

H.264 及 AVS 视频解码器中 IQ/IDCT 的设计与实现

杨意仲¹, 张申科¹, 秦肖², 张楚²

(1. 同济大学 电子科学与技术系, 上海 200092;

2. 西北工业大学 软件与微电子学院, 陕西 西安 710065)

摘要: H.264 及 AVS 解码器中 IQ/IDCT(反量化/反离散余弦变换)模块的设计和实现。设计中考虑到解码速度、算法复用、系统耦合的情况,给出了一种系统解码时间消耗与系统资源占用较少的硬件设计方案,并给出最终仿真及后端设计结果。

关键词: H.264 AVS 反量化 反离散余弦变换

AVS 是中国第一个具有自主知识产权的数字音视频编解码标准,全称是“信息技术先进音视频编码标准”。H.264/AVC 标准是由 ISO/IEC 的运动图像专家组(MPEG)和 ITU-T 的视频编码专家组(VCEG)共同制定的新一代图像压缩标准。这两种标准在编码效率上较之以往的视频压缩标准都有了显著的提高,但与此同时硬件实现解码的复杂度也随之上升,给硬件设计带来挑战。本文针对 H.264 及 AVS 高清视频解码芯片中 IQ/IDCT 模块的设计进行论述,在充分考虑 H.264 及 AVS 中 IQ/IDCT 算法的基础上,提出了一种针对片上系统(SoC)的速度较快且面积较小的硬件设计方法^[1]。

1 IQ/IDCT 算法简介

1.1 H.264 IQ/IDCT 算法介绍

在 H.264 以前的标准,如 H.263 或 MPEG-4,都是采用 8×8 的 DCT 变换,而 H.264 采用了 4×4 整数变换,从而降低了算法的复杂度,也避免了反变换的失配问题,减小了块效应。同时 H.264 中还将 IDCT 变换中的乘法

融合到 IQ 的计算中,进一步降低了算法的复杂度。H.264 反量化公式^{[2][3]}如式(1)所示:

$$W_{ij} = Z_{ij} Q_{step} \cdot PF \cdot 64 \quad (1)$$

其中 W_{ij} 为 IQ 后的系数,将作为反变换的输入, Z_{ij} 为 IQ 输入, Q_{step} 为量化步长, PF 为预缩放因子,系数 64 用来消除取整误差。H.264 中的亮度块量化因子 Q_{step} 一共分为 52 个级别,而色度分量一共有 40 个级别。 QP (量化参数)每增加 6, Q_{step} 增大一倍,可将公式(1)改写为:

$$W_{ij} = Z_{ij} V_{ij} 2^{\text{floor}(QP/6)} \quad (2)$$

其中 V_{ij} 为 $QP \% 6$ 以及位置 i, j 的函数。 QP 与 Q_{step} 对应关系、 V_{ij} 与 $QP \% 6$ 以及像素位置关系可查表获得^[2]。由于编码时 H.264 采用三种变换方式,所以在解码端相应应有三种反变换:(1)针对所有 AC(交流)系数块的 4×4 整数反变换。(2)针对亮度 DC(直流)系数块的 4×4 反哈达玛变换。(3)针对色度 DC 系数块的 2×2 反哈达玛变换。H.264 中 AC 系数块的反变换公式^{[2][3]}如式(3)所示。

$$Z=C_iWC_i^T \quad (3)$$

其中 Z 为变换后的输出, W 为由公式(2)得到的反量化后的系数, 而变换矩阵 C_i 如式(4)所示。

$$C_i = \begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{bmatrix} \quad (4)$$

H.264 中亮度 DC 系数块的 4×4 反哈达玛变换公式^{[2][3]}如式(5)所示。

$$Y=H_4XH_4^T \quad (5)$$

其中 H_4 为:

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \quad (6)$$

Y 为反哈达玛变换的输出, X 为亮度 DC 系数。

H.264 中色度 DC 系数的 2×2 反哈达玛变换公式^{[2][3]}如式(7)所示, 其中 H_2 如式(8)所示。

$$Y=H_2XH_2^T \quad (7)$$

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (8)$$

1.2 AVS 中 IQ/IDCT 算法介绍

AVS 中亮度块的 QP 共有 64 个级别而色度块有 52 个级别。AVS 反量化公式^[4]如式(9)所示。

$$X_{ij}=(Q_{ij} \times \text{Dequant} + 2^{(\text{shift}-1)}) \gg \text{shift} \quad (9)$$

