

Тема 2

Блок арифметического логического устройства

Теория

Арифметическое логическое устройство (АЛУ) выделено на структурной схеме микропроцессора на рис. 1. Оно выполняет 5 видов операций над 8-разрядными числами:

- 1) арифметическое сложение
- 2) вычитание
- 3) логическое сложение
- 4) логическое умножение
- 5) ИСКЛЮЧАЮЩЕЕ ИЛИ

Каждая из этих операций производится над двумя числами, называемыми операндами.

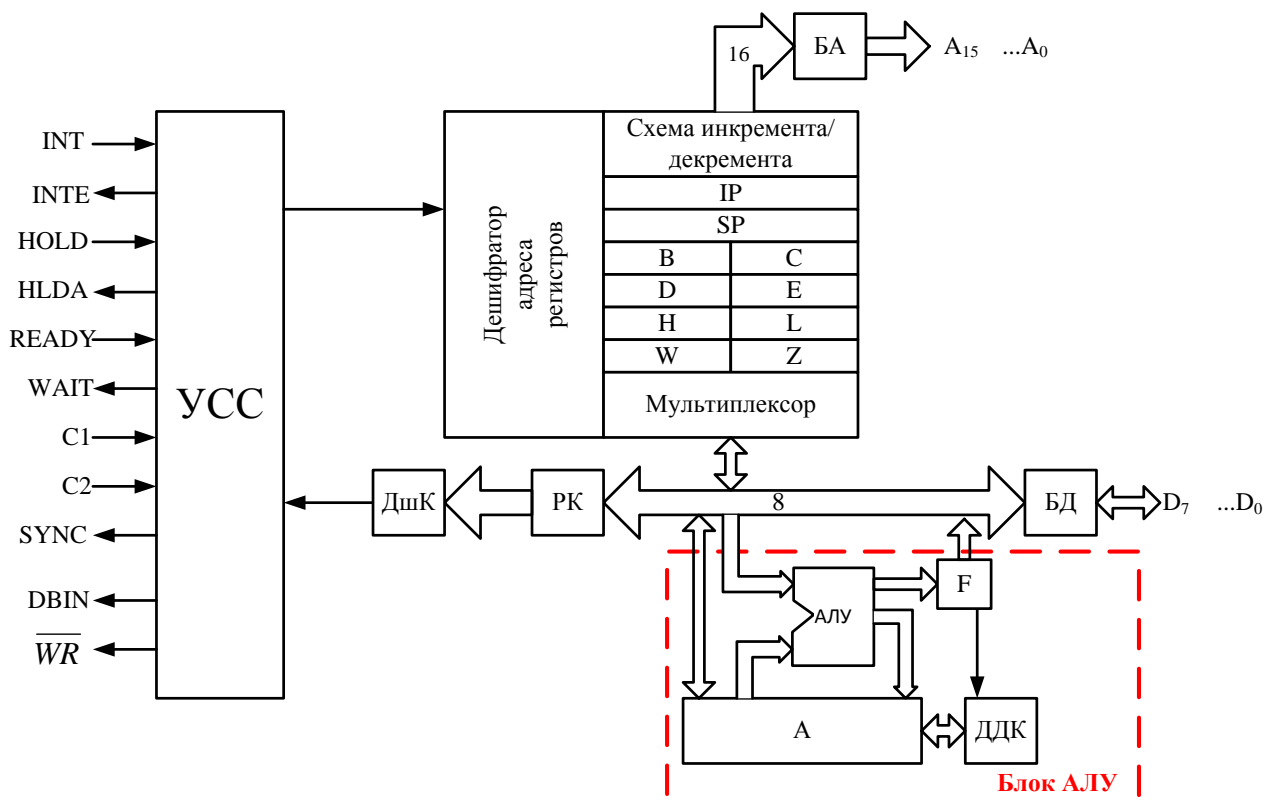


Рис. 1. Структурная схема МП с выделенным блоком АЛУ

Первый операнд всегда поступает в АЛУ из аккумулятора (рис. 1), второй может быть взят:

- ✓ из регистра микропроцессора;
- ✓ из ячейки памяти, адрес которой заранее задается в регистровой паре HL;
- ✓ из второго байта команды – т.е. из ячейки программной памяти.

Результат операции записывается всегда в аккумулятор.

Шина данных МП 8-разрядная, следовательно, возможно выполнение операций над числами от 0 до 255. По результатам операций устанавливаются следующие признаки (флажки):

Z (от англ. Zero - нуль) – признак 0. Он устанавливается в 1, если результат операции – число состоящее из одних нулей.

S (от англ. Sign - знак) – признак знака числа, устанавливается по старшему разряду аккумулятора. Число считается положительным, если $S=0$ и отрицательным, если $S=1$.

P (от англ. Parity - четность) – признак четности. Устанавливается в 1, если в результате выполнения операции получается число, содержащее четное количество единиц.

C (от англ. Carry - перенос) – признак переноса. Устанавливается в 1, если

- ✓ в результате арифметического сложения получается перенос из старшего разряда аккумулятора;
- ✓ если при вычитании уменьшаемое меньше вычитаемого (из меньшего числа вычитается большее число).

АС (от англ. Auxiliary carry – вспомогательный перенос) – признак переноса из младшей тетрады числа в старшую при выполнении сложения. Устанавливается в 1 при возникновении переноса из третьего разряда числа в четвертый (младший разряд - нулевой). Данный признак необходим при работе с двоично-десятичными числами.

