

УВК «Самсон» – базовая ЭВМ РВСН

1. Введение

Коллектив сотрудников лаборатории системного программирования ВЦ ЛГУ всегда состоял только из математиков – выпускников мат-меха ЛГУ. Но именно в этой лаборатории была разработана новая ЭВМ, не являющаяся копией какой-то иностранной ЭВМ, хотя это было очень принято в СССР в те годы. Разумеется, после того, как все было придумано и от моделировано, мы пригласили группу инженеров для её воплощения «в железе», но не было ни одного случая, когда какое-нибудь решение было принято инженерами против воли математиков – это была главная идея проекта.

Создание и особенно внедрение в производство новой оригинальной ЭВМ – большая и трудная задача. Автору доклада не хотелось бы создать впечатление «чертика из табакерки», поэтому я потрачу 2-3 страницы на предысторию, чтобы показать корни проекта, как он развивался, возможно, на конференции по истории ЭВМ это и есть самое интересное.

Кроме того, сразу оговорюсь, что в разработке «Самсона» принимало участие много людей, а во внедрении – и из многих организаций. Возможно, у них несколько иной взгляд на историю «Самсона», но, поскольку я был руководителем лаборатории, где всё начиналось, рискну высказать свою личную точку зрения.

2. Трансляторы

Я закончил мат-мех ЛГУ с отличием в 1971 году (первым защитил диплом в первом выпуске кафедры матобеспечения ЭВМ). Моим научным руководителем был доктор физ.-мат. наук Григорий Самуилович Цейтин – сильный и разносторонний ученый. В то время он руководил лабораторией матлингвистики НИИММ ЛГУ, достиг выдающихся успехов в матлогике и теории сложности вычислений, но основным его увлечением было программирование. Именно он увлек группу студентов идеей реализации Алгола 68 – новейшего в ту пору алгоритмического языка высокого уровня [1]. Кроме того, под руководством Г.С. Цейтина в начале 70-х гг. мы реализовали один из первых в стране диалоговых редакторов текста `disco` [2].

В 1974 году я познакомился с Б.А.Бабаяном, под фактическим руководством которого создавалась новая ЭВМ «Эльбрус». Нам поручили разработку интерпретатора Автокода «Эльбрус» на ЕС ЭВМ, ответственным исполнителем был Н.Ф. Фоминых [3,4]. Эта работа получила определенную известность. Поскольку выпуск «Эльбруса» задержался на несколько лет, нашим интерпретатором пользовались десятки коллективов по всему СССР. Некоторые важные заказчики расплачивались с нами фондами на ЕС ЭВМ – в те времена серьезная техника не продавалась свободно, а распределялась по фондам. Платить сотруднику больше полутора окладов было невозможно, поэтому нашим единственным интересом были те самые фонды. Мы говорили, что Коля Фоминых просидит где-то в глуши лесов, устанавливая интерпретатор и обучая как им пользоваться, а с ним расплатятся "борзыми щенками". Таким был наш первый опыт предпринимательства. Разумеется, ЭВМ получал ВЦ ЛГУ, но мы сумели договориться с руководством ВЦ, что "нашими" ЭВМ пользуемся только мы.

С 1972 года заведовать кафедрой матобеспечения ЭВМ в ЛГУ стал чл.-корр. АН СССР Святослав Сергеевич Лавров. Под его руководством в 1975 году мы начали еще более крупную работу по программному обеспечению «Эльбруса» (отладочные средства и несколько десятков пакетов прикладных программ), я в это время уже руководил группой в лаборатории системного программирования ВЦ ЛГУ, наша группа совмещала работы по «Эльбрусу» и Алголу 68.

В 1976 году С.С. Лавров был избран на пост директора Института Теоретической Астрономии АН СССР, его активность в ЛГУ заметно снизилась, в конце концов, он оставил за собой только кафедру матобеспечения ЭВМ, а я был назначен и.о. руководителя большого отдела ВЦ, который был создан специально под работы по «Эльбрусу». Эти работы мы успешно завершили. В дальнейшем наш коллектив от работ по «Эльбрусу» отошел, а С.С. Лавров продолжил со своими студентами, но по другим темам, например, трансляторы с АБВ и Паскаля.

