

(12) FASCÍCULO DE PATENTE DE INVENÇÃO

(22) Data de pedido: **1999.02.22**

(30) Prioridade(s): **1998.03.23 EP 98200914**

(43) Data de publicação do pedido: **2000.03.08**

(45) Data e BPI da concessão: **2009.10.14**
242/2009

(73) Titular(es):

KONINKLIJKE PHILIPS ELECTRONICS N.V.
GROENEWOUDSEWEG 1 5621 BA EINDHOVEN

NL

PHILIPS AB

SE

(72) Inventor(es):

RENATUS JOSEPHUS VAN DER VLEUTEN

NL

(74) Mandatário:

MANUEL GOMES MONIZ PEREIRA
RUA ARCO DA CONCEIÇÃO, N.º 3, 1º ANDAR 1100-028
LISBOA

PT

(54) Epígrafe: **CODIFICAÇÃO E DESCODIFICAÇÃO ARITMÉTICA DUM SINAL DE INFORMAÇÃO**

(57) Resumo:

DESCRIÇÃO

CODIFICAÇÃO E DESCODIFICAÇÃO ARITMÉTICA DUM SINAL DE INFORMAÇÃO

A invenção diz respeito a um método de codificação aritmética dum sinal de informação, a um aparelho para codificar o sinal de informação e a um aparelho para decodificar o sinal de informação codificado aritmeticamente.

A codificação aritmética é uma técnica bem conhecida para codificação sem perda de dados e uma introdução pode ser encontrada em qualquer livro corrente de codificação. Para uma compreensão mais completa das implementações da codificação aritmética que são mais relevantes para o trabalho corrente, o leitor é remetido para [Lang84]. A história da codificação aritmética é bem descrita no apêndice deste documento. Mais, [Howard94] dá uma explicação extensiva da codificação aritmética.

A implementação da codificação aritmética que é o sujeito da presente invenção usa dois registos de dimensão finita, que são usualmente chamados C e A. O diagrama de fluxo da operação de codificação é mostrado na figura 1. O registo C aponta para a base dum intervalo na linha numérica, a dimensão do qual é armazenada em A, ver, por exemplo, [Lang84] e [Penn88]. O intervalo é dividido em sub-intervalos, cada sub-intervalo correspondendo a um símbolo a ser codificado e a dimensão de cada sub-intervalo correspondendo à probabilidade do símbolo associado. Para codificar na prática um símbolo, o registo C é ajustado para apontar para a base do sub-intervalo correspondente ao símbolo e o registo A é ajustado à dimensão do sub-intervalo seleccionado. O registo A (bem como C) é então normalizado

(deslocado à esquerda) antes do próximo símbolo ser codificado. Em geral após re-normalização, o valor de A situa-se entre os valores k e $2k$: $k \leq A \leq 2k$. No presente exemplo será usado $k=1/2$.

Por exemplo, no caso binário, existem dois sub-intervalos e então duas possíveis actualizações dos registos C e A , dependendo se o bit a ser codificado é o símbolo mais provável (MPS) ou o símbolo menos provável (LPS). É assumido que o MPS é designado para o intervalo inferior. O bloco "Actualizar A e C " da figura 1 é mostrado para o caso binário na figura 2. A probabilidade do bit de entrada ser o LPS é referenciada por p (notar que $p \leq 1/2$, porque a probabilidade de MPS é $p \geq 1/2$). O bit de entrada a ser codificado é referenciado por b . Os valores de b e p são providenciados pelo bloco "Ler.....". Agora, se um MPS é para ser codificado C não muda, dado que o intervalo inferior é seleccionado e C já aponta para este intervalo. Contudo, A deve mudar e a sua actualização é $A=A-A.p$ (usando o facto de que a probabilidade de MPS iguala $1-p$). Se um LPS é para ser codificado, ambos C e A são mudados: C é actualizado para $C=C+A-A.p$ e a nova dimensão do intervalo é $A=A.p$. Deve ser ainda notado que, por um processamento prévio e posterior, pode ser assegurado que o MPS é sempre, por exemplo o bit "0" e o LPS é sempre o bit "1". Finalmente, a figura 2 mostra um bloco "multiplicação aproximada" porque ele mostra que a multiplicação $A.p$ pode ser concretizada com menor precisão somente com uma pequena perda de desempenho, reduzindo deste modo a complexidade do hardware. Técnicas para fazer a multiplicação aproximada são discutidas mais tarde a seguir.

Para o caso não binário, o bloco "Actualizar A e C " da figura 1 é mostrado na figura 3. O bloco "Ler....." providencia agora o símbolo a ser codificado, s , bem como dois valores de probabilidade: a probabilidade p_s do símbolo s e a

probabilidade cumulativa p_t de todos os símbolos alinhados antes do símbolo s . Como pode ser observado na figura 3, o símbolo M é tratado diferentemente dos outros, com vista a “encher” exactamente A . É mostrado em [Riss89] que é vantajoso consignar o MPS ao símbolo M .

Com vista a ser capaz de descodificar, o codificador deve conhecer o valor de C , dado que ele determina o símbolo que foi codificado. Assim, o que é enviado ao descodificador é o valor do registo C . Na prática, cada vez que o registo A é deslocado à esquerda no processo de re-normalização, o MSB de C (também referido como “bit portador”) é processado para transmissão ao descodificador. O problema com o uso dum registo de dimensão finita para C é que o bit que já foi transferido para fora de C pode mais tarde ter de ser ajustado por uma transmissão ocasionada por aumento de C . Para tomar cuidado com isto, é necessário controlo vai-um. As técnicas actuais resolvem completamente o problema no codificador, pelo que o codificador não é afectado por isto. Estas soluções, que minimizam a complexidade do descodificador, também serão discutidas mais tarde.

O diagrama de fluxo do descodificador é mostrado na figura 4. Para o caso binário, o bloco “Sair símbolo.....” é mostrado na figura 5. No caso não binário, o descodificador é mais complexo, dado que ele tem de encontrar o inverso de “ $C=C+D$ ”, sem conhecer o valor de s .

A invenção pretende providenciar melhorias para os codificadores aritméticos descritos acima. De acordo com a invenção, o método de codificação compreende uma sequência série de símbolos de n -bits, n sendo um numero inteiro tal que $n \geq 1$, usando primeiro e segundo registos de dimensão finita para armazenar um parâmetro A e um parâmetro C , respectivamente, o parâmetro C tendo uma relação com um

limite dum intervalo de valor e o parâmetro A tendo uma relação com a dimensão do dito intervalo.

o método compreendendo as etapas de:

(a) recolher um símbolo do sinal de informação e pelo menos um correspondente valor de probabilidade do símbolo associado para codificação,

(b) recuperar os valores para os parâmetros A e C a partir dos primeiro e segundo registos, respectivamente,

(c) dividir o intervalo de valor correspondente ao valor recuperado a partir do primeiro registo em sub-intervalos correspondendo ao dito pelo menos um valor de probabilidade, e seleccionar um dos sub-intervalos em resposta ao dito símbolo,

(d) actualizar pelo menos o parâmetro A de modo a trazer o seu valor em concordância com a dimensão do sub-intervalo seleccionado, com vista a tornar-se a nova dimensão do intervalo para codificar o próximo símbolo no sinal de informação,

(e) Armazenar o valor actualizado para o parâmetro A no primeiro registo,

(f) continuar o método na etapa (a) para codificar o próximo símbolo caracterizado por a etapa (b) compreender além disso a sub-etapa de truncamento do valor do parâmetro A $0.b_0b_1.....b_{i-1}b_i.....$ no bit b_{i-1} e adicionar '1' na posição do bit b_{i-1} ao valor truncado de A, se b_i igualar '1'. Numa outra elaboração, o método de codificação compreende uma sequência série de símbolos de n-bits, n sendo um integral para o qual se mantém $n \geq 1$, usando primeiro e segundo registos de dimensão finita para armazenar um parâmetro A e um parâmetro C, respectivamente, o parâmetro C tendo uma relação com um limite dum intervalo de valores e o parâmetro A tendo uma relação com a dimensão do dito intervalo,

o método compreendendo as etapas de

- (a) recolher um símbolo do sinal de informação e pelo menos um correspondente valor de probabilidade do símbolo associado para codificação,
- (b) recuperar os valores para os parâmetros A e C a partir dos primeiro e segundo registos, respectivamente,
- (c) dividir o intervalo de valor correspondente ao valor recuperado a partir do primeiro registo em sub-intervalos correspondendo ao dito pelo menos um valor de probabilidade, e seleccionar um dos sub-intervalos em resposta ao dito símbolo,
- (d) actualizar pelo menos o parâmetro A de modo a trazer o seu valor em concordância com a dimensão do sub-intervalo seleccionado, com vista a tornar-se a nova dimensão do intervalo para codificar o próximo símbolo no sinal de informação,
- (e) Armazenar o valor actualizado para o parâmetro A no primeiro registo,
- (f) continuar o método na etapa (a) para codificar o próximo símbolo

caracterizado por a etapa (b) compreender além disso a sub-etapa de truncamento do valor do parâmetro A $0.b_0b_1\dots b_{i-1}b_i\dots$ no bit b_{i-1} e, se $b_{i-1} = '0'$ e $b_i = '1'$, elevar b_{i-1} para $'1'$. Ainda numa outra elaboração, o método de codificação compreende uma sequência série de símbolos de n-bits, n sendo um numero inteiro tal que $n \geq 1$, usando primeiro e segundo registos de dimensão finita para armazenar um parâmetro A e um parâmetro C, respectivamente, o parâmetro C tendo uma relação com um limite dum intervalo de valor e o parâmetro A tendo uma relação com a dimensão do dito intervalo, o método compreendendo as etapas de

- (a) recolher um símbolo do sinal de informação e pelo menos um correspondente valor de probabilidade do símbolo associado para codificação,
- (b) recuperar os valores para os parâmetros A e C a partir dos primeiro e segundo registos, respectivamente,
- (c) dividir o intervalo de valor correspondente ao valor recuperado a partir do primeiro registo em sub-intervalos correspondendo ao dito pelo menos um valor de probabilidade, e seleccionar um dos sub-intervalos em resposta ao dito símbolo,
- (d) actualizar pelo menos o parâmetro A de modo a trazer o seu valor em concordância com a dimensão do sub-intervalo seleccionado, com vista a tornar-se a nova dimensão do intervalo para codificar o próximo símbolo no sinal de informação,
- (e) Armazenar o valor actualizado para o parâmetro A no primeiro registo,
- (f) continuar o método na etapa (a) para codificar o próximo símbolo

caracterizado por a etapa (b) compreender além disso a sub-etapa de truncamento do valor do parâmetro A $0.b_0b_1\dots b_{i-1}b_i\dots$ no bit b_{i-1} e, tornar o bit b_{i-1} igual a '1'.

