ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ
Сборник трудов
Института математики СО АН СССР

1966 г. Выпуск 25

О МОДЕЛИРОВАНИИ ДИСКРЕТНЫХ АВТОМАТОВ НА УЦВМ

А.А. Уткин

I. Методы моделирования

Приступая к обсуждению методов моделирования, необходимо прежде всего определить то содержание весьма широкого понятия "моделирование", которое имеется в виду здесь. Речь будет идти только о таком моделировании, при котором предметом исследования является поведение дискретного автомата, а средством – универсальная цифровая вычислительная машина (УЦВМ).

Моделирование дискретных автоматов на УЦВМ рассматривается здесь как самостоятельное направление, ставящее целью повышение производительности труда при проектировании новых цифровых машин путем использования существующих. Число известных автору опубликованных по этой теме работ к настоящему времени составляет несколько десятков, обозначился круг задач и сформировались некоторые методы моделирования, применимые к автоматам определенного класса. Можно ожидать, что ввиду большой прикладной значимости обсуждаемого направления арсенал мето-

дов моделирования будет быстро обогащаться.

Метод моделирования дискретных автоматов удобно характеризовать тремя составляющими, к которым относятся:

I) входной язык, т.е. средства для описания структуры и поведения автомата;

2) совокупность программ, обеспечивающих воспроизведение поведения автоматов на УЦВМ, - то, что мы будем в дальнейшем называть моделирующей системой;

3) методика исследования автомата, т.е. способ задания его входной и анализа выходной последовательностей.

Наиболее существенные черты метода определяются первыми двумя составляющими; что касается третьей, то она может изменяться в зависимости от выбранной цели исследования. Такими целями могут быть, например, следующие: определение некоторых характеристик автомата (его производительности, коэффициента использования отдельных компонент и т.п.) ; обнаружение и локализация ошибок в заданной структуре; оценка функциональной устойчивости избыточной структуры; заблаговременная подготовка тестов и диагностических программ для проектируемой машины.

Хронологически первым из представленных в законченном виде методов моделирования является, по-видимому, метод, предложенный Стокуэллом [I] и несколько позднее- независимо от него развитый с несущественными отличиями Леманом с соавторами [2]. В качестве входного языка в этом методе используется язык булевых функций. Именно все элементы автомата описываются булевыми уравнениями, левыми частями которых являются переменные, поставленные в соответствие выходным сигналам элементов, а правыми - дизъюнктивные нормальные формы от переменных, поставленных в соответствие входным сигналам элементов. Символом переменной служит обозначение, присвоенное элементу, с выходного полюса которого снимается соответствующий данной переменной сигнал. Множество всех уравнений частично упорядочивается так, чтобы любая переменная, входящая в правую часть уравнения с произвольным номером, встречалась в качестве левой части хотя бы в одном из уравнений с меньшим номером, либо соответствовала выходному сигналу элемента с задержкой, либо, наконец, соответствовала одному из входных полюсов автомата.

Упорядоченная таким образом совокупность уравнений составляет описание структуры: это описание кодируется и вводится в моделирующую УЦВМ.

Временной аспект в явном виде в описании отсутствует, поскольку предполагается, что моделируемый автомат синхронизирован единственной тактовой последовательностью: изменение состояний его входных полюсов, равно как и вызванное им изменение состояний выходных полюсов элементов, лишенных задержки, происходит одновременно и мгновенно в начале такта, а длительность задержек равна целому числу тактов. Это допущение позволяет построить относительно простую моделирующую систему, в основе которой лежит идея потактного вычисления значений булевых функций, составляющих описание структуры. Частичная упорядоченность описания обеспечивает при этом определенность значения всех переменных, входящих в левые части уравнений. Замещение старых значений новыми для переменных, соответствующих выходам элементов без памяти, выполняется немедленно по вычислении нового значения, а для переменных, соответствующих выходам элементов с задержкой, - по окончании моделирования текущего такта. В конце такта выводится на печать совокупность значений выходных сигналов автомата.

Методика исследования автомата, использованная в [I] и [2], предельно проста: задается (или генерируется моделирующей системой) входная последовательность; получаемая в результате моделирования выходная последовательность поэлементно сравнивается с ожидаемой, которая, подобно входной, задается или генерируется. Каждый случай несовпадения отмечается специальным сигналом печатающего устройства либо запоминается. Обнаружение таких несовпадений и составляет цель исследования.

Способ описания структуры, использованный в цитированных работах, принципиально не отличается от классических способов, развитых в теории релейных схем. Это не умаляет, однако, значимости этих работ, так как она состоит прежде всего в практическом использовании законченного метода моделирования, в четкой постановке задачи моделирования и в демонстрации эффективности нового направления. Метод Стокуэлла-Лемана является значительным шагом вперед по сравнению с более ранними попытками моделирования (см., например, [3] - [5]).

Весьма близки к рассмотренному методу по принципам построения моделирующей системы и по классу моделируемых автоматов методы, предложенные в [6] и [7] . Их общей особенностью

является использование в качестве входного языка графического изображения структуры автомата, которое непосредственно кодируется и в виде нескольких списков вводится в УЦВМ (список элементов, список связей, дополнительные списки). Кодирование в этом случае оказывается более сложным, чем в первом методе. Методы различны по мощности набора элементов и по способу кодирования структуры . В [7] задача частичного упорядочения решается моделирующей УЦВМ; там же при построении моделирующей системы используется алгоритмический язык ЛЯПАС [8] .Другим примером применения алгоритмического языка (ФОРТРАН -IУ) может служить работа [9] , исходящая, как и [I],[2] , из булевого описания структуры.

Интересная разновидность метода Стокуэлла-Лемана представлена в работе [IO] , которая акцентирует внимание на исследовании влияния переходных процессов в реальных элементах на поведение автомата.

Существенно иные цели моделирования поставлены в [II]-[I3]: исследование вопросов временного совмещения, оценка выбранной системы команд, определение эффективного быстродействия проектируемой машины.При этом, естественно, снимается вопрос о детальном описании структуры, и центр тяжести переносится на такие вопросы, как описание поведения блоков автомата и взаимодействия между ними , моделирование течения времени и т.п.(см., например, работу [I3] , которая, кстати, иллюстрирует применение АЛГОЛа для построения моделирующей системы).

2. Проблемы моделирования

Продолжая обзор работ по моделированию , мы попытаемся сформулировать и кратко проанализировать основные проблемы обсуждаемого направления.

2.I. Снижение затрат ручного труда

Одним из самых серьезных препятствий на пути внедрения моделирования в практику является трудоемкость этапа составления описания и кодирования структуры автомата. Обеспечение удобств при составлении описания структуры автоматов широкого класса, легкости проверки описания и внесения поправок, а также создание средств для автоматического кодирования описания, объем которого, как правило, оказывается весьма большим, и

составляет содержание первой проблемы.

