9.3.2 *Sensor Actuator Controller (SAC)* — The presentation of the extended ISPM Sensor Actuator Controller (SAC) object attributes and services are as indicated in Table 27.

Table 27 SAC Object Instance Attributes and Services

Sensor, Actuator, Controller Object (SAC) *Class ID = 02, Instance ID = 01*		
Attributes		
ID	*Attribute Name*	*CDM Tag*
65	Number of Bins	SacA65
66	Count Mode	SacA66
67	Duration	SacA67
Services		
ID	*Service Name*	*CDM Tag*
33	Clear Counts	SacS33

9.3.3 *Sensor-AI-LCS* — The presentation of the Sensor Analog Input Laser Current Sensor (Sensor-AI-LCS) object attributes and services are as indicated in Table 28.

Table 28 Sensor-AI-LCS Object Instance Attributes and Services

Sensor-AI-LCS *Class ID = 137, Instance ID = 1 through r*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
128	Reading Valid	LcsA1
129	Full Scale	LcsA2
130	Alarm Settling Time	LcsA3
131	Warning Settling Time	LcsA4
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.3.4 *Sensor-AI-SLS* — The presentation of the Sensor Analog Input Stray Light Sensor (Sensor-AI-SLS) object attributes and services are as indicated in Table 29.

Table 29 Sensor-AI-SLS Object Instance Attributes and Services

Sensor-AI-SLS *Class ID = 138, Instance ID = 1 through r*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
128	Reading Valid	SlsA1
129	Full Scale	SlsA2
130	Alarm Settling Time	SlsA3
131	Warning Settling Time	SlsA4
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.3.5 *Sensor-AI-MNS* — The presentation of the Sensor Analog Input Medium Noise Sensor (Sensor-AI-MNS) object attributes and services are as indicated in Table 30.

Table 30 Sensor-AI-MNS Object Instance Attributes and Services

Sensor-AI-MNS *Class ID = 139, Instance ID = 1 through r*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
128	Reading Valid	MnsA1
129	Full Scale	MnsA2
130	Alarm Settling Time	MnsA3
131	Warning Settling Time	MnsA4
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.3.6 *Sensor-AI-Counter* — The presentation of the Sensor Analog Input Counter (Sensor-AI-Counter) object attributes and services are as indicated in Table 31.

Table 31 Sensor-AI-Counter Object Instance Attributes and Services

Sensor-AI-Counter *Class ID = 140, Instance ID = 1 through r*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
128	Reading Valid	CounterA1
129	Full Scale	CounterA2
130	Alarm Settling Time	CounterA3
131	Warning Settling Time	CounterA4
132	Upper Size	CounterA5
133	Lower Size	CounterA6
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.3.7 *Assembly-ISPM#1* — The presentation of the Assembly #1 In-Situ Particle Monitor (Assembly-ISPM#1) object attributes and services are as indicated in Table 32.

Table 32 Assembly-ISPM#1 Object Instance Attributes and Services

Assembly-ISPM#1 *Class ID = 141, Instance ID = 01 through r*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
--	No additional attributes defined	--
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.3.8 *Assembly-ISPM#2* — The presentation of the Assembly #2 In-Situ Particle Monitor (Assembly-ISPM#2) object attributes and services are as indicated in Table 33.

Table 33 Assembly-ISPM#2 Object Instance Attributes and Services

Assembly-ISPM#2 *Class ID = 142, Instance ID = 01 through r*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
--	No additional attributes defined	--
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.3.9 *Assembly-ISPM#3* — The presentation of the Assembly #3 In-Situ Particle Monitor (Assembly-ISPM#3) object attributes and services are as indicated in Table 34.

Table 34 Assembly-ISPM#3 Object Instance Attributes and Services

Assembly-ISPM#3 *Class ID = 143, Instance ID = 01 through r*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
--	No additional attributes defined	--
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.3.10 *Assembly-ISPM#4* — The presentation of the Assembly #4 In-Situ Particle Monitor (Assembly-ISPM#4) object attributes and services are as indicated in Table 35.

Table 35 Assembly-ISPM#4 Object Instance Attributes and Services

Assembly-ISPM#4 *Class ID = 144, Instance ID = 01 through r*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
--	No additional attributes defined	--
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.3.11 *Assembly-ISPM#5* — The presentation of the Assembly #5 In-Situ Particle Monitor (Assembly-ISPM#5) object attributes and services are as indicated in Table 36.

Table 36 Assembly-ISPM#5 Object Instance Attributes and Services

Assembly-ISPM#5 *Class ID = 145, Instance ID = 01 through r*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
--	No additional attributes defined	--
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.3.12 *Assembly-ISPM#6* — The presentation of the Assembly #6 In-Situ Particle Monitor (Assembly-ISPM#6) object attributes and services are as indicated in Table 37.

Table 37 Assembly-ISPM#6 Object Instance Attributes and Services

Assembly-ISPM#6 *Class ID = 146, Instance ID = 01 through r*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
--	No additional attributes defined	--
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.3.13 *Assembly-ISPM#7* — The presentation of the Assembly #7 In-Situ Particle Monitor (Assembly-ISPM#4) object attributes and services are as indicated in Table 38.

Table 38 Assembly-ISPM#7 Object Instance Attributes and Services

Assembly-ISPM#7 *Class ID = 147, Instance ID = 01 through r*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
--	No additional attributes defined	--
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.3.14 *Assembly-ISPM#8* — The presentation of the Assembly #8 In-Situ Particle Monitor (Assembly-ISPM#8) object attributes and services are as indicated in Table 39.

Table 39 Assembly-ISPM#8 Object Instance Attributes and Services

Assembly-ISPM#8 *Class ID = 148, Instance ID = 01 through r*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
--	No additional attributes defined	--
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.3.15 *Assembly-ISPM#9* — The presentation of the Assembly #9 In-Situ Particle Monitor (Assembly-ISPM#9) object attributes and services are as indicated in Table 40.

Table 40 Assembly-ISPM#9 Object Instance Attributes and Services

Assembly-ISPM#9 *Class ID = 149, Instance ID = 01 through r*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
--	No additional attributes defined	--
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.4 *Specific Device Model for Endpoint Device* — These sections detail the network mapping required to support the Specific Device Model for Endpoint Devices. Table 41 summarizes the Endpoint Device object types. Subsequent Tables 42 to 48 detail the attributes and services associated with each Endpoint Device object type.

Table 41 Endpoint Device Object Types

SDM Object Identifier	*Object Name*	*Modbus Class ID*
EPD1 (DM)	Device Manager	1
EPD2 (SAC)	Sensor Actuator Controller	2
EPD3	Sensor-BI-TH-EP	150
EPD4	Assembly-EPD#1	151
EPD5	Assembly-EPD#2	152
EPD6	Assembly-EPD#3	153
EPD7	Assembly-EPD#4	154

9.4.1 *Device Manager* — The presentation of the Device Manager object attributes and services that are specified in the CDM is specified in Section 7. The presentation of the extended Device Manager object attributes and services that are specified in the SDM for Endpoint Devices is specified in Table 42.

Table 42 DM Object Instance Attributes and Services

Device Manger (DM) *Class ID = 0001, Instance ID = 0001*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
--	No additional attributes defined	--
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.4.2 *Sensor Actuator Controller* — The presentation of the Sensor Actuator Controller object attributes and services that are specified in the CDM is specified in Section 7. The presentation of the extended Sensor Actuator Controller object attributes and services that are specified in the SDM for Endpoint Devices is specified in Table 43.

Table 43 SAC Object Instance Attributes and Services

Sensor Actuator Controller (SAC) *Class ID = 0002, Instance ID = 0001*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
65	Number of Endpoint Objects	SacA65
Services		
ID	*Service Name*	*SDM Tag*
33	Reset Endpoint	S33
34	Download Recipe	S34
35	Upload Recipe	S35
36	Calibrate	S36

9.4.3 *Sensor-BI-TH-EP* — The presentation of the Sensor, Binary Input, Threshold type, Endpoint object attributes and services is specified in Table 44.

Table 44 Sensor-BI-TH-EP Object Instance Attributes and Services

Sensor-BI-TH-EP *Class ID = 0150, Instance ID = 0001 - 1024*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
129	Value	SbithepA16
130	Reading Valid	SbithA64
131	State	SbithA65
132	Status	SbithA66
133	Minimum Time	EpA1
134	Maximum Time	EpA2
135	Target Time	EpA3
136	Elapsed Time	EpA4
137	Time Stamp	EpA5
138	Recipe Identifier	EpA6
139	Step Identifier	EpA7
Services		
ID	*Service Name*	*SDM Tag*
129	Endpoint On	S1
130	Endpoint Off	S2
131	Endpoint Start	S3
132	Endpoint Suspend	S4
133	Endpoint Resume	S5

9.4.4 *Assembly-EPD#1* — The presentation of the Assembly #1 Endpoint Device object attributes and services is specified in Table 45.

Table 45 Assembly-EPD#1 Object Instance Attributes and Services

Assembly-EPD#1 *Class ID = 0151, Instance ID = 0001*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
--	No additional attributes defined	--
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.4.5 *Assembly-EPD#2* — The presentation of the Assembly #2 Endpoint Device object attributes and services is specified in Table 46.

Table 46 Assembly-EPD#2 Object Instance Attributes and Services

Assembly-EPD#2 *Class ID = 0152, Instance ID = 0001*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
--	No additional attributes defined	--
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.4.6 *Assembly-EPD#3* — The presentation of the Assembly #3 Endpoint Device object attributes and services is specified in Table 47.

Table 47 Assembly-EPD#3 Object Instance Attributes and Services

Assembly-EPD#3 *Class ID = 0153, Instance ID = 0001*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
--	No additional attributes defined	--
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.4.7 *Assembly-EPD#4* — The presentation of the Assembly #4 Endpoint Device object attributes and services is specified in Table 48.

Table 48 Assembly-EPD#4 Object Instance Attributes and Services

Assembly-EPD#4 *Class ID = 0154, Instance ID = 0001*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
--	No additional attributes defined	--
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

NOTICE: SEMI makes no warranties or representations as to the suitability of the standard set forth herein for any particular application. The determination of the suitability of the standard is solely the responsibility of the user. Users are cautioned to refer to manufacturer's instructions, product labels, product data sheets, and other relevant literature respecting any materials mentioned herein. These standards are subject to change without notice.

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

SEMI E54.10-0600
SPECIFICATION FOR SENSOR/ACTUATOR NETWORK SPECIFIC DEVICE MODEL FOR AN IN-SITU PARTICLE MONITOR DEVICE

This specification was technically approved by the Global Information & Control Committee and is the direct responsibility of the North American Information & Control Committee. Current edition approved by the North American Regional Standards Committee on March 2, 2000. Initially available at www.semi.org May 2000; to be published June 2000.

1 Purpose

1.1 This specification is part of a suite of standards that specify the implementation of SEMI standards for the Sensor/Actuator Network. The specific purpose of this specification is to describe a network independent application model comprised of device objects that are common to all In-Situ Particle Monitor Devices on a semiconductor equipment Sensor/Actuator communications network.

2 Scope

2.1 An In-Situ Particle Monitor (ISPM) is a device that measures and counts particles. These devices classify by size and count, particles in the environment (gaseous, liquid, or vacuum) utilizing a technique such as detecting light from a sample region of the environment's space. These counts are accumulated in bins and then reported. The number of bins varies by vendor and model.

2.2 This specification specifically addresses the minimum attributes, services, and behavior an In-Situ Particle Monitor (ISPM) device must support to be interoperable on the Sensor/Actuator Network.

2.3 This specification is intended to ensure a high-degree of device interoperability on the Sensor/Actuator Network, while still allowing flexibility for product differentiation and technology evolution.

2.4 The model specified in this specification is used in conjunction with SEMI E54.1 (Standard for Sensor/Actuator Network Common Device Model (CDM)) to completely describe the ISPM as it appears from the network interface.

2.5 This specification, together with SEMI E54, SEMI E54.1, and one of the Sensor/Actuator Network Communication Specifications, form a complete interoperability specification for the ISPM.

2.6 To comply with this specification, a device must implement and support, at a minimum, the required attributes, services, and behavior identified in these documents. Support for optional attributes, services, and behavior is not required to be compliant to this specification. Optional attributes, services, and behavior are specified in these documents to promote further device interoperability as features evolve and are adopted by more manufacturers. If optional attributes, services, and behavior are implemented for this device they must be implemented as identified in this document.

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

3 Limitations

3.1 This specification is a companion to a suite of specifications that together make up the Sensor/Actuator Network Communication standard. Therefore, using portions of this specification that relate to network communications necessarily requires an understanding of the associated network specification.

3.2 As this document is a specification for the In-Situ Particle Monitor Device Model, it does not contain any definition of objects, attributes, services, or behavioral descriptions that are already defined in SEMI E54.1. Additional attributes, attribute assignments, services, and/or service parameters that are ISPM Device specific and/or implementation specific are contained in this specification.

3.3 While this specification is sufficient to completely describe the ISPM as it appears from the network, it does not fully describe behavior of the ISPM which is not visible from the network. This allows flexibility in implementation techniques and product differentiation between manufacturers. Manufacturer specific objects may be defined by the manufacturer but are, by definition, outside the scope of this standard.

3.4 This specification is compatible, but not compliant with SEMI E39. This means that although this specification does not require compliance with SEMI E39, it is extensible such that implementations may be developed that are fully compliant with both standards. Note that the concepts and terminology of this specification are compatible with those of SEMI E39. However, SEMI E39 has specific requirements that are

intended for higher level applications and thus are not applied to the In-Situ Particle Monitor Device Model.

3.5 Operation over the entire range specified for an attribute within a specific object is not a requisite for compliance with this specification.

4 Referenced Standards

4.1 *SEMI Standards*

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

SEMI E54 — Sensor/Actuator Network Standard

SEMI E54.1 — Standard for Sensor/Actuator Network Common Device Model

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 *LCS* — an ISPM sensor named the Laser Current Sensor.

5.1.2 *MNS* — an ISPM sensor named the Median Noise Sensor.

5.1.3 *SLS* — an ISPM sensor named the Stray Light Sensor.

5.2 *Terminology Defined in This Document*

5.2.1 *In-Situ Particle Monitor (ISPM)* — a self-contained device, consisting of a laser that generates light, a light detector, counters, diagnostics and control and signal-processing electronics, commonly used in the semiconductor industry to measure and count particles in a specific area.

5.3 This document inherits the terminology defined by SEMI E54.1 [3].

5.3.1 *Attribute*

5.3.2 *Behavior*

5.3.3 *Boolean*

5.3.4 *Byte*

5.3.5 *Character*

5.3.6 *Common Device Model (CDM)*

5.3.7 *Data Type*

5.3.8 *Data Units*

5.3.9 *Device*

5.3.10 *Device Manager (DM) Object*

5.3.11 *Device Model*

5.3.12 *Double Integer (DINT)*

5.3.13 *Enumerated Byte*

5.3.14 *Full Scale Range*

5.3.15 *Instance*

5.3.16 *Last Valid Value (LVV)*

5.3.17 *Long Integer (LINT)*

5.3.18 *Long Real (LREAL)*

5.3.19 *Manufacturer*

5.3.20 *Nibble*

5.3.21 *Null Character*

5.3.22 *Object*

5.3.23 *Real (REAL)*

5.3.24 *S, A, and C Objects*

5.3.25 *Sensor Actuator Controller (SAC) Object*

5.3.26 *Service*

5.3.27 *Short Integer (SINT)*

5.3.28 *Signed Integer (INT)*

5.3.29 *State Diagram*

5.3.30 *Test String*

5.3.31 *Unsigned Double Integer (UDINT)*

5.3.32 *Unsigned Double Long Integer (UDLINT)*

5.3.33 *Unsigned Integer (UINT)*

5.3.34 *Unsigned Long Integer (ULINT)*

5.3.35 *Unsigned Short Integer (USINT)*

6 Requirements and Specifications

6.1 In order to implement this standard in an In-Situ Particle Monitor Device, it is necessary to also implement SEMI E54.1 and one of the Sensor/Actuator Network Communication Standards [2]. See Section 1.1 for more information on a complete interoperability standard.

6.2 This specification also requires the implementation of a Date_And_Time data structure used to represent the current device Date and Time. Table 1 defines the format of the Date_And_Time data type.

Table 1 Date_And_Time Format

Data Item	*Description*	*Range*
1	Number of milli-seconds since midnight	Unsigned Double Integer (UDINT)
2	Number of days since	Unsigned Integer

Data Item	*Description*	*Range*
	1/1/72	(UINT)

7 Conventions

7.1 This document embraces the conventions and notations stated in Section 6 of SEMI E54.1 [3].

8 Device High Level Structure

8.1 *General Description*

8.1.1 The high-level object view of an In-Situ Particle Monitor (ISPM) Device is shown in Figure 1.

8.1.2 Note that the "ISPM" device object is depicted in Figure 1 only for the purposes of illustrating a high level view of the device and its component objects. In the context of this document, this object is not addressable, does not have addressable attributes, does not have accessible services, and does not exhibit any defined behavior.

