

基于 10Gbase-KR 协议的 PCS 层弹性缓冲器设计

杨钞翔,张 春,王自强,王志军,谢 翔,姜汉钧

(清华大学 微电子学研究所,北京 100084)

摘 要: 为了实现高速数据流的跨时钟域传输,根据万兆以太网 10Gbase-KR 协议设计了一款性能稳定的弹性缓冲器,并对设计进行逻辑综合和门级仿真,验证了设计的正确性. 该弹性缓冲器采用常半满控制方式,深度为 16,数据传输速率为 10Gbps,读写时钟频率为 156.25 MHz,通过检测读写地址差值变换情况来比较读写速度,并自动插入或删除 IDLE 字符,以此调节弹性缓冲器读写速度完成时钟频率补偿,实现了高速数据流的跨时钟域正确传输.

关键词: 弹性缓冲器;10Gbase-KR 协议;跨时钟域;频率补偿

中图分类号: TN492

文献标识码: A

文章编号: 1000-7180(2018)03-0014-05

Design of Elastic Buffer in Physical Coding Sublayer Based on 10 Gbase-KR

YANG Chao-xiang, ZHANG Chun, WANG Zi-qiang, WANG Zhi-jun,
XIE Xiang, JIANG Han-jun

(Institute of Microelectronics, Tsinghua University, Beijing 100084, China)

Abstract: An elastic buffer is designed to satisfy the need of clock domain crossing of the high-speed data flow based on the 10Gbase-KR ethernet protocol. Then the logical synthesis and gate-level simulation is conducted to demonstrate the design is correct. The elastic buffer is controlled with the normal half-full method, of which the depth is 16, the data transfer rate is 10Gbps, and the read-write clock frequency is 156.25MHz. By detecting the changes of the difference between the reading and writing address, it can make a contrast with the reading and writing speed and automatically insert or delete the IDLE characters. In the way, it can adjust the elastic buffer to realize clock frequency compensation and the clock domain crossing of the high-speed data flow correctly.

Key words: elastic buffer; 10Gbase-KR; clock domain crossing; frequency compensation

1 引言

弹性缓冲器最早可追溯到 60 年代,在 1963 年 Maurice Karnaugh 就已获得这一技术专利,当时该技术应用于 PCM 电话网络. 如今高速传输方式转向串行时,弹性缓冲器的价值再次显现,在 USB、InfiniBand 架构、光纤通道和以太网等领域都有着广泛使用. 本文主要针对万兆以太网 10Gbase-KR 协议设计弹性缓冲器.

10Gbase-KR 属于万兆以太网协议,对应于 2007 年发布的 IEEE 802.3ap 标准^[1],主要用于背

板应用,如刀片服务器、路由器和交换机的集群线路卡等,所以又称之为“背板以太网”. 10Gbase-KR 协议定义了一条串行通道,采用 64/66B 编码方式实现 10Gbps 高速传输. 协议包含 MAC 层(介质访问控制层)、PCS 层(物理编码子层)、PMA 层(物理介质连接层)等,由于 MAC 层和 PCS 层的时钟往往不同源,必然存在频率偏差,所以 MAC 层的数据要传给 PCS 层,必须通过一个异步缓冲器跨时钟域. 又由于传输速率高达 10Gbps,即使频率偏差很小,随着时间不断累加异步缓冲器迟早会被“写满”或者“读空”,所以需要在 PCS 层设计一个具有“弹性”的

异步缓冲器,既完成数据的跨时钟域,又通过频率补偿调节两边读写指针速率,使缓冲器一直保持在“半满”最优状态^[2].

2 弹性缓冲器

2.1 作用

10Gbase-KR 协议规定通道内数据传输速率为 10Gbps,MAC 层传入 PCS 层的数据位宽为 64 bits,所以时钟频率为 156.25 MHz,对应时钟周期为 6 400 ps,时钟允许的误差范围为 ± 100 ppm,即总线允许的频率范围为 156.234 375~156.265 625 MHz,时钟周期范围为 6 399.360~6 400.640 ps.由于 MAC 层和 PCS 层时钟不同源,必定存在时钟偏差,根据时钟单元间距 UI 计算,156.25 MHz 的一个时钟单元间距为 6 400 ps,考虑有 ± 100 ppm 的余裕,则最坏情况下一个时钟单元间距为 6 399.360 ps 和 6 400.640 ps,此时两端最大时钟间距偏差为 1.28 ps.因此每隔 5 000 个周期,就可能移动一个时钟周期(6 400 ps),必将导致数据传输异常甚至错误.所以需要在 MAC 层与 PCS 层之间加入弹性缓冲器进行数据缓存,才能避免数据传输错误,且至少每隔 5 000 个时钟周期就需要向数据中插入 IDLE 字符用于频率补偿. IDLE 字符由协议定义,可供插入或者删除,穿插在数据当中,当数据包传送到上层时,上层协议可以自动解析 IDLE 字符,不影响真实数据的传输^[3].

