Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Design of adiabatic two's complement multiplier-accumulator based on CTGAL*

Peng-jun WANG†1, Jian XU1, Shi-yan YING2

([1]*Institute of Circuits and Systems, Ningbo University, Ningbo 315211, China*)

([2]*College of Information Engineering, Zhejiang University of Technology, Hangzhou 310014, China*)

†E-mail: wangpengjun@nbu.edu.cn

Received Jan. 5, 2008; Revision accepted Mar. 10, 2008; Crosschecked Dec. 25, 2008

Abstract: We propose a new design scheme for a Booth encoder based on clocked transmission gate adiabatic logic (CTGAL). In the new design the structural complexity of the Booth encoder is reduced while the speed of the multiplier is improved. The adiabatic two's complement multiplier-accumulator (MAC) is furthermore a design based on the CTGAL. The computer simulation results indicate that the designed circuit has the correct logic function and remarkably less energy consumption compared to that of the MAC based on complementary metal oxide semiconductor (CMOS) logic.

Key words: CTGAL circuit, Adiabatic circuit, Booth arithmetic, Multiplier, Two's complement MAC
doi:10.1631/jzus.A0820013 **Document code:** A **CLC number:** TN79

INTRODUCTION

In a digital system, the multiplier-accumulator (MAC) composed of adders and multipliers plays an important role in many digital signal processing systems and is also the primary contributor to power dissipation (Suvakovic and Salama, 2003). Studying the low-power MAC will bring some progress in the design of a low-power integrated circuit. As traditional complementary metal oxide semiconductor (CMOS) circuits adopt direct current (DC) power supply, the energy is usually transformed from electric energy to heat in an irreversible manner during charging or discharging of the node capacitances. Instead of using a DC power supply, adiabatic CMOS circuits are driven by alternating current (AC) power supply (Pedram and Wu, 1999; Wu *et al*., 2000; Blotti and Saletti, 2004; Li *et al*., 2007). In this way, the resonant tank is formed by the inductor in the power clock generator and the node capacitances in the adiabatic circuit to facilitate energy transfer between magnetic energy and electric energy. So the charge on the node capacitances can be recycled to achieve power recovery, and the irreversible energy conversion from electric energy to heat caused by the dissipative element, i.e., resistances, can be largely reduced or avoided (Arsalan and Shams, 2005; Park *et al*., 2005).

Thus, according to the principle of adiabatic computing (Wu and Hang, 2000; Fang *et al*., 2003), a new design scheme for the Booth coder is proposed based on the study of clocked transmission gate adiabatic logic (CTGAL) and Booth arithmetic. An adiabatic two's complement MAC is furthermore designed using the CTGAL. The computer simulation result of a 4×4 adiabatic two's complement MAC verifies that the designed circuits have the correct logic function and the notable characteristic of low-power consumption.

* Project supported by the National Natural Science Foundation of China (No. 60776022), the Science and Technology Fund of Zhejiang Province (No. 2008C21166), the Key Scientific Research Fund of the Department of Education of Zhejiang Province (No. 20061666), the Professor Fund (No. JSL2007001), the Scientific Research Fund (No. XK0610030) and the K. C. Wong Magna Fund in Ningbo University, China

DESIGN OF CTGAL GATES

CTGAL circuit

A CTGAL circuit adopts two-phase power clocks (Wang and Yu, 2006; 2007), as shown in Figs.1a and 1b, where Φ is the power clock and $\bar{\Phi}$ is the clock-controlled clock. $\bar{\Phi}$ and Φ are a pair of complement clocks.

