to the reliability of the 1B processor's hardware design was the formation and execution of an extensive, formal signal integrity effort. This effort, which involved virtually every circuit designer, was led by a core team, which addressed such common design issues as ground bounce, overshoot/undershoot, and technology compatibility. The recommendations and requirements derived were then distributed to the 1B processor hardware design team.

Logic simulation, static timing analysis, and transmission line analysis were all used to perform hardware verification. A team within the 1B Development Department provided tools to help characterize the 1B processor operation. To the extent possible, these tools were run before designs were released to be laid out and were used to analyze virtually every signal path, or *net*, in the 1B processor, both before and after layout. The transmission line analysis tool (using Thevenin/Norton equivalent models of the device I/O) analyzed the ringing, overshoot, undershoot, and signal settling time of each net. This output was used as input for the modular timing verifier (MOTIVE), the static timing analysis tool. When any of these tools identified a potential problem, measurements were taken on the real system. Numerous changes and enhancements were made to the design as a result of this overall analysis. This effort eliminated field changes that often result from timing or noise margin problems.

The Reliability Assurance Team was also formed early in the project. Among its other activities, this cross-functional team analyzed such factors as field trouble tickets, customer assistance requests, and network and system outages from the 1A processor and 4ESS switch;

it then grouped their causes into three categories—hardware, software, and procedural. An effort was made to address these in the design of the 1B processor. The hardware design of the 1B processor was influenced by all three categories. To reduce the probability of procedural errors, a number of changes were made to enhance the craft interface, including marking faceplates to indicate service/repair groups, adding light-emitting diodes to some packs to allow for common repair procedures, attaching specialized tools and documentation to the cabinet door, and providing hardware and software tools for cable troubleshooting. To improve the reliability of the 1B processor, anyone in a design organization associated with the 1B project could raise or address an issue before the Reliability Assurance Team.

**Design Verification.** After the initial design was implemented, a subset of the hardware design team executed a series of design verification tests above and beyond the typical hardware verification effort. One successful verification effort was stress for life (STRIFE) testing, a thermal stress of the hardware design above and below the normal operating and design temperature. In the parlance of the 1B processor, STRIFE became the term given to environmental stress testing focused on design robustness, whereas environmental stress screening/environmental stress testing (ESS/EST) was used for stress testing focused on eliminating manufacturing defects. The requirements call for the 1B processor to run between 0°C and +50°C. To test the processor, the Columbus factory's thermal stress capability was used to raise the temperature of the hardware above 80°C and lower it below -20°C for extended periods of time while the system ran diagnostics and other tests. Initially, numerous design bugs were uncovered and fixed. By the time the 1B processor entered production, it could pass this test easily.

The hardware design team also conducted a timing stress program. In this set of tests the clock frequency of the 1B processor was incrementally increased until failures occurred. The failures were analyzed and the failing paths identified. These failures were then correlated to the earlier timing analysis as a check on the timing analysis tool. The direct mapping of these empirical failures to those predicted by the tool provided confidence in the overall timing design of the product.

Early in the project, testing was also performed to check for compliance to such North American Equipment and Building Standards (NEBS) as seismic, electrostatic discharge, electromagnetic capability/electromagnetic interference (EMC/EMI), and safety. These tests precipitated substantial changes to the structural and electrical integrity of the equipment. Although the framework of the 1B processor was taken from the 5ESS switch, it was stretched vertically, altering its characteristics substantially. Steel was added to reinforce the base of the frame, and grounding straps, filters, and gasketing were added for EMI/EMC. In the end, the 1B processor passed all tests with margin.

Another part of the 1B processor verification testing was an extensive craft procedural testing program. This included not only performing normal craft procedures (such as pack changes and cable repair), but intentionally trying abnormal craft procedures, such as removing the wrong circuit pack or removing the circuit pack from the wrong service group. Although errors were expected in the latter, the intent was to ensure that the 1B processor could recover without manual intervention as soon as a usable configuration was available.

