

Тема 6. Продолжение: Элементы и узлы ЭВМ

Основные вопросы:

6.1. Классификация элементов и узлов ЭВМ

6.2. Комбинационные схемы

6.2.1. Дешифраторы и шифраторы

6.2.2. Компараторы и сумматоры

6.2.3. Сумматор параллельного действия

6.2.4. Сумматор с групповым переносом

6.1. Классификация элементов и узлов ЭВМ

При рассмотрении структуры любой ЭВМ обычно проводят ее детализацию. Как правило, в структуре ЭВМ выделяют следующие структурные единицы:

- **устройства,**
- **узлы,**
- **блоки и**
- **элементы.**

Такая детализация соответствует вполне определенным операциям преобразования информации, заложенным в программах пользователей.

Нижний уровень обработки реализуют *элементы*. Каждый элемент предназначен для обработки единичных электрических сигналов, соответствующих битам информации.

Узлы обеспечивают одновременную обработку группы сигналов - информационных слов.

Блоки реализуют некоторую последовательность в обработке информационных слов — функционально обособленную часть машинных операций (блок выборки команд, блок записи-чтения и др.).

Устройства предназначены для выполнения отдельных машинных операций и их последовательностей (АЛУ, УУ...).

В общем случае любая структурная единица ЭВМ обеспечивает преобразование входной информации X в выходную Y (см. рис. 6.1.).



Рис. 6.1. Представление схемы ЭВМ

В этом случае зависимостями $y_j = f(x_1, x_2, \dots, x_i, \dots, x_n)$, где x_i — i -й вход; n — число входов; y_i — i -й выход; m — число выходов в устройстве, можно описывать алгоритм работы любого устройства ЭВМ.

Все современные вычислительные машины строятся на комплексах (системах) **интегральных микросхем (ИС)**. Электронная микросхема называется **интегральной**, если ее компоненты и соединения между ними выполнены

- в едином технологическом цикле,
- на едином основании и
- имеют общую герметизацию и защиту от механических воздействий.

Каждая микросхема представляет собой миниатюрную электронную схему, сформированную послойно в кристалле полупроводника: кремния, германия, арсениде галлия и т.д. В состав микропроцессорных наборов включаются различные типы микросхем, но все они должны иметь единый тип межмодульных связей, основанный на стандартизации параметров сигналов взаимодействия (амплитуда, полярность, длительность импульсов и т.п.). Основу набора обычно составляют большие ИС (**БИС**) и даже сверхбольшие интегральные схемы (**СБИС**). На очереди следует ожидать появления ультра больших ИС (**УБИС**). Кроме них обычно используются микросхемы с малой и средней степенью интеграции (**СИС**).

Функционально **микросхемы** могут соответствовать **устройству, узлу или блоку**, но каждая из них состоит из комбинации простейших **логических элементов**, реализующих функции формирования, преобразования, запоминания сигналов и т.д.

Элементы ЭВМ можно классифицировать по различным признакам. Наиболее часто такими признаками являются:

- **тип сигналов,**
- **назначение элементов,**
- **технология их изготовления и т.д.**

В ЭВМ широко применяют два способа физического представления сигналов: **импульсный и потенциальный**. При **импульсном** способе представления сигналов единичному значению некоторой двоичной переменной ставится в соответствие наличие импульса (тока или напряжения), нулевому значению — отсутствие импульса (рис. 6.2.,а). Длительность импульсного сигнала не превышает один такт синхроимпульсов.

При **потенциальном или статическом** представлении сигналов единично значение двоичной переменной отображается высоким уровнем напряжения, а нулевое значение - низким уровнем (рис. 6.2.,б).

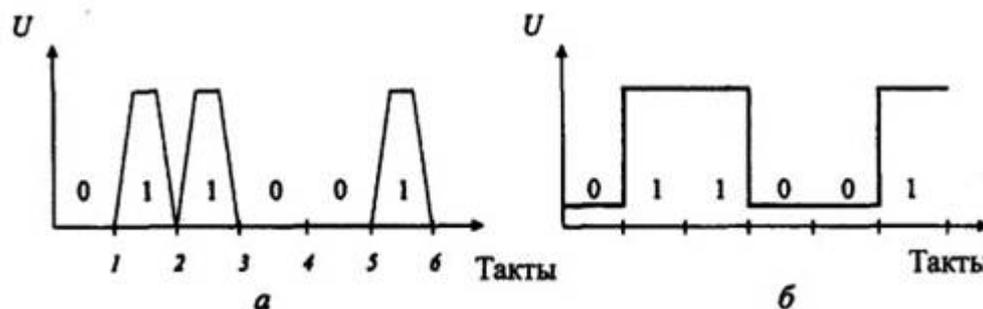


Рис. 6.2. Представление информации в ЭВМ: а — импульсные сигналы; б — потенциальные сигналы

Независимо от вида сигналов различают **последовательный и параллельный коды** передачи и представления информации в ЭВМ.

При **последовательном коде** представления данных используются одиночные шины или линии передачи, в которых сигналы, соответствующие отдельным разрядам данных, разнесены во времени. Обработка такой информации производится последовательно разряд за разрядом. Такой вид представления и передачи данных требует весьма экономичных по аппаратурным затратам схем обработки данных. Время же обработки определяется числом обрабатываемых сигналов (разрядов).

Параллельный код отображения и передачи информации предполагает параллельную и одновременную фиксацию всех разрядов данных на различных шинах, т.е. параллельный код данных развернут в пространстве. Это дает возможность ускорить обработку во времени, но затраты на аппаратурные средства при этом возрастают пропорционально числу обрабатываемых разрядов.

Во всех вычислительных машинах используются и **параллельно-последовательные коды** представления информации. При этом информация отображается частями. Части поступают на обработку последовательно, а каждая часть данных представляется параллельным кодом.

По своему назначению элементы делятся на:

- **формирующие,**
- **логические и**
- **запоминающие.**

К **формирующим элементам** относятся различные формирователи, усилители, усилители-формирователи и т.п. Данные элементы служат для выработки определенных электрических сигналов, восстановления их параметров (амплитуды, полярности, мощности, длительности).

В каждой ЭВМ имеются специальные блоки, формирующие сигналы тактовой частоты, серии синхронизирующих и управляющих сигналов, координирующих работу всех схем ЭВМ. Интервал времени между импульсами основной частоты называется тактом. Длительность такта является важной характеристикой ЭВМ, определяющей ее потенциальную производительность. **Время выполнения любой операции ЭВМ связано с определенным числом тактов.**

Простейшие **логические элементы** преобразуют входные сигналы в соответствии с элементарными логическими функциями, рассмотренными ранее. В свою очередь, полученные сигналы могут формировать следующий уровень сигналов и т. д. Сложные преобразования в соответствии с требуемыми логическими зависимостями могут приводить к построению многоуровневых схем. Каждая такая схема представляет собой композицию простейших логических схем.

Запоминающим элементом называется элемент, который способен принимать и хранить код двоичной цифры (единицы или нуля). Элементы памяти могут запоминать и сохранять исходные значения некоторых величин, промежуточные значения обработки и окончательные результаты вычислений. Только запоминающие элементы в схемах ЭВМ позволяют проводить обработку информации с учетом ее развития.

6.2. Комбинационные схемы

Обработка входной информации X в выходную Y (см. рис. 6.1) в любых схемах ЭВМ обеспечивается преобразователями или цифровыми автоматами двух видов: *комбинационными схемами и схемами с памятью*.

Комбинационные схемы (КС) — это схемы, у которых выходные сигналы $Y = (y_1, y_2, \dots, y_m)$ в любой момент дискретного времени однозначно определяются совокупностью входных сигналов $X = (x_1, x_2, \dots, x_n)$, поступающих в тот же момент времени t . Реализуемый в КС способ обработки информации называется комбинационным потому, что результат обработки зависит только от комбинации входных сигналов и формируется сразу при поступлении входных сигналов. Поэтому одним из достоинств комбинационных схем является их высокое быстродействие. Преобразование информации однозначно описывается логическими функциями вида $Y=f(X)$.

Логические функции и соответствующие им комбинационные схемы подразделяют на

- *регулярные и*
- *нерегулярные структуры.*

Регулярные структуры предполагают построение схемы таким образом, что каждый из ее выходов строится по аналогии с предыдущими. В **нерегулярных структурах** такая аналогия отсутствует. Примером построения нерегулярной структуры может служить разработка схемы в примере схемы, фиксирующей появление “неправильной” тетрады в двоично-десятичном представлении чисел (см. предыдущую лекцию).

В практике проектирования ЭВМ накоплен огромный опыт по синтезу различных схем. Многие регулярные структуры положены в основу построения отдельных ИС малой и средней степени интеграции или отдельных функциональных частей БИС и СБИС. Из регулярных комбинационных схем наиболее распространены **дешифраторы, шифраторы, схемы сравнения, комбинационные сумматоры, коммутаторы** и др.

Рассмотрим принципы построения подобных регулярных структур.

6.2.1. Дешифраторы и шифраторы

Дешифраторы (ДШ) — это комбинационные схемы с n входами и $m = 2^n$ выходами. Единичный сигнал, формирующийся на одном из m выходов, однозначно соответствует комбинации входных сигналов.

