

Adaptive Buskodierverfahren zur verlustleistungseffizienten Datenübertragung auf Bussen digitaler Systeme

C. Kretzschmar, R. Siegmund, and D. Müller

Technische Universität Chemnitz, Fakultät für Elektrotechnik und Informationstechnik, Reichenhainer Straße 70, 09126 Chemnitz, Germany

Zusammenfassung. In diesem Beitrag wird ein Überblick über transitionsmindernde Buskodierverfahren zur verlustleistungsminimierten Datenübertragung zwischen Modulen digitaler Systeme gegeben. Ein neues adaptives Verfahren, das Adaptive Minimum Weight Codes (AMWC) Verfahren wird vorgestellt, welches Datenworte auf einen Code mit minimalem Gewicht unter Nutzung dynamisch rekonfigurierbarer Codetabellen abbildet. Aufgrund der Anpassbarkeit der Kodiervorschrift benötigt es keine Kenntnis statistischer Parameter der zu übertragenden Datenströme und eignet sich deshalb insbesondere zur Kodierung von Datenströmen mit über der Zeit veränderlichen Parametern. Im Gegensatz zu anderen bislang publizierten adaptiven Verfahren, welche aufgrund ihres enormen Eigenverlustleistungsanteiles nicht effizient implementierbar sind, besitzt die Implementierung von AMWC einen geringeren Hardwareoverhead und verändert das Systembusinterface nicht. Der Beitrag stellt theoretische Grundlagen vor und gibt eine hardwareeffiziente Implementierung an. Experimentelle Untersuchungen ergaben eine Reduktion der Schaltaktivität um 38%.

In this paper state-of-the-art transition-minimizing bus encoding schemes for power-efficient data transmission between modules of digital systems are summarized. A new adaptive scheme, the Adaptive Minimum Weight Codes (AMWC) is presented which maps data words unambiguously on a minimum weight code using dynamically reconfigurable code tables. Due to its adaptability our scheme does not require a priori knowledge about statistical parameters of data streams to be transmitted. Therefore it is especially suited for data streams with time-varying parameters. Unlike other adaptive techniques presented to date, which are infeasible to implement into hardware due to their tremendous overhead in self dissipated power, the implementation of our encoding technique requires less hardware overhead and does not modify the system bus

interface. The fundamentals of the encoding scheme and a hardware-efficient implementation are given. Experimental results showed a reduction in bus transition activity of up to 38%.

1 Einleitung

Die Verlustleistungsaufnahme digitaler informationsverarbeitender Systeme entwickelt sich zunehmend zum limitierenden Faktor bezüglich der weiteren Erhöhung der Performanz. Vor allem für portable, batteriegespeiste Geräte wie Laptops und Mobilfunkgeräte, deren Betriebsdauer direkt durch deren Verlustleistungsaufnahme bestimmt wird, ist eine Reduktion der umgesetzten Verlustleistung extrem wichtig. Als weiterer Aspekt steigt die Zuverlässigkeit verteilter, auf einem Chip integrierter Informationsverarbeitungssysteme höchster Leistung, bei denen, bedingt durch Anzahl und Packungsdichte der Transistoren, die Abführung der erzeugten Wärmemenge problematisch wird.

Mit bis zu 80% (Stan and Burleson, 1995) entfällt ein hoher Prozentsatz der in einem Chip umgesetzten Verlustleistung auf Systembusse, da die parasitären Leitungskapazitäten interne Kapazitäten um ein Vielfaches übersteigen. Die auf einem Systembus eines CMOS-Schaltkreises umgesetzte Verlustleistung kann in guter Näherung durch die nachfolgende Gleichung berechnet werden: $P_V = \frac{1}{2} V_{dd}^2 f \sum_{i=0}^{n-1} C_{L_i} \alpha_i$ wobei n die Busbreite, f die Taktfrequenz, V_{dd} die Versorgungsspannung, C_{L_i} die Kapazität und α_i die Schaltaktivität der Busleitung i repräsentieren. Während die drei erstgenannten Größen im allgemeinen von der Technologie und den Anforderungen an die Performanz bestimmt werden, läßt sich die Schaltaktivität auf Systemebene durch den Einsatz geeigneter transitionsmindernder Kodierverfahren beeinflussen.