**Change Control.** Throughout the development interval, more than 2,400 hardware design changes were made. Because this was anticipated, a change control procedure for hardware products was implemented early in the development cycle. This procedure documented each proposed change, the details of the resolution, any coordination to other hardware and/or software, and the deployment of the change. The committee that managed this process contained representatives from the design and manufacturing organizations, Corporate Product Realization, and the customer (NSD). Beyond keeping change management under control, this process continues to ensure that the product meets customers' expectations.

### Diagnostic Design, Development, and Testing

Diagnostic developers for the 1B processor faced two unique challenges not encountered in earlier projects. First, they had to *prove* compliance with requirements for diagnostic coverage and trouble-locating procedures (TLPs). Many past projects had high diagnostic coverage as a design goal, but did not have to prove that the requirement had been met. Now, diagnostics had to prove they were able to detect 95 percent of all classical pin faults possible in a duplex 1B processor. Also, circuit packs suspected of being faulty had to appear on the TLP list in the first position at least 60 percent of the time, positions 1 to 3 at least 90 percent of the time, positions 1 to 7 at least 92 percent of the time, and somewhere on the list at least 95 percent of the time. Both of these coverage requirements were derived from the larger, "like a 1A," requirement, and also from the overall system downtime requirement.

Achieving and verifying this high pin fault coverage was complicated by the fault-tolerant hardware design. If a fault occurred, some areas of the hardware, such as clock circuitry, were designed to shift control automatically to the nonfaulted mate of that circuitry. Therefore, some faults, such as loss of a clock signal, could not be detected because the hardware always moved immediately to a nonfaulted state.

The second unique challenge for diagnostic developers was that customers expected to be able to detect and isolate cable faults in the 1B processor. This had never been a requirement on switching products and was not part of the original project plan. Because our customers expected cable faults to be isolated as well as detected, our developers had to find a way to resolve this issue.

The scope of the diagnostic development effort for the 1B processor was large. Over a four-year period, diagnostics were written for 29 unique circuit pack codes. A significant part of this large effort was devoted to proving that diagnostic fault and TLP coverage requirements had been met. Large amounts of both capital and expense money were expended to create a complex simulation environment and to provide circuit packs for the physical fault-insertion (PFI) effort.

**Meeting the Challenges.** Meeting the challenges described earlier required extraordinary teamwork and commitment from the diagnostic developers, as well as creative use of the tools and technology available. The developers made extensive use of logic simulation, fault simulation, and PFI to meet the coverage challenge and verify the accomplishment. Meeting the cable fault isolation challenge required the innovative efforts of a cross-functional team of 1B processor developers.

**Fault detection and TLP coverage.** Fault simulation was used as a starting point for evaluating progress in meeting both fault detection and TLP coverage requirements. On an ongoing basis, it provided a partial look at fault coverage and was useful for checking the coverage of specific classes of faults. Fault simulation could not be used to verify that the customer requirement had been met because the process was much slower than anticipated and the project lacked a complete system-level simulation model of the 1B processor.

Fault simulation proved useful, however, in generating the initial suspect pack lists produced by the TLP portion of the diagnostic routines. Random faults were generated and simulated. A list of the specific tests that detected each fault in each diagnostic program provided developers with preliminary lists of faulty circuit packs. These lists were used as TLP information and were later refined through PFI testing and analysis of specific sections of code.

PFI was the main tool used to verify that customer requirements for diagnostics had been met. Developers applied classical "stuck at one" or "stuck at zero" faults to special circuit packs that had been manufactured with most integrated circuits socketed. Fault-insertion boards were created as an interface between the device being faulted and the remainder of the circuit pack logic. By flipping switches on the fault-insertion board, developers could create the desired fault. During the course of the 1B processor project, three distinct PFI assessment efforts were made.

An initial assessment of fault and TLP coverage provided developers with information on how well they had met their design targets. A set of 475 faults, selected to encompass every circuit pack, identified weak areas and enabled the diagnostic developers to direct their efforts specifically toward improvement. Adding tests, however, did not always improve fault detection. The fault-tolerant hardware design sometimes made it impossible to detect certain classes of faults in a straightforward manner. Some types of faults could only be uncovered by using complicated approaches.

