

Methoden zur applikationsspezifischen Verlustleistungsoptimierung für eingebettete Prozessoren

Sebastian Hesselbarth, Holger Blume

Leibniz Universität Hannover, Institut für Mikroelektronische Systeme, Appelstraße 4, 30167 Hannover

Email: {hesselbarth, blume}@ims.uni-hannover.de

Kurzfassung

Dieser Beitrag beschreibt eine Methodik zur Verlustleistungsmodellierung von eingebetteten Prozessoren im Entwurfsstadium auf Basis der Hardwarebeschreibung. Die Methodik wurde exemplarisch auf einen typischen RISC-Prozessor angewendet. Die gewonnenen Verlustleistungsmodelle zeigen eine geringe Abweichung hinsichtlich der mittleren Verlustleistungsaufnahme von unter 5% und eine hohe Güte bezüglich des zeitlichen Verlaufes der Verlustleistungsaufnahme im Vergleich zur sehr zeitaufwendigen Simulation der Gatter-Netzliste. Zudem lassen sich die Modelle zusammen mit der funktionalen Emulation des Prozessors auf einem FPGA abbilden. Die hohe Ausführungsgeschwindigkeit der Emulation erlaubt sowohl eine umfassende, verlustleistungsorientierte Optimierung der Anwendungen durch den Applikationsentwickler als auch eine anwendungsorientierte Optimierung der Prozessorarchitektur durch den Hardwareentwickler.

1. Einleitung

Die große Nachfrage nach leistungsfähigen, mobilen Endgeräten führt zunehmend zum Einsatz immer komplexerer, programmierbarer Prozessoren in solchen Systemen. Stetig sinkende Strukturgrößen moderner Fertigungstechnologien und die damit einhergehende Integration vieler, spezialisierter Prozessorkerne in heterogenen Multi-Core System-on-Chip erlaubt es zwar, der Forderung nach einer höheren Rechenleistung nachzukommen, der mobile, batteriegetriebene Einsatz erfordert jedoch eine zunehmende Beachtung der Verlustleistungsaufnahme. Insbesondere durch den zeit- und kostenintensiven Herstellungsprozess ist es wünschenswert, bereits frühzeitig aussagekräftige, anwendungsorientierte Verlustleistungswerte während der Entwicklungsphase zu gewinnen. Diese lassen sich bei Prozessoren im Entwurfsstadium nur über sehr zeitaufwendige Simulation der Schaltaktivität der technologieabhängigen Gatter-Netzliste bestimmen, so dass eine Bewertung der Verlustleistung der Zielanwendungen mit realen Eingangsdaten aufgrund der erforderlichen Rechenzeit unmöglich ist.

Für den Applikationsentwickler ergeben sich so keine Möglichkeiten, die Auswirkungen von verschiedenen Anwendungen, algorithmischen Varianten oder sogar einzelner Anweisungen im Programmcode auf die Verlustleistung des Prozessors zu beobachten. Diese können aber alle einen signifikanten Einfluss auf die resultierende Verlustleistung haben. Eine umfassende Optimierung der Applikation hinsichtlich einer geringen Verlustleistung findet deshalb frühestens nach der Realisierung des System-on-Chip statt, wobei aber ein großes Optimierungspotential verloren geht. Der zeitintensive Entwicklungsprozess behindert zudem eine Verkürzung der Innovationszyklen und erschwert es damit, sich nachhaltig in konkurrenzintensiven Märkten zu behaupten.

Dieser Beitrag beschreibt eine Methodik zur Verlustleistungsmodellierung von eingebetteten Prozessoren im Ent-

wurfsstadium auf Basis der Hardwarebeschreibung. Die ermittelten Verlustleistungsmodelle können zusammen mit der funktionalen Emulation auf einem FPGA abgebildet werden. Die Methodik erlaubt es so, eine applikationsbezogene Optimierung der Verlustleistung durchzuführen, ohne auf die sehr zeitaufwendige Simulation der Gatter-Netzliste zurückgreifen zu müssen.

