EE Times Connecting the Global Electronics Community

Register | Login

Home | News | Opinion | Messages | Authors | Video | Slideshows | Teardown | Education | EELife | Events

About Us | Newsletter Sign Up

Android | Automotive | Embedded | Industrial Control | Internet of Things | MCU | Medical | Memory | Open Source | PCB

BREAKING NEWS NEWS & ANALYSIS: Hock Tan Carves Up Brocade

designlines SoC

Design How-To

Going from GDSII to OASIS

Dr Philippe Morey-Chaisemartin, Xyalis

8/4/2008 01:00 PM EDT

Post a comment

NO RATINGS
1 saves
LOGIN TO RATE

Like 2 Tweet Share G+1 1

Why switching to OASIS ?

It's a banality to say that nowadays, databases for digital chips are more than huge. The physical description of an SOC, encoded in the classical GDSII format, now often goes over 20Gbytes. Files of up to 200Gbytes have been reported by mask houses. Even if storage systems and data transfer links can handle such sizes, it is obvious that such big files are difficult to manipulate.

GDSII was introduced by Calma in 1978 as a successor of GDS format created in 1971. Since almost 30 years, no major change have been made to this de-facto standard while chips complexity was multiplied by as much as 10^6. In addition to file size issue, numerical values needed to describe geometries of nanoscale structures on 300mm wafers will soon reach the 32 bits limits of GDSII format.

The OASIS format was developed to address such issues and its first official specification was released in 2004 [1].

This article describes how size and precision limitation issues are managed in OASIS format. It also singles out some critical points of this format and finally gives some ways to really get full benefits from OASIS and to circumvent potential pitfall and problem using this new standard.

How data size reduction works

The primary goal of the OASIS format is to reduce the data base size. This can be done in multiple ways: optimization of the file structure, suppression of all redundancies and compaction of the values.

1) Reduction of geometric description size
All geometries contained in the physical description of a chip are made of polygons, themselves described as lists of coordinates. The optimization can be done by reducing the number of coordinates needed to describe a polygon, and by reducing the size (in terms of used bytes) of each coordinate.

- Numeric values
  As the other goal of OASIS is to remove some precision limitation of numeric values, reducing the size seems incompatible. In fact, OASIS stores all numeric values with variable length encoding. The numeric values are split in bytes of 7 bits. The eighth bit is used to signify that an additional byte is needed. By this method a small value will only use 1 byte, while a big value could use 4 or more

Most Recent Comments

ETisME the US recovered only because the USD remained strong, foreign countries using USD as reserve do not want to see the reserve dissappear, this is why US can continue running on...

11/4/2016
10:06:29 PM

Navigate to Related Links

ASML Invests $1.9B in Next-Gen EUV

U.S. Needs Big Bet on Chips

Anticipating a More Virtual Moore's Law

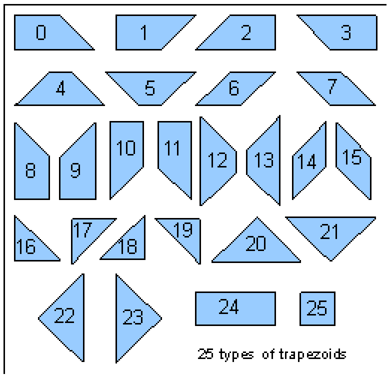
AI Chip Startup Shares Insights

bytes. This brings two advantages:
- first, statistically, most of the values are small enough to use less than 4 bytes,
- second, there is no limitation -at least- in the standard. A value may have an infinite precision.

- Polygons
  Each polygon is described as a list of coordinates. In GDSII, all coordinates are made of a pair of X and Y absolute values. In OASIS, as small values use less space, each coordinate may be considered as relative to the previous one. As most of the geometries are made of small polygons (compared to chip or wafer size), describing polygons with relative coordinates dramatically reduce data size.
  Additionally, most of the polygons have standard shapes: squares, rectangles, trapezoids. In GDSII format, there are no specific description for basic shapes: a polygon description starts at a point, follows each point with its X and Y coordinate and ends with the coordinates of the starting point. A simple square then needs five points, each of them needing two values (X and Y).
  In OASIS, a square is identified by one point and its size. We then have only 3 values (on which one is almost always small) compared to 10 needed in GDSII format. This only requires to identify a square from any other shape.
  In the same way, rectangles or trapezoids are identified specifically: no less than 25 different types of trapezoids can be described. Each of them will then use the minimum number of values for its full description. At this step, we should point out some particularities of OASIS format which allow to describe a rectangle as a rectangle or as a specific type of trapezoid .

