(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 527 582 B9

(12) CORRECTED EUROPEAN PATENT SPECIFICATION

(15) Correction information:
Corrected version no 1 (W1 B1)
Corrections, see
Description Paragraph(s) 27, 28, 38
Claims EN

(48) Corrigendum issued on:
25.03.2009 Bulletin 2009/13

(45) Date of publication and mention of the grant of the patent:
13.02.2008 Bulletin 2008/07

(21) Application number: **03765793.9**

(22) Date of filing: **18.07.2003**

(51) Int Cl.:
H04L 27/30 *(2006.01)* ***H04J 11/00*** *(2006.01)*
H04Q 7/38 *(2006.01)* ***H04Q 7/36*** *(2006.01)*
H04B 7/26 *(2006.01)*

(86) International application number:
PCT/US2003/022637

(87) International publication number:
WO 2004/010591 (29.01.2004 Gazette 2004/05)

(54) **ORTHOGONAL VARIABLE SPREADING FACTOR (OVSF) CODE ASSIGNMENT**

ZUTEILUNG VON ORTHOGONALEN KODES MIT VARIABLEN SPREIZFAKTOREN (OVSF)

ATTRIBUTION DE CODE DE FACTEUR D’ETALEMENT VARIABLE ORTHOGONAL (OVSF)

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL PT RO SE SI SK TR

(30) Priority: **18.07.2002 US 396822 P**
31.07.2002 US 399864 P

(43) Date of publication of application:
04.05.2005 Bulletin 2005/18

(73) Proprietor: **INTERDIGITAL TECHNOLOGY CORPORATION**
Wilmington, DE 19801 (US)

(72) Inventors:
- **ZUNIGA, Juan, Carlos**
Montreal, Quebec H4L 3J4 (CA)
- **IACONO, Ana, Lucia**
Garden City, NY 11530 (US)
- **STERN-BERKOWITZ, Janet**
Little Neck, NY 11363 (US)

(74) Representative: **Giver, Sören Bo et al**
Awapatent AB
P.O. Box 5117
200 71 Malmö (SE)

(56) References cited:

EP-A- 1 035 676	**US-A- 6 091 757**
US-A- 6 163 524	**US-A1- 2002 051 437**
US-B1- 6 222 875	**US-B1- 6 233 231**
US-B1- 6 400 755	**US-B1- 6 400 755**
US-B1- 6 526 065	**US-B2- 6 552 996**

- **TSENG Y-C ET AL: "CODE PLACEMENT AND REPLACEMENT STRATEGIES FOR WIDEBAND CDMA OVSF CODE TREE MANAGEMENT" GLOBECOM’01. 2001 IEEE GLOBAL TELECOMMUNICATIONS CONFERENCE. SAN ANTONIO, TX, NOV. 25 - 29, 2001, IEEE GLOBAL TELECOMMUNICATIONS CONFERENCE, NEW YORK, NY : IEEE, US, vol. VOL. 1 OF 6, 25 November 2001 (2001-11-25), pages 562-566, XP001090318 ISBN: 0-7803-7206-9**
- **MINN T ET AL: "DYNAMIC ASSIGNMENT OF ORTHOGONAL VARIABLE-SPREADING-FACTOR CODES INW-CDMA" IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE INC. NEW YORK, US, vol. 18, no. 8, August 2000 (2000-08), pages 1429-1440, XP000998091 ISSN: 0733-8716**
- **CHENG R-G ET AL: "OVSF CODE CHANNEL ASSIGNMENT FOR IMT-2000" VTC 2000-SPRING. 2000 IEEE 51ST. VEHICULAR TECHNOLOGY CONFERENCE PROCEEDINGS. TOKYO, JAPAN, MAY 15-18, 2000, IEEE VEHICULAR TECHNOLGY CONFERENCE, NEW YORK, NY : IEEE, US, vol. VOL. 3 OF 3. CONF. 51, 15 May 2000 (2000-05-15), pages 2188-2192, XP000968392 ISBN: 0-7803-5719-1**

Printed by Jouve, 75001 PARIS (FR)

Description

[0001] This invention generally relates to wireless communication systems. In particular, the invention relates to code assignment is such systems.

[0002] BACKGROUND

[0003] In code division multiple access (CDMA) communication systems, communications are sent over a shared frequency spectrum. To minimize cross interference between the transmitted signals, orthogonal codes are used. Communications transmitted with orthogonal codes through an ideal wireless channel experience no cross code correlation, although in practice due to multipath, the orthogonal nature of the codes may break down to some extent.

[0004] Figure 1 is an illustration of a code tree for orthogonal variable spreading factors (OVSF). At the top of the tree is a code for a spreading factor of one. Codes in each row further down the tree have a spreading factor twice the factor of the row above that code. For the code tree of the time division duplex (TDD) mode of a third generation partnership project (3GPP) wideband code division multiple access (WCDMA) communication system, the maximum spreading factor as shown in Figure 1 is sixteen (16). In other systems, the maximum spreading factor may be significantly larger, such as 256 or 512.

[0005] Figure 1 is an illustration of an OVSF code tree having a maximum spreading factor of 16. The tree has one code of spreading factor 1, C1(1) of a value "1". Two codes, C2(2) having a value of "1,1" and C2(1) having a value of "1,-1", have a spreading factor of 2. Four codes, C4(4) to C4(1), have a spreading factor of 4. Eight codes, C8(8) to C8(1) have a spreading factor of 8 and sixteen codes, C16(16) to C16(1) have a spreading factor of 16.

[0006] The lines connecting the codes in the tree identify codes that are not orthogonal to each other in the tree. Codes connected by lines in either only an upward or downward direction are not orthogonal. To illustrate, code C 16(16) is not orthogonal to code C2(2), since tracing a path from code C16(16) to code C2(2) can be done using four upward connecting lines. Conversely, codes C16(16) and C8(7) are orthogonal to each other. Tracing a path from C16(16) to C8(7) involves using two upward connecting lines and one downward connecting line.

[0007] To minimize interference between communications, it is desirable to only assign codes that are orthogonal to each other. It is also desirable to assign the codes in an optimal manner such that the maximum number of orthogonal codes are available for use. In an operating wireless communication system as codes are assigned and released, unoptimal code assignments may occur reducing the system's capacity.

[0008] Figure 2A is an illustration of an unoptimal OVSF code assignment. One problem with using OVSF codes is the efficient use of the codes. After the assigning and release of codes, codes C16(16), C16(13), C16(9) and C16(5) are still active, as shown with filled circles. To prevent the assignment of a code not orthogonal to these codes, all the codes above these codes are blocked from assignment, as shown by an "X". Accordingly, the use of C16(5) blocks C8(3), C4(2), C2(1) and C1(1). As shown in Figure 2A, a total often codes are blocked, C1 (1), C2(1), C2(2), C4(2), C4(3), C4(4), C8(3), C8(5), C8 (7) and C8(8) are blocked. As a result of assigning four spreading factor (SF) sixteen codes, no SF 2 codes are available and only one SF4 code is available. Accordingly, services requiring an SF 2 code or multiple SF 4 codes can not be supported.

[0009] Figure 2B illustrates an efficient assignment of four SF 16 codes. Codes C16(16) to C16(13) are used. Only five codes are blocked as a result of this assignment, codes C1(1), C2(2), C4(4), C8(8) and C8(7). As a result, one SF 2 code is available and three SF 4 codes are available to support additional services. The assignment of Figure 2B allows for greater latitude in code assignment than Figure 2A.

[0010] Accordingly, it is desirable to have alternate approaches to code assignment.

In "Code Placement and Replacement Strategies for Wideband CDMA OVSF Code Tree Management", Nov 25-29, 2001, IEEE, pages 562-566, Tseng Y-C et al. disclose code reassignment in a OVSF code tree of a WCDMA system. When the code tree becomes fragmented, a dynamic code assignment algorithm is used to vacate the tree with a minimum cost by relocating busy codes one-by-one in a recursive manner. The steps of the algorithm are performed in case a new call requests a certain transmission rate, but there is no free code of such rate. The first step involves checking if there is enough free capacity in the code tree.

US 2002/051437 discloses code switching in a OVSF code tree. When a new call requests a transmission rate, and there are no assignable codes, code switching is performed in order to be able to accept the call request.

In "Dynamic Assignment Of Orthogonal Variable-Spreading-Factor Codes in W-CDMA", IEEE Journal on Selected Areas in Communications, Vol 18, NO. 8, August 2000, pages 1429-1440, Minn T. et al. disclose elimination of code blocking in a OVSF code tree by reassigning codes. The scheme minimizes the codes that must be assigned to support a new call. At the time of a new call request, it is checked whether the system can support the rate required by the new call. If the system has excess capacity to support the new call, the codes are reassigned.

EP 1035676 discloses a method of communicating a selected channelisation code for a downlink to a user.

[0011] SUMMARY

[0012] The invention includes a system and methods for orthogonal variable spreading factor (OVSF) code assignment, de-allocation and code tree pruning. The invention also includes OVSF code identification and storage schemes. Embodiments are provided for code tree pruning, code assignment and de-allocation. These embodiments apply to both slotted and non-slotted code division multiple access systems.

[0013] BRIEF DESCRIPTION OF THE DRAWING(S)

[0014] Figure 1 is an illustration of an OVSF code tree.

[0015] Figures 2A and 2B are illustrations of code allocations and the resulting blocked codes.

[0016] Figure 3 is a simplified block diagram of an OVSF code assignment system.

[0017] Figure 4 is a preferred OVSF code identification scheme.

[0018] Figure 5 is an illustration of a code vector.

[0019] Figure 6 is a flow chart for OVSF code tree pruning over multiple time slots.

[0020] Figure 7 is a flow chart for OVSF code tree pruning.

[0021] Figure 8 is a flow chart of periodic code tree pruning.

[0022] Figure 9 is a flow chart of OVSF code assignment and code tree updating procedures.

[0023] Figure 10 is a flow chart of OVSF code de-allocation and code tree updating procedures.

[0024] Figure 11 is a flow chart modifying Figure 8 for use with a default midamble code assignment scheme.

[0025] DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT(S)

[0026] Hereafter, a wireless transmit/receive unit (WTRU) includes but is not limited to a user equipment, mobile station, fixed or mobile subscriber unit, pager, or any other type of device capable of operating in a wireless environment.

[0027] Figure 3 is a simplified diagram of a wireless communication system using OVSF code assignment. A radio network controller (RNC) 26 has a radio resource management (RRM) device 46 for use in assigning and releasing codes. The RRM device 46 has an associated memory for use in storing the code assignments and other information. A Node-B has a code assignment device 42 which receives the resource allocation from the RRM device 46. The code assignment device 42 assigns codes to the uplink, downlink or both communications. A transmitter (TX) 40 uses the downlink code assignments to transfer communications through the downlink channel(s) 36 of the wireless interface 22. A receiver (RX) 38 uses the uplink code assignments to receive communications over the uplink channel(s) 34. A WTRU 20 has a code assignment device 32. for assigning uplink and downlink codes. The code assignments may be signaled from the RRM device 46 to the WTRU 20. The TX 28 uses the uplink code assignments to send uplink commumications and the RX 30 uses the downlink code assignments to receive downlink communications.

[0028] The following are embodiments for assigning and/or reassigning OVSF codes. Figure 4 is an illustration of a preferred technique for identifying the codes of the code tree. In the technique of Figure 4, the codes are identified as sequential numbers. The spreading factor 1 code, C1(1), is identified as 1. The spreading factor 2 codes are identified as: C2(2) as 2 and C2(1) as 3. The spreading factor 4 codes are identified as: C4(4) as 4, C4(3) as 5, C4(2) as 6, C4(1) as 7. The spreading factor 8 codes are identified as: C8(8) as 8, C8(7) as 9, C8(6) as 10, C8(5) as 11, C8(4) as 12, C8(3) as 13, C8(2) as 14 and C8(1) as 16. The spreading factor 16 codes are identified as: C16(16) as 16, C16(15) as 17, C16(14) as 18, C16(13) as 19, C16(12) as 20, C16(11) as 21, C16 (10) as 22, C16(9) as 23, C16(8) as 24, C16(7) as 25, C16(6) as 26, C16(5) as 27, C16(4) as 28, C16(3) as 29, C16(2) as 30 and C16(1) as 31. For systems using spreading factors larger than 16, such as FDD/CDMA, the code tree can be expanded accordingly. Although Figure 4 illustrates one number scheme other versions may be used. To illustrate the sequential numbering across the rows may be done from right to left or in another order.

[0029] As shown at the bottom of Figure 4, the sequential code identifiers can be grouped by spreading factor and remain in sequence. To illustrates, SF 16 code identifiers are identifiers 16 to 31. The use of such a code identifier technique allows for reduced storage space for code assignments. Figure 5 is an illustration of such a code assignment storage. Each code identifier, 1 to 31, has an associated bit or word 48_1 to 48_{31} to indicate the current status of the code, such as being "currently assigned", "blocked" or "available". A "blocked" code is a code that cannot be assigned due to another "currently assigned" code. To illustrate, C8(1) is blocked if either C16(1), C16(2), C4(1), C2(1) or C(1) is being used. Additional status information may be provided, such as an indicator of the number of blocking codes may also be provided.

[0030] This code identification storage approach allows for essentially a concise vector to contain the code assignment information, instead of utilizing some form of list or table. However, the following code assignment approaches can be used or adapted for use with a vector, table, list or other storage techniques.

