



## Учебная дисциплина





### Вычислительные средства АСОИУ (часть 17)




### Подсистемы суперскалярных микропроцессоров


# Термины и определения

---

-  Процессор с одним конвейером называется скалярным.
-  Суперскалярный ЦП имеет более одного конвейера.




---

 Суперскалярная обработка основывается на способности процессора выполнять более одной простой операции за один такт.

# Первые ЦП с конвейером

---

 Первой успешно реализованной ЭВМ, построенной по указанным принципам, является ЭВМ «Эльбрус-1», созданная в 1978 г. в ИТМиВТ им. С.А.Лебедева. Архитектурные решения, заложенные в эту ЭВМ, используются в современных микропроцессорах.








# Исполнение команд на конвейере




# Стадии команды RISC ЦП

---

-  получение инструкции (англ. *Instruction Fetch*);
-  декодирование инструкции (англ. *Instruction Decode*) и чтение регистров (англ. *Register fetch*);
-  выполнение (англ. *Execute*);
-  доступ к памяти (англ. *Memory access*);
-  запись в регистр (англ. *Register write back*);

# Конфликт конвейера


---

 При написании ассемблерного кода (либо разработке компилятора, генерирующего последовательность инструкций) делается предположение, что результат выполнения инструкций будет точно таким, как если бы каждая инструкция заканчивала выполняться до начала выполнения следующей за ней.





# Конфликт конвейера




---

 Ситуация, когда одновременное выполнение нескольких инструкций может привести к логически некорректной работе конвейера, известна как «конфликт конвейера (англ. *Pipeline hazard*)».

# Типы конфликтов


---


-  Наибольшую задержку в работе конвейера вызывают команды условных переходов, так как только после полного их выполнения становится
-  возможным определить адрес следующей исполняемой команды.

- 
- 
-  В современных микропроцессорах для сокращения потерь времени при обработке таких команд широкое применение нашли
  -  предсказание перехода и исполнение альтернативных ветвей программы.


# Определение понятия конвейер

---

 **Конвéйер** — способ организации вычислений, используемый в современных процессорах и контроллерах с целью повышения их производительности (увеличения числа инструкций, выполняемых в единицу времени), технология, используемая при разработке компьютеров и других цифровых электронных устройств.





---


 Идея заключается в разделении обработки компьютерной инструкции на последовательность независимых стадий с сохранением результатов в конце каждой стадии. Это позволяет управляющим цепям процессора получать инструкции со скоростью самой медленной стадии обработки, однако при этом намного быстрее, чем при выполнении эксклюзивной полной обработки каждой инструкции от начала до конца.

# Обработка команды за 4 шага


---

 Примем обозначения стадий обработки:

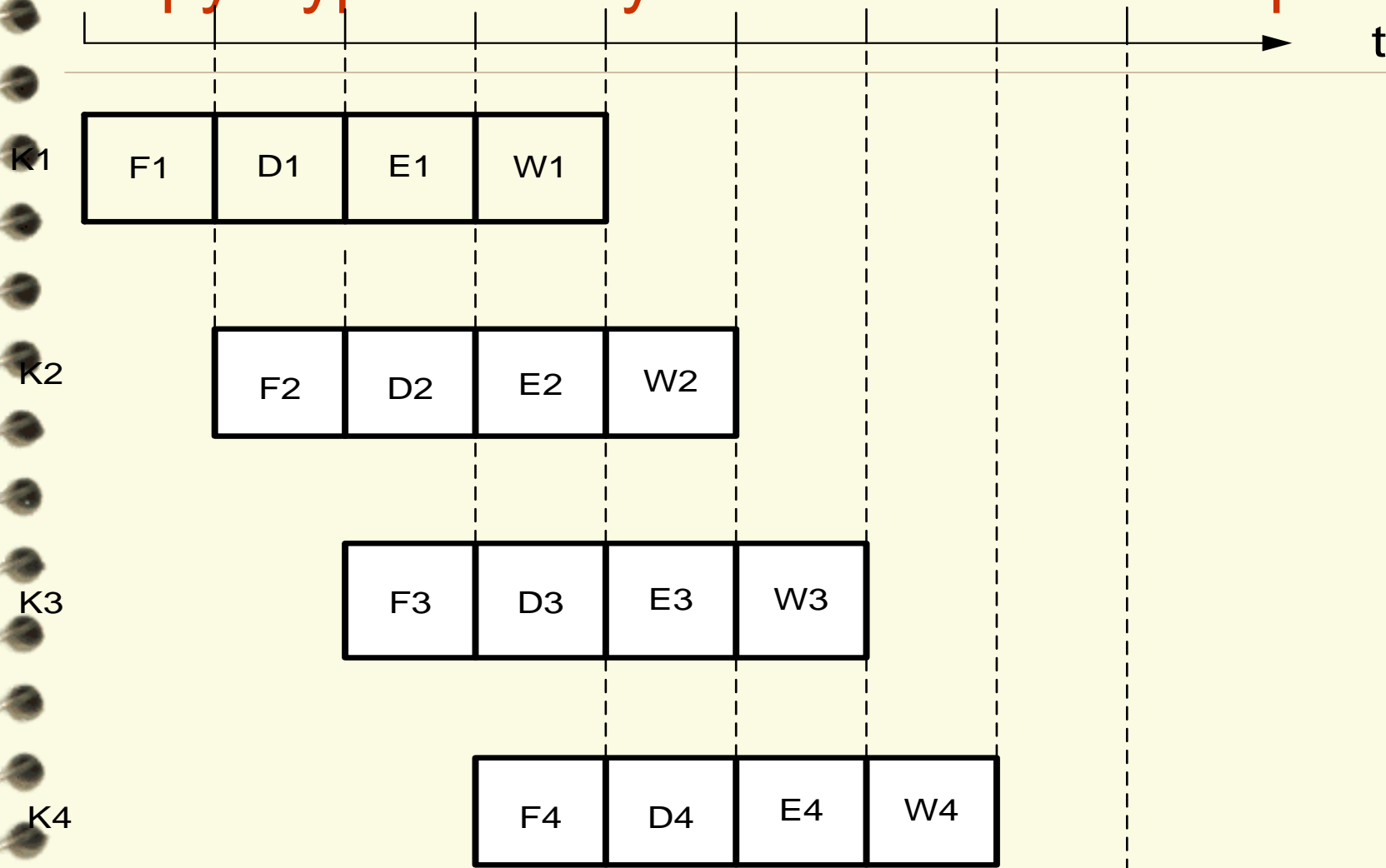
 F – выборка, т.е. чтение команды из памяти;

