

БЕЛОРУСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И
РАДИОЭЛЕКТРОНИКИ

Кафедра информатики

Курсовая работа

по дисциплине: “Архитектура вычислительных систем”

по теме “Обзор векторных вычислительных систем”

Выполнил: студент гр. 853505

Говин В.П.

Проверил: руководитель работы

Леченко А.В.

Минск 2020

Оглавление

Введение.....	3
Постановка задачи.....	4
Описание и классификация векторных процессоров.....	4
Применение векторных вычислительных систем на практике.....	7
Структура векторной вычислительной системы.....	8
Обзор системы типа SIMD.....	9
Преимущества SIMD.....	11
Недостатки SIMD.....	12
Обзор GPU.....	13
Внешний графический процессор (eGPU).....	15
Обзор STAR-100.....	16
Выводы.....	19
Литература.....	20

Введение

Высокоскоростное исполнение программы – очень важный фактор, из которого вытекает необходимость в оптимизации и улучшении уже существующих вычислительных систем (рис.1). Задачи, с которыми сталкиваются современные системы, подразумевают в первую очередь оперирование большими объёмами данных. Общая вычислительная способность системы за единицу времени напрямую зависит от скорости выполнения простейших задач, из чего и появляется необходимость в доработках систем.

На практике решение ряда задач на универсальных ЭВМ зачастую невозможно, так как на получение результата затрачивается слишком много времени и ресурсов. Поэтому с ростом объемов обрабатываемых данных растёт и спрос на быстродействующие системы. Главным принципом остаётся вычисление результата за минимальный промежуток времени, а также минимизация затрат для достижения этих результатов.

Стоимость суперЭВМ всегда была выше стоимости обычных ЭВМ. Это обусловлено множеством факторов, наиболее важным из которых является большая стоимость разработки и создания необходимого оборудования, в то время как серийность данных устройств будет значительно меньше широко-распространённых стандартных ЭВМ. Также немалых средств требует и разработка программного обеспечения, необходимого для реализации потенциала и быстродействия будущей вычислительной системы.

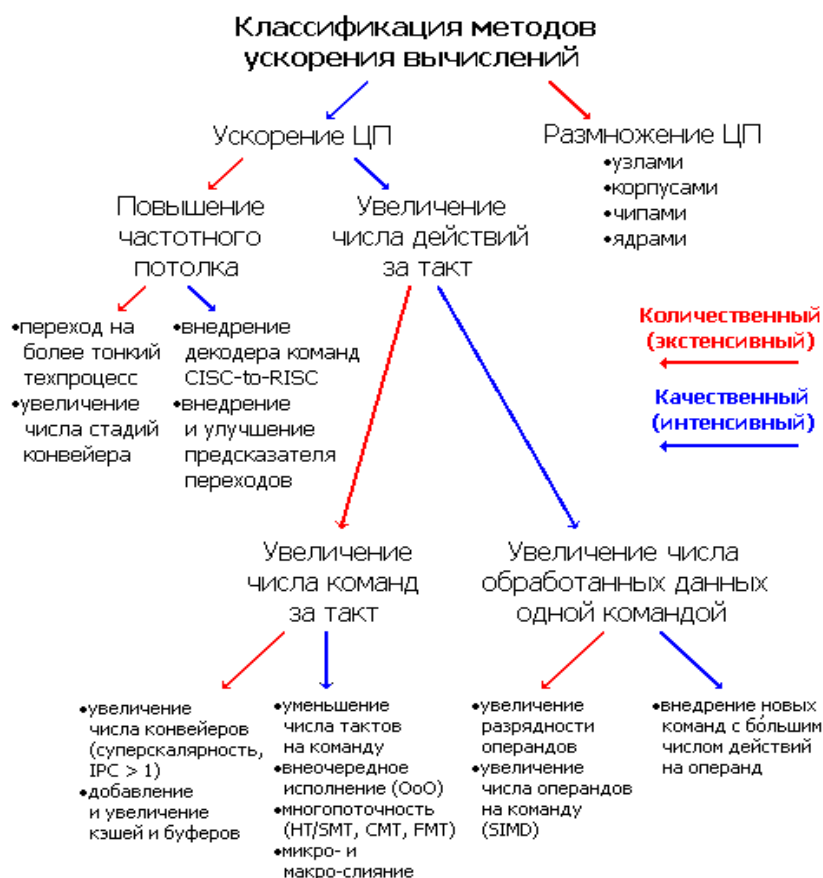


рис. 1. Классификация методов ускорения вычислительного процесса.

Постановка задачи: необходимо проанализировать информацию о различных типах векторных вычислительных систем, приведённую в различных источниках, и сделать выводы о потенциале их дальнейшего развития.

Описание и классификация векторных процессоров

Векторным называется процессор, который для исполнения команд оперирует массивами данных (векторами). К их реализации существуют несколько подходов. В ряде случаев, когда необходима лишь частичная реализация потенциала векторных вычислений, разрабатываются расширения для уже существующих универсальных ВС, но наиболее распространён подход, при котором векторный процессор разрабатывается как отдельная ВС.

В векторной обработке вычисления можно разделить на два типа:

- векторно-конвейерные (рис.2 а).
- векторно-параллельные (рис.2 б).

