



Введение в Архитектуру ЭВМ

Для вопросов по курсу: natalya.shevskaya@moevm.info
Префикс в теме письма [CS_23XX]

*Шевская Наталья Владимировна
СПбГЭТУ “ЛЭТИ”, ФКТИ, МОЭВМ*



Позиционные системы счисления

- Десятичная система счисления

$$56789_{10} = 5 * 10^4 + 6 * 10^3 + 7 * 10^2 + 8 * 10^1 + 9 * 10^0$$

- Двоичная система счисления

$$10011_2 = 1 * 2^4 + 0 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0$$

- Восьмеричная система счисления

$$567_8 = 5 * 8^2 + 6 * 8^1 + 7 * 8^0$$

- Шестнадцатеричная система счисления

$$56A8C_{16} = 5 * 16^4 + 6 * 16^3 + A * 16^2 + 8 * 16^1 + C * 16^0$$



Булева Алгебра

Логическое
отрицание “НЕ”

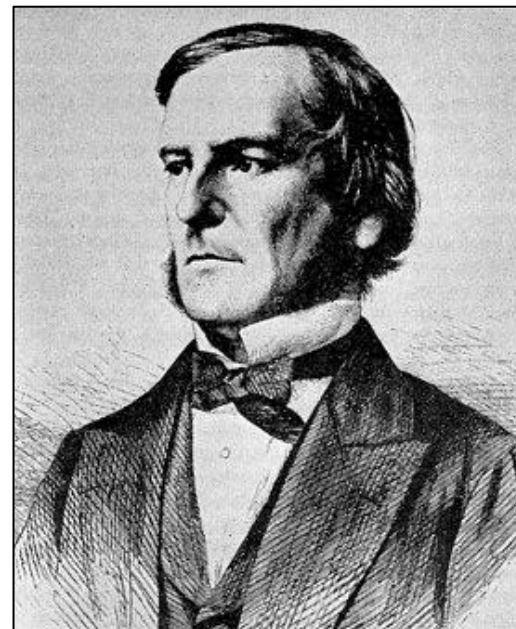
a	$\neg a$
0	1
1	0

Логическое
умножение “И”

a	b	$a \& b$
0	0	0
0	1	0
1	0	0
1	1	1

Логическое
сложение “ИЛИ”

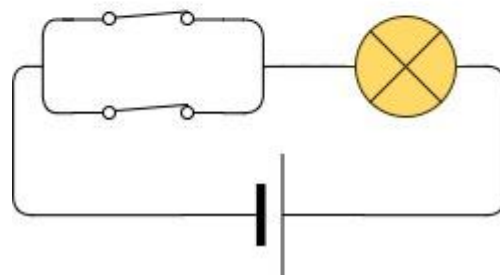
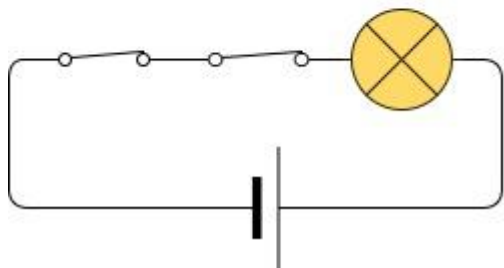
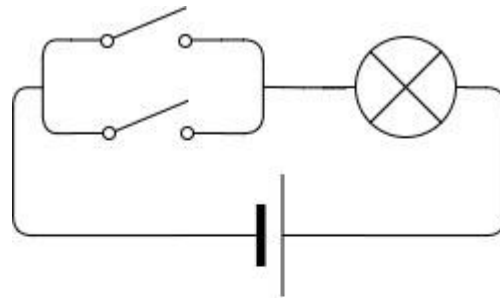
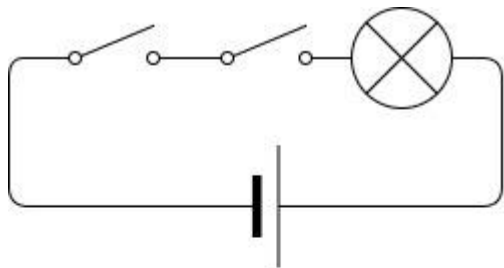
a	b	$a b$
0	0	0
0	1	1
1	0	1
1	1	1



Джордж Буль
(2 ноября 1815 -- 8 декабря
1864)