In Abschnitt 2 wird zunächst der Stand der Forschung bezüglich der Verlustleistungsmodellierung von Prozessoren dargestellt. Abschnitt 3 beschreibt die in diesem Beitrag verwendete Hybrid Functional Level/Instruction Level Power Analysis (Hybrid FLPA/ILPA). Die Methodik zur Modellierung der Verlustleistung für Prozessoren im Entwurfsstadium wird in Abschnitt 4 detailliert präsentiert. Abschnitt 5 beschreibt die Anwendung der Verlustleistungsmodelle auf einem Emulationssystem und in Abschnitt 6 werden die Verlustleistungsmodelle hinsichtlich ihrer Genauigkeit evaluiert. Die Zusammenfassung des präsentierten Beitrages erfolgt in Abschnitt 7.

2. Stand der Forschung

Die in der Forschung untersuchten Modellierungsansätze zur Verlustleistungsbestimmung von Prozessoren nehmen überwiegend eine Abstraktion der Verlustleistungsbestimmung von der berechnungsintensiven Gatter-Ebene auf höhere Ebenen vor. Die Ansätze zur Modellierung der Verlustleistung basieren zum einen auf der Instruction Level Power Analysis (ILPA) [1], bei der anhand der ausgeführten Instruktionen die Gesamtverlustleistung einer Anwendung bestimmt wird, oder auf der Functional Level Power Analysis (FLPA) [2], bei der den Funktionseinheiten eines Prozessors eine bestimmte Verlustleistung zugeordnet wird. Ein weiterer vielversprechender Ansatz zur Modellierung der Verlustleistung ist die Hybrid FLPA/ILPA [3], bei der die Vorteile der FLPA und der ILPA in einem kombinierten Modell vereint werden.

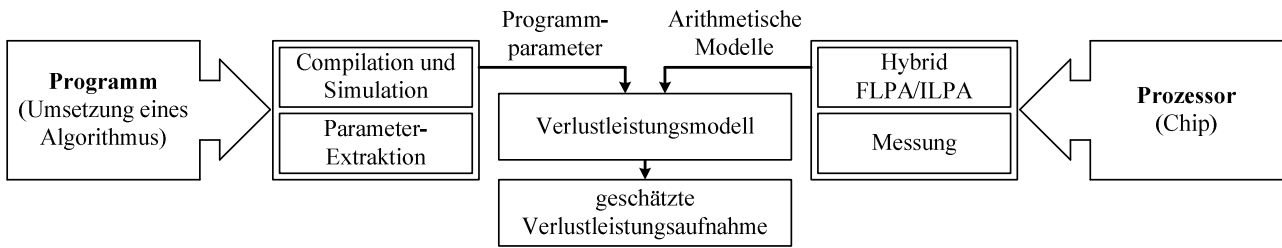


Bild 1 Schätzung der Verlustleistungsaufnahme nach dem Prinzip der Hybrid FLPA/ILPA

In der jüngeren Forschung werden die grundlegenden Konzepte der FLPA und der ILPA weiter untersucht und verbessert. So stellen Rethinagiri et al. [4] einen FLPA-Modellierungsansatz vor, der auf Basis einer virtuellen Plattform den Prozessor simuliert. Das Modell erreicht eine Genauigkeit für die mittlere Verlustleistung von ca. 3,5% gegenüber der Messung am Chip.

Park et. al. [5] präsentieren einen ILPA-basierten Ansatz, der neben der ausgeführten Instruktionsklasse auch die Hamming-Distanz zwischen zwei Eingangsoperanden der Instruktion als Einflussfaktor auf die Verlustleistung identifiziert. Verglichen mit der Verlustleistungssimulation der Gatter-Netzliste erreicht das Modell eine Genauigkeit der mittleren Verlustleistung von ca. 5%. Die Integration des Verlustleistungsmodells in einen Instruktionssatzsimulator erlaubt zudem eine Betrachtung des Verlustleistungsverlaufes bei der Ausführung von Programmsequenzen.

Die geringe Ausführungsgeschwindigkeit der Instruktionssatzsimulation und damit der simulationsbasierten Verlustleistungsmodelle war Ursache dafür, dass erste Maßnahmen zur grundlegenden Beschleunigung der Ausführung der Verlustleistungsmodelle untersucht wurden.