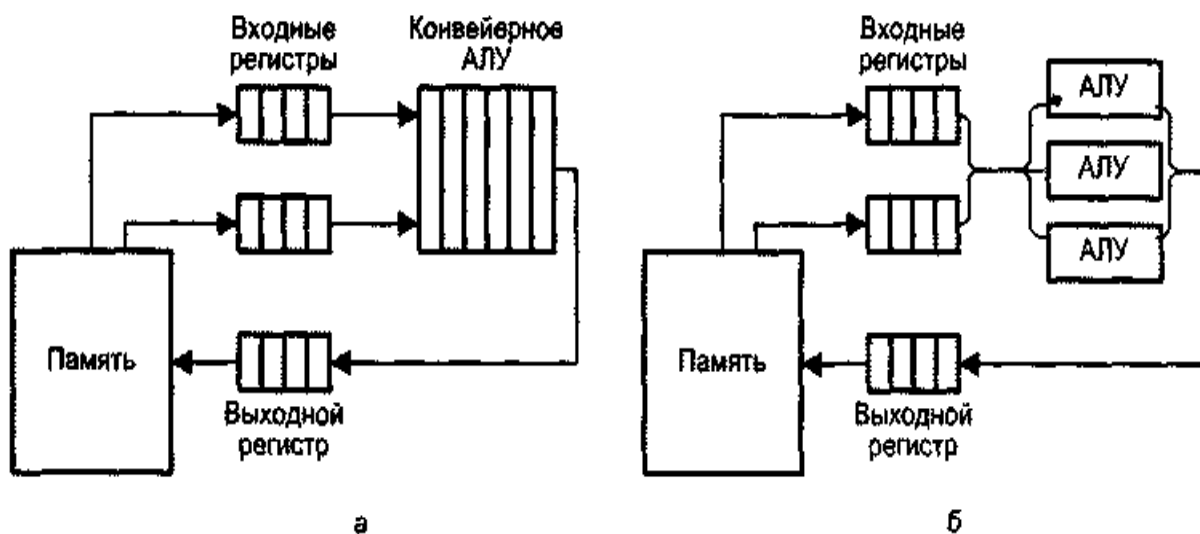


рис. 2. Типы векторных процессоров (а – векторно-конвейерный, б – векторно-параллельный).

Принцип работы процессора, основанного на векторно-параллельных вычислениях (рис.3), подразумевает наличие нескольких независимых функциональных блоков, выполняющих операции над элементами массивов (векторов) с плавающей запятой независимо друг от друга (каждый блок оперирует своей парой элементов).



рис. 3. Схема векторно-параллельного процессора.

Конвейерные вычисления (рис.4) – подход к обработке данных, при котором выполнение задачи разбивается на несколько этапов, которые обрабатываются последовательно. За счёт одновременной работы каждой из ступеней конвейера и параллельной обработке данных и образуется выигрыш в производительности.

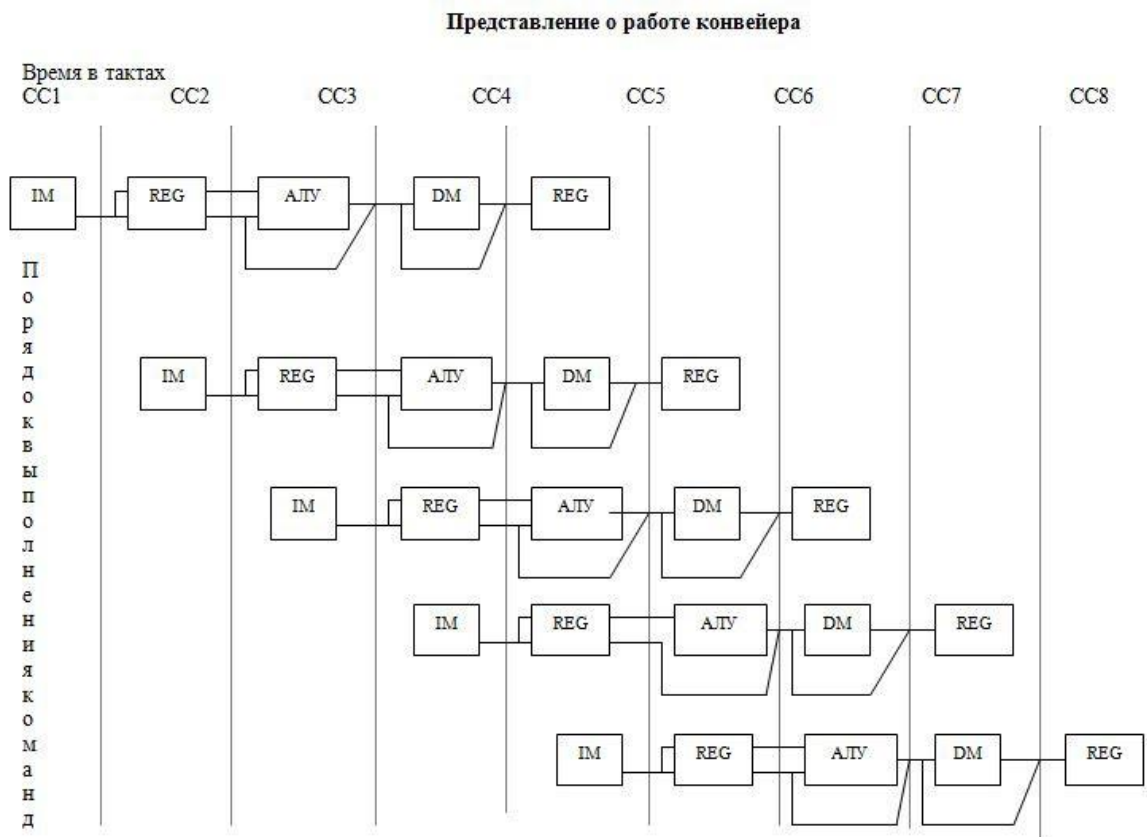


рис. 4. Принцип конвейеризации.

