

IP-блок упаковки кодов переменной длины

И.А. Беляев

ОАО НПЦ «ЭЛВИС», vanes@elvees.com

Аннотация — Во многих алгоритмах сжатия данных на последнем этапе используются коды переменной длины (variable-length codes, VLC). Для формирования потока сжатых данных, образованного такими кодами, необходима их упаковка в слова фиксированной длины. Предлагаемый IP-блок осуществляет упаковку кодов переменной длины до 32 бит включительно в 32-битные слова фиксированной длины. Преимуществом предлагаемого IP-блока является ряд программно-вызываемых функций, повышающих эффективность работы с памятью и упрощающих для процессора системы на кристалле решение задачи формирования потока сжатых данных. IP-блок реализован в виде RTL-модели на языке SystemVerilog. При логическом синтезе IP-блока на библиотеке с проектными нормами 40 нм были получены площадь 7,8К вентилей ($5,5 \cdot 10^{-3} \text{ мм}^2$) и рабочая частота 500 МГц.

Ключевые слова — IP-блок, сжатие данных, коды переменной длины, упаковка, VLC, система на кристалле (SoC).

1. ВВЕДЕНИЕ

В современном информационном обществе сжатие цифровой информации с целью хранения и передачи является насущной необходимостью. Особенно это актуально в отношении фото- и видеоинформации, которая является наиболее важной для человека, так как составляет большую часть его информации об окружающем мире, и в то же время является наиболее затратной с точки зрения хранения и передачи.

Среди многочисленных стандартов сжатия данных многие и наиболее популярные (JPEG, MPEG4, H.264/AVC, H.265/HEVC, VP8, VP9) используют на последнем этапе алгоритма сжатия процедуру кодирования кодами переменной длины. При этом различным входным элементам ставятся в соответствие коды, состоящие из различного числа бит. Идущие один за другим эти коды образуют поток сжатых данных. Для формирования потока сжатых данных, а также его хранения и передачи, необходима упаковка кодов переменной длины в слова фиксированной длины, соответствующей ширине памяти (рис. 1).

Формирование потока сжатых данных в памяти в зависимости от устройства, которое его осуществляет, может происходить тремя способами (рис. 2):

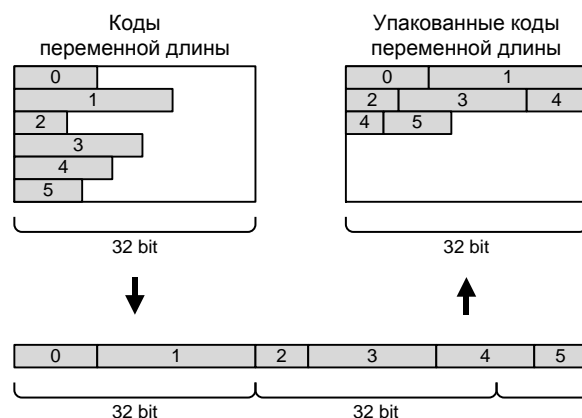


Рис. 1. Упаковка кодов переменной длины в 32-битные слова постоянной длины

- 1) Полностью программно. В этом случае поток сжатых данных в памяти формируется процессором общего назначения. Возможна максимальная гибкость при выборе параметров сжатия данных, однако скорость сжатия невысока. Данный способ подходит для задач сжатия с невысокой вычислительной сложностью или при отсутствии требования сжатия в реальном времени.
- 2) Полностью аппаратно. В этом случае поток сжатых данных в памяти формируется аппаратным ускорителем. Скорость сжатия максимальна по сравнению с двумя остальными способами, однако отсутствует гибкость при выборе параметров кодирования. Данный способ подходит для решения однотипных задач сжатия с высокой вычислительной сложностью в реальном времени (камеры видеонаблюдения с поддержкой сжатия).
- 3) Программно-аппаратно. В этом случае процессор общего назначения формирует в памяти только нетривиальные участки потока сжатых данных, оставляя большую часть работы аппаратному ускорителю. Скорость сжатия близка к полностью аппаратному способу, и при этом существует достаточная гибкость в выборе параметров кодирования. Данный способ подходит для решения широкого круга задач сжатия с высокой вычислительной сложностью в реальном времени (мультимедийные процессоры, видеоаналитика).

