IC Design in High-Cost Nanometer-Technologies Era

Wojciech Maly
Carnegie Mellon University
ECE Department
Pittsburgh, PA 15213
412 268 6637
maly@ece.cmu.edu

ABSTRACT
Nanometer IC technologies are on the horizon. They promise a lot. They will cost a lot as well. Therefore, we need to ask today: How may the billions of dollars, which we will need to spend on nanometer-fablines, affect IC design domain? This paper attempts to address the above question by analyzing design-manufacturing interface. The partial answer is derived from a simple transistor cost model proposed in the body of the paper.

Keywords
Design of ICs, cost modeling of IC design and manufacturing, IC technology trends, nanometer-technologies, and IC layout regularity.

1. INTRODUCTION
Innovative engineering, with its roots in science, physics and math, has a natural tendency to simply ignore limitations imposed by the economics. Microelectronics, a relatively new branch of engineering, has this tendency as well. But this has been the case not without reasons. The last 30 years has shown that in microelectronics, the only limit has been a "blue sky".

Is this trend going to continue? Will nanometers-technologies be economically feasible as the cost of manufacturing facilities approaches many billions of dollars? If "yes", how will the economic force innovators to be more respectful? And finally, more precisely, what do we need to do in the IC design domain to obey rules of the economic reality, while still allowing a continuation of the progress along the trend defined by Moore's Law [1,2]? Of course, the answers to these intriguing questions are not trivial - especially in the context of recent events on Wall Street. The engineers themselves cannot provide them either.

This paper is intended to address only a small subset of the issues outlined above by attempting to see what might be the consequences of the exponentially growing cost of IC manufacturing facilities on the IC design domain. The discussion presented in the paper is conducted along the lines already defined in [3] and [4]. But, it is focused on the constructive response of the IC design domain to the nanometer-technology cost-based challenge, rather than on the assessment of feasibility of the ITRS Roadmap itself.

DAC 2001, June 18-22, 2001, Las Vegas, Nevada, USA.

The paper is organized as follows. First, we discuss the cost of the development of an IC as a function of key design and manufacturing technology parameters and we propose a cost model, which captures essence of the discussed relationships. Next, we analyze the recent trends in the change of these parameters (extracted from the available industrial data) and we compare them to "official" predictions of the ITRS [3]. We conclude that the observed trends must be changed by the appropriate modifications of contemporary design practices. Finally, we conclude with discussion of the IC design strategies and styles that seem to be most suitable for the future IC nanometer-technologies cost challenge.

2. COST MODEL
The development and manufacturing of integrated circuits may require financial resources ranging today from a few millions up to billions of dollars. Modern ICs are also very different in size and complexity. Thus, in order to assess objectively the cost effectiveness of the IC development and fabrication investments it makes sense to measure the costs of manufacturing, test and design in terms of "dollars" spent per single transistor in a fully functional IC. This paper uses such a measure and assumes that it should be minimized to assure the best usage of the invested financial resources.

2.1 Cost of manufacturing
One can express the cost of manufacturing of a single functioning transistor, C_{tr}, as [3]:

$$C_{tr} = \frac{C_w}{N_{tr} N_{ch} Y} \quad (1)$$

where:
C_w is fabrication cost of the manufacturing wafer,
N_{tr} is number of transistors per chip,
N_{ch} is number of chips per wafer and
Y is manufacturing yield.

Formula (1) can be rearranged by observing that transistor density, T_d, can be expressed in the following way:

$$T_d = \frac{N_{tr}}{A_{ch}} = \frac{N_{tr}}{N_{tr} \lambda^2 s_d} = \frac{1}{\lambda^2 s_d} = \frac{d_d}{\lambda^2} \quad (2)$$

Consequently,

$$C_{tr} = \frac{C_{sq} \lambda^2 s_d}{Y} = \frac{C_{sq} \lambda^2}{d_d Y} = C_{sq} \lambda^2 \frac{s_d}{Y} \quad (3)$$

where:
A_{ch} is the area of the IC chip,
C_{sq} is the manufacturing cost computed per cm ^{2} of fabricated wafer,
λ is the minimum feature size,
s_d is the **design decompression** index (or design sparseness) expressed in terms of a **number of minimum feature size squares needed to draw an average transistor**,

d_d is the **design density** index expressed as an inverse of the **number of minimum feature size squares needed to draw an average single transistor**.

