

## Глава 3

# Математическое моделирование цифровых устройств

---

---

Целью моделирования цифровых устройств (ЦУ) является получение картины их логико-временного поведения при различных входных воздействиях. В настоящее время принято рассматривать три области представления ЦУ: поведенческую, структурную и физическую [1]. Для каждой из этих областей выделяют различные уровни абстракции: системный, языков регистровых передач (ЯРП), логический и схемный. При этом в поведенческой области дается функциональное представление ЦУ, в структурной области описываются блоки архитектуры, физическая область отражает реальные корпуса микросхем или кристалл (chip).

Процесс автоматизированного проектирования ЦУ можно представить в виде последовательного спуска по уровням абстракции от системного до выполнения ЦУ на кристалле. Проектирование ЦУ начинается с разработки технического задания, на базе которого строится функциональная схема, последовательно преобразуемая в реальное устройство. Далее функциональная модель трансформируется в модель уровня языков регистровых передач, которая строится с использованием таких компонентов, как регистры, модули памяти, операционные и управляющие автоматы. На следующем этапе выполняется синтез логических схем для каждого компонента. На уровнях ЯРП и логическом, как правило, выполняется верификация проекта. Системный и ЯРП уровни абстракции относятся скорее к разделам цифровой техники и здесь рассматриваться не будут.

Модели ЦУ, используемые при проектировании радиоустройств на логическом и схемном уровнях, можно разделить на две основные группы: физические и логические (функциональные).

В первом случае отдельные элементы, из которых состоит ЦУ, представляются их электрическими макромоделями, состоящими из базовых элементов теории цепей (см. главу 1), на основании которых формируется полная электрическая модель ЦУ [6]. Система дифференциальных уравнений во временной области, соответствующая полной электрической модели ЦУ, может быть получена одним из алгоритмов, описанных в главе 2, и обычно представляет собой систему дифференциальных уравнений высокого порядка, решение которой требует больших машинных ресурсов. Физические модели элементов ЦУ позволяют наиболее полно представить работу устройства во времени с учетом реальных задержек срабатывания элементов, но их целесообразно использовать на заключительном этапе проектирования ответственных ЦУ из-за больших затрат времени на моделирование, когда необходимо иметь данные об устройстве, которые нельзя получить с помощью более простых моделей.

В логических моделях каждый элемент ЦУ представляется упрощенной формальной моделью в виде логического соотношения, описывающего логику функционирования элемента, таблиц истинности в виде «примитивных» простых кубов либо

графов переходов. При меньшей детализации работы по сравнению с физическими моделями логические модели обладают во много раз большим быстродействием и позволяют на начальных этапах проектирования ЦУ решить ряд важных для практики задач:

- ◆ проверить правильность логического функционирования ЦУ;
- ◆ сравнить характеристики различных вариантов схемных решений;
- ◆ разработать процедуры тестового контроля ЦУ и проверить их правильность и полноту;
- ◆ проверить работу цепей установки ЦУ в начальное состояние.

В настоящее время разработано множество алгоритмов, пакетов прикладных программ и даже специализированных высокоуровневых языков описания (*hardware design languages* — HDL) для моделирования ЦУ на логическом уровне. Рассмотрим некоторые из них применительно к комбинационным цифровым устройствам.

### 3.1. Описание языков моделирования и элементов цифровых устройств в моделях логического уровня

Логическое моделирование выполняется в предположении, что информация обрабатывается элементами ЦУ и передается в схеме другим элементам в одном направлении — от входов к выходам, причем физическая природа сигналов (ток или напряжение), распространяющихся в ЦУ, не конкретизируется. Сигналы задаются символами, которые описывают состояние и работу элементов ЦУ. Совокупность символов, используемых при моделировании, называют алфавитом логического моделирования. Между реальными сигналами и символами алфавита всегда можно установить однозначное соответствие [1, 7].

Простейший из используемых алфавитов для моделирования ЦУ на логическом уровне — двоичный, включающий всего два символа: 0 и 1. Он обеспечивает максимальную скорость моделирования, но не позволяет выявить неоднозначность работы ЦУ и характер переходных процессов.

