

ЗАМЕТКИ О ТРОИЧНОЙ ЦИФРОВОЙ ТЕХНИКЕ - ЧАСТЬ 2

Н. П. Брусенцов

В предыдущих выпусках наших заметок были обсуждены логические и инженерные аспекты реализации троичных цифровых устройств. При этом было показано, что естественным и весьма эффективным путем физического воплощения трехзначной логики является использование пороговых элементов с положительными и отрицательными весами входов, а также с сигналами положительной и отрицательной полярности. Данная техника была разработана при создании троичных машин "Сетунь" и "Сетунь 70" [1] и отлично показала себя на всех этапах конструирования, производства и эксплуатации аппаратуры. Пороговое осуществление операций с положительными и отрицательными сигналами легко осваивается людьми, поэтому как понимание логики построенных из пороговых элементов троичных устройств, так и разработка устройств, реализующих заданные функции трехзначных переменных, не связаны с особыми затруднениями. Что же касается надежности функционирования, то, вопреки беспочвенному утверждению [2, с.28], будто "плохая надежность аппаратуры и неадекватное программное оснащение" воспрепятствовали продолжительному использованию "Сетуни", следует сказать, что первая троичная машина была значительно надежнее большинства ее двоичных сверстниц и имела добротное и не такое уж бедное для минимашин того времени программное оснащение [3].

Экспериментальный образец машины "Сетунь", построенный в 1957-1958 гг., находился в эксплуатации 15 лет, причем из 4 тыс. использованных в нем пороговых элементов типа быстродействующих магнитных усилителей с питанием импульсами тока были заменены вследствие отказов только 3 элемента (все 3 на первом году эксплуатации): 2 из-за пробоя диодов типа Д1, по-видимому, обладавших дефектами изготовления, и 1 из-за нарушения изоляции между обмотками импульсного трансформатора. Машина устойчиво работала при значительной нестабильности напряжения питающей электросети и в достаточно широком диапазоне температур окружающей среды (от +15 до +30° С). Серийные экземпляры машин "Сетунь" успешно эксплуатировались в различных климатических зонах как с холодным, так и с жарким, а также резко континентальным климатом (например, в Ашхабаде, Душанбе, Махачкале, Иркутске, Якутске, Одессе), причем без какого-либо сервисного обслуживания и практически без запасных частей. Едва ли это может свидетельствовать о плохой надежности аппаратуры.

Нельзя согласиться и с имеющимся в [2] заключением, будто "Сетунь" была арифметической машиной, совершенно неподходящей для какой-либо оценки троичных логических операций, и что не было предпринято развернутых попыток использования ее для критического сравнения двоичных и троичных машин в отношении арифметических операций. Разработка "Сетуни", в отличие от упоминаемого в [2] моделирования на двоичной микропрограммной машине троичного компьютера TERNAC, не предназначалась специально для исследования возможностей троичной техники и сравнения ее с двоичной техникой. Целью разработки было создание недорогой и простой в использовании малой цифровой машины широкого назначения (нечто в роде того, что десять лет спустя стали называть миникомпьютером), и эта цель была

успешно достигнута. То же, что требуемая машина получилась троичной, было, пожалуй, делом случая: ввиду ненадежности имевшихся в то время транзисторов были созданы магнитные логические элементы, выполненные на нелинейных трансформаторах импульсов тока с диодами (быстродействующие магнитные усилители с питанием импульсами тока), причем оказалось, что эти элементы не только весьма удобны для построения троичных цифровых устройств, но и что троичные устройства получаются существенно более экономными в отношении количества оборудования и потребляемой мощности, более быстрыми и структурно более простыми, чем двоичные устройства, реализованные на тех же элементах [4]. Однако, хотя исследование троичной цифровой техники и не было самоцелью разработки "Сетуни"; разработка эта внесла в понимание проблем и возможностей троичной техники больше, чем теоретическое изучение экономности троичного кодирования и методов синтеза функций трехзначной логики.

