Н. Н. МАКСИМОВА, Н. В. ОРЛОВА,
Ф. З. РОХЛИН, В. Н. ШАМГУНОВА

МОДЕЛИРОВАНИЕ ПРОГРАММЫ ВЫСОКОПРОИЗВОДИТЕЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ МАШИНЫ НА МАШИНЕ МЕНЬШЕЙ ПРОИЗВОДИТЕЛЬНОСТИ

В настоящей статье рассматривается вопрос о моделировании на цифровой вычислительной машине (ЦВМ) выполнения программ, составленных в коде другой ЦВМ. Наибольший интерес представляет, по-видимому, случай моделирования большой ЦВМ на малой. Перечислим некоторые из возможных применений этого случая:

а) отработка системы команд проектируемой машины;

б) подготовка и отладка программ для машины, которая в ближайшее время должна быть получена данной организацией, с целью сокращения времени между установкой новой машины и фактическим ее использованием;

в) рациональное распределение функций между машинами различных классов: использование малых машин преимущественно для отладки программ, а больших — только для решения задач по отлаженным программам;

г) составление и отладка различных частей одной программы несколькими органицазиями, располагающими различными машинами;

д) обучение программированию на машинах, отсутствующих в данной организации.

Моделирующие программы, составленные для каждого из перечисленных применений, будут иметь, очевидно, много общего.

Вопросы, аналогичные рассматриваемым в этой статье, обсуждаются в ряде публикаций [1, 2].

Рассмотрим структуру моделирующей программы, предназначенной для отладки программ.

Назовем такую программу имитатором. Моделируемую ЦВМ и ЦВМ, на которой производится моделирование, будем называть соответственно машиной А и машиной В, а программу и команды, написанные в коде машины А и моделируемые на машине В, соответственно псевдопрограммой и псевдокомандами.

Имитатор должен быть таким, чтобы выполнялись требования удобства отладки псевдопрограмм:

1) все псевдокоманды, а следовательно, и псевдопрограмма в целом должны выполняться так же, как они выполнялись бы машиной А;

2) должна быть обеспечена возможность останова псевдопрограммы в любой момент ее выполнения и по любому адресу;

3) при нажатии на кнопку «пуск» на пульте машины В после одного из указанных в п. 2 : остановов должна выполняться следующая псевдокоманда псевдопрограммы;

4) необходима по возможности полная имитация пульта и панели сигнализации машины А;

5) необходима возможность работы имитатора в шаговом режиме, аналогичном шаговому режиму работы машины А;

6) желательно, чтобы структура имитатора позволяла достаточно просто вводить различные режимы работы, облегчающие отладку программ.

Чтобы удовлетворить этим требованиям, из двух возможных типов построения моделирующей программы — компилирующего и интерпретирующего — следует выбрать последний. Кроме того, если бы имитатор был построен по типу компилирующих программ, то при любом изменении псевдопрограммы приходилось бы каждый раз повторять компилирование.

Недостатком интерпретирующей системы является низкая скорость моделирования, что не имеет, однако, решающего значения в тех случаях, когда имеется в виду применение имитатора для отладки псевдопрограмм.

Сделаем некоторые предположения, касающиеся сравнительных характеристик машин А и В.

Оперативная память машины В меньше оперативной памяти машины А. Внешняя память машины В меньше внешней, но больше оперативной памяти машины А. Последнее обстоятельство позволяет проимитировать всю оперативную память и часть внешней памяти машины А, используя для этого внешнюю память машины В. Заметим в связи с этим, что при отладке программ, использующих внешнюю память, в подавляющем большинстве случаев достаточно иметь гораздо меньший объем внешней памяти, чем при работе тех же программ в режиме счета, в то время, как наличие всех ячеек оперативной памяти является почти всегда очень желательным.

Такие характеристики машины, как ее организация, набор и структура команд, разрядность ячеек памяти, форма представления чисел, система счисления, могут быть весьма различными в обеих машинах. Чем больше эти различия, тем сложнее будет имитатор.

Значительные трудности могут возникнуть при имитации устройств ввода и вывода машины А, если они сильно отличаются от соответствующих устройств машины В. Однако, имея в виду

отладку псевдопрограмм, почти всегда можно ограничиться лишь частичной имитацией этих устройств, поскольку обращение к ним, как правило, оформляется в виде подпрограмм.

Имитация основных регистров устройства управления, арифметического устройства и пульта машины А

Имитацию этих регистров осуществляют ячейки оперативной памяти машины В, причем в зависимости от разрядности каждого регистра, для его имитации фиксируется часть разрядов либо одной ячейки, либо нескольких ячеек.

