

面向实时视频编码应用的控制系统

张晓波, 胡薇薇, 丁文锐, 张其善

(北京航空航天大学电子信息工程学院, 北京 100083)

摘要: 从提高外设与内核并行性的角度引入流水线的思想, 实现基于查询和乒乓缓冲的控制架构, 针对结构中存在的问题提出改进的双级控制架构, 在该架构下实现基于 H.264 标准的实时视频编码。实际测试表明, 该系统能在保证系统稳定性的同时, 有效提高编码器的编码效率。

关键词: 视频编码; 嵌入式平台; 控制架构

Control System for Real-time Video Encoding Application

ZHANG Xiao-bo, HU Wei-wei, DING Wen-rui, ZHANG Qi-shan

(School of Electronics and Information Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100083)

【Abstract】 With the perspective of increasing the degree of parallelism of the on-chip peripherals and the core, this paper introduces the pipeline idea, implements a control architecture based on polling and ping-pong buffers, and develops it to an improved dual-stage control architecture. H.264-based video encoder is implemented under the architecture. Practical tests show that this system can meet the requirements of real-time capability and stability in video communications.

【Key words】 video encoding; embedded platform; control architecture

1 概述

目前最新的视频编码国际标准H.264^[1]以其优异的压缩性能和良好的网络亲和性成为实时视频通信领域的研究热点, 但编码复杂度的提高使这一算法难以在嵌入式平台上实现。除了针对硬件平台进行算法本身的优化之外, 一个合理的控制架构应该有效地利用冗余的硬件资源提高系统并行度, 目前已有学者从工程应用角度提出了不同的软件架构以满足视频通信对系统实时性的苛刻要求^[2-3]。文献[2]提出利用双核DSP(Digital Signal Processor)的A核控制系统流程、B核完成视频编码的系统架构, 实现了视频通信系统的基本功能, 但在编码实时性上缺乏考虑; 文献[3]提出了基于部件的TSSA(Trimedia Streaming Software Architecture)软件架构, 该架构更多地考虑了各部件的重用性, 任务调度需要实时操作系统支持, 适合较大型的多媒体应用; 而在低成本的应用中, 更多地采取传统的“主循环加中断”的软件架构。

2 实时视频编码控制的基本流程

2.1 基本流程

实时视频编码从逻辑上看包括以下4个步骤: 视频源的采集, 下采样, 压缩编码以及码流输出。

视频源是 PAL(Phase Alteration Line)制式兼容的 CCD (Charge Coupled Device)摄像头的模拟视频信号输出, 经 ADC (Analog to Digital Converter)进行 PCM(Pulse Code Modulation)编码后进入系统, 这时 PPI(Parallel Peripheral Interface)看到的是每秒 25 帧的 ITU-R 656 兼容的码流, 并被过滤为一场 D1 有效视频存入 SDRAM(Synchronous Dynamic Random Access Memory), 这涉及一个从系统外设到存储器的 DMA (Direct Memory Access)过程。

由于用于视频编码的数据源是 CIF(Common Intermediate Format)格式的图像, 因此需要将一个 1 440 × 288(4:2:2)的图

像采样为 352 × 288(4:2:0), 这涉及一个从存储器到存储器的 DMA 过程。

压缩编码主要由内核负责, 其中部分算法(如插值数据)可以分配相应的 DMA 通道, 由内核负责与其同步, 并随编码函数返回。从系统级的角度考虑, 认为这部分 DMA 是包含在编码算法内部的。

压缩后的码流以 230 Kb/s 的波特率经由 RS-422 兼容的 UART(Universal Asynchronous Receiver Transmitter)串行输出, 这涉及一个从存储器到外设的 DMA 过程。

系统流程如图 1 所示。

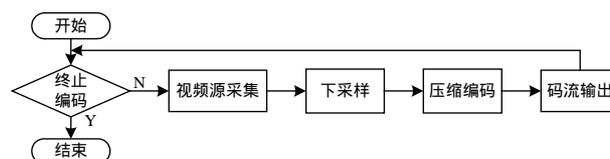


图 1 系统流程

2.2 流水线结构

为了满足实时性的要求, 需要减小处理器内核的开销。在以上 4 步操作中, 视频源采集、下采样和码流输出这 3 步由 DMA 控制器^[4-5]负责完成。内核仅需要在使能 DMA 前配置其工作模式, 并在 DMA 完成之后负责与其同步, 相比于海量数据搬移所带来的时间上的损耗, 这部分开销可以忽略不计。

基于以上考虑, 将一个内核过程和 3 个 DMA 过程作为流水线的 4 个节点, 在一个流水线周期内完成这 4 类操作。

基金项目: 国家“863”计划基金资助项目(2006AA701424)

