

We've Heard of ROLMs; But What the Heck is a Dual-ROLM?

Recollections of a Pioneering Real-Time Control System

By William E. Richardson

In the mid-1970s, a small group of engineers at International Paper developed and deployed a real-time process control system running on the ROLM 1602 line of computers. In 1981, this resulted in the first fully computer-controlled paper mill in the world. No manual back-up systems existed at a brand new mill in Mansfield, Louisiana. They never were planned. Either the computers ran or the mill shut down. Since no equivalent vendor-supplied system was available at the time, this group had to write its own special operating system for the ROLMs (all in assembler), develop and deploy a hardware and software scheme for full computer system redundancy, and start up the mill one section at a time.

Nothing like this had ever been attempted before. This is the story of how and why all this came to be.

Computer-Based Process Control Begins at International Paper

In 1975, when I joined a small group of International Paper engineers and computer scientists called Process Control, that group had already been operating for 10 years. Established first in the early 1960s as a joint venture between IBM and IP, its first home was at IP's Georgetown, South Carolina, mill. In those days, the card stock for IBM's punch cards was produced at Georgetown; and the computer giant was interested in seeing if computers could be used in controlling the papermaking process. They were indeed able to do that, with a computer at Georgetown one of the first in the world to successfully control a paper machine grade change. After a year of operating the Process Control group jointly with IBM, IP decided to go it alone. Eventually, the group was moved from Georgetown to Mobile, Alabama, where a good number of corporate level operations were then headquartered.

In those early days, the group facilitated installations of the new IBM 1800, which had been specifically developed for process control, in numerous IP mills. The 1800 did not directly control the valves, pumps, and motors of the process. That was done by analog control modules and hard-wired push buttons installed in the control panels of the various mill control rooms. It was those analog controllers that the 1800 "supervised." The first time I entered a control room where the 1800 was in charge, I was a bit spooked by the sight of the knobs on the controllers rotating as the invisible hand of the computer changed the control loop set points in real time. Process Control was successful with the implementation of a number of optimizing routines using the 1800s; but one often found the processes being run manually. The 1800 eventually was reduced to not much more than a data-gathering device.

When minicomputers became available in the late 60s and early 70s, the group began to experiment with direct digital control (DDC), wherein the minicomputer, in this case a General Automation SPC-12, was directly connected to the instrumentation and control valves through a set of digital-to-analog (D/A) and analog-to-digital (A/D) converters. There was a non-disc based, memory-resident real-time operating system available with

the SPC-12, GARTMX; but just about all the operational software had to be written by members of the group. Control loop software algorithms were derived mathematically using z-transforms¹, then programmed to run in the SPC-12 using that computer's native assembler.

The SPC-12 platform worked and was installed in several IP mills of the day, one of which was in Mobile only a few miles from Process Control's office, so it was easy to get to. Another was located in Panama City, Florida; and that location highlighted one of the problems with the system – the hardware was notoriously unreliable and often had to be reloaded...by one of the process control engineers driving over from Mobile (with a car-load of spare parts), some four hours away. Routine maintenance at that distance was a real problem.

The shortcomings of the SPC-12 system were many:

- Regular computer failures occurred, at first unexplained. Detailed examination of computers that had undergone mill service later revealed the mill environment was too tough on machines that had been designed for more benign environments, such as those existing in office buildings or research labs. In those days, paper mills were tough places for electronics. Vibrations from the heavy equipment required to make power, pulp, and paper were strong enough to work printed circuit boards loose from their connectors. And traces of process chemicals in the air were strong enough to gradually corrode those connectors, even internal computer wiring and the legs of exposed transistors and integrated circuits. (Process Control eventually developed a conformal coating specification that all suppliers of mill computer equipment were required to meet or exceed. In addition, later projects included environmentally-protected control rooms, complete with airlock double doors, to keep out the corrosion-causing chemicals.)
- When the computer was not running, the mill had to add an extra operator. As noted, it took a process control engineer to reload it and restart it. And they were all in Mobile.
- Loading was done on a teletype machine via paper tape at 110 baud. The entire system load took a very long time and was not always successful, so the engineer had to do it all over.

In short, there was no way that any SPC-12 mill was going to eliminate its manual and/or analog back-up controls. So a search for a more rugged computer was undertaken by the Process Control group. And since the prime factor in the search was reliability, it was no big surprise when the group selected a mil-spec machine, the ROLM 1602. They were enormously expensive compared to the SPC-12s; but by their very design they were most unlikely to fail. All the printed circuit boards were enclosed in pie plates and then inserted into a heavy, shock-resistant steel chassis with its own cover. No air flow could occur across the pc boards or their connectors. All heat dissipation was done through the chassis itself. The earlier 1602 models got hot enough on the outside that mill personnel

¹ This derivation of the algorithms was done by John Botts, Ed Johns, and Bill Wellborn.

would put wet work gloves on top of them to dry. Mil-spec connectors provided the computer with all electrical connections to external devices.



With the selection of the ROLM (see photo to left, showing the computer, with the front panel on top), there was more work to do on the software side. The ROLM didn't come with any software usable by Process Control except a debugger, which could only be used off-line. ROLM did have a real-time operating system, but it was disc-based – and the disks were the large 15-inch platters familiar to the group because they were about the same size as the 1800 disks. Group members rightly felt that such disks were fine for mills that had

1800s, since those computers were located in office areas, but that the same kinds of disks would never cut it out in the manufacturing environment. The answer was simple but imposed some significant limits: adapt the memory-resident GARTMX operating system for the ROLMs. That meant, of course, rewriting GARTMX in ROLM assembler.² The other programs, such as the loop control software, also had to be ported to the ROLM. Most of this work was done by Brent Siegel, Tom Henderson, and Charlie Jolliff.

The techniques of assembler coding are similar from one processor to another, but the actual instruction sets are different. In the case of the ROLMs, they used the same instruction set as the Data General NOVA computers, with some enhancements. (That's why the early ROLMs were also known as Rugged NOVAs.) At any rate, re-coding all the useful material from SPC-12 assembler to ROLM assembler had to be done prior to the first installation of a ROLM in a mill.

Ultimately ROLMs were installed in several IP mills. Like their predecessors, they were limited in function compared to what was to come later. They were essentially digital replacements for analog controllers, with black and white CRTs for the operator interface, displaying one loop at a time. If the operator wanted to go on manual control, it was no harder to do than putting an analog loop on manual. Further, those first ROLMs did not do motor control (on-off logic). That was still done separately using hard-wiring, relays, and later, programmable logic controllers (PLCs).

Steps Toward Mansfield and the Evolution of the Dual-ROLM

One of the mills where ROLMs replaced SPC-12s was at Natchez, Mississippi. The real-time computer application on that mill's batch digesters became an early prototype of

² GARTMX was deficient in a significant way. All scheduling was completely poll time based, using only one interrupt from the real time clock. Branch and nest interrupt handling logic, therefore, was a major feature that was added by Brent Siegel at the time of porting GARTMX to the ROLM. This was vital to the success of the ROLM platform in all future mill process control installations.

using one computer to back up another. Digesters are used to produce pulp from wood chips. A batch digester is a large pressure cooker. Chips are loaded at the top of the digester, the digester is then capped off, injected with steam and chemicals, and goes through several stages of cooking. At the end of those cycles, a large valve at the bottom of the digester is opened and the pulp is released under pressure into a blow tank for later washing and further processing.

At Natchez, first SPC-12s, then ROLMS, were used to fill, cap, cook, and blow one digester after another in a time-optimized sequence that would always effectively utilize the available raw materials -- chips, steam, and chemicals -- in such a way as to provide high quality pulp. In order to do this, the computer also had to have its "fingers" on some of the pumps and motors involved in the process, not only the analog-type loops. This meant that the computer had to have a package that was able to perform tasks that were interlocked with other tasks and with external hardware. For that reason, a configurable interlock system was programmed and added to the SPC-12 platform and later ported to the ROLM.

The Natchez system also involved some limited back-up, where one computer could take over some functions of another that needed service. Manual and analog control back-ups were still involved in parts of the Natchez pulp mill. Nonetheless, it was at Natchez that the concept of using a process control computer as a back-up for another process control computer was first tried in International Paper.

By the time I joined the group in 1975, the Natchez project was not only complete but was deemed a great success. The ROLMs performed as advertised. None of the computers employed was vulnerable to vibrational problems or corrosion problems. (Associated equipment such as the I/O gear, however, had to be dealt with regarding these issues.)

The next opportunity to implement a significant change in real-time computer technology came with the expansion of the Texarkana, Texas, mill. That facility had been built in the early 1970s, coming on-line with paper production in 1972. In 1976 the company decided to add a third paper machine, along with associated upgrades elsewhere in the mill to make the third machine possible. Once again the Process Control group from Mobile was asked to engineer and install the control systems.

One of the managers in Process Control was a fellow named Jim Chappelle. One day he called several of us into a small conference room and started talking about the vision he had for this next job. As noted, we had control systems to install in various parts of the Texarkana Mill, not just the paper mill proper. Once again, we were looking at computerizing a set of batch digesters, among other things. Most group members assumed that we would just repeat the Natchez digester control project for Texarkana, where the batch process was actually simpler than Natchez's. But that is not what Jim had in mind.

He challenged us to think differently, to advance the state of automation that we had already reached. He envisioned a system where, like Natchez, one computer backed up the other; but he saw the back-up computer as being able to do *everything* the on-line computer could do. He also envisioned the back-up coming on line automatically, should

anything happen to the on-line computer. And he wanted that automatic transfer to occur *without disturbing the process or with any loss of control functionality*. He asked us to figure out how we could do that and do it for the batch digesters only.

Now Jim wasn't a software guy. He had never written code or designed software packages of any type. But he was a hardware expert, the best I have ever had the pleasure to work with. He was completely self-taught, since he had never gone to college. Yet he could run rings around any degreed engineer. So his challenge really was, "Can you guys design a back-up system from a software standpoint that can utilize a pair of computers this way?" We told him we could, although it would take a while to work out the details -- to say the least!

Let's pause here just to inventory what we had to work with in order to get to this "dual" system:

- Two ROLM computers, each of them with 48K of 16-bit words of core memory
- A cross assembler and a relocatable loader to generate ROLM object code. These programs ran on our office's IBM 1800.
- Source code loaded into the IBM 1800 via punch cards. Assembled source code printouts only available from the IBM. Object code transferred to paper tape from 1800.
- Object code loaded into ROLMs via fast paper tape readers. These were an enormous improvement over the 110 baud teletype loaders for the SPC-12s. (NOTE: Floppy disks did not become commonly available until the latter half of the 1970s.)
- ROLM interface to the input/output (I/O) gear through an I/O bus converter (IOBC). The I/O gear for Texarkana was made by Reliance Electric.
- Serial port cards in the ROLM available for connection to the operator interface device (provider yet unknown)
- 16-bit parallel card in each ROLM available for any needed ROLM-to-ROLM communications
- ROLM front panel interface with switches and LED display of memory location contents, one word at a time. Ability to patch core memory locations through such a panel, provided the computer was halted. (See photo, above)



To complicate matters a bit more, the Texarkana mill had no 1800. There were plans to install a large Harris machine in the mill process control office. The cross assembler and loader software would then be moved to it. But Harris installation was not scheduled to happen till well after the digesters had to be running. Thus, during the digester start-up phase, all changes to the ROLMs would have to be made by on-site patch or by generating a revised system in Mobile, where it would be reassembled and reloaded to paper tape, which would then be physically transported to the mill.

Jim built a working mock-up operator panel in his home shop. When it was complete, four of us helped him muscle it into his pick-up truck and bring it to the office where it was connected to the computer control equipment and a process simulator running on the



1800. Later, the manufacturer of the real operator's panel delivered it to our Mobile office for our simulation use and for the installation of the computer equipment we were responsible for. (This had been the practice for earlier projects as well. The photo to the left shows Bill Wellborn and Jim Chappelle standing in front of a boiler control panel upon its delivery to Mobile.)

