REDUCING BUS TRANSITION ACTIVITY BY LIMITED WEIGHT CODING WITH CODEWORD SLIMMING

Vijay Sundararajan, Keshab K. Parhi *

Dept. of ECE University of Minnesota
E-mail: vijay@ee.umn.edu, parhi@ee.umn.edu

ABSTRACT

Transitions on high capacitance busses in VLSI systems result in considerable power dissipation. Various coding schemes have been proposed in literature to encode the input signal in order to reduce the number of transitions. Number of transitions can be reduced by introducing redundancy in data transferred over the busses. For a given amount of redundancy there exists a lower bound on the average number of transitions. In this paper we derive a new coding scheme which leads to extremely practical techniques for bus transmission that reduce bus transitions to within 3.96-8.42% of the lower bound depending on the redundancy employed. There is also a net reduction in power dissipation ranging from 8.53-21.88% over an uncoded bus transmission scheme. This savings in power dissipation is identical to that for bus-invert coding per word transmitted the higher efficiency brought about by codeword slimming, however, results in shorter codewords than bus-invert coding which in turn results in higher energy efficiency in word transmission. Applications suitable for this new technique include systems relying on bit-serial implementation and systems with bit-parallel implementations where the cost of extra parallel-to-serial and serial-to-parallel data-format converters is marginal compared to the power savings obtained.

1. INTRODUCTION

Busses constitute an important resource for addressing and data transfer in implementation of VLSI systems. Reducing the power consumed in busses while transferring data is therefore a high priority objective in minimizing power consumption for the entire system. The fact that the power consumed in bus accesses account for a significant fraction of the total power consumed in VLSI systems has been independently established by many researchers, [1], [2]. This is because the capacitance of shared busses is quite large in comparison to the capacitance of other data-path units. There are essentially two ways to reduce power consumption in busses. The first one involves minimizing bus accesses by either reducing the number of data-path units connected to large busses [2] or reducing the number of accesses of READ/WRITE busses for large memory units by algorithm transformations[3]. The second way to reduce power consumed in busses is to reduce the effective capacitance of busses by reducing bus transition activity. In this regard many researchers have studied reduction of bus transition activity by resorting to coding, similar to error-correcting codes, [4], [5], [6].

Typically system busses can be broadly categorized as address busses and data busses. The nature of "data" transmitted over these two bus types have marked dissimilarities. The "addresses" transmitted over address busses routinely differ by small increments. Transferring these increments separately, [7], can lead to significant reduction in power consumption. The "data" transmitted over data busses, however, do not usually exhibit such a relationship. Strategies to reduce the power consumed in data busses therefore follow a different plan.

In [4] lower/upper bounds are established on the average power consumed in data-busses using Shannon's channel coding theorem [8] and the concept of entropy. Additionally, an asymptotically "optimal" coding strategy based on the popular data-compression scheme, Lempel-Ziv coding, [8], has been suggested as a power saving strategy in [4]. Unfortunately, the overhead of implementing a Lempel-Ziv encoder/decoder may outweigh the gains due to the reduction of bus transition activity obtained. In applications where the transmission order of data is immaterial; large power savings have been demonstrated in [9] by reordering data using a Minimum Cost Hamiltonian Path (MCH) formulation. The idea is to reorder the words to be transmitted in such a way so that the net switching activity in transmitting all the words is minimized. Unfortunately, hardware implementation of a MCH solver even using simplistic heuristics is a difficult proposition, and even the simplest implementations are likely to consume a lot of power offsetting the benefits due to reordering of the transmitted words. One extremely practical approach which, however, is quite inefficient in terms of the bounds established in [4] has been proposed in [6]. This method called bus-invert coding, see Fig. 1, uses an extra bit line and complements data if doing so reduces transition activity between successively transmitted data words. When words are transmitted in a complemented form the extra bit-line is high and this bit-line is low if the words are transmitted in an uncomplemented form. The encoder here will just consist of a Hamming distance computation between successive pair of data-words, a simple comparator and a set of conditional inverters or XOR gates. The decoder, on the other hand, will only consist of a set of conditional inverters or XOR gates. In [5] limited weight coding, which includes bus-invert coding as a special case, has been introduced as a strategy for reducing power consumption in data-busses. Once again this coding technique is highly inefficient with respect to the bounds in [4]. In this paper we propose a new scheme of instantaneously decodable codes that involve transmitting words in a bit-serial word-parallel manner rather than the traditional word-serial bit-parallel transmission order. We demonstrate that such a scheme can lead to extremely practical coding which comes close to the lower bound of efficiency in [4]. In one such case related to bus-invert coding we demonstrate the design of an extremely practical encoder/decoder which is comparable in hardware com-

