МЕЖГОСУДАРСТВЕННЫЙ СОВЕТ ПО СТАНДАРТИЗАЦИИ, МЕТРОЛОГИИ И СЕРТИФИКАЦИИ
(МГС)

INTERSTATE COUNCIL FOR STANDARDIZATION, METROLOGY AND CERTIFICATION
(ISC)

МЕЖГОСУДАРСТВЕННЫЙ СТАНДАРТ

ГОСТ IEC 60848— 2016

ЯЗЫК СПЕЦИФИКАЦИЙ GRAFCET ДЛЯ ПОСЛЕДОВАТЕЛЬНЫХ ФУНКЦИОНАЛЬНЫХ СХЕМ

(IEC 60848:2013, IDT)

Издание официальное

Москва
Стандартинформ
2019

gost.e-ecolog.ru

Предисловие

Цели, основные принципы и основной порядок проведения работ по межгосударственной стандартизации установлены в ГОСТ 1.0—2015 «Межгосударственная система стандартизации. Основные положения» и ГОСТ 1.2—2015 «Межгосударственная система стандартизации. Стандарты межгосударственные, правила и рекомендации по межгосударственной стандартизации. Правила разработки, принятия, обновления и отмены»

Сведения о стандарте

1 ПОДГОТОВЛЕН Открытым акционерным обществом «Всероссийский научно-исследовательский институт сертификации» (ОАО «ВНИИС») на основе собственного перевода на русский язык англоязычной версии стандарта, указанного в пункте 5

2 ВНЕСЕН Федеральным агентством по техническому регулированию и метрологии

3 ПРИНЯТ Межгосударственным советом по стандартизации, метрологии и сертификации (протокол от 27 сентября 2016 г. № 91-П)

За принятие проголосовали:

Краткое наименование страны по МК (ИСО 3166) 004—97	Код страны по МК (ИСО 3166) 004—97	Сокращенное наименование национального органа по стандартизации
Армения	AM	Минэкономики Республики Армения
Беларусь	BY	Госстандарт Республики Беларусь
Казахстан	KZ	Госстандарт Республики Казахстан
Киргизия	KG	Кыргызстандарт
Россия	RU	Росстандарт

4 Приказом Федерального агентства по техническому регулированию и метрологии от 20 апреля 2017 г. № 294-ст межгосударственный стандарт ГОСТ IEC 60848—2016 введен в действие в качестве национального стандарта Российской Федерации с 1 сентября 2017 г.

5 Настоящий стандарт идентичен международному стандарту IEC 60848:2013 «Язык спецификаций GRAFCET для последовательных функциональных схем» («GRAFCET specification language for sequential function charts», IDT).

Международный стандарт разработан подкомитетом 3B «Документация» Технического комитета ТС 3 «Информационные структуры, документация и графические символы» Международной электротехнической комиссии (IEC).

6 ВВЕДЕН ВПЕРВЫЕ

7 ПЕРЕИЗДАНИЕ. Январь 2019 г.

Информация об изменениях к настоящему стандарту публикуется в ежегодном информационном указателе «Национальные стандарты», а текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ежемесячном информационном указателе «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (www.gost.ru)

© Стандартинформ, оформление, 2017, 2019

В Российской Федерации настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения . . . 1
2 Нормативные ссылки . . . 1
3 Термины и определения . . . 1
3.1 Термины GRAFCET . . . 1
3.2 Термины. Общие положения . . . 2
4 Общие принципы . . . 3
4.1 Контекст . . . 3
4.2 GRAFCET как язык спецификации алгоритма работы системы . . . 4
4.3 GRAFCET. Краткая презентация . . . 4
4.3.1 Общие положения . . . 4
4.3.2 Структура . . . 4
4.3.3 Элементы для интерпретации . . . 4
4.4 Правило синтаксиса . . . 6
4.5 Правила эволюции . . . 6
4.5.1 Общие положения . . . 6
4.5.2 Исходная ситуация . . . 6
4.5.3 Сброс перехода . . . 6
4.5.4 Эволюция активных этапов . . . 6
4.5.5 Одновременные эволюции . . . 6
4.5.6 Одновременная активация и деактивация этапа . . . 6
4.6 Входные события . . . 6
4.6.1 Общие положения . . . 6
4.6.2 Спецификация входных событий . . . 7
4.7 Внутренние события . . . 7
4.7.1 Общие положения . . . 7
4.7.2 Внутренние события, описываемые через этап активации . . . 7
4.7.3 Внутренние события, описываемые через этап деактивации . . . 7
4.7.4 Внутренние события, описываемые через сброс перехода . . . 7
4.8 Режимы выхода . . . 7
4.8.1 Общие положения . . . 7
4.8.2 Непрерывный режим (присваивание по состоянию) . . . 7
4.8.3 Режим хранения (локализация по событию) . . . 8
4.9 Применение правил эволюции . . . 8
4.9.1 Общие положения . . . 8
4.9.2 Эволюция не-перехода . . . 8
4.9.3 Эволюция перехода . . . 8
4.9.4 Результат эволюции перехода при присваивании . . . 9
4.9.5 Результат эволюции перехода при локализации . . . 9
4.10 Сравнение двух выходных режимов . . . 10
4.10.1 Общие положения . . . 10
4.10.2 Определение значения выходов . . . 10
4.10.3 Анализ значения выходов для диаграммы grafcet в определенный момент времени . . . 10

4.10.4 Операции, связанные с эволюцией перехода . . . 10
4.10.5 Возможный конфликт в значениях выходов . . . 10
5 Графическое отображение элементов . . . 11
6 Графическое представление последовательных структур . . . 22
6.1 Общие положения . . . 22
6.2 Основные структуры . . . 22
6.2.1 Последовательность . . . 22
6.2.2 Цикл одинарной последовательности . . . 23
6.2.3 Выбор последовательностей . . . 23
6.2.4 Пропуск этапа . . . 24
6.2.5 Пропуск обратной последовательности . . . 24
6.2.6 Активация параллельных последовательностей . . . 24
6.2.7 Синхронизация последовательностей . . . 24
6.2.8 Синхронизация и активация параллельных последовательностей . . . 25
6.3 Конкретные структуры . . . 25
6.3.1 Запуск последовательности от исходного этапа . . . 25
6.3.2 Завершение последовательности с помощью пит-этапа . . . 26
6.3.3 Запуск последовательности с исходным переходом . . . 27
6.3.4 Завершение последовательности пит-переходом . . . 27
7 Структурирование . . . 28
7.1 Общие положения . . . 28
7.2 Разбиение диаграммы grafcet . . . 28
7.2.1 Связанная диаграмма grafcet . . . 28
7.2.2 Парциальная диаграмма grafcet . . . 28
7.3 Структурирование с использованием подчинения парциальной диаграммы grafcet . . . 29
7.4 Структурирование с использованием вложения . . . 30
7.5 Структурирование с использованием макроэтапов . . . 34
Приложение А (справочное) Пример управления прессом . . . 35
Приложение В (справочное) Пример: Автоматическая система взвешивания-смешивания . . . 36
Приложение С (справочное) Взаимосвязь между языками спецификаций GRAFCET по IEC 60848 и SFC по IEC 61131-3 . . . 42
Библиография . . . 43

Введение

Настоящий стандарт ориентирован на инженеров-проектировщиков, инженеров по техническому обслуживанию и т. д., занимающихся спецификацией алгоритмов работы системы, например устройства контроля и управления системы автоматизации, компонента безопасности и т. п. Язык спецификаций GRAFCET должен также служить средством общения между конструкторами и пользователями автоматизированных систем.

МЕЖГОСУДАРСТВЕННЫЙ СТАНДАРТ

ЯЗЫК СПЕЦИФИКАЦИЙ GRAFCET ДЛЯ ПОСЛЕДОВАТЕЛЬНЫХ ФУНКЦИОНАЛЬНЫХ СХЕМ

GRAFCET specification language for sequential function charts

Дата введения — 2017—09—01

1 Область применения

Настоящий стандарт распространяется на язык спецификаций GRAFCET, предназначенный для функционального описания поведения последовательного контроля системы управления.

Настоящий стандарт устанавливает символы и правила для графического представления этого языка, а также для его интерпретации.

Настоящий стандарт был подготовлен для автоматизированных производственных систем промышленного назначения. Однако он может применяться и в других областях.

Методы разработки спецификаций с использованием языка GRAFCET выходят за рамки настоящего стандарта. Одним из методов является, например, «язык SFC», установленный в IEC 61131-3, который определяет набор языков программирования для программируемых контроллеров.

Примечание — Для получения дополнительной информации о взаимосвязи между IEC 60848 и языками интерпретации, такими как язык SFC по IEC 61131-3, см. приложение С.

2 Нормативные ссылки

Приведенные ниже документы являются обязательными для применения настоящего стандарта. Для датированных ссылок применяют только указанное издание, для недатированных — последнее издание приведенного стандарта (со всеми поправками к нему).

(Не действует.)

3 Термины и определения

В настоящем стандарте применены следующие термины с соответствующими определениями.

Примечание — Определения терминов в 3.1 применяются только в контексте языка спецификаций GRAFCET.

3.1 Термины GRAFCET

3.1.1 **операция** (action): Элемент языка GRAFCET, соответствующий одному этапу, показывающий действие, которое должно быть выполнено над выходными или внутренними переменными величинами.

3.1.2 **прямая связь** (directed link): Элемент языка GRAFCET, показывающий пути эволюции между этапами путем соединения этапов с переходами и переходов с этапами.

3.1.3 **диаграмма grafcet** (grafcet chart): Функциональная схема на языке спецификаций GRAFCET.

Примечание — «Диаграмма grafcet» может сокращенно называться «grafcet».

Издание официальное

3.1.4 **входное событие** (input event): Событие, характеризующееся изменением хотя бы одной из множества входных переменных величин последовательной части системы.

3.1.5 **внутреннее событие** (internal event): Событие, характеризующееся входным событием, ассоциированным с ситуацией в последовательной части системы.

3.1.6 **интерпретация** (interpretation): Часть языка спецификаций GRAFCET, позволяющая связывать между собой:

- входные переменные и структуру посредством условия перехода;
- выходные переменные и структуру посредством операций.

3.1.7 **ситуация** (situation): Состояние системы, описанное на языке спецификаций GRAFCET и характеризующееся активными шагами в данный момент.

3.1.8 **этап** (step): Элемент языка GRAFCET, используемый для определения состояния последовательной части системы.

Примечание 1 — Этап может быть активным или неактивным.

Примечание 2 — Совокупность активных этапов характеризует ситуацию системы.

3.1.9 **эволюция перехода** (transient evolution): Эволюция характеризуется сбросом в исходное состояние нескольких последовательных переходов при наступлении одного входного события.

3.1.10 **переход** (transition): Элемент языка GRAFCET, показывающий возможную эволюцию активности между двумя или более этапами.

Примечание — Возможная эволюция реализуется при сбросе перехода.

3.1.11 **условие перехода** (transition-condition): Элемент языка GRAFCET, ассоциированный с переходом и показывающий результат логического выражения.

Примечание — Условие перехода может быть либо «истинным», либо «ложным».

3.1.12 **переменная** (variable): Скалярная величина, определяемая ее именем и логическим, числовым значением.

3.1.13 **входная переменная** (input variable): Переменная, способная оказывать влияние на алгоритм работы, представленный диаграммой grafcet.

ПРИМЕР: Логическая переменная, показывающая нарушение температурного предела.

Примечание — Переменная может принадлежать окружающей среде или какому-либо другому компоненту системы.

3.1.14 **выходная переменная** (output variable): Переменная, определяемая алгоритмом работы, описываемым диаграммой grafcet.

ПРИМЕР: Заданная температура ПИД-регулятора (пропорционально-интегро-дифференцирующего регулятора).

Примечание — Переменная может принадлежать окружающей среде или какому-либо другому компоненту системы.

3.1.15 **внутренняя переменная** (internal variable): Переменная, используемая внутри диаграммы grafcet и невидимая для других компонентов системы и окружающей среды.

ПРИМЕРЫ: Этап переменной X* (символ 2.1 из таблицы 1), длительность этапа T* (символ 2.2 из таблицы 1), счетчик цикла в диаграмме grafcet.

3.2 Термины. Общие положения

3.2.1 **диаграмма, граф** (chart, graph): Графическое представление описания алгоритма работы системы, например соотношения между двумя или более переменными величинами, операциями или состояниями.

3.2.2 **структура** (structure): Часть языка спецификаций GRAFCET, позволяющая описывать возможную эволюцию между ситуациями.

3.2.3 **система** (system): Совокупность взаимосвязанных элементов, рассматриваемых в определенном контексте как одно целое и отделенных от окружающей их среды.

Примечание 1 — Такими элементами могут быть материальные и концептуальные объекты, а также их результаты (например, формы организации, математические методы, языки программирования).

Примечание 2 — Система считается отделенной от окружающей среды и от других внешних систем посредством виртуальной поверхности, что позволяет сократить количество связей между ними и системой.

Примечание 3 — Язык GRAFCET может быть использован для описания логики работы системы любого вида.

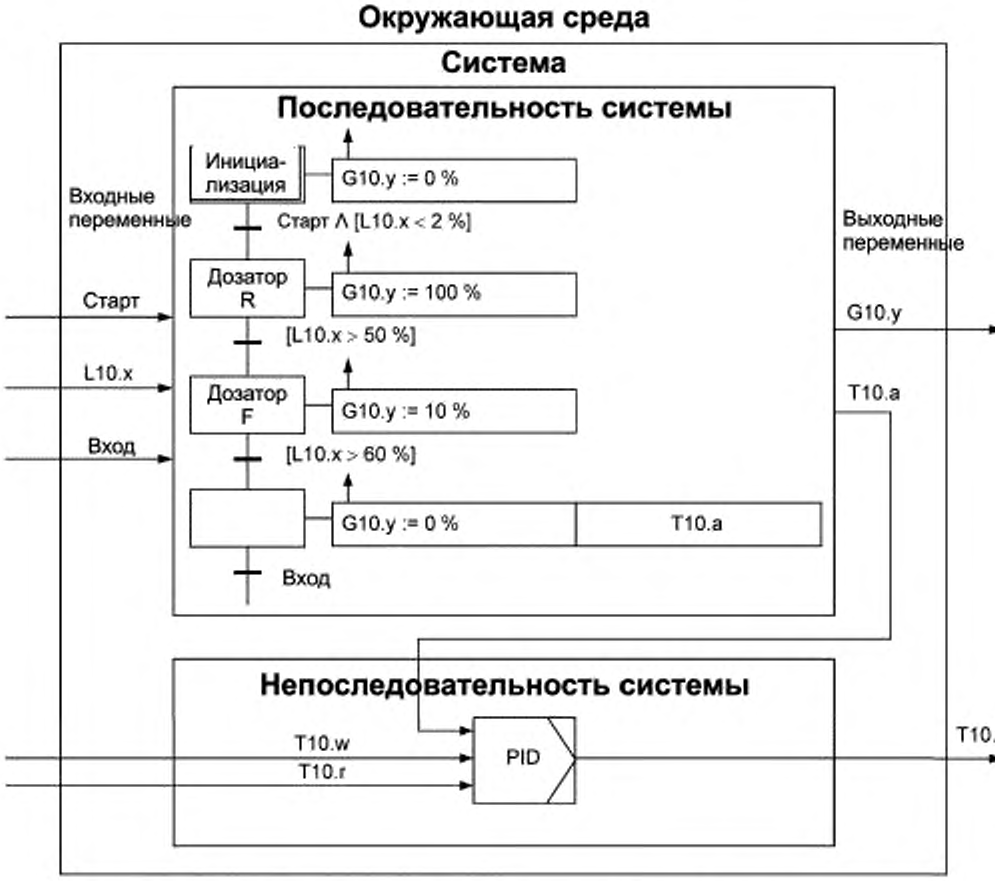
[IEC 60050-351:1998, 351-11-01]

4 Общие принципы

4.1 Контекст

Реализация автоматизированной системы требует, в частности, описания связи между причиной и следствием. Чтобы выполнить это требование, ниже будет описана логическая составляющая желаемого поведения системы.

Логической составляющей физической системы является ее последовательная часть (см. рисунок 1). Алгоритм работы указывает, каким образом выходные переменные зависят от входных переменных. Целью диаграммы grafcet является спецификация алгоритма работы последовательной части системы.



