`Cholla`: A Computer Assisted Silicon Processing Facility

Overview

by Douglas K. Brotz and Charles M. Geschke

April 1980

Abstract: Cholla, a system providing computer assistance to the silicon wafer processing facility at the Integrated Circuit Laboratory of the Xerox Palo Alto Research Center, is described. Brief descriptions of the environment in which the system will be used and of the hardware and software components of the system are given. The goals of the Integrated Circuit Laboratory supported by Cholla are twofold: 1) to provide a fast turnaround silicon wafer fabrication facility supporting designers of VLSI components, and 2) to support VLSI process research. Cholla provides this support, allowing variability in individual wafer processing, through its management of information associated with VLSI fabrication, precise control of silicon processing equipment, and capture of data associated with such processing.

XEROX

Palo Alto Research Center
3333 Coyote Hill Road / Palo Alto / California 94304

Introduction

The Integrated Circuit Laboratory (ICL) of the Xerox Palo Alto Research Center has been created to meet several long range goals of the corporation. A central activity of ICL is the management of a silicon device fabrication facility. This fabrication line will serve both as a quick turnaround facility supporting research in the design of electronic components implemented via Very Large Scale Integration (VLSI) and as a VLSI fabrication process laboratory supporting research in the technology of making VLSI devices.

The state-of-the-art integrated circuit processing to be performed in ICL requires a sophisticated system for its control. The "clean room" environments used during integrated circuit (IC) fabrication preclude the use of dust creating paper--video displays with their associated computers are much preferred. Support of design analysis and VLSI technical development requires the capture and storage of large amounts of operational and test data. To fully realize the potential of new techniques in IC processing, detailed and complicated "recipes" will be developed that are best applied via direct computer control of IC processing equipment.

Cholla will provide information handling facilities and computer control of silicon processing equipment. The facilities provided will include:

1) A language for IC processing run specification and documentation.

2) Computer assisted operation sequence control.

3) Capture and storage of operational and test data.

4) Computer control of equipment parameter setup and verification.

5) Equipment status and performance reporting.

6) File facility to provide stored information for analysis.

Cholla will be designed and built by members of both the Computer Science Laboratory and the Integrated Circuit Laboratory. This document will serve as a guide to the framework of the complete system. A major consideration in this design is proper phasing of the implementation. It is not the intention of the designers to provide every possible feature within an initial system, but rather to provide a working system that will allow new features to be added as their utility becomes known. Components and features of Cholla will be implemented as resources permit. Initial services will be provided through modification of and additions to existing PARC hardware and software. The Cholla design is such that further pieces of the system, e.g., data analysis, may be added in a straightforward manner.

Plan of this Document

There are several ways in which Cholla may be viewed. One may view it from the perspective of a process line operator, process researcher, hardware installer, software maintainer, etc. Each of

these perspectives will be briefly touched on in this document, but not in as great detail as would be expected by any particular perspective. More detailed information will be available in several other documents, including a Cholla User's Manual and a Cholla Software Design Document. Cholla may also be viewed as a software system, a hardware system, or a PARC project. Distinctions between these perspectives are deliberately blurred in this document. This document will give a consistent overview of the entire project, and should be the first document to read for anyone interested in any aspect of Cholla.

The remainder of this document will be concerned with introducing common terms and their definitions, and with descriptions of system components, both hardware and software. As with any complex technology, there are numerous concepts that must be understood before one may understand particular systems that deal with that technology. Thus, descriptions of the VLSI process line environment and of the computing environment in which Cholla will be implemented will be given next, followed by descriptions of Cholla, which operates in those environments.

The Process Line Environment

The minute tolerances required during the VLSI fabrication process require very specialized equipment and skilled operators. The equipment employed includes furnaces, ion implanters, wafer scrubbers, acid etchers, and sophisticated test equipment, to name just a few of the machines involved. Impurities in the substances used can have such a disastrous effect on the end product that most of this equipment is used within a *clean room* environment. A clean room is a restricted access area in which dust is kept to a minimum to avoid contaminating wafers in progress in the fabrication process. In such an area, operators must wear protective clothing that minimizes the possibility of dust entering the clean room environment. Paper, which creates a significant amount of dust, is prohibited in such an environment. Electronic systems that carry instructions to operators and to machines are virtually required in these clean rooms.