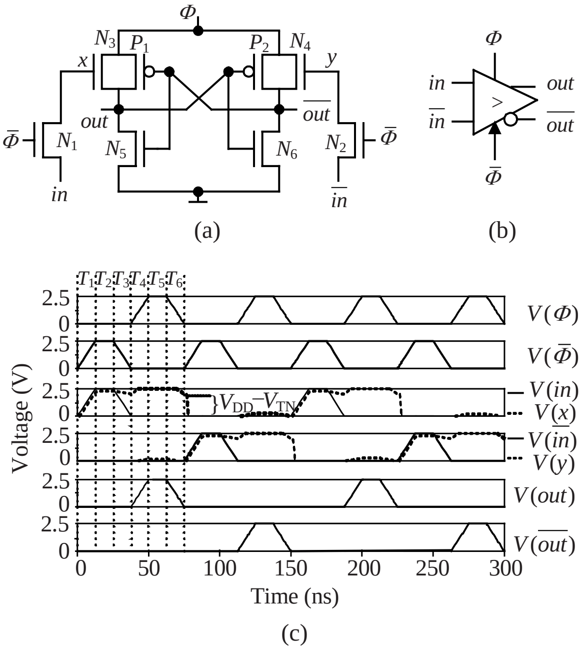


Fig.1 CTGAL circuit
(a) Scheme; (b) Symbol; (c) Simulation wave

One power clock is divided into six equal parts T_1~T_6, as shown by the simulation waveforms in Fig.1c. The operation procedure of a CTGAL circuit is divided into two steps: (1) The sampling phase T_1~T_3. The input signals *in* and $\overline{in}$ have the same phase as the clock-controlled clock $\bar{\Phi}$. Two clocked NMOS (N-channel metal oxide semiconductor) transistors N_1 and N_2, which are controlled by $\bar{\Phi}$, sample the input signals. (2) The valuing, holding and recovery phase T_4~T_6. During the period T_4, when the input signals and the clock-controlled clock are all at low voltage, there must be one node floating on high-voltage $V_{DD}-V_{TN}$ (V_{TN} is the threshold voltage of the NMOS transistor) between floating nodes *x* and *y*, for example *x*. When Φ rises gradually, the output node *out* is valued by the power clock. As N_3 is turned on at the beginning of the evaluation, the *out* is charged by the Φ completely. At the same time, due to the parasitic capacitance between this floating high-voltage node and Φ, the voltage of the floating high-voltage node *x* can be bootstrapped to a higher level than $V_{DD}-V_{TN}$. Period T_5 is the holding phase. During the period T_6, Φ decreases gradually, and the voltage of the floating high-voltage node *x* finally comes back to $V_{DD}-V_{TN}$. When P_1 and P_2 are turned off, N_3 or N_4 is still turned on to ensure that the charge on *out* or $\overline{out}$ can be completely recovered by Φ. So the circuit has no non-adiabatic dissipation caused by threshold value loss at the beginning of the evaluation and the end of the recovery.

CTGAL gates

In order to implement the logic gates (such as the AND gate), firstly the clocked NMOS transistors (such as N_1 and N_2 in Fig.1a) are used to sample each input signal. Then the complementary NMOS logic blocks are used to replace the transistors N_3 and N_4 in the CTGAL circuit. In this way the diverse gates can be gained, such as two-input AND gate and the 2-to-1 data selector shown in Fig.2.

ADIABATIC TWO'S COMPLEMENT MAC

Adiabatic two's complement MAC, as shown in Fig.3, consists of a multiplier and an accumulator. The multiplier is composed of a partial-product generator, a partial-product compressor and an adder (Zheng *et al.*, 2004). The accumulator is an adder with feedback signals, which accumulates the results of the multiplication. In order to improve the speed of the multiplier, a new Booth encoder is proposed to generate firstly the partial-products, which can effectively reduce the number of the partial-products. Then the 4-2 compressor and Wallace tree are used to compress the partial-products, so that the number of additions can be reduced. Lastly, the Ladner-Fischer parallel prefix adder, which has less delay compared to the other parallel prefix adder, is used to obtain the sum of the multiplication results.