Эффективный путь решения этой проблемы состоит, по-видимому, в разработке специальных языков для описания структуры и поведения дискретных автоматов - подобно тому, как для решения проблем автоматизации программирования задач различных классов оказалось целесообразным создание алгоритмических языков. В этом отношении большой интерес представляет работа [I4], в которой обоснована целесообразность построения языка с упомянутым назначением, сформулированы требования к нему и приведен конкретный пример такого языка. На обсуждении этого языка мы остановимся подробнее несколько ниже, сейчас же уместно охарактеризовать конкурирующий подход к решению проблемы. Он состоит в использовании уже существующих алгоритмических языков, созданных для целей автоматизации программирования. Преимущества этого подхода очевидны: он позволяет использовать готовые трансляторы и накопленный опыт применения языков и, следовательно, избежать "капитальных затрат" по созданию новых языков и трансляторов к ним. Можно еще добавить, что использование алгоритмического языка, удобного для решения логических задач, является хорошим средством решения одной из рассматриваемых ниже проблем - совершенствования методики исследования автоматов. Однако, когда речь заходит о детальном исследовании тонких структурных свойств автомата, сразу начинает сказываться специфичность алгоритмических языков: будучи удобными в ряде случаев для представления поведения автомата, они не содержат средств для детального описания его структуры. Поэтому при разработке методов моделирования широкого назначения "капитальные затраты" по созданию специального языка и соответствующего транслятора могут быть вполне оправданы.

Язык для описания структуры и поведения цифровых машин (LOTIS), предложенный в [I4] , предусматривает различные уровни детализации описания автомата - в него введены средства как для структурного, так и для функционального описания. Одной из главных особенностей языка, делающей его пригодным для описания автоматов широкого класса, синхронных и асинхронных, является разнообразие способов задания времени. Наконец, описание автомата на языке LOTIS обладает иерархичностью, что обеспечивает легкость его синтаксической проверки и внесения поправок. Синтаксис языка подобен синтаксису АЛГОЛа, причем аналогом процедуры здесь является ш а г , описывающий ряд

элементарных действий, которые имеют потенциальную возможность осуществиться одновременно. Шаги объединяются в последовательности и функции, из которых, в свою очередь, составляются группы. Каждая группа описывает отдельное устройство машины. По мнению его автора, LOTIS может быть использован не только для целей моделирования, но и в смежных областях - например, при решении с помощью УЦВМ задач разбивки структуры на компоненты и вопросов схемной реализации.

Моделирующая система, входным языком для которой служил бы язык, подобный рассмотренному, была бы весьма мощным, универсальным аппаратом моделирования. Затраты на создание такой системы будут, по-видимому, выражаться величиной порядка нескольких человеко-лет.

Можно отметить еще ряд работ, касающихся построения языков для описания структуры и поведения дискретных автоматов. В [15], например, предложено понятие регистровой передачи (аналогичное понятие используется также в [14]). В [16] приведено краткое описание языка регистровых передач, который обсуждается с позиций, отличных от позиций моделирования. Именно, определены две формы описания автомата - на языке регистровых передач и на языке булевых функций. Перевод описания из первой формы во вторую рассматривается как решение задачи синтеза автомата, а обратный перевод - как решение задачи анализа. Показано, что обе эти задачи могут быть решены с помощью одного и того же синтаксически управляемого транслятора. В качестве последнего используется транслятор АЛГОЛа-60, управляемый так называемыми синтаксическими таблицами, формально определяющими упомянутые языки.

В [17] приведен пример языка для описания структуры автомата и для записи моделирующей программы; при этом предполагается первоначальное задание структуры в форме системы булевых функций и составление новой моделирующей программы для каждого нового объекта моделирования.

Еще одним примером языка для описания структуры дискретных автоматов является ЦИМОД [18], опирающийся на представление автомата как микропрограммно управляемой схемы и содержащий некоторые элементы языка АЛГОЛ-60.

Отметим еще систему LOCS [19], состоящую из языка для описания цифровых вычислительных машин и совокупности программ для ИБМ 7090 и предназначенную, в первую очередь, для моделирования цифровых машин на уровне "крупноструктурных" элементов и системы команд.

2.2. Расширение класса моделируемых автоматов

Как уже отмечалось, ряд практически важных целей моделирования требует детального структурного описания автомата. Оказывается, что подавляющее большинство развитых к настоящему времени методов моделирования применимы в этом смысле к автоматам вполне определенного класса, довольно узкого по сравнению с классом всех реально существующих и проектируемых автоматов. Именно они рассчитаны на моделирование синхронных автоматов с единственной синхронизирующей (тактовой) последовательностью, и в этом отношении характерными являются методы, приведенные в качестве примеров в разделе I. Таким образом, представляет интерес разработка методов моделирования автоматов со сложной системой синхронизации и асинхронных.

Разумеется, общим решением этой проблемы было бы создание универсального метода моделирования на базе гибкого и богатого специализированного языка для описания структуры и поведения автоматов широкого класса. Однако не следует отвергать и отдельных частных решений ввиду их меньшей стоимости и, возможно, большей простоты технической эксплуатации.

2.3. Обогащение методики исследования автоматов

Здесь имеется в виду развитие методов и алгоритмов испытания автоматов путем воспроизведения их поведения на УЦВМ. Достижение таких целей, как локализация ошибок в структуре, оценка функциональной устойчивости (надежности) избыточной структуры и т.п. потребует методик более тонких, чем поэлементное сравнение моделируемой выходной последовательности с ожидаемой. Об одном из возможных путей - исполнении пробных программ, записанных на языке моделируемого автомата - упоминается, например, в [14]. Более детально эта идея разработана в [19]. Рассматривая методику исследования в соответствии с принятой концепцией как одну из составляющих метода моделирования, можно сформулировать требование общего характера, которому должен удовлетворять хороший метод моделирования. Оно состоит в том, что эта составляющая должна быть наименее консервативной по сравнению с остальными двумя, легко изменяемой в зависимости от цели моделирования; изменение методики

не должны подразумевать изменений языка и моделирующей системы.

2.4. Обеспечение высокого быстродействия моделирующей системы

Эта проблема неизбежно возникает при разработке любого конкретного метода моделирования. Она становится особенно острой, когда моделирующая система предназначается для воспроизведения поведения автоматов, структура которых задается с высокой степенью детализации. Дело в том, что одновременное действие множества элементов реального автомата воспроизводится в УЦВМ, как правило, путем поочередного рассмотрения моделей этих элементов. Для сложных автоматов такое воспроизведение может потребовать столь значительного времени, что практическое выполнение желаемых испытательных программ окажется невозможным. Одним из путей преодоления этой трудности может служить применение языковых средств, подобных предусмотренным в языке LOTIS , - именно, использование различных уровней детализации в одном описании. Ясно, однако, что это не снимает обсуждаемой проблемы, так как и отдельные детально рассматриваемые части автомата могут оказаться достаточно большими.

