

Глава 3. Арифметико- Логические Устройства ЭВМ

Курсы лекций «ЭВМ и периферийные устройства»,
«Архитектура вычислительных систем»

Лектор - доцент, к.т.н.

Кузьмин Александр Александрович

Типы арифметических устройств и их структуры

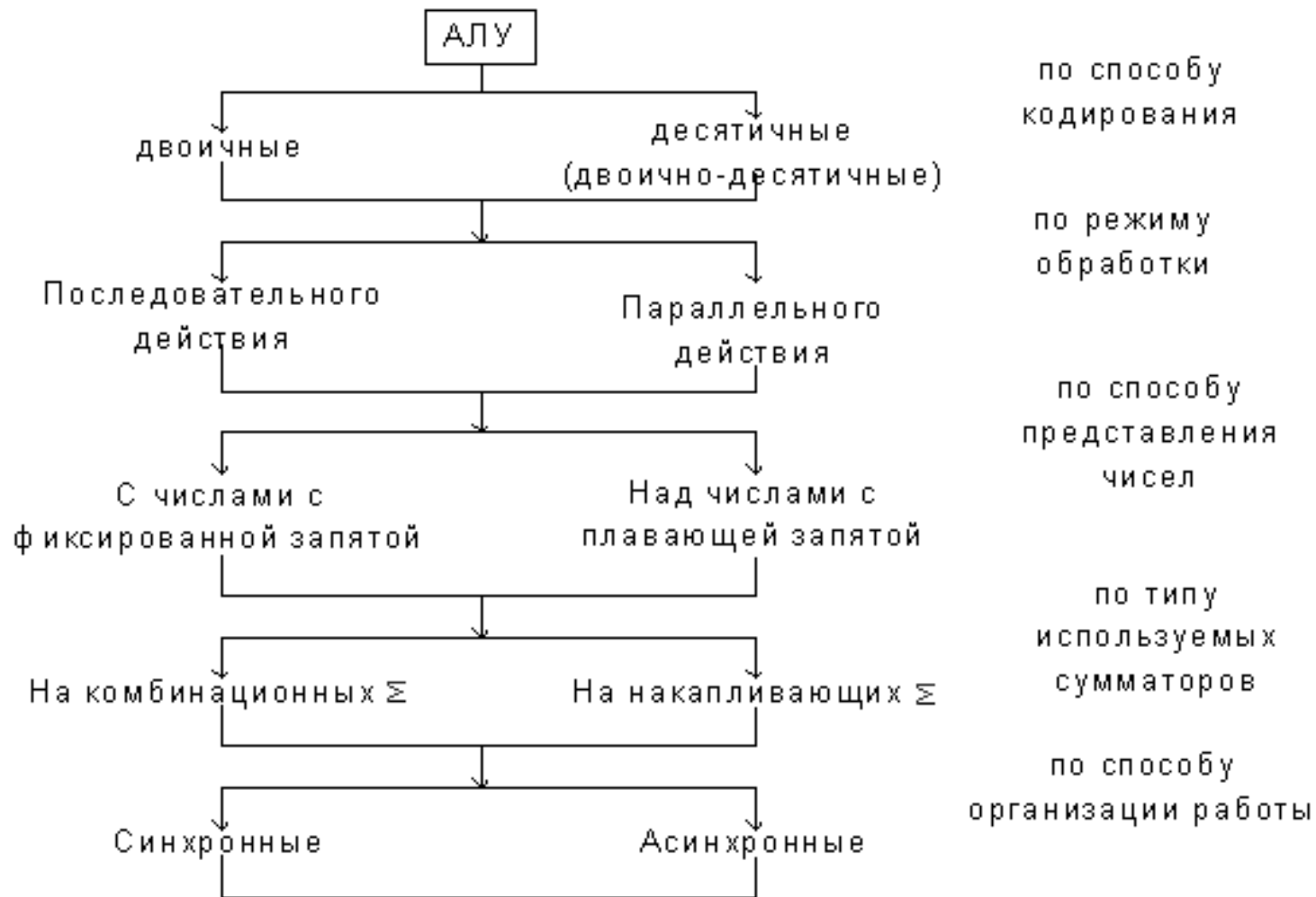
АЛУ и совокупность регистров составляют центральное операционное устройство и, совместно с УУ, образует основную часть процессора.

В состав операций, реализуемых АЛУ, обычно входят:

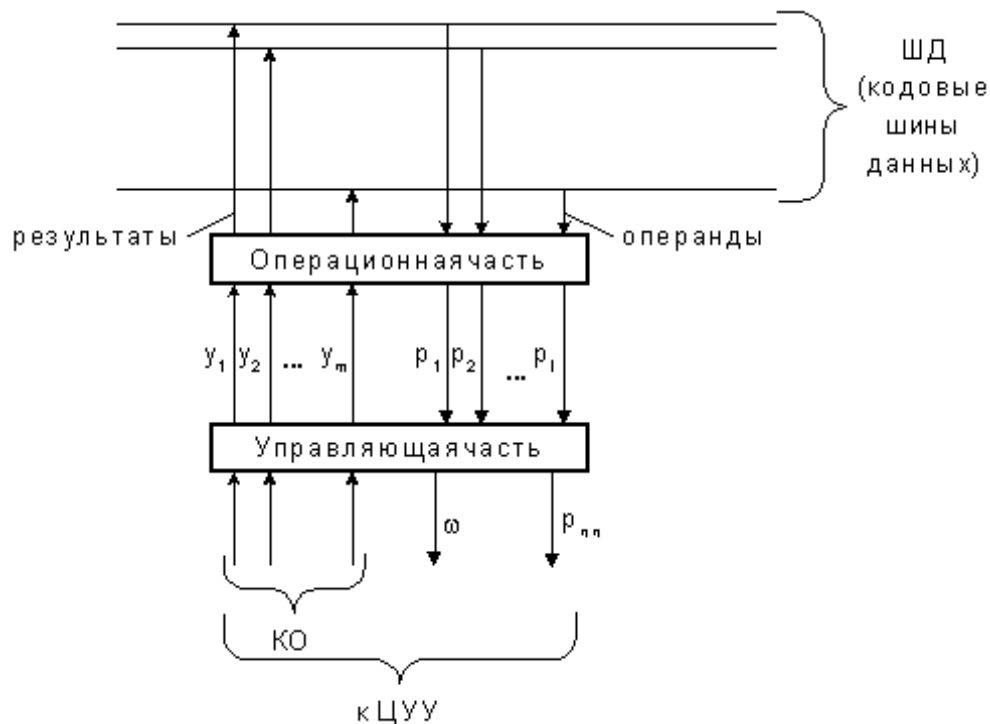
- арифметические операции (сложение, вычитание, умножение, деление, вычитание модулей, сравнение кодов на равенство и т.п.);
- логические (побитовые «И», «ИЛИ», «НЕ», «Сложение по модулю 2», именуемое иногда «Исключающее ИЛИ», и, реже, другие сходные операции над многоразрядными кодами);
- операции специальной, так называемой «машинной», арифметики (сдвиги: циклический, арифметический и логический; нормализация; счет в двоично-десятичных кодах, с удвоенной точностью и др.);
- операции над кодами как над строками, символьной информацией, текстами;
- дополнительные, специальные преобразования числовых и нечисловых кодов (например, десятичная коррекция, перестановка бит адреса и т.д.).

