

1. Виды рисков сбоя в асинхронных комбинационных схемах

Риском сбоя называется возможность появления на выходе цифрового устройства сигнала, не предусмотренного алгоритмом его работы и могущего привести к ложному срабатыванию.

Функциональная устойчивость определяется стабильностью реализации цифровым устройством заданного алгоритма работы при наличии разброса задержек выполнения операций в логических элементах, задержек сигналов в линиях связи и электромагнитных наводок паразитных сигналов. Термин "функциональная устойчивость" можно рассматривать также как синоним **алгоритмической устойчивости**.

В схемотехническом плане проблема функциональной устойчивости может быть сведена к устранению опасных состязаний (гонок) сигналов устройства. Проблема гонок в цифровой схемотехнике является очень серьезной. Большинство труднообнаруживаемых и удивительно разнообразно проявляющихся ошибок в цифровых схемах связано именно с гонками, возможность появления которых разработчик не предвидел или не заметил.

Состязаниями (гонками) сигналов называется процесс их распространения в различных цепях цифрового устройства при существовании разбросов временных задержек этих цепей.

Цепь - совокупность логических и других элементов и линий связи между ними.

Алгоритмическим переходом называется изменение сигнала на выходе какой-либо схемы, предусмотренное алгоритмом ее работы.

Неалгоритмическим переходом называется изменение выходного сигнала, не предусмотренное алгоритмом ее работы.

Опасными называются такие состязания, которые могут привести к неалгоритмическому переходу в цифровой схеме при заданных условиях ее работы, а **неопасными** называются такие состязания, которые не могут привести к неалгоритмическому переходу.

Схемой, свободной от влияния опасных состязаний, называется такая цифровая структура, в которой неалгоритмический переход, возникший в части схемы из-за опасных состязаний, не изменяет алгоритма работы схемы в целом при заданных условиях ее работы.

Задержка логической схемы складывается из задержек срабатывания логических элементов и задержек распространения сигналов по цепям связи между ними. Важнейшим параметром, характеризующим инерционность логического элемента, является среднее время задержки выходного сигнала по отношению к входному $t_{зд.ср.}$. Задержка каждого конкретного элемента зависит от количества и типа нагрузок, от паразитной емкости монтажа, числа лет с момента выпуска и ряда других факторов.

Изменение сигнала на каждом выходе схемы и внутреннем узле реально происходит не мгновенно, а образует некоторый сложный динамический процесс. Нахождение этих процессов называется **динамическим анализом** комбинационной схемы.

Статические риски сбоя

Риск сбоя называется **статическим**, если $y(X_1) = y(X_2)$, где y - булева функция.

На рис. 6 показана работа элементов **И** и **ИЛИ** при подаче на их входы двух последовательных во времени наборов $X_1 = x_1x_0 = 01$ и $X_2 = x_1x_0 = 10$. Значение сигнала

у₁ для элемента **И** на этих наборах должно оставаться постоянным и равным 0, а значение сигнала у₂ для элемента **ИЛИ** – равным 1. Это выполняется для случаев, показанных на рис. 6, а и г.

