Вопросы к зачету КСИС:

1. Что такое микропроцессорная система? Особенности традиционной цифровой системы.

Микропроцессорная система - электронная система, предназначенная для обработки входных сигналов и выдачи выходных сигналов. В качестве входных и выходных сигналов при этом могут использоваться

* аналоговые сигналы
* одиночные цифровые сигналы,
* цифровые коды,
* последовательности цифровых кодов.

Характерная особенность традиционной цифровой системы состоит в том, что алгоритмы обработки и хранения информации в ней жестко связаны со схемотехникой системы. То есть изменение этих алгоритмов возможно только путем изменения структуры системы, замены электронных узлов, входящих в систему, и/или связей между ними.

1. Что такое микропроцессорная система? Особенности системы на гибкой логике?

**Микропроцессорные системы** ­– электронные системы, предназначенные для автоматизации обработки информации и управления различными процессами.

Преодолеть этот недостаток позволяют электронные системы **гибкой логики**, которые могут легко адаптироваться под любую задачу, перестраиваясь с одного алгоритма на другой без изменения электронной схемы. В таких системах изменение условий влияет только на изменение программы, в соответствии с которой работает система

1. Преимущества и недостатки систем на жесткой и гибкой логике?

Характерной особенностью традиционных цифровых систем является тот факт, что алгоритм обработки и хранения информации в ней жёстко связан со схемотехникой системы, то есть для конкретно поставленной задачи разрабатывается и реализуется конкретная электронная схема. Любое изменение исходных условий задачи повлечет за собой и изменение её схемотехнического решения, т.е. изменение алгоритма функционирования системы возможно только путём изменения её структуры. Такие схемы называют **схемами жёсткой логики**.

Преимуществом систем жёсткой логики является их высокое быстродействие, так как такие системы никогда не имеют аппаратной избыточности, а скорость выполнения алгоритмов определяется в ней только быстродействием отдельных логических элементов.

Самым большим недостатком цифровой системы на жёсткой логике является тот факт, что при изменении условий задачи схему нужно проектировать и изготавливать заново.

Преодолеть этот недостаток позволяют электронные системы **гибкой логики**, которые могут легко адаптироваться под любую задачу, перестраиваясь с одного алгоритма на другой без изменения электронной схемы. В таких системах изменение условий влияет только на изменение программы, в соответствии с которой работает система

Такая система является программируемой (перепрограммируемой). Именно к системе гибкой логики и относятся микропроцессорные системе.

Конечно, аппаратно такая схема может быть избыточна, так как должна функционировать и для решения самой простой, и для решения самой сложной задачи. А решение трудной задачи требует гораздо больше аппаратных средств, чем решение простой.

1. Что такое Большая интегральная схема? Какие бывают процессоры по количеству БИС?

**Больша́я интегра́льная схе́ма** (БИС) — **интегральная схема** соответствующая большой степени интеграции (плотности упаковки (размещения) на одном кристалле (чипе) элементов интегральных схем). Число схемных элементов в БИС от 103 до 10**4**.

По числу больших интегральных схем (БИС) в микропроцессорном комплекте различают микропроцессоры **однокристальные, многокристальные и многокристальные секционные**. Процессоры даже самых простых ЭВМ имеют сложную функциональную структуру, содержат большое **количество** электронных элементов и множество разветвленных связей

1. Виды процессоров по назначению? Преимущества и недостатки?

**арифметический процессор** , **буферный процессор** , **процессор данных** , **процессор баз данных** , **текстовый процессор** , **процессор ввода-вывода** , **интерфейсный процессор** , **лингвистический процессор** , **сетевой процессор** , **межсетевой процессор** , **процессор передачи данных** , **терминальный процессор** , **специализированный процессор** и др. ;

- **Универсальные микропроцессоры** могут быть применены для решения широкого круга разнообразных задач. При этом их эффективная производительность слабо зависит от решаемых задач. Специализация МП, т.е. его проблемная ориентация на ускоренное выполнение определенных функций позволяет резко увеличить эффективную производительность при решении только определенных задач.

**специализированных микропроцессоров,** можно выделить различные микроконтроллеры, ориентированные на выполнение сложных последовательностей логических операций, математические МП, предназначенные для повышения производительности

1. Виды процессоров по виду обрабатываемых сигналов? Преимущества и недостатки?

