第 24 卷第 11 期 半 导 体 学 报 Vol. 24, No. 11
2003 年 11 月 CHINESE JOURNAL OF SEMICONDUCTORS Nov., 2003

Static CMOS Implementation of Logarithmic Skip Adder*

Jia Song, Liu Fei, Liu Ling, Chen Zhongjian and Ji Lijiu

(*Institute of Microelectronics, Peking University, Beijing* 100871, *China*)

Abstract: Circuit design of 32-bit logarithmic skip adder (LSA) is introduced to implement high performance, low power addition. ELM carry lookahead adder is included into groups of carry skip adder and the hybrid structure costs 30% less hardware than ELM. At circuit level, a carry-incorporating structure to include the primary carry input in carry chain and an "and-xor" structure to implement final sum logic in 32-bit LSA are designed for better optimization. For 5V, 1μm process, 32-bit LSA has a critical delay of 5.9ns and costs an area of 0.62mm^2, power consumption of 23mW at 100MHz. For 2.5V, 0.25μm process, critical delay of 0.8ns, power dissipation of 5.2mW at 100MHz is simulated.

Key words: logarithmic skip; carry incorporating; sum logic; circuit design

EEACC: 2570D; 1265B; 5120

CLC number: TN432 **Document code**: A **Article ID**: 0253-4177(2003)11-1159-07

1 Introduction

Addition is the basic operation of ALU (arithmetic logic unit), MAC and address generation components affecting total performance of microprocessors. Portable application challenges designers with compact, low power and high-speed designs. Dynamic implementation[1] is the choice for high performance logic. Unfortunately, its complicated clock distribution scheme and signal transition due to precharging cause high power consumption[2]. Some high performance dynamic adders[3,4] are reported but the power dissipation is high[4]. Static design can offer the advantages of low cost and good performance if proper circuit techniques are adopted.

The most cost-efficient parallel adder is carry skip adder (CSA) adding only carry bypass logic on ripple carry adder. The fastest are logarithmic-time adders such as carry lookahead adder (CLA)[5] which spreads carry chain to achieve $O(\log_2 n)$ delay for n-bit wide addition. But in CSA, carries ripple in groups and performance is not satisfied. Logarithmic time adders such as conditional sum, parallel-prefix CLA are fast and regularly structured. But conditional sum adder needs two sets of serial adder to be selected conditionally by incoming carry and thus is not cost-efficient. Parallel-prefix adders such as Kogge-Stone and Lander-Fischer need area and power overhead for the spread carry chain. Also they suffer from exponentially growing interconnection complexity[6] and fanout load. ELM, a tree structured CLA, is found to have the best power-delay product among logarithmic-time adders[7]. Here we show that performance and power dissipation of 32-bit adder can be further compromised by a hybrid structure of CSA

* Project supported by Advanced Research (Nos. 41308010402 and 415011005)

Jia Song male, was born in 1970, PhD candidate. His research interests include digital & mixed-signal design and computer arithmetic.

Liu Fei male, was born in 1974, PhD candidate. His research interest includes mixed-signal circuits and system design.

Ji Lijiu male, professor, advisor of PhD candidates. His research interests focus on digital & mixed-signal CMOS design.

Received 2 January 2003, revised manuscript received 16 June 2003

and ELM, called logarithmic skip adder (LSA). Static CMOS circuit design of LSA is elaborated on in the paper. The design is targeted to low power, compact 32-bit adder without sacrificing performance.

