Ускоренное умножение

$$t_{умн} = \sum_{i=1}^{n-1} (t_{сдв} + p_i t_{сл}),$$

Методы ускорения умножения можно условно разделить на --

- аппаратные
- логические

Алгоритм Мак-Сорли с младших разрядов множителя с обработкой двух разрядов множителя за такт

Триггер	$Y_{i,}\ Y_{i+1}$	Знак арифметич. действия	Кратные множимого
0	**00**	Нет действий	
0	**01**	+	**1A**
0	**10**	+	**2A**
0	**11**	**-**	**1A***
1	**00**	+	**1A**
1	**01**	+	**2A**
1	**10**	**-**	**1A***
1	**11**	Нет действий	

9 *(-9)= - 81

00.00000 00000 11.01<u>11</u>

<u>11.0111 т=0, -1А</u>

11.0111

11.110111 модиф. сдвиг

<u>01.0010 т=1, +2А</u>

00.1111110000

00.0011111100 обычный сдвиг

<u>**11**.0111 т=0 -1А</u>

11.1010111100

11 1110101111- модиф. сдвиг

Триггер	$Y_{i,}$ Y_{i+1}	Знак арифметич. действия	Кратные множимого
0	**00**	Нет действий	
0	**01**	+	**1А**
0	**10**	+	**2А**
0	**11**	-	**1А***
1	**00**	+	**1А**
1	**01**	+	**2А**
1	**10**	-	**1А***
1	**11**	Нет действий	

ЭВМ

Алгоритм Мак-Сорли со старших разрядов множителя с обработкой двух разрядов множителя за такт

Y_i	Y_{i+1} Y_{i+2}	Арифмет. действия	Кратные множимого
0	00	Нет действий	
0	01	+	2A
0	10	+	2A
0	11	+	4A
1	00	-	4A
1	01	-	2A
1	10	-	2A
1	11	Нет действий	

Пример 9*9=81

00.00000 00000   **0.1001**

00 00000 10010 +2A

0000000 10010

00.00010 01000сдвиг

0 0000 00 10010+2A

00.00010 11010

11.111 11 10111коррекция -1A

00.00010 10001

Y_i	Y_{i+1} Y_{i+2}	Арифмет. действия	Кратные множимого
0	00	Нет действий	
0	01	+	2A
0	10	+	2A
0	11	+	4A
1	00	-	4A
1	01	-	2A
1	10	-	2A
1	11	Нет действий	

0.1001

Алгоритм Лемана

$$\sum_{i=k}^{n} 2^i = 2^n - 2^k$$

$$0111 == 100\bar{1}$$

На каждом шаге умножения выполняются действия, которые можно описать с помощью логических выражений :

$$\alpha_i = (b_i \oplus b_{i-1})\bar{\alpha}_{i-1}$$

$$S_i = \alpha_i b_{i+1} \vee S_{i-1}\bar{\alpha}_i$$

α_i - двоичная переменная, единичное значение которой указывает на необходимость выполнения арифметических операций,

b_i -разряды множителя,

S_i -определяет знак арифметической операции,

S_i =0, операция сложения,

S_i =1, операция вычитания.

Основные этапы алгоритма

1. Прием сомножителей, запоминание знаков, формирование модулей.
2. Анализ сомножителей на 0. Формирование знака результата. Установка значения СчЦ.
3. Анализ младшего разряда множителя. Если младший разряд множителя равен 0, то производится сдвиг до появления первой 1. Если младший разряд множителя равен 1, то выполняется вычитание, если следующий разряд 1 и сложение, если следующий разряд 0.
4. Сдвиг суммы ЧП вправо на один разряд прямой или модифицированный сдвиг, если промежуточная сумма частичных произведений отрицательна.. Пункты 3, 4 повторяются для всех цифровых разрядов множителя.
5. Формирование результата.

