

①②

DEMANDE DE BREVET D'INVENTION

A1

②② Date de dépôt : 09.07.02.

③③ Priorité : 09.07.01 KR 00144673; 25.08.01 KR 00151605; 29.08.01 KR 00152596.

④③ Date de mise à la disposition du public de la demande : 14.03.03 Bulletin 03/11.

⑤⑥ Liste des documents cités dans le rapport de recherche préliminaire : *Ce dernier n'a pas été établi à la date de publication de la demande.*

⑥① Références à d'autres documents nationaux apparentés :

⑦① Demandeur(s) : SAMSUNG ELECTRONICS CO LTD — KR.

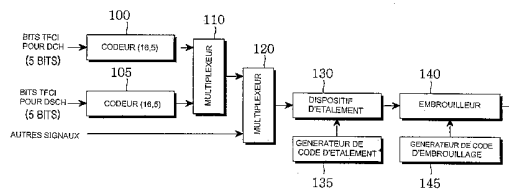
⑦② Inventeur(s) : HWANG SUNG OH, LEE KOOK HEUI, KIM JAE YOEL et PARK SANG HWAN.

⑦③ Titulaire(s) :

⑦④ Mandataire(s) : CABINET BONNET THIRION.

⑤④ PROCÉDE ET APPAREIL POUR EFFECTUER UNE APPLICATION DE SYMBOLES CODÉS VERS UNE TRAME RADIO DANS UN SYSTÈME DE COMMUNICATION MOBILE AMRC.

⑤⑦ L'invention décrit un procédé pour effectuer une application de premiers et seconds symboles TFCl codés vers une trame radio dans un appareil d'émission d'un système de communication mobile, pour coder k premiers bits TFCl et (10-k) seconds bits TFCl, la somme des premiers et seconds symboles TFCl codés étant égale à 32. Le procédé comprend le multiplexage (110) des symboles codés de façon à répartir uniformément les premiers et seconds symboles TFCl codés, conformément à un mode de transmission et un débit de données de la trame radio, en fournissant en sortie 32 symboles codés; et l'application des 32 symboles codés multiplexés vers la trame radio pour respecter le nombre de symboles codés pouvant être appliqués sur une trame radio, déterminé conformément au mode de transmission et au débit de données de la trame radio.



La présente invention concerne de façon générale un appareil et un procédé de transmission pour un mode de division ferme dans un système de communication mobile AMRT, et elle concerne en particulier
5 un appareil et un procédé d'application pour transmettre des bits d'Indicateur de Combinaison de Format de Transport (ou TFCI pour "Transport Format Combinaison Indicator").

10 De façon générale, un canal partagé de liaison descendante (DSCH pour "downlink shared channel") est partagé par une pluralité d'utilisateurs sur la base d'une répartition dans le temps. Le canal DSCH est établi en association avec un canal dédié (DCH pour
15 "dedicated channel") pour chaque utilisateur. Le canal DCH est transmis sur un canal physique dédié (DPCH pour "dedicated physical channel"), et le DPCH est construit en combinant un canal de commande physique dédié (DPCCH pour "dedicated physical
20 control channel") et un canal de données physique dédié (DPDCH pour "dedicated physical data channel"), sur la base d'un répartition dans le temps.

Le DSCH est transmis sur un canal partagé de liaison descendante physique (PDSCH pour "physical
25 downlink shared channel"), et l'information de commande de canal pour le PDSCH est transmise sur le DPCCH dans le DPCH. L'information de commande transmise sur le DPCCH comprend une information sur (i) l'ordre de Commande de Puissance Emise (TPC pour
30 "Transmitted Power Control") pour commander la puissance d'émission de liaison montante à partir d'un équipement d'utilisateur (UE pour "User Equipment"), (ii) un champ pilote utilisé pour l'estimation de variations de canal, la mesure de
35 puissance d'émission et l'acquisition de synchronisation de

créneau à partir d'un Noeud B vers un UE, et (iii) l'Indicateur de Combinaison de Format de Transport (TFCI pour "Transport Format Combinaison Indicator"). Parmi cette information, la TPC et le Pilote sont utilisés à titre
 5 d'information de commande physique pour le PDSCH et le DPCH, et le TFCI est utilisé pour indiquer des caractéristiques d'information (par exemple le débit de transfert d'information et la combinaison de différentes informations, par exemple la combinaison d'information vocale et
 10 d'information de paquets) des données qui sont transmises sur le DSCH et le DPDCH.

Comme indiqué ci-dessus, le TFCI, c'est-à-dire l'information de commande indiquant des caractéristiques d'information des données transmises sur les canaux physiques
 15 DSCH et DPDCH, a une longueur de 10 bits et est codé en 32 bits. Ainsi, l'information sur une quantité de données est exprimée avec 10 bits, et l'information à 10 bits est codée en 32 bits pour être transmise sur le canal physique.

Le TFCI est transmis sur le canal physique selon le
 20 procédé suivant, spécifié dans le document 3GPP (3rd Generation Partnership Project) Technical Specification 25.212 for the UMTS (Universal Mobile Telecommunication System).

a_k = k-ième bit d'information de l'information de combinaison de transport ($0 \leq k \leq 9$)

25 b_l = l-ième bit codé d'information de combinaison de transport ($0 \leq l \leq 31$)

d_m = m-ième bit codé transmis d'information de combinaison de transport.

L'information a_k est l'information à 10 bits indiquant le débit, le type et la combinaison des données transmises sur le DPDCH, l'information b_l est constituée de 32 bits codés obtenus en codant l'information a_k , et l'information d_m est un bit codé transmis lorsque l'information b_l est transmise sur le DPCH. Ici, la valeur m est variable
 35 conformément à des conditions.

Des conditions pour déterminer le nombre de bits d_m sont déterminées sur la base d'un mode de transmission du DPCCH et d'un débit de données du DPCH. Le mode de transmission du DPCH comprend un mode de transmission normal et un mode de transmission compressé. Le mode de transmission compressé est utilisé lorsqu'un UE ayant un émetteur-récepteur RF désire effectuer une mesure dans une autre bande de fréquence. Une opération dans le mode de transmission compressé suspend temporairement la transmission dans la bande de fréquence présente, pour permettre à l'UE d'effectuer une mesure dans une autre bande de fréquence. Des données à transmettre au cours de la période de suspension de transmission sont compressées immédiatement avant et après la période de suspension de transmission.

Le "débit de données du DPCH", qui est l'une des conditions pour déterminer le nombre de bits d_m , fait référence à un débit de données physique du DPCH et est déterminé conformément à un facteur d'étalement (SF pour "spreading factor") des données. Le SF va de 4 à 512 et le débit de données va de 15 kbit/s à 1920 kbit/s. Lorsque le SF devient plus élevé, le débit de données devient inférieur. La raison pour laquelle le nombre de bits d_m est déterminé conformément au débit de données du DPCH consiste en ce que la taille (ou la longueur) du champ de TFCI transmettant les bits TFCI du DPCCH est variable conformément au débit de données du DPCH.

Le nombre de bits d_m transmis pour chacune des conditions pour déterminer d_m est calculé de la façon suivante.

A1. Mode de transmission normal, le débit de données du DPCH étant inférieur à 60 kbit/s

Dans le cas d'une condition A1 pour déterminer le nombre de bits d_m , le nombre de bits d_m devient de 30. Dans la norme 3GPP, une unité de transmission de base du canal physique est une trame radio. La trame radio a une longueur

de 10 ms et est constituée de 15 créneaux temporels. Chaque créneau temporel a des champs pour transmettre le TFCI. Dans le cas de A1, chaque créneau temporel a 2 champs de transmission de TFCI, ce qui fait que le nombre de bits de code de transmission de TFCI, d_m , qui peuvent être transmis pour une trame radio devient 30. Par conséquent, bien que le nombre de bits codés b_1 basé sur le bit d'information a_k devienne 32, les deux derniers bits d'information de combinaison de transport d_{30} et d_{31} ne sont pas transmis à cause d'une limitation du nombre des champs de TFCI qui sont réellement transmis.

A2. Mode de transmission normal, le débit de données de DPCH étant supérieur à 60 kbit/s

Dans le cas d'une condition A2 pour déterminer le nombre de bits d_m , une longueur du champ de TFCI dans le créneau temporel devient 8 bits, et le nombre total de bits d_m qui peuvent être transmis sur le DPCH pour une trame radio devient 120. Lorsque le nombre total de d_m est 120, l'information b_1 est transmise de façon répétée, de la manière suivante.

$$d_0(b_0), \dots, d_{31}(b_{31}), d_{32}(b_0), \dots, d_{63}(b_{31}), \dots, d_{96}(b_0), \dots, d_{119}(b_{23})$$

Dans le cas de A2, les 0-ième à 23-ième bits b_1 sont répétés 4 fois, et les 24-ième à 31-ième bits b_1 sont répétés 3 fois pour la transmission.

A3. Mode de transmission compressé, le débit de données de DPCH étant inférieur à 60 kbit/s ou égal à 120 kbit/s

Dans le cas d'une condition A3 pour déterminer le nombre de bits d_m , une longueur du champ de TFCI dans le créneau temporel devient 4 bits, et le nombre de TFCI qui peuvent être transmis pour une trame radio est variable conformément au nombre de créneaux temporels utilisés dans

le mode de transmission compressé. Dans le mode de transmission compressé, le nombre de créneaux temporels de suspension de transmission va d'un minimum de 1 à un maximum de 7, et le nombre de bits d_m est compris entre 32 et 56. Le nombre de bits codés transmis d_m est limité à un maximum de 32, pour ainsi transmettre la totalité des 0-ième à 31-ième bits b_1 au d_m changé, et ne pas transmettre les bits b_1 à l'autre d_m .

