

Организация вычислительных систем

Операционные системы

Лекция №0

Ульяновск, УлГТУ, кафедра «Информационные системы»

Цифровой компьютер— устройство или система, способное выполнять заданную, четко определенную последовательность операций.

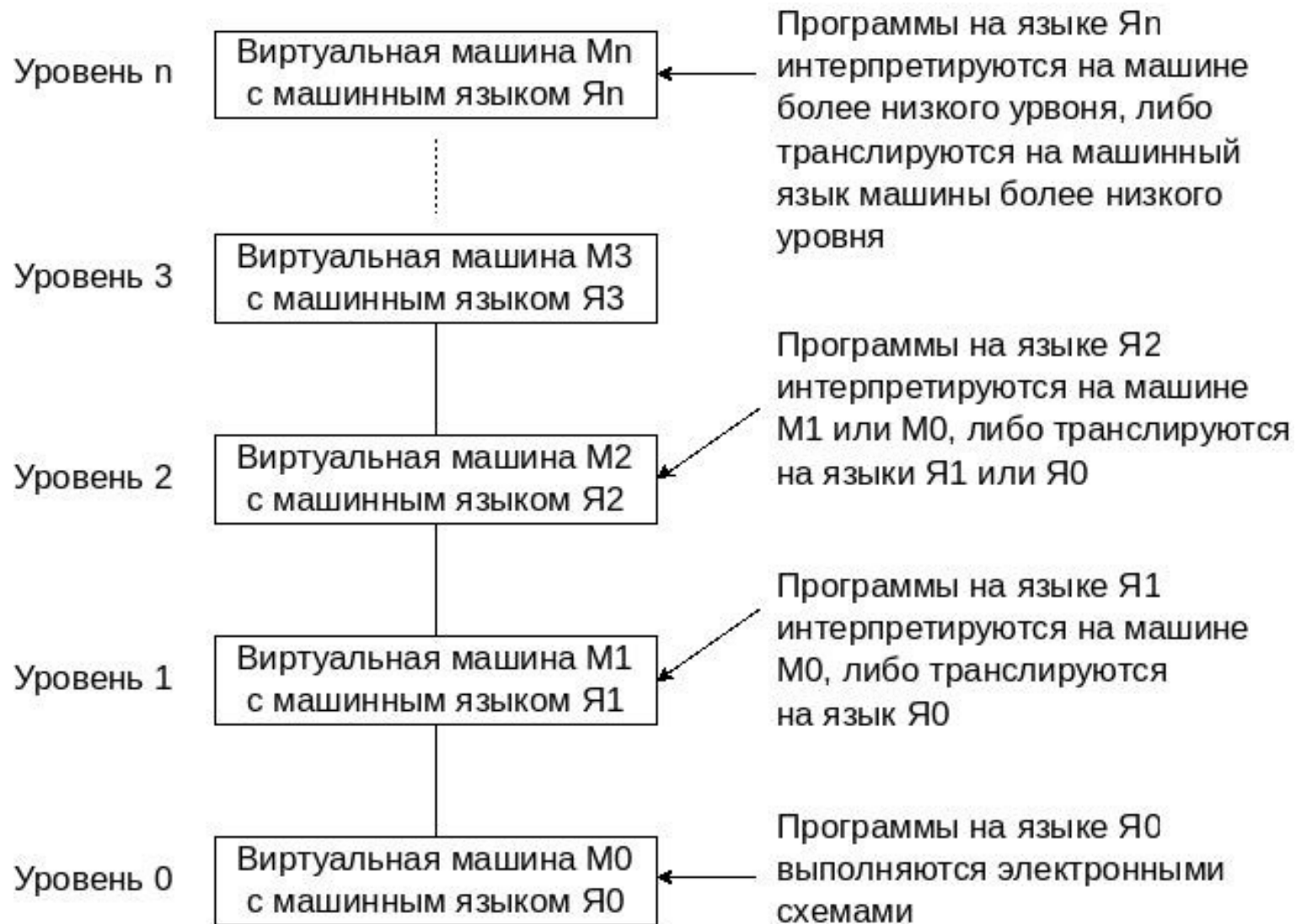
Программа — последовательность команд, описывающих решение определенной задачи.

Машинный язык — система команд (набор кодов операций) конкретной вычислительной машины, которая интерпретируется непосредственно процессором или микропрограммами этой вычислительной машины.

Трансляция — преобразование программы, представленной на одном из языков программирования, в программу на другом языке и, в определенном смысле, равносильную первой.

Интерпретация — покомандное выполнение исходного кода программы без предварительной компиляции.

Многоуровневая компьютерная организация (продолжение)



