SPIE Handbook of Microlithography, Micromachining and Microfabrication

Volume 1: Microlithography

2.9 Appendix: GDSII Stream Format

Table of Contents
Previous section: 2.8 Acknowledgement
Next section: 2.10 References
Order the complete book from SPIE

Integrated Disordered Optical Resonators, CNF Project # 980-01. PI: Michal Lipson, Electrical and Computer Engineering, Cornell University, mlipson@ece.cornell.edu. Figure: SEM of disordered resonator in SOI.
Full report in PDF

2.9 Appendix: GDSII Stream Format

Portions of the GDSII Stream Format Manual, Documentation No. B97E060, Feb. 1987, reprinted with permission of Cadence Design Systems, Inc., 555 River Oaks Parkway, San Jose, CA 95134. See also the web site http://www.cadence.com.

The following is a description of the GDSII Stream data format (Release 6.0), the most commonly used format for electron beam lithography and photomask production [197]. This appendix omits the description of tape formatting, since disk files and disk file images on tape and other media are now the norm [198].

The pattern data is considered to be contained in a "library" of "cells". Cells may contain geometrical objects such as polygons (boundaries), paths, and other cells. Objects in the cell are assigned to "layers" of the design. Different layers typically represent different processing steps for exposure on separate mask plates. Geometrical objects may also be tagged with "datatypes", which can be used for any purpose, but are most commonly used to group together similarly sized objects for compensation of the proximity effect.

There is no explicitly stated limit to the level of hierarchy (the degree of cell nesting); however, most CAD programs impose a limit of around 32 levels. GDSII interpreters will either impose such a limit explicitly, or will impose an implicit limit by running out of memory during recursive operations.

2.9.1 Order of records:

A GDSII Stream file has a great deal of flexibility, but must contain at least the following:

1. A header record
2. One or more Stream records
3. Library name record
4. End of library token

An example of a common record order (see below for record descriptions) follows:

HEADER	version number
BGNLIB	last modification date
LIBNAME	library name
GENERATIONS	see below
UNITS	data units
BGNSTR	begin structure
STRNAME	structure name
BOUNDARY	begin boundary (polygon)
LAYER	layer number
DATATYPE	a label associated with this item
XY	coordinates
ENDEL	end of element
(etc.)	
ENDSTR	end of structure (cell)
ENDLIB	end of library

2.9.2 Record description

The GDSII Stream file format is composed of variable length records. The minimum record length is four bytes. Records can be infinitely long. The first four bytes of a record are the header. The first two bytes of the header contain a count (in eight-bit bytes) of the total record length. The count tells you where one record ends and another begins. The next record begins immediately after the last byte included in the count. The third byte of the header is the record type (also known as a "token"). The fourth byte of the header describes the type of data contained within the record (see table below). The fifth through last bytes of a record are data.

2.9.3 Data type description

The data type value is found in the fourth byte of the record. Possible types and values are:

Data Type	Value
No data present	0
Bit array	1
Two-byet signed integer	2
Four-byte signed integer	3
Four-byte real (not used)	4
Eight-byte real	5
ASCII string	6

Two- and four-byte signed integers use the usual twos complement format for negative values. The more significant bytes appear first in the file, so that by default no byte swapping is required when reading the integers with a big-endian

CPU (e.g., Intel processors). Byte swapping is required when reading or writing integers with a little-endian machine, such as a VAX.

Real numbers are *not* represented in IEEE format. A floating point number is made up of three parts: the sign, the exponent, and the mantissa. The value of the number is defined to be (mantissa) (16) $^{(exponent)}$. If "S" is the sign bit, "E" are exponent bits, and "M" are mantissa bits then an 8-byte real number has the format

SEEEEEEE MMMMMMMM MMMMMMMM MMMMMMMM
MMMMMMMM MMMMMMMM MMMMMMMM MMMMMMMM

The exponent is in "excess 64" notation; that is, the 7-bit field shows a number that is 64 greater than the actual exponent. The mantissa is always a positive fraction greater than or equal to 1/16 and less than 1. For an 8-byte real, the mantissa is in bits 8 to 63. The decimal point of the binary mantissa is just to the left of bit 8. Bit 8 represents the value 1/2, bit 9 represents 1/4, and so on.

