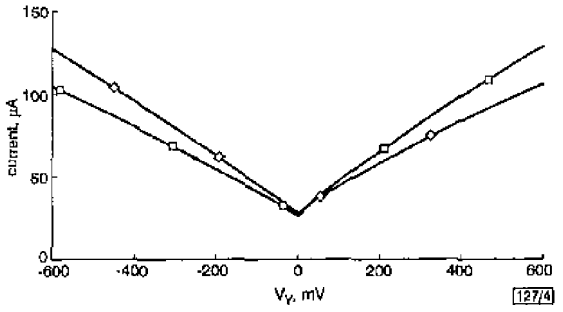

Fig. 4 *Current driven from supplies to CCII*

□ I (positive supply)
◇ I (negative supply)

Table 1: W/L of transistors of circuit shown in Fig. 2

Transistor	Aspect ratio
M1–M2, M5–M6	10/1
M3–M4, M7–M8	2/1
M9–M10	11.5/2.5
M11–M12	1/0.5
M13	10/2.5
M14	6/1.5
M15–M16	72/1.5
M17	60/1.5
M18–M19	6/1.5
M20	5/1.5
M21–M23	11.5/2.5
M24	23/2.5

PSPICE simulations: The performances of the CCII shown in Fig. 2 are simulated using PSPICE. Transistor aspect ratios are given in Table 1 and a 0.5μm MIETEC CMOS process is assumed. The supply voltages are $V_{DD} = -V_{SS} = 1.5$V, and I_{B1} is set to 6μA. Table 2 shows a summary of the results of the simulations.

Table 2: Summary of simulation results

THD	50dB
Input resistance R_x	2Ω
Open-loop gain	42dB
3dB frequency (voltage amplifier topology)	30MHz
Standby current	32μA
Voltage offset (from Y to X)	0.05mV
Current offset (from X to Z+)	0.8μA

Fig. 3 shows the voltage transfer characteristics from node Y to node X and to node Z+ when node X is terminated by $R_x = 40$kΩ and $R_z = 80$kΩ. V_Y is set to 0.5Vp-p.

Fig. 4 shows the current driven from the supplies when the voltage V_Y is scanned from -0.6V to 0.6V with the same R_x and R_z. It is clear that, when node Y is at ground, the standby power consumption is very low.

Conclusion: In this Letter, the novel realisation of a CCII has been introduced. Simulation results show that the circuit exhibits excellent performance while consuming low standby power, which makes it suitable for low-power analogue signal processing.

26 October 1999
Electronics Letters Online No: 20000129
DOI: 10.1049/el:20000129

A.M. Ismail and A.M. Soliman (*Electronics and Communications Engineering Department, Cairo University, Giza, Egypt*)

References

1 ELWAN, H.O., and SOLIMAN, A.M.: 'Low-voltage low-power CMOS current conveyors', *IEEE Trans. Circuits Syst. I*, 1997, **44**, pp. 828–835
2 SEDRA, A.S., ROBERTS, G.W., and GOHH, F.: 'The current conveyor: history, progress and new results', *IEE Proc., Circuits Devices Syst.*, 1990, **137**, pp. 78–87
3 TOUMAZOU, C., LIDGEY, F.J., and MAKRIS, C.A.: 'Extending voltage-mode op-amps to current-mode performance', *IEE Proc., Circuits Devices Syst.*, 1990, **137**, pp. 116–130
4 ISMAIL, A.M., and SOLIMAN, A.M.: 'Wideband CMOS current conveyor', *Electron. Lett.*, 1998, **34**, pp. 2368–2369
5 PALMISANO, G., and PALUMBO, G.: 'A simple CMOS CCII+', *Int. J. Circuit Theory Appl.*, 1995, **23**, pp. 599–603
6 SURAKAMPONTORN, W., RIEWRUJA, V., KUMWACHARA, K., and DEJHAN, K.: 'Accurate CMOS-based current conveyors', *IEEE Trans. Instrum. Meas.*, 1991, **40**, pp. 699–702
7 SISKOS, S., VLASSIS, S., and PITAS, I.: 'Analog implementation of fast min/max filtering', *IEEE Trans. Circuits Syst. II*, 1998, **45**, pp. 913–918

Realisation of exponential V-I converter using composite NMOS transistors

Weihsing Liu, Cheng-Chieh Chang and Shen-Iuan Liu

A CMOS voltage-to-current converter with exponential characteristics is presented. The Taylor's series expansion is used for realising the exponential function. In a 0.35μm CMOS process, the HSPICE simulation results show a 15dB linear range with a linearity error of < ±0.5dB. The total power consumption is < 0.8mW with ±1.5V supply voltage. The circuit can be used in the design of a variable gain amplifier (VGA).