К концу 70х годов работы по транслятору с Алгола 68 для ЕС ЭВМ вышли на заключительную фазу – комплексная отладка на ЕС ЭВМ [5]. Работа приобрела практический, производственный характер, в этот момент Г.С. Цейтин в значительной мере потерял к ней интерес. В течение 2-3х лет мы еще занимались под его руководством изучением и реализацией языков Искусственного Интеллекта и моделированием действий роботов, но большого интереса эти работы в нашем коллективе не вызвали.

В этот период нашим фактическим научным руководителем стал академик Андрей Петрович Ершов. Он возглавил рабочую группу по Алголу 68 ГКНТ СССР, я был назначен его заместителем. Мы собирались 4-5 раз в год в разных городах СССР, обсуждали проблемы реализации Алгола 68 для разных ЭВМ (в том числе, и для «Эльбруса»). Одновременно разрабатывалось несколько трансляторов с Алгола 68 под руководством Е.Л.Ющенко, А.Н.Маслова, М.Р.Левинсона, А.Н.Терехова и А.П.Ершова.

Хотя полным успехом завершились только работы под руководством А.Н.Маслова (для «Эльбруса») и А.Н.Терехова (для ЕС ЭВМ), все остальные работы также привнесли много новых идей и дали толчок дальнейшим исследованиям. Отдельного упоминания заслуживает русский перевод Официального Сообщения об Алголе 68. В процессе работы под этим переводом в рабочей группе по Алголу 68 были решены и обоснованы многие важные проблемы терминологии, введены в оборот многие понятия программной инженерии.

Осенью 1976 года я в течение трёх месяцев читал лекции в Будапеште и привёз оттуда первые материалы по языку Ада и магнитную ленту с венгерской реализацией языка Пролог. Пролог мы изучили, выполнили несколько экспериментов, но нам декларативный способ не понравился, по крайней мере, на Прологе. Помню, как долго мы возились с задачей сборки деталей пирамидки из разбросанных на столе кружков разного диаметра. Сейчас я понимаю, что декларативный способ может быть с успехом использован, но только для определенного класса задач.

С языком Ада история интереснее. Мы ожидали, что Аду ждёт успех, поскольку это базовый язык Министерства обороны США. Я решил заняться реализацией языка Ада, используя накопленный опыт реализации Алгол 68. Но тут один из сотрудников нашей лаборатории Илья Гиндыш попросил меня отдать эту работу ему. Он на два года старше меня и ему также хотелось попробовать руководить темой. Я с некоторым сожалением согласился. Но уже через полгода руководителем этой темы в нашей лаборатории стал мой молодой выпускник Аркадий Попов. Так я в первый раз понял, что "полки не дают, полки берут". Аркадий Попов решил сделать всё "по науке", а не так, как мы работали над Алголом 68 (мы больше следовали технике быстрого прототипирования). Два года эта группа разрабатывала подробные спецификации, написали два или три толстых тома, но в первые же месяцы реализации натолкнулись на ошибку в проекте, 2 года работы пропали зря. Нет не зря! Мы получили бесценный личный опыт. В конце концов, эта группа завершила реализацию Ады и её окружения на ЕС ЭВМ, а затем и на ПЭВМ. Как и наш транслятор с Алгола 68, реализация языка Ада стала первой в СССР [6].

3. Работа с промышленностью

В 1980 году оборонный отдел Обкома КПСС Ленинграда обратился к ректору ЛГУ В.Б.Алесковскому с просьбой о помощи, а он поручил мне курировать разработки в области ПО для военных заказов. Мне было сказано, что в оборонной промышленности столкнулись с массой проблем при создании ПО систем оборонного назначения, поэтому решено, что ЛГУ в лице нашей лаборатории должен помочь. Моего согласия никто не спрашивал, да в те времена этого и не требовалось. Так мы начали работать с ЛНПО "Красная Заря", "Импульс", "Морфизприбор", "Ленинец", "Аврора", "Гранит" – ведущими предприятиями Ленинграда, работающими в интересах различных родов войск и ведомств. С большинством этих предприятий мы работаем до сих пор.

Первыми из военных организаций к нам обратились сотрудники ЛНПО "Красная Заря" с просьбой помочь в программировании широкого класса задач управления и связи, в частности, создания функционального программного обеспечения (ФПО) телефонных станций, управляемых специализированными ЭВМ (СЭВМ). Несколько лет ушло на изучение предметной области, пробные реализации, решение организационных вопросов.