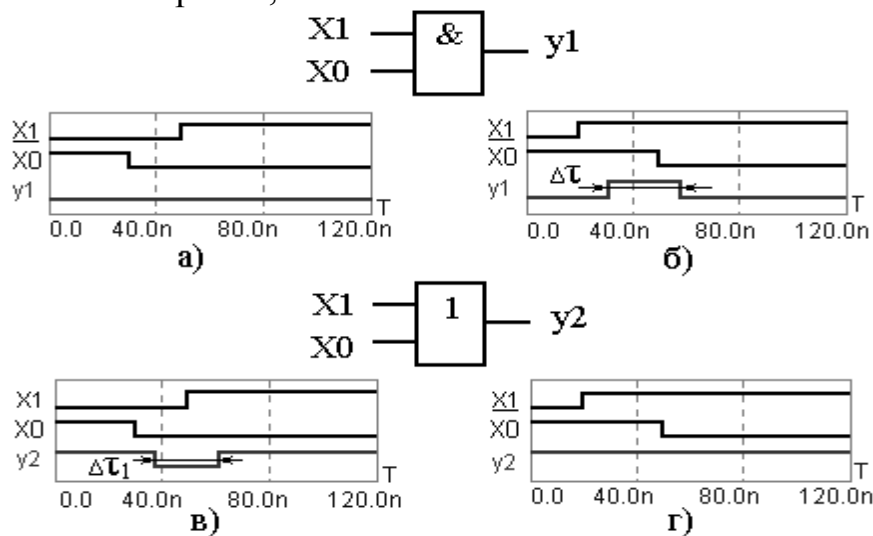


Рис. 6. Разброс во времени моментов переключения

Для случаев на рис. 6, б и в, то видно, что на выходе схемы **И** появится логический сигнал 1 длительностью $\Delta\tau$, а на выходе схемы **ИЛИ** - сигнал 0 длительностью $\Delta\tau_1$. Эти ложные сигналы и являются *рисками сбоя*, причем видно, что они могут проявляться, а могут и отсутствовать.

Риск сбоя называется *статическим в нуле* S_0 , если $y(X_1) = y(X_2) = 0$. Риск сбоя называется *статическим в единице* S_1 , если $y(X_1) = y(X_2) = 1$. На рис. 6, б имеет место статический риск сбоя в нуле S_0 , а на рис. 6, в - статический риск сбоя в единице S_1 .

Существенно, что полученные помехи (иногда их называют "иголками", "мерцаниями") - это не пренебрежимо короткий всплеск напряжения малой амплитуды. При достаточно большой разности $\Delta\tau$ это уже полноценный сигнал, на который могут реагировать последующие элементы. Такие помехи очень опасны для цифровых схем, тем более что их практически невозможно увидеть на осциллографе, и разработчик просто не будет догадываться об их существовании.

Динамические риски сбоя

Риск сбоя называется *динамическим*, если $y(X_1) \neq y(X_2)$, где y - булева функция.

Риск сбоя называется *динамическим* D_+ при изменении выходного значения $0 \rightarrow 1$, если $y(X_1) = 0$, а $y(X_2) = 1$.

Риск сбоя называется *динамическим* D_- , если $y(X_1) = 1$, а $y(X_2) = 0$.

Рассмотрим схему (рис. 7, а), состоящую из последовательно соединенных двухвходовых вентилях. На входы первого вентиля поступают переменные x_1 и x_2 . На входы второго поступают переменная x_0 и результат логической операции первого вентиля. В целом такая схема реализует логическую функцию: $y = x_2x_1 + x_0$.

1. Допустим, что набор входных сигналов в какой-либо момент времени, имеющий состояние $X_1 = x_2x_1x_0 = 010$, изменяется на комбинацию входных сигналов: $X_2 = x_2x_1x_0 = 101$. При этом входные сигналы изменяются не одновременно, а с некоторой задержкой (рис. 7, б).

Поскольку $y(X_1) = 0$, а $y(X_2) = 1$, из рис. 7, б видно, что на выходе схемы происходит многократное изменение уровня вместо идеального алгоритмического перехода $0 \rightarrow 1$.

2. Допустим, что набор входных сигналов в какой-либо момент времени, имеющий состояние $X_1 = x_2x_1x_0 = 011$, изменяется на комбинацию входных сигналов: $X_2 = x_2x_1x_0 = 100$, также, как и в случае 1 не одновременно. На выходе схемы, вместо идеального алгоритмического перехода $1 \rightarrow 0$, наблюдается многократное переключение.

