

Архитектура ЭВМ и систем

Лекция № 4

Функциональная организация процессора

План лекции

1. Основные характеристики и классификация процессоров.
2. Физическая и функциональная структура процессора.
3. Производительность процессоров и архитектурные способы её повышения.

Функциональная организация процессоров

Процессор – это функционально-законченное программно-управляемое устройство, непосредственно осуществляющее процесс обработки цифровой информации в компьютере.

В большинстве случаев процессоры современных компьютеров реализуют на одном кристалле с использованием технологии СБИС. Соответствующую интегральную схему называют микропроцессором. Понятие микропроцессора в функциональном отношении совпадает с понятием процессор и отражает лишь особенности, связанные с его реализацией.

Основные характеристики и классификация процессоров

Процессор выполняет следующие функции:

- 1) вычисление адресов команд и операндов;
- 2) выборку и дешифрацию команд из оперативной памяти;
- 3) выборку данных из оперативной памяти, микропроцессорной памяти и регистров адаптеров внешних устройств;
- 4) приём и обработку запросов и команд от внешних устройств;
- 5) обработку данных и их запись в оперативную память, регистры микропроцессора и регистры адаптеров внешних устройств;
- 6) выработку управляющих сигналов для всех прочих узлов и блоков компьютера;
- 7) переход к следующей команде.

Основные характеристики и классификация процессоров

Основные характеристики процессора:

1) *Разрядность внутренних регистров* – количество бит, которые процессор способен обработать за один приём.

Разрядность шины данных определяет количество разрядов, над которыми одновременно могут выполняться операции.

Разрядность шины адреса определяет объём памяти (адресное пространство), с которым может работать процессор.

Адресное пространство – это максимальное количество ячеек памяти, которое может быть непосредственно адресовано микропроцессором.

2) *Рабочая тактовая частота (МГц)* определяет быстродействие процессора, поскольку каждая команда выполняется за определённое число тактов. Чем короче машинный такт, тем выше производительность процессора.

Быстродействие компьютера также зависит и от тактовой частоты шины системной платы, с которой работает процессор.

Основные характеристики и классификация процессоров

3) *Кэш-память*, устанавливаемая на плате микропроцессора, имеет два уровня:

3.1) L_1 – память первого уровня, находящаяся внутри основной микросхемы (ядра) процессора и работающая всегда на полной частоте процессора (впервые появилась в микропроцессорах Intel 386SLC и 486).

3.2) L_2 – память второго уровня, кристалл, размещаемый на плате микропроцессора и связанный с ядром внутренней шиной (впервые введена в микропроцессорах Pentium II). Эта память может работать на полной или половинной частоте процессора.

3.3) L_3 – память третьего уровня; размещается на плате процессора (впервые введена в микропроцессорах Pentium 4 Extreme Edition). Эта память может работать на полной или половинной частоте процессора.

В современных процессорах кэш-памяти всех уровней работают на полной частоте процессора.

Основные характеристики и классификация процессоров

4) *Состав инструкций* – перечень, вид и тип команд, автоматически выполняемых микропроцессором. Определяет непосредственно те процедуры, которые могут выполняться над данными и те категории данных, над которыми могут выполняться эти процедуры. Существенное изменение состава инструкций произошло в микропроцессорах Intel 80386 (этот состав принят за базовый), Pentium MMX, Pentium III, Pentium 4.

5) *Конструктив* подразумевает те физические разъёмные соединения, в которые устанавливается микропроцессор. Разные разъёмы имеют различную конструкцию (щелевой разъём – *Slot*, разъём-гнездо – *Soket*), разное количество контактов.

Основные характеристики и классификация процессоров

Классификация процессоров.

1. По назначению:

1.1. **Универсальные** - предназначены для решения широкого круга задач, в системе команд заложена алгоритмическая универсальность.

1.2. **Специализированные** - предназначены для решения определённого круга задач или даже одной задачи, имеют ограниченный набор команд (например, *процессоры для обработки данных, математические процессоры, микроконтроллеры*).

