

Corner2-EPC: A Layout Image Compression Algorithm for Electron Beam Lithography

N. Chaudhary, Y. Luo, S. A. Savari¹

*Department of Electrical & Computer Engineering, Texas A&M University
savari@ece.tamu.edu*

The widespread use of electron-beam direct-write (EBDW) lithography has been hindered by its low throughput in the patterning of wafers. In order to address this issue, future EBDW lithography systems must transmit terabits of information per second, and parallelism and lossless data compression are techniques which can be used together to tackle the data transfer problem^{2,3}. This would involve a datapath system in which compressed layout images are cached in storage disks and sent to the processor memory board. To satisfy the throughput requirements, the decoder embedded within the array of electron beam writers must be able to rapidly recover the original layout images from the compressed files.

The Block C4 algorithm⁴ (BC4) is intended for a raster scanning system in which each pixel is quantized to one of 32 levels. The algorithm is based on a two-dimensional version of a Lempel-Ziv (LZ) code⁵. Carroll et al.⁶ incorporated a simpler lossless data compression scheme in the digital pattern generator of the multiple electron beam lithography system REBL⁷ where each pixel takes on one of 32 values. The scheme of Carroll et al. more closely resembles a one-dimensional version of an LZ code. Carroll et al. reported that the compression ratio and throughput of the decoder needed were not consistently achieved.

The lossless compression scheme Corner2⁸ for binary layout images on raster-scanning systems was motivated by the GDSII format. Corner2 greatly outperformed BC4 in compression ratio, encoding/decoding times and memory usage, but it did not account for the characteristics of electron-beam proximity corrected layout images. We propose a scheme Corner2-EPC which extends the image transformation technique of Corner2 to handle pixels which can be quantized to an arbitrary number of gray levels; we used 32. The experimental results in Table 1 are for a 25-layer image compression block based on the FREEPDK45 45nm library with a minimum element of 60 nm. The experiments were performed on Intel i7-2600 CPU processors at 3.40 GHz with 8 GB of RAM using a Windows7 Enterprise operating system and the electron beam proximity correction algorithm of GenISys, Inc., BEAMER_v4.6.2_x64. To perform the experiments for BC4 it was necessary to split each layout image into four segments and run the algorithm on each segment; the Corner2-EPC experiments were on full layout images. The input to both algorithms is in .png format, which initially represents the value of each pixel with eight bits instead of five. Corner2-EPC always considerably outperforms BC4 in terms of encoding/decoding speed and memory requirements. On most layers Corner2-EPC also outperforms BC4 in compression ratios.

Acknowledgments: This work would not have been possible without the support of GenISys, Inc. The authors also thank D. Lie and S. Mukhopadhyay for their circuit data and V. Dai for providing an implementation of the Block C4 algorithm.

1. This work was supported in part by NSF grant ECCS-1201994.
2. H. J. Levinson, *Principles of Lithography* (SPIE, Bellingham, 2010), Third Edition, pp. 466-467.
3. V. Dai and A. Zakhor, *IEEE Trans. Image Process.*, **15**, 2522-2530, (2006).
4. H. Liu, V. Dai, A. Zakhor, and B. Nikolic, *SPIE J. Micro/Nanol., MEMS, MOEMS*, **6**, 013007, (2007).
5. T. C. Bell, I. H. Witten, and J. Cleary, *Text Compression*, (Prentice Hall, New Jersey, 1990).
6. A. Carroll et al., *Proc. SPIE* **9049**, 904917, (2014).
7. P. Petric et al., *Proc. SPIE* **7271**, 727107, (2009).
8. J. Yang and S. A. Savari, *SPIE J. Micro/Nanolithogr., MEMS, MOEMS*, **10**, 043007, (2011).

Table 1. Experimental results for a 25-layer image compression block

Layer	Corner2-EPC Compression Ratio	Block C4 Compression Ratio	Ratio of Corner2-EPC Encoding Time to Block C4 Encoding Time	Ratio of Corner2-EPC Decoding Time to Block C4 Decoding Time
1	1,690.3	909.6	0.00152	0.168
2	64,367.8	1,339.7	0.00123	0.224
3	327.9	436.4	0.00215	0.240
4	347.5	503.0	0.00191	0.185
5	335.1	308.1	0.00238	0.230
6	6,229,328.2	1364.0	0.00136	0.223
7	1,575,966.8	1363.6	0.00119	0.258
8	207.3	279.6	0.00253	0.237
9	520.6	568.5	0.00175	0.187
10	639.1	538.7	0.00176	0.219
11	2,908,329.9	1,363.9	0.00118	0.203
12	407.4	498.5	0.00183	0.205
13	3,739.3	1,105.7	0.00142	0.199
14	4,551.7	1,114.3	0.00176	0.191
15	6,778,974.8	1,364.1	0.00119	0.215
16	2,742.8	1,019.1	0.00156	0.186
17	911,008.5	1,362.9	0.00115	0.224
18	1,268,143.8	1,363.4	0.00119	0.181
19	15,895,527.1	1,364.3	0.00119	0.233
20	896,829.4	1,362.9	0.00119	0.223
21	12,458,656.4	1,364.3	0.00118	0.213
22	17,729,626.4	1,364.3	0.00119	0.220
23	12,458,656.4	1,364.3	0.00119	0.215
24	38,607.2	1,331.3	0.00137	0.146
25	1,322,726.8	1,362.7	0.00137	0.166