Современные многоуровневые машины



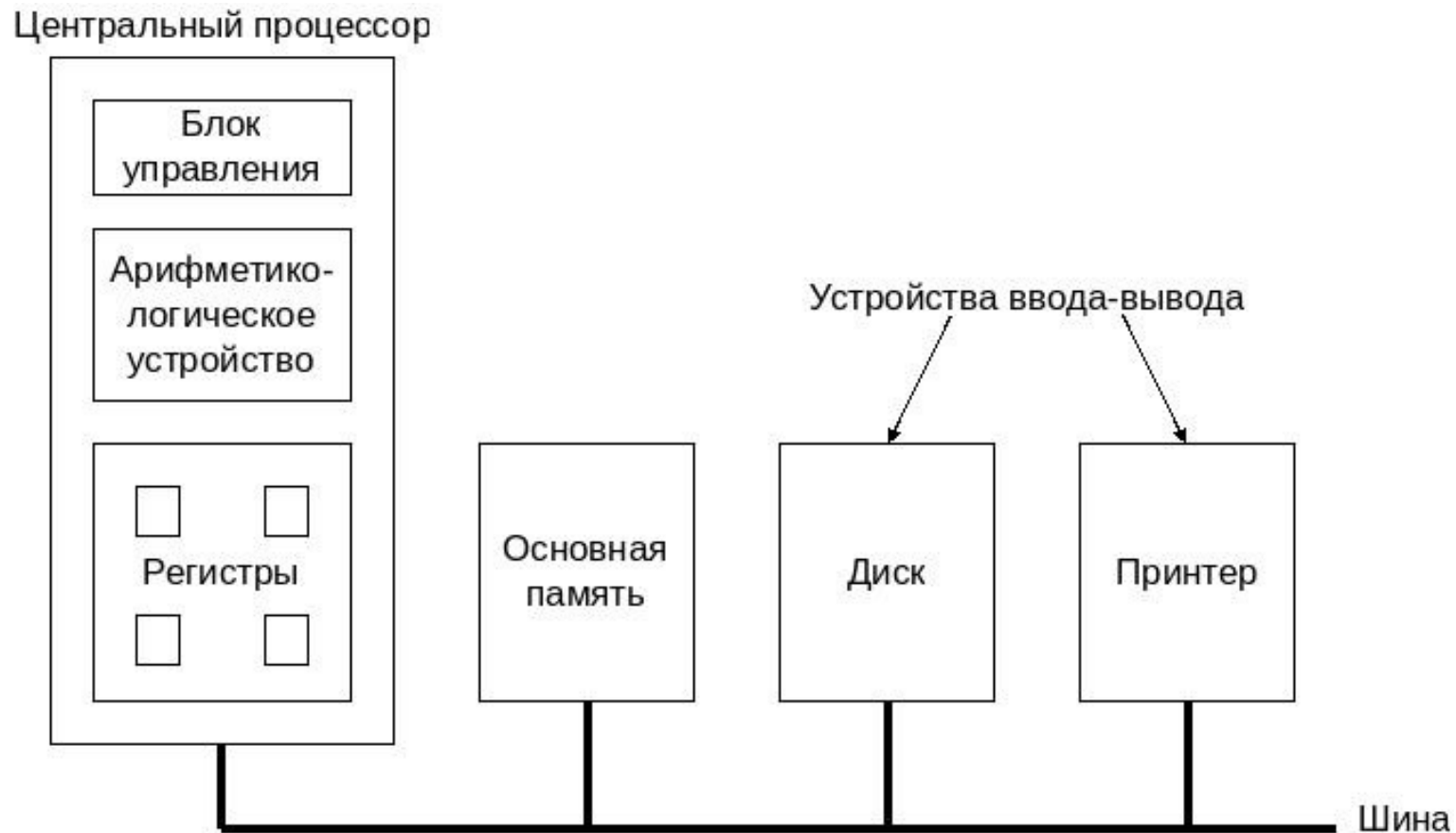


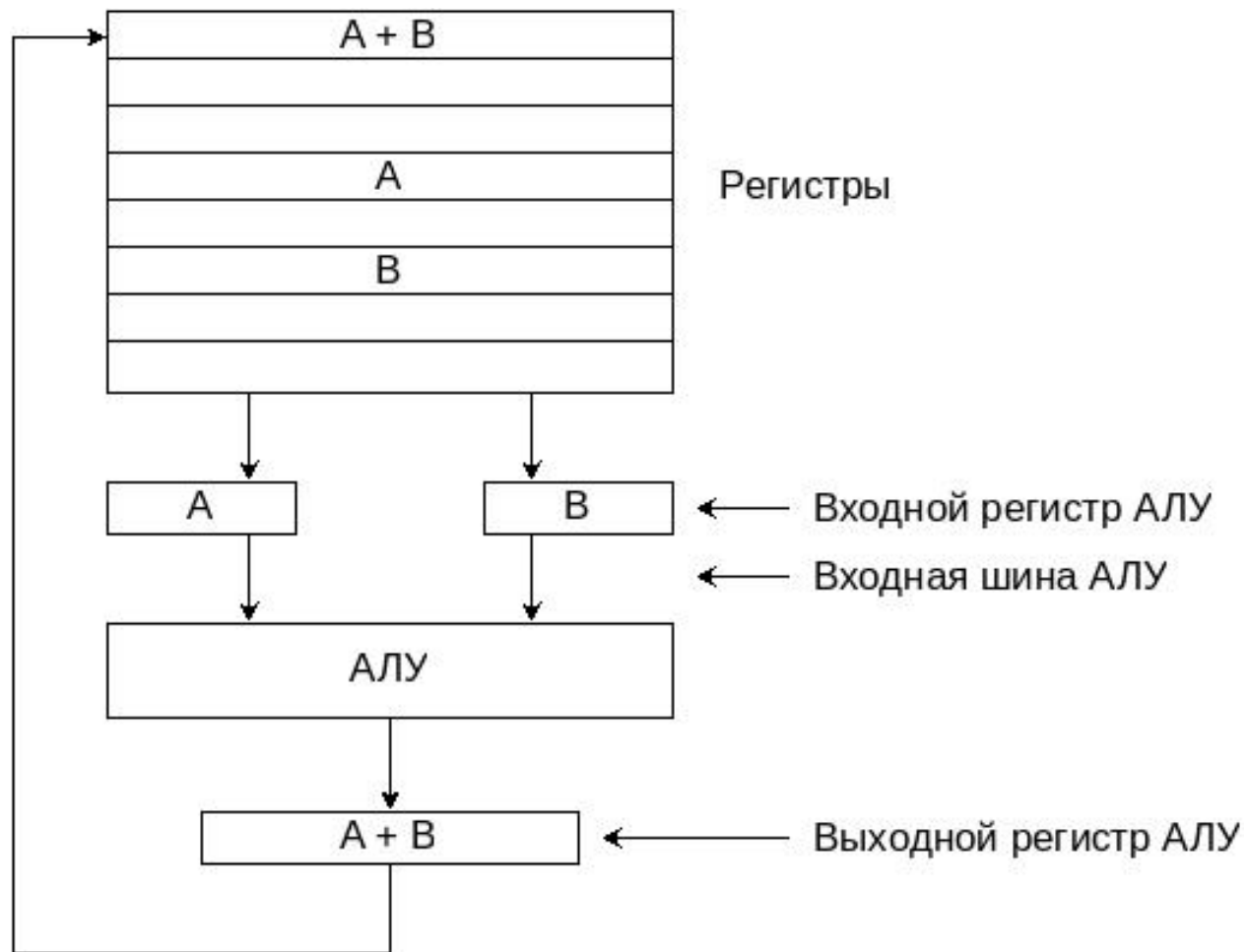
Схема компьютера с одним центральным процессором и двумя устройствами ввода-вывода.

Центральный процессор выполняет каждую команду за несколько шагов:

1. Вызывается и переносится в регистр команд следующая команда из памяти.
2. Изменение положения счетчика команд. (Указывает на следующую команду).
3. Определение типа вызванной команды.
4. Определение адреса слова при использовании слова из памяти.
5. Перенос слова в регистр центрального процессора при необходимости.
6. Выполнение команды.
7. Переход к шагу 1 для выполнения следующей команды.