2. По количеству выполняемых программ:

2.1. **Однопрограммные** - переход к выполнению следующей программы происходит только после завершения текущей программы.

2.2. **Многопрограммные** – одновременно выполняются несколько программ.

Основные характеристики и классификация процессоров

3. По *структурному признаку*:

3.1. **С фиксированной разрядностью** – имеют строго определённую разрядность.

3.2. **С наращиваемой разрядностью** – позволяют секциями увеличивать число разрядов.

4. По *числу БИС (СБИС) в микропроцессорном комплекте*:

4.1. **Однокристалльные** – все аппаратные части процессора реализованы в виде одной БИС (СБИС); возможности таких процессоров ограничены ресурсами кристалла и корпуса.

4.2. **Многокристалльные** – функционально законченные части логической структуры процессора реализуются в виде отдельной БИС или СБИС.

4.3. **Многокристалльные секционные** – функционально законченные части логической структуры процессора разбиваются на секции, которые реализованы в виде БИС.

5. По *разрядности обрабатываемой информации* микропроцессоры могут быть 4, 8, 12, 16, 24, 32 и 64-разрядными.

На практике наибольшее распространение имеют 32-разрядные процессоры; всё большее применение находят 64-разрядные процессоры.

Основные характеристики и классификация процессоров

6. По *технологии изготовления*:

6.1. **Изготовленные по униполярной технологии** – *p*-канальные (*p*-МОП), *n*-канальные (*n*-МОП), комплиментарные (КМОП) БИС.
(МОП – металл – окисел – проводник).

6.2. **Изготовленные по биполярной технологии** – БИС на базе *транзисторно-транзисторной логики (ТТЛ)*, *эмиттерно-связанной логики (ЭСЛ)* и *интегральной инжекторной логики (И²Л)*.

Вид технологии изготовления БИС во многом определяет степень интеграции микросхем, быстродействие, энергопотребление, помехозащищённость и стоимость процессоров.

По комплексу этих признаков можно отдать предпочтение микропроцессорам, выполненным по *n*-МОП и КМОП-технологиям, обеспечивающим высокую плотность компоновки, высокое быстродействие и относительно малую стоимость. ЭСЛ обеспечивает самое высокое быстродействие процессоров, но низкую плотность компоновки и высокое энергопотребление. Технология И²Л даёт усреднённые характеристики микропроцессоров.

Основные характеристики и классификация процессоров

7. По характеру системы команд:

7.1. **Процессоры с полным набором инструкций**, или **CISC-процессоры (Complex Instruction Set Command)** – имеют большой набор разноформатных команд, что позволяет применять эффективные алгоритмы решения задач, но усложняет схему процессора, и в общем случае не обеспечивает максимального быстродействия.

Архитектура CISC присуща классическим процессорам.

7.2. **Процессоры с сокращённым набором инструкций**, или **RISC-процессоры (Reduced Instruction Set Command)** – содержат набор простых, чаще всего встречающихся в программах инструкций. Более сложные команды автоматически собираются из простых команд. Все простые команды имеют одинаковый размер и выполняются за один машинный такт.

Примеры процессоров: PowerPC (Apple), PPC (IBM) и др.

7.3. **Процессоры со сверхбольшим командным словом**, или **VLIW-процессоры (Very Long Instruction Word)** – одна инструкция содержит несколько операций, которые должны выполняться параллельно. Задача распараллеливания решается во время компиляции программы.

Примеры процессоров: Itanium (Intel), McKinley (HP) и др.

Основные характеристики и классификация процессоров

8. По числу и способу использования внутренних регистров:

8.1. **Аккумуляторные процессоры** – это процессоры с одним регистром результата. Относительно просты аппаратно, упрощённый формат команд, но требуется предварительная загрузка операнда в аккумулятор.

