

# EMD

**(Electronic Module Description Specification)**

**Version 1.0 Draft 123**  
**February 27, 2013**

Ratified TBD

Why EMD?

EMD is a replacement for Electronic Board Description (EBD).

The major advantages of EMD over EBD are:

1. Not limited to Board description (e.g. not limited to SIMM and DIMM)
  - a. Allow connections between external (visible) pins
  - b. Allow connections between internal components
  - c. Allows description of Multi Chip Modules (MCM)
  - d. Allows description of Interposers
  - e. Allows description of Connectors
  - f. Allows description of Cables
2. Replaces Path Descriptions with
  - a. IBIS ISS subckts
  - b. Touchstone Files
3. Supports broadband, lossy, distributed interconnect
4. Support for Interconnect Corners (Typ, Slow, Fast)
5. Support for parameterized subckts
6. Support for true differential interconnect
7. Support for coupled (cross talk) models
8. Support for power distribution models (PDM)
9. Support for coupling between power and signal interconnect
10. Support for pullup and pulldown termination in module
11. Parameter Tree format allows easier implementation of enhancements
12. Eliminates EBD Path description
  - a. No more Forks!!!!

The major disadvantages of EMD over EBD

NONE. An EBD file can be programmatically converted to an EMD file with absolutely no loss of functionality.

We need to decide if we will allow an EMD file to contain an EBD component.

# 1 ELECTRICAL MODULE DESCRIPTION

## INTRODUCTION:

A “module level component” is the generic term to be used to describe an electronic module which can contain components, modules, and which can connect to another module through a set of user visible pins. The electrical connectivity of such a module level component is referred to as an “Electrical Module Description”. An electrical module description file (an .emd file) is defined to describe the connections of a module level component between the module pins and its module pins, components and other modules. A module can be a package with a single component, a package with multiple components, a board with zero, one, or multiple components, a board with zero, one, or multiple components, an interposer, a connector, and a cable.

A fundamental assumption regarding the electrical module description is that the interconnect between the module pins, components, and modules can be represented by IBIS-ISS subckts and Touchstone files directly. Also, this current description does allow one to describe electrical coupling between connections.

A component is represented by a .ibs file which may represent either a die and package, or a bare die.

A connection is represented by a list of its visible module pins, component pins and module pins that have a small insertion loss at Nyquist between all of the pins.

What is, and is not, included in an Electrical Module Description is defined by its boundaries. For the definition of the boundaries, see the Description section under the Interconnect Branch.

### Usage Rules:

A .emd file is intended to be a stand-alone file, not referenced by or included in any .ibs or .pkg file. Electrical Module Descriptions are stored in a file whose name looks like <filename>.emd, where <filename> must conform to the naming rules given in Section [Error! Reference source not found.](#)<sup>3</sup> of this specification. The .emd extension is mandatory.

The Parameter Tree shall contain a Root, Branches and Leaves. The Root, Branch and Leaf names shall be “token strings” that may contain the characters “ “, “(“, or “)”. They may not contain the tab or “ character. If the token string contains “ “, “(“, or “)”, the token must be surrounded by the “s. A Leaf may contain one or more values. Values shall be token strings. The rules for values token strings shall be the same as the rules for Root, Branch and Leaf token strings (may be contextual specific).

### Contents:

A .emd file is structured using a Parameter Tree Structure. It must contain a descriptive Root, and the following Branches listed below:

IBIS Version 5.1

- General\_Information
- Module\_Pin\_List
- Modules\_and\_Components
- Connections
- Interconnect

The General\_Information Branch must contain the following Leaves:

- EMD\_Version
- File\_Name
- File\_Rev

The General\_Information Branch may also contain the following optional Leaves:

- Manufacturer
- Date
- Source
- Notes
- Disclaimer
- Copyright

The actual module description is contained in the following Branches:

- Module\_Pin\_List
- Modules\_and\_Components
- Connections
- Interconnect

|

**ROOT DEFINITION**

*Root:*            **<Root Name>**

*Required:*        Yes

*Description:*    Marks the beginning of an Electrical Module Description.

*Usage Rules:*    The root name of the module level component.

*Example:*

("16Meg X 8 SIMM Module")

|

### BRANCH DEFINITIONS:

**Branch:** General\_Information

**Required:** Yes

**Description:** Branch contains Leaves containing requires and optional Leaves

**Usage Rules:** This Branch contains the required leaves EMD\_Version, File\_Name, File\_Rev and may contain the optional Leaves Manufacturer, Date, Source, Notes, Disclaimer, Copyright

**Example:**

```
(16Meg_X_8_SIMM_Module  
  (General_DescriptionInformation
```

### GENERAL INFORMATION LEAF DEFINITIONS:

**Leaf:** EMD\_Version

**Required:** Yes

**Description:** Declares the version .emd file.

**Usage Rules:** Following the leaf is the emd version.

**Example:**

```
(16Meg_X_8_SIMM_Module  
  (General_InformationDescription  
    (EMD_Version 1.0)
```

**Leaf:** File\_Name

...

**Leaf:** File\_Rev

...

**Leaf:** Manufacturer

**Required:** No

**Description:** Declares the manufacturer of the components(s) that use this .emd file.

**Usage Rules:** Following the leaf is the manufacturer's name. It must not exceed 40 characters, and can include blank characters. Each manufacturer must use a consistent name in all .emd files.

