Chapter 2
The Promises and Limitations of 3-D Integration

Axel Jantsch, Matthew Grange and Dinesh Pamunuwa

2.1 Introduction

Due to their compact geometry, 3-D integrated systems hold promises to significantly reduce latency, power consumption and area, while increasing bandwidth. In the following we quantify the potential and limits of 3-D integration by analyzing the theoretical performance of various 2-D and 3-D topologies.

We adopt T. Claasen's notion of *intrinsic computational efficiency* of silicon [1]. The intrinsic computational efficiency is obtained when all the silicon area is filled with elementary operations, say 32-bit adders, and no area is "wasted" for data communication and control.

Figure 2.1 shows how the intrinsic computational efficiency, measured in millions of operations per second per Watt, is increasing with technological progress. The left part of the curve is copied from T. Claasen's original paper [1], while the right part is based on our own model, as introduced below. For comparison we have marked the performance of two recent multi-core processors from Tilera Inc. [2].

Different architectures such as micro-processors, DSPs, FPGAs and custom hardware, will approximate this line to a higher or lower degree depending on how well an application matches the architecture and how much flexibility is built-in. But no real processing unit can match or exceed it. Larger and more general purpose processors exhibit a greater gap because they utilize more area and power on interconnect, control and provision of programmability.

Communication performance, area and power consumption directly benefit from 3-D integration due to geometric properties. Figure 2.2 shows how the geometric distance between cores grows very differently in 2-D and 3-D structures with the number of cores. Since for global and long distance communication the geomet-

A. Jantsch (✉)
Department of Electronic Systems, School of Information and Communication Technology,
Royal Institute of Technology, 120 Forum, 16440 Kista,
Sweden
Tel.: +46 8 790 4124
e-mail: axel@kth.se

A. Sheibanyrad et al. (eds.), *3D Integration for NoC-based SoC Architectures,*
Integrated Circuits and Systems, DOI 10.1007/978-1-4419-7618-5_2,

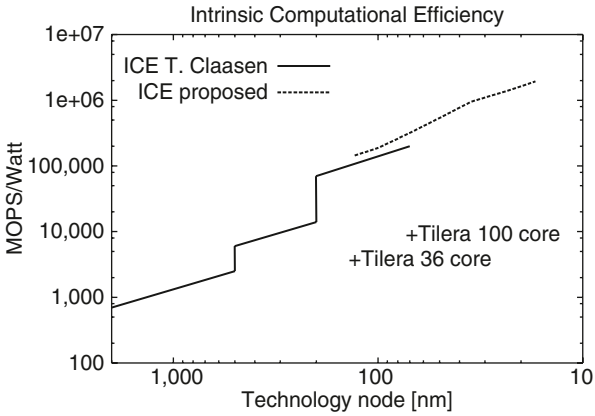

Fig. 2.1 Computational efficiency vs. minimum feature size is shown here. The discrepancy in the overlapping section between T. Classon's work and this study is due to the significant variation in the energy consumption of different adder architectures. The performance of the two processors shown fall outside of this range due to the overhead of the control circuitry and interconnect which is not accounted for by this metric

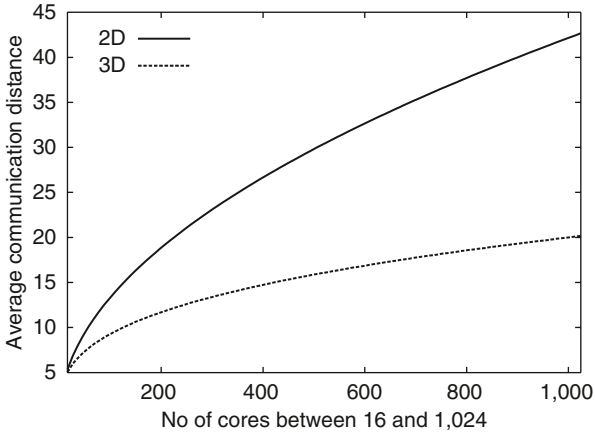

Fig. 2.2 The average geometric distance for a multicore system for a 2-D and a 3-D realization

ric distance translates linearly to latency, we can expect to cut communication latency by 50%. A number of recent studies of communication performance in 3-D structures [3–7] demonstrate the significant potential of 3-D integration technology for reducing power consumption and increasing performance.

3-D integration enables stacking of memory on top of processors, thus realizing a direct low latency and high bandwidth memory access link. However, to exploit the benefits, the memory architecture has to be adapted to allow for multi-port, parallel memory access. Several recent studies have explored various memory and cache architectures while exploiting the third dimension. For instance Li et al. [8] propose a 3-D distributed L2 cache and observe a 50% access latency reduction, essentially due to shorter wires within the L2 cache. Loh [9] explores the effect of parallel memory access by means of multiple memory controllers and ranks in a 3-D stacked DRAM based memory architecture and reports a performance increase of more than 280% over a conventional memory architecture for a set of benchmark applications. In our model we assume that in a 3-D topology, DRAM is used as embedded memory because it can be placed on a separate die, thus leveraging on the capability of 3-D to integrate different process technology in the same system.

In Sect. 2.2, we introduce and motivate our analytical model for performance comparison. Then, we describe the technological parameters for performance, power consumption and area for the 2-D and 3-D topologies (Sect. 2.3). Also, we explain our scaling methodology. Section 2.4.1 delves into the impact of how memory is distributed in the system. Not surprisingly a distributed memory exhibits higher performance than a centralized memory. However, after distributing 80% of the memory, further diffusion of the storage has little added benefit. In Sect. 2.4.3 we become more concrete by assuming specific system sizes and frequencies. This allows for analyzing performance limits under power, frequency and area constraints. Finally, we provide further discussions of the model and our conclusions in Sect. 2.5.

