对数跳跃加法器的算法及结构设计

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: 本文介绍一种新型加法器结构))) 对数跳跃加法器, 该结构结合进位跳跃加法器和树形超前进位加法器算法, 将跳跃进位分组内的进位链改成二叉树形超前进位结构, 组内的路径延迟同操作数长度呈对数关系, 因而结合了传统进位跳跃结构面积小、功耗低的特点和 ELM 树形 CLA 在速度方面的优势. 在结构设计中应用 Lingcs 算法设计进位结合结构, 在不增加关键路径延迟的前提下, 将初始进位嵌入到进位链. 32 位对数跳跃加法器的最大扇出为 5, 关键路径为 8 级逻辑门延迟, 结构规整, 易于集成. spectre 电路仿真结果表明, 在 0.25μm CMOS 工艺下, 32 位加法器的关键路径延迟为 760ps, 100MHz 工作频率下功耗为 512mW.

关键词: 加法器; 对数跳跃; 结构设计; 进位结合
中图分类号: TP342+.21 文献标识码: A 文章编号: 0372-2112 (2003) 08-1186-04

Algorithm and Structure Design of Logarithmic Skip Adder

JIA Song, LIU Fei, LIU Ling, CHEN Zhong-jian, JI Li-jiu
(Institute of Microelectronics, Peking University, Beijing 100871, China)

Abstract: LSA (Logarithmic Skip Adder) Algorithm is introduced in this paper. LSA is a hybrid structure of carry skip adder and ELM carry lookahead adder, which replaces serial carry chain with binary tree structure. In structure design, a carry-incorporated structure to include the primary carry input in carry chain is found to reduce logic depth. With its regular structure, 32-bit LSA has a logic depth of 8 and a fanout no more than 5. Circuit simulating results in 0.25um CMOS process show a critical path delay of 760ps.

Key words: adder; logarithmic skip; structure design; carry-incorporated

1 引言

加法运算是微处理器中 ALU(算术逻辑单元)、MAC(乘累加器)和地址产生等部件中的基础操作, 这些部件通常位于处理器的关键路径上, 是影响处理器性能的瓶颈之一. 长期以来, 已经有许多成熟的算法和结构, 目前高速、低功耗加法器的设计依然是研究的热点. 在加法器的常用算法中, 行波进位(RCA)结构硬件和功耗开销最小, 但时间延迟随数据位数线性增加, 不适合高性能的需要; 超前进位加法器(CLA)的时延同操作位数呈对数关系, 该算法把进位链全部展开, 实现起来面积和功耗开销较大. 速度和开销介于 RCA 和 CLA 之间的是进位跳跃加法器(CSA), CSA 也可以看作是超前进位结构, 但是利用组进位产生信号和组进位传递信号之间的逻辑依赖关系省掉了组进位产生信号的逻辑, 因此节约开销. 但同 CLA 相比, CSA 组内的进位串行传递, 因此关键路径长, 速度较慢. 值得一提的是树形结构的 CLA[1~3], 也被称为 Parallel Prefix 加法器, 由于结构规整易于 VLSI 实现受到关注. 树形结构 CLA 具有结构紧凑、速度快的优点, 其关键路径长度与数据字长的对数呈正比, 理论上达到了极限. ELM 树形加法器被认为是功耗、延时乘积最优的一种并行加法器[4,5], 同其他 CLA 相比, 该结构把求和逻辑嵌入到进位链中, 减少单元之间的连线. 但其二叉树结构的进位链中扇出很大, 例如 32 位的 ELM 加法器, 最大扇出达到 17, 对于 VLSI 实现来说需要多级 Buffer, 增加关键路径延迟, 不利于获得高性能.

本文提出一种加法器结构))) 对数跳跃加法器(Logarithmic Skip Adder). LSA 结合进位跳跃加法器开销小和树形结构 CLA 延时短的优点, 克服相应的缺点. 通过结构分析和电路仿真可以看出该算法适合于加法器 VLSI 实现.