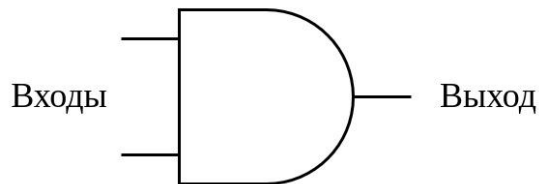
Применение. Схемы И и ИЛИ



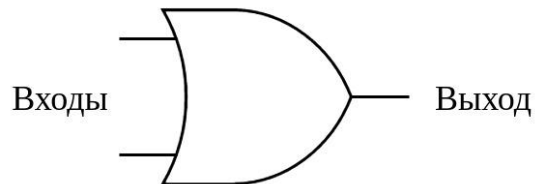


Применение. Вентили и инвертор

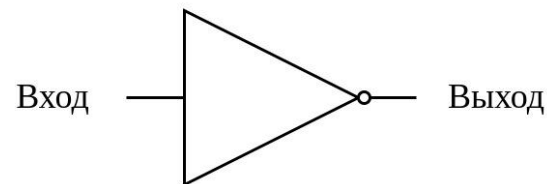
И



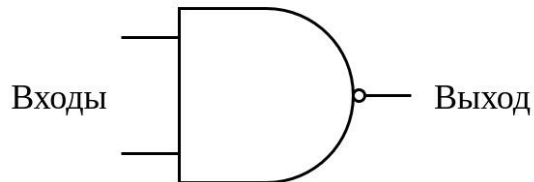
ИЛИ



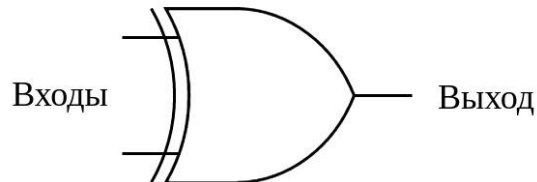
Инвертор



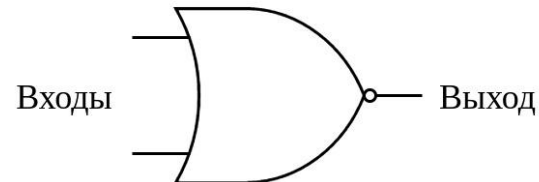
И-НЕ



Исключающее
ИЛИ

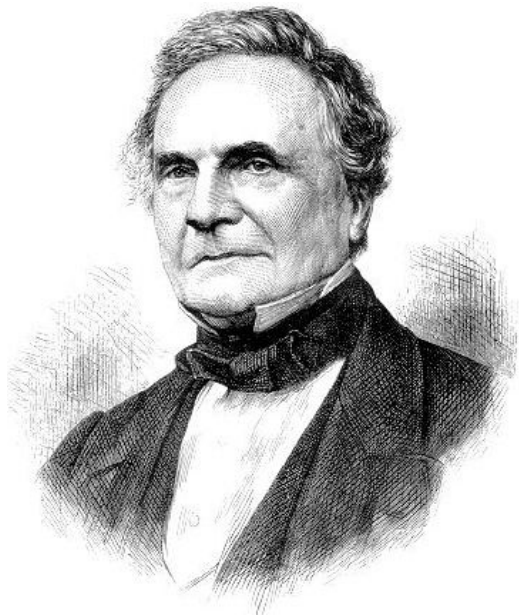


ИЛИ-НЕ





Разностная машина (Difference Engine)



Чарлз Бэббидж (26 декабря 1791 -
18 октября 1871) английский
математик, изобретатель



Первое изобретение (незавершенное):
Разностная машина (1820 - 1833 гг.)

Конструкция разностной машины
основывалась на использовании
десятичной системы счисления.



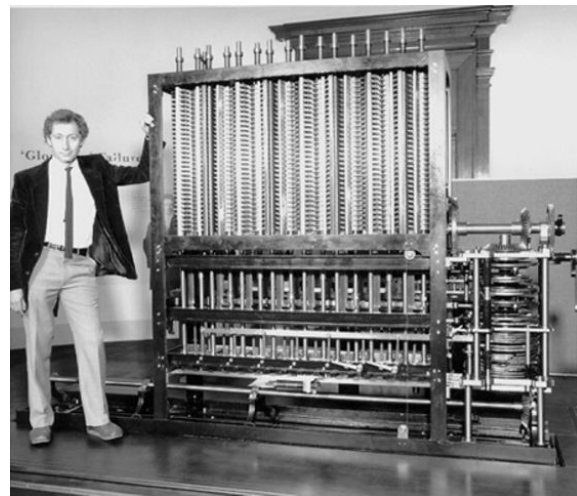
Аналитическая машина

Конструкция:

- хранилище (память)
- «мельница» (арифметическое устройство)
- устройство ввода-вывода

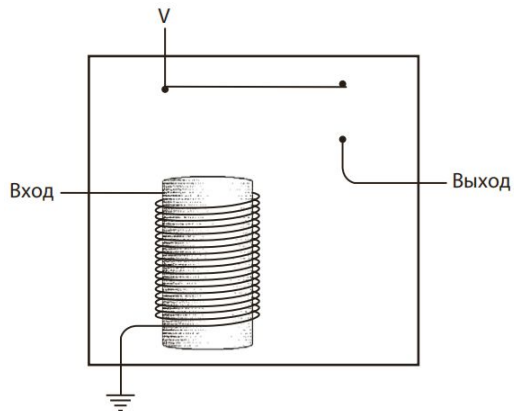


Ада Лавлейс (10 декабря 1815 – 27 ноября 1852) создала первую в мире программу для аналитической машины Беббиджа

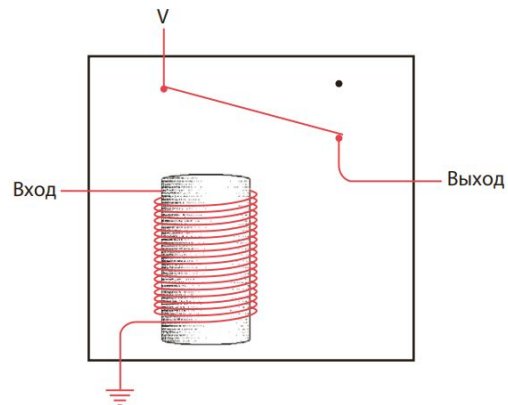




Реле. Схема



До подачи напряжения



После подачи
напряжения --
переключатель
притягивается

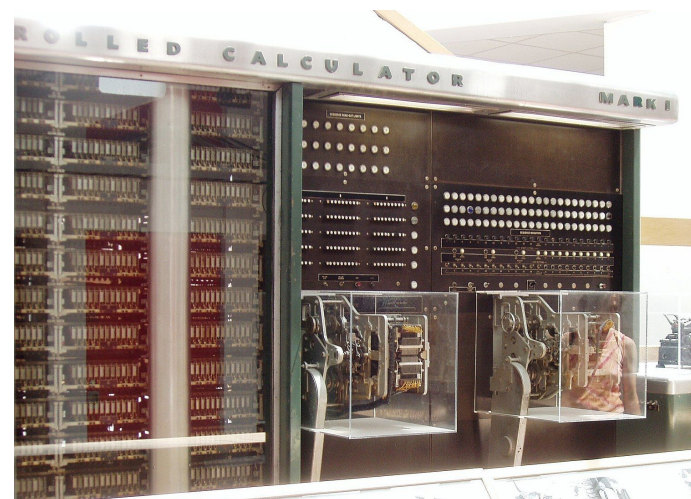


Марк I

1943 г., первый цифровой компьютер:

«Automated Sequence Controlled Calculator»,
позже получивший имя «Марк I».

Компьютер оперировал 72 числами,
состоящими из 23 десятичных разрядов,
делая по 3 операции сложения или вычитания
в секунду. Умножение выполнялось в течение
6 секунд, деление — 15,3 секунды.

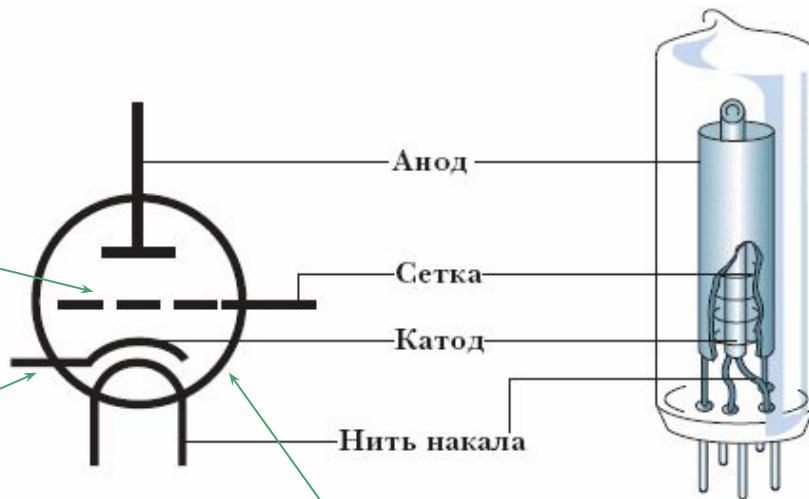




Радиолампа. Схема

На сетку подается напряжение, “облако” электронов катода дополнится электронами сетки и они начнут движение к аноду

На катод подается напряжение, и за счет нити накала катод нагревается и создается “облако” электронов



вакуумная колба



ENIAC

ЭНИАК (1945) - первый **электронный** цифровой вычислитель.