Silicon devices are constructed on *wafers*, small disks which are later broken apart into many separate, identical *chips* when the processing is complete. Usually several wafers are processed together and are carried in plastic *cassettes*, each of which can hold about two dozen wafers. The entire process of fabricating these wafers usually requires on the order of one hundred individual *steps*. Each step is performed at some machine on the fabrication line. These machines are capable of varying the steps they perform according to certain *parameters*. For instance, a furnace step may vary in the length of time or particular temperature at which wafers are baked. Parameters are input to the silicon processing equipment by means of dials manipulated by the operators or directly through electronic connections. The entire sequence of steps along with the particular parameters used to produce a silicon chip is called a *process*. A *run* consists of one or more cassettes of wafers with a certain set of designs applied to pattern those wafers carried through a particular process.

At several critical points in a run, measurements are made to insure that certain steps have been performed correctly and that the process is proceeding properly. Although most sections of each wafer are given the same pattern, that of the device being produced, some areas on each wafer are used for test patterns which will be used in the testing phases. Some of these *in situ* tests performed during processing are destructive to the wafer, and thus a run normally includes several extra wafers that are carried through the process solely for this kind of testing.

After the silicon processing has been completed, several *parametric* tests are performed on the remaining wafers to check that the devices have been produced successfully so far. Parametric tests are used primarily to check general electrical properties of the finished wafers and are reasonably performed at the silicon processing facility. Additionally, *functional* tests may be performed to determine if the finished circuit properly performs the task for which it was designed. Functional testing is highly circuit dependant, and may sometimes be left to the circuit designer. After testing at the processing facility, the good wafers are broken into their individual chips which are then *bonded* and *packaged* into their completed form, suitable for inclusion on a circuit board.

At nearly every step in a run, information must be recorded on exactly how that step was performed. This information will later be consulted by process researchers to determine how to improve the processes. As mentioned above, pencils, pens, etc. cannot be used in the processing environment. Electronic systems for recording such information are required.

For a wide variety of reasons, not every section of a wafer that is destined to become a finished chip will actually be fabricated properly. The percentage of good chips that are produced in a run is called the *yield*. One of the main goals of process researchers is to find processes that result in improved yields. Flaws in circuit logic design (as opposed to flaws in circuit manufacture) that result in the silicon device not performing as desired are not the concern of the device maker, and do not affect this yield measurement.

To an electronic component designer, a silicon processing line is a place to which a design is submitted and which returns packaged silicon chips that implement that design. Such a person is for the most part unconcerned about the actual workings of the processing line or even with the details of the specific process used to create these chips. At a gross level, this designer is interested in the general process used, such as NMOS or CMOS, as these choices restrict the particular logic design techniques that may be employed. The designer may be interested in properties of the process that affect the speed and power requirements of the resulting devices as well. However, beyond the choice of the general process and some general goals, the logic designer usually cares little about the specific parameters used during the production of these chips.

The component designer is concerned with the turnaround time required for the production of these chips. Typically, several iterations of the design process followed by device manufacture will be required before a properly operating device is made. For these reasons, a quick turnaround processing line is desirable for a research environment. The customers of the ICL processing line will generally not be concerned with high volume production, but with fast, high quality, limited production.

Thus, there are many requirements placed on the operators of the processing line and on the process researchers. Each run through the line will be different in many ways, with varying processes and designs. Process researchers must be able to gather and analyze information regarding the performance of the line and to quickly change its operation. Operators must be able to apply varying processes to their individual runs without reducing quality.

Two ways of using operators on a silicon processing line are possible. These are called the *specialist* and the *modelmaker* approaches. In the specialist approach, each operator is used for specific steps only, similar to the traditional operation of an assembly line. This approach may work well for high volume production lines, where the process applied never varies. On the other hand, the modelmaker approach allows an operator to carry a single run through every step in its process. This approach is considered to be preferable in a highly variable processing environment as in the ICL, and it may well instill more pride in craftsmanship of individual runs. Although there may be specialists operating a few of the machines at the ICL, it is primarily the modelmaker approach that will be employed in the ICL processing line.

Another aspect of the processing facility is its rapidly changing nature. Silicon device fabrication technology is currently changing constantly, as new processes are discovered that provide denser packing of components on a chip with better yields. The equipment used on the ICL processing line will change frequently, with the associated requirement that a system used for its control be able to adapt to these rapid changes.

Life History of a Run

To better understand the requirements of a system to control the ICL processing line, the following model of a run's history is given.

