

Energy Savings with Appropriate Interconnection Networks in Parallel DSP

Thorsten Dräger and Gerhard Fettweis
Technische Universität Dresden
Institut für Nachrichtentechnik
Mannesmann Mobilfunk Stiftungslehrstuhl für mobile Nachrichtensysteme
01062 Dresden, Germany
{draeger,fettweis}@ifn.et.tu-dresden.de

Abstract

This paper presents an instruction level power model for a very long instructions word (VLIW) single instruction multiple data (SIMD) digital signal processor (DSP). We show that power consuming memory accesses can be reduced in such SIMD processor architectures with an appropriate network connecting the parallel register files and/or data paths. Several network topologies are analyzed for a variety of digital signal processor algorithms concerning energy issues.

1. Introduction

In recent years the field of mobile computing devices has grown dramatically. More and more computing power as well as long battery lifetimes are requested. This is contradictory, since more computing power is mostly causing higher power consumption. In conjunction with an increasing computation power request, higher power consumption results in higher energy consumption, which reduces battery lifetime. Hence, there is a need for power/energy reduction techniques.

Power may be reduced by software [17] or by hardware [19]. The design for low power cannot be achieved without accurate power prediction. For low power optimization in software an instruction-based power model is needed, which can also be used for low power optimization in hardware. Increasing computing power may be realized by higher frequencies as well as by using parallelism [4] [19]. Parallelism

can be achieved in two independent ways: The instruction set architecture can be parallelized resulting e.g. in a VLIW architecture and parallel data paths can be implemented allowing parallel computation of results e.g. in a SIMD architecture.

These two parallel approaches cause problems but also allow optimization concerning energy. The parallel instruction set architecture complicates the generation and handling of an instruction-level power model. Chapter 2 focuses on this topic. In chapter 3 we show how energy consumption may be reduced in parallel SIMD architectures. But if several parallel data paths are implemented, the architecture needs a network which interconnects the data paths with each other. Today this network is often realized by a crossbar switch, which is known to consume significant power for high parallelism. Chapter 4 analyzes various alternatives to a crossbar switch focusing on energy consumption and also taking speed considerations into account.

2. Instruction-Level Power Model for VLIW

We set up an instruction-level power model for the M3-DSP [3][15], which was developed at *Technische Universität Dresden*. The goal was to predict the average power consumption of the DSP, running a particular sequence of instructions. From this average power and the number of cycles of the given sequence at a certain clock rate it would be possible to calculate the energy consumption. In order to use this power model with an energy optimizing compiler the predicted energy consumption for a given sequence has to be calculated efficiently during compile time [9][10].

First of all, prediction must be performed fast. That means, that exact simulation in a processor model during runtime is not possible, using e.g. *Spice* or physical measurements of the instruction sequence. Hence, results of simulations or measurements must be stored in a model, so that the compiler is able to use the stored results. But not all possible sequences can be stored, because there are too many. Thus the model must contain some basic elements, which can be composed to predict an arbitrary sequence.

Switching operations are the main reason for power dissipation in processors. Switching depends on operands, addresses and instructions. Here we focus on instruction dependent power consumption, using operands and addresses which generate an average operand or address dependent power consumption.

Because of switching activity inter-instruction effects must be analyzed as well as the instructions itself. For the M3-DSP as well as for other processors these effects are not negligible and prohibit a simple per-instruction model [17]. With an instruction pair model the inter-instruction effects can be taken into account, and each instruction sequence can be constructed out of pairs [17].

A complete pair model would generate a table of size $O(n^2)$, where n is the number of different instructions of the instruction set neglecting different addresses or operands. Even building tables of that size would be very time consuming, imagining all different VLIWs. In [7] and [16] it is proposed to group all instructions according to the processor units they are using. Separate tables can be generated for each group, which results in a much smaller power model. Grouping the VLIWs can be done easily by cutting the VLIW according to the functional processor units (FU). These independent instruction groups are called functional instruction words (FIWs). Taking the M3-DSP for instance, the FIW for the address generation unit (AGU) comprises all instructions for memory access and address pointer modification. The FIW for the data path unit (DPU) covers all algorithmic and logical instructions. All six FIWs of the M3-DSP with corresponding instructions are shown in table 1.

