SEMI E99-1000[E2]
THE CARRIER ID READER/WRITER FUNCTIONAL STANDARD: SPECIFICATION OF CONCEPTS, BEHAVIOR, AND SERVICES

This specification was technically approved by the Global Physical Interfaces and Carriers Committee and is the direct responsibility of the Japanese Physical Interfaces and Carriers Committee. Current edition approved by the North American Regional Standards Committee on August 28, 2000. Initially available at www.semi.org August 2000; to be published October 2000. Originally published February 2000; previously published June 2000.

[E] This document was modified in October 2000 to correct a typographical error. Changes were made to Table 7. This document was modified in December 2001 to correct a typographical error. Changes were made to Table R1-1 and Table R1-2.

1 Purpose

1.1 The purpose of the Carrier ID Reader/Writer Functional Standard effort is to provide a common specification for concepts, behavior, and services (functions) provided by a Carrier ID Reader and a Carrier ID Reader/Writer to an upstream controller. A standard interface will increase interchangeability of Carrier ID Reader/Writers so that users and equipment suppliers have a wider range of choices.

2 Scope

2.1 The Carrier ID Reader/Writer Interface Standard addresses the functional requirements for a generic Carrier ID Reader/Writer interface with an upstream controller.

2.2 The specification includes required behavior and required communications for both a Carrier ID Reader and a Carrier ID Reader/Writer.

2.3 This specification does not require, define, or prohibit asynchronous messages sent by the Carrier ID Reader or Reader/Writer.

2.4 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 standard does not define the specific protocol to be used for the Carrier ID Reader/Writer. Supplements to this standard are required to describe how the functions of the Carrier ID Reader/Writer are implemented for specific protocols.

4 Referenced Standards

4.1 *SEMI Standards*

SEMI E1.9 — Provisional Mechanical Specification for Cassettes Used to Transport and Store 300 mm Wafers

SEMI E15 — Specification for Tool Load Port

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

SEMI E39 — Object Services Standard: Concepts, Behavior, and Services

SEMI E44 — Guide for Procurement and Acceptance of Minienvironments

SEMI E47.1 — Provisional Mechanical Specification for Boxes and Pods Used to Transport and Store 300 mm Wafers

SEMI E62 — Provisional Specification for 300 mm Front-Opening Interface Mechanical Standard (FIMS)

SEMI E87 — Provisional Specification for Carrier Management (CMS)

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 *CIDRW* — represents both the Carrier ID Reader and the Carrier ID Reader/Writer

5.1.2 *FOUP* — Front Opening Unified Pod

5.2 *Definitions*

5.2.1 *attribute* — information about or associated with some entity or object.

5.2.2 *carrier* — 1. a container, such as a FOUP or open cassette, with one or more positions for holding substrates (SEMI E87). 2. any cassette, box, pod, or boat that contains wafers.

5.2.3 *carrier ID* — an identifier for a carrier. A value that uniquely identifies a given carrier in a factory. The identifier may be represented physically with any one of various technologies. For electronic tags with read/write capability, in some cases the user may designate a portion of the data to be used as a carrier ID.

SUPERSEDED

Documents Document Sharing Hub

Full document details: SEMI E99-1000E2

5.2.4 *carrier ID reader* — a unit (subsystem) that detects and decodes data from the ID tag.

5.2.5 *carrier ID reader/writer* — a unit (subsystem) with the functionality of both a carrier ID reader and a carrier ID writer.

5.2.6 *carrier ID tag (tag, ID tag)* — a physical device for storing Carrier ID and other information. There are two basic types of tags, read-only tags and read/write tags.

5.2.7 *carrier ID writer* — a unit which encodes data for and writes it to the carrier ID tag.

5.2.8 *cassette* — an open structure that holds one or more substrates (SEMI E44).

5.2.9 *controller* — a system that provides control (performs required operations when certain conditions occur or when interpreting and acting upon instructions) and communicates with a higher level manager. Controllers exist at all levels within a factory. Examples of controllers include the Multiple ID Reader/Writer Controller, the Equipment Controller, and the Load Port Controller.

5.2.10 *front-opening unified pod (FOUP)* — a box (that complies with SEMI E47.1) with a non-removable cassette (so that its interior complies with SEMI E1.9) and with a front-opening interface (that mates with a FIMS port that complies with SEMI E62).

