

Обработка информации с помощью VGA адаптера для FPGA камеры

Х.А. Джабраилов¹, В.С. Селезнев², М.Ю. Абдулханова², Е.О. Антонова²,

И.А. Джабраилов²

¹Московский технический университет связи и информатики

²Московский автомобильно-дорожный государственный технический университет

Аннотация: В данной статье описывается первый этап выполнения научно-исследовательской работы по разработке камеры на базе FPGA для задач идентификации транспортных средств, широко распространённых в автоматизированных пунктах весогабаритного контроля. Поскольку FPGA является альтернативой обычных процессоров, в особенности которой входит возможность выполнение нескольких задач параллельно, то камера оборудованная FPGA сможет выполнять функции обнаружения и идентификации транспортных средств одновременно. Таким образом, камера будет осуществлять не только передачу изображения, но и передачу результата обработки для проблемно-ориентированных систем управления, принятия решений и оптимизации обработки потока данных, после чего серверу будет необходимо только подтвердить или опровергнуть результаты работы камеры, что значительно снизит время обработки изображений со всех автоматизированных пунктов весогабаритного контроля. В ходе разработки была реализована простая плата VGA порта, программа статического изображения для вывода его на монитор в разрешении 640x480, программа счётчика пикселей. В качестве FPGA используется EP4CE6E22C8, мощность которого для достижения результата более чем достаточно.

Ключевые слова: Методы системного анализа, оптимизация, FPGA, VGA адаптер, Verilog, камера для распознавания, разработка платы, обработка информации, статистика.

Введение

В дорожной отрасли на примере автоматизированных пунктов весового и габаритного контроля, можно отметить, что используются обычные камеры фото-видео фиксации вместо «умных» камер. Все этапы обработки изображений, полученных с камер, включающие в себя определение регистрационного номера транспортного средства, идентификация

транспортного средства по классификации EUR 13 и прочие осуществляется на серверах центров обработки информации, а также в центрах организации дорожного движения. Ввиду чего, задача разработки принципиально новых методов анализа и синтеза элементов систем управления с целью улучшения их технических характеристик является актуальной. Применение камер, оборудованных программируемой логической интегральной схемой (ПЛИС), позволит снизить объем вычислительных операций сервера, снизив нагрузку на него путем реализации первичной обработки изображения непосредственно на самой камере [1]. Последние поколения ПЛИС именуемые FPGA (Field-Programmable Gate Array) имеют большой потенциал для выполнения различных методов обработки изображения, таких, как пороговая или адаптивная бинаризация и другие [2].

Постановка задачи исследования

Поскольку для задачи идентификации транспортных средств по классификации EUR 13, необходимо выполнить все этапы обработки изображения, такие как предобработка, сегментация, детектирование и постобработка на ПЛИС, то первый этап в разработке камеры, оборудованной ПЛИС, заключается в реализации контроллера VGA на FPGA.

Для визуализации процесса обработки изображения используется монитор VGA, который представляет собой матрицу пикселей. Разрешение экрана составляет 640 горизонтальных пикселей на 480 вертикальных. Монитор работает по принципу последовательной передачи информации о каждом пикселе в формате RGB, где каждый цвет (красный, зеленый, синий) передается в виде аналогового сигнала. Поскольку монитор не оборудован видеопамью и не может запомнить информацию о каждом пикселе, то требуется непрерывный поток информации на монитор для сохранения

изображения. Частоты синхронизаций стандарта VGA представлены в таблице 1.

Таблица № 1

Частоты синхронизаций стандарта VGA

Название	Частота
Вертикальная	60 Гц
Горизонтальная	31.46875 КГц
Пиксельная	25.175 МГц

Решение задачи

Отладочная плата, на которой происходит реализация проекта, оборудована кварцевым резонатором с частотой колебания 50 МГц. Отсюда возникает первая цель: выполнить делитель частоты для получения пиксельной частоты. Исходя из значения опорной частоты, чтобы получить значения пиксельной частоты, равной 25.175 МГц, необходимо ее умножить на коэффициент деления равный 0.5035. Коэффициент можно представить в виде дроби, где в числителе 5035, а в знаменателе 10000. Сократив данную дробь, в числителе получается 1007, а в знаменателе 2000. Поскольку необходим меандр, то длительность активного уровня, должен быть равен длительности паузы между отдельными импульсами. Часть кода на языке VHDL, отвечающий за реализацию делителя частоты, представлен на рис. 1.

```
clk0_divide_by => 1,  
clk0_duty_cycle => 50,  
clk0_multiply_by => 2,  
clk0_phase_shift => "0",
```

Рис. 1. – Код делителя частоты

Следующая задача заключается в получении из пиксельной частоты горизонтальную и вертикальную развёртки. Временные задержки горизонтальной и вертикальной развёрток представлены в таблицах 2 и 3.

