

Арифметические алгоритмы системы кодирования 1 из 4 с активным нулем и оценка параметров быстродействия и занимаемой площади блока суммирования

В.В. Лосев, Д.В. Орлов

Национальный исследовательский университет МИЭТ, dsd@miee.ru, torlov@mail.ru

Аннотация — Данная статья посвящена реализации арифметических алгоритмов системы кодирования 1 из 4 с активным нулем. Осуществлен синтез поведенческого описания блока сумматора согласно предлагаемому алгоритму. Реализована тестовая структура на языке Verilog-HDL для проверки, отладки и исследования полученной структуры. Проведено исследование блока суммирования по быстродействию и занимаемой площади.

Ключевые слова — синтез, система кодирования 1 из 4 с активным нулем.

I. ВВЕДЕНИЕ

Основным требованием к вычислительным средствам в системах реального времени (СРВ) является быстродействие, объем памяти, достоверность выдаваемой информации – отсутствие в ней ошибок в течение определенного времени, безопасность, защищенность и экономичность эксплуатации. Требуемая достоверность выдаваемой информации определяется используемыми методами контроля. Существующие на сегодняшний день методы программного или аппаратного (например, мажоритарного) контроля приводят либо к увеличению объема программ, либо к увеличению объема аппаратуры с увеличением потребляемой энергии.

Для обеспечения достоверности контроля выдаваемых данных, сокращения потребляемой энергии, безопасности и защищенности разработана идея применения системы кодирования данных, известной как кодирование один из Р (1 из Р) с использованием активного нуля, которая даёт возможность создавать устройства с постоянным потреблением энергии. Р называют основанием системы кодирования.

Применение такой системы кодирования в микропроцессоре, в первую очередь, ставит задачу создания определенной структуры арифметико-логического устройства (АЛУ) и алгоритмов заложенных в нем.

Главной проблемой применения системы кодирования 1 из Р является выбор величины Р - основания системы кодирования, которая определяет рациональность ее использования для вычислительных систем. В работе [1] показано, что рациональным кодом в системе 1 из Р является код 1 из 4 с активным нулем. Рациональность оценивается на базе сравнения двоичной системы по величине доли использования правильных данных с основанием Р по отношению к данным с двоичным основанием к доли объема аппаратуры при кодировании 1 из Р относительно аппаратуры двоичного кодирования.

В данной статье раскрываются такие вопросы как: правила представления чисел в системе кодирования 1 из 4 с активным нулем, точность представления информации, арифметические алгоритмы блока АЛУ системы кодирования 1 из 4 с активным нулем, оценка быстродействия и занимаемой площади блока сумматора в результате синтеза поведенческого описания согласно предлагаемому алгоритму.

II. АЛГОРИТМ СЛОЖЕНИЯ

Перечень операций, выполняемых в АЛУ, зависит от назначения цифровой вычислительной машины и от функций, выполняемых АЛУ при обеспечении работы остальных устройств. При представлении операций в виде последовательностей микроопераций АЛУ должно состоять из элементов, реализующих эти микрооперации.

АЛУ выполняет все необходимые арифметические и логические операции, но применяется система кодирования информации 1 из Р с активным нулем.

Рациональным кодом в системе 1 из Р является код 1 из 4 с активным нулем, при этом:

«0» = «0001» в коде 1-4 и «00» в двоичном;

«1» = «0010» в коде 1-4 и «01» в двоичном;

«2» = «0100» в коде 1-4 и «10» в двоичном;

«3» = «1000» в коде 1-4 и «11» в двоичном.

Числа представлены в коде 1 из 4 с активным нулем. Количество разрядов в числе 16. Шестнадцатый

разряд используется для кодирования знака числа. Знак (+) представляется кодом 0010, знак минус (-) – кодом 0001. Предварительно числа масштабируются и записываются в память. Числа представлены кодом с фиксированной запятой и хранятся в прямом коде.

Точность представления информации определяется 15 разрядами, код 1 из 4, что соответствует 30 двоичным разрядам. Распределение разрядов показано на рисунке 1.