My educational background was electrical engineering. At the time I started with IP, however, I had just come out of jobs that had taken me into sales and marketing assignments. Since that had familiarized me with product specification methods and sales negotiations, I was assigned to be the chief Process Control engineer working with vendors such as ROLM and our own Equipment Purchasing department. Consequently, I was involved with all the suppliers we would need to complete our tasks, but especially with those where new technology was called for. An important example of this kind of thing was finding a way to upgrade our operator interface.

After discussion among team members, we decided we needed a graphical digester display and we wanted color graphics. The turnkey control systems suppliers had not yet made the move to color themselves. Some of those vendors were quite dismissive of the very idea. In one meeting with sales representatives of the Foxboro Company, I brought up the lack of color graphics on their operator CRTs and one salesman actually laughed at me, saying that *nobody* saw any use for color. We stuck to our guns, however, for the digester ROLM system.

Consequently, Brent Siegel, who by that time had become our lead system software engineer, and I went to Norcross, Georgia, to talk to a man who was making a new product, at about one per week, in his basement. The product was an intelligent color CRT. The company was Intelligent Systems Corporation (ISC). The CRT contained an 8080 processor and enough memory to off-load a lot of the operator interface coding from the ROLM, including the floating point software that would be required to display data in engineering units for the digester operators. Such off-loading was important because of the limited amount of memory we would have in the ROLM. (That limited memory also caused all programmers to work under a general directive to seek coding methodology in the ROLM that made effective use of subroutines and other forms of re-entrant code.)

When Brent and I returned to Mobile, we recommended that we buy an ISC for development work. That was done and the ISC was specified as the terminal we would use at Texarkana for the digester part of that project. The key to doing graphical displays on these terminals was a program called the String Interpreter, which ran in the ISC itself. Brent wrote that program.

The String Interpreter allowed us to store displays as formats on the ISC EPROM memory board. The formats looked very much like FORTRAN type format statements. There were fields that allowed one to draw a picture (box, line, fill, etc.) and there were fields that displayed data (floating point, integer, ASCII, etc.) The String Interpreter parsed the format and drew a picture as defined by the draw commands. Every time a data field was encountered it would fetch the data from the data table that was sent by the ROLM. Every time the ROLM wanted to display a picture on the CRT it would send a format number followed by a stream of data. The String Interpreter called ISC code subroutines as required to actually write into display memory. The important thing here was to be sure that the data matched the format. The formats to draw pictures like digester displays got quite long; and when data didn't match up it could be hard to find the problem, since you were looking at hundreds of bytes of data that had to match the format string exactly. The String Interpreter was the key to the ROLM's ability to call up graphic displays. It was used in every ROLM system from that point forward.

In general, as far as identifying programming talent for the group, for years it had been the standing practice in Process Control to assign all new engineers to write software modules for the various systems. I did well on the assignment I had been given, which netted me the lead programmer job for two critical modules for the Texarkana digester system. One of these was the program used for coordinating the actions of all 16 batch digesters. I worked with a senior engineer named Richard Jackson on this one. He handed me the process logic and I did the coding.

The other assignment turned out to be more significant and more difficult – working out the software scheme for doing the bumpless control transfer from one computer to the other. Here I had to work out the logic on my own. I came up with a workable scheme, consulted with Jim, Brent, and others, and began to program it. I spent a lot of my time in the beginning asking questions. Can we do this or that? Can we get this particular hardware set-up? If we go this way, how will the other system modules have to be modified? It was a total team effort.

Here's what we came up with:

- 1) In order to work, for one computer to fully back up the other, the two systems would have to be loaded with identical software.
- 2) Since the computers would be running asynchronously, they would have to talk to each other. In addition something external and neutral would have to "referee" to make sure that each knew its current status. Fighting for control would be totally unacceptable and dangerous.
- 3) Redundancy would be required not only with the computers themselves, but for all essential elements out through the I/O gear. The field instrumentation, however, would be common.

- 4) For reference, one computer was to be named the “A” computer, while the other would be called the “B” computer. Nevertheless, there would be no “bias” toward one computer versus the other running the process. Whichever computer was currently running the process would be known operationally as the *foreground computer* while the other would be known as the *background computer*.
- 5) Switching a computer from foreground to background could be called for by deliberate operator action or automatically by certain serious failures. Such failure modes would have to be defined.

The IP ROLM real time OS had a simple hierarchical structure. Programs running under it had three basic levels – *functions*, *activities*, and *interrupt handlers*.

Functions ran regularly based upon a defined time schedule. In general, they tended to be longer programs that could be temporarily paused if necessary. Examples of functions were the program that handled the direct digital control of the various control loops (such as those controlling steam valves), the program that managed the process interlocks, and the program that processed operator entries and generated CRT displays.

Activities were programs that also had to run on a pre-defined clock basis. The difference was that activities could cause functions to pause and could not be superseded by any other activity until they completed. Activities were used for time-sensitive actions that just would not work properly unless they regularly ran within a finite time window. Consequently activities had shorter runtimes and were more connected with system level actions than with process control actions.

Finally, at the top of the ROLM system food chain, were the *interrupt handling* routines – where the interrupts would come from the internal real time clock, from the I/O gear, from the operator’s keyboard, etc. Since interrupts could happen at any time, the routines servicing them had to be very short, often simply buffering data to a location where a function or activity would later use it. Interrupt handlers could suspend any other operation in progress except a higher priority interrupt handler.

It was quickly clear to me that the program managing the switchover from one computer to the other would have to be an activity and would have to run at a rapid repetition rate. The fastest any activity could run was every 10 milliseconds, the speed at which the real time clock counted down. We estimated that a successful switchover would have to occur in less than a quarter of a second, in order to make sure that no control action would be compromised. I decided to try for every 20 milliseconds and see how it went. It turned out to be a good choice and was never changed. Most of the time when the switchover program ran, it had nothing much to do except check some special status bits. During a switchover, however, the program ran multiple passes as it waited for the other computer to confirm where it was in the process. At those moments, it was easily the most critical program running.

The key to making sure each computer knew its current role was what I called the referee relay. It was a single-pole double-throw relay that could be pulsed by either computer to make it “point” toward itself. Each computer monitored one of the two relay outputs. If the computer saw a “one,” it was the foreground. If the computer saw a “zero,” it was the background. During a controlled switchover, each computer looked at this relay several

times (and during several passes through the program) and the two cross-checked each other through the ROLM-to-ROLM communications cable to make sure they agreed during the transfer of control (and the repositioning of the relay).

After I had the switchover program ready to go, I told Richard that I wanted to be the first to try it, to “push the button.” I became distracted on some other matter and before I knew it, Richard announced that he had “grossed it in” by pushing the button himself. I asked him if the two computers switched places. He told me that they had – several times back and forth – before settling out. The process was anything but bumpless. As I headed back to my desk to analyze what had happened, I was grateful we were just running on the office simulator. From that point on the phrase “grossing it in” became part of our normal lexicon and usual got a laugh from the other engineers whenever it was used.

The problems with switchover were fixed and soon we went to Texarkana with a working system. I don’t mean to imply that the system was “bug-less.” There were still kinks to work out, details that we had not covered under simulation. One interesting example of a non-bug but one that exposed a design weakness that we fixed on-site is that of the variable display differences.

The Texarkana digester panel was quite simple. It was a stand-up panel, since operators were used to walking up and down panels, adjusting set points, and starting and stopping pumps, motors, and chip belts. It also seemed that there was a management concern that an operator might fall asleep if he were allowed to sit in a chair at a desk.³ There was very little on these new panels other than two ISC terminals and keyboards, one set dedicated to each of the ROLMs in the pair. The terminals and keyboards were not switched during a switchover. There was also a separate set of I/O gear for each computer. What that meant was that each terminal displayed the instrument readings as seen by its associated I/O cards as processed through its ROLM. These seldom read exactly the same on the terminals. We started to get questions from the operators as to which display was correct. We tried explaining that the computers were running asynchronously and neither display was exact and didn’t need to be, since “all the control action was done in fixed point, not the displayed floating point numbers and.....” As you can imagine, this explanation went a little further than the operators wanted – Too Much Information or TMI, as we would say in 2011.

We concluded that the visuals of the panel just weren’t right, that we needed to make them the same. And so we did. We simply added a bit of code that transferred the displayed field data from the foreground to the background computer. Then each computer could just check its own status. If it was the foreground computer, it displayed its own data. If it was the background computer, it displayed the transferred data. The operators were happy and we learned a design lesson for future systems.

By spring of 1978, the main work on the Texarkana digester system was complete, although we fielded service calls from the mill and made changes as needed for some time, since that facility had only a two-man process control department.

³ Interestingly enough, there were high chairs in the control room. After the operators got used to the automated control and understood how little they would have to do manually, the clever ones would put one of these high chairs in front of the panel and use pointer sticks to reach the buttons on the panel without getting out of their chairs.

The first dual-ROLM system had been deployed.

Mansfield and Full Mill Control

Getting the Project and Getting Started

During the late 1970s and early 1980s, new project work came early and often. Just as we were finishing up at Texarkana, a batch digester project requiring a control system was initiated at the Pine Bluff, Arkansas, mill. Brent Siegel was assigned as the lead engineer on that one. He took the dual-ROLM concept there, along with a number of technical advancements associated with the fast-changing computer landscape. Sit-down terminals were one of new things we introduced with this project. This is also where the first pressurized / filtered air computer room was built. An early floppy-disk type system, called a Flex File, was used to load the ROLMs. It was plug compatible with the paper tape system we used at Texarkana. Later, on the same project, we changed to an Altos Computer to do this loading.

The biggest hardware modification we made with Pine Bluff was to change the I/O gear supplier. Reliance I/O (analog to digital) was very expensive and very slow. Each rack required a multiplexor card that had to be used to address and convert each A/D point separately. It was a 12 bit card and required two reads to get all 12 bits. It took a long time to read a 100 or so analog points. Computer Products offered individually addressed 12 bit I/O at a cost considerably below Reliance on a per point basis. In addition Reliance digital I/O was only 8 bits per card, so it ate up a lot of rack space. Computer Products offered 16 bit digital I/O.

As for me, in 1978 I was asked to put together a demonstration ROLM system in the Mobile office for a new mill, with the temporary name IPG-1, since it was not yet known where it would be located. IPG stood for Industrial Packaging Group, so we knew it would be producing an unbleached product. One of the reasons we needed a demo system was because it was anything but a foregone conclusion that International Paper would pick our ROLM-based system to do the process control at the new mill.

First, there had recently been a significant shift in the management philosophy of the company regarding the engineering and construction of major capital jobs. For many years, manufacturing operations had been organized regionally, not along product lines. The Southern Kraft Division, which included central technology and engineering operations headquartered in Mobile, had the largest number of mills. This division's philosophy was simple: "We are papermakers. We run mills, therefore we know how we want our mills to work. Therefore we will design them and build them ourselves." Consequently, the division maintained its own in-house engineering group, of which we were a part, and its own construction company. In 1976, the corporation reorganized along product lines. The Northern Division and the Southern Kraft Division ceased to exist and were replaced by organizational units such as the Industrial Packaging Group. Not all those in the reorganized corporate management team agreed with the "do it yourself" philosophy.

Second, we were not only talking about controlling an entire mill at IPG-1; we were talking about the control of *continuous* processes, not batch processes. In the case of the batch digester systems we had successfully deployed, the sudden loss of computer control (should both computer systems fail) was bad but was not catastrophic. All the valves, pumps, and motors were designed to fail safe. In the case of a set of batch digesters, that meant that most key valves were set up to close upon control failure, thereby “bottling up” the digester. Even a digester in the middle of a cook could be sustained in this state for a while, perhaps for as long as 15 – 30 minutes, without drastically affecting the quality of the pulp. There was time, in other words, to reload and restart a computer should that be necessary. We would have had some explaining to do to the mill manager, but, beyond that, there would have been limited financial consequences, if any at all.

