Comparing Executable Specifications regarding Power at Algorithmic Level (ANSI-C/SystemC)

Frank Poppen, OFFIS Institute for Information Technology
Alexander Jährling, Chip Vision Design Systems AG
Wolfgang Nebel, OFFIS and Oldenburg University

ABSTRACT

In this paper we demonstrate that estimating power at abstraction levels higher than gate or even RTL early in the design flow is a feasible approach towards broadening the design-space exploration process, shortening time to market and designing integrated circuits with reduced power dissipation. We compare different implementations of a benchmark design at different levels of abstraction, starting from the algorithm level. Different algorithms for a 128-point FFT/iFFT processor for ultrawide band communication systems [8] are estimated for power using the ChipVision tool ORINOCO [7]. We compare the results with estimations at lower abstraction levels using the Cadence tools RTL Compiler, BuildGates and ncsim. We conclude that it is possible to make the right decisions regarding power at algorithmic abstraction level without coding a single line of HDL.

1 Introduction

Next to the classic constraints as area and timing, designing in the nanometer scales comes up with even more daunting challenges like signal integrity, design for yield and manufacturing or power dissipation. The latter is actually not a new challenge and EDA tools are available that allow for gate- and RT-level power estimation and optimization. For maximum savings though, it is mandatory to consider power in the earliest stages of a design flow, since only at highest levels of abstraction, e.g. algorithmic level, the complete design-space is still open for exploration and optimization.

Figure 1 illustrates the covered design-space starting from different levels of abstraction. The term design-space qualifies the set of all possible solutions to implement one specification in silicon. The boundaries of the design-space are defined by design constraints and the limitations of the design-technology. Each descent in the abstraction level during the design process establishes implementation decisions that narrow the cone of reachable solutions until – in the end – one single solution remains.

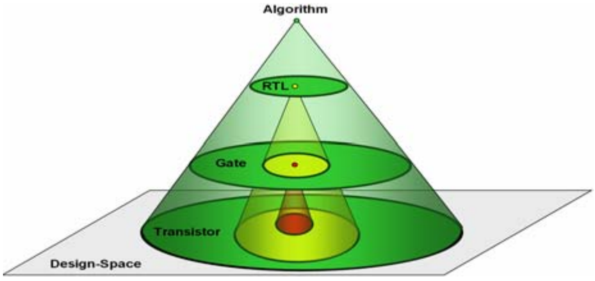

Figure 1: Design-space exploration starting from different abstraction levels.

Ignoring design constraints as for example power at high abstraction levels, will most likely lead to the situation that the cone of reachable solution is already too tight to satisfy the requirements and an iteration step back up the abstraction level is necessary to widen the cone again. This is time consuming, costly and should be avoided.

With this in mind a framework was developed in the European project POET (Power Optimization for Embedded Systems). POET was funded by the European Union’s program IST (Information Society Technology) for a period of three and a half years. The main objective was to develop a new design methodology and tool suite for power estimation and optimization in heterogeneous embedded System on Chip (SoC) designs. The key innovation of the approach is to enable design-space exploration for low power system architectures, algorithm optimizations and system partitioning. Amongst others, the POET tools base on [1-6] and manage and optimize all major contributors to power dissipation in large SoC designs.

In this paper we compare different design implementations of a 128-point FFT/iFFT processor for ultrawideband communication systems [8] at different levels of abstraction in the design flow regarding the power dissipation. A standard methodology would be, to manually translate a C-golden-model into RTL-HDL, synthesize the design and perform gate-level power estimations after simulating the design for activity extraction and back-annotation. This methodology gives quite accurate estimation results but makes an evaluation of different C-algorithms time consuming and therefore costly. The idea is, to enable a development team to directly compare C/SystemC algorithms at the algorithmic level. Different C-algorithms of a FFT/iFFT are compared with ChipVision’s high level power estimation tool

ORINOCO [7]. The results are being compared with precise estimations at gate-level using the Cadence tools RTL Compiler, BuildGates and ncsim. By this, the paper shows that it is possible to make the right decisions regarding power dissipation at algorithmic abstraction level without coding a single line of HDL.

In the following Chapter 2, we introduce part of the POET design flow that was used to generate the presented results. The benchmark design is briefly commenced in Chapter 3. More information on the design can be found in [8]. The core of this work is Chapter 4 with results and a discussion. The document finishes with the conclusions in Chapter 5 followed by the references and authors information.