Построен на 18000 радиолампах.

Самый большой компьютер в истории (30т).

Домашние компьютеры обогнали его по быстродействию в 1977 г.

Вычисления производились в десятичной с.с.





EDVAC. Архитектура фон Неймана

Использовалась двоичная с.с., использовались условные переходы.

Время операции сложения - 864 микросекунды, умножения - 2900 микросекунд (2,9 миллисекунды).

Первый компьютер на базе архитектуры фон Неймана.





Архитектура Фон-Неймана

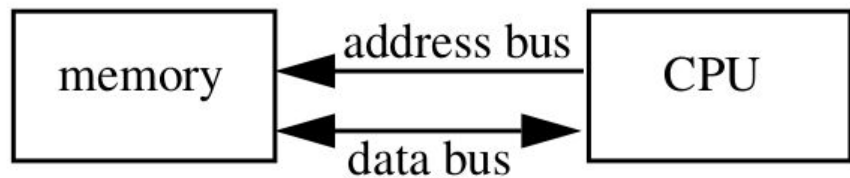
1. Адресность
2. Однородность памяти
3. Программное управление





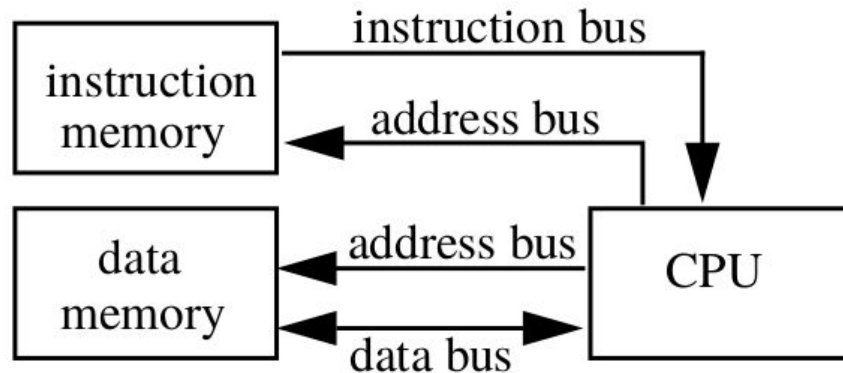
Типы архитектур

Von Neumann Architecture



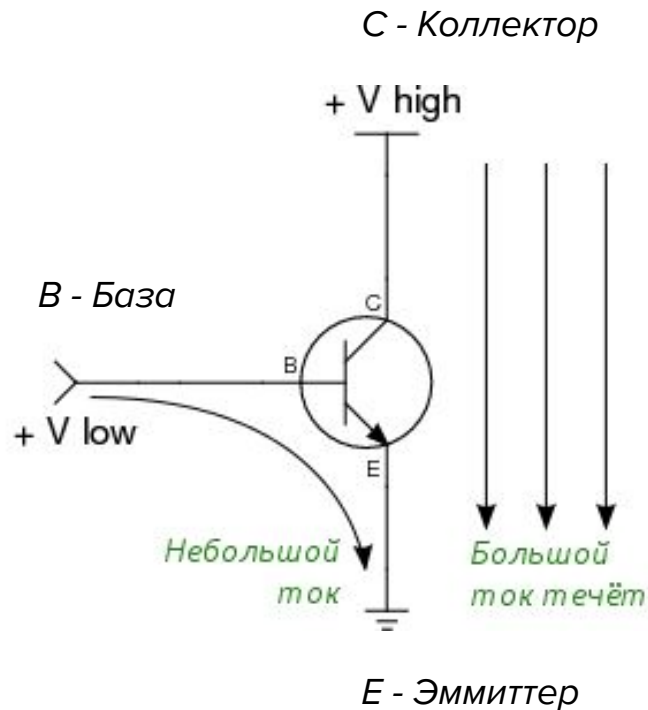
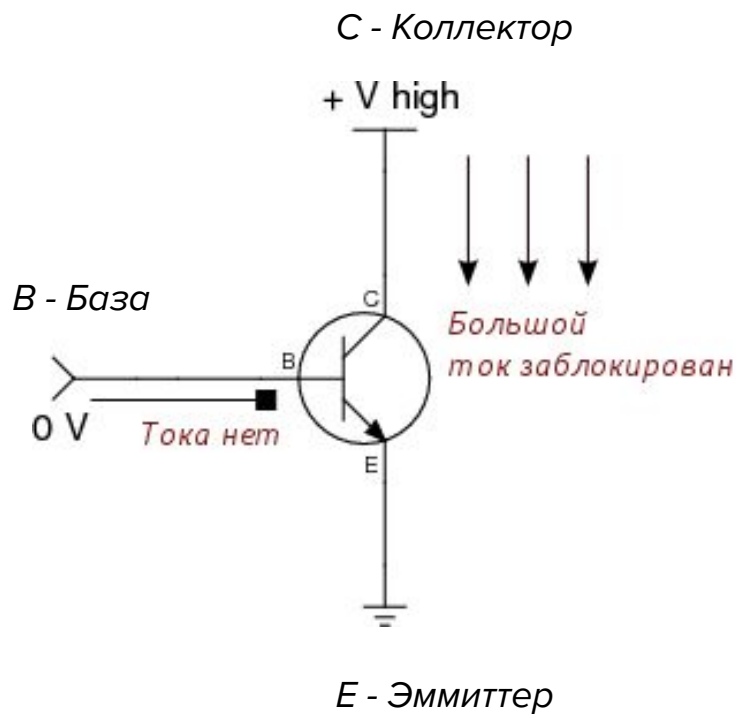
both instructions and
data go through here

