SEMI E30-1000
GENERIC MODEL FOR COMMUNICATIONS AND CONTROL OF MANUFACTURING EQUIPMENT (GEM)

This standard was technically approved by the Global Information & Control Committee and is the direct responsibility of the North American Information & Control Committee. Current edition approved by the North American Regional Standards Committee on July 14 and August 28, 2000. Initially available at www.semi.org August 2000; to be published October 2000. Originally published in 1992; previously published June 2000.

CONTENTS

SEMI E30-1000
GENERIC MODEL FOR COMMUNICATIONS AND CONTROL OF MANUFACTURING EQUIPMENT (GEM)

This standard was technically approved by the Global Information & Control Committee and is the direct responsibility of the North American Information & Control Committee. Current edition approved by the North American Regional Standards Committee on July 14 and August 28, 2000. Initially available at www.semi.org August 2000; to be published October 2000. Originally published in 1992; previously published June 2000.

1 Introduction

1.1 *Revision History* — This is the first release of the GEM standard.

1.2 *Scope* — The scope of the GEM standard is limited to defining the behavior of semiconductor equipment as viewed through a communications link. The SEMI E5 (SECS-II) standard provides the definition of messages and related data items exchanged between host and equipment. The GEM standard defines which SECS-II messages should be used, in what situations, and what the resulting activity should be. Figure 1.1 illustrates the relationship of GEM, SECS-II and other communications alternatives.

The GEM standard does NOT attempt to define the behavior of the host computer in the communications link. The host computer may initiate any GEM message scenario at any time and the equipment shall respond as described in the GEM standard. When a GEM message scenario is initiated by either the host or equipment, the equipment shall behave in the manner described in the GEM standard when the host uses the appropriate GEM messages.

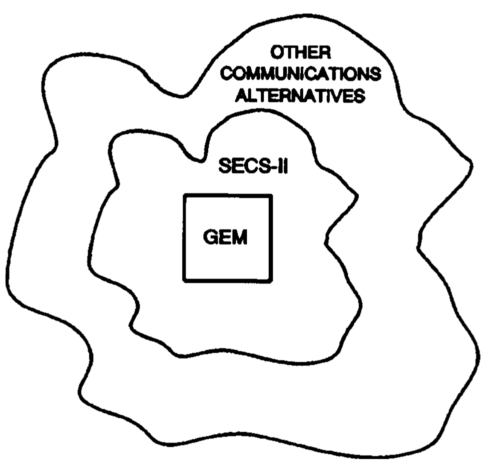


Figure 1.1
GEM Scope

The capabilities described in this standard are specifically designed to be independent of lower-level communications protocols and connection schemes (e.g., SECS-I, SMS, point-to-point, connection-oriented or connectionless). Use of those types of standards is not required or precluded by this standard.

This standard does not purport to address safety issues, if any, associated with its use. It is the responsibility of the users of this standard to establish appropriate safety and health practices and determine the applicability of regulatory limitations prior to use.

1.3 *Intent* — GEM defines a standard implementation of SECS-II for all semiconductor manufacturing equipment. The GEM standard defines a common set of equipment behavior and communications capabilities that provide the functionality and flexibility to support the manufacturing automation programs of semiconductor device manufacturers. Equipment suppliers may provide additional SECS-II functionality not included in GEM as long as the additional functionality does not conflict with any of the behavior or capabilities defined in GEM. Such additions may include SECS-II messages, collection events, alarms, remote command codes, processing states, variable data items (data values, status values or equipment constants), or other functionality that is unique to a class (etchers, steppers, etc.) or specific instance of equipment.

GEM is intended to produce economic benefits for both device manufacturers and equipment suppliers. Equipment suppliers benefit from the ability to develop and market a single SECS-II interface that satisfies most customers. Device manufacturers benefit from the increased functionality and standardization of the SECS-II interface across all manufacturing equipment. This standardization reduces the cost of software development for both equipment suppliers and device manufacturers. By reducing costs and increasing functionality, device manufacturers can automate semiconductor factories more quickly and effectively. The flexibility provided by the GEM standard also enables device manufacturers to implement unique automation solutions within a common industry framework.

The GEM standard is intended to specify the following:

— A model of the behavior to be exhibited by semiconductor manufacturing equipment in a SECS-II communication environment,

— A description of information and control functions needed in a semiconductor manufacturing environment,

— A definition of the basic SECS-II communications capabilities of semiconductor manufacturing equipment,

— A single consistent means of accomplishing an action when SECS-II provides multiple possible methods, and

— Standard message dialogues necessary to achieve useful communications capabilities.

The GEM standard contains two types of requirements:

— fundamental GEM requirements and

— requirements of additional GEM capabilities.

The fundamental GEM requirements form the foundation of the GEM standard. The additional GEM capabilities provide functionality required for some types of factory automation or functionality applicable to specific types of equipment. A detailed list of the fundamental GEM requirements and additional GEM capabilities can be found in Chapter 8, GEM Compliance. Figure 1.2 illustrates the components of the GEM standard.

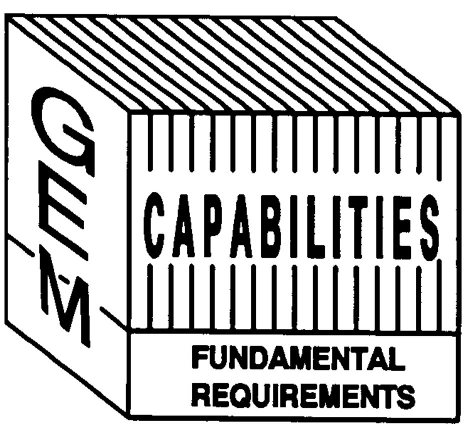


Figure 1.2
GEM Components

Equipment suppliers should work with their customers to determine which additional GEM capabilities should be implemented for a specific type of equipment. Because the capabilities defined in the GEM standard were specifically developed to meet the factory automation requirements of semiconductor manufacturers, it is anticipated that most device manufacturers will require most of the GEM capabilities that apply to a particular type of equipment. Some device manufacturers may not require all the GEM capabilities due to differences in their factory automation strategies.

1.4 *Overview* — The GEM standard is divided into sections as described below.

Section 1 — Introduction

This section provides the revision history, scope and intent of the GEM standard. It also provides an overview of the structure of the document and a list of related documents.

Section 2 — Definitions

This section provides definitions of terms used throughout the document.

Section 3 — State Models

This section describes the conventions used throughout this document to depict state models. It also describes the basic state models that apply to all semiconductor manufacturing equipment and that pertain to more than a single capability. State models describe the behavior of the equipment from a host perspective.

Section 4 — Capabilities and Scenarios

This section provides a detailed description of the communications capabilities defined for semiconductor manufacturing equipment. The description of each capability includes the purpose, definitions, requirements, and scenarios that shall be supported.

Section 5 — Data Definitions

This section provides a reference to the Data Item Dictionary and Variable Item Dictionary found in SEMI Standard E5. The first subsection shows those data items from SECS-II which have been restricted in their use (i.e., allowed formats). The second subsection lists variable data items that are available to the host for data collection and shows any restrictions on their SECS-II definitions.

Section 6 — Collection Events

This section provides a list of required collection events and their associated data.

Section 7 — SECS Message Subset

This section provides a composite list of the SECS-II messages required to implement all capabilities defined in the GEM standard.

Section 8 — GEM Compliance

This section describes the fundamental GEM requirements and additional GEM capabilities and provides references to other sections of the standard where detailed requirements are located. This section also defines standard terminology and documentation that can be used by equipment suppliers and device manufacturers to describe compliance with this standard.

Section A — Application Notes

These sections provide additional explanatory information and examples.

Section A.1 — Factory Operational Script

This section provides an overview of how the required SECS capabilities may be used in the context of a typical factory operation sequence. This section is organized according to the sequence in which actions are typically performed.

Section A.2 — Equipment Front Panel

This section provides guidance in implementing the required front panel buttons, indicators, and switches as defined in this document. A summary of the front panel requirements is provided.

Section A.3 — Examples of Equipment Alarms

This section provides examples of alarms related to various equipment configurations.

Section A.4 — Trace Data Collection Example

This section provides an example of trace initialization by the host and the periodic trace data messages that might be sent by the equipment.

Section A.5 — Harel Notation

This section explains David Harel's "Statechart" notation that is used throughout this document to depict state models.

Section A.6 — Example Control Model Application

This section provides one example of a host's interaction with an equipment's control model.

Section A.7 — Examples of Limits Monitoring

This section contains four limits monitoring examples to help clarify the use of limits and to illustrate typical applications.

1.5 *Applicable Documents*

1.5.1 *SEMI Standards* — The following SEMI standards are related to the GEM standard. The specific portions of these standards referenced by GEM constitute provisions of the GEM standard.

SEMI E4 — SEMI Equipment Communications Standard 1 — Message Transfer (SECS-I)

SEMI E5 — SEMI Equipment Communications Standard 2 — Message Content (SECS-II)

SEMI E13 — Standard for SEMI Equipment Communication Standard Message Service (SMS)

SEMI E23 — Specification for Cassette Transfer Parallel I/O Interface

1.5.2 *Other References*

Harel, D., "Statecharts: A Visual Formalism for Complex Systems," Science of Computer Programming 8 (1987) 231-274[1].

NOTE 1: As listed or revised, all documents cited shall be the latest publications of adopted standards.

2 Definitions

2.1 *alarm* — An alarm is related to any abnormal situation on the equipment that may endanger people, equipment, or material being processed. Such abnormal situations are defined by the equipment manufacturer based on physical safety limitations. Equipment activities potentially impacted by the presence of an alarm shall be inhibited.

2.1.1 Note that exceeding control limits associated with process tolerance does not constitute an alarm nor do normal equipment events such as the start or completion of processing.

2.2 *capabilities* — Capabilities are operations performed by semiconductor manufacturing equipment. These operations are initiated through the communications interface using sequences of SECS-II messages (or scenarios). An example of a capability is the setting and clearing of alarms.

2.3 *collection event* — A collection event is an event (or grouping of related events) on the equipment that is considered to be significant to the host.

2.4 *communication failure* — A communication failure is said to occur when an established communications link is broken. Such failures are protocol specific. Refer

1 Elsevier Science, P.O. Box 945, New York, NY 10159-0945, http://www.elvesier.nl/homepage/browse.htt

to the appropriate protocol standard (e.g., SEMI E4 or SEMI E37) for a protocol-specific definition of communication failure.

2.5 *communication fault* — A communication fault occurs when the equipment does not receive an expected message, or when either a transaction timer or a conversation timer expires.

2.6 *control* — To control is to exercise directing influence.

2.7 *equipment model* — An equipment model is a definition based on capabilities, scenarios, and SECS-II messages that manufacturing equipment should perform to support an automated manufacturing environment. (See also Generic Equipment Model.)

2.8 *event* — An event is a detectable occurrence significant to the equipment.

2.9 *GEM compliance* — The term "GEM Compliance" is defined with respect to individual GEM capabilities to indicate adherence to the GEM standard for a specific capability. Section 8 includes more detail on GEM Compliance.

2.10 *Generic Equipment Model* — The Generic Equipment Model is used as a reference model for any type of equipment. It contains functionality that can apply to most equipment, but does not address unique requirements of specific equipment.

2.11 *host* — The SEMI E4 and E5 standards define Host as "the intelligent system that communicates with the equipment."

2.12 *message fault* — A message fault occurs when the equipment receives a message that it cannot process because of a defect in the message.

2.13 *operational script* — An operational script is a collection of scenarios arranged in a sequence typical of actual factory operations. Example sequences are system initialization powerup, machine setup, and processing.

2.14 *operator* — A human who operates the equipment to perform its intended function (e.g., processing). The operator typically interacts with the equipment via the equipment supplied operator console.

2.15 *process unit* — A process unit refers to the material that is typically processed as a unit via single run command, process program, etc. Common process units are wafers, cassettes, magazines, and boats.

2.16 *processing cycle* — A processing cycle is a sequence wherein all of the material contained in a typical process unit is processed. This is often used as a measure of action or time.

2.17 *scenario* — A scenario is a group of SECS-II messages arranged in a sequence to perform a capability. Other information may also be included in a scenario for clarity.

2.18 *SECS-I* — SEMI Equipment Communications Standard 1 (SEMI E4). This standard specifies a method for a message transfer protocol with electrical signal levels based upon EIA RS232-C.

2.19 *SECS-II* — SEMI Equipment Communications Standard 2 (SEMI E5). This standard specifies a group of messages and the respective syntax and semantics for those messages relating to semiconductor manufacturing equipment control.

2.20 *SMS* — SECS Message Service. An alternative to SECS-I to be used when sending SECS-II formatted messages over a network.

2.21 *state model* — A State Model is a collection of states and state transitions that combine to describe the behavior of a system. This model includes definition of the conditions that delineate a state, the actions/reactions possible within a state, the events that trigger transitions to other states, and the process of transitioning between states.

2.22 *system default* — Refers to state(s) in the equipment behavioral model that are expected to be active at the end of system initialization. It also refers to the value(s) that specified equipment variables are expected to contain at the end of system initialization.

2.23 *system initialization* — The process that an equipment performs at power-up, system activation, and/or system reset. This process is expected to prepare the equipment to operate properly and according to the equipment behavioral models.

2.24 *user* — A human or humans who represent the factory and enforce the factory operation model. A user is considered to be responsible for many setup and configuration activities that cause the equipment to best conform to factory operations practices.

3 State Models

The following sections contain state models for semiconductor manufacturing equipment. These state models describe the behavior of the equipment from a host perspective in a compact and easy to understand format. State models for different equipment will be identical in some areas (e.g., communications), but may vary in other areas (e.g., processing). It is desirable to divide the equipment into parallel components that can be modeled separately and then combined. An example of a component overview of an equipment is provided as Figure 3.0.

Equipment manufacturers must document the operational behavior of their equipment using state model methodology. State models are discussed in Sections 3.1 and A.5 and in a referenced article. Documentation of a state model shall include the following three elements:

— A *state diagram* showing the possible states of the system or components of a system and all of the possible transitions from one state to another. The states and transitions must each be labeled. Use of the Harel notation (see A.5) is recommended.

— A *transition table* listing each transition, the beginning and end states, what stimulus triggers the transition, and any actions taken as a result of the transition.

— A *definition of each state* specifying system behavior when that state is active.

Examples of the above elements are provided in Section A.5.

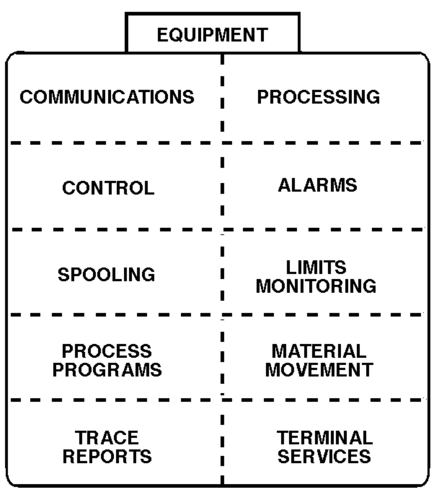


Figure 3.0
Example Equipment Component Overview

The benefits of providing state models are:

1. State machine models are a useful specification tool,
2. A host system can anticipate machine behavior based upon the state model,
3. End-users and equipment programmers have a common description of machine behavior from which to work,
4. "Legal" operations can be defined pertaining to any machine state,
5. External event notifications can be related to internal state transitions,
6. External commands can be related to state transitions,
7. State model components describing different aspects of machine control can be related to one another (example: processing state model with material transport state model; processing state model with internal machine safety systems).

3.1 *State Model Methodology* — To document the expected functionality of the various capabilities described in this document, the "Statechart" notation developed by David Harel has been adopted. An article by Harel is listed in Section 1.5 and should be considered "must" reading for a full understanding of the notation. The convention used in this and following sections is to describe the dynamic functionality of a capability with three items: a textual description of each state or substate defined, a table that describes the possible transitions from one state to another, and a graphical figure that uses the symbols defined by Harel to illustrate the relationships of the states and transitions. The combination of these items define the state model for a system or component. A summary of the Harel notation and a more detailed description of the text, table, and figure used to define behavior with this methodology is contained in the Application Note A.5.

The basic unit of a state model is the state. A state is a static set of conditions. If the conditions are met, the state is current. These conditions might involve sensor readings, switch positions, time of day, etc. Also part of a state definition is a description of reactions to specific stimuli (e.g., if message Sx,Fy is received, generate reply message Sx,Fy + 1). Stimuli may be quite varied but for semiconductor equipment would include received SECS messages, expired timers, operator input at an equipment terminal, and changes in sensor readings.

To help clarify the interpretation of this document and the state models described herein, it is useful to distinguish between a state and an event and the relationship of one to the other. An event is dynamic rather than static. It represents a change in conditions, or more specifically, the awareness of such a change. An event might involve a sensor reading exceeding a limit, a switch changing position, or a time limit exceeded.

A change to a new active state (state transition) must always be prompted by a change in conditions, and thus an event. In addition, a state transition may itself be termed an event. In fact, there are many events that may occur on an equipment, so it is important to classify events based on whether they can be detected and whether they are of interest. In this document, the term event has been more narrowly defined as a detectable occurrence that is significant to the equipment.

A further narrowing of the definition of event is represented by the term "collection event," which is an event (or group of related events) on the equipment that is considered significant to the host. It is these events that (if enabled) are reported to the host. By this definition, the list of collection events for an equipment would typically be only a subset of total events. The state models in this document are intended to be limited to the level of detail in which the host is interested. Thus, all state transitions defined in this standard, unless otherwise specified, shall correspond to collection events.

3.2 *Communications State Model* — The Communications State Model defines the behavior of the equipment in relation to the existence or absence of a communications link with the host. Section 4.1 expands on this section by defining the Establish Communications capability. This model pertains to a <u>logical</u> connection between equipment and host rather than a <u>physical</u> connection.

3.2.1 *Terminology* — The terms <u>communication failure</u>, <u>connection transaction failure</u>, and <u>communication link</u> are defined for use within this document only and should not be confused with the same or similar terms used elsewhere.

- See SEMI E4 (SECS-I) or SEMI E37 (HSMS) for a protocol specific definitions of communications failure.
- A <u>connection transaction failure</u> occurs when attempting to establish communications and is caused by
  - — a communication failure,
  - — the failure to receive an S1,F14 reply within a reply timeout limit, or
  - — receipt of S1,F14 that has been improperly formatted or with COMMACK[2] not set to 0.
- A reply timeout period begins after the successful transmission of a complete primary message for which a reply is expected. (See SEMI E4 (SECS-I) or SEMI E37 (HSMS) for a protocol-specific definition of reply timeout.)
- A <u>communication link</u> is established following the first successful completion of any one S1,F13/F14 transaction with an acknowledgement of "accept". The establishment of this link is logical rather than physical.
- Implementations may have mechanisms which allow outgoing messages to be stored temporarily prior to being sent. The noun <u>queue</u> is used to cover such stored messages. They are <u>queued</u> when placed within the queue and are <u>dequeued</u> by removing them from this storage.
- <u>Send</u> includes "queue to send" or "begin the process of attempting to send" a message. It does not imply the successful completion of sending a message.
- The host may attempt to establish communications with equipment at any time due to the initialization of the host or by independent detection of a communications failure by the host. Thus, the host may initiate an S1,F13/F14 transaction at any time.

3.2.2 *CommDelay Timer* — The <u>CommDelay timer</u> represents an internal timer used to measure the interval between attempts to send S1,F13. The length of this interval is equal to the value in the EstablishCommunicationsTimeout. The CommDelay timer is not directly visible to the host.

<u>EstablishCommunicationsTimeout</u> is the user-configurable equipment constant that defines the delay, in seconds, between attempts to send S1,F13. This value is used to initialize the CommDelay timer.

The CommDelay timer is <u>initialized</u> to begin timing. The CommDelay timer is initialized only when the state WAIT DELAY is entered.

The CommDelay timer is <u>expired</u> when it "times out," and the time remaining in the interval between attempts to send is zero. When the timer expires during the state WAIT DELAY, it triggers a new attempt to send S1,F13 and the transition to the state WAIT CRA[3] .

3.2.3 *Conventions*

- The attempt to send S1,F13 is made only upon transit into the state WAIT CRA. The CommDelay Timer should be set to "expired" at this time.

2 Establish Communications Acknowledge Code, defined in Section 4.1. See the SEMI E5 Standard for further definition of this Data Item.

3 CRA is the mnemonic defined for Establish Communications Request Acknowledge (S1,F14).

- The CommDelay timer is initialized only upon transit into the state WAIT DELAY. A next attempt to send S1,F13 shall occur only upon a transit to the state WAIT CRA.

3.2.4 *Communication States* — There are two major states of SECS communication, DISABLED and ENABLED. The system default state must be user-configurable at the equipment (e.g., via a jumper setting or non-volatile memory variable).

Once system initialization has been achieved, the operator shall be able to change the communication state selection at any time via equipment terminal functions or momentary switch. A two-position type switch must not be used due to possible conflict with the system default.

The ENABLED state has two substates, NOT COMMUNICATING and COMMUNICATING, described below. The equipment must inform the operator of the current communication state via continuous display at the equipment, including the NOT COMMUNICATING and COMMUNICATING sub-states.

In the event of a connection transaction failure, a user-configurable equipment constant EstablishCommunicationsTimeout is used to establish the interval between attempts to send an S1,F13 (Establish Communications Request) while in the NOT COMMUNICATING substate.

Figure 3.2.1 shows the relationship between the superstates and substates of the Communications State Model. A description of the events triggering state transitions and the actions taken is given in Table 3.2.

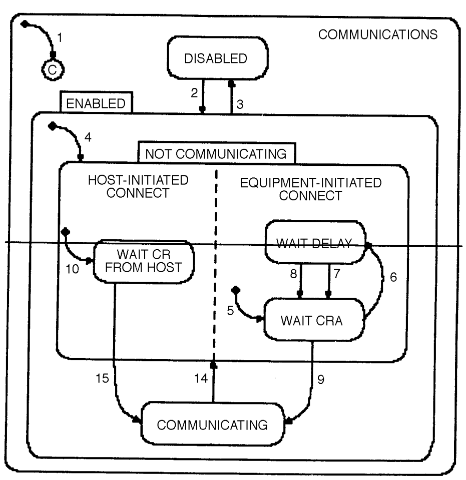


Figure 3.2.1
Communications State Diagram

The states of the Communications State Model are defined as follows:

DISABLED

In this state SECS-II communication with a host computer is non-existent. If the operator switches from ENABLED to DISABLED, all SECS-II communications must cease immediately. Any messages queued to send shall be discarded, and all further action on any open transactions and conversations shall be terminated.[4] Handling of messages currently being transmitted is an issue for lower level message transfer protocols and is not addressed in this standard.

The DISABLED state is a possible system default.

ENABLED

ENABLED has two substates, COMMUNICATING and NOT COMMUNICATING. Whenever communications are enabled, either during system initialization or through operator selection, the substate of NOT COMMUNICATING is active until communications are formally established. Lower-level protocols (such as SECS-I) are assumed to be functioning normally in that they are capable of supporting the communication of SECS-II syntax.

The ENABLED state is a possible system default.

ENABLED/NOT COMMUNICATING

No messages other than S1,F13, S1,F14, and S9,Fx shall be sent while this substate is active. The equipment shall discard any messages received from the host other than S1,F13 or S1,F14 (Establish Communications Acknowledge). It shall also periodically attempt to establish communication with a host computer by issuing an S1,F13 until communications are successfully established. However, only one equipment-initiated S1,F13 transaction may be open at any time.

The NOT COMMUNICATING state has two AND substates, HOST-INITIATED CONNECT and EQUIPMENT-INITIATED CONNECT, both of which are active whenever the equipment is NOT COMMUNICATING. These two substates clarify the behavior of the equipment in the event that both

4 Refer to SEMI E5, Section 5, for definitions of SECS-II transaction and conversation protocols.

the equipment and the host attempt to establish communications during the same period of time[5] .

NOT COMMUNICATING/EQUIPMENT-INITIATED CONNECT

This state has two substates, WAIT CRA and WAIT DELAY. Upon any entry to the NOT COMMUNICATING state, whenever EQUIPMENT-INITIATED CONNECT first becomes active, a transition to WAIT CRA occurs, the CommDelay timer is set to “expired,” and an immediate attempt to send S1,F13 is made.

NOT COMMUNICATING/EQUIPMENT-INITIATED CONNECT/WAIT CRA

An Establish Communications Request has been sent. The equipment waits for the host to acknowledge the request.

NOT COMMUNICATING/EQUIPMENT-INITIATED CONNECT/WAIT DELAY

A connection transaction failure has occurred. The CommDelay timer has been initialized. The equipment waits for the timer to expire.

NOT COMMUNICATING/HOST-INITIATED CONNECT

This state describes the behavior of the equipment in response to a host-initiated S1,F13 while NOT COMMUNICATING is active.

NOT COMMUNICATING/HOST-INITIATED CONNECT/WAIT CR FROM HOST

The equipment waits for an S1,F13 from the host. If an S1,F13 is received, the equipment attempts to send an S1,F14 with COMMACK = 0.

ENABLED/COMMUNICATING

Communications have been established. The equipment may receive any message from the host, including S1,F13. When the equipment is COMMUNICATING, SECS communications with a host computer must be maintained. This state remains active until communications are disabled or a communication failure occurs. If the equipment receives S1,F13 from the host while in the COMMUNICATING substate, it should respond with S1,F14 with COMMACK set to zero. If the equipment receives S1,F14 from a previously sent S1,F13, no action is required.

5 Note that in the Harel notation, an exit from any AND substate is an exit from the parent state and thus from all other AND substates of that parent substate.

In the event of communication failure, the equipment shall return to the NOT COMMUNICATING substate and attempt to re-establish communications with the host.

It is possible that the equipment may be waiting for an S1,F14 from the host in EQUIPMENT-INITIATED CONNECT/WAIT CRA at the time an S1,F13 is received from the host in HOST-INITIATED CONNECT/WAIT CR FROM HOST. When this situation occurs, both equipment and host have an open S1,F13/S1,F14 transaction. Since communications are successfully established on the successful completion of any S1,F13/S1,F14 transaction, either of these two transactions may be the first to complete successfully and to cause the transition from NOT COMMUNICATING state to COMMUNICATING. In this event, the other transaction shall remain open regardless of the transition to COMMUNICATING until it is closed in a normal manner.

If the equipment has not yet sent[6] an S1,F14 to a previously received S1,F13 at the time when COMMUNICATING becomes active, the S1,F14 response shall be sent in a normal manner. A failure to send the S1,F14 is then treated as any other communication failure.

If the equipment-initiated S1,F13/S1,F14 is still open when the transition to COMMUNICATING occurs, subsequent failure to receive a reply from the host is considered a communication fault by equipment. An S9,F9 should be sent when a transaction timer timeout occurs[7]. (See Section 4.9 for definitions of communication faults and message faults, as well as detail on Stream 9 Error Messages.)

3.2.5 *State Transitions* — Table 3.2 contains a full description of the state transitions depicted in Figure 3.2.1.

When the operator switches from the DISABLED state to the ENABLED state, no collection event shall occur, since no messages can be sent until communications have been established. The process of establishing communications serves to notify the host that communications are ENABLED. No other collection events are defined for the Communications State Model.

6 This includes transmissions that may have started but not yet successfully completed at the time that the transition to COMMUNICATING occurs.

7 The existence of a transaction timer is not a requirement in some protocols, such as SMS (SEMI E13).

Table 3.2 Communications State Transition Table

#	*Current State*	*Trigger*	*New State*	*Action*	*Comment*
1	(Entry to COMMUNICA-TIONS)	System initialization.	System Default	None.	The system default may be set to DISABLED or ENABLED.
2	DISABLED	Operator switches from DISABLED to ENABLED.	ENABLED	None.	SECS-II communications are enabled.
3	ENABLED	Operator switches from ENABLED to DISABLED.	DISABLED	None.	SECS-II communications are prohibited.
4	(Entry to ENABLED)	Any entry to ENABLED state.	NOT COMMUNICATING	None.	May enter from system initialization to ENABLED or through operator switch to ENABLED.
5	(Entry to EQUIPMENT-INITIATED CONNECT)	(Any entry to NOT COMMUNICA-TING)	WAIT CRA	Initialize communications. Set CommDelay timer “expired.” Send S1,F13.	Begin the attempt to establish communications.
6	WAIT CRA	Connection transaction failure.	WAIT DELAY	Initialize CommDelay timer. Dequeue all messages queued to send.	If appropriate, dequeued messages shall be placed in spool buffer in the order generated. Wait for timer to expire.
7	WAIT DELAY	CommDelay timer expired.	WAIT CRA	Send S1,F13	Wait for S1,F14. May receive S1,F13 from Host.
8	WAIT DELAY	Received a message other than S1,F13.	WAIT CRA	Discard message. No reply. Set CommDelay timer “expired”. Send S1,F13.	Indicates opportunity to establish communications.
9	WAIT CRA	Received expected S1,F14 with COMMACK = 0.	COMMUNICATING	None.	Communications are established.
10	(Entry to HOST-INITIATED CONNECT)	(Any entry to NOT COMMUNICA-TING)	WAIT CR FROM HOST	None.	Wait for S1,F13 from Host.
14	COMMUNICATING	Communication failure. (See SEMI E4 or SEMI E37 for a protocol-specific definition of communication failure.)	NOT COMMUNICATING	Dequeue all messages queued to send.	Dequeued messages may be placed in spool buffer as appropriate.
15	WAIT CR FROM HOST	Received S1,F13.	COMMUNICATING	Send S1,F14 with COMMACK = 0.	Communications are established.

3.3 *Control State Model* — The CONTROL state model defines the level of cooperation between the host and equipment. It also specifies how the operator may interact at the different levels of host control. While the COMMUNICATIONS state model addresses the ability for the host and equipment to exchange messages, the CONTROL model addresses the equipment's responsibility to act upon messages that it receives.

The CONTROL model provides the host with three basic levels of control. In the highest level (REMOTE), the host may control the equipment to the full extent possible. The middle level (LOCAL) allows the host full access to information, but places some limits on how the host can affect equipment operation. In the lowest level (OFF-LINE), the equipment allows no host control[8] and only very limited information.[9]

The control model and communications model (when implemented) do not interact directly. That is, no action or state of one model directly causes a change in behavior of the other. It is true, however, that when the communication state is NOT COMMUNICATING then most message transaction are not functional. When messages cannot be transmitted, the control capabilities and all other GEM capabilities are affected.

Refer to Figure 3.3 as the CONTROL substates and state transitions are defined.

OFF-LINE

When the OFF-LINE state is active, operation of the equipment is performed by the operator at the operator console. While the equipment is OFF-LINE, message transfer is possible. However the use of messaging for any automation purpose is severely restricted. While the OFF-LINE state is active, the equipment will only respond to those messages used for the establishment of communications or a host request to activate the ON-LINE state.

While OFF-LINE, the equipment will respond with an Sx,F0 to any primary message from the host other than S1,F13 or S1,F17. It will process and respond to S1,F13 and S1,F17. S1,F17 is used by the host to request the equipment to transition to the ON-LINE state. The equipment will accept this request and send a positive response only when the HOST OFF-LINE state is active (see transition 11 definition below).

While the OFF-LINE state is active, the equipment shall attempt to send no primary message other than S1,F13,[10] S9,Fx,[11] and S1,F1 (see ATTEMPT ON-LINE substate). If the equipment receives a reply message from the host other than S1,F14 or S1,F2, this message is discarded.

No messages enter the spool when the system is OFF-LINE. Spooling may be active when the Communications State of NOT COMMUNICATING is active. This might occur during OFF-LINE, but since the equipment will not attempt to send messages except as mentioned in the previous paragraph[12], no messages will enter the spool.

OFF-LINE has three substates: EQUIPMENT OFF-LINE, ATTEMPT ON-LINE, and HOST OFF-LINE.

OFF-LINE/EQUIPMENT OFF-LINE

While this state is active, the system maintains the OFF-LINE state. It awaits operator instructions to attempt to go ON-LINE.

OFF-LINE/ATTEMPT ON-LINE

While the ATTEMPT ON-LINE state is active, the equipment has responded to an operator instruction to attempt to go to the ON-LINE state. Upon activating this state, the equipment attempts to send an S1,F1 to the host.

Note that when this state is active, the system does not respond to operator actuation of either the ON-LINE or the OFF-LINE switch.

8 The host may establish communications. This does not affect equipment operation and for that reason is not termed a control operation.

9 The host may determine the equipment identification via the S1F13/F14 transaction.

10 Sending of S1,F13 is based upon the COMMUNICATIONS state model.

11 S9,Fx messages may be issued only in response to the messages to which the equipment will normally respond while OFF-LINE (i.e. S1,F13 and S1,F17).

12 The equipment may send S1,F1 or S1,F13, but since Stream 1 messages are not eligible for spooling, they will not enter the spool either.

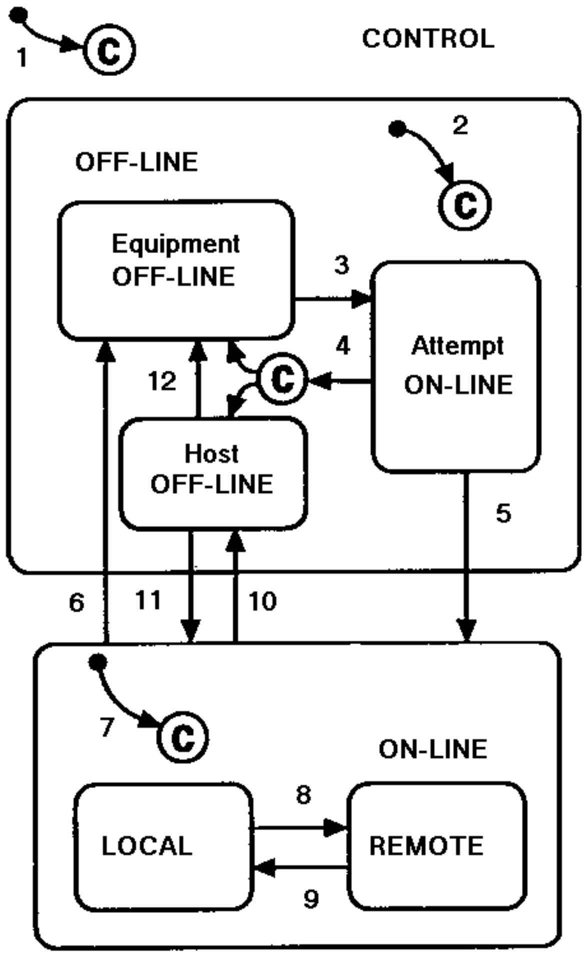


Figure 3.3
CONTROL State Model

OFF-LINE/HOST OFF-LINE

While the HOST OFF-LINE state is active, the operator's intent is that the equipment be ON-LINE. However, the host has not agreed. Entry to this state may be due to a failed attempt to go ON-LINE or to the host's request that the equipment go OFF-LINE from ON-LINE (see the transition table for more detail). While this state is active, the equipment shall positively respond to any host's request to go ON-LINE (S1,F17). Such a request shall be denied when the HOST OFF-LINE state is not active.

ON-LINE

While the ON-LINE state is active, SECS-II messages may be exchanged and acted upon. Capabilities that may be available to the host should be similar to those available from the operator console wherever practical.

The use of Sx,F0 messages is not required while the ON-LINE state is active. Their use is discouraged in this case. The only allowed use is to close open transactions in conjunction with message faults.

ON-LINE/LOCAL

Operation of the equipment is implemented by direct action of an operator. All operation commands shall be available for input at the local operator console of the equipment.

The host shall have the following capabilities and restrictions when the LOCAL state is active:

- The host shall be prohibited from the use of remote commands that cause physical movement or which initiate processing. During processing, the host shall be prohibited from the use of any remote command that affects that process.
- During processing, the host shall be prohibited from modifying any equipment constants that affect that process. Other equipment constants shall be changeable during processing. The host shall be able to modify all available equipment constants when no processing is in progress.
- The host shall be capable of initiating the upload and download of recipes to/from the recipe storage area on the equipment. The host shall be capable of selecting recipes for execution so long as this action does not affect any currently executing recipe.
- The host shall be able to configure automatic data reporting capabilities including alarms, event reporting, and trace data reporting. The host shall receive all such reports at the appropriate times.
- The host shall be able to inquire for data from the equipment, including status data, equipment constants, event reports, process program directories, and alarms.
- The equipment shall be able to perform Terminal Services as defined in GEM.

The host shall be allowed any other capabilities that were not specifically restricted in the above items as long as the LOCAL state is active.

NOTE 2: Capabilities mentioned above which are not implemented on a specific equipment may be ignored in this context.

ON-LINE/REMOTE

For equipment which supports the GEM capability of remote control (see Section 4.4), while the REMOTE state is active, the host shall have access, through the communications interface, to the necessary commands to operate the equipment through the full process cycle in an automated manner. The equipment does not restrict any host capabilities when REMOTE is active. The degree of control executed by the host may vary from factory to factory. In some cases, the operator maybe required to interact during remotely controlled processes. This interaction may involve set-up operations, operator assist situations, and others. This state is intended to be flexible enough to accommodate these different situations.

To support the different factory automation policies and procedures, it shall be possible to configure the equipment to restrict the operator in specific non-emergency procedures. These restrictions shall be configurable so that the equipment may be set up to allow the operator to perform necessary functions without contention with the host. The categories for configuration shall include (but are not limited to):

— change equipment constants (process-related),
— change equipment constants (non-process-related),
— initiate process program download,
— select process program,
— start process program,
— pause/resume process program,
— operator assist,
— material movement to/from equipment,
— equipment-specific commands (on a command-by-command basis if needed).

NOTE 3: Capabilities mentioned above which are not implemented on a specific equipment may be ignored in this context.

No capabilities that are available to the operator when the LOCAL state is active should be unconditionally restricted when the REMOTE state is active. The supplier may provide for configurable restriction of operator capabilities not included in the list above if desired. No configurability is necessary for any operator functions not available to the host.

The control functions must be shared to some degree between the host and the local operator. At the very least, the operator must have the capability to change the CONTROL state, actuate an Emergency Stop, and interrupt processing (e.g., STOP, ABORT, or PAUSE). All of these capabilities except Emergency Stop may be access-limited.[13]

The host software should be designed to be compatible with the capabilities allotted to the operator.

13 Definition of the method of limiting operator access (password, key, etc.) to a capability is not within the scope of this document.

Table 3.3 CONTROL State Transition Table

#	*Current State*	*Trigger*	*New State*	*Action*	*Comments*
1	(Undefined)	Entry into CONTROL state (system initialization).	CONTROL (Substate conditional on configuration).	None	Equipment may be configured to default to ON-LINE or OFF-LINE . (See NOTE 1.)
2	(Undefined)	Entry into OFF-LINE state.	OFF-LINE (Substate conditional on configuration.)	None	Equipment may be configured to default to any substate of OFF-LINE.
3	EQUIPMENT OFF-LINE	Operator actuates ON-LINE switch.	ATTEMPT ON-LINE	None	Note that an S1,F1 is sent whenever ATTEMPT ON-LINE is activated.
4	ATTEMPT ON-LINE	S1,F0.	New state conditional on configuration.	None	This may be due to a communication failure (See NOTE 2), reply timeout, or receipt of S1,F0. Configuration may be set to EQUIPMENT OFF-LINE or HOST OFF-LINE.
5	ATTEMPT ON-LINE	Equipment receives expected S1,F2 message from the host.	ON-LINE	None	Host is notified of transition to ON-LINE at transition 7.
6	ON-LINE	Operator actuates OFF-LINE switch.	EQUIPMENT OFF-LINE	None	"Equipment OFF-LINE" event occurs. (See NOTE 3.) Event reply will be discarded while OFF-LINE is active.
7	(Undefined)	Entry to ON-LINE state.	ON-LINE (Substate conditional on REMOTE/LOCAL switch setting.)	None	"Control State LOCAL" or "Control State REMOTE" event occurs. Event reported based on actual ON-LINE substate activated.
8	LOCAL	Operator sets front panel switch to REMOTE.	REMOTE	None	"Control State REMOTE" event occurs.
9	REMOTE	Operator sets front panel switch to LOCAL.	LOCAL	None	"Control State LOCAL" event occurs.
10	ON-LINE	Equipment accepts "Set OFF-LINE" message from host (S1,F15).	HOST OFF-LINE	None	"Equipment OFF-LINE" event occurs.
11	HOST OFF-LINE	Equipment accepts host request to go ON-LINE (S1,F17).	ON-LINE	None	Host is notified to transition to ON-LINE at transition 7.
12	HOST OFF-LINE	Operator actuates OFF-LINE switch.	EQUIPMENT OFF-LINE	None	"Equipment OFF-LINE" event occurs.

NOTE 1: The configuration mentioned for transitions 1 and 2 should be a single setting. This would provide the user with a choice of entering the EQUIPMENT OFF-LINE, ATTEMPT ON-LINE, HOST OFF-LINE, or ON-LINE states.

NOTE 2: Communication failures are protocol specific. Refer to the appropriate protocol standard (e.g., SEMI E4 or SEMI E37) for a protocol-specific definition of communication failure.

NOTE 3: Any host initiated transaction open at the equipment must be completed. This may happen either by sending the appropriate reply to the host prior to sending the event message or by sending an Sx,F0 message following the event message (i.e., after the transaction).

3.4 *Equipment Processing States* — The behavior of the equipment in the performance of its intended function must be documented. This processing state model is highly dependent on the equipment process, technology, and style. However, there are expected to be common aspects to these models.

The Processing State Diagram, Figure 3.4, is provided as an example of an implementation model. This model demonstrates the expected nature of the processing state model documentation. There is no requirement that these specific states be implemented.

The equipment must generate collection events for each processing state transition, as well as provide status variables (ProcessState, PreviousProcessState) whose values are the current processing state and the previous processing state.

In referring to the Processing State Diagram, note that the initialization state INIT is not an actual processing state. It is shown here simply to indicate that the IDLE processing state is entered upon completion of equipment system initialization. On the following pages detailed descriptions are provided for the equipment processing states and state transitions (numbered) as shown in the diagram.

3.4.1 *Description of Equipment Processing States*

IDLE

In this state the equipment is awaiting instructions.

PROCESSING ACTIVE

This state is the parent of all substates where the context of process program execution exists.

PROCESS

This state is the parent of those substates that refer to the active preparation and execution of a process program.

SETUP

In this state all external conditions necessary for process execution are satisfied, such as ensuring material is present at the equipment, input/output ports are in the proper state, parameters such as temperature and pressure values are within limits, etc. If all setup operations are already complete, then this becomes a fall through state and a transition to the next state takes place.

READY

In this state the equipment is ready for process execution and is awaiting a START command from the operator or the host.

EXECUTING

Executing is the state in which the equipment is executing a process program automatically and can continue to do so without external intervention.

PAUSE

In this state processing is suspended and the equipment is awaiting a command.

Each state transition is defined in the following table. Note that all transitions in this table should be considered collection events.

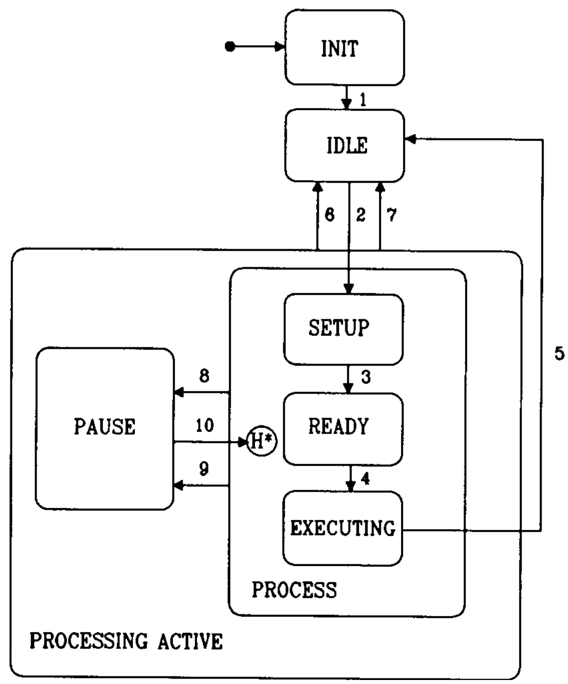


Figure 3.4
Processing State Diagram

Table 3.4 Processing State Transition Table

#	*Current State*	*Trigger*	*New State*	*Action*	*Comments*
1	INIT	Equipment initialization complete.	IDLE	None	None
2	IDLE	Commit has been made to set up.	SETUP	None	None
3	SETUP	All setup activity has completed and the equipment is ready to receive a START command.	READY	This activity is equipment-specific.	None
4	READY	Equipment has received a START command from the host or operator console.	EXECUTING	This activity is equipment-specific.	None
5	EXECUTING	The processing task has been completed.	IDLE	None	None
6	PROCESSING ACTIVE	Equipment has received a STOP command from host or operator console.	IDLE	None	None
7	PROCESSING ACTIVE	Equipment has received an ABORT command from host or operator console.	IDLE	This activity is equipment-specific.	None
8	PROCESS	The equipment decides to PAUSE due to a condition such as alarm.	PAUSE	This activity is equipment-specific.	For this type of problem, an operator assist is usually required.
9	PROCESS	Equipment has received a PAUSE command from host or operator console.	PAUSE	This activity is equipment-specific.	None
10	PAUSE	Equipment has received a RESUME command from host or operator console.	Previous PROCESS substate	This activity is equipment-specific.	None

4 Equipment Capabilities and Scenarios

This section describes the details of the capabilities required by GEM and provides scenarios for their use. Capabilities are operations performed by semiconductor manufacturing equipment. These operations are initiated through the communications interface using SECS-II messages. A scenario is a group of SECS-II messages arranged in a sequence to perform a capability. Other information may be included with the scenario for clarity. For each capability, the reader is provided with a statement of purpose, pertinent definitions, a detailed description, requirements, and scenarios.

The following capabilities are discussed:

Establish Communications

Event Notification

Dynamic Event Report Configuration

Variable Data Collection

Trace Data Collection

Limits Monitoring

Status Data Collection

On-line Identification

Alarm Management

Remote Control

Equipment Constants

Process Program Management

Material Movement

Equipment Terminal Services

Error Messages

Clock

Spooling

Control

4.1 *Establish Communications* — The Establish Communications capability provides a means of formally establishing communications following system initialization or any loss of communications between communicating partners, and thus of notifying the communication partner that a period of non-communication has occurred.

4.1.1 *Purpose* — Communications between host and equipment are formally established through use of the Establish Communications Request/Establish Communications Acknowledge transaction.

The use of S1,F1/F2 for this purpose is ambiguous since the transaction can be used for other purposes and may occur at any time.

The S1,F13/F14 transaction, used in conjunction with the Communications State Model, provides a means for equipment to notify the host, or the host to notify the equipment, that there has been a period of inability to communicate. The successful completion of this transaction also signals a possible need for synchronization activities between host and equipment.

4.1.2 *Definitions*

COMMACK — Acknowledge code returned in the Establish Communications Acknowledge message. See the SEMI E5 Standard for a full definition of this data item.

EstablishCommunicationsTimeout — An equipment constant used to initialize the interval between attempts to re-send an Establish Communications Request. This value specifies the number of seconds for the interval. See the SEMI E5 Standard for a full definition of this variable data item.

4.1.3 *Description* — There are potential problems when one side of the communications link fails and the other side does not detect it. From the point of view of the host, a loss of communications has many possible causes. In some cases, host-controlled settings on the equipment may need to be reset. In other cases, the equipment may have continued an automatic processing sequence during the period of no communication and may have changed states. The definition of a formal protocol for establishing communications alerts the host to the need to synchronize itself with the equipment's current status.

Equipment shall consider communications as formally established whenever either of the following conditions have been satisfied:[14]

— Communications Request has been sent to the host and an Establish Communications Acknowledge has been received within the transaction timeout period and with an acknowledge code of Accept, or

— Communications Request has been received from the host, and an Establish Communications Acknowledge response has been successfully sent with an acknowledge code of Accept.

When the equipment sends an Establish Communications Request to the host, this notifies the host of the possible need to synchronize itself with the equipment.

When the equipment is attempting to establish communications, an Establish Communications Request shall be sent periodically until communications have been formally established as described above. The interval between attempts must be user-configurable and begins as soon as a connection transaction failure is detected (see Section 3.2).

Attempting to establish communications is not a low-level connectivity issue, but rather a logical application issue used by either party to notify its partner that the host may need to perform synchronization activities with the equipment.

4.1.4 *Requirements*

— Equipment must support the Communication State Model (see Section 3.2).

— Equipment must provide the EstablishCommunicationsTimeout equipment constant described above.

14 Satisfaction of either of these conditions will result in a transition to the COMMUNICATING substrate. See Section 3.2 for further detail.

4.1.5 *Scenarios*

4.1.5.1 *Host Attempts to Establish Communications*

```
COMMENT                               HOST    EQUIPMENT                         COMMENT

                                                           Communications state is Enabled
                                                           (any substate)
Establish Communications  S1,F13-->
Request
                                  <--S1,F14                Reply COMMACK = Accept and
                                                           Communications state =
                                                           COMMUNICATING
```

4.1.5.2 *Equipment Attempts to Establish Communications and Host Acknowledges*

```
COMMENT                                 HOST    EQUIPMENT                         COMMENT

                    Communications State = NOT COMMUNICATING
                                                           [LOOP]
                                                           [LOOP]--SEND
                                  <--S1,F13                Establish Communications Request
Establish Communications  S1,F14-->
Acknowledge                                                [IF] S1,F14 rcvd w/o timeouts
                                                           [THEN] exit_loop--SEND
                                                           [ELSE] Delay for interval in
                                                                  Establish Commuications-
                                                                  Timeout
                                                           [ENDIF]
                                                           [END_LOOP]--SEND
                                                           [IF] COMMACK = Accept
                                                           [THEN] Communications state=
                                                                  Communicating
                                                                  exit_loop--
                                                           [ELSE] Reset timer for delay,
                                                                  and delay for interval
                                                                  specified in
                                                                  EstablishCommunications—
                                                                  Timeout
                                                           [ENDIF]
                                                           [END_LOOP]
```

4.1.5.3 *Simultaneous Attempts to Establish Communications* — For equipment that supports interleaving, it is possible that either the host or equipment could send an Establish Communications Request before receiving the request from its partner. As communications are established by the successful acceptance of any one Establish Communications Request, it is immaterial who sends the request first. The roles of host and equipment may be reversed.

Equipment Receives S1,F14 From Host Before Sending S1,F14:

COMMENT	**HOST**	**EQUIPMENT**	**COMMENT**
Communications State = NOT COMMUNICATING			
		<--S1,F13	Establish Communications Request
Establish Communications Request	S1,F13-->		
Reply COMMACK = Accept	S1,F14-->		S1,F14 received from Host and Communications established[15] and Communications state = COMMUNICATING
		<--S1,F14	Reply COMMACK = Accept[16]

Equipment Sends S1,F14 To Host Before Receiving S1,F14:

COMMENT	**HOST**	**EQUIPMENT**	**COMMENT**
Communications State = NOT COMMUNICATING			
		<--S1,F13	Establish Communications Request
Establish Communications Request	S1,F13-->		
		<--S1,F14	Reply COMMACK = Accept[15] Communications established[15] and Communications state = COMMUNICATING
Reply COMMACK = Accept	S1,F14-->		S1,F14 received from Host

4.2 *Data Collection* — Data collection allows the host to monitor equipment activity via event reporting, trace data reporting, limits monitoring, and query of selected status or other variable data.

4.2.1 *Event Data Collection* — Event data collection provides a dynamic and flexible method for the user to tailor the equipment to meet individual needs with respect to data representation and presentation to the host. The event-based approach to data collection provides automatic notification to the host of equipment activities and is useful in monitoring the equipment and in maintaining synchronization with the equipment.

Event data collection may be broken into two logical parts: host notification when an event occurs and dynamic configuration of the data attached to the event notification.

4.2.1.1 *Event Notification* — This section describes the method of notifying the host when equipment collection events occur.

4.2.1.1.1 *Purpose* — This capability provides data to the host at specified points in equipment operation. This asynchronous reporting eliminates the need for the host to poll the equipment for this information. Events on the equipment may trigger activity on the part of the host. Also, knowledge of the occurrence of events related to

15 Communications are establisted at the successful completion of the S1,F13/14 transaction where COMMACK is set to zero.

16 Communications are established on the successful transmission of S1,F14, even if there is an open S1,F13.

the equipment state models allows the host to track the equipment state. An equipment's behavior is related to its current state. Thus, this capability helps the host understand how an equipment will behave and how it will react to host behavior.

4.2.1.1.2 *Definitions*

Collection Event — An event (or grouping of related events) on the equipment that is considered significant to the host.

Collection Event ID (CEID) — A unique identifier of a collection event. See the SEMI E5 Standard for a full definition of this data item.

Event — A detectable occurrence significant to the equipment.

Report — A set of variables predefined by the equipment or defined by the host via S2,F33/F34.

