

**Сергиенко Анатолий Михайлович**<sup>1)</sup>, д.т.н.  
**Молчанов Алексей Андреевич**<sup>1)</sup>, аспирант  
**Сергиенко Павел Анатольевич**<sup>1)</sup>, магистр  
**Мозговой Иван Владиславович**<sup>1)</sup>,

<sup>1)</sup>Киевский политехнический институт им. И. Сикорского,  
(37, просп. Перемоги, 03056, Киев-56, Украина  
Тел. (044) 2049337, e-mail aser@comsys.kpi.ua)

## **Система динамической визуализации на базе аппаратного GIF-декомпрессора**

Предложена система динамической индикации для систем имитационного моделирования и управления на основе распаковки GIF-файлов в реальном времени. Такая распаковка выполняется в аппаратном декомпрессоре, реализованном в RISC-процессоре с системой команд, адаптированной к задачам декомпрессии файлов и сконфигурированном в ПЛИС.

*Ключевые слова:* GIF, ПЛИС, система реального времени, дисплей

**Введение.** При реализации систем имитационного моделирования, управления сложными объектами применяются средства отображения информации на основе LCD-дисплеев и компьютеров общего назначения. При этом средствами машинной графики генерируются изображения измерительных приборов, которые отображают реальные значения сигналов, обстановки объектов. Широко используются специальные программные средства для отображения, такие как Labview [1]. Однако реализация отображения в реальном времени требует значительных вычислительных затрат с использованием графического акселератора и нередко имеет чрезмерную латентную задержку. Кроме того, разработка программ, генерирующих такие изображения, является трудозатратной.

В работе предлагается генерировать динамически изменяемые изображения путем быстрой управляемой распаковки GIF-файлов. При этом высокая скорость распаковки обеспечивается путем аппаратно-программной реализации декомпрессора в ПЛИС.

**Принцип динамической визуализации с помощью GIF-файлов.** Файл изображения в формате GIF обеспечивает хранение растровых графических изображений без потерь с использованием алгоритма сжатия LZW. Причем в одном файле могут храниться несколько различных фреймов изображений разной величины со своими цветовыми палитрами, которые отображаются в разных областях экрана дисплея. Важным является то, что фрейм имеет признак прозрачности пиксела, обеспечивающий наложение нескольких фреймов друг на друга. Примером использования такого файла является GIF-анимация в Web-приложениях.

Динамическая визуализация информации с помощью GIF-файлов состоит в том, что отображаемые фреймы выбираются для распаковки в порядке, который определяется алгоритмом отображения. Например, в одном фрейме может храниться шкала стрелочного измерительного прибора, а в других фреймах – различные положения его стрелки. Тогда отображение стрелки сводится к выбору фрейма с ее изображением в зависимости от значения отображаемого параметра и помещении его

поверх фрейма шкалы. Однако такой режим распаковки GIF-файла не предусмотрен его стандартом и не реализуется никаким известным программным распаковщиком. До последнего времени, пока не истек срок действия патента на алгоритм LZW, практически не было внедрений нестандартного применения GIF-файлов. Подготовка фреймов и формирование GIF-файла могут выполняться с помощью существующих средств разработки GIF-анимации.

**Структура системы динамической визуализации.** Система включает в себя центральный процессорный элемент (ЦПЭ) с RISC-архитектурой, ОЗУ данных и ПЗУ, в котором хранятся распаковываемые GIF-файлы. Часто фоном для визуализации данных является видеоизображение, например, снимаемое видеокамерой. Тогда распаковываемые фреймы должны накладываться на видеокادر с частотой кадров в блоке обработки кадра (рис.1).

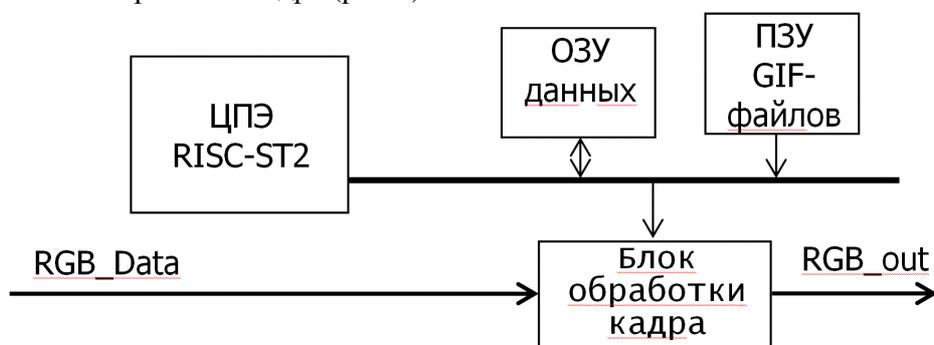


Рис.1. Структура системы

Этот блок содержит буфер распакованных фреймов, таблицу палитры, заполняемые из распакованного файла, генератор адресов развертки и смеситель с данными видеокadra RGB\_Data.