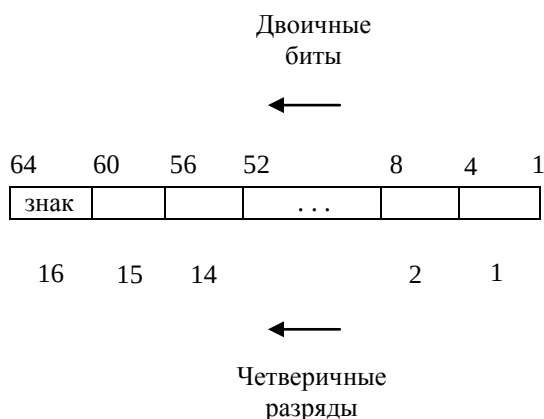


Рис. 1. Распределение разрядов числа в коде 1 из 4

Основной (базовой) операцией, на основе которой строятся все остальные арифметические операции блока АЛУ, является операция сложения. Суммирование начинается с младших четверичных разрядов чисел и продолжается по направлению к старшим четверичным разрядам. При этом учитываются переносы, которые возникают при сложении разрядов (в таблице 1 обозначены буквой П).

Таблица 1

Суммирование одного четверичного разряда в коде «1 из 4» с активным нулем

Десяти- чные числа		0	1	2	3
	1 из 4	0001	0010	0100	1000
0	0001	0001	0010	0100	1000
1	0010	0010	0100	1000	0001+П
2	0100	0100	1000	0001+П	0010+П
3	1000	1000	0001+П	0010+П	0100+П

Если при суммировании четверичных разрядов возникает перенос, то подается сигнал на следующий по старшинству четверичный разряд. Если перенос возникает при суммировании предыдущих четверичных разрядов, то к результату суммирования текущего разряда добавляется единица, что соответствует простому циклическому сдвигу результата суммирования разряда в левую сторону на 1 бит, то есть в сторону, соответствующую увеличению

результата. При этом также может возникнуть ситуация переноса, когда результат уже был равен «1000» и прибавление к нему единицы (сдвиге влево на 1 бит) приводит к ситуации «0001» + Перенос.

Важно, что один разряд не может выработать сразу 2 переноса, так как крайняя ситуация, когда складываются числа «1000» и «1000» и с предыдущего разряда пришел сигнал переноса, дает результат «1000» + перенос. Перенос с самого старшего разряда (15-ого) приводит к переполнению, вследствие чего схемой вырабатывается сигнал ошибки, и работа АЛУ останавливается.

III. ВЫЧИТАНИЕ

Вычитание в данном коде сводится к суммированию числа в прямом коде (уменьшаемого) и числа в обратном коде (вычитаемого).

Для преобразования числа в обратный код (рис. 2) необходимо каждый разряд числа «транспонировать» относительно центра. Преобразование из обратного кода в прямой происходит так же.

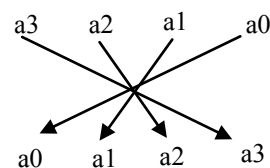


Рис. 2. Преобразование в обратный код

После преобразования вычитаемого в обратный код числа складываются обычным образом. Далее анализируется результат. Если при сложении уменьшаемого и вычитаемого возникает перенос со старшего разряда, то к получившемуся результату прибавляется единица, при этом результат является положительным числом. Если переноса со старшего разряда не возникает, то результат получен в обратном коде и преобразуется в прямой код, а знак результата записывается отрицательным. Получившийся результат подается на выход схемы.

IV. УМНОЖЕНИЕ

При умножении числа P1 и P2 берутся в прямых кодах, также результат умножения снимается с сумматора и записывается в P3 в прямом коде. Операция умножения состоит из 17 сложений:

1. 2 сложения на формирование P0100 и P1000 (удвоенного и утроенного значений P1, соответственно)

2. 15 сложений на формирование произведения двух чисел.

Операция умножения состоит из двух этапов:

1) Предварительный этап:

Формирование P0100 и P1000 (удвоенного и утроенного значений P1, соответственно).

Алгоритм умножения двух чисел заключается в последовательном формировании суммы частичных произведений. Эта сумма формируется каждый раз при сдвиге множителя (P2) на один четверичный разряд вправо и при последующем анализе кода, записанного в младшем разряде P2.