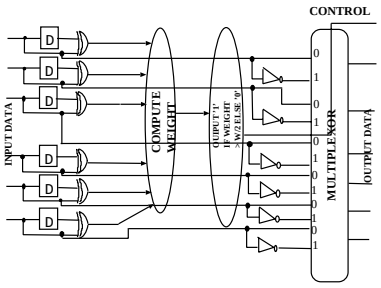

Figure 1. A Bus-invert encoder, the delay block shown indicates that the XOR is computed between successive data-words. The output of the XOR gates is the displacement between successive data-words. The displacement between successive data-words is input to a block (COMPUTE WEIGHT) which counts the number of 1's on it. In the next block, if the number of 1's is greater than $\frac{W}{2}$ then a 1 is output otherwise a 0 is output by this block. Finally, the output of the previous block is the control input to a group of multiplexors which output the input bit if the control is 0 and output the complement of the input bit if the control is 1. The control bit along-with the output from the multiplexors constitutes the output data-word.

*This research has been supported by the Army Research Office under grant number DA/DAAG55-98-1-0315.

plexity to the encoder/decoder for bus-invert coding.

2. PAST WORK

2.1. Redundant Limited Weight Coding

This method uses redundant bitlines to limit the Hamming distance between successively transmitted data-words to be under a predetermined constant d. Limiting the Hamming distance between successively transmitted words automatically limits the transition activity. The weight of a data-word is defined as the number of $1's$ on it. The displacement, $D(A,B)$, between two data-words, $A = a_0 \cdots a_{W-1}$, $B = b_0 \cdots b_{W-1}$, of size W is defined as follows,

$$D(A,B) = a_0 \bigoplus b_0 \cdots a_{W-1} \bigoplus b_{W-1}. \quad (1)$$

Example 1 *Let A = 10101, and B = 11010 be two 5-bit words then the displacement between A, B, D(A, B) = 01111. Also the weight of the displacement is 4.*

The Hamming distance between two data-words A, B is the weight of the the displacement $D(A,B)$ between them. We will now try to determine the minimum redundancy required to limit the displacement between successive data-words of size W to a fixed value $d \leq \frac{W}{2}$. Let us assume that we need M bits per data-word with $M > W$ to guarantee that the Hamming distance between successive data-words on the bus is $\leq d$. There are 2^W data-words of size W, and this is also the number of possible displacements $D(A,B)$ between successive data-words A, B. Also, the number of words of size M which have a weight $\leq d$ is given by $\sum_{i=0}^{d} \binom{M}{i}$. We will code the displacement, $D(A,B)$, between successive data-words A, B using M bits in such a way that the weight of $D(A,B) \leq d$. Using the resulting codewords for successive displacements we can compute the codewords for data to be transmitted over the data-bus. In order to restrict the Hamming distance between successive words, A, B, to a value $\leq d$. We must satisfy the following inequality,

$$\sum_{i=0}^{d} \binom{M}{i} \geq 2^W. \quad (2)$$

The minimum value M satisfying (2) will give us the most efficient word size to use for transmitting the data-words over the data-bus.

Example 2 *Consider a limited weight coding scheme for W-bit data with $d \leq \frac{W}{2}$. It turns out that, for even values of W, $\sum_{i=0}^{\frac{W}{2}} \binom{W+1}{i} = 2^W$. Hence when $d = \frac{W}{2}$ then $M = W + 1$.*

Bus-invert coding is a special case of a limited weight code with $d = \frac{W}{2}$. For ease of exposition we assume an even value for W in the rest of the paper.

2.2. Bounds on Bus Transition Activity

2.2.1. Random Variables and Entropy

Let X be a discrete random variable with alphabet X and probability mass function p(x) = Pr(X = x), $x \in X$. A measure of the information content of X is given by its *entropy* H(X), which is defined as follows [8],

$$H(X) = -\sum_{x \in \chi} p(x) log_2 p(x)\ bits. \quad (3)$$

This definition of the measure of information implies that the greater the *uncertainty* in the source output, the higher is its information content. In a similar fashion, a source with zero uncertainty would have zero information content and therefore its entropy, from (3), would be equal to zero.

