



Рис. 4. Эпюры напряжений, иллюстрирующие работу многоуровневого оптического элемента в регулируемом умножителе частоты:
 $U_{вх}$ — напряжение на входе элемента; $U_{вых}$ — напряжение на выходе формирователя импульсов; t — время

Приведенные примеры показывают, что предлагаемый многоуровневый оптический элемент позволяет существенно упростить процесс преобразования информации и увеличить скорость ее обработки за счет существенного упрощения идентификации уровней состояний элемента и может быть использован при построении многоуровневой оптической логики, в быстродействующих АЦП, регулируемых умножителях частоты и т. д.

СПИСОК ЛИТЕРАТУРЫ

1. Gibbs H. M. Optical bistability: controlling light with light.— N. Y.: Academic Press, 1985.
2. Smith P. W., Turner E. H., Maloney P. J. Electrooptic nonlinear Fabry-Perot devices // IEEE J. of Quant. Electron.— 1978.— QE-14, N 3.— P. 207.
3. А. с. 1509809 СССР. Оптический мультистабильный элемент/А. А. Жмудь.— Опубл. 23.09.89, Бюл. № 35.
4. Бори М., Вольф Э. Основы оптики.— М.: Наука, 1973.
5. Жмудь А. А. Перестраиваемые характеристики полупроводниковых инжекционных лазеров // Автометрия.— 1989.— № 2.

Поступило в редакцию 29 сентября 1988 г.

УДК 63.072 : 681.3

В. С. ЛОПАТИН, И. Е. МЕДВЕДКОВА, В. Е. МЕЖОВ
 (Воронеж)

РЕАЛИЗАЦИЯ ИЕРАРХИЧЕСКОГО МОДЕЛИРОВАНИЯ В ПАКЕТЕ ПРИАМ

Разработка пакета программ иерархического алгоритмического моделирования (ПРИАМ) [1], который осуществляет поведенческое моделирование проектов вычислительных систем, обеспечивает возможность организации сквозного цикла автоматизированного проектирования различных изделий, и в первую очередь полужаказных БИС на системах проектирования типа «Кулон-4». В данной заметке описан основной алгоритм моделирования, заложенный в основу пакета.

Проекты вычислительных систем, моделируемые с помощью пакета ПРИАМ, могут содержать широкий диапазон уровней абстракции (системный, архитектурный, регистровый, логический).

© 1990 Лопатин В. С., Медведкова И. Е., Межов В. Е.

Проект моделируемой системы (в дальнейшем будет употребляться термин **проект**) описывается следующим образом. Его структура задается с помощью процедурного языка ЯЗКОП как совокупность «черных ящиков», выводы которых соединены между собой линиями связи. В дальнейшем они будут называться просто **линиями**. Алгоритм функционирования (или **модель**) «черного ящика» создается пользователем на процедурном паскалеподобном проблемно ориентированном языке высокого уровня АЛОС. Модель включает в себя списки, характеризующие выводы и локальные переменные модели, и собственно описание алгоритма функционирования, который рассматривается как совокупность параллельно выполняемых процессов. Процесс есть ключевое понятие языка АЛОС. Моделирование проекта базируется на обращении к алгоритмам функционирования компонентов проекта, причем описание вычислительных систем «сверху вниз» (от основных понятий до логической схемы) осуществляется общими методами, что позволяет проводить моделирование многоуровневых схем, содержащих комбинацию уровней абстракции во время моделирования.

Рассмотрим организацию процесса моделирования в пакете ПРИАМ. Поскольку алгоритм моделирования должен обеспечивать реализацию возможностей языка по способу учета временных характеристик моделей и способу представления потоков данных, рассмотрим соответствующие особенности языка АЛОС.

1. **Способ учета временных характеристик моделей.** Временные характеристики моделей используются в двух случаях: для задания времени изменения состояния на выводах модели и определения времени активизации процесса.

Средства, описывающие условия активизации процессов, очень важны, так как в значительной степени ими определяются границы применения языка. Язык АЛОС предполагает способы активизации процессов по контрольному списку и квалификатору.

Контрольным списком называется список выводов модели, за состоянием которых устанавливается контроль. Если хотя бы один из выводов контрольного списка процесса изменит свои значения, то этот процесс активизируется.

Задание способа активизации процесса, зависящего от модельного времени, проводится при помощи **квалификаторов**. Квалификатор может быть одного из двух типов: синхронизации или задержки.

Квалификатор синхронизации связывает активизацию процесса с одним из стробов, определяемых в общей части модуля. Сам квалификатор синхронизации задает номер фазы синхронизатора.

Квалификатор задержки задает задержку, т. е. количество тактов модельного времени, через которое периодически будет проводиться активизация процесса.

В языке АЛОС предусматривается также возможность задания условий выполнения процесса в виде некоторого логического выражения. Значение этого выражения вычисляется всякий раз при активизации процесса, и выполнение последнего произойдет только в случае, если это значение равно TRUE. Данная возможность очень важна, так как позволяет связать выполнение процесса со значением глобальных или локальных переменных.