По виду обрабатываемых входных сигналов различают цифровые и аналоговые микропроцессоры.

- **Цифровые** – т.е. работающие с числовыми данными.

- **Аналоговые**– предназначены для обработки аналоговых сигналов и имеющие в качестве входных и выходных данных аналоговые сигналы. По сути, все современные аналоговые МП являются цифровыми сигнальными МП, имеющими на входе встроенные аналого-цифровые преобразователи (АЦП), а на выходе – встроенные цифро-аналоговые преобразователи (ЦАП).

Поэтому входные аналоговые сигналы передаются в МП через преобразователь в цифровой форме, обрабатываются и после обратного преобразования в аналоговую форму поступают на выход

1. Виды процессоров по временной организации работы? Преимущества и недостатки?

По характеру **временной организации работы** микропроцессоры делят на синхронные и асинхронные.

Синхронные микропроцессоры - микропроцессоры, в которых начало и конец выполнения операций задаются устройством управления (время выполнения операций в этом случае **не зависит от вида выполняемых команд** и величин операндов).

Асинхронные микропроцессоры позволяют начало выполнения каждой следующей операции определить по сигналу фактического окончания выполнения предыдущей операции.

1. Структура процессора?

Центральный процессор – электронный блок либо интегральная схема (микропроцессор), исполняющая машинные инструкции (код программ), главная часть аппаратного обеспечения компьютера или программируемого логического контроллера.

Центральный процессор состоит из 3-х частей:

1. Ядро процессора, которое выполняет основную работу. Оно позволяет читать, расшифровывать, выполнять и отправлять инструкции. Ядро состоит из следующих частей:

* Арифметико-логическое устройство (АЛУ). Выполняет основные математические и логические операции. Все вычисления производятся в двоичной системе.
* Устройство управления (УУ). Управляет работой CPU с помощью электрических сигналов. От него зависит согласованность работы всех частей процессора и его связь с внешними устройствами. блок управления определяет последовательность микрокоманд, выполняемых при обработке машинных кодов (команд).

Каждое ядро может выполнять только одну задачу, хоть и за долю секунды. Одноядерный процессор выполняет каждую задачу последовательно. Для современного объёма операций этого мало, поэтому ценятся CPU с более чем одним ядром, чтобы выполнять несколько задач одновременно. Например, двухъядерный выполняет две задачи одновременно, трехъядерный ― три и т. д.

1. Запоминающее устройство. Это небольшая внутренняя память центрального процессора. Она состоит из регистров и кеш-памяти. В регистрах хранятся текущие команды, данные, промежуточные результаты операции. В кеш-память загружаются часто используемые команды и данные из оперативной памяти. Обратиться в кеш быстрее, чем в оперативную память, поэтому объём кеш-памяти влияет на скорость выполнения запросов.
2. Шины ― это каналы, по которым передаётся информация. Они как рельсы для перевозки данных.

Главной характеристикой процессора является производительность. Она зависит от двух параметров: **тактовая частота**и**разрядность.**

**Тактовая частота** ― число выполненных операций в секунду. Измеряется в мегагерцах (МГц — миллион тактов в секунду ) и гигагерцах (ГГц — миллиард тактов в секунду). Чем больше тактовая частота, тем быстрее работает машина.

**Разрядность** ― количество информации (байт), которое можно передать за такт. Разрядность процессора бывает 8, 16, 32, 64 бита. Современные процессоры 32-х и 64-битные.

1. Типы архитектур процессора? По набору команд? По организации адресного пространства?

CISC

CISC (англ. Complex Instruction Set Computer — «компьютер с полным набором команд») — тип процессорной архитектуры, в первую очередь, с нефиксированной длиной команд, а также с кодированием арифметических действий в одной команде и небольшим числом регистров, многие из которых выполняют строго определенную функцию.

Самый яркий пример CISC архитектуры — это x86 (он же IA-32) и x86\_64 (он же AMD64).

В CISC процессорах одна команда может быть заменена ей аналогичной, либо группой команд, выполняющих ту же функцию.

плюсы и минусы архитектуры: высокая производительность благодаря тому, что несколько команд могут быть заменены одной аналогичной, но большая цена по сравнению с RISC процессорами из-за более сложной архитектуры, в которой многие команды сложнее раскодировать.

