

# Design and Implementation of a $5 \times 5$ trits Multiplier in a Quasi-Adiabatic Ternary CMOS Logic

Diego Mateo and Antonio Rubio

**Abstract**—*Adiabatic switching* is a technique to design low-power digital IC's. Fully adiabatic logics have expensive silicon area requirements. To solve this drawback, a quasi-adiabatic ternary logic is proposed. Its basis is presented, and to validate its performance, a  $5 \times 5$  ternary digit multiplier is designed and implemented in a  $0.7\text{-}\mu\text{m}$  CMOS technology. Results show a satisfactory power saving with respect to conventional and other quasi-adiabatic binary multipliers, and a decrease of the area needed with respect to a fully adiabatic binary one.

**Index Terms**—Adiabatic switching, low-power digital design, low-power multiplier, ternary CMOS logic.

## I. INTRODUCTION

THE design of digital very low-power integrated circuits has become a strategic topic of research [1]. Different techniques at the different levels of the design can be applied to achieve low consumption. One of these techniques is adiabatic switching [2]–[6], which is based on two basic principles: slowing down the transport of charge, and recovering the charge stored in the parasitic capacitors. The different adiabatic logics that have been developed until now can be classified as *fully adiabatic* logics [2], [3] and nonfully or *quasi-adiabatic* logics [4], [6]. The advantage of the first logics over the others is their smaller consumption, and the disadvantage is the increase of the silicon area required, due basically to the implementation of computational reversibility needed to obtain recovery of charge [2]. An alternative to this problem is presented in this paper: a *quasi-adiabatic ternary* (QAT) CMOS logic is proposed in order to obtain the ternary circuit benefits of reducing the area [7]. The consumption of the QAT logic is similar or smaller than the dissipation in other quasi-adiabatic logics.

The basis of the logic is presented in Section II. In Section III, the implementation and measurement of a  $5 \times 5$  trits (ternary digits) multiplier are shown, and in Section IV, its performances are compared with those from other multipliers. In Section V, the conclusions are summarized.

## II. QAT CMOS LOGIC

In this section, the basis of the logic is presented: first, its basic cells; then, how to interconnect them to build a more

Manuscript received October 30, 1997; revised November 24, 1997. This work was supported by the Spanish Research Commission (CICYT) under Project Contract TIC95-0469.

The authors are with the Department of Electronic Engineering, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain.

Publisher Item Identifier S 0018-9200(98)03106-0.

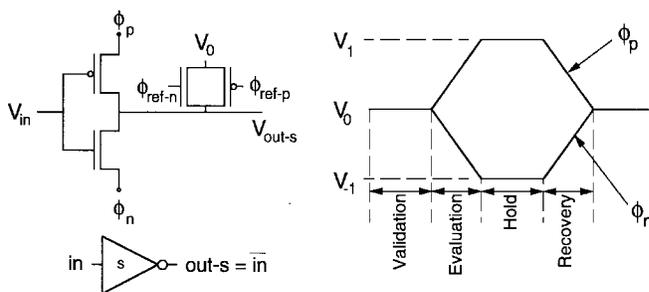


Fig. 1. Structure, symbol, and clocks of the STI (simple ternary inverter).

complex system; and finally, the existing compromise between noise margins and energy consumption.

### A. Basic Cells

Adiabatic logics usually consider four basic phases in one computational cycle for each logic stage (see Fig. 1):

- 1) input *validation*;
- 2) output *evaluation*: power supplies (clocks) are activated by slow ramp signals, computing the input information;
- 3) *hold*: the value of the output is read by the next gate;
- 4) output *recovery*: the clocks are deactivated, returning the output to its previous value.

In the QAT logic, the same four phases are used, but are adapted to the ternary valuation. The algebra used to implement the ternary valuation is the Yoeli–Rosenfeld algebra [8], which allows easily integrated CMOS implementations [9]. The three logic levels (“−1,” “0,” “1”) are represented by the voltage levels  $V_{-1}$ ,  $V_0$ , and  $V_1$ , respectively, with  $V_1 > V_0 > V_{-1}$ . Ternary gates presented here are based on the dynamic ternary gates shown in [9]. They are made from conventional binary CMOS structures with a maximum positive power supply voltage ( $V_1$ ) chosen in such a way that, when an intermediate voltage ( $V_0$ ) is applied to the input of the gate, both types of transistors  $p$  and  $n$  are off. This condition implies that

$$\begin{aligned} (V_1 - V_0) &= |V_{tp}| - \Delta_p \\ (V_0 - V_{-1}) &= V_{tn} - \Delta_n \end{aligned} \quad (1)$$

where  $\Delta_n, \Delta_p \geq 0$ .