Временные характеристики изменения состояния вывода задаются аналогичными способами. Различие состоит только в том, что квалификатор задержки понимается здесь как задержка на линии, а не как период, с которым происходит активизация. Еще одна особенность заключается в том, что дополнительные условия для активизации линий, кроме временных, не задаются.

2. **Способы представления потоков данных.** Наличие гибкой системы представления различных типов данных имеет исключительную важность для языка описания аппаратуры как основного средства адекватного представления потоков данных системы. Именно исходя из этого были выбраны и метод обработки (компилятивный) исходной информации, и ориентация на паскалеподобный язык.

В итоге пользователю пакета ПРИАМ доступен широкий диапазон типов данных для объявления констант и переменных, реализованных в языке АЛОС. Язык АЛОС включает в себя все возможные типы данных языка Паскаль [2]:

- простые — целый, реальный, булевый, символьный;
- произвольные — перечисляемые и ограниченные;
- структурные — множество, массив, запись, указатель.

Язык АЛОС включает специфическое понятие — **тип линии**. Это подмножество типов данных, использование которых допустимо для описания потоков данных проекта.

Для типов линий допустимыми являются:

- а) типы — целый, реальный, булевый и перечисляемые;
- б) ограниченные типы — целый и перечисляемый;
- в) записи, в которых каждое поле из классов а и/или б;
- г) массивы типов из классов а, б или в.

Кроме того, для типов линий в языке АЛОС предполагается введение нового типа, не предусмотренного в Паскале, — регистра. Тип регистр используется для объявления битовых строк.

При описании модели каждому ее выводу ставится в соответствие тип линии. Выводы моделей, в свою очередь, связываются с выводами компонентов при описании структуры проекта.

3. **Алгоритм иерархического моделирования пакета ПРИАМ.** Рассмотрим те-

перь реализацию процесса моделирования. В пакете ПРИАМ использована модификация событийного алгоритма. Уточним основные понятия, которые будут далее использоваться.

Очередь событий — совокупность связанных списков, соотнесенных с определенным моментом модельного времени (**тактом**). Каждый из связанных списков определяет событие, т. е. совокупность изменений состояния моделируемого проекта, относящихся к конкретному такту.

Активность — элемент события, определяющий элементарное изменение состояния проекта. **Активизация** — извлечение активности из связанного списка и обращение к процедурам, ответственным за выполнение данной активности.

Планирование — это занесение активности в связанный список, определяющий событие того такта, когда должно произойти планируемое изменение в состоянии проекта.

Все процессы, активизация которых зависит от изменения состояния на данной линии, называются ее последователями. Перед началом моделирования для всех линий проекта создаются списки их процессов-последователей.

Отличие данной модификации от классического варианта событийного алгоритма [3] заключается в следующем:

1. В пакете ПРИАМ планируемой активностью может быть не только изменение состояния на линии, но и выполнение процесса, входящего в описание модели для одного из компонентов проекта. Поэтому событие состоит, собственно, из двух связанных подписок, один из которых содержит информацию о линиях, а другой — о процессах. Подписка линий включает следующую информацию: номер линии; новое значение линии; текущее значение линии.

Подписка процессов содержит: номер компонента; номер процесса; параметры для организации перевода этого процесса в состояние ожидания (данные параметры будут конкретизированы ниже). Обработка активности зависит от того, какому подписке она принадлежит. Элемент подписки линий обрабатывается в следующем порядке: проводится замена старого значения линии на новое; планируются процессы-последователи этой линии на текущий такт.

Элемент подписки процессов обрабатывается так: проверяется выполнение условия активизации процесса, и если оно не выполнено, то обработка процесса прекращается; выполняется процедура, реализующая данный процесс. Если при выполнении этой процедуры какому-либо выводу модели должно быть присвоено новое значение, то осуществляется планирование той линии, к которой подсоединен данный вывод на время, оговоренное временными характеристиками оператора присвоения. Если моделирование процесса не приводит к изменению значений выводов модели, то обработка его заканчивается.

2. Активизация процесса происходит не только при изменении состояния на выводе, но и в такты времени, определенные синхронизулами при заданном периоде.

3. Изменение состояния на линии осуществляется не только в моменты времени, определенные задержкой вывода, но и в следующих случаях: подача синхронимпульса и изменение состояния на заданных выводах. Поэтому планирование линии зависит от способа задания ее временных характеристик.

4. Поскольку значение на линиях проекта может быть любым из того диапазона типов линий, который разрешен языком АЛОС для типов линий (см. предыдущий раздел), невозможно до начала моделирования зарезервировать память для хранения значений на линиях проекта. Для того чтобы не сужать диапазон типов линий, ограничивая при этом возможности языка, и не резервировать для хранения значений на линиях максимально большую память, была организована система динамического распределения памяти. Основой этой системы являются:

— специальная организация структуры пакета, которая отражает разделение всех функций системы моделирования на **постоянные**, например, передвижение модельного времени, обращение к моделям и т. д., и **переменные**, к которым, в частности, относится и организация динамического распределения памяти для конкретного проекта;

— динамическое выделение памяти для хранения значений линий проекта при каждом планировании изменения значений на линии;

— специальная организация данных в виде объединенных в списки записей, поля которых определяются конкретными параметрами моделей данного проекта.