[0031] One preferred code status word uses a two bit element. The first bit indicates if the code is blocked. A "0" indicates that the code is not blocked and a "1" indicates that the code is blocked. The second bit indicates if the code is blocked by one code in its sub-tree or both codes in the sub-tree. A "0" indicates one code and a "1" indicates two codes. For example, if code 8 is used and code 9 is free, then code 4 is blocked by one code only, and it is marked as "10"; if codes 8 and 9 are used, then code 4 is blocked by both codes in its sub-tree, and it is marked as "11". If a code is being used currently, it is marked as "11". "01" preferably indicates that the code cannot be used in that timeslot.

[0032] Figure 6 is a preferred tree pruning approach for use in slotted CDMA communication systems, such as in TDD/CDMA. Tree pruning is a reassignment of codes to allow for more flexibility in future code assignments. Although the following describes code pruning in a left to right fashion, other code assignment orders can be used such as right to left among others.

[0033] A new call is to be added and codes are assigned to the call, step 100. The pruning is begun with a first time slot of set of time slots for potential assignment and reassignment. To illustrate, in a TDD/CDMA system in which slots 2, 4, 6 and 10 may be used for uplink transmissions, slot 2 could be deemed the first slot, slot 4 the second and so on. The leftmost available code of the highest spreading factor (SF) in that time slot is evaluated, step 102. If there are no codes of that SF available in the first time slot, subsequent time slots are searched until a code is found. Although the flow chart of Figure 6 is described as by moving from a highest to lowest spreading factor, the differing spreading factors can be processed from lowest to highest or in another order.

[0034] After a code is found, a highest SF code in any time slot that can be moved left into this time slot is determined, step 104. That determined code is relocated leftwards, step 110. If other codes can be moved leftwards, these codes are also relocated, step 112. The relocation procedure is repeated for the next time slot, steps 106 and 108. If there are no time slots that have not been evaluated, the leftward relocation is performed for the next higher spreading factor, steps 114 and 118. This process is repeated until the lowest spreading factor is reached, such as SF 1, step 116. Codes having an SF of 1 are preferably not relocated and the process ends.

[0035] The algorithm of Figure 6 is preferably invoked for every new call. The algorithm, in summary, starts pruning the trees with the lowest layer codes and from the rightmost timeslot and code. If a code can be reallocated to any other place to the left, the code is moved. After all the possible codes in the lowest layer are moved, the algorithm moves up one level and repeats the same procedure. It continues until one layer before the highest is reached (i.e. in the TDD case, codes with SF1 typically do not need to be reallocated).

[0036] Another approach, as per Figure 7, limits the tree pruning to pruning within individual time slots. This code approach can be applied also to non-slotted system as well, After the new call is assigned a code, step 120, the rightmost code with a highest SF is selected, step 122. If the rightmost highest SF code can be relocated leftward, step 124, the code is relocated leftward, step 130. If another highest SF code can be relocated leftward, step 132, it is relocated, step 130. The leftward relocating is repeated for each lower spreading factor in order, step 126. After the lowest SF is reached, such as SF 1, step 128, the procedure is ended.

[0037] In summary, after every new arrival, the algorithm of Figure 7 is invoked. The algorithm starts pruning the tree with the lowest layer codes and from the rightmost code. If a code can be reallocated to any other place to the left, the code is moved. In the preferred approach, each code is reallocated as far to the left as possible. After all the possible codes in the lowest layer are moved, the algorithm moves up one level and repeats the same procedure. It continues until one layer before the highest is reached.

[0038] Figure 8 is another preferred algorithm for tree pruning. This algorithm acts only periodically and limits the maximum number of reallocations. Such an approach reduces the overall processing required. Additionally, the overhead associated with reassignment is also reduced. This algorithm also looks at the tree's code fragmentation. Code tree fragmentation is a measure of the number of blocked codes within that tree. Although this approach is described with its preferred application in a slotted CDMA system, it can also be applied to non-slotted systems.

[0039] After a certain number of arrivals, such as "X", step 134, the algorithm is invoked. The algorithm initially looks at the fragmentation of the trees in each time slot. If at least one of the trees is fragmented more than a certain threshold, such as T, step 136, the algorithm starts pruning the trees with the lowest layer codes and from the rightmost timeslot and code, step 138. If a code can be reallocated to any other place to the left, step 150, the code is moved and a counter, such as tracked by a variable "Num_reall", that keeps track of the number of reallocations is updated. If the maximum number of reallocations is reached, step 152, the algorithm ends. If the maximum number of reallocations is not reached, the algorithm continues. After all the possible codes in the highest spreading factor are moved, steps 154 and 142, the algorithm moves up to the next highest spreading factor and repeats the same procedure, steps 146 and 148. The procedure is repeated for each lower spreading factor, unit the lowest spreading factor, steps 148, 156.

[0040] Variations to these code management schemes may also be used. Codes with a low SF can be allocated right-to-left, whereas codes with high SF can be allocated from left-to-right, thus concentrating codes for similar services in the same timeslots. The code tree pruning algorithms may also make use of the code management scheme in order to give priority to arrivals that request either a high or low SF.

[0041] Figure 9 is a preferred algorithm for assigning OVSF codes. A preferred use for this algorithm is in uplink code assignment for the TDD/CDMA, although the algorithm can be used for other applications. The preferred algorithm uses a code vector having two bit code status elements as previously described and the code tree of Figure 4.

[0042] Initially, a variable, such as "x" is set to Q, where Q is the SF of the code requiring assignment, step 200. An initial node having that SF in the tree is checked to see if it is free (v(x) = "00"), step 202. If it is, the parent node (the connected node having a lower spreading factor) is checked to see if it is blocked ("10"), step 204. If it is blocked, node x is assigned, step 216. A variable "S" representing the assignment is set to "x", step 218, and the algorithm proceeds to step 220.

[0043] If the parent node is not blocked, assigning that node may not be optimal ("non_optimal" = "x"), step 206. The next node is checked (x =x+1), step 208. If the next node has the same SF (x < 2*Q), step 210, the algorithm goes to step 202 to determine if that node is optimal or not. If not, the "non_optimal" node is selected for assignment, step 212. The variable "S" is set to "non-optimal", step 214, and the algorithm proceeds to step 220.

[0044] The element of node S is marked as assigned (v(S) = "11"), step 220. The connected node having an SF lower than the assigned node is updated (Upnode = ⌊s/2⌋), step 222. IfUpNode = 0, that indicates that all the nodes have been updated, step 224, and the algorithm proceeds to step 234. If not v(UpNode) = "00", indicating that the node is available, step 226, that node is blocked (v(UpNode) = "10"), step 228. The next higher node (having a lower SF) is checked (UpNode = ⌊UpNode/2⌋, step 230. Ifv(UpNode) = "11" (indicating that the node was blocked by one and now is blocked by two codes), step 232, the procedure to go up the tree is finished, step 234.

[0045] The algorithm proceeds to update down the tree (initialize DownNode = S*2), step 236. The number of nodes to be checked below the current node is two (initialize number_nodes = 2), step 238. The algorithm checks to see that the bottom of the tree has not been passed (DownNode < 32), step 240. If the bottom is passed, the update is complete, step 254. If the bottom was not passed, a count is initialized (initialize count = 0), step 242. The node indicated by the count is blocked (v(DownNode+count) = "11"), step 244. The algorithm increments the count and proceeds to the next node (count = count + 1), step 248. If count is < number_nodes, step 248, the algorithm checks the next node and goes to step 244. If count is >= number_nodes, the algorithm checks on level lower in the tree (DownNode = 2*DownNode), step 250. The nodes checked in the lower level is doubled (number_nodes = 2*number_nodes), step 252.

[0046] Figures 10a and 10b show a preferred algorithm to update a tree after a code has been released (deallocated). A variable "S" is set to the value of the node being de-allocated, step 256. That node, S, is marked as free (v(S) = "00"), step 258. The nodes higher in the tree are updated (initialize UpNode = [s/2]), step 260. The algorithm checks to see if the top of the tree has not been passed (UpNode = 0), step 262. If the top of the tree is passed, lower nodes in the tree are checked, step 272. If the top is not passed, the current node is blocked by one code (v(UpNode) = "10"), step 264, that node is marked as being available (v(UpNode) = "00"), step 266. The algorithm proceeds to check the next higher node in the tree (UpNode = [UpNode/2]), step 268, and the algorithm checks this node. If the node was blocked by two codes, it is set as being blocked by one node (v(UpNode) = "10"), step 270.

[0047] To update down the tree, the "DownNode" variable is initialized (DownNode = S*2), step 274. The number of nodes to update below the node is initialized (number_nodes = 2), step 276. If the bottom of the tree is passed (DownNode >= 32), the tree has been updated, step 292. If not (DownNode < 32), a count is initialized (count = 0), step 280. The node indicated by the count is freed for assignment (v(DownNode+count) = "00"), step 282. The count is incremented (count = count + 1), step 284. If the number of nodes updated is not passed (count < number_nodes), the updating is repeated for the next count, step 286. If it is passed (count >= number_nodes), the next level down the tree is checked (DownNode = 2*DownNode), step 288. The number of analyzed nodes in the lower level is twice that of the prior level (number_nodes = 2* number_nodes), step 290, and the updating is repeated.

[0048] One embodiment of the invention relates to code assignment for the default midamble allocation scheme of the TDD mode of 3GPP W-CDMA. Midambles are part of the physical channel configuration for TDD. Different midamble allocation schemes exist for TDD. One is a WTRU specific midamble allocation, applying to both the uplink and downlink. For the WTRU specific midamble allocation, a specific midamble is explicitly assigned to all physical channels of a specific user. Another scheme is default midamble allocation, which also applies to both the uplink and downlink. For this scheme, the midamble used is determined by a pre-defined association between the midambles and the assigned channelization codes.

[0049] Another scheme is the common midamble allocation, which applies only to the downlink. In this scheme, a common midamble is allocated to all physical channels in a specific timeslot. For the WTRU specific and common midamble schemes, the codes may be assigned to users without considering the midamble assignment.

[0050] However, for the default midamble, the association between the midambles and the channelization codes is used in the code allocation scheme. Depending on the burst type and the value of Kcell, as defined by the standards, different associations apply. The burst type and the value of Kcell are individually configurable in the Radio Network Controller (RNC) for each timeslot by, for example, an operations and maintenance (O& M) function, or a function in the RNC such as radio resource management (RRM) dynamic channel allocation (DCA). The configuration is signaled to the WTRU via RRC signaling and, accordingly, the association is known by the WTRU.

[0051] For default midamble allocation, the association between midambles and channelization codes defines primary and secondary channelization codes. The following rules apply: each secondary channelization code is associated with a single primary channelization code; each secondary channelization code uses the same midamble as the primary channelization code with which it is associated; secondary codes can only be allocated if the associated primary code is also allocated; all channelization codes associated with the same midamble can only be allocated to the same WTRU; and for common and WTRU specific midamble allocations, all codes can be considered to be primary codes.

[0052] In order to perform code assignment for the default midamble scheme, each code is preferably given an additional flag indicating whether it is a primary or secondary channelization code, a midamble designation (which could also be called a code group designation) with one of Kcell different possible values (e.g., 0 through Kcell-1), and a tag to be used for the WTRU association.

[0053] Figure 11 is a flow chart for replacing steps 200-220 of Figure 9 for use in allocating codes for the default midamble scheme. The tag for WTRU association could be provided for each channelization code or just for each midamble, since each code is already associated with a given midamble. The association with a midamble for a WTRU would imply association with that WTRU.

[0054] The procedure attempts to find the most optimal available code (code marked "00") for a given user (WTRU). The first (most desirable) choice is a secondary code associated with a midamble that is already assigned to the user. This type of code is classified as a class 1 code. The second choice is a primary code that has a parent, which is already blocked by one code (parent marked as "10"). This code is classified as a class 2 code. The third choice is a primary code that has a parent which is not blocked, which is classified as a class 3 code. Secondary codes associated with a midamble belonging to a different user would not be considered for assignment.

[0055] The most preferred algorithm scans from left to right, starting at the left-most code in the tree in that level, although other scanning directions may be used. For each available code, we determine which, if any, class it belongs to. If a class 1 is found, an immediate assignment is made. Otherwise, the left most codes in classes 2 and/or 3 are remembered for possible assignment in the case the scan ends without finding a class 1 code.

[0056] The following variables are used in the preferred algorithm: Optimal_Primary for a code in class 2 and Non_Optimal_Primary for a code in class 3. A variable, x, is initialized the value of the spreading factor, x = Q, step 300. Q is the SF of the code to allocate. Optimal_Primary and Non_Optimal_Primary are set to zero. The variable v(x) is checked to see if it equals "00", indicating that it is free, step 302. Ifx is a secondary code, step 304, and the midamble associated with code x is assigned to the user, step 306, the selected code S is set to x (S=x), step 308. Then, the algorithm jumps to step 336.

[0057] If x is not a secondary and is a primary, step 310, and Optimal_Primary = 0, step 312, (an optimal code has not been found optimal yet), the left most primary code is saved as optimal for later use, if a secondary code is not found. The parent code is checked to see if it is blocked by one (v(x/2) = "10"), step 314. If it is, Optimal_Primary is set to the current x (Optimal_Primary = 0), step 320, and the next code is checked (x=x+1), step 322. If the parent is blocked, a check is done to determine whether a non-optimal primary code is found (Non_Optimal_Primary = 0), step 318. Subsequently, the next code is checked (x=x+1), step 322.