Bei der Entwicklung von Prozessoren hat sich die funktionale Emulation des Prozessors im Entwurfsstadium auf FPGAs als geeignetes Mittel zur Beschleunigung der Anwendungsentwicklung erwiesen (*HW/SW-Codesign*), so dass auch erste Forschungen zur Kombination der funktionalen Emulation mit der Berechnung von Verlustleistungsmodellen [6] durchgeführt wurden. Da sich die für die funktionale Emulation erforderliche Netzliste jedoch von der technologieabhängigen Gatter-Netzliste unterscheidet, muss auch hier eine Abstraktion der Verlustleistungsbestimmung von der Gatter-Ebene erfolgen.

So stellen Bhattacharjee et. al. [7] einen Ansatz zur Verlustleistungsemulation auf Basis von Ereigniszählern vor. Im Vergleich zur Gatter-Simulation erreichen die Modelle eine Abweichung von unter 10% bezogen auf die mittlere Verlustleistung, erfordern jedoch eine Modifikation der Applikation um die Ereigniszähler auszuwerten.

Einen weiteren Ansatz zur Emulation von Verlustleistungsmodellen stellen Bachmann et. al. [8] vor. Dieser basiert auf der automatisierten Extraktion von Signalen aus der Hardwarebeschreibung des Prozessorsystems mit einem signifikanten Einfluss auf die Verlustleistung. Die Extraktion der Signale erfolgt anhand üblicher Bestandteile der Signalnamen, eine weitere Einbeziehung der architektonischen Merkmale des Prozessors oder der ausgeführten Instruktionen erfolgt nicht. Bezogen auf die mittlere Ver-

lustleistung erreichen die Autoren eine Modellabweichung von unter 5% im Vergleich zur Gatter-Simulation.

3. Hybrid FLPA/ILPA

Bei den bisherigen Ansätzen zur Modellierung der Verlustleistungsbestimmung haben insbesondere die Modelle auf Basis der Hybrid Function Level/Instruction Level Power Analysis (Hybrid FLPA/ILPA) [3,9] eine gute Genauigkeit erreicht. Diesen Modellen liegt für die funktionale Ebene die Annahme zugrunde, dass sich für die Bestimmung der Verlustleistungsaufnahme ein Prozessor in eine begrenzte Anzahl von Funktionseinheiten unterteilen lässt, die jeweils unabhängig von anderen Einheiten zur gesamten Verlustleistungsaufnahme beitragen.

Die Analyse der Instruktionsebene erlaubt eine Verfeinerung der Modelle durch weitere Eingangsparameter, wie z.B. Art der Instruktion, die auf einer Einheit ausgeführt wird. Bei der Hybrid FLPA/ILPA werden die untersuchten Prozessoreinheiten gezielt durch geeignete, synthetische Programmsequenzen angeregt und die Verlustleistungsaufnahme des Prozessors am Chip gemessen. Mit den gewonnenen instruktionsbasierten Verlustleistungsmodellen kann dann durch Parameter-Extraktion aus der Compilation und Simulation (siehe Bild 1) eine Aussage über die zu erwartende Verlustleistungsaufnahme der Anwendung getroffen werden.

Die bisherigen Forschungen zur Hybrid FLPA/ILPA erfolgten nur an existierenden Prozessoren, d.h. nach Beendigung der Entwurfsphase des Prozessorkernes. Dieser Beitrag erweitert die Anwendung der Verlustleistungsmodelle auf Basis der Hybrid FLPA/ILPA auf Prozessoren im Entwurfsstadium und die Integration der Bestimmung der Verlustleistung in die funktionale Emulation.