Возможны два способа организации моделирующей системы. В первом из них описание структуры моделируемого автомата хранится в удобном виде в памяти моделирующей машины и периодически просматривается("сканируется") специальной программой , которая и воспроизводит поведение автомата.Такой способ характерен для большинства цитированных методов моделирования. Очевидно,что скорость сканирования существенно зависит от формы представления структуры и состояния автомата в оперативной памяти машины.Эта форма должна,прежде всего,обеспечить быструю выборку необходимой информации.Этому требованию,как правило,не удовлетворяет внешняя форма описания,используемая при его составлении.Таким образом,возникает задача перевода описания с входного языка на внутренний,принятый в машине;эта задача решается обычно отдельной программой, входящей в состав моделирующей системы. При выборе внутреннего языка приходится искать разумный компромисс между противоречивыми стремлениями обеспечить максимальные удобства сканирования и достигнуть компактности представления структуры. Аналогичное обстоятельство имеет место при выборе формы представления состояния автомата.

Существенного повышения быстродействия можно достигнуть, применяя специальные приемы сканирования. Подобного рода попытка предпринята, например в [2] , где было замечено, что статистически доля нулевых сигналов в схеме моделируемого автомата значительно превышает долю единичных, и предложен способ, названный "сканированием по выходам" (в отличие от ранее применявшегося "сканирования по входам"). Идея способа состоит в предварительном "запасании" в удобной форме значений входных сигналов элементов схемы. Другая попытка значительного повышения скорости сканирования представлена в предлагаемом ниже методе моделирования. По аналогии с [2] она названа "сканированием по возбужденным выходам", однако основная идея её существенно отличается от рассмотренной и состоит в том, чтобы пропускать, не сканируя, те части описания,которые касаются элементов, не возбужденных в данный момент времени.

Вторым возможным способом организации моделирующей системы является трансляция описания, дополненного некоторым исходным алгоритмом испытания, непосредственно в моделирующую программу, которая циклически исполняется, воспроизводя поведение автомата. Описание структуры и поведения автомата после такой трансляции, естественно, становится ненужным. Такой способ моделирования осуществляется в [9] . Быстродействие получаемой таким образом моделирующей программы определяется свойствами входного языка, качеством транслятора и формой представления состояния автомата в машине. Есть основания считать, что этот способ позволяет, вообще говоря, достигнуть более высокого быстродействия, чем первый, так как построение программы в этом случае основано на принципе компиляции, в то время как в первом используется режим интерпретации.

Отметим еще один круг вопросов, которого приходится касаться при практическом осуществлении моделирования. Речь идет о том, что можно назвать надежностью моделирования. При том большом объеме информации, который несет вводимое в машину описание структуры и поведения моделируемого автомата, неизбежны ошибки описания и кодирования. Следовательно, необходимо предусмотреть средства отладки описания - подобно тому, как в системах автоматизации программирования предусматриваются средства отладки программ.

В заключение необходимо подчеркнуть, что все перечисленные проблемы являются в настоящее время далеко не решенными.

Более того, история обсуждаемого направления столь коротка, что еще рано говорить о сколько-нибудь солидном опыте решения этих проблем. В то же время не вызывает сомнения актуальность задачи развития методов моделирования и внедрения их в практику - доводы в пользу этого можно найти в любой из цитированных работ. Поэтому ясно, что любая попытка решения тех или иных проблем моделирования будет представлять определенный интерес.

3. Моделирование синхронных автоматов со сложной системой синхронизации

Предлагаемый ниже метод моделирования был полностью отработан и опробован в марте-ноябре 1965 года. Таким образом, можно было бы изложить здесь этот метод во всех деталях, включая программы, входящие в состав моделирующей системы, и инструкцию по использованию метода. Однако, полагая, что далеко не все из этих деталей представляют интерес для читателя, мы ограничимся лишь обсуждением основных идей, положенных в основу метода. Более подробные сведения могут быть взяты из [20],[21].

3.1. Класс моделируемых автоматов

Рассматриваемый метод возник в результате попытки найти эффективный подход к моделированию автоматов, характеризующихся двумя особенностями. Первая заключается в том, что структура автомата содержит элементы импульсного типа. Существенной чертой такого элемента является то, что его выходной сигнал определен лишь в некоторых, достаточно малых, временных интервалах, разделенных интервалами, в которых значение этого сигнала не определено. Для обеспечения временного согласования в рассматриваемых автоматах используется не один, а ряд синхронизирующих периодических сигналов, причем различные подмножества элементов автомата инициируются различными сигналами, с разными периодами и фазами. Единицей измерения этих периодов и фаз может служить величина периода повторения некоторого сигнала, относительно которой все периоды и фазы выражаются натуральными числами и которая в дальнейшем называется микротактом. Наличие описанной системы синхронизации является второй особенностью рассматриваемого класса автоматов.

Типичными представителями этого класса являются схемы из ферротранзисторных элементов. Именно они послужили отправной точкой при разработке предлагаемого метода. Необходимость разработки нового метода моделирования диктовалась двумя причинами: 1) существующие способы описания структуры, опирающиеся на язык булевой алгебры, казались неудобными для применения к структурам из ферротранзисторных элементов. Действительно, отдельный ферротранзисторный элемент, представляющий, по существу, элемент памяти емкостью в один бит и управляемый несколькими функционально различными входами, не имеет удобного описания в булевых выражениях. Что касается совокупностей этих элементов, реализующих булевы функции, то весьма важным для них оказывается временной аспект, отсутствующий в обычном булевом описании; 2) применяемый в существующих моделирующих системах способ потактного вычисления значений выходных сигналов всех элементов схемы для автоматов рассматриваемого класса весьма неэффективен. В самом деле, за длительность такта здесь пришлось бы принять микротакт и выполнять огромную вычислительную работу впустую. Это станет понятным, если учесть, что в каждом из отдельных таких тактов большая доля элементов схемы не изменяет своего состояния и состояний, связанных с ними входов других элементов.

Целью моделирования с самого начала была, в первую очередь, проверка правильности структуры автомата, заданной с любой желаемой степенью детальности. К этому надо добавить, что исходная информация об автомате представлялась в виде так называемой принципиальной схемы, содержащей довольно полные сведения о тонкой структуре автомата и мало удобной для функциональных обобщений. Таким образом, необходимость разработки нового метода моделирования обусловлена как своеобразием класса моделируемых автоматов, так и формой представления исходной информации и характером цели моделирования.

Задачу проверки схемы автомата на отсутствие ошибок можно рассматривать как простейшее приложение аппарата моделирования. Предлагаемый метод решает эту задачу достаточно успешно. Что касается более сложных задач - таких, как локализация ошибок, экспериментальное исследование надежности и т.п., то для их решения, кроме разработки самого аппарата моделирования, необходимы специальные усилия по развитию методики исследования автоматов.

