

УДК 004.932:004.383.3/5

АНАЛИЗ АППАРАТНЫХ ПЛАТФОРМ ДЛЯ ОБРАБОТКИ ИЗОБРАЖЕНИЙ

А. И. Секирин, Н. Ф. Вустяк

Донецкий национальный технический университет, Донецк

e-mail: krama59@yandex.ru

Аннотация

В статье рассматриваются вопросы эффективности применения различных аппаратных платформ для построения алгоритмов высокоскоростной обработки изображений. Рассматриваются платформы графических процессоров, платформы процессоров обработки сигналов и платформы программируемых логических интегральных схем. Для каждой платформы рассмотрены средства разработки программного обеспечения, что является неотъемлемой частью проектирования высокоскоростных алгоритмов обработки изображений и выбора платформы для реализации этих алгоритмов. Рациональный выбор платформы для высокоскоростной обработки изображений позволит значительно сократить время разработки и ресурсы готового изделия.

Введение

В последнее время широкое распространение получили методы анализа быстротекущих процессов при помощи высокоскоростной обработки видеоизображений [1]. Это стало возможным благодаря тому, что современные высокоскоростные цифровые камеры, основанные на матричных фотодиодных приемниках или приемниках на основе приборов с зарядовой связью (ПЗС), представляют собой сверхбольшие интегральные схемы с внутрикристалльным аналого-цифровым преобразователем (АЦП), что дает следующие преимущества:

- на выходе матриц формируется уже оцифрованный видеосигнал, причем матрицы могут быть как монохромные, так и цветные (RGB);
- наличие интегрированного АЦП в матрице значительно снижает шумы сигнальных цепей между аналоговой матрицей и самим АЦП;
- производитель матрицы уже оптимизировал тип используемого АЦП под решаемые задачи;
- применение таких матриц значительно упрощает разработку блока цифровой обработки видеоизображения.

Для реализации высокоскоростных алгоритмов цифровой обработки изображений требуются значительные вычислительные ресурсы. Обеспечение требуемых характеристик и быстродействия алгоритмов можно осуществить двумя способами: конструкторским (аппаратным) или алгоритмическим (программным). Учитывая специфичность портативного оборудования, как электронного изделия, для дальнейших разработок выбран

аппаратный способ реализации высокоскоростных алгоритмов обработки изображений.

Основной задачей выбора платформы для высокоскоростной обработки изображений является получение потока видеоизображения с помощью аппаратного блока, позволяющего производить ряд стандартных преобразований, использующего минимальное количество памяти и не имеющего существенной задержки выходного (преобразованного) изображения, при этом обладающего таким энергопотреблением, которое позволяет использовать данный блок в портативных (мобильных) устройствах.

Необходимость разработки аппаратного блока обработки видеоизображения обусловлена все возрастающей потребностью в использовании портативных систем, применяемых в мобильных устройствах, что свидетельствует об актуальности проводимых исследований.

Анализ платформы графических процессоров

Использование CPU (Central Processing Unit) для реализации высокоскоростных алгоритмов обработки изображений не предусматривается. Основные причины такого подхода можно сформулировать в нескольких тезисах. Если современные CPU имеют суперскалярную архитектуру и внеочередное исполнение команд (out-of-order execution), то GPU (Graphics Processing Unit) выполняют инструкции по порядку (in-order) и имеют VLIW-архитектуру (very long instruction word) с возможностью распараллеливания процессов, что значительно упрощает вычислительные блоки, повышает быстродействие (в разы) и снижает

потребление энергии. Более того, CPU имеют сравнительно короткие векторные команды, а широкие векторы в GPU их сглаживают.

Также CPU-ядра обычно оптимизированы для работы на высокой частоте с высокими токами утечек (leakage), а GPU оптимизированы для работы на низкой частоте и при низком напряжении питания с меньшими утечками. В общем, в отличие от CPU, большая часть энергии в случае расчетов на GPU тратится именно на вычисления.