Современный подход – применение наряду с основными АЛУ битовых процессоров и других операционных устройств.

Условно-схематично введём классификацию АЛУ:



Обобщенную структуру АЛУ, как операционного устройства, и систему его связей с другими устройствами можно представить себе так:



Управляющая часть определяет порядок работы отдельных узлов АЛУ в зависимости от кода операции (КО), поступающего из ЦУУ.

На рисунке: $\{y_i\}$ – сигналы управления; далее будем их именовать управляющими функциональными сигналами (УФС); $\{p_j\}$ – осведомительные сигналы; ω – сигнал окончания операции; p_{nn} – сигнал (признак) переполнения разрядной сетки.

Организация АЛУ параллельного действия при работе над числами в естественной форме

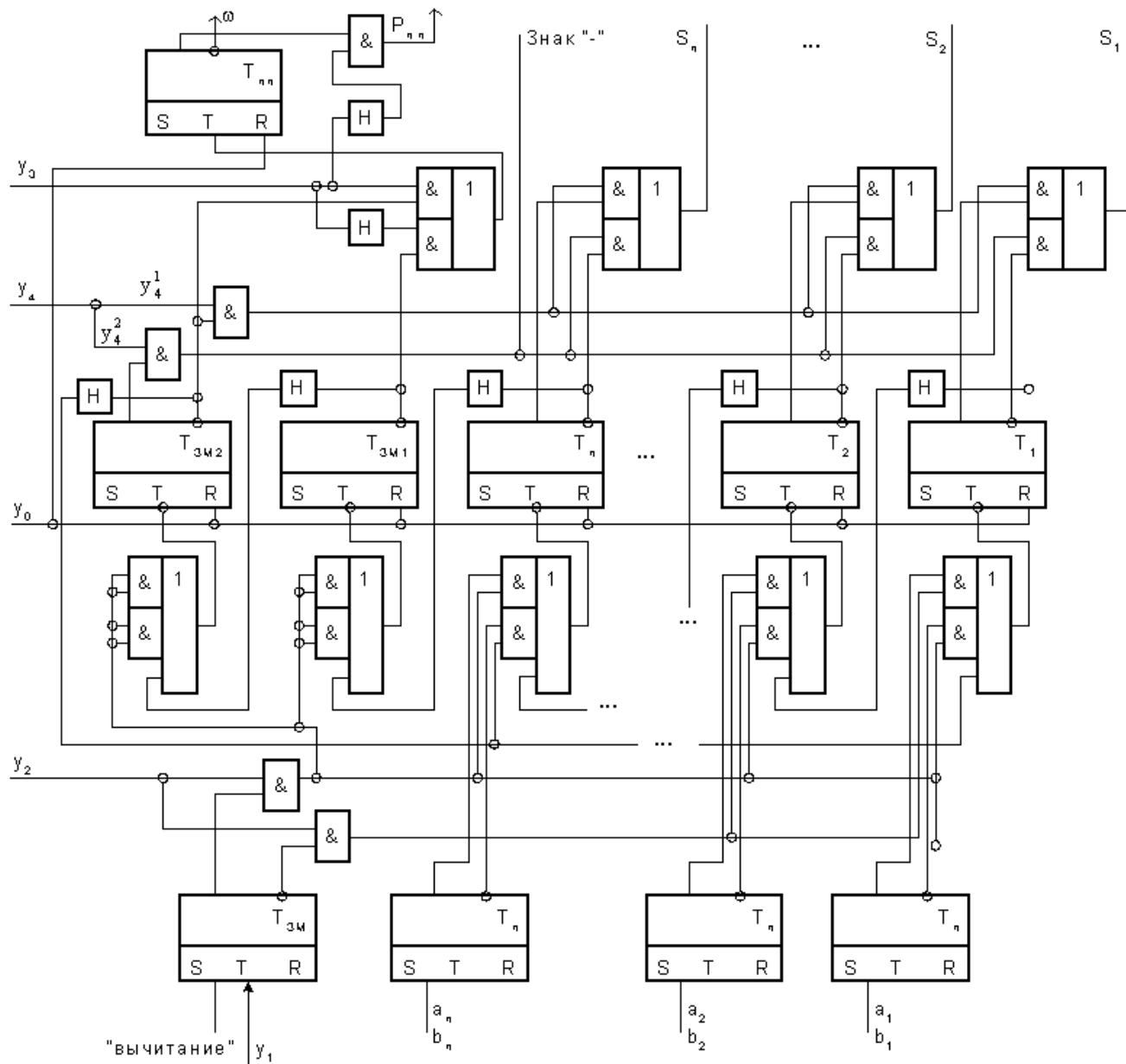
Суммирование и вычитание чисел при использовании накапливающего сумматора.

Будем полагать в этом разделе

- хранение чисел в ОЗУ осуществляется в прямом коде;
- коды модифицированные;
- в АЛУ при суммировании используется модифицированный обратный код.

Выполняется операция $S(A)=S(A)+B$.

Схема АЛУ на накапливающем сумматоре приведена ниже. Работу схемы легко уяснить по рисунку.



О времени выполнения операции:

$$T_{\text{смакс}} = t + (n + 2)(\tau_{\Pi} + \tau_{\Gamma}) + \tau_{\Gamma} ,$$

где t – интервал подачи слагаемых, τ_{Π} – задержка импульса переноса, τ_{Γ} – переходные процессы в триггере сумматора. Если предположить, что, $\tau_{\Pi} \gg \tau_{\Gamma}$ то при $\tau_{\Pi} \approx t$ получаем: $T_{\text{смакс}} \approx (n + 3)t$. Столь огромное время выполнения суммирования связано только с применением простейшей схемы последовательного переноса.

Для повышения быстродействия АЛУ применяют различные способы ускорения распространения волны переносов. Наименования разных типов переносов (схем распространения волны переносов): последовательный, параллельный, групповой, сквозной.

