



<https://forms.gle/KeoVcGCEZdeM13f37>





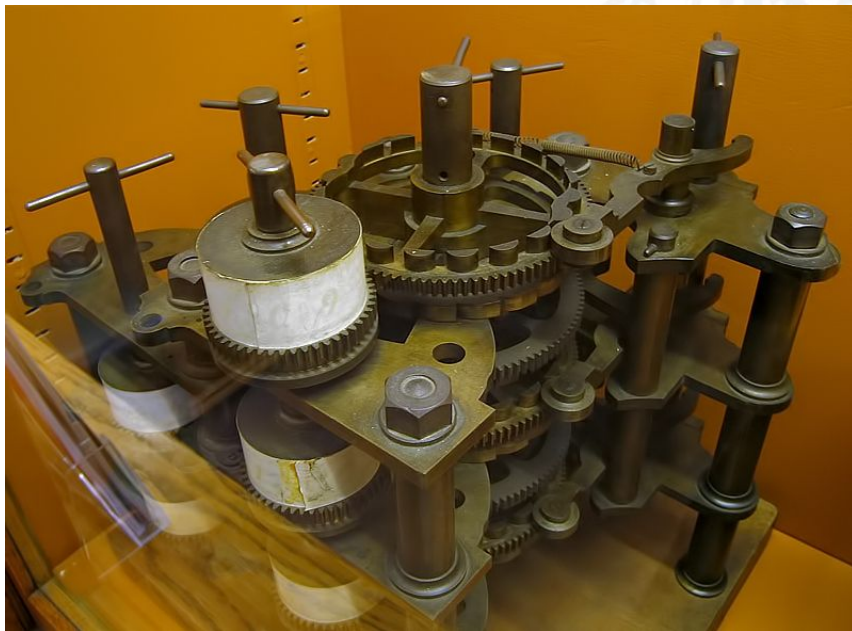
Архитектура ЭВМ и операционные системы



Архитектура ЭВМ

Направление «Искусственный интеллект и наука о данных», 23.Б16-мм, 23.Б18-мм

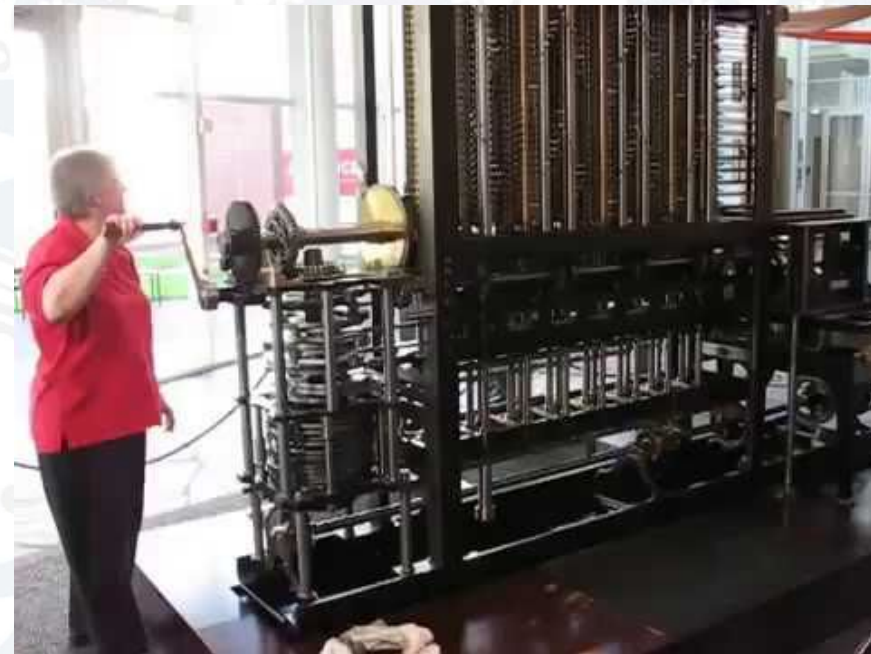
20.10.2023



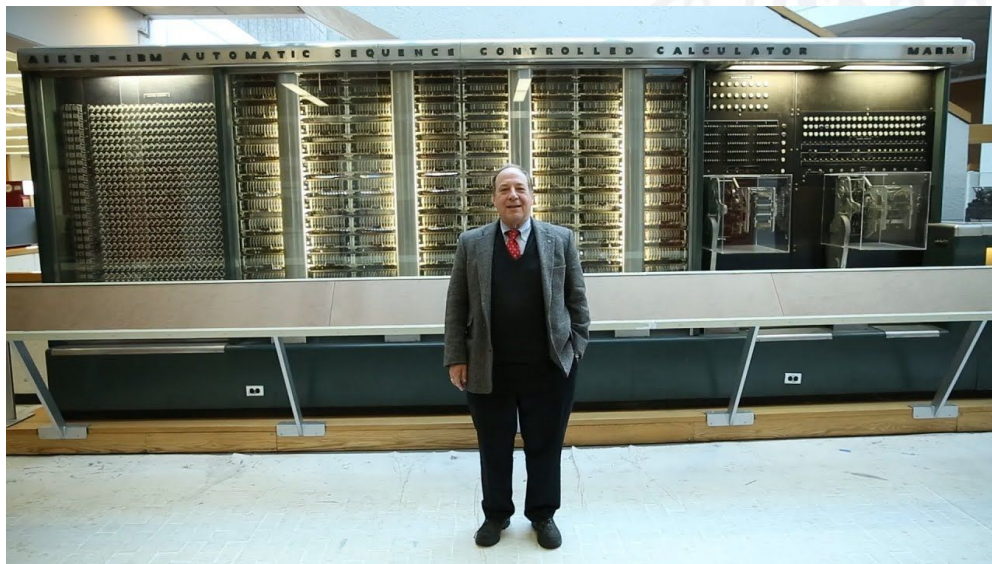
Фрагмент разностной машины
Бэббиджа

- **Разностная машина** — механический «компьютер», предназначенный для автоматизации вычислений
- Изобретена английским математиком Чарльзом Бэббиджем **в начале 19 века**
- В свое время машина так и не была построена до конца, но считается **прообразом современного компьютера**