**Процессор распаковки GIF-файлов.** Для выполнения распаковки GIF-файлов разработан специализированный 16-разрядный RISC-процессор. За его основу было взято ядро RISC-ST, описанное в [2]. Длина команды увеличена до 18 разрядов, что позволило добавлять новые команды, оптимизировать эффективность их реализации и лучше задействовать встроенную память ПЛИС.

Благодаря тому, что современные ПЛИС имеют шестизрядные логические таблицы (ЛТ), имеется возможность добавлять в архитектуру микропроцессорного ядра различные команды, несущественно увеличивая аппаратные затраты и период тактового интервала. Так, для ускорения алгоритма распаковки к ядру добавлены команды обработки отдельных битов слов, выделения битовых полей заданной длины, слияния полей, сдвига слов, подсчета числа нулевых старших разрядов. Для реализации быстрого доступа к ассоциативной таблице словаря введена команда вычисления хэш-функции.

Регистровая память процессора имеет 32 регистра. Используются такие виды адресации, как регистровая, базовая, индексная с прединкрементом. Адресно

доступны до 256 периферийных регистров. Память данных разбита на страницы по 4096 байт и имеет максимальную емкость 256 мегабайт. Этого достаточно, чтобы эффективно адресовать пространство кадров изображения. Для доступа к памяти данных используется стандартный открытый интерфейс Wishbone. Подробнее архитектура процессора описана в [3].

**Экспериментальные результаты.** Микропроцессорное ядро RISC-ST2 вместе с блоком памяти данных и блоком памяти программ описано на языке VHDL. При его размещении в ПЛИС Kintex-7 фирмы Xilinx оно занимает всего 878 ЛТ и имеет максимальную тактовую частоту более 200 МГц. Таким образом, это ядро имеет небольшие аппаратные затраты — столько, сколько в ПЛИС приходится затрат на 2–3 блока памяти объемом по 4 килобайта. Эти затраты в семь раз меньше, чем у распространенного 32-разрядного ядра MicroBlaze [4]. Для создания для него матобеспечения был разработан кроссассемблер.

Запрограммированный распаковщик GIF-файлов на декомпрессию одного пиксела изображения тратит, в среднем, 21 такт. При тактовой частоте микропроцессорного ядра 100 МГц распаковка выполняется со скоростью 4,8 млн. байт/с.

При выполнении алгоритма LZW наибольшую задержку имеет чтение строки из словаря и перепись ее в буфер изображения. Эту долю времени можно сократить, если добавить аппаратно реализованную функцию формирования и пересылки распакованной строки. Тогда скорость распаковки возрастает до 8,5 млн. байт/с. Для достижения еще большей скорости декомпрессии в работе [3] предлагается создать многопроцессорную систему на базе конфигурируемых RISC-процессоров, которые параллельно выполняют распаковку независимых блоков данных.

**Выводы.** Динамическая визуализация на базе аппаратно-программного GIF-декомпрессора обеспечивает высокое быстродействие и эстетическое качество отображаемой информации. Для ее реализации был разработан экспериментальный процессор распаковки GIF-файлов, размещенный в ПЛИС, обеспечивающий формирование изображения со скоростью около 5 млн. пикселей в секунду. Система динамической визуализации на его основе может найти применение в системах имитационного моделирования и управления, которые требуют быстрой реакции.

1. Ponce-Cruz C., Ramirez-Figueroa F.D. Intelligent Control Systems with LabVIEW. – Springer, 2010. – 215 p.
2. Сергиенко А.М. VHDL для проектирования вычислительных устройств. –К.: –"ДиаСофт". 2003. –210 с.
3. Сергиенко А.М., Кліменко І.А., Сергієнко П.А. Реконфігурована багатопроцесорна обчислювальна система на ПЛІС. Вісник НТУУ «КПІ». Ін форматика, управління та обчислювальна техніка: Зб. наук. пр. – К.:Век+, 64, 2016. – С. 47-50.
4. Meyer-Baese U. Digital Signal Processing with Field Programmable Gate Arrays. 4-th Ed. – Springer. – 2014. – 930 p.