

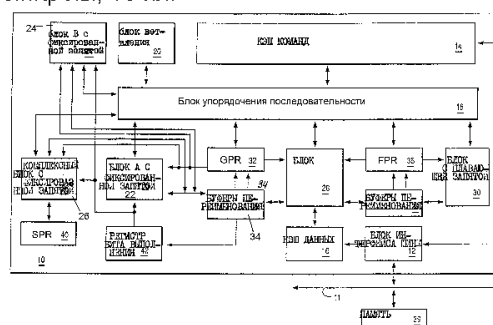
⁽¹²⁾ ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ РОССИЙСКОЙ ФЕДЕРАЦИИ

(71) Заявитель:
Интернэшнл Бизнес Машинз Корпорейшн (US)

(72) Изобретатель: Питер Сеунгюн Сонг (KR)

(73) Патентообладатель:
Интернэшнл Бизнес Машинз Корпорейшн (US)

изобретения заключается в том, что упорядоченная обратная запись оказывает меньшее отрицательное влияние на время выполнения команд системой обработки. 21 з.п.ф-лы. 16 ил.





(19) **RU** ⁽¹¹⁾ **2 142 157** ⁽¹³⁾ **C1**
(51) Int. Cl.⁶ **G 06 F 9/38**

RUSSIAN AGENCY
FOR PATENTS AND TRADEMARKS

(12) **ABSTRACT OF INVENTION**

(21), (22) Application: 96119968/09, 27.12.1994

(24) Effective date for property rights: 27.12.1994

(30) Priority: 31.03.1994 US 08/221,329

(46) Date of publication: 27.11.1999

(85) Commencement of national phase: 30.09.96

(86) PCT application:
EP 94/04313 (27.12.94)

(87) PCT publication:
WO 95/27245 (12.10.95)

(98) Mail address:
129010, Moskva, ul.B.Spasskaja, d.25, str.3,
"Gorodisskij i Partnery", Emel'janovu E.I.

(71) Applicant:

Internehsnl Biznes Mashinz Korporejshn (US)

(72) Inventor:

Piter Seungjun Song (KR)

(73) Proprietor:

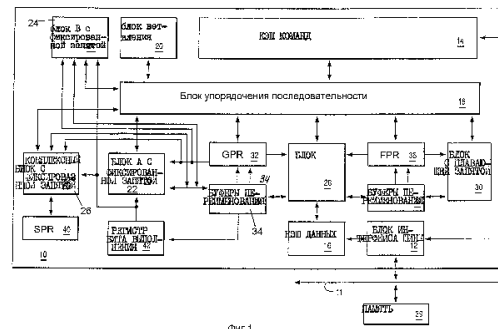
Internehsnl Biznes Mashinz Korporejshn (US)

(54) **METHOD FOR OPERATIONS OF PROCESSING SYSTEM**

(57) Abstract:

FIELD: data processing. SUBSTANCE: method involves transmission of sequence of instructions into executing units for execution. Prior to execution of an instruction, method involves checking possibility of exception as the result of this instruction. Execution of instruction finishes, if no exception follows its execution and execution of each instruction preceding this one. Sequence of execution of instructions is constant independently from fact whether execution unit finishes or not execution of each instruction, which cannot be followed by exception. EFFECT: ordered

reverse storage has lesser negative impact on duration of execution of instructions by processing system. 22 cl, 16 dwg



Изобретение относится к системам обработки информации, в частности к системе обработки и к способу ее функционирования.

Система суперскалярной обработки включает множество исполнительных блоков для одновременного выполнения множества команд. В некоторых системах обработки, команды могут выполняться вне очереди относительно их запрограммированной последовательности в потоке команд. Несмотря на это, одна или более выбранных команд могут быть преобразованы в последовательную форму выполнения, когда такая команда выполняется только в порядке ее запрограммированной последовательности. Кроме этого, некоторые из таких систем обработки предназначены для записи результатов выполнения команд в ячейки памяти (например, регистры конфигурации) в порядке запрограммированной последовательности команд. В соответствии с ранее известными способами, такая упорядоченная обратная запись оказывает отрицательное влияние на время завершения обработки команд системой обработки. Более того, подобная упорядоченная обратная запись оказывает отрицательное влияние на время выполнения команды, преобразованной в последовательную форму выполнения, системой обработки.

Таким образом, возникает необходимость в системе обработки и в способе работы, для которых упорядоченная обратная запись оказывает меньшее отрицательное влияние на время выполнения команд системой обработки и на выполнение последовательно выполняемой команды посредством системы обработки.

В системе обработки и способе ее функционирования множество команд передаются в последовательности на исполнительную схему для выполнения. До выполнения каждой команды определяется, возможна ли исключительная ситуация в результате выполнения данной команды. Команда завершается, если определяется, что выполнение данной команды не вызывает исключительной ситуации, как и каждой команде последовательности, предшествующей этой команде, независимо от того, закончила ли исполнительная схема выполнение каждой команды, для которой невозможна исключительная ситуация.

Преимущество, обеспечиваемое настоящим изобретением, заключается в том, что упорядоченная обратная запись оказывает меньшее отрицательное влияние на время выполнения команд системой обработки и на выполнение последовательно выполняемых команд системой обработки.

Пример осуществления настоящего изобретения и его преимущества поясняются следующим описанием и иллюстрирующими его чертежами, на которых представлено следующее:

фиг. 1 - блок-схема процессора для обработки информации в соответствии с предпочтительным вариантом осуществления;

фиг. 2 - блок-схема исполнительного блока с фиксированной запятой процессора по фиг. 1;

фиг. 3 - блок-схема блока упорядочения последовательности процессора по фиг. 1;

фиг. 4 - концептуальная иллюстрация буфера переупорядочения последовательности блока упорядочения по фиг. 3;

5 фиг. 5a,b - иллюстрации различных стадий команд, обрабатываемых процессором по фиг. 1;

фиг. 6 - концептуальная иллюстрация буфера переупорядочения процессора по фиг. 1;

10 фиг. 7 - иллюстрация различных стадий четырех команд по фиг. 6;

фиг. 8a-d - концептуальная иллюстрация буферов переименования процессора по фиг. 1;

15 фиг. 9 - иллюстрация различных стадий четырех команд без использования существенных признаков предпочтительного варианта осуществления;

фиг. 10a-d - концептуальные иллюстрации буферов переименования процессора по фиг. 9;

20 фиг. 11 - блок-схема альтернативного исполнения буферов переименования процессора по фиг. 1;

фиг. 12a-c - концептуальные иллюстрации буфера переупорядочения процессора по фиг. 1;

25 фиг. 13 - иллюстрация различных стадий пяти команд по фиг. 12a-c;

фиг. 14a-g - концептуальные иллюстрации буферов переименования по фиг. 1;

30 фиг. 15 - иллюстрация различных стадий пяти команд по фиг. 12a-c без использования существенных признаков предпочтительного варианта осуществления;

фиг. 16a-h - концептуальные иллюстрации буферов переименования процессора по фиг. 1 для различных циклов обработки команды, проиллюстрированных на фиг. 15.

35 Пример осуществления настоящего изобретения и его преимущества поясняются с помощью фиг. 1-16h, где одинаковыми позициями обозначены одинаковые и соответствующие элементы, изображенные на чертежах.

40 На фиг. 1 показана блок-схема процессора 10 системы для обработки информации в соответствии с предпочтительным вариантом осуществления. В предпочтительном варианте процессор 10 выполнен в виде единой интегральной схемы суперскалярного микропроцессора. Соответственно, как описывается далее, процессор 10 содержит различные блоки, регистры, буферы, памяти и другие части, каждая из которых образована в интегральной схеме. Кроме этого, в 45 предпочтительном варианте осуществления процессор 10 работает по методу RISC (компьютер с сокращенным набором команд). Как показано на фиг. 1, системная шина 11 связана с блоком интерфейса шины ("BIU") 12 процессора 10. BIU 12 управляет передачей информации между процессором 10 и 50 системной шиной 11.

BIU 12 связан с кэшем (быстродействующей буферной памятью) команд 14 и с кэшем данных 16 процессора 10. Кэш команд 14 выдает команды на блок упорядочения последовательности 18. В 60 ответ на такие команды от кэша команд 14, блок упорядочения 18 избирательно выдает команды на другие исполнительные схемы процессора 10.

Кроме блока упорядочения 18, в

предпочтительном варианте выполнении исполнительные схемы процессора 10 включают в себя множество исполнительных блоков, а именно: блок ветвления 20, блок обработки с фиксированной запятой А ("FXUA") 22, блок обработки с фиксированной запятой В ("FXUB") 24, комплексный блок обработки с фиксированной запятой ("CFXU") 26, блок загрузки/хранения ("LSU") 28 и блок обработки плавающей запятой ("FPU") 30. Блоки FXUA 22, FXUB 24, CFXU 26 и LSU 28 вводят информацию из исходного операнда из регистров конфигурации общего назначения ("GPR") 32 и из буферов переименования с фиксированной запятой 34. Кроме того, блоки FXUA 22 и FXUB 24 вводят "бит переноса" из регистра бита переноса ("CA") 42. Блоки FXUA 22, FXUB 24, CFXU 26 и LSU 28 выдают результаты (информацию операнда назначения) их работы для хранения на выбранных входах буферов переименования с фиксированной запятой 34. Кроме этого, CFXU 26 вводит и выводит информацию исходного операнда и операнда назначения из регистров специального назначения ("SPR") 40.

FPU 30 вводит информацию своего исходного операнда из регистров конфигурации с плавающей запятой ("FPR") 36 и из буферов переименования с плавающей запятой 38. FPU 30 выдает результаты (информацию операнда назначения) своей работы для хранения на выбранных входах буферов переименования с плавающей запятой 38.

В ответ на команду ввода LSU 28 вводит информацию из кэша данных 16 и копирует такую информацию в один из выбранных буферов переименования 34 и 38. Если такая информация не хранится в кэше данных 16, то кэш данных 16 вводит (посредством BIU 12 и системной шины 11) такую информацию из системной памяти 39, связанной с системной шиной 11. Более того, кэш данных 16 имеет возможность выдавать (посредством BIU 12 и системной шины 11) информацию из кэша данных 16 на системную память 39, связанную с системной шиной 11. В ответ на команду Store LSU 28 вводит информацию из одного из выбранных GPR 32 и FPR 36 и копирует такую информацию в кэш данных 16.

Блок упорядочения 18 вводит и выводит информацию от GPR 32 и FPR 36. Блок ветвления 20 вводит из блока упорядочения 18 команды и сигналы, указывающие текущее состояние процессора 10. В ответ на такие команды и сигналы блок ветвления 20 выдает (на блок упорядочения 18) сигналы, указывающие соответствующие адреса памяти, в которых хранится последовательность команд для выполнения процессором 10. В ответ на такие сигналы от блока ветвления 20 блок упорядочения 18 вводит указанную последовательность команд из кэша команд 14. Если одна или более последовательностей команд не сохранились в кэше команд 14, то кэш команд 14 вводит (посредством BIU 12 и системной шины 11) такие команды из системной памяти 39, связанной с системной шиной 11.

В ответ на команды, введенные из кэша команд 14, блок упорядочения 18 избирательно пересылает команды на один из выбранных блоков 20, 22, 24, 26, 28 и 30. Каждый исполнительный блок выполняет

одну или более команду из конкретного класса команд. Например, FXUA 22 и FXUB 24 выполняют первый класс математических операций с фиксированной запятой для исходных операндов, такие как сложение, вычитание, операцию AND, операцию OR, операцию XOR. CFXU 26 выполняет второй класс операций с фиксированной запятой для исходных операндов, такие как умножение и деление с фиксированной запятой. FPU 30 выполняет действия с плавающей запятой с исходными операндами, такие как умножение и деление с плавающей запятой.

После запоминания информации в одном из буферов переименования 34 такая информация связывается с ячейкой памяти (например, с одним из регистров GPR 32 или CA 42) в соответствии с командой, для которой назначается выбранный буфер переименования. Информация, хранящаяся в одном из буферов переименования 34, копируется в связанный с ней один из GPR 32 (или в регистр CA 42) в ответ на сигналы от блока упорядочения 18. Как описывается ниже, в соответствии с фиг. 6-10 блок упорядочения 18 направляет такую скопированную информацию, хранящуюся в одном из выбранных буферов переименования 34, в ответ на "завершение" команды, которая сформировала эту информацию. Такое копирование называется "обратной записью".

После запоминания информации в одном из выбранных буферов переименования 38 такая информация связывается с одним из FPR 36. Информация, сохраненная в одном из выбранных буферов переименования 38, копируется в связанный с ней один из FPR 36 в ответ на сигналы от блока упорядочения 18. Блок упорядочения 18 управляет таким копированием информации, запоминаемой в одном из выбранных буферов 38, в ответ на "завершение" команды, которая сформулировала эту информацию.

Высокая производительность процессора 10 достигается путем обработки множества команд одновременно в различных блоках выполнения 20, 22, 24, 26, 28 и 30. Соответственно каждая команда обрабатывается как последовательность стадий, каждая из которых выполняется параллельно с выполнением стадий других команд. Такой способ называется "конвейерной обработкой". В соответствии с важным аспектом предпочтительного варианта выполнения команда обычно обрабатывается как шесть стадий, а именно выборка, декодирование, диспетчеризация, выполнение, завершение и обратная запись.

На стадии выборки блок упорядочивания 18 избирательно вводит (из кэша команд 14) одну или более команд по одному или более адресу памяти, по которому хранится последовательность команд, рассматриваемых далее в связи с блоком ветвления 20 и блоком упорядочивания 18.

На стадии декодирования блок упорядочивания 18 декодирует до четырех выбранных команд.

На стадии диспетчеризации блок упорядочивания 18 избирательно передает до четырех декодированных команд на один из выбранных (в ответ на декодирование на стадии декодирования) исполнительных блоков 20, 22, 24, 26, 28 и 30 после приема

содержимого буфера переименования для каждого результата выполнения переданной команды (информация операнда назначения). В состоянии диспетчеризации информация операнда передается на выбранные исполнительные блоки для передаваемых команд. Процессор 10 передает команды в порядке, соответствующем запрограммированной последовательности.

На стадии выполнения исполнительные блоки выполняют переданные им команды и выдают результаты осуществления (информацию операнда назначения) их работы для сохранения на выбранных входах буферов переименования 34 и 38, как описано ниже. Таким образом, процессор 10 имеет возможность выполнять команды вне очереди по отношению к их запрограммированной последовательности.

На стадии завершения блок упорядочения 18 указывает, что команда "завершена", как описано ниже со ссылкой на фиг. 3, 4. Процессор 10 "завершает" выполнение команд в соответствии с их запрограммированной последовательностью.

На стадии обратной записи устройство упорядочения 18 управляет копированием информации из буферов переименования 34 и 38 в GPR 32 и FPR 36 соответственно. Блок упорядочения 18 управляет таким копированием информации, запоминаемой в выбранном буфере переименования, как описано ниже со ссылками на фиг. 6-10. Аналогично, в состоянии обратной записи конкретной команды процессор 10 обновляет свои состояния конфигурации в ответ на конкретную команду. Процессор 10 обрабатывает соответствующие стадии "обратной записи" команд в соответствии с их запрограммированной последовательностью. Как описано ниже со ссылками на фиг. 6-10, процессор 10 в определенных ситуациях предпочтительно совмещает стадии завершения команд и стадии обратной записи.