 D – декодирование команды и выборка её исходных операндов;

 E – выполнение операции


 W – запись, т.е. сохранение результата по целевому адресу.

# Структура 4-х ступенчатого конвейера



# Производительность конвейерной обработки


---

 При написании ассемблерного кода (либо разработке компилятора, генерирующего последовательность инструкций) делается предположение, что результат выполнения инструкций будет точно таким, как если бы каждая инструкция заканчивала выполняться до начала выполнения следующей за ней.




# Производительность конвейерной обработки


---

 Использование конвейера сохраняет справедливость этого предположения, однако не обязательно сохраняет порядок выполнения инструкций. Ситуация, когда одновременное выполнение нескольких инструкций может привести к логически некорректной работе конвейера, известна как «конфликт конвейера»

# Конфликт конвейера

---

 Обстоятельства, в силу которых приостанавливается работа конвейера, называют конфликтами.

 Типы конфликтов:

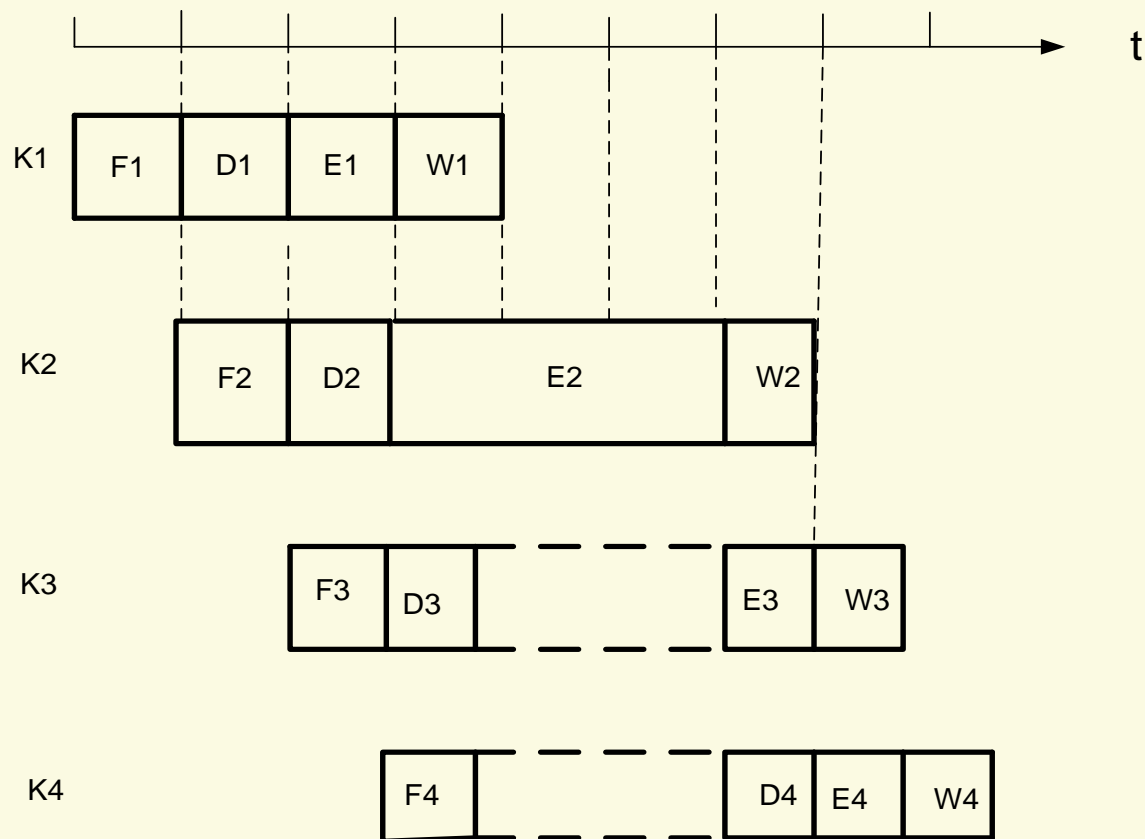
 - конфликты по управлению,

 - конфликты структурные,

 - конфликты по данным,



 - конфликты по условным переходам

# Конфликт по управлению

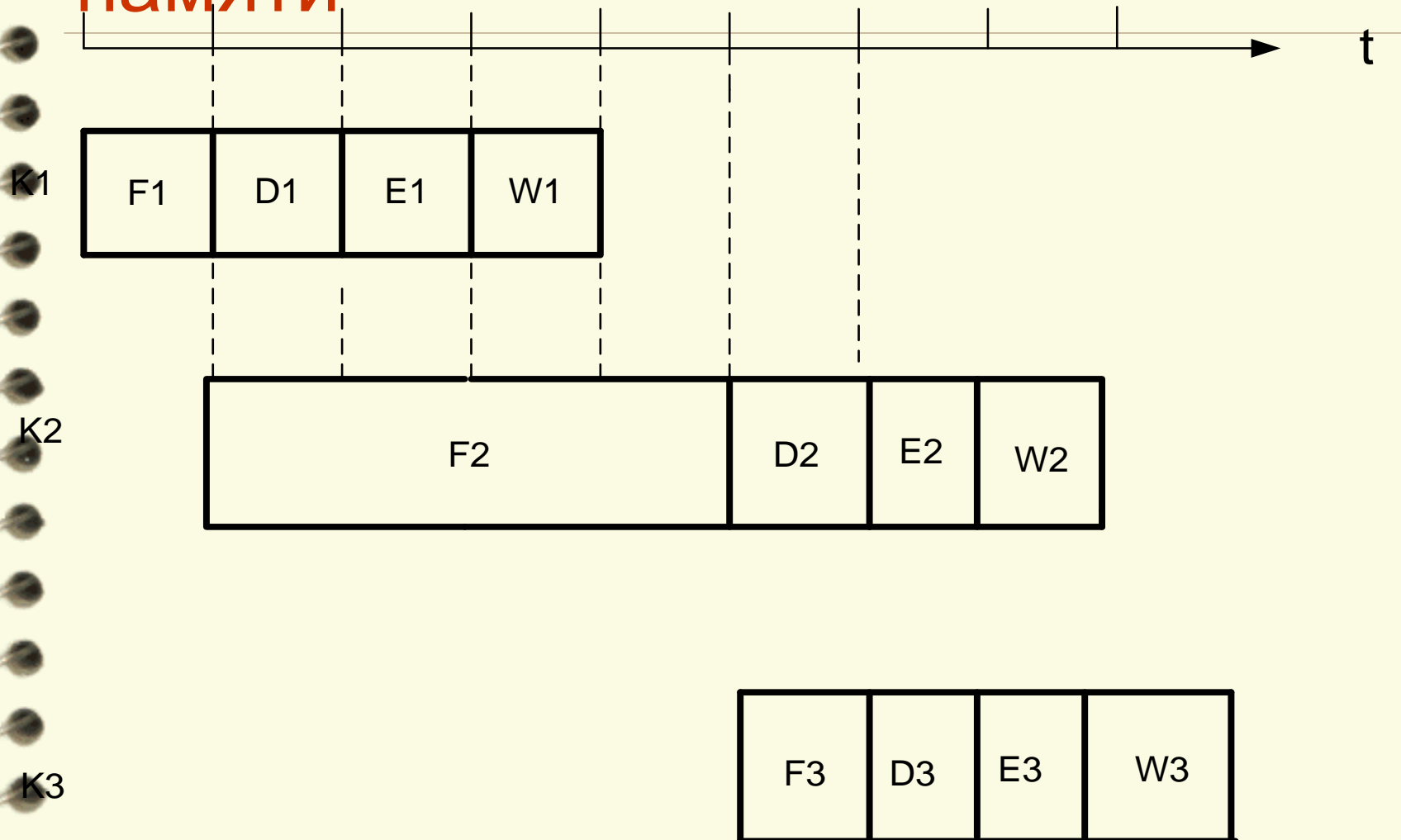


# Конфликт по промаху в кэш-памяти

---



-  Задержка происходит из-за своевременного поступления команды из-за отсутствия её в кэш-памяти.
-  Блок выборки команды K2 занимает длительность четырёх тактов.