4. Verlustleistungsmodellierung für Prozessoren im Entwurfsstadium

Die vorgestellte Methodik zur Verlustleistungsmodellierung verwendet exemplarisch mit dem *LEON-2 Sparc v8* Prozessor einen typischen RISC-Prozessor für eingebettete Systeme, wie z.B. in [10] zur Protokollverarbeitung von drahtlosen Funksystemen. Die Modellierung der Verlustleistung wird zunächst auf die Integer Unit (IU, Bild 2) beschränkt, da sich dort für den Anwendungsentwickler die größten Einflussmöglichkeiten ergeben und ca. 75% der Gesamtverlustleistung des Prozessorkernes auf diese Ein-

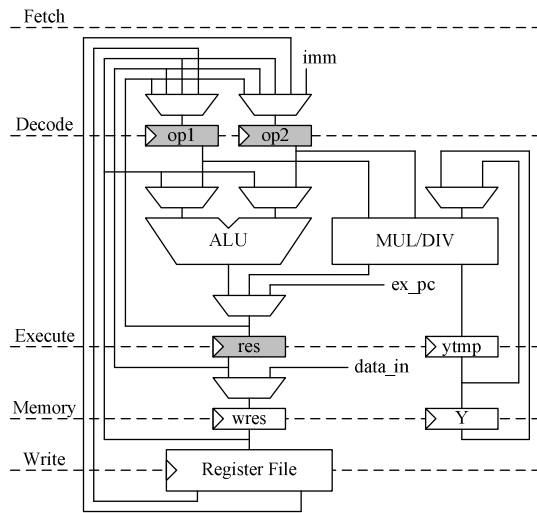


Bild 2 Schematischer Aufbau der Integer Unit (IU) des LEON-2 Sparc v8 RISC

heit entfallen. Grundsätzlich ist die Methodik jedoch auch auf andere Prozessorkerne und die verbleibenden Module des RISC-Prozessors übertragbar.

1.2. Identifikation von Einflussgrößen auf die Verlustleistung

Bei der Optimierung der Verlustleistung ist besonders die dynamische Verlustleistung

$$P_{dyn} = \alpha \cdot C \cdot V^2 \cdot f$$

mit der Schaltaktivität α , der elektrischen Kapazität der Schaltung C , der Versorgungsspannung V und der Taktfrequenz f relevant. Für ein gegebenes Prozessorsystem sind die Kapazität, Versorgungsspannung und Prozessorfrequenz weitgehend konstant, so dass einzig die Schaltaktivität α als variable Größe verbleibt. Die genaue Bestimmung der Schaltaktivität erfordert jedoch eine Betrachtung der Gatter-Ebene, so dass für eine Abstraktion auf höhere Ebenen andere, beobachtbare Größen identifiziert werden müssen.

Die Modellierung der Verlustleistung nach der Hybrid FLPA/ILPA basiert zunächst auf der Annahme, dass die Verlustleistung von der ausgeführten Instruktion abhängt, d.h. unterschiedliche Anweisungen führen zu einer unterschiedlichen Verlustleistungsaufnahme. Darüber hinaus weisen auch gleiche Anweisungen z.T. eine starke Abhängigkeit der Verlustleistung von den verarbeiteten Daten auf, so dass in diesem Beitrag die zeitliche Hamming-Distanz

$$HD(x, t) = \sum_i \begin{cases} 1 & \text{wenn } x_i(t) \neq x_i(t-1) \\ 0 & \text{sonst} \end{cases}$$

mit dem Bitvektor x der Länge i zum Zeitpunkt t als beobachtbares Maß für die Schaltaktivität verwendet wird. Der größte Teil der Verlustleistungsaufnahme der Integer Unit tritt innerhalb der Execution Stage (siehe Bild 2) in der Arithmetisch/ Logischen Einheit (ALU) und in der Multiplizierer-/ Dividierer-Einheit (MUL/DIV) auf.

Um das Ausmaß der Schaltaktivität innerhalb dieser Einheiten approximieren zu können, wird von den Eingangsoperanden $op1$ und $op2$ sowie von dem Ergebnis res der Operation jeweils die zeitliche Hamming-Distanz als Parameter gebildet und zur Modellierung der Verlustleistung verwendet.

