

SEMI E30-0307

GENERIC MODEL FOR COMMUNICATIONS AND CONTROL OF MANUFACTURING EQUIPMENT (GEM)

This standard was technically approved by the global Information & Control Committee. This edition was approved for publication by the global Audits and Reviews Subcommittee on November 21, 2006. It was available at www.semi.org in February 2007 and on CD-ROM in March 2007. Originally published in 1992; previously published November 2003.

CONTENTS

1 Introduction

- 1.1 Revision History
- 1.2 Scope
- 1.3 Intent

Figure 1, GEM Scope

- 1.4 Overview

Figure 2, GEM Components

- 1.5 Referenced Standards and Documents

2 Terminology

3 State Models

- 3.1 State Model Methodology
- 3.2 Communications State Model

Figure 3, Example Equipment Component Overview

Figure 4, Communications State Diagram

Table 1, Communications State Transition Table

- 3.3 Control State Model

Figure 5, Control State Model

Table 2, CONTROL State Transition Table

- 3.4 Equipment Processing States

Figure 6, Processing State Diagram

Table 3, Processing State Transition Table

4 Equipment Capabilities and Scenarios

- 4.1 Establish Communications
- 4.2 Data Collection

Figure 7, Limit Combination Illustration: Control Application

Figure 8, Elements of One Limit

Figure 9, Limit State Model

Table 4, Limit State Transition Table

- 4.3 Alarm Management

Figure 10, State Diagram for Alarm ALIDn

Table 5, Alarm State Transition Table

Table 6

- 4.4 Remote Control
- 4.5 Equipment Constants
- 4.6 Process Recipe Management

4.7 Material Movement

4.8 Equipment Terminal Services

4.9 Error Messages

4.10 Clock

4.11 Spooling

Figure 11, Spooling State Diagram

Table 7, Spooling State Transition

4.12 Control

5 Data Items

5.1 Data Item Restrictions

5.2 Variable Item List

6 Collection Events

Table 8, GEM Defined Collection Events

7 SECS-II Message Subset

STREAM 1: Equipment Status

STREAM 2: Equipment Control and Diagnostics

STREAM 5: Exception (Alarm) Reporting

STREAM 6: Data Collection

STREAM 7: Process Program Load

STREAM 9: System Errors

STREAM 10: Terminal Services

STREAM 13:

STREAM 14: Object Services

STREAM 15: Recipe Management

STREAM 19: Recipe and Parameter Management

8 GEM Compliance

8.1 Fundamental GEM Requirements

Table 9, Fundamental GEM Requirements

Figure 12, GEM Requirements and Capabilities

8.2 GEM Capabilities

Table 10, Section References for GEM Capabilities

8.3 Definition of GEM Compliance

8.4 Documentation

Figure 13, Host View of GEM

Table 11, GEM Compliance Statement

Table 12, SML Notation

A. Application Notes

A.1 Factory Operational Script

A.1.1 Anytime Capabilities

A.1.2 System Initialization and Synchronization

A.1.3 Production Set-Up

A.1.4 Processing

A.1.5 Post-Processing

A.2 Equipment Front Panel

A.2.1 Displays and Indicators

A.2.2 Switches/Buttons

A.3 Examples of Equipment Alarms

Table A.3, Alarm Examples Per Equipment Configuration

A.4 Trace Data Collection Example

A.5 Harel Notation

Figure A.5.1, Harel Statechart Symbols

Figure A.5.2, Example of OR Substates

Figure A.5.3, Example of AND Substates

A.5.1 State Definitions

A.5.2 Transition Table

Table A.5, Transition Table for Motor Example

A.6 Example Control Model Application

Figure A.6.1, Example of the Simplified “Effective” Control Model

A.7 Examples of Limits Monitoring

A.7.1 Introduction

A.7.2 Examples

Figure A.7.1, Valve Monitoring Example

Figure A.7.2, Environment Monitoring Example

Figure A.7.3, Calibration Counter Example

A.8 Recipe Parameter Modification for Process and Equipment Control

A.8.1 Introduction

A.8.2 Equipment Constants

A.8.3 Example

Figure A.8.1, CMP Single Wafer “Polishing” System with Host Recipe Parameter Modification Capability

Index

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 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.