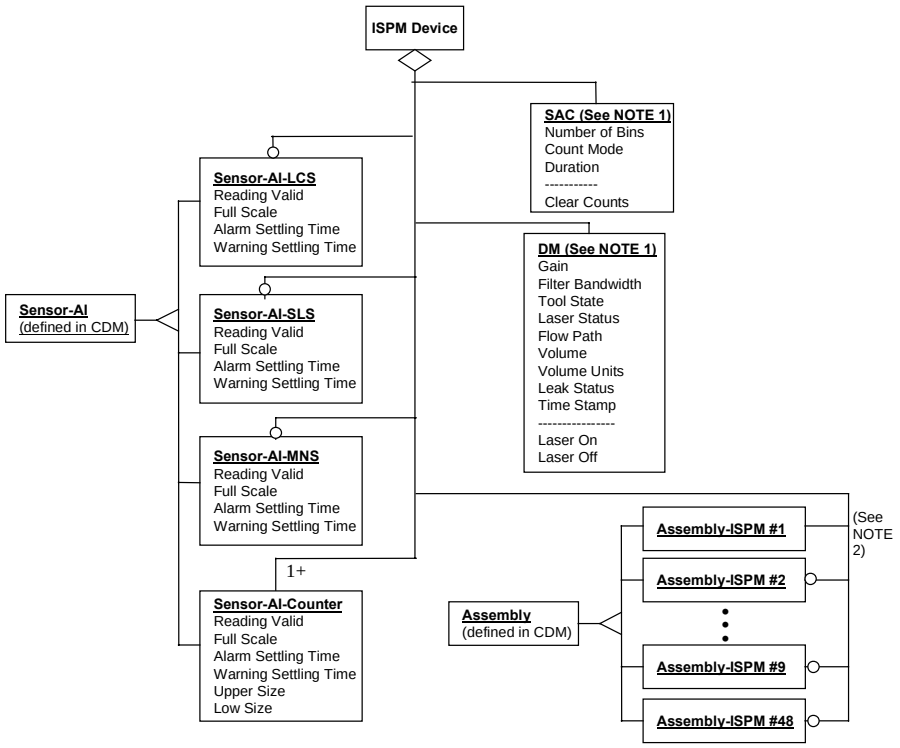
8.1.3 In the remainder of this section, this document defines in detail all of the component objects unique to the ISPM device. References, rather than definitions, are included for the DM, the SAC, and other objects defined in SEMI E54.1 [3].

8.1.4 Many of the objects defined in this document inherit properties from other objects. The properties inherited include attribute, services, and behavior definitions. These other objects are specified here or in SEMI E54.1 [3].

8.1.5 This document provides for future extensions as well as manufacturer specific enhancements by reserving object attribute identifiers and object service identifiers. Specifically, all object definitions in this document specify or reserve the first 64 attribute identifiers (A1 through A64) and the first 64 service identifiers (S1 through S64) allowing manufacturers to specify identifiers beyond these ranges. Additionally, byte enumerated attributes are specified or reserved from 0 to 63 allowing manufacturers to specify an enumeration beyond this range (64 to 255).

8.1.6 *In-Situ Particle Monitor (ISPM) Device Description*

8.1.6.1 An In-Situ particle Monitor device profile is composed of the component objects and object relationships shown in Figure 1.



NOTE 1: The DM and SAC are taken from the CDM. Additional attributes and services are added to support the ISPM device.

NOTE 2: Assembly-ISPM #1 and Assembly-ISPM #6 objects are required. Other Assembly-ISPM objects are optional.

Figure 1
In-Situ Particle Monitor Device High Level Structure

8.1.7 *General Requirements*

8.1.7.1 *Device Objects* — All objects are defined in terms of their object name and Class/Object identifier. Identifiers for all objects described in this document are summarized in Table 2.

Table 2 In-Situ Particle Monitor Device Objects

Referenced Document Section	*Object Name*	*Class/Object Identifier*	*Minimum #*	*Maximum #*
8.2	Device Manager (DM)	ISPMD1	1	1
8.3	Sensor Actuator Controller (SAC)	ISPMD2	1	1
8.5	Sensor-AI-LCS	ISPMD3	0	1
8.6	Sensor-AI-SLS	ISPMD4	0	1
8.7	Sensor-AI-MNS	ISPMD5	0	1
8.8	Sensor-AI-Counter	ISPMD16	1	1024
8.9	Assembly-ISPM#1	ISPMD17	1	1
8.9	Assembly-ISPM#2	ISPMD18	0	1
8.9	Assembly-ISPM#3	ISPMD19	0	1
8.9	Assembly-ISPM#4	ISPMD20	0	1
8.9	Assembly-ISPM#5	ISPMD21	0	1
8.9	Assembly-ISPM#6	ISPMD22	1	1
8.9	Assembly-ISPM#7	ISPMD23	0	1
8.9	Assembly-ISPM#8	ISPMD24	0	1
8.9	Assembly-ISPM#9	ISPMD25	0	1
8.9	Assembly-ISPM#48	ISPMD64	0	1
—	Reserved	ISPM26–ISPM63	—	—
—	Manufacturer Specified	> ISPM64	—	—

8.1.7.2 *Object Services* — Not all object services listed in this document can necessarily be requested over the network. They are included in this document because their behavior may generate network activity.

8.1.7.3 *Object Behavior* — For all service requests received over the network that are not supported by the object, or contain a parameter value which is beyond the supported range, or which is otherwise invalid, a network specific service error response is generated.

8.2 *Device Manager Object (DM)* — The Device Manager object is the device component responsible for managing and consolidating the device operation as specified in SEMI E54.1 [3]. The following sections specify the components of the DM object that are not specified in SEMI E54.1 or require further definition than specified in SEMI E54.1.

8.2.1 *Device Manager Object Attributes* — Required and optional DM object attributes are listed in Table 3.

Table 3 DM Object Attributes

Attribute Name	*Attribute Identifier*	*Access Network*	*Required*	*Form*
Device Type	A1	R	Yes	refer to CDM [1]
Exception Detail Alarm	A13	R	No	refer to CDM [1]
Exception Detail Warning	A14	R	No	refer to CDM [1]
Gain	A33	RW *	No	REAL
Filter Bandwidth	A34	RW *	No	REAL
Tool State	A35	R	No	USINT, enumerated
Laser Status	A36	R	Yes	USINT, enumerated
Flow Path	A37	RW	No	USINT
Volume	A38	R	No	REAL
Volume Units	A39	R	No	Byte, enumerated

Attribute Name	*Attribute Identifier*	*Access Network*	*Required*	*Form*
Leak Status	A40	R	No	Byte, enumerated
Time Stamp	A41	RW	No	Date_And_Time
Reserved	A42–A64	—	—	Reserved for SDM future expansion
Manufacturer Specified	> A64	—	—	Manufacturer Specific attributes

NOTE 1: "*" Indicates that the specific attribute is nonvolatile. Nonvolatile requires that the current attribute value be maintained through a component power cycle.

8.2.1.1 *Device Type* — An attribute that uniquely identifies the type of the device on the network. The device type attribute is assigned as follows:

In-Situ Particle Monitor Device = "ISPM"

8.2.1.2 *Exception Detail Alarm (Optional)* — An attribute which identifies the detailed alarm status of the device. Table 4 defines the bit assignments associated with the alarm exception detail.

Table 4 Exception Detail Alarm Bit Assignments

Bit	*Device Specific Alarm[0]*
0	Reserved
1	Interlock
2	Sensor Not Detected
3	Leak Detected
4	Reserved
5	Sensor Type Changed
6	Reserved
7	Reserved

Bit	*Device Specific Alarm[1]*
0	High Laser Current
1	High Stray Light
2	High Median Noise
3	High Calibration
4	High Fill Rate
5	High Flow Rate
6	Reserved
7	Reserved

Bit	*Device Specific Alarm[2]*
0	Low Laser Current
1	Low Stray Light
2	Low Median Noise
3	Low Calibration
4	Low Fill Rate
5	Low Flow Rate
6	Reserved
7	Reserved

8.2.1.3 *Exception Detail Warning (Optional)* — An attribute which identifies the detailed warning status of the device. Table 5 defines the bit assignments associated with the warning exception detail.

Table 5 Exception Detail Warning Bit Assignments

Bit	*Device Specific Warning[0]*
0	Reserved
1	0
2	0
3	0
4	Reserved
5	0
6	Reserved
7	Reserved

Bit	*Device Specific Warning[1]*
0	High Laser Current
1	High Stray Light
2	High Median Noise
3	High Calibration
4	High Fill Rate
5	High Flow Rate
6	Reserved
7	Reserved

Bit	*Device Specific Warning[2]*
0	Low Laser Current
1	Low Stray Light
2	Low Median Noise
3	Low Calibration
4	Low Fill Rate
5	Low Flow Rate
6	Reserved
7	Reserved

8.2.1.4 *Manufacturer Exception Detail Size (Optional)* — An attribute that specifies the number of exception detail bytes included in the alarm or warning details.

8.2.1.5 *Gain (Optional)* — An attribute that specifies the Gain value of the amplifier for the photodiode signal. The factory configured out-of-box value is determined by the manufacturer based on the sensor type.

8.2.1.6 *Filter Bandwidth (Optional)* — An attribute that determines the bandwidth in kilohertz of the amplifier for the photodiode signal. The factory configured out-of-box value is determined by the manufacturer based on the sensor type.

8.2.1.7 *Tool State (Optional)* — An attribute that identifies the current state of the tool that is utilizing the counter. In many cases, the ISPM device can detect and report the Tool State; this variable is utilized for this type of reporting. This attribute is enumerated and is specified by the manufacturer.

8.2.1.8 *Laser Status* — An attribute that records the current status of the laser. This attribute is enumerated. The possible enumeration and the requirement for support are as follows:

0 = LASER OFF (required)
1 = LASER TURNING ON (optional)
2 = LASER ON BUT NOT STABLE (optional)
3 = LASER ON AND STABLE (required)
4 = LASER INTERLOCKED (optional)
5 = SENSOR MISSING (required)
6–63 = Reserved
64–255 = Manufacturer Specified (optional)

8.2.1.9 *Flow Path (Optional)* — An attribute that identifies the current flow path (from multiple sources) being utilized. The source may be modified by a Host at anytime. Changing the "Flow path" attribute connects the ISPM device to a different source. When changing sources, data is not valid for a time interval whose length is a function of the distance to the source.

8.2.1.10 *Volume (Optional)* — An attribute that maintains the sample volume. This attribute is used primarily by liquid particle counters utilizing a sampling technique in a small container to extrapolate the particle count in the volume of interest. The value of this attribute is the size of the small container.

8.2.1.11 *Volume Units (Optional)* — An attribute that specifies the unit for the "Volume" attribute. This attribute is enumerated and can take on one of the following values: milliliters or gallons. (See Appendix 1 of SEMI E54.1 for assigned values.)

8.2.1.12 *Leak Status (Optional)* — An attribute that indicates which one of the potential leak locations is actually leaking. This attribute is enumerated and is specified by the manufacturer. The enumerated value corresponds to a specific leak location.

8.2.1.13 *Time Stamp (Optional)* — An attribute that records the time when the last counting event completed.

8.2.1.14 *Duration (Optional)* — An attribute that defines the interval of time during which the particle counts are collected and put into the bin assemblies.

8.2.1.15 *Initial and Default Values*

Table 6 DM Object Attribute Initial and Default Values

Attribute	*Initial Value*	*Default Value*	*Comment*
Device Type	ISPM	ISPM	ISPM = In-Situ Particle Monitor Device
Exception Detail Alarm	0	0	
Exception Detail Warning	0	0	
Laser Status	0	0	Laser Off

8.2.2 *Device Manager Object Services* — The services provided by the Device Manager object are defined in SEMI E54.1 [1]. The Device Manager object supports the additional services listed below.

Table 7 Device Manager Object Services

Service	*Service Identifier*	*Type*	*Description*
Laser On	S1	R	Used to prompt the device to take the laser from the LASER OFF state to LASER ON AND STABLE state as defined by the Laser Status attribute.
Laser Off	S2	R	Used to prompt the device to take the laser immediately to the LASER OFF state as defined by the Laser Status attribute.
Reserved	S3–S64	—	Reserved for future expansion
Manufacturer Specified	> S64	—	Manufacturer Specific services

8.2.2.1 *Laser On (Optional)* — This service is used to prompt the device to take the laser from the LASER OFF state through supported laser states 1 and/or 2, to state 3, as defined by the Laser Status attribute. If the laser turns on successfully, a "success" response code is returned. If the sensor is missing or interlocked off, a "object state conflict" error response is returned and the laser remains in the appropriate state.

8.2.2.2 *Laser Off (Optional)* — This service is used to prompt the device to take the laser to the LASER OFF state immediately. If the laser turns off successfully, a "success" response is returned. If the sensor is in an interlocked state, a "object state conflict" error response is returned and the laser remains in the interlocked off state.

8.2.3 *Device Manager Object Behavior* — The behavior exhibited by the Device Manager object is defined in SEMI E54.1 [1]. Additional behavior is detailed below.

8.2.3.1 *Device Manager EXECUTING State Behavior* — Required sub-states within the EXECUTING state, descriptions of these substates, and a transition matrix associated with these sub-states are given in Figure 2, Table 8, and Table 9 respectively.

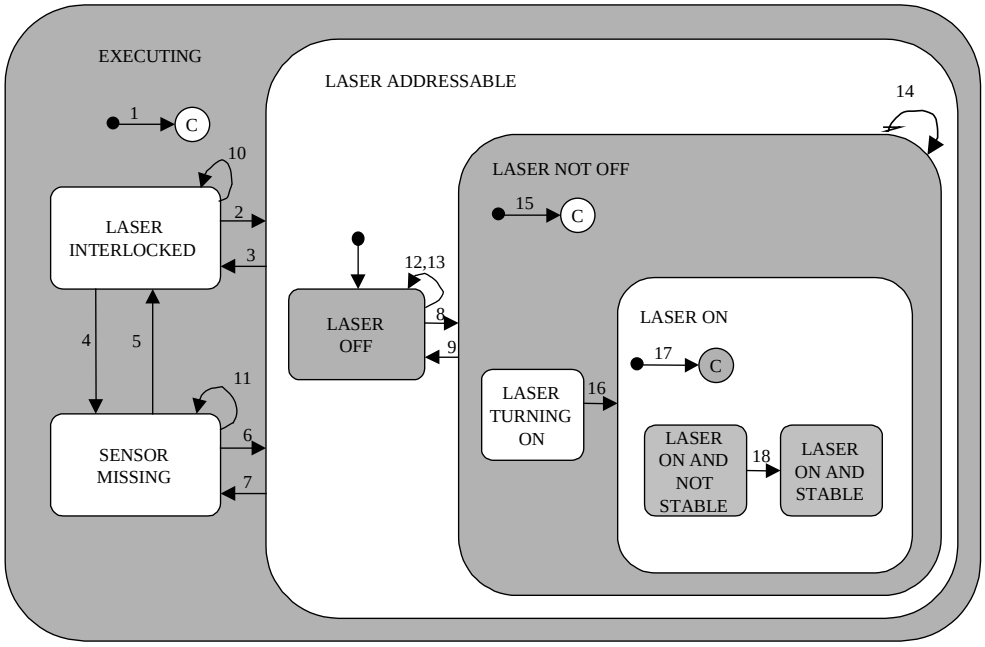


Figure 2
Device Manager Object Behavior Within the EXECUTING State

Table 8 Device Manager Behavior EXECUTING Sub-State Description

State	*Description*
EXECUTING	Laser is in one of the following enumerated states as indicated in the Laser Status Device Manager attribute: LASER INTERLOCKED (optional), LASER MISSING (optional), or LASER OFF. Device will respond to Laser On and Laser Off services as appropriate to move between sub-states within the EXECUTING and LASER ADDRESSABLE states.
LASER ADDRESSABLE	Laser is in one of the following enumerated states as indicated in the Laser Status Device Manager attribute: LASER OFF, LASER ON AND STABLE, LASER TURNING ON (optional), or LASER ON BUT NOT STABLE (optional). Device will respond to Laser On and Laser Off services as appropriate to move between sub-states within the LASER ADDRESSABLE state.
LASER OFF	This is the entry sub-state to LASER ADDRESSABLE; Laser is Off; Laser Status Device Manager attribute is in the enumerated state: LASER OFF. Device is not performing the "counting" process. (See NOTE 1.)
LASER ON AND STABLE	Sub-state of LASER ADDRESSABLE; device is "counting" (See NOTE 1); Laser Status Device Manager attribute is the enumerated state: LASER ON AND STABLE. Laser is performing the "counting" process.
LASER INTERLOCKED	Laser is Interlocked off; Laser Status Device Manager attribute is in the enumerated state: LASER INTERLOCKED. Device is not performing the "counting" process. (See NOTE 1.) Sensor may or may not be missing. Device cannot move to the LASER ADDRESSABLE or SENSOR MISSING states until the interlock is removed.
SENSOR MISSING	Laser sensor is missing; Laser Status Device Manager attribute is in the enumerated state: SENSOR MISSING. Device is not performing the "counting" process. (See NOTE 1.) Device cannot move to the LASER ADDRESSABLE state until the sensor is replaced.
LASER NOT OFF	Laser is in one of the enumerated states as indicated in the Laser Status Device Manager attribute: LASER TURNING ON (optional), LASER ON AND NOT STABLE (optional), or LASER ON AND STABLE. Device will respond to Laser Off service as appropriate to move between sub-states within the LASER ADDRESSABLE state.
LASER TURNING ON	An optional sub-state of LASER NOT OFF. This is a conditional entry sub-state to LASER NOT OFF. Device is preparing to turn on. Device is not performing the "counting" process. (See NOTE 1.)
LASER ON	Laser is in one of the enumerated states as indicated in the Laser Status Device Manager attribute: LASER ON AND NOT STABLE (optional) or LASER ON AND STABLE. Device will respond to Laser Off service as appropriate to move between sub-states within the LASER ADDRESSABLE state.
LASER ON AND NOT STABLE	Sub-state of LASER ON; device is not "counting" (See NOTE 1); Laser Status Device Manager attribute is the enumerated state: LASER ON AND NOT STABLE. Laser is not performing the "counting" process.

