Journal of Applied Sciences

ISSN 1812-5654

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 10 (23): 3051-3059, 2010
ISSN 1812-5654

Transmission Gate based High Performance Low Power Multiplier

[1]C.N. Marimuthu and [2]P. Thangaraj
[1]Maharaja Engineering College, Avinashi, Coimbatore, India
[2]Kongu Engineering College, Perundurai, Erode, India

Abstract: The performance analysis of various multiplier architectures are compared in terms of power, delay and area occupation in the view of low-power low-voltage signal processing for low-frequency applications. A novel practical approach has been set up to investigate and graphically represent the mechanisms of glitch generation and propagation. It is found that spurious activity is a major source of power dissipation in multipliers. Measurements point out that the shorter full-adder chains in the Wallace multiplier dissipates less energy than as compare to other longer full-adder chains traditional array multipliers. The benefits of transistor sizing are also evaluated. In this study transmission gates combined with static CMOS circuits to reduce glitches in Wallace multiplier architecture is proposed to improve the energy-efficiency as compare to traditional array architecture. The reduced number of Vdd-to-ground paths, reduced glitches due to level restoring gates, the equalized internal signal delay and shortening of full adder chains are the unique techniques used to reduce power dissipation in proposed transmission gate based Wallace multipliers as compare to prior designs.

Key words: Arithmetic, glitch, low frequency, low power, multiplier, switching activity, transmission gate

INTRODUCTION

With the development of deep sub-Micron CMOS technologies and the increase in complexity of VLSI chips, the market for portable applications, digital signal processors and ASIC implementations has focused significant effort on the design of Low-power systems. Low-power circuits have many advantages over those that do not employ power-saving strategies. First, the digital system for portable applications such as personal communications, hearing aids and personal digital assistants, allow the use of lighter batteries and/or prolong the battery life (Chong *et al*., 2005). Second, low-power techniques decrease the costs of cooling and packaging. The circuit reliability deteriorates with increased heat dissipation, so low-power techniques can improve the robustness of CMOS circuits. As an essential logic component in microprocessors and digital signal processing systems (Mosch *et al*., 2000) a multiplier significantly contributes to the overall system power consumption. In this study, we present a multiplier that uses several novel techniques to minimize its power consumption. Digital multipliers are major source power dissipation in digital signal processors. Array architecture is a popular technique to implement these multipliers due to its regular compact structure. High power dissipation in these structures is mainly due to the switching of a large number of gates during multiplication.

In addition, much power is also dissipated due to a large number of spurious transitions on internal nodes. However, recent research on signal transition activity indicated that array multipliers have an architectural disadvantage. This is mainly due to no uniform path delays in the structure, which results in multiple signal transitions on internal nodes before they settle to a final value. These multiple transitions are spurious or redundant and, consequently, dissipate unnecessary power. In fact, in a recent study of an array multiplier, almost 50% of the dynamic power was consumed due to these spurious transitions (Chen and Chu, 2007).

In the signal processing offered in modern audio applications, multipliers are certainly among the most power-hungry elaboration units. At the same time, they are very frequently used components in Application-Specific Integrated Circuits (ASICs) and fundamental blocks in Digital Signal Processors (DSPs). Being rather complex combinational modules with numerous unbalanced reconvergent paths, multipliers suffer particularly from spurious switching activity generation and propagation, which can even dominate the total dynamic consumption (Alioto and Palumbo, 2002). While trying to optimize the efficiency of multipliers, many works in the past investigated (Chang *et al*., 2005; Shams *et al*., 2002) only the basic constitutive cell, namely the full-adder. This way of proceeding overlooks the previously-mentioned relevant aspect of glitch

Corresponding Author: C.N. Marimuthu, Maharaja Engineering College, Avinashi, Coimbatore, India propagation and does not take wire parasitic into account either. The easiest solution to reduce spurious activity propagation is certainly pipelining. Yet, the large power and area overheads due to the introduction of Flip-Flops (FFs) limit its use to high speed implementations (Sulistyo and Ha, 2003). Apart from that, three fundamental approaches have been proposed in the literature so far to abate glitch generation and propagation in parallel multipliers, namely:

- Shortening full-adder chains
- Equalizing internal delays
- Aligning sum and carry signals

The first technique consists in rearranging the full-adder cells in order to carry out the same operation within shorter paths. The advantage is that fewer glitches are generated and propagated. When this can be done with no extra logic, as in a Wallace tree introduced by Wallace (1964), the energy efficiency is destined to increase with no other limitation than a growing routing complexity. Yet, a large proportion of spurious activity still remains.

In the second technique, the delays of the internal signals are equalized by redesigning the full adders (Mahant-Shetti *et al*., 1999). The efficiency is generally dependent on parasitic and process variations.

The third technique consists in the alignment of the internal signals by means of self-timed circuits (Sobelman and Raatz, 1995). For example (Chong *et al*., 2005; Carbognani *et al*., 2006) independent delay line triggers special cells that implement the functionality of both a full-adder and a latch. These circuits present superior glitch suppression (Sulistyo and Ha, 2003). However, large energy overhead and strong process dependence represent a heavy burden.

