

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: 2008141155/08, 19.03.2007

(24) Дата начала отсчета срока действия патента:
19.03.2007

Приоритет(ы):

(30) Конвенционный приоритет:
17.03.2006 US 11/378,712

(43) Дата публикации заявки: 27.04.2010 Бюл. № 12

(45) Опубликовано: 10.04.2012 Бюл. № 10

(56) Список документов, цитированных в отчете о
поиске: RU 2236700 C2, 20.09.2004. US 5909573,
01.06.1999. US 2005/0283593 A1, 22.12.2005. US
2003/0222883 A1, 04.12.2003. US 2001/056531
A1, 27.12.2001.(85) Дата начала рассмотрения заявки РСТ на
национальной фазе: 17.10.2008(86) Заявка РСТ:
US 2007/064331 (19.03.2007)(87) Публикация заявки РСТ:
WO 2007/109631 (27.09.2007)

Адрес для переписки:

129090, Москва, ул. Б. Спасская, 25, стр.3,
ООО "Юридическая фирма Городисский и
Партнеры"

(72) Автор(ы):

ДИФФЕНДЕРФЕР Джеймс Норрис (US),
РИХЛИК Богуслав (US)

(73) Патентообладатель(и):

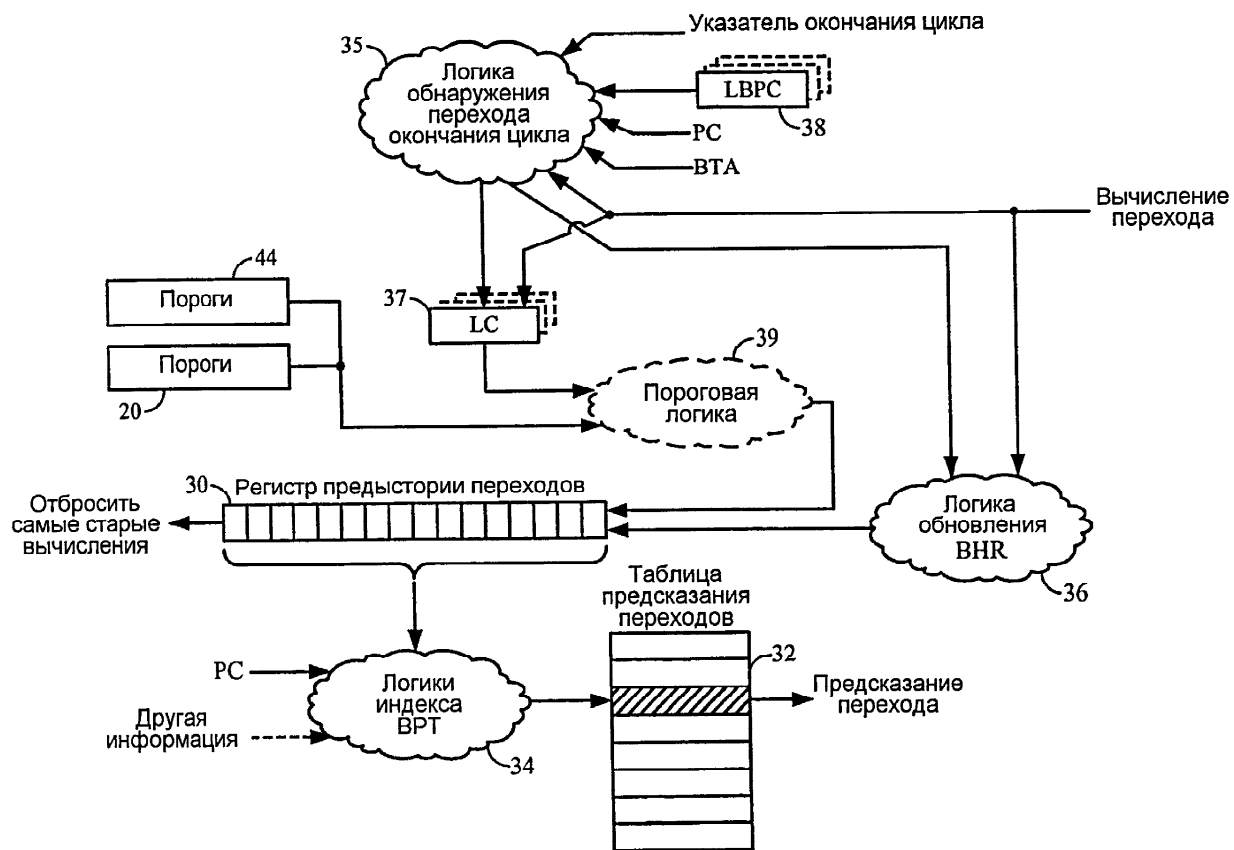
КВЭЛКОММ ИНКОРПОРЕЙТЕД (US)

(54) ПРЕДСТАВЛЕНИЕ ПЕРЕХОДОВ ЦИКЛА В РЕГИСТРЕ ПРЕДЫСТОРИИ ПЕРЕХОДОВ С
ПОМОЩЬЮ МНОЖЕСТВА БИТ

(57) Реферат:

Изобретение относится к процессорам и, в частности, к способу представления переходов цикла в регистре предыстории переходов с помощью множества бит. Техническим результатом является улучшение производительности процессора и сохранение мощности. Способ включает: определение, что команда условного перехода является командой перехода завершения цикла; подсчитывание общего числа итераций цикла, завершеного посредством команды перехода

завершения цикла; сохранение многоразрядной величины в регистре предыстории переходов (BHR), причем эта величина указывает общее число итераций цикла после завершения цикла; и индексацию таблицы устройства предсказания переходов (BPT) с помощью, по меньшей мере, BHR после завершения цикла для получения предсказания перехода для команды перехода завершения цикла, причем предсказание перехода основано, по меньшей мере, частично на предыстории переходов. 4 н. и 23 з.п. ф-лы, 4 ил.



Фиг.4



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(12) ABSTRACT OF INVENTION(21)(22) Application: **2008141155/08, 19.03.2007**(24) Effective date for property rights:
19.03.2007

Priority:

(30) Priority:
17.03.2006 US 11/378,712(43) Application published: **27.04.2010 Bull. 12**(45) Date of publication: **10.04.2012 Bull. 10**(85) Commencement of national phase: **17.10.2008**(86) PCT application:
US 2007/064331 (19.03.2007)(87) PCT publication:
WO 2007/109631 (27.09.2007)

Mail address:

**129090, Moskva, ul. B. Spasskaja, 25, str.3, OOO
"Juridicheskaja firma Gorodisskij i Partnery"**

(72) Inventor(s):

**DIFFENDERFER Dzhejms Norris (US),
RIKhLIK Boguslav (US)**

(73) Proprietor(s):

KVEhLKOMM INKORPOREJTED (US)**(54) PRESENTATION OF CYCLE TRANSITIONS IN TRANSITIONS PREHISTORY REGISTER USING MULTIPLE BITS**

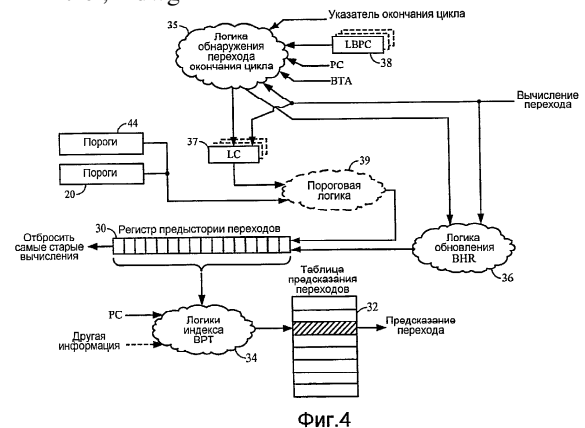
(57) Abstract:

FIELD: information technology.