2.2 Computational Efficiency of Silicon

2.2.1 Computation

Intrinsic Computational Efficiency (*ICE*) of silicon [1] is the number of 32-bit add operations per Joule, or the number of operations per second per Watt. As a concrete example, let's assume one 32-bit adder covers an area of 2956 μm^2 and each 32-bit addition consumes 6.9 pJ in 130 nm technology; it covers an area of 437.3 μm^2 and consumes 1.85 pJ in 50 nm technology. Thus, we get as intrinsic computational efficiency the following.

$$ICE^{130} = 1/(6.9\text{ pJ}) = 144\text{ GOPS/W}$$

$$ICE^{50} = 1/(1.83\text{ pJ}) = 540\text{ GOPS/W}$$

The *ICE* reflects the amount of computation that can be done within an energy envelope, but it does not measure the amount of computations per area or per volume. We now define the *Intrinsic Computational Density* (*ICD*) as the number of 32-bit adders that fit into 1 mm^2. For instance, with the example above we get

$$ICD^{130} = 1/(2{,}956\ \mu m^2) = 338.3\ \frac{\text{operators}}{mm^2}$$

$$ICD^{50} = 1/(437.3\ \mu m^2) = 2{,}286.8\ \frac{\text{operators}}{mm^2}$$

Figure 2.1 shows the intrinsic computational efficiency as a function of technology nodes. Theo Claasen's plot from 1999 is repeated and for comparison the *ICE* figures of our model (Sect. 2.3) for technology nodes between 180 and 17 nm are added.

2.2.2 Adding Memory and Communication

ICE and *ICD* give the upper bound of what amount of computation can be done with a given silicon technology under the assumption that the entire area is densely packed with computation units. However, we also need to account for memory, where the data are stored before and after processing, and for interconnect, which allows the data to move between processing units and memory. In the following we study variants of *ICE* and *ICD* under less ideal assumptions in different 2-D and 3-D configurations. In particular we explore the following factors and configurations.

- **Ratio of computation to memory μ:** We distinguish between the temporal ratio μ_T and the spatial ratio μ_S. The relative number of memory accesses for each operation is μ_T, while μ_S is the number of memory words in the system for each operator. With memory we mean SRAM, caches and the like but also off-chip DRAM. If $\mu_T = 1$, for each operation there is 1 memory access. Typical values will be between 1 and 3. On the other hand, the amount of memory is usually much higher than the operators. Hence, typical values for μ_s are between 1,000 and 10,000 as we discuss later.
- **Ratio of on-chip versus off-chip memory:** ω: If $\omega = 1$, all memory is on-chip; if $\omega = 0$, all memory is off-chip. In a 3-D topology, with on-chip we mean all dies in the 3-D stack.
- **Memory distribution factor Δ:**
  - $\Delta = 0$: completely distributed memory where the distance between a computation unit and the memory is always 0;
  - $\Delta = 1$: completely central memory where the distance between a computation unit and the memory is always the diameter of the system (or off-chip)
  - for example $\Delta = 0.05$: models a cache system where 95% of all memory accesses are local and 5% are far away.

  The idea is to account for the communication required to write to and read from memory. If the memory is completely distributed, we assume all memory reads and writes are local and no long-range communication is required. Obviously, this is a simplification but any specific architecture-application pair can be characterized by a Δ value between 0 and 1, denoting the amount of global on-chip communication occurring.
- We explore different **2-D and 3-D topologies** but we typically compare systems with the same total silicon area. For example if the total area is 400 mm^2, the configurations considered are
  - 2D: one plain silicon die of size 20×20 mm^2;
  - 3D2: two dies stacked upon each other, each 200 mm^2;
  - 3D4: four dies stacked upon each other, each 100 mm^2;
  - 3D8: eight dies stacked upon each other, each 50 mm^2;
  - 3D16: sixteen dies stacked upon each other, each 25 mm^2.

2.2.3 Effective Computational Efficiency

To include the effects of memory and interconnect, we define the *Effective Energy* (*EE*) for a 32-bit addition as follows.

$$EE^{tn}_{arch} = E^{tn}_{32} + \mu_T(\omega(e_1 + \Delta \times E\_int^{tn}_{arch}) + (1-\omega)(e_1 + E\_int^{tn}_{arch} + E_{offchip}))$$

for a given technology node, *tn*, and a given architecture, *arch*, $\in$ {2D, 3D2, 3D4, 3D8, 3D16}. The three main terms correspond to the energy consumption of an addition, of on-chip memory access and of off-chip memory access, respectively.

- e_1 is the amount of energy it takes to read or write one 32-bit word in on-chip SRAM.
- $E\_int^{tn}_{arch}$ is the energy it takes to transport one 32-bit word from a non-adjacent on-chip memory to the local cache (see Table 2.1). For example, if the total silicon area is 400 mm^2, we have

$$E\_int^{tn}_{arch} = \begin{cases} (10+10)e_2(tn) & \text{if arch} = 2D \\ (7.07+7.07)e_2(tn) + e_3(tn) & \text{if arch} = 3D2 \\ (5+5)e_2(tn) + 2e_3(tn) & \text{if arch} = 3D4 \\ (3.5+3.5)e_2(tn) + 4e_3(tn) & \text{if arch} = 3D8 \\ (2.5+2.5)e_2(tn) + 8e_3(tn) & \text{if arch} = 3D16 \end{cases}$$

- e_2 is the energy it takes for a 32-bit word to be transported 1 mm horizontally in a given technology.
- e_3 is the energy it takes to move a 32-bit word from one vertical level to the next via a set of TSVs.
- $E_{offchip}$ is the energy it takes to get off-chip and to read or write the off-chip memory. It includes the I/O drivers, the inter-chip communication and the energy consumption of the memory chip.

The idea of *E_int* is to capture the communication energy in different architectures to get from an arbitrary point in the system to a particular point at the system boundary. For a 2-D 20 × 20 mm^2 die, the distance is on average 10 mm in each dimension, hence it is 20 mm. For a 3-D structure we have to traverse half of the vertical levels on average. For example for a 3*D*4 we have to traverse two vertical levels.

Thus, the effective energy *EE* gives the required energy for a 32-bit addition if memory access and communication is taken into account. The factors μ_T, ω and Δ are abstractions of architectural choices and features. Based on *EE* we define the *Effective Computational Efficiency* (*ECE*) as follows.

$$ECE^{tn}_{arch} = \frac{1}{EE^{tn}_{arch}}$$

which gives the amount of computation we can do within the energy envelope of 1 J; or the amount of computations per second we can do within the power envelope of 1 W.