Арифметическое сложение

Осуществляется по правилам двоичной арифметики

$$\begin{array}{r} +0 \quad +0 \quad +1 \quad +1 \\ \underline{0} \quad \underline{1} \quad \underline{0} \quad \underline{1} \\ 0 \quad 1 \quad 1 \quad 0 \end{array}$$

1 ← Перенос в следующий разряд

Пример сложения чисел 5 и 100 приведен на рис. 2. 8-разрядный двоичный код числа 5 – это первый операнд с разрядами $x_0 \dots x_7$, 8-разрядный двоичный код числа 100 – это второй операнд с разрядами $y_0 \dots y_7$. Двоичные разряды результата сложения обозначены $z_0 \dots z_7$.

По результатам этого сложения будут установлены признаки (рис. 2)–

$$Z=0, S=0, P=1, C=0, AC=0.$$

Очевидно, что для реализации данного действия необходимо создать на кристалле полупроводника логическую схему, функциональное представление которой показано на рис. 3.

Эта схема состоит из двух 4-разрядных сумматоров. Разряды суммы $z_0 \dots z_7$ фиксируются в аккумуляторе А как результат сложения.

Перенос из сумматора 1 в сумматор 2 представляет собой признак вспомогательного переноса AC. Выходной перенос из старшего разряда сумматора 2 – это признак переноса C.

Логическое ИЛИ-НЕ всех разрядов суммы дает признак нуля Z (если хотя бы один разряд результата отличен от нуля, $Z=0$).