В это время мы свято верили, что, используя хороший АЯВУ (алгоритмический язык высокого уровня), можно добиться существенного (в разы) повышения производительности труда. Никаких сомнений, что лучший в мире АЯВУ - это нами любимый Алгол 68, у нас не было.

Небольшой набор базовых операторов, каждый из которых обладает ясной и точно определённой семантикой, при этом достаточно выразителен и нагляден, плюс мощные правила суперпозиции, позволяющие строить сколь угодно сложные конструкции из простых (признаемся, иногда это превращается в недостаток, когда кто-то слишком умный хочет обязательно воспользоваться всей мощностью языка) дают полное право Алголу 68 считаться языком достаточно высокого уровня. Полный видовой контроль периода компиляции обеспечивает высокую защищённость пользователя от ошибок и одновременно обеспечивает потенциальную возможность генерации эффективного объектного кода. Предоставляемые языком возможности описания новых видов (типов) и операций (а также переопределения стандартных операций) в сочетании с возможностью раздельной трансляции и иерархического накопления контекстов дают возможность определять специализированные языки, не выходя за рамки Алгола 68, для которых не надо реализовывать новые трансляторы.

Многие пользователи годами писали программы на кем-то подготовленных для них специализированных языках, не подозревая, что они пишут на Алголе 68 и пользуются одним и тем же транслятором.

Мы начали внедрять Алгол 68 в различные организации. Лучше всего получалось внедрение в военные организации, для которых требования надёжности превалировали над всеми остальными требованиями. Военные для своих задач чаще всего применяли не стандартные ЕС, СМ и персональные ЭВМ, для которых у нас трансляторы с Алгола 68 уже были реализованы, а специализированные ЭВМ (СЭВМ). Пришлось в массовом порядке заняться реализацией кросс-трансляторов.

Очень быстро мы осознали, что все СЭВМ обладают одной общей чертой – они удовлетворяют самым жёстким инженерным требованиям, но программировать для них, а, тем более, создавать для них трансляторы, очень тяжело. Видимо, создатели этих СЭВМ не задумывались о проблемах программирования.

4. HLL-компьютер «Самсон»

Мы решили создать свою ЭВМ, следуя принципу «спасение утопающих - дело рук самих утопающих». Эта идея созревала очень долго.

Во-первых, создавая ПО для «Эльбруса», мы прониклись идеей HLL- компьютеров (компьютеров, ориентированных на поддержку основных операторов АЯВУ). Мы любили отечественную ЭВМ «Эльбрус» и не любили «цельносозданные» ЕС ЭВМ И СМ ЭВМ. Но нам не нравилось, что «Эльбрус» очень большой и требует водяного охлаждения, у этой ЭВМ главной задачей было обеспечение сверхвысокого быстродействия. За то, что мы сделали большой комплекс ПО для «Эльбруса», нам дали фонды на эту ЭВМ для мат-меха, который как раз в это время (1979 год) переехал в Петергоф. Мы построили зал площадью 500 квадратных метров, градирню (специальный бассейн для очистки воды), получили первые ящики с оборудованием, но тут университетские инженеры буквально взбунтовались. Они несколько раз съездили на завод-изготовитель, познакомились с правилами эксплуатации и заявили, что в университетских условиях обеспечить все необходимые требования невозможно. С огромным сожалением от идеи установки «Эльбруса» на мат-мехе пришлось отказаться.

Во-вторых, следуя предложенной Н.Виртом идее Р-кода, которую он применял для переноса транслятора с языка Паскаль на разные ЭВМ (то, что у него была и аппаратная реализация в виде ЭВМ Lilith, мы тогда не знали), мы решили разработать полноценную архитектуру виртуальной ЭВМ. Тогда появились первые однобайтовые микропроцессоры 6502 и i8080 и первые персональные ЭВМ. Мы реализовали для них интерпретаторы нашей виртуальной ЭВМ и транслятор с Алгола 68 в её коды (Н.Н. Вояковская, Б.А. Федотов), получив возможность удобной работы в 16-разрядной архитектуре [7]. При замедлении в 4-5 раз мы в 2-3 раза выигрывали в длине объектного кода (за счёт более высокого уровня системы команд), что тогда было очень важно. Чаще всего, первые персональные ЭВМ имели только 64 кб оперативной памяти, но микропроцессоры уже были довольно быстрыми.