Introduction: Since there is no intrinsic logarithmic MOS device operating in the saturation region for CMOS technologies, one method to generate the exponential characteristics is by use of a 'pseudo-exponential' generator [1, 2]. Alternatively, the Taylor's series expansion can also be used for implementation of the exponential [3]. According to the Taylor's series expansion, a general exponential function can be expressed as

$$\exp(ax) \simeq 1 + \frac{a}{1!}x + \frac{a^2}{2!}x^2 + \frac{a^3}{3!}x^3 + \cdots + \frac{a^n}{n!}x^n + \cdots \quad (1)$$

where a is the coefficient and x is the independent variable; if $|ax| \ll 1$, the higher order terms of eqn. 1 can be neglected and the final approximation equation can be given as [3]

$$\exp(ax) \simeq 1 + \frac{a}{1!}x + \frac{a^2}{2!}x^2 \quad (2)$$

Eqn. 2 can be implemented by the composition of a V-I squarer circuit, a linear V-I converter and a constant bias current which are equivalent to the second-order, the first-order and the zero-order terms in eqn. 2 [3]. In this Letter, a simple and universal approach is presented to realise eqn. 2. Simulation results will be given to verify the validity of the approach.

Circuit implementation: Consider the composite NMOS transistor [4] shown in Fig. 1. Assuming that transistors M1 and M2 are identical and that both of them operate in the saturation region without body effects, the following equation applies:

$$Id1 + Id2 = \frac{Kn}{2}(Vin - Vx - Vtn)^2 + \frac{Kn}{2}(Vx - Vc - Vtn)^2 \quad (3)$$

where $Id1$ and $Id2$ are the drain currents of transistors M1 and M2, respectively, $Kn = \mu_n Cox$ is the process parameter and Vtn is the threshold voltage. According to eqn. 3, if $Vx = Vc/2$, then eqn. 3 can be rewritten as

$$Id1 + Id2 = Kn\left(\frac{-Vc}{2} - Vtn\right)^2\left(1 + \frac{Vin}{\left(\frac{-Vc}{2} - Vtn\right)} + \frac{Vin^2}{2\left(\frac{-Vc}{2} - Vtn\right)^2}\right) \quad (4)$$