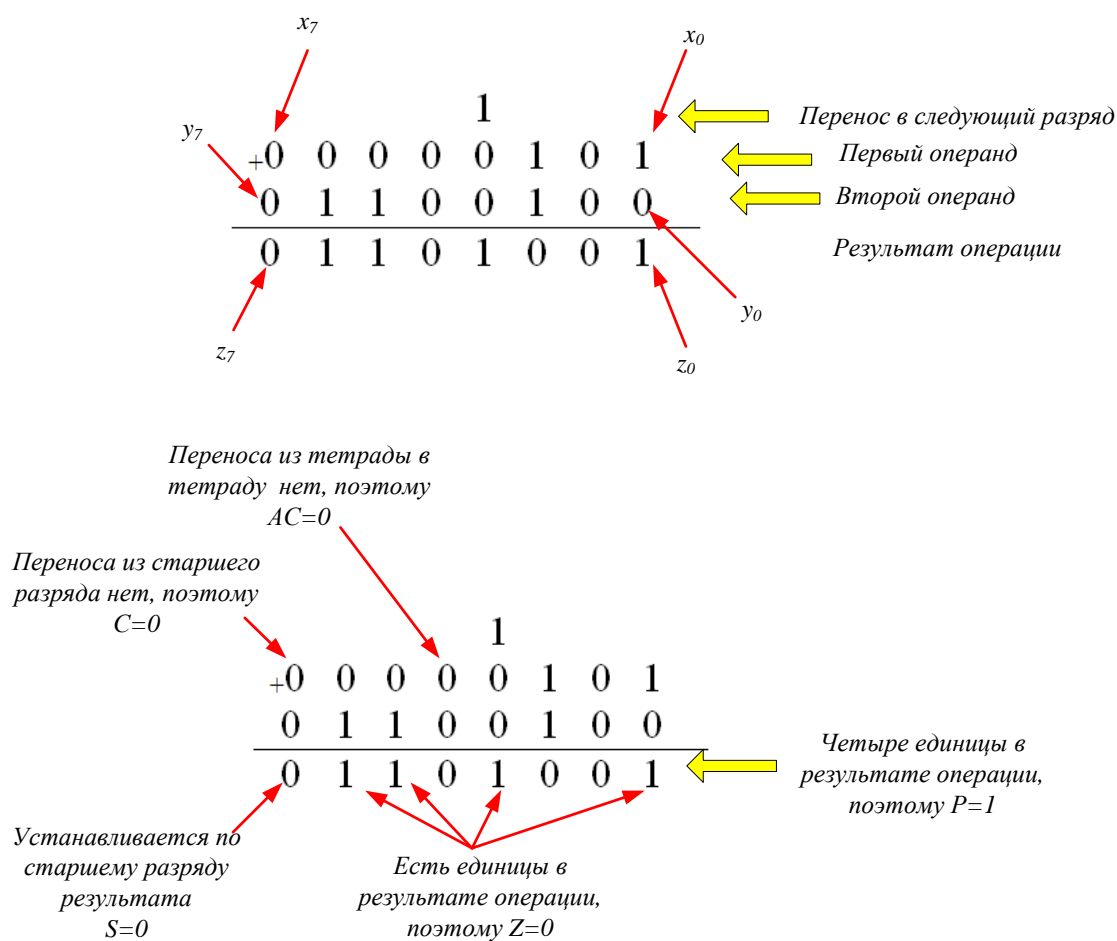


Рис. 2. Пример 8-разрядного арифметического сложения

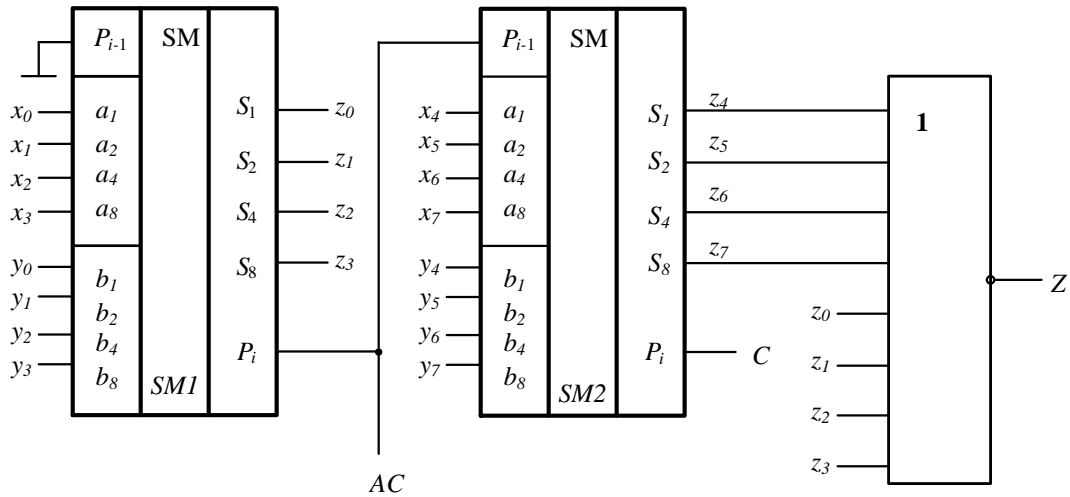
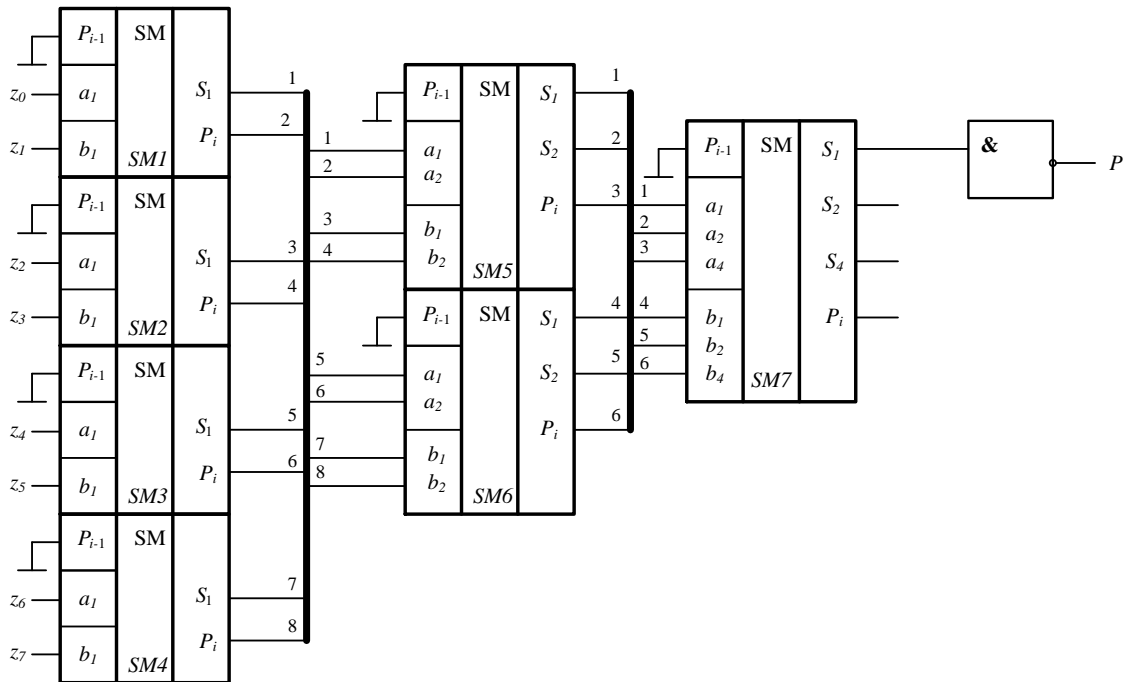


Рис. 3. Функциональная схема реализации арифметического сложения в АЛУ и получение флажков (кроме четности P)

Для получения признака четности P арифметически складываются все разряды суммы. Функциональная схема этого суммирования приведена на рис. 4. Если младший разряд суммы на сумматоре 7 равен 0, число единиц в результате четное, P=1 (на выходе инвертора). Если этот разряд равен 1 – нечетное и P=0.



с. 4. Функциональная схема получения признака четности

Вычитание

Осуществляется в дополнительном коде. Для второго операнда сначала получают обратный код (инверторы на рис

. 5), потом к нему прибавляется единица с помощью сумматора 1. Сложение уменьшаемого и вычитаемого в дополнительном коде осуществляется на сумматоре 2. На рис. 5 не показана схема получения признаков четности и вспомогательного переноса (см. рис. 4).

При программировании и отладке программ нужно учитывать, что если производилось вычитание из большего числа меньшего, результат получается в дополнительном коде. И поскольку в этом случае число получается отрицательным, для проверки нужно переходить от дополнительного кода к прямому.