SUBSTANCE: method involves: determination that conditional branch instruction is the branch instruction of cycle termination; counting total number of cycle iterations, which cycle has been terminated by means of cycle termination branch instruction; saving multidigit value in branch history register (BHR), where this value specifies total number of cycle iterations, after cycle termination; and indexing the table of branch prediction unit (BPT) using at least BHR after cycle termination to obtain branch prediction for cycle termination branch instruction. In this method branch prediction is based at least partially on history of branches.

EFFECT: improvement of processor performance and saving power.

27 cl, 4 dwg



Область техники, к которой относится изобретение

Настоящее изобретение в целом относится к процессорам и, в частности, к способу представления переходов цикла в регистре предыстории переходов с помощью множества бит.

Уровень техники

Микропроцессоры выполняют вычислительные задачи в большом разнообразии приложений. Улучшенная производительность процессора почти всегда является желательной, чтобы предусматривать более быструю работу и увеличенные функциональные возможности посредством изменений программного обеспечения. В широко распространенных встроенных приложениях, таких как портативные электронные устройства, также является желательным сохранение мощности.

Широко распространенные современные процессоры используют конвейерную архитектуру, в которой последовательные команды, причем каждая имеет множество шагов выполнения, совмещают при выполнении. Для максимальной производительности команды должны непрерывно проходить через конвейер. Любая ситуация, которая заставляет команды задерживаться в конвейере, вредно влияет на производительность. Если команды должны быть сброшены из конвейера, а затем повторно выбраны, могут страдать как производительность, так и потребление мощности.

Обычно все реальные программы включают в себя команды условного перехода, фактическое поведение перехода которых обычно не известно, до тех, пока команда не будет вычислена глубоко в конвейере. Чтобы исключить задержку, которая может происходить в результате ожидания фактического вычисления команды перехода, широко распространенные современные процессоры используют некоторый вид предсказания перехода, посредством чего поведение перехода команд условного перехода предсказывают заблаговременно в конвейере. На основании предсказанного вычисления перехода процессор спекулятивно выбирает (предварительно выбирает) и выполняет команды из предсказанного адреса, либо из целевого адреса перехода (если взят переход, который предсказали), либо из следующего последовательного адреса после команды перехода (если не взят переход, который предсказали). Когда фактическое поведение перехода определено, если переход был предсказан неверно, спекулятивно выбранные команды должны быть сброшены из конвейера и выбраны новые команды из правильного следующего адреса. Предварительная выборка команд в ответ на неправильное предсказание перехода неблагоприятно влияет на производительность процессора и потребление мощности. Следовательно, желательно увеличение точности предсказания перехода.

Известные способы предсказания переходов включают в себя как статические, так и динамические предсказания. Вероятное поведение некоторых команд перехода может быть статически предсказано с помощью программатора и/или компилятора. Одним примером является подпрограмма проверки ошибок. Обычный код выполняется правильно, и ошибки являются редкими. Следовательно, команда перехода, осуществляющая функцию “переход по ошибке”, будет вычислять “не взят” очень высокий процент времени. Такая команда может включать в себя бит статического предсказания перехода в коде операции, установленный с помощью программатора или компилятора со знанием обычного вероятного результата условия перехода.

Динамическое предсказание обычно основано на предыстории вычисления переходов (и в некоторых случаях на предыстории точности предсказания переходов)

предсказываемой команды перехода и/или других команд перехода в той же самой программе. Обширный анализ фактической программы показывает, что шаблоны вычисления переходов недавнего прошлого могут быть полезным указателем вычисления будущих команд перехода.

Один известный вид динамического предсказания переходов, изображенный на фиг.1, использует регистр предыстории переходов (BHR) 100, чтобы сохранять последние n вычислений переходов. В простом осуществлении BHR 30 содержит сдвиговый регистр. Обычный результат последнего вычисления перехода сдвигают (например, 1 указывает взятый, а 0 указывает не взятый), причем самое старое вычисление в регистре замещают. Процессор может поддерживать локальный BHR 100 для каждой команды перехода. В качестве альтернативы (или дополнительно), BHR 100 может содержать вычисления недавнего прошлого всех команд условного перехода, иногда известный в данной области техники как глобальный BHR или GHR. Как использован в настоящем описании, BHR относится как к локальному, так и глобальному регистру предыстории переходов.

Как изображено на фиг.1, BHR 100 может индексировать таблицу устройства предсказания переходов (BPT) 102, который, к тому же, является локальным или глобальным. BHR 100 может индексировать BPT 102 непосредственно или может быть объединен с другой информацией, такой как программный счетчик (PC) команд перехода в логике 104 индекса BPT. Дополнительно могут быть использованы другие входы в логики 104 индекса BPT. Логика 104 индекса BPT могут конкатенировать (соединять) входы (известные в данной области техники как gselect), выполнять операцию “исключающего ИЛИ” (gshare) над входами, выполнять хэш-функцию или комбинировать или преобразовывать входы множеством способов.

В качестве одного примера, BPT 102 может содержать множество счетчиков насыщения, MSB которых служат в качестве двухрежимных устройств предсказания переходов. Например, каждый элемент таблицы может содержать двухбитовый счетчик, который допускает одно из четырех состояний, причем каждому состоянию назначено взвешенное значение предсказания, такое как:

11 - Взят сильно предсказанный.

10 - Взят слабо предсказанный.

01 - Не взят слабо предсказанный.

00 - Не взят сильно предсказанный.

Счетчик увеличивается каждый раз, когда соответствующая команда перехода вычисляет “взят” и уменьшается каждый раз, когда команда вычисляет “не взят”. MSB счетчика является двухрежимным устройством предсказания переходов, он будет предсказывать, что переход либо будет взят, либо не будет взят, независимо от силы или веса лежащего в основе предсказания. Счетчик насыщения уменьшает ошибку предсказания нечастого направления вычисления перехода. Переход, который постоянно вычисляет одно направление, будет насыщать счетчик. Нечастое вычисление другого направления будет изменять значение счетчика (и силу предсказания), но не бимодальное значение предсказания. Таким образом, нечастое вычисление будет неправильно предсказывать только один раз, не дважды. Таблица счетчиков насыщения является только иллюстративной, вообще BHR может индексировать таблицу, содержащую множество механизмов предсказания переходов.