当 MAC 层时钟频率大于 PCS 层时钟频率时, FIFO 写入速度大于读出速度,异步缓冲器将会被写满溢出,此时需要通过删除数据中的 IDLE 字符,相当于降低写入速度.反之,当 MAC 层时钟频率小于 PCS 层时钟频率时, FIFO 写入速度小于读出速度,异步缓冲器将会被读空,此时需要在数据中插入 IDLE 字符,相当于提高写入速度.通过这样在适当时机插入或删除 IDLE 字符,调节弹性缓冲器两边读写速度的方式可自动补偿两边的频率偏差,使弹性缓冲器一直保持在“半满”最优状态.

2.2 位置和容量

MAC 层的数据位宽为 64 bits,时钟频率 156.25 MHz,通过 XGMII 接口输入到 PCS 层,弹性缓冲器主要用于 MAC 层与 PCS 层之间数据的跨时钟域和时钟频率补偿,所以将它设计在 PCS 层的最前端.使 MAC 层的数据经 XGMII 接口,先输入到弹性缓冲器,进行缓存处理后再传给 PCS 层后续的编码、扰码等模块,完成 PCS 层所有操作后再传给 PMA 层,弹性缓冲器位置图如图 1 所示.

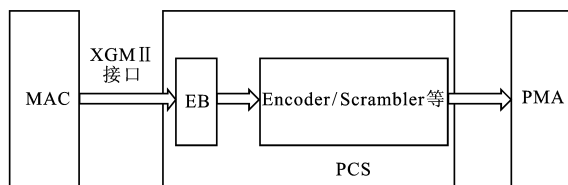


图 1 弹性缓冲器位置图

弹性缓冲器 EB 接收到的数据为 64 bits,此外还带有 8 bits 控制码,所以设置弹性缓冲器的宽度为 72 bits、深度为 16,这样足以满足设计要求.

3 基本原理

弹性缓冲器的实现方式常见的有两种:常半满方式和流程控制方式.常半满方式通过比较弹性缓冲器两边读写地址差值,自动插入或者删除 IDLE 字符,调节缓冲器两边读写速率,实现时钟频率补偿,控制弹性缓冲器一直维持在半满最优状态.当写指针速度大于读指针速度时,删除数据中的 IDLE 字符,降低写入速度.当写指针速度小于读指针速度时,在数据中插入 IDLE 字符,提高写入速度.流程控制方式和常半满方式有所区别,这种方式不用把缓冲器维持在半满状态,而是只维持在一个底端进入点的状态,当检测到 IDLE 字符时,若写指针速度大于读指针速度,则工作方式同常半满方式一样,但是当写指针速度小于读指针速度时,则产生使能信号,控制读指针暂停读操作,将缓冲器维持在底端进入点的状态^[4].

两种方式相比,常半满方式实现的弹性缓冲器其缓冲器深度较深,等待时间较长,但其电路实现简单,而且不用门控时钟,有利于 DFT 的实现.流程控制方式则可降低缓冲器的深度,相应等待时间较短,但需要添加额外的电路模块去产生相应的使能信号,相比更为复杂,同时该使能信号必须控制后续解码器等电路的时钟,即会产生门控时钟,不利于 DFT 的实现.本文采用常半满方式实现弹性缓冲器.