8.2. **Многоаккумуляторные процессоры** – результат выполнения операции может сохраняться в любом регистре общего назначения или ячейке памяти. В командах оба операнда задаются явно, результат помещается на место одного из операндов.

8.3. **Стековые процессоры** – используется большой аппаратный стек, в котором определённым образом располагаются операнды. Команды извлекают операнды из вершины стека и туда же помещают результат операции.

Физическая и функциональная структура процессора

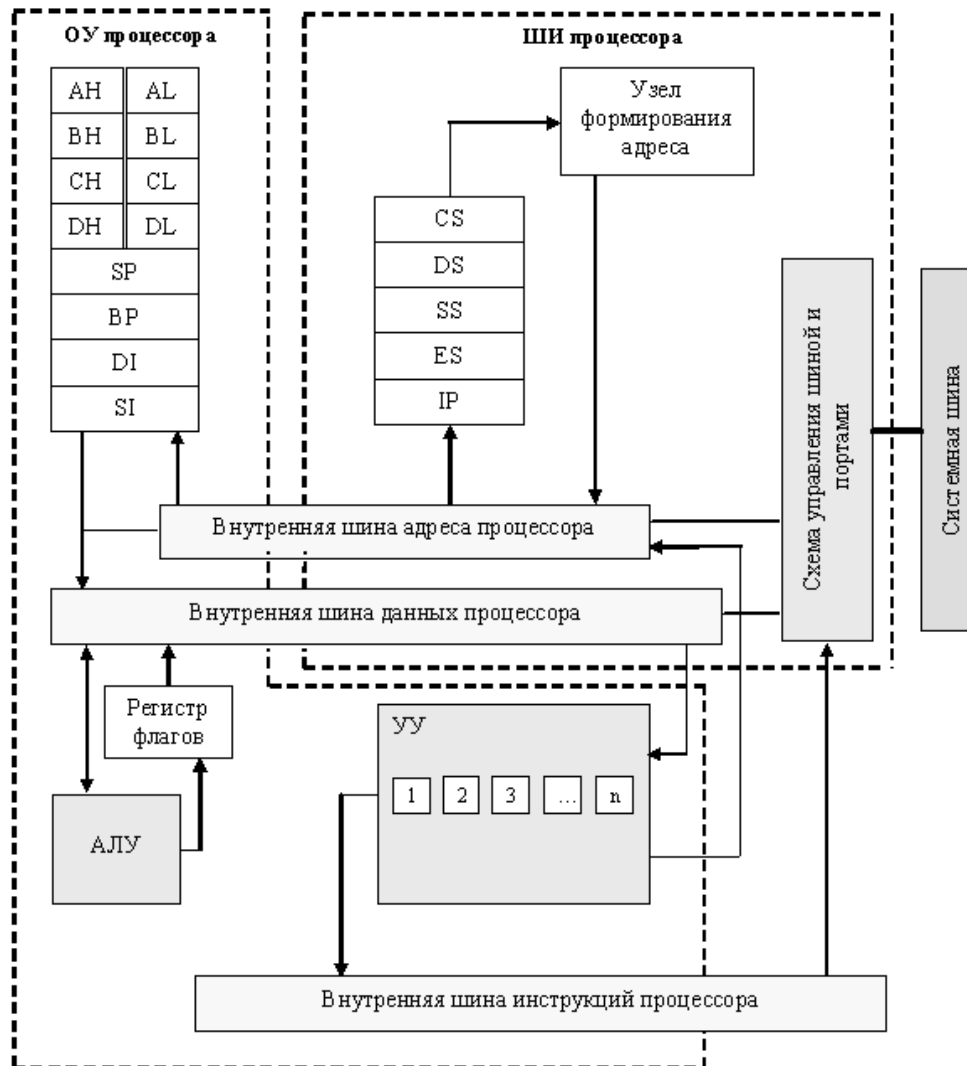
Физическая структура процессора является достаточно сложной.

