

Арифметические комбинационные схемы

Двоичные сумматоры

Сумматором называется функциональный узел компьютера, предназначенный для сложения двух n -разрядных чисел (слов).

На базе сумматора можно реализовать другие арифметические операции:

- **Операция вычитания** заменяется сложением чисел представленных в обратном или дополнительном коде.
- **Операции умножения и деления** сводятся к реализации операций многократных сложений и сдвигов.

Поэтому сумматор является важнейшей частью любого арифметико-логического устройства (АЛУ).

Функция сумматора на его условно-графическом обозначении (УГО) на принципиальной электрической схеме обозначается буквами SM или S .

Многоразрядный двоичный сумматор состоит из отдельных схем, которые называются **одноразрядными двоичными сумматорами**; они выполняют все действия по сложению значений одноименных разрядов двух чисел (**операндов**).

Сумматоры классифицируются, по следующим признакам:

- **способу сложения** – параллельные, последовательные и параллельно-последовательные;
- **числу входов** – одноразрядные и многоразрядные сумматоры;
- **организацией хранения результата сложения** – комбинационные, накапливающие, комбинированные;
- **организацией цепей переноса между разрядами** – с последовательным, параллельным или комбинированным переносом (с групповой структурой);
- **системой счисления** – позиционные (двоичные, двоично-десятичные) и непозиционные (код Грея, избыточные коды);
- **разрядностью (длиной) операндов** – 8-, 16-, 32-, 64-разрядные;
- **способом представления отрицательных чисел** – в обратном или дополнительном кодах, а также в их модификациях;
- **временем сложения** – синхронные, асинхронные.

Сумматоры характеризуются следующими параметрами:

- **быстродействием** – временем выполнения операции сложения t_{Σ} , которое отсчитывается от начала подачи операндов на входы сумматора до получения результата суммирования на его выходах. Часто быстродействие характеризуется количеством сложений в секунду $F_{\Sigma} = 1/t_{\Sigma}$ (имеются в виду операции типа регистр-регистр, это когда числа хранятся в регистрах АЛУ);
- **аппаратурными затратами** – когда стоимость одноразрядной схемы сложения определяется общим числом логических входов используемых элементов; стоимость многоразрядного сумматора определяется общим количеством используемых микросхем;
- **потребляемой мощностью** сумматора.

Сумматор осуществляет арифметическое суммирование n -разрядных кодов.

$$\begin{array}{r} X = (x_{n-1}, \dots, x_0) - 1\text{-е слагаемое} \\ + \\ Y = (y_{n-1}, \dots, y_0) - 2\text{-е слагаемое} \\ \hline \Sigma = (\mathbf{s}_n, s_{n-1}, \dots, s_0) - \text{сумма} \end{array}$$

Правила сложения двух одноразрядных двоичных чисел, где "+" – арифметическое сложение:

$$0 + 0 = 0$$

$$0 + 1 = 1 + 0 = 1$$

$$1 + 1 = 0 \text{ и перенос } 1 \text{ в старший разряд.}$$

Устройство реализующее эти правила называется **одноразрядным полусумматором** и имеет **два входа** и **два выхода**.

Правила сложение трех одноразрядных чисел производится следующим образом, где "+" – арифметическое сложение:

$$0 + 0 + 0 = 0$$

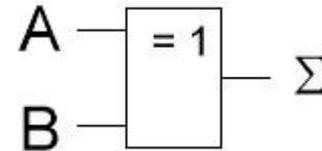
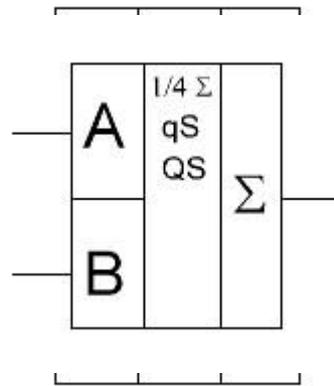
$$0 + 0 + 1 = 1$$

$$0 + 1 + 1 = 0 \text{ и перенос } 1 \text{ в старший разряд}$$

$$1 + 1 + 1 = 1 \text{ и перенос } 1 \text{ в старший разряд.}$$

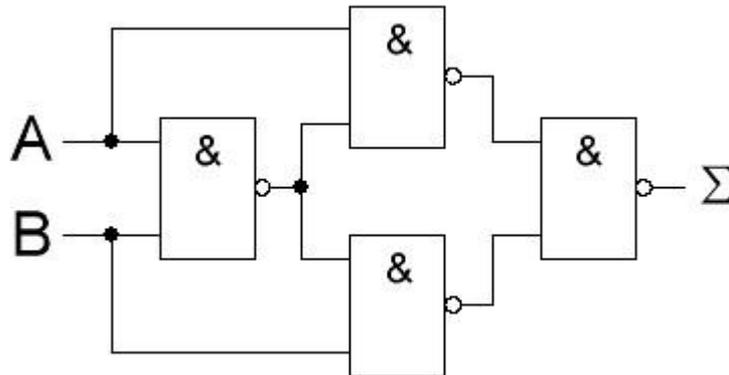
Устройство реализующее эти правила называется **одноразрядным полным сумматором** и имеет **три входа** и **два выхода**.

Четверть сумматор



$$\Sigma = \bar{A} \cdot B + A \cdot \bar{B}$$

$$\begin{aligned} \Sigma &= \bar{A} \cdot B + A \cdot \bar{B} = \bar{A} \cdot A + \bar{A} \cdot B + \bar{B} \cdot B + A \cdot \bar{B} = A \cdot (\bar{A} + \bar{B}) + B \cdot (\bar{A} + \bar{B}) = \\ &= A \cdot \overline{A \cdot B} + B \cdot \overline{A \cdot B} = \overline{\overline{A \cdot B} \cdot \overline{A \cdot B}} \end{aligned}$$

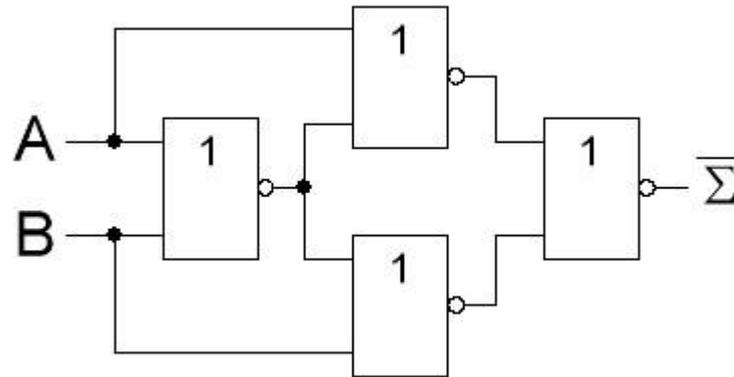


ТТЛ, ТТЛШ – ИС типа ЛП5
155, 531, 1533
555 – ЛП12

КМОП – ИС типа ЛП2 (561, 564)
ЛП12 (1561)

