

基于 SOPC 的 JPEG2000 编码器设计

作者：王刚毅 胡小开 指导教师：任广辉

(哈尔滨工业大学 图像与信息技术研究所, 哈尔滨 150001)

摘要: 为了满足网络及多媒体领域的应用, JPEG2000 的硬件实现具有重要意义。本文提出了一种改进的行式二维小波变换器结构, 设计了位平面并行的位平面编码器和四级流水线结构的算术编码器, 并将其整合于一个 SOPC 中, 实现了 JPEG2000 编码系统。整个设计通过 Altera 公司 Stratix II 系列的 EP2S60F1020C5 平台验证, 在最高时钟频率 98MHz 下能达到编码分辨率 512×512 灰度图像 52frame/s 的速度, 满足了实时编码的要求。

关键字: JPEG2000, SOPC, 小波变换, 位平面编码, 算术编码

Design of JPEG2000 Encoder based on SOPC

Author: Wang Gangyi Hu Xiaokai Teacher: Ren Guanghui

(Institute of Image and Information Technology, Harbin Institute of Technology,
Harbin 150001)

Abstract: In order to meet the application of network and multimedia, the hardware implementation of JPEG2000 is of great significance. This paper proposes an improved architecture of two-dimensional wavelet transformer, designs a parallel bit-plane encoder and a four-pipeline arithmetic encoder, then integrates the three modules into a SOPC. The JPEG2000 coding system is tested on the EP2S60F1020C5, at the highest clock frequency of 98MHz, it can process 52 gray images with the resolution of 512×512 in a second, meet the requirement of real-time encoding.

Key Words: JPEG2000, SOPC, wavelet transform, bit-plane encoder, arithmetic encoder

1 引言

因特网的发展带来了数据压缩方面的一场革命。随着网络的普及与多媒体概念的深入人心, 广大设计师对图象质量、文档尺寸、图像读取速度的要求越来越高, 图象压缩已经成为数据压缩的一个核心组成部分。因此在图像的压缩舞台上, 如 GIF、JPEG、PNP 各种图象压缩格式应运而生。JPEG 是其中应用最广泛、最成功的一种压缩标准。

但是, 随着多媒体应用领域的激增, JPEG 标准已渐渐无法满足人们的需求。其主要问题是在低比特率编码时块状编码失真明显, 这是因为 JPEG 标准中采用的离散余弦变换(DCT)将图像划分成 8×8 的方块域, 各块内产生的编码失真在块边缘形成了不连续状况。因此, 自 1997 年起, JPEG 组织开始了静止图像新压

缩标准 JPEG2000 的开发工作，并于 2000 年完成了其核心编码系统的标准制定。JPEG2000 使用了离散小波变换(DWT)作为其核心变换算法，优化截断的嵌入式块编码(EBCOT)作为核心编码法。JPEG2000 不仅提供了超越 JPEG 的压缩性能，更提供了一系列丰富的功能，以满足对图像编码技术日益增长的需求。

JPEG2000 在提供出色性能及众多功能的同时，也不可避免地增加了实现复杂度。欲实现相同的压缩质量，JPEG2000 的运算量通常比 JPEG 大数倍。在嵌入式实时应用领域，如此大的计算量用 DSP 等纯软件方案实现比较吃力，且很难达到理想的效果，而如果使用纯硬件的 ASIC 方案，则在灵活性上受到限制。因此，拥有高性能及高灵活性的 FPGA+SOPC 平台是 JPEG2000 最适宜的解决方案。

2 JPEG2000 编码过程分析

JPEG2000 编码过程如图 2-1 所示，输入图像首先进入预处理模块进行色彩分量变换及分片；而后以图像分片为独立单元进入小波变换器，经小波分解后得到不同分辨率的小波子带系数；然后分别对变换后的小波系数选取适当的量化步长进行量化；将量化后的系数划分成小的数据单元，称为码块，接着对每个码块进行位平面编码和算术编码，最后按照标准规定的语法打包生成压缩码流。其中量化和 EBCOT 两个模块在码流控制模块的控制下工作，以得到规定大小或质量的输出码流。

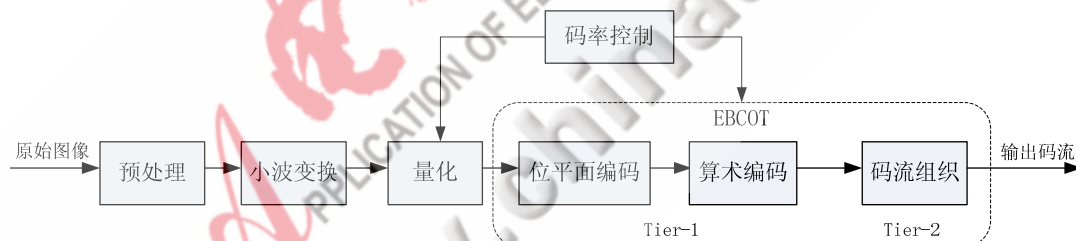


图 2-1 JPEG2000 编码过程

3 关键模块设计

3.1 小波变换

JPEG2000 标准采用二维离散小波变换对输入的图像片进行分解。二维小波变换的过程可等效为分别对水平方向和垂直方向进行一维小波变换。一次分解的结果为四个子带分量，分别为：水平方向低频、垂直方向低频的 LL 分量；水平方向高频、垂直方向低频的 HL 分量；水平方向低频、垂直方向高频的 LH 分量；水平方向高频、垂直方向高频的 HH 分量。不断对每次分解结果中的 LL 分量再进行二维离散小波变换，即可实现多阶变换。

3.1.1 一维小波变换器的设计

在 JPEG2000 标准推荐了 5/3 和 9/7 两种小波变换，前者为整数变换，主要用于无损压缩；后者为实数变换，只能用于有损压缩，但对低频和低频分量具有更好的分解效果，故本设计使用了 9/7 小波变换。

设输入序列为 $x(n)$ ，输出序列为 $y(n)$ ，其中 $y(2n)$ 为低频分量输出， $y(2n+1)$ 为高频分量输出，9/7 小波变换的卷积表达式如式(3-1)、(3-2)所示。其中 $h(i)$ 、 $g(i)$ 为常数。

$$y(2n) = \sum_{i=-4}^4 h(i)x(2n+i) \quad (3-1)$$

$$y(2n+1) = \sum_{i=-3}^3 g(i)x(2n+1-i) \quad (3-2)$$

如采用提升算法，则多相位分解阵的分解式为式(3-3)。

$$\tilde{P}(z) = \begin{bmatrix} 1 & \alpha(1+1/z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \beta(1+z) & 1 \end{bmatrix} \begin{bmatrix} 1 & \gamma(1+1/z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ \delta(1+z) & 1 \end{bmatrix} \begin{bmatrix} \zeta & 0 \\ 0 & 1/\zeta \end{bmatrix} \quad (3-3)$$

其中 $\alpha=-1.586134342$ ， $\beta=-0.0529801185$ ， $\gamma=0.882911076$ ， $\delta=-0.443506852$ ， $\zeta=1.2301741049$ 。图 3-1 为提升算法的结构框图。