Итак, мы решили создать HLL-компьютер, пусть он не будет такой быстрый, как «Эльбрус», но он должен быть маленьким, без водяного охлаждения, но, всё-таки, с приличной производительностью. К этому моменту мы изучили несколько зарубежных HLL-компьютеров, знали мы, разумеется, и архитектуру советского HLL-компьютера «Мир», разработанного под руководством В.М. Глушкова в Киеве. Оказалось, что обычно HLL-компьютер примерно в 2 раза дороже сопоставимого с ним по скорости традиционного компьютера. Хотелось как-то решить эту проблему.

Вначале мы решили сделать специализированный HLL-компьютер. В это время мы уже работали с ЛНПО «Красная Заря», хорошо знали УК1010, СУБК СС, СУБК СМ, «Нева» и другие СЭВМ, ориентированные на телекоммуникационные задачи. Но из идеи специализированного для телекоммуникационных задач HLL-компьютера ничего не вышло. Для установления соединения нужны логические команды, работа с битовыми шкалами, арифметика целых чисел. Для техобслуживания нужна работа со строками, файлы, диалог с оператором, сетевые возможности и т.д. Таким образом, нам нужна именно универсальная ЭВМ, несмотря на то, что она применяется в достаточно узкой предметной области.

Пару лет мы пытались как-то выйти из этой ситуации: универсальные ЭВМ дороги и громоздки, СЭВМ не могут решить задачу полностью. Я знаю АТС, в которой три разных ЭВМ с разными ОС управляют собственно телефонией, техобслуживанием и управлением сетью соответственно. Представляете, сколько нужно офицеров, чтобы обеспечить сопровождение этой АТС?

Решение этой задачи было подсказано нашим опытом разработки трансляторов. Сначала мы заметили, что существенную часть любой ЭВМ составляют схемы контроля, более того, мы видели ситуации, когда ЭВМ не работала именно из-за ошибок в схеме контроля. Отключаешь схему контроля – и всё в порядке! Потом мы заметили, что многие проверки, выполняемые аппаратурой, повторяют аналогичные проверки, которые перед этим делал транслятор с АЯВУ. Поскольку мы разрабатывали и архитектуру ЭВМ, и её системное ПО, мы решили, что подобная избыточность нам не нужна. Мы решили создать ЭВМ, для которой в принципе нельзя составлять программы на ассемблере, тем более, в кодах. Наша технологическая цепочка всегда включает трансляцию с АЯВУ, поэтому то, что проверил транслятор, аппаратура должна принимать "на веру". Зачем нужна проверка на неправильный код операции, неправильный номер регистра или некорректную адресацию, если транслятор такого не генерирует?

Транслятор и операционная система гарантируют, что если на какой-то сегмент памяти есть ссылка (в виде физического адреса) из стека адресов, то этот сегмент не может быть переписан в какое-то другое место в памяти. Поэтому возможна очень эффективная реализация виртуальной памяти. Транслятор может точно подсчитать аппетит каждой процедуры на регистры в стеке, поэтому не нужны аппаратные проверки на переполнение или исчезновение стека. Таких примеров, когда транслятор помогает упростить аппаратуру, нашлось много. Но и хорошо спроектированная аппаратура может облегчить реализацию транслятора. HLL-компьютеры содержат не только простые команды "сложить", "вычесть", "перейти", но и такие сложные, как "вызов процедуры", "цикл", "вырезка элемента массива", что упрощает транслятор.

Очень важна ортогональность системы команд, например, в ЕС ЭВМ нам очень досаждали такие случаи: сложение, вычитание и умножение полуслов есть, а деления полуслов – нет; сложение и вычитание слов требуют одного регистра, а умножение и деление слов – двух и так далее. При проектировании «Самсона» мы стремились избежать таких глупостей, хотя иногда приходилось тратить лишний код операции для очень редко используемой команды, чтобы сохранить ортогональность.

