

1. Фон-неймановские принципы

Наличие каких компонентов компьютера следует из формулы <ЭВМ - это машина с хранимой программой>?

- ✓ Оперативная память
- ✓ Процессор

Анализ структуры двоичного кода содержимого ячейки памяти фон-неймановской ЭВМ:

- ✓ не позволяет определить - команда это или данные

В фон-неймановской ЭВМ данные и команды размещаются:

- ✓ В общей памяти

Что такое адрес ячейки памяти ЭВМ?

- ✓ Ее порядковый номер

Если в выполняемой процессором команде не указан адрес следующей команды программы, то процессор выбирает эту команду -

- ✓ из следующей по порядку ячейки памяти
- ✓ из ячейки памяти, адрес которой находится в регистре-счетчике команд процессора

Процессор фон-неймановской ЭВМ находит в памяти код очередной команды для исполнения,

- ✓ выбирая адрес этой команды из регистра-счетчика команд процессора

Данные в ячейках памяти фон-неймановской ЭВМ представляются и хранятся:

- ✓ В виде двоичного кода

Для того, чтобы правильно находить команды выполняемой программы в оперативной памяти, не путая их с данными, в фон-неймановском компьютере

- ✓ в состав процессора вводится регистр-счетчик команд

В чем преимущество размещения команд и данных в общей памяти?

- ✓ Эффективнее используется объем оперативной памяти

Память фон-неймановского компьютера называется памятью с _____ выборкой

- Произвольной

2.1. Представление информации в ЭВМ

Какая система счисления выбрана в фон-неймановской ЭВМ для внутреннего представления чисел?

- ✓ Двоичная

Какое максимальное целое двоичное число без знака можно представить с помощью 16-ти разрядного двоичного кода?

- $(2^{16})-1$

Какое максимальное целое положительное число можно представить с помощью 16-ти разрядного двоично-дополнительного кода?

- $(2^{15})-1$

Восьмеричное представление шестнадцатиразрядного двоичного числа имеет вид 012111. Образуйте восьмеричное представление эквивалентного ему числа с противоположным знаком.

165667

Перенос в старший разряд при отсутствии переноса из старшего разряда свидетельствует:

- ✓ Об ошибке выполнения операции сложения целых положительных чисел
- ✓ Об ошибке выполнения операции сложения целых чисел со знаком
- ✓ Об ошибке выполнения операции сложения целых отрицательных чисел

Арифметическое переполнение это:

- ✓ Наличие переноса в старший разряд при отсутствии переноса из старшего разряда
- ✓ Наличие переноса из старшего разряда при отсутствии переноса в старший разряд

При выполнении операции сложения двоичных чисел $00100110 + 01001100$ имеет место:

- ✓ Отсутствует арифметическое переполнение и переносы в старший разряд и из старшего разряда

Преобразуйте число 1001001001001001 из двоичной системы счисления в восьмеричную 111111

Результат сложения двух шестнадцатиразрядных двоичных чисел без знака

$0001010001001001 + 1001001001111001$:

- Правильный

При сложении двух двоичных чисел $0001010001001001 + 1001001001111001$ арифметическое переполнение:

- Нет

Результат сложения двух двоичных чисел со знаком $0001010001001001 + 1001001001111001$:

- правильный

При сложении двух двоичных чисел $0001010001001001 + 1001001001111001$ перенос в старший разряд:

- нет

При сложении двух двоичных чисел $0001010001001001 + 1001001001111001$ перенос из старшего разряда:

- нет

$0001010001001001 + 1001001001111001 =$

1010011011000010

2.2 Двоичная арифметика

Восьмеричное представление шестнадцатиразрядного двоичного числа имеет вид 012111. Образуйте восьмеричное представление эквивалентного ему числа с противоположным знаком.

165667

Восьмеричное представление шестнадцатиразрядного двоичного числа имеет вид 111111. Образуйте восьмеричное представление эквивалентного ему числа с противоположным знаком.

066667

Восьмеричное представление шестнадцатиразрядного двоичного числа имеет вид 111131. Образуйте восьмеричное представление эквивалентного ему числа с противоположным знаком.

066647

Арифметическое переполнение это:

- ✓ Наличие переноса в старший разряд при отсутствии переноса из старшего разряда
- ✓ Наличие переноса из старшего разряда при отсутствии переноса в старший разряд

Младший разряд двоичного числа это:

- ✓ Самый правый разряд

Перенос из старшего разряда при отсутствии переноса в старший разряд свидетельствует:

- ✓ Об ошибке выполнения операции сложения целых отрицательных чисел
- ✓ Об ошибке выполнения операции сложения целых положительных чисел
- ✓ Об ошибке выполнения операции сложения целых чисел со знаком
- ✓ Об ошибке выполнения операции сложения целых чисел без знака

При выполнении операции сложения двоичных чисел $00100110 + 01001100$ имеет место:

- ✓ Отсутствует арифметическое переполнение и переносы в старший разряд и из старшего разряда

При выполнении операции сложения двоичных чисел $10100110 + 11001100$ имеет место:

- ✓ Только перенос из старшего разряда
- ✓ Арифметическое переполнение

Преобразуйте число 1001001001001001 из двоичной системы счисления в шестнадцатеричную

9249

Преобразуйте число 1111111111111111 из двоичной системы счисления в шестнадцатеричную

FFFF

Преобразуйте число 1001001001001001 из двоичной системы счисления в восьмеричную

111111

Преобразуйте число 1010101010101010 из двоичной системы счисления в восьмеричную

125252

Преобразуйте число 1100110011001100 из двоичной системы счисления в восьмеричную

146314

Преобразуйте число 1111111111111111 из двоичной системы счисления в восьмеричную

177777

Преобразуйте число 9B8C из шестнадцатеричной системы счисления в двоичную

1001101110001100

Преобразуйте число FFFF из шестнадцатеричной системы счисления в двоичную

1111111111111111

Преобразуйте двоичное число 0001010001001001 в его двоично-дополнительный код

1110101110110111

Преобразуйте двоичное число **1000001010100100** в его двоично-дополнительный код
0111110101011100

Преобразуйте двоичное число **1001001001100001** в его двоично-дополнительный код
0110110110011111

Преобразуйте двоичное число **1001001111001001** в его двоично-дополнительный код
0110110000110111

Преобразуйте число **012111** из восьмеричной системы в шестнадцатиразрядное двоичное число
0001010001001001

Преобразуйте число **101234** из восьмеричной системы в шестнадцатиразрядное двоичное число
1000001010011100

Преобразуйте число **111311** из восьмеричной системы в шестнадцатиразрядное двоичное число
1001001011001001

Преобразуйте число **161111** из восьмеричной системы в шестнадцатиразрядное двоичное число
1110001001001001

Результат сложения двух шестнадцатиразрядных двоичных чисел без знака
0001010001001001 + 1001001001111001 :

- Правильный

Результат сложения двух шестнадцатиразрядных двоичных чисел без знака
1001001001001010 + 1000001010100100 :

- Неправильный

Результат сложения двух шестнадцатиразрядных двоичных чисел без знака
1001001100001001 + 0001011001001001 :

- Правильный

Результат сложения двух шестнадцатиразрядных двоичных чисел без знака
1110011100101110 + 1101001010001001 :

- неправильный

Результат сложения двух двоичных чисел со знаком **0001010001001001 + 1001001001111001 :**

- правильный

Результат сложения двух двоичных чисел со знаком **1001001001001100 + 1011100000110101 :**

- неправильный

Результат сложения двух двоичных чисел со знаком **1110011100101110 + 1101001010001001 :**

- правильный

При сложении двух двоичных чисел **0001010001001001 + 1001001001111001** арифметическое переполнение:

- нет

При сложении двух двоичных чисел $1001001001001100 + 1011100000110101$ арифметическое переполнение:

- есть

При сложении двух двоичных чисел $1110011100101110 + 1101001010001001$ арифметическое переполнение:

- нет

При сложении двух двоичных чисел $0001010001001001 + 1001001001111001$ перенос в старший разряд:

- нет

При сложении двух двоичных чисел $1001001001010001 + 1001001001001001$ перенос в старший разряд:

- нет

При сложении двух двоичных чисел $1110011100101110 + 1101001010001001$ перенос в старший разряд:

- есть

При сложении двух двоичных чисел $0001010001001001 + 1001001001111001$ перенос из старшего разряда:

- нет

При сложении двух двоичных чисел $1001001001010001 + 1001001001001001$ перенос из старшего разряда:

- есть

При сложении двух двоичных чисел $1110011100101110 + 1101001010001001$ перенос из старшего разряда:

- есть

$$0001010001001001 + 1001001001111001 = 1010011011000010$$

$$0111001001001001 + 1001001001110001 = 0000010010111010$$

$$1001001001001111 + 0110101100011010 = 1111110101101001$$

$$1110011100101110 + 1101001010001001 = 1011100110110111$$

3. Принципы построения цифровых устройств

Таблица истинности логической операции "И"

X	Y	X И Y
0	0	0
0	1	0
1	0	0
1	1	1

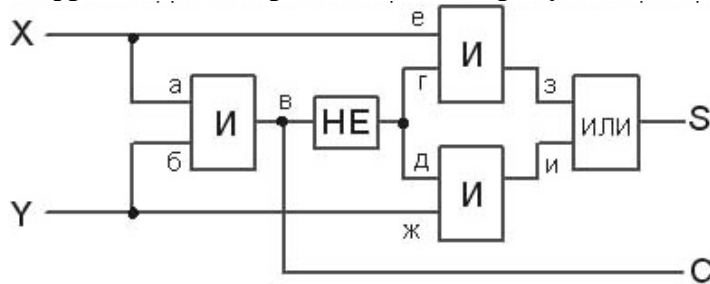
Таблица истинности логической операции "ИЛИ"

X	Y	X ИЛИ Y
0	0	0
0	1	1
1	0	1
1	1	1

Таблица истинности логической операции "НЕ"

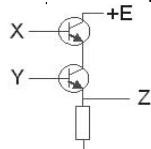
X	не X
0	1
1	0

Какая цифровая схема представлена на рисунке?



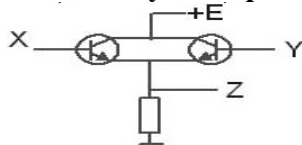
полусумматор

Какую логическую операцию реализует данная электронная схема?



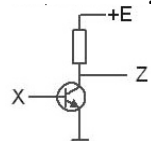
И

Какую логическую операцию реализует данная электронная схема?



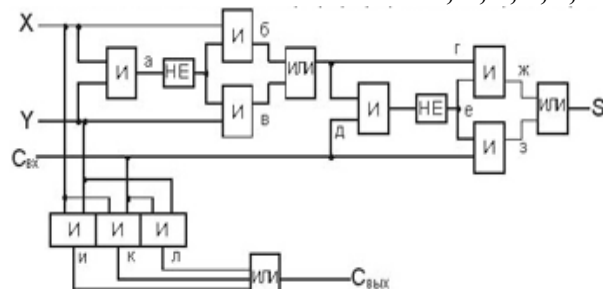
ИЛИ

Какую логическую операцию реализует данная электронная схема?