# Конфликт по промаху в кэш-памяти

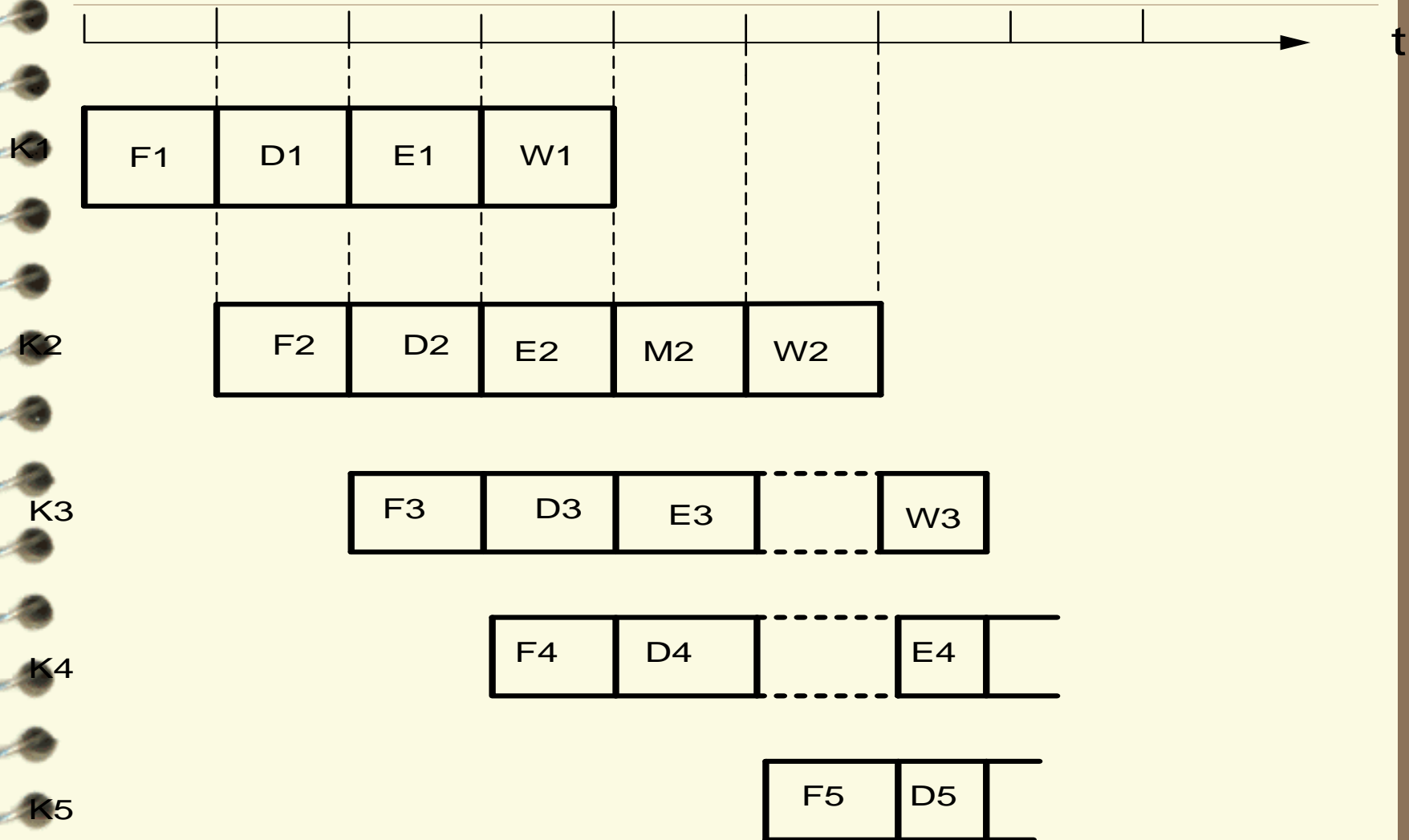


# Структурные конфликты

---

-  Структурные конфликты возникают, когда двум командам требуется одновременный доступ к аппаратному ресурсу.
-  Структурные конфликты наиболее вероятны при обращении к памяти.

# Структурные конфликты

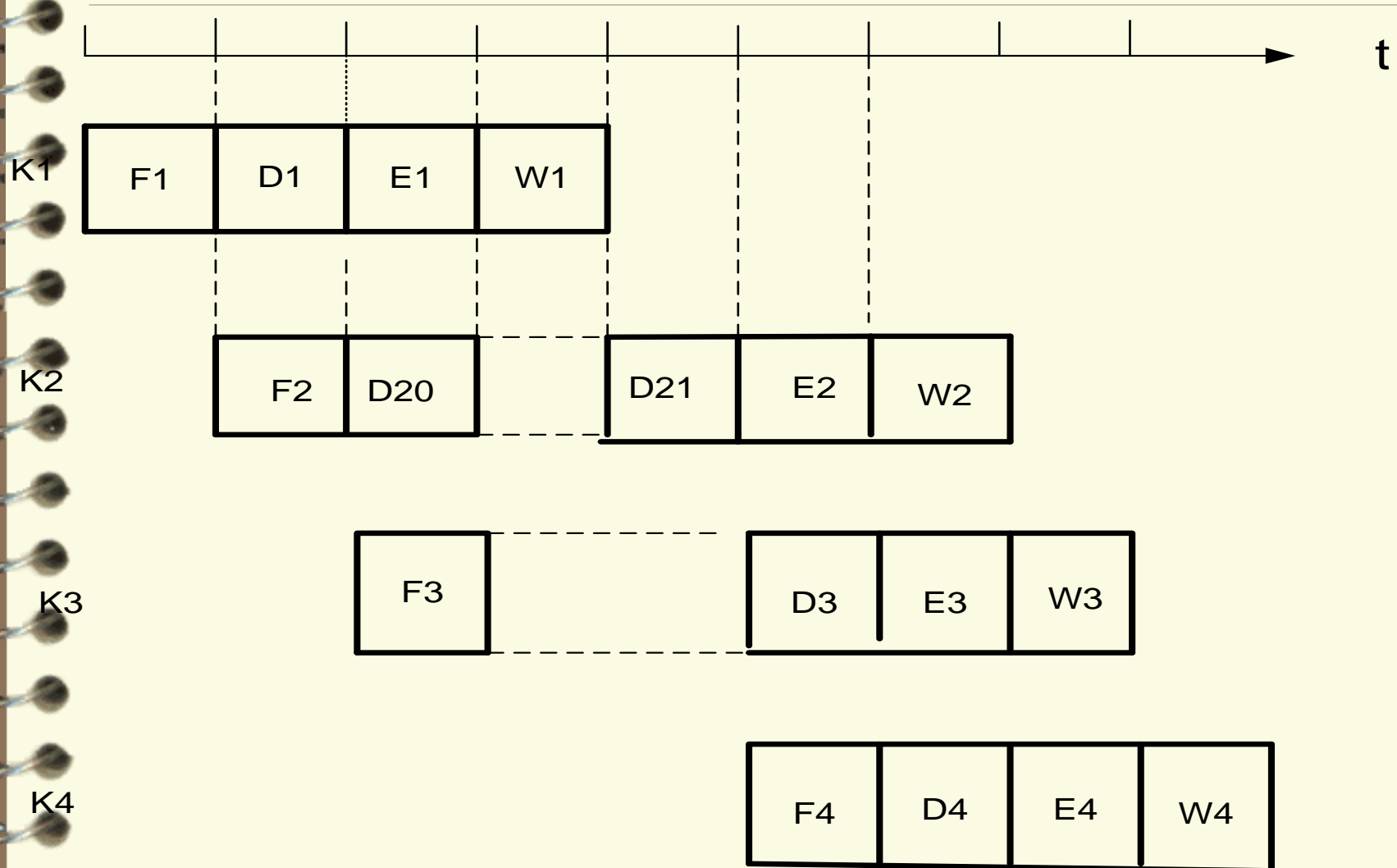


# Задержка конвейера из-за конфликта по данным

- Конфлик по данным – это ситуация когда конвейер останавливается из-за отсутствия данных, над которыми осуществляется операция.
- Рассмотрим следующие операции:
  - $A := 3 + A;$
  - $B := 4 * A$
- Конвейерное выполнение второй команды не успеет получить изменённое значение  $A$  в первой команде.