A second PFI assessment was performed to retest faults being inserted in the Accelerated Life Test Program at the Integrated Test Network (ITN), as described in the paper by C. L. DeCaluwe et al.[3] This was necessary because concurrent development and customer testing efforts required increasing both fault detection and TLP coverage while the customer was evaluating the diagnostic software. In many cases, deficiencies identified by the customer had already been fixed, but the software containing the changes had not been loaded. As a result, the diagnostic development team had to prove to the customer that the change had been made by re-inserting the same physical fault and collecting the diagnostic response.

The final fault and TLP coverage assessment verified that the diagnostic developers had met the NSD customer requirements. A random sample of 338 faults was used to evaluate both diagnostic fault coverage and TLP coverage. This sample size was chosen by assuming a binomial distribution and using standard statistical methods to calculate the sample size required to report results of 5 percent with 95-percent confidence. Developers successfully met the customer requirements. Classical pin fault coverage was 93±5 percent, with 95-percent confidence. Defective circuit packs were first on the TLP list 75 percent of the time, in positions 1 to 3 at least 95 percent of the time, in positions 1 to 7 at least 98 percent of the time, and somewhere on the list at least 98 percent of the time.

**Cable fault resolution.** The NSD customers expected the 1B processor to include diagnostics that could detect and isolate cable faults. They were worried that cable faults could lead to long service outages without some way to specifically identify and isolate them. Experience showed that diagnostics almost always failed when a cable was bad, but the failure was reported as a faulty circuit pack. This experience supported the customer concern, because changing all the circuit packs on a TLP list could require significant time and possibly still not fix the problem.

Developers needed a creative solution to the problem and got one from a cross-functional team composed of a diagnostician, a hardware designer, and a simulation expert. They approached and solved the problem by using readily available information in new ways. A large body of information existed, including:

- Tables of data showing pack/cable connectivity,
- Tables listing the net names, locations, and cable groups, and
- Diagnostic test failures identifying suspect circuit packs.

The team created a way to use this information to satisfy customers.

The software tool developed by the team started with the list of suspect circuit packs and accessed the pack/cable connectivity information to produce a list of all possible cables connected to the suspect packs. Next, this information was reassessed according to the weights assigned to the suspect circuit packs in the original diagnostic failure. The user was then able to organize the list by cable groups, net names, or equipment locations. The result was an ordered list, headed by the cables most likely to be faulty.

To test the effectiveness of this tool, a blind trial was conducted in which one staff member inserted cable faults and a team of developers or field support personnel resolved the problem. The length of time taken to identify the faulty cable was recorded. For every case tested, the team was able to identify the cable faults correctly in less than four hours, even with the most difficult type of signal faulted. This test was also used to gather input for further improvements to the tool and the presentation of the information it generated.

The tool developed by this creative team can be used in the field when needed, and was immediately useful in the factory during final frame assembly. It effectively addressed customers' concerns in the short time frame required by the overall project schedule and saved the expense of developing a separate cable diagnostic function.

### Fault Recovery Design, Development, and Testing

While almost all of the 4ESS switch application software that ran on the 1A processor worked without change on the 1B processor, 1B software developers had to rewrite about 60 percent of the fault recognition and recovery software. The development of the 1B processor fault recovery software presented several interesting challenges, including:

- Forming and training a team of developers and testers who were unfamiliar with the 4ESS switch and/or the 1A processor, and then fostering good communication among them;
- Reconstructing the key system requirements for a 20-year-old implementation now performing at better than 99.999-percent availability;
- Overcoming a restrictive development environment by putting in place tools and techniques that would enable developers to interact with the old 1A processor environment from a modern development environment;
- Using hardware and software simulation to uncover hardware/software interface problems early enough to make the required changes and still keep on schedule; and
- Making development successful by having other developers apply a disciplined program of aggressive testing using PFI and design-based techniques.