Из регистров устройства управления следует имитировать счетчик команд, регистр команд, так называемые индикаторы особых ситуаций (в частности, триггеры, по состояниям которых происходят передачи управления при выполнении команд условной передачи управления, и триггер сигнала переполнения), индекс-регистры (если таковые имеются в машине А) и, вообще, все регистры, несущие определенную функциональную нагрузку с точки зрения программирования.

В зависимости от структуры арифметического устройства машины А могут имитироваться различные его регистры. В простейшем случае необходимо имитировать регистр результата.

Кнопки и тумблеры пульта имитируются также ячейками оперативной памяти машины В. Включению какого-либо тумблера на пульте машины А будет соответствовать занесение единицы в соответствующий разряд соответствующей ячейки машины В.

Панель сигнализации имитируется тем, что после любого останова псевдопрограммы печатается содержимое тех регистров арифметического устройства и устройства управления, которые выведены на панель сигнализации машины А.

Имитация оперативной памяти

Для имитации оперативной памяти машины А отводится часть внешней памяти машины В, которую для краткости будем называть ОЗУ машины А. Ячейкам ОЗУ машины *А* присваиваются соответствующие адреса оперативной памяти машины А.

С целью сокращения времени работы имитатора предлагается следующая система обмена информацией между ОЗУ машины А и оперативной памятью машины В. В оперативной памяти машины В выделяется n групп подряд расположенных ячеек, называемых в дальнейшем карманами. Для простоты будем считать, что карманы состоят из одинакового количества ячеек. В каждый карман может быть записано содержимое любых подряд расположенных ячеек ОЗУ машины А, число которых не должно превосходить емкости кармана.

Поскольку в процессе работы имитатора в один и тот же карман могут вызываться различные участки ОЗУ машины А, необходи-

мо предусмотреть достаточно гибкую систему привязки карманов к ячейкам ОЗУ машины А. В простейшем случае можно для каждого кармана зафиксировать по две ячейки оперативной памяти машины В (ячейки привязки), в которых будут храниться начальный и конечный адреса участка ОЗУ машины А, находящегося в данный момент в соответствующем кармане. При вызове в данный карман другого участка ОЗУ машины А эти адреса должны соответствующим образом изменяться.

Обмен информацией между карманами и ОЗУ машины А происходит следующим образом.

Пусть псевдопрограмма запрашивает ячейку с адресом a в оперативной памяти машины А. Тогда проверяется, не содержится ли ячейка с адресом a в одном из карманов. Это осуществляется посредством сравнения адреса a с адресами, записанными в ячейках привязки. Если требуемая ячейка содержится в одном из карманов, то вычисляется ее адрес в оперативной памяти машины В и запрос псевдопрограммы считается удовлетворенным.

В противном случае выбирается карман, о котором с большей степенью достоверности можно сказать, что вероятность обращения к ячейкам этого кармана в течение некоторого промежутка времени, отсчитываемого от данного момента, будет не больше вероятности обращения к ячейкам любого другого кармана в течение того же промежутка времени. Этот выбор делается на основе анализа так называемой системы приоритета, о которой будет сказано несколько ниже. Далее содержимое выбранного кармана записывается в ОЗУ машины А на место, определяемое парой адресов, находящейся в его ячейках привязки. Заметим, что в тех случаях, когда содержимое выбранного кармана не изменялось после последнего заполнения его информацией, то есть когда в нем находятся, скажем, константы или неизменяемые псевдокоманды, запись его содержимого в ОЗУ машины А является необязательной.

После выбора кармана и пересылки, в случае необходимости, его содержимого на прежнее место в ОЗУ машины А, в этот карман считается участок ОЗУ машины А, содержащий требуемую ячейку и не содержащий ячеек, хранящихся в данный момент в других карманах. Последнее требуется для того, чтобы исключить возможность одновременного присутствия одинаковых ячеек в различных карманах, так как наличие такой возможности привело бы к неоправданному усложнению имитатора. Одновременно с заполнением кармана в его ячейки привязки записываются начальный и конечный адреса соответствующего участка ОЗУ машины А. Затем, как и в первом случае, вычисляется адрес требуемой ячейки в оперативной памяти машины В, после чего запрос считается удовлетворенным.

Система приоритета должна в каждый момент времени указывать карман, обращение к которому в ближайшие последующие моменты времени будет происходить реже, чем к другим карманам.

Таким образом, система приоритета должна, так сказать, прогнозировать использование карманов в ближайшем будущем. Очевидно, что такого рода предсказания могут быть сделаны на основе анализа использования карманов в прошлом, то есть на основе учета предыстории работы псевдопрограммы.

Возможны различные реализации системы приоритета. Рассмотрим две из них.