5.2.11 *fundamental requirements* — the requirements for information and behavior that must be satisfied for compliance to a standard. Fundamental requirements apply to specific areas of application, objects, or services.

NOTE 2: All portions of the carrier ID reader/writer specification are considered to be fundamental requirements unless explicitly described as optional. See also optional requirement.

5.2.12 *load port* — the interface location on a tool where wafer carriers are delivered. It is possible that wafers are not removed from, or inserted into, the carrier at this location.

5.2.13 *message interleaving* — the practice of sending a new message request before receiving the reply to an earlier request.

5.2.14 *multiple ID reader/writer controller* — a unit controlling the Reader/Writer function of one or multiple ID Reader/Writer Heads, communicates the command/data with the equipment controller or the equivalent controller such as Load port Controller in the equipment configuration.

5.2.15 *optional capability* — a specification that is not required for an implementation to be compliant to a standard. The supplier developing the CIDWR has the option to provide these additional capabilities or not depending on supplier's product configuration. See also fundamental requirement.

5.2.16 *reader/writer head* — a structured portion which functions to detect the ID code and/or to write the ID code. The ID reader/writer unifying a head function inside its body can be placed as a head. The ID reader/writer not unifying a head function will be located separately from the head.

5.2.17 *tag fault* — any condition that causes errors when reading or writing to the tag, including power faults and tag damage.

5.2.18 *upstream controller* — a controller that directs the Carrier ID Reader/Writer through the communication interface.

5.3 *Data Types*

5.3.1 *enumerated* — may take on one of a limited set of possible values. These values may be given logical names, but they may be represented by any single-item data type.

5.3.2 *form* — type of data: positive integer, unsigned integer, integer, floating point (float), enumerated, boolean, text, formatted text, structure, list, ordered list.

5.3.3 *integer* — may take on the value of any negative or unsigned integer. Messaging protocol may impose a limit on the range of possible values.

5.3.4 *structure* — a specific set of items, of possibly mixed data types, in a specified arrangement.

5.3.5 *text* — a character string. Messaging protocol may impose restrictions, such as length or ASCII representation.

5.3.6 *unsigned integer* — may take on the value of any positive integer or zero. Messaging protocol may impose a limit on the range of possible values.

6 Conventions

6.1 *State Model Methodology*

6.1.1 This document uses the Harel state chart convention for describing dynamic operation of defined objects. The outline of this convention is described in an attachment of SEMI E30. The official definition of this convention is described in "State Charts: A Visual Formalism for Complex Systems" written by D. Harel in Science of Computer Programming 8, 1987[1].

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

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E99-1000E2

6.1.2 Transition tables are provided in conjunction with the state diagrams to explicitly describe the nature of each state transition. A transition table contains columns for Transition number, Previous State, Trigger, New State, Actions, and Comments. The "trigger" (column 3) for the transition occurs while in the "previous" state. The "actions" (column 5) includes a combination of:

- Actions taken upon exit of the previous state.
- Actions taken upon entry of the new state.
- Actions taken which are most closely associated with the transition.

6.1.2.1 No differentiation is made between these cases.

6.2 *Object Notation*

6.2.1 The object models in Related Information 2 use the Object Modeling Technique (OMT) developed by Rumbaugh, James, et al, in Object-Oriented Modeling and Design.[2] An overview of this notation is provided in SEMI E39, Object Services Standard: Concepts, Behavior, and Services.

Num	*Previous State*	*Trigger*	*New State*	*Actions*	*Comments*

6.3 *Service Message Representation*

6.3.1 Services are functions or methods that may be provided by either the equipment or the host. A service message may be either a request message, which always requires a response, or a notification message, that does not require a response.

6.3.2 *Service Definition*

6.3.2.1 A service definition table defines the specific set of messages for a given service resource, as shown in the following table:

Message Service Name	*Type*	*Description*

Type can be either "N" = Notification or "R" = Request & Response.

6.3.2.2 Notification type messages are initiated by the service provider (e.g., the equipment) and the provider does not expect to get a response from the service user. Request messages are initiated by a service user (e.g., the host). Request messages ask for data or an activity from the provider. Request messages expect a specific response message (no presumption on the message content).