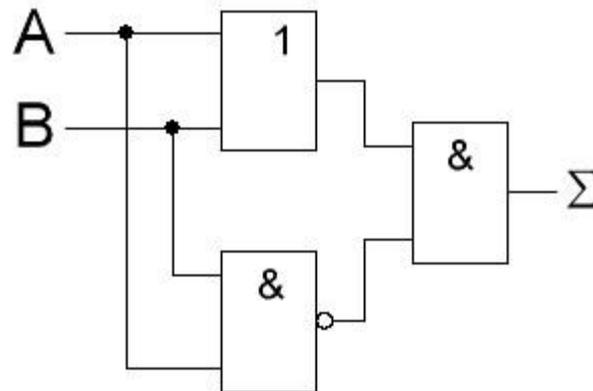
$$\Sigma = \overline{A} \cdot B + A \cdot \overline{B} = \overline{A} \cdot A + \overline{A} \cdot B + \overline{B} \cdot B + A \cdot \overline{B} = \overline{\overline{A} \cdot (A+B)} + \overline{\overline{B} \cdot (A+B)} =$$

$$= \overline{A+A+B+B} = \overline{A+A+B} + \overline{B+A+B}$$



$$\Sigma = \overline{A} \cdot B + A \cdot \overline{B} = \overline{A} \cdot A + \overline{A} \cdot B + \overline{B} \cdot B + A \cdot \overline{B} = \overline{A} \cdot (A+B) + \overline{B} \cdot (A+B) =$$

$$= (A+B) \cdot (\overline{A} + \overline{B}) = (A+B) \cdot \overline{A \cdot B}$$

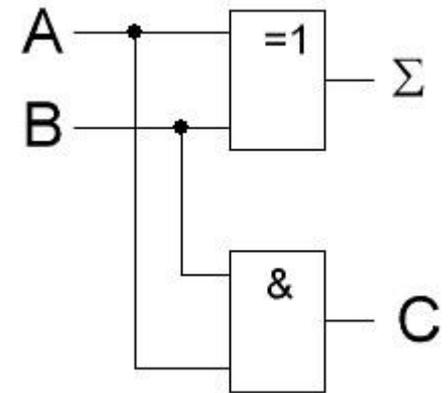


Полусумматор

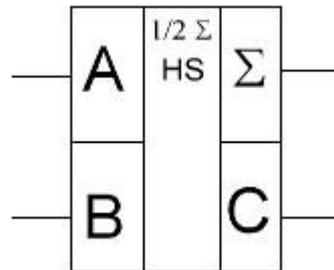
A	B	Σ	C
---	---	----------	---

$$\left\{ \begin{array}{l} \Sigma = \bar{A} \cdot B + A \cdot \bar{B} = A \oplus B \\ C = A \cdot B \end{array} \right.$$

1	1	0	1
---	---	---	---



C – от англ. *Carry* – нести, переносить



Условно-графическое обозначение
полусумматора

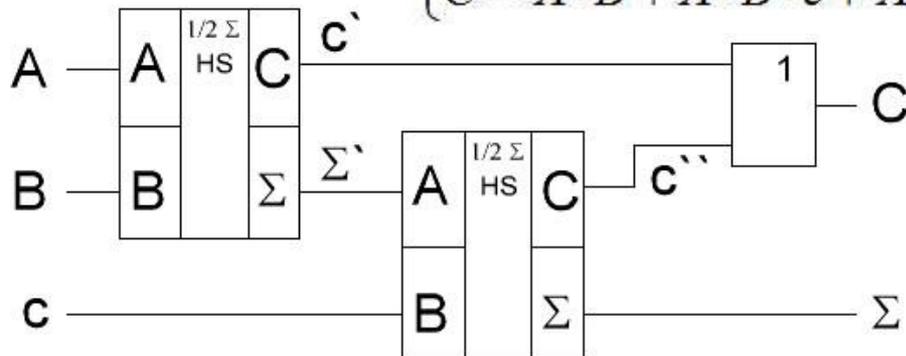
Полный одноразрядный сумматор

	B	A	c	Σ	C
$\Sigma =$	0	0	0	0	0
$C =$	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

$\Sigma = \bar{A} \cdot \bar{B} \cdot \bar{c} + A \cdot B \cdot c$
 $C = \bar{A} \cdot B \cdot \bar{c} + A \cdot B \cdot c = A \cdot B + A \cdot c + B \cdot c$

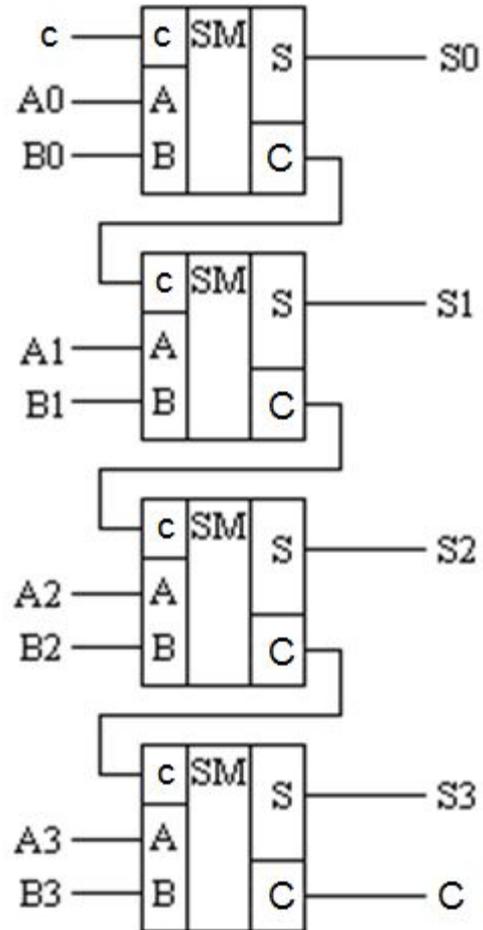
Условно-графическое обозначение полного одноразрядного сумматора

$$\begin{cases} \Sigma = (\bar{A} \cdot B + A \cdot \bar{B}) \cdot \bar{c} + (\bar{A} \cdot \bar{B} + A \cdot B) c = \Sigma' \cdot \bar{c} + \bar{\Sigma}' \cdot c \\ C = A \cdot B + \bar{A} \cdot B \cdot c + A \cdot \bar{B} \cdot c = A \cdot B + \Sigma' \cdot c = c' + c'' \end{cases}$$