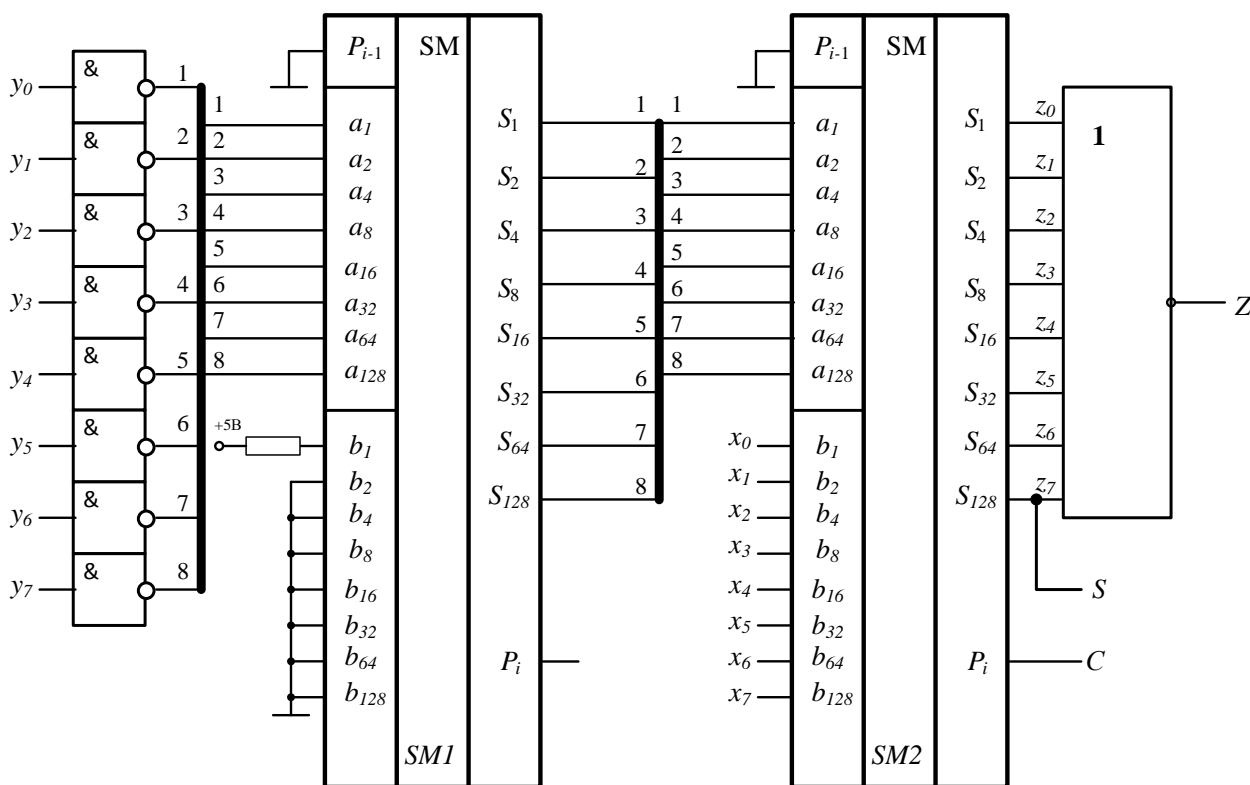


Рис. 5. Функциональная схема вычитания (сложения в дополнительном коде)

Например, при вычитании 05H-06H будет получено число FFH, т.е. (-1) в дополнительном коде. Для проверки сначала получим обратный код значащей части числа –

$$\begin{array}{r} \underline{1\ 1\ 1\ 1\ 1\ 1\ 1\ 1} \leftarrow \text{ДК} \\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1 \\ \hline 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0 \leftarrow \text{ОК} \end{array}$$

После чего обращением получаем прямой код – 00000001=1.

Логическое умножение

Осуществляется по правилам логической функции И над одноименными разрядами 8-разрядных чисел – младший разряд первого операнда логически умножается на младший разряд второго операнда и т.д. до старшего разряда (рис. 6). Признаки переноса и вспомогательного переноса обнуляются: AC=0 и C=0. Остальные флажки устанавливаются по полученному в аккумуляторе результату (см. выше). Пример выполнения логического умножения представлен на рис. 7.