**Example:**

```
("16Meg X 8 SIMM Module"  
  (General_InformationDescription  
    (Manufacturer "Quality SIMM Corp.")
```

**Leaf:** Date

**Leaf:** Source

*Leaf:*        **Notes**  
*Leaf:*        **Disclaimer**  
*Leaf:*        **Copyright**

|

*Branch:*        **Module\_Pin\_List**

*Required:*     Yes

*Description:*  Branch contains Leaves containing module visible pins.

*Usage Rules:* This Branch contains leaves for each module visible pin. Each visible pin leaf shall consist of a pin\_name and a connection\_name.

*Example:*

```
(16Meg_X_8_SIMM_Module
  (Module_Pin_List
    (A4 PWR1)
    (A5 PWR1)
    (A1 DQ1)
    (A2 DQ2)
    (A3 DQ3) | Connector pin_names might be A.A3, Bside.A3
              | Will we need a list of Connectors?
  ) | Suopply pins and connection are included (pullup)
```



**Branch:** Modules\_and\_Components

**Required:** Yes

**Description:** Branch contains Leaves containing modules and components in the module.

**Usage Rules:** This Branch contains leaves for each module and component in the module. Each module and component leaf shall consist of a reference designator. If the leaf is a component its values are an IBIS file name and component. If the leaf is a module its value is the EMD file name.

**Example:**

```
(16Meg_X_8_SIMM_Module
  (Modules_and_Components
    (u23      pp100.ibs  Processor)
    (u24      simm.emd  ?) _|.ebd?
    (u25      ls244.ibs  "NoName 74LS244a")
    (r26      r10K.ibs   My_10K_Pullup)
  )
)
```

*Branch:*       **Connections**

*Required:*    Yes

*Description:*  Branch contains Leaves containing connections.

*Usage Rules:* This Branch contains leaves for each connection. Each connection shall consist of a connection\_name followed by the visible pins and component and modules pins that are connected.

(Connections are also known as Extended Nets in some EDA tools.)

A Pin can occur in one and only one connection.

An unconnected pin would not be in any connection. Or may be in a connection by itself.

*Example:*

```
(16Meg_X_8_SIMM_Module
  (Connections
    (DQ1 A1 U1.17 U2.17 U3.17)
    (DQ2 A2 U1.18 U2.18 U3.18)
    (DQ3 A3 U1.19 U2.19 U3.19)
    (DQx U1.20 U2.20 U3.20)
    (PWR1 A4 A5 U1.30 U2.30 U3.30 U3.31)
    (UN1 A16)
    |_____(F12 A.F12 B.F12)   Connector connection
  )
)
```

**Branch:**       **Interconnect**

**Required:**    Yes

**Description:** Branch contains Branches containing Model ~~Connection~~-Interconnect Protocols for IBIS-ISS subckts and Touchstone files.

**Usage Rules:** This Branch contains branches containing Model Interconnect ~~Connection~~ Protocols (~~MCP~~MIP) for IBIS-ISS subckts and Touchstone files.. Each MIP ~~MCP~~ shall contain leaves (or Branches) defining the IBIS-ISS file and subckt name or Touchstone file. Each MIP shall contain and a Branch defining the ports of the IBIS-ISS subckt or Touchstone file.

An IBIS-ISS Interconnect may also contain a Parameter Branch, which shall contain Branches or Leaves for each Parameter that can be passed as parameters on the IBIS-ISS instance. If the parameter is a leaf, then it shall contains a single value that is a legal IBIS-ISS parameter value. If the parameter is a Branch, it may contain a single Corner leaf with Typ, Min, and Max values.

Note that an Interconnect might be for a single connection, a differential connections, a group of coupled signal connections, a supply connection, a group of supply connections, a group of supply and signal connections. A connection may appear in one or more Interconnects.

All pins are not necessarily in an Interconnect.

In a Interconnect has a connection, all pins in the connection must be included (does this make sense for supply?)

**Example:**

```
(16Meg_X_8_SIMM_Module
  (Interconnect
    (DQ1
      (IBIS_ISS_File xyz.iss-dq1)
      (IBIS_ISS_Circuit (Corner dq1_typ dq1_slow dq1_fast)
        (Parameters
          (Impedance 50ohms)
          (Delay      (Corner 30ps 40ps 20ps)
            (Tstonefile 'xyz_dqx.s2p')
          )
        )
      (Ports
        (1 A1)
        (2 U1.17)
        (3 U2.17)
        (4 U3.17) | Ports number not monotonic
      )
    )
  )
  (DQ2
    (Tstonefile (Corner DQ2_Typ.s4p DQ2_Slow.s4p
      DQ2_Fast.s4p))
    (Ports
```

```
        (1 A2)
        (2 U1.18)
        (3 U2.18)
        (4 U3.18)
    )
)
(DQ1_DQ2
  (Tstonefile DQ1_DQ2.s128p)
  | Radek: Unused port terminations
  (Ports
    (21 A1)
    (22 U1.17)
    (23 U2.17)
    (24 U3.17)
    (15 A2)
    (16 U1.18)
    (17 U2.18)
    (18 U3.18)
  )
)
)
```