In diesem Beitrag werden neue, im Rahmen dieses Projektes entstandene adaptive Kodierverfahren vorgestellt, die 2 Ansätze verfolgen: eine leitungs-basierte sowie eine wortbasierte Kodierung. Das *Adaptive Partielle Businvert Kodier-*

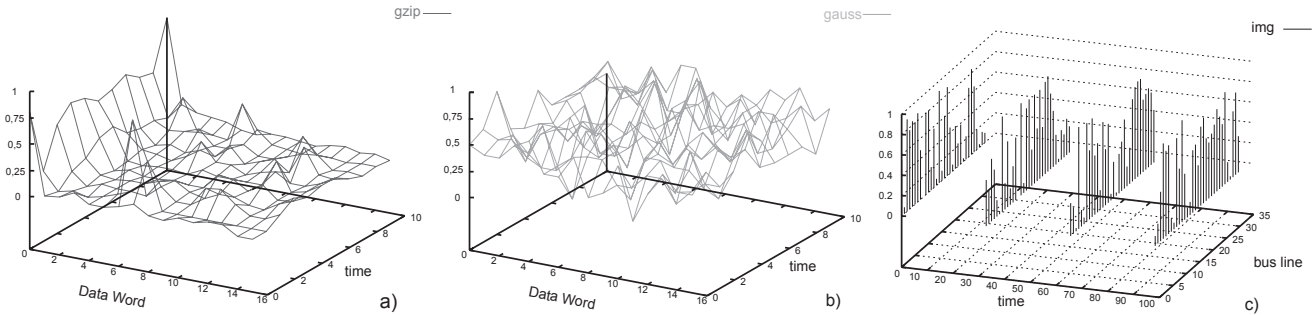


Abbildung 1. Statistische Parameter verschiedener Datenströme: (a) und (b) Wahrscheinlichkeitsdichte, (c) Schaltaktivitätsprofil.

verfahren (APBI), ein leitungsorientiertes arbeitendes Verfahren minimiert die Hammingdistanz zweier aufeinanderfolgender Codewörter, indem es die Leitungen mit der höchsten Transitionswahrscheinlichkeit selektiert und den entstehenden Teilbus Businvert-kodiert. Im Unterschied dazu verfolgen das *Adaptive Wahrscheinlichkeitsbasierte Mapping (APBM)* und das *Adaptive Minimum Weight Codes Verfahren (AMWC)* eine wortbasierte Kodierung. Sie bilden Datenwörter wahrscheinlichkeitsbasiert auf Codewörter ab. Durch die Nutzung von zur Laufzeit rekonfigurierbaren Codetabellen eignen sie sich speziell für eine verlustleistungsoptimierte Übertragung auf Systembussen rekonfigurierbarer Hardwarearchitekturen. Da die genannten Verfahren die für sie relevanten statistischen Parameter des zu übertragenden Datenstromes zur Laufzeit extrahieren, benötigen sie im Gegensatz zu statischen Verfahren vorab keine Kenntnis statistischer Parameter zur Optimierung der Kodiervorschrift. Deshalb eignen sie sich vor allem zur Übertragung von a-priori unbekanntem Datenströmen oder solchen mit nicht stationären statistischen Parametern. Simulative Untersuchungen ergaben für solche Datenströme eine Reduktion der Schaltaktivität um 42% für APBI, um 41% für APBM und um 38% für AMWC.

2 Bisherige Arbeiten und Motivation

Aus informationstheoretischer Sicht lassen sich Buskodierverfahren danach klassifizieren, ob sie zur effizienten Minimierung der Transitionen vorab Kenntnis statistischer Parameter der zu übertragenden Datenströme benötigen oder ohne diese Kenntnis auskommen. Die Verfahren der ersten Klasse gehören dabei nahezu ausschließlich zu den statisch arbeitenden Verfahren, d.h. die Kodiervorschrift wird vorab an einen als applikationsspezifisch betrachteten Datenstrom angepaßt und statisch in Coder- und Decoderhardware implementiert. In der Literatur wurden verschiedene zur ersten Klasse gehörende, applikationsspezifische Verfahren zur Systembus-Kodierung publiziert, wobei die einzelnen Verfahren unterschiedliche charakteristische Eigenschaften des zu übertragenden Datenstromes ausnutzen. Die T0-Kodierung (Benini et al., 1997), geeignet für Adreßbuse, überträgt aufeinanderfolgende Adressen durch das Set-