6.3.3 *Service Parameter Dictionary*

6.3.3.1 A service parameter dictionary table defines the description, format and its possible value for parameters used by services, as shown in the following table:

Parameter Name	*Description*	*Format: Possible Value*

6.3.3.2 A row is provided in the table for each parameter of a service.

6.3.4 *Service Message Definition*

6.3.4.1 A service message definition table defines the parameters used in a service, as shown in the following table:

2 James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William Lorensen, Object-Oriented Modeling and Design, Englewood Cliffs, New Jersey: Prentice-Hall, 1991.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E99-1000E2

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*

6.3.4.2 The columns labeled REQ/IND and RSP/CNF link the parameters to the direction of the message. The message sent by the initiator is called the "Request". The receiver terms this message the "Indication" or the request. The receiver may then send a "Response" which the original sender terms the "Confirmation".

6.3.4.3 The following codes appear in the REQ/IND and RSP/CNF columns and are used in the definition of the parameters (e.g., how each parameter is used in each direction):

Code	*Description*
M	Mandatory Parameter — Must be given a valid value.
C	Conditional Parameter — May be defined in some circumstances and undefined in others. Whether a value is given may be completely optional or may depend on the value of the other parameter.
U	User-Defined Parameter.
-	The parameter is not used.
=	(For response only.) Indicates that the value of this parameter in the response must match that in the primary (if defined).

7 Overview

7.1 The Carrier ID Reader/Writer Model defines the behavior and services (functions) for both Carrier ID Readers and Carrier ID Reader/Writers. The Carrier ID Reader/Writer is a small intelligent system, typically used as a subsystem within equipment.

7.1.1 The primary functionality of the Carrier ID Reader/Writer is to read the identifier of the carrier (Carrier ID) from the Carrier ID tag. Various technologies may be used to encode the Carrier ID and to read it. Some technologies do not allow data to be written.

7.1.2 The acronym CIDRW is used to refer to both the Carrier ID Reader and the Carrier ID Reader/Writer. The requirements for the Carrier ID Reader are a subset of the requirements for the Carrier ID Reader/Writer.

7.1.3 An object model for the CIDRW is provided in Related Information 2 – Object Model.

7.2 *Number of Heads*

7.2.1 A Reader/Writer Head is a device that is positioned on a load port for reading or reading/writing information from a Carrier ID tag. A Carrier ID Reader/Writer provides one or more ID Reader/Writer Heads and is connected to an upstream controller by a single interface. This allows the upstream controller to control either one head or multiple heads using the same interface specification.

7.2.2 *Single Head Configuration*

7.2.2.1 In the case of a single head, the head may be presented as an integrated part of the CIDRW.

7.2.3 *Multiple Head Configuration*

7.2.3.1 In the case of multiple heads, some services are logically performed by the CIDRW, and the individual heads logically perform others. The individual heads are numbered sequentially and may be referenced individually by the upstream controller. Note that the upstream controller does not communicate directly with the heads. All communications are between the upstream controller and the CIDRW unit.

7.2.3.2 In the multi-head case, the CIDRW shall allow independent control of the heads. Multiple transactions invoking services performed by the individual heads may be open at the same time. For example, when a read command is sent to one head, the host can send additional commands such as a read command and a status confirmation command to another head before the first head sends the response to the first command.

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E99-1000E2

7.3 *Upstream Controller*

7.3.1 The CIDRW provides certain services when requested by the upstream controller. The upstream controller sends a message requesting the specified service, and the CIDRW sends a message with the response.

7.3.2 This standard assumes that an upstream controller initiates each service message to the CIDRW, and the CIDRW sends its response to the upstream controller. The upstream controller must watch the response time to monitor the communication timeout. This standard will define the recommended method for handling exceptions caused by timeouts.

7.3.3 In addition, some CIDRW may provide asynchronous notification to the upstream controller. Examples of this include, but are not limited to, the detection of a fault condition resulting in an alarm. Asynchronous notification is not required for compliance to this standard.

8 Attributes

8.1 An attribute is an item of information about an entity that is maintained and is available by request. There is certain information concerning the CIDRW that is of potential interest to the upstream controller, including the manufacturer, the model, and the serial number of the device. This information is considered as attributes of the CIDRW and is available on request.

