Differential Signaling with a Reduced Number of Signal Paths

Anthony Carusone, Kamran Farzan, and David A. Johns

***Abstract*—Differential signaling is often used for digital chip-to-chip interconnects because it provides common-mode noise rejection. Unfortunately, differential signals generally require $2N$ signal paths to communicate N signals. In this paper, a method for differential signaling is described that requires as few as $N+1$ signal paths for N signals. Using this method, the signal values appear incrementally between neighboring matched signal paths. The technique, called incremental signaling, is similar to dicode ($1-D$) partial response signaling except that the sequence is transmitted in parallel over a bus of wires rather than sequentially in time. Theoretical and simulated bit error rates are presented for several possible implementations of an encoder/transmitter and receiver/decoder for a digital data bus including peak detection and maximum likelihood sequence detection (MLSD). Peak detection uses $N+1$ signal paths and results in a 3-dB performance degradation with respect to independent noise compared with fully differential signaling. The Viterbi algorithm for MLSD uses $N+2$ signal paths but provides only a 1.25 dB improvement over peak detection due to correlated noise on the ($1-D$)-coded sequence. Modified Viterbi algorithms that use $N+2$ signal paths are introduced to cancel the correlated noise sources, resulting in a bit error rate performance comparable with fully differential signaling.**

***Index Terms*—Chip-to-chip interface, differential signaling, maximum likelihood sequence detection.**

I. Introduction

Recently, the noise margin on digital chip-to-chip interconnects has been decreasing for two main reasons. One reason is that supply voltages in digital CMOS processes are decreasing, thereby reducing the voltage available for driving I/Os. A second reason is that small signal swings are being used to reduce dynamic power dissipation on high-speed buses. It has long been known that fully differential signals effectively reject common-mode noise and even-order distortion terms. Since common-mode noise is prevalent on matched PCB traces, differential signaling is effective for both voltage [1], [2] and current-mode [3] digital chip-to-chip interfaces. Fully differential signals are now used in the scalable coherent interface [4], [5] and RamLink [6] standards. Unfortunately, a practical problem with their implementation is that two signal paths are required for each signal. For example, using fully differential signals for a 64-bit data bus would require 128 pins on each IC package and 128 PCB trances routed between ICs. These additional costs are often prohibitive.

This paper describes a general technique for obtaining many of the advantages of fully differential signals while using a reduced number of signal paths. Specifically, N differential signals are communicated over as few as $N+1$ signal paths. In Section II, the basic idea is described. The technique is similar to partial response signaling except that the encoded sequence is transmitted in parallel over a set of wires rather than sequentially in time over a serial connection. Possible implementations are then discussed. As in partial response systems, the simplest approach is peak detection, which requires only $N+1$ signal paths and is discussed in Section III. Maximum likelihood sequence detection (MLSD) is another popular technique for partial response systems. It uses $N+2$ signal paths and is discussed in Section IV. Modifications to the Viterbi algorithm are described in Section V, which bring the performance of MLSD close to that of fully differential signaling using only $N+2$ signal paths. Theoretical and simulated bit error rate (BER) results are presented for each approach on a digital data bus. As in partial response systems, the approaches are general and multilevel signaling is possible. However, in this paper, results are only presented for a binary data bus. Finally, a system that combines this approach with constant-weight digital encoding is proposed in Section VI.

Manuscript received August 2000; revised February 2001. This paper was recommended by Associate Editor M. Helfenstein.

The authors are with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 1A4 Canada.

Publisher Item Identifier S 1057-7130(01)04202-1.

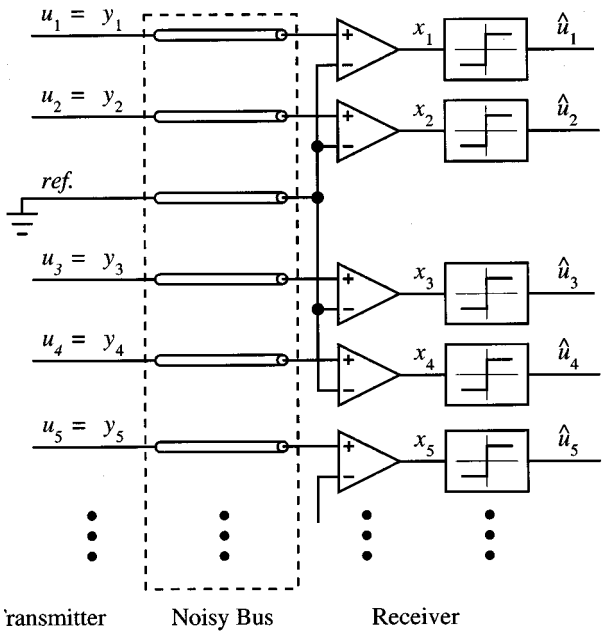

Fig. 1. A practical single-ended signaling system or "pseudodifferential" signaling with additional reference lines added after every four signal paths.

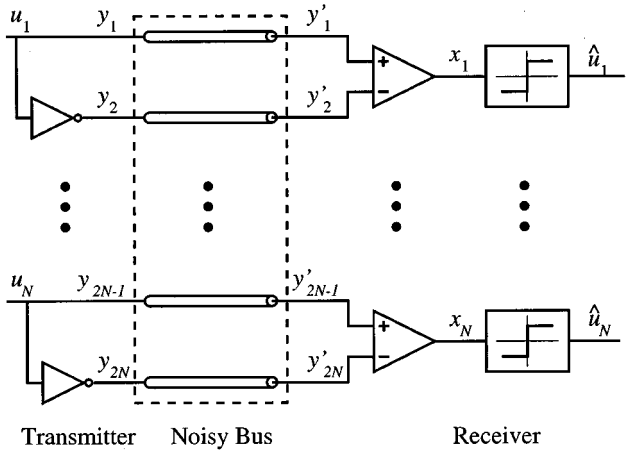

Fig. 2. Block diagram of a fully differential signaling system for binary data.