[0058] If the next code is within the same spreading factor, (x<2*Q), step 324, the procedure is repeated for the next code and the algorithm goes to step 302. If the next code is not in the same spreading factor, a check is done to see if an optimal primary code is found (Optimal_Primary = 0), step 326. If an optimal primary has been found, the code is set to the optimal primary (S = Optimal_Primary), step 328. The midamble of the assigned code S is assigned to the user, step 330.

[0059] If a non-optimal primary is found (Non_Optimal_Primary = 0), step 332, S is set to the non-optimal primary (S = Non_Optimal_Primary), step 334. The midamble of the assigned code S is assigned to the user, step 330. After the midamble is assigned, the assigned code is marked as being assigned (v(S) = "11"), step 336, and the code tree is updated accordingly, step 338. If no secondary or primary codes are available, a code cannot be assigned, step 340.

[0060] Since the de-allocation procedure simply consists of updating the code tree, the previously described procedure for code de-allocation can be used in case of default midamble allocation with the addition of one step. After the completion of the de-allocation of a primary code, the midamble associated with that code would be disassociated from that user (WTRU).

[0061] Preferably, the entity that invokes the procedure after deciding which code to de-allocate (e.g., a Radio Resource Management function) ensures that a primary code is only de-allocated if all its secondary codes are not assigned. Optionally, a step could be added at the beginning of the de-allocation procedure to verify that a primary code is not being de-allocated when its associated secondary codes are still assigned. If any associated secondary is still assigned, the de-allocation procedure would exit and indicate a code assignment failure.

Claims

1. A method for reassigning codes of an orthogonal variable spreading factor, OVSF, code tree in a code division multiple access communication system, the code tree having a plurality of codes at differing spreading factors and having a plurality of branches between the codes of different spreading factors, if a particular code of the code tree is assigned, a lower spreading factor code of the code tree connected to that code by one of the branches is blocked from assignment, if a code of the code tree is blocked, a code of the tree of a lower spreading factor connected to that code by one of the branches is blocked from assignment, the method comprising:

    starting at a first spreading factor (step 138):

        (a) determining selected assigned codes of that spreading factor that can be reassigned to other codes of that spreading factor in a desired direction of the code tree (step 140); and
        (b) reassigning the selected assigned codes to the other codes (steps 150, 154);

and for each other spreading factor, repeating steps (a) and (b) (steps 146, 156),

characterised in that the steps of determining and reassigning are invoked periodically and are performed if the code tree has a number of blocked codes which is above a certain threshold (step 136).

2. The method of claim 1 wherein the other codes are within a same time slot as the selected codes.

3. The method of claim 1 wherein at least one of the other codes is in a different time slot than the selected codes.

4. The method of claim 1 wherein the first spreading factor is a highest spreading factor.

5. A radio network controller, RNC, having a radio resource management device, the radio resource management device having a memory for storing a representation of an orthogonal variable spreading factor, OVSF, code tree, the code tree having a plurality of codes at differing spreading factors and having a plurality of branches between the codes of different spreading factors, if a particular code of the code tree is assigned, a lower spreading factor code of the code tree connected to that code by one of the branches is blocked from assignment, if a code of the code tree is blocked, a code of the tree of a lower spreading factor connected to that code by one of the branches is blocked from assignment, the radio resource management device capable of performing steps comprising:

starting at a highest spreading factor.

(a) determining selected assigned codes of that spreading factor that can be reassigned to other codes of that spreading factor in a desired direction of the code tree (step 140); and
(b) reassigning the selected assigned codes to the other codes (steps 150,154); and for each lower spreading factor in sequence, repeating steps (a) and (b) (steps 146, 156)

characterised in that the steps of determining and reassigning are invoked periodically and are performed if the code tree has a number of blocked codes which is above a certain threshold (step 136).

6. The RNC of claim 7 wherein the other code are within a same time slot as the selected codes.

7. The RNC of claim 7 wherein at least one of the other codes is in a different time slot than the selected codes.

8. The RNC of claim 5 wherein the representation of the code tree is stored as a vector.

9. The RNC of claim 8 wherein the vector includes an element for each code of the code tree in sequence.

10. The RNC of claim 9 wherein all the element of codes having a same spreading factor are contiguous.

11. The RNC of claim 9 wherein each element has two bits, one bit of the two bits indicating whether the code of that element is blocked and another bit of the two bits indicating whether one or multiple codes are blocked by the code of that element.

Patentansprüche

1. Verfahren zum Neuzuweisen von Kodes eines Kodebaums mit einem orthogonalen variablen Spreizfaktor OVSF in einem Codemultiplex-Vielfachzugriff-Kommunikationssystem, wobei der Kodebaum eine Vielzahl von Kodes mit unterschiedlichen Spreizfaktoren hat und eine Vielzahl von Zweigen zwischen den Kodes mit unterschiedlichen Spreizfaktoren hat, wobei, wenn ein bestimmter Kode des Kodebaums zugewiesen wird, ein niedrigerer Spreizfaktorkode des Kodebaums, der durch einen der Zweige mit diesem Kode verbunden ist, gegen eine Zuweisung gesperrt wird, wobei, wenn ein Kode des Kodebaums gesperrt ist, ein Kode des Baums mit einem niedrigeren Spreizfaktor, der durch einen der Zweige mit diesem Kode verbunden ist, gegen eine Zuweisung gesperrt wird, wobei das Verfahren aufweist:

beginnend mit dem ersten Spreizfaktor (Schritt 138):

(a) Bestimmen ausgewählter zugewiesener Kodes mit diesem Spreizfaktor, die an andere Kodes mit diesem Spreizfaktor in einer gewünschten Richtung des Kodebaums neu zugewiesen werden können (Schritt 140); und
(b) Neuzuweisen der ausgewählten zugewiesenen Kodes an die anderen Kodes (Schritte 150, 154); und Wiederholen der Schritte (a) und (b) für jeden anderen Spreizfaktor (Schritte 146, 156),

dadurch gekennzeichnet, daß die Schritte zum Bestimmen und Neuzuweisen regelmäßig aufgerufen werden und durchgeführt werden, wenn der Kodebaum eine Anzahl gesperrter Kodes hat, die über einer gewissen Schwelle ist (Schritt 136).

2. Verfahren nach Anspruch 1, wobei die anderen Kodes in dem gleichen Zeitschlitz sind wie die ausgewählten Kodes.

3. Verfahren nach Anspruch 1, wobei zumindest einer der anderen Kodes in einem zu den ausgewählten Kodes unterschiedlichen Zeitschlitz ist.

4. Verfahren nach Anspruch 1, wobei der erste Spreizfaktor ein höchster Spreizfaktor ist.

5. Funknetzsteuerung RNC mit einer Funkressourcenverwaltungsvorrichtung, wobei die Funkressourcenverwaltungsvorrichtung einen Speicher zum Speichern einer Darstellung eines Kodebaums mit einem orthogonalen variablen Spreizfaktor OVSF hat, wobei der Kodebaum eine Vielzahl von Kodes mit unterschiedlichen Spreizfaktoren hat und eine Vielzahl von Zweigen zwischen den Kodes mit unterschiedlichen Spreizfaktoren hat, wobei, wenn ein bestimmter Kode des Kodebaums zugewiesen wird, ein niedrigerer Spreizfaktorkode des Kodebaums, der durch einen der Zweige mit diesem Kode verbunden ist, gegen eine Zuweisung gesperrt wird, wobei, wenn ein Kode des Kodebaums gesperrt ist, ein Kode des Baums mit einem niedrigeren Spreizfaktor, der durch einen der Zweige mit diesem Kode verbunden ist, gegen eine Zuweisung gesperrt wird, wobei die Funkressourcenverwaltungsvorrichtung fähig ist, Schritte durchzuführen, die aufweisen:

beginnend mit einem höchsten Spreizfaktor:

(a) Bestimmen ausgewählter zugewiesener Kodes mit diesem Spreizfaktor, die an andere Kodes mit diesem Spreizfaktor in einer gewünschten Richtung des Kodebaums neu zugewiesen werden können (Schritt 140); und
(b) Neuzuweisen der ausgewählten zugewiesenen Kodes an die anderen Kodes (Schritte 150, 154); und der Reihe nach Wiederholen der Schritte (a) und (b) für jeden niedrigeren Spreizfaktor (Schritte 146, 156),

dadurch gekennzeichnet, daß die Schritte zum Bestimmen und Neuzuweisen regelmäßig aufgerufen werden und durchgeführt werden, wenn der Kodebaum eine Anzahl gesperrter Kodes hat, die über einer gewissen Schwelle ist (Schritt 136).

6. RNC nach Anspruch 7, wobei die anderen Kodes in dem gleichen Zeitschlitz sind wie die ausgewählten Kodes.

7. RNC nach Anspruch 7, wobei zumindest einer der anderen Kodes in einem zu den ausgewählten Kodes unterschiedlichen Zeitschlitz ist.

8. RNC nach Anspruch 5, wobei die Darstellung des Kodebaums als ein Vektor gespeichert ist.

9. RNC nach Anspruch 8, wobei der Vektor der Reihe nach ein Element für jeden Kode des Kodebaums umfaßt.

10. RNC nach Anspruch 9, wobei alle Elemente von Kodes mit einem gleichen Spreizfaktor benachbart sind.

11. RNC nach Anspruch 9, wobei jedes Element zwei Bits hat, wobei ein Bit der zwei Bits anzeigt, ob der Kode dieses Elements gesperrt ist und ein anderes Bit der zwei Bits anzeigt, ob einer oder mehrere Kodes durch den Kode dieses Elements gesperrt sind.

Revendications

1. Procédé de réattribution de codes d'un facteur d'étalement variable orthogonal, OVSF, d'un arbre de codes dans un système de communication à accès multiple par répartition de codes, l'arbre de codes ayant une pluralité de codes à des facteurs d'étalement différents et une pluralité de branches entre les codes de facteurs d'étalement différents, si un code particulier de l'arbre de codes est attribué, un code de facteur d'étalement inférieur de l'arbre de codes relié à ce code par une des branches est bloqué et ne peut pas être attribué, si un code de l'arbre de codes est bloqué, un code de l'arbre du facteur d'étalement inférieur relié à ce code par une des branches est bloqué et ne peut pas être attribué, le procédé comprenant les étapes suivantes:

en commençant à un premier facteur d'étalement (étape 138) :

(a) déterminer des codes attribués sélectionnés de ce facteur d'étalement pouvant être réattribués à d'autres codes de ce facteur d'étalement suivant une direction souhaitée de cet arbre de codes (étape 140) ; et
(b) réattribuer les codes attribués sélectionnés aux autres codes (étapes 150, 154) ; et pour chaque autre facteur d'étalement, répéter les étapes (a) et (b) (étapes 146 et 156) ;

caractérisé en ce que les étapes de détermination et de réattribution sont invoquées de manière périodique et sont réalisées si l'arbre de codes a un certain nombre de codes bloqués au-dessus d'un certain seuil (étape 136).

2. Procédé selon la revendication 1, dans lequel les autres codes sont compris dans le même intervalle temporel que celui des codes sélectionnés.

3. procédé selon la revendication 1, dans lequel au moins un des autres codes est dans un intervalle de temps différent de celui des codes sélectionnés.

4. Procédé selon la revendication 1, dans lequel le premier facteur d'étalement est un facteur d'étalement supérieur.

5. Contrôleur de réseau radio, RNC, ayant un dispositif de gestion de ressource radio, le dispositif de gestion de ressource radio ayant une mémoire destinée à mémoriser une représentation d'un arbre de codes à facteur d'étalement variable orthogonal, OVSF, l'arbre de codes ayant une pluralité de codes à des facteurs d'étalement différents et une pluralité de branches entre les codes de facteurs d'étalement différents, si un code particulier de l'arbre de codes est attribué, un code de facteur d'étalement inférieur de l'arbre de codes relié à ce code par une des branches est bloqué et ne peut pas être attribué, si un code de l'arbre de codes est bloqué, un code de l'arbre d'un facteur d'étalement inférieur relié à ce code par une des branches est bloqué et ne peut pas être attribué, le dispositif de gestion de ressource radio pouvant exécuter les étapes consistant à :

en commençant au niveau d'un facteur d'étalement supérieur :

(a) déterminer des codes attribués sélectionnés de ce facteur d'étalement pouvant être réattribués à d'autres codes de ce facteur d'étalement suivant une direction souhaitée de cet arbre de codes (étape 140) ; et
(b) réattribuer les codes attribués sélectionnés aux autres codes (étapes 150, 154) ; et pour chaque facteur d'étalement inférieur en séquence, répéter les étapes (a) et (b) (étapes 146, 156) ;

caractérisé en ce que les étapes de détermination et de réattribution sont invoquées de manière périodique et sont réalisées si l'arbre de codes a un certain nombre de codes bloqués au-dessus d'un certain seuil (étape 136).

6. RNC selon la revendication 7, dans lequel les autres codes sont compris dans un même intervalle de temps que celui des codes sélectionnés.

7. RNC selon la revendication 7, dans lequel au moins un des autres codes est compris dans un intervalle de temps différent de celui des codes sélectionnés.