L10.x — уровень бака; T10.a — температурный контур — автоматический режим; T10.r — температурный контур — заданное значение; G10.y — дозирующий клапан — положение; T10.w — температурный контур — точка установки; T10.y — температурный контур — регулируемая величина

Рисунок 1 — Графическое представление последовательной части системы

4.2 GRAFCET как язык спецификации алгоритма работы системы

Язык спецификаций GRAFCET позволяет создать диаграмму grafcet, показывающую ожидаемое поведение рассматриваемой последовательной системы. Этот язык характеризуется главным образом его графическими элементами, которые, будучи связанными с буквенно-цифровым выражением переменных, обеспечивают синтетическое представление об алгоритме работы на основе косвенного описания ситуации в системе.

Описание алгоритма работы в состояниях заключается в следующем: «односимвольные» состояния соответствуют ситуациям диаграммы grafcet, подразумевающим уникальность ситуации в данный момент. Состояния соединены друг с другом посредством условия эволюции, что позволяет описывать переход от одной ситуации к другой ситуации.

Из соображений удобства описание поведения системы на основе состояний целесообразнее заменить описанием на основе этапов. На диаграмме grafcet могут быть активны одновременно несколько этапов, тогда ситуация будет характеризоваться совокупностью активных этапов в рассматриваемый момент. Эволюция одной совокупности этапов к другой осуществляется посредством одного или нескольких переходов, каждый из которых характеризуется:

- его предыдущими этапами;
- его последующими этапами;
- ассоциированным с ним условием перехода.

Примечание — Указанные причины приводят к правилу синтаксиса, определяющего чередование этап-переход.

4.3 GRAFCET. Краткая презентация

4.3.1 Общие положения

Язык спецификаций GRAFCET используется для конструирования диаграмм grafcet, обеспечивающих графическое и синтетическое отображение алгоритма работы последовательных систем. Отображение (см. рисунок 2) может быть реализовано в форме:

- структуры, допускающей возможные эволюции между описываемыми ситуациями;
- интерпретации, определяющей связь между входными, выходными переменными и структурой (для реализации данной интерпретации необходимы правила эволюции, присвоения и локализации).

Символы, связанные с элементами GRAFCET, представляющие этапы процесса и связи между этапами, приведены и проиллюстрированы примерами в таблицах 1—4 раздела 5 настоящего стандарта.

4.3.2 Структура

Структура включает в себя следующие базовые элементы:

- Этап (определение 3.1.8, символ 1). Этап может быть активным или неактивным, совокупность активных этапов в диаграмме grafcet в любой данный момент времени представляет ситуацию этой диаграммы grafcet в этот момент.
- Переход (определение 3.1.10, символ 7). Переход показывает, что между двумя или более этапами может развиваться эволюция активности. Эта эволюция реализуется при сбросе перехода.
- Прямая связь (определение 3.1.2, символ 10). Прямая связь соединяет один или несколько этапов с переходом или переход с одним или несколькими этапами.

4.3.3 Элементы для интерпретации

Для интерпретации используют следующие элементы:

- Условие перехода (определение 3.1.11, символ 13). Условие перехода, связанное с каждым переходом, представляет собой логическое выражение, которое является «истинным» или «ложным» и которое состоит из входных переменных и/или внутренних переменных.
- Операция (определение 3.1.1). Операция в прямоугольнике показывает, что следует сделать с выходной или внутренней переменной либо путем присваивания (непрерывная операция, символ 20), либо локализации (операция хранения, символ 26).

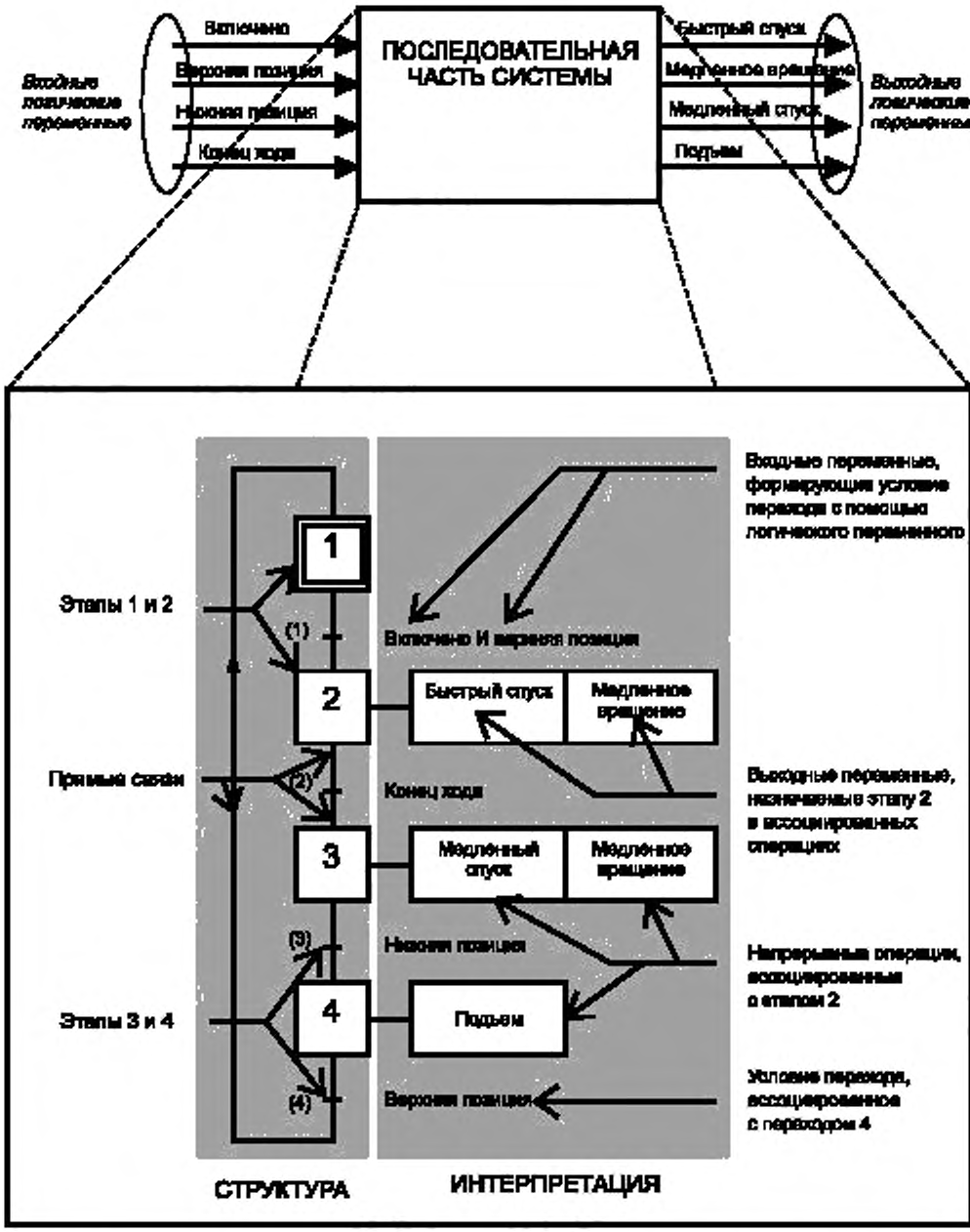


Рисунок 2 — Элементы структуры и интерпретации, используемые в диаграмме grafcet для описания поведения последовательной части системы, определяемой ее входными и выходными переменными

4.4 Правило синтаксиса

Чередование этап-переход и переход-этап всегда должно соблюдаться независимо от последовательности.

Следствия:

- два этапа никогда не будут соединены ни прямой, ни косвенной связью;
- прямая связь должна соединять только этап с переходом или переход с этапом.

4.5 Правила эволюции

4.5.1 Общие положения

Так как каждая ситуация характеризуется множеством активных этапов в данный момент времени, то правила эволюции grafcet влияют только на поэтапное применение принципа эволюции между ситуациями последовательной части системы.

4.5.2 Исходная ситуация

Исходной ситуацией является ситуация в начальный момент времени. Следовательно, она характеризуется множеством этапов, активных в этот момент. Выбор ситуации в начальный момент времени зависит от методологии и типа последовательной части рассматриваемой системы.

Правило 1: Исходная ситуация, выбранная конструктором, представляет собой ситуацию в начальный момент времени.

4.5.3 Сброс перехода

Правило 2: Переход называют разрешенным, если все непосредственно предшествующие этапы, связанные с этим переходом, активны. Сброс перехода происходит:

- когда переход разрешен и
- когда связанное с ним условие перехода «истинно».

4.5.4 Эволюция активных этапов

Правило 3: Сброс перехода одновременно инициирует активацию всех ближайших последующих этапов и деактивацию всех ближайших предшествующих этапов.

4.5.5 Одновременные эволюции

Эволюция между двумя активными ситуациями означает, что ни одна промежуточная ситуация не является возможной и изменение от одного отображения ситуации с помощью множества этапов к другому отображению происходит мгновенно.

Правило 4: Несколько переходов, которые могут быть сброшены одновременно, одновременно и сбрасываются.

4.5.6 Одновременная активация и деактивация этапа

Если этап участвует в описании предшествующей ситуации и в следующей за ним, он может, следовательно, только оставаться активным.

Правило 5: Если во время операции активный этап одновременно активируется и деактивируется, он остается активным.

4.6 Входные события

4.6.1 Общие положения

Правила эволюции показывают, что вызвать эволюцию диаграммы grafcet может только изменение значений входных переменных. Такое изменение, называемое «входным событием», определяется предыдущим значением и последующим значением всех входных переменных, характеризующих это одно событие. На практике множество входных событий специфицируется только изменением характеристического состояния (прямой переход или обратный переход) одной или нескольких логических входных переменных.

Примечание — Прямой перепад логической переменной, обозначаемый значком «↑» перед логической переменной, означает, что этот прямой перепад является истинным только для перехода рассматриваемой переменной от значения 0 к значению 1. Обратный переход логической переменной, обозначаемый значком «↓» перед логической переменной, означает, что этот обратный переход является истинным только для перехода рассматриваемой переменной от значения 1 к значению 0.

Говорят, что «событие происходит» на момент изменения состояния входных переменных, которые его характеризуют.

4.6.2 Спецификация входных событий

Спецификация входных событий осуществляется с помощью логического выражения одной или нескольких характеристических переменных, часто в условии перехода. Реже, она может также напрямую указывать внутреннее событие (см. 4.7).

↑a ПРИМЕР 1:

Выражение «↑a» описывает множество всех входных событий, для которых предыдущее значение входной переменной «a» равно 0 и его последующее значение равно 1 независимо от значения других входных переменных системы.

a·↑b ПРИМЕР 2:

Выражение «a·↑b» описывает множество всех входных событий, для которых последующее значение входной переменной «a» равно 1, а предшествующее значение входной переменной «b» равно 0, и ее последующее значение равно 1 независимо от значения других входных переменных системы.

a ПРИМЕР 3:

Выражение «a» описывает множество всех входных событий, для которых последующее значение входной переменной «a» равно 1 независимо от значения других входных переменных системы.

Примечание — Будучи использованным в условии перехода, это выражение может привести к эволюции перехода (см. 3.12).

4.7 Внутренние события

4.7.1 Общие положения

Только некоторые входные события могут произойти из данной ситуации. Связь между ситуацией и входным событием, которое может произойти из этой ситуации, называют внутренним событием (см. 3.6). Данное понятие в основном используется конструктором, чтобы определить выходную локализацию на множестве внутренних событий (см. 4.8.3). Описание множества внутренних событий осуществляется одним из следующих способов.

4.7.2 Внутренние события, описываемые через этап активации

Этап активации, помеченный графически (символ 27), описывает множество внутренних событий, каждое из которых имеет этот этап активации в качестве последствия.

4.7.3 Внутренние события, описываемые через этап деактивации

Помеченный графически этап деактивации (символ 28) описывает множество внутренних событий, каждое из которых имеет этот этап деактивации в качестве последствия.

4.7.4 Внутренние события, описываемые через сброс перехода

Помеченный графически сброс перехода (символ 29) описывает множество внутренних событий, каждое из которых имеет сброс этого перехода в качестве последствия.

4.8 Режимы выхода

4.8.1 Общие положения

Операции включают связи, устанавливающие соединение между эволюцией диаграммы grafcet и выходами. Два режима выхода, непрерывный режим и режим хранения, характеризуют зависимость выходов от эволюции и от входов системы.

4.8.2 Непрерывный режим (присваивание по состоянию)

В непрерывном режиме ассоциация операции с этапом показывает, что выходная переменная имеет «истинное» значение, если этап активен и если проверено условие присваивания. Условие присваивания представляет собой логическое выражение от входных переменных и/или внутренних переменных (см. символ 22). Если хотя бы одно из условий не выполняется и при условии, что никакая другая операция, относящаяся к тем же самым выходам, не отвечает условиям, соответствующая выходная переменная принимает значение «ложно».

Присваивание относится к назначению выходным переменным конкретных значений («истинно» или «ложно»).

Множество локальных присваиваний (по отношению к активным этапам в данный момент) определяет присваивание всех выходных переменных для этой ситуации.

Правило присваивания: для данной ситуации значениям выходов, связанных с непрерывными операциями присваивается:

- значение «истинно» для каждого выхода, относящегося к операциям, связанным с активными этапами, и для которых подтверждены условия присваивания;
- значение «ложно» для других выходов (которым не присвоено «истинное» значение).

4.8.3 Режим хранения (локализация по событию)

В режиме хранения ассоциация операции с внутренними событиями используется для указания, что выходная или внутренняя переменная принимает и поддерживает принудительно присвоенное значение, если осуществляется одно из этих событий.

Для описания ассоциации операций с событиями (этап активации, этап деактивации, сброс перехода и т. д.) необходимы представления в явной форме.

Значение выходной или внутренней переменной, относящейся к операции хранения, остается неизменным до нового оговоренного события, изменяющего его значение.

Локализация относится к хранению в рассматриваемый момент определенного значения, оказавшего влияние на выходную или внутреннюю переменную.

Правило локализации: значение выходной или внутренней переменной, относящееся к операции хранения и связанной с событием, распределяется на указанную переменную, если происходит обусловленное внутреннее событие; значение этой переменной «ложно» (логическая переменная) или равно нулю (числовая переменная) при инициализации.

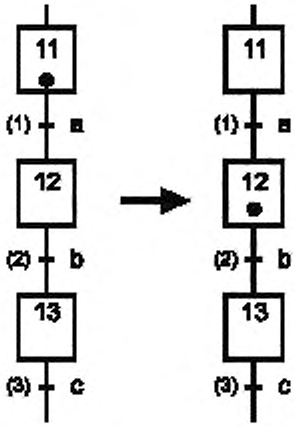
4.9 Применение правил эволюции

4.9.1 Общие положения

Интуитивно понятное толкование эволюции, именуемое «шаг за шагом», обозначает прогрессивный способ, позволяющий при возникновении входного события и с учетом предшествующей ситуации определять следующую ситуацию для этого события путем последовательного применения правил эволюции к каждому переходу. Объектом интерпретации является устройство, позволяющее косвенную детализацию эволюции, но конструктор должен позаботиться о том, чтобы сброс переходов на этом пути не затрагивал эффективную активацию промежуточных ситуаций.

4.9.2 Эволюция не-перехода

Понятие «эволюция не-перехода» означает, что входное событие приводит только к одной стадии эволюции (одновременный сброс одного или нескольких переходов).



ПРИМЕР: «Эволюция не-перехода»

Предшествующая ситуация: этап 11 активен, a = 0, b = 0 и c = 0.

Интуитивно понятное толкование эволюции:

Изменение в значении «a» включает сброс перехода (1) и активацию этапа 12, переход (2) не может быть сброшен, поскольку b = 0, последующая ситуация будет: этап 12 активен.

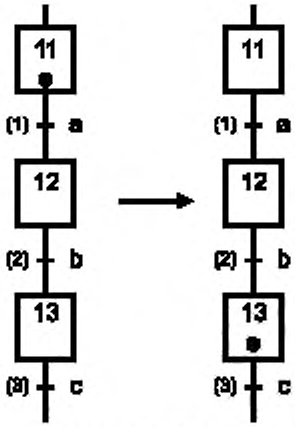
Реальная интерпретация эволюции:

Появление одного из входных событий, таких как изменение «a» от 0 к 1, ведет прямо к следующей ситуации: этап 12 активен.

4.9.3 Эволюция перехода

В некоторых случаях применение правил эволюции может привести к последовательному сбросу некоторых переходов (в нескольких этапах эволюции), если условия переходов, ассоциированные с последующими переходами, уже имеют состояние «истинно», когда первые рассматриваемые переходы, сбрасываются. Соответствующее описание, рассматриваемое как переходное, использует путь, показывающий развитие от предшествующей ситуации к последующей ситуации (см. 3.9).