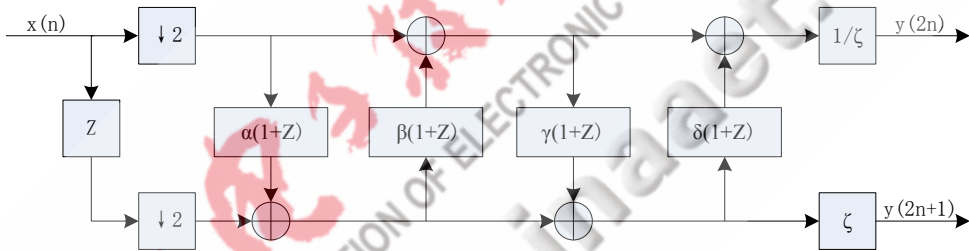


图 3-1 一维提升结构框图

其中， $\downarrow 2$ 表示基 2 抽取， Z 表示延迟一个数据。图 3-2 为根据图 3-1 得到的流水线图。

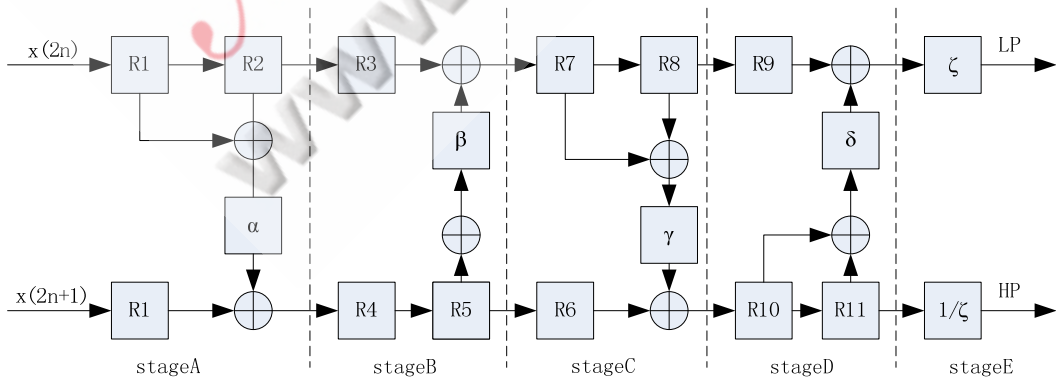


图 3-2 一维小波变换的流水线图

该流水线结构分为 Stage A 至 Stage E 共五级，其中 Stage A、Stage C 为偶数点对奇数点的预测，Stage B、Stage D 为奇数点对偶数点的更新，Stage E 对小波系数的增益进行调整并输出低通分量 LP 和高通分量 HP。数据在 Stage A 至

Stage D 中每级停留 2 个时钟周期，在 Stage E 中停留 1 个时钟周期，故数据在该结构中的总延时为 9 个时钟周期，其间完成 6 次乘法和 8 次加法。

为验证设计的正确性，使用 ModelSim 对设计进行了验证。如图 3-3 所示，为功能仿真波形图。

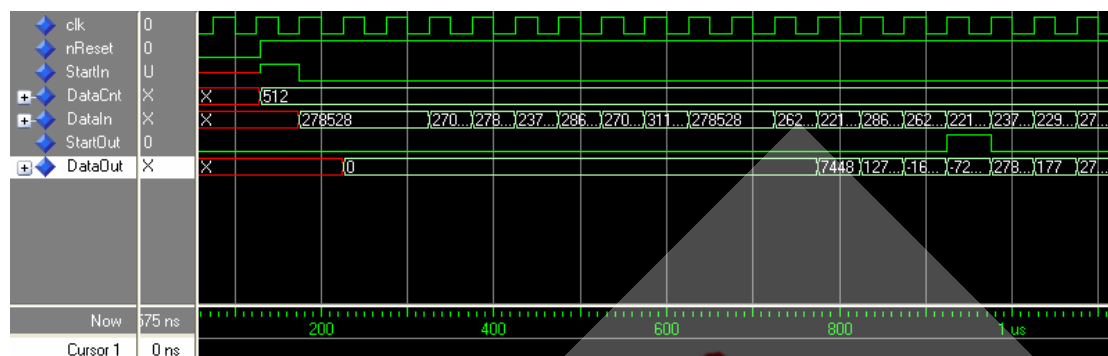


图 3-3 一维小波变换功能仿真波形图

3.1.2 二维小波变换器的设计

在一维小波变换器基础上，人们提出了多种二维小波变换器的 VLSI 结构，各具特点。

行列式方案^[3]是最早被提出的一种方案，输入的原始图像首先经一维小波变换器进行小波行变换，地址控制器控制输出的小波系数按行存放在存储器中，结构如图 3-4 所示。当所有行均变换完成后，地址控制器将小波系数按列取出，送回到一维小波变换器，并将变换结果按列存回存储器，这样便完成了一层小波分解。而后，地址控制器再将存储器中的 LL 子带系数用同样的方式进一步分解。当完成所有层的分解后，地址控制器控制存储器输出最终的小波系数。但结构的简单带来的后果是性能的下降，有行变换和列变换无法同时进行，原始图像不能连续输入和系统的内存需求大这几个缺点。

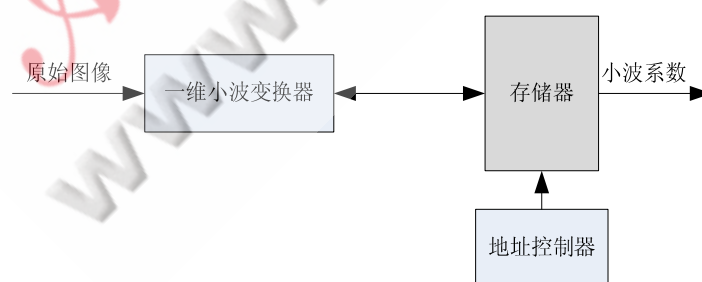


图 3-4 行列式结构小波变换器

针对行列式的这些缺点，Christos Chrysaflis 和 Antonio Ortega 等人提出了行式结构^[4]，如图 3-5 所示。输入的原始图像首先经一个一维小波变换器进行行变换，结果保存在行缓存中。当行缓存中的行达到一定数量时，一维小波列变换器开始工作。列变换器是一种变形的一维小波变换器。它并不是一次性对整列的数据进行小波变换，而是每次取出一列的几个数据，完成一次预测或更新，而后将数据存回行缓存器，并处理下一列数据。因此，这是一种沿着行方向进行的列变

换器。相比于行列式结构，行式结构中行变换与列变换可以同时进行，原始图像可连续输入，内存需求也比较小，充分提高了二维小波变换的性能。



图 3-5 行式结构小波变换器

本文基于行式结构的二维小波变换，提出了一种改进结构，如图 3-6 所示。在经典的行式结构基础上，本文设计的结构增加了两个缓存——输入缓存和 LL 缓存，分别用于暂存输入数据及 LL 子带数据，但行缓存的容量大大减小，故本结构的内存需求反而小于经典行式结构。

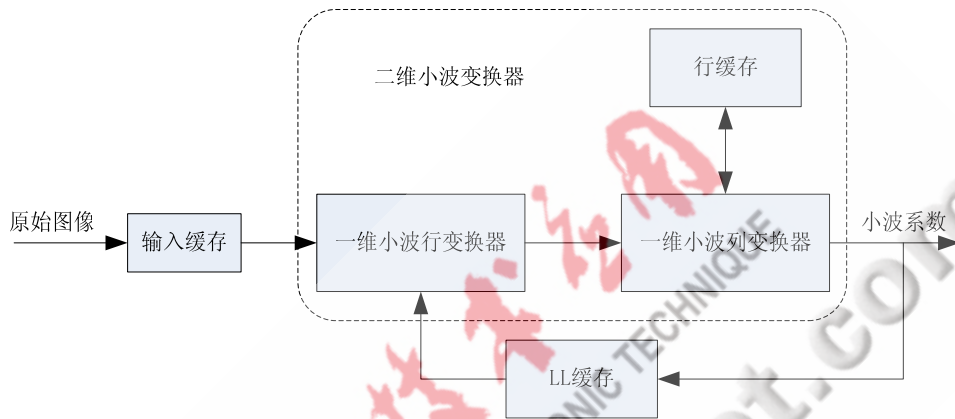


图 3-6 改进的行式二维小波变换器

图像进入二维小波变换器后，首先由一维小波行变换器进行行变换，而后进入一维小波列变换器。在状态机的控制下，一维小波列变换器从行缓存中逐列读出数据，与行变换器的输出合为一列，进行一步列方向上的预测或更新，并输出部分最终结果，其它中间结果存回行缓存。