Принципиальные результаты разработки "Сетуни" настолько естественны и немудрены, что, с точки зрения формальных теоретиков многозначной логики, их попросту не видно. Похоже, что именно этим объясняется приведенное выше заключение о слабости логических возможностей "Сетуни" и неиспользовании ее для сравнения троичной арифметики с двоичной. Результаты состояли в том, что было экспериментально доказано, что троичная машина, по меньшей мере в условиях электромагнитной техники, оказывается существенно экономнее, быстрее, проще и математически совершенней функционально эквивалентной двоичной машины, выполненной на элементах того же типа. Кроме того, было показано, что троичные устройства могут быть эффективно и просто реализованы на основе способа выполнения логических операций, названного впоследствии пороговой логикой, причем именно в трехзначном варианте с положительными и отрицательными весами логических входов данный способ становится практически приемлемым, благодаря значительному ослаблению требований к точности и стабильности параметров физических элементов и сигналов. Не менее важным было и то, что трехзначная логика с ее 33 одноместных и 39 двуместных операций, трактуемая некоторыми философами как логика, таинственного микромира, предстала перед инженером как давно известная ему логика положительного, отрицательного и равного нулю тока (или заряда), а перед программистом - как логика элементарных чисел: 0, 1, - 1 или логика значений, принимаемых алгебраическим знаком числа: +, -, 0. При этом выяснилось, что, хотя эта трехзначная логика сложнее двузначной, она вместе с тем удобнее для человека, легче осваивается и применяется.

Вывод о неразвитости логических возможностей "Сетуни" сделан из того, что в наборе ее команд имеется только одна логическая операция - операция умножения тритов (поразрядное умножение), вследствие чего якобы машина не подходила для оценки троичных логических операций. Невозможно понять, почему для оценки логических операций (и что это за оценка) надо строить машину, и "Сетунь", как уже было сказано, была создана для иных целей, но указанной единственной операции для программирования логики было достаточно. Может быть, как раз это и является той оценкой троичных операций, которую имеют в виду критики. Операция умножения тритов была включена в набор команд "Сетуни" как наиболее часто встречающаяся и легко понимаемая. Это аналогично наличию единственной логической операции

конъюнкции в наборе команд PDP-8. Однако операция умножения тритов несравнимо богаче двужначной конъюнкции [4]. Использование ее в сочетании с константами и арифметической операцией сложения позволяет осуществить произвольную функцию трехзначной логики. Впрочем, последнее имеет, пожалуй, лишь теоретическое значение, так как для программиста основными средствами выражения логики являются команды условного перехода, которых у "Сетуни" три, а команда умножения тритов используется главным образом для выделения, очистки или изменения знака частей слова.

Следует отметить, что при создании "Сетуни" идеалом было не предоставление программисту как можно большего набора различных команд, а, напротив, стремление сделать этот набор как можно меньшим, но вместе с тем обеспечивающим возможность эффективно запрограммировать широкий круг различных применений. Это должно было способствовать удешевлению машины и, с другой стороны, облегчить освоение ее программистами. Указанное стремление увенчалось успехом в известной степени благодаря симметричному троичному коду - в наборе команд "Сетуни" всего 24 команды.

АРХИТЕКТУРА ЦИФРОВОЙ МАШИНЫ "СЕТУНЬ"

Подтвердим сказанное кратким описанием архитектуры машины "Сетунь" [5]. Ее можно охарактеризовать как одноаккумуляторную, с индекс-регистром и одноадресным форматом команд. Длина аккумулятора 18 тритов, длина ячеек главной памяти 9 тритов, что эквивалентно 14,3 бита. Команды занимают по одной ячейке памяти, а операнды могут занимать как по одной, так и по две ячейке, рассматриваемые как одно 18-тритное слово.