**Team Building, Training, and Requirements.** The 1A processor does not have all the components of an operating system and is programmed in a custom assembly language. Because most of the development team was unfamiliar with the 4ESS switch—they came from environments using high-level languages, and had little experience in low-level assembler coding—the first objective was to educate the team. The basic requirement was that the 1B processor be "like a 1A"; a critical objective was to determine what those requirements were. The first step the group took, therefore, was to write a development-level requirements document for the maintenance and fault recovery of the 1B processor. This document was reviewed by the entire project team and the customer.

A second major challenge was to resist feature creep—that is, the urge to develop new features to make the 1B processor better than the 1A processor by using new and different recovery strategies. To stay on schedule, the fault recovery team concentrated on writing software that faithfully represented the maintenance features of the 1A processor. They cleaned up the code, removed obsolete features, and fixed bugs, but resisted the urge to add new maintenance features. Ideas for improvements were documented and deferred to future product releases.

Another key to success was identifying both the customer and customer expectations. This particular customer demanded that for software faults causing service-affecting incidents, the corresponding fix must be delivered in either 1, 7, or 21 days, depending on the severity of the problem. Knowing the customer's expectations helped the project team focus on some key deficiencies in the new hardware architecture that would have made quick trouble resolution difficult, if not impossible. Identifying these architectural problems early in the development cycle allowed the project team to change the hardware and software so that later, if problems occur in the field, the necessary data will be collected automatically, enabling developers to identify and fix those problems rapidly.

**An Effective Development Environment.** The development team consisted of people whose experience was with developments based on the modern UNIX* operating system. By comparison, the 1A processor development environment, although powerful and reliable, was primitive and difficult to use. Because bugs would undoubtedly be introduced into the new tools, it was also highly undesirable to rewrite or attempt to port the 1A processor compilers to a new UNIX-based system during the initial development of the 1B processor.

Tool development was initiated early with a UNIX-based environment that would provide developers with a familiar environment, but would still use the existing 1A processor tools. New tools performed source code change control and automatically delivered compilation jobs to the 1A processor development environment system, collected the results, and sent them to the developer. Instead of using software patches for testing, the team developed new methods to incrementally test complete object modules. This allowed them to use the old system without changing it, but also accommodated the needs of the developers, facilitating steady progress.

The development team learned how the 1A processor worked and how the new 1B processor hardware was expected to work. As they began developing new 1B processor software, they realized that, to make the project a success, they had to be able to communicate and voice their questions about the new system effectively. This was facilitated by weekly group meetings and an online bulletin board called *Notes*. (The notes source was obtained from Ray Essick, Department of Computer Science, University of Illinois at Urbana-Champaign.) *Notes* allowed developers to submit articles on a variety of topics to which anyone could respond with follow-up comments. For example, one topic was a forum for brainstorming scenarios that would push the fault recovery algorithms to the limit. *Notes* became a forum for sharing ideas, tricks, and techniques that made the job easier, shedding light on how the hardware or software worked, and documenting ideas that could be deferred until later generics. Key aspects of this tool were its informal nature and ready accessibility to all developers of 1B processor fault recovery.

**Hardware and Software Simulation.** Simulation proved invaluable to the success of the project. Hardware simulation helped the fault recovery developers understand how the new 1B processor pipeline worked during fault scenarios, which in turn led to significant hardware design changes that made fault recovery possible. Software simulation allowed them to perform initial testing and debugging at their desks, leaving the laboratories available for better uses. The fault recovery developers simulated the hardware and postulated various fault recovery scenarios. This helped them understand