为验证设计的有效性，使用 ModelSim 进行了仿真。图 3-7 为二维小波变换器的功能仿真波形图。

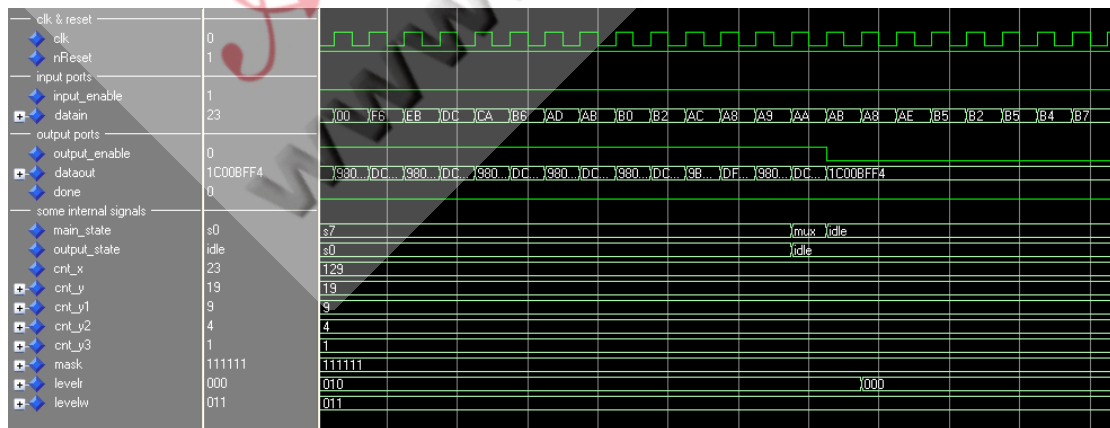


图 3-7 二维小波变换器的功能仿真波形图

3.2 EBCOT Tier-1 编码器

原始图像在进行小波变换和量化后，便进入编码过程。JPEG2000 中的编码

算法为嵌入式块编码，称为 EBCOT。该算法不仅能实现对图像的有效压缩，同时产生的码流具有分辨率可扩展、信噪比可扩展、随机访问和处理等优良特性。整个编码过程被分为两个步骤——Tier-1 编码和 Tier-2 编码。Tier-1 负责对每个码块独立地进行熵编码，由位平面编码和算术编码组成；Tier-2 对 Tier-1 中生成的压缩数据按一定顺序进行装配，生成 JPEG2000 标准码流。下面主要介绍位平面编码器和算术编码器的设计。

3.2.1 位平面编码器

位平面编码主要负责生成当前编码信息的上下文矢量(CX)和判决(D)，并与比特流一起送到算术编码器进行熵编码。由小波变换经量化得到的系数，可以将其分割成固定大小的代码块，大小一般可取 32×32 或 64×64 。各代码块可按每个系数样本比特位做位平面展开，最高层是符号位平面，下几层是幅值位平面。编码以位平面方式进行，从最高幅值位平面(MSB)开始，自上而下到最低幅值位平面(LSB)结束。在每个位平面上，除了最高幅值位平面只进行清理扫描(Cleanup)，其它位平面按顺序进行有效性传播扫描(Significant Propagation)、量级细化扫描(Magnitude Refinement)和清理扫描(Cleanup)，扫描方式如图 3-8 所示，以每 4 行组成一个条带，而每个条带内按列进行扫描。连续扫描第一列的第一组 4 个数据后，然后转向扫描第二列的第一组 4 个数据，如此一直扫描到最后一列的第一组 4 个数据，然后转向扫描第一列的第二组 4 个数据，一直到最后一列的第二组 4 个数据；按照这样的顺序一次扫描整个位平面。

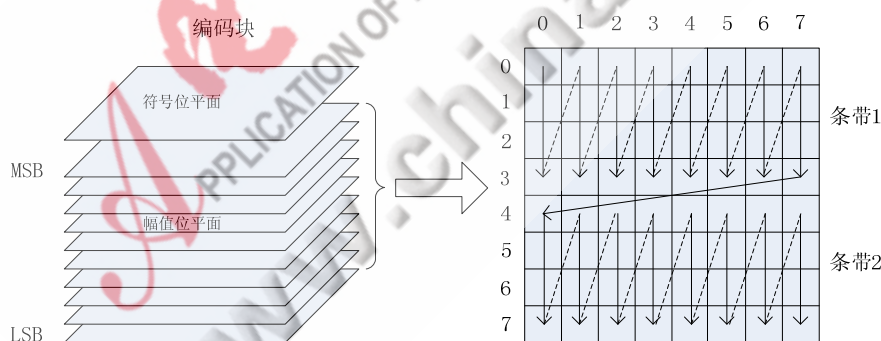


图 3-8 位平面编码扫描方式

在三个扫描过程中，每个系数样本比特位除了符号位 χ 和幅值位 v ，还有三个状态位：有效性位 σ 、量级细化位 δ 和访问位 η ，这三个状态位初始值为 0。

位平面编码算法是按从最高位平面到低位平面的顺序进行编码，属于串行扫描编码方式，编码速度受到很大的限制。所以，对位平面编码器进行了位平面并行优化^[5]。如图 3-15 所示，对于每个位平面都有一个编码器，每个编码器可以同时对其对应的位平面进行编码，这样编码速度得到了非常大的提高。但是基于位平面并行的位平面编码器必须解决状态位的更新问题。因为编码算法中，当前一层的位平面编码必须等到上一层位平面编码完毕之后才能开始编码，在上一层编码中 CUP 扫描编码会更新有效性状态位 σ ，SPP 扫描编码会更新有效性位 σ 和访