3.2. Язык

Предварительные замечания

Перед началом разработки нового метода моделирования были

рассмотрены несколько возможных путей его построения. Было решено, в частности, строить моделирующую систему на базе языка ЛЯПАС [8] с тем, чтобы воспользоваться системой автоматического программирования ПС-ЛЯПАС [22], хорошо зарекомендовавшей себя при решении разнообразных логических задач. О преимуществах, получаемых при использовании алгоритмического языка с готовой программирующей системой, говорилось выше. Естественно, что эти преимущества ощущаются наиболее полно, когда алгоритмический язык служит входным языком метода моделирования, т.е. в непосредственном виде используется для описания структуры моделируемого автомата. От этой возможности пришлось отказаться. Дело в том, что применение языков программирования для описания структуры автомата наталкивается на серьезные трудности, о которых говорилось в предыдущем разделе и источником которых является специфичность объектов, подлежащих представлению в языке. Было неясно, насколько успешно удастся справиться с этими трудностями за ограниченный срок.

В результате выбор был остановлен на следующем комбинированном пути. Моделирующая система записывается на языке ЛЯПАС; что же касается структур автоматов заданного класса, то для их описания разрабатывается узко специализированный символический язык, допускающий построение компактного и быстродействующего транслятора и обладающий некоторыми минимальными средствами для записи простейших алгоритмов испытания моделируемых автоматов.

Операнды и некоторые операции

В качестве атомарного объекта представления выбран элемент памяти емкостью в I бит. Операнд языка, названный одиночным элементом (о.э.), служит его прямой моделью. Определен ряд операций над одиночными элементами, с помощью которых можно описывать некоторые реальные элементы: (I) установка о.э. в состояние "I"; (2) установка о.э. в состояние "0"; (3) перевод о.э. из состояния "I" в состояние "0" с возбуждением выходного полюса (здесь и далее под возбуждением понимается присвоение выходному сигналу значения "I"); (4) возбуждение выходного полюса одиночного элемента, находящегося в состоянии "I", без изменения его состояния. Определены также операции (5,6) над произвольным (конечным) множеством одиночных элементов, интерпретируемых как входные полюса комбинационной схемы, реализущей произвольную булеву

функцию: вычисление значения булевой функции, заданной в дизъюнктивной (5) (конъюнктивной (6)) нормальной форме с переводом всех входных полюсов в состояние "0".

Другим объектом представления является упорядоченное множество элементов памяти (емкостью в I бит) фиксированной мощности (равной 32). Операнд, представляющий такое множество, назван регистром. Введены средства для оперирования с произвольной частью регистра, интерпретируемой как подмножество множества элементов, представляемых регистром. Компонента (разряд) регистра, представляющая один элемент такого множества, эквивалентна одиночному элементу (но имеет иное символическое обозначение), и над ней определены все те операции, которые определены над одиночном элементом. Операции (I) ÷ (4) определены также над регистром и его частью (как покомпонентные операции). Кроме того, над регистром определены еще некоторые операции (например, (7) - правый и (8) - левый сдвиги на заданное число разрядов с возбуждением выходных полюсов тех компонент, которые до сдвига находились в состоянии "I").

Остальные операнды языка - ячейку памяти и её часть мы оставим без пояснений, отметив лишь, что различие между ячейкой памяти и регистром заключается в различик между наборами операций, определенными для первой и для второго.

О символике и синтаксисе

Приняты следующие символические обозначения для операндов.

Одиночный элемент обозначается присвоенным ему номером, регистр - буквой r с нижним индексом, являющимся номером этого регистра. Все номера, о которых речь идет здесь и ниже, представляются в восьмеричной системе счисления.

Часть регистра, состоящая из произвольно выбранных компонент, обозначается символом этого регистра с добавлением перечня номеров выбранных компонент, заключенного в круглые скобки. Компоненты нумеруются подряд, начиная с крайней левой, числами от 0 до 37. В случае единственной компоненты скобки можно опустить. Примеры: r_{15} (7,0,I,30), r_0 (20), $r_0 20$.

Часть регистра, состоящая из плотно следующих друг за другом компонент, обозначается добавлением к символу регистра пары номеров граничных компонент, заключенной в квадратные скоб-

ки; при этом меньший номер пары следует первым.

Пример: $\tau_{100}[5,35]$.

Ячейка памяти обозначается буквой m , за которой следует номер (адрес) ячейки. Такая адресация называется прямой. Возможна косвенная адресация, когда номером ячейки служит значение содержимого некоторого регистра, трактуемое как число. В этом случае за символом m следует обозначение этого регистра. Выделение части допускается только для прямо адресованной ячейки. Примеры: m I, $m\tau_{20}$, m I00 [I5 , 37] .

Простейшим законченным выражением в языке является операция, запись которой состоит из символа оператора и следующих за ним символов его операндов. Для обозначения операторов используются как буквенные символы, некоторым образом отражающие смысл операторов, так и небуквенные. Последние отчасти взяты из набора обозначений, применяемых в схемах из ферротранзисторных элементов. Вот примеры операций, рассмотренных под предыдущей рубрикой (в порядке введенной там нумерации):

(I) " $\vdash\!\circ$ I2"; (2) " $\vdash\!\!\rightarrow$ I2" ; (3) " $\vdash\!\!\bullet$ I2"; (4) " $\vdash\!\blacksquare$ I2" ;

(5) "$DK(0,1,2/0,11/0,'1,'2',11'/\tau_0 0,\tau,5'/10,\tau_0 0')$";

(6) "$KD(\tau_0 0,10'/5',47)$"; (7) "$ПC1\tau_7$"; (8) "$ЛC24\tau_0$".

Здесь кавычками выделены выражения в языке. Символ / отделяет одну от другой элементарные конъюнкции (дизъюнкции), входящие в дизъюнктивную (конъюктивную) нормальную форму булевой функции, значение которой вычисляется оператором ДК(КД). Сами конъюнкции (дизъюнкции) задаются перечнем операндов, с выходных полюсов которых снимаются сигналы, являющиеся значением переменных, входящих в конъюнкцию (дизъюнкцию). Символ "штрих"(') означает, что соответствующая переменная входит в данную конъюнкцию (дизъюнкцию) с инверсией.

Важную роль в языке играет операнд, названный п о д р а з у м е в а е м ы м . Он не имеет никакого символического обозначения, а по своему смыслу соответствует выходному сигналу какого-либо элемента схемы. Точнее, для каждого данного оператора значением подразумеваемого операнда является результат предыдущей операции; в свою очередь, любой оператор изменяет, вообще говоря, значение этого операнда, присваивая ему результат выполненной операции. Действия всех рассмотренных выше операторов не зависят от значения подразумеваемого операнда, и поэтому эти операторы относятся к классу безусловных. Существует другой класс операторов, называемых условными. Условный оператор действует только тогда, когда значение подразумеваемого операнда, трактуемое как число, не есть нуль; в противном случае действует ближайший о п е р а т о р н а ч а л а с т р о к и .