其中: Q_{ij} 为残差数据, Dequant 为反量化参数, shift 为移位参数, 均由查表得到, 请参考文献[4]。反量化过程只需要乘法、移位、加法运算, 易于硬件实现。AVS 采用基于 8×8 块大小的类 DCT 整数变换, 其变换系数采用整数后可以完全避免反变换中产生误差, 保证不会损失数据的精度, 使编解码端的数据相匹配。AVS 反变换公式^[4]如式(10)所示。

$$Y_{ij}=TX_{ij}T' \quad (10)$$

T 如式(11)所示, T' 为 T 的转置

$$T = \begin{bmatrix} 1 & 10 & 2 & 9 & 1 & 6 & 1 & 2 \\ 1 & 9 & 1 & -2 & -1 & -10 & -2 & -6 \\ 1 & 6 & -1 & -10 & -1 & 2 & 2 & 9 \\ 1 & 2 & -2 & -6 & 1 & 9 & -1 & -10 \\ 1 & -2 & -2 & 6 & 1 & -9 & -1 & 10 \\ 1 & -6 & -1 & 10 & -1 & -2 & 2 & -9 \\ 1 & -9 & 1 & 2 & -1 & -10 & -2 & 6 \\ 1 & -10 & 2 & -9 & 1 & 6 & 1 & -2 \end{bmatrix} \quad (11)$$

2 硬件设计和复用

IQ/IDCT 模块完成 H.264 与 AVS 中亮度块和色度块的反量化和反变换运算, 并将运算后的数据传给解码后续模块。IQ/IDCT 顶层硬件结构框图如图 1 所示。

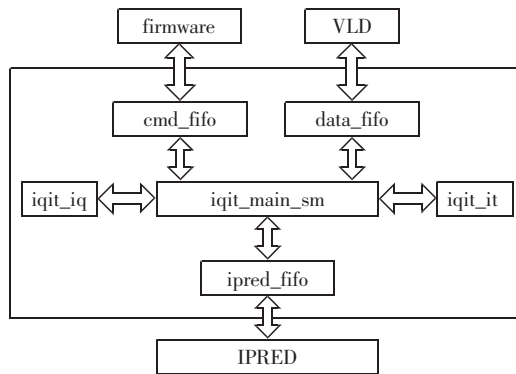


图 1 IQ/IDCT 顶层硬件结构框图

2.1 FIFO 的设计

设计中, IQ/IDCT 模块通过 FIFO 与上下级模块接口, 包括与 firmware 接口的 cmd_fifo(命令 FIFO)、与 VLD(熵解码模块)接口的 data_fifo(数据 FIFO)、与 IPRED(帧内预测模块)接口的 ipred_fifo(数据 FIFO)。考虑到解码过程中 VLD、IPRED 和 IQ/IDCT 并行工作以及在 H.264 与 AVS 中都以宏块为基本解码单位, 将 FIFO 深度设定为能存两个宏块的数据。

2.2 iqt_iq 模块的设计

iqt_iq 模块完成 H.264 与 AVS 反量化运算, 在系统设计中 VLD 模块和 IQ/IDCT 模块接口的 data_fifo 宽度为 198bit, 1 个像素占用 12bit, 每从 data_fifo 读一次数据将获得 16 个像素的残差即一个 4×4 块。H.264 的反量化包括 AC 系数的反量化与 DC 系数的反量化, AVS 没有 DC 系数只需对 AC 系数进行反量化。H.264 与 AVS 反量化运算的核心部分由乘法与移位操作完成, 在硬件上它们的反量化运算可以复用。该模块还负责 H.264 中 DC 系数反量化运算完成后的锁存和查找。iqt_iq 模块的硬件结构如图 2 所示^[5]。

当 iq_start 信号有效后, iq_sm 被启动, MUX 模块根据来自 iq_sm 的控制信息选择输出数据给乘法器。由于 AC 和 DC 系数的反量化都由 iqt_iq 模块完成且 DC 系数先要经过反哈达玛变换, 所以输入的 AC 系数和 DC 系数的位宽不同。LUT 模块实现所有 IQ 运算中需要的查找表提供反量化运算需要的参数、乘法器的固定参数和移位参数。考虑到解码速度和资源问题用了四组乘法器, 乘法器根据 MUX 和 LUT 传来的数据完成乘法操作。完成乘法运算后的数据经寄存器锁存后传给 SHIFT 模块。SHIFT 模块根据 LUT 传来的移位参数和 iq_sm 传来的控制信号完成移位操作, 移位操作完成后数据交由寄存器锁存。反量化运算完成后, iq_sm 发出 iq_done 信号通知主状态机运算完成。iq_sm 为整个 iqt_iq 模块的控制模块, 控制整个反量化运算。在反量化运算中, 除 H.264 的色度 DC 系数外, 其他系数都以 4×4 块为单元进行反量化运算, 而 H.264 色度 DC 系数以 2×2 块为单元进行反量化运算。H.264 的色度 DC 系数块的反量化运

