# SEMI E118-1104[E]
# SPECIFICATION FOR WAFER ID READER COMMUNICATION INTERFACE — THE WAFER ID READER FUNCTIONAL STANDARD: CONCEPTS, BEHAVIOR AND SERVICE

This specification was technically approved by the Global Information and Control Committee and is the direct responsibility of the Japan Information and Control Committee. Current edition approved by the Japan Regional Standards Committee on July 19, 2002. Initially available at www.semi.org October 2002; to be published November 2002.

[E] This standard was editorially modified in February 2005 to correct errors in Figure R2-1.

**NOTICE:** The designation of SEMI E118 was updated during the 1104 publishing cycle to reflect revisions to SEMI E118.1.

INACTIVE

## 1 Purpose

1.1 The purpose of the Wafer ID Reader Functional Standard is to provide a common specification for concepts, behavior, and services (functions) provided by a Wafer ID Reader to an upstream controller.

1.2 A standard interface will increase interchange-ability of Wafer ID Readers so that users and equipment suppliers have a wider range of choices.

## 2 Scope

2.1 The Wafer ID Reader Functional Standard addresses the functional requirements for a generic Wafer ID Reader interface with an upstream controller.

2.2 The specification includes required behavior and required communications for a Wafer ID Reader.

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

2.4 This specification intents to read the Wafer ID specified by SEMI Standard and accessed in the equipment where the wafers transfer in / out is executed with a carrier unit.

**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.

## 3 Limitations

3.1 This standard does not define the specific protocol to be used for the Wafer ID Reader. Supplements to this standard are required to describe how the functions of the Wafer ID Reader are implemented for specific protocols.

## 4 Referenced Standards

4.1 *SEMI Standards*

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 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 — Specification for Carrier Management (CMS)

SEMI E90 — Specification for Substrate Tracking

SEMI E99 — The Carrier ID Reader/Writer Functional Standard: Specification of Concepts, Behaviors, and Services

SEMI E101 — Provisional Guide for EFEM Functional Structure Model

SEMI T7 — Specification for Back Surface Marking of Double-Side Polished Wafers With a Two-Dimensional Matrix Code Symbol

4.2 *ISO/IEC Standard*[1]

ISO/IEC 16022 — International Symbology Specification – Data Matrix

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

1 International Organization for Standardization, ISO Central Secretariat, 1, rue de Varembé, Case postale 56, CH-1211 Geneva 20, Switzerland. Telephone: 41.22.749.01.11; Fax: 41.22.733.34.30 Website: www.iso.ch



Documents Document Sharing Hub

Full document details: SEMI E118-1104E

## 5 Terminology

5.1 *Abbreviations and Acronyms*

5.1.1 *FOUP* — Front Opening Unified Pod

5.1.2 *WIDR* —Wafer ID Reader

5.2 *Definitions*

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

5.2.2 *carrier* — any cassette, box, pod, or boat that contains wafers.

5.2.3 *cassette* — an open structure that holds one or more substrates [SEMI E44].

5.2.4 *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 Wafer ID Reader Controller, the Equipment Controller.

5.2.5 *data matrix code symbol* — a two-dimensional array of square cells arranged in contiguous rows and columns. In certain ECC200 symbols, data regions are separated by alignment patterns. The data region is surrounded by a finder pattern [ISO/IEC 16022].

5.2.6 *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).

5.2.7 *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 1: All portions of the Wafer ID Reader specification are considered to be fundamental requirements unless explicitly described as optional. See also optional requirement.

5.2.8 *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.

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

5.2.10 *Multiple Wafer ID Reader controller* — a unit controlling the Reader function of one or multiple ID Reader Heads, communicates the command/data with the equipment controller or the equivalent controller such as equipment controller.

5.2.11 *optional capability* — a specification that is not required for an implementation to be compliant to a standard. See also fundamental requirement.

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

5.2.13 *substrate handler* — a physical subsystem which transfers substrates between the carriers and the process part of the equipment.

5.2.14 *upstream controller* — a controller that directs the Wafer ID Reader through the communication interface.

5.2.15 *Wafer ID* — an identifier for a wafer. A value that uniquely identifies a given wafer in a factory. The identifier may be represented physically with LASER technologies, etc.

5.2.16 *Wafer ID mark* — a physical structure for storing Wafer ID and other information.

5.2.17 *Wafer ID Reader* — a unit (subsystem) that detects and decodes data from the Wafer ID mark, and that communicates with the upstream controller. Wafer ID Reader may be composed of Multiple Wafer ID Reader controller and reader heads.

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.

INACTIVE



Documents
Document Sharing Hub
Full document details: SEMI E118-1104E

## 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[2].

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[3]. 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:

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

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

INACTIVE



Documents Document Sharing Hub
Full document details: SEMI E118-1104E

| *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*

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

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

6.3.5.1 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".

