



TECHNICAL REPORT	EIA-EDIF-1™	
	Issued 1988-09 Stabilized 2015-07	
Introduction to EDIF, Volume 1		

NOTICE

This document has been taken directly from the original TechAmerica document and contains only minor editorial and format changes required to bring it into conformance with the publishing requirements of SAE Technical Standards. The release of this document is intended to replace the original with the SAE International document. Any numbers established by the original document remain unchanged.

The original document was adopted as an SAE publication under the provisions of the SAE Technical Standards Board (TSB) Rules and Regulations (TSB 001) pertaining to accelerated adoption of specifications and standards. TSB rules provide for (a) the publication of portions of unrevised specifications and standards without consensus voting at the SAE committee level, and (b) the use of the existing specification or standard format.

SAE Technical Standards Board Rules provide that: "This report is published by SAE to advance the state of technical and engineering sciences. The use of this report is entirely voluntary, and its applicability and suitability for any particular use, including any patent infringement arising therefrom, is the sole responsibility of the user."

SAE reviews each technical report at least every five years at which time it may be revised, reaffirmed, stabilized, or cancelled. SAE invites your written comments and suggestions.

Copyright © 2016 SAE International

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of SAE.

TO PLACE A DOCUMENT ORDER: Tel: 877-606-7323 (inside USA and Canada)
Tel: +1 724-776-4970 (outside USA)
Fax: 724-776-0790
Email: CustomerService@sae.org
<http://www.sae.org>

SAE WEB ADDRESS:

Full document details: [SAE EIAEDIF1](#)

SAE values your input. To provide feedback on this Technical Report, please visit <http://www.sae.org/technical/standards/EIAEDIF1>

