micross components

The DDX File Format

A review of ES59008-6-1 and IEC 62258 Part 2

ES59008 & IEC62258 are now available from BSI Publications

Alun Jones

www.micross.com

micross components

Background

- GOOD-DIE project established under ESPRIT to co-ordinate European efforts on die information
- Project task to review current standards for die
- Requirements noted for technical data on die products, its availability, exchange and use in CAD systems
- Current standards were not seen as adequate for all needs
- ENCAST project continuing on the standards development.

micross components

Acknowledgements

- The project participants are grateful for financial support from the European Commission under the ESPRIT 4th Framework initiative
- Other organisations who have contributed to the work include :
  - IMEC, Alcatel, Philips Semiconductors AG, SAAT Technology Ltd, Micross Components Ltd., TS2 Ltd., Mintech Semiconductors, Austin Semiconductors Ltd., Eltek Semiconductors Ltd, Codus Ltd, Lucas Aerospace, Infineon, Die Products Consortium, Hitachi, British Standards Institution, CENELEC, National Semiconductor, Chip Supply Inc, MCC, Micronas, GEC-Plessey Semiconductors, Temic, VDE, University of Manchester UK,

micross components

Data requirements for semiconductor die – ES 59008

- Part 1 General requirements
- Part 2 Vocabulary
- Part 3 Mechanical, material and connectivity requirements
- Part 4 Specific requirements and recommendations
  - Part 4-1 Test and quality
  - Part 4-2 Handling, assembly and storage
  - Part 4-3 Thermal
  - Part 4-4 Electrical simulation
- Part 5 Particular requirements and recommendations for die types
  - Part 5-1 Bare die
  - Part 5-2 Bare die with added connection structures
  - Part 5-3 Minimally-packaged die
- **Part 6 Exchange data formats and data**
  - **Part 6-1 Data exchange - DDX**
  - Part 6-2 Data dictionary

micross components

IEC 62258

Minimum requirements for the procurement and use of semiconductor die products

- Part 1: Requirements for procurement and use
- **Part 2: Exchange data formats**
- Part 3: Recommendations for good practice in handling, packing and storage (Technical Report)
- Part 4: Questionnaire for die users and suppliers (Technical Report)
- Part 5: Requirements for information concerning electrical simulation
- Part 6: Requirements for information concerning thermal simulation
- Part 7: XML schema for data exchange (Technical Report)
- Part 8: EXPRESS model schema for data exchange (Technical Report)
- Part 9: Visual inspection (Technical Report)
- Part 10: Die data sheets (Technical Report)

micross components

Exchange data formats - DDX

- Defines the format required for the transfer of data by electronic media -

  Device Data eXchange format …
- Contains fundamental Geometric and Electrical interconnect data
- File contents purely textual / ASCII
  - *Totally readable*

micross components

Exchange data formats – DDX

AIMs and GOALs

- Simple textual structure allows :
  - Easily read, adjusted and understood
    - can be created by a text editor or spreadsheet
  - Simple and tolerant syntax
    - case and space insensitive for key parameters
    - easily written by engineers and non-engineers !!
    - simple parsing requirements for CAD systems.
  - Simple BLOCK structure
    - may be used as a component or project library file
  - Small file size
    - containing only the bare essentials
  - Permits reference to external data / simulation files
  - Flexibility, permitting additional component types

micross components

Exchange data formats - DDX

Fundamentals

- **Data types**
  - Textual String
  - Textual Name *(limited ASCII set for O/S)*
  - Real Numeric
  - Integer Numeric
  - Date *(ISO 8601:1988 format compliant)*
- **Parameters**
  - **`Parameter = Value ;`** …… That's it !!!
- **Structures**
  - Main **Block** structure for a specific device & form.
  - Structures for
    - **Terminals**
    - **Terminal types**

micross components

Exchange data formats - DDX

Fundamentals : Block Structure

- **Typical DEVICE Block Structure :**

```
DEVICE SN7400AJ bare_die {
  relevant data for device "SN7400AJ" as a bare die..
}
DEVICE PAL22V10CEP bare_die {
  relevant data for device "PAL22V10CEP" as a bare die..
}
DEVICE PAL22V10CEP bumped_die {
  relevant data for device "PAL22V10CEP" as a bumped die..
}
```

- **Current Device forms :**
  - bare_die
  - bumped_die
  - mpd (minimally packed device)
  - lead_framed_die

Exchange data formats - DDX

micross components

Parameters

- **Text data**
  - Device Version, Manufacturer, Function, Mask, Mask revision, materials
  - DDX file version, data source
- **Geometric data**
  - Dimension Units, Geometric View, Geometric Origin
  - Size, Thickness and dimensional tolerances
  - Terminal shapes, sizes, location & orientation
- **Interconnect & Electrical data**
  - Terminal names, I/O properties & function, substrate connections
  - Fabrication Technology. Operational Temperature & Power Dissipation
  - Data specific to external Simulation model files
- **Additional data**
  - Device & technology specific details. Delivery form data.
  - Fiducial details and graphic links. Additional material details.