Моделирование многозначными алфавитами позволяет получить больше информации о ЦУ. Так, для анализа состязаний сигналов на элементах ЦУ используют троичный алфавит, содержащий кроме 0 и 1 еще символ неопределенного состояния  $X$ , которому ставится в соответствие процесс перехода из 0 в 1 и обратно либо безразличное состояние сигнала (рис. 3.1, *a*).

Для примера на рис. 3.2, *a*, приведены таблицы истинности для наиболее распространенных базовых элементов цифровых схем И, ИЛИ, НЕ при моделировании трехзначным алфавитом. При этом полагалось, что  $\bar{X} = X$ .

Для уточнения характера процесса смены состояний элементов ЦУ используется пятизначный алфавит, показанный на рис. 3.1, *б*, в котором символы 0, 1 и  $X$  имеют тот же смысл, что и в трехзначном алфавите, а символ  $E$  описывает гладкий переход сигнала из 0 в 1. Соответственно, символ  $\bar{E}$  будет описывать обратное изменение сигнала из 1 в 0. Таблицы истинности для элементов И, ИЛИ, НЕ при моделировании их пятизначным алфавитом приведены на рис. 3.2, *б*.

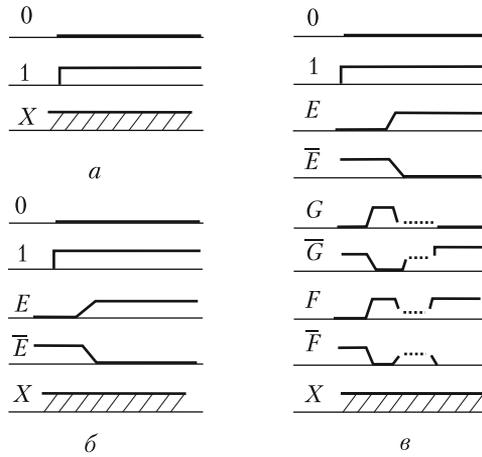


Рис. 3.1. Алфавиты моделирования: а — трехзначный; б — пятизначный; в — девятизначный

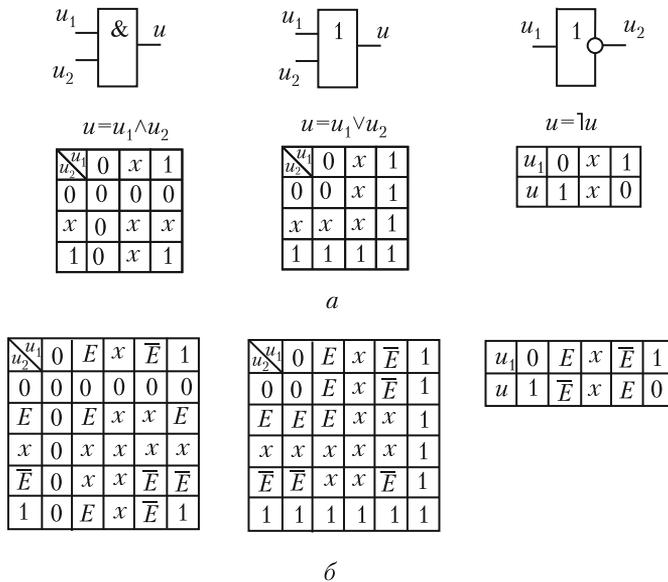


Рис. 3.2. Таблицы истинности простых логических элементов: а — для трехзначного алфавита; б — для пятизначного алфавита

Более подробно характер переходных процессов в ЦУ описывается при использовании девятизначного алфавита, включающего символы 0, 1, X, E,  $\bar{E}$ , G,  $\bar{G}$ , F,  $\bar{F}$ . По сравнению с пятизначным алфавитом здесь добавлены символы G,  $\bar{G}$  и F,  $\bar{F}$ , соответствующие статическим и динамическим сбоям при переключении из 0 в 1 и из 1 в 0 (см. рис. 3.1, в).

Для некоторых технологий (например, КМОП) требуется введение дополнительного сигнала Z, соответствующего состоянию высокого импеданса для схем с отклю-

чающимся выходом. В настоящее время широко используются лишь несколько вариантов многозначных алфавитов, которые применяются в логическом моделировании и для генерации тестов. Большая часть из них включается в универсальную систему многозначных алфавитов [1].

При моделировании ЦУ с помощью многозначных алфавитов отдельные элементы моделируются многозначными таблицами истинности, что приводит к увеличению времени моделирования и требуемого объема памяти.