Harvard Architecture





Транзистор



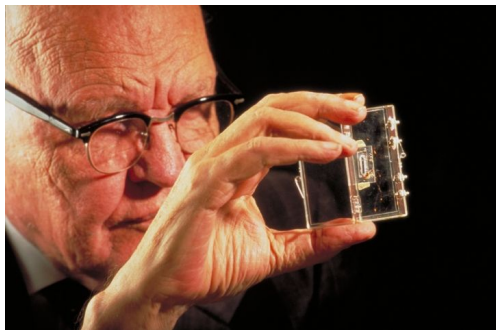


Переход на транзисторные компьютеры

Изготовление из единого куска кремния:

- июль 1958 г. Джек Килби
- январь 1959 Роберт Нойс

Гордон Мур, 1965 г. “Закон Мура”

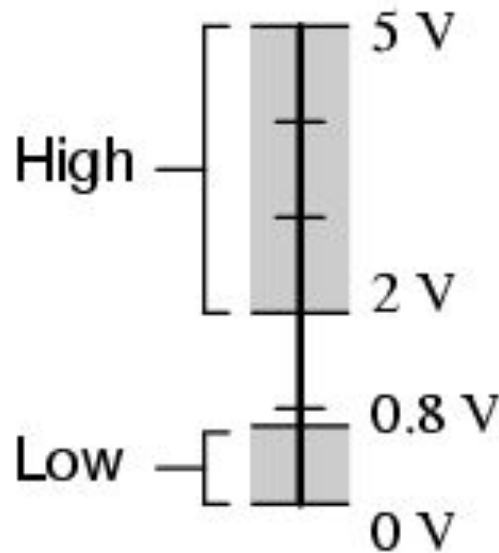




Связь цифрового и аналогового мира

Для микросхемы ТТЛ (транзисторно-транзисторной логики):

- 0 соответствует Low
- 1 соответствует High



Сумматор



Сумматор. Начало

- Используем переключатели, лампочки и логические вентили.
- Практически все операции в выч. машине представимы через сумму.
- Для двоичных чисел принцип тот же, что и для десятичных -- сложение в столбик
- Как представить “один в уме” в логической схеме?
- Сперва:

сумма	0	1
0	0	1
1	1	10

Допишем незначащие нули, что результат был двухбитовым



** занимал одинаковое кол-во ячеек*

сумма	0	1
0	00	01
1	01	10



Сумматор: сумма и перенос

- вынесем отдельно бит, который “в уме” (перенос разряда)

сумма	0	1
0	00	01
1	01	10

 \Rightarrow

перенос	0	1
0	0	0
1	0	1

 \cup

“сумма”	0	1
0	0	1
1	1	0

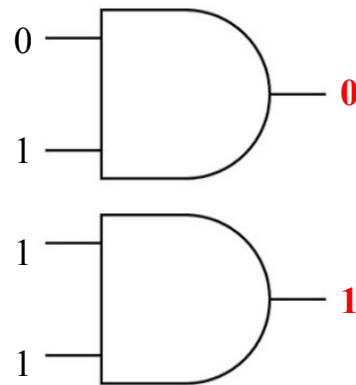
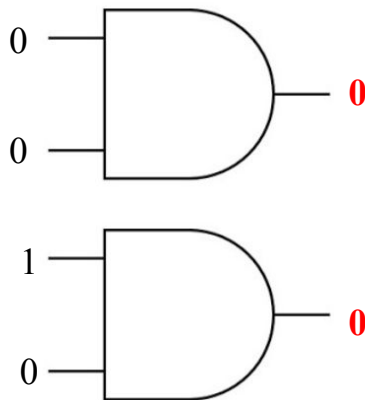
Сумматор \Rightarrow И, & AND \cup Искл. ИЛИ XOR



Сумматор. Вентиль “И” для переноса

- нарисуем работу вентиля по таблице истинности \Rightarrow нарисует, как работает “1 в уме” (перенос разряда) в логической схеме

перенос	0	1
0	0	0
1	0	1

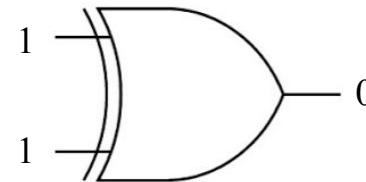
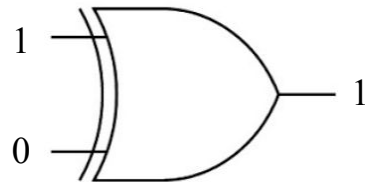
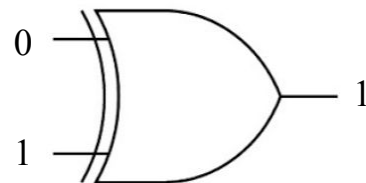
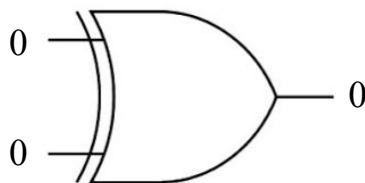




Сумматор. Искл. ИЛИ для суммы

- нарисуем работу вентиля по таблице истинности \Rightarrow нарисуем, как работает “сумма” в логической схеме

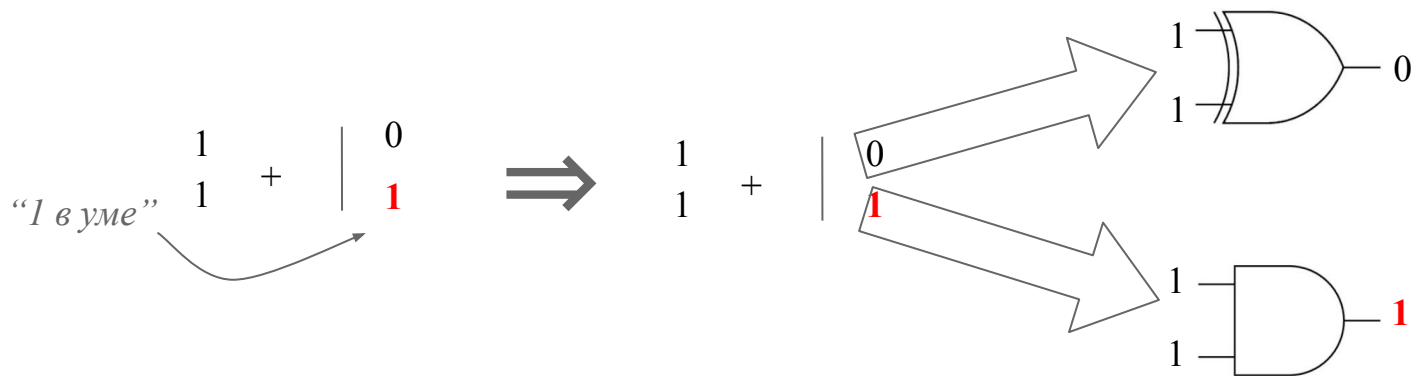
“сумма”	0	1
0	0	1
1	1	0





Сумматор. Как соединить вентили в схему?