10 A4. Mode de transmission compressé, le débit de données du DPCH étant supérieur à 120 kbit/s ou égal à 60 kbit/s

Dans le cas d'une condition A4 pour déterminer le nombre de bits d_m , une longueur du champ de TFCI dans le créneau temporel devient 16 bits, et le nombre de TFCI qui peuvent être transmis pour une trame radio est variable conformément au nombre de créneaux temporels utilisés dans le mode de transmission compressé. Dans le mode de transmission compressé, le nombre de créneaux temporels de suspension de transmission va d'un minimum de 1 à un maximum de 7, et le nombre de bits d_m va de 128 à 244. Le nombre de bits codés d_m transmis est limité à un maximum de 128, pour ainsi transmettre de façon répétée, 4 fois, les 0-ième à 31-ième bits b_1 au d_m changé, et ne pas transmettre les bits b_1 à l'autre d_m .

25 Dans le mode de transmission compressé de A3 et A4, les bits d_m sont disposés dans une période qui est aussi éloignée que possible que la période de suspension de transmission, afin de maximiser la fiabilité de transmission des bits d_m .

30 Les conditions A1, A2, A3 et A4 sont utilisées lorsque le TFCI indique la combinaison de transport et le type du DPCH. Un procédé de division du TFCI en TFCI pour DSCH et en TFCI pour DPCH, pendant la transmission, peut être divisé en deux procédés séparés.

35 Un premier procédé est un procédé pour un mode de

division ferme (HSM pour "hard split mode"), et un second procédé est un procédé pour un mode de division logique (LCM pour "logical split mode").

Le TFCI pour DCH sera appelé TFCI(champ 1) ou un premier TFCI, et le TFCI pour DSCH sera appelé TFCI(champ 2) ou un second TFCI.

Dans le procédé LSM, le TFCI(champ 1) et le TFCI(champ 2), sous la forme d'un seul TFCI, sont codés avec un sous-code (32,10) du code de Reed-Muller du second ordre. Le TFCI(champ 1) et le TFCI(champ 2) expriment une information de TFCI à 10 bits avec divers rapports, et les 10 bits d'information sont codés avec un code à un bloc, c'est-à-dire un sous-code (32,10) du code de Reed-Muller du second ordre conformément aux conditions A1, A2, A3 et A4, avant d'être transmis. Les rapports du TFCI(champ 1) au TFCI(champ 2) comprennent 1:9, 2:8 3:7, 4:6, 5:5, 6:4, 7:3, 8:2 et 9:1. La somme des premiers bits d'information TFCI et des seconds bits d'information TFCI peut être inférieure à 10. Dans le LSM, si la somme des premiers bits d'information TFCI et des seconds bits d'information TFCI est inférieure à 10, des 0 sont insérés en un nombre égal au nombre de bits manquants. Il en résulte que les premiers bits d'information TFCI et les seconds bits d'information TFCI peuvent être codés avec un code de Reed-Muller (32,10) avant d'être transmis.

Dans le procédé HSM, le TFCI(champ 1) et le TFCI(champ 2) sont exprimés de manière fixe avec respectivement 5 bits, et chaque information est émise en utilisant un code bi-orthogonal (16,5), et ensuite les 16 bits pour le TFCI(champ 1) et le TFCI(champ 2) sont transmis en alternance conformément aux conditions A1, A2, A3 et A4. Dans le cas où le nombre maximal des premiers bits d'information TFCI et le nombre maximal des seconds bits d'information TFCI sont tous deux limités à 5, si le nombre des premiers bits d'information TFCI ou des seconds bits d'information TFCI dépasse 5, il n'est pas possible d'utiliser le procédé

HSM. Par conséquent, si le nombre des premiers bits d'information TFCI ou des seconds bits d'information TFCI est inférieur à 5, des 0 en un nombre égal au nombre de bits vides sont insérés avant d'être codés en utilisant un code bi-orthogonal (16,5).

La figure 1 illustre une structure d'un émetteur basé sur le procédé HSM classique. En se référant à la figure 1, on note qu'un codeur bi-orthogonal (16,5) 100 code un TFCI(champ 1) à 5 bits pour le DCH en 16 symboles codés, et fournit les 16 symboles codés à un multiplexeur 110. En même temps, un codeur bi-orthogonal (16,5) 105 code un TFCI(champ 2) à 5 bits pour le DSCH en 16 symboles codés, et fournit les 16 symboles codés au multiplexeur 110. Le multiplexeur 110 multiplexe dans le temps les 16 symboles codés provenant du codeur 100 et les 16 symboles codés provenant du codeur 105, et il fournit en sortie 32 symboles après combinaison. Un multiplexeur 120 multiplexe dans le temps les 32 symboles émis par le multiplexeur 110 et d'autres signaux et il fournit son information de sortie à un dispositif d'étalement 130. Le dispositif d'étalement 130 étale le signal de sortie du multiplexeur 120 avec un code d'étalement qui est fourni par un générateur de code d'étalement 135. Un embrouilleur 140 embrouille le signal étalé avec un code d'embrouillage qui est fourni par un générateur de code d'embrouillage 145.

Si un UE est placé dans une région de transfert souple, le procédé LSM est soumis à de nombreuses restrictions, pour les raisons suivantes. Pour la commodité de l'explication, on donnera une brève description d'un réseau de transmission sans fil 3GPP. Un Réseau d'Accès Radio (RAN pour "Radio Access Network") comprend une Unité de Commande de Réseau Radio (RNC pour "Radio Network Controller"), un Noeud B commandé par la RNC, et un Equipement d'Utilisateur (UE pour "User Equipment"). La RNC commande le Noeud B, le Noeud B remplit la fonction d'une station de base, et l'UE remplit la fonction d'un terminal. La RNC peut être divisée en une

Unité de Commande de Réseau Radio de Desserte (SRNC pour "Serving Radio Network Controller") et une Unité de Commande de Réseau Radio de Commande (CRNC pour "Control Radio Network Controller"), conformément aux relations avec l'UE. La SRNC, qui est une RNC auprès de laquelle l'UE est enregistré, traite des données à émettre vers l'UE et à recevoir à partir de celui-ci, et commande l'UE. La CRNC, qui est une RNC à laquelle l'UE est connecté au moment présent, connecte l'UE à la SRNC.

10 Lorsque des Noeuds B en communication avec l'UE appartiennent à différentes RNC, les Noeuds B qui n'émettent pas sur le canal DSCH ne peuvent pas reconnaître une valeur des bits TFCI codés pour le DSCH, ce qui fait qu'il n'est pas possible d'émettre correctement des bits TFCI codés vers
15 l'UE.

Dans le HSM envisagé ci-dessus, les bits d'information TFCI pour le DSCH et les bits d'information TFCI pour le DCH sont codés indépendamment, ce qui fait que l'UE n'a aucune difficulté pour décoder des bits TFCI reçus. Cependant, dans le HSM du type 3GPP actuel, le nombre de bits TFCI pour le DCH et le nombre de bits TFCI pour le DSCH sont tous deux fixés à 5 bits pour exprimer 32 bits d'information. Par conséquent, lorsque davantage de bits TFCI pour le DCH ou le DSCH sont nécessaires, le HSM ne peut pas être
25 utilisé.

Un but de la présente invention est donc de procurer un appareil et un procédé pour émettre/recevoir des bits TFCI dans un système de communication mobile AMRC.

Un autre but de la présente invention est de procurer un appareil et un procédé pour effectuer une application de symboles TFCI codés vers un canal physique dans un système de communication mobile AMRC.

Un autre but supplémentaire de la présente invention est de procurer un appareil et un procédé pour effectuer une application de symboles TFCI codés pour le DCH et de symbo-

les TFCI codés pour le DSCH, séparés selon un rapport spécifique, vers un canal physique dans un système de communication mobile AMRC.

5 Un autre but encore de la présente invention est de procurer un appareil et un procédé pour recevoir des symboles TFCI codés ayant fait l'objet d'une application vers un canal physique avant d'être transmis dans un système de communication mobile AMRC.

10 Encore un autre but de la présente invention est de procurer un appareil et un procédé pour recevoir des symboles TFCI codés pour le DCH et des symboles TFCI codés pour le DSCH, séparés selon un rapport spécifique, ayant fait l'objet d'une application vers un canal physique avant d'être transmis dans un système de communication mobile AMRC.

15 Un premier aspect de la présente invention procure un procédé pour effectuer une application de premiers symboles TFCI codés et de seconds symboles TFCI codés vers une trame radio dans un appareil d'émission d'un système de communication mobile, pour coder k premiers bits TFCI et $(10-k)$ seconds bits TFCI, une somme des premiers symboles TFCI codés et des seconds symboles TFCI codés étant égale à 32. Le procédé comprend le multiplexage des symboles codés de façon que les premiers symboles TFCI codés et les seconds symboles TFCI codés soient uniformément répartis conformément à un mode de transmission et un débit de données de la trame radio, et la fourniture en sortie de 32 symboles codés; et l'application vers la trame radio des 32 symboles codés multiplexés pour respecter le nombre de symboles codés qui peuvent être appliqués vers une trame radio, ce nombre étant déterminé conformément au mode de transmission et au débit de données de la trame radio.