Before a run begins, several events will have taken place. A circuit designer will have prepared a cicuit layout using various Computer Aided Design (CAD) tools. This will result in a set of *masks*, patterns to be applied to the silicon wafers at different steps in the fabrication process. These masks will be applied to form the different layers of silicon which together will implement the designer's device. Most likely, the designer will have simulated the performance of this device, but real testing will await the actual delivery of the finished chips. (The CAD systems are not a part of Cholla; they are being built outside of this project.)

When a design in the form of a mask set is ready, the next step is to select the process to be used to produce the silicon wafers. Usually a process researcher will make the decision as to precisely which process to use, although the circuit designer may participate in this decision. A process definition will be entered into Cholla, along with additional information such as test programs, mask set, packaging requirements, and cicuit designer identification. A process definition includes all details regarding the processing steps for the run. These details cover which machines

will be used in which sequence and all parameters to be used at each step. Some steps will be performed under direct computer control; in these cases, the recipes (programs) for interacting with each piece of equipment thus controlled will be specified. The total information entered regarding these steps and parameters thus constitutes the *run definition*.

The run definition is then entered into a priority queue of *pending runs*. The manager of the processing line will make the decisions regarding when a run will begin. Among other things, an operator will be assigned this run when it is activated. When the run is activated, it enters the set of *active runs*. While in either the pending or active run sets, this run's status (current step number, etc.) may be monitored by the processing line manager.

After being activated, the run is in the hands of its operator. Its wafers will be taken from step to step by this operator, who will perform all or nearly all actions. These actions include the building up of new layers of silicon on the wafers as well as the *in situ* tests. Cholla will be consulted by the operator for exact instructions at each step. All steps will require information to be input back into Cholla regarding its performance. Such information as the time at which the step began and ended will be entered by Cholla automatically. Other information specific to each step will be entered by the operator. Some steps will have direct control of input to the processing equipment from Cholla. Additionally, Cholla may receive results for some steps directly from the processing equipment. At each step, the process line manager may query Cholla to find out the status of this run.

When the wafers have been completed, they are transferred to the parametric testers. The steps to be performed for such testing are also listed in the run definition held by Cholla. A large amount of information will typically be generated during the test phase. This information too will be stored by Cholla. Packaging and possibly more testing will also be performed. These steps, like the other steps, are also controlled by Cholla. Finally, the packaged chips will be finished, with all information regarding their creation saved on the Cholla file store.

Not directly involved with a particular run, but important to the process researchers nonetheless, is a large amount of general data. Such information as environmental factors, e.g., clean room temperature and humidity, may be entered into Cholla's files. Other information regarding materials used at different times, etc. may also be entered.

The Computing Environment

The ICL is a laboratory in the Xerox Palo Alto Research Center. At PARC a rich and unique computing environment has been in place for some years. Key components of this computing environment are small, personal computers, linked to one another by means of a passive communication network called the *Ethernet*. The Ethernet, a simple length of coaxial cable, on which each conected computer may send and receive packets of information, makes *distributed applications* possible.

A distributed application is a computer system that is made up of components that run separately on several different computers. A typical example of such a system is one in which the function of reading and writing information on disk files is separated out to one computer, while the specific application program runs in a different computer. The application program computer then reads and writes to its disk files by means of communicating with the remote filing computer, which performs the actual file functions itself. Such a remote computer providing a specified service is called a *server*.

There are several advantages to designing computer applications that make use of such servers. For instance, a file server as described above could have considerable code devoted to specific filing functions, such as backup and crash recovery, which a single computer application program might not have space or time to do properly. A considerable amount of experience with systems using these kinds of servers has been gained at PARC, and the systems have proved to be simple to understand and extremely reliable.

The type of computer that we will use in Cholla will initially be the *Alto*, a personal mini-computer designed and used at PARC. Newer computers, particularly the *D0*, are being built and will offer better performance in a machine that is compatible with the existing network. If and when newer machines are designed and built at PARC (with the ICL playing a significant role in their design and manufacture), they can be incorporated incrementally into Cholla. A schematic diagram of the initial hardware configuration for Cholla appears in Figure 1 at the end of this document.

In the sections that follow, individual hardware and software components of Cholla are described. Note that several of the components are servers, rather than user machines. Some of the components are new, e.g., the Equipment Controller. Other components are standard parts of the existing PARC network (with some modifications), e.g., the Mail Server, Data Line Scanner and File Store. Much of the software in Cholla will be derived from the Laurel Message System program. That program provides filing, display and editing of information in a convenient and reliable manner. Figure 2 at the end of this document shows a sample display screen as presented by Laurel. Laurel's style of information display will be used by several software components of Cholla, particularly by the Process Follower software that that is used by fabrication line operators.