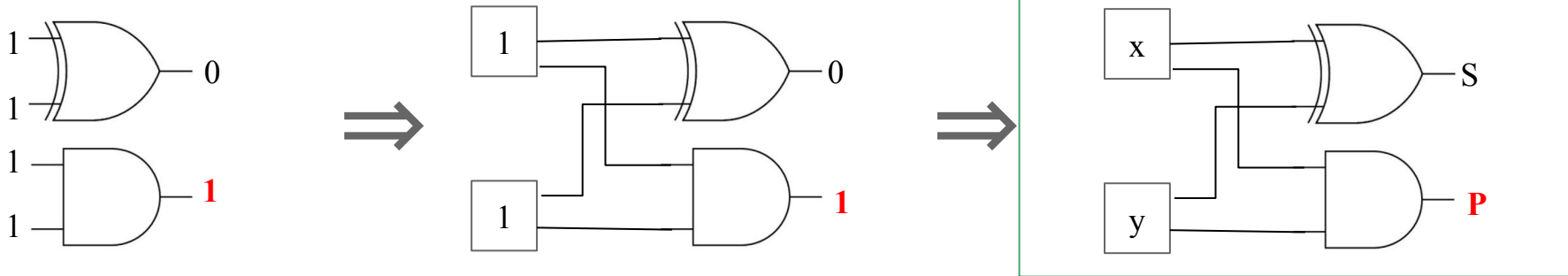
- одноразрядное сложение в двоичной системе
- “положим” сложение в столбик на бок





Полусумматор

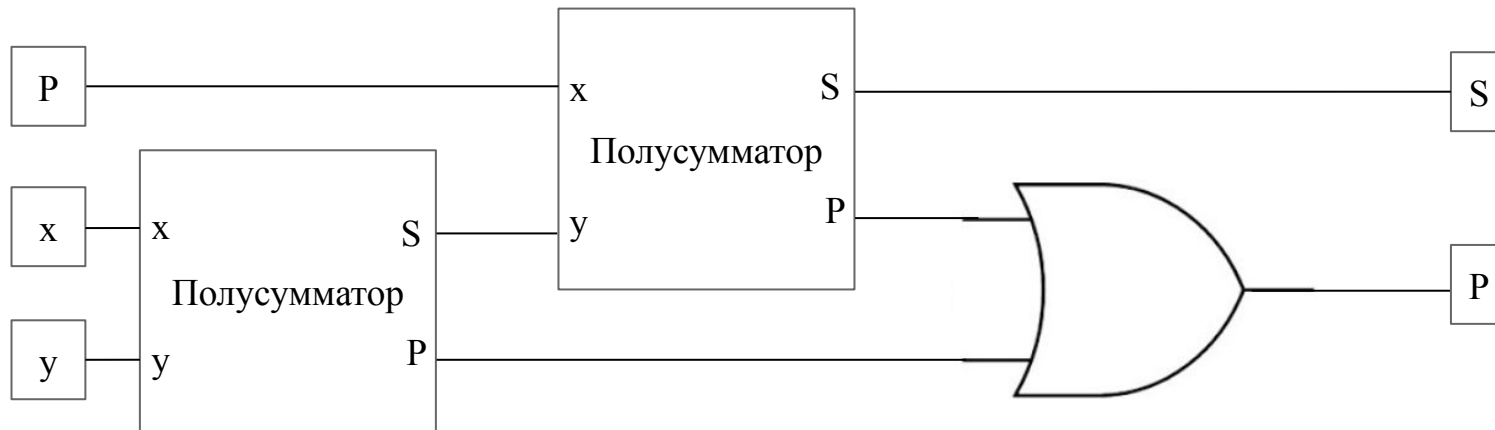
- избавляемся от дублирования входов
- прокладываем провода подлиннее
- введем обозначения:
 - входных сигналов x , y (одноразрядные)
 - выходных сигналов S (сумма), P (перенос)





Полный сумматор

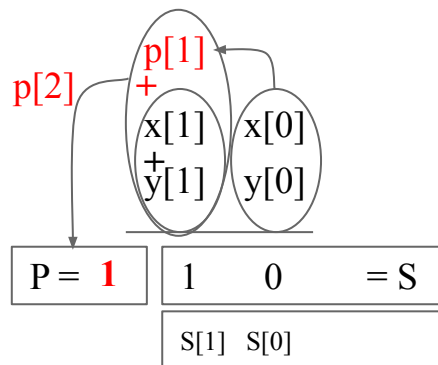
- для трех двоичных цифр (с учетом переноса P , поданного на вход)





$$+ \begin{array}{c} \begin{array}{cc} [1] & [0] \\ \hline \mathbf{x} = & 1 & 1 \\ \hline \mathbf{y} = & 1 & 1 \\ \hline \end{array} \\ \begin{array}{ccc} \mathbf{1} & 1 & 0 \end{array} \end{array}$$

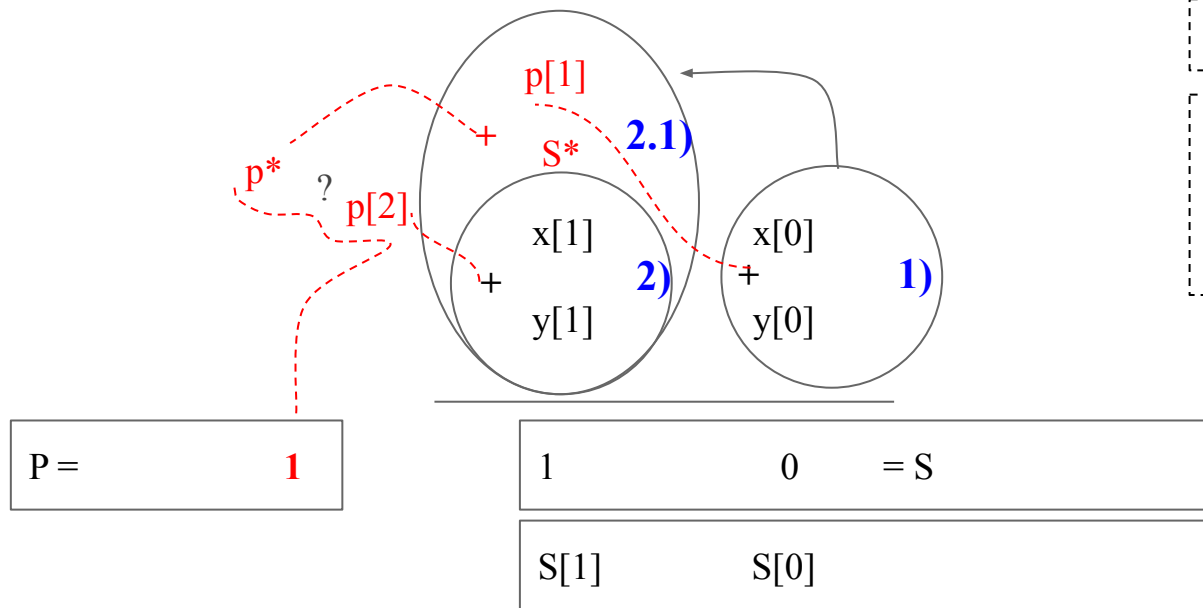
```
x[0] = 1
x[1] = 1
y[0] = 1
y[1] = 1
```



- дважды держали “1 в уме” \Rightarrow два переноса \Rightarrow два вентиля И
- два разряда в итоговой “сумме” \Rightarrow два вентиля Искл. ИЛИ:
 - $S[0] \Rightarrow$ получился естественным путем, породил первую единицу в уме $p[1]$
 - $S[1] \Rightarrow$ получился за счет сложения $x[1]+y[1]+p[1]$, что породило вторую единицу в уме **$p[2]$**
 - $x[1]+y[1] \Rightarrow$ известный нам полусумматор



Детализация



красным -- то, что держим в уме

каждый “+” в этой схеме генерирует два элемента в сумматоре:

- элемент “суммы” и
- элемент “перенос”

$$1) x[0] + y[0] = 1 \ 0 \Rightarrow S[0] = 0 \text{ и } p[1] = 1$$

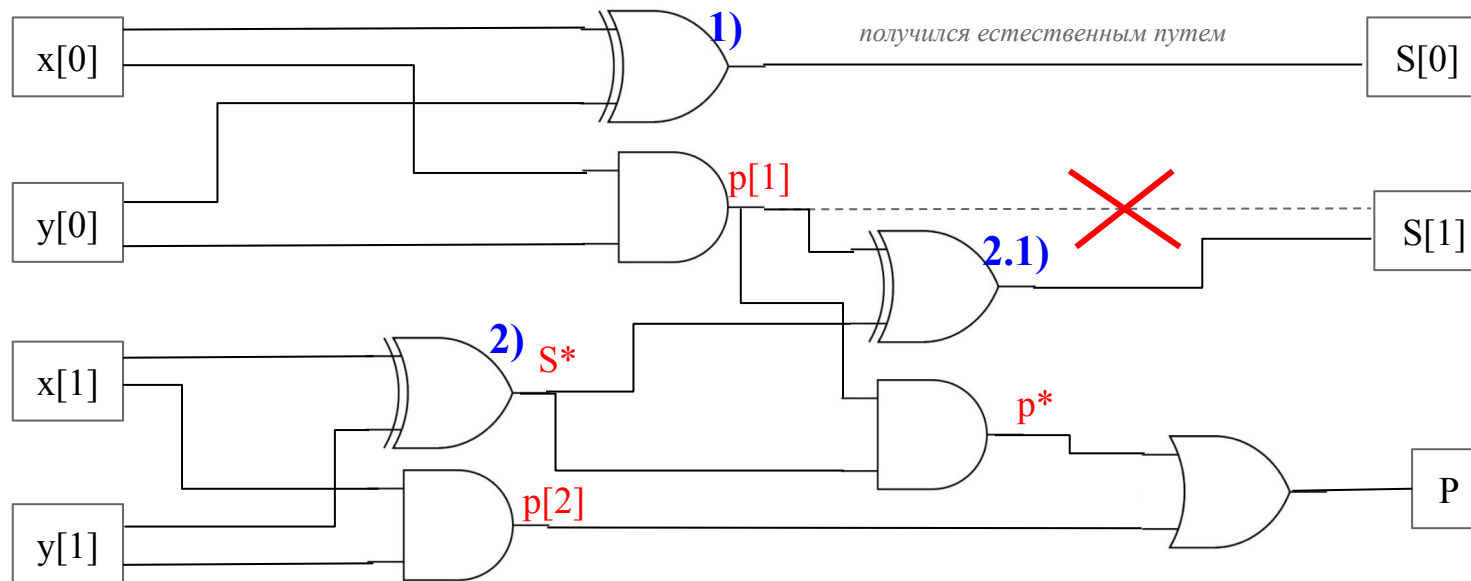
$$2) x[1] + y[1] = 1 \ 0 \Rightarrow S^* = 0 \text{ и } p[2] = 1$$

$$2.1) \text{ реализуем } p[1] : S^* + p[1] \Rightarrow S[1] = 1 \text{ и } p^* = 0$$

остались незадействованные p^* и $p[2] \Rightarrow$ ИЛИ (см. след. слайд)