Оказалось, что гарантировать корректную работу аппаратуры мы можем только для статических АЯВУ с полным контролем типов в период компиляции таких, как Алгол 68, Паскаль, Модула 2, Ада, но не таких, как С и PL/I. На этих принципах нам удалось создать компактную ЭВМ, мы дали ей имя "Самсон" [8], но не в честь сильного и легковверного библейского героя, а в честь самого большого фонтана у нас в Петродворце. Мы получили 3 авторских свидетельства (общая организация, совмещение регистров со стеком, организация виртуальной памяти), решили несколько интересных научных задач, привлекли в лабораторию много хороших студентов. Прежде чем был спаян хоть один провод, мы долго экспериментировали на моделях, отражающих потактовую структуру ЭВМ. Многие команды были включены в состав архитектуры именно в результате моделирования.

Реализацией потактного симулятора занималась моя дочь Карина Терехова – тогда студентка младших курсов мат-меха. Мы с ней прогоняли сотни тестов, исследуя статическую и динамическую статистику использования разных команд и определяя, сколько тактов занимает исполнение того или иного теста. После каждого прогона теста мы брали только верхние по частоте 20-25 команд. Штук 5 команд исполнялись миллионы раз, следующие по частоте 10 команд – сотни тысяч раз, еще десяток – десятки тысяч раз, а остальные – сотни или тысячи раз, что для нас было неинтересно.

Верхнюю строку всегда занимала команда «читать из памяти в стек целых» (40%). Нам удалось сделать однобайтовую команду, которая читала на стек одну из первых 15 переменных процедуры. Оказалось, что практически всегда только эта команда и

используется, поскольку редко встречаются процедуры с большим числом локальных переменных.

Каждый год на курсе «Архитектура ЭВМ» я рассказывал студентам, как, сидя на колхозном поле, обрубая хвостики морковки, я уговорил Колю Фоминых подумать, как сделать более эффективную реализацию этой команды. Коля полдня отнекивался, а потом все-таки придумал, как вместе 4 тактов сделать 3. Только за счет этого мы повысили производительность «Самсона» на 10%!

Команда «условный переход по ложному условию» занимала 10% всех исполняемых команд (понятно почему – if, while), а команда « \Leftarrow » – 8%. Мы их склеили (получив команду «переход по равно») и выиграли 1 байт длины кода и 2 такта исполнения. Таких примеров можно привести множество. Само собой разумеется, из целей ортогональности и облегчения трансляции мы ввели команды «переход» по всем шести типам сравнений, хотя остальные команды использовались не так часто.

Мы объездили несколько заводов в СССР с предложением наладить выпуск нашей машины. Но везде натолкнулись на отказ по той причине, что у нас не было западного прототипа. Такие тогда были времена. Нас выручил завод "Оргтехника" в г. Пловдиве (Болгария), который был побратимом Ленинграда. Все работы были оформлены как подарок к 70-летию советской власти, успешная демонстрация работы новой ЭВМ членам Политбюро БКП и нашим партийным начальникам состоялась 5 ноября 1987 года в 14 часов. В 12 часов "Самсон" еще не работал, мы не спали двое суток, под "горячую руку" попал даже наш ректор В.Б.Алесковский, о чем он со смехом напомнил мне через много лет.

Болгары выпустили 100 экземпляров "Самсона", только после этого нам удалось убедить нашего основного заказчика в СССР – Управление правительственной связи КГБ. Для этого пришлось дополнительно разработать троированную архитектуру с очень высокой надёжностью. Но система команд была точно такая же, поэтому болгарский "Самсон" использовался как инструментальная ЭВМ.

Собственно, идея троированного компьютера (три процессора, три памяти, три входных мажоритара и три выходных) принадлежит не нам. Нам была известна троированная ЭВМ «Микрон», созданная в НИИ АА под руководством академика В.С. Семенихина. Мы использовали такую же структурную схему, но вся реализация была своя.

Очень важной идеей, которую мы реализовали в "Самсоне", было динамическое микропрограммирование. Н.Ф.Фоминых предложил удобную технологию микропрограммирования на Алголе 68 [9], которая в десятки раз упростила эту сложную задачу. Для стандартной системы команд "Самсона" мы занимали не более двух третей микропамяти, оставшуюся треть пользователи занимали для команд, придуманных специально для их приложений. Во многих случаях это позволяло ускорить критичные по времени фрагменты программы в 10-20 раз.