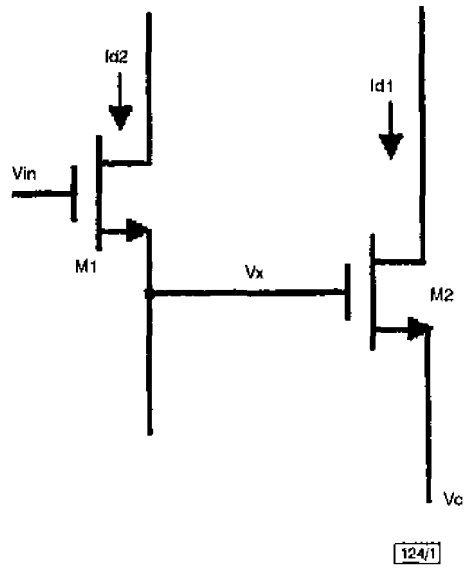

Fig. 1 *Composite NMOS transistor*

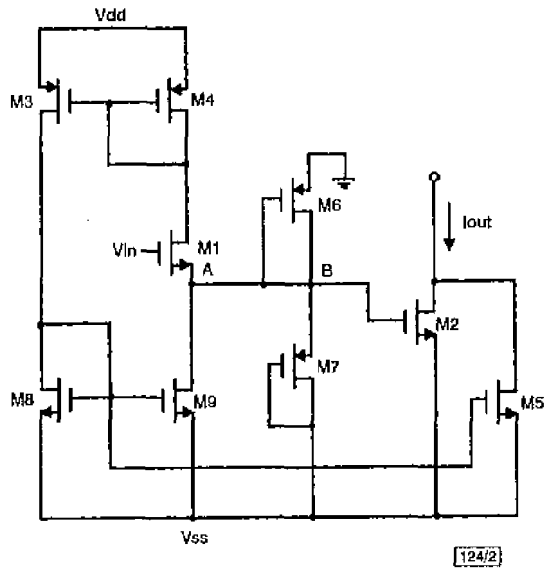

Fig. 2 *Proposed composite NMOS transistor exponential V-I converter*

The proposed exponential V-I converter can be realised as shown in Fig. 2, the voltages Vc and Vx equal to the negative supply voltage, Vss, and the gate voltage of M2, respectively. The transistors M8, M9, M3 and M4 are two-current mirrors. Since $Id4 = Id1 = Id9$, no current will flow from node A to node B, and transistors M6 and M7 will force the gate voltage of M2 to half of the supply voltage, Vss, (i.e. $Vx = Vss/2$). If $x = Vin$, $a = 2/(Vss - 2Vtn)$, $Ib = Kn(-Vss - 2Vtn)^2/4$ and, according to eqn. 4, the output current of Fig. 2 can be given as

$$\begin{aligned} Iout &= (Id1 + Id2) \\ &= Kn\left(\frac{-Vss}{2} - Vtn\right)^2\left(1+\frac{Vin}{\left(\frac{-Vss}{2}-Vtn\right)}+\frac{Vin^2}{2\left(\frac{-Vss}{2}-Vtn\right)^2}\right) \\ &= Ib\left(1+ax+\frac{a^2x^2}{2}\right) \simeq Ib\exp(ax) \end{aligned} \quad (5)$$

The threshold voltage Vtn of the NMOS transistor is ~0.4V in our process. If $|Vss > 2Vtn| = 0.8$ V, then $(-Vss - 2Vtn) > 0$ and all the requirements in the proposed approach can be satisfied. For the proposed converter to operate properly, all of the transistors should be biased in the saturation region. Theoretically, the input range of this circuit could be

$$\frac{Vss}{2} + Vtn1 \le Vin \le Vdd - |Vtp4| + Vtn1 \quad (6)$$

where $Vtn1$ and $Vtp4$ are the threshold voltages of transistors M1 and M4. With $Vtn1 = 0.4$V and $Vtp4 = -0.55$V, the input range is between -0.35 and 1.35V.

Simulation results: HSPICE simulation results are shown in Fig. 3 where a 0.35μm CMOS process is used. The aspect ratios of the transistors in Fig. 2 are listed in Table 1. As Vin varies from –0.3 to 0.6V, the pseudo-exponential function achieves 15dB with a linearity error of $< \pm 0.5$dB. When $Vin > 0.6$V, the simulation result deviates from the ideal line due to neglecting the higher order terms in eqn. 1. If the threshold voltages of transistors M1 and M2 in Fig. 1 are not equal (i.e. $Vtn1 \neq Vtn1$, $Vtn1 = Vtn2 + \Delta Vtn$), eqn. 4 can be modified to read

$$\begin{aligned} &Id1 + Id2 = \\ &Kn\left(\frac{-Vc}{2}-Vtn1\right)^2\left(1+\frac{Vin}{\left(\frac{-Vc}{2}-Vtn1\right)}+\frac{Vin^2}{2\left(\frac{-Vc}{2}-Vtn1\right)^2}\right) \\ &+ Kn\Delta Vtn\left(\frac{\Delta Vtn}{2}-\frac{Vc}{2}-Vtn1\right) \end{aligned} \quad (7)$$

where an additional constant current term is generated, and it will only result in the offset current with respect to the pseudo-exponential function.

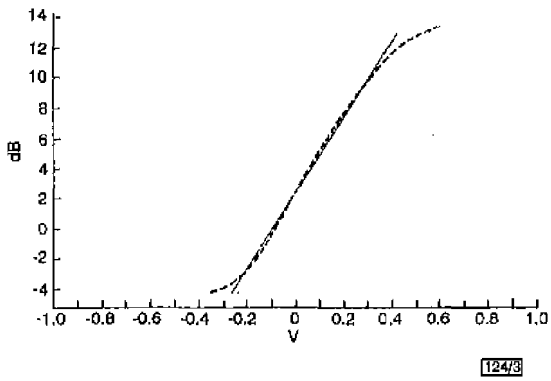

Fig. 3 *Simulation results, Iout against Vin, (with normalised dB scale)*

ideal
proposed

Table 1: Aspect ratios of transistors of circuit in Fig. 2

Transistors	Aspect ratios
	μm/μm
M1, M2, M8, M9	1/1
M6, M7	100/1
M3, M4, M5	1/1

Conclusions: A new composite NMOS transistor based exponential V-I converter has been presented. This exponential V-I converter is compact and power efficient. The proposed circuit could be of use in the design of VGAs [5].

