



(19) **RU** ⁽¹¹⁾ **2 214 680** ⁽¹³⁾ **C2**
(51) МПК⁷ **H 04 B 1/69**

РОССИЙСКОЕ АГЕНТСТВО
ПО ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

(12) **ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ РОССИЙСКОЙ ФЕДЕРАЦИИ**

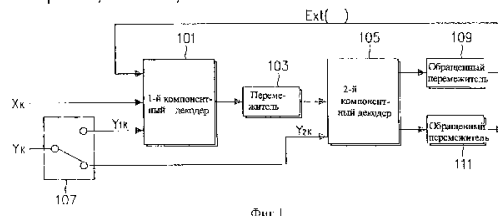
(21), (22) Заявка: 2001117857/09, 30.12.1999
(24) Дата начала действия патента: 30.12.1999
(30) Приоритет: 31.12.1998 KR 1998/62724
(43) Дата публикации заявки: 10.06.2003
(46) Дата публикации: 20.10.2003
(56) Ссылки: СИТНЯХОВСКИЙ И.В. и др. Цифровые системы передачи. - М.: Радио и связь, 1987, с.32-42, рис.2.10, 2.14, 2.15. KR 98/153966 В, 16.11.1998. US 5241383 А, 31.08.1993. EP 0300775 А, 16.01.1991. US 4379347 А, 05.04.1983.
(85) Дата перевода заявки РСТ на национальную фазу: 29.06.2001
(86) Заявка РСТ: KR 99/00842 (30.12.1999)
(87) Публикация РСТ: WO 00/41328 (13.07.2000)
(98) Адрес для переписки: 129010, Москва, ул. Б. Спасская, 25, стр.3, ООО "Юридическая фирма Городисский и Партнеры", пат.пов. Ю.Д.Кузнецову, рег.№ 595

(71) Заявитель:
САМСУНГ ЭЛЕКТРОНИКС КО., ЛТД. (KR)
(72) Изобретатель: КИМ Мин-Гоо (KR), КИМ Беонг-Дзо (KR), ЛИ Янг-Хван (KR), КИМ Се-Хиунг (KR)
(73) Патентообладатель:
САМСУНГ ЭЛЕКТРОНИКС КО., ЛТД. (KR)
(74) Патентный поверенный:
Кузнецов Юрий Дмитриевич

(54) СПОСОБ НОРМАЛИЗАЦИИ ЗНАЧЕНИЯ МЕТРИКИ КОМПОНЕНТНОГО ДЕКОДЕРА В СИСТЕМЕ МОБИЛЬНОЙ СВЯЗИ И УСТРОЙСТВО ДЛЯ ЕГО ОСУЩЕСТВЛЕНИЯ

(57) Изобретение относится к системам связи и может использоваться в средствах мобильной связи. Технический результат заключается в использовании изменения множества значений метрик на множестве временных отрезков. Декодер содержит схему принятия решения, которая формирует сигнал решения, когда все значения метрик превышают заранее установленное значение. Вычитатель вычитает заранее установленное значение из значений метрик в ответ на сигнал решения, чтобы нормализовать

значение метрик. Схема принятия решений содержит множество запоминающих устройств, которые запоминают соответствующее значение метрик. 3 с. и 3 з.п.ф-лы, 13 ил., 1 табл.





(19) **RU** ⁽¹¹⁾ **2 214 680** ⁽¹³⁾ **C2**
(51) Int. Cl.⁷ **H 04 B 1/69**

RUSSIAN AGENCY
FOR PATENTS AND TRADEMARKS

(12) **ABSTRACT OF INVENTION**

(21), (22) Application: 2001117857/09, 30.12.1999

(24) Effective date for property rights: 30.12.1999

(30) Priority: 31.12.1998 KR 1998/62724

(43) Application published: 10.06.2003

(46) Date of publication: 20.10.2003

(85) Commencement of national phase: 29.06.2001

(86) PCT application:
KR 99/00842 (30.12.1999)

(87) PCT publication:
WO 00/41328 (13.07.2000)

(98) Mail address:
129010, Moskva, ul. B. Spasskaja, 25, str.3,
ООО "Юридическая фирма Городисский и
Партнеры", пат.пов. Ю.Д.Кознетсову, рег.№ 595

(71) Applicant:
SAMSUNG EhLEKTRONIKS KO., LTD. (KR)

(72) Inventor: KIM Min-Goo (KR),
KIM Beong-Dzo (KR), LI Jang-Khvan (KR), KIM
Se-Khiung (KR)

(73) Proprietor:
SAMSUNG EhLEKTRONIKS KO., LTD. (KR)

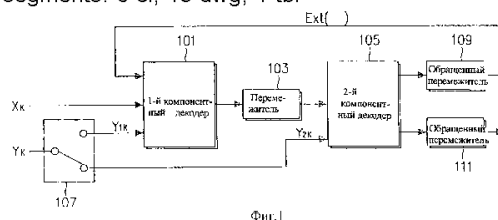
(74) Representative:
Kuznetsov Jurij Dmitrievich

(54) **METHOD OF NORMALIZATION OF VALUE OF METRIC OF COMPONENT DECODER IN MOBILE COMMUNICATION SYSTEM AND FACILITY FOR ITS IMPLEMENTATION**

(57) Abstract:

FIELD: communication systems, mobile communication systems. SUBSTANCE: decoder incorporates decision-making circuit that forms decision signal when all values of metrics exceed value determined beforehand. Subtractor subtracts predetermined value from values of metrics in response to decision signal in order to normalize value of metrics. Decision-making circuit incorporates assemblage of storages which record corresponding value of metrics.

EFFECT: use of change of collection of values of metrics across assemblage of time segments. 6 cl, 13 dwg, 1 tbl



Область техники

Настоящее изобретение относится к устройству и способу итеративного декодирования для системы мобильной связи и, в частности, касается устройства и способа для нормализации значения метрики, накопленного в компонентном декодере итеративного декодера в системе мобильной связи.

Уровень техники

В целом итеративное декодирование используется в таких системах мобильной связи, как IMT-2000 (или системах множественного доступа с кодовым разделением каналов МДКР-2000 и UMTS), в которых применяется турбокод. Итеративное декодирование используется также в системах дальней космической связи и системах спутниковой связи, где применяются каскадные сверточные коды, каскадные блочные коды или композиционные коды. Техническая область итеративного декодирования связана с так называемыми "мягкими" (не строго определенными) решениями и оптимальными характеристиками кода с исправлением ошибок.