Сложные команды обрабатываются с помощью **интерпретации** путем разбиения их на более мелкие (элементарные) команды.

Устройство центрального процессора



Тракт данных обычной фон-неймановской машины.

CISC (компьютер с полным набором команд) – концепция проектирования процессоров, которая характеризуется следующим набором свойств:

- ▶ нефиксированное значение длины команды;
- ▶ арифметические действия кодируются в одной команде;
- ▶ небольшое число регистров, каждый из которых выполняет строго определенную функцию.

RISC (компьютер с сокращенным набором команд) – архитектура процессора, в которой быстродействие увеличивается за счет упрощения инструкций, чтобы их декодирование было более простым, а время выполнения – меньшим.

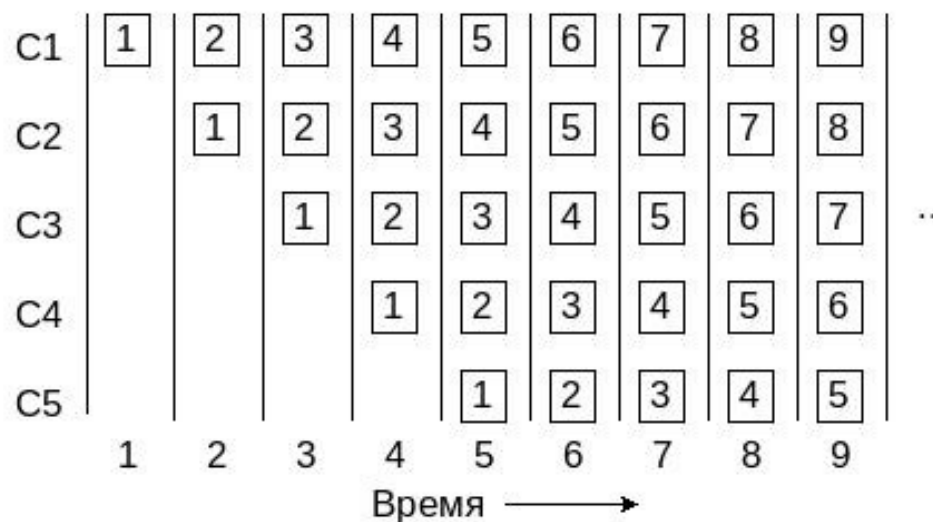
Принципы проектирования современных универсальных процессоров (принципы RISC):

1. Все команды должны выполняться непосредственно аппаратным обеспечением.
2. Компьютер должен запускать как можно больше команд в секунду.
3. Команды должны легко декодироваться.
4. К памяти должны обращаться только команды загрузки и сохранения.
5. Регистров должно быть много.

Параллелизм на уровне команд – Конвейеры



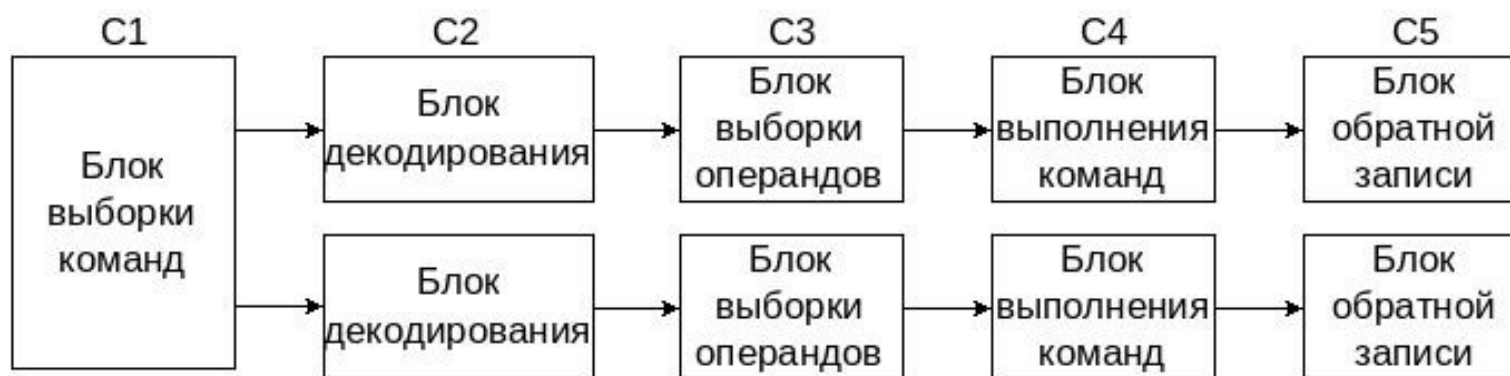
а)



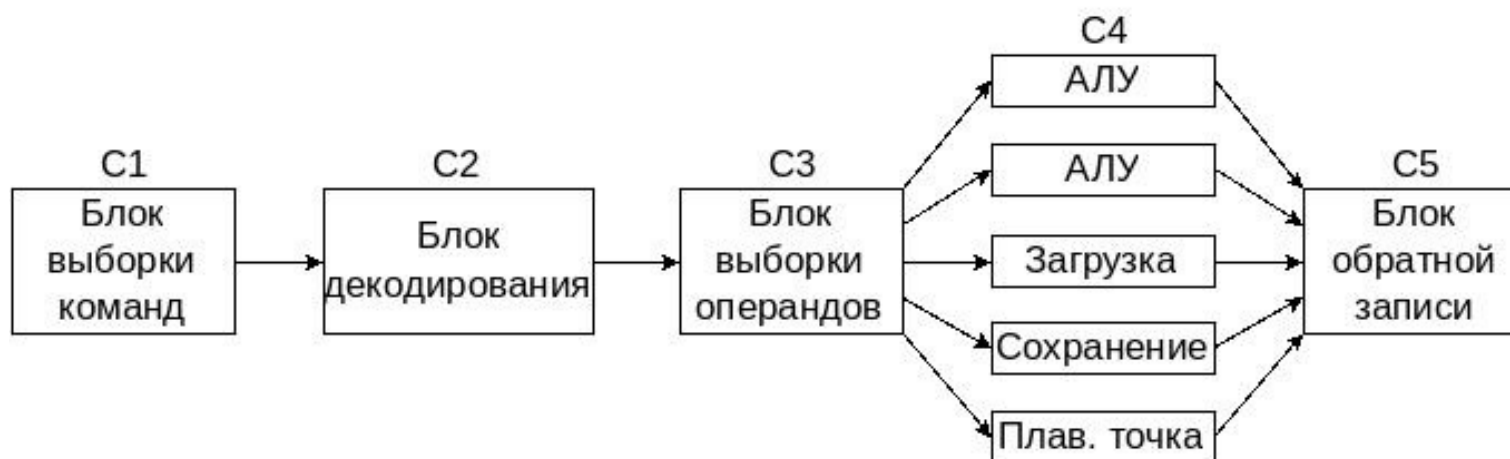
б)

Пятиступенчатый конвейер а); состояние каждой ступени в зависимости от количества пройденных циклов б) (показано 9 циклов).