II. Incremental Signals

In general, the problem is one of communicating N signals, x_k, $1 \leq k \leq N$, over M signal paths, y_l, $1 \leq l \leq M$. In a simple single-ended scheme, $M = N$ and the receiver operates by comparing the signal on each path to reference threshold levels. Of course, the receiver is susceptible to common-mode noise on the bus. To combat this problem, practical single-ended systems often include reference signals transmitted along the bus to provide some common-mode noise

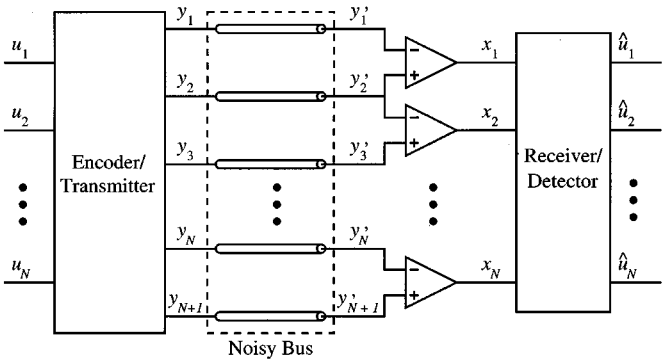

Fig. 3. A general incremental signaling system.

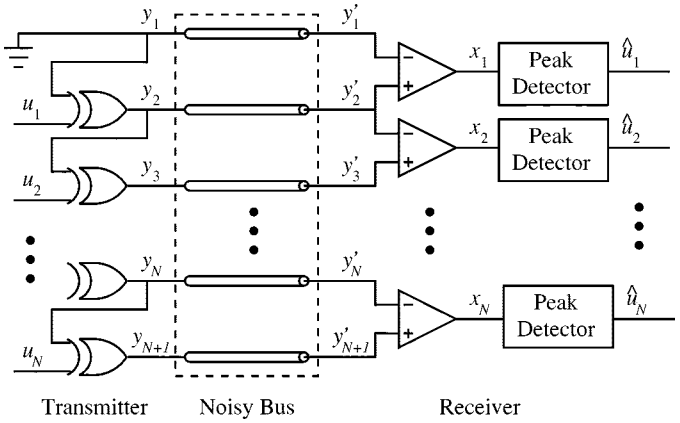

Fig. 4. Block diagram of a possible incremental signaling system for binary data using peak detection.

rejection. This approach has been referred to as "pseudodifferential" signaling. For instance, a system with an extra reference line after every fourth active signal path is shown in Fig. 1. Of course, this increases the pin count by 25%. Furthermore, there will always be some finite common-mode-to-differential conversion due to mismatches between signal paths along the bus. These mismatches can be minimized by comparing only *neighboring* signal paths.

Using fully differential signals requires $M = 2N$. The signals appear as the difference between neighboring matched signal paths, as shown in Fig. 2

$$x_k = y'_{2k} - y'_{2k-1}. \tag{1}$$

The primes in Fig. 2 and (1) denote noisy signals at the receiver, $y'_k = y_k + n_k$. Note that (1) implies that we have complete freedom to arbitrarily select one-half of the signals levels.[1] Clearly, there is some redundancy inherent in transmitting N signals over $2N$ signal paths. This redundancy is eliminated by having the signals appear incrementally as the difference between adjacent signal paths

$$x_k = y'_{k+1} - y'_k. \tag{2}$$

Using this scheme, hereafter called incremental signaling, the signals still appear differentially between two adjacent wires, so all of the noise-rejection advantages of fully differential signals are obtained using only $M = N + 1$ signal paths. However, in (2), there is only freedom to fix one signal path value, namely, y_1.

A completely general incremental signaling system is shown in Fig. 3. Several possible encoder/transmitter and receiver/decoder combinations are possible, each offering a different compromise between complexity and performance. Interpreting the received signals as the difference between adjacent signal path values as in (2) is analogous to applying the dicode $(1 - D)$ partial response operator to a time series

$$x(k) = y(k) - y(k-1). \tag{3}$$

Therefore, popular approaches to encoding/transmitting and receiving/decoding partial response signals are also applicable here.

III. Peak Detection

To keep the receiver hardware as simple as possible, the information bits can be precoded prior to transmission, as described in [7]. Specifically, the signal path values are encoded according to the following equation:

$$y_{k+1} = (y_k + u_k) \bmod L \tag{4}$$

[1]To minimize power consumption, differential signals are usually driven in a balanced fashion so that $y_{2k} = -y_{2k-1}$.

TABLE I
Signal Values for the Peak Detection System in Fig. 4 with Random Binary Data of Width $N = 6$ and No Noise

k	1	2	3	4	5	6	7
Input Data: u_k	1	1	0	1	0	0	
Transmit Data: $y_k = y_{k-1} \oplus u_{k-1}$ $(y_1 = 0)$	0	1	0	0	1	1	1
Received Signal: $x_k = y_{k+1} - y_k$	1	-1	0	1	0	0	
Recovered Data: $\widehat{u_k} = x_k \bmod 2 = \lvert x_k \rvert$	1	1	0	1	0	0	

where u_k is the (possibly multilevel) information symbol being encoded and L is the number of signal levels to be transmitted on the bus. The receiver must then interpret the received signals modulo-L

$$\hat{u}_k = x_k \bmod L. \tag{5}$$

The modular arithmetic in (4) and (5) has a particularly straightforward hardware implementation when the u_k are binary signals. The modular addition in (4) can be performed by exclusive-OR gates