На фиг.1 показан известный итеративный декодер, включающий два компонентных декодера. Согласно фиг.1 первый компонентный декодер 101 принимает сигналы X_k систематического кода, первый сигнал контроля по четности Y_{1k} , получаемый от демультиплексора 107 (который демультиплексирует входные сигналы контроля по четности Y_k) и первый внешний информационный сигнал. Первый компонентный декодер 101 выполняет декодирование полученных сигналов, выдавая первично декодированный сигнал, связанный с результатами декодирования. Этот сигнал состоит из составляющей X_k сигналов систематического кода и второй внешней информационной составляющей. Перемежитель 103 выполняет перемежение первично декодированного сигнала. Второй компонентный декодер 105 получает первично декодированный сигнал, поступающий с выхода переमेжителя 103, и второй сигнал контроля по четности Y_{2k} , получаемый от демультиплексора 107. Второй компонентный декодер 105 декодирует полученные первично декодированные сигналы и второй сигнал контроля по четности Y_{2k} , выдавая вторично декодированный сигнал на обращенный переमेжитель 111. Далее второй компонентный декодер 105 через обращенный перемежитель 109 подает внешнюю информационную составляющую на первый компонентный декодер 101.

Как показано на фиг.2, первый компонентный декодер включает блок 113 вычисления метрик ветвления (BMB) для вычисления метрик ветвления и блок 115 суммирования-сравнения-выбора (ССВ) для вычисления метрик и выполнения сравнения в каждом состоянии для выбора пути с меньшим количеством ошибок.

В общем случае указанный итеративный декодер вычисляет значение метрики M_t согласно приведенному ниже уравнению (1).

$$M_t = M_{t-1} + u_t \cdot X_{Lc} X_{y_{t,1}} + \sum_{j=2}^N x_{t,j} \cdot X_{Lc} X_{y_{t,j}} + u_t \cdot X_{Lc}(u_t)$$

где M_t - накопленное значение метрики за время t ;

U_t - кодовое слово для систематического бита, кодовое слово для каждого бита X_k ;

$X_{t,j}$ - кодовое слово для избыточного бита;

$Y_{t,j}$ - полученное значение для канала (систематическое + избыточное);

L_c - значение надежности канала, и

$L(u_t)$ - априорное значение надежности за время t .

Из уравнения (1) следует, что при каждом вычислении метрики значение метрики M_t непрерывно растет благодаря второму, третьему и четвертому членам. При аппаратной реализации значения метрик должны лежать в пределах заданного диапазона, чтобы избежать проблемы переполнения. Однако главным назначением итеративного декодера является выполнение итеративного декодирования с целью улучшения характеристик декодирования (то есть частоты ошибок по битам (ЧОБ) или частоты ошибок по кадрам (ЧОК)). Таким образом, в процессе выполнения итеративным декодером своей функции после ряда последовательных итераций значения метрик могут увеличиться и выйти за пределы заданного диапазона. Следовательно, если при разработке аппаратной части декодера предполагается задание определенного диапазона для значений метрик, значение метрик может превысить заданный диапазон, и возникнет проблема переполнения.

Сущность изобретения

Следовательно, задачей настоящего изобретения является создание устройства и способа для нормализации значения метрики компонентного декодера, причем при превышении всеми накопленными значениями метрик для текущих состояний некоторого порогового значения, эти накопленные значения метрик нормализуются до заданного уровня после вычитания из них заданного значения.

Для достижения вышеуказанного результата предлагается декодер, использующий изменение множества значений метрик на множестве временных отрезков. Декодер включает схему принятия решения, которая формирует сигнал решения, когда все значения метрик превысили заранее установленное значение. Вычитатель вычитает это заранее установленное значение из значений метрик в ответ на сигнал решения, чтобы нормализовать значения метрик. Схема принятия решения включает множество запоминающих устройств для запоминания соответствующих значений метрик с заранее установленным количеством битов. Логический элемент И-НЕ формирует сигнал решения ("1", или сигнал высокого уровня), когда все значения самых старших битов (ССБ), предусмотренных в соответствующих запоминающих устройствах, равны "1" (высокий уровень). Вычитатель устанавливает в ноль ССБ в каждом запоминающем устройстве, когда логический элемент И-НЕ выдает сигнал решения высокого уровня, в

результате чего из каждого значения метрики вычитается заранее установленная величина.

Краткое описание чертежей

Вышеуказанные и другие задачи, признаки и преимущества настоящего изобретения поясняются в последующем подробном описании, иллюстрируемом чертежами, на которых представлено следующее:

фиг. 1 - блок-схема, показывающая итеративный декодер, содержащий два компонентных декодера;

фиг.2 - подробная блок-схема, показывающая компонентные декодеры по фиг. 1;

фиг. 3 - схема, иллюстрирующая операции ССВ компонентного декодера, который имеет устройство нормализации значений метрик в блоке ССВ компонентного декодера, согласно первому варианту настоящего изобретения;

фиг. 4 - блок-схема, показывающая процедуру нормализации значений метрик согласно первому варианту настоящего изобретения;

фиг. 5 - схема, иллюстрирующая операции ССВ компонентного декодера, который имеет устройство нормализации значений метрик в блоке ССВ компонентного декодера, согласно второму варианту настоящего изобретения;

фиг. 6 - схема, где показан формат запоминающего устройства для значений метрик, для нормализации значений метрик согласно второму варианту настоящего изобретения;

фиг. 7 - блок-схема, иллюстрирующая процедуру нормализации значений метрик согласно второму варианту настоящего изобретения;

фиг. 8А и 8В - диаграммы, иллюстрирующие правильный путь, неправильный путь и разность путей, а также схему квантования для кодовых символов;

фиг. 9А-9С - диаграммы, иллюстрирующие правильный путь и неправильный путь в соответствии с отношением сигнал-шум; и

фиг. 10 - диаграмма, показывающая значение $\Delta_{\max}$ в состоянии насыщения в зависимости от отношения энергии (сигнала) к мощности шума E_b/N_0 .

Подробное описание предпочтительного варианта осуществления изобретения

ССВ для компонентного декодера согласно настоящему изобретению нормализует значения метрик после вычитания из них заданного значения, когда значения метрик превышают пороговое значение.

Есть два способа для нормализации накопленных значений метрик согласно настоящему изобретению. Согласно первому способу накопленные значения метрик нормализуются с использованием минимального накопленного значения метрики, когда одно из накопленных значений метрик соответствующих состояний превышает пороговое значение. Согласно второму способу накопленные значения метрик нормализуются с использованием заранее установленного значения, когда все накопленные значения метрик превышают пороговое значение.

Нормализация для ССВ по настоящему изобретению может быть использована взамен нормализации для блока ССВ 115 итеративного декодера 101, описанного выше в связи с фиг.2.