Параллелизм на уровне команд – Суперскалярные архитектуры



а)



б)

Сдвоенный пятиступенчатый конвейер с общим блоком выборки команд а); суперскалярный процессор с пятью функциональными блоками б).

Векторные процессоры – архитектура при которой используются векторные регистры для сложения однородных данных.

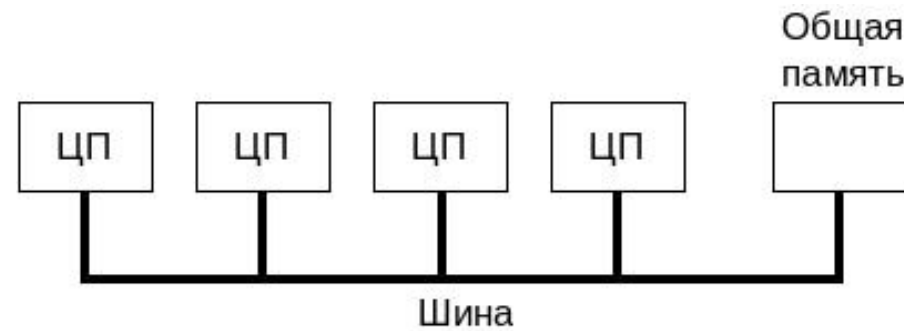
SISD (одиночный поток команд, одиночный поток данных) – архитектура компьютера, в которой один процессор выполняет один поток команд, оперируя одним потоком данных.

SIMD (одиночный поток команд, множественный поток данных) – архитектура компьютера, в которой один процессор выполняет один поток команд, оперируя множеством потоков данных.

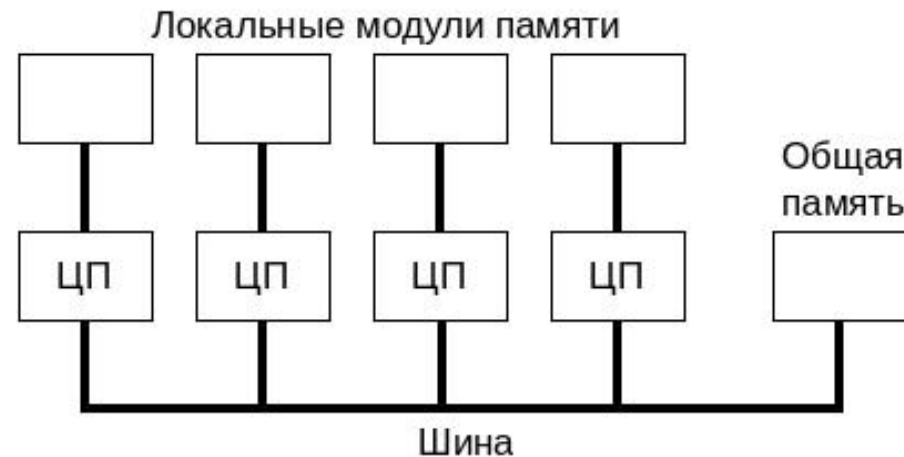
MISD (множественный поток команд, одиночный поток данных) — архитектура компьютера, в которой несколько функциональных модулей (два или более) выполняют различные операции над одними данными.

MIMD (множественный поток команд, множественный поток данных) — архитектура компьютера, в которой несколько функциональных модулей (два или более) выполняют различные операции над множеством данных.

Параллелизм на уровне процессоров – Мультипроцессоры и мультимикомпьютеры



а)

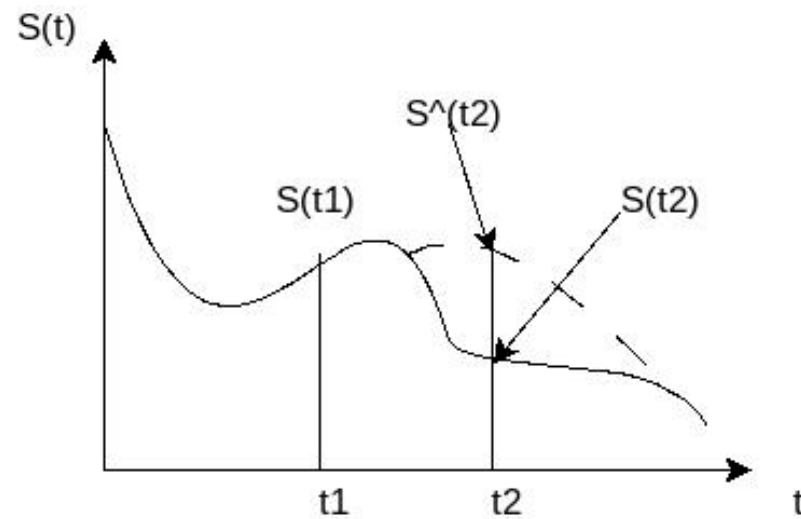


б)