4.2.1.1.3 *Detailed Description* — The equipment supplier must provide a set of predefined collection events. Specific collection events are required by individual capabilities and state models. Examples of collection events include:

— The completion of each action initiated by a host requested command,

— Selected processing and material handling activities,

— Operator action detected by the equipment,

— A state transition,

— The setting or clearing of an alarm condition on the equipment, and

— Exception conditions not considered alarms.

See Section 6 for a list of required collection events.

The reporting of a collection event may be disabled per event by the user to eliminate unwanted messages. An event report message shall be sent to the host upon the occurrence of a particular collection event if the collection event (CEID) has been enabled. Attached to each event message is one or more event reports which contain variable data. Section 4.2.1.2 describes the capability which allows for the dynamic customization of event reports. The values of any data contained in an event report message must be current upon the occurrence of the event. This implies that event reports be built at the time of the event occurrence.

The equipment shall also provide the S6,F15/F16 transaction to allow the host to request the data from a specific event report.

4.2.1.1.4 *Requirements*

— The equipment supplier shall provide documentation of all collection events defined on the equipment and the conditions for each event to occur.

— The equipment supplier shall provide unique CEIDs for each of the various collection events that are available for reporting.

— The equipment supplier shall provide a method for enabling and disabling the reporting of each event. This method shall either be available via the host interface (see Section 4.2.1.2) or the equipment's operator console.

— For each event, the equipment supplier shall provide either

- a default set of report(s) linked to the event which contain data pertinent to that event, or
- the ability for the user to configure the data linked to that event via the equipment's operator console or host interface (see Section 4.2.1.2).

4.2.1.1.5 *Scenarios*

Collection Event Occurs on the Equipment:

```
COMMENT                           HOST    EQUIPMENT                          COMMENT

                                                   [IF] Event Report is Multi-block
                                         <--S6,F5    [THEN] send Multi-block inquire
Multi-block grant                S6,F6-->
                                                   [ENDIF]
                                         <--S6,F11   Equipment sends Event Report
Host acknowledges Event Report S6,F12-->
```

Host Requests Event Report:

```
COMMENT                           HOST    EQUIPMENT                           COMMENTS

Host requests an event report S6,F15-->
                                     <--S6,F16    Equipment sends event report.
```

4.2.1.2 *Dynamic Event Report Configuration* — This section describes a capability which allows the host to dynamically modify the equipment event reporting setup.

4.2.1.2.1 *Purpose* — This capability is defined to provide the data reporting flexibility required in some manufacturing environments. It allows the host to increase or decrease the data flow according to need. For example, if the performance of an equipment degrades, the data flow from that equipment may be increased to help diagnose the problem.

4.2.1.2.2 *Definitions*

EventsEnabled — A variable data item that consists of a list of currently enabled collection events (CEIDs). See SEMI E5 for a full definition of this variable data item.

Report ID (RPTID) — A unique identifier of a specific report. See SEMI E5 for a full definition of this data item.

Variable Data (V) — A data item containing status (SV), data (DVVAL), or constant (ECV) values. See SEMI E5 for a full definition of this data item.

Variable Data ID (VID) — A unique identifier of a variable data item. The set of VID's is the union of all SVID's, ECID's, and ID's for DVVAL's (DVNAME's). See SEMI E5 for a full definition of this data item.

4.2.1.2.3 *Detailed Description* — The equipment shall support the following event report configuration functionality through the SECS-II interface:

— Host definition/deletion of custom reports,

— Host linking/unlinking of defined reports to specified collection events, and

— Host enabling/disabling the reporting of specified collection events.

NOTE 4: The equipment may also supply alternative means for defining reports and linking reports to events (e.g., via the operator console). Implementation of alternate means is not required.

The equipment can be instructed by the host to enable or disable reporting of collection events on an individual or collective basis. A status value (SV) shall be available that consists of a list of enabled collection events. (See Section 5.2, Variable Item List, EventsEnabled variable.)

Reports may be attached to an event report message (S6,F11). These reports are specifically linked to the desired event and typically contain variable data relating to that event. The reports may be provided by the equipment supplier or created by the user. The user must be able to create reports and link them to events via the SECS-II interface.

The data reported in the event report messages may consist of Status Values (SV's), Equipment Constant Values (ECV's), or Data Values (DVVAL's). Note that data values shall be valid and current on certain events and certain states and might not be current at other times. The implementor shall document when a data value will be current and available for reporting.

4.2.1.2.4 *Requirements*

— The equipment manufacturer must provide documentation of all variable data available from the equipment. This is to include variable name, variable type or class (SV, ECV, DVVAL), units, format codes, possible range of values, and a description of the meaning and use of this variable.

— The equipment manufacturer must provide unique VIDs for the various variable data (V) available for data collection in the equipment. For example, this means that no SV shall have a VID which is the same as the VID of any ECV or DVVAL.

— All variable data must be available for report definition and event data collection. See Section 5.2, Variable Item List, for a list of required variable data.

— All report definitions, report-to-event links, and enable/disable status of event reports must be retained in non-volatile storage.

4.2.1.2.5 *Scenario*

Collection Event Reporting Set-up:

```
COMMENT                              HOST   EQUIPMENT                             COMMENT

[IF] Define Report is Multi-block
[THEN] send Multi-block inquire S2,F39-->
                                        <-- S2,F40   Multi-block grant.
[ENDIF]
Send report definitions S2,F33-->                    DATAIDs, RPTIDs, and VIDs
                                                     received.
                                        <-- S2,F34   DRACK[17] = 0 the reports are OK
[IF] Link Event/Report is Multi-block
[THEN] send Multi-block inquire S2,F39-->
                                        <-- S2,F40   Multi-block grant.
[ENDIF]
Link reports to events S2,F35-->                     CEIDs and the corresponding
                                                     RPTIDs are received.
                                        <-- S2,F36   LRACK = 0 the event linkages
                                                     are acceptable.
Enable specific collection S2,F37-->                 Enable/ disable codes (CEEDs)
events                                               and the respective event
                                                     reporting CEIDs received.
                                        <-- S2,F38   ERACK = 0 OK, will generate the
                                                     specified reports when the
                                                     appropriate collection events
                                                     happen
```

4.2.2 *Variable Data Collection*

4.2.2.1 *Purpose* — This capability allows the host to query for the equipment data variables and is useful during initialization and synchronization.

4.2.2.2 *Definitions*

Report ID (RPTID) — A unique identifier of a specific report. See SEMI E5 for a full definition of this data item.

Variable Data (V) — A variable data item containing status, discrete, or constant data. See SEMI E5 for a full definition of this data item.

4.2.2.3 *Detailed Description* — The host may request a report containing data variables from the equipment by specifying the RPTID. It is assumed that the report has been previously defined (e.g., using the Define Report S2,F33 transaction (see Section 4.2.1)). The values of any status variables (SV's) and equipment constants (ECV's) contained within the report must be current. Discrete data values (DVVAL's) are only guaranteed to be valid upon the occurrence of a specific collection event. If DVVAL cannot be specified in equipment due to some restrictions depend on hardware and/or software conditions, the zero length item is reported.

4.2.2.4 *Requirements*

— Variable data items (V's) and associated units of measure must be provided by the equipment manufacturer.

17 Define Report Acknowledge Code, see SEMI E5 for full definition of this Data Item.

4.2.2.5 *Scenario*

Host Requests Report:

```
COMMENT                                   HOST    EQUIPMENT                              COMMENT

Host requests data variables
contained in report RPTID    S6,F19-->
                                         <--S6,F20     Equipment responds with a list of
                                                       variable data for the given
                                                       RPTID.
```

4.2.3 *Trace Data Collection*

4.2.3.1 *Purpose* — Trace data collection provides a method of sampling data on a periodic basis. The time-based approach to data collection is useful in tracking trends or repeated applications within a time window, or monitoring of continuous data.

4.2.3.2 *Definitions*

Data Sample Period (DSPER) — The time delay between samples. See SEMI E5 for a full definition of this data item.

Reporting Group Size (REPGSZ) — The number of samples included per trace report transmitted to the host. See SEMI E5 for a full definition of this data item.

Status Variable (SV) — Status data item (included in trace report). See SEMI E5 for a full definition of this data item.

Status Variable ID (SVID) — A unique identifier of a status variable. See SEMI E5 for a full definition of this data item.

Total Samples (TOTSMP) — Number of samples to be taken during a complete trace period. See SEMI E5 for a full definition of this data item.

Trace Request ID (TRID) — An identifier associated with a trace request definition. See SEMI E5 for a full definition of this data item.

4.2.3.3 *Detailed Description* — The equipment shall establish a trace report as instructed by the host (S2,F23). For a trace report (S6,F1), the host shall designate a name for the trace report (TRID), a time interval for data sampling (DSPER), the total number of samples to be taken (TOTSMP), the number of samples per trace report (REPGSZ), and a listing of which data will be sent with the report (SVID's). The number of trace reports sent to the host is determined by total samples divided by reporting group size (TOTSMP/REPGSZ).

The equipment shall sample the specified data (SV's) at the interval designated by the host (DSPER) and shall send a predefined trace report to the host for the specified reporting group size (REPGSZ). The trace report definition shall be automatically deleted from the equipment after the last trace report has been sent.

The host may modify or re-initiate a trace function currently in progress by specifying the same TRID in a trace request definition, at which point the old trace shall be terminated and the new trace shall be initiated, or the host can instruct the equipment to terminate a trace report prior to its completion by specifying TOTSMP = 0 for that TRID, at which point the trace report definition shall be deleted.

A detailed example is included as Application Note A.4.

4.2.3.4 *Requirements*

— The equipment must have a local mechanism (e.g., internal clock) for triggering the periodic sampling and transmission of trace reports to the host.

— A minimum of four (4) concurrent traces shall be supported by the equipment. The same SVID may be collected in multiple traces simultaneously.

— All SVID's available at the equipment shall be supported for trace data collection. The exception to this is any SV that will not fit into a single block.

NOTE 5: SEMI E5 provides for SV's to be of a list format. Since this may in practice be a variable list, there is a potential problem with such an SV supported by the Trace Data Collection capability. This is a problem with the SEMI E5 standard. Care should be exercised in the use of SV's using the list format.

4.2.3.5 *Scenario*

Host Initiates Trace Report:

```
COMMENT                               HOST  EQUIPMENT                         COMMENT

Trace Data initialization
requested                        S2,F23-->
                                         <--S2,F24     Acknowledge, trace initiated
                                                       [DO] TOTSMP REPGSZ times
                                                       [DO] REPGSZ many times: collect
                                                       SVID1,...SVIDn data, delay
                                                       time by DSPER.
                                                       [END_DO]
                                         <--S6,F1      Send SV1,...SVn
Acknowledge receipt               S6,F2-->
                                                       [END_DO]
Optional:
Request trace termination prior to
completion (TOTSMP = 0)          S2,F23-->
                                         <--S2,F24     Acknowledge premature termination
```

4.2.4 *Limits Monitoring* — This capability relates to the monitoring of selected equipment variables and has three primary aspects:

— Defines a standard set of monitoring zones and limits.

— Provides for reporting to the host when selected equipment variables transition between monitoring zones.

— Empowers the host to modify the values of the variable limit attributes for these same selected equipment variables.

4.2.4.1 *Purpose* — The limits monitoring capability provides the host a means of monitoring equipment conditions by a flexible, efficient, and asynchronous method which is consistent across equipment. It eliminates the need for constant polling of equipment by the host for current status values. Further, this capability allows the host to implement changes in the monitoring range as needed. This capability has application to both production operation and diagnostic/testing scenarios, and it also has applicability to statistical process control.

4.2.4.2 *Definitions*

LimitVariable — DVVAL containing the VID of a specific equipment variable for which a zone transition collection event has been generated.

EventLimit — DVVAL containing the LIMITID of the limit crossed by LimitVariable.

TransitionType — DVVAL which defines the direction of the zone transition which has occurred: 0 = transition from lower to upper zone, 1 = transition from upper to lower zone.

Limit — Used in this section to represent the set of variable limit attributes that completely describe a variable monitoring “barrier.” The attributes include VID, Units, UPPERDB, LOWERDB, LIMITMAX, and LIMITMIN. In some contexts it may be interpreted more narrowly as the combination of UPPERDB and LOWERDB.

$LIMITID_n$ — Refers to the identifier of a specific limit (as defined by UPPERDB and LOWERDB) among the set of limits for a monitored equipment variable. LIMITIDs are consecutively numbered, beginning at one through the number of limits possible (seven minimum).

Monitoring Zone — A subset of the possible range of values for a variable of interest to the host. A single limit divides the range into two zones. Multiple limits may be combined to divide the range even further.

Zone Transition — The movement of a variable value from one monitoring zone to another. This transition is a collection event and has a corresponding CEID.

Deadband — An overlap of two zones implemented to prevent constant zone transitions by a variable sitting on or near a limit (i.e., “chattering”).

UPPERDB — A variable limit attribute that defines the upper boundary of the deadband of a limit.[18] The value applies to a single limit (LIMITID) for a specified VID.

18 The format and units must be the same as the format of the variable being monitored.

Thus, UPPERDB and LOWERDB as a pair define a limit.

LOWERDB — A variable limit attribute that defines the lower boundary of the deadband of a limit.[18] The value applies to a single limit (LIMITID) for a specified VID. Thus, UPPERDB and LOWERDB as a pair define a limit.

UPPER ZONE — The range of values lying above a limit.

LOWERZONE — The range of values lying below a limit.

LIMITMAX — The maximum value for any limits of a specific equipment variable. This value is set by the equipment manufacturer and typically coincides with the maximum value allowed for the monitored variable.

LIMITMIN — The minimum value for any limits of a specific equipment variable.[19] This value is set by the equipment manufacturer and typically coincides with the minimum value allowed for the monitored variable.

Undefined — When used in reference to variable limits, it indicates that monitoring/reporting of zone transitions involving that particular limit are disabled.

4.2.4.3 *Description* — The limits monitoring capability provides the host with a minimum of seven configurable limits or barriers that may be applied to selected equipment status variables (SV's) of the types floating point, integer, and boolean. When one of these barriers is crossed, a collection event is generated to alert the host to a change in monitoring zone or state of the monitored variable. These seven limits may be combined in a variety of ways to match the needs of the host system.[19] An illustration of a combination of five of the limits to provide one type of variable monitoring is shown in Figure 4.2.1.[20] This section describes the key aspects of limits monitoring. Detailed implementation examples of limits monitoring are provided as Application Note A.7.

NOTE 6: While the SEMI E5 standard allows SV's to be lists, such variable lists are not allowed under this capability.

4.2.4.3.1 *Monitoring Limit Characteristics* — A limit is defined by a set of attributes that include the variable (VID) to which the limit corresponds, the units of that variable, the maximum and minimum possible values of the limit (LIMITMAX and LIMITMIN) and the specific borders of the limit (UPPERDB and LOWERDB). See Figure 4.2.2. There is a limitation to the values of UPPERDB and LOWERDB which may be stated as:

LIMITMAX≥UPPERDB≥LOWERDB≥LIMITMIN

A limit divides the possible range of variable values into two parts, the upper zone and the lower zone. At any time, the monitored variable is considered to be in one and only one of these zones. However, as Figure 4.2.2 shows, these two zones have an area of overlap. This is called the deadband.

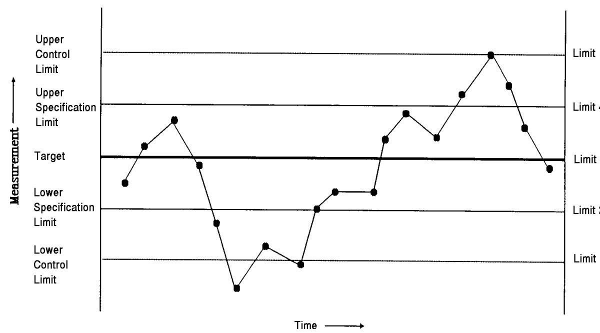


Figure 4.2.1
Limit Combination Illustration: Control Application

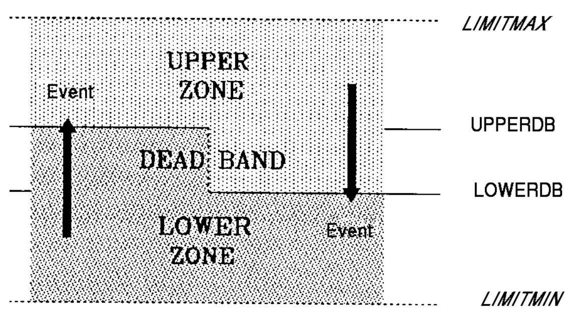


Figure 4.2.2
Elements of One Limit

The deadband is a key concept of limits monitoring, especially for floating point variables. Its purpose is to prevent a phenomenon known as chattering — the repeated changing of zones due to small, rapid fluctuations in variable value while near the zone boundary. In practice, the value of a variable must reach the opposite boundary of the deadband before a zone transition can occur. Thus, if a variable's value reaches the UPPERDB and transitions into the upper zone, it will not return to the lower zone until it falls back to the

19 Note that while at least seven limits per variable are available from the equipment, the host need not use all seven.

20 This illustration shows the reading which might be available to the equipment, not the limit excursions reported to the host. Reporting is covered later in the section.

LOWERDB. The difference between UPPERDB and LOWERDB should always be greater than the typical amplitude of those fluctuations deemed to be insignificant. In some cases, the width of the deadband may set to zero (i.e., UPPERDB = LOWERDB). At first glance, this would seem to make indeterminate the current zone when an integer value sits on the limit. This is not the case, however, when movement of the value is considered. To illustrate, an example is given, assuming that UPPERDB = LOWERDB = 100. The list shows consecutive readings of the variable and the resultant zone:

99 Lower Zone (Initial Reading)

101 Upper Zone (Zone Transition)

100 Lower Zone (Zone Transition)

100 Lower Zone

99 Lower Zone

100 Upper Zone (Zone Transition)

Transition from one zone into another generates a collection event, as might be reported via S6,F11. The host has the option of receiving notification by enabling event reporting for the event. For each variable that has monitoring capability, one CEID is reserved to indicate zone transitions for that variable. To aid in the determination of the nature of a transition event, three DVVAL's have been defined:

LimitVariable — The VID of the monitored variable to which the collection event refers.

EventLimit — Contains the LIMITID of the limit reached or crossed by LimitVariable.

TransitionType — Defines the direction of the zone transition which has occurred: 0 = transition from lower to upper zone, 1 = transition from upper to lower zone.

Sampling frequency is an important element of limits monitoring and should be considered during equipment specification. If changes in variable value are relatively fast compared to sampling frequency, it is possible for some zone transitions to be missed or for multiple zone transitions to occur between readings. Since it is possible for zone transitions to occur "simultaneously" or for limits to be identically defined, the DVVAL EventLimit has been defined to allow for a list of multiple zone transitions of a variable to be reported with a single collection event.

It also should be emphasized that a single CEID is used to report transitions in both directions across a limit. Thus, reporting for one direction but not the other cannot be configured.

The functionality of each limit for each variable can be described with the state model shown in Figure 4.2.3. Below, the three states are described more fully, followed by a table defining the transitions.

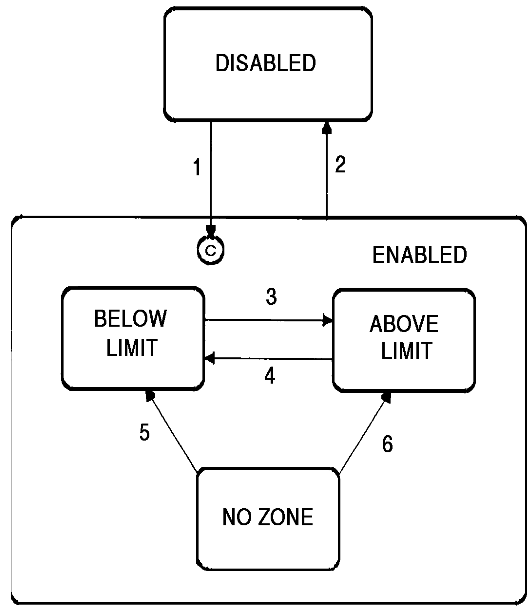


Figure 4.2.3
Limit State Model

ABOVE LIMIT

A variable is considered to be above a limit when its value increases to equal or exceed the upper boundary of the deadband, UPPERDB. The significance attached to this state is a function of the host's usage.

BELOW LIMIT

A variable is considered to be below a limit when its value decreases to equal or fall below the lower boundary of the deadband, LOWERDB. The significance attached to this state is a function of the host's usage.

NO ZONE

In some circumstances it is possible for the variable value to be in neither the upper zone nor the lower zone. This may occur upon definition of a new limit or upon equipment startup when the value of the variable lies in the deadband. In this case, the active state of the limit is considered to be NO ZONE. The limit shall remain in this state until the variable value reaches either boundary of the deadband.

4.2.4.3.2 *Modification of Limit Values* — Values for the monitoring limits on any monitored variable may be modified by the host using the message transaction

S2,F45/F46 (Define Variable Limit Attributes). The equipment must reject any S2,F45 message which contains limit information which conflicts with the following rules:

— LIMITMAX≥UPPERDB≥LOWERDB≥LIMIT-MIN;

— If either UPPERDB or LOWERDB is defined, both must be defined; if either UPPERDB or LOWERDB is undefined, both must be undefined.

The first rule is defined and graphically depicted in Figure 4.2.2. The second rule refers to the host's ability to turn any limit "on" or "off". While a minimum of seven limits must be available for each monitored variable, it will be common for the host application to require less than seven or even none of the limits be used. The limits not needed can be disabled by leaving the values for UPPERDB and LOWERDB "undefined". Limits may be disabled for a VID or for all monitored VIDs by using zero length lists in the S2,F45 message.

All monitored variables must be one of three types: integer, floating point, or Boolean. This may be accomplished by using the following formats: 11, 20, 3(), 4(), 5().

NOTE 7: The binary format is not allowed. If the ASCII format is used, the equipment shall perform a conversion into one of the numeric types before performing any value comparisons, both for limit validations and zone transitions.

4.2.4.3.3 *Limit Values Request* — The host may request the current limit values for a specified VID using the message transaction S2,F47/F48 (Variable Limit Attribute Request).

4.2.4.4 *Requirements*

— A minimum of seven limits per monitored variable must be available.

— One CEID per monitored variable must be supplied for zone transition reporting.

— All limit definitions must be kept in non-volatile storage.

— The equipment must enforce the limit validation rules defined above.

— The specification and documentation of which variables may be monitored with this capability is the responsibility of the equipment manufacturer based on the specific instance of equipment. This subject also may be addressed by equipment models of classes of semiconductor equipment.

Table 4.2 Limit State Transition Table

#	*Current State*	*Trigger*	*New State*	*Action*	*Comment*
1	DISABLED	Limit attributed defined w/ S2,F45.	ENABLED	None	The substate of ENABLED is determined by the current value of the monitored variable.
2	ENABLED	Limit attributes set to undefined w/ S2,F45.	DISABLED	None	None
3	BELOW LIMIT	Variable Increased to be ≥UPPERDB	ABOVE LIMIT	None	Zone Transition.
4	ABOVE LIMIT	Variable decreases to be ≤LOWERDB	BELOW LIMIT	None	Zone Transition.
5	NO ZONE	Variable decreases to be ≤LOWERDB	BELOW LIMIT	None	Zone Transition.
6	NO ZONE	Variable increases to be ≥UPPERDB	ABOVE LIMIT	None	Zone Transition.

4.2.4.5 *Scenarios* — Zone Transition Event occurs in equipment:

```
COMMENTS                          HOST    EQUIPMENT                             COMMENTS

Send enabled event report to host as shown in Section 4.2.1.
```

Host defines Limit Attributes:

```
COMMENTS                          HOST    EQUIPMENT                             COMMENTS

[IF] S2,F45 is Multi-block
[THEN] Send Multi-block inquire S2,F39-->
                                        <--S2,F40    Multi-block grant.
[END IF]
Host defines new variable limit
attributes.                     S2,F45-->
                                        <--S2,F46    Equipment acknowledges host
                                                     request.
```

Host queries equipment for current Limits:

```
COMMENTS                          HOST    EQUIPMENT                             COMMENTS

Host queries equipment          S2,F47-->
for current variable limit
attribute definitions.
                                        <--S2,F48    Equipment returns report
                                                     containing requested variable
                                                     limit attribute values.
```

4.2.5 Status Data Collection

4.2.5.1 *Purpose* — This capability allows the host to query the equipment for selected status information and is useful in synchronizing with equipment status.

4.2.5.2 *Definitions*

Status variable value (SV) — A data item containing the value of a status variable. See SEMI E5 for a full definition of this data item.

Status variable ID (SVID) — A unique identifier of a status variable. See SEMI E5 for a full definition of this data item.

4.2.5.3 *Detailed Description* — The host may query equipment status by specifying the desired SVID's. Upon such a request, the equipment sends the host the value of the selected status variables. The host also may request the description (name and units) of any or all available status variables.

4.2.5.4 *Requirements*

— The equipment manufacturer must provide unique SVID's for the various status variables (SV) available for data collection in the equipment.

— All status data is available for status data collection. See Section 5.2, Variable Item List, for a list of status variables.

— All SV's must contain valid data whenever provided to the host.

4.2.5.5 *Scenario*

Request Equipment Status Report:

```
COMMENT                            HOST   EQUIPMENT                       COMMENT

Host requests report of selected
status variable values             S1,F3-->
                                          <--S1,F4     Equipment responds with the
                                                       requested status variable
                                                       data.
```

Request Equipment Status Variable Name list:

```
COMMENT                            HOST   EQUIPMENT                       COMMENT

Host requests equipment to identify
selected status variables.        S1,F11-->
                                          <--S1,F12    Equipment responds with the
                                                       requested status variable
                                                       descriptions.
```

4.2.6 *On-line Identification*

4.2.6.1 *Purpose* — Implementation of SEMI E5 (a GEM Fundamental Requirement) requires the equipment to accept the S1,F1 from the Host at any time while it is ONLINE and COMMUNICATING, and respond with S1,F2. The On-line Identification capability describes the host-initiated scenario. The equipment-initiated scenario is used for a different purpose and is defined in sections 3.3 and 4.12 describing the GEM "Control Model".

4.2.6.2 *Definitions*

Equipment Model Type (MDLN) — ASCII string containing the equipment model. See SEMI E5 for a full definition of this data item.

Equipment Software Revision Code (SOFTREV) — ASCII string containing the equipment software revision. See SEMI E5 for a full definition of this data item.

4.2.6.3 *Detailed Description* — On-line Identification allows the host to verify the presence and identity of the equipment.

4.2.6.4 *Requirements* — (Host-Initiated) An S1,F2 response from the equipment must provide MDLN and SOFTREV information which reflects the hardware and software configuration of the equipment.

SOFTREV must uniquely identify different releases of equipment software. Any change in equipment software must result in a corresponding change to SOFTREV.

The equipment-initiated S1,F1 is not required except as described in sections 3.3 and 4.12 describing the GEM "Control Model".

4.2.6.5 *Scenario:*

Host Initiated

```
COMMENT                            HOST   EQUIPMENT                       COMMENT

Are you there?                     S1,F1-->
                                          <--S1,F2     Equipment replies with MDLN and
                                                       SOFTREV.
```

4.3 *Alarm Management* — The alarm management capability provides for host notification and management of alarm conditions occurring on the equipment.

4.3.1 *Purpose* — Historically, a precise definition of an equipment alarm has been absent. Consequently, differing interpretations have resulted in inconsistent implementations. This is addressed by providing a more rigorous definition (see definition in Section 4.3.2 below) of an alarm.

In addition, it is often important for equipment to report more extensive information to the host than has been available in the S5,F1/F2 (Alarm Report Send/ Acknowledge) transaction. The data required in such cases is very dependent on equipment type, host information requirements, and alarm situation. This issue is addressed by providing event reporting methods that are tied to alarm state changes.

Lastly, the alarm management capability provides mechanisms for:

— Reporting the time of an alarm state change,

— Uploading a list of alarm texts,

— Enabling and disabling the notification of specific alarms, and

— Host query of alarms set and enabled status on the equipment.

4.3.2 *Definitions*

Alarm — An alarm is related to any abnormal situation on the equipment that may endanger people, equipment, or material being processed. Such abnormal situations are defined by the equipment manufacturer based on physical safety limitations. Equipment activities potentially impacted by the presence of an alarm shall be inhibited.

Note that exceeding control limits associated with process tolerance does not constitute an alarm nor do normal equipment events such as the start or completion of processing.

AlarmsEnabled — Status value consisting of a list of the alarm ID's currently enabled for reporting to the host. See SEMI E5 for a full definition of this variable data item.

AlarmsSet — Status value consisting of a list of the alarm ID's currently in the ALARM SET (or unsafe) state. See SEMI E5 for a full definition of this variable data item.

ALCD — Alarm code data item used in the S5,F1 (Alarm Report Send) and S5,F6 (List Alarm Data) messages. This code is divided into two parts, the alarm set/cleared bit and the 7 bit alarm category code. Only the set/cleared bit is used—bit 8 = 1 means alarm set, = 0 means alarm cleared. The alarm category code is not used. See SEMI E5 for a full definition of this data item.

ALID — Alarm identifier. See SEMI E5 for a full definition of this data item.

ALTX — Data item contained in the S5,F1 and S5,F6 messages containing a brief textual description of an alarm. See SEMI E5 for a full definition of this data item.

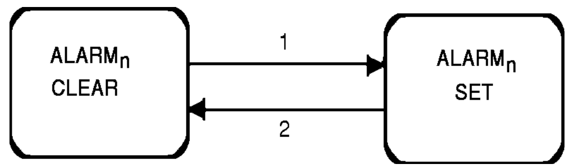


Figure 4.3
State Diagram for Alarm ALID$_n$

4.3.3 *Detailed Description* — Two alarm notification mechanisms are defined to achieve the flexibility necessary for the reporting required by host systems. First, stream 5 alarm reporting enables a brief, yet fixed, method for notification of alarm occurrences using the S5,F1/F2 transaction. Second, to address the host's potential need for more extensive and flexible data reporting, two collection events ("alarm set" and "alarm cleared") are defined for each possible alarm condition on the equipment to allow the use of event data collection mechanisms. In the latter case, reports are sent by the equipment using the Event Report/ Acknowledge transaction (see Section 4.2 on event data collection).

The alarm$_n$ detected and cleared events are derived from the state model for an alarm (see Figure 4.3 and Table 4.3.1). In this model each of n alarms can be in one of two possible states, either ALARM CLEAR or ALARM SET. The transition from the ALARM CLEAR to the ALARM SET state is defined as the collection event "Alarm$_n$ detected" (transition 1). Conversely, the transition from ALARM SET to ALARM CLEAR is defined as the collection event "Alarm$_n$ cleared" (transition 2).

NOTE 8: The alarm capability is intended as an addition to standard safety alarms (e.g., lights, horns). There is no intent to replace direct operator notification of such problems, nor is there the expectation that the host can prevent or directly address such alarms.

Table 4.3.1 Alarm State Transition Table

#	*Current*	*Trigger*	*New State*	*Action*	*Comment*
1	$ALARM_n$ CLEAR	$Alarm_n$ is detected on the equipment.	$ALARM_n$ SET	Initiate local actions (if any) to ensure safety. Update AlarmsSet and $ALCD_n$ values. Generate and issue alarm message if enabled.	Inhibited activities require operator or host intervention prior to resuming.
2	$ALARM_n$ SET	$Alarm_n$ is no longer detected on the equipment.	$ALARM_n$ CLEAR	Update AlarmsSet and $ALCD_n$ values. Generate and issue alarm message if enabled.	Inhibited activities require operator or host intervention prior to resuming.

The equipment manufacturer is responsible for identifying all alarms on their equipment by:

— Applying the above alarm definition,

— Consulting Application Note A.3 for examples of alarms for various equipment configurations,

— Noting that the presence of an alarm typically requires some action or intervention before resuming safe operation of the equipment, and by

— Referring to Table 4.3.2 below which delineates the differences between events and alarms.

Table 4.3.2

EVENT	*ALARM*
Any occurrence detectable by the equipment.	Related to only those occurrences that are abnormal, undesirable, AND endanger people, equipment, or physical material being processed.
Certain events may trigger a state transition(s).	Each alarm has an associated two-state state model: ALARM SET (or unsafe) and ALARM CLEAR (or safe).
Equipment activities are not necessarily inhibited by the occurrence of an event (unless it is associated with an alarm or intentional inhibit).	The presence of an alarm inhibits equipment activities to ensure safe operation until the alarm condition is cleared.
Certain events may occur in an expected sequence.	Alarms may occur at any time.

4.3.3.1 *Enable/Disable Alarms* — Upon request from the host, the equipment shall enable or disable reporting of certain alarms. Enabling or disabling a given alarm shall impact the communication of both the alarm set and clear messages equally (i.e., turn them both on or both off). This is not the case for enabling/disabling of the associated collection events, where the alarm-set and alarm-cleared events can be enabled and disabled separately. The current enable/disable settings must be stored in non-volatile memory. Changes to the enable/disable settings must be reflected in the AlarmsEnabled status value.

NOTE 9: The alarm itself is not being enabled or disabled, but the reporting of the alarm through SECS-II messages is being enabled or disabled.

4.3.3.2 *Send Alarm Report* — Upon detecting a change in the status of a given alarm, an associated alarm state model shall transition to the opposite state. Following initiation of local actions necessary to ensure safety, the equipment must update the AlarmsSet and associated ALCD values and send an alarm message and/or an event message to the host assuming one or both are enabled. Alarm messages must be sent before the corresponding event messages if both are enabled.

NOTE 10: The ALCD is divided into two parts, the alarm set/cleared bit and the 7 bit alarm category code. Only the set/cleared bit is used—bit 8 = 1 means alarm set, = 0 means alarm cleared. The alarm category code is not used.

4.3.3.3 *List Alarm Text* — Upon request from the host, the equipment sends values of alarm text associated with a specified list of alarm ID's using the S5,F5/F6 (List Alarms Request/Data) transaction.

4.3.3.4 *List Currently Set Alarms* — The host may obtain a listing of currently set alarms by employing any data collection method specified by GEM and referencing the variable data item called "AlarmsSet" (e.g., including " AlarmsSet in a report definition). When reported, AlarmsSet must contain a list of all currently set alarms. Each alarm set or clear occurrence must cause a change to the AlarmsSet status value prior to reporting it to the host.

4.3.3.5 *List Currently Enabled Alarms* — The host may obtain a listing of currently enabled alarms by employing any data collection method specified by GEM and referencing the variable data item called "AlarmsEnabled" (e.g., including AlarmsEnabled in a report definition). When reported, AlarmsEnabled must contain a list of all alarm ID's currently enabled for reporting. The equipment must update the value of AlarmsEnabled upon corresponding change(s) to the enable/disable settings.

4.3.4 *Requirements*

— A set of alarms relating to the physical safety limits of operator, equipment, or material being processed must be defined for the equipment by the equipment manufacturer.

— The equipment must maintain all enable/disable states and report definitions for alarms and collection events in non-volatile memory.

— Each alarm defined on the equipment must have a brief description of its meaning, an associated unique alarm identifier (ALID), alarm text (ALTX), an alarm status (ALCD) and two unique collection event identifiers (CEIDs), one for set and one for cleared.

— Enabled alarm reports must be sent prior to corresponding enabled event reports.

4.3.5 *Scenarios*

NOTE 11: Consult event reporting sections of this document for descriptions of enabling, disabling, and sending collection event reports.

Enable/Disable Alarms:

```
COMMENTS                         HOST   EQUIPMENT                          COMMENT

Enable/disable Alarm             S5,F3-->
                                        <-- S5,F4    Acknowledge
```

Upload Alarm Information:

```
COMMENTS                         HOST   EQUIPMENT                          COMMENT

Request alarm data/text          S5,F5-->
                                        <-- S5,F6    Send alarm data/text
```

Send Alarm Report:

```
COMMENTS                         HOST   EQUIPMENT                          COMMENT

Alarm occurrence detected by the equipment.
                                        <-- S5,F1    Send alarm report (if enabled).
Acknowledge                      S5,F2-->
                                        <-- S6,F11   Send event report (if enabled).
Acknowledge                     S6,F12-->
```

4.4 *Remote Control*

4.4.1 *Purpose* — This capability provides the host with a level of control over equipment operations.

4.4.2 *Definitions*

Host Command Parameter (CPNAME/ CPVAL/CEPVAL) — A parameter name/value associated with a particular host command (S2,F41/S2,F49). The equipment manufacturer must provide unique names (CPNAMEs) for any supported command parameters. Command parameters are not specified in this document but are left to equipment manufacturers to define. Equipment models of specific classes of semiconductor equipment also may address this issue. Note that if there are no associated parameters a zero-length list is sent. The data item CEPVAL, which can be defined as a list, allows grouping of related parameters within a main parameter. If the CEPVAL is defined as a single (non-list) item, then it is the equivalent of a CPVAL.

The uses of OBJSPEC in the header structure of the S2,F49 Enhanced Remote Command allows the equipment supplier to define a set of unique identifiers for different objects within the equipment such as: equipment sub-systems, sub-system components, processing stations, ports, and exchange stations.

4.4.3 *Description* — The equipment responds to host commands that provide the following functions relative to individual equipment implementations:

— Start processing

— Select a process program or recipe

— Stop processing

— Temporarily suspend processing

— Resume processing

— Abort processing

Additional commands may be implemented by the equipment manufacturer (e.g., vent chamber, clear material, open door).

Remote commands shall be interpreted as "request action be initiated" rather than "do action." The equipment may then respond via S2,F42/S2,F50 with HCACK = 4 if the command "is going to be performed." This alleviates any transaction timeouts for commands that may take a long time to perform. The completion of the action initiated by the remote command (i.e., HCACK = 0 or 4) must result in either a state transition or other action that generates a collection event upon normal/abnormal completion.

The format for all remote commands is ASCII, with a maximum length of 20 characters. The character set is restricted to the printable characters (hexadecimal 21 through 7E). Note that spaces are not allowed.

The following remote commands (RCMDs), if implemented on the equipment, shall be supported as described below (see Section 3.4 for a description of Equipment Processing States).

NOTE 12: The terms "current cycle" and "safe break point" used below are to be defined by the supplier or within the models of classes of semiconductor equipment.

START — This command is available to the host when a process program or recipe has been selected and the equipment is in the "ready" processing state. The START command instructs the equipment to initiate processing. Variable parameter settings may be included as name/value command parameters CPNAME/CPVAL/CEPVAL.

PP-SELECT — This command instructs the equipment to make the requested process program(s) available in the execution area. The process programs (PPIDs) are specified via the command parameter list. A status variable (PPExecName) contains the PPID of the process program(s) currently selected.

RCP-SELECT — This command uses the Enhanced Remote Command S2,F49 to instruct the equipment to prepare the requested recipes for execution in the execution area. The recipes and variable parameters are specified via command parameter lists. Each recipe specification may be accompanied by new variable parameter settings, if any, in the command parameter list. A status variable RcpExecName contains the recipe specifiers or identifiers of the recipes currently selected.

STOP — Command to complete the current cycle, stop in a safe condition and return to the "idle" processing state. Stop has the intent of stopping the process. The equipment is not required to support the continuation of processing. Stop leaves material either fully processed or partially processed so that the processing can be later completed. For example, for a single wafer process tool, five wafers have been processed while the remaining wafers remain unprocessed.

PAUSE — Command to suspend processing temporarily at the next safe break point. Pause has the intent of resuming the process at the same point where it was paused. The process may be RESUMED, STOPPED, or ABORTED while in a PAUSED condition. RESUME shall be able to continue the process from the same point where it was paused.

RESUME — Command to resume processing from the point where the process was paused.

ABORT — Command to terminate the current cycle prior to its completion. Abort has the intent of immediately stopping the process and is used because of abnormal conditions. Abort makes no guarantee about the subsequent condition of material. In the above example, the wafers being processed at the time of the abort may not be completely processed. Other AbortLevels > 1 may be defined by the manufacturer or addressed by models of specific classes of semiconductor equipment.

CPNAME = AbortLevel, CPVAL = 1 means terminate current cycle at the next "safe break point," retrieve all material, stop in a safe condition and return to the idle state in the processing state machine.

4.4.4 *Requirements*

— The following Remote Commands, as defined under Descriptions, must be implemented on equipment to satisfy minimum requirements for this capability:

  — START

  — STOP

— The RCMD value for all commands supported on the equipment must be recognized if sent with all upper-case characters (e.g., "STOP", "START", "PP-SELECT", "PAUSE", etc.). In addition to accepting strings with all upper-case characters, the equipment can optionally accept strings with all lower-case characters or mixed-case strings. The equipment documentation should describe whether or not the optional lower-case or mixed-case strings are supported.

— Stream 2 currently provides for Host Command Send and Enhanced Remote Command. The equipment shall support one or both methods, based on appropriateness.

— The Enhanced Remote Command is used to address size, complexity, or the need to target a specific sub-system within the equipment, (i.e., processing station, port, exchange station, material handler, chamber).

4.4.5 *Scenarios*

4.4.5.1 *Host Command Send Scenario*

```
COMMENTS                          HOST    EQUIPMENT                          COMMENT

Host Command                 Send S2,F41-->
                                           <-- S2,F42   Host Command Acknowledge

                                                        [IF] Command Accepted
                                                        (HCACK = 0 or 4)

                                           <-- S6,F11   [THEN] Event Report-state
                                                        change or other collection
                                                        event occurrence.
Event Report Acknowledge          S6,F12-->
```

4.4.5.2 *Enhanced Remote Command Scenario*

COMMENTS	HOST	EQUIPMENT	COMMENT
Enhanced Remote Command	S2,F49-->		
		<--S2,F50	EnhancedRemote Command Acknowledge
			[IF] Command Accepted (HCACK = 0 or 4)
		<--S6,F11	[THEN] Event Report-state change or other collection event occurrence.
Event Report Acknowledge	S6,F12-->		

4.5 *Equipment Constants*

4.5.1 *Purpose* — This capability provides a method for the host to read and to change the value of selected equipment constants on the equipment.

4.5.2 *Definitions* — None.

4.5.3 *Description* — This capability allows the host to reconfigure equipment constants to support a variety of situations. The following functions are included:

Host Sends Equipment Constants — Allows the host to change the value of one or more equipment constants.

Host Equipment Constant Request — Allows the host to determine the current value of equipment constants.

Host Equipment Constant Namelist Request — Allows the host to retrieve basic information about the equipment constants available at the equipment.

4.5.4 *Requirements*

— Equipment constants must be stored in non-volatile memory.

— The equipment must be in a "safe" condition to accept new constant(s) settings as defined by the equipment manufacturer.

— The equipment must provide a collection event to alert the host whenever an equipment constant is changed by the operator. Information indicating which constant was changed shall be available for the event report.

4.5.5 *Scenarios*

Host Sends Equipment Constants:

COMMENTS	HOST	EQUIPMENT	COMMENTS
Host sends	S2,F15-->		
equipment constants		<-- S2,F16	EAC = 0 equipment sets constants

Host Equipment Constants Request:

COMMENTS	HOST	EQUIPMENT	COMMENTS
Host constant request	S2,F13-->		
		<-- S2,F14	Equipment constant data

NOTE: This capability also can be accomplished using S6,F19 & S6,F20. See Section 4.2.

Host Equipment Constant Namelist Request:

```
COMMENTS                          HOST    EQUIPMENT                              COMMENTS

Host constant namelist          S2,F29-->
request                                 <-- S2,F30   Equipment constant namelist
```

Operator Changes Equipment Constant

```
COMMENTS                                  HOST   EQUIPMENT                       COMMENTS

                                                     Operator changes equipment
                                                     constant at equipment
                                                     operator console.
                                        <-- S6,F11   Equipment reports equipment
                                                     constant change.
Host acknowledges event         S6,F12-->
```

4.6 *Process Program Management* — Process programs and recipes must be managed through interaction between the equipment and the host system.

4.6.1 *Purpose* — Process program management provides a means to transfer process programs or recieps, and to share the management of those process programs or recipes, between the host and equipment.

4.6.2 *Definitions*

PPError — A text data value with information about verification errors of a process program that failed verification. If the equipment provides an event for recipe verification and/or recipe verification failure, then PPError shall be a DVVAL. Otherwise, PPError shall be an SV.

PPFormat — A variable (SV) indicating the type or types of process programs and recipes that are supported.

1 = Unformatted process programs

2 = Formatted process programs

3 = Both unformatted and formatted process programs

4 = Execution recipes

4.6.2.1 *Definitions for Process Programs*

Process Program — A process program is the pre-planned and reusable portion of the set of instructions, settings, and parameters under control of the equipment that determine the processing environment seen by the manufactured object and that may be subject to change between runs or processing cycles.

Process Program Identifier — A text string (PPID) used to identify a process program.

Formatted Process Program — A process program that is presented as an ordered sequence of command codes with their associated parameters as dictated by S7,F23, and S7,F26. Where formatted process programs are supported, equipment must also provide information sufficient to allow a user at the host to create, display, modify, and partially verify their contents (for example, that information provided in S7,F22).

Unformatted Process Program — An unformatted process program is transferred without structure as the single data item PPBODY (refer to SEMI E5 for a complete description of PPBODY).

Process Program Change Event — The collection event associated with the occurrence of the creation, modification, or deletion of a process program by the operator.

PPChangeName — A data value (DVVAL) containing the PPID of the process program affected by the event Process Program Change Event. See SEMI E5 for a full definition of this variable data item.

PPChangeStatus — The action taken on the process program named by PPChangeName. This variable is valid for the collection event Process Program Change Event. See SEMI E5 for a full definition of this variable data item.

PPExecName — The status variable containing the PPID(s) of the currently selected process program(s). See SEMI E5 for a full definition of this variable data item.

PP-SELECT — The remote command used to select one or more process programs for execution. The process programs are specified by PPID via the command parameter list.

Process Program Verification — Verification is syntax checking of a process program. Verification ensures only that a process program is structured correctly. It does not ensure that the program has the correct parameters to run a particular process or product (see Process Program Validation). Equipment supporting unformatted process programs should provide a variable DVVAL PPError that provides information to the user concerning the error or errors when an attempt to verify a process program fails.

NOTE 14: It may not be possible for the equipment to verify unformatted process programs other than to check the size of the program and internal program checksums. Equipment has no standard means of indicating the type of error encountered in an unformatted process program.

Process Program Validation — Validation is type-and-range checking of parameters in a process program, and is performed after verification.

4.6.2.2 *Definitions for Recipes*

Execution recipe — A type of recipe stored by the equipment for purposes of editing, verification, and execution.

For complete definitions of execution recipes and their standard attributes, see SEMI E42, Section 6.

Execution Recipe Change Event — The collection event associated with the occurrence of the modification or deletion of an execution recipe stored by the equipment.

Note 15: A recipe is modified whenever its body is changed.

New Execution Recipe Event — The collection event associated with the creation of a new execution recipe at the equipment.

Object form recipe — A recipe with body in a proprietary format that may be presented without structure.

RcpChangeName — A data value (DVVAL) containing the identifier of the recipe affected by the event Execution Recipe Change Event or New Execution Recipe Event. See the SEMI E5 Standard for a full definition of this variable data item.

RcpChangeStatus — The action taken on the recipe named by RcpChangeName. This variable is valid for the collection event Execution Recipe Change Event or New Execution Recipe Event. See the SEMI E5 Standard for a full definition of this variable data item.

RcpExecName — The status variable containing the specifiers of the currently selected recipe(s). See the SEMI E5 Standard for a full definition of this variable data item.

RCP-SELECT — The remote command used to select one or more recipes for execution. See Section 4.4.3.

Recipe Attribute — Information about the recipe that is transferred with the recipe as a name/value pair. The value may be a single item or a list.

Recipe — A recipe contains both a set of instructions, settings, and parameters that the equipment uses to determine the processing environment (its body or process program) and a set of attributes that provide information about the recipe, such as the date and time the body was last changed.

SEMI E42 defines two types of recipes: *managed recipes* and *execution recipes*. For purposes of GEM, the term *recipe* refers to an *execution recipe* only.

Recipe identifier — A recipe identifier is a formatted text string (RCPID) used to identify the recipe.

Recipe specifier — A formatted text string (RCPSPEC) used in messages to indicate a specific recipe. A recipe specifier includes the recipe identifier. It may also include additional information, such as the name of the specific component of the equipment where the recipe is to be executed (e.g. a process chamber) and the name of a recipe repository on the host.

Recipe Verification — Verification is syntax checking of a recipe's body. Verification ensures that a recipe body is structured correctly and has the correct syntax. It may also provide a check of semantics. It does not ensure that the body has the correct parameters to run a particular process or product (see Recipe Validation).

NOTE 16: Unverified recipes shall be verified upon download.

Recipe Validation — Validation is type-and-range checking of parameters in a recipe, and is performed when the recipe is selected for execution. The recipe may be correct in its syntax and semantics but should fail validation if it can not be executed with the current equipment configuration.

Source form recipe — A recipe with a body that is presented as an ordered sequence of text. A source form recipe may be created and edited off-line to the equipment. Definition of syntax requirements shall be documented, in order to allow proper off-line editing.

Variable Parameters — Variable parameters are recipe parameters that are defined in the body of the recipe and whose run-time values may be set outside of the recipe when the recipe is selected for execution and/or when processing is started. Both the host and the operator may specify new settings as a parameter name/value pair.

Variable Parameter Definition — A variable parameter definition has three parts: the name of the variable parameter, its default setting, and restrictions on the run-time value selected. Variable parameter definitions are stored in the recipe attribute "Parameters".

4.6.3 *Description*

4.6.3.1 *Process Program Description*

Process programs allow the equipment's process, and/or the parameters used by that process, to be set and modified by the engineer to achieve different results. Different process programs maybe required for different products, while often the same process program will be used for all lots of a given product. The engineer must be able to create such programs, to modify current programs, and to delete programs from equipment storage.

For the host to ensure that the proper process programs are in place at the equipment, there must be a means of transferring them from equipment to host and from host to equipment. The host also may need to delete process programs from the equipment's storage to make room for a process program to be downloaded. In addition, the host must be kept informed whenever a local change occurs in the contents or status of a process program.

Both formatted and unformatted process programs may be uploaded and downloaded. This capability provides for both host- and equipment-initiated transfers. The equipment-initiated transfer may be used at the request of the process engineer or operator at the equipment.

If a process program exists with the same PPID as the one given in the SECS-II message, the old process program must be replaced. The PPID in the e process program in non-volatile storage.

4.6.3.2 *Recipe Description*

Specifications in Section 4.6.3.1 apply to recipes as well as process programs, with the following differences:

- A recipe contains a body corresponding to a process program. In addition, it contains attributes defined for execution recipes in SEMI E42, Section 6. Recipe attributes are transferred whenever the recipe is downloaded or uploaded.
- The same SECS-II messages are used for all execution recipes, regardless of the internal structure of the recipe body.
- If an execution recipe already exists with the same identifier as the one given in the SECS-II message, the downloaded recipe shall be rejected (not stored) unless the host has specified a "forced overwrite" in the data item RCPOWCODE.
- A recipe currently being edited shall be protected from inadvertent change or overwriting by a recipe with the same identifier that is downloaded during this time. If the downloaded recipe is accepted (stored), the equipment shall require the operator either to save the edited recipe to a new (unused) identifier or to discard it.
- For the equipment to initiate either an upload or download of a recipe, it shall request the host to initiate an upload or download procedure. In addition, it may be necessary to also specify the name of the repository (recipe namespace) at the host.

4.6.4 *Requirements*

- The equipment manufacturer shall provide a method to create, modify, and delete process programs or recipes. This method shall exist on either the equipment or on a separate computing system.
- A CEID shall be defined for a collection event for the creation, the deletion, or the modification (completion of an editing session) of a process program (Process Program Change Event). For recipes, there are two separate CEIDs and collection events, one for the creation of a new recipe (New Execution Recipe Event) and one when a recipe is changed or deleted (Execution Recipe Change Event). A New Execution Recipe Event shall occur whenever a new recipe identifier is created through download, edit, copy, or rename operations. A Execution Recipe Change Event shall occur whenever the body of an existing recipe is modified.
- The name (identifier) that the engineer or operator uses to refer to the process program or recipe is the same as the identifier used by the host.
- Upon request from the host or operator, the equipment shall perform the following actions with regard to process programs and recipes stored in non-volatile storage: upload, download, delete, and list current equipment process program or recipe directory.
- The equipment shall be able to store in non-volatile memory the number of process programs or recipes sufficient to execute three unique process cycles. For example, if a wire-bonder requires both an "ALIGN" process program and a "BOND" process program for a full process cycle, then it must provide non-volatile storage for at least three pairs of process programs. These stored process programs or recipes may not be modified in any way by the execution process, nor may the execution process be affected by the modification of any process program or recipes in storage, either by downloading or by local editing, while that process program or recipes is being executed.

- The equipment must provide verification and validation of all downloaded process programs and recipes.
- Stream 7 provides for formatted and unformatted process programs, while Stream 15 provides for recipes. The equipment must support at least one of these three methods.
- The equipment supplier shall document any restrictions on the length or test format of PPID. The maximum length allowed by equipment may be less than that allowed by SECS-II.
- Where recipes are supported, the following requirements also apply:
  - The variable PPFormat shall be provided to indicate to the user the messages supported by the equipment.
  - The recipe and its attributes shall comply to the requirements for execution recipes as defined in SEMI E42, Section 6.