8. RNC selon la revendication 5, dans lequel la représentation de l'arbre de codes est mémorisée en tant que vecteur.

9. RNC selon la revendication 8, dans lequel le vecteur comprend un élément pour chaque code de l'arbre de codes en séquence.

10. RNC selon la revendication 9, dans lequel l'ensemble des éléments des codes ayant un facteur d'étalement identique sont contigus.

11. RNC selon la revendication 9, dans lequel chaque élément a deux bits, un bit des deux bits indiquant si le code de cet élément est bloqué et un autre bit des deux bits indiquant si un ou plusieurs codes sont bloqués par le code de cet élément.

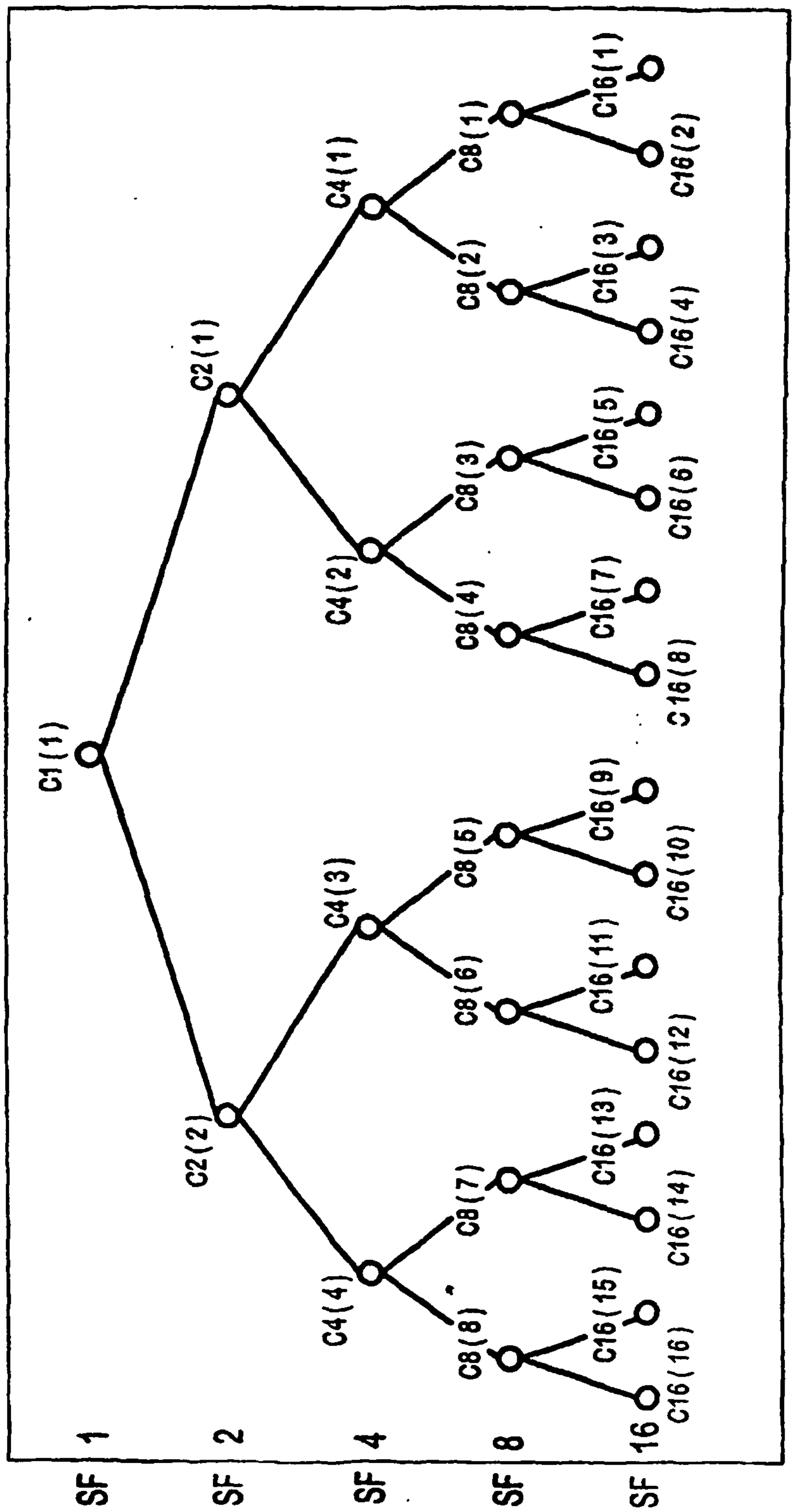

SF 1
SF 2
SF 4
SF 8
SF 16
C1(1)
C2(1)
C2(2)
C4(1)
C4(2)
C4(3)
C4(4)
C8(1)
C8(2)
C8(3)
C8(4)
C8(5)
C8(6)
C8(7)
C8(8)
C16(1)
C16(2)
C16(3)
C16(4)
C16(5)
C16(6)
C16(7)
C16(8)
C16(9)
C16(10)
C16(11)
C16(12)
C16(13)
C16(14)
C16(15)
C16(16)

FIG. 1

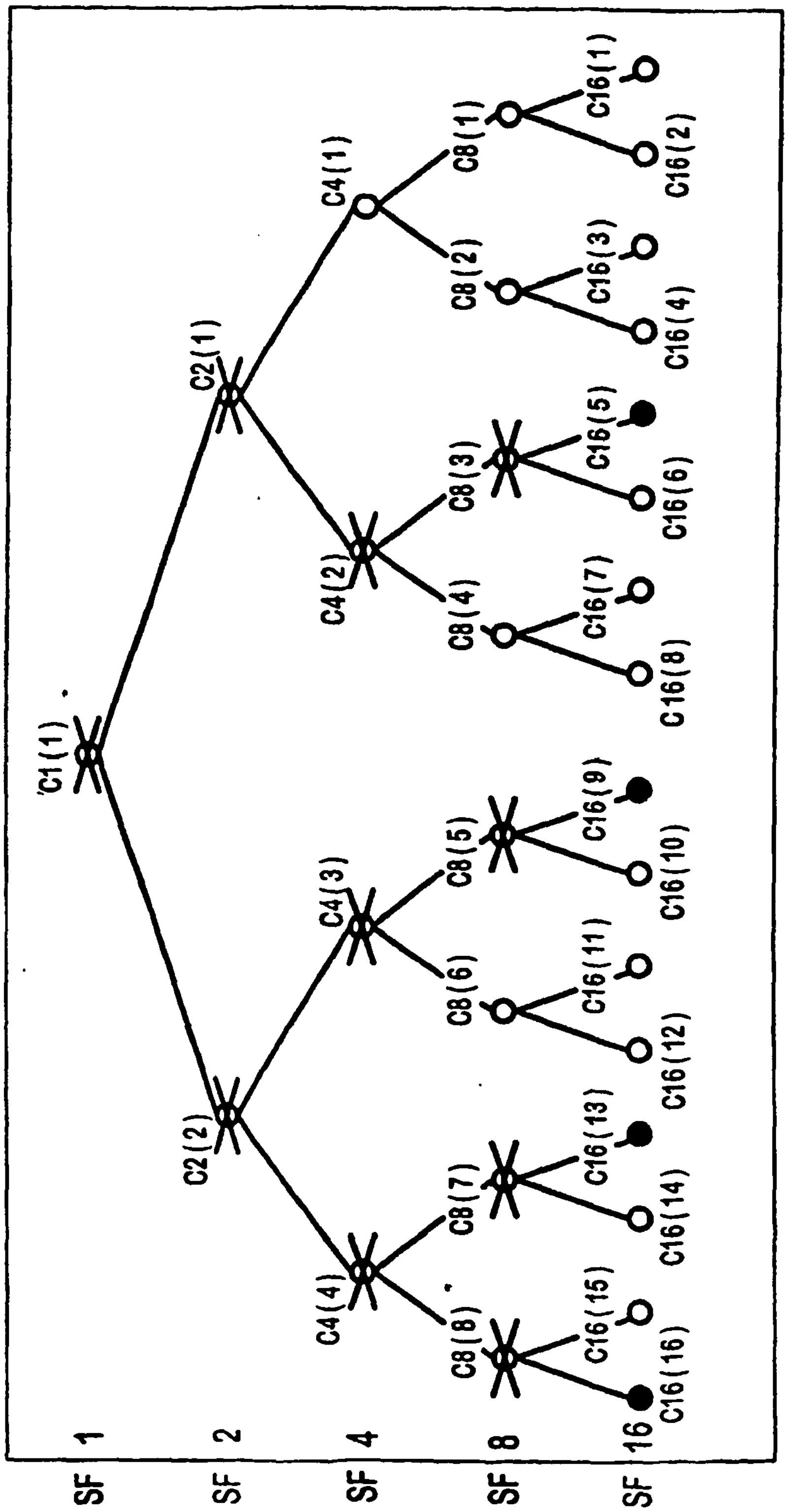
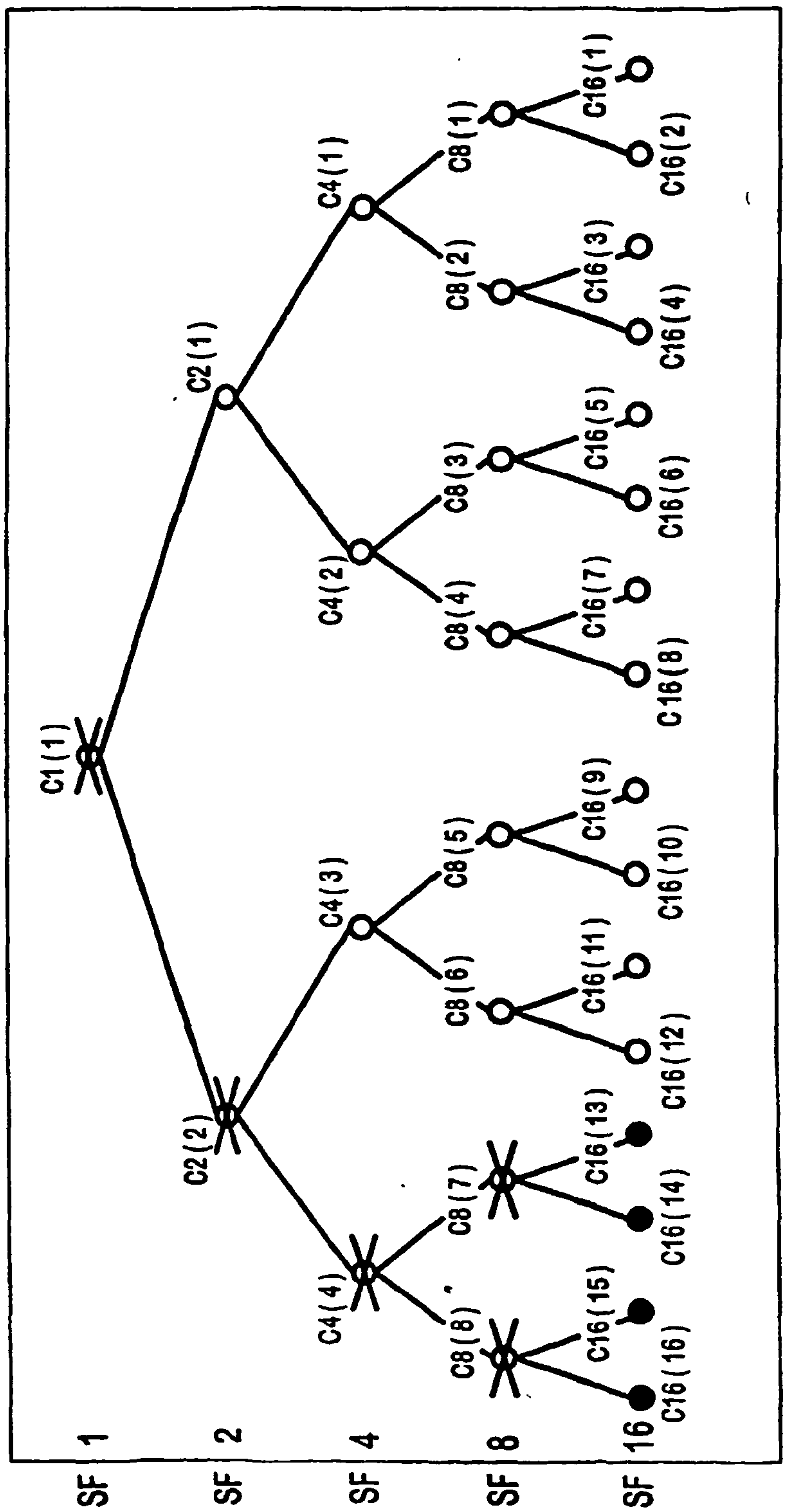

SF 1
SF 2
SF 4
SF 8
SF 16
C1(1)
C2(1)
C2(2)
C4(1)
C4(2)
C4(3)
C4(4)
C8(1)
C8(2)
C8(3)
C8(4)
C8(5)
C8(6)
C8(7)
C8(8)
C16(1)
C16(2)
C16(3)
C16(4)
C16(5)
C16(6)
C16(7)
C16(8)
C16(9)
C16(10)
C16(11)
C16(12)
C16(13)
C16(14)
C16(15)
C16(16)