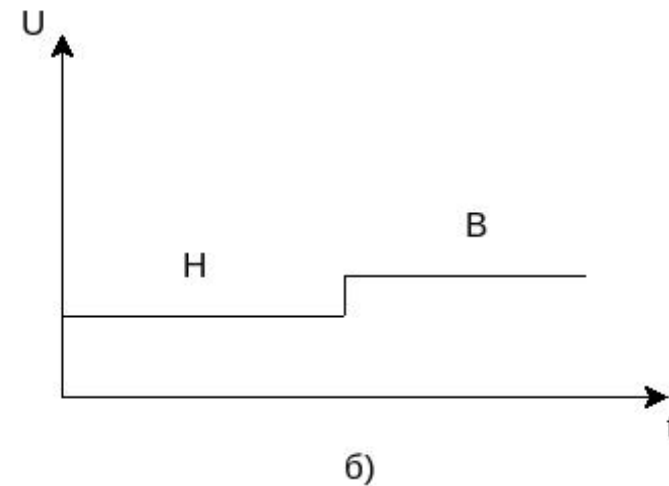
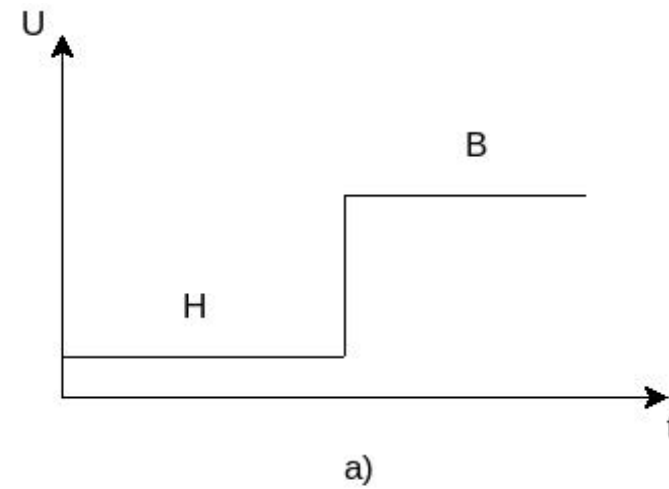
Мультипроцессор с единственной шиной и общей памятью а);
мультипроцессор с собственной локальной памятью для каждого

Иерархическая структура памяти





Пример аналогового сигнала.



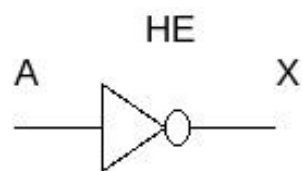
Пример цифрового сигнала.

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



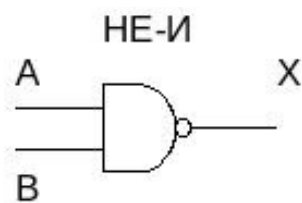
$$F = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

Представление функции.



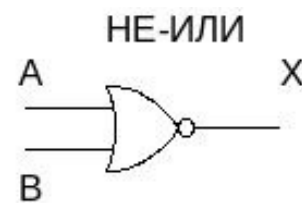
A	X
0	1
1	0

а)



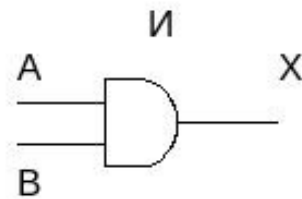
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

б)



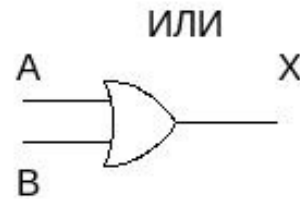
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

в)



A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

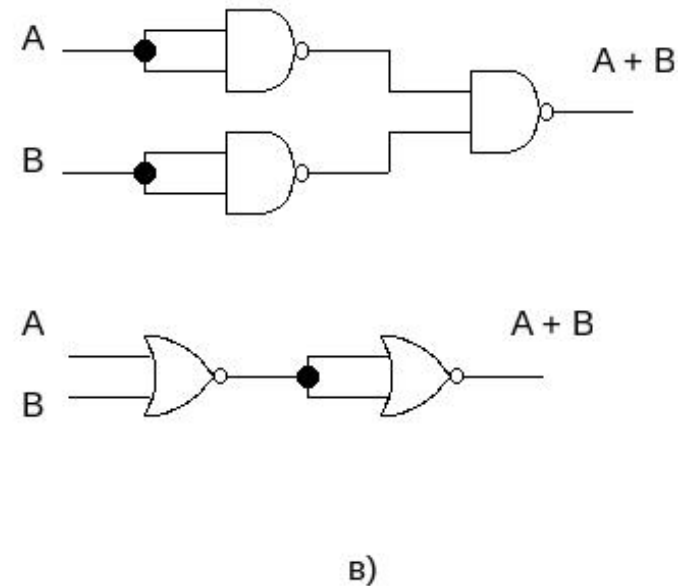
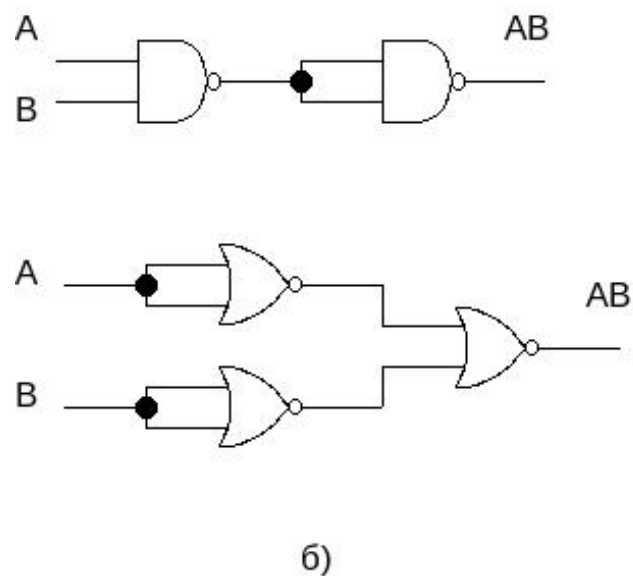
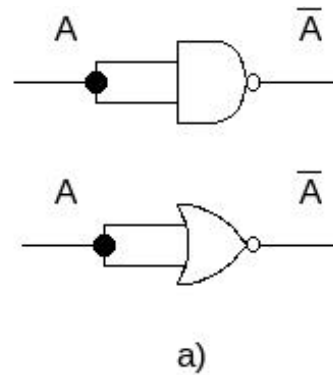
г)



A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

д)

Условные обозначения пяти основных вентилей.



Конструирование вентилях НЕ а), И б), и ИЛИ в)
только на базе вентилях НЕ-И или НЕ-ИЛИ.

1. Комбинаторные схемы:

- ▶ мультиплексоры – соединяет один из 2^n входов с одним выходом;
- ▶ демультиплексоры – соединяет один вход с одним из 2^n выходов;
- ▶ декодеры – в зависимости от входного значения устанавливает в состояние ИСТИНА один из выходов;
- ▶ компараторы – сравнивает 2 слова, возвращает 1, если они равны, иначе 0.

2. Арифметические схемы:

- ▶ схемы сдвига – сдвиг значения влево и вправо;
- ▶ сумматоры – сложение чисел;
- ▶ арифметико-логические устройства – выполнения операций И, ИЛИ, сложение на двоих словах.

