IMPLEMENTATION OF LOW-POWER LOW-FREQUENCY MULTIPLIERS

S.Rakesh Sharma[1], Dr .Arunesh Kumar Yadhav[2]

[1]-Asst Professor,ECE,Sapthagiri College of Engineering,Dharmapuri.TN

abirainarakesh@yahoo.com

[2]-Asst Professor,Dept of Physics,I.T.S Engineering College,Noida,UP

Abstract -In this paper various 16-bit multiplier architectures are compared in terms of dissipated energy, propagation delay, energy-delay product (EDP), and area occupation, in view of low-power low-voltage signal processing for low-frequency applications. It is mechanisms of glitch generation and propagation. It is found that spurious activity is a major cause of energy dissipation in multipliers. Due to shorter full-adder chains, the Wallace multiplier dissipates less energy than other traditional array multipliers (8.2 W/MHz versus 9.6 W/MHz for 0.18µm CMOS technology at 0.75 V). The benefits of transistor sizing are also evaluated (Wallace including minimum-size transistors dissipates 6.2 W/MHz). By combining transmission gates with static CMOS in a Wallace architecture, a new approach is proposed to improve the energy-efficiency further (4.7 W/MHz), beyond recently published low-power architectures. The reduced number of V_{dd}-to-ground paths also contributes to a significant decrease of static consumption.

I. INTRODUCTION

Lowering down the power consumption and enhancing the processing performance of the circuit designs are the two important design challenges of wireless multimedia and digital signal processor (DSP) applications. In these applications 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.

In the constantly growing portable market it is mandatory to search for new promising low-power techniques to increase battery lifetime. This work focuses specifically on all audio (MP3 players, audio chip-sets for cellular phones,) and biomedical applications, where tight constraints on both area occupation and energy consumption are imposed, while timing requirements are relaxed. These specifications are particularly valid for hearing aids, which are normally operated at very low supply voltage. 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 [1], which can even dominate the total dynamic consumption. While trying to optimize the efficiency of multipliers, many works in the past investigated only the basic constitutive cell, namely the full-adder. This way of proceeding overlooks the previously-mentioned relevant aspect of glitch 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. Apart from that, three fundamental approaches have been proposed in the literature so far to abate glitch generation and propagation in parallel multipliers, namely:

1) Shortening full-adder chains;
2) Equalizing internal delays;
3) 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, 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. 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. For example, an independent delay line triggers special cells that implement the functionality of both a full-adder and a latch. These circuits present superior glitch suppression. 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:

1) It limits the area increase, which is relevant in ;
2) It can do without large consuming transistors, needed
3) It is more robust to process and voltage variation.

This work 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 crossover currents. Based on these results, new multiplier architecture is introduced, called TG-Mult that reduces spurious activity further compared with both traditional and recently published architectures. At the same time, TG-Mult 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.

The reminder of this paper is organized as follows. Section II introduces the operation of multiplication. Section III shows the simulation and measurement setup used throughout this work. Section IV compares recently published with traditional architectures. In the same section, the new multiplier TG-Mult is introduced. Measurements are presented in Section V. After the discussion of the results in Section VI, Section VII draws the conclusions. Eventually, the Appendix shortly describes a new practical methodology to determine spurious activity through simulations.

II. SIGNED MULTIPLICATION

Given two unsigned binary 16-bit wide numbers and, the multiplication operation is defined as follows:

$$Z = X \cdot Y = \sum_{j=0}^{15} \left(\sum_{i=0}^{15} (X_i Y_j) 2^{(i+j)} \right) \qquad (1)$$

Where Z represents the product, X_i the ith bit of the multiplicand and Y_j 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:

$$Z = \sum_{j=0}^{7} \left(\sum_{i=0}^{15} (X_i Y b_j) 2^{(i+2j)} \right) \qquad (2)$$

Where $Yb_j \in \{-2, -1, 0, 1, 2\}$ represents the jth operand of the multiplier after Booth recoding. As can be noticed from (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.

