Accelerating DFM Electronic Data Processes using the Cell BE Microprocessor Architecture.

F.M. Schellenberg, T. Kingsley, N. Cobb, D. Dudau and R. Chalasani
Mentor Graphics, 1001 Ridder Park Dr. San Jose, CA 95131

J. McKibben and S. McPherson
Mercury Computer Systems, 2105 S. Bascom Ave, Campbell, CA, 95008

***Abstract*—The demands of computational lithography are increasing as Moore's Law drives IC generations from 65nm through 45nm to 32nm and beyond. To achieve results for RET processing with reasonable turnaround times, large scale parallelization and hardware acceleration are being applied to computational lithography tasks. One processor, the Cell BE, is investigated for its impact on typical tasks of computational lithography. The Cell BE architecture is well suited for the computational tasks required for RET (FFTs and matrix multiplication), and can produce a speedup by a factor of 10 under certain configurations.**

***Index Terms*— Cell processor, computational lithography, hardware acceleration, parallel computing, RET.**

I. Introduction

In the past 10 years, the final stage of IC design has undergone a dramatic transformation. In the conventional process flow for IC creation, the designer originally creates the IC specification and RTL. The RTL is then converted by synthesis to a transistor level Netlist, and the Netlist converted through place and route to individual polygons which define the physical structures to be created by photolithography on a silicon wafer. [1]

Before these layouts are converted into photomasks for lithographic printing, the process waits for the layout to pass verification before the commitment of manufacturing resources. In the past, this check consisted primarily of a check against a deck of design rules (a design rule check, or DRC) and a comparison of a Netlist derived from the layout with the original source Netlist (Layout vs. Source, or LVS). Electrical properties extracted from the layout can also be determined using parasitic extraction tools, and the subsequent timing expected for the layout can be determined. If any of these checks fail, certain cells or sections can be reworked until the desired result is achieved and the layout "passes" physical verification. [2]

With the advent of sub-wavelength lithography in 1997 [3,4], features on the wafer are distorted in printing, sometimes severely for the smaller features. The layout as traditionally constructed will therefore inevitably fail. There are numerous optical adaptations that can be applied to improve the image fidelity, and generally fall under the name of Resolution Enhancement Techniques (RET) [5]. Most involve some pre-distortion or compensation of the layouts to correct for predictable process distortions.

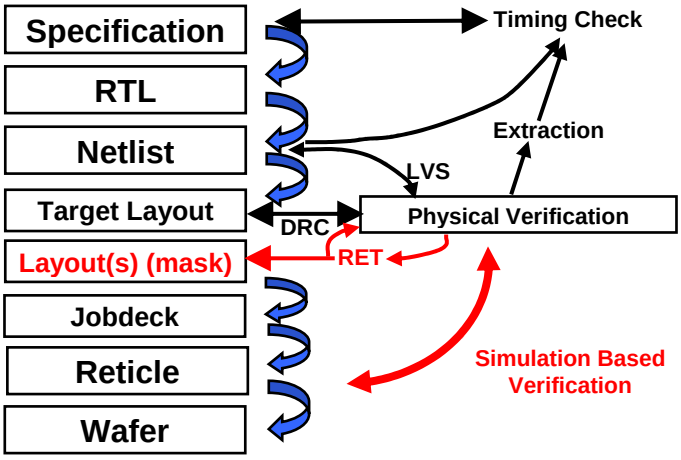


Figure 1. Typical IC design steps. Traditional physical verification encompasses DRC, LVS and extraction. Newer Physical Verification flows simulate wafer results and correct layouts with RET tools as well.

To implement these at the point of physical verification, software tools that allow simulation of the lithographic processes and implement these RETs have been added to the physical verification suite [6]. This is illustrated in Fig. 1. These add a layer of symmetry to physical verification, allowing forward prediction to go along with the already existing checks for upstream compatibility (LVS) and self consistency (DRC). Initially introduced for the 180nm generation, each successive generation of ICs has had a growing number of IC layers that require RET treatments to achieve yield, as illustrated in Fig. 2.