Ядро процессора содержит главный управляющий и исполняющие модули – блоки выполнения операций над целочисленными данными. К локальным управляющим схемам относятся:

- блок с плавающей запятой,
- модуль предсказания ветвлений,
- регистры микропроцессорной памяти,
- регистры кэш-памяти,
- шинный интерфейс и многое другое.

Под **логическим ядром** понимается схема, по которой сделан процессор. **Физически ядро** представляет собой кристалл, на котором с помощью логических элементов реализована принципиальная схема процессора.

Физическая и функциональная структура процессора



Процессор состоит из двух частей: операционного устройства (ОУ) и шинного интерфейса (ШИ).

ОУ содержит:

- арифметико-логическое устройство (АЛУ),
- устройство управления (УУ),
- регистр флагов,
- регистры общего назначения (РОН),
- регистры-указатели,
- индексные регистры.

ШИ содержит:

- адресные регистры,
- блок регистров (буфер) команд,
- узел формирования адреса,
- схемы управления шиной и портами.

Обе части процессора работают параллельно, причём ШИ работает быстрее ОУ.

Упрощённая структура процессора

Физическая и функциональная структура процессора

УУ вырабатывает управляющие сигналы, поступающие во все блоки вычислительной машины. В составе УУ можно выделить следующие функциональные блоки:

1. *регистр команд* – запоминающий регистр, в котором хранится код команды: код операции и адреса операндов;
2. *дешифратор операций* – логический блок, который в соответствии с поступающим из регистра команд кодом операции выбирает один из множества имеющихся у него выходов;
3. *постоянное запоминающее устройство (ПЗУ) микропрограмм* хранит управляющие импульсы для выполнения в блоках ВМ процедур обработки информации; импульс по выбранному дешифратором операций проводу считывает из ПЗУ микропрограмм необходимую последовательность управляющих сигналов;
4. *узел формирования адреса* (располагается в ШИ) – устройство для вычисления полного адреса ячейки памяти (регистра) по реквизитам, поступающим из микропроцессорной памяти или регистра команд;
5. *кодовые шины данных, адреса и инструкций* – часть внутренней интерфейсной шины процессора.

Таким образом, УУ формирует управляющие сигналы для выполнения процессором своих функций (см. слайд № 4).

Физическая и функциональная структура процессора

АЛУ предназначено для выполнения арифметических и логических операций преобразования информации. Функционально в простейшем варианте АЛУ состоит из следующих компонент:

1. *сумматора* выполняющего процедуру сложения двоичных кодов, имеет разрядность двойного машинного слова (32 бита);
2. *регистров* – быстродействующих ячеек памяти различной длины: регистр 1 имеет разрядность 32 бита, регистр 2 - 16 бит; при сложении в регистр 1 помещается первое слагаемое, а потом результат, в регистр 2 помещается второе слагаемое;
3. *схемы управления*, принимающей по кодовым шинам инструкций управляющие сигналы от УУ и преобразующей в сигналы для управления работой регистров и сумматора.

АЛУ выполняет арифметические операции только над двоичными числами с фиксированной точкой. Для обработки чисел с плавающей точкой привлекается математический сопроцессор или специально составленные программы.

Физическая и функциональная структура процессора

Регистры ОУ – часть микропроцессорной памяти.

Рассмотрим регистры на примере базового процессора Intel 8086, который содержит всего 14 двухбайтовых регистра.

В состав ОУ входят следующие регистры:

1. *регистры общего назначения (РОН), или универсальные*: AX – (AH, AL), BX – (BH, BL), CX – (CH, CL), DX – (DH, DL). Могут использоваться для временного хранения любых данных, при этом можно работать с каждым регистром целиком, а можно отдельно, с каждой его половиной; но каждый из РОН может использоваться и как специальный при выполнении некоторых конкретных команд;
2. *регистры смещений*: SP, BP, SI, DI. Являются неделимыми и предназначены для хранения относительных адресов ячеек памяти внутри сегментов (смещений относительно начала сегментов);
 - 2.1. *SP (Stack Pointer)* – смещение вершины стека;
 - 2.2. *BP (Base Pointer)* – смещение начального адреса поля памяти, непосредственно отведённого под стек;
 - 2.3. *SI (Source Index), DI (Destination Index)* – предназначены для хранения адресов индекса источника и приёмника данных при операциях над строками и им подобных.