- **Аналитическая машина** — все еще механический «компьютер», который считался более мощным и универсальным инструментом, чем разностная машина
- Изобретена всё тем же английским математиком Чарльзом Бэббиджем в **первой половине 19 века**
- В этой машине впервые появились **память (перфокарты), вычислительное устройство** и устройства **ввода и вывода**.
- Механизм был **программируемым**: для него были написаны **первые в мире программы (Ада Лавлейс)**



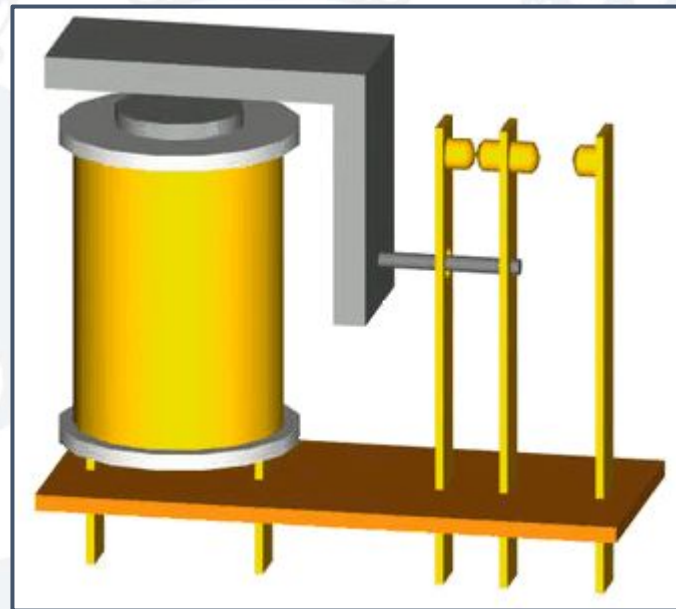
Аналитическая машина Бэббиджа
и какая-то женщина (не Ада Лавлейс)



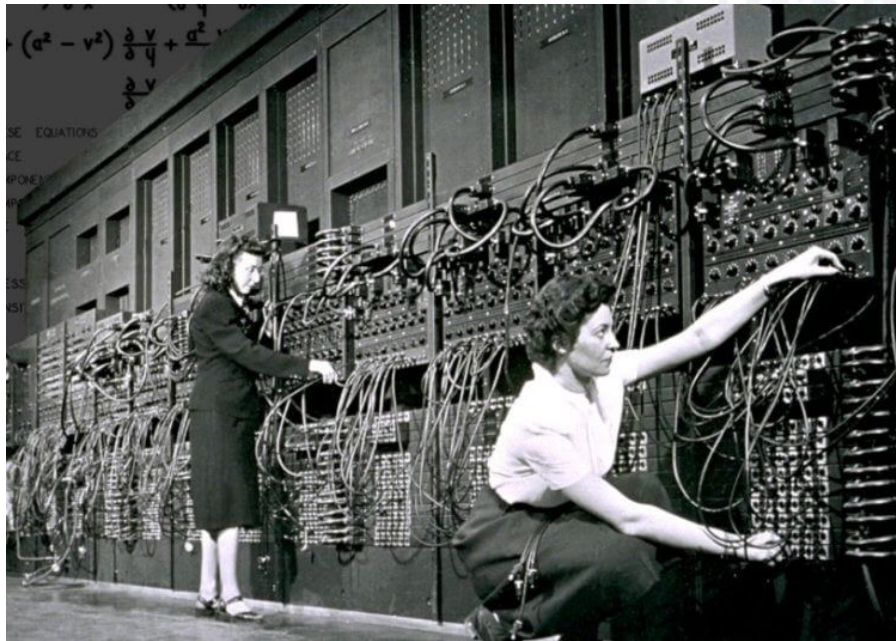
Mark I и какой-то мужчина (не Говард Эйкен)

