Inter-Office Memorandum

DRAFT - DRAFT - DRAFT - DRAFT

To	Cholla Interest	Date	June 8, 1981
From	Richard M. Barth Douglas K. Brotz	Location	Palo Alto
Subject	Cholla Interface	Organization	PARC/ICL-CSL

XEROX

Filed on: [Ivy]<Cholla>ChollaInterface.bravo

Cholla is the system that provides computer assistance to the silicon wafer fabrication line in the Integrated Circuit Laboratory (ICL). There are several large components in Cholla, including communication, file storage, direct operation of wafer processing equipment, and a lot tracking and recording system. This document describes the functions available through the Cholla lot tracking and recording user interface. Since this user interface is the one that Cholla users (process line operators, technicians, and the process line manager) will use almost exclusively to manipulate process line information, this interface will be referred to as the "Cholla interface".

The Cholla interface is provided by an enhanced version of Laurel 6. (It is assumed that the reader of this document has an understanding of the Laurel 6 user interface.) All user interaction with the Cholla interface is done in the Laurel style, including the manner in which one edits text and invokes commands. Cholla may be operated as if it were Laurel, providing all of its electronic mail functions. When Cholla is operated as Laurel, all the normal Laurel command buttons are available on the screen. In addition, several Cholla specific interfaces are available, through which the Cholla functions are provided.

The three window format of Laurel is retained in all of the Cholla specific interfaces. These windows will be referred to in this document; they are (from top to bottom on the screen) the table-of-contents window (or TOC window), the display window, and the composition window. As in Laurel, free form text editing is available only in the composition window.

Transfers to and from Cholla

When the Laurel commands are on the screen, the system behaves essentially as Laurel 6, but an additional screen command, **Bulletin board**, appears in the upper menu. When invoked, the **Bulletin board** command transforms the screen command menus into those of the Cholla Bulletin Board Interface, and the Cholla specific functions become available. The current mail file is cleaned up (deletions made permanent, etc.) when the **Bulletin board** command is invoked as if the **Quit** command had been invoked.

There are three separate collections of screen commands in Cholla. These are called the *Bulletin Board Interface*, the *Step List Interface*, and the *Step Interface*. In each of these screen arrangements, there are commands available to switch to one or more of the other interfaces. The specific commands available in each of these interfaces are described below. Each interface includes the standard status messages, i.e., date and time, free disk pages and in-box status, and the **User** command from the standard Laurel upper menu. Each of these interfaces also includes a **Laurel** command in the upper menu, which is used to terminate that particular Cholla interface and to return to the normal Laurel 6 interface.

The **Laurel** command appears in the place where the **Quit** command appears on the Laurel screen. It is intended that Cholla processing continue all the time, therefore, the **Quit** command is "hidden away" in the Laurel interface.

The Bulletin Board Interface is the one that is used when Cholla starts up. It may be considered the "home" interface of Cholla.

The Bulletin Board Interface

The Bulletin Board Interface provides an overview of the current activities in the fabrication line. The table-of-contents window displays a list of items that are of interest to the operators, technicians, and process line manager. These items include descriptions of the current wafer lots, equipment status, messages of the day, and short messages to specific operators. The bulletin board is not directly editable. Instead, edits to the bulletin board are made by sending stylized messages to a bulletin board process contained in each Cholla workstation.

When the Bulletin Board Interface is invoked through the **Bulletin board** command in any of the other interfaces, the windows are adjusted so that the TOC window, in which the bulletin board is listed, is as large as possible. The standard Laurel operations of scrolling, thumbing, and selecting table-of-contents entries are available.

A *title* entry is a text entry intended to label a section of the bulletin board. The first entry in the bulletin board is always a title entry, which says "Cholla Bulletin Board". Other title entries may be present further on in the bulletin board to set off separate sections of the bulletin board.