# Задержка конвейера из-за конфликта по данным




## Продвижение операндов

---

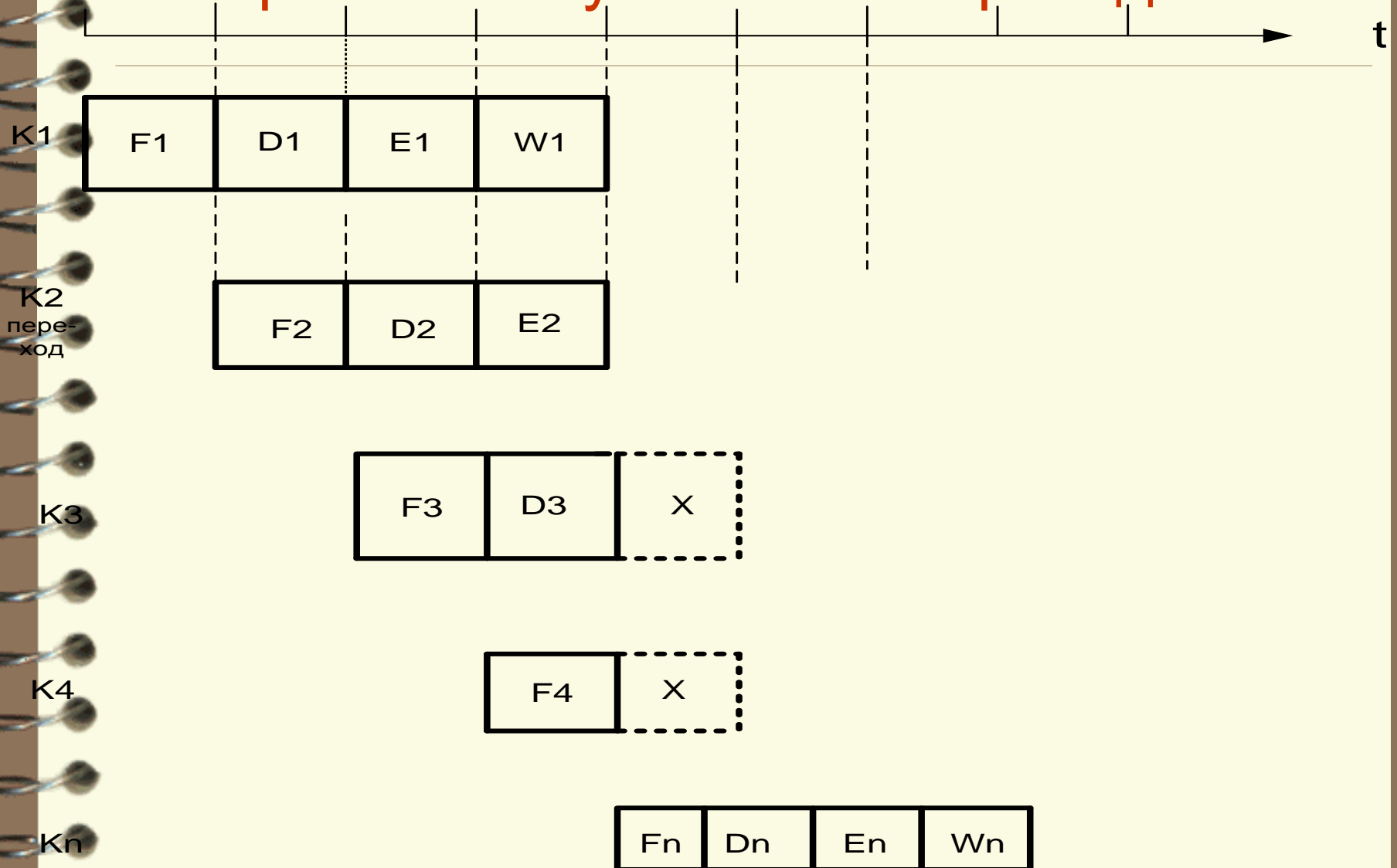
 Конфликт по данным обусловлен тем, что одна из команд, K2 ожидает запис данных в регистровый файл. Эти данные появляются на выходе АЛУ по завершении шага E1. Поэтому задержку можно сократить и даже устранить, передав результаты команды K1 непосредственно команде K2.

## Конфликты по условным переходам

---

 Временные потери, понесённые из-за команды перехода, называют накладными расходами перехода. Для сокращения расходов перехода адрес перехода нужно вычислять раньше. Обычно в блоке выборки команды имеется выделенная подсистема, предназначенная для быстрого выявления команд перехода и вычисления их целевых адресов.

# Конфликты по условным переходам



# Методы преодоления конфликтов по условным переходам

---


 Метод выжидания;


 Метод возврата;

 Задержанные переходы.

# Метод выжидания



---

 Подавление операций в конвейере, путем блокировки выполнения любой команды, следующей за командой условного перехода, до тех пор, пока не станет известным направление перехода.

 Преимущество: простота реализации.


# Метод возврата

---

-  Прогнозировать условный переход как невыполняемый.
-  При этом аппаратура должна просто продолжать выполнение программы, как если бы условный переход вовсе не выполнялся.

# Метод возврата


---

 Однако, если условный переход на самом деле выполняется, то необходимо просто очистить конвейер от команд, выбранных вслед за командой условного перехода и заново повторить выборку команд





# Метод возврата

---

 Альтернативная схема прогнозирует переход как выполняемый. Как только команда условного перехода декодирована и вычислен целевой адрес перехода, мы предполагаем, что переход выполняемый, и осуществляем выборку команд и их выполнение, начиная с целевого адреса.