2 对数跳跃加法器算法

本设计提出的对数跳跃加法器是进位跳跃和树形超前进位算法的混合结构, 利用超前进位算法来增加进位在分组之间的跳跃距离, 从而缩短关键路径. 图 1 是等分的进位跳跃结构的框图[7], n 位操作数, 等分为 r 组, 每组 k 位. 传统跳跃进位[8]结构的分组中采用行波进位, 分组内的路径延迟与操作数位数呈线性关系; 本文提出的对数跳跃加法器将图 1 中分组内的进位链改成二叉树形超前进位结构, 组内的路径延迟同操作数长度呈对数关系, 称为对数跳跃加法器. 下面分析对

收稿日期: 2002-07-29; 修回日期: 2002-11-01
基金项目: 专用集成电路设计服务体系(No. 41501100501); 基于 0.15~1Lm 的 IP 库研究(No. 41308010402)

数跳跃加法器的关键路径延迟和逻辑实现开销.

加法器进位链的逻辑可以用下面的公式[2]表示:

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

相邻两位或两组向上一级的进位信号$(g_{i+1,i}, p_{i+1,i})$可以用一个$\circ$算子来计算. 其中, $g_i = a_i b_i$是进位产生信号, $p_i = a_i \oplus b_i$是进位传递信号, 在本文中用单位门延迟模型来估算各种加法器的关键路径长度. 单位门延迟模型假设每个逻辑门的平均延迟为 1.

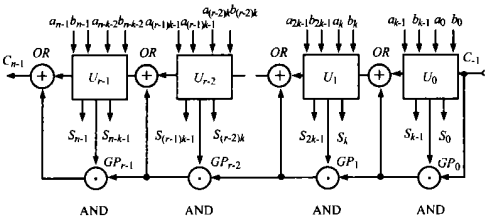

图 1　进位跳跃结构

采用分组内树形进位链的对数跳跃加法器中, 从(a_0, b_0)到最高位进位c_{n2}的关键路径可以用下式表示:

$$\$_{LSA} = \$_{p,g} + (2\log_2 k)\$_t + (n/k - 2)\$_s \quad (2)$$

其中, $\$_{p,g}$是进位产生和进位传递信号的一级门延迟. $\$_t$是分组内树形进位链的一级门延迟, 逻辑如式(1)所示. $\$_s$是组间跳跃的与或门逻辑的延迟. 将式(2)对 k 求导数, 并让导数为零, 可以算出 k 的最优值及对应情况下对数跳跃结构关键路径延迟:

$$k_{opt.} = (\ln 2/2)\, n \quad (3)$$

$$\$_{opt.} = (2\log_2 n - 1.11)\$ \quad (4)$$

其中$\$$是平均门延迟. 而采用分组内行波进位链结构的传统跳跃进位加法器的关键路径以及最佳分组如下所示, 其中$\$_r$是行波进位的一级门延迟:

$$\$_{CSA} = \$_{p,g} + 2(k-1)\$_r + (n/k - 2)\$_s \quad (5)$$

$$k_{opt.} = \sqrt{n/2} \quad (6)$$

$$\$_{opt.} = 2(\sqrt{2n} - 2)\$ \quad (7)$$

比较式(3), (6), 对数跳跃结构可以进行更大的分组, 因而减少组间跳跃. 对比式(4), (7)可以看出, 对数跳跃加法器的关键路径是操作数长度的对数, 而进位跳跃加法器是平方根关系. 在操作数位数较大的情况下, 对数跳跃结构可以满足高性能应用在速度上的要求.

进位跳跃结构利用进位产生和进位传递信号之间的逻辑依赖关系减少逻辑实现开销. 当进位链中第 m 组(m∈[0, r-1])中每位的两操作数都相异, 即该组进位传递信号: $p_{mk+k-1,mk} = \prod_{mk}^{mk+k-1} p_i = 1$., 则该段进位链向上的进位就是从低位进入的该段的进位, 也就是说低位来的进位跳过这一段进位链向上传递. 这一过程可以用以下公式表示:

$$c_{mk+k-1} = (g_{mk+k-1,mk} + p_{mk+k-1,mk} c_{mk-1}) + p_{mk+k-1} c_{mk-1} \quad (8)$$

括号内的表达式是完整的进位链逻辑, 最后一项是进位跳跃逻辑, 当组进位为 1 时, 该项把括号内的逻辑吸收, 快速产生向上的进位. 对于传统的进位跳跃结构, 只在开销最小的行波进位加法器基础上增加组进位传递逻辑, 即式(8)中第三项. 对于本设计中的对数跳跃结构, 则是用并行进位链实现式(8)中括号内的逻辑. 而超前进位结构则需要并行进位链和串行进位链两套逻辑. 因此对数跳跃结构的实现开销位于超前进位结构和传统进位跳跃结构之间.