В предпочтительном варианте выполнения необходим один машинный цикл для завершения каждой стадии выполнения команды. Тем не менее, для выполнения некоторых команд (например, комплексных команд с фиксированной запятой, выполняемых CFXU 26) необходимо более одного машинного цикла. Соответственно может возникнуть различное запаздывание между выполнением различных команд и стадиями завершения в зависимости от различного времени, необходимого для завершения предыдущих команд.

На фиг. 2 представлена блок-схема FXUA 22 процессора 10. FXUA 22 содержит устройство резервирования 50a и 50b. Аналогично, каждый блок ветвления 20, FXUB 24, CFXU 26, LSU 28 и FPU 30 имеет соответствующие устройства резервирования. Для ясности далее описывается работа только FXUA 22 и его устройства резервирования в качестве примера работы других блоков и их соответствующих устройств резервирования.

Каждое устройство резервирования 50a, b имеет возможность сохранять информацию для соответствующих команд, переданных от блока упорядочения 18 для выполнения в FXUA 22. Каждое устройство резервирования содержит соответствующее поле регистра

назначения, поле операнда A, поле операнда B и поле кода операции. Кроме того, в соответствии с важным аспектом предпочтительного варианта осуществления каждое устройство резервирования содержит соответствующее поле разрешения выполнения ("ЕОК").

В своем поле регистра назначения устройство резервирования указывает по меньшей мере один регистр назначения (как определено блоком упорядочения 18) для соответствующей команды устройства резервирования. Аналогично, в своих соответствующих полях операнда A и операнда B устройство резервирования запоминает информацию об исходном операнде (от GPR 32, от буферов переименования 34, FXUB 24, CFXU 26 или LSU 28) для соответствующей команды устройства резервирования. Устройство резервирования в своем поле операции хранит код операции (в соответствии с тем, что определено блоком упорядочения 18), указывающий операцию, которая должна быть выполнена FXUA 22 над информацией исходного операнда, в ответ на соответствующую команду устройства резервирования.

В ответ на сигналы от логики управления 56 исполнительная логика 54 вводит информацию исходного операнда из полей операндов A и B устройства резервирования и выполняет операцию (указанную запомненным кодом операции устройства резервирования) над ними. Информация, полученная в результате выполнения такой операции, выдается от исполнительной логики 54 на буферы переименования 34, FXUB 24, CFXU 26 и LSU 28. Такая информация сохраняется в одном из выбранных буферов переименования 34. В ответ на указание регистра назначения выходной информацией мультиплексора 58 запомненная информация назначается (в выбранном буфере переименования) одному из GPR 32.

В соответствии с важным аспектом варианта предпочтительного выполнения, устройство резервирования запоминает информацию ЕОК (в соответствии с тем, что определяется блоком упорядочения 18) в его поле ЕОК. Предпочтительно такая информация ЕОК соответствует определенным ситуациям, когда процессор 10 задерживает выполнение последовательно выполняемой команды. Преобразование в последовательную форму выполнения является одним из способов задержки выполнения команды в процессорах многократного конвейера с внеочередным выполнением, примером которых является процессор 10. В первой ситуации, когда процессор 10 задерживает выполнение преобразованной в последовательную форму выполнения команды, эта команда не должна выполняться теоретически. Во второй ситуации, когда процессор 10 задерживает выполнение преобразованной в последовательную форму выполнения команды, выполнение команды задерживается до момента, когда информация всех ее исходных операндов станет доступной и правильной.

Что касается первой ситуации (когда команда теоретически не должна

выполняться), процессор 10 в нормальном состоянии выполняет команды теоретически так, чтобы команды выполнялись вне очереди относительно их запрограммированной последовательности в потоке команд. Соответственно, не обязательно, чтобы результаты выполнения команд (информация операнда назначения) были доступны в соответствии с запрограммированной последовательностью этих команд. Однако процессор 10 записывает результаты выполнения команд обратно в регистры конфигурации (например GPR 23 и FPR 36) в порядке, соответствующем запрограммированной последовательности команд. Для этой цели процессор 10 имеет буферы переименования 34 и 38 для промежуточного хранения результатов выполнения команд до подходящего момента (т.е. до окончания выполнения всех предыдущих команд без условий исключения) для записи временно хранимых результатов обратно в регистры конфигурации.

Тем не менее, в качестве примера в предпочтительном исполнении некоторые команды действуют на SPR 40 (фиг. 1), причем результаты выполнения команды записываются непосредственно в SPR 40 без временного хранения в буферах переименования. Примером такой команды является команда Move To, в соответствии с которой CF XU 26 перемещает информацию в один из SPR 40 от одного из GPR 32. Как показано на фиг. 1, CF XU 26 связан с SPR 40. Команда Move To немедленно обновляет один из SPR 40 после выполнения. В качестве другого примера результат выполнения команды Store непосредственно записывается в ячейку памяти в кэш данных 16 без временного хранения в буферах переименования.

Процессор 10 не выполняет теоретически такие команды (когда результаты выполнения команд записываются непосредственно в регистры конфигурации или в ячейку памяти без временного хранения в буферах переименования), так что процессор 10 имеет возможность обеспечивать точные прерывания и точные исключения. Более того, так как результаты выполнения команд непосредственно записываются в регистр конфигурации или ячейку памяти без предварительного хранения в буферах переименования, то такие команды обрабатываются без стадии обратной записи. Соответственно, для гарантии завершения в порядке очереди и обратной записи процессор 10 задерживает выполнение такой команды до завершения выполнения всех предыдущих команд.

В соответствии с другой ситуацией (когда выполнение команды задерживается до момента, когда информация всех ее исходных операторов станет доступной и правильной), указанной на фиг. 3, устройство резервирования временно сохраняет информацию соответствующей команды, для которой информация исходного операнда не доступна на момент передачи этой команды от блока упорядочивания 18. После того как информация исходного операнда станет доступной от исполнительного блока, устройство резервирования вводит и сохраняет такую информацию исходного операнда. В подходящий момент устройство

резервирования передает информацию этого исходного операнда исполнительской логике 54.

В предпочтительном исполнении большинство команд в качестве исходных операндов определяют один или более GPR 32 и FPR 36. Соответственно, в предпочтительном варианте выполнения устройства резервирования включают цепь для продвижения информации от исполнительных блоков.

Тем не менее, в предпочтительном варианте выполнения устройства резервирования не содержат схему для продвижения информации от исходных операндов других типов, таких как регистр CA 42 или SPR 40. Это объясняется тем, что размер и стоимость таких схем не соответствует частоте команд, которые определяют такие неподвинутые исходные операнды. Вместо этого процессор 10 в предпочтительном варианте выполнения задерживает выполнение команды, определяющей такие неподвинутые исходные операнды, по меньшей мере до завершения выполнения всех предыдущих команд. В ответ на завершение выполнения всех предыдущих команд эти неподвинутые исходные операнды считываются из одного из определенных регистров конфигурации (например, SPR 40). Примерами таких команд являются (1) расширенная арифметическая операция, которая считывает регистр CA 42, и (2) команда Move From, которая перемещает информацию от одного из SPR 40 в один из GPR 32.

Наконец, процессор 10 задерживает выполнение команды последовательного выполнения по меньшей мере до завершения всех предыдущих команд в двух ситуациях. В первой ситуации команда не должна выполняться теоретически. Во второй ситуации команда определяет по меньшей мере один исходный операнд, для которого процессор 10 не должен привлекать схему продвижения. Предпочтительно, чтобы такие ситуации обрабатывались блоком упорядочивания 18, определяющим информацию ЕОК, которая должна храниться в устройстве резервирования.

Если команда должна выполняться последовательно, то блок упорядочивания 18 очищает бит ЕОК (информации ЕОК, которая хранится в поле ЕОК устройства резервирования) и устанавливает его в состояние логического 0 во время передачи команды на FXUA 22. После установления бита ЕОК в состояние логического 0 блок упорядочивания 18 предотвращает выполнение команды блоком FXUA 22, даже если эта команда готова к выполнению. Соответственно FXUA 22 выполняет такие последовательно выполняемые команды только в ответ на выдачу блоком упорядочивания 18 сигнала по линии 60, как описано ниже.

В отличие от вышесказанного, если команда не должна выполняться последовательно, то блок упорядочивания 18 устанавливает бит ЕОК в состояние логической 1 во время передачи команды на FXUA 22. Посредством установки бита ЕОК в состояние логической 1 блок упорядочивания 18 позволяет FXUA 22 выполнять команду, как только информация исходного операнда

команды будет правильной и доступной.

На фиг. 3 представлена блок-схема блока упорядочивания 18. Как указывалось ранее, на стадии выборки блок упорядочивания 18 выполняет избирательный ввод команд, количеством до четырех, из кэша команд 14 и сохраняет такие команды в буфере команд 70. На стадии декодирования логика декодирования 72 вводит и декодирует до четырех выбранных команд из буфера команд 70. На стадии диспетчеризации логика передачи 74 избирательно передает до четырех декодированных команд на выбранный (в ответ на выполнение декодирования на стадии декодирования) один из исполнительных блоков 20, 22, 24, 26, 28 и 30.

На фиг. 4 представлена концептуальная иллюстрация буфера переупорядочения 76 блока упорядочивания 18 предпочтительного варианта выполнения. Как показано на фиг. 4, буфер переупорядочения 76 имеет шестнадцать входов, соответственно обозначаемых номерами буферов 0-15. Каждый вход имеет пять основных полей, а именно поле "типа команды", поле "количества назначений GPR", поле "количества назначений FPR", поле "завершение" и поле "исключение". Более того, поле типа команды имеет подполе "исполнительного блока" и подполе "ЕОК".

Обратимся к фиг. 3, после того как логика передачи 74 передаст команду на исполнительный блок, блок упорядочивания 18 назначает переданную команду соответствующему входу в буфере переупорядочения 76. Блок упорядочивания 18 назначает входы буфера переупорядочения 76 переданным командам на основании первым пришел - первым обслужен и циклическим образом так, что блок упорядочивания 18 назначает вход 0, за которым следуют последовательно входы 1-15, а затем вновь вход 0. После назначения соответствующему входу переданной команды в буфере переупорядочения 76 логика передачи 74 выдает информацию, соответствующую переданной команде, для хранения в различных полях и подполях назначенного входа буфера переупорядочения 76.

Например, для входа 0 на фиг. 4 буфер переупорядочения 76 указывает, что команда была передана на FXUA 22. Более того, вход 0 указывает, что переданная команда выполняется последовательно, так что ЕОК=0 и процессор 10 должен задержать выполнение всех предыдущих команд. Кроме того, для входа 1 буфер переупорядочения 76 указывает, что последующая команда должна выполняться последовательно так, что ЕОК=0.

В соответствии с другими важными аспектами предпочтительного исполнения вход 0 далее указывает, что переданная команда имеет один GPR регистр назначения (такой, как "количество назначений GPR" = 1), имеет регистры нулевого назначения FPR (такие, как "количество назначений FPR" = 0), еще не завершена (так, что "исключение" = 0).

Исполнительный блок выполняет переданную команду, блок выполнения

преобразует вход, назначенный команде в буфере переупорядочения 76. Более конкретно, в ответ на завершение выполнения переданной команды исполнительный блок изменяет поле "завершение" входа (так, чтобы "завершение" = 1). Если исполнительный блок встретит исключение во время выполнения переданной команды, то исполнительный блок изменяет поле "исключение" входа (так, чтобы "исключение" = 1).

Согласно фиг. 3 входы буфера переупорядочения 76 считываются логикой завершения 80 и логикой исключения 82 блока упорядочивания 18. Более того, в соответствии с одним из важных аспектов предпочтительного исполнения входы буфера переупорядочения 76 считываются логикой последовательного исполнения 84 устройства упорядочивания 18. В ответ на поля "исключения" буфера переупорядочения 76 логика исключения 82 передает исключения, которые встретились во время выполнения переданных команд.

В ответ на поля "завершение" и поля "исключения" буфера переупорядочения 76 логика завершения 80 выдает сигналы логике передачи 74, логике последовательного выполнения 84, а также буферу переупорядочения 76. Посредством этих сигналов логика завершения 80 указывает "завершение" команд в соответствии с их запрограммированной последовательностью. Логика завершения 80 указывает "завершение" команды, если она удовлетворяет следующим условиям:

условие 1 - исполнительный блок (которому была передана команда) заканчивает выполнение команды (так что "завершение" = 1 для назначенных команде входов в буфере переупорядочения 76);

условие 2 - не было обнаружено никаких исключительных ситуаций ни на каком из этапов обработки команды (так что "исключительная ситуация" 0 для входов, назначенных команде в буфере переупорядочения 76);

условие 3 - любая ранее переданная команда удовлетворяет условиям 1 и 2.

В ответ на информацию буфера переупорядочения 76 логика передачи 74 определяет подходящее количество дополнительных команд для передачи.

В соответствии с важным аспектом варианта предпочтительного выполнения в ответ на сигналы от логики завершения 80 логика последовательного выполнения 84 избирательно выдает сигналы на FXUA 22 по линии 60. Если связанный с переданной на FXUA 22 командой бит ЕОК (в поле "типа команды" входа, назначенного команде в буфере переупорядочения 76) установлен в состояние логического 0, то логика последовательного выполнения 84 выдает сигнал по линии 60 в ответ на "завершение" всех команд, предшествующих переданной команде. FXUA 22 выполняет такую переданную команду только в ответ на передачу логикой последовательного выполнения 84 сигнала по линии 60. После передачи логикой последовательного выполнения 84 сигнала по линии 60 такой переданной командой (для которой назначенный бит ЕОК установлен в состояние логического 0) будет самая старая команда,

ожидающая выполнения в устройстве резервирования 50а, b FXUA 22, так как команды "завершаются" в соответствии с их запрограммированной последовательностью.

Аналогично, логика последовательного выполнения 84 избирательно выдает сигналы на исполнительные блоки 20, 24, 26, 28 и 30 по линиям 86, 88, 90, 92 и 94, соответствующим образом связанным с ней.

На фиг. 5а, b проиллюстрированы различные стадии команд. Согласно фиг. 5а выполнение (цикл 6) последовательно выполняемой команды INST n+1 (например команды Move To или команды Move From) задерживается до завершения выполнения всех предыдущих команд (например, до цикла 5 предыдущей команды INST n). Тем не менее процессор 10 предпочтительно не задерживает передачу (цикл 3) или выполнение (цикл 4) команды INST n+2, которой предшествует команда последовательного выполнения INST n+1. Таким образом, процессор 10 обеспечивает непрерывную передачу команд (таких как INST n+2), которым предшествует последовательно выполняемая команда (такая, как INST n+1). Кроме этого, исполнительный блок процессора 10 имеет возможность подавать команды вне очереди от устройства резервирования исполнительных блоков на свою исполнительную логику (например, исполнительную логику 54 по фиг. 2), даже если самая старая команда, ожидающая выполнения в устройстве резервирования исполнительного блока, преобразуется в последовательную форму выполнения.

