

which is not cross-coupled is employed, r_{cm} has a similar order of magnitude to r_{dm} . This leads to a poor CMRR, unless an additional common-mode feedback loop is introduced.

r_{dm} can be further increased by slightly oversizing M_{n2} and M_{n3} compared to M_{n1} and M_{n4} . However, it depends on process tolerance, which is difficult to predict and is therefore excluded in this Letter.

Finally, the voltages on the high impedance nodes are converted again to the current by the small-signal transconductance (g'_{mn}) of M_{n5} and M_{n6} . The overall differential current gain A_{idm} and the common-mode current gain A_{icm} of the proposed circuit are:

$$A_{idm} \approx g'_{mn}(r_{dm}/r_{out}) \quad (5)$$

$$A_{icm} \approx g'_{mn}r_{cm} \quad (6)$$

Since the common-mode gain is about one if $g_{mn} \approx g'_{mn}$, the CMRR of the proposed COA is nearly equal to A_{idm} .

The dominant pole, f_p is determined by the high impedance node and equal to:

$$f_p = \frac{1}{2\pi C_p(r_{dm}/r_{out})} \quad (7)$$

where C_p is the half of the total capacitance connected to each high impedance node, including compensation capacitance if it exists. Hence, the gain-bandwidth product (GBW) of the proposed COA becomes $g'_{mn}/(2\pi C_p)$.

Simulation: A SPICE simulation has been performed using the level 28 model parameters provided for LG Semicon 0.6 μ m n-well CMOS technology with $V_{thn} = 0.771$, $V_{thp} = -0.868$, $k_n = 131.1\mu\text{A}/\text{V}^2$ and $k_p = 41.8\mu\text{A}/\text{V}^2$. The supply voltage V_{dd} is 3V and the bias current I_{bias} is 40 μ A. All the devices used have the same length of 1 μ m. The widths are: 10 μ m for M_{n1} to M_{n4} ; 20 μ m for M_{n5} to M_{n8} ; 30 μ m for M_{p1} to M_{p8} . PMOS transistors M_{p5} and M_{p6} are added to bias M_{n5} and M_{n6} , respectively. NMOS transistors M_{n5} and M_{n6} are auxiliary output devices. All body terminals of PMOS and NMOS transistors are connected to V_{dd} and ground, respectively.

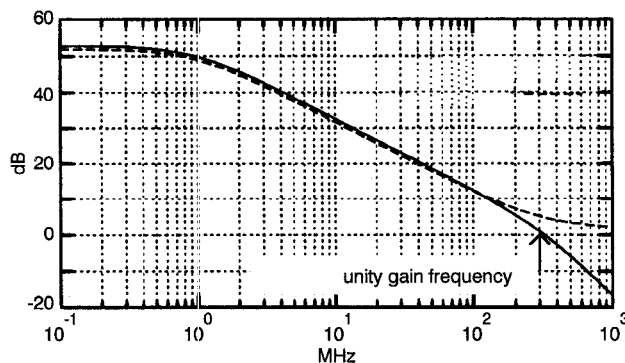


Fig. 2 Simulated differential-mode open loop gain and CMRR

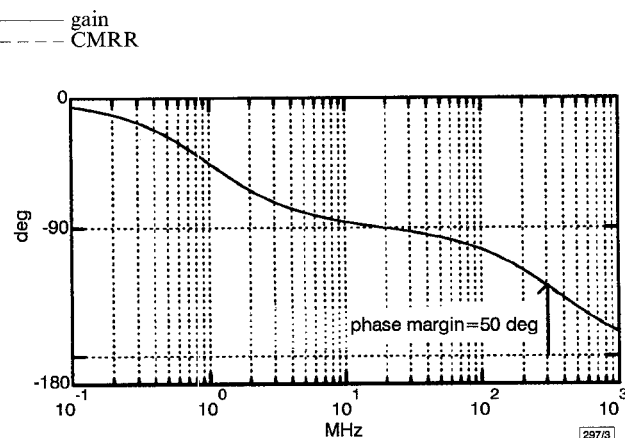


Fig. 3 Simulated phase characteristics

The frequency characteristics of the proposed COA are shown in Figs. 2 and 3. It is seen that the proposed COA exhibits an open loop gain of 53dB. The 3dB bandwidth of the COA is seen to be 1MHz. Also, the unity gain bandwidth is found to be 300MHz, at which the phase margin is 50° without inserting com-

pensation capacitors. The CMRR obtained is also shown in Fig. 2. As mentioned, the CMRR has a magnitude nearly equal to the differential-mode open loop gain. A further enhanced CMRR will be obtained if an additional common-mode feedback control is introduced. The proposed COA was simulated in a closed loop unity gain buffer. Fig. 4 shows the transient response to a step input of $\pm 10\mu\text{A}$. The 1% settling time is found to be 7.1ns. The input and output resistances of the COA are 21 and 597k Ω , respectively. Finally the static power dissipation was < 860 μ W.