$$y_{k+1} = y_k \oplus u_k. \tag{6}$$

The modulo-L receiver can be just two differential comparators operating as a peak detector. A system block diagram of this approach is shown in Fig. 4. Note that the precoder includes a cascade of N exclusive-OR gates. However, it is possible to shorten the logic's critical path if it is limiting speed through the use of carry-lookahead or pipelining techniques. Table I shows all of the signal values for a sample binary sequence of length $N = 6$ in the absence of noise. The receiver in Fig. 4 sees the noisy signals, $y'_k = y_k + n_k$. A bit error will occur when independent noise on y'_k and y'_{k+1} causes the differential signal $x_k = y'_{k+1} - y'_k$ to cross one of the slicer thresholds. If the bus signals are members of a binary alphabet with spacing $2A$, σ^2 is the variance of independent Gaussian noise on each y_k, and $\eta = A/\sigma$, then it can be shown that errors will occur with the following probability:

$$P_{e_{\text{peak detection}}} = \frac{3}{2} Q\left(\frac{1}{\sqrt{2}}\,\eta\right) \tag{7}$$

where $Q(\cdot)$ is defined as

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\alpha^2/2}\, d\alpha. \tag{8}$$

For comparison, the fully differential system shown in Fig. 2 would have the following probability of error:

$$P_{e_{\text{fully differential}}} = Q\left(\sqrt{2}\,\eta\right). \tag{9}$$

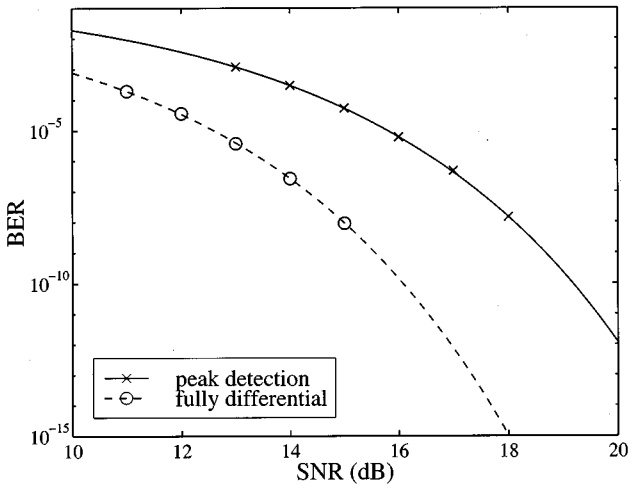

Fig. 5. Bit error rates for the fully differential system in Fig. 2 and the peak detection system in Fig. 4. Lines are theoretical results and markers indicate simulation results.

Of course, since there are approximately twice as many lines being driven in a fully differential system, the power consumption is doubled. In order to take this into account, a normalized signal-to-noise ratio is defined as follows:

$$\text{SNR} = \frac{(\text{number of lines being driven})}{(\text{number of information bits})} \times \eta^2. \tag{10}$$

Equations (7) and (9) are rewritten in terms of signal-to-noise ratio (SNR)

$$P_{e_{\text{peak detection}}} = \frac{3}{2} Q\left(\sqrt{\frac{\text{SNR}}{2}}\right) \tag{11}$$

$$P_{e_{\text{fully differential}}} = Q\left(\sqrt{\text{SNR}}\right). \tag{12}$$

Equations (11) and (12) are plotted along with simulation results in Fig. 5. The theoretical and simulation results indicate a 3-dB decrease in the performance of a digital transmission system using a peak detection-based incremental signaling scheme compared with fully differential signaling. In the following sections, we shall see how it is possible to recover most of that 3 dB using somewhat more complicated receiver hardware.

IV. MLSD with the Viterbi Algorithm

As in magnetic storage systems, which use partial response signals, it is possible to use MLSD for incremental signaling receivers. An incremental signaling system utilizing MLSD at the receiver is shown in Fig. 6. Notice that no precoding logic is necessary. However, it is necessary to utilize $N+2$ rather than $N+1$ signal paths. The extra signal path appears at the end of the bus to terminate the Viterbi algorithm's trellis. Without this extra signal path, the last few bits on the bus would be decoded with approximately the same probability of error as a regular peak detection-based system.

It can be shown (Appendix A) that the probability of error using MLSD is bounded by the following expression:

$$P_{e_{\text{MLSD}}} < Q\left(\frac{\sqrt{6 \cdot \text{SNR}}}{3}\right) + 3Q\left(\sqrt{\text{SNR}}\right). \tag{13}$$

This expression, along with simulation results, is plotted in Fig. 7. The theoretical expressions from (11) and (12) are also shown for comparison. At high SNR, the bound in (13) provides an accurate estimate

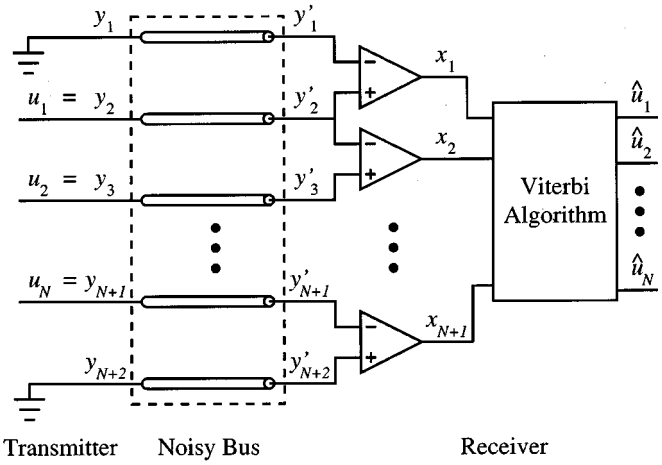

Fig. 6. Block diagram of an incremental signaling system using the Viterbi algorithm for MLSD at the receiver.