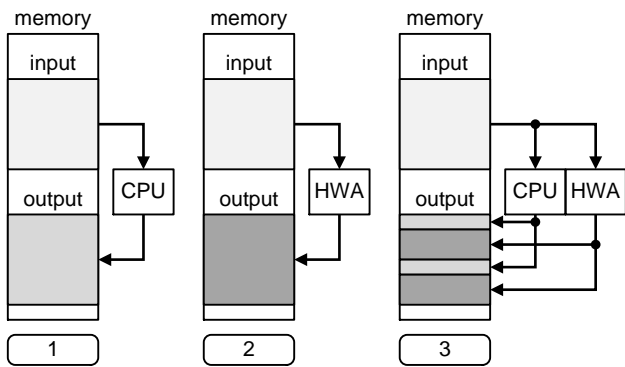


Рис. 2. Способы формирования потока сжатых данных в памяти

Предлагаемый IP-блок предназначен для третьего способа формирования потока сжатых данных, т.е. его архитектура и набор возможностей ориентированы на эффективную совместную работу аппаратного ускорителя и процессора общего назначения. Необходимым условием эффективности в данном случае является получение на выходе IP-блока полностью готового к сохранению в памяти или передаче в сеть потока сжатых данных, т.е. потока, для которого не потребуется последующая дополнительная обработка процессором общего назначения.

Можно выделить две большие проблемы, которые приводят к неэффективной совместной работе аппаратного ускорителя и процессора общего назначения:

- 1) Неспособность аппаратного ускорителя продолжать поток сжатых данных, созданный в памяти процессором. В этом случае процессору придётся «склеивать» два потока данных, что приведёт к потере производительности и значительному количеству дополнительных обращений к памяти (рис. 3).
- 2) Отсутствие аппаратного байт-стаффинга. В этом случае процессору придётся проверить каждый байт потока сжатых данных, что также приведёт к потере производительности и значительному количеству дополнительных обращений к памяти.

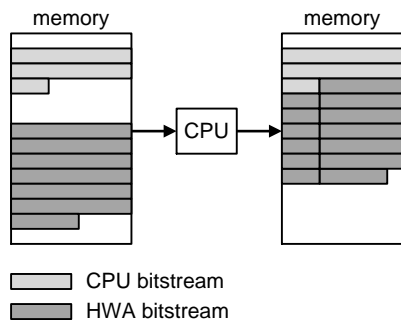


Рис. 3. Соединение процессором в памяти двух потоков сжатых данных

Предлагаемый IP-блок решает приведённые выше проблемы, а также имеет ряд дополнительных возможностей для эффективного взаимодействия аппаратного ускорителя и процессора общего назначения.

II. ФУНКЦИОНАЛЬНОСТЬ

При достаточно большом количестве работ, посвящённых разработке IP-блоков сжатия данных, функциональность и архитектурные особенности блоков упаковки кодов переменной длины приводятся лишь в немногих ([1]-[7]). Так, подробно архитектура ядра блоков упаковки кодов переменной длины, состоящего в основном из буферов и сдвиговых регистров, приводится в работах [2], [3] и [6]. В работах [4] и [5], посвящённых сжатию видеоданных по стандарту H.264/AVC, приводится архитектура блоков байт-стаффинга, а также затрагивается проблема добавления заголовков в поток сжатых данных.

Основным отличием предлагаемого IP-блока от аналогов является ряд программно-вызываемых функций, которые обеспечивают более широкий круг решаемых задач, и которые являются необходимыми для эффективного использования аппаратного ускорителя совместно с процессором. Предлагаемый IP-блок позволяет решать следующие задачи, связанные с упаковкой кодов переменной длины:

- упаковка кодов переменной длины в слова фиксированной длины;
- байт-стаффинг потоков H.264 и JPEG (возможно расширение для поддержки байт-стаффинга других потоков);
- выравнивание потока сжатых данных по границе байта (дописывание потока нулями или единицами до достижения ближайшей границы байта);
- вставка в поток сжатых данных произвольной информации (заголовков, маркеров и т.п.);
- продолжение («дописывание») потока сжатых данных, созданного в памяти другим устройством кодирования;
- выгрузка в память последнего неполного слова фиксированной длины;
- получение информации об общей длине потока сжатых данных.

Таким образом, предлагаемый IP-блок является достаточно сложным устройством и предназначен для использования в мультиформатных аппаратных кодеках в составе системы на кристалле, в которой возможно программное управление со стороны процессора, и в которой кодирование нетривиальных частей потока типа заголовков и маркеров происходит программно.

III. АРХИТЕКТУРА

IP-блок состоит из трёх подблоков (рис. 4):

- CONTROL – блок управления;
- BITPACKER – блок упаковки;
- BYTESTUFFER – блок байт-стаффинга.

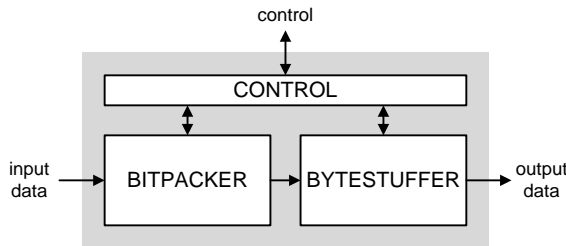


Рис. 4. Структура IP-блока упаковки кодов переменной длины

A. Блок управления (CONTROL)

Блок управления (CONTROL) позволяет осуществлять настройку IP-блока на решение определённой задачи. Управление IP-блоком осуществляется посредством записи и чтения его программно-доступных регистров.