In [4] it is shown that if we use M bits on an average to transmit data-words of size W from a source with entropy of H bits per word then the average transition activity T is bounded on both sides as,

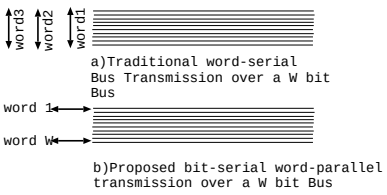

Figure 2. The traditional word-serial bus transmission scheme and the proposed word-parallel bit-serial transmission scheme.

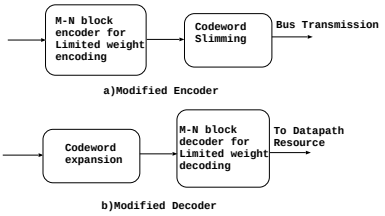

Figure 3. Encoder and decoder for efficient bus transmission. Note that the data-to-displacement converter before the limited weight encoding/decoding block and the displacement-to-data converter after the limited weight encoding/decoding block are not shown in the figure.

$$H^{-1}(\frac{H}{M})M \leq T \leq (1 - H^{-1}(\frac{H}{M}))M, \quad (4)$$

and the bounds in (4) are asymptotically achievable.

In [4] the authors then proceed to show the construction of an encoder/decoder based on a popular technique called Lempel-Ziv coding which asymptotically achieves the lower bound in (4) for the transition activity. However, implementing a Lempel-Ziv encoder/decoder can be extremely power intensive. Therefore we need to design simpler schemes that can bring transition activity closer to the lower bound in (4).

3. INSTANTANEOUSLY DECODABLE EFFICIENT CODING

The coding rate of a bus encoding scheme is defined as $R = \frac{W}{M}$. With traditional bus transmission schemes the *universe* of code rates which can be achieved is fairly limited and it is often not possible to achieve the lower bounds of efficiency specified in (4). Imagine a bus transmission scheme where data words are transmitted in a word-parallel bit-serial fashion. That is in a W bit bus each bitline will have independent bit-by-bit transmission of data words, see Fig. 2. Now each bitline can be considered independently for minimization of transition activity.

Now, imagine a coding scheme where encoding is done on the transitions/displacement between successive bits transmitted over the bitline rather than the data bits themselves. A *M-N block code* is the result of an encoding scheme which acts on M successive transitions/displacement bits and codes these using N bits. These N coded displacement bits can then be used to recover the coded data bits to be transmitted over the bitline by using a displacement-to-data conversion. The purpose of encoding is to reduce the weight of the N encoded displacement bits so that bus transition activity is reduced. For *unique* and *instantaneous* decodability of the codewords none of the allowed codewords must be a prefix of another codeword. This is a necessary and sufficient condition for instantaneous decodability. If we use standard limited weight codes then instantaneous decodability is ensured as all the codewords are distinct and of the same size and hence no codeword is the prefix of another. However, this leads to poor efficiency with respect to the bounds in (4). It turns out that starting from a limited weight coding scheme, with weight limit d, we can generate a new coding scheme which has much higher efficiency by doing the following,

Codeword Slimming

1. Examine every codeword. If the codeword has a weight strictly less than d, then retain it as such.

2. Otherwise look for the shortest leading set of bits in the codeword that have a weight of d, i.e., that have d 1's among them. Throw away the remaining bits in the codeword and the reduced bit-string gives us the new codeword.

In the rest of the paper we will refer to the above technique as *codeword slimming*. In Fig. 3 we see a high level view of the new encoding/decoding strategy. The initial data transmitted after limited weight encoding and codeword slimming can be recovered by the reverse process of *codeword expansion* followed by limited weight decoding.

Theorem 1 *The coding scheme remains reversible after codeword slimming. That is, the original data can be recovered from the coded data after codeword slimming.*

Proof:The proof consists of outlining a decoding strategy for the received data. We first calculate the displacement between successive bits and then do the following. Count the number of 1's till one of two things happen: 1)the number of received bits equals N (the underlying limited weight block code is M-N), or 2)the number of received 1's equals d, in which case we zero-pad the received string to a length of N bits. This process is then continued over the remaining received bits. The previously mentioned process constitutes codeword expansion the expanded data can then be subjected to a M-N limited weight decoding process and finally the transmitted data can be recovered. The complexity of the decoding process is only marginally higher than limited weight decoding.