2 LSA structure

Carry chain in adder is defined[8] as

$$(g_{i+1,i}, p_{i+1,i}) = (g_{i+1}, p_{i+1}) \mathrm{op} (g_i, p_i) = (g_{i+1} + p_{i+1} g_i, p_{i+1} p_i) \quad (1)$$

where $g_i = a_i b_i$, $p_i = a_i \oplus b_i$ are carry generation and carry propagation signals in ith position. The "op" operator calculates carry from adjacent 2 bits or 2 groups. The carry propagation signal of a group of bits can be calculated as:

$$p_{i,j} = p_i p_{i-1} \cdots p_j \quad (2)$$

Figure 1 is the schemcatic diagram of a 32-bit LSA architecture. It is a hybrid structure of CSA and CLA. The binary tree-structured carry chain of ELM adder is introduced to replace the serial chain in the groups of CSA. As a result, carries can skip longer distance and great speed advantage is achieved over the conventional CSA. While compared to ELM, carry chain in LSA does not fully spread to 32-bit binary tree, thus area and power cost are kept low. The structure is called logarithmic skip adder because the skip distance is $O(\log_2 n)$ to the bits in groups.

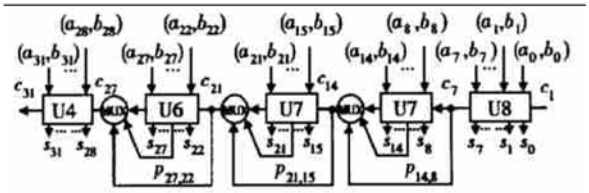

Fig. 1 32-bit logarithmic skip adder architecture

In Fig. 1, U8, U7, U6, and U4 are group adders in ELM structure. Through multiplexers, the "MUX" in Fig. 1, group carry propagation signals select the right carry signals in the same way as in CSA. In Fig. 2 and Fig. 3 shown are the U4 and U8 group in Fig. 1. The groups are 4 and 8-bit ELM adders as the other 7, 6-bit groups in Fig. 1. The logic in units "S", "E", "P", "G", "T" in the figures constructs binary tree and unit "S" calculates sum logic.

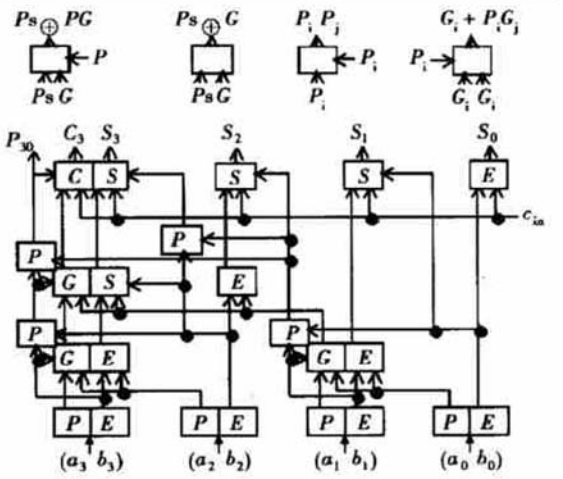

Fig. 2 U4 group

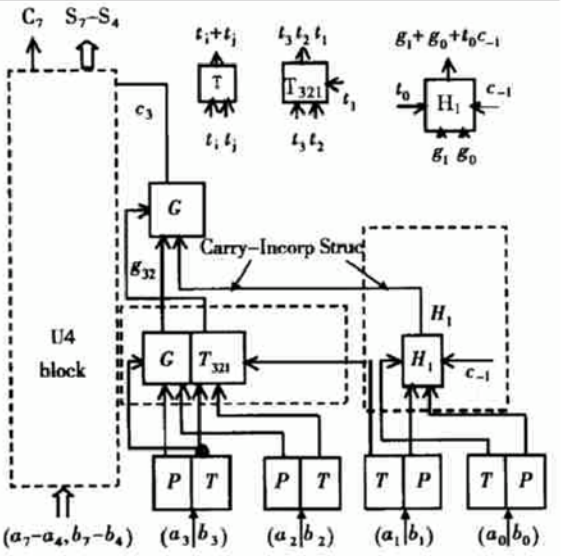

Fig. 3 U8 group

More significant 4 sums[9] of U8 group can be derived:

$$S_i = p_i \oplus c_{i-1} = p_i \oplus (g_{i-1} + p_{i-1} c_{i-2}) = p_i \oplus g_{i-1} \oplus p_{i-1} c_{i-2} = \mathrm{ps}_i \oplus p_{i-1,4} c_3 \quad (3)$$

where $i \in [4, 7]$, $p_{i-1,4} = p_{i-1} p_{i-2} \cdots p_4$ and ps_i is the partial sum in ith bit. For 8-bit ELM structure, the more significant 4 partial sums, the ps_i in Eq. (3) are local and can be processed in parallel with the logic of less significant 4 bits. They are transformed to final sums by one level of "and-xor" log-

ic, namely the unit "S" in Fig. 2.