To further reduce the table size, the instructions of each FIW can be separated in classes according to their power consumption behavior. As in [8], instructions of the M3-DSP with similar functionality often fall

AGU address generation	DMU data manipulation	DAU data alignment	DTU data transfer	PCU program control	SU side unit
read	MAC	shift	zur. zip	loop	special
write	LOGIC
ptr. mod.

Table 1. part of instruction groups and classes of the M3-DSP

FU	functional instruction word pair	normalized power cons.
all	1 st FIW 2 nd FIW	
	NOP NOP	1.00
AGU	read read	2.24
	read NOP	1.88
	read write	2.93
	read Ptr. modification	1.82
	write write	2.70
	write NOP	2.54
	write Ptr. modification	2.52
	...	
DMU	SISD MAC NOP	1.31
	...	
	SIMD MAC NOP	5.60
	...	
DAU	shift NOP	1.60
	...	
DTI	zurich zip NOP	1.17
	...	
	vector NOP	1.17
	...	
	3-element NOP	1.22
	...	

Table 2. Part of the power model for the M3-DSP

in the same class. All logical instructions (AND, OR etc.) for example have similar properties in power consumption in contrast to e.g. the MAC instruction.

Following [8], the power model is generated by performing physical current measurements of the processor device. This is the most accurate method to gain information for the model. The processor runs an appropriate sequence of a certain instruction pair in an infinite loop while measuring. The sequence should

cycle	slice 0	slice 1	...	slice p-1
t	$b_0 \cdot x(n)$	$b_0 \cdot x(n+1)$...	$b_0 \cdot x(n+p-1)$
t+1	$+b_1 \cdot x(n-1)$	$+b_1 \cdot x(n)$...	$+b_1 \cdot x(n-1+p-1)$
⋮	⋮	⋮	⋮	⋮
t+M	$+b_M \cdot x(n-M)$	$+b_M \cdot x(n-M+1)$...	$+b_M \cdot x(n-M+p-1)$
t+M+1	$b_0 \cdot x(n+p)$	$b_0 \cdot x(n+p+1)$...	$b_0 \cdot x(n+p+p-1)$
⋮	⋮	⋮	⋮	⋮

Figure 1. Parallel FIR-filter implementation calculating one sample per slice

be long enough to neglect the loop instruction itself, or two measurements with sequences of different sizes can be done to correct the measured values.

With the described approach the difference between estimated and measured power consumption is less than 2% for the M3-DSP, which is sufficient for our needs.

A part of the generated power model is shown in table 2. Different FIW pairs are listed with the measured power consumption normalized to the *no operation* (NOP) instruction (all FIW are NOP). Aside from SIMD MAC instructions which perform 16 MAC calculations in parallel, the most energy consuming instructions are memory accesses (*AGU read* and *AGU write*). Hence, one strategy to reduce power dissipation may be the reduction of memory accesses. In the next chapter we show that memory accesses can be reduced in parallel architectures by using data transfers which consume significant less power as shown in table 2.

3. Parallelization Reduces Memory Accesses

Many algorithms for digital signal processing can be parallelized in SIMD manner. So the calculation is calculated on parallel data paths, all performing the same operation. Then the same operands may be used in different data paths in the same cycles or some cycles later. Filter algorithms often show this property. Equation (1) and figure 1 show one possible parallelization of a real M-tap FIR-filter. Here n represents the number of the output sample (y) with $n = 1 \dots N$. In figure 1 case the filter coefficients b_k in all p parallel data paths are the same in one cycle, and all but one input sample values (x) are the same in the next cy-

cle in the neighboring data path. SIMD-parallelization as shown for the FIR-filter was also analyzed for cascaded biquad-filters and lattice-ladder filters with similar reuse opportunities.

$$\begin{aligned}
 y(n) &= \sum_{k=0}^M b_k \cdot x(n-k) \\
 y(n+1) &= \sum_{k=0}^M b_k \cdot x(n+1-k) \\
 &\vdots \\
 y(n+p-1) &= \sum_{k=0}^M b_k \cdot x(n+p-1-k)
 \end{aligned} \tag{1}$$

For sequential calculation of an FIR-filter with element memory support, each of the N output samples needs $O(M)$ filter coefficients and $O(M)$ input samples. This results in $O(M \cdot N)$ memory accesses for all samples. The memory accesses can be reduced by doing parallel calculation, depending on the capabilities of the interconnection network unit (ICU). Most reduction can be achieved for the FIR-scheduling of figure 1, if the ICU supports a broadcast data transfer to pass the appropriate filter coefficient b_k to all data paths, if the ICU supports a *Zurich Zip* [1] data transfer to shift all input samples x by one to the right and filling data path 0 with a new sample, and if N/p is an integer. Thereby only $O(M \cdot N/p)$ memory accesses are needed. If we additionally consider a group memory architecture, in which p data elements are read and written at once per memory access, the number of memory accesses is further reduced by $1/p$.