Table 2.1 Notation and metrics of comparison

Abstraction of architectural design parameters	
tn	Minimum feature size of a technology node (nm)
$arch$	Architecture of system (2D, 3D2, 3D4, 3D8, 3D16)
ω	Ratio of on- to off-chip memory ($\omega = 1$: all memory is on-chip, $\omega = 0$: all off-chip)
Δ	Memory distribution factor ($\Delta = 1$: all centralized memory, $\Delta = 0$: all local)
μ_T	Number of memory accesses per h/w operation ($\mu_T = 1$: one mem. access per op)
μ_s	Amount of memory per h/w operator (typically $\mu_s = 1{,}000–10{,}000$)
σ	Interconnect sharing factor ($\sigma = 1$: no sharing, $\sigma = 0$: completely shared)
n	Number of die layers for 3-D architectures
$area$	Area of die
Technology and architecture dependent parameters	
E_{32}	Energy for a 32-bit add operation
e_1	Energy for a 32-bit read/write to local SRAM
e_2	Energy to transport a 32-bit word over 1 mm on a planar on-chip bus
e_3	Energy to transport a 32-bit word over one vertical layer across TSVs
a_1	Area for a 32-bit memory word in SRAM or DRAM
a_2	Area for a 1 mm long 32-bit planar on-chip bus
a_3	Area for 32 TSVs
$E_{offchip}$	Energy to read/write to off-chip memory. Includes I/O drivers, inter-chip communication and memory chip energy consumption
Primary comparison metrics	
ICE	Number of 32-bit add operations per Joule
ICD	Number of 32-bit adders per mm^2
$A\_int^{tn}_{arch}$	Interconnect area required to transport a 32-bit word from a non-adjacent on-chip memory to the local cache: $\sqrt{\frac{area}{n}}a_2 + \frac{n}{2}a_3$
$E\_int^{tn}_{arch}$	Interconnect energy required to transport a 32-bit word from a non-adjacent on-chip memory to the local cache: $\sqrt{\frac{area}{n}}c_2 + \frac{n}{2}c_3$
EE^{tn}_{arch}	Effective Energy for a 32-bit addition: $EE^{tn}_{arch} = E^{tn}_{32} + \mu_T(\omega(e_1 + \Delta E\_int^{tn}_{arch}) + (1-\omega)(e_1 + \mathrm{E}\_int^{tn}_{arch} + E_{offchip}))$
ECE^{tn}_{arch}	Amount of computation achieved with 1 J: $\frac{1}{EE^{tn}_{arch}}$
EA^{tn}_{arch}	Effective area for a 32-bit addition without off-chip memory: $EA^{tn}_{arch} = A^{\mathrm{tn}}_{32} + \mu_S\ \omega\ a_1 + \sigma A\_int^{tn}_{arch}$

For example the special case of $\mu_T = 0$ (no memory read or write) and 2-D in a 130 nm technology we get

$$ECE^{130}_{2D} = \frac{1}{EE^{tn}_{arch}} = \frac{1}{E^{130}_{32}} = ICE^{130} = 144.3 \text{ GOPS/W}$$

2.2.4 Effective Computational Density

Similarly, the area cannot be filled with computational units only. We need to take memory and interconnect into account as well. We define the *Effective Area* (*EA*) as follows.

$$EA^{tn}_{arch} = A^{tn}_{32} + \mu_S \; \omega \; a_1 + \sigma \; A\_int^{tn}_{arch}$$

EA is defined similarly to *EE* (see Table 2.1) but the off-chip component is omitted since we do not include the area for off-chip memory. Again, with "off-chip" we really mean "out-of-package". Different dies in a 3-D stack are considered "on-chip".

- a_1 is the area for a 32-bit memory word. Depending on the geometry we assume either SRAM or DRAM memory. For a 2-D system we use the area of embedded SRAM, while for a 3-D system we use DRAM. Concretely we use $60F^2$ area for one SRAM cell [10] and between $8F^2$ and $4F^2$ for DRAM cells [11], where F is the minimum feature size.
- $A\_int^{tn}_{arch}$ is the interconnect area required for transporting a 32-bit word to memory. For example, in a 400 mm^2 system we have

$$A\_int^{tn}_{arch} = \begin{cases} (10+10)a_2(tn) & \text{if arch} = \text{2D} \\ (7.07+7.07)a_2(tn) + a_3(s) & \text{if arch} = \text{3D2} \\ (5+5)a_2(tn) + 2a_3(s) & \text{if arch} = \text{3D4} \\ (3.5+3.5)a_2(tn) + 4a_3(s) & \text{if arch} = \text{3D8} \\ (2.5+2.5)a_2(tn) + 8a_3(s) & \text{if arch} = \text{3D16} \end{cases}$$

- σ is the interconnect sharing factor. If $\sigma = 1$, no sharing takes place and every operator has its own, private interconnect across the system. If $\sigma = 0$, the interconnect is optimally shared and the interconnect area per operator is 0. In analogy to μ_S it gives the ratio of area occupied by operators versus interconnect. Typical values are between 0.01 and 0.1. For instance the Tilera TILE64 [12] with 64 cores has eight 32-bit operators per core[1]. For 512 operators on the 64 core Tilera die with an 8×8 mesh interconnect, the sharing factor is $\sigma = 16/512 = 0.031$.
  Note, that only the global interconnect for the dataflow is counted, while the local interconnect and global control lines are ignored.
- a_2 is the required area for a 1 mm long 32-bit bus.
- a_3 is the required area for 32 TSVs to connect from one horizontal layer to the next.

For instance, if one 32-bit adder has the size 437.5 μm^2 in 50 nm technology and $\mu_s = \sigma = 0$, we get an $ECD^{50} = ICD^{50} = 2{,}286.8$ operations/mm^2 and we can fill a 400 mm^2 chip with 914,720 adders. In a more realistic scenario with $\mu_s = 4{,}000$ and $\sigma = 0.031$ we get 345 operators on a 400 mm^2 area.

[1] Again, this is a simplification, because there are fewer operators but they are pipelined. In effect, 8 operations can be completed per cycle in the best case, motivating the number 8 that we use in this example.