Предусмотрена прямая коммутация кода нуля для подачи его на сумматор. Код единицы выбирается на вход сумматора из P1.

Код двойки получается путем сложения значения P1 с самим собой в сумматоре. Результат сложения записывается в P0100.

Код тройки получается путем сложения значения P1 со значением Регистра 0100. Результат сложения записывается в Регистр 1000.

2) Основной этап:

Формирование произведения двух чисел. Начинается после формирования значений P0100 и P1000.

а) Анализируется младший разряд в P2. В четверичном разряде возможна запись четырех различных значений:

- 0 – соответствует код нуля;
- 1 – соответствует код P1;
- 2 – соответствует код P0100;
- 3 – соответствует код P1000.

Происходит сложение текущего содержимого P3 со значением, соответствующим коду младшего разряда в P2 (код нуля; код P1; код P0100; код P1000). При первом сложении в P3 находится код нуля.

На выходе сумматора находится 16-разрядное число, которое сдвигается вправо на один четверичный разряд (с потерей младшего разряда данного числа). Оно становится 15-разрядным. Это число записывается в P3.

Происходит сдвиг вправо на один четверичный разряд в P2. Происходит увеличение на единицу значения Счетчика Сдвигов.

б) Пункт 1 повторяется до тех пор, пока значение счетчика Сдвигов не достигнет значения 14.

При получении числа на выходе сумматора происходит проверка значения Счетчика Сдвигов. Если оно не равно 14, то происходит сдвиг числа вправо на один четверичный разряд (с потерей младшего разряда данного числа).

Если оно равно 14, то анализируется младший разряд полученного числа. Если в нем находится код 0 или 1, то происходит сдвиг числа вправо на один четверичный разряд (с потерей младшего разряда

данного числа). Если же в младшем разряде числа находится код 2 или 3, то происходит сначала сдвиг числа вправо на один четверичный разряд, а после этого происходит прибавление единицы к полученному результату.

V. ДЕЛЕНИЕ

При делении делимое (P1) подается на сумматор в прямом коде, а делитель (P2) подается на сумматор в обратном коде. Алгоритм операции деления основан на пошаговом вычитании делителя из делимого.

1) Происходит вычитание делителя (значения P2 в обратном коде) из делимого (значения P1 в прямом коде). После выполнения вычитания происходит проверка на наличие сигнала переполнения с 15-ого разряда в сумматоре. Необходимо отметить, что полученное число при первом вычитании должно быть отрицательным и находиться в обратном коде (т.е. должен отсутствовать сигнал переполнения с 15-ого разряда в сумматоре). Данное условие возникает из того положения, что всегда происходит деление меньшего числа на большее. В противном случае выдается сигнал ошибки. Если разность отрицательна (нет сигнала переполнения с 15-ого разряда), то полученное число находится в обратном коде. Необходимо произвести сдвиг делимого (P1) на один четверичный разряд влево, и сдвиг влево на один четверичный разряд в P3. Увеличивается значение Счетчика Сдвигов на единицу (в начале операции в нем находится код нуля). Если разность положительна, и это не первое вычитание (есть сигнал переполнения в 15-ом разряде в сумматоре), то полученное число находится в прямом коде, и для получения правильного значения необходимо прибавить к нему единицу. Полученный результат записывается из сумматора на место делимого в P1, также производится сдвиг влево на один бит внутри младшего разряда в P3. Отметим, что активным разрядом в P3 всегда является текущий младший разряд. Внутри каждого младшего разряда максимально возможно 3 сдвига на один бит влево, т.е. сигнал переполнения в 15-ом разряде может появляться максимально в трех подряд вычитаниях. На четвертом обязательным является отсутствие сигнала переполнения с 15-ого разряда.

2) Если нет сигнала переполнения с 15-ого разряда, то данный пункт алгоритма выполняется до тех пор, пока значение Счетчика Сдвигов не достигнет 15. В P3 находится конечный результат (частное), выдается сигнал конца операции, и АЛУ встает в режим ожидания.

Описанный выше алгоритм используется для случая деления чисел с остатком. Но нельзя исключать возможность деления двух чисел и без остатка.

