

О преимуществах троичных машин и эффективности троичных вычислений

А.А. Бурцев¹, В.А. Бурцев²

¹ ФГУ ФНЦ НИИСИ РАН, Москва, Россия, burtsev@niisi.msk.ru

² Калужский филиал МГТУ имени Н.Э. Баумана, Калуга, Россия, basel99@mail.ru

Аннотация. В статье рассматриваются преимущества троичных машин и демонстрируется эффективность троичных вычислений на примерах разработанных авторами программ решения ряда учебных задач в диалоговой системе структурированного программирования (ДССП), созданной в МГУ им. М.В. Ломоносова для троичной виртуальной машины (ТВМ).

Ключевые слова: троичная симметричная система счисления, троичная виртуальная машина, ДССП, троичная логика, троичное ветвление, троичный цикл.

1. Введение

Общепризнано, что в настоящее время двоичная микроэлектроника приблизилась к пределу своих технологических возможностей. Поэтому возрождается интерес к исследованиям путей построения компьютерной техники на основе других систем счисления.

Одной из наиболее перспективных считается троичная симметричная система счисления (с цифрами $-1, 0, +1$). На заре развития вычислительной техники её безусловные преимущества отмечал ещё Дональд Кнут [1], предвещая ей успешное применение в будущем.

Более полувека назад эффективность вычислительной техники, построенной на основе такой системы счисления, была убедительно подтверждена практической разработкой и применением троичных цифровых машин (ЦМ) «Сетунь» и «Сетунь-70» [2], разработанных в НИЛ ЭВМ МГУ под руководством Брусенцова Н. П. О преимуществах троичных машин Брусенцовым была написана серия статей под общим названием «Заметки о троичной цифровой технике» [3]. А относительно недавно в преимуществах троичной вычислительной техники убедились и разработчики квантовых компьютеров, заявивших, что «будущее квантовых компьютеров — в троичных вычислениях» [4].

В настоящее время в качестве среды разработки и прогона учебных программ можно использовать программный комплекс ДССП-ТВМ [5], созданный в НИЛ ТИ на факультете ВМК МГУ. Он включает имитатор троичной машины (ТВМ) и систему разработки программ для нее на языке ДССП-Т — троичном варианте языка ДССП.

Далее в статье перечисляются преимущества

троичных машин, основанных на троичной симметричной системе счисления, и демонстрируется эффективность троичных вычислений на примерах разработанных авторами программ решения ряда учебных задач в ДССП-ТВМ.

2. Преимущества троичных компьютеров

Кратко перечислим основные преимущества троичных компьютеров (более подробно они объясняются в [3] и [6]).

1. Среди целочисленных позиционных систем счисления наиболее экономичной является троичная. Доказательство этого утверждения можно найти во многих источниках, например, в [6] (см. с.29), поэтому приводить его здесь не будем, а дадим лишь некоторые пояснения.

Понятие экономичности системы счисления непосредственно связано с объемом оборудования, которое требуется изготовить для работы с машинными словами как с числами. Представим себе аппаратуру как магнитную азбуку (кассу), из знаков которой нам требуется составлять всякое p -ичное число из n разрядов. Сколько литер (знаков) z необходимо изготовить в такой кассе? Очевидно, $z = n \cdot p$ знаков (по p знаков на каждую позицию). А какой диапазон чисел d можно будет при этом представить? Ясно, что $d = p^n$. В предположении, что z делится на p , можно выразить d формулой $p^{(z/p)}$.

Вот такой величиной d и оценивается экономичность системы счисления по основанию p . Она определяет количество чисел, которое можно представить в этой системе счисления p при помощи заданного числа знаков z . Наиболее экономичной считается система счисления, у которой эта величина $d = p^{(z/p)}$ максимальна.

Оценим величину экономичности систем счисления по основанию 2, 3, 4, 5, 10 для 60 знаков и представим результаты в таблице 1.

Таблица 1. Оценка экономичности систем счисления по основанию $p=2, 3, 4, 5, 10$ для $z=60$ знаков.

p	число разрядов n	диапазон чисел p^n
2	30	$2^{30} = 1073741824$
3	20	$3^{20} = 3486784401$
4	15	$4^{15} = 1073741824$
5	12	$5^{12} = 244140625$
10	6	$10^6 = 1000000$

Как видно из этой таблицы, наибольшая величина экономичности (среди всех рассмотренных) оказывается у троичной системы счисления. Заметим также, что у 2-ной и 4-ной систем счисления эта величина одинаковая.