A *step list* entry contains the name of a step list (or lot) that is in progress in the fabrication area along with several indicators of that lot's current status. The step list name, e.g., MPC580 Lot 12, Bruce Furnace Maint., etc., indicates the name of the step list to which this entry refers. The next step of that step list, the machine on which that step will be performed, and the state of that step list, e.g., *in progress*, *done* or several types of *hold*, are also included in a step list entry in the bulletin board. A description of each step that has been started but not finished in this step list is included in this entry. Following these required parts of the entry, arbitrary comments may be appended by sending an appropriate message to the bulletin board process. The step list entries may be organized in the bulletin board list by operator responsible for that step list or by equipment required by current or future steps.

An *equipment status* entry contains information relevant to each piece of equipment in the fabrication area. An indication of which step of which lot is being performed, which lot has reserved that piece of equipment, which lots are within n steps of needing that equipment (where n is a small positive integer, perhaps 1), or whether the equipment is down, awaiting parts, etc. is included in the entry.

An *operator message* is a short piece of text that is intended for a specific operator in the processing area. These messages are preceded by the operator's name followed by a colon.

A *global message* is a message meant to be read by everyone who uses Cholla.

The commands that appear on the screen when the Bulletin Board Interface is active include **User**, **Laurel**, **Step list {}**, **Filter**, **New form**, **Modify**, **Get {}**, **Talk {}**, and **Deliver**. The **User** and **Laurel** commands have been described earlier. Descriptions of the other commands follow.

The **Step list {}** command, located in the upper menu, changes the interface presented from the Bulletin Board interface to the Step List Interface. If exactly one step list entry is selected prior to invoking the **Step list** command, then this step list is used to fill in the brackets automatically. This step list name may be confirmed or replaced by the name of a different step list to be displayed. The Step List Interface is described in more detail below.

The **Filter {}** command, located in the upper menu, controls the items displayed in the table-of-contents window when the Bulletin Board Interface is active. The command is followed by brackets, in which the desired filter is entered. The user may filter the bulletin board display so that only her own messages are displayed, only the equipment status entries are displayed, etc. Particular uses of the **Filter {}** command include displaying only those lots associated with a particular operator and displaying only those entries that reference a particular piece of equipment.

The lower window of the Bulletin Board Interface is similar to the lower window in Laurel. It is used for composing and delivering messages just as in Laurel. When the Bulletin Board Interface is entered from Laurel, the contents of the lower window are left unchanged to facilitate the composition of a message to the bulletin board process in reaction to a message received through the Laurel interface.

The **New form** command, located in the lower menu, is the same as the **New form** command in Laurel. It is used to initialize the contents of the lower window to a skeleton message to be edited into a message to be delivered.

The **Modify** command, located in the lower menu, is used to initiate a change in any entry in the Bulletin Board. This command may be invoked only when exactly one item in the bulletin board is selected. The lower window is initialized to a skeleton form that refers to the selected entry. Subsequent editing and delivery of the message will complete the modification of the bulletin board. If the composition window is at minimum size (as when the Bulletin Board Interface is entered), then the **Modify** command will expand that window so that the entry is made visible.

The **Get {}** command, located in the lower menu, is the same as the **Get {}** command in Laurel. It is used to initialize the lower window to various stylized forms for requesting changes to the bulletin board.

The **Talk {}** command, located in the lower menu, allows access to a low-level equipment driver. The brackets in the Talk {} command are filled in with the name of the piece of equipment that is to be controlled. When invoked, this command will transform the lower menu into one that contains commands and parameter brackets suitable for low-level control of the specific piece of equipment. The lower window records the interactions with a typescript. Each low-level talker menu includes a **Terminate** command that terminates the equipment talker and returns the lower menu to its normal Bulletin Board Interface appearance.

The **Deliver** command, located in the lower menu, is the same as the **Deliver** command in Laurel. Bulletin board changes are always sent through this command.

The Step List Interface

The Step List Interface provides access to the step list for a particular lot or maintenance sequence. The table-of-contents window displays the list of steps in order as specified by the particular step list. This list is also displayed in the Step Interface described below. The essential difference between the Step List and Step Interfaces is that the Step List Interface is designed for modifying entire step lists, whereas the Step Interface is designed for operating on individual steps. The commands available, especially in the lower menu, differ accordingly.