Девять тритов, кодирующих команду $K(1:9)$, разделяются на код операции $K(6:8)$ и адресную часть $K(1:5)K(9)$. Первые четыре трита $K(1:4)$ адресной части составляют адрес пары 9-тритных ячеек, обращение к которым производится тройко в зависимости от значения $K(5)$: при $K(5)=0$ доступна одна из этих ячеек, при $K(5)=1$ - другая, при $K(5)=-1$ - 18-тритное слово, младшей частью которого является ячейка, соответствующая $K(5)=1$, а старшей - ячейка, соответствующая $K(5)=0$. Трит $K(9)$ управляет индексацией так, что исполнительный адрес EA определяется выражением

$$EA(1:5) = K(1:5) + K(9) * F(1:5),$$

где $F(1:5)$ - индекс-регистр. Поскольку $K(9)$ принимает значения 0, 1, -1, то EA равно либо $K(1:5)$, либо $K(1:5)+F(1:5)$, либо $K(1:5)-F(1:5)$.

Адресное пространство главной памяти простирается от $-(35-1)/2$ до $(35-1)/2$, т. е. от -121 до +121, но так как каждый третий адрес использован для указания пары ячеек, обладающих каждая отдельным адресом, то адресовать можно только 162 девятитритных ячейки и 81 пару ячеек. Последовательная выборка команд производится с пропуском адресов, соответствующих парам ячеек, т. е. программный счетчик принимает следующие значения: -120, -119, -117, -116, ..., -3, -2, 0, 1, 3, 4, ..., 117, 118, 120, 121.

При обменах с внешней памятью, которая реализована на магнитном барабане и является основной памятью машины, а также при вводе и выводе главная память представляется разделенной на страницы, содержащие по 54 ячейки. Номером страницы служит значение старшего трита принадлежащих этой странице адресов. Другими словами, память структурирована как массив

$m(-1:1, -80:81, 1:9)$, в котором средняя компонента индекса пропускает каждое третье значение. Обмен с магнитным барабаном производится целыми страницами, причем трит $K(1)$ в команде обмена указывает номер участвующей в обмене страницы главной памяти, а триты $K(2:5)$ - номер страницы магнитного барабана. Ввод/вывод в девятеричном коде осуществляется целыми страницами, а в алфавитно-цифровом коде - записями произвольной длины с окончанием по литере "стоп" и по исчерпанию страницы памяти.

Тройка тритов $K(6:8)$ командного слова, используемая в качестве кода операции, принимает 27 различных значений, чем ограничивается количественный состав набора команд машины. Однако в действительности набор состоял из 24 команд, так как 3 значения кода операции были зарезервированы и в серийных машинах не использовались.

Из предоставленных программисту 24 команд две команды управляют страничным обменом между главной памятью и магнитным барабаном, одна команда задает операции ввода/вывода, причем номера устройств и режимов работы указываются в поле $K(2:5)$. Операции, соответствующие остальным командам, определены над регистрами процессора и ячейками главной памяти.

В процессоре имеется пять программно доступных регистров: аккумулятор $S(1:18)$, регистр множителя $K(1:18)$, индекс-регистр $F(1:5)$, программный счетчик $C(1:5)$, регистр знака результата $w(1:1)$.

Над аккумулятором и указываемой адресной частью ячейкой памяти определены следующие операции:

засылка в аккумулятор $S:=m(EA)$;

сохранение результата $m(EA):=S$;

сложение $S := S+m(EA)$;

вычитание $S:=S - m(EA)$;

умножение тритов $S:=SDm(EA)$;

сдвиг $S:=S \cdot 3^m(EA, 1:5)$;

нормализация $m(EA):=S_{\text{норм}}$; $S:=0$; $S(1:5):=n_{\text{сдв}}$;

Значение доставляемого из главной памяти операнда $m(EA)$ может быть в зависимости от $K(5)$ либо длинным, 18-тритным, либо коротким, 9-тритным. В последнем случае оно соответствует старшей половине 18-тритного аккумулятора, т. е. дополняется до 18-тритного 9 нулями справа. Запятая, отделяющая дробную часть числа от целой, подразумевается между вторым слева и третьим тритами аккумулятора.

В операции сдвига операнд $m(EA, 1:5)$ является целым числом, представленным пятью старшими тритами соответствующей ячейки памяти.