25 Un second aspect de la présente invention procure un appareil pour transmettre des premiers bits TFCI et des seconds bits TFCI sur une trame radio dans un appareil d'émission d'un système de communication mobile. L'appareil com-
35

prend au moins un codeur pour coder k premiers bits TFCI avec un premier rendement de codage pour fournir en sortie $(3k+1)$ premiers symboles TFCI codés, et pour coder $(10-k)$ seconds bits TFCI avec un second rendement de codage, pour
5 fournir en sortie $(31-3k)$ seconds symboles TFCI codés; et un dispositif d'arrangement de symboles codés pour multiplexer les symboles codés de façon que les premiers symboles TFCI codés et les seconds symboles TFCI codés soient uniformément répartis conformément à un mode de transmission et un débit
10 de données de la trame radio, et pour fournir en sortie les symboles codés multiplexés conformément au nombre de symboles codés qui peuvent être transmis sur une trame radio.

Selon un troisième aspect, la présente invention procure un procédé pour transmettre des premiers bits TFCI
15 et des seconds bits TFCI sur une trame radio dans un appareil d'émission d'un système de communication mobile. Le procédé comprend le codage de k premiers bits TFCI avec un premier rendement de codage, pour fournir en sortie $(3k+1)$ premiers symboles TFCI codés; le codage de $(10-k)$ seconds
20 bits TFCI avec un second rendement de codage pour fournir en sortie $(31-3k)$ seconds symboles TFCI codés; l'utilisation d'un dispositif d'arrangement de symboles codés pour multiplexer les symboles codés de façon que les premiers symboles TFCI codés et les seconds symboles TFCI codés soient unifor-
25 mément répartis conformément à un mode de transmission et un débit de données d'une trame radio; et la présentation en sortie des symboles codés multiplexés conformément au nombre de symboles codés qui peuvent être transmis sur une trame radio.

30 Selon un quatrième aspect, la présente invention procure un appareil pour décoder k premiers bits TFCI et $(10-k)$ seconds bits TFCI dans un appareil de réception d'un système de communication mobile pour recevoir $(3k-1)$ premiers symboles TFCI codés pour un DCH (Canal Dédié) et $(31-3k)$ seconds symboles TFCI codés pour un DSCH (Canal Partagé
35

de Liaison Descendante). L'appareil comprend un dispositif de réarrangement de symboles codés pour séparer les premiers symboles TFCI codés et les seconds symboles TFCI codés, transmis sur un DPCH (Canal Physique Dédié), conformément à une valeur de k , pour le réarrangement; et au moins un décodeur pour décoder les premiers symboles TFCI codés, pour fournir en sortie les k premiers bits TFCI, et pour décoder les seconds symboles TFCI codés pour fournir en sortie les $(10-k)$ seconds bits TFCI.

10 Selon un cinquième aspect, la présente invention procure un procédé pour décoder k premiers bits TFCI et $(10-k)$ seconds bits TFCI dans un appareil de réception d'un système de communication mobile, pour recevoir $(3k-1)$ premiers symboles TFCI codés pour un DCH (Canal Dédié) et $(31-3k)$ seconds symboles TFCI codés pour un DSCH (Canal Partagé de Liaison Descendante). Le procédé comprend la séparation des premiers symboles TFCI codés et des seconds symboles TFCI codés, transmis sur un DPCH (Canal Physique Dédié), conformément à une valeur de k , pour le réarrangement; et le décodage des premiers symboles TFCI codés pour fournir en sortie les k premiers bits TFCI, et le décodage des seconds symboles TFCI codés pour fournir en sortie les $(10-k)$ seconds bits TFCI.

25 Les buts, caractéristiques et avantages de la présente invention mentionnés ci-dessus, ainsi que d'autres, ressortiront davantage de la description détaillée qui va suivre, à lire en se référant conjointement aux dessins annexés dans lesquels :

30 La figure 1 montre une structure d'un émetteur classique basé sur un mode de division ferme (HSM);

 La figure 2 montre une structure d'un émetteur de Noeud B conforme à un mode de réalisation de la présente invention;

35 La figure 3 montre une autre structure d'un émetteur de Noeud B conforme à un mode de réalisation de la présente

invention;

La figure 4 montre une structure détaillée du codeur illustré sur les figures 2 et 3;

La figure 5 montre une structure d'une trame radio de liaison descendante transmise d'un Noeud B vers un UE;

La figure 6 montre une structure détaillée du dispositif d'arrangement de symboles illustré sur la figure 2;

La figure 7 montre une structure détaillée du sélecteur illustré sur la figure 3;

La figure 8 montre une autre structure détaillée du dispositif d'arrangement de symboles illustré sur la figure 3;

La figure 9 montre une structure d'un récepteur d'UE conforme à un mode de réalisation de la présente invention;

La figure 10 montre une autre structure d'un récepteur d'UE conforme à un autre mode de réalisation de la présente invention;

La figure 11 montre une structure détaillée du décodeur utilisé dans le récepteur de la figure 10;

La figure 12 montre un procédé de sélection de codes à utiliser pour un premier TFCI et un second TFCI conformément à un mode de réalisation de la présente invention;

La figure 13 montre une autre connexion entre des codeurs et un dispositif d'arrangement de symboles conformes à un mode de réalisation de la présente invention;

La figure 14 montre une autre connexion supplémentaire entre un codeur et un dispositif d'arrangement de symboles conformes à un mode de réalisation de la présente invention;

La figure 15 montre une autre connexion supplémentaire entre un codeur et un dispositif d'arrangement de symboles conformes à un mode de réalisation de la présente invention;

La figure 16 montre une opération de codage conforme à un mode de réalisation de la présente invention;

La figure 17 montre une opération de décodage conforme à un mode de réalisation de la présente invention;

Les figures 18A et 18B montrent deux structures différentes d'un dispositif d'arrangement de symboles conformes
5 à un mode de réalisation de la présente invention; et

La figure 19 montre une structure d'un dispositif d'arrangement de symboles codés conforme à un mode de réalisation de la présente invention.

On décrira ci-dessous un mode de réalisation préféré
10 de la présente invention, en se référant aux dessins annexés. Dans la description suivante, des fonctions ou des structures bien connues ne sont pas décrites en détail, du fait qu'elles obscurciraient l'invention par des détails inutiles.

15 La présente invention procure un appareil et un procédé pour diviser un total de 10 bits d'information d'entrée en bits d'information pour un DCH et en bits d'information pour un DSCH dans un rapport de 1:9, 2:8 3:7, 4:6, 5:5, 6:4, 7:3, 8:2 et 9:1 dans le procédé HSM, et pour ensuite coder
20 séparément les bits d'information pour le DCH et les bits d'information pour le DSCH. Si la somme du nombre de premiers bits d'information TFCI et du nombre de seconds bits d'information TFCI est inférieure à 10, l'appareil et le procédé conformes à un mode de réalisation de la présente
25 invention augmentent la fiabilité des premiers bits d'information TFCI ou des seconds bits d'information TFCI avant le codage. Selon une variante, l'appareil et le procédé augmentent la fiabilité à la fois des premiers bits d'information TFCI et des seconds bits d'information TFCI avant le codage.

30 On présentera tout d'abord une description d'un codeur pour le cas dans lequel la somme des premiers bits d'information TFCI et des seconds bits d'information TFCI est égale à 10.

Une trame radio transmet 30, 120, 32 et 128 symboles
35 TFCI codés, conformément respectivement aux conditions A1,

A2, A3 et A4. Dans chaque cas à l'exclusion de la transmission répétée, un rendement de codage de base est $10/32$, et dans la condition A1, un rendement de codage devient $10/30$ du fait de la transmission limitée du canal physique. Par conséquent, lorsque les bits d'information TFCI pour le DSCH et les bits d'information TFCI pour le DCH sont divisés selon un rapport spécifique de 1:9, 2:8 3:7, 4:6, 5:5, 6:4, 7:3, 8:2 et 9:1, il est naturel de maintenir le rendement de codage en divisant les symboles codés selon les rapports ci-dessus. Le maintien du rendement de codage signifie le maintien d'un rendement de codage de base de $(32,10)$. Dans le HCM, la raison pour maintenir un gain de codage du TFCI pour le DSCH et du TFCI pour le DCH, qui sont codés différemment, est de maintenir un gain de codage en maintenant de façon similaire le rendement de codage de $(32,10)$, bien que le TFCI pour le DSCH et le TFCI pour le DCH soient codés séparément. On décrira un exemple de division des bits codés conformément au rapport des bits d'entrée, en faisant l'hypothèse de la condition A1.

Dans la condition A1, si 10 bits d'information d'entrée sont divisés dans un rapport 1:9, alors 30 symboles de sortie codés sont divisés dans un rapport 3:27, et si les 10 bits d'information d'entrée sont divisés en 2:8, alors les 30 symboles de sortie codés sont divisés dans un rapport de 6:24. En outre, si les 10 bits d'information d'entrée sont divisés dans un rapport de 3:7, alors les 30 symboles de sortie codés sont divisés dans un rapport de 9:21, et si les 10 bits d'information d'entrée sont divisés dans un rapport de 4:6, alors les 30 symboles de sortie codés sont divisés dans un rapport de 12:18. Cependant, dans les conditions A2, A3 et A4, les 32 symboles codés sont tous transmis, ou les 32 symboles codés sont transmis de façon répétée, ce qui fait que les symboles codés ne peuvent pas être divisés correctement comme dans la condition A1.