Современные CPU могут выполнять загрузку двух 256-битных регистров из кэша L1 параллельно или 64 байта за такт. GPU обладает унифицированным блоком разделяемой памяти/кэша L1.

Производительность блока одинакова для двух архитектур CPU и GPU. Но, GPU в составе имеет несколько блоков параллельной обработки процессов, а CPU только от 1 до 4 на ядро. Отсюда следует, что CPU менее производительны в сравнении с GPU для реализации высокоскоростных алгоритмов обработки изображений и требуют больших затрат на быстродействие и энергопотребления.

На рынке графических процессоров (GPU), в основном, представлены производители Intel, NVIDIA и AMD. На основе этих чипов строятся различные видеокарты, выпускаются встраиваемые (SoC) в процессор видео процессоры. Не каждый графический процессор может использоваться для пиксельной высокоскоростной обработки изображений.

Не все графические процессоры допускают программирование вычислительных процессов. Основным критерием выбора графического процессора (с учетом его быстродействия, объема оперативной памяти) является структура процессора (т.е. сколько потоков он обрабатывает за один такт) и возможность доступа к программированию вычислительных процессов в шейдерных процессорах.

Количество потоков GPU может быть от нескольких сотен до нескольких тысяч, в зависимости от чипа. Так, например, ядро чипа видеокарты GeForce RTX 2080 Ti (Turing) имеет конфигурацию 4352:272:88, что означает: 4352-унифицированных шейдерных процессора (независимых потоков), 272 текстовых блока и 88 блоков растеризации.

Список платформ был бы не полный, если не вспомнить платформы для мобильных устройств типа планшетов и мобильных телефонов. Наиболее распространенным представителем этого направления являются SoC процессоры от производителя Qualcomm. Основное отличие процессоров Qualcomm от других платформ это наличие сразу двух блоков DSP и GPU. Блок DSP отвечает за

вычислительные процессы (обработку изображений в пиксельной графике), а блок GPU за обработку векторной графики. Эти блоки связаны друг с другом. Блок DSP имеет VLIW-архитектуру, когда несколько исполнительных блоков исполняют одну и ту же инструкцию в параллели, а некоторые математические операции дополнительно ускорены при помощи специальных инструкций, чтобы обеспечить быстрое исполнение типичных операций при обработке сигналов типа быстрого преобразования Фурье (Fast Fourier Transform - FFT).

Хотя производитель не указывает точных характеристик и структуры блоков DSP и GPU, по результатам работы алгоритмов обработки видеоизображений производительность этих блоков выше производительности CPU в сотни раз при низкой частоте обработки (от 300 до 500 МГц).

Среда программирования GPU является основным этапом реализации высокоскоростных алгоритмов обработки изображений. В настоящее время на рынке программного обеспечения существует несколько сред разработки для реализации высокоскоростных алгоритмов обработки изображений.

Отметим, что DirectX 11.1 и OpenGL 4.1 не реализуют в полном объеме обработку пиксельной графики. Эти среды предназначены для обработки векторной графики. Программировать вычислительные процессы, тем более по разработанным алгоритмам, - не реализуемая задача. Использование OpenCL 1.2 возможно, но его возможности для пиксельной графики ограничены.

На сегодняшний день рекомендована среда разработки Vulkan API, созданная компанией Khronos Group (разработчик OpenCL). Vulkan API, предоставляет улучшенную абстракцию для программирования графики на современных видеокартах. Vulkan API создана для программистов, которым нужно больше производительности, несмотря на более сложный и кропотливый процесс написания приложения. Среда Vulkan предназначена для программирования и вычислительных процессов алгоритмов обработки изображений и векторной графики. Эта среда универсальна для различных операционных систем (Windows, Linux, Android, Apple и др.), предназначена для, практически, всех видов и производителей GPU.

Но этот универсализм имеет недостаток. Компилятор автоматически не разбрасывает процессы по структуре GPU. Разбираться с архитектурой GPU программисту придется самостоятельно по документации на GPU и самому конфигурировать вычислительные потоки. Изначально, Vulkan была «заточен» под чипы GPU и SoC для процессоров GPU от AMD.

