An Information Theoretic Perspective on E-beam Direct Write as Complementary Lithography

S. A. Savari[1]
Department of Electrical & Computer Engineering, Texas A&M University
savari@ece.tamu.edu

A well-known challenge to the success of electron-beam direct-write (EBDW) lithography is its low throughput in the patterning of wafers. In order to solve this problem, future EBDW lithography systems must transfer information at rates on the order of terabits per second, and two complementary approaches to addressing the data transfer problem are parallelism and lossless data compression[2].

Complementary lithography[3] has received attention as an approach for fabricating unidirectional and gridded layouts; here optical lithography produces unidirectional lines at a fixed pitch and an EBDW lithography system cuts those lines. We focus here on the transfer of cut information via images consisting of binary pixels because cut locations and shapes are permitted some latitude in complementary lithography[4], but the discussion extends to gray scaling.

Practical data compression schemes are influenced by the number of possible objects that may be transmitted for a particular application. For images of cuts, the "information" present is determined both by the number of pixels and by the constraints regarding the location of cuts. In Figure 1, there are cuts that cross multiple lines, but we can obtain an upper bound on the "information" within a cut image by assuming that the cuts along different lines are independent. Shannon[5] was inspired by the constraints present in telegraphy to invent the finite-state channel, which we apply here, as a model for transmitting one-dimensional information. Let us assume that for the example of Figure 1 we choose pixels that are 10 nm $\times$ 10 nm so that a cut will be 30 nm $\times$ 20 nm as illustrated in Figure 2. Then for the row of pixels inside the line, Shannon's theory can be used to show that the binary information per pixel is at most $\log_2(1/x)$, where $x + x^8 = 1$; this is about 0.301 bits per pixel. Furthermore, three successive rows of pixels are identical and the next one may be blank if cuts are independent from one line to the next, so the information content for the image is at most 0.076 bits per pixel; there are compression schemes to approach this limit[6,7]. If successive cuts are known to be further apart, then the information content decreases and data compression on cut images becomes even more effective. The encoding of one-dimensional data within other applications is often called entropy encoding.

Video communication involves large volumes of data and high throughputs. An older paper[8] discusses video decoders with 70 Gb/s of computation and memory accesses and claims (see Figure 3) that the entropy decoding accounts for less that 6% of that run time. There may be an opportunity to leverage existing work on hardware implementations of entropy decoders for complementary lithography.

Acknowledgments: The author is grateful to Y. Borodovsky, M. Rieger, J. Yang, M. Guillorn, G. Lopez, R. McCay, and T. Groves for stimulating conversations.

1. This work was supported in part by NSF grants CCF-1017303 and ECCS-1201994.

2. H. J. Levinson, *Principles of Lithography* (SPIE, Bellingham, 2010), Third Edition, pp. 466-467.

3. Y. Borodovsky as discussed in D. Lam, E. Liu, M. Smayling, and T. Prescop in *Alternative Lithographic Technologies III, Proc. of SPIE Vol. 7970, 2011*, edited by D. J. C. Herr, 797011.

4. D. K. Lam, Future Fab International, **42**, 55-60, (2012).

5. C. E. Shannon, Bell Syst. Tech. J., **27**, 379-423, (1948).

6. S. A. Savari and R. G. Gallager, IEEE Trans. Inf. Theory, **40**, 100-107, (1994).

7. S. A. Savari, IEEE Trans. Inf. Theory, **46**, 1246-1262, (2000).

8. T.-C. Chen et al., IEEE Circuits & Devices Magazine, 22-31, May/June 2006.

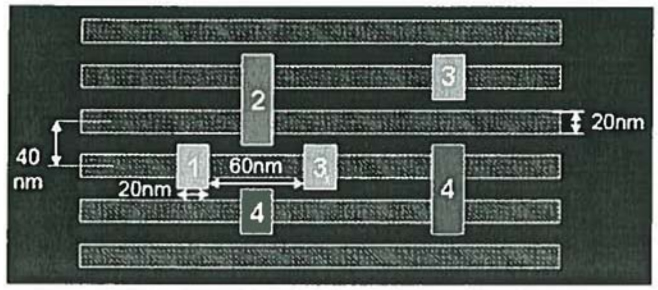

Figure 1. Intel's proposal for 20 nm line/space (from Reference 3).

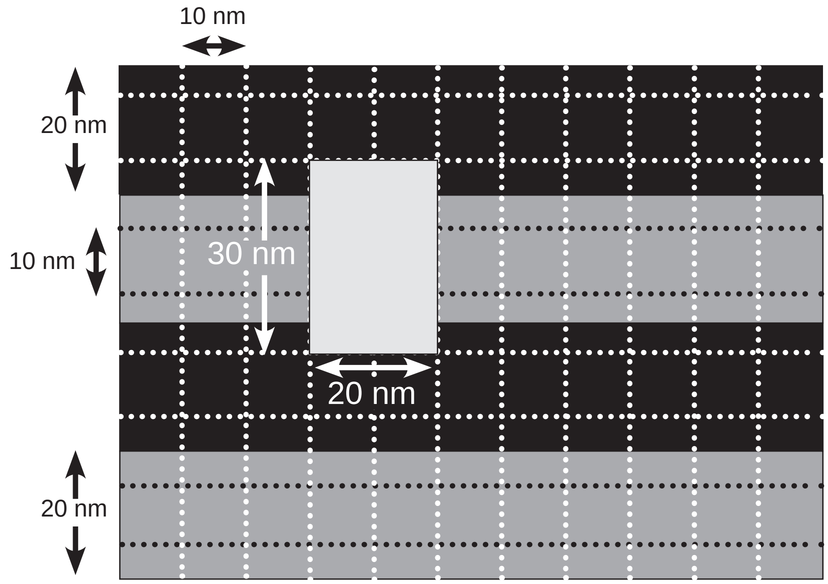

Figure 2. A discretization of Figure 1.

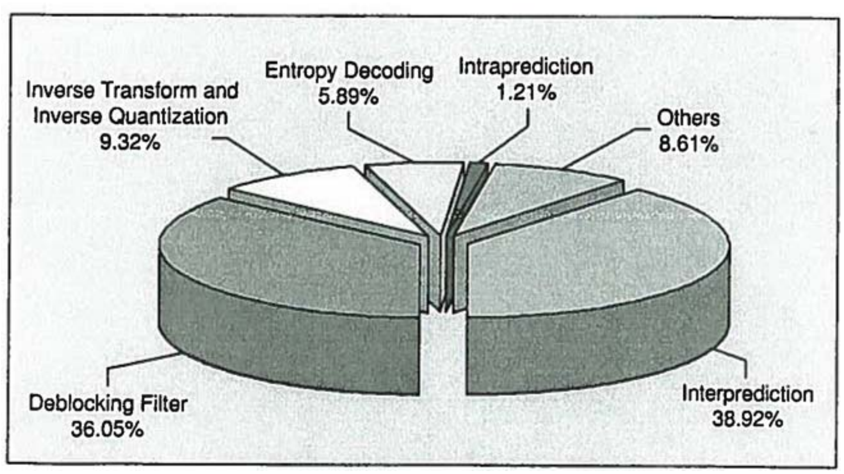

Figure 3. Run-time profile of video decoding for the H.264/AVC standard (from Reference 8).