Par conséquent, dans le mode de réalisation de la

présente invention, les rendements de codage des symboles codés définis en association avec les bits d'entrée peuvent être exprimés comme le montre le Tableau 1.

Tableau 1

Rapport de bits d'entrée	Rapport de symboles codés	Rendement de codage utilisé	
		Rendement de codage du 1ier TFCI	Rendement de codage du 2ième TFCI
1:9	3:29	(3:1)	(29:9)
	4:28	(4:1)	(28:9)
	5:27	(5:1)	(27:9)
2:8	6:26	(6:2)	(26:8)
	7:25	(7:2)	(25:8)
	8:24	(8:2)	(24:8)
3:7	9:23	(9:3)	(23:7)
	10:22	(10:3)	(22:7)
	11:21	(11:3)	(21:7)
4:6	12:20	(12:4)	(20:6)
	13:19	(13:4)	(19:6)
	14:18	(14:4)	(18:6)
6:4	18:14	(18:6)	(14:4)
	19:13	(19:6)	(13:4)
	20:12	(20:6)	(12:4)
7,3	21:11	(21:7)	(11:3)
	22:10	(22:7)	(10:3)
	23:9	(23:7)	(9:3)
8:2	24:8	(24:8)	(8:2)
	25:7	(25:8)	(7:2)
	26:6	(26:8)	(6:2)
9:1)	27:5	(27:9)	(5:1)
	28:4	(28:9)	(4:1)
	29:3	(29:9)	(3:1)

rendements de codage dans le Tableau 1 conformément au rapport des bits d'entrée. Le mode de réalisation de la présente invention fixe la somme des symboles codés à 30 en appliquant la valeur exigée minimale au rendement de codage notable (30,10) pour le cas A1, le plus fréquemment utilisé, parmi les conditions A1, A2, A3 et A4, et en fixant le rendement de codage du premier TFCI et le rendement de codage du second TFCI à un minimum de $1/3$, et il alloue ensuite les 2 symboles codés restants respectivement aux symboles codés du premier TFCI et aux symboles codés du second TFCI. Par conséquent, le mode de réalisation de la présente invention augmente à la fois le rendement de codage du premier TFCI et le rendement du codage du second TFCI, ou augmente soit le rendement de codage du premier TFCI soit le rendement de codage du second TFCI, en utilisant les 2 symboles codés restants comme des symboles codés du premier TFCI ou des symboles codés du second TFCI. Le mode de réalisation augmente le rendement de codage du premier TFCI ou du second TFCI, parmi les critères pour déterminer les rendements de codage, lorsqu'il est nécessaire d'augmenter les performances en augmentant seulement le rendement de codage du premier TFCI ou le rendement de codage du second TFCI, dans une condition dans laquelle la somme du nombre de symboles codés pour le premier TFCI et du nombre de symboles codés pour le second TFCI doit devenir 32.

Une fois qu'un rapport des bits d'entrée dans le Tableau 1 est déterminé, l'un de 3 procédés de codage est utilisé conformément au rapport des symboles codés.

La présente invention procure un codeur capable d'effectuer un codage à tous les rendements de codage illustrés dans le Tableau 1. En se référant au Tableau 1, on note que si un rapport des bits d'entrée (ou un rapport de quantité d'information, c'est-à-dire un rapport des premiers bits TFCI et des seconds bits TFCI) est 1:9, un rapport des symboles codés devient 3:29, 4:28 ou 5:27. Si le rapport des

bits d'entrée est 2:8, le rapport des symboles codés devient 6:26, 7:25 ou 8:24, et si le rapport des bits d'entrée est 3:7, le rapport des symboles codés devient 9:23, 10:22 ou 11:21. Si le rapport des bits d'entrée est 4:6, le rapport
 5 des symboles codés devient 12:20, 13:19 ou 14:18. Si le rapport des bits d'entrée est 6:4, le rapport des symboles codés devient 18:14, 19:13 ou 20:12, et si le rapport des bits d'entrée est 7:3, le rapport des symboles codés devient 21:11, 22:10 ou 23:9. Si le rapport des bits d'entrée est
 10 8:2, le rapport des symboles codés devient 24:8, 25:7 ou 26:6, et si le rapport des bits d'entrée est 9:1, le rapport des symboles codés devient 27:5, 28:4 ou 29:3;

Par conséquent, si le rapport des bits d'entrée est 1:9, alors il est nécessaire d'avoir {codeur (3,1), codeur
 15 (29,9), codeur (4,1) et codeur (28,9)} ou {codeur (5,1) et codeur (27,9)}. Si le rapport des bits d'entrée est 2:8, alors il est nécessaire d'avoir {codeur (6,2), (codeur 26,8), codeur (7,2) et codeur (25,8)} ou {codeur (8,2) et codeur (24,8)}. Si le rapport des bits d'entrée est 3:7
 20 alors il est nécessaire d'avoir {codeur (9,3), (codeur 23,7), codeur (10,3) et codeur (22,7)} ou {codeur (11,3) et codeur (21,7)}. Si le rapport des bits d'entrée est 4:6, alors il est nécessaire d'avoir {codeur (12,4), et (codeur 20,6), codeur (13,4) et codeur (19,6)} ou {codeur (14,4) et
 25 codeur (18,6)}. Par conséquent, lorsqu'on considère les 24 codeurs et le codeur (16,5) et le codeur (32,10) utilisés à l'heure actuelle, il existe un besoin portant sur un codeur capable de fonctionner comme les 18 codeurs avec une seule structure, afin d'augmenter les performances et de réduire
 30 la complexité du matériel.

De façon générale, la distribution de distance de Hamming pour des mots de code des codes correcteurs d'erreurs peut être utilisée comme une mesure indiquant les performances de codes correcteurs d'erreurs linéaires. La "dis-
 35 tance de Hamming" signifie le nombre de symboles différents

de zéro dans un mot de code. Ainsi, pour un certain mot de code "0111", le nombre de 1 inclus dans le mot de code est 3, ce qui fait que la distance de Hamming est 3. La plus faible valeur parmi les valeurs de distance de Hamming est
5 appelée une "distance minimale $d_{\min}$ ", et une augmentation de la distance minimale du mot de code améliore les performances de correction d'erreurs des codes correcteurs d'erreurs. En d'autres termes, le "code optimal" désigne un code ayant des performances de correction d'erreurs optimales. Ceci est
10 exposé en détail dans un ouvrage intitulé The Theory of Error-Correcting Codes, F.J. Macwilliams, N.J.A. Sloane, North-Holland.

De plus, dans le but d'utiliser une seule structure de codeur pour les codeurs ayant différentes longueurs, afin
15 de réduire la complexité du matériel, il est préférable de raccourcir le code ayant la plus grande longueur, c'est-à-dire le code (32,10). Pour le raccourcissement, il est nécessaire de poinçonner les symboles codés. Cependant, pendant le poinçonnage, la distance minimale du code varie
20 conformément aux positions de poinçonnage. Par conséquent, il est préférable de calculer les positions de poinçonnage de façon que le code poinçonné ait la distance minimale.

Par exemple, en termes de distance minimale, il est très préférable d'utiliser un code (7,2) optimal ayant l'un
25 des rendements de codage illustrés dans le Tableau 1, obtenu en répétant 3 fois un code simplex (3,2), et en poinçonnant ensuite les deux derniers symboles codés. Le Tableau 2 illustre la relation entre des bits d'information d'entrée du code simplex (3,2) et des mots de code simplex (3,2) émis
30 sur la base des bits d'information d'entrée.

Tableau 2

Bits d'information d'entrée	Mots de code simplex (3,2)
00	000
01	101
10	011
11	110

Le Tableau 3 illustre la relation entre les bits d'information d'entrée et des mots de code simplex (7,2) obtenus en répétant 3 fois le mot de code simplex (3,2) et en poinçonnant ensuite les deux derniers symboles codés.

Tableau 3

Bits d'information d'entrée	Mots de code simplex (7,2)
00	000 000 0
01	101 101 1
10	011 011 0
11	110 110 1

Cependant, les mots de code simplex (7,2) obtenus en répétant 3 fois le mot de code simplex (3,2) et en poinçonnant ensuite les deux derniers symboles codés peuvent être obtenus en raccourcissant le code de Reed-Muller (16,4) existant.

On présentera tout d'abord à titre d'exemple une description du procédé de raccourcissement. Le code de Reed-Muller (16,4) est une combinaison linéaire de 4 mots de code de base de longueur 16, le "4" étant le nombre de bits d'information d'entrée. Le fait de recevoir seulement 2 bits parmi les 16 bits d'information d'entrée équivaut à utiliser une combinaison linéaire de seulement 2 mots de code de base parmi les 4 mots de code de base de longueur 16, et à ne pas utiliser les mots de code restants. De plus, en restreignant

l'utilisation des mots de code de base et en poinçonnant ensuite 9 symboles parmi 16 symboles, il est possible de réaliser un codeur (7,2) utilisant le codeur (16,4). Le Tableau 4 illustre le procédé de raccourcissement.