Для видеокарт с чипом NVIDIA используется среда разработки CUDA SDK. В основе интерфейса программирования приложений CUDA лежит язык Си с некоторыми расширениями. Для успешной трансляции кода на этом языке в состав CUDA SDK входит собственный Си-компилятор командной строки от компании NVIDIA. Компилятор создан на основе открытого компилятора Open64 и предназначен для трансляции host-кода (главного, управляющего кода) и device-кода (аппаратного кода, файлов с расширением *.cu) в объектные файлы, пригодные для сборки конечной программы или библиотеки в любой среде программирования. В CUDA имеется возможность подключения к приложениям, использующим OpenGL и Direct3D. CUDA кроссплатформенное программное обеспечение для таких операционных систем, как LINUX, MacOSX и WINDOWS.

Для видеочипов GPU и SoC процессоров GPU от производителя Intel используется программная среда Intel C for Metal (ICM). Сам язык ICM представляет собой язык C++ с некоторыми ограничениями и расширениями. По сравнению с C++ в ICM отсутствуют указатели, выделение памяти и статические переменные. Под запретом рекурсивные функции.

Зато присутствует явная модель векторного программирования: векторные типы данных, матрицы и поверхности. Векторные операции на этих типах данных, векторные условия if/else, независимо выполняются для каждого элемента вектора.

Также присутствуют встроенные функции для доступа к аппаратной фиксированной функциональности Intel GPU. Работу с векторами, матрицами и поверхностями в реальных задачах облегчают объекты подмножеств. Из базовых объектов можно выбирать только интересующие блоки или, как частный случай, отдельные элементы по маске.

Процессоры GPU являются наиболее эффективной платформой для реализации высокоскоростных алгоритмов обработки видеоизображений. Но избыточность их аппаратной и программной реализации не всегда может быть использовано в портативной аппаратуре.

Анализ платформы процессоров обработки сигналов

К особенностям цифровых высокоскоростных алгоритмов обработки видеоизображений следует отнести большой объем информации, поступающей в реальном времени в виде непрерывного потока данных [2, 3]. Их обработка требует от аппаратных средств высокой производительности и обеспечения возможности интенсивного обмена

данными с внешними устройствами. Выполнение указанных требований достигается благодаря использованию специальных процессоров с проблемно-ориентированной системой команд. Такие процессоры называются процессорами обработки сигналов (Digital Signal Processors - DSP). Они унаследовали от универсальных процессоров ряд свойств. В них широко используются методы сокращения длительности командного цикла, характерные для RISC-процессоров, такие как конвейеризация на уровне отдельных команд и микрокоманд, размещение операндов большинства команд в регистрах (регистровая адресация), использование теневых регистров для сохранения состояния вычислений при переключении контекста, разделение памяти и шин для команд и данных (гарвардская архитектура). Для DSP-процессоров присущ ряд особенностей:

- наличие аппаратного умножителя, позволяющего выполнять умножение двух чисел за один командный такт, в то время как в универсальных процессорах на умножение обычно затрачивается несколько тактов для выполнения последовательности операций сдвига и сложения;

- включение в систему команд таких операций, как умножение с накоплением ($B = A * X + B$), инверсия бит адреса и другие функции над битами;

- аппаратная поддержка программных циклов и кольцевого буферирования.

Использование формата с плавающей точкой, при котором данные представляются в виде экспоненциальной функции, с одной стороны, существенно упрощает и ускоряет обработку изображений, повышает надежность программы, при этом не требуется выполнения операций округления и нормализации данных, отслеживания ситуаций переполнения и потери значимости данных; с другой стороны, усложняет функциональные устройства и технологию производства микросхем, увеличивает процент отбраковки готовых изделий и, как следствие, стоимость процессоров.

Основными производителями DSP являются Texas Instruments (54% рынка), Freescale Semiconductor (14,1% рынка) и Analog Device (8% рынка).