Например, разработка структуры ДШ для $n=3$ позволяет получить таблицу истинности (табл. 6.1) и логические зависимости.

Таблица истинности дешифратора

Входы			Выходы					
x_1	x_2	x_3	Y_0	Y_1	...	Y_5	...	Y_7
0	0	0	1	0		0		0
0	0	1	0	1		0		0
0	1	0	0	0		0		0
0	1	1	0	0	...	0	...	0
1	0	0	0	0		0		0
1	0	1	0	0		1		0
1	1	0	0	0		0		0
1	1	1	0	0		0		1

Дешифраторы широко используются в ЭВМ для выбора информации по определенному адресу, для расшифровки кода операции и др. Логические зависимости дешифратора:

$$\begin{aligned}
 y_0 &= \bar{x}_1 \bar{x}_2 \bar{x}_3, & y_4 &= x_1 \bar{x}_2 \bar{x}_3, \\
 y_1 &= \bar{x}_1 \bar{x}_2 x_3, & y_5 &= x_1 \bar{x}_2 x_3, \\
 y_2 &= \bar{x}_1 x_2 \bar{x}_3, & y_6 &= x_1 x_2 \bar{x}_3, \\
 y_3 &= \bar{x}_1 x_2 x_3, & y_7 &= x_1 x_2 x_3.
 \end{aligned}$$

На рис. 6.3 представлены структурная схема ДШ₂, построенная в базе (И, НЕ), и условное ее обозначение на принципиальных электрических схемах ЭВМ. Кружочки у линий, выходящих из логических элементов, указывают на инверсию функций, реализуемых элементами.

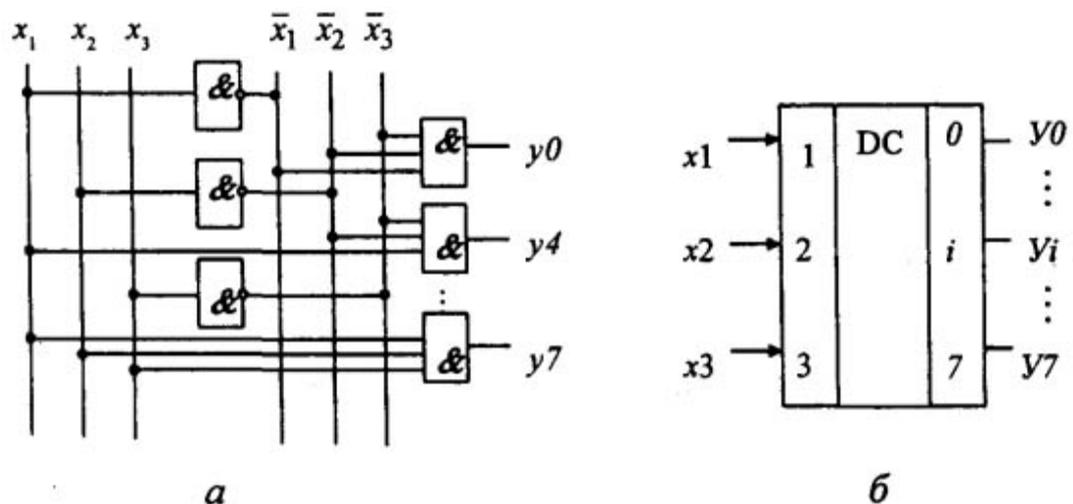


Рис. 6.3. Структурная схема дешифратора (а) и обозначение дешифратора на принципиальных электрических схемах (б)

Шифратор (ШР) решает задачу, обратную схемам ДШ, т. е. по номеру входного сигнала формирует однозначную комбинацию выходных сигналов. Рассмотрим пример построения ШР, который иллюстрируется таблицей истинности 6.2 и схемами на рис. 6.4.

Таблица истинности шифратора

Входы							Выходы		
x_1	x_2	x_3	x_4	x_5	x_6	x_7	y_0	y_1	y_2
1	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	1	1
0	0	0	1	0	0	0	1	0	0
0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	1	1	1	1

Логические зависимости, соответствующие таблице истинности:

$$y_2 = x_1 \vee x_3 \vee x_5 \vee x_7,$$

$$y_1 = x_2 \vee x_3 \vee x_6 \vee x_7,$$

$$y_0 = x_4 \vee x_5 \vee x_6 \vee x_7.$$

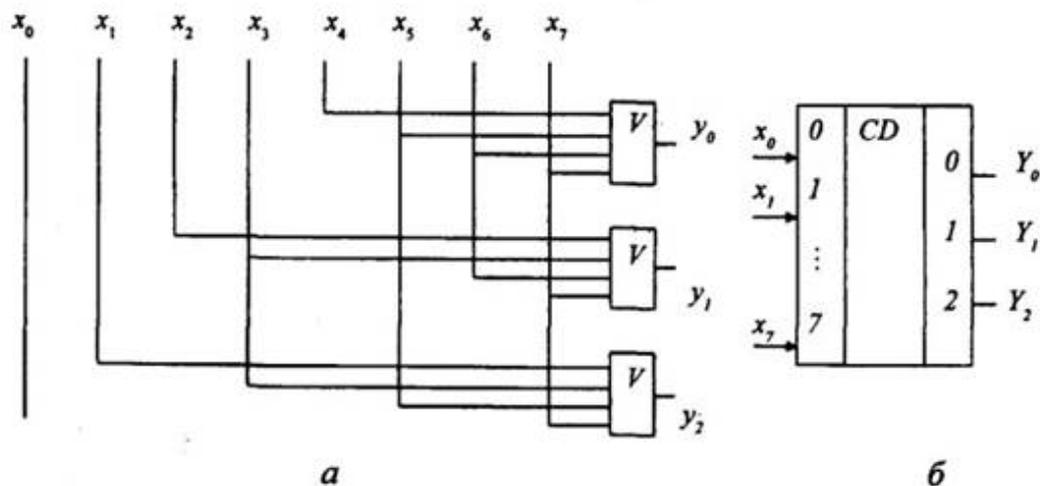


Рис. 6.4. Структурная схема шифратора (а) и обозначение шифратора на принципиальных электрических схемах (б)

Обратим внимание, что таблицы 6.1 и 6.2 во многом похожи: входы и выходы в них поменялись местами. Состояния входов табл.6.2. содержат только по одному единичному элементу. Другие произвольные комбинации входов недопустимы.

6.2.2. Компараторы и сумматоры

Схемы сравнения или компараторы обычно строятся как поразрядные. Они широко используются и автономно, и в составе более сложных схем, например, при построении сумматоров.

Таблица истинности (см. табл.6.3) отражает логику работы 1-го разряда схемы сравнения при сравнении двух двоичных векторов A и B .

Таблица истинности компаратора

Входы		Выходы
a_i	b_i	Y_i
0	0	1
0	1	0
1	0	0
1	1	1

Логическая зависимость в соответствие с таблицей истинности:

$$Y_i = \bar{a}_i \bar{b}_i \vee a_i b_i = \overline{\bar{a}_i b_i \vee a_i \bar{b}_i} = \overline{a_i \oplus b_i}$$

На рис. 6.5 показана структурная схема компаратора. Помимо выхода Y_2 , фиксирующего равенство значений разрядов, показаны выходы Y_1 и Y_3 , соответствующие сигналам "больше" и "меньше".

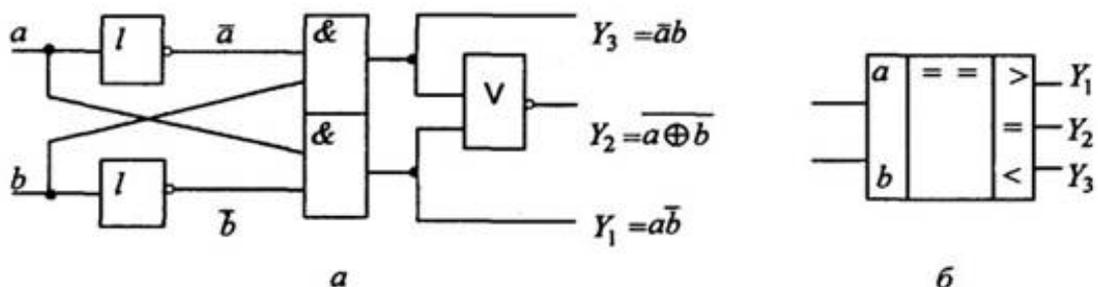


Рис. 6.5. Структурная схема компаратора (а) и обозначение компаратора на принципиальных электрических схемах (б)

Комбинационный сумматор. Принципы построения и работы сумматора вытекают из правил сложения двоичных цифр. **Схема сумматора также является регулярной** и широко используется в ЭВМ. При сложении одноразрядных двоичных цифр можно выявить закономерности в построении и многоразрядных сумматоров.

Сначала рассмотрим сумматор, обеспечивающий сложение двух двоичных цифр a_1 и b_1 , считая, что переносы из предыдущего разряда не поступают. Этой логике отвечает сложение младших разрядов двоичных чисел. Процесс сложения описывается таблицей истинности (см. табл. 6.4) и соответствующими логическими зависимостями, где S_i - функция одноразрядной суммы и P_i — функция формирования переноса. Перенос формируется в том случае, когда $a_i=1$ и $b_i=1$.