对于分组内的树形进位链结构, 我们选择 ELM 算法, 图 5 是一个 7 位 ELM 加法器, 该结构通过构造二叉树形进位链, 并将求和逻辑嵌入到进位树中, 在本位进位产生的同时可以产生本位和, 与其他 CLA 相比节省一级门延迟. 其高四位求和逻辑如下式[6]所示:

$$S_i = p_i \oplus c_{i-1} = p_i \oplus (g_{i-1} + p_{i-1} c_{i-2})$$
$$= p_i \oplus g_{i-1} \dot{\vee} p_{i-1} c_{i-2} = ps_i \oplus p_{i-1,4} c_2 \quad (9)$$

其中, ps 为部分和, 对于 7 位结构, 高四位的部分和只与高四位的逻辑有关, 而与低三位和初始进位无关. 因此ps_i与进位传递信号$p_{i-1,2}$可以在进位传递的时候并行计算, 当低三位的进位传到高四位后, 只要经过一级- 与异或 逻辑, 就可以得到本位和.

ELM 算法被认为是功耗、延时乘积最优的一种并行加法器结构[4, 5], 但从图 5 可以看出, ELM 树形结构的扇出随操作位数迅速增加($fanout_{max} = n/2 + 1$), 不利于电路实现. 而本设计中采用 ELM 结构实现对数跳跃加法器的组内并行进位链逻辑, 利用其路径短、结构规整、连线少的优点, 同时由于是分组结构, 可以限制其过大的扇出.

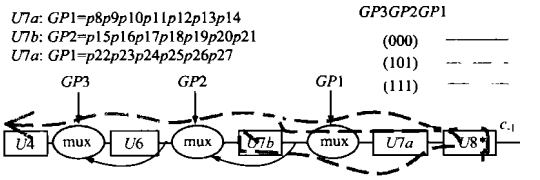

图 2　32 位对数跳跃加法器结构

3　对数跳跃加法器结构

图 2 是一个 32 位对数跳跃加法器的结构. 其中的各个模块如图 3、4、5 所示. 共有 5 个模块, 都采用 ELM 树形结构. 各组之间是完成进位跳跃的二选一多路器, 选择控制信号是前一组的组进位传递信号. 用多路器实现进位跳跃逻辑可以利用先行获得的选择信号提高进位跳跃的速度.

下面说明 32 位 LSA 的关键路径. 图 5 所示的 U7 模块中低位段有 3 位, 高位段有 4 位, 这样的结构使得该模块的初始进位到进位输出及各位和的延时是 3 级逻辑门. U6 模块的低位段是 2 位, 高位段是 4 位, 从初始进位到进位输出的延时是 2 级逻辑. 式(10)是 U7 模块的进位链的公式推导, 其中$p_{14,8}$是图 1 中 GP 1, $\circ$ 算子如式(1)所示. 从(a_i, b_i)(其中, i∈[8, 14])经过一级逻辑得到(g_i, p_i), 经过式(10)所示二叉树结构的 3 级门延迟获得该模块的组进位信号$(g_{14,8}, p_{14,8})$, 再经过式(11)所示一级多路器, 这样经过 5 级门延迟获得c_{14}.

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

$$c_{14} = P_{14,8}(g_{14,8} + p_{14,8} c_7) + p_{14,8} c_7 \quad (11)$$

通过类似的推导,图 2 结构的 32 位对数跳跃加法器的关键路径延时是 8 级逻辑, 图 2 中画了三种情况下的关键路径, 当 $GP_3GP_2GP_1$ 为(000)时,关键路径是从 U7a 的某位输入到 U7b 的某位和. 当为(101)时,关键路径从 U8* 模块的某位输入跳过 U7a 到 U7b 的某位和. 当为(111)时, 从 U8* 模块的某位输入跳过 U7a、U7b、U6 到 U4 的某位和. 表 1 中详细列出不同进位跳跃和传递情况下的关键路径. 图 2 所示 32 位对数跳跃加法器的关键路径为 8 级逻辑门延迟. 表 1 中最后一列是用 0.25Lm 工艺参数对 32 位 LSA 的电路仿真结果.