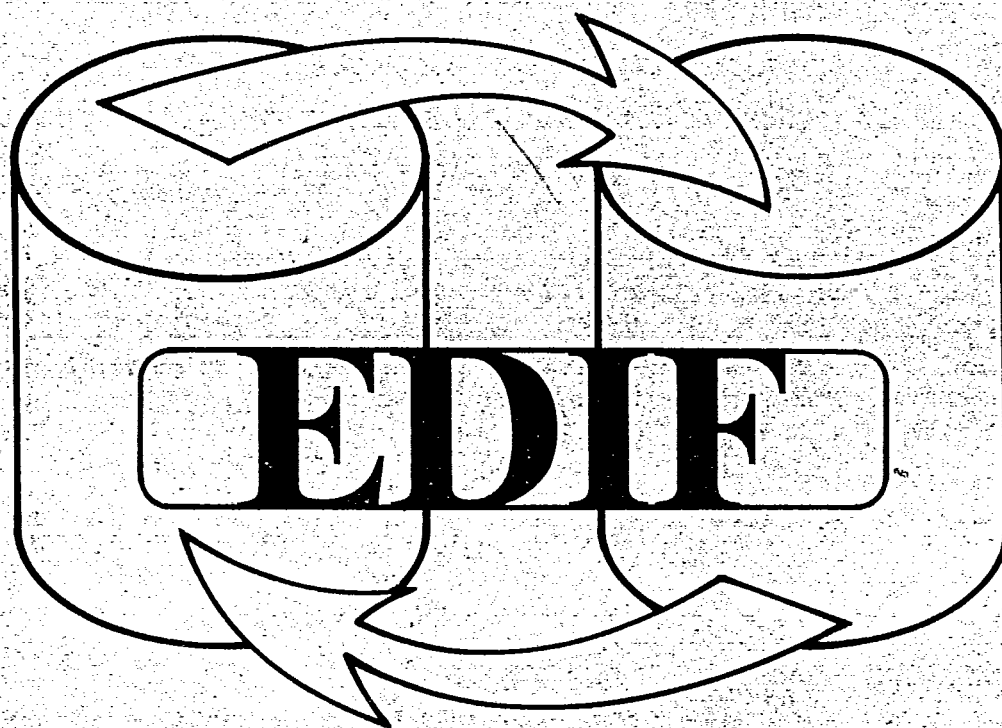
Documents

Document Sharing Hub

*Electronic Design
Interchange Format*

EIA/EDIF-1
SEPT. 1988

Introduction to EDIF



EDIF Monograph Series

Volume 1



*Electronic Industries Association
EDIF Steering Committee*

EDIF Series

Volume 1

Introduction to EDIF

1988

Electronic Industries Association

EDIF Steering Committee



Documents

Document Sharing Hub

NOTICE

EIA Engineering Standards and Publications are designed to serve the public interest through eliminating misunderstandings between manufacturers and purchasers, facilitating interchangeability and improvement of products, and assisting the purchaser in selecting and obtaining with minimum delay the proper product for his particular need. Existence of such Standards and Publications shall not in any respect preclude any member or non-member of EIA Publications, nor shall the existence of such Standards and Publications preclude their voluntary use by those other than EIA members, whether the standard is to be used either domestically or internationally.

Recommended Standards and Publications are adopted by EIA without regard to whether or not their adoption may involve patents on articles, materials, or processes. By such action, EIA does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the Recommended Standard or Publication.

The Electronic Industries Association (EIA) and the EDIF Committee do not possess full access to the data concerning the uses and applications for which someone may use the information contained within this document. Therefore, no responsibility is assumed by the EIA or EDIF Committee for any application or design, nor for any infringements of patents or rights of others which may result from the use of this document. A user of the EDIF assumes the risk that the EDIF may be unsuitable for the user's purposes. The EIA and EDIF Committee reserve the right to change or amend this document at any time.



Documents
Document Sharing Hub

Introduction to EDIF

Published by
ELECTRONIC INDUSTRIES ASSOCIATION
Engineering Department
2001 Eye Street, N.W.
Washington, D.C. 20006
Copyright © 1988. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the Electronic Industries Association.

Consulting Editor: Paul Stanford
Technical Editor: Paul Mancuso

Special thanks are due to Hilary Kahn for providing some of the material on which this monograph is based, as part of the Alvey CAD031 Project.

ISBN 0-7908-0001-2

Price: \$20.00

Foreword

This introduction to EDIF is the first in a series of monographs which will make the standard (ANSI/EIA RS548) easier to understand. It is intended to be used as a companion to the EDIF Reference Manual Version 2.0.0, which is also published by the EIA (ISBN 0-7908-0000-4).

This volume gives an overview of the format. Concepts are explained in a general way; constructs, such as *library*, appear in italics throughout this volume. Whenever more detail is desired about a topic the italicized constructs should be consulted in the EDIF Reference Manual. The EDIF Reference Manual contains the official definition of the format and should always be taken as the authoritative source of information. Specific application guides will be provided in later volumes of this series.

The diagrams which appear throughout this volume illustrate the general nesting structure of the format to provide a visual aid to the reader.

Contents

Preface	i
Section 1: Introduction	1-1
1.1 Introduction	1-2
1.2 The Structure of an EDIF Description	1-2
1.3 Views	1-3
1.4 Hierarchy	1-4
1.5 Ports and Nets	1-5
1.6 Levels	1-5
Section 2: Basic Syntax and Semantics	2-1
2.1 Introduction	2-1
2.2 EDIF Metasyntax	2-1
2.3 EDIF Syntax	2-2
2.4 Denoting Names or Identifiers	2-2
2.4.1 The Role of Names in EDIF	2-4
2.4.2 Name Class	2-5
2.4.3 Name Scope	2-6
2.4.4 Renaming for External Use	2-7
2.4.5 Displaying Names	2-7
2.4.6 Referencing EDIF Objects	2-7
2.5 Values	2-8
2.5.1 Integer Value	2-8
2.5.2 Number Value	2-9
2.5.3 String Value	2-9
2.5.4 Boolean Value	2-10
2.5.5 Point Value	2-10
2.5.6 MiNoMax Value	2-10
2.5.7 Displaying Values	2-11
2.6 User Extensions	2-12
2.6.1 Properties	2-12
2.6.2 UserData	2-12
2.7 Comments	2-12
Section 3: The EDIF File	3-1
3.1 Introduction	3-2
Section 4: Libraries	4-1
4.1 Cell Libraries	4-2
4.2 External Libraries	4-3
Section 5: Technology	5-1
5.1 Describing a Technology	5-2
5.2 Figure Groups	5-3
Section 6: Cells	6-1
6.1 Introduction	6-2