The carry logic of group U7 can be derived as follows:

$$(g_{14,8}, p_{14,8}) = \{[(g_{14}, p_{14})\,\mathrm{op}(g_{13}, p_{13})] \times \mathrm{op}[(g_{12}, p_{12})\,\mathrm{op}(g_{11}, p_{11})]\} \times \mathrm{op}\{[(g_{10}, p_{10})\,\mathrm{op}(g_9, p_9)]\,\mathrm{op}[(g_8, p_8)\,\mathrm{op}(c_7)]\} \quad (4)$$

$$c_{14} = p_{14,8}(g_{14,8} + p_{14,8}c_7) + p_{14,8}c_7 \quad (5)$$

where the "op" operator is defined in Eq. (1) and $p_{14,8}$ is the group carry propagation. The logic depth from (a_i, b_i) $(i \in [8, 14])$ to the group carry $(g_{14,8}, p_{14,8})$ is four as shown in Eq. (4). Another level of logic is added by the multiplexer shown in Eq. (5) to get c_{14}.

Based on the unit gate delay model, the worst-case delay of 32-bit LSA can be obtained. The critical path delay is 8 logic gates. While the 32-bit CSA have a logic depth of 12[5] and the critical path of 32-bit ELM adder is 7 logic gates delay[7]. Meanwhile it is worth notifying that in Fig. 1 the 32-bit LSA, which has no more than 8-bit ELM adders as groups to be skipped, keeps fanouts within 5. While 32-bit ELM adder has one fanout load of 17 and two other fanouts of 9. The large fanout in logic design means multi-level buffers to be inserted in critical path to improve driving capacity. Considering logic depth and fanout together, we can claim that 32-bit LSA boasts the same performance as ELM adder.

We treat units "E", "S", "P", "G", "T" in Figs. 2 and 3 as cost unit (CU), so the power dissipation and area of 32-bit adders may be estimated. The n-bit binary tree structure in ELM adder totals $i = \log_2 n + 1$ levels of leaves. The first level calculates p, g signals and last level, the ith one, calculates final sums. The number of cost units for 32-bit ELM adder is $32\times2 + 16\times3 + 8\times5 + 4\times9 + 2\times17 + 1\times33 = 255$. As 32-bit LSA takes 8, 7, 6, 4-bit ELM as building blocks, we can reasonably treat them as four 8-bit ELM blocks for simplicity and it costs $4\times(8\times2 + 4\times3 + 2\times5 + 1\times5) + 3 = 175$ (CU) to construct a 32-bit LSA. Thus the cost improvement of 32-bit LSA over ELM is 30%.

Based on the above analysis we conclude that LSA is suitable for low power and high performance addition.

3 Circuit design of LSA

Circuit scheme at transistor level is designed for better performance optimization. In general all NMOS devices here are sized equivalently minimum width, namely n-serial transistors are sized $n\lambda$ where λ is minimum gate length. PMOS devices are twice sized to compensate for hole's reduced mobility. Some transistors are oversized to balance critical paths. Also, buffers are inserted to cut down glitches although not able to eliminate all. These measures are aimed to reduce energy by decreasing effective capacitances and switch activities.

Two parts of 32-bit adder will be explained in detail: the carry-incorporating structure and the sum logic. Circuits are simulated using two technology: Wuxi Huajing 5V, 1.0μm CMOS n-well, single poly, dual metal and Dongbu 2.5V, 0.25μm CMOS process, with the objective to observe circuit performance during process immigration.

