

面向视频处理的粗粒度可重构单元设计

赵启林, 李 冰, 时美强, 熊 军
(东南大学 集成电路学院, 江苏 南京 210096)

摘 要: 针对视频处理算法, 设计了一种面向视频处理的粗粒度可重构处理单元. 它可以执行 8 位数据的加法、减法、乘法、乘加和求两数差的绝对值等操作, 可以有效地支持高计算密度的视频处理算法. 可重构处理单元使用 Verilog 设计, 采用 CMOS 0.18 μm 工艺 DC 综合, 面积为 97 913 μm^2 , 关键路径为 4.51ns, 总的动态功耗为 4.2mW. 完成一次 8 $\times$ 8 像素块的 2D-DCT 算法和全搜索块匹配 MAD 算法分别需要 10 和 15 个时钟周期.

关键词: 可重构处理单元; 视频压缩编码; 并行计算

中图分类号: TN402

文献标识码: A

文章编号: 1000-7180(2010)05-0082-05

Design of a Coarse-grain Reconfigurable Cell for Video Processing

ZHAO Qi-lin, LI Bing, SHI Mei-qiang, XIONG Jun

(Institute of Integrated Circuit, Southeast University, Nanjing 210096, China)

Abstract: This paper presents VORC (Video Oriented Reconfigurable Cell), a coarse-grain reconfigurable cell optimized for video processing. The cell has been designed to provide a dense support for video processing 8-bit arithmetic operations such as addition, subtraction, multiplication, multiply-accumulation and absolute value of the difference. The cell has been described with verilog and synthesized under 0.18 μm COMS proces with Synopsys Design Compiler. The area is 97 913 μm^2 and the critical path is 4.51nm and the total dynamic power is 4.2mW. The entire 8 $\times$ 8 2D-DCT and FSBM MAD respectively require 10 and 15 cycles.

Key words: reconfigurable processing cell; video compression; parallel computation

1 引言

现代多媒体应用, 包括图像处理、音视频处理等, 都需要很高的计算性能, 而且要具有快速匹配相应算法的能力. 高计算性能和高灵活性的要求使得可重构计算技术成为令人关注的解决方案^[1-3]. 事实上, 可重构电路结构既具有类似于 ASIC 的高性能有具有可编程的灵活性, 已经成为未来多媒体处理芯片的主要结构之一. 在这种结构中, 可重构处理单元的功能和运算速度决定了整个可重构系统的运算处理能力.

在视频处理算法分析的基础上, 设计了一个数据位宽为 8 位的粗粒度可重构处理单元, 它可以实现 8 位数据的加法、减法、乘法、乘加和求两数的绝

对值等操作. 该结构具有很高的扩展性, 将多个可重构处理单元组成阵列, 可以实现视频处理算法的并行执行, 提高处理速度.

2 电路设计

2.1 电路结构

面向视频处理的 8 位可重构处理单元的结构^[4-5]如图 1 所示. 它由 2 个 8 位 $\times$ 4 位的乘法器^[6-7]、7 个 4 位加法器、14 个选择器、6 个寄存器、1 个比较器、1 个触发器和 1 个与门组成. 这种简单的结构在选择器的选择端信号的控制下, 可以实现多种 8 位数据操作, 包括两个 8 位数据的加法、减法、乘法、相乘后累加和求两个数差的绝对值. 表 1 为操作码所对应的操作类型以及相应的执行周期.

表1 操作类型和操作码的对应关系

操作类型	操作码 (S[9:1], Cin1, abs_en)	执行周期 (周期数)
加法: $A+B$	010111010, 0, 0	1
减法: $A-B$	010101001, 1, 0	1
乘法: $A \times B$	010011111, 0, 0	1
乘加: $A \times B + C \times D$	101011100, 0, 0	2
绝对值: $ A-B $	010101001, 1, 1	1 或 2

2.2 算术操作

执行8位加法操作时,需要一个时钟周期.首先,通过乘法器1和乘法器2分别计算 $A[7:0] \times 0001$ 和 $00000001 \times B[7:4]$;然后,通过加法器1和加法器2分别计算 $A[3:0] + B[3:0] + \text{Cin1}$ 和 $A[7:4] + B[7:4] + \text{CO1}$;最后经过加法器5计算 $A[7:4] + B[7:4] + \text{CO1} + \text{Cin2}$ 得出最终结果,暂存在寄存器3和寄存器4中,通过 $R[7:0]$ 输出.