Documents

Document Sharing Hub

Contents

Section 7: Views	7-1
7.1 Introduction	7-2
7.1.1 Instances of Views of Other Cells	7-4
7.1.2 Arrays of Instances	7-5
Section 8: Interface and Contents	8-1
8.1 Interface	8-2
8.2 Contents	8-3
Section 9: Relating the Different Views	9-1
9.1 The ViewMap	9-1
Section 10: Level 1	10-1
10.1 Introduction	10-1
10.2 Extensions to Basic Constructs	10-1
10.2.1 Functions that Return Numbers	10-1
10.2.2 Functions that Return Booleans	10-2
10.2.3 Functions that Return Strings	10-2
10.2.4 Functions That Return Points	10-2
10.3 Constraint	10-3
Section 11: Level 2	11-1
11.1 Introduction	11-1
11.2 EDIF Looping Structures	11-1
11.2.1 Block	11-1
11.2.2 Iterate	11-1
11.2.3 While	11-2
11.2.4 Escaping from Loops	11-2
11.3 Variables and Assignment	11-2
11.4 Conditional Constructs	11-2
Section 12: Conclusion	12-1
12.1 Summary	12-1



Preface

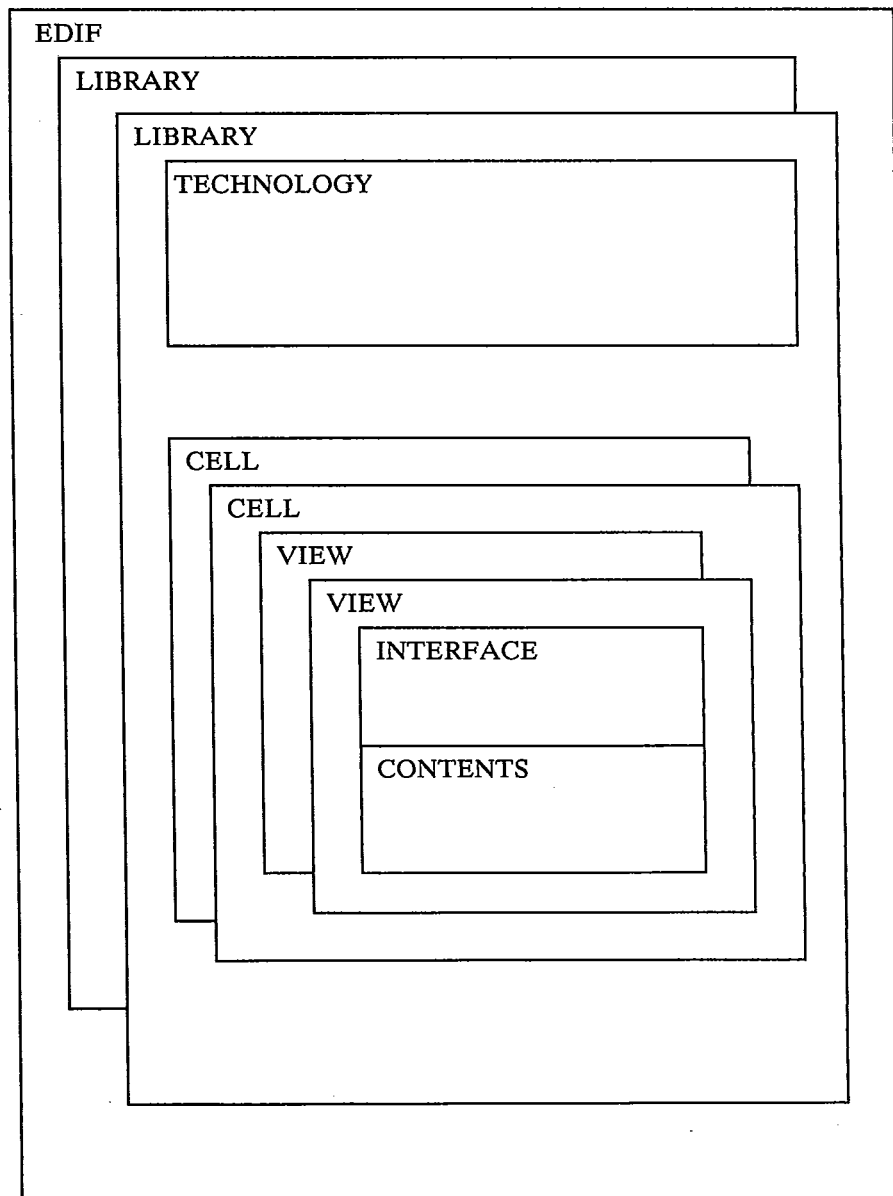
The increase in the number of silicon foundries and CAD/CAE system and workstation companies, and the problem of movement of data among in-house design systems has made a standard interchange format for electronic design data essential. The benefits of the general adoption of a standard interchange format are important and far reaching. The customer receives a much wider variety of feasible choices, and free competition in both CAE/CAD system/tool businesses and foundry businesses is enhanced. The electronic circuit designer and customer of CAE/CAD system and foundry service can expect compatibility among these products. The designer can choose equipment and services best suited for a particular task with a minimum of overhead. Foundries can work with customers regardless of the CAE/CAD equipment that the customer is using. The CAD/CAE tool or system vendor can maintain compatibility with a variety of foundries and other systems with a minimum investment in software development for that purpose.