Note, that this way the cost of a transistor is expressed in terms of the process dependent characteristics (C_{sq}, λ and Y) as well **as the process independent design attribute s_d or d_d.**

2.2 Design density

2.2.1 Utility of design density measure

The design density and design decompression index [3] defined in the previous section may be used to describe very fundamental attributes of an IC design. To assess this claim first observe that transistor density (2), is a strong function of minimum feature size λ. Thus, the improvements in transistor density observed over the years can be attributed to both the shrinking minimum feature size, as well as to the improvement in design density (inverse of design decompression index). Since traditionally progress of microelectronics has been reported (measured) by using the transistor density values, it was difficult to trace how the design itself contributed to the increases in the scale of integration for a particular device or design style.

The second important observation about design density and design decompression indices is that their values have a relatively large range. For instance, the smallest values of s_d obtained for SRAM memories are in range of 30, while s_d in some ASIC designs can reach values in the range of 1000. Even within one design style category (e.g. designs using the same library of cells) one can see designs having substantially different design densities. These differences can be attributed to both differences in product architecture and differences in specific design algorithms/methodologies employed in a particular design flow. This implies that the design density might a useful figure of merit for both the design cost effectiveness (i.e. manufacturability) and an indicator of the quality of the design process itself.

2.2.2 Design decompression index in large industrial designs

To put the above statements into an appropriate perspective we conducted the study involving several tens of design characteristics published in the open literature (see e.g. [5-29]). A portion of the obtained results is shown in Table A1 at the end of the paper and the extracted s_d values are plotted in the graph in Figure 1.

The data in Figure 1 carries important messages. First of all, one can see that there is a clear tendency among major microprocessor producers to introduce products with worsening design densities. (See in Table 1A products with the available separate memory and logic data.) Such a trend could be explained by the growing need for more interconnect (e.g. more buses). But for the newest technologies, with 6+ metal layers, this could not be a reason for a two or more fold increase of s_d. Hence, it is fair to assume that the time to market pressure must be a factor deciding about compactness of modern custom-designed ICs. The second observation is that market position, and consequently importance of manufacturing cost issues, does affect design strategy that affects the design cost effectiveness. For instance, for a long period of time AMD - the market follower - introduced products of higher design density than its immediate competitor. Of course, AMD's strategy was to compete with Intel by using less expensive transistors. This was perhaps the optimum strategy until AMD entered a direct performance confrontation by introducing its K7 microprocessor - whose s_d is well above 300 squares per transistor.

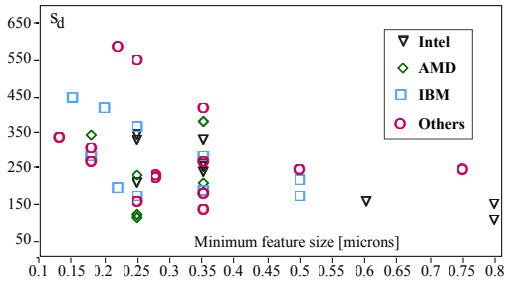

Figure 1. Design decompression index of microprocessors and ASIC ICs as a function of minimum feature size extracted from devices listed in Table A1.

Hence, the general conclusion of above analysis is that design density is a good indicator of, at least, design compactness. Thus, indeed it should be viewed as a key figure of merit in assessing the design customization level that, in turn, must be strongly correlated to the design time and manufacturing costs and, finally, to the product manufacturability.

2.2.3 Desired design density trends

Since, one of the key objectives of this paper is to find out what could, and should, be done to address the anticipated high cost of nanometer-technologies, it is useful to see now what s_d values should be achieved, if ITRS [2] objectives are to be met? To answer this question the ITRS transistor density data was plotted against minimum feature size. The results are shown in Figure 2.

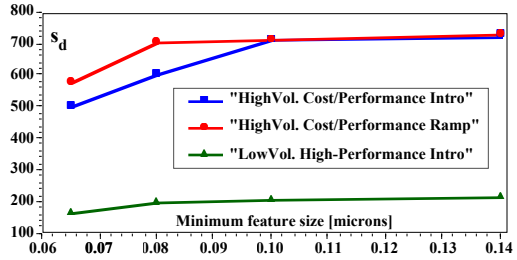

Figure 2. Design decompression index s_d for microprocessors from ITRS [2] data.

Even more interesting are the results of computing s_d, by assuming (as the ITRS document assumes) that the cost of a die should stay on the same stable level.