Presented at the IEEE Electronic Data Process (EDP) Workshop,
Monterey CA, April 12, 2007.
Contact: frank_schellenberg@mentor.com or jmckibbe@mc.com

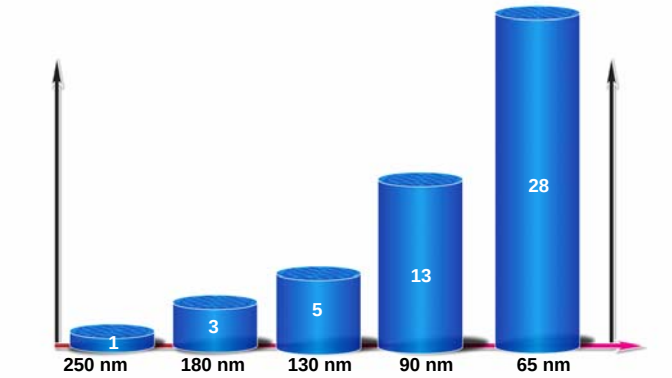


Figure 2. Increase in the number of IC layers requiring RET for various technology nodes.

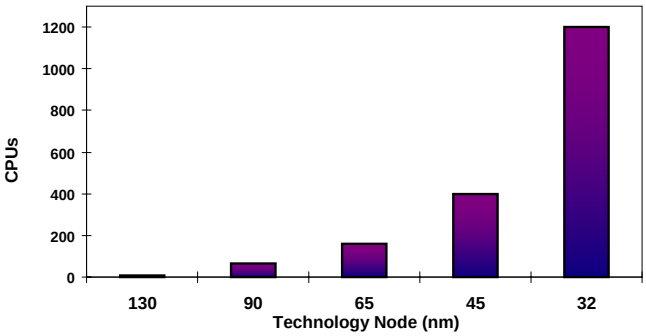


Figure 3. Increase in the number of CPUs used to complete typical RET jobs at various technology nodes. .

One of the characteristics of Moore's Law [7] is that the number of transistors in each subsequent generation increases exponentially. With more transistors, each one smaller, the growth in polygon data for an IC can become huge. The number of CPUs needed to complete an RET job in a reasonable amount of time therefore also is growing. This is illustrated in Fig. 3.

Part of this growth is driven by the manner in which the image simulations are computed. When the first image simulation programs were introduced, the computation was carried out on a uniform grid representing the area under simulation [8, 9] This has some advantages for computation, in that the mathematics for image computation is the same no matter what the layout pattern is. However, for layouts where significant portions are empty, a regular simulation engine will still spend time computing the image of these blank spaces, when the result (a uniform bright or dark field) is already known.

To eliminate these redundant computations in RET simulators, sparse computation algorithms were developed [10]. In a sparse image computation, image intensities are computed only at designated simulation sites positioned at or near polygon edges, and the computation results used locally to govern the changes in positions of the local edge segments. In this way, computing resources are not wasted computing images of blank space with no features. As the exposure wavelength has remained at 193 nm while the layout features become smaller, the density of these sites increases significantly, as shown in Fig. 4. The advantage of sparse computation begins to disappear, and at some generation, generally believed to be between the 45nm and 32nm nodes, a crossover point occurs where sparse computation is actually more expensive than the a dense, grid-based approach. [11] This is illustrated in Fig. 5.

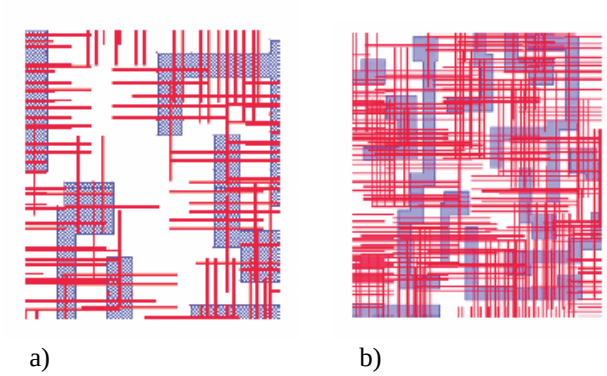


Figure 4. a) Simulation sites for a sparse 65nm node layout, and b) Simulation sites for a "sparse" 45 nm layout.