Суть этих схем переносов легко уяснить по логическим формулам, описывающим зависимость разрядов суммы и переносов между разрядами, где логическими аргументами являются бинарные значения разрядов слагаемых. (См., например, приложение ПЗ.1.)

Логические формулы признаков переноса и гарантированного отсутствия переноса в i -ом разряде:

$$p_i = x_i \cdot y_i \vee (x_i \vee y_i) p_{i-1}$$

$$q_i = \bar{x}_i \cdot \bar{y}_i \vee (\bar{x}_i \vee \bar{y}_i) \overline{p_{i-1}}$$

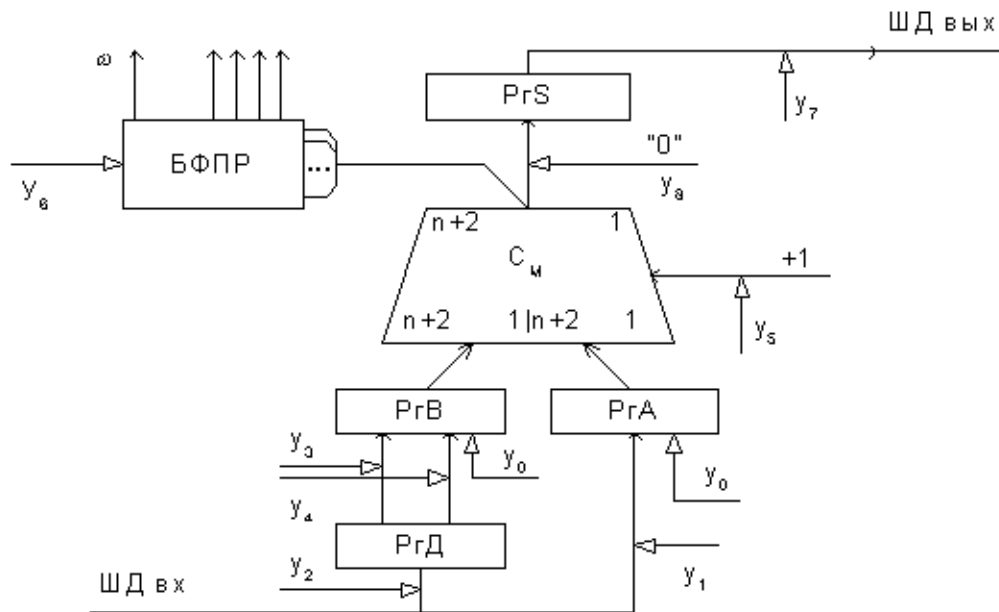
$w_i = p_i \vee q_i$ - сигнал «перенос из i -го разряда закончен»;

$$W = \bigwedge_{i=1}^n w_i \quad \text{– признак окончания переноса.}$$

Принципы построения АЛУ для сложения и вычитания на комбинационных сумматорах

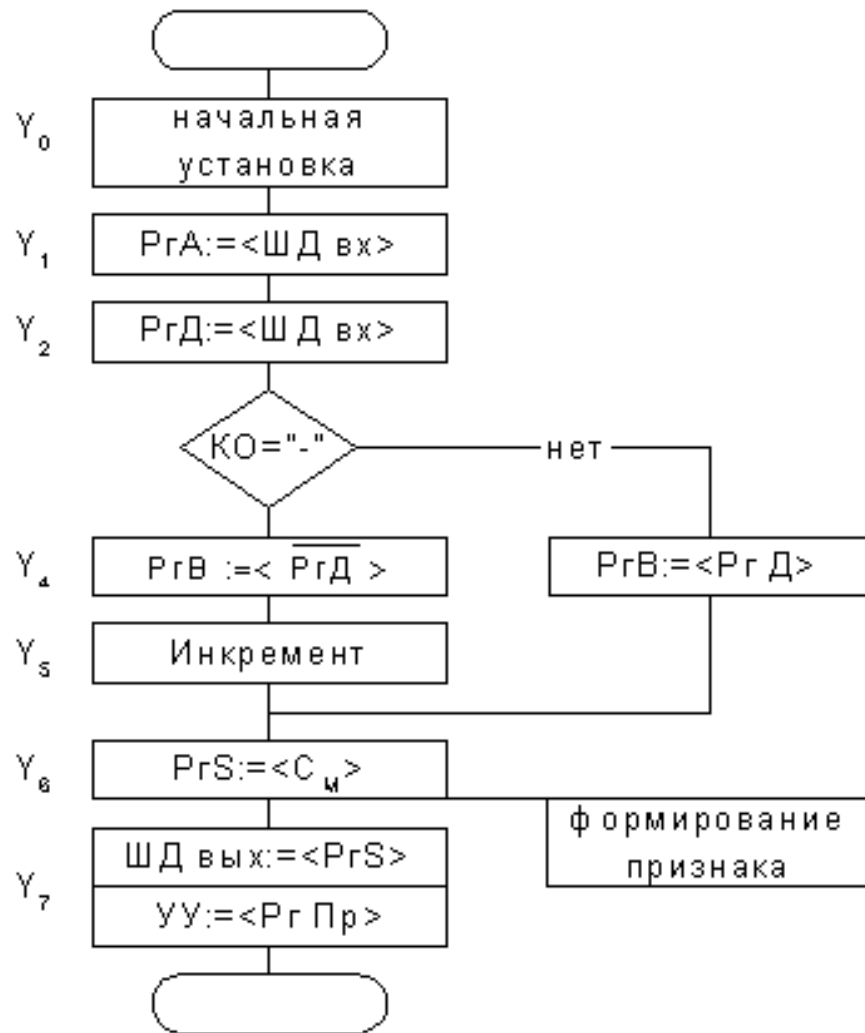
Для определённости предполагается::

- *запятая* располагается *справа* от младшего (правого) разряда чисел, то есть все числа полагаются целыми;
- используется *дополнительный модифицированный* код как в АЛУ, так и в ОЗУ;
- числа $n+2$ – разрядные ($n+1$ –й и $n+2$ –ой разряды отведены под знак).



Работа очень легко может быть пояснена с помощью структурной схемы алгоритма:

На рисунке структурной схемы обозначен БФПР – блок формирования признаков результата. Он состоит из комбинационной схемы и регистра признаков (РгПр). Кроме сигнала завершения операции наиболее распространены следующие признаки: переполнение, обнуление, знак результата и «чётность».



Организация АЛУ (параллельного действия) в режиме умножения чисел с фиксированной запятой

Умножение двоичных чисел с фиксированной запятой сводится к последовательности сложений и сдвигов: произведение суть сумма частичных произведений, каждое из которых получается умножением множимого на один разряд множителя.