FIG. 2A



SF 1
SF 2
SF 4
SF 8
SF 16
C1(1)
C2(1)
C2(2)
C4(1)
C4(2)
C4(3)
C4(4)
C8(1)
C8(2)
C8(3)
C8(4)
C8(5)
C8(6)
C8(7)
C8(8)
C16(1)
C16(2)
C16(3)
C16(4)
C16(5)
C16(6)
C16(7)
C16(8)
C16(9)
C16(10)
C16(11)
C16(12)
C16(13)
C16(14)
C16(15)
C16(16)

FIG. 2B

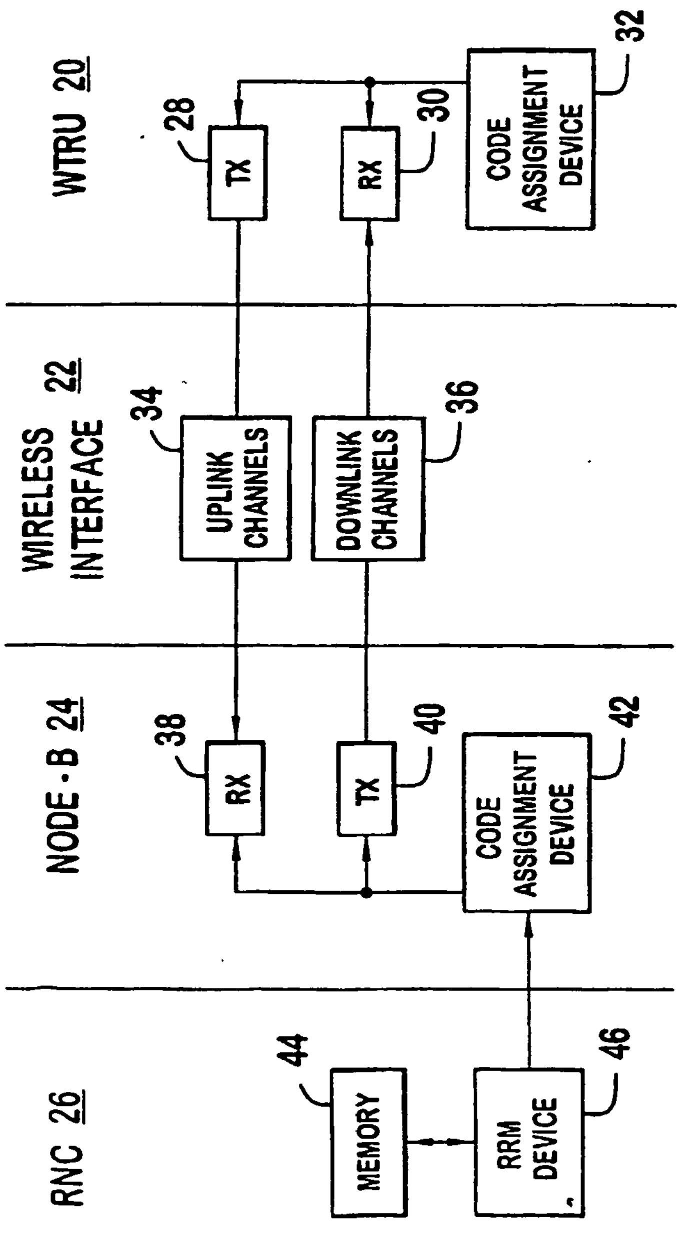

RNC 26
MEMORY
44
RRM DEVICE
46
NODE-B 24
RX
38
TX
40
CODE ASSIGNMENT DEVICE
42
WIRELESS INTERFACE 22
UPLINK CHANNELS
34
DOWNLINK CHANNELS
36
WTRU 20
TX
28
RX
30
CODE ASSIGNMENT DEVICE
32

FIG. 3

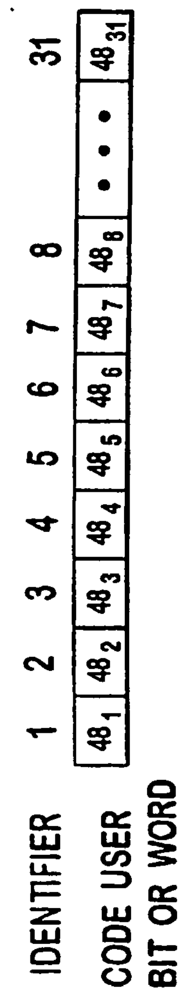

IDENTIFIER
1
2
3
4
5
6
7
8
31
CODE USER BIT OR WORD
48 1
48 2
48 3
48 4
48 5
48 6
48 7
48 8
• • •
48 31

FIG. 5

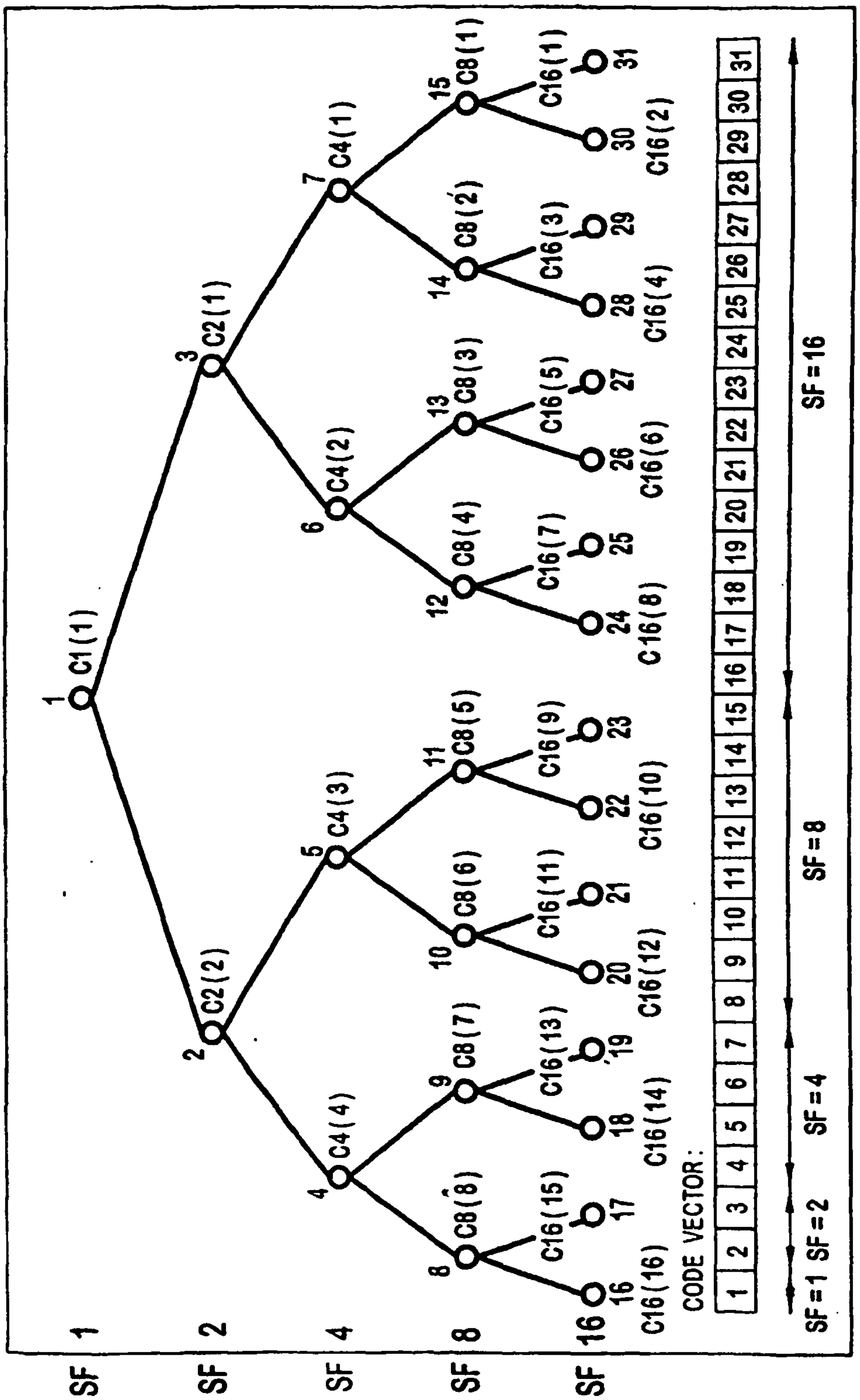

SF 1
SF 2
SF 4
SF 8
SF 16
C1(1)
C2(2)
C2(1)
C4(4)
C4(3)
C4(2)
C4(1)
C8(8)
C8(7)
C8(6)
C8(5)
C8(4)
C8(3)
C8(2)
C8(1)
C16(16)
C16(15)
C16(14)
C16(13)
C16(12)
C16(11)
C16(10)
C16(9)
C16(8)
C16(7)
C16(6)
C16(5)
C16(4)
C16(3)
C16(2)
C16(1)
CODE VECTOR:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
SF=1 SF=2 SF=4 SF=8 SF=16

FIG. 4

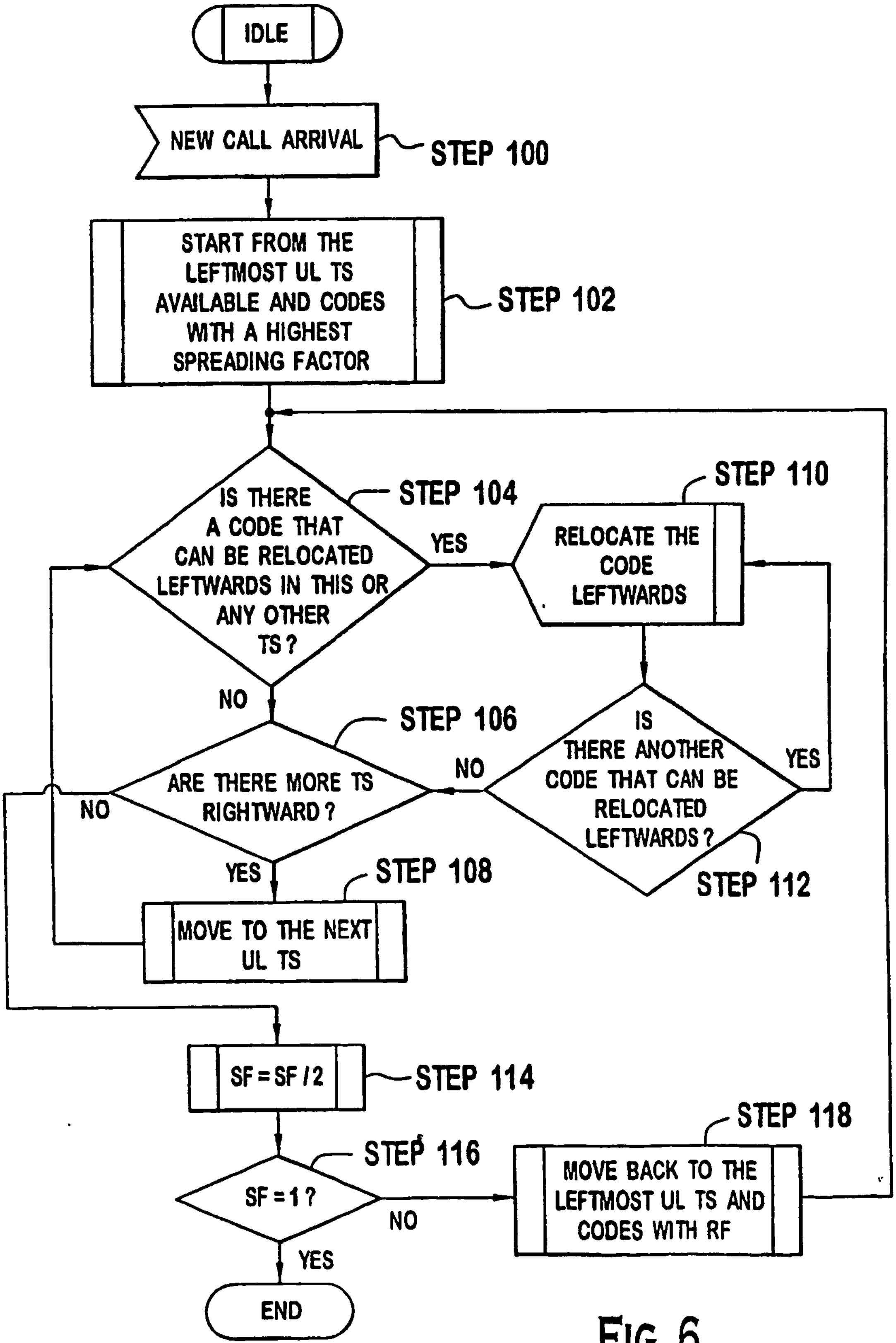

IDLE
NEW CALL ARRIVAL
STEP 100
START FROM THE LEFTMOST UL TS AVAILABLE AND CODES WITH A HIGHEST SPREADING FACTOR
STEP 102
IS THERE A CODE THAT CAN BE RELOCATED LEFTWARDS IN THIS OR ANY OTHER TS ?
STEP 104
YES
RELOCATE THE CODE LEFTWARDS
STEP 110
NO
ARE THERE MORE TS RIGHTWARD ?
STEP 106
NO
IS THERE ANOTHER CODE THAT CAN BE RELOCATED LEFTWARDS ?
YES
NO
STEP 112
YES
MOVE TO THE NEXT UL TS
STEP 108
SF = SF / 2
STEP 114
SF = 1 ?
STEP 116
NO
MOVE BACK TO THE LEFTMOST UL TS AND CODES WITH RF
STEP 118
YES
END

