

# GDSII to OASIS Converter – Performance and Analysis

Nageswara Rao G  
8 November 2004

## 1 Introduction

For more than three decades GDSII has been the de-facto standard format for layout design data. But for some time it has been felt that the GDSII's inefficient data representation makes it inadequate for modern real-life designs. With layout data volume growing non-linearly owing to RET, the deficiencies of GDSII are getting highlighted more than ever before.

To address the issue of handling huge layout data, Semiconductor Equipment and Materials International (SEMI) designed a new layout file format. This format, Open Artwork System Interchange Standard (OASIS), should represent hierarchical layout data in less than a tenth of the space taken by GDSII.

SoftJin Infotech has developed tools and libraries to read and write GDSII and OASIS files, and a translator from GDSII to OASIS. SoftJin is releasing the source code of these tools and libraries for free download. The objective is to promote the early adoption of OASIS.

This white paper discusses the most significant tool of this release, the GDSII-to-OASIS converter. Section 2 explains why OASIS files are smaller than their GDSII equivalents. Section 3 briefly describes the GDSII-to-OASIS converter's options. Section 4 gives the benchmark results, on both real-life industrial data and academic design data, with different options. Section 5 gives our conclusions.

## 2 GDSII and OASIS

OASIS has several features that make layout files smaller than the equivalent GDSII files. The following is an overview of those features.

The most significant is the repetition specification that may be used for an element type. Repetitions are better than the AREF elements of GDSII in two ways. First, they can be used for any element type, not just for structure references. Second, repetitions can come in several types. Some represent special cases—such as elements in a uniformly spaced horizontal line—compactly. The most general repetition type is a list of deltas; using this, a single record can describe all the copies of an element in a cell, wherever they are placed.

SoftJin Infotech Pvt. Ltd

India: 227/70/A, Sigma Arcade, Marathahalli, Bangalore 560 037, India, Tel: +91-80-25234641, Fax: +91-80-25234643

Branch Off: 2900 Gordon Ave, Suite 100-11, Santa Clara, CA 95051, USA, Tel: (408) 773-1714, Fax: (408) 773-1745

Email: [anuvad-info@softjin.com](mailto:anuvad-info@softjin.com) Web: [www.softjin.com](http://www.softjin.com)

Compression using zlib is included in the file format. Each cell can be compressed separately. This gives almost as good results as compressing the entire file while allowing random access to each cell.

Many element fields like layer, datatype, and repetition are made *modal variables*. When defined in some element, they apply to the following elements (and thus may be omitted) until they are redefined.

OASIS uses variable-length integers with a continuation bit in each byte instead of fixed length 16-bit and 32-bit integers. This allows it to represent smaller values using fewer bytes and makes it possible to store arbitrarily large values.

In GDSII, path and polygon elements contain the coordinates of each point. In OASIS, the elements contain the coordinates of the first point and deltas between adjacent points. Since the deltas are usually much smaller than the coordinates, they take up less space. For Manhattan and octangular (45 degree) shapes OASIS stores the delta even more compactly, as a single integer.

OASIS stores each element in a single record and does away with record header overhead of GDSII. It also defines new element types RECTANGLE and TRAPEZOID to store these more compactly than general polygons.

OASIS defines *name records* that assign a reference number to a cell name or text string. Other records can then use the (shorter) reference number instead of the name or string.

### 3 Converter Options

The GDSII-to-OASIS converter, *gds2oasis*, has two options whose effects we analyze. The compression option `-z` controls whether cells are compressed using zlib. In OASIS terms the contents of each cell are put into a CBLOCK record.

The optimization option `-O` controls the kind of repetition specifications generated. With `-O1`, *gds2oasis* uses only the most general (and bulkiest) repetition type for all repetitions—a list of deltas, perhaps with a grid multiplier. It sorts all the positions in the repetition so that regular patterns in the positions appear as sequences of identical deltas. With `-O2`, *gds2oasis* looks for uniformly spaced lines and arrays of instances, and uses more compact forms of repetitions for those. Only lines and arrays parallel to the axes are recognized, and extra points in the middle of the pattern prevent the pattern from being recognized. We feel that it is not worthwhile to write sophisticated pattern recognizers because, as the data below suggest, zlib does a good job of recognizing and compressing regular patterns.

## 4 Performance Data and Analysis

We tested the performance of *gds2oasis* using six different sets of GDSII files from different sources.