When the Step List Interface is invoked through the **Step list** command in any of the other interfaces, the windows are adjusted so that the TOC window, in which the step list is displayed, is as large as possible. The standard Laurel operations of scrolling, thumbing, and selecting table-of-contents entries are available.

Each entry in the step list contains the following items in the following order:

<state> <step name> <parameters> <data record name> -- <comments>

The <state> is a single character from the set ~ (not started), s (started), p (in progress), d (process done), and * (finished). The meanings of these states will be described further on under the Step Interface.

The <step name> is the name of this step, optionally including a version number. When the <state> is ~ (not started), the name part of the <step name> refers to a specification mail file, from which the description of the processing to be done during the step will be taken. If the version number is present, then the particular specification in that file corresponding to this version number is used. If no version number is present, then the specification in that file that is specified as the generic specification at the time that the step is started will be used. Once a step has been started (see the Step Interface), the step name refers to the entry in this lot's *Done* file.

The <parameters> part depends on the requirements of the specification referred to by the <step name>. Specifications may leave some parameters unspecified. These parameters are named in the specification, and must appear in the <parameters> section of the step list entry if any exist. The <parameters> section is surrounded by parentheses, and contains a list of name: value pairs separated by commas. For example, if a specification uses parameters Dose and Time, then a valid <parameters> section would be (Dose: 150, Time: 75). The treatment of units is to be determined.

The <data record name>, which may include a version number, is the name of the data record used to record the results of this step. The name refers to a mail file in which data record templates are held. If a version number is present, then the particular data record in that file corresponding to this version number is used. If no version number is present, then the data record template in that file that is specified as the generic data record at the time that the step is started will be used. When a step is started , the data record is copied to this lot's *Done* file and that copy is used from that point on.

The <comments> section follows the double hyphen. This part of the step list entry may contain arbitrary text, and may be edited by the operator. For example, the <comments> may contain special notes to the operator or notes from the operator regarding unusual circumstances connected with that step.

The commands that appear on the screen when the Step List Interface is active include **User**, **Laurel**, **Bulletin board**, **Step list {}**, **Prototypes**, **Step**, **Display list {}**, **Edit**, and **Update**. The **User** and **Laurel** commands have been described earlier. Descriptions of the other commands follow.

The **Bulletin board** command, located in the upper menu, changes the interface presented from the Step List Interface to the Bulletin Board Interface. When the Bulletin Board Interface appears, the step list entry corresponding to the step list displayed in the **Step list {}** command is selected.

The **Step list {}** command contains in its brackets the name of the current step list. Any step list may be displayed by invoking this command and typing in the name of that step list. When the **Step list {}** command is invoked, the **Step** command replaces the **Display list {}** command in the middle menu.

The **Prototypes** command displays the table-of-contents of the Prototypes.mail file. This mail file contains the templates of processing sequences, and is useful as source text for editing another step list. When the Prototypes.mail table-of-contents is displayed, the **Display list {}** command replaces the **Step** command in the middle menu.

The **Step** command, located in the middle menu, changes the Interface presented from the Step List Interface to the Step Interface (see below). (This command is not available when the Prototypes.mail table-of-contents is displayed.) From either the Step List or Step Interfaces, this command behaves like a **Display** command in Laurel, with the exception that both the display and composition windows are filled in. The display window is filled in with the contents of the specification referred to by the selected step list entry, and the composition window is filled in with the contents of the data record referred to by the selected step list entry. No modifications to the stored data or the text displayed are made by the **Step** command, see the **Start** command of the Step Interface for more details.

The **Display list {}** command, located in the middle menu, is used to display an arbitrary step list in the display window. This command is available only when the Prototypes.mail table-of-contents is displayed. **Display list {}** is useful for obtaining boilerplate step lists for use as source selections when editing the current step list. The brackets are filled in automatically to correspond to the selected step list in the Prototypes.mail file when the **Display list {}** command is invoked.

