

全高清 CABAC 解码器的设计与实现

盛怀亮, 林 涛

(同济大学超大规模集成电路研究所, 上海 200092)

摘要: 提出一种针对 H.264 标准的 CABAC 解码器的硬件加速器的设计方案。通过采用高效的状态机和良好的 SRAM 组织结构, 使平均解码速率达每周期 1 bit, 可以解码基于高档次的 H.264 码流, 实现对高清码流(1 920 × 1 088)的实时解码, 在中芯国际 0.18 μm 工艺标准单元库的基础上进行综合, 面积占 47 444 门, 工作时钟频率达 196 MHz。

关键词: H.264 标准; CABAC 解码器; 高档次; 硬件加速器

Design and Implementation of Full High Definition CABAC Decoder

SHENG Huai-liang, LIN Tao

(Institute of Very Large Scale Integrated Circuit, Tongji University, Shanghai 200092)

【Abstract】 This paper proposes a design scheme of hardware accelerator for CABAC decoder in H.264. It develops an efficient FSM and SRAM system so that the decoder can generate 1 bit every cycle, and it is capable of decoding High Profile(HP) video stream, achieving the requirement of real-time decoding. An ASIC implementation of the design is carried out in a 0.18 μm standard cell library of silicon technology, the estimated frequency is 196 MHz and the area includes 47 444 gates.

【Key words】 H.264; CABAC decoder; High Profile(HP); hardware accelerator

1 概述

H.264/AVC是由ITU-T VCEG和ISO/IEC MPEG联合制定的新一代视频编码标准, FExt(Fidelity Range Extensions)是该标准针对高保真视频压缩的扩展, 包括HP(High Profile), Hi10P(High 10 Profile), Hi422P(High 4:2:2 Profile), Hi444P(High 4:4:4 Profile)4个档次, 它们与MP(Main Profile)完全兼容, 且在稍微增加算法复杂度的基础上, 引入一系列先进的编码工具^[1], 极大地提高了对高清视频的编码效率。

在对高清视频进行编码的条件下, 文献[2]就 FExt 中的 HP 与 MP 在编码效率和计算复杂度方面进行了比较, 得出在 CABAC 熵编码方式下, 采用 HP 编码的平均比特率比使用 MP 节省 9.9%; 在 HP 条件下, 采用 CABAC 编码的平均比特率比使用 CAVLC 节省 17.9%。可以看出, CABAC 在对高清视频的编解码上具有巨大的优势, 但其算法复杂度较高。为了实现对高清视频的实时解码, 本文提出一种基于 HP 的 CABAC 硬件加速器的设计方案。

2 CABAC 解码流程

CABAC 的解码流程如图 1 所示。

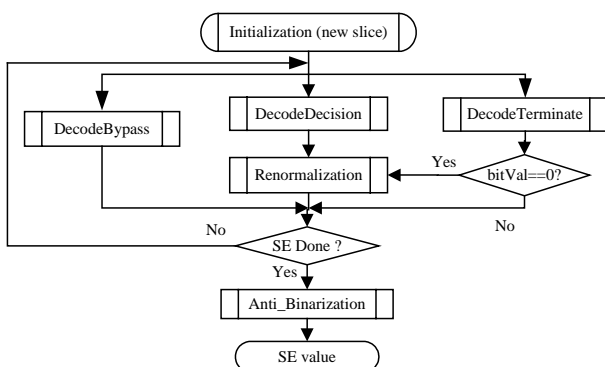


图 1 CABAC 解码流程

在 H.264 解码器的输入码流中, 数据的基本单位是句法元素(Syntax Element, SE), 视频码流就是由一个个句法元素衔接而成的。每个句法元素由若干个比特组成, 它表示某个特定的物理意义。在 H.264 中, 序列参数集(SPS)、图像参数集(PPS)及片头(slice header)的句法元素使用指数哥伦布(Exp_Golomb)算法解码, 其解码相对简单, 而 CABAC 主要解片层数据句法元素, 其码流占据整个视频码流的 90% 以上, 解码过程复杂得多。CABAC 解一个句法元素大致分为 3 个步骤: 解码初始化, 二进制算术解码, 反二进位化^[3-4]。

