

# Логические основы

# устройства компьютера

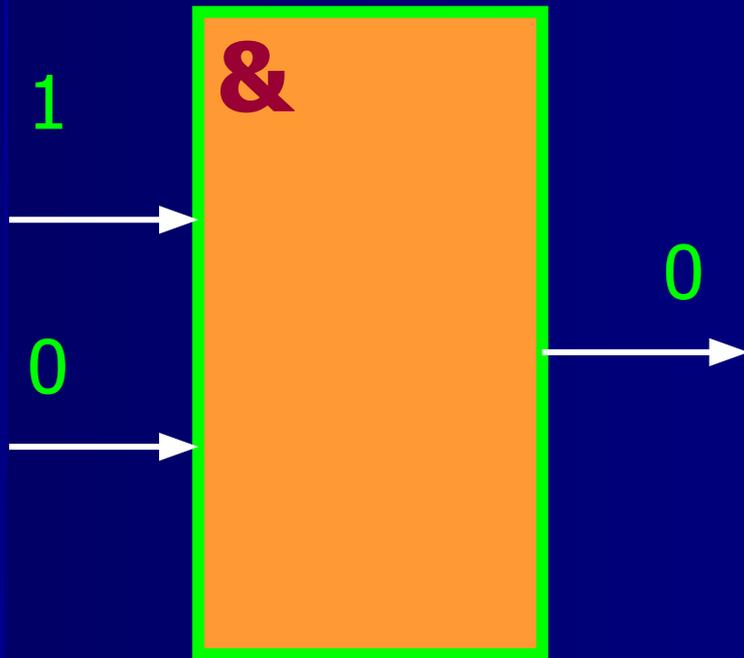
Логические операции «И», «ИЛИ», «НЕ»  
лежат в основе работы преобразователей  
информации любого компьютера



**Клод Шеннон**  
(1916 г.)

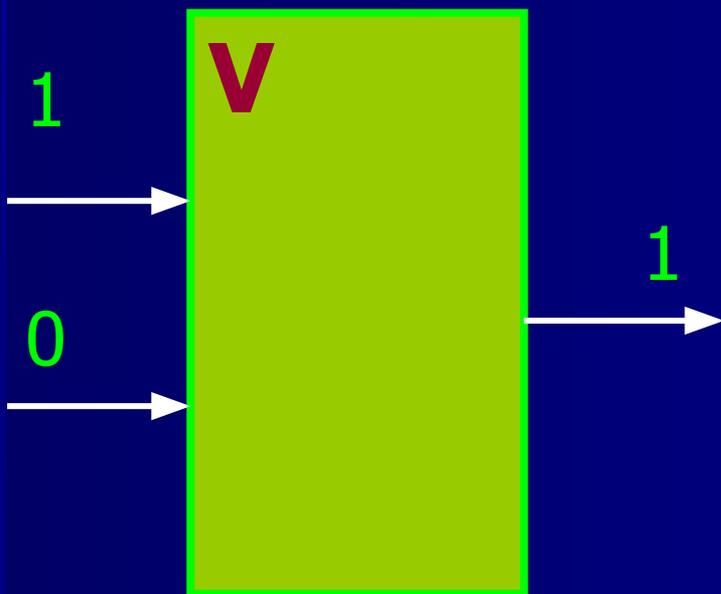
американский математик,  
доказал применимость  
булевой алгебры в теории  
контактных и релейно-  
контактных схем (в 1938  
году)

# КОНЬЮНКТОР



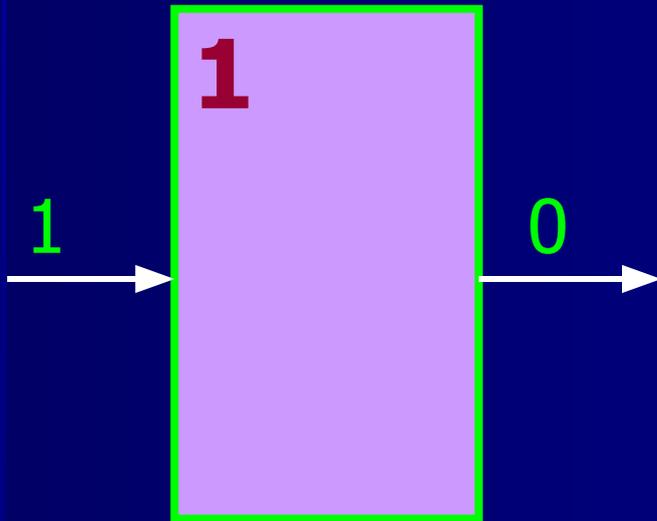
Логический элемент «И», преобразует входные сигналы и выдает результат логического умножения