5

Tableau 4

Bits d'info. d'entrée	Mots de code															
0000	0(*)	0	0	0	0(*)	0	0	0	0(*)	0	0(*)	0(*)	0(*)	0(*)	0(*)	0(*)
0001	0(*)	1	0	1	0(*)	1	0	1	0(*)	1	0(*)	1(*)	0(*)	1(*)	0(*)	1(*)
0010	0(*)	0	1	1	0(*)	0	1	1	0(*)	0	1(*)	1(*)	0(*)	0(*)	1(*)	1(*)
0011	0(*)	1	1	0	0(*)	1	1	0	0(*)	1	1(*)	0(*)	0(*)	1(*)	1(*)	0(*)
0100	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0101	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0
0110	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
0111	0	1	1	0	1	0	0	1	0	1	1	0	1	0	0	1
1000	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1001	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0
1010	0	0	1	1	0	0	1	1	1	1	0	0	1	1	0	0
1011	0	1	1	0	0	1	1	0	1	0	0	1	1	0	0	1
1100	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
1101	0	1	0	1	1	0	1	0	1	0	1	0	0	1	0	1
1110	0	0	1	1	1	1	0	0	1	1	0	0	0	0	1	1
1111	0	1	1	0	1	0	0	1	1	0	0	1	0	1	1	0

En se référant au Tableau 4, on note que chaque mot de code (16,4) est une combinaison linéaire des 4 mots de code de base de longueur 16. Pour obtenir le code (6,2), on utilise seulement les 2 mots de code supérieurs parmi les 4 mots de code de base. Ainsi, les 12 mots de code inférieurs restants sont automatiquement inutilisés. Par conséquent, seuls les 4 mots de code supérieurs sont utilisés. En outre, pour générer un mot de code de base de longueur 7 parmi les 4 mots de code de base supérieurs, il est nécessaire de poinçonner 9 symboles. Il est possible d'obtenir les mots de code simplex

(7,2) du Tableau 3 en poinçonnant les symboles indiqués par (*) dans le Tableau 4 et en collectant ensuite les 7 symboles codés restants.

On présentera ici une description d'une structure d'un codeur pour créer {code optimal (3,1), code optimal (29,9), code optimal (4,1) et code optimal (28,9)} et {code optimal (5,1) et code optimal (27,9)} utilisés pour le rapport de bits d'information de 1:9, une structure d'un codeur pour créer {code optimal (6,2), code optimal (26,8), code optimal (7,2) et code optimal (25,8)} et {code optimal (8,2) et code optimal (24,8)} utilisés pour le rapport de bits d'information de 2:8; une structure d'un codeur pour créer {code optimal (9,3), code optimal (23,7), code optimal (10,3) et code optimal (22,7)} et {code optimal (11,3) et code optimal (21,7)} utilisés pour le rapport de bits d'information de 3:7, une structure d'un codeur pour créer {code optimal (12,4), code optimal (20,6), code optimal (13,4) et code optimal (19,6)} et {code optimal (14,4) et code optimal (18,6)} utilisés pour le rapport de bits d'information de 4:6, et une structure d'un codeur pour créer un code optimal (16,5), et un code optimal (32,10) utilisés pour le rapport de bits d'information de 5:5, en raccourcissant un sous-code (32,10) du code de Reed-Muller du second ordre. De plus, on décrira également ci-dessous une structure d'un décodeur correspondant au codeur.

1. Premier Mode de Réalisation d'Emetteur

Un mode de réalisation de la présente invention procure un appareil et un procédé pour diviser 10 bits d'information dans un rapport de 1:9, 2:8, 3:7, 4:6, 5:5, 6:4, 7:3, 8:2 ou 9:1 avant le codage dans le mode de division ferme, comme on le fait dans le mode de division logique lorsque le rapport des bits d'information d'entrée est 5:5.

La figure 2 illustre une structure d'un émetteur conforme à un mode de réalisation de la présente invention. En se référant à la figure 2, on note que des bits TFCI pour

le DSCH et des bits TFCI pour le DCH, divisés selon l'un des rapports de bits d'information ci-dessus, sont respectivement fournis à des premier et second codeurs 200 et 205. Ici, les bits TFCI pour le DSCH sont désignés par TFCI(champ 1) ou premiers bits TFCI, tandis que les bits TFCI pour le DCH sont désignés par TFCI(champ 2) ou seconds bits TFCI. Les bits TFCI pour le DSCH sont générés par un premier générateur de bits TFCI 250, et les bits TFCI pour le DCH sont générés par un second générateur de bits TFCI 255. Le nombre des premiers bits TFCI est différent par rapport au nombre des seconds bits TFCI conformément aux rapports de bits d'information ci-dessus. De plus, un signal de commande indiquant une information de longueur de code, c'est-à-dire une information sur une valeur de longueur du mot de code fixé conformément au rapport de bits d'information, est appliqué aux premier et second codeurs 200 et 205. L'information de longueur de code est générée par un générateur d'information de longueur de code 260, et elle a une valeur variable conformément aux longueurs des premiers bits TFCI et des seconds bits TFCI.

Lorsque le rapport de bits d'information est 6:4, le codeur 200 reçoit un signal de commande de longueur pour permettre au codeur 200 de fonctionner comme un codeur (20,6), un codeur (19,6) ou un codeur (18,6), à la réception de 6 bits TFCI pour le DSCH, et il remplit la fonction de l'un des 3 codeurs, tandis que le codeur 205 reçoit un signal de commande de longueur pour permettre au codeur 205 de fonctionner comme un codeur (12,4), un codeur (13,4) ou un codeur (14,4), à la réception de 4 bits TFCI pour le DCH, et il fonctionne comme l'un des 3 codeurs. Lorsque le rapport de bits d'information est 7:3, le codeur 200 reçoit un signal de commande de longueur pour permettre au codeur 200 de fonctionner comme un codeur (23,7), un codeur (22,7) ou un codeur (21,7) à la réception de 7 bits TFCI pour le DSCH, et il fonctionne comme l'un des 3 codeurs, tandis que le codeur

205 reçoit un signal de commande de longueur pour permettre au codeur 205 de fonctionner comme un codeur (9,3), un codeur (10,3) ou un codeur (11,3) à la réception de 3 bits TFCI pour le DCH, et il fonctionne comme l'un des 3 codeurs.

5 Lorsque le rapport de bits d'information est 8:2, le codeur 200 reçoit un signal de commande de longueur pour permettre au codeur 200 de fonctionner comme un codeur (26,8), un codeur (25,8) ou un codeur (24,8) à la réception de 8 bits TFCI pour le DSCH, et il fonctionne comme l'un des 3 codeurs,

10 tandis que le codeur 205 reçoit un signal de commande de longueur pour permettre au codeur 205 de fonctionner comme un codeur (6,2), un codeur (7,2) ou un codeur (8,2) à la réception de 2 bits TFCI pour le DCH, et il fonctionne comme l'un des 3 codeurs. Lorsque le rapport de bits d'information est 9:1, le codeur 200 reçoit un signal de commande de longueur pour permettre au codeur 200 de fonctionner comme un codeur (29,9), un codeur (28,9) ou un codeur (27,9) à la réception de 8 bits TFCI pour le DSCH, et il fonctionne comme l'un des 3 codeurs, tandis que le codeur

15 205 reçoit un signal de commande de longueur pour permettre au codeur 205 de fonctionner comme un codeur (3,1), un codeur (4,1) ou un codeur (5,1) à la réception de 1 bit TFCI pour le DCH, et il remplit la fonction de l'un des 3 codeurs. Le signal de commande de longueur doit être généré de façon que la somme des premiers bits TFCI et des seconds bits TFCI devienne 32. Ainsi, si le premier codeur de TFCI est un codeur (4,1), le second codeur de TFCI doit être un codeur (28,9) plutôt qu'un codeur (29,9) ou un codeur (27,9). Si le second codeur de TFCI devient le codeur

25 (29,9), le nombre de bits codés b_1 devient 33, et si le second codeur de TFCI devient le codeur (27,9), le nombre de bits codés b_1 devient 31. Dans ce cas, l'émetteur n'est pas compatible avec l'émetteur classique qui utilise deux codeurs (16,5) ou un codeur (32,10). De plus, l'émetteur n'est

30 pas compatible avec l'émetteur classique dans l'application

35

des bits b_1 vers les bits d_m .

La figure 4 illustre une structure détaillée des codeurs 200 et 205. Ainsi, le codeur 200 pour le codage du premier codeur de TFCI et le codeur 205 pour le codage du
5 second codeur de TFCI ont la structure de la figure 4. Cependant, au moment de la génération des premiers mots de code TFCI et des seconds mots de code TFCI avec un retard, le premier codeur de TFCI et le second codeur de TFCI peuvent être réalisés avec un seul codeur. La figure 3 montre
10 une structure d'un émetteur pour générer les premiers mots de code TFCI et les seconds mots de code TFCI avec un retard.

Premièrement, en se référant à la figure 2, on présentera une description détaillée d'un codeur conforme à la présente invention pour le cas dans lequel le rapport des
15 premiers bits TFCI et des seconds bits TFCI est 1:9.

Lorsque le rapport de bits d'information est 1;9, le codeur 200 fonctionne comme un codeur (3,1) et le codeur 205 fonctionne comme un codeur (29,9); le codeur 200 fonctionne comme un codeur (4,1) et le codeur 205 fonctionne comme un
20 codeur (28,9); ou le codeur 200 fonctionne comme un codeur (5,1) et le codeur 205 fonctionne comme un codeur (27,9).