Строка представляет собой цепочку следующих друг за другом операций, начинающуюся оператором начала строки. Этот оператор, символом которого служит $\boxed{\cdot}$, вычисляет новое значение подразумеваемого операнда, беря в качестве такового значение выходного сигнала элемента, соответствующего операнду этого оператора. Этим операндом не может являться ячейка памяти или её часть. (Здесь будет удобно заметить, забегая вперед, что наш атомарный объект представления -одиночный элемент или компонента регистра,-будучи по существу элементом памяти в I бит, в машине должен представляться двумя двоичными разрядами, один из которых несет значение состояния, другой - значение выходного сигнала). Место операнда оператора начала строки может быть и не занятым; в таком случае значение подразумеваемого операнда остается прежним.

Пару операторов, условный и безусловный, будем называть парой взаимных аналогов, если их действия различаются только вышеописанной реакцией на значение подразумеваемого операнда. Символы условных операторов отличаются от символов соответствующих безусловных аналогов лишь наличием вертикальной (или горизонтальной) черты слева (или снизу). Так, все рассмотренные выше безусловные операторы имеют условные аналоги, обозначаемые следующими символами:

$\vdash\!\circ$ $\vdash\!\!\rightarrow$ $\vdash\!\!\bullet$ $\vdash\!\blacksquare$ ДК КД ПС ЛС

(условный оператор (у.о.) установки в состояние "0"; у.о. установки в состояние "I"; у.о. перевода из состояния "I" в состояние "0" с возбуждением выходного полюса; у.о. возбуждения выходного полюса элемента, находящегося в состоянии "I" без изменения этого состояния; у.о. вычисления значения дизъюнкции конъюнкций; у.о. вычисления значения конъюнкции дизъюнкций; у.о. правого сдвига; у.о. левого сдвига).

С помощью строк, содержащих условные и безусловные операторы, удобно описывать различного рода цепные и ветвящиеся структуры. Следующий пример иллюстрирует это. Строки

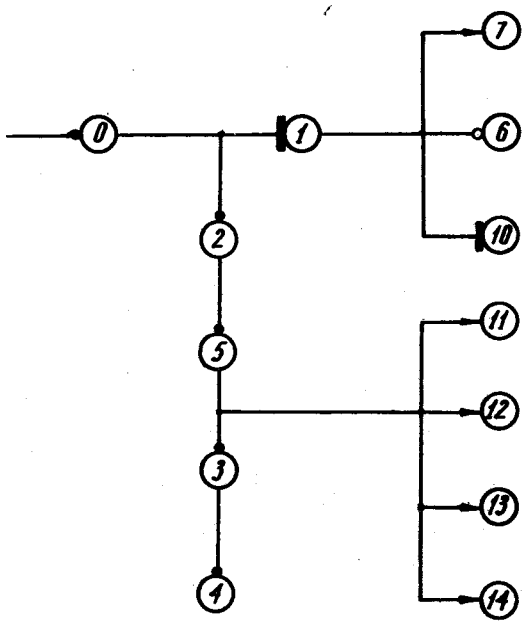
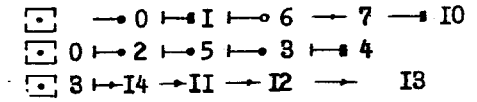


Рис. I



описывают структуру из одиночных элементов, представленную на рис.I. Изображения входов здесь совпадают с символами соответствующих операторов и не требуют дополнительных пояснений.

Остальные сведения о языке

В порядке уточнения определений некоторых из операторов, рассмотренных под предыдущей рубрикой, следует добавить, что операторы —• и ⊢—• присваивают подразумеваемому операнду значение выходного сигнала их операндов, операторы сдвига-значение сотояния регистра после сдвига,а операторы ДК,ДК,КД и КД-вычисленное значение булевой функции. Остальные из рассмотренных операторов пояснений не требуют,т.к. их действие сводится либо только к вычислению значения выходного сигнала,либо только к изменению состояния операнда.

Что касается области определения вышерассмотренных операторов, то для операторов сдвига она ограничивается регистром, для операторов вычисления значения булевой функции - одиночными элементами и компонентами регистра, для операторов —• , ⊢—• , —▪ ,⊢—▪ - одиночным элементом, регистром и частью регистра , для операторов —o , ⊢—o , —► , ⊢► она никак не ограничивается, т.е. операндом этих последних операторов может служить любой операнд языка.

Рассмотрим теперь остальные операторы языка.

Оператор присвоения ⇒ , определенный для регистра , его части, прямо адресованной ячейки памяти и её части, присваивает значение подразумеваемого операнда своему правому операнду. Точнее, этот оператор устанавливает правый операнд в состояние, определяемое значением подразумеваемого операнда. Следует заметить, что подразумеваемый операнд всегда представляется 32-мерным двоичным вектором, и в случае части регистра или ячейки памяти из этого вектора берутся лишь компоненты, одноименные с входящими в эту часть.

Оператор присвоения :⇒ присваивает значение подразумеваемого операнда косвенно адресованной ячейке памяти.

Эти два оператора не имеют условных аналогов.

Обратными к только что рассмотренным по своему действию являются операторы присвоения значения подразумеваемому опе-

ранду: операторы * и * * присваивают этому последнему значение своего правого операнда, при этом оператор * определен для регистра, прямо адресованной ячейки памяти и их частей, а оператор * * – для косвенно адресованной ячейки. Оба оператора имеют условные аналоги, обозначаемые через $\underline{*}$ и $\underline{**}$, соответственно.

Оператор объединения $\cup$ осуществляет поразрядное логическое сложение вектора, представляющего значение подразумеваемого операнда, с вектором, представляющим состояние его правого операнда, и присваивает полученный результат подразумеваемому операнду.

Аналогичным образом действуют операторы пересечения $\cap$, сложения по модулю два $\oplus$, арифметического сложения $+$, выполняя, соответственно, операции покомпонентного логического умножения, покомпонентного сложения по модулю два, арифметического сложения. Областью их определения являются регистр, ячейка памяти, их части, косвенно адресованная ячейка памяти.

Оператор инверсии $\neg$ осуществляет покомпонентную инверсию подразумеваемого операнда.

Последние пять операторов не имеют условных аналогов.

Оператор печати ПЧ и его условный аналог $\underline{ПЧ}$ выдают на печать значение состояния своего правого операнда, которым может служить любой операнд языка. Оператор печати массива ПМ и его условный аналог $\underline{ПМ}$ выдают на печать содержимое массива ячеек моделируемой памяти, заданного номером начальной ячейки и числом ячеек (мощностью массива). Нульместный оператор конца моделирования ! и его условный аналог $\underline{!}$ прекращают воспроизведение поведения автомата.

Характерной особенностью языка является способ задания времени, состоящей в разбиении описания автомата на части, каждая из которых представляет совокупность действий, имеющих потенциальную возможность одновременного осуществления. Такая часть описания носит название ф р а г м е н т а . Начало фрагмента отмечается меткой ФР, за которой следует натуральное число, являющееся номером фрагмента. Завершается фрагмент нульместным оператором " ; " конца фрагмента. В строке, содержащей этот оператор, не должно присутствовать ни одного условного оператора. Далее, для каждого фрагмента задаются его в р е м е н н ы е х а р а к т е р и с т и к и – период и фаза

синхронизирующего сигнала, инициирующего действия, описываемые этим фрагментом, и считается, что ни одно из действий не может осуществиться в те моменты времени, когда значение данного сигнала равно нулю.