4.6.5 *Scenarios for Process Programs*

4.6.5.1 *Process Program Creation, Editing, or Deletion Process Program Created, Edited, or Deleted by Operator*

```
COMMENTS                          HOST   EQUIPMENT                      COMMENTS

                                                     New process program created,
                                                     edited, or deleted by operator
                                                     at equipment.
                                                     PPChangeName = PPID
                                                     PPChangeStatus = 1 (Created)
                                                                    = 2 (Edited)
                                                                    = 3 (Deleted)
                                                     [IF] CEID for Process Program
                                                     Change Event enabled
                                                     [THEN]
                                         <-- S6,F11  Send Event Report
Event Report Acknowledge         S6,F12-->
                                                     [END_IF]
```

Process Program Deletion by the Host:

```
COMMENTS                          HOST   EQUIPMENT                      COMMENTS

Delete Process Program Send      S7,F17-->
                                         <-- S7,F18  Delete Process Program
                                                     Acknowledge.
                                                     The process program is removed
                                                     from non-volatile storage.
```

4.6.5.2 *Process Program Directory Request*

```
COMMENTS                          HOST   EQUIPMENT                      COMMENTS

Current EPPD Request             S7,F19-->
                                         <-- S7,F20  Current EPPD Data
```

4.6.5.3 *Process Program Upload*

Host-Initiated Process Program Upload — Formatted:

```
COMMENTS                        HOST    EQUIPMENT                          COMMENTS

Formatted Process Program      S7,F25-->
Request                                 <-- S7,F26   Formatted Process Program
                                                     Data
```

Host-Initiated Process Program Upload — Unformatted:

```
COMMENTS                        HOST    EQUIPMENT                          COMMENTS
Process Program Request          S7,F5-->
                                        <-- S7,F6[21]   Process Program Data
```

Equipment Initiated Process Program Upload — Formatted:

```
COMMENTS                        HOST    EQUIPMENT                          COMMENTS

                                                     [IF] Process program is multi-
                                                          block
                                                     [THEN]
                                        <-- S7,F1    Process Program Load Inquire
Process Program Load Grant       S7,F2-->
                                                     [END_IF]
                                        <-- S7,F23   Formatted Process Program Send
Formatted Process Program
Acknowledge                     S7,F24-->
```

Equipment-Initiated Process Program Upload — Unformatted:

```
COMMENTS                        HOST    EQUIPMENT                          COMMENTS

                                                      [IF] Process program is multi-
                                                     block
                                                      [THEN]
                                        <-- S7,F1     Process Program Load Inquire
Process Program Load Grant       S7,F2-->
                                                      [END_IF]
                                        <-- S7,F3     Process Program Send
Process Program Acknowledge      S7,F4-->
```

21 If the process program does not exist, a zero-length list will be sent.

4.6.5.4 *Process Program Download*

NOTE 17: Formatted process programs must be verified immediately following any download.

While the Process Program Load Inquire/Grant transaction (S7,F1/F2) is required only for the transfer of multi-block process programs, its use is recommended prior to all host-initiated downloads. It provides a means of verifying process program size.

Host-Initiated Process Program Download — Formatted:

```
COMMENTS                          HOST   EQUIPMENT                          COMMENTS

[IF]
Process program is multi-block
[THEN]
Process Program Load Inquire   S7,F1[22] -->
                                         <-- S7,F2    Process Program Load Grant
[ENDIF]
Formatted Process Program Send  S7,F23-->
                                         <--S7,F24    Formatted Process Program
                                                      Acknowledge Verify process
                                                      program
                                                      [IF] S7,F27 is multi-block
                                                      [THEN]
                                         <--S7,F29    Process Program Verification
                                                      Inquire
                                  S7,F30->            Process Program Verification
                                                      Grant
                                                      [END IF]
                                         <--S7,F27    Process Program Verification
                                                      Send
Process Program Verification
Acknowledge                     S7,F28-->
```

Host-Initiated Process Program Download — Unformatted:

```
COMMENTS                          HOST   EQUIPMENT                          COMMENTS

[IF]
Process program is multi-block
[THEN]
Process Program Load Inquire   S7,F1[21] -->
                                         <--S7,F2     Process Program Load Grant
[END_IF]
Process Program Send              S7,F3-->
                                         <--S7,F4     Process Program Acknowledge
```

22 S7,F1 should be used only to request permission to transfer a multi-block formatted or unformatted process program. It should not be used to select a process program. For selecting a process program for execution, the remote command PP-SELECT should be used.

Equipment-Initiated Process Program Download — Formatted:

COMMENTS	HOST	EQUIPMENT	COMMENTS
		<--S7,F25	Formatted Process Program Request
Formatted Process Program Data	S7,F26-->		
			[IF] S7,F27 is multi-block [THEN]
		<--S7,F29	Process Program Verification Inquire
	S7,F30-->		Process Program Verification Grant
			[END IF]
		<--S7,F27	Process Program Verification Send
Process Program Verification Acknowledge	S7,F28-->		

Equipment-Initiated Process Program Download — Unformatted:

COMMENTS	HOST	EQUIPMENT	COMMENTS
		<--S7,F5	Process Program Request
Process Program Send	S7,F6-->		

4.6.6 Scenarios for Recipes

4.6.6.1 *Recipe Creation, Editing, or Deletion*

Recipe Created by Operator:

COMMENTS	HOST	EQUIPMENT	COMMENTS
			New recipe created by operator at equipment RcpChangeName = RCPID RcpChangeStatus = 1 (Created) [IF] CEID for New Execution Recipe Event enabled [THEN]
		<--S6,F11	Send Event Report
Event Report Acknowledge	S6,F12-->		
			[END_IF]

Recipe Edited or Deleted by Operator:

COMMENTS	HOST	EQUIPMENT	COMMENTS
			New recipe edited, or deleted at equipment RcpChangeName = RCPID RcpChangeStatus = 2 (Modified) or 5 (Deleted) [IF] CEID for Execution Execution Recipe Change Event enabled [THEN]
		<--S6,F11	Send Event Report
Event Report Acknowledge	S6,F12-->		
			[END_IF]

Recipe Deletion by the Host:

COMMENTS	HOST	EQUIPMENT	COMMENTS
Recipe Delete Request	S15,F35-->		
		<--S15,F36	Recipe Delete Acknowledge. The recipe is removed from non-volatile storage.

4.6.6.2 *Recipe Directory Request*

Host requests a list of identifiers of currently stored recipes.

COMMENTS	HOST	EQUIPMENT	COMMENTS
GetAttr Request	S14,F1-->		
		<--S14,F2	GetAttr Data

4.6.6.3 *Recipe Upload*

Host-Initiated Recipe Upload:

COMMENTS	HOST	EQUIPMENT	COMMENTS
Recipe Upload Request	S15,F31-->		
		<--S15,F32	Recipe Upload Data

Equipment-Initiated Recipe Upload:

COMMENTS	HOST	EQUIPMENT	COMMENTS
			RCPCMD = Upload
			[IF] multi-block request
		<--S15,F1	[THEN] Recipe Management Multi-block inquire
Recipe Management Multi-block Grant	S15,F2-->		
			[END_IF]
		<--S15,F21	Recipe Action Request
Recipe Action Acknowledge	S15,F22-->		
Host requests upload	S15,F31-->		
		<--S15,F32	Recipe Upload Data

4.6.6.4 *Recipe Download*

The Recipe Management Multi-block Inquire/Grant transaction (S15,F1/F2) is required for the transfer of multi-block recipes. However, its use is recommended prior to all downloads, as it provides recipe size to the equipment.

NOTE 18: If the data item RCPOWCODE is TRUE in S15,F27, then a pre-existing recipe with the same identifier shall be overwritten.

Host-Initiated Recipe Download:

```
COMMENTS                          HOST   EQUIPMENT                             COMMENTS
[IF]
Recipe is multi-block
[THEN]

Recipe Management Multi-
block Inquire                    S15,F1-->

                                         <--S15,F2    Recipe Management Multi-block
                                                      Grant
[END IF]
Recipe Download Request         S15,F27-->
                                                      [IF] RCPOWCODE  =  TRUE
                                                      delete any pre-existing recipe
                                                      with the same identifier before
                                                      storing new recipe
                                                      [END_IF]
                                         <--S15,F28   Recipe Download Acknowledge
```

Equipment-Initiated Recipe Download:

```
COMMENTS                          HOST   EQUIPMENT                             COMMENTS
                                                      RCPCMD  =  Download
                                                      [IF] multi-block request
                                         <--S15,F1    [THEN] Recipe Management Multi-
                                                      block inquire
Recipe Management Multi-block    S15,F2-->
Grant
                                                      [END_IF]
                                         <--S15,F21   Recipe Action Request

Recipe Action Acknowledge       S15,F22-->
[IF]
Recipe is multi-block
[THEN]

Recipe Management Multi-block    S15,F1-->
Inquire
                                         <--S15,F2    Recipe Management Multi-block
                                                      Grant
[END IF]
Recipe Download Request         S15,F27-->
                                                      [IF] RCPOWCODE  =  TRUE
                                                      delete any pre-existing recipe
                                                      with the same identifier before
                                                      storing new recipe
                                                      [END_IF]
                                         <--S15,F28   Recipe Download Acknowledge
```

4.6.6.5 *Recipe Verification*

Host requests equipment to verify a recipe that it has stored:

COMMENTS	HOST	EQUIPMENT	COMMENTS
Recipe Verify Request	S15,F29-->		
			Equipment verifies recipe
		<--S15,F30	Recipe Verify Data

4.7 *Material Movement* — The material movement capability includes the physical transfer of material (WIP, tools, expendable materials, etc.) between equipment, buffers, and storage facilities. The transfer of material can be performed by operators, AGV robots, tracks, or dedicated fixed material handling equipment.

4.7.1 *Purpose* — This capability is limited in implementation, serving to notify the host of the appearance or removal of material at the equipment's ports.

4.7.2 *Definitions*

Port — A point or area on the equipment at which a change in equipment ownership of material may occur.

4.7.3 *Description* — This capability consists of alerting the host whenever material is sent or received from any of the ports on the equipment. Event-specific information, such as port identification and material identification, also may be useful, but definition of these and other related DVVAL's are left to the implementation.

4.7.4 *Requirements* — The equipment must supply two CEIDs, one to report when material is sent from any port and the other to report when material is received at any port.

4.7.5 *Scenarios*

COMMENTS	HOST	EQUIPMENT	COMMENTS
			Material is sent or received at an equipment port.
		<--S6,F11	Send collection event to host.
Host acknowledges.	S6,F12-->		

4.8 *Equipment Terminal Services* — Equipment Terminal Services allows the host to display information on the equipment's display device or the operator of the equipment to send information to the host.

4.8.1 *Purpose* — Equipment Terminal Services allows the factory operators to exchange information with the host from their equipment workstations.

4.8.2 *Definitions* —Message Recognition: a positive action by the equipment operator indicating the operator has viewed the text of a host initiated message.

4.8.3 *Detailed Description* — The equipment must be capable of displaying information passed to it by the host for the operator's attention. The information, or an indication of a message, must remain on the equipment's display until the operator indicates message recognition. Message recognition results in a collection event that informs the host that the operator has actually viewed the information.

The equipment must be capable of passing information to the host that has been entered from the operator's equipment console. This information is intended for host applications and is not processed by the equipment.

The equipment has no responsibility for interpreting any of the data passed to or from the host using this method.

4.8.4 *Requirements*

— Any new Terminal Display message sent by the host shall overwrite an unrecognized message at the same equipment terminal.

— The equipment must provide a display device capable of displaying at least 160 characters to the operator.

— The equipment must provide a mechanism for displaying information sent to it by the host.

— The equipment must provide an indicator to notify the operator when an unrecognized message is present.

— The equipment must provide a mechanism for the operator to indicate message recognition (e.g., push button, terminal function).

— The equipment must provide a means for alpha numeric data entry that can be used by the operator.

— The equipment must support operator entry of at least 160 characters per message.

— The equipment must have a mechanism to send operator-entered messages to the host.

— The equipment must support single-block messages as a minimum. Support of multi-block messages is optional.

— A Terminal Display message received by the equipment with a zero length TEXT data item shall be accepted and replace any previous unrecognized message, but shall not itself be considered an unrecognized message. This provides a method of clearing an unrecognized message and turning off the unrecognized message indicator.

4.8.5 *Scenarios*

Host sends information to an equipment's display device:

COMMENTS	HOST	EQUIPMENT	COMMENTS
Host sends textual information to equipment for display to the operator on terminal x.	S10,F3-->		
		<--S10,F4	Equipment acknowledges request to display text. (Equipment sets unrecognized message indicator.) Operator indicates message recognition. (Equipment clears unrecognized message indicator.)
		<--S6,F11	Message recognition event. (See Section 4.2.1, Event Data Collection, for details.)
Host acknowledges Optional:	S6,F12-->		
		<--S10,F1	Operator responds with text via terminal x.
Host acknowledges receipt of operator text.	S10,F2-->		

Host sends information to an equipment's display device and then overwrites the information before operator recognizes message:

COMMENTS	HOST	EQUIPMENT	COMMENTS
Host sends textual information to equipment for display to the operator on terminal x.	S10,F3-->		
		<--S10,F4	Equipment acknowledges request to display text. (Equipment sets unrecognized message indicator.)
Host sends textual information to equipment for display to the operator on terminal x. This message overwrites the first one sent by the host since it is still unrecognized.	S10,F3-->		
		<--S10,F4	Equipment acknowledges request to display text. (Equipment sets unrecognized message indicator.) Operator indicates message recognition. (Equipment clears unrecognized message indicator.)
		<--S6,F11	Message recognition event. (See Section 4.2.1, Event Data Collection, for details.)
Host acknowledges	S6,F12-->		

Operator sends information to the host:

COMMENTS	HOST	EQUIPMENT	COMMENTS
		<--S10,F1	Operator sends textual information via equipment terminal x.
Host acknowledges receipt of operator initiated message.	S10,F2-->		
Optional: Host responds with information for display to the operator on terminal x.	S10,F3-->		
		<--S10,F4	Equipment acknowledges receipt of request to display text. (Equipment sets unrecognized message indicator.) Operator indicates message recognition.
		<--S6,F11	Message recognition event. (See Event data collection for details.)
Host acknowledges	S6,F12-->		

Host sends a multi-block display message:

COMMENTS	HOST	EQUIPMENT	COMMENTS
Send information	S10,F5-->		
		<--S10,F6	Accepted or denied. [IF] Message from host is multi-block and multi-block is not supported by the equipment, [THEN]
		<--S10,F7	Send Multi-block Not Allowed. [END_IF]

4.9 *Error Messages*

4.9.1 *Purpose* — Error messages provide the host with information describing the reason for a particular message or communication fault detected by the equipment.

4.9.2 *Definitions*

Communication Fault — Refer to Section 2 for the definition.

Message Fault — Refer to Section 2 for the definition.

4.9.3 *Detailed Description* — The equipment must inform the host that it cannot process a message due to an incorrect:

— device ID,

— message stream type,

— message function,

— message format, or

— data format.

The equipment must inform the host if the message has more data than it can handle.

The equipment must inform the host if the equipment's transaction timer expires.

The equipment shall treat the above conditions as application-level errors and shall not take any further action on any message in error.

Error messages are invoked whenever the equipment detects communication or message faults.

4.9.4 *Requirements* — Support of all Stream 9 messages.

4.9.5 *Scenario*

Message Fault Due to Unrecognized Device ID:

COMMENTS	HOST	EQUIPMENT	COMMENTS
Host sends a message.	Sx,Fy-->		
			Equipment detects an unrecognized device ID within the message from the host.
		<--S9,F1	Equipment reports to the host that an "unrecognized device ID" was detected in the received message.

Message Fault Due to Unrecognized Stream Type:

COMMENTS	HOST	EQUIPMENT	COMMENTS
Host sends a message.	Sx,Fy-->		
			Equipment detects an unrecognized stream type within the message from the host.
		<--S9,F3	Equipment reports to the host that an “unrecognized stream type” was detected in the received message.

Message Fault Due to Unrecognized Function Type:

COMMENTS	HOST	EQUIPMENT	COMMENTS
Host sends a message.	Sx,Fy-->		
			Equipment detects an unrecognized function type within the message from the host.
		<--S9,F5	Equipment reports to the host that an “unrecognized function type” was detected in the received message.

Message Fault Due to Illegal Data Format:

COMMENTS	HOST	EQUIPMENT	COMMENTS
Host sends a message.	Sx,Fy-->		
			Equipment detects illegal data format within the message from the host.
		<--S9,F7	Equipment reports to the host that “illegal data format” was detected in the received message.

Communication Fault Due to Transaction Timer Timeout:

COMMENTS	HOST	EQUIPMENT	COMMENTS
			Equipment does not receive an expected reply message from the host and a transaction timer timeout occurs.
		<--S9,F9	Equipment reports to the host that a transaction timer timeout occurred.

Message fault due to data too long:

```
COMMENTS                         HOST   EQUIPMENT                       COMMENTS

Host sends a message.            Sx,Fy-->
                                                     Equipment detects that the
                                                     message from the host contains
                                                     more data than it can handle.
                                        <--S9,F11    Equipment reports to the host
                                                     that "data too long" was
                                                     detected in the received
                                                     message.
```

Communication Fault Due to Conversation Timeout:

```
COMMENTS                         HOST   EQUIPMENT                       COMMENTS

Host sends a message.            Sx,Fy-->
                                        <--Sx,Fy+1   Equipment sends reply.
                                                     Equipment is now expecting a
                                                     specific message from the host
                                                     as a result of the previous
                                                     transaction. Equipment has not
                                                     received the expected message
                                                     from the host and a
                                                     conversation timeout occurs.
                                        <--S9,F13    Equipment reports to the host
                                                     that a conversation timeout
                                                     occurred.
```

4.10 *Clock* — The clock capability enables host management of time-related activities and occurrences associated with the equipment and across multiple pieces of equipment.

4.10.1 *Purpose* — The purpose of the clock capability is to enable time stamping of collection event and alarm reports. Time stamping is useful for resolving relative order of event/alarm occurrences and scheduling of equipment activities by the host.

The ability for the host to instruct the equipment to set an internal clock to a specified time value, and for the equipment to request the current date and time, is needed for effective time management and synchronization between host and equipment.

4.10.2 *Definitions*

Clock — Clock is a status variable containing the current value of time at the equipment. Clock may be included in report definitions and/or queried separately by the host. See SEMI E5 for a full definition of this variable data item and its required formatting.

TIME — TIME is a data item contained in messages used by the host to set time at the equipment and by the equipment or host to request the current time from the other. (See SEMI E5 for a full definition of this data item.)

4.10.3 *Detailed Description* — The clock capability assumes the existence of a relative time reference on the equipment. This time reference is used as a basis for updating the time value of an equipment status variable called "Clock." The time reference must reflect the current time to within a resolution range of seconds to centiseconds (refer to the format for Clock in the SEMI E5 Standard). The purpose of time stamping with centiseconds is to resolve the order in which nearly simultaneous events occur rather than to provide a more precise record of the time of day at which they occurred. Where more than one event occurs within a given period of clock resolution, the centiseconds reported in the time stamps for these events must reflect the actual order in which the events were detected. Equipment with a clock resolution of less than a second should report centiseconds. Otherwise, centiseconds should be assigned to reflect the relative order in which events were detected. Equipment unable to resolve time to less than a second and unable to reflect the relative order in which events were detected may report centiseconds as "00".

The host employs the "Date and Time Set Request" message (S2,F31) to initialize the value of Clock to the value contained in the TIME data item. Similarly, the equipment may employ the "Date and Time Request" message (S2,F17) to obtain a new initialization time for Clock. As before, the value of TIME returned by the host is used to set Clock. Note, in the event that the precision of TIME is seconds and that for Clock is centiseconds, in both cases the initial value of Clock shall contain "00" for its Centisecond digits upon initialization. Additionally, for any field in TIME that is not supported by the equipment, the local value of this field is equipment dependent. For example, Equipment that cannot resolve time to less than a second might round or ignore centiseconds and always set the Centisecond field to "00".

4.10.4 *Requirements*

— The resolution and update rate of the internal time reference must be sufficient to distinguish between two nearly simultaneous collection events and/or alarms.

— The equipment supplier shall provide documentation describing the resolution of the internal time reference.

— The equipment supplier shall provide documentation describing how centisecond values are assigned, including the case of unresolvable simultaneous events.

4.10.5 *Scenarios*

Equipment Requests TIME (Optional Scenario):

COMMENTS	HOST	EQUIPMENT	COMMENTS
		<--S2,F17	Equipment requests a time value from the host.
Host responds with a TIME value	S2,F18-->		
			Equipment sets its internal time reference to the value of TIME received from the host.

Host Instructs Equipment to Set Time:

COMMENTS	HOST	EQUIPMENT	COMMENTS
Host instructs equipment to set its time.	S2,F31-->		
		<--S2,F32	Equipment sets its internal time reference to the value of TIME received from the host and acknowledges completion.

Host Requests Equipment's Current Time Value:

COMMENTS	HOST	EQUIPMENT	COMMENTS
Host requests equipment time.	S2,F17-->		
		<--S2,F18	Equipment returns its internal time reference value to the host.

4.11 *Spooling* — Spooling is a capability whereby the equipment can queue messages intended for the host during times of communication failure and subsequently deliver these messages when communication is restored. Spooling is limited to primary messages of user-selected streams.

4.11.1 *Purpose* — The purpose of spooling is to provide a method for retaining equipment message data that might otherwise be lost due to communication failure. The motive for producing this functionality is to retain valuable data used to track material and to improve product quality. The spooling capability fills a gap in the SEMI E5 standard. In the past, without a spooling capability, the equipment has typically discarded messages that could not be delivered, or turned messaging off altogether. It is intended that the host initiate the spool unload process immediately following the reestablishment of communications.

4.11.2 *Definitions*

MaxSpoolTransmit — An equipment constant containing the maximum number of messages that the equipment shall transmit from the spool in response to an S6,F23 “Transmit Spooled Messages” request. If MaxSpoolTransmit is set to zero, no limit is placed on the messages sent from the spool. Multi-block inquire/grant messages are not counted in this total.

OverWriteSpool — A boolean equipment constant used to indicate to the equipment whether to overwrite data in the spool area or to discard further messages whenever the spool area limits are exceeded.

Send Queue — Refers to the queue into which equipment generated SECS messages are placed in preparation for transmission to the host.

Spool — The spool is an area of non-volatile storage in which the equipment stores certain messages that cannot be delivered to the host (when the equipment is in the NOT COMMUNICATING substate of COMMUNICATIONS ENABLED). The spool area can be thought of as a sequential “ring” buffer. The term spool is also used to denote the action of placing messages into the spool area.

SpoolCountActual — A status variable used to keep a count of the messages actually stored in the equipment’s spool area. Multi-block inquire/grant messages are not spooled and not included in this count.

SpoolCountTotal — A status variable used to keep a count of the total number of primary messages directed to the spool, regardless of whether placed or currently retained in the spool. Multi-block inquire/grant messages are not spooled and not included in this count.

SpoolFullTime — A status variable containing the timestamp when the spool last became full. If the spool was not filled during the last spooling period, this will contain a time value prior to the current SpoolStartTime.

SpoolStartTime — A status variable containing the timestamp from when spooling was last activated.

4.11.3 *Description*

4.11.3.1 *Spooling State Model* — There are two major states of spooling: SPOOL INACTIVE and SPOOL ACTIVE. SPOOL ACTIVE has two components: SPOOL UNLOAD and SPOOL LOAD. These are each broken into substates. A description of all spooling states, substates, and applicable state transitions follows. The POWER OFF and POWER ON parent states are common to all equipment subsystems. They are shown here to illustrate the retention of spooling context during a power down situation.

NOTE 20: Disabling SECS communications does not affect the current spooling state since no messages are generated until communications are subsequently enabled. Spooling is effectively frozen in this case.

POWER OFF

The equipment has lost power for any reason (e.g., power failure, power switch set to off).

POWER ON

The equipment is powered up.

SPOOL INACTIVE

This is the normal operating mode. No spooling occurs. The spool area is empty. Primary SECS-II messages are transmitted normally.

SPOOL ACTIVE

All primary SECS-II messages ready for sending and for which spooling is enabled (see S2,F43) are directed to the spool area. All other primary messages, except Stream 1, are discarded. The equipment shall attempt to send any secondary messages that are generated and discard these messages should the attempt to send fail.

Spool state transitions from SPOOL INACTIVE to SPOOL ACTIVE if the communications state changes from COMMUNICATING to NOT COMMUNICATING (Communication State Transition Table, #14) or from WAIT CRA to WAIT DELAY (Communication STATE Transition Table, #6).

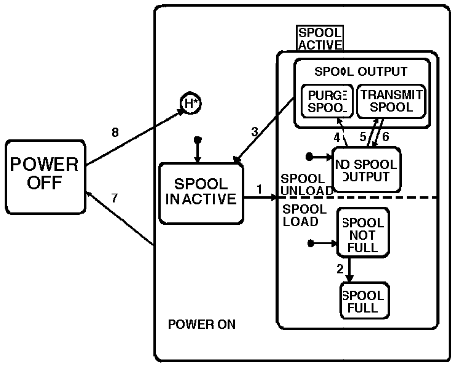


Figure 4.11
Spooling State Diagram

Once communications are established, the host must initiate the spool unload sequence to restore full functionality (see below). Since the equipment will deliver secondary messages, the host may inquire for information or send commands as needed.

The SPOOL ACTIVE state has two AND states: SPOOL LOAD and SPOOL UNLOAD. This means that they operate independently, though sharing data and some state change stimuli. See Section 3.1 for explanation of state model terms and notation.

SPOOL LOAD

The SPOOL LOAD component enters messages into the spool area. It is divided into two substates: SPOOL NOT FULL and SPOOL FULL. SPOOL NOT FULL is the default entry substate of the parent state SPOOL LOAD.

SPOOL NOT FULL

As primary SECS-II messages are directed to the spool area, the equipment shall "write" the SECS-II message to the end of the spool. Status variables SpoolCountTotal and SpoolCountActual shall be incremented each time a message is placed in the spool area.

SPOOL FULL

In this state, all of the allocated spooling area is filled. Choice of the following options shall be controlled by the setting of a Boolean equipment constant called "OverWriteSpool." The first message to be dealt with is that which could not be fit into the spool prior to transition from SPOOL NOT FULL (see transition table below).

OverWriteSpool is True: The equipment deletes as many of the "oldest" records (e.g., SECS-II messages) contained in the spool area necessary to make space for the new message and then adds the message. Status variable SpoolCountTotal shall be incremented whenever a message is submitted to the spool area. Status variable SpoolCountActual shall be manipulated to keep an accurate count of the number of messages contained in the spool area. For example, if it is necessary to delete three messages in the spool area to spool one message, SpoolCountActual would have three subtracted and then one added to the total.

OverWriteSpool is False: Any subsequent primary messages shall be discarded. When such a message is discarded, SpoolCountTotal is incremented.

SPOOL UNLOAD

The SPOOL UNLOAD component of SPOOLACTIVE deals with movement of messages out of the spool. It has an active substate (SPOOL OUTPUT) and a passive substate (NO SPOOL OUTPUT). NO SPOOL OUTPUT is the default entry substate, since the equipment is NOT COMMUNICATING at the time spooling is initiated. When communications between equipment and host are restored, there is an opportunity for the host to recover spooled messages. No action is taken until the host initiates the spool output process via the S6,F23 (Request Spooled Data). The host has the option to either receive the spooled messages (see substate TRANSMIT SPOOL) or discard all messages in the spool (see substate PURGE SPOOL).

NO SPOOL OUTPUT

In this state, no messages are removed from the spool.

SPOOL OUTPUT

The SPOOL OUTPUT state encompasses the removal of messages from the spool. Its substates are TRANS-MIT SPOOL and PURGE SPOOL.

TRANSMIT SPOOL

The host elects to receive all messages contained in the spool area. The equipment is expected to keep track of the oldest record (i.e., message) within the spool area. When communications are re-established with the host and transmission of the spool area is started, the oldest record must be the first record transmitted, then the next oldest record, etc. There is no prioritization of messages to be sent from the spool.

As each spooled message is successfully transmitted to the host, it is removed from the spool area upon successful completion of the transaction. Spool-CountActual is decremented as each message is removed from the spool. The equipment shall transmit messages only from the spool area until all spooled messages have been completely transmitted to the host.

Flow control of the spool transmit process is achieved in two ways. First, only one open transaction on the equipment is allowed during spool unload. Thus, if a message requires a reply, the equipment shall wait for that reply before transmitting the next spooled message. Messages which require no reply may be transmitted sequentially as rapidly as the message transfer mechanism will allow.

The second flow control method is to allow the host to limit the maximum number of messages sent from the spool in response to the S6,F23 request. An equipment constant named MaxSpoolTransmit may be set by the host to achieve this behavior. If MaxSpoolTransmit is set to five, for example, the equipment will send the first five messages from the spool and then transition to the NO SPOOL OUTPUT state, awaiting the next S6,F23 request. There is no event report generated when MaxSpoolTransmit is reached. The host is responsible for determining this situation by a) counting the messages received, b) timing out waiting for the next message, c) inquiring to the equipment for the curent value of the SpoolCountActual status variable, or d) some combination of the above. If MaxSpool-Transmit is set to zero, the spool shall be transmitted completely in response to S6,F23.

Normal spooling continues during the spool transmit process. If the SPOOL LOAD component should transition to SPOOL FULL, it shall not have any effect on the SPOOL UNLOAD component. Once full, the spool cannot make the transition back to SPOOL NOT FULL except via the SPOOL INACTIVE state. Space made available due to the spool unload process shall not be used in this case.

When a multi-block message is to be transmitted from the spool, any required inquire/grant transaction shall be initiated. If the host's response denies permission to send the multi-block message, the equipment shall discard that message and continue with the transmit process. This sequence shall count as one message in the MaxSpoolTransmit count.

There is one area where SPOOL LOAD and SPOOL UNLOAD may interact: When the spool is full and OverWriteSpool is True. During the spool transmit process, spooled messages are being removed and new primary messages are being written to the spool. These new messages are overwriting the oldest messages available, unless the unload process has freed sufficient spool space. There is a possibility that the unload and overwrite processes may compete for control of the same message area. For example, if the spool holds messages ABCDE, with A oldest and E newest, A might be sent to the host, B (and the space from A) overwritten by the new message F, C sent to the host, D and E (and the space from message C) overwritten by G, etc. The loss of continuity may be "disorienting" to the host program receiving the messages. It is expected that the unload process will be fast relative to the generation of new messages, so that this occurrence will be rare.

Should a communication failure occur during the spool transmit process, spooling shall continue as before the transmit process began. However, the spool unload sequence shall terminate (i.e., transition to NO SPOOL OUTPUT will occur — see transition table below).

PURGE SPOOL

The equipment shall discard all messages in the spool and, when the spool is empty, zero SpoolCountActual.

Spooling State Transitions

A table follows detailing all spooling state transitions as presented in the state transition diagram.

Table 4.11 Spooling State Transition

#	*Current State*	*Trigger*	*New State*	*Action*	*Comment*
1	SPOOL INACTIVE	Communication state changes from COMMUNICATING to NOT COMMUNICATING or from WAIT CRA to WAIT DELAY and Enable Spool is true.	SPOOL ACTIVE	SpoolCountActual and SpoolCountTotal are initialized to zero. Any open transactions with the host are aborted. SpoolStartTime (SV) is set to current time. Alert the operator that spooling is active.	The default state in each AND substate is entered. The message which could not be sent remains in the send queue and is dealt with in Spool Active state. The collection event Spooling Activated has occurred.
2	SPOOL NOT FULL	Message generate which will not fit into spool area.	SPOOL FULL	SpoolFullTime (SV) is set to current time. Alert the operator that spool is full.	The message which would not fit into the spooling area is dealt with after the transition. No collection event is generated.
3	SPOOL OUTPUT	Spool area emptied.	SPOOL INACTIVE	Spooling process disabled. Alert the operator that spooling has been terminated.	The collection event Spooling Deactivated has occurred. Transition from the AND substate Spool Unload component occurs.
4	NO SPOOL OUTPUT	S6,F23 received w/RSDC = 1.	PURGE SPOOL	No action.	Initiates purging process. No collection event is generated since this is based on host request.
5	NO SPOOL OUTPUT	S6,F23 received w/RSDC = 0.	TRANSMIT SPOOL	No action.	Initiates message transmission from spool. No collection event is generated since this is based on host request.
6	TRANSMIT SPOOL	Communication failure or MaxSpoolTransmit reached.	NO SPOOL OUTPUT	Spool transmission process suspended.	If communications failure, the event Spool Transmit Failure has occurred. No collection event generated for MaxSpoolTransmit reached.
7	POWER ON	Equipment power source discontinued.	POWER OFF	No action.	Spooling context has been maintained in non-volatile storage prior to this transition.
8	POWER OFF	Equipment power source restored.	POWER ON	Spooling context restored from non-volatile memory.	If spooling were active prior to power down, it shall continue. If TRANSMIT SPOOL were active at powerdown, transition #6 is expected to follow since communications state is initially NOT COMMUNICATING.

4.11.3.2 *Enabling Spooling* — The equipment shall provide the host with the ability to enable and disable Spooling for any message (except Stream 1 messages (i.e., S1,F1, S1,F13)) via the S2,F43/F44 transaction. Spooling may be enabled for an entire Stream, for individual messages within a stream, or for any combination of the two. Streams and Functions not referenced in this message are not spooled. Spooling can be totally disabled by sending an S2,F43 with a zero length list for the first item (see S2,F43 definition). In addition, the equipment shall provide an equipment constant, EnableSpooling, to allow setting the enable or disable of spooling. NOTE: When EnableSpooling is false, the SPOOL state cannot transition from SPOOLINACTIVE to SPOOLACTIVE. Changing EnableSpooling does not change the spool state, purge the spool or change streams and functions enabled for spooling. Once the equipment is “SPOOL ACTIVE”, the host must initiate the spool unload sequence to restore full functionality even though “Enable Spooling” has changed to false.

4.11.4 *Requirements* — The following items are required to support the spooling capability:

— At a minimum, the equipment shall reserve for spooling non-volatile storage with sufficient capacity to store all of the primary SECS-II messages that would occur during a normal processing cycle.

— While spooling is enabled, the equipment shall reply to primary messages sent by the host with the appropriate secondary message.

— Secondary messages which cannot be delivered shall be discarded, never spooled.

— All spooling-related status variables and setup information (as per S2,F43) must be stored in non-volatile memory along with any other information required for the potential unloading of the spool area after a power loss.

— Upon powerup, the equipment shall retain all spooling context from the time the equipment was last shutdown or reset. This means that spooling, if previously active, continues upon system powerup.

— The Equipment must reject any message that attempts to set “Spooling” for Stream 1.

— If a multi-block primary message need for the inquire/grant is to be sent during SPOOL ACTIVE, the message should be placed in the spool and the grant resolved during spool transmit.

4.11.5 *Scenarios*

Define the Set of Messages to be Spooled:

This Scenario is used to set up the list of messages that the equipment should spool (or by defining none, to disable spooling).

COMMENTS	HOST	EQUIPMENT	COMMENTS
Host defines messages to be spooled in case of communications failure.	S2,F43-->		
		<--S2,F44	Equipment acknowledges setup.

Define the Maximum Number of Messages to Send in Response to S6,F23:

This Scenario sets the value of the equipment constant MaxSpoolTransmit.

COMMENTS	HOST	EQUIPMENT	COMMENTS
Host sends value for equipment constant MaxSpoolTransmit.	S2,F15-->		
		<--S2,F16	Equipment acknowledges equipment constant change.

Request or Delete Spooled Data (MaxSpoolTransmit = 0):

This Scenario is used to initiate the transfer of the spooled data from the equipment to the host or to purge the spool.

```
COMMENTS                         HOST   EQUIPMENT                          COMMENTS

                Communications were lost and then re-established.
Host requests data that
includes spool-related
status variables.                 S1,F3-->
                                         <--S1,F4     Send status data.
NOTE: S1,F3 is one of
various methods that
could be used. Request
or delete spooled data.          S6,F23-->
                                         <--S6,F24    Request spooled data
                                                      acknowledgement.
                                                      [IF] RSDC = 0 (Spool data
                                                      requested.)
                                                      [THEN]
                                                      The appropriate Streams and
                                                      Functions are used to transmit
                                                      the spooled data to the host.
                                                      [ELSE_IF] RSDC = 1
                                                      [THEN]
                                                      Spool data discarded.
                                                      [END_IF]
                                         <--S6,F11    Spooling Deactivated event
                                                      report sent.
Acknowledge                      S6,F12-->
```

Request or Delete Spooled Data (MaxSpoolTransmit > 0):

This Scenario shows the affect of MaxSpoolTransmit < SpoolCountActual on the Spool Transmit process. For the purpose of illustration, the value of MaxSpoolTransmit is 5 and the SpoolCountActual is 8 (at the time communications are re-established). No messages are added to the Spool during the transmit process.

```
COMMENTS                          HOST   EQUIPMENT                          COMMENTS

                Communications were lost and then re-established.
Host requests data that
includes spool-related
status variables.                 S1,F3-->
                                         <--S1,F4     Send status data (e.g.
                                                      SpoolCountActual = 8, MaxSpool-
                                                      Transmit = 5).
Host requests spooled data
(RSDC=0).                        S6,F23-->
                                         <--S6,F24    Request spooled data
                                                      acknowledgement.
                                                      The five oldest messages in the
                                                      Spool are transmitted to the
                                                      host. Spooling remains active.
Host recognizes that MaxSpoolTransmit
is reached.
Host requests additional         S6,F23-->
spooled data (RSDC = 0).
                                         <--S6,F24    Request spooled data
                                                      acknowledgement.
                                                      The three remaining messages
                                                      are transmitted from the spool.
                                         <--S6,F11    Spooling Deactivated event
                                                      report sent.
Acknowledge                      S6,F12-->
```

4.12 *Control* — The control-related capabilities allow for configuration and manipulation of the control state model. In this way the host and/or user may modify the equipment's control-related behavior.

4.12.1 *Purpose* — This section complements the CONTROL state model description found in Section 3.3. It defines the requirements for implementation of this model.

4.12.2 *Definitions* — None.

4.12.3 *Description*

4.12.3.1 *Control Configuration* — The control state model has two areas of configuration. The first area is related to default entry states of the state model. Upon system initialization, the system must activate either the ON-LINE or OFF-LINE state. Upon entry to OFF-LINE, the system must in turn activate one of the substates of OFF-LINE (EQUIPMENT OFF-LINE, ATTEMPT ON-LINE, or HOST OFF-LINE). In both these cases, the user shall configure the equipment to make the choices appropriate to that factory. Entry to the ON-LINE state also involves a choice of substates. In this case, the equipment reads the front panel REMOTE/LOCAL switch to determine the appropriate state.

The second area of configuration involves the transition to be made if the ON-LINE attempt should fail. The model may be set to transition to either HOST OFF-LINE or to EQUIPMENT OFF-LINE should the S1,F1 transaction be terminated unsuccessfully. Choosing HOST OFF-LINE allows the host to cause the equipment to transition to ON-LINE when the host becomes ready. This is accomplished via the message S1,F17 (see below).

4.12.3.2 *Changing Control State* — In the control state model, both the operator and the host can affect the control state. The operator retains ultimate authority to set the equipment OFF-LINE by means of an OFF-LINE switch mechanism. The operator also can cause the equipment to attempt to go ON-LINE. Under some circumstances, the host can initiate the transition to ON-LINE.

If the operator requests ON-LINE, the equipment will send an S1,F1 to the host. The host may confirm ON-

LINE with an S1,F2 or deny ON-LINE by sending an S1,F0.[23]

When the equipment is ON-LINE, the host may request that it transition to OFF-LINE. It will transition into the HOST OFF-LINE substate. When the equipment HOST OFF-LINE state is active, the host may request that it transition to ON-LINE. The combination of these two allow the host to cycle the equipment between ON-LINE and OFF-LINE.

Only the operator may change the ON-LINE substate (REMOTE or LOCAL).

4.12.4 *Requirements*

— The equipment shall supply a method for configuring the default CONTROL state to be activated upon system initialization. The choice of states must be among ATTEMPT ON-LINE, EQUIPMENT OFF-LINE, HOST OFF-LINE, and ON-LINE.

— The equipment shall supply a method for configuring which state should be activated when the attempt to go ON-LINE fails. The option is a transition to either the HOST OFF-LINE state or the EQUIPMENT OFF-LINE state.

— The equipment shall supply a momentary switch which will initiate the transition to OFF-LINE and another which will begin the process to go ON-LINE. Discrete position switches shall not be used. These should be designed so that they may not be actuated simultaneously. The switch may be mounted on the front panel or be available via keyboard input at the operator console.

— The equipment shall supply a discrete two-position switch which the operator may use to indicate the desired substate for ON-LINE (i.e., REMOTE or LOCAL). The switch may be mounted on the front panel or be available via keyboard input at the operator console. If implemented in software, this setting shall be retained in non-volatile storage.

— The equipment shall supply an indicator on the front panel which displays the full identification of the current CONTROL state/substate (e.g., OFF-LINE/ATTEMPT ON-LINE). This may be accomplished either with labelled display lights or via the operator console display. It is recommended that this indicator be visible at all times.

— The equipment shall supply a status variable that contains the current state/substate of the CONTROL state model.

— Whenever the ON-LINE/REMOTE state is active and the operator issues a command to the equipment, the equipment shall cause an "operator command issued" event.

23 If there is no host response (i.e. reply timeout), the equipment shall treat it as a denial.

4.12.5 *Scenarios*

4.12.5.1 *Operator-Initiated Scenarios*

Host Accepts ON-LINE:

COMMENTS	HOST	EQUIPMENT	COMMENTS
Operator actuates ON-LINE switch when equipment OFF-LINE state is active.			
		<--S1,F1	Equipment requests ON-LINE.
Host grants ON-LINE.	S1,F2-->		
		<--S6,F11	"Control State LOCAL (or REMOTE)" event.
Acknowledge.	S6,F12-->		

Host Denies ON-LINE:

COMMENTS	HOST	EQUIPMENT	COMMENTS
Operator actuates ON-LINE switch when equipment OFF-LINE state is active.			
		<--S1,F1	Equipment requests ON-LINE.
Host denies ON-LINE.	S1,F0-->		

Operator Sets OFF-LINE:

COMMENTS	HOST	EQUIPMENT	COMMENTS
Operator actuates OFF-LINE switch when equipment ON-LINE state is active.			
		<--S6,F11	"Equipment requests OFF-LINE" event.
Acknowledge.	S6,F12-->		

Operator Sets REMOTE:

COMMENTS	HOST	EQUIPMENT	COMMENTS
Operator sets switch from LOCAL to REMOTE.			
		<--S6,F11	"Control State REMOTE" event.
Acknowledge.	S6,F12-->		

Operator Sets LOCAL:

COMMENTS	HOST	EQUIPMENT	COMMENTS
Operator sets switch from REMOTE to LOCAL.			
		<--S6,F11	"Control State LOCAL" event.
Acknowledge.	S6,F12-->		

4.12.5.2 *Host-Initiated Scenarios*

Host Sets OFF-LINE:

COMMENTS	HOST	EQUIPMENT	COMMENTS
Host requests OFF-LINE.	S1,F15-->		
			[IF] Equipment is OFF-LINE [THEN]:
		<--S1,F0	Equipment does not process requests.
			[ELSE] Equipment ON-LINE
		<--S1,F16	Equipment acknowledges request and transitions to OFF-LINE.
		<--S6,F11	"Equipment OFF-LINE" event.
Acknowledge	S6,F12-->		
			[END_IF]

Host Sets ON-LINE:

COMMENTS	HOST	EQUIPMENT	COMMENTS
Host requests ON-LINE.	S1,F17-->		
			[IF] Equipment is HOST OFF-LINE state not active. [THEN]
		<--S1,F18	Equipment denies request (ONLACK = 1).
			[ELSE] Equipment HOST OFF-LINE state is active.
		<--S1,F18	Equipment acknowledges request (ONLACK != 1).
		<--S6,F11	"Control state LOCAL (or REMOTE)" event. (only if ONLACK = 0)
Acknowledge	S6,F12-->		
			[END_IF]

5 Data Items

The following sections specify which data items and variable data items are required.

Except for the specified format restrictions, all data items and variable data items follow the definitions contained in SEMI E5.

5.1 *Data Item Restrictions* — The following is a subset of the data items used by SECS-II messages specified in this standard. Each data item listed in this section is restricted in its SEMI E5-defined usage. Most are limited to a single format from their standard list of formats. Data items used by SECS-II messages contained in this document, but which have no restrictions are not duplicated here.

NOTE 21: One data item, ALCD, is restricted in other than its format. Take note of this restriction as described below.

NOTE 22: The equipment supplier shall document any restrictions on length or format of CPNAME. Suppliers shall document the behavior of spaces in a CPNAME. The maximum length of CPNAME shall be 40. This change became effective in September 1995.

ACKC7A	Format: 51
ALCD	Only bit 8 (alarm set/cleared) of the binary byte is used. Bits 1–7, denoting alarm category, are not used.
ALID	Format: 5()
CCODE	Format: 52
CEID	Format: 5()
CPNAME	Format: 20
DATAID	Format: 5()
DATALENGTH	Format: 5()
ECID	Format: 5()
LENGTH	Format: 5()
PPID	Format: 20
RCMD	Format: 20
REPGSZ	Format: 5()
RPTID	Format: 5()
SEQNUM	Format: 52
SMPLN	Format: 5()
SVID	Format: 5()
TEXT	Format: 20
TOTSMP	Format: 5()
TRID	Format: 5()
VID	Format: 5()

5.2 *Variable Item List* — The following variable data items from the Variable Item Dictionary in SEMI E5 are required. Format restrictions are noted.

DVVAL's:

AlarmID Format: 5()
EventLimit
LimitVariable
PPChangeName Format: 20
PPChangeStatus
PPError
RcpChangeName
RcpChangeStatus
TransitionType

ECV's:

EstablishCommunicationsTimeout
MaxSpoolTransmit
OverWriteSpool
TimeFormat

SV's:

AlarmsEnabled
AlarmsSet
Clock
ControlState
EventsEnabled
PPError
PPExecName Format: 0,20
PPFormat
PreviousProcessState
ProcessState
RcpExecName
SpoolCountActual
SpoolCountTotal
SpoolFullTime
SpoolStartTime

6 Collection Events

Table 6.1 provides the list of collection events required to support the capabilities addressed within this standard. Also shown are typical variable data that would most likely be included in the associated collection event report and a reference to the event trigger and to the appropriate section of the standard.

This list does not represent all events that might be needed to properly monitor/control equipment. Many events are unique to the specific equipment characteristics. The needed additions are a matter for other standards and for collaboration between equipment supplier and user.

See Section 5.2 for further detail of variable data items.

Table 6.1 GEM-Defined Collection Events

Event Designation	*Typical Variable Data*	*Reference*
Control-Related Events:		Section 3.3
Equipment OFF-LINE	ControlState, Clock	ON-LINE->OFF-LINE
Control State LOCAL	ControlState, Clock	REMOTE->LOCAL or OFF-LINE->LOCAL
Control State REMOTE	ControlState, Clock	LOCAL->REMOTE or OFF-LINE->REMOTE
Operator Command Issued	OperatorCommand	Operator Activity while REMOTE state is active.
Processing-Related Events:	Note: Any transition in the implemented processing state model must have a corresponding collection event.	Section 3.4
Processing Started	Clock, PreviousProcessState	Entry into EXECUTING state.
Processing Completed	Clock, PreviousProcessState	Normal exit of EXECUTING state.
Processing Stopped	Clock, PreviousProcessState	Result of STOP command from host or operator.
Processing State Change	Clock, ProcessState, PreviousProcessState	Any processing state transition.
Alarm Management Events:		Section 4.3
Alarm$_n$Detected	Clock, AlarmID, AlarmsSet, Associated variable data	ALARM$_n$CLEAR->ALARM$_n$SET
Alarm$_n$Cleared	Clock, AlarmID, AlarmsSet	ALARM$_n$SET->ALARM$_n$CLEAR
Equipment Constant Events:		Section 4.5
Operator Equipment Constant Change	ECID	Operator activity
Limits Monitoring:		Section 4.2.4
Limit Zone Transition$_n$ (separate CEID per variable)	Clock, LimitVariable, EventLimit, Transition Type	Entry into BELOW LIMIT or ABOVE LIMIT states.
Process Program Management Events:		Section 4.6
Process Program Change	PPChangeName, PPChangeStatus	Operator activity
Process Program(s) Selected	PPExecName	Operator/Host activity
Material Movement Events:		Section 4.7
Material Received	Clock	
Material Removed	Clock	
Spooling Events:		Section 4.11
Spooling Activated	SpoolStartTime	SPOOL INACTIVE->SPOOL ACTIVE
Spooling Deactivated	SpoolCountTotal	SPOOL OUTPUT->SPOOL INACTIVE
Spool Transmit Failure	Clock, SpoolCountActual SpoolCountTotal	TRANSMIT SPOOL->NO SPOOL OUTPUT
Terminal Services Events:		Section 4.8
Message Recognition	Clock	Operator
New Execution Recipe Event	RcpChangeName, RcpChangeStatus	Section 4.6.2.2
Execution Recipe Change Event	RcpChangeName, RcpChangeStatus	Section 4.6.2.2

7 SECS-II Message Subset

This section lists the required set of SECS-II messages as referenced in this document. Definitions for these messages can be found in SEMI E5. All primary messages (for which SEMI E5 defines replies) should have replies available. Replies are required or optional as specified in SEMI E5.