表 1 32 位 LSA 的关键路径

GP_3GP_2 GP_1	Critical path	Logic depth	Delay (ps)
000	(a_i, b_i) y s_j i I [8, 14], j I [18, 21]	8	731
001	(a_i, b_i) y s_j i I [0, 7], j I [18, 21]	8	748
010	(a_i, b_i) y s_j i I [8, 14], j I [18, 21]	8	754
	(a_i, b_i) y s_j i I [8. 14], j I [24, 27]		
011	(a_i, b_i) y s_j i I [0, 7], j I [24, 27]	8	701
100	(a_i, b_i) y s_j i I [8, 14], j I [18, 21]	8	750
101	(a_i, b_i) y s_j i I [0, 7], j I [18, 21]	8	763
110	(a_i, b_i) y s_j i I [8, 14], j I [24, 27]	8	755
	(a_i, b_i) y s_j, c_{31} i I [8, 14], j I [28, 31]		
111	(a_i, b_i) y s_j i I [0, 7], j I [18, 21]	8	728
	(a_i, b_i) y $s_j C_{31}$ i I [0, 7], j I [24, 27]		
	(a_i, b_i) y $s_j C_{31}$ i I [0, 7], j I [28, 31]		

对于 32 位加法器的关键路径, 4 位一组等分的传统 CSA 需要 14 级门延迟, ELM 结构需 6 级逻辑, 而图 2 结构 LSA 需要 8 级逻辑. 从逻辑结构的扇出上看, LSA 的最大扇出是 5, 而 ELM 结构有一个 17 的扇出和两个 9 的扇出, 过大的扇出需要额外的多级 buffer 提高驱动能力, 实际上增加关键路径长度.

对于加法器的 VLSI 实现来说, 电路的性能通常由加法器结构的关键路径门延迟、扇出和连线的复杂程度等因素综合决定. 从结构分析可以看出 LSA 在具有较短关键路径的同时保持小扇出; 分组中 ELM 结构最大为 8, 省去高位的逻辑, 同时保持其连线简单的优点. 此外 U8* 、U7、U6 模块中都有 U4 模块, 保持树形 CLA 结构规则的优点, 有利简化设计和布图.

4 进位结合结构的设计

在加法器的应用中, 进到最低位的初始进位经常是需要的. 特别是当用加法器实现补 2 的加/减功能时, 初始进位在减法中是必须的. 在常用的加法器实现中, 初始进位的信息是通过在全部部分进位产生后再用一级逻辑计算最终有效进位, 如图 3 中 4 位 ELM 加法器结构, 或者是在最低位加一级逻辑, 用初始进位产生最低位向上的有效进位, 再将进位逐级传递. 无论哪种方案都需要在关键路径上增加一级逻辑延迟. 本设计针对对数跳跃加法器, 采用 Lingcs 算法[9], 提出进位结合结构(carry incorporated structure), 在不增加逻辑门延迟的情况下把初始进位嵌入到加法器的进位链中. 结构图如图 4 中 U8* 模块的低位段所示.

当把初始进位加到树形加法器的最低位, 从结构上看二叉树失去原有的平衡, 最低位(a_0, b_0)到进位输出的延迟比其他路径长. 4 位树形结构的进位链逻辑可用下式表示:

$$c_3 = \{[(g_3, p_3)\ \mathrm{op}\ (g_2, p_2)]\ \mathrm{op}[(g_1, p_1)\ \mathrm{op}\ (g_0, p_0)]\}\ \mathrm{op}\ (c_{-1}) \quad (12)$$

由于有初始进位 c_{-1}, 从(a_0, b_0)到 c_3 需要四级逻辑, 到 c_7 需要五级逻辑, 比其他路径多一级门延迟. 下面采用两条措施解决这一问题. 本设计提出的进位结合结构中, 首先用 $t_i = a_i + b_i$ 作为进位传递信号, 这样在关键路径上就用一个或门代替了电路实现起来速度较慢的异或门, 减小了延时. 其