Независимо от механизма предсказания переходов, используемого в BPT 102, BHR 100, либо отдельно, либо в комбинации с другой информацией, такой как PC команд перехода, индексирует BPT 102 таким образом, чтобы получить предсказания

переходов. С помощью сохранения предыдущих вычислений переходов в BHR 100 и использования вычислений в предсказании перехода, причем предсказываемую команду перехода коррелируют с прошлым поведением перехода, его собственным прошлым поведением в случае локального BHR 100 и поведением других команд

перехода в случае глобального BHR 100. Корреляция является ключом для точных предсказаний переходов, по меньшей мере, в случае часто повторяемого кода.

Следует заметить, что фиг.1 изображает вычисления переходов, сохраняемые в BHR 100, то есть фактическое вычисление команды условного перехода, которое может быть известно только глубоко в конвейере, например на этапе выполнения конвейера. Несмотря на то, что это является окончательным результатом, на практике обычные высокопроизводительные процессоры запоминают предсказанное вычисление перехода из BPT 102 в BHR 100 и позднее корректируют BHR 100 как часть операции восстановления ошибочного предсказания, если предсказание оказывается ошибочным. Изображенные фигуры не отражают этот признак осуществления ради ясности.

Обычной структурой программы, которая может уменьшить устройства предсказания ветвления, использующего BHR 100, является цикл. Цикл заканчивается командой условного перехода, которая проверяет условие завершения цикла, такое как, достигла ли индексная переменная, которую увеличивают каждый раз через цикл, значения завершения цикла. Если нет, выполнение переходит назад в начало цикла для другой итерации и к вычислению другого условного перехода завершения цикла.

Если цикл выполняется посредством большого числа итераций, “взятые” переходы назад команды перехода завершения цикла частично или полностью насыщают BHR 100. Когда число итераций цикла станет равно или превысит разрядность BHR 100, в конце цикла n-битовый BHR 100 будет содержать точно n-1 единиц (взяты), за которыми следует один ноль (не взят), соответствующий длинной последовательности взятых вычислений переходов, получающихся в результате итераций цикла, и заканчивающийся одним не взятым вычислением перехода, когда цикл завершается. Это фактически разрушает эффективность BHR 100, так как все корреляции с предыдущими вычислениями переходов (либо для локального, либо для глобального BHR) потеряны. В этом случае BHR 100, возможно, отобразится в один и тот же элемент BPT 102 для данной команды перехода (в зависимости от других входов в логические схемы 104 индекса BPT), а не в элемент, содержащий предсказание перехода, которое отражает корреляцию команды перехода с предыдущими вычислениями переходов.

Кроме того, насыщенный BHR 100 может увеличить эффект наложения в BPT 102. То есть все команды перехода после циклов с обычными итерациями будут отображены в один и тот же элемент BPT 102, если BHR 100 непосредственно индексирует BPT 102. Даже когда BHR 100 объединяют с другой информацией, вероятность наложения увеличивается. Это неблагоприятно влияет на точность предсказания не только для команды перехода после цикла, но также для всех команд перехода, которые накладываются на свой элемент в BPT 102.

Если цикл выполняется посредством меньшего числа итераций, чем разрядность BHR 100, BHR 100 не насыщается, и сохраняется некоторая предыстория предыдущего вычисления переходов. Однако биты, представляющие предысторию предыдущего вычисления переходов, замещают в BHR 100 многочисленными “взятыми” результатами команды перехода завершения цикла. В частности, когда число итераций цикла изменяется, это имеет два вредных воздействия на предсказание

перехода. Во-первых, команда перехода будет отображена в существенно большее число элементов в BPT 102, чтобы собрать данные об одной и той же корреляции с предыдущими вычислениями переходов, при этом требуется большая BPT 102, чтобы поддерживать ту же самую точность для того же самого числа команд перехода, чем
 5 потребовалось бы без перехода завершения цикла, влияющего на BHR 30. Во-вторых, устройству предсказания переходов в BPT 102 потребуется больше времени, чтобы “обучиться”, при этом увеличивается объем программы, которая должна выполняться до того, как BPT 102 начнет предоставлять точные предсказания переходов.

10 Заявка на патент под № 11/066508, права на которую переданы владельцу настоящей заявки и включенная в настоящее описание в качестве ссылки, предлагает запретить обновление BHR 100 для команд перехода завершения цикла. Это устраняет общеизвестные вредные влияния насыщения или частичного насыщения BHR 100 на
 15 точность предсказания перехода. Однако при этом не удастся собрать и использовать корреляции, которые могут существовать между поведением цикла и последующим вычислением перехода.

В широко распространенных приложениях вычисление команды перехода может быть коррелировано с числом итераций предшествующего цикла. Например, научная
 20 программа может собирать данные о точках в цикле и после цикла переходить к подпрограмме статистического анализа, в которой анализируют собранные данные. Однако, если цикл повторяется относительно малое число раз, при этом собирают данные о немногих точках и выдают небольшую выборку, статистический анализ может быть ненадежным и может быть пропущен. В этом случае вычисление перехода
 25 условной команды в подпрограмму статистического анализа сильно коррелировано с числом итераций цикла сбора данных.

В другом примере приложение может использовать структуру цикла для поиска в списке, в журнале транзакций, файле предыстории или в подобной структуре данных.
 30 Если элемент, соответствующий параметрам поиска, часто появляется в списке, потребуется относительно мало итераций цикла, чтобы найти элемент. Следовательно, нечастое появление элемента может требовать большого числа итераций посредством цикла поиска. Вычисление следующей команды перехода может быть коррелировано с частотой, с которой конкретный элемент появляется в списке, и, следовательно,
 35 коррелировано с числом итераций цикла, необходимых, чтобы найти элемент.

При запрете обновления BHR 100 в ответ на вычисления команды перехода завершения цикла не удастся собрать данные о любой корреляции между числом итераций цикла и поведением перехода следующей команды перехода. С другой
 40 стороны, поддержание полной предыстории вычислений команды перехода завершения цикла полностью или частично насыщает BHR 100, при этом теряют корреляцию с вычислениями команд перехода до цикла.

Сущность изобретения

В ответ на свойство команды условного перехода, связанной с циклом, такое как
 45 свойство, указывающее, что переход является переходом завершения цикла, поддерживают подсчет числа итераций цикла, и в BHR сохраняют многоразрядную величину, указывающую подсчет итераций цикла. В одном варианте осуществления многоразрядная величина может содержать фактический подсчет цикла, причем в
 50 этом случае число бит является переменным. В другом варианте осуществления число бит является фиксированным (например, два), и подсчеты итераций цикла отображают в многоразрядную величину с помощью сравнения с порогом. Отдельные подсчеты итераций могут быть поддержаны для вложенных циклов, и многоразрядная

величина, сохраненная в BHR, может указывать подсчет итераций цикла только для внутреннего цикла, только для внешнего цикла или для обоих циклов.