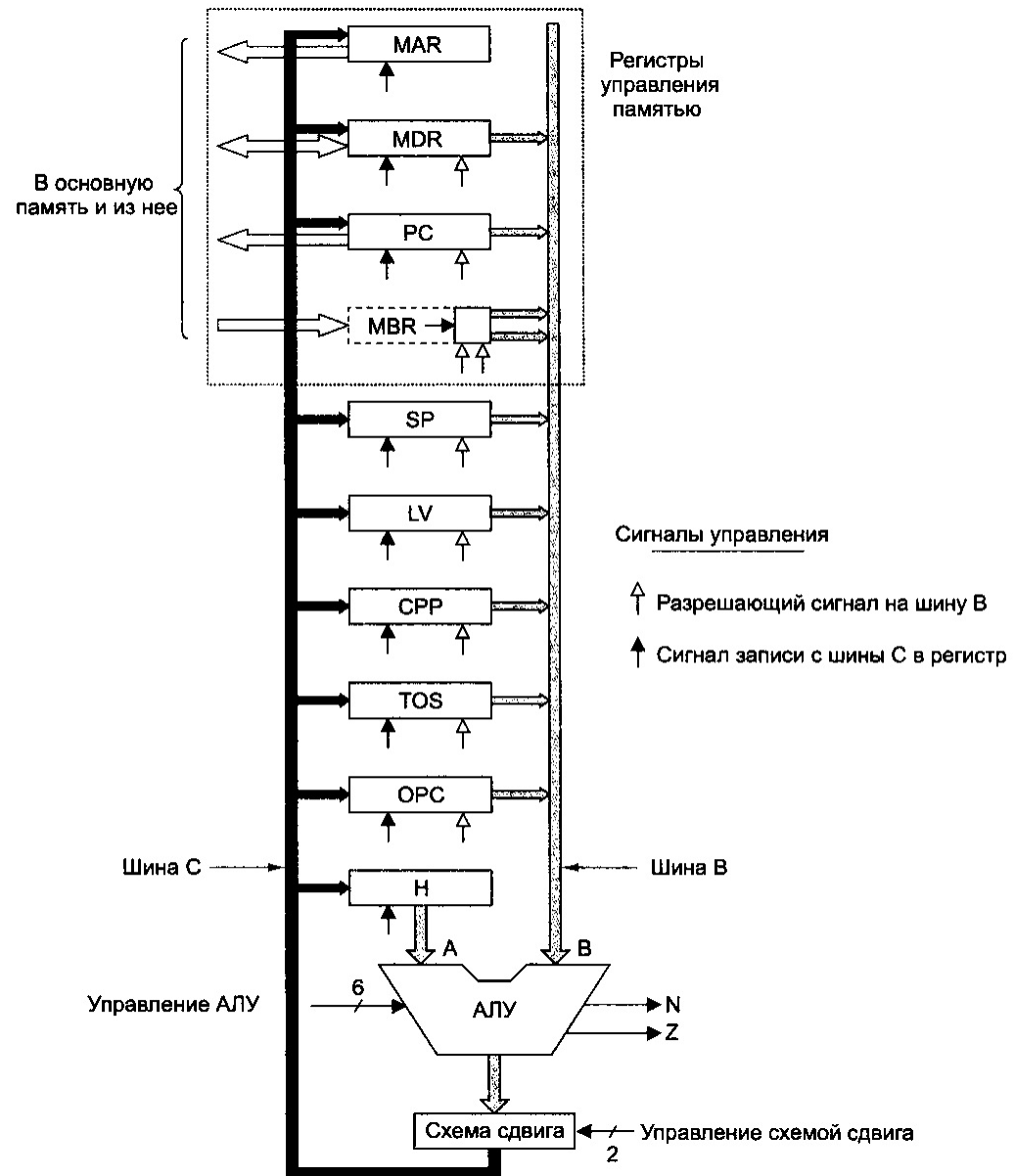
3. Тактовые генераторы – вызывает серию импульсов для синхронизации событий.

4. Схемы памяти:

- ▶ SR-защелки – запоминает предыдущее состояние;
- ▶ синхронные SR-защелки – изменяет значение только в определенный момент;
- ▶ синхронные D-защелки – изменяет значение только в определенный момент, устраняет состояние неопределенности;
- ▶ триггеры – смена состояния при переходе синхронизирующего сигнала с 0 на 1 и с 1 на 0;
- ▶ регистры – конфигурация нескольких триггеров.

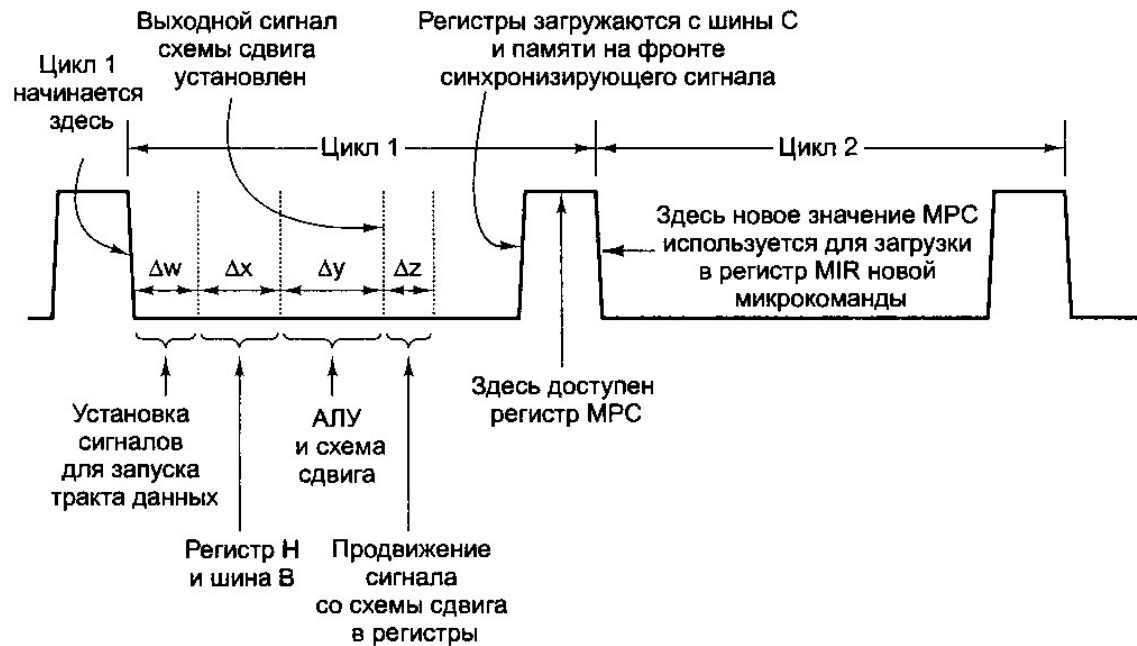
Сигналы АЛУ:

- ▶ F_0 и F_1 – задание входных сигналов A и B;
- ▶ ENA и ENB – разрешение входных сигналов A и B;
- ▶ INVA – инверсия входного сигнала A;
- ▶ INC – перенос бита в младший разряд (прибавление единицы к результату).