4 具体实现

弹性缓冲器本质上为一个异步 FIFO(First In First Out),可以在两个不同时钟域系统之间快速而方便地进行数据传输.本次设计的弹性缓冲器结构示意图如图 2 所示,它包含的主要模块有:双口存储器模块、写地址产生逻辑模块、读地址产生逻辑模块、二进制码转格雷码模块、跨时钟域模块、格雷码

转二进制码模块、读写地址差值计算模块^[5]. 图中写数据和读数据都为 72 bits,其中包括 64 bits 纯数据和 8 bits 控制码. 双口存储器采用两写一读模式,写地址有两个分别为 w_1 和 w_2 ,读地址只有一个 r_1 . 写入部分属于 MAC 子层,受 XGMII_CLK 时钟控制,读出部分属于 PCS 子层,与 PMA 子层共用一个时钟,受 PMA_CLK 时钟控制.

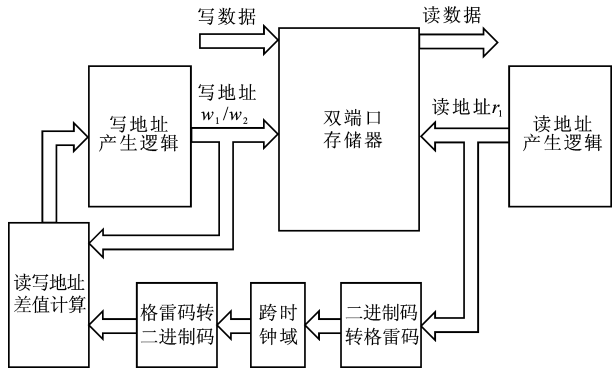


图 2 弹性缓冲器结构示意图

4.1 双端口存储器

双端口存储器受两个时钟域 XGMII_CLK 和 PMA_CLK 同时控制,XGMII_CLK 时钟控制写入部分,PMA_CLK 时钟控制读出部分. 双端口存储器深度为 16,数据位宽为 72 bits. 由于弹性缓冲器深度为 16,所以读写地址采用二进制编码 4'b0000~4'b1111. 弹性缓冲器需要通过插入或者删除 IDLE 字符调节双端口存储器两边读写速率,使其维持在半满最优状态,其具体操作可分为三种:正常读写、插入 IDLE、删除 IDLE.

(1)正常读写

当弹性缓冲器两边时钟读写速度相当时,不用进行插入或者删除 IDLE 字符的操作,只需正常读写,如图 3 所示. 左边部分为第一拍,假如写指针 w_1 和 w_2 分别指向地址 8 和 9,读指针 r_1 指向地址 0,从 MAC 层传来的待写数据为 72 bits,包含低 36 位 A 和高 36 位 B,只需要将 A 和 B 分别写入 w_1 对应地址 8 的低 36 位和高 36 位, w_2 对应地址 9 的高低 36 位都不操作,为空. 右边部分为下一拍,此时 w_1 和 w_2 指针在第一拍的基础上都加 1,新地址对应 9 和 10,待写数据为 C 和 D,将 C 和 D 分别写入地址 9 的低 36 位和高 36 位,地址 10 的高低 36 位都不操作,为空. 只要不插入或删除 IDLE 字符,就按照此方式正常写入数据,其中写指针 w_1 和 w_2 受 XGMII_CLK 时钟控制. 读指针 r_1 受 PMA_CLK 时钟控制,每一拍都自加 1,正常读出数据即可.

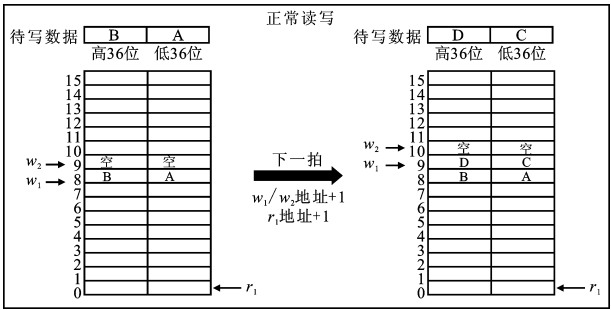


图 3 正常读写操作图

(2)插入 IDLE

双端口存储器写速度小于读速度时,弹性缓冲器会进行插入 IDLE 的操作,以此提高写速度实现平衡,如图 4 所示. 写指针 w_1 和 w_2 受 XGMII_CLK 时钟控制,读地址 r_1 受 PMA_CLK 时钟控制. 图中左边部分为第一拍,将待写数据 A 和 B 分别写入 w_1 对应地址 8 的低 36 位和高 36 位, w_2 对应地址 9 的低 36 位插入 IDLE 字符,高 36 位不操作为空或者也插入 IDLE 字符. 下一拍对应图中右边部分, w_1 和 w_2 地址都加 1,新地址对应 9 和 10,此时将待写数据 C 和 D 分别写入地址 9 的高 36 位和地址 10 的低 36 位,在地址 10 的高 36 位插入 IDLE 字符. 读指针 r_1 每一拍自加 1,正常读出数据即可.