Two more general techniques for glitch suppression, which do not specifically address multiplier architectures, have been proposed. The first one acts on transistor sizes to adjust the cell delays, in order to balance reconverging paths, hence reducing glitch generation. The second publication implements a special resistive cell to increase internal ramp times. Compared to these two low-power strategies, the hereby introduced technique presents the following advantages:

- It limits the area increase, which is relevant in Uppalapati *et al*. (2005)
- It can do without large consuming transistors, needed by Agrawal (1997)
- It is more robust to process and voltage variation

This study confirms the relevant power efficiency of the Wallace tree over other traditional structures, by presenting a comprehensive study on the spurious activity propagation. The effect of transistor sizing is also evaluated: in low-frequency low-voltage applications, minimum-size devices decrease the switching capacitance without leading to large cross over currents (Buergin *et al*., 2006). Based on these results, new multiplier architecture is introduced, called TG-Multiplier that reduces spurious activity further compared with both traditional and recently published architectures. At the same time, TG-Multiplier has positive effects on leakage reduction and it is robust to process variation and voltage scaling, without imposing any overhead in terms of energy. The introduced technique combines static CMOS with transmission gates that abate glitches via Resistance-Capacitance (RC) equivalent low-pass filtering. Additionally, it guarantees limited overhead of propagation delay and area, hence finding potential application in low-frequency portable devices, such as hearing aids.

SIGNED MULTIPLICATION

Given two unsigned binary bit wide numbers and, the multiplication operation is defined as follows:

$$Z = X.Y = \Sigma\ (\Sigma(X_iY_j)2^{(i+j)}) \tag{1}$$

where, Z represents the product, X_i the i th bit of the multiplicand and Yj the jth bit of the multiplier. The modified Baugh-Wooley algorithm allows the conversion from unsigned to signed multiplication.

When the Booth radix-4 recoding is applied, transforms into the following:

$$Z = X.Y = \Sigma\ (\Sigma(X_iYb_j)2^{(i+2j)}) \tag{2}$$

where, Yb_j^a {-2,-1, 0, 1, 2} represents the jth operand of the multiplier after Booth recoding. As can be noticed from Eq. 1 and 2, Booth recoding allows the number of partial products to be halved, hence halving the number of additions. Yet, the precalculation of and the multiplication of the multiplicand by -2,-1 and 2 require extra-logic, which is paid in terms of power dissipation and area occupation.

MULTIPLIER ARCHITECTURES

Traditional multiplier architectures: Equation 1 and 2 suggest a matrix of full- and half-adders; the way these cells are connected together defines the specific multiplier architecture. The most widespread architectures are the following:

- Array multiplier
- Carry-save multiplier (CSM)
- Wallace tree

Array multiplier: Advances in VLSI technology have made it possible to build combinational multipliers, extra logic to allow the product to be computed in one step, arrays of simple combinational elements like add, shift operation.Array multiplier is an efficient layout of combinational multiplier. It may be pipelined to decrease clock period at the expense of latency. Consider the multiplication of two unsigned binary integers

$$X = x_{n-1} \ldots x_1 x_0$$

$$Y = y_{n-1} \ldots y_1 y_0$$

The product P can be expressed as:

$$P = X.Y = \left(\sum_{i=0}^{n-1} 2^i x_i\right)\left(\sum_{i=0}^{n-1} 2^j y_j\right)$$

or

$$P = \sum_{i=0}^{n-1} 2^i \left(\sum_{i=0}^{n-1} x_i y_j 2^j\right)$$

Each of the n^2-1 bit product terms x_iy_j can be computed by an AND gate. An n × n array of AND gates can compute all the x_iy_j terms simultaneously. The terms are summed by an array of n (n-1) full adders. The resulting circuit is similar to a two-dimensional ripple-carry adder. The shifts implied by the 2^i and 2^j factors are implemented by the spatial displacements of the adders along the x and y dimensions.

The array multiplier offers lower speed but consume smaller areas than wallace tree multiplier. As shown array multiplier in Fig. 1a and b, multiplicand bits are simultaneously input to all the partial product generators at every stage. All full adders start comuting at the same time without waiting for the propagation of sum and carry signals from the previous stage. The outputs of the multiplier have many glitiches. As for the power dissipation, the signal transition activity directly influences the dynamic power dissipation. This results in sprious transitions at the output and wastes power. Furthermore, as shown in Fig. 2, since these spurious transition are propagated to the next stage continuously, their numbers grow stage by stage like snow ball. This causes a significant increase in power dissipation.