Special nonconstant power supplies are used to give and recover the energy used to compute. They use slow ramp signals to achieve adiabatic transfer of charge. The simple ternary inverter (STI) is implemented from a conventional CMOS inverter by using the clock signals  $\phi_p$  and  $\phi_n$  as

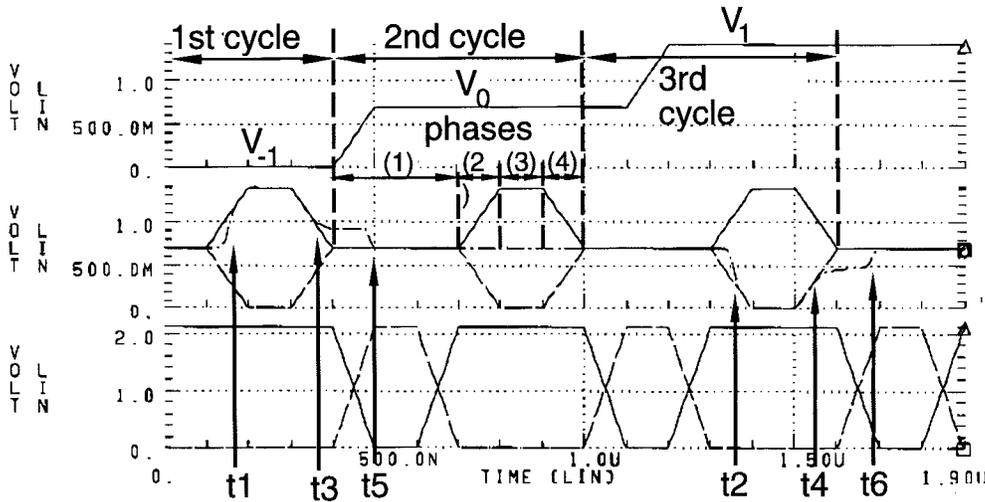


Fig. 2. Three different computational cycles of the STI for the three different inputs “-1,” “0,” and “1.”

positive and negative power rails (Fig. 1). In Fig. 2, three computational cycles of the STI are shown. When  $V_{in} = V_0$  (second cycle), both transistors are off, and the output remains at its precharged voltage ( $V_0$ ) after the evaluation phase. When  $V_{in} = V_{-1}$  (first cycle), the PMOS is on, and therefore  $V_{out}$  follows  $\phi_p$  from  $V_0$  to  $V_1$  in the evaluation phase. When  $V_{in} = V_1$  (third cycle), the NMOS is on, and  $V_{out}$  follows  $\phi_n$  from  $V_0$  to  $V_{-1}$ .

Because of the nonzero values of  $\Delta_n$  and  $\Delta_p$ , switching is not fully adiabatic since  $V_{ds}$  (in  $n$  and  $p$  transistors) may be different from 0 in the evaluation phase ( $t_1$  and  $t_2$  in Fig. 2). Two parts may be distinguished in a switching of this logic: the first one is nonadiabatic, and its energy waste, taking  $V_{tn} = |V_{tp}|$  and  $\Delta_n = \Delta_p = \Delta$ , is

$$E_{nonad} \simeq \frac{1}{2} C_L \Delta^2. \tag{2}$$

The second one is fully adiabatic, and its energy waste has the typical dependence in adiabatic circuits of  $T^{-1}$  [2] [see (3)].

Moreover, because of  $\Delta_n$  and  $\Delta_p$ , the output voltage does not return to the desired precharge value in the recovery phase ( $t_3$  and  $t_4$  in Fig. 2). A refreshment technique is proposed to solve this requirement. A CMOS transmission gate is used to precharge the output, activated by  $\phi_{ref-p}$  and  $\phi_{ref-n}$ . In the same figure, the output is precharged in moments  $t_5$  and  $t_6$ . The energy waste in this nonadiabatic transport of charge is  $(1/2)C_L\Delta^2$ .