А. Первый вариант

Далее со ссылками на фиг.3 описывается первый вариант осуществления изобретения. На фиг.3 показана структура блока ССВ, имеющего устройство нормализации значений метрик для кодового ограничения $K=3$ согласно первому варианту настоящего изобретения.

Ниже со ссылками на фиг.3 описано устройство нормализации значений метрик. На фиг.3 показаны четыре "текущих состояния", каждое из которых имеет значение метрики. При $K=3$ количество сдвиговых регистров для значений метрик равно 2, а количество возможных состояний 4. Компаратор 117, имеющий пороговое значение, определяет значение метрики для каждого состояния. Когда все определенные значения метрик превышают пороговое значение, компаратор 117 выдает заданное значение на сумматоры 125a - 125d, причем каждый сумматор подсоединен между одним текущим состоянием и одним следующим состоянием. Затем в сумматорах 125a - 125d из накопленных значений метрик для текущих состояний вычитается заданное значение, а полученные в результате значения выдаются в следующие состояния. В данном описании термин "накопленные значения метрик текущего состояния" используется взамен термина "значения метрик текущего состояния" и наоборот, чтобы подчеркнуть тот факт, что значения метрик для текущего состояния при последовательных вычислениях метрик накапливаются.

На фиг. 4 показана процедура для нормализации значений метрик согласно первому варианту настоящего изобретения. Согласно фиг.4 компаратор 117 на этапе 401 определяет значения метрик для четырех текущих состояний. После определения значений метрик компаратор 117 на этапе 403 проверяет, превышает ли по меньшей мере одно из определенных накопленных значений метрик пороговое значение. Если ни одно из накопленных значений метрик не превышает пороговое значение, то компаратор 117 переходит на этап 407 для выполнения обычной операции ССВ. Если по меньшей мере одно из накопленных значений метрик превышает пороговое значение, то компаратор 117 на этапе 405 выдает на сумматоры 125a-125d минимальное из четырех обнаруженных накопленных значений метрик. Затем в сумматорах 125a-125d из всех четырех накопленных значений метрик вычитается минимальное накопленное значение метрики, а затем выполняется переход в следующие состояния. После этого, как показано на этапе 407, декодер переходит к выполнению обычной операции ССВ.

В. Второй вариант

Ниже описывается второй вариант осуществления изобретения.

На фиг.5 показана структура блока нормализации ССВ согласно второму варианту настоящего изобретения. Согласно фиг.5 компаратор включает в себя множество запоминающих устройств 130, 132, 134 и 136 для запоминания накопленных значений метрик соответствующих состояний, логический элемент И 121 для определения того, все ли накопленные значения метрик, запомненные в запоминающих устройствах

130, 132, 134 и 136, превышают пороговое значение, и инвертор 119 для установки в ноль самого старшего бита (ССБ) соответствующих запоминающих устройств 130, 132, 134 и 136 в ответ на сигнал высокого уровня, выдаваемый логическим элементом И 121.

Формат запоминающих устройств описан со ссылками на фиг.6. Здесь предполагается, что каждое накопленное значение метрики $u_0^k, u_1^k, u_2^k, u_3^k$ имеет 8 битов на выборку плюс один дополнительный бит для предотвращения переполнения накопленного значения метрики. Таким образом, накопленное значение метрики имеет всего 9 битов на выборку. Как показано на фиг.5, логический элемент И 121 получает девятый бит, который является самым старшим битом (ССБ) запоминающих устройств 130, 132, 134 и 136, и формирует выходной сигнал высокого уровня, когда все входные сигналы равны "1". То есть, когда ни один из ССБ запоминающих устройств 130, 132, 134 и 136 не равен "1", логический элемент И 121 не формирует выходной сигнал (сигнал низкого уровня). Когда все ССБ из запоминающих устройств имеют высокий уровень или "1", логический элемент И 121 формирует сигнал высокого уровня. Когда логический элемент И 121 выдает сигнал высокого уровня, инвертор 119 выдает сигнал установки в ноль бит ССБ запоминающих устройств 130, 132, 134, 136, устанавливая тем самым биты ССБ в ноль. Это эквивалентно вычитанию значения 256 из каждого накопленного значения метрики, что позволяет выразить накопленные значения метрик с помощью 8 битов.

Далее предположим, что разность между накопленными значениями метрик двух состояний составляет $\Delta^k = (u_i^k - u_j^k) \leq \Delta_{\max}$, где i и j представляют собой одно из значений 0, 1, 2 и 3, а k - произвольный момент времени. Также предположим, что максимальная разность между накопленными значениями метрик для двух состояний составляет $\Delta_{\max} = 255 = 2^8 - 1$. И наконец, предположим, что u_1^k - минимальное значение метрики, а u_3^k - максимальное значение метрики, как показано на фиг.6.

Что касается переполнения, то, если все биты ССБ значений метрик в момент k равны "1", минимальное значение равно 256.

Согласно вышеуказанным допущениям, если ССБ для u_3^k составляет "1", то ССБ других состояний будут равны "0", либо "1". Пока все биты ССБ для u_i^k (где $0 \leq i \leq 3$) не станут равными "1", выходной сигнал переноса не появится, даже если ССБ u_3^k (а возможно одно или два других значения метрик) равен "1". То есть, пока все биты ССБ не станут равными "1", выходной сигнал переноса на девятом бите не возникает для любого из них. Это означает, что Δ^k не превышает $\Delta_{\max}$.

На фиг.7 показана блок-схема, иллюстрирующая процедуру нормализации значений метрик согласно второму варианту. Согласно фиг.5 и 7 на этапе 501 компаратор 117 определяет накопленные значения метрик в конкретных единицах, логический элемент И 121 в компараторе 117а

определяет (или получает) биты ССБ накопленных значений метрик для соответствующих текущих состояний. Получив ССБ текущих состояний, компаратор 117 определяет, все ли накопленные значения метрик соответствующего текущего состояния превышают пороговое значение. То есть логический элемент 121 компаратора 117 определяет, все ли биты ССБ равны "1", как представлено в блоке принятия решений 503. Если ни один из ССБ не равен "1", то компаратор 117 переходит к этапу 507 для выполнения обычной операции ССБ. Если все биты ССБ накопленных значений метрик равны "1", то компаратор 117а переходит к этапу 505, где из каждого значения метрики вычитается пороговое значение. То есть все биты ССБ устанавливаются в ноль. Это соответствует логическому элементу И 121, подающему сигнал высокого уровня на инвертор 119, который в ответ на этот сигнал выдает сигнал установки в ноль битов ССБ соответствующих накопленных значений метрик, устанавливая тем самым ССБ в ноль. После установки битов ССБ в ноль компаратор 117а, как показано на этапе 507, выполняет обычную операцию ССБ.