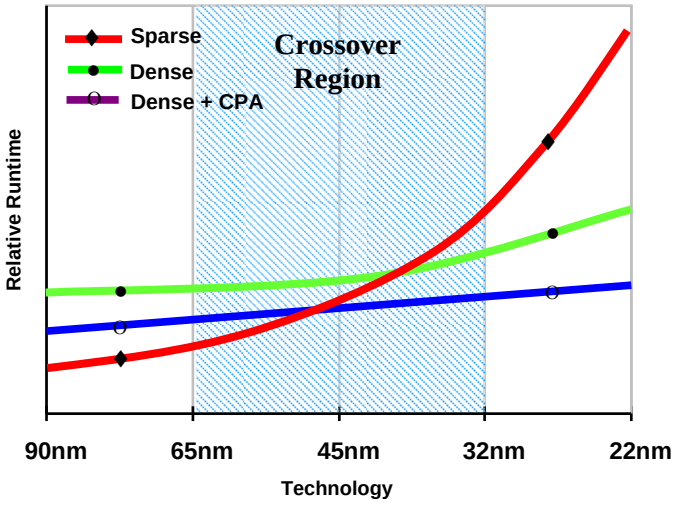


Figure 5. Predictions of relative runtime using sparse and dense simulations. CPA stands for coprocessing accelerator.

II. Speed Improvements through Parallel Processing

The implementation of the dense, grid-based computing for simulation invites again the use of parallel processing, tuned for the particulars of grid-based image computation to achieve scaling in processing speed.

Although physical verification tools have had architectures that allow for multi-threaded processing for many years, the division of labor in the past has typically been to partition the problem by cell, managing hierarchy of the layout and assigning various layout cells for processing by the available CPUs. In general, all the CPUs were identical, and the computations could proceed.

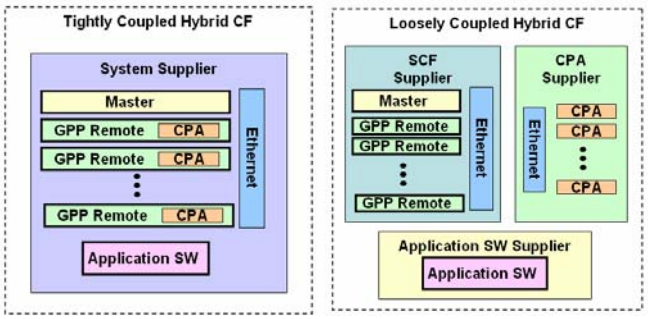
The approach now required needs further refinement. This can come from the recognition that, in dense grid-based image computations, the computation of an image is fundamentally the computation of a 2-D Fourier Transform. [12] Furthermore, work carried out over the past half century has thoroughly explored the theorem that any linear imaging system can be mapped to a 2-D Fourier Transform, with all optical properties of the system accounted for using a complex pupil function derived from the properties (e.g. aberrations) of the lens. Imaging then becomes a matter of taking a Fourier Transform, performing multiplication with the representation of the Pupil Function in the imaging plane, and then taking an inverse Fourier Transform to produce the image. [13]

There is therefore a premium demand not only to distribute jobs for various cells to various processors, but to send those that contain dense image computations to processors especially suited for the computation of Fourier Transforms.

The FFT, or Fast Fourier Transform, has been well known as an algorithm that produces the Fourier Transform for a function sampled on a discrete grid with $O(N \log N)$ calculations instead of the $O(N^2)$ computations that a direct computation of the Fourier integral would require [14]. This efficiency can also apply to multi-dimensional Fourier Transforms, which are essentially sequences of FFTs carried out on the 2-D matrix by row and column. If the layout to be computed can be suitably digitized and the Transform computed, the details of the computation become an exercise in FFTs and matrix multiplication in the Fourier plane. These tasks (2-D FFT and array matrix multiplication) can be accelerated with specific hardware configured to make this especially fast.

One architecture that can be used for this task is a dedicated co-processor, as shown in Fig. 6a. Here, each CPU assigned to compute part of the task is coupled locally to a dedicated co-processing accelerator (CPA), such as a programmable FPGA. This tightly coupled architecture may appear to have several advantages, notably the ability to create a very high bandwidth connection between the CPU and the CPA.