Физическая и функциональная структура процессора

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

Слово состояния процессора (PSW – Processor State Word), или регистр флагов, имеет размер 2 байта и содержит одноразрядные признаки или флаги. Свободные биты отведены для использования в будущем.

Всего в регистре девять флагов: шесть из них *условные* или *статусные*, отражают результаты операций, выполненных ОУ, остальные три – *управляющие*, определяют режим исполнения программы.

Физическая и функциональная структура процессора

Статусные флаги.

1. *CF (Carry Flag)* – флаг переноса. Устанавливается в 1, если при выполнении арифметических и некоторых операций сдвига возникает «перенос» из старшего разряда.
2. *PF (Parity Flag)* – флаг чётности. Проверяет младшие 8 битов результатов над данными. Чётное число единиц приводит к установке этого флага в 1, нечётное – в 0.
3. *AF (Auxiliary Carry Flag)* – флаг логического переноса в двоично-десятичной арифметике. Устанавливается в 1, если арифметическая операция приводит к переносу или займу четвёртого справа бита однобайтового операнда. Используется при арифметических операциях над двоично-десятичными кодами и кодами ASCII.
4. *ZF (Zero Flag)* – флаг нуля. Устанавливается в 1, если результат операции равен 0, в противном случае ZF обнуляется.
5. *SF (Sign Flag)* – флаг знака. Устанавливается в 1, если результат арифметической операции является отрицательным, в 0, если результат положительный.
6. *OF (Overflow Flag)* – флаг переполнения. Устанавливается в единицу при арифметическом переполнении, когда результат выходит за пределы разрядной сетки.

Физическая и функциональная структура процессора

Управляющие флаги.

1. *TF (Trap Flag)* – флаг трассировки. Единичное состояние этого флага переводит процессор в режим пошагового выполнения программы.
2. *IF (Interrupt Flag)* – флаг прерываний. При нулевом состоянии этого флага прерывания запрещены, при единичном – разрешены (о механизме прерываний речь пойдёт в следующей главе).
3. *DF (Direction Flag)* – флаг направления. Используется в строковых операциях для задания направления обработки данных; при единичном состоянии строки обрабатываются «справа налево», при нулевом – «слева направо».

Физическая и функциональная структура процессора

ШИ предназначен для связи и согласования микропроцессора с системной шиной компьютера, а также для приёма, предварительного анализа команд выполняемой программы и формирования полных адресов операндов и команд.

Сегментные (адресные) регистры совместно с *узлом формирования адреса* реализуют сегментацию памяти.

Поскольку команды и данные на уровне кодов неотличимы друг от друга, то для различия команд и данных используется их размещение в различных областях памяти – сегментах.

Сегмент – это прямоугольная область памяти, характеризующаяся начальным адресом и длиной.

Начальный адрес (адрес начала сегмента) – это номер (адрес) ячейки памяти, с которой начинается сегмент.

Длина сегмента – это количество входящих в него ячеек памяти.

Физическая и функциональная структура процессора

Все ячейки внутри сегмента перенумеровываются, начиная с нуля. Адресация ячеек внутри сегмента ведется относительно начала сегмента; адрес ячейки в сегменте называется **смещением**, или **эффективным адресом – ЕА** (относительно начального адреса сегмента).

Текущий сегмент можно указать с помощью загрузки соответствующего сегментного регистра:

Физическая и функциональная структура процессора

1. *CS (Code Segment)* – определяет начало текущего сегмента кода (располагаются команды программы). Выборка команды производится с использованием в качестве эффективного адреса содержимого регистра *IP (Instruction Pointer)*, а в качестве адреса сегмента – содержимого *CS*.