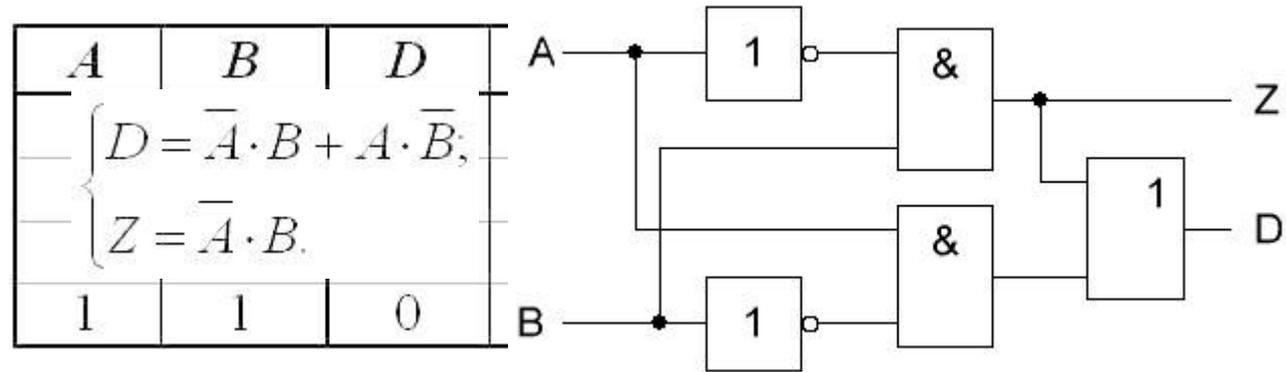


$$\begin{cases} \Sigma = A \oplus B \oplus c \\ C = A \cdot B + c(A \oplus B) \end{cases}$$

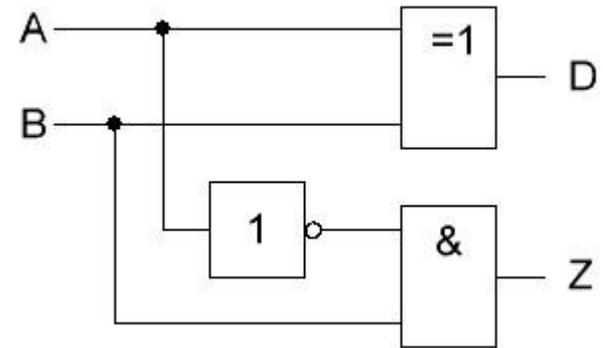
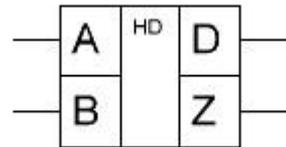
Многоразрядный сумматор с цепью переносов



Полувывчитатель



$$\begin{cases} D = A \oplus B; \\ Z = \bar{A} \cdot B. \end{cases}$$



Условно-графическое обозначение
полувывчитателя

Полный вычитатель

Полный вычитатель необходим для многоразрядного вычитания.

Таблица истинности
полного вычитателя

<i>A</i>	<i>B</i>	<i>z</i>	<i>D</i>	<i>Z</i>
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$\begin{array}{r}
 \begin{array}{r}
 0 \\
 -0 \\
 \hline
 0
 \end{array}
 \quad
 \begin{array}{r}
 1 \\
 -1 \\
 \hline
 0
 \end{array}
 \quad
 \begin{array}{r}
 1 \\
 -0 \\
 \hline
 1
 \end{array}
 \quad
 \begin{array}{r}
 1 \\
 -0 \\
 \hline
 1
 \end{array}
 \quad
 \begin{array}{r}
 1 \\
 -1 \\
 \hline
 1
 \end{array}
 \quad
 \begin{array}{r}
 1 \\
 -0 \\
 \hline
 1
 \end{array}
 \end{array}$$

заяем 1

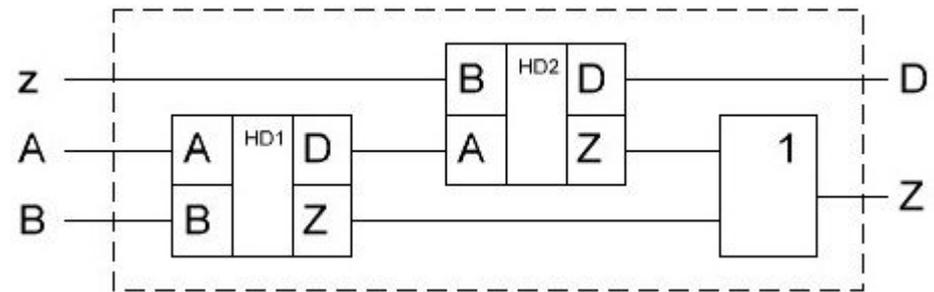
	2^4	2^3	2^2	2^1	2^0	
	1	1	0	1	1	$27 = 1Bh = 1\ 1011$
	-1	0	1	1	0	$- 22 = 16h = 1\ 0110$
Заяем →	1					
	0	0	1	0	1	5

Полный вычитатель может строится на двух полувычитателях.

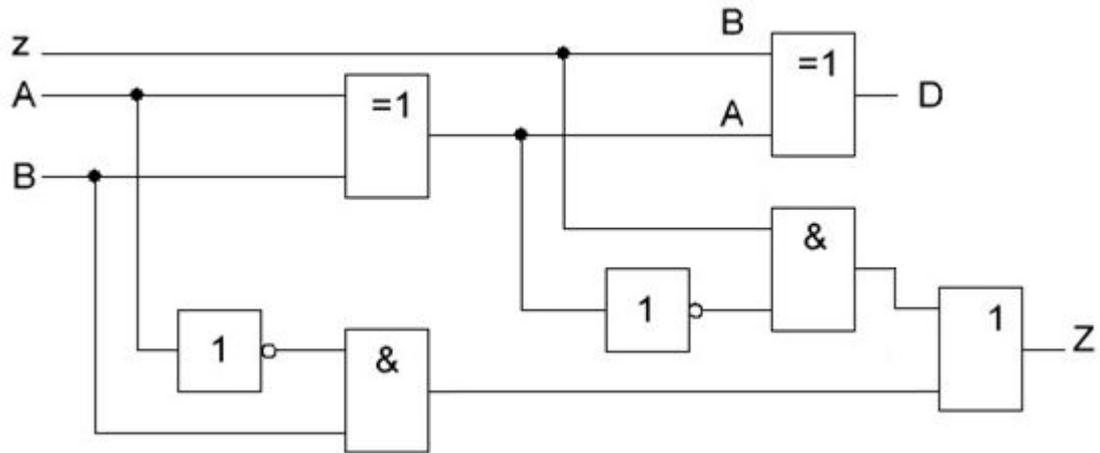
В полувычитателе **HD1** сначала образуется разность **A – B**.