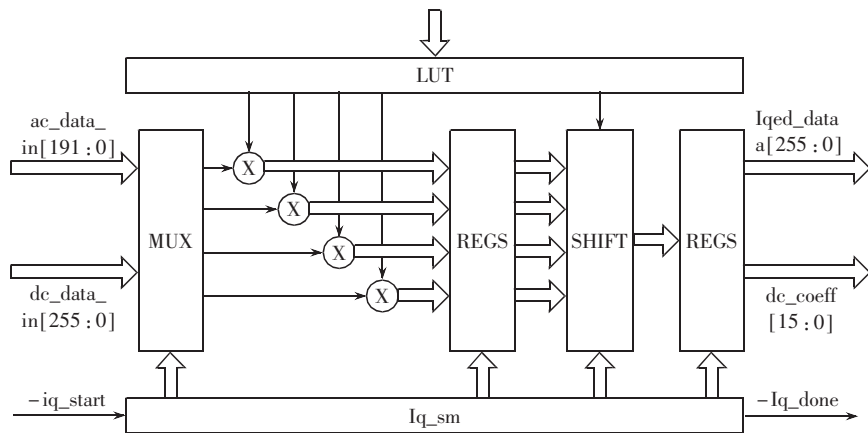


图2 iquit_iq 模块的硬件结构图

算需要 2 个时钟,其他系数块需要 6 个时钟。

2.3 iquit_it 模块设计

iquit_it 模块完成 H.264 与 AVS 的反变换运算。设计中为提高运算速度、减少逻辑资源,采用 H.264 协议与 AVS 协议中推荐的一维快速算法。H.264 的 AC 系数和亮度 DC 系数反变换的算法图如图 3 所示,当图中参数取 1/2 时为 AC 系数反变换算法图,取 1 时为亮度 DC 系数反变换算法图。H.264 的色度 DC 系数反变换算法图如图 4 所示。AVS 的反变换算法图如图 5 所示。

观察 H.264 与 AVS 一维算法图发现 H.264 反变换的所有算法可以借用 AVS 算法图左上角圈内的算法完成。由于采用一维快速算法,反变换运算只需要加法与移位操作。当反变换运算完成后,iquit_it 模块输出 it_done 信号通知主状态机反变换运算完成。H.264 色度 DC 系数块(2×2 块)需要 4 个时钟完成反变换运算,而 H.264 其他系数块(4×4 块)

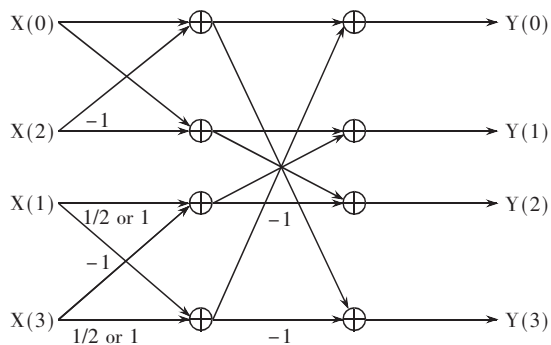


图3 H.264 亮度 DC/AC 系数一维算法图

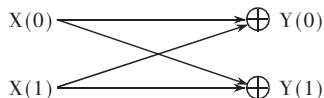


图4 H.264 色度 DC 系数一维算法图

需要 12 个时钟。AVS 完成一个 8×8 块单元的反变换运算需要 20 个时钟。

2.4 iquit_main_sm 模块设计

该模块为整个 IQ/IDCT 模块的核心控制模块。图 6 为其状态图。

在状态说明前先对状态图中的 zero_pass 路径做出解释,H.264 与 AVS 在预测方面具有比以前压缩标准更好的特性,使得预测图像与实际图像的差值比以前的标准更小。于是残差块全为 0 的概率大大提高。所以在残差块全为 0 的情况下,没有必要再对数据进行 IQ/IDCT 运算,于是在设计中加入了 zero_pass 通路,当为全 0 块时可以根据情况省掉相关运算,以节省