Произведение двух n -разрядных чисел составляет до $2n$ разрядов:

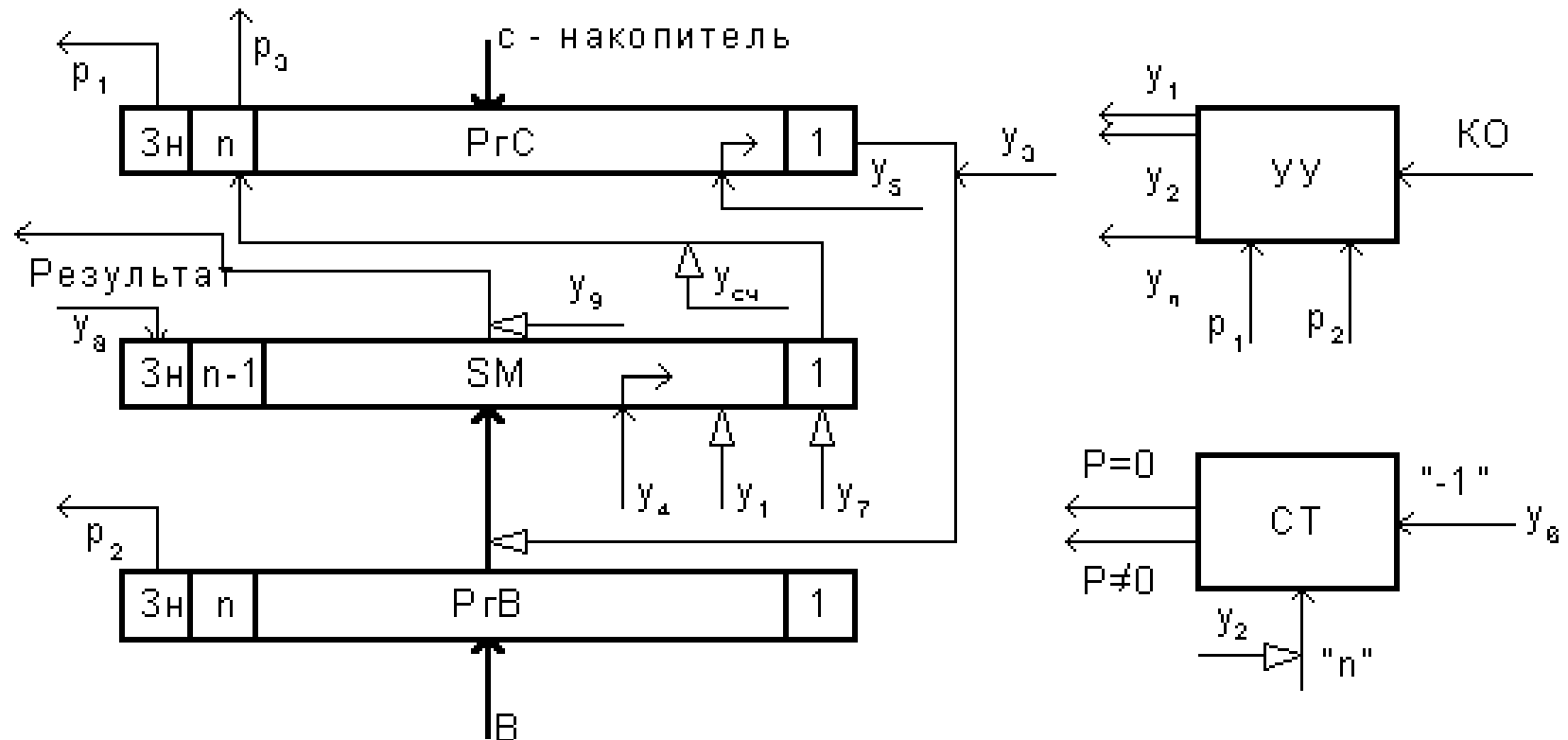
$$n = \log_2(2^n - 1)$$

$$\log_2(2^n - 1)^2 \approx 2n$$

Известно, что в зависимости от порядка просмотра множителя и способа формирования суммы частичных произведений различают четыре способа организации умножения. (См. приложение ПЗ.2.)

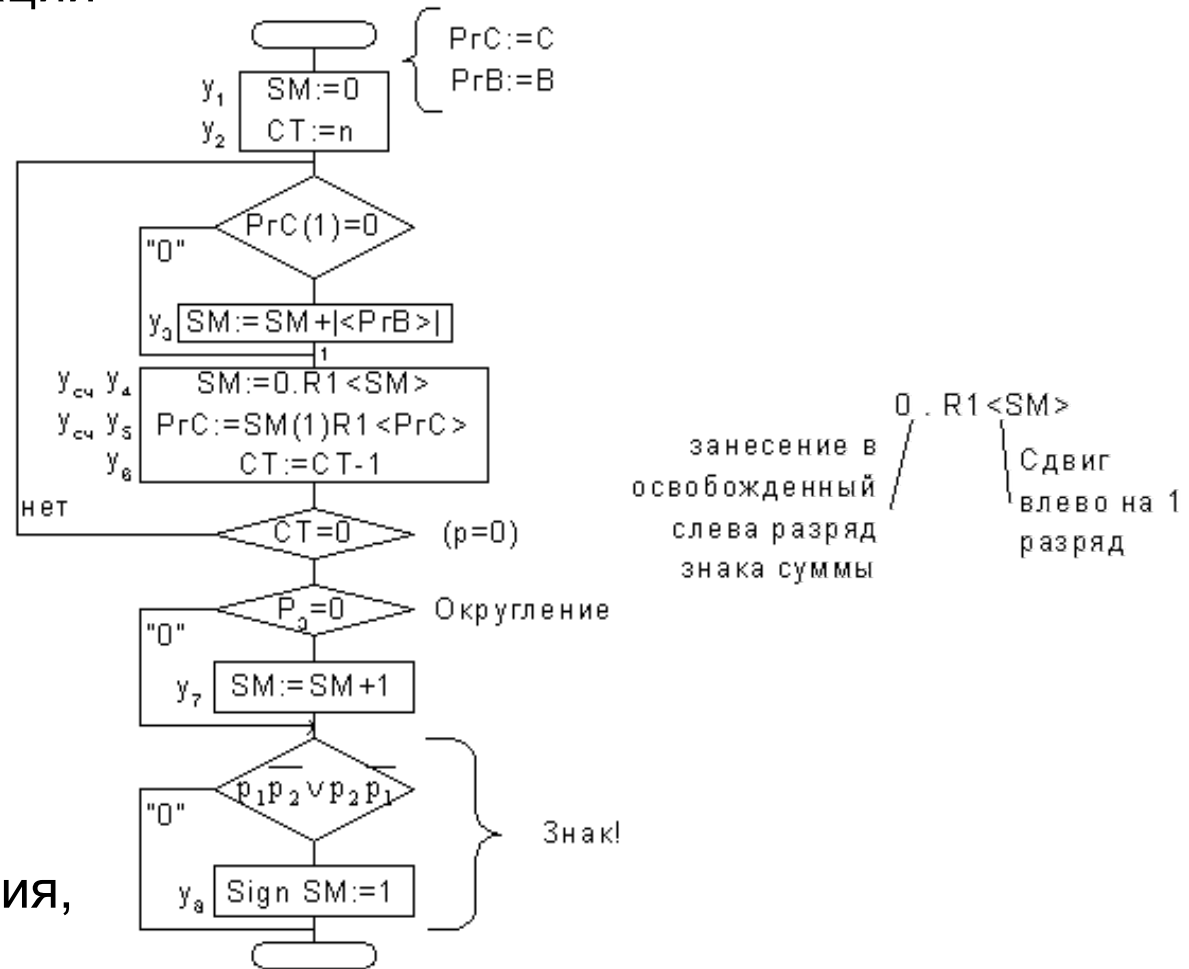
В приложении способы рассмотрены с точки зрения их размещения в регистрах АЛУ.