The structures and power supplies of the positive and negative ternary inverters (PTI and NTI) are shown in Fig. 3. Their behavior is similar to the STI, with small differences. The precharge value of the positive TI is the high level  $V_1$ , so the positive rail is attached to  $V_1$ , and the negative rail goes from  $V_1$  to  $V_{-1}$  in the evaluation phase, and back to  $V_1$  in the recovery phase. To diminish the nonadiabatic switching, a transmission gate is used in the negative net instead of a single NMOS transistor. The negative TI is symmetrical to the positive one. Then, both PTI and NTI need, at the input, both signals  $in$  and its complementary  $\bar{in}$  ( $\bar{in} = STI(in)$ ). As explained in the next paragraph, PTI and NTI are only

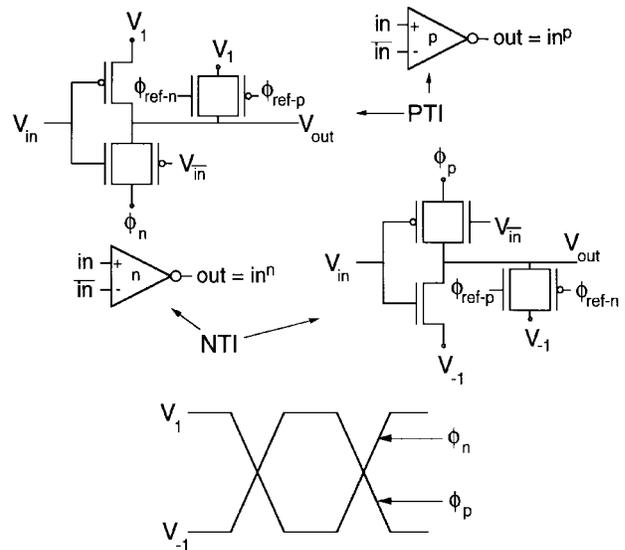


Fig. 3. Structures, symbols, and clocks of the NTI and PTI.

used in the decoder block, and they always have both signals available at their inputs.

In Fig. 4, the structure of the implementation of a generic function in the QAT logic is shown. Variables  $x_i^{-1}$ ,  $x_i^0$ ,  $x_i^1$ , and  $x_i^{-10}$  are four unary functions of each input variable  $x_i$  [9], and they are generated by using a *decoder*, which is implemented from two PTI's, two NTI's, and two STI's. The *output block* is implemented by a one-level structure, where the  $n$  and  $p$  nets are, in general, not complementary:  $n$  net implements the low level (“-1”), and  $p$  net implements the high level (“1”); when both nets are off, the output will remain at the intermediate level (“0”). Therefore, high and low levels are static (the output is clamped at  $V_1$  or  $V_{-1}$ ), but the intermediate level is dynamic (the output is in high impedance).

In adiabatic logics, each signal is activated and deactivated in each computational cycle. Then, if some information is needed in any moment after its generation, it must be delayed. To do that, a *delay cell* (DC) is used, which is implemented from two cascaded STI's.

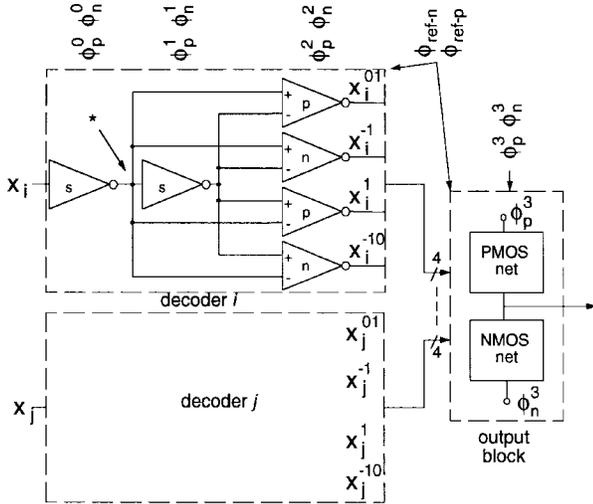


Fig. 4. Structure of a generic function in the QAT logic.