However, this architecture has a disadvantage in that the co-processing computational power is essentially dedicated to aid the task of the CPU. Once the CPU's job is finished, it sits idle, waiting for the next portion of the job to be assigned to it for computation. In this time, the CPA coupled to the idle processor also sits idle.



a) b)

Figure 6. a) Tightly coupled parallel processing architecture, with coprocessor accelerators (CPAs) linked to each general purpose processor (GPP), and b) loosely couple parallel architecture, with the CPAs linked to the GPPs through an Ethernet.

A different architecture, in which the CPAs are loosely coupled to the general purpose CPUs through a network, can allow more flexibility and ultimately more utilization of the entire computing power of the system. Such an architecture is illustrated in Fig. 6b. For this architecture, the selection of CPAs that can perform both accelerated processing of FFTs and image computations, and at other times can also perform routine computations as if it were a CPU, can provide the most overall efficient computation of the task at hand.

III. The Cell BE Processor

The recent development of the Cell processor [15] provides a computing engine that can provide both accelerated FFTs and matrix multiplication [16]. Developed to be the core of the Sony Playstation 3 gaming system, the Cell Broadband Engine (Cell BE) consists of a single 64-bit PowerPC processor element (PPE) and eight synergistic processor elements (SPEs), as illustrated in Fig. 7.

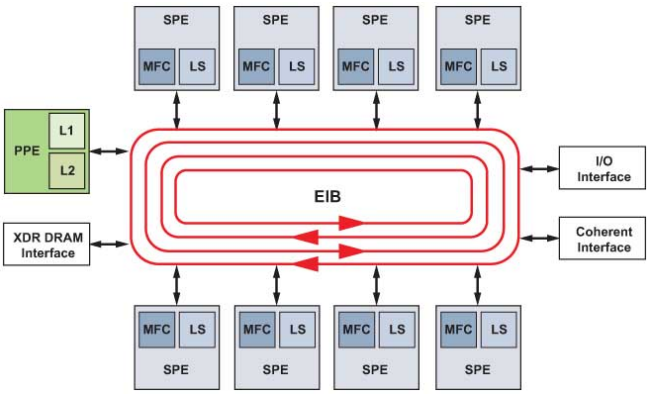


Figure 7. Architecture of the Cell BE processor.

Unlike a typical coprocessor, each SPE contains a synergistic processing unit (SPU), a local memory (256KB, used for both the SPU's code and data), and a memory flow controller (MFC). The PPE and the SPUs are all connected with a high bandwidth (~25 GBytes/sec) element interconnect bus (EIB). With 8 concurrent transfers over the bus between SPEs, this leads to a possible inter-SPE bandwidth of ~200 GB/sec. This bus also provides a concurrent I/O link to the main external memory. The Cell itself operates at a nominal clock speed of 3.0 GHZ, but higher frequencies are possible.

This architecture is extremely efficient at computing both FFTs and large matrix multiplications. [16] Individual 1-D FFTs can be assigned to individual SPEs and processed in parallel to accelerate the 2-D FFT computations. Fig. 8 shows the relative performance of the Cell processor for FFTs, as compared to general purpose processors. The Cell is at least an order of magnitude faster, often more than 30x faster than other state-of-the-art computing platforms. For the 64K FFT using a 3.2 GHz Mercury blade, the performance represents a speed of 90.8 GFlops.

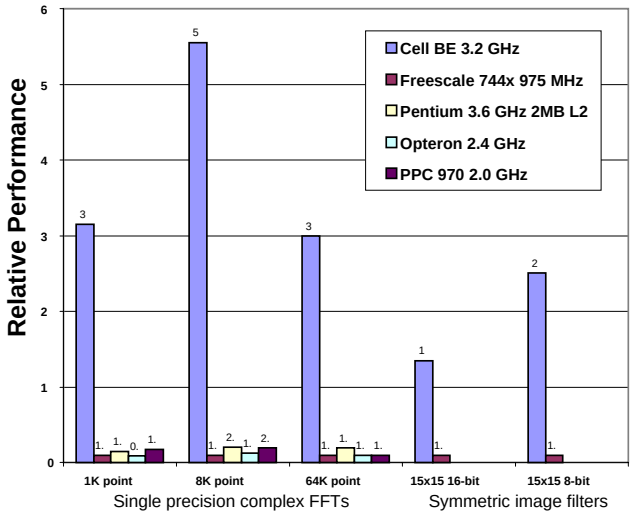


Figure 8. Measured speedup results for FFTs and image filters for the Cell BE and other CPU platforms.