# Задержанные переходы

---

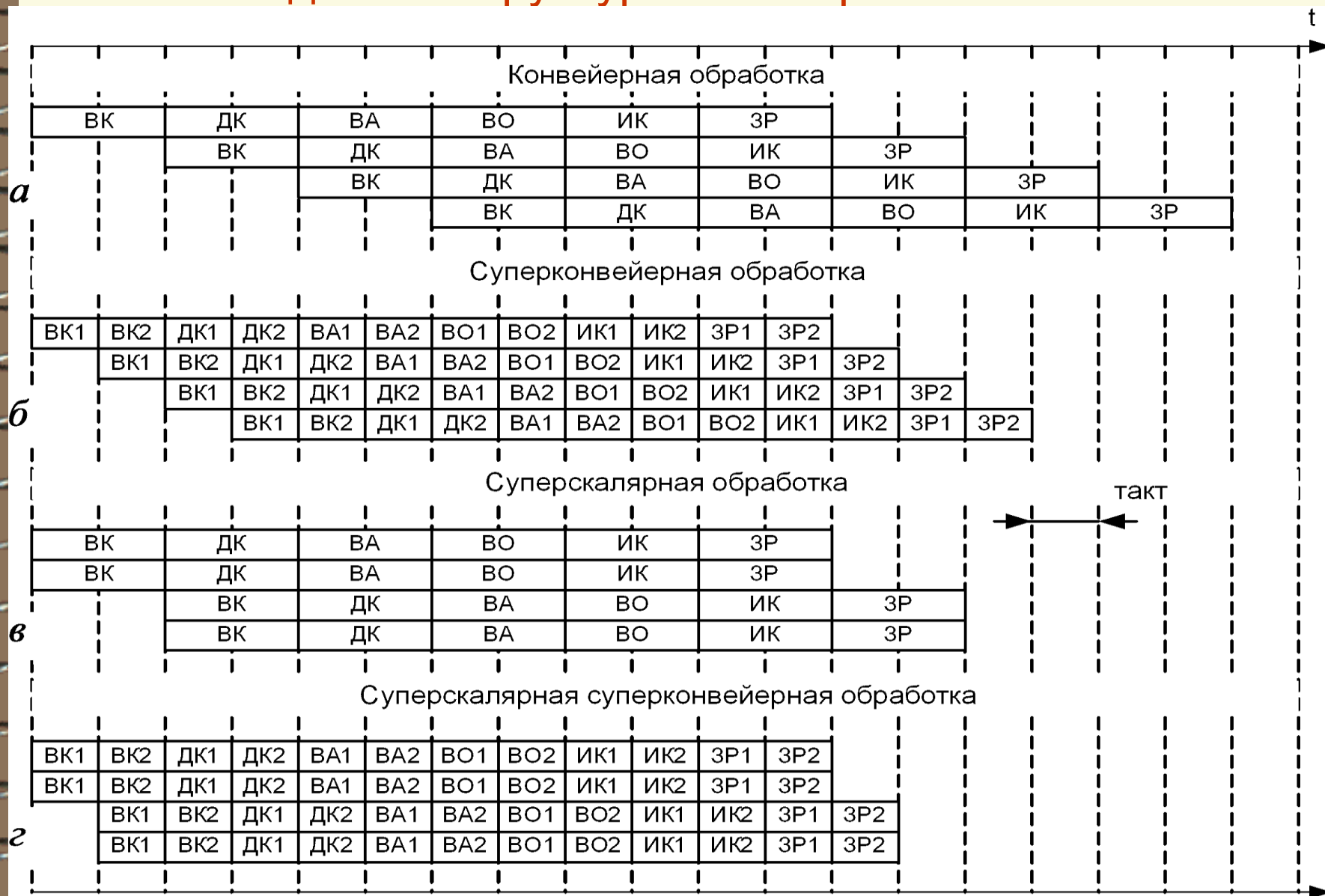
-  В задержанном переходе такт выполнения с задержкой перехода длиной  $n$  есть:
-  Команды 1 -  $n$  находятся в слотах (временных интервалах) задержанного перехода.

# Задержанные переходы

---

 Компилятор должен соблюдать требования при подборе подходящей команды для заполнения слота задержки. Если такой команды не находится, слот задержки должен заполняться пустой операцией.