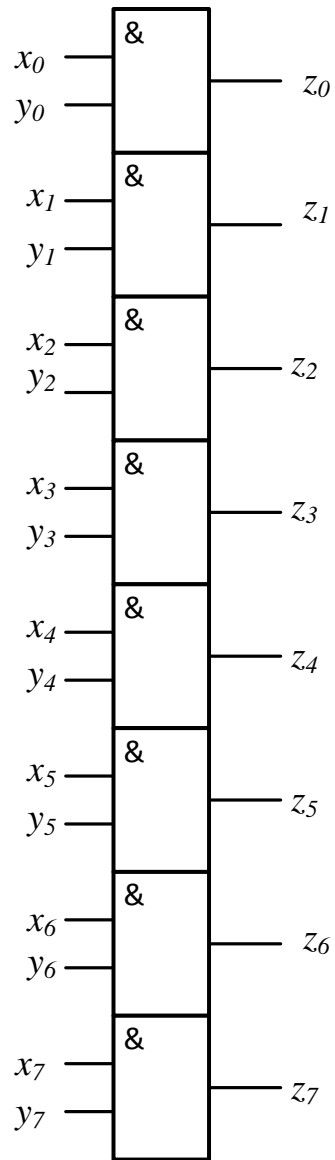


Рис. 6. Функциональная схема поразрядного логического умножения

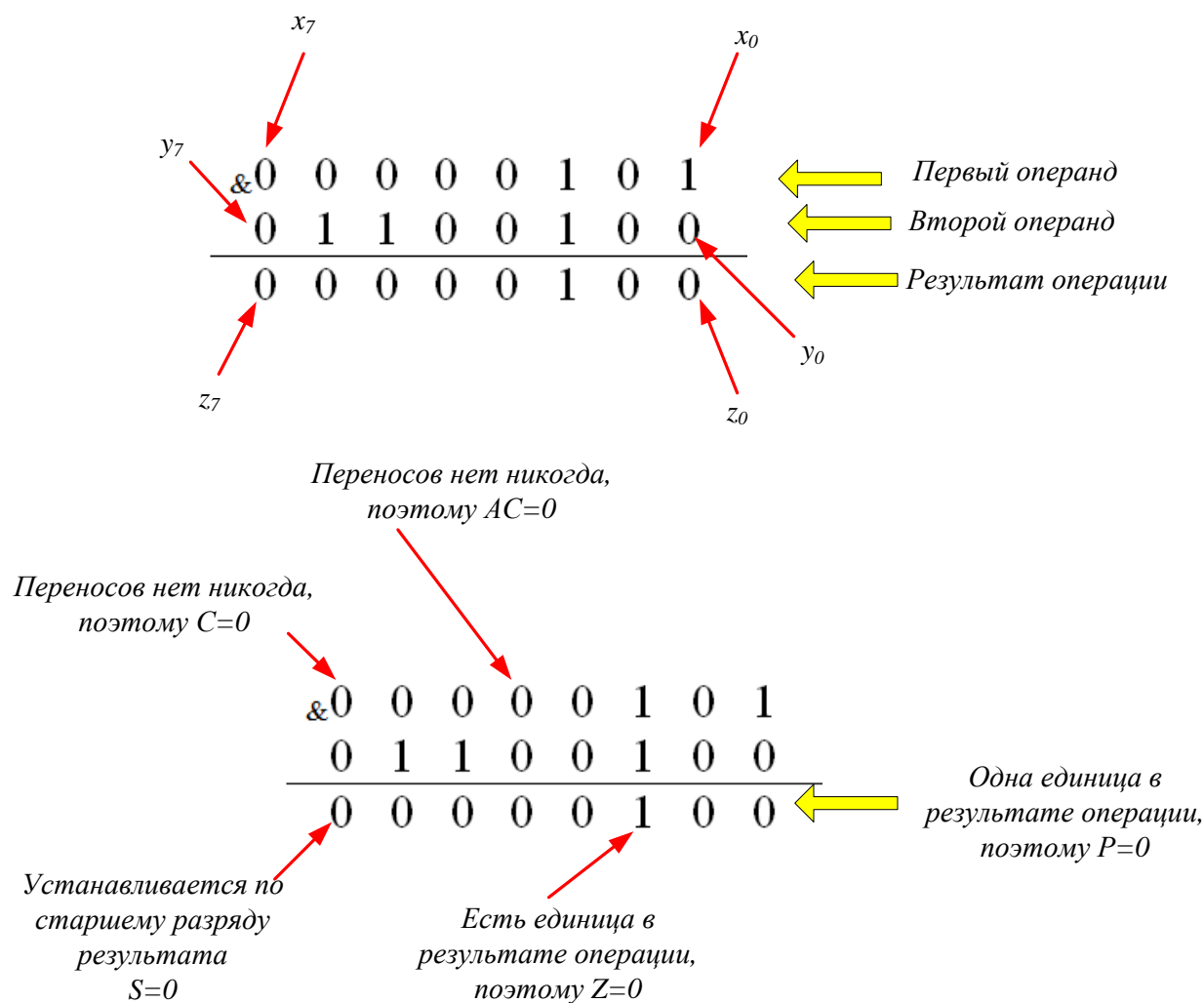


Рис. 7. Пример логического умножения

Логическое сложение

Осуществляется по правилам логической функции ИЛИ над одноименными разрядами 8-разрядных чисел – младший разряд первого операнда логически складывается с младшим разрядом второго операнда и т.д. до старшего разряда (рис. 8).