1. Set A – Real-life designs in hierarchical GDSII from IC design company A
2. Set B – Real-life benchmark designs in hierarchical GDSII from foundry B
3. Set C – Real-life designs in hierarchical GDSII from foundry C
4. Set D – Real-life designs in flattened GDSII from company D
5. Set E – Assorted design data from academic institutions
6. Set F – Assorted design data in hierarchical GDSII from different companies

Sets A, E and F contained only Manhattan geometries. Around 5-10% of the geometries in sets B and C were any-angle, and over 50% of the geometries in set D were any-angle.

Tables 1 through 3 show the results of running *gds2oasis* on the ten largest GDSII files in all sets, with different combinations of options.

No.	Set	GDSII size	-O1		-O2	
			OASIS size	Ratio	OASIS size	Ratio
1	A	174,744,638	9,738,051	17.9	512,495	341.0
2	E	263,643,136	2,067,231	127.5	447,274	589.4
3	C	337,809,266	16,777,216	20.1	8,126,464	41.6
4	C	418,628,412	9,027,640	46.4	8,855,232	47.3
5	C	719,812,025	55,995,830	12.8	55,646,034	12.9
6	A	1,039,144,960	37,209,885	27.9	14,782,967	70.3
7	B	1,256,468,671	181,508,180	6.9	179,446,881	7.0
8	B	1,258,752,308	182,039,370	6.9	179,971,967	7.0
9	C	1,262,421,268	152,742,437	8.3	151,614,368	8.3
10	A	2,649,387,008	163,263,992	16.2	146,365,453	18.1
<b>TOTAL</b>		<b>9,380,811,692</b>	<b>810,369,832</b>	<b>11.6</b>	<b>745,769,135</b>	<b>12.6</b>

**Table 1: Conversion ratio (without embedded compression)**

Table 1 shows the result of converting without zlib compression. The **Ratio** columns show the conversion ratio—the ratio of GDSII file size to OASIS file size. This ranges between 7 and 589. The last line of the **Ratio** columns is the weighted average of all ratios—the sum of all the GDSII file sizes divided by the sum of all the OASIS file sizes. Note that files generated using –O2 are, as expected, always smaller than the files generated using –O1, though the difference is frequently slight.

No.	Set	GDSII size	-O1 -z		-O2 -z	
			OASIS size	Ratio	OASIS size	Ratio
1	A	174,744,638	276,483	632.0	239,475	729.7
2	E	263,643,136	220,314	1196.7	174,255	1513.0
3	C	337,809,266	4,492,288	75.2	4,392,960	76.9
4	C	418,628,412	4,302,099	97.3	4,298,872	97.4
5	C	719,812,025	23,431,756	30.7	23,433,504	30.7
6	A	1,039,144,960	7,156,634	145.2	6,977,915	148.9
7	B	1,256,468,671	84,759,308	14.8	84,918,480	14.8
8	B	1,258,752,308	85,117,889	14.8	85,277,105	14.8
9	C	1,262,421,268	77,271,312	16.3	77,352,228	16.3
10	A	2,649,387,008	116,250,078	22.8	117,042,786	22.6
<b>TOTAL</b>		<b>9,380,811,692</b>	<b>403,278,161</b>	<b>23.3</b>	<b>404,107,580</b>	<b>23.2</b>

**Table 2: Conversion ratio (with embedded compression)**

Table 2 shows the same set of files, but with zlib compression enabled. Conversion ratios now range from about 15 to 1513. With compression enabled there is little difference between -O1 and -O2 for most files. For some files -O2 -z actually gives worse results than -O1 -z. We think that this is because the different types of repetition that -O2 introduces destroys some of the regularities in the file that zlib exploits.