III. SIMULATION SETUP AND INTEGRATION ON SILICON

All investigated multipliers have been placed and routed employing Silicon Ensemble by Cadence, targeting a 0.18- µm process. In this way, it has been possible to extract realistic node capacitances with the Hyper Extract tool by Cadence. Four selected multiplier architectures (Wallace_ minsize, TG-Mult, Wallace, and CSM), which will be discussed in Section IV, have been integrated on silicon in such a way as to minimize the parasitics. In particular:

   1) The cell density has been kept as high as 90% in all multiplier designs;
   2) Multiplier inputs are strengthened by buffers that are located as close as possible to the multiplier cores, without contributing to their power measurement;
   3) Multiplier outputs are regenerated by means of buffers, which are again located in proximity of the cores, without being part of them.

The pad frame supply is kept distinct from the core supply. Therefore, it was possible to measure the consumption of the core separately. Additionally, independent power rings for each multiplier, supplied through analog pads, enable clean and accurate power measurements.

IV. MULTIPLIER ARCHITECTURES

A. Traditional Multiplier Architectures

Equations (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:

   1) carry-save multiplier (CSM);
   2) CSM with radix-4 Booth recoding (CSM_Booth);
   3) Wallace tree.

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 (see Section II). 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. 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. In this paper, a final ripple carry adder (RCA) has been chosen in all architectures, trading speed for energy, as demonstrated.

The first three lines of Table I summarize the relevant measured and simulated figures of the traditional multiplier architectures, implemented with regular CMOS mirror adders out of the standard cell library. The area occupation of the different multipliers does not present large variations. As expected, the Wallace tree is faster than the other traditional architectures. In terms of energy

dissipation, the Wallace tree appears more efficient. The high regularity of the CSM leads to a more compact layout.

TABLE I
AREA, DELAY, EDP AND ENERGY CONSUMPTION OF VARIOUS MULTIPLIER ARCHITECTURES (AT 0.75 V)

Architecture	Simulation				Measurements	
	Area µm²	Delay ns	EDP 10⁻²¹ Js	Total energy µW/ MHz	Dynamic energy	Static power
CSM	21000	63	523	8.3	9.6	42
CSM_Booth	23000	57	564	9.9	-	-
Wallace	21000	55	385	7.0	8.2	30
CSM-minsize	24000	64	320	5.0	-	-
CSM_ Booth_minsize	27000	54	281	5.2	-	-
Wallace_minsize	23000	52	224	4.3	6.2	9
Leapfrog	29000	58	493	8.5	-	-
Chong	41000	127	266	5.6	-	-
TG – Mult	23000	72	266	3.7	4.7	3

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:
1) Different delays of sum and carry bit in the full-adders;
2) Uneven collection of the terms in the full-adders;
3) 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.

The approach described in the Appendix allows, for each multiplier architecture, the estimation of the average switching activity, the functional activity, and the spurious activity caused by glitches inside the matrix of full-adders, as shown in Table II. This demonstrates that not all discussed points play the same role in the generation of spurious activity. The first three lines show the activities in the traditional architectures, averaged over all sum and carry bits. They give valuable indications on the glitch propagation along the full-adder chains. The functional switching activity is nearly the same in all structures. As CSM and Wallace require almost the same number and type of standard cells, they would also dissipate the same amount of energy, if the spurious activity were zero. Therefore, the improved efficiency of the Wallace is the consequence of the full-adder network organization that is less prone to glitch generation.

