SLOVENSKI STANDARD

SIST ENV 50208-6:2002

01-september-2002

Data Interchange format for simulated and measured data (ISMD)

Data Interchange format for Simulated and Measured Data (ISMD)

Einheltliches Format zum Datenaustausch für simulierte und gemessene Größen

Ta slovenski standard je istoveten z: ENV 50208-6:1997

ICS:

35.240.50	Uporabniške rešitve IT v industriji	IT applications in industry

SIST ENV 50208-6:2002 **en**

EUROPEAN PRESTANDARD

PRÉNORME EUROPÉENNE

EUROPÄISCHE VORNORM

ENV 50208-6

June 1997

ICS 35.240.00

English version

Data Interchange format for Simulated and Measured Data (ISMD)

This European Prestandard (ENV) was approved by CENELEC on 1996-07-02 as a prospective standard for provisional application. The period of validity of this ENV is limited initially to three years. After two years the members of CENELEC will be requested to submit their comments, particularly on the question whether the ENV can be converted into a European Standard (EN).

CENELEC members are required to announce the existence of this ENV in the same way as for an EN and to make the ENV available promptly at national level in an appropriate form. It is permissible to keep conflicting national standards in force (in parallel to the ENV) until the final decision about the possible conversion of the ENV into an EN is reached.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and United Kingdom.

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

Central Secretariat: rue de Stassart 35, B - 1050 Brussels

Ref. No. ENV 50208-6:1997 E

Foreword

This European Prestandard has been prepared by CENELEC Technical Committee TC 217, Electronic Design Automation.

The text of the draft was submitted to CENELEC members for comments and vote and was approved at the meeting of TC 217 on 1996-07-02.

The following date was fixed:

– latest date by which the existence of the ENV has to be announced at national level (doa) 1997-09-01

Contents

Introduction		**4**
1	**Scope**	**5**
2	**Structure of the ISMD File**	**6**
3	**Elements of Exchange Format**	**7**
3.1	Equipment Description	7
3.2	Device Under Test (DUT) Description	8
3.3	Definition of Sources and Stimuli	9
3.4	Definition of Outputs	11
3.5	Data Section	11
3.6	INFO Section	12
3.7	Environment Section	13
3.8	Comments	13
3.9	END command	13
4	**Formal Rules**	**13**
4.1	Command Character and Reserved Keywords	14
4.2	Upper/Lower Case	14
4.3	Numbers, Values, Names	15
4.4	Delimiters	15
5	**Examples:**	**16**
5.1	Measured Data, Standard Device, Column Format	16
5.2	Measured Data, Standard Device, Table Format	16
5.3	Simulated Data, Circuit, Table Format	17
5.4	Simulated AC Data, Circuit, Table Format	17
5.5	Multiple Measured Data, Standard Device, Table Format	18
Annex A (informative):	**Literature**	**19**
Annex B (informative):	**Abbreviations**	**20**

Introduction

This document is part of a series of documents describing a technology assessment cycle of submicron CMOS technologies. The series consists of nine closely related documents: this one and [1, 2, 3, 4, 5] which are mainly technology performance related. Reliability related issues are covered by [6, 7, 8]. A documentation of the steps and the objective of the entire technology assessment cycle [1)] is the content of [1]. The transistor model which is able to deal with the effects of modern submicron CMOS technologies is presented in [2]. The parameter extraction procedures [5] are based on the measurement results which are derived from the devices of the modules of the Parameter Extraction Test Chip (PTC). The PTC is a part of the **European Mini Test Chip** (ETC) [3]. This mini test chip provides a minimum set of test structures to characterize a technology. The test structures of the ETC are generated automatically by a computer program [3] for a given MOS technology. The measurements which are the base of the extractions are documented in [4]. A common data exchange format (**Interchange Format for Simulated and Measured Data** - ISMD) can be used to ship the measured data of the PTC to companies which have no access to the PTC of the desired technology or the required measurement equipment. This document describes the exchange format of measured and simulated data. The format has been developed as a part of the work of the JESSI AC 41 consortium [2)]. Test structures for a fast reliability assessment with respect to the major failure mechanisms in CMOS technologies are described in [6]. The measurement techniques required for the characterization of the reliability test structures are documented in [7]. The evaluation of these measurements is subject of document [8].