Таблица 6.4

Таблица истинности комбинационного полусумматора

Входы		Выходы	
a_i	b_i	S_i	P_i
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Логические зависимости:

$$S_i = \bar{a}_i \cdot b_i \vee a_i \cdot \bar{b}_i = a_i \oplus b_i;$$

$$p_i = a_i \cdot b_i.$$

Приведенные зависимости соответствуют логике работы самого младшего разряда любого сумматора. Структурная схема одноразрядного сумматора (полусумматора) представлена на рис.6.6.

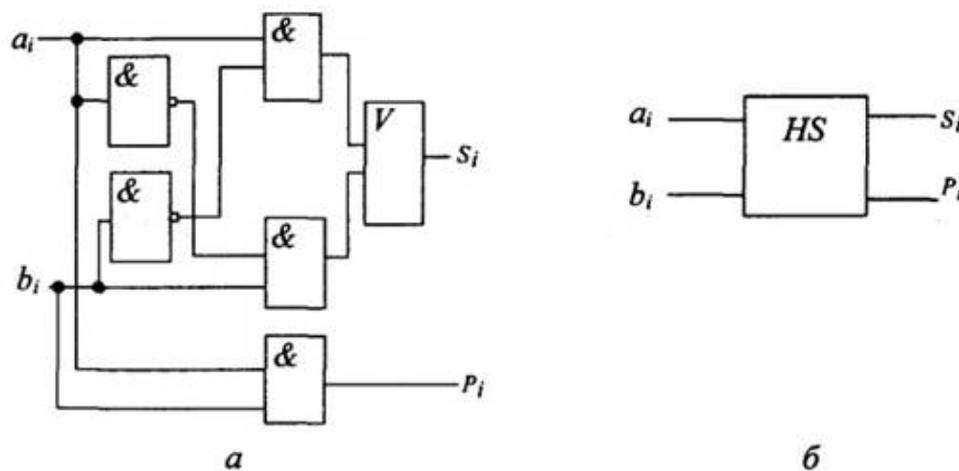


Рис.6.6. Структурная схема полусумматора (а) и обозначение полусумматора на принципиально электрических схемах (б)

Логические зависимости полусумматора S_i и компаратора очень похожи, так как они инверсны по отношению друг к другу.

Уравнения, положенные в основу одноразрядного сумматора, используются и при построении многоразрядных сумматоров. Логика работы каждого разряда сумматора описывается таблицей 6.5, которую можно считать его таблицей истинности.

Таблица 6.5

Правила сложения двоичных цифр

Значения двоичных чисел а и в			Разряд суммы S_i	Перенос в следующий разряд P_i
a_i	b_i	P_{i-1}		
0	0	0	0	0
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

Таблица истинности сумматора, учитывающего сигналы переноса, отличается от таблицы полусумматора (табл.6.4) дополнительным входом p — переносом из предыдущих разрядов.

Исходные логические зависимости, формируемые по табл.6.5, имеют следующие совершенные ДНФ:

$$S_i = \bar{a}\bar{b}p \vee \bar{a}b\bar{p} \vee a\bar{b}\bar{p} \vee abp,$$

$$P_i = \bar{a}bp \vee a\bar{b}p \vee ab\bar{p} \vee abp$$

Преобразование этих выражений приводит к следующим зависимостям:

$$\begin{aligned} S_i &= p(\bar{a}\bar{b} \vee ab) \vee b(\bar{a}\bar{p} \vee ap) \vee a(\bar{b}\bar{p} \vee bp) = \\ &= p(\overline{a \oplus b}) \vee b(\overline{a \oplus b}) \vee a(\overline{b \oplus p}); \end{aligned}$$

$$p_i = ab \vee ap \vee bp.$$

В приведенных выражениях индексы у переменных в правых частях уравнений опущены.

Из анализа логических зависимостей видно, что структурная схема i -го разряда сумматора требует включения в свой состав трех схем сравнения для формирования разрядной суммы и шести схем совпадения (см. рис. 6.7).

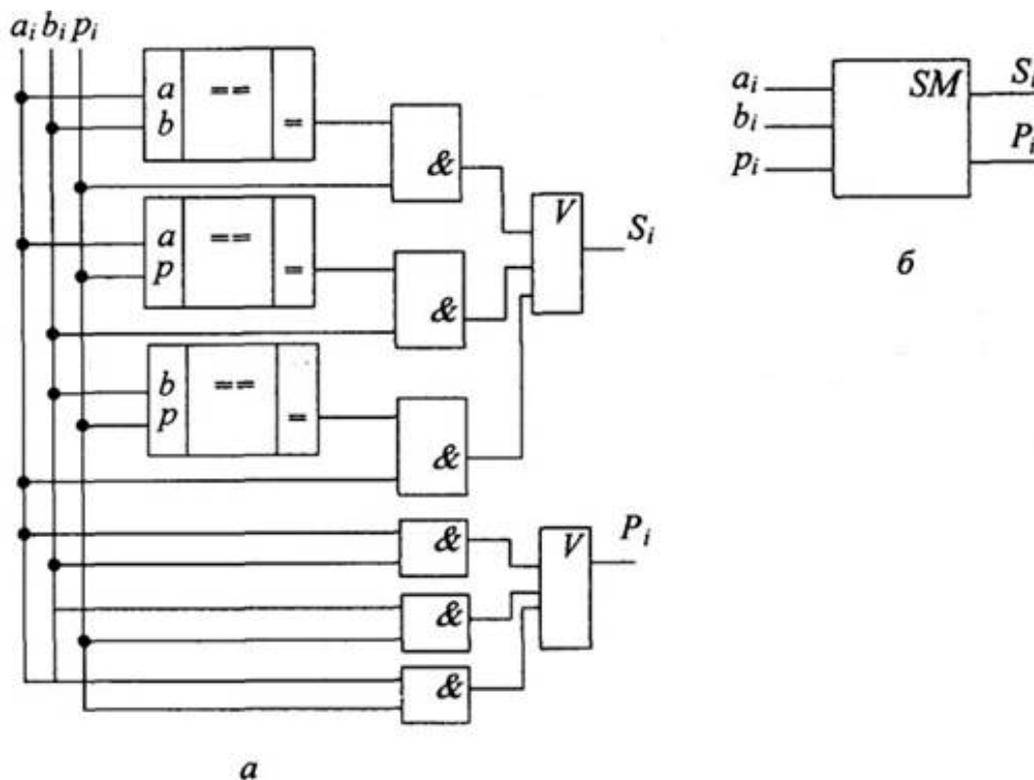


Рис. 6.7. Структурная схема одного разряда комбинационного сумматора:
 а — структурная схема одного разряда; б — условное изображение

Структурная схема многоразрядного комбинационного сумматора на электрических схемах изображена на рис.6.8.

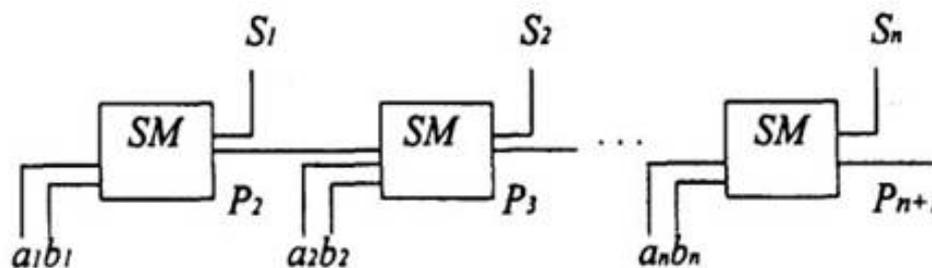


Рис. 6.8. Структурная схема многоразрядного комбинационного сумматора

6.2.3. Сумматор параллельного действия

Можно построить сумматор, в котором сложение производится как поразрядная операция и на распространение переноса не требуется дополнительного времени.

Для описания работы такого сумматора удобно ввести две функции:

γ — функция генерации переноса
 $\gamma = x \& y$

Функция равна 1, когда перенос в этом разряде возникает независимо от переноса на его входе.

π — функция прозрачности
 $\pi = x \vee y$

Функция равна 1, когда при возникновении переноса на входе данного разряда, на его выходе также возникает перенос, т.е. тракт разряда прозрачен для входного переноса.

С помощью введенных функций можно представить работу тракта переноса одного разряда сумматора

$$C_{i+1} = \gamma_i \vee C_i \& \pi_i,$$

где C_i – перенос из предыдущего разряда, C_{i+1} – перенос в следующий разряд.

Тогда перенос на входе $(i+1)$ разряда сумматора можно представить так:

$$C_{i+1} = \gamma_i \vee \gamma_{i-1} \pi_i \vee \gamma_{i-2} \pi_{i-1} \pi_i \vee \gamma_{i-3} \pi_{i-2} \pi_{i-1} \pi_i \vee \dots$$

Затраты оборудования на реализацию такого сумматора, особенно при большом количестве разрядов, достаточно велики.

6.2.4. Сумматор с групповым переносом

Сумматор разбивают на несколько групп. Каждая такая группа — мини сумматор — имеет свой штатный вход переноса.

Суть группового переноса заключается в том, что в дополнении к тракту переноса внутри группы, строят тракт переноса второго уровня между группами, который вырабатывает сигналы групповых переносов, подаваемые на входы «мини сумматоров».