В векторно-конвейерном варианте процессора (рис.5) присутствует только один функциональный блок, который представляет собой конвейер. Именно он отвечает за обработку элементов векторов. Так как выполнение операций над числами с плавающей запятой достаточно трудоёмко, весь процесс происходит в несколько этапов, каждый из которых выполняется за один шаг конвейера. Подача следующей пары операндов на вход происходит сразу же после того, как первая ступень конвейера освобождается. Также возможен вариант выполнения параллельных операций при помощи нескольких конвейерных функциональных блоков, но в таком случае процессор будет использовать принципы как конвейерных, так и параллельных вычислений, и на практике такие варианты вычислительных систем не встречаются.

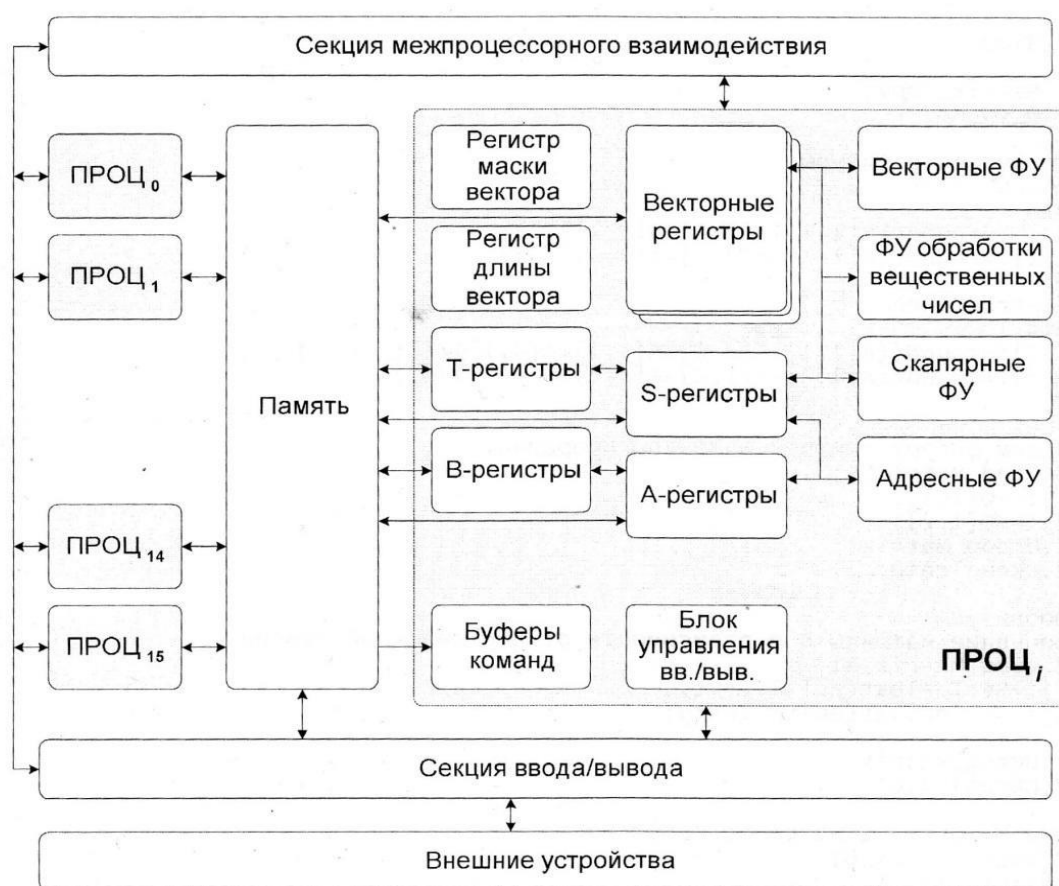


рис. 5. Схема векторно-конвейерного процессора

Применение векторных вычислительных систем на практике

На сегодняшний день большое распространение получила задача разработки 3D моделей реальных объектов. Данная задача является очень трудоёмкой относительно вычислительного процесса. Обработываются огромных размеров массивы чисел с плавающей запятой. Так как массивы можно представить в виде матриц и векторов, то обработка сводится к матричным операциям. Матричные операции в свою очередь сводятся к однотипным действиям над парами элементов. В универсальных ВС, оперирующих скалярами, такие операции на больших входных данных требуют много времени на обработку. Так как такого рода операции можно выполнять параллельно, для такого класса задач целесообразно использовать векторные вычислительные системы

Векторы представляются в виде одномерных массивов данных. Многомерные массивы представляют в виде набора одномерных массивов и имеют свою специфику размещения, что в свою очередь влияет на шаг изменения адреса элемента в памяти (рис.6). В векторной обработке для индекса (шага) с которым элементы вектора извлекаются из памяти, существует термин stride. Stride - шаг по индексу в векторной обработке.



