DOI: 10.13718/j.cnki.xdzk.2024.04.019
张思言，杜周南，任一心，等. 一种双三次插值实时超分辨率 VLSI 设计 [J]. 西南大学学报(自然科学版)，2024，46(4)：202-212.

一种双三次插值实时超分辨率 VLSI 设计

张思言，　杜周南，　任一心，　邓涛，　唐曦

西南大学 物理科学与技术学院，重庆 400715

摘要：视频超分辨率技术具有广阔的应用前景，但基于深度学习方法的算法复杂度过高，难以实现实时计算. 因此，近年来研究者们开始探索基于现场可编程逻辑门阵列(Field Programmable Gate Array，FPGA)的超分辨率算法加速器，以利用 FPGA 的优势来提高算法的性能和能耗，实现实时的视频超分辨率. 设计了一种基于 FPGA 的高效高速双三次线性插值超大规模集成电路(Very Large Scale Integration Circuit，VLSI)架构，可用于 4 倍实时视频超分辨率. 该 FPGA 架构解决了实现双三次插值过程中所需的复杂内存访问模式的问题，并提出了一种基于乒乓操作的数据重排硬件设计，将算法输出的特定顺序数据重新以行为主进行排列，使得硬件能够直接或较为简单地对接 HDMI 等视频接口. 此外，采用状态机、流水线等方式降低设计功耗和减少时序违例，使得整个硬件设计可以更高频率运行. 本研究在 Zynq-7020 FPGA 上实现了硬件架构，能够实时将 qHD(960×540)的视频超采样为 UHD(3 840×2 160)高清视频. 实验结果表明，该硬件设计只需缓存 1 行图像像素，延迟仅为 9.6 μs，帧率达到 192.9 Hz，成功实现实时处理. 游戏图像数据集的测试结果表明，该设计峰值信噪比最高可达 35.67 dB，结构相似度达到 96.3%.

关　键　词：双三次插值；实时超分辨率；现场可编程逻辑门阵列；超大规模集成电路

中图分类号：TP301.6　　**文献标志码**：A　　开放科学(资源服务)标识码(OSID)：

文 章 编 号：1673-9868(2024)04-0202-11

A Real Time Super Resolution VLSI Design Based on Bicubic Interpolation

ZHANG Siyan，　DU Zhounan，　REN Yixin，
DENG Tao，　TANG Xi

School of Physical Science and Technology，Southwest University，Chongqing 400715，China

收稿日期：2023-05-10
基金项目：国家重点研发计划项目(2023YFB2905403)；重庆市教委科学技术研究重点项目(KJZD-K202100204)；重庆市自然科学基金项目(CSTB2023NSCQ-MSX0120).
作者简介：张思言，硕士研究生，主要从事图像处理研究.
通信作者：唐曦，博士，高级实验师.

Abstract：Video super-resolution technology has broad application prospects, but the algorithm complexity based on deep learning methods is too high to achieve real-time computation. In recent years, researchers have begun to explore super-resolution algorithm accelerators based on FPGA, in order to utilize the advantages of FPGA to improve algorithm performance and energy consumption, and achieve real-time video super-resolution. In this paper, an efficient high-speed bicubic linear interpolation VLSI architecture was designed based on FPGA, which can be used for 4× real-time video super-resolution. The FPGA architecture solved the problem of complex memory access mode required in the process of implementing bicubic interpolation, and proposed a hardware design of data rearrangement based on ping-pong operation, which rearranged the data in a specific order output by the algorithm into a row-major data arrangement, so that the hardware can directly connect to video interfaces such as HDMI. In addition, state machine, loop unrolling, pipeline and other methods were used to reduce design power consumption and timing violations, so that the entire hardware design can run at a higher frequency. The hardware architecture was implemented on Zynq-7020 FPGA, which enabled real-time oversampling of qHD (960×540) videos to UHD (3 840×2 160) high-definition videos. The experimental results show that the hardware design only needs to cache one row of image pixels with a latency of only 9.6 μs and a frame rate of 192.9 Hz, and successfully achieves real-time processing. The test results on the game image dataset show that the design has a peak signal-to-noise ratio of up to 35.67 dB and a structural similarity of 96.3%.

Key words：bicubic interpolation；real-time super-resolution；field programmable gate array (FPGA)；very large scale integration circuit (VLSI)

近年来，由于显示技术的发展，显示器的制造成本逐渐降低，支持 4K 超高清(UHD)分辨率的电视机在市场上成为主流. 然而，主流的视频源还是以高清(HD)和全高清(FHD)分辨率为主，因此，高质量的实时视频超分辨率技术对于 4K 影音系统的发展十分关键. 目前，超分辨率方法已被广泛研究，学者们提出了各种解决方案[1-8]，这为本文的研究工作提供了重要的理论基础. 然而，要做到实时计算，现有的冯·诺依曼架构计算机难以满足技术要求，因此对新硬件的设计需求迫在眉睫[9-13].

图像超分辨率重建是一个病态问题(ill-posed)，它需要从一个低维的 LR(Low Resolution)图像估计出一个高维的 HR(High Resolution)图像. 设低分辨率图像 y 是由高分辨率图像 x 通过一系列变换得到的：

$$y = \boldsymbol{DBW}x \tag{1}$$

式中：$\boldsymbol{D}$ 为亚采样矩阵，$\boldsymbol{B}$ 为光学模糊矩阵，$\boldsymbol{W}$ 为几何运动模糊矩阵. 显然，存在多个 x 的解可以得到同一个 y. 从信息论的角度而言，在亚采样的过程中，一部分信息已丢失，已经无法还原出 x. 因此图像超分辨率技术具有很大的研究与应用价值.