On va maintenant décrire en détail, en se référant à la figure 5, le fonctionnement du codeur (3,1), du codeur (29,9), du codeur (4,1), du codeur (28,9), du codeur (5,1)
25 et du codeur (27,9).

Premièrement, on décrira le fonctionnement du codeur (3,1). En se référant à la figure 4, on note qu'un bit d'entrée a_0 est normalement appliqué au codeur, et les bits d'entrée restants a_1 , a_2 , a_3 , a_4 , a_5 , a_6 , a_7 , a_8 et a_9 sont
30 tous remplis avec "0". Le bit d'entrée a_0 est appliqué à un multiplieur 410, le bit d'entrée a_1 est appliqué à un multiplieur 412, le bit d'entrée a_2 est appliqué à un multiplieur 414, le bit d'entrée a_3 est appliqué à un multiplieur 416, le bit d'entrée a_4 est appliqué à un multiplieur 418, le bit
35 d'entrée a_5 est appliqué à un multiplieur 420, le bit d'en-

trée a6 est appliqué à un multiplieur 422, le bit d'entrée a7 est appliqué à un multiplieur 424, le bit d'entrée a8 est appliqué à un multiplieur 426 et le bit d'entrée a9 est appliqué à un multiplieur 428. En même temps, un générateur de code de Walsh 400 génère un mot de code de base W1 = 101010101010101101010101010100, et applique au multiplieur 410 le mot de code de base W1 généré. Le multiplieur 410 multiplie ensuite le bit d'entrée a0 par le mot de code de base W1 en une unité de symbole, et il applique son signal de sortie à un opérateur OU exclusif (OU-EX) 440. En outre, le générateur de code de Walsh 400 génère d'autres mots de code de base W2, W4, W8 et W16, et il les applique respectivement aux multiplieurs 412, 414, 416 et 418. Un générateur de code ne contenant que des 1, 402, génère un mot de code de base ne contenant que des 1 (ou séquence ne contenant que des 1), il applique au multiplieur 420 le mot de code de base ne contenant que des 1 qui est généré. Un générateur de masque 404 génère des mots de code de base M1, M2, M4 et M8, et il applique respectivement aux multiplieurs 422, 424, 426 et 428 les mots de code de base M1, M2, M4 et M8 générés. Cependant, du fait que les bits d'entrée a1, a2, a3, a4, a5, a6, a7, a8 et a9 appliqués aux multiplieurs 412, 414, 416, 418, 420, 422, 424, 426 et 428 sont tous égaux à 0, les multiplieurs 412, 414, 416, 418, 420, 422, 424, 426 et 428 émettent des 0 vers l'opérateur OU exclusif 440, et par conséquent ils n'affectent pas le signal de sortie de l'opérateur OU exclusif 440. Ainsi, une valeur déterminée par la combinaison OU-EX des valeurs de sortie des multiplieurs 410, 412, 414, 416, 418, 420, 422, 424, 426 et 428 par l'opérateur OU exclusif 440, est égale à la valeur de sortie du multiplieur 410. Les 32 symboles émis par l'opérateur OU exclusif 440 sont appliqués à un poinçonneur 460. A ce moment, une unité de commande 450 reçoit une information de longueur de code et fournit au poinçonneur 460 un signal de commande indiquant des positions de poinçonnage basées sur

l'information de longueur de code. Le poinçonneur 460 poinçonne ensuite les 1-ier, 3-ième, 5-ième, 6-ième, 7-ième, 8-ième, 9-ième, 10-ième, 11-ième, 12-ième, 13-ième, 14-ième, 15-ième, 16-ième, 17-ième, 18-ième, 19-ième, 20-ième, 21-ième, 22-ième, 23-ième, 24-ième, 25-ième, 26-ième, 27-ième, 28-ième, 29-ième, 30-ième et 31-ième symboles codés parmi un total de 32 symboles codés des 0-ième à 31-ième symboles, conformément au signal de commande provenant de l'unité de commande 450. En d'autres termes, le poinçonneur 460 poinçonne 29 symboles parmi les 32 symboles codés, et il émet donc 3 symboles codés non poinçonnés.

Secondement, on décrira le fonctionnement du codeur (29,9). En se référant à la figure 4, on note que neuf bits d'entrée a0, a1, a2, a3, a4, a5, a6, a7 et a8 sont normalement appliqués au codeur, et le bit d'entrée a9 restant est rempli avec "0". Le bit d'entrée a0 est appliqué au multiplieur 410, le bit d'entrée a1 est appliqué au multiplieur 412, le bit d'entrée a2 est appliqué au multiplieur 414, le bit d'entrée a3 est appliqué au multiplieur 416, le bit d'entrée a4 est appliqué au multiplieur 418, le bit d'entrée a5 est appliqué au multiplieur 420, le bit d'entrée a6 est appliqué au multiplieur 422, le bit d'entrée a7 est appliqué au multiplieur 424, le bit d'entrée a8 est appliqué au multiplieur 426 et le bit d'entrée a9 est appliqué au multiplieur 428. En même temps, un générateur de code de Walsh 400 fournit au multiplieur 410 le mot de code de base W1 = 101010101010101101010101010100, au multiplieur 412 le mot de code de base W2 = 01100110011001101100110011001100 au multiplieur 414 le mot de code de base W4 = 00011110000111100011110000111100, au multiplieur 416 le mot de code de base W8 = 00000001111111100000001111111100, et au multiplieur 418 le mot de code de base W16 = 0000000000000001111111111111101. Ensuite, le multiplieur 410 multiplie le mot de code de base W1 par le bit d'entrée a0 dans l'unité de symbole et il applique son si-

gnal de sortie à l'opérateur OU exclusif 440, le multiplieur 412 multiplie le mot de code de base W2 par le bit d'entrée a1 dans l'unité de symbole et il applique son signal de sortie à l'opérateur OU exclusif 440, le multiplieur 414 multiplie le mot de code de base W4 par le bit d'entrée a2 dans l'unité de symbole et il applique son signal de sortie à l'opérateur OU exclusif 440, le multiplieur 416 multiplie le mot de code de base W8 par le bit d'entrée a3 dans l'unité de symbole et il applique son signal de sortie à l'opérateur OU exclusif 440, et le multiplieur 418 multiplie le mot de code de base W16 par le bit d'entrée a4 dans l'unité de symbole et il applique son signal de sortie à l'opérateur OU exclusif 440. De plus, le générateur de code ne contenant que des 1, 402, génère un mot de code de base ne contenant que des 1, de longueur 32, et il applique au multiplieur 420 le mot de code de base ne contenant que des 1 qui est généré. Le multiplieur 420 multiplie ensuite le mot de code de base ne contenant que des 1 par le bit d'entrée a5 dans l'unité de symbole, et il applique son signal de sortie à l'opérateur OU exclusif 440. Le générateur de masque 404 applique au multiplieur 422 le mot de code de base M1 = 0101 0000 1100 0111 1100 0001 1101 1101, au multiplieur 424 le mot de code de base M2 = 0000 0011 1001 1011 1011 0111 0001 1100, et au multiplieur 426 le mot de code de base M4 = 0001 0101 11111 0010 0110 1100 1010 1100. Ensuite, le multiplieur 422 multiplie le mot de code de base M1 par le bit d'entrée a6 dans l'unité de symbole et il applique son signal de sortie à l'opérateur OU exclusif 440, le multiplieur 424 multiplie le mot de code de base M2 par le bit d'entrée a7 dans l'unité de symbole et il applique son signal de sortie à l'opérateur OU exclusif 440, et le multiplieur 426 multiplie le mot de code de base M4 par le bit d'entrée a8 dans l'unité de symbole et il applique son signal de sortie à l'opérateur OU exclusif 440. En outre, le générateur de masque 404 génère l'autre mot de code de base M8, et il applique au

multiplieur 428 le mot de code de base M8 généré. Cependant, du fait que le bit d'entrée a9 appliqué au multiplieur 428 est 0, le multiplieur 428 envoie 0 à l'opérateur OU exclusif 440, et de ce fait il n'affecte pas le signal de sortie de l'opérateur OU exclusif 440. Ainsi, une valeur déterminée en combinant par une fonction OU-EX les valeurs de sortie des multiplieurs 410, 412, 414, 416, 418, 420, 422, 424, 426 et 428 par l'opérateur OU exclusif 440, est égale à une valeur déterminée en combinant par une fonction OU-EX les valeurs de sortie des multiplieurs 412, 414, 416, 418, 420, 422, 424 et 426. Les 32 symboles émis par l'opérateur OU exclusif 440 sont appliqués au poinçonneur 460. A ce moment, l'unité de commande 450 reçoit une information de longueur de code et elle fournit au poinçonneur 460 un signal de commande indiquant des positions de poinçonnage basées sur l'information de longueur de code. Le poinçonneur 460 poinçonne ensuite les 6-ième, 10-ième et 11-ième symboles codés, parmi un total de 32 symboles codés allant du 0-ième au 31-ième symboles, conformément au signal de commande provenant de l'unité de commande 450. En d'autres termes, le poinçonneur 460 poinçonne 3 symboles parmi les 32 symboles codés, et il fournit ainsi en sortie 29 symboles codés non poinçonnés.