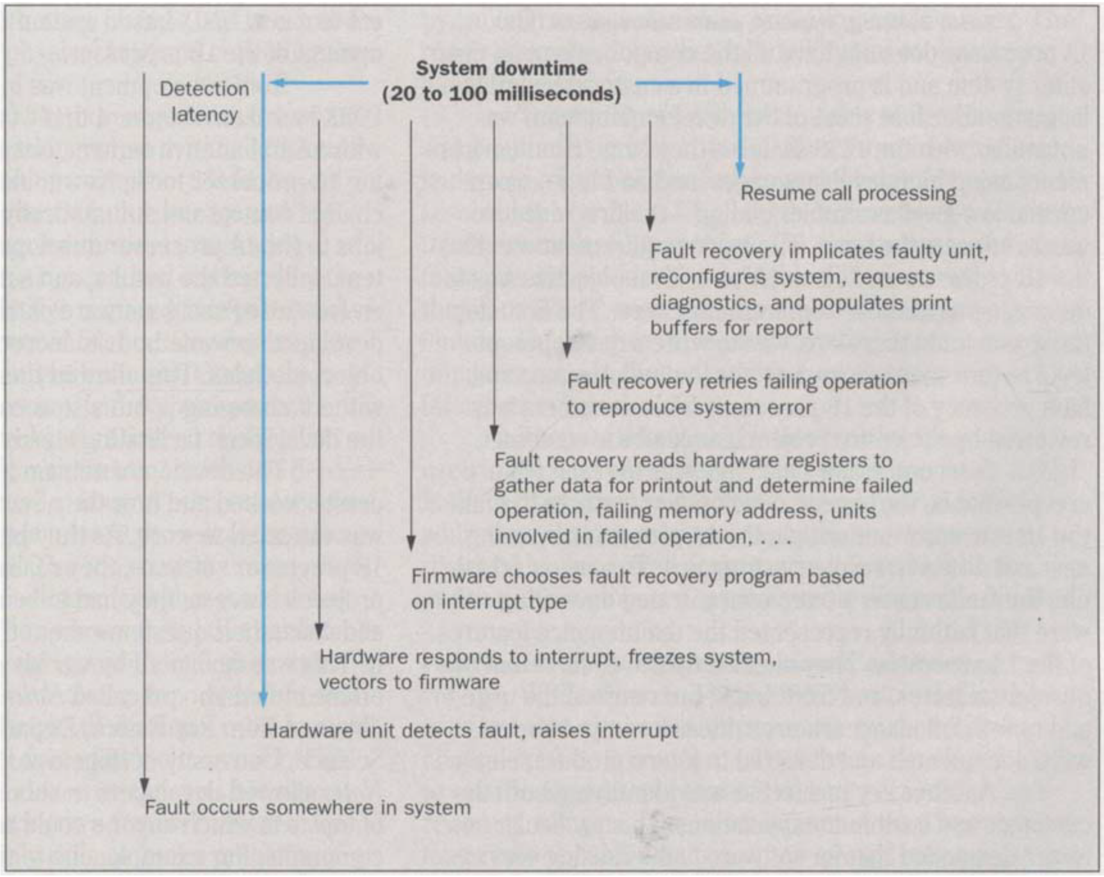


**Figure 2. A typical 20- to 100-millisecond system interrupt scenario for faults that do not require disk access. During the system downtime, 2 to 20 new calls would be blocked, but stable calls would not be affected.**

how the hardware worked, demonstrated the types of damage that fault recovery software would have to repair, and proved to be a common frame of reference for discussions between both hardware and software developers.

When a hardware design flaw was suspected, the fault recovery developer created a simulation to prove or disprove the theory. The simulation model then became an indispensable communication vehicle that allowed software developers to present problems to the hardware developers, greatly reducing the time and effort that normally would have been expended on convincing the hardware developer that a problem actually existed. Through this process, developers found and solved problems critical to the reliability of the product early in the development cycle.

Software developers wrote a proprietary software simulator that allowed them to simulate execution of the 1B processor code on their desktop workstations. Performing the initial debugging and running code first at their desks saved significant amounts of lab time.

**A Disciplined Approach to Testing.** Initially, each developer unit tested his or her own code, after which it was integrated into a system load and subjected to integration testing. Capability testing—the final aspect of development, and critical to project success—made aggressive use of PFI to test the fault recovery capabilities that were developed. Capability testing was planned and executed by developers who had developed other areas of fault recovery. The testers, also competent developers, analyzed the hardware and software to be tested and carefully chose faults to test all aspects of the fault recovery software.

The hardware faults were chosen based on realistic failure scenarios—for instance, a short between two adjacent pins. During the testing program, faults were inserted in all functional blocks of the hardware. These faults were chosen to expose the software to all fault signatures during the course of testing. The software faults chosen simulated the corruption of memory and also caused the worst possible scenarios for software to handle. The objective was to demonstrate automatic recovery even if drastic recovery was required to correct the damage.