Таблица № 2

Горизонтальные задержки

Участок пикселей	Количество пикселей	Время
Горизонтальная видимой участок	640	25.422045680238 мкс
Время переднего фронта	16	0.63555114200596 мкс
Синхронизирующий импульс	96	3.8133068520357 мкс
Время заднего фронта	48	1.9066534260179 мкс

Таблица № 3

Вертикальные задержки

Участок пикселей	Количество пикселей	Время
Горизонтальная видимой участок	480	15.253227408143 мс
Время переднего фронта	10	0.31777557100298 мс
Синхронизирующий импульс	2	0.063555114200596 мс
Время заднего фронта	33	1.0486593843098 мс

Таким образом, суммарное количество пикселей при горизонтальной развёртке должно соответствовать 800 за временной интервал в 31.777557100298 мкс, а при вертикальной 525 пикселей за 16.683217477656 мс [3, 4]. Для упрощения задачи горизонтальные пиксели можно представить

в виде оси абсцисса, а вертикальные – в виде оси ординат. Часть кода на языке Verilog для задания счета пикселей представлена на рис. 2.

```
wire CounterXmaxed = (CounterX == 800); // 16 + 48 + 96 + 640  
wire CounterYmaxed = (CounterY == 525); // 10 + 2 + 33 + 480
```

Рис. 2. – Код счетчика пикселей

Провести проверку работоспособности программ можно, подключив осциллограф к выводным контактам (пинам) горизонтальной и вертикальной развертки на плате, для оценки частотных характеристик. Результаты измерения представлены на рис. 3 – 4.