Основные элементы структурной схемы АЛУ на базе накапливающего сумматора:



Основное внимание следует сосредоточить на умножении модулей чисел.

Микропрограмму операции умножения представим структурной схемой (предполагается, что числа хранятся в ОЗУ в прямом коде).



$$T_{у\max} = n(T_c + t_{сдв.})$$

где

T_c – время суммирования,

$t_{сдв}$ – время сдвига.

В приложении П3.3 рассмотрены два аппаратных и два программных способа ускорения операции умножения в АЛУ.

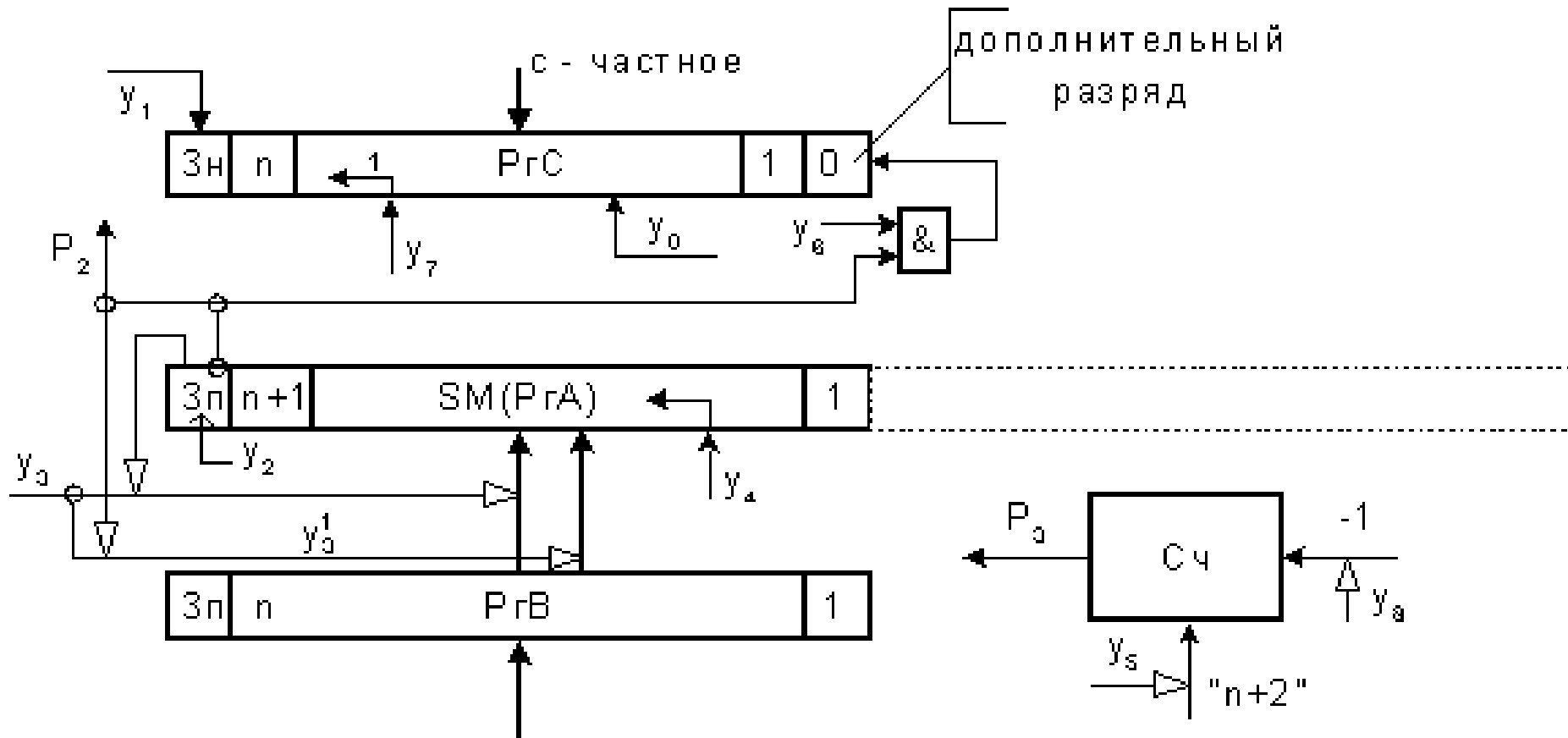
Организация АЛУ параллельного действия в режиме деления чисел с фиксированной запятой

Деление сводится к последовательности вычитаний делителя сначала из делимого (1-ый шаг), а затем из образующихся в процессе деления и сдвигаемых влево частичных остатков.