Операция нормализации заключается в преобразовании путем сдвига значения S в $S_{\text{норм}}$, удовлетворяющее условию $0,5 < |S_{\text{норм}}| < 1,5$. При этом $S_{\text{норм}}$ сохраняется в главной памяти, а число произведенных сдвигов $n_{\text{сдв}}$ помещается в $S(1:5)$. В случае $S=0$ принято $S_{\text{норм}}=0$, $n_{\text{сдв}}=0$.

Регистр множителя K используется совместно с S в командах с операцией умножения. Таких команд три:

- 1) $R:=S; S:=R*m(EA);$
- 2) $S:=S+R*m(EA);$
- 3) $S=R*S+m(EA).$

Кроме того, имеется команда засылки в R, т. е. $R:=m(EA)$, и команда засылки в K с остановом процессора: $K:=m(EA); stop$.

При выполнении операций сложения, вычитания и умножения возможно переполнение аккумулятора. Переполнение вызывает останов процессора, причем на пульте машины наряду с индикацией содержимого S и R отображаются также значения двух тритов переполнения S(-1:0).

С индекс-регистром F(1:5) связаны четыре операции:

засылка $F:=m(EA, 1:5);$

сохранение результата $m(EA):=0; m(EA, 1:5):=F;$

сложение $F:=F+m(EA, 1:5);$

сложение с C $F:=C+m(EA, 1:5) .$

Все перечисленные выше операции над регистрами S,R,F сопровождаются фиксацией в однитритном регистре w знака полученного результата. В зависимости от значения w производится выбор продолжения программы при выполнении команд условного перехода, которых имеется три:

- 1) if $w=0$ then $C:=EA;$
- 2) if $w=1$ then $C:=EA;$
- 3) if $w=-1$ then $C:=EA.$

Кроме того, имеется команда безусловного перехода $C:=EA$ и команда сохранения текущего значения программного счетчика $m(EA)=C$.

Ручное управление машиной осуществляется с пульта управления кнопками "стоп", "пуск", "начальный пуск", "команда ПУ". Последняя позволяет выполнить команду, код которой набран на клавишах пульта. Имеются также клавиши для задания останова по данному значению программного счетчика и по команде с данным значением исполнительного адреса, причем задается тип команды: ввода/вывода, обмена с магнитным барабаном, прочие.

На первый взгляд описанная архитектура не отличается от архитектуры двоичных одноаккумуляторных машин: те же регистры, те же операции. Однако при внимательном рассмотрении выявляются следующие принципиальные и важные для практики особенности.

1. Числа всех типов представлены единым натуральным кодом. В двоичных машинах для представления чисел разных типов и даже разного назначения приходится применять различный код, например: числа с фиксированной запятой представляют обычно дополнительным или обратным кодом, мантиссы чисел с плавающей запятой - прямым кодом, порядки - кодом с избытком, адреса памяти - натуральным двоичным кодом.

2. Операции определены над операндами, длина которых варьируется и может быть неодинаковой у первого и второго операнда. При этом не требуется

никаких усложнений операционного устройства, ни вспомогательных команд в роде двоичной команды "расширения знака".

3. При усечении слова, например в случае присваивания длинного значения короткому регистру, автоматически получается наилучшее при данной укороченной длине представление первоначального значения, и вместе с тем сохраняемая часть слова копируется неизменной. В двоичной машине соответствующие возможности с известными оговорками можно обеспечить, лишь предусмотрев два варианта операций - с округлением и без округления.

4. Единственная операция сдвига выполняет функции всех двоичных операций сдвига - логического, арифметического, с округлением и без округления, причем выполняет безукоризненно, чего нельзя сказать, например, об операции двоичного арифметического сдвига [6].

5. Знак числа в соответствии с общепринятой математической трактовкой этого понятия представлен трехзначной функцией, и разбиение чисел по знаку производится на положительные, отрицательные и равные нулю в противоположность сбивающей с толку двоичной традиции двузначного знака и отнесения нуля к положительным числам.

6. Интервал значения мантиссы нормализованного числа - от 0,5 до 1,5 по абсолютной величине - характеризуется значительно лучшей устойчивостью по сравнению с используемыми в двоичных машинах интервалами 0,5 - 1,0 и 1,0 - 2,0.

