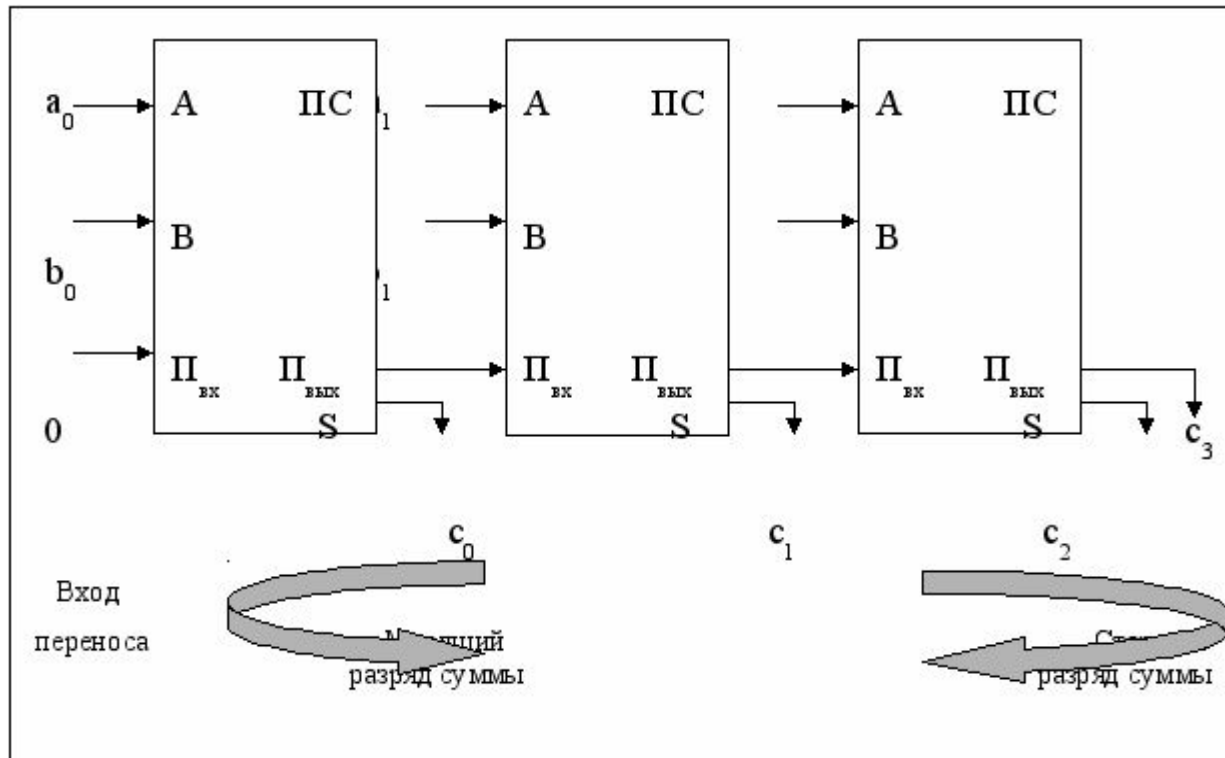


# Сумматор двоичных чисел



Результат  
 $C = (c_3 c_2 c_1 c_0)$

Рис. 9

# Сумматор двоичных чисел

И в двоичной системе счисления, и в алгебре логики информация представлена в виде двоичных кодов (0,1).

Для того, чтобы максимально упростить работу компьютера, все математические операции сводятся к операции сложения. Поэтому главной частью процессора является **сумматор**, который как раз и обеспечивает такое сложение.