Соответствующие промежуточные этапы, рассматриваемые как нестабильные, не активируются, тем не менее их рассматривают как бы «виртуально» активированными и деактивированными на пути интуитивной эволюции, так же как и соответствующие переходы, которые были «виртуально» сброшены.



ПРИМЕР: «Эволюция перехода»

Предшествующая ситуация: этап 11 активен a = 0, b = 1 и c = 0.

Интуитивно понятное толкование эволюции:

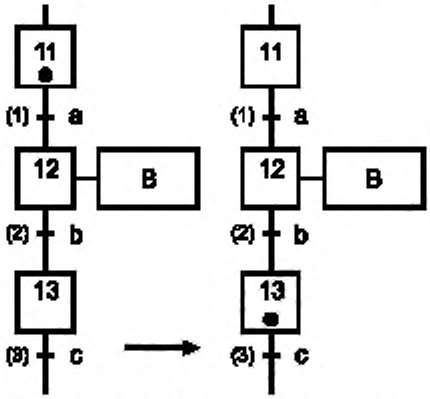
Изменение в значении «a» включает сброс перехода (1) и виртуальную активацию этапа 12, затем виртуально сбрасывается переход (2), поскольку b = 1, что приводит к последующей ситуации: этап 13 активен.

Реальная интерпретация эволюции:

Появление одного из входных событий, таких как изменение «a» от 0 к 1, ведет к следующей ситуации: этап 13 активен.

4.9.4 Результат эволюции перехода при присваивании

Присваивание выходного значения при непрерывной операции, ассоциированной с этапом, который является нестабильным этапом в случае эволюции перехода, не оказывает влияния на результат, так как этап на самом деле не активирован (см. 4.8.2).



ПРИМЕР: «Непрерывная операция, ассоциированная с нестабильным этапом»

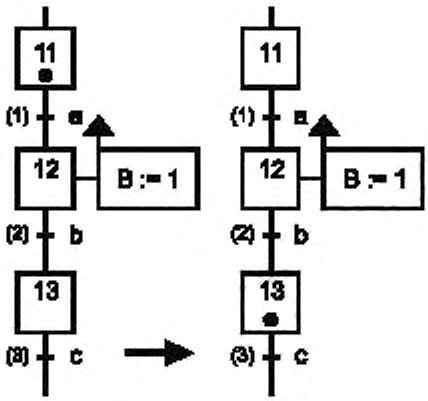
Предшествующая ситуация: этап 11 активен, a = 0, b = 0 и c = 0.

Появление одного из входных событий, таких как изменение «a» от 0 к 1, ведет прямо к следующей ситуации: этап 13 активен.

Предшествующая ситуация (этап 11 активен) и последующая ситуация (этап 13 активен) присваивают выходной переменной B значение 0. Нестабильный этап 12 реально не активируется, присваивания B значения 1 при эволюции перехода не происходит.

4.9.5 Результат эволюции перехода при локализации

Локализация выхода к определенному значению при операции хранения (символ 26), ассоциированной с этапом, являющимся нестабильным этапом в случае эволюции перехода, является результативным, поскольку данная локализация ассоциируется с событиями, реализующими эту эволюцию (см. 4.8.3).

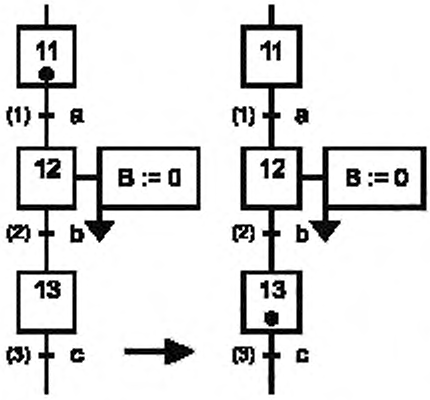


ПРИМЕР 1: «Операция хранения, ассоциированная с активацией нестабильного этапа»

Предыдущая ситуация: этап 11 активен, a = 0, b = 1 и c = 0.

Появление одного из входных событий, таких как изменение «a» от 0 к 1, ведет прямо к следующей ситуации: этап 13 активен.

Локализация значения 1 по выходной переменной B реализуется при наступлении одного из входных событий, имеющих следствием реальную или виртуальную деактивацию этапа 12.



ПРИМЕР 2: «Операция хранения, ассоциированная с деактивацией нестабильного этапа»

Предыдущая ситуация: этап 11 активен, a = 0, b = 1 и c = 0.

Появление одного из входных событий, таких как изменение «a» от 0 к 1, ведет прямо к следующей ситуации: этап 13 активен.

Локализация значения 0 по выходной переменной B реализуется при наступлении одного из входных событий, имеющих следствием реальную или виртуальную деактивацию этапа 12.

4.10 Сравнение двух выходных режимов

4.10.1 Общие положения

Выбор выходного режима зависит от используемой практики и методологии. Тем не менее конструктор должен обратить внимание на существенные различия между этими двумя режимами.

4.10.2 Определение значения выходов

В зависимости от выбранного режима величина выходов определяется следующим образом:

- в непрерывном режиме всем выходам присваивают значение в соответствии с ситуацией, «истинное» значение для выходов, однозначно указанных в операциях, ассоциированных с активными этапами, и «ложное» значение для остальных выходов, которые неявно устанавливаются в результате отсутствия действия (см. правило присваивания 4.8.2);
- в режиме хранения только соответствующие выходы изменяются в соответствии с указанным значением, другие сохраненные значения выходов остаются неизменными (см. правило локализации, 4.8.3).

4.10.3 Анализ значения выходов для диаграммы grafcet в определенный момент времени

В зависимости от выбранного режима анализ значения выходов получается следующим:

- в непрерывном режиме знание ситуации и значения входов является достаточным для определения значения выходов (см. 4.8.2);
- в режиме хранения знание ситуации и значения входов параметров не является достаточным, предшествующие эволюции также должны быть известны, чтобы определить значение выходов (см. 4.8.3).

4.10.4 Операции, связанные с эволюцией перехода

В зависимости от выбранного режима операции, связанные с эволюцией перехода, описываются следующим образом:

- в непрерывном режиме операции, ассоциированные с нестабильным этапом, не принимаются во внимание, поскольку этот этап не является активированным (см. 4.9.2);
- в режиме хранения операции, ассоциированные с событиями и связанные с эволюцией перехода, учитываются, поскольку происходят инициированные события, реализуя данную эволюцию (см. 4.9.3).

4.10.5 Возможный конфликт в значениях выходов

В зависимости от выбранного режима возможный конфликт в значениях выходов управляется таким образом:

- в непрерывном режиме принципы присваивания гарантируют отсутствие конфликта присваивания на конкретном выходе;
- в режиме хранения правила локализации не позволяют возникнуть конфликту присваивания на том же выходе. Конструктор должен убедиться, что две противоположные локализации не могут быть выполнены одновременно.

Примечание 1 — Оба выходных режима могут быть использованы в одной спецификации GRAFCET, но значение выходной переменной определяется либо присваиванием, либо локализацией. Спецификация локализации по выходной переменной (режим хранения) исключает эту выходную переменную из любого присваивания (непрерывный режим).

Примечание 2 — В разделе 5 настоящего стандарта приведены графические символы, разрешающие операции хранения (отображаемые в явном виде в соответствии с множеством определенных событий), чтобы отличать от непрерывных операций (что выражается в отсутствии какого-либо отображения).

Примечание 3 — В большинстве случаев спецификация алгоритма работы системы управления, исходя из современной производственной практики, требует использования непрерывного режима для всех булевых выходов на исполнительные устройства и режима хранения для описания задач внутреннего контроля. Эти задачи, такие как приращения счетчика или модификации значения цифрового регистра, не обязательно относятся к внутренним логическим переменным. Внутренние задачи, связанные с операциями хранения, а также вычисление выражений, связанных с условиями переходов, не описаны в настоящем стандарте, но связаны с использованием логического описания grafcet эволюции.

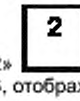
5 Графическое отображение элементов

Элементы GRAFCET имеют свои собственные символические отображения, которые при правильном объединении позволяют реализовать понятные синтетические функциональные диаграммы.

Примечание 1 — Задано только общее отображение символов; размеры и детали (толщина линий, параметры шрифтов и т. д.) оставлены на выбор пользователям.

Примечание 2 — Пунктирное изображение указывает на контекст символа.

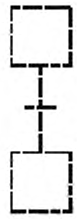
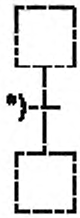
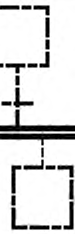
Таблица 1 — Этапы

Номер	Символ	Описание
[1]		**Этап:** В данный момент этап является либо активным, либо неактивным. Множество активных этапов определяет ситуацию данной системы в рассматриваемый момент. Соотношение высота-ширина прямоугольника может быть произвольным, хотя рекомендуется форма квадрата. Для целей идентификации этапы должны иметь метку, например буквенно-цифровой символ. Метка, присвоенная этапу, предназначена заменить звездочку в верхней половине общего символа. ПРИМЕР 1: «Этап 2»  ПРИМЕР 2: «Этап 3, отображенный в активном состоянии»  Примечание — Представляется целесообразным указывать, какие этапы активны в данный момент, отмечая эти этапы точкой. Точка не является частью символа этапа и используется только в пояснительных целях.
[2.1]	X*	**Переменная этапа:** Активное или неактивное состояние этапа может быть отображено логическими значениями «1» или «0» соответственно булевой переменной X*, в обозначении которой звездочку * заменяют меткой соответствующего этапа. ПРИМЕР: «Переменная этапа 8» X8
[2.2]	T*	**Продолжительность этапа:** Длительность активного этапа может быть представлена значением переменной времени T*, в обозначении которой звездочку * заменяют меткой соответствующего этапа. ПРИМЕР: «Продолжительность этапа fill» TFill
[3]		**Начальный этап:** Данный символ означает, что этот этап участвует в исходной ситуации. Примечание 1 — Применяются правила символа 1. Примечание 2 — Начальный этап может быть «нестабильным», см. 4.9.3. ПРИМЕР: «Начальный этап 12» 

Окончание таблицы 1

Номер	Символ	Описание
[4]		**Охватывающий этап:** Символ указывает, что данный этап содержит другие этапы, которые называют вложенными этапами. Примечание 1 — Применяются правила символа 1. Примечание 2 — Свойства и примеры использования охватывающего этапа приведены в 7.4.
[5]		**Начальный охватывающий этап:** Символ означает, что данный охватывающий этап участвует в исходной ситуации. Примечание — Начальный охватывающий этап содержит по меньшей мере один вложенный начальный этап.
[6]		**Макроэтап:** Уникальное отображение детализированной части функциональной диаграммы, называемое расширением макроэтапа. Примечание — Свойства и примеры использования макроэтапа приведены в 7.5.

Таблица 2 — Переходы

Номер	Символ	Описание
[7]		**Переход от одного этапа к другому:** Переход представлен линией, перпендикулярной звену связи, соединяющему два этапа. Примечание 1 — Переход разрешен, когда непосредственно предшествующий этап активен (см. правило эволюции № 2, 4.5.3). Примечание 2 — Только один переход возможен между двумя этапами (см. 4.4). Примечание 3 — Для удобства графического отображения допустимо размещать переходы на горизонтальных звеньях прямой связи (см. рисунок B.5, парциальная grafcet G1). Примечание 4 — Символика переходов не является предметом настоящего стандарта. Переходы могут быть описаны простым текстом, логическими выражениями, логическими схемами и т. д.
[8]		**Обозначение перехода:** Переход может иметь обозначение, размещаемое, как правило, с левой стороны, которое не должны быть ошибочно принято за ассоциированное условие перехода. Звездочку заменяют буквенно-цифровым обозначением перехода.
[9]		**Синхронизация предшествующего и/или последующего перехода:** Когда несколько этапов подсоединены к одному и тому же переходу, прямые связи от этих переходов и/или к этим переходам группируют, чтобы последующий или предшествующий символ синхронизации отобразить двойной горизонтальной линией. Примечание — Символ синхронизации соответствует ISO 5807:1985, 9.2.2.5.

Окончание таблицы 2

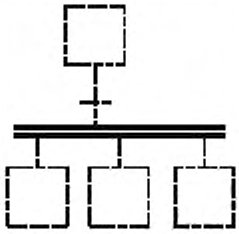
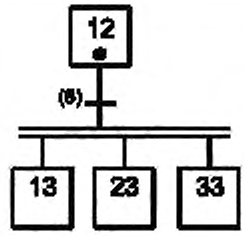
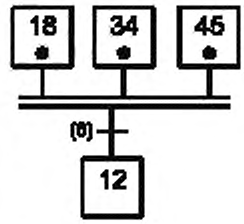
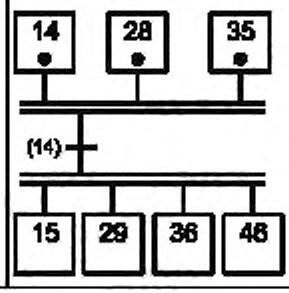
Номер	Символ	Описание
[9]		 **ПРИМЕР 1: Переход от одного этапа** (12) к нескольким (13, 23, 33). Переход (8) разрешен, если этап 12 является активным.  ПРИМЕР 2: Переход от нескольких этапов (18, 34, 45) к одному этапу (12). Переход (6) разрешен только тогда, когда все предшествующие этапы активны.  ПРИМЕР 3: Переход от нескольких этапов (14, 28, 35) к нескольким этапам (15, 29, 36, 46). Переход (14) разрешен только тогда, когда все предшествующие этапы активны.

Таблица 3 — Прямые связи

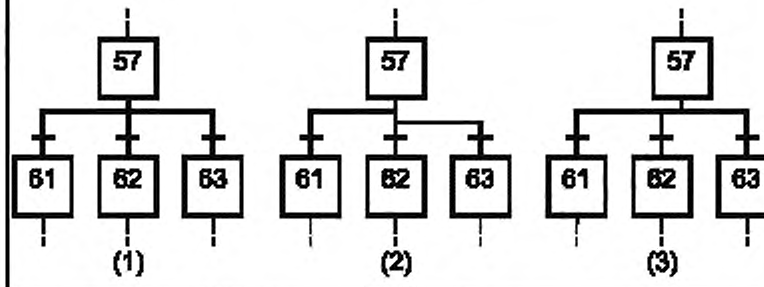
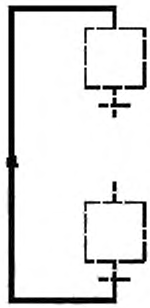
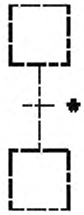
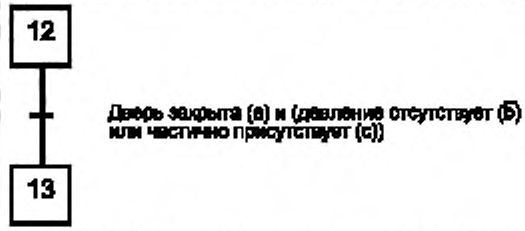
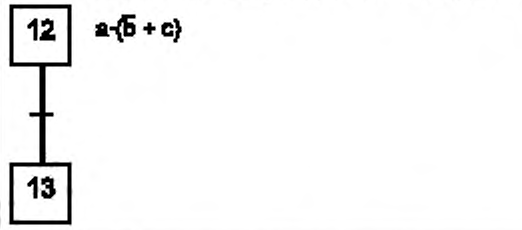
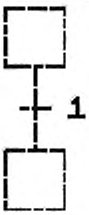
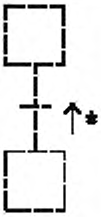
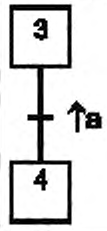
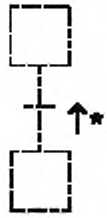
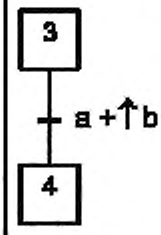
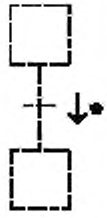
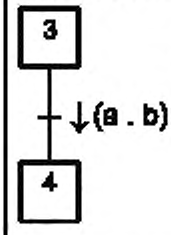
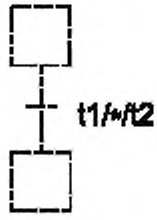
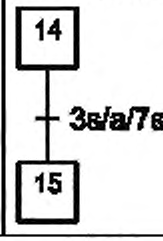
Номер	Символ	Описание
[10]		**Прямая связь от верхней точки к нижней точке:** Эволюцию путей между этапами обозначают прямыми связями, соединяющими этапы с переходами и переходы с этапами. Прямые связи могут быть горизонтальными или вертикальными. Диагональные связи допустимы только в тех редких случаях, когда они способствуют наглядности диаграммы. Пересечения вертикальных и горизонтальных связей разрешены, если отсутствует взаимозависимость между этими связями. Соответственно, таких пересечений следует избегать, когда связи относятся к одной и той же эволюции. ПРИМЕР: Хотя все три представления допустимы, но рекомендуются представления 2 и 3, чтобы избежать перепутывания связями при наличии взаимозависимости и без нее. 
[11]		**Прямая связь от нижней точки к верхней точке:** По определению, направление эволюции всегда идет от верхней точки к нижней точке. Если это соглашение не соблюдается или если требуется обеспечить более четкое понимание диаграммы, то используют стрелки.
[12]		**Связующая метка:** Если прямая связь должна быть прервана (например, в сложных диаграммах или когда одна диаграмма занимает несколько страниц), то должны быть указаны номер этапа назначения и номер страницы, на которой он находится. Звездочку заменяют соответствующей меткой. ПРИМЕР: Эволюция к этапу 83 на странице 13. 