Параллельный перенос между группами в сочетании с параллельным переносом внутри группы позволяет создавать быстрые сумматоры.

Рассмотрим пример построения **4-х разрядного параллельного сумматора.**

$$C_1 = \gamma_0 \vee \pi_0 C_0$$

$$C_2 = \gamma_1 \vee C_1 \& \pi_1 = \gamma_1 \vee (\gamma_0 \vee \pi_0 C_0) \& \pi_1 = \gamma_1 \vee \gamma_0 \pi_1 \vee \pi_0 \pi_1 C_0$$

$$C_3 = \gamma_2 \vee \gamma_1 \pi_2 \vee \gamma_0 \pi_1 \pi_2 \vee \pi_0 \pi_1 \pi_2 C_0$$

$$C_4 = \gamma_3 \vee \gamma_2 \pi_3 \vee \gamma_1 \pi_2 \pi_3 \vee \gamma_0 \pi_1 \pi_2 \pi_3 C_0$$

Остается построить схему, приведенную на рис.6.9:

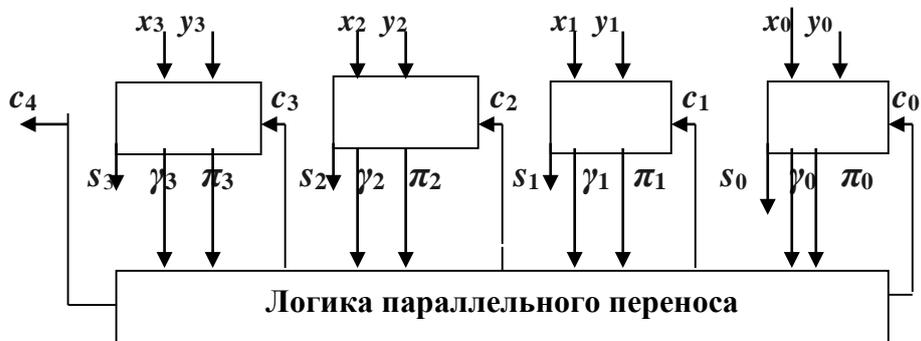


Рис. 6.9. Схема 4-х разрядного параллельного сумматора

Пример построения **16-х разрядного сумматора с параллельным переносом** на основе 4-х разрядного параллельного сумматора приведен на рис. 6.10.

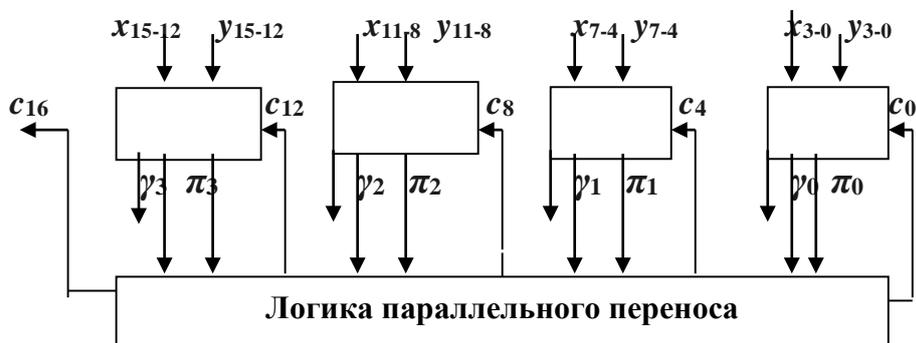


Рис. 6.10. Схема 16-и разрядного сумматора с групповым переносом

Тема 7. Элементы и узлы ЭВМ (продолжение темы б)

Основные вопросы:

7.1. Схемы с памятью

RS - триггер

T - триггер

JK - триггер

D - триггер

Регистр

Счетчик

7.2. Проблемы развития элементной базы

7.1. Схемы с памятью

Более сложным преобразователем информации являются схемы с памятью. Наличие памяти в схеме позволяет запоминать промежуточные состояния обработки и учитывать их значения в дальнейших преобразованиях.

Выходные сигналы $Y = (y_1, y_2, \dots, y_m)$ в схемах данного типа формируются не только по совокупности входных сигналов $X = (x_1, x_2, \dots, x_n)$, но и по совокупности состояний схем памяти $Q = (q_1, q_2, \dots, q_k)$. При этом различают текущий дискретный момент времени t и последующий $(t+1)$ момент времени (см. рис.7.1).

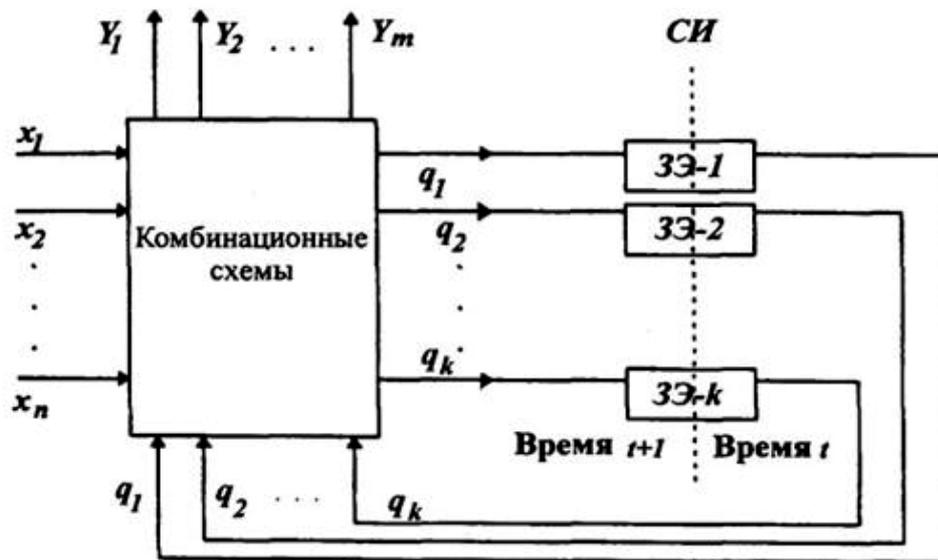


Рис. 7.1. Обобщенная структура схемы с памятью

Передача значения Q между моментами времени t и $(t+1)$ осуществляется обычно с применением двухступенчатой памяти и синхронизирующих импульсов (СИ).

В качестве простейшего запоминающего элемента (ЗЭ) в современных ЭВМ используют **триггеры**. В связи с успешным применением микроэлектроники в схемах основных устройств ЭВМ (процессоров и оперативной памяти) исчезли в качестве запоминающихся элементов схемы, использующие остаточную намагниченность — ферритовые сердечники. Самая простейшая схема триггера

может быть синтезирована по общим правилам, которые были рассмотрены в предыдущих лекциях.

RS-триггер

Автомат памяти — триггер, имеет вход R (**Reset** — сброс) для установки элемента в "нулевое состояние" и вход S (**Set** — установка) - для установки элемента в "единичное" состояние. При отсутствии сигналов $R=S=0$ элемент должен сохранять свое состояние до тех пор, пока не будут получены новые сигналы на входе R или S .

Условия работы триггера могут быть представлены в виде таблицы переходов (табл. 7.1.), представляющей собой модификацию таблицы истинности.

Таблица 7.1

Условия работы триггера

Входы		Состояние q_{t+1}		
R	S	q_t	q_{t+1}	Режим
0	0	0	0	Хранение 0
0	0	1	1	Хранение 1
0	1	0	1	Установка 1
0	1	1	1	Установка 1
1	0	0	0	Установка 0
1	0	1	0	Установка 0
1	1	0	?	Запрещенное состояние
1	1	1	?	Запрещенное состояние

Поясним содержание таблицы. Элемент памяти может сохранять значение $q_t=0$ или $q_t=1$ в зависимости от установки ранее установленного состояния. При отсутствии входных сигналов на входах R и S ($R=0$ и $S=0$) значения q_{t+1} первой строки таблицы в точности повторяют значения q_t . При поступлении сигнала $R=1$ (сигнала установки "нуля") элемент независимо от своего состояния принимает значение, равное нулю, $q_{t+1}=0$. Если же на вход S поступает сигнал установки "единицы" ($S=1$), то $q_{t+1}=1$ независимо от предыдущего состояния q_t . Одновременное поступление сигналов на входы R и S является *запрещенной ситуацией*, так как она может привести к непредсказуемому состоянию. В схемах формирования сигналов R и S должны быть предусмотрены блокировки, исключаящие их совпадения, $S=R=1$.

Для таблицы переходов 7.1 может быть построена диаграмма Вейча, которая приведена на рис. 7.2.

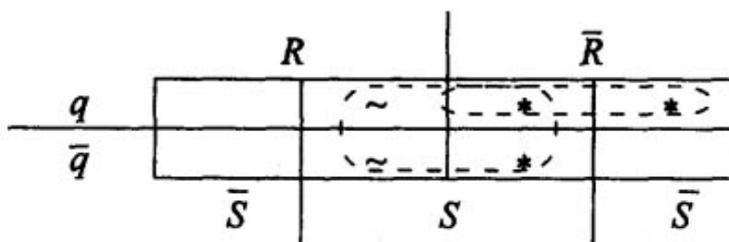


Рис. 7.2. Диаграмма Вейча для таблицы переходов триггера

На диаграмме знаком "~" отмечены запрещенные комбинации входных сигналов, которые могут быть использованы для упрощения логических зависимостей.