Пример тракта данных.

F_0	F_1	ENA	ENB	INVA	INC	Значение
0	1	1	0	0	0	A
0	1	0	1	0	0	B
0	1	1	0	1	0	\bar{A}
1	0	1	1	0	0	\bar{B}
1	1	1	1	0	0	A + B
1	1	1	1	0	1	A + B + 1
1	1	1	0	0	1	A + 1
1	1	0	1	0	1	B + 1
1	1	1	1	1	1	B - A
1	1	0	1	1	0	B - 1
1	0	1	0	1	1	-A
0	0	1	1	0	0	A И B
0	1	1	1	0	0	A ИЛИ B
0	1	0	0	0	0	0
0	1	0	0	0	1	1
0	1	0	0	1	0	-1

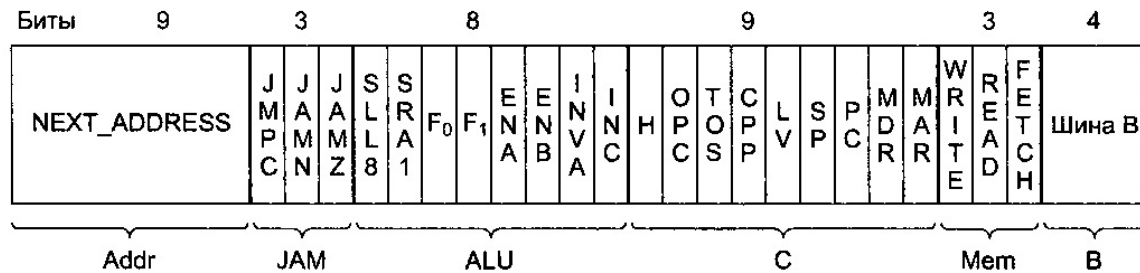


Временная диаграмма цикла тракта данных.

MPC – счетчик микропрограмм. MIR – регистр микрокоманд.

Цикл тракта данных:

1. Установка сигналов управления.
2. Загрузка значений регистров в шину В.
3. Работа АЛУ и схемы сдвига.
4. Запись результатов в регистры из шины С.



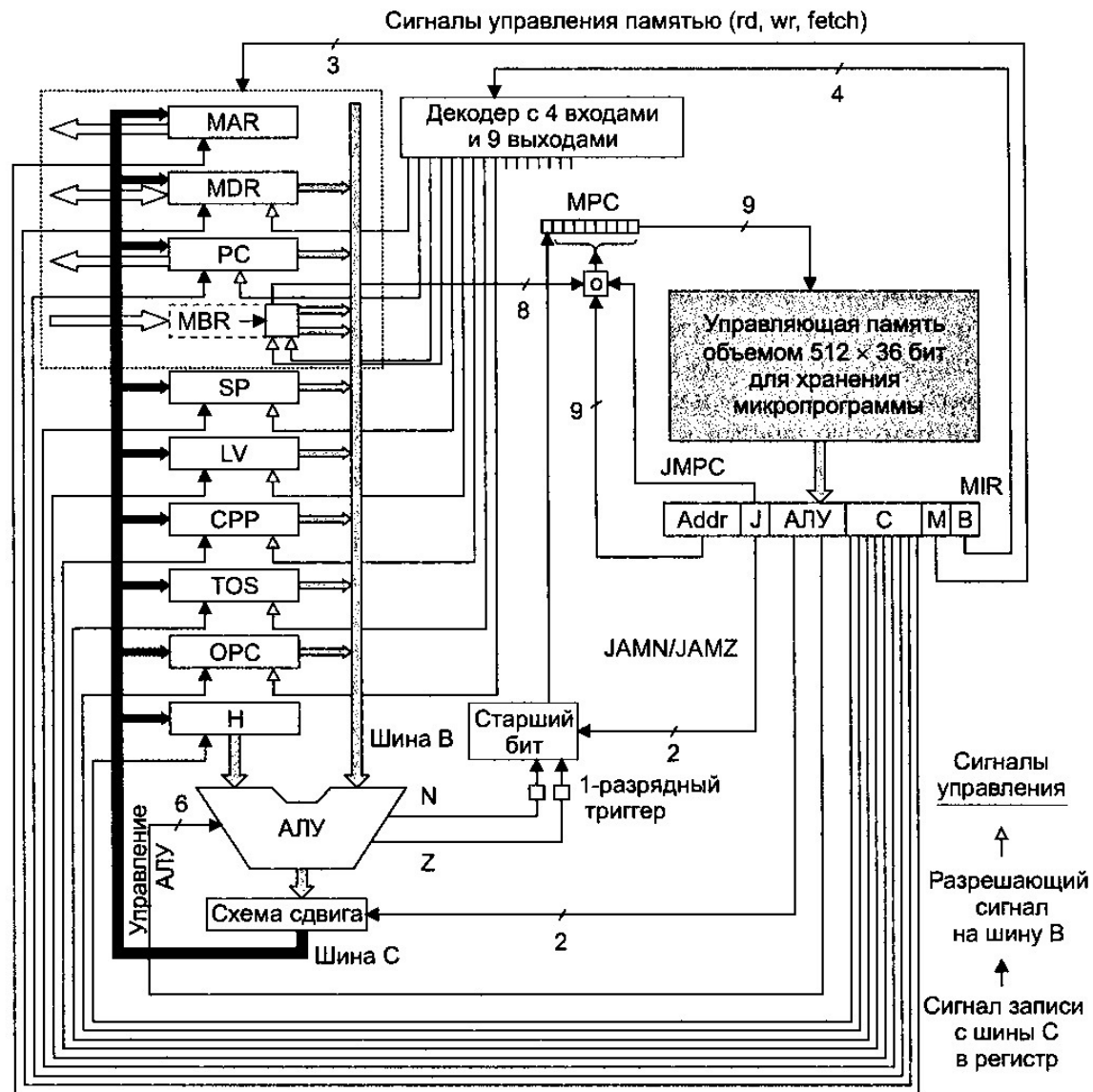
Регистры шины В

0 — MDR	5 — LV
1 — PC	6 — CPP
2 — MBR	7 — TOS
3 — MBRU	8 — OPC
4 — SP	9-15 — нет

Пример формата команды.