IV. PREDICTED SPEEDUP

If the amount of parallelization that a computing job allows can be determined, Amdahl's Law allows us to compute the speedup that may be achieved. [18]

Amdahl's Law states that

$$Speedup(S) = \frac{N}{s * N + (1-s)} = \frac{1}{1 + p * (1/N - 1)} \quad (1)$$

where:

s = the percentage of the job which is sequential
p = the percentage of the job which is non-sequential
$(s+p) = 1$
N is the number of CPUs for parallelization.

If the initial baseline number of processors is greater than 1, then this equation becomes

$$Speedup(S) = \frac{1}{1 + p * (n/N - 1)} \quad (2)$$

where:

n = the number of baseline processors.

As an example, for a representative 65nm layout computational lithography job with 4 hours sequential processing, and if, with n=25 CPUs, the total computation time is 80 hours, then s=0.05 and p=0.95.

Adding more processors to this configuration (e.g. 75 additional CPUs, for a total of N=100) decreases the compute time with a Speedup of

$$Speedup(S) = \frac{1}{1 + 0.95 * (25/100 - 1)} = 3.48 \quad (3)$$

With this speedup, our 80 hour job becomes a 23 hour job, completing within the typical specified target of 24 hours.

This can be extended to accommodate co-processor acceleration as well, with what can be called the Hybrid Amdahl's Law for Speedup with acceleration *(Spa):*

$$Spa = \frac{1}{1 + p * (n/N * (1 + a(1/uA - 1)) - 1)} \quad (4)$$

where p, n and N are as defined above, and :

a = the percentage of the job that can be accelerated
A = the total acceleration provided by the CPA,
u = the utilization of the CPAs by the general processors.

As a concrete example, we can extend the previous example where p=0.95, and assume in addition to the extra processors, a=0.85, u=1 (100%), and A=20. Then,

$$Spa = \frac{1}{1 + 0.95 * (25/100 * (1 + 0.85(1/20 - 1)) - 1)} = 10.45 \quad (5)$$

improving the processing time for the job by over an order of magnitude.

We should note that equation (4) also allows us to compare acceleration by simply adding processors vs. using specialized accelerators. Plots of this is shown in Figs. 9 and 10.

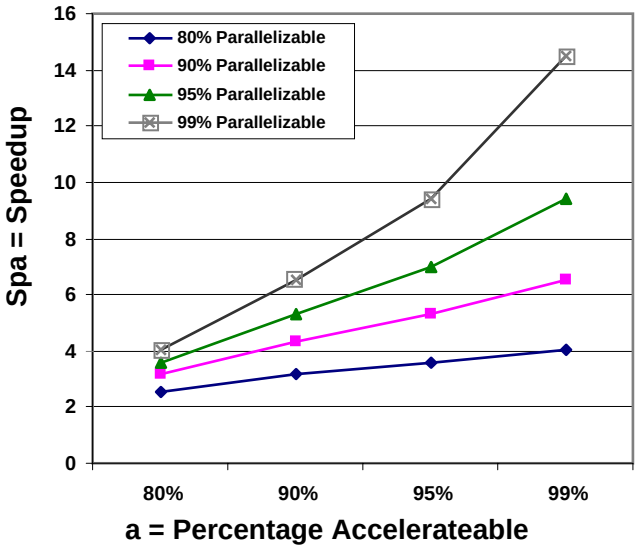


Figure 9. Speedup results as the percentage accelerateable *a* is varied.

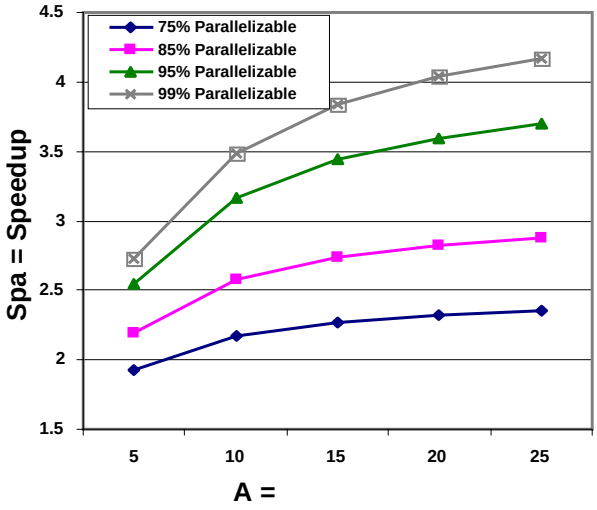


Figure 10. Speedup results as the acceleration *A* is varied.