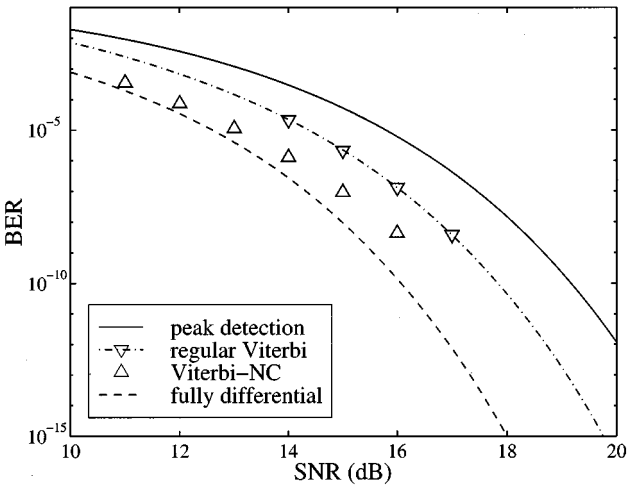

Fig. 7. Bit error rate of the MLSD system in Fig. 6 with the regular Viterbi algorithm and the Viterbi-NC algorithm. Lines are theoretical results and markers are simulation results.

of the bit error rate. MLSD provides approximately a 1.25-dB performance improvement over peak detection—1.75 dB worse than a fully differential system. Of course, high-speed hardware implementation of the Viterbi algorithm is an area of ongoing research. A digital implementation of the algorithm would require an analog-to-digital converter operating on each signal path and considerable digital signal processing. Analog implementations such as those developed for magnetic storage applications [8] are more realistic.

In some partial response systems, the use of an MLSD receiver provides a full 3-dB improvement in BER versus SNR. However, to achieve this 3-dB improvement, the Viterbi algorithm requires independent noise at its inputs [9]. In incremental systems such as Fig. 3, the receiver takes the difference $(1-D)$ between signal path values that are already noisy. Therefore, any independent noise that appears on a given signal path y_k' will [according to (2)] appear on both x_k and x_{k+1}. As a result, the noise on x_k and x_{k+1} is correlated. This causes suboptimal performance of the Viterbi algorithm and a 1.75-dB performance penalty.

V. The Viterbi Algorithm with Noise Cancellation

As described above, the Viterbi algorithm provides only a 1.25-dB performance improvement instead of 3 dB because consecutive values

in the sequence x_k are subject to correlated noise introduced by the $(1-D)$ operator. A similar problem has been encountered in magnetic storage systems where correlated noise is introduced by the magnetic media and by the receive equalizer. Several solutions have been proposed for magnetic storage systems. (See, for instance, [10]–[12].) Generally, the approach taken is to cancel the correlated noise on each sample using a linear noise prediction filter. These techniques, sometimes called noise-predictive maximum likelihood detection, are excellent when the noise fits an autoregressive model. However, they are not as well suited to systems where the noise includes a common-mode (dc) term. In this section, a modification to the Viterbi algorithm is described that cancels both the correlated noise (using noise prediction) and the common-mode noise on x_k resulting in bit error rates comparable to fully differential systems. First, noise prediction will be described in the absence of common-mode noise, similar to the approaches described in [10]–[12]. Then, a further modification is introduced to handle common-mode noise.

If there is no common-mode noise on the bus, the noisy received signals y_k' will be made up of the transmitted value y_k and independent noise n_k.

$$y_k' = y_k + n_k. \tag{14}$$

Therefore, the differential signals at the receiver x_k will include the desired received signal $(y_{k+1} - y_k)$ plus the noise terms n_k and n_{k+1}

$$\begin{aligned} x_k' &= y_{k+1}' - y_k' \\ &= (y_{k+1} + n_{k+1}) - (y_k + n_k) \\ &= (y_{k+1} - y_k) + n_{k+1} - n_k. \end{aligned} \tag{15}$$

Before calculating each branch metric in the Viterbi algorithm, estimates of n_k (called $\tilde{n}_k$) are obtained iteratively using a noise prediction filter

$$\tilde{n}_k = (x_{k-1} - \hat{x}_{k-1}) + \tilde{n}_{k-1}. \tag{16}$$

In (16), $\hat{x}_{k-1}$ refers to the expected value of x_{k-1} corresponding to the survivor path that terminates in the branch under consideration. So, a different estimate $\tilde{n}_k$ must be calculated for each branch in the trellis. For the correct path, $\hat{x}_k = (y_{k+1} - y_k)$. Furthermore, assuming $\tilde{n}_{k-1} = n_{k-1}$, we have from (16) that $\tilde{n}_k = n_k$. The branch metric is then computed using $(x_k + \tilde{n}_k)$ instead of x_k

$$x_k + \tilde{n}_k = (y_{k+1} - y_k) + n_{k+1} - n_k + \tilde{n}_k. \tag{17}$$

The terms $(-n_k + \tilde{n}_k)$ in (17) cancel, resulting in a 3-dB performance improvement.