Один вариант осуществления относится к способу предсказания перехода.

Идентифицируют свойство команды перехода, связанной с циклом. В ответ на свойство многоразрядную величину сохраняют в BHR, причем многоразрядная величина указывает на число итераций цикла, связанного с командой перехода.

Другой вариант осуществления относится к процессору. Процессор включает в себя устройство предсказания переходов, действующее с возможностью предсказания вычисления команд условного перехода, и конвейер выполнения команд, действующий с возможностью спекулятивного выбора и выполнения команд на основании предсказания из устройства предсказания переходов. Процессор также включает в себя BHR, действующий с возможностью сохранения вычисления команд условного перехода, и счетчик цикла, действующий с возможностью подсчета числа итераций цикла кода. Процессор дополнительно включает в себя схему управления, действующую с возможностью сохранения в BHR многоразрядной величины, указывающей число итераций цикла, связанного с командой условного перехода.

Краткое описание чертежей

Фиг.1 - функциональная блок-схема схемы устройства предсказания переходов, известного из предшествующего уровня техники.

Фиг.2 - функциональная блок-схема процессора.

Фиг.3 - блок-схема последовательности этапов способа выполнения команды перехода.

Фиг.4 - функциональная блок-схема схемы устройства предсказания переходов, включающей в себя один или более регистров PC последнего перехода и счетчиков циклов.

Подробное описание вариантов осуществления

Фиг.2 изображает функциональную блок-схему процессора 10. Процессор 10 выполняет команды в конвейере 12 выполнения команд в соответствии с логиками 14 управления. В некоторых вариантах осуществления конвейер 12 включает в себя различные регистры или схемы-защелки 16, организованные в каскадах конвейера, и одно или более арифметико-логических устройств (ALU) 18. Файл 20 регистра общего назначения (GPR) обеспечивает регистры, содержащие вершину иерархии памяти.

Конвейер 12 выбирает команды из кэш-памяти 22 команд (кэш-памяти I), причем трансляцией и разрешениями адресов памяти управляют с помощью буфера ассоциативной трансляции стороны команд (ITLB) 24. Когда команды условного перехода декодируют заблаговременно в конвейере 12, устройство 26 предсказания переходов предсказывает поведение перехода и предоставляет предсказание в устройство 28 предварительной выборки команд. Устройство 28 предварительной выборки команд спекулятивно выбирает команды из кэш-памяти I 22 в целевом адресе перехода, вычисленном в конвейере 12 для предсказаний “взятых” переходов, или в следующем последовательном адресе для переходов, предсказанных как “не взятые”. В любом случае предварительно выбранные команды загружают в конвейер 12 для спекулятивного выполнения.

Устройство 26 предсказания переходов включает в себя регистр предыстории переходов (BHR) 30, таблицу устройства предсказания переходов (BPT) 32, логику 34 индекса BPT и логику 36 обновления BHR. Устройство 26 предсказания переходов дополнительно может включать в себя один или более регистров 38 PC последнего перехода и один или более счетчиков 37 циклов, предоставляющих входные данные в

логики 36 обновления BHR.

Данные берут из кэш-памяти 40 данных (кэш-памяти D), причем трансляцией и разрешениями адресов памяти управляют с помощью главного буфера ассоциативной трансляции (TLB) 42. TLB 42 может быть специализированным TLB данных или может
 5 содержать интегрированный TLB, который управляет трансляциями и разрешениями адресов как для команд, так и для данных. Кроме того, в различных вариантах осуществления процессор 10, кэш-память I 22 и кэш-память D 40 могут быть интегрированы или объединены. Неудачи поиска в кэш-памяти I 22 и/или кэш-
 10 памяти D 40 вызывают доступ к главной (вне кристалла) памяти 44 под управлением интерфейса 46 памяти.

Процессор 10 может включать в себя интерфейс 48 ввода/вывода (I/O), управляющий доступом к разным периферийным устройствам 50, 52. Специалисты в данной области техники поймут, что возможны многочисленные варианты
 15 процессора 10. Например, процессор 10 может включать в себя кэш-память второго уровня (L2) для любой или обеих кэш-памятей I и D 22, 40. Кроме того, один или более функциональных блоков, изображенных в процессоре 10, могут быть не включены в конкретный вариант осуществления.

Многоразрядные указания итераций цикла

В соответствии с одним или более вариантами осуществления точность предсказания перехода повышается с помощью корреляции вычисления перехода, как с предыдущими вычислениями переходов, так и с подсчетом итераций предыдущих
 25 циклов. Это выполняют с помощью сохранения многоразрядной величины, указывающей подсчет (подсчеты) итераций цикла в BHR 30. С помощью отображения возможно больших подсчетов итераций цикла в относительно малое число бит сохраняют характеристическую информацию о цикле, в то же время препятствуют переходам завершения цикла полностью или частично насыщать один или более BHR
 30 30 в устройстве 26 предсказания переходов.

Этот процесс изображен в качестве блок-схемы последовательности этапов на фиг.3. Команду выбирают и декодируют (этап 52). Если команда не является командой условного перехода (этап 53), она продолжается в конвейере, и процесс возвращается к выбору следующей команды (этап 52). Если команда является
 35 командой условного перехода (этап 53), выполняют определение, является ли переход переходом завершения цикла (этап 54). Если нет, тогда при вычислении переходов обновляют BHR 30 (такими как в каскаде выполнения конвейера), чтобы записать вычисления перехода с использованием одного бита (этап 56), т. е. вычислена ли команда перехода как “взятая” или “не взятая”. Затем выполнение продолжается (этап 64) в целевом адресе перехода или в следующем последовательном адресе, соответственно. Если переход является переходом завершения цикла (этап 54), увеличивают подсчет итераций цикла (этап 58). Когда цикл завершается (этап 60), многоразрядную величину, указывающую число итераций цикла, записывают в BHR 30
 40 (этап 62). Затем очищают (этап 63) счетчик итераций цикла, и выполнение продолжается (этап 64) в следующем последовательном адресе.

Как запрос (этап 54), то есть идентификация команды перехода как команды перехода завершения цикла, так и поддержание подсчета итераций цикла (этап 58) могут быть выполнены множеством способов. В одном варианте осуществления увеличивают один или более счетчиков циклов (LC) 37, когда команда условного
 50 перехода, которую определяют как переход завершения цикла, вычисляет “взят”. Значение LC 37 или значение, полученное из него (как обсуждено более подробно в

настоящем описании), записывают в BHR 30, когда переход завершения цикла вычисляет “не взят”, указывающее выход из цикла. В этот момент времени также очищают LC 37.