26 October 1999
Electronics Letters Online No: 20000051
DOI: 10.1049/el:20000051

Weihsing Liu, Cheng-Chieh Chang and Shen-Iuan Liu (*Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan 10617, Republic of China*)

E-mail: lsi@cc.ee.ntu.edu.tw

References

1 MOTAMED, A., and ISMAIL, M.: 'CMOS exponential current-to-voltage converter', *Electron. Lett.*, 1997, **33**, (12), pp. 998–1000

2 HARJANI, R.: 'A low-power CMOS VGA for 50Mb/s disk drive read channels', *IEEE Trans. Circuits Syst. II*, 1995, **42**, (6), pp. 370–376

3 LIN, C., PIMENTA, T., and ISMAIL, M.: 'Universal exponential function implementation using highly-linear CMOS V-I converters for dB-linear (AGC) applications'. Proc. 1998 IEEE Midwest Symp. Circuits and Systems, 1999, pp. 360–363

4 BULT, K., and WALLINGA, H.: 'A class of analog CMOS circuits based on the square law characteristics of an MOS transistor in saturation', *IEEE J Solid-State Circuits*, 1987, **SC-22**, pp. 357–365
5 MOTAMED, A., HWANG, C., and ISMAIL, M.: 'A low-voltage low-power wide range CMOS variable gain amplifier', *IEEE Trans. Circuits Syst. II*, 1998, **45**, (7), pp. 800–811

Analytical prediction of buffer hit ratios

Jongmoo Choi, Seongje Cho, Sam H. Noh, Sang Lyul Min and Yookun Cho

An analytical model for the prediction of buffer hit ratios is presented. Prediction is carried out by means of the on-line detection of block reference patterns of applications and exploitation of functions that approximate the hit ratios for such detected patterns. Experimental results show that the proposed method accurately models the hit ratios of actual executions.

Introduction: The analytical prediction of buffer hit ratios is very useful in making informed operating system decisions such as when allocating buffer space to various applications. To this end, the study of methods for predicting the expected buffer hit ratio of applications has been the focus of much research both in the database and operating system fields [1 – 4]. In this Letter, we propose a new method for the analytical prediction of buffer hit ratios that is based on the characteristics of the block reference patterns of applications. The method first detects the block reference patterns of the applications on-line (i.e. during execution). Based on the detected result, it then predicts the expected buffer hit ratio using an analytical buffer model. The main focus of this Letter is on the second stage of this method, which employs previously known approximation methods. Such approximation methods have not been widely used in practice since there has been no reliable technique for detecting the reference patterns of the applications.

Detection of block reference patterns: We have proposed a block reference pattern detection technique where the pattern is detected by associating block attributes such as the backward distance and frequency with the forward distance of a block [5]. This is done on-line, allowing it to detect dynamic reference pattern changes within the applications. Four reference patterns, namely sequential, looping, temporally clustered and probabilistic patterns, are detected. (The characteristics of these patterns are elaborated on below.) It has been observed that the majority of reference patterns of applications that make use of the buffer cache fall into one of these categories [2, 6]. To optimise the hit ratio, we can apply different block replacement methods to each application according to the detected reference pattern. For sequential and looping patterns, we apply the most frequently used (MRU) replacement method, while for temporally clustered and probabilistic patterns, the least used (LRU) and least frequently used (LFU) methods are, respectively, applied. These choices are due to the optimality of the method under such reference patterns [2, 7].