Возможность микропрограммирования дополнительных команд была известна давно, например, все ЕС ЭВМ Ряда 2 этой возможностью обладали. Но никто не пользовался этой возможностью по причине высокой сложности работы. Мы превратили эту работу в деятельность, доступную даже студентам.

Первой реализацией троированного управляющего вычислительного комплекса (УВК) «Самсон» была центральная ЭВМ АМТС «Фобос-К» Управления правительственной связи КГБ [10]. К сожалению, эта АМТС по причинам, весьма далеким от науки и техники, в серию не пошла. Тогда, воспользовавшись связями в Оборонном отделе Обкома КПСС, я договорился с ЛНПО «Импульс» о выпуске этого УВК.

Сотрудники «Импульса» (особо надо упомянуть В.П. Сامةцкого и В.М. Зуева) переработали конструктив и отдельные элементы «Самсона» под свои требования, мы доработали инструментальные средства разработки. В 1992 году министр обороны России Грачев подписал приказ о принятии УВК «Самсон» на вооружение РВСН. Долгие годы это все было большим секретом, но в 1999 году РВСН праздновали свое 40-летие, к этому торжественному моменту была выпущена красочная толстая брошюра на мелованной бумаге. Первые 14 страниц посвящены УВК «Самсон», причем на 10-ой странице есть фраза «Архитектура и ОС УВК «Самсон» разработаны при помощи ГУП «Терком» и ЛГУ». Сначала эта фраза меня покорила, но затем я осознал, что брошюра не содержит грифа, так что секрета больше нет. Теперь в курсе «Архитектура ЭВМ» я рассказываю студентам про HLL-компьютеры на примере УВК «Самсон». Я не думаю, что кто-то из моих выпускников будет на нем программировать, здесь главный интерес – как и почему всё это делалось. Часто перед рассказом какого-то раздела я спрашиваю у студентов, как бы они решили какую-то задачу построения архитектуры и иногда даже получаю ответы.

Сейчас по заказу ФГУП «НИИ автоматики» (г. Москва) мы разрабатываем новый отказоустойчивый вычислительный комплекс (уже прошли испытания макета).

Насколько мы знаем, "Самсон", принятый на вооружение в 1992 году, до сих пор активно используется во многих важных системах.

Литература

1. Алгол 68. Методы реализации. Под ред. Г.С. Цейтина. – Л.: Изд-во Ленинградского университета, 1976 г., 224с.
2. С.Н. Баранов, А.Н.Терехов, Г.С. Цейтин «Инструкции к программе DICO», Методические материалы по программному обеспечению ЭВМ. Серия 4. Выпуск 5. Изд. ЛГУ, 1974 г.
3. Н.Ф. Фоминых "Интерпретатор автокода МВК "Эльбрус" для ЕС ЭВМ", диссертация на соискание ученой степени кандидата физико-математических наук, Ленинград, 1987 г.
4. Гиндыш И.Б., Попов А.П., Рухлин А.П., Терехов А.Н., Фоминых Н.Ф., Швецова Г.А. "Интерпретатор АК ЛГУ для ЕС ЭВМ", изд. ЛГУ, 1981 г.
5. Дейкало Г.Ф., Новиков Б.А., Рухлин А.П., Терехов А.Н. "Новые средства программирования для ЕС ЭВМ. Транслятор с языка Алгол 68 и диалоговая система JEC", "Финансы и статистика", М., 1984г.
6. Попов А. П. "Методы реализации языка Ада", диссертация на соискание ученой степени кандидата физико-математических наук, Ленинград, 1987 г.
7. Матиясевич Ю.В., Терехов А.Н., Федотов Б.А. "Унификация программного обеспечения микроЭВМ на базе виртуальной машины", "Автоматика и телемеханика", N5, М., 1990 г.
8. Евстюнин М.В., Кожокарь С.К., Терехов А.Н., Уфнарковский В.А. "Как Паскаль и Оберон попадают на "Самсон", Монография по технике программирования. Кишинев, "Штиинца", 1992 г.
9. Н.Ф.Фоминых, Использование стандартного расширяемого алгоритмического языка в микропрограммировании. В сб.: Диалоговые микрокомпьютерные системы. М., МГУ, 1986 г.
10. С.А. Валдаев «Связь, которая не подведет», М., Славянский диалог, 2001 г.