Совокупность фрагментов составляет описание структуры автомата. Признаком конца описания служит метка "точка". Описание предваряется перечнем временных характеристик входящих в него фрагментов и заданием емкости памяти моделируемого автомата, выраженной числом ячеек памяти. Элемент перечня временных характеристик состоит из четырех символов: символа ФР, номера фрагмента, периода синхронизирующего сигнала (выражающегося целым числом микротактов) и его фазы по отношению к опорному сигналу (также выражающейся целым числом микротактов). Задание емкости памяти, замыкающее перечень, состоит из символа m и числа ячеек.

3.3. Составление входного описания

Входным описанием здесь называется описание структуры моделируемого автомата, выраженное средствами языка, воспринимаемого моделирующей системой, и дополненное алгоритмом испытания автомата, записанным на этом же языке.

Входное описание составляется по графической схеме, полностью отражающей структуру моделируемого автомата.

Рекомендуется следующая методика составления описания:

а) Отметить входные полюса, через которые поступает подлежащая переработке информация, и в случае необходимости ввести в схему входные регистры в качестве непосредственного источника входной последовательности. Аналогичным образом следует поступить с выходными полюсами.

б) Отметить на схеме все входы, на которые поступают синхронизирующие сигналы, и определить период и фазу каждого из этих сигналов.

в) Присвоить восьмеричные номера всем одиночным элементам, регистрам и разрядам регистров. Необходимо помнить, что в тех случаях, когда значение состояния регистра интерпретируется как число, веса разрядов определяются выражением 2^{37-i}, где i – номер разряда. Для двухтактных регистров сдвига, часто встречающихся в схемах из ферротранзисторных элементов, целесообразно применять однотактное представление, объединяя по паре смежных элементов в один разряд и заменяя впоследствии операцию сдвига в одном из тактов (именно в том, который не

переносит информации между образованными таким образом разрядами) операцией $\dashv z_j$, где j - номер регистра.

Следует учитывать, что в качестве регистра может выступать любое упорядоченное множество одиночных элементов (мощность которого не превышает 32). Такому множеству целесообразно приписывать обозначение регистра, если все его элементы подвергаются однотипному воздействию, инициируемому одним и тем же синхронизирующим сигналом. Такое обозначение не только сокращает описание, но и экономит затраты машинного времени при воспроизведении поведения автомата.

г) Составить описание структуры, разделяя его на фрагменты.

Здесь необходимо подчеркнуть один важный момент. Если в одном и том же микротакте оказываются возбужденными несколько фрагментов, то они рассматриваются моделирующей системой в порядке возрастания их номеров. Это обстоятельство надо учитывать при выборе номеров фрагментов. В частности, с целью улучшения обозримости описания можно разделять один большой фрагмент на несколько фрагментов с одинаковыми временными характеристиками, учитывая только что сказанное.

д) Задать алгоритм испытания моделируемого автомата, обеспечивающий подачу на входные полюса автомата генерируемой этим алгоритмом либо хранящейся в памяти автомата входной последовательности, установку начальных условий и окончание эксперимента. Этот алгоритм представляется одним или несколькими фрагментами, номера которых согласуются с номерами остальных фрагментов описания с учетом сделанного в предыдущем пункте замечания.

е) Составить перечень временных характеристик фрагментов, замыкаемый заданием емкости памяти моделируемого автомата. Этот перечень поставить в начале описания.

Проиллюстрируем изложенную методику небольшим примером, в качестве исходной информации для которого возьмем схему последовательного полного двоичного сумматора, представленную на рис.2. Здесь x и y - двоичные слагаемые, последовательно поступающие (младшими разрядами вперед) на схемы C и Π, вычисляющие, соответственно, значения очередного разряда суммы z и переноса v в следующий разряд; w - перенос в текущий разряд, образующийся из v с помощью элемента $з$ задержки на один такт. Схемы C и Π суть логические элементы, реализующие, соответственно, булевы функции

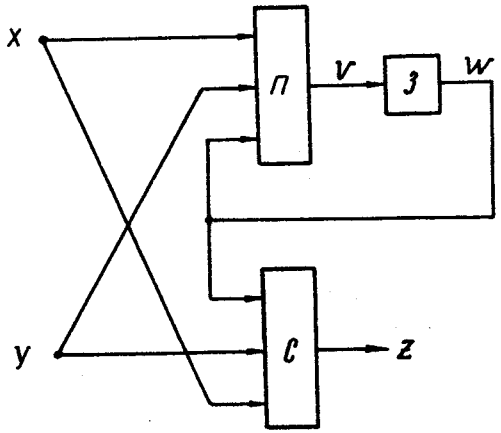


Рис. 2

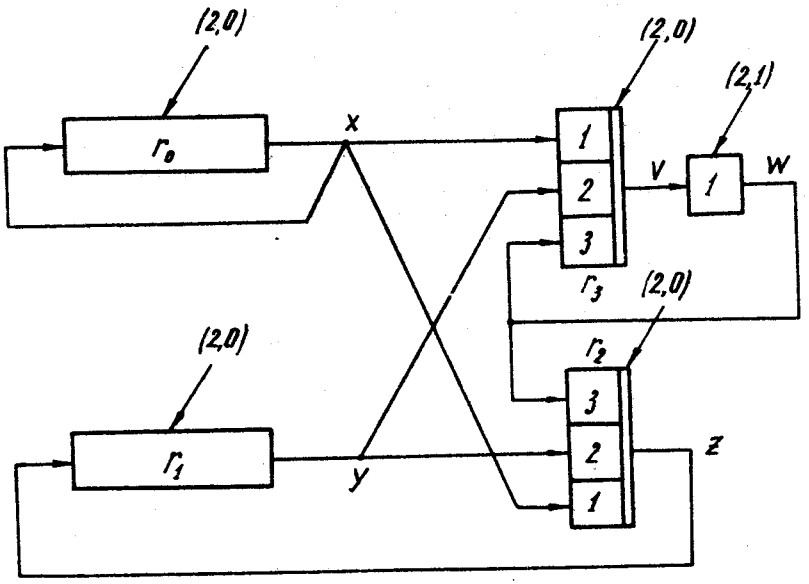


Рис. 3

$z = xy\omega + x'y'\omega + x'y\omega' + xy'\omega'$ и $v = xy + x\omega + y\omega$.

В соответствии с изложенной методикой пределаем следующее:

а) Введем входные регистры r_0 и r_1, из которых на схему будут поступать слагаемые x и y. Выходной полюс z замкнем на вход регистра r_1, а регистр r_3 сделаем регенеративным.

б) Сдвиг регистров r_0 и r_1 с подачей очередных разрядов слагаемых на схемы C и Π будем производить по синхронизирующему сигналу с периодом 2 и фазой 0; по этому же сигналу будем вычислять новые значения z и v; синхронизирующий сигнал с периодом 2 и фазой I будем использовать для передачи значения ω на входы схем C и Π.