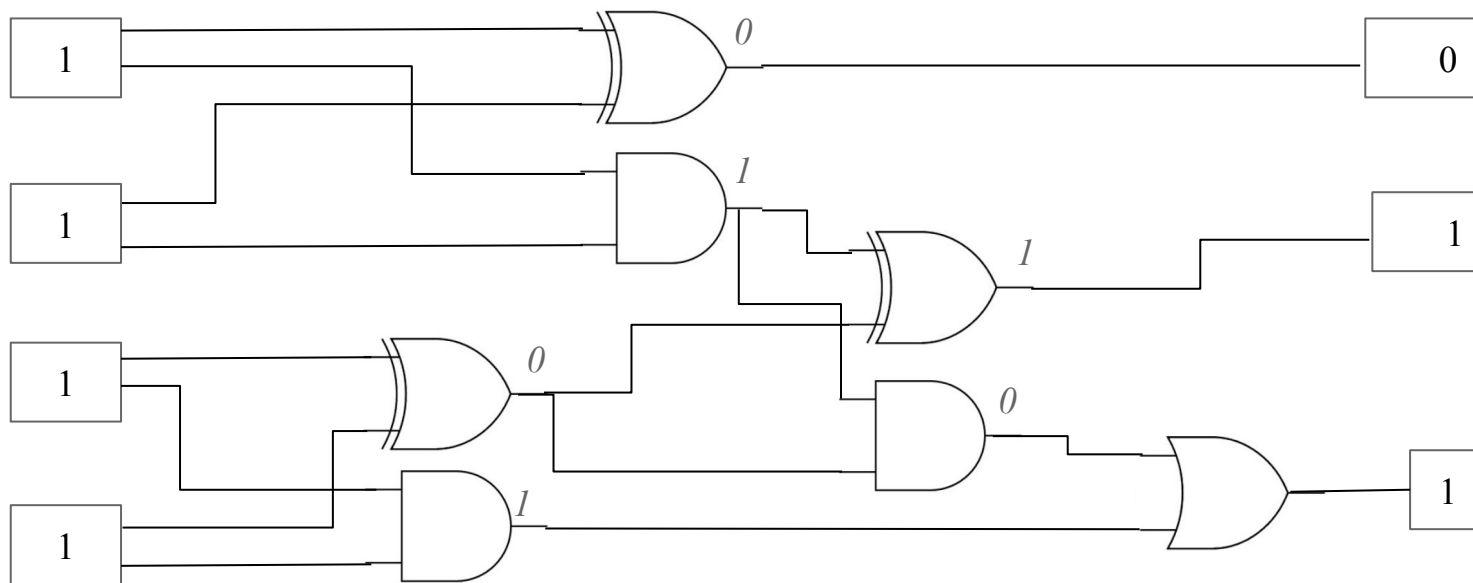
Сумматор. Схема двухразрядного сложения



три полусумматора в схеме -- найди их

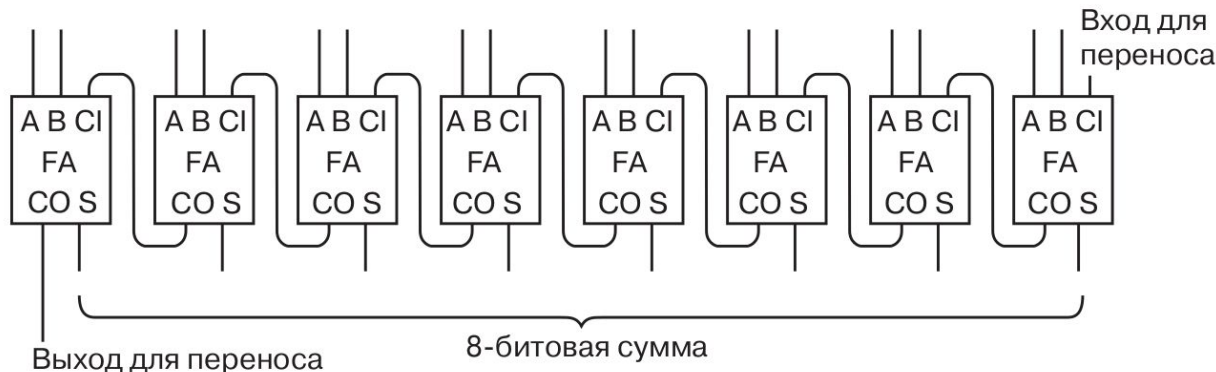


Сумматор. Пример



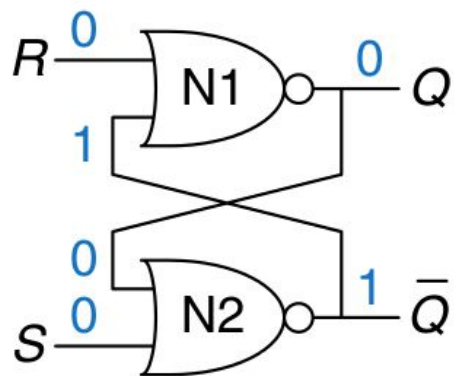


Сумматор. 8-битовая сумма

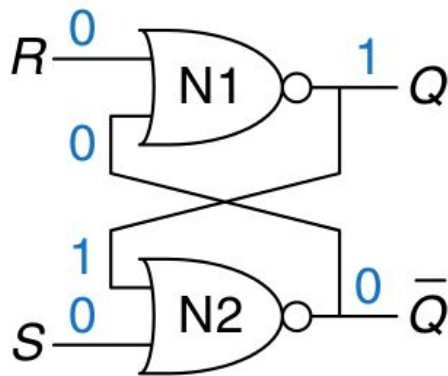


- Первый вход для переноса -- заземление.
- Обозначени:
 - CI / CO - Carry In / Carry Out (вход / выход для переноса)
 - FA -- Full Adder (HF -- Half Adder)
 - S -- Summ

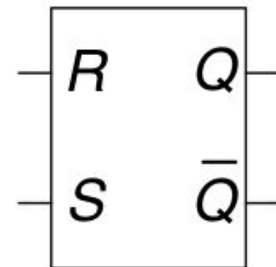
SR(RS)-Защелка



(a)



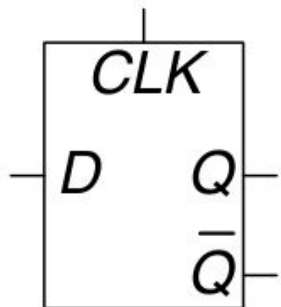
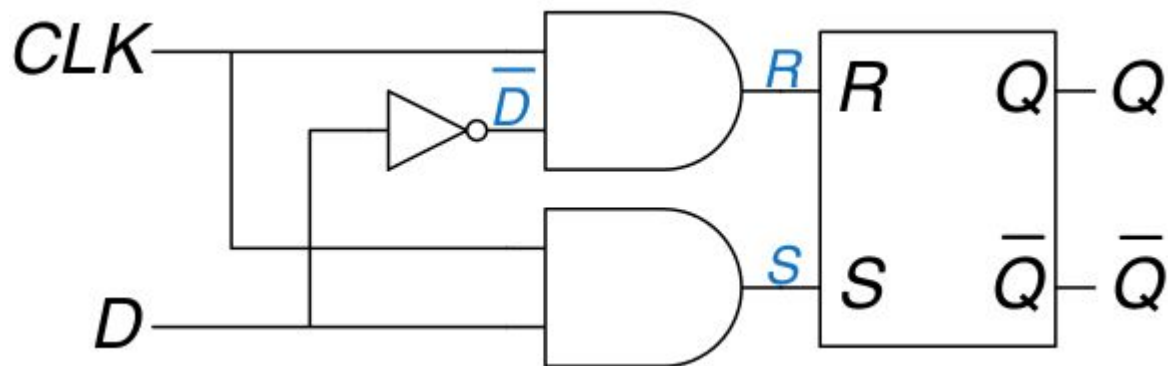
(b)