- **Mark I** — программируемый автоматический вычислитель, один из первых **электромеханических** компьютеров
- Изобретен и реализован американским инженером IBM Говардом Эйкеном в 1940-ых годах
- Главное отличие от предшественников — **использование электромеханических реле**

- При подаче тока на входы (снизу), в катушке образуется **электромагнитное поле**
- Поле притягивает серую металлическую штуку, которая деформирует контакты справа
- Таким образом, если подать ток на входы, то **положение реле переключается**



Принцип работы электромагнитного реле



ENIAC и какие-то женщины (не Эккерт и Мокли)

- ENIAC — программируемый автоматический вычислитель, первый **электронный** компьютер
- Изобретен и реализован группой инженеров под руководством американских ученых **Эккерта и Мокли** в 1943-44 годах. В качестве научного консультанта выступал **Джон фон Нейман** — венгеро-американский математик
- Главное отличие от предшественников — **использование электрических ламп**
- В дальнейшем и по сей день применяются более совершенные **транзисторы**

- **Электронная лампа и транзистор** работают по **схожему принципу**: на вход №1 подается ток и «открывает» основной поток
- Принцип работы можно **сравнить с краном на трубе**: открытие и закрытие крана это подача и отсутствие сигнала на базе транзистора
- Главные преимущества резисторов перед лампами заключается в том, что **резисторы меньше по размеру и дешевле ламп**

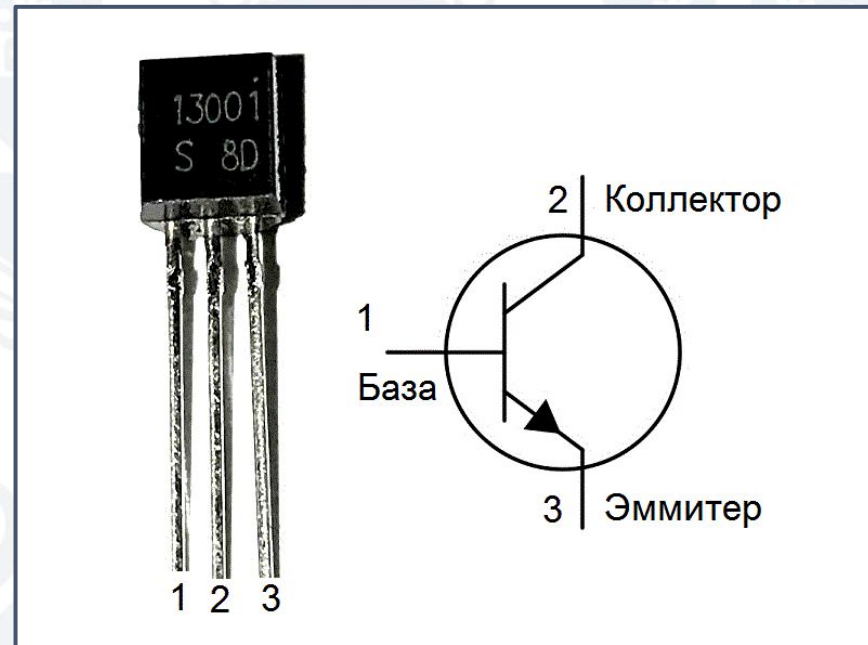
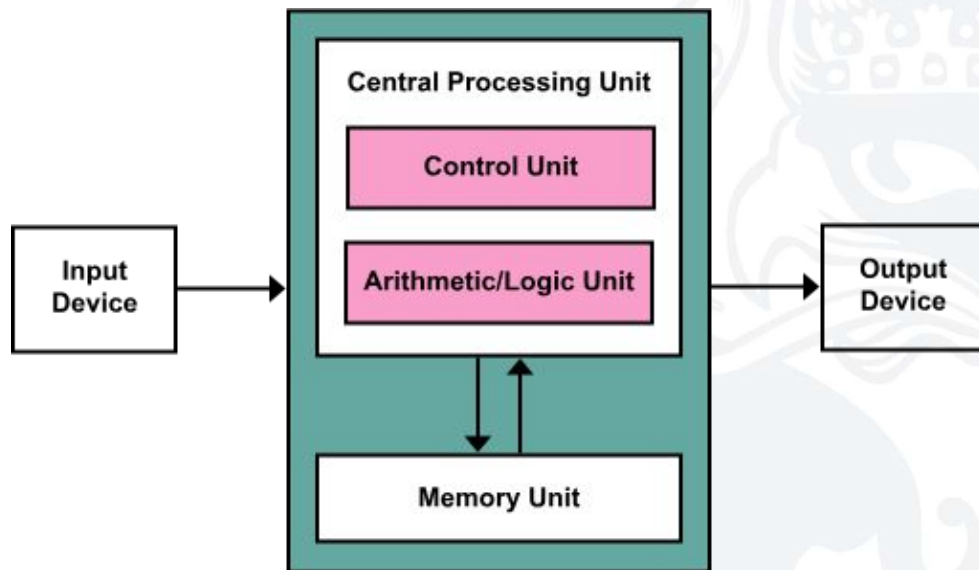