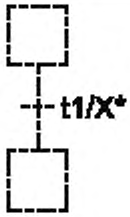
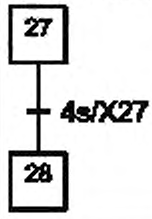
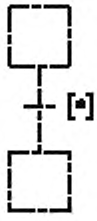
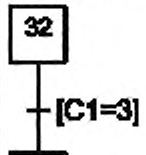
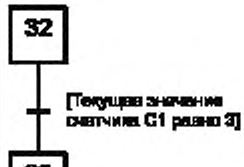
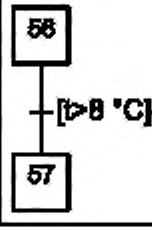
Таблица 4 — Ассоциированные условия перехода

Номер	Символ	Описание
[13]		**Условие перехода:** Логическое выражение, называемое условием перехода, которое может быть «истинным» или «ложным», ассоциировано с каждым переходом. Если существует соответствующая логическая переменная, то она равна 1, когда условие перехода «истинно», или равна 0, когда условие перехода «ложно». Логическое выражение, формирующее условие перехода, содержит одну или несколько логических переменных (входная переменная, переменная этапа, значение предиката и т. д.). Звездочку заменяют описанием условия перехода в виде текста, логического выражения или с помощью графических символов. ПРИМЕР 1: Условие перехода записано в виде текста.  ПРИМЕР 2: Условие перехода записано в виде логического выражения. 
[14]		**Условие перехода всегда «истинно»:** Символ «1» означает, что условие перехода всегда «истинно». Примечание — В этом случае эволюцией должен быть переход (см. 4.9.3), сброс перехода обусловливается только активностью предшествующего этала.
[15]		**Положительный перепад логической переменной:** Символ «↑» означает, что условие перехода верно только при изменении состояния переменной * (положительный перепад: изменение от значения 0 к значению 1, см. примечание в 4.6). Данный символ является общим и распространяется на все логические высказывания либо для элементарной переменной, либо для множества из нескольких логических переменных.  ПРИМЕР 1: Ассоциированное условие перехода «истинно» только при изменениях от состояния 0 к состоянию 1. Примечание — В соответствии с правилом эволюции № 2 переход сбрасывается только на положительном перепаде после того, как переход был переведен в активное состояние на этапе 3.

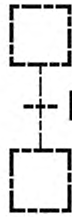
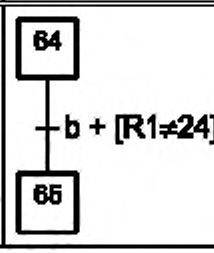
Продолжение таблицы 4

Номер	Символ	Описание
[15]		 ПРИМЕР 2: Ассоциированное условие перехода «истинно» только тогда, когда значение переменной «истинно» или когда значение b изменяется от состояния 0 к состоянию 1.
[16]		**Отрицательный перепад логической переменной:** Символ «↓» означает, что условие перехода истинно только при изменении состояния переменной * (отрицательный перепад: изменение от значения 1 до значения 0, см. примечание в 4.6). Символ является общим и распространяется на все логические высказывания либо для элементарной переменной, либо для множества из нескольких логических переменных.  ПРИМЕР: Ассоциированное условие перехода истинно только тогда, когда логическое произведение «a . b» изменяется из состояния 1 в состояние 0.
[17]		**Времязависимое условие перехода:** Запись «t1/*/t2» указывает, что условие перехода истинно только по прошествии времени t1 от момента возникновения положительного перепада (↑*) ограниченной по времени переменной и снова становится ложным по прошествии времени t2 от возникновения отрицательного перепада (↓*). Звездочку заменяют значением переменной, задающей задержку по времени, например переменной этапа или входной переменной. Времена t1 и t2 заменяют их числовыми значениями, выраженными в соответствующих единицах времени. Переменная, задающая задержку по времени, должна оставаться истинной в течение периода, равного или большего времени t1, чтобы условие перехода было истинным. Примечание — Элемент задержки в этой записи определен в соответствии с IEC 60617-S01655 (2004-09).  ПРИМЕР: Условие перехода истинно через 3 с после изменения из состояния 0 в состояние 1 и становится ложным через 7 с после изменения из состояния 1 в состояние 0.

Продолжение таблицы 4

Номер	Символ	Описание
[18]		**Обычное упрощение символа 17:** Если при текущей работе задержка переменной этапа на время t2 равна нулю, то условие перехода становится ложным при деактивации этапа *, который активировал задержку. Звездочку заменяют меткой этапа, который должен быть задержан по времени. Задержанный по времени этап должен оставаться активным в течение времени, равного или большего, чем t1 для условия перехода, чтобы быть истинным. Данная запись может использоваться, когда задерживаемый по времени этап не является предшествующим этапом перехода. ПРИМЕР: Условие перехода будет истинным в течение 4 с после активации этапа 27 и будет ложным при сбросе перехода, деактивирующего предшествующий этап.  В этом случае продолжительность активности этапа 27 составляет 4 с.
[19]		**Логическое значение предиката:** Символ «[*]» означает, что логическое значение предиката является переменной условия перехода. Поэтому, когда утверждение * подтверждено, предикат принимает значение 1, в противном случае предикат принимает значение 0. Звездочку заменяют утверждением, которое должно быть проверено. Логическая переменная предиката может быть ассоциирована с другими логическими переменными, чтобы образовать логическое высказывание об условии перехода.  ПРИМЕР 1: Условие перехода «истинно», когда утверждение «C1 = 3» подтверждено.  Пример 1а: Условие перехода «истинно», когда текущее значение счетчика C1 равно 3. Примечание — Форма утверждения не принципиальна; например, может быть использован литературный язык.  ПРИМЕР 2: Условие перехода «истинно», когда утверждение «t > 8 °C» подтверждено и когда логическая переменная k принимает значение «1», другими словами, когда температура t превышает значение 8 °C и при этом достигается высокий уровень k.

Окончание таблицы 4

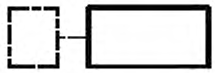
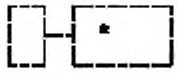
Номер	Символ	Описание
[19]		 ПРИМЕР 3: Условие перехода «истинно», когда логическая переменная «b» принимает значение 1 или когда подтверждено утверждение «R1 ≠ 24», другими словами, когда деталь занимает положение b или когда регистр R1 еще не достиг значения 24.

Символы, отображающие операцию, представлены и описаны ниже, в таблицах 5 и 6. Операции могут быть типа непрерывных действий (см. таблицу 5) или действий типа хранения (см. таблицу 6).

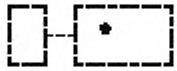
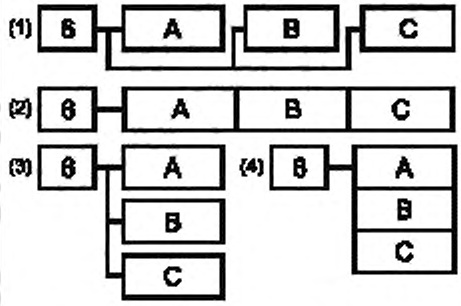
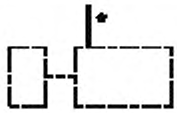
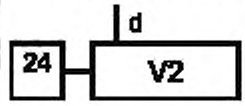
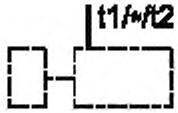
Операция хранения имеет метку (символ 26), расположенную в прямоугольнике, которая описывает, каким образом выходная переменная локализуется в заданное значение в соответствии с правилом локализации (см. 4.8.3).

В спецификации события, связанной с операцией хранения, необходимо указать, когда происходит локализация соответствующего выхода (см. правило локализации 4.8.3). Четыре способа описания (символы от 27 до 29) позволяют легко специфицировать разные множества внутренних событий, ассоциированных с операциями хранения.

Таблица 5 — Непрерывные операции

Номер	Символ	Описание
[20]		**Непрерывная операция:** Непрерывная операция обязательно ассоциирована с этапом. Несколько операций могут быть ассоциированы с одним этапом. Отношение высота-ширина прямоугольника произвольно, хотя рекомендуется прямоугольник той же высоты, что и у этапа. В отсутствие явного символа операции хранения (символы от 27 до 29) прямоугольник, ассоциированный с этапом, всегда обозначает непрерывную операцию.
[21]		**Метка присваивания выхода:** Каждая операции должна содержать внутри прямоугольника метку, относящуюся к этой операции. Метка непрерывной операции обозначает выходную переменную, которой присваивается значение «истинно» в соответствии с правилом присваивания (см. 4.8.2). Звездочку заменяют описанием выходной переменной. Текстовое выражение метки может принимать повелительную или указательную форму, но неотъемлемой частью его является указание выхода. Порядок представления операций не означает какую-либо последовательность операций. ПРИМЕР 1: Различные формы — буквенная или символьная — метки операции, относящейся к выходу, когда при значении «истинно» будет инициироваться открытие клапана 2. 

Продолжение таблицы 5

Номер	Символ	Описание
[21]		ПРИМЕР 2: Различные представления (1, 2, 3, 4) ассоциации нескольких операций с одним этапом.  Примечание — Эти четыре представления строго эквивалентны. Представления (2) и (4) можно рассматривать как упрощения представлений (1) и (3) соответственно.
[22]		**Условие присваивания:** Логическое высказывание, именуемое условием присваивания, которое может быть «истинным» или «ложным», влияет на любую непрерывную операцию. Отсутствие обозначения указывает, что условие всегда «истинно». Звездочку заменяют описанием условия присваивания в текстовом формате или в виде логического выражения между входными переменными и/или внутренними переменными. Это условие присваивания никогда не включает в себя перепад переменной (см. символы 15 и 16), потому что непрерывная операция, естественно, не имеет памяти, и присваивание по событию лишено смысла (см. 4.8.3). ПРИМЕР 1: Выходу V2 присваивается «истинное» значение, когда этап 24 активен и когда условие присваивания d «истинно». В противном случае выходу V2 присваивается «ложное» значение. Другими словами (в форме логического уравнения): V2 = X24d.  Примечание — X24 представляет собой переменную этапа, которая отражает активность этапа 24. ПРИМЕР 2: Выходу V2 присваивается «истинное» значение, когда этап 24 является активным (условие присваивания всегда «истинно»). В противном случае выходу V2 присваивается «ложное» значение. Другими словами (в форме логического уравнения): V2 = X24. 
[23]	 	**Времязависимое условие присваивания:** Запись «t1/*/t2» указывает, что условие присваивания становится «истинным» только по прошествии времени t1 после возникновения положительного перепада (↑*, см. символ 15) переменной времени * и становится «ложным» снова по прошествии времени t2 после возникновения отрицательного перепада (↓*, см. символ 16). Звездочку заменяют переменной времени, например переменной этапа или входной переменной.

Окончание таблицы 5

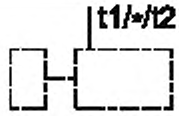
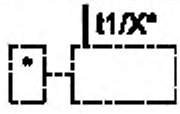
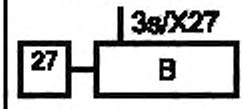
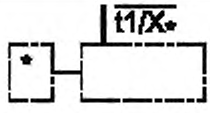
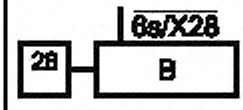
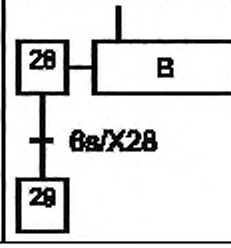
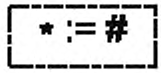
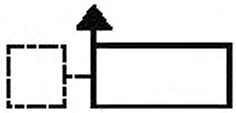
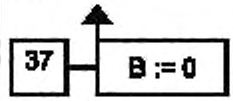
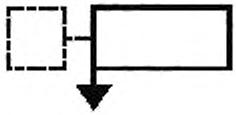
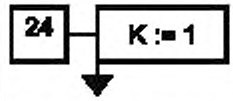
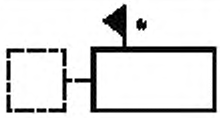
Номер	Символ	Описание
[23]		Символы t1 и t2 заменяют их фактическими значениями, выраженными в выбранных единицах времени. Ограниченная переменная должна сохранять «истинное» значение в течение времени, равного или большего, чем t1, для того чтобы условие присваивания было «истинным». Примечание — Данная запись представляет собой элемент задержки, определенный в IEC 60617-S01655 (2004-09). ПРИМЕР: Условие присваивания «истинно» только спустя 3 с после перехода «а» из состояния «0» в состояние «1» и «ложно» спустя 7 с после перехода «а» из состояния «1» в состояние «0».  Значение выхода B зависит от активности этапа 27 и от значения условия присваивания (см. правила присваивания 4.8.2).
[24]		**Отложенная операция:** Отложенная операция представляет собой непрерывную операцию, в которых условие присваивания становится «истинным» только по прошествии указанного времени t1 после активации, ассоциированной с этапом *, используется с целью задержки присваивания соответствующему выходу значения «истинно». ПРИМЕР: Выходу B присваивается «истинное» значение по прошествии 3 с момента активации этапа 27.  Примечание — Если время активности этапа 27 менее 3 с, то выходной переменной B «истинное» значение не присваивается.
[25]		**Ограниченная по времени операция:** Ограниченная по времени операция представляет собой непрерывную операцию, в которой условие присваивания становится «истинным» только по прошествии указанного времени t1 после активации, ассоциированной с этапом *, используется для ограничения продолжительности присваивания «истинного» значения соответствующему выходу.  Примечание — Если время активности этапа 28 меньше 6 с, то выходной переменной B присваивается «истинное» значение только в период активности этапа 28. **Эквивалентное отображение:** В ассоциированном условии перехода для последующего этапа, чтобы ограничить время локализации «истинного» значения на соответствующем выходе (см. символ 18), может быть использован упрощенный оператор задержки.  ПРИМЕР 2: Эквивалентное отображение примера 1 с символом 18. Выходу B присваивается «истинное» значение только в течение 6 с после активации этапа 28.