2. *DS (Data Segment)* – определяет начало текущего сегмента данных. Ссылки на данные (за некоторым исключением) осуществляются относительно содержимого этого регистра.

3. *SS (Stack Segment)* – определяет начало текущего сегмента стека. Как правило, все адреса данных, связанных со стеком, задаются относительно содержимого этого регистра.

4. *ES (Extended Segment)* – определяет начало дополнительного текущего сегмента, который обычно рассматривается как вспомогательный сегмент данных (при межсегментных пересылках).

Производительность процессоров и архитектурные способы её повышения

Производительность – количество вычислительной работы, выполняемой в единицу времени - одна из наиболее важных характеристик процессора.

Факторы, влияющие на производительность:

- тактовая частота,
- число команд программы,
- среднее время выполнения отдельной команды.

Для упрощённой оценки производительности процессора используют показатель, указывающий число команд, выполняемых за секунду:

- *MIPS (Meg Instructions Per Second)* для целочисленных задач;
- *MFLOPS (Meg Floating Point Operations Per Second)* для вычислений с плавающей точкой.

Однако этот показатель не учитывает особенности архитектуры конкретных процессоров. Поэтому для сравнительных характеристик различных процессоров используются относительные оценки производительности, для получения которых применяют специальные тестовые программы.

Производительность процессоров и архитектурные способы её повышения

Архитектурные способы повышения производительности процессоров:

1. *Увеличение «естественного» параллелизма* – повышение разрядности обработки и передачи данных (разрядность процессоров повысилась с 4 до 32 и 64 разрядов).
2. *Конвейерная (многофазная) обработка данных* – вычислительный процесс делится на несколько фаз, для каждой из которых используются свои средства и буфер для хранения результата (ступень конвейера).
3. *Многоэлементная обработка данных* – параллельная обработка данных в нескольких операционных блоках (ОУ) процессора.

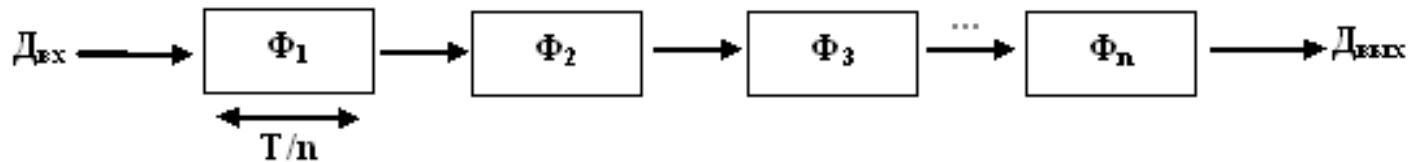
Производительность процессоров и архитектурные способы её повышения

При *многофазной* обработке процесс обработки данных разбивается на несколько стадий (фаз), выполняемых последовательно.

Между фазами имеются буферы для хранения промежуточных результатов.

После выполнения первой фазы результат запоминается в буфере и начинается обработка второй фазы. Средства выполнения первой фазы освобождаются, и на них поступает следующая порция данных.

Если длительность фаз обработки одинакова и составляет T/n , то при таком способе производительность системы увеличится в n раз. Этот способ соответствует конвейерной обработке (параллельной обработке во времени).



Многофазная (конвейерная) обработка данных

Производительность процессоров и архитектурные способы её повышения

Такты работы процессора

	1	2	3	4	5	6	7	8	9	10
Команда i	IF	ID	RD	EX	WB					
Команда $i+1$		IF	ID	RD	EX	WB				
Команда $i+2$			IF	ID	RD	EX	WB			
Команда $i+3$				IF	ID	RD	EX	WB		
Команда $i+4$					IF	ID	RD	EX	WB	
Команда $i+5$						IF	ID	RD	EX	WB

Рассмотрим организацию конвейера на уровне исполнения машинной команды. Типовые этапы выполнения команды:

1) выборка команды *IF* (*Instruction Fetch*), 2) дешифрация команды *ID* (*Instruction Decode*), 3) чтение операндов *RD* (*Read Memory*), 4) исполнение заданной в команде операции *EX* (*Execute*), 5) запись результата *WB* (*Write Back*).