3.1 Carry incorporating structure

Primary carry input to the least significant bit of adder often needs an additional gate delay to be included in carry chain. The process is shown as:

$$c_3 = g_{32} + t_{32}(g_1 + t_1(g_0 + t_0c_{-1})) \quad (6)$$

where $t_i = a_i + b_i$, $t_{32} = t_3t_2$. Due to primary carry, four-gate delay is needed from (a_0, b_0) to c_3: one for (g_0, t_0) and three for carry chain in Eq. (6). Here using Ling's algorithms[10] we propose a new idea called carry-incorporating structure which is shown in less significant part of U8 in Fig. 3. In the four less significant bits of the block, a pseudo-carry called H_1 replaces c_1 to speed up carry propagation. The structure is expressed as:

$$H_1 = g_1 + g_0 + t_0c_{-1} \quad (7)$$

$$g_{32} = g_3 + t_3g_2 \quad (8)$$

$$t_{321} = t_3t_2t_1 \quad (9)$$

$$c_3 = g_{32} + t_{321}H_1 \quad (10)$$

And c_3 is available after 3-gate delay, namely one for (g, t), one for H_1 and another one for c_3. Thus 1 gate delay is saved compared to the scheme in Eq. (6) and there are 4 levels of gate delay from c_{-1} and (a_0, b_0) to c_7 the same as the path from (a_7, b_7) to c_7.

Then we designed carry-incorporating structure at transistor level for better optimization. Equation 7 is rewritten:

$$H_1 = a_1b_1 + a_0b_0 + (a_0 + b_0)c_{-1} \quad (11)$$

Complementary CMOS structure can be used to implement above logic function in 2-gate delay. But it has 3 serial transistors in path and thus causes extra delay. The circuit structure in Fig. 4 avoids such a time-consuming scheme using 20 transistors including necessary inverters. Circuit simulation results using 1.0μm process parameters showed that the proposed structure has a delay of 590ps while the complementary CMOS structure has a delay of 670ps. So the (a_0, b_0) to H_1 path can be realized in 2 gate delay and above analysis for Eq. 7 is justified.

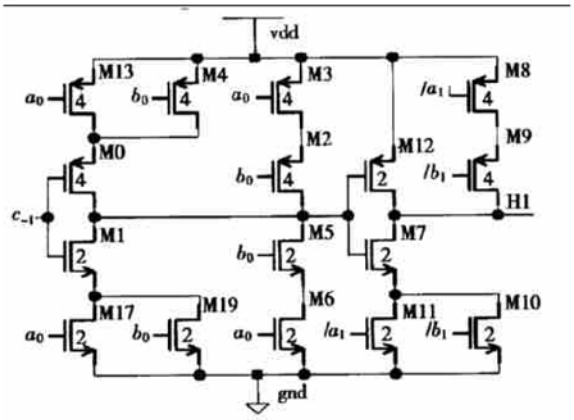

Fig. 4 Proposed carry-incorp structure

3.2 Sum logic of LSA

Sum logic in Eq. (2) shows that more significant 4-bit sums of U8 are ready after c_3 from less significant 4 bits passes an "and-xor" logic gate. CMOS implementation of the logic ($s = Ps \oplus (pc)$) is shown in Fig. 5.