Полувычитателем **HD2** вычитается заем **z** полученный в предыдущем разряде от результата вычитания полученного в **HD1**.

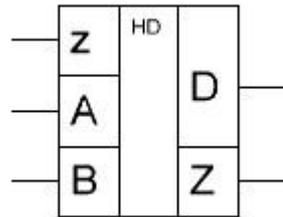
При осуществлении займов как при вычитании в **HD1** так и при вычитании в **HD2** на выходе **Z** появляется сигнал 1.



Структурная схема полного вычитателя



Принципиальная схема полного

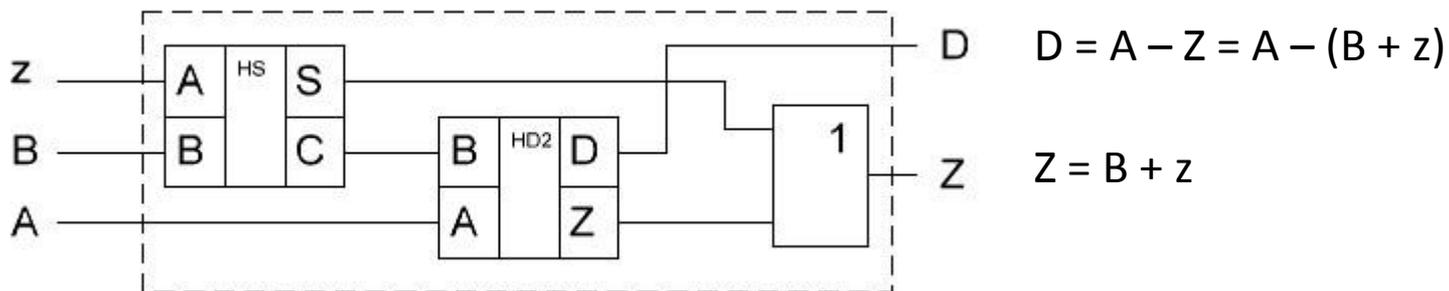


Условно-графическое обозначение
полного вычитателя

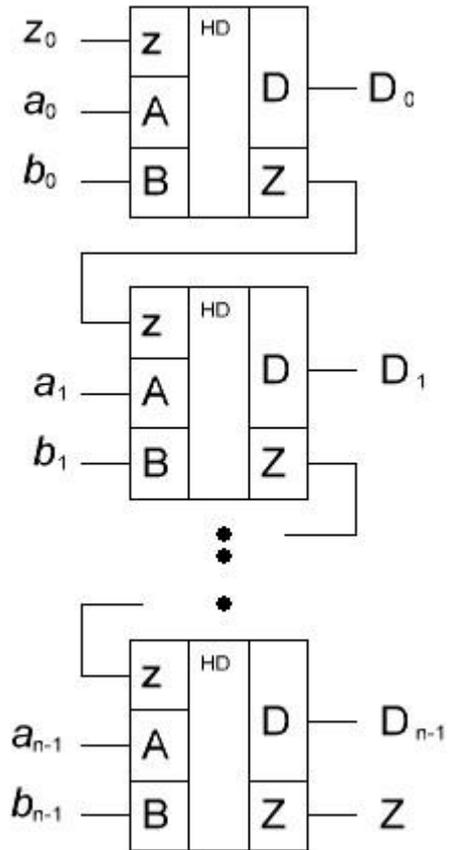
Полный вычитатель является схемой, которая может к значению вычитаемого прибавить сигнал займа (1-сигнал) и такое увеличенное вычитаемое вычесть из уменьшаемого.

$$D = A - (B + z)$$

Полный вычитатель, построенный на полусумматоре и полувычитателе.



Многоразрядный вычитатель со последовательным переносом



Дополняющие формы представление двоичных чисел

Для того, чтобы иметь малые затраты на аппаратную часть (*hardware*) компьютера, были предприняты усилия по сведению к одному алгоритму процедуры вычитания и сложения. Этого можно добиться, если применять **двоичные цифры в их дополняющей форме**.

Различают:

а) **единичное дополнение** (*one's complement*) и

(обратный код, поразрядное дополнение)

$$1001100111 \rightarrow 0110011000$$

б) **двойное дополнение** (*two's complement*) (точное дополнение).

$$11011101011 \rightarrow 00100010100 + 1 \rightarrow 00100010101$$

Если вычитаемое больше, чем уменьшаемое, т.е. $B > A$, то возникающее отрицательное число представляется в выходном регистре неправильно.

Для правильного представления оно должно дополняться, то есть в случае отрицательного результата его инвертируют и прибавляют 1.