The problem with this approach is how to initialize $\tilde{n}_1$ for the iterative computation in (16). In the absence of common-mode noise, the following definition works well:

$$\tilde{n}_1 = y_1 + A. \tag{18}$$

However, since (18) relies upon the single-ended signal y_1, any common-mode noise will appear on all $\tilde{n}_k$s and hinder the system's noise performance. To solve this problem, an estimate of the common-mode noise is calculated for each state in the trellis based upon the surviving path for that state

$$\tilde{n}_{\mathrm{CM}} = \frac{1}{k}\sum_{t=1}^{k}(y_t' - \hat{y}_t) \tag{19}$$

$$= n_{\mathrm{CM}} + \frac{1}{k}\sum_{t=1}^{k} n_t. \tag{20}$$

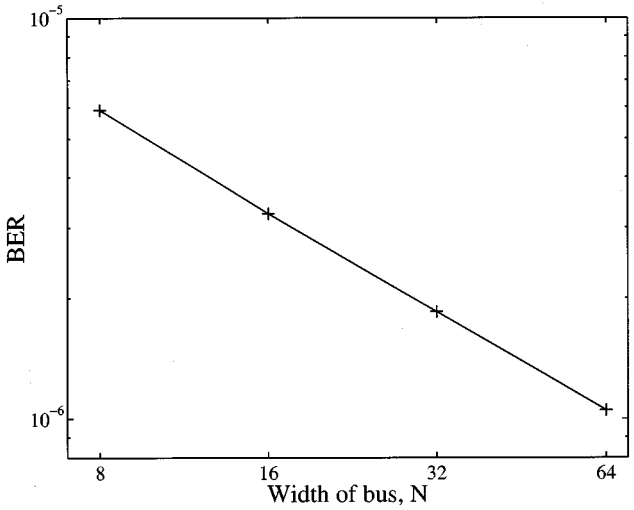

Fig. 8. BER versus bus width N using the Viterbi-NC algorithm for MLSD at SNR $=$ 14 dB.

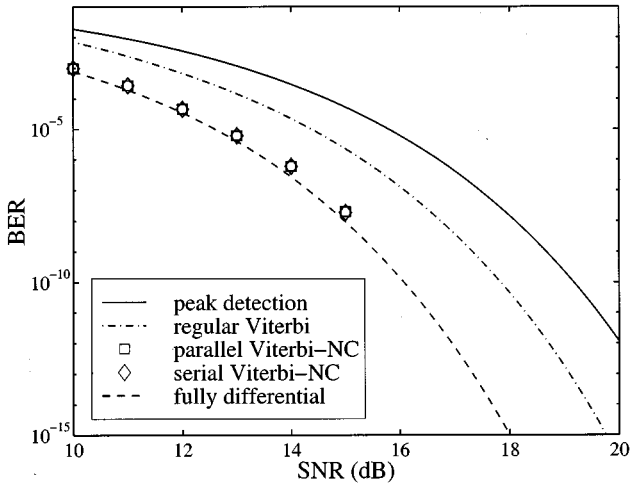

Fig. 9. Simulation results for the serial and parallel Viterbi-NC algorithms.

The term $\hat{y}_t$ in (19) is the expected value of y_t' corresponding to the survivor path for the state under consideration.

So, the procedure for computing branch metrics is

1) Obtain an estimate of the correlated noise term $\tilde{n}_k$ using (16).
2) Obtain an estimate of the common-mode noise $\tilde{n}_{\mathrm{CM}}$ using (19).
3) Compute the branch metric as usual using the value $x_k + \tilde{n}_k - \tilde{n}_{\mathrm{CM}}$ instead of x_k.

The combination of noise prediction and common-mode noise estimation will be called the Viterbi algorithm with noise cancellation (Viterbi-NC). The second term in (20), $(1/k)\sum_{t=1}^{k} n_k$, represents the error in the common-mode noise estimate $\tilde{n}_{\mathrm{CM}}$. This error term limits the noise performance of the Viterbi-NC algorithm to somewhat less than that of a fully differential system (Fig. 7). As k increases, $(1/k)\sum_{t=1}^{k} n_k \rightarrow 0$ and $\tilde{n}_{\mathrm{CM}} \rightarrow n_{\mathrm{CM}}$. Therefore, one would expect most of the errors to occur at the start of the bus when the common-mode noise estimate is still poor. Furthermore, the overall bit error rate should improve as the bus gets wider. As a verification, Fig. 8 shows simulation results for the Viterbi-NC algorithm at different bus widths

Since $\tilde{n}_{\mathrm{CM}}$ is an accurate estimate of n_{CM} at the end of the bus, the following improvements are possible.

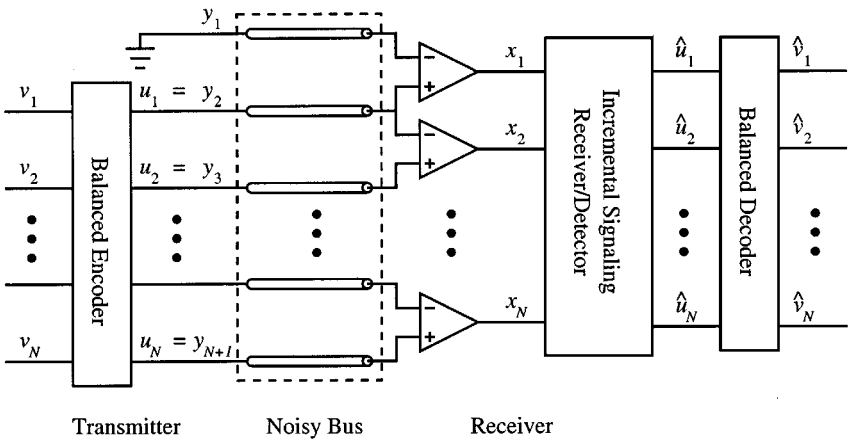

Fig. 10. A system combining incremental signaling with balanced codes.