FIG. 6

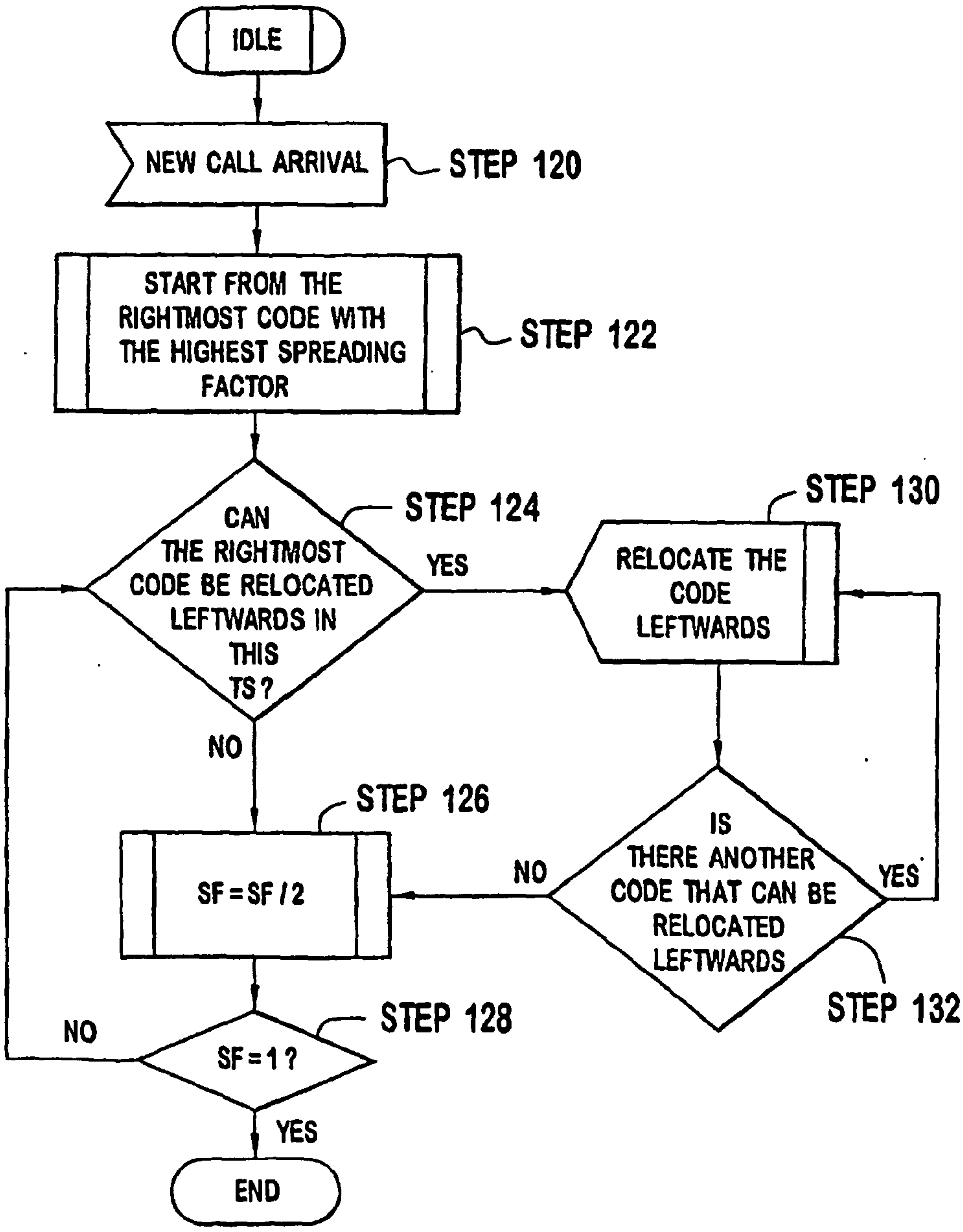

IDLE
NEW CALL ARRIVAL
STEP 120
START FROM THE RIGHTMOST CODE WITH THE HIGHEST SPREADING FACTOR
STEP 122
CAN THE RIGHTMOST CODE BE RELOCATED LEFTWARDS IN THIS TS ?
STEP 124
YES
NO
RELOCATE THE CODE LEFTWARDS
STEP 130
STEP 126
SF = SF / 2
NO
IS THERE ANOTHER CODE THAT CAN BE RELOCATED LEFTWARDS
YES
STEP 132
NO
SF = 1 ?
STEP 128
YES
END

FIG. 7

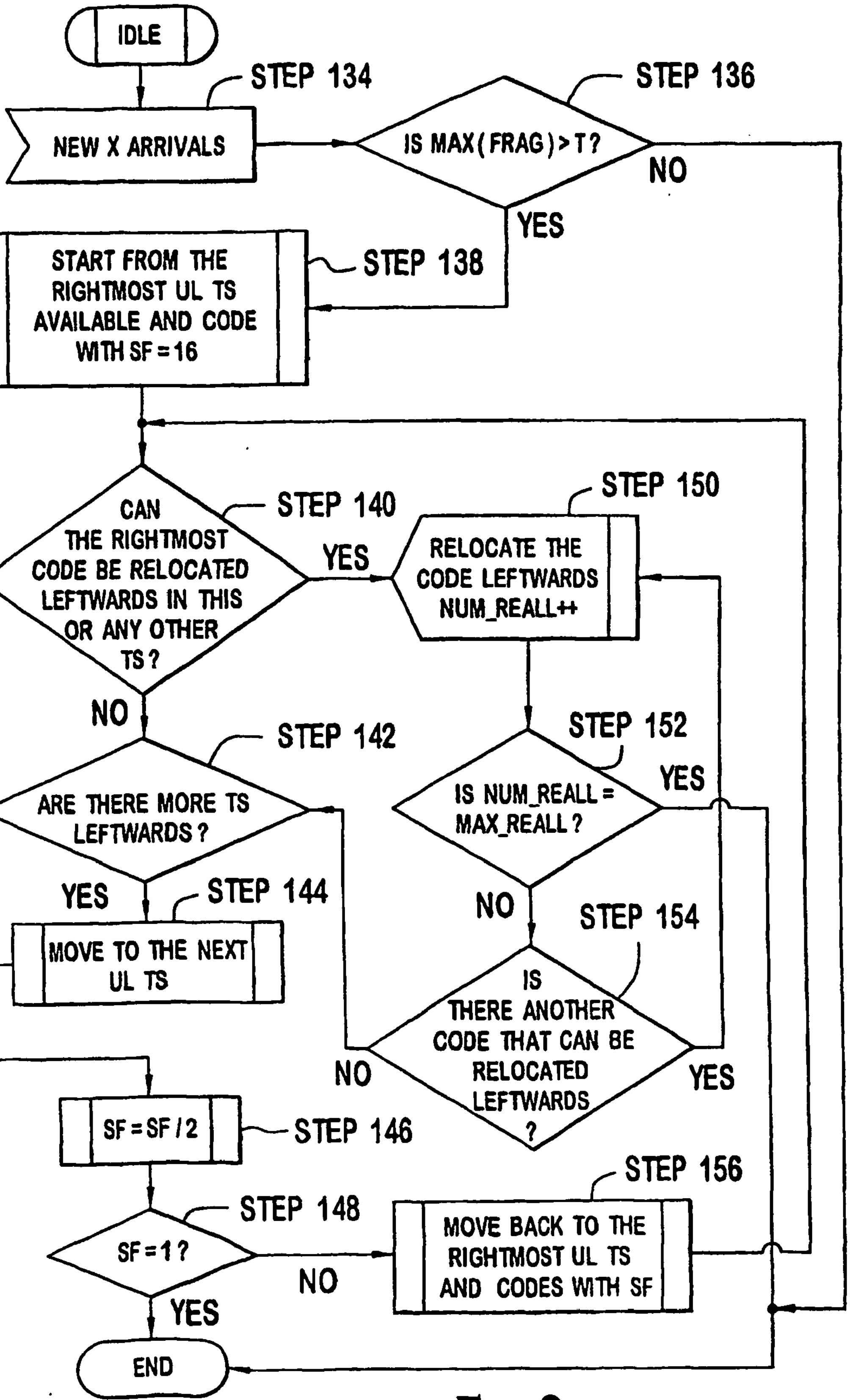

IDLE
STEP 134
NEW X ARRIVALS
STEP 136
IS MAX(FRAG)>T?
NO
YES
START FROM THE RIGHTMOST UL TS AVAILABLE AND CODE WITH SF=16
STEP 138
CAN THE RIGHTMOST CODE BE RELOCATED LEFTWARDS IN THIS OR ANY OTHER TS?
STEP 140
YES
STEP 150
RELOCATE THE CODE LEFTWARDS NUM_REALL++
NO
STEP 142
ARE THERE MORE TS LEFTWARDS?
NO
YES
STEP 144
MOVE TO THE NEXT UL TS
STEP 152
IS NUM_REALL = MAX_REALL?
YES
NO
STEP 154
IS THERE ANOTHER CODE THAT CAN BE RELOCATED LEFTWARDS?
NO
YES
SF=SF/2
STEP 146
STEP 148
SF=1?
NO
YES
STEP 156
MOVE BACK TO THE RIGHTMOST UL TS AND CODES WITH SF
END

FIG. 8

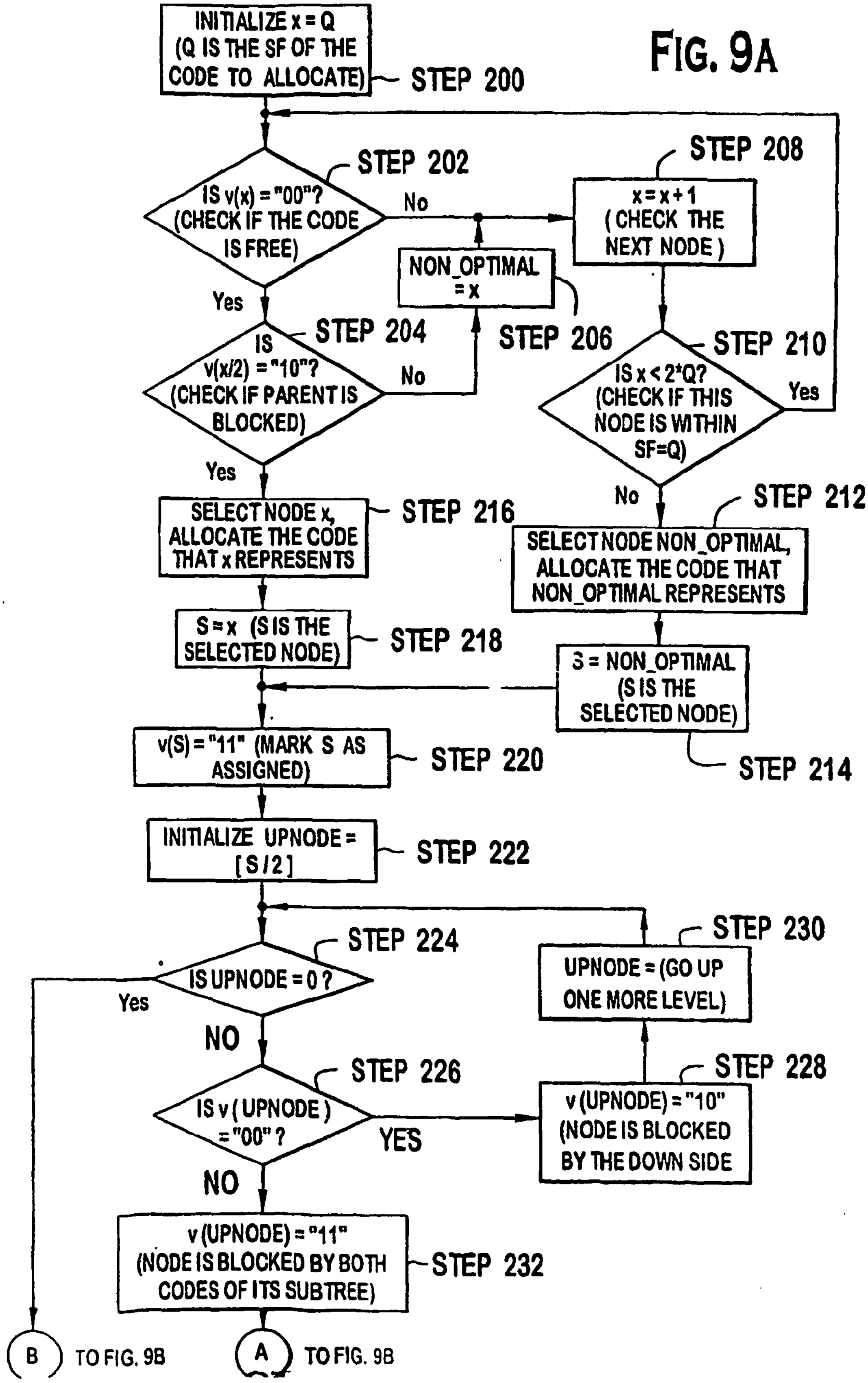
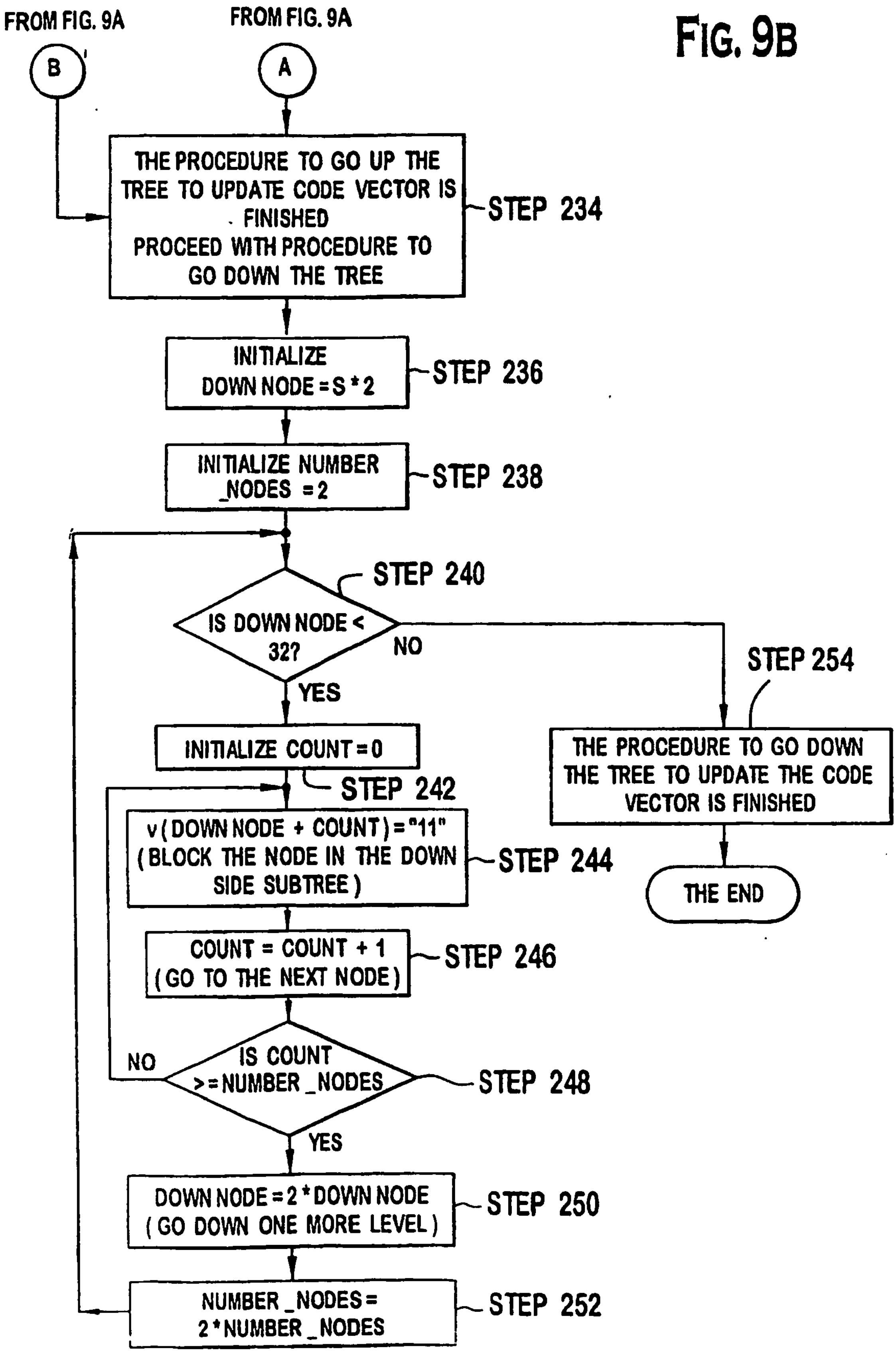

