



# **VI. ПРОЕКТИРОВАНИЕ КОМБИНАЦИОННЫХ ЛОГИЧЕСКИХ СХЕМ**

## ***ТЕМА 6.4. СУММАТОРЫ***



## СУММАТОРЫ

Сумматоры выполняют операцию сложения двух чисел и операцию вычитания, которая заключается в суммировании вычитаемого с дополнительным кодом вычитателя.

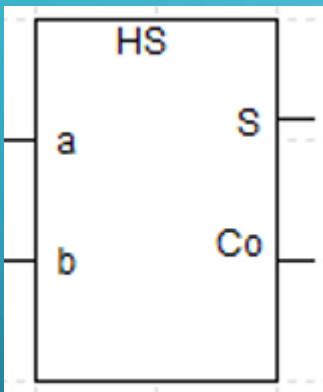
Структура всех сумматоров основывается на двух логических схем суммирования, известных как:

полусумматор (**Half sum**, обозначенная HS) и

полный сумматор (**Full sum**, обозначенная SM)

# ПОЛУСУММАТОР

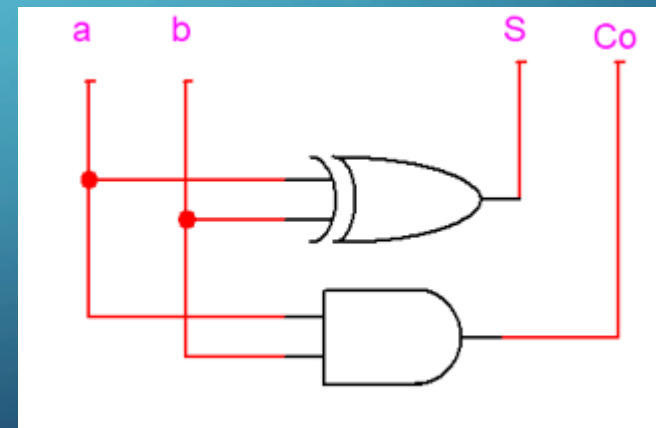
Полусумматор - это КЛС, который складывает 2 бита без учета переноса из предыдущего разряда. На выходе формируется Сумма (S) и Перенос в следующий разряд (Co).



a	b	S	Co
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

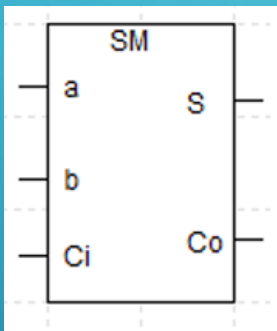
$$S = a \oplus b$$

$$C_0 = ab$$



# ПОЛНЫЙ СУММАТОР

**Полный сумматор** - это КЛС, который складывает 2 бита и перенос  $C_i$  из предыдущего разряда. На выходе полный сумматор формирует Сумму ( $S$ ) и Перенос в следующий разряд ( $C_o$ ).



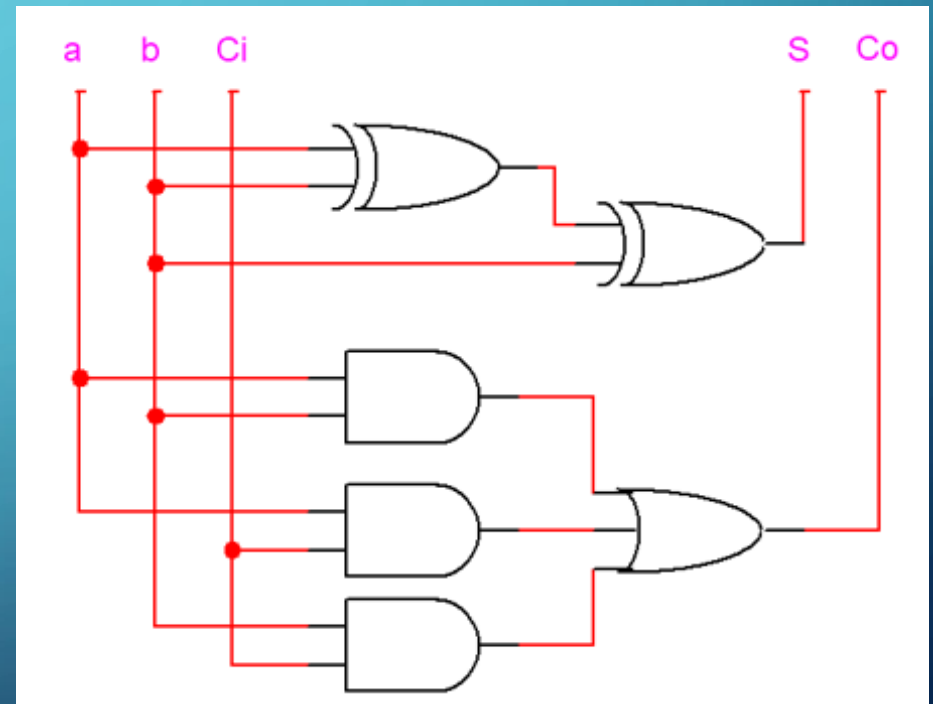
a	b	$C_i$	S	$C_o$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