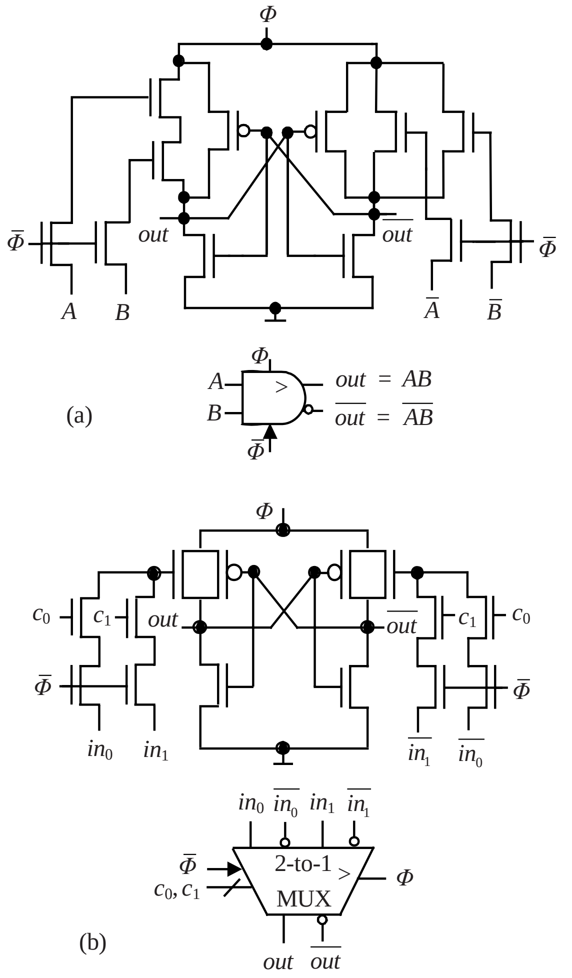


Fig.2 CTGAL gates
(a) AND gate; (b) 2-to-1 selector (XOR gate)

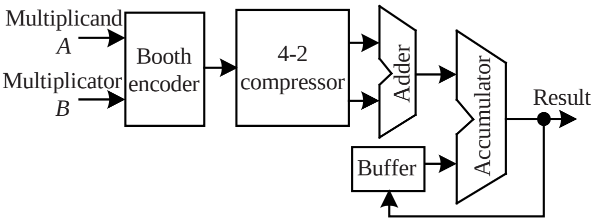


Fig.3 Block diagram of the adiabatic MAC

New Booth encoder

In the conventional multiplication operation, only one bit of the multiplicator is checked to generate the partial-products. The conventional partial-product generator has a short delay time but many partial-products. For the n-bit multiplicator there will be n partial-products in the conventional multiplication operation. As the k-bit of the multiplicator can be checked in Booth arithmetic, only n/k partial-products are generated. So the number of the partial-products is reduced. The improved Booth arithmetic (k=2) can implement the two's complement multiplication. The multiplicand and multiplicator are both in the form of two's complement. And the addition and subtraction are used to obtain the partial-products. The process of generating the partial-products is shown as follows:

Step 1: Initialize the product to '0'; the multiplicand A and multiplicator B are ready.

Step 2: Add '0' at the right of the least significant bit (LSB) of the multiplicator, so the length of the multiplicator is (n+1)-bit.

Step 3: Three-bit data ($B_{i+1}B_iB_{i-1}$) of the multiplicator are checked and one bit is overlapped every time from the LSB to the most significant bit (MSB).

Step 4: According to the different compounding (000~111) of the three-bit data ($B_{i+1}B_iB_{i-1}$), there are five operations for the partial-products: "+0", "+A", "−A", "+2A", "−2A" (+0 denotes no operation, and the partial-products remain unchanged; "+A" denotes adding the multiplicand A to the partial-products; "−A" denotes subtracting the multiplicand A from the partial-products; the rest may be deduced by analogy), as shown in Table 1. In particular, the operations "−A" and "−2A" are accomplished through adding the two's complement of A and 2A, respectively.

Step 5: The multiplicand A is shifted right two bits.

Step 6: Repeat Steps 3~5 n/2 times to obtain the partial-products.