NOTE 1: The "counting" process is defined in Section 8.8.3.1.

Table 9 Device Manager Behavior EXECUTING Sub-State Transition Matrix (See NOTE 1.)

#	*Current State*	*Trigger*	*New State*	*Action*	*Comment*
1	Entry into EXECUTING	Device detects interlock condition, and availability and state of laser.	Conditional: LASER ADDRESSABLE/ LASER OFF or LASER INTERLOCKED or SENSOR MISSING	If the sensor is in an interlocked or missing condition then state is LASER INTERLOCKED. If laser interlock and or laser missing is not supported then state is LASER OFF. Set Laser Status attribute to appropriate value.	Entry state depends on availability of laser and physical laser interlock setting on device.
2	LASER INTERLOCKED	Device detects removal of interlock condition	LASER MISSING	Set Laser Status attribute to	Device moves to LASER OFF state.

#	Current State	Trigger	New State	Action	Comment
		and determines that sensor is not missing.		appropriate value.	
3	LASER ADDRESSABLE	Device detects that it has been set to a laser interlock condition.	LASER INTERLOCKED	Set Laser Status attribute to appropriate value.	Valid for all sub-states of LASER ADDRESSABLE.
4	LASER INTERLOCKED	Device detects removal of interlock condition and determines that sensor is missing.	SENSOR MISSING	Set Laser Status attribute to appropriate value.	
5	SENSOR MISSING	Device detects that it has been set to a laser interlock condition.	LASER INTERLOCKED	Set Laser Status attribute to appropriate value.	
6	SENSOR MISSING	Device detects replacement of sensor.	LASER ADDRESSABLE/ LASER OFF	Set Laser Status attribute to appropriate value.	Device moves to LASER OFF.
7	LASER ADDRESSABLE/ LASER OFF	Device detects that sensor is missing.	SENSOR MISSING	Set Laser Status attribute to appropriate value.	
8	LASER OFF	Laser On request	LASER ON AND STABLE	Take the object from the LASER OFF state through optional laser states LASER TURNING ON and LASER ON BUT NOT STABLE to the required LASER ON AND STABLE state, turning on the laser and stabilizing it respectively. Set the Laser Status attribute to the appropriate values throughout the transition. Issue Laser On response and begin the "counting" process. (See NOTE 2.)	LASER TURNING ON and "LASER ON BUT NOT STABLE" are optional intermediate states between LASER OFF and LASER ON AND STABLE. Laser must not be missing or in an interlocked off state. Service response is not issued until transition to LASER ON AND STABLE is completed.
9	LASER ON AND STABLE	Laser Off request	LASER OFF	Stop the "counting" process. (See NOTE 2.) Turn the laser off. Issue Laser Off response.	
10	LASER INTERLOCKED	Laser Off request or Laser On request	LASER INTERLOCKED	Error response	Object cannot move from this state until interlock is turned off.
11	SENSOR MISSING	Laser Off request or Laser On request	SENSOR MISSING	Error response	Object cannot move to LASER ADDRESSABLE state until sensor is replaced.
12	LASER OFF	Laser Off request	LASER OFF	Error response	Laser is already off.
13	LASER OFF	Laser On request	LASER OFF	Device attempts to take the laser from the "Laser Off" state through optional laser	Behavior associated with determining that laser will not turn on properly or

#	Current State	Trigger	New State	Action	Comment
				state “Laser Turning On” and to required state “Laser On But Not Stable”. Set the Laser Status attribute to the appropriate values throughout the transition attempt. Laser either won’t turn on properly or won’t stabilize. Turn laser back off and generate Error response.	won’t stabilize is manufacturer specific.
14	LASER ON		LASER ON AND STABLE	Error response	Laser is attempting to turn on.
15	Entry into LASER NOT OFF	Laser On request	Conditional: LASER NOT OFF/LASER TURNING ON or LASER ON/LASER ON AND NOT STABLE or LASER ON AND STABLE	Set Laser Status attribute to appropriate value.	Entry state depends on device support for optional states LASER TURNING ON and LASER ON / LASER ON AND NOT STABLE or LASER ON AND STABLE
16	LASER TURNING ON	Device detects that the laser is on.	LASER ON/LASER ON AND NOT STABLE or LASER ON AND STABLE	Set Laser Status attribute to appropriate value.	Behavior associated with determining that laser is in the process of turning on and is manufacturer specific.
17	Entry into LASER ON	Laser On request	Conditional: LASER ON/LASER ON AND NOT STABLE or LASER ON AND STABLE	Set Laser Status attribute to appropriate value.	Entry state depends on device support for optional states LASER ON AND NOT STABLE or LASER ON AND STABLE.
18	LASER ON AND NOT STABLE	Device determines that the laser is stable.	LASER ON AND STABLE	Set Laser Status attribute to appropriate value.	Laser is on. Behavior associated with determining that laser will turn on properly or won’t stabilize is manufacturer specific.

NOTE 1: Note that this matrix augments the Device Manager Behavior State Transition Matrix as defined in SEMI E54.1 [3]. All transitions are in addition to those specified in SEMI E54.1.

NOTE 2: The “counting” process is defined in Section 8.8.3.1.

8.3 *Sensor Actuator Controller Object (SAC)* — The Sensor Actuator Controller object is the device component responsible for coordinating the interaction of the ISPM device with the sensory/actuation/control environment as specified in SEMI E54.1 [3]. The following sections specify the components of the SAC object that are not specified in the Common Device model or require further definition than specified in the CDM.

8.3.1 *Sensor Actuator Controller Object Attributes* — The attributes provided by the Sensor Actuator Controller object are defined in SEMI E54.1 [3]. Table 10 contains the additional attributes required for the Sensor Actuator Controller object.

Table 10 SAC Object Attributes

Attribute Name	*Attribute Identifier*	*Access Network*	*Required*	*Form*
Number of Bins	SacA65	R*	Yes	UINT
Count Mode	SacA66	RW*	Yes	Enumerated, USINT
Duration	SacA67	R	No	REAL

NOTE 1: "*" Indicates that the specific attribute is nonvolatile. Nonvolatile requires that the current attribute value be maintained through a component power cycle.

8.3.1.1 *Number of Bins* — An attribute that specifies the number of counters in the device. This value represents the number of Sensor-AI-Counter objects in the device.

8.3.1.2 *Count Mode* — An attribute that specifies whether all count object values are cleared (to zero) when read. This attribute is an enumerated USINT that can take on one of the following values:

0 = The particle counter count is not affected by reading the count.
1 = The particle counter count is cleared to zero when the count ("value" attribute of the Counter object) is read or reported through a service request to or from an Assembly-ISPM object that contains that count.
2 = The particle counter count is cleared to zero when the count ("value" attribute of the Counter object) is read or reported (regardless of the connection) through a service request to or from either an Assembly-ISPM object that contains that count ("value" attribute), or the Counter object that contains the count.
3–63 = Reserved
64–255 = Manufacturer Specified

8.3.1.3 *Duration (Optional)* — An attribute that defines the interval of time during which the particle counts (value of *Value* object attribute) are collected and put into the bin assemblies.

8.3.1.4 *Initial and Default Values*

Table 11 SAC Object Attributes Initial and Default Values

Attribute	*Initial Value*	*Default Value*	*Comment*
Number of Bins	Manufacturer Specified	Manufacturer Specified	
Count Mode	LVV	0	Do not clear particle counter value.
Duration	Manufacturer Specified	Manufacturer Specified	

8.3.2 *Sensor Actuator Controller Object Services* — The services provided by the Sensor Actuator Controller object are defined in SEMI E54.1 [1]. Table 12 contains the additional services required for the Sensor Actuator Controller object.

Table 12 SAC Object Services

Service	*Service Identifier*	*Type*	*Description*
Clear Counts	S33	R	Used to clear the "Value" attribute of all Sensor-AI-Counter sensor objects.
Reserved	S34–S64	—	Reserved for future expansion
Manufacturer Specified	> S64	—	Manufacturer Specific services

8.3.2.1 *Clear Counts (Optional)* — This service is used to instruct all of the Sensor-AI-Counter object to perform a one-time clear of their respective Sensor-AI-Counter "*Value*" attributes by setting the value to zero. There are no parameters required for this service.

8.3.3 *Sensor Actuator Controller Object Behavior*

8.3.3.1 The behavior exhibited by the Sensor Actuator Controller object is defined in SEMI E54.1 [3]. Additional behavior is detailed below.

8.3.3.2 The "*Value*" attribute is cleared (to zero) in each ISPM counter sensor object, after being read, if the "*Count Mode"* attribute has a value of 1. It may also be cleared by a "Clear Counts" service of the SAC object or the "Reset" service issued to the S-Analog Sensor class.

8.4 *Sensor-AI Object* — The Sensor-AI object is the device component responsible for coordinating the behavior common to all analog input sensor elements in the ISPM device as specified in SEMI E54.1 [3].

8.4.1 *Sensor-AI Object Attributes* — The attributes provided by the Sensor-AI object are defined in SEMI E54.1 [1]. There are no additional attributes required for the Sensor-AI object.

8.4.2 *Sensor-AI Object Services* — The services provided by the Sensor-AI object are defined in SEMI E54.1 [3]. There are no additional services required for the Sensor-AI object.

8.4.3 *Sensor-AI Object Behavior* — The behavior exhibited by the Sensor-AI object is defined in SEMI E54.1 [3]. There is no additional behavior required for the Sensor-AI object.

8.5 *Sensor-AI-LCS Object* — The Sensor-AI-LCS (Laser Current Sensor) object inherits the attributes, services, and behavior of the Sensor-AI as defined in SEMI E54.1 [3]. The Sensor-AI-LCS is the device component responsible for retrieving a reading from a physical laser current sensor, optionally correcting the reading with a manufacturer specified algorithm, or algorithms, then making the value available through the "*Value*" attribute.

8.5.1 *Sensor-AI-LCS Object Attributes* — The attributes provided by the Sensor-AI object are defined in SEMI E54.1 [3]. The Sensor-AI object attribute content and its attribute extensions to support the Sensor-AI-LCS object are listed in Table 13.

Table 13 Sensor-AI and Sensor-AI-LCS Object Attributes

Attribute Name	*Attribute Identifier*	*Access Network*	*Required*	*Form*
Value	SaiA16	R	Yes	UINT
Gain	SaiA65	R	Yes	REAL
Data Type	SaiA66	R	Yes	USINT
Data Units	SaiA67	R	Yes	Enumerated, UINT milliAmps only value supported
Reading Valid	LcsA1	R	Yes	Enumerated, Byte
Full Scale	LcsA2	R	Yes	UINT
Alarm Settling Time	LcsA3	R	Yes	UINT
Warning Settling Time	LcsA4	R	Yes	UINT
Reserved	LcsA5–LcsA64	—	—	Reserved for future expansion.
Manufacturer Specified	> LcsA64	—	—	Manufacturer Specific attributes

8.5.1.1 *Value* — The attribute that maintains the laser current sensor value. This value is always read as milliAmps.

8.5.1.2 *Gain* — The attribute that specifies the gain of the laser current measuring circuit.

8.5.1.3 *Data Type* — The attribute which defines the format of the data associated with the "value" attribute.

8.5.1.4 *Data Units* — The attribute which specifies the units for the "*Value*" attribute. This attribute is an enumerated UINT that can take only one value:

MilliAmps (as assigned in Appendix 1 of SEMI E54.1).

8.5.1.5 *Reading Valid* — An attribute which specifies whether the "*Value*" attribute contains a valid value. This attribute is an enumerated byte that can take on one of the following values:

0 = Invalid, or
1 = Valid.

8.5.1.6 *Full Scale* — An attribute which specifies the maximum value allowed for the "*Value*" attribute.

8.5.1.7 *Alarm Settling Time* — An attribute which specifies the maximum time in milliseconds that the "*Value*" attribute may take to stabilize before an alarm is generated.

8.5.1.8 *Warning Settling Time* — An attribute which specifies the maximum time in milliseconds that the "*Value*" attribute may take to stabilize before a warning is generated.

8.5.1.9 *Initial and Default Values*

Table 14 Sensor-AI and Sensor-AI-LCS Object Attributes Initial and Default Values

Attribute	*Initial Value*	*Default Value*	*Comment*
Value	0	0	
Gain	LVV	1.0	
Data Type	LVV	UINT	
Data Units	LVV	MilliAmps	See Appendix 1 of SEMI E54.1.
Reading Valid	0	0	Invalid
Full Scale	LVV	0XFFFF	
Alarm Settling Time	LVV	1000	Milliseconds
Warning Settling Time	LVV	1000	Milliseconds

8.5.2 *Sensor-AI-LCS Object Services* — The services provided by the Sensor-AI-LCS are inherited from the Sensor-AI object defined in SEMI E54.1 [3]. There are no additional services required for the Sensor AI-LCS object.

8.5.3 *Sensor-AI-LCS Object Behavior* — The behavior exhibited by the Sensor-AI-LCS object is inherited from the Sensor-AI object defined in SEMI E54.1 [3]. Additional specific behavior associated with the Sensor-AI-LCS object is defined below.

8.5.3.1 *Sensor-AI-LCS OPERATING Application Process* — A reading is retrieved from a physical laser current sensor. This reading may be corrected with a manufacturer-specified algorithm. This corrected reading becomes the input to the gain formula to generate the *value* attribute as referenced in SEMI E54.1 [3]. This process is executed only when the Device Manager Object is in the EXECUTING state as defined in the Common Device Model. When not in the EXECUTING state, the value of the "*Reading Valid*" attribute shall be set to "Invalid". When in one of the sub-states of the EXECUTING state, the validity of the "*Value*" attribute shall be manufacturer specific.

8.6 *Sensor-AI-SLS Object* — The Sensor-AI-SLS (Stray Light Sensor) object inherits the attributes, services, and behavior of the Sensor-AI as defined in SEMI E54.1 [3]. The Sensor-AI-SLS is the device component responsible for retrieving a reading from a physical stray light sensor, optionally correcting the reading with a manufacturer specified algorithm, or algorithms, then making the value available through the "*Value*" attribute.

8.6.1 *Sensor-AI-SLS Object Attributes* — The attributes provided by the Sensor-AI object are defined in SEMI E54.1 [3]. The Sensor-AI object attribute content and its attribute extensions to support the Sensor-AI-SLS object is listed in Table 15.

Table 15 Sensor-AI and Sensor-AI-SLS Object Attributes

Attribute Name	*Attribute Identifier*	*Access Network*	*Required*	*Form*
Value	SaiA16	R	Yes	UINT
Gain	SaiA65	R	Yes	REAL
Data Type	SaiA66	R	Yes	USINT
Data Units	SaiA67	R	Yes	Enumerated, UINT milliVolts only value supported
Reading Valid	SlsA1	R	Yes	Enumerated, Byte

Attribute Name	*Attribute Identifier*	*Access Network*	*Required*	*Form*
Full Scale	SlsA2	R	Yes	UINT
Alarm Settling Time	SlsA3	R	Yes	UINT
Warning Settling Time	SlsA4	R	Yes	UINT
Reserved	SlsA5–SlsA64	—	—	Reserved for future expansion.
Manufacturer Specified	> SlsA64	—	—	Manufacturer Specific attributes

8.6.1.1 *Value* — The attribute that maintains the stray light sensor value. This value is always read as milliVolts.

8.6.1.2 *Gain* — The attribute that specifies the gain between the photodiode (or some other light detection source) and the stray light reading.

8.6.1.3 *Data Type* — The attribute which defines the format of the data associated with the "value" attribute.

8.6.1.4 *Data Units* — The attribute which specifies the units for the "*Value*" attribute. This attribute is an enumerated UINT and can take only one value:

MilliVolts (as assigned in Appendix 1 of SEMI E54.1).

8.6.1.5 *Reading Valid* — An attribute which specifies whether the "*Value*" attribute contains a valid value. This attribute is an enumerated byte that can take on one of the following values:

0 = Invalid, or
1 = Valid.

8.6.1.6 *Full Scale* — An attribute which specifies the maximum value allowed for the "*Value*" attribute.

8.6.1.7 *Alarm Settling Time* — An attribute which specifies the maximum time in milliseconds that the "*Value*" attribute may take to stabilize before an alarm is generated.

8.6.1.8 *Warning Settling Time* — An attribute which specifies the maximum time in milliseconds that the "*Value*" attribute may take to stabilize before a warning is generated.

8.6.1.9 *Initial and Default Values*

Table 16 Sensor-AI and Sensor-AI-SLS Object Attributes Initial and Default Values

Attribute	*Initial Value*	*Default Value*	*Comment*
Value	0	0	
Gain	LVV	1.0	
Data Type	LVV	UINT	
Data Units	LVV	MilliVolts	See Appendix 1 of SEMI E54.1.
Reading Valid	0	0	Invalid
Full Scale	LVV	0XFFFF	
Alarm Settling Time	LVV	1000	Milliseconds
Warning Settling Time	LVV	1000	Milliseconds

8.6.2 *Sensor-AI-SLS Object Services* — The services provided by the Sensor-AI-SLS object are inherited from the Sensor-AI object defined in SEMI E54.1 [3]. There are no additional services required for the Sensor AI-SLS object.

8.6.3 *Sensor-AI-SLS Object Behavior* — The behavior exhibited by the Sensor-AI-SLS object is inherited from the Sensor-AI object defined in SEMI E54.1 [3]. Additional specific behavior associated with the Sensor-AI-SLS object is defined below.