IPG-1 was going to be a different ballgame. The pulp mill was sure to use a more modern digester design – the continuous digester – wherein chips are loaded in a steady stream at the top of a tall vessel. As the chip mass moves down the vessel, at various points steam and chemicals are injected into it. By the time the chip mass gets to the blow valve at the bottom, it is no longer wood; it is pulp. Halting a continuous digester unexpectedly while it is cooking chips almost always results in a plugged vessel which can take hours to clear. In the meantime any paper machine that depends upon that digester for its stock will have to be shut down.

As IPG-1 started to take shape and eventually became the Mansfield Mill, because of its chosen location near Mansfield, Louisiana, we learned more about what would need to be controlled:

- Two power boilers
- Two recovery boilers
- Pumping station for mill water located 30 miles from the mill itself
- Waste treatment and water treatment
- Three continuous digesters (pine, hardwood, and semi-chem)
- Caustic plant and lime kiln
- Two paper machines, one making linerboard, the other making corrugating medium

To make matters more interesting, the mill was being designed to make all its own power. There would be no feed from the local power company. If the boilers weren't up there would be no steam to run the turbines; and there would be no electrical power anywhere in the mill except in the power plant, where a small diesel generator could be started.

Anything that was controlling all the above process equipment had to be designed for 24/7/365. No unplanned downtime for the control systems would be acceptable.

Oh, yes. Mansfield was going to be a big investment on the company's part – about \$650 million in 1980. In 2010 dollars, that would be about \$1.8 billion.

Being allowed to implement dual-ROLM technology to run Mansfield was going to be anything but a slam dunk. That's why we knew we needed the IPG-1 demonstration project.

We chose a simulated recovery boiler control package for the demo system. A recovery boiler is a particularly clever piece of the pulping process cycle that allows a mill to recover chemicals while generating steam and power. When pulp is discharged from a digester (batch or continuous), part of the product coming out is a substance called black liquor. Through a series of washing steps, the pulp is separated from the black liquor. The black liquor is then pumped to a recovery boiler, where it is discharged into the boiler as fuel. The combustible part, mostly lignin removed from the wood chips, burns, generating the heat needed for steam generation. The non-combustible chemicals fall to the bottom of the boiler where they form a smelt bed. The smelt can then be reprocessed to recover a portion of the chemicals needed for the digester to produce more pulp.

A recovery boiler has special safety issues that a power boiler doesn't have. If any water gets into the molten smelt bed, a significant explosion will occur, putting the power house workforce in danger. Such an explosion is enough to buckle the furnace's sides, causing major damage and making the boiler unusable until repairs are made, which typically run in the millions of dollars. Safety is, and always has been, at the very top of International Paper's agenda. So demonstrating control of a recovery boiler was a bold way of making a statement that we were ready to take on any challenge of computerizing a continuous process. The demo system included not only the combustion controls of a recovery boiler but also its burner safety system.⁴

All this was done, of course, under simulation in our Mobile office. All the elements were there for a continuing series of "show and tell" sessions. Computers, racks of I/O, ISC terminals complete with recovery boiler graphics, were there for all to see and ask questions about. Most of the sessions were conducted by Jim Chappelle, the visionary leader that conceived the fully-redundant idea. He would go through a detailed list of reasons why management should assign our people to install the dual-ROLMs in Mansfield. Here's the kind of things he said to all who saw the demo system:

"This is a concept-driven proposal. The concept is to build a control system that is capable of eliminating human error in the control of processes. If a man can control a process using measured variables and the starting and stopping of needed equipment, then an automated control system has the capability of doing the same thing. This could be a real asset to a company, if such a system existed.

"There are two main problems why you cannot buy such a system. First, process control and motor control are viewed as separate systems and are not yet integrated in vendor systems. Second, you would be putting all your eggs in one basket -- a single device could fail or part of it fail which would shut the process down.

"Still, such a system could 1) eliminate human error, 2) lead to better control through data analysis, 3) lead to better safety for man and equipment, 4) perform automatic start-ups and shut downs, and 5) afford better and more efficient operations (cost savings)

⁴ In the end, we were not allowed to put this last item into the dual-ROLM system, although we successfully demonstrated the ability to do so. Industry codes for recovery boilers included the requirement that the safety system be independent of the combustion control system.

“The required elements of such a system would be 1) a reliable computer platform, 2) reliable input/output equipment, 3) a reliable power supply system, 4) a reliable back-up system, and 5) a solid switchover system to get to the back-up.

“These requirements dictate that the system be a dual one – a second system shadowing the one on line. Once this is accomplished you could do the following:

- **Emulate the best operator approach**
- **Improve upon the best operator approach**
- **Improve on safety**
- **Come up with the logic for auto start-up and shut down**
- **Gain a real-time handle on the cost of the process**
- **Schedule maintenance on equipment**

“We admit that in proposing the above, we are facing huge reliability and safety challenges. We also recognize that we need buy-in by management and mill operators. We believe that we have proved we can do the job through our previous successful projects at Texarkana and Pine Bluff.

“Finally, keep this in mind. If all the process inputs and outputs, both analog and digital, loops and motors, go into the computer, then all it takes for optimized control is to get some smart process and process control engineers to develop and implement new and innovative strategies for controlling the entire set of mill processes, optimizing the results for factors such as product quality, cost, and production throughput.”

In the demo system, one of the ROLM computers sat inside a 19” rack on a wooden platform, but not bolted in, as it would be in a real installation. Above it was the IOBC made by Computer Products. Jim put the computer there for a reason. At one point during his pitch, he would lift the front edge of the heavy ROLM chassis several inches and let it go. It would fall back with a loud bang, proving that the ROLM was indeed extremely rugged. Then he would go on with the demo, displaying several boiler graphics on the ISC.

After one such “dropping demonstration,” Jim realized that the displays had frozen and wouldn’t update. So he “sang and danced” hurriedly through the rest of the presentation, thinking all the while that the touted ROLM *had indeed* failed. After his guests departed none the wiser, he started examining the set-up. There was no problem with the ROLM. He discovered that every time he dropped the computer, a PC board in the IOBC had been working its way out of its slot, a little bit at a time. Finally it came out just far enough to freeze the whole system up! Lesson learned – if we got the Mansfield job, we would have to make sure that our entire set-up was *very* hardened – both hardware and software.

Another thing we were asked to do by management was to double-check with key vendors of process control systems to make sure that there wasn’t one of them that could meet Jim’s criteria. Consequently we set up meetings, over several days, with four of the most notable system suppliers to our industry – Honeywell, Foxboro, AccuRay (now part of ABB), and Measurex (years later to be acquired by Honeywell). We spent a lot of time with each of the four (the meetings were attended not just by Process Control people but also by general engineering management and perhaps some business unit personnel). Each company (a) admitted that they couldn’t do what we wanted but (b) claimed they

could develop something, perhaps by joining with one of the others. The results of these meetings were passed on up the line to our business unit and corporate management.

Jim Chappelle was called before top company officials to make a final pitch for the dual-ROLM. I asked him later what he said in that meeting. He told me that he believed he made a good case for our doing the job. He also told me that he issued this caveat to the management team: “When the company has done major capital projects in the past and there have been bumps in the road – breakdowns in supplier equipment such as a chip belt or a washer or a refiner – you haven’t liked it, but you understood. If we do Mansfield and we have a computer problem that holds things up, you aren’t going to *want* to understand.” Management got his point. This was unique, new technology, technology not well-understood by men who had come up through the ranks as power plant specialists or papermakers. They were going to have to take a risk with us. And if we ran into difficulties, they were going to have to stand behind us. In the end, that’s just what they did.

In mid-1979, management announced several key decisions regarding the new mill. First, engineering and construction would be done by Brown & Root, not by the internal IP resources. Second, all engineering and technology staff groups would have input into what Brown & Root would be doing but not final say. Third, the process control systems would be the dual-ROLMs provided by our in-house Process Control group. This would be the only exception to the “Brown & Root Rule,” above.

We were ready to do what no company in our industry (or any process industry that I know of) had ever attempted – design, construct, start up, and run a large manufacturing facility with full computer control and no manual back-up mechanism for running it should the computers fail to perform. International Paper had put full faith, indeed the economic success of a major division, behind something that only two years earlier knowledgeable people, some of them with computer backgrounds, said *couldn’t be done*.

Irvin Pickett, one of our engineers, recalls a day during the Texarkana Digester simulation work when a controls salesman came into our office to talk to Hugh Joyner, another engineer. During that conversation, Hugh mentioned our on-going work on a fully redundant system. The salesman scoffed and told Hugh that no one could design a system that could switch control from one computer to another in real time. Hugh walked him to the computer room, over to the simulator panel, and pushed the switchover button. The two computers exchanged places cleanly; and the salesman left the office shaking his head in disbelief.

We hit the ground running. I don’t think there was anyone in the development and start-up team who thought we couldn’t pull this off. We had caught Jim Chappelle’s vision and we were going to get the job done. That doesn’t mean that there weren’t plenty of skeptics in the company; but none of them was in Process Control.

Hardware, Software, and Staffing

By this time we had enough information to understand how we should split up the mill into sets of dual-ROLM systems. We had several options, but most of the “20,000 foot

view” elements came out naturally. We defined eight systems that would cover the entire mill.⁵

<u>System</u>	<u>Process Area</u>
1	Two Power Boilers
2	Two Recovery Boilers
3	Power Miscellaneous
4	Hardwood Continuous Digester
5	Pine Continuous Digester
6	Semi-Chem Digester / Lime Kiln / Caustic Plant
7	Paper Machine # 1
8	Paper Machine # 2

At the heart of each system would be a dual-ROLM, with all its fully redundant gear. Each computer in the dual pair would independently link, via RS-232, to a large Harris computer located in the mill’s process control office. These links would be used for gathering mill data for management displays all over the mill and for the management reports that would be daily needed. Supervisory controls would be resident in the Harris and would communicate with the ROLMs via the same RS-232 links. These supervisory controls would be written and implemented after the start-up of all the mill process areas, after any issues had been ironed out in the ROLMs.

I immediately began coordinating the work on specifying and ordering all the equipment we would need. This time we would not be relying on the old 1800 to generate code for the ROLMs. We would be obtaining our own ROLM 1666 with a disk drive in order to do our assembler development work. No more punch cards! We would actually be able to write our assembler code on the 1666 via black and white CRTs at our desks.

Another change was in the deployed ROLMs themselves. They would be equipped with 64K memory rather than just 48K. That might sound like a trivial matter in 2011, but it represented a third more than we had been working with and put us at 100% of what we could get with a 1602B at that time. Brent was skeptical. He was concerned that adding memory could actually *reduce* the amount of logic that we could cram into a ROLM. By proclaiming to our engineers that we now have “plenty” of memory, we would threaten our practice of insuring that code is written efficiently. He claimed that if you gave a programmer more memory, he would find a way to fill it up. Ever since, I have called that statement “Siegel’s Law,” and it couldn’t be more true than it is today with our personal computers!

If we took a step forward with more memory, some would say we took a step backward with a return to stand-up control panels. Prior to construction start, future Mansfield personnel visited the Pine Bluff digester installation, took one look at the operators sitting down, and told Brent, who was there at the time, that there was no way they wanted their operators sitting. Further, they requested individual stations with pushbuttons on the

⁵ There were certain parts of the mill that came from the equipment suppliers with their own control systems. Examples were the woodyard chip-handling systems, electrical generators, and the paper machine gauging systems. These were well-developed and, although they were not redundant, management quite rightly had no intention of having us “re-invent the wheel” and port those controls to the ROLMs. Still, the heart of the mill was totally dependent upon the dual-ROLMs.

upright parts of each control panel – the ones to the right of the ISC terminals would be for selecting loops and displaying loop status (loops on automatic had a green light on; loops on manual, red) and the ones to the left would be for starting and stopping pumps and motors with the appropriate green and red designations for status. This made the entire benchboard very long, even though the ISC terminals, the main operator and engineer interface to the ROLMs, took up very little space.

