

Signed higher-radix full-adder algorithm and implementation with current-mode multi-valued logic circuits

T. Temel, A. Morgul and N. Aydin

Abstract: A novel algorithm for full-addition of two signed, higher-radix numbers is proposed and implemented by combining multi-valued logic min, max, literal and cyclic operators. Owing to disjoint terms involved, multi-valued logic min and max operators are replaced with ordinary transmission operation and sum, respectively. A multi-valued logic cyclic gate is designed by using a current-mode threshold circuit while the literal is realised by only voltage-mode switching circuits. The threshold circuit employed within the cyclic gate exhibits improved dynamic behaviour compared to its previous counterparts employing voltage-mode binary logic switching circuits. It also allows much higher radices compared to previous current-mode threshold circuits owing to its superior mismatch properties. Thus, the cyclic gate achieves a superior performance compared to its predecessors. As a direct extension to cyclic operation in radix-8, a resultant single-digit, radix-8 full-adder and its 3-bit counterpart voltage-mode circuits are designed and their performance compared. It is shown that the developed signed addition algorithm can be realised by using the proposed full-adder. Finally, the algorithm is also exploited for a multi-digit case. Simulation results demonstrate that proposed architectures can be used in high-performance arithmetic units.

1 Introduction

There have been well-known architectures and algorithms in two-valued binary logic [1]. However, higher-radix design techniques can be referred to as a more generic means for achieving theoretically less interconnected architectures [2, 3]. The main obstacles in developing higher-radix systems are the lack of a proper algebra and difficulty in realisation. The former involves highly intelligent mathematical derivations and relevant logic simplification procedures [4], while the latter requires novel circuit design techniques. In literature, various algebraic forms for representing a higher-radix system have been devised [4–6]. It still remains a problematic area to contrive a suitable algorithm and associated simplification technique in higher-radix computation. Most literature studies yield very complicated expressions. Furthermore, realisation relies mostly on grouping the binary digits in a 2's power radix such that conventional binary circuits can be utilised [7]. When a non-prime radix is chosen, simplification becomes very prohibitive. Furthermore, the intense voltage-mode switching activities involved in binary logic operations result in unavoidably large power dissipation at higher frequencies owing to increased interconnectivity and large switching voltage-range, i.e., full supply voltage.

The real advantage of higher-radix arithmetic is fewer arithmetic operations. Especially in bulk signal processing units, they may bring in highly sophisticated but less complicated expressions. In design, owing to a larger dynamic range, current is a suitable choice over voltage. The main drawback in current-mode design is level degradation in cascaded stages and inherent static power dissipation. Therefore, it is essential to restore the predefined logic levels at certain number of cascaded stages [8]. Most restoration circuits adopt switching a current at a threshold level through binary logic circuits. It is theoretically shown in [9] that the attainable radix in such a realisation is limited and the transient response is rather spiky. A detailed description of full current-mode logic level restoration architecture can be found in [9, 10].

In this study, a novel algorithm for implementing signed addition in a higher-radix is proposed. The algorithm involves in multi-valued logic (MVL) min, max, cyclic and literal operators. Proposed algebra allows the first two operators to be replaced with an ordinary sum and transmission, or the well-known binary logic AND operation, respectively. The numerical outcome of sign-extended addition of two numbers is represented as a resulting sign, sum and carry digits. Owing to low-level boundaries, literals are easily produced with voltage-mode switching. Alternatively, the cyclic operation is implemented with a highly robust current-mode threshold circuit, hence allowing successive digits to be manipulated in a straightforward manner as cascaded stages.

All circuit simulations presented in this study are obtained from post-layout full extraction with AMS 0.6 μm , CMOS, level 49 HSPICE parameters.

2 Definitions

Consider an r -valued, m -variable function $f(X)$ where $X = \{x_0, x_1, \dots, x_m\}$ and each x_i takes on values from the set

© The Institution of Engineering and Technology 2006

IEE Proceedings online no. 20045096

doi:10.1049/ip-cds:20045096

Paper first received 11th August 2004 and in revised form 4th August 2005

T. Temel was with the Department of Electrical and Electronic Engineering, Bogazici University, Turkey and is now with the Department of Electronic Engineering, Fatih University, 34500, Büyükkçekmece-Istanbul, Turkey

A. Morgul is with the Department of Electrical and Electronic Engineering, Bogazici University, 34342, Istanbul, Turkey

N. Aydin is with the Department of Software Engineering, Bahcesehir University, 34349, Besiktas-Istanbul, Turkey

E-mail: turgaytemel@hotmail.com

$R = \{0, 1, \dots, r-1\}$. The function $f(X)$ is a mapping $f: R^m \rightarrow R$. A logic level $k \in R$ is associated with a continuous interval of quantity x , such that

$$x \rightarrow k : x \in [k - 0.5I_b, k + 0.5I_b] \quad (1)$$

where I_b is the base current. It is associated with logic level 1, and taken to be $10 \mu\text{A}$ in this study. Other levels are defined as integer multiples of I_b . It should be noted that level 0 is associated with range $[0, 5 \mu\text{A})$. In this study, all circuit implementations will be given for radix-8, i.e. $r = 8$.

In arithmetic operations involving some variables with two polarities, such as signed addition, a logic function can be expressed in terms of disjoint conjunction/product terms. If product terms are expressed in terms of suitable unary operations, as will be shown later, it is possible to reach generic mappings of those arithmetic operations. Disjoint terms corresponding to each polarity can be manipulated with ordinary arithmetic operations, which can easily be realised in current-mode designs, such as summing two currents in a node. Hence, the problem turns out to be forming these disjoint terms properly in terms of MVL operators, which are defined as follows.