8.6.3.1 *Sensor-AI-SLS OPERATING Application Process* — A reading is retrieved from a physical stray light sensor. This reading may be corrected with a manufacturer-specified algorithm. This corrected reading becomes the input to the gain formula to generate the *value* attribute as referenced in SEMI E54.1 [3]. This process is executed only when the Device Manager Object is in the EXECUTING state as defined in the Common Device Model. When

not in the EXECUTING state, the value of the "*Reading Valid*" attribute shall be set to "Invalid". When in one of the sub-states of the EXECUTING state, the validity of the "*Value*" attribute shall be manufacturer specific.

8.7 *Sensor-AI-MNS Object* — The Sensor-AI-MNS (Median Noise Sensor) object inherits the attributes, services, and behavior of the Sensor-AI as defined in SEMI E54.1 [3]. The Sensor-AI-MNS is the device component responsible for retrieving a reading from a physical median noise sensor, optionally correcting the reading with a manufacturer specified algorithm, or algorithms, then making the value available through the "*Value*" attribute.

8.7.1 *Sensor-AI-MNS Object Attributes* — The attributes provided by the Sensor-AI object are defined in SEMI E54.1 [3]. The Sensor-AI object attribute content and its attribute extensions to support the Sensor-AI-MNS object is listed in Table 17.

Table 17 Sensor-AI and Sensor-AI-MNS Object Attributes

Attribute Name	*Attribute Identifier*	*Access Network*	*Required*	*Form*
Value	SaiA16	R	Yes	UINT
Gain	SaiA65	R	Yes	REAL
Data Type	SaiA66	R	Yes	USINT
Data Units	SaiA67	R	Yes	Enumerated, UINT milliVolts only value supported
Reading Valid	MnsA1	R	Yes	Enumerated, Byte
Full Scale	MnsA2	R	Yes	UINT
Alarm Settling Time	MnsA3	R	Yes	UINT
Warning Settling Time	MnsA4	R	Yes	UINT
Reserved	MnsA5–MnsA64	—	—	Reserved for future expansion.
Manufacturer Specified	> MnsA64	—	—	Manufacturer Specific attributes

8.7.1.1 *Value* — The attribute that maintains the median noise sensor value. This is the noise level at the output of the sensor amplifier. The value is calculated by taking the median value of the last manufacturer specific number of rolling samples of the sensor amplifier output. This value is always read as milliVolts.

8.7.1.2 *Gain* — The attribute that specifies the gain of the median noise measuring circuit.

8.7.1.3 *Data Type* — The attribute which defines the format of the data associated with the "value" attribute.

8.7.1.4 *Data Units* — The attribute which specifies the units for the "*Value*" attribute. This attribute is an enumerated UINT and can take only one value:

MilliVolts (as assigned in Appendix 1 of SEMI E54.1).

8.7.1.5 *Reading Valid* — An attribute which specifies whether the "*Value*" attribute contains a valid value. This attribute is an enumerated byte that can take on one of the following values:

0 = Invalid, or
1 = Valid.

8.7.1.6 *Full Scale* — An attribute which specifies the maximum value allowed for the "*Value*" attribute.

8.7.1.7 *Alarm Settling Time* — An attribute which specifies the maximum time in milliseconds that the "*Value*" attribute may take to stabilize before an alarm is generated.

8.7.1.8 *Warning Settling Time* — An attribute which specifies the maximum time in milliseconds that the "*Value*" attribute may take to stabilize before a warning is generated.

8.7.1.9 *Initial and Default Values*

Table 18 Sensor-AI and Sensor-AI-MNS Object Attributes Initial and Default Values

Attribute	*Initial Value*	*Default Value*	*Comment*
Value	0	0	
Gain	LVV	1.0	
Data Type	LVV	UINT	
Data Units	LVV	MilliVolts	See Appendix 1 of SEMI E54.1.
Reading Valid	0	0	Invalid
Full Scale	LVV	0XFFFF	
Alarm Settling Time	LVV	1000	Milliseconds
Warning Settling Time	LVV	1000	Milliseconds

8.7.2 *Sensor-AI-MNS Object Services* — The services provided by the Sensor-AI-MNS object are inherited from the Sensor-AI object defined in SEMI E54.1 [3]. There are no additional services required for the Sensor AI-MNS object.

8.7.3 *Sensor-AI-MNS Object Behavior* — The behavior exhibited by the Sensor-AI-MNS object is inherited from the Sensor-AI object defined in SEMI E54.1 [3]. Additional specific behavior associated with the Sensor-AI-MNS object is defined below.

8.7.3.1 *Sensor-AI-MNS OPERATING Application Process* — A reading is retrieved from the physical median noise sensor. This reading is maintained in a rolling count of manufacturer specified samples and then the median value of the manufacturer specified samples is calculated. This median reading becomes the input to the gain formula to generate the *value* attribute as referenced in SEMI E54.1 [3]. This process is executed only when the Device Manager Object is in the EXECUTING state as defined in SEMI E54.1. When not in the EXECUTING state, the value of the "*Reading Valid*" attribute shall be set to "Invalid". When in one of the sub-states of the EXECUTING state, the validity of the "*Value*" attribute shall be manufacturer specific.

8.8 *Sensor-AI-Counter Object* — The Sensor-AI-Counter object inherits the attributes, services, and behavior of the Sensor-AI as defined in SEMI E54.1 [3]. The Sensor-AI-Counter is the device component responsible for maintaining a count of the number of particles deposited in each bin and then making the value available through the "*Value*" attribute.

8.8.1 *Sensor-AI-Counter Object Attributes* — The attributes provided by the Sensor-AI object are defined in SEMI E54.1 [3]. The Sensor-AI object attribute content and its attribute extensions to support the Sensor-AI-Counter object is listed in Table 19.

Table 19 Sensor-AI and Sensor-AI-Counter Object Attributes

Attribute Name	*Attribute Identifier*	*Access Network*	*Required*	*Form*
Value	SaiA16	R	Yes	UDINT
Data Type	SaiA66	R	Yes	USINT
Data Units	SaiA67	R	Yes	Enumerated, UINT
Reading Valid	CounterA1	R	Yes	Enumerated, Byte
Full Scale	CounterA2	R	Yes	UDINT
Alarm Settling Time	CounterA3	R	Yes	UINT
Warning Settling Time	CounterA4	R	Yes	UINT
Upper Size	CounterA5	RW (See NOTE 1.)	Yes	REAL
Lower Size	CounterA6	RW (See NOTE 1.)	Yes	REAL
Reserved	CounterA7–CounterA64	—	—	Reserved for future expansion.
Manufacturer Specified	> CounterA64	—	—	Manufacturer Specific attributes

NOTE 1: Consult the manufacturer's specification for specific "Write" capabilities of this attribute.

8.8.1.1 *Value* — The attribute that maintains the particle count in the bin.

8.8.1.2 *Data Type* — The attribute that defines the format of the data associated with the "*Value*" attribute.

8.8.1.3 *Data Units* — The attribute which specifies the units for the "*Value*" attribute. This attribute is an enumerated UINT that can take on one of the following values: raw counts, count per second, counts per milliliter, and counts per gallon. (See Appendix 1 of SEMI E54.1.)

8.8.1.4 *Reading Valid* — An attribute which specifies whether the "*Value*" attribute contains a valid value. This attribute is an enumerated byte that can take on one of the following values:

0 = Invalid, or
1 = Valid.

8.8.1.5 *Full Scale* — An attribute which specifies the maximum value allowed for the "*Value*" attribute.

8.8.1.6 *Alarm Settling Time* — An attribute which specifies the maximum time in milliseconds that the "*Value*" attribute may take to stabilize before an alarm is generated.

8.8.1.7 *Warning Settling Time* — An attribute which specifies the maximum time in milliseconds that the "*Value*" attribute may take to stabilize before a warning is generated.

8.8.1.8 *Upper Size* — An attribute which specifies the upper bound of the particle size, in microns, included in the bin counts.

8.8.1.9 *Lower Size* — An attribute which specifies the lower bound of particle size, in microns, included in the bin counts.

8.8.1.10 *Initial and Default Values*

Table 20 Sensor-AI and Sensor-AI-Counter Object Attributes Initial and Default Values

Attribute	*Initial Value*	*Default Value*	*Comment*
Value	0	0	
Data Type	LVV	UDINT	
Data Units	LVV	0	Raw counts
Reading Valid	0	0	Invalid
Full Scale	LVV	0XFFFFFFFF	
Alarm Settling Time	LVV	1000	Milliseconds
Warning Settling Time	LVV	1000	Milliseconds
Upper Size	LVV	Manufacturer Specified	Manufacturer Specific
Lower Size	LVV	Manufacturer Specified	Manufacturer Specific

8.8.2 *Sensor-AI-Counter Object Services* — The services provided by the Sensor-AI-Counter object are defined in SEMI E54.1 [1]. There are no additional services required for the Sensor AI-Counter object.

8.8.3 *Sensor-AI-Counter Object Behavior* — The behavior exhibited by the Sensor-AI-Counter object is inherited from the Sensor-AI object defined in SEMI E54.1. Additional specific behavior associated with the Sensor-AI-Counter object is defined below.

8.8.3.1 *Sensor-AI-Counter OPERATING Application Process*

8.8.3.1.1 A reading is retrieved from the physical median noise sensor. This reading is filtered through an analysis of a manufacturer specified number of samples. The *value* attribute is set to this processed reading. This process is called "counting" and is executed only when the Device Manager Object is in the EXECUTING state as defined in the Common Device Model.

8.8.3.1.2 Whenever an Interlock alarm, Sensor Not Detected alarm, Leak Detected alarm, and Sensor Type Change alarm is active the ISPM device shall stop "counting" until the alarm clears (see Section 8.2).

8.8.3.1.3 The ISPM is a device that measures and counts particles. As part of counting, particles are classified by size and count and accumulated in bins over time intervals then reported. The number of bins varies by vendor and device model.

8.8.3.1.4 The *value* attributes for all Sensor-AI-Counter objects, shall be held at zero unless the device laser is on and stable (i.e., Laser Status attribute of Device Manager Object has a value indicating "LASER ON AND STABLE"). The *value* attribute of any Sensor-AI-Counter object may be cleared (to zero) in a counter sensor after being read; the conditions under which this clearing behavior occurs may be further specified in the associated Network Communication Standard or by the manufacturer. The *value* shall also be cleared (to zero) by a valid "Clear Counts" service from the SAC object or the "Reset" service issued to the Sensor AI Class Object.

8.8.3.1.5 When attempting to set the "Upper Size" attribute to a value above the capability of the sensor, or to a value equal to or below the current "Lower Size" attribute value, an error response shall be generated indicating an invalid operation has been attempted. Similarly, an attempt to set the "Lower Size" attribute to a value below the capability of the sensor, or to a value equal to or above the current "Upper Size" attribute value, an error response shall be generated indicating an invalid operation has been attempted.

8.9 *Assembly-ISPM Objects* — Assembly-ISPM objects inherit attributes, services, and behavior from the Assembly object. The Assembly object is the device component which provides a mechanism of grouping more than one attribute from one or more objects into a single data structure for communication over the network.

8.9.1 Table 21 identifies the Assembly-ISPM objects defined for the ISPM device.

Table 21 Assembly List

Object	*Access Network*	*Required*	*Form*
Assembly-ISPM#1	R	Yes	Status and Counters (Counters ≤ 1024); Default Assembly
Assembly-ISPM#2	R	No	Status, Count and Counters (Counters ≤ 1024)
Assembly-ISPM#3	R	No	Status Diagnostics, and Counters (Counters ≤ 1024)
Assembly-ISPM#4	R	No	Status, Tool State, Time Stamp, Duration, and Counters (Counters ≤ 1024)
Assembly-ISPM#5	R	No	Status, Tool State, Time Stamp, Duration, Diagnostics, and Counters (Counters ≤ 1024)
Assembly-ISPM#6	R	Yes	Status
Assembly-ISPM#7	R	No	Exception Detail Alarm
Assembly-ISPM#8	R	No	Exception Detail Warning
Assembly-ISPM#9	R	No	Exception Detail Alarm and Exception Detail Warning
Assembly-ISPM#40	RW	No	Device Configuration

8.9.2 *Assembly-ISPM Objects Attributes* — Table 22 provides a list of attributes common to all Assembly-ISPM object types.

Table 22 Assembly-ISPM Object Attributes

Attribute Name	*Attribute Identifier*	*Access Network*	*Required*	*Form*
Data (See NOTE 1.)	A1	RW	Yes	Structure as defined below

NOTE 1: Inherited from the Assembly object as shown in Figure 1.

8.9.2.1 *Data Attribute Format for Assembly-ISPM Objects* — The Data attribute of all Assembly-ISPM objects is a structured attribute containing an ordered list of attributes within its structure. In the following tables (23 through 31), the structure of the Data attribute for each of the Assembly-ISPM object types is defined.

Table 23 Assembly-ISPM #1 Object Data List

Data Index	*Source Object ID*	*Source Attribute ID*	*Description*
1	DM1	A12	Device Manager Exception Status
2	ISPMD16	SaiA16	Sensor-AI-Counter-ISPM #1 Value
3	ISPMD16	SaiA16	Sensor-AI-Counter-ISPM #2 Value
.	ISPMD16	SaiA16	Sensor-AI-Counter #3 to Sensor-AI-Counter-ISPM #N-1 Value
N≤1024 (See NOTE 1.)	ISPMD16	SaiA16	Sensor-AI-Counter-ISPM #N Value

NOTE 1: "N" is the value of the Sensor-AI-Counter object attribute "Number of Bins". Note also that the number of bins reported in an assembly is fixed throughout the life of the device, and is specified by the manufacturer.

Table 24 Assembly-ISPM #2 Object Data List

Data Index	*Source Object ID*	*Source Attribute ID*	*Description*
1	DM1	A12	Device Manager Exception Status
2	ISPMD2	SacA65	SAC Number of Bins
3	ISPMD16	SaiA16	Sensor-AI-Counter-ISPM #1 Value
4	ISPMD16	SaiA16	Sensor-AI-Counter-ISPM #2 Value
.	ISPMD16	SaiA16	Sensor-AI-Counter-ISPM #3 to Sensor-AI-Counter-ISPM #N-1 Value
N≤1024 (See NOTE 1.)	ISPMD16	SaiA16	Sensor-AI-Counter-ISPM #N Value

NOTE 1: "N" is the value of the Sensor-AI-Counter object attribute "Number of Bins". Note also that the number of bins reported in an assembly is fixed throughout the life of the device, and is specified by the manufacturer.

Table 25 Assembly-ISPM #3 Object Data List

Data Index	*Source Object ID*	*Source Attribute ID*	*Description*
1	DM1	A12	Device Manager Exception Status
2	ISPMD3	SaiA16	Sensor-AI-LCS Value (Laser Current Diagnostic)
3	ISPMD4	SaiA16	Sensor-AI-SLS Value (Stray Light Diagnostic)
4	ISPMD5	SaiA16	Sensor-AI-MNS Value (Median Noise Diagnostic)
5	–	–	Reserved (2 bytes)
6	ISPMD2	SacA65	SAC Number of Bins
7	ISPMD16	SaiA16	Sensor-AI-Counter-ISPM #1 Value
8	ISPMD16	SaiA16	Sensor-AI-Counter-ISPM #2 Value
.	ISPMD16	SaiA16	Sensor-AI-Counter-ISPM #3 to Sensor-AI-Counter-ISPM #N-1 Value

Data Index	*Source Object ID*	*Source Attribute ID*	*Description*
N ≤ 1024 (See NOTE 1.)	ISPMD16	SaiA16	Sensor-AI-Counter-ISPM #N Value

NOTE 1: "N" is the value of the Sensor-AI-Counter object attribute "Number of Bins". Note also that the number of bins reported in an assembly is fixed throughout the life of the device, and is specified by the manufacturer.

Table 26 Assembly-ISPM #4 Object Data List

Data Index	*Source Object ID*	*Source Attribute ID*	*Description*
1	DM1	A12	Device Manager Exception Status
2	DM1	A35	Device Manager Tool State
3	DM1	A41	Device Manager Time Stamp
4	ISPMD2	SacA67	SAC Duration
6	ISPMD2	SacA65	Sensor-AI-Counter Number of Bins
7	ISPMD16	SaiA16	Sensor-AI-Counter-ISPM #1 Value
8	ISPMD16	SaiA16	Sensor-AI-Counter-ISPM #2 Value
.	ISPMD16	SaiA16	Sensor-AI-Counter-ISPM #3 to Sensor-AI-Counter-ISPM #N-1 Value
N ≤ 1024 (See NOTE 1.)	ISPMD16	SaiA16	Sensor-AI-Counter-ISPM #N Value

NOTE 1: "N" is the value of the Sensor-AI-Counter object attribute "Number of Bins". Note also that the number of bins reported in an assembly is fixed throughout the life of the device, and is specified by the manufacturer.