访问位 η ，MRP 扫描编码会更新量值细化位 δ ，也就是说，当前层的编码状态位是基于上一层编码更新。

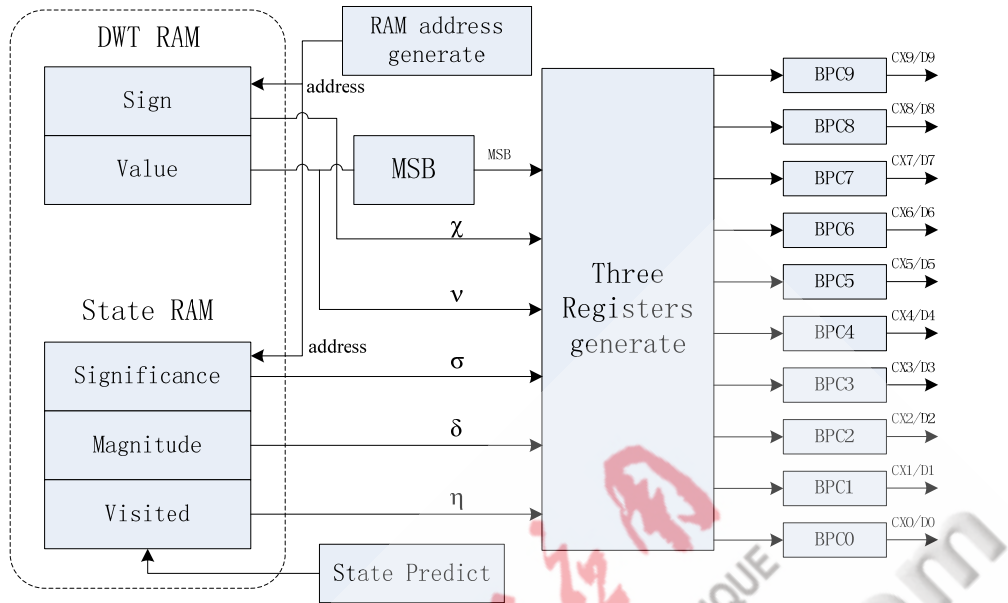


图 3-9 基于位平面并行优化的位平面编码结构

为解决上述问题，必须在并行编码之前提前预测更新每一层位平面的状态位值。如图 3-9 中 State Predict 模块，此模块就是用于状态位的预测：

(1) 对于有效性位 σ ，在符号编码原语中，只要幅值位为 1，就执行符号编码原语，置有效性位为 1，所以有效性可用公式 3-1 预测。

(2) 对于量级细化位 δ 只有在量级细化扫描编码中会更新，可以由公式 3-2 预测。

(3) 对于访问位 η ，因为位平面编码算法中，在每个位平面编码开始时就重设所有系数的访问位为 0，所以在此处就不需要提前预测更新了。

$$\sigma_p = \sum_{i=p+1}^{MSB} v_i \quad (3-1)$$

$$\delta_p = \sigma_{p+1} \quad (3-2)$$

在并行位平面编码器结构中，还包含了 5 个模块。其中有两块 RAM 模块，一块单口 RAM 存储符号位和幅值位，由于三个状态位在编码之前需要提前更新，所以选用双口 RAM 存储；RAM address generate 模块用于生成 RAM 地址，从 RAM 中读取数据；MSB 模块是最高位平面查找，在编码过程中需要跳过幅值位全为 0 的位平面，在编码前通过 MSB 模块查找最高位平面；Three Registers generate 模块将从 RAM 中读取的数据组合成三寄存器结构，三寄存器结构中包含了需要编码的一列 4bit 数据及其领域数据，然后位平面编码以三寄存器结构为单位进行编码；BPC 编码模块执行三扫描编码，生成上下文矢量 CX 和判决 D。

为验证设计的正确性，使用 ModelSim 对设计进行了验证。如图 3-10 所示，为基于位平面并行的位平面编码器的仿真波形。

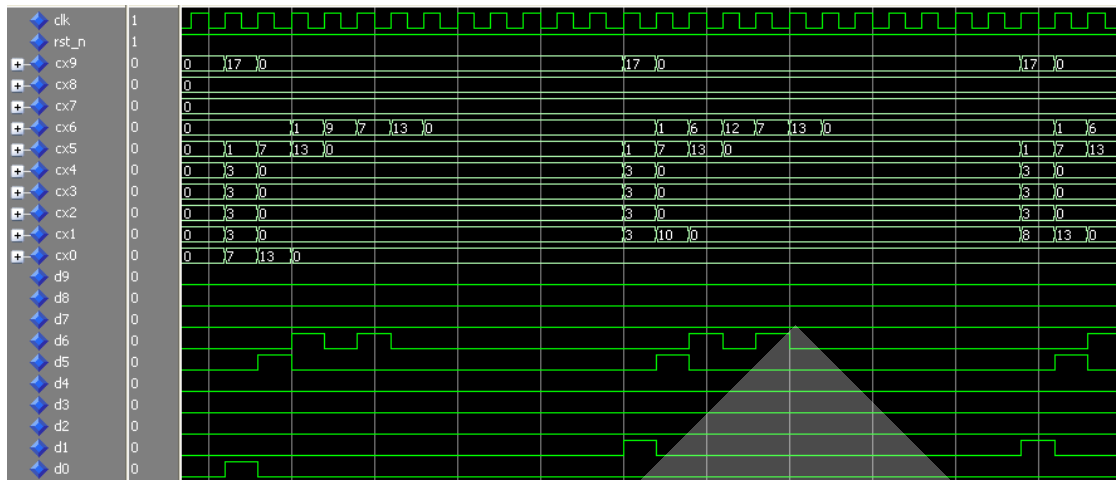


图 3-10 基于位平面并行的位平面编码器仿真

位平面编码器经过位平面并行优化，使各层位平面的扫描编码并行执行，编码速度相比于串行的扫描编码提高了近 10 倍。

3.2.2 算术编码器

算术编码是一种高效清除字符串冗余的算法，它按分数比特逼近信源熵，比 Huffman 编码具有更高的效率。

在 JPEG 中，采用霍夫曼编码作为无损压缩 DCT 系数的熵编码方法。霍夫曼编码按 DCT 系数大小来分配可变长码，故可用查表法简单实现。但是，由于预先调查了解符号序列的统计特性后制表，所以，此后一旦有与其性质相违背的符号输入进来，就难免使压缩特性恶化。

JPEG2000 中引入的自适应算术编码解决了这一问题。算术编码器并不把概率区间直接赋给输入码流中的符号 0 或 1，而是把它们赋给 MPS(More Probable Symbol)区间或 LPS(Less Probable Symbol)区间，然后将这两个区间在线段上排列。其中 MPS 对应的区间排列在 LPS 之上。这样，当接收到一个输入比特时，编码器按如下步骤进行间隔细分：

- (1) 根据输入上下文(CX)确定 LPS 的概率；
- (2) 确定该输入数据为 MPS 还是 LPS；
- (3) 根据步骤(2)的结果选定一个子间隔作为新的概率区间；
- (4) 选定区间中任意值表示输入码流；
- (5) 转到步骤(1)，接收新数据。

编码时采用 16bit 寄存器 A 表示概率区间，28bit 寄存器 C 表示生成码字。则上述划分过程如图 3-11 所示。

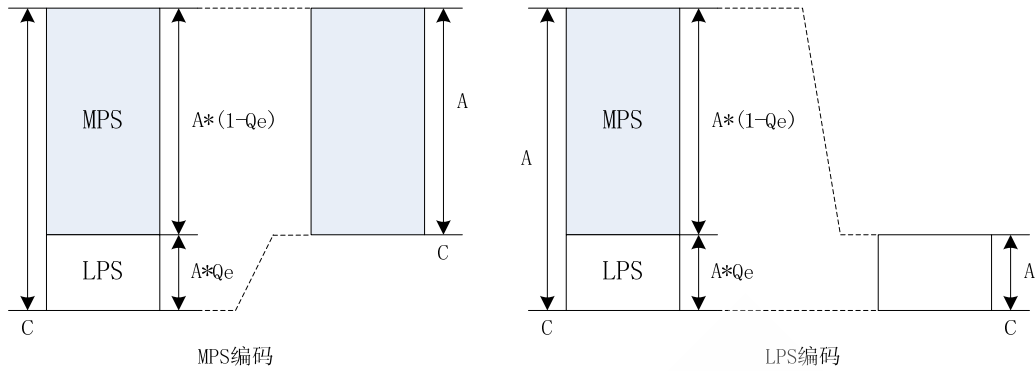


图 3-11 MPS 编码与 LPS 编码

对于 MPS 和 LPS 区间长度的计算，如式(3-3)、(3-4)所示。

$$A - (Qe \times A) = \text{对 MPS 的子区间} \quad (3-3)$$

$$Qe \times A = \text{对 LPS 的子区间} \quad (3-4)$$

由于乘法运算比加法运算需要更大的计算量，且增加了表示运算结果的精度。为了能够尽量避免上式中的乘法运算，设概率区间宽度 A 的值能近似 1，这样就可以将式(3-3)、(3-4)做改写为式(3-5)、(3-6)。

$$A - Qe = \text{对 MPS 的子区间} \quad (3-5)$$

$$Qe = \text{对 LPS 的子区间} \quad (3-6)$$

MPS 和 LPS 区间接这样的运算一直划分下去，在某一时间点上，概率区间就会比必要的精度范围($0.75 \leq A < 1.5$)小，在这种情况下就要通过重新归一化 A 与 C 维持精度范围，并且输出一个字节数据 B 。在区间的划分过程中，有时会出现 LPS 子区间比 MPS 子区间大的现象，为防止这种情况发生，当 LPS 变得比 MPS 大时，要进行 MPS 与 LPS 的状态交换。