рис. 6. Способы размещения матриц в памяти и их индексация.

Структура векторной вычислительной системы

На практике весь вычислительный процесс заключается не только в векторных вычислениях, но и в скалярной обработке, которая также играет важную роль. Именно поэтому в векторных вычислительных системах наряду с векторным присутствует также и скалярный процессор.

Система типа SIMD выполняет программу, в которой присутствуют как скалярные, так и векторные операции. Процессор обработки команд последовательно достаёт из памяти ВС команды и передаёт их напрямую в процессор нужного типа. Функциональные блоки строятся по такому принципу, чтобы каждая ступень конвейерного ФБ выполняла операцию за один такт, причём число ступеней у различных ФБ различается. Для увеличения производительности конвейеры в функциональных блоках дублируются, как, например, это сделано в системе Cray C90.

Vector linking – технология, используемая в некоторых векторных процессорах. Её суть заключается в том, что регистр результата становится входным регистром для следующей операции, если это необходимо. Эта технология также называется “зацепление регистров”. Как только вычисление необходимых компонент завершено, сразу начинается вычисление результата. Операция свёртки является ярким примером операции, подходящей для такого типа вычислений.

С середины 90-х годов прошлого века векторные ВС стали уступать другим более технологичным видам систем в производительности, но тем не менее разработка корпорации NEC – ВС SX-9 (рис.7) – является векторно-конвейерной ВС, пиковая производительность которой с 16 ядрами составляет 26,2 триллионов операций с плавающей запятой в секунду (TFLOPS).



рис. 7. Векторная система NEC SX-9.

Обзор системы типа SIMD

SIMD (Single Instruction Multiple Data) системы (рис.9) – вычислительные системы, обладающие множеством процессоров, обычно их количество от 1024 до 16384. Обработывают такие системы огромный поток данных, применяя при этом одну инструкцию. Векторные системы в какой-то мере можно отнести к отдельному классу SIMD-систем. Обработка данных векторными системами схожа с обработкой данных скалярными системами. Главное отличие в том, что векторные компьютеры обрабатывают массивы данных, в то время как скалярные обрабатывают элементы этих массивов (рис.8).

Scalar Operation

$$\begin{array}{l} A_1 \times B_1 = C_1 \\ A_2 \times B_2 = C_2 \\ A_3 \times B_3 = C_3 \\ A_4 \times B_4 = C_4 \end{array}$$

SIMD Operation

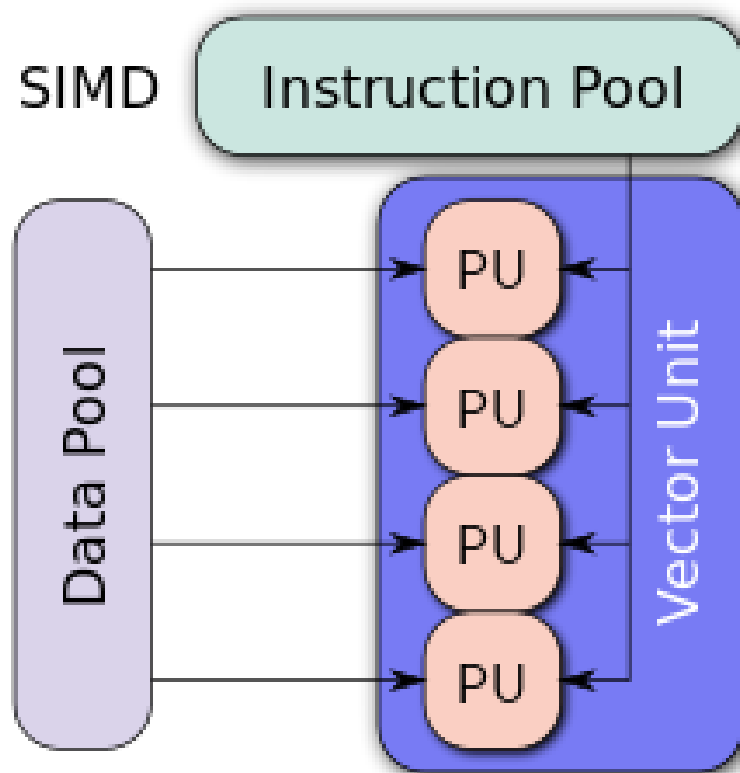
$$\begin{array}{l} A_1 \\ A_2 \\ A_3 \\ A_4 \end{array} \times \begin{array}{l} B_1 \\ B_2 \\ B_3 \\ B_4 \end{array} = \begin{array}{l} C_1 \\ C_2 \\ C_3 \\ C_4 \end{array}$$

рис. 8. Отличие скалярных операций от векторных.