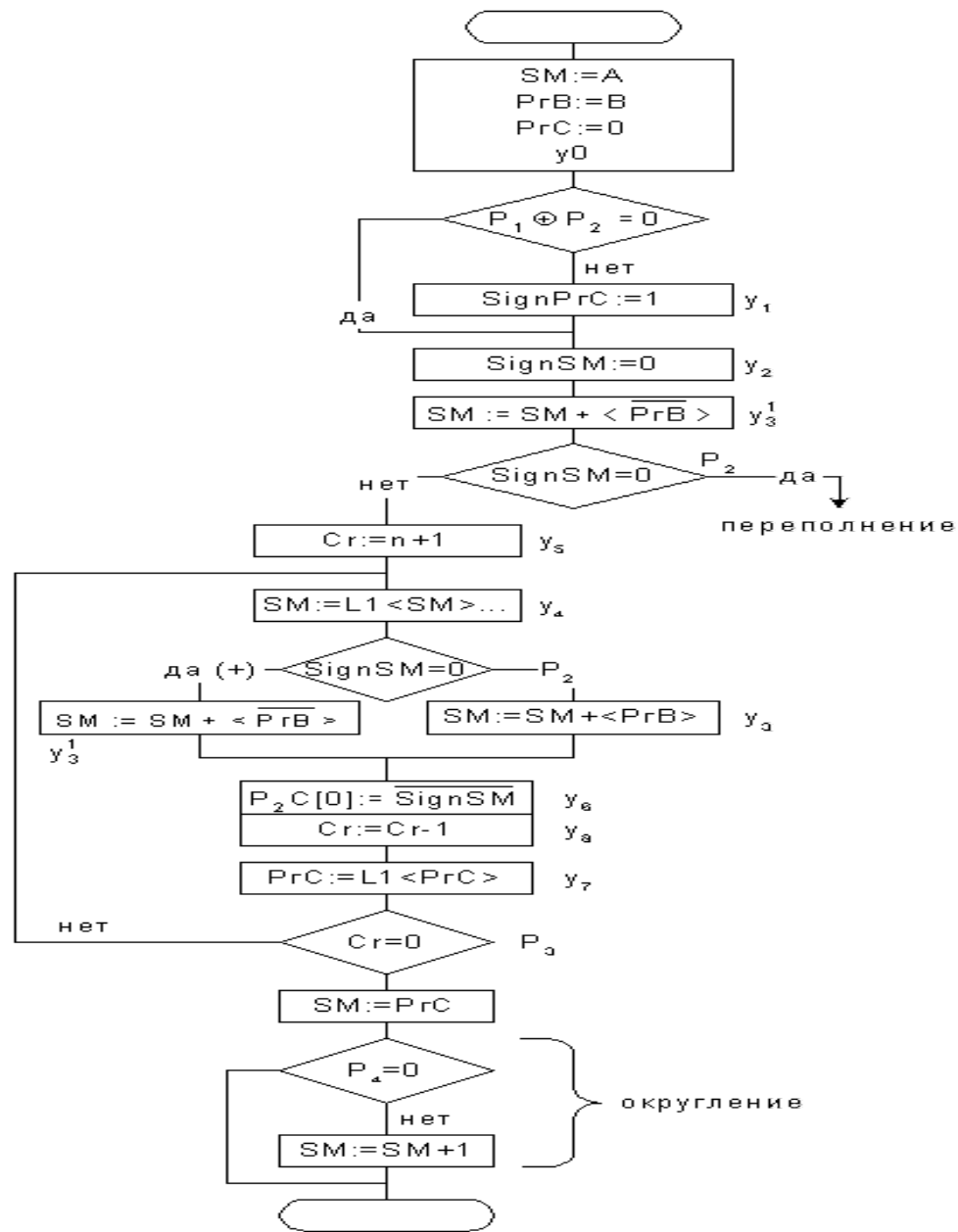
Первый шаг именуется иногда пробным делением. Смысл его предельно прост: надо, чтобы частное помещалось в разрядную сетку. Следовательно, если предположить, что в ЭВМ числа меньше 1 (так фиксирована запятая!), то делимое по модулю должно быть меньше делителя.

Нужно вспомнить, что в обычном алгоритме деления требуется восстановление остатка. Но существует и алгоритм деления без восстановления остатка.

Элементы структурной схемы для организации деления:



Структурная схема,
описывающая
микропрограмму
операции
деления без
восстановления
остатка (для
модулей чисел):



Использование АЛУ при реализации логических функций

Способов вычисления (реализации) логических функций в компьютере достаточно много:

- прямое использование многоразрядных логических (поразрядных) операций;
- применение битовых процессоров;
- непосредственное хранение значений (хранение таблиц истинности);
- бинарные программы;
- применение арифметических функционалов и др.

Суть способов и сравнение их эффективности можно достаточно ясно рассмотреть на примере.

Предположим, что необходимо в ЭВМ реализовать две переключательных функции (ПФ), то есть выполнить вычисление их значений (сформировать эти значения) на получаемых входных наборах аргументов. Исходно эти ПФ могут быть описаны тем или иным способом, в частности приводимой здесь таблицей истинности.

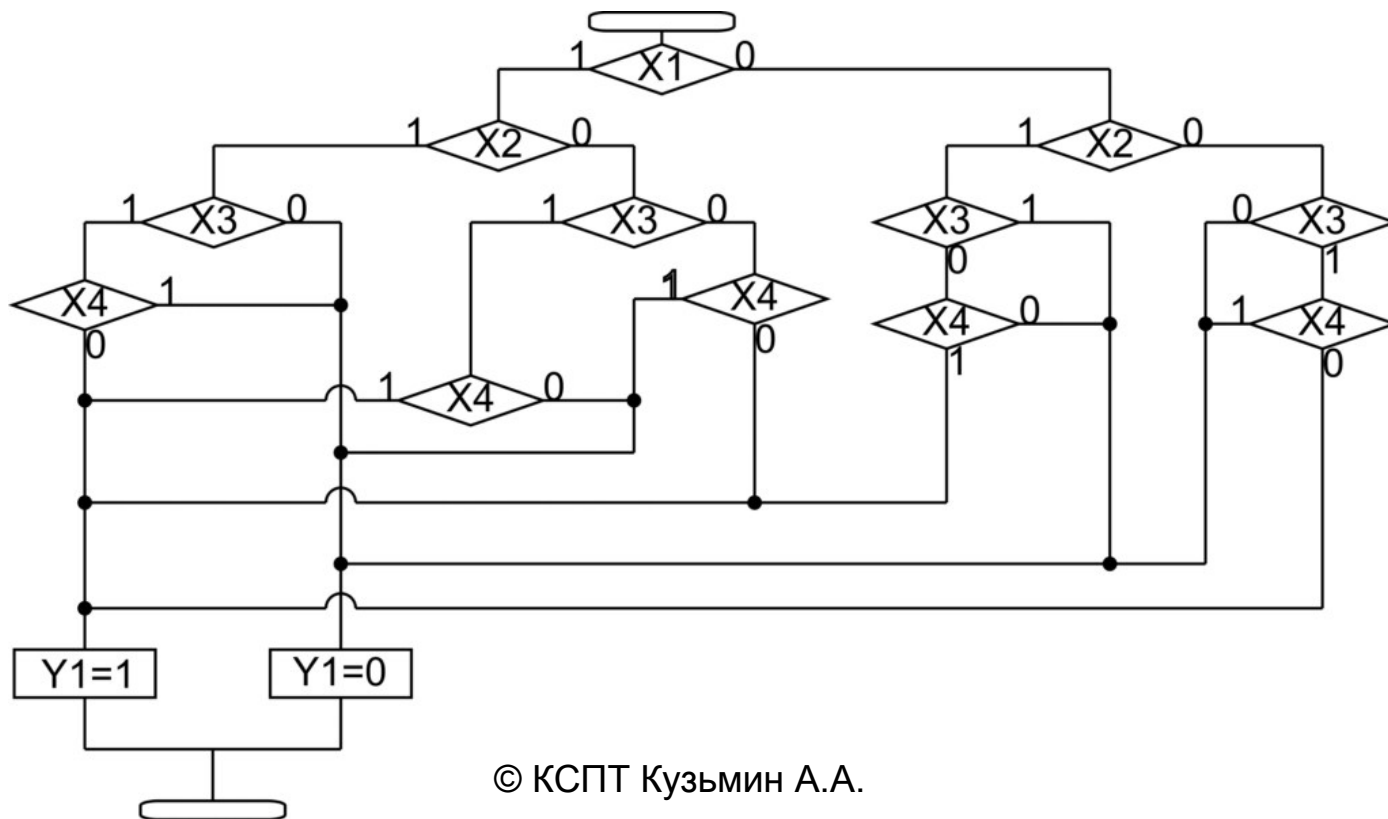
x_4	x_3	x_2	x_1	f_2	f_1
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	0
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	0	0
1	0	1	0	0	1
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	0	1
1	1	1	0	1	0
1	1	1	1	0	0