Исходя из возможности деления чисел без остатка, после каждого вычитания (при отсутствии сигнала

переполнения с 15-ого разряда) производится проверка полученного результата на ноль. Данная проверка вводится из-за особенности представления чисел в системе кодирования 1-4. Результатом нуля после вычитания в обратном коде является значение тройки (1000) во всех разрядах.

При получении нулевого числа в результате вычитания необходимо произвести сдвиг влево на один бит внутри младшего разряда Р3. После этого при каждом последующем вычитании результат будет всегда являться отрицательным числом (не будет сигнала переполнения с 15-ого разряда), вследствие чего будут необходимы сдвиги влево на один четверичный разряд чисел в Р1 и Р3. Данные действия выполняются до тех пор, пока значение Счетчика Сдвигов не достигнет 15.

VI. ОЦЕНКА БЫСТРОДЕЙСТВИЯ И ЗАНИМАЕМОЙ ПЛОЩАДИ БЛОКА СУММИРОВАНИЯ

Блок суммирования в АЛУ является базовым, т.к. на нем завязаны все арифметические операции, следовательно, его показатели быстродействия и занимаемой площади являются наиболее важными.

Для оценки и сравнения интересующих показателей блока суммирования рассматривались 3 схемы, реализованных в базисе БМК по КМОП технологии 1,2 мкм:

1 - полученный в результате синтеза сумматор с системой кодирования 1 из 4, описанный на языке Verilog-HDL по предложенному ранее алгоритму;

2 - 32-х разрядный сумматор, построенный на однокоразрядных сумматорах;

3 - 32-х разрядный сумматор с ускоренным переносом, построенный на стандартных библиотечных элементах.

Для оценки быстродействия и проверки корректной работы блоков суммирования был описан на языке Verilog-HDL специальный тестовый модуль. В него входят автоматизированная схема сравнения для проверки правильности работы, а также элемент, который позволяет узнать быстродействие схемы. Быстродействие сумматора определяется в случае максимально возможного числа переносов в старший разряд. Площади и быстродействие блоков суммирования представлены в таблице 2.

Таблица 2

Занимаемая площадь и быстродействие блоков

№ схемы	Занимаемая площадь в ячейках	Быстродействие, нс
1	525	24,38
2	222	38,42
3	208	30,12

Как видно из таблицы, сумматор, реализованный по методу кодирования 1 из 4, работает быстрее, но его площадь более чем в два раза превосходит остальные варианты. Отметим, что если при сравнении площадей рассмотренных вариантов учитывать уровень достоверности и точности вычислений, то варианты 2 и 3 обладают значительно худшими характеристиками. Для достижения вариантами 2 и 3 уровня достоверности и точности вычислений, сравнимого с вариантом 1, необходимы дополнительные методы аппаратного контроля, что увеличит площадь устройства. Для уменьшения площади устройства, выполненного по варианту 1, необходимо применение дополнительных методов оптимизации схемы, а не простой синтез из поведенческого описания.

Отметим, что реализация устройств по методу кодирования 1 из 4 позволяет разработать систему команд со специальными операциями. Данная система команд позволяет исключить операционную систему и связанные с ней затраты на быстродействие, а также сократить затраты на требуемое повышение быстродействия оттранслированного ПО с языка высокого уровня и требуемое повышение быстродействия.

VII. ЗАКЛЮЧЕНИЕ

В статье представлены алгоритмы работы арифметических операций блока АЛУ с применяемой системой кодирования информации 1 из 4 с активным нулем. Проведена оценка быстродействия и занимаемой площади блока суммирования как базового блока, на котором основаны все арифметические операции. При этом анализ показал, что быстродействие схемы с системой кодирования информации 1 из 4 с активным нулем в 1,3-1,5 раз выше быстродействия других схем, а занимаемая площадь в 2,5 больше. Это означает, что в дальнейшем необходимо применять дополнительные методы оптимизации для уменьшения площади без потери быстродействия.

ЛИТЕРАТУРА

- [1] Хетагуров Я.А. Надежный, защищённый экономичный микропроцессор не двоичного кодирования для вычислительных систем реального времени, ISSN 1561-1531. Промышленные АСУ и Контроллеры. 2008. №07.