Производители DSP, проектируя новые микросхемы, достаточно четко позиционируют их для использования в тех или иных приложениях. Это оказывает влияние и на их архитектуру, и на быстродействие, и на оснащение процессора тем или иным набором периферийных модулей.

Для обработки видеоизображений, видеонаблюдения, цифровых камер и 3D графики наиболее часто используются процессоры Texas Instruments (серии: TMS320DM64x/DaVinci,

TMS320C64xx, TMS320C62xx), Philips (серии: PNX1300, PNX1500, PNX1700), Freescale (серия MPC52xx).

Для обработки видеоизображений в медицинских приборах (для биометрии, измерительных систем в медицине) используются процессоры Texas Instruments (серии: TMS320C62xx, TMS320C67xx, TMS320C55xx, TMS320C28xx) и Analog Device серии SHARC+.

При выборе DSP-процессора для реализации высокоскоростных алгоритмов обработки изображений следует акцентировать внимание на его характеристиках.

Как и при выборе GPU, следует обратить внимание на число потоков, обрабатываемых процессором DSP. Проведем, для примера, сравнение процессора Texas Instruments (TI) TMS320C6748 с процессором Analog Device (AD) SHARC+ Dual-Core (серия ADSP-SC570/SC571/SC572/SC573/ADSP-21571/21573). В процессоре TI 8 функциональных блоков, два файловых регистра и два маршрута данных. Всего в этом процессоре $2 \times 2 \times 8 = 32$ потока, при скорости ядра 456 МГц. В рассматриваемом процессоре SHARC+ Analog Device 2 ядра и 1 поток на ядро при частоте ядра 500 МГц. Понятно, что процессор TI можно использовать для реализации высокоскоростных алгоритмов обработки изображений. Процессор SHARC+ для наших задач использовать невозможно, невзирая на наличие в ядре ускорителей КИХ и БИХ фильтров.

Важной характеристикой при выборе сигнального процессора является формат данных и разрядность. Все процессоры DSP работают либо с целыми числами, либо с числами в формате с плавающей точкой, причем для целых чисел разрядность составляет 16 или 32 бита, а для чисел с плавающей точкой она равна 32 бита.

При выборе DSP необходимо тщательно анализировать алгоритм обработки и входные сигналы для правильного выбора разрядности и типа арифметики. Иногда, при невозможности подобрать подходящий процессор с плавающей точкой, используют DSP с фиксированной точкой и сжатие динамического диапазона обрабатываемых сигналов (компрессию). Но это приводит к увеличению сложности алгоритма обработки сигнала и повышает требования к быстродействию.

Ключевой параметр при выборе процессора - это скорость. Она влияет на время выполнения обработки входного сигнала и, следовательно, определяет его максимальную частоту.

Организация системы памяти процессора влияет на его производительность. Это связано с тем, что ключевые команды DSP-процессора являются многооперандными, и ускорение их

работы требует одновременного чтения нескольких ячеек памяти. Например, команда MAC требует одновременного чтения 2 операндов и самой команды для того, чтобы ее можно было выполнить за 1 такт. Это достигается различными методами, среди которых применение многопортовой памяти, разделение на память программ и память данных (Гарвардская архитектура), использование кэша команд.

Процессор DSP с плавающей точкой, обычно, предполагает работу с большими массивами данных и сложными алгоритмами, имеет либо встроенную память большого объема, либо большую разрядность адресных шин для подключения внешней памяти (а иногда и то, и другое). Выбор типа и объема памяти должен быть результатом тщательного анализа алгоритма, для которого используется DSP.

Степень сложности разработки алгоритмов для реализации высокоскоростных алгоритмов обработки изображений определяется средствами разработки. При этом необходимо иметь в виду, что удобства в разработке кода программы для разработчика (обычно связываемые с использованием при программировании DSP языков высокого уровня) в большинстве случаев оборачивается получением менее компактного и менее быстрого кода.