### RISC

RISC (англ. Reduced Instruction Set Computer — «компьютер с сокращённым набором команд») — архитектура процессора, в котором быстродействие увеличивается за счёт упрощения инструкций: их декодирование становится более простым, а время выполнения — меньшим.

По сравнению с CISC эта архитектура имеет константную длину команды, а также меньшее количество схожих инструкций, позволяя уменьшить итоговую цену процессора и энергопотребление, что критично для мобильного сегмента. У RISC также большее количество регистров.

Примеры RISC-архитектур: PowerPC, серия архитектур ARM (ARM7, ARM9, ARM11, Cortex).

В RISC быстрее CISC.

### MISC

MISC (англ. Minimal Instruction Set Computer — «компьютер с минимальным набором команд»).

Ещё более простая архитектура, используемая в первую очередь для ещё большего уменьшения итоговой цены и энергопотребления процессора.

Для увеличения производительности во всех вышеперечисленных архитектурах может использоваться “спекулятивное исполнение команд”. Это выполнение команды до того, как станет известно, понадобится эта команда или нет.

### VLIW

VLIW (англ. Very Long Instruction Word — «очень длинная машинная команда») — архитектура процессоров с несколькими вычислительными устройствами. Характеризуется тем, что одна инструкция процессора содержит несколько операций, которые должны выполняться параллельно.

Все способы адресации памяти можно разделить на:  
1) прямой  
2) косвенный

1. Особенности шинной структуры связей? Преимущества и недостатки?

При шинной структуре связей все сигналы между устройствами передаются по одним и тем же линиям связи, но в разное время (это называется мультиплексированной передачей).

Причем передача по всем линиям связи может осуществляться в обоих направлениях (так называемая двунаправленная передача). В результате количество линий связи существенно сокращается, а правила обмена (протоколы) упрощаются.

Группа линий связи, по которым передаются сигналы или коды как раз и называется шиной

+при шинной структуре связей легко осуществляется пересылка всех информационных потоков в нужном направлении,

-Однако при шинной структуре связей вся информация передается по линиям связи последовательно во времени, по очереди, что снижает быстродействие системы по сравнению с классической структурой связей.

Большое достоинство шинной структуры связей состоит в том, что все устройства, подключенные к *шине*, должны принимать и передавать информацию *по* одним и тем же правилам (протоколам обмена информацией *по* *шине* ). Соответственно, все узлы, отвечающие за обмен с *шиной* в этих устройствах, должны быть единообразны, унифицированы.

Существенный недостаток шинной структуры связан с тем, что все устройства подключаются к каждой линии связи параллельно. Поэтому любая неисправность любого устройства может вывести из строя всю систему, если она портит линию связи

1. Что такое системная магистраль? Что в нее входит?

Информационная магистраль ЭВМ – это подсистема, обеспечивающая обмен данными между структурными компонентами компьютера. Физически представлена набором разноуровневых проводников, интерфейсов, линий связи. Системная магистраль содержит 3 различные шины:

- шина адреса – набор однонаправленных сигнальных линий, по которым от ЦП к памяти или к устройствам ввода/вывода передаются коды адреса

- шина данных – набор двунаправленных сигнальных линий, по которым передаются данные между процессором и памятью либо устройствами ввода-вывода

- шина управления – передаёт управляющие сигналы для устройств.

1. Режимы обмена информацией на системной магистрали?

1. При синхронном обмене процессор заканчивает обмен данными самостоятельно, через установленное время, то есть без учета интересов устройства-исполнителя;

2. При асинхронном обмене процессор заканчивает обмен только тогда, когда устройство-исполнитель подтверждает выполнение операции специальным сигналом

Достоинства синхронного обмена — более простой протокол обмена, меньшее количество управляющих сигналов. Недостатки — отсутствие гарантии, что исполнитель выполнил требуемую операцию, а также высокие требования к быстродействию исполнителя.

Достоинства асинхронного обмена — более надежная пересылка данных, возможность работы с самыми разными по быстродействию исполнителями. Недостаток — необходимость формирования сигнала подтверждения всеми исполнителями, то есть дополнительные аппаратурные затраты.

или

Циклы обмена информацией делятся на два основных типа:

- цикл записи (вывода), в котором процессор записывает (выводит) информацию;

- цикл чтения (ввода), в котором процессор читает (вводит) информацию.