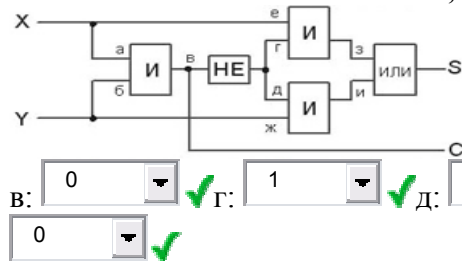
НЕ

Указать состояние сигналов б, в, г, е, з, S и C_{вых} при X=0, Y=1 и C_{вх}=0



б: ☒ в: ☒ г: ☒ е: ☒ з: ☒ S: ☒ C_{вых}: ☒

Указать состояние сигналов в, г, д, з, и, S и C на выходе при X=0 и Y=0



В: Г: Д: З: И: S: C:

Состояние сигналов S и Cвых на выходе двоичного полусумматора при X=1 и Y=1

S: Cвых:

Таблица истинности двоичного полусумматора

X	Y	S	Cвых
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

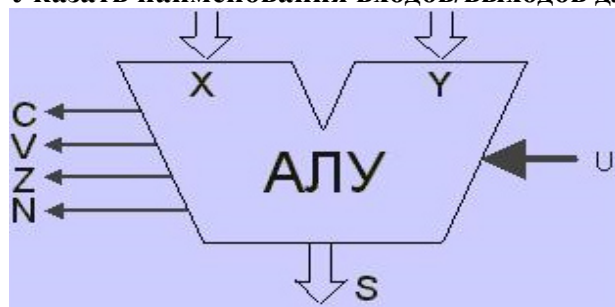
Состояние сигналов S и Cвых на выходе полного двоичного сумматора при X=0, Y=1 и Cвх=1

S: Cвых:

Таблица истинности полного одноразрядного двоичного сумматора

X	Y	Cвх	S	Cвых
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Указать наименования входов/выходов для Арифметико-логического устройства



Нулевой результат ☐

Арифметическое переполнение ☐

Код операции ☐

Перенос из старшего разряда ☐

Входной операнд 1 ☐

Входной операнд 2 ☐

Отрицательный результат

Пусть на входе логической схемы значение сигнала $X=0$.

В каком состоянии будет находиться сигнал Z на выходе логических схем:

схема "И" - $Z=$

схема "ИЛИ" - $Z=$

схема "НЕ" - $Z=$

Какой логической операции соответствует приведенная таблица истинности?

Входы			Выход
0	0	1	1
0	1	1	1
1	1	1	1
0	0	0	0
1	0	1	1
0	1	0	1
1	1	0	1
1	0	0	1

<ИЛИ>

Какой логической операции соответствует приведенная таблица истинности?

Входы			Выход
1	0	0	0
1	1	1	1
1	0	1	0
0	1	0	0
0	1	1	0
1	1	0	0
0	0	0	0
0	0	1	0

<И>

Пусть на входе логической схемы значение сигнала $X=1$.

В каком состоянии будет находиться сигнал Z на выходе логических схем:

схема "И" - $Z=$

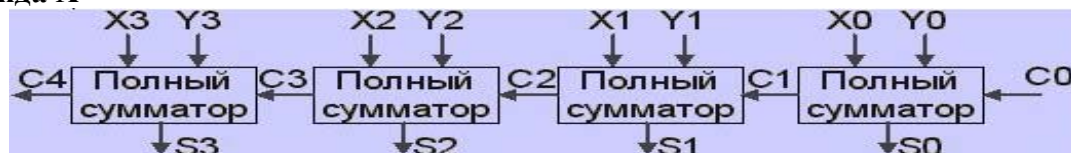
схема "ИЛИ" - $Z=$

схема "НЕ" - $Z=$

Состояние сигналов S и $S_{\text{вых}}$ на выходе двоичного полусумматора при $X=1$ и $Y=0$

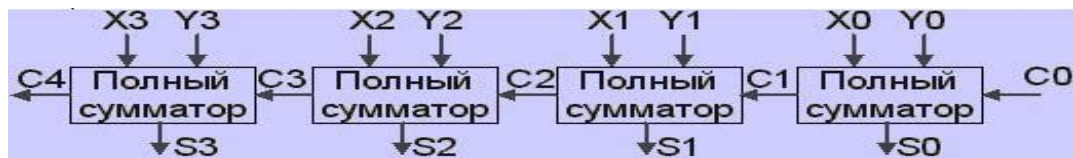
$S:$ $S_{\text{вых}}:$

Для схемы четырехразрядного двоичного сумматора указать младший разряд входного операнда X



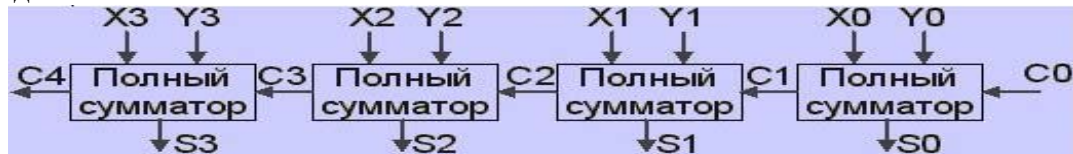
X_0

Для схемы четырехразрядного двоичного сумматора указать младший разряд входного операнда Y



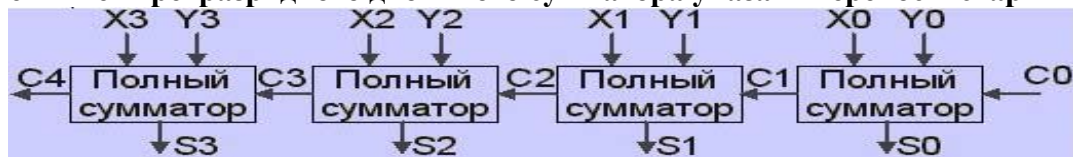
Y0

Для схемы четырехразрядного двоичного сумматора указать младший разряд выходного операнда S



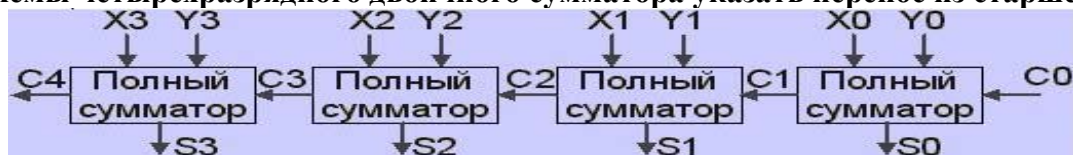
S0

Для схемы четырехразрядного двоичного сумматора указать перенос в старший разряд



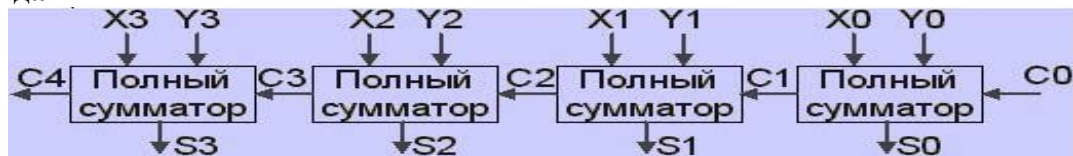
C3

Для схемы четырехразрядного двоичного сумматора указать перенос из старшего разряда



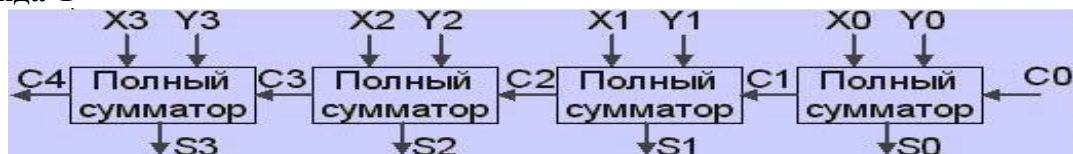
C4

Для схемы четырехразрядного двоичного сумматора указать старший разряд входного операнда X



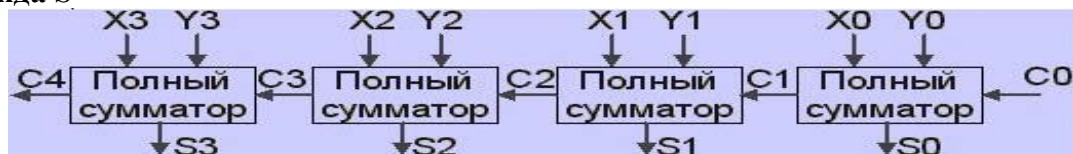
X3

Для схемы четырехразрядного двоичного сумматора указать старший разряд входного операнда Y



Y3

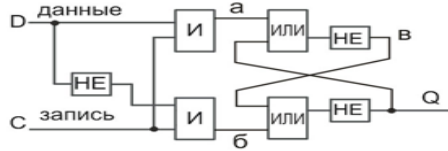
Для схемы четырехразрядного двоичного сумматора указать старший разряд выходного операнда S



S3

4. Элементы памяти

Как называется устройство, приведенное на схеме?



Триггер

Как время обращения к ячейке памяти зависит от ее адреса?

время обращения к ячейке памяти не зависит от ее адреса

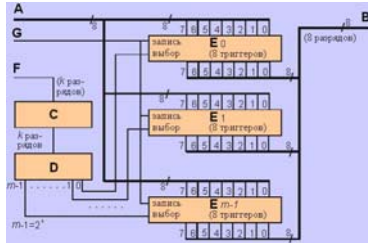
К какому объему памяти можно адресоваться с помощью 20-разрядного адреса?

- ✓ 1 Мегабайт
- ✓ 1 048 576 Байт

Сколько разрядов в физическом адресе требуется для адресации памяти объемом 1 Гигабайт?

30

Укажите название блока А запоминающего устройства



- Входная шина данных

Укажите название блока В запоминающего устройства

- Выходная шина данных

Укажите название блока С запоминающего устройства

- Регистр адреса

Укажите название блока D запоминающего устройства

- Дешифратор адреса

Укажите название блока E запоминающего устройства

- Ячейка памяти

Укажите название блока F запоминающего устройства

- Шина адреса

Укажите название блока G запоминающего устройства

- Шина записи/чтения

Сколько ячеек памяти содержит запоминающее устройство, если шина F является восьмиразрядной (K=8)?

256

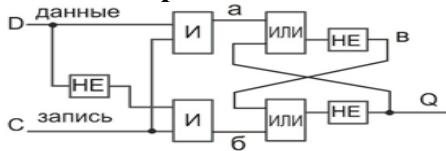
Во сколько ячеек памяти можно одновременно записывать информацию?

- в одну

К какой ячейке памяти меньше время обращения - к первой или последней?

- время обращения одинаковое

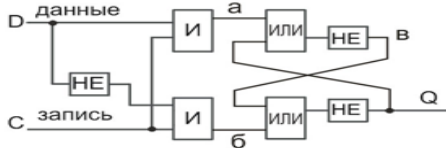
В момент времени t_1 схема находится в следующем состоянии: $D=0, C=0, Q=0$.



В каком состоянии будет выход $Q=?$ в момент t_2 , после того как вход D перейдет в состояние $\langle 1 \rangle$?

0

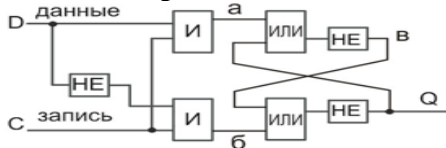
В момент времени t_1 схема находится в следующем состоянии: $D=0, C=0, Q=0$



В каком состоянии будет выход $Q=?$ в момент t_2 , после того как вход C перейдет в состояние $\langle 1 \rangle$?

0

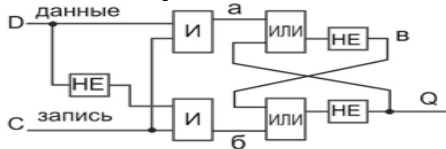
В момент времени t_1 схема находится в следующем состоянии: $D=0, C=0, Q=1$



В каком состоянии будет выход $V=?$ в момент t_2 , после того как вход D перейдет в состояние $\langle 1 \rangle$?