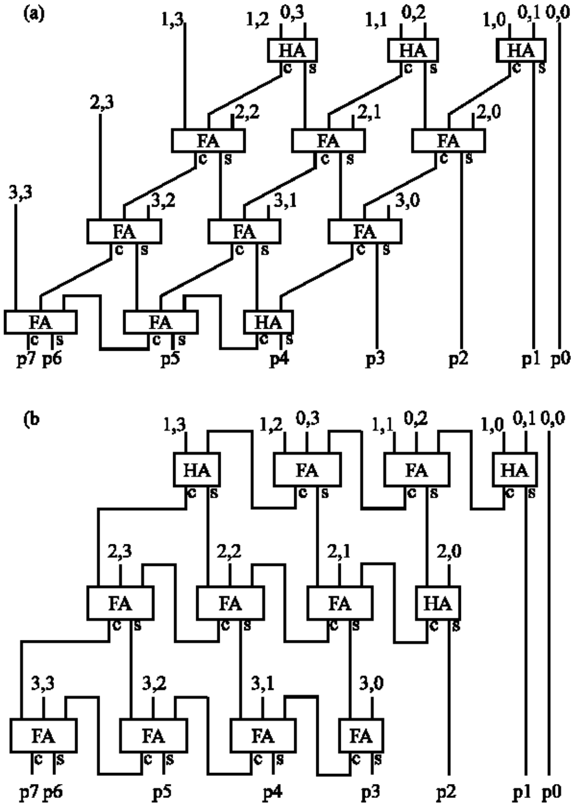


Fig. 1: (a) array multiplier-row ripple and (b) array multiplier-column ripple

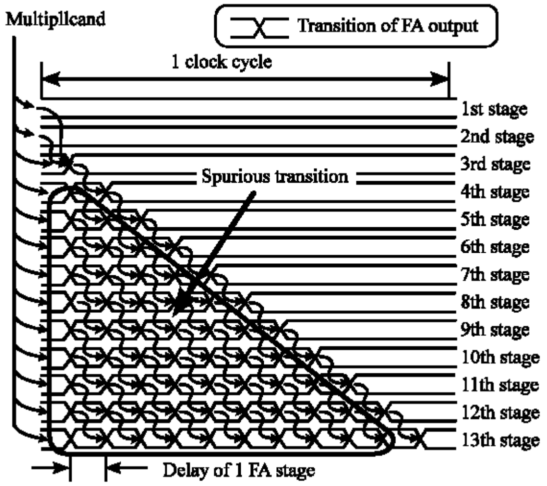


Fig. 2: Spurious transition in multiplier

Carry save multiplier: A carry save adder is just a set of one-bit full adder without any carry-chaining. Therefore, an n-bit CSA receives three n-bit operands namely A(n-1)... A(0), B(n-1)... B(0) and CIN(n-1)... CIN(0) and generates two n-bit result values, SUM(n-1)... SUM(0) and COUT(n-1)…….COUT(0). The most important application of a carry save adder is to calculate the partial products in integer multiplication. The CSM is a very regular structure, in which the carry bits descend a row while propagating from the least significant to the most significant bit. Booth recoding has been introduced to speed up the operation of multiplication. The number of partial products is halved at the expense of some extra logic inside the Booth encoder. Tree multipliers are different full-adder rearrangements, compared to array multipliers, such as the CSM. In particular, in the Wallace-tree multiplier the AND terms are added all at once before entering the full-adder matrix shown in Fig. 3. This results in an irregular architecture, which allows the longest path to be shortened up to the final addition. The latter can be carried out according to well-known adder topologies.

The final unit in a parallel multiplier is fast adders, which performs fast addition for the sum and carry bit vectors from outputs of the PPRT. There are many different fast adders that suit parallel multipliers, such as carry look ahead, carry skip adder, carry save adder. Glitches are the main responsible for the different dissipation of traditional multiplier architectures. Assuming that all input signals arrive at the same time, the spurious activity originates from the following:

- Different delays of sum and carry bit in the full-adders
- Uneven collection of the terms in the Full-adders
- Irregularity of the multiplier architecture

While the first point is applicable to all the previously-mentioned architectures, the second one holds only for the CSM structures, whereas the Wallace suffers more from the third one. Fundamental skeleton of self-timed multipliers, as analyzed. Standard full-adders are replaced by the so-called latch-adder cells, which combine the functionality of a full-adder and a latch. The output is retained until the enable signal arrives, hence actually limiting the spurious activity to the final RCA. Latch-adders are, however, ratioed cells. Therefore, transistor sizing is critical and it depends on the technology; as a consequence, minimum-size devices cannot be used. Additionally, the switching of the enable transistors entails a large energy overhead.