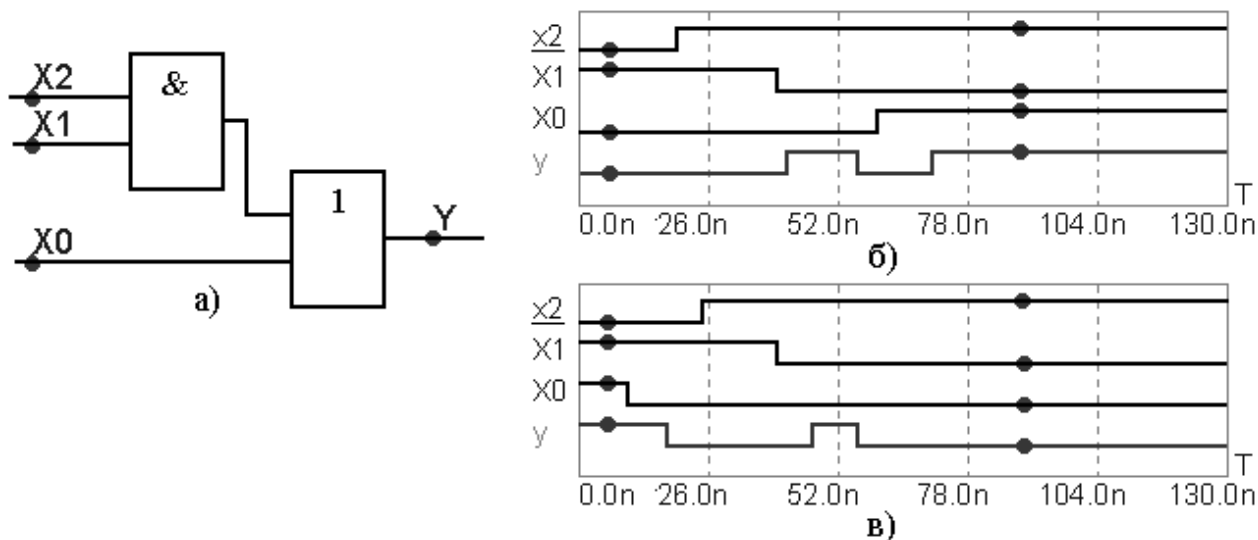


Рис. 7. Наихудший случай разброса во времени моментов переключения

Функциональный риск сбоя

Рассмотрим теперь поведение выходного сигнала той же функции при множественной смене наборов. Пусть набор 0 переходит в набор 7. Так как все переменные изменяются, а моменты их изменения в общем случае неизвестны, то возможны различные переходы от набора 0 к набору 7 (см. рис. 10, а).

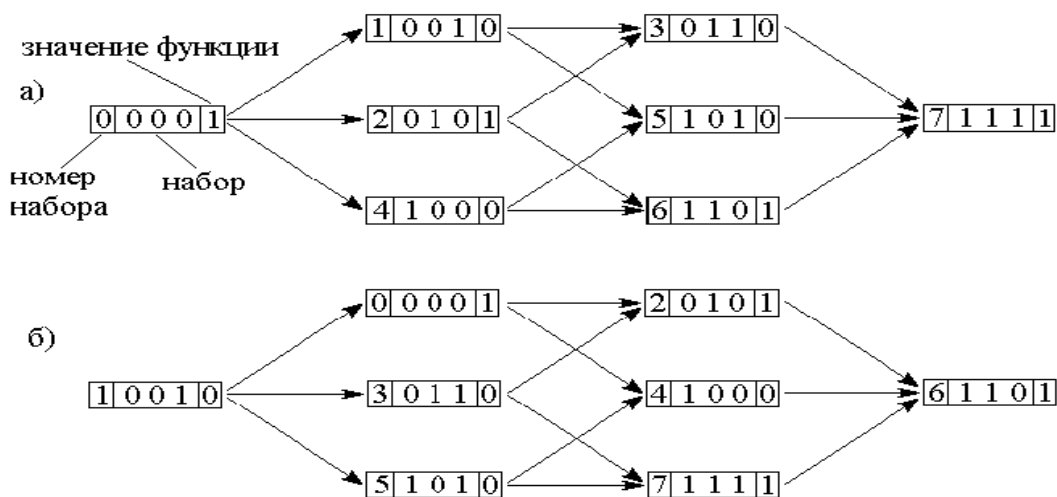


Рис. 10. Функциональный риск сбоя

Есть единственный путь смены наборов: $0 \rightarrow 2 \rightarrow 6 \rightarrow 7$, при котором не будет статического риска сбоя, так как $y(X_1 = 0) = y(X_2 = 7) = 1$. Во всех остальных случаях будет статический риск сбоя в единице S_1 , причем никакими аппаратными средствами устранить его нельзя, так как значения выхода на промежуточных наборах определяются *характером самой функции*.