If the operator pushed a button in the motor start/stop group, that motor would go from on to off or vice-versa, depending upon which button was pushed, as long as the interlock program allowed it. Pushing a pushbutton on the loop select side called up that loop on an operator/engineer CRT.

Three ISC CRTs were inserted in the panels. Unlike the Texarkana system, all three



CRTs were connected to whichever computer was foreground. The two centrally-placed CRTs, one on top of the other, were for operators and engineers to do their work. A single keyboard accessed them both. The lower CRT could be switched to the Harris, so that an operator or manager could see data displays from other mill areas or initiate supervisory control action. The third CRT was inserted near the top of the panel and was for the displaying of error messages. Error messages could be operational such as “Pump failed to start” or “Interlock violation” or they could be system level such as “Background computer out-of-date.”

The photo, above, shows the two control CRTs with the error CRT (a bit hard to see) above the motor start/stop plates. This particular panel is one of two control panels

connected to System # 3 (the Power Miscellaneous system). The other control panel is at the opposite end of the control room.⁶

The systems in the pulp mill did not have the pushbuttons for loop select (see photo, below). Upon the insistence of management, instead they had auto/manual (A/M) stations, Moore 350s.⁷ These were truly unnecessary, but they did not affect full computer control. The computer outputs passed through them to the field devices. In the event that both computers were down, the only thing they could possibly do is allow an operator to open or close valves. The process would be down anyway, since all the pumps and motors would stop without the computers running.⁸



The interesting thing about all the pushbutton plates is that they were really keyboards. The operator/engineering entry keyboard *looked like* a keyboard. The set of pushbuttons that made up the loop select part of the benchboard, on the other hand, *didn't look like* a keyboard; but that's exactly what it was. It was the same thing with the motor start/stop part of the benchboard. Special cards inside the control panels converted each pushbutton

⁶ Each dual-ROLM system typically controlled two process areas via two control panels. System # 1, for instance, controlled the two power boilers, with each boiler having its own separate control panel. Not all the dual-ROLM systems, however, broke along such obvious symmetrical lines. System # 3, for instance, picked up a lot of the areas common to the entire power plant but not assignable to one boiler or another. Examples: water treatment, waste treatment, feed water pumps, and evaporators.

⁷ Some on the team believe there was pressure from a major pulp mill equipment supplier to put in A/M stations, since this is what their own field support personnel knew best.

⁸ The A/M stations took up so much space that they occupied both the left and right uprights of the panel. As shown in the above photo, the motor start/stop pushbutton plates had to be placed along the benchboard, on each side of the keyboard in the middle.

array to serial inputs for the ROLMs. As far as the ROLMs knew, they were just reading three keyboards!⁹

Besides the panel-mounted pushbuttons, there were other hardware issues to address at Mansfield. When switchover occurred, we had to take power off the analog outputs (AOs) and digital outputs (DOs) that were going to background and bring up power for the other set of parallel outputs. Jim Chappelle came up with a clever way of using triacs to do this (OPTO 22s, for noise immunity), without momentarily doubling the current going to the field equipment. Sets of I/O inside the panels also had to be switched. To make matters more interesting, System # 3, the Miscellaneous Power system, had something called super-remotes that had to be switched.

The water source for the mill was located 30 miles away – Toledo Bend reservoir. We called that location the river pumping station. Water was pumped from that site to a head tank that was 17 miles from the mill. Both of these locations had Computer Products gear that had to be redundant – the super-remotes. The super-remote sites communicated with the power house through serial links to modems and then through telephone lines. Those were the only modems provided by the phone company. Normally when we needed modems, we used Gandalf modems, little blue boxes that communicated with each other at 9600 baud. At one time we had so many of these devices sitting around the Mobile office, that I started to refer to them as “Tribbles,” after the furry little fast-multiplying blue creatures made famous in the *Star Trek* episode “The Trouble with Tribbles.” Alas, the name never caught on.

Switching the super-remotes became another challenge, and since I was the “switchover guy,” I picked up the software duties. During start-up, I was never allowed out to either of the two super-remote sites, since the team wanted me in the power house rack room where I could communicate by phone with the remotely-located engineer and make any software changes if there were problems.

And speaking of switchover, we had to define the hardware conditions under which an automatic switchover would occur. Clearly, total failure of a computer was a given. In order to sense this, each computer would be responsible for resetting a timer, called an operations monitor, or OPMON, about once per second.

Let’s suppose that the A computer is running as foreground and its OPMON times out. The B computer is always watching the A computer’s OPMON. If that timer goes to zero, the B computer would “kickdown” (take the power off) the A computer, forcing it to halt. B would immediately take over as the new foreground. The halted A computer could then be examined to find out why it failed to reset its OPMON. Memory board failure or power supply failure could be possible hardware failures. A possible software failure might entail the A computer entering an endless loop and not being able to get back to reset its OPMON timer.

The definition of other failure modes grew out of our continuing work in the office under simulated conditions. Remember the “dropping demonstration” failure during Jim’s

⁹ In later mill projects, the pushbuttons, the large stand-up benchboards, and the A/M stations were all dropped as unnecessary.

office presentation? After that happened, we began to monitor test cards inserted in every Computer Products I/O chassis. If the foreground computer failed to communicate with any test card on a routine basis, it would set a bit in a special memory location and signal the other computer to take over. Conversely, if the background computer could not communicate with any one of its test cards, it would set the same bit in its memory. That way if a switchover was called for, the background computer would be prevented from taking over. In short, any defined failure condition that could cause a controlled switchover could also prevent one.

As at Texarkana before, the key to a successful back-up scheme was to guarantee that both computers had exactly the same instructions and data in their respective memories. While the computers would be running different execution paths based upon whether they were foreground or background, the computers had to be ready to swap roles at all times. There were legitimate times when the software loads were different, but these were very specific, tightly controlled by the engineering staff, and didn't last very long. As long as there were differences or one of the two computers was down for maintenance, switchovers could not occur and could not even be attempted.

If the background computer wasn't available for switchover, you didn't want that condition to persist for long. It was something like driving down the highway with no spare tire; but even this analogy is not exact. No car can have its flat tire replaced while it's moving at 70 miles per hour down an interstate highway. Yet that's the kind of thing we were going to be doing. With a paper machine running at 2000 feet per minute, we were going to change out the control hardware; and the paper machine wasn't going to miss a beat.

In the meantime we had issues we needed to deal with that didn't directly involve the hardware and software, but certainly would affect the overall project's success. One of these was staffing. It was going to be up to our group to make sure that we left the mill with the proper people to run the systems after start-up, both the dual-ROLMs and the Harris. Jim Chappelle and John Botts, the other Process Control group supervisor, began to look at the people that we had on board already and determine if any of them would be willing to transfer to the mill staff. Our group would then have to hire a number of other engineers, either to fill out the mill staff or replace those who would be leaving ours.

Wayne Dobson, a senior Process Control employee, had been through many projects, dating back to the SPC-12 days. He was slated in as the first Superintendent of Process Control for Mansfield. Several other members of the group also agreed to take mill positions. Other engineers were hired from the outside or transferred to us from other mills with the understanding they would transfer permanently to Mansfield at start-up. All the newcomers were on board early enough to learn the ROLMs and how to configure them for controlling the Mansfield processes. We held classes for them in Mobile in the work we would be doing together – everything from hardware details, software layouts and debugging, and especially in configuring the various systems to run the processes.

By this time, ROLM itself started to take notice of what we were doing. The primary targets of the 1600-series were, of course, military applications. International Paper was

one of ROLM's very few industrial customers. Further, we were clearly doing things that even military contractors weren't attempting.

Brent Siegel and I went out to ROLM's headquarters in Santa Clara for a two-week training session on the 1666 and its capabilities. We soon discovered that we often knew more than the instructors, since we had been working with the system for months. In Mobile, we had discovered, for instance, that by sending certain control codes to another programmer's CRT, we could lock it up. So naturally we had to do that a few times



during lab sessions...to some of the other unsuspecting students.

Our chief sales contact during the Mansfield days was the ROLM regional sales manager, Ralph Petri, who worked out of an office in Virginia. Ralph stayed very much "in the loop" as far as the details of the dual-ROLM; and he talked to ROLM management and his other customers about our progress.

(That's Ralph in the photo, above. He's the one with the sport coat on. The others are, left to right, myself, Bill Campbell (ROLM software support), and Brent Siegel. We were horsing around in front of our Mobile office when this picture was taken.)

On one occasion Ralph arranged for Walter Loewenstern, one of ROLM's founders (he was the "L" in ROLM), to visit us in Mobile. On another occasion, he brought in Dr. Jim Mahaffey, a nuclear industry expert on the faculty of Georgia Tech, for a visit. After Mahaffey saw our plans and our progress, he revealed that such an innovative computer application as ours was totally foreign to the nuclear industry. He wished it were not so. The Three Mile Island incident, of course, had occurred earlier that year.

On September 12, 1979, Mobile had its own disaster to deal with – Hurricane Frederic was heading right toward it. A few staff members showed up at the office that morning. We knew how important the facility was to the Mansfield project, but there was little any of us could do to protect it, with the hurricane only hours away. Brent and I discussed the situation and then did what we thought would be the next best thing. We spent a couple of hours making sure that we had two up-to-date copies of each 1666 development disk, which were the large 15-inch platters. Brent took one set to his house and I took the other set to mine. The storm came ashore as a Category 3 in the early evening and by 2:00 AM the next morning what was left of the eye came over my house, which sustained no damage during the storm. Neither did Brent's.

The next morning, with power out everywhere, I made it into the office. When I got to our computer room, I found two of the engineering managers, one of whom was Ray Williamson, our top boss and an IP vice-president. I looked around, saw a minor amount of water on the floor near one of the windows, assured Ray that we had copies of all the

software at two different homes and it was safe. He grunted an acknowledgement, but I am not sure he understood the importance of what I was telling him. During the Mansfield start-up, however, he would begin to understand just how important software was.

Equipment protection was important too. Thus all of Mansfield’s control rooms and rack rooms were designed to be environmentally protected, as per the Pine Bluff model. Air lock doors and filtered air were the rule for both. In addition, we recognized that immediately following an unplanned power outage, no matter what the cause, we would need the ROLM systems to continue to run, since the operators would want to monitor the process variables as each area of the mill went to its “safe” state. For this reason, each dual-ROLM system was outfitted with an uninterruptible power supply (UPS). Each UPS was powered by a battery bank near its control and rack rooms; and the battery bank was housed in an explosion-proof room.

We were modifying our overall design as we debugged and configured the systems. Each engineer had certain prime responsibilities, but everyone helped out in other areas as well. It was a very intense effort but a complete team effort, with everyone working toward the end goal – a summer, 1981, start-up of the control systems. Below is the prime assignment table. Those in the shaded area transferred to the mill staff during control system start-up, with those who were configuration specialists becoming process control shift engineers.

Team Member	Assignment(s)
Jim Chappelle	Supervision, Start-Up Team Leader, Hardware
John Botts	Supervision
Brent Siegel	ROLM system-level software
Bill Richardson	Purchasing interface, ROLM system-level software
Wayne Dobson	Mill Process Control Superintendent
Tom Henderson	Interlock, digital control software, Mill Harris leader
Joe Brown	Power plant configuration
Bruce Slade	Power plant configuration
Steve Jex	Power plant configuration
Green Rives	Pulp plant configuration
Mike Creswell	Power plant configuration
David Janavich	Harris programming
Irvin Pickett	Lime kiln / Caustic / Semi-Chem configuration
David Hayes	Paper mill configuration
Bill Brittain	Pulp mill configuration
Frank Spafford	Harris programming
Dan Balzli	Hardware
Will Williams	Hardware, Harris programming
Stacy Starnes	Hardware

In keeping with the normal Process Control practice, the original intention with assignments was to give each department member a variety of responsibilities – from assembler programming in the ROLM and the ISCs, to FORTRAN programming in the

Harris, to configuring systems for the various process areas. Some of this worked out. Some of it didn't.