Table 27 Assembly-ISPM #5 Object Data List

Data Index	*Source Object ID*	*Source Attribute ID*	*Description*
1	DM1	A12	Device Manager Exception Status
2	DM1	A35	Device Manager Tool State
3	DM1	A41	Device Manager Time Stamp
4	ISPMD2	SacA67	SAC Duration
5	ISPMD3	SaiA16	Sensor-AI-LCS Value (Laser Current Diagnostic)
6	ISPMD4	SaiA16	Sensor-AI-SLS Value (Stray Light Diagnostic)
7	ISPMD5	SaiA16	Sensor-AI-MNS Value (Median Noise Diagnostic)
8	–	–	Reserved (2 bytes)
9	ISPMD2	SacA65	SAC Number of Bins
10	ISPMD16	SaiA16	Sensor-AI-Counter-ISPM #1 Value
11	ISPMD16	SaiA16	Sensor-AI-Counter-ISPM #2 Value

Data Index	*Source Object ID*	*Source Attribute ID*	*Description*
.	ISPMD16	SaiA16	Sensor-AI-Counter-ISPM #3 to Sensor-AI-Counter-ISPM #N-1 Value
N ≤ 1024*	ISPMD16	SaiA16	Sensor-AI-Counter-ISPM #N Value

NOTE 1: "N" is the value of the Sensor-AI-Counter object attribute "Number of Bins". Note also that the number of bins reported in an assembly is fixed throughout the life of the device, and is specified by the manufacturer.

Table 28 Assembly-ISPM #6 Object Data List

Data Index	*Source Object ID*	*Source Attribute ID*	*Description*
1	DM1	A12	Device Manager Exception Status

Table 29 Assembly-ISPM #7 Object Data List

Data Index	*Source Object ID*	*Source Attribute ID*	*Description*
1	DM1	A12	Device Manager Exception Status
2	DM1	A13	Device Manager Exception Detail Alarm

Table 30 Assembly-ISPM #8 Object Data List

Data Index	*Source Object ID*	*Source Attribute ID*	*Description*
1	DM1	A12	Device Manager Exception Status
2	DM1	A14	Device Manager Exception Detail Warning

Table 31 Assembly-ISPM #9 Object Data List

Data Index	*Source Object ID*	*Source Attribute ID*	*Description*
1	DM1	A12	Device Manager Exception Status
2	DM1	A13	Device Manager Exception Detail Alarm
3	DM1	A14	Device Manager Exception Detail Warning

Table 32 Assembly-ISPM #40 Object Data List

Data Index	*Source Object ID*	*Source Attribute ID*	*Description*
1	DM1	A33	Device Manager Gain
2	DM1	A13	Device Manager Filter Bandwidth
3	ISPMD2	SacA65	SAC Number of Bins
4	ISPMD3	SaiA73	Sensor-AI-LCS Alarm Trip Point High (See NOTE 1.)
5	ISPMD3	SaiA76	Sensor-AI-LCS

Data Index	*Source Object ID*	*Source Attribute ID*	*Description*
			Warning Trip Point High (See NOTE 1.)
6	ISPMD3	SaiA74	Sensor-AI-LCS Alarm Trip Point Low (See NOTE 1.)
7	ISPMD3	SaiA77	Sensor-AI-LCS Warning Trip Point Low (See NOTE 1.)
8	ISPMD4	SaiA73	Sensor-AI-SLS Alarm Trip Point High (See NOTE 1.)
9	ISPMD4	SaiA76	Sensor-AI-SLS Warning Trip Point High (See NOTE 1.)
10	ISPMD4	SaiA74	Sensor-AI-SLS Alarm Trip Point Low (See NOTE 1.)
11	ISPMD4	SaiA77	Sensor-AI-SLS Warning Trip Point Low (See NOTE 1.)
12	ISPMD5	SaiA73	Sensor-AI-MNS Alarm Trip Point High (See NOTE 1.)
13	ISPMD5	SaiA76	Sensor-AI-MNS Warning Trip Point High (See NOTE 1.)
14	ISPMD5	SaiA74	Sensor-AI-MNS Alarm Trip Point Low (See NOTE 1.)
15	ISPMD5	SaiA77	Sensor-AI-MNS Warning Trip Point Low (See NOTE 1.)

NOTE 1: Objects to which these attributes are linked are optional. If the associated object is not supported in the ISPM device, the data field for the associated attribute in this assembly is set to zero and should be interpreted as having no meaning.

SEMI E54.11-0301
SPECIFIC DEVICE MODEL FOR ENDPOINT DEVICES

This specific device model was technically approved by the Global Information & Control Committee and is the direct responsibility of the North American Information & Control Committee. Current edition approved by the North American Regional Standards Committee on November 22, 2000. Initially available at www.semi.org January 2001; to be published March 2001.

1 Purpose

1.1 This specification is part of a suite of standards that specify the implementation of SEMI standards for the Sensor/Actuator Network. The specific purpose of this specification is to describe a network independent application model comprised of device objects that are common to all Endpoint Devices on a semiconductor equipment Sensor/Actuator communication network.

2 Scope

2.1 An Endpoint device (EPD) is a device that measures and monitors process characteristics to determine when a specific threshold or event has been obtained usually to signal the completion of a process or process step. These endpoint devices are, but not limited to, devices that may classify a process endpoint by determining the size and count of particles in the process environment, detecting and determining optical light from a sample region of the environment's space, or determining motor current of an equipment component.

2.2 This specification specifically addresses the minimum attributes, services and behavior an Endpoint (EPD) device must support to be interoperable on the Sensor/Actuator Network.

2.3 This specification is intended to ensure a high-degree of device interoperability on the Sensor/Actuator Network, while still allowing flexibility for product differentiation and technology evolution.

2.4 The model specified in this specification is used in conjunction with the Sensor/Actuator Network Common Device Model (CDM) to completely describe the Endpoint device (EPD) as it appears from the network interface.

2.5 This specification, together with the Sensor/Actuator Network Standard, the Sensor/Actuator Network Common Device Model, and a Sensor/Actuator Network Communication Specification, form a complete interoperability specification for the EPD.

2.6 To comply with this specification, a device must implement and support, at a minimum, the required attributes, services, and behavior identified in these documents. Support for optional attributes, services and behavior is not required to be compliant to this specification. Optional attributes, services, and behavior are specified in these documents to promote further device interoperability as features evolve and are adopted by more manufacturers. If optional attributes, services, and behavior are implemented for this device they must be implemented as identified in this document.

2.7 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 health practices and determine the applicability or regulatory limitations prior to use

3 Limitations

3.1 This specification is a companion to a suite of specifications that together make up the Sensor/Actuator Network Communication standard. Therefore, using portions of this specification that relate to network communications necessarily requires an understanding of the associated network specification.

3.2 As this document is a specification for the Endpoint Device Model, it does not contain any definition of objects, attributes, services, or behavioral descriptions that are already defined in the Sensor/Actuator Network Common Device Model (CDM). Additional attributes, attribute assignments, services and/or service parameters that are Endpoint Device specific and/or implementation specific are contained in this specification.

3.3 While this specification is sufficient to completely describe the EPD as it appears from the network, it does not fully describe behavior of a specific endpoint device type which is not visible from the network. This allows flexibility in implementation techniques and product differentiation between manufacturers. Manufacturer specific objects may be defined by the manufacturer but are, by definition, outside the scope of this standard.

3.4 This specification is compatible, but not compliant with SEMI E39. This means that although this specification does not require compliance with SEMI E39, it is extensible such that implementations may be developed that are fully compliant with both standards. Note that the concepts and terminology of this specification are compatible with those of SEMI E39.

However, SEMI E39 has specific requirements that are intended for higher level applications and thus are not applied to the Endpoint Device Model.

3.5 Operation over the entire range specified for an attribute within a specific object is not a requisite for compliance with this specification.

4 Referenced Standards

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

4.1 *SEMI Standards*

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

SEMI E54 — Sensor/Actuator Network Standard

SEMI E54.1— Standard for Sensor/Actuator Network Common Device Model

5 Terminology

5.1 *Abbreviations and Acronyms*

5.1.1 *EP* — an EPD sensor named Endpoint.

5.2 *Definitions*

5.2.1 *Endpoint Device (EPD)* — a self-contained device, consisting of device specific signal-processing electronics, which is capable of monitoring and measuring the occurrence of a process endpoint.

5.2.2 *Endpoint Detection Event* — consists of the device operation of monitoring, measuring, analyzing, waiting, and reporting endpoint.

5.3 This document inherits the terminology defined by SEMI E54.1

5.3.1 *Attribute*

5.3.2 *Behavior*

5.3.3 *Boolean (BOOL)*

5.3.4 *Byte*

5.3.5 *Character*

5.3.6 *Common Device Model (CDM)*

5.3.7 *Data Type*

5.3.8 *Data Units*

5.3.9 *Device*

5.3.10 *Device Manager (DM) Object*

5.3.11 *Device Model*

5.3.12 *Double Integer (DINT)*

5.3.13 *Enumerated Byte*

5.3.14 *Full Scale Range*

5.3.15 *Instance*

5.3.16 *Last Valid Value (LVV)*

5.3.17 *Long Integer (LINT)*

5.3.18 *Long Real (LREAL)*

5.3.19 *Manufacturer*

5.3.20 *Nibble*

5.3.21 *Null Character*

5.3.22 *Object*

5.3.23 *Real (REAL)*

5.3.24 *S, A, and C Objects*

5.3.25 *Sensor Actuator Controller (SAC) Object*

5.3.26 *Service*

5.3.27 *Signed Integer (INT)*

5.3.28 *Short Integer (SINT)*

5.3.29 *State Diagram*

5.3.30 *Test String*

5.3.31 *Unsigned Double Integer (UDINT)*

5.3.32 *Unsigned Double Long Integer (UDLINT)*

5.3.33 *Unsigned Integer (UINT)*

5.3.34 *Unsigned Long Integer (ULINT)*

5.3.35 *Unsigned Short Integer (USINT)*

6 Requirements and Specifications

6.1 In order to implement this standard in an Endpoint Device, it is necessary to also implement SEMI E54.1 and one of the Sensor/Actuator Network Communication Standards (SEMI E54.4–E54.9). See Section 3 for more information on a complete interoperability standard.

6.2 This specification also requires the implementation of a Date_And_Time data structure used to represent the current device Date and Time. Table 1 defines the format of the Date_And_Time data type.

Table 1 Date_And_Time Format

Data Item	*Description*	*Range*
1	Number of milliseconds since midnight	Unsigned Double Integer (UDINT)
2	Number of days since 1/1/72	Unsigned Integer (UINT)

7 Conventions

7.1 This document embraces the conventions and notations stated in section 6 of SEMI E54.1.

8 Device High Level Structure

8.1 *General Description*

8.1.1 The high-level object view of an Endpoint Device (EPD) profile is shown in Figure 1.

8.1.2 Note that the profile for an "EPD Device" object is depicted in Figure 1 only for the purposes of illustrating a high level view of the device and its component objects. In the context of this document, this object is not addressable, does not have addressable attributes, does not have accessible services and, does not exhibit any defined behavior.

8.1.3 In the remainder of this section, this document defines in detail the component objects unique to the EPD device. References, rather than definitions, are included for the DM, the SAC and other objects defined in SEMI E54.1.

8.1.4 Many of the objects defined in this document inherit properties from other objects. The properties inherited include attribute, service and behavior definitions. These other objects are specified here or in SEMI E54.1.

8.1.5 This document provides for future extensions as well as manufacturer specific enhancements by reserving object attribute identifiers and object service identifiers. Specifically all object definitions in this document specify or reserve the first 64 attribute identifiers (A1 through A64) and the first 64 service identifiers (S1 through S64) allowing manufacturers to specify identifiers beyond these ranges. Additionally, byte enumerated attributes are specified or reserved from 0 to 63 allowing manufacturers to specify an enumeration beyond this range (64 to 255).

8.1.6 *Endpoint Device (EPD) Description* — An Endpoint device profile is composed of the component objects and object relationships shown in Figure 1.

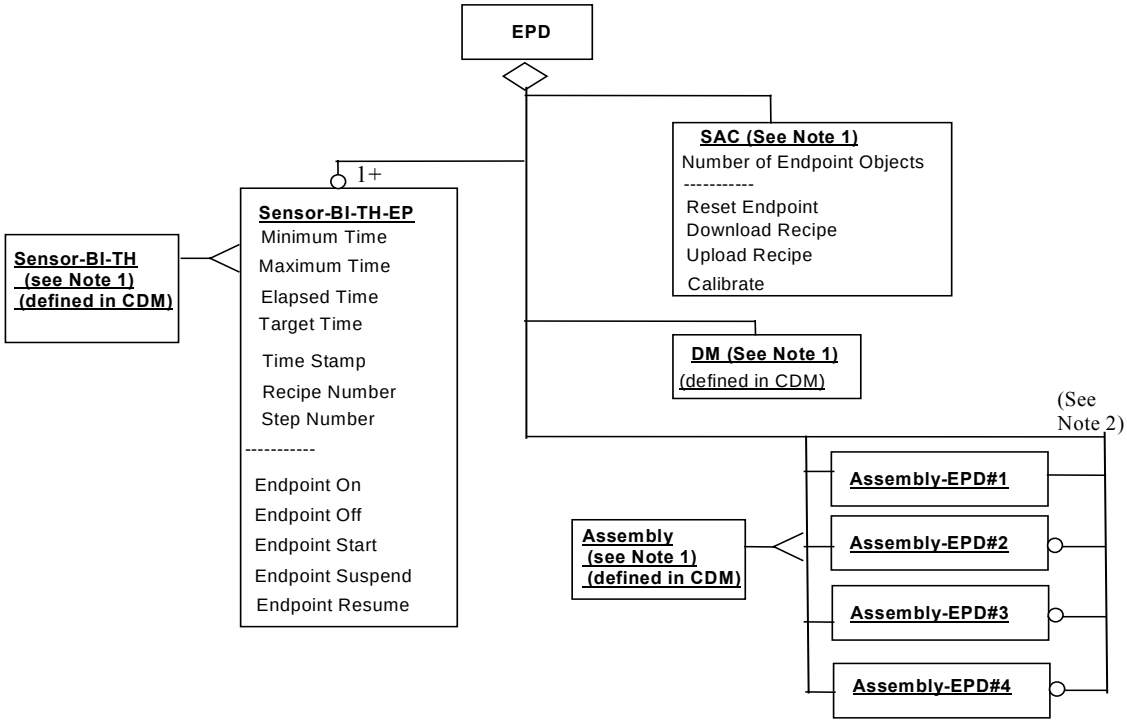


Figure 1
Endpoint Device High Level Structure

NOTE 1: The Sensor-BI-TH, DM, SAC, and Assembly are defined in the CDM. Additional attributes and services are added to support the EPD.

NOTE 2: Assembly-EPD #1 object is required. Other Assembly-EPD objects are optional.

8.1.7 *General Requirements*

8.1.7.1 *Device Objects* — All objects are defined in terms of their object name and Class/Object identifier. Identifiers for all objects described in this document are summarized in Table 2.

Table 2 Endpoint Device Objects

Referenced Document Section	***Object Name***	***Class/Object Identifier***	***Minimum #***	***Maximum #***
8.2	Device Manager (DM)	EPD1	1	1
8.3	Sensor Actuator Controller (SAC)	EPD2	1	1
8.5	Sensor-BI-TH-EP	EPD3	1	1024
8.6	Assembly-EPD#1	EPD4	1	1
8.6	Assembly-EPD#2	EPD5	0	1
8.6	Assembly-EPD#3	EPD6	0	1
8.6	Assembly-EPD#4	EPD7	0	1
—	Reserved	EPD7 – EPD63	-	—
—	Manufacturer Specified	> EPD64	-	—

8.1.7.2 *Object Services* — Not all object services listed in this document can necessarily be requested over the network. They are included in this document because their behavior may generate network activity.

8.1.7.3 *Object Behavior* — A network specific service error response is generated for all service requests received over the network that are not supported by the object, or contain a parameter value which is beyond the supported range, or which is otherwise invalid.

8.2 *Device Manager Object (DM)* — The Device Manager object is the device component responsible for managing and consolidating the device operation as specified in SEMI E54.1. The following sections specify the components of the DM object that are not specified in the Common Device Model or require further definition than specified in SEMI E54.1.

8.2.1 *Device Manager Object Attributes* — Required and optional DM object attributes are listed in Table 3.

Table 3 DM Object Attributes

Attribute Name	*Attribute Identifier*	*Access Network*	*Required*	*Form*
Device Type	A1	R	Yes	Refer to SEMI E54.1.
Exception Detail Alarm	A13	R	No	Refer to SEMI E54.1.
Exception Detail Warning	A14	R	No	Refer to SEMI E54.1.
Reserved	A33-A64	—	—	Reserved for SDM future expansion
Manufacturer Specified	> A64	—	—	Manufacturer Specific attributes

8.2.1.1 *Device Type* — An attribute that uniquely identifies the type of the device on the network. The device type attribute is assigned as follows:

Endpoint Device = “EPD”

NOTE 2: If the “Endpoint Device” functionality is implemented by another Sensor/Actuator Network Specific Device Model, the ‘*Device Type*’ attribute value may be specified by the manufacturer to identify the other Specific Device Model.

8.2.1.2 *Exception Detail Alarm (Optional)* — An attribute that identifies the detailed alarm status of the device. Table 4 defines the bit assignments associated with the alarm exception detail.

Table 4 Exception Detail Alarm Bit Assignments

Bit	*Device Specific Alarm[0]*
0	Unexpected Conditions Detected
1	Reserved
2	Sensor Not Detected
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Endpoint Failure

8.2.1.3 *Exception Detail Warning (Optional)* — An attribute that identifies the detailed warning status of the device. Table 5 defines the bit assignments associated with the warning exception detail.

Table 5 Exception Detail Warning Bit Assignments

Bit	*Device Specific Warning[0]*
0	Unexpected Conditions Detected
1	Reserved
2	0
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Endpoint Warning

8.2.1.4 *Manufacturer Exception Detail Size (Optional)* — An attribute that specifies the number of exception detail bytes included in the alarm or warning details.