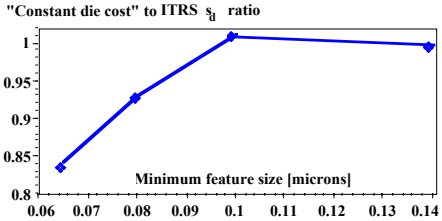

Figure 3. Design decompressions s_d indices ratio for microprocessors (MPUs), computed from ITRS [2] data and by assuming that the cost of the die must remain constant.

Figure 3 shows results of such computations executed by deriving from formula (3) value of s_d needed to keep the cost of

the MPU die at the level of 1999 as predicted for high-volume cost/performance MPU by ITRS [2]. In the calculations we used: maximum acceptable cost of the MPU die C_{ch} = \$34.0, the cost of manufacturing per unit area C_{sq} = 8.0 \$/cm^2, and yield Y = 0.8. The numbers of transistors on the die and the minimum feature sizes were as dictated by the ITRS document. The plot in Figure 3 shows the ratio of s_d computed directly from ITRS data and with the method explained above.

As one can clearly see all of the data presented in this section indicates that the current industrial trend to increase s_d is not aligned with the ITRS predictions and with the ITRS-independent necessity to keep the cost of the die at the same level. Note that this lack of alignment was demonstrated by using a very optimistic scenario i.e. assuming no increase in C_{sq} and no decrease in yield, is highly unlikely, given all the physics and economics based challenges of nanometer-technology. Hence, the **cost contradiction** mentioned in [4], between market expectation and the cost of manufacturing, is combined with the contradiction between the **time-to-market driven modern-design-mentality (resulting in the tendency to increase s_d)** and, again, with **the growing cost of new processes**. To see more clearly this trade-off it is useful to discuss a new element of the cost picture: the cost of design.

2.3 Total cost of transistor

It is obvious that very often the manufacturing cost is only a portion of the relevant nanometer-technology IC development cost. Fortunately, the manufacturing cost model can be easily extended to include another important cost contributor: the cost of design. This can be done adding into model (3) an extra component Cd_{sq} that describes the cost of design of each cm^2 of fabricated silicon. This way the cost of a single transistor in fully functional IC is:

$$C_{tr} = \frac{\lambda^2 s_d}{Y}\left(Cm_{sq} + Cd_{sq}\right) \tag{4}$$

where

$$Cd_{sq} = \frac{\left(C_{MA} + C_{DE}\right)}{N_w\, A_w} \tag{5}$$

and

Cm_{sq} is used now to represent C_{sq} of model (3),
C_{MA} is the cost of the lithography masks,
C_{DE} is the total cost of the design activities,
N_W is the number of fabricated wafers.

Note that for high volume IC products (large N_w) C_{tr} described by (3) and (4) becomes equal.

2.4 Cost of design

Modeling of C_{DE} is, unfortunately, not a simple matter. The reason is that IC design effort is related to such variables as the complexity of the designed product, the level of competence and the experience of the design team and the distance between design specs and best theoretically possible performance of the available technology, to name a few. In addition there is very little information in the public domain describing cost of the design as a function of any design attributes.

On the other hand, it is easy to see that cost of the design must be strongly correlated to the number of design iterations. And that this number, in turn, is a direct derivative of our ability to correctly predict all the consequences of design decisions undertaken at each level of the design abstraction. For instance, timing closure would be much easier (faster, cheaper) to reach if it were be possible during logic synthesis to predict interconnect delays. But, often this can be only done successfully after synthesis process is accomplished. Hence, it is natural to assume that for the design cost to be much lower, timing objectives must be relaxed or new better methods, which can easily and accurately predict interconnect delay before placement and routing is done, must be developed. Note that the above problem becomes even more difficult with the decrease of the minimum feature size as the IC element's electrical characteristics are stronger and stronger functions of the geometry of the increasingly larger neighborhood.

To capture the nature of the above design cost modeling complexity in this paper we use a simplistic assumption. Namely, we have assumed, that the design effort grows as the inverse of the distance between the achieved s_d and that "the best possible" s_{d0}, which equals to the s_d of a fully custom-designed microprocessor with the best performance, available on the market for a given technology level. Simply, by analyzing data from [5-29] it was assumed that the best achievable s_d should be a number very close to 100. Notice that custom designs, by virtue of being the most compact are usually also the top performers. Of course, the above assumptions can be easily undermined by many design examples, which do not obey the above relationship. Nevertheless, they are used in this paper to propose a "first approximation design cost model" and this way to fill the big void, existing between IC design and manufacturing worlds. More specifically we propose that:

$$C_{DE} = \frac{A_0\, N_{tr}^{p1}}{(s_{d0} - s_d)^{p2}} \tag{6}$$

where, in addition to already defined symbols: A_0, p_1, p_2 are tuning parameters. Note that in this equation values of A_0, p_1, p_2 are used to capture the cost of unsuccessful design iterations which are due to the inaccurate predictions of product physical characteristics, used in the early stages of the design cycle. (In the computations presented in this paper we used the following values for A_0, p_1, p_2, respectively: 1000, 1.0, 1.2. They have been derived from a limited set of real life design/cost data available to the author of this paper[1].