TABLE II
AVERAGE SWITCHING ACTIVITIES IN VARIOUS MULTIPLIER ARCHITECTURES (SEE THE APPENDIX)

architecture	α_F	α_S	α_{TOT}	α_S/α_{TOT}
CSM	0.42	1.03	1.45	71%
CSM_Booth	0.42	1.50	1.92	78%
Wallace	0.40	0.68	1.08	63%
Wallace_minsize	0.40	0.60	1.00	60%
Leapfrog	0.44	1.68	2.12	79%
Chong	0.42	**0.03**	**0.45**	**7%**
TG-Mult	0.40	0.37	0.77	48%

While Table II gives average figures, the spurious activity diagrams (see Fig. 2) represent the activity of glitches of all internal signals in the full-adder networks (see the Appendix). A general observation is that sum bits are more glitch prone than carry bits, because summing is equivalent to an EXOR, the output of which changes each time an input toggles (see Fig. 2(a) for the CSM with Booth recoding). Therefore, spurious activity typically rises along vertical paths. Booth recoding entails relevant glitch propagation, as shown in Fig. 2(a). The reason is mainly the encoding unit above the full-adder matrix, which creates a significant amount of glitches starting already in the first row. The Wallace tree is the most efficient multiplier among the analyzed traditional architectures, but it still suffers from 63% spurious activity according to simulations (see Table II).

B. Traditional Architectures with Minimum-Size Transistors