x3 \ x1x2	00	01	11	10
0		1		1
1	1		1	

x3 \ x1x2	00	01	11	10
0			1	
1		1	1	1

$$S = a \oplus b \oplus C_i$$

$$C_o = ab + aC_i + bC_i$$

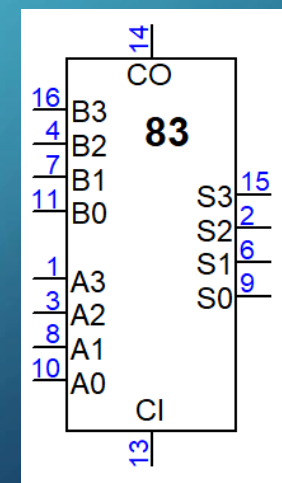
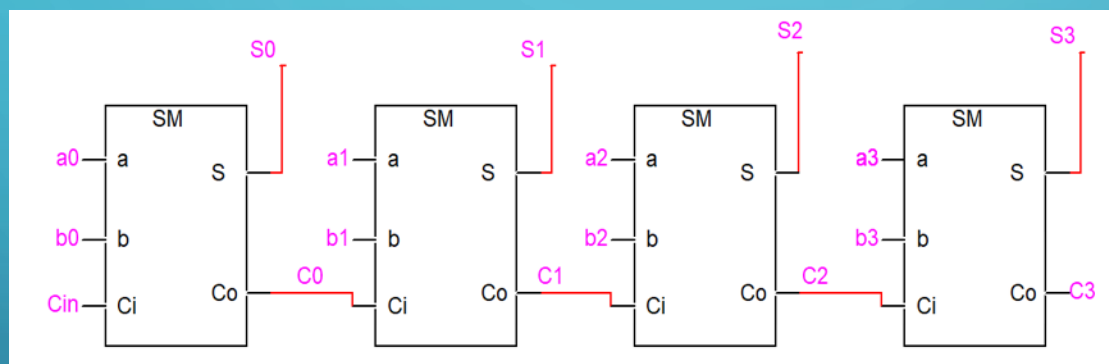


# ПОСЛЕДОВАТЕЛЬНЫЙ СУММАТОР

Сумматоры для сложения  $n$ -разрядных двоичных слов реализуются путем соединения 1-разрядных сумматоров.

В этой схеме сложение выполняется параллельно, и распространение переноса происходит последовательно, что приводит к суммированию времени задержки, так что конечный результат обеспечивается с довольно большой задержкой по сравнению с входным сигналом.

$$A = a_3 a_2 a_1 a_0$$
$$B = b_3 b_2 b_1 b_0$$



**Преимущество** последовательного сумматора - невысокая стоимость.

**Недостатком** является то, что время задержки увеличивается пропорционально количеству разрядов.

# СУММАТОР С УСКОРЕННЫМ ПЕРЕНОСОМ

Решение в этом случае является архитектурным и предполагает отказ от последовательного переноса в пользу параллельного (ускоренного). Фактически это означает переопределение логической функции, которая формирует сигнал переноса.

$$c_{i+1} = a_i b_i \vee a_i c_i \vee b_i c_i = a_i b_i \vee (a_i \vee b_i) c_i.$$

Определим 2 функции: G – Функция генерации переноса

$$G_i = a_i b_i$$

P – функция передачи переноса

$$P_i = a_i \vee b_i$$

В этом случае, формулу для переноса можно записать следующим образом:

$$c_{i+1} = G_i \vee P_i c_i$$

$$c_1 = G_0 \vee P_0 c_{IN};$$

$$c_2 = G_1 \vee P_1 c_1 = G_1 \vee P_1 (G_0 \vee P_0 c_{IN}) = G_1 \vee P_1 G_0 \vee P_1 P_0 c_{IN};$$