В современных условиях скорость разработки (и, следовательно, выхода нового изделия на рынок) может принести больше выгод, чем затраты времени на оптимизацию кода при написании программы на ассемблере.

При выборе DSP-процессора и средств разработки необходимо учитывать особенности архитектуры.

Пользователи компиляторов языков высокого уровня иногда замечают, что они генерируют лучший код для процессоров с плавающей точкой. Это происходит по нескольким причинам: во-первых, большинство языков высокого уровня изначально не поддерживают арифметику с фиксированной точкой, во-вторых, система команд DSP с фиксированной точкой более ограничена, и в-третьих, процессоры с плавающей точкой обычно накладывают меньшие ограничения на объем используемой памяти.

Оптимальный и быстродействующий код получается при компиляции программ на языках высокого уровня для VLIW-процессоров DSP (процессоры со сверхдлинным словом команды) с простой ортогональной RISC-системой команд и большими регистровыми файлами. Однако даже для этих процессоров генерируемый компилятором код получается более медленным по сравнению с оптимизированным вручную ассемблерным кодом. Возможность сначала смоделировать процесс обработки сигнала в

MathLab [8] с дальнейшей автоматической трансляцией его в программу для DSP позволяет избавиться от множества серьезных ошибок еще на начальном этапе разработки.

Тщательный выбор цифрового процессора еще на начальном этапе разработки может помочь избавиться от излишних затрат, связанных с выбором неподходящего DSP, и сократить как время разработки в целом, так и время и средства на выявление ошибок.

Анализ платформы ПЛИС

Применение ПЛИС (программируемых логических интегральных схем) актуально для встраиваемых приложений реального времени, где важными факторами являются быстродействие и энергозатраты аппаратуры. Аппаратный параллелизм процессов в ПЛИС наиболее применим для реализации высокоскоростных алгоритмов обработки изображений. Однако простой перенос последовательных алгоритмов в ПЛИС не приводит к увеличению быстродействия, так как такие алгоритмы предназначены для последовательных (однопоточных) CPU. Для эффективного использования ПЛИС необходимо преобразовать последовательные алгоритмы в ракурсы параллелизма процессов и ресурсы конкретной ПЛИС. Все требования к аппаратуре, изложенные выше, сохраняются в ПЛИС.

Анализ эффективности аппаратуры для реализации высокоскоростных алгоритмов обработки изображений в ПЛИС невозможно в отрыве от анализа их конкретных производителей. Основные разработчики ПЛИС: ACSTEL, ALTERA, XILINX. Подход к анализу интересующих нас характеристик ПЛИС одинаков для всех моделей и версий, поэтому ограничимся рассмотрением продуктов разработчика XILINX. В рамках статьи не рассматриваются встраиваемые в ПЛИС процессоры: NIOS, NIOS II, PicoBlaze, MicroBLAZE. Эти процессоры, в основном, предназначены для управления, а не для обработки изображений в принципе.

Минимальные характеристики ПЛИС для реализации алгоритмов высокоскоростной обработки видеоизображений:

- системная тактовая частота, не менее 500 МГц;
- наличие аппаратных кэшей: L1 около 32К и L2 около 512К;
- количество DSP блоков, не менее $54 \times$ число последовательных звеньев в каждом потоке алгоритма, допустим не менее $54 \times 4 = 216$;
- DSP блок должен работать с арифметикой в формате 16/32 бит Float point;
- объем блочной памяти (BRAM) не менее $216 \times 32 = 1728$ Kbit;

- возможность работы с памятью DDR2 или DDR3.