0

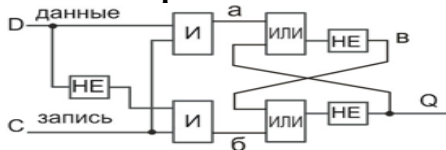
В момент времени t_1 схема находится в следующем состоянии: $D=0, C=0, Q=1$.



В каком состоянии будет выход $Q=?$ в момент t_2 , после того как вход C перейдет в состояние $\langle 1 \rangle$?

0

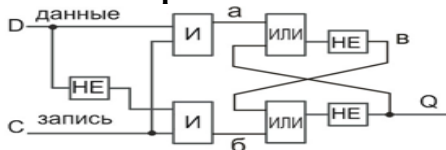
В момент времени t_1 схема находится в следующем состоянии: $D=0, C=1, Q=0$



В каком состоянии будет выход $V=?$ в момент t_2 , после того как вход C перейдет в состояние $\langle 0 \rangle$?

1

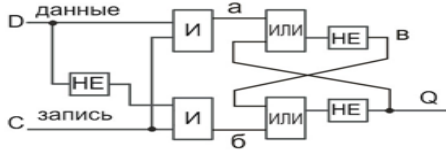
В момент времени t_1 схема находится в следующем состоянии: $D=1, C=0, Q=0$



В каком состоянии будет выход $б=?$ в момент t_2 , после того как вход C перейдет в состояние $\langle 1 \rangle$?

0

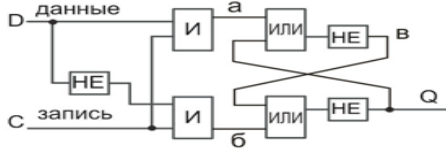
В момент времени t_1 схема находится в следующем состоянии: $D=1, C=0, Q=1$.



В каком состоянии будет выход б=? в момент t_2 , после того как вход D перейдет в состояние $<0>$?

0

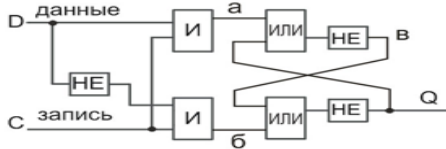
В момент времени t_1 схема находится в следующем состоянии: $D=1, C=0, Q=1$.



В каком состоянии будет выход б=? в момент t_2 , после того как вход C перейдет в состояние $<1>$?

0

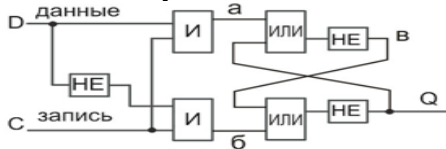
В момент времени t_1 схема находится в следующем состоянии: $D=1, C=1, Q=1$.



В каком состоянии будет выход в=? в момент t_2 , после того как вход C перейдет в состояние $<0>$?

0

В момент времени t_1 схема находится в следующем состоянии: $D=1, C=0, Q=0$.



В каком состоянии будет выход а=? в момент t_2 , после того как вход D перейдет в состояние $<0>$?

0

5. Базовая структура ЭВМ

Укажите, какие из перечисленных блоков (элементов) входят в состав Центрального процессора фон-неймановской ЭВМ

- ✓ Регистры общего назначения
- ✓ Регистр-счетчик команд
- ✓ Регистр команд
- ✓ Арифметико-логическое устройство
- ✓ Регистр состояния процессора
- ✓ Регистр-указатель стека
- ✓ Устройство управления

Укажите, правильное определение (функции) блока (элемента) процессора фон-неймановской ЭВМ - <Регистр команд>

- ✓ предназначен для временного хранения кода текущей команды выполняемой ЭВМ программы на время ее выполнения

Укажите, правильное определение (функции) блока (элемента) процессора фон-неймановской ЭВМ - <Регистр-счетчик команд>

- ✓ предназначен для формирования адреса ячейки памяти, из которой должна быть выбрана следующая команда выполняемой программы

Укажите, правильное определение понятия или блока (элемента) фон-неймановской ЭВМ - <Система команд процессора>

- ✓ совокупность команд, которые может выполнять процессор

В каком регистре (регистрах) процессора можно посмотреть, произошел или нет при выполнении текущей команды перенос из старшего разряда?

- ✓ Регистр состояния процессора

Содержимое, какого (каких) регистра (регистров) всегда интерпретируется процессором как адрес ячейки памяти, в которой размещена следующая команда выполняемой программы?

- ✓ Регистра-счетчика команд

Укажите, какой (какие) из перечисленных блоков (элементов) фон-неймановской ЭВМ выполняет приведенную функцию - <Используется процессором как временная быстродействующая память для входных и выходных данных (операндов) АЛУ, а также для реализации методов адресации операндов>

- ✓ Регистр общего назначения

Укажите, какому (каким) из перечисленных понятий фон-неймановской ЭВМ соответствует приведенное определение - <Совокупность команд, которые способен выполнять процессор>

- ✓ Система команд процессора

Если в текущей команде не указано, из какой ячейки памяти выбирать следующую команду, то в фон-неймановской ЭВМ:

- ✓ процессор выбирает эту команду из ячейки памяти, следующей за ячейкой памяти текущей команды

Процессор фон-неймановской ЭВМ получает адрес следующей команды

- ✓ из регистра - счетчика команд

6. Цикл выполнения команд ЭВМ

Цикл выполнения команды это –

- ✓ Последовательность микроопераций центрального процессора, реализующих конкретную команду

Путем перетаскивания пунктов указать правильную последовательность операций при выполнении команды "Сложение"

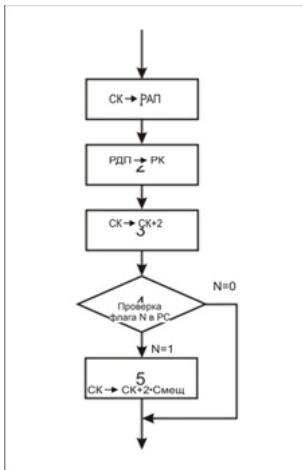
Используемые сокращения:

СК - регистр счетчик команд процессора,
РК - регистр команд процессора,
РС - регистр состояния процессора,
УУП - устройство управления процессора,
АЛУ - арифметико-логическое устройство,
ОЗУ - оперативное запоминающее устройство,
РАП - регистр адреса памяти ОЗУ,
РДП - регистр данных памяти ОЗУ

- 1) Содержимое СК пересылается в РАП ОЗУ

- 2) Из ячейки памяти, адрес которой указан в РАП, извлекается код команды и через РДП и шину данных передается в РК
- 3) Содержимое СК автоматически увеличивается
- 4) УУП, интерпретируя код команды, находящейся в РК, определяет, что это операция сложения, требующая наличия двух операндов
- 5) УУП, интерпретируя код команды, определяет адрес первого операнда и помещает его в РАП
- 6) Первый операнд считывается из указанной в РАП ячейки, и через РДП по шине данных передается в регистр процессора, подключенный к первому входу АЛУ
- 7) УУП, интерпретируя код команды, определяет адрес второго операнда и помещает его в РАП
- 8) Второй операнд считывается из указанной в РАП ячейки, и через РДП по шине данных передается в регистр процессора, подключенный ко второму входу АЛУ
- 9) УУП разрешает АЛУ выполнение операции сложения находящихся на его входах операндов
- 10) Состояние результата выполненной операции отражается в РС
- 11) Результат операции пересылается в ОЗУ по адресу, сформированному устройством управления
- 12) Адрес следующей команды пересылается из СК в РАП

Цикл выполнения команды условного перехода - "Ветвление, если меньше нуля"



7.1. Система команд и адресация операндов

Адресная часть команды это

- ✓ А. часть двоичного кода команды, представляющая информацию о местонахождении (адресах) операндов, используемых в команде
- ✓ D. часть двоичного кода команды, представляющая информацию об адресе следующей команды.

Адресный код в команде это

- ✓ А. информация об адресе операнда, содержащаяся в команде

Исполнительный адрес команды это

- ✓ В. адрес ячейки памяти, к которой производится фактическое обращение при выполнении команды

Машинная команда это

- ✓ Двоичный код, определяющий операцию, которую должен выполнять процессор, и, если необходимо, участвующие в этой операции операнды

Операционная часть команды это

- ✓ В. часть двоичного кода команды, обозначающая операцию, которую должна инициировать команда

Поле адресации операнда в команде включает в себя код метода адресации

- ✓ С. код метода адресации
- ✓ D. номер регистра общего назначения

Формат команды это

- ✓ В. определенный в данной системе команд способ кодирования в командном коде информации о задаваемой командой операции и используемых ею операндах

Абсолютный метод адресации

- ✓ В. Во втором слове команды, т.е. в следующей за кодом команды ячейке памяти, указывается адрес операнда

Абсолютный метод адресации это

- ✓ А. Косвенно-автоинкрементный метод адресации через регистр счетчик команд

Автодекрементный метод адресации

- ✓ F. При интерпретации команды содержимое указанного в команде регистра вначале уменьшается на 1 или 2, после чего уменьшенное содержимое регистра интерпретируется процессором как адрес ячейки памяти, в которой находится операнд

Автоинкрементный метод адресации

- ✓ А. Содержимое указанного в команде регистра интерпретируется процессором как адрес ячейки памяти, в которой находится операнд, и после выборки операнда содержимое регистра увеличивается на 1 или 2 таким образом, чтобы указывать на адрес следующей по порядку ячейки памяти

Автоинкрементный метод адресации

- ✓ Содержимое указанного в команде регистра интерпретируется процессором как адрес ячейки памяти, в которой находится операнд, и после выборки операнда содержимое регистра увеличивается на 1 или 2 таким образом, чтобы указывать на адрес следующей по порядку ячейки памяти

Автодекрементный способ адресации

- ✓ При интерпретации команды содержимое указанного в команде регистра вначале уменьшается на 1 или 2, после чего уменьшенное содержимое регистра интерпретируется процессором как адрес ячейки памяти, в которой находится операнд

Косвенная адресация со смещением (индексный метод адресации)

- ✓ С. В дополнительном слове команды указывается так называемое смещение (индексное слово). Исполнительный адрес операнда определяется как сумма содержимого указанного в команде регистра и смещения (индексного слова)

Косвенно-автодекрементный метод адресации

- ✓ При интерпретации команды содержимое указанного в команде регистра вначале уменьшается на 2, после чего уменьшенное содержимое регистра интерпретируется процессором как адрес ячейки памяти, в которой находится адрес операнда

Косвенно-автоинкрементный метод адресации

- ✓ В. Содержимое указанного в команде регистра интерпретируется процессором как адрес ячейки памяти, в которой находится адрес операнда, и после выборки операнда

содержимое регистра (адрес адреса) увеличивается таким образом, чтобы указывать на адрес следующей по порядку ячейки

Косвенно-относительный метод адресации это

- ✓ Г. Косвенно-индексный метод адресации через регистр счетчик команд

Косвенно-регистровый метод адресации

- ✓ Е. Содержимое указанного в команде регистра интерпретируется процессором как адрес операнда

Непосредственный метод адресации

- ✓ А. Операнд указывается непосредственно в команде, после кода операции. Процессор получает в этом случае адрес операнда непосредственно из своего регистра-счетчика команд

Непосредственный метод адресации это

- ✓ А. Автоинкрементный метод адресации через регистр счетчик команд

Относительный метод адресации

- ✓ С. Во втором слове команды указывается относительный адрес операнда, т.е. величина смещения адреса операнда относительно адреса самой команды (текущего содержимого регистра-счетчика команд процессора).