INITIALIZE x = Q (Q IS THE SF OF THE CODE TO ALLOCATE)
STEP 200
IS v(x) = "00"? (CHECK IF THE CODE IS FREE)
STEP 202
No
Yes
STEP 208
x = x + 1 (CHECK THE NEXT NODE)
NON_OPTIMAL = x
STEP 206
IS v(x/2) = "10"? (CHECK IF PARENT IS BLOCKED)
STEP 204
No
Yes
IS x < 2*Q? (CHECK IF THIS NODE IS WITHIN SF=Q)
STEP 210
Yes
No
SELECT NODE x, ALLOCATE THE CODE THAT x REPRESENTS
STEP 216
SELECT NODE NON_OPTIMAL, ALLOCATE THE CODE THAT NON_OPTIMAL REPRESENTS
STEP 212
S = x (S IS THE SELECTED NODE)
STEP 218
S = NON_OPTIMAL (S IS THE SELECTED NODE)
STEP 214
v(S) = "11" (MARK S AS ASSIGNED)
STEP 220
INITIALIZE UPNODE = [S/2]
STEP 222
IS UPNODE = 0 ?
STEP 224
Yes
NO
UPNODE = (GO UP ONE MORE LEVEL)
STEP 230
IS v (UPNODE) = "00" ?
STEP 226
YES
NO
v (UPNODE) = "10" (NODE IS BLOCKED BY THE DOWN SIDE
STEP 228
v (UPNODE) = "11" (NODE IS BLOCKED BY BOTH CODES OF ITS SUBTREE)
STEP 232
B
TO FIG. 9B
A
TO FIG. 9B

FIG. 9A

FIG. 9B


FROM FIG. 9A
B
FROM FIG. 9A
A
THE PROCEDURE TO GO UP THE TREE TO UPDATE CODE VECTOR IS FINISHED PROCEED WITH PROCEDURE TO GO DOWN THE TREE
STEP 234
INITIALIZE DOWN NODE = S * 2
STEP 236
INITIALIZE NUMBER _NODES = 2
STEP 238
STEP 240
IS DOWN NODE < 32?
NO
YES
STEP 254
INITIALIZE COUNT = 0
STEP 242
THE PROCEDURE TO GO DOWN THE TREE TO UPDATE THE CODE VECTOR IS FINISHED
v (DOWN NODE + COUNT) = "11" (BLOCK THE NODE IN THE DOWN SIDE SUBTREE)
STEP 244
THE END
COUNT = COUNT + 1 (GO TO THE NEXT NODE)
STEP 246
NO
IS COUNT >= NUMBER _NODES
STEP 248
YES
DOWN NODE = 2 * DOWN NODE (GO DOWN ONE MORE LEVEL)
STEP 250
NUMBER _NODES = 2 * NUMBER _NODES
STEP 252

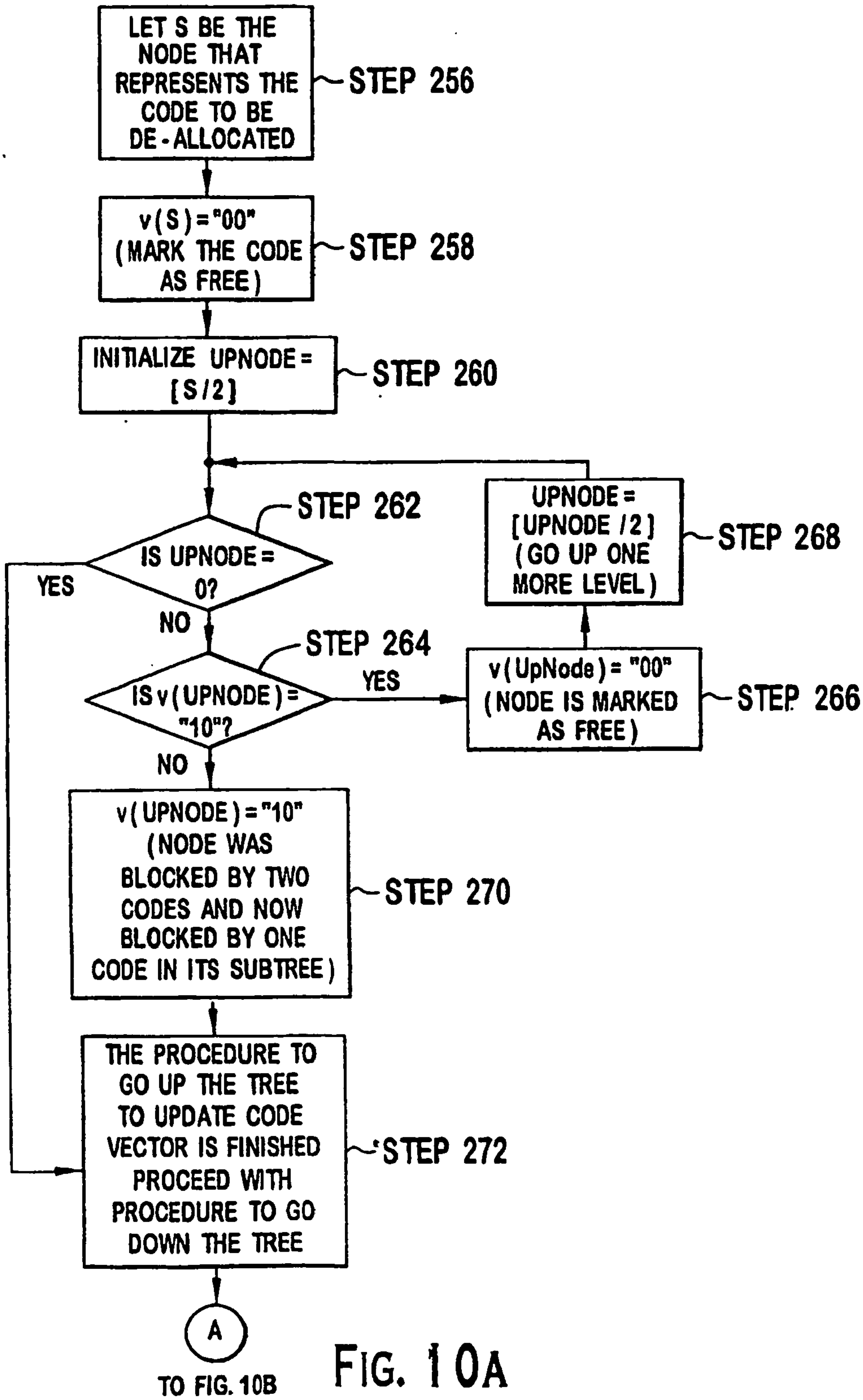

LET S BE THE NODE THAT REPRESENTS THE CODE TO BE DE-ALLOCATED
STEP 256
v(S)="00" (MARK THE CODE AS FREE)
STEP 258
INITIALIZE UPNODE= [S/2]
STEP 260
STEP 262
IS UPNODE= 0?
YES
NO
STEP 264
IS v(UPNODE)= "10"?
YES
NO
UPNODE= [UPNODE /2] (GO UP ONE MORE LEVEL)
STEP 268
v(UpNode)= "00" (NODE IS MARKED AS FREE)
STEP 266
v(UPNODE)="10" (NODE WAS BLOCKED BY TWO CODES AND NOW BLOCKED BY ONE CODE IN ITS SUBTREE)
STEP 270
THE PROCEDURE TO GO UP THE TREE TO UPDATE CODE VECTOR IS FINISHED PROCEED WITH PROCEDURE TO GO DOWN THE TREE
STEP 272
A
TO FIG. 10B