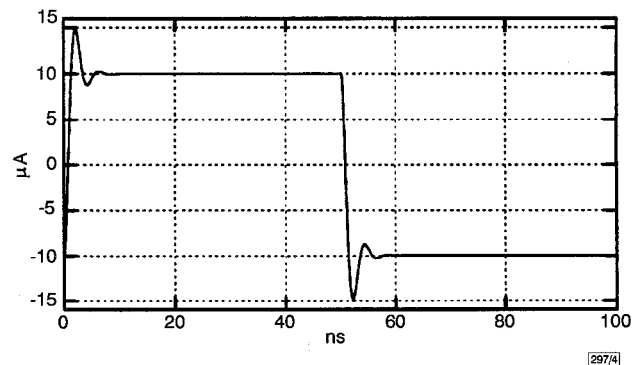


Fig. 4 Simulated transient response

Conclusions: A fully differential, wide bandwidth and good CMRR current operational amplifier has been presented. The proposed circuit employs a tapped cascode current mirror as an input stage and two cross-coupled simple current mirrors to enhance CMRR. A SPICE simulation shows a 53dB differential-mode gain, a 52dB CMRR, a 300MHz unity gain bandwidth, and a 50° phase margin without compensation.

© IEE 1998

3 November 1997

Electronics Letters Online No: 19980106

Sibum Jun and Dae Mann Kim (Department of Electrical Engineering, Pohang University of Science and Technology, San 31 Hyoja-dong, Nam-gu, Pohang, 790-784, Republic of Korea)

References

- 1 ABOU-ALLAM, E., and EL-MASRY, E.I.: 'A 200MHz steered current operational amplifier in 1.2 μ m CMOS technology', *IEEE J. Solid-State Circuits*, 1997, **32**, (2), pp. 245-249
- 2 ABOU-ALLAM, E., and EL-MASRY, E.I.: 'High CMRR CMOS current operational amplifier', *Electron. Lett.*, 1994, **30**, pp. 1042-1043
- 3 ZELE, R.H., LEE, S., and ALLSTOT, D.J.: 'A high gain current-mode operational amplifier'. Proc. IEEE ISCAS, 1992, pp. 2852-2855
- 4 KAULBERG, T.: 'A CMOS current-mode operational amplifier', *IEEE J. Solid-State Circuits*, 1993, **28**, pp. 849-852
- 5 TOUMAZOU, C., LIDGEY, F.J., and JAIGH, D.J.: 'Analogue IC design: The current-mode approach' (Peregrinus, London, 1990)

Design of turbo-code interleaver using Hungarian method

A.K. Khandani

A method is presented for optimising the structure of the turbo-code interleaver using the Hungarian method (linear sum assignment problem). Numerical results are presented which show a substantial improvement with respect to a random interleaver.

Introduction: Consider a turbo code in which the underlying recursive convolutional code (RCC) is generated by the transfer function $G(d) = N(d)/D(d)$. We know that the impulse response of $G(d)$ is periodic with period $p \leq 2^r - 1$, where r is the constraint length of the code [1]. If $p = 2^r - 1$, the resulting impulse response is called a maximum length sequence (MLS). This is the case for the transfer functions used in turbo codes.

If we look at the impulse response of $G(d)$ as a periodic sequence, we obtain $K = 2^r - 1$ non-zero sequences which are time shifts of each other. We refer to these sequences as different

phases of the periodic signal. It can be shown that the set of phases of an MLS (plus the all-zero sequence) constitute a group under binary addition [1]. The order of each element in this group is two, meaning that the sum of each phase with itself results in the all-zero sequence (denoted as the zero phase).

Consider the situation that a given RCC encoder is excited by a random binary sequence composed of w impulses located at time instants t_{i_1}, \dots, t_{i_w} where $i_1 < \dots < i_w$, $w \geq 2$. This corresponds to an information sequence of weight w at the input. We refer to $[t_{i_1}, t_{i_w}]$ as the non-zero span, and to t_{i_w} as the terminating edge of the sequence. Note that if two such sequences are added, then the terminating phase corresponding to their sum is obtained by adding the terminating phases of the two sequences. Also note that changing the position of some of the impulses by some multiples of the period does not change the terminating phase of the output sequence.