Пример: Разряды множителя 0111($b_3\, b_2\, b_1\, b_0$)

$$\alpha_0 = (b_0 \oplus b_{-1})\bar{\alpha}_{-1} = (1 \oplus 0)1 = 1$$

$$S_0 = \alpha_i b_{i+1} \vee S_{i-1}\bar{\alpha}_i = \alpha_0 b_1 \vee S_{-1}\bar{\alpha}_0 = 1\cdot 1 \vee 0 \cdot 0 = 1$$

10*7 01010 * 0.0111

Перевод в с.с. $10\bar{1}$ $0.0111 >> 0.100\bar{1}$

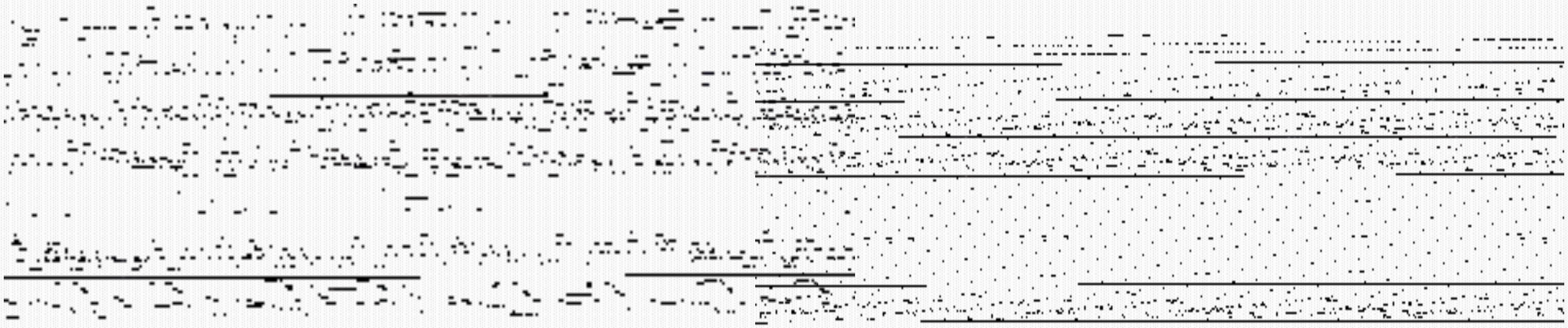
00000 00000 $0.100\bar{1}$
+10110
10110 00000
11011 00000
11110 11000
+01010
01000 11000
00100 01100
00010 00110====2+4+64

ПРИМЕР ПЕРЕВОДА В СИСТЕМУ $10\bar{1}$
101110=46

$1100\bar{1}0$ =46

$10\bar{1}00\bar{1}0$= 46

Матричные методы умножения



Каждый элемент a_i b_j (i, j = 1, n) принимает значение 0 или 1. Произведение A·B может быть получено, если суммировать элементы матрицы .

$$P = A \cdot B = \left(\sum_{i=0}^{n-1} a_i \cdot 2^i\right) \cdot \left(\sum_{j=0}^{n-1} b_j \cdot 2^j\right) = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} a_i b_j \cdot 2^{i+j}$$