Functional Components

File Store

The File Store is a central component of Cholla, providing storage for all of the information used in the system. A *remote access protocol* will be used to access this information from the various other machines in the Cholla system. Disk files, in which Cholla information is stored, may

be stored locally (on a disk connected directly to a processor running a program that uses the file) or remotely (on a disk connected to some other processor in the network). Remote files offer advantages in that information is sharable, i.e. it may be accessed from several different computers, and that information is stored in only one place, insuring that the information is always correct and current. When a portion of a disk file is required by some computer in Cholla, a request is sent to the File Store, which will return the specified portion or write upon the specified portion as requested.

With the File Store used in this manner, several separate computers may read the same information at the same time. Only one computer may modify this information at any given time. The File Store machine will make sure that the information used throughout Cholla is consistent and checkpointed to reduce the possibility of disruption due to failure of any system component.

The current candidate systems for the Cholla File Store are IFS (Interim File System) that is in current use in the Xerox Internet and the Juniper File System, an experimental filing system being developed at PARC. The choice of which file store to use depends on several factors, among which are availability of suitable hardware for either system, availability of software usable in Cholla and maintenance of such software. The other parts of Cholla are designed in such a way that either kind of file store will be compatible with Cholla.

Printer

An electronic Xerographic printer, probably a Dover or Penguin printer, is included in Cholla. The printer will be used by the scientists working in ICL in the ways that computer printers are usually used. In addition, the printer provides an important backup facility for the operation of the ICL fabrication line. Standard procedures will include printing the run definitions for each active and pending run at frequent intervals (say, once per day). In case of a system failure, the run definitions will be available, and the fabrication line will be able to proceed in a manual mode, much the same as it operates today.

Eventually, a color printer (Puffin) may be added to the ICL computing environment. Output from the CAD systems may be printed on this printer, providing better graphic indications than black and white printers can. No support for color printers is included in the initial Cholla implementation. Such support may be added during later stages of the Cholla project.

Process Followers

Operators on the fabrication line will use the computers indicated as "Process Followers" in Figure 1. These machines will read the run definitions as stored on the File Store, and will be used for input of results into the run definitions by the operators. When direct control of a piece of process line equipment is provided, initiation of such control will be performed at a process follower.

These process followers are redundant with one another in that all such workstations may be used by any operator interchangeably. Storage of the run definitions on the File Store makes this possible. There will be several process follower workstations in the clean room. The number of these workstations required depends on the number of operators expected to be working at any given time, on the layout of the fabrication area, and on the expected congestion at any location in the fabrication area. The general criterion for placement of process follower workstations is that any operator should have convenient access to a workstation at any step in the fabrication process. As different areas of the fabrication line are separated by doors and large pieces of equipment, and more than one operator is expected to be working in the same area at some times, several workstations are required. In the initial Cholla installation, the number of process follower workstations in the clean room will be five.

Equipment Controller

This computer is responsible for the direct control of individual machines on the processing line. Several pieces of process line equipment are capable of receiving parameters electronically and of sending back results or of being monitored electronically. The Equipment Controller will interact with such machines in the style appropriate for each machine.

Many of the process line machines are capable of direct connections via an RS 232C interface. The RS 232C links will connect directly to another component of Cholla, the Data Line Scanner, which itself is connected to the Ethernet. The Equipment Controller will communicate with such a process machine via the Ethernet, through the Data Line Scanner machine and the connecting RS 232C link (see Figure 1).

The Data Line Scanner is an existing machine in the PARC network. It has the capability of connecting to up to 64 different RS 232C links. It is capable of processing the signals in and out of these links and of receiving and sending packets on the Ethernet, but not much more. The "smarts" involved in interacting with the process line machines will be contained therefore in the Equipment Controller.

The Equipment Controller itself will have a Rushmore board for interfacing to IEEE 488 links. Several pieces of equipment on the ICL line (particularly test equipment) use this type of interface. Such machines will thus be directly connected to the Equipment Controller rather than indirectly connected through the Data Line Scanner.

In the (distant) future, microcomputers containing Ethernet interfaces may be built. Such microcomputers may be directly installed with process line equipment, allowing in essence a direct connection of the process line equipment to the Ethernet. Such an arrangement would be the simplest form for interaction between Cholla and the process line machines as control information could be sent directly via the existing Etherent, bypassing the extra links required today.