zen eines Inkrement-Bits auf einer zusätzlichen Leitung unter Beibehalten des ursprünglichen Buszustandes. Kombinationsverfahren in Benini et al. (1998) bieten eine angepasste Kodierung für mehrere über einen Bus übertragene Datenströme mit unterschiedlichen statistischen Eigenschaften. Die Businvert-Kodierung (Stan and Burlinson, 1995) ist ein 1bit redundantes Kodierverfahren, das die Hammingdistanz zweier aufeinanderfolgender Codewörter durch eventuelle Invertierung minimiert. Bei einer Ungleichverteilung der Schaltaktivität schließt das Partielle Businvert-Kodierverfahren (PBI) (Shin et al., 1998) Leitungen mit geringerer Schaltwahrscheinlichkeit von der Kodierung aus. Das Wahrscheinlichkeitsbasierte Mapping (PBM), das in Ramprasad et al. (1999) zusammen mit werte- und differenzbasierenden Kodierverfahren vorgestellt wurde, berechnet aus der Wahrscheinlichkeitsverteilungsfunktion (pdf) eine eindeutige Abbildung von Datenwörtern auf Codewörter. Ein weiteres Verfahren, das die temporale Korrelation aufeinanderfolgender Datenwörter in Form der Verbundwahrscheinlichkeitsverteilung (jpdf) berücksichtigt, wurde in Benini et al. (1999) vorgestellt. Die Komplexität der notwendigen Codetabelle wird dabei auf $2^{2n} \times n$ im Gegensatz zu $2^n \times n$ für PBM erhöht, was für reale Busbreiten nicht mehr implementierbar ist.

Die bisher genannten statischen Kodierverfahren, die für einen charakteristischen Datenstrom optimiert werden, erreichen dann eine sehr hohe Kodiereffizienz, wenn sie auf Datenströme mit identischen statistischen Parametern angewendet werden. Andernfalls sinkt ihre Kodiereffizienz drastisch. Wie unsere Untersuchungen zeigen, weisen reale Applikationen jedoch nicht stationäre Parameter auf, wie Abb. 1 am Beispiel der Wahrscheinlichkeitsdichte zweier Datenströme und eines Schaltaktivitätsprofils zeigt. Oft steht auch vorab kein solcher Datenstrom zur Verfügung oder das gesamte Applikationsspektrum ist im Vorfeld noch nicht bekannt. Eine effiziente Kodierung erfordert deshalb *adaptive* Kodierverfahren, die relevante statistische Parameter zur Laufzeit extrahieren und die Kodiervorschrift periodisch aktualisieren. Adaptive Ansätze werden in Benini et al. (1999) und Satoshi Komatsu et al. (2000) publiziert. Durch ihre Anpassungsfähigkeit an gegenwärtige statistische Parameter erreichen adaptive Verfahren eine effizientere Transitionsminde-

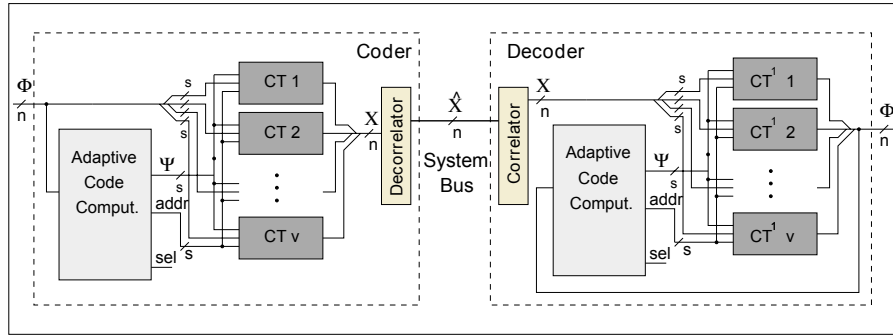


Abbildung 2. Blockschaltbild eines AMWC Coder-Decoder Systems.