Example 3 *Consider the following limited weight 2-3 code,* $00 \leftrightarrow 000$, $01 \leftrightarrow 001$, $10 \leftrightarrow 010$, $11 \leftrightarrow 100$. *This code limits the weight of successive pairs of displacements to 1. The codewords after codeword slimming would be* $00 \leftrightarrow 000$, $01 \leftrightarrow 001$, $10 \leftrightarrow 01$, $11 \leftrightarrow 1$. *Now the data-string* 1001010101100110 *gives rise to the following displacement string (with first bit retained as such)* 1101111111010101. *After encoding the displacement string we get the following string* 1001111001001001, *which limits the number of transitions to* 7 *down from the initial* 11. *The actual data transmitted over the bus will be* 1110101110001110. *Note that even though in this case the coded string has the same length as the uncoded string in general this will not be true.*

Theorem 2 *The coding scheme remains instantaneously decodable after codeword slimming.*

Proof: We only have to prove the prefix property for all the codewords, i.e., we need to show that none of the codewords, after codeword slimming, is a prefix of another codeword. Since the initial coding scheme was instantaneously decodable to begin with and hence prefix free; the codewords of length M, i.e., the unmodified codewords after codeword slimming still satisfy the prefix property. The only codewords that are "shrunk" after codeword slimming have exactly d 1's. And since the original codewords can have no more than d 1's, due to the limited weight property, therefore these modified codewords can not be the prefix of any other codeword. And hence the coding scheme is instantaneously decodable.

4. BUS-INVERT CODING WITH CODEWORD SLIMMING

As already mentioned bus-invert coding is a special form of limited weight coding with an extra bit of redundancy. The advantage of bus-invert coding is that it leads to an extremely practical encoder/decoder that can be implemented easily in hardware. Therefore applying codeword slimming after bus-invert coding can lead to easily implementable bus-encoding schemes that are also extremely efficient with respect to the bounds in (4). We will first compute a closed form expression for the transition activity with bus-invert coding and then demonstrate the efficiency of bus-invert coding followed by codeword slimming.

Example 4 *For a W-bit bus driven by a data source with equal probability for all 2^W data words the transition activity is $\frac{W}{2}$. Now consider bit-serial word-parallel transmission as described earlier and consider a W-(W + 1) limited weight code applied to the individual bitlines following the bus-invert coding philosophy. So first the displacement of successive bits in the data string to be transmitted is computed. Then the displacement of the data to be transmitted is broken down into substrings of length W. These substrings are then padded with an additional bit. If the number of 1's in these substrings is $> \frac{W}{2}$ the substring is complemented and the extra padded bit is set to 1. Otherwise the substring is kept unchanged with the extra padded bit at 0. Now the number of substrings of length W that have precisely k 1's is given by $\binom{W}{k}$. Assuming W to be even, and using elementary combinatorics, the average transition activity per word for a bitline is therefore given by,*

$$
\begin{aligned}
Tr_W^{W+1} &= \frac{\sum_{k=0}^{\frac{W}{2}} k \times \binom{W}{k} + \sum_{k=\frac{W}{2}+1}^{W} (W-k+1) \times \binom{W}{k}}{2^W}, \\
&= \frac{\sum_{k=0}^{\frac{W}{2}-1} 2k \times \binom{W}{k} + \sum_{k=0}^{\frac{W}{2}-1} \binom{W}{k} + \frac{W}{2} \times \binom{W}{\frac{W}{2}}}{2^W}, \\
&= \frac{\sum_{k=1}^{\frac{W}{2}-1} 2W \times \binom{W-1}{k-1} + \sum_{k=0}^{\frac{W}{2}-1} \binom{W}{k} + \frac{W}{2} \times \binom{W}{\frac{W}{2}}}{2^W}, \\
&= \frac{(W+1)2^{W-1} - \frac{W+1}{2} \times \binom{W}{\frac{W}{2}}}{2^W}. \qquad (5)
\end{aligned}
$$

Therefore, there is a net reduction in switching activity over transmitting the uncoded bits. The inefficiency of this code with respect to (4) is however brought about by using $W+1$ bits for every string of W bits in the displacement of the original data-string. Codeword slimming will reduce the average length of a coded bit-string to a value L such that $W < L < W+1$. The average length of a codeword for bus-invert coding with codeword slimming is,

$$
\frac{(W+1) \times (2^W - \sum_{k=\frac{W}{2}}^{W} \binom{k-1}{\frac{W}{2}-1}) + \sum_{k=\frac{W}{2}}^{W} k \times \binom{k-1}{\frac{W}{2}-1}}{2^W}. \qquad (6)
$$

The first term in the numerator of (6) corresponds to the codewords of length $W+1$ the second term corresponds to the codewords with length $< W+1$.