Analytical buffer model: In this Section, we show how to predict the hit ratio for applications that show the aforementioned reference patterns. A sequential reference pattern is a reference pattern where all blocks are accessed one after another and never re-accessed. Formally, a reference $<r_1, r_2, ..., r_w>$ is sequential if $r_i \neq r_j$ for all $1 \leq i, j \leq w$. The expected buffer hit ratio of this pattern is therefore $HIT_{seq}(B) = 0$, where B is the number of buffer blocks allocated to the pattern, irrespective of which replacement method is used. A looping reference pattern is a reference pattern where all blocks are accessed repeatedly with a regular interval. Formally, a reference $<r_1, r_2, ..., r_w>$ is a looping reference if for some $l \leq w$, $r_i \neq r_j$ for all $1 \leq i, j \leq l$, and $r_{i+l} = r_i$ for $1 \leq i \leq w - l$. The subsequent $<r_1, r_2, ..., r_l>$ is called a loop, and l is the length of the loop. If we apply the MRU method to the looping pattern, the expected buffer hit ratio of this pattern is $HIT_{loop}(B) = (\min[l, B] \times (w/l))/w = \min[l, B]/l$. A temporally clustered reference pattern is a reference pattern where blocks accessed more recently are the ones more likely to be accessed in the future. This reference pattern can be characterised by the LRU stack model [7, 8]. In the model, all blocks are ordered by their last reference time in the LRU stack. When a block is accessed, it moves to the top position of the stack (position 1) and the blocks that were above the accessed block are pushed down one position (from position i to position $i + 1$) to make room for the accessed block. Each position i of the stack has a reference probability a_i. Hence, the block located in position i will be accessed with probability a_i. When a reference satisfies $a_1 \geq a_2 \geq ... \geq a_B$, it is a temporally clustered reference pattern. If we apply the LRU method to this reference pattern, the expected buffer hit ratio is $HIT_{lc}(B) = \sum_{i=1}^{B} a_i$. The question now is how to obtain the a_i values. This may be possible by observing the hit counts of the stack position i and dividing by the length of reference w using ghost buffers [3]. However, measuring the hit counts of all stack positions individually is impractical due to its overhead. Hence, we devise an efficient approximation method that utilises Belady's lifetime function. We use Belady's lifetime function as it is well known that the function accurately approximates the buffer hit ratio of references that show the temporally clustered pattern [8]. Belady's lifetime function, A_i, which represents the buffer hit ratio with buffer size i, is given as

$$A_i = a_1 + a_2 + \cdots + a_i = 1 - c \times i^{-k} \qquad (1)$$

where c and k are control parameters. The control parameters determine the degree of temporal locality; as c becomes smaller or as k becomes larger, the degree of temporal locality increases.

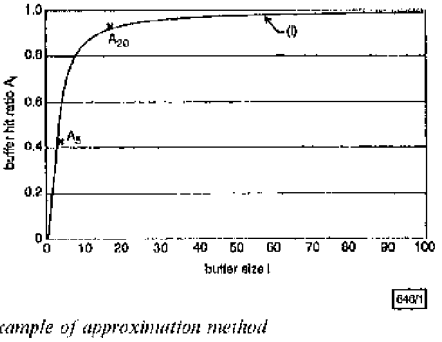

Fig. 1 *Example of approximation method*
(i) Belady's lifetime function

Using Fig. 1, we show how the approximation method works. Assume we have measured two buffer hit ratios A_5 and A_{20}, depicted as cross points in the Figure. Using these values and eqn. 1, we can calculate the values of c and k. With these values, we can estimate the value of a_i for all i as $a_i = A_i - A_{i-1}$. Hence, we can predict $HIT_{lc}(B)$ for all B. As the number of measured A_is increases, the degree of accuracy of the approximation also increases. In our experiments, the results of which we show later, we used four measurements (A_{10}, A_{20}, A_{50}, and A_{100}). We then computed the c and k values with each pair of A_is. Finally, we set c and k to the average of these computed values. A probabilistic reference pattern is a reference pattern where each block has a stationary reference probability and all blocks are accessed independently with the associated probabilities. The probabilistic pattern can be characterised by the independent reference model [7]. In this model, each block b_i has its stationary and independent reference probability p_i. The expected buffer hit ratio is then $HIT_{prob}(B) = \sum_{i=1}^{B} p_i$ (assuming $p_i \geq p_j$ for $i \leq j$). Similarly to the temporally clustered pattern, we use an approximation method to estimate p_i for all i. For the probabilistic reference pattern, it is well known that the Zipfian probability distribution function provides an accurate approximation for the buffer hit ratio [7].

The Zipfian probability distribution function, P_i, which represents the buffer hit ratio with buffer size i, is given as

$$P_i = p_1 + p_2 + \cdots p_i = (i/n)^{\log \alpha / \log \beta} \qquad (2)$$

where n is the total number of accessed blocks, and control parameters α and β are interpreted as follows: of the total references, α fraction of the accesses are made to the β fraction of the total n blocks. Using a similar calculating process as the temporally clustered pattern, we can calculate the values of α and β; specifically, by solving eqn. 2 with measured P_i values. Note that n is