Как указывалось ранее, в качестве моделей элементов ЦУ на логическом уровне используются формальные модели, в которых связь между входными и выходными сигналами элемента задается с помощью булевых уравнений либо таблиц истинности. Это относится прежде всего к простейшим базовым элементам цифровых схем И, ИЛИ, НЕ, сложению по модулю 2 и другим, в виде комбинаций которых может быть реализовано любое цифровое устройство. Более сложные элементы цифровых устройств — триггеры, регистры, устройства памяти и т. п. — в одних случаях представляются комбинациями простейших базовых компонентов, в других, на уровне регистровых передач, их описывают в терминах выполняемых ими операций, не прибегая к разложению на составляющие их простейшие элементы. Возможна и комбинация обоих подходов, когда в одних и тех же программах моделирования используются модели простейших базовых элементов цифровых схем и модели, более сложные на уровне регистровых передач.

Описание соединений элементов цифровых устройств в компьютере может быть выполнено в форме списков, логической сети либо таблиц в зависимости от конкретного языка описания ЦУ, используемого в программе.

Как следует из сказанного, математическая модель ЦУ будет представлять собой систему булевых уравнений, каждое из которых описывает один элемент ЦУ, либо из многомерных таблиц истинности, то есть той же самой системы булевых уравнений, но представленной в виде связанных таблиц. Такие математические модели могут быть организованы в компьютере различным образом.

В зависимости от способа организации модели ЦУ можно разделить на компилирующие и интерпретивные. В моделях компилирующего типа исходное описание ЦУ в виде логической сети или таблицы транслируется на язык машинных кодов и оформляется в виде объектного модуля, который затем и выполняется компьютером в процессе моделирования. Для такого перевода описания ЦУ на язык машинных кодов используется специальный компилятор, являющийся частью моделирующей программы. Достоинством такого подхода служит большая скорость работы программы моделирования, недостатком же — необходимость повторных компиляций при внесении изменений в ЦУ.

В моделях ЦУ интерпретивного типа связи между отдельными элементами ЦУ представляются в виде таблиц, описание ЦУ в машинные коды не переводится и каждое логическое уравнение, представляющее математическую модель того или иного элемента ЦУ, решается с помощью специальной подпрограммы. Выбор очередной подпрограммы производится специальной интерпретирующей программой, которая использует записанные в таблицу адреса перехода от одного элемента ЦУ к другому. Обращение к подпрограмме, моделирующей элемент, будет происходить всякий раз, когда интерпретатору понадобится выполнить данную логическую операцию. Модели интерпретивного типа оказываются более простыми и менее трудоемкими в разработке, однако они имеют меньшее быстродействие по сравнению с компилятивными.

В зависимости от того, учитываются или не учитываются задержки в срабатывании ЦУ при моделировании модели, ЦУ подразделяются на синхронные и асинхронные, а по способу организации процесса решения системы логических уравнений — на сквозные и событийные.

Описание особенностей всех видов схемных моделей ЦУ приводится в следующих разделах.

## 3.2. Синхронное моделирование цифровых устройств двоичным алфавитом

При синхронном моделировании ЦУ на логическом уровне не учитываются задержки срабатывания отдельных элементов, из которых состоит ЦУ. Это позволяет свести моделирование к последовательному вычислению сигналов на выходах элементов ЦУ по значениям сигналов на их входах с помощью булевых уравнений или таблиц истинности, моделирующих каждый элемент. При этом считается, что независимыми переменными, синхронизирующими работу модели устройства, являются события — изменение сигналов на входах ЦУ. После определения состояния, в которое перейдет ЦУ при очередной смене входных сигналов, осуществляется сдвиг модельного времени до момента наступления очередного события. В промежутке между двумя событиями смены сигналов на входах ЦУ не происходит, а если на входах устройства появятся новые сигналы, то они будут отнесены к началу следующего такта моделирования. Контроль временного интервала между событиями необходим для построения временных диаграмм, наглядно представляющих работу ЦУ в предположении нулевой длительности переходных процессов.

По принципу работы такая модель соответствует работе синхронного цифрового автомата, в котором сигналы на входы ЦУ поступают только в моменты подачи синхросигналов, в остальное время входные сигналы не могут изменять состояние ЦУ. Переходные процессы в устройстве обязательно заканчиваются к моменту прихода следующего синхросигнала.