STREAM 1: Equipment Status

S1,F1 Are You There Request (R)	S,H<->E
S1,F2 On-Line Data (D)	S,H<->E
S1,F3 Selected Equipment Status Request (SSR)	S,H->E
S1,F4 Selected Equipment Status Data (SSD)	M,H<-E
S1,F11 Status Variable Namelist Request (SVNR)	S,H->E
S1,F12 Status Variable Namelist Reply (SVNRR)	M,H<-E
S1,F13 Establish Communications Request (CR)	S,H<->E
S1,F14 Establish Communications Request Acknowledge (CRA)	S,H<->E
S1,F15 Request OFF-LINE (ROFL)	S,H->E,reply
S1,F16 OFF-LINE Acknowledge (OFLA)	S,H<-E
S1,F17 Request ON-LINE (RONL)	S,H->E,reply
S1,F18 ON-LINE Acknowledge (ONLA)	S,H<-E

STREAM 2: Equipment Control and Diagnostics

S2,F13 Equipment Constant Request (ECR)	S,H->E
S2,F14 Equipment Constant Data (ECD)	M,H<-E
S2,F15 New Equipment Constant Send (ECS)	S,H->E
S2,F16 New Equipment Constant Acknowledge (ECA)	S,H<-E
S2,F17 Date and Time Request (DTR)	S,H<->E
S2,F18 Date and Time Data (DTD)	S,H<->E
S2,F23 Trace Initialize Send (TIS)	S,H->E
S2,F24 Trace Initialize Acknowledge (TIA)	S,H<-E
S2,F29 Equipment Constant Namelist Request (ECNR)	S,H->E
S2,F30 Equipment Constant Namelist (ECN)	M,H<-E
S2,F31 Date and Time Send (DTS)	S,H->E
S2,F32 Date and Time Acknowledge (DTA)	S,H<-E
S2,F33 Define Report (DR)	M,H->E
S2,F34 Define-Report Acknowledge (DRA)	S,H<-E
S2,F35 Link Event Report (LER)	M,H->E
S2,F36 Link Event Report Acknowledge (LERA)	S,H<-E
S2,F37 Enable/Disable Event Report (EDER)	S,H->E,reply

S2,F38 Enable/Disable Event Report Acknowledge (EDEA)	S,H<-E
S2,F39 Multi-Block Inquire (DMBI)	S,H->E
S2,F40 Multi-Block Grant (DMBG)	S,H<-E
S2,F41 Host Command Send (HCS)	S,H->E
S2,F42 Host Command Acknowledge (HCA)	S,H<-E
S2,F43 Reset Spooling Streams and Functions (RSSF)	S,H->E
S2,F44 Reset Spooling Acknowledge (RSA)	M,H<-E
S2,F45 Define Variable Limit Attributes (DVLA)	M,H->E
S2,F46 Variable Limit Attribute Acknowledge (VLAA)	M,H<-E
S2,F47 Variable Limit Attribute Request (VLAR)	S,H->E
S2,F48 Variable Limit Attributes Send (VLAS)	M,H<-E
S2,F49 Enhanced Remote Command	M,H->E
S2,F50 Enhanced Remote Command Acknowledge	M,H<-E

STREAM 5: Exception (Alarm) Reporting

S5,F1 Alarm Report Send (ARS)	S,H<-E
S5,F2 Alarm Report Acknowledge (ARA)	S,H->E
S5,F3 Enable/Disable Alarm Send (EAS)	S,H->E
S5,F4 Enable/Disable Alarm Acknowledge (EAA)	S,H<-E
S5,F5 List Alarms Request (LAR)	S,H->E
S5,F6 List Alarm Data (LAD)	M,H<-E

STREAM 6: Data Collection

S6,F1 Trace Data Send (TDS)	S,H<-E
S6,F2 Trace Data Acknowledge (TDA)	S,H->E
S6,F5 Multi-block Data Send Inquire (MBI)	S,H<-E
S6,F6 Multi-block Grant (MBG)	S,H->E
S6,F11 Event Report Send (ERS)	M,H<-E
S6,F12 Event Report Acknowledge (ERA)	S,H->E
S6,F15 Event Report Request (ERR)	S,H->E
S6,F16 Event Report Data (ERD)	M,H<-E
S6,F19 Individual Report Request (IRR)	S,H->E
S6,F20 Individual Report Data (IRD)	M,H<-E
S6,F23 Request Spooled Data (RSD)	S,H->E
S6,F24 Request Spooled Data Acknowledgement Send (RSDAS)	S,H<-E

STREAM 7: Process Program Load

S7,F1 Process Program Load Inquire (PPI)	S,H<->E, reply
S7,F2 Process Program Load Grant (PPG)	S,H<->E
S7,F3 Process Program Send (PPS)	M,H<->E
S7,F4 Process Program Acknowledge (PPA)	S,H<->E
S7,F5 Process Program Request (PPR)	S,H<->E
S7,F6 Process Program Data (PPD)	M,H<->E
S7,F17 Delete Process Program Send (DPS)	S,H->E
S7,F18 Delete Process Program Acknowledge (DPA)	S,H<-E
S7,F19 Current EPPD Request (RER)	S,H->E
S7,F20 Current EPPD Data (RED)	M,H<-E
S7,F23 Formatted Process Program Send (FPS)	M,H<->E
S7,F24 Formatted Process Program Acknowledge (FPA)	S,H<->E
S7,F25 Formatted Process Program Request (FPR)	S,H<->E
S7,F26 Formatted Process Program Data (FPD)	M,H<->E
S7,F27 Process Program Verification Send (PVS)	S,H<-E
S7,F28 Process Program Verification Acknowledge (PVA)	S,H->E
S7,F29 Process Program Verification Inquire (PVA)	
S7,F30 Process Program Verification Grant (PVG)	

STREAM 9: System Errors

S9,F1 Unrecognized Device ID (UDN)	S,H<-E
S9,F3 Unrecognized Stream Type (USN)	S,H<-E
S9,F5 Unrecognized Function Type (UFN)	S,H<-E
S9,F7 Illegal Data (IDN)	S,H<-E
S9,F9 Transaction Timer Timeout (TTN)	S,H<-E
S9,F11 Data Too Long (DLN)	S,H<-E
S9,F13 Conversation Timeout (CTN)	S,H<-E

STREAM 10: Terminal Services

S10,F1 Terminal Request (TRN)	S,H<-E
S10,F2 Terminal Request Acknowledge (TRA)	S,H->E
S10,F3 Terminal Display, Single (VTN)	S,H->E
S10,F4 Terminal Display, Single Acknowledge (VTA)	S,H<-E
S10,F5 Terminal Display, Multi-block (VMN)	M,H->E
S10,F6 Terminal Display, Multi-block Acknowledge (VMA)	S,H<-E
S10,F7 Multi-block Not Allowed (MNN)	S,H<-E

STREAM 14: Object Services

S14,F1 GetAttr Request	S,H <-> E
S14,F2 GetAttr Data	M,H <-> E

STREAM 15: Recipe Management

S15,F1 Recipe Management Multi-block Inquire	S,H <-> E
S15,F2 Recipe Management Multi-block Grant	S,H <-> E
S15,F21 Recipe Action Request	M,H <-> E
S15,F22 Recipe Action Acknowledge	M,H <-> E
S15,F27 Recipe Download Request	M,H -> E
S15,F28 Recipe Download Acknowledge	M,H <- E
S15,F29 Recipe Verify Request	M,H -> E
S15,F30 Recipe Verify Data	M,H <- E
S15,F31 Recipe Upload Request	S,H -> E
S15,F32 Recipe Upload Data	M,H <- E
S15,F35 Recipe Delete Request	M,H -> E
S15,F36 Recipe Delete Acknowledge	M,H <- E

8 GEM Compliance

This section defines compliance to the GEM standard. It describes the fundamental GEM requirements and additional GEM capabilities. It provides references to other sections of the standard where detailed requirements are located. This section also defines standard terminology and documentation that can be used by equipment suppliers and device manufacturers to describe compliance with this standard.

The GEM standard contains two types of specifications:

— fundamental GEM requirements and

— requirements pertaining to additional GEM capabilities.

The fundamental GEM requirements form the foundation of the GEM standard. The additional GEM capabilities provide functionality required for some types of factory automation or functionality applicable to specific types of equipment. Figure 8.1 illustrates the relationship of the fundamental GEM requirements and the additional GEM capabilities.

8.1 *Fundamental GEM Requirements* — All equipment shall comply with the fundamental GEM requirements listed in Table 8.1. Compliance to these requirements involves precise and complete adherence to all sections of the GEM standard referenced in Table 8.1.

Table 8.1 Fundamental GEM Requirements

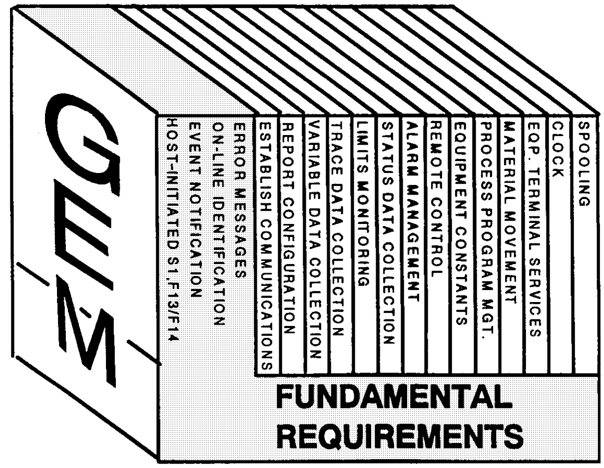
Requirement	*Section References*
State Models	3.0, 3.1, 3.3
Equipment Processing States	3.4
Host-Initiated S1,F13/F14 Scenario	4.1.5.1
Event Notification	4.2.1.1
On-line Identification	4.2.6
Error Messages	4.9
Control (Operator-Initiated)	4.12 (except 4.12.5.2)
Documentation	8.4

In addition, compliance requires adherence to the portions of the following sections that are applicable to the fundamental GEM requirements:

— Variable data items (GEM, Section 5)

— SECS-II data item restrictions (GEM, Section 5)

— Collection events (GEM, Section 6)



Vertical text represents capabilities.
Some capabilities are also fundamental requirements.

Figure 8.1
GEM Requirements and Capabilities

8.2 *GEM Capabilities* — The following table lists all GEM capabilities and the sections of the GEM standard where they are specified. These sections contain the detailed requirements for implementing a GEM capability. Requirements for an individual capability include any referenced portions of the document. As an example, the Alarm Management capability requires implementation of the status variables “AlarmsEnabled” and “AlarmsSet” as defined in Section 5.

Table 8.2 Section References for GEM Capabilities

Capability	*Section References*
Establish Communications	4.1, 3.2
Event Notification	4.2.1.1
Dynamic Event Report Configuration	4.2.1.2
Variable Data Collection	4.2.2
Trace Data Collection	4.2.3
Limits Monitoring	4.2.4
Status Data Collection	4.2.5
On-line Identification	4.2.6
Alarm Management	4.3
Remote Control	4.4
Equipment Constants	4.5
Process Program Management	4.6
Material Movement	4.7
Equipment Terminal Services	4.8
Error Messages	4.9
Clock	4.10

Capability	*Section References*
Spooling	4.11
Control (Operator-Initiated)	4.12 (except 4.12.5.1)
Control (Host-Initiated)	4.12.5.1

8.3 *Definition of GEM Compliance* — The term "GEM Compliance" is defined with respect to individual GEM capabilities to indicate adherence to the GEM standard for a specific capability. Equipment is GEM-compliant for a specific GEM capability if, and only if, the following three criteria are met:

— The fundamental GEM requirements are satisfied.

— The capability is implemented to conform with all applicable definitions, descriptions, and requirements defined for the capability in this standard.

— The equipment does not exhibit behavior related to this capability that conflicts with the GEM behavior defined for the capability.

For example, equipment that provides SECS-II messages for management of process programs must precisely implement the GEM Process Program Management capability to be "GEM-Compliant for Process Program Management."

Equipment may supply additional functionality not specified in the GEM standard by using any messages defined in the SECS-II standard as long as the additional functionality does not conflict with compliance to GEM capabilities.

Figure 8.2 illustrates the host view of equipment communications in relationship to the components of the GEM standard. The GEM capabilities are built upon the fundamental GEM requirements and present GEM-compliant behavior to the host when they are not obstructed by conflicting functionality. Additional non-GEM capabilities and non-obstructing extensions to GEM capabilities provide additional functionality while maintaining GEM behavior from the host view.

One additional term is defined to facilitate discussion of GEM capability. Equipment is "Fully GEM Capable" if and only if it meets the following two criteria:

— The equipment supplies all the GEM capabilities listed in Section 8.2.

— Every implemented GEM capability is GEM-Compliant.

8.4 *Documentation* — This section describes documentation requirements in addition to those specified in Sections 3 and 4 of this standard. All documentation of the SECS-II interface shall be supplied as a single volume, including Message Documentation, a Compliance Statement and the documentation required by Sections 3 and 4.

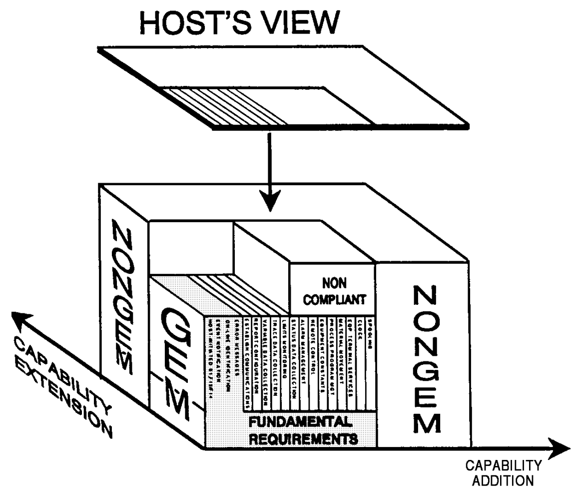


Figure 8.2
Host View of GEM

8.4.1 *Message Documentation* — The equipment supplier shall provide message documentation in conformance with Chapter 8 (Message Documentation) of SEMI E5.

8.4.2 *GEM Compliance Statement* — The SECS-II interface documentation provided by an equipment supplier shall address GEM compliance. This documentation shall include a GEM Compliance Statement that accurately indicates for each capability whether it has been implemented and whether it has been implemented in a GEM-compliant manner. The format for this statement is supplied as Table 8.3.

The table consists of three columns. The first column lists the requirements and capabilities. The other two columns pose questions to the supplier:

Implemented: Does the equipment provide functionality that is similar to that defined for the GEM requirement or capability?

GEM-Compliant: Has that requirement or capability been implemented in a GEM-compliant manner?

8.4.3 The equipment supplier may provide documentation on the format of required data items (see Section 5) using SECS Message Language Notation (SML). The SML formats are provided in Table 8.4.

Table 8.3 GEM Compliance Statement

GEM COMPLIANCE STATEMENT		
FUNDAMENTAL GEM REQUIREMENTS	**IMPLEMENTED**	**GEM-COMPLIANT**
State Models	❒ Yes ❒ No	❒ Yes (See NOTE 1.) ❒ No
Equipment Processing States	❒ Yes ❒ No	
Host-Initiated S1= F13/F14 Scenario	❒ Yes ❒ No	
Event Notification	❒ Yes ❒ No	
On-Line Identification	❒ Yes ❒ No	
Error Messages	❒ Yes ❒ No	
Documentation	❒ Yes ❒ No	
Control (Operator Initiated)	❒ Yes ❒ No	
ADDITIONAL CAPABILITIES	**IMPLEMENTED**	**GEM-COMPLIANT** (See NOTE 2.)
Establish Communications	❒ Yes ❒ No	❒ Yes ❒ No
Dynamic Event Report Configuration	❒ Yes ❒ No	❒ Yes ❒ No
Variable Data Collection	❒ Yes ❒ No	❒ Yes ❒ No
Trace Data Collection	❒ Yes ❒ No	❒ Yes ❒ No
Status Data Collection	❒ Yes ❒ No	❒ Yes ❒ No
Alarm Management	❒ Yes ❒ No	❒ Yes ❒ No
Remote Control	❒ Yes ❒ No	❒ Yes ❒ No
Equipment Constants	❒ Yes ❒ No	❒ Yes ❒ No
Process Program Management	❒ Yes ❒ No	❒ Yes ❒ No
Material Movement	❒ Yes ❒ No	❒ Yes ❒ No
Equipment Terminal Services	❒ Yes ❒ No	❒ Yes ❒ No
Clock	❒ Yes ❒ No	❒ Yes ❒ No
Limits Monitoring	❒ Yes ❒ No	❒ Yes ❒ No
Spooling	❒ Yes ❒ No	❒ Yes ❒ No
Control (Host-Initiated)	❒ Yes ❒ No	❒ Yes ❒ No

NOTE 1: Do not mark YES unless all fundamental GEM requirements are implemented and GEM-compliant.

NOTE 2: Additional capabilities may not be marked GEM-compliant unless the fundamental GEM requirements are GEM-compliant.

Table 8.4 SML Notation

Item Format	*SECS-II Format Code*		*SML Item Format Mnemonic*
	Binary	*Octal*	
LIST	000000	00	L [length]
Binary	001000	10	B
Boolean	001001	11	BOOLEAN
ASCII	010000	20	A [length] or A [min., max.]
JIS-8	010001	21	J [length] or J [min., max.]
8-byte integer (signed)	011000	30	I8
1-byte integer (signed)	011001	31	I1
2-byte integer (signed)	011010	32	I2
4-byte integer (signed)	011100	34	I4
8-byte floating point	100000	40	F8
4-byte floating point	100100	44	F4
8-byte integer (unsigned)	101000	50	U8
1-byte integer (unsigned)	101001	51	U1
2-byte integer (unsigned)	101010	52	U2
4-byte integer (unsigned)	101100	54	U4

A. Application Notes

NOTE: The material contained in these Application Notes is not an official part of this SEMI standard and is not intended to modify or supersede the official standard. Rather, these notes are auxiliary information describing possible methods for implementing the protocol described by the standard and are included as reference material. The standard should be referred to in all cases. SEMI makes no warranties or representations as to the suitability of the material set forth herein for any particular application. The determination of the suitability of the materialis solely the responsibility of the user.

A.1 Factory Operational Script

An Operational Script is a series of capabilities arranged in a typical factory operation sequence. The intent of having an Operational Script is to help put the SECS-II message Scenarios into a context. Although this context will vary, it represents a typical operational sequence found in most semiconductor device manufacturers' applications.

— System Initialization

— Synchronization

— Machine Setup

— Production Setup

— Processing

— Post-Processing

— Shutdown

The following script is not intended to be complete, but to serve as an example to be further developed on an implementation basis.

A1.1 *Anytime Capabilities* — All capabilities can generally occur at anytime during the operational script sequence.

A1.2 *System Initialization and Synchronization* — Upon system initialization, the default setting for communication (enabled or disabled) becomes effective, as well as any equipment constants or other information retained in non-volatile storage. The initial communication status is displayed at the equipment.

Assuming the communication state is enabled, the equipment will attempt to establish communication with a host computer. See Section 4.1 for a description of the scenario for establishing communications.

Upon receiving an indication that the equipment was previously not communicating, the host would typically perform synchronization activities including setting the equipment's clock and requesting selected status information. Note that synchronization activity is host application-dependent and may be implemented using various scenarios.

A1.3 Production Set-Up — The host typically has the following information:

— what material

— what process step

— what process program to use (PPID)

— current equipment status, VID's, SVID's

— data collection requirements (trace data & event data)

— VID's needed

— Equipment constants (ECID's)

Based upon the above information, the host will perform setup activities as required. It must be verified that the correct process program is available and selected at the equipment.

A1.3.1 *Auxiliary Material and Manual Set-Up* — Auxiliary material can be checked and verified at this point. If status variables exist for auxiliary material, they may be requested by the host.

Any other manual, non-process, and/or non-product specific set-up also may take place at this point. The operator may interact with the equipment and the host. If the operator interacts with the equipment, the equipment communications link with the host should stay operational.

The operator and the host may exchange information via equipment terminal services.

A1.3.2 *Product/Process Set-Up* — Specific product and/or process information is communicated to the equipment prior to processing material.

A1.3.3 *Material Load* — The host may instruct an operator or a material handling system to deliver material to the equipment.

Once the material has arrived at the equipment, the equipment or the operator will notify the host.

A1.3.4 *Production Data Collection Set-Up* — The host instructs the equipment to collect event-based data. Reports are defined and linked to events. Event reports can be enabled or disabled.

The host instructs the equipment to collect data from the equipment based on time intervals.

The host configures the equipment to monitor specific variables and to send event reports when variables transition between monitoring zones.

A1.4 *Processing*

A1.4.1 *Start Process Executing* — The host or operator issues a command to start.

A1.4.2 *Equipment Signals End of Run* — When process execution is completed, the equipment generates events. If any of the events are enabled, they will be sent as event reports.

A1.5 *Post-Processing* — The equipment has completed processing material. It now makes the material available to the operator or material handling system for removal. The equipment signals the host that it is available for more work.

A1.5.1 *Material Unload* — Material is unloaded from the equipment by an operator or material handling system.

A.2 Equipment Front Panel

In the GEM standard, several requirements are stated that involve the display or input of information at the equipment front panel. The "equipment front panel" refers to an area on the equipment that is available to the operator under normal use (i.e., without removing maintenance access panels). This may include a CRT display, keyboard, switches, and lights.

This application note provides some guidance for implementation of the GEM front panel capabilities. All of these requirements map directly to state models and capabilities defined in Sections 3 and 4. All capabilities may be implemented in either hardware (buttons, switches, lights) or in a software/CRT equivalent.

A2.1 *Displays and Indicators* — The intent of various displays is to inform the operator of either the current state of the equipment or of a recent change of state (or both). Therefore, it is most useful if these displays are continuously visible and easily recognized at a distance. Required displays/indicators include:

Communications State: This means that three distinct states must be represented: DISABLED, ENABLED/NOT COMMUNICATING, and ENABLED/COMMUNICATING.

Terminal Services: An "New Host Message" indicator must be supplied.

A2.2 *Switches/Buttons* — Note that discrete switches also contain information for the user. However, these tend to represent the desired states of the operator/user. The equipment's response to a change of a switch may not be instantaneous. Still, the current position of switches should be available to the operator.

It may be appropriate to limit the access to some switches and buttons. This might be done via any of the standard methods, keys, passwords, combinations, etc. This is especially true for system default switches that would not often be changed. Required switches/buttons include:

Communications State System Default: In what communications state should the equipment be when system initialization is complete? The choices are DISABLED and ENABLED.

Communications State Selector: This is a toggle or button that will initiate a transition from ENABLED to DISABLED or vice versa.

Message Recognition Button: This button is used to initiate an event message to the host which indicates that the "New Host Message" has been read. This button should function only when the New Host Message Indicator is activated and when the received message is displayed in the terminal display.

A.3 Examples of Equipment Alarms

Table A.3 provides alarm examples pertaining to various configurational aspects of equipment.

NOTE: It is important to stress that these are just examples intended to illustrate that alarms pertain to situations in which there exists a potential for exceeding physical safety limits associated with people, equipment, and material being processed as per the GEM definition of an alarm.

NOTE: The alarm capability is intended as an addition to standard safety alarms (e.g., lights, horns). There is no intent to replace direct operator reaction to such problems. Nor is there the expectation that the host can necessarily prevent or directly address such alarms.

An actual machine shall have an associated set of alarms defined by the manufacturer that pertains to its specific configuration and design. The equipment manufacturer is responsible for supplying documentation associated with these alarm definitions.

Table A.3 Alarm Examples Per Equipment Configuration

Subsystem	*Alarm Description*	*ALID*	*Trigger*	*Reset*	*Operator*	*Equipment*	*Material*
Mainframe Power Supply	Overvoltage		Voltage supply over maximum limit			X	
	Undervoltage		Voltage supply under minimum limit			X	
Internal Power Distribution Bus	AC Low		AC under minimum limit			X	X
Cooling System	Overtemp		Temperature over maximum		X		X
	Pressure Low		Pressure below minimum				X

Subsystem	The subsystem of the equipment to which the alarm is related
Alarm Description	Description of the alarm
ALID	The Alarm ID as specified by SECS-II
Trigger	Text description of what caused the alarm
Reset	Description of how to resolve the alarm condition
Affected	Who or what is affected by the alarm trigger: Operator, Equipment, and Material

A.4 Trace Data Collection Example

This example shows an implementation of the Trace Data Collection capability defined in Section 4.2.3.

S2,F23 sent by host:

```
   TRID = ABCD
   DSPER = 000100 (One minute per
period)
   TOTSMP = 9
   REPGSZ = 3
   SVID1 =  Temperature
   SVID2 =  Relative humidity
```

S6,F1 looks like this (starting at time 1 a.m.):

1st transmission <L,4>

```
     1. ABCD (trace ID)
     2. 3 (last sample of the
  transmission)
     3.    88   5   01   01   03   00
         Year Month Day  Hour Min  Sec
     4. <L, n> n = 2 SVID's x REPGSZ of
        3 = 2 x 3 = 6
        72 (temperature)
        0.29 (relative humidity)
        73 (temp.)
        0.30 (r.h.)
        71 (temp.)
        0.30 (r.h.)
```

2nd transmission <L,4>

```
     1. ABCD
     2. 6
     3. 88  05  01  01  06  00
                    hr  min
     4. <L,6>
           73
           0.31
           71
           0.32
           71
           0.31
```

3rd and last transmission <L,4>

```
1. ABCD
2. 9
3. 88  05  01  01  09  00
               hr  min
4. <L,6>
   71
   0.30
   72
   0.30
   71
   0.31
```

A.5 Harel Notation

Harel's statecharts extend traditional state-transition diagrams with several additional concepts, most important of which are hierarchy and concurrence. Statecharts depict the behavior of a system by showing states it may take, events that prompt a change of state, and the composition of states. What follows is a very brief description of the symbols defined for use and how these are useful to describe a system. See Figure A.5.1 for the basic notational symbols.

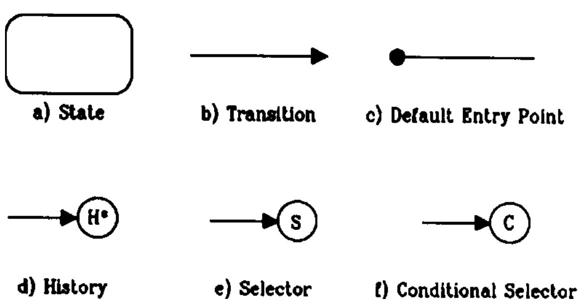


Figure A.5.1
Harel Statechart Symbols

States are represented by rounded boxes. A state transition is shown graphically with a line from the old state terminating with the arrow symbol at the new state. Transitions are unidirectional-while the reverse transition may be possible, it is considered a different transition with different conditions for initiation and different resultant actions.

States may be subdivided into substates to facilitate more concise definition of behavior. Thus, a hierarchy is defined whereby any state may be a substate of some parent state and in turn be the parent of its own substates. Substates must be one of two types, termed AND substates and OR substates.

A parent maybe divided into two or more OR substates of which one and only one is the active substate at any time. The accepted term for this exclusivity is XOR. Figure A.5.2 gives an example of a simple case of OR substates. In this example, some system (perhaps a motor) has a state named FUNCTIONAL. When the motor is FUNCTIONAL, it may be either ON or OFF, but never both.

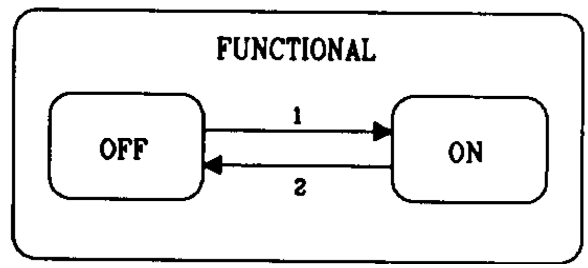


Figure A.5.2
Example of OR Substates

Another way of dividing a parent state corresponds roughly to subsystems. These AND substates represent parallelism, such that every AND substate of an active parent state is considered active. Harel also uses the term "Orthogonal Component" to refer to AND substates. However, these parallel substates tend to be highly interactive and interdependent. For this reason, the word orthogonal is considered confusing and has been excluded from use in this document. Figure A.5.3 shows an example of AND substates representing (in part) an automobile. Note the convention of attaching the name of the parent state AUTOMOBILE to the outside of the state in a small box. The substates shown are independent components and may have their own substates (of either the AND or OR type):

— LIGHTS may be ON or OFF;

— DOOR may be OPEN or CLOSED;

— ENGINE is constructed of components such as pumps, pistons, carburetor, etc.

Exiting one of a set of AND substates requires the exit of all others. In some cases, a transition arrow will be shown from only one of the substates with the others implied.

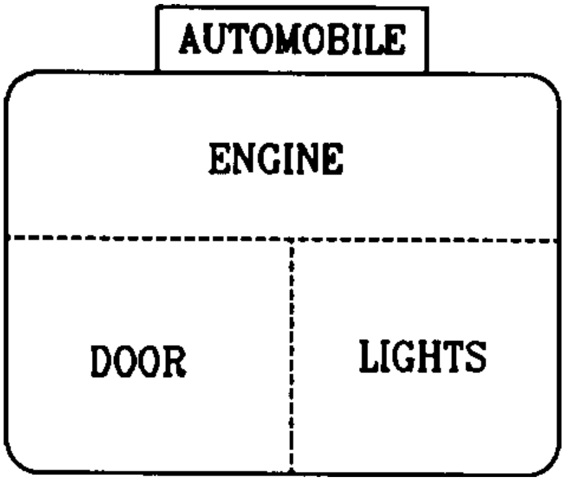


Figure A.5.3
Example of AND Substates

A simplification that also helps to prevent in determinacy is implemented with the symbol for default entry point. This symbol will indicate which OR substate is initially active when there is not an explicit choice. This lack of specification is indicated by a transition arrow from one state to another that does not cross the boundary of the parent to point specifically to a substate.

An entrance to a state terminating in a history symbol (see Figure A.5.1) indicates that the OR substate to be entered should be that which was active the last time the parent state was active (i.e., last time the car was running, the radio was on). The history symbol H refers to the choice of substates of the parent. The symbol H* extends further to the lowest level substates defined. In the absence of memory of a "last time", the default entry is used.

The selector and conditional selector symbols serve to abbreviate complex entrances to states. Their meaning is similar and indicate that the choice of OR substate upon entry of a parent state depends on some condition that is not shown. The selector is usually used to combine several similar transition events, while the conditional selector will typically require some computation or test of conditions external to the stimulus for state transition. Please examine the referenced article for more detail.

NOTE: Within the body of this document, the term statechart is not used in favor of the more traditional term state diagram.

A5.1 *State Definitions* — The state diagram provides a concise description of the function of a system. However, a full definition requires detail that cannot be included on the diagram. A description of each state is required that covers the boundaries of the state and any responses that occur within that state to the environment. The convention in this document is to provide state names in ALL CAPS to help the reader identify where these are used. A sample state description of the ON state depicted in the Figure A.5.2 might be:

ON

The switch is in the on position. Power is available to the motor. Speed of the motor will change in proportion to the speed knob adjustment.

A5.2 *Transition Table* — The last piece of the state model is the transition table. It consists of several columns that list the transition number from the diagram, the starting and ending state for the transition, and three columns titled trigger, action, and comment. The trigger column describes the combination of events and conditions that initiates the transition (e.g., message Sx,Fy received). The trigger should be related to a single clearly defined event at the equipment. The action column identifies the activities associated directly with the transition. These activities may be of three types: a) actions taken upon exit of the old state, b) actions taken upon entry to the new state, and c) actions not associated with either state. These are not differentiated in this document. The final column allows for additional comments that help to clarify the transition. Table A.5, an example of transition table, illustrates the motor example in Figure A.5.2.

Table A.5 Transition Table for Motor Example

#	*Current State*	*Trigger*	*New State*	*Action*	*Comment*
1	OFF	Switch turned to on position.	ON	Power supplied to motor.	Power supply assumed available. Motor begins to turn.
2	ON	Switch turned to off position.	OFF	Power supply to motor dis-connected.	Motor begins deceleration.

A.6 Example Control Model Application

This section provides one example of a host's interaction with an equipment's control model. A host system must have a view of the control model to understand and predict equipment behavior. However, the implementor may simplify the host's view by assuming that some configuration settings are fixed and that the host-initiated features are not implemented. Applying these assumptions simplifies the behavior the host expects to see.

Figure A.6.1 shows the effective control model[24] based on the following host assumptions:

— The fundamental requirements are met, but the additional host-intiated control capability is not implemented.

— The configuration for the default entry to CONTROL is set to an OFF-LINE substate (either ATTEMPT ON-LINE or EQUIPMENT OFF-LINE).

— The destination state for transition 4 (failure of S1,F1 transaction) is configured to EQUIPMENT OFF-LINE.

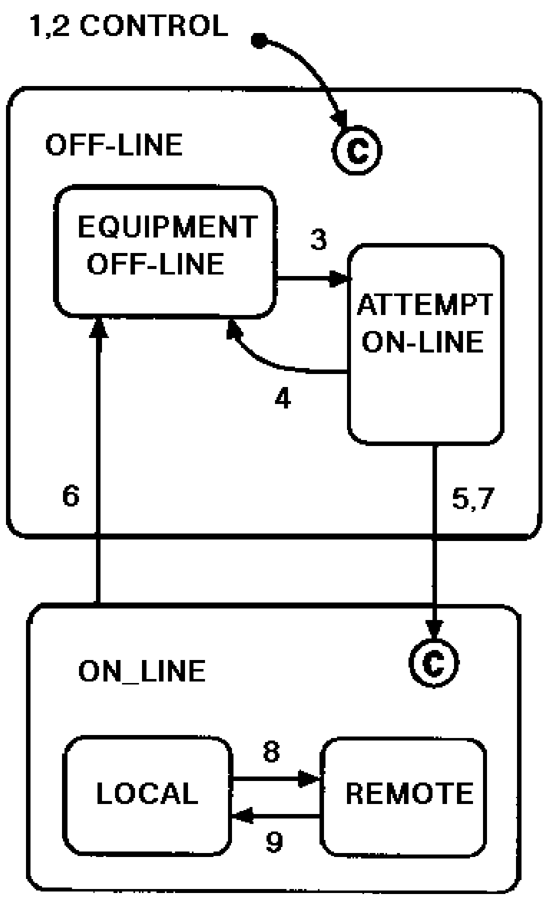


Figure A.6.1
Example of the Simplified "Effective" Control Model

This view of the model has two further settings that the host recognizes as changeable at the equipment. The first is the configuration of which substate of OFF-LINE to be activated upon system initialization. The second is the front panel switch that determines whether the active system substate is LOCAL or REMOTE when ON-LINE.

This application has the following implications:

— This application requires that the equipment begin with the OFF-LINE state active. Thus, an equipment initiated S1,F1/F2 transaction must be completed before the equipment will begin sending all messages to the host.

— If a failed attempt to go ON-LINE is made by the equipment, it will not allow the host to complete the transition at a later time. An operator will be required to re-initiate the transition to ON-LINE when the host becomes ready.

— Once ON-LINE, the equipment will remain ON-LINE until an operator sets the equipment OFF-LINE at the equipment front panel.

— Since all transitions into the HOST OFF-LINE state are eliminated, this state is effectively eliminated from the host view of the control model.

This application retains the following features:

— The ON-LINE state is achieved only after the host acknowledges the equipment by replying to the S1,F1 with and S1,F2. This confirms to the operator attempting to put the equipment ON-LINE that the host application is ready for work to begin.

— It provides the operator the means to set the equipment OFF-LINE for non-host-related activities[25] (e.g., maintenance, test lots).

— The operator has the ability to operate the equipment with either the REMOTE or LOCAL state active. As the equipment transitions to ON-LINE, the preferred substate is automatically chosen (based on a front panel switch).

— The user may configure which substate of OFF-LINE the equipment will initially activate at system initialization. If ATTEMPT ON-LINE is chosen, the equipment will automatically attempt the transition to the ON-LINE state as system initialization.

24 See Section 3.3 for details of the control model.

25 Which activities are "non-host-related" varies from factory to factory. In general, fewer activities are "non-host-related" as a factory's automation level increases.

A.7 Examples of Limits Monitoring

A7.1 *Introduction*

A7.1.1 Four limits monitoring examples are included below to help clarify the use of limits and to illustrate typical applications. The first example shows how to apply limits to boolean values. The second illustrates application of several limits to a floating point variable in a classical control zone style. The third example shows an integer counter variable used to prompt for equipment maintenance.

A7.2 *Examples*

A7.2.1 Example 1 — Valve Monitoring

A7.2.1.1 The ACME Shine-Um-Rite Model 13 includes a sump which contains the chemical agent used to clean bare wafers. A chemical feeder system serves to refill the sump when the level drops below a certain level. The fill is accomplished via an on-off value driven by sensors in the sump. Facilities must be informed of the proportion of the time the valve is open (approximates usage) and any time the value remains open for more than 5 minutes (valve likely broken).

A7.2.1.2 To implement this requirement, a limit was defined for the Boolean status variable which contains the current state of the valve (i.e., 0 = Closed, 1 = Open). See Figure A7.1 for illustration. LIMITID1 was defined with UPPERDB = LIMITMAX = 1 (Open) and LOWERDB = LIMITMIN = 0 (Closed). As a result, any time the valve opens, a collection event is generated with TransitionType = 0 and when the valve closes, a collection event is generated with TransitionType = 1. An event report containing the DVVAL LimitVariable was attached to each collection event and reporting for the event was enabled.

NOTE: Boolean values are defined as 0 = False/Closed/Off and any value > 0 = True/Open/On — never depend on a value of 1.

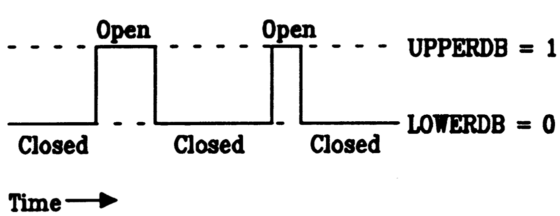


Figure A7.1
Valve Monitoring Example

A7.2.2 Example 2 — Environment Monitoring

A7.2.2.1 ACME also makes a Model 2 Stepper. The environmental control system of this equipment is designed to hold the internal temperature relatively constant, but is sensitive to large changes in the external environment, opening of access doors, etc. To ensure that processing conditions are appropriate, the internal stepper temperature is monitored to ensure it remains in a safe operating zone (within "Shutdown" limits). In addition, a second set of limits are used within the Shutdown limits to bound the "Normal" operating range. Frequent excursions from the normal range into the "warning" range will prompt service on the environmental control system. The target temperature range is specified as 98–100°, the shutdown limits as 95–103°.

A7.2.2.2 Event reports are desired when the internal temperature moves outside of the normal operating zone into a warning zone (above or below), when the temperature moves back into the normal operating zone from the warning zones, and when the temperature moves out of the warning zones into the shutdown zones. Furthermore, temperature fluctuations of 0.5° should not trigger multiple event reports.

A7.2.2.3 Probably the most intuitive use of the limits monitoring capability is in establishing normal, warning, and shutdown zones for a particular equipment variable. Limits may be combined to provide such a scenario. The method is described below and illustrated in Figure A7.2. Please note that in the figure, limits are denoted as solid lines for simplicity, with deadbands indicated using the ± notation.

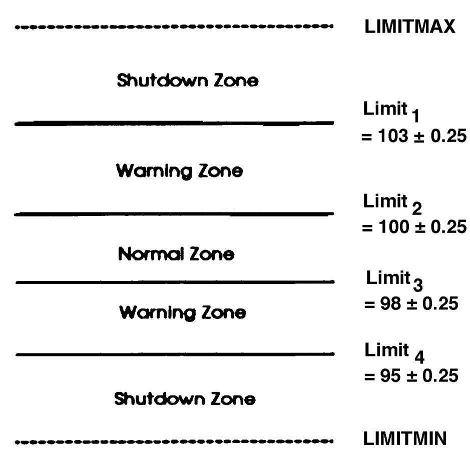


Figure A7.2
Environment Monitoring Example

A7.2.3 Example 3 — Calibration Counter

A7.2.3.1 Another ACME equipment is the multi-chamber Duz-It-All Model 7. This machine includes redundant chambers to increase throughput. One particular chamber on this equipment requires periodic calibration. The need for calibration is a non-linear function of the number of wafers processed in that chamber. A status variable exists which contains the number of wafers processed since the last calibration was performed. Maintenance is definitely required after every 8 cycles, but the machine must be checked after 5 and 7 cycles to determine whether early calibration is necessary. This checking may be done by examining certain other equipment status variables.

A7.2.3.2 To meet this need, three limits are defined for the counter variable. Three limits were set, at 5, 7, and 8. Deadbands are set to zero, since chattering is not a problem. All the pertinent information is placed in an event report which is attached to the CEID for the limits of the counter to negate the need for further message exchange. Event reports are generated as each limit is reached (one zone transition each), and when the counter is reset following calibration (one, two, or three zone transitions referenced in one report). Figure A7.3 illustrates this example. Note that disabling the report upon counter reset (downward transitions) is not possible.

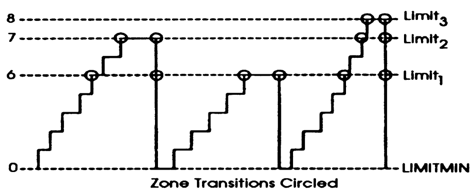


Figure A7.3
Calibration Counter Example

A7.2.4 Example 4 — Derived Variables

A7.2.4.1 The flagship of the ACME line is the new HotDog Furnace. This is a vertical furnace which exposes wafers to a variety of temperatures during processing. The temperature profile during the run is critical to the process and is typically contains a number of plateau's at different levels during the run. The owner wishes to monitor the temperature and be alerted whenever the actual temperature profile differs from the ideal by > 0.5°. The derived variable was created to provide a steady target range during the run, no matter what the desired temperature range happened to be. Deviation from "ideal temperature profile" was chosen as the new variable to be monitored. The equipment already had access to the profile for the run, which described the desired temperature at a given time into the process. The manufacturer added a calculation each time the actual temperature was sampled, subtracting the ideal temperature from the actual. They provided as status variables the actual temperature, the ideal temperature, and the new "deviation from profile" variable. One limit was activated and set to 0.5 degrees and a second set to –0.5 degrees (each with a deadband width of 0.05). Thus, when the temperature deviation from setpoint exceeds ± 0.5, an event is generated containing the current desired temperature and the actual temperature. For good measure, additional data was added, providing time since start of run to document the precise point in the process that the problem occurred.

A7.2.4.2 In order to achieve the desired behavior, the host defines four monitoring limits. Two of the limits establish the target zone. These are responsible for reporting transitions from normal to warning zones in either direction. The other two limits establish the transitions between the warning zones and the error zones. The difference between UPPERDB and LOWERDB for each limit is 0.5. This may also be expressed as limit ± 0.25. Combining limits does not change the way the equipment treats limits monitoring, but rather builds a method of interpreting limits from the host's point of view.

A.8 Process Parameter Modification for Process and Equipment Control

A8.1 *Introduction*

A8.1.1 In many equipment control applications there is a need for a GEM host to modify one or a small set of process parameters associated with a recipe. The number of parameters modified, frequency of modification (e.g., wafer-to-wafer, batch-to-batch, etc.), range of modification, etc., is largely a function of the equipment control application. Utilizing GEM, at least two methods are envisioned for modifying process parameters on a tool. With the first method, "Equipment Constants" can be used to relate process parameters of the updated recipe. Equipment Constants can also be used in a mode where they relate suggested modifications to process parameters from the stored recipe, i.e., the constants contain only the +/- differential from a nominal value. The former mode is preferred because it better ensures data integrity between the controller and tool. With the second method the entire recipe could be downloaded, but this results in an enormous amount of communication overhead. Note that, in all cases, the Equipment Constants do not replace the process parameters inside a recipe, but are associated with (e.g., linked to) these parameters to relate modifications. The remainder of this application note provides a description of how process parameter modification can be implemented using existing GEM capabilities. The method may be used in a GEM compliant system provided that the specific GEM capabilities described are supported.

A8.2 *Equipment Constants*

A8.2.1 Incremental process parameter modification for process and equipment control can be supported over a GEM interface by using the *Equipment Constants* GEM capability (see Section 4.5). With this capability, each process parameter (or process parameter at a step) that can be modified, e.g., for purposes of process control, is associated with an equipment constant. Using the Equipment Constant GEM scenarios (see Section 4.5.5) the host can (1), send process parameters or parameter modifications, (2), request current values of modifiable recipe parameters, (3), retrieve name lists of equipment constants associated with modifiable parameters, and (4), be informed by the equipment when an operator changes one of the modifiable process parameters.

A8.2.2 The equipment constants should represent the actual values of the process parameters with which they are associated. Depending on the equipment operation and control application, the equipment constant could represent the actual value of a process parameter at a recipe step, or over the entire recipe. The equipment constants could also be utilized to represent the differentials of process parameters from nominal values. However it is important to note that, when using differential values to relate process parameter modifications, any loss of synchronization between equipment and host could result in an incorrect assessment of the value of a process setpoint by the host. Note also that, upon system startup, and whenever the appropriate process parameters are modified, the equipment constants should also be modified as necessary to always reflect the (absolute or relative) values of the associated process parameters.

A8.2.3 In order to maintain synchronization between equipment and host, it is recommended that equipment constants associated with recipe parameters be applied to only override the currently selected and active recipe. A selected recipe is considered to be active whenever the equipment is in the "PROCESSING" state and the recipe is the currently selected recipe (process program). If multiple recipes are utilized during one process event, e.g., cluster tool scenario, it is recommended that separate equipment constants be utilized for each recipe/process parameter pair.

A8.2.4 Note that timing and traceability issues associated with utilizing the equipment constants capability (for process control) are application specific and beyond the scope of this application note. Equipment that provides for recipe parameter overrides though setting of Equipment Contstants should also provide additional Equipment Constants for the set that includes the name of the associated process program and a Boolean variable to enable and disable the override feature. In addition, the supplier should document for each parameter: (1), the associated Equipment Constant, and (2), any restrictions on the state (active or not) or the recipe in which the parameter may be modified. Also, since override of the process setpoints may be provided by a Host controller application element, it is recommended that the equipment provide an event report whenever the associated recipe has been modified.

A8.3 *Example*

A8.3.1 In the example of Figure A.8.1, a Chemical-Mechanical Planarizer (CMP) single wafer "polishing" system includes a GEM compliant planarizer (equipment), a thickness metrology unit and a (host) controller. The tool polishes a wafer to a target thickness. The post-process thickness is measured by the metrology unit and reported to the host controller. The controller utilizes a feedback control algorithm to determine the appropriate polish "time" recipe parameter for the next wafer. This time should be reported to the CMP equipment utilizing the equipment's GEM interface so that the information can be utilized for the next wafer processing event.

A8.3.2 The mechanism described in this application note could be utilized to implement process control as follows. A settable equipment constant is associated or "linked" with the "time" parameter on the equipment. Equipment system documentation indicates the linkage and the conditions under which the linkage is valid. When the controller determines an appropriate "time" parameter value for the next wafer to be polished, it sets the equipment constant to this value (see Section 4.5). The equipment is configured to accept this equipment constant change and, if the equipment is in (or possibly when the equipment reaches) the appropriate state, the recipe parameter is modified.

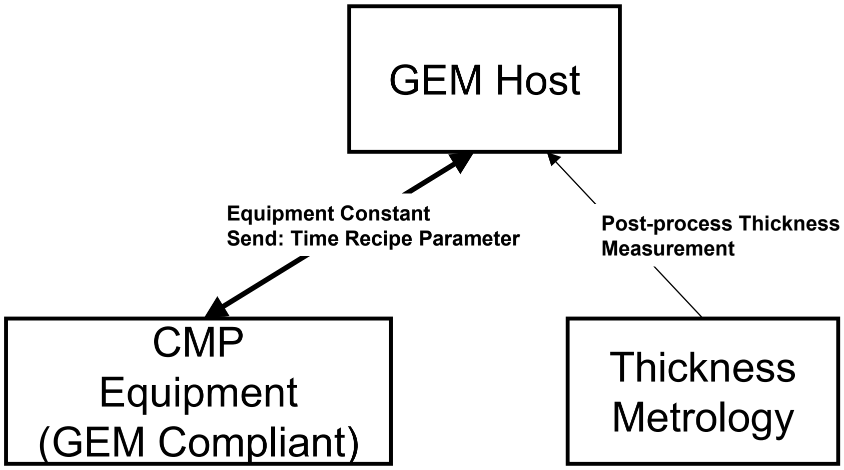


Figure A.8.1
CMP Single Wafer "Polishing" System with Host Recipe Parameter Modification Capability

NOTICE: These standards do not purport to address safety issues, if any, associated with their use. It is the responsibility of the user of these standards to establish appropriate safety and health practices and determine the applicability of regulatory limitations prior to use. SEMI makes no warranties or representations as to the suitability of the standards set forth herein for any particular application. The determination of the suitability of the standard is solely the responsibility of the user. Users are cautioned to refer to manufacturer's instructions, product labels, product data sheets, and other relevant literature respecting any materials mentioned herein. These standards are subject to change without notice.

The user's attention is called to the possibility that compliance with this standard may require use of copyrighted material or of an invention covered by patent rights. By publication of this standard, SEMI takes no position respecting the validity of any patent rights or copyrights asserted in connection with any item mentioned in this standard. Users of this standard are expressly advised that determination of any such patent rights or copyrights, and the risk of infringement of such rights, are entirely their own responsibility.

Index

A

B

C

D

E

F

G

H

I

L

M

Multi-Block Inquire 65
Multi-block Not Allowed 47, 66
must ....
6, 7, 9, 10, 14, 15, 16, 18, 21, 22, 23, 24, 26, 27, 28, 30, 31, 34, 35, 36, 37, 39, 40, 42, 44, 45, 47, 48, 50, 53, 54, 55, 56, 58, 59, 63, 68, 71, 72, 74, 76, 77

N

New Equipment Constant Acknowledge 64
New Equipment Constant Send 64
non-volatile memory 9, 34, 37, 40, 55, 56
non-volatile storage 22, 28, 39, 40, 41, 52, 55, 56, 59, 71
NOT 3, 9, 10, 11, 12, 19, 20, 52, 53, 54, 55, 72
NOT COMMUNICATING 9, 10, 11, 12, 19, 20, 52, 53, 55, 72

O

On-Line Data 64
On-line Identification 17, 30, 67
On-Line Identification 69

P

PAUSE 14, 16, 17, 36
PAUSED 36
Post-Processing 2, 71, 72
PPChangeName 39, 41, 62, 63
PPChangeStatus 39, 41, 62, 63
PPExecName 36, 39, 62, 63
PPID 36, 39, 40, 41, 62, 71
PP-SELECT 36, 39, 43
PreviousProcessState 16, 62, 63
PROCESS 16, 17
process program 6, 13, 14, 16, 35, 36, 39, 40, 41, 42, 43, 71
Process Program 1, 17, 39, 41, 42, 43, 44, 63, 65, 66, 67, 68, 69
Process Program Change 39, 63
Process Program Data 41
Process Program Download 42, 43
Process Program Load 1, 42, 43, 65
Process Program Load Grant 42, 43
Process Program Load Inquire 42, 43, 65
Process program management 39
Process Program Management 1, 17, 39, 63, 67, 68, 69
Process Program Request 41, 44, 66
Process Program Send 41, 42, 43, 44, 66
Process Program Validation 39
Process Program Verification 39, 43, 44, 66
Process Program Verification Send 66
Process Set-Up 71
processing 3, 5, 6, 7, 13, 14, 16, 17, 18, 21, 31, 35, 36, 39, 56, 63, 71, 72, 77, 78
Processing 1, 2, 16, 17, 36, 63, 67, 69, 71, 72
PROCESSING ACTIVE 16, 17
ProcessState 16, 62, 63

R

RCMD 36, 62
READY 16, 17

S

T

U

V

SEMI E30.1-0200
INSPECTION AND REVIEW SPECIFIC EQUIPMENT MODEL (ISEM)

This standard was technically approved by the Global Information and Control Committee and is the direct responsibility of the North American Information and Control Committee. Current edition approved by the North American Regional Standards Committee on September 3, 1999. Initially available on SEMI OnLine November 1999; to be published February 2000. Originally published June 1998.

CONTENTS

SEMI E30.1-0200
INSPECTION AND REVIEW SPECIFIC EQUIPMENT MODEL (ISEM)

This standard was technically approved by the Global Information and Control Committee and is the direct responsibility of the North American Information and Control Committee. Current edition approved by the North American Regional Standards Committee on September 3, 1999. Initially available on SEMI OnLine November 1999; to be published February 2000. Originally published June 1998.

1 Purpose

1.1 This standard establishes a Specific Equipment Model (SEM) for Inspection and Review Equipment (ISEM). The model consists of equipment characteristics and behavior that are to be implemented in addition to the SEMI E30 fundamental requirements and additional capabilities. The intent of this standard is to facilitate the integration of ISEM equipment into an automated (semiconductor fabrication) factory. This document accomplishes this by defining an operational model for ISEM equipment as viewed by a factory automation controller. This definition provides a standard host interface and equipment operational behavior (e.g., control, state models, data reports, and reporting levels). Several topics require additional activity that are within the scope of this standard: substrate pattern maps; defect classification code management; and review data management.

2 Scope

2.1 The scope of this standard is limited to the definition of Inspection, Review, and Inspection/ Review equipment behavior as perceived by a SEMI Equipment Communications Standard II (SEMI E5/SECS-II) host that complies with SEMI E30. It defines the external view of the equipment through the SECS link; it does not define the internal operation of the equipment. This standard expands the SEMI E30 requirements and capabilities in the areas of the processing state model, remote commands, variable items, alarms, and data collection.

2.2 This standard is intended for ISEM equipment that generates data and information about anomalies and defects found on substrates. Inspection equipment finds anomalies. Anomalies are occurrences on a substrate that have been judged to be unexpected, abnormal, incongruous, or inconsistent. Anomalies may be examined using review equipment, at which time they may be classified as defects or non-defects. Some inspection equipment may generate, and some review equipment may use, coordinate data to locate anomalies on a substrate. The accuracy of the coordinate data generated or used is equipment-dependent.

2.3 This standard does not purport to address safety issues, if any, associated with its use. It is the responsibility of the users of this standard to establish appropriate safety and health practices and determine the applicability of regulatory limitations prior to use.

3 Limitations

3.1 This document addresses three distinct types of equipment: inspection, review, and inspection/review. The term ISEM equipment refers to all three types of equipment. These three equipment types are differentiated by the basic functions they perform:

3.1.1 Inspection Equipment that looks for anomalies on a substrate and reports information regarding those anomalies. Inspection equipment may determine the location of anomalies relative to a coordinate system. Inspection equipment may also provide other types of data related to the anomaly.

3.1.2 Review Equipment that accepts information about anomalies on a substrate, gathers information on those anomalies, and reports that data.

3.1.3 Inspection/Review Equipment having the characteristics of both inspection and review equipment.

4 Referenced Standards

4.1 *SEMI Standards*

SEMI E5 — SEMI Equipment Communications Standard 2 Message Content (SECS-II)

SEMI E30 — Generic Model for Communications and Control of SEMI Equipment (GEM)

SEMI E37 — High-Speed SECS Message Services (HSMS) Generic Services

SEMI E37.1 — High-Speed SECS Message Services Single-Session Mode (HSMS-SS)

SEMI E58 — Automated Reliability, Availability, and Maintainability Standard (ARAMS): Concepts, Behavior, and Services

SEMI M20 — Specification for Establishing a Wafer Coordinate System

SEMI M21 — Specification for Assigning Addresses to Rectangular Elements in a Cartesian Array

4.2 *Other References*

Harel, D., "Statecharts: A Visual Formalism for Complex Systems," Science of Computer Programming 8, (1987), 231-274

NOTE 1: As listed or revised, all documents cited shall be the latest publications of adopted standards.

5 Terminology

5.1 *Abbreviations and Acronyms*

5.1.1 *GEM* — generic equipment model

5.1.2 *PE* — pattern element

5.1.3 *TCP/IP* — Transmission Communication Protocol/Internet Protocol

5.2 *Definitions*

5.2.1 *align* — to put into proper relative position or orientation.