In the context of low-voltage low-frequency applications, suggests that about 30% power savings are possible by implementing minimum-size devices. Rows four to six in Table I show that traditional architectures implemented with minimum-size transistors are indeed much more efficient and, in some cases, even faster with the given constraints. The unexpected increase in area results from suboptimal cell layout due to design rule constraints. Minimum-size transistors slightly attenuate glitch propagation; because they increase the transistor channel resistances (see Section VI). Yet, spurious activity still represents 60% of the total activity in the Wallace with minimum-size devices [see Fig. 2(c)], as shown in the fourth row of Table II.

C. Recently Published Architectures

Since spurious activity propagation is the most limiting factor to multiplier efficiency, different low-power techniques have been developed. In this section, two architectures based on CSM with Booth recoding are analyzed, namely the Leapfrog proposed in [9] and the Chong introduced in [1]. The Leapfrog architecture aims at equalizing the internal signal propagation delays.
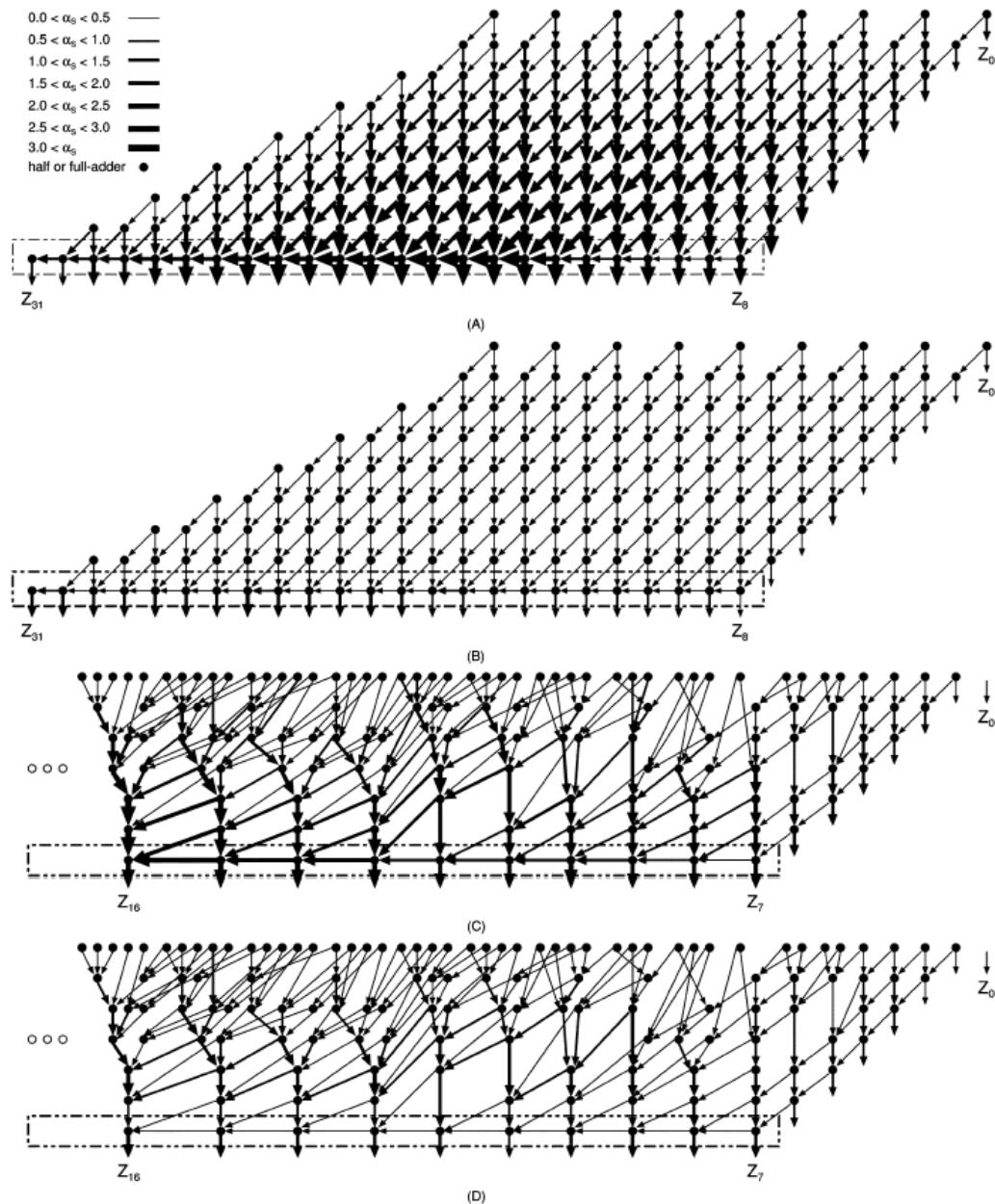
Fig. 2. Spurious activity diagrams: (a) CSM_Booth; (b) Chong; (c) Wallace_minsize; and (d) TG-Mult.