Синхронные модели можно использовать для моделирования синхронных и асинхронных ЦУ, и термин «синхронное моделирование», общепринятый в литературе, означает только неучитывание задержек в работе отдельных элементов ЦУ.

Как показано в разделе 3.1, математическая модель ЦУ без учета задержек представляет собой систему булевых уравнений, связывающих между собой сигналы на входах и выходах элементов ЦУ. В общем виде такая система может быть записана следующим образом:

$$\begin{cases} \mathbf{U} = \mathbf{L}(\mathbf{U}_Q, \mathbf{U}); \\ \mathbf{F} = \mathbf{F}(\mathbf{U}), \end{cases} \quad (3.1)$$

где  $\mathbf{U}$  — вектор сигналов на внутренних узлах схемы ЦУ (входах и выходах элементов);  $\mathbf{L}$  — логический функционал;  $\mathbf{U}_Q$  — вектор сигналов на входах ЦУ;  $\mathbf{F}$  — вектор выходных сигналов, интересующих разработчика.

Система логических уравнений (3.1) при каждом событии — изменении входных сигналов — решается итерационно относительно вектора  $\mathbf{U}$ , по координатам которого определяются выходные сигналы устройства  $\mathbf{F}$ .

Алгоритмы решения системы логических уравнений делят на две группы: сквозные и событийные. В первом случае при итерационном решении системы логических уравнений математической модели ЦУ на каждой итерации решаются все логические уравнения и процесс решения системы  $\mathbf{U} = \mathbf{L}(\mathbf{U}_Q, \mathbf{U})$  во многом напоминает решение итерационными методами системы линейных алгебраических уравнений. Сквозное моделирование может выполняться методом простой итерации, когда для определения сигналов в ЦУ на последующей итерации  $\mathbf{U}_k$  используются сигналы на предыдущей итерации  $\mathbf{U}_{k-1}$ . Итерационный процесс решения системы логических уравнений по методу простой итерации может быть записан в виде

$$\mathbf{U}_k = \mathbf{L}(\mathbf{U}_Q, \mathbf{U}_{k-1}) \quad (3.2)$$

или в скалярной форме  $u_{ik} = L_i(\mathbf{U}_Q, u_{1,k-1}, \dots, u_{n,k-1})$ , где  $i = 1, \dots, n$ .

Если решение системы логических уравнений ведется по методу итерации Зейделя, то сигналы, определенные при решении предыдущих уравнений системы, используются для нахождения сигналов, определяемых последующими уравнениями. В этом случае итерационная формула в скалярной форме записи имеет вид

$$\begin{aligned} u_{i,k} &= L_i(\mathbf{U}_Q, u_{1,k}, \dots, u_{i-1,k}, u_{i+1,k-1}, \dots, u_{n,k-1}); \\ u_{i+1,k} &= L_{i+1}(\mathbf{U}_Q, u_{1,k}, \dots, u_{i,k}, u_{i+2,k-1}, \dots, u_{n,k-1}). \end{aligned} \quad (3.3)$$

Итерационный процесс решения системы логических уравнений продолжается до тех пор, пока сигналы на всех узлах ЦУ на двух последних итерациях не совпадут, что свидетельствует о получении установившегося состояния ЦУ. Для ускорения итерационного процесса расположение уравнений в системе логических уравнений математической модели ЦУ необходимо ранжировать в порядке распространения сигналов через элементы устройства.

При программной реализации алгоритмов значения сигналов на узлах ЦУ на каждой итерации  $\mathbf{U}_k$  хранятся в специальном массиве — рабочем поле РП( $k$ ), причем в рабочем поле, соответствующем нулевой итерации РП(0), хранятся сигналы, соответствующие начальному состоянию ЦУ перед изменением входных сигналов. Изменения входных сигналов записываются в рабочее поле на первой итерации, РП(1). На последующих итерациях до получения решения сигналы на входных узлах ЦУ при синхронном моделировании не изменяются. После получения установившегося решения сдвигается модельное время и начинается моделирование следующего события, а в качестве начального состояния ЦУ для него используется решение, полученное на предыдущем шаге.

Для простоты изложения рассмотрим процесс сквозного синхронного моделирования ЦУ двоичным алфавитом на примере устройства, приведенного на рис. 3.3, а.