Схема работы транзистора



Абстракция архитектуры
фон Неймана

- При создании ENIAC фон Нейман заложил архитектуру компьютера, которая активно используется до сих пор
- Архитектура содержит в себе **управляющий модуль** (Control Unit), который достает из **памяти** (Memory Unit) команды и данные и передает их в **арифметико-логическое устройство** (ALU), которое умеет производить вычисления и отдавать ответ обратно в управляющий модуль
- Помимо этих трех элементов в ней есть **устройства ввода и вывода**



Архитектура фон Неймана удовлетворяет трем принципам:

- **Принцип однородности памяти:** команды и данные (которые использует управляющее устройство) хранятся в одной и той же памяти и внешне неразличимы
- **Принцип адресности:** ячейки памяти имеют адреса, по которым управляющее устройство может получать их содержимое
- **Принцип программного управления:** алгоритм выполнения задач состоит из команд. Команды хранятся в памяти в виде слов

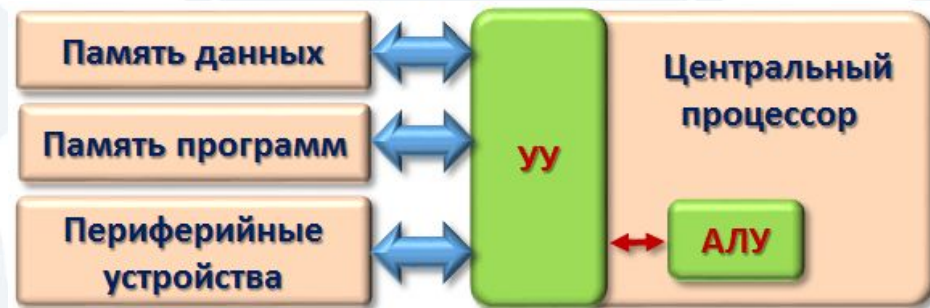


Недостатки архитектуры фон Неймана



Санкт-Петербургский
государственный
университет

- Устройства в архитектуре фон Неймана **общаются по общей шине** — транспортному каналу, который **блокируется одним из устройств** для передачи в него данных или инструкций
- Как это влияет на скорость работы? **Сначала нужно получить команды, а потом данные**
- Проблему решает **гарвардская архитектура**, которая имеет отдельную память для данных и для команд: управляющее устройство может одновременно доставать инструкции и данные по разным шинам
- Однако, **она не прижилась из-за сложности** построения отдельной памяти для команд





- Современные компьютеры прошли эволюцию от полностью **механических вычислителей**, которые работали от паровой энергии и на шестеренках, до **электромеханических реле, электронных ламп и транзисторов**
- Главными отличительными признаками современных компьютеров от машин 18-19 веков являются **возможность их программирования и наличие памяти**
- Популярная модель архитектуры ЭВМ — архитектура фон Неймана — была придумана автором в середине 20 века и используется по сей день





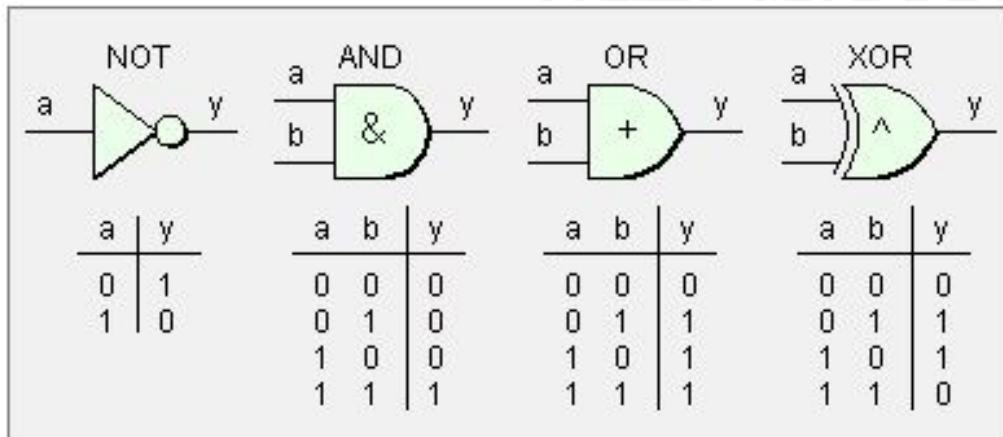
Архитектура ЭВМ и предпосылки возникновения сетей