2.5 Generalized transistor cost model

The cost model (4) is, as we have indicated before, a very simple first order approximation of the real life cost relationships and could be substantially expanded. There are two kinds of possible expansions. To the first kind belong any extra cost components or effects, which have been neglected for the sake of simplicity and due to the limited size of the paper. The best example of such an omission is cost of test, which could be easily included within the proposed cost-modeling framework. Another simplification is the omission in the model (4) of an extra parameter u, which would be very handy in modeling various level of hardware utilization. Such a parameter could be used to model cost of a transistor in, for instance, FPGA devices, in which only a subset of the fabricated transistors is involved in delivering useful function. (As we will show later parameter u can be easily added to the presented models by simply substituting yield Y with the product uY.) It also could be used to take into account

[1] It must be stressed, however, that this data should not be used for anything other than illustration purpose.

some of the functions that may require only a certain subset of components in the available system. (For example some of the applications may not need to use floating point unit which remains part of the IP and is still fabricated.)

The second category of inaccuracies of model (4) is due to unavoidable simplifications made in describing model's (4) parameters. The most important omitted facts are that:

- The cost of the fully manufactured wafer (and therefore Cm_{sq}) is a function of wafer diameter, minimum feature size, process maturity and, first of all, volume [30].
- Yield is a complex function of wafer diameter, minimum feature size [30,31,34], design density, process maturity as well as volume.
- Design, test and mask fabrication costs are related to minimum feature size, design size, design style, characteristics of the used IP and many other important factors - some of them have been already mentioned.

Hence, for the sake of "intellectual integrity" one can propose a generalized version of model (4):

$$C_{tr} = \frac{s_d \lambda^2 [Cm_{sq}(A_w, \lambda, N_w) + Cd_{sq}(A_w, \lambda, N_w, N_{tr}, s_{d0})]}{uY(A_w, \lambda, N_w, s_d, N_{tr})} \quad (7)$$

assuming that the ultimate objective of the cost studies should be modeling using model (7) with its overall complexity (and that without such a modeling capability the cost challenge of nanometer-technologies might become overwhelming).

Finally, one must also stress that all of the simplifications of the model (4) were made in such a way that it produces lower bound estimations of the transistor cost (the most optimistic). So, if results obtained from the model (4) are too high, from a practical point of view, then the full model would produce cost estimates even worse, or in other words, more pessimistic. Hence, the model (4) should still be useful as a "compass" helping to maneuver among possible cost-based stumbling blocks on the way towards minimization of C_{tr}, (i.e. minimization of "cost per function" in the terminology proposed by the NTRS and ITRS documents).

3. DESIGN FOR COST EFFICIENCY

The discussion presented so far has provided the conceptual bases for answering the most important question of this paper: What directions of IC design domain should be investigated now to meet the challenges generated by the anticipated high costs of nanometer-technologies? The answer of this paper is: Develop design strategies and CAD tools that can minimize C_{tr} approximated by models (4) and (7). The next question is then: What strategic optimization variables could we use? The simplest, but not always exhaustive, answer is: In the IC design domain we can control three variables: s_d ,Y and C_{DE}. To address main topic of this paper in the remainder of this section we will elaborate on the key elements of the strategy for choosing s_d and Y and minimizing C_{DE}.

3.1 Optimization of design variables

The nature of the relationships captured by the cost model (4) is such that from the cost of manufacturing standpoint the smaller s_d the lower the transistor cost C_{tr}. But, there is a limit of s_d, which can be achieved, because of rapidly growing cost of the design, C_{DE}, with the increase of the design density (decrease of s_d). Such a rapid growth is due to the already mentioned increase in the number of the unsuccessful design iterations. One can expect that this problem will be more severe for the nanometer-technologies in which prediction of physical parameters of the deigned product will be much more complex and costly. (In addition note that this tradeoff is modulated by the manufacturing volume – for small volume the design cost matters more.)

The strength of the above relation is illustrated in Figure 4 (a) and (b) by using model (4) with the following parameters: N_{tr} = 10,000,000 transistors, N_{wr} = 5000 and Y = 0.4 as well as N_{wr} = 50000 and Y = 0.9, for graphs (a) and (b), respectively. Notice that the location of the optimum s_d changes substantially with the volume and yield. This observation leads to a very important conclusion: Neither the smallest die size nor maximum yield, as it was the case in the past, should be the objective of the cost oriented IC design activities. It is the appropriate ratio of both which can provide the minimum transistor cost.

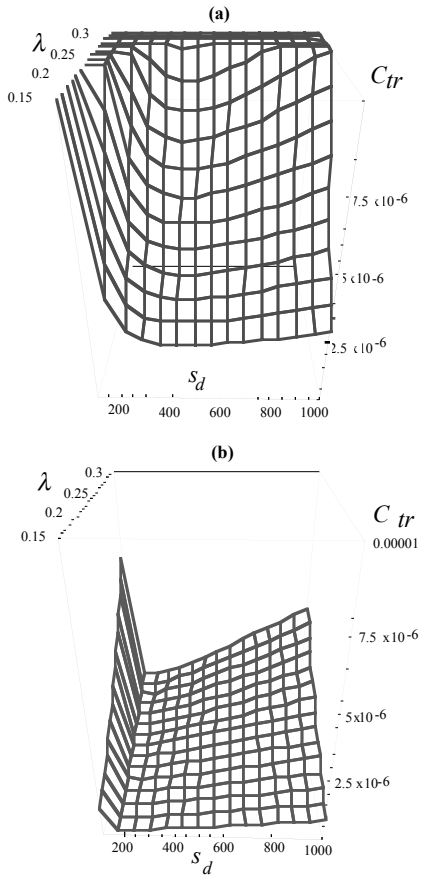

Figure 4. Cost of the transistor modeled by (4).

Recall now that the model (4) is imperfect and does not describe a number of important relationships. What is needed, therefore, is a set of more accurate approximations of model (7). Over the last couple of years one could observe an increased interest in using yield models in the design domain, in general, and in Design for Manufacturability (DfM), in particular [3,4,31,34,35]. But the progress in the technology development in recent years has been much faster than in any of the above, needed, modeling areas. One could even risk the

opinion that accelerated rates of technology development resulted in a decreasing ability, not only in the modeling of the causes of yield loss, but also in the understanding of the involved physics. So, despite of the increased interest in DfM there is little known how new technologies that use very new materials will behave, as far as process imperfections and equipment instabilities are concerned.

Therefore, in addition to the change of the design objective function, (from smallest possible die size to the minimum of C_{tr}.) there exist an urgent need to develop the cost and yield - oriented nanometer-technology relevant modeling techniques. Such techniques must be developed by the alliance of design, test and process development rodmaping strategies, which should adopt the yield/cost-modeling task as one of the key development objectives.

3.2 Cost efficiency of the design process

It was already hinted in this paper that the large number of poorly converging design iterations forms a core of the design cost/time problem and must be contained to minimize C_{DE} in (4). Thus, to keep C_{DE} on a manageable level one must improve the quality of the prediction of the parameters used in the design process. Traditionally, adequate predictions of the physical characteristics of the designed product have been achieved by using simulation. The problem is that for the nanometer-technologies the regions of mutual interaction between IC elements will grow in their relative sizes with. (For instance, the region, which has to be simulated in order to compute optical deformations of a certain design pattern must include a relevant neighborhood of this pattern [33]. The nature of the progress in lithography is such that number of relevant neighbors is growing with as minimum feature size decreases.) Thus, with the exponential increase in the number of elements of an IC, the complexity of the required simulation becomes enormous. Hence, not only during early stages of IC design but also during final design verifications the results of simulation will become uncertain and it is likely that loops of unsuccessful design iterations, that may involve failing manufacturing experiments, will grow as well.

The only solution to the above problem seems to be in the reuse of the results of very expensive simulation efforts (probably strongly supported by the test structure-based manufacturing characterization [33]) and in the adoption of geometric regularity of the design patterns [33]. High regularity would provide required accuracy of the prediction by reusing results of very accurate simulation of smaller segments of the design (patterns) across single products or entire family of products. (This way one will be able to increase an effective volume used in the computation of C_{DE}.) Hence, the conclusion is: Only by **applying in the design highly geometrically regular structures, created out of the limited smallest possible number of unique geometrical patterns**, one can hope to contain design cost of nanometer IC on the manageable level.

4. CONCLUSIONS

The development of nanometer-technologies is a natural consequence of the past successes of microelectronics. However, this success is likely to continue if, and only if, all new elements brought about by nanometer-technologies era are identified early enough and adequately addressed. The goal of this paper was to raise the level of awareness of the IC design community that design for cost minimization is likely to emerge as a high priority action item on IC design agendas. Two conclusions of fundamental importance were formulated during the discussion presented in this paper. First, it was argued that design for cost minimization must be guided by an adequately accurate cost objective function and performed by **using all design variables influencing** some measures of **design density and yield simultaneously**. The second conclusion is that the design cost itself must be carefully controlled. It was suggested in the paper that the first major steps towards this goal should be development of **new design styles** supported by CAD tools that use highly **regular, repetitive** (across many products) and **experimentally pre-characterized** design building blocks.

ACKNOWLEDGMENTS

The analysis reported in this paper is a byproduct of many, vibrant, insightful (and still ongoing) discussions with many creative individuals. The author of this paper would like to thank all of them especially stressing contributions of: H. Eisenmann, P.K. Nag, M. Niewczas, M. Palusinski, L. Pileggi, H. Schneider, A.J. Strojwas, P. Simon and T. Waas. He would like also to thank Andrew B. Kahng and Thomas Vogels for the help with the manuscript and GSRC for the financial support.

REFERENCES

[1] G. Moore, " VLSI: Some Fundamental Changes," IEEE Spectrum, Vol. 16, No. 4, 1979, pp. 79-87.

[2] The International Technology Roadmap for Semiconductors, 1999 Edition.

[3] W. Maly, "Cost of Silicon Viewed from VLSI Design Perspective," in Proc. of DAC-94, San Diego, June 1994, pp. 135-142.

[4] W. Maly, "High Levels of IC Manufacturability: One of the Necessary Prerequisites of the 1997 SIA Roadmap," Proc. of IEDEM-98, San Francisco, CA, pp. 759 – 762.

[5] J. Alvarez, et al, "450MHz Power PC Microprocessor with Enhanced Instruction Set and Copper Interconnect," Proc. of ISSCC 1999, MP 5.6.

[6] Y. W. Kim, et al, "A 0.25µm 600GHz 1.5V Ol 54b ALPHA Microprocessor," Proc. of ISSCC 1999, WP 25.4.

[7] G. Northrop, et al, "600 MHz G5/390 Microprocessor," Proc. of ISSCC1999, MP 5.2.

[8] S. Hesley, et al, "A 7th-Generation x86 Microprocessor," Proc. of ISSCC1999, MP 5.4.

[9] P. Barnes, "A 500MHz 64b RISC CPU with 1.5 MP On-Chip Cache," Proc. of ISSCC 1999, MP 5.1

[10] S. Fischer, et al, "A 600MHz 1A-32 Microprocessor with Enhanced Data Streaming for Graphics and Video," Proc. of ISSCC 1999, MP 5.7.

[11] J. Schutz, et al, "A 450MHz IA32 P5 Family Microprocessor," Proc of ISSCC 1998, FP 15.4.

[12] R. Stephany, et al, "A 200MHz 32b 0.5W CMOS RISC Microprocessor," Proc. of ISSCC 1998, FP 15.5.

[13] N. Rohrer, et al, "A 480MHz RISC Microprocessor in a 0.12µm L_{elf} CMOS Technology with Copper Interconnects," Proc. of ISSCC 1998, FP 15.6.

[14] K. Suzuki, et al, "A 500MHz 32b 0.4µm CMOS RISC Processor LSI," Proc. of ISSCC 1994, TP 12.7. M10 D. Dobberpuhl, et al, "A 200MHz 64b Dual-Issue CMOS Microprocessor," Proc. of ISSCC 1992, TA 6.2.

[15] D. Allen, et al, "A 0.2µm 1.8V S01 550MJz 64b Power PC Microprocessor with Copper Interconnects," Proc. of ISSCC 1999, WP 25.7

[16] D. Draper, et al, "Circuit Techniques in a 266-Hz MMX Enabled Processor," Journal of Solid State Circuits, Vol. 32, No. 11, pp 1650-1664.

[17] P. Turgeon, et al, "Storage Hierarchy to Support a 600MHz G5 S/390 Microprocessor," Proc. of ISSCC 1999, MP5.3.