Логическая зависимость, описывающая работу элемента памяти, принимает вид:

$$q_{t+1} = \bar{R}_t \bar{S}_t q_t \vee \bar{R}_t S_t \bar{q}_t \vee \bar{R}_t S_t q_t = \bar{R}_t S_t \vee q_t \bar{R}_t.$$

Уравнение получено путем эквивалентных преобразований. Добавление в него комбинаций, соответствующих запрещенным ситуациям и помеченных знаком "~", т.е.

$$R_t S_t q_t \vee R_t S_t \bar{q}_t = R_t S_t,$$

позволяет еще больше упростить уравнение триггера:

$$q_{t+1} = \bar{R}_t S_t \vee q_t \bar{R}_t \vee R_t S_t = S_t (\bar{R}_t \vee R_t) \vee q_t \bar{R}_t = S_t \vee q_t \cdot \bar{R}_t.$$

Для реализации полученной зависимости в базисе **И - НЕ** применим правило де Моргана и получим функцию

$$q_{t+1} = \overline{\overline{S_t \vee q_t \bar{R}_t}} = \bar{S} \cdot \overline{q_t \bar{R}_t}.$$

По данной зависимости можно построить схему элемента памяти - **асинхронного RS-триггера**, которая приведена на рис. 7.3. В этой схеме следует только соединить выход q_{t+1} со входом q_t . (на рис. эта связь отмечена штриховой линией).

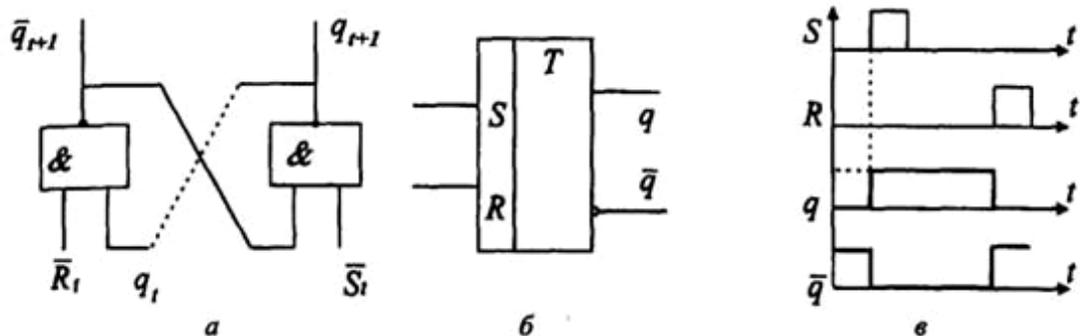


Рис. 7.3. Схема асинхронного RS-триггера: а — схема; б — обозначение на принципиальных электрических схемах; в - временная диаграмма

RS-триггер нашел широкое распространение в схемах ЭВМ. Одиночные триггеры этого типа часто используются в различных блоках управления. В асинхронных RS-триггерах имеется один существенный недостаток, обусловленный самой логикой их построения (см. табл.7.1), т.е. в них сигналы R и S должны быть разнесены во времени. Дополнение этого триггера комбинационными схемами синхронизации на входе и выходе позволяет получить триггеры с более сложной

логикой работы: синхронные **RS-триггеры**, **T-**, **JK-**, **D-** триггеры и целый ряд комбинированных **RST-**, **JKRS-**, **DRS-** триггеров.

Прописные буквы в названиях триггеров обозначают:

- **R (Reset — сброс)** — вход установки триггера в нулевое состояние $Q=0$;
- **S (Set — установка)** — вход установки триггера в единичное состояние $Q=1$;
- **T (Toggle — релаксатор)** — счетный вход триггера;
- **J (Jerk — внезапное включение)** — вход установки **JK** -триггера в единичное состояние $Q=1$;
- **K (Kill - внезапное выключение)** — $Q=0$;
- **D (Delay — задержка)** — вход установки триггера в единичное или нулевое состояние на время, равное одному такту;
- **C (Clock — часы)** — вход синхронизирующих тактовых импульсов.

На рис. 7.4 показаны схемы синхронного одноконтного (а) и двухконтного (б) **RS**-триггеров.

Двухкаскадная схема **RS-триггера** (рис.7.4.б) нашла наиболее широкое применение для построения **n-разрядных** схем запоминания — всевозможных регистровых схем. Штриховыми линиями на схеме указаны дополнительные точки подключения сигналов установки и сброса.

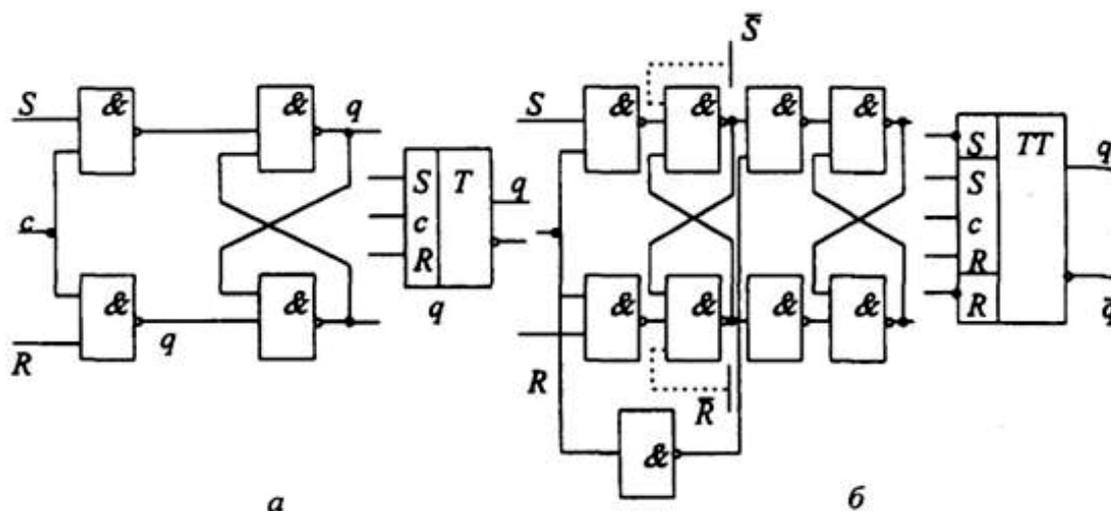


Рис. 7.4. Электрическая и функциональная схемы синхронных **RS**-триггеров: а — одноконтный, б — двухконтный

T-триггер

Схема **T-триггера** или иначе — триггера со счетным входом представлена на рис. 7.5. При значении $T=0$ триггер сохраняет свое ранее установленное состояние - режим хранения состояния, при $T=1$ триггер переходит в противоположное состояние. Таблица переходов (см. табл. 7.2.) и диаграмма работы (см. рис. 7.5.б) отражают динамику работы этого элемента.

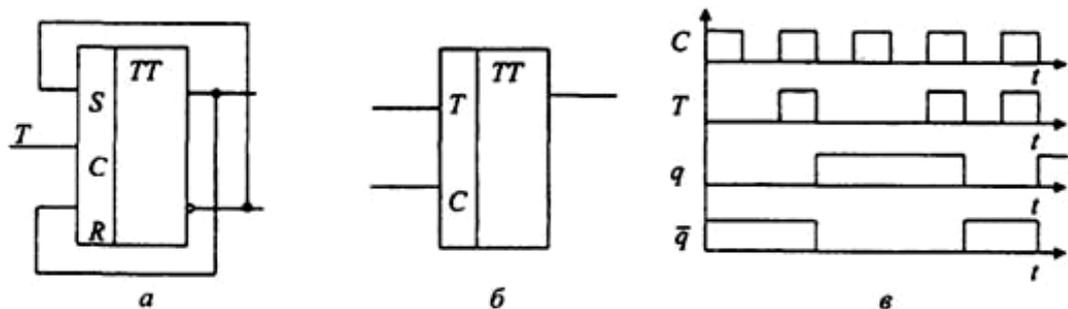


Рис. 7.5. Схема триггера со счетным входом: а- функциональная; б — условное обозначение; в — временная диаграмма

Таблица 7.2

Таблица переходов Т-триггера

Входные сигналы	Состояние q_t		Режим
	0	1	
X_t	0	1	
0	0	1	Хранение
1	1	0	Инверсия

По таблице переходов можно получить логическую функцию, реализуемую **T-триггером**:

$$q_{t+1} = \bar{q}_t \cdot x_t \vee q_t \cdot \bar{x}_t = q_t \oplus x_t.$$

Полученная зависимость очень похожа на функцию, выведенную для одноразрядного комбинационного полусумматора. На рис.7.5.,а показано, как двухтактный **RS-триггер** можно преобразовать в **T-триггер**, соединяя выходы с входами.

JK-триггер

Наиболее сложным типом триггера является **JK-триггер**. Он является объединением двухтактного **RS-** и **T-** триггеров. В табл.7.3 представлена таблица переходов **JK-триггера**.