In order to keep the mantissa in the range of 1/16 to 1, the results of floating point arithmetic are *normalized*. Normalization is a process whereby the mantissa is shifted left one hex digit at a time until its left *four* bits represent a non-zero quantity. For every hex digit shifted, the exponent is decreased by one. Since the mantssa is shifted four bits at a time, it is possible for the left three bits of a normalized mantissa to be zero. A zero value is represented by a number with all bits zero. The representation of negative numbers is the same as that of positive numbers, except that the highest order bit is 1, not 0.

2.9.4 Record types

Records are always an even number of bytes long. If a character string is an odd number of bytes long it is padded with a null character. The following is a list of record types. The first two numbers in brackets are the record type and the last two numbers in brackets are the data type (see the table above). *Note that the data type (e.g. "two-byte signed integer") refers to the type of data to follow in the record, not to the number of bytes in the record.* The first two bytes of the record *header* contain a count (in eight-bit bytes) of the total record length. The third byte of the header is the record type (also known as a "token") shown below, and the fourth byte is the data type. All record numbers are shown in hexidecimal. For example, in the HEADER record, "00" is the token, and "02" is the data type.

HEADER [0002]	Two-byte signed integer: contains data representing the GDSII version number. Values are 0, 3, 4, 5, and 600. With release 6.0 the bersion number changes to three digits.
BGNLIB [0102]	Two-byte signed integer: contains last modification time of library (two bytes each for the year, month, day, hour, minute, and second) as well as time of last access (same format) and marks beginning of library.

word 1	0x1C (hex) # bytes in record
word 2	0x0102 (the token for bgnlib)
word 3	year of last modification

words 4-8	month, day, hour, minute, second
word 9	year of last access time
words 10-14	month, day, hour, minute, second

LIBNAME [0206]	ASCII string: contains a string which is the library name. The string must adhere to CDOS file name conventions for length and valid characters, and may contain file extensions such as ".db".
UNITS [0305]	Eight-byte real: contains 2 8-byte real numbers. The first is the size of a database unit in user units. The second is the size of a database unit in meters. For example, if your library was created with the default units (user unit = 1 m and 1000 database units per user unit), then the first number would be 0.001 and the second number would be 10^{-9}. Typically, the first number is less than 1, since you use more than 1 database unit per user unit. To calculate the size of a user unit in meters, divide the second number by the first.
ENDLIB [0400]	No data is present. This marks the end of a library.
BGNSTR [0502]	Two-byte signed integer: contains creation time and last modification time of a structure (in the same format as that of BGNLIB) and marks the beginning of a structure.
STRNAME [0606]	ASCII string: contains a string which is the structure name. A structure name may be up to 32 characters long. Legal characters are 'A' through 'Z', 'a' through 'z', '0' through '9', underscore, question mark, and the dollar sign, '$'.
ENDSTR [0700]	No data is present. This marks the end of a structure.
BOUNDARY [0800]	No data is present. This marks the beginning of a bounary element (polygon).
PATH [0900]	No data is present. This marks the beginning of a path element.
SREF [0A00]	No data is present. This marks the beginning of a structure reference element (a reference or "call" to another cell in the library).
AREF [0B00]	No data is present. This marks the beginning of an array reference element (an array of cells).
TEXT [0C00]	No data is present. This marks the beginning of a text element.
LAYER [0D02]	Two-byte signed integer: contains the layer number. The value must be from 0 to 63.
WIDTH [0F03]	Four-byte integer: contains the width of a path or text lines in database units. A negative value for width means that the width is absolute; i.e., I is not affected by the magnification