Os melhoramentos apresentados nesta invenção dizem respeito aos blocos de multiplicação aproximada (que são usados em ambos o codificador e o decodificador) e o controlo vai-um o qual toma lugar no bloco "Re-normalizar.....", apenas no codificador.

Estes e outros aspectos da invenção serão descritos em maior detalhe aqui a seguir, no qual

a figura 1 mostra um diagrama de fluxo do codificador aritmético,

a figura 2 mostra o diagrama de fluxo do bloco codificador "Atualizar A e C" da figura 1, para o caso binário. A probabilidade LPS é p e o valor do bit que é para ser codificado é designado por b ,

a figura 3 mostra o diagrama de fluxo do bloco codificador "Atualizar A e C" na figura 1, para o caso não binário. O valor do símbolo que é para ser codificado é designado por s e a sua probabilidade é designada p_s . Os $M+1$ símbolos são numerados $0, \dots, M$. $p_t = p_0 + p_1 + \dots + p_{s-1}$ é a probabilidade acumulativa de todos os símbolos alinhados antes do símbolo s ,

a figura 4 mostra o diagrama de fluxo do descodificador,

a figura 5 mostra o diagrama de fluxo do bloco descodificador "Símbolo de saída....." da figura 4, para o caso binário. A probabilidade LPS é p e o valor do bit que é descodificado é posto em b ,

a figura 6 mostra o diagrama de fluxo do bloco codificador denominado "Re-normalizar....." na figura 1,

a figura 7 mostra o diagrama de fluxo do bloco codificador denominado "Inicializar" na figura 1,

a figura 8 mostra o diagrama de fluxo do bloco codificador denominado "Terminar" na figura 4,

a figura 9 mostra o diagrama de fluxo do bloco descodificador denominado "Inicializar" na figura 4,

a figura 10 mostra o diagrama de fluxo do bloco descodificador denominado "Re-normalize....." na figura 4,

a figura 11 mostra uma realização do aparelho codificador, e

a figura 12 mostra uma realização do aparelho descodificador.

No que respeita aos melhoramentos para a multiplicação, pode ser dito o seguinte. O problema de "evitar" a multiplicação $A.p$ era resolvido em [Lang81] aproximando p por 2^{-Q} , onde Q é um número inteiro. A multiplicação por p corresponde então simplesmente a uma transferência para a direita de Q posições. Q é chamado o número desviador. Mais tarde, tal

como em [Riss89] o registo A era normalizado tal que $0,75 \leq A \leq 1,5$. Era então assumido que $A=1$, portanto $A.p \approx p$.

Ainda melhor desempenho era obtido em [Chev91a] e [Chev91b]. Eles reivindicam a aproximação de A eliminando todos os dígitos binários menos significativos que um pré-determinado binário de 1 dígito, e a sua realização preferida é usar o segundo binário de 1 dígito mais significativo. Assim na realização preferida, o valor de A é aproximado por um número binário contendo dois bits não zero, o que implica que a multiplicação pode ser realizada usando uma única operação de deslocamento e adição. Finalmente [Feyg93] descreve uma aproximação melhorada de A, que também pode ser implementada usando uma única operação de deslocamento e adição.

O(s) método(s) que é(são) usado na presente invenção para aproximar a multiplicação são como se segue. Deixar a probabilidade (p) ser descrita usando NP bits. Por exemplo, se $NP=8$, então $p=1/4=2^{-2}=0.01$ (binário) será representada pelo binário 01000000, isto é "0." não é incluído, dado que é o mesmo para todas as probabilidades. A dimensão do registo A é escolhida como $NA=NP+NX$ bits, em que NX representa o numero de bits que são usados para aproximar o valor de A que é usado para a multiplicação. Por exemplo, seja $NX=3$ e $A=3/4=0.11$, então A será um registo de $8+3=11$ bits contendo 11000000000 (notar que também o "0." é deixado cair, visto que se normaliza A de modo tal que é sempre menor do que um). Para a multiplicação, aproxima-se A por um número de 3-bits; neste caso, é claro que a melhor aproximação é 110. O resultado da multiplicação aproximada $A.p$ será então 00110000000, isto é outra vez um numero de 11-bits. Esta maneira de implementar a multiplicação aproximada foi sugerida, entre outros, em [Feig93].

Abaixo, será dada uma discussão de como o número A de N -bits deverá ser aproximado por N bits.

A primeira maneira de aproximar A (o método P1) compreende a medida para arredondar A para N bits em vez de o truncar. Arredondar significa que A é truncado para N bits se o bit de ordem $(N+1)$ de A é um 0 e que 1 é adicionado a esta representação truncada se o bit de ordem $(N+1)$ é um 1. Por exemplo, se $A=1101\dots$, a aproximação 3-bits será 111. O arredondamento que é aplicado aumenta a complexidade, uma vez que, em cerca de metade dos casos, tem de ser adicionado 1 à representação truncada, o que significa que deve ser feita uma operação de adição ou uma tabela de busca.

Como uma alternativa (método P2) é proposto adoptar um que é chamado "arredondamento parcial". Por arredondamento parcial um 1 é apenas adicionado à representação truncada de A no caso em que o bit de ordem $(N+1)$ é um "1" e o bit de ordem (N) é um "0". Na implementação isto significa que o bit de ordem N da aproximação de A iguala o OR lógico dos bits de ordem N e $(N+1)$ do A original. Por exemplo, $A=1011\dots$ será aproximado por 101 e $A=1001\dots$ será aproximado por 101 igualmente, enquanto $A=1000\dots$ será aproximado por 100. Notar que o arredondamento parcial resulta na mesma aproximação que o "arredondamento completo" em cerca de 75% dos casos.

Numa outra alternativa (método P3) é proposto aproximar A truncando-o para N bits e ajustar sempre o bit de ordem N da aproximação para 1, com a ideia de que isto reduz ainda mais a complexidade duma implementação de hardware, dado que isto elimina metade dos possíveis valores aproximados para A .

O desempenho de alguns métodos conhecidos foi comparado com os três novos métodos descritos acima. Os desempenhos dos vários métodos são listados no quadro I. Os métodos que são

mostrados no quadro, em adição aos três métodos de aproximação (P1,P2,P3) descritos acima, são um método de referência, denominado “referencia”, o qual computa a multiplicação completa $NA \times NP$ -bits e apenas então trunca A para NA bits, o método de [Moff95], o método de [Chev91a], o método descrito na secção 3 de Feyg93, denominado Feyg93(1), e o método descrito na secção 4 de Feyg93, denominado Feyg93(2).

Método	$NX=2$	$NX=3$	$NX=4$
referencia	100	100	100
Moff95	101,25	100,33	100,09
Chev91a	101,24	100,57	100,45
Feyg93(1)	101,22	100,32	100,19
Feyg93(2)	100,52	100,12	100,04
P1	100,33	100,12	100,03
P2	100,57	100,19	100,05
P3	101,66	100,44	100,10

Os números que são listados são dimensões relativas de arquivos comprimidos, com 100 correspondendo a “sem perda” devido a uma aproximação não perfeita da multiplicação. Por exemplo, um número de 100,57 significa que a dimensão do arquivo comprimido é aumentada de 0,57% devido à multiplicação aproximada.

Como esperado, o desempenho do método P2 é melhor que o método Moff95, mas não tão bom como o método P1.

O método P2 é um bom compromisso. Mais especificamente, P2 para $NX=3$ e $NX=4$ providencia um bom equilíbrio entre desempenho e complexidade, dado que o seu desempenho é praticamente o mesmo que o do método P1 (ver o quadro acima), com uma menor complexidade. Para $NX=2$, o método P1 é o método

preferido, enquanto o método P3 pode ser usado para $NX=5$ ou maior.

No caso não binário, métodos que podem aproximar o valor de A por arredondamento para cima têm o potencial problema que pode não haver “espaço” deixado para o MPS, quando a dimensão do alfabeto aumenta [Feyg93]. Para o método Feyg93(2) o pior caso limite na dimensão de alfabeto é 11 [Feyg93]. Os novos métodos de aproximação presentemente propostos têm a vantagem de que a quantidade pela qual A pode ser aumentado por arredondamento para cima diminui quando NX aumenta. Portanto, se existe uma aplicação na qual a distribuição de probabilidade é tal que a dimensão de alfabeto é limitada (e o desempenho de métodos que podem apenas truncar, ou arredondar para baixo, é insuficiente) um alfabeto maior pode ser manejado aumentando NX .

O problema de controlo vai-um na etapa de re-normalização no codificador era originalmente resolvido por uma técnica chamada “enchimento de bits” [Lang81]. Esta técnica “pega” a transmissão inserindo um bit 0 de enchimento no fluxo codificado no caso em que uma série de bits 1 tenha sido encontrada. As desvantagens desta técnica são que ela reduz a eficiência da compressão por causa dos bits de enchimento extras, e porque requer processamento especial no decodificador.

Um método para prevenir o vai-um sem afectar o desempenho de compressão foi descrito em [Witt87]. Este método tem a desvantagem da complexidade do decodificador ser algo aumentada. A ideia de [Witt87] foi adaptada em [Cham90], de modo que ele pode ser usado sem aumentar a complexidade do decodificador.

Aqui, é apresentada uma solução diferente que também não aumenta a complexidade do decodificador. O diagrama de fluxo

do procedimento de re-normalização do codificador de acordo com a invenção é mostrado na figura 6. O principal melhoramento é o bloco com $C < 0.10 \dots ?$. Para este mesmo bloco, a técnica anterior usa $C + A \leq 1.0 \dots ?$ e portanto requer uma operação de adição extra, comparada com a presente proposta.