zung, die jedoch durch einen erhöhten Hardwareoverhead erkauft wird, so daß für die bisher erwähnten Verfahren der Eigenverlustleistungsanteil die effiziente Transitionsminderung relativiert. Deshalb entstanden im Rahmen dieses Projektes mit APBI, APBM und AMWC Methoden, die sowohl an veränderte Parameter anpassungsfähig als auch verlustleistungsoptimiert implementiert sind. APBI (Kretzschmar et al., 2000) erweitert PBI um eine adaptive Komponente, die periodisch Leitungen eines zu kodierenden Subbusses selektiert. Da Schaltungen vor allem mobiler Systeme immer häufiger auf rekonfigurierbare Hardwarearchitekturen abgebildet werden, wurden mit APBM und AMWC Verfahren entwickelt, die speziell in solchen Systemen effizient die Verlustleistung reduzieren, jedoch nicht ausschließlich auf diese Hardware beschränkt sind. APBM und AMWC bilden Datenworte mit dem Ziel der Minimierung des Gewichtes des kodierten Datenstromes auf Codeworte ab. Beide implementieren zur Laufzeit dynamisch rekonfigurierbare Codetabellen, die mit einem an die gegenwärtige Wahrscheinlichkeitsdichte angepaßten Code periodisch rekonfiguriert werden.

3 Adaptive Minimum Weight Codes (AMWC)

3.1 Überblick

AMWC nutzt die in Kretzschmar et al. (2001) von den Autoren publizierten Grundlagen des Adaptiven Wahrscheinlichkeitsbasierten Mapping (APBM), das eine sehr hohe Kodiereffizienz erreicht. Es wurde jedoch für einen minimalen Hardwareoverhead optimiert, welcher sich annähernd proportional zur Eigenverlustleistung verhält. AMWC ist ein nicht-redundanz erhöhendes Kodierverfahren, das Datenworte ϕ eines nichtkodierten Datenstromes Φ eindeutig auf Codeworte χ eines Codes Ψ abbildet. Das Blockschaltbild eines Coder-Decoder-Systemes ist in Abb. 2 dargestellt.

Der Coder implementiert dynamisch rekonfigurierbare Codetabellen, die die Abbildungsvorschrift von Datenworten auf einen Code Ψ mit minimalem Gewicht enthalten. Zur Reduktion der Komplexität der benötigten Codetabellen erfolgt eine Aufteilung des n Bit breiten Systembusses in v Subbusse mit einer Breite von s Bit, die unter Reduktion der spatio-temporalen Korrelation getrennt voneinander kodiert werden. Ein

in den Coder integrierter Dekorrelator wandelt die Einsen im Codewort in eine Transition auf dem Bus, während eine Null im Codewort in eine Nicht-Transition übersetzt wird. Der im Decoder vorhandene Korrelator führt eine Rücktransformation der über den Bus übertragenen Transitionen in Einsen sowie der Nicht-Transitionen in Nullen durch, wodurch das ursprüngliche Codewort rückgewonnen und durch Anwendung der inversen Kodiervorschrift Ψ^{-1} in den Codetabellen des Decoders dekodiert wird. Durch den Einsatz des Dekorrelator/Korrelator-Paares wird das Problem der Minderung von Transitionen auf dem Bus auf eine Minimierung des Gewichtes, d.h. der Anzahl an Einsen, des kodierten Datenstromes X abgebildet, so daß ein transitionsmindernder Code folgende Eigenschaft besitzt:

$$\bar{\alpha}_{\hat{\chi}} = \sum_{\chi \in X} W(\chi) \cdot p_{\phi} \rightarrow \min \quad (1)$$

wobei die mittlere Schaltaktivität auf dem Bus $\bar{\alpha}_{\hat{\chi}}$ dann ein Minimum erreicht, wenn das Gewicht W über alle Codeworte χ gewichtet mit der Wahrscheinlichkeit ihres Auftretens ein Minimum erreicht. Für den Code Ψ bedeutet dies, daß häufig auftretende Datenworte auf Codeworte mit einer geringen Anzahl an Einsen abgebildet werden, während seltener zu übertragende Datenworte durch Codeworte mit einer höheren Anzahl an Einsen repräsentiert werden.