Далее со ссылками на фиг. с 8А по 10 следует описание величины $\Delta_{\max}$, определенной выше.

При $\Delta^k < \Delta_{\max}$ переполнение не возникает. $\Delta_{\max}$ имеет более низкое значение при низком значении отношения E_b/N_0 и имеет более высокое значение при высоком значении отношения E_b/N_0 . То есть разность между значениями метрик имеет более низкое значение при низком E_b/N_0 и более высокое значение при высоком E_b/N_0 . Причина этого заключается в том, что шум при низком E_b/N_0 возрастает, приводя к уменьшению упомянутой разности, а при высоком E_b/N_0 шум крайне мал, что увеличивает разность $\Delta_{\max}$ между значениями метрик. Следовательно, очень важно, каким устанавливается значение $\Delta_{\max}$ при высоком E_b/N_0 . В первом случае можно просто считать, что $\Delta_{\max}$ имеет бесконечное значение при бесконечном значении E_b/N_0 . Однако, например, в алгоритме Витерби с нестрогим определенными ("мягкими") выходными данными (алгоритм SOVA) разность метрик ограничивается константой, определенной как d_{free} .

Например, предположим, что имеется 4 бита на выборку, скорость кода $R = 1/3$, $K = 9$ и сверточный код передает кодовое слово со всеми нулями "000". В этом случае при высоком значении отношения E_b/N_0 большинство ошибок появляется во время сравнения/выбора между путем со всеми нулями и путем d_{free} , как показано на фиг.8А. Здесь значение метрики ветвлений и значение метрики пути вычисляется по приведенным ниже уравнениям (2) и (3) соответственно.

$$BM_{(k)} = \sum_{l=1}^3 c_{k,l} x_{y_{(l)}k_j} \quad \text{Ур. 2}$$

$$PM_{(k)} = \sum_{k=1}^N \sum_{l=1}^3 c_{k,l} x_{y_{(l)}k_j} = u_{ki} \quad \text{Ур. 3}$$

где $l=0, 1, 2$ и 3 , $C_{k,j}$ - кодовое слово, $u_{(i)k,j}$ - принятый сигнал, g - данный момент времени, l - индекс кодового слова, а i - соответствующий индекс.

Следовательно, $\Delta_i^k = u_i^{s,k} - u_i^{c,k} \leq \Delta_{\max}$, где "s" обозначает выбранный путь, а "c" обозначает конкурентный путь. Необходимо вычислять $\Delta_{\max}$ при высоком E_b/N_0 , когда Δ_i^k имеет максимальное значение. Это означает, что если Δ_i^k меньше $\Delta_{\max}$ при высоком E_b/N_0 , когда Δ_i^k имеет максимальное значение, то тогда разность между значениями метрик не превышает $\Delta_{\max}$.

В данном состоянии "i" существует разность метрик между двумя путями: пути со всеми нулями и пути d_{free} . На фиг.8В показано, что разность между этими двумя путями зависит от кодового символа d_{free} .

Другими словами, метрика выбранного пути представляет собой значение, полученное путем суммирования метрики пути для нулевого пути со значением метрики в первом состоянии в предыдущий момент времени, а метрика конкурентного пути представляет собой значение, получаемое путем суммирования метрики пути, соответствующей конкурентному пути, со значением метрики во втором состоянии в предыдущий момент времени. В данный момент, поскольку метрика пути между вторым состоянием и временем сравнения больше, чем метрика пути между первым состоянием и временем сравнения,

разность $\Delta_{\max}$ равна или больше Δ_i^k . Следовательно, когда удовлетворяются условия для $\Delta_{\max}$, также удовлетворяются условия и Δ_i^k . То есть тот факт, что разность не превышает $\Delta_{\max}$, означает, что разность значений метрик между двумя состояниями в вышеуказанный момент времени не превышает $\Delta_{\max}$.

Разность метрик задается следующим выражением:

$(d_{\text{free}} \text{ (сверточного кодера)})=18$ для $K=9$, $R=1/3$)

$$\begin{aligned} \Delta^k &= |u_0^{s,k} - u_0^{c,k}| \\ &= |M - (M + d_{\text{free}} \times 15)| \\ &= |d_{\text{free}} \times 15| = 18 \times 15 = 270 \end{aligned}$$

где M обозначает значение метрики в точке ветвления выбранного пути и конкурентного пути. Следовательно, когда удовлетворяется условие $\Delta_{\max} \leq 270$, значение разности между соответствующими состояниями не превышает $\Delta_{\max}$. Поскольку предполагалось, что на выборку приходится 4 бита, количество запоминающих устройств для запоминания значения метрики составит 8, а поскольку для предотвращения переполнения добавляется 1-битовое запоминающее устройство, то $\Delta_{\max} = 2^9 = 512$. Поскольку $270 < 512$, вышеуказанное условие удовлетворяется.

На фиг. 9А показано значение $\Delta_{\max}$ при высоком отношении сигнал-шум, при этом значение $\Delta_{\max}$ вычисляется по формуле

$$\Delta_{\max} = d_{\text{free}} \times \text{Max}(Q[\text{ctot}]) \dots (4)$$

где Q обозначает уровень квантования, а $\text{Max}(Q[.])$ обозначает расстояние между "0" и "1". Например, для 4 битов на выборку $Q=16$ и $\text{Max}(Q[.])=15$, а для 3 битов на выборку $Q=8$, а $\text{Max}(Q[.])=7$.

На фиг.9В показано значение $\Delta_{\max}$ при среднем отношении сигнал-шум, причем значение $\Delta_{\max}$ в этот момент вычисляется по формуле

$$\Delta_{\max} = (d_{\text{free}} + \delta) \times \text{Max}(Q[.]) \dots \text{Ур.5}$$

где величина δ , обусловленная шумами, имеет очень маленькое значение, причем оно меньше или равно $2 \times d_{\text{free}} \times \text{Max}(Q[.])$ в сверточном кодере (СК). Однако это не так в случае, когда δ суммируется, как это показано в уравнении (5).

На фиг. 9С показано значение $\Delta_{\max}$ при низком отношении сигнал-шум, при этом в этот момент значение $\Delta_{\max}$ вычисляется по формуле

$$\Delta_{\max} = (d_{\text{free}} - \delta) \times \text{Max}(Q[.]) \dots (6)$$

Следовательно, как показано на фиг.10, необходимо отметить, что значение $\Delta_{\max}$ постепенно возрастает с ростом E_b/N_0 , причем начиная с определенной точки возникает насыщение. Если $\Delta_{\max}$ удовлетворяет уравнению (5), уравнение (6) также удовлетворяется.

Далее описаны характеристики сверточного кодера в системе CDMA-2000.