8.2.1.5 *Initial and Default Values*

Table 6 DM Object Attribute Initial and Default Values

Attribute	*Initial Value*	*Default Value*	*Comment*
Device Type	EPD	EPD	EPD = Endpoint Device
Exception Detail Alarm	0	0	
Exception Detail Warning	0	0	

8.2.2 *Device Manager Object Services* — The services provided by the Device Manager object are defined in SEMI E54.1. There are no additional services required for the Device Manager object.

8.2.3 *Device Manager Object Behavior* — The behavior exhibited by the Device Manager object is defined in SEMI E54.1. There is no additional behavior specified for the Device Manager object.

8.3 *Sensor Actuator Controller Object (SAC)* — The Sensor Actuator Controller object is the device component responsible for coordinating the interaction of the EPD device with the sensory/actuation/control environment as specified in SEMI E54.1. The following sections specify the components of the SAC object that are not specified in the Common Device Model or require further definition than specified in the Common Device Model.

8.3.1 *Sensor Actuator Controller Object Attributes* — The attributes provided by the Sensor Actuator Controller object are defined in SEMI E54.1. Table 7 contains the additional attributes required for the Sensor Actuator Controller object.

Table 7 SAC Object Attributes

Attribute Name	*Attribute Identifier*	*Access Network*	*Required*	*Form*
Number of Endpoint Objects	SacA65	R	No	UINT

8.3.1.1 *Number of Endpoint Objects* — An attribute that specifies the number of endpoint objects in the device. This value represents the number of Sensor-BI-TH-EP objects in the device. If this attribute is not supported then the default value of 1 Endpoint object is supported.

8.3.1.2 *Initial and Default Values*

Table 8 SAC Object Attributes Initial and Default Values

Attribute	*Initial Value*	*Default Value*	*Comment*
Number of Endpoint Objects	Manufacturer Specified	Manufacturer Specified	There must be at least one Endpoint object.

8.3.2 *Sensor Actuator Controller Object Services* — The services provided by the Sensor Actuator Controller object are defined in SEMI E54.1. Table 9 contains the additional services required for the Sensor Actuator Controller object.

Table 9 SAC Object Services

Service	*Service Identifier*	*Type*	*Description*
Reset Endpoint	S33	R	Used to 'Reset' all of the Sensor-BI-TH-EP sensor objects.
Download Recipe	S34	R	Used to download the Sensor-BI-TH-EP sensor objects with manufacturer specific recipe data.
Upload Recipe	S35	R	Used to upload, from the device, manufacturer specific recipe data.
Calibrate	S36	R	Used to execute a device manufacture specific calibration procedure.
Reserved	S37–S64	—	Reserved for future expansion
Manufacturer Specified	> S64	—	Manufacturer Specific services

8.3.2.1 *Reset Endpoint (Optional)* — This service is used to instruct all of the Sensor-BI-TH-EP objects to perform a 'Reset' of their endpoint monitoring and measuring events. A one-time reset of their respective Sensor-BI-TH-EP attributes is performed. There are no parameters required for this service.

8.3.2.2 *Download Recipe (Optional)* — This service is used to set the recipe parameters associated with the endpoint sensor or a specific Sensor-BI-TH-EP object. The format and type of data comprising recipe data and the mechanism implemented to interrupt recipe data and distribute its contents to the appropriate Sensor-BI-TH-EP object is manufacturer specific. The parameter 'Recipe Data' may be a formatted data structure or a list of individual data items. The following table describes the parameters specified for this service.

Table 10 Download Recipe Service Parameter Definitions

Parameter	*Request/ Indication*	*Response/ Confirmation*	*Data Type*	*Description*
Recipe Data #1	M	U	Manufacturer Specific	The recipe parameter format is manufacturer specific. The implementation behavior required to support this service is also manufacturer specific.
Recipe Data #2	M	U	Manufacturer Specific	The recipe parameter format is manufacturer specific. The implementation behavior required to support this service is also manufacturer specific.
Recipe Data #3 through Recipe Data #N-1	M	U	Manufacturer Specific	The recipe parameter format is manufacturer specific. The implementation behavior required to support this service is also manufacturer specific.
Recipe Data #N	M	U	Manufacturer Specific	The recipe parameter format is manufacturer specific. The implementation behavior required to support this service is also manufacturer specific.

8.3.2.3 *Upload Recipe (Optional)* — This service is used to read the recipe parameters associated with the endpoint sensor or a specific Sensor-BI-TH-EP object. The format and type of data comprising recipe data and the mechanism implemented to assemble recipe data to be read is manufacturer specific. The parameter 'Recipe Data' may be a formatted data structure or a list of individual data items. The following table describes the parameters specified for this service.

Table 11 Upload Recipe Service Parameter Definitions

Parameter	*Request/ Indication*	*Response/ Confirmation*	*Data Type*	*Description*
Recipe Data #1	M	M	Manufacturer Specific	The recipe parameter format is manufacturer specific. The implementation behavior required to support this service is also manufacturer specific.
Recipe Data #2	M	M	Manufacturer Specific	The recipe parameter format is manufacturer specific. The implementation behavior required to support this service is also manufacturer specific.
Recipe Data #3 through Recipe Data #N-1	M	M	Manufacturer Specific	The recipe parameter format is manufacturer specific. The implementation behavior required to support this service is also manufacturer specific.
Recipe Data #N	M	M	Manufacturer Specific	The recipe parameter format is manufacturer specific. The implementation behavior required to support this service is also manufacturer specific.

8.3.2.4 *Calibrate (Optional)* — This service is used to set the calibration parameters associated with the endpoint sensor or a specific Sensor-BI-TH-EP object. The format and type of data comprising calibration data and the mechanism implemented to interrupt calibration data and execute a calibration algorithm is manufacturer specific. The parameter 'Calibration Data' may be a formatted data structure or a list of individual data items. The following table describes the parameters specified for this service.

Table 12 Calibration Service Parameter Definitions

Parameter	*Request/ Indication*	*Response/ Confirmation*	*Data Type*	*Description*
Calibration Data #1	C	C	Manufacturer Specific	The calibration parameter format is manufacturer specific. The implementation behavior required to support this service is also manufacturer specific.
Calibration Data #2	C	C	Manufacturer Specific	The calibration parameter format is manufacturer specific. The implementation behavior required to support this service is also manufacturer specific.
Calibration Data #3 through Calibration Data #N-1	C	C	Manufacturer Specific	The Calibration parameter format is manufacturer specific. The implementation behavior required to support this service is also manufacturer specific.
Calibration Data #N	C	C	Manufacturer Specific	The calibration parameter format is manufacturer specific. The implementation behavior required to support this service is also manufacturer specific.

8.3.3 *Sensor Actuator Controller Object Behavior*

8.3.3.1 The behavior exhibited by the Sensor Actuator Controller object is defined in SEMI E54.1. Additional behavior is detailed below.

8.3.3.2 The *"Reset Endpoint"* service will issue an "Endpoint Restart" object service to each Sensor-BI-TH-EP sensor object.

8.4 *Sensor-BI-TH Object* — The Sensor-BI object is the device component responsible for coordinating the behavior common to all Boolean input threshold sensor elements in the EPD device as specified in SEMI E54.1.

8.4.1 *Sensor-BI-TH Object Attributes* — The attributes provided by the Sensor-BI-TH object are defined in SEMI E54.1. There are no additional attributes required for the Sensor-BI-TH object.

8.4.2 *Sensor-BI-TH Object Services* — The services provided by the Sensor-BI-TH object are defined in SEMI E54.1. There are no additional services required for the Sensor-BI-TH object.

8.4.3 *Sensor-BI-TH Object Behavior* — The behavior exhibited by the Sensor-BI-TH object is defined in SEMI E54.1. There is no additional behavior required for the Sensor-BI-TH object.

8.5 *Sensor-BI-TH-EP Object* — The Sensor-BI-TH-EP (Endpoint) object inherits the attributes, services, and behavior of the Sensor-BI-TH as defined in SEMI E54.1. The Sensor-BI-TH-EP is the device component responsible for retrieving a reading, or readings, from the device specific signal-processing sensors, optionally processing the readings with a manufacturer specified algorithm, or algorithms, and then making the endpoint result available through the "*Value*" attribute.

8.5.1 *Sensor-BI-TH-EP Object Attributes* — The attributes provided by the Sensor-BI-TH object are defined in SEMI E54.1. The Sensor-BI-TH object attribute content and its attribute extensions to support the Sensor-BI-TH-EP object are listed in the table below.

Table 13 Sensor-BI, Sensor-BI-TH, and Sensor-BI-TH-EP Object Attributes

Attribute Name	*Attribute Identifier*	*Access Network*	*Required*	*Form*
Value	SbithepA16	R	Yes	BOOL
Reading Valid	SbithA64	R	Yes	BOOL
State	SbithA65	R	No	Enumerated, Byte
Status	SbithA66	R	No	Enumerated, Byte
Minimum Time	EpA1	RW*	No	UDINT
Maximum Time	EpA2	RW*	No	UDINT
Target Time	EpA3	RW*	No	UDINT
Elapsed Time	EpA4	R	No	UDINT
Time Stamp	EpA5	R	No	Date_And_Time
Recipe Identifier	EpA6	RW*	No	Text String
Step Identifier	EpA7	RW*	No	Text String
Reserved	EpA8 –EpA64	—	—	Reserved for future expansion
Manufacturer Specified	> EpA64	—	—	Manufacturer Specific attributes

NOTE 1: "*" Indicates that the specific attribute is nonvolatile. Nonvolatile requires that the current attribute value be maintained through a component power cycle.

8.5.1.1 *Value* — The attribute that maintains the current endpoint event result. The value of the *'Value'* attribute is read as a Boolean (True or False) endpoint detection event result.

8.5.1.2 *Reading Valid* — An attribute which specifies whether the "*Value*" attribute contains a valid value. This attribute is Boolean that can take on one of the following values:

0 = INVALID
1 = VALID

NOTE 3: The 'Reading Valid' attribute is identified as an optional attribute of the SBITH object class but is identified as a required attribute for the SBITHEP object class.

8.5.1.3 *State* — An attribute that records the current state of the endpoint object. This attribute is an enumerated byte. The possible enumeration and the requirement for support are as follows:

0 = ENDPOINT OFF (required)
1 = ENDPOINT IN PROCESS (required)
2 = ENDPOINT IDLE (optional)
3 = ENDPOINT SUSPENDED (optional)
4 = ENDPOINT FAILURE (optional)

5-63 = Reserved
64-255 = Manufacturer Specified (optional)

8.5.1.4 *Status* — An attribute which specifies whether endpoint detection event reporting is active based upon the services 'Endpoint On', 'Endpoint Start', and 'Endpoint Off or On'. This attribute is an enumerated byte that can take on one of the following values:

0 = Endpoint Off
1 = Endpoint On

8.5.1.5 *Minimum Time* — An attribute that specifies the minimum time in milliseconds for an endpoint detection event. No attempt to report an endpoint detection event will take place until the minimum time specified has expired.

8.5.1.6 *Maximum Time* — An attribute that specifies the maximum time in milliseconds for an Endpoint detection event before an alarm is reported.

8.5.1.7 *Target Time* — An attribute that specifies the expected time in milliseconds for an Endpoint detection event.

8.5.1.8 *Elapsed Time* — An attribute that specifies in milliseconds the amount of time that has elapsed since the beginning of the current endpoint detection event. This attribute will behave as a count up timer that is frozen when the endpoint event is detected.

8.5.1.9 *Time Stamp* — An attribute that specifies the time when the endpoint detection event completed.

8.5.1.10 *Recipe Identifier* — An attribute that specifies a manufacturer specific endpoint algorithm or algorithms to be utilized to determine the current endpoint detection event. The interpretation of this attribute is manufacturer specific.

8.5.1.11 *Step Identifier* — An attribute that specifies a process recipe step that is associated with the current endpoint detection recipe and/or event. The interpretation of this attribute is manufacturer specific.

8.5.1.12 *Initial and Default Values*

Table 14 Sensor-BI and Sensor-BI-EP Object Attributes Initial and Default Values

Attribute	*Initial Value*	*Default Value*	*Comment*
Value	FALSE	FALSE	
Reading Valid	0	0	Invalid Reading
State	LVV	0	ENDPOINT OFF
Status	0	Endpoint Off	
Minimum Time	LVV	0	Milliseconds
Maximum Time	LVV	0	Milliseconds
Target Time	LVV	0	Milliseconds
Elapsed Time	LVV	0	Milliseconds
Time Stamp	LVV	Manufacture Specified	Date_And_Time
Recipe Identifier	LVV	Manufacture Specified	
Step Identifier	LVV	Manufacturer Specified	

8.5.2 *Sensor-BI-EP Object Services* — The services provided by the Sensor-BI-EP are inherited from the Sensor-BI object defined in SEMI E54.1. The Sensor-BI-EP object supports the additional services listed in Table 15 below.

Table 15 Sensor-BI-EP Object Services

Service	*Service Identifier*	*Type*	*Description*
Endpoint On	S1	R	Used to prompt the endpoint object to go from the ENDPOINT OFF state to ENDPOINT ON/ENDPOINT IDLE state as defined by the State and Status attribute.
Endpoint Off	S2	R	Used to prompt the endpoint object to go immediately to the ENDPOINT OFF state as defined by the State and Status attribute. All endpoint detection processes are aborted and all active timers are stopped.
Endpoint Start	S3	R	Used to prompt the endpoint object to go from the ENDPOINT OFF or ENDPOINT IDLE (optional) state to the ENDPOINT IN PROCESS state and begin the endpoint detection event process.
Endpoint Suspend	S4	R	Used to prompt the endpoint object to go to the ENDPOINT SUSPENDED state from the ENDPOINT IN PROCESS state. All endpoint detection event processes and active timers are suspended.
Endpoint Resume	S5	R	Used to prompt the endpoint object to go from the ENDPOINT SUSPENDED state to the ENDPOINT IN PROCESS state. All endpoint detection event processes and active timers resume from where they were suspended.
Reserved	S6–S64	—	Reserved for future expansion
Manufacturer Specified	> S64	—	Manufacturer Specific services

8.5.2.1 *Endpoint On (Required)* — This service is used to prompt the endpoint object to go from the ENDPOINT OFF state to the ENDPOINT ON / ENDPOINT IDLE state as defined by the State attribute. If the device does not support the ENDPOINT IDLE state then the endpoint object goes immediately to the ENDPOINT ON / ENDPOINT IN PROCESS state. If the State attribute is already set to ENDPOINT IDLE or ENDPOINT IN PROCESS, the endpoint object is set to the appropriate state. If the device turns on successfully, a "success" response is returned. If the device fails to turn on properly, a "fail" response is returned and the endpoint object remains in the appropriate state.

8.5.2.2 *Endpoint Off (Required)* — This service is used to prompt the endpoint object to go immediately to the ENDPOINT OFF state from the ENDPOINT ON state as defined by the State attribute. All endpoint detection event processes are aborted and all active timers are stopped. If the state attribute is already set to ENDPOINT OFF, the endpoint object remains in the appropriate state. If the device turns off successfully, a "success" response is returned. If the device fails to turn off properly, a "fail" response is returned and the endpoint object remains in the current state.

8.5.2.3 *Endpoint Start (Optional)* — This service is used to prompt the endpoint object to go from the ENDPOINT OFF or ENDPOINT IDLE state to the ENDPOINT IN PROCESS state and begin the endpoint detection event process with the initial endpoint attribute parameter values. If the State attribute is not currently in the ENDPOINT OFF or ENDPOINT IDLE state, an "object state conflict" error response is returned and the endpoint event remains in the appropriate state.

8.5.2.4 *Endpoint Suspend (Optional)* — This service is used to prompt the endpoint object to go from the ENDPOINT IN PROCESS state to the ENDPOINT SUSPENDED state and suspend the endpoint detection event process. All active timers are suspended and held at their current values. If the State attribute is not currently in the ENDPOINT IN PROCESS state, an "object state conflict" error response is returned and the endpoint object remains in the appropriate state. The Endpoint Suspend service is conditional on the Endpoint Resume service being supported.

8.5.2.5 *Endpoint Resume (Optional)* — This service is used to prompt the endpoint object to go from the ENDPOINT SUSPENDED state to the ENDPOINT IN PROCESS state and resume the endpoint detection event process. The process resumes using the endpoint attribute parameter values and timer readings that were saved when the endpoint was suspended by the Endpoint Suspend service request. If the Endpoint Resume service is issued while the State attribute is not in the ENDPOINT SUSPENDED state, an "object state conflict" error response is returned and the endpoint event remains in its existing state. The Endpoint Resume service is conditional on the Endpoint Suspend service being supported.

8.5.3 *Sensor-BI-TH-EP Object Behavior* — The behavior exhibited by the Sensor-BI-TH-EP object is

inherited from the Sensor-BI-TH object defined in SEMI E54.1. Additional specific behavior associated with the Sensor-BI-TH-EP object is defined below.