No.	GDSII size	Conversion Ratio			Conversion Ratio		
		-O1	-O1 -z	zlib	-O2	-O2 -z	zlib
1	174,744,638	17.9	632.0	35.3	341.0	729.7	2.1
2	263,643,136	127.5	1196.7	9.4	589.4	1513.0	2.6
3	337,809,266	20.1	75.2	3.7	41.6	76.9	1.8
4	418,628,412	46.4	97.3	2.1	47.3	97.4	2.1
5	719,812,025	12.8	30.7	2.4	12.9	30.7	2.4
6	1,039,144,960	27.9	145.2	5.2	70.3	148.9	2.1
7	1,256,468,671	6.9	14.8	2.1	7.0	14.8	2.1
8	1,258,752,308	6.9	14.7	2.1	7.0	14.8	2.1
9	1,262,421,268	8.2	16.3	2.0	8.3	16.3	2.0
10	2,649,387,008	16.2	22.8	1.4	18.1	22.6	1.3
<b>TOTAL</b>	<b>9,380,811,692</b>	<b>11.6</b>	<b>23.3</b>	<b>2.0</b>	<b>12.6</b>	<b>23.2</b>	<b>1.8</b>

**Table 3: Improvement in conversion ratio due to embedded compression**

Table 3 shows zlib's contribution to the conversion ratio. Each column labeled **zlib** is the ratio of the preceding two columns. Note that zlib compression doubles the conversion ratio for both -O1 and -O2. We also tried compressing with gzip the OASIS files generated using -O1 and -O2. The sizes of these compressed files were nearly the same as the sizes of the files generated with -z. Evidently the cells are large enough that compressing cells individually is as good as compressing the whole file.

Tables 4 through 6 show results for the ten files for which we got the highest compression ratios. These results are not really significant. We give them only as examples of the spectacular improvements over GDSII that OASIS can yield. Table 4 shows the results without zlib compression, Table 5 shows the results with compression, and Table 6 shows zlib's contribution.

No.	Set	GDSII Size	-O1		-O2	
			OASIS size	Ratio	OASIS size	Ratio
1	B	84,249,047	5,366,257	15.7	707,933	119.0
2	B	173,631,345	9,738,051	17.8	512,495	338.8
3	F	65,089,536	2,226,559	29.2	225,016	289.3
4	D	8,964,096	35,322	253.8	28,739	311.9
5	D	3,991,552	14,717	271.2	9,206	433.6
6	E	263,643,136	2,067,231	127.5	447,274	589.4
7	D	1,751,040	4,721	370.9	1,093	1602.0
8	D	8,955,904	26,734	335.0	18,382	487.2
9	F	32,208,896	991,205	32.5	67,688	475.8
10	C	23,911,676	2,439,770	9.8	33,703	709.5

**Table 4: Best results obtained (without embedded compression)**

No.	Set	GDSII Size	-O1 -z		-O2 -z	
			OASIS size	Ratio	OASIS size	Ratio
1	B	84,249,047	355,797	236.8	334,712	251.7
2	B	173,631,345	276,483	628.0	239,475	725.0
3	F	65,089,536	138,897	468.6	81,023	803.4
4	D	8,964,096	6,562	1366.1	8,895	1007.8
5	D	3,991,552	2,231	1789.1	2,722	1466.4
6	E	263,643,136	220,314	1196.7	174,255	1513.0
7	D	1,751,040	1,021	1715.0	991	1766.9
8	D	8,955,904	3,259	2748.0	4,890	1831.5
9	F	32,208,896	35,691	902.4	16,555	1945.6
10	C	23,911,676	22,677	1054.4	8,833	2707.1

**Table 5: Best results obtained (with embedded compression)**

No.	GDSII size	Conversion ratio			Conversion ratio		
		-O1	-O1 -z	zlib	-O2	-O2 -z	zlib
1	84,249,047	15.7	236.8	15.1	119.0	251.7	2.1
2	173,631,345	17.8	628.0	35.2	338.8	725.0	2.1
3	65,089,536	29.2	468.6	16.0	289.3	803.4	2.8
4	8,964,096	253.8	1366.1	5.4	311.9	1007.8	3.2
5	3,991,552	271.2	1789.1	6.6	433.6	1466.4	3.4
6	263,643,136	127.5	1196.7	9.4	589.4	1513.0	2.6

7	1,751,040	370.9	1715.0	4.6	1602.0	1766.9	1.1
8	8,955,904	335.0	2748.0	8.2	487.2	1831.5	3.8
9	32,208,896	32.5	902.4	27.8	475.8	1945.6	4.1
10	23,911,676	9.8	1054.4	107.6	709.5	2707.1	3.8
<b>TOTAL</b>	<b>666,396,228</b>	<b>29.1</b>	<b>626.9</b>	<b>21.5</b>	<b>324.8</b>	<b>763.9</b>	<b>2.3</b>