Таблица 6 — Операции хранения

Номер	Символ	Описание
[26]		**Локализация значения # к переменной *:** В случае операции хранения выражение указывает на установку переменной * в значение #, когда происходит одно из событий, ассоциированных с операцией (см. правило локализации 4.8.3). Операция хранения, поддерживающая эту локализацию, должна быть ассоциирована со спецификацией внутренних событий (символы от 27 до 29). Локализация может быть описана текстом внутри прямоугольника операции. ПРИМЕР 1: Установить значение логической переменной A в состояние «истинно». Высказывание: «A:= 1» описывает локализацию значения 1 логической переменной A, когда происходит одно из событий, ассоциированных с операцией. ПРИМЕР 2: Установить значение логической переменной b в состояние «ложно». Высказывание: «b:= 0» описывает локализацию значения 0 логической переменной b, когда происходит одно из событий, ассоциированных с операцией. ПРИМЕР 3: Приращение счетчика. Высказывание: «C:= C+1» описывает локализацию значения C+1 числовой переменной C, когда происходит одно из событий, ассоциированных с операцией.
[27]		**Операция на активации:** Операция на активации является операцией хранения, ассоциированной с множеством внутренних событий, каждое из которых имеет своим следствием активацию связанного с ним этапа. Традиционно данная операция отображается стрелкой, символизирующей активацию этапа, на левой стороне замкнутого прямоугольника.  **ПРИМЕР: Логической переменной B присваивается** значение 0, когда происходит одно из событий, приводящих к активации этапа 37.
[28]		**Операция на деактивации:** Операция на деактивации является операцией хранения, ассоциированной с множеством внутренних событий, каждое из которых имеет своим следствием деактивацию связанного с ним этапа. Традиционно данная операция отображается стрелкой, символизирующей деактивацию этапа, на левой стороне замкнутого прямоугольника.  **ПРИМЕР: Логической переменной K присваивается** значение 1, когда происходит одно из событий, отображаемых при деактивации этапа 24.
[29]		**Операция на событии:** Операция на событии относится к операции хранения, связанной с каждым из внутренних событий, описанных выражением * при условии, что этап, с которым связана операция, является активным. Традиционно данная операция отображается стрелкой на верхней стороне замкнутого прямоугольника, снабженной символом, показывающим, что действие обусловлено возникновением одного из внутренних событий, указанных в выражении *. Рекомендуется, чтобы логическое выражение *, описывающее множество внутренних событий, содержало один или более перепадов входной переменной.

Окончание таблицы 6

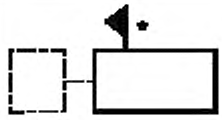
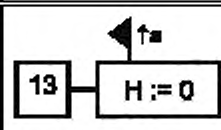
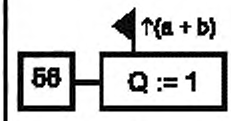
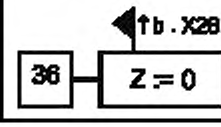
Номер	Символ	Описание
[29]		 ПРИМЕР 1: Логической переменной H присваивается значение 0, когда происходит одно из событий, представленных как «↑a», и одновременно этап 13 оказывается активным. Примечание — Комбинация между множеством входных событий, представленных выражением «*a», и активностью этапа 13 отображает фактически множество внутренних событий (см. определение 3.6).  ПРИМЕР 2: Логической переменной Q присваивается значение 1, когда происходит одно из событий, представленных как «↑(a + b)», и одновременно этап 56 оказывается активным.  ПРИМЕР 3: Логической переменной Z присваивается значение 0, когда происходит одно из событий, представленных как «↑b», и одновременно этапы 36 и 28 оказываются активными.

Таблица 7 — Комментарии, связанные с элементами диаграммы grafcet

Номер	Символ	Описание
[30]	«*»	**Комментарий:** Комментарий относительно графических элементов диаграммы заключают в кавычки. Звездочку заменяют соответствующим комментарием. ПРИМЕР 1: Комментарий «этап ожидания», относящийся к этапу 45. ПРИМЕР 2: Комментарий «пробой отверстия», относящийся к операции, ассоциированной с этапом 28. ПРИМЕР 3: Комментарий «синхронизация», относящийся к переходу.

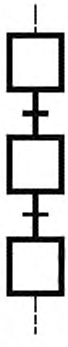
6 Графическое представление последовательных структур

6.1 Общие положения

Конструктор может строить диаграммы grafcet, используя различные характерные структуры, при условии строгого соблюдения правила синтаксиса при чередовании этап/переход.

6.2 Основные структуры

6.2.1 Последовательность



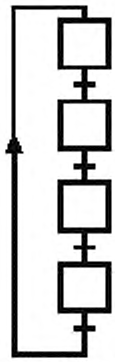
Последовательность представляет собой порядок следования этапов, например:

- каждый этап, за исключением последнего, имеет только один последующий переход;
- каждый шаг, за исключением первого, имеет только один предыдущий переход, допускаемый на одном этапе последовательности.

Примечание 1 — Последовательность называют «активной», если по крайней мере один из ее этапов является активным. Последовательность называют «неактивной», когда ни один из ее этапов не является активным.

Примечание 2 — Последовательность может включать в себя произвольное количество этапов.

6.2.2 Цикл одинарной последовательности



Вариант петлей последовательности характеризуется следующим:

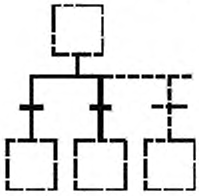
- каждый этап имеет только один последующий переход;
- каждый этап имеет только один предшествующий переход, допускаемый на одном этапе последовательности.

Примечание 1 — Цикл одинарной последовательности может представлять собой частичную диаграмму grafcet (см. 7.2.2).

Примечание 2 — Цикл одинарной последовательности должен удовлетворять по крайней мере одному из следующих условий, чтобы была разрешена активация его этапов:

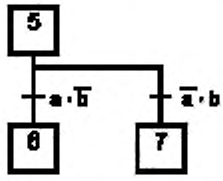
- иметь по крайней мере один начальный этап;
- быть подчиненным в принудительном порядке частичной диаграмме grafcet более высокого уровня (см. 7.3);
- принадлежать к одному из вложений охватывающего этапа (см. 7.4).

6.2.3 Выбор последовательностей

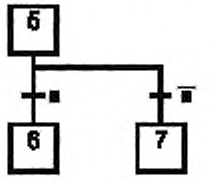


Выбор последовательностей заключается в выборе эволюции между несколькими последовательностями, начиная с одного или нескольких этапов. Эта структура отображается как множеством одновременно разрешенных переходов, так и возможных эволюций.

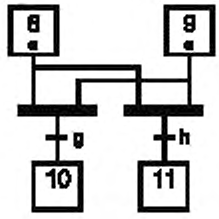
Примечание — Эксклюзивная активация выбранной последовательности не гарантируется структурой. Конструктор должен убедиться, что временные, логические или механические аспекты переходного условиях являются взаимно исключающими.



ПРИМЕР 1: Исключение между последовательностями достигается путем логического исключения двух датчиков. Если «a» и «b» одновременно «истинны», когда этап 5 активен, то никакой переход не может быть сброшен.

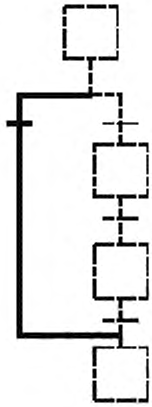


ПРИМЕР 2: Ряд приоритетов. В данном примере приоритет отдается переходу 5/6, который сбрасывается когда «a» «истинно».



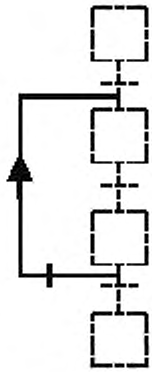
ПРИМЕР 3: Выбор последовательностей следует синхронизации двух предшествующих последовательностей. Выбор из последующих последовательностей с помощью g и h возможен только тогда, когда оба перехода сбрасываются путем одновременной активации этапов 8 и 9 (см. символ 9).

6.2.4 Пропуск этапа



Это частный случай выбора последовательностей, позволяющий исключить всю последовательность, или один, или несколько этапов последовательности, если, например, операции, связанные с этими этапами, оказываются ненужными.

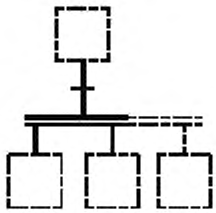
6.2.5 Пропуск обратной последовательности



Это частный случай выбора последовательностей, при котором разрешено повторение процесса исключения последовательности до тех пор, пока, например, не будет выполнено заданное условие.

Примечание — Для удобства графического отображения переходы могут быть размещены на горизонтальных линиях связи (см. примечание 3, символ 7).

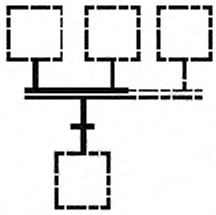
6.2.6 Активация параллельных последовательностей



В данной структуре, чтобы указать одновременную активность нескольких последовательностей от одного или нескольких этапов, используют символ синхронизации 9.

Примечание — После их одновременной активации эволюция активных шагов в каждой из параллельных последовательностей становится таким образом независимой.

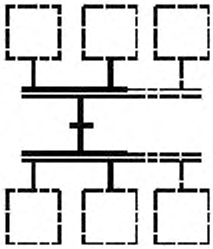
6.2.7 Синхронизация последовательностей



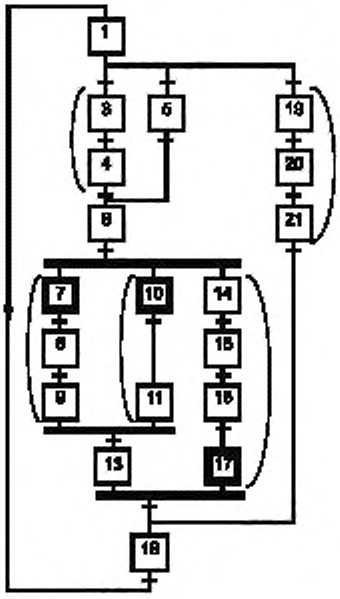
В данной структуре, чтобы указать на задержку между окончанием предшествующей последовательности и активацией последующей последовательности, используют символ синхронизации 9.

Примечание — Переход разрешен только тогда, когда все предшествующие этапы активны.

6.2.8 Синхронизация и активация параллельных последовательностей



В данной структуре, чтобы указать на задержку между окончанием предшествующей последовательности и активацией последующей последовательности, используют символ синхронизации 9.



ПРИМЕР: Диаграмма grafcet, в которой можно выделить следующие основные структуры:

- последовательности (некоторые из них отмечены скобками);
- выбор последовательностей (от этапа 1 до этапов 3, 5 и 19);
- активация параллельных последовательностей (по направлению вниз от этапа 6);
- две синхронизации последовательностей (от этапов 9 и 11 к этапу 13 и от этапов 13 и 17 к этапу 18).

Примечание 1 — Данный пример иллюстрирует только структуру диаграммы grafcet, без описания ее интерпретации.

Примечание 2 — Данная диаграмма grafcet не является типичным примером, поскольку она не обязательно должна содержать контур обратной связи.

6.3 Конкретные структуры

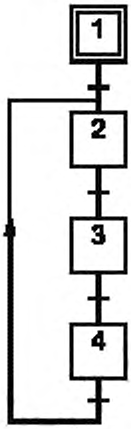
6.3.1 Запуск последовательности от исходного этапа



Исходным является этап, который не имеет никакого предшествующего перехода.

Примечание 1 — Для того чтобы разрешить активацию исходного этапа, должно быть выполнено хотя бы одно из следующих условий:

- исходный этап является отправным;
- исходный этап подвергается действию принудительной команды от частичной диаграммы grafcet более высокого уровня (см. 7.3);
- исходный этап является одним из активированных этапов вложения (см. 7.4).



ПРИМЕР 1: Отправной исходный этап:

Отправной исходный этап 1 является активным только в момент инициализации, этапы 2, 3 и 4 образуют цикл из одинарной последовательности.

Примечание 2 — Представлена только структура диаграммы grafcet без описания ее интерпретации.

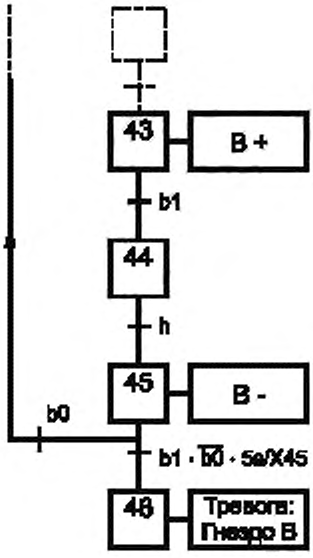
6.3.2 Завершение последовательности с помощью пит-этапа



Пит-этапом является этап, который не имеет никакого последующего перехода.

Примечание 1 — Деактивация пит-этапа возможна только одним из двух следующих способов:
- принудительная команда от частичной диаграммы grafcet более высокого уровня (см. 7.3);
- деактивация охватывающего этапа, если пит-этап является его вложением (см. 7.4).

Примечание 2 — Этап может быть исходным этапом и пит-этапом одновременно, тогда он образует одноэтапную последовательность, используемую для представления комбинаторного поведения.



ПРИМЕР: Пит-этап:

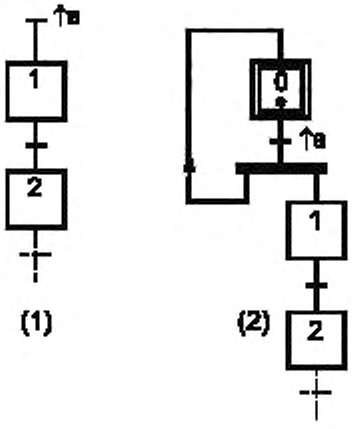
Пит-этап 46 активируется только при выполнении логического условия «b1 · b0» через 5 секунд после активации этапа 45 (см. символ 18). Выходу «Тревога: Гнездо B» присваивается истинное значение.

6.3.3 Запуск последовательности с исходным переходом



Исходный переход — это такой переход, который не имеет никаких предшествующих этапов. По умолчанию исходный переход всегда разрешен, и он сбрасывается, как только условие перехода * становится «истинным».

Примечание 1 — Активация последующего этапа исходного перехода является эффективной, пока остается «истинным» его условие перехода, независимо от условий переходов для переходов, разрешенных на этом этапе (см. правило эволюции № 5, 4.5.5). Чтобы избежать постоянной активации последующего этапа исходного перехода, рекомендуется, чтобы ассоциированное условие перехода становилось «истинным» только тогда, когда происходит входное событие или внутреннее событие. С этой целью логическое выражение, формирующее условие перехода, всегда должна включать в себя входной перепад.



ПРИМЕР: Исходный переход и эквивалентная структура:

Отображения (1) и (2) описывают эквивалентное поведение: этап 1 активируется каждый раз при изменении логической переменной от значения 0 к значению 1. Для поддержания исходного этапа 0 активным в отображении (1) используется исходный этап, в отображении (2) используется символ синхронизации и контур обратной связи.

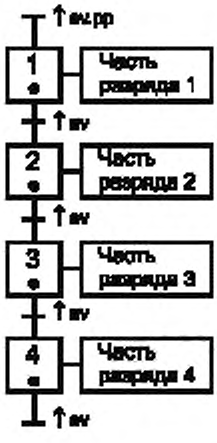
Примечание 2 — Точка в обозначении этапа 0 указывает, что данный этап всегда является активным.

6.3.4 Завершение последовательности пит-переходом



Пит-переход — это переход, не имеющий последующего этапа.

Примечание 1 — Когда пит-переход разрешен и когда ассоциированное с ним условие перехода * является «истинным», деактивация расположенных выше него этапов происходит только в результате сброса перехода.



ПРИМЕР: Структура сдвигового регистра:

Структура сдвигового регистра представляет собой удачный пример использования исходного перехода и пит-перехода. В этом примере каждый активный этап указывает на присутствие звена соответствующего разряда. Наличие звена (pp) на входе и продвижение передачи между разрядами (↑ av) активирует этап 1 путем сброса исходного перехода. При каждом продвижении передачи (↑ av) разрешенные переходы одновременно сбрасываются, включая пит-переход снизу от этапа 4.

Примечание 2 — Отображение соответствует частному случаю, когда все этапы одновременно активны.

7 Структурирование

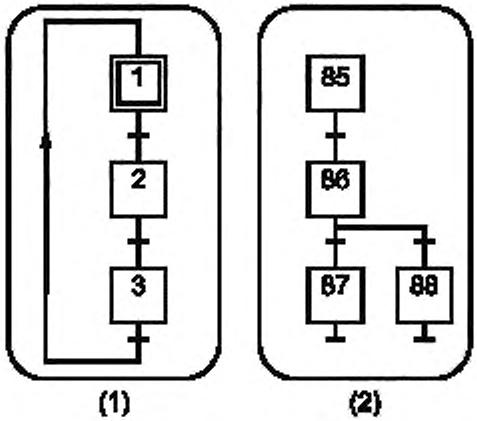
7.1 Общие положения

Сложность автоматизированных систем требует соответствующих методов структурирования спецификации. Это структурирование в зависимости от соответствующей методологии может ограничиться простым делением спецификации или может интегрировать иерархические понятия подчинения или вложения.

7.2 Разбиение диаграммы grafcet

7.2.1 Связанная диаграмма grafcet

Связанная диаграмма grafcet представляет собой структуру, в которой на диаграмме grafcet неизменно присутствует преемственность связей (чередование этапов и переходов) между любыми двумя элементами, этапом или переходом.