**B. Interconnection of QAT Gates**

Pipeline techniques are used when adiabatic basic gates are interconnected to build complex functions in order to have a good throughput [2]. Different adiabatic pipelines have been previously implemented, using between 1 [10] and 48 clocks [11] (in the first case, less area is needed, but the dissipation saving is small). In order to recover the stored charge, computational reversibility is applied. Breaking the reversibility at some points saves area (it is not necessary to implement a fully reversible computer), but it produces an extra waste of energy at these points, from where charge cannot be recovered. In QAT logic, the computation is done quasi-adiabatically in each block by a *local retractile cascade* that uses ten clocks (Fig. 5), and reversibility is broken at the end of the block. Therefore, the charge stored at the first node of each gate (marked with “\*” in Fig. 4) is not recovered, and its energy is dissipated. When doing the layout, this node must be carefully designed in order to minimize its capacitance.

Any system in this logic can be implemented using a pipeline of two phases. The clocks  $\phi_p^2$  and  $\phi_n^2$  of phase 1 are common to  $\phi_p^1$  and  $\phi_n^1$  of phase 2, so the total number of clocks is 18 and not 20.

**C. Logic Levels, Power Consumption, and Noise Margins**

In fully adiabatic logics, the energy needed to carry out one switching is [2]

$$E_{\text{ful-adi}} = \frac{RC_L}{T} C_L V^2. \quad (3)$$

In some quasi-adiabatic logics [4], this energy is

$$E_{\text{qua-ad-1}} \simeq \frac{1}{2} C_L V_d V \quad (4)$$

where  $V_d$  is the drop voltage of the diodes used to precharge the nodes. And in other quasi-adiabatic logics [6], [10], the energy is given by

$$E_{\text{qua-ad-2}} \simeq \frac{1}{2} C_L V_t^2 \quad (5)$$

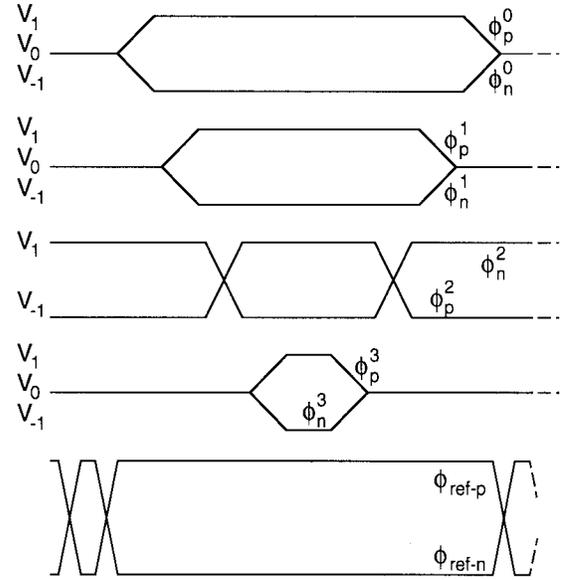


Fig. 5. Computing sequence of the power supplies for one phase of the pipeline.

where  $V_t$  is the threshold voltage. In QAT logic, the energy used to switch the one node has a similar expression:

$$E_{\text{QAT}} \simeq \frac{1}{2} C_L \Delta^2 \quad (6)$$

where  $\Delta$  is smaller than  $V_t$ . Choosing values for  $\Delta_p$  and  $\Delta_n$  as small as possible, the consumption is minimized. But there is a limit, related to the noise margin, as is shown next.

In any ternary logic, different *noise margins* must be defined. In QAT logic, static and dynamic outputs are used. The noise margins of the static outputs are related to the threshold voltages of the transistors ( $V_{t_n}, V_{t_p}$ ). The noise margins of the dynamic nodes are related to the parameters  $\Delta_n$  and  $\Delta_p$ , previously presented, and smaller than  $V_t$ 's. These are therefore the noise margins taken into account. Specifically, the worst noise margin is for the STI situated in the input of a delay cell when it has an intermediate level “0” at the input. This gate has the minimum noise margin because its output is in high impedance for a longer time than the output of the other cells (due to the timing used for the clocks). To obtain an analytical expression of the noise margins of that STI, the subthreshold current model given in [12] is used. The two noise margins (positive and negative) of this STI having a “0” at the input are