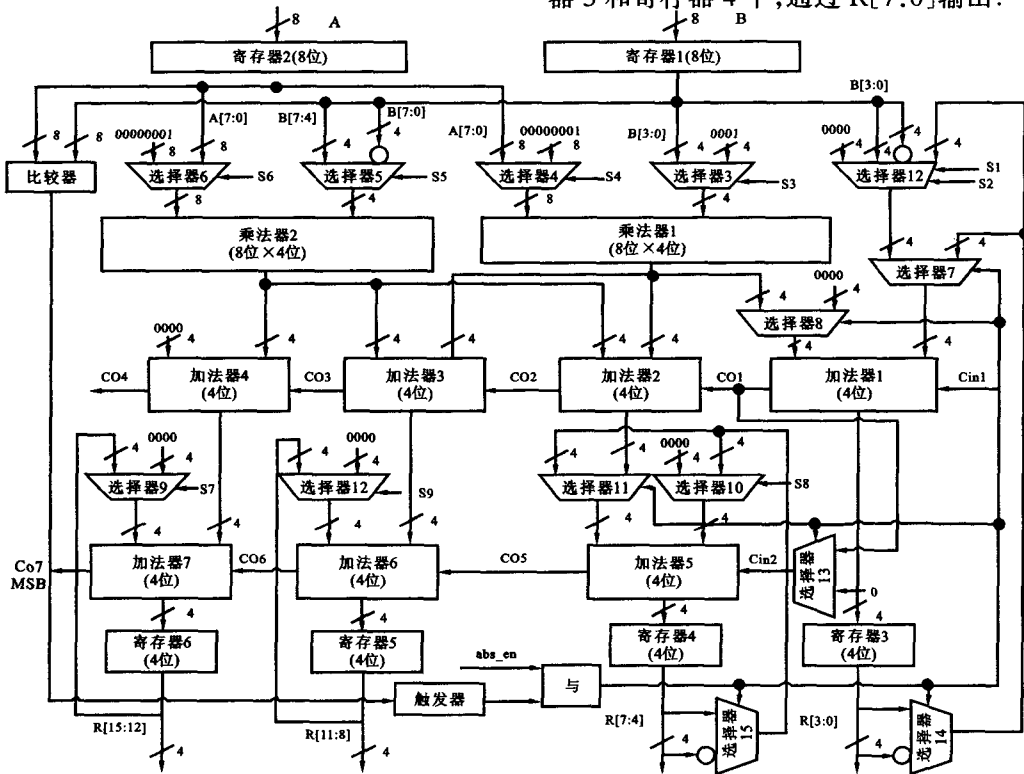


图1 可重构处理单元的结构

8位减法运算的执行过程与8位加法操作类似,只是操作数 $B[7:0]$ 作为减数,要使用二进制补码表示,并且将 Cin1 信号设置为1,然后通过加法来实现减法运算,其差值通过 $R[7:0]$ 输出.

8位 $\times$ 8位的乘法运算,只需要一个时钟周期即可完成.乘法器1和乘法器2分别计算 $A[7:0] \times B[3:0]$ 和 $A[7:0] \times B[7:4]$,然后将2个12位的部分积以每四位为一组分分成高中低三组,乘法器1产生的部分积的高中低三组值分别送入加法器3、加法器2和加法器1,同样,将乘法器2产生的积的高中低三组值分别送入加法器4、加法器3和加法器2.最终的乘积经过加法器5,6,7暂存到寄存器3,4,5,6中.通过输出端 $R[15:0]$ 输出.

为了实现相乘后累加操作,增加了加法器5,6,7和5个选择器,以实现 $A \times B + C \times D$ 运算.这一改进只是在8位乘法电路的基础上,增加了少量的电

路,就完成了相乘后累加操作,既节省了电路面积又提高了电路的灵活性,并且可以有效地支持视频处理算法中的DCT算法(3.1详细介绍).该操作的执行过程分为两步:第一,首先计算 $A \times B$,并将乘积保存在寄存器中,需要1个时钟周期;第二,通过4个选择器21,10,12,9将保存在寄存器中 $A \times B$ 的想对应的值送入加法器1,5,6,7,完成 $A \times B + C \times D$,这一步也是只需要1个时钟周期.因此,完成两次乘法和一次加法只用到2个时钟周期,这可以提高数据的处理速度.