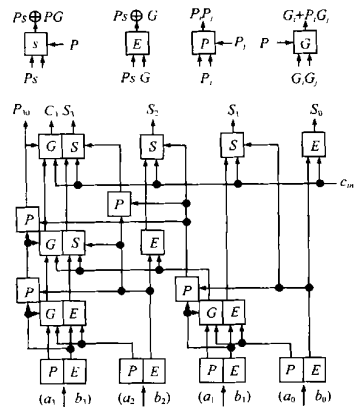

图 3 U4 模块

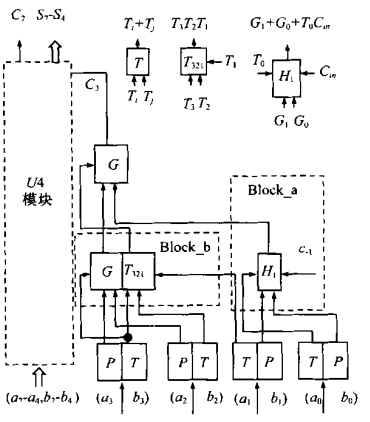

图 4 U8* 模块

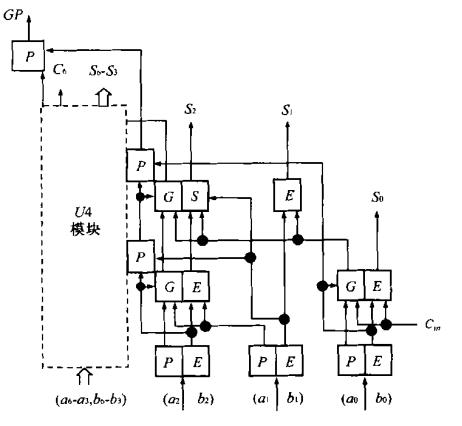

图 5 U7 模块

次, 在图 4 所示的低位段, 我们把 c_{-1}、g_0、t_0、g_1 送入 block_a 而把 t_1、g_2、t_2、g_3、t_3 送入 block_b, 而不是像传统二叉树把低 2 位的四个信号和初始进位送入 block_a, 把高 2 位的四个信号送入 block_b. 在新的进位结合结构中, block_a 的输出 H 并不是有效地进位信号, 称为伪进位信号. 该信号在下一级与 block_b 的输出信号结合后, 输出有效进位信号 c_3. 这些信号的逻辑表达式如下:

$$H_1 = g_1 + g_0 + t_0 c_{-1} \quad (13)$$

$$G_{32} = g_3 + t_3 g_2 \quad (14)$$

$$T_{321} = t_3 t_2 t_1 \quad (15)$$

$$C_3 = G_{32} + T_{321} H_1 \quad (16)$$

这样原来用两级门实现的 C_1 的运算变成可以用一级逻辑门实现的 H_1 和 T_{321}. 本设计提出的进位结合结构可以减少一级逻辑的关键路径延迟. 这样, 同 (a_7, b_7) 到 c_7 的路径一样, 从 c_{-1} 和 (a_0, b_0) 到 c_7 的关键路径也是四级逻辑.

在图 2 结构的 U7、U6 模块中, 进位结合是通过减少操作数的位数来实现的. 7 操作数或 6 操作数加上初始进位可以获得平衡的树形结构. 图 4 所示的 $U8^*$ 模块运用进位结合结构, 使得包含初始进位 c_{-1} 的操作数到本模块的进位输出的延时是 4 级逻辑.

5　仿真结果

为了验证以上结构分析的结果, 我们采用 0.25Lm, 2.5V, CMOS 工艺对静态 CMOS 实现的 32 位对数跳跃加法器进行 spectre 仿真, 共用 1372 个 MOS 晶体管. 为了便于评价性能, 将本设计的仿真结果同近期文献报道 32 位加法器的结果进行了比较, 见表 2.