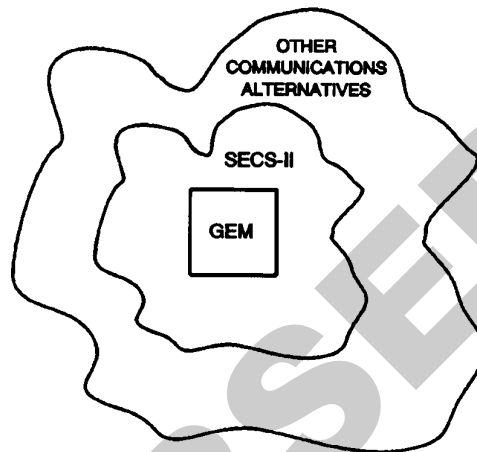


Figure 1
GEM Scope

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.

NOTICE: 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 or other 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 § 8, GEM Compliance. Figure 2 illustrates the components of the GEM standard.

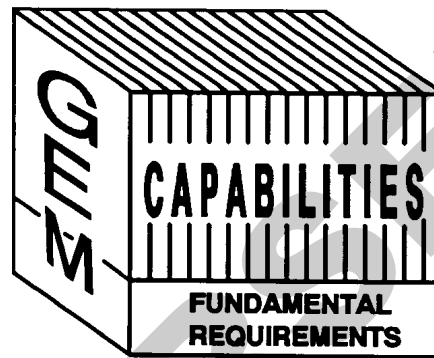


Figure 2
GEM Components

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 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 Referenced Standards and 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

SEMI E139 — Specification for Recipe and Parameter Management (RaP)

1.5.2 Other Documents

Harel, D., “*Statecharts: A Visual Formalism for Complex Systems*,” Science of Computer Programming 8 (1987) 231-274¹.

NOTICE: Unless otherwise indicated, all documents cited shall be the latest published versions.

2 Terminology

2.1 Definitions

2.1.1 *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.

2.1.1.1 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.

2.1.2 *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.

2.1.3 *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.

2.1.4 *communication failure* — a communication failure is said to occur when an established communications link is broken. Such 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.

2.1.5 *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.

2.1.6 *control* — to control is to exercise directing influence.

2.1.7 *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.).

2.1.8 *event* — an event is a detectable occurrence significant to the equipment.

2.1.9 *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. § 8 includes more detail on GEM Compliance.

2.1.10 *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.

2.1.11 *host* — SEMI E4 and SEMI E5 define Host as “the intelligent system that communicates with the equipment.”

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

2.1.13 *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.

2.1.14 *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.

2.1.15 *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.

2.1.16 *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.

2.1.17 *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.

¹ Elsevier Science, P.O. Box 945, New York, NY 10159-0945; Website: <http://www.elsevier.nl>

2.1.18 *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.

2.1.19 *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.

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

2.1.21 *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.

2.1.22 *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.

2.1.23 *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.

2.1.24 *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.

Equipment manufacturers must document the operational behavior of their equipment using state model methodology. State models are discussed in §§ 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 § A.5.

EQUIPMENT	
COMMUNICATIONS	PROCESSING
CONTROL	ALARMS
SPOOLING	LIMITS MONITORIN
PROCESS RECIPES	MATERIAL MOVEMEN
TRACE REPORTS	TERMINAL SERVICES

Figure 3
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 § 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. § 4.1 expands on this section by defining the Establish Communications capability. This model pertains to a logical connection between equipment and host rather than a physical connection.

3.2.1 Terminology — The terms communication failure, connection transaction failure, and communication link 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 connection transaction failure 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 COMMACK² 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 communication link 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 queue is used to cover such stored messages. They are queued when placed within the queue and are dequeued by removing them from this storage.
- Send 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 CommDelay timer 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.

EstablishCommunicationsTimeout 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 initialized to begin timing. The CommDelay timer is initialized only when the state WAIT DELAY is entered.

The CommDelay timer is expired 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 CRA³.

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.
- 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.

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

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

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 4 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 1.

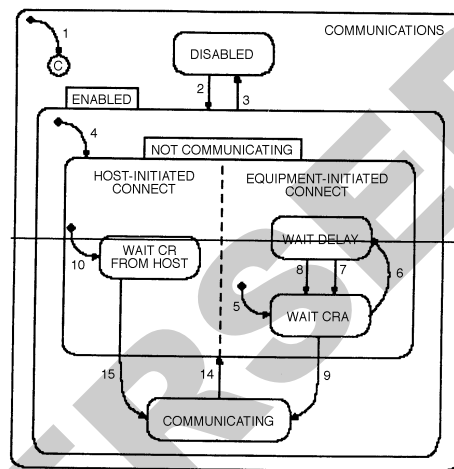


Figure 4
Communications State Diagram

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

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

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 the equipment and the host attempt to establish communications during the same period of time⁵.