Table 1 Booth arithmetic

Input signals			Operations	Select signals
B_{i+1}	B_i	B_{i-1}		
0	0	0	+0	se_0
0	0	1	+A	se_1
0	1	0	+A	se_1
0	1	1	+2A	se_2
1	0	0	−2A	se_2
1	0	1	−A	se_1
1	1	0	−A	se_1
1	1	1	+0	se_0

The five operations in Step 4 are classified as three kinds: "+0", "±A", "±2A". Three select signals are needed: se_0 selects operation "+0", se_1 selects operation "±A", and se_2 selects operation "±2A". The select signals are shown as the following expressions Eqs.(1a)~(3b) by using the Karnaugh diagram:

$$se_0 = B_{i+1}B_iB_{i-1} + \overline{B}_{i+1}\overline{B}_i\overline{B}_{i-1}, \tag{1a}$$

$$\overline{se_0} = \overline{B}_{i+1}B_{i-1} + B_{i+1}\overline{B}_i + B_i\overline{B}_{i-1}, \tag{1b}$$

$$se_1 = B_i \oplus B_{i-1}, \tag{2a}$$

$$\overline{se_1} = B_i \,\overline{\oplus}\, B_{i-1}, \tag{2b}$$

$$se_2 = \overline{B}_{i+1}B_iB_{i-1} + B_{i+1}\overline{B}_i\overline{B}_{i-1}, \tag{3a}$$

$$\overline{se_2} = \overline{B}_{i+1}\overline{B}_i + B_{i+1}B_{i-1} + B_i\overline{B}_{i-1}. \tag{3b}$$

From Table 1, one select signal se_1 (or se_2) corresponds to the addition or subtraction operation at one time. If the signal se_1 (or se_2) is valid, which operation should be done poses a problem. According to the result of exclusive-or operation between B_{i+1} and the multiplicand, the appropriate operation can be selected.

When B_{i+1} equals '0', the addition is selected; the exclusive-or operation result of the multiplicand and B_{i+1} is the same with the multiplicand. When B_{i+1} equals '1', the subtraction is selected; the exclusive-or operation result of the multiplicand and B_{i+1} is the multiplicand's reverse code. According to this, the circuit structure of Booth encoding, as shown in Fig.4, can be obtained. The operation of adding '1' in subtraction will be finished in the following steps: partial-product compression and partial-product addition. Except for the generation circuit of the select signals, only one 3-to-1 selector and one exclusive-or gate are needed for each bit of the multiplicand. So the delay of the Booth encode equals two-stage gate delay, which is just one gate delay more than that of the AND array. But the number of partial-products generated by the Booth encode is just half that of the AND array.

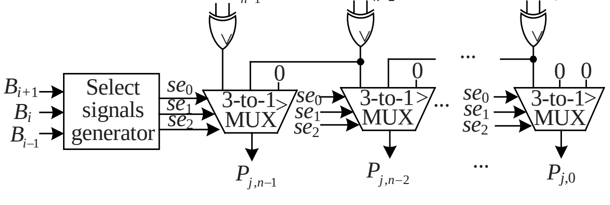


Fig.4 New Booth encoder

Partial-product compressor

In order to improve the addition speed of the partial-products, a 4-2 compressor is adopted to compress the partial-products (Prasad and Parhi, 2001). The improved 4-2 compressor based on CTGAL, as shown in Fig.5, has five input data (x_1, x_2, x_3, x_4, and C_{in}) and three output data (S, C, and C_{out}). The operation of the adiabatic 4-2 compressor includes three steps, and two-phase power clocks are adopted. The output signal C_{out} and the input signal C_{in} have the same phase.

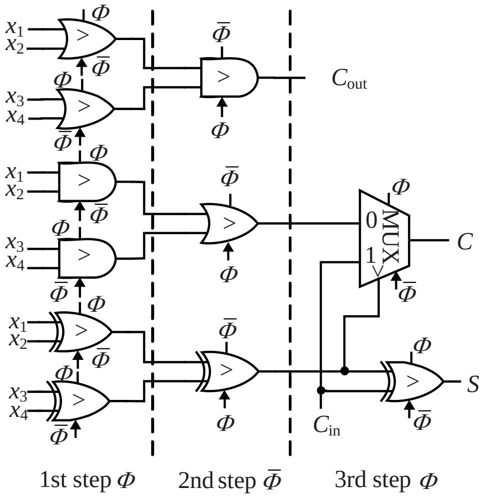


Fig.5 Improved 4-2 compressor

The 4-2 compressor shown in Fig.5 is used to obtain the sum of four 1-bit data. So for four n-bit data, n 4-2 compressors are used. The output signal C_{out} of the current 4-2 compressor is directly connected to the input signal C_{in} of the near high-bit 4-2 compressor. In the addition operation of four n-bit data, all the 4-2 compressors can start to work at the same time.