# Разновидности структур конвейеров




# Функциональные подсистемы микропроцессоров

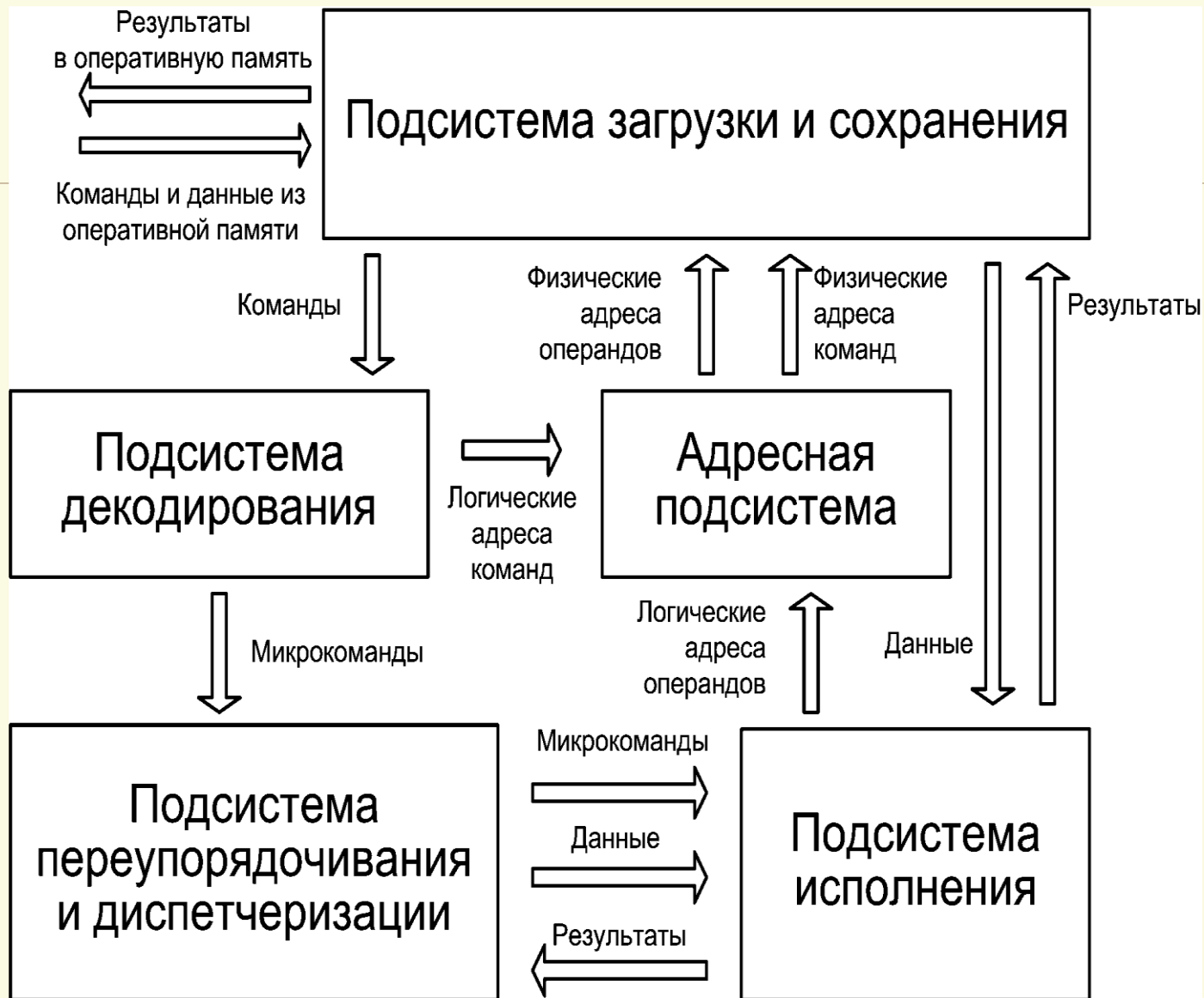
---

 Адресная подсистема

 Подсистема декодирования

 Подсистема переупорядочивания и диспетчеризации


 Подсистема исполнения




# Подсистема загрузки и сохранения

---


- Подсистема загрузки и сохранения процессоров состоит
- из устройства шинного интерфейса, кэш-памяти второго уровня, кэш-памяти команд первого уровня,
- кэш-памяти данных первого уровня, блока выборки команд и блока
- переупорядочивания запросов к памяти.




---


 Для процессоров AMD характерно разделение двунаправленной системной шины на несколько однонаправленных (технология HyperTransport).






---


 Инициаторами запросов, выдаваемых процессором на системную шину, могут являться *блок выборки команд*

 (Fetch Instruction Unit — FIU) и *буфер переупорядочивания запросов к памяти* (Memory Ordering Block — MOB).

# Адресная подсистема


---

 В функции подсистемы вычисления адресов процессоров входят:


-  • обеспечение виртуализации адресного пространства, реализуемое *буферами быстрого страничного преобразования, блоком связи с памятью и блоком вычисления адреса следующей команды;*




- предсказание направления ветвления, реализуемое *буфером меток перехода* и *блоком вычисления адреса следующей команды*.




---


 Более существенные задержки в работе конвейера вызывают команды условного перехода .

 Частичное решение этой проблемы достигается применением различных способов предсказания.


## Подсистемы декодирования, переупорядочивания и диспетчеризации


---

 Подсистема декодирования служит для определения последовательности микрокоманд, необходимых для реализации поступающей последовательности инструкций программы







---

 Состоит из предекодеров и декодеров инструкций, а также памяти микропрограмм и блока переименования регистров.



---

 Команды из кэш-памяти команд первого уровня L1I помимо *блока вычисления адреса следующей команды* поступают в блок декодирования, где выполняется *предобразование* каждой поступившей инструкции в последовательность микрокоманд.

- 
-  Преобразование машинных инструкций в последовательность
  -  микрокоманд, исполнение которых может происходить
  -  переупорядоченно.




# Конфликт по данным

---

- Если две инструкции программы используют
- одни и те же ячейки памяти, могут возникнуть три типа
- конфликтов по данным.

# Конфликт по данным

---

 Конфликт по данным типа

 «чтение после записи «

 «запись после чтения»»

 «запись после записи»»

# Конфликт по данным

---

- ❏ Для выявления и устранения конфликтов типов «чтение после записи» и «запись после чтения» применяется механизм
- ❏ переименования, основанный на использовании *регистров замещения*.




# Решение конфликтов

---

- Для хранения микрокоманд записи
- используется специальное устройство, называемое *буфером переупорядочивания запросов к памяти*.

# Буфер микрокоманд

---

-  Буфер микрокоманд, готовых к исполнению, служит для
-  временного хранения микрокоманд и их операндов перед направлением их
-  в освободившиеся исполнительные устройства.