The role of the interleaver is to modify the pattern of those low-weight input sequences which have resulted in a low weight output after passing through the first RCC, such that the output weight of the subsequent RCCs is high. In the design of the interleaver, we have the following three implicit objectives in mind: (i) breaking the low-weight input sequences such that a terminating zero-phase is avoided in at least one of the outputs, (ii) providing a large span for the input sequences which have resulted in a zero terminating phase in all the component codes, and (iii) providing a large distance for the terminating edge of the sequences from the right edge of the corresponding block in at least one of the encoders.

Breaking of the weight-two input sequences: The design of the interleaver in turbo codes has been usually concentrated around the task of breaking the weight-two input sequences. In the following, we design an interleaver which is in a sense optimum with respect to this task. Specifically, we conclude that there exists an interleaver which breaks all the weight-two sequences with a span limited to K^2 , while there does not exist an interleaver which breaks all the weight-two sequences of a larger span.

We refer to time positions within a given block which are congruent to i modulo K as C_i . If the system is already in phase a , then an impulse at position $t \in C_i$ will result in phase $b = a \oplus i$ at the output of the corresponding RCC, where \oplus denotes the group addition of the phases (an impulse at position $t \in C_a$ will result in the zero phase). For the sake of simplicity, we assume that the cardinality of C_i values is an integer multiple of K , or equivalently, N is an integer multiple of K^2 .

In the process of interleaving, if two elements within a given C_i , $i = 1, \dots, K$, are mapped into two positions within a given C_j , then these positions constitute a weight-two, zero-phase sequence which remains zero-phase after interleaving. The interleaver can force the elements mapped to different positions of a given C_i , $i = 1, \dots, K$, to be of different C_j values iff the number of positions in each C_i is $< K$. This in turn means that the block length N should satisfy $N < K^2$. In practice, $N > K^2$ and the interleaver will not be able to break all the weight-two sequences. Noting the periodicity property, it is easy to see that the fraction of the unbroken weight-two sequences within a given C_i is at least $1/K$. The interleaver should be designed such that the span of such unbroken sequences is maximised. The value of the minimum span is upper-bounded by K^2 (which is achieved if we sample each C_i by an interval of K into K subsets and leave the weight-two sequences within each of these subsets unbroken). The structure of the interleaver achieving these features is very simple. It suffices to partition the input block into sub-blocks of length K and apply a cyclic shift of i positions, $i = 0, 1, \dots$, to the elements of the i th sub-block. Obviously, the effective number of cyclic shifts applied to the i th sub-block is equal to: $i \pmod{K}$. In this case, after K sub-blocks, we come back to our original point where no cyclic shift is performed. The same procedure repeats in a periodic manner. We refer to this structure as a *uniform interleaver*. Since our simulation results show that the corresponding interleaver has a poor performance, we conclude that it is not sufficient to consider only the effect of the weight-two input sequences.

As in a uniform interleaver, the minimum span of the unbroken weight-two sequences is quite large, the performance is not determined by the weight-two sequences. This means that we can permute the elements mapped to each C_i in the interleaved block

without being concerned about the interaction of these elements with themselves (in producing low-weight error events). In this case, the performance of the code is determined by the interaction between the elements located in different values of C_i , and also by the right margin (distance to the right edge of the block) of the elements. In the following, we explain a combinatorial optimisation technique which attempts to optimise these quantities.

Optimising the interleaver using Hungarian method: We define a distance measure between the elements of each C_i with respect to the rest of the block and then permute the elements of the C_i values (one C_i at a time) such that this distance measure is minimised. Specifically, we define the distance between two elements to be unity, if the separation between them plus the separation between their permuted versions is less than a given threshold. To incorporate the edge effects, we assign a separate distance to each element which is set equal to unity if the right margin of the element plus the right margin of its interleaved version is less than a given threshold.

The overall distance of a given C_i , denoted as D_i , is defined as the sum of the distances of its elements. The optimisation procedure permutes the elements originally mapped to a given C_i to minimise this distance. In practice, we order the C_i values according to the value of their D_i and select the first one (with the maximum D_i) for readjustment. If this readjustment decreases the value of the corresponding D_i , we start a new iteration, and if it does not change the value of D_i , then we proceed to the next C_i in the list.

