

К ПЯТИДЕСЯТИЛЕТИЮ ЭВМ «СЕТУНЬ 70»

Рамиль Альварес Хосе, к.ф.-м.н.
Факультет ВМК МГУ им. М. В. Ломоносова
ramil@cs.msu.su

Владимирова Юлия Сергеевна, к.ф.-м.н.
Факультет ВМК МГУ им. М. В. Ломоносова
vladimirova@cs.msu.su
Москва, РФ

Первая в мире троичная ЭВМ «Сетунь» [1] была разработана в МГУ в 1959 г. и выпускалась серийно Казанским заводом математических машин с 1961 по 1965 г. ЭВМ «Сетунь» создавалась для решения задач средней сложности и предназначалась для использования в конструкторских бюро, высших учебных заведениях, научно-исследовательских институтах, заводах и т.п. Всего было произведено 50 машин, из которых 30 приобрели университеты и технические вузы, а остальные – промышленные и исследовательские организации. После снятия ЭВМ «Сетунь» с производства на основе опыта её создания и применения была разработана новая троичная ЭВМ «Сетунь 70» [2-4]. Её проектирование началось в 1967 г., а в апреле 1970 г. был завершён опытный образец. Архитектуру машины разрабатывали главный конструктор ЭВМ «Сетунь» Н.П. Брусенцов и главный разработчик программного оснащения ЭВМ «Сетунь» Е.А. Жоголев.

Ключевые слова: троичные ЭВМ, интерпретирующие программы, программное обеспечение, Н.П. Брусенцов, Е.А. Жоголев, ЭВМ «Сетунь 70».

Первоначальный вариант архитектуры ЭВМ «Сетунь 70». В ЭВМ «Сетунь 70» были сохранены основные принципы ЭВМ «Сетунь», в частности, то, что машина рассматривается как основа для программной надстройки. Вместе с тем ЭВМ «Сетунь 70» не была модернизацией ЭВМ «Сетунь», а представляла собой самостоятельную разработку, в которой были применены новые структурные и архитектурные решения.

Схемотехническая база была той же, что и в ЭВМ «Сетунь». Разница заключалась в том, что в ЭВМ «Сетунь 70» была применена однопроводная передача сигналов, в то время как в ЭВМ «Сетунь» для передачи троичного сигнала использовались два провода.

При создании машины ориентировались на повышение компактности представления программ и упрощение их трансляции с языков программирования на машинный язык. Для этого в качестве машинного языка была использована польская инверсная запись (ПОЛИЗ) [3, с. 4], ранее хорошо себя зарекомендовавшая в системе автоматического кодирования, реализованной на ЭВМ «Сетунь». Традиционно ПОЛИЗ используется при трансляции с языков программирования на язык машины.

ПОЛИЗ – это постфиксная бесскобочная форма записи математических выражений, в которой операции всегда следуют за операндами. Например, выражение $a * (b + c)$ в ПОЛИЗ записывается как:

$$a b c + *$$

Операнды заносятся в стек – память, устроенную по магазинному принципу, т.е. значения выдаются из стека в порядке, обратном тому, в котором они поступали. Операции изымают значения из стека, а результаты операций снова помещаются в стек. В отличие от обычной машинной памяти в стеке нет произвольного доступа к значениям – доступны вершина и подвершина. Значение из вершины не копируется, а изымается, вершиной при этом становится подвершина. Одноместная операция преобразует вершину – по-

требляет из неё значение, вычисляет результат и записывает новое значение в вершину. Двухместная операция потребляет два значения – из вершины и подвершины и записывает в вершину результат.

Процессор ЭВМ «Сетунь 70» может помимо стека обращаться к регистрам, и в дополнение к операциям над стеком в системе команд предусмотрены операции пересылок в оперативную память.

В машинном языке ЭВМ «Сетунь 70» не было понятия машинной команды, содержащей код операции и адреса операндов. Вместо команды использовалось понятие слога. Имеются два вида слогов: операционный слог, производящий действия в стеке, и адресный слог, осуществляющий пересылки значений в стек из памяти. Программа представляет собой последовательность слогов.

При создании системы операций опирались на опыт ЭВМ «Сетунь», в которой было всего 24 машинных команды. Архитектура ЭВМ «Сетунь» создавалась в расчёте на то, что основная часть всего необходимого функционала реализуется программно. Таким образом достигалась относительная простота архитектуры и невысокая цена машины. Например, действия над плавающей арифметикой в ЭВМ «Сетунь» осуществлялись интерпретирующей программой (ИП). ИП состояла из ведущей части и набора подпрограмм, располагавшихся на магнитном барабане. Вызов подпрограммы требовал нескольких обращений к магнитному барабану: сначала с него в оперативную память переписывались аргументы, затем – сама подпрограмма, после чего осуществлялся переход к её выполнению. Обмен между магнитным барабаном и оперативной памятью осуществлялся зонами по 54 девятиразрядных слова.