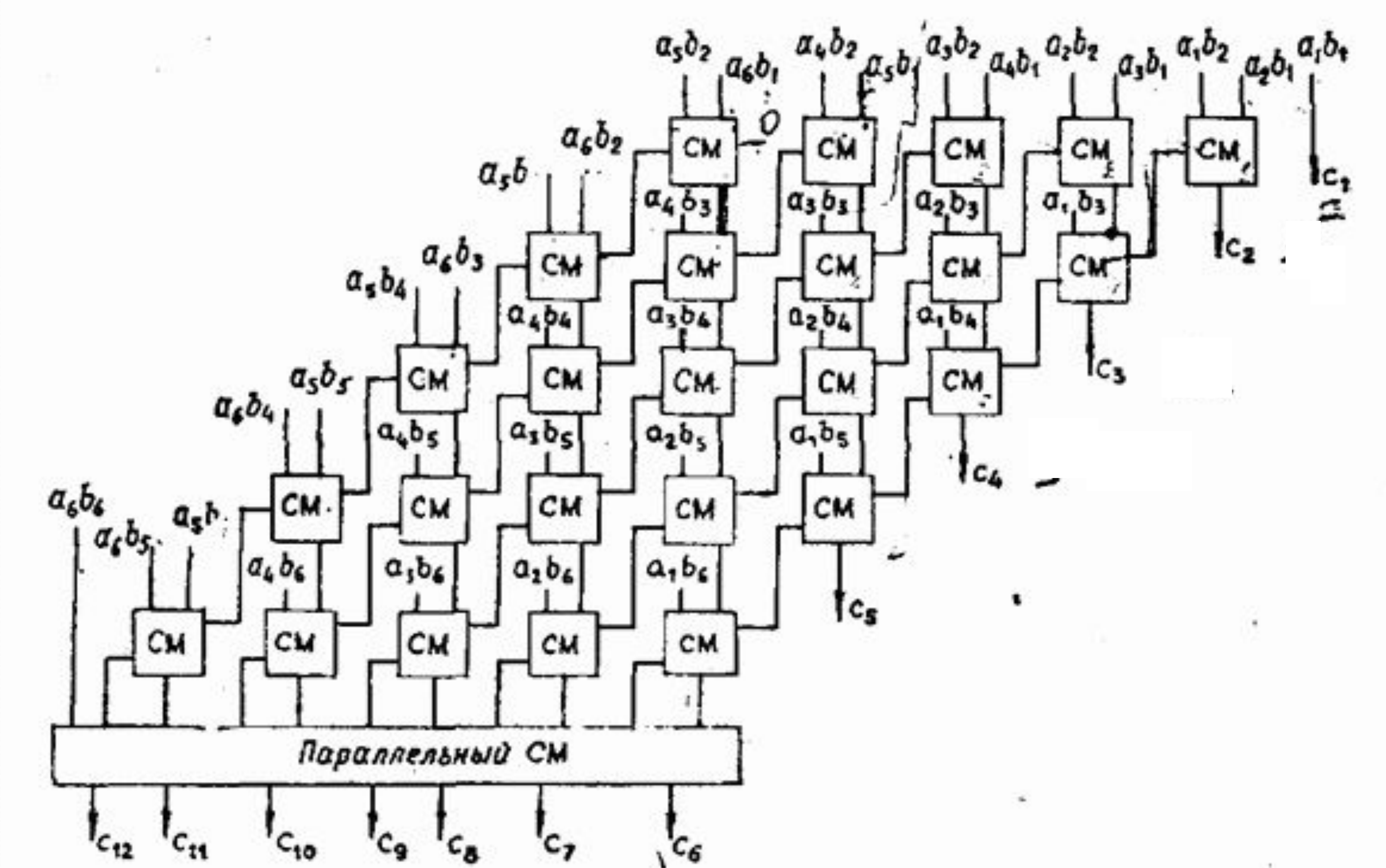


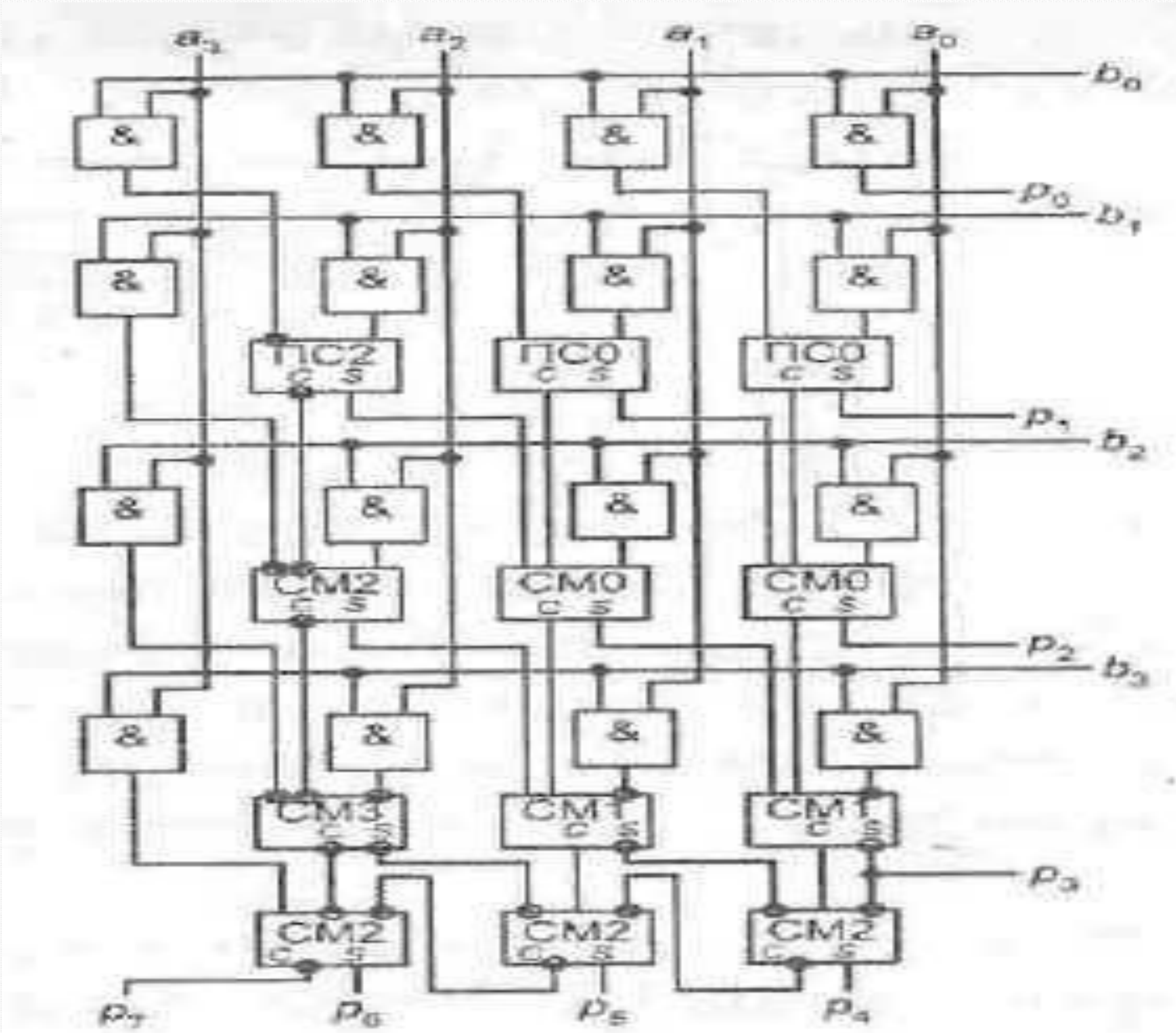
Схема устройства для реализации дерева спуска, реализующая матричный алгоритм умножения

Матричный умножитель п х п содержит $п^2$ схем «И», n ПС и ($п^2$ - 2п) СМ.

Если принять, что для реализации полусумматора требуются два логических элемента, а для полного сумматора — пять, то общее количество логических элементов в умножителе составляет