$$NM_{00}^+ = \Delta_n + \phi_t \left( n_n \ln \left( \frac{C_L}{10T} \frac{n_n \phi_t}{(W/L)_n I_{D0_n}} \right) - n_p \right) \quad (7)$$

$$NM_{00}^- = \Delta_p + \phi_t \left( n_p \ln \left( \frac{C_L}{10T} \frac{n_p \phi_t}{(W/L)_p I_{D0_p}} \right) - n_n \right) \quad (8)$$

where  $T$  is the period of charge/decharge ( $10T$  is the time that the output is in high impedance),  $\phi_t$  is the thermal voltage, and  $C_L$  is the total capacitance at the output of the STI. A simple approximation for these two expressions is

$$NM_{00}^+ \simeq \Delta_n \quad (9)$$

$$NM_{00}^- \simeq \Delta_p. \quad (10)$$

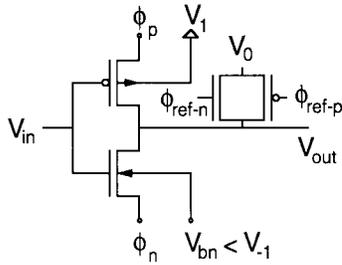


Fig. 6. Control of  $V_{tn}$  by using the body effect in order to have symmetrical logic levels.

To guarantee a minimum noise margin,  $\Delta_n$  and  $\Delta_p$  must be greater than a certain minimum. Therefore, there is a tradeoff between noise margin and energy consumption. Values about 0.3 V have been used for  $\Delta$ 's in experimentation, and with a proper design of the layout, we think it is possible to use lower values.

For a technology with a threshold voltage of 1 V, the energy saved when switching a node, with respect to other quasi-adiabatic logics (5), is

$$\frac{E_{\text{qua-adi}}}{E_{\text{QAT}}} \approx \left(\frac{V_t}{\Delta}\right)^2 \approx 10. \quad (11)$$

Once  $\Delta_n$  and  $\Delta_p$  have been chosen, for fixed threshold voltages, logic levels are obtained from (1):

$$\begin{aligned} V_0 &= V_{-1} + V_{tn} - \Delta_n \\ V_1 &= V_{-1} + V_{tn} - \Delta_n + |V_{tp}| - \Delta_p \end{aligned} \quad (12)$$

where  $V_{-1}$  can be considered as the reference voltage. To have freedom in the choice of logic levels, it is necessary to have access to the technology in order to control the values of the threshold voltages. Another possibility that gives a certain freedom when choosing the voltage levels is to modify the threshold voltages by using the body effect. In the case of the multiplier that has been implemented, the threshold voltages of the technology used (ATMEL-ES2 *ecpd07*) are  $|V_{tp0}| = 1$  V and  $V_{tn0} = 0.85$  V. Considering equal values for  $\Delta_n$  and  $\Delta_p$ , nonsymmetrical logic levels are obtained (12). To have symmetrical levels,  $V_{tn0}$  is increased by using the body effect, as shown in Fig. 6.

### III. IMPLEMENTATION OF A MULTIPLIER

In order to validate the functionality of the QAT logic and evaluate its performance, a  $5 \times 5$  trits multiplier has been implemented in a  $0.7 \mu\text{m}$  double-metal single-poly CMOS technology. The IC has a total area of  $2.6 \text{ mm}^2$ , and its photo can be seen in Fig. 7. A Wallace tree has been used to implement the multiplier. In Fig. 8, the structure of a  $3 \times 3$  trits multiplier is shown. The consumption of the multiplier is measured by using a Tek-DSA602A digitizing signal analyzer by sampling the voltage and current of each clock. In Fig. 9, *out* is an output of the multiplier, giving "1" as a result;  $\phi_{p32}$  and  $\phi_{n32}$  are the clocks  $\phi_p$  and  $\phi_n$  of the output block corresponding to the last level of the multiplier;  $i_{p32}$  is the voltage over a resistance of 1 k $\Omega$  of the current delivered by the clock  $\phi_{p32}$ , and  $P_{p32}$  is the corresponding power. The