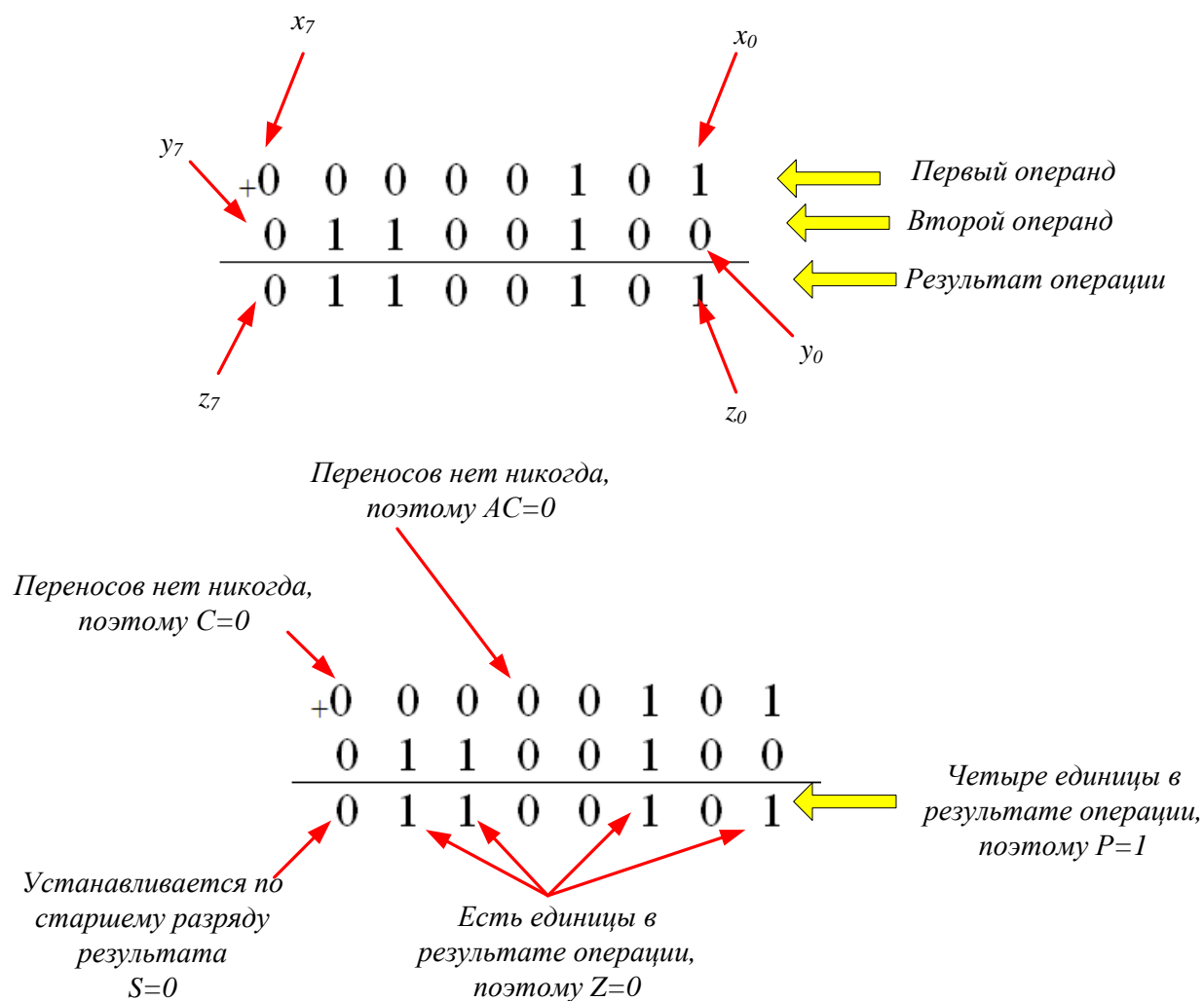


Рис. 8. Пример логического сложения

Признаки переноса и вспомогательного переноса обнуляются $AC=0$ и $C=0$. Остальные флажки устанавливаются по полученному в аккумуляторе результату (см. выше). Функциональная схема логического сложения представлена на рис. 9 (без формирования флажков).

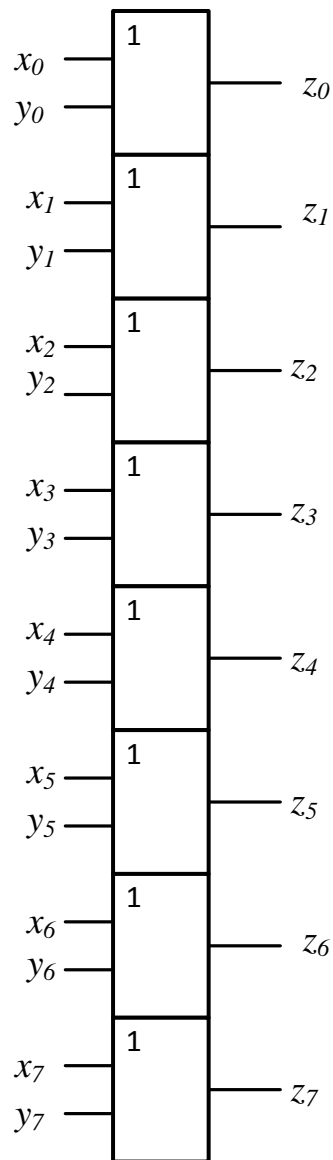


Рис. 9. Функциональная схема поразрядного логического сложения

ИСКЛЮЧАЮЩЕЕ ИЛИ

У данной логической операции несколько названий:

- ✓ Исключающее ИЛИ;
- ✓ НЕРАВНОЗНАЧНОСТЬ;
- ✓ Сложение по модулю 2.

Осуществляется по правилам логической функции ИСКЛЮЧАЮЩЕЕ ИЛИ над одноименными разрядами 8-разрядных чисел в соответствии со следующими правилами –

$\neq 0$	$\neq 0$	$\neq 1$	$\neq 1$
0	1	0	1
<hr/>	<hr/>	<hr/>	<hr/>
0	1	1	0

Данная логическая операция имеет еще одно название - НЕРАВНОЗНАЧНОСТЬ, поскольку 1 получается только тогда, когда действие осуществляется с отличающимися друг от друга операндами. Логическое выражение для НЕРАВНОЗНАЧНОСТИ $y = \bar{a} \cdot b + a \cdot \bar{b}$.

Пример выполнения операции приведен на рис. 10. Поскольку операция логическая, признаки переноса и вспомогательного переноса обнуляются $AC=0$ и $C=0$. Остальные флажки устанавливаются по полученному в аккумуляторе результату (см. выше).

Функциональная схема поразрядного Исключающего ИЛИ приведена на рис. 11.