双三次插值作为一种经典的超分辨率算法，具有能够高质量重建图像的特点，但其计算复杂度较高. 对于 2 倍超分辨率问题，假设图像的高宽为 $m \times n$，那么时间复杂度为 $O(64mn)$. 因此，设计基于现场可编程逻辑门阵列(Field Programmable Gate Array，FPGA)的双三次插值高效硬件架构需要解决几个难题. 第一，需要设计复杂内存访问模式，以高效地获取数据，减少功耗. 低效的内存获取会导致运算的暂停，一般的方法是使用行缓存，然而使用过多的片上内存(On-chip Memory)又会对 FPGA 的性能带来更多要求，难以得到广泛应用. 第二，需要设计高效的处理单元(Processing Element，PE)和控制电路，以最大化设计建立时间的余量，使得系统能够高速运行. 第三，还需考虑量化对精度的损失，FPGA 中缺少对浮点数大量运算的支持，而且使用跳转表(Look-up Table，LUT)实现浮点数运算的过程十分复杂，需要大量资源，因此需要对模型进行量化. 第四，模型量化可以大幅度降低资源的使用量，比如 8 位的加法器比 10 位的加

法器少20%的资源使用量，因此需要考虑量化损失和资源使用量的平衡. 最后，需要设计PE输出数据的重排电路，这是因为PE的输出为包含多个行的图片块的形式，无法直接按行排列输出图像. 设计好重排电路后，即可简单地与常见的视频接口对接，直接将视频数据输出到显示器中.

Nuno等[14]将双三次插值算法分解为3个主要模块. 第1个模块生成插值系数，第2个模块执行双三次插值，第3个模块是控制单元. 因此，第2个模块对应算法的核心部分. 双三次插值公式在4个并行子模块中实现，每个子模块代表该方程的4行之一. 该设计在Virtex-Ⅱ Pro FPGA上实现，观察到的最大工作频率为100 MHz. 在这种情况下，需要32个乘法器和890个逻辑块(LBs)来支持算法的运行. Zhang等[15]将相邻像素之间的间隔分为8个子间隔. 每个子间隔的系数在离线计算后存储起来，以便每次进行插值时使用. 该方法的准确性取决于子间隔的数量. 然而，这种架构的缺点是插值质量较差且内存利用率高. 另外，作者没有提供该设计的硬件资源成本.

本文提出的双三次插值架构中，提供了一种有效的实时双三次插值硬件实现，以达到比现有实现更低的内存访问次数和运行速度，并提出了基于该双三次插值硬件实现的完整视频超分辨率硬件系统. 本文的贡献如下：① 提出了一种内存访问方案，称为移位寄存器反转块遍历(Shift Register Reverse Block Traversing，SRRBT)，用来高效获取图像数据，为PE提供连续的数据流；② 设计了高效的流水线PE，并使用状态机、流水线等优化方式提高设计的运行频率，降低资源的使用率；③ 提出了一种PE输出重排方法，称为同余缓存阵列(Modulo Buffer Matrix，MBM)，以连续按行排列输出像素点；④ 分析了量化位宽对算法精度的影响，并在不太影响精度的前提下降低量化位宽；⑤ 在Zynq-7020 FPGA上实现和验证了1个960×540到3 840×2 160的4倍实时超分辨率系统.

1 双三次插值

在数学上，双三次插值是对三次拉格朗日插值在二维平面上的扩展，其插值得到的表面比最近邻插值、双线性插值更加光滑，具有更高的图像质量，因此被广泛用于图像处理.

双三次插值通常通过在两个维度上卷积Sa函数的多项式近似来计算，Keys提出[16]的卷积所用核函数为：

$$W(x)=\begin{cases}(a+2)\,|\,x\,|^3-(a+3)\,|\,x\,|^2+1 & |\,x\,|\leqslant 1\\ a\,|\,x\,|^3-5a\,|\,x\,|^2+8a\,|\,x\,|-4a & 1<|\,x\,|<2\\ 0 & \text{其他}\end{cases} \tag{2}$$

式中：a为核函数系数. Keys指出，当$a=-0.5$时，关于采样间隔的收敛率可以达到三阶.

如果取$a=-0.5$，设$f(t)$为待插值的函数，且已知-1，0，1，2处的4个点f_{-1}，f_0，f_1，f_2，那么由$W(x)$得出的插值函数$p(t)$可以写为：

$$p(t;f_{-1},f_0,f_1,f_2)=1/2\begin{bmatrix}1 & t & t^2 & t^3\end{bmatrix}\begin{bmatrix}0 & 2 & 0 & 0\\ -1 & 0 & 1 & 0\\ 2 & -5 & 4 & -1\\ -1 & 3 & -3 & 1\end{bmatrix}\begin{bmatrix}f_{-1}\\ f_0\\ f_1\\ f_2\end{bmatrix} \tag{3}$$

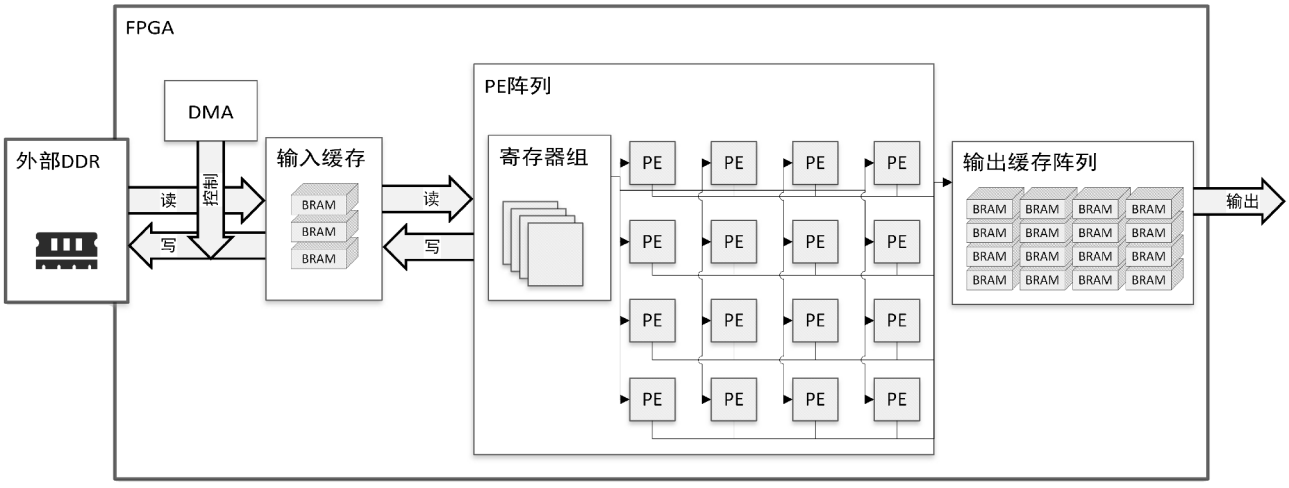
双三次插值使用1个点的邻域16个点对其进行插值，分x，y 2个方向使用三次插值进行计算. 每次三次插值时，HR图像点邻域的4个LR图像像素点坐标设为-1，0，1，2，那么在4倍缩放的情况下，HR图像点的坐标，也就是t的取值只有固定的4个值，即$t=\frac{1}{8}$，$\frac{3}{8}$，$\frac{5}{8}$，$\frac{7}{8}$，这样$p(t)$可以重新写成$p_i=p\left(\frac{2(i-1)+1}{8}\right)$的形式，$i=1$，2，3，4：

$$p_i = \boldsymbol{w}_i^T \boldsymbol{f} \tag{4}$$

式中：$\boldsymbol{w}_i$ 为 4 维权重向量，$\boldsymbol{f} = [f_{-1} \quad f_0 \quad f_1 \quad f_2]^T$. 这样，1 个方向的一次插值可以简化为四次乘累加(Multiply Accumulate，MAC).

2　双三次插值硬件设计

一个基于 FPGA 的加速器通常包含了 PE、输入缓存、输出缓存、控制单元等部分. 控制单元首先从外部 DDR(Dual Data Rate)内存通过 DMA(Direct Memory Access)读取数据到输入缓存中，然后 PE 直接读取输入缓存到内部运算寄存器中，其数据流可以用图 1 表示. 其中外部 DDR 内存访问成为问题，因为做一次插值运算需要准备好 16 个数据点，若按地址访问 DDR，需要 16 个时钟周期才能准备好数据. 一种方法是使用乒乓缓存分离数据读取和运算，然而这种方法会增加逻辑资源和片上内存的使用以及降低数据吞吐量，从而降低帧率.



图片数据预先储存在外部 DDR 中，然后通过 DMA 连接到 FPGA.

图 1　硬件设计数据流程图

双三次插值也是一种卷积，其输出尺寸大于输入尺寸，所以具有分数步长，这种卷积被称为分数步长卷积(Fractional Stride Convolution)[5]. 双三次插值也可以看作是一种反卷积[17-20]，因为尽管数学形式上不同，分数步长卷积和反卷积本质上是一样的. Zhang 等[21]提出的反卷积硬件结构设计方法论解决了反卷积硬件设计中存在的重叠求和问题，提出了反向循环(Reverse Looping)和步隙跳过(Stride Hole Skipping)方法. 本文参考其提出的方法论，采用遍历输出图的方式设计硬件.

将输出图像分割为多个 4×4 小块(Block)，每个小块上的 16 个像素都只依赖于 LR 图像中的 4×4 小块，这样就建立起 LR 图像小块和 HR 图像小块之间的一一对应关系.

假设 HR 图像像素的索引为 o_w、o_h，通道数索引为 o_c，图像的宽度、长度、通道数分别为 O_W、O_H、O_C，那么其所在 HR 小块对应的 LR 小块的左上角像素的索引 i_w、i_h 可以用以下公式描述：

$$i_w = \left\lfloor \frac{o_w - 2}{4} \right\rfloor - 1 \tag{5}$$

$$i_h = \left\lfloor \frac{o_h - 2}{4} \right\rfloor - 1 \tag{6}$$

插值所用的权值已经通过式(4)推出，为了计算权值，还需计算插值坐标 t，2 个方向上的坐标 t_x、t_y 和 o_w、o_h 的关系可以由以下公式推出：

$$t_x = \frac{(o_w - 2) - \left\lfloor \frac{o_w - 2}{4} \right\rfloor \times 4}{4} + \frac{1}{8} \tag{7}$$

$$t_y=\frac{(o_h-2)-\left\lfloor\frac{o_h-2}{4}\right\rfloor\times 4}{4}+\frac{1}{8} \tag{8}$$

这样，先在 x 方向进行4次插值，再在 y 方向上进行1次插值，总共5×4=20次MAC即可得出一个HR像素，20×16=320次MAC即可计算出一个HR图像的小块. 整个计算过程伪代码如下.

算法 双三次插值FPGA实现

1) for $o_h=0$ to O_H-1 do

2) for $o_w=0$ to O_W-1 do

3) for $o_c=0$ to O_C-1 do

4) $i_h\leftarrow\left\lfloor\frac{o_h-2}{4}\right\rfloor-1$

5) $i_w\leftarrow\left\lfloor\frac{o_w-2}{4}\right\rfloor-1$

6) $t_y\leftarrow\frac{(o_h-2)-\left\lfloor\frac{o_h-2}{4}\right\rfloor\times 4}{4}+\frac{1}{8}$

7) $t_x\leftarrow\frac{(o_w-2)-\left\lfloor\frac{o_w-2}{4}\right\rfloor\times 4}{4}+\frac{1}{8}$

8) $b_1\leftarrow p(t_x;\text{ in }[i_h][i_w][o_c],\text{ in }[i_h][i_w+1][o_c],\text{ in }[i_h][i_w+2][o_c],\text{ in }[i_h][i_w+3][o_c])$

9) $b_2\leftarrow p(t_x;\text{ in }[i_h+1][i_w][o_c],\text{ in }[i_h+1][i_w+1][o_c],\text{ in }[i_h+1][i_w+2][o_c],\text{ in }[i_h+1][i_w+3][o_c])$

10) $b_3\leftarrow p(t_x;\text{ in }[i_h+2][i_w][o_c],\text{ in }[i_h+2][i_w+1][o_c],\text{ in }[i_h+2][i_w+2][o_c],\text{ in }[i_h+2][i_w+3][o_c])$

11) $b_4\leftarrow p(t_x;\text{ in }[i_h+3][i_w][o_c],\text{ in }[i_h+3][i_w+1][o_c],\text{ in }[i_h+3][i_w+2][o_c],\text{ in }[i_h+3][i_w+3][o_c])$

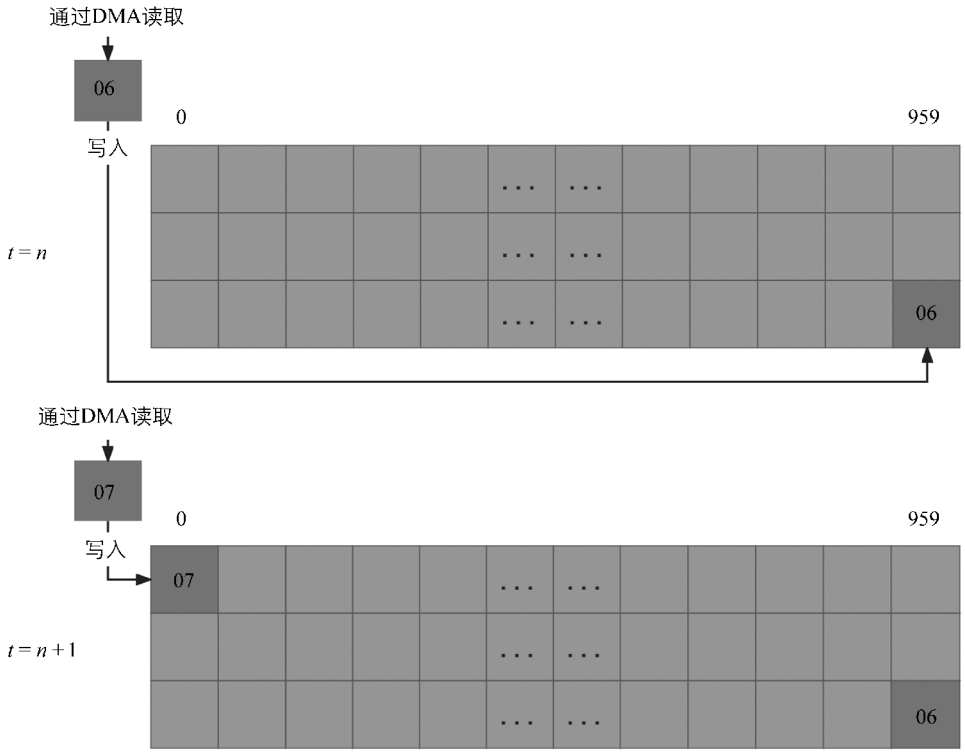
12) $\text{out }[o_h][o_w][o_c]\leftarrow p(t_y;b_1,b_2,b_3,b_4)$

2.1 卷积核遍历方式

不同一般反卷积算法，本文遍历输出图而不是输入图，这样输出图小块之间不存在重叠部分，这样就避免了额外的外部DDR内存操作. 同样，输出图小块的移动对应输入图小块移动一个像素. 另外对于边界区域的输出图小块，需要对输入图进行填充(Padding). 为此提出SRRBT方法，实现高效地从外部DDR获取图像数据以更新PE的内部运算寄存器.

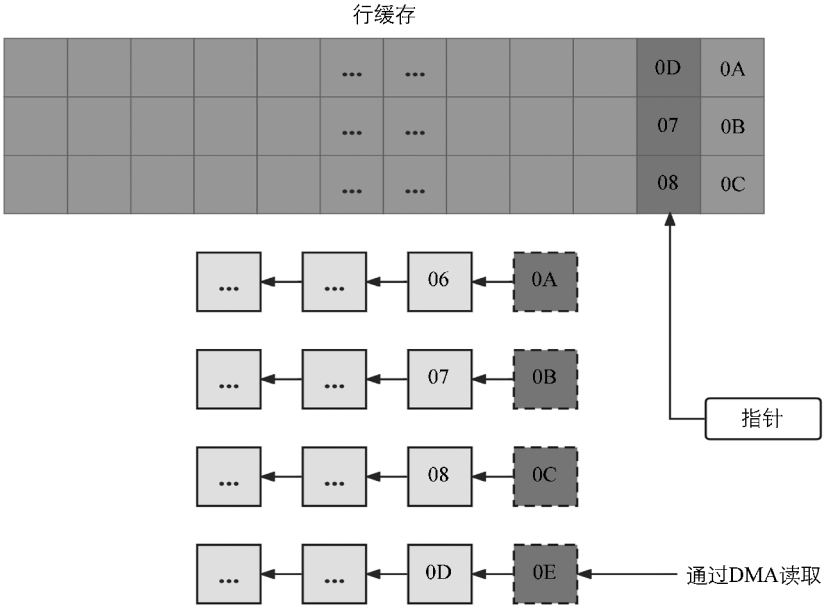
首先，使用BRAM(Block RAM)实现三块行缓存，从外部DDR读取的像素将被循环写入这三块行缓存中. 图2展示了行缓存的更新方式，当最后一块行缓存的最后一个像素被写完后，下一个像素将重新从第一块行缓存的第一个像素开始写. 然后，使用移位寄存器来提供输入图小块的前3列，这些值将从三块行缓存以及外部DDR的输入来更新，整个输入图小块的更新方式如图3所示. 这样，使用行缓存、移位寄存器、外部DDR提供的数据，就可以在每个时钟周期都更新输入图小块，实现右移一个像素的效果，将其写入PE的内部运算寄存器中.

对于边界处，需要填充宽度为2的像素，在卷积神经网络中，常使用0填充、镜像填充、复制填充等方式，对于图像超分辨率任务，复制填充较为合适.



最后一行的最后一个像素写完后，将开始重新写第一行第一个像素.

图 2　行缓存的更新方式示意图

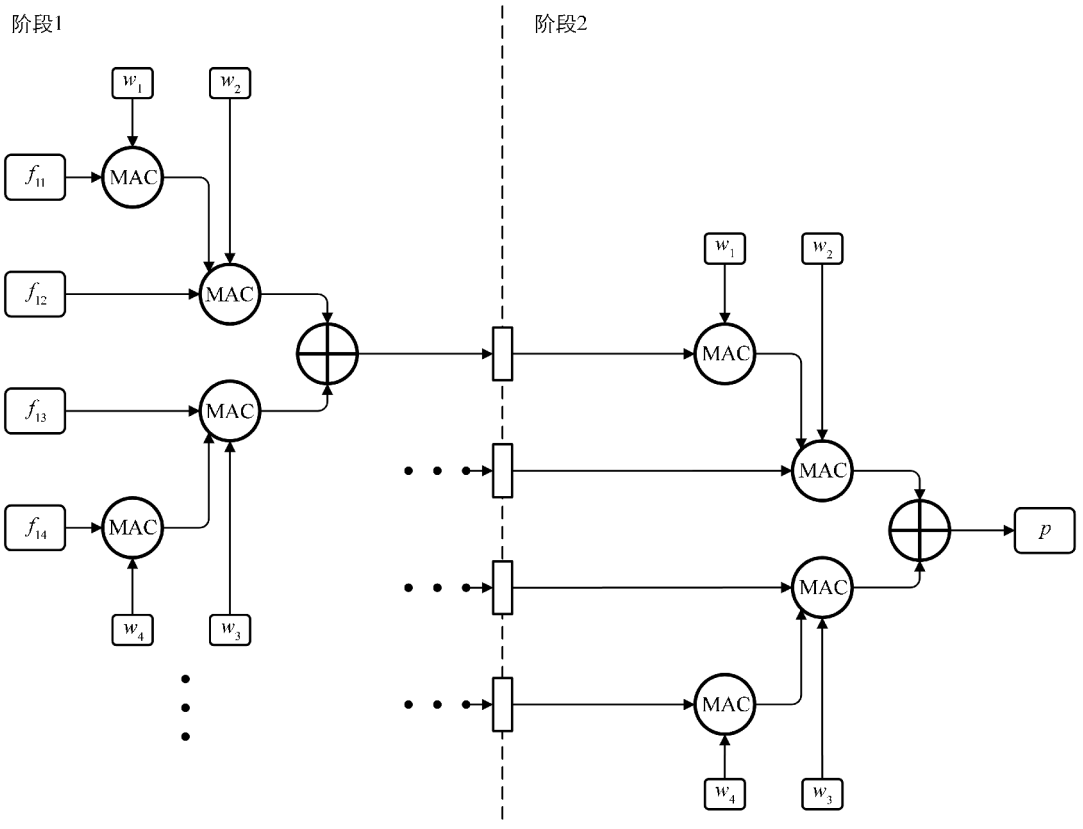


灰色方块表示触发器，虚线边框方块表示 BRAM 的输出以及通过 DMA 传输的数据.

图 3　输入图小块的更新方式示意图

2.2　流水线 PE 设计

按前文所述，计算一个像素需要 4 次 x 方向插值和 1 次 y 方向插值，且 y 方向的插值依赖于 x 方向的插值，如果设计在一拍内完成计算，那么数据的延迟是 2 次插值延迟的总和，极大地减少了建立时间余量. 本文设计了 2 级流水线 PE，将计算分为 x 方向插值和 y 方向插值 2 个阶段，以减少数据的延迟，提高运行频率. 插值运算由 4 次 MAC 完成，采用了金字塔的结构设计硬件完成 4 次 MAC 操作，以减少数据延迟，MAC 所用的权重为式(1)所推导的固定常值，设计结构如图 4 所示.



2个阶段之间的矩形表示触发器，f_{11} 到 f_{14} 表示LR图像小块的第一行像素，p 为输出，图中省略了第2行像素到第4行像素的部分.

图4 流水线PE设计流程图

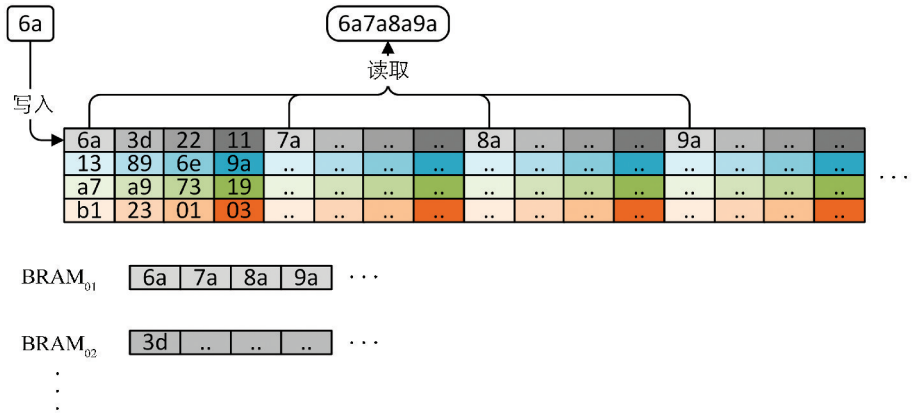
相较于现有双三次插值PE设计[14-15]，本文提出的设计所用的插值系数是固定的，并可以被配置. 由于不同位置的插值系数数值和位宽不同，其对应的PE电路和使用资源量也不同，因此相较于同质的PE设计，本文的设计能够大量节约资源使用量. 实验表明，本文PE中LUT使用量最大值为1 388，最小值为632，相较于同质的PE设计，可以节约32%的LUT使用量.

2.3 输出重排电路

对PE的输出进行重排，避免了将输出结果重新写回外部DDR的必要，使得硬件输出可以直接对接高清视频接口，比如VGA接口. 本文提出了同余缓存阵列的方式将一行像素存储在多个缓存中，实现数据重排，并通过乒乓操作缓存和输出像素.

下面详细介绍同余缓存阵列. 对于一行像素，使用4块BRAM缓存，分别为BRAM0、BRAM1、BRAM2、BRAM3，将坐标为 k 的像素存入BRAM $k \bmod 4$ 中，地址为 $\left\lfloor \frac{k}{4} \right\rfloor$，坐标互相同余的像素被存入同一块BRAM中，故称为同余缓存阵列. BRAM缓存配置为双端口RAM，可以同时实现1个像素的写入和4个像素的读取，那么对于一个行像素，可以实现在一个时钟周期内写入4个像素或者读取16个像素. 由于PE输出为4×4的矩阵，包含4行，所以使用16块BRAM：BRAM_{ij}，i，$j \in 0$，1，2，3，以组成同余缓存阵列. 在一个时间周期内，可以同时将PE输出的16个像素写入到同余缓存中，也可以同时读取一行中连续的16个像素值，实现了输入输出速率的匹配，从而达到了数据重排的效果. 图5详细描述了这一过程.

最后使用乒乓操作分离数据写入和读取，先写满一个同余缓存阵列，然后再读取这个同余缓存阵列，同时写另外一个同余缓存阵列，两个同余缓存阵列交替读写，保证数据可以连续输入和输出.



缓存阵列由 4×4 块 BRAM 组成，缓存 4 行像素，每块 BRAM 可以同时写入 1 个像素或者读取 4 个像素，上方的图为缓存像素在图像中的排列，下方的图为实际物理储存排列方式.

图 5　缓存阵列读写示意图

3　实验结果

在本节中分析设计的图像质量和运行速度. 将 PSNR(峰值信噪比)和 SSIM(结构相似性)这 2 种常用指标作为衡量图像质量的标准. 表 1 表明，本文硬件实现输出的图像质量基本可以与流行的双三次插值软件设计相媲美.

表 1　OpenCV 的 bicubic 实现与本文硬件实现在游戏测试图片中的性能比较

图片编号	OpenCV PSNR	OpenCV SSIM	本文实现的 PSNR	本文实现的 SSIM
0	31.27	0.856	31.21	0.856
1	29.50	0.846	29.41	0.845
2	28.84	0.855	28.76	0.855
3	32.06	0.853	31.98	0.853
4	33.31	0.928	33.15	0.928
5	32.44	0.921	32.20	0.920
6	34.33	0.938	34.26	0.938
7	33.96	0.924	33.80	0.923
8	34.61	0.940	34.43	0.939
9	32.64	0.897	32.52	0.896
10	32.10	0.883	31.97	0.882
11	32.38	0.898	32.27	0.896
12	34.17	0.938	33.99	0.936
13	35.32	0.962	34.98	0.960
14	34.91	0.936	34.68	0.934
15	31.50	0.882	31.41	0.880
16	29.65	0.845	29.54	0.843
17	35.94	0.940	35.62	0.936
18	36.20	0.936	36.00	0.935
19	34.00	0.913	33.81	0.912
平均	32.956 5	0.904 55	32.799 5	0.903 35

此外，本文还分析了量化对图像质量的影响. 为了不损失原图像的像素信息，PE的输入为9位有符号整数，中间结果也使用9位有符号整数表示，而权值则使用不同的量化位宽进行量化. 因此实验计算了不同权值量化位宽对资源使用量和图像质量的影响，如图6所示，使用PSNR和SSIM作为图像质量的衡量参数，LUT使用量和寄存器使用量作为资源使用量的衡量参数.

结果表明，LUT使用量和寄存器使用量基本随量化位宽的增加而线性增加，PSNR和SSIM也基本随量化位宽的增加而增加，而在8位时基本达到拐点，再增加位数对PSNR的贡献不大，因此，本文的硬件设计采用8位量化位宽对权值进行量化.