При проведении строгого доказательства рассматривают функцию $F(x) = x^{\wedge}(z/x)$ на диапазоне вещественных чисел, и определяют, при каком x она принимает максимальное значение. Для поиска экстремума приравнивают её производную нулю и получают в качестве ответа число Эйлера $e=2,71828...$. А среди целых чисел к ней ближе всего число 3. Вот поэтому троичная система оказывается самой экономичной системой счисления среди всех целочисленных.

2. В троичном машинном слове из n разрядов можно закодировать значительно больший диапазон значений: 3^n , а в двоичном – только 2^n , т.е. троичный код имеет выигрыш в $(1,5)^n$ у двоичного, что демонстрируется таблицей 2:

Таблица 2. Диапазоны значений для 3^n и 2^n

n	2^n	3^n	$3^n / 2^n$
5	32	243	$\approx 7,6$
8	256	6561	$\approx 25,6$
16	65536	43046721	$\approx 656,8$
27	134217728	7625597484987	$\approx 56815,1$
32	4294967296	1853020188851841	$\approx 431439,9$

Из приведённой таблицы, в частности, можно увидеть, что 27-тритным словом можно перекрыть диапазон 32-битного слова уже примерно в 1775 раз ($3^{27}/2^{32}$).

3. Троичная симметричная система является минимальной, в которой можно единообразно представлять положительные и отрицательные числа (без каких-либо «ухищрений» в форме дополнительного кода).

4. В троичной симметричной системе счисления упрощаются реализации операции сдвигов и сравнения чисел. И не требуется различать арифметические и логические команды сдвигов (благодаря единообразному представлению положительных и отрицательных чисел).

5. А также естественно выражаются результаты трехзначных операций сравнения чисел

($<,=,>$) и функции «знак числа» ($- , 0, +$).

6. Можно непосредственно воплощать троичную логику, т.е. вычислять логические выражения с тремя значениями: $+1$ («да»), -1 («нет»), 0 («не знаю» или «может быть»).

7. Для округления числа достаточно просто отсечь младшие незначащие разряды, т.к. отсекаемый остаток не будет превышать (по абсолютной величине) половины значения последнего значимого разряда.

В этих преимуществах можно наглядно убедиться, выполняя троичные вычисления, т.е. вычисления на троичных машинах.

3. Примеры эффективности троичных вычислений

Прежде всего, заметим, что троичная машина может исполнить всё то, что может двоичная. Но при этом она может предоставить для построения компьютерной программы гораздо более широкие возможности. В их числе следует отметить возможность эффективной реализации троичного ветвления, цикла с троичным условием, операций троичной арифметики и троичной логики.

Эффективность троичных вычислений продемонстрируем на примерах разработки в ДССП-ТВМ программ для решения ряда учебных задач в двух вариантах: 1) для троичной машины и 2) для двоичной, т.е. с использованием только тех операций, которые можно исполнить на двоичной машине.

3.1. Пример №1

Рассмотрим задачу попадания точки в определенную область (заштрихованную фигуру) на плоскости (см. рис. 1) с трёхзначным вариантом ответа: точка попадает внутрь области ($+1$), на границу области (0), вне области (-1).

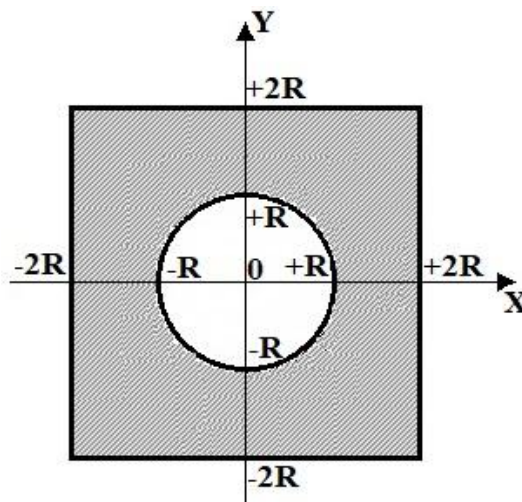


Рис. 1. Заданная фигура на плоскости

Для решения этой задачи на троичной машине можно просто составить троичное логическое выражение:

```
inSqr(x,y,R) = (sign(2R-|x|)) &^t (sign(2R-|y|))
inCircle(x,y,R) = sign(R*R-x*x-y*y)
inFigt(x,y,R) = inSqr(x,y,R)&^t~inCircle(x,y,R)
```