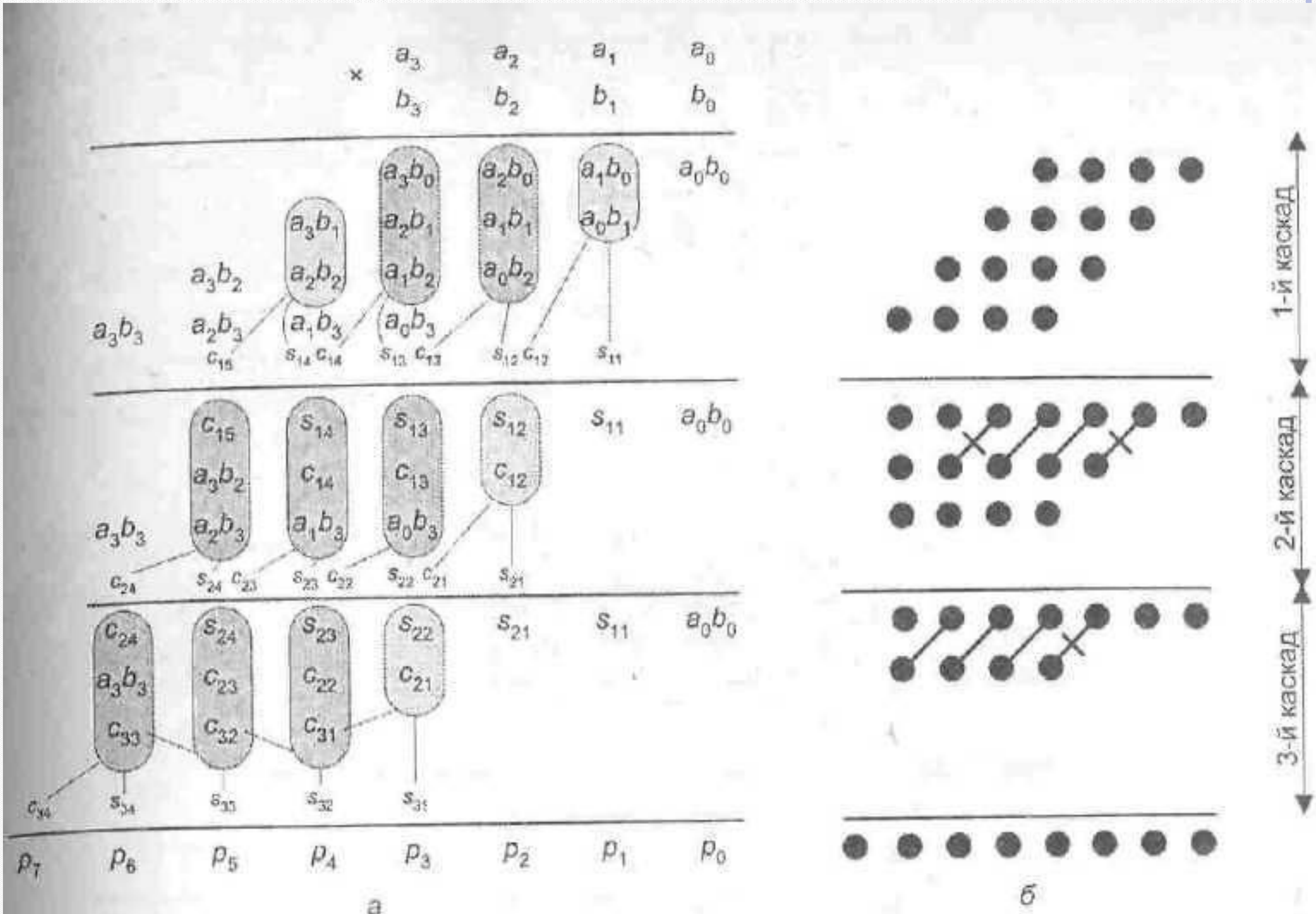
$п^2 + 2п + 5(п^2 – 2n) = 6n^2 – 8n$.

Полагая задержки в схеме «И» и полусумматоре равными θ, а в полном сумматоре — 2θ. общую задержку в умножителе можно оценить выражением *{4n* - 5)θ.


a_3
a_2
a_1
a_0
b_0
b_1
b_2
b_3
p_0
p_1
p_2
p_3
p_4
p_5
p_6
p_7
&
ПС2
ПС0
СМ2
СМ0
СМ3
СМ1
c
s

ЭВМ

Древовидные умножители включают в себя три ступени:
- ступень формирования битов частичных произведений, состоящую из n^2 элементов «И»;
- ступень сжатия частичных произведений — реализуется в виде дерева параллельных сумматоров (накопителей), служащего для сведения частичных произведений к вектору сумм и вектору переносов. Сжатие реализуется несколькими рядами сумматоров, причем каждый ряд вносит задержку, свойственна одному полному сумматору;
- ступень заключительного суммирования, где осуществляется сложение вектора сумм и вектора переносов с целью получения конечного результата.



Суммирование ЧП с помощью дерева Уоллеса : а — логика суммирования; б — точечная диаграмма

ЭВМ

Мак-Сорли со старших разрядов

0.01000 8 × 6

```
+000 0000000000
 000 0001 0000
----------------
 000001 0000
←00000100 0000
+1...11111 0000
----------------
 0...000 11 0000 * последний
                   сдвиг не выполняется
```

$\underbrace{0\,0\,1\,1\,0}$: +2A, −2A.

0.01000 +8

0.10000 +2A

1.10000 −2A.