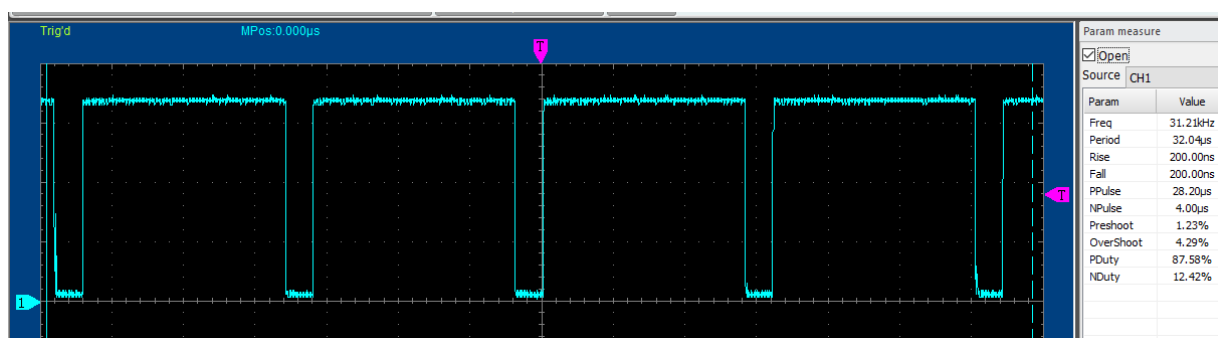


Рис. 3. – Горизонтальная развертка

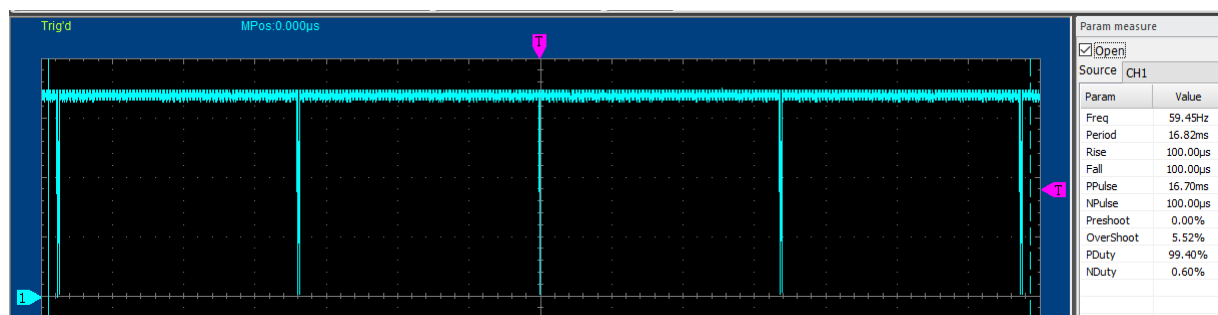


Рис. 4. – Вертикальная развертка

Исходя из показаний осциллографа, получается, что вертикальная и горизонтальная развертка не попадает в допустимый диапазон частот. Следовательно, для проверки работоспособности программы требуется более точный частотомер или же нужно генерировать изображение с последующей передачей его непосредственно на монитор.

Исходя из этого возникает последняя задача, заключающаяся в генерации статического изображения на борту FPGA с последующей передачей цветовой схемы на монитор. Поскольку по стандарту VGA передача цвета осуществляется аналоговым сигналом, а FPGA работает на цифровых сигналах, необходимо разработать следующую электрическую схему, с помощью которой можно из параллельных цифровых сигналов получать значения аналоговых, необходимых для работы VGA монитора [5, 6]. Электрическая схема представлена на рис. 5, компьютерная модель на рис. 6.