Все задачи, решаемые IP-блоком, можно разделить на две группы: активные и фоновые. К фоновым задачам относятся упаковка кодов переменной длины и байт-стаффинг, к активным – все остальные.

При решении фоновой задачи нет необходимости в управлении IP-блоком со стороны программы. После однократной настройки он взаимодействует только с кодером, выдающим коды переменной длины, и выходным буфером, принимающим готовый поток.

Для решения активной задачи необходим контроль над IP-блоком со стороны программы, который заключается, во-первых, в настройке IP-блока, а во-вторых, в проверке его состояния, т.е. готовности решения. Активная задача выполняется однократно, после чего IP-блок готов к решению новой задачи.

B. Блок упаковки (BITPACKER)

Блок упаковки (BITPACKER) упаковывает коды переменной длины (длиной до 32 бит включительно) в слова фиксированной длины 32 бита.

Основной архитектуры блока BITPACKER являются кольцевой 64-битный буфер потока сжатых данных (CP0, CP1) и 64-битный сдвиговый регистр со скользящим окном в 32-бита (рис. 5). Такая архитектура отличается от [2], [3] и [6] меньшим количеством сдвиговых регистров, что позволяет экономить площадь (например, площадь [2] составляет около 2,6К вентилей при 24-битном входе и выходе, в то время как площадь блока BITPACKER составляет 2,2К вентилей при 32-битном входе и выходе).

В процессе работы блока BITPACKER коды переменной длины упаковываются один за другим в кольцевом 64-битном буфере. Как только половина кольцевого буфера оказывается заполненной, она выгружается во внешний буфер или в память.

Схема работы блока BITPACKER наглядно представлена на рис. 6.

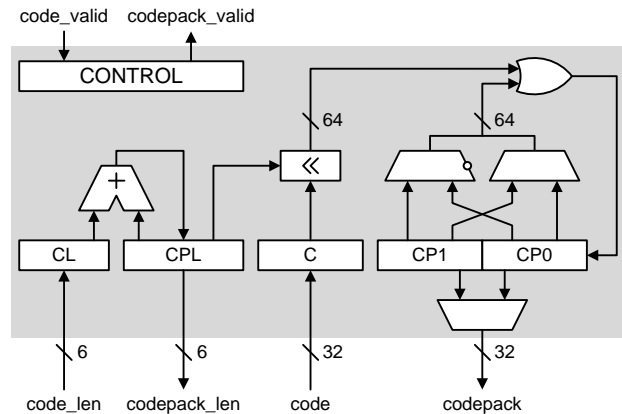


Рис. 5. Структура блока упаковки кодов переменной длины (BITPACKER)

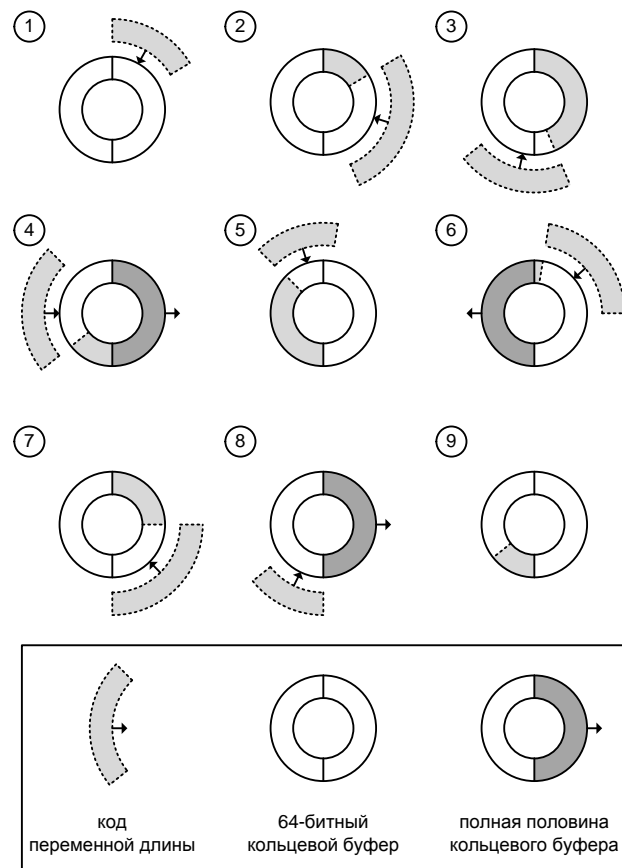


Рис. 6. Схема работы блока BITPACKER

С. Блок байт-стаффинга (BYTESTUFFER)

Блок байт-стаффинга (BYTESTUFFER) проверяет поток сжатых данных на наличие запрещённых последовательностей байт, и в случае обнаружения таковых вставляет в поток специальные проставочные байты.