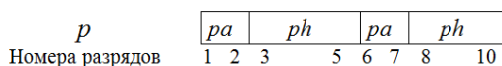
ЭВМ «Сетунь 70» так же была рассчитана на программную реализацию большей части необходимых функций. В её архитектуру было включено ПЗУ для хранения компонент программного оснащения, а для их вызова был предусмотрен набор из 27-ми макроопераций. Например, в опытном образце ЭВМ «Сетунь 70» макрооперации реализовывали подпрограммы операционной системы такие, как заказ на обмен с магнитным барабаном, приостановления работы программы до нажатия кнопки на пульте или возврата из подпрограммы [5]. Аппарат макроопераций позволял расширять машинный язык. Предполагалось также, что набор макроопераций может меняться заменой ПЗУ, таким образом специализация машины могла подстраиваться под определённую задачу.

В ЭВМ «Сетунь 70» было три типа операций: 27 основных, 27 служебных и макрооперации. Основные и служебные операции реализовывались аппаратно. Выполнение макрооперации представлял собой аппаратный переход к ведущей программной части, которая выбирает соответствующий адрес подпрограммы в ПЗУ и передаёт на него управление. При выполнении макроопераций допускались переходы к подпрограммам в оперативной памяти.

Машина могла работать в одном из трёх режимов: режиме пользователя, режиме прерываний или режиме макрооперации. Выполнение основных операций было разрешено в любом режиме, служебных – в режиме макроопераций и прерываний, макроопераций – только в основном режиме.

Режим прерываний необходим для обработки особых ситуаций таких как исчерпание страницы памяти, выполнение операции, запрещённой в текущем режиме и т.п., и для обеспечения прерывания выполнения основной программы по сигналу от внешних устройств. Механизм прерываний использовался для повышения эффективности работы машины за счёт обеспечения одновременной работы внутренних и периферийных устройств.

В режиме прерываний использовался системный стек, расположенный так же как и арифметический стек в одной из страниц оперативной памяти. Указатели на оба стека хранились в десятиразрядном регистре p , состоящем из двух пар полей:



где ph – номер страницы в оперативной памяти, в которой содержится стек, и pa – положение вершины стека. Разряды $p[1:5]$ хранили указатель на используемый в данный момент стек. При входе в режим прерывания аппаратно менялись местами значения полей: $p[1:2] \leftrightarrow p[6:7]$ и $p[3:5] \leftrightarrow p[8:10]$, а перед выходом из режима прерывания тот же обмен производился программно.

Описание архитектуры ЭВМ «Сетунь 70» в первоначальном варианте имеется в [2] на модифицированном Алголе-60.

Изменения в архитектуре ЭВМ «Сетунь 70». Машина создавалась в ограниченные сроки – существовало негласное обязательство завершить работу над машиной к апрелю 1970 г. Для того, чтобы можно было разрабатывать математическое обслуживание, не дожидаясь, когда заработает машина, на ЭВМ «Сетунь» была создана программная эмуляция нового процессора [6]. Машина «в металле» к положенному сроку была построена и выполняла все положенные тесты, но математическое обслуживание, под которое эта машина создавалось, так и не было реализовано.

В 1970 г. дальнейшая разработка машины была прекращена в административном порядке с формулировкой «разработка вычислительных машин не дело МГУ». Лаборатория была переселена вместе с опытным образцом ЭВМ «Сетунь 70» на чердак студенческого общежития МГУ [7, с. 181] и стала искать для себя другую сферу применения. Профессор МВТУ Б.В.Анисимов посоветовал заняться компьютерным обучением, о котором «никто не скажет, что оно не нужно» [7, с. 181]. Темой НИР стала «Разработка автоматизированной системы обучения на базе малой цифровой машины».

К началу работы над системой обучения почти полностью отсутствовало программное обеспечение для ЭВМ «Сетунь 70», и разработка велась с использованием только команд машинного языка. Составление и отладку программ затрудняла также страничная структура памяти. В это время стало известно о предложенной Э. Дейкстрой технологии структурированного программирования [8], ориентированной на сокращение трудоёмкости разработки программ, обеспечение легкости их понимания, верификации и модификации. Эту технологию стали применять на ЭВМ «Сетунь 70», но выяснилось, что управляющие команды машины плохо соответствовали операторам управления, используемым в структурированных программах, и машинная эффективность структурированных программ оказывалась низкой, что с ограниченными ресурсами ЭВМ «Сетунь 70» было неприемлемо.

Структурированное программирование обычно понимается как методика составления больших программ на высокоуровневых языках программирования. Суть её заключается в постепенной декомпозиции программ: выполняемое программой действие разбивается на несколько поддействий, пока подпрограммы не окажутся состоящими из нескольких команд языка программирования. В итоге программа принимает вид иерархической структуры подпрограмм, каждая из которых состоит из небольшого количества обособленных частей, и выполняет чётко определённую функцию, что и обеспечивает их понятность, простоту отладки и модификации. Указанный способ программирования называется нисходящим («top-down»). Возможен и противоположный ему восходящий способ («bottom-up»), который используется, в частности, для развития программных систем и при отладке программ.

Достоинства структурированного программирования достигаются некоторым снижением эффективности программ: структурированная программа длиннее обычной и выполняется дольше. В большинстве случаев такое снижение эффективности оправданно повышением качества программы в течение всего её жизненного цикла.

Для обеспечения возможности создания на машинном языке структурированных программ в 1975 г. была перестроена архитектура машины. Выяснилось, что стековая архитектура машины хорошо согласовывается с принципами структурированного программирования: его внедрение на уровне машинного языка достигалось небольшими измене-

ниями – добавлением системного стека и команд ветвления, цикла и перехода к подпрограмме [9].

В машине изначально имелся аппарат упрятывания и восстановления значения программного счётчика, рассчитанный на три состояния: счётчик программы, счётчик макрооперации и счётчик обработки прерывания. Он был преобразован в стек адресов возврата, содержащий 26 позиций, что обеспечивает значительную глубину вложенности подпрограмм.

Вместо не нашедших применения в программах операций $E=0$, $E+1$, $E-1$ и $T=C$ были введены структурированные операции: BRT, DOW, JSR и неструктурированная операция JMP. Эти операции позволили в сочетании с адресными слогами конструировать команды:

JSR P – команда выполнения подпрограммы P;

DOW P – команда повторения подпрограммы P, пока значение в вершине стека не равно нулю;

BRT P1 P2 P3 – вызов одной из трёх подпрограмм в зависимости от знака числа в вершине арифметического стека;

JMP A – неструктурированная команда обобщенного перехода на адрес, указанный в двух следующих за операцией слогах.

Таким образом, принятая первоначально постфиксная форма записи программы оказалась нарушенной введением префиксных операций: новые операционные слоги используют следующие за ними в программе слоги как адреса вызываемых ими подпрограмм.

Для структурированных операций адреса подпрограмм задаются одним слогом в формате:



Подпрограммы находятся в оперативной памяти, ch – адрес страницы, ca – адрес подпрограммы на странице. В неструктурированной операции JMP адрес перехода задается в двух слогах в другом формате.

Вызываемая подпрограмма после своего выполнения должна возвращать управление команде, непосредственно следующей за последним адресным слогом команды: P или P3. Поэтому подпрограмма должна завершаться операцией возврата из подпрограммы RMC, которая восстанавливает программный счётчик значением вершины стека возврата. Операция RMC относится к служебным, она не доступна в основном режиме, но может вызываться в режиме макроопераций. Для того, чтобы обеспечить возможность использования слогов BRT, DOW и JSR основным режимом работы машины сделали макроопераций, а режим пользователя, который ранее был основным фактически перестал использоваться. Стал также доступен вызов макрооперации в режиме макроопераций.

Следует отметить ещё одно отличие структурированных операций ЭВМ «Сетунь 70» от аналогичных операторов в языках программирования высокого уровня. Оно заключается в том, что в ЭВМ «Сетунь 70» тела процедур заменены ссылками на подпрограммы. Такая структура программ делает понятнее их логику. Кроме того, она позволила преодолеть трудности, связанные с отладкой и выполнением программы в условиях страничной памяти: структурированные операции разбивали программы на сравнительно небольшие части, каждую из которых можно было отлаживать и размещать в памяти независимо от других.

Система команд ЭВМ «Сетунь 70» после перестройки приведена в [4].

В дальнейшем указанные особенности архитектуры ЭВМ «Сетунь 70» послужили основой диалоговой системы структурированного программирования ДССП, реализованной на машинах серии ДВК и на персональных компьютерах [10].

Для ЭВМ «Сетунь 70» было реализовано базовое программное обеспечение: операционная система, выполняющая функции загрузчика, отладчика и монитора, организацию обмена с магнитным барабаном и осуществление макроопераций, тесты, предназначенные для проверки работы машины, однопроходный ассемблер с входным языком структурированного программирования; редактор текстов; библиотека сервисных программ. Единственным практическим применением ЭВМ «Сетунь 70» стала система обучения «Наставник» [7]. Для её функционирования к машине был подключен класс с 27 терминалами учащихся. Впервые экспериментальный образец «Наставника» был применен в марте 1974 г. – профессор Н.С. Бахвалов подготовил и провел автоматизированный коллоквиум по курсу «Численные методы» для студентов 3-го курса факультета ВМК МГУ. Выяснилось, что «Наставник» позволяет существенно повысить эффективность работы преподавателя: в течение одного рабочего дня при помощи системы можно аттестовать до 150 студентов. Проверка показала, что система обеспечивает высокую степень достоверности оценок. В течение нескольких лет «Наставник» использовался на ВМК МГУ для обучения языку Фортран.

Одним из применений «Наставника» стала система проведения автоматической диагностики цветового зрения, реализованная совместно с кафедрой психофизиологии психологического факультета МГУ под руководством профессора Е.Н. Соколова [11]. Для создания этой системы аппаратура «Наставника» была доукомплектована цветным телевизором, управляемым компьютером. Система демонстрировала испытуемому в течение получаса два световых сигнала различного цвета, которые он оценивал по девятибалльной шкале различия. Для ввода оценок использовалась клавиатура терминала. Обработка полученных данных позволяла построить объективные характеристики цветового зрения каждого из испытуемых.

Опытный образец ЭВМ «Сетунь 70» функционировал в составе автоматизированной системы обучения «Наставник» на факультете ВМК МГУ до замещения его микрокомпьютером «Электроника НЦ 80–20» (ДВК 2) в 1986 г., после чего был передан в Политехнический музей.



Рис. ЭВМ «Сетунь-70»

ИСТОЧНИКИ И ЛИТЕРАТУРА

1. Брусенцов Н.П., Маслов С.П., Розин В.П., Тишулина А.М. Малая цифровая вычислительная машина «Сетунь». – М.: Изд-во Моск. ун-та, 1965.
2. Брусенцов Н.П., Жоголев Е.А. Структура и алгоритм функционирования малой вычислительной машины. // Вычислительная техника и вопросы кибернетики. Вып. 8. – Л.: Изд-во Ленингр. ун-та, 1971. С. 34-51.
3. Брусенцов Н.П., Жоголев Е.А., Маслов С.П. Общая характеристика малой цифровой машины «Сетунь-70» // Вычислительная техника и вопросы кибернетики. Вып. 10. – Л.: Изд-во Ленингр. ун-та, 1974. С. 3-21.
4. Брусенцов Н.П., Рамиль Альварес Х. Троичная ЭВМ «Сетунь 70» // Труды SORUCOM-2011. Вторая Международная конференция «Развитие вычислительной техники и её программного обеспечения в России и странах бывшего СССР». – Великий Новгород, 2011. С. 71-75.
5. Рамиль Альварес Х., Киселев Г.А., Галимов Ю.Ю. Операционная система малой ЦМ «Сетунь 70» // Вычислительная техника и вопросы кибернетики, Л.: – Изд. ЛГУ, т. 16, 1979. С. 20-30.
6. Рамиль Альварес Х. Программная интерпретация структуры и системы команд малой ЭВМ // Вычислительная техника и вопросы кибернетики. Вып. 9 – Изд-во МГУ. М., 1972. С. 49–58.
7. Брусенцов Н.П., Маслов С.П., Рамиль Альварес Х. Микрокомпьютерная система обучения «Наставник». – М.: «Наука», 1990.
8. Dijkstra E.W. Notes on structured programming. EWD 249 – Technical University, Eindhoven, Netherland, 1969..
9. Брусенцов Н.П., Рамиль Альварес Х. Структурированное программирование на малой цифровой машине. // Вычислительная техника и вопросы кибернетики. Вып. 15. – М.: Изд-во Моск. ун-та, 1978. С. 3-8.
10. Брусенцов Н.П., Захаров В.Б., Руднев И.А., Сидоров С.А., Чанышев Н.А. Развиваемый адаптивный язык РАЯ диалоговой системы программирования ДССП. Учебное пособие. – М.: Издательство Моск. ун-та, 1987.
11. Брусенцов Н.П., Соколов Е.Н., Зимачев М.М., Измайлов Ч.А., Маслов С.П., Рамиль Альварес Х. Автоматизированная диагностика цветового зрения. // Психологический журнал, т.1, № 3, 1980. С. 58-84.

Опубликовано:

Рамиль А. Х., Владимирова Ю. С. К пятидесятилетию ЭВМ Сетунь 70 // Труды SORUCOM-2020 5-ая международная конференция Развитие вычислительной техники в России, странах бывшего СССР и СЭВ: история и перспективы https://www.sorucum.org/books/SORUCOM-2020_RU-final.pdf. — 2020. — С. 261–265.