作者简介: 张晓波(1982-), 男, 硕士研究生, 主研方向: 信息传输与处理; 胡薇薇、丁文锐, 博士研究生; 张其善, 教授、博士生导师

收稿日期: 2007-11-20

E-mail: zxbuaa@gmail.com

因为它们是同时进行的，所以各级流水线不能复用彼此的内存空间。这实际上是以增加空间上的负担换取时间上的性能提升。图 2 为系统的流水线时空图，其中， C 代表视频源采集； S 代表下采样； E 代表压缩编码； T 代表码流输出，数字 0 和 1 分别代表 2 个乒乓缓冲。可见，经过 4 个周期后，流水线被充满。

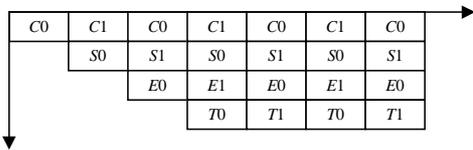


图 2 系统流水线时空图

3 基于查询和乒乓缓冲的控制架构

上述 4 级流水线是根据操作的功能划分的，理想的情况是各级流水线按时钟节拍完全同步，这样只需要事先为它们划分各自所需的存储空间，然后简单地将上一级的输出连接到下一级的输入，系统便可以稳定地工作。然而事实上各级流水线的耗时并不相等，这样流水线的周期只能是耗时最长一级的流水线周期。快速的节点在完成自身的操作后不能马上启动下一个周期，必须等待慢速的节点完成操作。因此，需要提供一种机制来匹配这种时间上的不对称，以确保数据的正确和系统的稳定。

3.1 实现方式

3.1.1 基于查询的同步方式

从这 4 级流水线所对应控制器的性质来看，只有压缩编码由内核参与完成，其他 3 个节点都由相应的 DMA 控制器负责。DMA 控制器无法判断应该等待还是马上启动下一个周期，因此，同步工作应交由内核处理。

处理器内核在压缩编码前将 3 个 DMA 通道配置为 STOP 模式^[5]，然后在描述符^[6]的最后一级打开 DMA 完成中断，并且在系统级屏蔽它。使能 3 个 DMA 通道后，处理器内核开始进行压缩编码，这时流水线的 4 个节点开始并行工作。当有 DMA 完成后，相应的 DMA 状态寄存器置位，但因为事先被屏蔽，所以不会对内核产生中断。内核可能在此之前或之后完成编码算法，然后进入一个不断查询的状态，只有当 3 个 DMA 状态寄存器均置位后才跳出这个循环，并且开启下一个流水线周期。

以上就是基于查询的同步方式。屏蔽中断是因为单个 DMA 完成只是必要条件，而 DMA 状态寄存器能提供足够的信息，所以没有必要支付响应中断的开销。

3.1.2 乒乓缓冲机制

解决了流水线节点之间的同步问题，下一步就需要将它们连接起来。联系 2 级流水线的就是存储器中的数据。为了保证每一级的输入在当前流水线周期是稳定的，不会被上一级的输出所覆盖，需要在相邻 2 级流水线之间开辟 2 块存储区，即所谓的乒乓缓冲^[5]，轮流用作上一级的输出和下一级的输入。图 3 是从空间上看到的流水线结构。

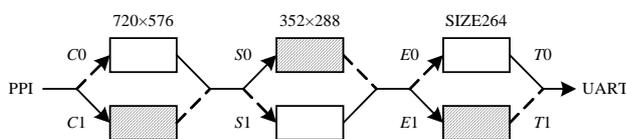


图 3 系统流水线结构

其中，空白矩形代表当前被读的缓冲；阴影矩形代表当

前被写的缓冲；实线箭头指示当前的数据流向；虚线箭头指示下一个周期的数据流向，它们按流水线周期交替进行。可见，经过 4 个周期后，流水线被充满，之后每个周期处理一帧图像。内核通过乒乓机制保证每个流水线节点的输入输出都是有效的。

3.2 问题分析

通过查询机制能确保流水线节点之间很好地同步，内核在得知另外 3 个 DMA 节点均完成各自的数据搬移工作后才开启下一个流水线周期，不用强行限定各节点在一段时间内必须完成。乒乓机制保证前后 2 级流水线之间没有数据冲突，实现流水线各节点的并行工作，但在保证数据正确和系统稳定的同时，无法实现最好的编码性能。

3.2.1 时间冗余度的减小

在理想情况下，流水线的上一周期与下一周期所处理的场景应该是“紧致”的，因此，流水线周期取决于数据源采集的周期，只要压缩编码或者码流输出所耗时间不超过这个值，就不会对系统的实时性产生负面影响。例如 PAL 制式摄像头的采样频率是每秒 25 帧，则编码器的速度至少略快于这个值。