1.3. Automatisierte Modellierung der Verlustleistung

Die Bestimmung der Verlustleistung eines Prozessors im Entwurfsstadium erfordert zunächst eine Simulation der Schaltaktivität auf Basis der technologieabhängigen Gatter-Netzliste. Die gewonnenen Schaltaktivitäten werden dann als Eingangsdaten für eine Simulation der Verlustleistung mit Synopsys PrimeTime PX verwendet. Diese Simulationen bilden zu diesem Zeitpunkt die einzige Möglichkeit eine Verlustleistung der Schaltung zu bestimmen, da noch keine Realisierung existiert, an der eine Messung der Verlustleistung vorgenommen werden könnte.

Aufgrund der sehr zeitaufwendigen Simulationen auf Gatter-Ebene können diese aber nur für sehr kurze Programmsequenzen mit vertretbarem Zeitaufwand durchgeführt werden. Um trotzdem eine Charakterisierung des Prozessorkernes bezüglich seiner Verlustleistungsaufnahme zu erreichen, werden für den Instruktionssatz des Prozessors sogenannte Micro-Benchmarks automatisiert generiert, in denen eine bestimmte Operation wiederholt ausgeführt wird. Dabei werden konstante Eingangsoperanden der Instruktion mit Zufallswerten gesetzt und auch von den Instruktionen verarbeitete Variablen mit zufälligen Werten initialisiert. Aufgrund der kurzen Micro-Benchmarks beträgt die Laufzeit einer Simulation mit der Auswertung der gewonnenen Daten nur wenige Minuten, während sonst selbst bei kurzen, vollständigen Programmen der Zeitaufwand für die Simulation mehrere Stunden bis Tage dauern kann. Zudem können die unabhängigen Benchmarks auch parallel ausgeführt werden, so dass der zusätzliche Aufwand für die Charakterisierung des gesamten Instruktionssatzes des Prozessors gering ist.

Aus den Simulationen der Schaltaktivität und der Verlust-

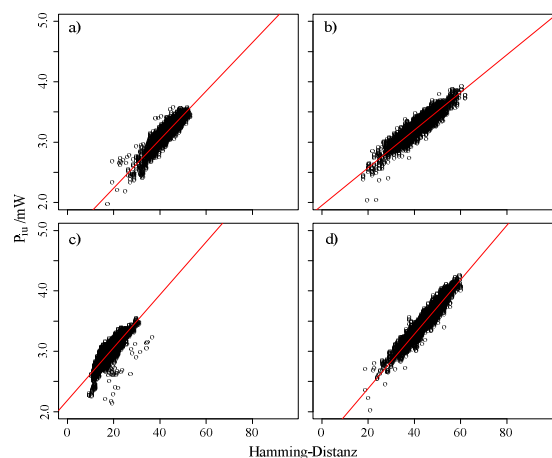


Bild 3 Instruktionsspezifische Verlustleistungsmodelle in Abhängigkeit der Hamming-Distanz a) ADD, b) OR, c) LOAD, d) XOR.

leistung der Schaltung für die generierten Programmsequenzen werden instruktionsspezifische Verlustleistungsmodelle mit einer linearen Regressionsanalyse ermittelt. Die Beschränkung auf die linear-additive Modellierung der Verlustleistungsaufnahme einer Instruktion I durch

$$\tilde{P}_{W,I}(t) = k_0 + \sum_i a_i \cdot HD(x_i, t)$$

mit den Modellparametern k_0 und a_i erfolgt dabei insbesondere mit Rücksicht auf eine spätere Implementierung der Modelle auf einem FPGA zusammen mit der funktionalen Emulation. Die Ergebnisse der Verlustleistungsmodellierung und der Approximation durch eine Regressionsgerade sind in Bild 3 für ausgewählte Instruktionen dargestellt. Die unterschiedlichen Modellkoeffizienten wirken sich dabei insbesondere auf die unterschiedliche Steigung der Regressionsgeraden für die Instruktion aus.