[18] G. Braceras, et al, "A 940 MHz Data-Rate 8Mb CMOS SRAM," Proc. of ISSCC 1999, TA 11.5.

[19] C. Zhao, et al, "An 18Mb, 12, GB/s CMOS Pipeline-Burst Cache SRAM with 1.54Gb/s/pin," Proc. of ISSCC1999, TA 11.6.

[20] T. Saito, et. al, "A 622Mb/s CMOS ATM Switch Access LSI with Maintenance Cycle Interleaved Pipeline Architecture," Proc. of ISSCC1999, TA 9.4.

[21] G. Kizhepat, "A Single-Chip MPEG-2 Video Audio and System Encoder," Proc. of ISSCC 1999, TP 15.6.

[22] F. Lu, et al, "A Single-Chip Universal Burst Receiver for Cable Modem/Digital Cable-TV Applications," IEEE 2000 Custom Integrated Circuits Conference," pp. 311 – 314.

[23] J. Lachman, et al, "A 500 MHz 1.5MB Cache with On-Chip CPU, " Proc. of ISSCC 1999, TA 11.2.

[24] H. Nambu, et al, "A 1.8ns Access, 550MHz 4.5Mb CMOS SRAM," Proc. of ISSCC 1998, SP 22.7.

[25] B. Bateman, et al, "A 450MHz 512kB Second-Level Cache with a 3.6GB/s Data Bandwidth," Proc. of ISSCC 1998, SP 22.6.

[26] K. Kulargi, et al, "A Microprocessor with a 128b CPU, 10 Floating-Point MAC's, 4 Floating-Point Dividers, and an MPEG2 Decoder," Proc. of ISSCC 1999, TP 15.1.

[27] T. Saito, et. al, "A 622Mb/s CMOS ATM Switch Access LSI with Maintenance Cycle Interleaved Pipeline Architecture," ISSCC 1999, TA 9.4.

[28] G. Kizhepat, "A Single-Chip MPEG-2 Video Audio and System Encoder," Proc. of ISSCC 1999, TP 15.6.

[29] F. Lu, et al, "A Single-Chip Universal Burst Receiver for Cable Modem/Digital Cable-TV Applications," IEEE 2000 Custom Integrated Circuits Conference," pp. 311 – 314

[30] W. Maly, H. Jacobs and A. Kersch, "Estimation of Wafer Cost for Technology Design," in Proc. IEDM-93, Washington D.C., Dec. 5-8, 1993, pp. 35.6.1 - 35.6.4.

[31] Y. Fei, P. Simon and W. Maly, "New Yield Models for DSM Manufacturing" IEDM 2000, San Francisco, December 10-13, 2000 pp. 845-848

[32] J. Khare, D. Feltham and W. Maly, "Accurate Estimation of Defect-Related Yield Loss in Reconfigurable VLSI Circuits," IEEE Journal of Solid State Circuits, Feb. 1993, No. 2, pp. 146-156.

[33] M. Niewczas, W. Maly and A.J. Strojwas, "An Algorithm for Determining Repetitive Patterns in Very Large IC layouts", in IEEE Transactions on CAD April 1999, Vol. 18, No. 4, pp. 494-501.

[34] D. Ciplickas, A. Strojwas, X. Li, R. Vallishayee, and W. Maly, "Advanced Yield Learning Through Predictive Micro-Yield Modeling," Proc. of ISSM '98, Tokyo, Japan, pp. 315-318, Oct. 1998.

[35] W. Maly, H. T. Heineken, J. Khare, and P. K. Nag, "Design-Manufacturing Interface: Part I – Vision," Proc. of DATE '98, pp. 550 – 556, Paris, France, Feb. 23-26, 1998.

Table A1.