5.2.2 *alignment* — a procedure in which a coordinate system is established on a substrate.

5.2.3 *alignment mark* — a feature on the substrate selectively used for alignment.

5.2.4 *anomaly* — an occurrence on a substrate that has been judged to be unexpected. Something abnormal, incongruous, or inconsistent.

NOTE 2: After an anomaly is reviewed, it may be classified as a defect.

5.2.5 *batch* — a group of substrates or lots intended for a process sequence versus single substrate processing.

5.2.6 *carrier* — a container with one or more fixed positions at which material can be held.

5.2.7 *carrier location* — a physical place within the equipment capable of holding a carrier.

5.2.8 *cassette* — a container with one or more substrate locations (see *slot*).

5.2.9 *defect* — 1) A physical, optical, chemical, or structural irregularity that degrades the ideal substrate structure or the thin films built over the substrate. 2) An undesirable classified anomaly.

5.2.10 *defect classification* — the categorization of defects according to some systematic division based on their physical, optical, chemical, or structural properties.

5.2.11 *die* — 1) A field sub-unit. 2) An area of substrate that contains the device being manufactured.

5.2.12 *ended* — the end of a state that may be when it is normally completed, or its early end due to an allowed or atypical condition (e.g., a STOP command, or an error or alarm condition).

5.2.13 *factory automation controller* — a computer system that provides integration of factory shop control and business systems with semiconductor equipment.

5.2.14 *feature* — 1) A line or a point (as a feature within a pattern). 2) A physical characteristic of the substrate (e.g., a substrate flat).

5.2.15 *field* — the printed pattern from a reticle.

5.2.16 *field of view* — the imaging area as seen at magnification of the inspection or review equipment.

5.2.17 *global alignment* — a procedure which establishes a coordinate system for the entire substrate (see *alignment*).

5.2.18 *group* — a logical collection of regions.

5.2.19 *group alignment* — a procedure which establishes a coordinate system for an area, which is a contiguous group (see *alignment*).

5.2.20 *inspect* — to detect anomalies and/or information about anomalies.

5.2.21 *inspection* — an examination to detect anomalies.

5.2.22 *inspection equipment* — equipment that looks for anomalies on a substrate and reports information regarding those anomalies. Inspection equipment may determine the location of anomalies relative to a coordinate system. Inspection equipment may also provide other types of data related to the anomaly.

5.2.23 *inspection/review equipment* — equipment having the characteristics of both inspection and review equipment.

5.2.24 *ISEM job* — the information required to specify an inspection or review that may include material identification and location and process program identifications as well as any other parameters required to obtain a desired result.

5.2.25 *layer* — one of a sequential series of overlaying photomasks that make up a device series.

5.2.26 *lot* — a group of one or more substrates of the same type (e.g., wafers, masks, CDs).

5.2.27 *major flat* — the flat of longest length that is commonly located with respect to a specific crystal plane (ASTM F 1241-89).

5.2.28 *mask* — a selective barrier to the passage of radiation. For example, a transparent plate containing an opaque pattern that is used to transfer that pattern to another substrate.

5.2.29 *material* — a piece or pieces of substrate, one or more substrates, a lot, a batch, or a run.

5.2.30 *metrology equipment* — any equipment that collects and reports information on specific predetermined locations or features on a substrate with consistent data structure or that reports general information about the entire substrate.

5.2.31 *notch* — a U-shaped cut on the edge of a substrate that is commonly located with respect to a specific crystal plane.

5.2.32 *overlay* — the actual distance between two features on different layers of a substrate, compared to the expected distance.

5.2.33 *pattern* — 1) The physical features on a substrate surface. 2) An ideal pattern is the arrangement of features expressed in a calculated or mathematical manner.

5.2.34 *pattern element* — 1) Any recognizable set of features. 2) A rectangular sub-unit of a pattern or a pattern element. There may be multiple levels of pattern elements.

5.2.35 *primary fiducial* — a key characteristic of a substrate used to align the substrate during processing (such as a *notch* or *major flat*).

5.2.36 *region* — a single field of view which may be a collection of sites.

5.2.37 *registration* — the actual distance between two features on the same layer of a substrate, compared to the expected distance.

5.2.38 *reticle* — a mask that contains the patterns to be reproduced on a substrate; the image may be equal to or larger than the final projected image.

5.2.39 *review* — the process of classification of anomalies which may result in the appending of additional data to inspection data. Used to create a field on a substrate.

5.2.40 *review equipment* — equipment that accepts information about anomalies on a substrate, gathers information on those anomalies, and reports that data.

5.2.41 *run (noun)* — the material processed during the EXECUTING state.

5.2.42 *run (verb)* — the actions of a process between the READY state and the STOPPING state.

5.2.43 *safe state* — a state in which the equipment presents no danger to the product or user. This implies that safety interlocks are in place such that the equipment can be serviced without harm to the operator and that the material being processed has been removed from the processing station into an accessible location.

5.2.44 *site* — a single x,y coordinate where an action can be performed (e.g., *alignment* or *review*). The area associated with a site is determined by the equipment accuracy (e.g., optics, stage algorithms).

5.2.45 *slot* — a physical location within a Carrier capable of containing a substrate. Also referred to as a carrier location.

5.2.46 *substrate* — the basic unit of material, processed by semiconductor equipment, such as wafers, CDs, flat panels, or masks.

6 Communication Requirements

6.1 It is required that any ISEM-compliant equipment follow the Communications State Model in SEMI E30. In addition, ISEM-compliant equipment shall support the High Speed Messaging Service Standard (SEMI E37/HSMS). It is a minimum requirement to support Single Session (SEMI E37.1/HSMS-SS) sending SECS-II messages over TCP/IP. The reason behind this requirement is the size of the process programs used by this class of equipment and the amount of data produced.

7 State Models

7.1 *Processing State Model Requirements*

7.1.1 The processing state model included in this standard is a requirement for ISEM equipment. This standard requires implementation of all SEMI E30 state models (such as control, communication, and on-line/off-line). A state model consists of state model diagrams, state definitions, and a state transition table. All state transitions in this standard, unless otherwise specified, shall correspond to collection events.

7.1.2 A state model is the host's view of the equipment and does not necessarily describe the internal equipment operation. All ISEM state model transitions shall be mapped sequentially into the appropriate internal equipment events that satisfy the requirements of those transitions. In certain implementations, the equipment may enter a state and have already satisfied all of the conditions required by the ISEM state model for transition to another state. The equipment makes the required transition without any additional actions in this situation.

7.1.3 Some equipment may need to include additional states other than those in this standard. Additional states may be added but shall not change the ISEM-defined state transitions. All expected transitions between ISEM states shall occur.

7.2 *Processing State Model Diagram* — Processing state models are detailed for ISEM equipment in Figure 1. This diagram contains all states and transitions that are common to all three types of ISEM equipment. The WORKING state is different for each type of equipment. The same state names and transition identifiers are used to identify common states and transitions of the three types of equipment. The working states for the three types of equipment are presented in the following sections. All states and transitions are described in the section following the diagrams.

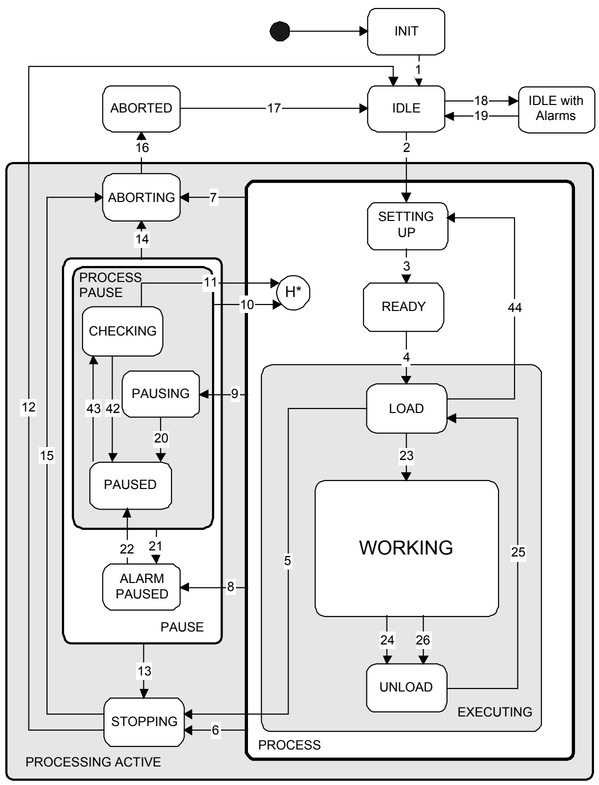


Figure 1
Generic ISEM Processing State Model Diagram

7.2.1 *Working State for Inspection Equipment Model* — The processing state model for inspection equipment is identical to the Generic ISEM Processing State Model (Figure 1). Only the WORKING state is unique to the inspection equipment processing state model. This is shown in Figure 2.

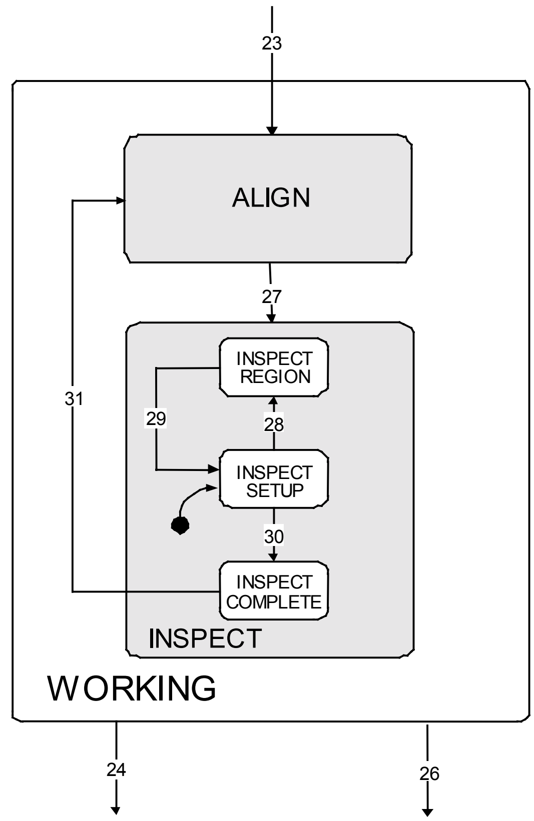


Figure 2
Working State for Inspection Equipment

7.2.2 *Working State for Review Equipment* — The processing state model for review equipment is identical to the generic ISEM Processing State Model (Figure 1). Only the WORKING state is unique to the review equipment processing state model. This is shown in Figure 3.

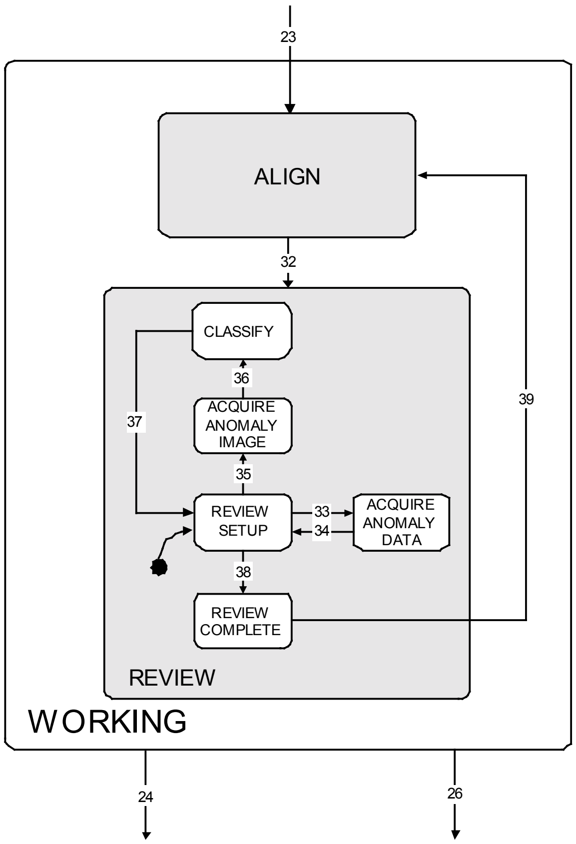


Figure 3
Working State for Review Equipment

7.2.3 *Working State for Inspection/Review Equipment* — The processing state model for inspection/review equipment is identical to the generic ISEM Processing State Model (Figure 1). Only the WORKING state is unique to the inspection/review equipment processing state model. This is shown in Figure 4.

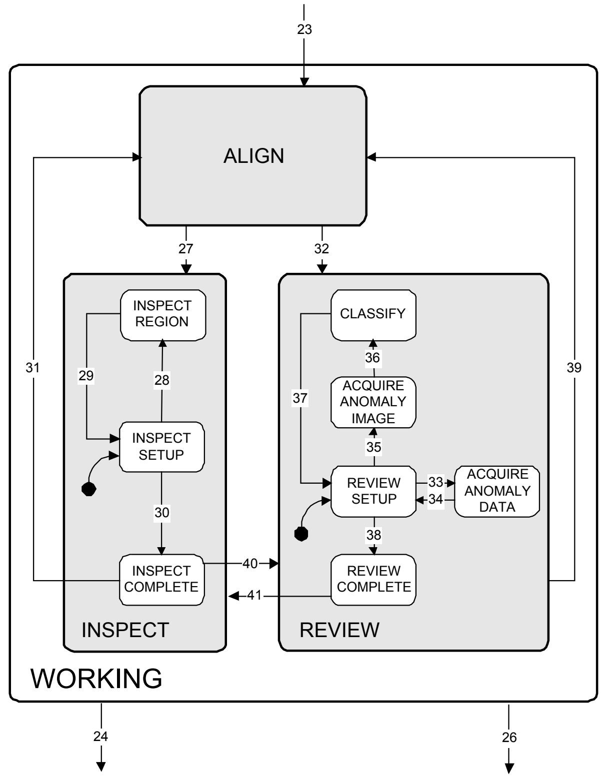


Figure 4
Working State for Inspection and Review Equipment

7.3 *Processing State Definitions*

7.3.1 *ABORTED* — All activity is suspended as a result of an ABORT command. Any alarm and abort conditions shall be cleared before exit from this state. The CLEANUP command is available to the operator or host to transition the equipment from the ABORTED state to IDLE state.

7.3.2 *ABORTING (PROCESSING ACTIVE Sub-State)* — The equipment has received an ABORT command. All normal activity is suspended. The equipment is taking appropriate action to put the equipment and material in a "safe state" where possible. Data may be invalid or not available.

7.3.3 *ACQUIRE ANOMALY DATA (REVIEW Sub-State)* — Data is being acquired about anomaly locations.

7.3.4 *ACQUIRE ANOMALY IMAGE (REVIEW Sub-State)* — The equipment is obtaining a view of the anomaly.

7.3.5 *ALARM PAUSED (PAUSE Sub-State)* — An alarm has occurred in the PROCESS or PROCESS PAUSE states, and the equipment is waiting for the alarm to be cleared or a command (STOP or ABORT).

7.3.6 *ALIGN (WORKING Sub-State*) — The equipment or operator is performing an alignment of the material to the equipment. If needed, within this state, the equipment shall refine or establish its SEMI M20 coordinate system and establish any secondary coordinate systems.

7.3.7 *CHECKING (PROCESS PAUSE Sub-State)* — The equipment verifies that the process program update request is valid. No process program parameters are changed unless "all" reported updates are valid. This is a similar procedure to that which is done in SETTING UP before the equipment is ready to transition to the READY state. Valid commands in this state are STOP, ABORT, and RESUME.

7.3.8 *CLASSIFY (REVIEW Sub-State)* — The operator or equipment is determining the classification of an anomaly.

7.3.9 *EXECUTING (PROCESS Sub-State)* — The equipment is processing material automatically and can continue to do so without external intervention but normally may include interaction with the host or operator.

7.3.10 *IDLE* — Checks for queued ISEM jobs or awaits a PP-SELECT, MAP-CARRIER, or PP-ASSIGN command. IDLE is free of alarm and error conditions. Any transition into this state shall deselect any selected Process program(s).

7.3.11 *IDLE with ALARMS* — An alarm has occurred in the IDLE state, and the equipment is waiting for all alarms to be cleared.

7.3.12 *INIT* — Equipment initialization is occurring. Equipment remains in this state unless initialization is successful.

7.3.13 *INSPECT (WORKING Sub-State)* — The current alignment area of the substrate is being inspected for anomalies.

7.3.14 *INSPECT COMPLETE (INSPECT Sub-State)* — The equipment has completed inspection of the current alignment area. Based on the recipe, the equipment determines if (a) additional alignment areas are required to do more inspections, (b) the recipe on this material is complete, or (c) a review of the current inspection area is required.

7.3.15 *INSPECT REGION* — A region on a substrate is being inspected for anomalies.

7.3.16 *INSPECT SETUP (INSPECT Sub-State)* — The equipment is in this sub-state immediately upon entering the INSPECT state. The equipment is determining if all conditions are satisfied to begin inspecting the regions in the current alignment as defined by the recipe and any commands.

7.3.17 *LOAD (EXECUTING Sub-State)* — The equipment is determining if the process program has completed. When additional processing is required, then the next unprocessed substrate shall be transferred to the equipment processing location, such as the stage. If equipment determines that there are more process programs in the "CARRIERBLD" ISEM job, the equipment makes the transition to setup for the next process program specified. Otherwise, the equipment transitions to IDLE through STOPPING.

7.3.18 *PAUSE (PROCESS ACTIVE Sub-State)* — PROCESS shall be suspended at the next opportunity. Actions to put the equipment in a "safe state" shall be performed. The equipment is awaiting a command (STOP or ABORT).

7.3.19 *PAUSED (PROCESS PAUSE Sub-State)* — PROCESS has been suspended, and the equipment is waiting for a command (RESUME, PP-UPDATE, STOP, or ABORT).

7.3.20 *PAUSING (PROCESS PAUSE Sub-State*) — PROCESS shall be suspended at the next opportunity, and the equipment shall be put in a "safe state." ABORT, STOP, and RESUME commands are valid in this state.

7.3.21 *PROCESS (PROCESSING ACTIVE Sub-State)* — This state is the parent of those sub-states which refer to the active preparation and execution of a process program.

7.3.22 *PROCESSING ACTIVE* — This state is the parent of all sub-states where the context of a process program execution exists.

7.3.23 *PROCESS PAUSE (PAUSE Sub-State)* — The equipment is free of alarm conditions in the PAUSE state. The equipment is awaiting for a command (ABORT, RESUME, or STOP).

7.3.24 *READY (PROCESS Sub-State)* — The equipment is ready to begin processing and is awaiting a START command from the operator or host.

7.3.25 *REVIEW (WORKING Sub-State)* — Classification is being done on anomalies previously found in the current alignment area of the substrate.

7.3.26 *REVIEW COMPLETE (REVIEW Sub-State)* — The equipment has completed review of the current alignment area. Based on the recipe, the equipment determines if (a) additional alignment areas are required to do more classifications, (b) the recipe on this material is complete, or (c) an inspection is required.

7.3.27 *REVIEW SETUP (REVIEW Sub-State)* — The equipment is in this sub-state immediately upon entering the REVIEW state. The equipment is determining if all conditions are satisfied to begin reviewing the anomalies in the current alignment as defined by the recipe and any commands.

7.3.28 *SETTING UP (PROCESS Sub-State)* — The equipment is being setup so that external conditions are satisfied to start processing the material. This includes the receipt of any process programs and material to be processed and their validation. Any of these conditions may be satisfied on entry to SETTING UP. For example, the selected process program may have already been loaded (e.g., if it was the default process program), or the specified material may have already been placed on the equipment material port. Additional information may come from the host during the execution of this state.

7.3.29 *STOPPING (PROCESSING ACTIVE Sub-State)* — The equipment has completed all process programs within a "CARRIERBLD" ISEM job or has been instructed to stop processing and shall do so gracefully at the next opportunity. All cleanup necessary is being completed within this state with regard to material, data, control system, etc. Data is normally preserved. Any alarm or error condition in this state causes a transition to ABORTING.

7.3.30 *UNLOAD (EXECUTING Sub-State)* — The substrate is being removed from the processing location.

7.3.31 *WORKING (EXECUTING Sub-State)* — The equipment is processing a specific material.

7.4 *Processing State Transition Table*

Table 1 Processing State Transition Table

Transition #	*Previous State*	*Trigger*	*New State*	*Actions*	*Comments*
1	INIT	All equipment initialization is complete with no alarms or error conditions.	IDLE	Equipment awaits for a PP-SELECT, PP-ASSIGN, or MAP-CARRIER command.	All equipment requires INIT to be free of errors and alarms when exited. IDLE state entry requires that no process program is selected.
2	IDLE	A ISEM job has been or is queued (PP-ASSIGN) or selected (PP-SELECT).	SETTING UP	The setup procedure is equipment-dependent.	Commit has been made to setup. Material may have been placed on the equipment before SETUP is entered. When the job becomes active, the process program gets selected.
3	SETTING UP	All setup activity has completed, and the equipment is ready to receive a START command.	READY	The equipment is waiting for a START command. START may be initiated by an operator or may be included in the process program.	The selected process program is available for execution. When running multiple process programs within a ISEM job, the equipment makes the next process program available for execution.
4	READY	The equipment receives a START command from the user or from within the body of the process program selected for execution.	LOAD	The equipment determines if processing is completed. If not, it transfers the next substrate to the processing location.	Equipment transitions to STOPPING when all process programs in the selected ISEM job are executed. If a new process program within the ISEM job needs to be executed, the equipment transitions to SETTING UP.
5	LOAD	The processing job is complete, and there are no more substrates to load or process programs to run.	STOPPING	Equipment initiates a cleanup to remove the completed ISEM job and process program.	Normal completion of the run.
6	PROCESS	The equipment has received a STOP command.	STOPPING	The equipment unloads the material and brings the equipment to a "safe state."	Data is typically preserved and is valid.

Transition #	*Previous State*	*Trigger*	*New State*	*Actions*	*Comments*
7	PROCESS	The equipment has received an ABORT command.	ABORTING	The equipment is put in a “safe state” if necessary.	Data may be invalid or not available.
8	PROCESS	An alarm occurs.	ALARM PAUSED	PROCESS activity is suspended, and the equipment is waiting for all alarms to be cleared.	ALARM PAUSED is a PAUSE Sub-State.
9	PROCESS	The equipment has received a PAUSE command.	PAUSING	PROCESS shall be suspended at the next opportunity. Actions to put the equipment in a “safe state” will be performed.	This transition is required if the user wants to make changes to the current process program being executed.
10	PROCESS PAUSE	The equipment has received a RESUME command.	Previous PROCESS State	Proceed with the suspended process Sub-State.	If a RESUME command is received in the CHECKING state, then the PP-UPDATE command is canceled. Some equipment may only allow RESUME remote command from the PAUSE state.
11	CHECKING	The equipment has validated “all” requested updates to the current process program being executed; changes are done before entering into the next state.	Previous PROCESS State	Verification is appropriate in this state to check the changes made to the process program updated.	If an alarm occurs in the CHECKING state, then the PP-UPDATE command is canceled.
12	STOPPING	The equipment cleanup is complete, and the equipment is free of alarms.	IDLE	Equipment waits for a command/or determines if there is a ISEM job queued.	IDLE state entry requires that no process program is selected.
13	PAUSE	The equipment has received a STOP command.	STOPPING	The equipment proceeds with cleanup.	Normally, data is preserved and is valid.
14	PAUSE	The equipment has received an ABORT command.	ABORTING	Any unsafe condition is resolved if possible.	Data may be invalid or not available.
15	STOPPING	The equipment has received an ABORT command or an alarm was received while STOPPING.	ABORTING	Any unsafe condition is resolved if possible.	Data may be invalid or not available.
16	ABORTING	Unsafe conditions have been resolved where possible.	ABORTED	The equipment is waiting for alarm and ABORT conditions to be cleared.	The only state change allowed is to IDLE.
17	ABORTED	All alarms and abort conditions have been cleared.	IDLE	Equipment is waiting for a command (PP-SELECT, PP-ASSIGN, or MAP-CARRIER).	If needed, the CLEANUP command clears the abort conditions. IDLE state entry requires that no ISEM job or process program be selected.
18	IDLE	An alarm is set.	IDLE with ALARMS	The equipment waits for all alarms to be cleared.	None
19	IDLE with ALARMS	All alarms have been cleared.	IDLE	None	IDLE is free of alarms.

Transi-tion #	*Previous State*	*Trigger*	*New State*	*Actions*	*Comments*
20	PAUSING	The equipment has achieved a “safe state.”	PAUSED	The equipment is waiting for a command (PP-UPDATE, RESUME, STOP, or ABORT).	None
21	PROCESS PAUSED	An alarm is set.	ALARM PAUSED	The equipment waits for all alarms to be cleared or for a STOP or ABORT command.	None
22	ALARM PAUSED	All alarms are cleared.	PAUSED	The equipment is waiting for a command (RESUME, PP-UPDATE, STOP, or ABORT).	None
23	LOAD	Material transfer to processing location is complete.	WORKING	The substrate is being processed.	None
24	WORKING	The processing of the specific material being processed successfully completed.	UNLOAD	This material is transferred from the processing location.	Normal completion of the substrate.
25	UNLOAD	The material unload is complete.	LOAD	The equipment returns to LOAD and determines if processing is complete, if not, transfers the next substrate to the processing location.	None
26	WORKING	The processing of the specific material being processed abnormally ended.	UNLOAD	This material is transferred from the processing location.	Error exit from WORKING. Data may be invalid.
27	ALIGN	The material alignment is complete, and inspection is required.	INSPECT	The equipment determines if another region needs to be inspected.	This transition is to the INSPECT SETUP Sub-State of INSPECT.
28	INSPECT SETUP	All inspect setup activity is complete, and the inspection is not complete.	INSPECT REGION	The equipment inspects the current alignment region.	None
29	INSPECT REGION	The region inspection has ended.	INSPECT SETUP	The equipment determines if another region needs to be inspected.	None
30	INSPECT SETUP	Inspection of this alignment group is complete.	INSPECT COMPLETE	The equipment determines if (a) additional alignment areas are required to do more inspections, (b) the recipe on this material is complete, or (c) a review of the current alignment area is required.	The next transition is conditional.
31	INSPECT COMPLETE	The inspection of this alignment area ended, and additional inspections may be required.	ALIGN	An inspection group is complete, and additional inspections may be required.	None
32	ALIGN	The material alignment is complete, and review is required.	REVIEW	The material is reviewed.	This transition is to the REVIEW SETUP Sub-State of REVIEW.
33	REVIEW SETUP	Anomaly data is needed to perform the review.	ACQUIRE ANOMALY DATA	Anomaly data is being acquired.	Anomaly data may come from the host or equipment.

Transition #	*Previous State*	*Trigger*	*New State*	*Actions*	*Comments*
34	ACQUIRE ANOMALY DATA	Anomaly data has been acquired for the review, or no more anomaly data is available.	REVIEW SETUP	The equipment determines what to do.	
35	REVIEW SETUP	The equipment has anomaly data, and the review is not complete.	ACQUIRE ANOMALY IMAGE	The equipment acquires the anomaly image at the specified site.	The image may be a stored image or from an imaging device.
36	ACQUIRE ANOMALY IMAGE	The equipment has acquired the anomaly image for the specified site.	CLASSIFY	The operator or equipment classifies the anomaly.	None
37	CLASSIFY	All anomalies have been classified for the site.	REVIEW SETUP	The equipment determines what to do.	None
38	REVIEW SETUP	The review of the align area is complete.	REVIEW COMPLETE	Transition to next state is to be determined.	None
39	REVIEW COMPLETE	The review of this alignment area ended, and additional review is required.	ALIGN	A review group is complete, and additional alignment is required.	None
40	INSPECT	The alignment area inspection is complete, and review is required.	REVIEW	The material is reviewed.	This transition is to the REVIEW SETUP Sub-State of REVIEW.
41	REVIEW COMPLETE	The review is complete, and inspection is required.	INSPECT	The material is inspected.	This transition is to the INSPECT SETUP Sub-State of INSPECT.
42	CHECKING	Validation of requested process program changes failed.	PAUSED	The equipment is waiting for a new command.	No process program parameters have been changed.
43	PAUSED	The equipment receives a PP-UPDATE command.	CHECKING	The equipment begins validating requested changes to the process program.	No process program parameters are updated or changed before “all” requested changes are validated.
44	LOAD	Previous process program has completed, and there are additional process programs assigned to the “CARRIERBLD”. See Section 13.	SETTING UP	The equipment performs setup according to specifications of the next process program.	PROCESS-BLD-GROUP may include an AUTOSTART command within its body. Otherwise, the equipment waits for a START command.

8 Collection Event List

This section identifies data collection events and defines (Stream 6) reporting levels for variable items. The host can use the report definition scenario defined in SEMI E30 to define reports at ISEM-defined levels. The intent of this section is to ensure data is available at specific events and to optimize data reporting to the SECS-II host by allowing data to be grouped at reporting levels.

8.1 *Requirements*

8.1.1 This standard requires all collection events listed in the SEMI E30 standard. This standard requires the ISEM events in Table 2 for data collection (RunDataComplete, SubstrateDataComplete, GroupDataComplete, RegionDataComplete, and AnomalyDataComplete). These events are separate from the processing state transitions. These collection events shall occur before or on the processing state transition specified in Table 1. This was done to ensure that the data and the material remain synchronous. As a result, in some cases material processing may be delayed due to extended data processing time.

8.1.2 The most fundamental level of data defined for ISEM equipment is the anomaly level for review equipment and region level for inspection equipment. For example, review equipment has data available for individual anomalies at the AnomalyDataComplete event. Anomaly data may be grouped for level reporting. For example, data for anomalies found within a region on a substrate would be available at the RegionDataComplete event. This data would be available as a list variable item for Region Anomalies. All anomalies found on a substrate would be available at the SubstrateDataComplete event. This could either be 1) a list of list variable items for Region Anomaly, or 2) a single list variable item of all Substrate Anomalies. In this way, data can be reported with less high-level event reports, rather than as more low-level event reports.

8.1.3 Data produced by ISEM equip ment is customarily grouped for reporting by processing, material, and equipment constraints which are called reporting levels (i.e., run, substrate, group, site, and anomaly data). Level data is grouped by these constraints for a reporting level. Data shall be grouped within a reporting level according to other constraints by degree of processing (e.g., raw sensor, basic, or analyzed data), or statistically (e.g., summary, correlation, or comparison).

Table 2 Collection Events for ISEM Data Reporting

Reporting Level	*Data Collection Event*	*Inspection Equipment*	*Review Equipment*
Run	RunDataComplete	STOPPING → IDLE and LOAD → SETTING UP	STOPPING → IDLE and LOAD → SETTING UP
ProcessGroup	ProcessBuildGroup-Complete	LOAD → SETTING UP	LOAD → SETTING UP
Substrate	SubstrateDataComplete	UNLOAD → LOAD	UNLOAD → LOAD
Group	GroupDataComplete	INSPECT COMPLETE → ALIGN *or* WORKING → UNLOAD	REVIEW COMPLETE → ALIGN *or* WORKING → UNLOAD
Region	RegionDataComplete	INSPECT REGION → INSPECT SETUP	*Not Defined*
Anomaly	AnomalyDataComplete	*Not Defined*	CLASSIFY → REVIEW SETUP

NOTE 1: The data collection event shall occur before or on the processing state transition.

9 Variable Items

The purpose of this section is to define the list of variable item requirements for inspection and review equipment. Values of these variables shall be available to the host via collection event reports and host status queries. These variable items are separated into three categories: (a) common to all ISEM equipment; (b) specific to inspection equipment; (c) and specific to review equipment.

If equipment supports the data item functionality defined by ISEM, then it is required and shall be implemented as specified in Table 4 “Variable Item Dictionary”. That is, a variable item is only required if the equipment supports the functionality necessary to support it. For example, if an inspection instrument only has the hardware to count detected anomalies and lacks the hardware to determine their size, then the ISEM requires it to report anomaly count (e.g., as SubstrateAnomalyCount), but reporting anomaly size (as AnomalySize) is not required by the ISEM.

9.1 *Requirements*

- All variable items and data item restrictions defined in SEMI E30 are required on ISEM equipment.
- All variable items in the ISEM Variable Item Dictionary for specific equipment classifications are required for ISEM equipment. The data item restrictions are also required.

9.1.1 Variable items are categorized in the Variable Item Dictionary as follows:

- *Common Variables (CV)* — variables common to all ISEM equipment.
- *Inspection-Specific Variables (ISV)* — variables required only for inspection equipment.
- *Review-Specific Variables (RSV)* — variables required only for review equipment.

9.2 *Variable Items and Reporting Levels* — Table 3 defines reporting levels and associated Data Collection Events for which Variable Items are valid for.

Table 3 Variable Items and Reporting Levels

Level	*Reporting Level*	*Data Collection Event*
R	Run	RunDataComplete
P	ProcessGroup	ProcessGroupComplete
S	Substrate	SubstrateDataComplete
G	Group	GroupDataComplete
X	Region	RegionDataComplete
A	Anomaly	AnomalyDataComplete
ALL	Run, Substrate, Group, Region, and Anomaly	All of the above.

9.2.1 Variable items are documented in the ISEM Variable Item Dictionary using the following format:

Variable Name	*Type*	*Description*	*Level*	*Class*	*Format*	*Comments*

Where:

Variable Name: A unique name for the variable item (this name is for reference only).

Type: Defined as Common (CV), Inspection (ISV), Review (RSV), or Inspection/Review specific variables (IRSV).

Description: If class is DVVAL, then the description shall contain a statement of when data is valid in terms of ISEM events.

Level: The report level at which this variable is used <R|S|G|X|A|ALL> as defined in Table 3. It also indicates when the variable item is valid.

Class: The data type of the item.

Format: <SECS Message Language (SML) mnemonic> acceptable formats are SEMI E5 lists, ASCII, floating point, unsigned integer, or signed integer. A description of "ANY" indicates that only the above formats are acceptable and is left to the tool vendor to decide.

Comments: Any additional information pertinent to the variable name.

9.3 *Variable Item Types*

9.3.1 *Equipment Constants (ECV)* — The value can be changed by the host using S2,F15. The operator may have the ability to change some or all of the values. The value of an equipment constant may be queried at any time by the host using the S2,F13/14 transaction or Stream 6 reports.

9.3.2 *Status Variables (SV)* — The values are valid at all times. A SV may not be changed by the host or operator but may be changed by the equipment. A host or operator command may change an equipment status, thus changing an SV. The value of status variables may be queried by the host at any time using the S1,F3/4 or Stream 6 reports.

9.3.3 *Data Variables (DVVAL)* — These are variables which are valid upon the occurrence of a specific collection event and which may or may not be valid at other times, depending upon the equipment. An attempt to read a variable item when it is invalid will not result in an error, but the data reported may not have relevant meaning.

9.4 *Variable Item Dictionary*

9.4.1 *Data Validity* — The "Level" column in Table 4 defines when the variable item is valid. The entry in this column corresponds to a reporting level defined in Section 9.2 "Variable Items and Reporting Levels". For example, "RunAnomalyCount" is valid at the "RunDataComplete" event, and "AnomalySize" is valid at all reporting level data collection events.

Table 4 Variable Item Dictionary

Variable Name	*Type*	*Description*	*Level*	*Class*	*Format*	*Comment*
ActiveLocation	CV	The current carrier location that has substrates in the executing state of the processing state model.	ALL	SV	U2	Valid in all data collection events as defined in Table 3.
AlignList	CV	A list of alignment sites information being used by the current active process program.	ALL	DVVAL	L,n 1. <AlignName> : n.	The order in which the alignment name appears in the list is important and is equipment-dependent.
AlignName	CV	Alignment name	ALL	DVVAL	A[1..16]	An item in the AlignList variable.
AnomalyArea	CV	The area within the bounds of an anomaly (in units of $micron^2$).	ALL	DVVAL	F4	
Anomaly-Attributes	CV	Miscellaneous anomaly information that is equipment-dependent and defined by the equipment supplier.	ALL	DVVAL	L,n 1. <$attribute_1$> : n. <$attribute_n$>	Mainly used as part of other anomaly-related data (list) (i.e., AnomalySize).
Anomaly-Comment	RSV	Operator-generated comment associated with the anomaly.	A	DVVAL	A[1..80]	
AnomalyData2D	CV	Coordinate data for an anomaly.	A	DVVAL	L,5 1. <AnomalyID> 2. <CoordSys> 3. <Coord2D> 4. <Anomaly-Attributes>	
AnomalyID	CV	A unique anomaly identifier.	A	DVVAL	A[1..16]	
AnomalySize	CV	The X,Y extent of the anomaly in microns. The dimensions of the smallest rectangle that contains the anomaly whose sides are parallel to the X and Y axis.	A	DVVAL	L,2 1. <XExtent> 2. <YExtent>	XExtent and YExtent are of Format F4.
AnomalyTable-Name	CV	Name identifier of anomaly table.	ALL	SV	A[1..80]	
AnomalyTable-Type	CV	Type of anomaly table. (See Section 11.)	ALL	SV	A[1..20]	“TABLE-AREA-DEF”, “TABLE-ALIGN-DEF”, “TABLE-ANOMALY-DEF”, “TABLE-M21-ANOMALY-DEF”
BatchID	CV	The batch identification of the current material inspected/reviewed.	ALL	DVVAL	A[1..16]	
CarrierBuild	CV	ID of the CARRIERBLD ISEM job that the inspection/review data is associated with.	ALL	SV	A[1..80]	See Section 14.
CarrierID	CV	Physical identification of the current material inspected/reviewed.	ALL	DVVAL	A[1..16]	
CarrierNumber	CV	Used to identify carriers in	ALL	DVVAL	A[1..16]	

Variable Name	Type	Description	Level	Class	Format	Comment
		multi-lot runs (batch).				
Classification	RSV	Classification code of an anomaly.	A	DVVAL	A[1..80]	
Coord2D	CV	The two-dimensional coordinate of an anomaly.	ALL	DVVAL	L,2 1. <CoordX> 2. <CoordY>	
CoordSys	CV	The identification for a coordinate system definition. SEMI M20, M20P, or SEMI M21.	ALL	DVVAL	A[1..16]	"M20" "M21" "M20P"
CoordX	CV	The coordinate in the X direction of a site (anomaly, alignment site, or the lower left-hand of an area or element).	ALL	DVVAL	F4	
CoordY	CV	The coordinate in the Y direction of a site (anomaly, alignment site, or the lower left-hand of an area or element).	ALL	DVVAL	F4	
DefaultPriority	CV	The default priority given a material location if none is assigned.	ALL	EC	U2	
DeltaX	CV	The X-axis translation of M20P coordinate system relative to the SEMI M20 coordinate system.	ALL	DVVAL	F4	
DeltaY	CV	The Y-axis translation of M20P coordinate system relative to the SEMI M20 coordinate system.	ALL	DVVAL	F4	
ElementID	CV	The SEMI M21 address for a specific rectangular element on a substrate.	ALL	DVVAL	I4[2]	May refer to a field or die.
ElementList	CV	A list of SEMI M21 elements where processing can be attempted.	ALL	DVVAL	L,n 1. <ElementID> : n.	
Fiducial	CV	The physical feature used to associate a fiducial line used for orientation (i.e., flat or notch on a substrate).	ALL	DVVAL	A[1..16]	"FLAT" "NOTCH"
GroupAnomaly-Count	CV	Anomaly count for the current or last group (i.e., field or die inspected).	G	DVVAL	U2	
GroupArea	CV	Square Area (microns^2) of the last group inspected.	G	DVVAL	F4	
GroupComment	RSV	Operator-generated comment associated with the group.	G	DVVAL	A[1..80]	
GroupID	CV	Inspection/review group identification for the current inspection/review.	ALL	DVVAL	U2	
InspectionPPID	CV	Process program used on the inspection/review tool	ALL	DVVAL	A[1..80]	

Variable Name	*Type*	*Description*	*Level*	*Class*	*Format*	*Comment*
		for the current inspection/review.				
InspectionRunID	CV	Run identification in the current inspection/review.	ALL	DVVAL	A[1..16]	
LotID	CV	Lot identification of the current material inspected/reviewed.	ALL	DVVAL	A[1..16]	
M20Data	CV	The silicon substrate size, fiducial type, and orientation to use.	ALL	DVVAL	L,3 1. <SubstrateSize 2. <Fiducial> 3. <Orientation>	
M21Data	CV	The data necessary to establish an ISEM SEMI M21 layout on a substrate.	ALL	DVVAL	L,2 1. L,3 1. <M21XSize> 2. <M21YSize> 3. <Tile> 2. L,n 1. L,3 1. <ElementID> 2. <CoordX> 3. <CoordY> : n.	
M21XSize	CV	The value of the SEMI M21 coordinate system in the X-direction.	ALL	DVVAL	F4	
M21YSize	CV	The value of the SEMI M21 coordinate system in the Y-direction.	ALL	DVVAL	F4	
Offset	CV	The distance of the actual or found location of a site relative to its defined or expected location.	ALL	DVVAL	L[2]	Refers to SiteDeltaX, SiteDeltaY which are of Format F4.
OperatorAction	CV	The action taken by the operator on the equipment's operator I/O.	ALL	DVVAL	A[1..80]	
OperatorComment	CV	Operator-generated comment, not associated with any reporting level.	ALL	DVVAL	A[1..80]	(See also Run-Comment, Substrate-Comment, Area-Comment, RegionComment, Site-Comment, and AnomalyComment.)
OperatorID	CV	Identification of the operator of the inspection/review equipment.	ALL	DVVAL	A[1..16]	This information may be added by the host in the ISEM Tables. (See Section 11.)
Orientation	CV	How the wafer is loaded on the equipment.	ALL	EC	F4	"0" degrees indicates that the wafer has the primary fiducial towards the operator.
ProcessBuild-GroupID	CV	Name of the current or last process program executed.	ALL	SV	A[1..80]	See Section 14.
ProcessEquipmentID	CV	Identification of the process equipment used with the	ALL	DVVAL	A[1..16]	This information may be added by the host in

Variable Name	Type	Description	Level	Class	Format	Comment
		current material immediately prior to the inspection/review.				the ISEM Tables. (See Section 11.)
ProcessEquipmentLocation	CV	Location (code) of the process equipment used with the current material immediately prior to the inspection/review.	ALL	DVVAL	A[1..16]	This information may be added by the host in the ISEM Tables. (See Section 11.)
ProcessEquipmentPPID	CV	Identification of the process program used with the process equipment used on the current material immediately prior to the inspection/review.	ALL	DVVAL	A[1..80]	This information may be added by the host in the ISEM Tables. (See Section 11.)
ProcessLevel	CV	Identification of the processing level of the current material.	ALL	DVVAL	A[1..16]	This information may be added by the host in the ISEM Tables. (See Section 11.)
ProcessRunID	CV	Run identification for the process prior to current inspection/review.	ALL	DVVAL	A[1..16]	This information may be added by the host in the ISEM Tables. (See Section 11.)
ProductID	CV	The product identification of the current material inspected/reviewed.	ALL	DVVAL	A[1..16]	This information may be added by the host in the ISEM Tables. (See Section 11.)
RegionComment	CV	Operator-generated comment associated with the region.	X	DVVAL	A[1..80]	
RunAnomaly-Count	ISV	Total number of all anomalies found on all substrates in the last run.	R	DVVAL	U2	
RunComment	CV	Operator-generated comment associated with the run.	ALL	DVVAL	A[1..80]	
RunInspected-AreasCount	ISV	The total number of inspected/reviewed areas on all substrates in the last run.	RS	DVVAL	U2	
RunInspection-PPCount	ISV	The total number of process programs used for the current or last run.	RS	DVVAL	U2	
RunSubstrate-Count	CV	The total number of substrates completed in the current inspection run, which remains valid until the next START command.	ALL	DVVAL	U2	
ScaleFactor	CV	A correction factor applied to the translation of one coordinate system to another.	ALL	DVVAL	F4	In most cases, a scaling of 1 (one) is expected.
SiteID	CV	Inspection/Review group identification for the current site inspection/review.	ALL	DVVAL	U2	
SlotID	CV	Carrier slot number from which the current substrate was taken.	ALL	DVVAL	U2	

Variable Name	*Type*	*Description*	*Level*	*Class*	*Format*	*Comment*
SlotList	CV	The list of carrier slots with substrates to be processed.	ALL	DVVAL	L,n 1. <SlotID> : n.	
SubstrateAnomalyCount	ISV	The total number of anomalies for the current substrate.	S	DVVAL	U2	For the most recent inspection.
Substrate-Comment	CV	Operator-generated comment associated with the substrate.	ALL	DVVAL	A[1..80]	
SubstrateID	CV	Substrate identification for the current inspection/review.	ALL	DVVAL	A[1..16]	
Substrate-InspectedAreas-Count	ISV	The total number of inspected areas on the current substrate.	S	DVVAL	U2	
SubstrateRegion-Count	CV	Total area count for the current or last substrate inspected.	ALL	DVVAL	U2	
SubstrateSize	CV	The nominal diameter (in mm) of the current or last substrate inspected/reviewed.	ALL	DVVAL	U2	
SubstrateTotal-AreaInspected	ISV	Total square area inspected/reviewed (micron^2) of the current substrate.	S	DVVAL	F4	
Theta	CV	The rotational difference in radians between a primary and secondary coordinate system.	ALL	DVVAL	F4	
Tile	CV	The layout of the pattern in the substrate.	ALL	DVVAL	A[1..16]	"NTILE" non-tiled, "CTILE" column-tiled, and "RTILE" row-tiled.
XLateData	CV	Variable for the equipment to report offset of the found or actual pattern-based coordinate system relative to the substrate-based coordinate system on the substrate being tested.	ALL	DVVAL	L,4 1. <DeltaX> 2. <DeltaY> 3. <Theta> 4. <ScaleFactor>	

10 Alarm List

Since each model of equipment differs in configuration, it is not practical to provide an exhaustive list of all possible alarms. Instead, the ISEM is requiring the two tables provided as described in SEMI E30 (document section). Alarm List Table, which is intended to provide for equipment configuration-specific alarms and Alarm ID, Alarm Set/Cleared Event Table.

10.1 *Alarm List Table*

10.1.1 The alarm list table contains examples of alarms that pertain to various configuration aspects of equipment. These examples are intended to illustrate that alarms pertain to situations in which there exists a potential for exceeding physical safety limits associated with people, equipment, and material being processed as per the SEMI E30 definition of an alarm. (See SEMI E30 for further reference.)

10.2 *Alarm ID, Alarm Set/Cleared Event Table*

10.2.1 The Alarm ID, Alarm Set/Cleared Event Table documents the association of each ALID to a set and cleared event as required by SEMI E30. (See SEMI E30 for further reference.)

11 ISEM Tables

A fundamental requirement of ISEM equipment is to transfer anomaly and review data between itself and the host. ISEM equipment may also be required to transfer "Area" and "Alignment Site" data needed for run setup. Anomaly and review data sets (as well as area and alignment site data sets) are commonly handled as lists and tables. ISEM equipment shall use tables when transferring this kind of data between itself and the host. List shall be used to refer to sub-sets of this table data.

11.1 *ISEM Table Data*

11.1.1 ISEM Tables are used to specify area, alignment site, and anomaly coordinate lists for ISEM equipment. ISEM Tables are transferred between the host and the equipment using SECS-II Stream 13 messages (Unformatted Data Set Transfers). For example, an ISEM Table may be used to transfer anomaly data (e.g., "M21" coordinates) generated by inspection equipment to the host, which in turn may then be transferred to review equipment from the host. ISEM Tables also include attributes items that are associated with the table, not with the table data. ISEM Table attributes are used to include information associated with table data, like the number of columns (NumCols), number of rows (NumRows), and table size (DataLength). (See SEMI E58 for additional information.) Product or process-related information may also be included on the attribute section of the ISEM Tables (e.g., LotID, ProductID, OperatorID, ProcessEquipmentID) (see Table 4). The ISEM does not specify additional table attribute variable items that may be associated with the table.

11.2 *TABLE-DEFs*

11.2.1 ISEM Tables are documented using TABLE-DEF structures (Figure 5). Each TABLE-DEF structure has a unique name (TableID) and type (TableType). Each column in the TABLE-DEF has a name (e.g., "AREANAME", "ALIGNNAME", OR "ANOMALYID"); row names are specific instances or values that correspond to the column headers in TABLE-DEF. A specific TABLE-DEF row is designated by referring to the TABLE-DEF name and the specific row name. ISEM Tables are transferred using standard (SECS-II message) Stream 13 messages (Figure 6), and each ISEM-defined TABLE-DEF item maps into a Stream 13 message item. Align and area data tables are host-defined, and anomaly tables are equipment-defined.

TableType = <*TableDef*> TableID = <*TableName*>

Row Name	*$Column_1$*	*$Column_2$*	*$Column_3$*	…	*$Column_n$*
Row_1 Name					
…					
Row_m Name					

Figure 5
TABLE-DEF Structure

An example of usage of S13,F13 to transfer data sets is shown below:

```
L,7
   1. <DATAID>
   2. <OBJSPEC>
   3. "TableDef"                        <TBLTYP>
   4. "TableName"                       <TBLID>
   5. L,n                              # of table attributes
        1. L,2
                1. "NumRows"            <ATTRID1>
                2. <m>                  <ATTRDATA1>
        2. L,2
                1. "NumCols"            <ATTRID2>
                2. <n>                  <ATTRDATA2>
        3. L,2
                1. "DataLength"         <ATTRID3>
                2. <table length>       <ATTRDATA3>
        4. L,2
                1. "LotID"              <ATTRID4>
                2. "ABC123"             <ATTRDATA4>
           n. L,2
                1. "ProductID"          <ATTRIDn>
                2. "CPUTYPE"            <ATTRDATAn>
        6. L,n                          # of columns
             1. "AREANAME"              <COLHDR1>(1st column description)
             .
             n. "ATTRIBUTE5"            <COLHDRn>(nth column description)
        7. L,m                          # of rows
             1. L,n                     # of columns
                1. <Item 1,1>           table item in row 1, column 1
                .
                n. <Item 1,n>           table item in row 1, column n
                .
             m. L,n                     # of columns
                1. <Item m,1>           table item in row m, column 1
                .
                n. <Item m,n>           table item in row m, column n
```

Figure 6
S13,F13 with ISEM TABLE-DEF Data

11.3 *Required ISEM Tables* — ISEM equipment shall support all three table types: area, align, and anomaly. SEMI M21 anomaly table type may be supported, but it is optional. The ISEM equipment shall be able to store simultaneously at least 3 (three) defined tables of each type supported to guarantee the validity of any table while that table is being transferred (a table transfer transaction is in process).

ISEM align and area tables are stored by the equipment during the current inspection or review run (i.e., until a new remote command PP-SELECT or PP-ASSIGN is sent, or until they are modified with PP-UPDATE).

Table 5 ISEM Table Types (TABLE-DEFs)

Table Type	*Req/Opt*	*Description*
"TABLE-AREA-DEF"	R	*Area Definition* — A set of areas and their attributes, typically the list of areas to be inspected.
"TABLE-ALIGN-DEF"	R	*Alignment Site Definition* — A set of alignment sites and their attributes.
"TABLE-ANOMALY-DEF"	R	*Anomaly Coordinate Data Definition* — A set of anomalies and their attributes, with coordinates given in the SEMI M20 or M20P coordinate system.
"TABLE-M21-ANOMALY-DEF"	O	*Anomaly SEMI M21 Coordinate Data Definition* — A set of anomalies and their attributes, with coordinates given in the SEMI M21 coordinate system.

As indicated in Table 5, the SEMI M21 anomaly definition table “TABLE-M21-ANOMALY-DEF” is optional. The others are required.

ISEM requires that the following columns be included in the TABLE-DEFs. Table 6 defines the column headers and the allowed formats. Anomaly attributes and attribute headings are defined by the supplier, based on equipment capability.

11.3.1 “TABLE-AREA-DEF”

TableType: “TableAreaDef” TableID: <*AreaTableName*>

Area Name	*Coordx*	*Coordy*	*Coordsys*	*Xtentx*	*Xtenty*	*Attribute (1)*	*...*	*Attribute*

11.3.2 “TABLE-ALIGN-DEF”

TableType: “TableAlignDef” TableID: <*AlignTableName*>

AlignName	*Coordx*	*Coordy*	*Coordsys*	*Attribute (1)*	*...*	*Attribute(N)*

11.3.3 “TABLE-ANOMALY-DEF”

TableType: “TableAnomalyDef” TableID: <*AnomalyTableName*>

Anomalyid	*Coordx*	*Coordy*	*Coordsys*	*AnomalyAttribute*	*...*	*AnomalyAttribute*

11.3.4 “TABLE-M21-ANOMALY-DEF”

TableType = “TableM21AnomalyDef” TableID = <*M21AnomalyTableName*>

Anomalyid	*Coordx*	*Coordy*	*Elementid*	*AnomalyAttribute*	*...*	*AnomalyAttribute*

Defect data shall be transferred between the host and inspection/review/analysis equipment using SEMI E5, S13,F13. Columns in the table are defined by Table 11.3.5 below.