It starts from the observation that in the Leapfrog full-adder cell, the propagation delay to both the sum and the carry out from the carry in port is half that from the other two inputs. As shown in Fig. 3(a), the sum bit bypasses Row 1 and is directly fed into Row 2, whereas the carry bit regularly ripples through Row 1 down to Row 2. The basic idea is to provide an extra time to the slow sum bit by skipping a row, so that it can catch up with the faster carry output. The energy efficiency of this architecture is nevertheless dependent on the technology, on the process, and on the parasitic loads. Whenever the initial assumption on the propagation delays cannot be granted, the Leapfrog remains vulnerable to glitch generation and propagation, as shown

in Table II.The Chong multiplier implements the fundamental skeleton of self-timed multipliers, as analyzed. Standard full-adders are replaced by the so-called latch-adder cells [see Fig. 3(b)], 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 [see Fig. 2(b)]. 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.
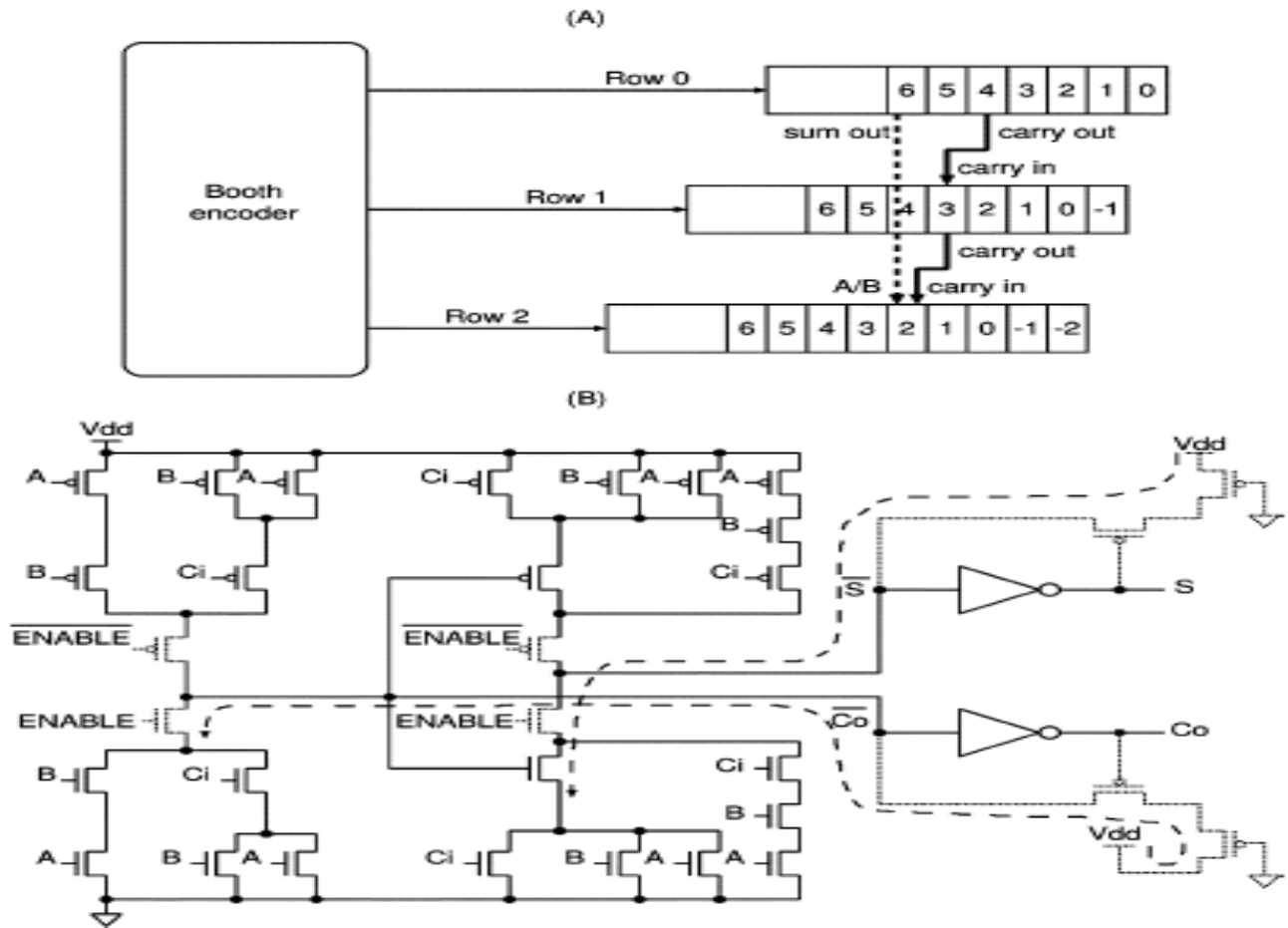
Fig. 3. (a) Key concept of the Leapfrog architecture [9]. (b) Latch–adders implemented in Chong [1]. The dotted components represent the add-on to the regular mirror adder cell. The dashed lines show potential race conditions.

These reasons prevent the architecture from fully exploiting the benefits of the almost complete glitch suppression (as shown in Table II).

D. Proposed Architecture: TG-Mult

The results of the previous sections can be summarized in the following four statements:
1) Spurious activity limits multiplier efficiency;
2) Wallace reduces glitch generation and propagation;
3) minimum-size transistors increase energy efficiency;
4)A more sophisticated approach (Chong) indeed succeeds in decreasing the spurious activity at the expense, however, of a large energy overhead and technology dependent transistor level techniques.