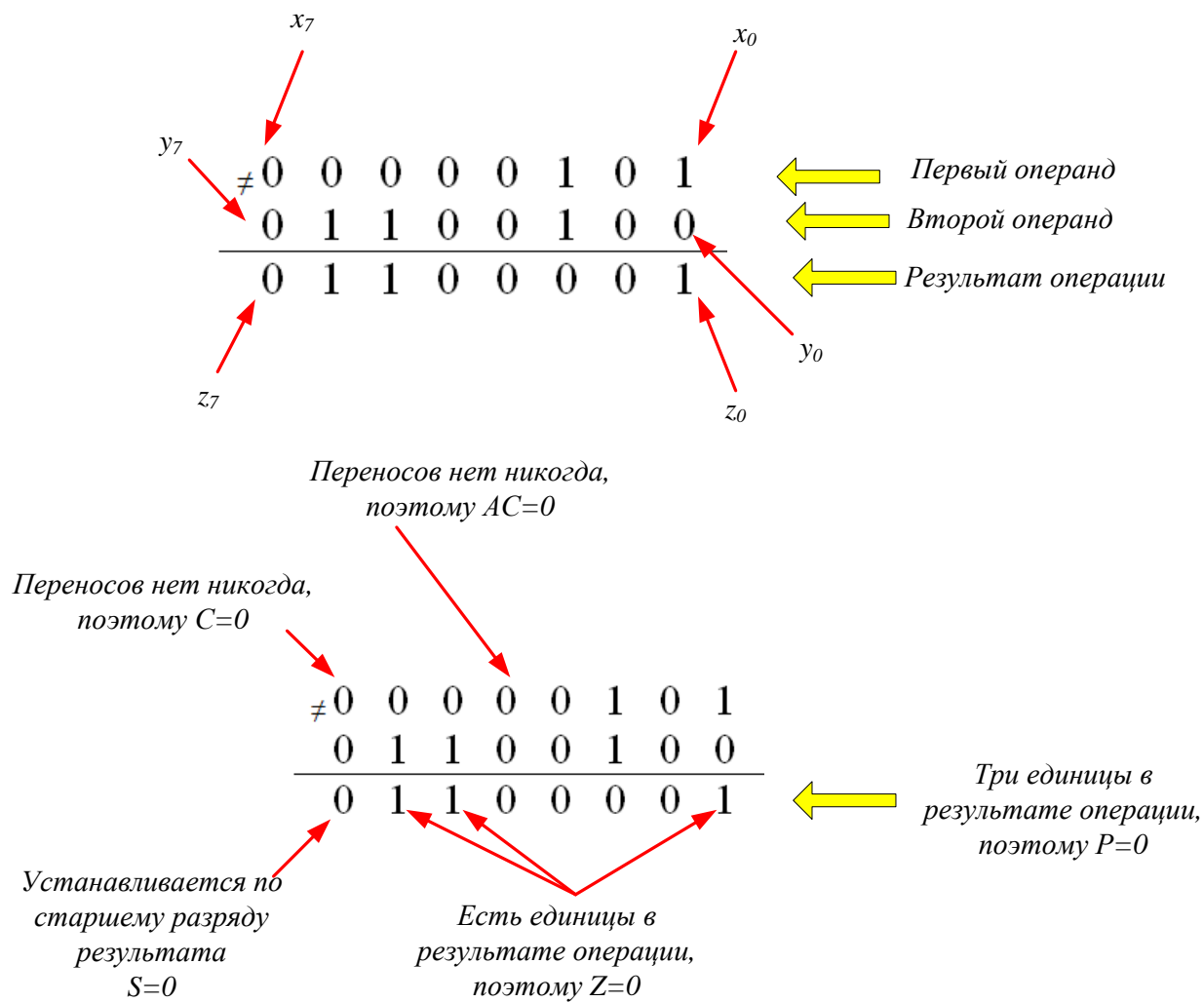


Рис. 10. Пример выполнения Исключающего ИЛИ

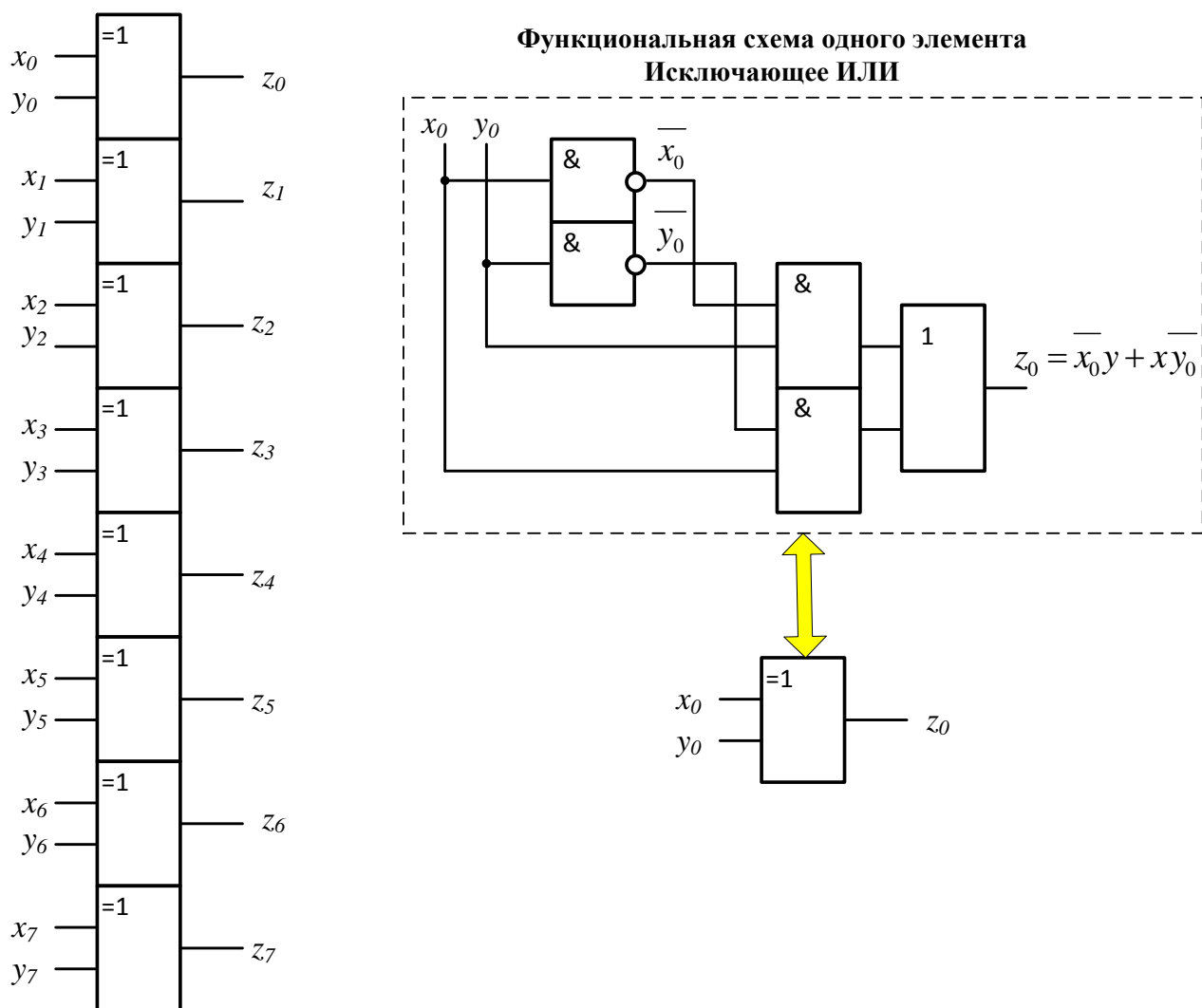


Рис. 11. Функциональная схема Исключающего ИЛИ для 8-разрядных чисел

Регистр признаков

Рассмотренные флажки сохраняются в специальном регистре признаков F в формате, представленном на рис. 12.

Разряды данных	D7	D6	D5	D4	D3	D2	D1	D0
Флажки	S	Z	0	AC	0	P	1	C

Рис. 12. Формат представления информации в регистре признаков