Mail Server

As Cholla is based primarily on the Laurel Message System, electronic mail offers a convenient mechanism for communication of information such as run definitions and auxiliary information such as environmental measurements. The mail server will hold mailboxes for the Cholla system and several of the process researchers at ICL. This component of Cholla will be taken as is from the Laurel group at PARC.

There are several reasons for including a Mail Server in ICL. Although an IFS-type File Store includes a Mail Server within itself, the File Store performance can be adversely affected by running mail service concurrently with file service. The Juniper File Store does not include mail service within itself. The electronic mail system that we use at Xerox is based on a transport mechanism that is the subject of research at PARC. As this mechanism evolves, its newest features will be available in the standard Mail Server supported by the Laurel group. Newer features of this mechanism will not become available in IFS mail servers. To remain compatible with this system, a separate Mail Server is indicated.

As newer features of Mail Servers become available, Cholla will be able to use them to provide more automated services of its own. For example, by using messages with special *types*, Cholla will be able to include intelligent message readers that can add data to Cholla files automatically. The combination of standard Laurel with such message reading programs will provide a convenient interface for the capture of various kinds of data, including environmental factors, materials deliveries, etc.

The Mail Server is only one example of the leverage the Cholla gains from its association with the computing network at PARC. As with the File Store, another existing PARC project, the Mail Server offers substantial benefit to Cholla.

Test Input Workstation

This workstation is indicated in the schematic diagram of Figure 1 as the Process Follower in the Test Equipment Area. It is identical in function to the process follower workstations in the clean room. This workstation will be used for instructions to the testers and for input of test results.

Process Manager Workstation

This computer is essentially the same as a Process Researcher's workstation. The manager of the ICL processing line will use this workstation and will generally run special software for status monitoring, initiating runs, and for demonstrating the system to visitors. This workstation will be capable of running any Cholla user software including the process follower software that the operators use.

Process Researcher Workstations

ICL will have a large community of process researchers. These researchers will use their own personal computers, much the same as other researchers at PARC use personal computers. Process researchers may run any software that is available, but will generally be interested in specific packages for the tasks they are interested in. Included in the specialized software these machines will run are CAD software and software for the analysis of Cholla data. Although the software that is run by these researchers is not part of Cholla, the a large amount of Cholla data will be accessed through these workstations. We include these workstations in our inventory of components for completeness.

Software Components

The design of software is an iterative process. A design leads to implementation, which in turn provides information that modifies the design. Of particular importance in this process is the speed performance that is obtained by the software under construction. The performance of a software system is critical with regard to user acceptance, yet is difficult to predict on the basis of a design only. Thus, the software design presented here is at a broad and general level. The Cholla system will function generally as outlined here, but its eventual implementation will undoubtedly be shaped by experience during its construction.

When designing an ambitious system as Cholla, it is tempting to try to automate every function. We have attempted to do the opposite--every function should be capable of being performed manually. We do this for two reasons. First, it will allow a gradual introduction of the system into the production environment. Instead of needing a large, complete set of automated functions before any piece works properly, we will be able to bring the system up piece by piece. As time and resources permit, additional functions and more automation of existing functions can be included in the system. Second, the system should not be so complicated that should a part of it fail, the entire fabrication line would effectively be shut down. Manual backup procedures will be available to overcome the temporary (or even permanent) failure of any piece of Cholla.

Process and Test Data Storage

Underlying the entire software design is the design of the formats in which data is kept by the system. There are general requirements on this data that indicate proper means of storing all data. These requirements include:

1) Flexibility: Data will be accessed by different means for different purposes. Some of these purposes are known today, e.g., access by process followers. Other purposes are not known specifically today, e.g., retrieval for data analysis. This requirement suggests storing the data in as general and unencoded form as possible.

2) Robustness: Data should be stored in such a way that its integrity is preserved even in the presence of software or hardware failures. Furthermore, should the data become damaged, it should be repairable using simple tools, such as a text editor.

3) Access time: Certain applications, e.g., process following, require fast access to the data. Other applications, e.g., data analysis, may be able to cope with somewhat slower access time.

To meet these requirements, data will generally be stored in the form of Laurel mail files having a variety of structured, unique names. These files contain only text; therefore they may be read by a wide variety of existing programs. A special structure, the Laurel Table-of-Contents file (or TOC file) provides fast access to any message contained within such a file. In general, information will be stored as individual messages within Laurel mail files.

Each message will conform to a simple kind of structure. The form

<fieldname> : <fieldvalue>

will be used for storage of parameters and results within messages. A table of formats for <fieldvalue>'s specific to particular <fieldnames>'s will be kept by Cholla and will be used to aid in the extraction of data. In addition, range checking can be provided via this table along with format, so that Cholla can check input for conformance to expected norms.

Test data is generally produced by special purpose equipment in specific formats. Such data will be stored as separate files, with their own unique names. Messages in the run definition mail files that correspond to testing steps will have fields for the file name(s) of associated test data files. Through this simple technique, lengthy test output is associated with processing steps without disturbing the mail file structure of the run definitions.

The standard naming and directory functions provided by the File Store will be used to keep track of the collections of files associated with each run. By using appropriately constructed names for Cholla files, searches and access to data will be quite fast. We are consciously avoiding complicated database style storage as this tends to slow down even routine data access. With Cholla's simple formats for the storage of all data, other programs for purposes not yet understood can easily be constructed to access the data stored by Cholla.

The File Store must support remote access with multiple readers. Generally append access is all that is needed for writing, as new files will be built by appending rather than by writing in place.

Process Definition

Standard process definitions will be stored in the form of Laurel mail files. A utility program will be provided for the construction of such files from scratch or from existing mail files. Generally, a new process will be defined by slightly modifying an existing process. Such a new process will be stored as a completely separate mail file, so that each process definition can be easily accessed or compared.

Run definition will generally be accomplished via the message system. A designer will send a message containing run related information to the process line manager. Such a message will generally include the name of a process definition file, the names of masks to be used with this run, identification of the designer, and any special instructions.

The process line manager will create a run definition file as follows. First, a copy of the process definition file is made. This copy is the run definition. Next, the first message of the run definition is modified by inserting information from the designer's message. Note that this modification process is essentially the same as that performed by the line operators using the Process Following portion of Cholla. Finally, the run definition is stored on the File Store with appropriate names. This accomplishes the insertion of the run definition into the pending runs queue. Activation of the run is performed using similar techniques of message modification and file naming.

Modification of a run definition after that run has become active is also possible. The general strategy is that steps which have already been performed are not modifiable, but the portion of the run definition which has not yet been performed is modifiable. The same utility program that allows initial process definition will also allow run modification. We expect that such modifications will be performed infrequently.

Process Following

This is the part of Cholla that the operators will use as a normal part of their work. The state of the run definition files in the File Store will mirror the actual state of the wafers in the run. An operator will log on to a Process Follower workstation and identify the particular run being attended to. A Laurel-like display (see Figure 2) will show a range of steps surrounding the current step (or steps) in its table of contents area. Any step may be displayed in the displayed message area, and a current step may be displayed in the composed message area. The operator will read the instructions for the current step and perform them. Any results that are required for this step will be entered into the step through the text editor and some will be checked automatically by Cholla for consistency. Several fields will be filled in automatically, such as the time the step was started, the time the step was finished, the operator's name, etc.

An operator will be able to view the history of this run by selecting and displaying any message from the run definition. Steps already performed will have all result fields filled in. Steps not yet performed will have several blanks for insertion of result data. The operator will be able to get more specific information regarding a particular step by invoking an "Explain" command, which will display a detailed message regarding operation of a piece of equipment.

Some steps may have automatic loading of parameters to and retrieval of results from process equipment. Starting, stopping, and interrupting equipment control will be accomplished through a set of commands given through the process followers. Such commands will have the effect of

transmitting information from the process follower through the equipment controller to the controlled equipment and back again to the process follower. The actual task of controlling processing equipment is performed by the Equipment Controller, leaving the process follower free to perform other tasks while the controlled step is in progress.

This direct control of IC processing equipment is purely an optimization of manual control. This functionality is provided to increase processing line throughput and reliability. It is not an essential feature of Cholla, and any direct control step may be performed manually.

Status Reporting

Reporting the status of runs will be performed by examining the files in the file store. Special directories and names of the files contained in those directories will be used to determine which runs are in a particular status. To determine the detailed status of a particular run, that run's files will be found and examined to determine information such as current step number, etc.

The current status of particular pieces of processing equipment, e.g., furnaces, will also be available. This service will be possible for equipment that is capable of being checked by the Equipment Controller. As more processing equipment is installed that allow this function, the scope of equipment status reporting will increase.