2.3 Technology Parameter Scaling

To capture the performance benefits for feature size scaling of each successive generation of technology, physical properties of various on-chip communication transactions and logical operations were modeled for each node. The technology parameters are broken down into several categories; global planar 2-D wires, logical operations, memory transactions and 3-D TSV signaling.

2.3.1 Planar 2-D Wire Models

The minimum feature size on a die scales by roughly 0.7 each generation, however global on-chip wires do not scale as aggressively as intermediate or local wires. In Ho et al.'s "The Future of Wires" [13] conservative and aggressive wire scaling trends for decreasing feature sizes are discussed. Capacitance, including parallel plate, fringe and coupling terms, and resistance are extracted from field solver simulations and compact models fitted to extract parameters for future technology nodes given the predicted global wire dimension, barrier thickness, spacing, resistivity of the medium, vertical and horizontal dielectric constants and the switching probability of the surrounding wires. Using the *RC* characteristics of the wires, typical repeater insertion strategies, and scaling supply voltages, we determine the energy per bit for a requisite wire length across technology nodes from 180 nm down to 17 nm. We assume bus widths of 32 and the percentage of bits switching per transaction to be one half the total number of bits.[2] Driver, receiver and repeater energies are also considered for each wire in the bus. The energy consumed in transmitting a 32-bit word per network link is calculated from the following equation:

$$E_{link} = k\left[\frac{1}{2}V_{dd}^2\left(C_{w_{self}} + 2C_c + hC_{rep}\right)\right] \times Bus\_width \times Switch\_factor$$

where k is the number of repeaters and h the size of each, $C_{w_{self}}$ is the self capacitance of a global wire, C_c is the coupling capacitance to neighboring wires, C_{rep} is the total input gate and output drain capacitance of the repeater and V_{dd} is the supply voltage for a given technology.

2.3.2 3-D TSV Interconnect Models

We have conducted field solver simulations of cylindrical, copper-filled through silicon vias to extract the relevant *RLC* parasitics. TSV electrical characteris-

[2] This figure depends less on architectural choices, but more on how information is coded, protected and compressed. Although a simplification, it is important to note that the same assumption is used for all architectures and the relative comparisons and main trends are not sensitive to the chosen value for switching activity.

tics depend on their geometrical parameters as well as material properties such as the dielectric properties of the barrier and insulating layers and the dopant concentration in the substrate. For the purpose of extracting parasitics and subsequent analysis, a representative structure for a TSV is assumed to be a copper-filled via with uniform circular cross-section and an annular dielectric barrier of SiO_2 or Si_3N_4 surrounding the Cu cylinder with a thickness of 0.2 μm [14]. The dimensions vary depending on the technology node; the cross-section is assumed to be uniformly circular, with radii of 10, 8, 6, 4, 2 and 1 μm and a constant length of 50 μm. The pitch of the TSVs is twice the radius to match planar global wire spacing trends. In this work we have considered a substrate conductivity of 10 S/m representing typical values used in digital processes. The topology considered is three parallel coupled TSVs, a representative unit in any size row of TSVs.

We use the extracted parasitics with the same methodology as the planar wires, where the bus width and switch factor match the 2-D parameters. The energy is calculated for a single transaction from one layer to the next adjacent layer, so the hop length is fixed. Driver and receiver energy is also considered, however no repeaters are required for the 3-D interconnect. The TSVs are arranged in a row, thus the total area of the interconnect is a straightforward relationship to the pitch and radius of the TSVs.

2.3.3 Logical Operation and DRAM Scaling

We extract the energy per operation of several logic operations such as a 32-bit addition or SRAM read, by using published data [15, 16] for a particular technology node and scaling the energy and area for future or past generations. Dally in [16] publishes the energy per add operation of a 32-bit adder in 130 nm 1.2 V technology as 5 pJ. A reasonable approximation, ignoring leakage, for the energy in other technology nodes can be obtained by scaling according to the following:

$$Energy_{new} = Energy_{130\ \mathrm{nm}} \times \left(\frac{Feature_{size_{new}} \times Vdd_{new}^2}{Feature_{size_{130\ \mathrm{nm}}} \times Vdd_{130\ \mathrm{nm}}^2} \right).$$

The area can be scaled in a similar manner by a straightforward relation of the minimum feature sizes between nodes.

The off-chip DRAM transaction energy is not a simple function of the feature size, and depends on the DRAM architecture, its peripheral circuitry and also characteristics of off-chip drivers and terminations, and chip, package and board trace parasitics. We have used the Micron System Power Calculator [11] to estimate the average read/write power for different generations of DRAM, from SDRAM to DDR3. We then divide this power by the number of bits and the simulated transaction data rate to determine the energy per bit per generation of off-chip DRAM. We have matched the DRAM generation to the technology node, such that 180 nm corresponds to SDRAM and 17 nm to DDR3. There are a number of complexities

associated with off-chip transactions, such as bus controller architecture, termination power, transaction delay, and the number of peripheral I/O devices, which cause the energy to vary over a wide range depending on these choices. In our study we have been consistent with the values we use in order to minimize the impact on comparisons between different schemes.

2.4 ECE Trends and Dependencies

Next, we study the dependency of *ECE* on parameters like Δ and ω and then we investigate the limits of *ECE* and raw performance under power, area and frequency constraints.

To see the overall trend, Fig. 2.3 illustrates how the *ECE*, the performance for a given power envelope, will develop as technology scales. As a reference the plot shows the *ECE* of two recent multi-core Tilera processors. 3-D topologies have a 3 times higher *ECE*, mainly due to lower communication power consumption in a more compact geometry. Moreover, this increased efficiency of 3-D is gained at a much smaller area and lower frequency for the same performance, as will be illustrated below in Sect. 2.4.3.

2.4.1 Distributed versus Central Memory

To study the effect of the memory distribution factor on the *ECE* we assume that for every operation on average one word has to be read or written from or to memory (or cache)[3]. Hence, $\mu_T = 1.0$.