# Дизъюнктор

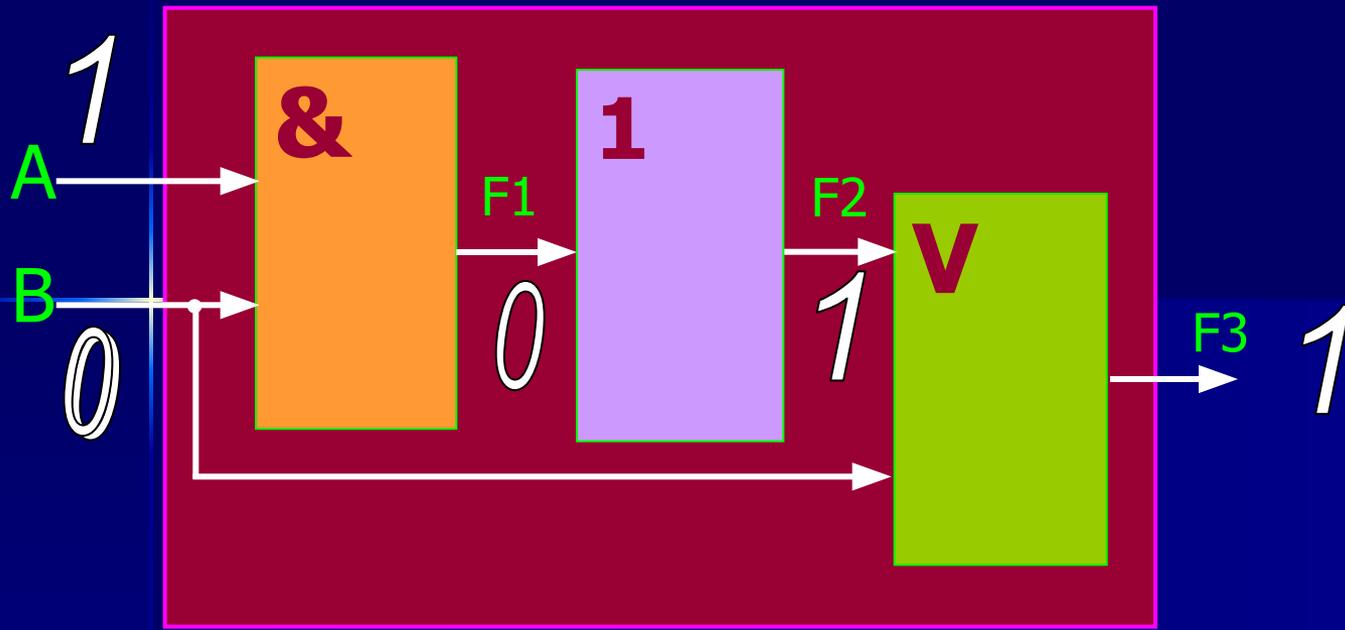


- Логический элемент «**или**», преобразует входные сигналы и выдает результат логического сложения.

# Инвертор



Логический элемент «**НЕ**».  
Преобразует входной сигнал и выдает результат логического отрицания.



$$\overline{A \& B} \vee B$$

Структурная формула ЛУ

Функциональная схема логического устройства

Зная функциональную схему, можно составить структурную формулу данного ЛУ.

Анализируя структурную формулу, можно создать функциональную схему и понять, как работает данное ЛУ.

# КОНТРОЛЬНЫЕ ВОПРОСЫ

- Какие логические операции лежат в основе преобразователей информации в ПК?
- Как называются логические элементы ПК?
- Что такое структурная формула?
- Что можно увидеть на функциональной схеме?
- Какие устройства ПК построены на логических элементах?
- Какие основные операции выполняет центральный процессор?
- Как «работает» память ПК?

Не знаете?  
тогда идем  
дальше!

# Логические устройства ПК

Так как все многообразие операций в ПК сводится к сложению двоичных чисел, то главной частью процессора (АЛУ) является сумматор.

Рассмотрим сложение одноразрядных двоичных чисел:

Слагаемые		Перенос	Сумма
A	B	P	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Слагаемые		Перенос	Сумма
A	B	P	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$P = A \& B$$

$$S = (A \vee B) \& (\overline{A \& B})$$

1
4
2

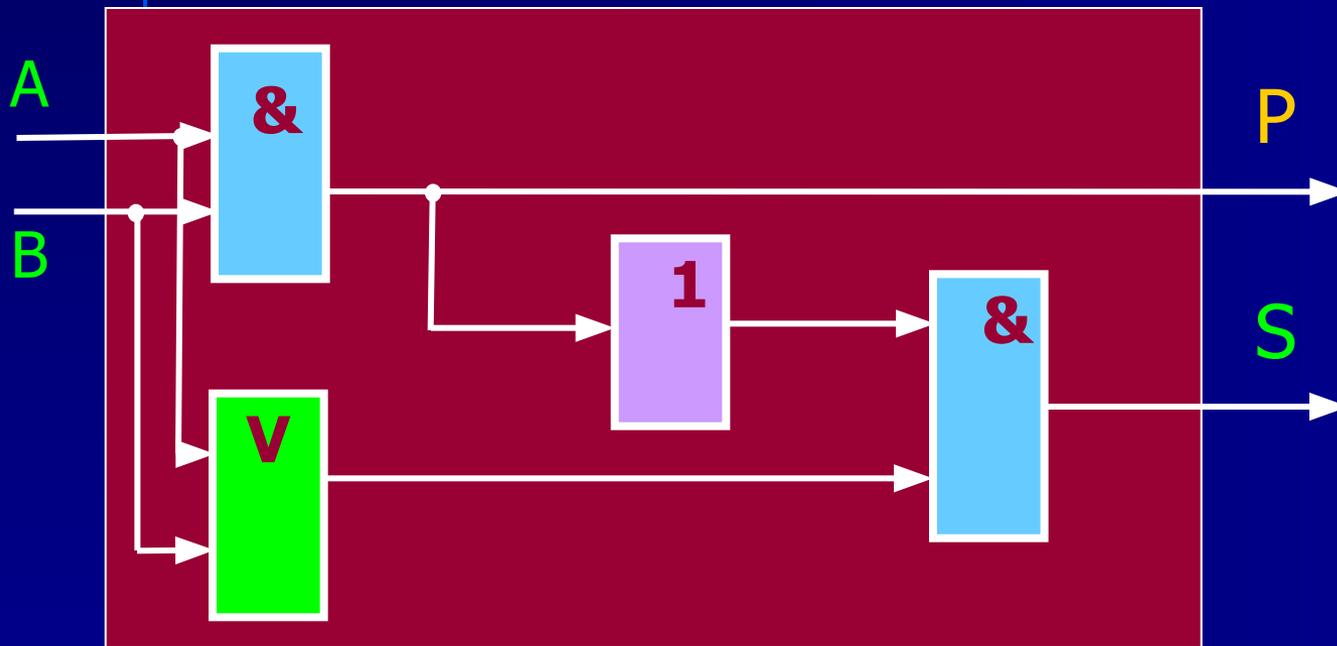
Докажем это, построив таблицу истинности для данного ЛВ

A	B	1	2	3	4
		$A \vee B$	$A \& B$	$\text{NOT}(2)$	$1 \& 3$
0	0	0	0	1	0
0	1	1	0	1	1
1	0	1	0	1	1
1	1	1	1	0	0

$$P = A \& B$$

$$S = (A \vee B) \& (\overline{A \& B})$$

Теперь, на основе полученных логических выражений, можно построить схему данного устройства



Данная схема называется полусумматором, так как суммирует одноразрядные двоичные числа без учета переноса из младшего разряда.

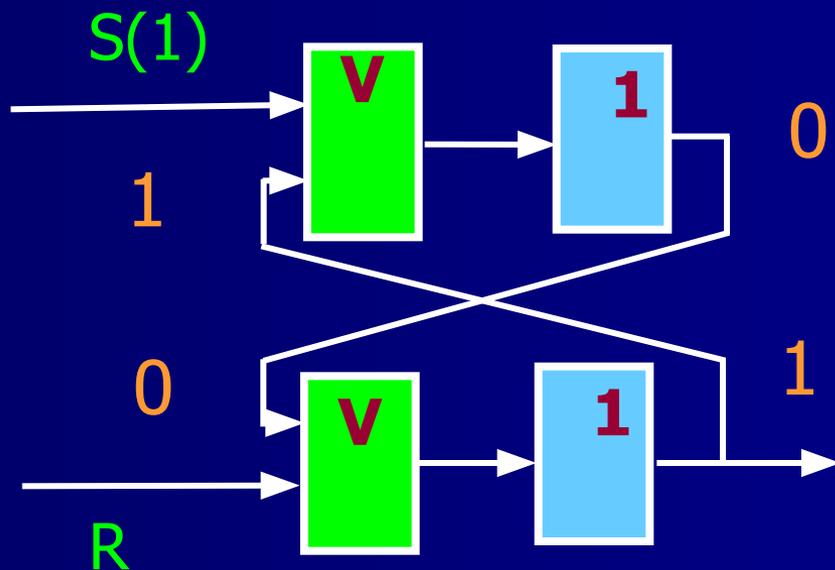
Многоразрядный сумматор процессора состоит из полных одноразрядных сумматоров, причем выход (перенос) сумматора младшего разряда подключен ко входу сумматора старшего разряда.