Таблица 7.3

Таблица переходов JK-триггера

Входные сигналы		Состояние q		Режим
J	K	0	1	
0	0	0	1	Хранение
0	1	0	0	Установка 0
1	0	1	1	Установка 1
1	1	1	0	Инверсия

Если первые три строки таблицы переходов полностью повторяют соответствующие строки табл. 7.1, то последняя строка, с запрещенной комбинацией для RS-триггера, соответствует режиму переключения **T-триггера**.

Схема **JK-триггера** изображена на рис. 7.6.

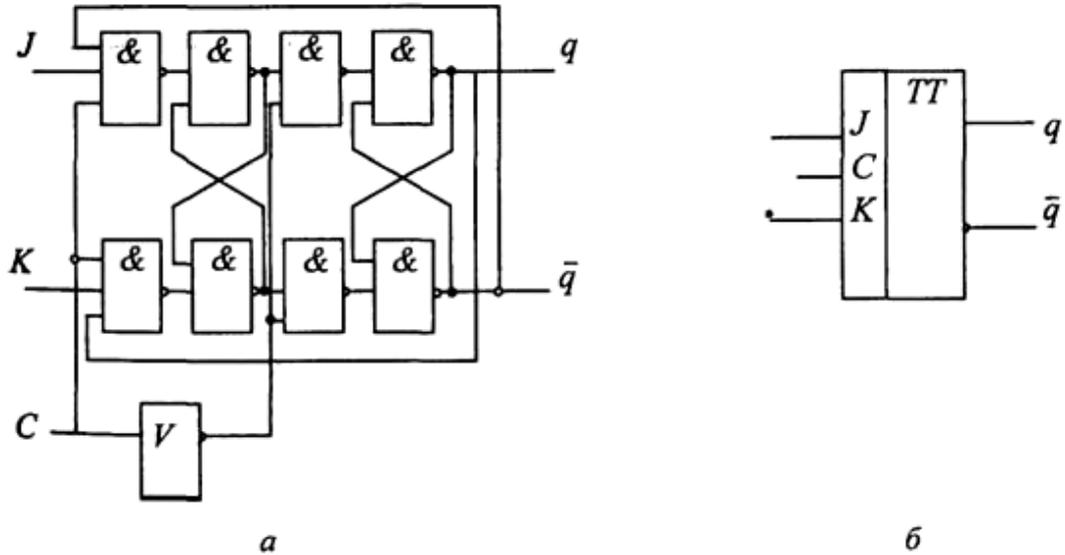


Рис. 7.6. JK-триггер: *a* — функциональная схема; *б* — условное обозначение

D-триггер

D-триггер обычно строится на основе двухтактного **RS-** или **JK-триггера**. Он предназначен для хранения состояния (1 или 0) на один период тактовых импульсов (с задержкой на 1 такт). Таблица его переходов представлена в табл. 7.4. На рис. 7.7., *a* и *б* представлены варианты его построения, а на рис. 7.7., *в* — его условное обозначение.

Таблица 7.4

Таблица переходов D-триггера

Входные сигналы	Состояния q		Режим
	0	1	
D			
0	0	0	Установка 0
1	1	1	Установка 1

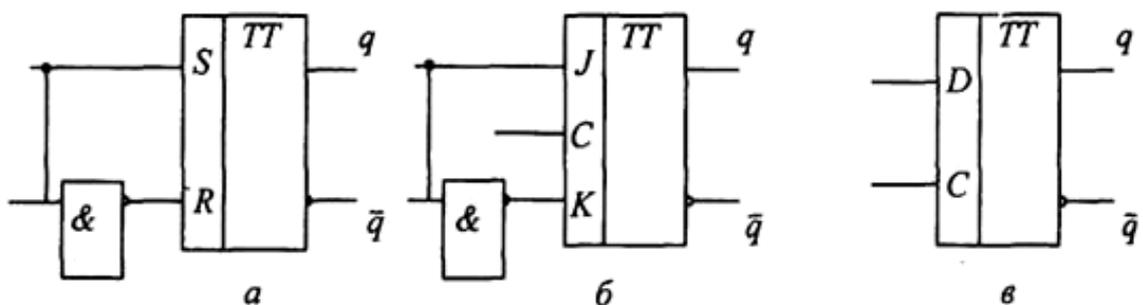


Рис. 7.7. D-триггер: *a* — функциональная схема на основе RS-триггера; *б* — функциональная схема на основе JK-триггера; *в* — условное обозначение

Все перечисленные элементы памяти позволяют хранить одну единицу информации - бит или одну двоичную цифру.

При построении ЭВМ широко используются функциональные схемы, обеспечивающие операции хранения и преобразования информации над группами битов (машинными словами). Такие сложные схемы называются **узлами**. К типовым узлам относят: **регистры, счетчики, сумматоры**. Все они также принадлежат к регулярным структурам, состоящим из одинаковых параллельно работающих одноразрядных схем.

Регистр

Регистром называется узел, предназначенный для приема, временного хранения и выдачи машинного слова. Регистры могут также использоваться для некоторых операций преобразования данных: для сдвига кода числа (слова) на определенное число разрядов влево или вправо, для преобразования последовательного кода числа в параллельный и наоборот и т.д. Эти дополнительные функции регистров обеспечиваются путем усложнения схем хранения, выбора более сложных триггеров и подключения дополнительных логических схем на их входах и выходах.

Таким образом, **регистры** представляют собой **совокупность триггеров**, число которых соответствует числу разрядов в слове, и **вспомогательных схем**, обеспечивающих выполнение различных операций над словом.

На рис. 7.8. показана функциональная схема **n -разрядного регистра**, построенного на **RS-триггерах**. Информация в регистр записывается под действием сигнала "Запись". Предварительно перед установкой кода на регистр обычно на все разряды **R** подается сигнал сброса. На рисунке показано, что подключение к входам **R** дополнительных инверторов позволяет избежать этой предварительной операции. Здесь на вход каждого разряда поступает парафазный код двоичной цифры (x_i — на вход **S** и $\neg x_i$ — на вход **R**), т.е. прямое и инверсное значения кода подаются в противофазе.

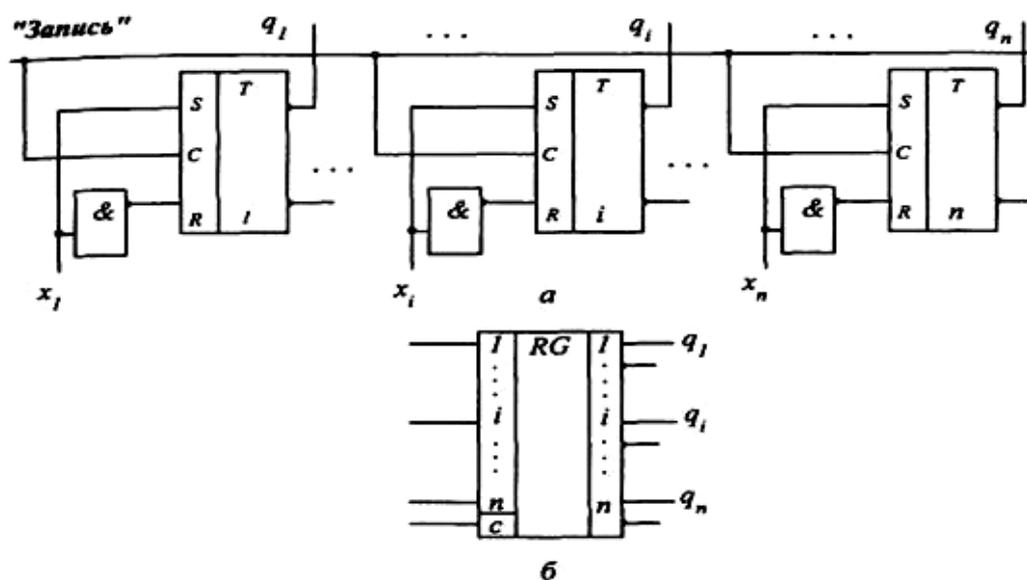


Рис.7.8. Схема регистра на RS-триггерах: а — функциональная схема; б — условное обозначение регистра

На рис.7.9 изображена функциональная схема того же регистра, дополненная логическими элементами для преобразования хранящегося на регистре кода. По сигналу "Прямой код" с регистра считывается прямой код хранящихся данных, а по сигналу "Обратный код" — инверсное значение каждого разряда слова. Если оба эти сигнала поступают одновременно, то считывается парафазный код хранящейся информации. Более сложная логика на входе и выходе запоминающих элементов позволяет строить сдвигающие регистры.