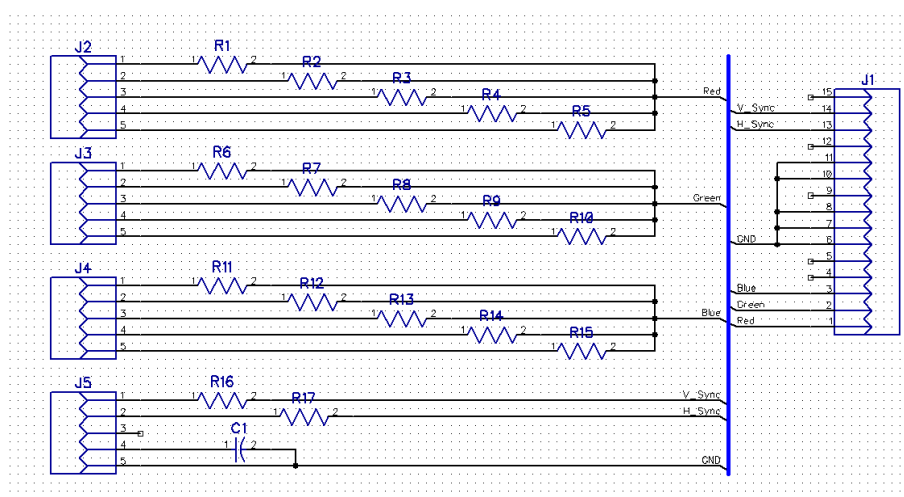


Рис. 5. – Электрическая схем порта VGA

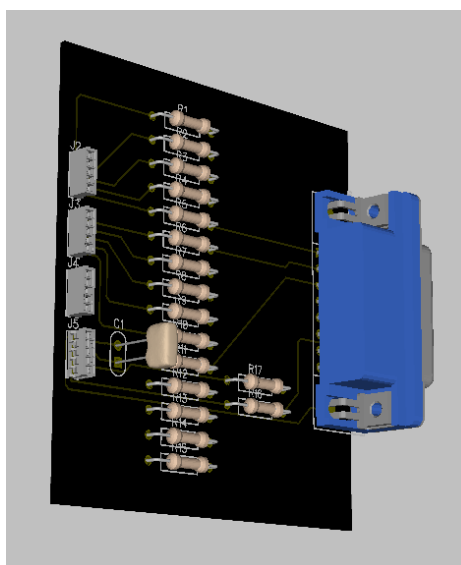


Рис. 6. – Компьютерная модель порта VGA

Теперь необходимо генерировать статическое изображение в виде трех квадратов различных цветов. Поскольку монитор VGA использует аддитивную цветовую модуль RGB, то необходимо определить области этих квадратов. Поскольку известно, что видимая область изображения составляет 640 пикселей по горизонтали и 480 по вертикали, то можно рассчитать вершины трех квадратов так, чтобы в результате получились координаты начала квадрата для каждого цвета и его конца. При реализации электрической схемы, представленной на рисунке 5, используется цветовая передача RGB555, что в свою очередь обозначает, что 5 бит передают значение одного цвета [7]. Реализация программы статического изображения представлен на рис. 7 и результат работы программы на рис. 8.

```
module static_image(  
    input wire CLK, // pixel frequency: 25.175 MHz  
    output wire [4:0] VGA_R, // 5-bit VGA red output  
    output wire [4:0] VGA_G, // 5-bit VGA green output  
    output wire [4:0] VGA_B // 5-bit VGA blue output  
);  
  
    wire A, B, C;  
  
    assign A = ((x > 140) & (y > 60) & (x < 300) & (y < 220)) ? 1 : 0;  
    assign B = ((x > 200) & (y > 120) & (x < 360) & (y < 280)) ? 1 : 0;  
    assign C = ((x > 260) & (y > 180) & (x < 420) & (y < 340)) ? 1 : 0;  
  
    assign VGA_R[4] = sq_A; // Area A is Red  
    assign VGA_G[4] = sq_B; // Area B is Green  
    assign VGA_B[4] = sq_C; // Area C is Blue  
  
endmodule
```

Рис. 7. – Программа статического изображения

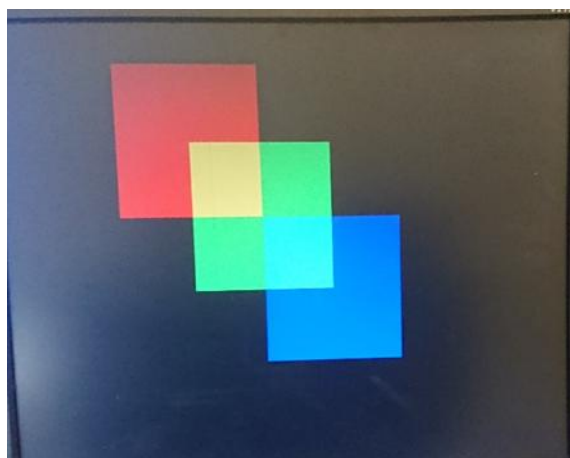


Рис. 8. – Результат работы программы

Проектирование устройства производится с помощью программного обеспечения Quartus компании Intel. Для облегчения программы в качестве основного делителя частоты лучше использовать PLL (Phase-Locked Loop) [8, 9]. Таким образом, объем используемых логических элементов снизится на порядок и составит всего 91 логический элемент, что составляет около 1% от общего числа логических элементов FPGA Cyclone IV E (EP4CE6) [10, 11].

Основные выводы

Проведя первый этап выполнения научно-исследовательской работы по разработке камеры на базе FPGA для задач идентификации транспортных средств, заключающийся в реализации платы VGA порта и написании программы для вывода статического изображения, а также написании программы счетчика пикселей, была выявлена следующая особенность подключения ПЛИС к VGA монитору: так как монитор не оснащен встроенной видеопамятью, то возникает необходимость в непрерывной подаче информационного потока, чтобы поддерживать изображение.

Для получения необходимой пиксельной частоты со значением 25.175 МГц, был написан код, описывающий делитель опорной частоты с коэффициентом 0.5035. Затем из полученной частоты были выделены горизонтальная и вертикальная развёрстки с суммарным количеством пикселей 800 и 525 соответственно.

На последнем этапе работы была разработана электрическая схема, преобразующая параллельный цифровой сигнал в аналоговый, который необходим для работы VGA монитора и вывода статического изображения.

Данная научно-исследовательская работа проводилась при финансовой поддержке Фонда содействия развитию малых форм предприятий в научно-технической сфере в рамках программы «УМНИК» в соответствии с исследовательским проектом № 15411ГУ/2020.