- Layers
  The layer and data type values are also included in the optimization of geometries description. In GDSII, each polygon description includes the layer and the data type number. In OASIS, these values are specified only if different from the previous value (as it is made in CIF format).

  It should be noticed, that, as for other numeric values, layer and data type numbers may have an infinite precision, so the 256 values restriction of GDSII is eliminated and the new format can accommodate all the layers needed by advanced process description.



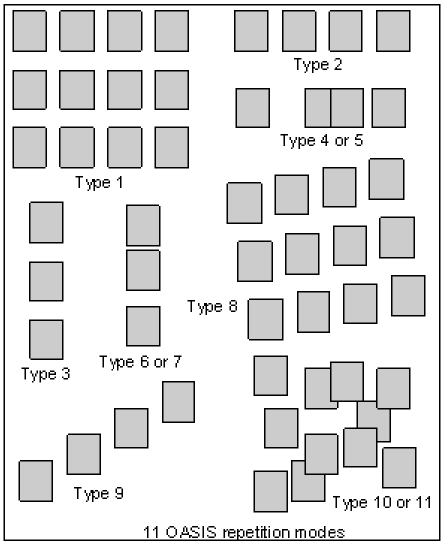
1. OASIS can support up to twentyfive types of trapezoids.

2) Optimization of geometric repetitions
Statistically, in any design many geometries are repeated. For example a simple contact may appear tens of time in a single small library cell. OASIS offers the possibility to instantiate multiple occurrence of the same geometry [2].

- Regular arrays
  As in GDSII for matrix of cells, the basic repetition mode is a regular array.

- Random distribution
  In addition to regular arrays, OASIS offers the possibility to instantiate the random distribution of the same polygon. In

this case, such polygon description is followed by the displacement to the first point of next identical polygon.



2. There are 11 ways to describe repetition in OASIS.

3) Optimization of cells call
A physical description of any chip is always hierarchical. A top cell call sub cells which are described separately.

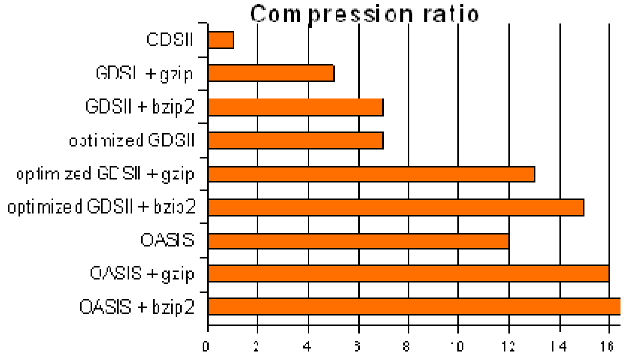
- Reference
  In the OASIS format, it is possible to make a reference to a cell through different methods. This includes reference by name (as in GDSII format) and reference by index. In the same way, when a cell is declared, different methods of declaration are allowed: declaration by name, declaration by index and automatic numbering of indexes. When a declaration by index is made, it references a line in a table which can be stored either at the beginning or at the end of the file.

- Multiple instantiation
  This is a significant improvement from GDSII. Arrays have been extended to non orthogonal matrix of cells. This kind of structures have been introduced to instantiate dummy tiles to improve CMP yield during manufacturing. If no special care is taken during this generation, the resulting GDSII database may dramatically increase, while the OASIS database remains much smaller. The second possibility offered by OASIS is to specify multiple placement of one cells by just giving the position of each instance in the shortest possible way.

4) Embedded compression
An other possibility offered by OASIS is to directly compress (gzip like method) some blocks inside the file. Usually a block is a full cell description. Each cell is then independently compressed, which makes random access in the file possible even if it has components in compressed format.

5) Performances
Depending on the database structure and on the chosen optimization, an OASIS file is between 5 and 20 times smaller than a GDSII [3][4].

That's a big improvement, but we need to compare these values on compressed files which is now a standard. The following diagram, gives some average compression ratio compared to the original GDSII file which has been given the reference value of 1.



3. Comparison betwqeen GDSII and OASIS with various compression methods.

The optimized GDSII data is obtained by replacing each repeated polygon by a cell containing this polygon and multiple calls to the cell. Cell names are chosen as short as possible to reduce file size as references are only made though names in GDSII [5].

It should be noticed that the compression with gzip -or bzip2- on an OASIS file is less efficient than on GDSII. This mostly comes from the fact that all numerical values are already compacted, i.e. reduced to the minimum number of bytes thanks to variable size coding. All unnecessary "0" bytes present in GDSII format (almost 50% of the file) are already removed from OASIS before this compression scheme is applied.