Идентификация команд перехода завершения цикла

5 Циклы повторяются с помощью перехода назад от конца цикла в начало цикла. В соответствии с одним вариантом осуществления допускают, что каждая команда условного перехода с целевым адресом перехода, меньшим чем адрес команды перехода или PC, то есть перехода назад, является командой перехода завершения
10 цикла. Этот способ идентификации переходов завершения цикла предлагает простоту. Как изображено на фиг.4, PC команды перехода сравнивают с целевым адресом перехода (BTA) в логиках 35 обнаружения переходов завершения цикла, когда команду перехода фактически вычисляют в конвейере. Если $BTA < PC$, счетчик
15 цикла (LC) 37 увеличивают. Этот вариант осуществления требует сравнение адреса, когда определен целевой адрес перехода, и может проигрывать от некоторой неточности, так как не все обратные переходы являются переходами завершения цикла.

Другим способом обнаружения перехода завершения цикла является распознавание
20 повторяемого выполнения одной и той же команды перехода. В одном варианте осуществления, как изображено на фиг.4, регистр 38 PC последнего перехода (LBPC) сохраняет PC последней выполненной команды перехода. В случае простого цикла, если PC команды перехода соответствуют LBPC 38 в логике 35 обнаружения перехода завершения цикла, то есть команде перехода, которая была последней вычисленной
25 командой перехода, считают, что команда перехода является командой перехода завершения цикла, и увеличивают LC 37.

Код, который включает в себя команды условного перехода внутри цикла, может усложнить обнаружение перехода завершения цикла. Например, общеизвестные
30 алгоритмы, предназначенные для поиска массива или связанного списка, создают цикл. Переход внутри цикла проверяет совпадение и выходит из цикла заблаговременно, если совпадение найдено, в то время как переход завершения цикла проверяет конец массива или связанного списка. В этом случае один LBPC 38 не может обнаружить переход завершения цикла, так как в противном случае он будет
35 перезаписан с помощью PC двух переходов. Таким образом, в одном варианте осуществления могут быть обеспечены два или более регистров LBPC 38 (как изображено на фиг.4), причем PC последовательно вычисленных команд перехода сохраняют в соответствующих регистрах LBPC ($LBPC_0, LBPC_1, \dots, LBPC_M$) 38. Это
40 предусматривает обнаружение переходов окончания циклов даже при наличии других условных переходов внутри цикла.

Команды перехода завершения цикла также могут быть статически отмечены с помощью компилятора или ассемблера. В одном варианте осуществления компилятор генерирует конкретный тип команды перехода, которую используют только для
45 переходов завершения циклов, например “BRLP”. Команду BRLP распознают, и LC 37 увеличивают каждый раз, когда команда BRLP вычисляет как взятый в каскаде конвейера исполнения. В другом варианте осуществления компилятор или ассемблер может вставить указание перехода завершения цикла в команду перехода, например с
50 помощью установки одного или более предварительно определенных бит в код операции команды. Биты перехода завершения цикла обнаруживают и LC 37 увеличивают, когда команда перехода вычисляет как взятый в каскаде конвейера исполнения. В большинстве случаев указатель завершения цикла может быть

предоставлен в логике 35 обнаружения перехода завершения цикла. Статичная идентификация переходов завершения циклов уменьшает аппаратное обеспечение и вычислительную сложность с помощью перемещения функции идентификации завершения цикла в компилятор или ассемблер.

5 Генерация и сохранение указателя числа итераций цикла

Независимо от способа, используемого, чтобы идентифицировать команду перехода завершения цикла (либо для внешнего, либо для внутреннего цикла), в соответствии с одним или более вариантами осуществления указание числа раз, 10 которые повторяется цикл, сохраняют в BHR 30, чтобы использовать любую корреляцию между числом итераций цикла и последующим вычислением команды условного перехода.

В одном варианте осуществления фактический подсчет итераций цикла сохраняют в BHR 30. То есть подсчет множества бит, например из LC 37, сдвигают в BHR 30, 15 замещая предыдущие вычисления переходов в MSB BHR 30. Для очень больших циклов этот подсчет потребует $\log_2(n)$ битов для итераций цикла, замещающих соответствующее число предыдущих вычислений переходов. Однако даже для больших подсчетов итераций цикла двоичное представление является существенным сжатием поведения перехода завершения цикла, сохраняют в регистрах предыстории переходов предшествующего уровня техники (т. е. $n-1$ единиц, за которыми следует 20 один ноль). В одном варианте осуществления подсчет итераций накапливают в LC 37 и сдвигают в BHR 30. В другом варианте осуществления (не изображен), по меньшей мере, младшие k бит BHR 30 могут быть осуществлены как двухрежимный сдвиговый регистр/счетчик, и подсчет итераций цикла может быть увеличен непосредственно 25 в BHR 30 (причем предыдущее содержимое BHR 30 сдвигают влево, когда подсчет увеличивается).

В другом варианте осуществления фиксированное многоразрядное указание числа итераций цикла сохраняют в BHR 30. Например, два бита могут быть выделены для 30 записи указания числа итераций цикла, со следующими значениями:

11 - очень большое число итераций цикла,

10 - большое число итераций цикла,

01 - среднее число итераций цикла,

35 00 - малое число итераций цикла.

В этом примере три значения порога, между малым и средним (00->01), между средним и большим (01->10) и между большим и очень большим (10->11) используют, чтобы отобразить фактический подсчет итераций цикла в двух битовое представление, 40 как указано с помощью пороговой логики 39. В одном варианте осуществления значения порога могут быть фиксированы для данного осуществления процессора, например для определенных встроенных приложений, в которых обычно известны общие размеры цикла. В другом варианте осуществления отображение может быть масштабированным, причем значения порога считывают из GPR 20, таблицы в 45 памяти 44 или тому подобного. В общеизвестных приложениях фиксированное многоразрядное указание числа итераций цикла может предоставлять достаточное указание подсчета итераций цикла, чтобы предусматривать существенную корреляцию с поведением вычисления перехода, а также одновременно ограничивают число 50 предыдущих вычислений переходов, замещенных из BHR 30 переходами завершения циклов.

Конечно, фактические подсчеты итераций цикла могут быть отображены в любое фиксированное число бит с соответствующим числом значений порога, например три

бит и семь порогов, четыре бит и пятнадцать порогов и т.д., чтобы достичь желаемого или требуемого баланса между точностью подсчета итераций цикла и смещением вычислений переходов в BHR 30.

Вложенные циклы

5 Цикл может содержать один или более вложенных, или внутренних циклов. Таким образом, в одном варианте осуществления обеспечено множество счетчиков LC ($LC_0, LC_1, \dots, LC_M$), которые могут соответствовать числу регистров LBPC 38. LC 37 увеличивают после взятого вычисления и совпадения с соответствующим $LBPC_M$ 38, чтобы поддерживать подсчеты итераций вложенных циклов.