Para completar a descrição do codificador, os blocos de inicialização e finalização são mostrados na figura 7 e figura 8, respectivamente. A variável de contagem é a mesma que a usada no bloco de re-normalização do codificador mostrado na figura 6. Dado que a dimensão de C é (NA+1) bits (no codificador), ele tem NA bits fracionais ("bits após o ponto), os quais são saída na finalização, como mostrado na figura 8.

A inicialização do codificador é mostrada na figura 9. O registo C é preenchido lendo (NA+1) bits a partir do fluxo. O primeiro bit lido é um "mudo", dado que ele é sempre um bit "0". A dimensão do registo C no decodificador é apenas NA bits, portanto um menos que no codificador. Não existe terminação especial no decodificador (o bloco "Finalização" na figura 4 está vazio). A re-normalização no decodificador (o bloco "Re-normalizar....." na figura 4) é mostrada na figura 10.

A figura 11 mostra uma realização do aparelho codificador de acordo com a invenção. O aparelho compreende terminais de entrada 100 e 102, para receber o sinal de informação e um sinal de probabilidade, respectivamente. O sinal de informação compreende uma sequência série de símbolos de n-bits, n sendo um numero inteiro tal que $n \geq 1$. O sinal de probabilidade aplicado ao terminal de entrada 102 compreende uma ou mais probabilidades para cada símbolo do sinal de informação. Para símbolos binários, o sinal de probabilidade compreende uma probabilidade para cada símbolo. Primeiro e

segundo registos de dimensão finita 104 e 106, respectivamente, estão presentes para armazenar o parâmetro A e o parâmetro C.

Uma unidade de processamento 108 está disponível para realizar a codificação aritmética do sinal de informação. Deve ser entendido que, sem ir em muito detalhe no que respeita a unidade de processamento 108, esta unidade compreende circuitos para recuperar os valores para os parâmetros A e C no primeiro e segundo registos, bem como circuitos para armazenar os valores actualizados e re-normalizados para A e C nos primeiro e segundo registos 104 e 106, respectivamente, depois de ter codificado o sinal. Mais, a unidade 108 compreende circuitos para dividir o intervalo de valor correspondente ao valor recuperado a partir do primeiro registo 104 em sub-intervalos correspondendo ao dito pelo menos um valor de probabilidade aplicado ao terminal 102, e circuitos para seleccionar um dos sub-intervalos em resposta ao dito símbolo aplicado ao terminal de entrada 100.

Circuitos para actualizar os parâmetros A e C estão também presentes, onde estes circuitos são requeridos de modo a levar o valor de A em conformidade com a dimensão do sub-intervalo seleccionado, e para levar o valor de C em acordo com um limite do dito sub-intervalo.

Um terminal de saída 110 está disponível dando saída a bits codificados em resposta aos símbolos codificados.

Os meios de recuperação para recuperar os parâmetros A e C a partir dos correspondentes registos compreendem ainda meios para truncar o valor do parâmetro A, antes de realizar o cálculo A.p. Mais especificamente, este truncamento pode ser como se segue: supor que o valor para A é expresso como $0.b_0b_1\dots b_{i-1}b_i\dots$. Este valor é truncado no bit b_{i-1} e um "1" é

adicionado na posição do bit b_{i-1} ao valor truncado de A se b_i é igual a "1".

Numa outra elaboração, o valor do parâmetro A é truncado no bit b_{i-1} e, se $b_{i-1}="0"$ e $b_i="1"$, o bit b_{i-1} é elevado para "1". Ainda numa outra elaboração, o parâmetro A é truncado no bit b_{i-1} e o bit b_{i-1} é tornado igual a "1".

Será apreciado que a unidade de processamento 108 é capaz de executar o método, como descrito nas figuras 1, 2, 3, 6, 7 e 8.

De preferência, o aparelho codificador é ainda providenciado com uma unidade de codificação de canal 112, bem conhecida na técnica, para codificação de canal (e, se necessário, codificação de correcção de erro) do sinal de informação codificado num sinal de informação codificado de canal, e uma unidade de escrita 114 para escrever o sinal codificado de canal sobre uma portadora de registo, tal como uma portadora de registo magnética 116, ou uma portadora de registo óptica 118.

A figura 12 mostra uma realização dum aparelho de descodificação de acordo com a invenção. O aparelho descodificador compreende um terminal de entrada 120 para receber o sinal de informação codificado. Primeiro e segundo registos de dimensão finita 122 e 124 estão presentes, para armazenar o parâmetro A e o parâmetro c, respectivamente.

Uma unidade de processamento 126 está disponível para realizar a descodificação aritmética do sinal de informação codificado que é recebido através da sua entrada 120, em resposta ao sinal de probabilidade fornecido à unidade de processamento 126 através duma entrada 134. O sinal de probabilidade pode ser obtido duma maneira bem conhecida. Um

exemplo de recolha das probabilidades para um sinal áudio de 1-bit é mostrado em [Bruek97]. Neste exemplo, as probabilidades são recolhidas a partir do sinal de saída descodificado que é fornecido à saída 128, nomeadamente executando uma filtragem de predição sobre o sinal de saída descodificado no filtro de predição 136 e gerando o sinal de probabilidade em resposta ao sinal de saída do filtro de predição 136 na unidade 138 que determina a probabilidade. Deve ser entendido que, sem ir em muito detalhe no que respeita à unidade de processamento 126, esta unidade compreende circuitos para recolher os valores para os parâmetros A e C a partir do primeiro e segundo registos, bem como circuitos para armazenar os valores actualizados e re-normalizados de A e C nos primeiro e segundo registos 122 e 124, respectivamente, depois de ter descodificado um símbolo. Mais, a unidade 126 compreende circuitos para executar as etapas mostradas nas figuras 4,5,9 e 10.

Os circuitos para recolher o valor do parâmetro A a partir do registo 122 compreende ainda meios para truncar o valor do parâmetro A antes de executar o calculo A.p. Este truncamento é da mesma maneira como descrito acima para o codificador, de modo que uma explicação adicional será dispensada.

De preferência, o aparelho descodificador é ainda provido com uma unidade de descodificação de canal 132, bem conhecida na técnica, para descodificação de canal (e, se necessário, correcção de erro) do sinal de informação codificado de canal para o sinal de informação codificado aritmeticamente para o descodificador aritmético 126, e uma unidade de leitura 130 para ler o sinal codificado de canal a partir duma portadora de registo, tal como uma portadora de registo magnética 116, ou uma portadora de registo óptica 118.

A codificação aritmética é aplicada em muitos esquemas de modem sem perda de dados e com perda para vídeo e áudio. Também pode ser aplicada na compressão de dados de computador (tal como, por exemplo, arquivos de texto). A aplicação encarada aqui, é na codificação sem perda de dados de sinais áudio de 1-bit. É feita referência a este respeito ao US ser. nº 08/966,375, correspondendo ao pedido de patente EP nº 97201680.2(PHN16405), US ser. nº 08/937,435, correspondendo ao pedido de patente internacional nº IB 97/01156(PHN16452).

Embora a invenção tenha sido descrita com referência às realizações preferidas dela, deve ser entendido que estas são exemplos não limitativos. Assim, várias modificações podem tornar-se aparentes aqueles peritos na técnica, sem sair do âmbito da invenção, como definida pelas reivindicações.

[Lang81] G.G. Langdon e al, "Compression of black-white images with arithmetic coding", IEEE Trans. on Com., Vol. COM-29, pp. 858-67, June 1981.

[Witt87] I.H. Witten et al, "Arithmetic coding for data compression", Communications ACM, Vol. 30, pp. 520-540, June 1987.

[Lang84] G.G. Langdon, "An introduction to arithmetic coding", IBM J. Res. Develop., Vol 28, pp. 135-149, March 1984.

[Penn88] W.B.Pennebaker et al, "An overview of the basic principles of the Q-coder adaptative binary arithmetic coder", IBM J. Res. Develop., Vol 32, pp. 717-26, Nov. 1988.

- [Riss89] J. Rissanen et al, "A multiplication-free multialphabet arithmetic code", IEEE Trans on Com, Vol. 37, pp. 93-8, Febr. 1989.

- [Cham90] USP 4,973,961

- [Chev91a] D. Chevion et al, "High efficiency multiplication free approximation of arithmetic coding" in Data Compression Conference (DCC '91), pp. 43-52, 1991.

- [Chev91b] USP 4,989,000

- [Feyg93] G. Feygin et al, "Minimizing error and VLSI complexity in the multiplication free approximation of arithmetic coding" in Data Compression Conference (DCC '93), pp. 118-127, Mar. 30-Apr. 1, 1993.

- [Howard94] P.G. Howard et al, "Arithmetic coding for data compression", Proc. IEEE, Vol. 82, n° 6, pp. 857-65, June 1994.

- [Moff95] Conference A. Moffat et al, "Aritmetic coding revisited" in Data Compression (DCC '95), pp. 202-11, 1995.

- [Bruek97] F. Bruekers et al, "Improved lossless coding of 1-bit audio signals", presented at 103rd Convention of the AES, Sept., 26-29, 1997, preprint 4563(1-6).

10-12-2009

REIVINDICAÇÕES

1. Método para codificar aritmeticamente um sinal de informação numérico compreendendo uma sequência série de símbolos de n -bits, n sendo um numero inteiro tal que $n \geq 1$, usando primeiro e segundo registos de dimensão finita para armazenar um parâmetro A e um parâmetro C, respectivamente, o parâmetro C tendo uma relação com um limite dum intervalo de valor e o parâmetro A tendo uma relação com a dimensão do dito intervalo, o método compreendendo as etapas de:

- (a) recolher um símbolo do sinal de informação e pelo menos um correspondente valor de probabilidade do símbolo associado para codificação,
- (b) recuperar os valores para os parâmetros A e C a partir dos primeiro e segundo registos, respectivamente,
- (c) dividir o intervalo de valor correspondente ao valor recuperado a partir do primeiro registo em sub-intervalos correspondendo ao dito pelo menos um valor de probabilidade, e seleccionar um dos sub-intervalos em resposta ao dito símbolo,
- (d) actualizar pelo menos o parâmetro A de modo a levar o seu valor em conformidade com a dimensão do sub-intervalo seleccionado, com vista a tornar-se a nova dimensão do intervalo para codificar o próximo símbolo no sinal de informação,
- (e) Armazenar o valor actualizado para o parâmetro A no primeiro registo,
- (f) continuar o método na etapa (a) para codificar o próximo símbolo

caracterizado por a etapa (b) compreender além disso a sub-etapa de truncamento do valor do parâmetro A $0.b_0b_1.....b_{i-1}b_i.....$ no bit b_{i-1} , e adicionar '1' na posição do bit b_{i-1} ao valor truncado de A, se b_i for igual a '1'.