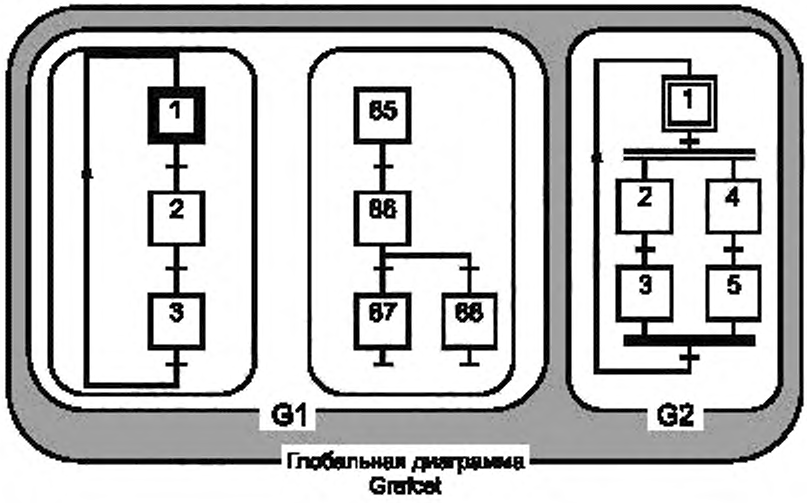
ПРИМЕР: Все элементы в зоне (1) образуют связанную диаграмму grafcet, так как ее этапы и ее переходы соединены прямыми связями. Элементы в зоне (2) также образуют связанную диаграмму grafcet.

Примечание — Несвязанная диаграмма grafcet не имеет технологического применения.

7.2.2 Парциальная диаграмма grafcet

Парциальная диаграмма grafcet (см. таблицу 8), составленная из одной или нескольких связанных диаграмм grafcet, получается в результате деления в соответствии с методическими критериями глобальной диаграммы grafcet, описывающей поведение последовательной части системы.

Парциальная диаграмма grafcet помещается в обрамление, на нижней границе которого указывается метка G* отображаемой диаграммы grafcet.



ПРИМЕР: Деление глобальной диаграммы grafcet на части:

Глобальная диаграмма grafcet составлена из парциальных диаграмм grafcet G1 и G2. Парциальная диаграмма grafcet G1 состоит из двух связанных диаграмм grafcet.

Таблица 8 — Парциальная диаграмма grafcet

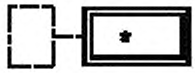
Номер	Символ	Описание
[31]	G*	Имя парциальной диаграммы grafcet: По умолчанию на парциальную диаграмму grafcet указывает буква G. Звездочку заменяют именем парциальной диаграммы grafcet.
[32]	XG*	Парциальная переменная grafcet: Парциальную диаграмму grafcet считают активной, когда по крайней мере один из ее этапов является активным, и наоборот, ее считают неактивной, когда ни один из ее этапов не является активным. Активное или неактивное состояние парциальной диаграммы grafcet может быть представлено логическими значениями «1» или «0» соответственно, переменной XG*, где звездочку * заменяют именем рассматриваемой парциальной диаграммы grafcet. ПРИМЕР: XG1 обозначает переменную парциальной диаграммы grafcet 1.
[33]	G#{..., ...}	Состояние парциальной диаграммы grafcet: Состояние парциальной диаграммы grafcet отображают набором активных этапов в рассматриваемый момент. Состояние парциальной диаграммы grafcet # задают в виде G#{..., ...}, где в фигурных скобках перечислены активные этапы, характеризующие ситуацию парциальной диаграммы grafcet в рассматриваемый момент. Значок # должен быть заменен именем парциальной диаграммы grafcet. ПРИМЕР: G12{8, 9, 11} указывает на состояние парциальной диаграммы grafcet 12 в рассматриваемый момент, в котором только шаги 8, 9 и 11 являются активными.
[34]	G#{*}	Текущее состояние парциальной диаграммы grafcet: Звездочка указывает по умолчанию состояние парциальной диаграммы grafcet # в рассматриваемый момент. Значок # заменяют именем парциальной диаграммы grafcet.
[35]	G#{ }	Пустое состояние парциальной диаграммы grafcet: Оно обозначает состояние парциальной диаграммы grafcet #, когда ни один из ее этапов не является активным. Значок # заменяют именем парциальной диаграммы grafcet.
[36]	G#{INIT}	Начальное состояние парциальной диаграммы grafcet: Оно обозначает состояние парциальной диаграммы grafcet # в начальный момент. Значок # заменяют именем парциальной диаграммы grafcet.

7.3 Структурирование с использованием подчинения парциальной диаграммы grafcet

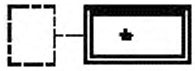
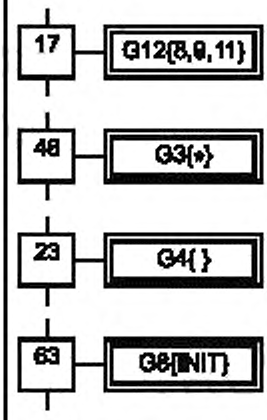
Это метод структурирования спецификации последовательной части системы, использующей принудительные команды (см. таблицу 9). Команды позволяют наложить заданное состояние на данную парциальную диаграмму grafcet, исходя из состояния другой части (см. пример в приложении В). Парциальная диаграмма grafcet может быть подчинена различным предшествующим этапам и командам.

Примечание — «Команды сходны с непрерывными операциями. В то время как операция воздействует на переменную, команда воздействует на парциальную диаграмму grafcet».

Таблица 9 — Подчинение парциальной диаграммы grafcet

Номер	Символ	Описание
[37]		Принудительная команда для парциальной диаграммы grafcet: Звездочку заменяют символом состояния парциальной диаграммы grafcet (символы от 33 до 36). Ассоциированная с активностью этапа парциальной диаграммы grafcet иерархически более высокого уровня, принудительная команда является внутренней командой, позволяющей наложить состояние на иерархически более низкую парциальную диаграмму grafcet. Принудительную команду отображают в виде двойного прямоугольника, связанного с этапом, чтобы отличить ее от операции. Принудительная команда является внутренней командой, имеющей высший приоритет выполнения по отношению к правилам эволюции.

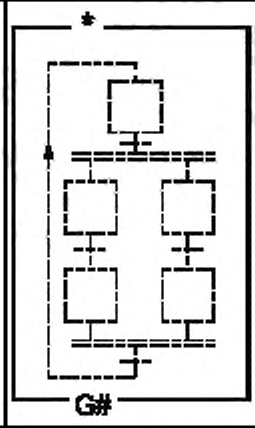
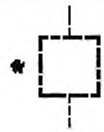
Окончание таблицы 9

Номер	Символ	Описание
[37]		Принуждаемая диаграмма grafcet не может развиваться в период поступления принудительной команды. Говорят, что диаграмма grafcet замораживается. Использование принудительной команды в спецификации требует иерархической структуры парциальных диаграмм grafcet, построенной таким образом, чтобы каждая принуждающая парциальная диаграмма grafcet находилась на более высоком уровне, чем парциальная принуждаемая диаграмма grafcet.  ПРИМЕР 1: Принуждение парциальной диаграммы grafcet к явному состоянию. Когда этап 17 активен, парциальная диаграмма grafcet 12 принудительно переходит в состояние, характеризующееся активностью этапов 8, 9 и 11. ПРИМЕР 2: Принуждение парциальной диаграммы grafcet к текущему состоянию. Когда этап 48 активен, парциальная диаграмма grafcet 3 принудительно переходит в то состояние, в котором она оказывается во время воздействия. Примечание 1 — Эту команду называют также «замораживающей командой». ПРИМЕР 3: Принуждение парциальной диаграммы grafcet к пустому состоянию. Когда этап 23 активен, парциальная диаграмма grafcet 4 принудительно переходит в пустое состояние (см. символ 35). Примечание 2 — В этом случае ни один этап G4 не является активным.

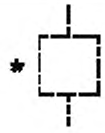
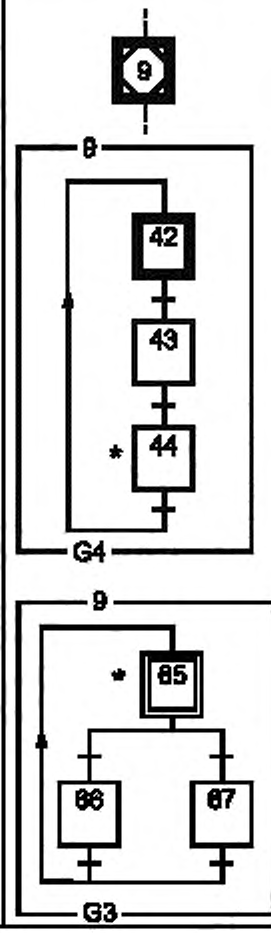
7.4 Структурирование с использованием вложения

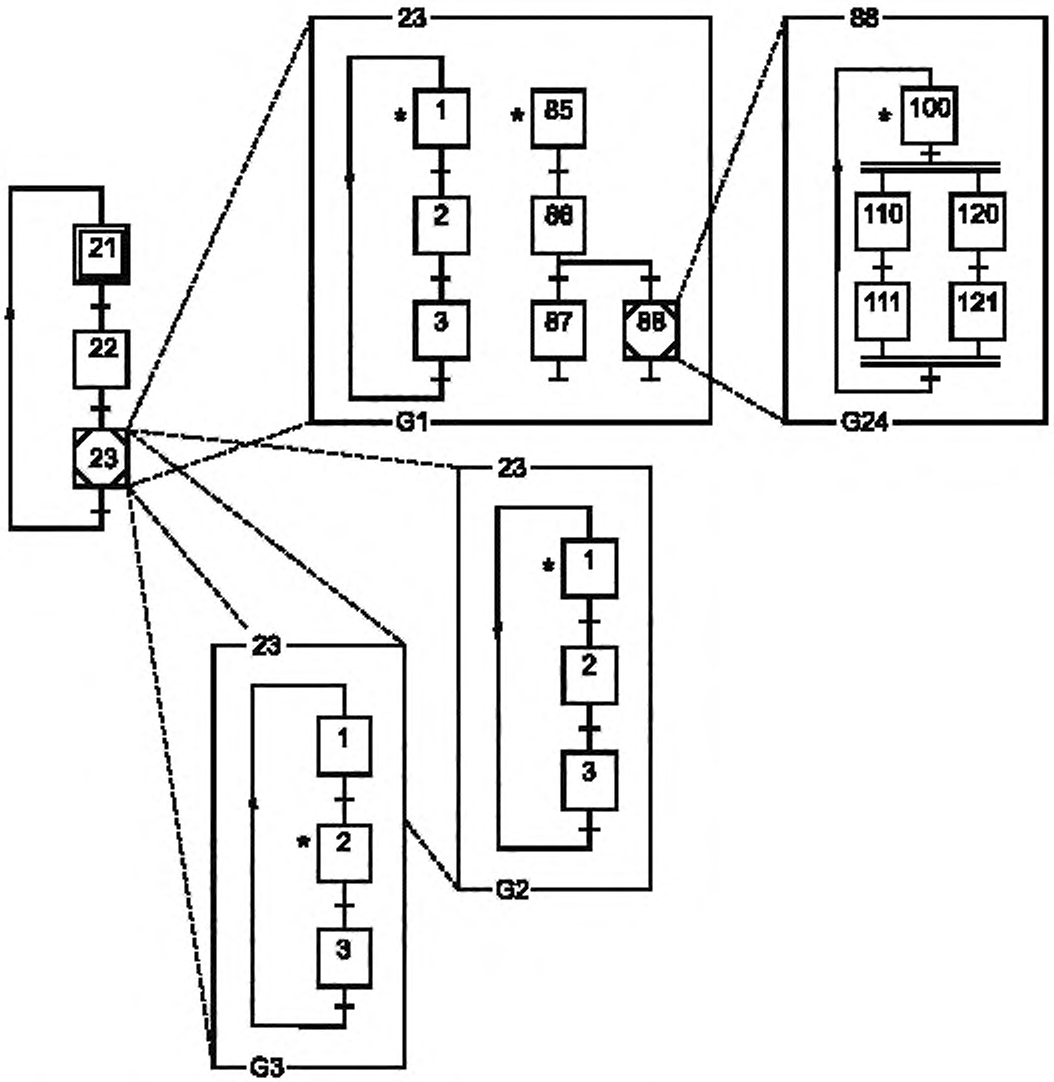
Множество этапов (парциальной диаграммы grafcet) называют вложенными в этап, рассматриваемый в качестве охватывающего этапа (см. таблицу 10). Тогда и только тогда, если при активном охватывающем этапе по крайней мере один из вложенных этапов становится активным, конструктор может использовать вложение в структуре диаграммы grafcet с учетом иерархии (см. пример на рисунке 3 или на рисунке В.6). Вложение может относиться только к одному охватывающему этапу.

Таблица 10 — Охватывающие этапы

Номер	Символ	Описание
[4]		Охватывающий этап (символ 4): Данное обозначение указывает, что этап содержит другие этапы, рассматриваемые как вложенные, в качестве одного или нескольких вложений того же охватывающего этапа. Охватывающий этап обладает всеми свойствами этапа, метка этапа должна заменить звездочку. Охватывающий этап может управлять одним или несколькими вложениями, имеющими по крайней мере один активный этап, когда охватывающий этап активен в каждом из них, и не имеющими ни одного активного этапа, когда охватывающий этап неактивен.
[38]		Графическое отображение вложения: Вложение # охватывающего этапа * может быть представлено парциальной диаграммой grafcet вложенных этапов, сгруппированных в рамке, на верхней стороне которой слева размещают имя * охватывающего этапа, а на нижней стороне слева размещают метку G# отображаемого вложения. Множество вложенных этапов образуют парциальную диаграмму grafcet, имя которой может быть использовано в качестве метки для соответствующего вложения.
[39]	X*/G#	Глобальное обозначение вложения: Вложение #№ охватывающего этапа * может быть глобально описано текстовым выражением, в котором охватывающий этап * обозначают переменной этапа X*, вложение обозначают символом /, а вложенные этапы — именем парциальной диаграммы grafcet G#, которой они принадлежат. Примечание — Данное отображение предполагает, что указанная парциальная диаграмма grafcet предварительно была определена.
[40]	X*/X#	Простое обозначение вложения: Текстовое выражение, означающее, что этап # вложен в охватывающий этап *, может использовать переменные этапа без использования имени самого вложения. Примечание — Данная запись пригодна для обозначения иерархического ряда этапов, вложенных друг в друга. Она также удобна для относительной идентификации этапов по уровню вложения. ПРИМЕР: X4/X25/X12 обозначает вложение этапа 12 в этап 25, который, в свою очередь, вложен в этап 4.
[5]		Начальный охватывающий этап (символ 5): Данное отображение показывает, что этот этап участвует в исходном состоянии. В этом случае по крайней мере один из этапов, вложенных в каждое вложение, также должен быть начальным этапом.
[41]		Связь активации, общее обозначение: Связь активации, обозначенная звездочкой с левой стороны символов вложенных этапов, указывает вложенные этапы, которые становятся активными при активации охватывающего этапа. Связь активации не следует путать с символом для начальных этапов, которые могут быть вложенными. Тем не менее не исключено, что начальный вложенный этап также имеет связь активации.

Окончание таблицы 10

Номер	Символ	Описание
[41]		Деактивация охватывающего этапа приводит к деактивации всех вложенных в него этапов. Подобная деактивация часто происходит при сбросе последующего перехода охватывающего этапа, но может также происходить от любых других средств деактивации (принуждение или охватывание более высокого уровня).  ПРИМЕР: Охватывающий этап 9 обязательно является начальным этапом, поскольку содержит в себе начальный этап 42. Вложение G4 охватывающего этапа 9 включает этапы 42, 43 и 44. Начальный этап 42 участвует в исходном состоянии, поэтому является активным в начальный момент. С другой стороны, как при каждой активации этапа 9, так и после эволюции grafcet этап 44 активируется. Вложение G3 охватывающего этапа 9 включает этапы 65, 66 и 67. Начальный этап 65 участвует в исходном состоянии, поэтому является активным в начальный момент. Он также активируется при каждой активации этапа 9 и в результате эволюции grafcet.



ПРИМЕР: Структурирование вложением:

Охватывающий этап 23 имеет три вложения, представленные парциальными диаграммами grafcet G1, G2 и G3. Парциальная диаграмма grafcet 24 является вложенной в этап 88 парциальной диаграммы grafcet 1. При активации охватывающего этапа 23 этапы 1 и 85 диаграммы G1 также активируются (то же самое справедливо и для других охватывающих этапов 23: G2 и G3). Таким образом, парциальные диаграммы grafcet G1, G2 и G3 выполняются параллельно.