10 С помощью сохранения многообразного указания числа итераций внутреннего цикла каждый раз, когда существует внутренний цикл, BHR 30 может стать частично или полностью насыщенным ко времени, когда внешний цикл повторяется относительно малое число раз. Таким образом, в одном варианте осуществления, если определяют, что цикл является внутренним циклом, подсчеты следующих итераций после первоначального выполнения внутреннего цикла могут быть запрещены из обновления BHR 30. В этом варианте осуществления ко времени, когда существует внешний цикл, BHR 30 включит в свой состав указание подсчета итераций внутреннего цикла только для первой итерации внешнего цикла и указание подсчета итераций внешнего цикла.

В другом варианте осуществления подсчет итераций внутреннего цикла может быть отвергнут с помощью перезаписи его значения в BHR 30 с помощью указания подсчета итераций внешнего цикла. Это сохраняет наибольшее число предыдущих вычислений переходов в BHR 30, в то же время допуская корреляцию следующих вычислений переходов с подсчетом итераций внешнего цикла.

В другом варианте осуществления подсчет итераций внешнего цикла может быть отвергнут с помощью запрещения обновления BHR 30, когда существует внешний цикл. Это опять сохраняет наибольшее число предыдущих вычислений переходов в BHR 30, в то же время допуская корреляцию следующих вычислений переходов с подсчетом итераций внутреннего цикла для получения предсказания следующего вычисления перехода в некоторых приложениях.

35 Команда условного перехода имеет обычные свойства, включая, например, PC или адрес команды перехода, тип команды и наличие или отсутствие бит указателя в коде операции. Как использованы в настоящем описании, свойства операции перехода и/или свойства программы, которые относятся к переходу, считают свойствами команды перехода. Например, соответствует ли PC команды перехода содержанию одного или более регистров 38 LBPC и передают ли целевой адрес перехода вперед или назад относительно PC команды перехода являются свойствами команды перехода.

Несмотря на то что настоящее раскрытие описано в настоящей заявке относительно конкретных признаков, аспектов и вариантов осуществления раскрытия, будет понятно, что многочисленные изменения, модификации и другие варианты осуществления возможны в широких рамках настоящего раскрытия и, таким образом, считается, что все изменения, модификации и варианты осуществления находятся в рамках раскрытия. Таким образом, настоящие варианты осуществления должны быть поняты во всех аспектах как иллюстративные, а не ограничивающие, и предполагается, что все изменения в рамках смысла и эквивалентности прилагаемой формулы изобретения включены в нее.

Формула изобретения

1. Способ предсказания перехода, содержащий этапы, на которых:
определяют, что команда условного перехода является командой перехода
завершения цикла;

подсчитывают общее число итераций цикла, завершеного посредством команды
перехода завершения цикла;

сохраняют многоразрядную величину в регистре предыстории переходов (BHR),
причем эта величина указывает общее число итераций цикла, после завершения цикла;
и

индексируют таблицу устройства предсказания переходов (BPT) с помощью, по
меньшей мере, BHR после завершения цикла для получения предсказания перехода для
команды перехода завершения цикла, причем предсказание перехода основано, по
меньшей мере, частично на предыстории переходов.

2. Способ по п.1, в котором определение того, что команда условного перехода
является командой перехода завершения цикла, содержит этап, на котором
обнаруживают, что условный переход является обратным относительно направления
хода выполнения программы.

3. Способ по п.1, в котором определение того, что команда условного перехода
является командой перехода завершения цикла, содержит этап, на котором
обнаруживают, что РС команды перехода соответствует содержимому регистра РС
последнего перехода (LBPC), сохраняющего РС команды последнего перехода, чтобы
обновить BHR.

4. Способ по п.1, в котором определение того, что команда условного перехода
является командой перехода завершения цикла, содержит этап, на котором
обнаруживают, что РС команды перехода соответствует содержимому любого из
множества регистров LBPC, сохраняющих РС последнего множества команд перехода,
чтобы обновить BHR.

5. Способ по п.1, в котором команда перехода является уникальной командой
перехода, генерируемой с помощью компилятора для завершения циклов.

6. Способ по п.1, в котором код операции команды перехода включает в себя один
или более бит, указывающих, что она является командой перехода завершения цикла.

7. Способ по п.1, в котором сохранение многоразрядной величины в BHR содержит
этап, на котором сохраняют предварительно определенное число бит в BHR.

8. Способ по п.7, дополнительно содержащий этап, на котором определяют
значение предварительно определенного числа бит в соответствии с фиксированным
отображением числа итераций цикла в многоразрядную величину.

9. Способ по п.7, дополнительно содержащий этап, на котором определяют
значение предварительно определенного числа бит в соответствии с масштабируемым
отображением числа итераций цикла в многоразрядную величину.

10. Способ по п.9, дополнительно содержащий этап, на котором считывают
множество порогов для определения масштабируемого отображения числа итераций
цикла в многоразрядную величину.

11. Способ по п.1, в котором сохранение многоразрядной величины в BUR
содержит этап, на котором сохраняют переменное число бит в BHR, причем число бит
изменяют в ответ на общее число итераций цикла.

12. Способ по п.11, дополнительно содержащий этап, на котором сохраняют
итерации цикла в младшем значащем множестве бит BHR.

13. Способ по п.11, дополнительно содержащий этап, на котором подсчитывают
итерации цикла в счетчике цикла и передают значение счетчика цикла в BHR, когда

цикл завершается.

14. Способ по п.1, в котором этап определения содержит этап, на котором обнаруживают первую команду перехода завершения цикла, связанную с первым циклом, и вторую команду перехода завершения цикла, связанную со вторым циклом, причем первый цикл вложен во второй цикл.

15. Способ по п.14, в котором сохранение многоразрядной величины в BHR содержит этап, на котором сохраняют многоразрядную величину, указывающую число итераций первого цикла, в BHR, и дополнительно содержит этап, на котором сохраняют многоразрядную величину, указывающую число итераций второго цикла, в BHR.

16. Способ по п.14, в котором сохранение многоразрядной величины в BHR содержит этап, на котором сохраняют только многоразрядную величину, указывающую число итераций первого цикла, в BHR.

17. Способ по п.14, в котором сохранение многоразрядной величины в BHR содержит этап, на котором сохраняют только многоразрядную величину, указывающую число итераций второго цикла, в BHR.