Для $K=9$ и $R=1/2$, $d_{\text{free}}=12$, а следующее $d_{\text{free}} \Rightarrow 14, 16, 18, 20$.

Для $K=9$ и $R=1/3$, $d_{\text{free}}=18$, а следующее $d_{\text{free}} \Rightarrow 20, 22$.

Для $K=9$ и $R=1/4$, $d_{\text{free}}=24$, а следующее $d_{\text{free}} \Rightarrow 26, 18$.

В таблице показано значение $\Delta_{\max}$ в сверточном кодере (СК).

Следовательно, количество битов, добавляемых для предотвращения переполнения для 8 битов на выборку, которые выделяются для значений метрик, определяется следующим образом.

Для $R=1/2$ количество битов равно 1, поскольку $2^8=256$, а $180 < 256$; для $R=1/3$ количество битов равно 2, поскольку $2^9=512$, а $270 < 512$; для $R=1/4$ количество битов равно 2, поскольку $2^9=512$, а $360 < 512$. Другими словами, поскольку для скорости кода $R=1/2$ требуется 8 битов, для предотвращения переполнения необходимо добавить только 1 бит. Кроме того, поскольку для скорости кода $R=1/3$ необходимо 9 битов, то понадобится добавить 1 бит. Следовательно, можно предотвратить переполнение, добавив к количеству битов, необходимых для данной скорости кода, только 1 бит.

Как было описано выше, новое устройство может предотвратить появление ошибок из-за переполнения путем нормализации накопленных значений метрик для декодирования, в результате чего повышается эффективность использования памяти.

Хотя изобретение было продемонстрировано и описано со ссылками

на конкретный предпочтительный вариант его осуществления, специалистам в данной области техники очевидно, что в него могут быть внесены различные изменения по форме и в деталях, не выходящие за рамки существа и объема изобретения, которые определены в формуле изобретения.

Формула изобретения:

1. Декодер, использующий изменение множества значений метрик на множестве временных отрезков, содержащий схему принятия решения, которая формирует сигнал решения, когда все значения метрик превышают заранее установленное значение, и вычитатель, который вычитает заранее установленное значение из значений метрик в ответ на сигнал решения.

2. Декодер по п.1, отличающийся тем, что схема принятия решения содержит множество запоминающих устройств, имеющих заранее установленное количество битов, причем каждое из них запоминает одно из множества значений метрик, и логический элемент И-НЕ, который в качестве входных сигналов получает значения самых старших битов (ССБ), подаваемых из соответствующих запоминающих устройств, причем на выход логического элемента И-НЕ выдается сигнал решения высокого уровня, когда все значения ССБ имеют высокий уровень.

3. Декодер по п.2, отличающийся тем, что вычитатель содержит инвертор, имеющий входной сигнал, который сопрягается с выходным сигналом логического элемента

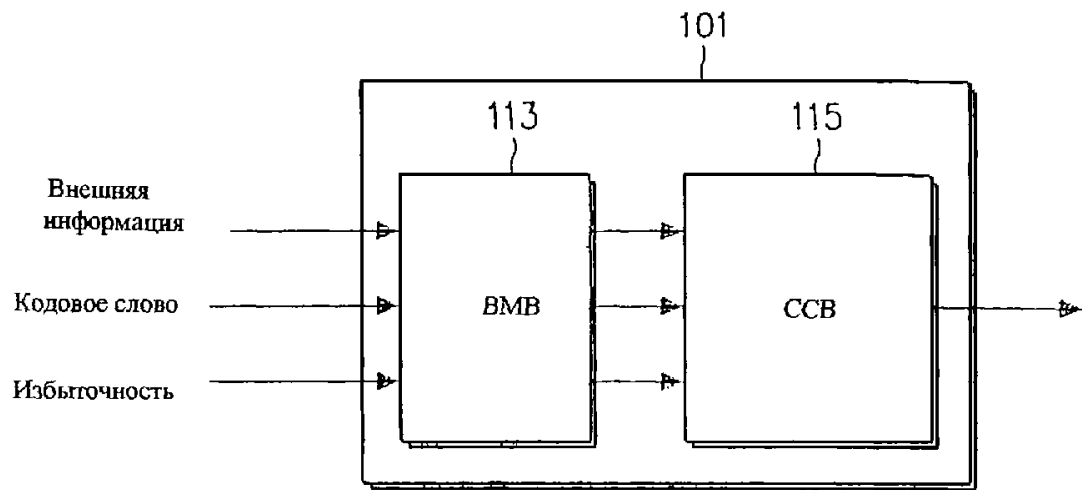
И-НЕ, и выходной сигнал, который сопрягается с ССБ запоминающих устройств, причем сигнал решения высокого уровня, выдаваемый логическим элементом И-НЕ на инвертор, формирует выходной сигнал инвертора, который устанавливает в ноль значения ССБ соответствующих запоминающих устройств.

4. Способ нормализации значений метрик в декодере, использующем изменение множества значений метрик на множестве временных отрезков, включающий этапы определения того, все ли значения метрик превосходят заранее установленный уровень, и вычитания заранее установленного значения из значений метрик для перехода к следующему состоянию, когда все значения метрик превысили это заранее установленное значение.