Da die Wahrscheinlichkeitsdichte der Datenworte vorab unbekannt oder nicht-stationär sein kann, führt der Codeberechnungsblock eine periodische Anpassung des Codes an die gegenwärtige Verteilung durch, mit dem die Codetabellen rekonfiguriert werden. Der Codeberechnungsblock wurde analog in Coder und Decoder integriert, um die Notwendigkeit der Übertragung des verwendeten Codes zwischen Coder und Decoder zu eliminieren. Unter Nutzung eines identischen initialen Codes und unter der Voraussetzung einer fehlerfreien Übertragung kann die neue Kodiervorschrift im Decoder aus den dekodierten Daten in analoger Weise zum Coder erfolgen.

3.2 Codeberechnung

Der Codeberechnungsblock führt eine Fensterung des Datenstromes durch, innerhalb dessen er die Wahrscheinlichkeitsdichte von Datenworten des unkodierten Datenstromes

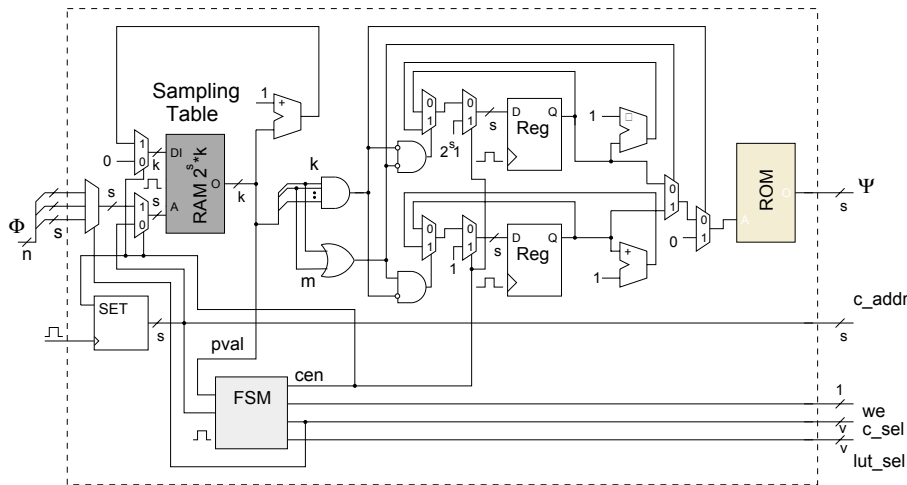


Abbildung 3. Blockschaltbild eines AMWC-Codeberechnungsblocks.

beobachtet. Im Anschluss berechnet er einen neuen, eindeutigen Code mit minimalem Gewicht, mit dem die Codetabellen rekonfiguriert werden. Im Gegensatz zu APBM wurde der Codeberechnungsblock, dessen Blockschaltbild in Abb. 3 abgebildet ist, in einfacher Ausführung realisiert. Dadurch erfolgen die Ermittlung der Statistik und die Aktualisierung der Codetabellen der einzelnen Subbusse sequentiell.

Der am Eingang befindliche Selektor wählt einen der v Subbusse zur Beobachtung aus. Die auftretenden Datenworte werden in der Samplingtabelle gezählt, um sie entsprechend ihrer Häufigkeit den im ROM befindlichen, nach ihrem Gewicht sortierten Codeworten zuzuordnen. AMWC führt im Codeberechnungsblock als neuen Ansatz, der bisher noch nicht angewendet wurde, eine *flexible Fensterbreite* ein. Das Beobachtungsfenster variiert entsprechend der Wahrscheinlichkeitsdichte der beobachteten Datenworte. Dieser Ansatz erlaubt eine besonders schnelle Anpassung der Kodiervorschrift für ungleich verteilte Wahrscheinlichkeiten der Datenworte. Die Realisierung erfolgt, indem die Zählerstände der k Bit tiefen Samplingtabelle auf ihren Maximalwert $2^k - 1$ getestet werden. Erreicht einer der Zähler diesen Wert, wird die Beobachtungsperiode abgeschlossen und die Codeberechnung beginnt. Die Zuordnung von Datenworten zu Codeworten erfordert normalerweise eine zeit- und ressourcenaufwendige Sortierung der in der Samplingtabelle erfaßten Häufigkeiten. Zur Vereinfachung wurden Codeworte in 2 Klassen unterteilt: günstige und weniger günstige Codeworte, wobei günstige Codeworte eine minimale Anzahl an Einsen besitzen, die nur wenige Transitionen auf dem Bus verursachen, während weniger günstige Codeworte eine grössere Anzahl an Transitionen verursachen. Durch Auswertung der oberen m Bit der Zählerstände kann sofort eine Entscheidung über die zuzuordnende Klasse von Codeworten getroffen werden. Eine Eins innerhalb der oberen m Bits bedeutet dabei, dass das zugehörige Datenwort mindestens 2^m mal innerhalb der Beobachtungsperiode aufgetreten ist und deshalb auf ein Codewort mit einer geringen Anzahl an Einsen abgebildet werden sollte. Andernfalls sollte ihm ein