It remains for us to find a method to permute the elements of each C_i to minimise the relevant D_i . This is achieved by considering that if a given position, say A , within C_i is assigned to another position, say B , then the change in the value of D_i depends only on A and B . This property results in a *linear objective function* for our assignment problem. To formulate the problem, we consider a set of zero-one variables, say δ_m^n where $\delta_m^n = 1$, if the m th position of C_i is assigned to its n th position. If the contribution of this assignment to the new value of D_i is denoted by $\delta_m^n(i)$, we can formulate the optimisation procedure as

$$\begin{aligned} \text{Minimise} \quad & D_i = \sum_{m=1}^N \sum_{n=1}^N \delta_m^n d_m^n(i) \\ \text{Subject to:} \quad & \sum_{m=1}^N \delta_m^n = 1 \quad \sum_{n=1}^N \delta_m^n = 1 \quad \text{and} \quad \delta_m^n \in \{0, 1\} \\ & \forall m, n \quad (1) \end{aligned}$$

Note that the constraints $\sum_m \delta_m^n = 1$ and $\sum_n \delta_m^n = 1$ reflect the permutation structure of the interleaver. This problem is known in the context of discrete optimisation as a linear sum assignment problem and can be solved very efficiently using the Hungarian method [2].

The key point behind obtaining a linear objective function, and consequently an easy solution, is that here we are not entering the interaction among the elements within each C_i into our computations. In general, incorporating such a mutual interaction between making decisions in a discrete optimisation problem (assignment in the present case) makes the problem substantially more complicated. As an example, the simplest form for inclusion of such mutual interaction would be to consider the interactions between pairs of decisions, which results in the so-called quadratic assignment problem (QAP) [3]. In a QAP, the objective function would involve the product of all pairs of δ_m^n values. It is worth mentioning that the problem in eqn. 1 can easily be solved over sets of cardinality in the order of few thousand, while the solution of a QAP over a set of cardinality more than ~ 20 is a challenging problem.

More generally, if we attempt to incorporate the effect of the higher weight input sequences, say weight w , into our optimisation routine, then the resulting assignment problem would be of degree w (involving the product of all possible combinations of δ_m^n values composed of w components). Such problems are generally extremely difficult to solve, and their solution methods are mainly heuristic (e.g. simulated annealing), or deterministic algorithms based on an exchange of pairs or triples.

As already mentioned, there is no strong motivation to attempt to design the interleaver based on the objective of breaking the zero-phase sequences. Based on this observation, we have also examined the application of the optimisation problem in eqn. 1 to

the case that the optimisation procedure starts from an initial interleaver which is an identity transformation (each element is mapped to itself). The interesting observation is that in this case, the results obtained are relatively close to the results obtained by starting from a uniform interleaver. In the following, we present some numerical results for the application of the optimisation procedure in eqn. 1, starting from a uniform and also from an identity interleaver.

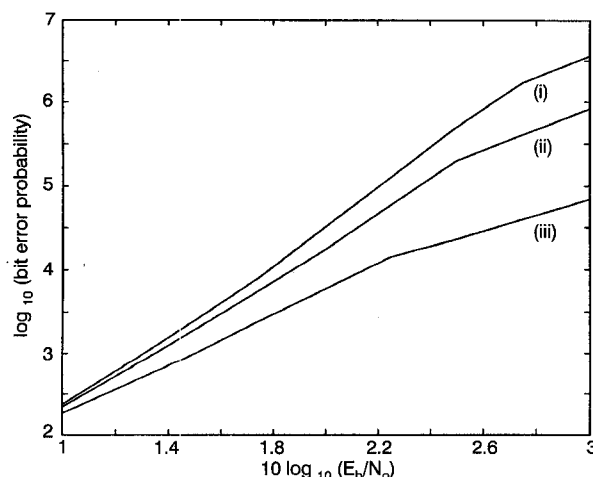


Fig. 1 Bit error performance of random optimised interleavers

- (i) optimised interleaver with uniform initialisation
- (ii) optimised interleaver with identity initialisation
- (iii) random interleaver

We have implemented the algorithm given in eqn. 1 for a simple rate 1/3 turbo-code, where the underlying RCCs are of constraint length $r = 3$ (period of $K = 2^r - 1 = 7$) and have the generator polynomial (eqns. 5 and 7). The code block length is equal to 196 and the number of iterations is 20. Fig. 1 shows the bit error probability of a random interleaver against that of an optimised interleaver. In the optimisation procedure, the threshold T is selected as $T = 14$ (simply as two times the period) and the distances to the end of the blocks are weighted by a relative factor of two. The reason for the selection of a simple code and a short block length is because we have been mainly interested in wireless applications where the delay is limited to 20ms (a delay of 20ms corresponds to a block length of 196 at a bit rate of 9.6K).

Acknowledgment: This work has been supported by the Information Technology Research Centre of Ontario (ITRC).