The latter reduction of memory accesses also reduces energy dissipation, if the group access needs less

then p times the energy of the element access, which depends on memory access implementation. Considering the first reduction approach with parallel calculation and element memory, energy reduction can be achieved if the data transfers needed consume less energy than the memory accesses not needed any more. This depends on the number of data transfers as well as on the energy consumed by each data transfer, and hence on the interconnection network.

Comparing sequential calculation with element memory and parallel calculation with group memory, memory access can be reduced by $1/p^2$. Power reduction depends on memory access implementation and on the interconnection network. In the next chapter different network architectures are analyzed concerning energy consumption.

4. Interconnection Network

We have seen that in the M3-DSP with its ICU, constructed of multiplexer chains forming a bus network, energy dissipation can be reduced by exchanging memory accesses with data transfers in parallel algorithms. Today's parallel DSP architectures commonly use crossbar switches as connection networks. This network may be efficient for small parallelism but grows with $O(p^2)$ for p parallel data paths. Hence, other efficient networks must be found for architectures with higher parallelism. In the following we analyze flexible stage networks, like the M3's multiplexer chain, and well known fixed stage networks, also composed of multiplexer chains, to find an efficient ICU architecture.

A network can be characterized by three parameters: topology, routing and flow control. In the topology or interconnection graph the vertices correspond to the network nodes, which are multiplexers connecting other multiplexer with a data path. The edges are the physical connections of the nodes. To find an effective topology five parameters can be found: Bisection width which measures the wiring density of the network, in and out degree which corresponds to the number of input and output wires, diameter which is the longest shortest path between two nodes, wire length and symmetry[2]. Concentrating on energy efficient networks, an energy estimation model is used to evaluate a network's energy consumption.

4.1. Network Power Estimation

In CMOS designs, power dissipation can be classified into three sources: switching power P_{sw} , internal or short circuit power P_{sc} and static power P_{st} [13]. Equation (2) shows the power dissipation for one node. These power dissipation sources can be mapped to the out degree and the wire length of the network topology.

$$P_{per\ node} = P_{sw} + P_{sc} + P_{st} \quad (2)$$

Switching power describes the power dissipation of charging and discharging the node capacitance (3). The transition density D is the average number of transitions per time [11]. It refers to whether a node is used or not during a data transfer performed on the network. This is depending on the routing of the network and the data transfer itself. The fanout corresponds to the out degree of the node. The load capacitance C_{load} can be formed of the gate capacitance C_{gate} and the wire capacitance C_{wire} (5). Here, we assume long wires and refer to [5], which results in much more wire than gate capacitance (6). Also, the importance of wire capacitance for the load capacitance increases with each migration in technology [13][5]. Hence, for simplicity we neglect the gate capacitance in this model (7). This results in a switching power proportional to the total wire length over the out degree of a node, depending on the network routing and on data transfers (4).

$$P_{sw} = \frac{V_{dd}^2}{2} \cdot \sum_i^{fanout} C_{load,i} \cdot D \quad (3)$$

$$\approx a \cdot \sum_i^{fanout} l_i \cdot D \quad (4)$$

$$\text{with } C_{load} = C_{gate} + C_{wire} \quad (5)$$

$$\text{and } C_{gate} \ll C_{wire} \quad (6)$$

$$\Rightarrow C_{load} \approx C_{wire} \sim l_{wire} \quad (7)$$

Short circuit power occurs, when both p and n channel devices conduct simultaneously, and thereby establishing a short. This power is composed of the supply voltage V_{DD} and the peak current I_{peak} (8), which we assume to be constant values [6][14]. The short circuit power appears only if a transition is performed

and hence is only depending on the transition density and a constant factor (9).