11.3.5 TABLE-STANDARD-DEFECT-DATA-SET-DEF

NOTE 1: Data for each substrate should be reported in column order shown below. Inspection tools should support relevant columns (see NOTE 6). Review and analysis tools should support all columns. Multiple data entries (list format data items) are allowed for a given attribute on a single defect.

NOTE 2: The inspection equipment must add a table attribute called “Substrate Header” (ATTRID). It must be a list that includes the following items in the given order: LotID (A[1..16]), SubstrateID (A[1..16]), ProcessEquipmentID (A[1..16]), substrate center[1] (L[2], CoordX (F4), CoordY (F4)) and centering method[2] (A[1..16]). Refer to Table 4, Variable Item Dictionary for descriptions of these items.

NOTE 3: See SEMI M21 for (0,0) die location methodology.

1 Vector from the origin of the substrate coordinate system to nominal substrate center location.
2 CoordSys (e.g., “SEMI M20”, “SEMI M21”, “M21P”, etc).

NOTE 4: Die origin is located at lower left-hand corner (LLHC).

NOTE 5: Data format must comply with the ISEM standard.

NOTE 6: In order to signify tool context, anomaly attributes are labeled as follows:

Column name starts with “insp*_” for inspection data, “rev*_” for defect review data, “and “anal*_” for analysis data, where “*” is a numeric string that ensures each set of columns added is uniquely named (e.g., “rev1_” and “rev2_”).

NOTE 7: Inspection, review and analysis tools must add a table attribute called “insp*_ Header,” “rev*_Header” and “anal*_Header” respectively (ATTRID) each time they add data to the table.[3] It must be a list that includes the following items in the given order: EquipmentID (A[1..16]), EquipmentType (A[1..16]), OperatorID (A[1..16]), and CLOCK[4] (A[16]).

Column #	*Column Name*	*Description*
1	Insp_Anomaly ID	ID # for the defect
2	Insp_Table specifier	Specifies table with other relevant information
3	Insp_Coordinate X	Intra die X Coordinate wrt LLHC of die in um
4	Insp_Coordinate Y	Intra die Y Coordinate wrt LLHC of die in um
5	Insp_X index	X axis die index wrt center of wafer (COW)
6	Insp_Y index	Y axis die index wrt center of wafer (COW)
7	Insp_X size	Defect size along X axis in microns
8	Insp_Y size	Defect size along Y axis in microns
9	Insp_Defect area	Defect area in square microns
10	Insp_Defect size	Linear measure of defect size in microns
11	Insp_Scatter intensity	Anomaly scattering intensity
12	Insp_Defect class number	Previously defined class number assigned to the defect
13	Insp_Test number	Inspection test in which defect was found
14	Insp_# Optical image count	Number of optical images stored for a given defect
15	Insp_Optical image data	Optical image data specifier
16	Insp_Cluster	= 1 if defect is part of a systematic defect cluster
17	Insp_Cluster class	Systematic defect class name
18	Sampled for SEM	= 1 if defect chosen for SEM review
19	Rev_SEM image data	SEM image data specifier
20	Rev_SEM class	SEM defect class name
21	Rev_Defect height	Defect height in microns
22	Sampled for analysis	= 1 if defect chosen for EDX, = 2 if defect chosen for FIB or other analysis
23	Anal_EDX data	EDX data specifier
24	Anal_FIB data	FIB data specifier

3 Where “*” is a numeric string that corresponds to the one in the column names the header refers to.

4 Date and time of the start of inspection, review, or analysis per SEMI E5 CLOCK data item variable.

11.4 TABLE-DEF Column Header Descriptions and Formats

Table 6 Description and Formats for ISEM Table Data

Column Header	*Description*	*Format*	*Comments*
"ALIGNNAME"	The identifier given to an alignment site.	A[1..16]	
ANOMALYATTRIBUTE (n) NOTE: String defined by equipment supplier.	Tool-specific information associated with an ANOMALY for which no specific ISEM data item has been defined.	U2, F4, F8, A[1..16]	Examples: Include information such as magnification, voltage, current, wavelength, brightness, color, height, or chemical spectra. The equipment supplier shall document all attributes that are supported.
"ANOMALYID"	A unique identifier for an anomaly.	A[1..16]	
"AREANAME"	A unique identifier given to an inspection area.	A[1..16]	
ATTRIBUTE(n) NOTE: String defined by equipment supplier.	Tool-specific information associated with an alignment or measurement site for which no specific ISEM data item has been defined.	U2, F4, F8, A[1..16]	Examples: Include information such as magnification, voltage, current, wavelength, number of scans, integration time, or film stack. The equipment supplier shall document all attributes that are supported.
"COORDSYS"	The identification for applicable coordinate system.	A[1..16]	Options are "M20", "M20P", and "M21".
"COORDX"	The x-coordinate for a site.	F4	Units are in microns.
"COORDY"	The y-coordinate for a site.	F4	Units are in microns.
"ELEMENTID"	The SEMI M21 address for a specific rectangular element on a substrate.	I4[2]	
"XTENTX"	The extent in the X-direction of an area to inspect as measured from the lower left-hand corner of the area given by CoordX.	F4	Units in microns.
"XTENTY"	The extent in the Y-direction of an area to inspect measured from the lower left-hand corner of the area given by CoordY.	F4	Units in microns.

12 Process Program Management

12.1 *Definition and Rules for ISEM Process Programs*

12.1.1 A process program contains information and/or instructions required for the Inspection/Review equipment to process a given run of material. The process program shall supply all of the information required for a remotely executed run to be processed without operator intervention.

12.2 *Requirements*

12.2.1 The ISEM requires that the SEMI E30 capability of Process Program Management be fully supported for this class of equipment. ISEM requires that the process program have a structure that enables the user to build process programs with default conditions that can be overridden for a run. ISEM requires the ability to vary the quantity of substrates processed, the alignment information used, and the number and/or location of the areas/anomalies to be inspected/reviewed through the uses of process program variable parameters. The concepts of process program structure and process program variable parameters are discussed in the following sections.

12.3 *Process Program Structure*

12.3.1 The purpose of this process program structure and the related concepts is to provide flexibility in using process programs to reduce the number of process programs needed. This structure enables the user or host to vary certain parameters of a given process program as needed for any particular run.

12.3.2 Often a process program may be very similar from one run to another and may differ only in a few parameters such as: which substrate slots to run, which areas to inspect, which parameters to run on each substrate, etc. Previously this small variation from run to run would require a large number of process programs to be created and maintained. The flexibility of the method described in this section will reduce the number of process programs.

12.4 *Process Program Variable Parameters* — A process program parameter specifies a value that temporarily modifies the value of a process program variable parameter. A process program variable parameter is formally defined within a process program body and contains (1) a variable parameter name that is unique in the body (CPNAME), and (2) a parameter default value for use when the process program is selected for execution without specification of an override value for this variable parameter (CPVAL/CPEVAL).

12.4.1 *Overriding Process Program Variable Parameters Default Values*

12.4.1.1 Any process-related information that is normally requested from the operator console in manual operation shall have a process program variable parameter identified in the process program and default values assigned in the body of the process program. An equipment would run the process program using the default values unless those values were overridden.

12.4.1.2 These process program variable parameters allow a host to tailor a process program for a specific run of material by temporarily modifying (replacing) the process program default values using a remote command of PP-UPDATE. The modification does not permanently change the process program; the modifications remain in effect only until the next run or until the next PP-UPDATE remote command is received.

12.4.2 *Requirements and Rules*

12.4.2.1 ISEM equipment is required to support variable process program parameters. Additionally, ISEM process programs are required to contain variable process program parameters that specify a name for each of the four previously defined ISEM table types. Specifically, parameters for "TABLE-AREA-DEF", "TABLE-ALIGN-DEF", "TABLE-ANOMALY-DEF", and "TABLE-M21-ANOMALY-DEF" table names are required. Only the names that refer to these TABLE-DEFs are required to be included in the process program body. The actual TABLE-DEF data is external to the process program body. The host may always assume that there are variable process program parameters for these four ISEM tables.

12.4.2.2 Before execution of a CARRIERBLD can begin, the presence of all the ISEM Tables that it references shall be verified by the equipment. If they are not all present, an error shall be reported. The equipment shall support data items that may be linked to the event report that specifies the name of missing ISEM Tables. S7,F27 is used for reporting this error condition.

12.4.2.3 The following table summarizes the variable process program parameters that ISEM equipment shall support and the remote command parameters that the host may use to override their values (as defined in Section 13 and Table 9).

Table 7 Required Variable Process Program Parameters

Variable Process Program Parameter and Host Command Parameter Name (CPNAME)	*Description*
"ALIGNLIST"	A list of location identifiers to be reviewed.
"ANOMALYLIST"	A list of location identifiers to be reviewed.
"AREALIST"	A list of area identifiers to be inspected or reviewed.
"ELEMENTLIST"	A list of array element identifiers to be inspected.
"SLOTLIST"	A list of carrier slot numbers with material to be inspected or to be reviewed.
"SUBSTRATELIST"	A list of substrate IDs with material to be inspected or to be reviewed.
"TABLE-ALIGN-DEF"	A set of alignment site definitions.
"TABLE-ANOMALY-DEF"	A set of (SEMI M20) anomaly location and attribute definitions.
"TABLE-AREA-DEF"	A set of area definitions.
"TABLE-M21-ANOMALY-DEF"	A set of SEMI M21 anomaly location and attribute definitions.

12.4.3 *Modifying Process Program Variable Parameters* — The remote commands of PP-SELECT, PP-ASSIGN, or PP-UPDATE are used to modify any of the identified process program variable parameters within the process program. The modification is done by including CPNAME/CPVAL pairs within the "PROCESS-BLD-GROUP", which is part of the remote commands of PP-SELECT or PP-ASSIGN, or by including a different list name in the PP-UPDATE remote command. A CPNAME in a process program shall be identical to the process program variable parameter name as specified in the "PROCESS-BLD-GROUP". See next section for details of these parameters.

12.5 *Use of Process Programs, Remote Commands, and "PROCESS-BLD-GROUP"* — This is a brief description of the steps involved in using the process program structure, the process program variable parameters, the TABLE-DEFs, and the modification of process program variable parameters through the use of "PROCESS-BLD-GROUP" and CPNAME/CEPVAL pairs in the enhanced remote command (S2,F49).

- A process program is created with certain items in the body identified as process program variable parameters, along with their default values.
- The host sends to equipment one or more TABLE-DEFs: such as "TABLE-AREA-DEF" and "TABLE-ALIGN-DEF", along with the names of those tables using Stream 13 commands (See SEMI E5). These tables are now resident on the equipment.
- The host sends an enhanced remote command (S2,F49) to the equipment (either PP-SELECT or PP-ASSIGN) that contains the information needed for processing ("CARRIERBLD"). The "CARRIERBLD" contains a "PROCESS-BLD-GROUP" for each different set of run parameters that are needed during the inspection or review process.
- "PROCESS-BLD-GROUP" shall contain the ProcessBuildGroupID to identify which program is selected.
- If the default list of slots in the process program needs to be changed for this run, then "PROCESS-BLD-GROUP" contains a "MATERIALLIST" and the list of selected slots or substrates, designated by SlotIDs or SubstrateIDs.
- If the default set of inspection areas in the process program needs to be changed for this run, then "PROCESS-BLD-GROUP" contains the name of a specific "TABLE-AREA-DEF" (in AreaTable Name). If not all of the inspection areas given in that "TABLE-AREA-DEF" are needed for this run, then "AREALIST" is used and includes a list of the names of the specific inspection areas which are needed. Those names refer to inspection areas defined in the "TABLE-AREA-DEF".
- If the default set of alignment sites in the process program needs to be changed for this run, then "PROCESS-BLD-GROUP" contains the name of a specific "TABLE-ALIGN-DEF" (in AlignTable Name). If not all of the alignment sites given in that "TABLE-ALIGN-DEF" are needed for this run, then "ALIGNLIST" is used and includes a list of the names of the specific alignment sites which are needed. Those names refer to alignment sites defined in the "TABLE-ALIGN-DEF".
- If the SEMI M21 coordinate system is being used and a pattern has been defined with pattern element names and if only certain of those elements need to be inspected for this run, then "PROCESS-BLD-GROUP" contains an "ELEMENTLIST" and the list of selected elements, designated by ElementIDs.
- If certain equipment-specific process program variable parameters need to have different values for this run, then for each needed parameter, "PROCESS-BLD-GROUP" contains a CPNAME (unique name of a specific process program variable parameter) and the new CEPVAL (new value for that parameter).
- The equipment executes the process program with the new values.
- If indicated in the process program, the equipment generates the list of anomalies found and sends it to the host in the format of a "TABLE-ANOMALY-DEF" using the ISEM table definition and Stream 13 transfer messages (see SEMI E5 for format specification).
- The host would send the "TABLE-ANOMALY-DEF" to a review equipment. The host might need to modify part of the table if required by the review equipment. This anomaly table is now resident on the review equipment. (NOTE: The host might choose to not send all of the table of anomalies, but rather a desired selection of them.)
- The host sends the equipment the enhanced remote command (PP-SELECT or PP-ASSIGN) that contains a "CARRIERBLD". The "CARRIERBLD" has a "PROCESS-BLD-GROUP" for each different set of run parameters that is needed by the review equipment. It is required for the host to use the Enhanced Remote Command S2,F49 for transferring the information

to the equipment. "PROCESS-BLD-GROUP" includes the name of a specific "TABLE-ANOMALY-DEF" in the parameter AnomalyTableName. If not all of the anomalies given in that "TABLE-ANOMALY-DEF" are needed for this run, then either "ANOMALYLIST" or "M21-ANOMALYLIST" is used and includes a list of the names of the specific anomalies which are desired. Those names refer to anomalies defined in the named "TABLE-ANOMALY-DEF".

- The equipment runs the process program using the selected values and reviewing the specific anomalies indicated.
- The equipment adds review information to the ANOMALYATTRIBUTE list. Then the review equipment sends this modified "TABLE-ANOMALY-DEF" to the host.

13 Remote Commands

The purpose of this section is to identify remote commands, command parameters, and valid commands versus states pertinent to the SEM.

13.1 *Requirements*

- The equipment shall support the SEMI E30 required remote commands.
- All the remote commands defined by ISEM are required unless they have been qualified by the statement "if the equipment supports this functionality, it shall use this command." In this case, they are only required if the equipment supports the functionality necessary to support the command. A good example of this is the MAP-CARRIER command. If the equipment does not have the hardware necessary to scan a carrier for the presence of substrates in slots, then the command is not required by the ISEM.
- The alphanumeric strings defined by ISEM for RCMD and CPNAME are required.

Host Command Parameter (CPNAME/CPVAL) — A parameter name/value associated with a particular host command when using stream function (S2,F41) and a (CPNAME/CEPVAL) parameter name/value when using the enhanced remote command (S2,F49). This document specifies unique names (CPNAMEs) and values (CPVALs and CEPVALs) for many command parameters. Note that if there are no associated parameters, a zero length list is sent.

The purpose of the remote commands is to allow host control over the following capabilities:

- Start processing
- Stop processing
- Temporarily suspend processing
- Resume processing
- Abort processing
- Select process programs, material, and/or sites to measure
- Report location of material found

The following remote commands (RCMDs) shall be supported as described below:

NOTE 3: The terms "current cycle" and "safe point" used below are to be defined by the supplier.

13.2 *Remote Commands Description*

1. *ABORT* — Terminate the current cycle prior to its completion. ABORT has the intent of immediately stopping the process and is used because of abnormal conditions. ABORT makes no guarantee about the subsequent condition of material except as noted in the "ABORTLEVEL" description.
2. *CLEANUP* — De-selection of the current ISEM job ("CARRIERBLD") and process program ("PROCESS-BLD-GROUP"), including the removal of all material to output locations and any equipment-specific activities needed to transition into the IDLE state. Completion of this command should generate a collection event report. If the equipment supports this functionality, it will use this command.
3. *MAP-CARRIER* — Requests the equipment to provide a list of carrier slots that contain material. MAP-CARRIER has the intent of providing the host with enough information about the location and/or ID of material so it may select material for processing accordingly. Completion of this command shall generate a collection event report. If the equipment supports this functionality, it must use this command.
4. *NEXT-MATERIAL* — Processing of the current substrate is halted at the first safe point and unloaded to the target carrier location. NEXT-MATERIAL has the intent of allowing the host to skip measurement of the current substrate. This is a trigger for processing state transition from WORKING to UNLOAD. If the equipment supports this functionality, it will use this command.
5. *PAUSE* — Suspend processing temporarily at the next safe point. PAUSE has the intent of resuming the process at the same point where it was paused.

RESUME or PP-UPDATE may be used to resume the process.

6. *PP-ASSIGN* — Instructs the equipment that supports queuing to create a new ISEM job ("CARRIERBLD") for the specified port ("LOCATIONID") when more than one port is available for processing. If only one port is available, "LOCATIONID" is not required. Priority may optionally be specified with this command. The "PRIORITY" specifies the priority of the newly created job in the ISEM job queue (a value of 0 (zero) assigns the highest priority to the job). Without specifying a priority, the job is queued with the default priority. Jobs with equal priority are queued in the order the PP-ASSIGN commands are received. This command is valid in all PROCESSING states.

7. *PP-SELECT* — Instructs the equipment to make the requested ISEM job(s) ("CARRIERBLD") available in the execution area. This is a trigger for the processing state transition from IDLE to SETTING UP. The first process program ("PROCESS-BLD-GROUP") specified in the "CARRIERBLD" is also validated during SETTING UP.

8. *PP-UNASSIGN* — Removes the ISEM job assignment ("CARRIERBLD") for a carrier or port. The carrier or port is removed from the process queue.

9. *PP-UPDATE* — Provides the ability to alter the current process program being executed during the PAUSED state. The process program variables specified in the PP-UPDATE command will replace previous definitions in the "PROCESS-BLD-GROUP". This command will trigger transition to CHECKING for process program parameter verification. A RESUME command is implied with the validation of "all" replaced values to resume the process. If the PP-UPDATE fails, the process program variables present prior to the PP-UPDATE are retained. If no parameters values are specified, the defaults are used.

10. *RESUME* — Resume processing from the point where the process was paused. This is the trigger for processing state transition from PROCESS PAUSE to the previous PROCESS state.

11. *START* — Instructs the equipment to initiate processing. This is the trigger for the processing state transition from READY to LOAD. An "AUTOSTART" command parameter may be included to allow for continuous processing.

12. *STOP* — Complete the current cycle, stop in a safe condition, and return to the IDLE processing state. Stop has the intent of stopping the process entirely. This command can be used to both: stop the current ISEM job or to stop all queued jobs. The equipment is not required to support the continuation of processing.

13.2.1 *Remote Commands and Associated Host Command Parameters* — This table describes the allowable command parameters (CPNAME) for each remote command (RCMD). Equipment shall support all parameters. The column marked Req/Opt specifies which parameters are required to be sent by the host and which parameters may be optionally sent by the host.

Table 8 Allowable Command Parameters

Remote Command	*Parameters*		
	CPName	*Req/Opt*	*Comments*
ABORT	"ABORTLEVEL"	R	
CLEANUP	"CARRIERID" "LOCATIONID" "SLOTID"	O O O	PORT and SLOT may be used to define a different carrier/slot destination for the substrates.
MAP-CARRIER	"CARRIERID" "LOCATIONID"	R* R*	* One is required.
NEXT-MATERIAL	"CARRIERID" "LOCATIONID" "SLOTID"	O O O	PORT and SLOT may be used to define a different carrier/slot destination for the substrates.
PAUSE	None	NA	None
PP-ASSIGN	"PRIORITY" "CARRIERBLD"*	O R	* More than one "CARRIERBLD" may be specified.
PP-SELECT	"CARRIERBLD"*	R	* More than one "CARRIERBLD" may be specified.
PP-UNASSIGN	"CARRIERBLD"	R	None

PP-UPDATE	"PPBUILDID" "ALIGNLIST" "ANOMALYLIST" "AREALIST" "ELEMENTLIST" "SLOTLIST" "SUBSTRATELIST" "TABLE-ALIGN-DEF" "TABLE-AREA-DEF" "TABLE-ANOMALY-DEF" "TABLE-M21-ANOMALY-DEF"	R R* R* R* R* R* R* R* R* R* R*	* At least one is required.
RESUME	None	N/A	None
START	"CARRIERBLD"	0	None
STOP	"CARRIERBLD"	0	None

13.2.2 *Host Command Parameter Names and Values*

Table 9 Host Command Parameters CPNAMES

CPName	*Parameter Value*		
	Description	*Range*	*Format*
"ABORTLEVEL"	ISEM-defined abort levels: HALT — Process halts, and the ABORTING process state is entered. CLEANUP — Process halts, material cleanup is performed, and the ABORTING process state is entered.	"1= HALT" "2 = CLEANUP"	U2
"ALIGNLIST"	L,n 1. $AlignName_1$: n. For the SEMI M20 or M20P coordinate system.		List of A[1..16] data items
"ALIGNNAME"	Alignment name See the "TABLE-ALIGN-DEF" definition for further explanation.		A[1..16]
"ANOMALYLIST"	L,n 1. $AnomalyID_1$: n. For the SEMI M20 or M20P coordinate system.		List of A[1..16] data items
"ANOMALYID"	Anomaly identifier See the "TABLE-ANOMALY-DEF" or the "TABLE-M21-ANOMALY-DEF" definition for further explanation.		A[1..16]
"AREALIST"	L,n 1. $AreaName_1$: n.		List of A[1..16] data items
"AREANAME"	Unique identifier for an area to be inspected. See the "TABLE-AREA-DEF" definition for further explanation.		A[1..16]
"AUTOSTART"	Specifies whether a START command is required from an external source (operator or host) to exit the READY state. 0 = NoAutoStart (A START command required.) 1 = AutoStart (No external START command required to begin execution.)	0–1	U2
"CARRIERID"	Identifier of the carrier that the inspection/review data is associated with.		A[1..16]

CPName	Parameter Value		
	Description	*Range*	*Format*
"ELEMENTLIST"	L,n 1. ElementID$_1$: n. For the SEMI M21 coordinate system.		List of A[1..16] data items
"LOCATIONID"	Unique identifier of the location to be used for the "CARRIERBLD" assignment.		U2
"PPBUILDID"	ProcessProgramBuildID		A[1..80]
"PPNAME"	ProcessProgramName		A[1..80]
"PRIORITY"	Assignment priority	0–9 Highest priority corresponds to 0.	U2
"SLOTLIST"	Specifies carrier slots containing substrate for the ISEM job. L,n 1. SlotID$_1$: n. SlotIDn	Zero length list specifies all slots.	List of U2 data items
"STOPLEVEL"	Stop levels defined by the ISEM.	"1 = LOCATIONID" "2 = CARRIERID"	Use defined CPVALs
"SUBSTRATELIST"	Specifies identifiers of substrate for the ISEM job. L,n 1. SubstrateID$_1$: n. SubstrateID$_n$	Zero length list specifies all substrate (independent of substrate identifier).	List of A[0..16] data items
"CARRIERBLD" E5 Format	L, 3 Three ISEM required parameters 1. L, 2 ❖ 1. "CARRIERID" **A[9]** -- CPName 2. CarrierID **A[1..16]** -- CPValue 2. L, 2 ❖ 1. "LOCATIONID" **A[10]** -- CPName 2. LocationID **U2** -- CPValue 3. L, 2 1. "PROCESS-BLD-LIST" 2. L, n ❖ List of n jobs 1. L, 2 1. "PROCESS-BLD-GROUP" 2. L, m First ISEM job 2. L, 2 1. "PROCESS-BLD-GROUP" 2. L, m Next ISEM job . . . n. L, 2 1. "PROCESS-BLD-GROUP" 2. L, m Last ISEM job	m = 3	List
"PROCESS-BLD-GROUP" (for Inspection)	L,m 1. L,2 1. "PPBUILDID" 2. ProcessBuildGroupID	*m* ≥ 2	List of *m* data items

CPName	Parameter Value		
	Description	*Range*	*Format*
	2. L,2 1. "PPNAME" 2. ProcessProgramID 3. L,2 1. "SLOTLIST" 2. L,n 1. SlotID$_1$: n. ***Or*** 1. "SUBSTRATELIST" 2. L,n 1. SubstrateID$_1$: n. 4. L,2 1. "AUTOSTART" 2. AutoStart 5. L,2 1. "TABLE-AREA-DEF" 2. AreaTableName 6. L,2 1. "AREANAME" 2. L,n 1. AreaName$_1$: n. 7. L,2 1. "TABLE-ALIGN-DEF" 2. AlignTableName 8. L,2 1. "ALIGNNAME" 2. L,n 1. AlignName$_1$: n. 9. L,2 1. "ELEMENTLIST" ** 2. L,n 1. ElementID$_1$: n. 10. L,2 1. CPNAME* 2. CEPVAL* m. L,2 1. CPNAME* 2. CEPVAL* **NOTES:** "PPBUILDID" and "PPNAME" are required. "SLOTLIST", "SUBSTRATELIST", "AREALIST", and "ALIGNLIST" are optional. * Supplier shall define as many of these CPNAME, CEPVAL		

<table>
<tr><th rowspan="2">CPName</th><th colspan="3">Parameter Value</th></tr>
<tr><th>Description</th><th>Range</th><th>Format</th></tr>
<tr><td></td><td>pairs as are supported by the equipment.
** “ELEMENTLIST” is required when using the SEMI M21 coordinate system in the definition of an AlignName or AreaName.</td><td></td><td></td></tr>
<tr><td>“PROCESS-BLD-GROUP” (for Review Equipment)</td><td>L,m
1. L,2
1. “PPBUILDID”
2. ProcessBuildGroupID
2. L,2
1. “PPNAME”
2. ProcessProgramID
3. L,2
1. “SLOTLIST”
2. L,n
1. SlotID$_1$
:
2. SlotID$_n$
Or
1. “SUBSTRATELIST”
2. L,n
1. SubstrateID$_1$
:
n. SubstrateID$_n$
4. L,2
1. “AUTOSTART”
2. AutoStart
5. L,2
1. “TABLE-ANOMALY-DEF”
2. AnomalyTableName
6. L,2
1. “ANOMALYNAME” or “M21-ANOMALYNAME”
2. L,n
1. AnomalyID$_1$ or M21AnomalyID$_1$
:
n.
7. L,2
1. “ALIGN-TABLE-DEF”
2. AlignTableName
8. L,2
1. “ALIGNNAME”
2. L,n
1. AlignName$_1$
:
n.
9. L,2
1. “ELEMENTLIST”**
2. L,n
1. ElementID$_1$
:
n.
10. L,2</td><td>$m \geq 2$</td><td>List of m data items</td></tr>
</table>

CPName	*Parameter Value*		
	Description	*Range*	*Format*
	1. CPNAME* 2. CEPVAL* m. L,2 1. CPNAME* 2. CEPVAL* **NOTES:** “PPBUILDID” and “PPNAME” are required. “SLOTLIST”, “SUBSTRATELIST”, “AREALIST”, and “ALIGNLIST” are optional. * Supplier shall define as many of these CPNAME, CEPVAL pairs as are supported by the equipment. ** “ELEMENTLIST” is required when using the SEMI M21 coordinate system in the definition of an AlignName or AreaName.		

NOTE 1: ❖ Required ISEM parameters: “CARRIERID”, “LOCATIONID”, “PROCESS-BLD-GROUP”

13.2.3 *Remote Commands vs. Processing States* — The following table indicates states where the remote commands are allowed. This is indicated with a “X” mark.

Table 10 Remote Commands vs. Processing States

	COMMAND											
PROCESSING STATE	*STOP*	*START*	*RESUME*	*PP-UPDATE*	*PP-SELECT*	*PAUSE*	*NEXT-MATERIAL*	*MAP-CARRIER*	*PP-ASSIGN*	*CLEANUP*	*ABORT*	*PP-UNASSIGN*
IDLE					X			X	X			X
ABORTED									X	X		X
PROCESSING ACTIVE												
STOPPING									X		X	
ABORTING												
PAUSE												
ALARM PAUSED	X								X		X	
PROCESS PAUSE												
PAUSING	X								X		X	
PAUSED	X		X	X					X		X	
CHECKING	X								X		X	
PROCESS												
SETTING UP	X					X		X	X		X	X
READY	X	X				X			X		X	X
EXECUTING												
LOAD	X					X			X		X	
UNLOAD	X					X			X		X	
WORKING												
INSPECT	X					X	X		X		X	
ALIGN	X					X	X		X		X	
REVIEW	X					X	X		X		X	

14 Scenarios

14.1 *Run Level Reporting Scenario* — This scenario only has expected events (i.e., no alarms or errors).

COMMENT	HOST	EQUIPMENT	COMMENT
			The equipment is in the IDLE processing state and in The ONLINE REMOTE control state. The host has defined, linked, and enabled RUN level report for CEIDs 2, 3, and 5.
Host sends a PP-SELECT command specifying a ”CARRIERBLD”	S2,F49-->		
		<--S2,F50	Command Acknowledge
			The equipment transitions from IDLE to SETTING UP, and material arrives at input port.
		<--S6,F11	SETTING UP -> READY (CEID 3)
Event Report Acknowledge	S6,F12-->		
START	S2,F41-->		
		<--S2,F42	Host Command Acknowledge
			READY -> LOAD. [WHILE] Note End of Run LOAD -> WORKING WORKING -> UNLOAD UNLOAD -> LOAD [END WHILE]
		<--S6,F11	LOAD -> STOPPING (CEID 5)
Event Report Acknowledge	S6,F12-->		
		<--S6,F11	Run Processed Data Valid event.
Event Report Acknowledge	S6,F12-->		
			The equipment transitions from STOPPING to IDLE.

14.2 *PP-UPDATE Remote Command Scenario* — Host issues the PP-UPDATE remote command.

COMMENT	HOST	EQUIPMENT	COMMENT
START	S2,F41-->		
		<--S2,F42	Positive Acknowledge.
		<--S6,F11	READY -> LOAD
Positive Acknowledge.	S6,F12-->		
			[WHILE] Not End of Run 1) LOAD -> WORKING 2) WORKING -> UNLOAD 3) UNLOAD -> LOAD [END WHILE]
Sometime during the [WHILE]:			
PAUSE	S2,F41-->		
		<--S2,F42	Positive Acknowledge.
		<--S6,F11	Transition to PAUSING
Positive Acknowledge.	S6,F12-->		
		<--S6,F11	PAUSING -> PAUSED
Positive Acknowledge.	S6,F12-->		
PP-UPDATE	S2,F49-->		
		<--S2,F50	Positive Acknowledge.
		<--S6,F11	PAUSED -> CHECKING
Positive Acknowledge.	S6,F12-->		
		<--S6,F11	CEID is posted. [IF] the updates are valid: Return to the previous process state through history. [ELSE] Return to the PAUSED state. The Process program remains unchanged. [ENDIF]
Positive Acknowledge.	S6,F12-->		

14.3 *PP-SELECT Remote Command Scenario*

COMMENT	HOST	EQUIPMENT	COMMENT
			The equipment is in the IDLE processing state and in the ONLINE REMOTE control state.
Host sends a TABLE-DEF to the equipment.(See Section 11.)	S13,F13-->		
		<--S13,F14	Table Data Acknowledge
Host sends more tables if needed.	S13,F13-->		
		<--S13,F14	Table Data Acknowledge
Host prepares the remote command to initiate an inspection run, including the “TABLE-AREA-DEF”, ”TABLE-ALIGN-DEF”, and ”ELEMENTLIST”, if required.			
Host sends a PP-SELECT command specifying a ”CARRIERBLD”.	S2,F49-->		
		<--S2,F50	Command Acknowledge
			The equipment transitions from IDLE to SETTING UP to READY.
START	S2,F41-->		
		<--S2,F42	Host Command Acknowledge
			READY -> LOAD. [WHILE] Not End of Run LOAD -> WORKING WORKING -> UNLOAD UNLOAD -> LOAD [END WHILE]
			LOAD -> STOPPING
		<--S13,F13	Equipment sends anomaly table with additional data, including the table name using Table Data Send command.
Table Data Acknowledge	S13,F14-->		
			The equipment transitions from STOPPING to IDLE.

14.4 *Event Report and ISEM Table Transfer Command Scenario*

COMMENT	HOST	EQUIPMENT	COMMENT
			The equipment is in the IDLE processing state and in the ONLINE REMOTE control state.
Host defines report with AnomalyTableName and AnomalyTableType	S2,F33-->		
		<--S2,F34	Define Report Acknowledge
Link report to RunDataComplete event	S2,F35-->		
		<--S2,F36	Link Event Report Acknowledge
Enable event	S2,F37-->		
		<--S2,F38	Enable Event Acknowledge
Host sends an ISEM table to the equipment.	S13,F13-->		
		<--S13,F14	Table Data Acknowledge
Host sends more tables if needed.	S13,F13-->		
		<--S13,F14	Table Data Acknowledge
Host sends a PP-SELECT command specifying a "CARRIERBLD"	S2,F49-->		
		<--S2,F50	Remote Command Acknowledge The equipment transitions from IDLE -> SETTING UP -> READY.
START	S2,F41-->		
		<--S2,F42	Host Command Acknowledge READY -> LOAD. [WHILE] Not End of Run LOAD -> WORKING WORKING -> UNLOAD UNLOAD -> LOAD [END WHILE] LOAD -> STOPPING STOPPING -> IDLE
		<--S6,F11	RunDataComplete event with AnomalyTableName and AnomalyTableType.
Event Report Acknowledge	S6,F12-->		
Host sends Table DataRequest	S13,F15-->		
		<--S13,F16	Equipment sends requested table TBLACK = 0 if no errors.

15 GEM Capabilities

The purpose of this section is to specify any SEMI E30 additional capabilities that are required to be supported by this class of equipment.

15.1 *Requirements*

15.1.1 This standard requires that the SEMI E30 fundamental requirements and additional capabilities have been implemented on the ISEM equipment with the exception of limits monitoring and trace reporting. If these capabilities are implemented, they shall be implemented as required by the SEMI E30 document. The following SEMI E30 additional capabilities required by ISEM are:

- Dynamic Event Report Configuration
- Variable Data Collection
- Status Data Collection
- Alarm Management
- Remote Control
- Equipment Constants
- Process Program Management
- Spooling
- Trace Data Collection (optional)
- Control (Host-Initiated)

RELATED INFORMATION 1

NOTE: This related information is not an official part of SEMI E30.1 and was approved for publication by full letter ballot procedures on September 3, 1999.

R1-1 Defect Classification Code Management

The purpose of this section is to provide a method and specific formats to define, identify, and communicate coordinate systems and site locations on substrates for alignment sites, anomaly locations, and other sites used by the ISEM equipment.

R1-1.1 *Classification Codes and Defect Classification* — One function of review equipment is to view previously identified anomalies and to associate a defect classification code with each anomaly. A classification code is an identifier for a classification description.

Typically, the review equipment has a set of defect classification codes and their descriptions available to the operator. Then, for each anomaly, the operator selects a particular code to be associated with that anomaly. This action is defect classification.

The set of valid classification codes and their descriptions may change from one run to another. For example, the same main process program could be used with different substrate levels, and each level may use a different set of classification codes. The purpose of this section is to provide the requirements so that a user can both define several sets of classification codes and their descriptions and can also manage these sets on ISEM equipment.

R1-1.2 *Requirements*

- Each set of classification codes and their descriptions shall have an identifier, known as a classification code set ID.
- Review equipment shall provide a means for the user (the host or the operator) to define a classification code set, consisting of (a) the classification code set ID and (b) the list of classification codes and their descriptions.
- Equipment shall provide a means to manage the various classification code sets.
- A main process program shall include a process program variable that specifies the particular classification code set ID to be used.
- Equipment vendor shall provide documentation to the user regarding how to define and manage classification code sets.

Comment: In one implementation, the equipment considers a classification code set to be a sub-process program or an ISEM table. This would allow the user to identify a classification code set by name (using a PPID) or a table name and thereby managing this sub-process program with the SEMI E30 Process Program Capability or with SEMI E58 ARAMS tables.

R1-2 Reporting Coordinates and Coordinate Systems

The purpose of this section is to provide a method and specific formats to define, identify, and communicate coordinate systems and site locations on substrates for alignment sites, anomaly locations, and other sites used by ISEM equipment.

The ISEM-required formats are intended to minimize the number and type of site location format transformations needing to be supported by both equipment suppliers and users.

All ISEM-required site location formats involve the use of an ISEM-defined right-handed Cartesian coordinate system, established on substrates in an ISEM-defined manner. The scope of the detailed methods in this section are specific to unpatterned and patterned wafers in this release, but the section is intended to be general enough in methodology so that it can be extended to other substrate types in future revisions of ISEM, if required.

The purpose of inspection and review equipment is to locate, evaluate, classify, and report anomalies on substrates. ISEM equipment may deal with either unpatterned or patterned substrates or both. In most cases, the anomaly location is part of the information reported and/or used by ISEM equipment. An anomaly location is reported at a particular site with x,y coordinates in a particular coordinate system. Site coordinates are also used by ISEM equipment for the alignment sites for defining a coordinate system on a substrate. A standard method is needed to define a coordinate system and to report site coordinates for both alignment sites, anomaly locations, and any other reference sites needed by ISEM equipment. A standard method is essential in order to transfer the anomaly site information from one equipment to another.

R1-2.1 *Site Location Accuracy* — Each equipment has an accuracy with which it can define or locate a site as being within a certain area. This area associated with a site is determined by the equipment accuracy, based on the accuracy of its motion and imaging systems to locate a site, as well as on the accuracy with which it can define the coordinate system on the substrate.

When equipment shall locate a particular site on a substrate based on the expected or design-based location, then the location of a site or feature on an actual substrate is further affected by the accuracy of the equipment which placed the pattern on the substrate.

R1-2.2 *Expected or Designed Locations vs. Actual Locations* — The placement of patterns, sites, and coordinate systems is designed to be at certain mathematically described locations relative to one another and to an ideal substrate. These are the expected or designed locations. When a pattern is written by equipment onto a specific substrate, the actual placement of the pattern, the pattern-elements, and their features may differ from the expected locations, due to variations in equipment performance and variations in substrate shape and dimensions.

R1-2.3 *Substrate Coordinate Systems (Unpatterned)* — A substrate coordinate system is a coordinate system which has both origin and axes defined by the shape and dimensions of the substrate and which does not depend on whether there is a pattern on the substrate or whether it is unpatterned. This coordinate system is used to locate or define sites relative to the substrate.

R1-2.4 *Substrate Pattern Coordinate System* — A substrate pattern coordinate system is a coordinate system which has its origin and axes defined by the pattern as a whole on the substrate. This coordinate system is used to locate or to define sites relative to the pattern on the substrate. The expected or designed location of the pattern on the substrate can be defined in terms of the placement of the origin and axes of the substrate pattern coordinate system relative to those of the substrate coordinate system. The actual location of a pattern on a substrate may differ from the expected location. The actual location is determined by locating two or more alignment sites on the patterned substrate. The alignment sites are specific points of certain features in the pattern. The coordinates of the alignment sites are given in the substrate pattern coordinate system. In many cases, equipment does not align to the specific pattern elements but instead uses the defined locations of the pattern elements within the substrate pattern coordinate system.

R1-2.5 *Pattern Element Coordinate System* — A pattern-element coordinate system is a coordinate system which has its origin and axes defined by the pattern of one specific rectangular element in a pattern (a defined arrangement) of equal-sized rectangular elements. This coordinate system is used to locate or to define sites relative to that specific pattern-element. The expected or designed location of the pattern-element within a pattern can be defined in terms of the placement of the origin and axes of the pattern-element coordinate system relative to those of the pattern coordinate system. The actual location of a pattern-element within a pattern on a substrate may differ from the expected location. The actual location is determined by locating two or more alignment sites within the pattern-element. The coordinates of the alignment sites are given in the pattern-element coordinate system.

R1-2.6 *Parallel Coordinate Systems* — A second coordinate system is considered to be parallel to a first coordinate system if the origin of the second can be defined as a translation from the origin of the first and if the axes of the second are parallel and in the same direction as those of the first.

R1-2.7 *Requirements* — The following is a list of requirements for ISEM equipment regarding coordinate systems and reporting site locations:

- ISEM equipment shall document whether it deals with coordinate systems based on (a) a substrate, (b) a substrate pattern, or (c) a pattern-element or whether it deals with several of these coordinate systems.
- ISEM equipment shall establish a substrate coordinate system using a standard, documented method. This coordinate system is not based on any pattern on the substrate. This coordinate system shall be a right-hand Cartesian coordinate system and shall be identified by a name.

  NOTE: For wafers, this method is defined in SEMI M20 (Specification for Establishing a Wafer Coordinate System), and the coordinate system is named "M20."
- For equipment dealing with substrate pattern coordinates, the substrate pattern coordinate system shall be established in a standard, documented method relative to the substrate coordinate system (the "unpatterned" coordinate system). This substrate pattern coordinate system shall be a right-hand Cartesian coordinate system and shall be designed to be parallel to the substrate coordinate system. The substrate pattern coordinate system shall be identified by a name. The location of its origin and axes relative to the substrate coordinate system shall be communicated in terms of the substrate coordinate system.

  NOTE: For wafers, this method is the one described below, and the substrate pattern coordinate system is named "M20P", and its origin and axis relative to the SEMI M20 coordinate system are given in terms of "M20" coordinates and are communicated using XlateData.
- For equipment dealing with pattern-element coordinates, the pattern-element coordinate system shall be established in a standard, documented

method relative either to the substrate pattern coordinate system or to another pattern-element coordinate system. The pattern-element coordinate system shall be a right-hand Cartesian coordinate system which is designed to be parallel to the substrate pattern coordinate system. The pattern-element coordinate system shall be identified by a name. The location of its origin and axis relative to the substrate pattern coordinate system shall be communicated in terms of the substrate pattern coordinate system.

NOTE: For wafers, this method is based on SEMI M21, and the coordinate system is named "M21" and its origin and axis relative to the "M20P" coordinate system are given in terms of the M20P coordinates.

- ISEM requires that equipment have the capability to use site location information that is based on the user's product designs, which the user shall provide in the appropriate ISEM-required format.
- ISEM-compliant equipment shall have the capability to define, locate, and report site information using only the ISEM-defined right-handed Cartesian coordinate system formats. This requirement does not preclude equipment from having additional capability for defining or reporting site location information using other formats.
- Coordinate system name and placement relative to the "higher" coordinate system shall be defined and communicated using the following ISEM data items, in terms of either expected or actual placement: CoordSys, XlateData, and their included data items.
- Alignment site information shall be defined and communicated using the following ISEM items: the variable item AlignList, the "ALIGNLIST", the Process program class of "TABLE-ALIGN-DEF", and their included information.
- Areas to be inspected shall be reported using the specific coordinate system defined by the user. The following ISEM items are used to define and communicate area locations: the variable item "AREALIST", the "AREALIST", and the Process program class of "TABLE-AREA-DEF", and their included information.
- The displacement of an actual coordinate system relative to its expected location shall be communicated using the ISEM data item: XlateData and its included data items.
- The displacement of an actual site location relative to its expected site location shall be communicated using the ISEM data item: Offset and its included data items.
- The equipment vendor shall document the requirements for the ISEM data items used in alignment of a coordinate system.
- The equipment vendor shall provide and document a means for the user to define and communicate a pattern map using SEMI M21 data. A pattern map defines the layout of equal-sized rectangular pattern-elements which make up a pattern. Each pattern-element shall have a name, using the SEMI M21 naming convention.

  NOTE: For patterned wafers, the naming method shall be that described in SEMI M21, and the pattern-element information shall be communicated using the ISEM data item of SEMI M21Data.

- For ISEM compliance, inspection equipment shall report various anomaly data; AnomalyID, coordinates, and attributes. Review equipment shall receive this data for anomalies and be able to locate them and perhaps modify the coordinates. Anomaly coordinates shall be reported using ISEM table named "TABLE-ANOMALY-DEF" and its included data.

R1-3 Coordinate System for a Substrate

R1-3.1 *SEMI M20 Coordinate System* — The SEMI M20 standard (Specification for Establishing a Wafer Coordinate System) describes how to map a right-handed Cartesian coordinate system to a substrate so that its origin is at the center of the substrate, and its negative *y*-axis bisects the substrate's primary fiducial. This coordinate system is defined by ISEM to be the "M20" coordinate system. The only information required by equipment in order to establish an "M20" coordinate system is the substrate size and the type of fiducial, which are communicated using the ISEM data items named **SubstrateSize** and **Fiducial**. Another ISEM data item named **Orientation** identifies how the substrate is loaded on the equipment. Note that the SEMI M20 standard requires that the "M20" coordinate system is fixed on the substrate and is not affected by how the substrate is loaded on equipment. Also, as stated in the SEMI M20 standard, an orientation of "0" degrees designates a substrate loaded on equipment, with the primary fiducial towards the operator or "down."

R1-3.2 *M20P Coordinate System* — ISEM defines the M20P coordinate system to be one which is aligned to the pattern on the substrate. The M20P coordinate system is useful because in many cases, it is more significant to the user to know the location of an anomaly relative to the pattern on the substrate rather than relative to the substrate shape and dimensions. ISEM also defines the M20P coordinate system to be one which is designed to be "parallel" to the SEMI M20 coordinate system. In practice, because of experimental errors, both the origins and the axes may differ slightly from their intended values of a simple translation and no rotation. Equipment should be designed to be able to locate the alignment sites, given the various possible experimental errors.

R1-3.3 *Establishing an M20P Coordinate System* — A minimum of two alignment sites is necessary to establish an M20P coordinate system on a substrate. Additional sites are often used to determine a scaling ratio of the dimensions of the actual coordinate system relative to the dimensions of the expected coordinate system and are reported using the ISEM data item of **ScaleFactor**.

XlateData is used to report actual coordinate system location. Most equipment cannot distinguish whether patterned substrate site location errors are due to the substrate, the layout on the substrate, or the equipment's ability to locate the sites. However, information that is available through the use of patterned-substrate alignment sites can provide a means for identifying potential equipment problems. For instance, assume that the only pattern-layout location error on a substrate is that due to the establishment of the location of the substrate center and fiducial. For many users and equipment systems, this is a good assumption. If this is the case, then the ISEM data item named **XlateData** can be used to track this error. Although the error may result from multiple sources, being able to track it on various equipment will enable users to apply statistical process control techniques to identify the specific sources.

Offset sites may be found by equipment at actual locations which deviate from their expected locations through either pattern layout errors or equipment "stage" or imaging errors. Again, in a controlled manufacturing process, these combined errors should be normally distributed, and non-normal deviations may indicate possible equipment problems. The actual position of a site relative to its expected position shall be reported through the use of the ISEM data item named **Offset**.

R1-4 Layout of Rectangular Pattern Elements on a Substrate Using SEMI M20 Coordinate System

Equipment shall be capable of routine, automated operation without needing substrate layout information (e.g., field or die maps). However, having the capability to provide substrate layout information to equipment from the host can be desirable. ISEM defines a means to do this in this section for substrates, based on SEMI M21 (Specification for Assigning Addresses to Rectangular Elements in a Cartesian Array.) The SEMI M21 standard is limited to defining how to assign "addresses" to elements and how to find the "array center" element. It does not specify how the rectangular pattern-elements are located on the substrate. In this section, ISEM defines how these pattern-elements are located on a substrate, using the data item named M21Data, and how to establish within-element coordinate systems. Any additional layout information, such as within-element structure details or element attribute information, is beyond the scope of ISEM.

R1-4.1 *ISEM "M21" Layouts*

- An "M21" layout consists of an array of equal-sized rectangular pattern-elements with no space between the pattern elements.
- ISEM defines the "M21" layout on a substrate to include all pattern-elements which are either wholly or partially within the circumference of the substrate.
- The ISEM approach is to define the pattern map by specifying the M20P coordinate for the lower left corner of the minimum number of pattern-elements needed to define the layout, along with the pattern-element addresses (names). For a non-tiled layout, the location and name of a single pattern-element is sufficient to establish the "M21" layout. For tiled layouts, the location and name of one pattern-element in each row or column are required. Note that the location of the lower left corner of some pattern-elements may be outside the circumference of the substrate.
- The "M21" pattern-element coordinate system shall have its *x* and *y* axes parallel to the respective M20P coordinate system axes and shall have their origins at the lower left corner of each element. The pattern-element coordinate system shall have a name and a specific pattern-element address identifier per SEMI M21.
- Layout definition is supported only for host-to-equipment communications. The user is responsible for ensuring that the pattern-element addresses provided to the equipment agree with the SEMI M21 specification. The equipment need not check this, other than to ensure that there are not conflicts within the provided layout, and shall report results with pattern-element addresses as provided by the user.

- "M21" layouts are established within the M20P coordinate system and need not require any additional alignment site data than is needed to establish the M20P coordinate system. However, as with M20P, additional alignment may be necessary because of errors in either the pattern layout or the equipment's ability to locate features. Offset shall be used to report the location corrections that result from any within-element alignments.

R1-5 How an M20P Coordinate System Is Established on a Substrate

The following example is fairly basic. For this example, the M20P coordinate system has a zero translation from the SEMI M20 coordinate system. Also, the equipment documentation states that 4 alignment sites are required. The equipment does M20P alignment on two alignment sites and does a low resolution and then a high resolution alignment at each site. Note that the specific alignment point is different at the two resolutions, so the coordinates are slightly different. The alignment sites are defined to the equipment via the process program class named "TABLE-ALIGN-DEF", as detailed below. The order of the sites in "TABLE-ALIGN-DEF" is not important. The sites are then selected via the CPNAME named "ALIGNLIST", which is included in the PP-SELECT command. The order of the sites listed in "ALIGNLIST" is important and is as-specified in the equipment's documentation. The first item is the alignment site for the first low resolution site, the second item is for the first high resolution site, the third item is the second low resolution site, and the fourth is the second high resolution site.

"TABLE-ALIGN-DEF"

AlignName	*Coordx*	*Coordy*	*Coordsys*	*Attribute (1)*
Coarse1	-60000	-200	"M20P"	
Fine1	-60020	-205	"M20P"	
Coarse2	+60000	+200	"M20P"	
Fine2	+59980	+195	"M20P"	

"ALIGNNAME"

L,4

1. <Coarse1>
2. <Fine1>
3. <Coarse2>
4. <Fine2>

Using this information, the equipment will go to the nominal "M20" location for Coarse1, then "find" where it actually is. The offset between the nominal "M20" location and the actual "M20" location is then used to "find" Fine1. The actual M20 location of Fine1 is saved. The process is then repeated for Coarse2 and Fine2. The equipment can now determine the "M20" to M20P offset from the nominal and actual coordinates. First, a summary of the data:

xN1 = -60020 yN1 = -205 Nominal *x* and *y* data for the first fine site

xA1 = -59800 yA1 = -150 Actual *x* and y data for the first fine site

xN2 = +59980 yN2 = +195 Nominal *x* and *y* data for the second fine site

xA2 = +60060 yA2 = +175 Actual *x* and *y* data for the second fine site

The equipment first calculates Theta, using, for example, the formula:

$$\Theta = tan^{-1} \; \frac{MA - MN}{1 + MA \;\; MN}$$

where *MA* and *MN* are, respectively, the slopes of the lines connecting the two actual fine sites and the line connecting the two nominal sites, in "M20" coordinates, calculated as follows:

$$MA = \left[\frac{yA_2 - yA_1}{xA_2 - xA_1} \right] \qquad MN = \left[\frac{yN_2 - yN_1}{xN_2 - xN_1} \right]$$

The equipment then calculates *DeltaX* and *DeltaY*, using, for example, the formulas:

$$DeltaX = \left[\frac{C\sin(\Theta) + D\cos(\Theta)}{(\sin(\Theta))^2 + (\cos(\Theta))^2} \right]$$

$$DeltaY = \left[\frac{C\sin(\Theta) - D\cos(\Theta)}{(\sin(\Theta))^2 + (\cos(\Theta))^2} \right]$$

where *C* and *D*, the adjusted site 1 coordinates in a rotation-adjusted coordinate system, are calculated, for example, using the formulas:

$$C = yA1 - ((xN1 \;\; \sin\Theta) + ((yN1 \;\; \cos\Theta)$$

$$D = xA1 - ((xN1 \;\; \cos\Theta) - ((yN1 \;\; \sin\Theta)$$

The equipment can also calculate a ScaleFactor term to indicate the relative ratio between the length of the vector connecting the nominal alignment sites and the length of the vector connecting the actual alignment sites. This can be used, for example, to judge whether there is a problem with the alignment process, since the difference between these two vectors should be small.