Литература

1. Палагин А В, Яковлев Ю С, Елисеева Е В Особенности подхода к выбору ПЛИС для проектирования PIM-систем // Математические машины и системы. 2012. № 3. С. 19-28.
2. Илюхин А В, Зарипова И И, Марсова Е В, Селезнёв В Устройства цифровой автоматики. Москва: (Московский автомобильно-дорожный государственный технический университет (МАДИ)). 2018. 110 с.
3. Ивашинников Б А, Коваленко М В, Лесков О М Реализация VGA-интерфейса на базе FPGA фирмы ALTERA // Общество с ограниченной ответственностью "Издательство Молодой ученый». М. 2016. № 17. 121 с.
4. Строгонов А В, Бабич Д А 2012 Разработка контроллера VGA на языке VHDL в среде QUARTUS // II Вестник Воронежского государственного технического университета. 2012. №6. С. 20-26.
5. Гарбар Е.А., Николаев А.А. Обработка информации при экспресс оценке наличия дефекта поверхности оцинкованного проката // Математическое и программное обеспечение систем в промышленной и социальной сферах. 2020. Т. 8. № 1. С. 23-30.
6. Чекушкин Е.А. Трековый процессор детектора снд. В книге: Студент и научно-технический прогресс // Материалы XLV Международной научной студенческой конференции "Студент и научно-технический прогресс". 2007. С. 145-146.
7. Столяров В.Я., Тихомиров В.А. Методика компьютерного выявления текстовых областей на изображении бирок, выполненных лазерной гравировкой на металле // Южно-Сибирский научный вестник. 2020. № 1 (29). С. 34-38.
8. Столяров В.Я. универсальный программный комплекс подготовки изображений идентификаторов перед распознаванием // Ученые записки

Комсомольского-на-Амуре государственного технического университета. 2020. № 5 (45). С. 78-86.

9. Страхов П.В., Бадасен Е.В., Шурыгин Б.М., Кондранин Т.В. Новый алгоритм геометрической коррекции изображений, получаемых авиационными сканерными системами, с помощью опорных точек без использования бортовых данных. // Сборник тезисов докладов Четырнадцатой Всероссийской открытой конференции "Современные проблемы дистанционного зондирования Земли из космоса". Электронный сборник тезисов докладов. 2016. С. 52-53. URL: conf.rse.geosmis.ru/thesisshow.aspx?page=133&thesis=5858

10. Ronneberger O., Fischer P., and Brox T., "U-Net: Convolutional Networks for Biomedical Image Segmentation," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2015. vol. 9351, pp. 234-241.

11. Cha Y.-J., Choi W., Büyükoztürk O., "Deep learning-based crack damage detection using convolutional neural networks," Computer-Aided Civil and Infrastructure Engineering. May 2017. Vol. 32(5), pp. 361-378.

References

1. Palagin A. V., Yakovlev Yu. S., Eliseeva E. V. Matematicheskie mashiny i sistemy. 2012. № 3. pp. 19-28.

2. Ilyukhin A. V., Zaripova I. I., Marsova E. V., Seleznev V. Ustroystva tsifrovoy avtomatiki [Digital automation devices]. Moskva: (Moskovskiy avtomobil'no-dorozhnyy gosudarstvennyy tekhnicheskij universitet). 2018. 110 p.

3. Ivashinnikov B. A., Kovalenko M. V., Leskov O. M. Obshchestvo s ogranichennoy otvetstvennost'yu "Izdatel'stvo Molodoy uchenyyu». M. 2016. № 17. 121 p.

4. Strogonov A. V., Babich D. A. II Vestnik Voronezhskogo gosudarstvennogo tekhnicheskogo universiteta. 2012. №6. pp. 20-26.



5. Garbar E.A., Nikolaev A.A. Matematicheskoe i programmnoe obespechenie sistem v promyshlennoy i sotsial'noy sferakh. 2020. T. 8. № 1. pp. 23-30.
6. Chekushkin E.A. Materialy XLV Mezhdunarodnoy nauchnoy studencheskoy konferentsii "Student i nauchno-tekhnicheskij progress". 2007. pp. 145-146.
7. Stolyarov V.Ya., Tikhomirov V.A. Yuzhno-Sibirskiy nauchnyy vestnik. 2020. № 1 (29). pp. 34-38.
8. Stolyarov V.Ya. Uchenye zapiski Komsomol'skogo-na-Amure gosudarstvennogo tekhnicheskogo universiteta. 2020. № 5 (45). pp. 78-86.
9. Strakhov P.V., Badasen E.V., Shurygin B.M., Kondranin T.V. Elektronnyy sbornik tezisov dokladov. 2016. pp. 52-53. URL: conf.rse.geosmis.ru/thesisshow.aspx?page=133&thesis=5858
10. Ronneberger O., Fischer P., and Brox T. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2015. pp. 234-241.
11. Cha Y.-J., Choi W. Büyüköztürk O. Computer-Aided Civil and Infrastructure Engineering. May 2017. pp. 361-378.