5. Emulation der Bestimmung der Verlustleistung

Für die Anwendung der Verlustleistungsmodelle innerhalb der funktionalen Emulation wird ein *BEE4* Emulationssystem der Firma *BEEcube* verwendet. Das Emulationssystem besteht aus insgesamt vier *Xilinx Virtex-6 LX550T* FPGAs mit jeweils 2x4G DDR3 RAM-Modulen. Die Kommunikation zur Initialisierung und Steuerung der emulierten Schaltung von einem Host-PC (siehe Bild 4) wird über Ethernet mit einem DMA-Modul gewährleistet. Die Anbindung des Host-PC erlaubt es, Anwendungen sowie deren Eingangsdaten in den Speicher der eingesetzten *LEON-2* CPU zu schreiben. Außerdem werden die Ergebnisse des Hardware-Moduls zur Bestimmung der Verlustleistung zur Visualisierung auf den Host-PC zurückgeladen.

1.4. Hardware-Modul zur Bestimmung der Verlustleistung

Das Hardware-Modul zur Bestimmung der Verlustleistung übernimmt die Anwendung der instruktionsspezifischen Verlustleistungsmodelle unter Beobachtung der internen Signale der Integer Unit des *LEON-2*. Die Hardware-Beschreibung des *LEON-2* wurde so angepasst, dass diese Signale von der Integer Unit in das Hardware-Modul zur Bestimmung der Verlustleistung geführt werden können. Der geringfügige Aufwand für die Anpassung beschränkt sich dabei auf die Identifikation der relevanten Signale und das Durchführen der Signale auf die oberste Hierarchieebene. Jegliche Verarbeitung dieser Signale, wie z.B. die Berechnung der Hamming-Distanz, findet innerhalb des Hardware-Moduls statt, so dass der Einfluss auf die Geschwindigkeit der funktionalen Emulation gering bleibt. Die Anwendung der Verlustleistungsmodelle basiert auf einem Speicher für die Modellkoeffizienten k_0 und a_i der jeweiligen instruktionsspezifischen Regressionsgeraden. Mit Hilfe der aktuell auf dem Prozessor ausgeführten Instruktion, der Hamming-Distanz der Modellgrößen und der instruktionsspezifischen Modellkoeffizienten wird dann ein Schätzwert für die aktuelle Verlustleistung berechnet.

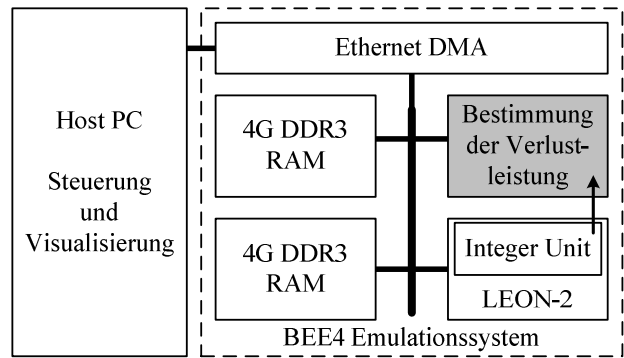


Bild 4 Integration der Anwendung der Verlustleistungsmodelle in die funktionale Emulation

Nach einer Mittelwertbildung über 64 Taktzyklen des Prozessors werden diese Schätzwerte in einen von der funktionalen Emulation unabhängigen Speicher geschrieben und können dort von dem Host-PC gelesen werden. Die Implementierung des Hardware-Moduls auf dem verwendeten Emulationssystem erfordert einen zusätzlichen Ressourcenbedarf an LUTs des FPGAs von 11,6% im Vergleich zum Prozessorkern bzw. umfasst 1,4% des Gesamtsystems aus Prozessor, Speichercontrollern und DMA-Modul (siehe Bild 4). Für die Speicherung der Modellkoeffizienten werden zusätzlich 2,5% der verfügbaren BlockRAMs des FPGAs benötigt. Auf die maximal erreichbare Taktfrequenz der funktionalen Emulation von ca. 100 MHz hat das Hardware-Modul zur Bestimmung der Verlustleistung keinen Einfluss.

