

System Level Optimization and Design Space Exploration for Low Power

Ansgar Stammermann, Lars Kruse, Wolfgang Nebel, Alexander Pratsch,
Eike Schmidt, Milan Schulte, and Arne Schulz

OFFIS Research Institute
Oldenburg, Germany

ansgar.stammermann@offis.de

ABSTRACT

We present a software tool for power dissipation analysis and optimization on the algorithmic abstraction level from C/C++ and VHDL descriptions. An analysis is most efficient on such a high level since the influence of design decisions on the power demand increases with increasing abstraction [1]. The ORINOCO[®] tool enables to compare different but functionally equivalent algorithms and bindings to RT-level architectures with respect to power consumption. The results of the optimized binding can be used to guide synthesis. In the experimental evaluation we compare the predicted optimization trend with synthesized implementations and prove the accuracy of our methodology and tool.

1. INTRODUCTION

Over the last few years, the integration density as well as the clock rate of microelectronic circuits have increased enormously. Today, a chip consists of several million gates. This development has led to the fact that the power dissipation of such systems has become an increasingly important criterion in the design process. A higher power dissipation reduces the battery operating time of mobile applications, raises the production costs and lessens the reliability of the circuit. A lot of techniques have already been proposed to take power consumption into account in high level synthesis. Among them are optimizations like algebraic transformations, usage of multiple supply voltages, loop transformations etc. See [2] for detailed overview.

In the extremely complex design of microelectronic circuits, the problem of estimating the effects of different design alternatives on the power dissipation arises. Design decisions made in a very early phase of the development process, in which the design consists of a yet very abstract description (algorithmic abstraction level), have the greatest influence on the power dissipation (Figure 1). Previously existing design tools require a highly advanced design process, however, in order to execute a power

dissipation analysis. Consequently, nearly complete iterations through the development process are necessary in order to measure the power consumption and meet power constraints. Such a procedure is time-consuming, cost-intensive and incompatible with the ever-shorter innovation cycles in circuit design.

The development of the software called ORINOCO[®] (OFFIS Research INstitute pOwer Characterizer, estimator and Optimizer) intends to create a power dissipation analysis and optimization on the algorithmic abstraction level. It allows for flexible treatment of different circuit variants and the ability to consider the influence of processed data on power dissipation.

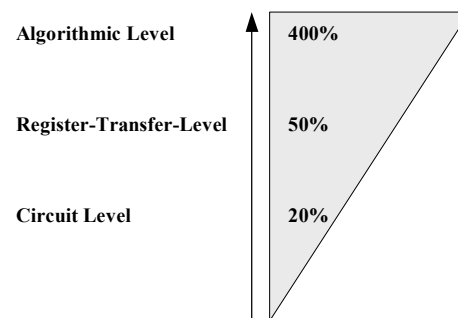


Figure 1. Influence of design decisions on the power dissipation at different levels of abstraction [1]

In Section 2 of the paper we describe the overall design flow for high level power optimization with ORINOCO[®]. The underlying analysis and optimization techniques are detailed in Section 3. An experimental evaluation is presented in Section 4 before providing a short summary in Section 5.

2. DESIGN FLOW

Our power dissipation analysis assumes the design flow depicted in Figure 2. The designer creates a system specification. This specification is subdivided into its different subtasks (tasks). For example, with an image processing chip, tasks could include motion estimation or a Fourier transform. For the realization of a task, several algorithms typically exist. As an example, a Fourier transform can be executed via a matrix multiplication or a fast Fourier transform. When an algorithm is found, the appurtenant

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISSS '01, October 1-3, 2001, Montreal, Quebec, Canada.
Copyright 2001 ACM 1-58113-418-5/01/0010...\$5.00.

architecture is determined during the high-level synthesis (scheduling, allocation, binding), i.e. the algorithm is transformed into hardware. Again, there are many design decisions to be made. Scheduling determines during which clock cycle an operation will be executed. In the allocation task, the type and number of the components to be used (adders, multipliers etc.) is determined. In doing so, components are distinguished not only by their function, but also by their internal architecture. An adder can be realized as a carry-ripple or carry-select adder. "Binding" means the mapping of operations onto the components that were determined during allocation.