# Порты запуска






---


- Отправка микрокоманд к исполнительным устройствам выполняется
- через специальные буферы, называемые *портами запуска*. Число
- портов и способ назначения исполнительных устройств
- определенным портам различаются в разных моделях процессоров.

# Подсистема исполнения


---

 Состоит из:


-  • целочисленных арифметико-логических устройств;
-  • устройств адресной арифметики;
-  • устройств обработки чисел с плавающей запятой;
-  • устройств выполнения целочисленных MMX-операций;
-  • устройств векторных вычислений над числами с плавающей запятой.

- 
- 
- ❏ Для ускорения выполнения арифметических операций с 32-
  - ❏ и 64-разрядными операндами в последующих моделях процессоров
  - ❏ Intel используется конвейеризация АЛУ совместно с увеличением
  - ❏ тактовой частоты работы ступеней





---

 Для этого применяется разделение на стадии, в каждой из которых используются АЛУ с меньшей разрядностью.




---

 Младшие

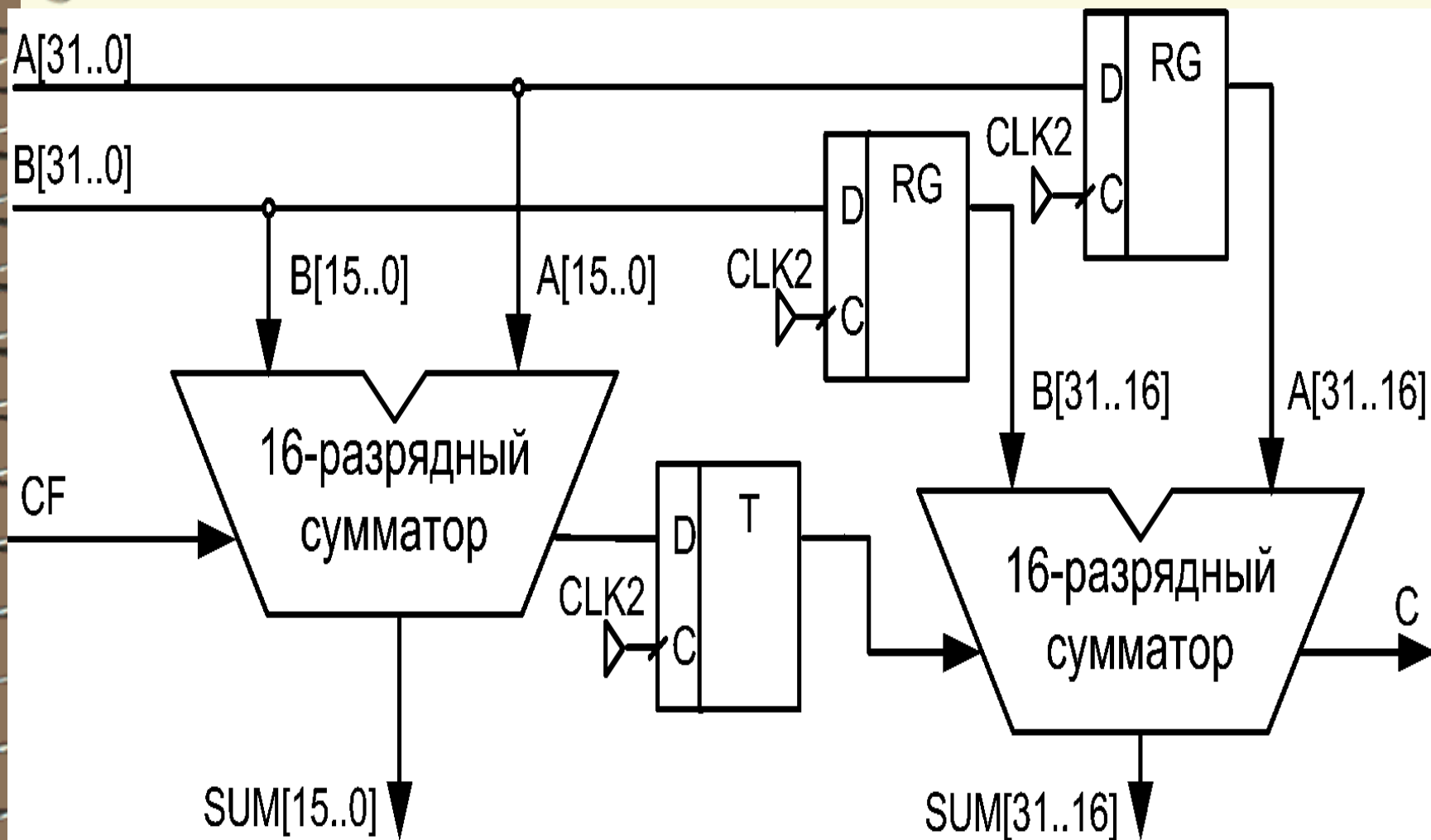
 16 разрядов числа суммируются (вычитаются) в первом такте,

 в то время как старшие разряды и перенос сохраняются в регистре.

 Старшие разряды чисел А и В обрабатываются в следующем


 такте одновременно с обработкой младших разрядов следующих операндов


.







# Ус-во умножения и FPU

---

 Устройства целочисленного умножения в современных процессорах основаны на древовидных умножителях.

 Устройство обработки чисел с плавающей запятой (ЧПЗ) процессоров Р6 функционирует в соответствии со стандартом IEEE 754.

# Ус-ва MMX и SSE

-  Устройство выполнения целочисленных MMX-операций (Multi-Media eXtensions) и SSE-операций (Streaming SIMD Extension)
-  предназначены для ускорения приложений, ориентированных на выполнение однотипных действий с большими массивами целочисленных и действительных данных.
-  С данными такого типа обычно работают мультимедийные, графические и коммуникационные
-  программы.