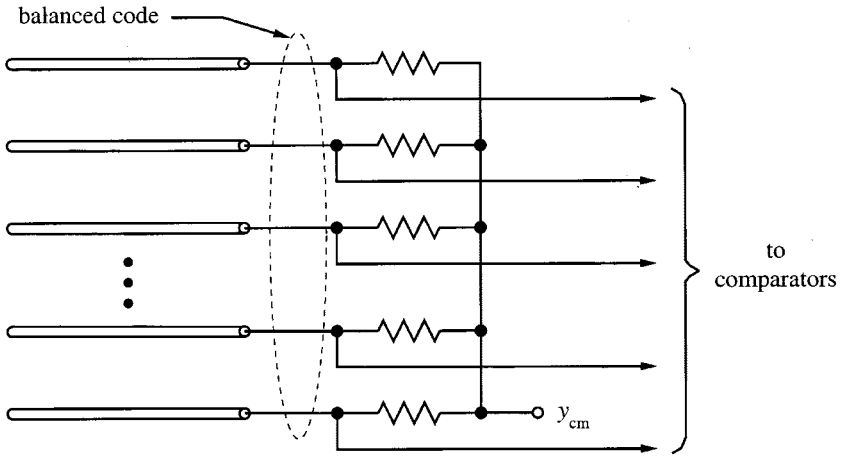

Fig. 11. Termination of a balanced bus at the receiver with common-mode tap $y_{\rm cm}$.

1) Apply the algorithm once from beginning to end, y_1' to y_N', and keep only half of the decoded sequence, $u_{(N/2)+1}$ to u_N. At the same time, the algorithm can be applied from the end to the beginning, y_N' to y_1', and keep the first half of the decoded sequence, u_1 to $u_{N/2}$. This will be referred as the "parallel Viterbi-NC" algorithm.
2) Run the Viterbi-NC algorithm once. Then apply the algorithm again using the final (accurate) estimate $\tilde{n}_{\rm CM}$ for common-mode noise. This will be referred to as the "serial Viterbi-NC" algorithm.

Simulation results for the parallel and serial Viterbi-NC algorithms are presented in Fig. 9 for $N = 32$. Both algorithms show roughly the same performance as fully differential signaling. Also, these results were found to be relatively insensitive to the bus width N. Of course, the complexity of these algorithms is significant, and efficient high-speed hardware implementations are an open issue.

VI. Balanced Codes

Another advantage of differential signaling over single-ended schemes is that switching noise both on-chip and radiated on the PCB are, to a first-order approximation, canceled since there are always an equal number of high and low signals on the bus. It is well known that an N-bit binary bus can be coded on $(N + \log_2 N)$ bits or less to have an equal number of ones and zeros at all times. (See, for instance, [13] or [14].) As long as precoding is not used at the transmitter, these codes can be combined with incremental signaling, as shown in Fig. 10. The resulting system would reject common-mode interference and minimize switching noise similar to a fully differential system, but with far fewer IC pins and PCB traces. For instance, a 32-bit bus could be implemented with just 38 interconnects instead of 64.

Interestingly, if a balanced code is used, all received signals can be terminated at a common point in the receiver, as shown in Fig. 11. The resulting signal at $y_{\rm cm}$ provides an estimate of the common-mode noise on the bus, which might simplify hardware implementations of the modified Viterbi algorithms described in Section V by taking $\tilde{n}_{\rm CM} = y_{\rm CM}$.

VII. Conclusions

A technique called incremental signaling has been discussed which allows for N differential signals to be communicated via as few as $N + 1$ signal paths. It is possible to implement the technique using just two differential comparators per bit. Common mode noise and even order distortion terms would be completely rejected by such a system. However, the BER performance would be 3 dB worse than a fully differential system with respect to independent noise sources. One possible approach for regaining the 3 dB of lost performance is MLSD. Several algorithms for MLSD were presented and their relative performance is summarized in Table II. Although the exact numbers depend upon the noise model used for the data bus simulations, general trends can be identified. Using the modified Viterbi algorithms, it is possible to obtain practically the same noise performance as fully differential signaling using just $N+2$ signal paths. The techniques presented are all

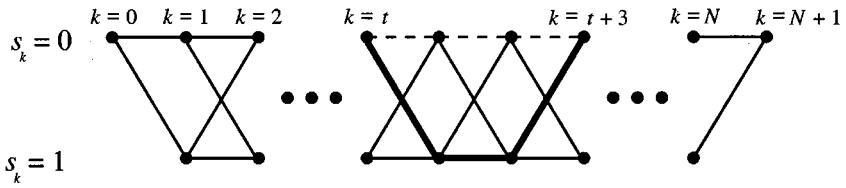

Fig. 12. Trellis diagram for a binary dicode system showing the correct path (bold line) and an adversary (dashed line) of length $l = 3$.

TABLE II
COMPARISON OF VARIOUS TRANSMITTER AND RECEIVER ARCHITECTURES FOR A 32-BIT BINARY BUS

	Fully Differential	Peak Detection	MLSD w/ Viterbi	MLSD w/ Viterbi-NC	Parallel Viterbi-NC	Serial Viterbi-NC
Relative SNR @ BER $= 10^{-8}$**:**	0.0 dB	3.0 dB	1.25 dB	0.75 dB	0.15 dB	0.15 dB
Total no. of signal paths	64	33	34	34	34	34

very general. They are compatible with either voltage or current mode drivers and the results can be extended to multi-level signals. Finally, incremental signaling can be combined with balanced bus encoding schemes described elsewhere to reduce switching noise and obtain an estimate of the common mode noise on the bus.