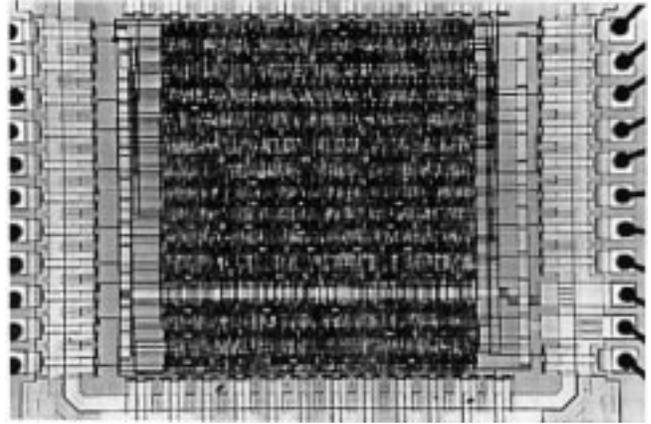


Fig. 7. Photograph of the IC.

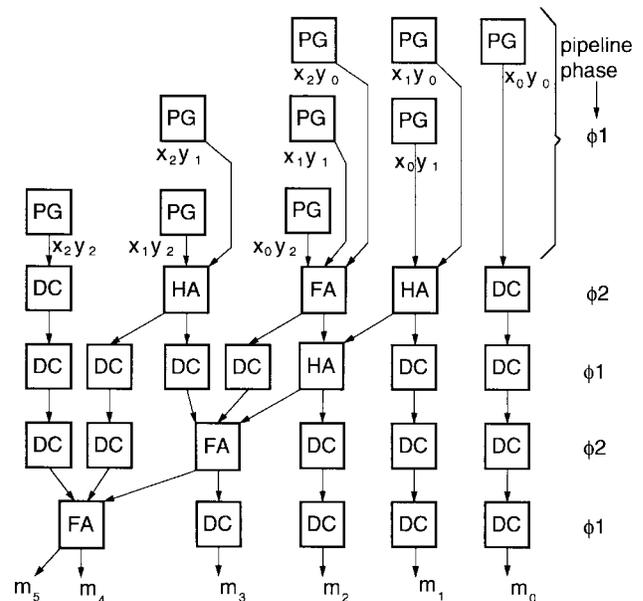


Fig. 8. Structure of a  $3 \times 3$  trits multiplier. PG are product generator cells, HA are half adders cells, FA are full adder cells, and DC are delay cells.

energy dissipated in the IC in this switching, and delivered by the mentioned clock, is the difference between the energy given by the clock ( $E_{in}$ ) and the energy returned to it ( $E_{ret}$ ). Doing that for all of the clocks, the total consumption of the QAT multiplier is obtained.

### IV. COMPARISON OF THE QAT LOGIC WITH OTHER LOGICS

In order to compare the performances of the QAT logic with other logics, in Table I, the QAT multiplier is compared with four other multipliers. These are  $8 \times 8$  bit multipliers since 8 bits is approximately the same amount of information as 5 trits ( $2^8 \approx 3^5$ ). The QAT multiplier is supposed to work in a QAT environment (the developed logic allows us to implement any system); if it is used in a binary environment, the conversion from binary to ternary and vice versa should be considered.

Two of the considered multipliers when doing the comparison use the fully adiabatic logics SCRL and CRL. Their