Codewort der anderen Klasse zugeordnet werden. Da sich die Codeworte sortiert nach ihrem Gewicht in der ROM-Tabelle befinden, kann auf eine feste Einteilung der Codeworte in die beiden Klassen verzichtet werden. Vielmehr beginnt die Auswahl zuzuordnender Codeworte mit dem günstigsten bzw. dem ungünstigsten Codewort und nähert sich mit jedem zugeordneten Codewort an, so daß als letztes Codewort eines aus dem mittleren Bereich der ROM-Tabelle zugeordnet wird. Diese Vorgehensweise erlaubt die Anwendung des aktualisierten Codes bereits nach $s + d$ Takten, wobei d eine mögliche Verzögerung beim Eintrag in die Codetabelle des Sub(de)coders darstellt. Im Gegensatz zu APBM entsteht kein exakter Code für die gegenwärtige Wahrscheinlichkeitsdichte, jedoch steht dieser Code bereits nach einem Drittel der Zeit zur Verfügung, wodurch auftretende Ungenauigkeiten teilweise kompensiert werden.

4 Ergebnisse

Die beschriebenen Kodierv Verfahren wurden als synthetisierbare VHDL-Modelle für eine Busbreite von 32 bit implementiert und simulative Untersuchungen unter Nutzung folgender Testdatenstrome durchgeführt:

- **art**: Ein zufälliger Datenstrom, generiert mit Mathematica®, mit einem sich über der Zeit ändernden Schaltaktivitätsprofil (**art1**) bzw. einem binomialverteilten Wahrscheinlichkeitsprofil (**art2**)
- **eps**: Eine Datei im Encapsulated Postscript Format.
- **gzip**: Beispiel für eine ausführbare Datei.
- **gauss**: Weißes Gaußsches Rauschen.

Die Kodiereffizienz von APBI wurde für eine Fensterbreite von 32 Samples untersucht. Eine Anpassung der Maske erfolgte alle 1, 2, 4, 8 bzw. 16 Fenster. Die Ergebnisse wurden

Tabelle 1. Relative Veränderung der Schaltaktivität bezüglich α_Φ

Seq	α_Φ	APBI ₁	APBI ₂	APBI ₄	APBI ₈	APBI ₁₆	$\alpha_{\hat{x}}$ BI	PBI	APBM	PBM	AMWC ₄	AMWC ₈
art1	2130360	37.1 %	42.0 %	42.0 %	42.0 %	42.0 %	12.7 %	12.8 %				
art2	160896								40.6 %	47.7 %	38.3 %	32.9 %
eps	221309	10.6 %	10.6 %	10.5 %	10.4 %	10.2 %	4.5 %	11.4 %	21.0 %	21.1 %	19.7 %	23.7 %
gzip	154620	8.8 %	8.6 %	8.0 %	7.0 %	6.6 %	5.5 %	7.8 %	27.4 %	28.6 %	20.0 %	15.6 %
gauss	4086760	7.1 %	7.1 %	7.1 %	7.1 %	7.1 %	11.2 %	11.2 %	0.5 %	1.5 %	-0.8 %	-1.7 %
Avg		15.9 %	17.1 %	16.9 %	16.6 %	16.5 %	8.5 %	10.8 %	22.4 %	24.7 %	19.3 %	17.8 %

Tabelle 2. Statisch vs. adaptiv: Relative Veränderung der Transitionen für APBM und PBM bezogen auf α_Φ