APPENDIX

Fig. 12 depicts the trellis diagram for a $(1 - D)$ encoded binary sequence. The state at each step, k, of the trellis is an element of the binary alphabet $\{0, 1\}$. Since the input sequence $[\, u_1 \quad u_2 \quad u_3 \quad \cdots]$ is finite in length, so is the trellis diagram. Furthermore, it is known that the initial and final states of the correct path through the trellis must both be 0. Following the approach used in [15] we seek $P_e(l)$, the probability that an adversary path from $k = t$ to $k = t + l$ which follows the state trajectory $[\, \overline{s}_t \quad \overline{s}_{t+1} \quad \cdots \quad \overline{s}_{t+l}\,]$ will be chosen over the correct path with state trajectory $[\, s_t \quad s_{t+1} \quad \cdots \quad s_{t+l}\,]$ where $\overline{s}_t = s_t$ and $\overline{s}_{t+l} = s_{t+l}$. If we define b_{ij} as the branch metric from state j to state i at step k, the metric of the adversary path minus the metric of the correct path is

$$w_l = \sum_{k=t+1}^{t+l} (b_{\overline{s}_{k-1}\overline{s}_k}(k) - b_{s_{k-1}s_k}(k)). \tag{21}$$

Assuming the Euclidean squared error of the detected sequence is to be minimized, a suitable branch metric is given by [16].

$$b_{ji}(k) = (j - i)u_k - (j - i)^2 A. \tag{22}$$

The path "closest" to the correct one will deviate from it at $k = t+1$, then run parallel to it until $k = l-1$. If only the closest adversary path is considered (a reasonable assumption for low bit error rates) all terms in the summation (21) equal zero except for the first and last. Substituting (22) into the remaining terms yields

$$\begin{aligned} w_l = {} & (\overline{s}_{t+1} - s_{t+1})u_{t+1} + (s_{t+l-1} - \overline{s}_{t+l-1})u_{t+l} \\ & - A((\overline{s}_{t+1} - s_t)^2 + (s_{t+l} - \overline{s}_{t+l-1})^2 - (s_{t+1} - s_t)^2 \\ & - (s_{t+l} - s_{t+l-1})^2). \end{aligned} \tag{23}$$

Therefore, w_l is a Gaussian random variable with an expected value of $-2A$ and $P_e(l)$ is the probability that w_l is greater than zero. For $l > 2$, the variance of w_l is

$$\begin{aligned} \text{var}[w_l] &= \text{var}[u_{t+1}] + \text{var}[u_{t+l}] \\ &= \text{var}[y_{t+2} - y_{t+1}] + \text{var}[y_{t+l+1} - y_{t+l}] \\ &= \text{var}[y_{t+2}] + \text{var}[y_{t+1}] + \text{var}[y_{t+l+1}] + \text{var}[y_{t+l}] \\ &= 4\sigma^2 \end{aligned} \tag{24}$$

and $P_e(l)$ is

$$P_e(l) = Q\left(\frac{2A}{\sqrt{4\sigma^2}}\right) = Q(\eta), \qquad l > 2. \tag{25}$$

For the special case $l = 2$, (23) simplifies to

$$\begin{aligned} w_2 = {} & (\overline{s}_{t+1} - s_{t+1})(u_{t+1} - u_{t+2}) \\ & - A((\overline{s}_{t+1} - s_t)^2 - (s_{t+1} - s_t)^2 - (s_{t+2} - s_{t+1})^2) \end{aligned} \tag{26}$$

and the variance of w_2 is

$$\begin{aligned} \text{var}[w_2] &= \text{var}[u_{t+1} - u_{t+2}] \\ &= \text{var}[(y_{t+2} - y_{t+1}) - (y_{t+1} - y_t)] \\ &= \text{var}[y_{t+2}] + \text{var}[2y_{t+1}] + \text{var}[y_t] \\ &= 6\sigma^2. \end{aligned} \tag{27}$$

Therefore, $P_e(2)$ is

$$P_e(2) = Q\left(\frac{2A}{\sqrt{6\sigma^2}}\right) = Q\left(\frac{\sqrt{6}}{3}\,\eta\right). \tag{28}$$

A union bound for the probability of a bit error can now be obtained by summing $P_e(l)$ over all possible values of l.

$$\begin{aligned} P_e &< \sum_{l=2}^{N+1} (l-1)\left(\tfrac{1}{2}\right)^{l-2} P_e(l) \\ &= P_e(2) + \sum_{l=3}^{N+1} (l-1)\left(\tfrac{1}{2}\right)^{l-2} P_e(l) \\ &= Q\left(\frac{\sqrt{6}}{3}\,\eta\right) + Q(\eta) \sum_{l=3}^{N+1} (l-1)\left(\frac{1}{2}\right)^{l-2}. \end{aligned} \tag{29}$$

The term $(l-1)$ is included because $(l-1)$ bit errors are caused by incorrectly choosing an adversary path of length l. The term $(1/2)^{l-2}$

represents the fraction of all paths of length l which have an adversary. Although the summation in (29) is finite, extending it to an infinite series has little effect on the result for any reasonable N. Hence,

$$\begin{aligned} P_e &< Q\left(\frac{\sqrt{6}}{3}\,\eta\right) + Q(\eta)\sum_{l=3}^{\infty}(l-1)\left(\frac{1}{2}\right)^{l-2} \\ &= Q\left(\frac{\sqrt{6}}{3}\,\eta\right) + 3Q(\eta) \\ &= Q\left(\frac{\sqrt{6\cdot \mathrm{SNR}}}{3}\right) + 3Q\left(\sqrt{\mathrm{SNR}}\right). \end{aligned} \tag{30}$$

REFERENCES

[1] D. R. Cecchi, M. Dina, and C. W. Preuss, "A 1 GB/S SCI line in 0.8μm BiCMOS," in *1995 ISSCC Dig. Tech. Papers*, 1995, pp. 326–327.

[2] W. G. Rivard and M. J. S. Smith, "A 1.2 μm CMOS differential I/O system capable of 400 Mbps transmission rates," in *Proc. 5th Annu. ASIC Conf. Exhibit*, Sept. 1992, pp. 427–431.