Stacy Starnes, for instance, didn't just work with the hardware. He also did some specialized programming. Bruce Slade wrote a special algorithm in the ROLM that was only used in System # 3. It controlled an innovative part of the waste treatment system. On the other hand, I was supposed to be heavily involved in configuring the power plant systems and Brent Siegel was supposed to fulfill a similar role with the pulp mill systems. Neither Brent nor I had time to actually do these things.

Brent was still making changes on the Pine Bluff digester system right up to the time when the Mansfield systems were scheduled to begin coming on line. When I went to the new mill for system start-up, he went to Pine Bluff to finish up there. As for me, during part of 1979 and 1980, I had renewed responsibilities at Texarkana, when the mill decided to add two additional digesters to the 16 they already had. It sounds simple enough, but I had "out-clevered" myself with the initial project. Since we were using a computer with 16 bits per word, I thought that I would set up much of the tracking of the digesters on a "one bit for each digester" basis. When digesters 17 and 18 were added, it forced me to go through the entire system, find all the places where a digester number was the bit number plus one, and add a second word for any digester number above 16. Not a big deal, but it was time consuming and required considerable retesting.

But what really forced me solidly into the system software camp and away from any possible participation in configuration was my responsibility for the switchover activity. It was a complex program, requiring me to think through scenarios involving the execution paths of two separate computers during actual switchovers. I would have to say to myself, "Well, OK. If the foreground computer does this, what are the ramifications in the background computer during that time? What could it be doing that would make the whole scheme come apart?" And vice-versa. So switchover itself became a real challenge to debug and it did have bugs, some of which weren't apparent till late in our simulation testing.

More importantly, whenever a failure of any kind occurred, it often looked like the switchover program was responsible. As I reflected on it later, it made sense. We had flags set that would tell us a lot about what was going on in a computer should anything go wrong – what function and/or activity was running at the time. The switchover often showed up as running. This was not particularly unusual. First, it ran every 20 milliseconds, repeating more often than any other program. Second, whenever a software failure occurred in any program, one that was significant enough to start overwriting memory locations or locking the computer into a tight loop, a switchover would be called for. Since the computers had identical code in them, the second computer would try to do the same bad thing the first tried, and both computers would be down. To the casual observer, it would look like the switchover failed when the failure actually occurred elsewhere.

I had to look more and more into other programs to understand and explain what had happened. There were dozens of functions and activities in the ROLM system, written by various members of Process Control. Some of those authors were no longer with the company or had moved on to completely new assignments. Brent and I had the

responsibility to make sure they all worked and didn't interfere with other programs or routines. We became the designated software troubleshooters for all the systems. During actual on-site start-up activities, management wanted one of us at the mill at all times. Most of the time we were both there.

We had a number of tools at our disposal for troubleshooting:

- Debug – this was the only program provided by ROLM that was loaded into each system. It allowed us to set traps and step through the various routines. It could only be used off-line or under simulated conditions, since hitting a trap would cause the computer to halt.
- Core dumps – If a system halted unexpectedly or stopped hitting the OPMON timer, we could dump the entire contents of memory to a floppy and examine any memory location or group of locations offline, comparing what was in those locations against what should have been there. Dumps were pretty much useless, however, if a computer went into a runaway state, where a program's memory locations were incorrectly altered by another program causing the first program to "go off into the sunset." Under those conditions, the flags usually pointed to the wrong program as the root cause of the problem.
- Gremlin catcher – This was a special routine that could be activated whenever a known core location was being inexplicably altered by a "mystery program." The catcher was often able to determine the identity of the offending routine. The computer was not halted when the trap was sprung; the gremlin catcher merely noted what programs were running during the catch.
- Switchover flags – As mentioned earlier, there were a number of possible hardware failures that would trigger a switchover. There were one or two words in memory where bits could be set that corresponded to such failures and could be examined using the front panel. Unless the hardware causes of these switchovers were fixed, the software would not allow a switch back to the earlier-failed computer system.

This last point illustrates a truism about the immunity to problems of the dual-ROLM system or any redundant processor system. It was pretty good in guarding against hardware failures and not as good in guarding against software failures, otherwise known as "bugs." As was noted earlier, identical software means the same problem exists in both computers. As time goes on and bugs are found and fixed, the number of bugs begins to approach zero (as long as the software never changes, of course). Any remaining bugs, however, become harder and harder to diagnose, find, and fix, because the exact conditions that activate the bug have not yet occurred or occur rarely. I once found a bug in a single ROLM system at the Moss Point, Mississippi, mill (now shut down) that had been there for eight years before the right conditions came about. Indeed there were several bugs at Mansfield that we didn't find until start-up was underway.

As had been arranged for in the past, one of the actual panels and computer systems was delivered to our Mobile offices for final off-site simulation. Prior to its delivery, we were using the same panel mock-up Jim Chappelle had built for Texarkana, modified for the Mansfield hardware, for checkout. Throughout this entire time of office simulation work,

we had multiple teams working on many tasks simultaneously. There was the configuration work on eight different control systems. This situation alone required that we come up with a new simulation methodology. In the past we were only working with one process, so the old hard-wired simulation equipment worked fine. Now hard-wiring was no longer practical since each process team would have to change their system to match the way the points were wired. This would have been time-consuming and could have led to errors in both the simulation and system itself.

Enter the ROLM-to-ROLM simulator, programmed by Green Rives, which eliminated the need for the hard-wired simulator. It used the parallel cable between the ROLMs to transfer some of the I/O tables both ways. Each system could now have a simulator specifically designed for mimicking its I/O points. The configuration team loaded their control software into one ROLM and the simulator for that system into the other. After that team was finished, the next team would do the same kind of thing. If we wanted to go back to the “switchover environment,” we could load both computers with the control software.

There was programming work going on involving special algorithms. There were fellows working on hardware testing and modification jobs. There were several system level debugging efforts underway. There was Harris development too. All these activities demanded time on the simulation equipment by up to 20 individuals a day. We had no choice. We had to go to around-the-clock shift work.

I volunteered for a 4 AM to noon shift. I figured it wouldn't be in big demand, and I was right. What I didn't count on was never being able to get away at noon. Since I was in the troubleshooting business, I found my services in big demand with those who came in at 8 AM. If I made it home by 6 PM, I felt I was lucky. It was all right. It just prepared me for the 14 – 16 hour days that were to come when we got the systems to the mill. On Easter Sunday, 1981, literally a few weeks before we had to ship everything to Mansfield, including ourselves, Brent and I worked all day in the office troubleshooting some system timing problems we were still fighting.

All of us on the team were dedicated to getting things right and getting them done on time. I think there was a general sigh of relief when we re-gathered at the mill in June. The simulation phase was finally over.

Start-Up: Mansfield Goes On Line

The hardware guys were the first on the firing line when we arrived on-site. They had to make sure that everything checked out from a wiring standpoint, especially in the power plant, where the first systems scheduled to come on line were the power boiler system and the power miscellaneous system, which controlled the water flow from Toledo Bend. Both Wayne Dobson and Joe Brown spent a lot of time at the river pumping station. It took about an hour by car to get there, even though the station was only 30 miles from the mill as the crow flies. When the call came in to the phone in the power plant rack room, I was often the guy that picked up that end of the connection, available to make any software modifications needed to communicate with the super-remotes and to test out the switching of its I/O.

One of the more interesting problems occurred with this super-remote I/O. For several weeks during hardware testing, the super-remotes for one of the paired computers would fail during the early morning hours then mysteriously start working again later in the morning. One of our engineers made several trips to the remote site, checked everything he could think of, but was unable to figure out what was happening. Finally, he decided to spend a night at the remote site to observe the failure. At about 3 AM a cooling fan cut off and the I/O went down at the same time. That I/O had been incorrectly plugged into a thermostatically controlled outlet that powered the cooling fan. When the ambient temperature dropped and the cooling fan stopped so did the I/O. When the temperature of the daytime began to reheat the station, both restarted. Diagnosis complete and problem solved.

The computers were running and the cabling well-checked out before the mill put fuel in the first power boiler; and it was indeed a milestone day when that finally came about. The power plant configuration team of Creswell, Slade, Brown, and Jex, with Wayne Dobson close at hand, went about tuning loops and modifying interlocks so that the boiler ran like the operators wanted it to run. Pretty soon it was time to add the second power boiler to the mix.

During the mill start-up, every one of us had a chance to shut down a running process at least once; and most of us did just that. There is a story that, in the early days of the power plant start-up, one of the fellows in the control room decided to push the panel's Emergency Stop button just to see if it worked. Each panel came with such a button, protected under a plastic cover so that it couldn't be pushed accidentally. It was there as last resort in the event that an operator felt that the computer system was out-of-control. The button would then trip relays and take all power off the I/O of *both computers* in the pair. Supposedly, it was one of the operators who tried out this button. I am not even sure the story is true. What actually *did* happen is that Joe Brown, while working under the panel, accidentally shorted out the switch, causing boiler shutdown. Joe picked up the moniker "Shut 'Em Down Brown" for that incident.

In my case, I remember quite well the circumstances under which I shut down the power boilers.

We knew that we would need to patch the systems on occasion without the benefit of a planned outage. We also knew that simply shutting down the background, entering the patch, restarting the background and then forcing it to become foreground would not work. The background would be "out-of-date" when it came up, so it could not become foreground without "bumping" the process or worse. Consequently we devised a clever process that *would* work. It ended up being called a hot line load.

Here's how it worked, as long as the change did not affect the core locations of certain key variable data tables, such as the DDC loop records or the interlock tables:

- 1) The control engineer stops data transfer from foreground to background, using an engineering function called "Stop R/R" to do this.
- 2) Engineer shuts down the background computer.
- 3) Engineer loads the new software into the background computer, either by patch or reloading the entire system build from floppy.

- 4) Engineer restarts the background and lets it run for a while. (R/R is still inhibited)
- 5) Engineer activates another engineering function, called “New System Update.” This causes R/R to restart, the variable tables to be transferred to the background, followed immediately by a switchover to the computer with the new software load.
- 6) Engineer lets the new software run until certain it is OK.
- 7) Engineer commands the foreground to download its entire contents into the background. This takes about a second. The background then has an identical copy of the new software, and the procedure is complete.

The procedure is only slightly more complex if the positions of the variable data tables have moved due to the new software being added. There is an additional step required to enter the offsets.

Because this was a multi-step process and would mean process shutdown if anything was done wrong, we had a policy of (1) management and the operators always had to know when you were going to do it and had final approval on timing, and (2) at least two process control personnel had to be involved, with one engineer double-checking the other.

Ralph Petri of ROLM had heard about this procedure and was amazed that we could make it work. After the power boiler system start-up, he came to the mill to see how we were doing. When he heard we had a hot line load planned for that evening, he was enthusiastic. He decided to stay overnight in Shreveport to watch us do it. The only problem was that Shreveport had few hotel rooms available, since there was a horse racing track nearby and all the rooms were occupied by race fans. It took us a while to find a motel with a vacancy for Ralph. I went with him when he registered. It was one of the sleaziest places I had ever seen – damp, dank, and dark; but by this time it was too late for Ralph to get a late flight out anyway.

The power plant people told us they would give us a shot at the hot line load at about 8 PM, so we drove back to the mill that evening. I cannot remember what the change had to do with, except that it was a coding change I had personally made in a non-switchover program. It obviously was important or we would have waited until a planned shutdown to put it in. It involved either System # 1 or System # 3, probably # 3, since that was the system with a lot of “specials” in it.

Finally, at about 9 PM, we were given the go-ahead. We began the procedure and made it through step 4, which is the easy part, because very few systems error out while in background. When we hit step 5, however, down went the system. We got the original foreground up again but the process was already down. As the operators got everything back running, I began looking at the code I had modified and immediately spotted the problem. I had written a “load” instruction using the wrong indexing accumulator. A “2” was supposed to be a “3” or vice-versa. One number off. That’s all it took. I clearly had done an inadequate job of checking my code.