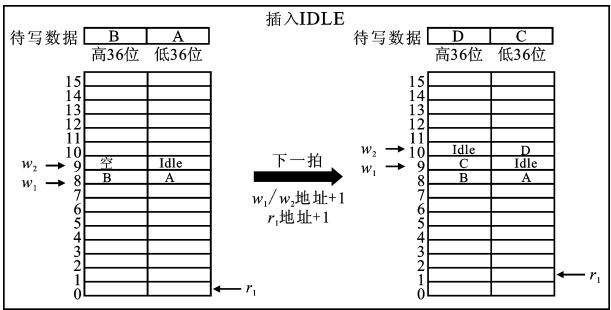


图 4 插入 IDLE 操作图

插入 IDLE 和正常读写两种操作对比,可以看出插入 IDLE 提高了写入速度,能够实现双端口存储器两边读写速度的平衡.

(3)删除 IDLE

双端口存储器写速度大于读速度时,弹性缓冲器会进行删除 IDLE 的操作,以此降低写速度实现平衡,如图 5 所示. 写指针 w_1 和 w_2 受 XGMII_CLK 时钟控制,读地址 r_1 受 PMA_CLK 时钟控制. 图中左边部分为第一拍,待写数据为 A 和 B,当数据包包含 IDLE 字符时,就可进行删除 IDLE 操作. 假设此时检测到 A 为 IDLE 字符,则将 A 舍弃,只将 B 写入 w_1 对应地址 8 的低 36 位,地址 8 的高 36 位和地

址 9 的高低 36 位都不操作,为空。下一拍对应图中右边部分, w_1 和 w_2 地址都保持不变,新地址仍为 8 和 9,假如检测到待写数据中 C 为 IDLE 字符,则将 C 舍弃,只将 D 写入地址 8 的高 36 位,地址 9 的高低 36 位都不操作,为空。读指针 r_1 每一拍自加 1,正常读出数据即可。

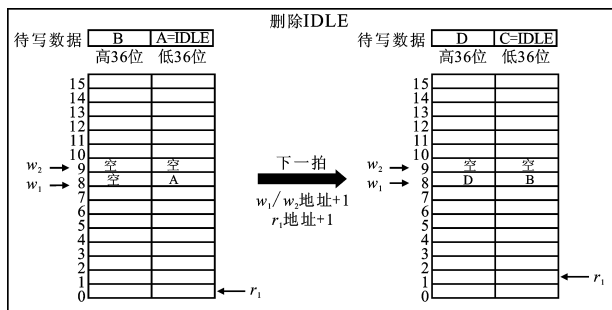


图5 删除 IDLE 操作图

删除 IDLE 和正常读写两种操作对比,可以看出删除 IDLE 降低了写入速度,能够实现双端口存储器两边读写速度的平衡。

4.2 写地址

写地址包含两个相邻的 w_1 和 w_2 ,一方面写地址要输入到双端口存储器,另一方面写地址还要传输到“读写地址差值计算模块”,具体作用将在后续内容中介绍。写地址 w_1 和 w_2 由写地址逻辑模块生成,这个模块属于 XGMII_CLK 时钟域,每一拍 w_1 和 w_2 可以进行 +0、+1、+2 三种操作,具体由“读写地址差值计算模块”的结果控制,在前面介绍双端口存储器时已经对具体操作进行了介绍,此处不再赘述。

4.3 读地址

读地址只有一个 r_1 ,由读地址逻辑模块生成,此模块受 PMA_CLK 时钟控制, r_1 每一拍只进行 +1,送到双端口存储器,按正常速度读出数据。此外,读地址 r_1 还需要送到二进制码转格雷码模块进行转换,再送到跨时钟域模块进行跨时钟域。因为地址数据采用 4b0000~4b1111 二进制编码,在跨时钟域时变化位数较多,setup/hold 关系难以保证,很容易发生亚稳态导致出错。所以一方面需要采用二进制码转格雷码,因为格雷码每次只变化 1 位,变化位数少,要么变化为下一值要么保持原值,能够很好地降低亚稳态的概率。另一方面还要用两级 D 触发器进行同步,研究表明两级 D 触发器能够很好降低亚稳态的概率。

读地址 r_1 完成以上操作后就能很好地实现了跨时钟域,然后再通过格雷码转二进制码模块转换成二进制码,将其输入到读写地址差值计算模块就

能计算出读写地址差值。

4.4 读写地址差值

读写地址差值模块属于 XGMII_CLK 时钟域,将写地址 w_1 和已经完成跨时钟域的读地址 r_1 送到此模块,即可计算出读写地址差值。定义此差值为 w_diff_r :

$$w_diff_r = \begin{cases} w_1 - r_1, & w_1 - r_1 \geq 0 \\ w_1 - r_1 + 16, & w_1 - r_1 < 0 \end{cases} \quad (1)$$

由于采用半满控制方式,设置 w_diff_r 初值为 8,当检测到 $w_diff_r \geq 10$ 时,则使能信号 rem_req 置 1,启动删除 IDLE 操作,降低双端口存储器的写入速度,直到 w_diff_r 恢复为 8 时 rem_req 使能信号清零,停止删除 IDLE 按照正常读写方式继续进行操作。当检测到 $w_diff_r \leq 6$ 时,则使能信号 add_req 置 1,启动插入 IDLE 操作,提高双端口存储器的写入速度,直到 w_diff_r 恢复为 8 时 add_req 使能信号清零,停止插入 IDLE 按照正常读写方式继续进行操作。如果读写速率相差太大,通过插入或者删除 IDLE 字符都不能完成频率补偿,弹性缓冲器将会被写满或者读空。当 w_diff_r 不受控制一直增加到 16 时,将 $tx_fifo_overflow$ 标志信号置 1 表示 FIFO 被写满,并进行复位。当 w_diff_r 不受控制一直减小到 0 时,将 $tx_fifo_underflow$ 标志信号置 1 表示 FIFO 被读空,并进行复位。

5 仿真结果及分析

本文采用 Verilog 硬件描述语言完成弹性缓冲器 RTL 级代码设计,设计测试激励,用 Modelsim 进行功能仿真,功能仿真正确后使用 Vivado 进行逻辑综合和门级仿真,门级仿真结果如图 6~图 9 所示。

5.1 插入 IDLE

弹性缓冲器写速度小于读速度时,门级仿真结果如图 6 所示,图中表明读写地址差值 w_diff_r 从初值 8 逐渐减小,当减小到 6 时使能信号 add_req 置 1,启动插入 IDLE 操作提高写速度进行频率补偿,之后差值 w_diff_r 从 6 逐渐回升,再次回到 8 时 add_req 清 0,停止插入 IDLE 字符。以后再检测到差值 $w_diff_r \leq 6$ 时,仍照此插入 IDLE。

将图 6 放大,时间标记仍为 61 789. 497 ns 附近,可观察插入 IDLE 的细节操作,如图 7 所示。其中 $txd[63:0]$ 和 $txc[7:0]$ 为从 MAC 层传来的待写数据, $txd_xfi[63:0]$ 和 $txc_xfi[7:0]$ 为从弹性缓冲器读出的数据。图中长箭头的起始端记为 txd 和 txc 的第 1 拍,终止端记为 txd_xfi 和 txc_xfi 的第 1 拍,

其中 txd=txd_xfi=64'h7b7a7978_77767574,txc=txc_xfi=7'b0000_0000,表明第 1 拍数据传输正确.第 2 拍 txd=txd_xfi=64'h07070707_070707fd,txc=txc_xfi=7'b1111_1111,说明第 2 拍数据传输也正确,并且 txc[7:4]=4'b1111,txd[63:32]=32'h07070707,表示待写数据包包含 IDLE 字符(IDLE=32'h07070707),可以在其后面紧跟着插入 IDLE 字符.第 3、4 拍 txd=64'h07060504_000000fb/64'h0f0e0d0c_0b0a0908,txd_xfi=64'h000000fb_07070707/64'h0b0a0908_07060504,实际上第 3 拍 txd_xfi 插入了 IDLE 字符(32'h07070707),后边数据接在 IDLE 之后顺序读出.