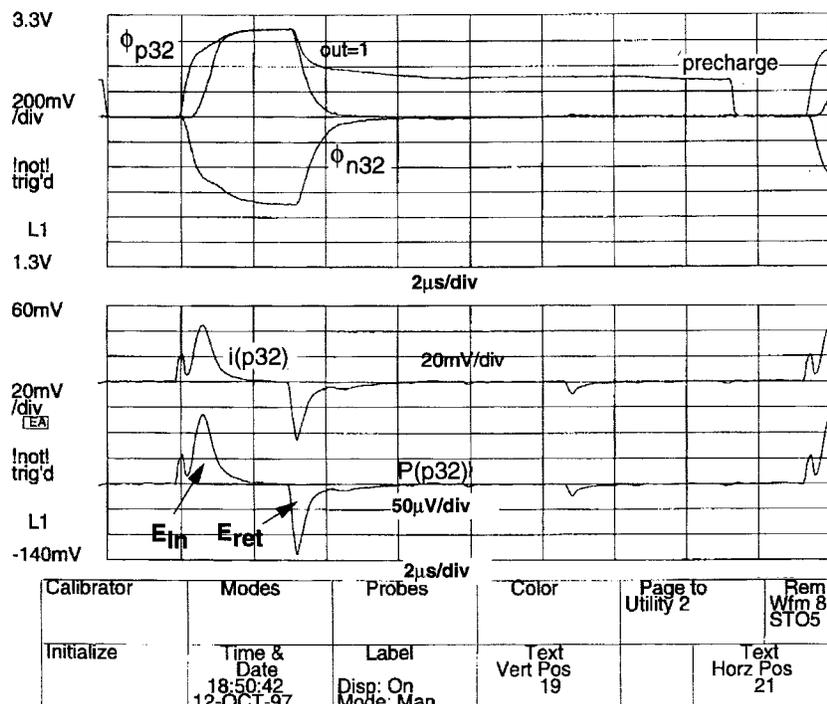


Fig. 9. Measurements of the multiplier.  $i_{p32}$  is the current from the clock  $\phi_{p32}$  and  $P_{p32}$  is the corresponding power.  $(E_{in} - E_{ret})$  is the energy dissipated in this switching.

TABLE I  
COMPARISON OF DIFFERENT MULTIPLIERS

	# devices	# clocks + power lines	Power	Delay	PDP	Suppl.
Fully Ad. SCRL 8x8 bits mult. [3]	~ 6300	48+3	2.5 $\mu$ W	1.6 $\mu$ s	4 pJ	5 V
Fully Ad. CRL 8x8 bits mult. [11]	~ 32000	4+3	20 $\mu$ W	0.4 $\mu$ s	8 pJ	5 V
Quasi-Ad. QSERL 8x8 bits mult. [13]	?	2+2	2.5 mW	200 ns	500 pJ	3.3 V
Static CMOS 8x8 bits mult.	2308	0+2	50 mW 18 mW	25 ns 38 ns	1250 pJ 700 pJ	5 V 3.3 V
QAT 5x5 trits mult.	3850	18+4	4 $\mu$ W	17 $\mu$ s	70 pJ	$V_{-1} = 1.8$ V $V_0 = 2.5$ V $V_1 = 3.2$ V

performances are obtained by theoretical analysis from the papers where these logics were presented, in [3] and [11], respectively, assuming the same *ecpd07* technology, and considering a charging/decharging period  $T = 100$  ns. The third one is a conventional CMOS multiplier implemented with CAD tools in the *ecpd07* technology, whose performances are obtained by simulating with Hspice the extracted netlist from the layout. The fourth one uses the quasi-adiabatic logic QSERL. Its consumption is obtained from the ratio between the consumption of the QSERL multiplier and a conventional one (ratio presented in [13] and obtained by simulation), using as data for the conventional multiplier the consumption shown in Table I. In the four multipliers,

the output capacitances considered are the same capacitances obtained when measuring the QAT multiplier (7.5 pF for each one).

The comparison is done as a function of the area, consumption, and delay of each multiplier. The delay is defined as the time needed to carry out one operation. The area is evaluated as a function of the number of devices and the number of power supplies. The parameter used to compare the global performance of the different multipliers is the *power-delay product* (PDP), or what is the same, the energy required to carry out one multiplication.

The power supplies used to make the measurements of the QAT multiplier use exponential waves instead of ideal

ramps because of simplicity in the implementation. The power dissipation shown in Table I is only the IC consumption, without the dissipation due to the clock generation since they have not been yet implemented with the ability to recover energy: taking into account the clock generation efficiency is under present investigation (previous works about power-clock generation show an efficiency from 80 to 90% [4], [13]).

From Table I, the following results can be summarized: the PDP of the QAT  $5 \times 5$  mult is worse than the PDP of the fully adiabatic binary  $8 \times 8$  multipliers due to the nonfully adiabatic switching and the breakage of reversibility, but it is still one order of magnitude better than the PDP of a conventional binary CMOS  $8 \times 8$  mult, and seven times better than the PDP of a multiplier implemented in other quasi-adiabatic logic. The area saving of the QAT multiplier in front of the smallest fully adiabatic but binary multiplier is 60% in the number of devices, as well as having an intrinsic benefit in routing because of having five trits in front of 8 bits.