The operators wouldn’t let us try again that evening, despite the fact that I had already found the problem. Poor Ralph wouldn’t be seeing a hot line load that night, and as far as

I know he never got to see one. Instead, he got to spend a night at the Sleazy Inn and fly back to Virginia in the morning. We did the hot line load the next day and all went well. As for me, I was grateful that the pulp mill hadn't yet started up.

Hot line loads became a very useful tool. So did another little program that we developed called core changer. It did just what it sounds like. It allowed an engineer to change a word or a short range of words in a running computer to whatever was needed. The words could be data, table entries, routine addresses, instructions – they just had to be translated manually into octal notation first, something that we were all used to. I had one experience using core changer on a live system, and it provides an illustration of the rare instance when redundancy protects against software mistakes.

Bruce Slade and I wanted to make a change in the RTX tables without waiting for a scheduled shutdown. Now the RTX tables were at the very heart of the operating system and were dangerous to mess with under any circumstances. But Bruce and I knew if we changed the foreground computer and messed it up, it would switch to the other computer with no problem (core changer ONLY changed code in the foreground). Sure enough we messed up, using the wrong numbers. We caused a switchover about three times before we figured out our mistake; but every time the other computer took over properly. We did this in the control room in full view of the operators and it didn't bother them at all.

The Scariest Three Weeks

During this same timeframe, we started experiencing unexplained shutdowns of the power plant ROLMs. First one ROLM in the pair would crash, the switchover would occur, and then the second would immediately crash. These crashes would happen at random intervals. Core dumps didn't reveal anything useful. The flags that were set did not suggest any sort of hardware problem. All we could do was reload the computers, go to the listing books, and start looking everywhere we could think.

The worst thing about the whole situation was that the mill went black every time we had such a failure. Sometimes the computers would fail two times in one day. Sometimes two or three days would pass without a failure. After a failure it would take 2-3 hours to get the power house back up. There was no power, no lights. Several hundred construction workers could do nothing until power was restored. It was a costly problem. The only thing mill management could do was make sure that there was plenty of diesel fuel available to operate the small emergency generator. Our process control start-up team brainstormed every possibility, thought of every potential problem, and combed through every possible scenario. Nothing stood out and all our searching of listings came up dry.

From the first day of our presence at the mill in June, Jim Chappelle and Wayne Dobson had worked out a basic duty schedule for staff members -- who would be on-site and when this person or that person could be spared for a few off-days back home. At first, as noted, it was the hardware guys who needed to spend the most hours at the mill. When the hardware was basically rung out, these fellows would take turns returning to Mobile to take a day or two of rest before coming back to the mill. After the boilers started to run, it was the configuration teams and the software guys who couldn't be spared as much, although we each had our times at home too. The idea was to make sure we had the right concentration of specialists on-site at all times.

During the developing crisis with the unexplained shutdowns, Jim Chappelle himself was back in Mobile several times. He was not only our team leader, he was also a Process Control group manager; and there were other projects underway being handled by staffers who had other assignments. These, too, needed some attention. Jim recounted returning to Shreveport by air and looking in the distance at the mill just before landing. If he could see steam coming out of the stacks, he knew the mill was running. If not, we were in another unscheduled shutdown. The next time I came in by air, I did the same thing. I saw steam and breathed a sigh of relief. By the time I arrived at the mill, however, the steam had stopped and the lights were out.

I went into the mill office building and found my way to the process control room. Glum faces all around. Pretty much the whole team had gathered in the office. I picked up a listing book, went over to the outside windows where there was enough light to read by, and started looking at some new possibilities that had occurred to me on my time “off.” Suddenly the office door swung open and in walked the mill manager, an experienced mill veteran named Ray Day. This was his last assignment before he retired. He looked worn out, and I am sure he was.

Ray walked over in my direction and asked the group: “Isn’t there anybody you can call for help? Somebody at ROLM?” In my calmest voice, as kindly as I could say it, I explained that we wrote the software -- all of it -- so....we *were* the help. There was no one we could call. Ray looked at me, nodded, then quietly turned and said to the group, “You guys had better get this problem solved.” He then walked out of the room and back to his office. He didn’t scream. He didn’t yell. I recalled what Jim had earlier told management: “There will be times when you won’t *want* to understand.” This was one of those times. It was clear that Ray wanted to explode. But much to his credit, he didn’t.

I had a similar experience later with Ray Williamson, the same fellow I had encountered back in Mobile after Hurricane Frederic. A couple of us were in the pulp mill control room troubleshooting another software problem. Ray had been silently observing us work, when suddenly he rushed at us and yelled out: “You know what you guys are?!!!” We stared at him and prepared for the worst. Ray caught himself just in time. “You’re wonderful!” he blurted, with a smile. We all laughed.

Actually, *the corporate and mill managers* were wonderful. They had trusted us when they gave us the job. Now, as much as it hurt to hold back their obvious concern, they signaled that they were trusting us to finish it. It is true that they had no choice, but they still could have yelled and hollered. Instead they let us do our job without any pointless harassment. They actually *did* understand.

Not everyone on our team went without harassment, however. It was the general practice in those days that when a new mill or a new mill process entered the start-up phase, the company would send in a number of process experts from other mills to offer advice and assist. These individuals would not be top managers. They would mostly be first-line supervisors who had been around a long time and who knew their specialty areas intimately. In this phase in this start-up, however, process knowledge wasn’t at all helpful. So mostly these men from other mills just sat around the control rooms

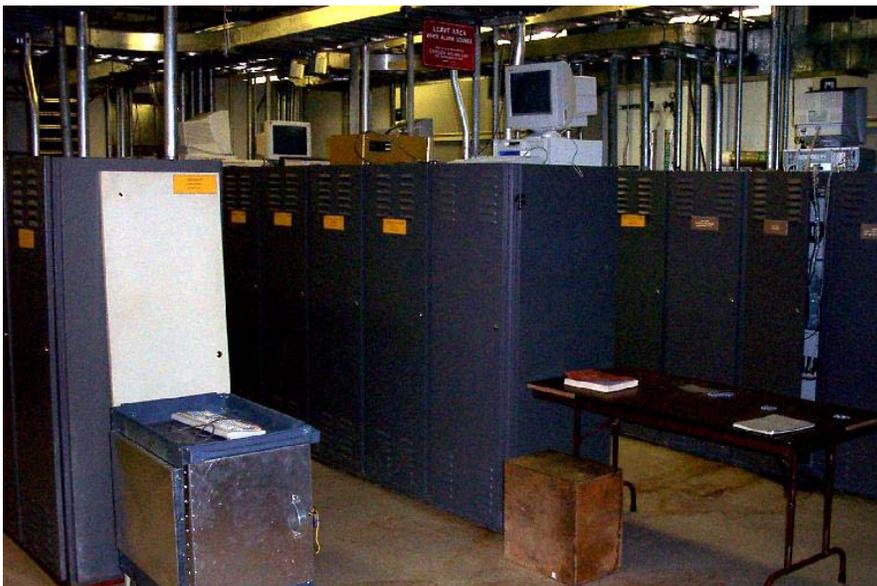
observing. They had nothing at all to offer, since the problems were computer problems, not process or equipment problems.

One such fellow, a pulp mill veteran, was sent from a mill in Mississippi. He did what he was told to do. He hung out in the pulp mill control room. Now it was pretty common knowledge that the architect of our full mill computer control concept and the man who finally sold it to corporate management was Jim Chappelle. As Chappelle busied himself on some hardware issues in the pulp mill panels, the observer sat there watching, telling Jim over and over, "Chappelle, you done destroyed this company. You have bankrupted IP. All of us are going to lose our jobs and you are responsible." Jim kept working, ignoring the taunts, which the man continued to pepper him with. Jim told me later that it was hard to work with that kind of thing going on. But Jim was a true leader. He was a real example to all of us of how to behave when times got rough.

Brent Siegel had just come off his assignment at Pine Bluff when he was immediately pulled into Mansfield to help with the power plant shutdown problem. John Botts, the other group manager from Process Control, normally stayed in Mobile to mind other things that were going on that needed day-to-day management attention. He came to the mill too. The message was clear to all of us, as if we needed to be told -- this problem was worrying management big time.

Wayne Dobson then gave us a clue. He reported that the operators had indicated to him that it seemed like they were pushing a button on a pushbutton plate every time the system crashed. Accordingly, I went to the power plant with another engineer to test all the buttons. My thinking was that one of the codes may have been misread or misinterpreted by the software, how I did not know. We made sure that the system was not currently controlling the process and that power was off the I/O. Then I very methodically and slowly pushed every button in sequence one at a time. The system kept running, so I concluded the button lead was a dead end.

We continued to review listings and chased every possible lead, even the wild, improbable ones. Often, first thing in the morning, Brent and I would grab the listing books and trudge across the mill yard to the pulp mill, where we would sit on the rack



room floor and try to think of what to do next, what to test on the running computers that were not yet controlling the process. (A typical Mansfield rack room is pictured in the photo to the left, although a few years after start-up.)

One morning, as we were sitting there, Brent dropped his listing book and started hollering, “Oh no! I know what it is. I’m going to get fired.” I pleaded with him to calm down and tell me what it was, assuring him that if he had found the problem, he wasn’t going to get fired. He told me that there was an error in the interrupt handling sequence that only involved one specific function, called increment/decrement function. Upon multiple interrupts calling this function, the computer could return to the program, after it had been suspended by an interrupt, with the wrong values restored to the registers. This would cause totally unpredictable results, including the computer wandering off into the sunset; and, if the condition continued after switchover, the back-up ROLM would encounter the same failure condition. I insisted that we test Brent’s hypothesis, so I went up the control room, repeatedly pushed the button as an operator would have, and both computers crashed. Then Brent put in a patch to fix the problem and I repeated the test. No failure of either computer!

As for the pushbuttons, my earlier testing had involved the wrong set. I *thought* Wayne had meant the motor starter pushbuttons were the issue when, in fact, it was the loop select pushbuttons all along. (Communication is a wonderful thing!) The pushbutton hardware was the same for both pushbutton sets, but the ROLM software was different. The loop select pushbuttons had two purposes. First, pushing either the upper or the lower button called that loop up on the ISC. Second, holding down the upper button caused the target of the selected loop to increment. Holding down the lower button caused it to decrement. Hence the involvement of increment/decrement function. Hence the multiple interrupts when a button was repeatedly pushed. Hence the crashes.

I was ecstatic that the problem had been uncovered, but I could understand why Brent was upset. The error was in a section of code that he had written, and it had been modified for Mansfield. He felt he should have seen this earlier. We made our way back to the office, announced to John Botts and the rest of staff that the problem was solved, and told them that we would have all the mill systems quickly patched. They were all fixed before lunch.

Wayne was extremely happy too, but he gave us a friendly hard time, saying, “I told you what it was!” True. Wayne had given us the right clue. I just didn’t follow up on it correctly. Pushing the buttons was the right thing to do, but I did an incomplete job.

This interrupt handler problem was the most serious we had to troubleshoot, since it went on for three weeks, we were reduced to guessing at what to do next, and management rightly became very nervous. After that, I cannot recall any major hitches in start-up that involved the computer controls. The pulp mill started up as planned, along with the recovery boilers.

David Hayes, who had worked on configuring the paper machine systems, developed an inspired way of getting the operators involved early. He utilized them in the early check-out phase prior to actual machine start-up. Intercom lines existed in all areas of the mill so that those in the control rooms could communicate with those in the rack rooms. He stationed one operator at the loop select panel and another at the motor/pump panel. Additional operators were placed in the rack room below. These operators had been taught to use hard-wired jumpers for controls simulation.

David's goals were to verify the correct assignment of I/O points, to verify all error messages ("Did not start", "Trip", "Level High," etc.) and demonstrate to the paper mill team that a no-manual-backup control system actually works. The papermakers agreed that going through this process made them feel better about the control system. The pay-back came at actual start-up. The paper mill experienced no internal issues. There was an overall increase in pushing the construction team to fix field-based control issues because the operators believed in the software and hardware system.