表 2　不同加法器的性能比较

adders	process(Lm)	Vdd(V)	power@100MHz	delay(ns)
LSA32	0.25	2.5	5.2mW	0.76
CMOS[11]	0.6	3.3	34.3mW	2.33
CPL[11]			34.5mW	2.24
DPL[11]			27.5mW	1.98
DPL32[10]	0.25	2.5	8mW@50MHz	1.5
ANT[12]	0.35	)	)	2.8

其中表里的 CMOS, CPL, DPL 是文献[11]报道的分别用互补 CMOS、CPL(Complementary Pass2transistor Logic) 逻辑和 DPL(Double Pass2transistor Logic) 实现的 32 位 CLA 加法器的仿真结果. DPL32 是文献[10]报道的 DPL 逻辑实现条件进位选择加法器的测试结果. ANT 是动态电路实现的 all2N2transistor32 位加法器的仿真结果[12]. LSA32 的功耗统计是表 1 中所列 8 种关键路径的平均功耗. 电路仿真结果显示, 32 位对数跳跃加法器可以工作在 1.25GHz 的频率, 功耗约为文献[10]中报道同等工艺条件下结果的三分之一, 因而适合高性能、低功耗的加法器 VLSI 实现.

6　结论

本文介绍一种对数跳跃加法器结构. 该加法器是进位跳跃和树形 CLA 的混合结构. 由于该结构将进位跳跃算法的分组内串行进位链改成二叉树形超前进位结构, 因此组内的路径延迟同操作数长度呈对数关系, 从而结合了进位跳跃结构节约功耗的特点和 ELM 树形 CLA 在速度方面的优势, 有利于提高速度, 减少功耗. 在结构设计中利用不等分结构缩短关键路径, 并设计进位结合结构, 在不增加关键路径延迟的前提下, 将初始进位嵌入到进位链. 通过结构分析可以看出, 32 位 LSA 结构规则, 关键路径上的最大扇出为 5. 在 0.25Lm CMOS 工艺下仿真结果表明关键路径延迟 760ps, 100MHz 频率下的功耗为 5.2mW, 适于高性能的加法器 VLSI 实现.

参考文献:

[1] P M Kogge, et al. A parallel algorithm for the efficient solution of a general class of recurrence equations [J]. IEEE Trans. on Computers, 1973, C222: 786- 793.

[2] Brent, et al. A Regular Layout for Parallel Adders [J]. IEEE Trans. on Computers, 1982, C31(3): 260- 264.

[3] S Knowles. A family of adders [A]. 1999, Proc. 14th IEEE Symp. On Computer Arithmetic [C]. Adelaide, Australia. 1999. 277- 284.

[4] T P Kelliher, et al. ELM2A fast addition algorithm discovered by a program [J]. IEEE Transactions on Computers, 1992, 41. 1181- 1184.

[5] C Nagendra, et al. Area2time2power tradeoffs in parallel adders [J]. IEEE Transactions on Circuits and Systems, 1996, 43(10): 689- 702.

[6] W2H Yeh. On the study of logarithmic time parallel adders [A]. IEEE Workshop on Signal Processing Systems [C]. Lafayette, LA, USA. 2000. 459- 466.

[7] V G Oklobdzija. VLSI Arithmetic [DB/OL]. http://www.ece.uc2davis.edu/acsel.

[8] I Koren. Computer Arithmetic Algorithms [M]. Prentice hall, New Jersey, USA. 1993. 84- 86.

[9] H Ling. High2speed binary adder [J]. IBM J. RES. DEVELOP. 1981, 25(3): 156- 165.

[10] M Suzuki, N Ohkubo, T Shinbo. A 1.52ns 322b CMOS ALU in double pass2transistor logic [J]. IEEE Journal of Solid2State Circuits, 1993, 28(11): 1145- 1150.

[11] U Ko. Low2power design techniques for high2performance CMOS adders [J]. IEEE Transaction on VLSI Systems, 1995, 3(2): 327- 333.

[12] C Wang, et al. A 1.25 GHZ 322bit tree2structured carry lookahead adder [A]. IEEE Intl. Symp. on Circuit and Systems [C]. Sydney, Australia, 2001. 4: 80- 83.

作者简介:

贾　嵩　男, 1970 年生于辽宁鞍山, 1992 年获哈尔滨工业大学精密仪器专业学士学位, 2000 年获南开大学微电子专业硕士学位, 目前在北京大学微电子研究院攻读博士学位, 研究领域包括 CMOS 数字和混合信号 VLSI 电路设计, 计算机体系结构.