18. Процессор, содержащий

устройство предсказания переходов, действующее с возможностью предсказания вычисления команд условного перехода;

конвейер выполнения команд, действующий с возможностью спекулятивного выбора и выполнения команд на основании предсказания из устройства предсказания переходов;

регистр предыстории переходов (BHR) в устройстве предсказания переходов, действующий с возможностью сохранения, по меньшей мере, вычисления команд условного перехода;

счетчик цикла (LC) в устройстве предсказания переходов, действующий с возможностью подсчета общего числа итераций цикла кода;

схему управления в устройстве предсказания переходов, действующую с возможностью сохранения в BHR многоразрядной величины, указывающей общее число итераций цикла, связанного с командой условного перехода, после завершения цикла; и

таблицу устройства предсказания переходов в устройстве предсказания переходов, индексируемую, по меньшей мере, посредством BHR, после завершения цикла, действующую с возможностью выдачи предсказания перехода для команды условного перехода, связанной с циклом, на основе, по меньшей мере, частично предыстории переходов.

19. Процессор по п.18, дополнительно содержащий регистр РС последнего перехода (LBPC), действующий с возможностью сохранения РС команды условного перехода, и в котором схема управления определяет команду условного перехода, связанную с циклом, если РС команда перехода соответствует содержимому регистра LBPC.

20. Процессор по п.19, дополнительно содержащий два или более регистров LBPC и соответствующие два или более LC, причем первый LBPC действует с возможностью сохранения РС команды условного перехода, связанной с первым циклом, и при этом первый LC действует с возможностью поддержания подсчета итераций первого цикла, а второй LBPC действует с возможностью сохранения РС команды условного перехода, связанной со вторым циклом, и при этом второй LC действует с возможностью поддержания подсчета итераций второго цикла, причем первый цикл

является вложенным во второй цикл.

21. Процессор по п.20, в котором схема управления действует с возможностью сохранения в BHR многоразрядной величины, указывающей число итераций одного выполнения первого цикла, и многоразрядной величины, указывающей число итераций второго цикла.

22. Процессор по п.20, в котором схема управления действует с возможностью сохранения в BHR многоразрядной величины, указывающей число итераций одного выполнения первого цикла, и не сохранения в BHR указания числа итераций второго цикла.

23. Процессор по п.20, в котором схема управления действует с возможностью сохранения в BHR многоразрядной величины, указывающей число итераций второго цикла, и не сохранения в BHR указания числа итераций любого выполнения первого цикла.

24. Процессор по п.18, в котором BHR действует с возможностью увеличения множества бит в ответ на каждое взятое вычисление команды условного перехода, связанной с циклом, таким образом, чтобы поддерживать подсчет итераций цикла непосредственно в BHR.

25. Процессор по п.18, дополнительно содержащий пороговую логику, действующую с возможностью отображения подсчета итераций цикла в фиксированную многоразрядную величину, в ответ на множество пороговых значений.

26. Устройство предсказания переходов, содержащее:
средство для определения того, что команда условного перехода является командой перехода завершения цикла;

средство для подсчета общего числа итераций цикла, завершеного посредством команды перехода завершения цикла;

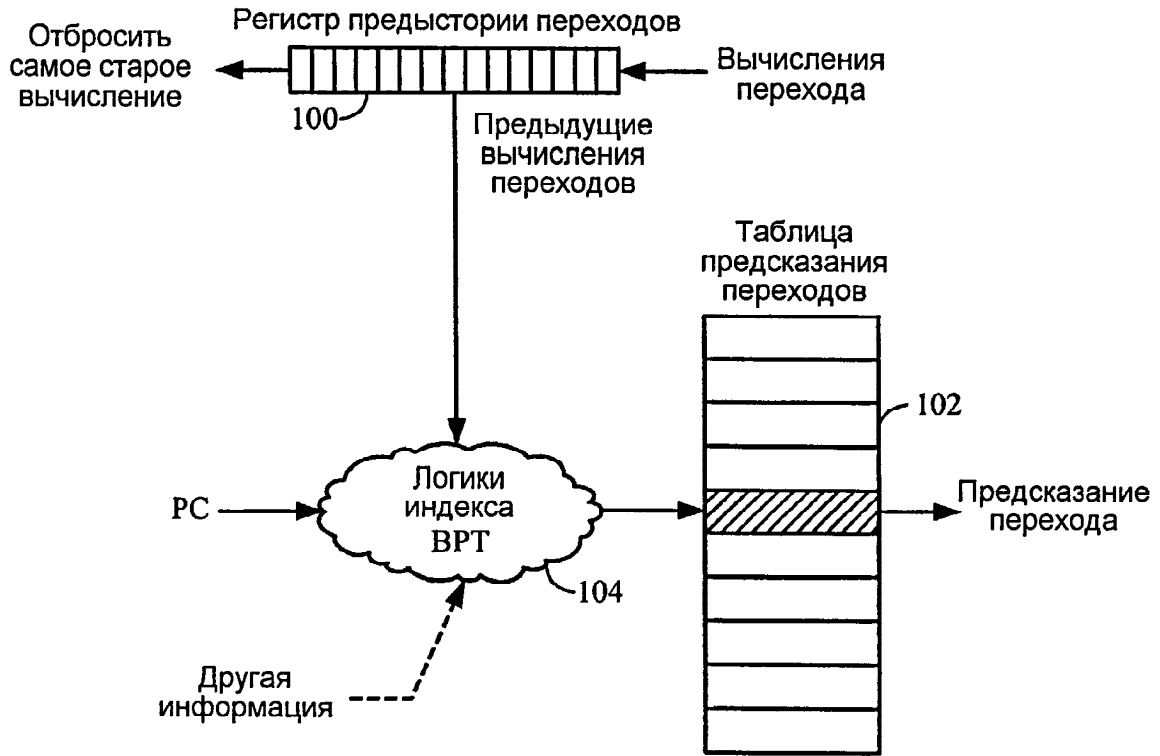
средство для сохранения многоразрядной величины в регистре предыстории переходов (BHR), причем эта величина указывает общее число итераций цикла, после завершения цикла; и

средство для индексирования таблицы устройства предсказания переходов (BPT) с помощью, по меньшей мере, BHR после завершения цикла для получения предсказания перехода для команды перехода завершения цикла, причем предсказание перехода основано, по меньшей мере, частично на предыстории переходов.