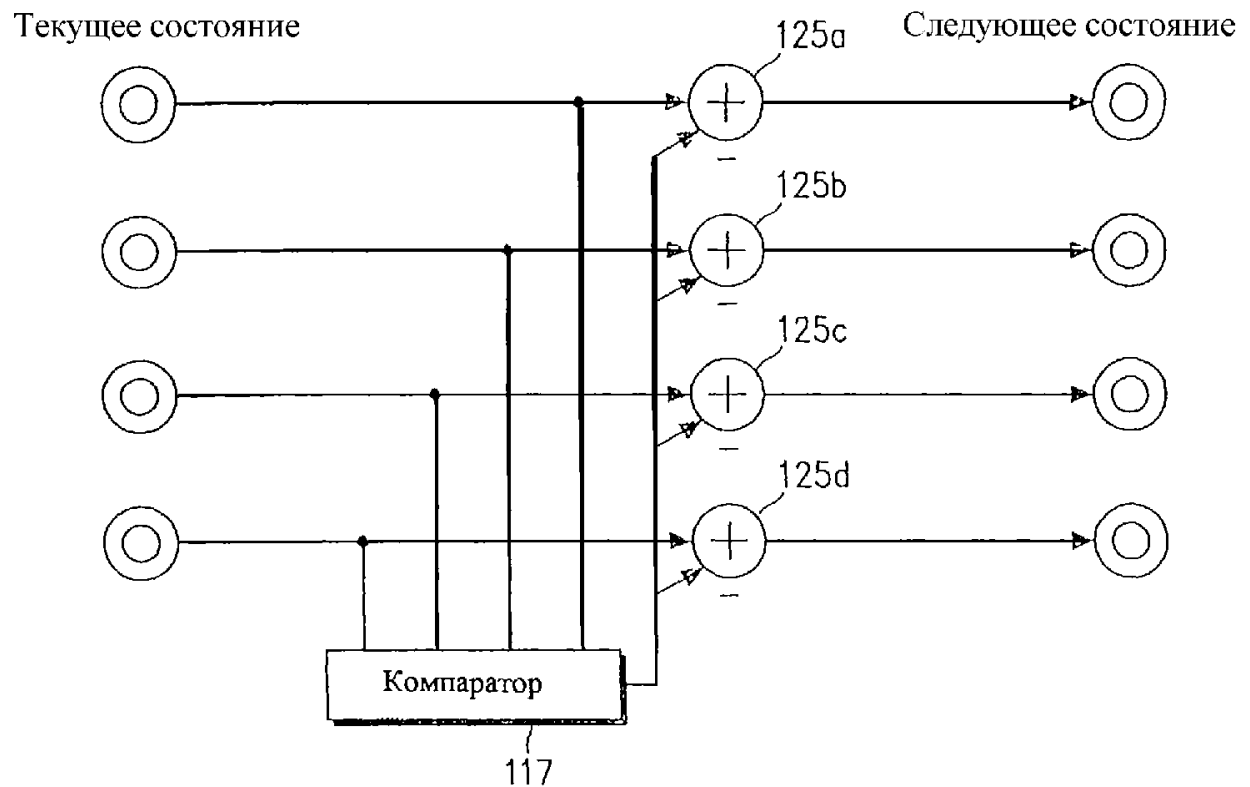
5. Способ по п.4, отличающийся тем, что вычитание заранее установленного значения из значений метрик включает установку в ноль ССБ множества запоминающих устройств, содержащих значения метрик.

6. Способ нормализации значений метрик в декодере, использующем изменение множества значений метрик на множестве временных отрезков и имеющем множество запоминающих устройств для запоминания значений метрик с заранее установленным количеством битов, включающий этапы определения того, все ли значения ССБ соответствующих запоминающих устройств равны 1, и установку в ноль значений ССБ, когда все значения ССБ равны 1.

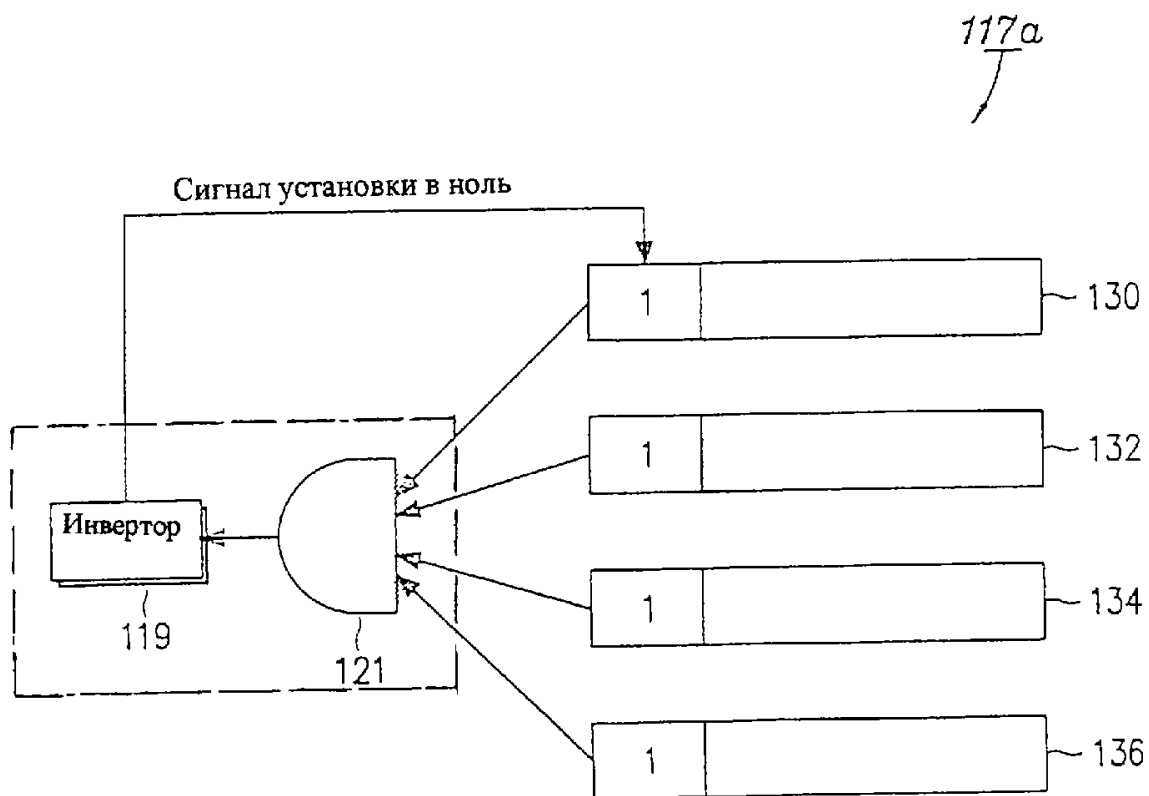
Характеристики СК	$\Delta_{\max}$ (предыдущее d_{free})	$\Delta_{\max}$ (следующее d_{free})
$K=9, R=1/2$	$15 \times 12 = 180$	$210 (15 \times 14) - 240 (15 \times 16)$
$R=1/3$	$15 \times 18 = 270$	$300 (15 \times 20) - 330 (15 \times 22)$
$R=1/4$	$15 \times 24 = 360$	$390 (15 \times 26) - 420 (15 \times 28)$