If there are more than four partial-products, a Wallace tree structure is introduced to improve the compression speed. A Wallace tree structure for eight partial-products is shown in Fig.6. Each four partial-products are compressed using a set of 4-2 compressor. The output signals of two sets of 4-2 compressors, which are independent, are the input signals of the next 4-2 compressor. It just costs two 4-2 compressor delay to compress the eight partial-products to two data. The addition speed of the partial-products is improved effectively by using the 4-2 compressors and the Wallace tree structure.

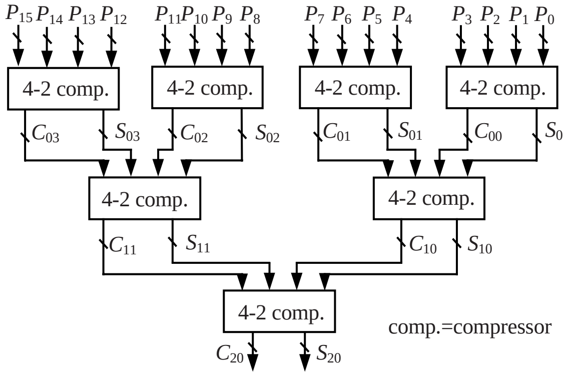

P_{15} P_{14} P_{13} P_{12}
P_{11} P_{10} P_9 P_8
P_7 P_6 P_5 P_4
P_3 P_2 P_1 P_0
4-2 comp.
C_{03} S_{03} C_{02} S_{02} C_{01} S_{01} C_{00} S_{00}
C_{11} S_{11} C_{10} S_{10}
C_{20} S_{20}
comp.=compressor

Fig.6 Structure of the Wallace tree

But the outputs of the last 4-2 compressor are not the final product. For $n\times n$ bit multiplication, two $2n$-bit data, S_i and C_i (i=0, 1, …, $2n-1$), are generated by using the Booth encoder and the partial-product compressor. The sum of the two $2n$-bit data is just the final product.

Addition

A Ladner-Fischer parallel prefix adder needs to obtain the sum of the two $2n$-bit data, which are generated by the partial-product compressor (Choi and Swartzlander, 2005). The Ladner-Fischer parallel prefix adder is composed of some operator "*O*". The process of the operator "*O*" and an 8-bit adiabatic Ladner-Fischer parallel prefix adder are shown in Figs.7a and 7b, respectively. Some CTGAL buffers are needed to modify the phase in the adiabatic Ladner-Fischer parallel prefix adder.

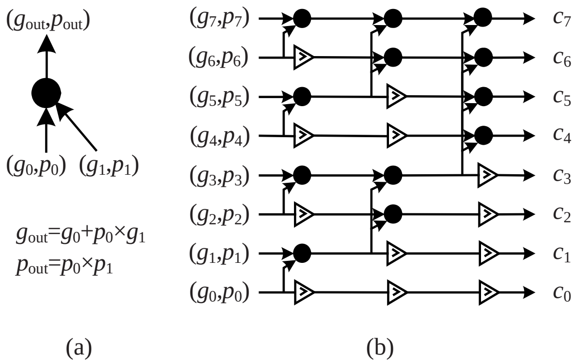

(g_{out},p_{out})
(g_0,p_0) (g_1,p_1)
$g_{out}=g_0+p_0\times g_1$
$p_{out}=p_0\times p_1$
(g_7,p_7) (g_6,p_6) (g_5,p_5) (g_4,p_4) (g_3,p_3) (g_2,p_2) (g_1,p_1) (g_0,p_0)
c_7 c_6 c_5 c_4 c_3 c_2 c_1 c_0
(a)
(b)