Несложно записать СДНФ для этих двух переключательных функций:

$$f_1 = \overline{x_1} \overline{x_2} \overline{x_3} \overline{x_4} \vee \overline{x_1} \overline{x_2} x_3 \overline{x_4} \vee \overline{x_1} x_2 \overline{x_3} \overline{x_4} \vee \overline{x_1} x_2 x_3 \overline{x_4} \vee \overline{x_1} x_2 x_3 x_4$$

$$f_2 = \overline{x_1} \overline{x_2} \overline{x_3} \overline{x_4} \vee \overline{x_1} \overline{x_2} x_3 \overline{x_4} \vee \overline{x_1} x_2 \overline{x_3} \overline{x_4} \vee \overline{x_1} x_2 x_3 \overline{x_4} \vee \overline{x_1} x_2 x_3 x_4$$

А так будет выглядеть бинарный граф для функции f_1 :



Чтобы реализовать те же переключательные функции в АЛУ достаточно обеспечить вычисление следующего линейного полинома:
 $F(X) = 5x_1 + 11x_2 + 5x_3 + 11x_4$,
а затем выявить 3-й и 4-й разряды двоичного результата.

x_4	x_3	x_2	x_1	$F(X)$	r6	r5	r4	r3	r2	r1	
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	5	0	0	0	1	0	1	
0	0	1	0	11	0	0	1	0	1	1	
0	0	1	1	16	0	1	0	0	0	0	
0	1	0	0	5	0	0	0	1	0	1	
0	1	0	1	10	0	0	1	0	1	0	
0	1	1	0	16	0	1	0	0	0	0	
0	1	1	1	21	0	1	0	1	0	1	
1	0	0	0	11	0	0	1	0	1	1	
1	0	0	1	16	0	1	0	0	0	0	
1	0	1	0	22	0	1	0	1	1	0	
1	0	1	1	27	0	1	1	0	1	1	
1	1	0	0	16	0	1	0	0	0	0	
1	1	0	1	21	0	1	0	1	0	1	
1	1	1	0	27	0	1	1	0	1	1	
1	1	1	1	32	1	0	0	0	0	0	

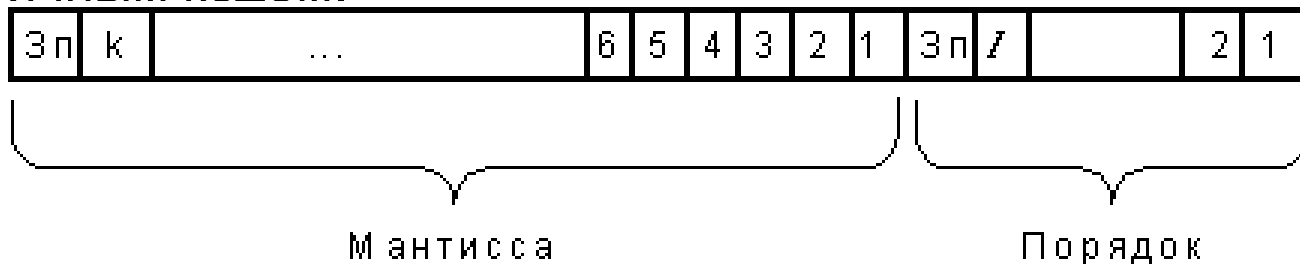
Организация АЛУ параллельного действия при работе над числами в нормальной форме

Следует напомнить структуру записи числа в нормальной форме:

$$A = \pm M_A \cdot d^{\pm p_a},$$

где – M_A мантисса (остальное в формуле – характеристика); p_a – порядок; d – основание характеристики.

Обычно $d=2$, но может быть и 8, и 16 (было в ЕС ЭВМ, ныне редко встречается). В любом случае величина d в разрядной сетке не указывается (а подразумевается!), а M_A и p_a представляются двоичным кодом.