в) Обозначим входы схемы C как разряды регистра r_2. Пусть слагаемое x поступает на разряд $r_2 1$, y – на $r_2 2$, ω – на $r_2 3$; аналогично через $r_3 1$, $r_3 2$, $r_3 3$ обозначим входы схемы Π, на которые поступают, соответственно, переменные x, y, ω. Элементу 3 присвоим номер I.

Дополненная таким образом схема с введенными обозначениями представлена на рис.3.

г) Составим описание полученной структуры. Оно будет состоять из двух фрагментов с временными характеристиками (2,0) и (2,I). Присвоим этим фрагментам номера 2 и 3, соответственно.

$ФР2\ ПС1 r_0\ ПС1 r_1\ ДК r_3\ (1,2/1,3/2,3) \mapsto 1$

$\boxdot\ ДК\ r_2\ (1,2,3/1',2',3/1',2,3'/1,2',3') \mapsto r_1 0$

$\boxdot\ r_0 17 \mapsto r_3 1 \to r_2 1 \to r_0 0$

$\boxdot\ r_1 17 \mapsto r_2 2 \to r_3 2$

$\boxdot\ ;$

$ФР3 \to 1 \mapsto r_2 3 \to r_3 3$

$\boxdot\ ;$

д) Зададим следующий простой алгоритм испытания схемы. В начальный момент в младший разряд регистра r_0 записывается единица. По прошествии каждых I6 периодов длиной 2, что соответствует одному циклу сложения I6-разрядных чисел, полученная сумма переписывается из регистра r_1 в ячейку памяти, номер которой задается содержимым некоторого вспомогательного регистра r_4. По истечении 32 периодов длиной 2 x I6 выдается на печать содержимое этих ячеек, и эксперимент заканчивается. Для описания этого эксперимента достаточно двух фрагментов с временными характеристиками (40,0) и (2000,0). При задании временных характеристик периоды и фазы представляются, как и любые другие натуральные числа во входном описании, в восьмеричной системе счисления: $(2 \times 16)_{10} = 40_8$, $(2 \times 16 \times 32)_{10} = 2000_8$). Присвоим им номера I и 0, соответственно:

$ФР0 \to 2\ \underline{ПМ}\ 1,40!$

$\boxdot \to 2 \to r_1 17 \to r_5 37;$

$ФР1 * r_1 :\Rightarrow m\, r_4 * r_4 + r_5 \Rightarrow r_4;$

Здесь элемент с номером 2 – вспомогательный, служащий для организации конца эксперимента; регистр r_5 хранит единицу в младшем разряде.

е) Входное описание вместе с перечнем временных характеристик и заданием емкости памяти выглядит теперь следующим образом:

```
ФР 2,  2,  0
ФР 3,  2,  I
ФР 0, 2000,    0
ФР I,  40, 0
m  40
```

$ФР2\ ПС r_0\ ПС r_1\ ДК r_3\ (1,2/1,3/2,3) \mapsto 1$

$\boxdot\ ДК\ r_2\ (1,2,3/1',2',3/1',2,3'/1,2',3') \mapsto r_1 0$

$\boxdot\ r_0 17 \mapsto r_3 1 \to r_2 1 \to r_0 0$

$\boxdot\ r_1 17 \mapsto r_2 2 \to r_3 2$

$\boxdot\ ;$

$ФР3 \to 1 \mapsto r_2 3 \to r_3 3$

$\boxdot\ ;$

$ФР0 \to 2\ \underline{ПМ}\ 1,40!$

$\boxdot \to 2 \to r_0 17 \to r_5 37;$

$ФР1 * r_1 :\Rightarrow m\, r_4 * r_4 + r_5 \Rightarrow r_4;$

Заключительным этапом подготовки входной информации является кодирование входного описания с помощью специальной кодировочной таблицы.

3.4. Моделирующая система

Моделирующая система состоит из двух программ, записанных на языке ЛЯПАС. Первая программа, транслятор, переводит входное описание в форму, удобную для использования внутри машины. Эту форму удобно трактовать как своего рода программу, воспринимаемую и исполняемую второй частью системы, операционным блоком. Будем называть эту программу моделирующей.

В существующем виде вся система рассчитана на использование одного лишь оперативного запоминающего устройства (ОЗУ) моделирующей УЦВМ емкостью не менее 4000 ячеек. При этом максимальный объем входного описания составляет примерно 3000 символов.

Представление операндов в моделирующей УЦВМ. Состояние каждого отдельного регистра представляется ячейкой ОЗУ. Одиночные элементы собираются в группы по 32 элемента, и состояние каждой такой группы представляется отдельной ячейкой ОЗУ так же, как состояние регистра. Массив ячеек, представляющих состояния одиночных элементов и регистров, будем называть комплексом состояний.

Второй массив ячеек, называемый комплексом выходов, используется для представления выходных сигналов одиночных элементов и регистров. Память моделируемого автомата естественным образом представляется третьим массивом.

Транслятор. Входное описание разделяется на две части. Из первой части, задающей временные характеристики фрагментов и емкость памяти, транслятор формирует таблицы периодов и фаз синхронизирующих сигналов; значение емкости памяти переписывается в отведенную для этой цели ячейку.

Вторая, основная часть описания, транслируется отдельно. Основными актами трансляции этой части являются следующие: 1) преобразование кодов операторов в коды операций; 2) преобразование кодов операндов в координаты операндов;

3) формирование таблиц начал строк и фрагментов.

Координаты операнда определяют место представляющих данный операнд разрядов ячейки в оперативном запоминающем устройстве машины. Именно первая координата задает номер элемента комплекса состояний (комплекса выходов, комплексов памяти). Она выражается числом, равным этому номеру. Вторая координата задает разряды ячейки; она представляется двоичным вектором размерности 32 (в соответствии с размерностью операндов ЛЯПАСа).

Стандартный цикл трансляции состоит из 1) поиска очередного кода операторов во входном описании; 2) преобразования этого кода в код операции и записи последнего в очередную ячейку комплекса кодов операций; 3) выборки кодов операндов, формирования координат операндов и записи их в соответствующие ячейки комплексов первых и вторых координат. Для оператора начала строки этот цикл дополняется формированием и записью адреса начала строки, а для метки начала фрагмента, кроме того, - формированием и записью адреса начала фрагмента.

Нестандартные циклы возникают при трансляции выражений, содержащих операторы вычисления значения булевой функции.

Заканчивается трансляция при обнаружении кода метки конца входного описания.

Трансляция входного описания максимального объема (не считая ввода) занимает на машине М-20 примерно 10 секунд. После выполнения своей работы транслятор становится ненужным, и его место в памяти машины занимается операционным блоком.

Операционный блок. В конструкции языка отражено стремление обеспечить возможно более высокое быстродействие моделирующей системы. Как только что было отмечено, скорость работы транслятора достаточно высока. Что касается операционного блока, то синтаксис языка позволяет осуществить способ весьма быстрого исполнения полученной моделирующей программы, который назван сканированием по возбужденным выходам.