Seq	APBM	PBM _{art2}	$\alpha_{\hat{x}}$ PBM _{eps}	PBM _{gzip}	PBM _{ga.}
art2	40.6 %	47.7 %	-16.5 %	-14.0 %	-2.3 %
eps	21.0 %	-33.5 %	21.1 %	-4.0 %	-24.3 %
gzip	27.4 %	-77.4 %	-5.1 %	28.6 %	-54.3 %
gauss	0.5 %	-0.3 %	0.5 %	0.3 %	1.5 %
Avg	22.4 %	-15.9%	0.0 %	2.7 %	-19.9%

mit dem Businvert und dem Partiellen Businvert Kodierverfahren verglichen, da APBI darauf basiert. Für die APBM-Simulationen wurde der Systembus in $v = 8$ Subbusse der Breite $s = 4$ aufgesplittet. Die Extraktion der pdf berücksichtigt jeweils 32 Samples. Die Ergebnisse wurden mit PBM als dem zugehörigen statischen Verfahren verglichen. Die Untersuchungen für AMWC wurden für Subbusbreiten von $s = 4$ und $s = 8$ durchgeführt. Die Breite der Samplingtabelle wurde von $k = 2$ bis $k = 7$ variiert, während die für die Codezuordnung relevanten Bits zwischen $m = 1$ und $m = 5$ verändert wurden. Als bezüglich Effizienz und Hardwareoverhead günstigste Kombinationen erwiesen sich für $s = 4$: $k = 4, m = 2$ und für $s = 8$: $k = 4, m = 3$. Tabelle 1 zeigt die prozentuale Reduktion der Schaltaktivität aller untersuchten Kodierverfahren.

Wie erwartet erreichend die adaptiven Verfahren durch ihre Anpassungsfähigkeit eine im Vergleich zu den statischen Kodierverfahren wesentlich stärkere Reduktion der Schaltaktivität. Besonders stark wirkt sich die Anpassungsfähigkeit dynamischer Verfahren bei der Kodierung von Datenströmen mit über der Zeit veränderlicher Schaltaktivität wie **art** aus. Im Gegensatz dazu kann die Kodiereffizienz von Businvert für Datenströme wie **gauss** mit einer gleichverteilten Schaltaktivität und von $p = 0.5$ und untereinander nicht korrelierten Leitungen nicht verbessert werden, da Businvert für solche Datenströme optimal ist (Stan and Burleson, 1995). Die wahrscheinlichkeitsbasierten Kodierverfahren besitzen im Durchschnitt ein höheres Reduktionspotential, das jedoch vom zu kodierenden Datenstrom abhängt. Treten die Datenworte wie bei **gauss** gleichverteilt auf, läßt sich durch eine

Abbildung auf Codeworte gleicher Länge nur eine marginale Reduktion der Schaltaktivität erreichen.

In einem weiteren Experiment wurde untersucht, wie sich die Kodiereffizienz statischer Verfahren wie PBM verändert, wenn sie für einen Datenstrom optimiert wurden, jedoch auf Datenströme mit abweichenden Parametern angewendet werden. Wie Tabelle 2 zeigt, steigt die Schaltaktivität sogar über die des unkodierten Datenstromes, so dass die mittels des optimierten Verfahrens ermittelten Werte für E_α nur auf Datenströme mit korrespondierender Statistik übertragbar sind. Im Gegensatz dazu passen sich APBM und AMWC ausgehend von einer initialen Abbildungsvorschrift an die statistischen Parameter an.

Zur Beurteilung der Gesamtverlustleistungsbilanz wurde in einem weiteren Experiment den Verlustleistungsanteil der Coder-Decoder Implementierungen der Kodierverfahren untersucht. Dazu wurden die Modelle auf die entsprechenden Primitive der XILINX-VIRTEX-Bibliothek abgebildet. Unter Berücksichtigung der genutzten CLB-Ressourcen wurde mit dem Xilinx Virtex Power Estimate Worksheet die Verlustleistung abgeschätzt. Die resultierenden Werte sind in Tabelle 3 zusammengefaßt.