如果压缩编码或码流输出的周期决定了流水线周期，则视频源采集 DMA 在完成自身的操作后不得等待一段时间，2 帧场景的时间差必然降低它们的相关度，这在按 P 帧编码时体现为耗时的增加和码流的增大。

3.2.2 采集效率的降低

更加严重的影响来自于频繁地开关 PPI，因为从概率上这必定会拉长采集周期。如图 4 所示的 PAL 制式帧结构^[5]，第 1~312 行为顶场 ($F=0$)；第 313~625 行为底场 ($F=1$)，场信息 F 包含在 ITU-R 656 数据流中。PPI 的帧同步机制是在 F 的下降沿，即遇到每一帧的顶场时才真正启动 DMA。因此，在使能 DMA 后总要忽略上一帧的部分数据，因为帧起始标志 (即 F 的下降沿) 是在使能 DMA 之前收到的。最坏的情况是刚好没收到帧起始标志，这样将忽略掉整帧数据。

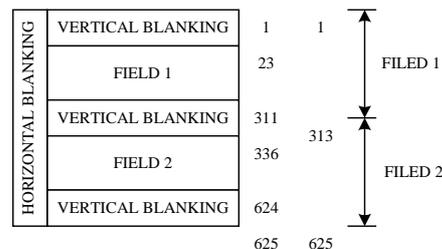


图 4 PAL 制式帧结构

4 改进的双级控制架构

要解决以上问题，首先不能关闭 PPI，这样便不会损失采集的效率。同时连续采集的 2 帧场景是“紧致”的，时间冗余度没有减小。不过同步方式需要做出相应的调整，即配合中断与查询的双级控制架构。

4.1 实现方式

为了提高数据源采集的效率，并保证算法的并行度及系统的稳定性，视频源采集 DMA 一经开启后就始终独立于内核，以预先配置的模式工作。它本身并不参与同步，只是周期性地更新自身状态，以便使流水线主动与它同步；另一方面，流水线缩减为下采样、压缩编码和码流输出 3 级，它们内部的同步方式保留上述查询与乒乓缓冲机制，然后内核在每个流水线周期开始前通过查询一个定义得很好的状态结构

与采集 DMA 同步。在采集端需要解决的问题是如何表现自身状态，在流水线端需要解决的问题是如何提取数据源。

4.1.1 多缓冲的采集端

在采集端开辟多个缓冲，原始数据通过 PPI 以 DMA 方式循环地填入。变化中的缓冲区在不同时间交替，而从外部看这些缓冲区，同一时刻只有一个处于变化状态，其余都是相对稳定的。由于 DMA 状态寄存器并不集中体现这些信息，因此需要定义一个状态结构包含缓冲区相应状态，并在每个描述符对应的 DMA 完成中断中进行更新。缓冲区的数目与“跳帧”的概率相关，根据具体的硬件资源适当增加缓冲区数量可以改善编码质量。

4.1.2 改进的同步方式

流水线同步方式与前文所述的思想一样，只是此时内核不能通过查询视频源采集 DMA 的状态寄存器来确定采集是否完成。采集被配置为一个循环链表描述的 DMA 过程。内核通过查询自定义的状态结构，按照默认的时间顺序改变下采样 DMA 起始地址的指向，由此保证正在变化的那一帧缓冲区的数据不会被采样，如果那一帧数据是默认的，就通过“跳帧”来解决。

图 5 是从空间上看到的双级控制架构的结构。可见，来自 PPI 的数据正通过采集 DMA 写入第 5 个缓冲区，在此之前一个 DMA 完成中断更新了状态结构，它提供内核需要的信息，包括目前有数据变化的缓冲区编号、下采样 DMA 对应采样的缓冲区编号以及是否需要“跳帧”等。而下采样之后的流水线仍通过乒乓机制同步。

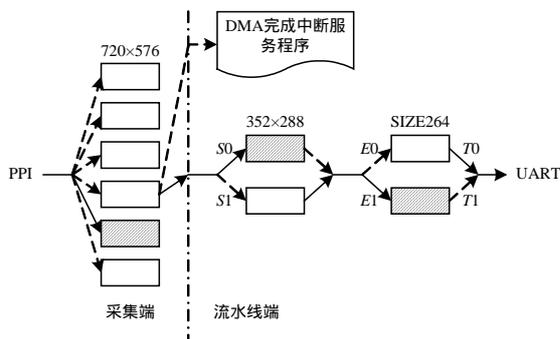


图 5 双级控制架构结构

4.2 性能分析

4.2.1 编码性能

对比 2 种控制架构在不同场景变化下的流水线周期，可以得出它们编码效率方面的差距。PAL 制式摄像头采样频率为每秒 25 帧，编码器对缓慢运动场景的处理能力约为每秒 20 帧，6 组数据分别对应不同剧烈程度的运动场景，用于测试的软件环境是 VisualDSP++ 4.5。实验数据如表 1 所示，其中， T_1 和 T_2 分别对应 2 种控制架构下的流水线周期。 T_1 对场景变化不敏感，约为 0.08 s，这实际上是采集一帧图像所耗时间，因为频繁开关 PPI 导致采集拉长了整个流水线周期。第 2 种控制架构流水线中不包含采集端， T_2 实际反映了编码周期，根据场景变化剧烈程度的不同，编码周期逐渐从 0.05 s 增加到 0.08 s。