6. Evaluation der Verlustleistungsmodellierung

Die Genauigkeit der präsentierten Verlustleistungsmodelle wurde anhand typischer Algorithmen für eingebettete Systeme untersucht. Dabei wurden neben unterschiedlichen Anwendungen auch algorithmische Varianten und unterschiedliche Einstellungen zur Compiler-Optimierung ausgewertet. Die Ergebnisse der Untersuchungen zur Genauigkeit sind in Tabelle 1 zusammengefasst.

Tabelle 1 Genauigkeit der Verlustleistungsmodelle für verschiedene Benchmark-Programme, Programmvarianten (*A/B*) und Compiler-Optimierungen (*O3/Os*)

Benchmark	Zyklen	%MAE	NRMSE
math (<i>O3</i>)	625.280	4,33%	8,81%
math (<i>Os</i>)	643.840	4,36%	8,08%
lowpass_3x3 (<i>A, O3</i>)	1.700.480	4,36%	6,75%
lowpass_3x3 (<i>A, Os</i>)	2.063.360	3,92%	10,30%
lowpass_3x3 (<i>B, O3</i>)	1.547.520	3,82%	9,50%
lowpass_3x3 (<i>B, Os</i>)	2.602.880	2,84%	6,12%
median_3x3 (<i>A, O3</i>)	8.657.920	4,97%	9,90%
median_3x3 (<i>A, Os</i>)	9.173.760	4,88%	10,07%
median_3x3 (<i>B, O3</i>)	8.812.800	3,71%	7,11%
median_3x3 (<i>B, Os</i>)	9.700.480	4,00%	6,39%
quicksort (<i>O3</i>)	9.111.040	2,15%	4,68%
quicksort (<i>Os</i>)	9.401.600	3,01%	6,89%

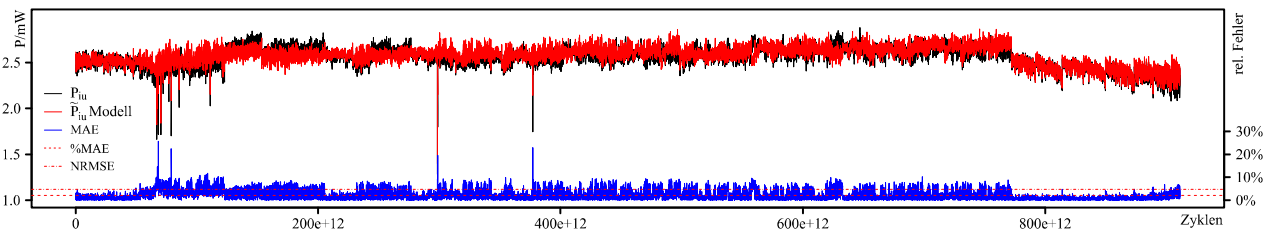


Bild 5 Vergleich des Verlustleistungsmodells mit der Verlustleistungssimulation für den quicksort-Benchmark

Die mittlere, absolute Abweichung der Verlustleistungsschätzung in Relation zur aktuellen Verlustleistung (*Percentage Mean Absolute Error, %MAE*) aus der Simulation beträgt dabei 2,15% bis 4,97%. Für eine Anwendung der Verlustleistungsmodelle durch den Software-Entwickler zur applikationsspezifischen Optimierung hinsichtlich der Verlustleistungsaufnahme ist aber neben der mittleren absoluten Abweichung des Modells auch der zeitliche Verlauf der Verlustleistung relevant. Als Maß für die Ähnlichkeit der Verläufe kann die auf die Dynamik der Beobachtung normierte Wurzel der mittleren, quadratischen Abweichung (*Normalized Root Mean Square Error, NRMSE*) herangezogen werden und liegt für die Verlustleistungsmodelle bei 4,68% bis 10,30%. Die gute Genauigkeit der Modelle über den Verlauf der Ausführung der Anwendung ist exemplarisch für den Benchmark *quicksort* in Bild 5 dargestellt.

7. Zusammenfassung

Dieser Beitrag stellt eine Methodik zur Verlustleistungsmodellierung von eingebetteten Prozessoren im Entwurfsstadium vor. Die Modelle basieren auf dem Ansatz der Hybrid FLPA/ILPA, wobei die Ableitung der instruktions-spezifischen Verlustleistungsmodelle automatisiert durch die Simulation der Schaltaktivität und Verlustleistung von *Micro-Benchmarks* erfolgt.