В предлагаемом IP-блоке возможно включение трёх режимов байт-стаффинга:

- отсутствие байт-стаффинга;
- байт-стаффинг по стандарту H.264;
- байт-стаффинг по стандарту JPEG.

Байт-стаффинг осуществляется параллельно для 32 бит (4 байт) потока, выдаваемых блоком VTPACKER. Это аналогично [4] и, в отличие от [5], обеспечивает высокую скорость работы и отсутствие блокировок блока упаковки не успевающим за ним блоком байт-стаффинга.

IV. РЕАЛИЗАЦИЯ И АППАРАТНЫЕ ЗАТРАТЫ

Предлагаемая архитектура IP-блока была реализована в виде RTL-модели на языке SystemVerilog. Модель для верификации IP-блока была также написана на языке SystemVerilog. Верификация IP-блока проводилась в среде Cadence IUS с использованием случайных тестовых последовательностей, что позволило достичь почти 100% тестового покрытия.

Логический синтез RTL-модели проводился с помощью Synopsys DesignCompiler на библиотеке элементов с проектными нормами 40 нм. В результате логического синтеза были получены площадь IP-блока 7,8К вентилей ($5,5 \cdot 10^{-3}$ мм²) и рабочая частота 500 МГц. Полученные характеристики IP-блока и их сравнение с аналогами приведены в табл. 1.

Таблица 1

Аппаратные затраты и производительность IP-блока

| | [2] | [4] | [5] | IP-блок |
|--|------|------|-----|---------|
| Технологический процесс (нм) | – | 180 | – | 40 |
| Рабочая частота (МГц) | – | 100 | 100 | 500 |
| Площадь (кол-во вентилей) | 2,6К | 5,0К | – | 7,8К |
| Пиковая производительность (байт/такт) | 3 | 4 | 1 | 4 |
| Байт-стаффинг | – | + | + | + |
| Мультиформатность | + | – | – | + |
| Программная управляемость | – | – | – | + |

V. ВЫВОДЫ

Предлагаемый IP-блок осуществляет упаковку кодов переменной длины до 32 бит включительно в 32-битные слова фиксированной длины. Достоинством предлагаемого IP-блока является набор программно-вызываемых функций, которые позволяют формировать поток сжатых данных, полностью готовый к выгрузке во внешнюю память или отправке в сеть. Это повышает эффективность работы с памятью системы на кристалле и упрощает для процессора решение задачи формирования потока сжатых данных. Были получены площадь IP-блока 7,8К вентилей и рабочая частота 500 МГц.

ЛИТЕРАТУРА

- [1] E. Sahin, I. Hamzaoglu. An Efficient Hardware Architecture for H.264 Intra Prediction Algorithm // Design, Automation and Test in Europe (DATE) Conference. 2007.
- [2] A VLSI architecture design of VLC encoder for high data rate video/image coding / Hao-Chieh Chang, Liang-Gee Chen, Y. Chang, Sheng-Chieh Huang // IEEE International Symposium on Circuits and Systems. 1999. Vol. 4. P. 398-401.
- [3] Shaw-Min Lei, Ming-Ting Sun. An entropy coding system for digital HDTV applications // IEEE Transactions on Circuits and Systems for Video Technology. 1991. Vol. 1. Issue 1. P. 147-155.
- [4] Dual-block-pipelined VLSI architecture of entropy coding for H.264 baseline profile / Tung-Chien Chen, Yu-Wen Huang, Chuan-Yong Tsai, Bing-Yu Hsieh, Liang-Gee Chen // Proc. International Symposium on VLSI Design, Automation and Test (VLSI-DAT). 2005. P. 271–274.
- [5] Hardware implementation for entropy coding and byte stream packing engine in H.264/AVC / Ngoc-Mai Nguyen, Edith Beigne, Suzanne Lesecq, Pascal Vivet, Duy-Hieu Bui, Xuan-Tu Tran // International Conference on Advanced Technologies for Communications (ATC). 2013. P. 360-365.
- [6] Hongqi Hu, Jingnan Sun, Jiadong Xu. High Performance Architecture Design of CAVLC Encoder in H.264/AVC // Congress on Image and Signal Processing. 2008. Vol. 1. P. 613-616.
- [7] Pei-Yin Chen, Yi-Ming Lin. Low-Cost CAVLC Encoder // IEICE Transactions on Electronics. 2006. Vol. E89-C. No. 12. P. 1950-1953.
- [8] ITU-T Recommendation H.264 // ISO/IEC 14496-10 (03.2010).
- [9] ITU Recommendation T.81 // ISO/IEC 10918-1 : 1993(E).