Десятичное
Вычитание

$$\begin{array}{r} 10 \\ - 6 \\ \hline 4 \end{array}$$

Двоичное
вычитание

$$\begin{array}{r} 1010 \\ - 0110 \\ \hline 100 \end{array}$$

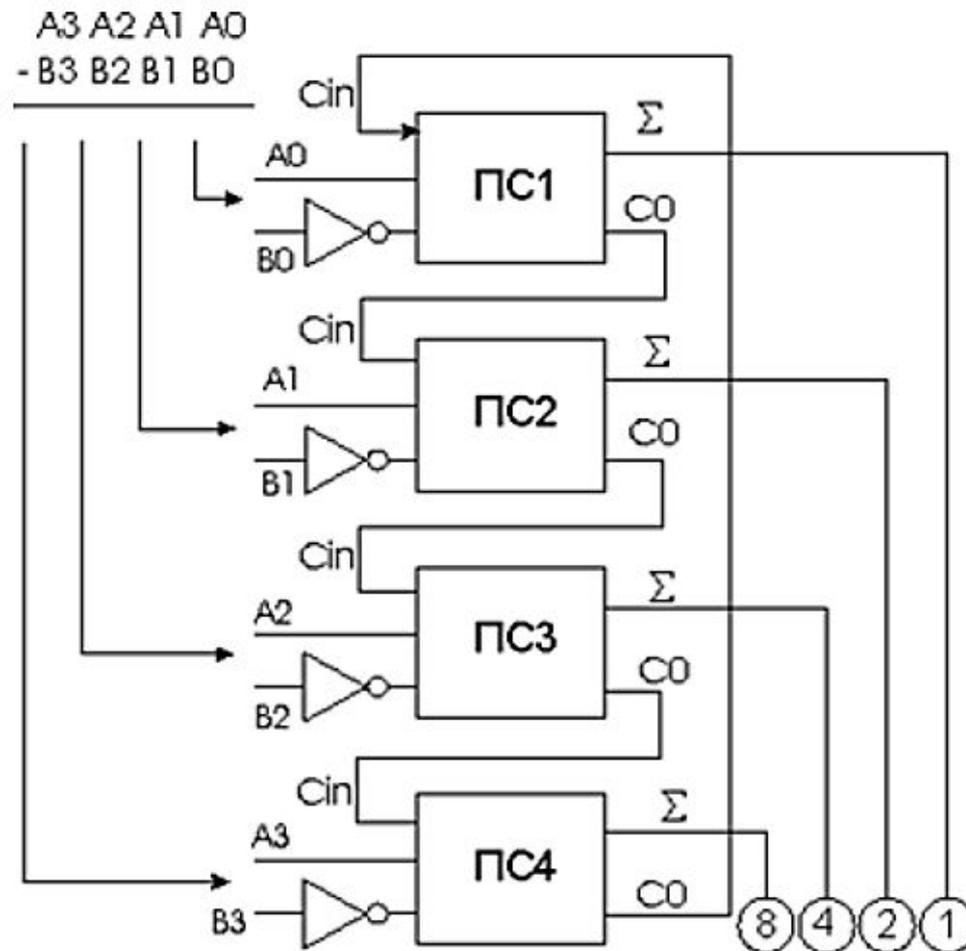
Дополнение до 1
и сложение



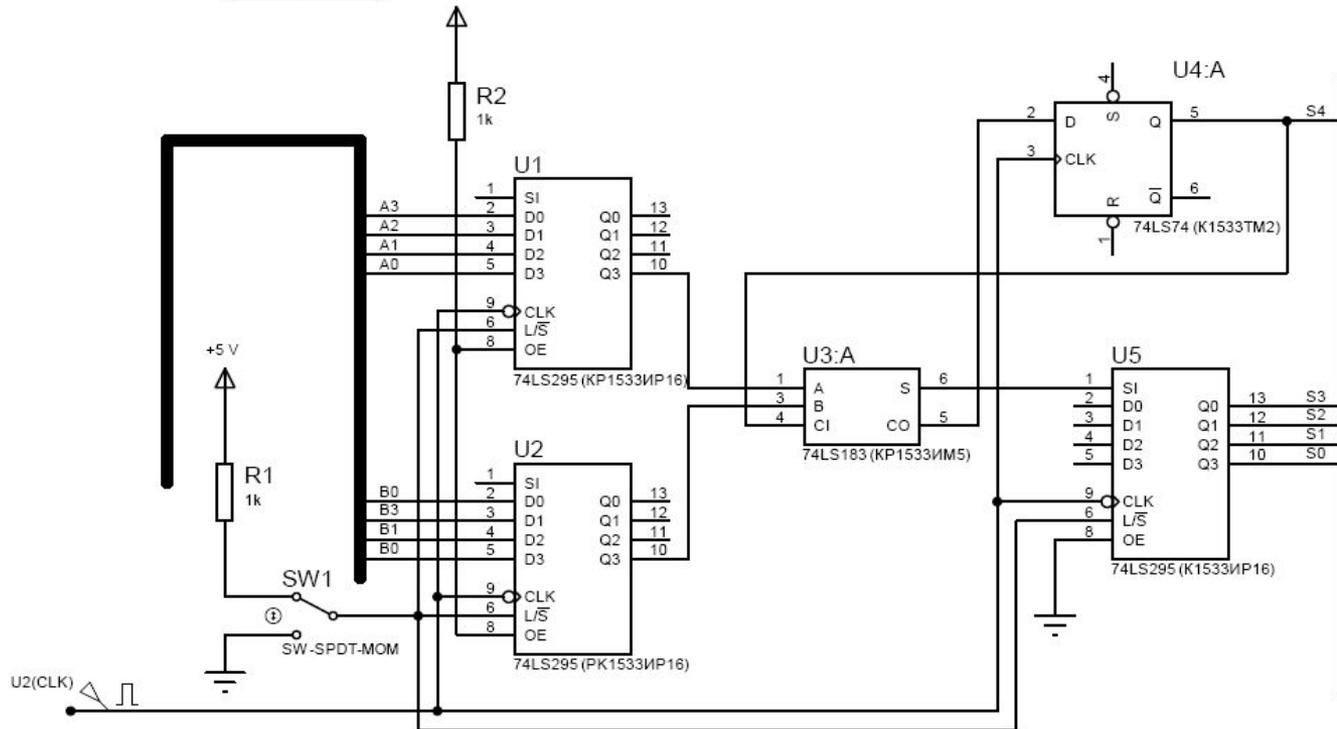
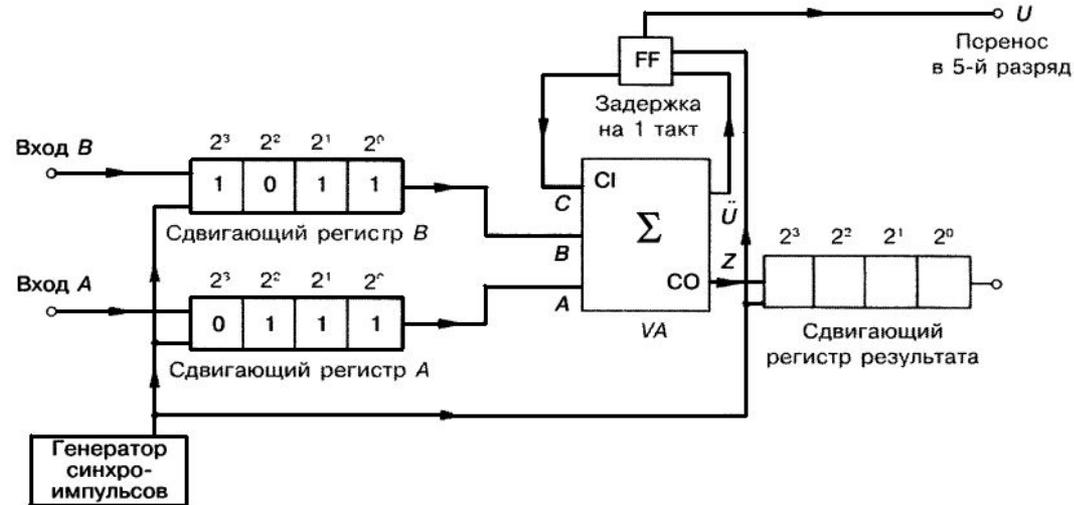
Специальный
прием вычитания

$$\begin{array}{r} 1010 \\ + 1001 \\ \hline 10011 \\ + 1 \\ \hline 100 \end{array}$$