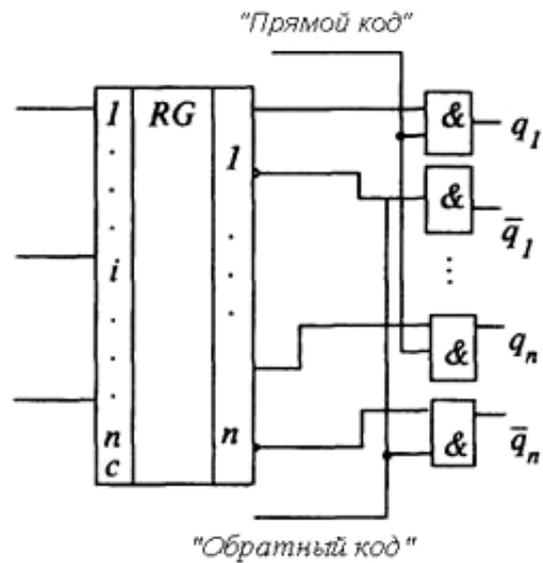


Рис. 7.9. Схема выдачи информации из регистра

Счетчик

Счетчик — узел ЭВМ, позволяющий осуществлять подсчет поступающих на его вход сигналов и фиксацию результата в виде многоразрядного двоичного числа. Счетчик, состоящий из n -триггеров, дает возможность подсчитывать до N сигналов, связанных зависимостью:

$$n = \log_2 N \text{ или } N = 2^n.$$

Счетчики используются для подсчета импульсов, сдвигов, формирования адресов и т.д. Функционально различают **суммирующие, вычитающие, реверсивные счетчики**. Они также отличаются друг от друга логикой работы дополнительных логических элементов, подключаемых к триггерам.

В основу построения любого счетчика положено свойство **T-триггеров** изменять свое состояние при подаче очередного сигнала на счетный вход T.

Рассмотрим работы трехразрядного суммирующего счетчика. В табл. 7.5 представлена логика его работы. На рис. 7.10 приводится схема трех разрядов суммирующего счетчика, построенного на **T-триггерах**.

Таблица 7.5

Таблица переходов трехразрядного счетчика

Вход x	Состояние								
	000	001	010	011	100	101	110	111	Режим
0	000	001	010	011	100	101	110	111	Хранение
1	001	010	011	100	101	110	111	000	Счет

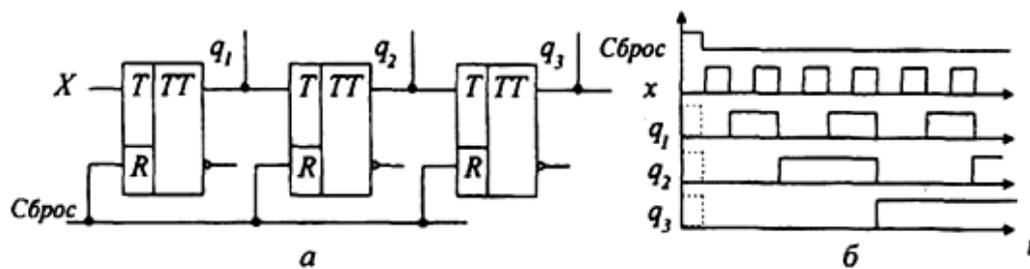


Рис. 7.10. Организация счетчика на T -триггерах: a — функциональная схема; b — временная диаграмма

Сумматор

Сумматор — узел ЭВМ, в котором суммируются коды чисел. Как правило, любой сумматор представляет собой комбинацию одноразрядных сумматоров. Сумматоры различают по принципам построения, выделяют:

- накапливающего типа и
- комбинационного типа.

Сумматоры накапливающего типа строят на сложных **JKRS-триггерах**, дополняя их выходы достаточно сложными схемами формирования и распространения переносов. Процесс сложения при этом осуществляется поэтапно. Сначала на триггерах сумматора фиксируется код первого операнда, затем на счетные коды разрядов подается код второго операнда. На каждом триггере формируются одноразрядные суммы и значения переносов между разрядами. Учет возникающих переносов задерживает формирование окончательного результата суммы и может требовать дополнительных тактов сложения. Из-за этого многоразрядные схемы сумматора накапливающего типа используются достаточно редко.

Более часто для построения сумматоров используются **сумматоры комбинационного типа**. Логика работы такого сумматора была представлена данными табл.6.5. Обычно у такого сумматора на входе и выходе имеются регистры для хранения и преобразования кодов операндов и результата (см. рис. 7.11).

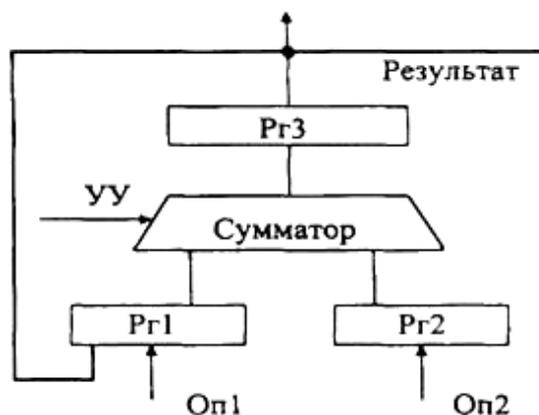


Рис. 7.11. Упрощенная схема сумматора ЭВМ

Регистр **Рг1** предназначен для хранения кода первого операнда, регистр **Рг2** — для хранения кода второго операнда. Сумматор по сигналам из устройства управления настраивается на выполнение определенной машинной операции,

соответствующей коду операции, находящемуся в коде команды. Результат выполняемой операции фиксируется в регистре **Rr3**. При необходимости этот результат может использоваться для продолжения вычислений. Для этого предусматривается возможность перезаписи содержимого регистра **Rr3** на **Rr1** в качестве значения одного из операндов при выполнении очередной операции.

7.2. Проблемы развития элементной базы

Одним из главных факторов достижения высокого быстродействия, а значит, и высокой производительности ЭВМ и ВС является построение их на новейшей элементной базе.

Элементная база в значительной степени определяет быстродействие и производительность ЭВМ. Смена поколений ЭВМ связана в значительной степени с переходом на **новые поколения элементной базы**. Новая элементная база означает достижение новых частотных диапазонов работы схем в рамках доступных технологий.

Элементная база служит показателем технического уровня развития страны, общества, цивилизации. Успехи в создании новой элементной базы определяются передовыми научными и техническими достижениями целого ряда наук: физики, химии, оптики, механики и др. Качество элементной базы является показателем технического прогресса.

Рост быстродействия определяется уровнем развития технологии.

Все современные ЭВМ и ВС строятся на микропроцессорных наборах, в основе которых — **БИС** и **СБИС**. Технологический принцип изготовления интегральных схем существует более 30 лет и называется **многослойная оптическая литография**.

Суть — циклическое послойное изготовление частей электронных схем по циклу: «**программа — рисунок — схема**».

По программам на намыленный фоторезисторный слой наносится рисунок будущего слоя микросхемы. Затем рисунок протравливается, фиксируется, закрепляется и изолируется от новых слоев. На основе этого создается пространственная твердотельная структура.

Например, СБИС Pentium содержит $3,5 \cdot 10^6$ транзисторов, размещенных в 5-ти слойной структуре.

На рис. 7.12 приведен кремниевый транзистор в разрезе. Критическим элементом такого транзистора является толщина изолирующего слоя оксида кремния, которая физически не может быть меньше 1,5 — 2 нм. Именно этот фактор определяет нижнюю границу размера самого транзистора.

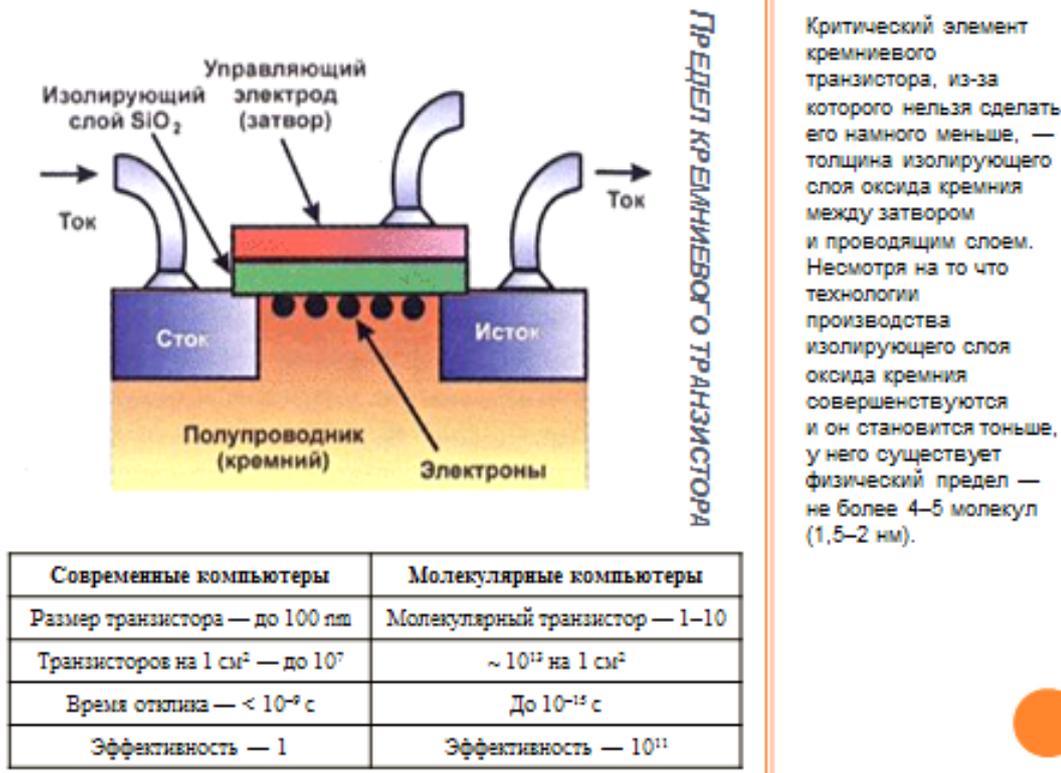


Рис. 7.12. Структура кремниевого транзистора

Степень микроминиатюризации, размер кристалла ИС, производительность и стоимость технологии напрямую определяются типом литографии. До настоящего времени доминирует **многослойная оптическая литография**, т.е. послойные рисунки на фоторезисторе микросхем наносятся световым лучом. Ведущие компании, производящие микросхемы, реализуют кристаллы с размерами примерно 400мм² — для процессоров (например, *Pentium*) и 200мм² — для схем памяти. Минимальный топологический размер (толщина линий) при этом составляет 0,5 — 0,35 мкм. Для сравнения можно привести такой пример. Толщина человеческого волоса составляет примерно 100 мкм. Значит, при таком разрешении на толщине волоса могут вычерчивать более двухсот линий.