● ● ● ● ● ●

Устройство АЛУ и памяти ЭВМ

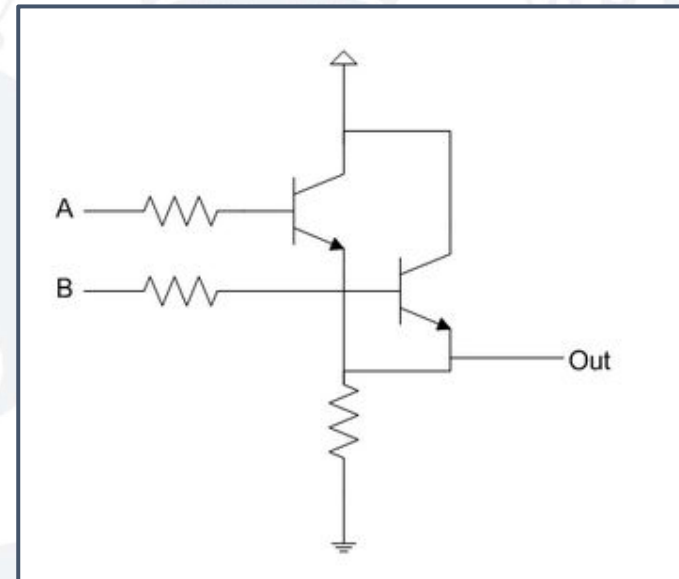
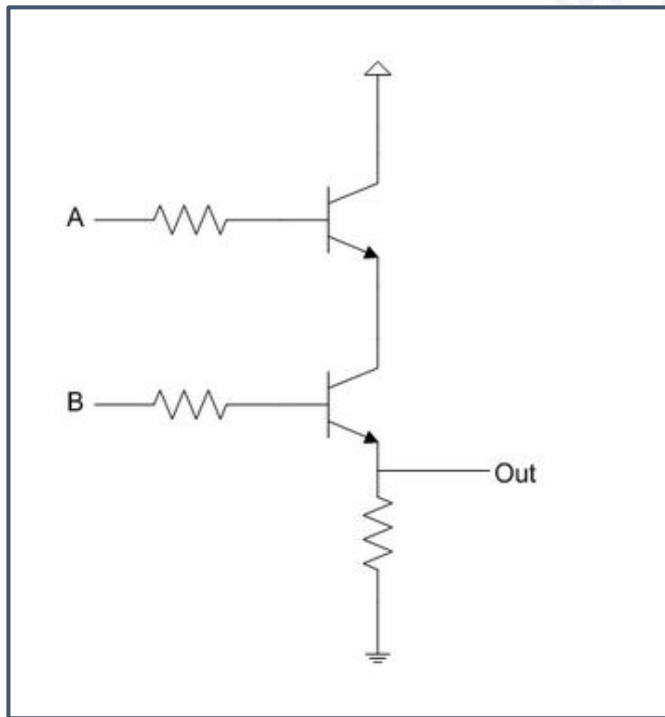
Направление «Искусственный интеллект и наука о данных», 23.Б16-мм, 23.Б18-мм

20.10.2023



Основные логические вентили

- Транзисторы помогают собирать более нетривиальные структуры — **логические вентили**
- Логические вентили моделируют работу одной из логических операций, то есть имеют один или два входа и один выход