为了支持视频处理算法中的运动估计算法(3.2节中详细介绍),扩展可重构处理单元的函数功能,增加了一种新的操作:求两数的差的绝对值.此时,需要增加一个使能控制信号 abs_en ,并在在原电路基础上增加一个比较器、1个触发器、一个与门和2个选择器.当使能信号 abs_en 为1时,电路执行 $|A$

$-B|$ 操作;首先使用比较器来确定差值的符号 MSB,如果 MSB 为 1,说明差值为负值,寄存器中的数据以二进制补码表示,需要对其进行取反加 1 操作,使其变为正值,此过程需要两个时钟周期.如果 MSB 为 0,说明差值为正值, $|A-B|$ 操作只需要一个时钟周期.最终的绝对值通过 $R[7:0]$ 输出.

从上面后两种运算功能的扩展,可以看出:使用可重构技术,只需要在原始电路的基础上,增加少量的电路就可以实现更加复杂的函数功能,并且节省运算时间,做到以较少的代价获得较高的性能.

3 算法分析和映射

3.1 DCT 算法的分析和映射

DCT 变换是一种基础的数学变换,在图形图像、音频和视频等信号处理领域中有着广泛的应用.静态图像和动态图像(视频)压缩国际标准 JPEG、JPEG2000、MPEG-2、MPEG-4 和 H. 264 等均用到 DCT 算法,并且不断有新的改进的 DCT 算法出现,如 H. 264 中使用整数 DCT. DCT 算法的运算量很大,使用基于指令的处理器(CPU 和 DSP)实现时,需要对每一次数据操作进行取指令、译码、取数据、执行操作和存储结果五个步骤,这会消耗大量的时间,无法满足现在人们对高清晰多媒体实时处理的要求.采用基于数据流架构的可重构处理器可以很好的解决这个问题^[2].

设源图像矩阵为 $x(i, j)$, 变换后的矩阵为 $X(u, v)$, $0 \leq i, j, u, v \leq 7$, 那么 $x(i, j)$ 的二维 DCT 变换的数学定义为

$$X(u, v) = \frac{1}{4} C(u) C(v) \sum_{x=0}^7 \sum_{y=0}^7 \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right) x(i, j) \quad (1)$$

$$\text{式中, } C(u)C(v) = \begin{cases} \frac{1}{\sqrt{2}}, & u, v = 0 \\ 1, & \text{其他} \end{cases}$$

从式(1)中可以看出:完成一个像素的运算总共需要 64 次实数乘法 and 64 次实数加法,完成一个 8×8 个像素大小的块的 DCT 需要 $64 \times 64 = 4096$ 次实数乘法和实数加法.一幅 CIF 格式的图像的分辨率为 352×288 , 包含 44×36 个 8×8 的 DCT 像素块.因此,完成整幅图像中数据块的 DCT 变换总共需要 $44 \times 36 \times 64 \times 64 = 6488064$ 次实数乘法和实数加法.这种变化的计算量很大,而且乘法会比加法消耗更多的运算时间.

二维 DCT 具有可分解性,即可以将一个二维的

DCT 运算分解成 2 个一维的 DCT 来计算^[8].目前的快速 DCT 算法先按行进行 8 次 8 点的一维 DCT,在按列进行 8 次 8 点的一维 DCT.8 点的一维 DCT 需要 64 次乘法和 64 次加法.一个 8×8 大小的图像块共需要做 $8 + 8 = 16$ 次一维 DCT,总共需要 1024 次实数乘法和加法.8 点的一维 DCT 运算的数学定义为

$$X(k) = C_k \sum_{n=0}^7 x(n) \cos \frac{(2n+1)k\pi}{16} \quad 0 \leq k \leq 7 \quad (2)$$

$$C_k = \begin{cases} \sqrt{\frac{1}{N}}, & K = 0 \\ \sqrt{\frac{2}{N}}, & \text{其他} \end{cases}$$

从式(2)可以看出,8 点的一维 DCT 运算就是一个长度为 8 的两个矢量的点积运算,也就是先相乘然后在累加,此算法在使用可重构单元处理时,完成一次乘加操作平均需要一个周期,减少了运算时间.

设 $x = (x[0], x[1], \dots, x[n])$ 和 $y = (y[0], y[1], \dots, y[n])$ 两个矢量的点积运算定义如下:

$$p = x \cdot y = \sum_{i=0}^n (x[n] \times y[n]) \quad (3)$$

图 2 为两个矢量点积运算的示意图,其中第一排的符号表示将两数相乘,第二排的符号表示将两数相加,最终的结果由 P 端口输出.由此,可以看出矢量点积的计算过程可以分为乘法和累加两个阶段,因此可以使用可重构处理单元中的相乘后累加操作来实现 2 维的 DCT 算法.