$$P_{sc} = V_{dd} \cdot I_{peak} \cdot D \quad (8)$$

$$= b \cdot D \quad (9)$$

$$\text{with } I_{peak} \approx const \quad (10)$$

Static power is considered to be insignificant in CMOS technology [13], and is therefore neglected. For the resulting power model (11) short-circuit power is shown to be typically between 10 – 30% of total power dissipation and switching power between 70 – 90% of total power [13]. Hence values for factors a and b can be received by simulation of networks performing various data transfers. Using the presented power model, energy optimal routing can be found by simulation.

$$P_{per\ node} \approx \left(a \cdot \sum_i^{fanout} l_i + b \right) \cdot D \quad (11)$$

4.2. Analysis

To find an appropriate energy efficient network, we examined four classes of network topologies with eight nodes. One class is the bus topology without extra skip (as the M3-DSPs ICU), with extra skip of length two, and with extra skip of both length two and four. Another class is the ring topology also without or with extra skips. Together with the hypercube topology these networks are all flexible stage networks, because they may use any given number of stages to perform a data transfer. Also the omega and butterfly topologies were simulated. These are fixed stage networks: all data transfers use the three stages of the network. Because the maximum number of used stages (s_{max}) may increase the critical path of the overall design, we define an upper limit for this number regarding flexible stage networks. For the bus network without extra skip we allow seven stages, because otherwise a transfer from first to last node wouldn't be possible in one step. In the ring network without extra skip we restrict s_{max} to four for the same reason. In the other bus and ring networks we reduce s_{max} by one for every extra skip. In the hypercube we restrict s_{max} to three for the same reason as for bus and ring.

Bus and ring topologies are shown in figure 2, for the other networks we refer to [18]. To analyze the influence of layout mapping, different layouts are considered, shown in figure 3.

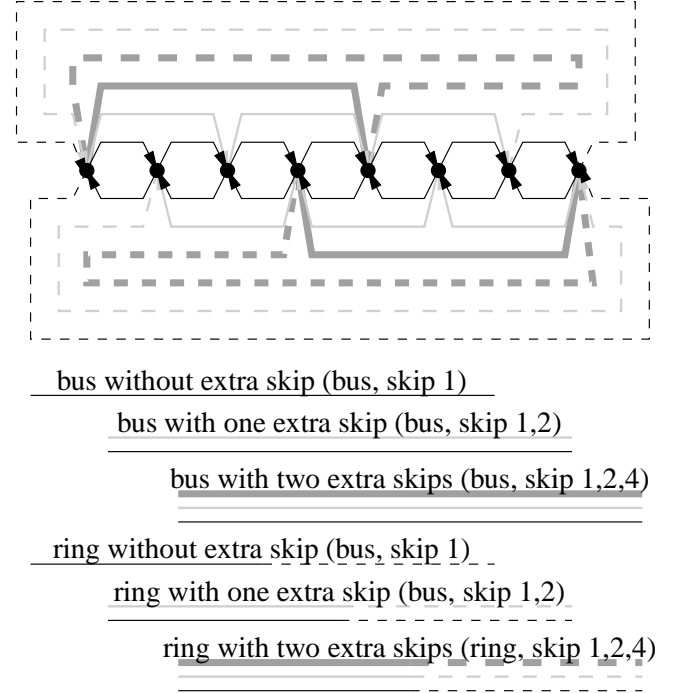


Figure 2. Bus and ring networks

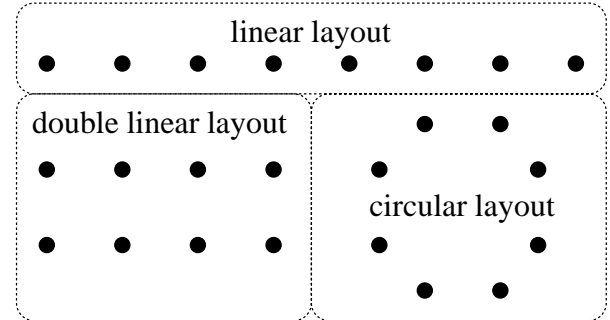


Figure 3. Different layouts

We simulated all data transfers (DT) needed for our target algorithms. In addition to the broadcast and zurich zip data transfer needed for FIR-filtering we took the *rotation* DT (e.g. (0; 1; 2; 3; 4; 5; 6; 7) → (2; 3; 4; 5; 6; 7; 0; 1), which means the value of node 0 is transferred to node 2 etc.). Another data transfer we considered, was the *swap* DT, which exchanges

the values of node pairs as in $(0; 1; 2; 3; 4; 5; 6; 7) \rightarrow (3; 4; 5; 0; 1; 2; 7; 6)$. We also took data transfers into account used by Cooley-Tuckey-FFT and Singleton-FFT/VITERBI algorithms including bit-reversal. Here we refer to [12].