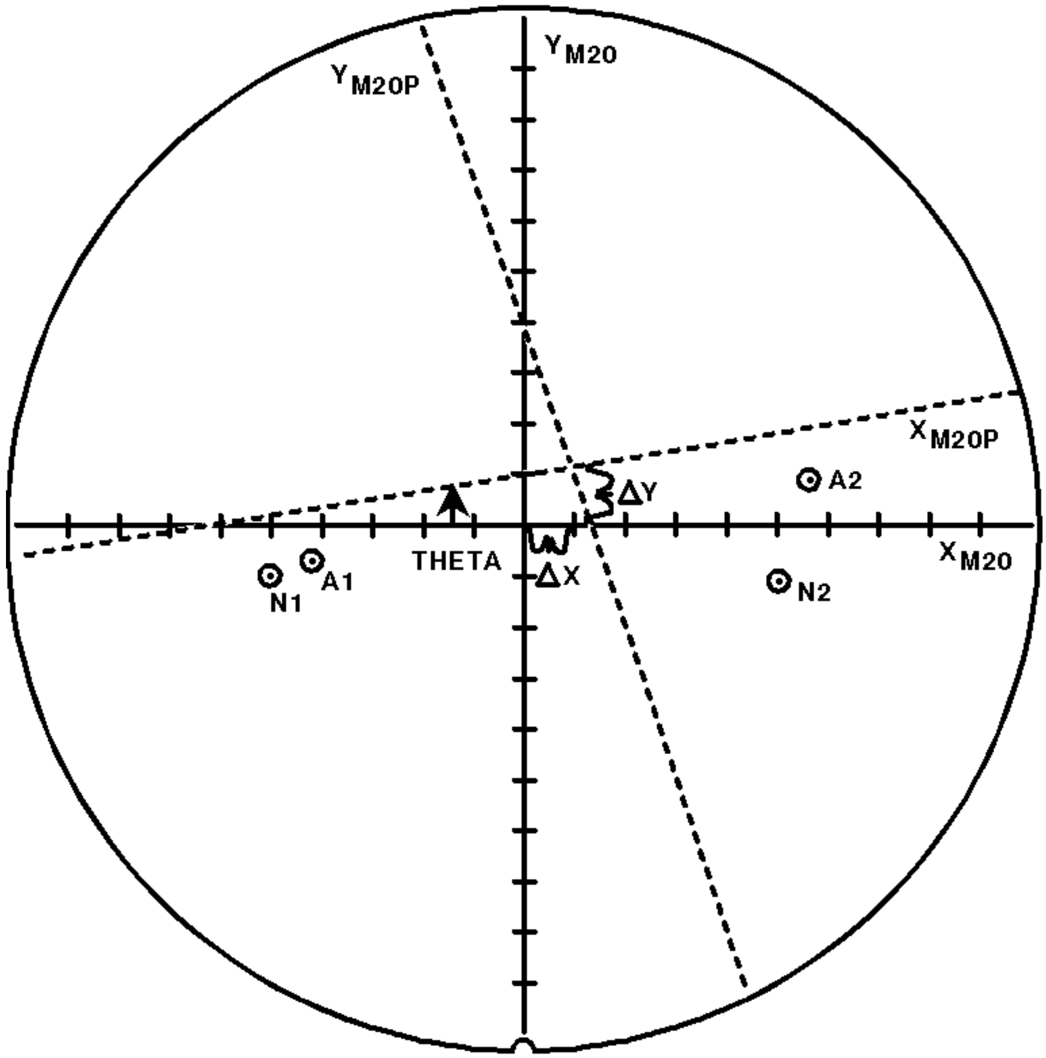
$$ScaleFactor = \frac{VA}{VN}$$

where *VA* and *VN* are the length of the vectors connecting the actual and nominal alignment sites, calculated using the formulas:

$$VN = \sqrt{(yN_2 - yN_1)^2 + (xN_2 - xN_1)^2}$$

$$VA = \sqrt{(yA_2 - yA_1)^2 + (xA_2 - xA_1)^2}$$

M20/M20P COORDINATE SYSTEMS EXAMPLE (EXAGGERATED)



N1 = -500000, - 100000
N2 = 500000, -100000
A1 = -420000, -70000
A2 = 560000, 85000

THETA = 8.988°
ΔX = 70312
ΔY = 96472

Figure 7
Review Data Management

RELATED INFORMATION 2
APPLICATION NOTES

NOTE: The material contained in these Application Notes is not an official part of SEMI E30.1 and is not intended to modify or supersede the official standard. Rather, these notes describe possible methods for implementing certain ISEM requirements described by the standard and are included as reference material.

R2-1 Using ISEM Table Attributes to Specify Process Related Data Item Variable Values

R2-1.1 Section 11.1 (ISEM Table Data) allows the host to use ISEM Table attributes to specify product and process related information related to the table data. The ISEM Variable Item Dictionary (Table 4) includes seven data items intended to be used for this purpose. They are identified with the comment “This information may be added by the host in the ISEM Tables.” Identifying the value of variable data items in Table attributes is not covered in use of ISEM Tables. One method to accomplish this is to use attribute identifiers (ATTRID*n*) that are the same identifiers that are used for the equipment variable data items (Table R2-1). (This is very similar to the method specified in Section 12.5 to identify values to override the default values of variable process program parameters using the “PP-SELECT” remote commands).

Table R2-1 ISEM Variable Items and Their Equivalent ISEM Table Attribute Identifiers

Variable Item (Table 4)		*ISEM Table Attribute Identifier*	*Description*
Name	*Type*		
OperatorID	DV	OperatorID	Identification of the operator of the inspection/review equipment.
ProcessEquipmentID	DV	ProcessEquipmentID	Identification of the process equipment used with the current material immediately prior to the inspection/review.
ProcessEquipmentLocation	DV	ProcessEquipmentLocation	Location (code) of the process equipment used with the current material immediately prior to the inspection/review.
ProcessProgramID	DV	ProcessProgramID	Identification of the process program used with the process equipment used on the current material immediately prior to the inspection/review.
ProcessLevel	DV	ProcessLevel	Identification of the processing level of the current material.
ProductID	DV	ProductID	The product identification of the current material inspected/reviewed.
ProcessRunID	DV	ProcessRunID	Run identification for the process prior to current inspection/review.

NOTE 1: The variable item may be identified using any appropriate SECS II data item format.

R2-2 Example ISEM Table with Item Attributes That Specify ISEM Variable Values Using S13,F13 Table Data Send

R2-2.1 Typical values for Data Items are indicated.

L, 8

1. <DATAID>
2. <OBJSPEC=null>
3. <TBLTYP=DefectData>
4. <TBLID=null>
5. <TBLCMD=1>

6. L, 2
    1. L, 2
        1. <ATTRID= "ProcessProgramID">
        2. <ATTRDATA= "My Recipe">
    2. L, 2
        1. <ATTRID= "ProcessLevel">
        2. <ATTRDATA= "My Level">
7. L, c
    1. <COLHDR1= "Insp_Anomaly ID">
    2. <COLHDR2= "Insp_Table specifier">
    3. <COLHDR3= "Insp_Coordinate X"
    4. <COLHDR4= "Insp_Coordinate Y"
    : // etc…as defined by Table 11.3.5
    c. <COLHDRc>
8. L, r
    1. L, c
        1. <TBLELT11> // A[1..16]
        2. <TBLELT12> // I4
        3. <TBLELT13> // I4
        4. ETC…
        :
        c. <TBLELT1c> // A
    :
    r. L, c
        1. <TBLELTr1>
        :
        c. <TBLELTrc>

SEMI E30.2-0698
HANDLER EQUIPMENT SPECIFIC EQUIPMENT MODEL (HSEM)

CONTENTS

SEMI E30.2-0698
HANDLER EQUIPMENT SPECIFIC EQUIPMENT MODEL (HSEM)

1 Purpose

1.1 This document establishes a Specific Equipment Model (SEM) for Handling equipment (HSEM). The SEM consists of equipment characteristics and behaviors that are applicable to this class of equipment and are required to be implemented in addition to the SEMI E30 fundamental requirements and additional capabilities. The intent of this document is to facilitate the integration of Handling equipment into an automated (semiconductor) factory. This document accomplishes this by defining an operational model for Handling equipment as viewed by a factory automation controller. This definition provides a standard host interface and equipment operational behavior.

2 Scope

2.1 The scope of this document is limited to the definition of Handling equipment behavior as perceived by a SECS-II host that complies with the SEMI E30 model. It defines the view of the equipment through the SECS communications link. It does not define the internal operation of the equipment. It includes a specific processing state model as the basis for the behavior of all equipment of this class.

2.2 This document requires that the SEMI E30 fundamental requirements and applicable additional capabilities (see Section 13,Additional SEMI E30 Capabilities) have been implemented on the handling equipment. This document expands the SEMI E30 standard requirements and capabilities in the areas of the processing state model, collection event, alarm documentation, remote commands, variable item, and process program management.

3 Limitations

3.1 *Communications* — It is required that any HSEM-compliant equipment follow the Communications State Model in SEMI E30. In addition, HSEM-compliant equipment shall support the High Speed Messaging Service (HSMS) communication standard sending SEMI E5 messages over TCP/IP. The reason behind this requirement is the amount of data available for monitoring from this class of equipment. This specification deals only with the behavior of the handler in communicating with the host. It is recognized that the handler may also have a communications link with other process equipment and that the other equipment may also have a communications link with the host. This specification is intentionally non-specific on the communications link requirements between handler and other process equipment to allow the user the greatest amount of flexibility in specific factory configurations.

3.2 *Multi-Process-Site HSEM Implementations* — This SEM makes some demands and assumptions about the Handler with multiple process-sites in a configuration. These requirements are as follows:

Handling equipment in a multiple process-site configuration (i.e., lead conditioning site, electrical test-site) provides identification and status information (see Variable Item) at both the site level and the subsite level. An example could be a handler with both a lead conditioning site and an electrical test-site, with the electrical test-site containing multiple subsites (i.e., test heads).

4 Referenced Documents

4.1 *SEMI Standards*

SEMI E4 — SEMI Equipment Communications Standard 1 Message Transfer (SECS-I)

SEMI E5 — SEMI Equipment Communications Standard 2 Message Content (SECS-II)

SEMI E10 — Standard for Definition and Measurement of Equipment Reliability, Availability, and Maintainability (RAM)

SEMI E30 — Generic Model for Communications and Control of SEMI Equipment (GEM)

SEMI E37 — High-Speed SECS Message Services (HSMS) Generic Services

SEMI E37.1 — High-Speed SECS Message Services Single-Session Mode (HSMS-SS)

SEMI E58 — Automated Reliability, Availability, and Maintainability Standard (ARAMS): Concepts, Behavior, and Services

5 Terminology

5.1 *alignment location* — Location that individual packaged units are placed at the process-site (e.g., electrical test).

5.2 *chaining* — The process of execution over multiple lots or runs with the same Process Program and the same handler operating conditions.

5.3 *electrical test-site* — A process-site on the equipment which is coupled with electrical testing equipment for purposes of performing package electrical testing.

5.4 *execution area* — The area from which a current copy of the process program instructions is executed.

5.5 *handling equipment* — An equipment class generally consisting of integrated mechanisms and controls for the purpose of manipulating packaged devices, trays, and tubes during the manufacturing process.

5.6 *indexing* — The controlled stepped movement of material through the handler.

5.7 *kit* — Specific items of hardware and software as specified by the equipment manufacturer that adapt the equipment for a specific unit or unit package.

5.8 *leadconditioning site* — A process-site on the handler where some form of conditioning occurs on the package leadfingers (i.e., warming).

5.9 *leadfinger (or substrate connector lead)* — (1) In ceramic packages, an area of refractory metal that has been plated and is designated for the attachment to a process-site. (2) The area of the unit designated for attachment to a process-site.

5.10 *leadframe* — A sheet metal framework upon which a chip (sometimes chips) is attached, wire-bonded, and then either molded with plastic epoxy or with ceramic and/or metal.

5.11 *media* — A temporary material carrier used to hold and transport units/devices (tubes, trays, etc.).

5.12 *media map* — Formatted data used to map functionally good and bad units/devices to an X, Y, Z location in the media. Maps can be requested by the handler for use prior to processing and then updated after processing.

5.13 *off-line programming (OLP) utility* — Utility to create, edit, and format process programs on a computer, as opposed to creating process programs at the equipment.

5.14 *process-site* — A location on the equipment where work is performed on a packaged device (i.e, electrical test-site, lead conditioning site).

5.15 *process subsite* — An addressable portion of a process-site.

5.16 *reset* — The action of changing the value of a variable, such as wafer count (usually to zero).

5.16.1 *safe state* — A state in which the equipment presents no danger to the product or user. This implies that safety interlocks are in place such that the equipment can be serviced without harm to the operator and that the material being processed has been removed from the processing station into an accessible location.

5.17 *slot* — A position in a carrier where a leadframe, tray, tube, or other media element may reside.

5.18 *tray* — A flat rectangular form of media for holding singulated packaged units. Also referred to as waffle packs or matrix trays. A tray is generally molded plastic with a defined matrix of cells or pockets tailored for specific packages.

5.19 *tube* — A hollow form of media for holding packaged units. Also referred to as rails or sticks. A tube is generally composed of extruded polymer with internal section dimensions and features tailored for a specific package.

5.20 *unit* — The functional integrated circuit (or chip) that is being handed to a specific process-site.

6 Requirements

6.1 *State Models* — The purpose is to define the equipment-specific processing state model and other state models necessary to portray the expected operational states of the equipment to enable host tracking and control in place of a local operator.

6.2 *State Model Requirements*

6.2.1 The processing state models in this document are required for implementing an HSEM-compliant handler in addition to the required state models in SEMI E30. A state model consists of the following state model diagram, state definitions, and state transition tables.

6.2.2 A state model represents the host's view of the handler, not necessarily the actual handler's internal operations.

6.2.3 All HSEM state model transitions shall be mapped sequentially into the actual equipment events that satisfy the requirements of those transitions. In certain implementations, the handler may enter a state and have already satisfied all of the conditions required by the HSEM state model for transition to another state. The handler makes the required transition without any additional actions in this situation.

6.2.4 Some equipment may need to include additional states. However, any additional states must not change the HSEM-defined state transitions. All expected transitions between HSEM states must occur.

6.3 *HSEM State Model*

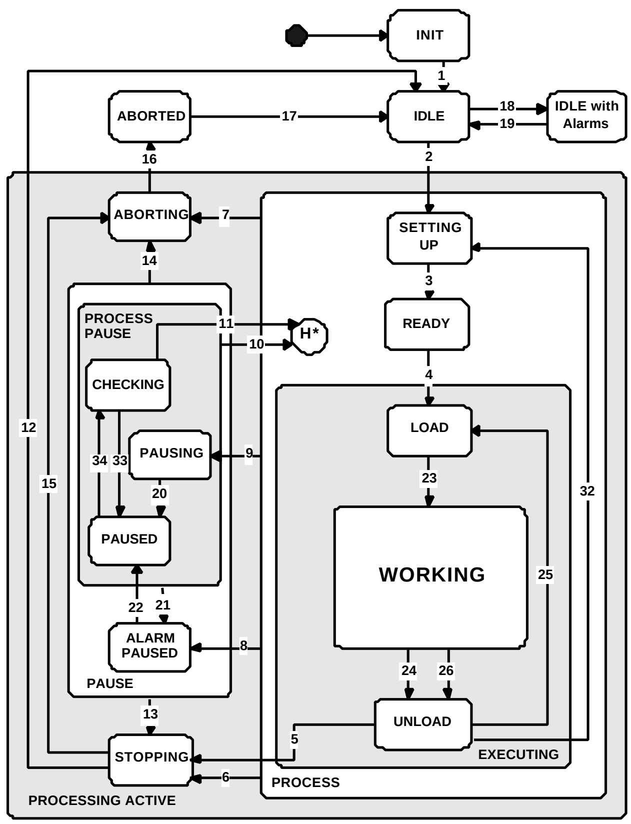


Figure 1
HSEM Processing State Model

6.4 *Description of Handler Processing States*

6.4.1 *ABORTED* — All activity is suspended as a result of an ABORT command. Any alarm and abort conditions must be cleared and verified by an operator before exit from this state.

6.4.2 *ABORTING (PROCESSING ACTIVE Sub-State)* — The handler has received an ABORT command. All activity is suspended. The handler is taking appropriate action to bring itself and material to a "safe" state where possible. Unit or Lot data may be invalid or not available.

6.4.3 *ALARM PAUSED (PAUSE Sub-State)* — An alarm has occurred in the PROCESS or PROCESS PAUSE states, and the handler is waiting for the alarm to be cleared.

6.4.4 *CHECKING (PROCESS PAUSED Sub-State)* — The handler verifies that updates made to the process program are valid (i.e., possible errors induced via an operator during the pause). This is a similar procedure to that which is done in SETTING UP before the handler is ready to transition to the READY state. At the completion of verification, an event is generated when the verification succeeds, and the operator or host must issue a RESUME command to the handler before it will resume processing from the point where it was paused.

6.4.5 *EXECUTING (PROCESS Sub-State)* — The handler is processing material automatically and can continue to do so without external intervention. This state may include interaction with the host or operator.

6.4.6 *IDLE* — Awaiting a command. IDLE is free of ALARMS and error conditions.

6.4.7 *IDLE with ALARMS* — An alarm has occurred in the IDLE state, and the handler is waiting for all alarms to be cleared.

6.4.8 *INIT* — Handler initialization is occurring.

6.4.9 *LOAD (EXECUTING Sub-State)* — This is the state the next unit or units are transferred from the input media to the process-site.

6.4.10 *PAUSE (PROCESSING ACTIVE Sub-State)* — The PROCESS state will be suspended at the completion of the current unit or next opportunity. Actions to put the handler in a safe state are performed. The handler is awaiting a command (RESUME, STOP, or ABORT) or for alarm(s) to be cleared.

6.4.11 *PAUSED (PROCESS PAUSE Sub-State)* — The PROCESS state has been suspended, and the handler is waiting for a command (RESUME, STOP, or ABORT). In this state, the operator may correct error conditions that do not affect the current Process Program selection. One of the possible corrective actions is for the operator to manually align the units being processed.

6.4.12 *PAUSING (PROCESS PAUSE Sub-State)* — The PROCESS state will be suspended at the completion of the current unit or next opportunity. The handler cannot transition to PAUSED state until the current unit is completed and the handler is in a "safe state".

6.4.13 *PROCESSING (WORKING Sub-State)* — Unit is moved to process-site (e.g., for electrical test, to insert into contactor).

6.4.14 *PROCESS COMPLETE (WORKING Sub-State)* — Unit process is complete. Unit is unloaded or returns to alignment for a step and repeat.

6.4.15 *PROCESS (PROCESS Sub-State)* — This state is the parent of those sub-states which refer to the preparation and execution of a process program.

6.4.16 *PROCESS PAUSE (PAUSE Sub-State)* — The handler is free of alarm conditions in the PAUSE state.

6.4.17 *READY (PROCESS Sub-State)* — The handler is ready to begin processing and is awaiting a START command from the operator or host.

6.4.18 *SETTING UP (PROCESS Sub-State)* — The handler is satisfying conditions so that processing can begin. This includes the receipt of any process programs, the material to be processed, and machine-specific calibration. While in this state, the handler can be single-stepped through each process in order for the operator to ensure that the handler is moving the unit correctly.

6.4.19 *STOPPING (PROCESSING ACTIVE Sub-State)* — The handler has completed a Process Program or has been instructed to stop processing and shall do so at the next opportunity. All necessary cleanup is completed within this state with regard to material, data, control system, etc. Data is preserved. Any error condition is cleared before exiting from this state.

6.4.20 *UNLOAD (EXECUTING Sub-State)* — The unit is being removed from the process-site, and the handler determines which transition to take.

6.4.21 *WORKING (EXECUTING Sub-State)* — The handler is processing a specific unit.

6.5 *HSEM Processing State Transitions Table*

Table 1. Processing State Transitions Table

#	*Current State*	*Trigger*	*New State*	*Actions*	*Comments*
1	INIT	All handler initialization is complete with no alarms or error conditions.	IDLE	None	None
2	IDLE	A Process Program is selected.	SETTING UP	Handler-dependent	Commit has been made to setup.
3	SETTING UP	All setup activity has completed, and the handler is ready to receive a START command.	READY	The handler is waiting for a START command.	The selected Process Program is available for execution, and material is present at the input port.
4	READY	The handler receives a START command.	LOAD	Transfers the next unit to the process-site.	LOAD is an EXECUTING sub-state.
5	UNLOAD	The material unloaded completes.	STOPPING	The handler completes the last unit.	None
6	PROCESS	The handler has received a STOP command.	STOPPING	The handler completes the current unit in the WORK-ING state and unloads it.	The handler begins its cleanup procedure.
7	PROCESS	The handler has received an ABORT command from operator, host, or self-generated.	ABORTING	The handler is put in a "safe" state.	Unit or lot data may be invalid or not available.
8	PROCESS	An alarm occurs.	ALARM PAUSED	PROCESS activity is suspended, and the handler is waiting for all alarms to be cleared.	ALARM PAUSED is a PAUSE sub-state.
9	PROCESS	The handler has received a PAUSE command.	PAUSING	The PROCESS state shall be suspended at the completion of the current unit. Any necessary actions to put the handler in a safe state are performed.	PAUSING is a PAUSE sub-state.
10	PROCESS PAUSE	The handler has received a RESUME command.	Previous PROCESS State	Proceeds from the point where processing was previously suspended.	None
11	CHECKING	Parameter checking completes successfully.	STATE based on conditional table.	Return to previous state.	None

Table 1. Processing State Transitions Table

12	STOPPING	The handler cleanup is complete, and the handler is free of alarms.	IDLE	None	None
13	PAUSE	The handler has received a STOP command.	STOPPING	The handler proceeds with cleanup.	Data is preserved and is valid.
14	PAUSE	The handler has received an ABORT command.	ABORTING	Any unsafe condition is resolved, if possible.	Data may be invalid or unavailable.
15	STOPPING	The handler has received an ABORT command.	ABORTING	Any unsafe condition is resolved, if possible.	Data may be invalid or unavailable.
16	ABORTING	Unsafe conditions have been resolved where possible.	ABORTED	The handler is waiting for alarm and ABORT conditions to be cleared.	The only state change allowed is to IDLE.
17	ABORTED	An operator has verified that all alarms and abort conditions have been cleared.	IDLE	None	None
18	IDLE	An alarm is set.	IDLE with ALARMS	The handler waits for all alarms to be cleared.	None
19	IDLE with ALARMS	All alarms have been cleared.	IDLE	None	The IDLE state is free of alarms.
20	PAUSING	The handler has completed Process-ing the Current unit in the WORKING state and achieved a safe condition.	PAUSED	The handler is waiting for a command (RESUME, STOP, or ABORT).	None
21	PROCESS PAUSED	An alarm is set by the handler.	ALARM PAUSED	The handler waits for all alarms to be cleared or for a STOP or ABORT command.	None
22	ALARM PAUSED	All alarms are cleared.	PAUSED	The handler is waiting for a command (RESUME, STOP, or ABORT).	None
23	LOAD	A unit(s) is loading to the process-sites.	WORKING	The unit(s) is being processed.	None
24	WORKING	The processing of the current unit(s) completes normally.	UNLOAD COMPLETED UNIT	This unit(s) is transferred from the process sites.	"Normal" completion of the unit(s).

Table 1. Processing State Transitions Table

25	UNLOAD COMPLETED	The material unloaded is complete, and material is available.	LOAD	Transfers the next unit(s) to the processing location.	None
26	WORKING	The processing of the current unit(s) completes abnormally.	UNLOAD COMPLETED	This unit(s) is transferred from the process-site.	"Abnormal" completion of the unit(s).
32	UNLOAD UNIT	The last unit *p* of a lot completes, and a new lot or last carrier is received.	SETTING UP	The handler waits for another SELECT command.	None
33	CHECKING	Error was detected in new parameter setting being validated in the CHECKING state.	PAUSED	None	None
34	PAUSED	A RESUME command was received.	CHECKING	Validation of the Process Program Parameters.	Host or operator is required to issue a RESUME command before processing shall continue.

6.6 *Process Model Conditions Table*

Table 2. Process Resume Conditions

Condition	*Next State*
Checking determines that process program conditions were changed.	SETTING UP
Previous State WORKING.	UNLOAD
Previous State READY.	READY
Previous State was SETTING UP.	SETTING UP

7 Collection Event List

7.1 *Requirements*

7.1.1 ALL SEMI E30-required Events are required by the HSEM. Since a Processing State Model is required by the HSEM, all state transitions are required Events.

7.1.2 All SEMI E30-required Events associated with the GEM Control, Communications, Alarm, and Spooling State Models are required.

7.1.3 This section of the HSEM lists only those collection events that are not associated with a change of state or those requiring specific data variables (DVVALs) or Reports defined in the HSEM.

7.2 *Collection Event Tables* — The first table contains required events and associated reports. The second table contains required events and associated data variables.

Table 3. Processing State Transitions Requiring Report Levels

Transition	*From State*	*To State*	*Required DVVALs or Report*
SETUP COMPLETE (3)	SETTING UP	READY	Setup Report

Table 4. Other Required Collection Events

Collection Event Name	*Required DVVALs*
LotComplete	See Lot Report.
DockStatusChange	DockingStatus
BinDataAvailable	See Bin Data Report.
LotStart	Lot/SubLot Start Report
CarrierEmpty	Media-ID, Lot-ID
CarrierFull	Media-ID, Lot-ID
SubLotComplete	SubLot-Report
ReaderFailed	Reader-Type, Barcode-Error-Type
UnitCntInterval	Unit-Count-Interval, Time-Stamp
MediaCntInterval	Media-Count-Interval
SkipCntInterval	Skip-Count-Interval
MediaChange	Media-ID, Product-ID, Media-Type
SubLotStart	Lot/SubLot Start Report

8 Variable Items

The purpose of this section is to define the list of variable items required by the HSEM. Values of these variables are available to the host through collection event reports and host status queries.

8.1 *Requirements*

8.1.1 All generic variable items defined in SEMI E30 are required by all HSEM equipment.

8.1.2 Variable items required by HSEM are categorized as follows:

- *Common Variables (CVs)* — Variables common to all testers.
- *Configuration-Specific Variables (CSVs)* — Variables associated with a specific configuration of the above equipment class.

8.1.3 Any supplier-defined variables shall be documented in the same format used by this document. The following minimum information is required:

<variable name> **Class:** <ECV, SV, or DVVAL> **Format:** <SML>

Description: <If class = DVVAL, description must contain statement of when data is valid.>

<If format = ASCII, then a length is required. It is assumed to be left-justified unless otherwise noted.>

8.2 *Data Types*

8.2.1 Equipment Constants (ECVs) can be changed by the host using S2,F15. The operator may be able to change some values, but the equipment does not change the values on its own. The value of an equipment constant may be queried by the host at any time, using the S2,F13/14 transaction. They reside in non-volatile memory of the equipment. Equipment constants remain in effect until they are overwritten, either by manual entry or by a NEW EQUIPMENT CONSTANT SEND.

8.2.2 Equipment constants have various uses in HSEM, including the following:

- Equipment offsets that match the performance of several pieces of equipment that would otherwise perform differently due to inherent manufacturing differences. Examples are home values and motion axis scaling factors.
- Setting the configuration of the equipment to allow for different material specifications, equipment options, material flows, frequency of automatic functions, etc. Examples are media and yield check frequency.
- Managing optional machine features. Examples are constants that indicate whether optional features, such as automated media stackers, are present and control the configuration and function of these optional subsystems when they are present.

8.2.3 Status Variables (SVs) are valid at all times. An SV may not be changed by the host but may be changed by the equipment or operator. The value of status variables may be queried by the host at anytime using the S1,F3/4 or S6,F19/20 transactions.

8.2.4 DVVALs are variables that are valid only upon the occurrence of specific collection events. An attempt to read a data variable at the wrong time shall not generate an error, but the data reported may not have relevant meaning.

8.2.5 *Data Item Requirements for Multi-Head, Multi-Site Equipment* — The identification for multi-head and multi-site data (data variable, status variables, events, etc.) is addressed in this specification through the use of status variables with list structures. In the table below, the subheading "Process-Site Group" contains variables which must be available for all process-sites on the handler equipment. When multiple process-sites exist, either a list structure or table structure may be used to show multiple occurrences of a specific variable.

8.3 *Variable Item Table*

Table 5. Variable Item Table

Variable Name	*Category*	*Description*	*Class*	*Format*	*Comments*
		Physical Handler Group			
EquipSerialID	CV	Identification of Equipment	SV	A[1..16]	Valid in all states.
HandlerComStatus	CSV	Status of comm link between handler/s (0 = Disabled, 1 = 1-way enabled, 2 = 2-way enabled, 3 = Not communicating)	SV	U4	Valid in all states.
KitID	CSV	ID of unique tooling unit	SV	A[1..24]	Valid in all states.
LightPoleStatus	CSV	Color/status (i.e., Red/flash)	SV	A[1..16]	Valid in all states.
LinkPortStatus	CSV	(3 = Input/Output linked, 2 = Input linked, 1 = Output linked, 0 = HANDLER not Linked)	SV	U4	Valid in all sub-states.
MediaID	CV	Media Serial Number	SV	A[1..24]	Valid in Executing state.
OperationType	CSV	Current Operation Mode (i.e., maintenance, production)	ECV	A[1..24]	Valid in all states.
OperatorID	CSV	Current Operator ID	ECV	A[1..24]	Valid in all states.
QueueStatus	CV	PPID Queued to be run	SV	U4	Valid in all states.
ReaderErrorType	CSV	Type of error detected by the material reader.	DVVAL	A[1..24]	Supplier-defined.
		Process-Site Components			
AlignmentCount	CSV	Number of units since last alignment (i.e., Homing/ Adjustment).	SV	U4	Valid in all states.
DockingStatus	CSV	Information on handler/ docking status (0 = Yes, 1 = No)	SV	U4	Valid in all states.
HardBinID	CV	Process-Site Bin Out Number	DVVAL	U4	Valid at BIN-DATA-AVAILABLE Event.
HardBinName	CV	Process-Site Bin Out Name	DVVAL	A[1..40]	Valid at BIN-DATA-AVAILABLE Event.
InsertionForce	CSV	Insertion-Force energy	DVVAL	F8	Valid in PROCESSING state.

Table 5. **Variable Item Table**

InsertionForceSetpoint	CSV	Insertion-Force set point (setpoint)	ECV	F8	Valid in all states.
LotID	CV	Lot Identification	SV	A[1..40]	Valid in all states.
LotProcessingTime	CV	The time since start of current Lot. HHMMSSCC	DVVAL	A[16]	Valid in Process-Sub-State.
MediaChangeTime	CV	Elapsed time to replace media and send ready.	SV	A[16]	Valid in all states.
MediaCount	CV	Number of media since last reset.	SV	U4	Valid in all states.
MediaCountInterval	CV	Event generated when number of media is completed.	SV	U4	Valid in all states.
PresentPositionActual	CSV	Present position actual.	ECV	U4	Valid in all states.
PresentPositionSetpoint	CSV	Present position set points (setpoint).	ECV	U4	Valid in all states.
ProcessSiteTemp	CSV	Process-site temperature (setpoint).	SV	F8	Celsius - Set point.
ProcessSiteID	CV	ID of process-site in configuration.	SV	U4	Valid in all states.
ProcessSiteStatus	CV	$Site_n$ Availability (1 = enabled, 0 = disabled)	DVVAL	F8	Valid in all states.
ProcessSubsiteID	CV	Subsite matrix location within process-site.	SV	U4	Valid in all states.
ProcessSubsiteStatus	CV	$Subsite_n$ Availability (1 = enabled, 0 = disabled)	DVVAL	F8	Valid in all states.
ProductID	CV	ID of product for which tester is currently configured.	DVVAL	A[1..40]	Valid in PROCESS states.
SkipCount	CSV	Number of units skipped since last reset (Skip + Process = Total)	SV	U4	Valid in all states.
SkipCountInterval	CSV	Event generated when number of units is skipped.	SV	U4	Valid in all states.
StartProcessPortID	CSV	Start process source (i.e., hand, keyboard, host)	SV	U4	Valid in all states.
SubLotID	CV	SubLot Identification	SV	A[1..40]	Valid in all states.

Table 5. Variable Item Table

UnitCount	CV	Number of units since last reset.	SV	U4	Valid in all states.
UnitCountInterval	CV	Event generated when number of units is completed.	SV	U4	Valid in all states.
UnitPosition	CSV	X, Y, Z media location of a unit.	SV	U4	Valid in all states.
UnitStatus	CSV	(1 = Processed, 0 = Skipped)	SV	U4	Valid in all states.

8.4 *HSEM Required Reports* — The reports below are required as "canned" or preconfigured reports by HSEM. HSEM does not require the equipment to guarantee the accuracy of data identified in these reports outside the PROCESSING ACTIVE state defined in the HSEM process state model.

8.4.1 *Setup Report* — Table 6 contains variables that are required to be available at the setup complete event.

Table 6. Required Variables at Setup Complete Event

Variable Name	*Notes*
KitID	Configuration Kit
MediaID	Serial # of Media
LotID	
PPExecName	(per SEMI E30)
EquipID	
ProductID	Current
InsertionForceSetPoint	

8.4.2 *Process Report* — Table 7 contains variables that must be available when the equipment is in the PROCESSING state.

Table 7. Required Variables for PROCESSING State

Variable Name	*Notes*
AlignmentCount	
ProcessSiteTemp	
LotProcessTime	
OperatorID	
OperationType	

8.4.3 *Lot/SubLot Report* — Table 8 below contains variables that must be available at the completion of a wafer.

Table 8. Lot/SubLotStartVariables

Variable Name	*Notes*
OperatorID	
LotID	
SubLotID	
PPExecName	(per SEMI E30)
ProductID	
SkipCount	

8.4.4 *SPC Report* — The table below contains variables that must be available and reported at the completion of a unit.

Table 9. Required Variables at Completion of Unit

Variable Name	*Notes*
UnitCount	
SkipCount	
SkipCountInt	
UnitCountInt	
MediaCount	
MediaCountInterval	
HardBinID	
MediaID	
OperatorID	

9 Process Program Management

9.1 *Process Program Requirements*

9.1.1 The HSEM requires that the SEMI E30 capability of process program management be fully supported for this class of equipment. The HSEM also requires that the process program have a structure that enables the user to build process programs with default conditions that can be overridden for a run. The concepts of process program structure and process program variable parameters are discussed in the following sections.The HSEM also requires the following:

- Minimum, maximum, and default parameter values must be defined for all process programs.
- Verification will occur when a process program is downloaded to the equipment; the program syntax must be verified by the equipment manufacturer.
- Parameter validation will occur when a process program is downloaded. Parameters must be type and range checked.

- Equipment should provide the functionality to manually or interactively modify the parameters set in the process program.
- An error message must be generated from the handler if the process program parameters are outside the range of the machine calibration.

9.2 *Process Program Structure*

9.2.1 A handler process program must contain the following information:

- Machine-specific configuration parameters
- Process-Site-specific information section
- Media-Type-specific information section
- Unit (Unit/Package) information section

9.2.2 When combined, this information constitutes a complete process program. It is emphasized that the HSEM does not enforce the exact format and data type of each section. However, it does provide direction as to what each section should consist of.

9.2.3 *Machine-Specific Configuration Parameters* — Each brand or type of handler may have one or more machine-specific configuration parameters. Examples of such parameters would be input configuration, number of process-sites, and output configuration. Even though they are supplier-specific, these parameters nevertheless play a vital role in the overall generation or creation of a process program. Since the machine-specific parameters can differ from one equipment manufacturer to another, the HSEM does not specify the exact number, data types, and format of these parameters. These details are left to the sole discretion of the equipment manufacturer.

9.2.4 *Process-Site-Specific Information* — This process-site-specific section contains information necessary for the configuration and execution of the various process-sites configured on the equipment. Each equipment may contain different process-site configurations. Since these configurations will differ, the HSEM does not specify the exact number, data types, and format of these parameters. HSEM does recommend a minimum list of data items for the common handler process-sites. These include:

9.2.5 *Thermal Conditioning Site Parameters*

- Temperature Set Point
- Upper-Temp Guardband Set Point
- Lower-Temp Guardband Set Point
- Soak Time
- Test-Site Temp

9.2.6 *Test-Site Parameters*

- Device Pick Up/Place
- Speed of Device Pick Up/Place
- Device Insertion/Retraction
- Speed of Device Insertion/Retraction
- Speed of Index Mechanism
- Insertion/Place Force/Stroke

9.2.7 *Lead Condition Site Parameters*

- Device Pick Up/Place
- Speed of Device Pick Up/Place
- Speed of Insertion
- Insertion Force/Stroke

9.2.8 *Sort Sites Parameters*

- Device Pick Up/Place
- Speed of Device Pick Up/Place
- Device Index
- Insertion Force/Stroke
- Sort Category Set
- *Sort Category* – Full/Empty/Partial
- *Sort Media* – In place or empty.

9.2.9 *Unit/Device-Specific Information* — The unit/device-specific section contains information necessary for the configuration and execution of the specific units to be handled by the equipment. HSEM requires a minimum list of data items be available to determine package dimensions, terminal dimensions, package height, and coplanarity.

9.2.10 *Media-Type-Specific Information* — The media-type-specific information section contains information necessary for the configuration and execution of the specific media type in use on the equipment. HSEM requires a minimum list of data items be available to determine row/column count, X/Y distance to a cell, device height in tray, media height, and X/Y pitch.

9.3 *Methods of Process Program Creation* — The method by which an equipment manufacturer creates a process program may be unique to that manufacturer. However, it is required that the customer at least be given both of the following options for the creation of a process program.

9.3.1 *Off-Line Development* — Using this method, the customer is given a set of software tools (process program compilers, decompilers, and debuggers) that enables him/her to generate or create a process program using the above mentioned information (flow, parameter, functional test, etc.). The newly generated process program then is downloaded onto a specific handler, verified, and is now ready to be selected and executed locally by the operator or remotely by the host computer. If this process is used, the supplied software tools should closely mimic or simulate a handler so that a user can create a complete process program. In many situations, minor adjustments may be needed to the process program on the equipment before it is completely ready for execution.

9.3.2 *On-Line Development* — The second option is to enable the user to download the above-mentioned information (tables or files) onto the equipment and create the actual process program on the equipment itself.

10 Remote Commands

The purpose of this section is to identify remote commands, command parameters, and valid commands versus states in the processing state models.

10.1 *Requirements*

- The equipment must support the GEM-required remote commands. (Some of the SEMI E30-required remote commands are restated here to define HSEM-specific requirements.)
- All the remote commands defined by HSEM are required.
- The alphanumeric strings defined by HSEM for remote commands (RMCD) and command parameters (CPNAME) are required.
- If additional remote commands are supported, then the "Remote Command Versus Valid States" matrix must be generated for these additional commands. Place an "X" in the table for each state in which a given command is valid.

10.2 *Remote Command Descriptions*

10.2.1 *ABORT-LOT* — This command terminates the current processing. ABORT makes no guarantee about completion of the current unit. Levels of ABORT may be specified (see Table 10, Remote Command Descriptions, for details).

10.2.2 *PAUSE* — This command transitions the handler to the PAUSING process state when the current unit/media completes processing.

10.2.3 *RESUME* — This command resumes processing from the point where the process was PAUSED. This command is only recognized if the handler is in the PAUSED state.

10.2.4 *PP-SELECT* — This command instructs the handler to copy the indicated Process Program from non-volatile storage to the handler's Process Program execution area. Process Program Variable Parameters can be specified in this command which modify the default values for these Variable Parameters in the Process Program.

10.2.5 *START* — This command is only available to the host or operator when a process program has been selected and the handler is in the READY processing state. The START command instructs the handler to initiate processing.

10.2.6 *STOP* — This command completes the current unit, stops in a safe condition, and returns to the IDLE processing state. STOP has the intent of bringing about a normal termination after completion of the current unit.

10.2.7 *LAST-CARRIER* — This command instructs the handler to treat the current carrier being processed on the handler as the last carrier of the lot. This forces the subsequent carrier to be considered the first carrier of the next lot.

10.2.8 *NEW-LOT* — This command instructs the handler to treat the next carrier to be processed as a new lot. A new LOT-ID and carrier-list must accompany the new lot command. This command forces subsequent carriers to be considered part of the lot, until all carriers in the carrier list have been processed or until a LAST-CARRIER command is received.

10.2.9 *RESET-TOOL-COUNTS* — This command will initialize equipment tool counts. The minimum set are those contained in the Variable Items section.

10.2.10 *PURGE* — Purge flush or clean the equipment of process material.

10.3 *Associated Remote Command Parameters*

Table 10. Remote Command Descriptions

Command	*Parameter*			
Name	*Name*	*Opt./Req.*	*Description*	*Format*
ABORT	"PROCESSSITEID"	OPT	ID of handler process-site.	U4
	"CLEANUP"	OPT	The handler finishes processing the current unit and removes all carriers that belong to the lot and enters the Aborting state.	A[7]
ABORT-CARRIER	"PROCESSSITEID"	OPT	ID of handler process-site.	U4
	"CLEANUP"	OPT	The handler will finish processing the current carrier.	A[7]
NEW-LOT	"PROCESSSITEID"	OPT	ID of handler process-site.	U4
	"LOTID"	REQ	ID of new LOT.	A[1..40]
PAUSE	"PROCESSSITEID"	OPT	ID of handler process-site.	U4
PP-SELECT	"PROCESSSITEID"	OPT	ID of handler process-site.	U4
	"LOTID"	OPT	Lot to be processed with this program.	A[1..40]
	"PPID"	REQ	The ID of the program to be used.	A[80]
	"MEDIALIST"	OPT	One or more media to be processed with this program.	ASCII List
RESET-TOOL-COUNTS	"PROCESSSITEID"	OPT	ID of handler process-site.	U4
	"SVIDLIST"	REQ	List of SVIDs to reset.	ASCII List
RESUME	"PROCESSSITEID"	OPT	ID of handler process-site.	U4
START	"PROCESSSITEID"	OPT	ID of handler process-site.	U4
STOP	"PROCESSSITEID"	OPT	ID of handler process-site.	U4
	"CLOSELOT"	OPT	Automatically close lot.	BOOL
PURGE			Purge handler of all material.	NO PARAMS

10.4 *Remote Commands and HSEM Process Model Mapping* — Table 11 illustrates the relationship between remote commands and states of the HSEM processing state model. An "X" indicates that a command is valid for use in this state. If a remote command is attempted during a non-valid state, the equipment would reject the remote command.

Table 11. Remote Commands vs. Process States

PROCESSING STATE \ **COMMAND**	STOP	START	RESUME	PP-SELECT	PAUSE	ABORT-MEDIA	ABORT
IDLE				X			
PROCESSING ACTIVE							
PROCESS							
SETTING UP					X		X
READY	X	X			X		X
EXECUTING							
LOAD	X					X	X
WORKING	X				X	X	X
UNLOAD	X					X	X
PROCESS PAUSE							
PAUSING							X
PAUSED	X		X			X	X
CHECKING				X			X
ALARM PAUSED			X				X
ABORTED						X	

Table 12. Remote Commands vs. Process States (cont.)

PROCESSING STATE \ **COMMAND**	PURGE	NEW-LOT	LAST-CARRIER	RESET-TOOL-COUNTS
IDLE		X		X
PROCESSING ACTIVE				
PROCESS				
SETTING UP			X	
READY	X	X	X	X
EXECUTING				
LOAD				
PROCESSING				
UNLOAD				
PROCESS PAUSE				
PAUSING				
PAUSED	X	X	X	X
CHECKING		X	X	
ALARM PAUSED	X			
ABORTED	X			

11 Scenarios

The purpose of this section is to document possible HSEM-specific operational scenarios.

11.1 *Normal Run Scenario*

This is an error-free run of a single lot with a single test-site. All optional SEMI E30 events are turned off by default.

COMMENT	HOST	EQUIPMENT	COMMENT
Host selects process program.	S2,F41-->		
		<--S2,F42	Equipment Ack
Setting Up			
Setup Complete			
		<--S6,F11	Event : PPLoadOk
Host Ack	S6,F12-->		
Process			
Process.SettingUp			
Process.Ready			
Host commands start-of-lot.	S2,F41-->		
		<--S2,F42	Equipment Acks Start.
			Handler cycles devices... ...until empty...
		<--S6,F11	Event : Empty
Host Acks Event	S6,F12-->		
Host commands end-of-lot.	S2,F41-->		
		<--S2,F42	Handler acks Rmt.Cmd.
Stopping			
		<--S6,F11	Event : Lot Completed
Host Acks Event	S6,F12-->		
Idle			

11.2 *Normal SPC Scenario* — This is a normal SPC run with all optional SEMI E30 events turned off by default.

COMMENT	**HOST**	**EQUIPMENT**	**COMMENT**
Host selects GEM Alarms to enable (list).	S5,F3-->		
		<--S5,F4	Alarms xyz ON
Host selects GEM Events to enable (list).	S2,F37-->		
		<--S2,F38	Events xyz ON
Host selects process program.	S2,F41-->		
		<--S2,F42	Equipment Ack
Setting Up			
Setup Complete			
		<--S6,F11	Event : PPLoadOk
Host Ack	S6,F12-->		
Process			
Process.SettingUp			
Process.Ready			
Host requests start-of-lot report.	S6,F15-->		
		<--S6,F16	Equipment sends report items.
Host commands start-of-lot.	S2,F41-->		
		<--S2,F42	Equipment Acks Start.
		<--S5,F1	Alarm :
NoDevicesPresent			
Host Acks Alarm	S5,F2-->		
			Time Passes.
		<--S6,F11	Event : PortLoaded
Host Acks Event	S6,F12-->		
			Handler cycles devices... ... until ...
		<--S5,F1	Alarm : LoadDeviceFail
Host Acks Alarm	S5,F2-->		

ProcessPause		
Host sends operator to clear jam.		
Host sends resume.	S2,F41-->	
	<--S2,F42	Handler Acks and resumes.
Processing		
		Handler cycles devices... ... until ...
Host asks for Temperature x.	S6,F15-->	
	<--S6,F16	Handler sends Temp.x.
		...
	<--S6,F11	Event : HandlerEmpty
Host Acks Event	S6,F12-->	
Host commands end-of-lot.	S2,F41-->	
	<--S2,F42	Handler acks Rmt.Cmd.
Stopping		
	<--S6,F11	Event : Lot Completed
Host Acks Event	S6,F12-->	
Idle		
Host requests end-of-lot-report.	S6,F15-->	
	<--S6,F16	Handler sends report.

11.3 *Multi-Site Run Scenario* — This is a run scenario with 64 test-sites and optional GEM events all turned on with no errors of any type occurring.

COMMENT	HOST	EQUIPMENT	COMMENT
Host selects GEM Alarms to enable (list).	S5,F3-->		
		<--S5,F54	Alarms xyz ON.
Host selects GEM Events to enable (list).	S2,F37-->		
		<--S2,F38	Events xyz ON.
Host selects Trace Data Item(s).	S2,F23-->		
		<--S2,F24	Trace Data Item x ON.
Host selects process program.	S2,F41-->		
		<--S2,F42	Equipment Ack
Setting Up			
Setup Complete			
		<--S6,F11	Event : PPLoadOk
Host Ack	S6,F12-->		
Process			
Process.SettingUp			
Process.Ready			
Host requests start-of-lot report.	S2,F41-->		
		<--S2,F42	Equip sends report items.
Host commands start-of-lot.	S2,F41-->		
		<--S2,F42	Equip Acks Start. ... for each trace item period.
		<--S6,F1	Trace Event x SEND.
Host receives and Acks Trace Item x.	S6,F2-->		
			... for each port event (tray/ tube).
		<--S6,F11	Event : PortLoaded

```
  Host Acks Event          S6,F12-->

                                            ... Handler cycles devices ...
                                            ... for each device report
                                                internal states.

                                   <--S6,F11       Event : DeviceClearsInput

  Host Acks Event          S6,F12-->

                                   <--S6,F11          Event : DeviceEntersTemp

  Host Acks Event          S6,F12-->

                                   <--S6,F11       Event : DeviceClearsTemp

  Host Acks Event          S6,F12-->

                                   <--S6,F11       Event : DeviceEntersQueue.x

  Host Acks Event          S6,F12-->

                                   <--S6,F11       Event : StartTest.Contactor.x

  Host Acks Event          S6,F12-->

                                   <--S6,F11       Event : EndTestReceived.x

  Host Acks Event          S6,F12-->

                                   <--S6,F11       Event : DeviceBinReceived.x

  Host Acks Event          S6,F12-->

                                   <--S6,F11       Event : DeviceClearsContactor.x

  Host Acks Event          S6,F12-->

                                   <--S6,F11      Event:DeviceEntersUnloadQueue.x

  Host Acks Event          S6,F12-->

                                   <--S6,F11       Event : DeviceUnloaded.HardbinX

  Host Acks Event          S6,F12-->

                                          ... for each full/empty tray/tube on
                                              input,output.

                                   <--S5,F1     Alarm : ContainerReplaceRequest.x

  Host Acks request.       S5,F2-->
```

Note that the messages reporting the above internal states may require sub-addressing of test-sites and ports similar to the tester SEM.

```
                                                            Handler cycles devices...
                                                            ... random messages
                                                            (e.g., ... variable request)

Host asks for variable   S6,F15-->
x (devices tested).

                                   <--S6,F16         Handler sends variable.
                                                     ...
                                                     ... eventually ends

                                   <--S6,F11            Event : HandlerEmpty

Host Acks Event          S6,F12-->

Host commands            S2,F41-->
end-of-lot.

                                   <--S2,F42            Handler acks Rmt.Cmd

Stopping

                                   <--S6,F11               Event : Lot

Completed

Host Acks Event          S6,F12-->

Idle

Host requests            S6,F15-->
end-of-lot report.

                                   <--S6,F16         Handler sends report.
```

12 Additional GEM Requirements

The purpose of this section is to specify any GEM additional capabilities that are required to be supported by this class of equipment.

12.1 *Requirements* — The following GEM additional capabilities required by HSEM are:

- Establish Communications
- Dynamic Event Report Configuration
- Variable Data Collection
- Status Data Collection
- Alarm Management
- Remote Control
- Equipment Constants
- Process Program Management
- Equipment Terminal Services

- Clock
- Spooling
- Control (Host-Initiated)

NOTICE: These standards do not purport to address safety issues, if any, associated with their use. It is the responsibility of the user of these standards to establish appropriate safety and health practices and determine the applicability of regulatory limitations prior to use. SEMI makes no warranties or representations as to the suitability of the standards set forth herein for any particular application. The determination of the suitability of the standard is solely the responsibility of the user. Users are cautioned to refer to manufacturer's instructions, product labels, product data sheets, and other relevant literature respecting any materials mentioned herein. These standards are subject to change without notice.

The user's attention is called to the possibility that compliance with this standard may require use of copyrighted material or of an invention covered by patent rights. By publication of this standard, SEMI takes no position respecting the validity of any patent rights or copyrights asserted in connection with any item mentioned in this standard. Users of this standard are expressly advised that determination of any such patent rights or copyrights, and the risk of infringement of such rights, are entirely their own responsibility.

SEMI E30.3-0698
TESTING EQUIPMENT SPECIFIC EQUIPMENT MODEL (TSEM)

CONTENTS

List of Tables

SEMI E30.3-0698
TESTING EQUIPMENT SPECIFIC EQUIPMENT MODEL (TSEM)

1 Purpose

1.1 This document establishes a Specific Equipment Model for testing equipment (TSEM). The TSEM consists of equipment characteristics and behaviors that apply to this class of equipment and are required to be implemented in addition to the fundamental requirements and additional capabilities specified in the Generic Equipment Model (GEM), SEMI E30. The intent of this document is to facilitate the integration of testing equipment into an automated semiconductor factory. This document accomplishes this by defining an operational model for testing equipment as viewed by a factory automation controller. This definition provides a standard host interface and equipment operational behavior.

2 Scope

2.1 The scope of this document is limited to the definition of testing equipment behavior as perceived by a Semiconductor Equipment Communications Standard (SEMI E5) host that complies with SEMI E30. The document defines the view of the equipment through the SECS communications link but does not define the internal operation of the equipment. It includes a specific processing state model as the basis for the behavior of all equipment of this class.

2.2 This document requires that the SEMI E30 fundamental requirements and applicable additional capabilities have been implemented on the test equipment. This document expands SEMI E30 Standard requirements and capabilities in the areas of the processing state model, collection events, alarm documentation, remote commands, variable items, and process program management.

3 Limitations

3.1 *Communications*

3.1.1 It is required that any TSEM-compliant equipment follow the Communications State Model in SEMI E30. In addition, TSEM-compliant equipment shall support the High-Speed Messaging Service (SEMI E37) Communication Standard sending SEMI E5 messages over TCP/IP to maximize the amount of data available for monitoring from this class of equipment. This specification deals only with the behavior of the tester in communicating with the host. It is recognized that the tester may also have a communications link with a test handler or prober and that the handler and/or prober may also have a communications link with the host. This specification is intentionally non-specific on the communications link requirements between handler and tester to allow the user the greatest amount of flexibility in specific factory configurations.