По тем же причинам при переходе от набора $X_1 = 1$, на котором $y(X_1) = 0$, к набору $X_2 = 6$, на котором $y(X_2) = 1$, возможен путь смены наборов: $1 \rightarrow 0 \rightarrow 4 \rightarrow 6$, когда имеет место динамический риск сбоя D_+ , также определяемый характером самой

функции. Во всех остальных случаях смены наборов будет чисто алгоритмический переход 01 (см. рис. 10, б).

Риски сбоя, проявляющиеся при многоместной смене наборов и определяемые характером самой функции, называются **функциональными**. Такие риски сбоя не могут быть устранены изменением логической структуры, реализующей булеву функцию.

Логический риск сбоя

Статический риск сбоя называют **логическим**, если он обусловлен выбранной схемной реализацией функции.

Пусть задана булева функция таблицей истинности.

x_2	x_1	x_0	y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

С помощью карты Карно минимизируем заданную функцию:

		$x_1 x_0$			
		00	01	11	10
x_2	0	1	0	0	1
	1	0	0	1	1

Получим минимальную ДНФ: $y = x_2 x_1 + \bar{x}_2 \bar{x}_0$.

На рис. 8, а приведена схема, реализующая минимизированную функцию.

Рассмотрим переход от состояния, определяемого набором $X_1 = x_2 x_1 x_0 = 110$ к набору $X_2 = x_2 x_1 x_0 = 010$ для функции y .

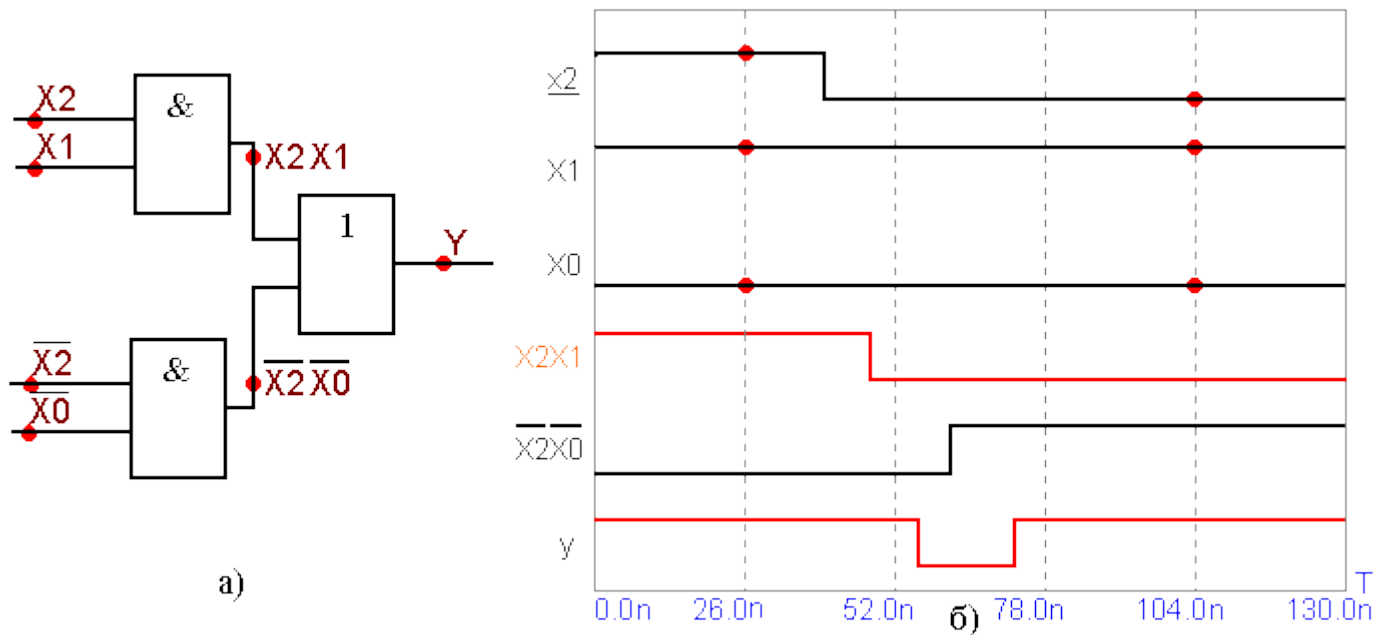


Рис. 8. Логический риск сбоя

Обратите внимание, что в данном случае осуществляется переход между соседними наборами в карте Карно. На рис. 8, б видно, что на элементе **ИЛИ** возможен статический риск сбоя в единице.