Potential problems
Although OASIS offers capabilities needed by new technologies and highly optimizes database size, it's far from being free of issues.

1) No restrictions means no limit
The first dramatic impact of having removed all the restrictions due to precision limits (i.e. 32 bits length for coordinates) is that anything is allowed. Any value can have an infinite precision. This is an interesting feature for fundamental mathematics but as no meaning to describe a circuit.
Description of a value is one thing, computation on infinite precision values is something else. All the tools which manipulate OASIS files will have an internal limit (due to hardware architecture). This makes them not 100% OASIS compliant even if they will be able to handle all OASIS files which should never use values of more than 64bits.

If we consider than 10^3 is almost the same as 2^{10}, a 32 bits value can describe a coordinate of +/- 2.10^9, a precision of 0.1 nm on a 20cm wafer. We are close to the limit of current process needs, but with 64 bits we are far from all future expected limits.

Adding an internal limit to coordinates at 64 bits is for sure safe, but some tools running on 32 bits architectures may have a limit at 32 bits. This makes a file created on a 64 bits platform unreadable on a 32 bits platform or worst of all, readable but introducing overflows and then converting positive coordinates into negative ones.
The risk is not very high with coordinates, but becomes dramatic for other integer values such as cell index or layer numbers as they can't be manage on standard computer architectures.

2) Tables and indexes
As described above, all the cells may be referenced through indexes. This index is an entry in a table containing cells name. This makes referencing quite easy, except that references may be stored at different places: beginning of the file, end of the file, or spread among the whole file. Worst of all, references can be also made by name. Even if all the combinations cannot be mixed in the same files, all the different possibilities exist. So an OASIS reader should be able to accept any kind of reference and must not be optimized for an option or the other.

It appears that a commonly used solution is to build the reference table at the end of the file. This makes an OASIS writer quite easy to manage. TheOASIS standard states that it is very convenient, while reading an OASIS file, for the position of this table (when present) to be at a fixed position, preferably at the end of the file. This should made its access, prior to full file parsing, very easy. That's true when the file is not compressed. Unfortunately, most of the users still compress their files in order to minimize the size of the database.

Uncompressing a file can only be done sequentially so, with GDSII format, which was originally developed to be read and written on tapes, there is no problem. The OASIS format uses the fact that all storage is now performed on random access media and allows direct access to any location in the file. When compression is used, if the OASIS reader uses this feature, it has a dramatic impact on read access time.

3) Equivalence with GDSII
OASIS is intended to replace the GDSII format, but still for many years, both format will co-exist. So managing heterogeneous environments and translating data between GDSII and OASIS is mandatory and will remain a constraint for many years.

- Layers
  The number of usable layers (256 layer numbers X 256 data types) is an important limitation of GDSII. OASIS has removed this limitation, and the layer number is now unlimited. This makes OASIS to GDSII conversion very difficult to manage. We can expect that the restrictions of the GDSII will remain a de-facto limit of OASIS files. It should be noticed that some CAD vendors have already removed this limitation by accepting layer numbers greater than 256, but this is an unofficial extension of GDSII format and is not supported by all the tools.

- Circles
  The possibility to describe circles is another added feature of OASIS. This is not possible in GDSII where circles must be approximated by polygons. Depending on how the polygon is generated (number of edges, position of the first point), the resulting GDSII will differ from the original OASIS.

  If we also consider that OASIS is dedicated to mask building, no mask writing equipment is able to make a circle without approximating it though polygons. This will make mask to database inspection a real problem.

4) Vulnerabilities
Despite its enhancement compared to GDSII, OASIS format may still contain inconsistent data. Usage of a checksum at the end of the file reduces the problem of data corruption during transfers but OASIS standard by itself doesn't specify how to interpret specific shapes. Worst of all, OASIS files may contain unidentified binary data.

- Viruses
  The possibility to insert any piece of binary code inside a file is a major issue in OASIS format. It is possible to define any property which can contain binary data. There is no restriction regarding its size nor its content. It then becomes possible, if not easy, to propagate some piece of code like viruses, trojans or worms in such a file. While the file by itself remains specs compliant.

  The OASIS file is not auto-executable, but there are already some case of viruses which have been propagated though pure data file thanks to readers security lacks. This is made easily in an OASIS file since there are no limits regarding size, so using overflow methods to corrupt readers memory and code integrity may be an interesting challenge for some hackers.