2 Tool Chain

Part of the tool chain developed within the POET project is coarsely depicted on the left side of Figure 2. The system is being defined in either ANSI-C or SystemC. ChipVision's tool ORINOCO [7] instruments the code with functions that record the dataflow activity during execution of the specification. This step is very similar to other activity extraction methodologies e.g. Cadence TCF or Synopsys SAIF. This information serves as input to the tool together with a library of functional units (FU). Included are power models of e.g. adders, subtractors, multipliers, registers, etc., specially generated for the chosen target technology. In this case, a regular, commercially available standard cell library for TSMC's 90 nm process.

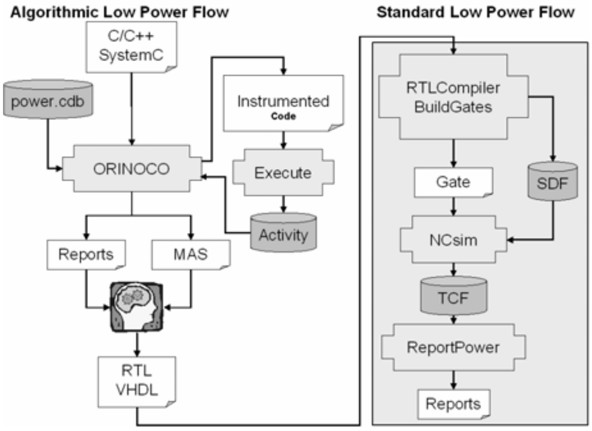

Figure 2: Overview design flow.

After scheduling, allocation and binding of FU resources and registers, the tool generates detailed reports on power dissipation and power efficient Micro Architecture Specifications (MAS). This supports the architecture designer, in writing a power-efficient RTL description or even making changes at algorithmic level to optimize power.

The right side of Figure 2 shows the used constellations to connect the tool chain. For synthesis, Cadence RTLCompiler and BuildGates were used together with enabled power optimization flags (e.g. "*set_attribute lp_insert_clock_gating true*"). Other compositions are possible, like e.g. executing HDL-simulations at register transfer level to save simulation time, but accepting lower accuracy. For the results in this document the netlist was simulated to receive signal activity information and store it in a file of the TCF format. HDL-simulations have been performed with Cadence NCsim. With the *report power* command the TCF information was annotated to the netlist to perform an activity aware power estimation of the design.

The tools of the standard commercial EDA flow are in principle replaceable by alternatives from other EDA companies [9-11].

3 Benchmark Design

The benchmark design evaluated in this paper is a 128-point FFT/iFFT processor for ultrawideband communication systems and was originally described in [8]. The proposed architecture, called mixed-radix multipath delay feedback (MRMDF), offers a higher throughput rate which is provided by using four parallel data paths. The MRMDF requires only minimum memory by using the delay feedback approach to reorder the input data and the intermediate results of each module. The benchmark was implemented in three ANSI-C functions or three RTL VHDL-components as shown in Figure 3.

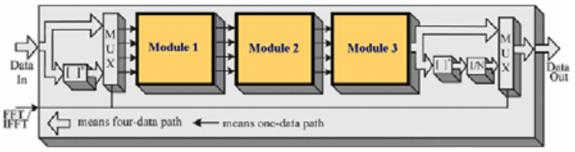

Figure 3: Benchmark FFT/iFFT design.

The first module implements a complex data register file, two complex multipliers, ROM to store so called twiddle factors and a butterfly unit (BU) that is conform to the radix-2 FFT algorithm (compare with Figure 4). The achievement of the proposed implementation is a 100 % usage of the two multipliers.

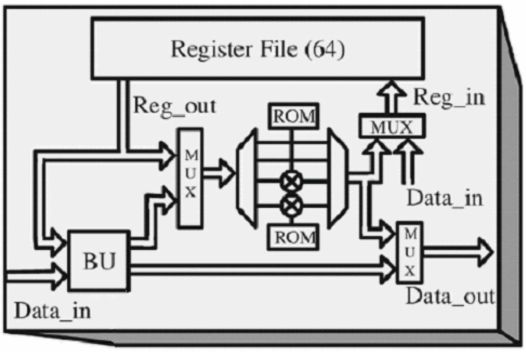

Figure 4: Internal architecture of module 1.

The second module consists of four BU_8 structures containing three delay elements each and one modified complex multiplier. The advantage of module 2 as depicted in Figure 5 is a saved gate count of about 38 %.