	factor of any parent reference. If omitted, zero is assumed.
XY [1003]	Four-byte signed integer: contains an array of XY coordinates in database units. Each X or Y coordinate is four bytes long. Path and boundary elements may have up to 200 pairs of coordinates. A path must have at least 2, and a boundary at least 4 pairs of coordinates. The first and last point of a boundary must coincide. A text or SREF element must have only one pair of coordinates. An AREF has exactly three pairs of coordinates, which specify the orthogonal array lattice. In an AREF the first point locates a position which is displaced from the reference point by the inter-column spacing times the number of columns. The third point locates a position which is displaced from the reference point by the inter-row spacing times the number of rows. A node may have from 1 to 50 pairs of coordinates. A box must have five pairs of coordinates with the first and last points coinciding.
ENDEL [1100]	No data is present. This marks the end of an element.
SNAME [1206]	ASCII string: contains the name of a referenced structure.
COLROW [1302]	Two-byte signed integers: the first 2 bytes contain the number of columns in the array. The third and fourth bytes contain the nunber of rows. Neither the number of columns nor the number of rows may exceed 32,767 (decimal) and both are positive.
NODE [1500]	No data is present. This marks the beginning of a node.
TEXTTYPE [1602]	Two-byte signed integer: contains the text type. The value of the text type must ge in the range of 0 to 63.
PRESENTATION [1701]	Bit array: contains 2 bytes of bit flags for text presentation. Bits 10 and 11, taken together as a binary number, specify the font. Bits 12 and 13 specify the vertical presentaton (00 means top, 01 means middle, and 10 means bottom). Bits 0 through 9 are reserved for future use and must be cleared. If this record is omitted, then top-left justification and font 0 are assumed.
STRING [1906]	ASCII String: contains a character string for text presentation, up to 512 characters long.
STRANS	Bit array: contains two bytes of bit flags for SREF, AFREF,

[1A01]	and text transformation. Bit 0 (leftmost) specifies reflecton. If it is set, then reflection about the X axis is applied before angular rotation. For AREFs, the entire array lattice is reflected, with the individual array elements riidly attached. Bit 13 flags absolute magnification. Bit 14 flags absolute angle. Bit 15 (rightmost) and all remaining bits are reserved for future use and must be cleared. If this record is omitted, then the element is assumed to have no reflection and its magnification and angle are assumed to be non-absolute.
MAG [1B05]	Eight-byte real: contains a magnification factor. If omitted, a magnification of 1 is assumed.
ANGLE [1C05]	Eight-byte real: contains the angular rotation factor, measured in degrees, counterclockwise. For an AREF, the angle rotates the entire array lattice (with the individual array elements regidly attached) about the array reference point. If this record is omitted, and algle of zero degrees is assumed.
REFLIBS [1F06]	ASCII string: contains the names of the reference libraries. This record must be present if there are any reference libraries bound to the current library. The name for the first reference library starts at byte 0 and the name of the second library starts at byte 45 (decimal). The reference library names may include directory specifiers (separated with ":") and an extension (separated with "."). If either library is not named, its place is filled with nulls.
FONTS [2006]	ASCII string: contains names of textfont definition files. This record must be present if any of the 4 fonts have a corresponding textfont definition file. This record must not be present if none of the fonts have a textfont file. The name of font 0 starts the record, followed by the remaining 3 fonts. Each name is 44 bytes long and is null if there is no corresponding textfont definition. Each name is padded with nulls if it is shorter than 44 bytes. The textfont definition file names may include directory specifiers (separated with ":") and an extension (separated with ".").
PATHTYPE [2102]	Two-byte signed integer: contains a value of 0 for square-ended paths that end flush with their endpoints, 1 for round-ended paths, and 2 for square-ended paths that extend a half-width beyond their endpoints. Pathtype 4 signifies a path with variable square-end extensions (see BGNEXTN and ENDEXTN).
GENERATIONS [2202]	Two-byte signed integer: contains a positve count of the number of copies of deleted or backed-up structures to retain. This number must be at least 2 and not more than 99. If the GENERATIONS record is not present, a value of 3 is assumed.
ATTRTABLE [2306]	ASCII string: contains the name of the attribute definition file. This record is present only if there is an attribute definition file bound to the library. The attribute definition file name

	may include directory specifiers and an extension (see FONTS). Maximum size is 44 bytes.
EFLAGS [2601]	Bit array: contains 2 bytes of bit flags. Bit 15 (rightmost) specifies template data. Bit 14 specifies external data (also referred to as "exterior" data). All other bits are currently unused and must be cleared to 0. If this record is omitted, then all bits are assumed to be 0. Further information about template data can be found in the *GDSII Reference Manual*. Information about external data can be found in the *CustomPlus User's Manual*.
NODETYPE [2A02]	Two-byte signed integer: contains the node type. The value of the node type must be in the range of 0 to 63.
PROPATTR [2B02]	Two-byte signed integer: contains the attribute number. The attribute number is an integer from 1 to 127. Attribute numbers 126 and 127 are reserved for the user integer and user string properties, which existed prior to Release 3.0.
PROPVALUE [2C06]	ASCII string: contains the string value associated with the attribute named in the preceding PROPATTR record. Maximum length is 126 characters. The attribute-value pairs associated with any one element must all have distinct attribute numbers. Also, there is a limit on the total amount of property data that may be associated with any one element: the total length of all the strings, plus twice the number of attribute-value pairs, must not exceed 128 (or 512 of the element is an sref, aref, or node). For example, if a boundary element used a property attribute 2 with property value "metal", and property attribute 10 with property value "property", then the total amount of property data would be 18 bytes. This is 6 bytes for "metal" (odd length strings are padded with a null) plus 8 for "property" plus 2 times the 2 attributes (4) equals 18.