Этот риск сбоя можно устранить, если изменить аппаратную реализацию данной функции. Для этого в карте Карно вводится дополнительный контур, показанный штриховой линией, тогда уравнение для функции приобретет вид $y = x_2 x_1 + \overline{x_2} \overline{x_0} + x_1 \overline{x_0}$ (рис. 9, а). Так как при переходе от набора X_1 к X_2 простая импликанта $x_1 \overline{x_0}$ не изменяется и равна 1, то риск сбоя, выявленный выше, не будет проявляться на выходе схемы. Эта ситуация отражена на временной диаграмме, представленной на рис. 9, б.

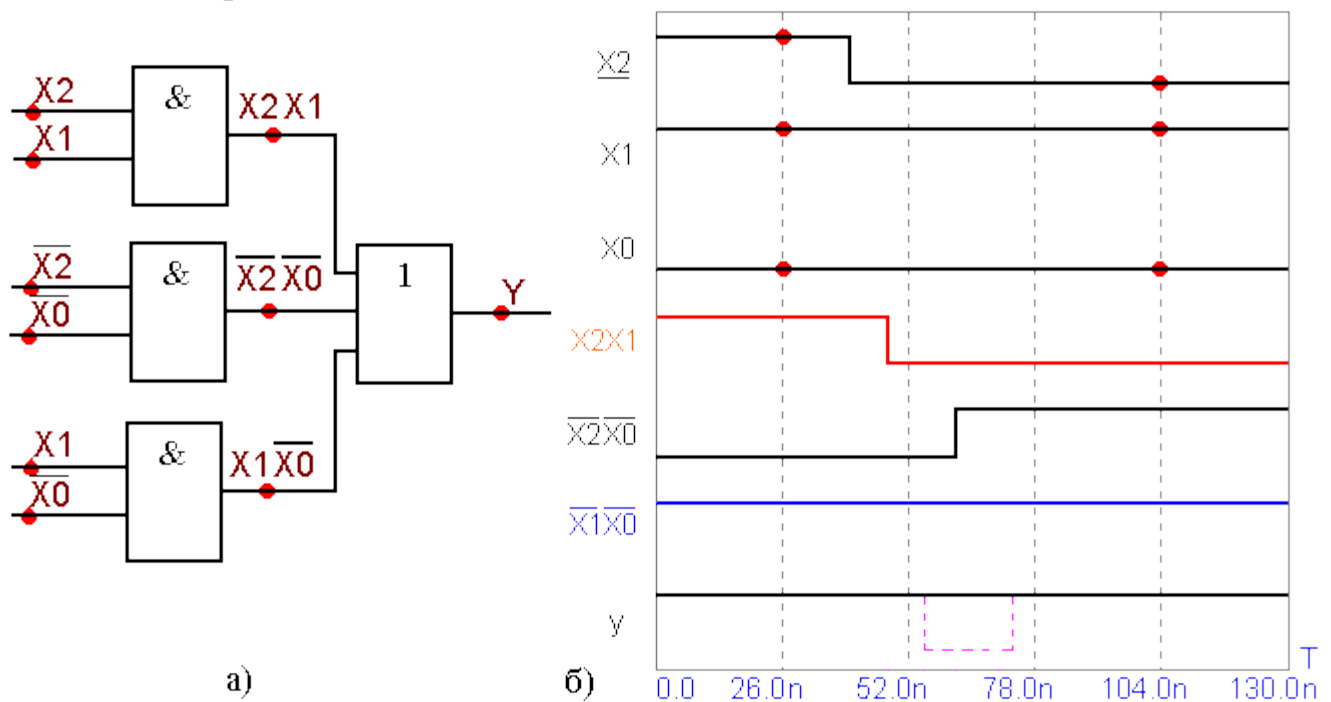


Рис. 9. Статический риск сбоя

2. Распределенная обработка информации. Примеры использования в промышленности.

Под распределенной обработкой информации понимается комплекс операций с информацией (традиционно описываемый термином «обработка информации»), проводимый на независимых, но связанных между собой вычислительных машинах, предназначенных для выполнения общих задач.