5. Процесс может находиться в одном из следующих состояний: активном (выполняемый в данный момент процесс); разрешенном (процесс не активен, но может стать активным при выполнении заданных условий его активизации); запрещенном (процесс даже при выполнении условий не активизируется); ожидания (процесс прерван и будет продолжен с того места, где был прерван, когда будут выполнены оговоренные условия).

Перевод процесса из одного состояния в другое осуществляется средствами **управления процессом**. Их можно разделить на средства для активизации процесса, запрещения и/или разрешения процесса другими процессами этой модели, перевода процесса в состояние ожидания.

Запрещение — разрешение процесса. По умолчанию любой процесс любой модели разрешен, т. е. при выполнении определенных пользователем условий он становится активизированным. Для перевода его в запрещенное состояние достаточно,

чтобы в любом из остальных процессов данной модели выполнялся оператор ЗАПРЕТИТЬ, относящийся к данному процессу. С этого момента модельного времени процесс становится запрещенным и остается в этом состоянии до тех пор, пока не будет выполнен оператор РАЗРЕШИТЬ. Эти операторы позволяют одним процессам управлять активностью других в пределах одной модели.

Переход процесса в запрещенное или разрешенное состояние организуется самой процедурой обработки операторов запрещения и разрешения путем переопределения признака разрешения процесса. Этим признаком служит номер процесса. Если он отрицателен, значит процесс запрещен.

Перевод процесса в состояние ожидания. Так же как языки СИМУЛА, SOI и другие [3, 4], АЛЮС позволяет переводить процесс в состояние ожидания при помощи оператора ожидания (ЖДАТЬ). Данный оператор определяет и условие возобновления процесса, т. е. перевода его из состояния ожидания в активное состояние. Причем выполнение процесса при переводе в активное состояние должно быть продолжено с того момента, на котором оно прервалось оператором ожидания. Средства задания условий возобновления процесса в операторе ЖДАТЬ те же, что и для условий активизации всего процесса.

Возможность перевода процесса в состояние ожидания реализована при помощи создания для линий временных списков процессов-последователей (помимо основных списков, о которых уже сообщалось и которые в том или ином виде всегда присутствуют в событийном алгоритме). Последовательность действий для перевода процесса в состояние ожидания такова:

1) включение этого процесса во временные списки тех линий, которые оговорены как контрольные в операторе ожидания (т. е. при изменении состояния на этих линиях процесс будет продолжен); процесс находится во временных списках до тех пор, пока ожидание не будет прервано;

2) установление для процесса признака ожидания, наличие которого означает, что данный процесс может быть активизирован только из временного списка какой-либо линии; одновременно определяется метка именно данного ожидания этого процесса (число операторов ЖДАТЬ не ограничивается), что позволяет потом продолжить процесс с места, где он был прерван.

Параметрами ожидания, включаемыми в подсписок процессов, о которых упоминалось выше, являются, таким образом, признак и метка ожидания и, кроме того, указатель на первый из временных списков, в который был включен данный процесс при переводе его в состояние ожидания.

При продолжении прерванного процесса он исключается из временных списков и переопределяется признак ожидания.

СПИСОК ЛИТЕРАТУРЫ

1. Лопатин В. С., Медведкова И. Е., Межов В. Е. Основные принципы построения пакета иерархического моделирования ПРИАМ // Автометрия.— 1990.— № 5.
2. Прайс Д. Программирование на языке Паскаль: Пер. с англ.— М.: Мир, 1987.
3. Автоматизация проектирования вычислительных систем. Языки, моделирование и базы данных/Под ред. М. Брайера: пер. с англ.— М.: Мир, 1979.
4. Wirt N. Modula: a language for modular multiprogramming // Software — Practice and Experience.— 1977.— N 7.

Поступило в редакцию 12 октября 1989 г.

УДК 776 : 621.3 : 049.77

В. А. САТЫГА
(Новосибирск)

СКОРОСТЬ ДИФРАКЦИОННОГО ИЗМЕНЕНИЯ ГЕОМЕТРИИ ВСТРЕЧНО-ШТЫРЕВОГО ПРЕОБРАЗОВАТЕЛЯ

При получении встречно-штыревого преобразователя (ВШП) с помощью проекционной фотолитографии представляет интерес прогнозирование скорости дифракционного изменения геометрии, используемого в технологии для изменения размеров элементов ВШП при неизменном шаге.

Распределение освещенности в плоскости изображения идеализированной системы проекционной фотолитографии для ВШП с распределением освещенности в плоскости предмета типа меандр выразится формулой

$$E = E_0 + E_{из} \left(\sin \frac{2\pi}{l_{из}} x + \frac{1}{3} \sin \frac{3 \cdot 2\pi}{l_{из}} x + \dots + \frac{1}{2P+1} \sin (2P+1) \frac{2\pi}{l_{из}} x \right), \quad (1)$$

© 1990 Сатыга В. А.