5. ANALYTICAL RESULTS

Table.1 provides a comparison of bus-invert coding and bus-invert coding with codeword slimming for various block encoding schemes. Also shown is the lower bound of efficiency from (4) for the data-rate employed in the transmission scheme. As can be seen the power savings over an uncoded bit-stream varies from 8.5%-21.88% with the savings diminishing for larger block codes. Due to this fact large block codes will never be used in practice as these lead to lower power gain as opposed to smaller block codes. Also, the decoding latency for smaller block codes is significantly smaller than that for larger block codes. Also as can be observed bus-invert coding with codeword slimming comes quite close to the lower bound of efficiency in (4) for all the cases considered in Table. 1. It can however be seen that for coding on larger blocks (32, 64) the gains of codeword slimming over regular bus-invert coding starts to diminish. Since power gain is larger and decoding latency is smaller for smaller block codes it is unlikely that a block size > 8 will be used for coding. At this block size codeword slimming increases the efficiency of the code by 10.02%. Also, to be observed is the fact that the average length of the codeword after codeword slimming is smaller than the average codeword length for bus-invert coding. This results in an increase in bus transmission speed after codeword slimming. The power savings over uncoded transmission are identical for regular bus-invert coding and

Table 1. Comparison of bus-invert coding and bus-invert coding with codeword slimming. Also, tabulated is the transition activity lower bound for both schemes. Column 1 identifies the block size for bus-invert coding. Columns 5 and 9 respectively tabulate the inefficiency of bus-invert coding and bus-invert coding with codeword slimming over their corresponding lower bounds for transition activity. The last column tabulates the power advantage of bus-invert coding with codeword slimming over uncoded bus transmission. Note that the transition activity tabulated is the sum of transition activities over all the bitlines of the bus and therefore lies between 0 and W for a W-bit bus.

W-W+1	Avg. Length Bus-Inv (bits)	Tran. Act. Bus-Inv. (bits/word)	Lower Bound Bus-Inv. (bits/word)	% ineff. Bus-Inv.	Avg. Length Slimm. (bits)	Tr. Act. Slimm. (bits/word)	Low. Bnd. Slimm. (bits/word)	% ineff. Slimm.	% Pow. over uncoded
4-5	5	1.5625	1.2150	28.6%	4.3758	1.5625	1.4411	8.42%	21.88%
6-7	7	2.4062	1.9684	22.24%	6.4531	2.4062	2.2283	7.98%	19.79%
8-9	9	3.2695	2.7567	18.6%	8.5078	3.2695	3.0390	7.58%	18.26%
10-11	11	4.1465	3.5684	16.2%	10.5488	4.1465	3.8672	7.22%	17.07%
12-13	13	5.0337	4.3966	14.49%	12.5811	5.0337	4.7078	6.92%	16.11%
14-15	15	5.9290	5.2365	13.22%	14.6072	5.9290	5.5581	6.67%	15.30%
16-17	17	6.8308	6.0894	12.18%	16.6291	6.8308	6.4188	6.42%	14.62%
32-33	33	14.1908	13.1307	8.07%	32.7283	14.1908	13.4972	5.14%	11.31%
64-65	65	29.2712	27.755	5.46%	64.8043	29.2712	28.1575	3.96%	8.53%

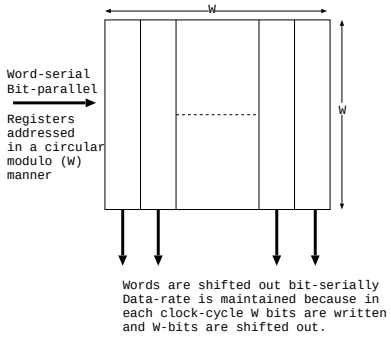

Figure 4. Converting a Word-Serial Transmission Scheme to a Word-parallel bit-serial transmission scheme.

bus-invert coding with codeword slimming. The reduction in the average length of a codeword and thereby increase in transmission speed due to codeword slimming, however, renders it more energy (power $\times$ delay) efficient as compared to regular bus-invert coding.