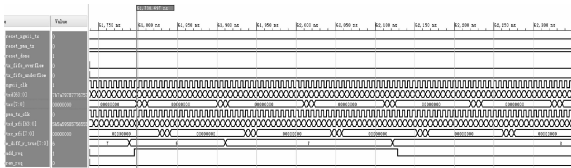


图 6 插入 IDLE 效果图

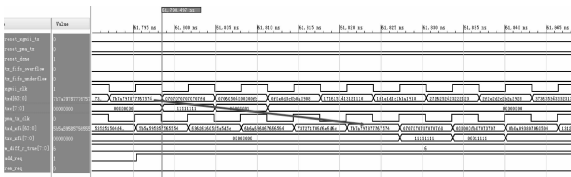


图 7 插入 IDLE 细节操作图

5.2 删除 IDLE

弹性缓冲器写速度大于读速度时,门级仿真结果如图 8 所示,图中表明读写地址差值 w_diff_r 从初值 8 逐渐增大,当增大到 10 时使能信号 rem_req 置 1,启动删除 IDLE 操作降低写速度进行频率补偿,之后差值 w_diff_r 从 10 逐渐回落,当再次回到 8 时 rem_req 清 0,停止删除 IDLE 的操作.以后再检测到差值 $w_diff_r \geq 10$ 时,仍照此删除 IDLE.

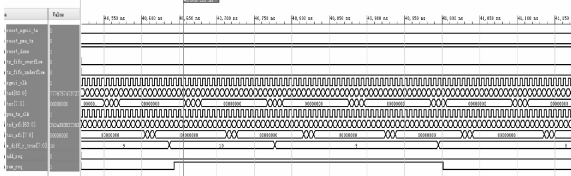


图 8 删除 IDLE 效果图

将图 8 放大,时间标记仍然为 40 636.997 ns 附近,可观察到删除 IDLE 的细节操作,如图 9 所示.其分析方法和插入 IDLE 情况相同,图中可以很容易地证明删除 IDLE 有效且数据传输正确.

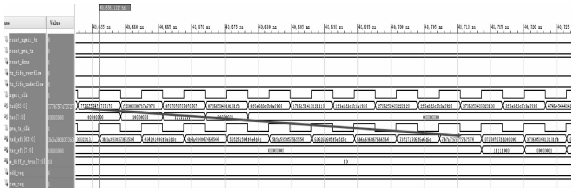


图 9 删除 IDLE 细节操作图

6 结束语

本文设计了一种基于 10GBASE-KR 万兆以太网协议的 PCS 层弹性缓冲器,采用常半满控制方式,检测弹性缓冲器读写地址差值变化情况,比较其读写速度,并动态插入或删除 IDLE 字符实现了时钟频率补偿,也正确完成了数据的跨时钟域传输.

参考文献:

[1] IEEE Std 802.3apTM—2007 (Amendment to IEEE Std 802.3TM—2005). LAN/MAN Standards Committee of the IEEE Computer Society[S]. IEEE 3 Park Avenue, New York,USA, 2007:1-2.

[2] Chang H, Peng T, Ke D, et al. Design and implementation of elastic buffer for Universal Serial Bus 3.0[J]. Ieice Electronics Express, 2016,57(5):5-35.

[3] Frazier H M. Aggregation at the physical layer[J]. IEEE Communications Magazine, 2008,46(2):S12.

[4] Michelogiannakis G, Balfour J, Dally W J. Elastic-buffer flow control for on-chip networks[C]// IEEE, International Symposium on High PERFORMANCE Computer Architecture. Vienna, Austria, IEEE, 2009:151-162.

[5] Verilog E, Cummings C E. Simulation and synthesis techniques for asynchronous FIFO design[J]. Snug, 2002,Section TB2(3):1-18.

作者简介:

杨鈔翔 男,(1991-),硕士.研究方向为数字集成电路设计.

张 春(通讯作者) 男,(1972-),博士,副研究员.研究方向为数模混合信号集成电路设计、嵌入式微处理器设计. E-mail:zhangchun@tsinghua.edu.cn.

王自强 男,(1975-),博士,副研究员.研究方向为模拟集成电路设计.

王志军 男,(1982-),硕士,工程师.研究方向为数字集成电路设计.

谢 翔 男,(1971-),博士,副研究员.研究方向为 SoC 设计、图像处理.

姜汉钧 男,(1980-),博士,副研究员.研究方向为数模混合集成电路设计.