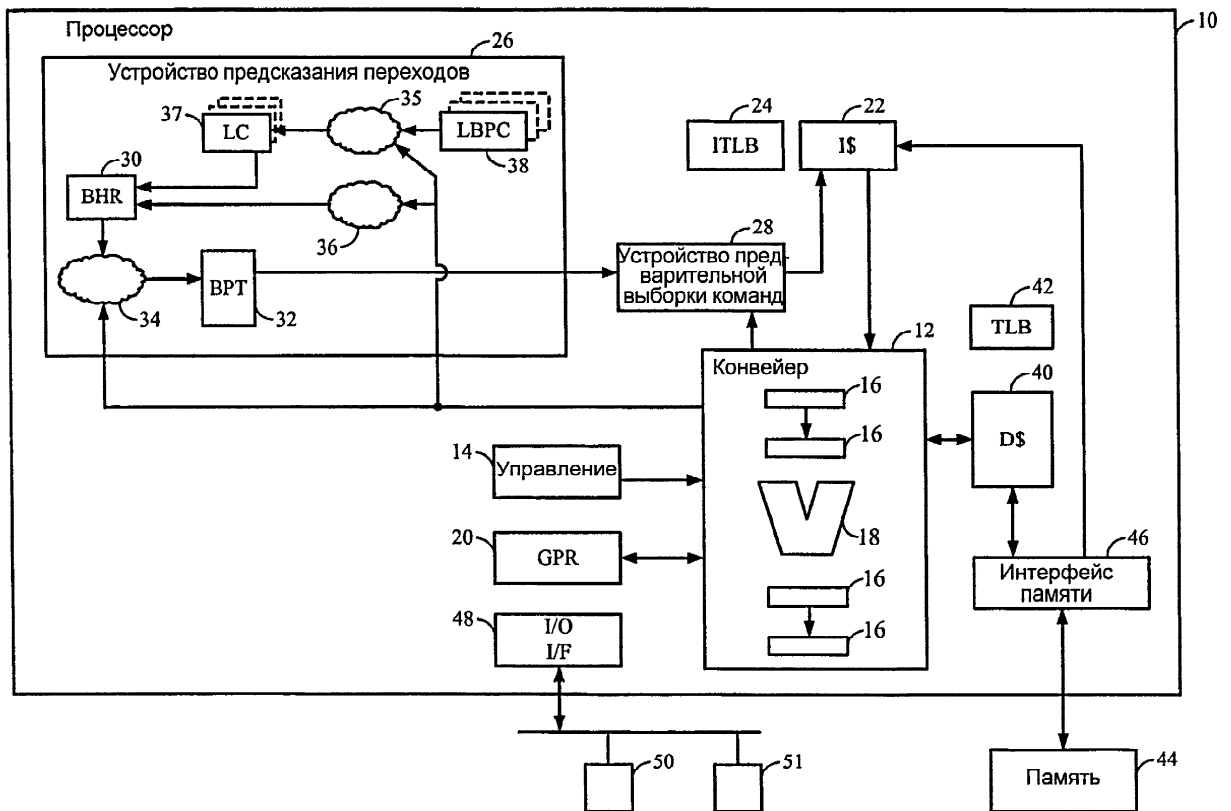
27. Машиночитаемый носитель, содержащий программный код, сохраненный на нем, причем машиночитаемый носитель содержит программный код для определения того, что команда условного перехода является командой перехода завершения цикла; программный код для подсчета общего числа итераций цикла, завершеного посредством команды перехода завершения цикла;

программный код для сохранения многоразрядной величины в регистре предыстории переходов (BHR), причем эта величина указывает общее число итераций цикла, после завершения цикла; и

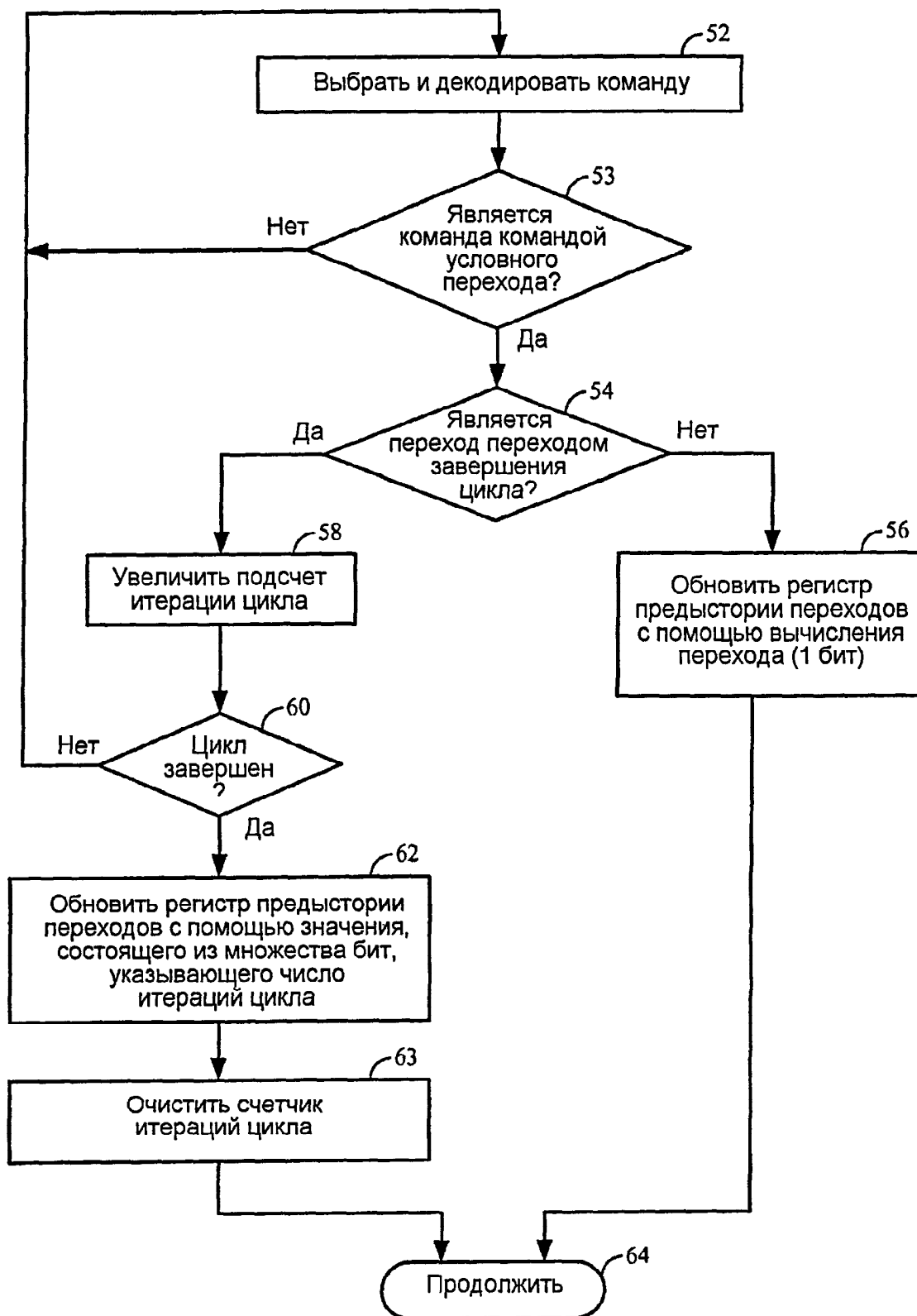
программный код для индексирования таблицы устройства предсказания переходов (BPT) с помощью, по меньшей мере, BHR после завершения цикла для получения предсказания перехода для команды перехода завершения цикла, причем предсказание перехода основано, по меньшей мере, частично на предыстории переходов.



Фиг.1



Фиг.2



Фиг.3