6 групп полей микрокоманды (36 сигналов):

1. Addr – адрес следующей микрокоманды.
2. JAM – определение метода выбора следующей микрокоманды.
3. ALU – функции АЛУ и схемы сдвига.
4. C – указание регистров для чтения из шины C.
5. Mem – функции памяти.
6. B – указание регистров для записи в шину B.

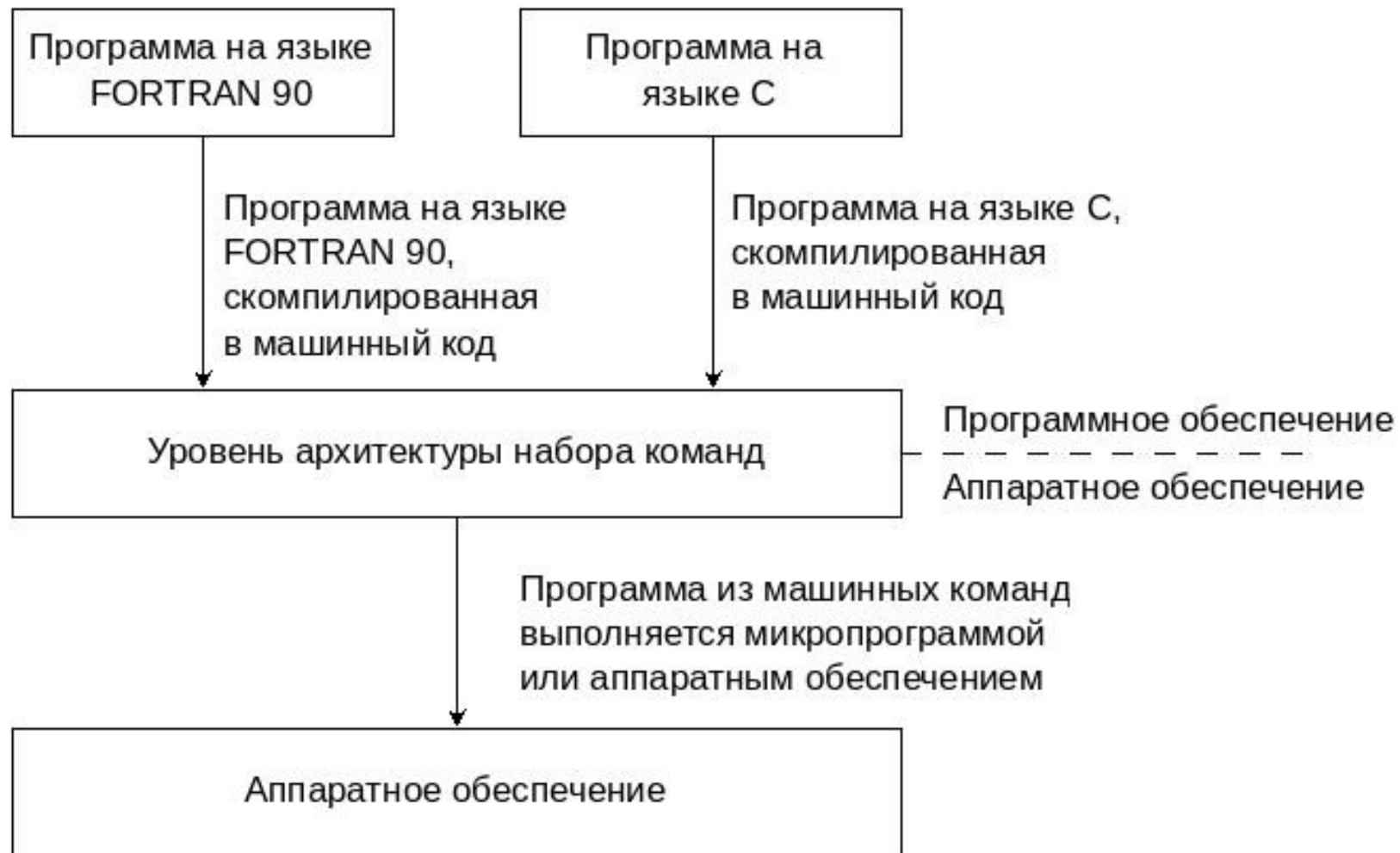


Пример микроархитектуры.

Концепции микроархитектуры для повышения производительности:

1. Выбор системы команд (RICS, CISC и т. д.).
2. Конвейеризация инструкций.
3. Кэш.
4. Прогнозирование ветвления (использование результатов последних ветвлений для предсказания будущих).
5. Суперскалярность.
6. Исполнение не по порядку (исполнение инструкций, данные для которых доступны в кэше в данный момент).
7. Подмена регистров.
8. Многопроцессорность и многопоточность и др.

Уровень архитектуры набора команд



Уровень архитектуры набора команд – промежуточное звено между компиляторами и аппаратным обеспечением.

Регистры общего назначения

	15	8	7	0
AX	AH		AL	
BX	BH		BL	
CX	CH		CL	
DX	DH		DL	

Индексные регистры

15	0
SI	
DI	

Регистры-указатели

15	0
BP	
SP	

Сегментные регистры

15	0
CS	
DS	
SS	
ES	

Регистр флагов

15	0
FLAGS	

Указатель команд

15	0
IP	

Регистры на примере процессора Intel 8086.

	Биты	8	1	5	5	5	8
1	Код операции	0	Выходной регистр	Входной регистр 2	Входной регистр 1		
2	Код операции	1	Выходной регистр	Входной регистр 1		Смещение	
3	Код операции					Смещение	

Форматы команд для трехадресной машины.

Поток управления – последовательность выполнения команд в ходе работы программы.

1. Процедуры.
2. Рекурсивные процедуры.
3. Сопрограммы.
4. Перехват исключений.
5. Прерывания.