Схема двоично-десятичной коррекции

В микропроцессоре реализована логическая схема добавления кода 6 к младшей тетраде двоичного числа, если признак вспомогательного переноса

АС=1 (рис. 13). Анализ полученных чисел в соответствии с правилами двоично-десятичного сложения аппаратно не реализован, его нужно осуществлять программным путем.

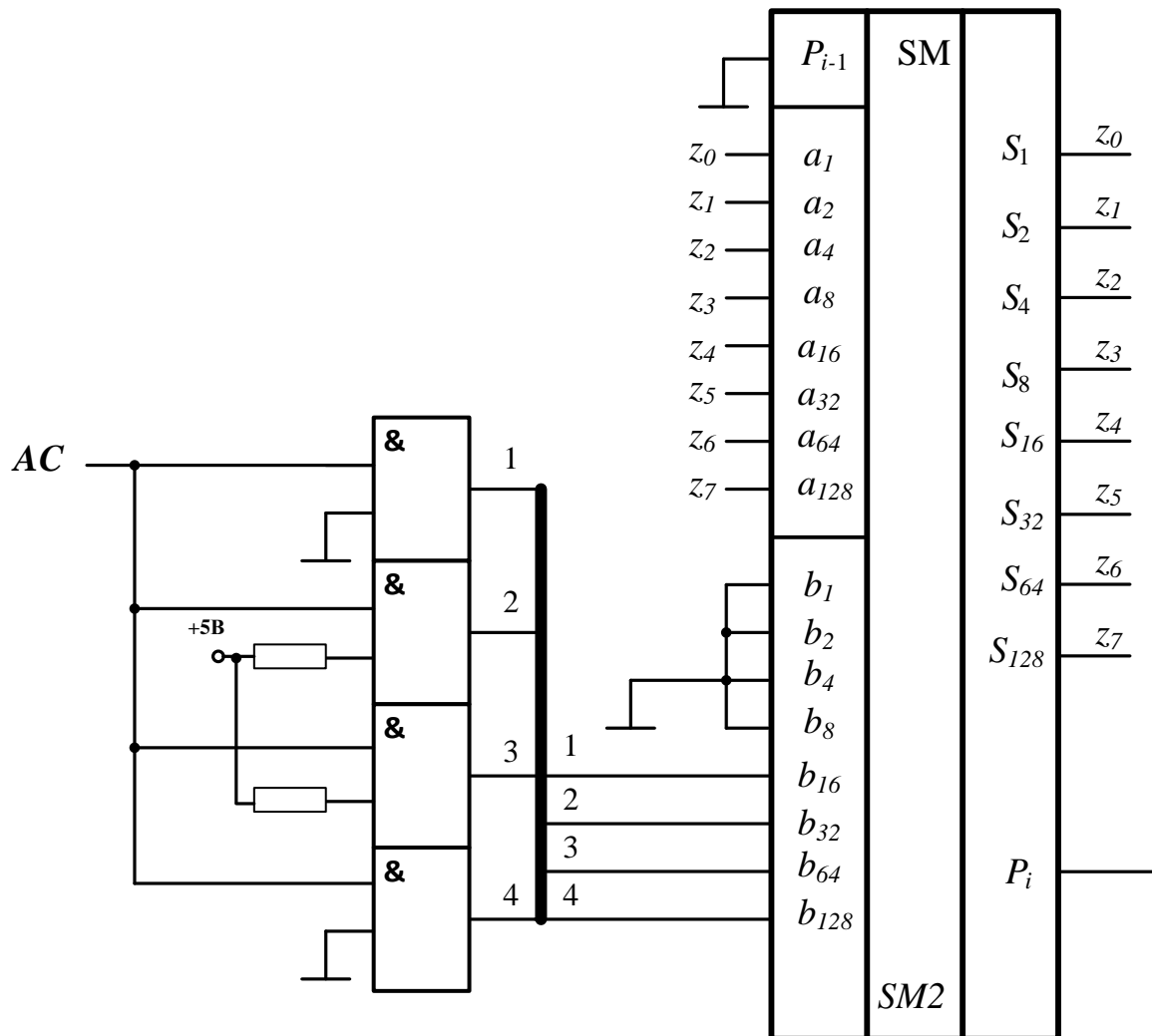


Рис. 13. Функциональная схема реализации ДДК