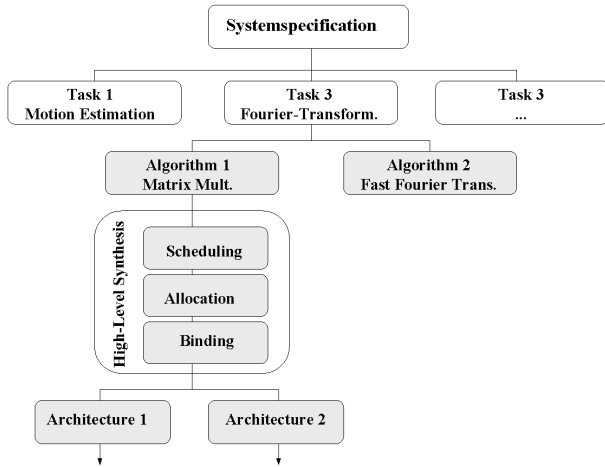


Figure 2. Design Flow

In regard to power dissipation, the designer must confront the following questions:

1. Which algorithm is the best?
2. What does the best architecture for an algorithm look like?
3. Which is the best architecture of a component?
4. How does the bit width of a component influence the power dissipation ?

It is the goal of the ORINOCO tool[®] to give the designer assistance in answering these questions.

3. UNDERLYING TECHNIQUE

The power dissipation analysis is based on simulation. A profiling of the hardware description, which can exist both in VHDL as well as in C/C++, is carried out in order to collect data statistics. At the same time, the expected circuit structure is derived from a control data flow graph, without carrying out a complete synthesis. Components of the RT level, upon which the design is later mapped, are first characterized [3]. The power assessment of the memory also takes place based on a characterization [4].

3.1 Model generation

For the power dissipation, analysis models are needed that describe the power dissipation of the individual components of the RT level. In general, the power dissipation depends on the input data, the characteristics of the components (e.g. bit width and architecture,) and the underlying cell library or technology.

The *Component Modeler* allows for the generation of such models (Figure 3). The models may contain parameters of both bit width and architecture. Once generated, models are stored in a library and are available for future analysis and optimization [3, 4].

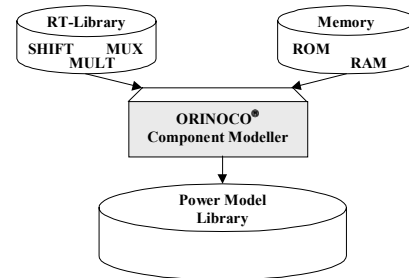


Figure 3. Model generation

Moreover, an open interface allows for the usage of user-defined components. This enables the simultaneous employment of IPs.

3.2 Low power allocation and binding

"Binding" concerns the mapping of operations or variables onto components during the high-level synthesis. A low power binding is a mapping in which the power dissipation of the components is minimal. Binding has a great influence on power dissipation, since different bindings lead to different input currents on the inputs of the components. Let's assume that two 1-bit variables with the input streams (1,1,0,0) and (1,1,0,0), respectively, are mapped to a register each (no resource sharing), thus leading to two transitions on the inputs. If the variables are mapped onto a single register (resource sharing), however, only one transition (1,1,1,1,0,0,0,0) arises. The data flows (1,1,0,0) and (0,0,1,1) on the other hand produce six transitions with only a single register and two transitions with two registers.

The basis for our calculation is a control data flow graph (CDFG) and simulation data. Through an analysis of the graph, a complete ordering of the execution sequence of operations of the same type (e.g. addition, multiplication) is determined (scheduling). This ordering determines which operations can be mapped together onto a component and which cannot. With the simulation data and the power dissipation models, the electric power consumption for a given binding of a component can be determined.

On the basis of these data, a lower and an upper bound to the power dissipation for each component type are calculated [5]. The problem is formulated as a bipartite-weighted matching problem, which is solvable in $O(n^3)$. The bounds calculated in this way do not necessarily correspond to a legal binding, i.e. no appropriate architecture can be synthesized. Therefore, heuristics are introduced in [6] that form a legal binding out of an illegal solution. A binding found in this way is not necessarily equal to the minimum or the maximum in regard to the power dissipation, but comes very close, as shown in our experimental evaluations.