3.2 *Multi-Head and Multi-Site TSEM Implementation*

3.2.1 This SEM makes some demands and assumptions about the tester when it supports multiple test heads containing multiple sites per head.These requirements are as follows:

- Test systems that are capable of operating more than one virtual (or logical) test system at a time provide the virtual tester configuration data via the VirtualConfig variable specified in Section 8.
- The identification for multi-head and multi-site data (data variable, status variables, events, etc.) will be provided via list structures in SEMI E5 messages as detailed in Section 8.
- In the case where equipment supports more than one virtual (or logical) tester, all events and data items must have distinct CEIDs, DVIDs, and SVIDs for each virtual (or logical) tester.

4 Referenced Documents

4.1 *SEMI Standards*

SEMI E4 — SEMI Equipment Communications Standard 1 Message Transfer (SECS-I)

SEMI E5 — SEMI Equipment Communications Standard 2 Message Content (SECS-II)

SEMI E10 — Standard for Definition and Measurement of Equipment Reliability, Availability, and Maintainability (RAM)

SEMI E30 — Generic Model for Communications and Control of SEMI Equipment (GEM)

SEMI E37 — High-Speed SECS Message Services (HSMS) Generic Services

SEMI E37.1 — High-Speed SECS Message Services Single-Session Mode (HSMS-SS)

SEMI E58 — Automated Reliability, Availability, and Maintainability Standard (ARAMS): Concepts, Behavior, and Services

5 Terminology

5.1 *alignment location* — Site location that individual packaged units are aligned to at the process-site (e.g., electrical test).

5.2 *calibration fixture* — Any electromechanical fixture required to perform system calibration. May consist of multiple components with different part and serial numbers.

5.3 *class* — Classes represent the most coarse view of the test results. At a minimum, there should be two classes defined for each process program: one representing good devices and another representing failed devices.

5.4 *class, hard-bin, and soft-bin* — Equipment is to maintain DVVALs which provide three levels of granularity for test results: class, hard-bin, and soft-bin. Classes, hard-bins, and soft-bins are expected to be defined within a process program, and their names are made available as DVVALs. When device testing has completed, the process program is to determine the class, hard-bin, and soft-bin with which the device is to be associated, based on the results of the testing. Finally, a summation of the number of devices associated with each class, hard-bin, and soft-bin is also maintained throughout the execution of a process program, and these are also made available as DVVALs.

5.5 *diagnostic fixture* — Any electromechanical fixture required to perform system diagnostics. May consist of multiple components with different parts and serial numbers.

5.6 *execution area* — The area from which a current copy of the process program instructions is executed.

5.7 *hard-bin* — Hard-bins represent the typical view of the test results. Within a process program, each hard-bin is associated with a single class. Generally, multiple hard-bins are associated with a particular class.

5.8 *kit* — Specific items of hardware and software as specified by the equipment manufacturer that adapt the equipment for a specific unit or unit package.

5.9 *leadfinger (or substrate connector lead)* — (1) In ceramic packages, an area of refractory metal that has been plated and is designated for the attachment to a process-site. (2) The area of the unit designated for the attachment to a process-site.

5.10 *leadframe* — A sheet metal framework upon which a chip (sometimes chips) is attached, wire-bonded, then molded with plastic epoxy or with ceramic and/or metal.

5.11 *off-line programming (OLP) utility* — Utility to create, edit, and format process programs on a computer as opposed to creating process programs at the equipment.

5.12 *soft-bin* — Soft-bins represent the most detailed view of the test results. Within a process program, each soft-bin is associated with a single hard-bin. Generally, multiple soft-bins are associated with a particular hard-bin.

5.13 *system calibration* — Test system process required to bring the test system into compliance with the test system manufacturer's system specifications.

5.14 *test executive* — The tester software which controls test program execution.

5.15 *test head* — A resource of the tester. The electromechanical interface between the device/unit and the tester.

5.16 *test-site* — A specific site on a test head.

5.17 *test-board* — The electromechanical interface necessary to enable temporary electrical contact between the device/unit to be tested and the tester resource. May consist of multiple components.

5.18 *testing equipment* — An equipment class generally consisting of integrated mechanisms and controls for performing electrical tests of packaged devices and/or wafer die during the manufacturing process.

5.19 *unit* — The functional integrated circuit (or chip) that is to be electrically tested.

5.20 *virtual (or logical) tester* — That portion or portions of the complete test system that is capable of operating as an independent tester in accordance with the state model shown in this document. For a single test system with one test head and a single test-site on the test head, the physical and virtual (or logical) tester are the same. In a two-headed test system, where each head can execute a unique process program autonomously and there is only one test-site on each head, two virtual (or logical) testers may be operating at the same time. If there are multiple test-sites on multiple heads, each capable of autonomous execution of a unique program, there are as many virtual (or logical) testers in operation as there are autonomous sites. The number of virtual (or logical) testers operating at any time depends on how the test system is currently configured (hardware and software) and may range from one to the maximum capability of the particular test system.

6 Requirements

6.1 *State Models*

6.1.1 The purpose is to define the equipment-specific processing state model and other state models necessary to portray the expected operational states of the equipment to enable host tracking and control in place of a local operator.

6.2 *State Model Requirements*

6.2.1 The processing state models in this document are required for implementing a TSEM-compliant tester, in addition to the required state models in SEMI E30. A state model consists of a processing state model diagram, processing state definitions, and a processing state transitions table. A state model represents the host's view of the tester, but not necessarily the actual tester operations. All TSEM state model transitions shall be mapped sequentially into the actual equipment events that satisfy the requirements of those transitions. In certain implementations, the tester may enter a state and have already satisfied all of the conditions required by the TSEM state model for transition to another state. In this situation, the tester makes the required transition without any additional actions.

6.2.2 Some equipment may need to include additional states. However, any additional states must not change the TSEM-defined state transitions. All expected transitions between TSEM states must occur.

6.3 *TSEM Process State Model*

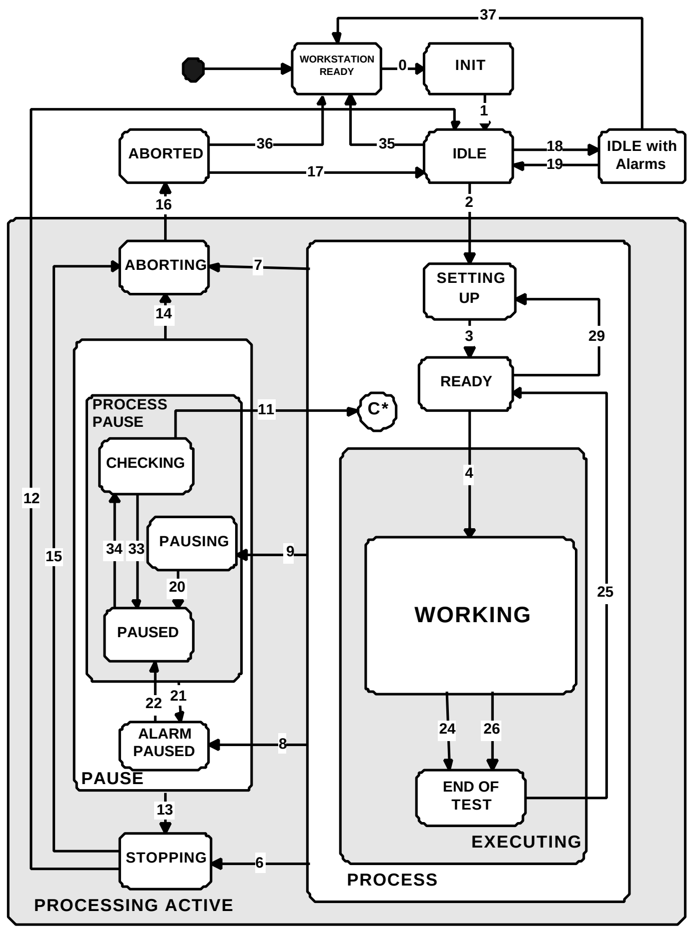


Figure 1
TSEM Processing State Model

6.4 *Description of Tester Processing States*

6.4.1 *ABORTED* — All activity is suspended as a result of an ABORT command. Any alarm and abort conditions must be cleared and verified by an operator before exit from this state.

6.4.2 *ABORTING (PROCESSING ACTIVE Sub-State)* — The tester has received an ABORT command. All activity is suspended. The tester is taking appropriate action to bring itself and material to a "safe" state where possible. Unit or Lot data may be invalid or not available.

6.4.3 *ALARM PAUSED (PAUSE Sub-State)* — An alarm has occurred in the PROCESS or PROCESS PAUSE states, and the tester is waiting for the alarm to be cleared.

6.4.4 *CHECKING (PROCESS PAUSE Sub-State)* — The tester verifies that updates made to the process program are valid. This is a similar procedure to that which is done in SETTING UP. At the successful completion of verification, a transition is made to the process state, based on the process model condition table.

6.4.5 *EXECUTING (PROCESS Sub-State)* — The tester is processing material automatically and can continue to do so without external intervention. This state may include interaction with the host or operator.

6.4.6 *IDLE* — Awaiting a command. IDLE is free of ALARMS and error conditions. A program may or may not be loaded in the execution space during this state.

6.4.7 *IDLE with ALARMS* — An alarm has occurred in the IDLE state, and the tester is waiting for all alarms to be cleared.

6.4.8 *INIT* — Tester initialization is occurring.

6.4.9 *PAUSED (PROCESS PAUSE Sub-State)* — The PROCESS state has been suspended, and the tester is waiting for a command (RESUME, STOP, or ABORT). In this state, the operator may correct error conditions and modify some conditions of the current Process Program selection.

6.4.10 *PAUSING (PROCESS PAUSE Sub-State)* — The current state will be suspended at the completion of the current unit(s), if any, and the tester will be brought to a "safe state."

6.4.11 *PROCESS (PROCESS Sub-State)* — This state is the parent of those sub-states that refer to the preparation and execution of a process program.

6.4.12 *PROCESS PAUSE (PAUSE Sub-State)* — The tester is free of alarm conditions in the PAUSE state.

6.4.13 *READY (PROCESS Sub-State)* — The tester is ready to begin processing and is awaiting a START command.

6.4.14 *SETTING UP (PROCESS Sub-State)* — The tester is satisfying conditions so that processing can begin. This includes the initialization of any process programs and process program-specific calibration. This may be accomplished independently by the tester or may require interaction with the operator and/or host.

6.4.15 *STOPPING (PROCESSING ACTIVE Sub-State)* — The tester has completed a Process Program or has been instructed to stop processing and shall do so at the next opportunity. All necessary cleanup is completed within this state with regard to material, data, control system, etc. Data is preserved. Any error condition is cleared before exiting from this state.

6.4.16 *END OF TEST (PROCESS Sub-State)* — The UNITS testing is complete.

6.4.17 *WORKING (EXECUTING Sub-State)* — The tester is processing a specific unit or units.

6.4.18 *WORKSTATION READY* — The tester workstation is running and ready for tester initialization. The START EXEC remote command is valid in this state.

6.5 *TSEM Processing State Transitions Table*

Table 1. Processing State Transitions Table

#	*Current State*	*Trigger*	*New State*	*Actions*	*Comments*
0	WORK-STATION-READY	Power on	INIT	None	None
1	INIT	All tester initialization is complete with no alarms or error conditions.	IDLE	None	None
2	IDLE	A Process Program is selected.	SETTING UP	Tester-dependent	Commit has been made to setup.
3	SETTING UP	All setup activity has completed, and the tester is ready to receive a START command.	READY	None	The selected Process Program is available for execution.
4	READY	The tester, operator, or host executes a START command, and auto-start is enabled.	WORKING	Begins testing the unit(s) at the test-site(s).	WORKING is an EXECUTING sub-state.
6	PROCESS	The tester has received a STOP command.	STOPPING	The tester completes the current unit(s) before entering the STOPPING state.	The tester begins its cleanup procedure.
7	PROCESS	The tester has received an ABORT command from operator, host, or self-generated.	ABORTING	The tester is put in a "safe" state.	Unit or lot data may be invalid or not available.
8	PROCESS	An alarm occurs.	ALARM PAUSED	PROCESS activity is suspended, and the tester is waiting for all alarms to be cleared.	ALARM PAUSED is a PAUSE sub-state.
9	PROCESS	The tester has received a PAUSE command.	PAUSING	The current state is suspended at the completion of the current unit(s). Any necessary actions to put the tester in a "safe" state will be performed.	PAUSING is a PROCESS PAUSE sub-state.

Table 1. Processing State Transitions Table

11	CHECKING	Parameter checking completes successfully.	STATE based on conditional table.	None	This is a conditional re-entry to the PROCESS state. (See Table 2.)
12	STOPPING	The tester cleanup is complete, and the tester is free of alarms.	IDLE	None	Data is preserved and is valid.
13	PAUSE	The tester has received a STOP command.	STOPPING	The tester proceeds with cleanup.	Data is preserved and is valid.
14	PAUSE	The tester has received an ABORT command.	ABORTING	Any unsafe condition is resolved, if possible.	Data may be invalid or unavailable.
15	STOPPING	The tester has received an ABORT command.	ABORTING	Any unsafe condition is resolved, if possible.	Data may be invalid or unavailable.
16	ABORTING	Unsafe conditions have been resolved, where possible.	ABORTED	The tester is waiting for alarm and ABORT conditions to be cleared.	The only state change allowed is to IDLE. Data is preserved and is valid.
17	ABORTED	An operator has verified that all alarms and abort conditions have been cleared.	IDLE	None	The IDLE state is a "clean" state.
18	IDLE	An alarm is set.	IDLE with ALARMS	The tester waits for all alarms to be cleared.	None
19	IDLE with ALARMS	All alarms have been cleared.	IDLE	None	The IDLE state is free of alarms.
20	PAUSING	The tester has completed Processing the Current unit(s) and achieved a "safe" condition.	PAUSED	The tester is waiting for a command (RESUME, STOP, or ABORT).	None
21	PROCESS PAUSE	An alarm is set.	ALARM PAUSED	The tester waits for all alarms to be cleared or for a STOP or ABORT command.	None
22	ALARM PAUSED	All alarms are cleared.	PAUSED	The tester is waiting for a command (RESUME, STOP, or ABORT).	None

Table 1. Processing State Transitions Table

24	WORKING	The processing of the current unit(s) has completed normally.	END OF TEST	The tester processes end of test data.	"Normal" completion of the test program execution.
25	END OF TEST	Tester is ready to receive a new start of test command.	READY	Waiting for start of next test.	
26	WORKING	The processing of the current unit(s) has completed abnormally.	END OF TEST	The tester processes end of test data.	"Abnormal" completion of the test program execution, etc.
29	READY	New Lot is received by tester.	SETTING UP	The tester performs setup based on the new command.	None
33	CHECKING	Error detected in a new parameter setting.	PAUSED	The tester waits for the parameter correction by operator or host.	None
34	PAUSED	A RESUME command with variable parameters was received.	CHECKING	Validation of the process program parameters begins.	None
35	IDLE	The tester executive has been stopped by the operator.	WORKSTATION READY	Waiting for a START EXECUTIVE command.	None
36	ABORTED	The tester executive has been aborted by the operator.	WORKSTATION READY	Waiting for a START EXECUTIVE command.	None
37	IDLE with ALARMS	The tester executive has been stopped by the operator.	WORKSTATION READY	Waiting for an ALARM Clear and START EXECUTIVE command.	None

6.6 *Process Model Conditions Table*

Table 2. Process Resume Conditions

Condition	*Next State*
Checking determines that process program conditions were changed.	SETTING UP
Previous State WORKING.	END OF TEST
Previous State READY.	READY
Previous State was SETTING UP.	SETTING UP

7 Collection Event List

7.1 *Requirements*

7.1.1 ALL SEMI E30-required Events are required by the TSEM. Since a Processing State Model is required by the TSEM, all state transitions are required Events.

7.1.2 All SEMI E30-required events associated with the SEMI E30 Control, Communications, Alarm, and Spooling State Models are required.

7.1.3 This section of the TSEM lists only those collection events that are not associated with a change of state or those requiring specific data variables (DVVALs) or Reports defined in the TSEM.

7.2 *Collection Event Tables* — The first table contains required events and associated reports. The second table contains required events and associated data variables.

Table 3. Processing State Transitions Requiring Report Levels

Transition	*From State*	*To State*	*Required DVVALs or Report*
SETUP COMPLETE (3)	SETTING UP	READY	Setup Report

Table 4. Other Required Collection Events

Collection Event Name	*Required DVVALs*
LotComplete	See Lot Report.
BoardChg	TestBoardID, CalFixtureID, or DiagFixture ID (Valid in CHECKING and SETTING UP.)
DockStatusChange	DockingStatus
BinDataAvailable	See Bin-Data Report.
SubLotComplete	See SubLot Report.
LotStart	Lot/SubLot Start Report
SubLotStart	Lot/SubLot Start Report

8 Variable Items

The purpose of this section is to define the list of variable items required by the TSEM. Values of these variables will be available to the host through collection event reports and host status queries.

8.1 *Requirements*

8.1.1 All generic variable items defined in SEMI E30 are required by all TSEM equipment.

8.1.2 Variable items required by TSEM are categorized as follows:

- *Common Variables (CVs)* — Variables common to all testers.
- *Configuration-Specific Variables (CSVs)* — Variables associated with a specific configuration of the above equipment class.

8.1.3 Any supplier-defined variables shall be documented in the same format used by this document. The following minimum information is required:

<variable name> **Class:** <ECV, SV, or DVVAL> **Format:** <SML>

Description: <If class = DVVAL, description must contain statement of when data is valid.>

<If format = ASCII, then a length is required. It is assumed to be left-justified unless otherwise noted.>

8.2 *Data Types*

8.2.1 Equipment Constants (ECVs) can be changed by the host using S2,F15. The operator may be able to change some values, but the equipment does not change the values on its own. The value of an equipment constant may be queried by the host at any time, using the S2,F13/14 transaction. They reside in non-volatile memory of the equipment. Equipment constants remain in effect until they are overwritten either by manual entry or by a S2,F15 (NEW EQUIPMENT CONSTANT SEND).

8.2.2 Equipment constants have various uses in TSEM, including the following:

- Equipment offsets that match the performance of several pieces of equipment that would otherwise perform differently due to inherent manufacturing differences. Examples are home values and motion axis scaling factors.
- Setting the configuration of the equipment to allow for different material specifications, equipment options, material flows, frequency of automatic functions, etc. An example is yield check frequency.
- Managing optional machine features. Examples are constants that indicate whether optional features such as automated media stackers are present and control the configuration and function of these optional subsystems when they are present.

8.2.3 Status Variables (SVs) are valid at all times. An SV may not be changed by the host but may be changed by the equipment or operator. The value of status variables may be queried by the host at anytime using the S1,F3/4 or S6,F19/20 transactions.

8.2.4 DVVALs are variables that are valid only upon the occurrence of specific collection events. An attempt to read a variable item at the wrong time does not generate an error, but the data reported may not have relevant meaning.

8.2.5 *Data Item Requirements for Multi-Head, Multi-Site Equipment* — The identification for multi-head and multi-site data (variable items, status variables, events, etc.) is addressed in this specification through the use of status variables. In Table 5, the subscript "v" is used to denote the number of virtual testers, "h" is used to denote the number of tester heads, "s" to denote the number of tester head sites, and "b" to denote the number of bins or classes.

8.3 *Variable Item Table*

Table 5. Variable Item Table

Variable Name	*Category*	*Description*	*Class*	*Format*	*Comments*
Physical Tester Group					
BaseConfig	CV	Base Tester Configuration listing all physical heads and sites.	SV	L, h TestHeadID L, s TestBoardSiteID	Valid in all states. Contains number of possible heads and sites for the tester.
ConfigInfo	CSV	Configuration Information	SV	A[256]	Valid in all states.
ConfigInfoType	CSV	Configuration information source (0 = Auto, 1 = Manual/File)	SV	U4	Valid in all states.
DatalogConfig	CV	Data Log Configuration	SV	A[256]	Valid in all states.
EquipSerialID	CV	Identification of Equipment	SV	A[1..40]	Valid in all states.
HandlerComStatus	CV	Status of comm link between handler/s (0 = Disabled, 1 = 1-way enabled, 2 = 2-way enabled, 3 = Not communicating)	SV	U4	Valid in all states.
LightPoleStatus	CSV	Color/status (i.e., Red/flash)	SV	A[1..16]	Valid in all states.
VirtualConfig	CSV	Current Virtual Configuration listing all virtual IDs.	DVVAL	L, v ...VirtualID L, h TestHeadID L, s TestBoardSiteID	Valid in PROCESS states. Contains active heads and sites for tester. List by virtual tester, head, and site.
Virtual Tester Group					
CalDate	CV	Date of last successful calibration.	SV	A[16]	Valid in all states. List by head.
CalInterval	CS	Time limit between calibrations.	SV	A[16]	Valid in all states. List by head.
CalStatus	CV	Status of last calibration (1 = OK, 0 = Failure)	SV	U4	Valid in all states. List by head.
ClassName	CV	Name tag for high-level class information.	DVVAL	A[1..40]	Valid at BIN-DATA-AVAILABLE state. List by bin.

Table 5. **Variable Item Table**

DiagStatus	CV	Status of last diagnostic (1 = OK, 0 = Failure)	SV	U4	Valid in all states. List by head.
HardBinName	CV	Test-Site Bin Out Name	DVVAL	A[1..40]	Valid at BIN-DATA-AVAILABLE Event. List by bin.
LotProcessingTime	CV	The time since start of current Lot.	DVVAL	A[16]	Valid in Process-sub-state.
OperatorID	CV	Current Operator ID	ECV	A[1..40]	Valid in all states.
ProductID	CV	ID of product for which tester is currently configured.	DVVAL	A[1..40]	Valid in PROCESS states.
SoftBinName	CV	Test-Site Category Name	DVVAL	A[1..40]	Valid at BIN-DATA-AVAILABLE Event. List by bin.
TestSiteInterval	CV	Interval count to generate event for a specific site $_{(n)}$ test.	ECV	U4	Valid in all states.
VirtualID	CV	ID of each virtual configuration.	SV	U4	
		Test Head Group			
CalFixtureID	CV	ID of calibration fixture in current configuration.	SV	A[1..40]	Valid in all states. List by head.
DiagDate	CV	Date of last diagnostic execution.	SV	A[16]	Valid in all states. List by head.
DiagFixtureID	CV	ID of diagnostic fixture in current configuration.	SV	A[1..40]	Valid in all states. List by head.
DiagFixtureList	CV	List of current diagnostic, calibration, and test boards in current configuration.	SV	L, h DiagFixtureID, CalFixtureID, TestBoardIDList	Valid in all states. List by head.
DockingStatus	CSV	Information on handler/ docking status (0 = Yes, 1 = No)	SV	U4	Valid in all states. List by head.
HeadConfig	CV	Number of sites currently configured per head.	SV	U4	Valid in PROCESS states. List by head.
LotID	CV	Lot Identification	DVVAL	A[1..40]	Valid in all states.
StartTestPortID	CSV	Start Test Source (i.e., hand, keyboard, host)	SV	A[1..40]	Valid in all states. List by head.
SubLotID	CV	SubLot Identification	DVVAL	A[1..40]	Valid in all states.

Table 5. Variable Item Table

TestBoardID	CV	ID of current test board.	SV	A[1..30]	Valid in all states. List by head.
TestBoardIDList	CV	List of IDs in current test board configuration.	SV	A[1..32]	Valid in READY state. List by head.
TestHeadID	CV	The ID of a test head.	SV	U4	Valid in READY state. List by head.
TestHeadStatus	CV	(2 = Not Available, 1 = enabled, 0 = disabled)	SV	U4	Valid in IDLE state. List by head.
		Test-Site Group			
ClassCnt	CV	Current count for a particular class.	DVVAL	U4	Valid in PROCESS states. List by head, bin, and site.
ClassID	CV	High-level class ID	DVVAL	U4	Valid in PROCESS states. List by head, bin, and site.
DeviceUnitID	CV	Unit Serial Number	DVVAL	U4	Valid in all states. List by head and site.
ExecutionCnt	CV	Number of test executives since last reset for the current PPID and current LOT at this test-site.	DVVAL	U4	Valid at END OF TEST sub-state. List by head and site.
HardBinCnt	CV	Test-Site Bin Out Count	DVVAL	U4	Valid in PROCESS states. List by head, bin, and site.
HardBinID	CV	Test-Site Bin Out Number	DVVAL	U4	Valid in PROCESS states. List by head, bin, and site.
LotUnitOutput	CSV	Bin output for a specific site by lot.	DVVAL	U4	Valid in PROCESS states. List by head and site.
SoftBinCnt	CV	Test-Site Category Count	DVVAL	U4	Valid in PROCESS states. List by head, bin, and site.
SoftBinID	CV	Test-Site Category Number	DVVAL	U4	Valid in PROCESS states. List by head, bin, and site.
SiteContacts	CSV	Number of contacts for a specific site.	DVVAL	U4	Valid in PROCESS states. List by head and site.
SubLotUnitOutput	CV	Bin output for a specific site by SubLot.	DVVAL	U4	Valid in PROCESS states.

Table 5. Variable Item Table

TestBoardSiteContacts	CSV	Number of contacts for a specific site since last reset.	DVVAL	U4	Valid in PROCESS states. List by head and site.
TestBoardSiteID	CV	X, Y location within testboard or probe card. (First element is the X coordinate, and second element is the Y coordinate.)	SV	U4(x)	Valid in all states. List by head and site.
TestBoardSiteInserts	CSV	Insertion-count on a test board site.	DVVAL	U4	Valid in PROCESSING state. List by head and site.
TestBoardSiteStatus	CV	Test Board Availability (1 = enabled, 0 = disabled)	DVVAL	U4	Valid in PROCESS states. List by head and site.
TestBoardStatus	CV	Test Board Availability (1 = enabled, 0 = disabled)	DVVAL	U4	Valid in PROCESS states. List by head and site.

8.4 *TSEM-Required Reports* — The reports below are required as "canned" or preconfigured reports by TSEM. TSEM does not require the equipment to guarantee the accuracy of data identified in these reports outside the PROCESSING ACTIVE state defined in the TSEM process state model.

8.4.1 *Setup Report* — Table 6 contains variables that are required to be available at the setup complete event.

Table 6. Setup Variables

Variable Name	*Notes*
DiagFixtureList	
LotID	
PPExecName	(per SEMI E30)
ProductID	Current
OperatorID	
TestBoardSiteStatus	
TestHeadID	
HandlerComStatus	
DockingStatus	

8.4.2 *Lot Complete Report* — Table 7 contains variables that must be available and reported at the completion of a lot.

Table 7. Lot Complete Variables

Variable Name	*Notes*
LotID	
PPExecName	(per SEMI E30)
DiagFixtureList	
SoftBinCnt	
HardBinCnt	
ClassCnt	
TestHeadID	
ContactCnt	
ExecutionCnt	
ProductID	
LotUnitOutput	
LotContacts	
LotProcessingTime	
OperatorID	

8.4.3 *SubLot Complete Report* — Table 8 below contains variables that must be available and reported at the completion of a SubLot.

Table 8. SubLot Complete Variables

Variable Name	*Notes*
SubLotID	
LotID	
PPExecName	(per SEMI E30)
DiagFixtureList	
SoftBinCnt	
HardBinCnt	
ClassCnt	
TestHeadID	
SubLotContacts	
SubLotUnitOutput	
ExecutionCnt	
ProductID	
OperatorID	

8.4.4 *Bin-Data-Report* — Table 9 contains variables that must be available once bin data is available.

Table 9. Bin-Data Variables

Variable Name	*Notes*
UnitID	
TestBoardSiteID	
HardBinID	
SoftBinID	
ClassID	

8.4.5 *Process Report* — Table 10 contains variables that must be available when the equipment is in the PROCESSING state.

Table 10. Process Variables

Variable Name	*Notes*
OperatorID	
LotID	
SubLotID	
PPExecName	(per SEMI E30)
OperationType	
DockingStatus	
HandlerCommStatus	
TestBoardSiteStatus	
LotProcessingTime	

8.4.6 *Lot/SubLot Report* — Table 11 contains variables that must be available at the completion of the lot or SubLot events.

Table 11. Lot/SubLot Start Variables

Variable Name	*Notes*
OperatorID	
LotID	
SubLotID	
PPExecName	(per SEMI E30)
DiagFixtureList	
TestHeadID	

8.4.7 *Calibration Report* — Table 12 below contains variables that must be available at the completion of a calibration.

Table 12. Calibration Variables

Variable Name	*Notes*
CalibrationInt	
PPExecName	(per SEMI E30)
OperatorID	
CalStatus	
TestHeadID	
CalFixtureID	

8.4.8 *Diagnostic Report* — Table 13 below contains variables that must be available at the completion of a diagnostic.

Table 13. Diagnostic Variables

Variable Name	*Notes*
OperatorID	
PPExecName	(per SEMI E30)
DiagFixtureList	
TestHeadID	
DiagStatus	

9 Process Program Management

9.1 *Process Program Requirements*

9.1.1 The TSEM requires that the GEM capability of process program management be fully supported for this class of equipment. The TSEM also requires that the process program have a structure that enables the user to build process programs with default conditions that can be overridden for a run. The concepts of process program structure are discussed in the following sections. The TSEM also requires the following:

- Minimum, maximum, and default parameter values must be defined for all process programs.
- *Verification* — When a process program is downloaded to the equipment, the program syntax must be verified by the equipment manufacturer. The process program may be rejected or may fail verification if the equipment is not in an allowable state to accept process program downloads (i.e., IDLE or SETUP).
- *Validation* — The downloaded process parameters must by type- and range-checked before execution.
- PPBODY — The contents of the downloaded process program body may contain the explicit parameters and data necessary for the runtime process program, or the body may contain reference information (i.e., PATH location) on where the explicit data is stored. In the latter case, it is required that the equipment combine or compile the reference data prior to the verification step.
- An error message must be generated from the tester if the process program parameters are outside the range of the machine capability.
- Diagnostic and calibration routines are considered process programs and must be verified and validated the same as a typical process program.

9.2 *Process Program Structure*

9.2.1 A tester process program must contain the following information:

- *Flow Information* — This provides information such as execution order of individual tests.
- *Parametric Information* — This section provides information such as AC/DC/IDC levels and timing.
- *Parameter Options/Values* — This section provides information such as test temperature, part frequency, etc.
- *Data Log Information* — This section provides information such as scope of data collection.
- *Functional Test Information* — This section provides information about vector and patterns.

9.2.2 This information must be supplied in a format that can be referenced as a complete process program. It is emphasized that the TSEM does not specify the exact number, data types, and format of this process program.

9.3 *Methods of Process Program Creation* — The method by which an equipment manufacturer creates a process program may be unique to that manufacturer. However, it is required that the customer at least be given both of the following options for the creation of a process program.

9.3.1 *Off-Line Development* — Using this method, the customer is given a set of software tools (process program compilers, decompilers, and debuggers) that will enable him/her to generate or create a process program using the above mentioned information (flow, parameter, functional test, etc.). The newly generated process program then is downloaded onto a specific tester, is verified, and is now ready to be selected and executed locally by the operator or remotely by the host computer. If this process is used, the supplied software tools should closely mimic or simulate a tester so that a user can create a complete process program. In many situations, minor adjustments or tweaks may be needed to the process program on the equipment before it is completely ready for execution.

9.3.2 *On-Line Development* — The second option made is to enable the user to download the above-mentioned information (tables or files) onto the equipment and create the actual process program on the equipment itself.

10 Remote Commands

The purpose of this section is to identify remote commands, command parameters, and valid commands versus states in the processing state models.

10.1 *Requirements*

- The equipment must support the GEM-required remote commands. (Some of the SEMI E30-required remote commands are restated here to define TSEM-specific requirements.)
- All the remote commands defined by TSEM are required.
- The alphanumeric strings defined by TSEM for remote commands (RMCD) and command parameters (CPNAME) are required.
- If additional remote commands are supported, then the "Remote Command vs. Valid States" matrix must be generated for these additional commands. Place an "X" in the table for each state in which a given command is valid.

10.2 *Remote Command Descriptions*

10.2.1 *ABORT* — This command terminates the current processing. ABORT makes no guarantee about completion of the current test(s). Lot level data will be preserved. Levels of ABORT may be specified (see Table 14 for details).

10.2.2 *PAUSE* — This command transitions the tester to the PAUSING process state when the current test(s) completes processing.

10.2.3 *RESUME* — This command resumes processing from the point where the process was PAUSED. This command is only recognized if the tester is in the PAUSED or CHECKING state.

10.2.4 *PP-SELECT* — This command instructs the tester to copy the indicated Process Program from non-volatile storage to the tester's Process Program execution area. Process Program Variable Parameters can be specified in this command which modify the default values for these Variable Parameters in the Process Program. Process program verification (CHECKING state) must occur when variable parameters accompany this command.

10.2.5 *START* — This command is only available to the host or operator when a process program has been selected and the tester is in the READY processing state. The START command instructs the tester to initiate processing. Parameters can be specified in this command.

10.2.6 *STOP* — This command completes the current test(s), stops in a safe condition, and returns to the IDLE processing state. STOP has the intent of bringing about a normal termination after completion of the current test(s). Parameters can be specified in this command. Lot level data will be preserved.

10.2.7 *UNLOAD-PGM* — This command instructs the tester to unload the specified process program, or programs in the case of multiple heads, from the execution area. Parameters can be specified in this command.

10.2.8 *NEW-LOT* — This command instructs the tester to treat the next units to be processed as a new lot. A new-lot ID and process program variables must accompany the new lot command. This command will force subsequent units to be considered part of the lot. Parameters can be specified in this command.

10.2.9 *CLOSE-LOT* — This command instructs the tester to close the current lot. The next lot will require the setup procedure to be performed.

10.2.10 *CALIBRATE* — This command instructs the tester to execute its calibration program. Parameters can be specified in this command.

10.2.11 *START-EXEC* — This command instructs the tester to start the tester executive.

10.2.12 *STOP-EXEC* — This command instructs the tester to stop the tester executive. Lot level data will be preserved.

10.2.13 *RUN-DIAGNOSTICS* — This command instructs the tester to run the specified preventive maintenance diagnostic routine. Parameters can be specified in this command.

10.2.14 *RUN-CONTINUITY* — This command instructs the tester to run the specified continuity routine. Valid in PAUSE state only. Parameters can be specified in this command.

10.2.15 *ENABLE-SITE* — This command instructs the tester to enable a specified test board site(s). Parameters can be specified in this command.

10.2.16 *DISABLE-SITE* — This command instructs the tester to disable a specified test head site. Parameters can be specified in this command.

10.2.17 *RESET-SITE-CNT* — This command instructs the tester to reset the counts on the specified board test board site(s) to zero.

10.3 *Associated Remote Command Parameters*

Table 14. Remote Command Descriptions

Command	*Parameter*			
Name	*Name*	*Opt./Req.*	*Description*	*Format*
ABORT	"VIRTUALID"	OPT	ID of virtual tester to abort.	U4
	"LOTCLOSE"	OPT	The tester will close out the current lot and enter the ABORTING state.	A[7]
CALIBRATE	"PPID"	OPT	The ID of the program to be used.	A[80]
CLOSE-LOT	"VIRTUALID"	OPT	ID of virtual tester.	U4
DISABLE-SITE	"TESTHEADID"	REQ	ID of test head containing the site.	U4
	"TESTSITEID"	REQ	ID of test-site to be disabled.	U4
	"VIRTUALID"	OPT	ID of virtual tester.	U4
ENABLE-SITE	"VIRTUALID"	OPT	ID of virtual tester.	U4
	"TESTHEADID"	REQ	ID of test head containing the site.	U4
	"TESTSITEID"	REQ	ID of test-site to be enabled.	U4
NEW-LOT	"VIRTUALID"	OPT	ID of virtual tester.	U4
	"LOTID"	REQ	ID of New LOT.	A[1..40]
PAUSE	"VIRTUALID"	OPT	ID of virtual tester.	U4

Table 14. Remote Command Descriptions

PP-SELECT	"VIRTUALID"	OPT	ID of virtual tester.	U4
	"LOTID"	OPT	Lot to be processed with this program.	A[1..40]
RESET-SITE-CNT	"VIRTUALID"	OPT	ID of virtual tester.	U4
	"TESTHEADID"	REQ	ID of test head containing the site.	U4
	"TESTSITEID"	REQ	ID of test-site to be reset.	U4
RESUME	"VIRTUALID"	REQ	ID of virtual tester.	U4
RUN-CONTINUITY	"PPID"	OPT	The ID of the program to be used.	A[80]
RUN-DIAGNOSTICS	"PPID"	OPT	The ID of the program to be used.	A[80]
START	None			
START-EXEC	None			
STOP	"VIRTUALID"	REQ	ID of virtual tester.	U4
	"CLOSELOT	OPT	Automatically close lot.	BOOL
STOP-EXEC	"VIRTUALID"	REQ	ID of virtual tester.	U4
UNLOAD-PGM	"VIRTUALID"	OPT	ID of virtual tester to unload.	U4
	"PPID"	REQ	ID of program to unload.	A[80]
	"ALL"	OPT	Will unload all test programs.	BOOL

10.4 *Remote Commands and TSEM Process Model Mapping* — Table 15 illustrates the relationship between remote commands and states of the TSEM processing state model. An "X" indicates that a command is valid for use in this state. If a remote command is attempted during a non-valid state, the equipment would reject the remote command.

Table 15. Remote Commands vs. Process States

COMMAND / **PROCESSING STATE**	STOP	START	RESUME	UNLOAD-PGM	PP-SELECT	PAUSE	NEW-LOT	STOP-EXEC	START-EXEC	ABORT
INIT									X	
IDLE				X	X			X		
IDLE with Alarms								X		
PROCESSING ACTIVE										
PROCESS										
SETTING UP	X					X	X			X
READY	X	X				X	X			X
ABORTING										
STOPPING										X
EXECUTING										
WORKING	X					X				X
ENDOFTEST	X					X				X
PROCESS PAUSE										
PAUSING										X
PAUSED	X		X				X			X
CHECKING										X
ALARM PAUSED	X									X
WORKSTATION READY									X	
ABORTED										

Table 16. Remote Commands vs. Process States #2

COMMAND / **PROCESSING STATE**	RESET-SITE-COUNT	RUN-CONTINUITY	CLOSE-LOT	RUN-DIAGNOSTICS	CALIBRATE
INIT					
IDLE	X	X	X	X	X
IDLE with Alarms					
PROCESSING ACTIVE					
PROCESS					
SETTING UP	X		X		
READY	X	X	X		
ABORTING					
STOPPING					
EXECUTING					
WORKING					
ENDOFTEST					
PROCESS PAUSE					
PAUSING					
PAUSED	X	X	X		
CHECKING					
ALARM PAUSED					
WORKSTATION READY					
ABORTED					

11 Scenarios

The purpose of this section is to document possible TSEM-specific scenarios illustrating the possible virtual configurations. The example below is for a single tester configured as two virtual testers configured as follows:

```
Physical Tester #1  EquipSerialID = T1000A
2 Virtual Equipment Instances,
         where VirtualID #10 represents
         Tester #0, Head #0, Sites 0 - 3.

         where VirtualID #20 represents
         Tester #0, Head #0, Sites 4 - 7,
         Head #1, ALL Sites (1-3).

Status Variable Request Scenario
Status Variable Request       S1,F3 -->
L,1
  1. U4 1001 (SVID for VirtualConfig)
                                        <--S1,F4 (status variable return)
                                               L,1
                                                 1. L,2 = # of virtual testers
                                                    1. L,2
                                                       1. 10 (VirtualID = 10)
                                                       2. L,1 = # of heads for Tester 10
                                                          1. L,2
                                                             1. 0 (TestHeadID)
                                                              2. L,3 = # of sites for
                                                                  head 1
                                                               1. 01 = TestBoardSiteID #1
                                                               2. 02 = TestBoardSiteID #2
                                                               3. 03 = TestBoardSiteID #3
                                                    2. L,2
                                                       1. 20 (VirtualID = 20)
                                                       2. L,2 = # of heads for Tester 20
                                                          1. L,2
                                                             1. 0 (TestHeadID)
                                                              2. L,4 = # of sites for
                                                                 head 0
                                                               1. 04 = TestBoardSiteID #4
                                                               2. 05 = TestBoardSiteID #5
                                                               3. 06 = TestBoardSiteID #6
                                                               4. 07 = TestBoardSiteID #7
                                                          2. L,2
                                                             1. 1 (TestHeadID)
                                                              2.  L,3 = # of sites for
                                                                 head 1
                                                               1. 01 = TestBoardSiteID #1
                                                               2. 02 = TestBoardSiteID #2
                                                               3. 03 = TestBoardSiteID #3
```

11.1 *Normal Run Scenario* — This is an error-free run of a single lot, with no additional lots queued. The Host determines the available resources of the tester by requesting the status variable, VirtualConfig. This variable returns a list of virtual testers and the resources assigned to each of them.

COMMENT **HOST** **EQUIPMENT** **COMMENT**

The Host would initiate a PP_SELECT using an available resource. PP_SELECT requires at least 2 parameters (PPID and VirtualID). Optional parameters are not shown.

```
Host Command Send              S2,F41-->
1. PP-SELECT
2. L,2
   1. L,2
         1. PPID
         2. A[80] "/home/recipes/tester_recipe"
   2. L,2
         1. VirtualID
         2. U4 10

                                         <--S2,F42    (host command acknowledge)
                                                              L,2
                                                                 1. HCACK "0"
                                                                 2. L,0
```

The tester then starts sending several events back to the host to identify transitions and non-transition events.

```
                                         <--S6,F11            (Event #2 Setting Up)
                               S6,F12-->
                                         <--S6,F11            (Event #3 Ready)
                               S6,F12-->
```

The host then sends a start to the tester. The START command requires the VirtualID only.

```
Host Command Send              S2,F41-->
1. START
2. L,1
   1. L,2
         1. VirtualID
         2. U4 10

                                         <--S2,F42    (host command acknowledge)
                                                              L,2
                                                                 1. HCACK "0"
                                                                 2. L,0
```

The tester then starts sending several events back to the host to identify transitions and non-transition events.

```
                    <--S6,F11              (Event #4 Working)
          S6,F12-->
                    <--S6,F11              (Event # EndOfTest)
          S6,F12-->
                    <--S6,F11              (Event # EndOfTest)
          S6,F12-->
                    <--S6,F11              (Bin data available)
          S6,F12-->
                    <--S6,F11              (Event #25 Ready)
          S6,F12-->
```

After several of these starts and events sequences, the tester notifies the host that the SubLot is complete and then receives close-lot and stop commands from the host.

```
                                <--S6,F11              (Sub-lot complete)
                      S6,F12-->

Host Command Send     S2,F41-->
1. CLOSE_LOT
2. L,0

                                <--S2,F42      (host command acknowledge)
                                                       L,2
                                                          1. HCACK "0"
                                                          2. L,0

                                <--S6,F11              (Lot-complete)
                      S6,F12-->
Host Command Send     S2,F41-->
1. STOP
2. L,0

                                <--S2,F42      (host command acknowledge)
                                                       L,2
                                                          1. HCACK "0"
                                <--S6,F11                 (Event #12 IDLE)
                      S6,F12-->
```

11.2 *Run Diagnostics/Calibration Scenario*

```
Run Diag/Calibration
Host Command Send              S2,F41-->
1. CALIBRATE
2. L,1
   1. L,2
         1. START
         2. A[80] "DIAG-ID"

                                         <--S2,F42    (host command acknowledge)
                                                              L,2
                                                                 1. HCACK "0"
                                                                 2. L,0
                                         <--S6,F11    (Event #2)
                               S6,F12-->
                                         <--S6,F11    (Event #3)
                               S6,F12-->

Host Command Send              S2,F41-->
 1. START
2. L,0

                                         <--S2,F42    (host command acknowledge)
                                                              L,2
                                                                 1. HCACK "0"

                                         <--S6,F11            (Event #4)
                               S6,F12-->
                                         <--S6,F11            (Event #24)
                               S6,F12-->
                                         <--S6,F11            (Diag Report)
                               S6,F12-->

Host Command Send              S2,F41-->
1. STOP
2. L,0

                                         <--S2,F42    (host command acknowledge)
                                                              L,2
                                                                 1. HCACK "0"
```

12 Additional SEMI E30 Requirements

The purpose of this section is to specify any GEM additional capabilities that are required to be supported by this class of equipment.

12.1 *Requirements* — The following GEM additional capabilities required by TSEM are:

- Establish Communications
- Dynamic Event Report Configuration
- Variable Data Collection
- Status Data Collection
- Alarm Management
- Remote Control
- Equipment Constants
- Process Program Management
- Equipment Terminal Services
- Clock
- Spooling
- Control (Host-Initiated)

13 TSEM Unique Capabilities

The purpose of this section is to specify additional capabilities required for the TSEM that are unique to this class of equipment.

13.1 *Test Handling Equipment Common Data* — The purpose of this subsection is to specify test handling variable item data and event data that must be available to the host during the electrical test process. Because equipment configurations vary regarding handling and test equipment, passing control and process information to the host system also can vary. By providing this information from either class of equipment, a greater variety of configurations is available to the user. The handler data identified in this section must be made accessible from the tester manufacturer's interface (i.e., placeholder IDs, events, and commands). The validity of the data will depend on the specific field configuration of the equipment. For example, if the field configuration places the handler as the primary contact to the host, the data identified in this section would not be needed, and the TSEM capability would be disabled. On the other hand, if the field configuration places the tester as the primary contact to the host, the data identified in this section would be necessary, and the TSEM capability then would be enabled.

13.2 *Variable Item Requirements* — Table 17 identifies variable items that must be available from the tester in addition to those identified in Section 8.

Table 17. Common Handler Variable Items

Variable Name	*Category*	*Description*	*Class*	*Format*	*Comments*
EquipID (Handler)	CV	Identification of Handler Equipment (per Head)	SV	A[16]	Valid in all states.
LotID	CV	Lot ID	SV	A[16]	Valid in EXECUTING.
OperatorID	CSV	Current Operator ID	ECV	A[24]	Valid in all states.

13.3 *Collection Event Data Item Requirements* — Table 18 identifies common collection events that the tester must be able to provide to the host, if available from the handler.

Table 18. Common Handler Collection Events

Collection Event Name	*Event #*
LotComplete	Equipment Specific #
SetupComplete	Equipment Specific #
LotStart	Equipment Specific #

13.4 *Host Access to Tester Data Log Information* — TSEM requires the equipment manufacturer to make data log information available to the host via the SEMI E5, Stream 13 data set message. Because equipment configurations vary for data log content and format, the only requirement TSEM makes is that the equipment manufacturer document the format and content of data log information used by the equipment and make that information available across the communications interface via Stream 13. (See SEMI E5.)

RELATED INFORMATION 1

NOTE: This related information is not an official part of SEMI E30.3 and is not intended to modify or supercede the official standard. These notes are presented as possible methods for SEM implementations and are included only as reference material.

R1-1 TSEM ARAMS Sub-State Codes

For TSEM implementations which are also compliant with SEMI E58, the equipment will support the following ARAMS sub-states:

A. PRODUCTIVE

1. 1XYY "PRD/Initialization for process program"

2. 1XYY "PRD/Testing units"

3. 1XYY "PRD/Determining result information"

4. 1XYY "PRD/Cleanup for process program"

B. STANDBY

1. 2XYY "SBY/Waiting for process program selection"

2. 2XYY "SBY/Waiting for start test"

3. 2XYY "SBY/Waiting for user input, equipment-initiated"

4. 2XYY "SBY/Waiting for user input, user-initiated"

NOTE: The *X* sub-state codes are reserved by SEMI E58 for standard codes and the *Y* is for supplier usage.

R1-1.1 *Mapping of TSEM Processing State Model to ARAMS State Model* — The TSEM Processing state model and the ARAMS state model are separate models which must both be maintained and supported if the equipment is to be both TSEM-compliant equipment and ARAMS-compliant. Although these state models are separate, there is a definite relationship between the two. All TSEM-compliant equipment which is also ARAMS-compliant will support the following mapping between the TSEM Processing state model and the ARAMS state model. This mapping only applies while the equipment is performing its intended function, that is, while it is in the Manufacturing superstate defined by ARAMS.

R1-1.2 *ARAMS State Mapping to TSEM Processing States*

Table 19. ARAMS/TSEM Processing State Transitions Table

#	*Current State*	*Trigger*	*New State*	*ARAMS Actions*	*Comments*
0	WORK-STATION-READY	Power on.	INIT	Based on previous ARAMS.	Based on previous ARAMS state.
1	INIT	All tester initialization is complete with no alarms or error conditions.	IDLE	SBY/Waiting for PP selection.	Equipment is capable of performing the intended function.
2	IDLE	A Process Program is selected.	SETTING UP	PRD/Initialization for PP selection.	None
3	SETTING UP	All setup activity has completed, and the tester is ready to receive a START command.	READY	SBY/Waiting for start test.	Start Test may be received.
4	READY	The handler, operator, or host executes a START command, and auto-start is enabled.	WORKING	PRD/Testing Units	None
6	PROCESS	The tester has received a STOP command.	STOPPING	PRD/Cleanup	None
7	PROCESS	The tester has received an ABORT command from operator, host, or self-generated.	ABORTING	PRD/Cleanup	None
8	PROCESS	An alarm occurs.	ALARM PAUSED	UDT	Not capable of performing intended function.
9	PROCESS	The tester has received a PAUSE command.	PAUSING	No state change.	PAUSING is incomplete, so no change.
11	CHECKING	Parameter checking completes successfully.	STATE based on conditional table.	Condition based on conditional state.	Reference # 2, 3, 4, 24, 26, and 29.
12	STOPPING	The tester cleanup is complete, and the tester is free of alarms.	IDLE	SBY/Waiting for PP selection.	Equipment is capable of performing the intended function.

Table 19. ARAMS/TSEM Processing State Transitions Table

13	PAUSE	The tester has received a STOP command.	STOPPING	PRD/Cleanup	None
14	PAUSE	The tester has received an ABORT command.	ABORTING	UDT or PRD Cleanup.	If uncleared alarms exist, then ARAMS state is UDT, else cleanup.
15	STOPPING	The tester has received an ABORT command.	ABORTING	UDT or no state change.	If uncleared alarms exist, then ARAMS state is UDT, else stop-ping.
16	ABORTING	Unsafe conditions have been resolved, where possible.	ABORTED	SBY/Waiting for input or UDT.	If uncleared alarms exist, then ARAMS state is UDT, else wait-ing for clear.
17	ABORTED	An operator has verified that all alarms and abort conditions have been cleared.	IDLE	SBY/Waiting for PP selection.	None
18	IDLE	An alarm is set.	IDLE with ALARMS	UDT	Equipment is NOT capable of performing the intended function.
19	IDLE with ALARMS	All alarms have been cleared.	IDLE	SBY/Waiting for PP selection.	Equipment is capable of performing the intended function.
20	PAUSING	The tester has completed processing the Current unit(s) and achieved a safe condition.	PAUSED	SBY/Waiting for input.	Waiting for resume.
21	PROCESS PAUSE	An alarm is set.	ALARM PAUSED	UDT	Equipment is NOT capable of performing the intended function.
22	ALARM PAUSED	All alarms are cleared.	PAUSED	SBY/Waiting for input.	Alarms cleared, waiting for resume.
24	WORKING	The processing of the current unit(s) has completed normally.	END OF TEST	PRD	ARAMS sub-state is active until the BIN-Data Available event occurs or the next start test is received.

Table 19. ARAMS/TSEM Processing State Transitions Table

25	END OF TEST	Tester is ready to receive a new start of test command.	READY	Either SBY/Waiting for start test or PRD determining results.	PRD/Determining sub-state has entered at END OF TEST. This sub-state is active until BIN-Data Available or until the next start test.
26	WORKING	The processing of the current unit(s) has completed abnormally.	END OF TEST	PRD determining results.	The sub-state is active until BIN-Data Available or until the next start test.
29	READY	New Lot is received by tester.	SETTING UP	PRD/Initialization for PP selection.	New lot requires process program initialization.
33	CHECKING	Error detected in a new parameter setting.	PAUSED	No state change.	None
34	PAUSED	A RESUME command with variable parameters was received.	CHECKING	No state change.	None
35	IDLE	The tester executive has been stopped by the operator.	WORKSTATION READY	No state change.	The equipment may not be aware of this transition. ARAMS specifies the rules to be followed upon reinitialization.
36	ABORTED	The tester executive has been aborted by the operator.	WORKSTATION READY	No state change.	The equipment may not be aware of this transition. ARAMS specifies the rules to be followed upon reinitialization.
37	IDLE with ALARMS	The tester executive has been stopped by the operator.	WORKSTATION READY	No state change.	The equipment may not be aware of this transition. ARAMS specifies the rules to be followed upon reinitialization.

R1-1.3 *Additional ARAMS Capabilities* — ARAMS specifies fundamental ARAMS requirements which must be met to be ARAMS-compliant. ARAMS also specifies additional capabilities which may be provided by ARAMS-compliant equipment. All TSEM-compliant equipment which is also ARAMS-compliant is required to provide the following additional capabilities as defined by ARAMS:

a. Dynamic Event Report Configuration

b. Accumulator Data

c. User-Generated ARAMS Sub-State Table(s)

d. Equipment-Generated ARAMS Sub-State Table(s)

e. User-Generated ARAMS Symptom Table(s)