The following records are not supported by Stream Release 3.0:

BOX [2D00]	No data is present. This marks the beginning of a box element.
BOXTYPE [2E02]	Two-byte signed integer: contains the box type. The value of the boxtype must be in the range of 0 to 63.
PLEX [2F03]	Four-byte signed integer: a unique positive number which is common to all elements of the plex to which this element belongs. The head of the plex is flagged by setting the seventh bit; therefore, plex numbers should be small enough to occupy only the rightmost 24 bits. If this record is omitted, then the element is not a plex member.

Plex numbers are not commonly used.

BGNEXTN [3003] Four-byte signed integer: applies to pathtype 4. Contains four bytes which specify in database units the extension of a path outline beyond the first point of the path. The value can be negative.

EXDEXTN [3103] Four-byte signed integer: Applies to pathtype 4. Contains four bytes which specify in database units the extension of a path outline beyond the last point of the path. The value can be negative.

MASK [3706] ASCII string: Required for Filtered format, and present only in Filtered Stream files. Contains the list of layers and data types included in the data file (usually as specified by the user when generating the Stream file). At least one MASK record must follow the FORMAT record. More than one MASK record may follow the FORMAT record. The last MASK record is followed by the ENDMASKS record. In the MASK list, data types are separated from the layers witha semicolon. Individual layers or data types are separated with a space. a range of layers or data types is specified with a dash. An example MASK list looks like this:

1 5-7 10 ; 0-63

ENDMASKS [3800] No data is present. This is required for Filtered format, and is present only in a Filtered Stream file. This terminates the MASK records. The ENDMASKS record must follow the last MASK record. ENDMASKS is immediately followed by the UNITS record.

LIBDIRSIZE [3902] Two-byte signed integer: contains the number of pages in the Library directory. This information is used only when reading the data into a new library. If this record is present, it should occur between the BGNLIB record and the LIBNAME record.

SRFNAME [3A06] ASCII string: contains the name of the Sticks Rules File, if one is bound to the library. This informationis used only when reading the data into a new library. If this record is present, it should occur between the BGNLIB and LIBNAME records.

LIBSECUR [3B02] Two-byte signed integer: contains an array of Access Control List (ACL) data. There may be from 1 to 32 ACL entries, each consisting of a group number, a user number, and access rights. This information is used only when reading the data into a new library. If this record is present, it should occur between the BGNLIB and LIBNAME records.

The following record types are either not used, not released, or are related to tape formatting:

TEXTNODE	[1400]
SPACING	[18]
UINTEGER	[1D]
USTRING	[1E]
STYPTABLE	[2406]
STRTYPE	[2502]
ELKEY	[2703]
LINKTYPE	[28]
LINKKEYS	[29]
TAPENUM	[3202]

TAPECODE	[3302]
STRCLASS	[3401]
RESERVED	[3503]

2.9.5 Stream syntax in Bachus Naur representation

An element shown below in CAPITALS is the name of an actual record type. An element shown in lower case means that name can be further broken down in to a set of record types. The following table summarizes the Bachus Naur symbols:

Symbol	Meaning
: :	"Is composed of"
[ ]	An element which can occur zero or one time.
{ }	Choose one of the elements within the braces.
{ } *	The elements within the braces can occur zero or more times.
{ } +	The elements within braces must occur one or more times.
< >	These elements are further defined in the Stream syntax list.
|	"Or"