3 CABAC 解码器的硬件设计

3.1 硬件框架设计

H.264 解码过程采用软硬件协同的设计方案, 对于 CABAC 熵解码部分, 采用硬件加速器实现, 其框架如图 2 所示, 主要由主控制器和片内存储器组成。

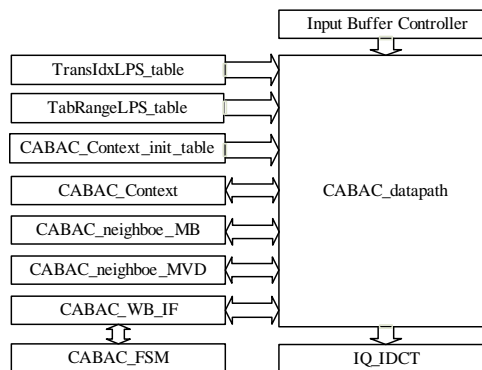


图 2 CABAC 解码器的硬件框架

作者简介: 盛怀亮(1982 -), 男, 硕士研究生, 主研方向: 视频编解码; 林 涛, 教授、博士生导师

收稿日期: 2008-05-10 E-mail: flysheng@sina.com

3.1.1 CABAC_FSM 模块

在解码一个宏块时，需要对句法元素的跳转进行控制，CABAC_FSM通过CABAC_WB_IF模块与CABAC_datapath模块进行交互，从而控制句法元素的解码顺序，这个层面的算法包含很多判决和分支操作^[5]，每个状态代表解码一个句法元素，由此控制整个解码器的状态时序。

CABAC 硬件加速器接收 CABAC_FSM 模块的命令，通过参考左面、上面宏块的相关信息，选择合适的概率模型，读取并更新概率模型来解特定的句法元素。本文的硬件加速器可以连续解码残差部分的 5 个句法元素：coded_block_flag, significant_coeff_flag, last_significant_coeff_flag, coeff_abs_level_minus1, coeff_sign_flag，对其他句法元素解码时，都是每接收一条命令解一个句法元素。此解码器支持 High Profile，即在 Main Profile 的基础上加入 transform_8 × 8_mode_flag, transform_size_8 × 8_flag, prev_intra8 × 8_pred_mode_flag, rem_intra8 × 8_pred_mode 等 8 × 8 变换及 8 × 8 帧内空间预测的句法元素，以实现支持对高清视频的支持。

基于此，本文设计了一个 2 级状态机来控制整个解码器的状态：一级状态机是针对残差类句法元素的解码；另一级则负责非残差类句法元素的解码，两者的工作流程如图 3、图 4 所示。其中，图 3 对应于图 4 中的 block(residual_coefficient) 部分。