Системы распределенной обработки информации (или распределенные вычислительные системы) в виде многомашинных вычислительных комплексов и компьютерных сетей представляют собой одну из наиболее прогрессивных форм организации средств вычислительной техники.

Появление и широкое распространение систем распределенной обработки информации обусловлено, с одной стороны, ускоренным развитием микроэлектроники, снижением стоимости вычислительных средств, увеличением их производительности при уменьшении габаритов, а с другой стороны – повышением требований к производительности, надежности и эффективности вычислительных систем, предъявляемых сферами их применения.

Распределенная обработка информации обычно понимается как синоним распределенных вычислений. Важно отметить, что распределение (или разделение) не идентично параллелизму. Распределение видов обработки информации состоит в том, чтобы поручить их вычислительным машинам, наилучшим образом приспособленным к этому. Параллелизм же подразумевает понятие одновременности обработки информации. При этом распределение позволяет в ряде ситуаций проводить эффективную параллельную обработку информации при выполнении больших объемов параллельных вычислений. Таким образом, в общем случае распределение не подразумевает параллелизма, но возможность «распараллелить» распределенную обработку информации существует.

Возможность взаимодействия вычислительных систем при реализации распределенной обработки информации определяют как их способность к совместному использованию данных или к совместной работе с использованием стандартных интерфейсов. Взаимодействие подразумевает понятие «открытых систем», то есть систем, способных к коммуникации в неоднородной среде. Взаимодействие между программами с точки зрения хронологии по-следовательно приобретало следующие формы:

- обмен (программы различных систем посылают друг другу сообщения, как правило, файлы);
- разделение (имеется непосредственный доступ к ресурсам нескольких машин, например, совместное использование файлов);
- совместная работа (машины играют в реализации программы взаимодополняющие роли).

Целью распределенной обработки информации является оптимизация использования ресурсов.

3. Выявить сходимость метода простых итераций для решения следующих уравнений и систем:

1) $1-x^2=0$

2) $\cos(y-1) + x = 0,8$

$y - \cos(x) = 2$

3) $AX=B$, где

$$A = \begin{pmatrix} 0,25 & -0,52 & 0,043 & 0 \\ 0,14 & -0,16 & -0,316 & 0 \\ 0,12 & 0,08 & -0,14 & -0,24 \\ 0,12 & -0,35 & -0,18 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0,44 \\ 1,42 \\ -0,83 \\ -1,42 \end{pmatrix}$$

Решение:

1) $x = \varphi(x)$ или исходное уравнение 1 преобразуется в вид $x = 1 + x - x^2$.

$\varphi'(x) = 1 - 2 \cdot x$. Для $|x| < 1$ $\varphi'(x) < 1$, значит метод сходится для корней в окрестности точки 0.

2) Приведем данную систему к стандартному виду $\begin{cases} x = 0,8 - \cos(y-1) \\ y = 2 + \cos(x) \end{cases}$

$$\varphi'(x, y) = \begin{pmatrix} \frac{\partial \varphi_1}{\partial x} & \frac{\partial \varphi_1}{\partial y} \\ \frac{\partial \varphi_2}{\partial x} & \frac{\partial \varphi_2}{\partial y} \end{pmatrix} = \begin{pmatrix} 0 & \sin(y) \\ -\sin(x) & 0 \end{pmatrix}; |\varphi'(x, y)| = -\sin(x) * \sin(y) \leq 1. \quad \text{Следовательно} \quad \text{метод}$$

простых итераций сходится.

3) в матричном виде общая формула имеет вид $x = B * x + c$. Мы имеем вид $A * x = b$. Преобразуем: $A * x + E * x = b + x$ или $x = (A + E) * X - b$.

$$B = \begin{pmatrix} 1.2500 & -0.5200 & 0.0430 & 0 \\ 0.1400 & 0.8400 & -0.3160 & 0 \\ 0.1200 & -0.3500 & 0.8200 & -0.2400 \\ 0.1200 & -0.3500 & -0.1800 & 1.0000 \end{pmatrix} \quad c = \begin{pmatrix} -0.44 \\ -1.42 \\ 0.83 \\ 1.42 \end{pmatrix}$$

$|B| = 0.7172 < 1$, значит метод сходится для данной системы линейных уравнений