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.

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 sent⁶ 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

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.

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

should be sent when a transaction timer timeout occurs⁷ (see § 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 1 contains a full description of the state transitions depicted in Figure 4.

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.

Table 1 Communications State Transition Table

#	Current State	Trigger	New State	Action	Comment
1	(Entry to COMMUNICATIONS)	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 COMMUNICATING)	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 COMMUNICATING)	WAIT CR FROM HOST	None.	Wait for S1,F13 from Host.

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

#	Current State	Trigger	New State	Action	Comment
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 control⁸ 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 5 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 paragraph¹², 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

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.

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.

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.

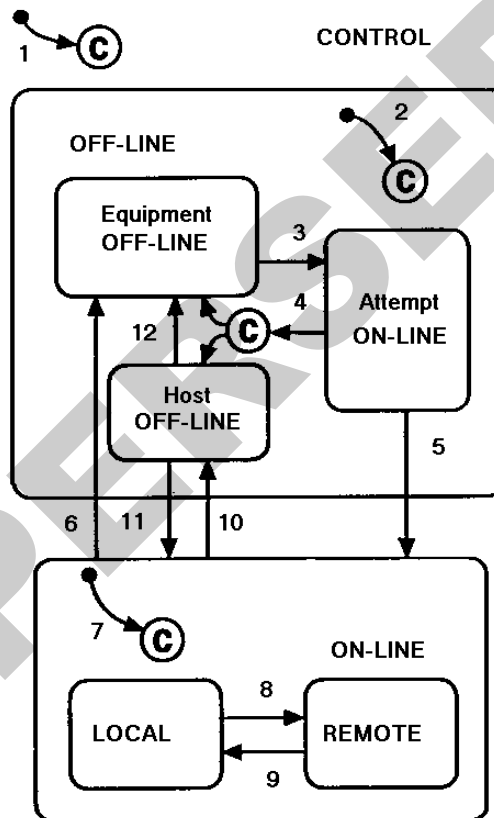


Figure 5
CONTROL State Model

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 Recipe 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 1: 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 § 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 Recipe download,
- select Process Recipe,
- start Process Recipe,
- pause/resume Process Recipe,
- operator assist,
- material movement to/from equipment, and
- equipment-specific commands (on a command-by-command basis if needed).

NOTE 2: 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 2 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 # 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 # 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 # 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.

#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.

#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.

#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 6, 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 Recipe 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 Recipe 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.

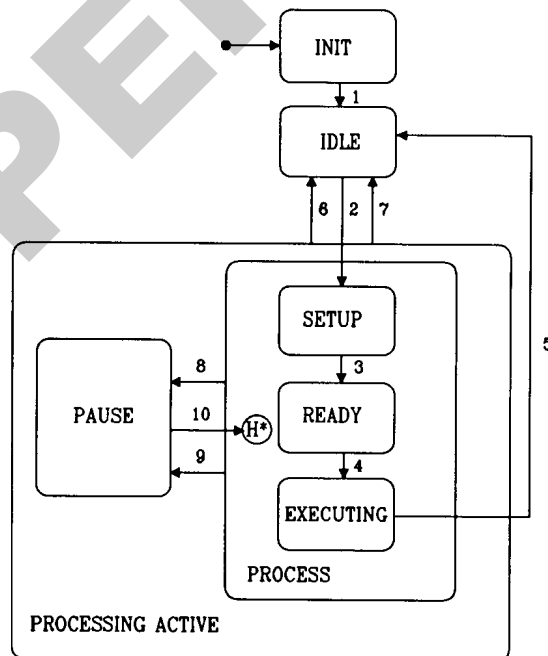


Figure 6
Processing State Diagram

Table 3 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 Recipe 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 SEMI E5 for a full definition of this data item.

EstablishCommunicationsTimeout — an equipment constant used to initialize the interval between attempts to resend an Establish Communications Request. This value specifies the number of seconds for the interval. See SEMI E5 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 § 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 § 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 § 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 Request	S1,F13-->		
		<--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 Acknowledge	S1,F14-->		
		[IF] S1,F14 rcvd w/o timeouts	
		[THEN] exit_loop--SEND	
		[ELSE] Delay for interval in Establish Communications- 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 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 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 SEMI E5 for a full definition of this data item.