Процессор согласно предпочтительному варианту выполнения достигает более высокой производительности, чем в альтернативных случаях. Как показано на фиг. 5b, в соответствии с одним из таких альтернативных способов передача (цикл 8) команды INST i+2 задерживается в ответ на декодирование (цикл 2) команды INST i+1, которая выполняется последовательно. В соответствии с таким альтернативным способом команда передачи команды INST i+2 выполняется только после "завершения" всех ранее переданных команд (например, после цикла 7 выполнения предыдущей последовательно выполняемой команды INST i+1). Узким местом такого альтернативного способа является то, что выполнение задерживается для любой команды (такой как INST i+2), которой предшествует последовательно выполняемая команда (такая, как INST i+1).

Как указывалось ранее, процессор 10 достигает непрерывной передачи команд, которым предшествует последовательно выполняемая команда. Команда Move From является последовательно выполняемой командой и поэтому выигрывает за счет использования способа последовательного выполнения согласно предпочтительному варианту выполнения. Более того, в соответствии с другим важным аспектом предпочтительного выполнения процессор 10 далее содержит схему для вывода результатов (информации операнда назначения) команды Move From для хранения в выбранном буфере переименования до обратной записи информации операнда назначения в один из

GPR 32.

Процессор 10 поддерживает такое переименование независимо от того, является ли регистр назначения команды Move From одним из GPR 32. Таким образом, процессор 10 использует схему продвигания своих устройств резервирования совместно со схемой поиска буферов переименования для корректного согласования информации операнда назначения (команды Move From) с исходным регистром команды, переданной после команды Move From. Соответственно, другие команды, которым предшествует команда Move From, могут передаваться в любое время после передачи команды Move From. Без использования такой схемы для команды Move From другие команды, которым предшествует команда Move From, не будут передаваться до "завершения" команды Move From, так как для выполнения других команд (в качестве информации их исходного операнда) может потребоваться информация операнда назначения команды Move From.

В соответствии с другим важным аспектом предпочтительного исполнения процессор 10 упорядочивает диспетчеризацию команд, реагируя на выборку команд и декодируя соответствующие исключительные ситуации ("IFDRE") оптимальным образом. Процессор 10 определяет выборку команды или исключительную ситуацию, связанную с декодированием, на этапе выборки или декодирования, если состояние исключительной ситуации может быть определено полностью на основании анализа команды и состояния процессора 10. Примерами таких выборок команд или исключительных ситуаций, связанных с декодированием, являются неисправность страницы доступа к команде, нарушение защиты памяти доступа к команде, нарушение привилегированной команды, а также несанкционированные команды.

В основном в ответ на выявление выборки такой команды или исключительной ситуации, связанной с декодированием, блок упорядочивания 18 передает команду, вызывающую IFDRE на устройство резервирования исполнительного блока, но одновременно устанавливает бит EOK в состояние логического 0, как указано ранее со ссылкой на фиг. 2-4. Более того, блок упорядочивания 18 запоминает указатель условия IFDRE посредством установки "исключительная ситуация" = 1 (одновременно "завершение" = 0) для входа, назначенного для команды, вызывающей IFDRE в буфере переупорядочения 76. Такой указатель определяет команду как команду, вызывающую IFDRE.

Как указывалось ранее со ссылками на фиг. 2-4, путем установки бита EOK в логическое состояние 0 блок упорядочивания 18 предотвращает выполнение исполнительным блоком переданной команды, даже если эта команда готова к выполнению. Соответственно, исполнительный блок выполняет такую команду только в ответ на сигнал от блока упорядочивания 18, переданный по одной из соответствующим образом подключенной линии 60, 86, 88, 90, 92 или 94, как указывалось ранее со ссылками на фиг. 2-4.

В ответ на завершение всех команд, предшествующих вызывающей IFDRE

команде, блок упорядочивания 18 обрабатывает выборку или исключительную ситуацию, связанную с декодированием, вместо того чтобы передавать сигнал на исполнительный блок посредством соответствующим образом подсоединенных линий 60, 86, 88, 90, 92 или 94. Таким образом, вызывающая IFDRE команда никогда не будет выполнена исполнительным блоком. Блок упорядочивания 18 определяет команду как вызывающую IFDRE команду в ответ на состояние: "исключительная ситуация" = 1 и "завершение" = 0 для входа, связанного с вызывающей IFDRE командой в буфере переупорядочения 76.

Таким образом, процессор 10 упорядочивает и ускоряет передачу команды посредством логики передачи 74 (фиг. 3), что ускоряет работу критической схемы в суперскалярном процессоре. Соответственно, процессор 10 в предпочтительном варианте выполнения достигает более высокой производительности по сравнению с альтернативными вариантами. В соответствии с одним из таких альтернативных способов процессор никогда не передает вызывающую IFDRE команду. Такой альтернативный способ приводит к сложной и медленной передаче, так как для каждой команды процессору необходимо определить выборку каждой команды и состояние исключительной ситуации, связанное с декодированием, перед определением, куда необходимо передавать команду. Например, в соответствии с таким альтернативным способом процессор определяет, куда необходимо передать n-ую команду в ответ на определение, имела ли n-ая команда или какая-либо более ранняя n-1-ая команда выборку или связанное с декодированием состояние исключительной ситуации.

В отличие от этого в предпочтительном варианте выполнения, логика передачи 74 (фиг. 3) работает независимо от выборки или состояний исключительной ситуации, связанных с декодированием. Даже после принятия решения о промежуточной передаче команды на исполнительный блок для выполнения блок упорядочивания 18 определяет, существует ли условие IFDRE для данной команды. Если блок упорядочивания 18 определит, что условие IFDRE существует для данной команды, то блок 18 выдаст указатель условия IFDRE для запрещения выполнения команды исполнительным блоком. Более конкретно, в ответ на определение блоком упорядочивания 18 существования условия IFDRE для данной команды блок 18 выдаст такой указатель во время действительной передачи посредством (1) установки состояния "исключительная ситуация" = 1 (при этом "завершение" = 0) для входа, назначенного команде, вызывающей IFDRE в буфере переупорядочения 76, и (2) установкой бита EOK в устройстве резервирования команды, вызывающей IFDRE, в логическое состояние 0.

Это обеспечивает преимущества, так как после принятия решения относительно передачи команды на практике обычно не учитываются обратные результаты такого решения, чтобы в определенных ситуациях не осуществлять передачу команды. Более того,

во время цикла передачи выполняются дополнительные операции после принятия решения относительно передачи данной команды. Соответственно, процессору 10 нет необходимости определять какое-либо состояние исключительной ситуации до определения, необходимо ли передавать данную команду. Для блока упорядочивания 18, в частности, выгодно передавать многократные команды во время одиночного цикла процессора 10.

Фиг. 6 является концептуальной иллюстрацией буфера переупорядочения 76, на которой показаны четыре команды во время завершения выполнения ("завершение" = 1) в течение того же цикла процессора 10. Фиг. 7 является иллюстрацией различных стадий четырех команд по фиг. 6. Фиг. 8a-d являются концептуальными иллюстрациями буферов переименования 34 процессора 10.

Обратимся к фиг. 6-8. В соответствии с важным аспектом предпочтительного исполнения обратная запись не зависит от завершения выполнения команды так, что состояние "обратная запись" команды, обрабатываемой процессором 10, отделимо от состояния "завершения" команды. Преимущественно, посредством отделения обратной записи от завершения подобным образом процессор 10 достигает эффективной работы, используя меньшее количество портов обратной записи между буферами переименования и регистрами конфигурации. Например, как показано на фиг. 1, процессор 10 предпочтительного варианта выполнения содержит два порта обратной записи между буферами переименования 34 и GPR 32, а также два порта обратной записи между буферами переименования 38 и FPR 36. Одновременно с использованием меньшего количества портов уменьшаются и физические размеры буферов переименования 34 и 38 и регистров конфигурации 32 и 36. Более того, лучше упорядочивается логика завершения 80 (фиг. 3), так что процессор 10 быстрее определяет, может ли быть "завершена" конкретная команда во время текущего цикла.

В предпочтительном варианте выполнения процессор 10 имеет возможность "завершать" до четырех команд за цикл. Кроме этого в предпочтительном выполнении каждая команда может иметь до двух операндов назначения. Соответственно, если процессор 10 не поддерживает отделение обратной записи от завершения, то процессору 10 необходимо будет использовать восемь портов обратной записи (например, между буферами переименования 34 и GPR 32) для завершения выполнения четырех команд в течение конкретного цикла, если каждая из четырех команд имеет два операнда назначения. Это происходит из-за того, что для "завершения" команды потребуется, чтобы порт обратной записи был доступен для каждого операнда назначения команды, который должен быть скопирован из буфера переименования в назначенный регистр конфигурации.

При использовании меньшего количества портов обратной записи становится более сложной проверка доступности порта обратной записи, так как в течение того же цикла рассматривается большее количество команд для обратной записи. Это происходит

из-за того, что доступность порта обратной записи для конкретной команды в течение конкретного цикла зависит от количества портов обратной записи, используемых для предыдущих команд в течение того же цикла или на предыдущих циклах.

Соответственно, путем отделения обратной записи от завершения логика завершения 80 (фиг. 3) процессора 10 становится более упорядоченной. Это происходит из-за того, что "завершение" команды зависит от следующих условий:

условие 1 - исполнительный блок (которому передается команда) завершает выполнение этой команды;

условие 2 - не возникло ни одной исключительной ситуации, связанной с какой-либо стадией обработки данной команды; а также

условие 3 - какая-либо из переданных команд удовлетворяет условию 1 и условию 2.

Посредством отделения завершения от обратной записи процессор 10 копирует информацию операнда назначения завершенной команды из буфера переименования для хранения в архитектурном регистре в течение конкретного цикла, если порт обратной записи доступен в течение этого цикла. Если порт обратной записи не доступен во время этого цикла, то процессор 10 копирует информацию операнда назначения завершенной команды из буфера переименования в регистр конфигурации во время следующего цикла, когда порт обратной записи доступен.

Согласно фиг. 6 буфер переупорядочения 76 запоминает информацию для четырех команд, завершающих выполнение во время одного и того же цикла процессора 10. На фиг. 7 показаны различные стадии четырех команд INST x, INST x+1, INST x+2, а также INST x+3, которые соответствующим образом назначаются номерам 7, 8, 9 и 10 буфера переупорядочения фиг. 6. Соответственно, команда INST x имеет один операнд назначения ("количество назначений GPR" = 1). Кроме этого, команда INST x+1 имеет один операнд назначения ("количество назначений GPR" = 1). В отличие от этого команда INST x+2 имеет два операнда назначения ("количество назначений GPR" = 2). Аналогично, команда INST x+3 имеет два операнда назначения ("количество назначений GPR" = 2). Как показано на фиг. 7, каждая из команд INST x, INST x+2 и INST x+3 завершает свое выполнение к концу цикла 4.

Фиг. 8a-d являются концептуальными иллюстрациями буферов переименования 34 процессора 10. Для простоты далее рассматривается только работа буферов переименования с фиксированной запятой 34 в качестве примера работы буферов переименования с плавающей запятой 38. Как показано на фиг. 8a-d, буферы переименования 34 содержат двенадцать буферов переименования, соответственно обозначенных номерами буферов 0-11. Блок упорядочивания 18 назначает номера 0-11 передаваемым командам по принципу "первым пришел - первым обслужен" циклическим образом, так что блок 18 назначает номер буфера переименования 0, за которыми следуют номера 1-11 буферов

переименования, а затем вновь номер 0 буфера переименования.

Согласно фиг. 8a-d буфер переименования 2 назначается для хранения информации операнда назначения для команды INST x. Буфер переименования 3 назначается для хранения информации операнда назначения для команды INST x+1. Так как команда INST x+2 имеет два операнда назначения, то оба буфера переименования 4 и 5 назначаются для хранения информации операнда назначения для команды INST x+2. Аналогично, оба буфера переименования 6 и 7 назначаются для хранения информации операнда назначения для команды INST x+3.

На фиг. 8a показано состояние указателя назначения 80, указателя обратной записи 82, а также указателя завершения 84 в начале циклов 4 и 5 по фиг. 7.

Процессор 10 поддерживает такие указатели для управления чтением и записью в буферы переименования 34. Процессор 10 поддерживает указатель назначения 80 для указания, назначен ли буфер переименования конкретной команде. Как показано на фиг. 8a, указатель назначения 80 указывает на буфер переименования 8, указывая тем самым, что буфер переименования 8 является следующим буфером переименования, доступным для назначения команде.

В соответствии с важным аспектом варианта предпочтительного выполнения процессор 10 поддерживает указатель обратной записи 82 для указания, доступен ли буфер переименования (ранее назначены конкретной команде) для переназначения другой команде. Как показано на фиг. 8a, указатель обратной записи 82 указывает на буфер переименования 2, тем самым указывая, что буфер переименования 2 является следующим буфером переименования, из которого процессор 10 будет копировать информацию операнда назначения (которая находилась в поле "информация" буфера переименования фигуры 8a) в одно из GPR 32 (в соответствии с тем, что указано в поле "номер регистра" буфера переименования по фиг. 8a).

Соответственно, процессор 10 продвигает указатель обратной записи 82 (после буфера переименования, ранее назначенного конкретной команде) в ответ на копирование процессором 10 результата (информации операнда назначения) выполнения конкретной команды из буфера переименования для хранения в регистре конфигурации. Таким образом, процессор 10 принимает назначенный буфер переименования для хранения результата (информации операнда назначения) конкретной команды до тех пор, пока процессор 10 не скопирует результат в регистр конфигурации.

Кроме этого, процессор 10 поддерживает указатель завершения 84 для указания (буферу переименования, ранее назначенному конкретной команде) удовлетворяет ли конкретная команда следующим условиям:

условие 1 - исполнительный блок (которому была передана команда) завершил выполнение команды;

условие 2 - не встретилось ни одной исключительной ситуации в связи с какой-либо стадией обработки данной

команды; а также

условие 3 - какая-либо из переданных команд удовлетворяет условию 1 и условию 2.

Как показано на фиг. 8а, указатель завершения 84 указывает на буфер переименования 2, тем самым указывая, что буфер переименования 2 является следующим буфером переименования, который имеет возможность удовлетворить условиям 1, 2 и 3. В соответствии с важным аспектом предпочтительного выполнения процессор 10 поддерживает указатель завершения 84 независимо от того, был ли скопирован результат выполнения команды из буфера переименования для хранения в регистре конфигурации.

Соответственно, "входы переименования" могут быть определены как буфер переименования, на который указывает указатель завершения 84, и как его последующие буферы переименования, которые предшествуют буферу переименования, указанному указателем назначения 80. "Входы обратной записи" могут быть определены как буфер переименования, на который указывает указатель обратной записи 82, и как его последующие буферы переименования, которые предшествуют буферу переименования, на который указывает указатель завершения 84. Входы обратной записи хранят результаты выполнения команд, которые "завершились", но результаты которых не были скопированы из буферов переименования в регистры конфигурации, например, из-за недоступности портов записи в регистры конфигурации.