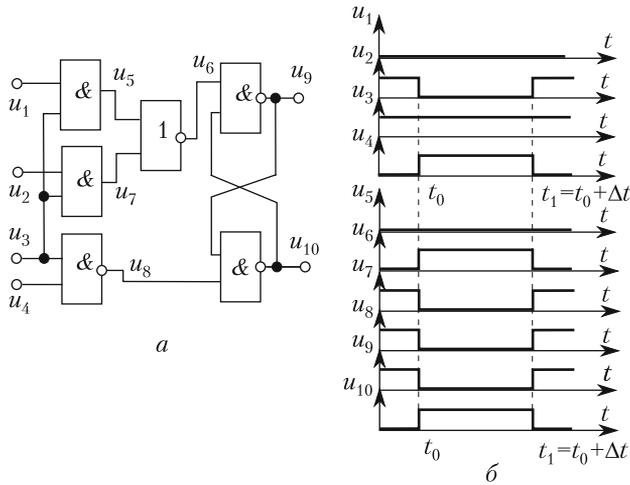


Рис. 3.3. ЦУ: *а* — схема; *б* — временные диаграммы ее работы

Математическая модель ЦУ представляет собой систему логических уравнений

$$\begin{aligned} u_5 &= u_1 \wedge u_3; & u_6 &= \overline{u_5 \vee u_7}; \\ u_7 &= u_2 \wedge u_3; & u_8 &= \overline{u_3 \wedge u_4}; \\ u_9 &= \overline{u_6 \wedge u_{10}}; & u_{10} &= \overline{u_8 \wedge u_9}, \end{aligned}$$

где  $u_i$  — сигналы на узлах устройства.

При сквозном моделировании по методу простой итерации в соответствии с (3.2) итерационные формулы будут записаны следующим образом:

$$\begin{aligned} u_{5,k} &= u_{1,k-1} \wedge u_{3,k-1}; & u_{6,k} &= \overline{u_{5,k-1} \vee u_{7,k-1}}; \\ u_{7,k} &= u_{2,k-1} \wedge u_{3,k-1}; & u_{8,k} &= \overline{u_{3,k-1} \wedge u_{4,k-1}}; \\ u_{9,k} &= \overline{u_{6,k-1} \wedge u_{10,k-1}}; & u_{10,k} &= \overline{u_{8,k-1} \wedge u_{9,k-1}}. \end{aligned}$$

Процесс сквозного моделирования ЦУ по методу простой итерации при изменении входных сигналов  $u_1, u_2, u_3, u_4$  (рис. 3.3, б) иллюстрируется табл. 3.1.

Временные диаграммы, соответствующие приведенным результатам моделирования, показаны на рис. 3.3, б.

Существенно сократить время моделирования можно за счет выполнения итерационного процесса по методу итерации Зейделя. В соответствии с (3.3) для нашего примера итерационные формулы будут записаны следующим образом:

$$\begin{aligned} u_{5,k} &= u_{1,k} \wedge u_{3,k}; & u_{6,k} &= \overline{u_{5,k} \vee u_{7,k-1}}; \\ u_{7,k} &= u_{2,k} \wedge u_{3,k}; & u_{8,k} &= \overline{u_{3,k} \wedge u_{4,k}}; \\ u_{9,k} &= \overline{u_{6,k} \wedge u_{10,k-1}}; & u_{10,k} &= \overline{u_{8,k} \wedge u_{9,k}}. \end{aligned}$$

Таблица 3.1

Изменение входных сигналов	Номер итерации	Сигналы на узлах ЦУ									
		входные				внутренние и выходные					
		$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$	$u_7$	$u_8$	$u_9$	$u_{10}$
$t = t_0$	0	0	1	1	0	0	0	1	1	1	0
$t = t_0$ , изменение сигналов	1	0	0	1	1	0	0	1	1	1	0
	2	0	0	1	1	0	0	0	0	1	0
	3	0	0	1	1	0	1	0	0	1	1
	4	0	0	1	1	0	1	0	0	0	1
Решение	5/0	0	0	1	1	0	1	0	0	0	1
$t = t_0 + \Delta t$ , изменение сигналов	1	0	1	1	0	0	1	0	0	0	1
	2	0	1	1	0	0	1	1	1	0	1
	3	0	1	1	0	0	0	1	1	0	1
	4	0	1	1	0	0	0	1	1	1	1
	5	0	1	1	0	0	0	1	1	1	0
Решение	6	0	1	1	0	0	0	1	1	1	0

Процесс сквозного моделирования ЦУ рис. 3.3, а по методу итерации Зейделя при таком же изменении входных сигналов, как и в предыдущем случае, приведен в табл. 3.2.