Циклический
перенос



Последовательный сумматор

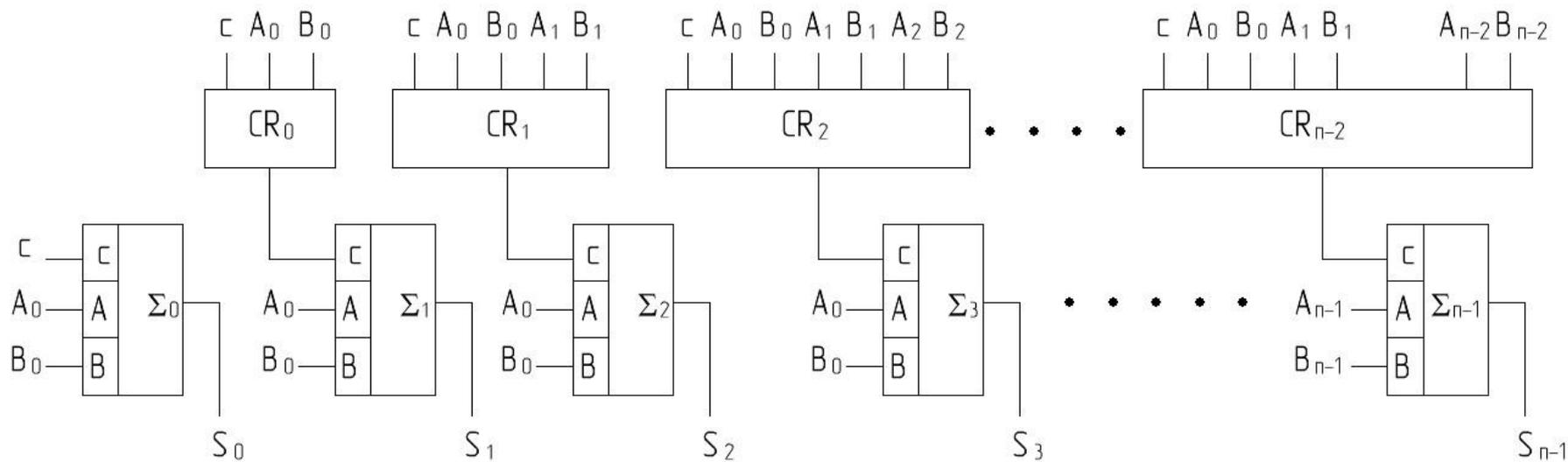


Цепи распространения переноса в многоразрядных сумматорах

Многоразрядный сумматор может быть составлен из одnorазрядных сумматоров, число которых равно числу разрядов слагаемых.

По характеру распространения переноса различают следующие виды сумматоров:

- с поразрядным последовательным переносом;
- с параллельным одновременным переносом;
- с групповым переносом.



Структура сумматора с цепями параллельного

задержки переноса
 Задержка T получения суммы сумматора с параллельным переносом складывается из одинаковых для всех (кроме первого) разрядов задержки блока переноса – $(2 \div 3) \tau$ в зависимости от логического базиса и задержки **трехвходовой схемы сложения по модулю 2** – $(2 \div 4) \tau$.

От числа разрядов ни задержка получения суммы, ни задержка получения выходного переноса $CR_{\text{групп}}$ не зависят. Аппаратурные затраты сумматора с параллельным переносом заметно превышают сумматора с последовательным переносом и быстро растут с ростом разрядности.

Диапазон разрядности, в пределах которого сумматор с параллельным переносом эффективен, невелик. Так при малой разрядности $2 \div 3$, и даже 4 он хуже сумматора с последовательным переносом и по аппаратурным затратам, и по затрачиваемому времени T .