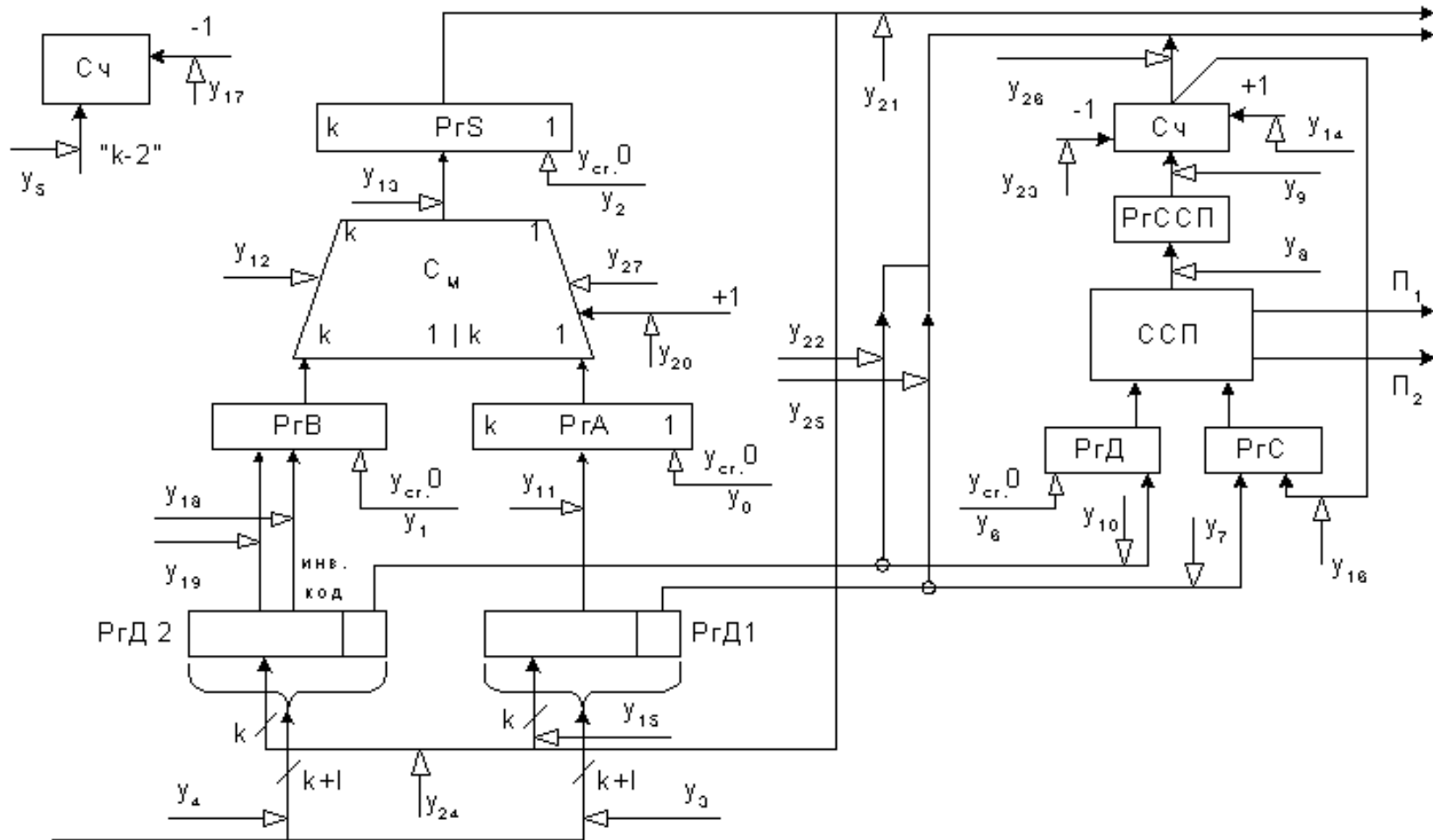


$$k = \log_2 |M_a| \quad ; \quad l = \log_2 |p_a|$$

Число должно быть нормализовано, т.е. при $d=2$: $0,5 \leq M_a < 1$; а при

$$d=8: \quad \frac{1}{8} \leq M_a \leq 1 .$$

Принципы организации и работы АЛУ при суммировании/вычитании чисел в нормальной форме



Коротко об алгоритме в целом. В нем выделяются 4 этапа:

- Выравнивание порядков чисел (порядок меньшего числа увеличивается до порядка большего при одновременном сдвиге в разрядах меньшего вправо на число разрядов, равное разности порядков).
- Собственно сложение (вычитание) мантисс с фиксированной запятой.
- Проверка переполнения и присвоение порядка.
- Нормализация.

(Третий и четвёртый этапы можно переставить).

Наиболее сложной (длинной, «медленной») процедурой является выравнивание порядков.

Структурная схема алгоритма ввиду громоздкости вынесена в приложение ПЗ.4.

Организация АЛУ, работающих в двоично-десятичных кодах

Простые двоичные коды обычно вполне устраивают пользователей универсальных и персональных ЭВМ. Но иногда применяются специальные коды: помехозащищенные, двоично-десятичные и др.

Особенности применения специальных кодов достаточно ярко проявляются при двоично-десятичном кодировании. Простейшим можно считать обычный двоично-десятичный код (код с весами разрядов: 8, 4, 2, 1), но есть и другие, имеющие свои довольно интересные свойства. Ниже дан комментарий к использованию некоторых двоично-десятичных кодов.

Эта таблица была опубликована в Главе 1 (слайд 9)

Деся- тич- ные циф- ры	Код «8,4, 2,1»	Код «2,4, 2,1»	Код «+3»	Код «7,4, 2,1»	Код «5,4, 2,1»	Код ω, x, y, z	Код ABCD	Реф- лекс- ный код	Код «2 из 5»	Двоично- пятиричный код
0	0000	0000	0011	0000	0000	0000	0000	0000	11000	0100001
1	0001	0001	0100	0001	0001	0001	0110	0001	00011	0100010
2	0010	0010	0101	0010	0010	0011	0011	0011	00101	0100100
3	0011	0011	0110	0011	0011	1000	0111	0010	00110	0101000
4	0100	0100	0111	0100	0100	0110	1111	0110	01001	0110000
5	0101	1011	1000	0101	1000	1111	0101	0111	01010	1000001
6	0110	1100	1001	0110	1001	1001	1101	0101	01100	1000010
7	0111	1101	1010	1000	1010	0010	1001	0100	10001	1000100
8	1000	1110	1011	1001	1011	1100	1100	1100	10010	1001000
9	1001	1111	1100	1010	1100	0100	1010	1000	10100	1010000

Коды «2,4,2,1» и «+3» относятся к группе самодополняющихся.

Код «7,4,2,1» использует в кодовых комбинациях минимальное число «единиц».

Коды ω, x, y, z и «Рефлексный» применяются в специальных алгоритмах обработки информации.

Код **ABCD** обладает особыми «надёжностными» свойствами, как и код «2 из 5» и двоично-пятиричный код.

Если каждая десятичная цифра кодируется двоичным кодом (например, четырехразрядным). АУ может быть организовано:

- 1) на десятичных сумматорах (по одному на каждую десятичную цифру);
- 2) на базе обычных двоичных сумматоров, а приспособление к десятичной арифметике выполняется алгоритмически или через специальные корректирующие устройства.

В варианте (1) структура SM превращается в схему:

