

функций, при этом их число окажется значительно меньшим, чем в первом случае. Такой подход позволит упростить процесс достижения заданного режима намагничивания и одновременно обеспечит решение задачи сжатого представления информации о магнитных свойствах испытуемого феррообразца, поскольку при достижении заданного режима становится известным спектральный состав намагничивающего сигнала.

ЛИТЕРАТУРА

1. Дружинин В. В. Магнитные свойства электротехнической стали.— М.: Энергия, 1974.
2. Шрамков Е. Г. Задачи, связанные с оценкой свойств ферромагнитных материалов.— В кн.: Новые методы и аппаратура для испытания ферромагнитных материалов: Труды институтов комитета. М.— Л.: Стандартгиз, 1962, вып. 64.
3. Ольховский Ю. Б., Новоселов О. Н., Мановцев А. П. Сжатие данных при телеизмерениях.— М.: Сов. радио, 1971.
4. Перов В. П. Прикладная спектральная теория оценивания.— М.: Наука, 1982.
5. Люстерник Л. А., Соболев В. И. Краткий курс функционального анализа.— М.: Наука, 1982.
6. Векслер А. З., Дружинин В. В. Стандартизованные испытания электротехнической стали на переменном токе.— Электротехника, 1965, № 2.
7. Талдыкин А. Т. Элементы прикладного функционального анализа.— М.: Высшая школа, 1982.
8. Воеводин В. В. Линейная алгебра.— М.: Наука, 1980.
9. Маслов Ю. Н., Музюкин В. П. К вопросу разработки устройств для получения синусоидальной магнитной индукции в ферромагнитных образцах.— В кн.: Электротехника и приборостроение: Сб. науч. трудов. Владимир: ВПИ, 1970, вып. 10.

*Поступила в редакцию 16 декабря 1982 г.;
окончательный вариант — 23 августа 1983 г.*

УДК 681.3.068

Е. Г. БРЫНДИН

(Новосибирск)

СХЕМОПРОЕКТИРОВАНИЕ МОДУЛЕЙ ИНТЕРПРЕТАТОРА ЦЕЛЕВОЙ ЭВМ

В этой статье делается попытка подойти к рассмотрению вопроса частичной автоматизации процесса построения модулей интерпретатора ЭВМ. Полностью автоматизировать этот процесс не удастся из-за того, что трудно построить, во-первых, унифицированный интерпретатор, который включал бы систему алгоритмов, имитирующих команды различных ЭВМ, во-вторых, конструктор, который настроил бы унифицированный интерпретатор на имитацию произвольной ЭВМ.

Общий взгляд на процесс интерпретации показывает, что практически невозможно полностью автоматизировать проектирование интерпретаторов для всего многообразия ЭВМ. Если теоретически и удастся придумать подобную систему, то вряд ли она окажется продуктивной.

Создание интерпретаторов имеет большое практическое значение. Одним из основных требований, предъявляемых к ним, является обеспечение большой скорости интерпретации программ целевой ЭВМ. Универсальные методы, как правило, снижают эффективность интерпретации. Универсальные интерпретирующие программы перегружены дополнительными, причем не касающимися целевой ЭВМ, интерпретирующими алгоритмами из-за того, что эти программы должны учитывать одновременно все многообразие принципов, привлекаемых для проектирования функциональных блоков различных ЭВМ.

Процесс проектирования вычислительных машин связан с многочисленными изменениями и корректировками при переходе от одной версии целевой ЭВМ к другой. Для настройки интерпретатора целевой ЭВМ на конкретную версию приходится проводить его соответствующее перепроектирование. Это требует обеспечения быстрой и легкой модификации программы-интерпретатора, чтобы не задерживать отладку программ целевой ЭВМ.

Модифицировать универсальные интерпретирующие программы трудно, так как они содержат много сведений о многообразии вычислительных функционально однородных структур различных ЭВМ. Это делает программу плохо обозримой, и найти в ней необходимые для модификации фрагменты нелегко.

Целесообразнее строить системы проектирования интерпретаторов для классов однотипных ЭВМ либо для развивающейся ЭВМ. Для построения таких систем предлагается модель системы схемотипирования модулей интерпретатора ЭВМ.

Пусть теперь $ЭВМ = \{ЭВМ_j\}$ обозначает множество однотипных ЭВМ или множество версий развивающейся ЭВМ.

Обозначим через P_j программу интерпретатора машины $ЭВМ_j$, функционирующую на инструментальной ЭВМ₀.

Разобьем программу на модули $\{m_i^j\}$ так, что $P_j = \bigcup_i m_i^j$.

Термин «модуль» используется для обозначения программы, состоящей из нескольких последовательных команд и имеющей свой собственный набор переменных $\{x_{i\alpha}^j\}$ и имя, по которому можно к ней обращаться.

Модуль m_i^j называется детерминированным, если всем переменным $x_{i\alpha}^j$ модуля присвоены значения.

Модуль m_i^j называется виртуальным, если значения некоторых переменных модуля не определены.

Пусть имеется множество таких детерминированных модулей $\{m_{i_0}^j\}$, что все модули будут с одинаковым набором переменных, а каждый модуль $m_{i_0}^j$ есть элемент из разбиения $\{m_i^j\}$ программы P_j .

Модули $\{m_{i_0}^j\}$ называются унифицированными, если они отличаются только значениями своих переменных.

Унифицированный модуль $m_{i_0}^j$, у которого не определены значения переменных, называется унифицированным виртуальным модулем.

Образование $S(\{x_{i_0\alpha}^j\}, m_{i_0}^j) \rightarrow \{m_{i_0}^j\}$ назовем схемотипированием. Схемотипирование отображает процесс конструирования унифицированных детерминированных модулей интерпретатора конкретной ЭВМ из унифицированных виртуальных модулей путем присваивания значений их переменным.

Проектирование модулей интерпретатора начинается на этапе общего проектирования на основании анализа всех функциональных блоков целевой ЭВМ, которые нужно интерпретировать. Для каждого функционального блока проектируется набор модулей, которые в совокупности интерпретируют работу блока. Объединение всех наборов интерпретирует работу целевой ЭВМ.

Модули проектируются детерминированными либо унифицированными виртуальными. Унифицированные виртуальные модули обладают функциональной индивидуальностью и включают однородные интерпретирующие алгоритмы. Унифицированные виртуальные модули представляют собой схем-программу, у которой набор переменных организован в виде незаполненной таблицы. Унифицированные модули проектируются на основе обобщения функций одинаковых функциональных узлов однотипных ЭВМ.

Структура унифицированных модулей позволяет автоматизировать процесс схемотипирования. Для этого разрабатывается метамашинный

язык, ориентированный на описание характеристик функциональных узлов машины, которые доопределяют унифицированные виртуальные модули до состояния интерпретации текущей версии целевой ЭВМ. Ниже будет описана технология изготовления интерпретатора.

Унифицированные виртуальные и детерминированные модули программируются на базовом языке и хранятся в архиве системы. Объединение унифицированных виртуальных и детерминированных модулей представляет собой виртуальный интерпретатор.

Схемопроектирование унифицированных детерминированных модулей для конкретной версии ЭВМ из унифицированных виртуальных модулей осуществляется после анализа характеристик функциональных узлов ЭВМ, которые затем описываются на метаязыке. Полученное в результате метаязыковое описание поступает в препроцессор, вырабатывающий значения переменных, которые требуются для заполнения таблиц унифицированных виртуальных модулей (для настройки их на интерпретацию текущей версии целевой ЭВМ). Затем препроцессор считывает из архива системы унифицированные виртуальные модули, присваивает им переменным соответствующие значения, заполняет таблицы, и записывает унифицированные детерминированные модули в архив системы. Скомпонованная программа из унифицированных детерминированных и неунифицированных детерминированных модулей транслируется с базового языка в коды инструментальной ЭВМ. Оттранслированная программа представляет собой инструментальную программу-интерпретатор целевой ЭВМ.

Техника схемопроектирования позволяет быстро и легко модифицировать унифицированные модули на стадии перепроектирования интерпретатора для новой версии целевой ЭВМ, для чего проводится перописание изменившихся характеристик функциональных узлов, интерпретируемых унифицированными модулями. По новому описанию путем схемопроектирования унифицированные виртуальные модули настраиваются на интерпретацию текущей версии.