Концептуально, входы обратной записи расположены между входами переименования и регистрами конфигурации. Преимуществом является то, что результат может обойти входы обратной записи и непосредственно записаться в регистры конфигурации, если порт обратной записи доступен на стадии завершения. Более того, подобно входам переименования процессор 10 обрабатывает входы обратной записи для выдачи информации на исполнительный блок в ситуациях, когда исполнительный блок выполняет команду, определяющую регистры конфигурации, связанные с такой информацией.

Например, на фиг. 8d показано состояние указателя назначения 80, указателя обратной записи 82, а также указателя завершения 84 на начальном цикле 6 по фиг. 7. Как показано на фиг. 8b, указатель назначения 80 не изменился, так как процессор 10 не передал дополнительных команд. В отличие от этого указатель завершения 84 был продвинут от регистра переименования под номером 2 к регистру переименования под номером 8, тем самым указывая на завершение в течение цикла 5 четырех команд INST x, INST x+1, INST x+2 и INST x+3, которые в совокупности имеют шесть операндов назначения.

Кроме того, как показано на фиг. 8b, указатель обратной записи 82 был продвинут от регистра переименования под номером 2 к регистру переименования под номером 4, тем самым указывая на обратную запись в течение цикла 5, информации операндов назначения для команд ISNT x и INST x+1. На фиг. 7 этот факт проиллюстрирован путем

указания завершения и обратной записи ("COMP/WBACK"), которые выполняются одновременно в течение цикла 5 для команд INST x и INST x+1. Таким образом, результаты (в регистре переименования под номером 2 и регистре переименования под номером 3) команд INST x и INST x+1 обходят входы обратной записи и непосредственно записываются в GPR 32, так как два порта обратной записи были доступны в начале цикла 5. Согласно фиг. 8b указатель обратной записи 92 не продвигается за регистр переименования под номером 4, так как оба порта обратной записи используются в течение цикла 5 для обратной записи результатов выполнения команд ISNT x и INST x+1.

На фиг. 8c показано состояние указателя назначения 80, указателя обратной записи 82 и указателя завершения 84 в начале цикла 7 по фиг. 7. Как показано на фиг. 7 и 8c, оба порта обратной записи используются в течение цикла 6 для обратной записи обоих результатов команды INST x+2. Соответственно указатель обратной записи 82 продвигается от регистра переименования под номером 4 к регистру переименования под номером 6. Указатель назначения 80 не изменяется, так как процессор 10 не передал дополнительных команд. Кроме этого, указатель завершения 84 тоже не изменяется, так как процессор 10 не завершил дополнительных команд.

На фиг. 8d показано состояние указателя назначения 80, указателя обратной записи 82 и указателя завершения 84 в начале цикла 8 по фиг. 7. Как показано на фиг. 7 и 8c, оба порта обратной записи используются в течение цикла 7 для обратной записи обоих результатов выполнения команды ISNT x+3. Соответственно, указатель обратной записи 82 передвигается от регистра переименования под номером 6 к регистру переименования под номером 8. Указатель назначения 80 не изменяется, так как процессор 10 не передал дополнительных команд. Кроме этого, указатель завершения 84 тоже не изменяется, так как процессор 10 не завершил выполнение дополнительных команд.

На фиг. 9 проиллюстрированы различные из состояний четырех команд в ситуации без использования существенных признаков предпочтительного варианта выполнения, в котором процессор 10 отделяет завершение от обратной записи. Для иллюстрации такой ситуации, на фиг. 9 показаны различные стадии четырех команд INST y, INST y+1, INST y+2 и INST y+3, которые соответствующим образом назначены номерам 7, 8, 9 и 10 буферов переупорядочения по фиг. 6.

Фиг. 10a-d являются концептуальными иллюстрациями буферов переименования процессора в соответствии с фиг. 9. Более конкретно, на фиг. 10a показано состояние указателя назначения 90 и указателя завершения 94 в начале циклов 4 и 5 по фиг. 9. На фиг. 10b показано состояние указателя назначения 90 и указателя завершения 94 в начале цикла 6 по фиг. 9. На фиг. 10c показано состояние указателя назначения 90 и указателя завершения 94 в начале цикла 7 по фиг. 9. На фиг. 10d показано состояние указателя назначения и указателя

завершения 94 в начале цикла 8 по фиг. 9.

Как показано на фиг. 10a-d, без использования существенных признаков предпочтительного исполнения, где процессор 10 отделяет завершение от обратной записи, конкретная команда (имеющая назначенный буфер переименования) должна завершиться только после того, как результат выполнения конкретной команды будет на самом деле скопирован из назначенного буфера переименования для хранения в регистре конфигурации. В отличие от этого в соответствии с важным свойством предпочтительного исполнения, в соответствии с которым процессор 10 отделяет завершение от обратной записи, процессор 10 далее поддерживает указатель обратной записи 82 для указания, является ли буфер переименования (ранее назначенный конкретной команде) доступным для повторного назначения другой команде. Более того, процессор 10 "завершает" конкретную команду независимо от того, был ли результат выполнения конкретной команды на самом деле скопирован из назначенного буфера переименования для хранения в архитектурном регистре. Соответственно, процессор 10 имеет возможность "завершить" до четырех команд в течение конкретного цикла, даже если каждая из четырех команд имеет два операнда назначения, а также даже тогда, когда в GPR 32 копируются не все операнды назначения в течение конкретного цикла.

На фиг. 11 представлена блок схема альтернативного исполнения буферов переименования 34, в которых хранятся "входы переименования" в буфере 110, отделенном от "входов переименования". "Входы переименования" хранятся в буфере 112. Информация от "входа переименования" буфера 112 передается для хранения на "вход обратной записи" буфера 110 в течение конкретного цикла в ответ на завершение, в течение конкретного цикла, конкретной команды для которой был назначен "вход переименования". Информация из "входа обратной записи" буфера 110 выдается для хранения в одном из GPR 32 в течение конкретного цикла в ответ на доступность, в течение конкретного цикла, одного из двух портов обратной записи для GPR 32.

Буфер 112 передает информацию на буфер 110 посредством одного из восьми портов, показанных на фиг. 11, так что буфер 112 имеет возможность передавать информацию на буфер 110 от "входов переименования" в количестве до восьми в течение любого конкретного цикла процессора 10. Соответственно, процессор 10 имеет возможность "завершить" до четырех команд в течение конкретного цикла, даже если каждая из четырех команд имеет два операнда назначения и даже если в GPR 32 копируются не все операнды назначения в течение конкретного цикла. Предпочтительным является то, что результат (информация операнда назначения) может обойти "входы обратной записи" буфера 110 и записаться непосредственно в GPR 32, если порт обратной записи доступен на стадии завершения. Процессор 10 выполняет такой обход посредством соответствующей работы

мультиплексоров 113a и 113b, которые связаны с буферами 110 и 112, а также с GPR 32, как показано на фиг. 11.

Преимущество альтернативного исполнения буферов переименования 34 по фиг. 11 заключается в том, что вход переименования в буфере 112 может быть повторно назначен другой команде после завершения (пока еще до обратной записи) команды, ранее назначенной входам переименования (для которой вход переименования был назначен ранее), так как информация входа переименования передается соответствующим образом для хранения на входе обратной записи буфера 110 в ответ на завершение ранее назначенной команды. В соответствии с другим аспектом фиг. 11 мультиплексор 114 соответствующим образом передает выбранную информацию на исполнительные блоки от буфера 110 или от буфера 112 в ситуациях, когда исполнительный блок выполняет команду, определяющую регистр конфигурации, связанный с такой информацией. Хотя процессор 10 предпочтительного исполнения имеет два порта обратной записи между буферами переименования и регистрами конфигурации, но подходящее количество портов обратной записи для конкретного исполнения является функцией вероятности переполнения входа переименования и входа обратной записи, что вызовет задержку передачи команды.

Как было указано ранее, со ссылкой на фиг. 2-5 процессор 10 задерживает выполнение последовательно выполняемой команды по меньшей мере до тех пор, пока не завершатся все предыдущие команды в двух ситуациях. В первой ситуации команда не должна выполняться теоретически. Во второй ситуации команда определяет по меньшей мере один исходный операнд, для которого процессор 10 не использует схему продвижения.

В соответствии с первой ситуацией некоторые команды действуют на регистры конфигурации специального назначения ("SPR") 40 (фиг. 1), когда результаты выполнения команд непосредственно записываются в SPR 40 без промежуточного хранения в буферах переименования. Примером таких команд является команда Move To, которая передвигает информацию в один из SPR 40 от одного из GPR 32. Как показано на фиг. 1, такие команды Move To выполняются посредством CFXU 26. Команда Move To немедленно обновляет один из SPR 40 во время выполнения. Аналогично, команда Store немедленно обновляет ячейку памяти в кэше данных 16 (фиг. 1) во время выполнения. Процессор 10 теоретически не выполняет таких команд (когда результаты выполнения команд записываются непосредственно в регистры конфигурации или ячейки памяти без промежуточного хранения в буферах переименования), так что процессор 10 имеет возможность обеспечить точные прерывания и точные исключительные ситуации. Соответственно, для гарантии завершения и обратной записи в порядке очереди процессор 10 задерживает выполнение команды Move To и команды Store до завершения выполнения всех предыдущих команд.

Если для команды, переданной на

исполнительный блок, соответствующий бит ЕОК (в поле "типа команды" входа, назначенного команде в буфере переупорядочения 76) устанавливается в состояние логического 0, то логика последовательного выполнения 84 выдает сигнал (по одной из соответствующих линий 60, 86, 88, 90, 92 и 94, связанных с исполнительным блоком) в ответ на "завершение" всех команд, предшествующих переданной команде. Исполнительный блок выполняет такие переданные команды только в ответ на сигнал от логики последовательного выполнения 84.

Фиг. 12а-с представляют собой концептуальную иллюстрацию буфера изменения порядка 76. Фиг. 13 является иллюстрацией различных стадий пяти команд по фиг. 12а-с. Фиг. 14а-г являются концептуальной иллюстрацией буферов переименования 34 процессора 10.

На фиг. 12а показано состояние буфера переупорядочения 76 в начале цикла 4 по фиг. 13. Соответственно на фиг. 12а буфер переупорядочения 76 хранит информацию для четырех команд, переданных в течение цикла 3 по фиг. 13. На фиг. 13 показаны различные стадии четырех команд INST а, INST а+1, INST а+2 и INST а+3, которые соответствующим образом назначаются номерам 3, 4, 5 и 6 буфера переупорядочения по фиг. 12а.

Как показано на фиг. 12а, команда INST а передается на FXUA 22 и имеет два операнда назначения ("количество назначений GPR" = 1). Команда INST а+2 является командой Store, передаваемой на LSU 28 и имеет нулевые операнды назначения ("количество назначений GPR" = 0); более того, буфер переупорядочения под номером 5 (назначенный команде INST а+2) имеет ЕОК = 0 в ответ на выполнение команды INSTG а+2, которая является последовательно выполняемой. Команда INST а+3 передается на CFUX 26 и имеет один операнд назначения ("количество назначений GPR" = 1).

В соответствии с важным аспектом предпочтительного исполнения блок упорядочивания 18 определяет, возможна ли исключительная ситуация вследствие выполнения команды. Блок упорядочивания 18 осуществляет это определение заранее, до выполнения команды. Если исключительная ситуация невозможна для этой команды, то блок 18 в основном устанавливает (в ответ на передачу команды) "завершение" = 1 для входа, связанного с буфером переупорядочения команды независимо от того, завершил ли на самом деле процессор 10 выполнение команды, выполнение которой невозможно.

Более того, в такой ситуации процессор 10 "завершает" выполнение команды в ответ на завершение процессором 10 всех предыдущих команд независимо от того, завершил ли на самом деле процессор 10 выполнение команды, для которой невозможна исключительная ситуация. Соответственно, процессор 10 "завершает" выполнение команды в ответ на определение, что исключительная ситуация не следует из выполнения команды и из выполнения каждой предыдущей команды, предшествующей команде в запрограммированной последовательности независимо от того,

завершил ли процессор 10 выполнение команд, выполнение которых невозможно. Таким образом, возможно предшествование состояния завершения выполнения команды состоянию выполнения команды; так процессор 10 поддерживает "раннее завершение" команды.

Поддерживая "раннее завершение", процессор 10 в основном имеет возможность быстрее выполнять следующие, последовательно выполняемые команды (такие, как Move To или команды Store), причем результаты выполнения команд записываются непосредственно в регистры конфигурации или в ячейку памяти без промежуточного хранения в буферах переименования). Все это верно независимо от возможностей процессора 10 обеспечивать точные прерывания и точные исключительные ситуации. Это объясняется тем, что все предыдущие команды либо (1) уже закончили свое выполнение без исключительных ситуаций, либо (2) закончат выполнение без исключительных ситуаций. Посредством такого способа процессор 10 продолжает обеспечивать завершение и обратную запись в порядке очереди.

Например, согласно фиг. 12а команды INST а+1 и INST а+3 не имеют возможности вызвать исключительную ситуацию, так что "завершение" = 1 в буфере переупорядочения под номером 4 и 6.

На фиг. 14а показано состояние указателя назначения ("AL") 80, указателя обратной записи ("WB") 82, а также указателя завершения ("CO") 84 буферов переименования 34 в начале цикла 4 по фиг. 13. Таким образом, фиг. 14а соответствует фиг. 12а. Как показано на фиг. 14а, WB 82 и CO 84 указывают на буфер переименования 2, а AL 80 указывает на буфер переименования 6. Буферы переименования 2 и 3 назначаются команде INST а. Буфер переименования 4 назначается команде INST а+1, а буфер переименования 5 назначается команде INST а+3. Следует учесть, что команде INST а+2 не назначается буфер переименования, так как команда INST а+2 имеет нулевые операнды назначения. В буферах переименования 2-5, "правильная информация" = 0, указывает, что в поле "информация" этих буферов переименования не содержится правильных данных. В начале цикла 4 (фиг. 13) "правильная информация" = 0, так как команды INST а, INST а+2, а также INST а+3 еще не закончили выполнение.

В буфере переименования 4 "исполнить правильно" = 1, указывая, что INST а+1 изменяет регистр CA 42. Необходимо учитывать, что, как указывалось ранее, команда INST а+1 не имеет возможности вызвать исключение, так что "завершение" = 1 в буфере переупорядочения 4 (фиг. 12а). Соответственно, процессор 10 поддерживает "раннее завершение" команд, которые изменяют регистр CA 42.

Как показано на фиг. 14а-г, информация бита CA временно хранится в буферах переименования 34. Несмотря на это процессор 10 в предпочтительном варианте выполнения не использует схему для продвижения информации такого бита CA к исполнительным блокам от буферов переименования 34. Если информация исходного операнда конкретной команды

содержится в регистре CA 42, то информация исходного операнда конкретной команды зависит от результатов выполнения предыдущей команды (если такая имеется), которая изменяет регистр CA 42.

На фиг. 12b показано состояние буфера переупорядочения 76 в начале цикла 5 по фиг. 13. Соответственно, на фиг. 12b буфер переупорядочения 76 хранит информацию для команды INST a+4, передаваемой в течение цикла 4 по фиг. 13. На фиг. 13 показаны различные стадии команды INST a+4, которая назначается буферу переупорядочения под номером 7. Как показано на фиг. 12b, команда INST a+4 передается на FXUA 22 и имеет один операнд назначения ("количество назначений GPR" = 1); более того, буфер переупорядочения под номером 7 (назначенный команде INST a+4) имеет EOK = 0 в ответ на команду INST a+4, которая выполняется последовательно.