## V. CONCLUSION

A new low-power logic has been presented, which uses quasi-adiabatic switching and partial energy recovery. A special feature of this logic is to be ternary, in order to diminish the area needed with respect to other adiabatic binary logics, keeping a satisfactory power saving. A  $5 \times 5$  trits multiplier has been implemented using this logic. Measurements show a power–delay product one order of magnitude better than a conventional  $8 \times 8$  bit CMOS multiplier and smaller than the PDP of a multiplier implemented in other quasi-adiabatic logic; and there is an important area saving with respect to fully adiabatic  $8 \times 8$  bit multipliers. As future work, power supplies with the capability of recovering energy must be included in the design.

## ACKNOWLEDGMENT

The authors thank the reviewers for their helpful comments.

## REFERENCES

- [1] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, pp. 473–484, Apr. 1992.
- [2] W. C. Athas, L. J. Svensson, J. G. Koller, N. Tzartzanis, and E. Y. Chou, "Low-power digital systems based on adiabatic-switching principles," *IEEE Trans. VLSI Syst.*, vol. 2, pp. 398–407, Dec. 1994.

- [3] S. G. Younis and T. F. Knight, "Asymptotically zero energy split-level charge recovery logic," in *Int. Workshop Low Power Design*, 1994, pp. 177–182.
- [4] A. G. Dickinson and J. S. Denker, "Adiabatic dynamic logic," *IEEE J. Solid-State Circuits*, vol. 30, pp. 311–315, Mar. 1995.
- [5] D. Mateo and A. Rubio, "quasi-adiabatic ternary CMOS logic," *Electron. Lett.*, vol. 32, pp. 99–101, 1996.
- [6] Y. Moon and D. K. Jeong, "An efficient charge recovery logic circuit," *IEEE J. Solid-State Circuits*, vol. 31, pp. 514–522, Apr. 1996.
- [7] S. L. Hurst, "Multiple-valued logic. Its status and its future," *IEEE Trans. Comput.*, vol. C-33, pp. 1160–1179, Dec. 1984.
- [8] M. Yoeli and G. Rosenfeld, "Logic design of ternary switching circuits," *IEEE Trans. Electron. Comput.*, vol. EC-14, pp. 19–29, Feb. 1965.
- [9] J. S. Wang, C. Y. Wu, and M. K. Tsai, "Low power dynamic ternary logic," *Proc. Inst. Elect. Eng.*, vol. 135, pp. 221–230, Dec. 1988.
- [10] T. Gabara and W. Fischer, "An integrated system consisting of an  $8 \times 8$  adiabatic-PPS multiplier powered by a tank circuit," in *Int. Solid-State-Circuits Conf.*, 1995, pp. 316–317.
- [11] S. G. Younis, "Asymptotically zero energy computing with split-level charge recovery logic," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, June 1994.
- [12] R. L. Geiger, P. E. Allen, and N. R. Strader, *Design Techniques for Analog and Digital Circuits*. New York: McGraw-Hill, 1990.
- [13] Y. Ye and K. Roy, "Reversible and quasistatic adiabatic logic," in *European Conf. Circuit Theory and Design*, 1997, pp. 912–917.



**Diego Mateo** received the M.S. and Ph.D. degrees from the Department of Electronic Engineering, Telecommunication Engineering Faculty of Barcelona, Spain, in 1993 and 1998, respectively.

He is currently an Assistant Professor at the Telecommunication Engineering Faculty, Universitat Politècnica de Catalunya, Barcelona. His research interests include VLSI design for testability and the design of low-power digital systems.



**Antonio Rubio** received the M.S. and Ph.D. degrees from the Industrial Engineering Faculty of Barcelona, Spain.

He has been an Associate Professor in the Department of Electronic Engineering, Industrial Engineering Faculty (UPC) and a Professor in the Department of Physics, Balearic Islands University. He is currently a Professor of Electronic Technology at the Telecommunication Engineering Faculty, Universitat Politècnica de Catalunya, Barcelona. His research interests include VLSI design and test, computer-aided design, automatic test pattern generation, device and circuit modeling, and high-speed circuit design.