After these phases of development testing were completed, the product was tested further by system testers and the customer in the Accelerated Life Testing Program at the ITN.

**A Management Team That Listens.** The success of this project was due to management's willingness to listen to bad news and to make the tough decisions that allowed developers to change the design or reschedule intermediate benchmarks, if necessary, to create a processor that exceeds customer expectations and displays best-in-class reliability.

Through simulation and analysis of the hardware, the fault recovery team discovered, well into the project, that certain failure scenarios could cause the system to recover without leaving any usable fault signature. Without accurate failure data, no timely way existed to correct the defect for the customer. To management's credit, a team was commissioned to remedy the problem, even though this required rescheduling and a major redesign of several circuit packs. The resulting product—a system with excellent reliability with which the customer is well pleased—has proven the value of this willingness to make the tough decisions.

## Processor Integration

The 1B processor did not lend itself to traditional modular integration. Instead, capabilities were delivered in phases. At planned intervals, a processor integration team collected all existing deliverables and combined them into a system software load, which was used as the basis for each development team's future work. It also provided an environment in which lab administrators could see how the various components worked together.

At a high level, the interfaces between hardware, diagnostics, and fault recovery software were fairly well defined. The hardware would detect trouble and cause an interrupt, which, in turn, would trigger fault recovery software to handle the interrupt. Fault recovery would schedule diagnostics for later execution and then return the system to normal, as shown in Figure 2. If this sequence varied, something was broken!

But it wasn't always that simple. Typically, the person who found the problem would assemble a small team in the lab to help identify the problem's source. The team would analyze the system's behavior, based on the delivered code, and determine if the system was broken or simply lacking a piece of functionality. If the system was broken, a problem report was written and assigned to a developer based on the collective judgment of the team examining the problem.

In this way, processor integration was entrusted to the processor integration team and lab administrators. They had a thorough working knowledge of the state of the development—they knew what should be working and what functionality would come later. They were empowered to bring developers into the lab at any time to investigate potential problems. This approach was used successfully to turn over a fully functional 1B processor to the 4ESS Switch System Integration/Product Verification group for further testing, as described in the paper by C. L. DeCaluwe et al.[3]

## Conclusion

The result of all this effort is an extremely satisfied customer. The number of hardware changes is below what was predicted and negotiated with the NSD customer. Changes that have occurred center around firmware updates and component defects. The total absence of hardware changes in the areas of noise and timing margins since the first office was shipped is significant. Traditionally, this area of hardware is subject to a sizable number of field-detected problems.

The rate of hardware failure in the field is also remarkably below the failure rates predicted. In fact, since its introduction, the hardware failure rate of the 1B processor has been lower than the mature failure rate of the 1A processor.

Requests for diagnostics changes have been virtually non-existent. Because the hardware failure rate is so low, little field experience exists to confirm the diagnostic and TLP coverage. But, because diagnostics have been used throughout manufacturing's assembly and system testing to help identify bad packs, the project team has great confidence in the quality and coverage of the diagnostic code.

Fault recovery software changes have been made at a higher rate than any other type of change. Fault recovery changes are being made to compensate for component problems until permanent solutions are available and have been proven to work. Interrupt output and recovery actions are being tuned as the system gains field experience. Developers are discovering inconsistencies between reality and the way they thought the 1A processor recovered, enabling them to modify 1B processor recovery algorithms accordingly.

The field performance of the 1B processor has been exceptional. To date, the 1B processor has not experienced any extended outages or required any manual recovery actions. No procedural errors have been reported. With all of the retrofits in the AT&T network completed, as well as several for local exchange carrier (LEC) customers, only two incidents of failed retrofits have been reported—one an installation oversight and the other a pre-existing software fault. All measures of service far exceed expectations for system performance at deployment—and even exceed expectations for system performance at maturity.

The 4ESS switch application code runs as well on the 1B processor as it did on the 1A, and the peripheral and maintenance interfaces have been maintained as planned. The hardware and software of the 1B processor have clearly set a new standard of reliability for AT&T products. Truly, the 1B processor is "like a 1A, only better."