Более того, так как команда INST a завершает свое выполнение в течение цикла 4, то "завершение" = 1 в буфере переупорядочения под номером 3 на фиг. 12b. Согласно фиг. 13, команда INST a+2 является командой Store. В течение цикла 4 LSU 28 завершило первый этап выполнения EXEC A команды Store INST a+2. В течение выполнения EXEC A LSU 28 преобразует команду Store и проверяет команду Store на защиту памяти. Соответственно "завершено" = 1 для буфера переупорядочения под номером 5 на фиг. 12b.

На фиг. 14b показано состояние указателя назначения ("AL") 80 указателя обратной записи ("WB") 82, а также указателя завершения ("CO") 84 буферов переименования 34 в начале цикла 5 по фиг. 13. Таким образом, фиг. 14b соответствует фиг. 12b. Как показано на фиг. 14b, WB 82 и CO 84 продолжает указывать на буфер переименования 2, а AL 80 продвигается от буфера переименования 6 и указывает на буфер переименования 7. Соответственно, буфер переименования 6 назначается команде INST a+4.

В буфере переименования 6 "правильная информация" = 0, что указывает, что поле "информация" этого буфера переименования не хранит правильных данных. В начале цикла 5 (фиг. 13) "правильная информация" = 0 для буфера переименования 6, так как выполнение команды INST a+4 еще не завершено. В отличие от этого для фиг. 14b "правильная информация" = 1 для буферов переименования 2 и 3 в ответ на завершение выполнения команд INST a и INST a+1 в течение цикла 4, как показано на фиг. 13, что указывает, что в полях "информация" буферов переименования 2 и 3 хранятся правильные данные (представленные на фиг. 14b как "данные" в полях "информация").

В буфере переименования 6 "исполнить правильно" = 1, что указывает, что INST a+4 изменяет регистр CA 42. Более того, информация исходного операнда команды INST a+4 содержится в регистре CA 42 так, что INST a+4 является последовательно выполняемой командой, причем информация исходного операнда зависит от результата выполнения предшествующей команды INST a+1, которая изменяет регистр CA 42. Хотя команда INST a+3 не изменяет регистр CA 42, но информация исходного операнда команды

INST a+3 тоже зависит от результатов выполнения предыдущей команды INST a+1, как указывается на фиг. 13 стрелкой 100.

На фиг. 12c показано состояние буфера переупорядочения 76 в начале цикла 6 по фиг. 13. На фиг. 14c показано состояние указателя назначения ("AL") 80, указателя обратной записи ("WB") 82, а также указателя завершения ("CO") 84 буферов переименования 34 в начале цикла 6 по фиг. 13. Таким образом, фиг. 14c соответствует фиг. 12c.

Как показано на фиг. 12c, информация буферов переупорядочения под номерами 3, 4, 5 и 6 была стерта в ответ на выполнение команд INST a, INST a+1, INST a+2 и INST a+3, "завершенных" в течение цикла 5, как показано на фиг. 13. Необходимо учитывать, что команда INST a+1 "ранее выполнена" в течение цикла 6 до выполнения команды INST a+1 в течение цикла 5, что показано на фиг. 13 посредством "ECOMP/EXEC"; команда INST a+1 является членом класса команд, для выполнения которых необходимо более одного цикла процессора 10.

Согласно фиг. 14c процессор 10 продвигает CO 84 после буфера переименования (например после буфера переименования 5 до начала цикла 6), который ранее был назначен конкретной команде (например, команде INST a+3) в ответ на завершение процессором 10 всех предыдущих команд (например, INST a, INST a+1, а также INST a+2), независимо от того, действительно ли процессор 10 завершил выполнение конкретной команды (например INST a+3, выполнение которой продолжается в течение циклов 6-8).

Кроме этого, в течение цикла 5 логика последовательного выполнения 84 (фиг. 3) передает сигнал по линии 92, связанной с LSU 28, в ответ на завершение выполнения команд INST a и INST a+1; это следует из того факта, что для команды INST a+2 (которая была передана на LSU 28 во время цикла 3) соответствующий назначенный бит EOK в буфере переупорядочения 5 установлен в логическое состояние 0.

На цикле 5 "завершается" команда Store a+2 в ответ на завершение (цикл 4) выполнения стадии EXEC A без исключений и в ответ на завершение (цикл 5) предыдущих команд INST a и INST a+1. На второй стадии выполнения EXEC B, LSU 28 действительно записывает информацию в ячейку памяти кэша данных 16. LSU 28 выполняет этап EXEC B команды Store INST a+2 только в ответ на переданный логикой последовательного выполнения 84 сигнал по линии 92.

Как показано на фиг. 14c, AL 80 продолжает указывать на буфер переименования 7, и CO 84 продвигается от буфера переименования 2 и указывает на буфер переименования 6 в ответ на "завершение" предыдущих команд INST a, INST a+1 и INST a+3 в течение цикла 5. Более того, как показано на фиг. 13, оба результата выполнения команды INST a копируются из буферов переименования 2 и 3 в GPR 32 (фиг. 1) в течение цикла 5. Соответственно, как показано на фиг. 14c, WB 82 продвигается из буфера переименования 2 и указывает на буфер переименования 4; более того, информация в буферах

переименования 2 и 3 стирается в ответ на копирование двух результатов выполнения команды INST а из буферов переименования 2 и 3 в GPR 32 (фиг. 1) в течение цикла 5.

На фиг. 14с "правильная информация" = 1 в буфере переименования 4 в ответ на завершение выполнения команды INST а+1 в течение цикла 5, как показано на фиг. 13. Это указывает, что поле "информация" буфера переименования 4 хранит правильные данные (представленные на фиг. 14с как "данные" в поле "информация") и что поле "перенос" или буфер переименования 4 хранит правильные данные (представленные на фиг. 14с как "CA" в поле "перенос"). Так как команда INST а+4 является последовательно выполняемой командой, информация исходного операнда которой зависит от результатов выполнения команды INST а+1, которая изменяет регистр CA 42, и так как поле "перенос" буферов переименования 34 не продвинуто, то процессор 10 задерживает выполнение команды INST а+4 до копирования результата "перенос" команды INST а+1 из буфера переименования 4 в GPR 32 (фиг. 1) на стадии обратной записи (WBACK) команды INST а+1 на цикле 6.

В отличие от этого информация исходного операнда команды INST а+3 тоже зависит от данных, которые являются результатом выполнения команды INST а+1, но поле "информация" буферов переименования 34 продвигается. Соответственно, в ответ на "правильная информация" = 1 в буфере переименования 4 (что указывает на завершение выполнения команды INST а+1 на цикле 5) процессор 10 начинает выполнение команды INST а+3 в течение цикла 6 до завершения стадии обратной записи команды INST а+1.

На фиг. 14d показано состояние указателя назначения ("AL") 80 указателя обратной записи ("WB") 82 и указателя завершения ("CO") 84 буферов переименования 34 в начале цикла 7 по фиг. 13. Как показано на фиг. 14d, AL 80 продолжает указывать на буфер переименования 7, а CO 84 продолжает указывать на буфер переименования 6. Как показано на фиг. 13, результат выполнения команды INST а+1 был скопирован из буфера переименования 4 в GPR 32 (фиг. 1) в течение цикла 6. Соответственно, как показано на фиг. 14d, WB 82 был продвинут от буфера переименования 4 и указывает на буфер переименования 5; более того, информация в буфере переименования 4 была стерта в ответ на копирование результата выполнения команды INST а+1 из буфера переименования 4 в GPR 32 (фиг. 1) в течение цикла 6.

В предпочтительном исполнении для последовательно выполняемой команды (например INST а+4), информация исходного операнда которой зависит от неподвинутого операнда (например, информация бита CA, предназначенная для регистра CA 42), процессор 10 выполняет последовательно выполняемую команду (например, INST а+4) в ответ на следующие указания буферов переименования 34:

- (1) все предыдущие команды "завершены"; и
- (2) если состояние обратной записи какой-либо предшествующей "завершенной"

команды еще не завершилось, то любая такая предыдущая команда не должна изменять неподвинутый операнд.

Соответственно, на фиг. 14d CO 84 указывает на буфер переименования 6, тем самым указывая команде INST а+4, что выполнение всех предыдущих команд "завершено". Более того, на фиг. 14d WB 82 указывает на буфер переименования 5, тем самым указывая, что хотя состояние обратной записи предыдущей "завершенной" команды INST а+3 еще не завершено, но команда INST а+3 не изменяет какую-либо неподвинутую информацию, предназначенную для регистра CA 42 (так как "перенос подтвержден" = 0 в буфере переименования 5). Таким образом, процессор 10 выполняет команду INST а+4 в течение цикла 7.

В альтернативном исполнении для последовательно выполняемой команды (например, INST а+4), информация исходного операнда которой зависит от неподвинутого операнда (например, информация бита CA, предназначенная для регистра CA 42), процессор 10 выполняет эту последовательно выполняемую команду (например, INST а+4) в ответ на следующие указания буфером переименования:

- (1) "завершены" все предыдущие команды; и

- (2) завершены состояния обратной записи всех предшествующих "завершенных" команд, так что WB 82 и CO 84 указывают на один и тот же буфер переименования.

На фиг. 14е показано состояние указателя назначения ("AL") 80, указателя обратной записи ("WB") 82, а также указателя завершения ("CO") 84 буферов переименования 34 в начале цикла 8 по фиг. 13. Как показано на фиг. 14е, AL 80 продолжает указывать на буфер переименования 7, а WB 82 продолжает указывать на буфер переименования 5. Команда INST а+4 выполнена и "завершена" в течение цикла 7 до завершения выполнения команды INST а+4 в течение цикла 7, как указывается на фиг. 13 посредством "EXEC/COMP". Соответственно, как показано на фиг. 14е, CO 84 был продвинут от буфера переименования 6 и указывает на буфер переименования 7.

Преимуществом является то, что процессор 10 продвигает CO 84 после буфера переименования (например, после буфера 6 до начала цикла 8), ранее назначенного конкретной команде (например, INST а+4), в ответ на завершение процессором 10 всех предыдущих команд (например, INST а, INST а+1 и INST а+3), независимо от того, действительно ли процессор 10 закончил выполнение всех предыдущих команд (например, INST а+3, выполнение которой продолжается в течение цикла 8).

На фиг. 14е, "правильная информация" = 1 в буфере переименования 6 в ответ на завершение выполнения команды INST а+4 в течение цикла 7, как показано на фиг. 13. Это указывает, что в поле "информация" буфера переименования 6 хранятся правильные данные (представленные на фиг. 14е как "данные" в поле "информация") и что в поле "перенос" буфера переименования 6 хранятся правильные данные (представленные на фиг. 14е как "CA" в поле

"выполнить").

На фиг. 14f показано состояние указателя назначения ("AL") 80, указателя обратной записи ("WB") 82 и указателя завершения ("CO") 84 буферов переименования 34 в начале цикла 9 по фиг. 13. Как показано на фиг. 14f, AL 80 и CO 84 продолжают указывать на буфер переименования 7, а WB 82 продолжает указывать на буфер переименования 5. На фиг. 14f "правильная информация" = 1 в буфере переименования 5 в ответ на завершение выполнения команды INST a+3 в течение цикла 8. Это указывает, что в поле "информация" буфера переименования 5 хранятся правильные данные (представленные на фиг. 14e как "данные" в поле "информация").

На фиг. 14g показано состояние указателя назначения ("AL") 80, указателя обратной записи ("WB") 82 и указателя завершения ("CO") 84 буферов переименования 34 в начале цикла 10 по фиг. 13. Как показано на фиг. 14, AL 80 и CO 84 продолжают указывать на буфер переименования 7. Как показано на фиг. 13, результаты выполнения команд INST a+3 и INST a+4 копируются из буферов переименования 5 и 6, соответственно, в GPR 32 (фиг. 1) в течение цикла 9. Соответственно, как показано на фиг. 14g, WB 82 продвигается от буфера переименования 5 и указывает на буфер переименования 7; более того, информация буферов переименования 5 и 6 была стерта в ответ на копирование результатов выполнения команд INST a+3 и INST a+4 из буферов переименования 5 и 6 в GPR 32 (фиг. 1) в течение цикла 9.

Фиг. 15 является иллюстрацией различных стадий пяти команд фиг. 12a-c в альтернативном исполнении без свойства "раннего завершения", соответствующего предпочтительному выполнению, фиг. 16a-h являются концептуальной иллюстрацией буферов переименования 34 процессора 10 на различных циклах обработки команды, проиллюстрированных на фиг. 15. Как показано на фиг. 15, без свойства "раннего завершения" предпочтительного исполнения выполнение команды INST a+4 задерживается до цикла 10, а завершение/обратная запись результатов выполнения команды INST a+4 задерживается до цикла 11. Это следует из того факта, что команда INST a+4 является последовательно выполняемой командой, выполнение которой задерживается до завершения выполнения всех предыдущих команд. Без свойства "раннего завершения", соответствующего предпочтительному выполнению, команда INST a+3 не завершается до цикла 9, тем самым задерживая выполнение команды INST a+4 до цикла 10.

В отличие от этого на фиг. 13 команда INST a+3 "завершается ранее" в течение цикла 5, а команда INST a+4 выполняется в течение цикла 7. Команда INST a+4 является последовательно выполняемой командой, информация исходного операнда которой зависит от результата выполнения предыдущей команды INST a+1, которая изменяет информацию неподвижного бита CA, предназначенную для регистра CA 42. Несмотря на это, на фиг. 13 выполнение команды INST a+4 осуществляется в течение

цикла 7, а завершение/обратная запись результатов выполнения команд INST a+4 осуществляется в течение цикла 9. Соответственно, на фиг. 13 с учетом свойства "раннего завершения", соответствующего предпочтительному выполнению, этапы выполнения, завершения и обратной записи команды INST a+4 осуществляются раньше, чем на фиг. 15, без учета свойства "раннего завершения".

Процессор 10 задерживает выполнение команды, определяющей неподвижные исходные операнды (например SPR 40 и регистр CA 42) по меньшей мере до завершения всех предыдущих команд. Это происходит потому, что там, где команда определит неподвижные исходные операнды, выполнение команды осуществляется в соответствии с информацией такого исходного операнда, которая доступна и верна в регистрах конфигурации. Примерами таких команд могут быть (1) расширенная арифметическая операция, которая считывает регистр CA 42, и (2) команда Move From, которая сдвигает информацию от одного из SPR 40 к одному из GPR 32.

Даже если "завершено" выполнение всех предыдущих команд, имеется возможность для "раннего завершения" предыдущей команды так, чтобы состояние ее завершения предшествовало состоянию ее выполнения. Как указывалось ранее со ссылкой на фиг. 1-5, процессор 10 содержит схему для продвижения информации к блокам выполнения от GPR 32, FPR 36, буферов переименования 34 и 38 и регистров управления. Если информация исходного операнда конкретной команды зависит от результатов выполнения предыдущих "раннее завершенных" команд и если информация исходного операнда обслуживается такой схемой продвижения процессора 10, то процессор 10 задерживает выполнение конкретной команды до завершения процессором 10 стадии выполнения "раннее завершенной" команды (когда результаты выполнения "раннее завершенной" команды будут доступными и правильными в буфере переименования).