В ходе выполнения команда продвигается по конвейеру, освобождая очередную ступень для следующей команды. Содержимое буферов, обновляется в каждом такте по завершению этапа исполнения очередной команды. Промежуточные буферы обеспечивают параллельную независимую работу блоков конвейерной цепочки: когда последующий блок начинает выполнять этап очередной команды, предыдущий блок может приступить к обработке следующей команды.

Производительность процессоров и архитектурные способы её повышения

Конвейерная обработка команд не уменьшает время выполнения отдельной команды, которое в конвейерном процессоре остаётся таким же, как и в обычном неконвейерном.

Однако благодаря тому, что при конвейерной обработке большая часть вычислительного процесса осуществляется в режиме одновременного выполнения команд, скорость выдачи результатов последовательно выполняемых команд увеличивается пропорционально числу ступеней конвейера.

В рассмотренном примере после завершения выполнения первой команды из памяти прочитана уже пятая, а не вторая, как при традиционном исполнении

Продолжительность выполнения отдельных этапов исполнения команды в общем случае зависит от типа команды и места размещения операндов.

Конвейерная обработка команд наиболее эффективна в том случае, если длительность всех фаз выполнения команды приблизительно одинаковая.

Производительность процессоров и архитектурные способы её повышения

На практике обеспечить непрерывную работу конвейера не всегда удаётся из-за различных конфликтов: по ресурсам, по данным, по управлению.

Конфликты по ресурсам возникают, когда нескольким командам с разных ступеней конвейера требуется одновременное использование одного и того же ресурса, например, памяти. Допустим, одной из команд конвейера требуется доступ к памяти для чтения операндов (RD), а другой – на этапе выборки команды (IF). При размещении команд и данных в одной памяти одновременный доступ к ней со стороны разных устройств невозможен.

Конфликт по данным возникает, когда или исходный операнд, или результат операции не доступны в положенное время. Такая ситуация возникает, когда команды имеют различную длину или взаимосвязаны по данным. В первом случае нарушается длительность фаз конвейера, что приводит к простоям отдельных ступеней. Во втором случае последующая команда вынуждена ждать завершения текущей, чтобы прочитать операнд из памяти, который и запишется текущей командой.

Конфликт по управлению возникает при наличии в программе команд передачи управления (условных и безусловных переходов), при возникновении запросов от устройств, исключений, нарушающих последовательное выполнение команд.

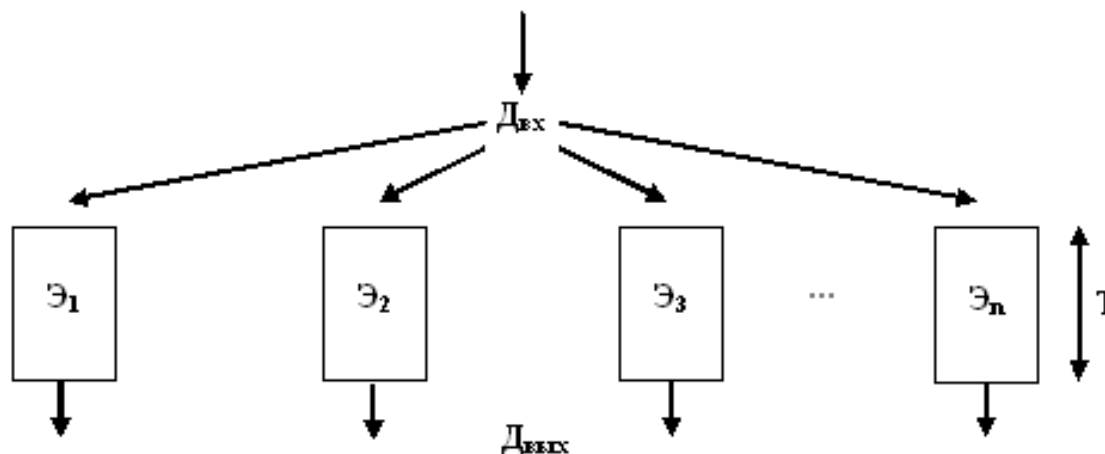
Производительность процессоров и архитектурные способы её повышения