FIG. 10A

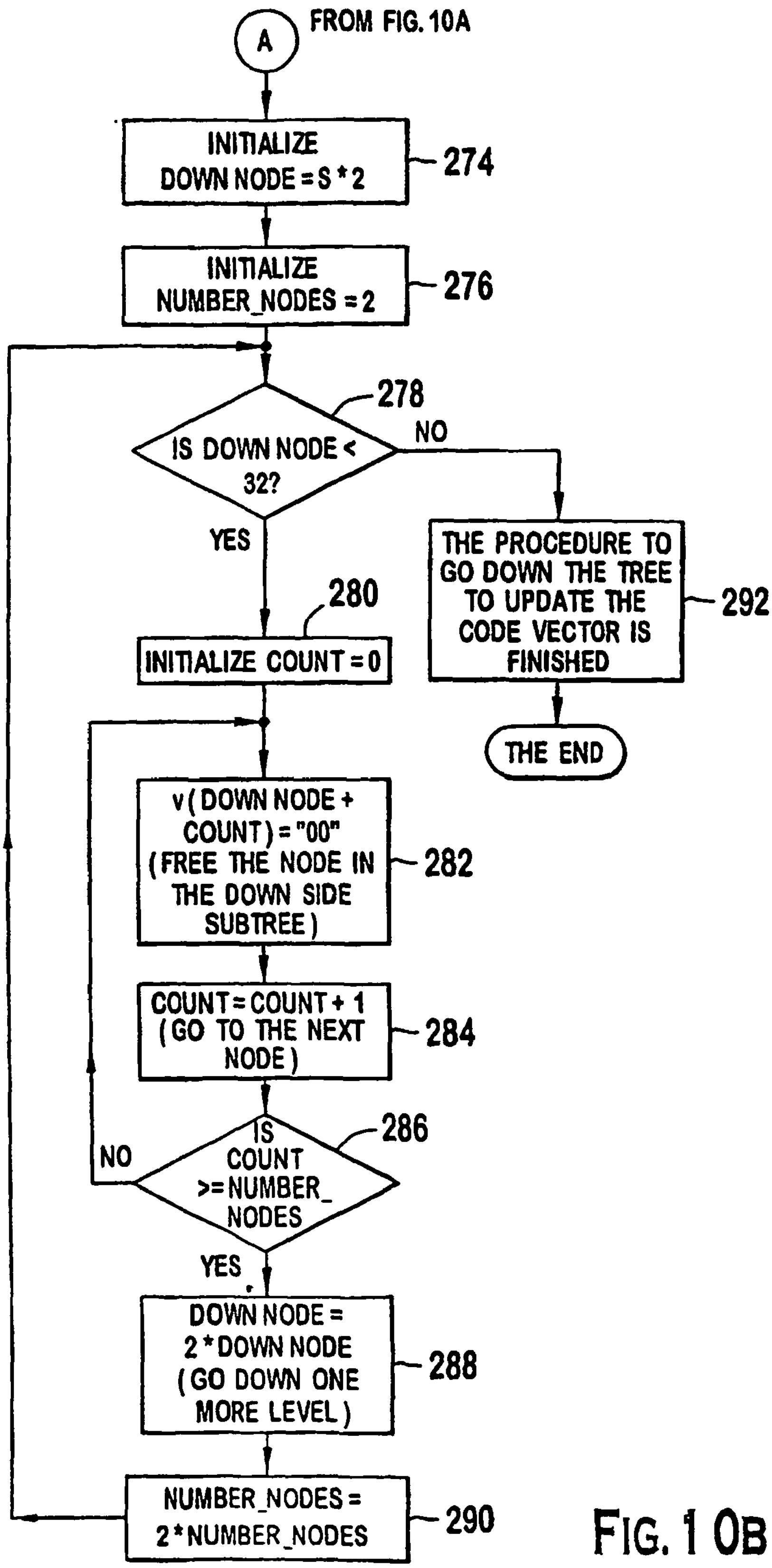
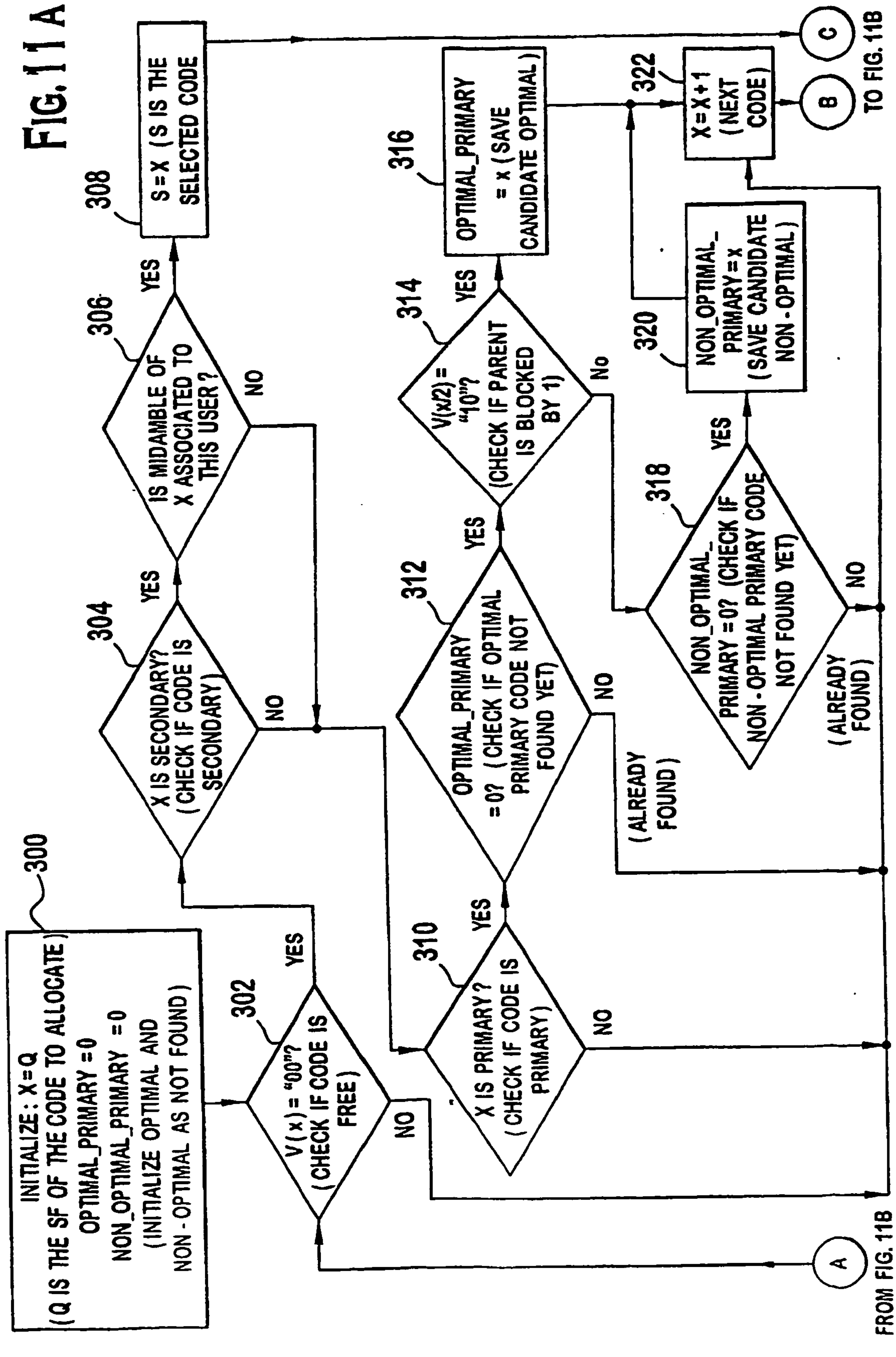

A
FROM FIG. 10A
INITIALIZE DOWN NODE = S * 2
274
INITIALIZE NUMBER_NODES = 2
276
278
IS DOWN NODE < 32?
NO
YES
THE PROCEDURE TO GO DOWN THE TREE TO UPDATE THE CODE VECTOR IS FINISHED
292
THE END
280
INITIALIZE COUNT = 0
v (DOWN NODE + COUNT) = "00" (FREE THE NODE IN THE DOWN SIDE SUBTREE)
282
COUNT = COUNT + 1 (GO TO THE NEXT NODE)
284
286
IS COUNT >= NUMBER_NODES
NO
YES
DOWN NODE = 2 * DOWN NODE (GO DOWN ONE MORE LEVEL)
288
NUMBER_NODES = 2 * NUMBER_NODES
290

FIG. 10B

FIG. 11 A



300
INITIALIZE: X=Q
(Q IS THE SF OF THE CODE TO ALLOCATE)
OPTIMAL_PRIMARY =0
NON_OPTIMAL_PRIMARY =0
(INITIALIZE OPTIMAL AND
NON-OPTIMAL AS NOT FOUND)
302
V(x) = "00"? (CHECK IF CODE IS FREE)
YES
NO
304
X IS SECONDARY? (CHECK IF CODE IS SECONDARY)
YES
NO
306
IS MIDAMBLE OF X ASSOCIATED TO THIS USER?
YES
NO
308
S=X (S IS THE SELECTED CODE
310
X IS PRIMARY? (CHECK IF CODE IS PRIMARY)
YES
NO
312
OPTIMAL_PRIMARY =0? (CHECK IF OPTIMAL PRIMARY CODE NOT FOUND YET)
YES
NO
(ALREADY FOUND)
314
V(x/2) = "10"? (CHECK IF PARENT IS BLOCKED BY 1)
YES
No
316
OPTIMAL_PRIMARY = x (SAVE CANDIDATE OPTIMAL)
318
NON_OPTIMAL_ PRIMARY =0? (CHECK IF NON-OPTIMAL PRIMARY CODE NOT FOUND YET)
YES
NO
(ALREADY FOUND)
320
NON_OPTIMAL_ PRIMARY=x (SAVE CANDIDATE NON-OPTIMAL)
322
X=X+1 (NEXT CODE)
A
FROM FIG. 11B
B
C
TO FIG. 11B

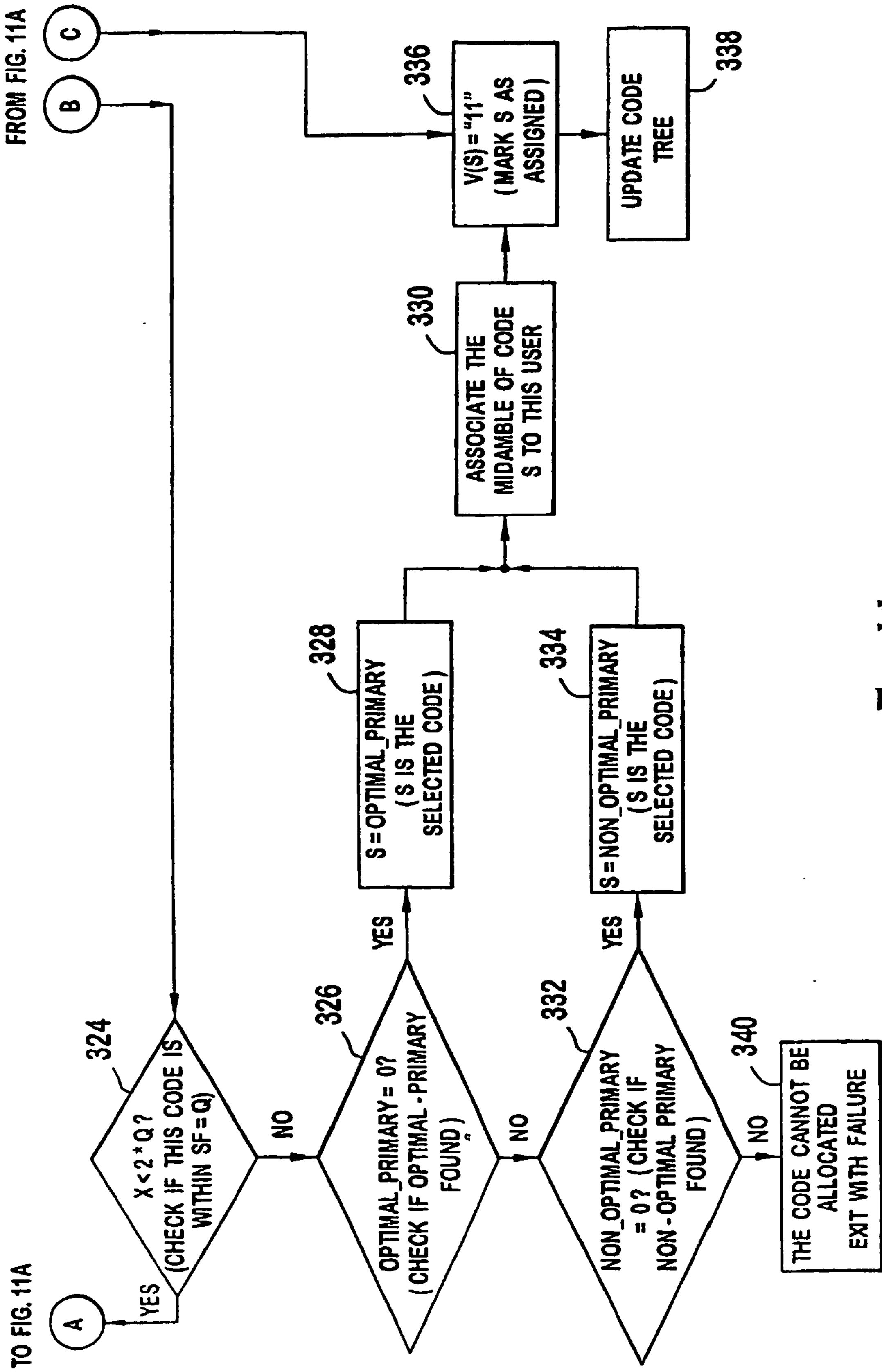

FROM FIG. 11A
B
C
TO FIG. 11A
A
324
X < 2 * Q ? (CHECK IF THIS CODE IS WITHIN SF = Q)
YES
NO
326
OPTIMAL_PRIMARY = 0? (CHECK IF OPTIMAL - PRIMARY FOUND)
YES
NO
328
S = OPTIMAL_PRIMARY (S IS THE SELECTED CODE)
332
NON_OPTIMAL_PRIMARY = 0? (CHECK IF NON - OPTIMAL PRIMARY FOUND)
YES
NO
334
S = NON_OPTIMAL_PRIMARY (S IS THE SELECTED CODE)
340
THE CODE CANNOT BE ALLOCATED EXIT WITH FAILURE
330
ASSOCIATE THE MIDAMBLE OF CODE S TO THIS USER
336
V(S) = "11" (MARK S AS ASSIGNED)
338
UPDATE CODE TREE

FIG. 11 B

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 2002051437 A **[0010]**
- EP 1035676 A **[0010]**

Non-patent literature cited in the description

- **TSENG Y-C.** *Code Placement and Replacement Strategies for Wideband CDMA OVSF Code Tree Management,* 25 November 2001, 562-566 **[0010]**
- **MINN T.** Dynamic Assignment Of Orthogonal Variable-Spreading-Factor Codes in W-CDMA. *IEEE Journal on Selected Areas in Communications,* August 2000, vol. 18 (8), 1429-1440 **[0010]**