TG-Mult (see Fig. 4) 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 transmission gate cells, the advantages of which will be discussed in Sections V and VI.
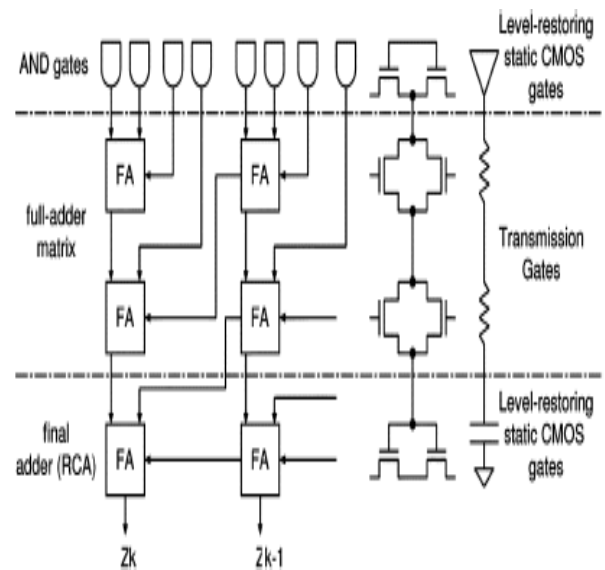

Fig. 4. Proposed TG-Mult and equivalent circuits for glitch filtering.

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 CMOS Wallace tree is maintained. Note, however, that transmission Fig. 4.

Proposed TG-Mult and equivalent circuits for glitch filtering. Gates pass full-swing signals, as opposed to pass-transistor logic (as for instance in the circuits proposed on [18, pp. 203–214]).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 implementations respectively [compare Fig. 2(d)–(c)]. Final switching activity figures are collected in the last row of Table II.

V. MEASUREMENT RESULTS

Measurements of dynamic power confirm the results of transistor level simulations (see Table I) 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 about 15% less energy than the reference CSM. Further 24% savings are possible by implementing minimum-size transistors. Yet, the proposed TG-Mult is by far more efficient than all the other architectures.

Table I also collects the average leakage consumption for all multipliers that have been implemented on silicon. CSM is the architecture that shows the largest static dissipation, while TG-Mult is by far the most leakage resistant architecture. The reasons are discussed in Section VI. Table I shows that at 1MHz frequency the static power is at most 0.4% of the total dissipation (CSM). While this appears negligible in most cases, it is worth observing that in sub-blocks with long stand-by intervals, leakage can turn out to be the dominant contribution to the overall energy dissipation. TG-Mult fits particularly well in such applications too.

VI. DISCUSSION

A conductive transmission gate acts, in first approximation, as a resistance (see [15, Par. 4.2.3]), the value of which is dependent on the working region of the two transistors. In the given technology, the resistance ranges from about 15 kΩ to almost 60 kΩ for a TG with minimum-size transistors. By connecting it to a typical node load of 10 fF, an equivalent *RC* filters with a time constant of few hundreds of picoseconds results. When several transmission gates are cascaded, the time constant easily reaches a few nanoseconds, enough to filter out the majority of glitches. In Fig. 5, the same internal carry bit signal is evaluated over a short time window in the Wallace_minsize architecture and the proposed TG-Mult.
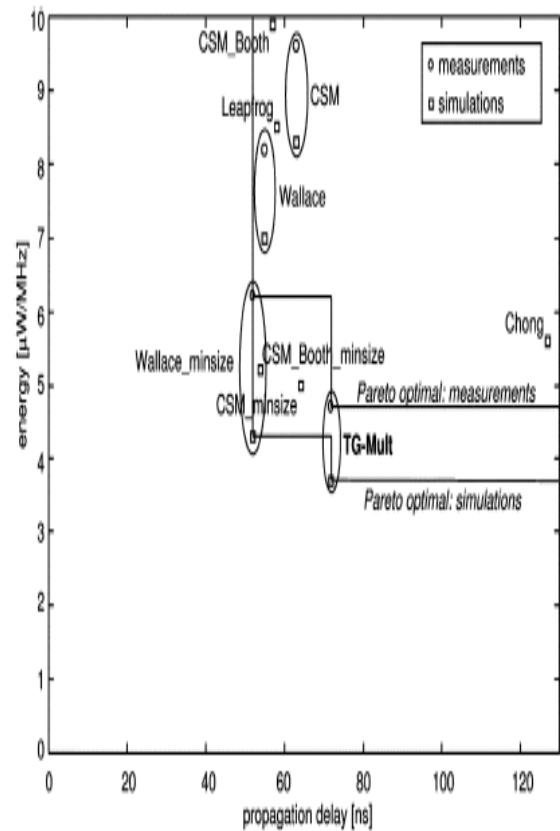