Troisièmement, on décrira le fonctionnement du codeur (4,1). En se référant à la figure 4, on note qu'un bit d'entrée a0 est normalement appliqué au codeur, et les bits d'entrée restants a1, a2, a3, a4, a5, a6, a7, a8 et a9 sont tous remplis avec "0". Le bit d'entrée a0 est appliqué au multiplieur 410, le bit d'entrée a1 est appliqué au multiplieur 412, le bit d'entrée a2 est appliqué au multiplieur 414, le bit d'entrée a3 est appliqué au multiplieur 416, le bit d'entrée a4 est appliqué au multiplieur 418, le bit d'entrée a5 est appliqué au multiplieur 420, le bit d'entrée a6 est appliqué au multiplieur 422, le bit d'entrée a7 est appliqué au multiplieur 424, le bit d'entrée a8 est appliqué au multiplieur 426 et le bit d'entrée a9 est appliqué au

multiplieur 428. En même temps, le générateur de code de Walsh 400 génère un mot de code de base W1 = 101010101010101101010101010100, et applique au multiplieur 410 le mot de code de base W1 généré. Le multiplieur 410 multiplie ensuite le bit d'entrée a0 par le mot de code de base W1 dans une unité de symbole, et il fournit son signal de sortie à l'opérateur OU-EX 440. En outre, le générateur de code de Walsh 400 génère les autres mots de code de base W2, W4, W8 et W16, et il les applique respectivement aux multiplieurs 412, 414, 416 et 418. Le générateur de code ne contenant que des 1, 402, génère un mot de code de base ne contenant que des 1 (ou séquence ne contenant que des 1), et il applique au multiplieur 420 le mot de code de base ne contenant que des 1 qui est généré. Le générateur de masque 404 génère des mots de code de base M1, M2, M4 et M8, et il applique respectivement aux multiplieurs 422, 424, 426 et 428 les mots de code de base M1, M2, M4 et M8 qui sont générés. Cependant, du fait que les bis d'entrée a1, a2, a3, a4, a5, a6, a7, a8 et a9 appliqués aux multiplieurs 412, 414, 416, 418, 420, 422, 424, 426 et 428 sont tous des 0, les multiplieurs 412, 414, 416, 418, 420, 422, 424, 426 et 428 émettent des 0 vers l'opérateur OU exclusif 450, ce qui fait qu'ils n'affectent pas le signal de sortie de l'opérateur OU exclusif 440. Ainsi, une valeur déterminée en combinant par une fonction OU-EX les valeurs de sortie des multiplieurs 410, 412, 414, 416, 418, 420, 422, 424, 426 et 428, par l'opérateur OU exclusif 440, est égale à la valeur de sortie du multiplieur 410. Les 32 symboles émis par l'opérateur OU exclusif 440 sont appliqués au poinçonneur 460. A ce moment, l'unité de commande 450 reçoit l'information de longueur de code et fournit au poinçonneur 460 un signal de commande indiquant des positions de poinçonnage basées sur l'information de longueur de code. Le poinçonneur 460 poinçonne ensuite les 1-ier, 3-ième, 5-ième, 7-ième, 8-ième, 9-ième, 10-ième, 11-ième, 12-ième, 13-ième, 14-ième, 15-ième, 16-ième,

17-ième, 18-ième, 19-ième, 20-ième, 21-ième, 22-ième, 23-ième, 24-ième, 25-ième, 26-ième, 27-ième, 28-ième, 29-ième, 30-ième et 31-ième symboles codés parmi un total de 32 symboles codés allant du 0-ième au 31-ième symboles, conformément au signal de commande émis par l'unité de commande 450. En d'autres termes, le poinçonneur 460 poinçonne 28 symboles parmi les 32 symboles codés, et il fournit donc en sortie 4 symboles codés non poinçonnés.

Quatrièmement, on décrira le fonctionnement du codeur (28,9). En se référant à la figure 4, on note que neuf bits d'entrée a0, a1, a2, a3, a4, a5, a6, a7 et a8 sont normalement fournis au codeur, et le bit d'entrée a9 restant est rempli avec "0". Le bit d'entrée a0 est appliqué au multiplieur 410, le bit d'entrée a1 est appliqué au multiplieur 412, le bit d'entrée a2 est appliqué au multiplieur 414, le bit d'entrée a3 est appliqué au multiplieur 416, le bit d'entrée a4 est appliqué au multiplieur 418, le bit d'entrée a5 est appliqué au multiplieur 420, le bit d'entrée a6 est appliqué au multiplieur 422, le bit d'entrée a7 est appliqué au multiplieur 424, le bit d'entrée a8 est appliqué au multiplieur 426 et le bit d'entrée a9 est appliqué au multiplieur 428. En même temps, le générateur de code de Walsh 400 applique au multiplieur 410 le mot de code de base W1 = 101010101010101101010101010100, au multiplieur 412 le mot de code de base W2 = 01100110011001101100110011001100 au multiplieur 414 le mot de code de base WA = 00011110000111100011110000111100, au multiplieur 416 le mot de code de base W8 = 00000001111111100000001111111100, et au multiplieur 418 le mot de code de base W16 = 0000000000000001111111111111101. Ensuite, le multiplieur 410 multiplie le mot de code de base W1 par le bit d'entrée a0 dans l'unité de symbole, et il applique son signal de sortie à l'opérateur OU exclusif 440, le multiplieur 412 multiplie le mot de code de base W2 par le bit d'entrée a1 dans l'unité de symbole et il applique son signal de sor-

tie à l'opérateur OU exclusif 440, le multiplieur 414 multiplie le mot de code de base W4 par le bit d'entrée a2 dans l'unité de symbole et il applique son signal de sortie à l'opérateur OU exclusif 440, le multiplieur 416 multiplie le mot de code de base W8 par le bit d'entrée a3 dans l'unité de symbole et il applique son signal de sortie à l'opérateur OU exclusif 440, et le multiplieur 418 multiplie le mot de code de base W16 par le bit d'entrée a4 dans l'unité de symbole et il applique son signal de sortie à l'opérateur OU exclusif 440. De plus, le générateur de code ne contenant que des 1, 402, génère un mot de code de base de longueur 32 ne contenant que des 1, et il applique au multiplieur 420 le mot de code de base ne contenant que des 1 qui est généré. Le multiplieur 420 multiplie ensuite le mot de code de base ne contenant que des 1 par le bit d'entrée a5 dans l'unité de symbole et il applique son signal de sortie à l'opérateur OU exclusif 440. Le générateur de masque 404 applique au multiplieur 422 le mot de code de base M1 = 0101 0000 1100 0111 1100 0001 1101 1101, au multiplieur 424 le mot de code de base M2 = 0000 0011 1001 1011 1011 0111 0001 1100, et au multiplieur 426 le mot de code de base M4 = 0001 0101 1111 0010 0110 1100 1010 1100. Ensuite, le multiplieur 422 multiplie le mot de code de base M1 par le bit d'entrée a6 dans l'unité de symbole et il applique son signal de sortie à l'opérateur OU exclusif 440, le multiplieur 424 multiplie le mot de code de base M2 par le bit d'entrée a7 dans l'unité de symbole et il applique son signal de sortie à l'opérateur OU exclusif 440, et le multiplieur 426 multiplie le mot de code de base M4 par le bit d'entrée a8 dans l'unité de symbole et il applique son signal de sortie à l'opérateur OU exclusif 440. En outre, le générateur de masque 404 génère l'autre mot de code de base M8, et il applique au multiplieur 428 le mot de code de base M8 généré. Cependant, du fait que le bit d'entrée a9 appliqué au multiplieur 428 est 0, le multiplieur 428 émet 0 vers l'opérateur OU exclusif

440, ce qui fait qu'il n'affecte pas le signal de sortie de l'opérateur OU exclusif 440. Ainsi, une valeur déterminée par une combinaison OU-EX des valeurs de sortie des multiplieurs 410, 412, 414, 416, 418, 420, 422, 424, 426 et 428 par l'opérateur OU exclusif 440 est égale à une valeur déterminée par une combinaison OU-EX des valeurs de sortie des multiplieurs 410, 412, 414, 416, 418, 420, 422, 424 et 426. Les 32 symboles émis par l'opérateur OU exclusif 440 sont appliqués au poinçonneur 460. A ce moment, l'unité de commande 450 reçoit une information de longueur de code et fournit au poinçonneur 460 un signal de commande indiquant des positions de poinçonnage basées sur l'information de longueur de code. Le poinçonneur 460 poinçonne ensuite les 6-ième, 10-ième, 11-ième et 30-ième symboles codés parmi un total de 32 symboles codés allant du 0-ième au 31-ième symboles, conformément au signal de commande émis par l'unité de commande 450. En d'autres termes, le poinçonneur 460 poinçonne 4 symboles parmi les 32 symboles codés, et il émet donc 28 symboles codés non poinçonnés.