Когда активируется охватывающий этап 88, этап 100 диаграммы G24 также активируется.

Деактивация этапа 88 деактивирует все этапы диаграммы G24.

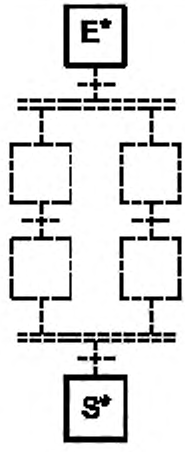
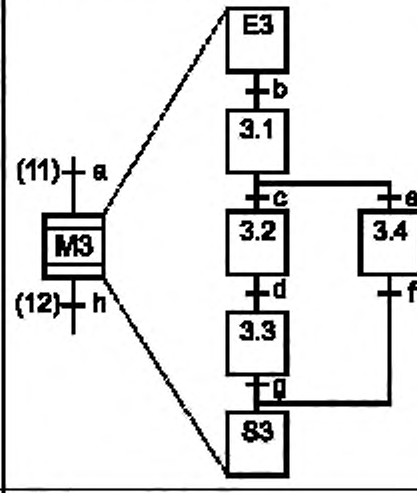
Деактивация этапа 23 деактивирует все этапы диаграмм G1, G2, G3 и, если этап 88 был активен, все этапы диаграммы G24.

Рисунок 3 — Пример диаграммы grafcet с вложениями (сопровождается описанием диаграммы)

7.5 Структурирование с использованием макроэтапов

Для улучшения понимания диаграмм grafcet спецификации могут быть представлены на нескольких уровнях типа «макроотображение», каждый уровень отображает функцию, подлежащую выполнению, безотносительно ко всем деталям, которые являются излишними на данной стадии описания. Использование макроэтапов (см. таблицу 11) позволяет последовательно описывать диаграмму от общего к частному.

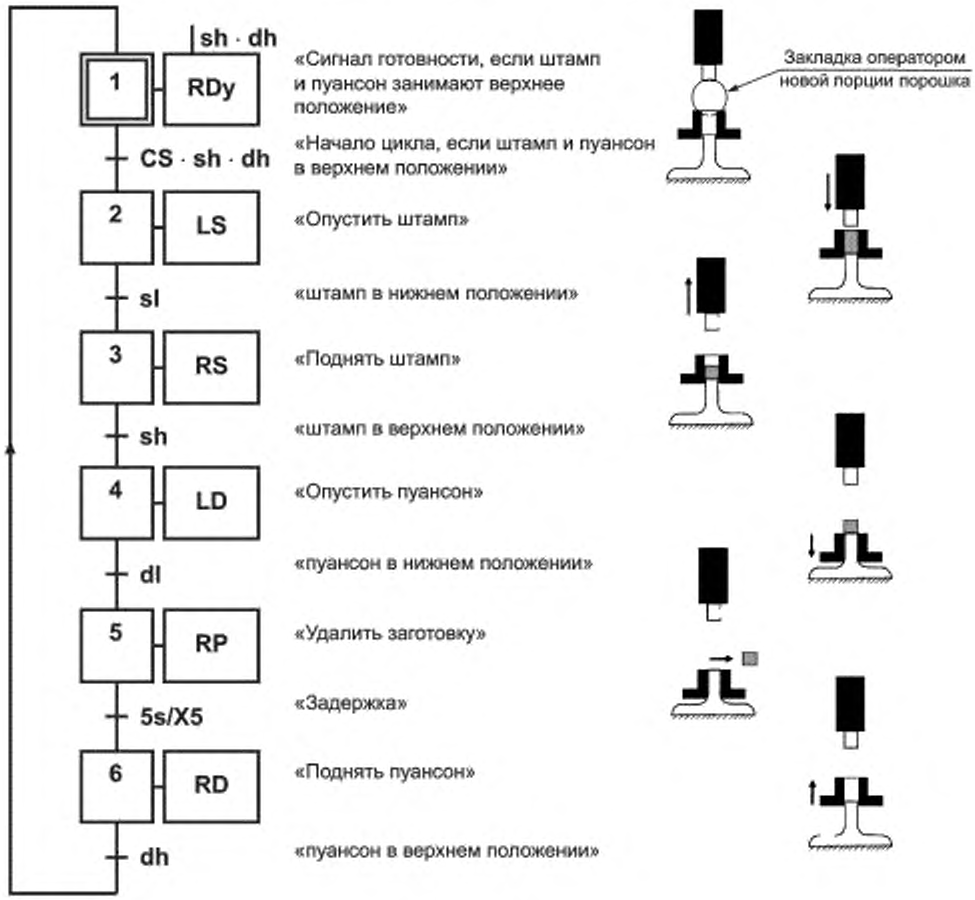
Таблица 11 — Макроэтапы

Номер	Символ	Описание
[6]		Макроэтап (напоминание символа 6): Уникальное представление детализированной диаграммы grafcet называют расширением макроэтапа. Расширение может относиться только к одному макроэтапу. Макроэтап не обладает всеми свойствами других видов этапа (символы от 1 до 5), поскольку только его выходной этап (см. символ 42) может подтверждать его последующие переходы. Звездочку заменяют меткой макроэтапа.
[42]		Расширение макроэтапа: Расширение макроэтапа M* представляет собой часть диаграммы grafcet от одного входного этапа E* и до одного выходного этапа S*. Входной этап E* становится активным, когда сбрасывается один из предшествующих переходов макроэтапа. Один или все последующие переходы макроэтапа могут быть разрешены только тогда, когда выходной этап S* активен. Примечание 1 — Расширение макроэтапа может состоять из одного или нескольких начальных этапов. Примечание 2 — Расширение макроэтапа может состоять из одного или нескольких макроэтапов.  ПРИМЕР: Макроэтап M3 представлен своим расширением: Сброс перехода 11 приводит к активации входного этапа E3 макроэтапа M3. Переход 12 разрешен только тогда, когда выходной этап S3 является активным. Сброс перехода 12 приводит к деактивации этапа S3.
[43]	XM*	Переменная макроэтапа: Макроэтап называют активным, когда по крайней мере один из его этапов является активным, и наоборот, его называют неактивным, когда ни один из его этапов не является активным. Активное или неактивное состояние макроэтапа может быть отображено логическими значениями «1» или «0» соответственно, переменной XM*, в котором звездочку * заменяют именем рассматриваемого макроэтапа.

Приложение А
(справочное)

Пример управления прессом

Пресс для формования порошковых материалов функционирует, как показано на диаграмме grafcet на рисунке А.1. Когда пресс находится в режиме ожидания на этапе 1, штамп и пуансон занимают верхнее положение и горит сигнал «готовность», чтобы указать оператору на возможность осуществления закладки новой порции порошка. Операции выполняются последовательно, как показано на диаграмме grafcet.



Обозначения:

Входы		Выходы	
CS	Старт цикла	RDy	Сигнал готовности
Sh	штамп в верхнем положении	LS	Опустить штамп
Sl	штамп в нижнем положении	RS	Поднять штамп
Dh	пуансон в верхнем положении	LD	Опустить пуансон
dl	пуансон в нижнем положении	RP	Удалить заготовку
		RD	Поднять пуансон

Рисунок А.1 — Отображение функционирования пресса с использованием диаграммы grafcet

Приложение В
(справочное)

Пример: Автоматическая система взвешивания-смешивания

В.1 Презентация примера

Продукты А и В, предварительно взвешенные на устройстве взвешивания С, а также растворимые брикеты, поступающие по одному на транспортер, подаются в смеситель N. Автоматическая система, структура которой приведена на рисунке В.1, позволяет получать смесь этих трех компонентов.

В.2 Цикл

Запускаемый нажатием кнопки «старт цикла» вызывает одновременное взвешивание продуктов и транспортировку брикетов, действуя следующим образом:

- взвешивание продукта А до отметки «а» устройством взвешивания, затем дозирование продукта В до отметки «b» с последующим опорожнением устройства взвешивания С в смеситель;
- транспортировка двух брикетов.

Цикл завершается вращением миксера и его финальным опрокидыванием по истечении времени «t1». Вращение миксера продолжается в течение опорожнения.

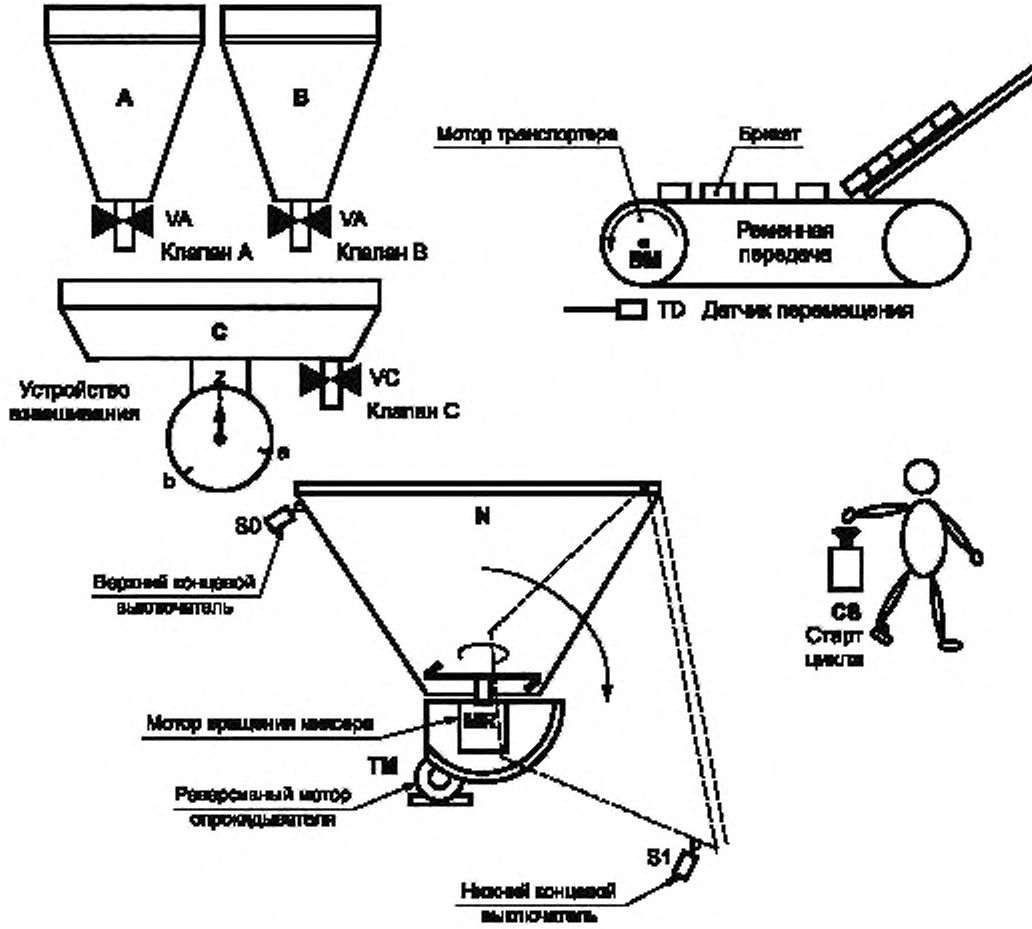
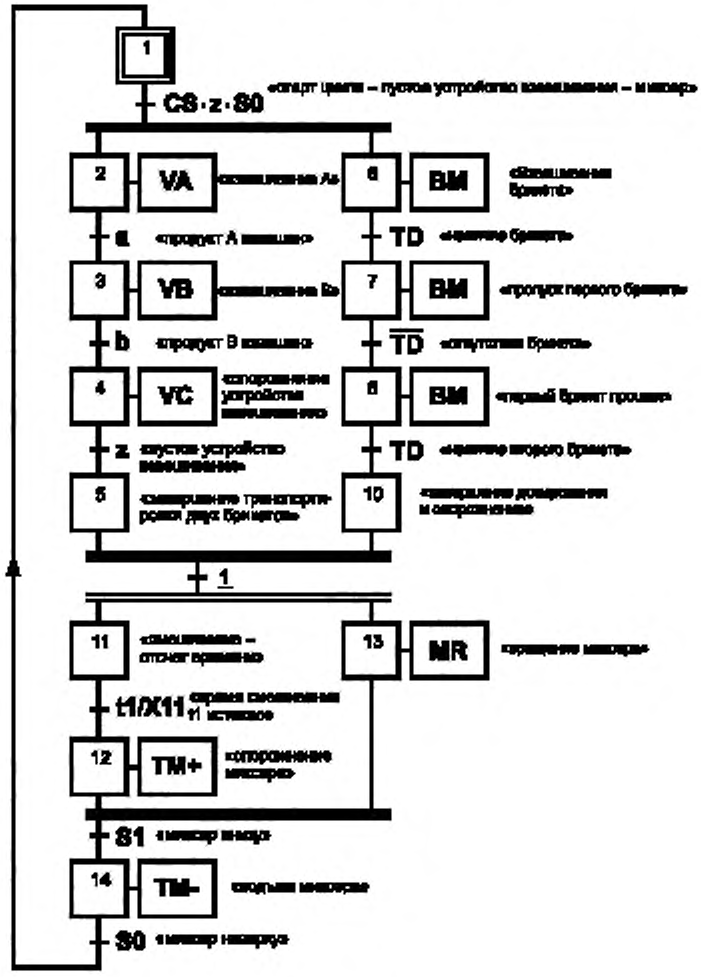


Рисунок В.1 — Структурная схема системы взвешивания-смешивания

В.3 Описание действия управляющих команд устройства взвешивания-смешивания

Логически действие управляющих команд устройства взвешивания-смешивания может быть описано с помощью любой из диаграмм grafcet, приведенных на рисунках В.2—В.4.



Обозначения:

Входы		Выходы	
CS	Старт цикла	BM	Мотор транспортера
TD	Детектор перемещения	MR	Мотор вращения миксера
a	Вес жидкости А достигнут	TM+	Мотор опрокидывания (внизу)
b	Вес жидкости А + В достигнут	TM-	Мотор опрокидывания (наверху)
z	Пустое устройство взвешивания	VA	Открывание клапана А
S0	Миксер наверху	VB	Открывание клапана В
S1	Миксер внизу	VC	Открывание клапана С

Рисунок В.2 — Диаграмма grafcet устройства взвешивания-смешивания, включающая только непрерывные операции

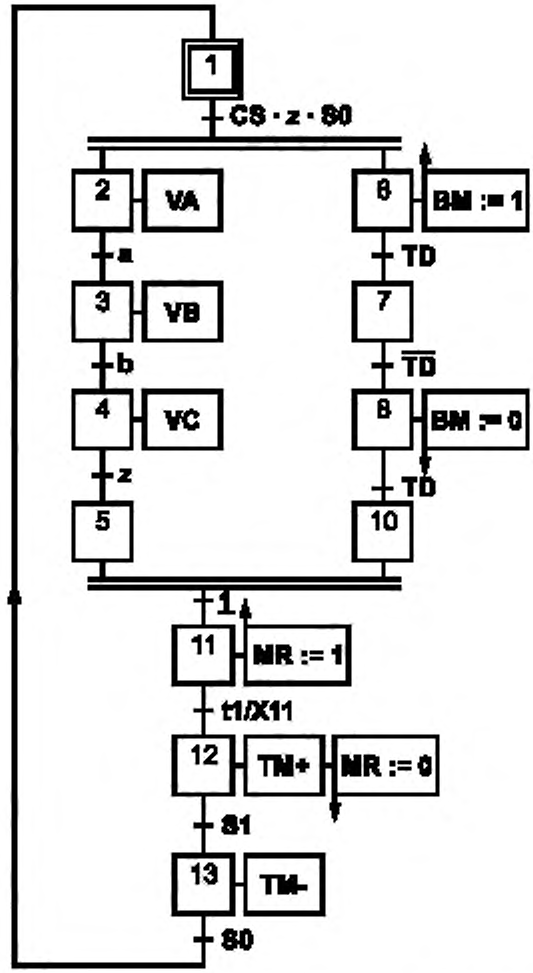


Рисунок B.3 — Диаграмма grafcet устройства взвешивания-смешивания, включающая непрерывные операции и операции хранения