Many interchange formats and hardware description languages have been developed over the past decade. However, each has suffered from one or more of the following drawbacks:

- Narrow focus – many formats have been developed to address only a specific aspect of the design data, such as mask artwork data or netlist information, but not both.
- Proprietary constraints – many formats are proprietary to specific semiconductor or CAD companies, severely limiting their effective use.
- Difficulty of Implementation – often, as a format evolves, it becomes increasingly difficult to develop parser/generator software for dealing with it; this is particularly true if upward compatibility is an issue.
- Difficulty of Extension – the rapid evolution of IC and PCB design methods causes new facets to be added to the data that must be transferred. Many languages or formats cannot grow to meet these new needs.

The Electronic Design Interchange Format has successfully addressed all of these problems and has made the movement of all design data a reality.

Section 1: Introduction



The Structure of an EDIF File

1.1 Introduction

An EDIF description is expressed in the form of a hierarchy, abstract at its higher levels, and becoming progressively more detailed as it descends the hierarchy. At the highest level of this structure the EDIF description contains, in a single logical file, all of the information that is shipped for a design. The file may contain several designs and several libraries of cell definitions. Each library is a collection of relevant EDIF design information including a grouping of cells according to a set of common characteristics.

It should be kept in mind that EDIF is a format for the transmission of design data. It is neither a programming language nor a data-base system. EDIF is not designed to be executable, and its definition does not specify any means of interacting with the information contained in an EDIF file, such as a graphics editor or a simulator. An EDIF file is intended to be read or written in its entirety by a database translator. It does, however, provide sufficient association between the elements of the format and electronic design automation concepts that such external interaction should be expected to produce predictable results. In other words, the meaning of a design should be preserved during its transfer to another design system through EDIF.

1.2 The Structure of an EDIF Description

A design description in EDIF is highly structured. Within an EDIF file there may be one or more libraries which contain information about technologies and cells. The information should be complete except for the contents of external libraries.

EDIF libraries may be locally defined, that is the descriptions of constituent cells are included in the current EDIF file. Libraries may also be external. In the latter case, cell details for the constituent cells may not be available at all in the current EDIF file. In addition to containing cell definitions, libraries also contain technology information. This is used to specify a variety of attribute details such as units of measurement, graphical default information for the library, and physical design rules.

EDIF supports a hierarchical approach to design representation; cells may be defined in terms of occurrences or instantiations of other cells. It is a fundamental requirement in EDIF that a "define-before-use" philosophy is maintained, so definitions of instantiated cells must be completed before the instantiations. This can result in single-pass readers which are both simple in structure and efficient.

1.3 Views

A cell definition in EDIF may be segmented into one or more views. These views classify the types of information that are needed to fully describe a cell. Examples of EDIF views are the SCHEMATIC view, the LOGICMODEL view, and the MASKLAYOUT view. The set of constructs available in different types of views allows a modular approach to be taken to the representation of different aspects of a cell description. It is largely a matter of individual CAD system preference to decide for which views of EDIF software will be implemented in order to best represent the data objects within the CAD databases.

Views are used to specify different representations of a cell. Each view holds a particular set of information which is relevant to that representation. The view type in EDIF is used to specify a particular representation and therefore limits the constructs which can be used within the *interface* and *contents* constructs. Information held within different view types may overlap since constructs are not restricted to one particular view type. This is very important since certain objects such as ports and nets are objects common to every view type except DOCUMENT and GRAPHIC. Also, connectivity may be expressed in more than one view.