Относительный метод адресации это

- ✓ Е. Индексный метод адресации через регистр счетчик команд

Регистровый метод адресации.

- ✓ I. В команде указывается номер регистра общего назначения, содержимое которого интерпретируется процессором как операнд

Регистровый метод адресации

- ✓ Содержимое указанного в команде регистра интерпретируется процессором как операнд

7.2. Перемещаемые программы

Какой метод адресации следует использовать в перемещаемой программе для адресации данных, расположенных в теле программы?

- ✓ непосредственный метод адресации
- ✓ относительный метод адресации

Какой метод адресации следует использовать в перемещаемой программе для адресации регистров внешних устройств?

- ✓ абсолютный метод адресации

Программа называется перемещаемой, если

- ✓ при ее размещении в разных местах памяти не требуется вносить изменений в ее код

8. Стек

Для записи числа в стек используется

- ✓ автодекрементный метод адресации

Для чтения числа из стека используется

- ✓ автоинкрементный метод адресации

Память с произвольным доступом

- ✓ В каждый момент времени для чтения доступна любая ячейка памяти независимо от ее расположения
- ✓ Информация об один раз прочитанных данных не теряется

Память с произвольным доступом

- ✓ Для доступа к данным надо указать адрес ячейки памяти
- ✓ Информация об один раз прочитанных данных не теряется

Память с произвольным доступом

- ✓ В каждый момент времени для чтения доступна любая ячейка памяти независимо от ее расположения
- ✓ Для доступа к данным надо указать адрес ячейки памяти

Память, организованная в виде стека

- ✓ Прочитать слово, находящееся на вершине стека, можно только один раз
- ✓ Записанные данные могут быть последовательно прочитаны только в порядке, обратном порядку их записи
- ✓ Для доступа к данным не надо указывать адрес ячейки памяти.
- ✓ Информация об один раз прочитанных данных теряется
- ✓ В каждый момент времени для чтения доступна только ячейка памяти, являющаяся вершиной стека

Для запоминающего устройства, организованного в виде стека, имеет место следующее

- ✓ Для доступа к данным не надо указывать адрес ячейки памяти
- ✓ В каждый момент времени для чтения доступна только ячейка памяти, являющаяся вершиной стека

Память, организованная в виде стека

- ✓ Прочитать слово, находящееся на вершине стека, можно только один раз
- ✓ Информация об один раз прочитанных данных теряется
- ✓ Записанные данные могут быть последовательно прочитаны только в порядке, обратном порядку их записи

Регистр - указатель стека всегда содержит

- ✓ адрес вершины стека
- ✓ адрес последней записанной ячейки стека

Способ доступа к ячейкам памяти организованным в виде стека

- ✓ Данные, записанные последними, читаются первыми
- ✓ Данные, записанные первыми, читаются последними

Способ доступа к ячейкам памяти с произвольным доступом

- ✓ В каждый момент времени можно прочитать данные из любой ячейки

9. Безусловные и условные переходы

Команда условного перехода используется для

- ✓ изменения содержимого регистра-счетчика команд в случае выполнения заданного условия
- ✓ для перехода на команду с указанным адресом в случае выполнения заданного условия

Команда безусловного перехода используется для

- ✓ изменения содержимого регистра-счетчика команд

В команде условного перехода, переход на новый адрес осуществляется в зависимости от

- ✓ состояния флажков регистра состояния процессора

10. Подпрограммы

Подпрограммой называется

- ✓ программный модуль, к которому можно обращаться из любого места программы любое число раз

Адрес возврата из подпрограммы запоминается

- в стеке

Вложенные подпрограммы это подпрограммы,

- ✓ вызываемые из других подпрограмм

Для возврата из подпрограммы

- нельзя использовать команду условного перехода

Для перехода к подпрограмме

- нельзя использовать команду безусловного перехода

При выполнении команды возврат из подпрограммы

- ✓ адрес возврата выталкивается из вершины стека и помещается в регистр-счетчик команд

При выполнении команды вызов подпрограммы

- ✓ содержимое регистра-счетчика команд процессора пересылается в стек
- ✓ задаваемый в команде адрес входа в подпрограмму помещается в регистр-счетчик команд

11. Архитектуры CISC и RISC

CISC-процессор это процессор с

- ✓ расширенным набором команд
- ✓ процессор с фон-неймановской архитектурой

RISC-процессор это процессор с

- ✓ сокращенным набором команд
- ✓ процессор с фон-неймановской архитектурой

Для компьютера с CISC-архитектурой является характерным

- Расширенный набор команд

Для компьютера с CISC-архитектурой является характерным

- ✓ Расширенное число разнообразных способов адресации

Для компьютера с CISC-архитектурой является характерным

- Большое число сложных форматов команд

Для компьютера с CISC-архитектурой является характерным

- ✓ Небольшое число регистров общего назначения

Для компьютера с RISC-архитектурой является характерным

- Сокращенный набор команд

Для компьютера с RISC-архитектурой является характерным

- ✓ Небольшое число простых способов адресации

Для компьютера с RISC-архитектурой является характерным

- Небольшое число простых форматов команд

Для компьютера с RISC-архитектурой является характерным

- ✓ Увеличенное число регистров общего назначения

12.1. Ввод-вывод по опросу готовности устройства

Готовность или неготовность внешнего устройства к вводу-выводу проверяется в

- регистре состояния внешнего устройства

Для программно-управляемого ввода/вывода по опросу флага готовности характерны:

- ✓ Простота программирования
- ✓ Низкая эффективность использования процессора
- ✓ Простота аппаратной реализации

Для того, чтобы осуществить передачу одного байта данных внешнему устройству в режиме ввода-вывода по опросу флага готовности внешнего устройства, необходимо:

- ✓ Переслать байт в регистр данных внешнего устройства
- ✓ Проверить разряд готовности в регистре состояния внешнего устройства

Каким образом в программным путем можно определить готовность внешнего устройства к обмену информацией?

- ✓ Проверяется состояние соответствующего разряда в регистре состояния внешнего устройства

При передаче массива данных во внешнее устройство в режиме ввода-вывода по опросу флага готовности внешнего устройства контроль количества переданных байтов или слов осуществляет

- ✓ Выполняемая процессором программа

Прием или передача данных осуществляется через

- регистр данных внешнего устройства

Режим ввода-вывода данных по опросу готовности внешнего устройства осуществляется:

- ✓ по инициативе выполняемой процессором программы
- ✓ под управлением выполняемой процессором программы

Состоянием разряда (флага) готовности внешнего устройства в его регистре состояния управляет:

- ✓ внешнее устройство

Что такое регистр данных внешнего устройства?

- ✓ Регистр, находящийся в интерфейсе внешнего устройства, через который осуществляется прием или передача данных

Что такое регистр состояния внешнего устройства?

- ✓ Регистр, находящийся в интерфейсе внешнего устройства, с помощью которого осуществляется проверка готовности внешнего устройства к приему или передаче данных
- ✓ Регистр, находящийся в интерфейсе внешнего устройства, с помощью которого можно разрешить или запретить прерывание внешнему устройству

12.2. Ввод-вывод в режиме прерывания

Адрес возврата из прерывания запоминается

- В стеке

При переходе в режим прерывания процессор запоминает в стеке:

- ✓ Текущее содержимое регистра состояния процессора
- ✓ Текущее содержимое регистра-счетчика команд

При получении от внешнего устройства сигнала требования прерывания процессор

- Продолжает выполнение текущей команды до ее завершения

Состояние разряда разрешения/запрета прерывания внешнего устройства в его регистре состояния устанавливается:

- ✓ А. из программы соответствующей командой

Что делает внешнее устройство при получении от процессора сигнала предоставления прерывания?

- ✓ Передает процессору адрес своего вектора прерывания

Что необходимо сделать для того, чтобы запретить прерывание ВСЕМ внешним устройствам?

- ✓ Установить состояние высокого приоритета процессора в его регистре состояния

Что произойдет, если два внешних устройства одновременно затребуют прерывание?

- ✓ Требование прерывания от устройства с более низким приоритетом будет проигнорировано
- ✓ Прерывание будет предоставлено устройству с более высоким приоритетом

Что происходит при выполнении команды возврат из прерывания?

- ✓ Адрес возврата в прерванную программу из вершины стека пересылается в регистр-счетчик команд процессора
- ✓ Выталкивается из стека и записывается в регистр состояния процессора прежнее слово состояния процессора

Что такое "вложенное" прерывание?

- ✓ Прерывание, которое происходит во время обслуживания другого прерывания

Что такое вектор прерывания?

- ✓ Ячейки памяти, закрепленные за определенным внешним устройством
- ✓ Ячейки памяти, в одной из которых указан адрес программы обслуживания прерывания от конкретного устройства
- ✓ Ячейки памяти, в одной из которых указано слово состояния процессора

Ввод-вывод данных в режиме прерывания осуществляется:

- ✓ под управлением выполняемой процессором программы
- ✓ по инициативе внешнего устройства

Для чего при реализации механизма прерывания необходим стек?

- ✓ Для запоминания текущего содержимого регистра состояния процессора
- ✓ Для реализации вложенных прерываний
- ✓ Для запоминания текущего содержимого регистра-счетчика команд

Какие операции необходимо осуществить в программе, для того, чтобы обеспечить возможность работы с внешним устройством в режиме прерывания?

- ✓ Установить в первой ячейке вектора прерывания адрес программы обслуживания прерывания

- ✓ Установить разряд разрешения прерывания в регистре состояния внешнего устройства в состояние "разрешено"
- ✓ Установить во второй ячейке вектора прерывания значение слова состояния процессора
- ✓ Разместить в памяти программу обслуживания прерывания
- ✓ Установить в регистре указателе стека адрес вершины стека

Какие принципиальные проблемы должны решаться при реализации ввода/вывода в режиме прерывания программы?

- ✓ Нужен механизм, обеспечивающий переход процессора на выполнение программы обслуживания именно того устройства, которое затребовало прерывание
- ✓ Нежелательна жесткая привязка программы обслуживания к определенному месту в памяти ЭВМ
- ✓ Необходим механизм, позволяющий программным путем запрещать или разрешать прерывания всем или некоторым внешним устройствам, когда это необходимо
- ✓ Необходимо разрешать конфликтные ситуации, возникающие при одновременном требовании прерывания несколькими внешними устройствами
- ✓ Работа основной (прерываемой) программы должна приостанавливаться без какого-либо ущерба для ее выполнения

Каким образом можно запретить прерывание конкретному внешнему устройству?

- ✓ Сбросить в нуль разряд приоритета в регистре состояния внешнего устройства

Каким образом при окончании обслуживания внешнего устройства, вызвавшего прерывание, осуществляется возврат в прерванную программу?

- ✓ Выполняется команда возврат из прерывания

Каким образом процессор узнает адрес, по которому в памяти ЭВМ находится программа обслуживания внешнего устройства, затребовавшего прерывание?

- ✓ Считывает этот адрес из первой ячейки вектора прерывания внешнего устройства

Можно ли для возврата из подпрограммы использовать команду возврат из прерывания?

- Нельзя

Можно ли для возврата из прерывания использовать команду возврат из подпрограммы?

- Нельзя

Откуда появляется информация в векторе прерывания?

- В. Устанавливается программным путем

По сравнению с вводом выводом по опросу флага готовности для режима прерывания характерны:

- ✓ Более сложная аппаратная реализация
- ✓ Более высокая эффективность использования процессора
- ✓ Более сложное программирование

После выполнения какого действия процессор посылает внешнему устройству сигнал предоставления прерывания?

- После запоминания в стеке текущего содержимого регистра-счетчика команд и регистра состояния процессора

Можно ли для возврата из прерывания использовать команду возврат из подпрограммы?

- ✓ Нельзя, потому, что эта команда не обеспечивает полного восстановления прежнего (т.е. до прерывания) состояния процессора

При включении питания компьютера разряд разрешения/запрета прерывания для внешнего устройства в регистре состояния его интерфейса автоматически устанавливается в состояние:

- прерывание запрещено

При передаче массива данных во внешнее устройство в режиме прерывания контроль количества переданных байтов осуществляет

- ✓ Выполняемая процессором программа

Установите правильную последовательность действий, происходящих при акте предоставления прерывания внешнему устройству

- | | | |
|----|---|----|
| 1 | Внешнее устройство посылает процессору сигнал требования прерывания | 1 |
| 2 | Процессор получает от внешнего устройства сигнал запроса прерывания | 2 |
| 3 | Процессор завершает выполняемую команду | 3 |
| 4 | Процессор запоминает в стеке содержимое регистра счетчика команд и регистра состояния | 4 |
| 5 | Процессор посылает внешнему устройству сигнал предоставления прерывания | 5 |
| 6 | Внешнее устройство получает от процессора сигнал предоставления прерывания | 6 |
| 7 | Внешнее устройство передает процессору адрес своего вектора прерывания | 7 |
| 8 | Процессор получает от внешнего устройства адрес его вектора прерывания | 8 |
| 9 | Процессор пересылает в регистр счетчик команд содержимое первой ячейки вектора прерывания | 9 |
| 10 | Процессор пересылает в свой регистр состояния содержимое второй ячейки вектора прерывания | 10 |
| 11 | Процессор считывает и выполняет первую команду программы обслуживания прерывания | 11 |

Последовательность действий, выполняемых при предоставлении прерывания внешнему устройству.

Используемые сокращения:

ВУ - внешнее устройство,

ОЗУ - оперативное запоминающее устройство,

ТрПр - сигнал требования прерывания,

ПрПр - сигнал предоставления прерывания,

ТрПДП - сигнал требования прямого доступа к памяти,

ПрПДП - сигнал предоставления прямого доступа к памяти,

ТрПДВУ - сигнал требования прямого доступа к внешнему устройству,

ПрПДВУ - сигнал предоставления прямого доступа к внешнему устройству.

- | | | | | | |
|----|-----------|--------------|---------------------|-------------------|-----------------|
| 1. | ВУ | посылает | процессору | сигнал | ТрПр |
| 2. | Процессор | получает от | ВУ | сигнал | ТрПр |
| 3. | Процессор | завершает | выполняемую команду | | |
| 4. | Процессор | запоминает в | стек | содержимое своего | счетчика команд |
| | | и | регистр состояния | | |
| 5. | Процессор | посылает | ВУ | сигнал | ПрПр |

6. ВУ получает от процессора сигнал ПрПр
7. ВУ передает процессору адрес своего вектора прерывания
8. Процессор получает от ВУ адрес его вектора прерывания
9. Процессор пересылает в свой счетчик команд содержимое первой ячейки вектора прерывания
10. Процессор пересылает в свой регистр состояния содержимое второй ячейки вектора прерывания
11. Процессор считывает и выполняет первую команду программы обслуживания прерывания

12.3. Подпрограммы

Адрес возврата из подпрограммы запоминается

- в стеке

Вложенные подпрограммы это подпрограммы,

- ✓ вызываемые из других подпрограмм

Для возврата из подпрограммы

- нельзя использовать команду условного перехода

Для возврата из подпрограммы

- нельзя использовать команду безусловного перехода

Для перехода к подпрограмме

- нельзя использовать команду условного перехода

Для перехода к подпрограмме

- нельзя использовать команду безусловного перехода

Подпрограммой называется

- ✓ программный модуль, к которому можно обращаться из любого места программы любое число раз

При выполнении команды возврат из подпрограммы

- ✓ адрес возврата выталкивается из вершины стека и помещается в регистр-счетчик команд

При выполнении команды вызов подпрограммы

- ✓ задаваемый в команде адрес входа в подпрограмму помещается в регистр-счетчик команд
- ✓ содержимое регистра-счетчика команд процессора пересылается в стек

12.4. Ввод-вывод в режиме прямого доступа к памяти

В режиме прямого доступа к памяти управление вводом-выводом осуществляется:

- ✓ Внешним устройством

Для каких из перечисленных внешних устройств целесообразно использовать ввод/вывод информации в режиме прямого доступа к памяти?

- ✓ Монитор
- ✓ Внешнее запоминающее устройство
- ✓ Сканер

Для перехода к режиму предоставления внешнему устройству прямого доступа к памяти

- ✓ Процессор не должен сохранять свое текущее состояние
- ✓ Процессор не должен доводить выполнение текущей команды до конца

По сравнению с другими режимами для ввода-вывода в режиме прямого доступа к памяти характерно:

- ✓ Более высокая скорость передачи данных
- ✓ Более сложная аппаратная реализация
- ✓ Более высокая эффективность использования процессора

После выполнения каких действий процессор посылает внешнему устройству сигнал предоставления прямого доступа к памяти?

- После завершения процессором очередного цикла обращения к каналу (к оперативной памяти)

При передаче массива данных во внешнее устройство в режиме прямого доступа к памяти контроль количества переданных байтов или слов осуществляет

- ✓ Контроллер внешнего устройства

При передаче массива данных во внешнее устройство в режиме прямого доступа к памяти формирование адресов ячеек памяти, из которых осуществляется пересылка данных, осуществляет

- ✓ Контроллер внешнего устройства

При приеме массива данных из внешнего устройства в режиме прямого доступа к памяти контроль количества переданных байтов или слов осуществляет

- ✓ Контроллер внешнего устройства

Целесообразно ли использовать для ввода данных с клавиатуры режим прямого доступа к памяти?

- Не целесообразно

Целесообразно ли использовать для обмена данными с жестким диском режим прямого доступа к памяти?

- Да целесообразно

Что делает внешнее устройство после получения от процессора сигнала предоставления прямого доступа к памяти?

- Осуществляет управление магистралями компьютера и его памятью для передачи данных

Что должен "уметь" делать контроллер прямого доступа к памяти?

- ✓ Формировать сигнал требования прямого доступа
- ✓ Формировать адреса ячеек памяти с данными
- ✓ Посчитывать число переданных байтов

Что означает аббревиатура DMA?

- D. Прямой доступ к памяти

12.5. Ввод-вывод - сравнение режимов

Расположите перечисленные режимы ввода/вывода в порядке возрастания скорости передачи данных

1 По опросу готовности внешнего устройства

Минимальная скорость



- | | |
|--------------------------|------------------------------------|
| 2 Режим прерывания | <div>Средняя скорость ▾</div> |
| 3 Прямой доступ к памяти | <div>Максимальная скорость ▾</div> |

Расположите перечисленные режимы ввода/вывода в порядке возрастания степени их пригодности для обмена данными с внешними запоминающими устройствами

- | | |
|--|--------------------------------------|
| 1 По опросу готовности внешнего устройства | <div>Практически не пригоден ▾</div> |
| 2 Режим прерывания | <div>Плохо пригоден ▾</div> |
| 3 Прямой доступ к памяти | <div>Хорошо пригоден ▾</div> |

Расположите перечисленные режимы ввода/вывода в порядке возрастания степени их пригодности для обмена данными со сканером

- | | |
|--|--------------------------------------|
| 1 По опросу готовности внешнего устройства | <div>Практически не пригоден ▾</div> |
| 2 Режим прерывания | <div>Плохо пригоден ▾</div> |
| 3 Прямой доступ к памяти | <div>Хорошо пригоден ▾</div> |

Расположите перечисленные режимы ввода/вывода в порядке возрастания степени их пригодности для передачи больших массивов данных

- | | |
|--|--------------------------------------|
| 1 По опросу готовности внешнего устройства | <div>Практически не пригоден ▾</div> |
| 2 Режим прерывания | <div>Плохо пригоден ▾</div> |
| 3 Прямой доступ к памяти | <div>Хорошо пригоден ▾</div> |

Расположите перечисленные режимы ввода/вывода в порядке возрастания степени их пригодности для передачи данных для отображения на экране монитора

- | | |
|--|--------------------------------------|
| 1 По опросу готовности внешнего устройства | <div>Практически не пригоден ▾</div> |
| 2 Режим прерывания | <div>Плохо пригоден ▾</div> |
| 3 Прямой доступ к памяти | <div>Хорошо пригоден ▾</div> |

Расположите перечисленные режимы ввода/вывода в порядке увеличения сложности их реализации

- | | |
|--|-------------------------------------|
| 1 По опросу готовности внешнего устройства | <div>Минимальная сложность ▾</div> |
| 2 Режим прерывания | <div>Средняя сложность ▾</div> |
| 3 Прямой доступ к памяти | <div>Максимальная сложность ▾</div> |

Расположите перечисленные режимы ввода/вывода в порядке увеличения эффективности использования процессорного времени

- | | |
|--|--|
| 1 По опросу готовности внешнего устройства | <div>Самая низкая эффективность ▾</div> |
| 2 Режим прерывания | <div>Средняя эффективность ▾</div> |
| 3 Прямой доступ к памяти | <div>Самая высокая эффективность ▾</div> |

Какой режим ввода вывода целесообразно использовать для обмена данными с видеомонитором?

- Режим прямого доступа к памяти

Какой режим ввода-вывода целесообразно использовать для обмена данными с жестким диском?

- Режим прямого доступа к памяти

Какой режим ввода вывода целесообразно использовать для обмена данными с клавиатурой?

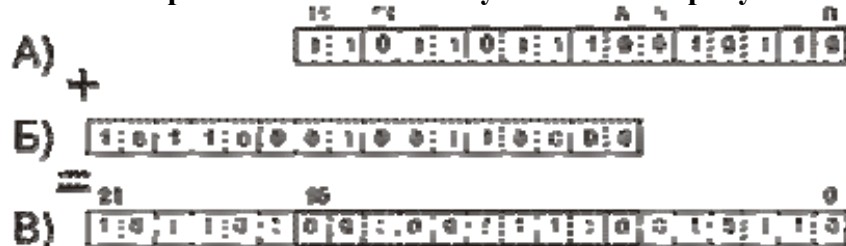
- Режим прерывания

Какой режим ввода вывода целесообразно использовать для обмена данными со сканером?

- Режим прямого доступа к памяти

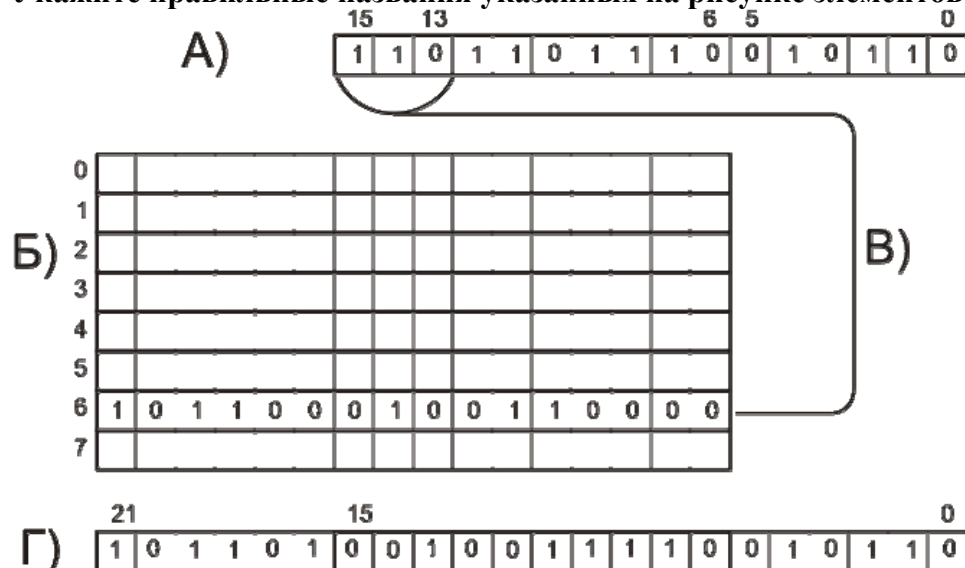
14. Управление памятью

Укажите правильные названия указанных на рисунке элементов



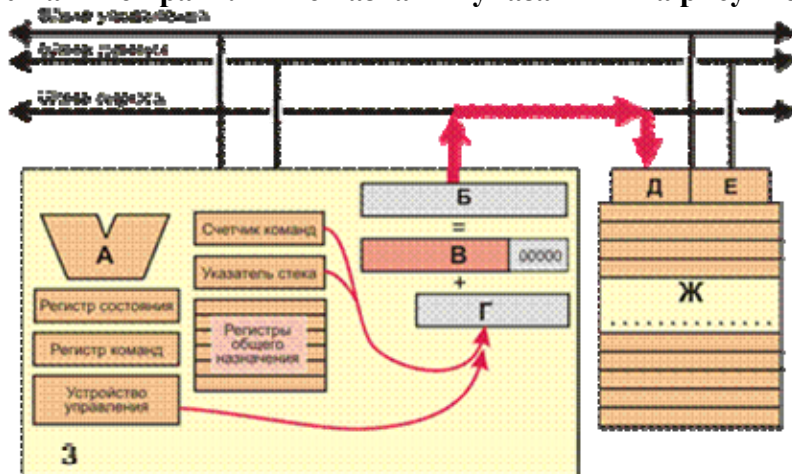
- A)
- B)
- B)

Укажите правильные названия указанных на рисунке элементов



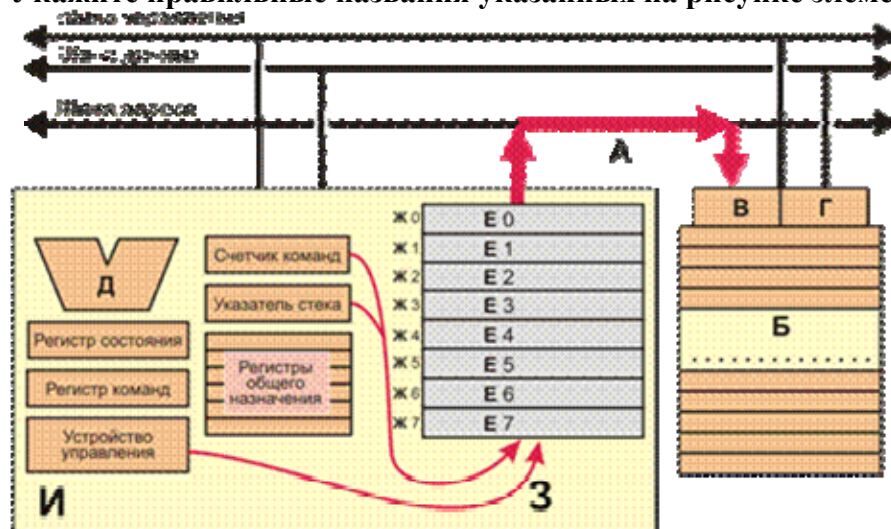
- A)
- B)
- B)
- Г)

Укажите правильные названия указанных на рисунке элементов



- А)
- Б)
- В)
- Г)
- Д)
- Е)
- Ж)
- З)

Укажите правильные названия указанных на рисунке элементов



- А)
- Б)
- В)
- Г)
- Д)
- Е)
- Ж)

З)

И)

Адреса физической и виртуальной памяти

- не совпадают

Виртуальной памятью (виртуальным адресным пространством) называют

- адресное пространство, которое представляется разработчику при создании кода программы

Виртуальные станицы одной задачи должны загружаться в физическую память

- распределяясь по любым свободным областям памяти

Для того, чтобы в 16-ти разрядной ЭВМ получить 20-ти разрядный физический адрес, необходимо

- ✓ прибавить к нему 20-ти разрядную константу

Для того, чтобы в 16-ти разрядной ЭВМ получить 22-х разрядный физический адрес, необходимо

- ✓ прибавить к нему 22-х разрядную константу

Для того, чтобы на компьютере можно было выполнять программу, размер которой превышает размер физической памяти,

- ✓ на диске должно быть выделено место для файла виртуальной памяти
- ✓ в компьютере должны быть специальные программные средства обеспечивающие управление виртуальным и физическим адресными пространствами
- ✓ в процессоре должны быть аппаратные средства преобразования виртуальных адресов в физические

Для того, чтобы на компьютере можно было выполнять программу, размер которой превышает размер физической памяти,

- ✓ должны быть специальные программные средства обеспечивающие управление виртуальным и физическим адресными пространствами
- ✓ в процессоре должны быть аппаратные средства преобразования виртуальных адресов в физические

К какому объему памяти можно адресоваться с помощью 16-разрядного адреса?

- ✓ 65 536 Байт
- ✓ 64 Килобайта

К какому объему памяти можно адресоваться с помощью 20-разрядного адреса?

- ✓ 1 Мегабайт
- ✓ 1 048 576 Байт

К какому объему памяти можно адресоваться с помощью 22-разрядного адреса

- ✓ 4 Мегабайта

К какому объему памяти можно адресоваться с помощью 30-разрядного адреса?

- ✓ 1 Гигабайт

К какому объему памяти можно адресоваться с помощью 32-разрядного адреса?

- ✓ 4 Гигабайта

Кэш-память

- ✓ имеет меньшую емкость, чем основная оперативная память
- ✓ недоступна для адресации из программы
- ✓ используется для ускорения работы компьютера

Кэш-память

- ✓ имеет более высокое быстродействие, чем основная оперативная память
- ✓ размещается между процессором и основной оперативной памятью

Кэш-память это

- ✓ промежуточная буферная память, предназначенная для согласования быстродействия процессора и основной памяти

Могут ли 16-разрядные процессоры работать с памятью объемом 2 Мегабайта?

- Могут некоторые 16-разрядные процессоры

Может ли в компьютере выполняться программа, размер которой превышает размер физической памяти?

- ✓ Может, если специальные программные средства обеспечивают управление виртуальным и физическим адресными пространствами
- ✓ Может, если в процессоре имеются аппаратные средства преобразования виртуальных адресов в физические
- ✓ Да может

Может ли в компьютере выполняться программа, размер которой превышает размер физической памяти?

- ✓ Может, если в процессоре имеются аппаратные средства преобразования виртуальных адресов в физические
- ✓ Может, если на диске выделено место для файла виртуальной памяти
- ✓ Может, если специальные программные средства обеспечивают управление виртуальным и физическим адресными пространствами

Может ли виртуальная память (логическое адресное пространство) быть больше физической памяти?

- Да может

Преобразование виртуального адреса в физический адрес осуществляется:

- прибавления двоичной константы к виртуальному адресу

Преобразование виртуального адреса в физический осуществляется путем:

- ✓ прибавления к виртуальному адресу константы

При отображении виртуального адресного пространства на физическое адресное пространство:

- ✓ непрерывная группа адресов (страница) виртуального адресного пространства преобразуется в соответствующую непрерывную группу адресов (страницу) физического адресного пространства

Регистры адреса страниц в диспетчере памяти процессора используются для:

- ✓ хранения констант перемещения для виртуальных страниц

Физической памятью называют

- множество имеющихся в ОЗУ компьютера ячеек оперативной памяти

Что такое свопинг

- Процедура загрузки виртуальных страниц задачи с диска в физическую память взамен выгружаемых неиспользуемых страниц

Сколько разрядов в физическом адресе требуется для адресации памяти объемом 1 Гигабайт?

30

Сколько разрядов в физическом адресе требуется для адресации памяти объемом 2 Мегабайт?

21

Сколько Разрядов в физическом адресе требуется для адресации памяти объемом 4 Гигабайта?

32

Сколько разрядов в физическом адресе требуется для адресации памяти объемом 4 Мегабайт?

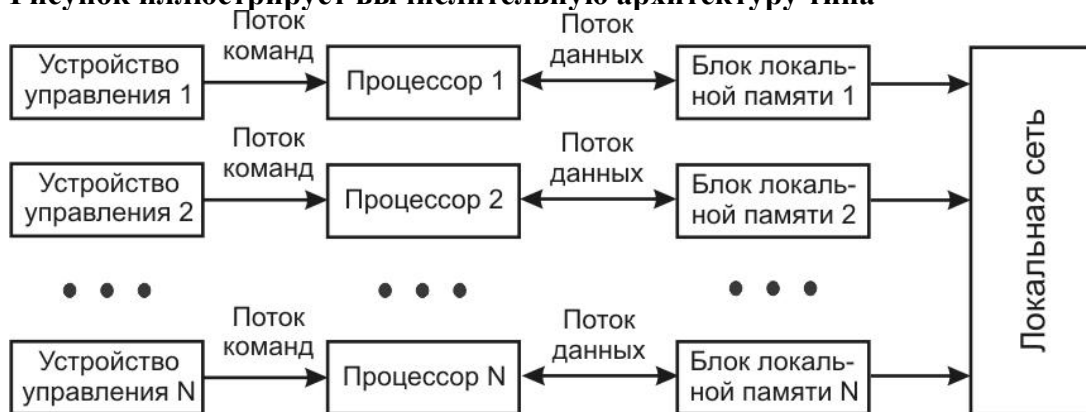
22

Сколько разрядов в физическом адресе требуется для адресации? памяти объемом 1 Мегабайт?