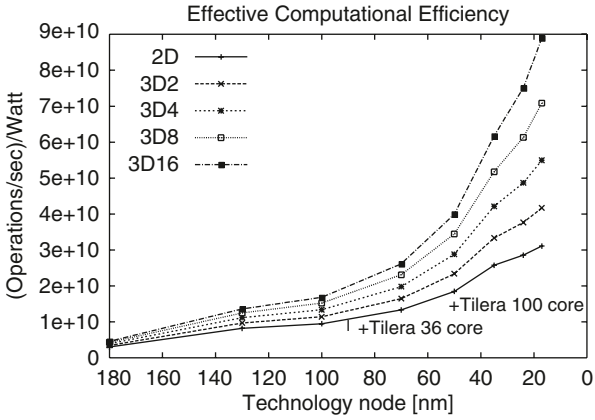

Fig. 2.3 Performance of different topologies at different technology nodes with $\Delta = 0.05$, $\omega = 1$ and $\mu_T = 1$. The data for the two Tilera processors are closer to the theoretical performance than in Fig. 2.1 due to the fact that the interconnect overhead has been accounted for, although the control circuitry overhead is neglected

[3] We assume registers and small register files close to the operators. Reading and writing of registers is not considered as memory access.

Figure 2.4 shows the *ECE* for various topologies when Δ, representing the proportion of centralized memory, varies between 0 and 1. For all topologies a centralized memory drags down *ECE* significantly from over 60 GOPS/W to about 5–10 GOPS/W. Hence, there is a benefit from distributing memory, but only a distribution of $\Delta < 0.2$ gives a significant effect. This benefit from distribution is more pronounced for a 2-D topology. Going from $\Delta = 1$ to $\Delta = 0.1$ improves *ECE* for 3D16 by a factor 5, while the improvement is 8 for 2D. Intuitively the reason for this is that the cost of transporting data across the chip to a central memory is much lower for a 3-D topology. Hence, if it is difficult to decentralize most memory accesses, the penalty will be lower for 3-D.

However, the cost of centralized memory becomes steeper for more advanced technologies. Figure 2.5 shows this effect for a 3D16 topology. While the difference between $\Delta = 0$ and $\Delta = 1$ is a factor 5.4 for 180 nm technology, it grows to a factor of 34 for a 17 nm technology.

Hence, even if a 3-D topology can mitigate the cost of centralized memory, it is still growing exceedingly as technology advances due to the inverse effect on the performance of logic versus interconnect as a result of scaling.

2.4.2 On-chip versus Off-chip Memory

While Figs. 2.4 and 2.5 assume all memory to be on-chip ($\omega = 1$), Fig. 2.6 shows the cost of having part of the memory off-chip. At $\omega = 1$ all memory is on-chip.

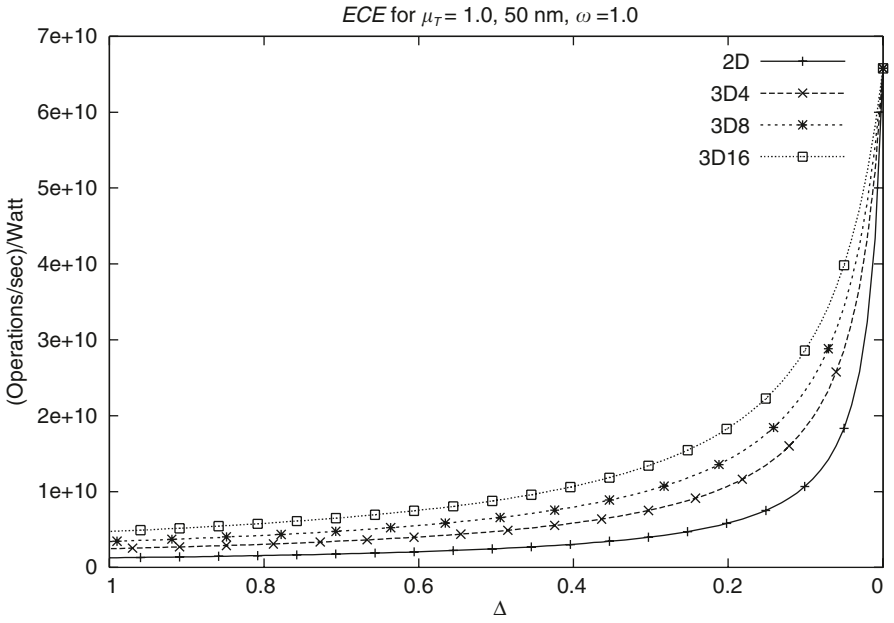

Fig. 2.4 The effect of the memory distribution factor Δ on *ECE* for different topologies

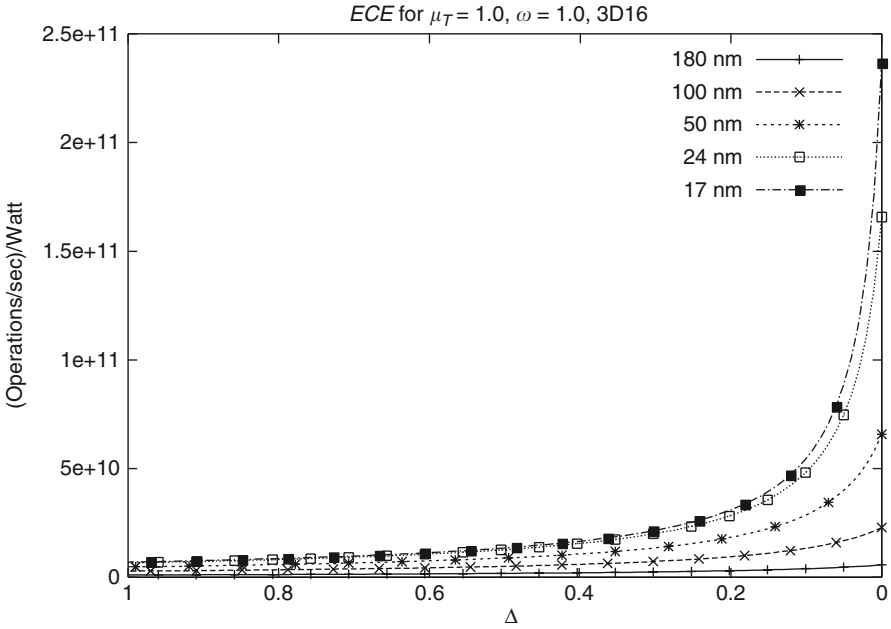

Fig. 2.5 The effect of the memory distribution factor Δ on *ECE* for 3D16 topology and various technology nodes with $\mu_T = 1.5$ and $\omega = 1$