2. Método para codificar aritmeticamente um sinal de informação digital compreendendo uma sequência série de símbolos de n -bits, n sendo um numero inteiro tal que $n \geq 1$, usando primeiro e segundo registos de dimensão finita para armazenar um parâmetro A e um parâmetro C, respectivamente, o parâmetro C tendo uma relação com um limite dum intervalo de valor e o parâmetro A tendo uma relação com a dimensão do dito intervalo, o método compreendendo as etapas de:

- (a) recolher um símbolo do sinal de informação e pelo menos um correspondente valor de probabilidade do símbolo associado para codificação,
- (b) recuperar os valores para os parâmetros A e C a partir dos primeiro e segundo registos, respectivamente,
- (c) dividir o intervalo de valor correspondente ao valor recuperado a partir do primeiro registo em sub-intervalos correspondendo ao dito pelo menos um valor de probabilidade, e seleccionar um dos sub-intervalos em resposta ao dito símbolo,
- (d) actualizar pelo menos o parâmetro A de modo a levar o seu valor em conformidade com a dimensão do sub-intervalo seleccionado, com vista a tornar-se a nova dimensão do intervalo para codificar o próximo símbolo no sinal de informação,
- (e) Armazenar o valor actualizado para o parâmetro A no primeiro registo,
- (f) continuar o método na etapa (a) para codificar o próximo símbolo

caracterizado por a etapa (b) compreender além disso a sub-etapa de truncamento do valor do parâmetro A $0.b_0b_1.....b_{i-1}b_i.....$ no bit b_{i-1} , e se $b_{i-1} = '0'$ e $b_i = '1'$ elevar b_{i-1} para $'1'$.

3. Método para codificar aritmeticamente um sinal de informação digital compreendendo uma sequência série de símbolos de n -bits, n sendo um numero inteiro tal que $n \geq 1$, usando primeiro e segundo registos de dimensão finita para armazenar um parâmetro A e um parâmetro C, respectivamente, o parâmetro C tendo uma relação com um limite dum intervalo de valor e o parâmetro A tendo uma relação com a dimensão do dito intervalo, o método compreendendo as etapas de:

- (a) recolher um símbolo do sinal de informação e pelo menos um correspondente valor de probabilidade do símbolo associado para codificação,
- (b) recuperar os valores para os parâmetros A e C a partir dos primeiro e segundo registos, respectivamente,
- (c) dividir o intervalo de valor correspondente ao valor recuperado a partir do primeiro registo em sub-intervalos correspondendo ao dito pelo menos um valor de probabilidade, e seleccionar um dos sub-intervalos em resposta ao dito símbolo,
- (d) actualizar pelo menos o parâmetro A de modo a levar o seu valor em conformidade com a dimensão do sub-intervalo seleccionado, com vista a tornar-se a nova dimensão do intervalo para codificar o próximo símbolo no sinal de informação,
- (e) Armazenar o valor actualizado para o parâmetro A no primeiro registo,
- (f) continuar o método na etapa (a) para codificar o próximo símbolo

caracterizado por a etapa (b) compreender além disso a sub-etapa de truncamento do valor do parâmetro A $0.b_0b_1.....b_{i-1}b_i.....$ no bit b_{i-1} , e, tornar o bit b_{i-1} igual a '1'.

4. Método segundo as reivindicações 1, 2 ou 3, a etapa de actualização compreendendo também a actualização do valor de

C de maneira a levar o valor do parâmetro C para uma relação correspondente a um limite do sub-intervalo seleccionado, de modo a tornar-se o novo parâmetro C para codificar o próximo símbolo do sinal de informação, a etapa de armazenamento compreendendo igualmente o armazenamento do valor actualizado do parâmetro C no segundo registo.

5. Método segundo as reivindicações 1, 2, 3 ou 4, a etapa de actualização compreendendo igualmente a sub-etapa de re-normalização para os valores dos parâmetros A e C, antes de armazenar os valores re-normalizados dos parâmetros A e C no primeiro e segundo registos, respectivamente, caracterizado por a sub-etapa de re-normalização compreender:

(g1) comparar o valor para A com um primeiro valor binário, se o valor para A não for menor que o dito valor binário, deixar a etapa de re-normalização, e se o valor para A for menor que o primeiro valor binário, então

(g2) multiplicar o valor para A com um primeiro valor de número inteiro,

(g3) voltar para (g1).

6. Método segundo a reivindicação 5, caracterizado por se A for inferior ao dito valor binário em (g1):

(g4) comparar o valor para C com um segundo e um terceiro valor binário, o segundo valor binário sendo superior ao dito terceiro valor binário, e, se o valor para C for menor que o dito segundo valor binário, e maior ou igual ao terceiro valor binário, então

(g4) subtrair um quarto valor binário ao valor para C de maneira a obter um valor intermédio para C

(g5) multiplicar o valor intermédio para C com um segundo valor de número inteiro.

7. Método segundo a reivindicação 5, caracterizado por o primeiro valor binário ser igual a 0.100...0.

8. Método segundo a reivindicação 5, caracterizado por o segundo valor binário ser igual a 1.000...0.

9. Método segundo a reivindicação 5, caracterizado por o primeiro valor de numero inteiro ser igual a 2.

10. Método segundo a reivindicação 6, caracterizado por o terceiro valor binário ser igual a 0.100...0.

11. Método segundo a reivindicação 6, caracterizado por o quarto valor binário ser igual a 0.1000...0.

12. Método segundo a reivindicação 6, caracterizado por o segundo valor de numero inteiro ser igual a 2.

13. Método segundo uma qualquer das reivindicações precedentes, caracterizado por compreender, além disso, a etapa de codificação de canal do sinal de informação codificado num sinal codificado de canal.

14. Método segundo a reivindicação 13, caracterizado por compreender, além disso, a etapa de registo do sinal de canal sobre uma portadora de registo.

15. Aparelho para codificar aritmeticamente um sinal de informação numérico compreendendo uma sequência série de símbolos de n-bits, n sendo um número inteiro tal que $n \geq 1$, o aparelho compreendendo:

- um primeiro e um segundo registos de dimensão finita (104,106) para armazenar um parâmetro A e um parâmetro C, respectivamente, o parâmetro C tendo uma relação com um

limite dum intervalo de valor e o parâmetro A tendo uma relação com a dimensão do dito intervalo,

- meios de entrada (100,102) para receber um símbolo do sinal de informação e pelo menos um valor de probabilidade correspondente para o símbolo associado para codificar,

- meios de recuperação para recuperar os valores para os parâmetros A e C nos primeiro e segundo registos, respectivamente,

- meios para dividir o intervalo de valor correspondente ao valor recuperado do primeiro registo em sub-intervalos correspondendo ao dito pelo menos um primeiro valor de probabilidade, e seleccionar um dos sub-intervalos em resposta ao dito símbolo,

- meios para actualizar pelo menos o parâmetro A de maneira a levar o seu valor em conformidade com a dimensão do sub-intervalo seleccionado com vista a tornar-se a nova dimensão do intervalo para codificar o próximo símbolo do sinal de informação,

- meios para armazenar o valor actualizado para o parâmetro A no primeiro registo,

caracterizado por os meios de recuperação compreenderem igualmente meios para truncar o valor do parâmetro A $0.b_0b_1\dots b_{i-1}b_i$ no bit b_{i-1} e adicionar '1' na posição do bit b_{i-1} no valor truncado de A, se b_i é igual a '1'.

16. Aparelho para codificar aritmeticamente um sinal de informação numérico compreendendo uma sequência série de símbolos de n-bits, n sendo um número inteiro tal que $n \geq 1$, o aparelho compreendendo:

- um primeiro e um segundo registos de dimensão finita (104,106) para armazenar um parâmetro A e um parâmetro C, respectivamente, o parâmetro C tendo uma relação com um

limite dum intervalo de valor e o parâmetro A tendo uma relação com a dimensão do dito intervalo,

- meios de entrada (100,102) para receber um símbolo do sinal de informação e pelo menos um valor de probabilidade correspondente para o símbolo associado para codificar,

- meios de recuperação para recuperar os valores para os parâmetros A e C nos primeiro e segundo registos, respectivamente,

- meios para dividir o intervalo de valor correspondente ao valor recuperado do primeiro registo em sub-intervalos correspondendo ao dito pelo menos um primeiro valor de probabilidade, e seleccionar um dos sub-intervalos em resposta ao dito símbolo,

- meios para actualizar pelo menos o parâmetro A de maneira a levar o seu valor em conformidade com a dimensão do sub-intervalo seleccionado com vista a tornar-se a nova dimensão do intervalo para codificar o próximo símbolo do sinal de informação,

- meios para armazenar o valor actualizado para o parâmetro A no primeiro registo,

caracterizado por os meios de recuperação compreenderem igualmente meios para truncar o valor do parâmetro A $0.b_0b_1\dots b_{i-1}b_i$ no bit b_{i-1} e se $b_{i-1}='0'$ e $b_i='1'$, elevar b_{i-1} a $'1'$.