Fig.7 Ladner-Fischer adder

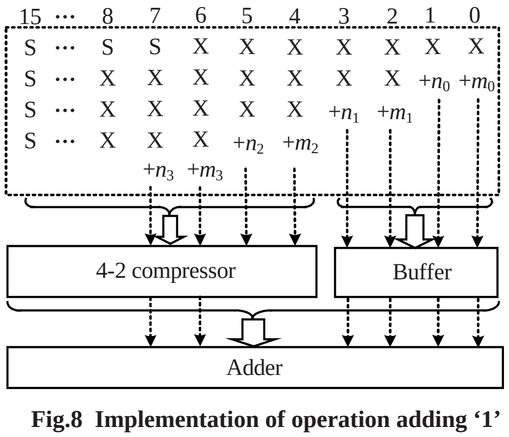
(a) Operator "*O*"; (b) "*O*" array of the Ladner-Fischer adder

The generated signal g_i and propagated signal p_i (i=0, 1, …, $2n-1$) can be obtained from the following equations: $g_i=S_i\cdot C_i$, $p_i=S_i\oplus C_i$. The carry signals cc_0, cc_1, …, cc_{n-1} are obtained from the array of operator "*O*" shown in Fig.7b. The sum $mac_{2n-1}\ldots mac_0$ is achieved by using the p_i and cc_i: $mac_0=p_0$, $mac_i=p_i\oplus cc_{i-1}$ (i=1, 2, …, $2n-1$).

For the accumulator the only difference from the ordinary adder is that the output signals of the accumulator are feedback to the input ports of the accumulator. These signals are also called the feedback signals. The output signals and the input signals should have the same phase for the correct logic function. Thus some CTGAL buffers are needed to modify the phase. In order to obtain the correct accumulating order, the input signal of the MAC should repeat several clock periods consecutively to have the same delay with the accumulator. Otherwise the new product may not have been added up to the latest accumulated value.

Amendment operation

When the operation "$-A$" or "$-2A$" is selected in Booth arithmetic, the two's complement of the multiplicand is needed. Since the Booth encoder only generates and shifts the reverse code, the corresponding partial-product needs amending. '1' is needed to add up to the LSB of the partial-products for amending. From Table 1 we can know that, when $B_{i+1}B_iB_{i-1}$ equals "101" or "110", operation "$-A$" is selected. So '1' may be added to columns 0, 2, 4, 6 of the partial-product array shown in Fig.8. When $B_{i+1}B_iB_{i-1}$ equals "100", operation "$-2A$" is selected. As the shift operation has been finished by the Booth encoder, '1' may be added to columns 1, 3, 5 and 7.



15	···	8	7	6	5	4	3	2	1	0
S	···	S	S	X	X	X	X	X	X	X
S	···	X	X	X	X	X	X	X	$+n_0$	$+m_0$
S	···	X	X	X	X	X	$+n_1$	$+m_1$		
S	···	X	X	X	$+n_2$	$+m_2$				
			$+n_3$	$+m_3$						

Fig.8 Implementation of operation adding '1'

X denotes the partial-product; S denotes the signal bit

Considering the two cases, we can produce amending numbers as follows: $m_i=B_{i+1}(B_i\oplus B_{i-1})$, $n_i = B_{i+1}\overline{B}_i\overline{B}_{i-1}$ (i=0, 2, 4, 6). The m_i is added to columns 0, 2, 4, 6 respectively, while the n_i is added to columns 1, 3, 5, 7 respectively. The partial-product, which needs to be amended, is corresponding to only one operation, "−*A*" or "−2*A*". Thus at most one of the two amending numbers m_i and n_i is equal to '1'. When the two amending numbers are both equal to '0', the operation may be "+*A*", "+2*A*" or "+0". So the corresponding partial-product does not need amending.