Finally, in December, the paper machines came on line and we were making product.

The Big Shutdown

As it became clear to everyone that the dual-ROLMs were going to be able to do the job, a number of process control personnel turned their attention to the Harris, the establishment of the data links to the ROLMs, the gathering of millwide data, and the generation of the needed management reports. ISC terminals were placed throughout the mill and on the desks of managers.



It was always part of the plan to implement supervisory, or higher level, controls on the Harris. (In the photo to the left, Green Rives and David Jananivich of Mansfield Process Control, work on the Harris.) These programs were to be written in FORTRAN and would be responsible for the optimization of the mill's processes. The first step in getting this done would be to establish two-way data links to each of the ROLMs, not only reading data from those computers but writing data into them, such as computed targets and setpoints.

I was not involved in any of the Harris work, so I had returned to Mobile for a while. Like all the others in the ROLM part of the project, I was on call whenever I was needed. Indeed, I did get a call; but it wasn't something expected. Wayne Dobson was on the other end of the phone when I picked up. He was in the power plant rack room and in a something of a panic. "What are you guys doing in Mobile to the power plant? The foreground goes down, it switches over, then before I can reload the original computer, the other one goes down. It keeps happening. The operators are trying to re-fire the power boilers before they lose steam pressure, but they are losing the fight."

I assured Wayne that we weren't doing anything to the system from Mobile (I don't think at that point we had any way to do that in the first place.). Wayne went back to his frantic work. Within the hour he called back.

Bill: "Did you save the power plant.

Wayne: "No. We finally lost it. And the rest of the mill with it."

Bill: "Did you find out what was going on."

Wayne: "Yes. The problem was coming from the Harris up here."

Bill: "What happened?"

What happened was that one of our Mobile staffers, a very good Harris programmer, Frank Spafford, had “out-clevered” himself. He was testing the data link software meant to make changes in the ROLMs, such as altering setpoints; but he had made an error and was changing some ROLM memory locations that shouldn’t have been touched. Unfortunately, he had picked the power boiler system to use as a test case. Even more unfortunately, he had decided that, if the first computer he attempted communication with didn’t respond, it was probably because one of our engineers had one of the pair down for maintenance. So his test software in the Harris said if it got no response from the A computer, try the same thing on the B computer.

Now I have learned over the years that computers do exactly what their programmers tell them to do. Down went the A computer, followed soon after by the B computer. To compound the error further, when neither computer responded, Frank tried the sequence again. And then again. The result was that the mill went black. Lines were plugged in the pulp mill, so once the ROLMs were re-started, it took 18 hours to get the mill back to where it had been.

I am sure there were managers who were very upset by what had happened; but, once again, there was no screaming for blood. Frank survived to return to Mobile and we established a new rule – if we needed to run test programs with the field ROLMs, we would disconnect the Harris-to-ROLM link to one of the computers in the pair first.

As I earlier stated, we all had our opportunities to shut down a process or more at Mansfield. Even so, the start-up for process control began about June, 1981, and finished at year’s end, with the paper machines on-line. At that time, it was the fastest start-up of a new mill that the company had ever experienced. That certainly wasn’t just due to the computers, but we were given credit for our contribution. Jim Chappelle was rightly honored by the CEO of International Paper for his innovative contributions in the whole Mansfield project; but he would be first to insist that the whole thing was a total team effort, and not just on the part of the process control guys.

Mansfield was a success because of the dedication of the full IP technology team – all the disciplines – pulp, paper, power, civil, environmental – and an excellent engineering and construction job by Brown & Root. Process Control had every reason to be proud of what it achieved from a computer technology standpoint. It was a “first” for sure.

It’s a Wrap – Final Reflections

International Paper management was so enthusiastic about the results of the dual-ROLM installations at Mansfield that they immediately moved us into another major project – the total reconfiguration of the Georgetown, South Carolina, mill. That mill, too, ended up with full dual-ROLM control systems. Shortly thereafter, there were major projects for us to do at Mobile Mill (now closed) and Pine Bluff (later sold and no longer owned by IP). At the end of 1981, when Mansfield first started making product, I estimated the technological process control lead of IP over its competitors in pulp and paper to be five years. By mid-decade, that lead was gone. Here’s some thoughts on why.

First, vendor-provided systems caught up. This was inevitable. I believe that our work actually spurred on the process control vendors. After all, we didn't try to hide what we were doing. We were up-front about that and told them that until they had such capabilities, we would continue to deploy our own system. Did they copy what we were doing? Not really. Vendors went a different way to get similar results. We always referred to the dual-ROLM as a DDC (direct digital control) system, wherein the failsafe element was present in a parallel duplicate. The vendors developed DCSs (distributed control systems), wherein the failsafe elements were handled by limiting the effects of failures to small control environments. The vendors did not copy us, but we gave them the challenge to think in terms of failsafe systems.

Furthermore, Moore's Law states that data density doubles every 18 months. This happens because (a) technology develops rapidly and (b) users are demanding more density because they need more memory for new applications. Conversely, they are able to write more applications because they have more density. Hence, Moore's Law is somewhat self-fulfilling.¹⁰ The result of all this added capacity both enticed and forced vendors into developing new software modules for control applications as fast as they could. It was cheap and all their competitors were doing it. All this accelerated the closing of the gap that I had perceived in 1981.

Second, the dual-ROLM systems, perhaps more than vendor systems, needed skilled engineers and computer scientists to configure them, modify them, and maintain them. These resources could be found in only one place – International Paper itself, either at the mill, in the corporate process control group, or borrowed from another mill. Once again, *we* were the help. A mill couldn't fall back on the large number of resources a vendor would have nor could it hire a dual-ROLM experienced engineer who could hit the ground running from the competition.

Third, initially our company did not staff up fast enough to take advantage of the computer technology lead we had. The platform was there, the data was there, but the personnel were not. Recall that Jim Chappelle's objective was "...to get some smart process and process control engineers to develop and implement new and innovative strategies for controlling the entire set of mill processes, optimizing the results for factors such as product quality, cost, and production throughput."

After start-up, Jim went back to the mill to interview Ray Day's successor as mill manager, Charles Harrison, as to his feelings about the computer technology the mill now had. When Jim asked him what he liked best about the system, Charles told him that he was extremely pleased with having the mill real-time data on his desk via an ISC terminal. He wouldn't want to lose that. When Jim asked him whether the mill had lower costs per mill employee than a conventionally controlled mill, however, he said he hadn't seen that.

If this was anyone's fault, it was ours in the corporate group for not having a pre-determined personnel plan to go forward with mill optimization. Our mills staffed for the

¹⁰ Siegel's Law would say that the heavier the data density, the sloppier the programmers get, since memory is cheap. That would be correct too. When we were working with only 64K of memory and no disk storage, we had to be very careful about our programs' efficiencies with regard to "density." We also had to work in assembler, which, except for some engineers working on imbedded systems, is a lost art.

day-to-day running and maintenance tasks that each facility required. They couldn't add engineers to their headcount just because they were bright and might come up with good optimization strategies. To compound this difficulty, our Mobile group was immediately assigned new engineering projects, and that took away from our ability to designate personnel for optimization work. By the time we figured out a workable strategy for doing this, it was 1983; and we had lost two years of the advantage.

That new strategy involved the formation of an entirely new corporate process control group, called Process Control Technology. John Botts became its first manager. This group had more than one assignment, but the key to its success was to get some of its employees into residence at the mills to develop and implement advanced computer controls on the Harris or its equivalent. This was done and met with some success. One of the very first control systems implementation engineers was Dr. Bob Taylor¹¹, who was hired from Canadian International Paper. He successfully implemented a "hang detection" algorithm at Mansfield that sensed ahead of time when conditions were building that would cause a continuous digester to plug up or hang, thereby shutting the digester down.

Fourth and finally, International Paper was in the business of making paper, not computer control systems. The original Process Control group was chartered and funded not only as an engineering group but also as a research and development group, experimenting with ways of utilizing computer technology in order to gain a competitive manufacturing advantage. With the advance of vendor systems, there was no longer a need for IP to experiment with computers as control systems, since there was no longer a competitive advantage in doing so. All this became quite clear in the mid-1980s, so we began a transition to vendor systems. There was an attempt to standardize on a single vendor for the corporation, but we could not get consensus on which vendor that should be.

But what about the dual-ROLMs in the field? Upgrading certain elements of them started almost immediately after Mansfield. The ISC terminals, for instance, turned out to be unreliable and were quickly replaced by similar terminals from HMW. The ROLM front panels later gave way to PC-based CRTs and flat screens. The Harris was replaced by the VAX line. As for the heart of the system itself, the 1602 computer series, ROLM was purchased by IBM and later Loral, and the 1602 series was eventually discontinued. We needed a transition path for our dual-ROLM systems.

Several mills quietly lobbied for our group to convert all the ROLM code to Microsoft code running on PCs. This would have been the wrong thing to do, since there would have been no competitive advantage to IP. Instead we contracted with Moore Products to convert the ROLM logic to run on their APACS hardware platform, with an eye on transitioning all the ROLM systems over to true APACS over a number of years. The project became known as the Dual-ROLM Conversion or DRC, and the resulting control equipment is still known as a DRC.¹² The ROLM mills have had the option of going this transitional route or scrapping the ROLMs outright and substituting a DCS of different

¹¹ Bob eventually made his way into IP manufacturing management, spent some time as a vice-president of SAPPI, and is now CEO of Neucel Specialty Cellulose in Vancouver, BC.

¹² This was a monumental task in its own right. Brent Siegel headed up the effort for International Paper, directing the work of several engineers at Moore.

manufacture than Moore (now part of Siemens). There are mills that are still running DRCs.

But are there any of the original ROLM systems still in service? Since I retired ten years ago, I had to ask. What I was told amazed me greatly. Incredibly, the first computer pair deployed at Mansfield, System # 1, controlling the power boilers, is still in service. It is the last remaining dual-ROLM system in the company. I am told that the processors are running just fine, but the degradation of the equipment attached to those processors will force the replacement of the entire system. That is scheduled to occur next year, 31 years after start-up.

The Mansfield project was the highlight of my 26 years with International Paper. I was part of a team that did something that had never been done before. We were blessed by having a great team (and this *was* a total team effort) led by a great leader, Jim Chappelle. He challenged us constantly to excel but had a real collegial style when he did it. It was hard work and we had our anxious moments. But Jim never lost faith, so neither did the rest of us.

William E. Richardson
Retired Manager, Process Control Technology, International Paper
January, 2011

William E. (Bill) Richardson holds bachelor degrees from Detroit's Sacred Heart Seminary (scholastic philosophy) and Flint's General Motors Institute, now Kettering University (electrical engineering) as well as master's degrees from Auburn University (electrical engineering) and the University of South Alabama (English). During his professional career, he worked for GM, Teleflex, ITT, Nartron, and finally International Paper, from which he retired in 2001.

Appendix A: Additional Photos from Georgetown

As is mentioned in the main article, the Georgetown project immediately followed the completion of Mansfield. The following photos are from the Georgetown mill, except where noted. They are included here because even the Georgetown photos depict much of what is discussed in the article regarding Mansfield. Differences are noted under each photo.

CONTROL ROOM AND CONTROL PANEL I/O



The photo, above, is of the Georgetown power plant control room, taken at the time that project was completed in 1983-4. The dual-ROLM control panels closely resemble those at Mansfield, although they are not identical. The two operator/engineer CRTs are side-by-side instead of stacked, as they were at Mansfield. Each CRT has its own keyboard and the operator has the choice to sit down or stand up. The pushbutton plates can be clearly seen in this photo, both to the left and above the CRTs.