17. Aparelho para codificar aritmeticamente um sinal de informação numérico compreendendo uma sequência série de símbolos de n-bits, n sendo um número inteiro tal que $n \geq 1$, o aparelho compreendendo:

- um primeiro e um segundo registos de dimensão finita (104,106) para armazenar um parâmetro A e um parâmetro C, respectivamente, o parâmetro C tendo uma relação com um

limite dum intervalo de valor e o parâmetro A tendo uma relação com a dimensão do dito intervalo,

- meios de entrada (100,102) para receber um símbolo do sinal de informação e pelo menos um valor de probabilidade correspondente para o símbolo associado para codificar,

- meios de recuperação para recuperar os valores para os parâmetros A e C nos primeiro e segundo registos, respectivamente,

- meios para dividir o intervalo de valor correspondente ao valor recuperado do primeiro registo em sub-intervalos correspondendo ao dito pelo menos um primeiro valor de probabilidade, e seleccionar um dos sub-intervalos em resposta ao dito símbolo,

- meios para actualizar pelo menos o parâmetro A de maneira a levar o seu valor em conformidade com a dimensão do sub-intervalo seleccionado com vista a tornar-se a nova dimensão do intervalo para codificar o próximo símbolo do sinal de informação,

- meios para armazenar o valor actualizado para o parâmetro A no primeiro registo,

caracterizado por os meios de recuperação compreenderem igualmente meios para truncar o valor do parâmetro A $0.b_0b_1.....b_{i-1}b_i$ no bit b_{i-1} e tornar o bit b_{i-1} igual a '1'.

18. Aparelho segundo as reivindicações 15, 16 ou 17 compreendendo igualmente meios para re-normalizar os valores para os parâmetros A e C, antes de armazenar os valores re-normalizados para os parâmetros A e C no primeiro e segundo registos, respectivamente, caracterizado por os meios de re-normalização compreenderem meios para

(g1) comparar o valor para A com um primeiro valor binário, se o valor para A não for menor que o dito valor binário,

deixar a etapa de re-normalização, e se o valor para A for menor que o primeiro valor binário, então

(g2) multiplicar o valor para A com um primeiro valor de número inteiro,

19. Aparelho segundo a reivindicação 18, caracterizado por os meios de re-normalização compreenderem ainda meios para.

(g4) comparar o valor para C com um segundo e um terceiro valor binário, o segundo valor binário sendo superior ao dito terceiro valor binário,

(g4) subtrair um quarto valor binário ao valor para C de maneira a obter um valor intermédio para C

(g5) multiplicar o valor intermédio para C com um segundo valor de número inteiro.

20. Aparelho segundo uma qualquer das reivindicações 15 a 19, caracterizado por compreender igualmente meios (112) para a codificação de canal do sinal de informação codificado num sinal codificado de canal.

21. Aparelho segundo a reivindicação 20, caracterizado por compreender igualmente meios (114) para registar o sinal de canal sobre uma portadora de registo.

22. Aparelho para descodificar aritmeticamente um sinal de informação codificado aritmeticamente num sinal de informação compreendendo uma sequência série de símbolos de n bits, n sendo um número inteiro tal que $n \geq 1$, o aparelho compreendendo:

- meios de entrada (120) para receber o sinal de informação codificado aritmeticamente,
- um primeiro e um segundo registo de dimensão finita (122,124), o primeiro registo (122) para armazenar um

parâmetro A, o parâmetro A tendo uma relação com a dimensão dum intervalo de valor, o segundo registo (124) para armazenar um parâmetro C, o conteúdo do segundo registo antes duma etapa de descodificação sendo obtido a partir do conteúdo do segundo registo numa etapa de descodificação precedente, decalando m bits do sinal de informação codificado aritmeticamente no segundo registo, m sendo um numero inteiro variável tal que $m \geq 0$,

- meios de geração (138) para gerar pelo menos um valor de probabilidade para um símbolo associado a ser descodificado,
- meios de recuperação para recuperar os valores para os parâmetros A e C a partir do primeiro e segundo registos, respectivamente,
- meios de derivação para derivar um símbolo em resposta ao dito pelo menos um valor de probabilidade, e em resposta a um valor para A e a um valor para C,
- meios para actualizar pelo menos o parâmetro A com vista a tornar-se a nova dimensão do intervalo para descodificar o símbolo seguinte do sinal de informação,
- meios para produzir (128) o símbolo descodificado,
- meios para armazenar o valor actualizado para o parâmetro A no primeiro registo,

caracterizado por os meios de recuperação compreenderem igualmente meios para truncar o valor do parâmetro A $0.b_0b_1\dots b_{i-1}b_i$ no bit b_{i-1} e adicionar '1' na posição do bit b_{i-1} no valor truncado de A, se b_i for igual a '1'.

23. Aparelho para descodificar aritmeticamente um sinal de informação codificado aritmeticamente num sinal de informação compreendendo uma sequência série de símbolos de n bits, n sendo um numero inteiro tal que $n \geq 1$, o aparelho compreendendo:

- meios de entrada (120) para receber o sinal de informação codificado aritmeticamente,
- um primeiro e um segundo registo de dimensão finita (122,124), o primeiro registo (122) para armazenar um parâmetro A, o parâmetro A tendo uma relação com a dimensão dum intervalo de valor, o segundo registo (124) para armazenar um parâmetro C, o conteúdo do segundo registo antes duma etapa de descodificação sendo obtido a partir do conteúdo do segundo registo numa etapa de descodificação precedente, decalando m bits do sinal de informação codificado aritmeticamente no segundo registo, m sendo um numero inteiro variável tal que $m \geq 0$,
- meios de geração (138) para gerar pelo menos um valor de probabilidade para um símbolo associado a ser descodificado,
- meios de recuperação para recuperar os valores para os parâmetros A e C a partir do primeiro e segundo registos, respectivamente,
- meios de derivação para derivar um símbolo em resposta ao dito pelo menos um valor de probabilidade, e em resposta a um valor para A e a um valor para C,
- meios para actualizar pelo menos o parâmetro A com vista a tornar-se a nova dimensão do intervalo para descodificar o símbolo seguinte do sinal de informação,
- meios para produzir (128) o símbolo descodificado,
- meios para armazenar o valor actualizado para o parâmetro A no primeiro registo,

caracterizado por os meios de recuperação compreenderem igualmente meios para truncar o valor do parâmetro A $0.b_0b_1\dots b_{i-1}b_i$ no bit b_{i-1} e se $b_{i-1}='0'$ e $b_i='1'$, elevar b_{i-1} a $'1'$.

24. Aparelho para descodificar aritmeticamente um sinal de informação codificado aritmeticamente num sinal de informação compreendendo uma sequência série de símbolos de n bits, n

sendo um número inteiro tal que $n \geq 1$, o aparelho compreendendo:

- meios de entrada (120) para receber o sinal de informação codificado aritmeticamente,
- um primeiro e um segundo registo de dimensão finita (122,124), o primeiro registo (122) para armazenar um parâmetro A, o parâmetro A tendo uma relação com a dimensão dum intervalo de valor, o segundo registo (124) para armazenar um parâmetro C, o conteúdo do segundo registo antes duma etapa de descodificação sendo obtido a partir do conteúdo do segundo registo numa etapa de descodificação precedente, decalando m bits do sinal de informação codificado aritmeticamente no segundo registo, m sendo um numero inteiro variável tal que $m \geq 0$,
- meios de geração (138) para gerar pelo menos um valor de probabilidade para um símbolo associado a ser descodificado,
- meios de recuperação para recuperar os valores para os parâmetros A e C a partir do primeiro e segundo registos, respectivamente,
- meios de derivação para derivar um símbolo em resposta ao dito pelo menos um valor de probabilidade, e em resposta a um valor para A e a um valor para C,
- meios para actualizar pelo menos o parâmetro A com vista a tornar-se a nova dimensão do intervalo para descodificar o símbolo seguinte do sinal de informação,
- meios para produzir (128) o símbolo descodificado,
- meios para armazenar o valor actualizado para o parâmetro A no primeiro registo,

caracterizado por os meios de recuperação compreenderem igualmente meios para truncar o valor do parâmetro A $0.b_0b_1.....b_{i-1}b_i$ no bit b_{i-1} e tornar o bit b_{i-1} igual a '1'.

25. O aparelho de descodificação segundo as reivindicações 22, 23 ou 24 caracterizado por compreender igualmente meios de descodificação de canal (132) para descodificar o canal do sinal de informação codificado aritmeticamente, antes da descodificação aritmética.

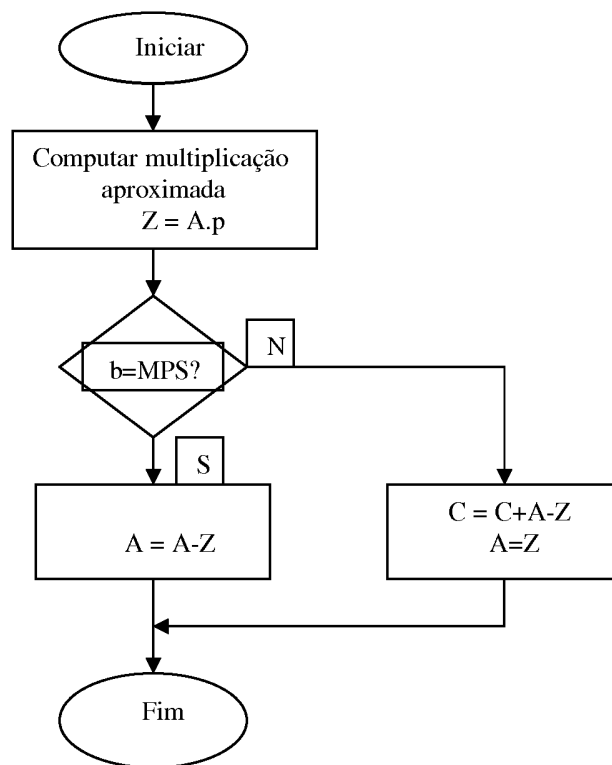
26. Aparelho segundo a reivindicação 25 caracterizado por compreender igualmente meios de leitura (130) para ler o sinal de informação codificado aritmeticamente com codificação de canal a partir duma portadora de registo.

10-12-2009

RESUMO

CODIFICAÇÃO E DESCODIFICAÇÃO ARITMÉTICA DUM SINAL DE INFORMAÇÃO

A invenção diz respeito a medidas para melhorar um codificador aritmético e um correspondente decodificador aritmético. Mais especificamente, são dadas propostas para truncar o parâmetro A , antes de realizar a multiplicação $A.p$. Além disso, também é dada uma proposta para o controlo vai-um na etapa de re-normalização no codificador.