The structure utilizing pass transistor tech-

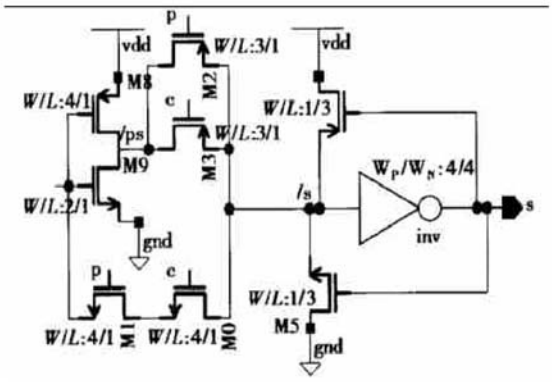

Fig. 5 "and-xor" structure

nique needs only 10 transistors. There are 2 critical paths. One is the path through M0 and M1 when signal ps is high. A weak "1" at node between M0 and M1 will charge the node "sbar" through M0 when the carry becomes high. The other is from node "psbar" through M2/M3 to node "sbar" when ps is high. The low-level signal "psbar" will discharge node "sbar" through a PMOS device. In the two cases, the level restorers M4 and M5 will act to restore voltage level at node "sbar" and cut static current through the transistors in inverter. As the latch formed by inverter and level restorers will fight during transition at node "sbar", careful transistor sizing is needed to guarantee proper work of the circuit.

Firstly, the 2 level restorers should be small enough so as not to beat the signal passing through the pass transistors during state transition at node "sbar". Only in this way, can the circuit function be correct. The M4 and M5 are thus inversely ratioed, namely the width is shorter than their length. Yet further drawing-up of the restoring devices increases the capacitance at the node "sbar", slowing down the transition of state at the node. Here the sizes of 2 restorers are swept to determine the optimal values. In Fig. 6 the sweeping waves of signals at node "sbar" and "s" are shown in the upper and lower part of the figure. The information shows that the voltage at node "sabr" is restored to V_{dd} by M4/M5. In the case of $W/L = 1\mu m/1\mu m$, the equivalent on-resistance of M5 is quite small and the signal through M0 and M1 must fight against the

strong current of M5, thus transition is slowed down. When the W/L becomes 2/3 and 1/3, the node "sbar" is easier to be charged/discharged and the propagation delay to node "s" is shortened.

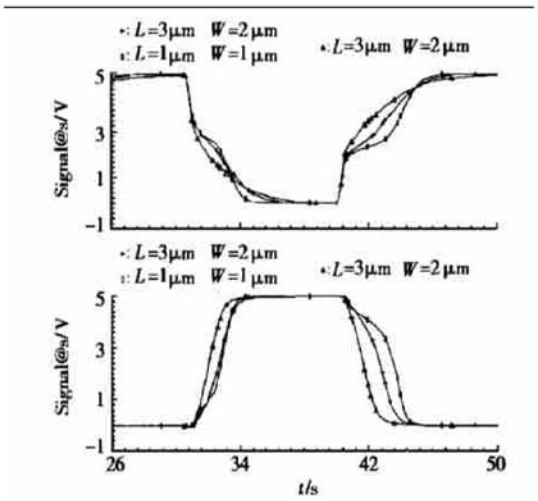

Fig. 6 Simulation waves of sum logic

Secondly, in LSA the carry will arrive after the ps and p, so the critical path from ps through M1, M0 and inverter to node "s" will never happen in addition operation. Based on the assumption, carry signal was put on the transistor M0 close to node "sbar" to speed up signal propagation. Also the M0/M1 and M2/M3 should be large enough to charge the capacitances of node "sbar" but not be over-enlarged to burden the loads of passing transistors.

Finally, the transistors were sized by starting without the level restorers. After the balanced rising and falling transition achieved, the sizes of passing transistors and inverter were decided. Then the restorers were added and tuned to get optimal result of the whole structure. As for power consideration during transistor sizing, there were some points where a small increase in speed caused a large increase in the power consumed. Here the gate widths were chosen at the point where the delay curve was approaching its minimum. This will ensure an optimal circuit speed without sacrificing power.

Figure 7 shows the layout in 1.0μm CMOS single poly, dual metal process. Simulation shows a delay of 1.2ns.