Considering group memory, data may not be aligned properly having an offset as illustrated in figure 4. We therefore simulated all described data transfers with all possible offsets in a next step. To find out if specific target data transfers have a significant impact on the performance of networks, random data transfers are analyzed in a third step.

slice no.	0	1	2	3	4	5	6	7
aligned data	d_0	d_1	d_2	d_3	d_4	d_5	d_6	d_7
non-aligned data (offset=3)	X	X	X	d_0	d_1	d_2	d_3	d_4

Figure 4. aligned and non-aligned data

5. Results

After simulating all combinations of different topologies, layouts and data transfers figure 5 shows the average power consumption over all data transfers without offset for each particular topology in different layouts. The general result is independent of the layout: flexible stage networks outperform fixed stage networks regarding power dissipation. Butterfly or omega networks consume 110%–330% more power than the other networks. Regarding flexible stage networks only, skip networks in ring or bus structure consume 15%–35% less energy than hypercube networks. In bus and ring energy consumption can be reduced with each additional skip. Comparing bus and ring networks with equal number of extra skips ring networks outperform bus networks: power dissipation is 35% smaller at most.

Concentrating on layout issues networks in linear layout under perform networks in double linear layout, which in turn under perform networks in circular layout. But for bus topologies the differences are significantly smaller than for the other networks. Ring and hypercube topologies don't profit from circular layout instead of double linear layout either.

Figure 6 shows the energy consumption for the particular data transfers without offset for all network

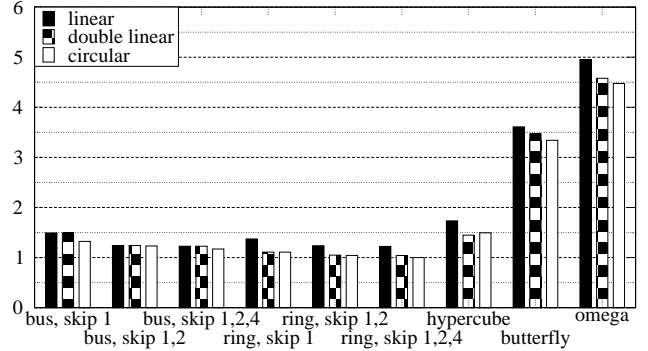


Figure 5. Average energy consumption for different layouts

topologies in double linear layout. The energy dissipation differs a lot comparing the results for different data transfers performed in the same network. In general broadcast data transfers need the least energy and the swap data transfer the most energy for all networks. The difference is 65%–240%. However, classification in energy expensive and non-expensive data transfers is not always possible. The zurich zip data transfer for instance, performs good on ring networks and bad on hypercube networks: in ring networks 20% more energy is consumed comparing broadcast with zurich zip data transfers, in hypercube networks the difference is 70%. Comparing all networks' and data transfers' energy consumption, rotation transfers perform significantly good on ring networks and Cooley-Tuckey-FFT transfers perform best on hypercube networks.

Considering all data transfers with offset, the average energy consumption is significantly higher for all networks than without offset, as depicted in figure 7. This effect is most notably for all ring networks and for bus networks with extra skip. Here, the energy consumption is 30% higher than without offset, whereas for the other networks the difference is less than 10%. This means, considering offsets equalizes the energy performance of the networks: hypercubes perform better than bus networks and the advantage of ring networks decreases considering offsets. For random transfers, energy consumption increases by another 10-20% compared with the data transfers with offset.