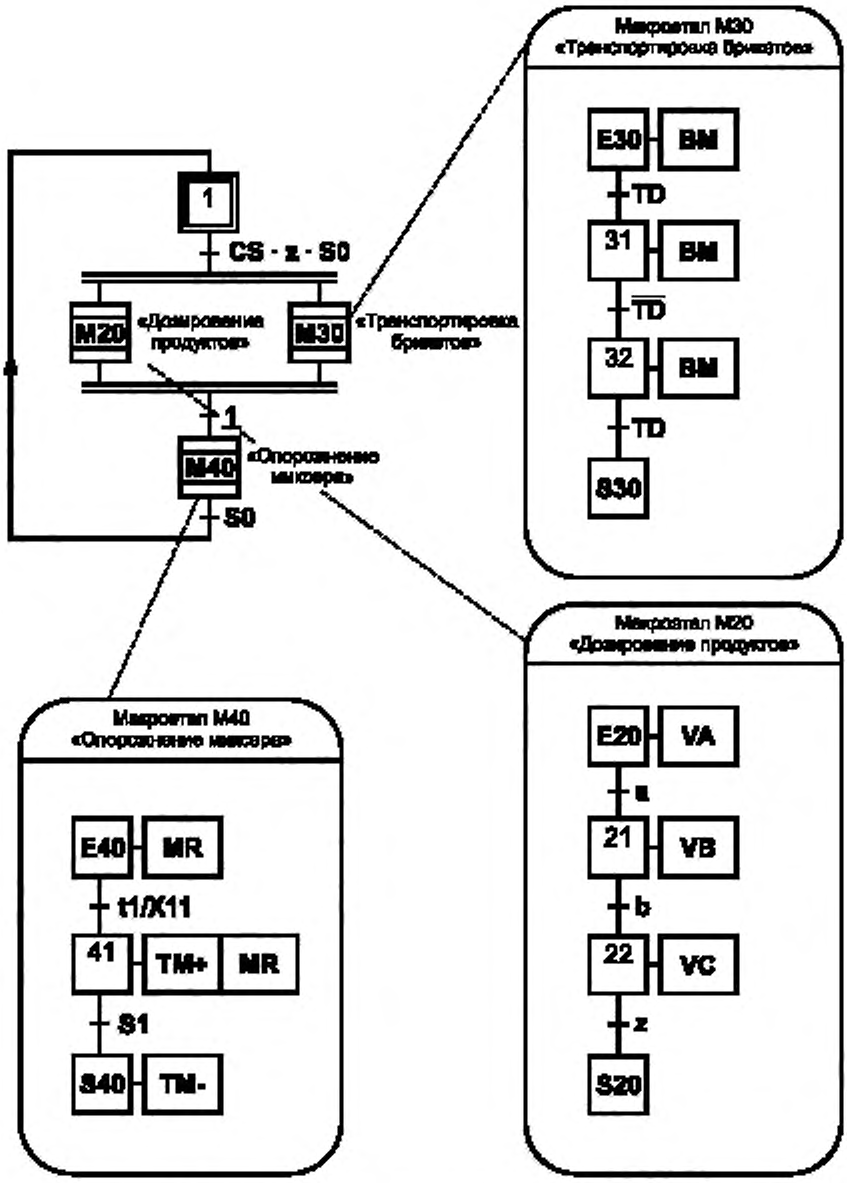
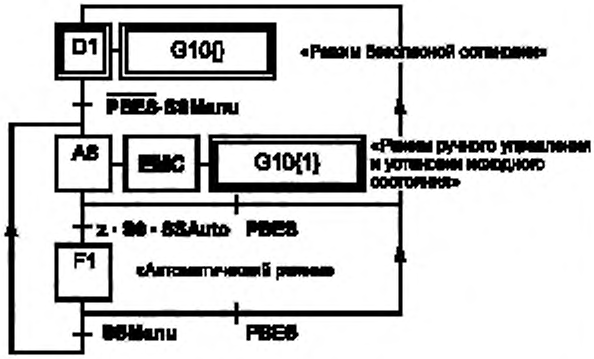


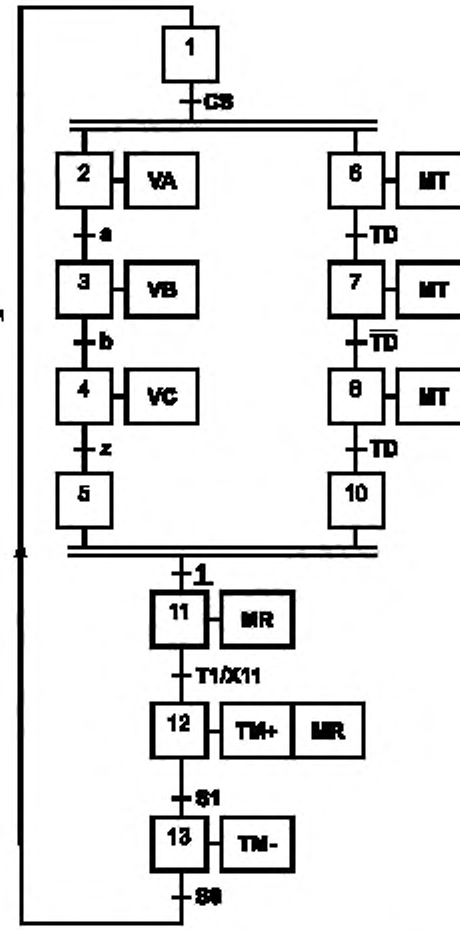
Рисунок B.4 — Диаграмма grafcet устройства взвешивания-смешивания с использованием глобального описания на основе макроэтапов и детального описания с помощью расширений макроэтапа

B.4 Устройство взвешивания-смешивания: структурирование в соответствии с режимами работы

Основываясь на анализе режимов работы автоматического устройства взвешивания-смешивания, можно выполнить иерархическое структурирование спецификации путем использования принудительных команд (см. рисунок B.5) или охватывающих этапов (см. рисунок B.6). Дополнительные входы и выходы, приведенные в таблице к рисунку B.5, привлечены вследствие необходимого учета команд, поступающих от оператора.



G1: Парциальная диаграмма grafcet режимов эксплуатации (верхний иерархический уровень)



G10: Парциальная диаграмма grafcet автоматического цикла (нижний иерархический уровень)

Обозначения:

Входы Выходы

PBES	Нажать кнопку аварийного останова	EMC	Разрешено ручное управление
SSAuto	Селекторный переключатель на автоматический режим		
SSManu	Селекторный переключатель на ручной режим		

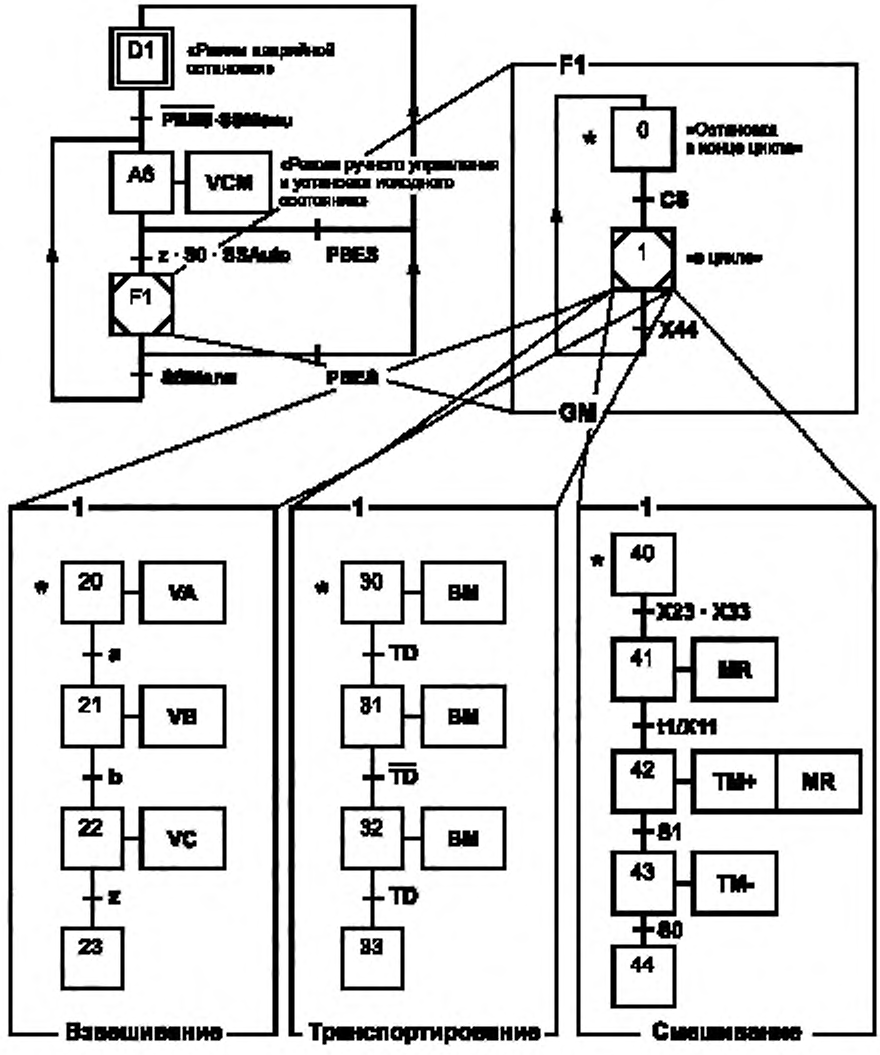
Принудительная иерархия содержит два уровня.

Принудительная команда, ассоциированная с этапом D1 парциальной диаграммы grafcet 1, принудительно переводит парциальную диаграмму grafcet 10 в пустое состояние (ни один из этапов в G10 не активен).

Принудительная команда, ассоциированная с этапом A6 диаграммы G1, принудительно переводит G10 в состояние, при котором активным является только этап 1 (однако ни один переход не сброшен).

Активация этапа F1 запускает нормальную эволюцию диаграммы G10 (так как она не зависит от принудительной команды).

Рисунок B.5 — Структурирование в режимах работы с использованием принудительных команд



Иерархия вложений содержит три уровня.

Когда этап D1 активен, никакой другой этап не является активным.

Когда этап A6 активен, никакой другой шаг не является активным и разрешено ручное управление (EMC).

Когда активируется охватывающий этап F1, активируется также этап 0 его вложения GM.

Когда активируется охватывающий этап 1, то активируются также этап 20 вложения «взвешивание», этап 30 вложения «транспортирование» и этап 40 вложения «смешивание».

Рисунок В.6 — Структурирование режимов работы с использованием охватывающего этапа

Приложение С
(справочное)

Взаимосвязь между языками спецификаций GRAFCET по IEC 60848 и SFC по IEC 61131-3

С.1 Введение

У каждого из стандартов, IEC 60848 и IEC 61131-3, имеется своя область распространения:

- для IEC 60848 — язык спецификации режимов работы (GRAFCET — графическое представление переходов и этапов), не зависящий от реализации какой-либо конкретной технологии;
- для IEC 61131-3 — специализированный язык программирования (SFC — последовательностная функциональная схема).

В GRAFCET по IEC 60848 для описания/спецификации поведения системы используется диаграмма grafcet, как бы при наблюдении системы «со стороны», в то время как язык SFC по IEC 61131-3 используется для описания программно реализованной структуры (ее части) «внутри» системы.

Если эти два языка применить для описания конкретной системы управления, то оба отображения (два документа различного вида) будут в данном случае выглядеть графически похожими. Тем не менее они имеют различный смысл, хотя и графически идентичны. Другими словами, структура программного обеспечения, описанная в программной диаграмме, ведет себя таким образом, что она может быть описана с использованием аналогичной диаграммы grafcet. Свойства основных элементов, связанных с представлениями графического элемента, тем не менее различны в этих двух случаях.

Использование спецификации по IEC 60848:2012 следует интерпретировать как этап перед разработкой программы по IEC 61131-3. В настоящее время отсутствует текстуальное представление диаграммы по IEC 60848:2012, позволяющее интерпретировать ее и превратить в программу для автоматической системы.

С.2 Детализация

a) GRAFCET обозначает язык спецификации логического поведения систем. Эта спецификация не зависит от реализации конкретной технологии.

b) SFC обозначает один из языков программирования, определенный в IEC 61131-3:1993. Этот язык инициирован IEC 60848:1988, но отсутствует тождество между двумя графическими изображениями и семантикой обоих языков.

c) В настоящее время не известно ни одного метода трансляции языка спецификации GRAFCET в программу на языке SFC: для этого требуется перевести теоретическую семантику первого языка в семантику реализации второго языка. Поэтому более синтетический и более эргономичный язык GRAFCET используют, как правило, для глобальной спецификации, в то время как язык SFC — для детализированного представления.

d) При пересмотре IEC 60848:1988 были сохранены сложившийся синтаксис первого издания и независимость по отношению к реализации, но были введены следующие модификации:

- более четкое определение элементов и правил эволюции (внутренние события, различие между присваиванием и локализацией, эволюция перехода и т. п.);
- как следствие, исключается потребность в более тонкой интерпретации спецификации поведения с учетом принципа или алгоритма реализации, чтобы понять эволюцию в сложных случаях;
- появление вспомогательных концептуальных инструментов, способных подтвердить спецификации, в частности путем расчета доказательности свойств;
- появление вспомогательных инструментов для реализации, способных гарантировать трансляцию в диаграммы или на языки, адаптированные к выбранным технологиям;
- более четкое определение средств структурирования (макроэтапы, охватывающие этапы и т. п.), ассоциированных с языком GRAFCET, способных поддерживать использование эффективных концептуальных методов;
- более совершенная формализация интерфейсов (предикат, локализация) между логическими переменными, с которыми обращается GRAFCET, и другими типами переменных, необходимых для целостного описания поведения системы, заданной к реализации.

e) Поиск систематической идентичности между элементом языка GRAFCET и соответствующим ему элементом языка SFC приводит к наложению запрограммированных семантик второго языка на первый. Подобный подход будет сводить роль стандарта GRAFCET, IEC 60848:1988, исключительно к общему определению символов и правил, предназначенных для поддержки определений элементов SFC по IEC 61131-3:1993.

f) IEC 61131-3:1993 определяет языки программирования, специально адаптированные для ПЛК (программируемые логические контроллеры). Поэтому представляется маловероятным, что данное программное описание алгоритма работы системы может быть применено в других технологических контентах (электромеханические, электронные, пневматические системы и т. п.).

g) Развитие распределенных автоматизированных устройств, предварительно смонтированных и/или предварительно запрограммированных, выдвигает необходимость описания поведения (гарантированного производителем) этих новых устройств формализованным и эргономичным образом на языке, способствующем их встраиванию в автоматизированную систему.

С.3 Прогноз развития

По сравнению с другими нестандартными представлениями кандидатов (например, диаграмма состояния) GRAFCET обладает преимуществами, обусловленными опытом, приобретенным большим числом конструкторов автоматов. Данное преимущество может быть использовано путем развития стандарта для расширения сферы применения этого языка и для того, чтобы сделать его более формальным и структурированным.

Будущий комплексный подход, рассматриваемый подкомитетом 3В МЭК и Техническим комитетом 65 МЭК, предусматривает текстуальный вариант IEC 60848:1988 в целях его автоматического преобразования в структуры и программы последующих редакций IEC 61131-3:1993 и IEC 61499 соответственно.

Библиография

IEC 60617	Graphical symbols for diagrams (Графические символы для диаграмм)
IEC 61131-3:2003	Programmable controllers — Part 3: Programming languages (Программируемые контроллеры. Часть 3. Языки программирования)
ISO 5807:1985	Information processing — Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts (Обработка информации. Символы, применяемые в документации, и обозначения для блок-схем данных программ и систем, схем программных сетей системных ресурсов)

УДК 681.3.04:006.354 МКС 35.080 Т55 IDT

Ключевые слова: язык спецификаций, grafcet, последовательная функциональная схема, автоматические системы

Редактор *Г.Н. Симонова*
Технический редактор *И.Е. Черепкова*
Корректор *Е.Р. Ароян*
Компьютерная верстка *Ю.В. Поповой*

Сдано в набор 22.01.2019. Подписано в печать 30.01.2019. Формат 60 × 84 1/8. Гарнитура Ариал.
Усл. печ. л. 5,58. Уч.-изд. л. 5,05.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

ИД «Юриспруденция», 115419, Москва, ул. Орджоникидзе, 11.
gost.e-ecolog.ru gost.e-ecolog.ru

Создано в единичном исполнении ФГУП «СТАНДАРТИНФОРМ»
для комплектования Федерального информационного фонда стандартов,
117418 Москва, Нахимовский пр-т, д. 31, к. 2.
gost.e-ecolog.ru gost.e-ecolog.ru

gost.e-ecolog.ru