Event — a detectable occurrence significant to the equipment.

¹⁵ Communications are established 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.

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 § 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. § 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*

1. The equipment supplier shall provide documentation of all collection events defined on the equipment and the conditions for each event to occur.
2. The equipment supplier shall provide unique CEIDs for each of the various collection events that are available for reporting.
3. 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 § 4.2.1.2) or the equipment's operator console.
4. 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 § 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 3: 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 § 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 VID's 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 § 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--> events			Enable/ disable codes (CEEDs) 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 § 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 4: 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] TOTSMR REPGSZ times [DO] REPGSZ many times: collect SVID ₁ ,...SVID _n data, delay time by DSPER. [END_DO] Send SV ₁ ,...SV _n
Acknowledge receipt	S6, F2-->	<--S6, F1	[END_DO]
Optional: Request trace termination prior to completion (TOTSMR = 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. 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 7.²⁰ This section describes the key aspects of limits monitoring. Detailed implementation examples of limits monitoring are provided as Application Note A.7.

NOTE 5: 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 8. There is a limitation to the values of UPPERDB and LOWERDB which may be stated as:

- $LIMITMAX \geq UPPERDB \geq LOWERDB \geq LIMITMIN$

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 8 shows, these two zones have an area of overlap. This is called the deadband.

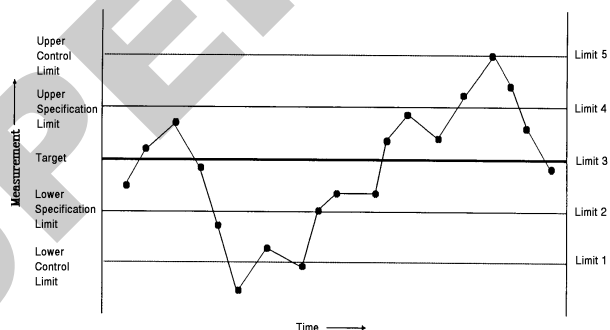


Figure 7
Limit Combination Illustration: Control Application

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

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.

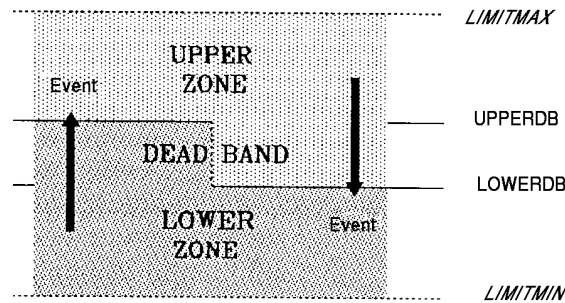


Figure 8
Elements of One Limit

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 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 9. Below, the three states are described more fully, followed by a table defining the transitions.

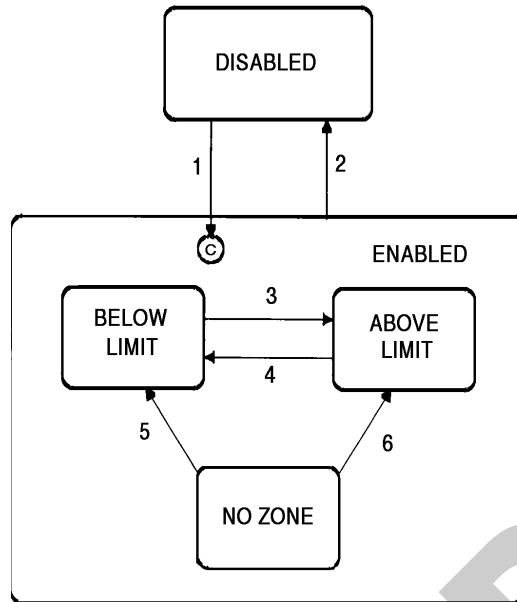


Figure 9
Limit State Model

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:

- $LIMITMAX \geq UPPERDB \geq LOWERDB \geq LIMIT-MIN$;
- 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 8. 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 6: 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 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 $\geq$ UPPERDB	ABOVE LIMIT	None	Zone Transition.
4	ABOVE LIMIT	Variable decreases to be $\leq$ LOWERDB	BELOW LIMIT	None	Zone Transition.
5	NO ZONE	Variable decreases to be $\leq$ LOWERDB	BELOW LIMIT	None	Zone Transition.
6	NO ZONE	Variable increases to be $\geq$ 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 for current variable limit attribute definitions.	S2,F47-->		
		<--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 § 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 § 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.