6. PRACTICALITY OF WORD-PARALLEL BIT-SERIAL TRANSMISSION

One question that still needs to be investigated is regarding the practicality of a data-reordering scheme in which data-words are transmitted individually over a bitline of the bus. This reordering is required because in a traditional word-serial bit-parallel transmission scheme codeword slimming would result in variable length of codewords over various bitlines of the bus thereby complicating the recovery of a dataword at the receiver end. For a W-bit bus a brute-force implementation of a serial to parallel conversion scheme would require the use of a W bit shift register for each bitline of the bus. Which implies the use of $W \times W$ flip-flops. Depending on the area and power penalty this solution incurs this might or might not be satisfactory. However for very large memories and very high capacitive buses the extra cost incurred by this solution might be minimal. On the receiver end depending of how the computational units receiving data from the bus are organized we may or may not require an inverse data-format converter. For instance a computational unit which is implemented in a bit-serial manner does not need a data-format converter, on the other hand a computational unit implemented in a bit-parallel manner would need a data-format converter. Obviously, the proposed technique is a natural low power solution for bit-serial implementations of VLSI systems. On the other hand even in bit-parallel implementations it might be possible to group the data-path in to various clusters and share data-format converters to get a low power implementation of the proposed scheme. Fig. 4 shows the implementation of such a converter, such converters were studied in detail and completely characterized in [10].

7. CONCLUSIONS

A new bus encoding strategy was introduced in this paper that modifies and improves the efficiency of an existing bus encoding technique known as limited weight encoding, [5]. For a special limited weight code called bus-invert coding extremely practical realizations coming close to lower bounds of efficiency, [4], were demonstrated. Applications for the proposed scheme include VLSI systems implemented in a bit-serial manner and bit-parallel systems where the data-format converters can be used without much power penalty. Future work will consist of trying to achieve equally efficient and practical codes corresponding to other limited weight encoding schemes. In addition research will be done to study the effect of codeword slimming in the presence of correlated data with nonuniform probability for data-words.

REFERENCES

[1] D. Liu and C. Svensson, "Power Consumption Estimation in CMOS VLSI Chips," *IEEE Journal of Solid State Circuits*, vol. 29, no. 6, pp. 663–670, 1994.

[2] R. Mehra, L. M. Guerra, and J. Rabaey, "Low Power Architectural Synthesis and the Impact of Exploiting Locality," *Journal of VLSI Signal Processing*, vol. 13, no. 2-3, pp. 239–258, 1996.

[3] S. Wuytack, *et al.*, "Global Communication and Memory Optimizing Transformations for Low Power Systems," in *Proceedings of International Workshop on Low Power Design*, (Napa, CA, USA), pp. 203–208, April 1994.

[4] S. Ramprasad, *et al.*, "Achievable Bounds on Signal Transition Activity," in *Proceedings of ICCAD'97*, (San Jose, CA, USA), pp. 126–129, Oct. 1997.

[5] M. Stan and W. Burleson, "Limited-Weight Codes for Low-Power I/O," *Proceedings of International Workshop on Low Power Design*, pp. 209–214, April 1991.

[6] M. Stan and W. Burleson, "Bus-Invert Coding for Low-Power I/O," *IEEE Transactions on VLSI Systems*, vol. 3, pp. 49–58, March 1995.

[7] L. Benini *et al.*, "Asymptotic Zero-Transition Activity Encoding for Addresses in Low-Power Microprocessor-based Systems," in *Proceedings of GLVLSI-97*, (Urbana, IL, USA), pp. 77–82, March 1997.

[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, 1991.

[9] R. Murgai, M. Fujita, and A. Oliveira, "Using Complementation and Resequencing to Minimize Transitions," *In Proceedings of the 35th ACM/IEEE Design Automation Conference*, pp. 694–697, Jun. 1998.

[10] K. K. Parhi, "Systematic synthesis of DSP data format converters using life-time analysis and forward-backward register allocation," *IEEE Trans. Circuits And Systems II Analog and Digital Signal Processing*, vol. 39, pp. 423–440, July. 1992.