As another specific example, again assuming that a=0.85, u=1 (100%), and A=20, for the case of acceleration only, N=n=25, and the computation yields Spa = 4.30, similar in magnitude to the speedup achieved above without acceleration by simply adding instead an additional 75 CPUs. Achieving a balance between the right amount of acceleration and the right number of CPUs then becomes a matter of cost of ownership for these options [19]. The gain in *Spa* vs. the marginal cost of the additional CPU or accelerator can be calculated, and the configuration appropriate for the typical job can be procured.

V. Future Extension

Although TCAD simulation has been around for decades [20], it should be noted that the phrase "computational lithography" was only recently coined with the establishment of RET. The use of extensive numerical processing in predicting circuit structures formed in semiconductor processing is expected to expand.

One possible route of expansion are the applications commonly referred to as Design for Manufacturing, or DFM. Here, computations are made not only at nominal conditions but also at various conditions that represent tolerable variations in processing conditions, such as focus, exposure, overlay, etc. The variation in line placement allows a physical verification tool suite to consider not only the nominal placement of a designed polygon edge, but also contours representing a "process variation" band, or PV-band [21, 22]. This is illustrated in Fig. 11. The Mentor Graphics product Calibre LFD (for Litho Friendly Design) is based on the computation of PV bands [23].

PV bands are computed using essentially the same mathematics of image computation used for the nominal image, and therefore are also amenable to acceleration using either additional CPUs or CPAs. At this point, turnaround times using LFD as normally configured are adequate, and the need for this acceleration for the IC generations where LFD is being applied is not essential. However, as Moore's Law drives dimensions smaller and the number of transistors higher, we expect LFD as well to take advantage of the same speedups applied here for RET software applications.

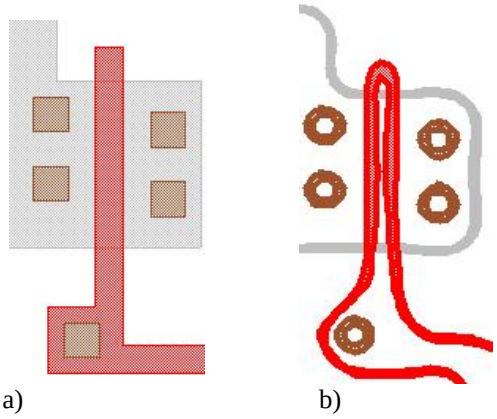


Figure 11. a) Original layout, and b) results corresponding PV bands.

Lithographic effects are a convenient first application, as the nature of the image computation using FFTs allows easy acceleration using hardware components such as the Cell processor. Both etch effects and CMP processing artifacts are amenable to numerical simulation, although an algorithm as efficient as the FFT has yet to be determined for these numerical models.

VI. Conclusions

At this point, it is clear that the insertion of process models in final stage of physical verification makes this job far more computationally demanding. For the computation of images, required by RET software and for future extensions to LFD and other DFM applications, hardware architectures which allow the acceleration of 2-D FFTs and matrix multiplication can improve the overall turnaround time by at least an order of magnitude.

Using software applications developed in the Mentor Graphics Calibre RET product suite and hardware configurations including Cell BE processors from Mercury Computing systems [24], speedups of 30x for RET applications are possible. The exact magnitude of the improvement, however, depends on the specific nature of the layout patterns to be processed, and the hardware configuration for sharing the computing jobs between main CPUs, coprocessors, and Cell processors with SPEs. Exact determination of an optimal solution will depend on the type of design being verified (e.g. memory ICs vs. random logic processors), the availability of network vs. local computing resources, and the speed of the network that can be maintained between processors.

For these initial results, however, a loosely coupled combination of CPUs and Cell BE processors appears to be able to offer significant advantages over the other options.

Acknowledgment

We thank Jean Marie Brunet, Andres Torres and John Sturtevant of Mentor Graphics, and Robert Cooper, Matt Sexton and Jonathan Greene of Mercury Computer Systems for their help in the preparation of this manuscript.