0.0101 × 0.1101 5 × 13.

```
 0000000000
+0....0 0101 00
----------------
 0..00010100
←0000101 0000
+...11111 10110
----------------
 0.001 000110
+1.11111 1011 * коррекция
----------------
 0001 000001
```

$\underbrace{0.\,1\,1\,0\,1}$ A: +4A, −2A.

0.0101 +5

0.10100 +4A

110110 −2A.

11011 −A.

Мак - Сорли с младших разрядов

00.1001 000.111 9 x 7

−A

+2A

−A = 11.0111

+2A = 010010

00 0000 000000

+ 11.0111

110111

>> 111110111

\+ 0.10010

00111111

>> 0000111111100

>> 0000000111111

1111 00 −4

0001 000 +4

4 x 14 T = 0

+A. −A +2A

00 11 10

T = 1. T = 0 T = 0

000000 000000

\+ 001000

001000

>> 000010000000

\+ 111100

1111100000000

>> 111111100000

\+ 000100

0000111000000

>> 0000000111000

32 16 8

=> 56

Алгоритм Лемана

$5 \times 7 \qquad 0.101 \times \underbrace{0.111}$

$0.1\,00\bar{1} \leftarrow_{c.c.} 10\bar{1}$

00000.00000
+ 11011

1 1 0 1 1 00000
→ 1 1 1 0 1 1
1 1 1 1 1 01100
+ 0 0 1 0 1

0 0 1 0 0 0 1 1 0 0
→ 0 0 0 1 0 0 0 1 1 0
→ 0 0 0 0 1 0 0 0 1 1

0.00100

$0.1101 \qquad 4 \times 13$

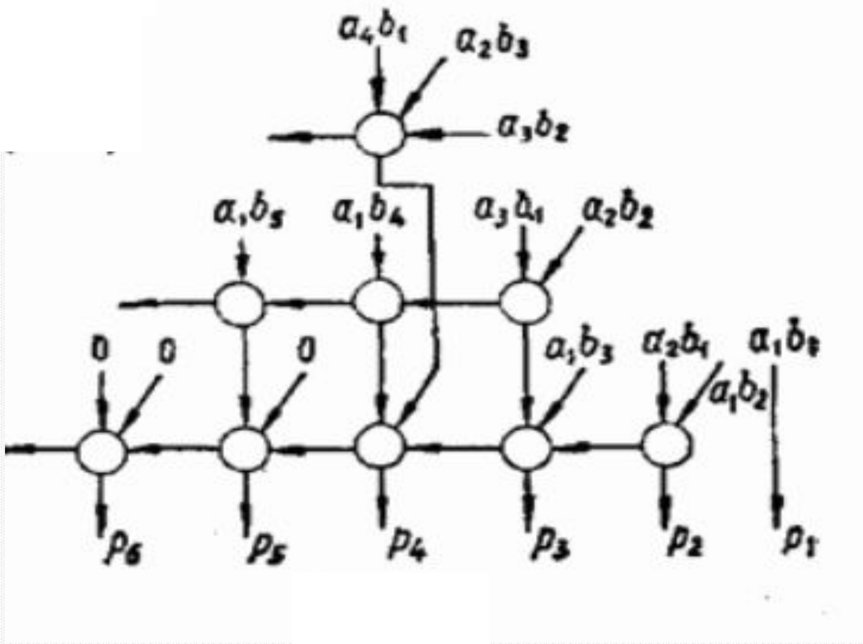
$0.10\bar{1}01$

000000 000000
+ 000100

000100 000000
→ 000010 00000
→ 000001 0 00...
+ 111100

111101 000000
→ 111110 100000
→ 111111 010000
+ 000100

Произведем распределение первичных и запаздывающих слагаемых



Для одноразрядных сумматоров время образования суммы больше, чем время образования переноса τ_n , поэтому наибольшее время спуска по дереву данного вида