The status reporting subsystem of Cholla will be a separate program that may be run on any workstation. Typically, this program will be run on the Process Line Manager's workstation, but a designer wanting to know the status of a particular run will also be able to run this program. The program will be designed to return the status as of the time that a status request is made. By putting the status request in a loop so that it is done at regular intervals, the illusion of continuous monitoring can be provided.

Equipment Control

The Equipment Control subsystem as outlined here is a rather ambitious part of the Cholla system. This piece involves software for which we do not have much experience, and thus may not be available in the first releases of Cholla. The overall Cholla design does accommodate this part, so that when it does become operational, it will mesh smoothly with the rest of Cholla.

The Equipment Control subsystem will be capable of talking to various pieces of process line equipment. This software will communicate with such equipment via RS 232C links (actually via the Ethernet through the Data Line Scanner machine and its RS 232C links), IEEE 488 links or directly via the Ethernet. Continuous monitoring of equipment will not be provided. Instead, the Equipment Controller will send initial parameters to equipment, and it will poll the equipment for status and/or results. (Initially, it may not even poll, but just ask for results in response to specific commands issued by an operator at a process follower workstation.)

Each separate piece of equipment that is capable of direct computer control has its own particular format for submission of parameters and return of results. Parameters for the control of equipment will be contained in run definitions in a readable text form. These parameters will have to be transformed into the particular sequences and codes expected by that piece of equipment. A separate, small device driver, or "recipe" will be developed for each piece of processing equipment to be controlled this way. These recipes will be stored on the Equipment Controller (or perhaps on the File Store) and will be referred to by name in the run definition step that calls for such direct control.

Data Analysis Interface

A significant goal of Cholla is to provide process researchers with information about runs that may be analyzed to provide insight into how to perform these processes better. The types of analyses that might be performed are limited only by one's imagination. In fact, limiting the types of analyses to some fixed set of services would be extremely shortsighted. As a result, this portion of the system is left open ended.

Rather than provide specific analysis programs, we will provide the raw data from which the analyses can proceed. The format of the mail files used by Cholla will be published and strictly adhered to. In addition, Cholla will provide a simple parser that will be capable of extracting specified fields and values from the stored run definitions. We expect that process researchers themselves will write software (or contract for software) that will perform the analyses that they consider appropriate. Since the data to be analyzed is stored in relatively simply structured files, analysis programs may be written in any language convenient to the process researcher. The parser we provide will be a separate subsystem and return its results in a file, so problems of incompatible languages will not arise. In fact, the parser will be quite simple, so that it would not be a major task to translate it into some other convenient language.