используя троичные логические операции:

```
sign(x) – определение знака числа x
(SGN – на языке ДССП-Т)
&^t – потритная конъюнкция
(TMIN – на языке ДССП-Т)
~ – отрицание
(NEG – на языке ДССП-Т)
```

Для вычисления этого выражения составим программу на языке ДССП-Т:

```
: inSquare?t {x,y,R}
*2 E3 ABS E2 ABS C3 E2
CMP E3 E2 CMP TMIN
{(|y|<?>2R)&t(|x|<?>2R)} ;
: inCircle?t {x,y,R}
2^ E3 2^ E2 2^ +
CMP {R^2<?>x^2+y^2} ;
: inFigt?t {x,y,R}
C3 C3 C3 inSquare?t
E4 E3 E2 inCircle?t NEG TMIN
{(|y|<?>2R)&t(|x|<?>2R)&t~(R^2<?>x^2+y^2)} ;
```

А для двоичной машины решение не получится сформулировать в виде логического выражения, но можно предложить вычислить нужный результат с помощью арифметических операций с применением функций min(x,y) и sign(z):

```
inSqr(x,y,R) = min(sign(2R-|x|),sign(2R-|y|))
inCircle(x,y,R) = sign(R*R-x*x-y*y)
inFigt(x,y,R) = min(inSqr(x,y,R),-inCircle(x,y,R))
```

и составить соответствующую программу на языке ДССП-Т:

```
: sign{x} C {x,x} BR- -1 sign01 {z};
: sign01{x} BR0 0 1 {z} ;
: min{a,b} C2 C2 < BR+ D E2D {min(a,b)};
: inSquare{x,y,R}
*2 E3 ABS E2 ABS C3 E2 - sign
E3 E2 - sign min
{min(sign(2*R-|x|),sign(2*R-|y|))} ;
: inCircle{x,y,R}
2^ E3 2^ E2 2^ + - sign
{sign(R^2-x^2-y^2)} ;
: inFigt{x,y,R}
C3 C3 C3 inSquare E4 E3 E2
inCircle NEG min
{min(inSquare(x,y,R),-inCircle(x,y,R))} ;
```

Поскольку для реализации функций sign и min приходится использовать команды ветвления, решение этой задачи с применением операций, присущих двоичной машине, потребует исполнения большего числа команд, т.е. будет менее производительным.

3.2. Пример №2

Рассмотрим задачу вычисления значения многовариантной функции, заданной следующей формулой:

$$F(x,y) = \begin{cases} x + y + 1 & \text{при } y > 0, x > 0 \\ x^3 & \text{при } y = 0, x > 0 \\ x + y^2 - 1 & \text{при } y < 0, x > 0 \\ y^3 & \text{при } y > 0, x = 0 \\ 100 & \text{при } y = 0, x = 0 \\ y^2 & \text{при } y < 0, x = 0 \\ y + x^2 + 1 & \text{при } y > 0, x < 0 \\ x^2 & \text{при } y = 0, x < 0 \\ y^2 + x^2 - 1 & \text{при } y < 0, x < 0 \end{cases}$$

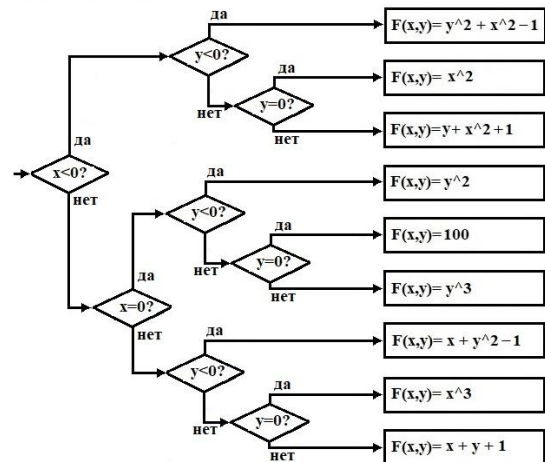


Рис. 2. Блок-схема с двоичными ветвлениями.