В первой системе каждому карману сопоставляется так называемая ячейка приоритета. В эту ячейку засылаются нули в начале работы имитатора и каждый раз, когда происходит обмен информацией между ОЗУ машины А и каким-либо карманом. В промежутки между двумя последовательными обменами к каждой ячейке приоритета прибавляется единица только тогда, когда имитатор обращается к одной из ячеек соответствующего кармана. Для приема информации из ОЗУ машины А выбирается карман, в ячейке приоритета которого находится наименьшее число.

Вторая система организована аналогично предыдущей, каждому карману сопоставляется ячейка приоритета. В начале работы имитатора в эти ячейки в произвольном порядке записываются числа 1, 2..., n так, что в каждую ячейку записывается по одному числу и любая пара ячеек содержит различные числа. Этим задается нумерация карманов. Пусть имитатор обращается к карману с номером $i \neq 1$. Тогда все номера, меньшие i, увеличиваются на единицу, номер i заменяется на номер 1, а все остальные номера остаются без изменений. Если $i=1$, то нумерация карманов не изменяется. Для приема информации из ОЗУ машины А выбирается карман с номером ?

Последняя организация системы приоритета аналогична организации стопки книг, в которой каждая вынутая книга помещается на верх стопки. Очевидно, что при такой организации стопки, часто употребляемые книги будут иметь тенденцию скапливаться наверху. На таком же принципе основано устройство так называемой магазинной памяти, широко используемой в последних марках отечественных и зарубежных ЦВМ.

Идея о возможности применения организации стопки книг к системе приоритета имитатора была подсказана Х. Ф. Кулеевым, за что авторы выражают ему свою искреннюю благодарность.

Структура имитатора

Имитатор работает в общих чертах следующим образом.

Предположим, что в начале работы имитатора в регистре команд (точнее, в ячейке, имитирующей регистр команд) находится подлежащая исполнению псевдокоманда, а в счетчике команд — ее адрес в ОЗУ машины А. Входной блок приводит имитатор в исходное состояние. В блоке 2 происходит предварительная расшифровка псевдокоманды, состоящая в разделении всех псевдокоманд на два класса. К первому классу относятся

все псевдокоманды, кроме команд управления, адресные части которых содержат адреса оперативной памяти машины А, а ко второму классу — все прочие псевдокоманды. До отнесения псевдокоманд к тому или иному классу псевдокоманда приводится к так называемому исполнительному виду, то есть с ней производятся все необходимые модификации в соответствии с содержащимися в ней признаками и содержимым некоторых регистров (скажем, индекс — регистров). В дальнейшем имитатор работает уже с модифицированной псевдокомандой, называемой для краткости просто псевдокомандой.

Если псевдокоманда отнесена к первому классу, то по содержащимся в ней адресам делаются запросы, которые удовлетворяются в соответствии с вышесказанным, и управление передается блоку 3.

Если же псевдокоманда оказывается отнесенной ко второму классу, то управление сразу передается блоку 3.

Блок 3 по коду операции псевдокоманды передает управление соответствующей подпрограмме интерпретации. Если данная псевдокоманда принадлежит к командам передач управления, то после ее интерпретации в

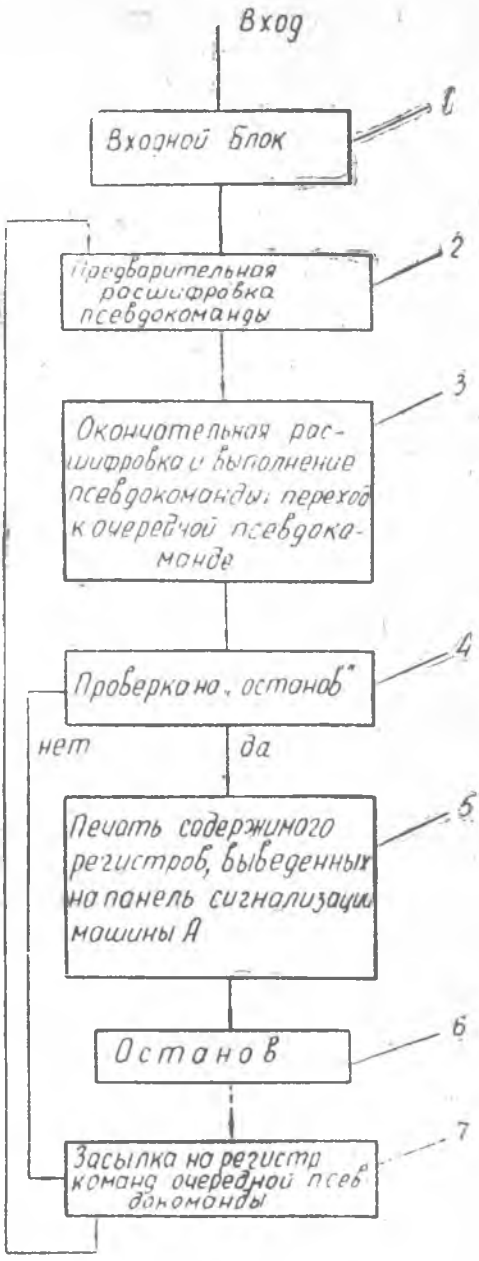

Рис. 1. Блок-схема имитатора

счетчике окажется адрес очередной псевдокоманды. В противном случае соответствующая подпрограмма выполняет требуемые псевдокомандой преобразования информации и вырабатывает значения соответствующих сигналов (например, сигналов ω, φ и т. п.). После этого к содержимому счетчика команд прибавляется единица.