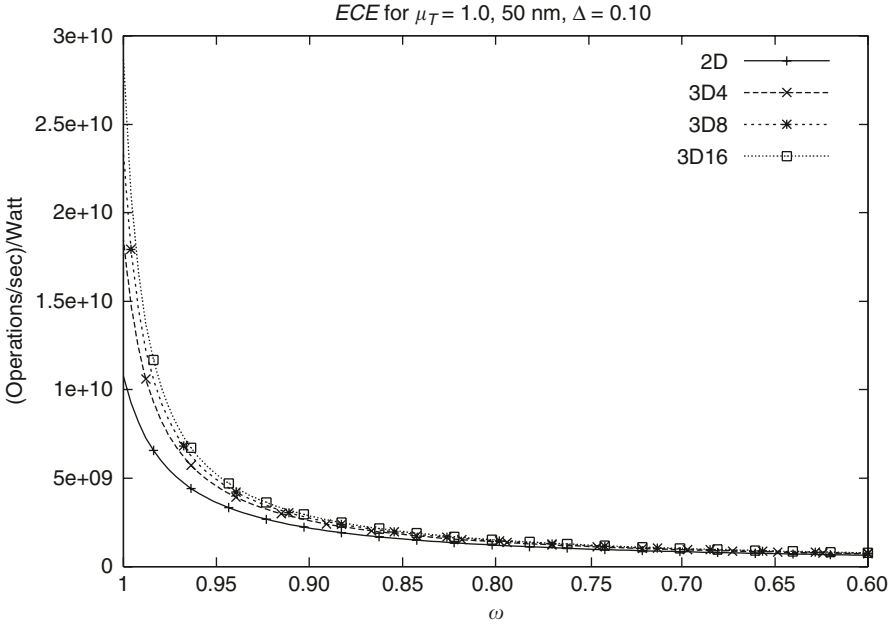

Fig. 2.6 The effect of the on-chip versus off-chip memory with a memory distribution factor Δ=0.1 on *ECE* for various topologies

As the fraction of on-chip memory accesses decreases, the *ECE* drops. When 20% of the memory accesses is off-chip ($\omega = 0.8$), the *ECE* drops by a factor 9 for 2-D and by a factor 19 for 3-D16 systems. Intuitively, *ECE* drops more for 3-D topologies because its *ECE* figures for all on-chip memory are more favorable, but the *ECE* figures for all off-chip memory are almost the same for all considered topologies.

2.4.3 Size Constrained System

ECE is a metric that does not consider at what frequency or within what space a set of computations is performed. It is an abstract metric for a technology rather than for a concrete system.

In order to better understand the limits of performance under given power, area and frequency constraints, we consider systems of a concrete size.

Figure 2.7 shows the effect of varying the ratio of memory and operators in a concrete system with 400 mm^2 area. To start with the area occupied by operators is relatively small (Fig. 2.7a). Most of the area is covered by memory for realistic memory-operator ratios of $1{,}000 \leq \mu_S \leq 5{,}000$. For instance TILE64 [12] has a maximum performance of 8 operations per cycle per core yielding 512 operations per

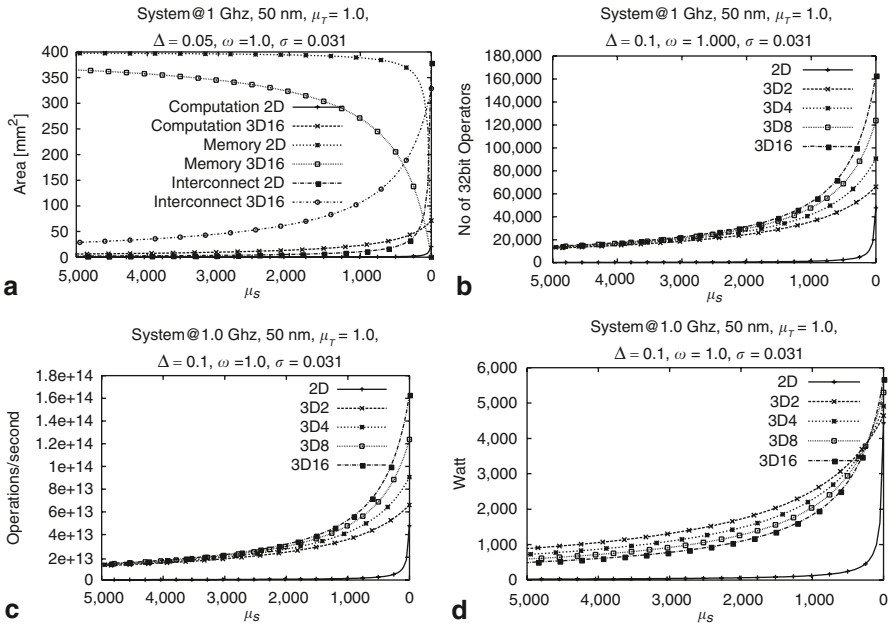

Fig. 2.7 Area, performance and power consumption for a concrete 400 mm^2 system with $\mu_T = 1.0$, $\sigma = 0.031$ and $\omega = 1.0$, clocked at 1 GHz, with a 50 nm technology. **a** Area distribution. **b** Number of operations. **c** Operations per second. **d** Power consumption is limiting how densely computational units can be packed

cycle. It has 4 MB (=1 M Word) of on-chip cache resulting in $\mu_s = 2^{20}/512 = 2{,}048$. The Niagara 2 [17] processor from Sun Microsystems, which is an 8 core 64 thread processor with 4 MB of on-chip cache, falls into a similar range.