References

[1] For a general reference on EDA, see L. Scheffer, L. Lavagno and G. Martin, *Electronic Design Automation for Integrated Circuits Handbook*, Boca Raton, FL. CRC Press, 2006.

[2] See Section III (Ch. 17 – 23) of Ref. [1].

[3] For plans and forecasts on IC technology nodes, see http://www.itrs.net/

[4] T. Terasawa, "Subwavelength optical lithography", in *Challenges in Process Integration and Device Technology*, *Proc. SPIE* vol. 4181, pp. 5-13, 2000.

[5] Alfred K.K. Wong, *Resolution Enhancement Techniques in Optical Lithography*, Bellingham, WA, SPIE Press, 2001.

[6] F.M. Schellenberg, "Adoption Costs and Hierarchy Efficiency for 100 nm and beyond", in *Design, Process Integration, and Characterization for Microelectronics*, *Proc. SPIE* Vol. 4692, pp 593-604, 2002.

[7] G. Moore, "Cramming more components onto integrated circuits," *Electronics* vol. 38, pp. 114-117, 1965.

[8] K.K.H. Toh & A.R. Neureuther, "Identifying and monitoring effects of lens aberrations in projection printing," in *Optical Microlithography VI, Proc. SPIE* 772, pp. 202-209 (1987)

[9] A. Neureuther and C. Mack, "Optical Lithography Modeling", Ch. 7 of *Handbook of Microlithography, Micromachining, and Microfabrication, Vol 1: Microlithography*, P. Rai-Choudhury, ed. Bellingham, WA, SPIE Optical Engineering Press, 1997, and references therein.

[10] N.B.Cobb, *Fast Optical and Process Proximity Correction Algorithms for Integrated Circuit Manufacturing*, Ph.D. Dissertation, University of California at Berkeley, 1998.

[11] N. Cobb & Y. Granik, "Dense OPC for 65nm and below" in *25th Annual BACUS Symposium on Photomask Technology, Proc. SPIE* vol. 5992, 599259 (2005).

[12] P.M. Duffieux, "L'Intégrale de Fourier et ses Applications à l'Optique", Rennes, Societé Anonyme des Imprimeries Oberthur, 1946.

[13] J.W. Goodman, *Introduction to Fourier Optics*, 3rd Edition, Greenwood Village, CO, Roberts & Co. 2005.

[14] E. Brigham, *The Fast Fourier Transform and Its Applications*, New York, Prentice Hall, 1988.

[15] For more on Cell processors, see http://www.ibm.com/developer/power/cell

[16] S. Williams, J. Shalf, L. Oliker, S. Kamil, P. Husbands, K. Yelick, *Proceedings of the ACM International Conference on Computing Frontiers CF'06*, May 3-5, 2006.

[17] J. Greene and R. Cooper, "A Parallel 64K Complex FFT Algorithm for the IBM/Sony/Toshiba Cell Broadband Engine Processor", *Tech. Conf. Proc. of the Global Signal Processing Expo (GSPx)*, 2005.

[18] G. Amdahl, "Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities", *AFIPS Conference Proceedings*, vol. 30, pp. 483-485, 1967

[19] T. Kingsley, J. Sturtevant, S. McPherson, M. Sexton. "Advances in compute hardware platform for computational lithography", in *Optical Microlithography XX, Proc. SPIE* vol. 6520, paper 6520-44, 2007 (in press).

[20] R. Dutton & Z. Yu, *Technology CAD – Computer Simulation of IC Processes and Devices*, Dordrecht, Netherlands, Kluwer Academic Publishers, 1993.

[21] J.A.Torres Robles, *Integrated Circuit Layout Design Methodology for Deep Sub-Wavelength Processes*, Ph.D. Dissertation, OGI School of Science and Engineering, July 2005.

[22] W. Hoppe, T. Roessler, & J. A. Torres, "Beyond rule-based physical verification", in *Photomask Technology 2006, Proc. SPIE* vol. 6349, 63494X, 2006.

[23] For more information on Mentor Graphics' Calibre products, see www.mentor.com/dsm .

[24] For more information on Mercury Computer Systems, see www.mc.com .