Дальнейшее развитие микроэлектроники связывают с электронными современными технологиями, в основе которых лежат **лазерная, ионная и рентгеновская литография**. Это позволит выйти на размеры проводников 0,08 мкм с заменой алюминиевых проводников на медные, что позволит повысить частоту работы схемы.

Такие высокие технологии порождают целый ряд проблем:

- ✓ строительство и использование дорогостоящих заводов (2 — 4 млрд \$) со «сверхчистыми помещениями класса 1»

Микроскопическая толщина линий, сравнимая с диаметром молекул, требует высокой чистоты используемых и напыляемых материалов, применения вакуумных установок и снижения рабочих температур. Действительно, достаточно попадания мельчайшей пылинки при изготовлении микросхемы, как она попадает в брак. Поэтому новые заводы по производству микросхем имеют уникальное оборудование, размещаемое

в чистых помещениях класса 1, микросхемы в которых транспортируются от оборудования к оборудованию в замкнутых сверхчистых мини-атмосферах класса 1000. Мини-атмосфера создается, например, сверхчистым азотом или другим инертным газом при давлении 10^{-4} Торр

- ✓ уменьшение размеров микросхем и повышение уровня интеграции ведет к необходимости борьбы с потребляемой и рассеиваемой мощностью

Уменьшение линейных размеров микросхем и повышение уровня их интеграции заставляют проектировщиков искать средства борьбы с потребляемой W_n и рассеиваемой W_p мощностью. При сокращении линейных размеров микросхем в 2 раза их объемы изменяются в 8 раз. Пропорционально этим цифрам должны меняться и значения W_n и W_p , в противном случае схемы будут перегреваться и выходить из строя. В настоящее время основой построения всех микросхем была и остается КМОП-технология (комплиментарные схемы, т.е. совместно использующие *n*- и *p*-переходы в транзисторах со структурой металл - окисел - полупроводник).

Известно, что $W=U \cdot I$. Напряжение питания современных микросхем составляет 5 — 2V. Появились схемы с напряжением питания 1V, что выходит за рамки принятых стандартов. Дальнейшее понижение напряжения нежелательно, так как всегда в электронных схемах должно быть обеспечено необходимое соотношение сигнал-шум, гарантирующее устойчивую работу ЭВМ.

Протекание тока по микроскопическим проводникам сопряжено с выделением большого количества тепла. Поэтому, создавая сверхбольшие интегральные схемы, проектировщики вынуждены снижать тактовую частоту работы микросхем. На рис. 7.13 показано, что использование максимальных частот работы возможно только в микросхемах малой и средней интеграции. Максимальная частота $f_{max}=10^{11}-10^{12}$ Гц доступна очень немногим материалам: кремнию *Si*, арсениду галлия *GaAs* и некоторым другим. Поэтому они чаще всего и используются в качестве подложек в микросхемах.

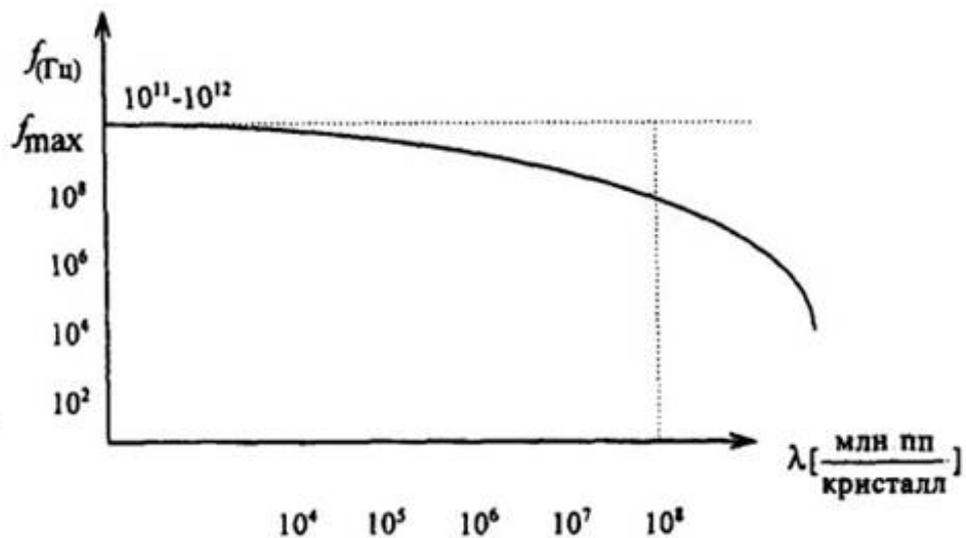


Рис. 7.13. Зависимость тактовой частоты f от степени интеграции Λ

Таким образом, переход к конструированию ЭВМ на СБИС и ультра-СБИС должен сопровождаться снижением тактовой частоты работы схемы.

Дальнейший прогресс в повышении производительности может быть обеспечен либо за счет **архитектурных решений**, либо за счет новых принципов построения и работы микросхем. Альтернативных путей развития просматривается не очень много. Так как микросхемы СБИС не могут работать с высокой тактовой частотой, то в ЭВМ будущих поколений СБИС видимо будут объединять в системы, где они будут работать параллельно, а управление ими будут осуществлять сверхскоростные интегральные схемы (ССИС) с малой степенью интеграции.

Большие исследования проводятся также в области использования явления **сверхпроводимости и туннельного эффекта** — эффекта Джозефсона. Работа микросхем при температурах, близких к абсолютному нулю (-273°C), позволяет достигнуть f_{max} , при этом $W_p=W_n=0$. Очень интересны результаты по использованию "теплой сверхпроводимости". Оказывается, что для некоторых материалов, в частности для солей бария, явление сверхпроводимости наступает уже при температурах около 150°C .

Успехов в развитии **микроэлектроники** можно ожидать за счет использования явления **сверхпроводимости** при плюсовых температурах («теплая сверхпроводимость»). Высказывались соображения, что могут быть получены материалы, имеющие сверхпроводимость при температурах, близких к комнатной. При этом потребляемая и рассеиваемая мощность будет близка к 0. Исследовательские работы в этом направлении являются закрытыми, а результаты могут совершить революцию в развитии средств ВТ новых поколений.

В настоящее время возможности микроэлектроники еще не исчерпаны!! Новые архитектурные решения привели к появлению:

- многоразрядных (64 и 128) микропроцессоров!
- многоядерных процессоров!

Основой для ЭВМ будущих поколений будут БИС и СБИС совместно с ССИС. При этом структуры ЭВМ и ВС будут широко использовать параллельную работу микропроцессоров.

Таким образом, можно сделать вывод, что в настоящее время возможности микроэлектроники еще не исчерпаны, но давление пределов уже ощутимо.

Альтернативными путями в области развития элементной базы следует считать разработку следующих классов компьютеров:

- **молекулярные компьютеры**

*В качестве еще одного из альтернативных путей развития элементной базы ЭВМ будущих поколений следует рассматривать и **биомолекулярную технологию**. В настоящее время имеются опыты по синтезу молекул на основе их стереохимического генетического кода, способных менять ориентацию и реагировать на ток, на свет и т.п.*

Однако построение из них биологических микромашин еще находится на стадии экспериментов.

- **биокомпьютеры (нейрокомпьютеры)**

Идея основана на теории перцептрона — **искусственной нейронной сети**, способной обучаться. Автор идеи — Ф.Розенблат. Он указал, что структуры, обладающие свойствами мозга и нервной системы, позволяют получить целый ряд преимуществ веществ: параллельность обработки; способность к обучению и настройке; способность к автоматической классификации; более высокую надежность; ассоциативность.

- **КВАНТОВЫЕ**

Принцип работы основан на способности электрона в атоме иметь различные уровни энергии, управлять которой можно с помощью электромагнитного поля (квантовые компьютеры); Основным блоком является qubit — Quantum Bit, который может иметь большое число состояний. Для таких блоков определен логически полный набор состояний элементарных функций. Есть эксперименты по созданию RISC-процессоров на RSFQ-логике (Rapid Single Flux Quantum) и проекты создания петафлопных компьютеров.

- **ОПТИЧЕСКИЕ**

В основе — способности света параллельно распространяться в пространстве; пока отсутствуют проекты создания чисто оптических компьютеров, но проводятся эксперименты по проектированию оптоэлектронных и оптонейронных отдельных устройств.