**Table 6: Improvement in conversion ratio due to embedded compression**

In Table 6, note that zlib's average contribution to the conversion ratio is 21.5 for -O1. Compare this with corresponding value 2.0 in Table 3. The reason for the difference is clear from Table 4. These files contain large repetitions with points in regular patterns. The pattern-recognition code that *gds2oasis* uses with -O2 detects these regular patterns and uses compact repetition types for those. That is why -O2 gives much better results than -O1. But because *gds2oasis* sorts the points when it uses the general form of repetition (a list of deltas), regular patterns in the points appear as sequences of identical deltas. Zlib of course compresses such sequences very well, so -O1 -z also gives much better results than -O1.

With -O2, zlib yields only a factor of 2, compared with 21.5 for -O1. This too is understandable. The -O2 code has removed most of the redundancy from the repetitions, so all that is left for zlib to work with is the redundancy in the remaining element data.

Tables 7 and 8 summarize the results for all files in all sets. Table 7 shows the results without compression and Table 8 shows the results with compression.

Set	Number of files	Total GDSII size	-O1		-O2	
			Total OASIS size	Ratio	Total OASIS size	Ratio
A	6	3,891,154,026	212,688,880	18.3	171,812,046	22.6
B	22	5,043,355,648	672,785,180	7.5	636,906,990	7.9
C	80	8,367,198,208	945,860,220	8.8	895,588,335	9.3
D	198	676,040,704	83,636,620	8.1	82,767,089	8.2
E	5	304,172,826	3,060,143	99.4	798,530	380.9
F	7	111,169,536	4,758,833	23.4	1,700,868	65.4
<b>ALL</b>	<b>318</b>	<b>18,393,090,948</b>	<b>1,922,789,876</b>	<b>9.6</b>	<b>1,789,573,858</b>	<b>10.3</b>

**Table 7: Summary of performance testing (without embedded compression)**

Table 7 gives the conversion ratios on all the 318 test cases without embedded compression. The average conversion ratio is about 10. Note that sets A, E, and F, which contain only Manhattan geometries, yield better results than sets B, C, and D, which contain any-angle geometries. We are unable to examine the files, so we cannot say exactly why this happens. The most likely reason is that the Manhattan-only files have many rectangles in a few standard sizes.

Set	Number of files	Total GDSII size	-O1 -z		-O2 -z	
			Total OASIS size	Ratio	Total OASIS size	Ratio
A	6	3,891,154,026	131,702,483	29.5	132,331,661	29.4
B	22	5,043,355,648	317,841,152	15.9	318,194,011	15.8
C	80	8,367,198,208	439,638,130	19.0	439,849,427	19.0
D	198	676,040,704	12,409,867	54.5	12,523,982	54.0
E	5	304,172,826	368,286	825.9	348,288	873.3
F	7	111,169,536	1,245,703	89.2	1,180,638	94.2
<b>ALL</b>	<b>318</b>	<b>18,393,090,948</b>	<b>903,205,621</b>	<b>20.4</b>	<b>904,428,007</b>	<b>20.3</b>

**Table 8: Summary of performance testing (with embedded compression)**

Table 8 gives the conversion ratios on all the 318 test cases with embedded zlib compression. The average conversion ratio is about 20 with both -O1 and -O2.

## 5 Conclusion

On the whole there is a large difference in the converter's performance for Manhattan geometries and for any-angle geometries. For Manhattan geometries the conversion ratio is over 20, but for any-angle geometries the ratio is only about 8. Apparently files with only Manhattan geometries are more likely to have repeated elements than files with any-angle geometries.

Compressing cells individually (embedded compression) with zlib provided another factor of compression of around 2. The external compression through gzip also gave similar factor of compression of 2. Our average compression factor of 2 for embedded as well as external compression is significantly less than the average compression factor of 4 for external compression reported in [1]. We do not know whether this difference is due to the data set or the conversion algorithm.

When cells are compressed using zlib, it might be counter-productive to recognize uniformly spaced arrays and lines and generate special repetition specifications for those. Using those special repetitions frequently makes the cells less compressible, and hence larger, than using a single repetition type of a list of deltas.

## 6 References

- [1] Steffen Schulze, Pat LaCour, Laurence Grodd. OASIS-based data preparation flows: Progress report on containing data size explosion.  
<http://www.mentor.com/dsm/techpapers/>