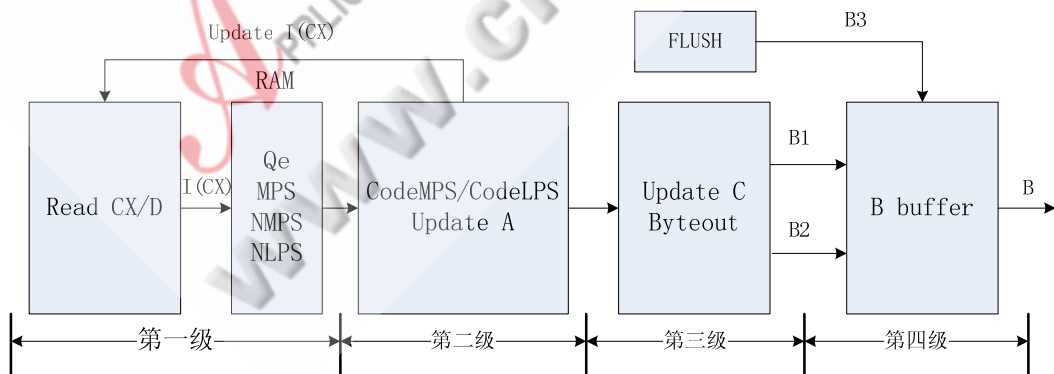


图 3-12 MQ 算术编码器的流水线结构

由于算术编码算法的复杂性，硬件实现组合电路关键路径延时较大，导致系统最高时钟频率较低，而且编码效率不高，所以必须对算法进行优化。如图所示，对算术编码器采用流水线实现，将复杂的算法分割成几级，如图 3-12 所示，第一级流水线读取 CX/D 信息，并根据索引 $I(CX)$ 读取从 RAM 中读取对应的 Qe 、MPS、NMPS 和 NLPS；第二级流水线进行 CodeMPS 规程和 CodeLPS 规程的判断，并更新寄存器 A ；第三级流水线更新寄存器 C ，并执行重新归一化中的 Byteout

规程输出数据 B; 第四级流水线进行数据 B 的缓冲。如下图为各级流水线仿真图。

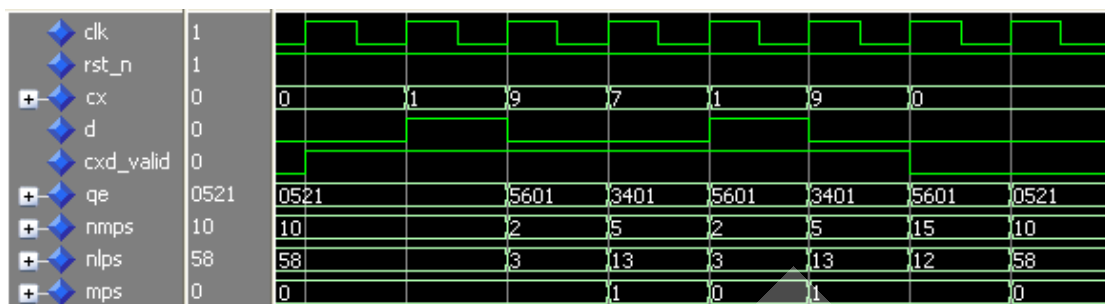


图 3-13 第一级流水线仿真

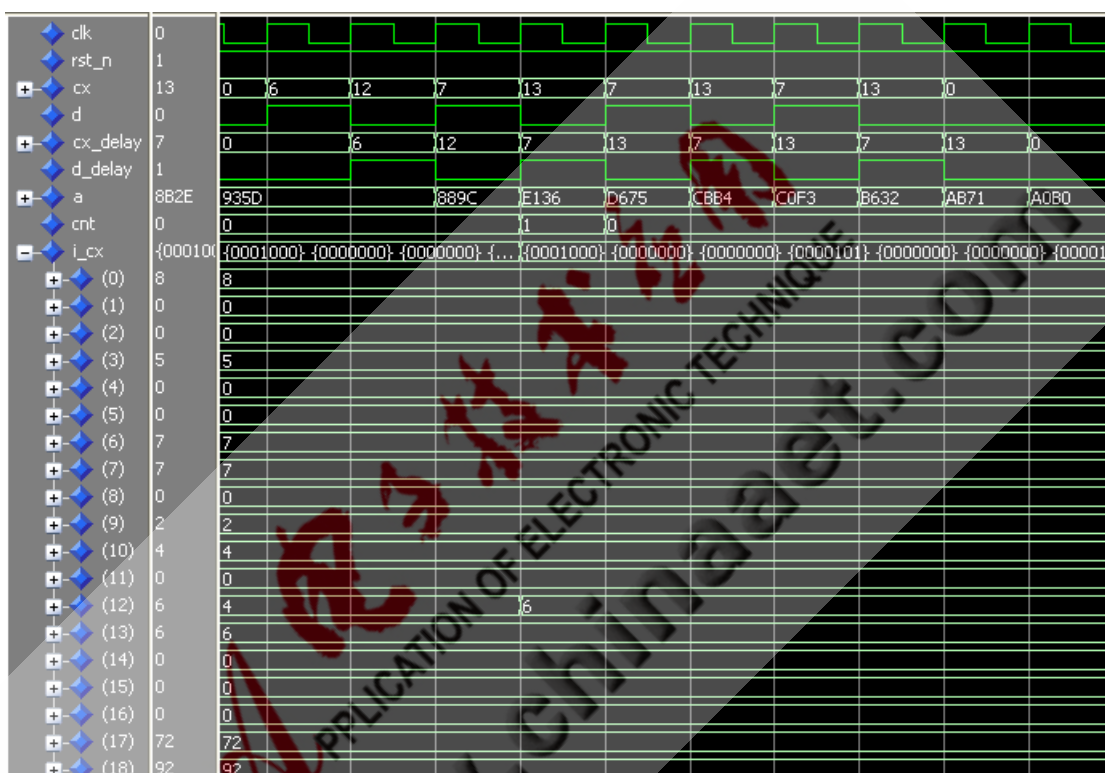


图 3-14 第二级流水线仿真

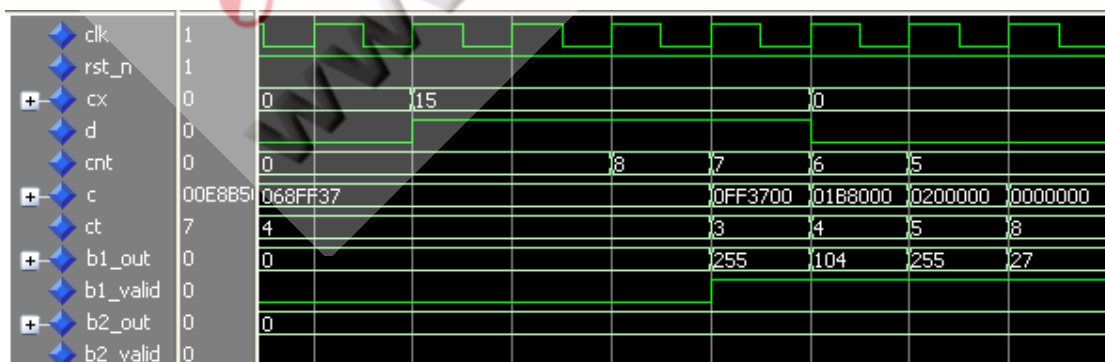


图 3-15 第三级流水线仿真图

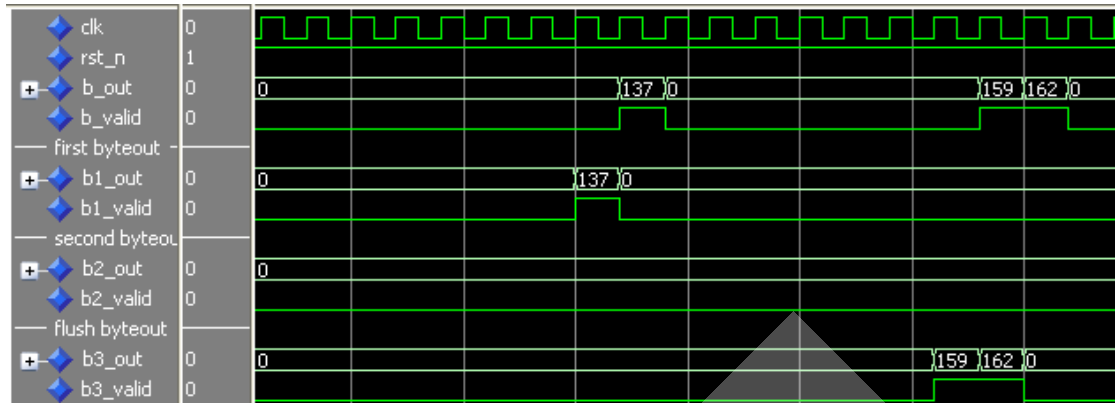


图 3-16 第四级流水线仿真图

算术编码器的流水线实现减少了组合电路关键路径延时, 增大了系统时钟频率, 如图 3-17 所示, 静态时序分析得到最高时钟频率达到 110MHz, 而且每一级并行执行, 编码速度得到了明显的提高, 达到 1bit/cycle。

Timing Analyzer Summary				
Type	Slack	Required Time	Actual Time	
1 Worst-case tsu	N/A	None	9.342 ns	
2 Worst-case tco	N/A	None	11.166 ns	
3 Worst-case th	N/A	None	-2.712 ns	
4 Clock Setup: 'clk'	0.947 ns	100.00 MHz (period = 10.000 ns)	110.46 MHz (period = 9.053 ns)	
5 Clock Hold: 'clk'	0.457 ns	100.00 MHz (period = 10.000 ns)	N/A	
6 Total number of failed paths				

图 3-17 算术编码器静态时序分析

4 JPEG2000 编码器的SOPC实现

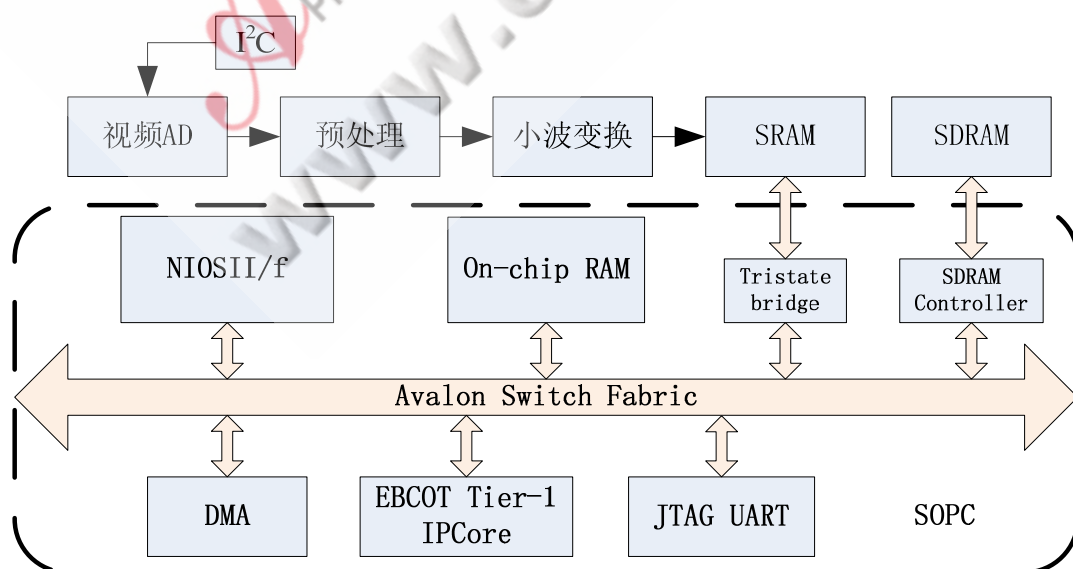


图 4-1 JPEG2000 编码器结构

如图 4-1 所示为 JPEG2000 编码器结构。预处理模块接收视频 AD 传来的图

像数据，经水平移位后发送给小波变换器，经小波变换后的数据经后处理模块存入 SRAM 中；当小波变换完成后，NIIOSII CPU 将 SRAM 中的数据取出，然后通过 DMA 将经过量化后的系数传给 EBCOT Tier-1 IPCore，编码完成后传回 NIIOSII 中；最后 NIIOSII CPU 进行码流组织后，将最终 JPEG2000 码流存入 FPGA 片内 RAM 中。而后，PC 端的 TCL 脚本通过 JTAG 接口将片内 RAM 中的 JPEG2000 码流取出，调用 JPEG2000 软解码器解码后，将结果显示在 PC 机显示器上。

4.1 预处理模块设计

预处理模块主要负责原始图像数据源与小波变换器的衔接，其功能与图像数据源直接相关，因此，该模块的设计需要“因地制宜”。针对本设计，下面以 PAL 制数据源为例介绍预处理模块的设计。

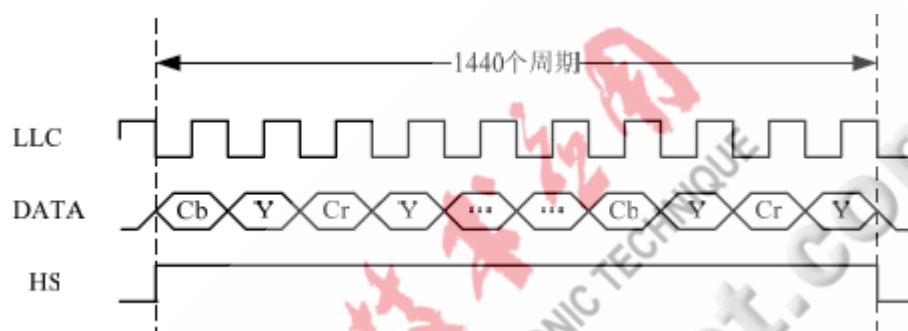


图 4-2 PAL 制一行数据的时序

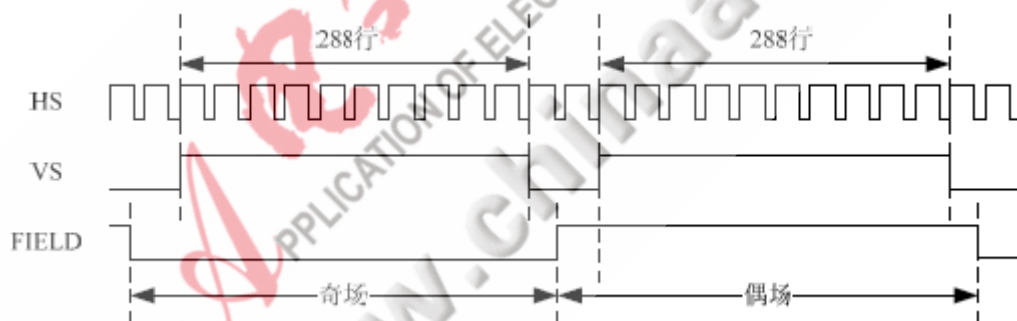


图 4-3 PAL 制一帧数据的时序

PAL 是一种被广泛使用的视频制式，每秒 25 帧，隔行扫描。标准的数字化 PAL 制分辨率为 720×576 ，其时序图如图 4-2、4-3 所示。

图 4-2 中，LLC 为 27MHz 点时钟，DATA 为 8bit 数据，HS 为行同步信号。当 HS 为高时，每个 LLC 上升沿对应一个数据。一行共 1440 个数据，其中亮度数据 Y 为 720 个，色差数据 Cb 和 Cr 各 360 个。