1/9

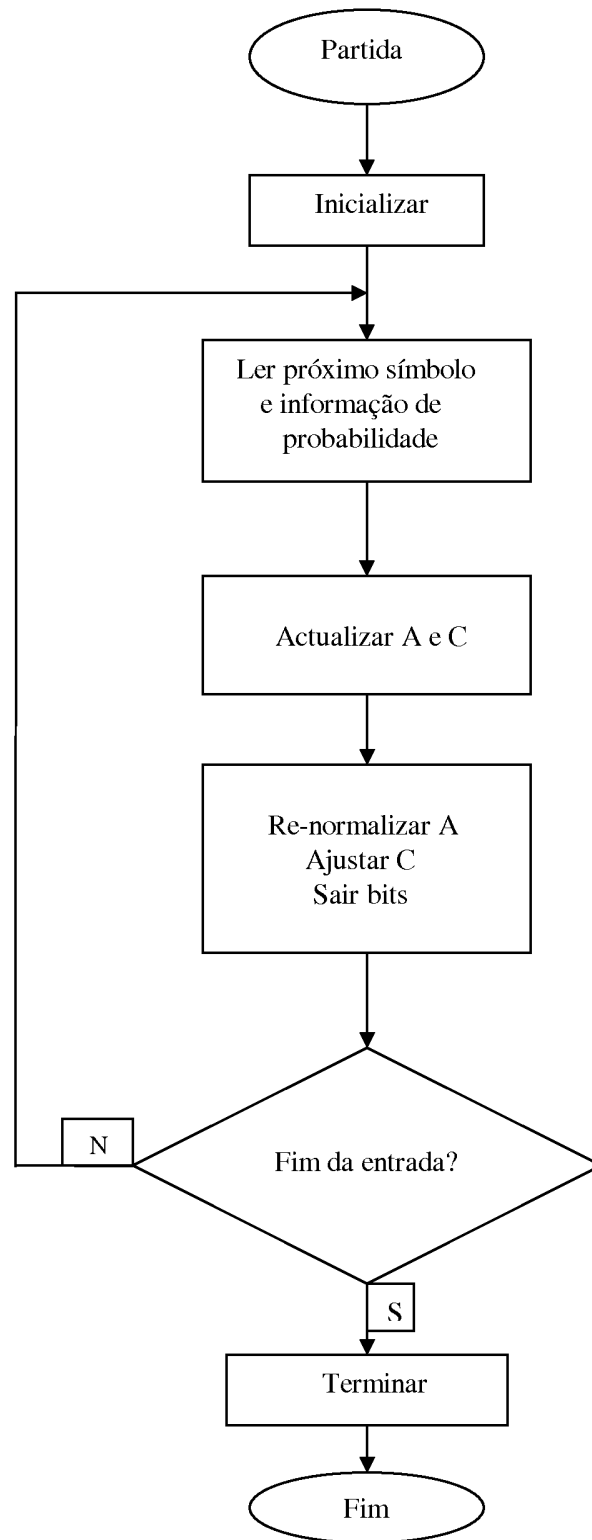


FIG. 1

2/9

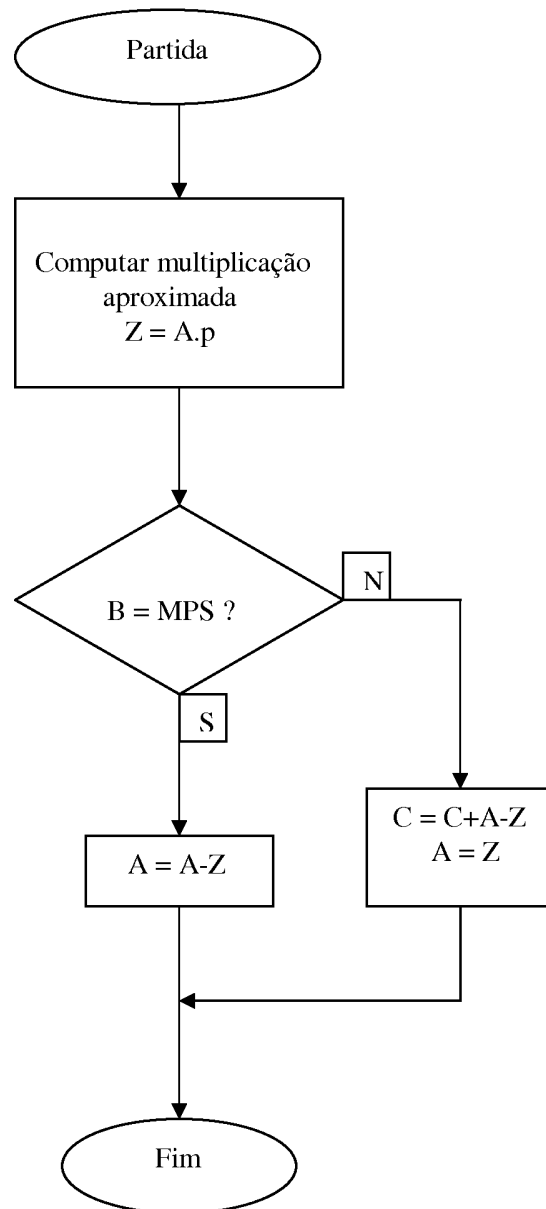


FIG. 2

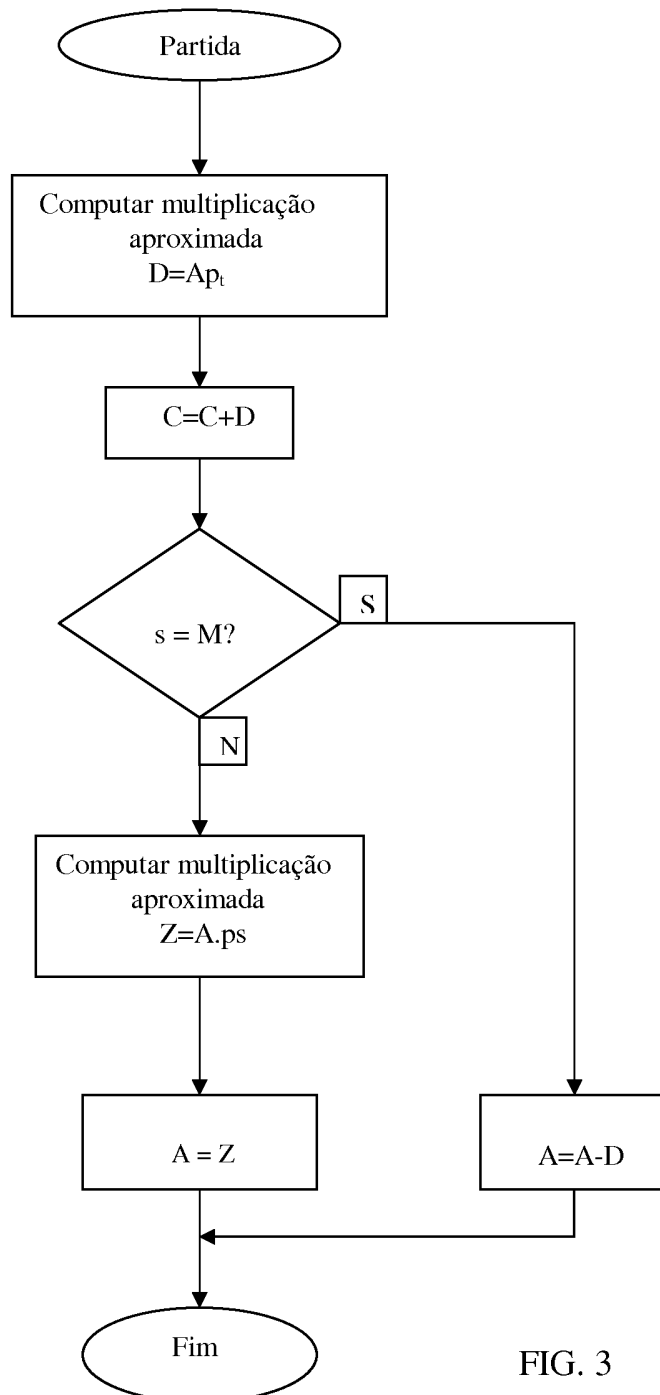


FIG. 3

4 / 9

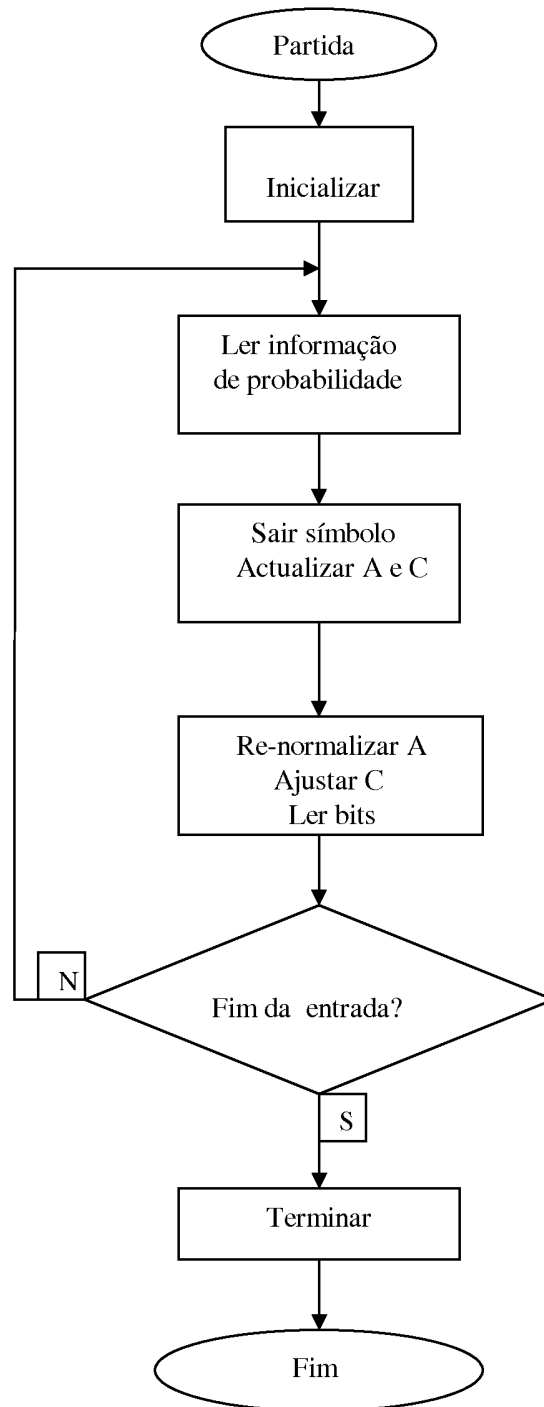


FIG. 4

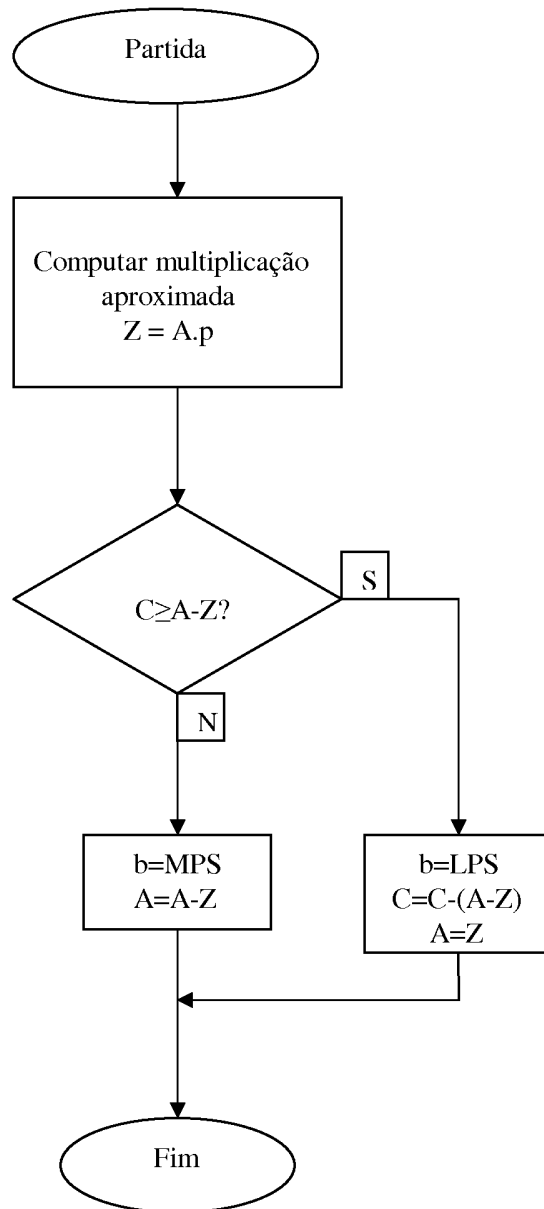


FIG. 5

6/9

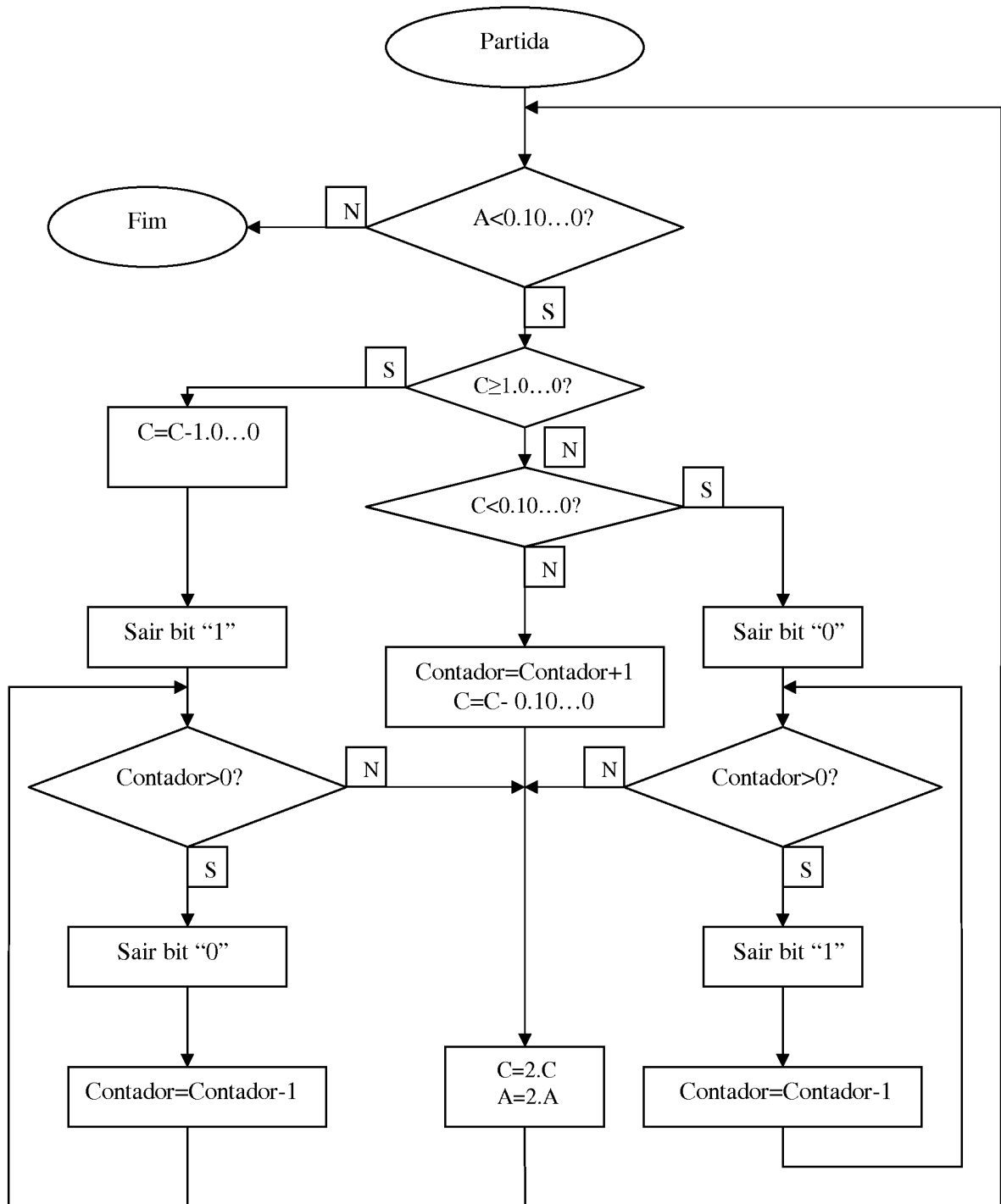


FIG. 6

7/9

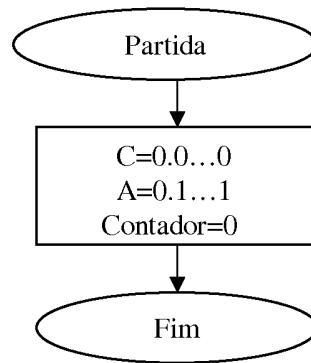


FIG. 7

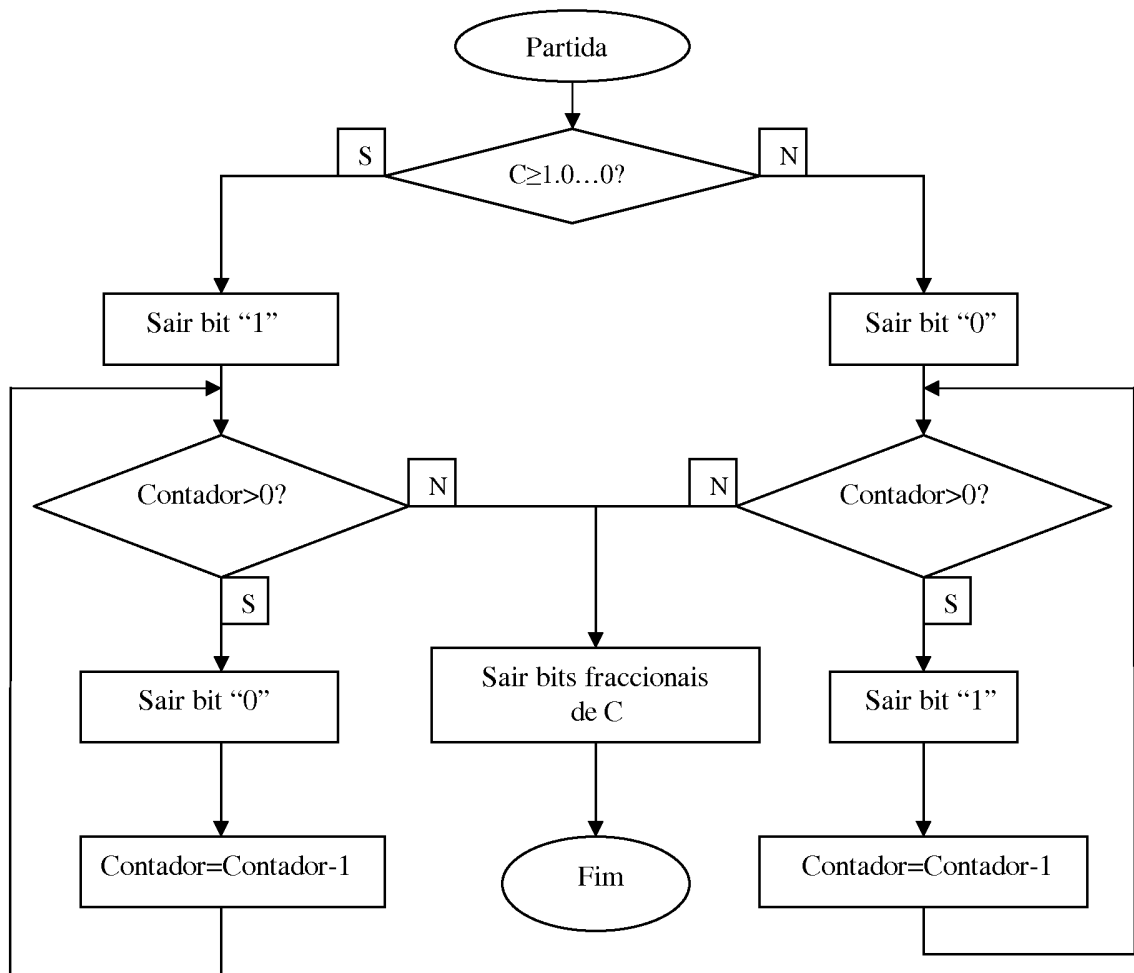


FIG. 8

8/9

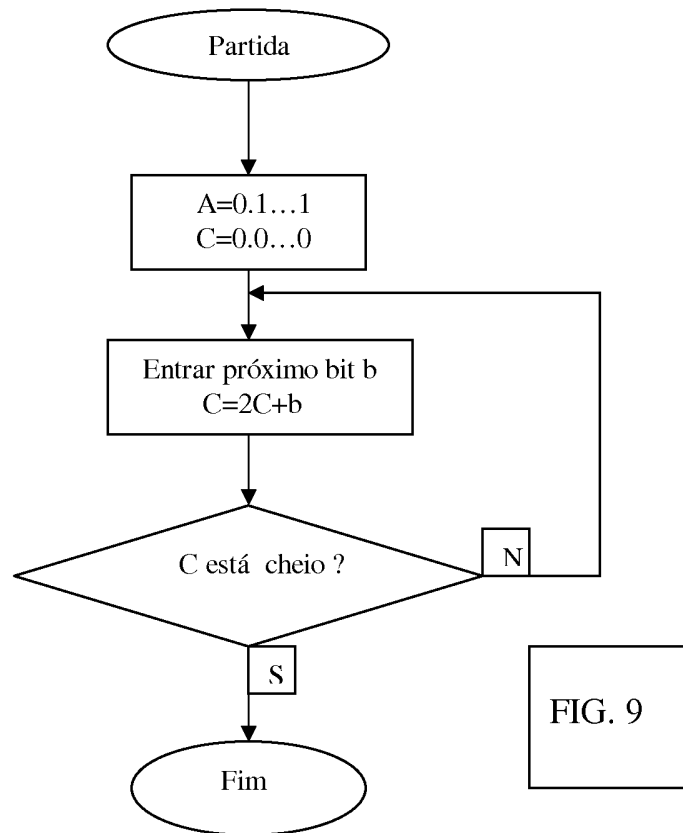


FIG. 9

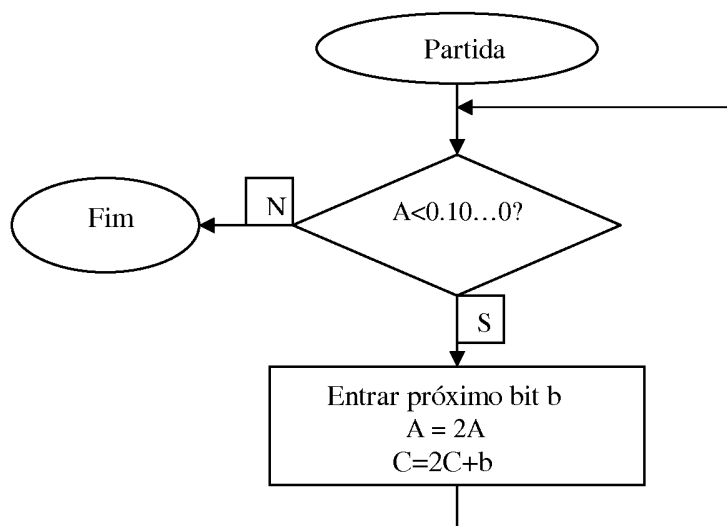
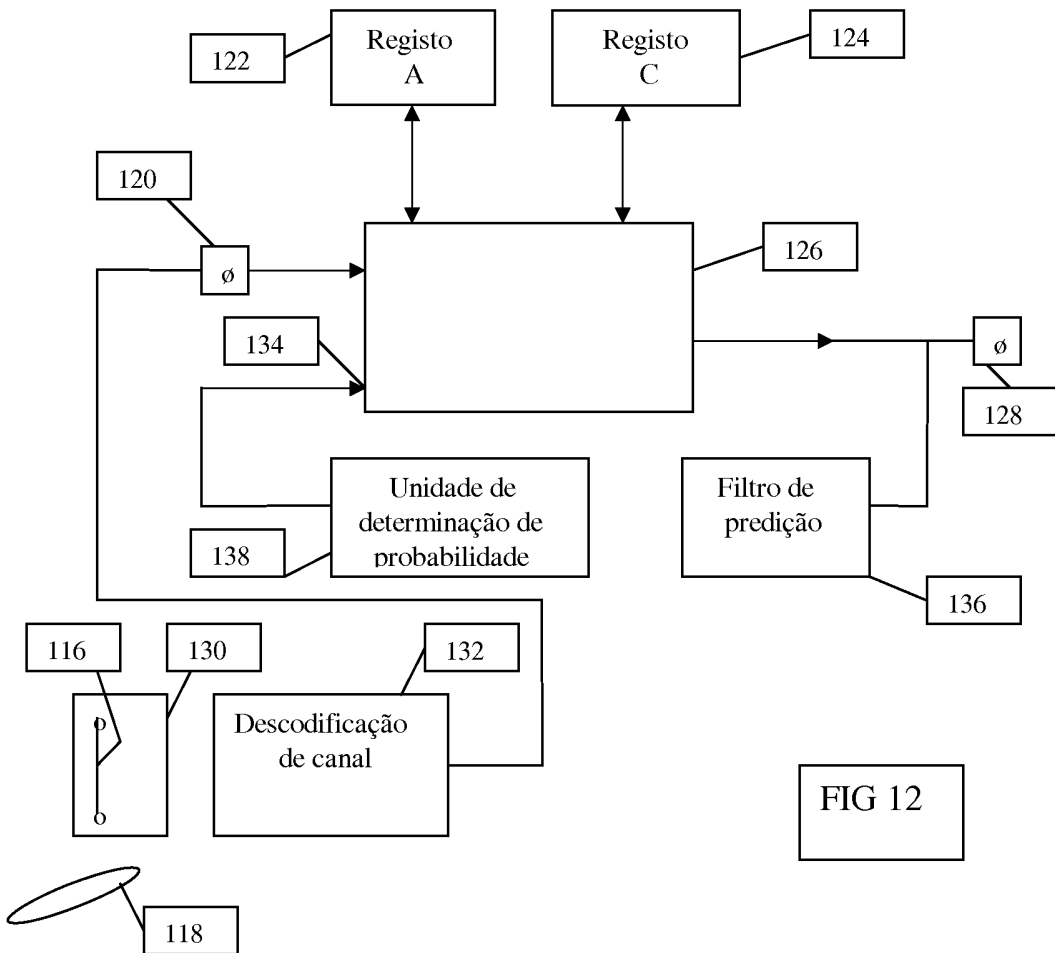
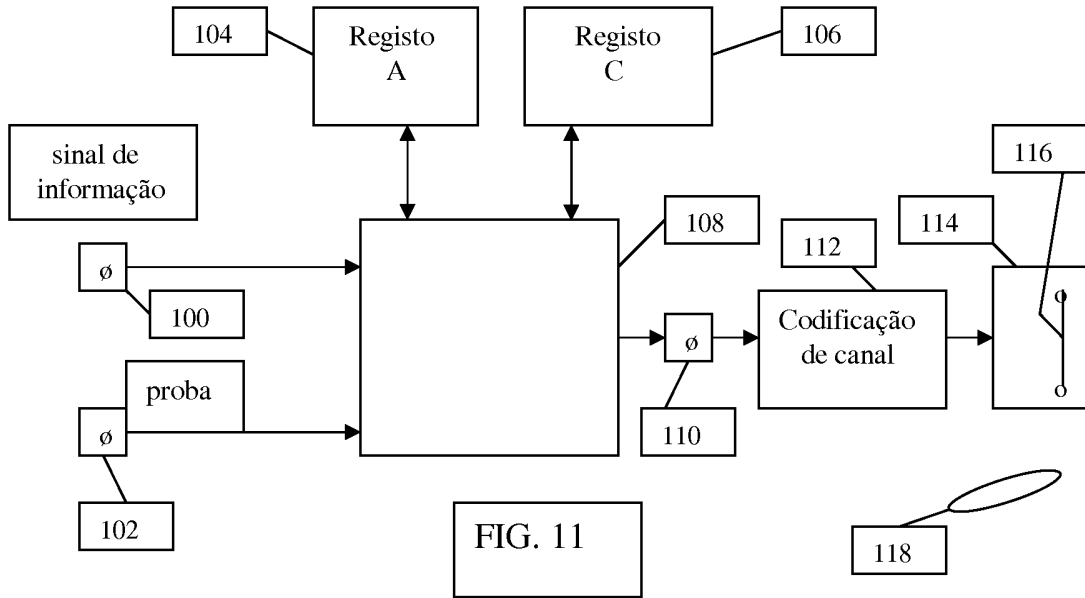


FIG. 10



77862



European Patent Office
80298 MUNICH