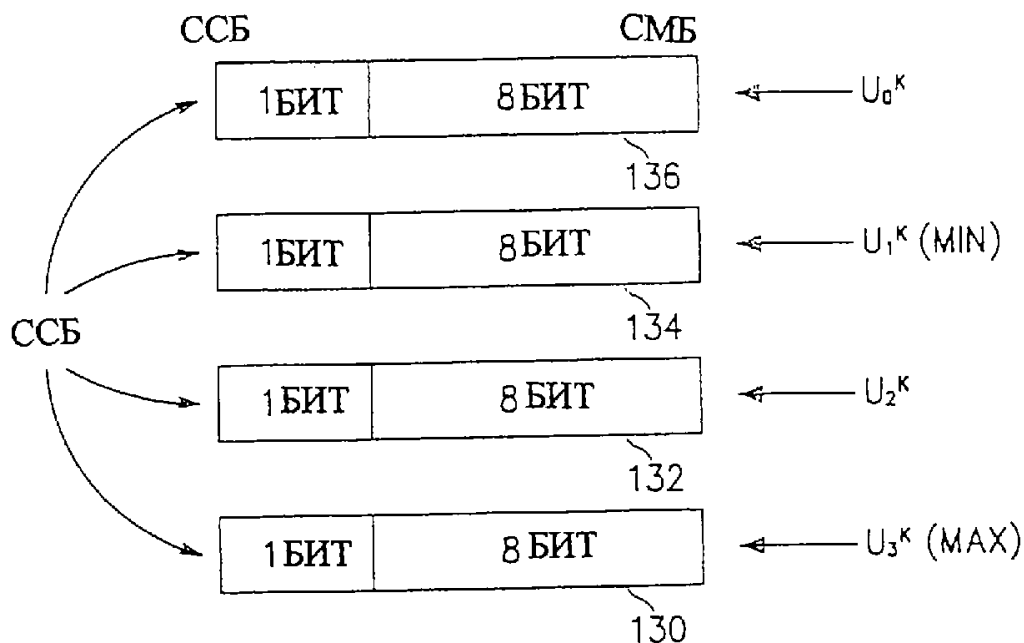


Фиг.2

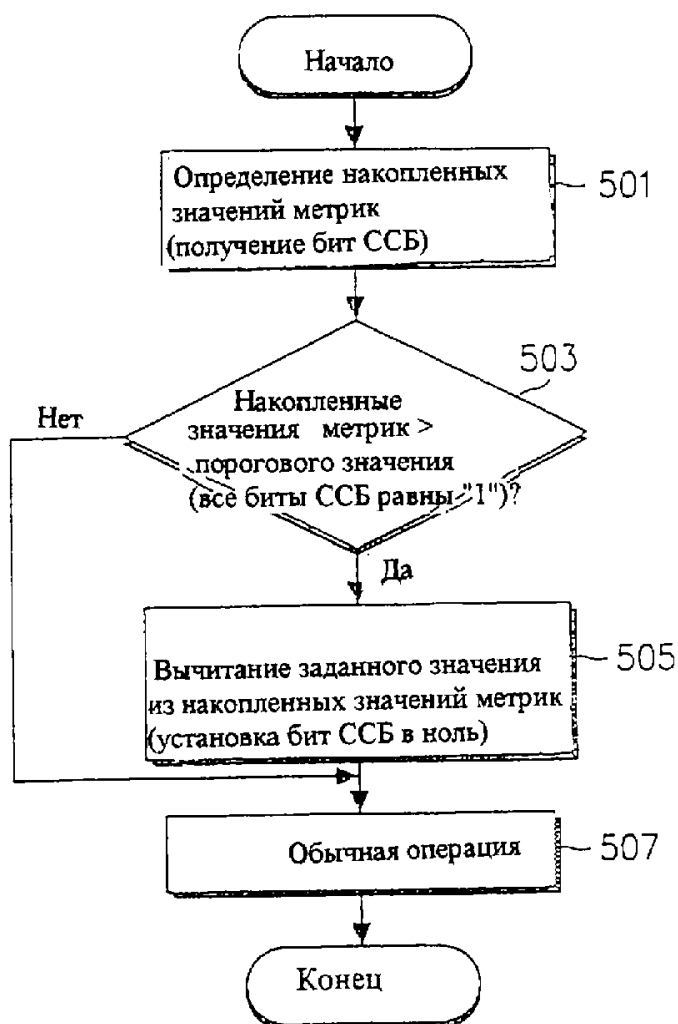


Фиг.3

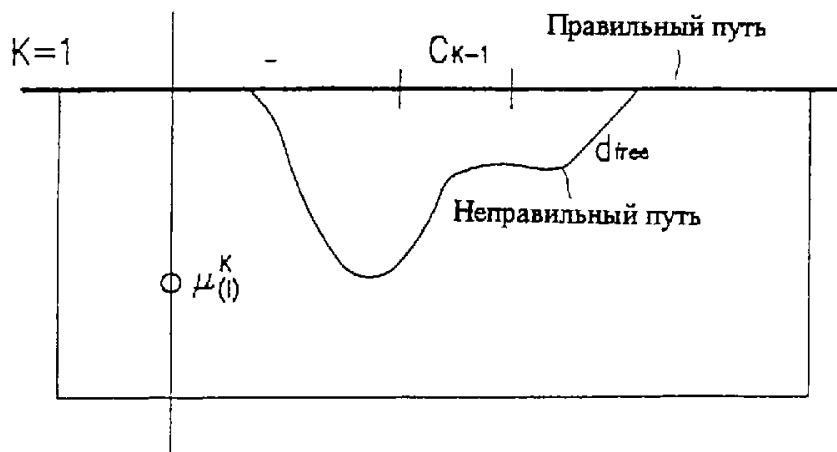




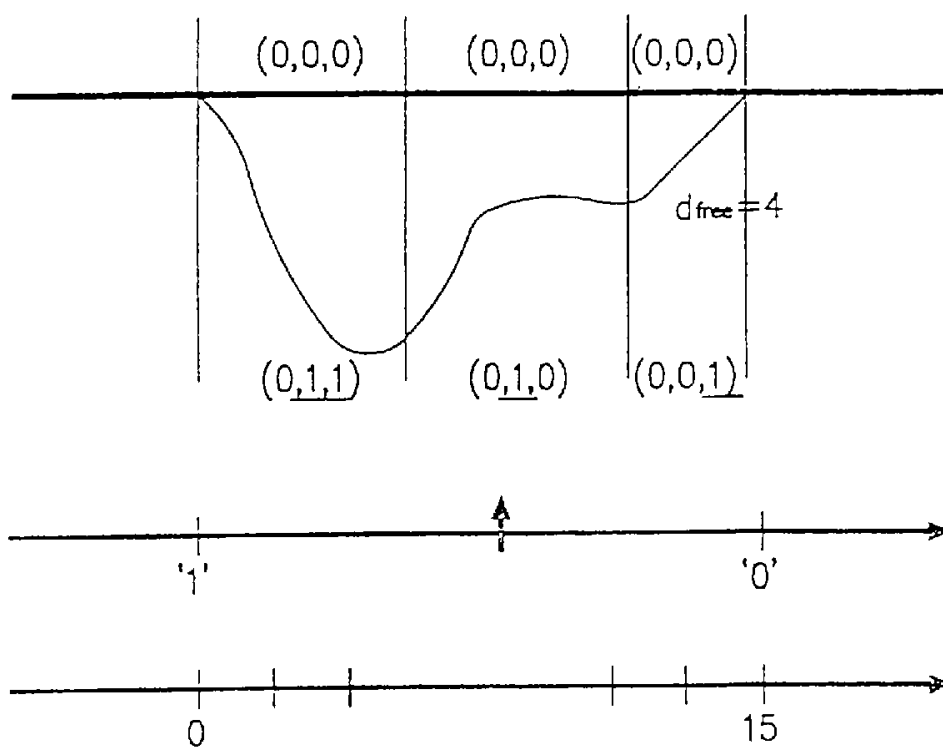
Фиг.6