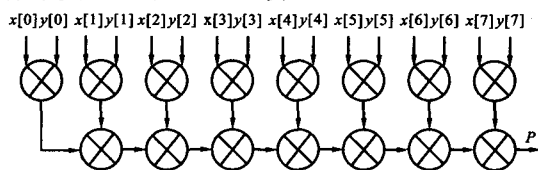


图 2 长度为 8 的两个矢量点积运算示意图

常用的视频处理对象是 8×8 的像素块,因此,使用 8×8 个可重构处理单元,这样可以直接将每一个像素点映射到相对应的一个可重构处理单元上^[1,4].2 维 DCT 算法到可重构单元的映射过程可分为三个过程:第一,通过数据总线,将 8×8 个像素值从帧缓冲器装载到 8×8 个可重构处理单元中,DCT 的变换矩阵系数可由立即数提供,此并行装载过程只需要 1 个时钟周期;第二,利用 2 维 DCT 的可分解性,首先使用可重构处理单元阵列中的行方向的单元,并行计算 8 点的 1 维 DCT,然后按列并

行计算 8 点的 1 维 DCT,共需要 $2 \times 8 = 16$ 个时钟周期;第三,将变换后的矩阵值存储到帧缓冲器中,需要 1 个时钟周期.因此,执行 1 次 8×8 的像素块的 2 维 DCT 运算需要 $1 + 2 \times 8 + 1 = 18$ 个时钟周期.

3.2 运动估计算法的分析和映射

运动估计是一项利用相邻图像帧之间的相关性,用前一帧的图像来预测下一帧的图像内容的技术,它使用参考帧中与待预测帧匹配最佳的图像块的运动矢量来表示待预测帧中的 $N \times N$ 个像素点组成的块在参考帧中的位置.该技术已经被 MPEG-1、MPEG-2、MPEG-4 和 H.261、H.263、H.264 等国际视频编码标准采用.目前,运动估计在多媒体应用如视频电话、高清晰度电视和计算机视觉等方面得到广泛的应用.相应地出现了多种新的运动估计和运动补偿的算法,在视频压缩技术中应用最多的是块匹配算法.

块匹配算法是从前一帧图像 F_{n-1} 的一定范围里搜索出与当前帧 F_n 的一个 $N \times N$ 像素块最接近的对应图像块的偏移,使用这一偏移量进行压缩编码.完全搜索块匹配算法(Full Search Block Mapping,FSBM)是指将当前块的像素值与搜索区域内所有候选块的像素值一一对应进行比较,然后找出其中偏移量最小的块的运动矢量.这种运算量相当巨大,占用的资源多,消耗的时间长,其复杂度占到整个视频编码系统的 80% 以上.用现有的基于指令集的处理单元完成这些运算将花费很多时间,采用基于数据流的可重构处理单元进行处理是一个很好的解决方案^[1].目前,在实际应用中,如 H.264 视频编码标准,都采用平均绝对之差(MAD)最小作为判别运动矢量的标准.

$$MAD(i, j) = \sum_k \sum_l |F_n(k, l) - F_{n-1}(k + i, l + j)| \quad (4)$$

式中, $F_n(k, l)$ 是当前帧 F_n 位于位置 (k, l) 的像素值, $F_{n-1}(k + i, l + j)$ 是前一帧 F_{n-1} 位于 $(k + i, l + j)$ 位置的像素值, i, j 都是搜索范围.在这些 MAD 值中找出最小的一个,其 (i, j) 就代表了最匹配的位移向量.

运动估计算法到可重构单元阵列的映射过程与 DCT 算法的映射过程相似,仍以 8×8 的像素块位处理对象,并将其映射到相对应的可重构处理单元上.整个映射过程可分为三步:第一,分别将待预测帧和参考帧的 8×8 的像素块中的 64 个像素值装载

到 64 个可重构处理单元中,此过程使用并行处理方式,需要 1 个时钟周期;第二,使用可重构处理单元阵列并行计算两数差的绝对值,最差情况下(差为负值时)需要 2 个时钟周期;第三,将 64 个绝对值并行存储到存储单元,需要 1 个时钟周期;第四,计算 64 个绝对值的累加和,首先将 64 个绝对值分为 32 组,每组 2 个,然后送入 32 个可重构处理单元,进行加法操作,得到 32 个和值,并将 32 个和值存储到存储器中,然后重复执行上述过程,直到计算得出 64 个绝对值的累加和,此过程需要 $3 \times 6 = 18$ 个周期.因此,一个 8×8 像素块的运动估计 FSBM 算法执行 1 次需要 $1 + 2 + 18 = 21$ 个时钟周期.