The original purpose of the ISMD is the exchange of measured and simulated data of the JESSI test chips. This exchange is supposed to occur exclusively between the partners of the JESSI AC 41 consortium. This limited scope, however, had to be extended as the JESSI AC 41 project showed results which are useful for non-JESSI members as well as JESSI members.

1) The Technology Assessment Support Center (TASC) provides more information about the computer programs which support measurement and extraction routines. Send a fax or a e-mail to TASC, Technology Assessment Support Center, c/o H.Richter, IMS, Allmandring 30a, D-70569 Stuttgart
FAX: +49-711-685-5930
e-mail: tasc@svlsi1.mikro.uni-stuttgart.de

2) This work is part of the JESSI Program 'Application'-Project AC 41 'Technology Assessment' and is sponsored by the national Public Authorities of Belgium, France, and Germany.

1 Scope

The purpose of the Interchange of Simulated and Measured Data (ISMD) is to exchange measured and simulated data of test chips to make it available to different parties. For example, this data exchange format can be used to ship the measured data of test chips to companies which have no access to a testchip of a certain technology or the required measurement equipment, nevertheless they want to use these data, e.g. to prove the efficiency of a technology for a certain product.

This document describes a data interchange format which is applicable to

1. the gathering of measured and/ or simulated data which originate from a variety of measurement and simulation programs,
2. the support of a data representation format which can be read by a variety of existing data analysis programs with little or no modifications.

This task implies the establishment of a set of syntax rules. These syntax rules transform the existing data formats created by various measurement and/or data analysis programs into a single standard data representation format. A formal definition of the data elements beyond the syntax rules and the overall data structure has to be omitted in order to guarantee the flexibility necessary to handle the existing data formats. Additionally, short technical development cycles require a high degree of flexibility of the data elements. The definitions of the detailed data elements are subject to bilateral agreements. Accompanying information are included by means of comments.

The syntax described is the base of the data gathering. It may be necessary to modify the list of command words or the list of keywords in order to meet the requirements of non-electrical measurements or electrical measurements without known correlation of process technologies or wafer lots, and so on. For this situation, however, the syntax rules are supposed to serve as guidelines for the documentation of the measured data and the circumstances of the measurement.

The list of measurement programs, data analysis programs, and simulation programs is not necessarily limited to those programs used by the members of the JESSI AC 41 consortium.

2 Structure of the ISMD File

The ISMD file is structured in seven significant sections (see figure 1). Each section is started by a special command (see 4.1). The central section is the DATA section. The preceding six sections are named DEVICE, EQUIPMENT, ENVIRONMENT, INFO, SOURCES and OUTPUTS. They provide additional information about the data. In addition, COMMENT sections not interpreted by parsers are included to facilitate human reading. Each ISMD file begins with one COMMENT sections (leading COMMENT) and is terminated by an END command.

These sections define the following items:
1) INFO section;
2) ENVIRONMENT section;
3) description of the generating EQUIPMENT (measurement device, simulator);
4) description of the DEVICE under test (standard element, circuit) including nodes;
5) definition of SOURCEs and/or stimuli;
6) definition of OUTPUT;
7) DATA section;

For the following sections, it is mandatory to appear at least once in a file: DEVICE, SOURCE, OUTPUT, DATA and END. DEVICE, SOURCE, and OUTPUT have to appear before the DATA section.

The content of a section remains valid for the following sections until it is overwritten by the content of a following section of the same type. Empty sections are allowed in order to terminate the range of validity of a preceding section.

The data using the ISMD syntax rules originate from a variety of programs. These programs write their output data into files using their own program-specific syntax rules. In general, those programs which use ASCII characters in order to write their outputs can be divided into two groups. The programs of the first group write tables which contain the data of both independent and dependent variables. The programs of the second group, however, write tables which list the values of the dependent variables exclusively. Consequently, these programs must encode the values of the independent variables by means of reconstruction algorithms. The syntax rules of those reconstruction algorithms vary from program to program. In order to handle the resulting data of both groups the syntax rules of the ISMD are ambiguous with respect to the data and sources sections (see figure 1, subclause 3.5, and subclause 3.3, respectively).

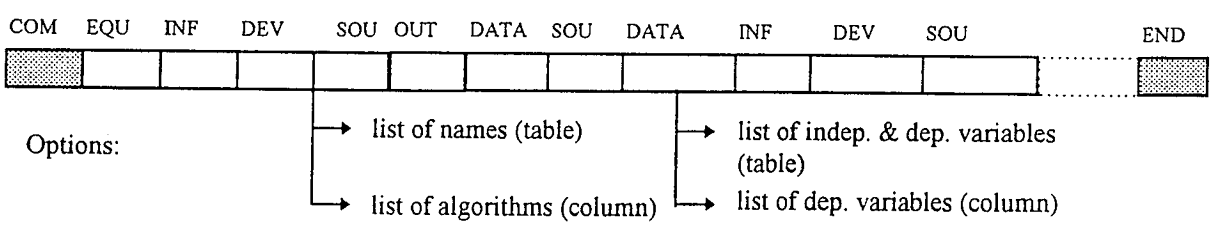

Figure 1: The internal structure of a ISMD data base file

The proposed format allows the combination of more than one data set into one file. The conditions for a dataset are always defined by the commands entered most recently in the file. For example, by re-entering a new device command, the same kind of data can be stored for different geometries of transistors. (see example 5.5).

3 Elements of Exchange Format

In the following clause, the ISMD syntax is described in a Backus-Naur like syntax description language. In order to avoid lengthly " " constructions, the graphical method of bold characters is used to express terminal characters. Within the text, production variables are printed cursive to separate them from surroundings.

The production rules use the follwing elements:

"="	(equal sign)	to relate *identifiers* to expressions,
"|"	(vertical bar)	to denote alternatives (or)
"{" "}"	(curly brackets)	indicating any number of repetitions ≥1
"[" "]"	(square brackets)	indicate optional elements (i.e. zero or one repetition)
strings printed **bold**		denote terminal elements
"."	(dot)	end of prodution rules
"**b**"		mandatory blank(s)
"**lf**"		end-of-line / line-feed character sequence.

ISMD files are constructed from six syntactical elements. The formal syntax is listed in syntax rules 0.1 to 0.6:

0.1	ISMDFile	=	CommentSection MainPart EndPart .
0.2	MainPart	=	FirstPart [{ MorePart }] .
0.3	FirstPart	=	EquipSection [EnvSection] [InfoSection] DevSection SourceSection OutputSection DataSection .
0.4	MorePart	=	{ NonDataSection } DataSection .
0.5	NonDataSection	=	EquipSection | EnvSection | InfoSection | DevSection | SourceSection | OutputSection .
0.6	EndPart	=	**END** [CommentText] **lf** .

Each section is described in the following subclauses, whereas clause 4 gives more formal rules on the ISMD format. In the following clauses also the short form of commands (e.g. EQU for Equipment) is used.

3.1 Equipment Description

The equipment description starts with the command **#EQUIPMENT=***EquipType*. This item indicates the equipment (e.g. parameter analyser HP4145, network analyser HP3577, TECAP [1]software, PSPICE[2] simulator etc.) that generated the data set and all the options and settings necessary to fully describe the equipment conditions (information of measurement points are

[1] TECAP is a computer program used to characterize transistors. It is copyrighted by HEWLETT-Packard Company, 5301 Stevens Creek Boulevard, Santa Clara, California

[2] PSPICE is a computer program used to simulate circuits, it is copyrighted by MicroSim, California, USA

located in the source section). This description is given as an ASCII text using as many lines as necessary and is terminated by the next ISMD command. Since this description is highly instrument specific, this part has to be defined for each instrument separately. The only restriction imposed by ISMD is that no command character can appear in the first column of the description. The equipment section is also the section to store the assignment between equipment channels like SMUs and device pins.

Syntax:

1.1	EquipSection	=	[CommentSection] EquipHeader [EquipBody] .
1.2	EquipHeader	=	CommandCharacter **EQU** = EquipType **lf** .
1.3	EquipBody	=	{ EquipLine | CommentLine } .
1.4	EquipLine	=	NonSpecChar { PrintableCharacter } **lf** .
1.5	EquipType	=	Name .

3.2 Device Under Test (DUT) Description

The DUT description starts with the command **#DEV=***DeviceType (DeviceType* is either StdDevType or **CIRCUIT**) . In this section, the electrical description of the DUT is given. Information like wafer number, position on wafer (chip number) have to be placed into the info section.

The following standard device types are implemented following the SPICE[3] conventions: **NMOS, PMOS, NPN, PNP, DIODE, RES, CAP** etc. The first item in the standard DUT definition is **DUT_NAME=***DeviceName*. It is followed by geometry and other parameters (e.g. **W, L, AD, PS** etc.) used according to the SPICE conventions followed by "=" and their *values*.

Non-standard devices have the *DeviceType* **CIRCUIT** and two mandatory items **DUT_NAME=***DeviceName* and **CIRCUIT_TYPE=***CircuitDescriptionType*. With **DUT_NAME** a special name is assigned to the circuit, **CIRCUIT_TYPE** specifies the type of circuit description e.g. SPICE, SABER [4] etc. The circuit is then implemented e.g. as SPICE or SABER circuits following their simulator specific rules.

Syntax:

2.1	DevSection	=	StdDevSection | CircuitDevSection .
2.2	StdDevSection	=	[CommentSection] StdDevHeader StdDevBody .
2.3	StdDevHeader	=	CommandCharacter **DEV=** StdDevType **lf** .
2.4	StdDevType	=	Name .
2.5	StdDevBody	=	{ StdDevLine | CommentLine } .
2.6	StdDevLine	=	DevParName = DevParValue **lf** .
2.7	DevParName	=	Name .
2.8	DevParValue	=	RealNumber | Name | CompDevParValue .
2.9	CompDevParValue	=	(DevParName = DevParValue { **b** DevParName = DevParValue}) .
2.10	CircuitDevSection	=	[CommentSection] CircuitDevHeader CircuitDevBody .
2.11	CircuitDevHeader	=	CommandCharacter **DEV= CIRCUIT lf**

[3] SPICE is a computer program used to simulate circuits, issued by the University of California, Berkeley
[4] SABER is a computer program used to simulate circuits, copyrighted by Analogy, USA

2.12	CircuitDevBody	=	DevNameLine CircuitTypeLine OtherCircuitLines .
2.13	DevNameLine	=	**DUT_NAME=** DeviceName .
2.14	CircuitTypeLine	=	**CIRCUIT_TYPE=** CircuitDescriptionType .
2.15	OtherCircuitLines	=	{ PrintableCharacter } **lf** .
2.16	DeviceName	=	Name .
2.17	CircuitDescriptionType	=	Name .

3.3 Definition of Sources and Stimuli

The source definition starts with the command #**SOU**. This section defines DC and AC sources (independent variables) to be used with the data set. Transient sources can only be defined within the table format.

The source section defines the instrument independent measurement points whereas all instrument dependent parameters like resolution, integration time have to be put into the equipment section. Sources can be defined either as constant values, lists of value, sweeps or a relation to another source.

Sweep sources and list sources are only used in connection with the column data format. Sweep sources may be nested. In this case, the first sweep source defined is swept slowest. Each source definition has to be used in a separate line in the file.

Syntax:

3.1	SourceSection	=	[CommentSection] SourceHeader SourceBody .
3.2	SourceHeader	=	CommandCharacter **SOU** **lf** .
3.3	SourceBody	=	{ SourceDefinition | CommentLine } | .
3.4	SourceDefinition	=	SourceName = TypePart ValuePart [**b** TypePart ValuePart] [**b** TypePart ValuePart] .
3.5	NormalSource	=	SourceName=TypePart ValuePart [**b** TypePart ValuePart] [**b** TypePart ValuePart] **.**
3.6	TypePart	=	**DC=** | **AC=** | **FREQ=** | .
3.7	ValuePart	=	Value |(SweepMod **b** StartValue **b** StopValue **b** NumberSteps) | (**LIST** **b** Value { **b** Value }) | (**SYNC** **b** OtherSource [Operator Factor]) .
3.8	SweepMod	=	**LIN** | **LOG** | .
3.9	StartValue	=	Value .
3.10	StopValue	=	Value .
3.11	StepNumber	=	IntegerNumber .
3.12	Operator	=	+ | * | - | / .
3.13	Factor	=	RealNumber .
3.14	Value	=	DCPart | ACPart | DCPart ACPart .
3.15	DCPart	=	RealNumber
3.16	ACPart	=	[RealPart / ImaginaryPart] | [Magnitude **P** PhaseInDegrees] .
3.17	RealPart	=	RealNumber .
3.18	ImaginaryPart	=	RealNumber .
3.19	Magnitude	=	RealNumber .
3.20	PhaseInDegrees	=	RealNumber .
3.21	SourceName	=	Name .
3.22	OtherSource	=	SourceName .