20

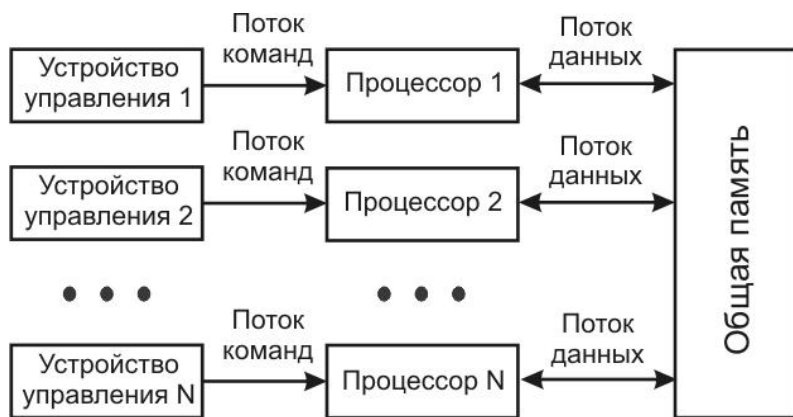
15. Повышение производительности

Рисунок иллюстрирует вычислительную архитектуру типа



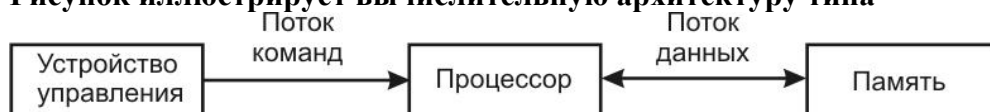
Множественный	поток команд
Множественный	поток данных
Мультипроцессор с распределенной памятью, кластер	

Рисунок иллюстрирует вычислительную архитектуру типа



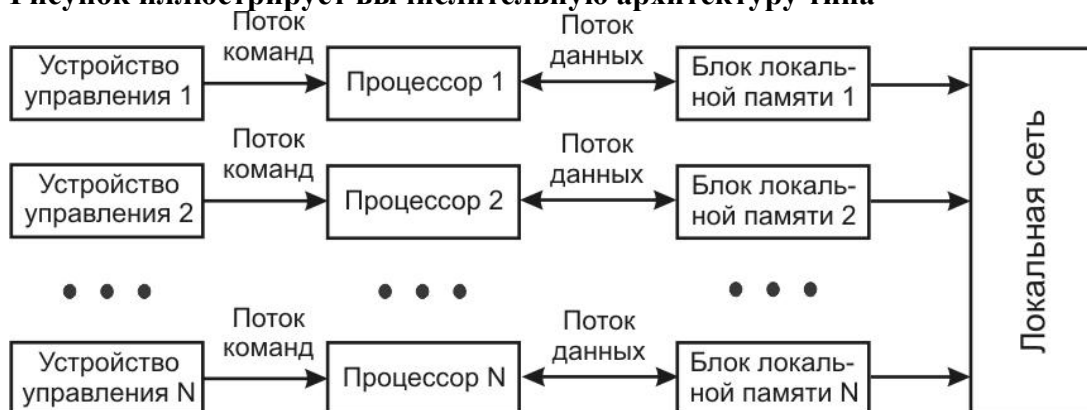
Множественный	поток команд
Множественный	поток данных

Рисунок иллюстрирует вычислительную архитектуру типа



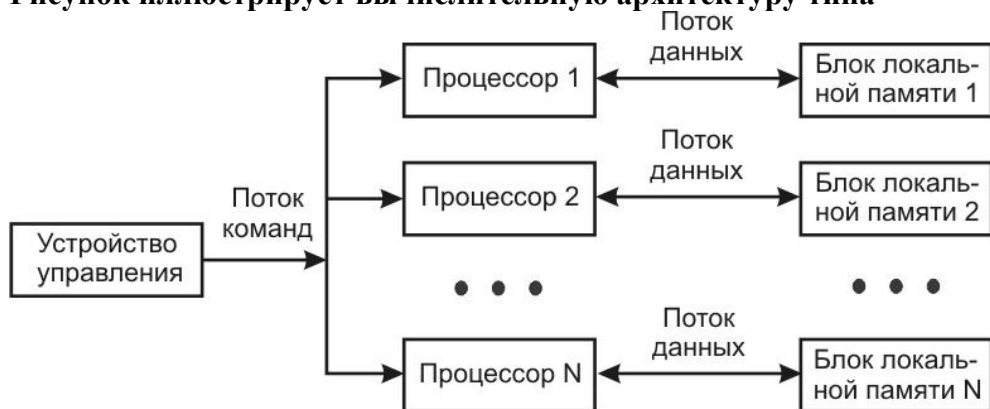
Одинарный	поток команд
Одинарный	поток данных

Рисунок иллюстрирует вычислительную архитектуру типа



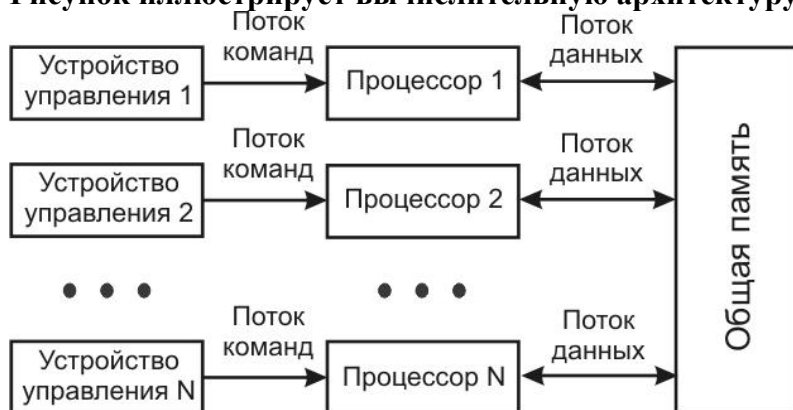
Множественный	поток команд
Множественный	поток данных

Рисунок иллюстрирует вычислительную архитектуру типа



Одинарный	поток команд
Множественный	поток данных

Рисунок иллюстрирует вычислительную архитектуру типа



Множественный	поток команд
Множественный	поток данных
Симметричный мультипроцессор с общей памятью	

Для многопроцессорных кластерных систем является характерным:

- ✓ Отказ одного узла не приводит к потере работоспособности всего комплекса
- ✓ Использование группы полноценных компьютеров, взаимосвязанных через локальную сеть
- ✓ Отказ одного узла не приводит к потере работоспособности всего комплекса
- ✓ Возможность наращивания мощности комплекса в процессе эксплуатации
- ✓ Каждый процессор работает со своим ОЗУ

Для многопроцессорных кластерных систем является характерным:

- ✓ Каждый процессор работает со своим ОЗУ
- ✓ Простая масштабируемость. Возможность включения в комплекс любого числа компьютеров
- ✓ Снижение соотношения цена/производительность
- ✓ Могут использоваться компьютеры массового производства
- ✓ Возможность наращивания мощности комплекса в процессе эксплуатации

Для многопроцессорных кластерных систем является характерным

- ✓ Возможность наращивания мощности комплекса в процессе эксплуатации
- ✓ Использование группы полноценных компьютеров, взаимосвязанных через локальную сеть
- ✓ Каждый процессор работает со своим ОЗУ
- ✓ Отказ одного узла не приводит к потере работоспособности всего комплекса
- ✓ Отказ одного процессора не приводит к потере работоспособности всего комплекса

Для многопроцессорных кластерных систем является характерным

- ✓ Простая масштабируемость. Возможность включения в комплекс любого числа компьютеров
- ✓ Возможность наращивания мощности комплекса в процессе эксплуатации
- ✓ Могут использоваться компьютеры массового производства
- ✓ Снижение соотношения цена/производительность
- ✓ Каждый процессор работает со своим ОЗУ

Для симметричных многопроцессорных систем является характерным:

- ✓ Наличие общей памяти, к которой имеют доступ все процессоры
- ✓ Отказ одного процессора не приводит к потере работоспособности комплекса в целом
- ✓ Доступ процессоров к общим средствам ввода-вывода
- ✓ Все процессоры способны выполнять одинаковый набор функций
- ✓ Наличие двух или более одинаковых или близких по характеристикам процессоров, установленных на одной материнской плате

Для симметричных многопроцессорных систем является характерным:

- ✓ В случае отказа одного из процессоров задача может выполняться на другом
- ✓ Управление всем комплексом общей операционной системой
- ✓ Все процессоры способны выполнять одинаковый набор функций
- ✓ Возможность наращивания мощности комплекса в процессе эксплуатации
- ✓ Наличие общей памяти, к которой имеют доступ все процессоры

Для симметричных многопроцессорных систем является характерным

- ✓ Наличие двух или более одинаковых или близких по характеристикам процессоров, установленных на одной материнской плате
- ✓ Наличие общей памяти, к которой имеют доступ все процессоры
- ✓ Доступ процессоров к общим средствам ввода-вывода
- ✓ Отказ одного процессора не приводит к потере работоспособности комплекса в целом
- ✓ Все процессоры способны выполнять одинаковый набор функций

Для симметричных многопроцессорных систем является характерным

- ✓ Все процессоры способны выполнять одинаковый набор функций
- ✓ В случае отказа одного из процессоров задача может выполняться на другом
- ✓ Управление всем комплексом общей операционной системой
- ✓ Возможность наращивания мощности комплекса в процессе эксплуатации
- ✓ Наличие общей памяти, к которой имеют доступ все процессоры

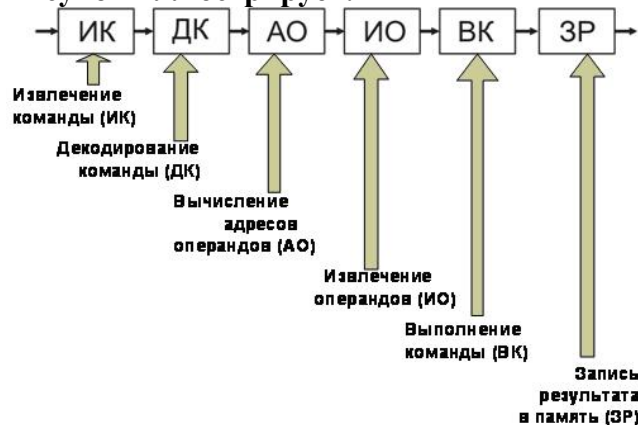
Распараллеливание алгоритма выполнения задачи это

- ✓ одновременное выполнение набором операционных устройств операций, составляющих задачу, над множеством одновременно поступающих входных данных

Распараллеливание алгоритма выполнения задачи это

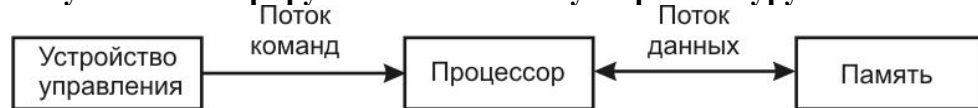
- ✓ одновременное выполнение набором операционных устройств операций, составляющих задачу, над множеством одновременно поступающих входных данных

Рисунок иллюстрирует:



- ✓ последовательность микроопераций, составляющих машинную команду

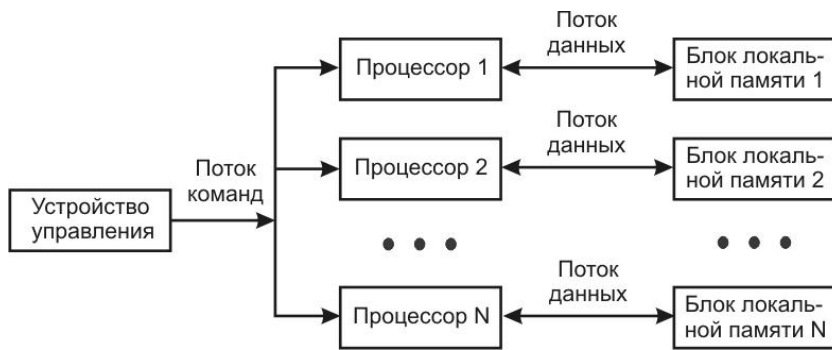
Рисунок иллюстрирует вычислительную архитектуру типа