GERMANY
Tel. +49 (0)89 2399 - 0
Fax +49 (0)89 2399 - 4465



van der Kruk, Willem Leonardus
Philips
Intellectual Property & Standards
P.O. Box 220
5600 AE Eindhoven
PAYS-BAS

For any questions about
this communication:
Tel.: +31 (0)70 340 45 00

Date
17.09.09

Reference phn 16.822 ep	Application No./Patent No. 99902770.9 - 2223 / 0983635
Applicant/Proprietor Koninklijke Philips Electronics N.V., et al	

Decision to grant a European patent pursuant to Article 97(1) EPC

Following examination of European patent application No. 99902770.9 a European patent with the title and the supporting documents indicated in the communication pursuant to Rule 71(3) EPC dated 11.05.09 is hereby granted in respect of the designated Contracting States.

Patent No. : 0983635
Date of filing : 22.02.99
Priority claimed : 23.03.98/EPA 98200914

Designated Contracting States
and Proprietor(s) : AT BE CH DE ES FI FR GB IE IT LI NL PT SE
Koninklijke Philips Electronics N.V.
Groenewoudseweg 1
5621 BA Eindhoven/NL

SE
PHILIPS AB
Kottbygatan 7,
Kista
164 85 Stockholm/SE

This decision will take effect on the date on which the European Patent Bulletin mentions the grant (Art. 97(3) EPC).

The mention of the grant will be published in European Patent Bulletin 09/42 of 14.10.09.

Examining Division

Winkler G

Schneiderlin J

Zanella C