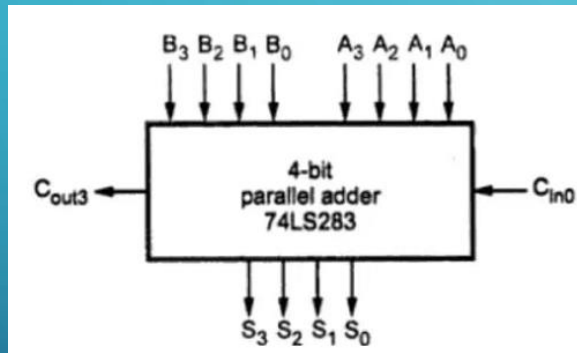
$$c_3 = G_2 \vee P_2 c_2 = G_2 \vee P_2 (G_1 \vee P_1 G_0 \vee P_1 P_0 c_{IN}) = G_2 \vee P_2 G_1 \vee P_2 P_1 G_0 \vee P_2 P_1 P_0 c_{IN}$$

## Схема 4х разрядного сумматора с ускоренным переносом:

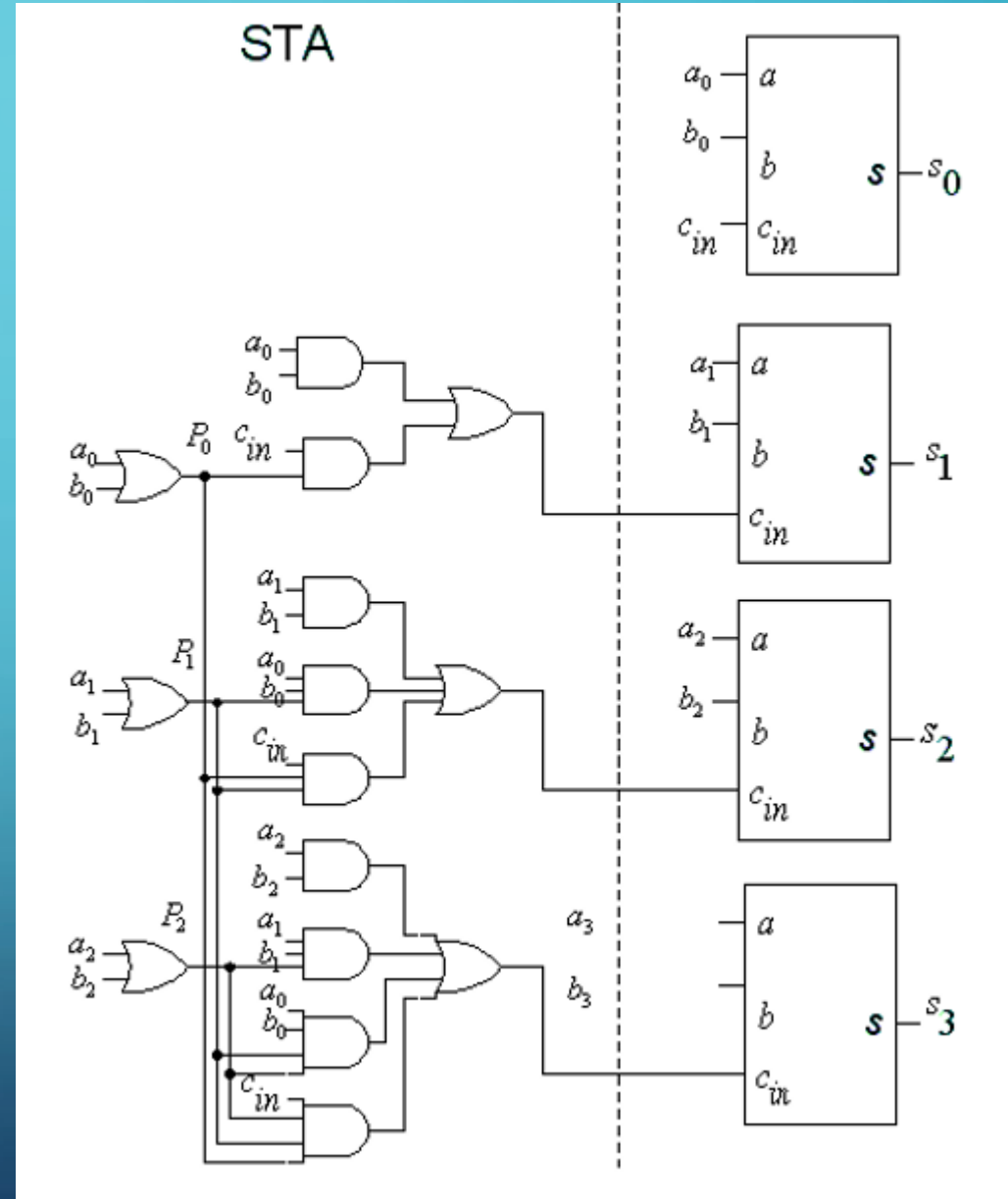
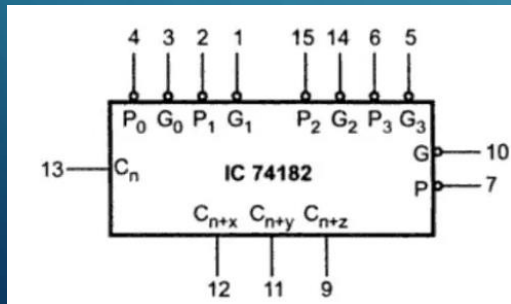
В состав сумматора с ускоренным переносом входят два модуля:

- **модуль суммирования**, состоящий из нескольких одноразрядных сумматоров
- **модуль для формирования ускоренного переноса**.

Интегральная схема 4-битного сумматора с ускоренным переносом 74\_283:



Микросхема генерации переноса 74\_182:



### **Преимущества** сумматора с ускоренным переносом

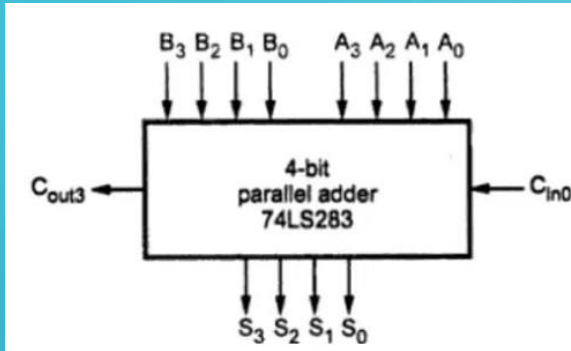
1. Перенос генерируется для каждого сумматора одновременно с использованием упрощенных уравнений, включающих  $P_i$ ,  $G_i$  и  $C_{in}$ .
2. Эта схема уменьшает задержку распространения. Это связано с тем, что выходной перенос зависит только от первого сигнала переноса, подаваемого на вход.
3. Это самый быстрый сумматор по сравнению с другими типами сумматоров.

### **Недостатки**

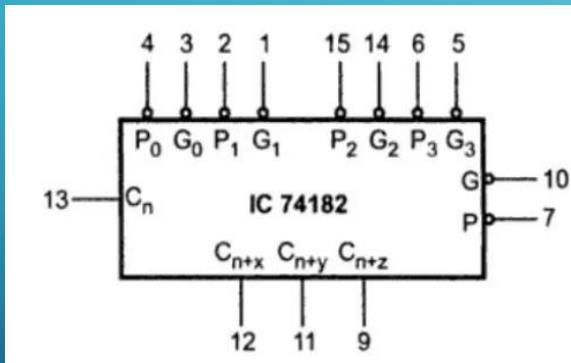
1. Схема сумматора с ускоренным переносом усложняется с увеличением числа разрядов.
2. Схема сумматора с ускоренным переносом стоит дорого, поскольку требует большего количества оборудования.
3. По мере увеличения числа переменных интегральная схема содержит больше логических вентилей, поэтому площадь этой схемы увеличивается.



Интегральная схема 4-битного сумматора с ускоренным переносом 74\_283:



Микросхема генерации переноса 74\_182:



$$C_{n+x} = G_0 + P_0 \cdot C_n$$

$$C_{n+y} = G_1 + P_1 \cdot (G_0 + P_0 \cdot C_n)$$

$$C_{n+z} = G_2 + P_2 \cdot (G_1 + P_1 \cdot (G_0 + P_0 \cdot C_n))$$

$$G' = (G_3 + P_3 \cdot (G_2 + P_3 \cdot (P_2 \cdot (G_1 + P_3 \cdot (P_2 \cdot (P_1 \cdot (G_0))')))))'$$

$$P' = (P_3 \cdot P_2 \cdot P_1 \cdot P_0)'$$