Эти и некоторые другие особенности троичной архитектуры обусловили математическое совершенство, компактность и простоту реализации набора команд "Сетуни", а главное, естественность, легкость понимания и применения машины пользователями.

АРХИТЕКТУРА ЦИФРОВОЙ МАШИНЫ "СЕТУНЬ 70"

Экспериментальный образец малой цифровой машины "Сетунь 70" эксплуатируется в МГУ с 1970 г., причем с 1974 г. на его основе функционирует автоматизированная система обучения "Наставник", обеспечивающая поточное обучение студентов языку программирования Фортран, проведение коллоквиумов и тестов, разработку учебных материалов, дидактические и психофизические эксперименты. Необычность архитектуры машины стимулировала разработку оригинальных системных программ и новой версии структурированного программирования.

В техническом отношении "Сетунь 70" характеризуется рядом усовершенствований по сравнению с "Сетуной". Так, реализация однопроводной передачи трехзначных сигналов позволила почти в 2 раза уменьшить число электрических соединений, логические элементы стали проще, миниатюрней и при большей релейности потребляют в 2,5 раза меньше энергии, значительно улучшены параметры троичной памяти и магнитной записи троичного кода. Дальнейшее развитие получила пороговая техника осуществления операций трехзначной логики. Разработанная применительно к электромагнитным средствам эта техника переносима и на полупроводниковые элементы, например, типа И2Л.

Все же с точки зрения практики в настоящее время большой интерес представляют архитектурные особенности "Сетуни 70".

Минимальная непосредственно адресуемая единица главной памяти "Сетуни 70" - 6-тритный трайт (~9,5 бита) - на редкость удобна. Трайт лишь немногим больше 8-битного байта, но уже достаточно велик, чтобы закодировать, например, алфавит, включающий русские и латинские заглавные и строчные буквы, цифры, математические и служебные знаки. В трайте целое число как 9-ричных, так и 27-ричных цифр. Два трайта - это 19 битов, три трайта - почти 29 битов и т. д.

Представление чисел симметричным троичным кодом позволило легко реализовать последовательное выполнение арифметических операций с варьируемой длиной операндов от 1 до 3 трайтов и длиной результата до 6 трайтов. Благодаря симметричности кода просто и естественно реализованы реверсивные счетчики и указатели стеков, играющих в архитектуре машины важную роль.

Существенной чертой "Сетуни 70" является стековая организация процессора. Введение арифметического стека, т. е. магазина для автоматического запоминания промежуточных результатов, было обусловлено выбором в качестве языка машины польской инверсной записи (ПОЛИЗ), которую предпочли как удобный выходной язык трансляторов и вследствие ее компактности [7]. Машинная программа представляется в ПОЛИЗ последовательностью слов (или слогов), в которой различаются операционные и адресные слова. Адресное слово означает засылку в стек значения, хранимого в главной памяти по соответствующему адресу. Операционное слово указывает одну из определенных над стеком операций, требуя выполнения этой операции процессором.

Ради экономии дорогой в те годы памяти были приняты короткие 6-тритные слова (трайты). Операционный трайт в двух старших тритах содержит нули, а в остальных 4-х тритах - код одной из 81 операций. Операции разбиты на три класса: 27 основных, 27 служебных и 27 макроопераций, программируемых пользователем. Соответственно этим классам имеется три режима работы машины: основной или пользователя, прерывания и макро.

В адресном трайте первый (старший) трит указывает длину операнда, второй трит - один из трех регистров "приписки", содержащих номера открытых для доступа страниц памяти, остальные четыре трита - адрес операнда (старшего трайта операнда) на выбранной по регистру "приписки" странице. Счетчик команд и указатель арифметического стека имеют собственные регистры "приписки". Ширина стека - три трайта. Помимо стека в операционном устройстве имеется регистр порядков, длиной в 1 трайт, и регистры множителя и младшей части результата, по 3 трайта каждый.

Для автоматического сохранения адреса возврата при обработке прерывания и при выполнении макрооперации имеется второй (системный) стек, число позиций которого в первоначальном варианте машины было ограничено двумя, а с введением команд структурированного программирования увеличилось до 26.