$$T_y = (n-1)(\tau_{см} + \tau_п),$$

$$2^j \leqslant m \leqslant 2^{j+1} - 1,$$

где m — число слагаемых. Время спуска по этому дереву составит

$$\tau_{сп} = j\tau_{см}.$$

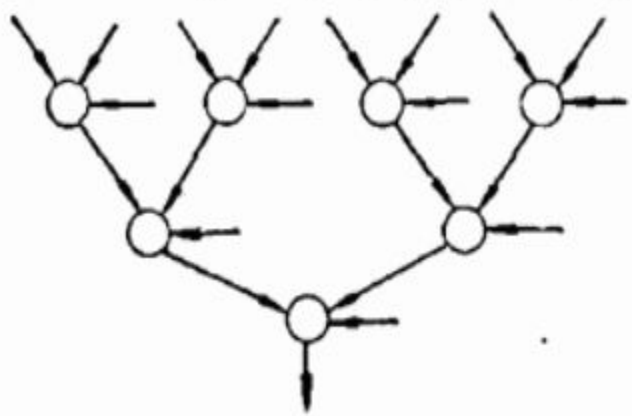


Рисунок 1

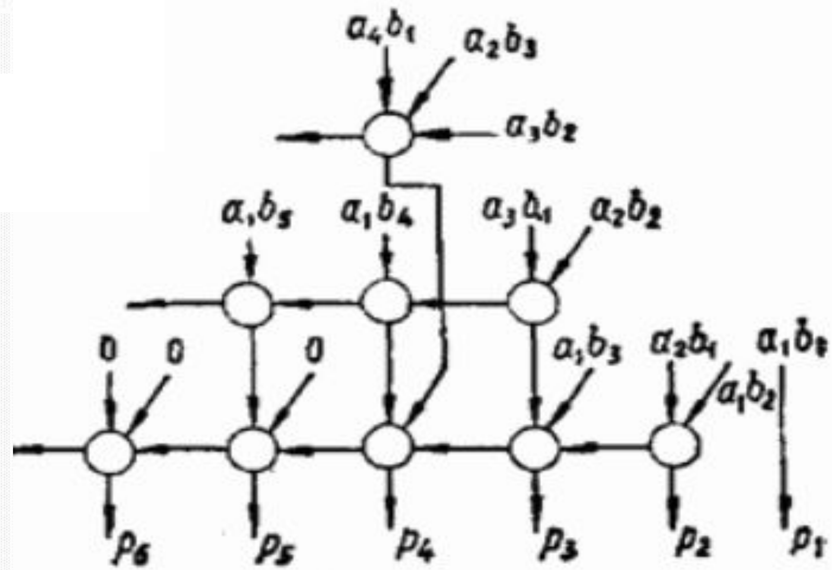