Exchange data formats - DDX

Parameters : The TERMINAL_TYPE structure

- **Single TERMINAL_TYPE declaration :**

  TERMINAL_TYPE *Terminal_type_name = Terminal_shape_type, Co-ordinates.,,,;*

- **Multiple TERMINAL_TYPE declaration :**

```
{
    Terminal_type_name = Terminal_shape_type, Co-ordinates .,,,;
    Terminal_type_name = Terminal_shape_type, Co-ordinates .,,,;
    Terminal_type_name = Terminal_shape_type, Co-ordinates .,,,;
}
```

Where ...

Terminal_type_name	Unique name for the .
Terminal_shape_type	Single character indicating the shape, as Rectangle, Circle, Ellipse, Polygon.
Co-ordinates .,,,;	A series of X & Y co-ordinates required to determine the Terminal Type shape.

Notes …

The origin for placement is defined as (0,0) relative to the given co-ordinates.

micross components

Exchange data formats - DDX

Parameters : The TERMINAL structure

- **Single TERMINAL declaration :**

  TERMINAL T_*n* = *conn_N*, *Term_type_name*, *X-co.*, *Y-co.*, *orient.*, *Term_name*, *IO_type*;

- **Multiple TERMINAL declaration :**

```
TERMINAL {
    T_n = conn_N, Term_type_name, X-co., Y-co., orient., Term_name, IO_type;
    T_n = conn_N, Term_type_name, X-co., Y-co., orient., Term_name, IO_type;
    T_n = conn_N, Term_type_name, X-co., Y-co., orient., Term_name, IO_type;
}
```

Where ...

T_n	Unique Terminal number.
conn_N	Non-unique connection number (may be used to indicate multiple pin connections)
Term_type_name	Name of **Terminal Type**, as previously declared
X-co, Y-co, orient	X and Y co-ordinates and orientation for the placement of the Terminal
Term_name	Non-unique name, used as placement text, only used for user reference and graphic display.
IO_type	Single character electrical function descriptor, e.g.

I … digital Input,
O … digital Output
V … Supply, may be any supply type,
G … Ground, etc.

Exchange data formats - DDX

An example (part 1)

- **The BLOCK Header with with block and device history.**

```
DEVICE Reindeer1 bare_die {
  # A hash indicates the remaining line is ignored
        BLOCK_CREATION_DATE = "1997-12-25";
        BLOCK_VERSION   = 1.0;
        MANUFACTURER    = "Santa Claus Logic Ltd.";
        DATA_SOURCE     = "GOOD-DIE Project database";
        VERSION         = "1.2.0";
```

- **Declaration of geometric view, units, size etc.,**

```
        GEOMETRIC_UNITS          = millimetre;
        GEOMETRIC_VIEW           = "top";
         SIZE                    = 1.312, 1.050;
         SIZE_TOLERANCE          = 0.00, 0.0005, 0.00 0.0005;
         THICKNESS               = 0.360;
         THICKNESS_TOLERANCE     = 0.00, 0.0007;
        GEOMETRIC_ORIGIN         = 0,0;
```

micross components

Exchange data formats - DDX

An example (part 2)

- **Additional details of Die type and usage.**

```
MASK_REVISION                    = “Mask 1.0”;
MAX_TEMP                         = 280;
POWER_RANGE                      = 0.500;
DIE_SEMICONDUCTOR_MATERIAL       = “Silicon”;
IC_TECHNOLOGY                    = “bipolar”;
DIE_SUBSTRATE_CONNECTION         = “CONN”,”Ground”;
DIE_TERMINAL_MATERIAL            = “Al”;
DIE_PASSIVATION_MATERIAL         = “Polyimide”
```

- **Delivery details.**

```
DIE_BACK_DETAIL                  = “Back-Lapped”;
DIE_DELIVERY_FORM                = “Die, Wafer”;
WAFER_SIZE                       = “4 inch”;
```

Exchange data formats - DDX

An example (part 3)

- **Definition of the number of bond pad types, bond pads and connections.**

```
TERMINAL_TYPE_COUNT    = 5;
TERMINAL_COUNT         = 8;
CONNECTION_COUNT       = 14;
```

- **Definition of the bond pad shapes and dimensions.**

```
TERMINAL_TYPE  {
    PADR1 = Rectangle, 0.144, 0.104;
    PADR2 = Rectangle, 0.264, 0.104;
    PADR3 = Rectangle, 0.084, 0.084;
    PADC1 = Circle,    0.100;
    PADP1 = Polygon,      (-0.0175,-0.042),(-0.042,-0.0175),
                          (-0.042, 0.0175),(-0.0175, 0.042),
                          ( 0.0175, 0.042),( 0.042, 0.0175),
                          ( 0.042,-0.0175),( 0.0175,-0.042);
               }
```

Exchange data formats - DDX

An example (part 4)