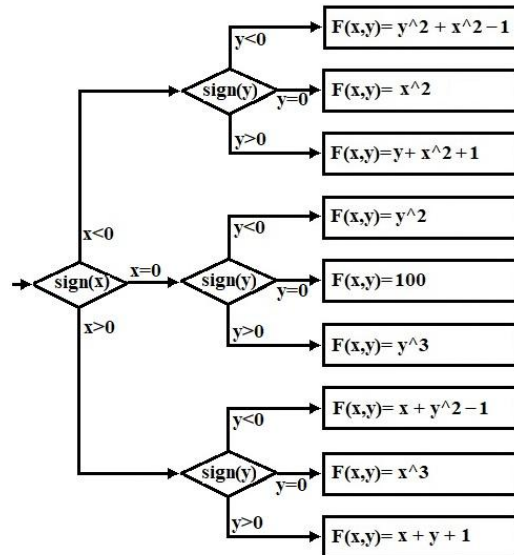


Рис. 3. Блок-схема с троичными ветвлениями.

Представим решение этой задачи в виде блок-схем программ. Для двоичной машины мы можем применять только команду двоичного ветвления (см. рис. 2). А в варианте для троичной машины можно применить команду троич-

ного ветвления (см. рис.3).

Процедуру вычисления функции с применением троичного ветвления можно оформить на языке ДССП-Т:

```
: MVF{x,y} C2 BRS x< x0 x> {z} ;
: x<{x,y} C BRS x<y< x<y0 x<y> {z};
: x<y<{x,y} ^2 E2 ^2 + 1-
      {z=y^2+x^2-1} ;
: x<y0{x,y} D ^2 {z=x^2} ;
: x<y>{x,y} E2 ^2 + 1+ {z=y+x^2+1};
: x0{x,y} C BRS x0y< x0y0 x0y> {z};
: x0y<{x,y} ^2 E2D {z=y^2};
: x0y0{x,y} DD 100 {z=100} ;
: x0y>{x,y} ^3 E2D {z=y^3};
: x>{x,y} C BRS x>y< x>y0 x>y> {z};
: x>y<{x,y} ^2 + 1- {z=x+y^2-1} ;
: x>y0{x,y} D ^3 {z=x^3} ;
: x>y>{x,y} + 1+ {z=x+y+1};
: ^2 {x} C * {x^2} ;
: ^3 {x} C C * * {x^3} ;
```

Сравнивая эти две блок-схемы, изображенные в виде деревьев, можно легко увидеть, что на рис. 2 представлено 4-х ярусное дерево, а на рис. 3 – 2-х ярусное дерево. Это означает, что при вычислении данной функции на троичной машине (согласно рис. 3) будет всегда исполняться 2 троичных ветвления, а в версии для двоичной машины (по рис. 2) может потребоваться исполнить операцию ветвления 2, 3 или даже 4 раза. Значит, на троичной машине вычисление этой функции в большинстве случаев будет исполнено за меньшее число команд, т.е. будет более производительным.

3.3. Пример №3

Решение следующей задачи для троичной машины основывается на применении цикла с троичным условием. В ДССП-ТБМ такой вид цикла обеспечивается командой повторения **DW-+**. Эта команда цикла совмещает в себе проверку, надо ли прекращать его повторение, с выбором одного из двух действий, предусмотренных для исполнения в качестве тела цикла на очередном его шаге.

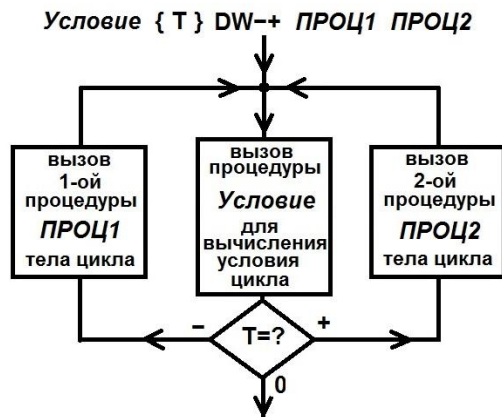


Рис. 4. Блок-схема выполнения цикла с троичным условием

Блок-схема выполнения этого цикла приведена на рис. 4 (подробнее он объясняется в [7]).