The **Edit** command, located in the lower menu, is used to initiate a change to the current step list. The current step list is copied into the composition window, replacing any prior contents of that window. Once in the composition region, the step list may be edited (the finished entries are protected). A subsequent **Update** command will complete the modification. If the composition window is at minimum size, then the **Edit** command will expand that window so that the step list is made visible.

The **Update** command, located in the lower menu, is used to replace the contents of the current step list with the contents of the composition window. If the composition window contains any text, that text is checked for proper syntax, and if the syntax is acceptable, the text replaces the current step list. As the **Update** command is potentially dangerous, a confirmation is required, and only certain users with permission may use this command. When an update to a step list is performed, the bulletin board entries that reference that step list are also updated.

The Step Interface

The Step Interface provides access to the individual steps of the current step list. The table-of-contents window displays the list of steps in order as specified by the particular step list just as in the Step List Interface.

When the Step Interface is invoked through the **Step** command in the Step List Interface, the windows are adjusted so that the TOC window is only a few lines high with the selected step entry visible, and the remaining space is divided equally between the display and composition windows. The standard Laurel operations of scrolling, thumbing, and selecting table-of-contents entries are available.

Editing a step's data record is allowed only if the state of that step is s (started), or d (process done). Attempting to edit a data record for a step in any other state results in flashed exception messages. A step that is marked ~ (not started) requires that it be started before editing may take place. A step marked p (in progress) requires that automatic processing be terminated (either normally or explicitly cancelled) before editing is allowed. Finally, attempting to edit a step that is marked * (finished) results in a warning and a request for confirmation. If confirmed, editing is allowed, but the state of the step is automatically reset to d (process done).

The *next step* is the first step in the step list that has not yet been started. If all steps prior to the next step are in finished or rejected state, then the next step is automatically selected when the Step Interface is entered. If some step prior to the next step is not finished or rejected, then the first such step is automatically selected.

The commands that appear on the screen when the Step Interface is active include **User**, **Laurel**, **Bulletin board**, **Step list {}, Step**, **Start**, **Process**, **Cancel**, **Process done**, **Reject** and **Finish**. The **User** and **Laurel** commands have been described earlier. The **Laurel** command is augmented so that if the composition window contains a data record that has been edited but not yet filed, then its contents are reflected back to the *Done* file before returning to Laurel. Descriptions of the other commands follow.

The **Bulletin board** command, located in the upper menu, changes the interface presented from the Step Interface to the Bulletin Board Interface. When the Bulletin Board Interface appears, the step list entry corresponding to the step list displayed in the **Step list {}** command is selected. If the composition window contains a data record that has been edited but not yet filed, then its contents are reflected back to the *Done* file.

The **Step list {}** command, located in the upper menu, is used to show which step list is currently displayed. Invoking this command from the Step Interface changes the Interface presented to the Step List Interface, using the current step list. If the composition window contains a data record that has been edited but not yet filed, then its contents are reflected back to the *Done* file.

The **Step** command, located in the middle menu, is the same as the **Step** command in the Step List Interface. If the state of the selected step is ~ (not started), the specification and data record are taken from the specification file and data record file as described under the Step List Interface. If the state of the selected step is not ~, then the *Done* file is consulted for the correct data record and specification. If the composition window contains a data record that has been edited but not yet filed, then its contents are reflected back to the *Done* file.

The **Start** command, located in the lower menu, initiates the recording of data for a step. Only the *next step* may be started. This command starts the step that is currently displayed in the display window. The specification and data record files are checked again to be sure that the correct specification and data record are being used. If the specification is not the same version as the one that this operator has last seen, then a warning message is flashed to alert the operator to this fact (see operator characterization files, below). The data record is partially filled in to show the exact specification that is being used and the specific values for any parameters, and this data record is copied to the *Done* file for the current step list. The state portion of the step list entry for this step is changed to s (started), and the bulletin board is updated. From now on, information regarding this step will be taken from the *Done* file. If any step other than the *next step* is selected at the time the **Start** command is invoked, then an exception message is flashed. If the composition window contained a data record that has been edited but not yet filed, then its contents are reflected back to the *Done* file.