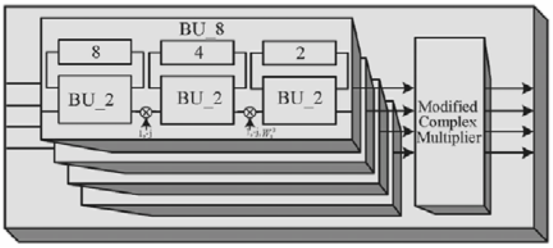

Figure 5: Internal architecture of module 2.

Module 3 is shown in Figure 6 and is the architectural simplest of the three components of the design. This is where the radix-8 FFT algorithm is realized with a suitable structure to ensure the correction of the FFT output data.

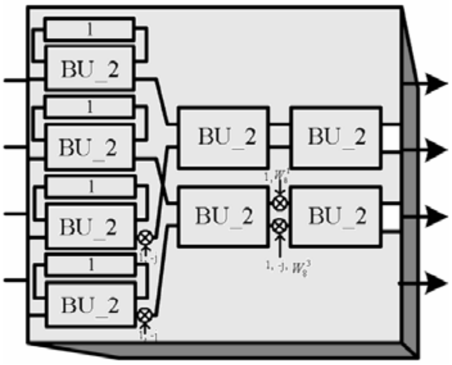

Figure 6: Internal architecture of module 3.

4 Discussion of Results

The results of this paper can be found in Table 1 and document a typical design-space exploration proceeding. The implementation of the semantically same FFT-design is changed during several iterations in a more or less trial and error approach. Each new alternative is analyzed and if considered good continued or discarded if not. The faster the iterations converge, the earlier the design processes finishes (improved time to market) or the more alternatives can be evaluated in the same time (broader design-space exploration). Additionally the trial and error process can be supported by early hints of what design alternatives will have most likely positive effects and do not lead to dead ends.

In Table 1 five design alternatives are listed. The first design “Shift Reg Variable Mult 17 bit” describes an architecture where the register file of module 1 is implemented as one shift-register. The multipliers of module 2 are regular variable types and the bit width of the data path is 17 bit. The architecture was implemented from scratch as an ANSI-C executable model. The effort for this task was documented with 40 person hours. The estimated power consumption of the C-model is 24.3 mW. Runtime of the estimation process was approx. one minute. In parallel, a regular design process was executed and a RTL VHDL design was implemented from the scratch requiring 101 person hours. The effort was 2.5 times higher compared to the C-model. The power estimation process takes another 45 to 60 minutes to complete and delivers 30.8 mW as result.

Design		Algo Level	Gate Level	Power Saving
Shift Reg Variable Mult (17 Bit 54 dB SNR)	**Power**	24.3 m W	30.8 mW	
	Aberration Algo - Gate	-21 %		
	Effort	40 h	101 h	
	Estimation Time	~1 min	45-60 min	+7 %
Shift Reg Constant Mult (17 Bit 54 dB SNR)	**Power**	23.9 m W	28.6 mW	
	Aberration Algo - Gate	-17 %		
	Effort	8h	36h	
	Estimation Time	~1 min	45-60 min	-3 %
Ring Buffer Constant Mult (17 Bit 54 dB SNR)	**Power**	23.3 m W	29.4 mW	
	Aberration Algo - Gate	-21 %		
	Effort	½ h	5 h	
	Estimation Time	~1 min	45-60 min	+24 %
Ring Buffer Clock Gate Constant Mult 17 bit	**Power**	21.3 m W	22.5 mW	
	Aberration Algo - Gate	-5 %		
	Effort	~1 min	11 h	
	Estimation Time	~1 min	45-60 min	+41 %
Ring Buffer Clock Gate Constant Mult 12 bit	**Power**	12 mW	13.3 mW	
	Aberration Algo - Gate	-10 %		
	Effort	¼ h	2 h	
	Estimation Time	~1 min	45-60 min	

Table 1: Estimated power dissipation at algorithmic and gate-level for different optimizations.

It is a reasonable assumption to expect that this estimate at gate-level is within 15-20 % accuracy if compared to real silicon. The previous estimate at higher algorithmic level is obviously less accurate but is underestimating by only 21 %.

The second design is a variance of the first one. The previous analyses at algorithmic level showed that the second module's multipliers of the benchmark were a main source of power dissipation. Please note that the values of Figure 7 are nWs (Energy) and not mW as in Table 1.