- Bond pad placement, naming, orientation and connectivity details.

```
TERMINAL {
    T_1 = 1 ,PADC1,-0.550, 0.416,  0,VCCA   ,P;
    T_2 = 3 ,PADP1,-0.502, 0.190,  0,INPUTA ,I;
    T_3 = 4 ,PADP1,-0.502,-0.192,  0,INPUTB ,I;
    T_4 = 7 ,PADC1,-0.399,-0.442,  0,GNDA   ,G;
    T_5 = 8 ,PADR2, 0.498,-0.442,  0,GNDB   ,G;
    T_6 = 11,PADR3, 0.511,-0.171,  0,OUTPUTA,O;
    T_7 = 12,PADR3, 0.511, 0.171,  0,OUTPUTB,O;
    T_8 = 14,PADR1, 0.558, 0.416,  0,VCCB   ,P;
}
```

- Details of a reference fiducial, supplied as a JIF graphic file

```
FIDUCIAL_TYPE fiduc1 = “Xmas.JIF”, 0.072, 0.055;
FIDUCIAL F1 = fiduc1, -0.612, 0.470, 0;
```

Exchange data formats - DDX

An example (part 5)

- **Details of a supplied pSpice simulation model.**

```
SIMULATOR_SPICE_MODEL_FILE          = "Rudolph.MOD";
SIMULATOR_SPICE_MODEL_FILE_DATE     = "1997-09-17";
SIMULATOR_SPICE_NAME                = "pSpice";
SIMULATOR_SPICE_VERSION             = "4.0.1";
SIMULATOR_SPICE_COMPLIANCE          = "2G6";
```

- **Details of a supplied Spectre simulation model.**

```
SIMULATOR_SPECTRE_MODEL_FILE        = "Prancer.S";
SIMULATOR_SPECTRE_MODEL_FILE_DATE   = "1998-11-05";
SIMULATOR_SPECTRE_NAME              = "Spectre";
SIMULATOR_SPECTRE_VERSION           = "4.2.1, 1992";
SIMULATOR_SPECTRE_COMPLIANCE        = "2G6, Level-3";
```

- **End of the block**

```
}
```

micross components

Exchange data formats - DDX

- **XML model File (.XML) example available**
  - **courtesy of Dr. Donald Radley, ENCASIT**
- **EXPRESS Model available**
  - **courtesy of**
    - **Prof. Hilary Kahn and Dr. Alan Williams**
    - **University of Manchester, UK**
- **STEP Physical File (SPF) example available**
  - **courtesy of Dr. Donald Radley, ENCASIT**

IEC 62258

micross components

- Part 2 includes DDX Format version at 1.3.0 as of 1st Maintenance issue, Oct 2008
- Full XML model in Part 7
- Full EXPRESS model in Part 8
- STEP Physical File example included
- Additional parameters
- Enhanced explanatory annexes.

micross components

Partial parameter listing (1)

- BLOCK_CREATION_DATE
- BLOCK_VERSION
- VERSION
- DEVICE_FORM
- DEVICE_NAME
- DIE_MASK_REVISION
- MANUFACTURER
- DIE_NAME
- DIE_PACKAGED_PART_NAME
- FUNCTION
- DATA_SOURCE
- DATA_VERSION
- GEOMETRIC_UNITS
- GEOMETRIC_VIEW
- SIZE
- THICKNESS
- GEOMETRIC_ORIGIN
- SIZE_TOLERANCE
- THICKNESS_TOLERANCE
- TERMINAL_COUNT
- TERMINAL_TYPE_COUNT
- CONNECTION_COUNT
- TERMINAL_TYPE
- TERMINAL
- IC_TECHNOLOGY
- DIE_SEMICONDUCTOR_MATERIAL
- DIE_SUBSTRATE_MATERIAL
- DIE_SUBSTRATE_CONNECTION
- DIE_PASSIVATION_MATERIAL
- DIE_TERMINAL_MATERIAL
- DIE_BACK_DETAIL
- WAFER_SIZE
- MAX_TEMP
- POWER_RANGE

micross components

Partial parameter listing (2)

- TEMPERATURE_RANGE
- Simulator MODEL FILE
- Simulator MODEL FILE DATE
- Simulator NAME
- Simulator VERSION
- Simulator COMPLIANCE
- DIE_DELIVERY_FORM
- PACKING_CODE
- BUMP_MATERIAL
- BUMP_HEIGHT
- BUMP_HEIGHT_TOLERANCE
- MPD_PACKAGE_MATERIAL
- MPD_PACKAGE_STYLE
- MPD_DELIVERY_FORM
- MPD_CONNECTION_TYPE
- MPD_CONNECTION_MATERIAL
- FIDUCIAL_TYPE
- FIDUCIAL
- TEST_AND_QUALITY
- TEST_YIELD_CODE
- TEST_FLOW
- TEST_TEMP
- TEST_SCREEN
- TEST_RELIABILITY