8.5.3.1 *Sensor-BI-EP OPERATING Application Process* — When in the ENDPOINT ON / ENDPOINT IN PROCESS state, the '*Value'* attribute is set to FALSE and the '*Reading Value'* attribute is set to VALID. *A* reading, or readings, may be retrieved from the device physical signal processing electronics. This reading may be filtered, analyzed, and corrected with a manufacturer-specified algorithm. This corrected reading becomes the input to the endpoint formula to generate the 'V*alue'* attribute as referenced in SEMI E54.1. This process is called "endpoint detection" and is executed only when the Sensor-BI-TH-EP object is in the OPERATING state as defined in SEMI E54.1. When an Endpoint event is detected, the '*Value'* attribute is set to TRUE and the '*Reading Valid'* attribute is VALID. When not in the OPERATING state, the value of the '*Reading Valid*' attribute shall be set to INVALID. When in one of the sub-states of the OPERATING state, the validity of the '*Value*' attribute shall be manufacturer specific. Required sub-states within the OPERATING state, descriptions of these sub-states, and a transition matrix associated with these sub-states are given in Figure 2, Table 16 and Table 17 respectively.

8.5.3.1.1 Whenever a 'Sensor Not Detected' alarm, 'Endpoint Failure' alarm or 'Unexpected Conditions Detected' alarm is active, the EPD device shall turn off the endpoint detection event process and hold the device in the ENDPOINT OFF state until the alarm clears (see Section 8.2).

8.5.3.1.2 A device can concurrently monitor and report many endpoint detection events. The number of endpoint detection events varies by vendor and device model.

8.5.3.1.3 The 'Value' attributes for all Sensor-BI-TH-EP objects shall be held at 'FALSE' until a valid endpoint detection event is determined. An endpoint detection event is initiated when the endpoint service request 'Endpoint Start' or Endpoint On (if the state ENDPOINT ON / ENDPOINT IDLE is not supported) is received and the endpoint object can successfully transition to the ENDPOINT ON / ENDPOINT IN PROCESS state.

8.5.3.1.4 When attempting to set the "Minimum Time", "Maximum Time", and "Target Time" attribute to a value above the capability of the endpoint detection event an error response shall be generated indicating an invalid operation has been attempted.

8.5.3.1.5 When attempting to set the "Recipe Identifier" or "Step Identifier" attribute to an identifier outside the range of the endpoint detection event an error response shall be generated indicating an invalid operation has been attempted.

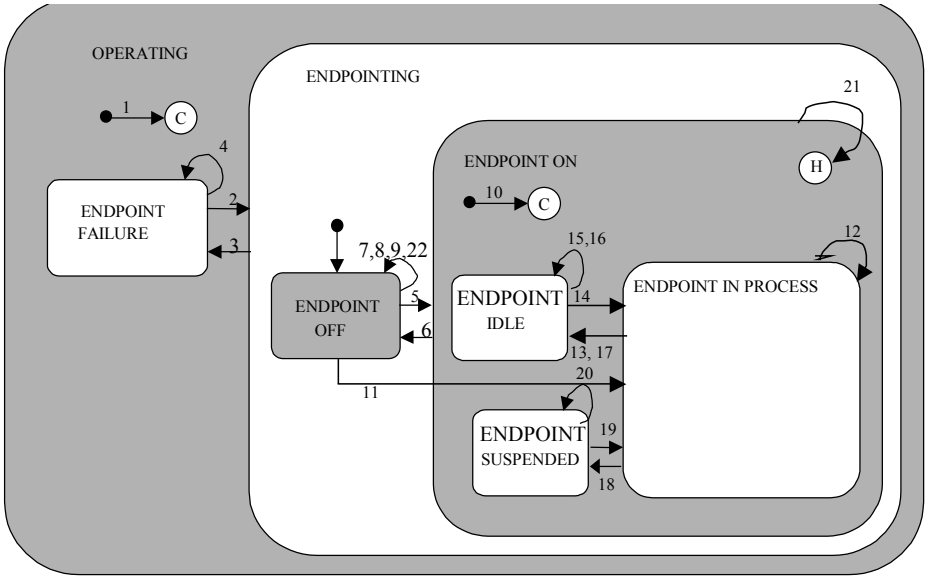


Figure 2
Sensor-BI-TH-EP Object Behavior Within the Operating State

Table 16 Sensor-BI-TH-EP Behavior OPERATING Sub-state Description

State	*Description*
OPERATING	Endpoint is in one of the following enumerated states as indicated in the Sensor-BI-TH-EP State attribute: ENDPOINT FAILURE (optional), ENDPOINT OFF, ENDPOINT IDLE (optional), ENDPOINT IN PROCESS, or ENDPOINT SUSPENDED (optional). Endpoint will respond to Endpoint On and Endpoint Off services as appropriate to move between sub-states within the ENDPOINTING and ENDPOINT ON states.
ENDPOINTING	Endpoint is in one of the following enumerated states as indicated in the Sensor-BI-TH-EP State attribute: ENDPOINT OFF, ENDPOINT IDLE, ENDPOINT SUSPENDED, and ENDPOINT IN PROCESS. Endpoint will respond to Endpoint On, Endpoint Off, Endpoint Start, Endpoint Suspend, and Endpoint Resume services as appropriate to move between sub-states within the ENDPOINTING state.
ENDPOINT OFF	This is a sub-state to ENDPOINTING; Endpoint Off is the status of the Sensor-BI-TH-EP Status attribute and ENDPOINT OFF is the enumerated state of the Sensor-BI-TH-EP State attribute. Endpoint object is **NOT** performing the "endpoint" detection event process. The endpoint object may be downloaded with new recipe parameters when in this state.
ENDPOINT ON	This is a sub-state of ENDPOINTING; Endpoint On is the status of the Sensor-BI-TH-EP Status attribute and ENDPOINT IDLE, ENDPOINT SUSPENDED or ENDPOINT IN PROCESS is the enumerated state of the Sensor-BI-TH-EP State attribute. Endpoint object may not be performing the "endpoint" detection event process.
ENDPOINT IDLE	This is a sub-state to ENDPOINT ON. Endpoint On is the status of the SENSOR-BI-TH-EP Status attribute and ENDPOINT IDLE is the enumerated state of the Sensor-BI-TH-EP State attribute. Endpoint object is not performing the "endpoint" detection event process. The endpoint object may be downloaded with new recipe parameters.
ENDPOINT IN PROCESS	This is a sub-state to ENDPOINT ON. Endpoint On is the status of the Sensor-BI-TH-EP Status attribute and ENDPOINT IN PROCESS is the enumerated state of the Sensor-BI-TH-EP State attribute. Endpoint object is performing the "endpoint" detection event process. The endpoint object may not be downloaded with new recipe parameters when in this state.
ENDPOINT SUSPENDED	This is a sub-state to ENDPOINT ON; Endpoint On is the status of the Sensor-BI-TH-EP Status attribute and ENDPOINT SUSPENDED is the enumerated state of the Sensor-BI-TH-EP State attribute. Endpoint object is **NOT** performing the "endpoint" detection event process. The endpoint object may not be downloaded with new recipe parameters when in this state.
ENDPOINT FAILURE	Endpoint electronics failure; Sensor-BI-TH-EP State attribute is in the enumerated state: ENDPOINT FAILURE. Endpoint is not performing the "endpoint" detection event process. Endpoint cannot move to the ENDPOINTING state until the electronics are repaired or replaced. A Perform Diagnostics or Reset service request to the Device Manager object may be sent to validate that the failure has been cleared. The endpoint object may not be downloaded with new recipe parameters when in this state.

Table 17 Sensor-BI-TH-EP Behavior EXECUTING Sub-state Transition Matrix*

#	*Current State*	*Trigger*	*New State*	*Action*	*Comment*
1	Entry into OPERATING	Power Up	Conditional: ENDPOINTING or ENDPOINT FAILURE (optional)	If the endpoint electronics are in a failure condition than State is ENDPOINT FAILURE (optional) or ENDPOINTING. If endpoint failure checking is not supported the state is ENDPOINTING / ENDPOINT OFF. Set Status attribute to appropriate value.	Entry state depends on availability of endpoint electronics checking.

#	*Current State*	*Trigger*	*New State*	*Action*	*Comment*
2	ENDPOINT FAILURE	Perform Diagnostics or Reset service request received and device determines that the failure no longer exists or the Endpoint device detects replacement of electronics that caused the failure	ENDPOINTING / ENDPOINT OFF	Set Status attribute to appropriate value.	Endpoint moves to ENDPOINT OFF state.
3	ENDPOINTING	Endpoint device detects an endpoint electronics failure	Conditional: ENDPOINT FAILURE or ENDPOINTING / EDNDPOINT OFF	Set State attribute to appropriate value.	The setting of the Status attribute is manufacturer specific.
4	ENDPOINT FAILURE	Endpoint Off, Endpoint On, Endpoint Start, Endpoint Resume, Endpoint Suspend, Download Recipe, Upload Recipe, and Calibrate request	ENDPOINT FAILURE	Error response	Object cannot move to ENDPOINTING / ENDPOINT OFF state until the electronics are repaired.
5	ENDPOINT OFF	Endpoint On request	Conditional: ENDPOINT ON / ENDPOINT IDLE or ENDPOINT ON / ENDPOINT IN PROCESS	If ENDPOINT IDLE supported: Take the object from the ENDPOINT OFF state to the ENDPOINT IDLE state and wait for an Endpoint Start request. Recipe parameter downloads are allowed. If ENDPOINT IDLE is not supported: Take the object from the ENDPOINT OFF state immediately to the ENDPOINT IN PROCESS state and begin the endpoint detection event process. Set the Status attribute to the appropriate values throughout the transition. Issue an Endpoint On response.	ENDPOINT IDLE is an optional intermediate state between ENDPOINT OFF and ENDPOINT IN PROCESS. Endpoint electronics must not be in failure. Service response is not issued until transition to ENDPOINT IDLE or ENDPOINT IN PROCESS is completed.
6	ENDPOINT ON	Endpoint Off request	ENDPOINT OFF	Stop the “endpoint” detection event process. Issue an Endpoint Off response.	This results in the same state transition as when a valid Abort service request is issued to the device.
7	ENDPOINT OFF	Endpoint Off request	ENDPOINT OFF	Error response	Endpoint is already off.

#	*Current State*	*Trigger*	*New State*	*Action*	*Comment*
8	ENDPOINT OFF	Endpoint Suspend, or Endpoint Resume request	ENDPOINT OFF	Error response	Endpoint event process remains off.
9	ENDPOINT OFF	Endpoint On or Endpoint Start request. Endpoint is unable to turn on properly.	ENDPOINT OFF	Endpoint attempts to take the object from the ENDPOINT OFF state to an ENDPOINT ON state. Endpoint won't turn on properly. Turn "endpoint" detection event process back off and generate an Error response.	Behavior associated with determining that the endpoint will not turn on properly is manufacturer specific.
10	Entry into ENDPOINT ON	None	Entry state for ENDPOINT ON state. Conditional: ENDPOINT ON / ENDPOINT IDLE or ENDPOINT ON / ENDPOINT IN PROCESS	ENDPOINT IDLE state entered if supported; otherwise ENDPOINT IN PROCESS state entered	Entry depends on endpoint electronics functioning properly.
11	ENDPOINT OFF	Endpoint Start request	ENDPOINT ON / ENDPOINT IN PROCESS	Go immediately to the ENDPOINT ON / ENDPOINT IN PROCESS state and begin the endpoint detection event process. Issue an Endpoint Start response	Behavior associated with determining that the endpoint will not turn on properly is manufacturer specific.
12	ENDPOINT IN PROCESS	Endpoint On, Endpoint Start, Endpoint Resume, Download Recipe, Upload Recipe	ENDPOINT IN PROCESS	Error response	Endpoint is already in process.
13	ENDPOINT IN PROCESS	Endpoint detection completed and idle state supported	ENDPOINT IDLE	Endpoint event detected. Endpoint event reported.	
14	ENDPOINT IDLE	Endpoint Start Request	ENDPOINT IN PROCESS	Endpoint event detection process is started. Issue an Endpoint Started success response.	
15	ENDPOINT IDLE	Download Recipe, and Upload Recipe Request	ENDPOINT IDLE	Valid recipes are Downloaded or Uploaded and a services response is generated.	
16	ENDPOINT IDLE	Endpoint Suspend and Endpoint Resume request	ENDPOINT IDLE	Error response	Endpoint is not in process so it can not be suspended.

#	*Current State*	*Trigger*	*New State*	*Action*	*Comment*
17	ENDPOINT IN PROCESS	Endpoint detection failed	Conditional: ENDPOINTING ON / ENDPOINT IDLE or ENDPOINTING / ENDPOINT OFF	Endpoint event not detected within the allotted endpoint active timers. Endpoint event failure reported.	
18	ENDPOINT IN PROCESS	Endpoint Suspend request and service supported	ENDPOINT SUSPENDED	Take the object from the ENDPOINT IN PROCESS state to ENDPOINT SUSPENDED state, turning off the endpoint event process. All active endpoint timers are held at their current values. Set the Status attribute to the appropriate value. Issue an Endpoint Suspended response and stop the "endpoint" detection process.	Endpoint event process is temporally suspended. All timers are suspended.
19	ENDPOINT SUSPENDED	Endpoint Resume request	ENDPOINT IN PROCESS	Take the object from ENDPOINT SUSPENDED state to ENDPOINT IN PROCESS. Begin the "endpoint" detection event process form the point it was suspended. All active endpoint timers are reinstated to their last held values. Issue an Endpoint Resume response.	Endpoint event process is resumed from the point it was suspended.
20	ENDPOINT SUSPENDED	Endpoint On, Endpoint Start, Endpoint Suspend , Download Recipe, Upload Recipe request	ENDPOINT SUSPENDED	Error response	Endpoint event process remains suspended.
21	ENDPOINT ON	Endpoint On, Endpoint Start, Endpoint Resume, Download Recipe, and Upload Recipe request	ENDPOINT ON	Error response	Endpoint detection event process remains on and continues endpoint process from its current attribute settings.
22	ENDPOINT OFF	Download Recipe or Upload Recipe request	ENDPOINT OFF	Valid recipes are Downloaded or Uploaded and a services response is generated.	

8.6 *Assembly-EP Objects* — Assembly-EP objects inherit attributes, services, and behavior from the Assembly object. The Assembly object is the device component that provides a mechanism of grouping more than one attribute from one or more objects into a single data structure for communication over the network.

Table 18 identifies the Assembly-EP objects defined for the EPD device.

Table 18 Assembly List

Object	*Access Network*	*Required*	*Form*
Assembly-EPD#1	R	Yes	Status; Default Assembly
Assembly-EPD#2	R	No	Value, Reading Valid
Assembly-EPD#3	R	No	Status, Number of Endpoint Objects and Value (Values ≤ 1024)
Assembly-EPD#4	R	No	Status, Exception Detail Alarm and Exception Detail Warning

8.6.1 *Assembly-EP Objects Attributes* — Table 19 provides a list of attributes common to all Assembly-EP object types.

Table 19 Assembly-EP Object Attributes

Attribute Name	*Attribute Identifier*	*Access Network*	*Required*	*Form*
Data (See NOTE 1.)	A1	RW	Yes	Structure as defined below.

NOTE 1: Inherited from the Assembly object as shown in Figure 1.

8.6.1.1 *Data Attribute Format for Assembly-EP Objects* — The Data attribute of all Assembly-EP objects is a structured attribute containing an ordered list of attributes within its structure. In the following tables (20 through 23), the structure of the Data attribute for each of the Assembly-EP object types is defined.

Table 20 Assembly-EPD #1 Object Data List

Data Index	*Source Object ID*	*Source Attribute ID*	*Description*
1	EPD1	A12	Device Manager Exception Status

Table 21 Assembly-EPD #2 Object Data List

Data Index	*Source Object ID*	*Source Attribute ID*	*Description*
1	EPD3	SbithepA16	Sensor-BI-TH-EP #1 Value Reading Valid

Table 22 Assembly-EPD #3 Object Data List

Data Index	*Source Object ID*	*Source Attribute ID*	*Description*
1	EPD1	A12	Device Manager Exception Status
2	EPD2	SacA65	SAC Number of Endpoint Objects
3	EPD3	SbithepA16	Sensor-BI-TH-EP #1 Value
4	EPD3	SbithepA16	Sensor-BI-TH-EP #2 Value
.	EPD3	SbithepA16	Sensor-BI-YH-EP #3 to Sensor-BI-TH-EP #N-1 Value
N ≤ 1024 (See NOTE 1.)	EPD3	SbithepA16	Sensor-BI-TH-EP #N Value

NOTE 1: 'N' is the value of the Sensor-BI-EP object attribute "Number of Endpointing Objects". Note also that the number of endpoints reported in an assembly is fixed throughout the life of the device, and is specified by the manufacturer.

Table 23 Assembly-EPD #4 Object Data List

Data Index	*Source Object ID*	*Source Attribute ID*	*Description*
1	EPD1	A12	Device Manager Exception Status
2	EPD1	A13	Device Manager Exception Detail Alarm
3	EPD1	A14	Device Manager Exception Detail Warning

SEMI E54.12-0701[E]
SPECIFICATION FOR SENSOR/ACTUATOR NETWORK COMMUNICATIONS FOR CC-LINK

This specification was technically approved by the Global Information and Control Committee and is the direct responsibility of the North American Global Information and Control Committee. Current edition approved by the North American Information and Control Committee on April 30, 2001. Initially available at www.semi.org May 2001; to be published July 2001.

[E] This standard was editorially modified in September 2001 to correct a typographical error and the omission of a required disclaimer. Section 2.3 was added and changes were made to Section 7.6.2.6.

1 Purpose

1.1 This specification is part of the SEMI Sensor/Actuator Network (SAN) suite of standards and defines a specific communications protocol based on the CC-Link standard. This Network Communication Standard (NCS) taken together with the SEMI Sensor/Actuator Network standard suite and the CC-Link standard completely and unambiguously defines an open standard providing an industry specific solution to off-the-shelf interoperability of networked devices in semiconductor manufacturing equipment.