© IEE 1998

10 September 1997

Electronics Letters Online No: 19980027

A.K. Khandani (Department of Electrical and Computer Engineering, University of Waterloo, ON, N2L 3G1, Canada)

References

- GOLOMB, S.W.: 'Shift register sequences' (Holden-Day, San Francisco, 1967)
- KREKÓ, B.: 'Linear programming', translated by J.H.L. Ahrens and C.M. Safe (Sir Isaac Pitman & Sons Ltd., 1968)
- BURKARD, R.E.: 'Locations with spatial interactions: the quadratic assignment problem' in MIRCHANDANI, P.B., and FRANCIS, R.L. (Eds.); 'Discrete location theory' (John Wiley, 1991)

Improved upper bound on bit error rate for truncated convolutional codes

Hichan Moon

The author presents the performance of convolutional codes truncated by input information of finite length. In a convolutional code, the error rates of each bit are different. The upper bound on the error rate for each bit within a convolutional code is derived and analysed.

Introduction: The performance of a convolutional code can be estimated from the upper bound on the bit error rate (BER) [1]. This bound is derived assuming that the input information sequence is infinitely long. The bound is accurate on channels with moderate to large signal-to-noise ratios, if the input information sequence is long enough. In real applications, the convolutional code is truncated by input information of finite length [2]. The conventional method for truncation is to encode an information message of fixed length L followed by m additional all zero inputs, where $m+1$ is the constraint length of the convolutional code. When the input information sequence is short, the conventional bound becomes loose. Moreover, the error rates of each information bit are different within a convolutional code, depending on the bit position. The error rate of the first or the last position bit is much lower than that of centrally positioned bit. Because of this non-uniform BER property, it is usual to place important bits at the beginning and end of a convolutional code. In this case, it is necessary to know the performance of each information bit.

In this Letter, the upper bound on error rate for each information bit is derived and analysed for convolutional codes truncated by input information of finite length L .

Upper bounds: The BER P_b of a binary rate k/n convolutional code is bounded as

$$P_b \leq \sum_{d=d_{min}}^{\infty} \frac{1}{k} \cdot A(d) \cdot P_d \quad (1)$$

where P_d is the probability that a path at distance d from the transmitted sequence is chosen in decoder; d_{min} is the free distance of the convolutional code, and $A(d)$ is the distance spectrum of the convolutional code [3]. If a binary code symbol is transmitted by binary phase shift keying (BPSK) over an additive white Gaussian noise channel with noise spectral density $N_0/2$, then P_d is given as

$$P_d = Q\left(\sqrt{\frac{2E \cdot d}{N_0}}\right) \quad (2)$$

where E is the energy per coded symbol. But, this bound becomes loose when the code is truncated by short input information sequence. Moreover, this conventional bound cannot explain the dependence of a code's BER on bit position.

To explain the different error rate of each bit in a convolutional code, it is necessary to obtain the BER upper bound for each bit position, respectively. Let $P_b^{(i)}$ denote the error rate of i th information bit in a convolutional code. The upper bound on $P_b^{(i)}$ is proposed as

$$P_b^{(i)} \leq \sum_{d=d_{min}}^{\infty} A_i(d) \cdot P_d \quad (3)$$

where $A_i(d)$ is the number of paths, at distance d from the transmitted sequence, that cause error in the i th information bit.

Number of error paths: Suppose the encoder of a binary rate $1/n$ convolutional code has m memory elements and the code is truncated by an information message of length L . A generalisation for rate k/n codes can easily be made. The encoder has its state $S = (s_1, s_2, \dots, s_m)$, where s_j is the content of the j th memory element of the encoder. Let S_t denote the state at trellis depth t . Without loss of generality, it can be assumed that all the zero sequence is transmitted.

Consider the error rate of the i th information bit. Paths which cause error in the i th information bit diverge from the all-zero state before trellis depth $i-1$, merge to the all-zero state after trellis depth i , and have a non-zero information bit between trellis depth $i-1$ and i . To compute $A_i(d)$, it is necessary to obtain the modified trellis diagram for i th information bit. The modified trellis diagram for the i th information bit can be obtained by modifying the trellis diagram as follows: (i) Let the all-zero state at trellis depth 0 be the starting point, and the all-zero state at the end of the trellis be the end point. Draw an arrow in each branch from each trellis depth to next trellis depth. (ii) Before trellis depth $i-1$, delete all branches which merge into the all-zero state. After trellis depth i , delete all branches which diverge from the all-zero state. If all branches merging to a node are deleted, delete the node and all branches diverging from that node. (iii) Label the branches which are between trellis depth $i-1$ and i with powers of indeterminate x