Definition 1: The complement of $x \in R$, [4], is defined as

$$\bar{x} = r - 1 - x \quad (2)$$

Definition 2: The max operator is represented as

$$\max(x, y) = x + y \quad \text{or} \quad \max(x, y) = x \cup y \quad (3)$$

Definition 3: The min operator is represented as

$$\min(x, y) = x \bullet y \quad \text{or} \quad \min(x, y) = x \cap y \quad (4)$$

Definition 4: A k -valued literal [6] with boundaries a and b is defined as

$$k \bullet \overset{a}{x} \overset{b}{=} \begin{cases} k & \text{if } a \leq x < b \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where

$$\overset{a}{x} \overset{b}{=} \begin{cases} r - 1 & \text{if } a \leq x \leq b \\ 0 & \text{otherwise} \end{cases}$$

and $k, a, b < r$. If $a = b$, the boundary b will be dropped, i.e.

$$\overset{a}{x} \overset{a}{=} \overset{a}{x}$$

Definition 5: The k -cyclic, [4, 7], operations on x are defined as

$$k\text{-clock-wise cyclic, } k\text{-CWC} : x^k \equiv (x + k) \bmod r \quad (6)$$

$$k\text{-counter clock-wise cyclic, } k\text{-CCWC} : x^k \equiv (x - k) \bmod r \quad (7)$$

where '+' is ordinary sum. It is seen that $x^k = x^{r-k}$

Definition 6: A k -valued product term, [6, 7], is of the form

$$P_k : \bigcap_{i \in \{1, 2, \dots, m\}} g(x_i) = k \quad (8)$$

where $g(x_i)$ is a unary operator on x_i s. It may be a literal [6], cyclic [4], or both [7], and ' \cap ' refers to min over $g(x_i)$.

Definition 7: Any MVL function, [6], can be expressed as

$$f(x_1, x_2, \dots, x_m) = \bigcup_{j \in \{1, 2, \dots, r-1\}} P_j \quad (9)$$

$x_1 \backslash x_2$	0	1	2
0	1	0	0
1	1	1	2
2	2	0	1

Fig. 1 MVL function

In this study, when used with only a literal, ' \bullet ' of the min operator will be omitted and '+' will imply its MVL context, i.e. max. As an example, for $r = 3$, $m = 2$, consider the MVL function Shown in Fig. 1.

It can be shown that

$$\begin{aligned} f(x_1, x_2) &= 1 \overset{0}{x_1} \overset{0}{x_2} + 1 \overset{1}{x_1} \overset{0}{x_2} + 2 \overset{2}{x_1} \overset{0}{x_2} + 1 \overset{1}{x_1} \overset{1}{x_2} + 2 \overset{1}{x_1} \overset{2}{x_2} + 1 \overset{2}{x_1} \overset{2}{x_2} \\ &= 1 \overset{0}{x_2} + 1 \overset{1}{x_1} + 1 \overset{1}{x_1} \overset{2}{x_2} + 2 \overset{2}{x_1} \overset{0}{x_2} + 2 \overset{1}{x_1} \overset{2}{x_2} \end{aligned}$$

using the function formalism and minimisation methods in [6] with min symbols omitted.

3 Building blocks

In [8], it was shown that all MVL operations defined previously can be expressed in terms of simpler arithmetic operations, called truncated difference and upper/lower threshold. Since these arithmetic operations are two-valued the relationship between MVL operations and these simpler operations is not invertible, for example they cannot be obtained from MVL operations above. They are defined as follows.

Current-mirrors: They are used for replicating a current with mirroring factor k and/or redirecting currents, Fig. 2. With supply voltage $V_{DD} = 2.7 \text{ V}$ and base current $I_b = 10 \mu\text{A}$, a linear operation within the range 0 through $75 \mu\text{A}$, i.e. $r = 8$, can be achieved with transistor aspect ratios 1.6/1 and 3.5/1 for N-type and P-type current mirrors, respectively. These ratios will be referred to as minimum feature sizes for the parameters chosen in this study. However, the output transistor(s) of a mirror is chosen slightly wider as a remedy

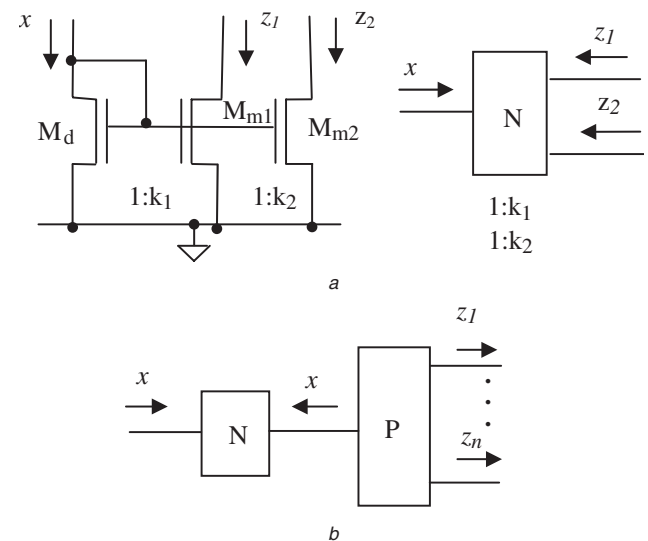


Fig. 2 Current-mirror and basic operations
a N-type current mirror
b Redirecting currents

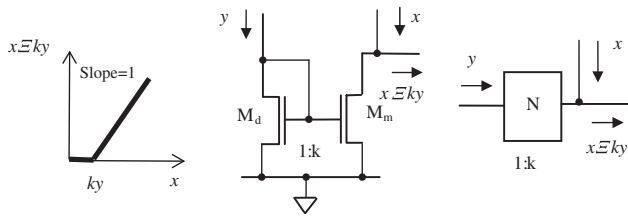


Fig. 3 Truncated difference operation DC characteristics, circuit and symbol

against the modulation effects. A detailed analysis of linear DC operation of current mirrors with various types of N-type and P-type connections can be found in [9].

Input/output circuit: A minimum-feature size diode-connected NMOS transistor is used as the input/output circuit of current-mode blocks, unless otherwise stated.

Truncated difference: This operation is defined by following equation and it can be realised as shown in Fig. 3.

$$x \oplus ky = \begin{cases} x - ky & \text{if } x \geq ky \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

Upper/lower threshold: These operations are defined as

$$\text{upper-threshold, } th_u: a|_b^c = \begin{cases} c & \text{if } a \geq b \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

$$\text{lower-threshold, } th_l: {}_b^c|a = \begin{cases} c & \text{if } a \leq b \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

Figure 4a shows upper threshold operation characteristics and its block representation. The quantity Δa is the total transition width owing to non-ideal switching behaviour in a possible realisation. A highly robust upper threshold design can be obtained with the circuit shown in Fig. 4b where added subscript 'p' denotes the positive feedback.

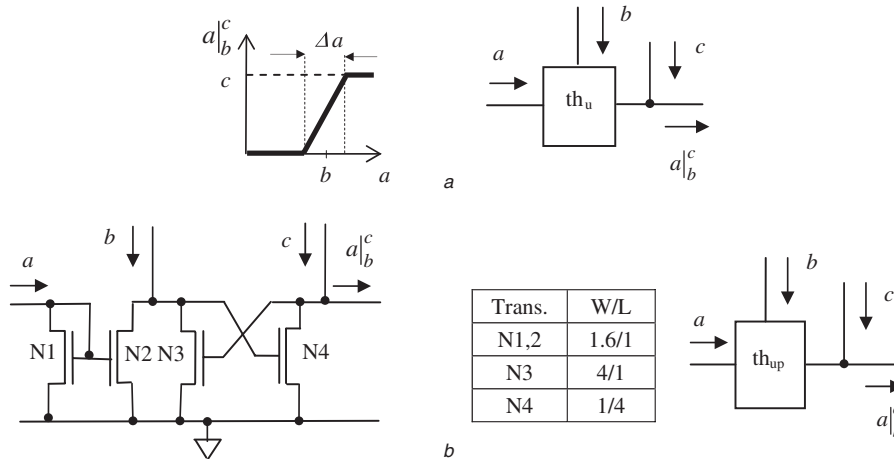


Fig. 4 Upper threshold operation

a Upper threshold operation DC characteristics and symbol

b Positive feedback upper threshold, th_{up} circuit, transistor aspect ratios for $r = 8$, and symbol

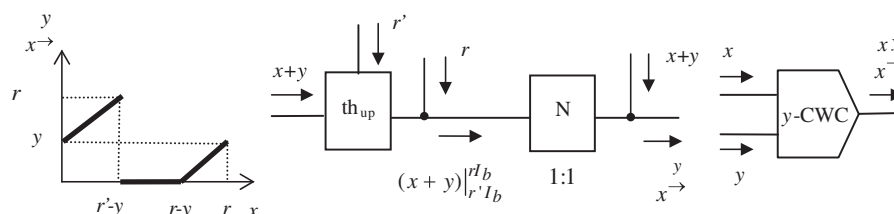


Fig. 5 Cyclic operation DC characteristics, implementation with th_{up} and operation symbol

The circuit is obtained by cascading a truncated difference circuit formed with N1 and N2 and a positive feedback current comparator circuit [11] formed with N3 and N4. Aspect ratios of N3 and N4 are such that $\Delta a = 2 \mu A$ for output current from 0 through $75 \mu A$. The inputs a and b are compared by the truncated difference circuit. N2 is 'on' if $a > b$ forcing the gate of N4 to a low voltage and turning it 'off'. So the output is equal to c . Otherwise, the node 'A' has a high potential which drives transistor N4 to turn 'on', hence setting the output to '0'. The positive feedback between transistors N3, N4 ensures the bi-stable operation, i.e. one of the transistors is 'on' while the other is 'off'. Lower-threshold operation can be achieved by interchanging inputs a and b in upper threshold configuration.

4 MVL cyclic operator gate implementation

Considering the logic level definition in (1), it can be shown that a y -clockwise-cyclic operation on x given a radix- r can be expressed as

$$y\text{-CWC}: x^y = (x + y) \Xi (x + y) |_{r'}^r = (x + y) \Xi x |_{r-y}^r \quad (13)$$

where $r' = r - 0.5$ and $x + y$ is the ordinary current sum at the input node. Figure 5 depicts th_{up} implementation of cyclic operation $z = x^y$ obtained by cascading the upper threshold and truncated difference circuits in accordance with (13). Since $x + y$ may be greater than r , aspect ratios of input mirror transistors N1, 2 of the th_{up} , need to be chosen properly for a suitable current input value. The additional mirror/copy of $x + y$ is reproduced with a P-type current mirror having unit-mirror-factor consisting of PMOS transistors of aspect ratios 7/1. Figure 6 illustrates the respective HSPICE simulation results.

Table 1 gives a comparison between th_{up} implementation of cyclic gate and previously proposed two design schemes

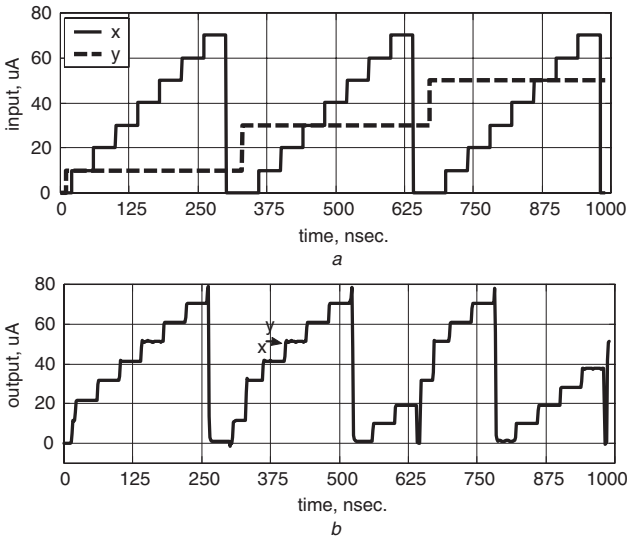


Fig. 6 Current waveforms
a Input current waveforms to th_{up} -based radix-8 cyclic circuit
b Output current waveforms

Table 1: Comparison of radix-8 cyclic circuit implementations

Topology	No. of transistors	Av. del. (ns)	Av. pow. diss. (mW)	σ_{Az} (μA)
With th_{up} [10]	9	1.8	0.78	3.1
With th_{utd} [9]	19	1.65	0.66	6.9
[7]	23	4.1	0.63	4.85

in [7] and [9], for $r = 8$. In [7], the cyclic gate is implemented with current-mode based on voltage-mode switching obtained through the current comparator scheme in [12]. In [9], a different topology is introduced with the use of a full-current-mode threshold circuit based on truncated difference, referred to as th_{utd} here. Designs are loaded with the output circuit described previously. The quantities are averaged over illustrated logic levels of x and y . The quantity σ_{Az} is the average standard deviation of output current and it is computed when z has reached maximum logic level, i.e. $75 \mu A$, as the worst-case condition. Mismatch properties are studied by using Monte Carlo analysis of 100 iterations with normal Gaussian variations $\sigma_{\Delta W/W, \Delta L/L} = 5\%$ and $\sigma_{\Delta V_{TO}} = 50$ mV.

It should be noted that the current-mode cyclic circuits operate faster than the voltage-mode architecture while the power dissipations are almost the same. However, th_{up} -based design requires the fewest transistors. An important parameter from the higher-level design perspective is the mismatch sensitivity. Table 1 reveals that the th_{up} -based design is also least sensitive to parameter deviations. In order to investigate the latest figure, we can define a quantity, GC , which is the maximum number of cascaded identical gates without logic level degradation. By assuming Δz output deviation of each stage independent, normally-distributed and input values such that each stage output is at logic level 8, then a reliable design condition can be set roughly as

$$GC \leq \left(\frac{I_b}{3\sigma} \right) \quad (14)$$

It is seen that the proposed architecture allows up to 3 cyclic gates to be cascaded while other designs demonstrate worse

GC capability. Therefore, the new topology allows one to build up higher-level arithmetic units.

In order to rebuild a logic level from deviated values owing to such effects as mismatching and channel modulation, the restoration circuit [9, 10] proposed recently can be utilised. The circuit employs cascaded th_{up} -based cyclic stages and requires 38 transistors for $r = 8$. With the same design parameters and same radix, it is found that the average delay is 2.6 ns while power dissipation is 0.25 mW for a linear input waveform up to $80 \mu A$ with unit slope.

5 Higher-radix full-adder with MVL cyclic operator

Considering the definition of cyclic operation in (6), it is possible to construct full-addition $x+y+C_{in}$ given $x, y, C_{in} \in R$. The result will be of a two-digit representation (CS) where S and C are the sum and carry digits, respectively. The individual digits are given by

$$S = x^{y+C_{in}} = \begin{cases} x + y + C_{in} & \text{if } x + y + C_{in} < r' \\ x + y + C_{in} - r & \text{otherwise} \end{cases} \quad (15)$$

$$C = \begin{cases} 0 & \text{if } x + y + C_{in} < r' \\ 1 & \text{otherwise} \end{cases}$$

In order to obtain an MVL full-adder, th_{up} -based cyclic circuit can be modified as shown in Fig. 7. The circuit includes an additional transistor switched by the current-comparator output voltage at the drain of N2 for generating carry. Aspect ratios of N1,2 are chosen 3.5/1 if the full-adder is the first unit otherwise 1.6/1. HSPICE simulation results are shown in Fig. 8.

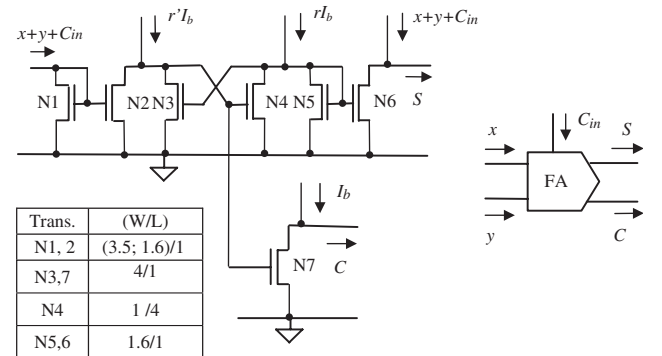


Fig. 7 Higher-radix full-adder circuit implementation, transistor aspect ratios for $r = 8$ and operation symbol

Having shown that th_{up} -based realisation is the most appropriate architecture for designing higher-radix arithmetic units, derived radix-8 MVL adder can be compared to some well-known full-adders. Since a 3-bit chunk in two-valued logic is associated to a single digit in radix-8, binary-logic adders are designed as 3-bit units. Despite some other full-adder designs in BiCMOS [13, 14], owing to availability, designs are given for CMOS. The proposed radix-8 full-adder, 3-bit ripple-carry adder (RCA), carry-look-ahead adder (CLA) [1], carry-skip adder (CSkA) [15], transmission-function based adder (TFA) [16] based on pass-transistor logic, residue arithmetic adder, as a radix-7 using radix-8 adders composed of two 3-bit TFAs, a multiplexer (RAA) [17], and radix-8 current-mode MVL adder (BCMVL) [18] based on current-mode binary-logic encoding of the current-comparator output voltages with respect to sum of input currents are designed with the same parameter set and loaded with a capacitive load of 70 fF at

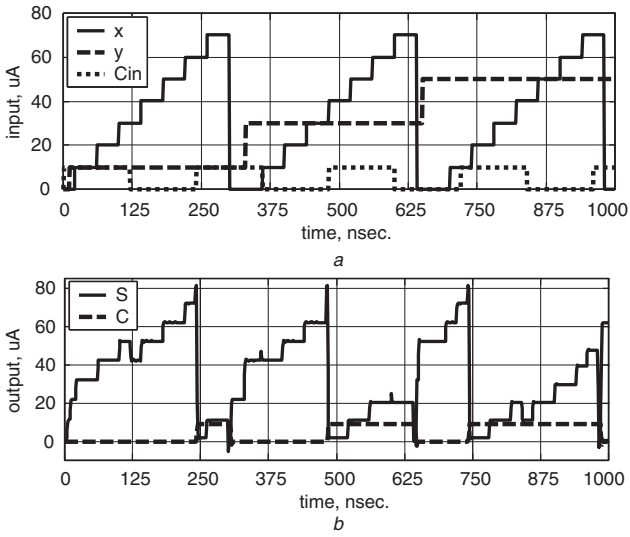


Fig. 8 Current waveforms
 a Input current waveforms to radix-8 full-adder
 b Output current waveforms, $T=40$ ns

output(s). For visualisation of differences in area, hence interconnections, layouts of radix-8 adder and 3-bit binary RCA are given in Fig. 9. In order to estimate the worst-case power dissipation, 500 random input vectors are applied to binary voltage-mode adders. Table 2 provides various

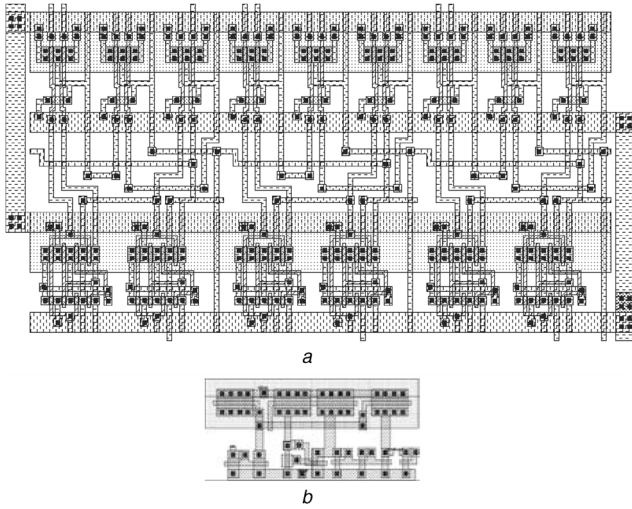


Fig. 9 Layout
 a 3-bit binary RCA adder
 b Radix-8 adder

Table 2: Important characteristics of radix-8 MVL and binary adders

Full adder	No. of transistors	Max. del. (ns)	Av. power diss. (mW)			Area, (μm) ²
			$T=40$ ns	$T=10$ ns	$T=7.5$ ns	
MVL, $r=8$	12	2.4	0.71	0.74	0.78	53×22
3-bit RCA, [1]	84	6.1	0.10	0.44	0.69	158×82
3-bit CLA, [1]	108	6.6	0.10	0.51	0.78	172×91
3-bit CSkA, [15]	94	5.3	0.08	0.34	0.41	107×101
3-bit TFA, [16]	66	5.0	0.04	0.15	0.22	93×64
RAA, $r=7$, [17]	165	5.3	0.12	0.49	0.65	117×185
BCMVL, $r=8$, [18]	48	6.3	0.53	0.77	0.98	115×84

T : time duration of single step length for MVL adders and excitation period for binary adders, respectively

aspects of post-layout HSPICE simulation results of the adders stated above.

Table 2 reveals that MVL design offers a much less area. It operates faster in return for higher power dissipation at large T values, which is mainly due to the static component. However, as excitation speeds up, dynamic power dissipation starts to dominate and it becomes highly pronounced for conventional adders while it almost remains constant for MVL design. Except for TFA and CSkA, power dissipation of proposed circuit is comparable to others' while it outperforms all other designs in terms of area and speed.

6 MVL higher-radix signed full-addition algorithm

By extending the use of MVL higher-radix full-adder previously proposed, it is possible to establish an algorithm for signed-addition of higher-radix numbers. The algorithm has some novel features, such as combining MVL operators with ordinary arithmetic operations. Consider a signed number, $x' \in R$, attached to a two-valued sign digit, s_x , and magnitude x , as $x' = (s_x, x)$. Then, x' can be converted into its r 's complement/unsigned form, x_u , as

$$x_u = (r - x) \bullet \overline{s_x}^{s_y} + x \bullet s_x^{s_y} \quad (16)$$

where '+' and ' \bullet ' are the MVL max and min operators, respectively. With the description in (16), the algorithm can be stated as follows:

- i) Consider n -digit, signed numbers, $x' = (s_x, x_{n-1} x_{n-2} \dots x_0)$, $y' = (s_y, y_{n-1} y_{n-2} \dots y_0)$ with $x_i, y_i \in R$.
- ii) Convert one of the numbers into its unsigned form, e.g. x' , if $s_x = s_y$; otherwise leave each digit as it is. Then, subtract each digit x_i , $i = 1, 2, \dots, n-1$, from $r-1$, and x_0 from r .

The resulting sum is of the form

$$z' = s_z C_z z = s_z C_z z_{n-1} z_{n-2}, \dots, z_0 = s_z C_z x_u^{y_u} \quad (17)$$

In (17), the cyclic operation is carried out digit-by-digit and s_z is the sign of summation given by

$$s_z = (1 - C) \bullet \overline{s_x}^{s_y} + 1 \bullet s_x^{s_y} \quad (18)$$

The term C is an auxiliary carry, and it is given by

$$C = \begin{cases} 0 & \text{if } x_{u,n-1} + y_{u,n-1} + C_{n-1} < r \\ 1 & \text{otherwise} \end{cases} \quad (19)$$

The final carry digit, C_z , is then given by

$$C_z = C \bullet s_x^{s_y} \quad (20)$$

It should be noted that, owing to disjoint terms, the max operator in (16) can be replaced with ordinary sum. Moreover, because a sign digit is of single-polarity and it

is two-valued, it is convenient to perform the sign digit computations in voltage-mode binary logic. However, carry transferral from successive digits must be in the current-mode. Following are some examples to justify the algorithm for $r=8$:

Example 1: $x' = 0532$, $y' = 0223$; $s_x = s_y = 0 \Rightarrow x_u = 532$, $y_u = 223$, $s_z = 0$, $z = (532) \xrightarrow{(223)} = (0)755 \Rightarrow C = 0$ and $C_z = 0$, $z' = 00755$

Example 2: $x' = 0532$, $y' = 0332$; $s_x = s_y = 0 \Rightarrow x_u = 532$, $y_u = 332$, $s_z = 0$, $z = (532) \xrightarrow{(332)} = (1)064 \Rightarrow C = 1$ and $C_z = 1$, $z' = 01064$

Example 3: $x' = 1532$, $y' = 0332$; $s_x = 1$, $s_y = 0 \Rightarrow x_u = 778 - 532 = 246$, $y = 332$, $z = (246) \xrightarrow{(332)} = (0)600 \Rightarrow C = 0$, $s_z = 1$ and $C_z = 0$, $z' = 10600$

Example 4: $x' = 0532$, $y' = 1332$; $s_x = 0$, $s_y = 1 \Rightarrow x_u = 532$, $y_u = 446$, $z = (532) \xrightarrow{(446)} = (1)200 \Rightarrow C = 1$, $s_z = 0$ and $C_z = 0$, $z' = 00200$

Example 5: $x' = 1532$, $y' = 1332$; $s_x = s_y = 1 \Rightarrow s_z = 1$, $x_u = 532$, $y_u = 332$, $z = (532) \xrightarrow{(332)} = (1)064 \Rightarrow C = 1$ and $C_z = 1$, $z' = 11064$.

In order to implement the above algorithm, a distinction between the voltage-mode and current-mode operations has to be made. Consider the following expressions where the subscript v/c of the quantity q on the left-hand side refers to voltage/current mode outcome of the right-hand side computations, respectively,

$$q_v = t_1 \bullet^{p_{x1} p_{x2}} x + t_2 \bullet^{p_{y1} p_{y2}} y \quad (21)$$

$$q_c = t_3 \bullet^{p_{x3} p_{x4}} x + t_4 \bullet^{p_{y3} p_{y4}} y \quad (22)$$

The above expressions should be interpreted as follows:

(i) The operator '+' in (21) is a conventional OR operator, while the operator '•' stands for AND or transmission operator through a transmission gate depending on its associated coefficient: If $t_{1,2}$ is voltage and two-valued, then '•' operator is AND operator. Otherwise '•' operator is a transmission operation where associated coefficient is the result of current comparator yielding a two-valued voltage-mode quantity.

(ii) The operator '+' in (22) is an ordinary summation while the operator '•' stands for the transmission where current t_3 and t_4 are switched through related literal operations.

7 Implementation of the algorithm

In order to realise the algorithm above, consider the circuit illustrated in Fig. 10. In the circuit, $s_{1,2}$ are derived switching voltage signals from $s_{x,y}$. Having set up the logic mode distinction, it is possible to generate the terms, s_z and C_z ,

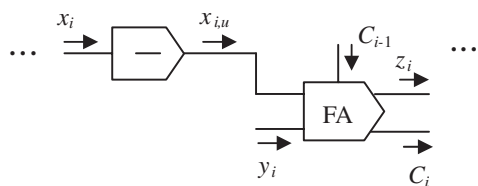


Fig. 13 Complete signed higher-radix adder

with the use of circuit shown in Fig. 11. In the circuit, all voltage-mode binary logic circuits are designed for optimum speed given minimum size, while the reference voltage, $V_{ref} \approx 1.1$ V, is obtained with a minimum feature-size, simple PMOS current-mirror [11].

The current-mode circuit that performs r 's complement of $x' = s_x x$ is illustrated in Fig. 12. The constant r^* is r for $i=0$ and $r-1$ for $i=1, \dots, n-1$.

Then, the complete MVL higher-radix signed addition of two numbers can be realised with the structure shown in Fig. 13, where $C_0 = 0$.

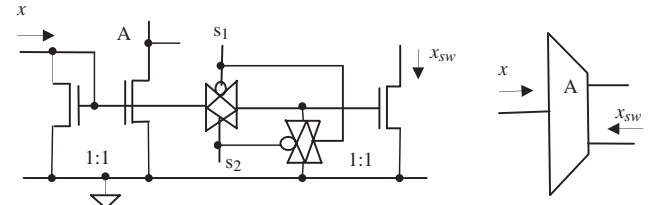


Fig. 10 Conversion of a current-mode operand into voltage and current-mode operands and operation symbol

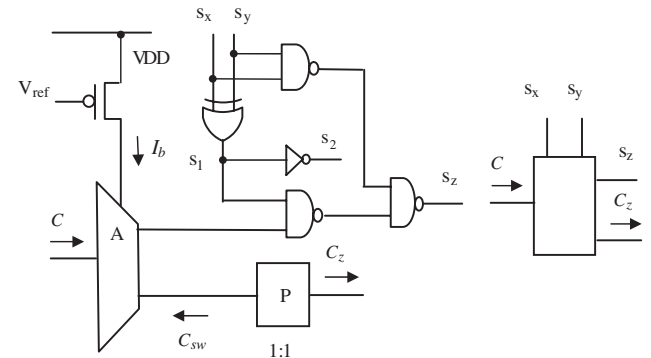


Fig. 11 Generating final carry, C_z , and output sign, s_z , digits, and operation symbol

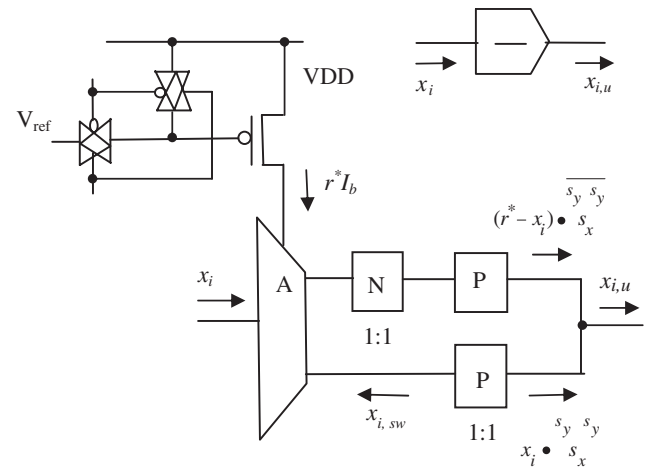


Fig. 12 The r 's complement circuit, and symbol

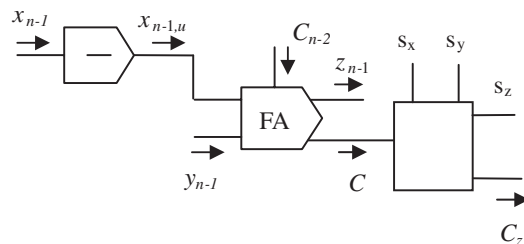


Table 3: Some characteristics of two-digit, signed radix-8 adder

Full adder	No. of transistors	Max. delay (ns)	Av. power diss. (mW)			Area, (μm) ²	$\sigma_{Az, \max}$ (μA)
			$T = 40 \text{ ns}$	$T = 10 \text{ ns}$	$T = 7.5 \text{ ns}$		
Two-digit radix-8	86	6.9	3.0	3.25	3.4	151 × 116	6.4

T = time duration of one step

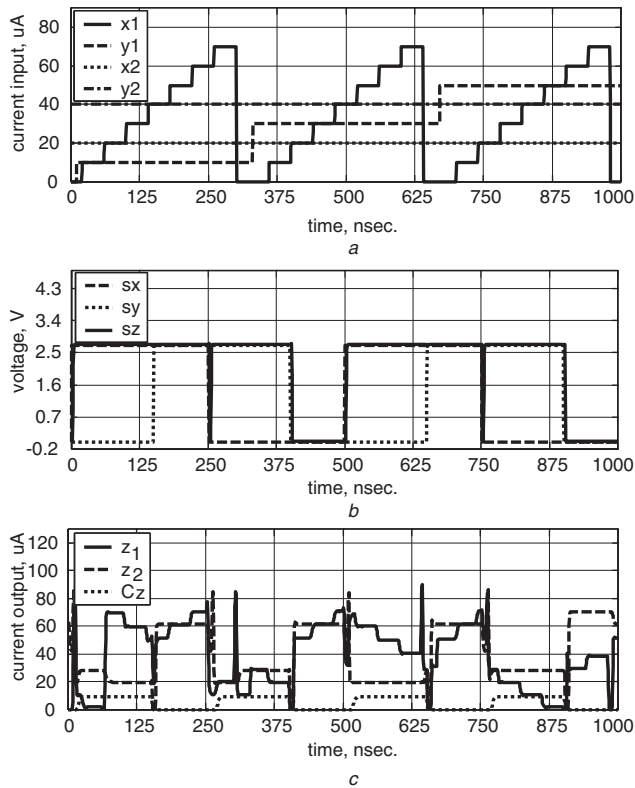


Fig. 14 Input currents, input and output sign voltages, and output currents

a Input currents to two-digit, radix-8 signed adder circuit
 b Input and output sign voltages
 c Output currents

8 Simulation results

The circuit illustrated in Fig. 13 has been implemented and simulated with HSPICE for two-digit numbers x and y whose magnitude and sign digit variations are depicted in Figs. 14a and b, respectively. The circuit outputs are loaded a capacitance of 70 fF. Corresponding output sign digit is also illustrated in Fig. 14b while the current-mode output digits are shown in Fig. 14c. Table 3 summarises some important specs of the designed circuit.

The term $\sigma_{Az, \max}$ stands for the maximum value of the standard deviation of current-mode digits across whole structure owing to level degradation. Since design is two-digit, it is seen each stage of signed addition has a level degradation of $3.2 \mu\text{A}$, which allows maximum two single-digit stages to be cascaded, i.e. three-digit signed radix-8 addition.

It should be noted that the power dissipation is nearly constant as the stimulation speeds up, which is an expected consequence of proposed full-adder scheme.

The higher radix signed adder algorithm described above has the merit of direct complement of the operands with respect to sign values. Taking the sign digits in voltage-mode allows one to combine MVL design literal concept

within a conventional voltage-mode binary logic environment, subject to circuit performance constraints described in [19]. However, voltage-mode switching in current-mode design imposes larger transient peaks, as discussed in [20].

9 Conclusions

For well-known information processing virtues of higher-radix algebra, multi-valued logic operators are exploited. Towards a feasible, and modular higher-radix adder unit, a hierarchical approach, starting from a threshold definition has been described. Owing to versatility and robustness of the proposed MVL circuits, it is expected to combine them in more complicated circuit blocks. Reduced interconnection and switching voltage range bring in improved dynamic behaviour. A novel signed addition algorithm is presented. Implementation of the algorithm shows that proposed topologies can be employed in a mixed-signal environment as a challenging alternative to conventional arithmetic units for high-speed, low-power performance design. Main advantage stems from better transient characteristics owing to less interconnection and less switching activity, which are the dominant factors in low-power design techniques as well. Inherent static power dissipation and the necessity of restoring the logic levels are prohibitive factors, which can be overcome in a certain extent by using a lower radix, and reduced logic level margins and sophisticated restorers.

10 References

- Bellaour, A., and Elmasry, M.I.: 'Low-power digital VLSI design' (Kluwer Academic Publishers, The Netherlands, 1995)
- Pomper, G., and Armstrong, J.R.: 'Representation of multivalued functions using the direct cover method', *IEEE Trans. Comput.*, 1981, **C-30**, pp. 674–679
- Mekhallalati, M.C., and Ibrahim, M.K.: 'A new high-radix maximally-redundant signed digit adder'. Proc. Int. Symp. on Circuits and Systems, ISCAS'99, Orlando, Florida, USA, 1999, pp. 459–462
- Vranesic, Z.G., Lee, E.S., and Smith, K.C.: 'A many-valued algebra for switching systems', *IEEE Trans. Comput.*, 1970, **C-19**, pp. 964–971
- Stankovic, R.S., and Moraga, C.: 'Methods for detection of some properties of multiple-valued functions', *IEE Proc. E, Comput. Digit. Tech.*, 1992, **139**, (3), pp. 421–429
- Allen, C.M., and Givone, D.D.: 'A minimization technique for multi-valued logic systems', *IEEE Trans. Comput.*, 1968, **C-17**, pp. 182–184
- Jain, A.K., Bolton, R.J., and Abd-El-Barr, M.: 'CMOS multiple-valued logic design, Part I and II', *IEEE Trans. Circuits Syst. Fundam. Theory Appl.*, 1993, **40**, (8), pp. 503–522
- Temel, T., and Morgül, A.: 'Implementation of multi-valued logic gates using full current-mode CMOS circuits', *Analog Integr. Circuits Signal Process.*, 2004, **39**, (2), pp. 191–204
- Temel, T.: 'Current-mode CMOS design of multi-valued logic circuits'. PhD thesis, Bogazici University
- Morgül, A., and Temel, T.: 'Current-mode level restoration: circuit for multi-valued logic', *Electron. Lett.*, 2005, **41**, (5), pp. 230–240
- Baker, J., Li, H.W., and Boyce, D.: 'CMOS: circuit design, layout and simulation' (Wiley-IEEE, New York, 1997)
- Freitas, D.A., and Current, K.W.: 'CMOS current comparator circuit', *Electron. Lett.*, 1983, **19**, (17), pp. 695–697
- Bellaour, A., and Touzenc, H.: '3.3 V BiCMOS current-mode logic circuits for high-speed adders', *IEEE J. Solid-State Circuits*, 1996, **31**, (8), pp. 1165–1169
- Ueda, K., Suzuki, H., Suda, K., Shinohara, H., and Mashiko, K.: 'A 64-bit carry look-ahead adder using pass transistor BiCMOS gates', *IEEE J. Solid-State Circuits*, 1996, **31**, (6), pp. 810–818

- 15 Strollo, A.G.M., and Napoli, E.: 'A fast and area-efficient complementary pass transistor logic carry skip adder'. Proc. 21st Int. Conf. on Microelectronics, Sept. 1997, pp. 701–704
- 16 Zhuang, N., and Wu, H.: 'A new design of the CMOS full adder', *IEEE J. Solid-State Circuits*, 1992, **27**, (5), pp. 840–844
- 17 Dugdale, M.: 'VLSI implementation of residue adders based on binary adders', *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, 1992, **39**, (5PP.), pp. 325–329
- 18 Radanovic, B., and Syrzycki, M.: 'Current-mode CMOS adders using multiple-valued logic'. Proc. IEEE Canadian Conf. on Electrical and Computer Engineering, May 1996, pp. 190–193
- 19 Vittoz, E.A.: 'The design of high performance analog circuits on digital CMOS chips', *IEEE J. Solid-State Circuits*, 1985, **SC-20**, (3), pp. 657–665
- 20 Yamakawa, T.: 'CMOS multi-valued circuits in hybrid mode'. Proc. 15th IEEE Int. Symp. on Multiple-Valued Logic, Ottawa, Canada, 1985, pp. 144–151