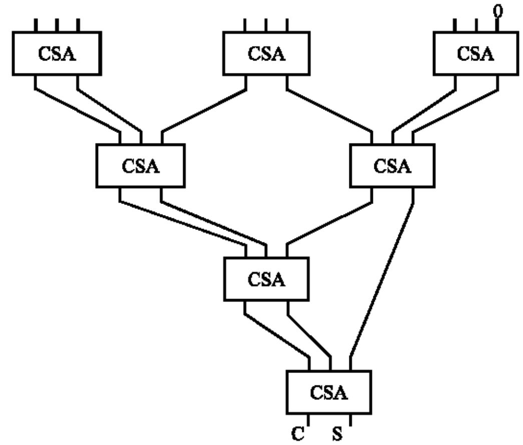


Fig. 3: Wallace carry save multiplier

WALLACE MULTIPLIER

Wallace (1964) showed that the partial products can be added in a fast way by using multiple levels of CSA shown in Fig. 4. In each level of the tree, the numbers are grouped into three. A CSA is used to add the numbers in each group. The process continues until there are only two numbers left to be added. To add these numbers, a carry propagate adder is used. Each level reduces the number of terms to be added by a factor of:

$$1.5\rightarrow O\ (\log_{1.5}n)$$

The Wallace tree was proposed by Wallace (1964). This method can be used to sum up all the bits of the partial product in each column. The summation is independent and simultaneous due to each modified Booth encoder works in parallel. It results in all bits of partial products arrive at the adder tree at the same time. Thus, the Wallace tree structure increases the speed of the multiplication by introducing parallelism. The Wallace

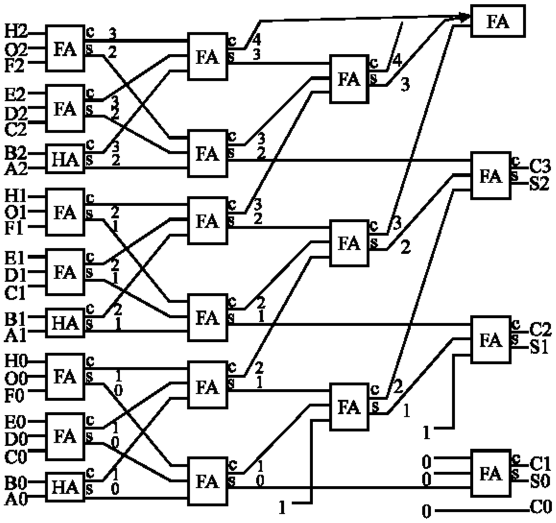


Fig. 4: Wallace multiplier

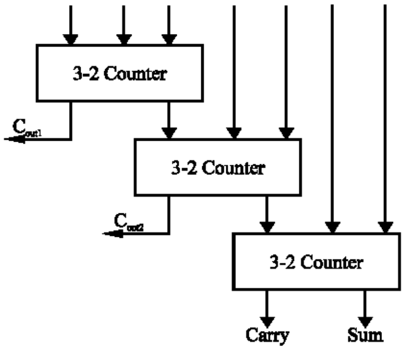


Fig. 5: Wallace tree with 3 :2 counter

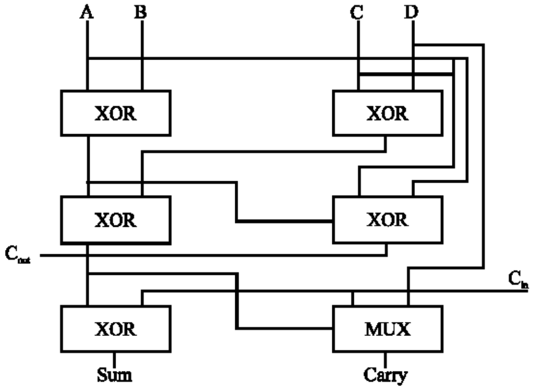


Fig. 6: Structure of 4:2 compressors

tree was first constructed by using 3-2 counters (carry save adders) shown in Fig. 5. A 3-2 counter is also called a 3-2 compressor, which has three inputs and two outputs. This counter has a maximum of two XOR delays. The Wallace tree uses 3-2 counters to sum up all the partial products with the same weight and produce two bits, one is the carry bit with the weight of n + 1 and the other is the sum bit with the weight of n. Compressors are mostly used in multipliers to reduce the operands while adding terms of partial products. A compressor C_i is a combinatorial device that compresses N input lines in the position i to 2 output lines i.e., sum and carry. The 4:2 compressors has 4 input lines i1, i2, i3 and i4 that must be summed and has two output lines s and c, which are so called results of compression. The additional lines are input and output carries. A design is developed for a multiplier which generates the product of two numbers using purely combinational logic, i.e., in one gating step as in Fig. 6. Using straight forward diode-transistor logic, it appears presently possible to obtain products. Multiplication of binary fraction is normally implemented as the addition of a number of summands; each some simple multiple of the multiplicand, chosen from a limited set of available multiplies on the basis of one or more multiplier digits. The multiplier unit requires a great deal of equipment, amounting perhaps to 10% of the to total semiconductor complement of a very large modern computer, but probably, because of its simplicity, costing rather less than 10% of the cost of the computer. In sense, this equipment is used inefficiently. It is useful for only arithmetic operations and even in these, circuits with delay times of 30 msec are used only about once per microsecond. However, some mismatch between propagation delay and repetition rate is apparently inherent in the type of circuit postulated and equally bad mismatches could probably be found in many present computers. If the word length is increased, the equipment cost rises as the square of the word length and the times as the logarithm of the word length. The inefficiency is not in tolerable.