表 1 流水线周期和编码效率

	0	1	2	3	4	5
T_1/s	0.078 0	0.081 7	0.084 2	0.084 1	0.084 4	0.085 2
T_2/s	0.051 2	0.058 4	0.062 3	0.070 9	0.075 3	0.081 4
效率	0.65 6	0.71 4	0.74 0	0.84 3	0.89 2	0.95 5

T_2 与 T_1 之比可以大致反映第 1 种控制架构的编码效率

以及由频繁开关 PPI 导致的时间上的损耗，这种损耗在编码周期较小时尤为明显，而这在第 2 种控制架构中可以很好地避免。同时由于连续采集的 2 帧场景是“紧致”排列的，不会出现时间冗余度降低的问题，这体现在图像质量和按 P 帧编码的速度上。

为匹配采集端与流水线端在速度上的不对称，采用周期性的“跳帧”来解决。表 2 显示在以上 6 种场景下编码 1 000 帧中测得的跳帧次数 N_{skip} 以及由此计算得到的跳帧频率 f_{skip} 。

表 2 编码 1 000 帧的跳帧次数和跳帧频率

	0	1	2	3	4	5
N_{skip}	34	49	60	72	80	86
f_{skip}	0.849	1.223	1.512	1.824	2.011	2.148

假设采集周期为 T_c ，流水线周期为 T_p ，并且已在采集端开辟 N 个循环缓冲。如果流水线周期因为编码速度较慢而被拉长，即编码周期决定了流水线周期并且 $T_p > T_c$ ，则“跳帧”频率为： $f_{\text{skip}} = 1/\Delta T = \frac{1}{N}((1/T_c) - (1/T_p))$ 。只要采集端和流水线端的速度不匹配，就一定会出现跳帧现象，但提高编码速度(减小 T_p)和开辟更多缓冲(增加 N)均可减小跳帧频率。实际应用中可根据具体硬件资源限制尽可能开辟多个缓冲来减小“跳帧”频率，并且通过强行插入 I 帧来修正“跳帧”对编码质量的影响。

4.2.2 稳定性

采集端按照自己的节拍周期性地更新自身状态，不会主动与流水线同步。内核在下采样 DMA 开启之前，通过查询状态结构保证源缓冲在当时是稳定的。如果流水线周期开启后采集 DMA 的目的起始地址恰好变成下采样 DMA 的源起始地址，则有可能因为下采样 DMA 速度较慢而发生数据冲突。但采集 DMA 的频率受限于 PAL 制式规定的采样率(每秒 25 帧)，下采样 DMA 是从存储器到存储器的 MDMA，在内核系统时钟比为 2:1 时，有效带宽利用率能达到系统时钟率的 $2/3$ ^[5]，远快于以上限制，因此，不会发生上述问题。

5 结束语

在嵌入式平台上实现实时视频编码面临的主要问题是如何利用有限的硬件资源提高编码速度。一个合理的控制架构应使处理器外设与内核并行工作，使控制带来的开销保持最低。本文针对 H.264 编码算法在实时视频通信领域的应用，提出了基于查询和乒乓缓冲的控制架构及改进的双级控制架构，实验数据表明其可以满足视频通信在实时性和稳定性方面的要求。本系统已成功应用于某视频压缩类的工程项目中。

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.
- [2] 向一君, 朱秀昌. 基于 Blackfin-DSP 视频编码的系统优化与多核同步[J]. 电子工程师, 2006, 32(2): 40-43.
- [3] 贺军. 基于 TSSA 软件架构的视频压缩编码系统[J]. 数字电视与数字视频, 2003, (8): 12-14.
- [4] Analog Device Inc. ADSP-BF561 Blackfin Embedded Symmetric Multiprocessor Data Sheet[EB/OL]. [2007-05-10]. <http://www.analog.com>.
- [5] Analog Device Inc. ADSP-BF561 Blackfin Processor Hardware Reference[EB/OL]. [2007-06-10]. <http://www.analog.com>.
- [6] 陈峰. Blackfin 系列 DSP 原理与系统设计[M]. 北京: 电子工业出版社, 2004.