图 4-3 中，VS 为场同步信号，FIELD 为奇偶场指示信号。当 VS 为高时，数据有效，一场共 288 行有效数据。当 FIELD 为低时为奇场，高时为偶场(有的视频 AD 相反)。连续的一个奇场和一个偶场构成一帧图像。预处理模块依次完成以下工作：

- (1) 接收视频 AD 输入的奇场数据，隔行存入 SRAM 中；
- (2) 接收偶场数据，插入奇数行数据的空隙中，完成了奇偶场的合并；

- (3) 将 RAM 中的数据分割为小波变换器所需的图像片；
 - (4) 对数据进行水平移位后，发送给小波变换器。
- 经过这 4 个步骤后，便完成了一帧图像的预处理。

4.2 共享内存的小波变换

视频 AD 传来的图像数据经过预处理之后直接传给了小波变换模块，变换完成后又将小波系数存入 SRAM 中。所以小波变换模块无需与 NIOSII CPU 进行数据通信，而 SRAM 作为共享内存，NIOSII 可以直接从 SRAM 中将小波系数读取出来，这样就不需要 NIOSII CPU 额外的控制就能完成小波变换。

4.3 EBCOT Tier-1 IPCore 设计

小波变换后数据经过量化后，由 NIOSII CPU 控制将数据传给 EBCOT Tier-1 模块，开始 EBCOT 编码。将 EBCOT Tier-1 做成自定义外设，可以通过 Avalon 总线与 NIOSII CPU 通信。如图 4-4 所示为 EBCOT Tier-1 IPCore 在 SOPC Builder 中的定制。

Use	Connections	Module Name	Description	Clock	Base	End
<input checked="" type="checkbox"/>		<input type="checkbox"/> cpu_0	Nios II Processor			
		instruction_master	Avalon Memory Mapped Master	clk		
		data_master	Avalon Memory Mapped Master			
		itag_debug_module	Avalon Memory Mapped Slave		IRQ 0	IRQ 31
<input checked="" type="checkbox"/>		<input type="checkbox"/> sdram	SDRAM Controller	clk	0x08000000	0x0bffffff
<input checked="" type="checkbox"/>		<input type="checkbox"/> PROM	On-Chip Memory (RAM or ROM)	clk	0x00210000	0x0021bfff
<input checked="" type="checkbox"/>		<input type="checkbox"/> wram	On-Chip Memory (RAM or ROM)	clk	0x00228000	0x0022dfff
<input checked="" type="checkbox"/>		<input type="checkbox"/> tristate_bridge_1	Avalon-MM Tristate Bridge	clk		
<input checked="" type="checkbox"/>		<input type="checkbox"/> sram_b	SLS_LIP3_SRAM		0x00300000	0x003fffff
<input checked="" type="checkbox"/>		<input type="checkbox"/> dma_0	DMA Controller			
	control_port_slave	Avalon Memory Mapped Slave	clk	0x00235000	0x0023501f	
	read_master	Avalon Memory Mapped Master				
	write_master	Avalon Memory Mapped Master				
<input checked="" type="checkbox"/>	<input type="checkbox"/> EBCOT_Tier1_IPCore	ebcot_slave				
	avalon_slave_0	Avalon Memory Mapped Slave	clk	0x00230000	0x00233fff	
<input checked="" type="checkbox"/>	<input type="checkbox"/> sysid	System ID Peripheral	clk	0x00235088	0x0023508f	

图 4-4 EBCOT Tier-1 IPCore 在 SOPC Builder 中的定制

为了提高传输速度，通过 DMA 将小波系数传入 EBCOT Tier-1 IPCore 中，EBCOT 编码完成后将码流数据传回 NIOSII CPU 进行码流组织。

4.4 TCL 脚本设计

整个 JPEG2000 编码完成后，码流数据存放在 FPGA 内的片上 RAM 中。在 PC 机端设计了 TCL 脚本，将压缩后的 JPEG2000 码流取出，并解码成标准位图。TCL(Tool Command Language)的中文全称是命令工具语言，广泛应用于 EDA 软件中，通过 TCL 脚本可以制作简单的界面，并在后台完成大多数 Quartus II 软件的功能。如图 4-5 所示，为本设计中 TCL 脚本的流程图及工作界面。

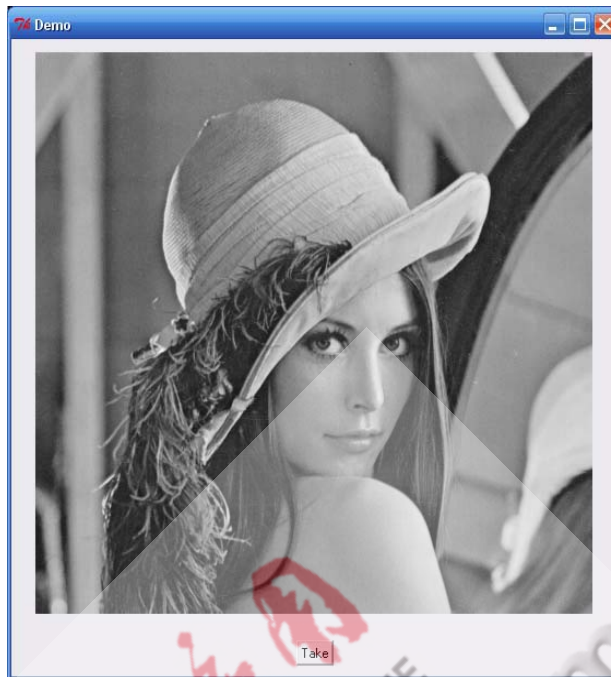
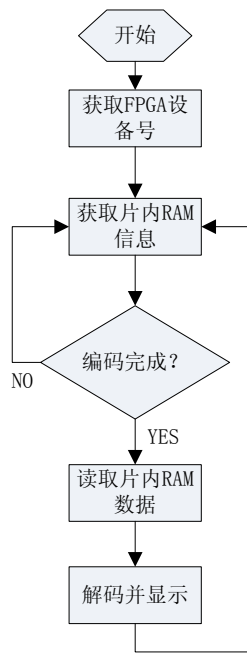
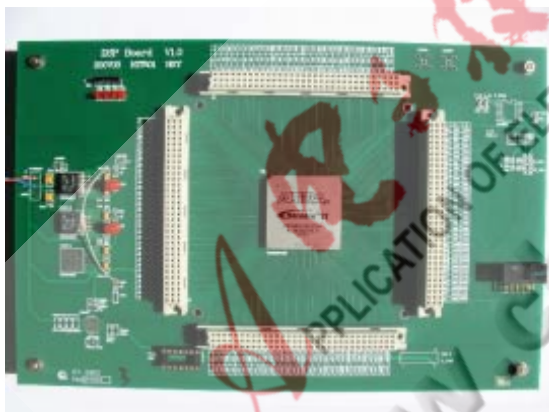