[3] J. H. Quigley, J. S. Caravella, and W. J. Neil, "Current mode transceiver logic, (Cmtl) for reduced swing CMOS, chip to chip communication," in *Proc. 6th Annu. ASIC Conf. Exhibit*, 1993, pp. 452–455.

[4] *Scalable Coherent Interface (SCI)*, IEEE Standard 1596-1992, 1992.

[5] *Low-Voltage Differential Signals (LVDS) for Scalable Coherent Interface (SCI)*, IEEE Standard 1596.3-1996, March 21, 1996.

[6] *High-Bandwidth Memory Interface Based on Scalable Coherent Interface (SCI) Signaling Technology (RamLink)*, IEEE Standard 1596.4-1996, Sept. 15, 1996.

[7] M. Tomlinson, "New automatic equalizer employing modulo arithmetic," *Electron. Lett.*, vol. 7, pp. 138–139, Mar. 1971.

[8] M. H. Shakiba, D. A. Johns, and K. W. Martin, "BiCMOS circuits for analog Viterbi decoders," *IEEE Trans. Circuits Syst. II*, vol. 45, pp. 1527–1537, Dec. 1998.

[9] G. D. Forney Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, Mar. 1973.

[10] S. A. Altekar and J. K. Wolf, "Improvements in detectors based upon colored noise," *IEEE Trans. Magn.*, vol. 34, pp. 94–97, Jan. 1998.

[11] J. D. Coker, E. Eleftheriou, F. L. Galbraith, and W. Hirt, "Noise-predictive maximum likelihood (NPML) detection," *IEEE Trans. Magn.*, vol. 34, pp. 110–117, Jan. 1998.

[12] A. Kavcic and J. M. F. Moura, "Correlation-sensitive adaptive sequence detection," *IEEE Trans. Magn.*, vol. 34, pp. 763–771, May 1998.

[13] L. G. Tallini and B. Bose, "Balanced codes with parallel encoding and decoding," *IEEE Trans. Comput.*, vol. 48, pp. 794–814, Aug. 1999.

[14] D. E. Knuth, "Efficient balanced codes," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 51–53, Jan. 1986.

[15] H. Kobayashi, "Correlative level coding and maximum-likelihood decoding," *IEEE Trans. Inform. Theory*, vol. 17, pp. 586–594, Sept. 1971.

[16] J. F. Hayes, "The Viterbi algorithm applied to digital data transmission," *IEEE Commun. Mag.*, vol. 13, pp. 15–20, Mar. 1975.

Systematic Design of High-Accuracy Current-Steering D/A Converter Macrocells for Integrated VLSI Systems

J. Vandenbussche, G. Van der Plas, W. Daems, A. Van den Bosch, G. Gielen, M. Steyaert, and W. Sansen

***Abstract*—This brief presents a systematic design methodology for digital-to-analog (D/A) converter macrocells for integrated VLSI systems. A generic behavioral model is included for system-level exploration to define the converter's specifications. The architecture and the sizes of the devices are then calculated using a performance-driven design methodology. Using a novel layout tool, the layout of the regular structures with complex connectivity, typical for D/A converters, is automatically generated. Finally, a detailed behavioral model is extracted, combining both complex dynamic behavior (glitch energy) and static behavior. A 12-bit and two 14-bit D/A converters have been designed using this approach. It is demonstrated how the applied methodology and supporting tools drastically reduce the total design time, thereby significantly increasing analog design productivity.**

***Index Terms*—Analog–digital, CMOS, data converter, high performance, mixed signal, systematic design.**

I. INTRODUCTION

In today's world, the design of heterogeneous microelectronic systems on one chip is becoming feasible due to the ever decreasing feature size of the silicon technology. These systems implement functions that require digital blocks [digital signal processing (DSP), microprocessor, RAM/ROM, etc.] as well as analog macrocells [digital-to-analog (D/A) and analog-to-digital (A/D) converters, drivers, etc.]. The use of cores and other intellectual property blocks offers the design productivity boost that is needed to generate these systems in the shortening time-to-market constraints. The design of analog functional blocks, however, requires a large amount of effort compared to digital or DSP systems, especially when large and complex blocks are considered—as, for instance, high-accuracy converters.

To tackle this problem, a number of approaches have been proposed. Analog synthesis tools promise an automated solution for the design of analog building blocks [1]. These tools use a top-down constraint-driven design methodology. Their application range is, however, limited to usually one or a few types of circuits, and only a few approaches cover the complete design flow from specification down to layout. In ASTRX/OBLX [2], asymptotic waveform evaluation (AWE) and a relaxed dc formulation is used to speed up the optimization-based sizing process. This, however, limits the applicability of the approach to linear systems. A more general approach using plain-vanilla SPICE simulations is presented in [3] but requires exuberant CPU times. In [4], [5] the high-level synthesis of the targeted class of A/D and D/A converters is covered, but the layout generation part is left to the designer. The AMGIE environment proposed in [6] automates the complete flow, from specifications over sizing synthesis down to layout, but is aimed at lower level building blocks. In [7], the complete design cycle of a

Manuscript received July 1999; revised February 2001. This work was supported in part by ESA-ESTEC. This paper was recommended by Associate Editor J.-T. Kong.

J. Vandenbussche, G. Van der Plas, A. Van den Bosch, G. Gielen, M. Steyaert, and W. Sansen are with the Department of Electrical Engineering, Katholieke Universiteit Leuven, ESAT-MICAS, Heverlee B-3001 Belgium (e-mail: jan.vandenbussche@esat.kuleuven.ac.be).

W. Daems is with the Department of Electrical Engineering, Katholieke Universiteit Leuven, ESAT-MICAS, Heverlee B-3001 Belgium and the National Fund of Scientific Research, Belgium.

Publisher Item Identifier S 1057-7130(01)04203-3.