6.3.5.2 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 Wafer ID Reader Model defines the behavior and services (functions). The Wafer ID Reader is a small intelligent system, typically used as a subsystem within equipment.

7.1.1 The primary functionality of the Wafer ID Reader is to read the identifier of the wafer (Wafer ID).

7.1.2 The acronym WIDR is used to refer to the Wafer ID Reader.

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

7.2 *Number of Heads*

7.2.1 A WIDR provides one or more ID reader 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 WIDR.

7.2.3 *Multiple Head Configuration*

7.2.3.1 In the case of multiple heads, some services are logically performed by the WIDR, 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 WIDR unit.

7.2.3.2 In the multi-head case, the WIDR 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

INACTIVE



Documents
Document Sharing Hub
Full document details: SEMI E118-1104E

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.

7.3 *Upstream Controller*

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

7.3.2 This standard assumes that an upstream controller initiates each service message to the WIDR, and the WIDR 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 WIDR 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 WIDR 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 WIDR and is available on request.

8.2 *WIDR Attribute Definition Table*

8.2.1 Table 1 defines the attributes of the WIDR subsystem.

**Table 1 WIDR Attribute Definitions**

| *Attribute Name* | *Description* | *Access* | *Reqd* | *Form* |
|---|---|---|---|---|
| *Fundamental* | | | | |
| Configuration | Number of heads. | RO | Y | Text |
| AlarmStatus | Current WIDR substate of ALARM STATUS. | RO | Y | Enumerated:<br>0 = NO ALARMS<br>1 = ALARMS |
| OperationalStatus | Current WIDR substate of OPERATIONAL. | RO | Y | Enumerated:<br>IDLE<br>BUSY<br>MAINTENANCE |
| SoftwareRevisionLevel | Revision (version) of software. | RO | Y | Text |
| *Optional* | | | | |
| DateInstalled | Date the subsystem was installed. | RO | N | Protocol dependent. |
| DeviceType | Identifiers subsystem as a WIDR. | RO | N | WIDR (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 *Reader Head Attributes*

8.3.1 Table 2 defines the attributes for the Reader Head.

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

INACTIVE



Documents Document Sharing Hub
Full document details: SEMI E118-1104E

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

**Table 2 Reader Head Attribute Definitions**

| *Attribute Name* | *Description* | *Access* | *Reqd* | *Form* |
|---|---|---|---|---|
| *Fundamental* | | | | |
| HeadStatus | The current state. | RO | Y | Enumerated:<br>IDLE<br>BUSY<br>NOT OPERATING |
| HeadID | Head number 0–31. | RO | Y if multi-head | Text. 2 digits. |
| *Optional* | | | | |
| HeadCondition | The current Maintenance status. | RO | N | Enumerated:<br>No alarms<br>Needs Maintenance<br>Read fault |
| 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 WIDR subsystem and one to be applied to each of the individual heads.

9.2 *WIDR State Model*

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

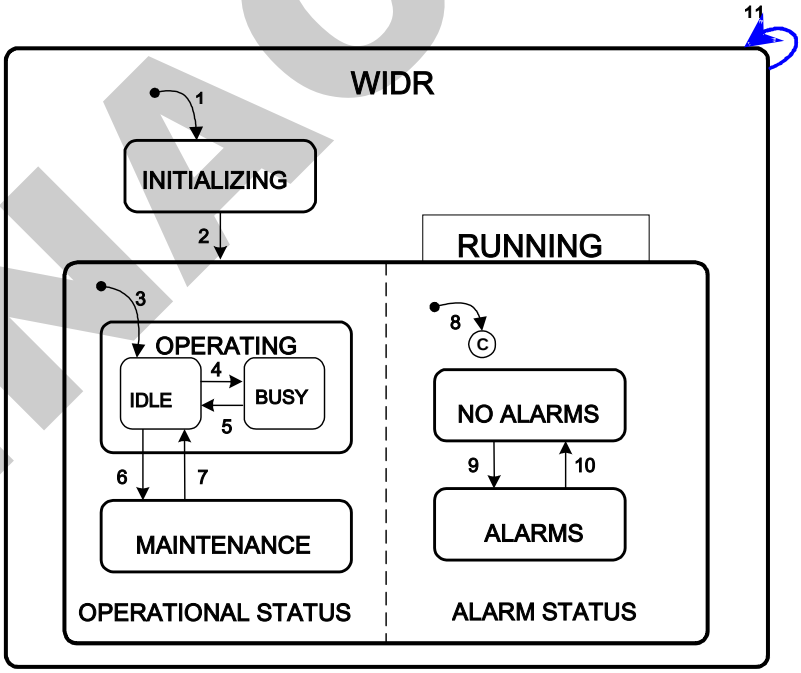


**Figure 1**
**WIDR State Model Diagram**

INACTIVE



Documents
Document Sharing Hub
Full document details: SEMI E118-1104E