Die Einfachheit der erstellten Modelle erlaubt eine Abbildung der Bestimmung der Verlustleistung in einem FPGA als Ergänzung zur funktionalen Emulation ohne signifikanten Einfluss auf die Emulationsgeschwindigkeit von 100 MHz oder den Bedarf an Ressourcen des FPGAs. Die Güte der Verlustleistungsmodelle wurde anhand typischer Benchmark-Programme untersucht und liegt maximal bei ca. 5% *MAE* bzw. ca. 10% *NRMSE*.

Durch die Emulation der Bestimmung der Verlustleistung können die sehr zeitaufwendigen Simulationen auf Gatter-Ebene mit einer Ausführungsgeschwindigkeit von wenigen hundert Hertz für die späteren Zielanwendungen des Prozessors vermieden werden und der Software-Entwickler erhält so die Möglichkeit eine verlustleistungsorientierte Optimierung der Applikation bereits während der Entwurfsphase des Prozessors durchzuführen. Die daraus gewonnenen Hinweise auf die Anforderungen der Anwendungen erlauben wiederum eine applikationsorientierte Optimierung der Prozessorbeschreibung und können so zu einer umfassenden Verbesserung der Ausnutzung des Optimierungspotentials führen.

8. Literatur

- [1] Tiwari, V.; Malik, S.; Wolfe, A.: Power analysis of embedded software: a first step towards software power minimization. *IEEE Transactions on VLSI Systems*, Vol. 2, No. 4, Dec. 1994, pp. 437-445.
- [2] Laurent, J.; Julien, N.; Senn, E.; Martin, E.: Functional level power analysis: An efficient approach for modeling the power consumption of complex processors. *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, 2004, pp. 666-667.
- [3] Blume, H.; Becker, D.; Rotenberg, L.; Botteck, M.; Brakensiek, J.; Noll, T.G.: Hybrid functional- and instruction-level power modeling for embedded heterogeneous processor architectures. *Journal of Systems Architecture*, Vol. 53, Issue 10, 2007, pp. 698-702.
- [4] Rethinagiri, S.; Atitallah, R.; Dekeyser, J.-L.; Niar, S.; Senn, E.: An Efficient Power Estimation Methodology for Complex RISC Processor-based Platforms, *Proceedings of GLSVLSI'12*, 2012, pp. 239-244.
- [5] Park, Y.-H.; Pasricha, S.; Kurdahi, F.J.; Dutt, N.: A Multi-Granularity Power Modeling Methodology for Embedded Processors, *IEEE Transactions on VLSI Systems*, Vol. 19, No. 4, 2011, pp. 668-681.
- [6] Coburn, J.; Ravi, S.; Raghunathan, A.: Power emulation: a new paradigm for power estimation, *Proceedings of 42nd Design Automation Conference*, 2005, pp. 700-705.
- [7] Bhattacharjee, A.; Contreras, G.; Martonosi, M.: Full-system chip multiprocessor power evaluations using FPGA-based emulation, *Low Power Electronics and Design (ISLPED)*, 2008, pp.335-340.
- [8] Bachmann, C.; Genser, A.; Steger, C.; Weiss, R.; Haid, J.: Automated Power Characterization for Run-Time Power Emulation of SoC Designs, *Digital System Design: Architectures, Methods and Tools (DSD)*, 2010, pp. 587-594.
- [9] Blume, H.; von Livonius, J.; Rotenberg, L.; Noll, T.G.; Bothe, H.; Brakensiek, J.: OpenMP-based parallelization on an MPCore multiprocessor platform – A performance and power analysis, *Journal of Systems Architecture*, Vol. 54, No. 11, 2008, pp. 1019-1029.
- [10] Stamenkovic, Z.; Wolf, C.; Schoof, G.; Gaisler, J.: LEON-2: General Purpose Processor for a Wireless Engine, *Design and Diagnostics of Electronic Circuits and Systems*, 2006, pp. 48-51.