1.2 CC-Link is a vendor independent, open device level network standard. Vendor independence and openness are guaranteed by the CC-Link Partner Association.

2 Scope

2.1 This document specifies a SAN communications standard based on the CC-Link specification that is in compliance with SEMI E54.1. As such, it specifies the protocol, services and behavior that compliant intelligent devices must support in order to interchange information over this SAN in a method compatible with SEMI E39.

2.2 In conjunction with a SEMI standard SAN Common Device Model (CDM) specification and one or more SEMI standard Specific Device Model (SDM) specifications (e.g., for a mass flow controller), this Network Communication Standard (NCS) with the related CC-Link standard describe the data structures, interactions and behavior that are characteristic of the various devices on the network. This composite model forms a complete interoperability standard for communications among intelligent sensors, actuators and controllers in semiconductor manufacturing equipment.

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

3 Limitations

3.1 This document specifies a semiconductor equipment SAN based solely on CC-Link; thus, a complete specification of this standard necessarily includes the CC-Link specifications. There are other semiconductor equipment SAN communications options. The specifications for these options are not included here.

3.2 The specifications within are strictly enhancements that provide additional capabilities over and above those currently required by CC-Link. Included throughout this document, primarily in Section 6, is information paraphrased from the CC-Link specifications — such as: protocol structure, capabilities, options and limitations. This information is provided here for reference only and is not intended to provide specification definitions. In all such areas, refer to the CC-Link specification documents for information. This document is limited to describing enhancements or limitations to the CC-Link specification that are imposed by this standard.

3.3 A complete specification of the conformance testing procedure shall include the CC-Link protocol conformance testing specification. Conformance testing shall also include enhancements and limitations to the CC-Link specification required by this standard.

4 Referenced Standards

NOTE 1: Unless otherwise indicated, all documents cited shall be the latest published versions.

4.1 *SEMI Standards*

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

SEMI E54.1 — Sensor/Actuator Network Common Device Model

SEMI E54.3 — Specification for Sensor/Actuator Network Specific Device Model for Mass Flow Device

4.2 *ISO[1] Standards*

7498 OSI — Basic Reference Model for Open Systems Interconnection

4.3 *CC-Link Partner Association[2]*

CC-Link Specification, Version 1.11 (or later)

5 Terminology

5.1 *Abbreviations and Acronyms*

5.1.1 *CDM* — Common Device Model

5.1.2 *NCS* — Network Communication Standard

5.1.3 *OSI* — Basic Reference Model for Open Systems Interconnection (ISO 7498)

5.1.4 *SAN* — Sensor/Actuator Network

5.1.5 *SDM* — Specific Device Model

5.2 *Definitions from Sensor/Actuator Network Common Device Model (SEMI E54.1)*

5.2.1 *Attribute*

5.2.2 *Behavior*

5.2.3 *Byte*

5.2.4 *Common Device Model*

5.2.5 *Device*

5.2.6 *Device Manager (DM) Object*

5.2.7 *Device Model*

5.2.8 *Instance*

5.2.9 *Network Communication Standard*

5.2.10 *Object*

5.2.11 *Sensor, Actuator and Controller (SAC) Object*

5.2.12 *Service*

5.2.13 *Specific Device Model*

5.2.14 *State Diagram*

5.3 *Definitions*

5.3.1 *broadcast polling method* — polling to each station and the data communication are executed by the same packet, and the data is transmitted to all of the stations in this method.

5.3.2 *cyclic transmission* — function to transmit the data from master station to all stations periodically, then for each station to transmit the response data to master station.

5.3.3 *intelligent device station* — station which can send cyclic transmission and transient transmission to master station.

5.3.4 *local station* — station which can send cyclic transmission and transient transmission to master station and other local stations.

5.3.5 *master station* — station that controls all stations on CC-Link. One (and only one) master station per system is required.

5.3.6 *profiles* — Application Object Model specifications.

5.3.7 *remote device station* — station that handles bit data and word data.

5.3.8 *remote I/O station* — station that handles only bit data.

5.3.9 *remote station* — generic name of remote I/O station and remote device station.

5.3.10 *slave station* — generic name of station other than master station.

5.3.11 *station* — equipment which can be connected with CC-Link and is assigned a station number of 0–64.

5.3.12 *transient transmission* — function to transmit the non-periodic data generated in master station, local station, and intelligent device station.

6 Communication Protocol High Level Structure

6.1 Message transfer is organized in cycles. A message cycle mainly consists of a request-frame followed by a corresponding acknowledge/response-frame of the addressed station.

6.2 A brief description of the CC-Link protocol as it relates to the ISO 7498 OSI model follows in the sections below. For protocol efficiency, CC-Link does not define layers 3 to 6.

NOTE 2: The information contained in this section is for reference only. It in no way represents specifications for CC-Link. See related documentation for these specifications.

6.3 *Physical Layer - Layer 1* — the Physical Layer conforms to the EIA RS-485 standard. See the CC-Link standard for more information.

6.4 *Data Link Layer - Layer 2* — the Data Link Layer conforms to the HDLC standard. See the CC-Link standard for more information.

6.5 *Application Layer - Layer 7* — the Application Layer defines services and protocols for Network

1 ISO - International Organization for Standardization, 1 rue de Varembe, Case postale 56, CH-1211, Geneva 20, Switzerland

2 CC-Link Partner Association - <www.cc-link.org>

Management, Cyclic Transmission and Non-Cyclic Transmission (or Transient Transmission). Also, Application Object Models are specified as "Profiles". See the CC-Link standard for more information.

6.6 Version 1.11 of the CC-Link Standard introduces two methods for request/response messaging.

6.7 Devices that support the CC-Link Transient Transmission capability, use this method, together with a messaging protocol to transmit Service Requests and Responses.

6.8 Devices that do not support the Transient Transmission capability, use the Cyclic Transmission method. A protocol is defined for changing the context of the cyclic data from I/O to a messaging protocol for Service Requests and Responses.

6.9 See the CC-Link standard for more information.

7 Required Object Types

7.1 This section describes a general mapping of the SEMI SAN Object Model to the CC-Link environment. Component definitions are clarified and the mapping of Attributes, Services and Behaviors are specified.

7.2 *Object Model* — The Object Model defined in the CDM is represented in the CC-Link NCS. Specifically, the DM and SAC objects are mapped.

7.2.1 Section 9 specifies the mapping of SDM Objects in CC-Link.

7.3 *Objects* — The required objects of the CDM are identified here. Additional objects that are contained in the SDM are given identifiers in the Device Profile. Section 9 specifies additional mapping information.

7.3.1 Table 1 lists the Object Identifiers specified for use in protocol messages.

Table 1 Object Identifiers

Object ID	*Object*
0	Invalid
1	DM Object
2	SAC Object
3–n	Application Objects as specified in Section 9

7.4 *Attributes* — All attributes are accessible via Get_Attribute and Set_Attribute services defined in the sections below.

7.4.1 *Attribute Identifiers* — Every object specified in the CDM and SDMs uses tags to identify its attributes. These tags are formatted with letters (identifying the object) followed by an upper case "A", followed by a numerical identifier. The Attribute ID used in the CC-Link NCS is simply the numerical portion of these tags.

7.5 See Table 2 for a list of DM attributes.

Table 2 DM Object Attribute Identifiers

SEMI CDM Attribute ID	*CC-Link Attribute ID*	*Attribute*
DmA1	1	Device Type
DmA2	2	Standard Revision Level
DmA3	3	Device Manufacturer Identifier
DmA4	4	Manufacturer Model Number
DmA5	5	Software or Firmware Revision Level
DmA6	6	Hardware Revision Level
DmA7	7	Serial Number
DmA8	8	Device Configuration
DmA9	9	Device Status
DmA12	12	Exception Status
DmA13	13	Exception Detail Alarm
DmA14	14	Exception Detail Warning
DmA15	15	Visual Indicator
DmA16	16	Alarm Enable
DmA17	17	Warning Enable

7.6 *Services*

7.6.1 *Service Identifiers* — The required services of the CDM are identified here. Additional services that are contained in the SDM are given identifiers in the Device Profile. Table 3 specifies the required services and ID numbers.

Table 3 Service Identifiers

Service ID	*Service*
0	Invalid
1	Reset
2	Abort
3	Recover
4	Get Attribute
5	Set Attribute
6	Execute
7	Perform Diagnostics

7.6.2 *Specified Services* — The following sections define the details associated with each of the services required by the CDM.

7.6.2.1 *Reset* — The Reset Request specifies no parameters. In addition to an explicit Reset Service Request, CC-Link specifies others methods whereby a Slave device can be reset.

7.6.2.2 *Abort* — The Abort Service Request specifies no parameters.

7.6.2.3 *Recover* — The Recover Service Request specifies no parameters.

7.6.2.4 *Get Attribute* — The Get Attribute Request specifies two parameters: the Object ID and the Attribute ID. Each are currently defined in the range 1–255. Both are expandable to 65,535.

7.6.2.5 *Set Attribute* — The Set Attribute Request specifies three parameters: the Object ID, the Attribute ID and the Attribute Value to set. The Object ID and the Attribute ID are defined the same as for the Set Attribute Service. The length of the Attribute Value is based on the specification of the attribute.

7.6.2.6 *Execute* — The Execute Service Request specifies no parameters.

7.6.2.7 *Perform Diagnostics* — The Perform Diagnostic Request specifies one parameter: Test ID. The Test ID parameter is one byte in length.

8 Protocol Compliance

8.1 The CC-Link Partner Association has established a qualified certification system, with test laboratories in Japan that include conformance testing and interoperability testing.

9 Specific Device Model Mappings

9.1 The following sections specify mappings for Sensor Actuator Network Specific Device Models.

9.2 *Mass Flow Device* — Reference SEMI E54.3 for a complete specification of the SDM for Mass Flow Devices. Accordingly, the following mapping rules apply to the identification tags for the Objects, Attributes and Services of this model.

9.2.1 *Objects* — Consistent with SEMI E54.3 and Section 7.3 above, the DM and SAC objects are identified as Object 1 and Object 2, respectively.

9.2.1.1 Table 4 shows the mapping of the SDM Object Instances specified in SEMI E54.3 (Instance numbers are listed under heading Inst. in the table) and the CC-Link Object ID (listed under ID in the table).

Table 4 MFD Object Identifiers

SDM Object Name	*SDM Object ID*	*Inst.*	*ID*
Sensor-AI-MF	MFD3	1	3
Actuator-AO-MF	MFD7	1	4
Controller	MFD8	1	5
SISO-Setpoint	MFD11	1	6
Sensor-AI-AT	MFD4	1	7

9.2.1.2 Additional objects may be defined by the manufacturer in the Device Profile for a given device.

9.2.2 *Attributes* — The mapping of Attribute Tags and Identifiers is defined in Section 7.4.1 for the CDM. The same method applies here for the SDM.

9.2.3 *Services* — The mapping of Service Tags and Identifiers is defined in Section 7.6.1 for the CDM. The same method applies here for the SDM.

9.3 *In-Situ Particle Monitor* — Reference SEMI E54.10 for a complete specification of the SDM for In-Situ Particle Monitor Devices. Accordingly, the following mapping rules apply to the identification tags for the Objects, Attributes and Services of this model.

9.3.1 *Objects* — Consistent with SEMI E54.10 and Section 7.3 above, the DM and SAC objects are identified as Object 1 and Object 2, respectively.

9.3.1.1 Table 5 shows the mapping of the SDM Object Instances specified in SEMI E54.3 (Instance numbers are listed under heading Inst. in the table) and the CC-Link Object ID (listed under ID in the table).

Table 5 ISPM Object Identifiers

SDM Object Name	*SDM Object ID*	*Inst.*	*ID*
Sensor-AI-LCS	ISPMD3	1	3
Sensor-AI-SLS	ISPMD4	1	4
Sensor-AI-MNS	ISPMD5	1	5
Assembly-ISPM#1	ISPMD17	1	6
Assembly-ISPM#2	ISPMD18	1	7
Assembly-ISPM#3	ISPMD19	1	8
Assembly-ISPM#4	ISPMD20	1	9
Assembly-ISPM#5	ISPMD21	1	10
Assembly-ISPM#6	ISPMD22	1	11
Assembly-ISPM#7	ISPMD23	1	12
Assembly-ISPM#8	ISPMD24	1	13
Assembly-ISPM#9	ISPMD25	1	14
Assembly-ISPM#40	ISPMD64	1	15
Sensor-AI-Counter	ISPMD16	1	16
Sensor-AI-Counter	ISPMD16	n	15 + n
Sensor-AI-Counter	ISPMD16	1024	1039

9.3.1.2 Additional objects may be defined by the manufacturer in the Device Profile for a given device.

9.3.2 *Attributes* — The mapping of Attribute Tags and Identifiers is defined in Section 7.4.1 for the CDM. The same method applies here for the SDM.

9.3.3 *Services* — The mapping of Service Tags and Identifiers is defined in Section 7.6.1 for the CDM. The same method applies here for the SDM.

NOTICE: SEMI makes no warranties or representations as to the suitability of the standards set forth herein for any particular application. The determination of the suitability of the standard is solely the responsibility of the user. Users are cautioned to refer to manufacturer's instructions, product labels, product data sheets, and other relevant literature respecting any materials mentioned herein. These standards are subject to change without notice.

The user's attention is called to the possibility that compliance with this standard may require use of copyrighted material or of an invention covered by patent rights. Mitsubishi has filed a statement with SEMI asserting that licenses will be made available to applicants throughout the world for the purpose of implementing this standard without unfair discrimination. Attention is also drawn to the possibility that some elements of this standard may be subject to patented technology or copyrighted items other than those identified above. Semiconductor Equipment and Materials International (SEMI) shall not be held responsible for identifying any or all such patented technology or copyrighted items. By publication of this standard, SEMI takes no position respecting the validity of any patent rights or copyrights asserted in connection with any item mentioned in this standard. Users of this standard are expressly advised that determination of any such patent rights or copyrights and the risk of infringement of such rights are entirely their own responsibility.

SEMI E54.13-0303
SPECIFICATION FOR SENSOR/ACTUATOR NETWORK COMMUNICATIONS FOR ETHERNET/IP™

This specification was technically approved by the Global Information and Control Committee and is the direct responsibility of the North American Information and Control Committee. Current edition approved by the North American Regional Standards Committee on November 22, 2002. Initially available at www.semi.org January 2003; to be published March 2003.

1 Purpose

1.1 *Introduction*

1.1.1 This standard defines a communication specification based on the EtherNet/IP[1] (Ethernet/Industrial Protocol) network to enable communications between intelligent devices on a sensor/actuator network (SAN) that operate according to SEMI specified device models (common and device specific) in a semiconductor manufacturing tool.

1.2 *Motivation*

1.2.1 EtherNet/IP is a communication system suitable for use in industrial environments. EtherNet/IP allows intelligent devices to exchange time-critical application information. These devices include simple I/O devices such as sensors/actuators, as well as complex control devices such as robots, programmable logic controllers, and process controllers.

1.2.2 EtherNet/IP uses CIP (Control and Information Protocol), the common network, transport and application layers also shared by DeviceNet (SEMI E54.4). EtherNet/IP provides:

- A cost effective solution to bridge factory Ethernet (IEEE 802.3) networks to SEMI E54.4 DeviceNet low-level device networks,
- Access to intelligence present in low-level devices, and
- Producer/Consumer model for Master/Slave and Peer-to-Peer application relationships.

1.3 *Background*

1.3.1 EtherNet/IP makes use of standard Ethernet and TCP/IP technology to transport CIP communications packets. The result is a common, open application layer on top of open and highly popular Ethernet and TCP/IP protocols.

1.3.2 EtherNet/IP provides a producer/consumer model for the exchange of time-critical control data. The producer/consumer model allows the exchange of application information between a sending device (e.g., the producer) and many receiving devices (e.g., the consumers) without the need to send the data multiple times to multiple destinations. For EtherNet/IP, this is accomplished by making use of the CIP network and transport layers along with IP Multicast technology. Many EtherNet/IP devices can receive the same produced piece of application information from a single producing device.

1.3.3 EtherNet/IP makes use of standard IEEE 802.3 technology; there are no non-standard additions that attempt to improve determinism. Rather, EtherNet/IP recommends the use of commercial switch technology, with 100 Mbps bandwidth and full-duplex operation, to provide for more deterministic performance.

2 Scope

2.1 *Specification* — This document specifies a Sensor/Actuator Network Communications Standard (NCS) based on the EtherNet/IP specification that enables communication with SAN devices configured according to SEMI SAN Common Device Model (CDM) and appropriate Specific Device Model (SDM) specifications.

2.2 *Use* — This document is used in conjunction with a SEMI standard SAN CDM specification and one or more SEMI standard SDM specifications (e.g. for a mass flow controller). Together, they describe the externally visible data structure and behavior of devices utilizing the EtherNet/IP networking capability in a SEMI compliant SAN system. The general sensor/actuator network document architecture is described in the SEMI E54.0 Sensor/Actuator Network Standard (the root SAN document).

2.3 *Document Structure* — The EtherNet/IP network communication standard complies with the SEMI SAN NCS template document structure, as described in SEMI E54.0. The standard document is composed of two main parts. The first part (Sections 1 through 8) specifies the SAN enabling protocol as well as the presentation (i.e., mapping) of CDM object structure and behavior onto the network (referred to as the "CDM mapping"). The second part (Section 9) specifies the presentation (i.e., mapping) of SDM object

1 EtherNet/IP is a trademark of Open DeviceNet Vendor Association (ODVA)