<stream format> ::=	HEADER BGNLIB [LIBDIRSIZE] [SRFNAME] [libsecur] libname [reflibs] [fonts] [attrtable] [generations] [<FormatType>] UNITS {<structure>}* ENDLIB
<FormatType> ::=	FORMAT | FORMAT {MASK}+ ENDMASKS
<structure> ::=	BNGSTR STRNAME [STRCLASS] {<element>}* ENDSTR
<element> ::=	{<boundary> | <path> | <SREF> | <AREF> | <text> | <node> | <box>} {<property>}* ENDEL
<boundary> ::=	BOUNDARY [EFLAGS] [PLEX] LAYER DATATYPE XY
<path> ::=	PATH [EFLAGS] [PLEX] LAYER DATATYPE [PATHTYPE] [WIDTH] [BGNEXTN] [ENDEXTN] XY
<SREF> ::=	SREF [EFLAGS] [PLEX] SNAME [<strans>] XY
<AREF> ::=	AREF [EFLAGS] [PLEX] SNAME [<strans>] COLROW XY
<text> ::=	TEXT [EFLAGS] [PLEX] LAYER <textbody>
<node> ::=	NODE [EFLAGS] [PLEX] LAYER NODETYPE XY
<box> ::=	BOX [EFLAGS] [PLEX] LAYER BOXTYPE XY
<textbody> ::=	TEXTTYPE [PRESENTATION] [PATHTYPE] [WIDTH] [<strans>] XY STRING
<strans> ::=	STRANS [MAG] [ANGLE]
<property> ::=	PROPATTR PROPVALUE

2.9.6 Example GDSII Stream file

The following is a dump of a minimal GDSII Stream file, consisting of just one

polygon (boundary). The GDSII file was created with the program DW2000 from Design Workshop. The binary dump was created on a VAX with the VMS command DUMP. The hex numbers are read backwards, from right to left, with each pair of digits representing a byte. Reading the first line below, we see that the file begins with the bytes 00 06 00 02, telling us that the first record contains 6 bytes, that the first record is type 00 (the header), and that record contains data of type 02 (two-byte signed integer). The corresponding ASCII representation on the right is read from left to right.

```
02000200 60000201 1C000300 02000600 ..........'.... 000000

01000E00 02000200 60002500 01000E00 .....%.'........ 000010

42494C45 4C504D41 58450602 12002500 .%....EXAMPLELIB 000020

413E0503 14000300 02220600 59524152 RARY.."".......>A 000030

1C00545A 9BA02FB8 4439EFA7 C64B3789 .7K...9D./..ZT.. 000040

60000000 01000E00 02000200 60000205 ...'..........' 000050

58450606 0C001100 01000E00 02000200 .............EX 000060

0100020D 06000008 04000045 4C504D41 AMPLE..... 000070

0000F0D8 FFFF0310 2C000000 020E0600 .....,..... 000080

FFFF204E 00001027 0000204E 00001027 '...N ..'...N .. 000090

0000F0D8 FFFFF0D8 FFFFF0D8 FFFFF0D8 ..... 0000A0

00000004 04000007 04000011 04001027 '..... 0000B0

00000000 00000000 00000000 00000000 ..... 0000C0
```

The following is an ASCII representation of this file created by the program SDUMP, [198] which translates the token numbers into the names listed in the previous section.

```
HEADER
3
BGNLIB
96
2
2
14
1
37
96
2
2
14
1
37
```

```
LIBNAME EXAMPLELIBRARY
GENERATIONS
3
UNITS
1.0000000000000E-03
1.0000000000000E-09
BGNSTR
96
2
2
14
1
0
96
2
2
14
1
17
STRNAME EXAMPLE
BOUNDARY
LAYER
1
DATATYPE
0
XY
-10000
10000
20000
10000
20000
-10000
-10000
-10000
-10000
10000
ENDEL
ENDSTR
ENDLIB
```

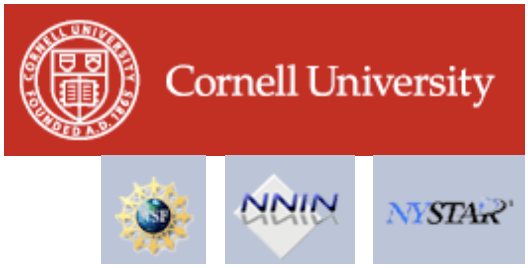
Table of Contents
Previous section: 2.8 Acknowledgements
Next section: 2.10 References

Please send us your comments, corrections, and submissions.

Order the complete book from SPIE

Back to Top

Search GO select keyword GO



Cornell NanoScale Science & Technology Facility (CNF)
250 Duffield Hall, Cornell University, Ithaca, New York 14853-2700
Voice: 607-255-2329, Fax: 607-255-8601, Email: information@cnf.cornell.edu
Design and Programming by Spider Graphics Corporation®