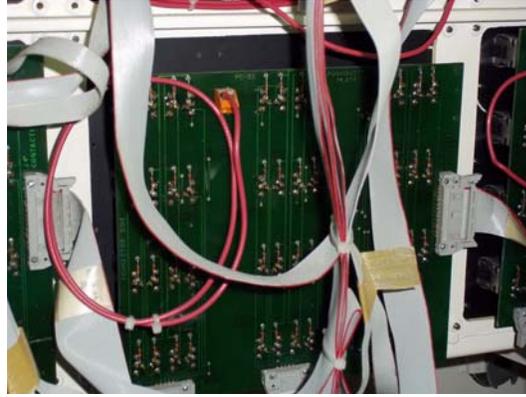
Here's a later photo of a Georgetown panel, looking much the same as it did in the 1980s (this photo was taken in the bleach plant). The biggest difference is that this panel is no longer run by a dual-ROLM but by a Siemens/Moore DRC. The logic of the dual-ROLM was ported directly to the DRC, so the control software and the system logic, including the switchover system, works the same way.

Note the two keyboards. These are spill-proof custom-made membrane keyboards designed specifically for the dual-ROLM system, now used by its successor system, the DRC.



Above is a close-up of the keyboard used at Georgetown. Mansfield started up with plunger-type keyboards, then later switched to membrane keyboards like this, but with a different layout (see photo, below).





These photos show two pushbutton plate inserts from the front and one as it looks from the back. The same type of plate is used for both loop select and for motor starters. There are two contacts in each button. For the motor start/stops only, *both* contacts have to make for the logic to say the button has been pressed. This will prevent the system from taking an unwanted action if a contact goes bad. If a contact goes bad, that actually disables that button and it must be replaced.

The Georgetown system uses a 16x16 matrix to identify which button is pushed. A microprocessor located in the control panel scans the contacts by firing digital outputs and reading back the matrix of digital inputs. The earlier Mansfield system did not use a microprocessor but a series of serial duplex cards to read the codes and transmit them to the ROLMs through the interface chassis.



Inside of a control panel you can see the redundant I/O used to read the keyboards and pushbuttons and turn the lights on and off. The I/O for the A system is on top, the B system on bottom.

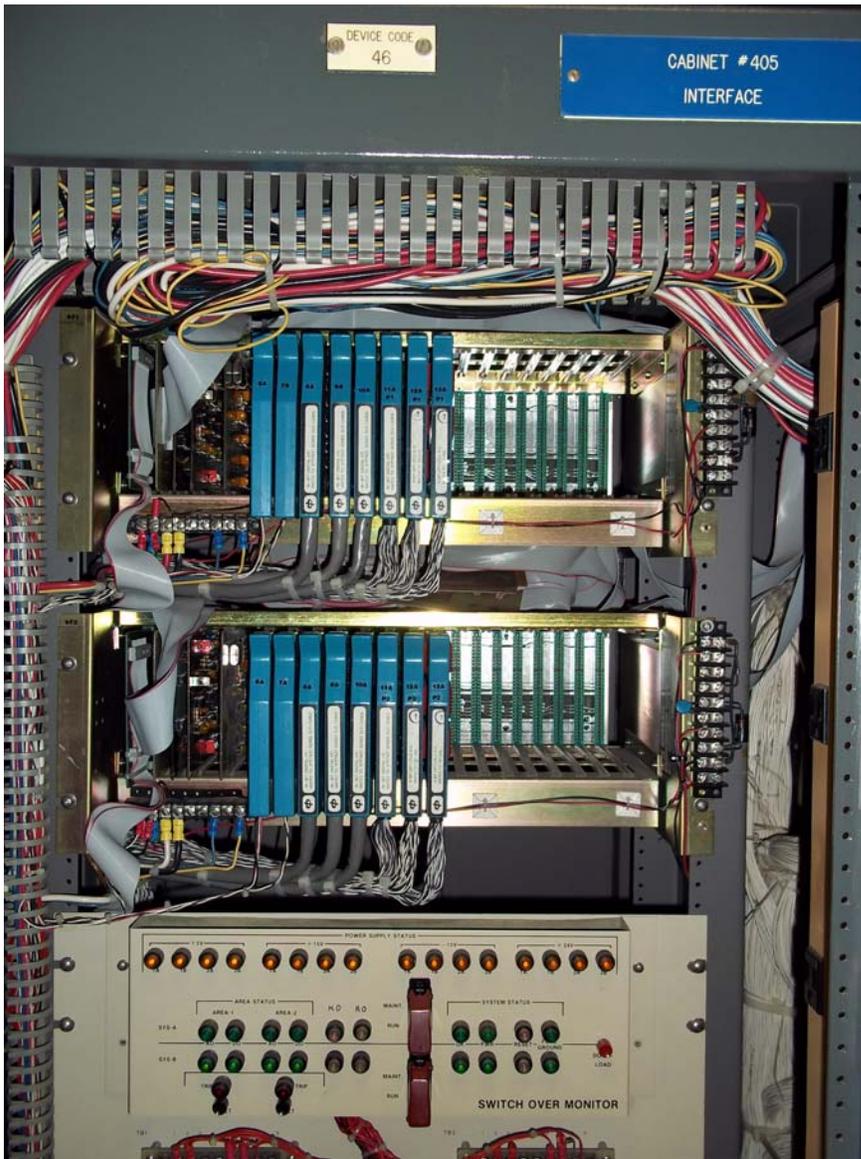
RACK ROOM CENTRAL BAYS



This is a photo taken of the bottom part of the computer bay of the last surviving dual-ROLM system, Mansfield System # 1. The two ROLM computers can be seen at the bottom of the bay, with their associated front panels above them. Originally at Mansfield there was only one front panel per computer pair. This made it often necessary to disconnect a large mil-spec connector from the A computer and reconnect it to the B computer and vice-versa. I can personally attest to the fact that doing this frequently was a real knuckle buster. So the second dedicated panel was added later. It's more clearly seen on the next page.

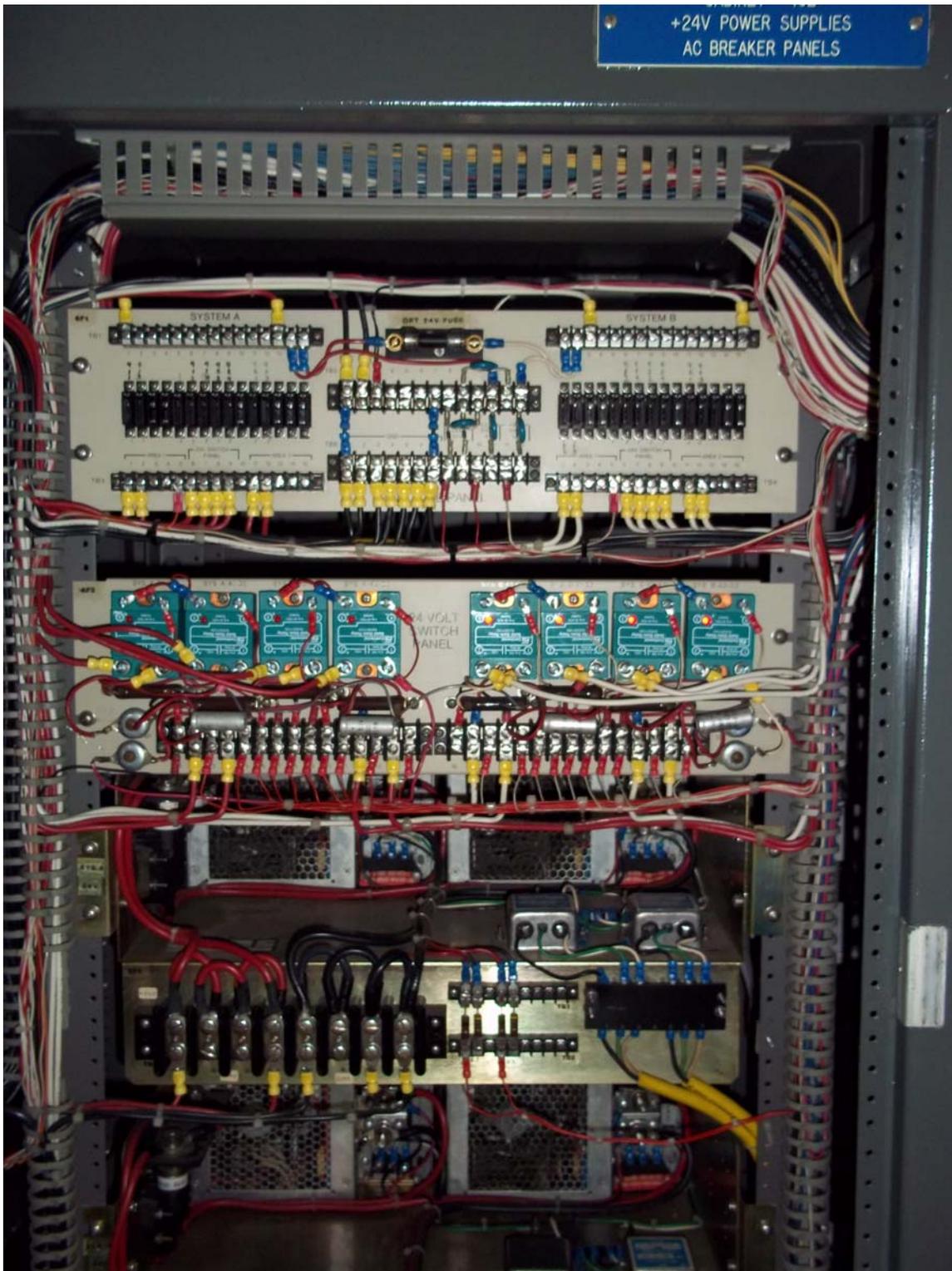


The upper part of the same bay shows the two front panels better. Above them are the two IOBCs, which interface the ROLMs to the Computer Products I/O gear.



This is a switchover cabinet in the main I/O set. The chassis above the switchover panel are the interface chassis. They contain the I/O to talk to the switchover relay, read the status of the power supplies, and the OPMON cards.

Status lights let the engineer know what is going on with the power supplies and which computer is currently foreground. Since this is a photo of a DRC at Georgetown, the original Mansfield dual-ROLM installation would have looked a bit different.



This photo shows a 24-volt power supply relay bay. The foreground computer uses these relays to power up its AO and DO cards to reach the field devices (and power down the corresponding relays for the background computer).



This photo shows a bay of AO cards and their connectors. Note the identical cards in identical chassis. The top two chassis are connected to the A computer; the bottom two are connected to the B computer.

Appendix B: Kudos and Credits -- Dramatis Personae

The following people greatly contributed to the success of the process control part of the Mansfield capital project and/or are mentioned by name in this write-up.

<i>Mobile Process Control – Corp. Eng.</i>	<i>Mansfield Management & Process Control</i>
Jim Chappelle	Ray Williamson
John Botts	Ray Day
Brent Siegel	Charles Harrison
Bill Richardson	Wayne Dobson
David Hayes	Tom Henderson
Irvin Pickett	Bruce Slade
Frank Spafford	Steve Jex
Bill Brittain	Joe Brown
Dan Balzli	Green Rives
Stacy Starnes	Mike Creswell
Bill Wellborn*	David Jananivich
Charlie Jolliff*	
Ed Johns*	
Richard Jackson*	<i>From IP Process Technology</i>
Hugh Joyner*	Bob Taylor
Beverly Sims – Dept. Secretary*	
	<i>From ROLM</i>
	Ralph Petri
	Bill Campbell
	Walter Loewenstern
	<i>From Georgia Tech</i>
	James Mahaffey

* Not directly involved in development/start-up of Mansfield but made essential contributions resulting in a successful project.

Of those listed, above, Stacy Starnes, Beverly Sims, and Joe Brown still work for International Paper. Bill Brittain functions as an independent contractor to IP for the ROLM/DRC systems. The rest of us are either retired or in careers with other companies. My apologies to anyone that I have inadvertently left off the list.

Those noted in **bold type** were given the opportunity to review this article before it was submitted. Many of them provided invaluable input to me as I wrote. In this regard, I am especially indebted to Bill Brittain, Brent Siegel, Wayne Dobson, and Jim Chappelle. We are older now and our memories are not all that good individually; but collectively we were able to pull together this history.

Bill Richardson