В результате исследования и обобщения структур ЭВМ [1, 2] на функциональном уровне удалось разработать унифицированные модули, интерпретирующие дешифратор команд, дешифратор приводений, блок неуправляемых прерываний, блок управления памятью.

Унифицированный модуль, интерпретирующий дешифраторы команд и приводений, представляет собой схем-программный коммутатор процедур имитации команд и приводений операндов. Для настройки этой схем-программы на имитацию дешифраторов команд и приводений конкретной ЭВМ нужно заполнить таблицы коммутатора символическими именами команд в соответствии с кодировкой команд и тегов данных.

Унифицированный модуль, интерпретирующий неуправляемые прерывания, тоже выполнен в виде схем-программы со смесной таблицей текстов — сообщений о прерываниях. Настройка модуля на интерпретацию неуправляемых прерываний конкретной ЭВМ происходит путем заполнения смесной таблицы текстов сообщениями о прерываниях согласно номеру прерывания.

Память машины в унифицированной схем-программе интерпретируется ресурсами инструментальной ЭВМ, которые образуют хранилище информации, разбитое на страницы. Каждой странице присваивается порядковый номер, по которому она в дальнейшем идентифицируется. Обмен информацией организуется только между регистром процессора и хранилищем. В случае необходимости вся информация может браться с регистра процессора. Ее расположение в хранилище определяется адресом, в качестве которого используется исполнительный адрес имитируемой машины. Связь между исполнительным адресом и ресурсом, моделирующим страницу, соответствующую адресу, осуществляется через регистры приписки. Регистры приписки оформляются в виде таблиц. Входом в таблицу является номер сегмента, вычисленный по исполнительному адресу. В каждом элементе таблицы находятся атрибуты ре-

сурса, который моделирует сегмент, соответствующий номеру элемента таблицы. Строятся таблицы в зависимости от порции обмениваемой информации между регистром процессора и хранилищем, а также от количества привлекаемых для моделирования оперативных ресурсов инструментальной ЭВМ.

Для описания характеристик функциональных узлов целевой ЭВМ, интерпретируемых унифицированными модулями, позднее был разработан метамашинный язык ИМЯ. На языке ИМЯ описываются характеристики, определяющие атрибуты унифицированных модулей, для настройки их на интерпретацию целевой ЭВМ.

Программа на языке ИМЯ есть последовательность описаний дешифратора, памяти, приведенных и прерываний.

Описание языка ИМЯ. Запись вида {X} означает цепочку из одного или более возможных вхождений X.

Дешифратор {имя: код}. В описании дешифратора указывается символическое имя команды, которое должно совпадать с именем процедуры, интерпретирующей соответственно коду команду. По этому описанию унифицированный модуль «Дешифратор команд» настраивается на интерпретацию дешифратора целевой ЭВМ.

Приведение {имя: имя}. Описание приведенных делается для ЭВМ, в которые явно введены типы данных. В описании приведения указываются символическое имя команды, которая требует приведения операндов до процессорного вида перед ее исполнением, и символическое имя приведения. Имя приведения должно совпадать с именем процедуры, интерпретирующей приведение. По описанию строится коммутатор приведенных в унифицированном модуле приведенных.

Прерывание {номер прерывания: сообщение}. Описание определяет только неуправляемые прерывания. В описании задается таблица текстов сообщений о прерываниях для унифицированного модуля «Прерывание».

Память {размер слова: количество оперативных страниц}. В описании задается размер слова обмена информацией между регистром процессора и хранилищем информации в унифицированном модуле «Память». Размер слова указывается в словах инструментальной машины и определяет разбиение хранилища на моделируемые страницы.

В описании также указывается количество оперативных страниц инструментальной ЭВМ, используемых для моделирования памяти. Число оперативных страниц определяет количество регистров приписки в унифицированном модуле «Память».

ЛИТЕРАТУРА

1. Bell G. C., Newell A. Computer structures: readings and examples.— N. Y.: Mc. Craw-Hill Book Company, 1971.
2. Королев Л. Н. Структуры ЭВМ и их математическое обеспечение.— М.: Наука, 1978.
3. Марчук Г. И., Котов В. Е. Модульная асинхронная разрываемая система.— Новосибирск, 1978. (Препринт/АН СССР, Сиб. отд-ние, ВЦ; № 86, 87).

Поступила в редакцию 26 октября 1983 г.