В блоке 4 проверяется, можно ли продолжать выполнение псевдопрограммы или необходимо остановиться, так как создалась ситуация, при которой в машине А происходит останов (например, выполняемая псевдокоманда является командой останова, или содержимое счетчика команд совпадает с адресом, набранным на адресном регистре пульта управления, или включен тумблер «остановка» по φ и $s=1$).

Пусть дальнейшее выполнение псевдопрограммы оказалось возможным. Тогда после блока 4 выполняется блок 7, в котором по адресу, находящемуся в счетчике команд, происходит запрос. После удовлетворения запроса на регистр команд помещается псевдокоманда из соответствующей ячейки оперативной памяти машины В (из соответствующего кармана).

На этом выполнение одной псевдокоманды заканчивается. После блока 7 управление передается блоку 2, и описанные процедуры выполняются в том же порядке, но уже по отношению к новой псевдокоманде.

Изменение состояния системы приоритета производится в блоках 2 и 7.

Если в результате проверок, выполняемых блоком 4, выяснилась необходимость остановки программы, то перед остановкой печатается (блок 5) содержимое ячеек, имитирующих те регистры машины А, которые выведены на ее панель сигнализации. Если после остановки нажать на кнопку «пуск» машины В, то управление передается блоку 7 (пунктирная стрелка) и выполнение псевдопрограммы продолжается.

Имитатор ЦВМ «Урал-4» на ЦВМ «Минск-11»

В соответствии с изложенной методикой была составлена в коде «Минска-11» программа, имитирующая работу «Урала-4». Для имитации одной неполной ячейки «Урала-4» зафиксированы первые 20 разрядов ячейки «Минска-11». Для имитации оперативной памяти «Урала-4» (4096 ячеек) отведено 4 зоны на магнитной ленте «Минска-11». В оперативной памяти «Минска-11» (2048 ячеек) может быть размещено до 18 карманов по 64 ячейки в каждом. Предусмотрена возможность увеличения (уменьшения) емкости карманов с соответствующим уменьшением (увеличением) их числа. Остальная часть оперативной памяти занята имитатором. Принята первая из описанных система приоритета.

Имитатор создавался для отладки программ логических задач (в частности, задач, связанных с машинным переводом). Поэтому в первую очередь были проимитированы операции сложения и

вычитания с фиксированной запятой, все логические операции и все операции управления. Однако структура имитатора позволяет вводить интерпретацию новых псевдокоманд путем незначительных локальных изменений. Поэтому список интерпретируемых псевдокоманд может быть при желании расширен вплоть до полного списка команд «Урала-4».

К настоящему моменту введены 5 режимов работы имитатора: один для имитации автоматической работы «Урала-4» (в соответствии с приведенной на рисунке блок-схемой) и четыре других для облегчения отладки псевдопрограмм. В одном отладочном режиме после выполнения каждой псевдокоманды производится печать основных (интересных с точки зрения отладки) регистров устройства управления и арифметического устройства. В трех других происходит печать тех же регистров при выполнении псевдокоманд, у которых адреса ячеек, в которых они расположены, коды операции или адресные части соответственно принадлежат заданным множествам. В каждом из трех режимов имеется таблица для записи соответствующего данному режиму множества.

Существует достаточно простая методика перехода от одного отладочного режима к другому во время выполнения одной псевдопрограммы. При необходимости можно довольно просто вводить другие отладочные режимы.

Скорость имитатора (без учета времени обмена между ОЗУ «Урала-4» и карманами) равна 15÷20 псевдокоманд в секунду (скорость «Минска-11» — 2000÷3000 операций в секунду).

Более чем годовой опыт работы с имитатором показал его приемлемость для целей отладки. При этом время отладки программ на «Урале-4» и на имитаторе примерно одинаково.

ЛИТЕРАТУРА

1. Н. М. Ермолаева, М. А. Пробст. Имитатор параллельной машины на последовательной машине «Гамма-Барабан». «Информационные системы», Институт научной информации АН СССР, 15—17 М., 1964.

2. I keno Nobuichi, Arai Katuhiko, Kimura Ken. Program simulation for a large scale computer with a small computer. «Rev. Electr. Commun. Lab.» 1964, 12, № 9—10, 644—649.