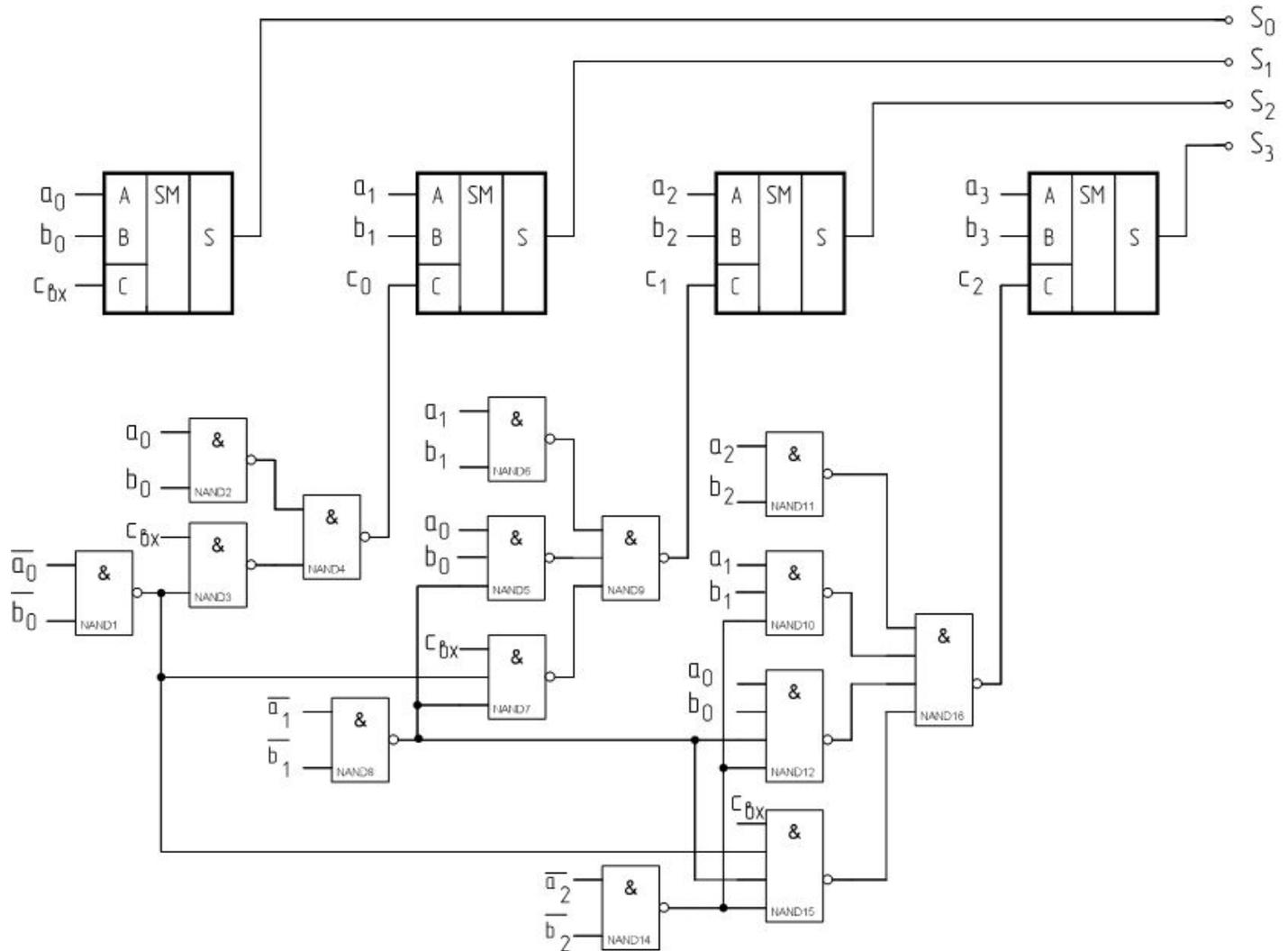


Схема формирования сигналов параллельного переноса

Сумматор с групповым переносом

Для ускорения переноса в сумматорах с большим числом разрядов применяют принцип *группового переноса*. Сумматор разбивают на группы, представляющие собой небольшие сумматоры с разрядностью обычно от 2 до 8.

Блоки переноса анализируют слагаемые, т.е. определяют состояние группы и если из группы должен быть перенос, то он появляется на выходе блока для подачи его на вход следующей группы и в цепочку распространения переноса от младшей группы к старшей. Тракт группового переноса построен так, что время распространения переноса в нем между группами оказывается меньше, чем если бы этот перенос распространялся по цепям внутригрупповых трактов

Переносы в группе определяются по формулам как для обычных сумматоров с параллельным переносом, но сами **сумматоры благодаря делению на группы существенно упрощаются**, т.к. у них все **блоки формирования переноса имеют одинаковую сложность**, тогда как **в сумматоре с параллельным переносом сложность схем переноса возрастает непрерывно** от предыдущего разряда к последующему.



Схема формирования сигналов группового переноса

Как и в обычном сумматоре, который можно рассматривать как частный случай сумматора с групповым переносом, когда разрядность каждой группы равна 1, тракт межгруппового переноса может быть построен:

- a) как **параллельным**, когда все групповые переносы вырабатываются параллельно как функции только слагаемых,
- b) так и **последовательным**, когда исходным материалом для переноса в каждую следующую группу служит перенос, поступающий на вход данной группы.

Параллельный перенос между группами в сочетании с параллельным переносом внутри группы дает самые быстрые сумматоры в диапазоне разрядности, приблизительно от 24 до 64. Задержка таких *параллельно-параллельных* сумматоров не зависит от разрядности и составляет $(9 \div 10)\tau$ в зависимости от используемого логического базиса. За скорость приходится платить, и аппаратурные **затраты таких сумматоров заметно превышают затраты сумматоров с другими типами переносов.**

В диапазоне разрядности примерно от 8 до 24 первенство по скорости переходит к сумматорам с параллельным переносом между группами и с последовательным внутри групп. Разрядность групп при этом выбирают небольшой – от 2 до 4.

Умножители

Параллельные умножители

При умножении используются понятия **множимое** и **множитель**.

Множимое — это базовое число.

Множитель является числом, на которое увеличивается множимое. В результате получается **произведение**.

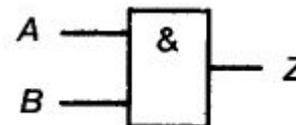
$$\begin{array}{rcccl} \text{Множимое} & & \text{Множитель} & & \text{Произведение} \\ 2 & \times & 3 & = & 6 \end{array}$$

Например, перемножим два двухразрядных двоичных числа: $2 \cdot 3 = 6$

$$\begin{array}{r} 10_2 \\ \underline{11_2} \\ 10_2 \quad \rightarrow \text{1-е слагаемое} \\ \underline{10_2} \quad \rightarrow \text{2-е слагаемое} \\ 110_2 \quad \rightarrow \text{Результат} \end{array}$$

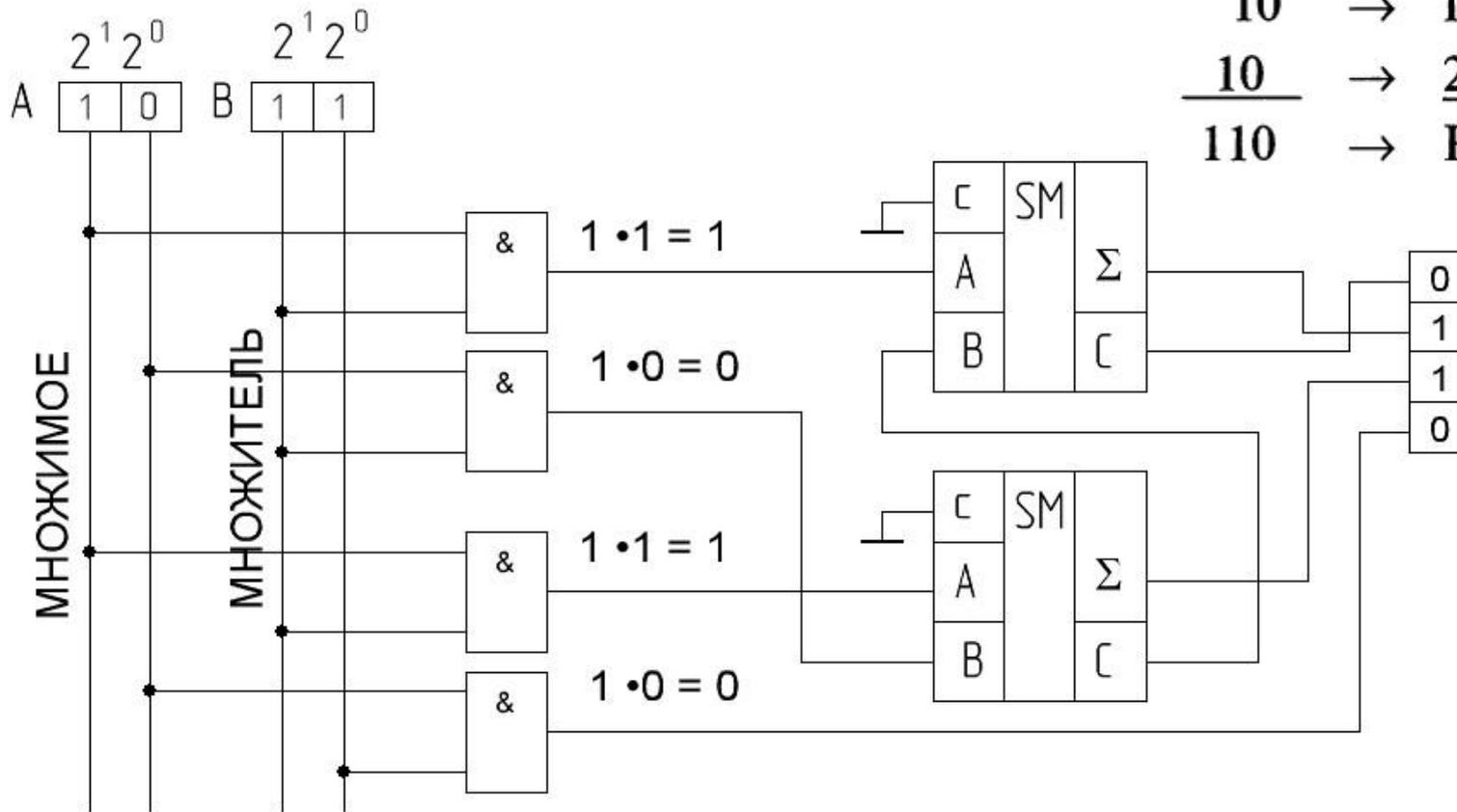
$$\begin{array}{l} 0 \cdot 0 = 0 \\ 0 \cdot 1 = 0 \\ 1 \cdot 0 = 0 \\ 1 \cdot 1 = 1 \end{array}$$