  When we know, that almost all chip databases represent sensitive significant data, sending a malicious OASIS file to corrupt a system security may be an underhand method for industrial espionage.

- Bad polygons
  The OASIS format has the same limitations as GDSII in terms of polygons shapes. There are no constraints on the allowed polygon. This is not directly a data format specification but may lead to different behaviors depending on the tool. For example, the following configurations may be encountered :
    - twisted polygons
    - self intersecting polygons
    - U-turns in path descriptions

  These shapes are syntactically correct but may be interpreted in different ways. This is a major issue in GDSII and too many chips were born dead due to such configurations. This should have been specified in the OASIS format but was not and still needs to be checked carefully before manufacturing the reticle.

How to get real benefits from OASIS format
Here are some basic rules when using a GDSII to OASIS converter or developing your own OASIS writer:

- Always reference cells by index. It appears that some files generated after OPC processing contain millions of cell. Referencing cells by name in such configuration will have a dramatic impact while parsing the file.
- Avoid compressing the file. Even if some readers like the one from Xyalis are able to directly analyze compressed file without seeking in the file, most of the tools will be slowed down due to index access requirements. It is much more efficient to use embedded gzip feature.
- Keep layer numbers; index and references in the limits allowed by GDSII format.
- Carefully analyze included binary code for viruses.

Conclusion
The OASIS format lifts the restriction on precision in numbers but doesn't correct all the limitations of GDSII and brings some new sources of errors and problems.

Depending on the method used for the optimization, the results in terms of file size and of analysis time may vary significantly. Many different ways of optimization are available but none of them can give the best result on any type of database. It's almost impossible, or it may cost too much time, to try all the methods and choose the best. So each CAD vendor will define a strategy and will generate its OASIS file by using a given method.

Some companies are starting to switch to OASIS format, while others remain on the GDSII format. For them, the only real issue related to GDSII is the file size, which is not considered as a blocking point. Extending disk and RAM capacities is still estimated to be a better deal than changing a qualified flow based on GDSII to a new one based on OASIS.

Due to the complexity of the OASIS standard, and to the fact that many different options are available to store the same data, the number of possible errors in an OASIS file dramatically increases compared to GDSII (at least 4 times). If we also consider that the weakness of GDSII regarding polygons shape interpretation has not been corrected in OASIS, it seems important to carefully validate all the databases using this new format.

Experience has proved that it took many years to correct all the errors in GDSII file generated by different tools. We are just at the beginning of OASIS, so detailed checks should be performed in order to achieve the same level of confidence than for existing GDSII based flows.

After many years developing tools based on GDSII format, Xyalis has released a OASIS format reader. It allows to check all critical points in an OASIS file including full specification compliance. It also validates the compatibility among 32/64 bits platforms, badly formed polygons, checks for the presence of unidentified binary code and more.

References
1) SEMI P39-1105 - OASIS -- Open Artwork System Interchange Standard. Abstract IEEE standards

2) Evaluation of the New OASIS Format for Layout Fill

2) Evaluation of the New OASIS Format for Layout Fill Compression Yu Chen & al.
3) GDSII to OASIS Converter " Performance and Analysis, Nageswara Rao G., Softjin Technologies, white paper.
4) OASIS vs. GDSII stream format efficiency, A.Reich & al., Proceedings of SPIE -- Volume 5256
5) Improved file sizes and cycle times through optimization of GDSII Stream, Chin Le & al., Proceedings of SPIE -- Volume 5992

About the author
Philippe Morey-Chaismartin is Xyalis' CTO. He earned a master in computer science in 1983 and a PhD in microelectronics in 1986. He can be reached at: his Xyallis address.

EMAIL THIS PRINT COMMENT

More Related Links

ASML Invests $1.9B in Next-Gen EUV
U.S. Needs Big Bet on Chips
Anticipating a More Virtual Moore's Law
AI Chip Startup Shares Insights
U.S. Convenes Chip Study Group

Comments

VIEW COMMENTS: **NEWEST FIRST** | OLDEST FIRST | THREADED VIEW

Be the first to post a comment regarding this story.

GLOBAL NETWORK EE Times Asia | EE Times China | EE Times Europe | EE Times India | EE Times Japan | EE Times Korea | EE Times Taiwan | EDN Asia | EDN China | EDN Japan | ESC Brazil

UBM Communities
EE Times | EDN | EBN | DataSheets.com | Embedded | TechOnline | Design News | DesignCon | ESC
Working With Us: About | Editorial Policies | Contact Us | Media Kits | Reprints