Case	S	R	Q	\bar{Q}
IV	0	0	Q_{prev}	\bar{Q}_{prev}
I	0	1	0	1
II	1	0	1	0

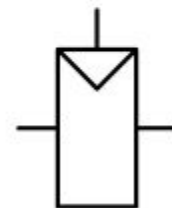
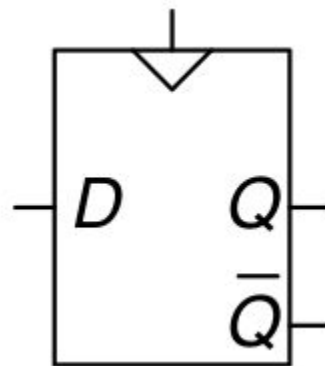
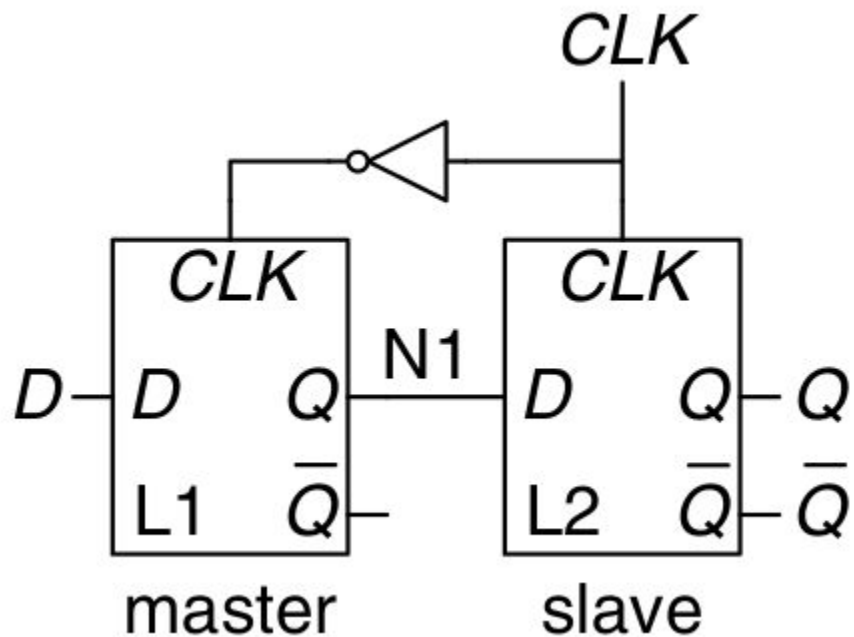


D-Защелка (D-Latch)



CLK	D	\overline{D}	S	R	Q	\overline{Q}
0	X	\overline{X}	0	0	Q_{prev}	\overline{Q}_{prev}
1	0	1	0	1	0	1
1	1	0	1	0	1	0

D-Триггер (D-flip-flop)





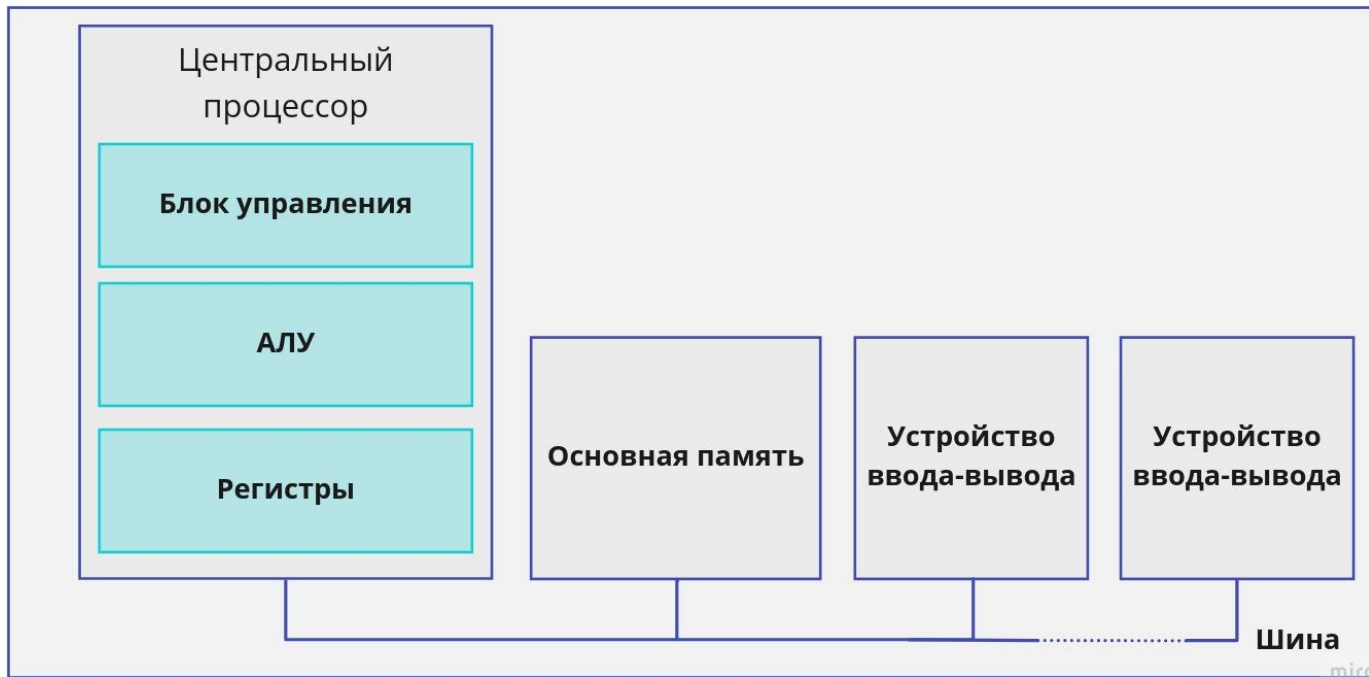
Регистр

Регистр – логическая схема, предназначенная для хранения двоичных чисел заданной разрядности.

Регистр состоит из группы триггеров (например, D-триггеров).



Как устроено простое вычислительное устройство





Источники

- Ч. Петцольд “Код”
- Э. Таненбаум “Архитектура Компьютера”
- <https://stepik.org/course/253> Курс на Stepik “Введение в Архитектуру ЭВМ”

Вопросы по курсу можно задавать:

Шевская Наталья Владимировна
natalya.shevskaya@moevm.info