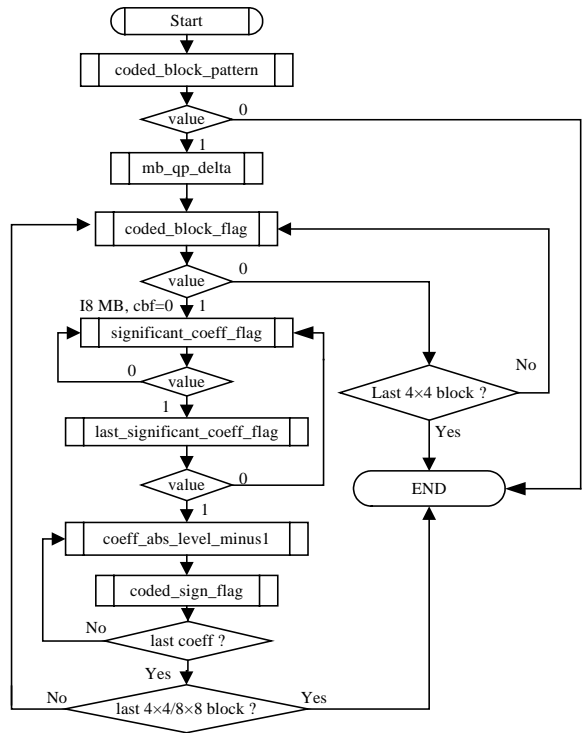


图 3 解码残差类句法元素的状态机

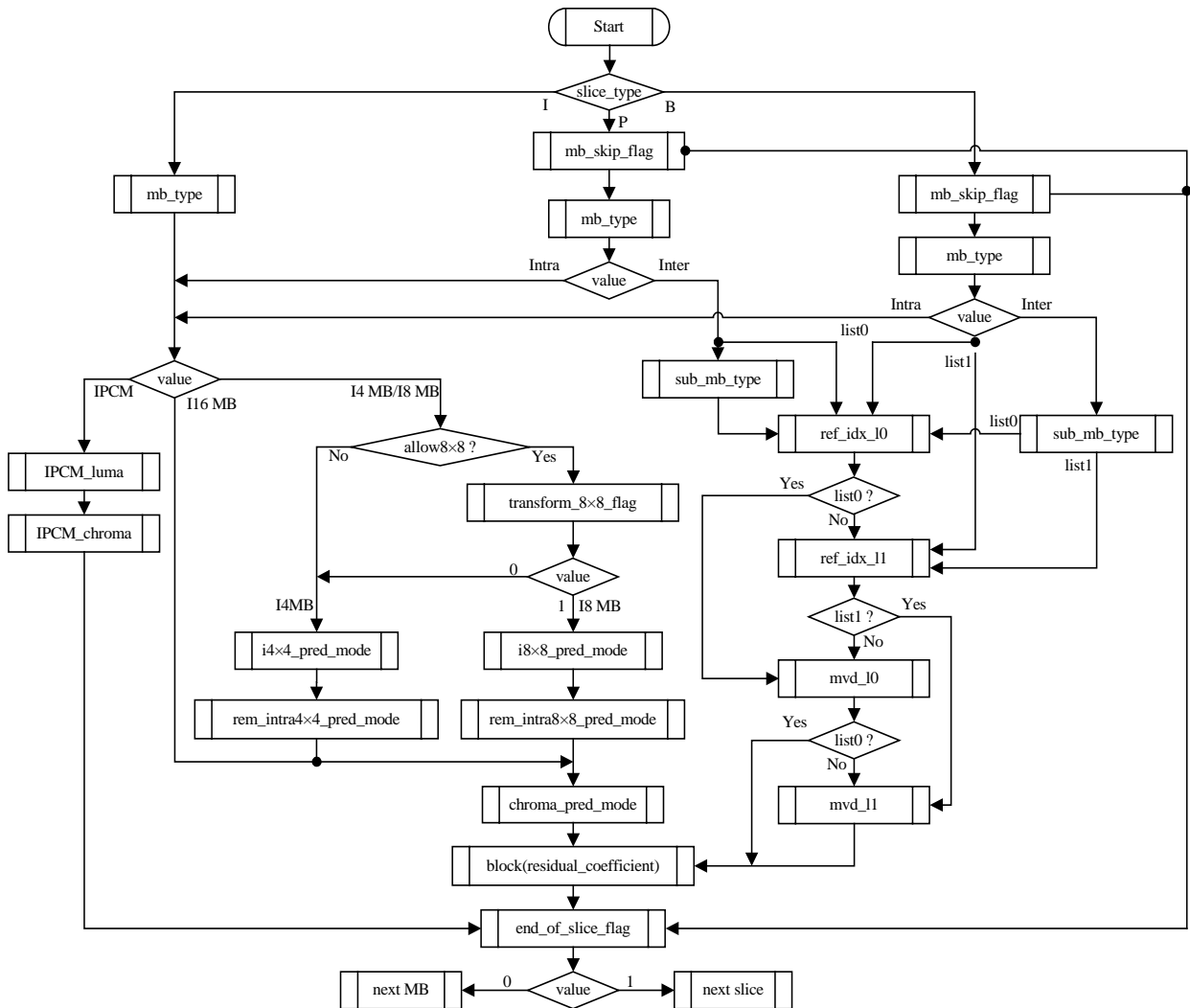


图 4 解码非残差类句法元素的状态机

3.1.2 CABAC_Context_init_table 模块

CABAC 开始解每个片时都要初始化上下文模型变量,期间会使用参数 (m,n) 的值进行计算, (m,n) 的值随 slice 类型的不同而不同,其中,P/B slice 会根据图像类型的不同划分为 0,1,2 这 3 种,因此,在 CABAC_Context_init_table 模块中设计了 2 块片内 ROM,分别用于存放 I slice 和 P/B slice 初始化时所用到的参数 (m,n) 。 (m,n) 用一个 16 bit 的数来表示,其中,高 8 位为 m ;低 8 位为 n 。H.264 共定义 460 个上下文模型,对于 I slice,须存储 460 组 (m,n) 值,而对于 P/B slice,共须存储 $460 \times 3=1380$ 组 (m,n) 的值。据此设计了 2 个宽度为 16 bit、深度分别为 512,2048 的 ROM。

3.1.3 CABAC_Context 模块

由于在 CABAC 解码的过程中,任一比特在完成解码后就会更新它所用的上下文模型的概率估计,由此实现对视频流实时统计特性的跟踪,因此需要一个既能读又能写的存储模块,为了加快解码速度,还为同一句法元素的下一比特在下一个周期开始前做好解码准备,因此,需要一个在写入新的上下文模型变量的同时能够读出待解的下一比特所需的上下文模型变量的存储器,使其在同一时钟周期内对不同的地址分别进行读和写操作。基于以上 2 点考虑,本方案把 CABAC_Context 模块设计为一块宽度为 7 bit、深度为 512 的双口 RAM(Dual-Port RAM, DPRAM),用于存放 460 个上下文模型变量。

3.1.4 CABAC_neighbor_MB 模块

CABAC_neighbor_MB 模块是一个用于存放相邻宏块信息的片内 SRAM。CABAC 在解当前宏块的句法元素时,需要参考当前宏块左面、上面宏块的信息来建立上下文模型。这就需要图像中一行宏块的信息进行保存,并在每解完一个宏块后进行更新。本文方案用于实时解高清码流(1920×1088),且支持 HP,即一行共有 $1920 \div 16=120$ 个宏块,而每个宏块需要保存 41 bit 的信息以供参考。因此,设计了一个宽度为 41 bit、深度为 120 的 SRAM 来实现此功能。