В XILINX такой чип находится в серии ZYNG-7000 [4, 5] – это NEON со встроенным SoC процессором. Этот процессор предназначен для обработки видео изображений. Согласно документации [6] на этот процессор, он имеет 32 параллельных потока и 32-х разрядный вектор вычислений. Алгоритм обработки изображений можно разбить на 32 параллельных потока и $2^{32} = 4\,294\,967\,296$ вычислений. Так как NEON встроен в ZYNG-7000, то необходимые, перечисленные выше, характеристики ПЛИС для реализации высокоскоростных алгоритмов обработки изображений есть в наличии. Процессор NEON имеет возможность работать с изображениями с разрешением до 1080p. Наличие процессора NEON в серии ПЛИС ZYNG-7000 упрощает разработку высокоскоростных алгоритмов обработки изображений, так как видео процессор уже встроен в ПЛИС, оптимизирован, то разрабатывать его нет необходимости. Процессоров NEON в чипе может быть один (Z-7007S - Z-7014S) или два (Z-7010 - Z-7100).

В статье [7] описан процессор IPPro. Авторы описывают эффективный 16-битный целочисленный процессор, способный работать на частоте 337 МГц, специально нацеленный на поток данных, наблюдаемый в сложных высокоскоростных алгоритмах обработки изображений.

Представленная архитектура использует выделенные инструкции доступа к потоку на входе и выходе, с 32-битной локальной памятью для хранения пикселей, промежуточных результатов и отдельной 32-битной памятью ядра для хранения коэффициентов фильтра, других параметров и констант.

Использование как параллелизма на уровне данных, так и параллелизма на уровне задач демонстрируется посредством сопоставления алгоритма кластеризации K-средних на архитектуру, показывая хорошую масштабируемость скорости обработки с использованием нескольких ядер IPPro.

В статье рассмотрен пример реализации разделения потока данных между ядрами IPPro и процессором ARM с этапами преобразования цвета и морфологической фильтрации, отображенными в IPPro. Опять же, реализация параллельных ядер IPPro может значительно ускорить высокоскоростные алгоритмы обработки изображений по сравнению с обычным программным обеспечением, не прибегая к утомительным усилиям по проектированию специального оборудования на других аппаратных платформах. Ядро IPPro может быть реализовано на других сериях XILINX, например на SPARTAN 3A.

Следует сделать замечание о программных средах для разработки алгоритмов. Во-первых, алгоритмы можно разрабатывать в MathLab [8] и в дальнейшем интегрировать в среду программирования ПЛИС. Во-вторых, среды программирования ПЛИС для ALTERA (QUARTUS) и XILINX (ISE и VIVADO) доступны в WEB версии и хорошо документированы, что вполне достаточно для пробных разработок. В эти среды программирования встроен симулятор, что упрощает отладку проектов до схемной разработки и изготовления образцов.

В ПЛИС можно реализовать алгоритм высокоскоростной обработки изображения любой сложности и быстродействия в рамках системной тактовой частоты.

Заключение

Рассмотренные платформы реализации высокоскоростных алгоритмов обработки видеоизображений можно применить в аппаратуре различных отраслей народного хозяйства и в медицине:

- текстильная промышленность - оценка качества и текстуры полотна в реальном режиме времени до его выхода из цикла изготовления [9];

- металл литейное производство - визуальная (компьютерная) оценка качества литья с использованием медианных методов распознавания образов;

- железнодорожный транспорт - в задачах распознавания состояния несущих труб различных систем (системы отопления, кабели электроснабжения), по визуальной оценке их состояния на компьютере моделируются технологии их ремонта или замены [10];

- железнодорожный транспорт - распознавание лиц в потоке пассажиров в целях безопасности на транспорте [10];

- промышленная дефектоскопия - в системах контроля следующими методами: визуальным, спектроскопией, тепловизионного контроля, рентгенокопией, ультразвукокопией (все указанные методы используют быстродействующие алгоритмы распознавания образов);

- в медицине широко применяется платформа ПЛИС для реализации высокоскоростных алгоритмов обработки видеоизображений в таких приборах как визуализатор вен (трансиллюминатор), капилляроскоп (прибор для наблюдения циркуляции крови в капиллярах и поверхностных венах), в приборах визуализации различных тканей человека (тканей груди женщин, глазного яблока) для выявления злокачественных опухолей;

- при разработке систем тестирования аналоговых и аналого-цифровых схем методами цифровой обработки сигналов [11].