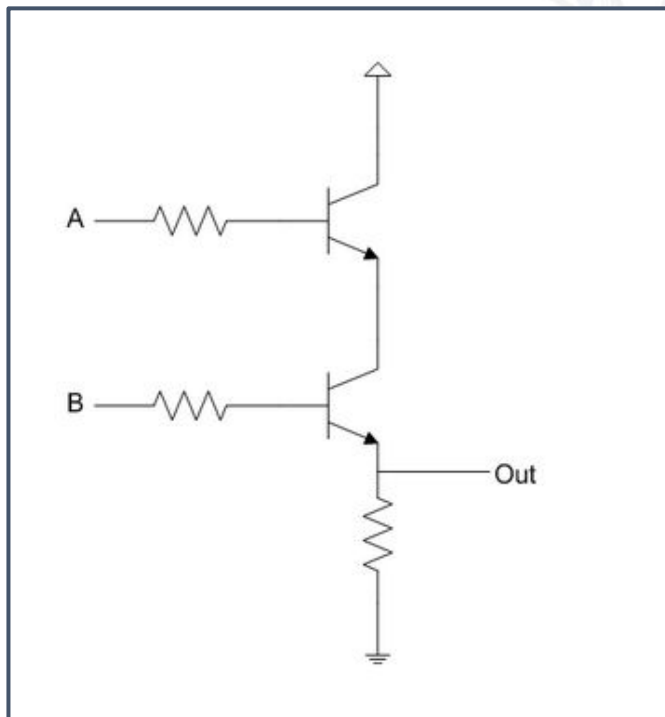


Схема логического вентиля
AND

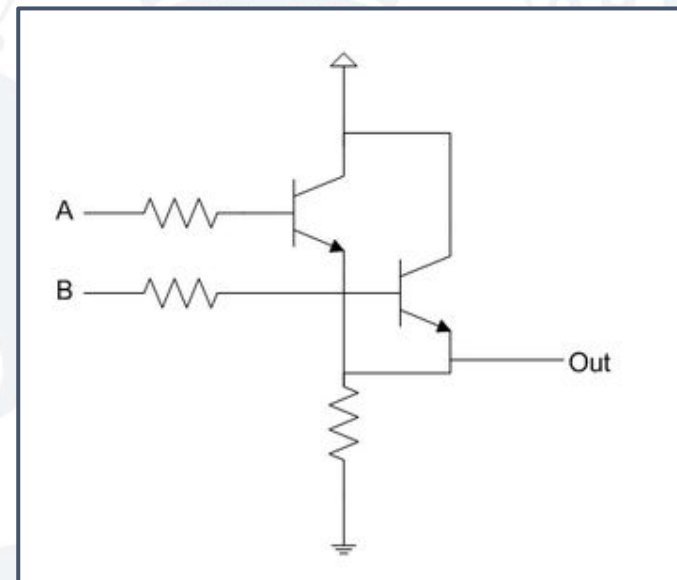


Схема логического вентиля **OR**

Сумматор двоичных чисел



Санкт-Петербургский
государственный
университет

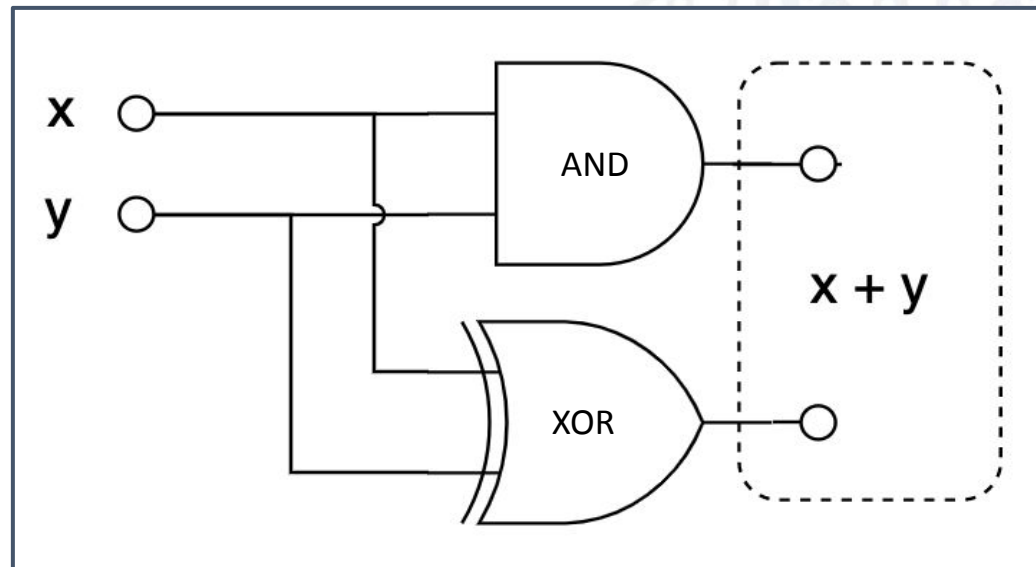
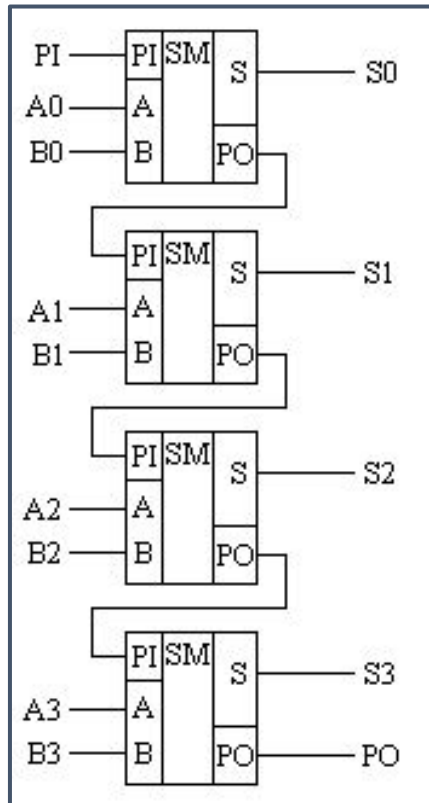


Схема сумматора одноразрядных
двоичных чисел

- С помощью логических вентиляей можно строить **логические схемы**
- Одна из таких — **сумматор** одноразрядных двоичных чисел
- На вход подаются два числа, на выходе после конкатенации слов — их сумма

x	y	x+y
0	0	00
0	1	01
1	0	01
1	1	10



- Примерно по таким же принципам строится **сумматор многоразрядных двоичных чисел**
- Верхний из выходов — бит разряда, который *запоминается*, пока остальные не будут вычислены. Вычисление начинается с младшего разряда
- Нижний из выходов в каждой из схем – **бит переноса**, который используется в следующей схеме
- **Арифметико-логическое устройство** почти полностью состоит из таких и похожих схем для умножения, деления и др.