Слагаемые		Перенос из младшего разряда	Перенос в старший разряд	Сумма
A	B	$P_0$	P	S
0	0	0	0	0
0	1	0	0	1
1	0	0	0	1
1	1	0	1	0
0	0	1	0	1
0	1	1	1	0
1	0	1	1	0
1	1	1	1	1

$$P = (A \& B) \vee (A \& P_0) \vee (B \& P_0)$$

$$S = (A \vee B \vee P_0) \& (-P_0) \vee (A \& B \& P_0)$$

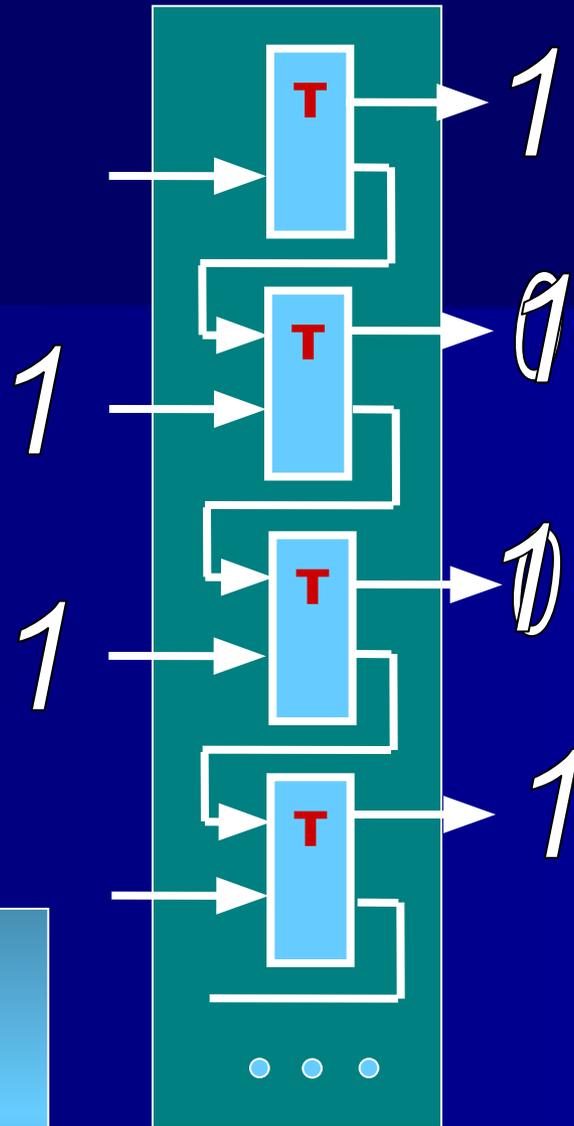
Для хранения информации в ОП и регистрах ЦП применяется устройство **ТРИГГЕР**. Ячейка памяти состоит из 8, 16 или 32 триггеров, что и определяет **разрядность ЦП**. Триггер строится из двух элементов «ИЛИ» и двух элементов «НЕ».



В обычном состоянии на входы подан «0». Для записи на вход S подается «1». Он его будет хранить и даже после того, как сигнал на входе «S» исчезнет. Чтобы сбросить информацию, подается «1» на вход R (Reset), после чего триггер возвращается к исходному «нулевому» состоянию.

- Несколько триггеров можно объединить в группы - **регистры** И использовать в качестве запоминающих устройств (ЗУ).
- Если в **регистр** входит **N триггеров**, то при таком ЗУ можно запоминать N-разрядные двоичные слова.
- ОЗУ ЭВМ часто конструируется в виде набора регистров.
- **Один регистр** образует **одну** ячейку памяти, каждая из которых имеет свой номер

Таким образом, ЭВМ состоит из огромного числа отдельных логических элементов, образующих все ее узлы и память.



# Практическая работа

Используя панель Рисования редактора MS Word, создайте:

1. Схемы логических элементов

2. Схему логического устройства  $\overline{A \& B} \vee B$

3. Схему полусумматора по формулам:

$$P = A \& B$$

$$S = (A \vee B) \& (\overline{A \& B})$$

4\*. Схемы переноса  $P$  и суммы  $S$  многоразрядного сумматора

$$P = (A \& B) \vee (A \& P_0) \vee (B \& P_0)$$

$$S = (A \vee B \vee P_0) \& (\neg P_0) \vee (A \& B \& P_0)$$

Автор презентации является участником  
конкурса компьютерных презентаций  
проводимого на сайте  
«Информатика в школе»  
[www.inf777.narod.ru](http://www.inf777.narod.ru)  
при спонсорстве издательского дома  
«Питер»