3.3 Work flow

The starting point for the analysis is the hardware description and its test bench (Figure 4). Additionally, a few boundary conditions must be set by the user:

The technology must be determined. The appropriate power dissipation model for this technology is assigned to each component type. Additionally, a component type must be assigned to each operator in the description. This determines components upon which an operator of the description may be mapped. Operations on the critical path can thus be realized through high-consumption, fast components, while others can be realized through low-consumption, slow components.

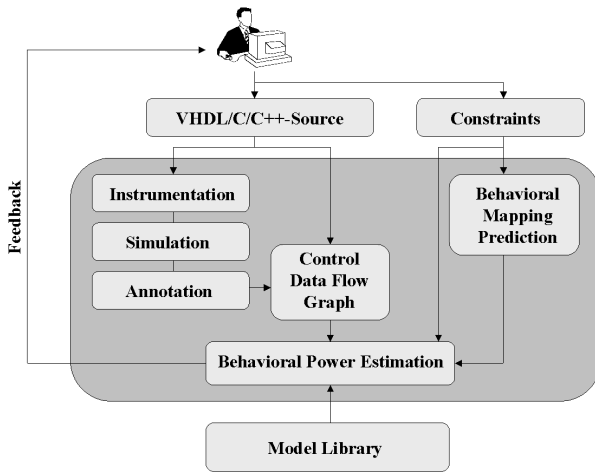


Figure 4. ORINOCO® Work flow

ORINOCO® is capable of reading in both VHDL as well as C/C++ descriptions. The work flow of the tool varies, however, depending on the language used. VHDL descriptions are simulated whereas C/C++ source code are translated and then executed.

3.3.1 VHDL description

The VHDL description is simulated in order to determine the input data streams on the individual operations. The description is automatically instrumented for that purpose. Special routines are inserted into the specification that write the input data streams of the individual operations into an activity file during a simulation. Simulation is possible with every commercial simulator (VSS, ModelSim, etc.). The work flow for VHDL descriptions is as follows:

1. The VHDL description is compiled and elaborated.
2. A control data flow graph is generated from the elaborated design.
3. The VHDL description is instrumented.
4. The modified VHDL description is simulated.
5. The input data streams from the activity file are annotated on the corresponding operations in the graph.
6. The order of execution of the operations is determined (scheduling).

3.3.2 C/C++ description

In contrast to VHDL, C/C++ descriptions are not simulated. They are translated and executed in order to determine the input data streams on the operations. During the translation, the generated

program is automatically instrumented. Special test routines that write input data streams into an activity file during execution are inserted. At the same time, a structure file is created, from which a control data flow graph is later generated. The work flow for C/C++ descriptions is as follows:

1. The C/C++ description is translated. In doing so, the executable program is instrumented and at the same time a structure file is written.
2. The program is executed.
3. A control data flow graph is generated from the structure file.
4. The input data streams from the activity file are annotated on the corresponding operations in the graph.
5. The order of execution of the operations is determined.

3.4 Power dissipation analysis

A large part of the power dissipation analysis is based on the solution of the low power binding. For every employed component type, the binding is calculated with a minimal (light gray) and maximal (dark gray) power dissipation for each possible number of resources (allocation) (Figure 5). The difference between the bars corresponds to the power dissipation that can be influenced by the selection of the binding. The underlying reservation tables can be written to a file for later guidance of the synthesis tool.