- Конечно, любая физическая реализация такой комбинационной логики будет срабатывать не мгновенно, **а с некоторой задержкой**
- Например, если у каждого вентиля задержка d , то в схеме многоразрядного сумматора s_0 будет вычислено с задержкой d , s_1 с задержкой $2d$, s_3 с задержкой $3d$ и так далее
- Длина задержки (которая, вообще говоря, всегда одинакова) **регулируется тактовым генератором**
- Этот механизм **похож на внутренние часы**, по которым синхронизируются все устройства
- При этом устройства делятся на быстрые и медленные: первые выполняют операции за меньшее число тактов





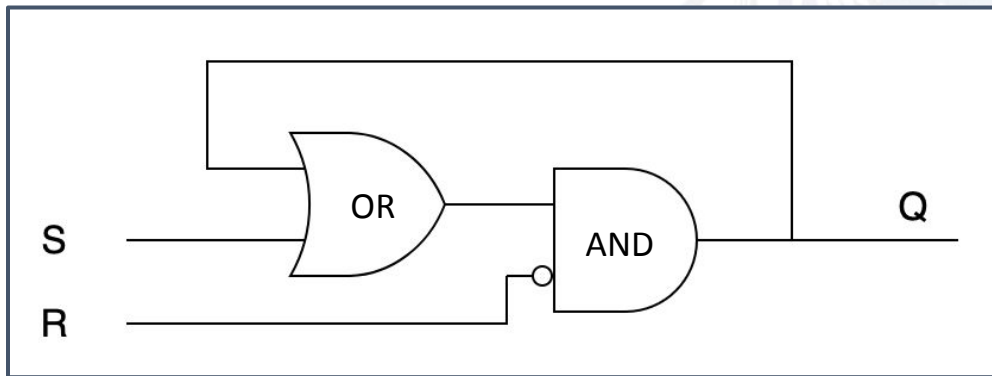
- Очень неудобно, когда разряды числа оказываются вычислены в разные моменты времени
- Хотелось бы научиться **в некоторых точках схемы остановиться и подождать**, пока какие-то значения не стабилизируются (например, дождаться вычисления всех разрядов числа)
- Чтобы подождать, нам потребуется некоторое время «помнить» значения в нужных точках схемы. Как добиться такого эффекта?



На чем основана любая память компьютера?



Санкт-Петербургский
государственный
университет



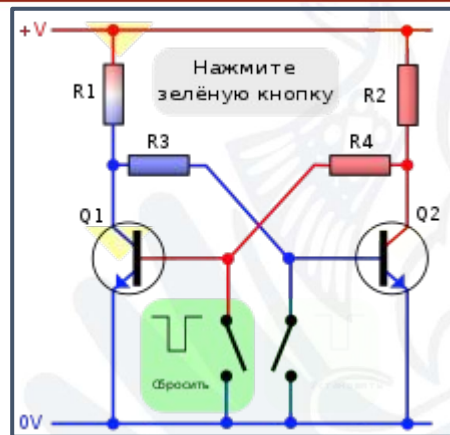
R	S	Q
1	0	0
0	1	1
0	0	Q^{-1}
1	1	—

- Чтобы добиться такого эффекта, мы выйдем за пределы комбинационной логики (в которой сигнал течет только «вперед») и **добавим в схему цикл**
- Теперь сигнал **на выходе Q** передается на один из **входов**
- Если подать сигнал на R, а на S оставить 0, то на выходе Q **всегда получим 0**
- Если подать сигнал на S, а на R оставить 0, то на выходе Q **всегда получим 1**
- Если переключить R и S в состояние 0, то схема перейдет в состояние **хранения** последнего «записанного» значения

Как на самом деле выглядит триггер

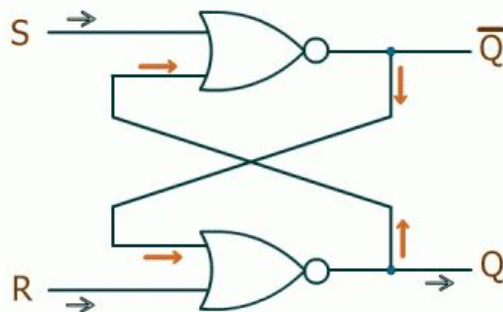


Санкт-Петербургский
государственный
университет



- Идея, которую мы поняли на предыдущем слайде, лежит в основе **RS-триггера** (R — reset, S — set)
- Его схема еще более запутана, но принцип остается тот же: мы добавили в схему цикл и замкнули сохраненное значение в схеме

Схема RS-триггера на вентилях ИЛИ-НЕ





По типу удаленности памяти от управляющего устройства выделяют три уровня:

- **Первый уровень:** регистры и кэш процессора. Эта память доступна управляющему устройству без задержек, ее регистры находятся прямо в процессоре
- **Второй уровень:** оперативная память. Эта память доступна процессору по общей шине, ее объем меньше и она работает быстрее, чем память третьего уровня
- **Третий уровень:** вся остальная память, включая внешние устройства. Самая большая по объему, самая дешевая и самая отдаленная от процессора память

При этом реализация всех трёх уровней так или иначе основана на идее **RS-триггера** (то есть является **программируемой**)





- В основе арифметико-логического устройства и памяти лежат так называемые **логические вентили**, которые состоят из транзисторов
- Логические вентили собираются в логические схемы, которые имеют несколько входов и выходов
- Ячейка памяти основывается на **RS-триггерах**, в его основе лежит идея обратной связи — **добавления цикла** в комбинационную схему





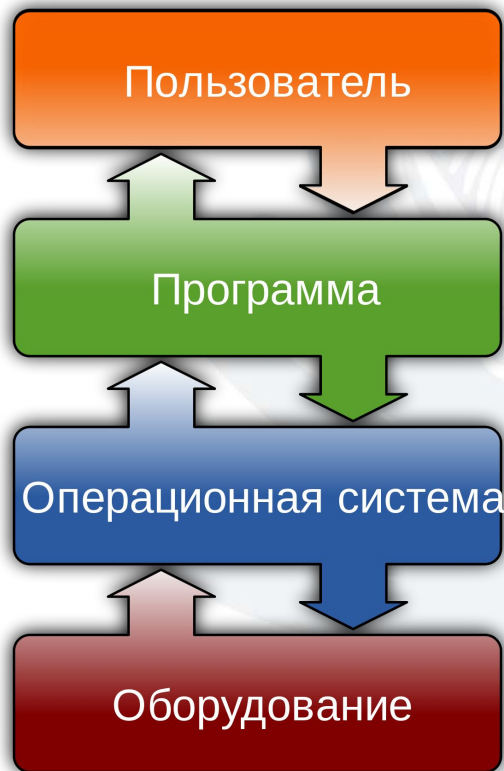
Архитектура ЭВМ и предпосылки возникновения сетей



Операционные системы

Направление «Искусственный интеллект и наука о данных», 23.Б16-мм, 23.Б18-мм

20.10.2023



- Понятно, что обычному человеку будет очень сложно, например, передавать биты в АЛУ, читать выход и расшифровывать ответ
- Более того, **человек будет узким местом** в этой системе, потому что современные **компьютеры выполняют операции гораздо быстрее**, чем человек сможет на них среагировать
- Поэтому работа с низкоуровневыми элементами любого компьютера скрыта под **несколькими слоями абстракции**
- Для удобного взаимодействия пользователя с компьютером разрабатываются **операционные системы**



- **Операционная система** — комплекс программ, обеспечивающий управление аппаратными средствами компьютера, организующий работу с файлами и выполнение прикладных программ, осуществляющий ввод и вывод данных
- На сегодняшний день операционная система — это **первый и основной набор программ, загружающийся в компьютер**



Если говорить подробнее, то в функции ОС будут включены:

- Управление процессами
- Управление памятью
- Управление файлами и внешними устройствами I/O
- Защита данных и администрирование
- Интерфейс прикладного программирования
- Пользовательский интерфейс

Чтобы обеспечить **эффективный контроль использования памяти**, ОС должна выполнять следующие функции:

- отображение адресного пространства процесса на конкретные области физической памяти
- распределение памяти между конкурирующими процессами
- контроль доступа к адресным пространствам процессов
- выгрузка процессов (целиком или частично) во внешнюю память, когда в оперативной памяти недостаточно места
- учет свободной и занятой памяти





- Разработка ОС — **сложный процесс**, требующий миллионов строк кода и сотен тысяч человеко-часов
- Он требует знания об аппаратуре, безопасности, о современных представлениях об интерфейсе, об алгоритмах, о представлении данных и т.д.
- Поэтому **операционных систем «мало»**
- Кроме того, большинство нужных знаний **быстро устаревают**:
 - конкретные программные решения (2-3 года)
 - архитектура и операционная платформа (5-7 лет)
 - основные идеи организации ОС (7-10-15 и более лет, циклически)
 - фундаментальные знания, математические основы (десятки и сотни лет)



- Абстракция над низкоуровневыми аппаратными элементами нужна не только для **удобного использования компьютера** человеком, но и для организации **эффективного хранения памяти, безопасности и быстродействия**
- Операционная система является **прослойкой** между человеком и написанными им программами и аппаратными устройствами компьютера
- Разработка ОС — сложный процесс, требующий огромных ресурсов и знаний, поэтому **таких систем очень мало**





Архитектура ЭВМ и операционные системы



Направление «Искусственный интеллект и наука о данных», 23.Б16-мм, 23.Б18-мм

20.10.2023