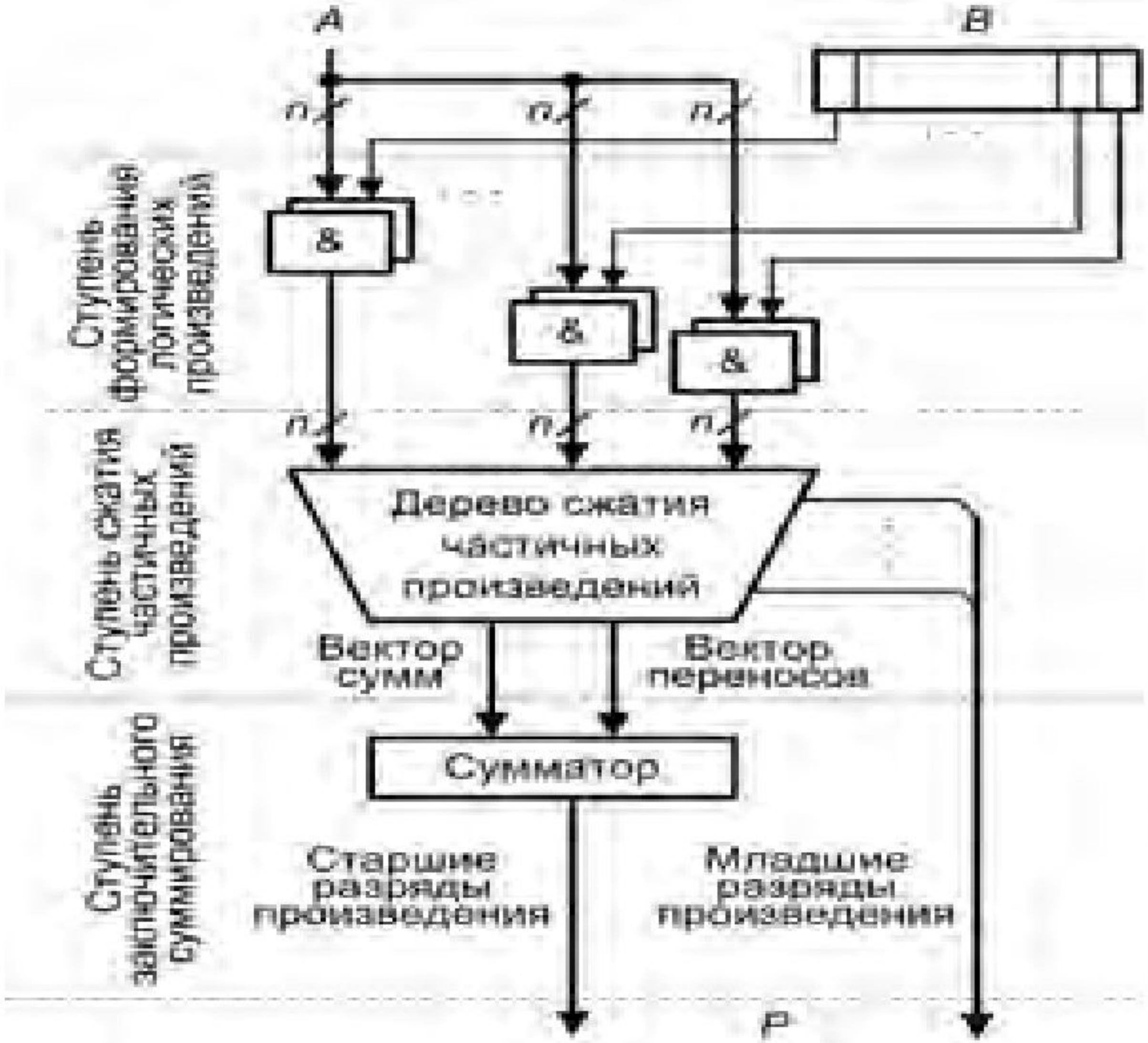
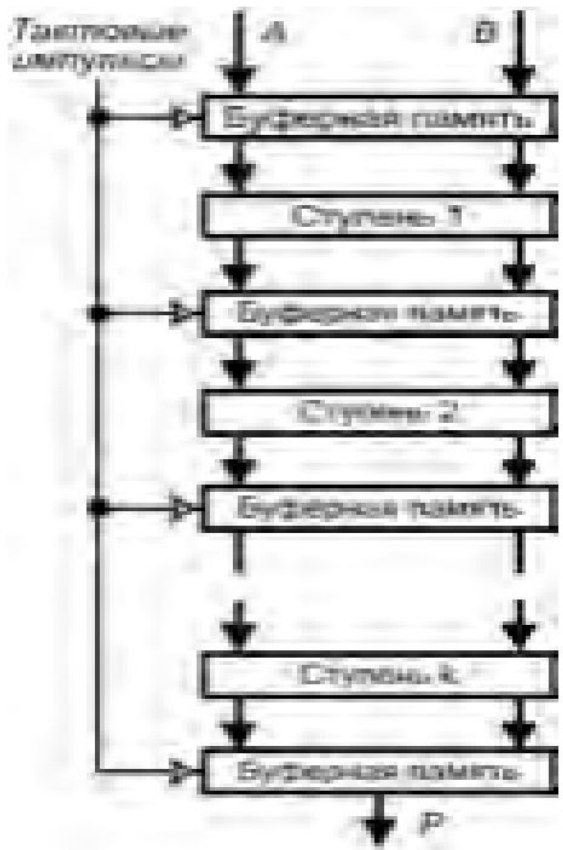


Рисунок 2, время спуска уменьшается до величины

$$T_y = 2[\tau_{см} + (n-2)\tau_п],$$



Структура древовидного умножителя



Структура конвейерного умножителя