CHALLENGES IN PHYSICAL CHIP DESIGN

RALPH H.J.M. OTTEN
Eindhoven University of Technology,
Eindhoven, The Netherlands
otten@ics.ele.tue.nl

PAUL STRAVERS
Philips Research,
Eindhoven, The Netherlands
paulus.stravers@philips.com

Introduction

Chip industry obeys a number of laws, various kinds of laws. Mathematical laws if accurate models can be formulated, physical laws, especially solid state physics, obtained by observation and induction, chemical laws pertinent for the manufacturing processes, economical and judicial laws that concern such industries. The most famous and most cited law of chip industry is the one that Gordon Moore formulated in 1964 after observing trends in the then very young field of integration of electronic circuits. Mathematically formulated, *Moore's law* reads as follows:

$$\frac{dN}{dt} \propto N, \tag{1}$$

where N is the maximum number of devices on a single chip. The proportionality constant is called the *moore exponent* which according to Moore, with years as the unit of time, equaled 0.7.

An even older law, also formulated after observing properties of early logic circuitry in computers, is known as Rent's rule.

$$\frac{dT}{dG} \propto \frac{T}{G}, \tag{2}$$

where T is the number of external connections of a part containing G gates. The proportionality constant is called the *rent exponent*.

Both laws seem to hold surprisingly accurate. Moore's law soon became the ultimate guideline for setting targets in the chip industry. In a sense it has thus become a self-fulfilling prophesy, although it is still remarkable that that industry was able to satisfy such ambitious goals. Rent's rule went through stages of neglect and popularity. A convincing case for the usefulness of such a law came with IBM's need for wire space estimations for gate arrays, as documented in the Donath's landmark paper [5]. Both, the moore and rent exponents, had to be tied to a more specific class of circuits. The recent report [17] of ICE established a moore exponent of 0.2 for microprocessors and 0.4 for memory (figure 1). Bakoglu [1] showed rent exponents between 0.12 and 0.63, distinguishing dynamic and static memory, microprocessors, gate arrays and high-speed processors.

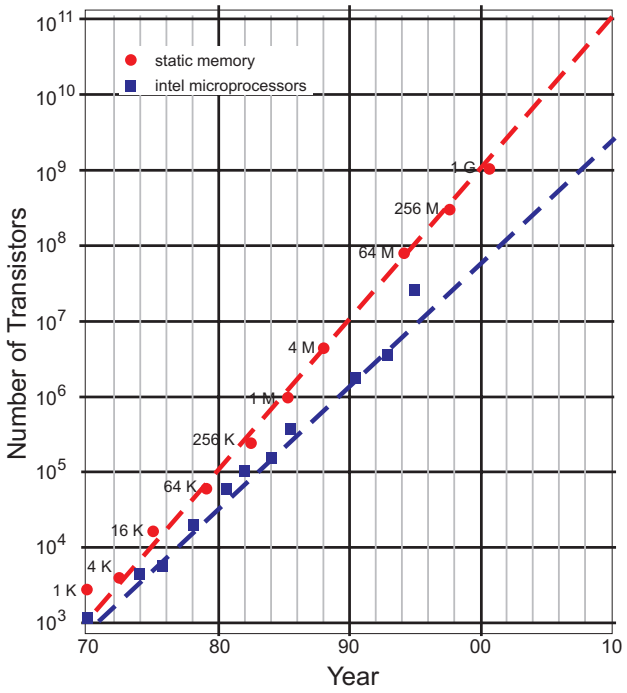

Figure 1: The complexity growth.

The same report [17] provides data that shows similar laws characterizing the scaling of dimensions over the last forty years. Figure 2, based on that source, shows that feature size, junction depth, and gate oxide thickness not only obey a similar law; they even share the same exponent, which of course is negative.

All these laws fit in the generic form

$$\frac{dU}{dV} \propto \frac{f(U)}{h(V)}$$

and we will call them *straverius laws*. They have an integral

$$p(U,V) = \int \frac{dU}{f(U)} - p \int \frac{dV}{h(V)} \tag{3}$$

where p is the particular exponent, and for many simple functions f and g they can be solved explicitely.