3.1.5 CABAC_neighbor_MVD 模块

在 CABAC 中,对句法元素 MVD 的解码比较复杂,在解当前宏块的 MVD 时,同样需要参考当前宏块左面、上面宏块的 MVD 信息来建立上下文模型。这也需要保存图像中一行宏块的 MVD 信息。与 CABAC_neighbor_MB 模块功能类似,CABAC_neighbor_MVD 模块是一个用于存放相邻宏块 MVD 信息的片内 SRAM。由于上下文建模时只须与 3 和 32 作比较,因此可用 7 bit 来表示 MVD 的信息,考虑到实际解 MVD 的过程中,每个宏块需要存放的数据为 $7 \times 7 \times 2 \times 2=196$ bit,如果开辟一块宽度为 196 bit、深度为 120 的片内 SRAM,则数据宽度和深度的比例比较大,占用的面积也比较大,因此,设计的 SRAM 宽度为 50 bit、深度为 480,可以大大减少芯片的面积。

3.1.6 TransIdxLPS_table, TabRangeLPS_table 模块

算术解码的计算复杂度主要体现在 2 个方面:(1)概率的估计和更新;(2)划分子区间的乘法运算 $R=R \times P_x$ 。H.264 标准在计算的复杂度和编码效率之间作了折中,分别建立了基于查表的概率模型和乘法模型。

在硬件实现上,设计了 2 个基于查找表的组合逻辑模块,其中,TransIdxLPS_table 用于概率估计和上下文模型的更新;TabRangeLPS_table 用于存储预先计算好的乘法结果,这样就

可以有效降低概率估计及更新和乘法计算的运算量。

3.1.7 CABAC_datapath 模块

CABAC_datapath 是 CABAC 解码的中央控制单元,主要负责上下文模型变量和解码引擎的初始化,解片层以下句法元素,控制读取和更新上下文模型变量、相邻宏块信息及相邻宏块的 MVD,并将解出的残差系数传给 IQ_IDCT 模块做后续处理,将其他句法元素的值通过 CABAC_WB_IF 模块传给 CABAC_FSM 作为控制状态机跳转的依据。本模块采用了几个高效的硬件设计方案:算术解码级有限状态机的设计,流水线的设计,解码流程的优化等。

3.2 硬件设计的优化

由于 H.264 CABAC 解码部分的运算总体上是串行解码,并行性较差,为了能够实时解码高清图像,达到平均一个时钟解一个比特的目标,必须对 CABAC 硬件加速器的设计进行优化,因此,本文提出以下几种高效的硬件设计方案。

3.2.1 有限状态机的设计

CABAC 解码系统主要包括 2 个层面的状态机:句法元素级的状态机和二进制算术解码级的状态机。其中,算术解码级的状态机是本设计的核心部分,其效率的高低直接影响到 CABAC 硬件加速器的解码速度。该状态机如图 5 所示。

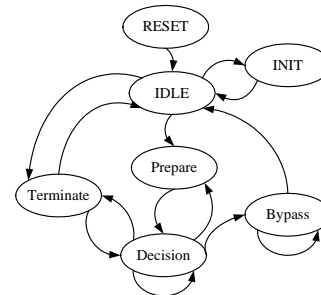


图5 算术解码级状态机

3.2.2 算术解码流程的优化

采用 Decode Decision 和 Decode Terminate 模式解码结束后,需要对解码引擎进行归一化处理,它是一个递归过程,递归的次数可以根据 CodIOffset, CodIRange 和查乘法表(TabRangeLPS_table)得到的 CodIRangeLPS 等变量的值,在解码结束之前计算出来。在解码结束后,归一化就可通过左移函数一次性完成,同时,递归的次数是 CodIOffset 的左移次数及读入码流的比特长度。本方案合并 Decode Decision 和 Decode Terminate 模式下的二进制算术解码并归一化这 2 个步骤,使其能在一个时钟内完成。另外,Decode Bypass 模式无归一化处理过程,所有操作可在一个时钟内完成。

3.2.3 流水线的设计

CABAC 解 1 bit 的过程可分为 2 个步骤:读取上下文模型变量;解码并更新上下文模型变量。本设计采用 2 级流水线结构,在一个时钟周期内,实现对 1 bit 的解码并读取下一个比特的上下文模型变量,这样就可以实现对概率模型的预读取,提高解码效率。

4 CABAC 解码器的硬件实现

根据上文的 CABAC 硬件加速器的硬件架构,利用 Verilog HDL 对其进行建模,并在 ModelSim 环境中编译、仿真,使用 Synplify 工具进行综合,工作频率可达 60 MHz,在 FPGA 验证平台上,可以解码基于 HP 的 H.264 码流,实现对

(下转第 241 页)