解码时间并降低解码芯片的功耗。

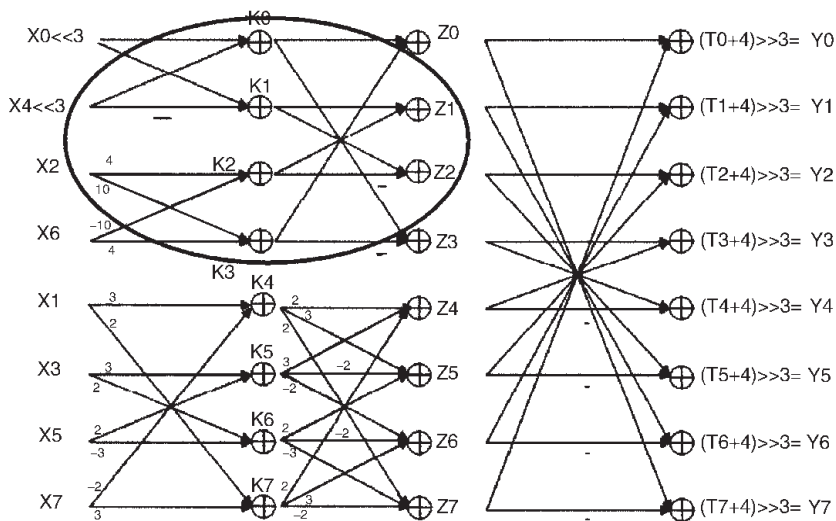


图5 AVS 反变换一维算法图

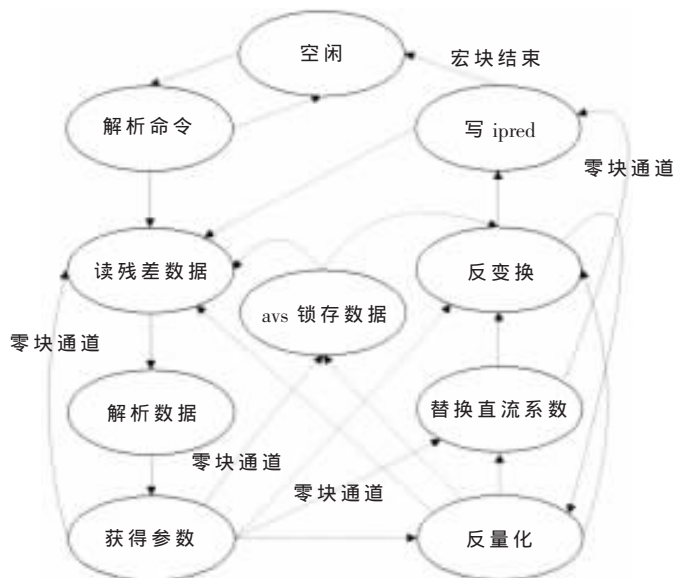


图6 iquit_main_sm 状态图

状态说明:

.idle: 如无解码要求, 状态机停留在 idle 状态, 否则状态机跳转到 parse_cmd 状态并发出命令读与 firmware 接口的 cmd_fifo。

.parse_cmd: 解析读取的命令。在解码系统中对 IQ/IDCT 模块有两级命令, 从 cmd_fifo 获得的第一个命令为序列级命令, 用来确定所解码流是 H.264 还是 AVS。随后命令为宏块级命令, 每开始解码一个宏块前都必须读一次命令, 获得该宏块的量化参数。如读取的命令为序列级命令状态则跳转到 idle 状态继续读命令; 如为宏块级命令状态则跳转到 read_data 状态。

.read_data: 发出命令读与 VLD 接口的 data_fifo, 读残差数据, 状态转移到 parse_data 状态。

.parse_data: 解析读取的数据。如为 H.264 码流数据的高六位提供残差数据标志信息: 色度亮度标志、直流交流标志、块索引。低 192 位为残差系数, 每个像素值占 12 位。如为 AVS 码流高六位无信息, 所以需要在代码中用计数的方式确定亮度色度标志, AVS 无 DC 块。parse_data 状态的下一状态为 get_qm 状态。

.get_qm: 根据残差数据标志信息确定反量化需要的固定乘法系数和移位参数。下面介绍该状态下状态的多种跳转。H.264: (1) 当前块为直流块且当前块为全 0 块状态则跳转到 read_data 状态。(2) 当前块为直流块且有非 0 系数状态则跳转到 it 状态, 因为协议规定对直流块先反变换运算再反量化运算。(3) 当前块为亮度全 0 交流块且为帧内 16×16 预测模式, 或者当前块为全 0 色度交流块状态则跳转到 replace_dc_coeff 状态。(4) 其他一律跳转到 iq 状态。AVS: 如当前块为全 0 块状态则跳转到 avs_latch_data 状态, 否则跳转到 iq 状态。