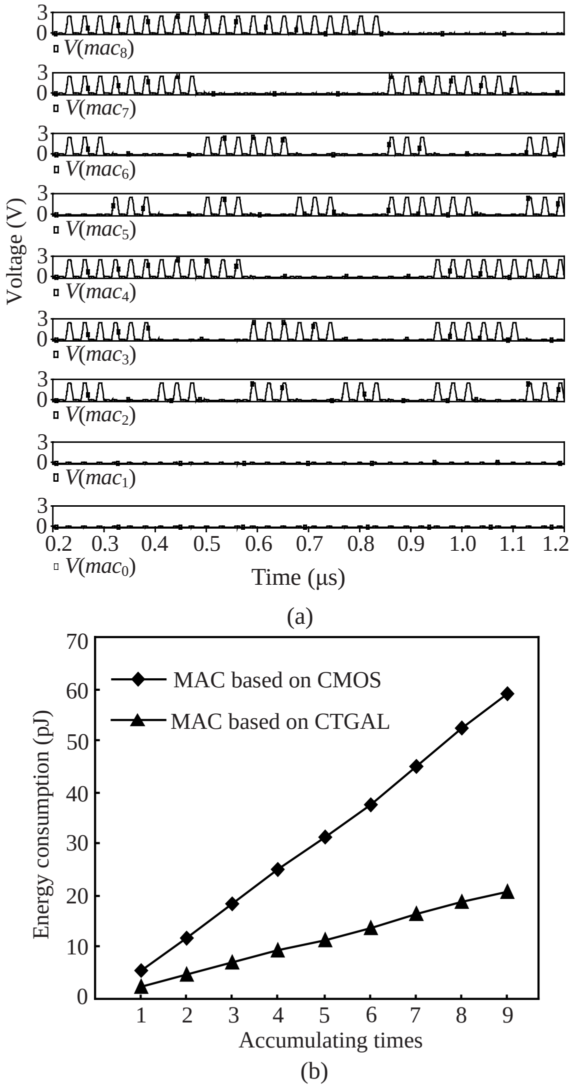
From columns 0 to 3, there are only one or two items of data in each column. So the 4-2 compressor is not needed in the four columns. The data in each column are directly propagated to the adder of the multiplier. The amendment operations in these columns are also implemented in the adder. From the 4th column to the most significant column, 4-2 compressors are needed in each column. The 4th and 5th columns both have three items of data, but the 4-2 compressor can compress four items of data. So the amending numbers can be seen as one of the input data of the 4-2 compressor in these two columns. The 6th and 7th columns both have four items of data, thus the amendment operations in these two columns are also implemented in the adder.

For the amendment operations in the adder, columns 0 and 1 both have one item of data. The amending numbers can be seen as one of the input data of the adder in these two columns. But columns 2, 3, 6, 7 all have two data items in each column and no input channel can be used in the adder. The generated signal g_i and the propagated signal p_i need redesigning. Let q denote the amending number, and the new g_i, p_i can be obtained by using the Karnaugh diagram: $g_i=S_iC_i+qC_i+qS_i$, $p_i = S_i \overline{\oplus} C_i \overline{\oplus} q$ (i=2, 3, 6, 7; q=m_i, n_i).

From the above analysis, we can see that the Booth encoder does not individually finish the subtraction operation, and that the structural complexity of the Booth encoder is reduced. But the partial-products need amending when doing subtraction in the Booth encoder. In order to amend the partial-products generated by the Booth encoder, we use the empty input channels of the 4-2 compressor and the adder. If there are no empty input channels, some of the generated signal g_i and the propagated signal p_i in the adder are redesigned to amend the partial-products. Thereby, the adiabatic two's complement MAC based on CTGAL is achieved.

SIMULATION AND ANALYSIS

Using the parameters of a TSMC 0.25-μm CMOS device, a 4×4 adiabatic two's complement MAC with 16-bit accumulator is simulated as an example by the computer. Let a negative number *A*=1010 (−6 in decimal) and a positive number *B*=0110 (6 in decimal) with the accumulation sum mac_{15}~mac_0. The function simulation result is shown in Fig.9a. The mac_{15}~mac_9 are not shown because they are at high voltage in 1.2 μs. The product is in the format of two's complement. As the delay of the 16-bit accumulator is three clock periods, the input signals repeat three clock periods. The simulation result indicates that the designed circuit has the correct logic function.



(a)

(b)

Fig.9 MAC simulation result
(a) Functional simulation wave; (b) Energy consumption

The low-power characteristic of the designed MAC was compared to that based on CMOS logic. Fig.9b shows the curves of transient energy consumption of the two circuits. When the accumulating times are more than eight, the MAC based on CTGAL attains an energy saving of about 65% compared with the CMOS logic MAC. The more the accumulating times, the greater the energy saving.