В отличие от этого, если информация исходного операнда не обслуживается такой схемой продвижения процессора 10, то процессор 10 задерживает выполнение конкретной команды до завершения процессором 10 этапа обратной записи "раннее завершенной" команды (к моменту, когда результаты выполнения "раннее завершенной команды" будут доступными и правильными в регистре конфигурации). Если "раннее завершенная" команда не завершила выполнение, то результат выполнения команды (информация операнда назначения) не доступна. В такой ситуации процессор 10 соответствующим образом задерживает этап обратной записи "раннее завершенной" команды до завершения выполнения команды.

Если команда подвергается раннему завершению, то блок упорядочивания 18 выдает сигнал указания во время передачи команды на назначенный исполнительный блок команды. В ответ на такой сигнал указания назначенный исполнительный блок команды не предпринимает попыток изменить

назначенный команде вход в буфере переупорядочения 76. Это происходит из-за того, что по завершении команды, назначенной входу буфера переупорядочения, вход буфера переупорядочения подвергается повторному назначению другой команде, так что информация входа буфера переупорядочения более не верна.

Соответственно, если команда подвергается раннему завершению, то назначенный команде вход буфера переупорядочения может быть верным до завершения выполнения команды исполнительным блоком этой команды. В предпочтительном исполнении в течение одного цикла процессора 10 блок упорядочения 18 проверяет четыре наиболее старые входа буфера переупорядочения, которые имеют правильную информацию, так что эти четыре команды являются кандидатами на одновременное завершение в течение одного цикла.

В предпочтительном исполнении команда Move From вводит информацию от SPR, а SPR изменяется только командой Move To. Полезно, чтобы процессор 10 выполнял команду Move From в ответ на завершение выполнения всех предыдущих команд, даже если одна или более предыдущих команд - "ранее завершены". Это возможно из-за того, что процессор 10 не "завершил ранее" команду (например, команду Move To или команду Store), которая не должна выполняться теоретически. Если предыдущие команды "завершены", то процессор 10 действительно завершил выполнение всех предыдущих команд Move To. Соответственно, процессор 10 выполняет команду Move From в ответ на завершение всех предыдущих команд, так как в такой ситуации процессор 10 на самом деле завершил выполнение всех предыдущих команд Move To.

В соответствии с другим важным аспектом варианта предпочтительного выполнения процессор 10 поддерживает режим исключительной ситуации неточной невосстанавливаемой плавающей запятой. Во время работы в таком режиме процессор 10 оповещает о состоянии исключительной ситуации с плавающей запятой, которая не точна или не информативна, при котором команда с плавающей запятой породила исключительную ситуацию. При обработке процессором 10 исключительной ситуации допускается, чтобы процессор 10 находился в состоянии, в котором он завершил различное количество команд до выполнения команды с плавающей запятой, которая вызвала исключительную ситуацию. При работе в этом режиме процессор 10 обеспечивает высокую производительность при выполнении последовательностей команд, которые нечувствительны к останову выполнения программы точно на команде с плавающей запятой, обусловившей состояние исключительной ситуации.

При работе в режиме исключительной ситуации с точной плавающей запятой процессор 10 не выполняет "раннее завершение" команды с плавающей запятой до завершения выполнения команды с плавающей запятой, даже если все предыдущие команды были завершены.

В отличие от этого при работе в режиме исключительной ситуации с неточной невосстанавливаемой плавающей запятой процессор 10 выполняет "раннее завершение" команды с плавающей запятой до завершения выполнения команды с плавающей запятой в ответ на завершения всех предыдущих команд. В режиме исключительной ситуации с неточной невосстанавливаемой плавающей запятой "раннее завершение" команды с плавающей запятой отличаются от "раннее завершенных" команд другого типа, так как процессор 10 "раннее завершает" команду с плавающей запятой, даже если последующее выполнение команды с плавающей запятой может фактически привести к состоянию исключительной ситуации с плавающей запятой. Каждый вход буфера переупорядочения указывает, является ли назначенная ему команда командой с плавающей запятой.

В ответ на завершение стадий выполнения и завершения команды с плавающей запятой результаты их выполнения временно сохраняются в буферах переименования с плавающей запятой 38. Эти результаты временно хранятся в буферах переименования 38 до копирования результатов выполнения всех ранее выполненных команд с плавающей запятой в FPR 36. Так процессор 10 обрабатывает соответствующие стадии "обратной записи" команд с плавающей запятой в порядке их запрограммированной последовательности.

Буферы переименования 38 временно хранят результаты выполнения команд с плавающей запятой для дальнейшего копирования в FPR 36. Кроме того, буферы переименования 38 сохраняют информацию о состоянии для изменения соответствующим образом состояния конфигурации с плавающей запятой и регистров управления ("PFSCR") (не показаны). В ответ на обнаружение исключительной операции с плавающей запятой (что указывается установкой бита FEX FPSCR) во время копирования результатов из буферов переименования 38 и FPR 86 и в FPSCR процессор 10 оповещает об исключительной ситуации.

При работе в режиме исключительной ситуации с неточной невосстанавливаемой плавающей запятой процессор 10 оповещает о состоянии исключительной ситуации с плавающей запятой, которая не является точной или информативной и для которой команда с плавающей запятой вызвала состояние исключительной ситуации. Преимущество состоит в том, что в такой ситуации процессор 10 предпочтительного варианта выполнения оповещает об исключительной ситуации следующим образом.

Во-первых, процессор 10 останавливает выполнение команды в течение цикла, на котором процессор 10 обнаружил исключительную ситуацию с плавающей запятой. Команда, на которой процессор 10 прекратил завершение, не имеет значения, так как состояние процессора 10 точно воспроизводит результаты всех выполненных команд, предшествующих команде, на которой процессор 10 остановил выполнение.

Во-вторых, процессор 10 заканчивает выполнение всех предыдущих "ранее завершённых" команд. Более того, процессор 10 копирует результаты выполнения таких команд в регистры конфигурации так, чтобы все входы обратной записи в буферах переименования 34 и 38 стали пустыми.

В-третьих, процессор 10 оповещает об исключительной ситуации с плавающей запятой. В режиме исключительной ситуации с неточной невосстанавливаемой плавающей запятой "ранее завершённые" команды с плавающей запятой отличаются от других типов "ранее завершённых" команд, так как процессор 10 "ранее завершает" команду с плавающей запятой, даже когда последующее выполнение команды с плавающей запятой фактически может вызвать состояние исключительной ситуации с плавающей запятой. Соответственно, никакие другие исключительные ситуации не генерируются другими командами, которые "ранее завершились" или фактически завершились.

В ответ на обнаружение исключительной ситуации, обусловленной выполнением команды с неплавающей запятой, процессор 10 задерживает сообщение об исключительной ситуации с неплавающей запятой до окончания выполнения процессором 10 всех предыдущих "ранее завершённых" команд с плавающей запятой без каких-либо исключительных ситуаций с плавающей запятой. Предпочтительным образом процессор 10 оповещает о такой исключительной ситуации следующим образом.

Во-первых, процессор 10 останавливает завершение команды в течение цикла, на котором процессор 10 обнаружил исключительную ситуацию с неплавающей запятой. Процессор 10 останавливает завершение команд на команде, вызвавшей исключительную ситуацию с неплавающей запятой, так что эта команда не завершается.

Во-вторых, процессор 10 заканчивает выполнение всех предыдущих "ранее завершённых" команд. Тем самым определяется, может ли процессор 10 закончить выполнение всех предыдущих "ранее завершённых" команд с плавающей запятой без обнаружения какой-либо исключительной ситуации с плавающей запятой. Более того, процессор 10 копирует результаты выполнения таких команд в регистры конфигурации так, чтобы все входы обратной записи в буферах переименования 34 и 38 стали пустыми. Таким образом, имеет место непротиворечивое состояние процессора 10, когда он оповещает об исключительной ситуации. Кроме того, выполняется ожидание освобождения всех других буферов обратной записи для установления непротиворечивого состояния процессора при распознавании исключительной ситуации.

В-третьих, если процессор 10 не обнаружил исключительную ситуацию с плавающей запятой, то он оповещает об исключительной ситуации с точной неплавающей запятой. В отличие от этого, если процессор 10 обнаружил исключительную ситуацию с плавающей запятой, то он оповещает об исключительной ситуации с неточной невосстанавливаемой

плавающей запятой.

Детально описанный пример осуществления настоящего изобретения и его преимущества представлены в качестве примера без какого-либо ограничения. В примере выполнения могут быть допустимы изменения, замены и доработки в соответствии с сущностью и объемом настоящего изобретения. Сущность и объем настоящего изобретения не должны ограничиваться конкретным примером осуществления, а должны определяться в соответствии с формулой изобретения и ее эквивалентами.

Формула изобретения:

1. Способ функционирования системы обработки (10), при котором осуществляют этапы передачи команд ($INST\ x$, $INST\ x + 1$, $INST\ x + 2$, $INST\ x + 3$), причем передают одну из указанных команд на один из исполнительных блоков для исполнения, и определения, возможна ли исключительная ситуация в результате выполнения упомянутой команды соответствующим исполнительным блоком, отличающийся тем, что выдают сигнал, указывающий, что упомянутая команда завершена, в ответ на определение, что исключительная ситуация невозможна в результате выполнения указанной команды соответствующим исполнительным блоком, причем выдача упомянутого сигнала не зависит от того, завершил ли соответствующий исполнительный блок выполнение упомянутой команды.

2. Способ по п.1, отличающийся тем, что упомянутая команда представляет собой конкретную команду, причем этап передачи дополнительно включает передачу команды ($INST\ a + 4$), преобразованной в последовательную форму выполнения, которой в последовательности предшествует упомянутая конкретная команда.

3. Способ по п.1, отличающийся тем, что этап определения включает определение того, возможна ли исключительная ситуация в результате выполнения упомянутой конкретной команды, а этап завершения включает завершение упомянутой команды в ответ на определение того, что исключительная ситуация невозможна в результате выполнения упомянутой команды.

4. Способ по п.2, отличающийся тем, что дополнительно включает этап выполнения соответствующим исполнительным блоком упомянутой последовательно выполняемой команды ($INST\ a + 4$) в ответ на выдачу упомянутого сигнала, прежде чем соответствующий исполнительный блок завершит выполнение упомянутой конкретной команды.

5. Способ по п.4, отличающийся тем, что последовательно выполняемой команде ($INST\ a + 4$) предшествует в упомянутой последовательности другая команда, и упомянутая последовательно выполняемая команда информационно зависит от результата этой другой команды.

6. Способ по п. 5, отличающийся тем, что упомянутый этап выполнения включает выполнение упомянутой последовательно выполняемой команды ($INST\ a + 4$) в ответ на выдачу упомянутого сигнала, при условии, что соответствующий исполнительный блок завершил выполнение упомянутой другой

команды.

7. Способ по п.4, отличающийся тем, что дополнительно включает этап выполнения первой стадии упомянутой последовательно выполняемой команды (INST a + 4) до завершения выдачи упомянутого сигнала.

8. Способ по п.7, отличающийся тем, что дополнительно включает этап выполнения второй стадии упомянутой последовательно выполняемой команды (INST a + 4) в ответ на выдачу упомянутого сигнала до завершения выполнения соответствующим исполнительным блоком упомянутой конкретной команды.

9. Способ по п.8, отличающийся тем, что упомянутая последовательно выполняемая команда является командой хранения (INST a + 2).

10. Способ по п.1, отличающийся тем, что дополнительно включает этап запоминания указателя в буфере (4, 6) переупорядочения в ответ на определение невозможности исключительной ситуации в результате выполнения соответствующим исполнительным блоком упомянутой команды.

11. Способ по п.10, отличающийся тем, что упомянутый этап запоминания включает хранение упомянутого указателя в упомянутом буфере переупорядочения при условии, что упомянутая команда соответствует типу теоретически выполняемой команды.

12. Способ по п.10, отличающийся тем, что дополнительно включает этап стирания упомянутого указателя в упомянутом буфере (3, 4, 5, 6) в ответ на выдачу упомянутого сигнала.

13. Способ по п.1, отличающийся тем, что дополнительно включает этап запоминания указателя в буфере (38) переупорядочения в ответ на конкретную команду, которая является командой с плавающей запятой для выполнения в режиме исключительной ситуации с неточной невосстанавливаемой плавающей запятой, даже если исключительная ситуация возможна в результате выполнения упомянутой команды с плавающей запятой.

14. Способ по п.1, отличающийся тем, что упомянутая команда представляет собой конкретную команду, а упомянутый этап выдачи включает выдачу упомянутого сигнала в ответ на определение того, что исключительная ситуация не возможна в результате выполнения соответствующим исполнительным блоком упомянутой команды, если по меньшей мере одно из следующих условий (а) и (b) истинно для каждой предшествующей команды, которая предшествует упомянутой конкретной команде в последовательности команд, передаваемых к одному из исполнительных блоков: (а) исключительная ситуация невозможна в результате выполнения соответствующим исполнительным блоком упомянутой предшествующей команды, и (b) исключительная ситуация не является результатом выполнения соответствующим исполнительным блоком упомянутой предшествующей команды.

15. Способ по п. 1, отличающийся тем, что дополнительно включает этап определения, является ли упомянутая команда командой с плавающей запятой для выполнения

соответствующим исполнительным блоком в режиме исключительной ситуации с неточной невосстанавливаемой плавающей запятой, причем упомянутый этап выдачи сигнала включает этап выдачи упомянутого сигнала в ответ на определение того, что упомянутая команда является командой с плавающей запятой для выполнения ее соответствующим исполнительным блоком в упомянутом режиме исключительной ситуации с неточной невосстанавливаемой плавающей запятой, причем выдача упомянутого сигнала не зависит от того, завершил ли соответствующий исполнительный блок выполнение упомянутой команды.

16. Способ по п.1, отличающийся тем, что упомянутая команда представляет собой команду с плавающей запятой.

17. Способ по п.16, отличающийся тем, что упомянутая команда с плавающей запятой выполняется соответствующим исполнительным блоком в режиме исключительной ситуации с неточной невосстанавливаемой плавающей запятой.

18. Способ по п.16 или 17, отличающийся тем, что дополнительно включает этап запоминания указателя в буфере переупорядочения в ответ на определение того, что упомянутая команда является командой с плавающей запятой для выполнения соответствующим исполнительным блоком в упомянутом режиме исключительной ситуации с неточной невосстанавливаемой плавающей запятой.

19. Способ по п.18, отличающийся тем, что дополнительно включает этап стирания упомянутого указателя в буфере переупорядочения в ответ на выдачу упомянутого сигнала.