图 4-5 TCL 脚本流程图及工作界面

4.5 硬件平台设计



(a) 下层 FPGA 板



(b) 上层视频处理板



(c) 上下层板合照



(d) 整体工作环境

图 4-6 视频评估板

硬件平台设计以 Stratix II 系列的 EP2S60F1020C5 为核心。这是 Altera 公司的一款高端芯片，基于 90nm 工艺制造，拥有 48000 个逻辑资源，2.5Mbit 片内存储器，288 个硬件乘法器，最大可工作频率为 450MHz，其丰富的片上资源可以满足绝大多数视频应用。

视频 AD 用于将模拟视频信号数字化，本设计中选用的芯片是 AD 公司的 ADV7189。该芯片支持 PAL，NTSC 等多种输入制式及 RGB，YCbCr，灰度等多种输出模式，可按需要通过 I²C 总线配置。

如图 4-6 所示，为该板的实物图。其中(a)为下层板，(b)为上层板，(c)为上下层合照，(d)为工作环境全图。

4.6 实现结果及分析

在 Quartus 8.1 环境下，分别对小波变换模块、位平面编码模块、算术编码模块和整个 JPEG2000 编码器进行了综合，结果如表 4-1 所示。

表 4-1 综合结果

	ALUTs	Registers	Memory(bits)
小波变换	3577	3397	223816
位平面编码	10946	3834	30720
算术编码	8581	3045	39680
JPEG2000	28175	13826	736584

对 JPEG2000 编码器进行静态时序分析，如图 4-7 所示，最高运行频率为 98MHz。

Timing Analyzer Summary				
	Type	Slack	Required Time	Actual Time
1	Worst-case tsu	N/A	None	13.562 ns
2	Worst-case tco	N/A	None	15.738 ns
3	Worst-case tpd	N/A	None	2.413 ns
4	Worst-case th	N/A	None	1.450 ns
5	Clock Setup: 'pll:inst3 altpll_component_clk2'	0.971 ns	90.00 MHz (period = 11.111 ns)	98.62 MHz (period = 10.140 ns)

图 4-7 JPEG2000 编码系统静态时序分析结果

如图 4-8 所示，为通过摄像头采集到的图像经 FPGA 编码后在 PC 端解码显示的结果。这两幅图是 512×512 的灰度图片，压缩比为 20:1。

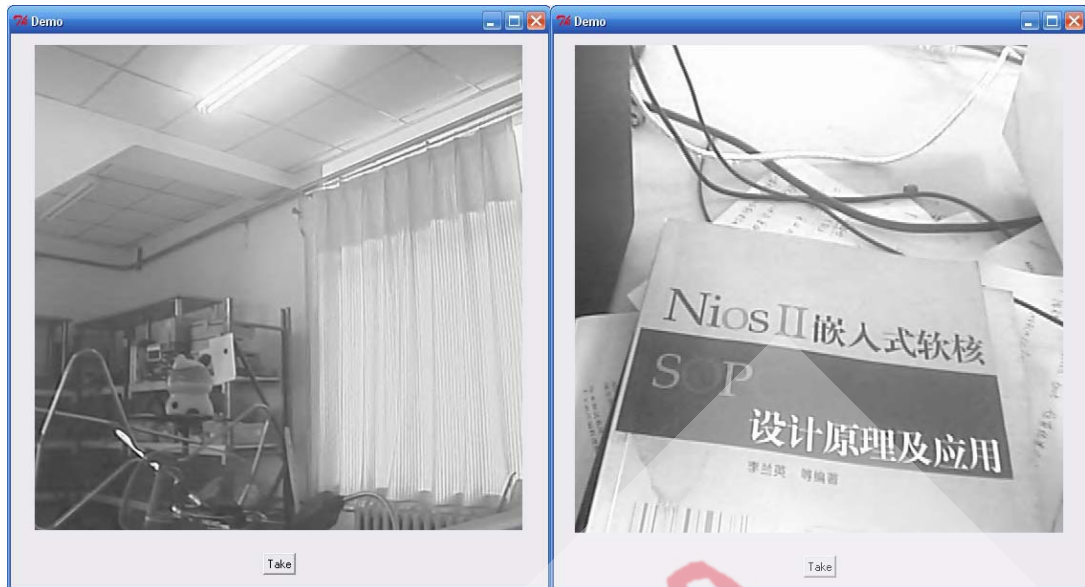


图 4-8 摄像头数据编码后 PC 端解码结果

为了进一步验证编码后的效果,本文也用一些标准测试图像在 FPGA 上做了编码测试,表 4-2 列出了编码结果。为对比,表中同时列出了标准测试软件 Jasper 在相同参数下的压缩结果。由表分析得,FPGA 上的 JPEG2000 编码结果在 PSNR 方面比 Jasper 的结果低 0.2~0.5dB,从图的质量上基本满足了要求。

表 4-2 FPGA 编码结果

	原图大小(KB)	压缩后大小(KB)	压缩比	PSNR(dB)	PSNR_jasper(dB)
Lena	256	13.0	20:1	35.89	36.07
Peppers	256	13.8	19:1	34.76	35.17
Washsat	256	12.1	21:1	34.75	35.06

JPEG2000 编码系统时钟为 54MHz,在压缩比为 20:1 的情况下编码分辨率 512×512 lena 图像,测得小波变换耗时 13.19ms, EBCOT Tier-1 编码器完成耗时 20.688ms,加上其他模块时间计算得 JPEG2000 编码分辨率 512×512 灰度图像速度为 29frame/s,如果工作在最高时钟频率 98MHz 下,编码速度能达到 52frame/s,基本满足了实时性要求。

5 结论

本文实现了基于 SOPC 的 JPEG2000 编码器。重点讨论了小波变换、位平面编码器和算术编码器的设计。

小波变换采用了改进的行式二维小波变换器。由于常规的行列式结构实现性能较差,而行式结构克服了这些缺点,使行变换与列变换可以同时进行,原始图像可连续输入,内存需求也比较小。本文基于行式结构,提出了一种改进的行式二维小波变换结构,充分提高了二维小波变换的性能。

位平面编码器采用了基于位平面并行的位平面编码器。常规的串行扫描方式

制约了位平面的编码速度，本文采用了位平面并行结构，在扫描编码之前提前预测状态位，使各层位平面能同时进行扫描编码，与串行方式相比编码性能提高了近 10 倍。

算术编码器采用了四级流水线结构。由于算术编码算法比较复杂，硬件实现的组合电路关键路径延时较大，导致时钟频率较低。本文采用了流水线结构，将算法分割成几段同时执行，不仅时钟频率达到 110MHz，而且编码速度达到 1bit/cycle。

参考文献

- 1 Tinku Acharya, Ping-Sing Tsai. JPEG2000 Standard for Image Compression Concepts, Algorithms and VLSI Architectures . 2005:163~185
- 2 小野定康, 铃木纯司. JPEG2000 技术. 科学出版社, 2004:130-199
- 3 S. Movva, S. Srinivasan. A novel architecture for Lifting-based Discrete Wavelet Transform. VLSI Design. 2003:202~207
- 4 A. O. Christos Chrysafis. Line Based, Reduced Memory, Wavelet Image Compression. IEEE Trans. IP. 2000:378~389
- 5 Tsung Han Tsai, Kuei Lan Lin, A high speed and low complexity integrated framework for JPEG2000, ICCS 2002, vol. 1, pp.493-496, Nov. 2002
- 6 李兰英. NiosII嵌入式软核SOPC设计原理及应用. 北京航空航天大学出版社, 2006: 445~449

原创性说明

郑重声明本论文是作者在导师指导下独立完成的。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已发表或撰写过的研究成果。