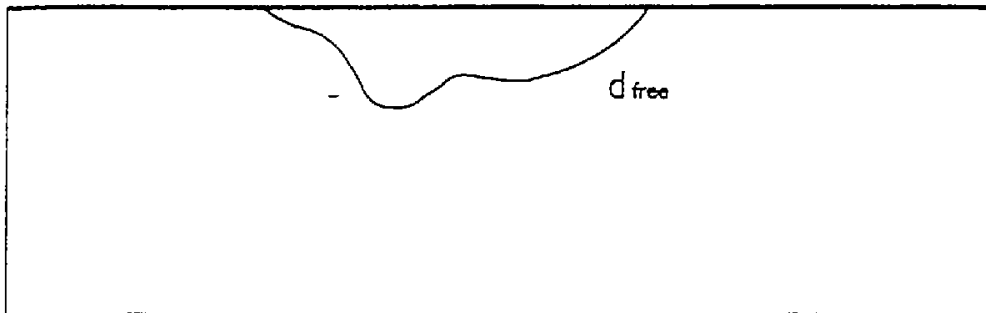
Фиг.7



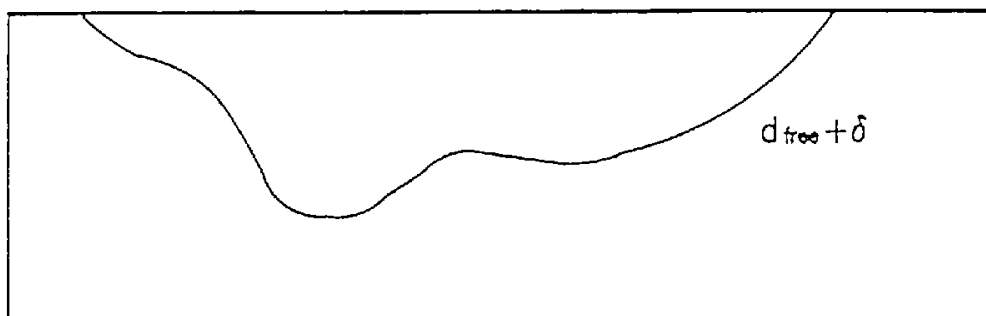
Фиг.8А



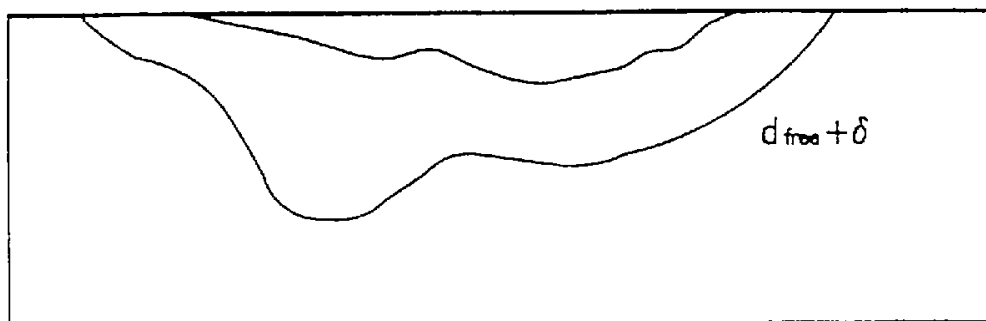
Фиг.8В



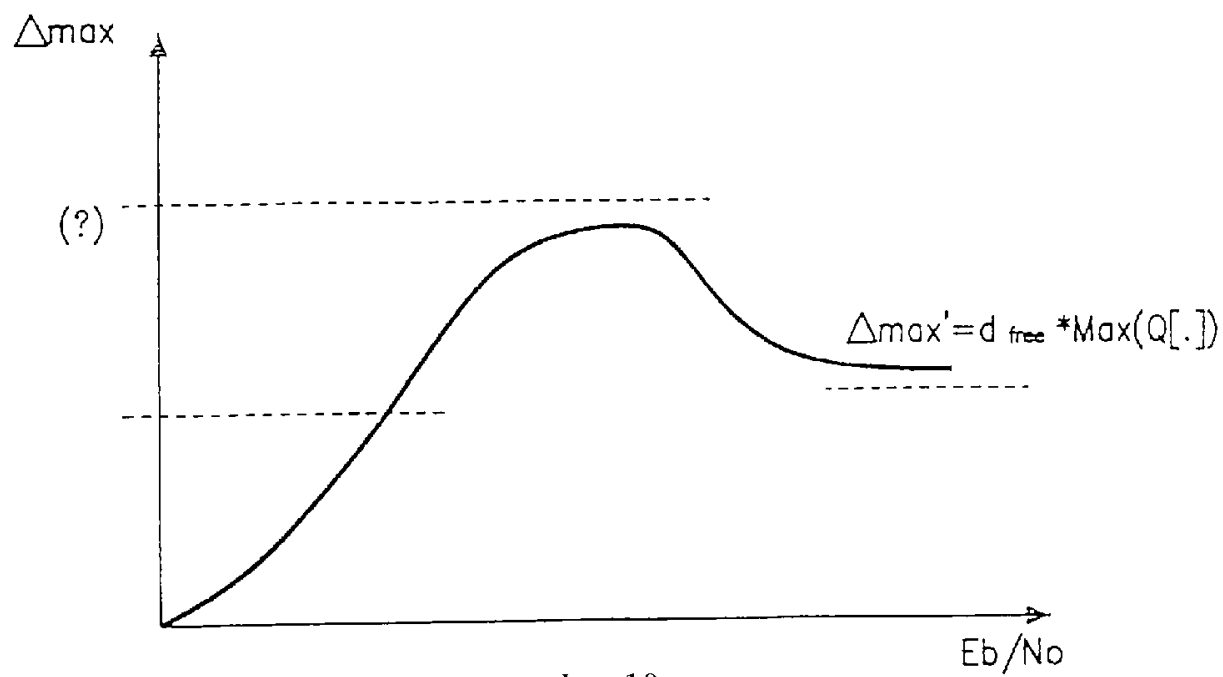
Фиг.9А



Фиг.9В



Фиг.9С



Фиг.10