20. Способ по п.16 или 17, отличающийся тем, что упомянутый этап выдачи упомянутого сигнала включает этап выдачи упомянутого сигнала в ответ на определение того, что упомянутая команда представляет собой команду с плавающей запятой для выполнения соответствующим исполнительным блоком в упомянутом режиме исключительной ситуации с неточной невосстанавливаемой плавающей запятой, если по меньшей мере одно из следующих условий (а), (b) и (с) истинно для каждой предшествующей команды, которая предшествует упомянутой команде с плавающей запятой в последовательности команд, передаваемых на соответствующий исполнительный блок: (а) упомянутая предшествующая команда представляет собой команду с плавающей запятой для выполнения соответствующим исполнительным блоком в упомянутом режиме с неточной невосстанавливаемой плавающей запятой, (b) исключительная ситуация невозможна в результате выполнения соответствующим исполнительным блоком упомянутой предшествующей команды, и (с) исключительная ситуация не является результатом выполнения соответствующим исполнительным блоком упомянутой предшествующей команды.

21. Способ по п. 16 или 17, отличающийся тем, что упомянутая команда представляет собой конкретную команду, причем этап передачи дополнительно включает передачу последовательно выполняемой команды

(INST x + 4), которой в последовательности предшествует упомянутая конкретная команда.
22. Способ по п.21, отличающийся тем, что дополнительно включает этап выполнения упомянутой последовательно выполняемой

команды (INST x + 4) соответствующим исполнительным блоком в ответ на выдачу упомянутого сигнала, прежде чем соответствующий исполнительный блок завершит выполнение упомянутой конкретной команды.

5

10

15

20

25

30

35

40

45

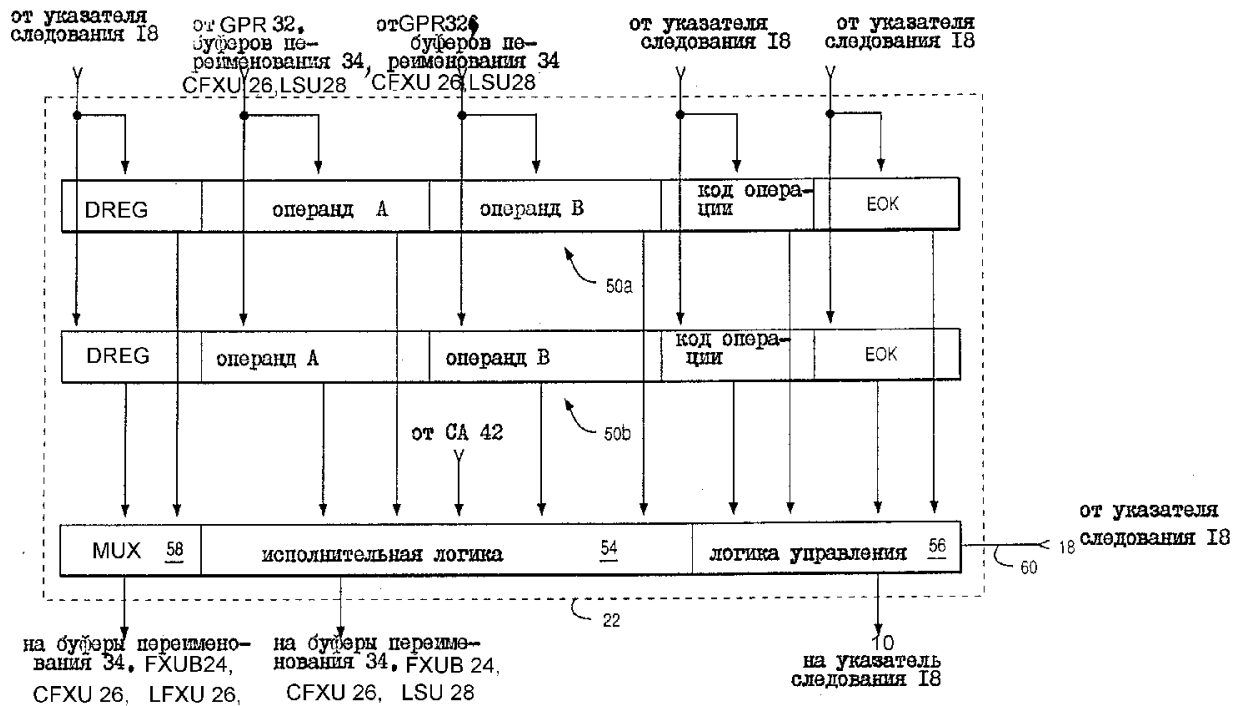
50

55

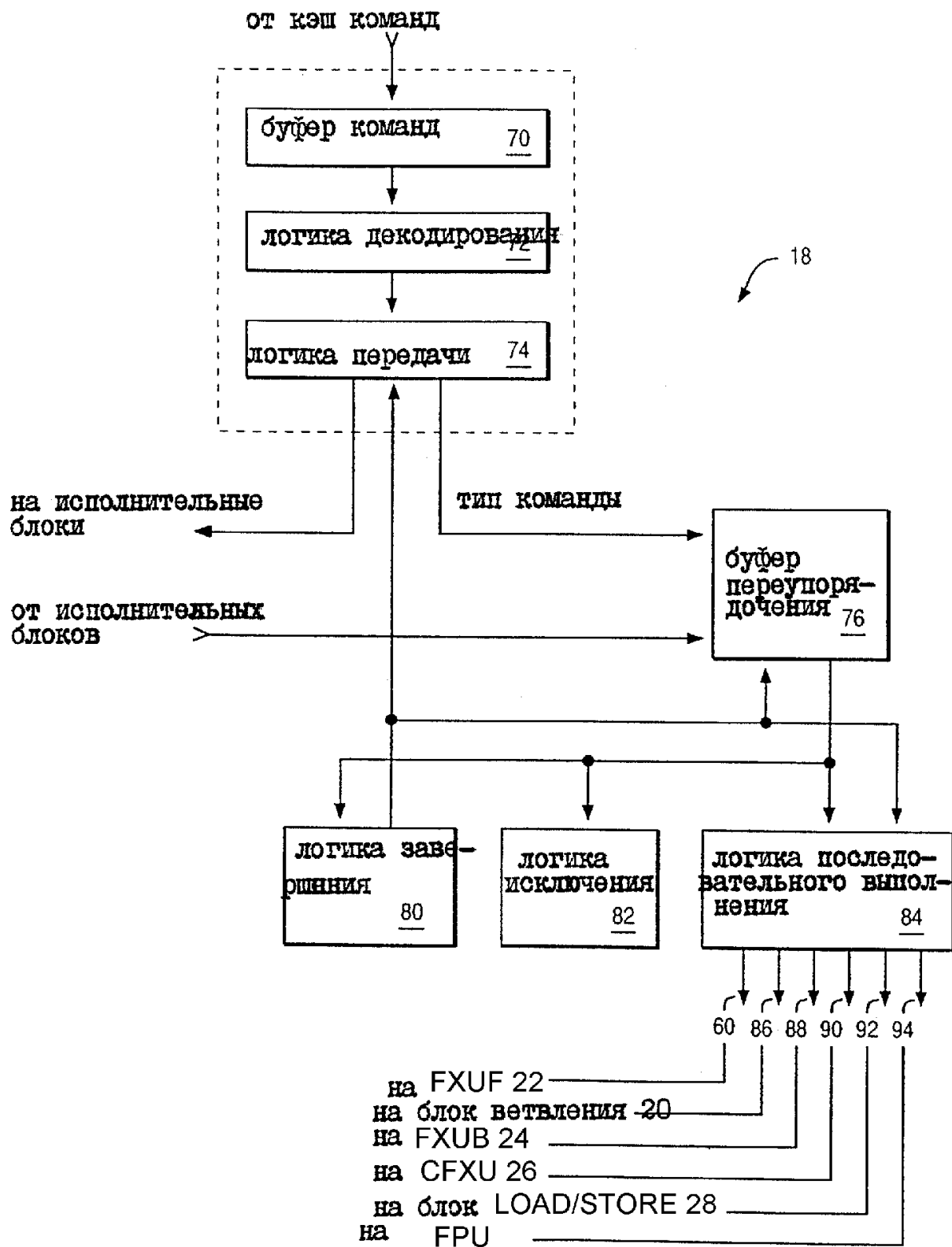
60

RU 2 1 4 2 1 5 7 C 1

RU ? 1 4 2 1 5 7 C 1



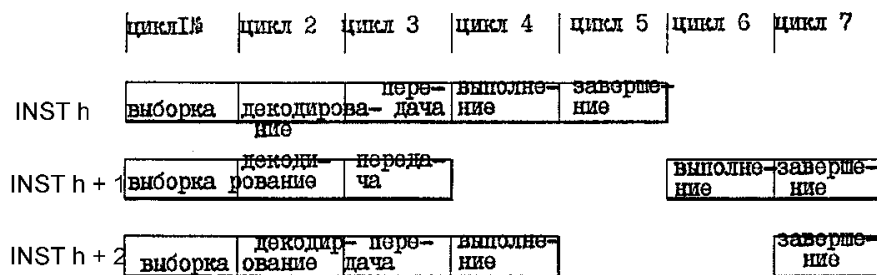
Фиг.2



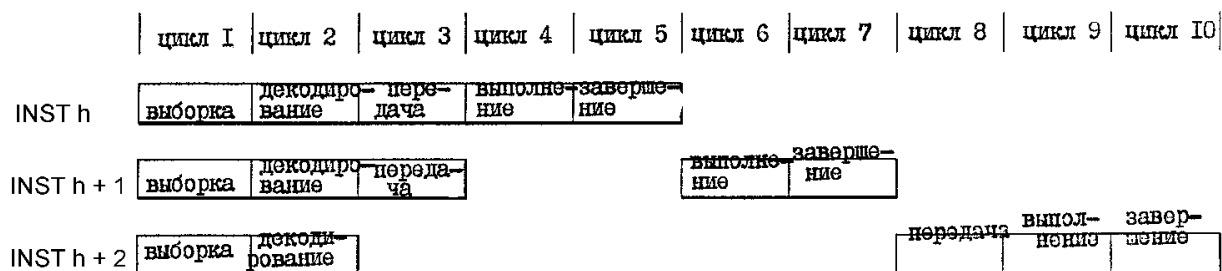
Фиг.3

номер буфера	тип команды		#GPR DEST	#FPR DEST	завер- шено	исключение
	исполнительный блок	EOK				
0	22	0	1	0	0	0
1	22	0	1	0	0	0
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						

Фиг.4



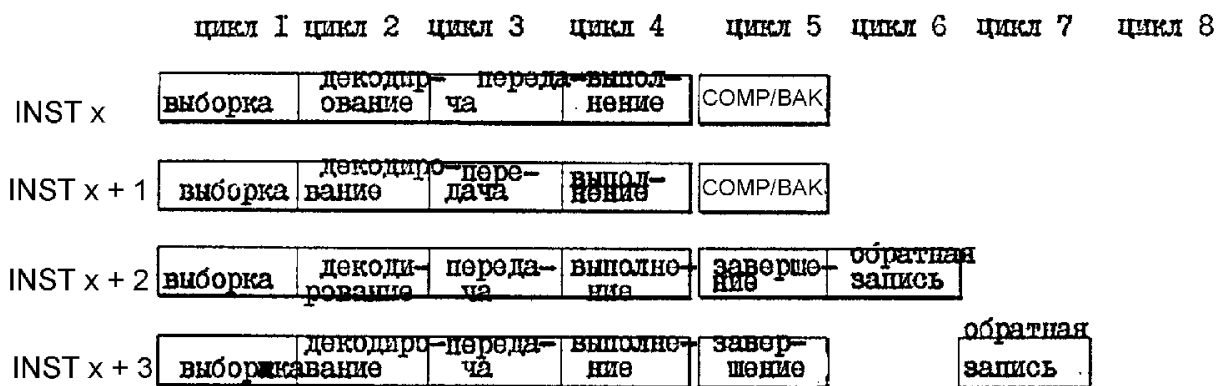
Фиг.5a



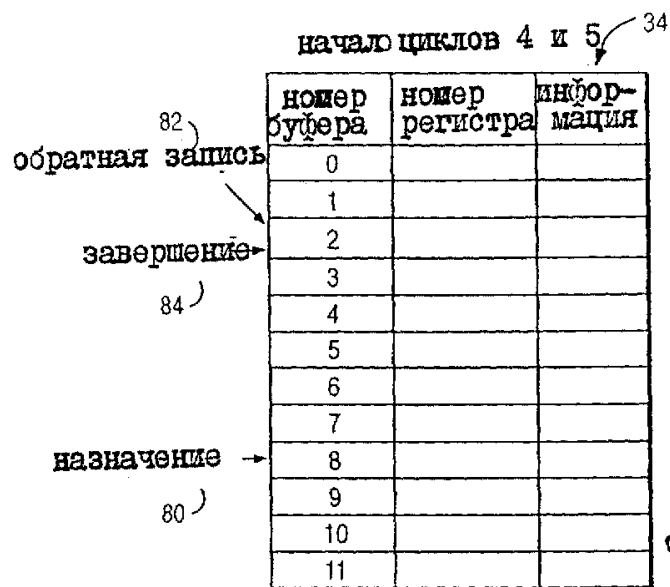
Фиг.5b

номер буфера	тип команды		#GPR DEST	#FPR DEST	завершено	исключе- ние
	исполнительный блок	EOK				
0						
1						
2						
3						
4						
5						
6						
7	24	1	1	0	1	0
8	22	1	1	0	1	0
9	28	1	2	0	1	0
10	28	1	2	0	1	0
11						
12						
13						
14						
15						