Для каждого момента дискретного времени, в котором работает моделируемый автомат, на основании таблиц временных характеристик фрагментов легко определить, какие из фрагментов описывают действия, выполняемые элементами автомата в данный момент; эти фрагменты будем называть возбужденными. Отказ от рассмотрения всех тех фрагментов, которые не

являются в данный момент возбужденными, позволяет достичь высокой производительности моделирования, сокращая (часто -значительно) объем сканируемой части описания.

Дальнейшее сокращение этого объема происходит за счет "обрыва" строк: если выход операнда, предшествующего какому-либо условному операнду, в данном микротакте не возбужден, то совершается переход к началу следующей строки.

В операционном блоке можно выделить две части. Первая выполняет функции управляющей: она ведет счет времени (числа микротактов), находит очередной возбужденный фрагмент, формирует номер подлежащей исполнению команды моделирующей программы и анализирует код операции. Вторая часть представляет собой совокупность подпрограмм, каждая из которых реализует один из операторов языка. Выбор входа в эту часть осуществляется на основании анализа кода операции специальным коммутатором, входящим в состав управляющей части.

Стандартный цикл операционного блока соответствует воспроизведению поведения автомата в течение одного микротакта. Он начинается с того, что порядковый номер рассматриваемого микротакта увеличивается на единицу и определяются возбужденные в данном микротакте фрагменты. Кроме того, всем элементам комплекса выходов присваиваются нулевые значения. Затем в порядке возрастания номеров сканируются возбужденные фрагменты; при этом над операндами выполняются действия,предписанные соответствующими командами моделирующей программы. Цикл завершается, когда заканчивается сканирование последнего из возбужденных фрагментов.

Нестандартными циклами являются нулевой и последний.В нулевом цикле устанавливаются в нулевое состояние все одиночные элементы и регистры; последний цикл завершается операцией конца моделирования.

3.5. Заключение

Приведем некоторые количественные оценки возможностей рассмотренного метода, полученные путем предварительного расчета и подтвержденные во время опытной эксплуатации моделирующей системы.

Быстродействие моделирующей системы характеризуется средним количеством операций моделирующей УЦВМ, затрачиваемых на

воспроизведение поведения одного элемента автомата в течение одного микротакта.Оно составляет примерно 25 $\frac{\text{операций}}{\text{элемент.микротакт}}$

Так, при моделировании автомата из ферротранзисторных элементов, содержащего около 100 одиночных элементов и нескольких регистров, на реализацию программы испытания длиной в 1000 микротактов затрачивалось 1 мин 15 сек времени на машине М-20 (не считая, разумеется, ввода).

Максимальная сложность моделируемого автомата может быть охарактеризована числом его элементов. Для существующего вида моделирующей системы, использующей только оперативное запоминающее устройство, и для машины М-20, объем оперативной памяти которой составляет 4096 ячеек, эта сложность выражается примерно тремя сотнями одиночных элементов и несколькими десятками регистров.

ЛИТЕРАТУРА

1. G.N.Stockwell. Computer logic testing by simulation.-IRE Trans.Mil.Electr.,1962,Mil-6,N3,275-282.
2. M.Lehmar ,Eshed Rayna, Z.Netter. The checking of computer logic by simulation on a computer.-Comp.J. 1963, 6,N 2, 154-162.
3. S.R.Gray,R.N.Kisch. A progress report on computer applications in computer design.-Proc.1956 WJCC, San Francisco, Febr. 1956,82-85.
4. J.H.Chang,O.M.Georg. Use of high-speed digital computers to study performance of complex switching networks incorporating time delays.-Comm.and Electr., 1960, 46, 982-987.
5. А.А.Уткин. Программирование некоторых задач анализа и синтеза логических схем.-Вычислительная техника.Автоматика.Теория информации.(Труды Сибирского физико-технического института при ТГУ),Томск,1961, вып.40, стр. 145 - 159.
6. И.М.Ермолаева. Машинный контроль схем устройств.- Вычисл. и информ. техника, М., Изд. ВИНИТИ АН СССР, 1962 , стр. 123 - 124.

7. А.А.Уткин. Моделирование релейных схем.–Логический язык для представления алгоритмов синтеза релейных устройств, М., "Наука", 1966, стр. 284–300.

8. А.Д.Закревский. Автоматизация синтеза дискретных автоматов на основе алгоритмического языка ЛЯПАС.–Вычислительные системы, Новосибирск, 1965, вып. 18, стр. 5 – 33.

9. R.P.Larsen,M.M.Mano. Modeling and simulation of digital networks.–Commun.ACM,1965,8,N 5, 308–312.

10. И.Я.Ландау. Об автоматизации проектирования ЭЦВМ.–Автоматика и телемеханика,1964, 25, №11, стр.1581–1587.

11. H.P.Goodman. The simulation of the Orion time-sharing system on Sirius.–The Computer Bulletin,1961,5,N 2, 51–55.

12. I.M.Bennett,R.I.Dakin. Computers as an aid in computer design assesment.–The Comp.J.,1961,3,N 4,253–255.

13. С.Д.Михновский. Цифровое моделирование при синтезе ЦВМ.–Материалы научных семинаров по теоретическим и прикладным вопросам кибернетики КДНТП.–Вопросы теории математических электронных цифровых машин. Киев, 1963,вып.5,стр.14 – 27.

14. H.P. Schlaeppi. A formal language for describing machine logic, timing, and sequencing (LOTIS).–IEEE Trans. Electr.Comput.,1964, EC-13, N 4, 439–448.

15. I.L.Lebow. Communication in digital systems.–Informat. Theory, Fourth London Symposium, London, Butterworth Scientif.Publs., 1961, 99–107.

16. H. Schorr. Computer-aided digital system design and analysis using a register transfer language.–IEEE Trans. Electr. Comput.,1964,EC-13,N6,730–737.

17. R.M. McClure. A programming language for simulation digital systems.–Journ.ACM,1965,12,N 1,14–22.

18. Х.Салум. Язык для записи и моделирования работы цифровых структурных схем (ЦИМОД).–Изв. АН Эст. ССР, 1965, т. 14, серия физ.-мат. и техн. наук, № 13, стр. 464 – 472.

19. M.S. Zucker., LOCS: An EDP machine logic and control simulator.1965 IEEE Int. Conv. Rec., part 3, March 1965,28–50.

20. Отчет о научно-исследовательской работе "Автоматизация некоторых этапов проектирования специализированных вычислительных и логических устройств, реализуемых на заданной системе элементов", СФТИ, при Томском Госуниверситете им. В.В.Куйбышева, Томск, 1965, ч. I.

21. То же, ч. II.

22. М.Я.Товштейн. ПС-ЛЯПАС для машины М-20.–Данный сборник. стр. 87–116.

Сибирский физико-технический ин-т

Поступила в редакцию 7. III. 1966 г.