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



Перенос в старший  
разряд

Перенос можно реализовать с помощью

операции логического умножения:  $P = A \& B$

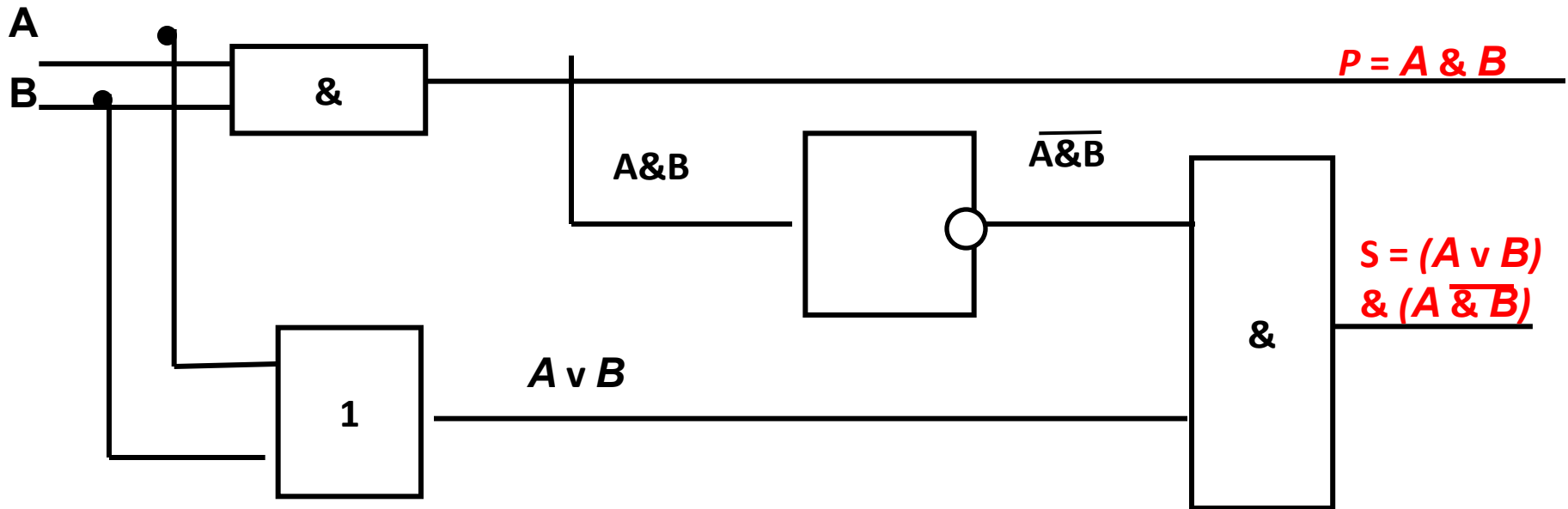
**B**

Для определения суммы можно применить следующее логическое выражение:

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

A	B	P	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

схема полусумматора

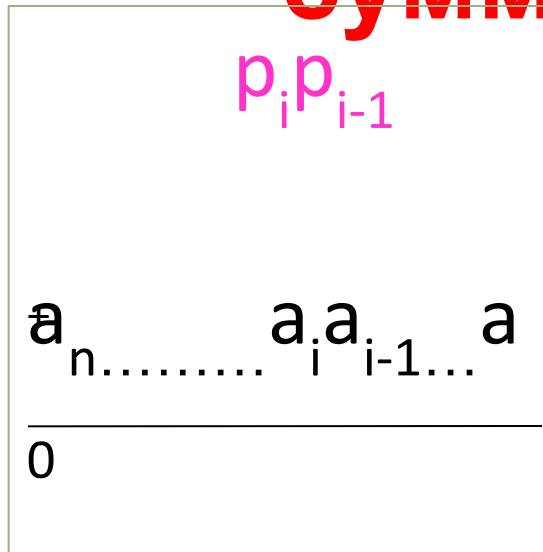


Данная схема называется  
**полусумматором**, так как реализует  
суммирование одноразрядных  
двоичных чисел без учёта переноса из  
младшего разряда

# ПОЛНЫЙ ОДНОРАЗРЯДНЫЙ сумматор

```

11111
 10101
+  11111
-----
110100
    
```



$b_n, \dots, b_i, b_{i-1}, \dots$

Таким образом, полный одноразрядный сумматор должен иметь три входа:  $A, B$  – слагаемые и  $P_0$  – перенос из младшего разряда и два выхода: сумму  $S$  и перенос  $P$ .

слагаемые		Перенос с из младшего разряда	Перенос	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

Из таблицы сложения видно, что перенос (P) принимает значение 1 тогда, когда хотя бы две логические переменные одновременно принимают значение 1.

Т.о., перенос реализуется путём логического сложения результатов попарного логического умножения входных переменных A, B, P<sub>0</sub>.

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

Для получения суммы (S) необходимо результат логического сложения входных переменных A, B, P<sub>0</sub> умножить на инвертированный перенос P.

$$S = (A + B + P_0) \& \bar{P}$$

слагаемые		Перенос из младшего разряда	Перенос	Сумма
A	B	P <sub>0</sub>	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

слагаемые		Перенос из младшего разряда	Перенос	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

Данное логическое выражение даёт правильные значения суммы во всех случаях, кроме одного, когда все входные логические переменные принимают значение 1.