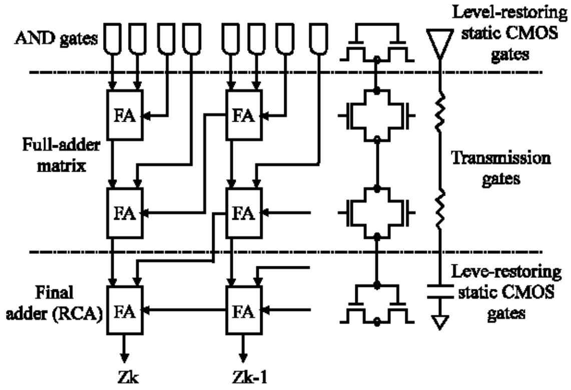


Fig. 7: Proposed TG multiplier

Proposed architecture: TG-Mult transmission gate cells, the advantages of which will be discussed latter. The full-adder cells in the final RCA are again level-restoring static CMOS gates to recover the driving capability. Therefore, from outside an electrical behavior similar to a standard (Carbognani *et al*., 2008) CMOS Wallace tree is maintained. Note, however, proposed TG-Multiplier and equivalent (as for instance in the circuits) are shown in Fig. 7. Hence, the static CMOS gates in the RCA do not need to restore the level of the full-adder matrix output Signals. TG-Mult reduces activity by 23 and 29% compared to the two Wallace-tree. The results of the previous sections can be summarized in the following four statements.

- Spurious activity limits multiplier efficiency
- Wallace reduces glitch generation and propagation
- Minimum-size transistors increase energy efficiency
- A more sophisticated approach (Chong *et al*., 2005) indeed succeeds in decreasing the spurious activity at the expense, however, of a large energy overhead and technology dependent transistor level techniques. TG-Multiplier is a simple architecture

based on the Wallace tree with minimum-size transistors. The AND gates that create the terms are implemented in level-restoring static. CMOS, which present purely capacitive inputs, hence decoupling the multiplier from the input drivers. The full-adder matrix makes use of standard 18-transistor.

SIMULATION RESULTS

Measurements of dynamic power confirm the results of transistor level simulations (Table 1 in terms of relative benefits), although simulated results tend to underestimate the measured consumption (accuracy ranges from 15 to 30%). According to measurements, the Wallace-tree multiplier dissipates less energy than the reference CSM. Further Area savings are possible by implementing minimum-size transistors. We are synthesizing all these multipliers using Xilinx. Simulation results for the array multiplier, carry save multiplier, Wallace tree multiplier and TG multiplier are shown in Fig. 8-11, respectively.

The power report of array multiplier, carry save multiplier, Wallace tree multiplier and TG multiplier are shown in Fig. 12-15, respectively. The power reports are taken by using Xilinx X-power analysis method with help of VCD file which is generated at the time of Model-Sim functional verification period and its report are tabulated in Table 1.

Table 1: Comparison of power, delay and area of various multipliers

Multiplier	Total power (mW)	Area (%)	Delay (pS)
Array multiplier	2979.60	45	300
Carry save multiplier	2108.79	45	270
Wallace multiplier	2013.06	50	200
Proposed TG based multiplier	881.52	50	100