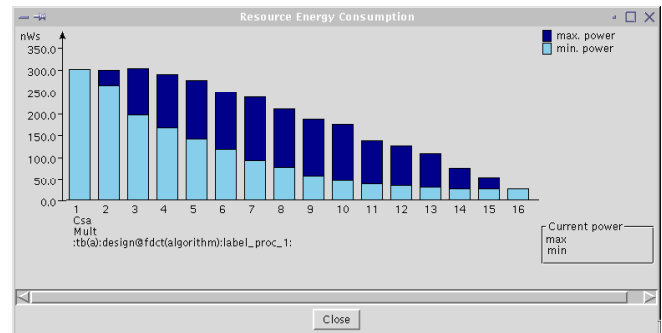


Figure 5. Minimal and maximal power dissipation of different bindings

The power dissipation assessment of the connecting structures, connecting wires and multiplexors is based on the power dissipation-optimal bindings. The interconnect structures are generated by a architecture extraction. For each input of a component that possesses two or more sources, a multiplexor is instantiated. This can emerge through control structures or resource sharing. The control structures are extracted from the control data flow graph. Resource sharing follows from the given binding. The power dissipation of the multiplexor is calculated from the simulation data and the corresponding power dissipation model. For the calculation of the power demand of the wires, their individual lengths must first be determined. The assessment of length is based on an extension of the Growth-Limited Multifold Clustering (GLMC) technique [7, 8]. This algorithm is based on clustering, a constructive placement algorithm. Together with simulation data and the wire model, the power dissipation for each wire is calculated. The power dissipation of memories and IO

ports is also calculated on the basis of the power models. In contrast to the interconnect structures, their number and characteristics are, however, independent of the underlying binding.

4. RESULTS

For the evaluation, the degree to which ORINOCO[®] is capable of predicting the relative power consumption of different algorithms performing the same functionality was evaluated. In a further step, the capability to optimize the power demand of different architectures for a single algorithm was examined. For comparison of different architectures for a real life example please refer to [9].

4.1 Analysis of different algorithms

For the analysis of different algorithms, the DIFFEQ benchmark [10], an algorithm for the solution of a differential equation, was chosen. This benchmark is synthesizable with the Synopsys Behavioral Compiler[®] and an appurtenant test bench supplies the necessary input data for the simulation. In Table 1, a section of the original description of the algorithm is represented under algorithm 1. This part of the description was modified in algorithm 2 and 3 without changing the semantics. First the expression $3*d$ and next the variable u were factored out.

Table 1. Different algorithms for DIFFEQ benchmark

Algorithm 1	$u := u - 3 * d * u * x - 3 * d * y$
Algorithm 2	$u := u - 3 * d * (u * x + y)$
Algorithm 3	$u := u * (1 - 3 * d * x) - 3 * d * y$

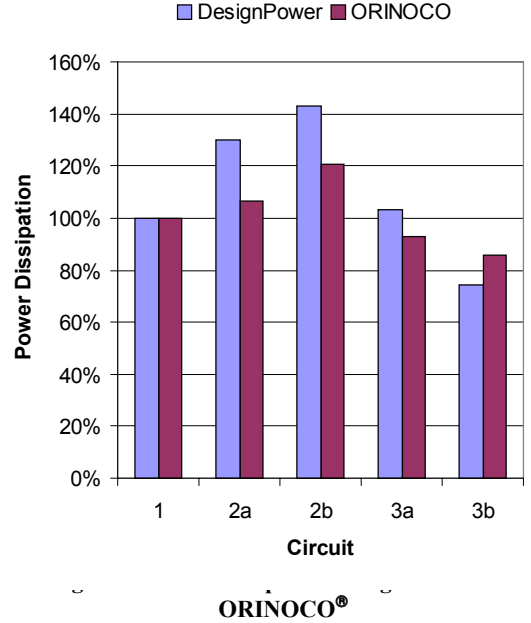
ORINOCO[®] first calculated the power dissipation for each algorithm. Subsequently, the algorithms were synthesized to gate level circuits. In the synthesis, the optimized bindings calculated by ORINOCO[®] were taken into consideration. The power dissipation of these synthesized circuits was estimated with Synopsys DesignPower[®].

In Figure 7, the normalized results for each circuit are contrasted. The power dissipation increased due to the modification in circuit 2. In 3, on the other hand, the power demand could be lowered. In 2b the power optimized bindings calculated by ORINOCO[®] were not considered. For both tools, no resource sharing was permitted. The power demand increased as expected. In 3b the multiplication with the constant 3 was replaced by the following instructions:

$$\begin{aligned}
 & u := u * (1 - 3 * d * x) - 3 * d * y \\
 \Rightarrow & t := 2 * d \\
 & t := t + d \\
 & u := u * (1 - t * x) - t * y
 \end{aligned}$$

Since a multiplication with the constant 2 is realized by a low-consumption shift, the power demand could thus be further lowered.