.iq: 完成 H.264 与 AVS 的反量化运算, 该状态通过与 iquit_iq 模块的一对握手信号 (iq_start, iq_done) 控制反量化运算。AVS 在 iq_done 信号有效时跳转到 avs_latch_data 状态。H.264: (1) 当前块为 DC 块, iq_done 信号有效时跳转到 read_data 状态。(2) 当前块为亮度交流块, 同时该宏块预测模式为帧内 16×16 模式或者当前块为色度交流块, iq_done 信号有效时跳转到 dc_replace_coeff 状态。(3) 其他情况跳转到 it 状态。

.avs_latch_data: 只有 AVS 才会出现该状态。AVS 中反变换运算为 8×8 整数变换, 而 H.264 中反变换运算为 4×4 整数变换, 同时在系统设计中 IQ/IDCT 与 VLD 接口的 FIFO 宽度设计为一个 4×4 块的大小。所以对于 AVS 码流需要锁存反量化后的数据。在该状态下, 若锁存的数据达到一个 8×8 块, 则状态转移到 it 状态否则转移到 read_data 状态。

.replace_dc_coeff: H.264 协议规定, 对于所有色度交流块和预测模式为帧内 16×16 的亮度交流块需要先替换直流系数再进行反变换运算。如当前块为全 0 块且它的直流系数也为 0, 则状态跳转到 write_ipred 状态, 否则

替换后跳转到 it 状态。

.it: 完成 H.264 与 AVS 的反变换运算, 该状态与 iquit_it 模块通过一对握手信号 (it_start, it_done) 控制反变换运算。当 it_done 信号有效后反变换运算完成, 如当前块为直流块状态则跳转到 iq 状态, 否则跳转到 write_ipred 状态。

.write_ipred: 写反量化反变换运算完成后的数据给 ipred_fifo 供 IPRED 解码用。如一个宏块的反量化反变换运算完成状态则跳转到 idle 状态, 否则跳转到 read_data 状态。

3 设计结果

本设计经过仿真, 在最坏情况下 IQ/IDCT 模块完成一个宏块的运算操作, H.264 需要 508 个时钟, AVS 需要 380 个时钟 (AVS 没有 DC 系数块)。将标准测试码流进行解码测试时, H.264 平均需要 280 个时钟完成一个宏块的 IQ/IDCT 解码运算, AVS 平均需 220 个时钟。当解码芯片工作在 166MHz 的情况下对高清图像进行解码时平均一个宏块的解码任务必须在 820 个时钟内完成, 由以上数据可知, IQ/IDCT 模块可以完成高清解码的任务。设计完成后在 FPGA 与 SMIC 0.18 工艺库实现的结果如表 1。

表 1 实验结果

	Gate count	Max frequency
FPGA (STRATIXII EP2S130F780C4)	5903 ALUTS	130MHz
SMIC 0.18	63211 gates	220MHz

本设计不仅考虑到 ASIC 设计中面积与时间的折衷以及 H.264 与 AVS 算法的复用, 而且还考虑到解码系统中各个模块之间的协调及同步工作, 完成了解码芯片中 IQ/IDCT 模块的硬件实现。由于在设计中采用了一系列的快速算法以及 H.264 与 AVS 算法的高度复用, 所以最终的设计在整个解码系统的解码过程中占用的时间、资源都相对较少。

参考文献

- [1] 毕厚杰. 新一代视频压缩编码标准-H.264/AVC[M]. 北京: 人民邮电出版社, 2005: 111-118.
- [2] H.264/MPEG-4 Part 10 White Paper- transform and quantization.
- [3] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, Proposed Draft Errata List with Revision-Marked Corrections for H.264/AVC[M], 2003(12).
- [4] 中国音视频标准工作组. 信息技术 先进音视频编码 第二部分: 视频 (Information technology-Advanced coding of audio and video-part2: video (报批稿), 2005.
- [5] KORDASIEWICZ R SHIRANI S. Hardware implementation of the optimized transform and quantization blocks of H.264[J], IEEE Electrical and Computer Engineering, 2004. Canadian Conference, Vol. 2: 943-946

(收稿日期: 2006-08-16)