The **Process** command, located in the lower menu, appears only when the current step displayed in the display and composition regions is capable of being processed through Cholla automatically. Thus, it does not appear if the state of the current step is not started, process done, rejected, or finished. Invoking the **Process** command initiates automatic processing. The state of the step in the step list entry is changed to p (in progress). When the automatic processing is finished, the state of the step is changed to d (process done).

The **Cancel** command, located in the lower menu, appears only when the current step is in state p (in progress), i.e., it is being processed through Cholla automatically. Invoking **Cancel** terminates automatic processing and puts the step in s (started) state. Once automatic processing has completed normally, the **Cancel** command disappears from the Cholla screen.

The **Process done** command, located in the lower menu, appears only when the state of the current step displayed in the display and composition regions is s (started). Invoking **Process done** makes the state of the step d (process done). Thus, this command provides the manual analog to the **Process** command, so that the operator can manually indicate when processing has completed but all testing and completion of the data record has not been finished.

The **Finish** command, located in the lower menu, is used to indicate that a step is completely done. All processing must be complete, and the data record must be completely filled out. The data record is checked to make sure that this is the case. If so, the contents of the composition window are reflected back into the *Done* file for this step list, and the state of the step in its step list entry is changed to * (finished). If not, an exception is flashed, a confirmation is required, and if confirmed, the step is marked * (finished), but a message is sent to the process line manager to inform him of the exceptional condition.

The **Reject** command, located in the lower menu, is used to indicate that a step has been rejected and that suitable recovery actions must be taken. If the specification for a step contains a field that describes additional steps that must be taken for standard recovery, then the **Reject** command makes the appropriate changes to the step list and bulletin board. If no such standard recovery action is indicated, then the **Reject** command changes the step list status to "Hold" and updates the bulletin board. Any step list placed in "Hold" status must be taken out of "Hold" status by means of appropriate edits to the step list and messages to the bulletin board process.

Specification files

The mail files that contain specifications for process steps are called *specification files*. Each specification file is named to correspond to the type of process step it describes, e.g., LPCVD.mail or DryEtch.mail.

The first message in each specification file must contain two header fields, labelled NextVersion: and GenericVersion:. The value of each of these fields is a version number. Each specification message contained in a specification file contains a Version: field that uniquely identifies it within the file. The NextVersion: field of the fiest message specifies the number for the next specification to be entered in that file. The GenericVersion: field refers to the specification with that version number. That specification is the one that is used by step lists that refer to a specification without explicitly mentioning a version number.

New specifications are added to a specification file by sending a message to the Cholla specification file process. That process will add the correct new version number and update the first message appropriately. Similarly, a new generic version number for a specification file is set by sending an appropriate message to that same process. It is the intent of the Cholla system that no editing of old specifications ever be done, instead new specifications are merely appended to the specification files.

Prototype file

A special mail file, named Prototypes.mail, holds the generic step lists for the standard processes used in the processing facility. Processes are identified by text in the Subject: field of the message, e.g.,

> Subject: NSIL II process.

A Laurel runnable program will be able to take a generic process description as appears in the Prototypes.mail file, to ask several questions regarding parameters, and to create a new specific step list for a new run. The results of this program appear in the lower window in a message that is ready to be delivered to the Cholla workstations. Final edits may be made manually before delivering the message to start the new lot.

Operator files

The particular specifications that an operator has read are recorded in a special file called an *operator file*. An operator file has the structure of a Laurel message, with fields named for the specification files. The value of each such field is the version number of that specification that the operator has seen most recently. Any time the **Start** command is invoked, the operator file corresponding to the current operator is checked. If the version of the specification that is being used is the same as the version for that specification that is listed in the operator file, then no special action occurs. If the versions are not the same, then a warning message is flashed and a confirmation is required. When the confirmation is given, the operator file is updated to include the version of the specification just read.