**Оптимизация сумматора для получения дополнительного кода**

Логические формулы ячейки суммирования двух битов Xi и Yi с переносом Ci имеют вид:

Ri = (Xi **xor** Yi) **xor** Ci  
C(i+1) = (Xi **and** Yi) **or** ((Xi **xor** Yi) **and** Ci)

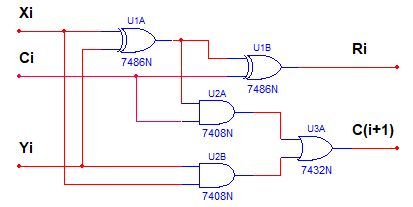
где:

i – индекс разряда

Ri – результат суммирования битов

C(i+1) – перенос в следующий разряд

Реализация такой ячейки имеет вид:



Для получения дополнительного кода, один из операндов сумматора, например Yi должен равняться нулю во всех ячейках. Если Yi =0, то:

Ri = (Xi **xor** 0) **xor** Ci  
C(i+1) = (Xi **and** 0) **or** ((Xi **xor** 0) **and** Ci)

или

Ri = (Xi **xor** 0) **xor** Ci  
C(i+1) = 0 **or** ((Xi **xor** 0) **and** Ci)

Следовательно, таблица истинности для операции **xor**, должна быть модифицирована, то есть, в ней отсекается столбец для которого Yi =0

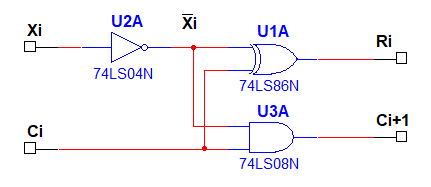
|  |  |  |
| --- | --- | --- |
| X/Y | 0 | 1 |
| 0 | 0 | 1 |
| 1 | 1 | 0 |

В этом случае логические формулы упрощаются и имеют вид:

Ri = Xi **xor** Ci  
C(i+1) = Xi  **and** Ci

**Синтез ячейки для получения дополнительного кода**

Кроме оптимизированной ячейки сумматора, ячейка преобразователя в дополнительный код должна вначале инвертировать входной бит, а только потом выполнить его сложение с нулем. Реализация такой ячейки будет иметь вид:



Для создания преобразователя в дополнительный код осталось только вспомнить, что перенос в самую младшую ячейку должен быть равен единице, (поскольку дополнительный код это инверсия его битов плюс единица), а перенос из самого старшего бита просто игнорируется.