**Trademarks**

*UNIX is a registered trademark of Novell in the United States and other countries, licensed exclusively through X/Open Company Limited.

**References**


1. J. C. Hsu and L. A. Seese, "AT&T Switching Evolution Challenge," *AT&T Technical Journal*, Vol. 74, No. 3, May/June 1995, pp. 6-9.
2. C. E. Betta, J. B. Lewis, and L. A. Russell, "Evolution of Switch Intelligence: An AT&T Network Perspective," *AT&T Technical Journal*, Vol. 74, No. 3, May/June 1995, pp. 10-17.
3. M. A. Hillis, G. W. Minkus, C. L. DeCaluwe, M. N. Meyers, and E. T Shafer, "An Improved Approach to Product Quality Through Testing," *AT&T Technical Journal*, Vol. 74, No. 3, May/June 1995, pp. 40-48.
4. Issue on the 1A Processor, *Bell System Technical Journal*, Vol. 56, No. 2, February 1977, pp. 119-327.


*(Manuscript approved April 1995)*

***Douglas C. Dowden*** *is head of the Processor/Controller Platform Department at AT&T Network Systems in Naperville (Indian Hill), Illinois. He is responsible for the planning and implementation of processor/controller platforms in the Global Public Networks Offer Business Unit of AT&T Network Systems. Mr. Dowden received a B.S. in computer engineering from the University of Illinois at Urbana–Champaign, and an M.S. in electrical engineering and computer science from the University of California at Berkeley. He joined AT&T in 1974.*

***Robert E. McLear III*** *is a technical manager in the 1B Processor Customer Support Group at AT&T Network Systems in Naperville (Indian Hill), Illinois. He is responsible for customer support for the 1B processor during its deployment interval. Mr. McLear received a B.S. and M.S. in computer science from Purdue University, West Lafayette, Indiana. He joined AT&T in 1979.*

***Alan H. Miller*** *is a technical manager in the Next Network Switch Planning and Development Department at AT&T Network Systems in Naperville (Indian Hill), Illinois, where he is responsible for developing software for the 1B processor. Mr. Miller received a B.S. in mathematics from Worcester Polytechnic Institute, Massachusetts, and an M.S. in systems engineering from the University of Michigan, Ann Arbor. He joined AT&T in 1967.*

***Susan D. Schlough*** *is a technical manager in the Processor Controller Conveyor Belt Department at AT&T Network Systems in Naperville (Indian Hill), Illinois. She is responsible for platform software and system integration for the switch processing platform that will be used in the Next Network Switch project. Ms. Schlough received a B.S. in chemistry from Loyola University of Chicago and a Ph.D. in physical chemistry from the University of Chicago, both in Illinois. She joined AT&T in 1978.*

**David A. Welch** *is a member of technical staff in the Next*

*Network Switch Planning and Development Department at AT&T Network Systems in Naperville (Indian Hill), Illinois, where he is responsible for system and maintenance architecture. Mr. Welch received a B.S. in electrical engineering technology from Northern Illinois University, DeKalb, and an M.S. in computer science from Northwestern University, Evanston, Illinois. He joined AT&T in 1977.*

**Ralph A. Wilson III** *is a technical manager in the Processor*

*Controller Conveyor Belt Department at AT&T Network Systems in Naperville (Indian Hill), Illinois. He is responsible for developing the bridge element hardware for the Next Network Switch, part of the 4ESS switch. Mr. Wilson joined AT&T in 1978, after receiving a B.S. from Montana State University, Bozeman, and an M.S. from Stanford University, Palo Alto, California, both in electrical engineering.*

**Stephen A. Zeile** *is a technical manager in the*

*Processor/Controller Platform Development Department at AT&T Network Systems in Naperville (Indian Hill), Illinois. He is responsible for hardware integration for the distributed processing environment. Mr. Zeile received a B.S. from the University of Michigan at Ann Arbor, and an M.S. from Stanford University, Palo Alto, California, both in mechanical engineering. He joined AT&T in 1977.*