If analysis programs are to be run on a separate machine (say a VAX) then it would be worthwhile to have this parser translated so that it too could run in the same environment. Alternatively, one could perform analyses in a number of steps, running the parser on a workstation, transferring the results to the analysis machine, and then running the analysis. If such a separate machine is used, the main problem will be whether it can directly access the information on the Cholla File Store. If so, then translating the parser and running it on the analysis machine will be worthwhile. If not, the three step process described above should be used.
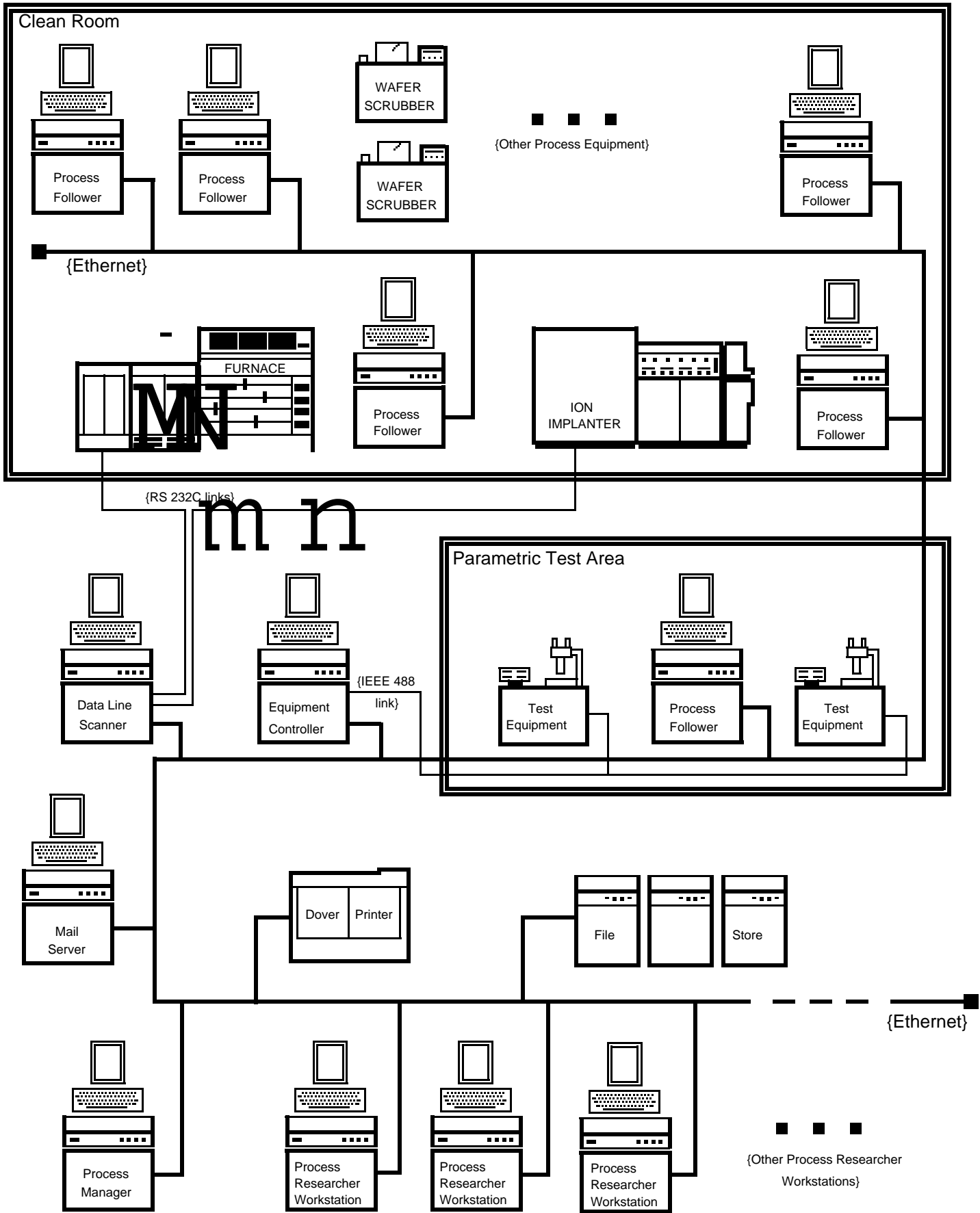
Figure 1. Cholla System Schematic

Laurel 5.1x   Thursday April 3, 1980 10:09 am PST
Mailbox empty at 10:08   366 free disk pages
User {Brotz.PA}   **New mail**   **Mail file** {RunA26Z68}   **Quit**

▸   9 Apr. 3   To: @Process57.NMos   1.8 Measure t-ox on test wafer first
                                                ox and send to C-V processing
                                                and test.
      10 Apr. 3   To: @Process57.NMos   1.9 Continue test wafer Si3N4, hold
                                                test wafer field, test wafer field ox.
      11 Apr. 3   To: @Process57.NMos   1.10 Pre-nitride clean.
  +   12 Apr. 3   To: @Process57.NMos   1.11 Si3N4 deposition. 700 Angstroms.
      13 Apr. 3   To: @Process57.NMos   2.1 Mask #1, field mask. Negative
                                                resist lithography.
      14 Apr. 3   To: @Process57.NMos   2.1b Inspection of lithographic
                                                results.
      15 Apr. 3   To: @Process57.NMos   2.2 Etch nitride in E1 plasma. (Dry □

Display   Hardcopy   Delete   Undelete   Move to {RunA26Z68}

Subject: 1.8 Measure t-ox on test wafer first ox and send to C-V processing and
 test.
To: @Process57.NMos

Test wafer first ox.

Operator: Brotz
Start time: April 3, 1980 9:54 am PST
Finish time: April 3, 1980 10:31 am PST

Record t-ox: 345

Record V-t: 0.56

End of Message.

                                                                                      □
New form   Answer   Forward   Get, Put {}   **Deliver**

Subject: 1.11 Si3N4 deposition. 700 Angstroms.
To: @Process57.NMos

Operator: ▊Name▊
Start time: ▊time▊
Finish time: ▊time▊

Test wafer Si3N4.

Record t-Nitride: ▊value▊
Record n-Nitride: ▊value▊

End of message.

Table of Contents Area

Displayed Message Area

Composed Message Area

Figure 2. Standard Laurel Display Screen