_____ поток команд, одинарный поток данных

- одинарный

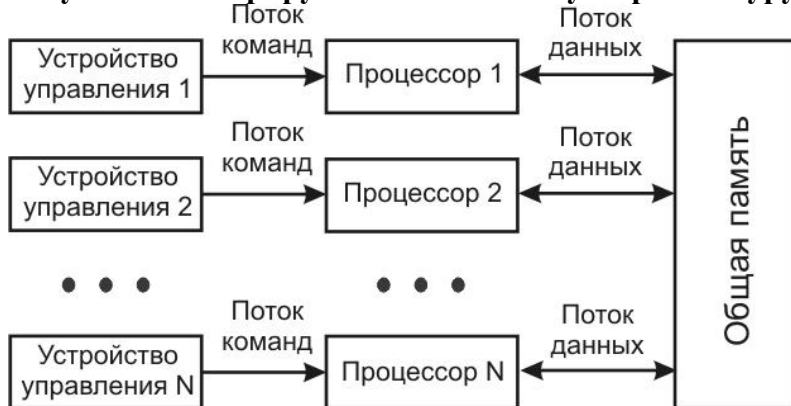
Рисунок иллюстрирует вычислительную архитектуру типа



_____ поток команд, множественный поток данных

- одинарный

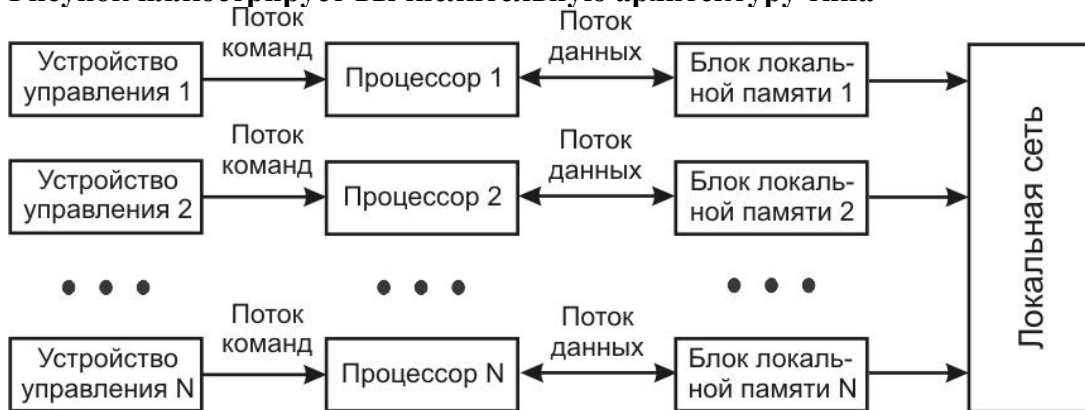
Рисунок иллюстрирует вычислительную архитектуру типа



_____ поток команд, множественный поток данных

- множественный

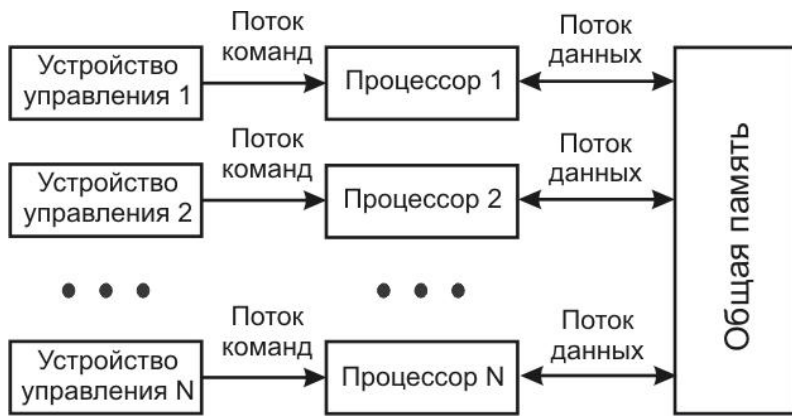
Рисунок иллюстрирует вычислительную архитектуру типа



_____ поток команд, множественный поток данных

- множественный

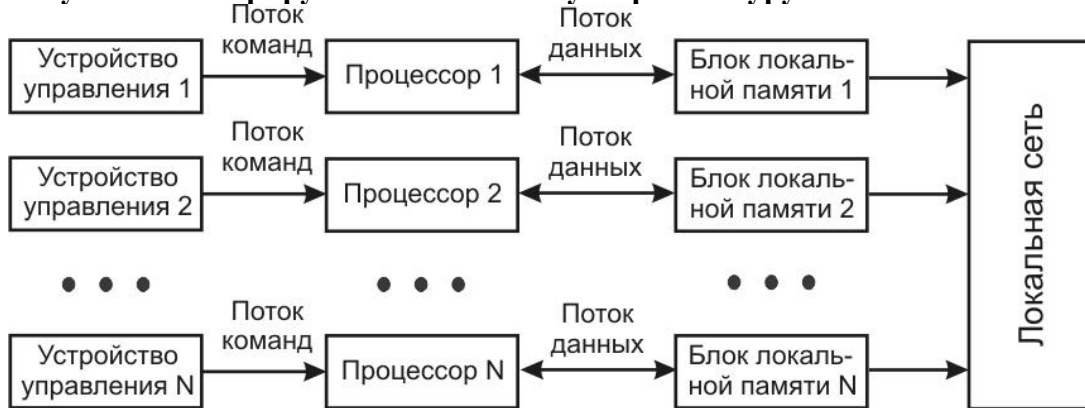
Рисунок иллюстрирует вычислительную архитектуру типа



Поток команд, _____ поток данных, мультипроцессор с распределенной памятью

- множественный

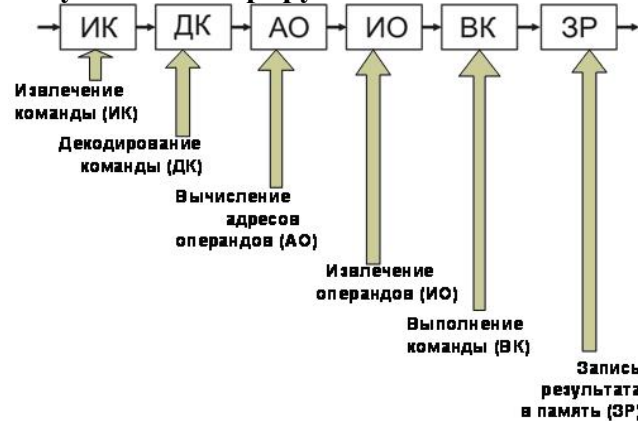
Рисунок иллюстрирует вычислительную архитектуру типа



поток команд, _____ поток данных, мультипроцессор с распределенной памятью/кластер

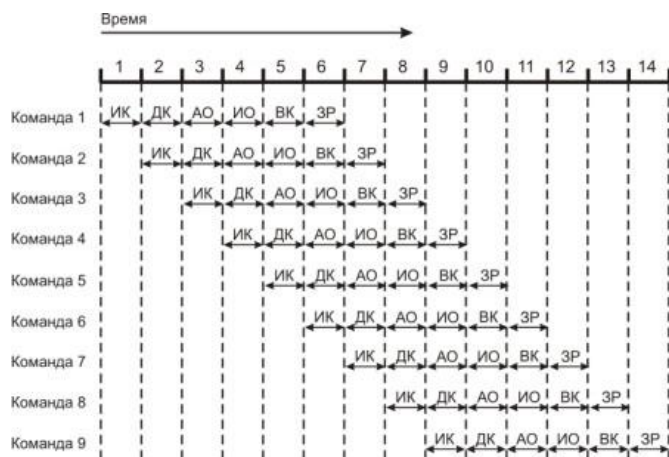
- множественный

Рисунок иллюстрирует



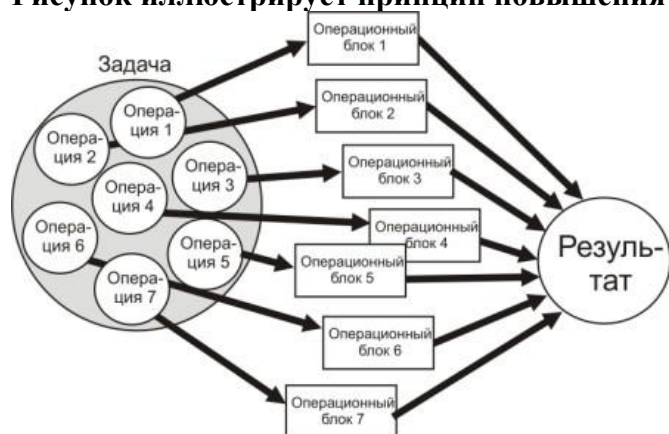
- последовательность микроопераций, составляющих машинную команду

Рисунок иллюстрирует принцип повышения производительности путем



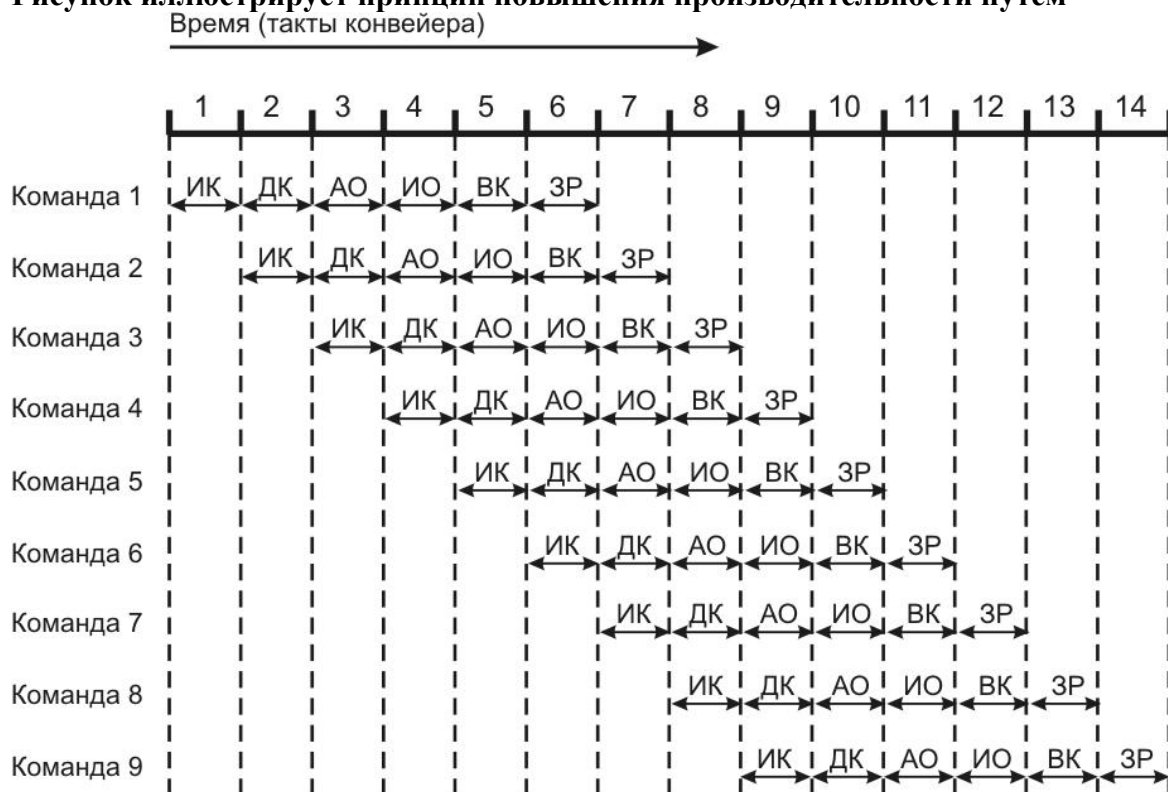
- ✓ конвейерного выполнения операций

Рисунок иллюстрирует принцип повышения производительности путем



- ✓ распараллеливания выполнения операций

Рисунок иллюстрирует принцип повышения производительности путем



- D. конвейерного выполнения операций