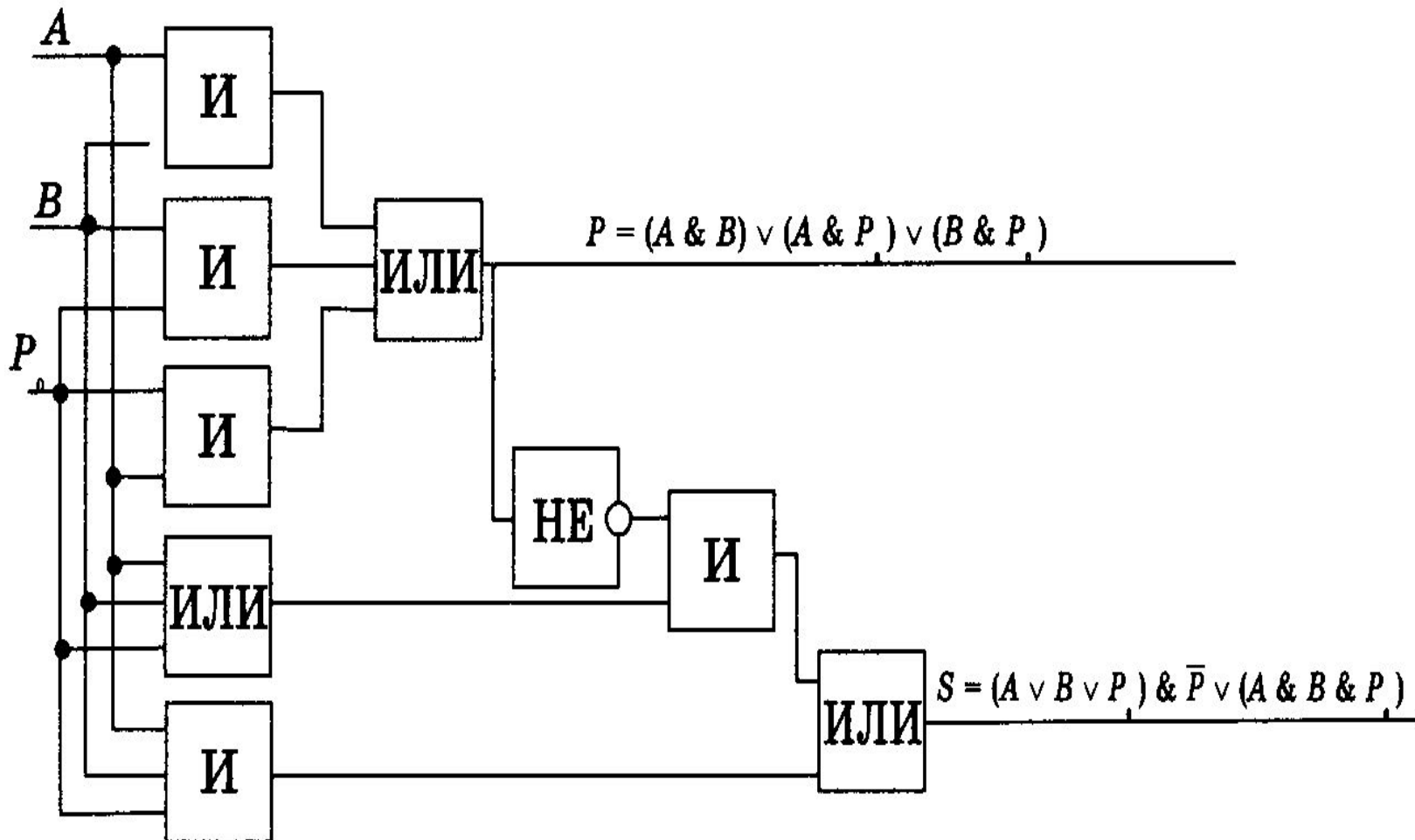
$$P = (1 \& 1) \vee (\overline{1 \& 1}) \vee (1 \& 1) = 1;$$

$$S = (1 \vee 1 \vee \overline{1}) \& P = 1 \& 0 = 0.$$

Для получения правильного значения суммы ( $S=1$ ) необходимо использовать следующую формулу:

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

# Теперь можно построить схему одноразрядного сумматора





# Многоразрядный сумматор

Но процессор, как правило, складывает многоразрядные двоичные числа.

Для того, чтобы вычислить сумму  $n$  – разрядных двоичных чисел, необходимо использовать **многоразрядный сумматор**, в котором на каждый разряд ставится одnorазрядный сумматор и выход – перенос сумматора младшего разряда – подключается к входу сумматора старшего разряда.