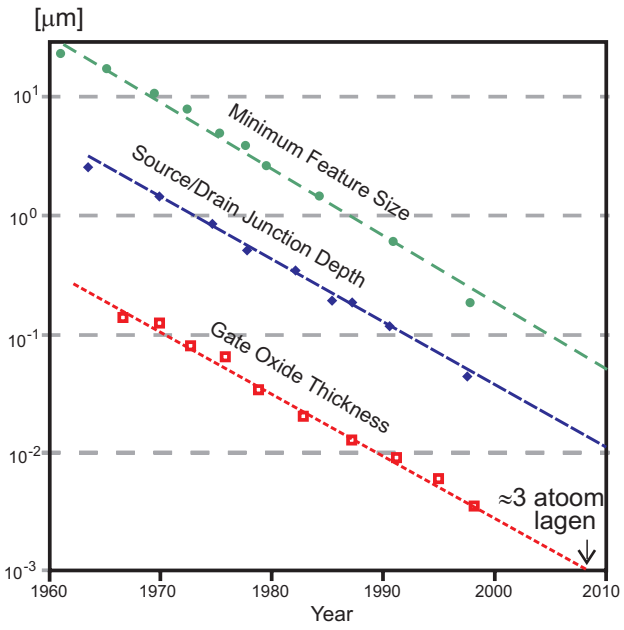

Figure 2: The scaling of the dimensions.

1 Confinement

1.1 Memory-to-compute ratio

Embedded computer chips exhibit a trend where with every new generation an increasing percentage of the chip area is dedicated to memory, while an ever decreasing percentage of the chip area is dedicated to computational structures.

This observation can be rationalised as follows. It has long been known [7] that a balanced computer system is equipped with an amount of memory that is proportional to the computational power of the processing unit. Gene Amdahl observed that mainframe computers follow the rule of 1 memory byte per instruction per second (i.e. a 10 MIPS CPU would come with 10M bytes of RAM).

To see how this rule affects the ratio of computational resources to memory resources on a chip, we note that each new generation of semiconductor process technology reduces the area of both computational and memory structures by a factor A, while increasing the maximum achievable clock frequency of a chip by a factor S.

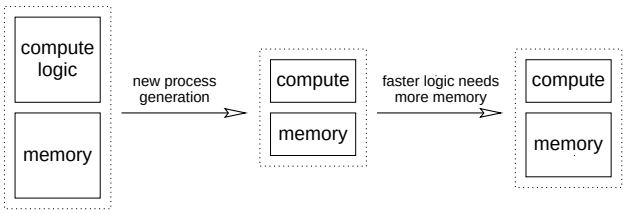

Figure 3: The effect of shrinking on the memory-to-compute ratio.

We introduce ρ, the ratio of memory area to compute area. For the left hand side of figure 3 this number is

$$\rho_{t_0} = \frac{M_{t_0}}{C_{t_0}} \tag{4}$$

where M_{t_0} represents the memory area and C_{t_0} the compute area. After some time (dictated by the moore exponent) a new process generation is introduced and we shrink both C_{t_0} and M_{t_0} by a factor A. We maintain the property of a balanced computing system by matching the now S times faster compute logic to an S times larger memory. The result is depicted on the right hand side of Figure 3 where we find

$$\rho_{t_1} = \frac{S(M_{t_0}/A)}{C_{t_0}/A} = S\rho_{t_0} \tag{5}$$

Reformulating this equation in the differential form, that we have introduced for straverius laws, assuming that new technology generations come with regular intervals, as was the case for $cmos$-technology up to now, we find

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} \propto \rho \tag{6}$$

Note that independent from its initial value, ρ_t increases exponentially over time as long as new silicon process generations are introduced with $S > 1$. For $cmos$-technology in the past decades, the value of S has been close to 1.4.

This law points to the conclusion that memory will increasingly dominate the available chip area in the future, while compute logic will necessarily be confined to a small fraction of the available on-chip silicon area. Also, because the compute logic is getting so small and the memories so big, the average wiring distance between the two is becoming relatively large, resulting in increased memory access latency, especially when expressed as the number of equivalent compute cycles. In section 2 we investigate two possible ways to deal with this confining technology trend.

1.2 Buffer area

Global wires are defined to be interconnections whose delay can be improved by inserting buffers. It was shown in [11]. that the delay then exceeds a *critical delay*, which is a process constant equal for all wiring layers (if the buffers are made by the same processing steps), and that the length exceeds a certain *critical length* that does depend on the layer:

$$l_{crit} = \frac{l}{n_{opt}} = \sqrt{\frac{br_o c_o\,(1 + c_p/c_o)}{arc}}. \tag{7}$$

This length is the optimal distance between buffers if the delay is to be minimized. This was derived with a model as in figure 4 where $R_{tr} = r_o/s$, $C_p = sc_p$ and the input to the next buffer $C_L = s \cdot c_o$. where subscript o and p refer to the minimum-size buffer. a and b are modeling

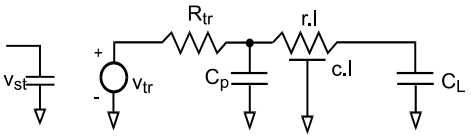

Figure 4: Generic restoring buffer model

constants (to be tuned, but close to 0.4 and 0.7 in this situation). Simple calculus then produces this optimal distance as well as an optimal size for the buffers:

$$A_{opt} = A_o \cdot s_{opt} = A_o \sqrt{\frac{r_o\, c}{c_o\, r}}.$$

Dividing the optimal size by the critical length gives the buffer area per unit length of an optimally buffered line:

$$\alpha = A_o \frac{c}{c_o} \sqrt{\frac{a}{b\,(1 + c_p/c_o)}}. \tag{8}$$

This is a remarkable result because the area is independent of the line resistance! The delay per unit length does depend on r:

$$2\sqrt{r c r_o c_o}\left(b + \sqrt{ab\left(1 + \frac{c_p}{c_o}\right)}\right)$$

which still makes a trade-off necessary when determining the cross-section of a wire.

In order to calculate the total area taken by buffers we need to know the wire length distribution of the chip. Suppose its probability density function is given by $P(l)$ then the buffer area is given by

$$\alpha N_I \int_{l_{crit}}^{l_{max}} l P(l) dl,$$

N_I being the total number of wires.

$P(l)$ is usually obtained by making a model with some simplifying assumptions and requiring that 2 must be satisfied. Concise derivations of *weibull*-distributions (the two-dimensional case is however not translation invariant!) and extensive calculations resulting in very long expressions (which is no objection when generating buffer area by computer) have been produced, but there is some agreement that the early result in [6]

$$P(l) = \frac{l^{2r-3}}{\int_{l_{min}}^{l_{max}} l^{2r-3}}$$

captures the essence. Whatever is used, the increase in buffer area percentage-wise is tremendous, not in the last place because buffers become very large for deep submicron circuits[1].

[1]After this tutorial was submitted to ICCAD, another motivation for multilayer paradigms was presented at DAC2000: S.J.Souri, e.a., "*Multiple Si layer ICs: motivation, performance analysis and design implications*" (Proceedings DAC2000,pp 213-220). They also make the buffer area argument, and show results where the area is larger when the rent exponent is smaller. This is strange, but their model is not explained.

1.3 Current drive capability

The increase in complexity predicted by Moore and realized by the industry, was possible not in the last place possible because the increase in current drive capability $I_{D,SAT}/W$ over several technology generations. When feature sizes get very small and voltages scale at a slower rate, the electrical field becomes high. At high values for the field strength the mobility of the carriers can no longer be considered constant, and the dependence of the drift velocity on the electric field will thus depart from the linear relationship observed under low-field conditions. A semi-empirical formula for the drift velocity of the charge carriers is proposed in [4]:

$$v_d = \frac{v_{sat}}{\left[1 + \left(\frac{E_c}{E}\right)^{\gamma}\right]^{1/\gamma}} \tag{9}$$

The saturation velocity v_{sat} can be considered in a first approximation the same both for holes and for electrons. E_c is the critical electrical field and the coefficient γ varies with the type of charge carriers: for holes close to 1, while for electrons close to 2.

Based on this formula a general linear dependency between the drain saturation current and the drain-source saturation voltage was derived in [2], a dependency valid for all transistor lengths (L) and for p- as well as n-transistor types.

We recall here some elementary relations: for the drain current under the drift model,

$$\mathbf{I_D} = -W|Q_I|\mathbf{v_d} \tag{10}$$

and the available mobile charge in the channel:

$$Q_I(y) = -C_{ox}\big(V_{GS} - V_T(V_{SB}) - (1+\delta)V_{CS}(y)\big) \tag{11}$$

For the drift velocity we use (9) with $E = dV_{CS}/dy$. The contact-to-source bias $V_{CS}(y)$ at an arbitrary point C in the channel is a monotonically increasing function of y. The solution at the two ends of the channel satisfies the boundary conditions: $V_{CS}(0) = 0$ and $V_{CS}(L) = V_{DS}$. Substituting (11) and (9) into (10) yields the following expression for the drain current in triode operation mode:

$$I_D = \frac{W C_{ox} v_{sat}\big(V_{GS} - V_T(V_{SB}) - (1+\delta)V_{CS}(y)\big)}{\left[1 + \left(\frac{E_c}{E}\right)^{\gamma}\right]^{1/\gamma}} \tag{12a}$$

$$= \frac{W C_{ox}(1+\delta) v_{sat} E}{\left(E^{\gamma} + E_c^{\gamma}\right)^{1/\gamma}}\left[\frac{V_{GS} - V_T(V_{SB})}{1+\delta} - V_{CS}(y)\right] \tag{12b}$$

For simplicity let us introduce the following notations:

$$b = \frac{V_{GS} - V_T}{1+\delta}, \quad c = E_c, \quad I = \frac{I_D}{W C_{ox}(1+\delta) v_{sat}},$$

$$u = V_{CS}(y) \quad \text{and} \quad V = V_{DS}$$

to get from (12b):

$$I = (b-u)\frac{du/dy}{\left[(du/dy)^{\gamma} + c^{\gamma}\right]^{1/\gamma}} \tag{13}$$

The above expression can,by separating the variables, be rewritten into

$$Ic\,dy = \left[(b-u)^{\gamma} - I^{\gamma}\right]^{1/\gamma} du \tag{14}$$

which would allow us to find an implicit relation between the drain current and the drain-to-source voltage in triode region by integrating (14) over the channel length:

$$F(V,I) = \int_0^V \left[(b-u)^{\gamma} - I^{\gamma}\right]^{1/\gamma} du = cLI \tag{15}$$

Note that $F(V,I)$ depends on V only, since I is uniquely determined by V: $I = I(V)$. When the transistor operates at the border between triode and saturation regime, the first derivative of the drain current with respect to V_{DS} equals to zero, that is $\frac{\partial I}{\partial V} = 0$. If we now differentiate the extreme sides of (15) we get:

$$\frac{\partial F}{\partial V} + \frac{\partial F}{\partial I}\cdot\frac{\partial I}{\partial V} = cL\frac{\partial I}{\partial V} \longrightarrow \frac{\partial I}{\partial V} = \frac{\partial F/\partial V}{cL - \partial F/\partial I} \tag{16}$$

We are looking for the curve Γ in the $i-v$ plane such that it contains exactly the points where $\frac{\partial I}{\partial V} = 0$, and therefore where $\frac{\partial F}{\partial V}(I,V) = 0$. As follows from the definition (15) of $F(V,I)$ we have

$$\frac{\partial F}{\partial V} = \frac{\partial}{\partial V}\int_0^V \left[(b-u)^{\gamma} - I^{\gamma}\right]^{1/\gamma} du = \left[(b-V)^{\gamma} - I^{\gamma}\right]^{1/\gamma} \tag{17}$$

This means that on Γ

$$i = b - v. \tag{18}$$

So, we found that for the general case the triode-saturation separation is given by the linear relation:

$$I_{D,SAT} = WC_{ox}v_{sat}[V_{GS} - V_T - (1+\delta)V_{DS,SAT}] \tag{19}$$

Expression (19) can be seen as the separation line between triode and saturation regions as in figure 5. It illustrates that no matter how short channels are the saturation current per

unit width, or current drive, is bounded above by $v_{sat}C_{ox}(V_{GS} - V_T)$.
This maximum achievable current from a transistor is not dependent on the channel length L. Consequntly, in the quest for higher speed through the relative increase of the current drive by down-scaling of the transistor length, there is an inherent limitation.

The drain current in triode region is the implicit solution of equation (15). For a p-device the charge carriers in the channel are holes, and, as mentioned before the γ coefficient takes values close to 1. In that case an explicit expression for the drain current is easily derived. For $\gamma \in (1,2]$ it leads to β-functions and it is better to use numerical software to generate the $I-V$ characteristics of n-devices, as was done for figure 5 where $\gamma \approx 2$. We remark that, for

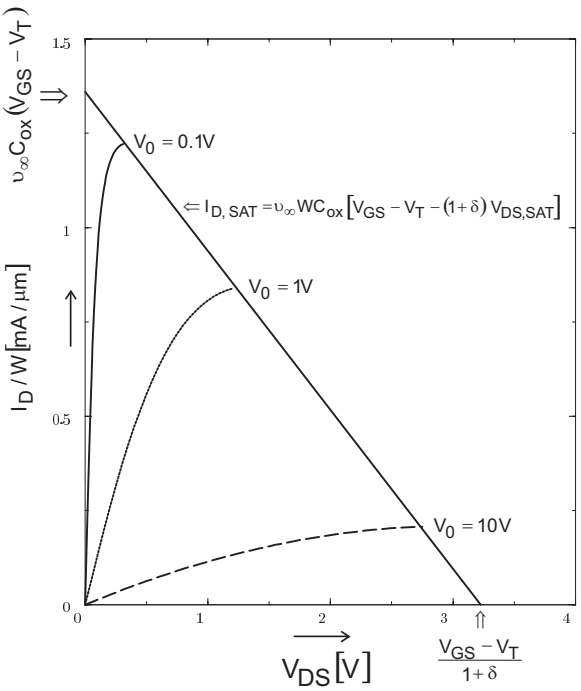

Figure 5: n-transistor I-V characteristic and the triode/saturation separation for a specific V_{GS} and three various technology generations ($V_o = L \cdot v_{sat}/\mu_o$)

a given technology generation, the current drive of an n-transistor comes closer to the $v_{sat}C_{ox}(V_{GS} - V_T)$ upper boundary than the current drive of a p-transistor.

This shows that due to the velocity saturation effect the current drive no longer improve significantly by scaling the transistor dimensions below a micron. Not only that the current drive improvement saturates, but also the capacitive load that a gate has to drive increases relative to the gate strength (as another detrimental effect of the interconnect lateral capacitance).

2 Escape routes

2.1 Homogeneous processors

Revisiting (6) we try to derive some of the consequences for future system-on-chip architectures. In this section, we focus on using (6) as a weapon against the increasing design complexity implied by (1) (Moore's Law). The issue at stake is that it is becoming increasingly hard to de-

sign reliable systems-on-chip with the hundreds of millions transistors that fit on new chip generations. The problems include high design costs, lack of engineers, slow simulators, and difficulties to manage these very complex design projects. At the same time globalisation of the economy and bored consumers put an increasing pressure on companies to bring new products to market in a very short time.

For these reasons it is of paramount importance to develop a system-on-chip methodology that scales trivially with (1). A simple approach could be to repeatedly place a self-contained computing unit on a chip until the available silicon area fills up. The units are then linked through a high-speed communication network so that the aggregate of compute units can work cooperatively on one or more algorithms.

Traditionally the problem with this type of system architectures is that the compute units must be sufficiently general-purpose, or otherwise the system is not usable in a sufficiently wide range of applications. But general-purpose computing engines often lack several orders of magnitude behind special-purpose hardware in terms of computational efficiency, i.e. speed and power consumption.

This is where (6) comes to rescue:

> **Postulate 1.** In a technology where memories are dominant and computational structures are cheap, we can afford to have instances of *many types* of computational units close to every memory bank.

This naturally leads to an architecture consisting of *clusters*, where each cluster consists of a memory plus a heterogeneous mix of computational units. If the mix is really heterogeneous, then each cluster can perform well on a wide range of applications, even though this means that only part of the compute units in each cluster is actually put to work.

Note that compute units in a cluster can have very specific functionality, for example they could include a complete MPEG-2 video decoder or a 3D graphics rendering engine. Even though such units are expensive by todays measure, according to Postulate 1 we can afford to instantiate them in a cluster because memory will dominate future chip area anyway and therefore compute logic becomes relatively cheap.

Other compute units in a cluster can be more general purpose, for example microcontrollers, DSPs and maybe even a few FPGA-like units can be used to implement functions that don't happen to be available as precooked engines in the cluster. Also, the microcontrollers can be used to manipulate control registers of other special-purpose compute engines in the cluster and to setup their input and output streams.

In this way, configuring a cluster for a specific task can be done after chip manufacturing and could in fact be done in the field or at the customer site. The computational efficiency of a cluster can be very high, despite its being field-configurable, because usually most of the work can be handled by one or more dedicated compute engines in the cluster, provided the cluster is truly heterogeneous and covers a wide range of applications.

This then resolves the ever recurring arguments against programmable and configurable systems: that their computational efficiency is at least one and often several orders of magnitude lower than dedicated solutions, resulting in much higher power consumption and lower computing speeds.

It also solves the problem of simulating large systems-on-chip. Because the chips are matched to an application after fabrication, the system functionality can be verified using the actual silicon instead of using HDL simulators that are easily a billion times slower and less accurate than the real thing. Of course, real-time debugging is an important issue.

An interesting consequence of the cooperating heterogeneous multi-purpose clusters is that now there is no need anymore for one cluster on a chip to have a different composition than any other cluster on that same chip. Since every cluster is multi-purpose, we assign specific tasks to the clusters based on their communication pattern, i.e. tasks that communicate a lot are assigned to adjacent clusters, or maybe even to the same cluster if enough resources inside that cluster are available. This ability of clusters to efficiently execute a wide range of tasks therefore is very helpful in avoiding long communication latencies and reducing power consumption for inter-cluster communications.

2.2 Multilayer processors

A different way of dealing with the confinement implied by (6) is to simply put most of the memory in different layers of the chip. In this way the silicon area dedicated to compute logic can scale with (1), escaping the confinement predicted by (6).

When the memory-to-compute ratio passes a certain threshold then a dedicated memory layer is added to the 3D stack. The wires run vertically through the stack and therefore their average length is significantly reduced compared to the 2D case. This is good news, because the execution time of many important applications depends heavily on the memory access *latency*, i.e. the time it takes to do a round trip from the compute logic to memory and then back.

In some multilayer technologies the vertical wire density is high enough (i.e. more than one via in 10,000 square featur sizes in 0.1μ technology, although thse via do not scale very well yet) to enable very wide buses running between layers. This means that very high bandwidths can be sustained between the compute layer and the memory layers. This of course is vitally important, and in combination with short latency provides an excellent memory subsystem with very good performance characteristics.

In [12] a study is presented that compares a multilayer implementation of a RISC processor to a conventional implementation. The conclusion is that a multilayer imple-

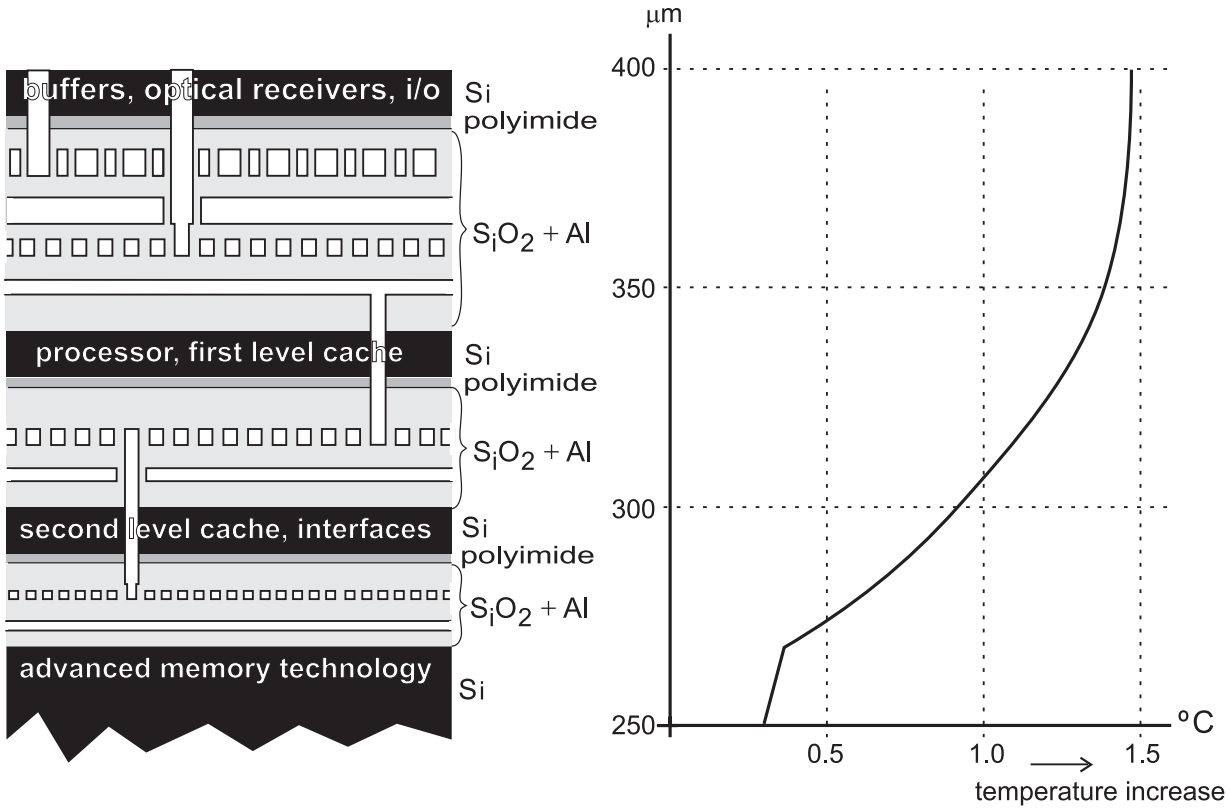

Figure 6: The layer dedication and the thermal analysis from [14]..

mentation can benefit significantly from the low latency, high bandwidth connection to the first level and second level caches. In [3] an analysis is presented showing that many of the techniques used to tolerate growing memory latencies do so at the expense of increased bandwidth requirements. Clearly, a multilayer microprocessor implementation that improves both latency and bandwidth can significantly relax the off-chip bandwidth requirements, resulting in lower pin counts and cheaper packages.

Although these studies focus on microprocessor implementations rather than complete systems-on-chip, the same arguments apply to much more complex architectures like the homogeneous multiprocessor presented in section 2.1. In this case the first level caches and local scratch memories could be allocated to a layer on top of the actual compute layer. On top of the caches and scratch memories, one or more layers can be stacked with DRAM, as dictated by the law in (6).

3 Liberation

3.1 Filling the layers

In an early paper [9] it was already stated that adding wiring layers could not reduce the essential interconnect complexity of circuit integration. The author also suggested that *flashing the clock* on the chip would not only be a temporary relief, but would also solve skew problems. With an active layer on top not only this would be feasible, but also selectivity with respect to particular clock phases is within reach. The same layer can be used for housing the buffers to speed up global interconnect as suggested in [10]. Optimal buffering then depends on the properties of the top layers: critical delay (τ_{crit}) depends on the active components, while the optimum segmentation (l_{crit}) depends also on the properties of interconnect in the *global tier*. Supply line shielding yields reliable interconnect characterization. This increases the line capacitance, and consequently the buffer area in the top layer and the power consumption of the global communication.

The processors, each with their own instruction and data caches fill up the next layer, in a regular formation, but each individualized to perform the operations to be assigned efficiently. Four wiring layers, the global tier with segmentation and buffering, and and a tier for more local interconnect is in between. The processing layer is without doubt producing the most heat. Experience reported in literature made clear that this is not a major problem [14][2]. The layer still suffers under 6 but access times to memory on other layers is certainly improved.

The other two layers are dedicated memory: to second-level cache and interface electronics for controlling main

[2]The heat simulation in [12] is also a four-layer processor, but the layers are not specified. But any different ordering in our case would only increase the problem if any.

memory, and to main memory. The latter is the base active layer, made in the most advanced technology, using agressive design rules.

3.2 The supporting technology

For more than twenty years chip technology research has worked on so-called three-dimensional integration. However, over this periode Moore's prediction could be fulfilled without having to break free from the single-active-layer confinement. In section 1 we discussed but three fundamental reasons why in the near future chips with a single active layer and conventional formation of the active components cannot maintain the growth in functionality and performance of the past decades. In section 3.1 an advantageous usage of four active layers has been outlined. The question whether this is economically justified, or even technogically feasible, was not touched.

Several research groups have shown fabrication technologies for producing chips with active components outside the base active layer. Roughly they can be classified as *growing* and *stacking* techniques. In the first category we find most of the early true integration solutions: recrystallization, layer growth and seeding. They have as a major disadvantage that the base layer has to undergo all those additional process cycles of heating and cooling, which will degrade the properties of the components in that layer. In the proposal of section 3.1 this is the most sensitive layer, produced with extreme agressiveness. This is clearly unacceptable. Recently low-temperature technologies for adding components outside the base layer have been published, but they are still far from "manufacturability"¿

Stacking implies the separate fabrication of active layers, later to be combined with each other. They have the obvious advantage of much improved control over the properties of the components. The individual layers do not even have to be be produced in the same technology. One of the first multilayer processors was made by transferring a soi-film on top of a bulk-silicon $cmos$-chip [16]. There is also no obvious limit to how many layers can be stacked by such a process. The main disadvantage is that aligning the layers with respect to each other. The same exercise used via's of 6μ on each side, and scalability was not expected soon.

But a number of advantages were easily recognized:

1. Interconnection lengths were considerably shorter, which in their case required proper partitioning. Folding datapath over more layers and determining the optimum crossing points can shorten cycle time considerably.

2. The total footprint was of course much smaller which is beneficial for yield and/or allows larger chips.

3. As mentioned, different technologies can quite easily be realized on the same chip as long as they can allow contact via's on both sides. The quality of components in one technology is not compromised because of favoring the quality of components in another technology. In the proposal optical receivers were included. Although buffers were planned in the same layer, their properties are not very critical.

Key remains the trade-off between via size and accurate alignment. Via's are expected to be big, requiring quite a bit of area overhead. The alignment requirements will demand strong geometrical constraints in laying out the individual layer. In [16] they made one layer the dictator, in the dedicated layer proposal, the enforced regularity of all but the top layer forces the placement.

Heat is not expected to be a problem for multilayer chips. In the proposal the heat generators are the top two layers, and all layers were targeted for bulk silicon processing. If several layers of soi-technology are used overheat might occur and should be investigated. In general, according to the relation of Wiedemann-Franz, good electrical conductors are good thermal conductors, but layers cannot be completely shielded by electrically conducting layers.

3.3 The supporting computer-aided design

Obviously, the escape routes proposed in 2 require a completely different design flow. Homogeneous processors do not benefit much from parts of a traditional flow. The emphasis should be more on modeling applications as networks of communicating processes in a suitable specification language [8]. Equally important is reuse of specification software, considering the short life spans of integrated circuits and the demand for short paths to the market.

General multilayer designs require complete new layout synthesis tools. Placement is obsolete ("*modern placement is floorplan design plus legalization!*") and even floorplan design for each layer not adequate because of the strong geometrical constraints. Wire planning will be more of a must, but has to acquire a more precise meaning in this application.

The challenges posed by the unavoidable escape routes, mentioned in this paper or still to be conceived, to break free from the confinements of conventional large scale integration methodologies, is the topic of this tutorial, for these methodologies will not carry 1 over the another decade!

References

[1] H.B. Bakoglu, "Circuits, interconnections, and packaging for vlsi", Addison-Wesley Pub Co, 1990

[2] S. Bruma, "Into deep submicron: a simulation perspective", PhD-thesis, Delft University of Technology, Delft, The Netherlands

[3] D. Burger and J.R. Goodman and A. Kägi, "Memory Bandwidth Limitations of Future Microprocessors",

23rd annual International Symposium on Computer Architecture (ISCA-96), 1996, pp.78–88.

[4] D.M. Caughey and R.E. Thomas, “Carrier mobilities in silicon empirically related to doping and field”. Proceedings IEEE, 55, 2192 (1967).

[5] W.E. Donath, “Placement and average interconnection lengths of computer logic” *IEEE Transactions on Circuits and Systems*, CAS-26, 4, April 1979

[6] W.E. Donath, “Wire length distribution for placements of computer logic”, *IBM Journal of Research and Development*,25, 3, May 1981, pp. 152-155.

[7] H. Garcia-Molina and L.R. Rogers, “Performance through memory”, *Proceedings of the ACM conference on Measurement and modeling of computer systems*,May 1987, pp. 122-131

[8] E.A. de Kock, G. Essink, W.J.M. Smits, P. van der Wolf, J.-Y. Brunel, W.M. Kruijtzer, P. Lieverse, K.A. Vissers, “YAPI: application modelling for signal processing systems” *Proceedings of the 37th Design Automation Conference*, Los Angeles, Ca, USA, June 2000, pp. 402-405.

[9] R.H.J.M. Otten “Complexity and diversity in ic layout design” *International Conference on Circuits and Computers*, New York, NY., USA, pp 764-767, October 1980.

[10] R.H.J.M. Otten, “Global wires harmful?”, Proc. 1998 International symposium on physical design’ Monterey, CA, USA, April 1998,pp.104-109.

[11] R.H.J.M. Otten and R.K. Brayton, “Planning for performance”, Proc. 1998 Design Automation Conference. San Fransisco, CA,,USA, June 1998, p. 122-127.

[12] M.B. Kleiner, S.A. Kühn, P.Ramm and W. Weber, “Performance improvement of the memory hierarchy of risc-systems by application of 3-d technology”, *IEEE Transactions on Components, Packaging and Manufacturing Technology*, Part B, vol 19, 4, November 1996., pp 709-718

[13] S.A. Kühn, M.B. Kleiner, P.Ramm and W. Weber, “Performance modeling of the interconnect structure of a three-dimensional integrated risc processor/cache system”, *IEEE Transactions on Components, Packaging and Manufacturing Technology*, Part B, vol 19, 4, November 1996., pp 719-718

[14] M.B. Kleiner, S.A. Kühn, P.Ramm and W. Weber, “Thermal analysis of vertically integrated circuits”, *Proceedings IEDM*, 1995, pp 487-490

[15] S.A. Kühn, M.B. Kleiner, P.Ramm and W. Weber, “Interconnect capacitances, crosstalk and signal delay in vertically integrated circuits”, *Proceedings IEDM*, 1995, pp 249-252

[16] S. Strickland, E. Ergin, D.R. Kaeli and P. Zavracky, “VLSI in the third dimension” *Integration, the VLSI journal*, 25, 1, September 1998, pp 1-16

[17] “Status2000: integrated circuit industry report”, Integrated Circuit Engineering Corporation, Scotsdale, AZ, USA, 2000