Keep in mind that our model illustrates trends and limits but does not account for control logic, decoders, arbiters, etc. The area contribution of that part is not seen in Fig. 2.7a. We usually attribute those elements to the processing units and hence, their area fraction is lower in Fig. 2.7a than we intuitively expect. However, the comparison of 2-D and 3-D topologies is interesting. Due to the higher density of memory in a 3-D architecture (DRAM vs SRAM in 2-D), the area dominance of memory in 2-D is much higher than in 3-D for the same μ_s. Consequently, more of the area in a 3-D system is filled with computation units and interconnect. (The relative ratio of the two latter is given by σ. A lower σ would reserve more of the area to computation.)

Figure 2.7b shows how the area not covered by memory, is used for computation in 2-D and 3-D topologies. For $\mu_s = 2{,}000$ we can afford 684 operators in a 2-D system, while we can squeeze in 24,551 operators in a 3D16 system. The reason 35 times more operators fit into the same area is mainly due to the much higher density of DRAM as opposed to SRAM that is common in 2-D based systems. This naturally translates to a similar increase of performance as Fig. 2.7c illustrates. It also results in a prohibitively high power consumption since the computation consumes much more power than the memory. Apparently, we cannot power all these computations in reality, but we can translate the increased potential that 3-D offers into either smaller chips, or lower frequency, or higher memory content.

Figure 2.8 shows performance and power consumption for a smaller system (100 mm^2) clocked at a somewhat lower frequency and at the 35 nm technology node. With $\mu_s > 4{,}000$ we get a practical power consumption and still a very respectable tera-scale performance.

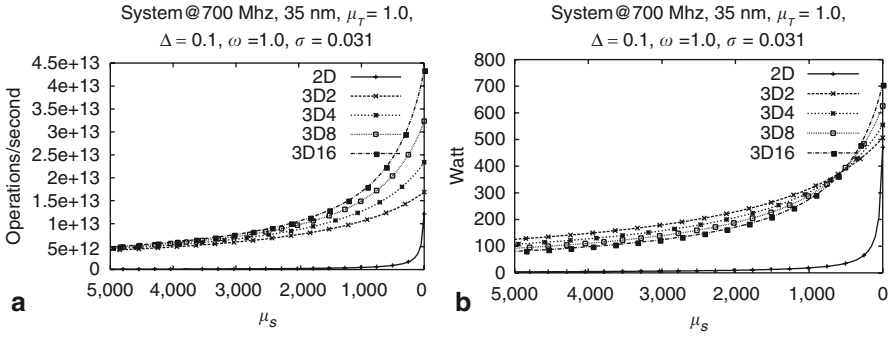

Fig. 2.8 Performance and power consumption for a concrete 100 mm^2 system with $\mu_T = 1.0$, $\sigma = 0.031$, $\omega = 1.0$ and $\Delta = 0.1$, clocked at 700 MHz, with a 35 nm technology. **a** Operations per second. **b** Power consumption

2.4.4 Power and Frequency Constrained Systems

For a given power budget, the performance is significantly higher for 3-D architectures as shown in Fig. 2.9. However, these performance and power figures are obtainable at small sizes and low frequencies for 3-D topologies. Figure 2.9a gives the maximum performance under a given power budget. A 3D16 topology offers 2.4 times higher performance per Watt than a 2-D topology. Interestingly, for every doubling of the stack height, we see a 20–30% increase of the performance per Watt figure. This trend is only slowly decreasing from 30% (2D–3D2) down to 20% (3D8–3D16) in Fig. 2.9a.

The somewhat higher *ECE* of 3-D topologies are obtainable at significantly lower frequency and smaller area. Figure 2.9b shows the area-frequency trade-off for a given power (100 W) and performance (6,170.5 GOPS). For any given area, the frequency required for a 2D topology is about 25 times the frequency of the 3D16 system. Since frequencies above a few GHz are hard and costly to realize, a 2-D chip faces a tough performance hurdle while 3-D topologies can approach their *ECE* limits at much lower frequencies.

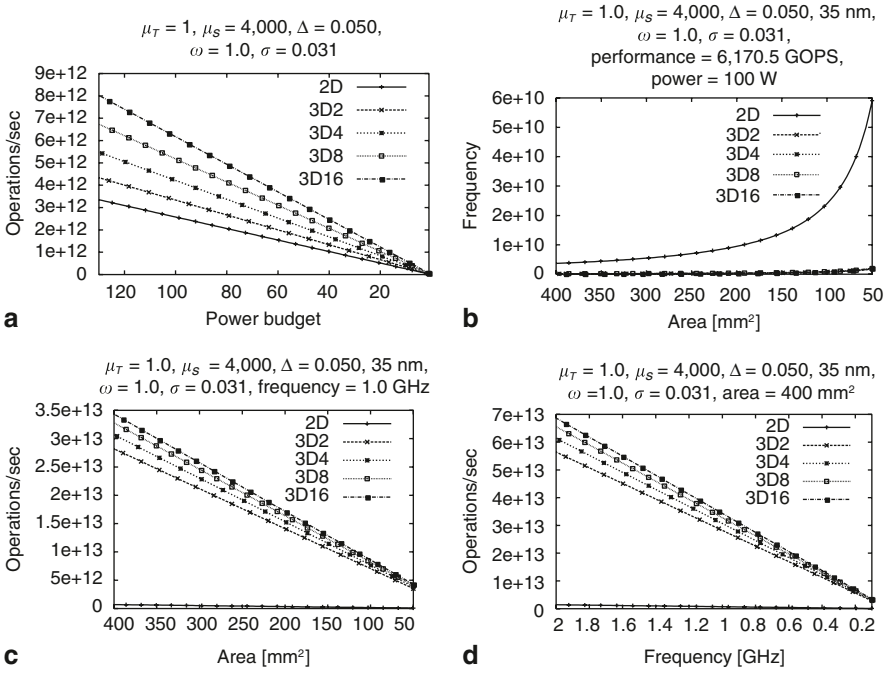

Fig. 2.9 Dependences of area, frequency, performance and power consumption of different topologies in a 35 nm technology. **a** Performance under power constraint of a 400 mm^2 system. **b** Under a given power budget of 100 W the performance is 6,170 GOPS. **c** Given a frequency of 1 GHz, the performance is a function of the area. **d** Given an area of 400 mm^2 the performance is a function of frequency

In Fig. 2.9c the frequency is set to 1 GHz and the performance increases with area size. For 3-D systems the performance increase is significantly steeper than for 2-D, because most added area in a 2-D chip is spent on memory and interconnect, and relatively little is invested in additional operators. Figure 2.9d illustrates the same trend; it fixes the area to 400 mm^2 and shows how the performance grows with the frequency.