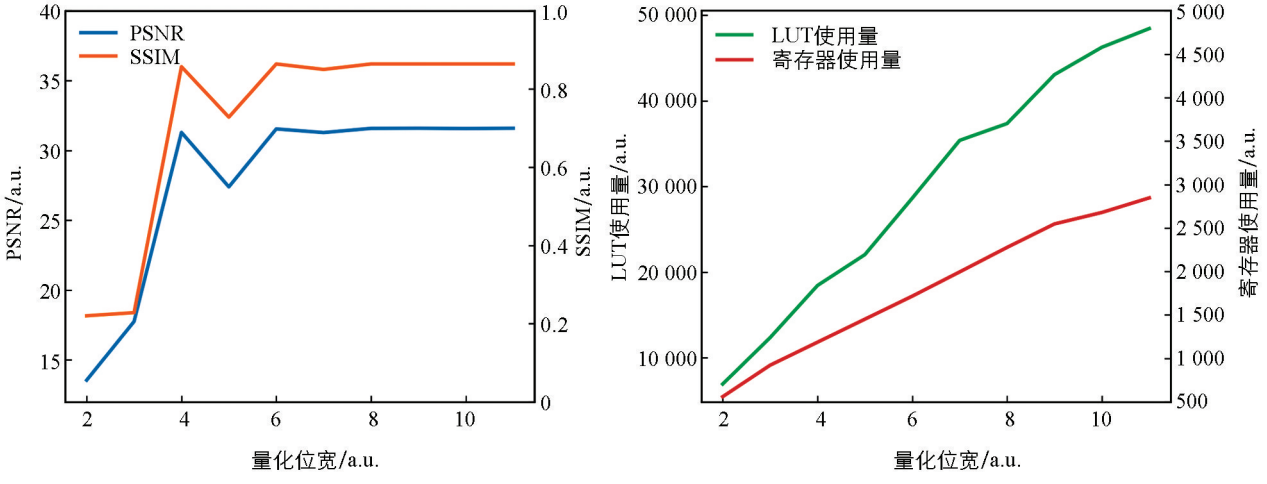


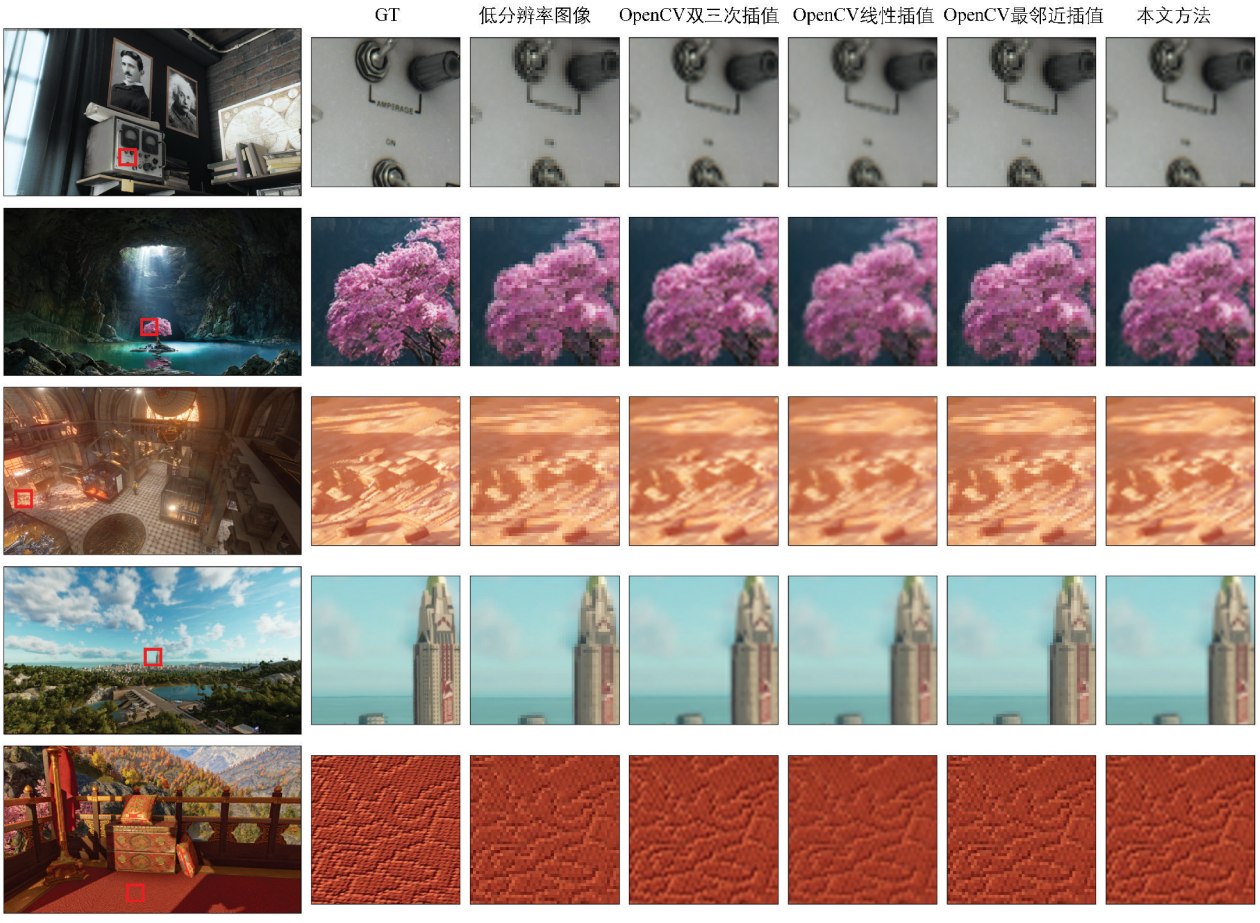
图6 量化对性能指标和硬件资源使用量的影响

本研究在Zynq-7020 FPGA上实现了硬件设计. 如图7所示，硬件实现可以达到和主流工程实现几乎一致的效果，可以重建较为精细的HR图像. 在运行速率方面，本文的设计运行在100 MHz工作频率的情况下可以实现192.9 Hz的帧率. 表2列举了本文的硬件实现、现有双三次插值实现以及酷睿i7-12700@4.90 GHz上软件实现的运行速率. 本文的硬件实现具有比当前最新处理器快几十倍的运行速率，这清楚地表明冯·诺依曼架构的处理器并不适用于实时超分辨率应用. 此外，与Nuno等人的硬件实现相比，本文的实现在更高分辨率输出下的帧率几乎提高了一倍. 考虑到帧率与输出图像大小成反比例，与乘法器的数量成正比，将Nuno等人的实现与本文的实现换算到相同输出图像大小和乘法器数量的情况下，帧率为56.43 Hz，但本文的实现要快将近4倍. 值得注意的是，尽管Zhang等人提出了只需20个乘法器的硬件架构，但作者并未提供对应硬件实现的详细信息. 结果表明，本文设计的实时超分辨率硬件可以满足高刷新率游戏的应用需求，能够在有限的硬件条件下极大地提高游戏帧率.