Ten view types are allowed, each with specific use; eight are used directly for design information and two are for general purposes:

- The NETLIST view is used to define connectivity. It contains the details of instantiated cells and the connectivity between them.
- The SCHEMATIC view is used for both connectivity and associated diagrammatic information. Schematics depict symbols, wires connecting symbols, and some descriptive text, which may or may not span more than one page. A schematic is usually associated with one cell and depicts how it is made from a set of instantiated cells. Each instantiated cell is represented by a symbol. Wires may be drawn to connect the symbols. The text indicates the associated names. Extra information is also represented as text. Since there is a limit on the coordinate system, such as page size or window size, only a certain amount of information can be displayed at once (if it is to be readable). Therefore a facility, called a *page*, is provided for dividing the schematic.
- The SYMBOLIC view is used to describe the high level topological information about the cell. It is a flexible view and can be used as an intermediary stage between SCHEMATIC and MASKLAYOUT views.
- The MASKLAYOUT view is used to describe physical information about a design. It describes the figures on layers which will appear on the actual masks but only within the cell boundaries. The top level cell potentially describes the overall chip, but the instance hierarchy must be expanded in order to get all the manufacturing detail.

- The **PCBLAYOUT** view is similar to the **MASKLAYOUT** view in that figures appear on layers, but the top level cell refers to a board. A differentiation has been made between chip designs and pcb designs because the details of manufacturing information differ.
- The **BEHAVIOR** view is incomplete in the present release of EDIF.
- The **LOGICMODEL** view is used to describe the logic simulation model of the cell. It can express model behavior in detail and includes support for multiple logic values and timing.
- The **DOCUMENT** view is used to document the cell in a printable form. The textual information can be divided into sections and numbered; diagrams or symbols can be included.
- The **GRAPHIC** view is used to hold graphical information for a cell and can be used to define a figure such as a symbol shared between several cells. It can also be used for logos and page borders.
- The **STRANGER** view is available to deal with the situation where a CAD system cannot represent some of its data in any of the main EDIF views. Any construct is permitted in this view. However the semantics of this view is not guaranteed to be portable or to be understood by another system.

Within a given view of a cell, the data is further structured into an *interface* and an optional *contents*. The role of the *interface* is to specify the type of information that an instantiating cell will need in order to specify an occurrence of a particular cell view. This information may include port information so that connections may be made to the instantiated cell, as well as geometric information which defines the size and shape of the cell or the symbol used to represent it. The *contents* of a view allows details of the internal definition of the cell to be elaborated. This is typically achieved by including instantiations of simpler cells together with connectivity requirements.

1.4 Hierarchy

EDIF assumes that a design is hierarchical and that the top level of the design can be represented by one cell.

EDIF describes hierarchy two levels at a time; i.e. the current cell indicates how it is made up from other more basic cells. The simpler cells must be defined before the current cell so that the define-before-use rule is not broken. As a result, circular referencing is not possible. EDIF uses *instance* within the *contents* of the current cell to name each simpler cell, to refer to the definition of each cell, and to indicate how this definition is to be manipulated by geometric transformations.

1.5 Ports and Nets

In addition to the cell concept, two important concepts are ports and nets. These are used to communicate signal information between cells.

A port is a connection point to the cell. A direction may be associated with a port in order to define the flow of information. The port can contain additional information such as loading.

The width of the net and the ports it connects must be compatible. There are three basic types of port: a simple port, a port array, and a port bundle. The port array implies that the ports in the list have identical characteristics and that the width is equivalent to the number of ports involved in the array. The port bundle allows a nested structure to be formed consisting of simple ports, port arrays and port bundles. The width of a port bundle is the total number of simple ports together with the width of each port array used in the complete structure. The order in which the structure is formed is significant.

Nets are not restricted to having a single signal width or array size. Nets are used to establish connectivity between ports. The connectivity of a net associates a name with a set of ports. This connectivity can be expressed in a hierarchical manner so that sets of port sets can be amalgamated. For a particular parent view, a port (of a parent or child cell) may only appear in one top level net. Nested nets must not extend the connectivity of their parent net.