4 功能仿真和性能分析

4.1 可重构处理单元的功能仿真

使用 Verilog HDL 完成可重构处理单元的 RTL 编码,并使用 ModelSim 对其功能进行仿真,5 种不同的运算操作的功能仿真波形如图 3 所示,成功实现该单元的功能的可重构性.



图3 功能仿真波形

4.2 可重构处理单元的综合结果

使用 Synopsys 的综合工具 Design Compiler 在 $0.18\mu\text{m}$ CMOS 工艺上对该可重构处理单元进行综合.综合结果如表 2 所示,一个可重构单元占用的面积为 $97\,913\mu\text{m}^2$,其关键路径时间为 4.51ns,动态功耗为 4.2mW.

表2 可重构单元的综合结果

	面积/ μm^2	关键路径/ns	动态功耗/mW
VORC	97 913	4.51	4.2

5 结束语

文中提出的面向视频处理的可重构单元,通过电路复用技术,可以在只增加少量电路的基础上,实现更为复杂的功能,以较低的代价获得较高的性能.该单元可以实现 8 位数据的加法、减法、乘法、乘加和求两数差的绝对值共 5 种运算,不仅具有较高的灵活性和可扩展性,而且关键路径为 4.51nm 具有类似于 ASIC 的性能,可以有效支持视频处理算法.将可重构处理单元组成阵列结构,可以实现视频处理算法的并行执行,提高运算速度.执行一个 8×8 像素块的 2 维 DCT 运算和块匹配算法的 MAD 运算分别只需要 18 个时钟周期和 21 个时钟周期.

参考文献:

- [1] 肖钰,刘雷波,魏少军. 应用于视频处理的可重构流处理器的设计与实现[J]. 电视技术,2008,32(6):19-24.
- [2] Singh H, Lee Minghau, Lu Guangming, et al. MorphoSys: an integrated reconfigurable system for data-parallel and computation-intensive applications[J]. IEEE Trans. Computers, 2000,49(5): 465-481.
- [3] Baumgarte V, Ehlers G, May F, et al. PACT XPP - A self-reconfigurable data processing architecture[J]. The journal of supercomputing, Springer-Verlag, 2003,126(2): 167-184.
- [4] Marco Lanuzza, Martin Margala, Pasquale Corsonello.

Cost-effective low-power processor-in-memory-based reconfigurable datapath for multimedia applications [C]// Proc. of int.l Symp. on low power electronics and design (ISLPED). San Diego, California, USA, 2005.

- [5] Lanuzza M, Perri S, Corsonello P, et al. A new reconfigurable coarse-grain architecture for multimedia applications [J]. Second NASA/ESA Conference on Adaptive Hardware and Systems, 2007(6):384.
- [6] 王卫涛,沈绪榜,单超. 32 位可重构多功能乘法器的设计与实现[J]. 微电子学与计算机,2008,25(8):199-201.
- [7] 李磊,赵建明. 高速可重组 16×16 乘法器的设计[J]. 微电子学与计算机,2007,24(6):120-122.
- [8] Gong D, He Y, Cao Z. New cost-effective VLSI implementation of a 2D discrete cosine transform and its inverse [J]. IEEE Trans. on circuits and systems for video technology, 2004, 14(4): 405-415.

作者简介:

赵启林 男,(1981-),硕士研究生.研究方向为可重构多媒体芯片设计.

李冰 男,副教授.研究方向为现场集成系统与信息专用集成电路设计(包括可重构进化硬件电路、通讯纠错码、网络安全、嵌入式系统、微控制器设计以及接口 IC 设计等).

时美强 男,(1984-),硕士研究生.研究方向为音视频编解码及其 VLSI 设计.

熊军 男,(1984-),硕士研究生.研究方向为视频编解码技术及其 ASIC 设计.

(上接第 81 页)

- [12] 孙文生,刘泽民.组播路由调度的神经网络方法[J]. 通讯学报,1998(11):1-6.
- [13] 陈萍,郭金锋. 对 Hopfield 神经网络求解 TSP 的研究[J]. 北京邮电大学学报,1999(2):58-61

作者简介:

方向 女,(1978-),硕士,讲师.研究方向为智能计算、网络体系结构与性能评价.

冯烟利 男,(1962-),教授.研究方向为网络体系结构与性能评价.