8.2 *CIDRW Attribute Definition Table*

8.2.1 Table 1 defines the attributes of the Carrier ID Reader/Writer subsystem.

Table 1 CIDRW Attribute Definitions

Attribute Name	*Description*	*Access*	*Reqd*	*Form*
Fundamental				
Configuration	Number of heads.	RO	Y	Text
Alarm Status	Current CIDRW substate of ALARM STATUS.	RO	Y	Enumerated: 0 = NO ALARMS 1 = ALARMS
Operational Status	Current CIDRW substate of OPERATIONAL.	RO	Y	Enumerated: IDLE BUSY MAINTENANCE
SoftwareRevisionLevel	Revision (version) of software.	RO	Y	Text
Optional				
DateInstalled	Date the subsystem was installed.	RO	N	Protocol dependent.
Device Type	Identifiers subsystem as either a Carrier ID Reader or a Carrier ID Reader/Writer.	RO	N	Text
HardwareRevisionLevel	Revision number of the hardware.	RO	N	Text
MaintenanceData	Supplier dependent.	RO	N	Text
Manufacturer	The name or identifier of the manufacturer.	RO	N	Protocol dependent.
ModelNumber	The manufacturers model designation.	RO	N	Text
SerialNumber	Subsystem serial number assigned by manufacturer.	RO	N	Protocol dependent.

8.3 *Read/Write Head Attributes*

8.3.1 Table 2 defines the attributes for the Read/Write Head.

NOTE 3: In the case of an integrated single head, the attributes in Table 2 are regarded as an extension of Table 1.

8.3.2 In the case of multiple heads, it must be possible to distinguish between the attributes of different heads.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E99-1000E2

Table 2 Read/Write Head Attribute Definitions

Attribute Name	*Description*	*Access*	*Reqd*	*Form*
Fundamental				
HeadStatus	The current state.	RO	Y	Enumerated: IDLE, BUSY, NOT OPERATING
HeadID	Head number 0–31	RO	Y if multi-head	Text. 2 digits.
Optional				
Cycles	Number of read and write operations performed.	RO	N	Unsigned integer
HeadCondition	The current Maintenance status.	RO	N	Enumerated: No alarms Needs Maintenance Read/Write fault Read/Write rate fault No power.
HeadDateInstalled	Date this head was installed.	RO	N	Protocol dependent.
HeadMaintenance-Data	Supplier dependent	RO	N	Text

9 State Models

9.1 To facilitate independent control of the individual heads, this section defines two separate state models, one for CIDRW subsystem and one to be applied to each of the individual heads.

9.2 *CIDRW State Model*

9.2.1 This section defines the state model for the CIDRW subsystem. Figure 1 shows the diagram for this model.

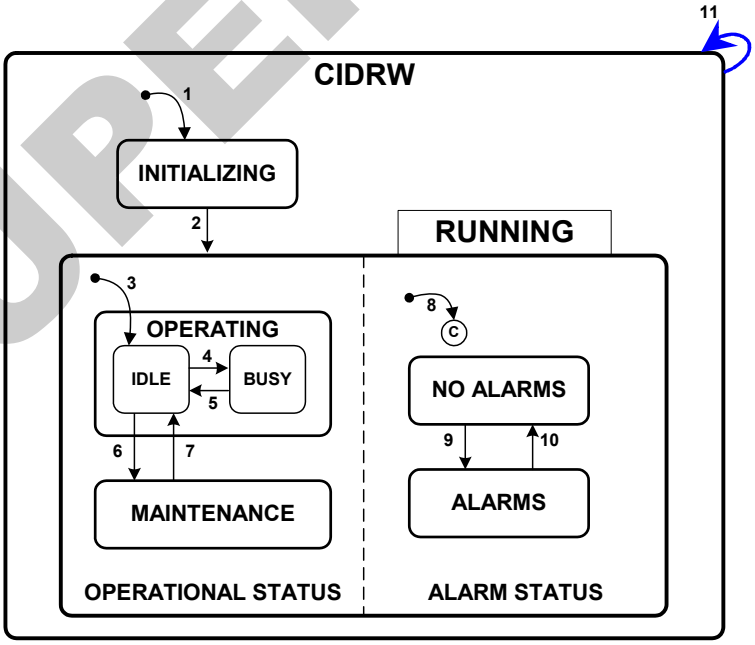

Figure 1
CIDRW State Model Diagram

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E99-1000E2