表2 运行速率比较

实现	工作频率	帧率	设备	放大倍数	输出图片大小	乘法器数量
软件	4.9 GHz	7.4 Hz	酷睿i7-12700	4	3 840×2 160	—
Nuno等[14]	100 MHz	95.24 Hz	Virtex-II Pro	任意	640×480	60
Zhang等[15]	—	—	—	—	—	20
本文的实现	100 MHz	192.9 Hz	Zynq-7020	4	3 840×2 160	960

注：“—”表示该项数值并未提供.



GT 为 Ground Truth.

图 7　4 倍上采样常见游戏影音图片的效果对比图

4　结论

本文提出了一种基于 FPGA 的高效高速双三次线性插值硬件架构设计用于实时视频超分辨率. 所提设计的主要优点在于其设计的高速性和简洁性，并易于集成到现成的影音系统中. 总之，本文平衡了硬件成本和质量，实现了一个低成本、低功耗、高质量、易集成的超分辨率重建硬件系统，用于从 qHD 到 UHD 的 4 倍实时超分辨率，可以满足 UHD 游戏影音体验等应用场景的需求.

参考文献：

[1] 张芳，赵东旭，肖志涛，等. 单幅图像超分辨率重建技术研究进展 [J]. 自动化学报，2022，48(11)：2634-2654.

[2] AGUSTSSON E，TIMOFTE R，VAN GOOL L. Anchored Regression Networks Applied to Age Estimation and Super Resolution [C] //2017 IEEE International Conference on Computer Vision (ICCV). Venice，Italy. IEEE，2017：1652-1661.

[3] CHEN M J，HUANG C H，LEE W L. A Fast Edge-Oriented Algorithm for Image Interpolation [J]. Image and Vision Computing，2005，23(9)：791-798.

[4] CHOI J S，KIM M. Super-Interpolation with Edge-Orientation-Based Mapping Kernels for Low Complex 2×Upscaling [J]. IEEE Transactions on Image Processing，2016，25(1)：469-483.

[5] DONG C，LOY C C，HE K M，et al. Learning a Deep Convolutional Network for Image Super-Resolution [C] //European Conference on Computer Vision. Cham：Springer，2014：184-199.

[6] TIMOFTE R，DE SMET V，VAN GOOL L. A+：Adjusted Anchored Neighborhood Regression for Fast Super-Reso-

lution [C] //Asian Conference on Computer Vision. Cham: Springer, 2015: 111-126.

[7] LI X, ORCHARD M T. New Edge Directed Interpolation [C] //Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101). Vancouver, BC, Canada. IEEE, 2002: 311-314.

[8] YANG C Y, YANG M H. Fast Direct Super-Resolution by Simple Functions [C] //2013 IEEE International Conference on Computer Vision. Sydney, NSW, Australia. IEEE, 2013: 561-568.

[9] CHANG J W, KANG K W, KANG S J. An Energy-Efficient FPGA-Based Deconvolutional Neural Networks Accelerator for Single Image Super-Resolution [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 30(1): 281-295.

[10] KIM Y, CHOI J S, KIM M. 2X Super-Resolution Hardware Using Edge-Orientation-Based Linear Mapping for Real-Time 4K UHD 60 Fps Video Applications [J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2018, 65(9): 1274-1278.

[11] LEE J, PARK I C. High-Performance Low-Area Video Up-Scaling Architecture for 4-K UHD Video [J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2017, 64(4): 437-441.

[12] SHIAU Y H, HUANG K Y, CHEN P Y, et al. A Low-Cost Hardware Design of Learning-Based One-Dimensional Interpolation for Real-Time Video Applications at the Edge [J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2021, 11(4): 677-689.

[13] SIVA M V, JAYAKUMAR E P. A Low Cost High Performance VLSI Architecture for Image Scaling in Multimedia Applications [C] //2020 7th International Conference on Signal Processing and Integrated Networks (SPIN). Noida, India. IEEE, 2020: 278-283.

[14] NUNO-MAGANDA M A, ARIAS-ESTRADA M O. Real-time FPGA-based Architecture for Bicubic Interpolation: An Application for Digital Image Scaling [C] //2005 International Conference on Reconfigurable Computing and FPGAs (ReConFig'05). Puebla, Mexico. IEEE, 2005: 8-11.

[15] ZHANG Y S, LI Y H, ZHEN J, et al. The Hardware Realization of the Bicubic Interpolation Enlargement Algorithm Based on FPGA [C] //2010 Third International Symposium on Information Processing. Qingdao, Shandong, China. IEEE, 2020: 277-281.

[16] KEYS R. Cubic Convolution Interpolation for Digital Image Processing [J]. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1981, 29(6): 1153-1160.

[17] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation [EB/OL]. (2014-10-22) [2024-02-22]. https://arxiv.org/pdf/1311.2524.pdf.

[18] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Image Net Classification with Deep Convolutional Neural Networks [J]. Communications of the ACM, 2017, 60(6): 84-90.

[19] SIMONYAN K, ZISSERMAN A. Very Deep Convolutional Networks for Large-Scale Image Recognition [EB/OL]. (2015-04-10) [2024-02-22]. https://arxiv.org/pdf/1409.1556v6.pdf.

[20] ZEILER M D, KRISHNAN D, TAYLOR G W, et al. Deconvolutional Networks [C] //2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Francisco, CA, USA. IEEE, 2010: 2528-2535.

[21] ZHANG X Y, DAS S, NEOPANE O, et al. A Design Methodology for Efficient Implementation of Deconvolutional Neural Networks on an FPGA [EB/OL]. (2017-05-07) [2022-10-09]. https://arxiv.org/abs/1705.02583v1.

责任编辑 汤振金

柳剑