Вар.	B	A	X
1	0	0	0
2	0	1	0
3	1	0	0
4	1	1	1



Элемент И является 1-битовым умножителем

Пример: $2 \cdot 3 = 6$



10 · 11

10 → 1-е слагаемое

10 → 2-е слагаемое

110 → Результат

Схема умножителя для получения слагаемых

Оба слагаемых должны теперь складываться поразрядно. Сложение происходит в 2-битовом параллельном сумматоре. И-элементы должны быть включены таким образом, чтобы второе слагаемое прибавлялось к первому слагаемому со смещением на одну позицию влево.

		A_2	A_1	A_0
	\times	B_2	B_1	B_0
0	0	A_2B_0	A_1B_0	A_0B_0
0	A_2B_1	A_1B_1	A_0B_1	0
A_2B_2	A_1B_2	A_0B_2	0	0
M_4	M_3	M_2	M_1	M_0

$$M_0 = A_0B_0$$

$$M_1 = A_1B_0 + A_0B_1$$

$$M_2 = A_2B_0 + A_1B_1 + A_0B_2$$

$$M_3 = A_2B_1 + A_1B_2$$

$$M_4 = A_2B_2$$

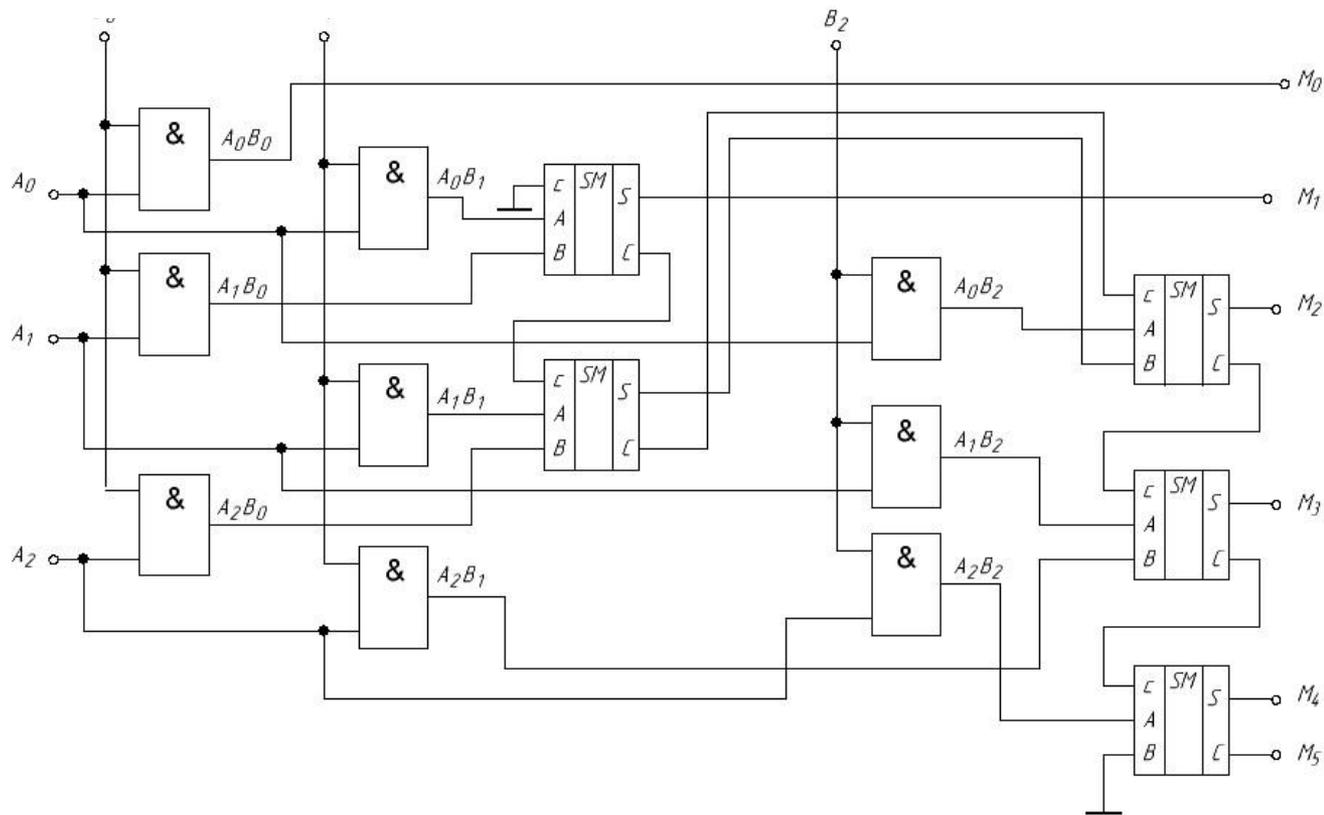


Схема 3-х битового умножителя для получения произведения

4-битовый параллельный умножитель

Пример: $9 \cdot 11 = 99$

$$\begin{array}{r}
 1\ 0\ 0\ 1 \cdot 1\ 0\ 1\ 1 \\
 \hline
 1\ 0\ 0\ 1 \quad \leftarrow \\
 1\ 0\ 0\ 1 \quad \leftarrow \\
 0\ 0\ 0\ 0 \quad \leftarrow \\
 1\ 0\ 0\ 1 \quad \leftarrow \\
 \hline
 1\ 1\ 0\ 0\ 0\ 1\ 1
 \end{array}$$

- 1-е слагаемое
- 2-е слагаемое
- 3-е слагаемое
- 4-е слагаемое
- Результат