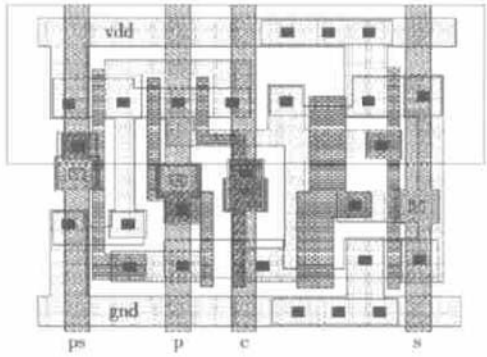

Fig. 7 Layout view of and-xor logic

4 Simulation results

The 32-bit LSA is simulated using 0.25μm and 1.0μm CMOS process parameters respectively. The adder was laid-out for 1.0μm technology and layout view is shown in Fig. 9 occupying an area of 2490μm×250μm and totaling 1390 transistors. Fig. 8 shows the simulation result for 1.0μm, 5V technology. The critical path is in the case of (00000000+ FFFFFFFF) with initial carry changing from "0" to "1". Simulation results of 32-bit LSA are listed in Table 1 together with recently reported outstanding adders.

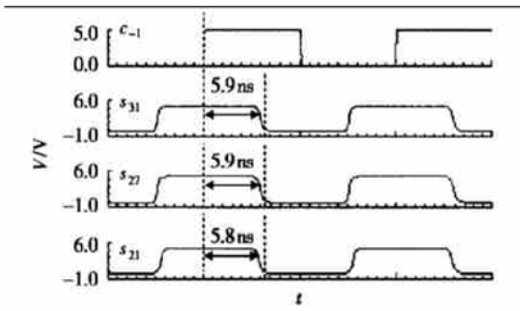

Fig. 8 Simulation result of LSA32

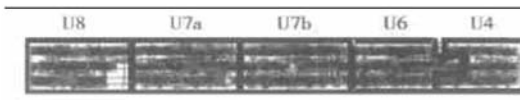

Fig. 9 Layout view of LSA32

Table 1 Performance comparison of adders

Adders	Process /μm	V_{dd} /V	Power @100MHz	Delay /ns	Area /mm^2
LSA32	1.0	5.0	23mW	5.9	0.62
	0.25	2.5	5.2mW	0.8	—
MCBA[14]	0.35	3.3	30μW@1MHz	2.8	—
DPL[12]	0.25	2.5	8mW@50MHz	1.5	0.6
CMOS[13]	0.6	3.3	34.3mW	2.33	—
CPL[13]			34.5mW	2.24	—
DPL[13]			27.5mW	1.98	—
ALU16[15]	0.6	3.3	54mW@200MHz	5	0.86

The CMOS, CPL, DPL structures are 32-bit carry lookahead adders reported[13] using CMOS, complementary pass-transistor logic and double pass-transistor logic. MCBA (manchester carry-by-pass adder)[14] is a carry skip adder. ALU16[15] is a 16-bit ALU whose adder uses ELM structure. In the deep submicro range of 0.25 and 0.35μm, LSA shows less area and power cost over DPL[12] whose conditional sum architecture needs extra hardware. The MCBA[14] of CSA structure shows advantage in power dissipation but much slower performance. In the range of 0.6 and 1.0μm, LSA shows the best power efficiency and its speed is not so slow considering the process difference and the fact that the circuit design of LSA in 1.0μm technology utilizes mostly minimum-equivalently sized transistors. The low-power, high performance property of logarithmic skip adder is due to the structure design, circuit design and transistor sizing.

An ALU with the 32-bit LSA as arithmetic component is included in an IP portfolio at Institute of Microelectronics, Peking University. The test chip is being fabricated at Wuxi Microelectronics Institute.

5 Conclusions

The 32-bit logarithmic skip adder is introduced. Focus is on the circuit design of building components of LSA at transistor level. Carry-incorporating structure to include the primary carry input in carry chain is redesigned to reduce logic depth of the adder. An "and-xor" circuit to implement sum logic of LSA is elaborated on. Simulation results show that high performance, low power and compact static CMOS adder design can be realized.