1.6 Levels

An interchange format for electronic design data must meet the needs of a wide range of users – from those wishing to transfer mask artwork data in its final form to those wishing to transfer a behavioral description of a microcomputer. A procedural format could be used to describe explicit mask artwork data. However, the data can be expressed as effectively using simple geometric primitives. To avoid complicating simple tasks, while still providing the power of procedural descriptions, EDIF descriptions can be specified in three different levels of complexity:

- Level 0: The basic level; only simple constants are permitted.
- Level 1: Level 0, with the addition of variables and expressions.
- Level 2: Level 1, with the addition of looping and conditional statements.

Level 1 is a proper subset of Level 2, and Level 0 is a proper subset of Level 1. Since more systems are capable of receiving low levels rather than high levels, it is advisable to declare the lowest level consistent with the features actually needed within a file. It is possible to interpret a complete design which is expressed in terms of high level constructs into a simple design database. However, this destroys the higher level structure in a way which is analogous to flattening an instance hierarchy. Such an operation may involve considerable effort, and will fail to transfer the high level cells that were used in the original design. Only a number of simple cells corresponding to each set of actual parameter values passed to the original high-level cell will be transferred.

Section 2:

Basic Syntax and Semantics

2.1 Introduction

In this section, the structure of a basic EDIF description is presented, and some of the generic issues, such as defining, naming, and referencing EDIF objects and values are discussed. All of the concepts described in this section apply to Level 0, Level 1, and Level 2. However, some of the constructs are extended in Levels 1 and 2.

A primary requirement of an interchange format is the ability to describe unambiguously the data it is transmitting. Consequently, it is important that all readers and writers of the format have exactly the same understanding of how the file is to be interpreted. A grammatical definition of EDIF is provided in the EDIF Reference Manual.

2.2 EDIF Metasyntax

The metasyntax used to describe EDIF constructs obeys the rules given below:

- Syntactic categories (non-terminals) are printed in *italics* and literal words and characters (terminal strings) are **emboldened**. Symbolic constants are printed in roman and are **UPPERCASE**.
- If a construct is enclosed in [square brackets] it is optional; if it is enclosed in {curly braces} it may be repeated zero or more times.
- A choice is indicated with a vertical bar |. Only one of the options may be chosen. A list of items enclosed within {curly braces} and separated by vertical bars indicates that any number of each item may be present and that the items may occur in any order. Inside such a list, if an item is permitted to occur at most once, it is enclosed within <chevrons>.
- As a guide for the reader, EDIF identifiers, or names, include either *NameDef* or *NameRef* as part of their specification to indicate whether, in the current use, the name is being defined or referenced.

- An ellipsis ... is used to indicate that further items have been omitted in an example.

2.3 EDIF Syntax

EDIF uses a Lisp-like syntax. The syntax was chosen to be easy to parse and provide a uniform way of extending the format when the need arises. The general form of an EDIF statement is:

```
( keywordNameRef
  { integerToken | stringToken | identifier | form }
)
```

A form begins with a left parenthesis and a keyword, and is followed by a list of items, ending with a closing right parenthesis. Those items may be other forms, thus building a nested structure, or they may be tokens consisting of data items such as integers or strings, or identifiers. Delimiters may occur either side of parentheses in EDIF forms but are not required. EDIF delimiters are the ASCII characters space, tab, carriage-return, and line-feed.

The list of legal identifiers that play the role of EDIF keywords is defined in the EDIF Reference Manual. Users of the format can also define their own keywords in terms of EDIF keywords using the *keywordMap* facility. Other mechanisms to support user extensions are also provided.

2.4 Denoting Names or Identifiers

Identifiers, or names, are used throughout an EDIF description to identify objects uniquely. The actual lexical form of EDIF identifiers is that of an ampersand & followed by between 1 and 255 alphanumeric or underscore characters. The ampersand may be omitted if the first character is alphabetic. All characters except the ampersand are significant. Identifiers are considered to be identical if their spelling differs only in the case of their constituent characters, or in the use of the ampersand character (which is not considered to be part of the name itself). In fact, case is only significant within strings in EDIF. For example,

VDD Vdd vdd &vdd – all represent the same identifier.

PortMap PORTMAP portMap – all represent the same keyword.

&100k – represents the external name 100k in EDIF, but