Литература

1. Кестер, У. Проектирование систем цифровой и смешанной обработки сигналов. – М. : Техносфера, 2010. – 328 с.

2. Bier, Jeff. Choosing a Processor: Benchmarks and Beyond. - Berkeley Design Technology, Inc., 2006.

3. Choosing a DSP Processor. - Berkeley Design Technology, Inc., 1996-2000.

4. Zynq-7000 product selection product selection guide [Электронный ресурс] – Режим доступа: <https://www.xilinx.com/search/site-keyword-search.html?q=zynq-7000-product-selection-guide.pdf>.

5. Zynq-7000 SoC Data Sheet: Overview [Электронный ресурс] – Режим доступа: https://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf

6. Qin, Haoliang Boost Software Performance on Zynq-7000 AP SoC with NEON [Электронный ресурс] – Режим доступа: https://www.xilinx.com/support/documentation/application_notes/xapp1206-boost-sw-performance-zynq7soc-w-neon.pdf

7. Lu, Rui. The Design of FPGA-based Digital Image Processing System and Research on Algorithms / Rui Lu, Xiaohui Liu, Xiaodan Wang, Jin Pan, Kuangyi Sun, Hellen Waynes // International Journal of Future Generation Communication and Networking, 2017. - Vol. 10. - No. 2 (2017). - PP. 41-54.

8. Гонсалес, Р. Цифровая обработка изображений в среде MATLAB / Р. Гонсалес, Р. Вудс, С. Эддинс. – М. : Техно-сфера, 2006. – 616 с.

9. Новиков, А. Н. Разработка теоретических и методологических принципов создания систем компьютерного зрения для автоматизации контроля качества текстильных материалов : дис. ... доктора техн. наук : 05.13.06 / А. Н. Новиков. - Москва, 2014. – 287 с.

10. Локтев, Д. А. Методы и моделирование измерительной системы контроля объектов транспорта по их изображениям : дис. ... доктора техн. наук : 05.13.18 / Д. А. Локтев. - Москва, 2020. – 336 с.

11. Нестеренко, Д. О. Тестирование аналоговых и аналого-цифровых схем методами цифровой обработки сигналов / Д. О. Нестеренко, Ю. Е. Зинченко, В. Н. Соленов // Информатика и кибернетика, 2020. - №4(22). – С. 73-76.

Секирин А. И., Вустяк Н. Ф. Анализ аппаратных платформ для обработки изображений. В статье рассматриваются вопросы эффективности применения различных аппаратных платформ для построения алгоритмов высокоскоростной обработки изображений. Рассматриваются платформы графических процессоров, платформы процессоров обработки сигналов и платформы программируемых логических интегральных схем. Для каждой платформы рассмотрены средства разработки программного обеспечения, что является неотъемлемой частью проектирования высокоскоростных алгоритмов обработки изображений и выбора платформы для реализации этих алгоритмов. Рациональный выбор платформы для высокоскоростной обработки изображений позволит значительно сократить время разработки и ресурсы готового изделия.

Ключевые слова: высокоскоростная обработка изображений, процессоры обработки сигналов, системы на кристалле, программируемые логические интегральные схемы.

Sekirin A. I., Vustyak N. F. Analysis of hardware platforms for image processing. The article discusses the effectiveness of the use of various hardware platforms to build algorithms for high-speed image processing. It covers graphics processing unit (GPU) platforms, signal processor (DSP) platforms, and Field-Programmable Gate Array (FPGA) platforms. For each platform, software development tools are considered, which is an integral part of the design of high-speed image processing algorithms and the choice of a platform for implementing these algorithms. Rational choice of a platform for high-speed image processing will significantly reduce development time and finished product resources.

Keywords: high-speed image processing, digital signal processors, systems on a chip, field programmable gate arrays.

Статья поступила в редакцию 08.11.2021
Рекомендована к публикации профессором Мальчевой Р. В.