В качестве примера существующих SIMD машин можно привести следующие модели:

- Gamma 2
- Quadrics Apemmile
- Hitachi S3600
- CPP DAP

Так как такие системы оперируют двумя типами процессоров – скалярными и векторными, то производительность такой системы при работе векторного процессора будет во много раз больше, потому что данные обрабатываются практически параллельно. Результаты доступны на один, два или три такта (основной временной параметр).



puc 9. Single Instruction stream Multiple Data stream

Преимущества SIMD

Типовой задачей для SIMD-системы является задача изменения яркости. Для её выполнения необходимо одно и то же значение добавить (при увеличении), либо вычесть (при уменьшении) к большому объёму данных. Итоговая яркость зависит от яркости RGB-составляющих (рис.10) каждого пикселя. Значения этих составляющих для каждого из пикселей считываются из памяти, проводится операция над ними и затем новые значения записываются обратно.

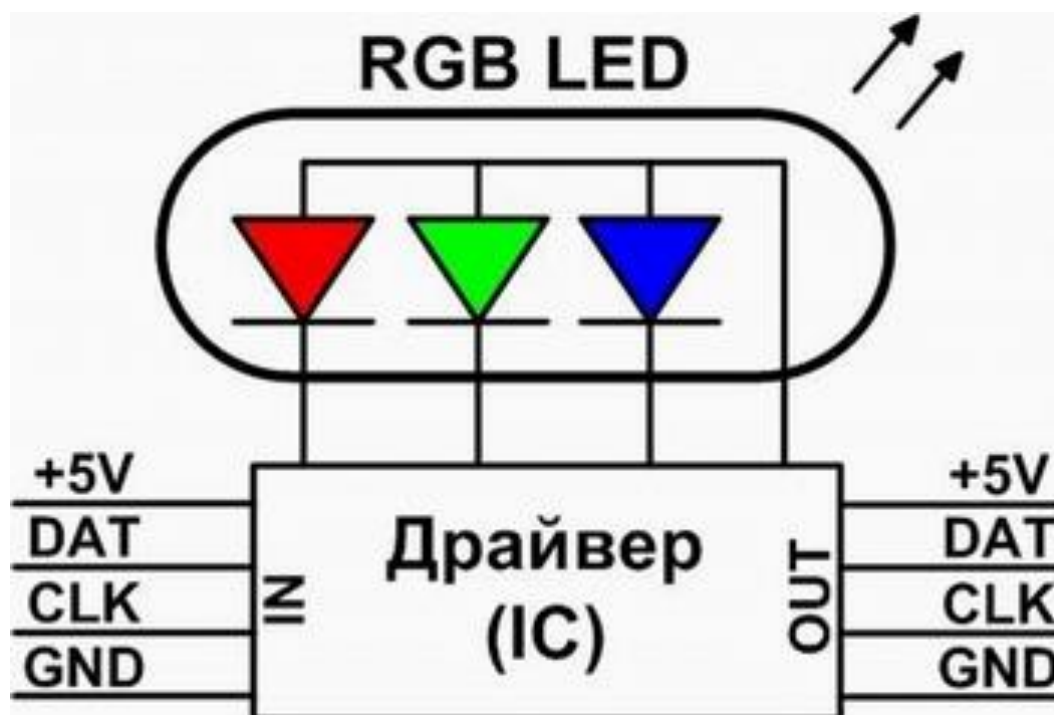


рис. 10. схема RGB пикселя.

SIMD-система эффективна для такого класса задач по нескольким причинам:

- Данные на вход подаются блоками, поэтому значения могут быть полностью загружены сразу. Вместо последовательного исполнения инструкций получения значений каждого пикселя, исполнится одна инструкция получения N пикселей, что в конечном итоге даёт прирост производительности в сравнении со скалярным вариантом процессора.
- Так как операции в SIMD системах выполняются над блоком данных, а не над каждым значение последовательно, это даст прирост производительности.

Таким образом система класса SIMD бесспорно является более эффективной, чем скалярные процессоры, для выполнения такого класса задач.

Недостатки SIMD

На сегодняшний день не существует идеальных вычислительных систем. Все они разрабатываются и создаются для решения определённого класса задач. SIMD-системы не являются исключением.

Из явных недостатков таких систем можно привести следующее:

1. Далеко не все алгоритмы можно векторизовать. Потоки тяжёлых задач нельзя так просто ускорить за счёт использования системы класса SIMD.
2. Так как такие системы оперируют большим количеством электроники (количество процессоров может достигать 13 тысяч и регистры больше стандартных скалярных), эти системы занимают значительно больше места их скалярных аналогов и потребляют в разы больше электроэнергии.
3. Для реализации SIMD алгоритмов необходимо писать программы вручную, так как далеко не все компиляторы умеют из C-программы генерировать SIMD инструкции.
4. SIMD-инструкции могут вызывать конфликты на низком уровне: ограничения на выравнивание данных.
5. Распределение результатов на нужные места — сложный процесс, что приводит к потере в эффективности.
6. Не во всех процессорах есть инструкции SIMD. В таком случае необходимо дописывать не-векторизованные аналоги.

Обзор GPU

Графический процессор (graphics processing unit(рис.11.)) - важнейшая составляющая современных персональных компьютеров, игровых приставок, мобильных устройств, которая отвечает за обработку графики. В конце 90-х годов графические процессоры получили широкое применение во многих устройствах.

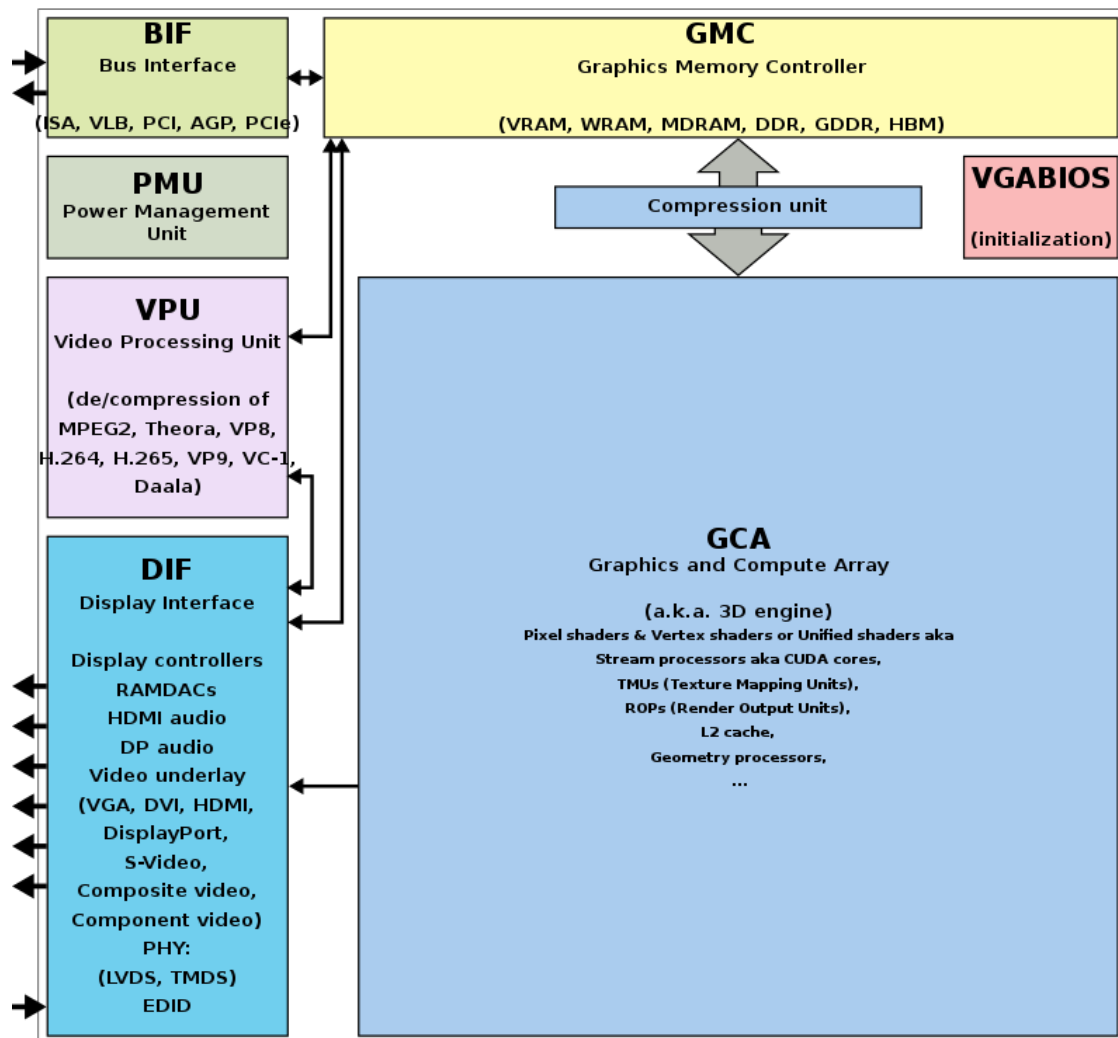


рис 11. Устройство графического процессора

Современные GPU довольно эффективно справляются с обработкой компьютерной графики, благодаря тому, что работают на схожей с системой SIMD конвейерной архитектуре. С задачей обработки графической информации они справляются в разы эффективнее, чем типичный CPU.

В современных видеокартах (рис.12.) графические процессоры используются для значительного ускорения обработки трёхмерной графики. Может применяться как в дискретной видеокарте, так и в интегрированных.

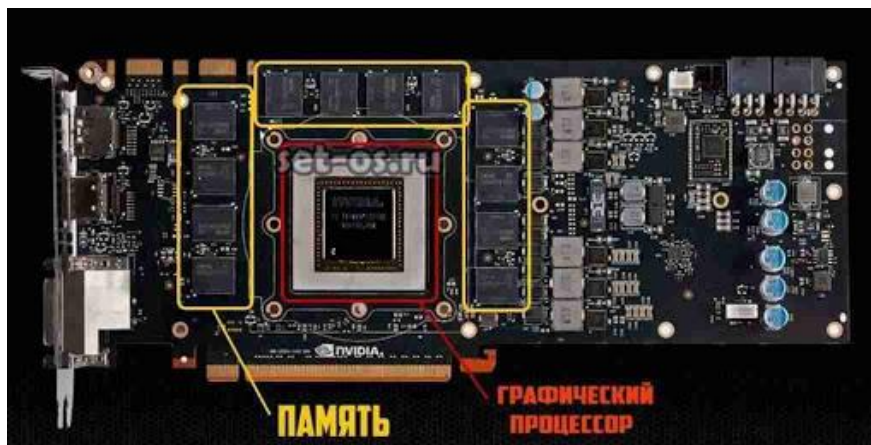


рис 12. Графический процессор в составе видеоадаптера.

GPU отличается от CPU в первую очередь архитектурой, которая направлена на обработку большого объема данных, так как расчёт моделей и объектов затрачивает много ресурсов. Также отличительной чертой GPU является сравнительно небольшой набор команд.

Высокая производительность GPU достигается в первую очередь за счёт особенностей архитектуры. Современные центральные процессоры имеют небольшое количество ядер, по сравнению с графическими процессорами. Графические процессоры изначально на этапе проектирования создавались как многоядерная вычислительная система, предназначенная для обработки большого числа потоков.

Как уже было сказано раньше, каждая архитектура разрабатывается для определённых задач. Если CPU спроектирован для последовательной обработки потока данных, то GPU с самого раннего этапа разработки проектировался для обработки компьютерной графики, то бишь для разовой обработки большого объема информации, поэтому его архитектура направлена на массивные параллельные вычисления.

Современные графические процессоры (в составе видеокарты) также могут быть использованы для общих вычислений (GPGPU). Примерами таковых могут служить чипы 5700XT (от AMD) или GTX 1660 Super (от nVidia).

Внешний графический процессор (eGPU)

Графический процессор также существует и в формате внешнего устройства – eGPU (рис.13). В таком случае он размещается вне корпуса персонального компьютера. Чаще всего такие устройства используются для увеличения вычислительной производительности портативных персональных компьютеров, так как в них устанавливается компактный и более энергоэффективный чип. Внешние GPU служат для увеличения вычислительной мощности.

В отличие от векторных вычислительных систем GPU не пользовались большой официальной поддержкой поставщиков несмотря на то, что работают по схожей схеме.



рис 13. Внешняя графическая вычислительная система от фирмы Lenovo.

Обзор STAR-100

В 1970 году произошёл анонс векторно-конвейерной вычислительной системы STAR-100. Фирма CDC вела её разработки с середины 70-ых и уже в 73-ем году в свет вышел первый аналог. Изначально данная система создавалась с учётом специфики языка программирования APL, в функционал которого входят широко развитые средства для обработки векторов, матриц и массивов.

В состав вычислительной системы STAR-100 входили две подсистемы: первая отвечала за обработку данных, вторая - за функции операционной системы. Ядром первой подсистемы был процессор, который состоял из конвейеров. В типовых конфигурациях STAR-100 процессоры состояли из трех конвейеров: K_1 , K_2 , K_3 (рис.14).

Конвейеры имели свои задачи:

- K_1 и K_2 предназначены для векторной обработки данных.
- K_3 предназначен для скалярной обработки данных.

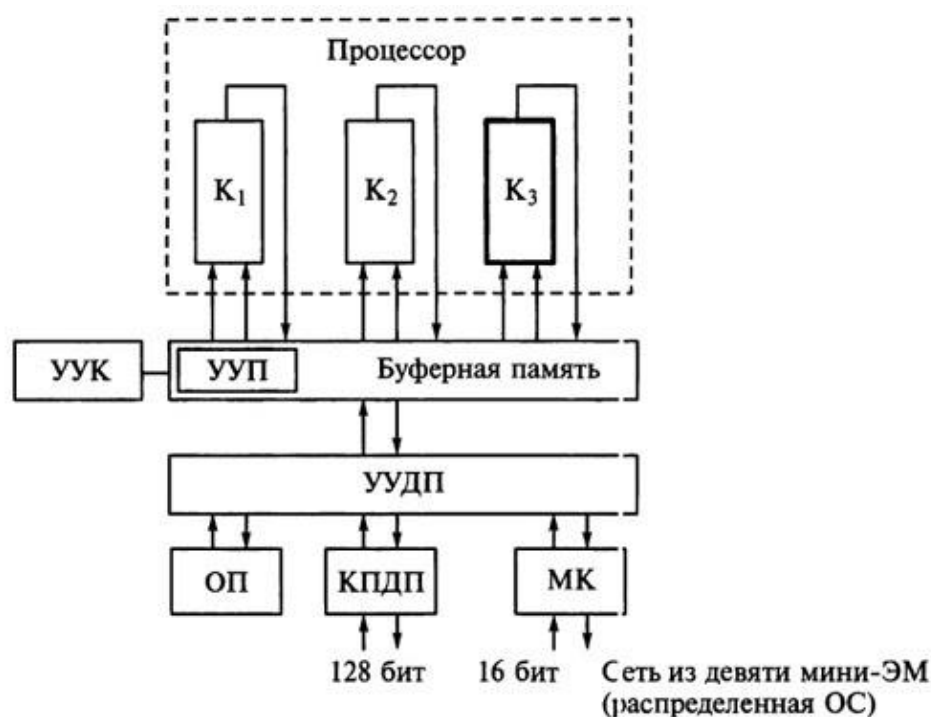


Рис. 4.3. Функциональная структура системы STAR-100:

K_1 , K_2 , K_3 — конвейеры; УУК — устройство управления командами; УУП — устройство управления потоками; УУДП — устройство управления доступом к памяти; ОП — оперативная память; КПДП — канал прямого доступа в память; МК — мультиплексный канал

Рис.14. архитектура ВС STAR-100.