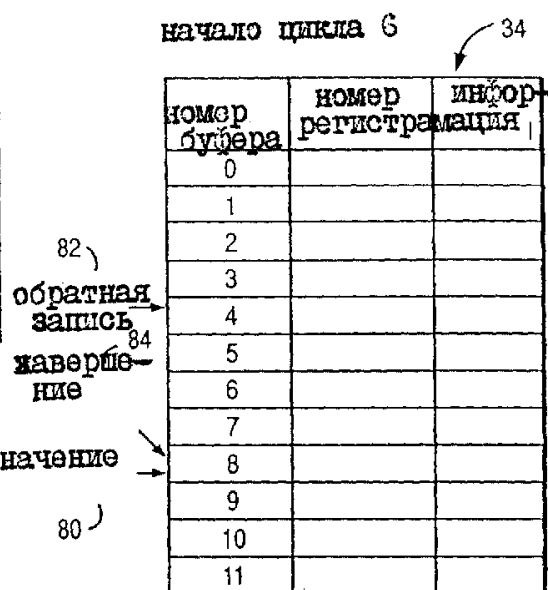
Фиг.6



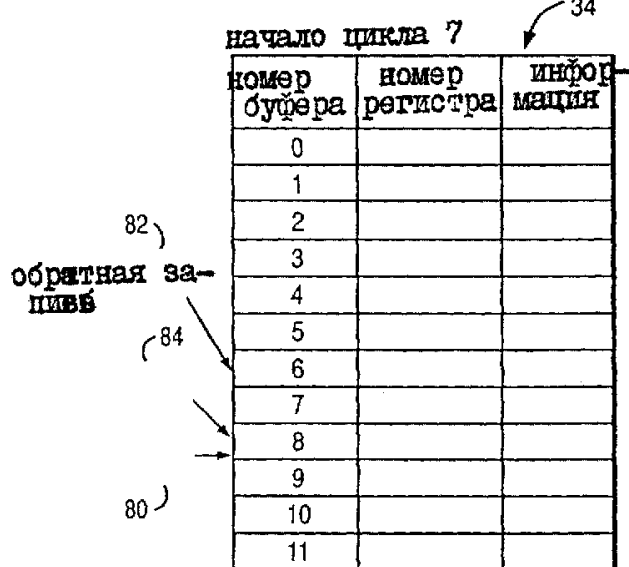
Фиг.7



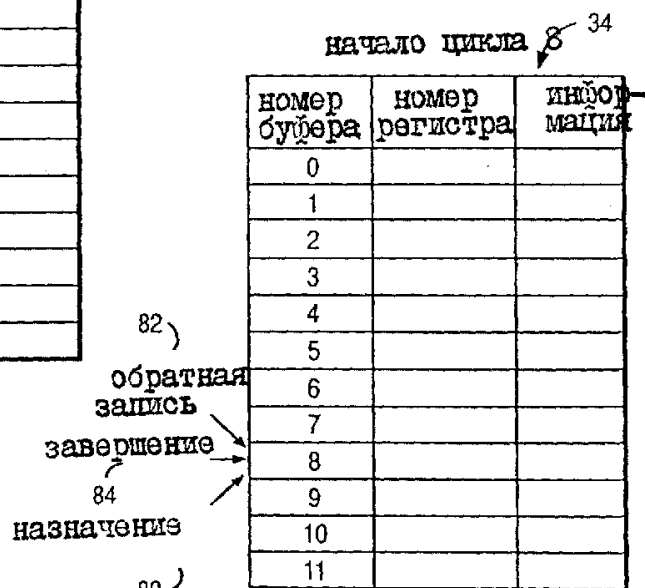
Фиг.8а



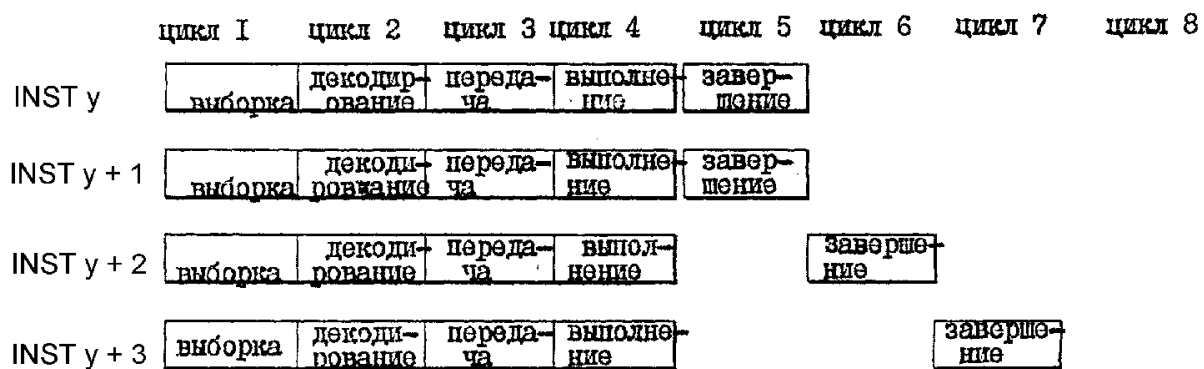
Фиг.8b



Фиг.8с



Фиг.8d



Фиг.9

RU 2142157 C1

RU 2142157 C1

начало циклов 4 и 5

номер буфера	номер регистра	инфо- рация
0		
1		
завершение 94) →	2	
	3	
	4	
	5	
	6	
	7	
назначение 90) →	8	
	9	
	10	
	11	

Фиг.10а

начало цикла 6

номер буфера	номер регистра	инфо- рация
0		
1		
2		
завершение 94) →	3	
	4	
	5	
	6	
	7	
назначение 90) →	8	
	9	
	10	
	11	

Фиг.10b

начало цикла 7

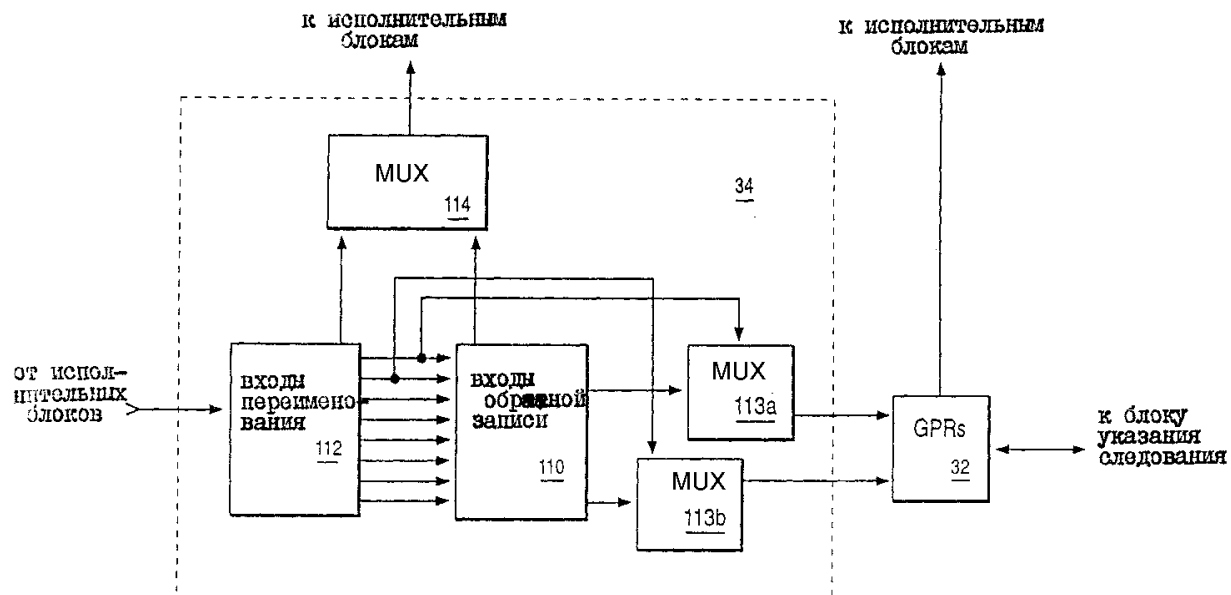
номер буфера	номер регистра	инфо- рация
0		
1		
2		
3		
завершение 94) →	4	
	5	
	6	
назначение 90) →	7	
	8	
	9	
	10	
	11	

Фиг.10с

начало цикла 8

номер буфера	номер регистра	инфо- рация
0		
1		
2		
3		
4		
5		
6		
заверше- 94) →	7	
назна- чение 90) →	8	
	9	
	10	
	11	

Фиг.10d



Фиг.11

76

номер буфера	тип команды		#GPR DEST	#FPR DEST	завер- шена	исключение
	исполнительный блок	EOK				
0						
1						
2						
3	22	1	2	0	0	0
4	24	1	1	0	1	0
5	28	0	0	0	0	0
6	26	1	1	0	1	0
7						
8						
9						
10						
11						
12						
13						
14						
15						

Фиг.12a

76

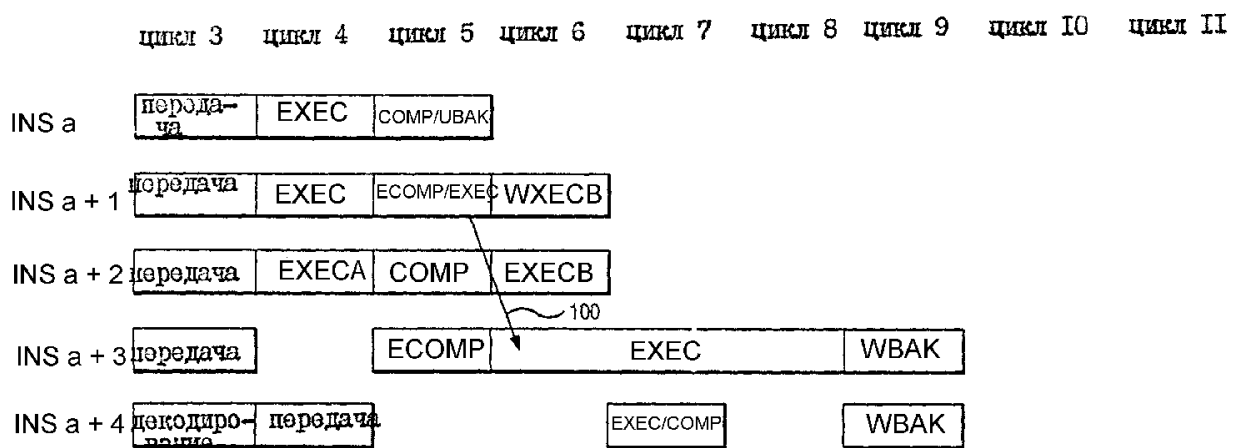
номер буфера	тип команды		#GPR DEST	#FPR DEST	завершена	исключе- ние
	исполнительный блок	EOK				
0						
1						
2						
3	22	1	2	0	1	0
4	24	1	1	0	1	0
5	28	0	0	0	1	0
6	26	1	1	0	1	0
7	22	0	1	0	0	0
8						
9						
10						
11						
12						
13						
14						
15						

Фиг.12b

76

номер буфера	тип команды		#GPR DEST	#FPR DEST	завер- шено	ис- ключение
	исполнительный блок	EOK				
0						
1						
2						
3						
4						
5						
6						
7	22	1	1	0	1	0
8						
9						
10						
11						
12						
13						
14						
15						

Фиг.12с



Фиг.13

начало цикла 4

34

84
CO, WB
82

AL
80

номер буфера	номер регистра	информация	правильная информация	выполнить	выполнить правильно
0					
1					
2			0		0
3			0		0
4			0		1
5			0		0
6					
7					
8					
9					
10					
11					

Фиг.14a

начало цикла 5

34

84
CO, WB
82

AL
80

номер буфера	номер регистра	информация	правильная информация	выполнить	выполнить правильно
0					правильно
1					
2		данные	1		0
3		данные	1		0
4			0		1
5			0		0
6			0		1
7					
8					
9					
10					
11					

Фиг.14b

начало цикла 6

34

номер буфера	номер регистра	инфор.	прав. инфор.	вып- олнить	выполнить правильно
0					
1					
2					
3					
82 WB →	4	данные	1	CA	1
	5		0		0
84 CO →	6		0		1
AL →	7				
80)	8				
	9				
	10				
	11				

Фиг.14с

начало цикла 7

34

номер буфера	номер регистра	инфор.	прав. инфор.	выпол- нить	выполнить правильно
0					
1					
2					
3					
82 WB →	4				
	5		0		0
84 CO →	6		0		1
AL →	7				
80)	8				
	9				
	10				
	11				

Фиг.14d

начало цикла 8

номер буфера	номер регистра	инфор. инфор.	прав. инфор.	выпол- нить	выполнить правильно
0					
1					
2					
3					
4					
WB → 5			0		0
84 CO, AL → 6		данные	1	CA	1
80 7					
8					
9					
10					
11					

Фиг.14е

начало цикла 9

номер буфера	номер регистра	инфор. инфор.	прав. инфор.	выпол- нить	выполнить правильно
0					
1					
2					
3					
4					
WB → 5		данные	1		0
80 AL, CO → 6		данные	1	CA	1
84 7					
8					
9					
10					
11					

Фиг.14f

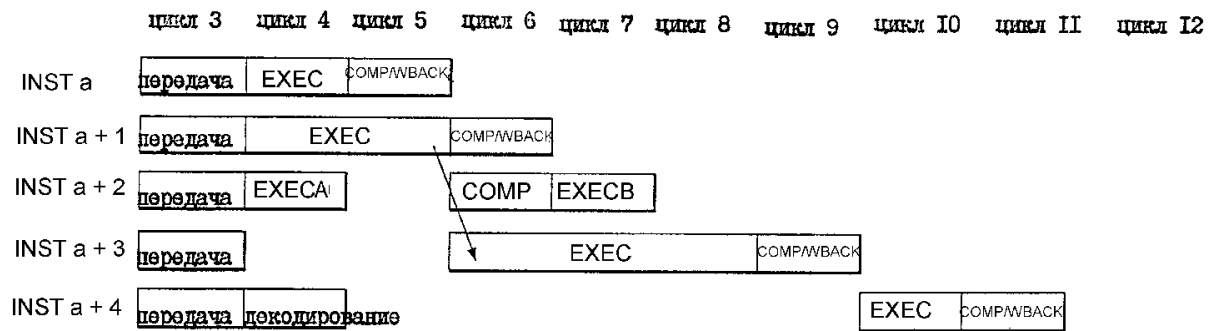
начало цикла IO

34

номер буфера	номер регистра	инфор. инфор.	прав. инфор.	выпол. нить	выполнить правильно
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					

80 84 82

Фиг.14g



Фиг.15

начало цикла 4

34

84
CO, WB
82

AL
80

номер буфера	номер регистра	инфор.	прав. инфор.	вып- олнить	можно выполнить правильно
0					
1					
2			0		0
3			0		0
4			0		1
5			0		0
6					
7					
8					
9					
10					
11					

Фиг.16а

начало цикла 5

34

84
CO, WB
82

AL
80

номер буфера	номер регистра	инфор.	прав. инфор.	выпол- нить	выполнить правильно
0					
1					
2		данные	1		0
3		данные	1		0
4			0		1
5			0		0
6			0		1
7					
8					
9					
10					
11					

Фиг.16b

начало цикла 6

34	номер буфера	номер регистра	инфр. инфр.	прав. инфр.	выпо- лнить	выполнить правильно
	0					
	1					
	2					
	3					
84	CO, WB		данные	1	CA	1
	5			0		0
	6			0		1
AIL	7					
80	8					
	9					
	10					
	11					

Фиг.16с

начало цикла 7 и 8

34	номер буфера	номер регистра	инфр. инфр.	прав. инфр.	вып- олнить	выполнить правильно
	0					
	1					
	2					
	3					
	4					
84	CO, WB			0		0
	6			0		1
AIL	7					
80	8					
	9					
	10					
	11					

Фиг.16d

начало цикла 9

номер буфера	номер регистра	инфор. инфор.	прав. инфор.	выпол- нить	выполнить правильно
0					
1					
2					
3					
4					
CO, WB →	5	данные	1		0
	6		0		1
AL →	7				
	8				
	9				
	10				
	11				

Фиг.16e

начало цикла 10.10

номер буфера	номер регистра	инфор. инфор.	прав. инфор.	выпол- нить	выполнить правильно
0					
1					
2					
3					
4					
5					
CO, WB →	6		0		1
AL →	7				
	8				
	9				
	10				
	11				

Фиг.16f

начало цикла II

34

84 CO, WB

82 AIL

80

номер буфера	номер регистра	инфр. инфр.	прав. нить	выпол- нить	выполнить правильно
0					
1					
2					
3					
4					
5					
6		данные	1	CA	1
7					
8					
9					
10					
11					

Фиг.16g

начало цикла I2

34

84 AIL

80 CO, WB

82

номер буфера	номер регистра	инфр. инфр.	прав. нить	выпол- нить	выполнить правильно
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					

Фиг.16h