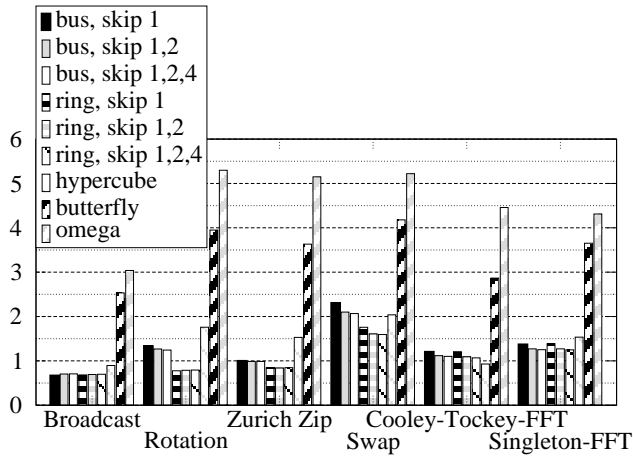


Figure 6. Energy consumption of data transfer classes in double linear layout

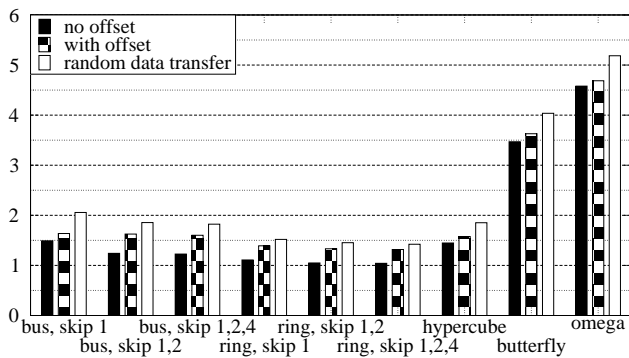


Figure 7. Energy consumption of data transfer classes with additional offset and random data transfers in double linear layout

Network efficiency is not only determined by energy consumption, but also by cycles needed to perform a data transfer. Figure 8 shows the average cycle count for all data transfers in the different networks in double linear layout. Here fixed stages networks outperform most of the other networks. In bus and ring networks every additional skip reduces the average cycle count significantly. Bus networks with two extra skips and ring networks with at least one extra skip outperform hypercubes. Ring networks with two extra skips perform even better than fixed stage networks. Considering data transfers with offset and random data transfers the results remain similar.

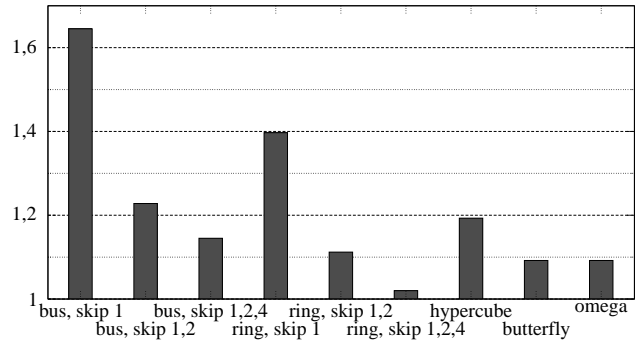


Figure 8. Average cycle count

As far as energy consumption is concerned, bus or even better ring networks outperform all others the more extra skips the better. Also for cycle count considerations ring networks with extra skips show very good performance. But implementation and routing may cause problems with these bus or ring networks with extra skips.

6. Conclusions

In chapter 2 of this paper we set up an instruction-level power model. The model complexity was reduced by generating separate lookup tables for energy consumption independent instruction groups. These groups were found, cutting the very long instruction word into its functional instruction words.

According to the generated power model of the M3-DSP, memory accesses are one of the most power consuming instructions, whereas data transfers dissipate significant small power. In chapter 3 it was shown, that memory accesses can be reduced in SIMD parallel architectures. The reduction is depending on the number of parallel data paths. It is realized by performing group memory accesses instead of element accesses, and by exchanging memory accesses in general with data transfers.

To achieve energy reduction by exchanging memory accesses with data transfers, an interconnection network is needed that realizes all requested transfers and consume little power. By analyzing various possible interconnection networks based on multiplexer chains in chapter 4, we found that flexible stage networks have significant less power dissipation than fixed stage networks. Different layouts don't change

the ranking of the different topologies. But for different data transfers the ranking does change. Therefore, we suggest application specific construction of the interconnection network. Considering non-aligned data in group memory, the topology specific differences in power dissipation are reduced. Hence, a separate pre-aligning process might be taken into account.