As is to be seen from the results, since the trend predicted by ORINOCO[®] from the algorithm could be confirmed by the synthesized circuit a prediction of the effects of different design alternatives is already possible with ORINOCO[®] at this point.



4.2 Analysis of different architectures for a single algorithm

As an example for the analysis of different architectures, a VHDL description of a FDCT was chosen. The circuit contains 16 multiplications that are considered in the following. Each multiplication is assigned to the component type CSA multiplier, i.e. all multiplications of the specification could be mapped onto a CSA multiplier.

ORINOCO[®] calculates the binding for each number of multipliers (1 to 16) with a minimal and maximal power dissipation. Subsequently, the circuit was synthesized for every number of multipliers with consideration to the best and worst binding for that gate level. The power dissipations for the synthesized architectures determined with Synopsys DesignPower[®] are entered in Table 2. No difference is present for 1 and 16 multipliers, since only one possible binding exists there.

The results reflect the trend predicted by ORINOCO[®]. The least power dissipation can be attained e.g. for a good binding with 14 or 15 multipliers (cf. Figure 5).

Table 2. Influence of the binding on the power dissipation

No. of mult.	Best binding [uW]	Worst binding [uW]	Difference
1	264,07	264,07	0,00
2	235,86	340,38	104,52
3	196,19	388,02	191,83
4	165,28	343,92	178,64
5	218,61	343,67	125,06
6	246,04	331,53	85,50
7	195,64	312,53	116,89
8	189,39	299,96	110,57
9	199,18	303,49	104,31
10	187,36	198,34	10,98
11	158,59	163,51	4,92
12	115,34	157,33	41,99
13	77,90	176,42	98,53
14	73,16	118,76	45,60
15	98,06	143,59	45,53
16	82,90	82,90	0,00

5. SUMMARY

ORINOCO[®] allows the designer to already make an efficient analysis and optimization of the design on the algorithmic level of abstraction. On this level, the influence of design decisions on the power dissipation increases considerably. It therefore makes possible decisive improvements in the design and compliance with given power consumption limits.

6. REFERENCES

- [1] A. Raghunathan, N.K. Niraj, S. Dey, "High-Level Power Analysis and Optimization", Kluwer Academic Publishers, 1998
- [2] E. Macii, M. Pedram, F. Somenzi, "High Level Power Modeling, Estimation, and Optimization", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 17(11), pp. 1061 – 1079, November 1998
- [3] G. Jochens, L. Kruse, E. Schmidt, W. Nebel, "A New Parameterizable Power Macro-Model for Datapath Components", in Proc. for DATE, pp. 29-36, 1999
- [4] E. Schmidt, G. Jochens, L. Kruse, W. Nebel, "Automatic Nonlinear Memory Power Modelling", in Proc. for DATE, p. 808, 2001
- [5] L. Kruse, E. Schmidt, G. Jochens, A. Stammermann, A. Schulz, E. Macii, and W. Nebel, "Estimation of Lower and Upper Bounds on the Power Consumption from Scheduled Data Flow Graphs", IEEE Transactions on Very Large Scale Integration (VLSI) Systems", pp. 3 – 14, February 2001
- [6] L. Kruse, E. Schmidt, G. Jochens, A. Stammermann, W. Nebel, "Low Power Binding Heuristics", PATMOS-99, pp. 41-50, 1999
- [7] A. Alvandpour, "Power Estimation and Low Power CMOS Circuit Techniques", Chapter 4, Linköping Studies in Science and Technology, Dissertation Nr. 587, 1999
- [8] A. Alvandpour, P. Larsson-Edefors, C. Svenssons, "GLMC: Interconnect Length Estimation by Growth-Limited Multifold Clustering", Proceedings of IEEE International Symposium on Circuits and Systems, pp. V 465-8, Geneva, Switzerland, 2000
- [9] A. Allara, M. Bombana, L. Kruse, E. Schmidt, A. Stammermann, "VHDL Behavioural Power Estimation for Telecom Devices", accepted at FDL, Lyon, September 2001
- [10] N. Dutt and C. Ramachandran, Benchmarks for the 1992 high level synthesis workshop, Tech. Rep. #92-107, Dep. Inform. Comput. Sci., Univ. California, Irvine, 1992