#	Die size [cm^2]	**Feature size [µm]**	Total # of Tr.	Transistors in Mem.	Transistors in Logic	Area [cm^2] Mem.	Area [cm^2] Logic	s_d **Mem.**	s_d **Logic**	Type of device
1	0.48	**1.5**	0.18		0.18		0.48		**118.5**	CPU
2	0.8	**0.8**	1.2		1.2		0.8		**104.1**	CPU
3	2.88	**0.8**	3.1	0.1	3	0.03	2.85	**46.88**	**148.4**	Pentium (P5)
4	0.9	**0.35**	3.1		3.1		0.9		**237**	Pentium (P5)
5	3.06	**0.6**	5.5		5.5		3.06		**154.5**	Pent. Pro
6	1.95	**0.35**	5.5	0.77	4.73	0.05	1.9	**53.15**	**327.9**	Pent. Pro
7	1.41	**0.35**	4.5		4.5		1.41		**255.7**	Pentium
8	2.03	**0.35**	7.5	1.23	6.28	0.08	1.95	**53.15**	**253.6**	Pent.II (P6)
9	1.31	**0.25**	7.5	1.23	6.28	0.04	1.27	**52.08**	**323.8**	Pent. II (P6)
10	0.95	**0.25**	4.5		4.5		0.95		**337.7**	Pent. MMX
11	1.23	**0.25**	9.5	-	9.5		1.23		**207.1**	Pentium III
12	1.61	**0.35**	4.3	1.15	3.15	0.06	1.47	**42.59**	**380.9**	K5
13	1.62	**0.35**	8.8	3.1	5.7	0.18	1.44	**47.4**	**206.2**	K6 (Mod. 6)
14	0.68	**0.25**	8.8	3.1	8.8	0.08	0.6	**41.29**	**109.0**	K6 (Mod. 7)
15	0.68	**0.25**	9.3		9.3		0.68		**116.9**	K6-2 (Mod.
16	1.35	**0.25**	9.3		9.3		1.35		**232.2**	K6-2 (Mod.
17	1.84	**0.18**	22	6	16	0.1	1.74	**51.44**	**335.6**	K7
18	1.2	**0.5**	2.8		2.8		1.2		**171.4**	Power PC
19	1.95	**0.5**	3.6		3.6		1.95		**216.6**	Power PC
20	1.62	**0.35**	12	6	6	0.28	1.34	**38.1**	**182.3**	Power PC
21	2.72	**0.35**	8		8		2.72		**277.5**	S390 Ca.
22	0.67	**0.25**	6.35		6.35		0.67		**168.5**	Power PC
23	1.39	**0.22**	34	24	10	0.5	0.9	**43.43**	**185.0**	PowerPC
24	2.1	**0.25**	25	18	7	0.53	1.58	**46.76**	**360.2**	G5
25	0.67	**0.2**	6.5	3	3.5	0.09	0.58	**72.92**	**416.0**	PowerPC
26	0.4	**0.15**	6.5	3	3.5	0.05	0.35	**74.07**	**444.4**	PowerPC
27	0.83	**0.18**	10.5	3.1	7.1	0.18	0.65	**174.2**	**280.3**	PowerPC
28	0.75	**0.35**	2.5	1.15	1.35	0.07	0.68	**49.6**	**411.8**	RISC
29	2.09	**0.25**	9.66	4.9	4.77	0.5	1.59	**163.2**	**533.3**	Alpha (SOI)
30	1.34	**0.5**	2.4		2.4		1.34		**223.3**	Media GX
31	1.94	**0.35**	6		6		1.94		**263.9**	6x86MX
32	1.01	**0.28**	5.7		5.7		1.01		**224.8**	RISC CPU
33	0.6	**0.28**	3.3		3.3		0.6		**231.9**	RISC CPU
34	4.69	**0.25**	116	92	24	2.3	2.38	**40**	**158.6**	PA RISC
35	0.34	**0.18**	7.2	5.2	2	0.15	0.19	**89.03**	**293.2**	MIPS64TM
36	0.2	**0.13**	7.2	5.2	2	0.09	0.11	**100.1**	**331.3**	MIPS64TM
37	2.76	**0.22**	12.9	3.7	9.2	0.16	2.6	**89.35**	**583.9**	MAJC 52000
38	1.77	**0.18**	47	34	13	0.6	1.17	**54.47**	**278.2**	z900
39	3.97	**0.18**	152	138	14	2.77	1.2	**61.88**	**264.5**	Alpha
40	0.72	**0.6**	0.8		0.8		0.72		**250.2**	DSP
41	2.26	**0.4**	12		12		2.26		**117.5**	DSP
42	1.78	**0.35**	4		4		1.78		**363.0**	DSP
43	2.72	**0.5**	2		2		2.72		**544.5**	MPEG-2
44	2.13	**0.4**	3.79		3.79		2.13		**350.9**	MPEG-2
45	1.55	**0.35**	3.1		3.1		1.55		**408.1**	MPEG-2
46	0.37	**0.35**	1		1		0.37		**299.2**	ASIC M.
47	3	**0.25**	10		10		3		**480**	ASIC T. Com
48	2.38	**0.18**	10.5		10.5		2.38		**699.5**	Video Game
49	2.25	**0.35**	2.4		2.4		2.25		**765.3**	ATM