Процессор, в котором процесс выполнения команды разбивается на 5-6 ступеней, называется обычным **конвейерным процессором**.

Если увеличить количество ступеней конвейера, то каждая отдельная ступень будет выполнять меньшую работу, а, следовательно, содержать меньше аппаратной логики. Благодаря более коротким задержкам распространения сигналов в каждой отдельно взятой ступени конвейера достигается повышение частоты работы и соответствующее повышение производительности процессора.

Процессор, имеющий конвейер существенно глубже 5 – 6-ти ступеней, называется **суперконвейерным**. Например, Pentium II содержит 12, UltraSPARC III – 14, Pentium 4 – 20 ступеней.

Производительность процессоров и архитектурные способы её повышения



Многоэлементная обработка осуществляется на нескольких параллельно работающих ОУ. Каждый элемент выполняет свою работу, осуществляя обработку порции данных от начала до конца.

Если время выполнения работы на отдельном элементе составляет T и в системе используется n элементов, то что среднее время выполнения такой работы составит T/n (реально – меньше).

В современных процессорах такой способ обработки связан с понятием *суперскалярной* архитектуры.

Простейшим примером вычислительного параллелизма является выполнение двух команд, операнды которых не связаны между собой:

$$\mathbf{A = B + C; D = E + F.}$$

Производительность процессоров и архитектурные способы её повышения

Для выполнения несвязанных операций в состав процессора включают набор арифметических устройств, каждое из которых обычно имеет конвейерную организацию.

Процессор, содержащий несколько ОУ, которые обеспечивает одновременное выполнение более одной скалярной команды, называется **суперскалярным** процессором.

Команда называется **скалярной**, если её входные операнды и результат являются числами (скалярами).

Традиционные процессоры с одним ОУ называются **скалярными**.

В суперскалярном процессоре обработка команд распараллелена не только во времени (конвейер), но и в пространстве (несколько конвейеров).

Производительность такого процессора оценивается темпом схода исполненных команд со всех его конвейеров.

Производительность процессоров и архитектурные способы её повышения

В настоящее время используются два способа суперскалярной обработки.

1. *Основывается на чисто аппаратном механизме выборки несвязанных команд программы из памяти и параллельном запуске их на исполнение.*

Трансляция программ для суперскалярного процессора ничем не отличается от трансляции программ для традиционного скалярного процессора. Сравнительно легко реализуются суперскалярные процессоры различных семейств, программно совместимые между собой. При этом не возникает проблем с использованием ранее созданного программного обеспечения. Все процессоры семейства Pentium реализованы по этому способу.

2. *Планирование параллельного исполнения нескольких команд возлагается на распараллеливающий компилятор.*

Он анализирует исходную программу и группирует такие команды, которые могут выполняться одновременно, в пакеты – длинные командные слова (VLIW). Для разных поколений процессоров требуется разработка новых версий компиляторов.

Идеи VLIW предложены российскими инженерами и учёными во главе с профессором Б.А. Бабаяном при разработке отечественной супер-ЭВМ «Эльбрус-3» (1990). В настоящее время VLIW-технология реализована в процессоре Эльбрус Е2К (Эльбрус Интернешнл), процессорах Crusoe (Transmeta), а также в семействе процессоров для цифровой обработки сигналов TMS320C60xx (Texas Instruments).