Во время каждого цикла устройства, участвующие в обмене информации, передают друг другу по системной магистрали информационные и управляющие сигналы в строго установленном порядке или, как еще говорят, в соответствии с принятым протоколом обмена информацией.

1. Регистры общего назначения? Как организованы?

Регистры – это специальные ячейки памяти, расположенные непосредственно в процессоре. Работа с регистрами выполняется намного быстрее, чем с ячейками оперативной памяти, поэтому регистры активно используются как в программах на языке ассемблера, так и компиляторами языков высокого уровня.

**eax**— регистр-аккумулятор. Применяется для хранения промежуточных данных.

**ecx** (Count register) — регистр-счетчик.

**ebx** (Base register) — базовый регистр.

**esi** (Source Index register) — индекс источника. Этот регистр в цепочечных операциях содержит текущий адрес элемента в цепочке-источнике (первоначальная строка).

**edi**(Destination Index register) — индекс приемника (получателя). Этот регистр в цепочечных операциях содержит текущий адрес в цепочке-приемнике (результирующая строка).

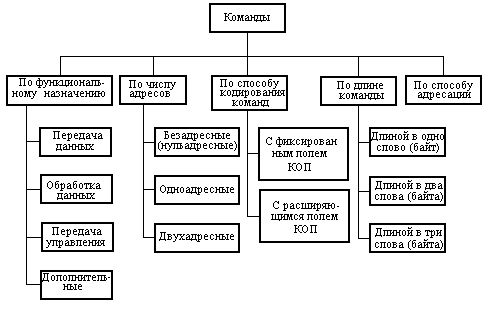
**esp** (Stack Pointer register) — регистр указателя стека. Содержит указатель вершины стека в текущем сегменте стека.

**ebp** (Base Pointer register) — регистр указателя базы кадра стека. Регистр предназначен для организации произвольного доступа к данным внутри стека.

1. Классификация команд?

В общем случае система команд процессора включает в себя следующие четыре основные группы команд:

* команды пересылки данных (Команды пересылки данных не требуют выполнения никаких операций над операндами. Операнды просто пересылаются (точнее, копируются) из источника (Source) в приемник (Destination).)
* арифметические команды;
* логические команды (логическое И, логическое ИЛИ, исключающее ИЛИ, очистку, инверсию, разнообразные сдвиги)
* команды переходов (предназначены для изменения обычного порядка последовательного выполнения команд. С их помощью организуются переходы на подпрограммы и возвраты из них, всевозможные циклы, ветвления программ,)



По архитектуре команд процессоры можно подразделить на

- процессоры с расширенной системой команд (CISC-процессоры) и

- процессоры с ограниченной системой команд (RISC- процессоры)

CISC- процессоры (Complex Instruction Set Computer) характеризуется широким набором команд, позволяющим обрабатывать данные различного формата с использованием различных сложных способов адресации. При этом многие процедуры реализации команд выносятся на аппаратный уровень. Такие процессоры ориентированы на использование в ЭВМ общего назначения, и им свойственно следующее:

* большинство команд имеют обращение к ОЗУ,
* длительность цикла исполнения каждой команды индивидуальна,
* имеет место ограниченный объем регистровой памяти.

RISC – процессоры (Reduced Instruction Set Computer) характеризуются не только ограничением количества команд, но и использованием самых простых способов адресации. Все команды разделены на две группы: к первой группе отнесены команды обработки данных, причем их состав таков, что все они короткие. Ко второй группе относятся все остальные команды, в том числе, команды обращения к памяти. Команды первой группы работают только с регистровой памятью, просты по своим функциям и имеют одинаковую длительность циклов. Выполнение длинных операций вынесено на программный уровень.

Кроме того, регистровая память в RISC - процессорах имеет существенно большую емкость; последнее необходимо для уменьшения в теле программы количества команд обмена с целью экономии процессорного времени на обмен между ОЗУ и РОН.

1. Методы адресации?

Под методом адресации – способ, с помощью которого команда указывает микропроцессору источники обрабатываемых кодов

Все способы адресации памяти можно разделить на:  
1) прямой, когда исполнительный адрес берется непосредственно из команды или вычисляется с использованием значения, указанного в команде, и содержимого какого-либо регистра (прямая адресация, регистровая, базовая, индексная и т.д.);  
2) косвенный: в команде содержится значение косвенного адреса, т.е. адреса ячейки памяти, в которой находится окончательный исполнительный адрес (косвенная адресация).

Неявная и регистровая адресация оперируют только с регистрами микропроцессора и не позволяют обращаться к ячейкам памяти ОЗУ и ПЗУ.

1. Сегментирование памяти? Для чего нужно?

Сегментация памяти - это метод управления памятью [операционной системы](https://translated.turbopages.org/proxy_u/en-ru.ru.d6422ed6-628e3890-14c69700-74722d776562/https/en.wikipedia.org/wiki/Operating_system) , позволяющий разделить [основную память](https://translated.turbopages.org/proxy_u/en-ru.ru.d6422ed6-628e3890-14c69700-74722d776562/https/en.wikipedia.org/wiki/Computer)[компьютера](https://translated.turbopages.org/proxy_u/en-ru.ru.d6422ed6-628e3890-14c69700-74722d776562/https/en.wikipedia.org/wiki/Primary_memory) на сегменты или секции.

Для хранения кодов адресов памяти используются не отдельные регистры, а пары регистров:

* сегментный регистр определяет адрес начала сегмента (то есть положение сегмента в памяти);
* регистр указателя (регистр смещения) определяет положение рабочего адреса внутри сегмента.

Она была предназначена для повышения надежности систем, выполняющих несколько процессов одновременно

1. Архитектура параллельных вычислительных систем? Преимущества и недостатки?

Параллельные вычислительные системы — это физические компьютерные, а также программные системы, реализующие тем или иным способом параллельную обработку данных на многих вычислительных узлах. Идея распараллеливания вычислений основана на том, что большинство задач может быть разделено на набор меньших задач, которые могут быть решены одновременно.

Писать программы для параллельных систем сложнее, чем для последовательных[[3]](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%BB%D0%BB%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B#cite_note-3), так как конкуренция за ресурсы представляет новый класс потенциальных ошибок в программном обеспечении ([багов](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%B0%D1%8F_%D0%BE%D1%88%D0%B8%D0%B1%D0%BA%D0%B0)), среди которых [состояние гонки](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D0%B5_%D0%B3%D0%BE%D0%BD%D0%BA%D0%B8) является самой распространённой. Взаимодействие и [синхронизация](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B8%D0%BD%D1%85%D1%80%D0%BE%D0%BD%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D0%B2&action=edit&redlink=1) между процессами представляют большой барьер для получения высокой производительности параллельных систем.

параллелизм на уровне битов, параллелизм на уровне инструкций, параллелизм данных, параллелизм задач

1. Классификация параллельных вычислительных систем?

Общая классификация архитектур ЭВМ по признакам наличия [параллелизма](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%BB%D0%BB%D0%B5%D0%BB%D0%B8%D0%B7%D0%BC_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) в потоках команд и данных

* ОКОД — Вычислительная система с **о**диночным потоком **к**оманд и **о**диночным потоком **д**анных  
  ([SISD](https://ru.wikipedia.org/wiki/SISD), **S**ingle **I**nstruction stream over a **S**ingle **D**ata stream).
* ОКМД — Вычислительная система с **о**диночным потоком **к**оманд и **м**ножественным потоком **д**анных  
  ([SIMD](https://ru.wikipedia.org/wiki/SIMD), **S**ingle **I**nstruction, **M**ultiple **D**ata).
* МКОД — Вычислительная система со **м**ножественным потоком **к**оманд и **о**диночным потоком **д**анных  
  ([MISD](https://ru.wikipedia.org/wiki/MISD), **M**ultiple **I**nstruction **S**ingle **D**ata).
* МКМД — Вычислительная система со **м**ножественным потоком **к**оманд и **м**ножественным потоком **д**анных  
  ([MIMD](https://ru.wikipedia.org/wiki/MIMD), **M**ultiple **I**nstruction **M**ultiple **D**ata).

Классификация машин MIMD-архитектуры:

* Переключаемые — с общей памятью и с распределённой памятью.
* Конвейерные.
* Сети — регулярные решётки, гиперкубы, иерархические структуры, изменяющие конфигурацию.

Основной характеристикой при классификации параллельных вычислительных систем является способ организации памяти :  
  
общая память   
распределенная память - каждый процессор имеет собственную локальную памятью, и прямой доступ к этой памяти других процессоров невозможен.

1. Системы с общей и распределенной памятью? Преимущества и недостатки?

В системах с общей памятью все процессоры имеют равные возможности по доступу к единому адресному пространству. Единая память может быть построена как одноблочная или по модульному принципу, но обычно практикуется второй вариант.

Преимущества таких систем: Не требуется обмена данными: данные, помещённые в память одним процессором, автоматически становятся доступными другим процессорам. Соответственно, система не должна тратить время на пересылку данных

В системе с распределенной памятью каждый процессор обладает собственной памятью и способен адресоваться только к ней. Модели архитектур с распределенной памятью принято обозначать как архитектуры без прямого доступа к удаленной памяти (NORMA, No Remote Memory Access). Такое название следует из того факта, что каждый процессор имеет доступ только к своей локальной памяти. Доступ к удаленной памяти (локальной памяти другого процессора) возможен только путем обмена сообщениями с процессором, которому принадлежит адресуемая память.

1. Способы распараллеливания?

Распараллеливание программ - процесс адаптации алгоритмов, записанных в виде программ, для их эффективного исполнения на вычислительной системе параллельной архитектуры.

Распараллеливание программ проводят двумя способами:

Первый способ — представление алгоритма задачи в виде частично-упорядоченной последовательности выполняемых работ.

Второй способ распараллеливания — по информации — используется тогда, когда можно распределить обрабатываемую информацию между процессорами для обработки по идентичным алгоритмам (по одному алгоритму)

1. Dataflow архитектура? Преимущества и недостатки?

В архитектурах с управлением потоком данных (Dataflow) отсутствует понятие «последовательность инструкций», нет Instruction Pointer'а, отсутствует даже адресуемая память в привычном нам смысле. Программа в потоковой системе — это не набор команд, а вычислительный граф. Каждый узел графа представляют собой оператор или набор операторов, а ветви отражают зависимости узлов по данным.

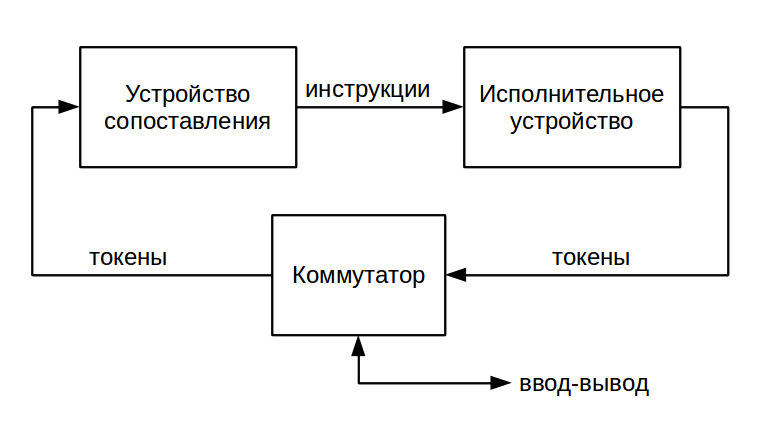
один из основных принципов dataflow: исполнение инструкций по готовности данных.

Одним из основных достоинств dataflow-архитектуры является ее масштабируемость: не составляет труда собрать систему, содержащую множество устройств сопоставления и исполнительных устройств.

1. Аппаратная реализация Dataflow?

В потоковых машинах данные передаются и хранятся в виде т.н. *токенов* (token). Токен — это структура, содержащая собственно передаваемое значение и *метку* — указатель узла назначения. Простейшая потоковая вычислительная система состоит из двух устройств: исполнительного (execution unit) и устройства сопоставления (matching unit).

Исполнительное устройство служит для выполнения инструкций и формирования токенов с результатами операций. Как правило, оно включает в себя память команд, доступную только для чтения. Готовность входных данных узла определяется по наличию набора токенов с одинаковыми метками. Для поиска таких наборов и служит устройство сопоставления.



1. Статическая Dataflow?

В ней каждый вычислительный узел представлен в единственном экземпляре, число узлов заранее известно, также заранее известно число токенов, циркулирующих в системе. Роль устройства сопоставления здесь выполняла память взаимодействий (activity store). В ней хранились пары токенов вместе с адресом узла назначения, флагами готовности и кодом операции. Любой вычислительный узел в этой архитектуре имел только два входа и состоял из одного оператора. При обнаружении готовности обоих операндов устройство выборки (fetch unit) считывало код операции, и данные отправлялись на обработку в исполнительное устройство (operation unit).

1. Динамическая Dataflow с помеченными токенами?

Системы с помеченными токенами (tagged-token architecture) в известной мере свободны от основного недостатка статической модели. В них число токенов, одновременно присутствующих на дуге, не ограничивается. Это открывает возможность выполнения итераций в произвольной последовательности, но требует учета принадлежности токенов к одной и той же итерации. Каждый токен содержит *тег*(фишку), состоящий из адреса команды, для которой предназначено заключенное в токене значение, и другой информации, определяющей вычислительный контекст, в котором данное значение используется, например номера итерации цикла. Этот контекст называется «цветом значения», а токен – «окрашенным», в силу чего метод имеет еще одно название –*метод окрашенных токенов*. Каждая дуга потокового графа может рассматриваться как вместилище, способное содержать произвольное число токенов с различными тегами. Правило активирования вершины в модели с помеченными токенами имеет вид: вершина активируется, когда на всех ее входных дугах присутствуют токены с идентичным цветом.

1. Динамическая Dataflow с явно адресуемыми токенами?

все токены в одной и той же итерации цикла имеют идентичный тег (цвет). При инициализации очередной итерации цикла формируется кадр токенов, содержащий токены, относящиеся к данной итерации или данному обращению, т.е. с одинаковыми тегами.

В схеме с явной адресацией токенов любое вычисление полностью описывается указателем команды (IP, Instruction Pointer) и указателем кадра (FP, Frame Pointer).

Команды, реализующие потоковый граф, хранятся в памяти команд и имеют формат: орс•i•dis. Здесь i (индекс в памяти кадров) определяет положение ячейки с нужным токеном внутри кадра, то есть какое число нужно добавить к FP, чтобы получить адрес интересующего токена. Поле dis указывает на местоположение команды, которой должен быть передан результат обработки данного токена. Адрес в этом поле также задан в виде смещения - числа, которое следует прибавить к текущему значению IP, чтобы получить исполнительный адрес команды назначенияв памяти команд. Если потребителей токена несколько, в поле dis заносится несколько значений смещения.

Каждому слову в памяти кадров придан бит наличия, единичное значение которого удостоверяет, что в ячейке находится токен, ждущий согласования, то есть что одно из искомых значений операндов уже имеется.

1. Гибридные Dataflow?

«Чистые» потоковые архитектуры, имели много слабых мест:

* на последовательных участках вычислительного графа они показывали резкое падение производительности.
* Б*о*льшая часть машинного времени тратилась на поиск соответствия операндов, выборку инструкций, а исполнительное устройство все это время простаивало, выполняя лишь по одной инструкции на каждую пару токенов.
* Сам принцип управления потоком данных не позволял организовать эффективный конвейер. Почти все устройства работали асинхронно, требовались буферы и очереди в линиях связи.

В попытках решить перечисленные проблемы стали появляться гибридные архитектуры, сочетающие в себе элементы как архитектур потока данных, так и потока управления.

1. Реконфигурируемые системы? Преимущества и недостатки

Реконфигурируемые вычислительные системы (РВС) — это системы, имеющие возможность менять свою модель вычислений, или, иначе говоря, позволяющие вносить существенные изменения в свою аппаратную часть.

Зачем нужны реконфигурируемые вычисления?

* • создание аппаратных блоков, обладающих свойством реконфигурируемости;
* • выявление участков приложения, которые требуется оптимизировать;
* • отображение оптимизируемых частей приложения на реконфигурируемую аппаратуру (создание оптимизированных аппаратных реализаций);
* • реконфигурация - способ записи новой конфигурации в аппаратные реконфигурируемые блоки;
* • включение реконфигурируемых блоков в вычислительный процесс.

Недостатки

* Принципиальная однозадачность. Для запуска новой задачи требуется остановка системы и перепрограммирование ПЛИС, входящих в ее состав.
* Сложность программирования.
* Избыточная аппаратная сложность.

Преимущества устройств конфигурируемой логики

Нет жесткой неизменяемой настройки и возможна разработка новых приложений после производства