CONCLUSION

The CTGAL circuit has no threshold value loss when charging and discharging the node capacitance of the circuit by using the bootstrap effect of NMOS transistors. So the power dissipation can be reduced significantly. The adiabatic MAC based on CTGAL has the obvious characteristic of energy recovery. The new Booth encoder of the MAC has a simple structure and short delay time. Some empty channels in the MAC are adequately utilized to finish the Booth encoder operation. The proposed adiabatic MAC based on CTGAL will influence the future study of low-power operation circuits and low-power digital processors.

References

Arsalan, M., Shams, M., 2005. Charge-recovery Power Clock Generators for Adiabatic Logic Circuits. Proc. 18th Int. Conf. on VLSI Design held jointly with 4th Int. Conf. on Embedded Systems Design, p.171-174. [doi:10.1109/ICVD.2005.64]

Blotti, A., Saletti, R., 2004. Ultralow-power adiabatic circuit semi-custom design. *IEEE Tran. VLSI Syst.*, **12**(11): 1248-1253. [doi:10.1109/TVLSI.2004.836320]

Choi, Y., Swartzlander, E.E.Jr., 2005. Parallel Prefix Adder Design with Matrix Representation. Proc. 17th IEEE Symp. on Computer Arithmetic, Cape Cod, p.90-98. [doi:10.1109/ARITH.2005.35]

Fang, Z.X., Wang, P.J., Liu, Y., 2003. Universal circuits theory for binary, multiple-value and adiabatic circuits. *Acta Electron. Sin.*, **31**(2):303-305.

Li, S., Zhou, F., Chen, C.H., Chen, H., Wu, Y.P., 2007. Quasi-static Energy Recovery Logic with Single Power-Clock Supply. IEEE Int. Symp. on Circuits and Systems, New Orleans, LA, p.2124-2127. [doi:10.1109/ISCAS.2007.378592]

Park, J., Hong, S.J., Kim, J., 2005. Energy-saving Design Technique Achieved by Latched Pass-transistor Adiabatic Logic. Proc. IEEE Int. Symp. on Circuits and System, Kobe, Japan, p.4693-4696. [doi:10.1109/ISCAS.2005.1465680]

Pedram, M., Wu, Q., 1999. Design Considerations for Battery-Powered Electronics. Proc. 36th Design Automation Conf., p.861-866. [doi:10.1109/DAC.1999.782166]

Prasad, K., Parhi, K.K., 2001. Low-power 4-2 and 5-2 Compressors. IEEE Conf. Record of the 35th Asilomar Conf. on Signals, Systems, and Computers, Lexington, MA, USA, p.129-133. [doi:10.1109/ACSSC.2001.986892]

Suvakovic, D., Salama, C.A.T., 2003. Energy efficient adiabatic multiplier-accumulator design. *J. VLSI Signal Processing Syst.*, **33**(1-2):83-103. [doi:10.1023/A:1021145919099]

Wang, P.J., Yu, J.J., 2006. Design of clocked transmission gate adiabatic logic circuit and SRAM. 8th Int. Conf. on Solid-State and Integrated Circuit Technology, Shanghai, p.2118-2120. [doi:10.1109/ICSICT.2006.306634]

Wang, P.J., Yu, J.J., 2007. Design of two-phase sinusoidal power clock and clocked transmission gate adiabatic logic circuit. *J. Electron.*, **24**(2):225-231. [doi:10.1007/s11767-005-0170-2]

Wu, Q., Pedram, M., Wu, X.W., 2000. Clock-gating and its application to low power design of sequential circuits. *IEEE Trans. Circuits Syst. I: Fundam. Theory Appl.*, **47**(3):415-420. [doi:10.1109/81.841927]

Wu, X.W., Hang, G.Q., 2000. Principle of adiabatic computing and CMOS circuits with energy recovery. *Chin. J. Comput.*, **23**(7):779-785 (in Chinese).

Zheng, W., Yao, Q.D., Zheng, M., Li, X.D., 2004. Design of low-power and high-speed multiplier. *J. Zhejiang Univ. (Eng. Sci.)*, **38**(5):534-538 (in Chinese).