The first choice network considering energy consumption as well as cycle count and the number of maximum used stages is the ring topology with extra skips. The weak points are implementation and routing aspects, which have to be further analyzed. Good alternatives are bus networks with extra skips which reduce implementation problems but also have difficult routing, or the hypercube topology with easy routing but slightly less performance in energy consumption.

References

- [1] J. Beraud and C. Galand. Digital Signal Processor Architecture With Plural Multiply/Accumulate Devices. *U.S. Patent no. 5 175 702, IBM Corp.*, December 29 1992.
- [2] W. Dally. *VLSI and Parallel Computation*, chapter 3. Morgan Kaufmann Publishers, Inc. San Mateo, California, 1990.
- [3] G. P. Fettweis, M. Weiss, W. Drescher, U. Walther, F. Engel, S. Kobayashi, and T. Richter. Breaking new grounds over 3000 mops: A broadband mobile multimedia modem dsp. In *ICSPAT'98. Toronto*, 13.-16. September 1998.
- [4] J. L. Hennessy and D. A. Patterson. *Computer Architecture a Quantitative Approach*. Morgan Kaufmann Publishers, Inc., second edition, 1996.
- [5] R. Ho, K. W. Mai, and M. A. Horowitz. The Future of Wires. In *Proceedings of the IEEE*, volume 89, April 2001.
- [6] Y. I. Ismail and E. G. Friedman. Dynamic and Short-Circuit Power of CMOS Gates driving Lossless Transmission Lines. *IEEE Transactions on Circuits and Systems*, 46(8), August 1999.
- [7] B. Klass, D. Thomas, H. Schmit, and D. Nagle. Modeling Inter-Instruction Energy Effects in a Digital Signal Processor. *Power-Driven Microarchitecture Workshop, International Symposium on Computer Architecture*, June 1998.
- [8] M. Lee, V. Tiwari, S. Malik, and M. Fujita. Power analysis and minimization techniques for embedded DSP software. *IEEE Trans. VLSI Systems*, 5(1):123–135, March 1997.
- [9] M. Lorenz, T. Dräger, R. Leupers, P. Marwedel, and G. Fettweis. Low-Energy DSP Code Generation Using a Generic Algorithm. In *Proceedings of International Conference on Computer Design (ICCD) 2001, Austin, Texas*, pages 431–437, September 2001.
- [10] M. Lorenz and P. Marwedel. Energiebewusste Compilierung für Digitale Signalprozessoren. In *this conference book*.
- [11] F. N. Najm. Transition Density: A New Measure of Activity in Digital Circuits. *IEEE Transactions on Computer Aided Design of Intergrated Circuits and Systems*, pages 310–323, February 1993.
- [12] A. V. Oppenheim and R. W. Schaffer. *Discrete-Time Signal Processing*. Prentice–Hall, 1989.
- [13] M. Pedram. Power Minimization in IC Design: Principles and Applications. *ACM Transactions on Design Automation of Electronic Systems*, 1(1):3–56, January 1996.
- [14] D. Rabe and W. Nebel. Short circuit power consumption of glitches. In *Proceedings of the 1996 international symposium on Low power electronics and design*, pages 125–128, Monterey, CA USA, August 1996.
- [15] T. Richter, W. Drescher, F. Engel, S. Kobayashi, V. Nikolajevic, M. Weiss, and G. Fettweis. A platform-based highly parallel digital signal processor. In *Proc. CICC 2001*, pages 305–308, San Diego, USA, 6.-9. May 2001.
- [16] S. Steinke, M. Knauer, L. Wehmeyer, and P. Marwedel. An Accurate and Fine Grain Instruction-Level Energy Model Supporting Optimizations. *International Workshop - Power and Timing Modeling, Optimization and Simulation, Yverdon-les-bains, Switzerland*, September 2001.
- [17] V. Tiwari, S. Malik, and A. Wolfe. Power analysis of embedded software: a first step towards software power minimization. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2(4):437–445, December 1994.
- [18] A. Varma and C. Raghavendra. *Interconnection networks for multiprocessor and multicomputers: theory and practice: manuscript for IEEE tutorial text*. IEEE Computer Society Press, 1994.
- [19] G. Yeap and F. Najm, editors. *Low Power VLSI Design and Technology*. World Scientific Publishing Co., 1996.