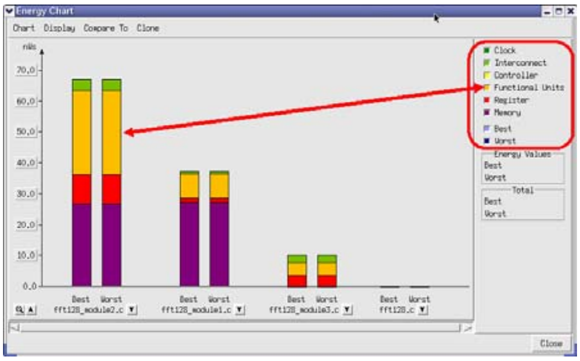

Figure 7: Functional units of module 2 disipate the most power (ORINOCO).

The variable multipliers were exchanged by constant multipliers and a power saving of 7 % was verified at gate-level. With 17 % the aberration is similar to the first experiment. The modification of the ANSI-C model required 8 h of work while the RTL VHDL consumed 36 h.

The shift register of module 1 was the target for the second modification. Within a shift register, in each cycle all values are constantly moving from flipflop to flipflop. We can assume that this requires more dynamic power than a ring buffer implementation would consume, where most of the values stay constant. Unfortunately, after modifying the C (½ h) and VHDL (5 h) we hardly detected an improvement at algorithmic level and the gate-level estimate even showed a 3 % degeneration of power consumption.

In the shift register architecture the flipflops are statically connected to the data path and its FUs. All values pass the appropriate register whenever they are used in a specific FU. This changed for the ring buffer architecture. It requires a structure of multiplexers to the FUs. This overhead apparently consumes too much power to notice a benefit.

The idea of using a ring buffer becomes a winner, when clock gating is applied (values of fourth design in Table 1). Clock gating disables registers' clock signals when the values are static for some time. This low power design methodology is not applicable for the shift register architecture since the values are constantly shifted and the clock can never be turned off. For the ring buffer, power is reduced by 24 %.

At the algorithmic level no effort was needed to receive this number. CG is considered by setting a simple flag in the estimation engine. Unluckily, at gate-level the effort was quite high with 11 h. Due to setup and hold violations induced by CG this extra effort had to be spent to achieve a correct gate-level simulation run. It is correct to assume that the violations disappear after place and route (P&R) and clock tree insertion. And it would improve the accuracy of the power estimate even more, but this additional step down the abstraction hierarchy would also dramatically increase the effort that has to be spent to get power estimation results. For a wide exploration of the design-space, we wanted to have all iterations as short as possible, though.

The last experiment included a reduction of the bit width of the data path from 17 bit down to 12 bit. For this purpose, the C model estimation flow supports pragmas to constrain the bit width of variables:

```
#pragma orinoco bitwidth 8
typedef int int8;
…
int8 my_var = 0;
```

The effort to adapt the concerned pragmas from 17 down to 12 bit was 15 minutes. Changing the bit width of signals in the RTL VHDL code required 2 h.

The introduced higher truncation error leads to a reduced signal to noise ratio (SNR). The effect for the benchmark design is discussed in [8]. The authors conclude that 12 bit are still more than sufficient. The improvement in power is 41 % leading to an overall improvement of 57 %.

5 Conclusions

In this paper we demonstrated that estimating power very early in the design flow is a feasible approach towards broadening the design-space exploration process, shortening time to market and designing integrated circuits with reduced power dissipation. In the documented example, a 128-point FFT/iFFT processor for ultrawide band communication systems, power dissipation could be improved by 57 %. The overall effort to receive power estimates was reduced from approximately 160 person hours down to 49 by a factor of three and more when compared to gate-level. Even though less accurate, the high level estimates where still within a very reasonable 5 to 21 % of the gate-level estimates.

From this numbers, we conclude that it is possible to make the right decision regarding power dissipation at algorithmic abstraction level and save significant engineering-time in the development process so that the available design-space can be explored more extensively.