Таблица 3.2

Изменение входных сигналов	Номер итерации	Сигналы на узлах ЦУ									
		входные				внутренние и выходные					
		$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$	$u_7$	$u_8$	$u_9$	$u_{10}$
$t = t_0$	0	0	1	1	0	0	0	1	1	1	0
$t = t_0$ , изменение сигналов	1	0	0	1	1	0	0	0	0	1	1
	2	0	0	1	1	0	1	0	0	0	1
Решение	3/0	0	0	1	1	0	1	0	0	0	1
$t = t_0 + \Delta t$ , изменение сигналов	1	0	1	1	0	0	1	1	1	0	1
	2	0	1	1	0	0	0	1	1	1	0
Решение	3	0	1	1	0	0	0	1	1	1	0

Как видно из табл. 3.2, решение системы логических уравнений по методу итерации Зейделя позволяет существенно сократить число итераций, однако этот метод неприменим при учете задержек срабатывания элементов.

Приведем пошаговое описание алгоритма для разового изменения входных сигналов.

**Алгоритм 3.1.** Сквозное моделирование ЦУ двоичным алфавитом по методу простой итерации

*Шаг 1.* Формируется рабочее поле на нулевой итерации  $РП(0)$ , соответствующее начальному состоянию устройства.

*Шаг 2.* Фиксируется изменение входных сигналов, формируется рабочее поле на первой итерации  $РП(1)$ .

*Шаг 3.* Номер итерации  $k$  полагается равным 2.

*Шаг 4.* Начиная с первого логического уравнения системы уравнений математической модели ЦУ определяется новое значение сигналов на выходе элемента ЦУ, описываемого данным уравнением.

*Шаг 5.* Если сигнал на выходе элемента изменился, то в рабочее поле для  $k$ -й итерации  $РП(k)$  вносится новое значение сигнала, иначе переписывается сигнал из рабочего поля  $РП(k - 1)$ , соответствующего предыдущей,  $(k - 1)$ -й итерации.

*Шаг 6.* Если вычислены все сигналы на  $k$ -й итерации, то переходим к шагу 7, иначе к шагу 4.

*Шаг 7.* Сравниваются рабочие поля на  $k$ -й и  $(k - 1)$ -й итерациях. Если они совпадают, то моделирование отклика ЦУ на разовое изменение входных сигналов окончено, иначе увеличиваем номер шага итерации на 1 и переходим к шагу 4.

Значение сигнала на выходе элемента ЦУ может измениться только в том случае, если изменился сигнал по крайней мере на одном из его входов. Это обстоятельство положено в основу работы событийных алгоритмов синхронного моделирования ЦУ, в которых, в отличие от алгоритмов сквозного моделирования, на каждой итерации решаются не все логические уравнения системы, а только уравнения для тех элементов, у которых на предыдущей итерации изменились входные сигналы. Для организации такого вычислительного процесса в конце каждой итерации решения системы логических уравнений после сравнения рабочих полей на соседних итерациях формируется список изменившихся сигналов, по которому можно определить активизированные элементы ЦУ, то есть элементы, на входах которых изменился сигнал и для которых на последующей итерации необходимо решать логические уравнения. Для ускорения процесса моделирования в программе обычно решается задача определения очередности решения активизированных уравнений. Итерационный процесс прекращается, когда в модели ЦУ не будет изменившихся сигналов и активизированных элементов.

Процесс моделирования событийным алгоритмом ЦУ, схема которого представлена на рис. 3.3, *a*, приведен в табл. 3.3. Начальные условия, изменения входных сигналов и временные диаграммы совпадают с таковыми для предыдущих примеров.

Как видно из работы событийного алгоритма, здесь наиболее удобной представляется интерпретивная организация математической модели ЦУ, однако событийное моделирование может быть и компилятивным. Как и сквозное моделирование, событийное может выполняться методами простой итерации и итерации Зейделя.