Fig. 6. Energy-delay graph of TG-Mult, compared to traditional and recently published architectures, Chong [1] and Leapfrog [9].

The waveforms show that the number of full-excursion glitches is two in the first one and goes to zero in TG-Mult, thanks to the much slower signal transition time. Large transition times normally lead to high crossover current dissipation. Yet, the low supply voltage makes this side effect negligible. The following two reasons allow TG-Mult to be robust against leakage:

1) The implementation of minimum-size devices;
2) The reduction of the number of -to-ground paths.

Similarly to Wallace_ minsize, 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 -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.
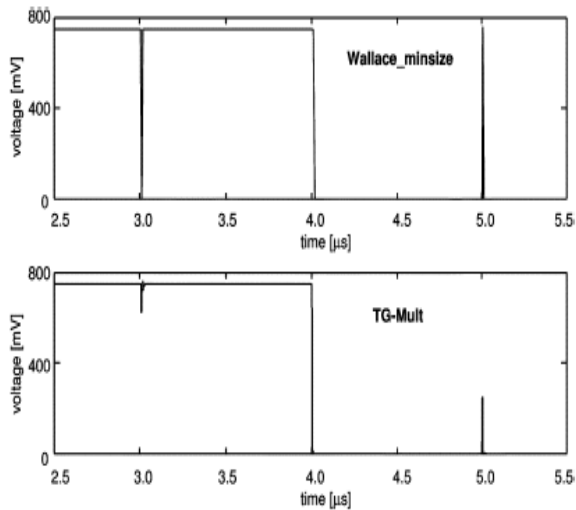
Fig. 5. Internal carry signal with glitches in Wallace_minsize and TG-Mult.

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.

VII. 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 43% energy savings over a regular Wallace architecture and more than 24% compared to a Wallace featuring minimum-size devices. Additionally, simulations show 34% energy savings over Chong [1] and 56% over Leapfrog [9]. The price to be paid is a delay overhead, which appears acceptable in low-frequency audio applications. An exhaustive overview is given in Fig. 6: TG-Mult provides anew and significantly better Pareto-optimal circuit. The EDP is the second best after Wallace _minsize. A limited area overhead compared to the traditional Wallace architecture should also be considered.

APPENDIX

The total switching activity indicates the average number of transitions per computation period that a signal under goes. It can be decomposed in two contributions

$$\alpha_{TOT} = \alpha_F + \alpha_S \qquad (3)$$

where α_F denotes the functional activity and the spurious activity, caused by glitches.

The value of has been determined by monitoring the total switching activity of a gate-level simulation, after having forced the propagation delay of all standard cells to zero. As a matter of fact, the spurious activity is the consequence of different delays in reconvergent fan out paths; hence, by forcing the cell delays to zero, (3) reduces to the mere functional activity ($\alpha_{TOT}= \alpha_F$), which can be easily determined. The value of is instead deduced from transistor-level simulations, by counting the number of crossings through. The average spurious activity results then by subtraction. The spurious activity diagrams (see Fig. 2) are representations of the generation and propagation of glitches. The dots represent the full- (and half-) adders, while the arrows represent the sum and the carry bit lines: the thicker the arrows, the larger the spurious activity. Spurious activity diagrams enable complete understanding of the glitch distribution.