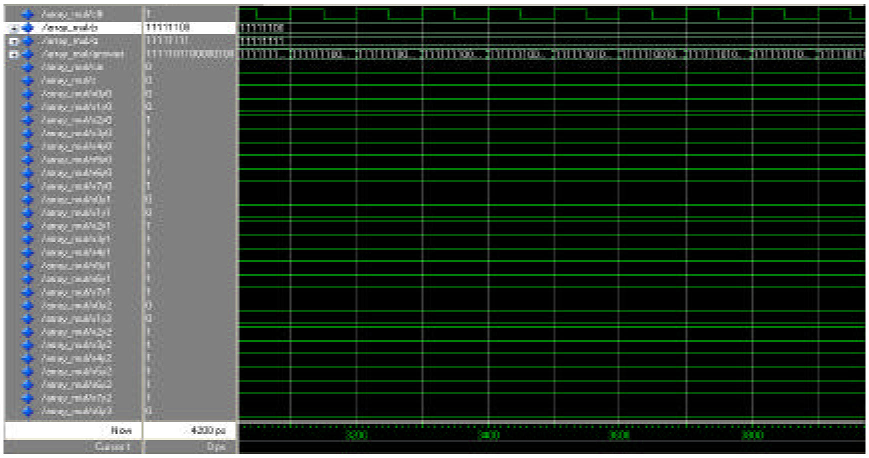


Fig. 8: Simulation results for array multiplier

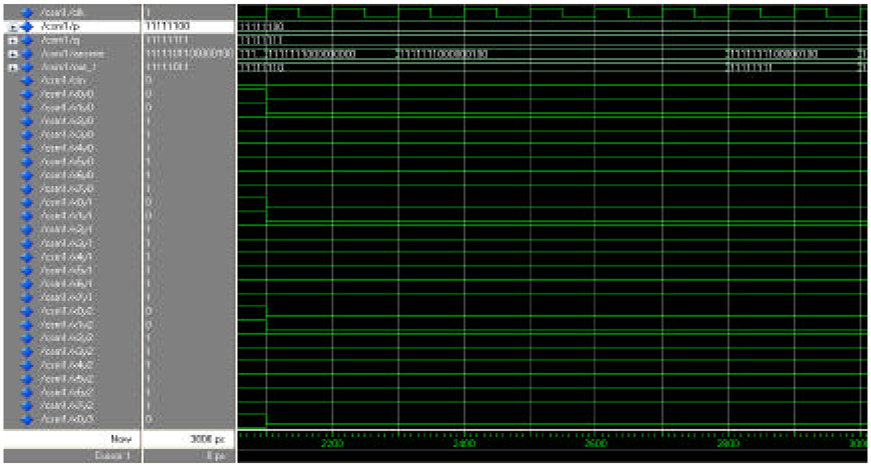


Fig. 9: Simulation results for carry save multiplier

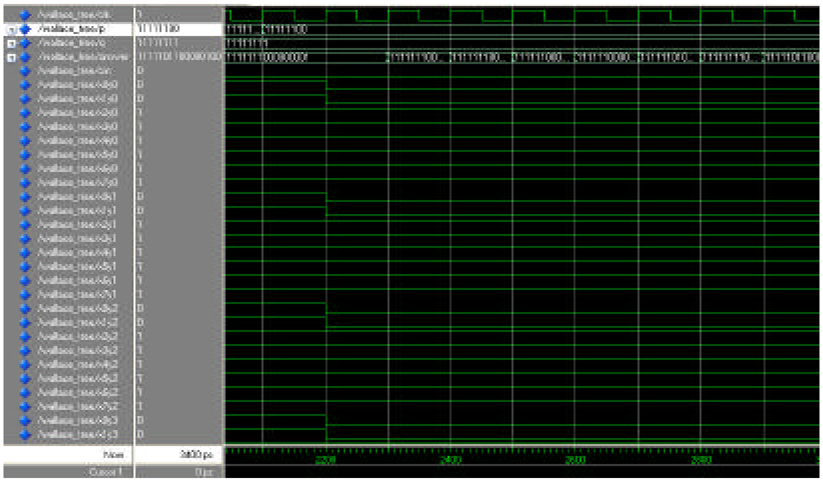


Fig. 10: Simulation results for Wallace tree multiplier

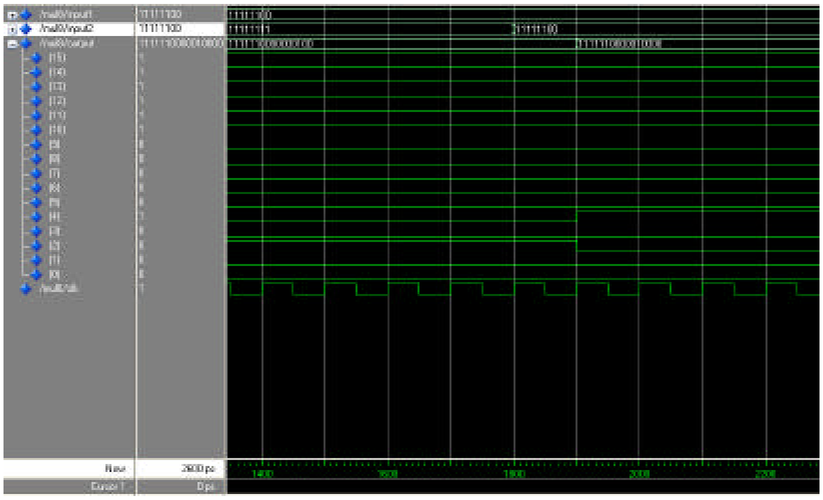
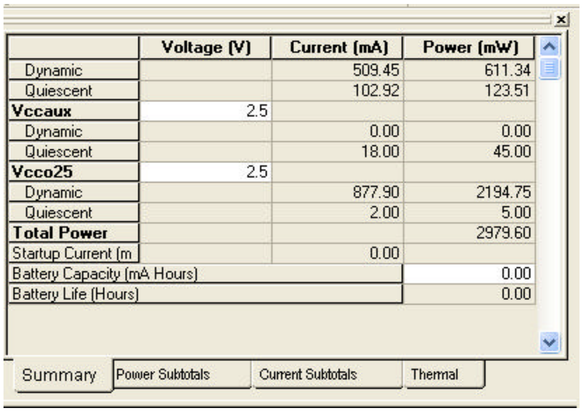
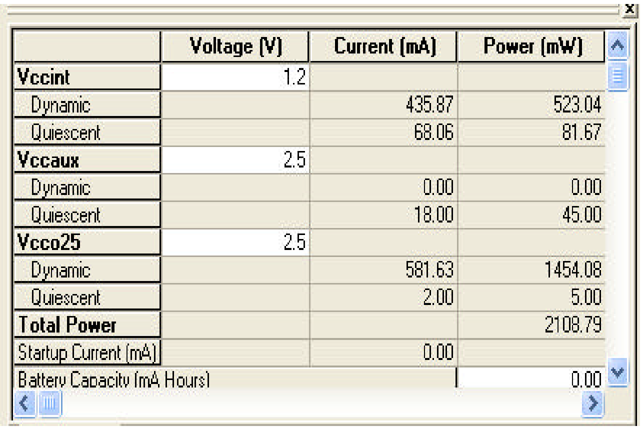


Fig. 11: Simulation results for TG multiplier



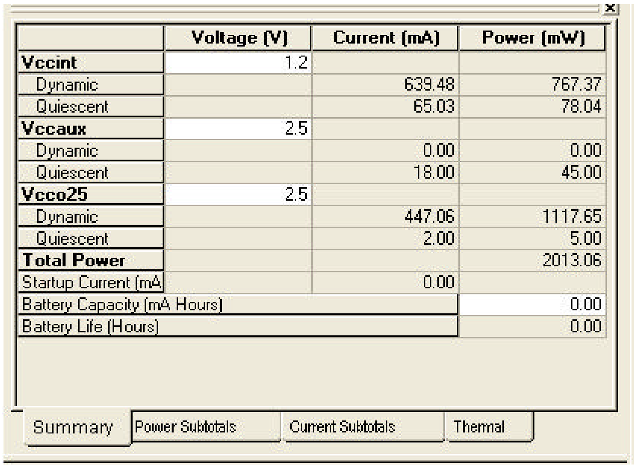
	Voltage (V)	Current (mA)	Power (mW)
Dynamic		509.45	611.34
Quiescent		102.92	123.51
Vccaux	2.5		
Dynamic		0.00	0.00
Quiescent		18.00	45.00
Vcco25	2.5		
Dynamic		877.90	2194.75
Quiescent		2.00	5.00
Total Power			2979.60
Startup Current (m		0.00	
Battery Capacity (mA Hours)			0.00
Battery Life (Hours)			0.00

Fig. 12: Power report of array multiplier



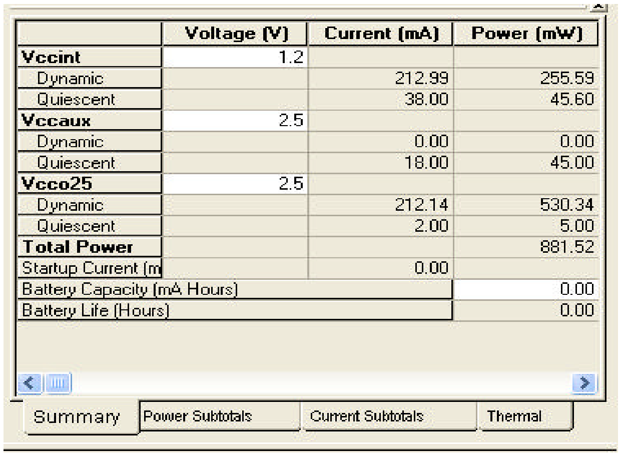
	Voltage (V)	Current (mA)	Power (mW)
Vccint	1.2		
Dynamic		435.87	523.04
Quiescent		68.06	81.67
Vccaux	2.5		
Dynamic		0.00	0.00
Quiescent		18.00	45.00
Vcco25	2.5		
Dynamic		581.63	1454.08
Quiescent		2.00	5.00
Total Power			2108.79
Startup Current (mA)		0.00	
Battery Capacity (mA Hours)			0.00

Fig. 13: Power analysis of Carry saves multiplier



	Voltage (V)	Current (mA)	Power (mW)
Vccint	1.2		
Dynamic		639.48	767.37
Quiescent		65.03	78.04
Vccaux	2.5		
Dynamic		0.00	0.00
Quiescent		18.00	45.00
Vcco25	2.5		
Dynamic		447.06	1117.65
Quiescent		2.00	5.00
Total Power			2013.06
Startup Current (mA		0.00	
Battery Capacity (mA Hours)			0.00
Battery Life (Hours)			0.00

Fig. 14: Power analysis of Wallace tree multiplier



	Voltage (V)	Current (mA)	Power (mW)
Vccint	1.2		
Dynamic		212.99	255.59
Quiescent		38.00	45.60
Vccaux	2.5		
Dynamic		0.00	0.00
Quiescent		18.00	45.00
Vcco25	2.5		
Dynamic		212.14	530.34
Quiescent		2.00	5.00
Total Power			881.52
Startup Current (m		0.00	
Battery Capacity (mA Hours)			0.00
Battery Life (Hours)			0.00

Fig. 15: Power Analysis of TG multiplier

The following two reasons allow TG-Multiplier to be robust against leakage:

- The implementation of minimum-size devices
- The reduction of the number of -to- ground paths

Similarly to Wallace, the implementation of minimum size devices results in the increase of the transistor channel resistances, hence decreasing the sub threshold currents. Substantial further power savings are due to the transmission-gate full adders, which reduce the number of Vdd -to-ground paths compared to CMOS mirror full-adders. Finally, the novel multiplier architecture is much more robust to process parameter and place-and-route variations than other glitch suppressing techniques. Compared to delay balancing and self-timed circuits, the new structure does not rely on the propagation delay of single cells. Therefore, the limited variation of transistor channel resistance or internal node capacitance may affect the RC time constant, but not the overall low-pass filtering property of transmission gates.

CONCLUSION

Multiplier energy efficiency is the result of careful tradeoffs among several, often contrasting factors, from architectural down to transistor level. The new multiplier structure introduced in this work (TG-Mult) succeeds in reducing spurious switching activity significantly without compromising the benefits with energy-hungry add-on sub circuits. Transmission gates combined with level-restoring static CMOS gates suppress glitches via RC low-pass filtering, while preserving unaltered driving capabilities. Measurements point out the proposed TG based low power multiplier has considerable energy savings over a regular Wallace architecture and more than as compared to a Wallace featuring minimum-size devices.

REFERENCES

Agrawal, V.D., 1997. Low power design by hazard filtering. Proceedings of the 10th International Conference on VLSI Design: VLSI in Multimedia Applications, Jan. 4-7, Hyderabad, India, pp: 193-197.

Alioto, M. and G. Palumbo, 2002. Analysis and comparison on full adder block in submicron technology. IEEE Trans. Very Large Scale Integr. Syst., 10: 806-823.

Buergin, F., F. Carbognani, N. Felber, H. Kaeslin and W. Fichtner, 2006. 29% power saving through semi-custom standard cell re-design in a front-end for hearing aids. Proceedings of the 49th IEEE International Midwest Symposium on Circuits and Systems, Aug. 6-9, San Juan, Puerto Rico, pp: 610-614.

Carbognani, F., F. Buergin, N. Felber, H. Kaeslin, W. Fichtner, 2006. A self timed 16 bit multiplier for low power low–frequency applications. Proceedings of the 49th IEEE International Midwest Symposium on Circuits and Systems, Aug. 6-9, San Juan, Puerto Rico, pp: 433-437.

Carbognani, F., F. Buergin, N. Felber, H. Kaeslin and W. Fichtner, 2008. Transmission gates combined with level restoring CMOS gates reduces glitches in low power low frequency multipliers. IEEE Trans. Very Large Scale Integr. Syst., 16: 830-836.

Chang, J.H., J. Gu and M. Zhang, 2005. A review of 0.18-_m full adder performances for tree structured arithmetic circuits. IEEE Trans. Very Large Scale Integr. Syst., 13: 686-695.

Chen, K.H. and Y.S. Chu, 2007. A low-power multiplier with the spurious power suppression technique. IEEE Trans. Very Large Scale Integr. Syst., 15: 846-850.

Chong, K.S., B.H. Gwee and J.S. Chang, 2005. A micropower low-voltage multiplier with reduced spurious switching. IEEE Trans. Very Large Scale Integr. Syst., 13: 255-265.

Mahant-Shetti, S.S., P.T. Balsara and C. Lemonds, 1999. High performance low power array multiplier using temporal tiling. IEEE Trans. Very Large Scale Integr. Syst., 7: 121-124.

Mosch, P., G.V. Oerle, S. Menzl, N. Rougnon-Glasson, K.V. Nieuwenhove and M. Wezelenburg, 2000. A 660-_W 50-Mops 1-V DSP for a hearing aid chip set. IEEE J. Solid-State Circuits, 35: 1705-1712.

Shams, A.M., T.K. Darwish and M.A. Bayoumi, 2002. Performance analysis of low-power 1-bit CMOS full adder cells. IEEE Trans. Very Large Scale Integr. Syst., 10: 20-29.

Sobelman, G. and D. Raatz, 1995. Low-power multiplier design using delayed evaluation. Proceedings of the IEEE International Symposium Circuits Systems, (ISCAS'95), Seattle, WA, pp: 1564-1567.

Sulistyo, J. and D. Ha, 2003. 5 GHz pipelined multiplier and MAC in 0.18 _m complementary static CMOS. Proceedings of the IEEE International Symposium on Circuits and Systems, (ISCAS'03), Bangkok, Thailand, pp: 117-120.

Uppalapati, S., M.L. Bushnell and V.D. Agrawal, 2005. Glitch-free design of low power ASICs using customized resistive feedthrough cells. Proceedings of the 9th IEEE VLSI Design and Test Symposium, (VLSIDTS'05), Bangalore, India, pp: 41-48.

Wallace, C.S., 1964. A suggestion for a fast multiplier. IEEE Trans. Comput., 13: 14-17.