References

[1] Rabaey J. Digital integrated circuits: a design perspective. Prentice Hall, 1998

[2] Zimmermann R, Fichtner W. Low-power logic styles: CMOS versus pass transistor logic. IEEE J Solid-State Circuits, 1997, 32(7): 1079

[3] Hwang I S, Fisher A L. Ultra fast compact 32bit CMOS adder in multiple-output domino logic. IEEE J Solid-State Circuits, 1989, 24: 358

[4] Dozza D, Gaddoni M, Baccarani G. A 3.5ns, 64bit, carry-lookahead adder. 1996 IEEE International Symposium on Circuits and Systems, 1996: 297

[5] Koren I. Computer arithmetic algorithms. Prentice Hall, 1993

[6] Huang Z, Ercegovac M D. Effect of wire delay on the design of prefix adders in deep-submicro technology. 2000 Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers, 2000: 1713

[7] Nagendra C, Irwin M J, Ownes R M. Area-time-power trade-offs in parallel adders. IEEE Trans Circuits Syst, 1996, 43(10): 689

[8] Brent R P, Kung H T. A regular layout for parallel adders. IEEE Trans Comput, 1982, C-31(3): 260

[9] Yeh W, Jen C. On the study of logarithmic time parallel adders. 2000 IEEE Workshop on Signal Processing Systems, 2000: 459

[10] Ling H. High-speed binary adder. IBM J Res Develop, 1981, 25(3): 156

[11] Kuroda T. Overview of low-power ULSI circuit techniques. IEICE Trans Electron, 1995, E78(4): 334

[12] Suzuki M, Ohkubo N, Shinbo T. A 1.5-ns 32b CMOS ALU in double pass-transistor logic. IEEE J Solid-State Circuits, 1993, 28(11): 1145

[13] Ko U, Balsara T. Low-power design techniques for high-performance CMOS adders. IEEE Trans VLSI Syst, 1995, 13(2): 327

[14] Eriksson H, Larsson-Edefors P, Alvandpour A. A 2.8ns 30μW/MHz area-efficient 32-B Manchester carry-bypass adder. 2001 IEEE International Symposium on Circuits and Systems, 2001, (4): 84

[15] Ryu B S, Oh H S, Shim K H. Multi-level approaches to low power 16bits ALU design. The First IEEE Asia Pacific Conference on Signal Processing Systems, 1999: 158

对数跳跃加法器的静态 CMOS 实现*

贾　嵩　刘　飞　刘　凌　陈中建　吉利久

(北京大学微电子研究院, 北京　100871)

摘要: 介绍了一种 32 位对数跳跃加法器结构. 该结构采用 ELM 超前进位加法器代替进位跳跃结构中的组内串行加法器, 同 ELM 相比节约了 30% 的硬件开销. 面向该算法, 重点对关键单元进行了晶体管级的电路设计. 其中的进位结合结构利用 Ling 算法, 采用支路线或电路结构对伪进位产生逻辑进行优化; 求和逻辑的设计利用传输管结构, 用一级逻辑门实现"与-异或"功能; 1.0μm CMOS 工艺实现的 32 位对数跳跃加法器面积为 0.62mm^2, 采用 1μm 和 0.25μm 工艺参数的关键路径延迟分别为 6ns 和 0.8ns, 在 100MHz 下功耗分别为 23 和5.2mW.

关键词: 对数跳跃; 进位结合; 求和逻辑; 进位链

EEACC: 2570D; 1265B; 5120

中图分类号: TN432　　**文献标识码**: A　　**文章编号**: 0253-4177(2003)11-1159-07

* 预研资助项目(Nos. 41308010402, 415011005)

贾　嵩　男, 1970 年出生, 博士研究生, 研究方向为数字和混合信号 VLSI 设计、计算机算术等.

刘　飞　男, 1974 年出生, 博士研究生, 主要从事数模混合电路设计研究.

吉利久　男, 教授, 博士生导师, 主要研究数字和混合信号 CMOS 电路设计、优化算法等.

2003-01-02 收到, 2003-06-16 定稿