Figure 2.9 demonstrates clearly the tremendous potential of 3-D stacked systems, which mainly stems from (1) the possibility to integrate dense DRAM tightly into the multi-core architecture, and (2) from the more power efficient interconnection in the third dimension, which essentially is due to shorter geometric distances.

2.5 Conclusion

Inspired by the intrinsic computational efficiency of silicon proposed by T. Claasen we have developed the concepts effective computational efficiency (*ECE*) and effective computational density (*ECD*). They consider memory and interconnect in addition to computational operators. A small number of parameters allow an abstract characterization of a broad range of architectures and topologies. We have used *ECE* and *ECD* to study the limits of performance of 2-D and 3-D topologies with technology down to 17 nm.

Our model is an abstraction of real systems and ignores many relevant aspects and details. Thus, it can only give upper bounds on the performance and real systems will not be able to exhibit comparable performance numbers. In particular we have not considered control logic, local interconnect, registers and register files. These components can consume a significant portion of the area and power of a real system.

Another limitation is our focus on throughput as the main performance characteristic, while ignoring latency. Latency is much harder to capture at an abstract level since it is influenced strongly by the many details of the architecture, the arbitration policies and resource management. In real systems the theoretical limits of throughput are often not achieved because raw capacity is over-provided and a lot of control logic is spent to keep critical latency figures low. It can be noted however, that a main benefit of 3-D topologies is the lower latency of memory transactions since high capacity memory can be located much closer to the computation units. This may mean that 3-D systems come closer to their intrinsic performance limits than 2-D topologies.

In summary, although our model constitutes an idealization of systems, it still expresses correct trends and bounds of real systems and we draw the following main conclusions from our study:

- 3-D systems have 2–3 times higher *ECE* due to lower interconnect power;
- 3-D systems have one order of magnitude higher memory density due to DRAM integration;

- Consequently, 3-D systems can accommodate many more computation units in a given area and with the same amount of memory;
- This allows for much higher performance but causes also very high power density.
- The same performance with the same power can be realized in 3-D topologies with much smaller area and at lower frequency.

The last point means a cost advantage for 3-D systems, which may compensate the more expensive 3-D integration technology.

References

1. T. Claasen. High speed: Not the only way to exploit the intrinsic computational power of silicon. *Proceedings of the International Solid State Circuits Conference* (*ISSCC*), 1999.
2. Tilera Corporation. Tilera Home Page. http://www.tilera.com.
3. W.J. Dally. Performance analysis of k-ary n-cube interconnection networks. *IEEE Transactions on Computers*, 39(6):775–785, 1990.
4. A.Y. Weldezion, M. Grange, D. Pamunuwa, Z. Lu, A. Jantsch, R. Weerasekera and H. Tenhunen. Scalability of network-on-chip communication architecture for 3-D meshes. *Proceedings of the International Symposium on Networks-on-Chip*, 2009.
5. V.F. Pavlidis and E.G. Friedman. 3-D topologies for networks-on-chip. *IEEE Transactions on Very Large Scale Integration Systems*, 15(10):1081, 2007.
6. R. Weerasekera, D. Pamunuwa, L.-R. Zheng and H. Tenhunen. Two-dimensional and three-dimensional integration of heterogeneous electronic systems under cost, performance and technological constraints. *IEEE Transactions on Computer-Aided Design*, 28(8):1237–1250, 2009.
7. B. Feero and P.P. Pande. Networks on chip in a three dimensional environment: A performance evaluation. *IEEE Transactions on Computers*, 58(1), 2009. http://www.micron.com/support/dram/power_calc.html
8. F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan and M. Kandemir. Design and management of 3 D chip multiprocessors using network-in-memory. *ACM SIGARCH Computer Architecture News*, 34(2):130–141, 2006.
9. G. Loh. 3D-stacked memory architectures for multi-core processors. *Proceedings for the 35th ACM/IEEE International Symposium on Computer Architecture (ISCA)*, 2008.
10. S. Lai and T. Lowrey. OUM—A 180 nm nonvolatile memory cell element technology for stand alone and embedded applications. *Proceedings of the International Electronics Device Meeting*, 2001.
11. J. Janzen. The micron system-power calculator. Micron web site, 2009. http://www.micron.com/support/dram/power_calc.html
12. S. Bell, B. Edwards, J. Amann, R. Conlin, K. Joyce, V. Leung, J. MacKay, M. Reif, L. Bao, J. Brown, M. Mattina, C.-C. Miao, C. Ramey, D. Wentzlaff, W. Anderson, E. Berger, N. Fairbanks, D. Khan, F. Montenegro, J. Stickney and J. Zook. TILE64TM Processor: A 64-Core SoC with mesh interconnect. *Proceedings of the International Solid State Circuits Conference*, 2008.
13. R. Ho, K.W. Mai and M.A. Horowitz. The future of wires. *Proceedings of the IEEE*, 89(4):490–504, 2001.
14. R. Weerasekera, M. Grange, D. Pamunuwa, H. Tenhunen and L.-R. Zheng. Compact modelling of through-silicon vias (TSVs) in three-dimensional (3-D) integrated circuits. *Proceedings IEEE International Conference on 3D System Integration* (*3D IC*), 2009.

15. S. Perri, P. Corsonello and G. Staino. A low-power sub-nanosecond standard-cells based adder. *Proceedings of the 2003 10th IEEE International Conference on Electronics, Circuits and Systems*, 2003.
16. W.J. Dally, U.J. Kapasi, B. Khailany, J.H. Ahn and A. Das. Stream processors: progammability and efficiency. *Queue*, 2(1):52–62, 2004.
17. U. Nawathe, M. Hassan, K. Yen, L. Warriner, B. Upputuri, D. Greenhill, A. Kumar and H. Park. An 8-Core 64-Thread 64b Power-Efficient SPARC SoC. *Proceedings of the Inetrnational Solid State Circuits Conference*, 2007.