REFERENCES

[1] K.-S. Chong, B.-H. Gwee, and J. S. Chang, "A micro power low-voltage multiplier with reduced spurious switching," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 13, no. 2, pp. 255–265, Feb. 2005.

[2] P. Mosch, G. v. Oerle, S. Menzl, N. Rougnon-Glasson, K. v. Nieuwenhove, and M. Wezelenburg, "A 660-_W 50-Mops 1-V DSP for a hearing aid chip set," IEEE J. Solid-State Circuits, vol. 35, no. 11, pp. 1705–1712, Nov. 2000.

[3] M. Alioto and G. Palumbo, "Analysis and comparison on full adder block in submicron technology," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 10, no. 6, pp. 806–823, Dec. 2002.

[4] J.-H. Chang, J. Gu, and M. Zhang, "A review of 0.18-_m full adder performances for tree stuctured arithmetic circuits," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 13, no. 6, pp. 686–695, Jun. 2005.

[5] A. M. Shams, T. K. Darwish, and M. A. Bayoumi, "Performance analysis of low-power 1-bit CMOS full adder cells," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 10, no. 1, pp. 20–29, Feb. 2002.

[6] J. Sulistyo and D. Ha, "5 GHz pipelined multiplier and MAC in 0.18 _m complementary static CMOS," in Proc. IEEE International Symposium on Circuits and Systems (ISCAS'03), Bangkok, Thailand, May 2003, pp. 117–120.

[7] C. S. Wallace, "A suggestion for a fast multiplier," IEEE Trans. Comput., vol. 13, no. 1, pp. 14–17, Feb. 1964.

[8] P. C. H. Meier, R. A. Rutenbar, and L. R. Carley, "Exploring multiplier architecture and layout for low power," in Proc. IEEE Custom Integr. Circuits Conf. (CICC), May 1996, pp. 513–516.

[9] S. S. Mahant-Shetti, P. T. Balsara, and C. Lemonds, "High performance low power array multiplier using temporal tiling," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 7, no. 2, pp. 121–124, Mar. 1999.

[10] G. E. Sobelman and D. L. Raatz, "Low-power multiplier design using delayed evaluation," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), Seattle, WA, Apr. 1995, pp. 1564–1567.

[11] F. Carbognani, F. Buergin, N. Felber, H. Kaeslin, and W. Fichtner, "A self-timed 16-bit multiplier for low-power low-frequency applications," in Proc. Mid-West Symp. Circuits Syst. (MWSCAS), San Juan, Puerto Rico, Aug. 2006, pp. 433–437.

[12] V. D. Agrawal, "Low-power design by hazard filtering," in Proc. 10th Int. Conf. VLSI Des., Hyderabad, India, Jan. 1997, pp. 193–197.

[13] S. Uppalapati, M. L. Bushnell, and V. D. Agrawal, "Glitch-free design of low power ASICs using customized resistive feedthrough cells," in Proc. 9th IEEE VLSI Des. Test Symp. (VDAT), Bangalore, India, Aug. 2005, pp. 41–48.

[14] F. Buergin, F. Carbognani, N. Felber, H. Kaeslin, and W. Fichtner, "29% power saving through semi-custom standard cell re-design in a front-end for hearing aids," in Proc. Mid-West Symp. Circuits Syst. (MWSCAS), San Juan, Puerto Rico, Aug. 2006, pp. 610–614.

[15] J. M. Rabaey, Digital Integrated Circuits. Upper Saddle River, NJ: Prentice-Hall, 1996.

[16] B. Parhami, Computer Arithmetic. New York: Oxford University Press, 2000.

[17] A. D. Booth, "A signed binary multiplication technique," Quarterly J. Mechan. Appl. Math., vol. 4, pp. 236–240, Jun. 1951.

[18] A. Bellaouar and M. I. Elmasry, Low-Power Digital VLSI Design—Circuits and Systems. Norwell, MA: Kluwer, 1995.