References

[1] L. Kruse, E. Schmidt, G. Jochens, A. Stammermann, A. Schulz, E. Macii, and W. Nebel, Feb. 2001 “Estimation of Lower and Upper Bounds on the Power Consumption from Scheduled Data Flow Graphs” IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 9, No. 1

[2] L. Kruse, Okt. 2001 “Estimating and Optimizing Power Consumption of Integrated Macro Blocks at the Behavioral Level” Dissertation, University of Oldenburg, Computer Science, Oldenburg, Germany

[3] A. Stammermann, D. Helms, M. Schulte, A. Schulz, and W. Nebel, Nov. 2003 “Binding, Allocation and Floorplanning in Low Power High-Level Synthesis” Proc. ACM/IEEE Int. Conference on Computer Aided Design, San Jose

[4] E. Macii, M. Pedram, F. Somenzi, 1998 “High-Level Power Modeling, Estimation, and Optimization” IEEE Transactions on Computer-Aided Design, Vol. 17, No. 11

[5] P. Babighian, L. Benini, E. Macii, January 2005 “A Scalable Algorithm for RTL Insertion of Gated Clocks based on Observability Don't Cares Computation,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 24, No. 1, pp. 29-42

[6] Flavius Gruian, 2002 “Energy Centric Scheduling for Real-Time Systems” Dissertation, department of computer science, Lund University

[7] ChipVision, “ORINOCO-DALE 2007.1 User Guide”, www.chipvision.com

[8] Yu-Wei Lin, Hsuan-Yu Liu, Chen-Yi Lee, Aug. 2005 „A 1-GS/s FFT/iFFT Processor for UWB Applications,” IEEE Journal of Solid-State Circuits, Vol. 40, No. 8

[9] Frank Poppen, Wolfgang Nebel, 2001 “Comparison of a RT and Behavioral Level Design Entry Regarding Power,” SNUG Europe 2001

[10] Frank Poppen, Wolfgang Nebel, 2001 “Evaluation of a Behavioral Level Low Power Design Flow Based on a Design Case,” SNUG Boston 2001

[11] Frank Poppen, Milan Schulte, Wolfgang Nebel, 2006 „Power Optimised Digital Filterbank as Part of a Psychoacoustic Human Hearing Model,” SNUG Europe 2006

The Authors

Frank Poppen received his diploma in computer science from the University Oldenburg, Germany in the year 1999. Since then he is working for the OFFIS Research Institute in Oldenburg Germany. Until the year 2001 he was working with the OFFIS "Low-Power-Design-Methodology" group after which he changed to the newly founded "Design-Center" group of the department "Embedded Hardware-/Software-Systems". During his time with OFFIS, Frank was working within several national and European wide projects on power optimization for embedded systems, ASIC design and measuring of productivity of flows/tools/designers. He published papers at the European and Boston Synopsys User Groups in 2001 and 2006, is a member of the SNUG-Europe technical committee since 2002 and the SNUG-Europe user technical chairperson since 2007. Frank has experience with tools from Synopsys (PowerCompiler, DesignCompiler, Behavioral-Compiler, VCS, CoreConsultant), Cadence (FirstEncounter, SiliconEnsemble, NCsim, RTcompiler, BuildGates), Mentor (ModelSim), ChipVision (ORINOCO), CoWare, MathWorks and technologies from Amis, Artisan, ES2, LSI, Mietec, TSMC, UMC and XFAB.

Alexander Jährling received his diploma in computer science from the University of Oldenburg Germany in the year 2000. He started to work with the OFFIS "Low-Power-Design-Methodology" group the same year. He changed to the ChipVision Design Systems AG in Oldenburg Germany in year 2001 where he developed actively ORINOCO until 2003. He then changed into application engineering where he is involved until today.

Wolfgang Nebel holds a Dipl.-Ing. degree in Electrical Engineering from Hannover University, Germany, and a Dr.-Ing. degree from the Computer Science Department of Kaiserslautern University. In 1987 Prof. Dr. Wolfgang Nebel joined Philips Semiconductors, Hamburg, and worked as software engineer, CAD project manager and finally became CAD software development manager. In 1993 he became full university professor for VLSI design at the Computer Science Department of the University of Oldenburg and has served as Dean of the Computer Science Department and Vice-President Research there. He is the chairman of the OFFIS Research Institute. Prof. Dr. Wolfgang Nebel has been involved in several CAD conferences, e.g. as program chair of EURO-VHDL 94 and 95, EURO-DAC 96, PATMOS 96, DATE 2001 and general chair of PATMOS 95 and ISLPED 2006. He is active in several additional program committees and professional organizations including ACM, ECSI, EDAA, GI, IEEE, IFIP WG 10.5, VDE. His research interests are in methodologies and tools for embedded system design, in particular: object oriented HW/SW specification and synthesis, as well as design for low power. Prof. Dr.Wolfgang Nebel is co-founder, chairman and CTA of ChipVision Design Systems AG, an EDA start-up company located in Oldenburg, San Jose and Munich.