STAR-100 проектировалась с программируемой структурой для более гибкой настройки, поэтому при относительно небольшом количестве ФБ можно было выполнять множество арифметических операций, для выполнения каждой из которых конвейер необходимо настроить под заданную операцию.

В K_1 и K_2 присутствовал булевый вектор, благодаря чему была обеспечена избирательная обработка компонентов векторов-операндов. Единица в i -м разряде булевского вектора означала, что операция над i -ми компонентами соответствующей пары векторов не производится.

В каждом конвейере была заложена возможность реализации операции сложения, а в двух из них - K_1 и K_2 - операций умножения и деления. Состав элементарных блоков обработки информации конвейеров был выбран с учетом распределения вероятностей использования микроопераций различных типов.

Каждый конвейер K_i ($i = 1, 2, 3$) мог включать в себя приблизительно 30 блоков обработки информации. Работа всех блоков осуществлялась параллельно, но каждый из них оперировал с вполне определенными элементами векторов данных либо со своими операндами.

Любой конвейер воспринимал 64-разрядный код либо как один 64-разрядный операнд, либо как два 32-разрядных операнда. Время выполнения операции над парой операндов в любом из блоков конвейеров не превышало 40 нс. Следовательно, данные могли поступать в конвейеры K_1 и K_2 со скоростью 103 млн опер./с.

В систему STAR-100 входил набор из 230 команд, 65 из которых предназначались для работы с векторами данных и 130 команд - для работы со скалярами. Средства управления подсистемой переработки данных были представлены композицией из устройства управления командами (УУК), устройства управления потоками (УУП) и устройства управления доступом к памяти (УУДП).

Первое (УУК) имело буфер опережающего просмотра команд (емкостью в четыре 512-разрядных суперслова) механизмом работы, организованном на стековой системе. Второе устройство (УУП) использовалось для управления потоками операндов и команд между УУДП, конвейерами и УУК.

В оперативной памяти хранились программы и данные. Она была реализована на магнитных сердечниках и имела емкость 512-1024 К 64-разрядных слов, т.е. до 8 Мбайт. Память могла включать в себя до 32 модулей и относилась к классу памяти с перемежающимися адресами. Время цикла памяти было равно 1,28 мкс, однако допускались одновременные обращения к составляющим модулям. Имелись четыре виртуальных канала обращения к памяти, которые реализовывались устройством управления доступом к памяти. Два канала использовались для чтения операндов (для каждого из конвейеров K_1 и K_2 , работавших параллельно, из памяти выбиралось по два 64-разрядных операнда); один - для записи результатов (64-разрядный результат от каждого из конвейеров K_1 и K_2); один - для обмена информацией с устройствами ввода-вывода (либо с одним быстродействующим устройством с полосой пропускания 128 бит, либо с восемью медленными устройствами в режиме разделения времени).

Буферная память была реализована вследствие того, что быстродействие оперативной памяти было существенно ниже быстродействия процессора. Буферная память представляла собой совокупность регистров с временем цикла 40 нс. Назначение канала прямого доступа в память (КПДП) и мультиплексного канала (МК) следует из их названий и структуры связей между устройствами STAR-100.

Функции операционной системы STAR-100 реализовывались специальной вычислительной сетью из девяти мини-машин. Система программирования STAR-100 включала компиляторы с языков APL-STAR, COBOL и FORTRAN.

Первый образец системы STAR-100 был установлен в Ливерморской радиационной лаборатории им. Лоуренса (Lawrence Livermore Laboratory). Были осуществлены поставки системы в правительственные организации и в армию США. Система STAR-100 использовалась для управления запуском антиракет в системе противоракетной обороны США; она широко применялась при решении сложных проблем науки, техники и экономики.

Вычислительная система STAR-100 поддерживала изменение числа конвейеров, числа и состава внешних устройств, емкости оперативной и внешней памяти. Создавались и усеченные варианты STAR-IB, а также система SUPERSTAR (или CDC 8600), которая в 1,5-4 раза превосходила по производительности STAR-100 и была более компактной (реализована на более совершенных интегральных схемах).

Выводы

Развитие в сфере векторных вычислений происходило на протяжении многих лет. За эти годы было создано и опробовано множество аналогов векторных ВС. Все они имели свои достоинства и недостатки, но с течением времени развитие технологий позволяло всё более эффективно производить обработку больших объемов данных.

На данный момент векторные вычисления играют важную роль в современных вычислительных системах, так как с ростом производительности растут и объёмы данных, которые необходимо обрабатывать. Данная сфера разработок по-прежнему является крайне перспективной благодаря тому, что подавляющее большинство современной техники так, или иначе, работает, основываясь на параллельных вычислениях.

Литература

- [1] Kunzman, D. M.; Kale, L. V. (2011), Programming Heterogeneous Systems, 2011
- [2] Цилькер Б.Я. Организация ЭВМ и систем : Учебник для вузов / Б.Я. Цилькер, С.А. Орлов. - 2-е изд. - СПб.: Питер, 2011.
- [3] Супер ЭВМ. Аппаратная и программная организация / Под. ред. С. Фернбаха: Пер. с англ.- М.: Радио и связь, 1991.
- [4] nsc.ru [интернет-ресурс].
- [5] bmstu.wiki [интернет-ресурс].
- [6] wikipedia.org [интернет-ресурс].
- [7] habr.com [интернет-ресурс].