Продemonстрируем эффективность применения троичного цикла на примере задачи подсчета разницы между количествами положительных и отрицательных чисел заданной последовательности (в предположении, что последовательность чисел задаётся в стеке до первого встреченного нулевого элемента).

Представим на языке ДССП-Т два варианта программы решения этой задачи: А) с применением троичного цикла (**DW-+**); Б) с применением двоичного цикла (**DW**) и команды ветвления.

Вариант А

```
: CounNPt {~, 0, X1, ..., Xn-1, Xn}
0 {d=0} {~, 0, X1, ..., Xn-1, Xn, d}
E2 {Xi?} DW-+ 1- 1+ {~, d} ;
```

Вариант Б

```
: CounNP {~, 0, X1, ..., Xn-1, Xn}
0 {d=0} {~, 0, X1, ..., Xn-1, Xn, d}
C2{Xn} DW CounNP1 {~, 0, d} E2D{d};
: CounNP1 {~, Xi, d} E2
{~, d, Xi} BR+ 1+ 1- {~, d} ;
```

Чтобы сравнить производительность этих процедур (**CounNPt** и **CounNP**) выполнения задачи, на программном имитаторе троичной машины (ТБМ) они запускались многократно для одинаковых последовательностей чисел различной длины с замером количества исполняемых ими команд. В результате проведённых испытаний было подтверждено, что алгоритм процедуры **CounNPt**, применяющий цикл с троичным условием, работает быстрее (более, чем в 2 раза), так как исполняется за меньшее число команд ТБМ (см. таблицу 3).

Таблица 3. Сравнение производительности процедур вариантов А и Б.

число элементов	количество исполненных команд	
	вариант А	варианта Б
10	41	85
20	71	155
40	131	295
50	161	365
100	311	715

5. Заключение

В статье представлены преимущества троичного компьютера. А также продемонстрировано, что в ряде случаев на таком компьютере можно более эффективно решать вычислительные задачи, применяя новые возможности, которые может предоставить программисту троичная машина и её система программирования.

About the benefits of ternary machines and the effectiveness of ternary computing

A.A.Burtsev, B.A.Burtsev

Abstract. The article considers the advantages of ternary computers and demonstrates the effectiveness of ternary calculations on the basis of examples of programs developed by the authors to solve some of educational tasks in the dialogue system of structured programming (DSSP), created at the Moscow State University named after M.V. Lomonosov for the ternary virtual machine (TVM).

Keywords: ternary symmetric notation system, ternary virtual machine, DSSP, ternary logic, ternary branching, ternary loop.

Литература

1. Д. Кнут. Искусство программирования на ЭВМ. М., Мир, 1978, Т.2, с. 216-219.
2. Н.П. Брусенцов, Х. Рамиль Альварес. Троичные ЭВМ “Сетунь” и “Сетунь 70”. «Первая Международная конференция "Развитие вычислительной техники в России и странах бывшего СССР: история и перспективы" SORUCOM-2006». Россия, г. Петрозаводск, 3-7 июля 2006 г. (труды конференции в 2-х ч.). Изд-во ПетрГУ, 2006. ч.1, 45-51.
3. Н.П. Брусенцов. Заметки о троичной цифровой технике. «Вычислительная техника и вопросы кибернетики», Вып. 15 (1978), 145–155.
4. Будущее квантовых компьютеров — в троичных вычислениях, http://www.infuture.ru/news.php?news_id=475pdf.
5. А. А. Бурцев, С. А. Сидоров. Троичная виртуальная машина и троичная ДССП. «Программные системы: теория и приложения», Т.6 (2015), №4, 29–97, http://psta.psiras.ru/read/psta2015_4_29-97.pdf.
6. Ю.С. Владимирова. Введение в троичную информатику: учебное пособие. М., АРГАМАК-МЕДИА, 2015.
7. А.А. Бурцев. Особенности программирования в ДССП для троичной машины. «Современные информационные технологии и ИТ-образование», Т. 13 (2017), № 14, 188–196.