В 1975 г. машина подверглась модернизации, выразившейся в небольшом по объему переделках, но принципиальном усовершенствовании архитектуры на

основе идеи структурированного программирования Э. Дейкстры [8, 9]. Двухстековая организация процессора и ПОЛИЗ оказались исключительно благоприятными для реализации структурированного программирования на уровне языка машинных команд. При этом в условиях новой дисциплины программирования стали несущественными затруднения, возникавшие в связи с мелкостраничной структурой памяти.

Все, что потребовалось сделать - это ввести команды ветвления, цикла и вызова подпрограммы вместо практически не употреблявшихся команд приращения, убавления и установки нуля в регистре порядков. Новые команды, в отличие от обычных в ПОЛИЗ однословных команд, представлены словосочетаниями. Например, команда JSR A вызова подпрограммы A включает операционный трайт JSR и следующий за ним трайт A, интерпретируемый как указатель начала подпрограммы, находящейся в 9-страничной оперативной памяти. Тело подпрограммы заканчивается трайтом RMC, означающим возврат к продолжению программы, из которой произведен вызов подпрограммы.

Команда тройственного ветвления BRT A1 A2 A3 состоит из 4-х трайтов и задает вызов одной из трех подпрограмм в зависимости от знака текущего значения вершины арифметического стека: если минус, то A1, если нуль, то A2, если плюс, то A3. Команда цикла DOWA повторно вызывает подпрограмму A, пока текущее значение вершины не равно нулю.

С появлением данных команд старые команды безусловного и условных переходов вышли из употребления, хотя и сохранены в машине. Развита на основе новых команд версия структурированного программирования стала истоком системы ДССП [10] для двоичных микрокомпьютеров.

ЛИТЕРАТУРА

1. Брусенцов Н.П. Пороговая реализация трехзначной логики электромагнитными средствами. В кн.: Вычислительная техника и вопросы кибернетики. Вып. 9. М.: Изд-во МГУ, 1972, с. 3-35.
2. Epstein G., Frieder G., Rine D. C. The development of multiple - valued logic as related to computer science. Computer, 1974, Sept., vol. 7, N 9, p. 20-32.
3. Аннотированный указатель программ для вычислительной машины "Сетунь". Составители: Н.П. Брусенцов, В.А. Морозов. Фонд алгоритмов и программ Минвуза СССР. М.: Изд. ВЦ МГУ. Вып. 1; 1968. Вып. 2, 1971.
4. Брусенцов Н.П. Опыт разработки троичной вычислительной машины. - Вестн. Моск. ун-та. Сер. 1: математика, механика, 1965, # 2, с. 39-48.
5. Брусенцов Н.П., Маслов С.П., Розин В.П., Тишулина А.М. Малая цифровая вычислительная машина "Сетунь". М.: Изд-во МГУ, 1965.
6. Steele G.L. Arithmetic shifting considered harmful. SIGPLAN Notices, Nov. 1977, vol. 12, N 11, p. 61-69.
7. Брусенцов Н.П., Жоголев Е.А. Структура и алгоритм функционирования малой вычислительной машины. В кн.: Вычислительная техника и вопросы кибернетики. Вып. 8. Л.: Изд-во ЛГУ, 1971, с. 34-51.

8. Дал У., Дейкстра Э., Хоор К. Структурное программирование. М.: Мир, 1975.

9. Брусенцов Н.П., Рамиль Альварес Х. Структурированное программирование на малой цифровой машине. В кн.: Вычислительная техника и вопросы кибернетики. Вып. 15. М.: Изд-во МГУ, 1978, с. 3-8.

10. Брусенцов Н.П., Златкус Г.В., Руднев И.А. ДССП - Диалоговая система структурированного программирования. В кн.: Программное оснащение микрокомпьютеров. М.: Изд-во МГУ, 1982, с. 11-40.

Статья опубликована в сборнике
Архитектура и программное оснащение цифровых систем.
М.: Изд-во Моск. ун-та, 1984. С. 114-123.