5 Schlußbemerkungen

Die Ergebnisse dokumentieren die hohe Effizienz der vorgestellten adaptiven Kodierverfahren. Sie reduzieren die Transitionen vor allem dann sehr effizient, wenn Datenströme mit sich zeitlich ändernden statistischen Parametern übertragen werden. Die Untersuchungen zeigten, daß statische Verfahren, die a-priori an die Statistik des zu übertragenden Datenstromes angepaßt werden müssen, nur dann effizient die Schaltaktivität reduzieren, wenn Datenströme mit korrespondierender Statistik kodiert werden. Andernfalls ist sogar eine Erhöhung der Schaltaktivität gegenüber dem nicht kodierten Datenstrom möglich.

Verglichen mit statischen Kodierverfahren besitzen adaptive Verfahren einen erhöhten Hardwareoverhead, der sich in einer gesteigerten Eigenverlustleistung widerspiegelt. Der für APBI realisierte Ansatz vergrößerter Maskenaktualisierungsintervalle konnte den Eigenverlustleistungsanteil von Coder-Decoder System bei Erhaltung der Kodiereffizienz wirksam senken. Die Implementierung von AMWC erziel-

Tabelle 3. Abschätzung der Eigenverlustleistung

Verfahren		FG	CY	DFE	RAM	BRAM	Power
AMWC ₄	cod	222	0	126	68	0	7 mW
	dec	225	0	130	68	0	8 mW
AMWC ₈	cod	147	21	161	0	6	6+41 mW
	dec	150	21	161	0	6	6+41 mW
APBM	cod	663	0	140	96	0	12 mW
	dec	714	31	173	128	0	14 mW
APBI	cod	679	9	337	0	0	14 mW
	dec	532	9	307	0	0	12 mW
PBM	cod	32	0	32	32	0	4 mW
	dec	32	0	32	32	0	4 mW
BI	cod	115	9	98	0	0	2 mW
	dec	532	9	65	0	0	1 mW

te eine drastische Senkung des Eigenverlustleistungsanteiles, der in APBM Coder-Decoder Systemen umgesetzt wird bei gleichzeitig effizienter Minderung der Schaltaktivität auf dem Bus.

Acknowledgement. Diese Arbeit wird von der DFG im Rahmen des Schwerpunktprogrammes VIVA unter dem Kennzeichen MU1024/5-2 gefördert.

Literatur

- Benini, L., Micheli, G., Macii, E., Sciuto, D., and Silvano, C.: Asymptotic Zero-Transition Activity Encoding for Address Buses in Low-Power Microprocessor-Based Systems, in Great Lakes VLSI Symposium, pp. 77–82, 1997.
- Benini, L., Micheli, G., Macii, E., Sciuto, D., and Silvano, C.: Address Bus Encoding Techniques for System-Level Power Optimization, in Design Automation and Test in Europe DATE, 1998.
- Benini, L., Macii, A., Macii, E., Poncino, M., and Scarsi, R.: Synthesis of Low-Overhead Interfaces for Power-Efficient Communication over Wide Buses, in 36th Design Automation Conference DAC, 1999.
- Kretzschmar, C., Siegmund, R., and Mueller, D.: Adaptive Bus Encoding Technique for Switching Activity Reduced Data Transfer over Wide System Buses, in Workshop on Power and Timing Modeling and Optimization PATMOS, pp. 66–75, Springer, Goettingen, Germany, 2000.
- Kretzschmar, C., Siegmund, R., and Mueller, D.: Auto-optimizing Bus Encoding for reduced Power Dissipation in Dynamically Reconfigurable Hardware, in International Conference on Engineering of Reconfigurable Systems and Algorithms ERSA, pp. 71–77, 2001.
- Ramprasad, S., Shanbhag, N., and Hajj, I.: A Coding Framework For Low-Power Address And Data Busses, IEEE Transactions on VLSI Systems, 1999.
- Satoshi Komatsu, Makoto Ikeda, and Kunihiko Asada: Bus Data Encoding with Adaptive Code-book Method for Low Power IP Based Design, in International Workshop on IP based design and Synthesis, pp. 77–81, 2000.
- Shin, Y., Chae, S., and Choi, K.: Partial Bus-Invert Coding for Power Optimization of System Level Bus, in Int'l Symposium on Low Power Electronic Design ISLPED, pp. 127–129, 1998.
- Stan, M. and Burleson, W.: Bus-Invert Coding for Low-Power I/O, in Transactions on VLSI Systems, 3, pp. 49–58, 1995.