Cinquièmement, on décrira le fonctionnement du codeur (5,1). En se référant à la figure 4, on note qu'un bit d'entrée a0 est fourni normalement au codeur, et les bits d'entrée restants a1, a2, a3, a4, a5, a6, a7, a8 et a9 sont tous remplis avec "0". Le bit d'entrée a0 est appliqué au multiplieur 410, le bit d'entrée a1 est appliqué au multiplieur 412, le bit d'entrée a2 est appliqué au multiplieur 414, le bit d'entrée a3 est appliqué au multiplieur 416, le bit d'entrée a4 est appliqué au multiplieur 418, le bit d'entrée a5 est appliqué au multiplieur 420, le bit d'entrée a6 est appliqué au multiplieur 422, le bit d'entrée a7 est appliqué au multiplieur 424, le bit d'entrée a8 est appliqué au multiplieur 426 et le bit d'entrée a9 est appliqué au multiplieur 428. En même temps, le générateur de code de Walsh 400 génère le mot de code de base W1 = 101010101010101101010101010100, et il applique au multi-

plieur 410 le mot de code de base généré. Le multiplieur 410 multiplie ensuite le mot de code de base W1 par le bit d'entrée a0 dans l'unité de symbole et fournit son signal de sortie à l'opérateur OU exclusif 440. En outre, le générateur de code de Walsh 400 génère les autres mots de code de base W2, W4, W8 et W16, et il fournit respectivement aux multiplieurs 412, 414, 416 et 418 les mots de code de base W2, W4, W8 et W16 qui sont générés. Le générateur de code ne contenant que des 1, 402, génère un mot de code de base de longueur 32 ne contenant que des 1, et il applique au multiplieur 420 le mot de code de base ne contenant que des 1 qui est généré. En outre, le générateur de masque 404 génère les mots de code de base M1, M2, M4 et M8 et il applique respectivement aux multiplieurs 422, 424, 426 et 428 les mots de code de base M1, M2, M4 et M8 qui sont générés. Cependant, du fait que les bits d'entrée a1, a2, a3, a4, a5, a6, a7, a8 et a9 appliqués aux multiplieurs 412, 414, 416, 418, 420, 422, 424, 426 et 428 sont tous des 0, les multiplieurs 412, 414, 416, 418, 420, 422, 424, 426 et 428 émettent des 0 vers l'opérateur OU exclusif 440, ce qui fait qu'ils n'affectent pas le signal de sortie de l'opérateur OU exclusif 440. Ainsi, une valeur déterminée en combinant par une fonction OUX les valeurs de sortie des multiplieurs 410, 412, 414, 416, 418, 420, 422, 424, 426 et 428 par l'opérateur OU exclusif 440, est égale à la valeur de sortie du multiplieur 410. Les 32 symboles émis par l'opérateur OU exclusif 440 sont appliqués au poinçonneur 460. A ce moment, l'unité de commande 450 reçoit une information de longueur de code et fournit au poinçonneur 460 un signal de commande indiquant des positions de poinçonnage basées sur l'information de longueur de code. Le poinçonneur 460 poinçonne ensuite les 1-ier, 3-ième, 5-ième, 7-ième, 9-ième, 10-ième, 11-ième, 12-ième, 13-ième, 14-ième, 15-ième, 16-ième, 17-ième, 18-ième, 19-ième, 20-ième, 21-ième, 22-ième, 23-ième, 24-ième, 25-ième, 26-ième, 27-ième, 28-ième, 29-ième,

30-ième et 31-ième symboles codés parmi un total de 32 symboles codés allant du 0-ième au 31-ième symboles, conformément au signal de commande émis par l'unité de commande 450. En d'autres termes, le poinçonneur 460 poinçonne 27 symboles
 5 parmi les 32 symboles codés, et il émet donc 5 symboles codés non poinçonnés.

Sixièmement, on décrira le fonctionnement du codeur (27,9). En se référant à la figure 4, on note que neuf bits d'entrée a0, a1, a2, a3, a4, a5, a6, a7 et a8 sont normale-
 10 ment appliqués au codeur, et le bit d'entrée a9 restant est rempli avec "0". Le bit d'entrée a0 est appliqué au multiplieur 410, le bit d'entrée a1 est appliqué au multiplieur 412, le bit d'entrée a2 est appliqué au multiplieur 414, le bit d'entrée a3 est appliqué au multiplieur 416, le bit
 15 d'entrée a4 est appliqué au multiplieur 418, le bit d'entrée a5 est appliqué au multiplieur 420, le bit d'entrée a6 est appliqué au multiplieur 422, le bit d'entrée a7 est appliqué au multiplieur 424, le bit d'entrée a8 est appliqué au multiplieur 426 et le bit d'entrée a9 est appliqué au multi-
 20 plieur 428. En même temps, le générateur de code de Walsh 400 applique au multiplieur 410 le mot de code de base W1 = 1010101010101101010101010100, au multiplieur 412 le mot de code de base W2 = 0110011001100110110011001100, au multiplieur 414 le mot de code de base W4 =
 25 00011110000111100011110000111100, au multiplieur 416 le mot de code de base W8 = 00000001111111100000001111111100, et au multiplieur 418 le mot de code de base W16 = 0000000000000001111111111111101. Ensuite, le multiplieur 410 multiplie le mot de code de base W1 par le bit d'entrée
 30 a0 dans l'unité de symbole et applique son signal de sortie à l'opérateur OU exclusif 440, le multiplieur 412 multiplie le mot de code de base W2 par le bit d'entrée a1 dans l'unité de symbole et applique son signal de sortie à l'opérateur OU exclusif 440, le multiplieur 414 multiplie le mot de code de base W4 par le bit d'entrée a2 dans l'unité de symbole et
 35 de base W4 par le bit d'entrée a2 dans l'unité de symbole et

applique son signal de sortie à l'opérateur OU exclusif 440, le multiplieur 416 multiplie le mot de code de base W8 par le bit d'entrée a3 dans l'unité de symbole et applique son signal de sortie à l'opérateur OU exclusif 440, et le multiplieur 418 multiplie le mot de code de base W16 par le bit d'entrée a4 dans l'unité de symbole et applique son signal de sortie à l'opérateur OU exclusif 440. Le générateur de code ne contenant que des 1, 402, génère un mot de code de base ne contenant que des 1 et il applique au multiplieur 420 le mot de code de base ne contenant que des 1 qui est généré. Le multiplieur 420 multiplie ensuite le mot de code de base ne contenant que des 1 par le bit d'entrée a5 dans l'unité de symbole, et il applique son signal de sortie à l'opérateur OU exclusif 440. Le générateur de masque 404 applique au multiplieur 422 le mot de code de base M1 = 0101 0000 1100 0111 1100 0001 1101 1101, au multiplieur 424 le mot de code de base M2 = 0000 0011 1001 1011 1011 0111 0001 1100, et au multiplieur 426 le mot de code de base M4 = 0001 0101 1111 0010 0110 1100 1010 1100. Ensuite, le multiplieur 422 multiplie le mot de code de base M1 par le bit d'entrée a6 dans l'unité de symbole et il applique son signal de sortie à l'opérateur OU exclusif 440, le multiplieur 424 multiplie le mot de code de base M2 par le bit d'entrée a7 dans l'unité de symbole et applique son signal de sortie à l'opérateur OU exclusif 440, et le multiplieur 426 multiplie le mot de code de base M4 par le bit d'entrée a8 dans l'unité de symbole et applique son signal de sortie à l'opérateur OU exclusif 440. En outre, le générateur de masque 404 génère l'autre mot de code de base M8, et il applique au multiplieur 428 le mot de code de base M8 qui est généré. Cependant, du fait que le bit d'entrée a9 appliqué au multiplieur 428 est 0, le multiplieur 428 émet des 0 vers l'opérateur OU exclusif 440, ce qui fait qu'il n'affecte pas le signal de sortie de l'opérateur OU exclusif 440. Ainsi, une valeur déterminée par la combinaison OU-EX des valeurs de sortie des

multiplieurs 410, 412, 414, 416, 418, 420, 422, 424, 426 et 428, par l'opérateur OU exclusif 440, est égale à une valeur déterminée par une combinaison OU-EX des valeurs de sortie des multiplieurs 410, 412, 414, 416, 418, 420, 422, 424 et 5 426. Les 32 symboles émis par l'opérateur OU exclusif 440 sont appliqués au poinçonneur 460. A ce moment, l'unité de commande 450 reçoit une information de longueur de code et applique au poinçonneur 460 un signal de commande indiquant des positions de poinçonnage basées sur l'information de 10 longueur de code. Le poinçonneur 460 poinçonne ensuite les 0-ième, 2-ième, 8-ième, 19-ième et 20-ième symboles codés, parmi un total de 32 symboles codés allant du 0-ième au 31-ième symboles, conformément au signal de commande émis par l'unité de commande 450. En d'autres termes, le poinçonneur 15 460 poinçonne 5 symboles parmi les 32 symboles codés, et émet donc 27 symboles codés non poinçonnés.

Le Tableau 5 ci-dessous illustre des configurations de poinçonnage avec lesquelles tous les codeurs du Tableau 1 peuvent être réalisés au moyen du codeur de la figure 4. Les 20 configurations de poinçonnage du Tableau 5 sont appliquées au poinçonneur 460 de la figure 4 pour réaliser un code (n,k) (avec $n = 3,4,\dots,14,18,19,\dots,29$ et $k = 1,2,3,4,6,7,8,9$).

Tableau 5

Code	0 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
(3,1)	10101	00000	00000	00000	00000	00000	00
(4,1)	10101	01000	00000	00000	00000	00000	00
(5,1)	10101	01010	00000	00000	00000	00000	00
(6,2)	11101	11000	00000	00000	00000	00000	00
(7,2)	11101	11010	00000	00000	00000	00000	00
(8,2)	11101	11011	00000	00000	00000	00000	00
(9,3)	11111	11001	10000	00000	00000	00000	00
(10,3)	11111	11011	01000	00000	00000	00000	00
(11,3)	11111	11000	01111	00000	00000	00000	00
(12,4)	00011	11111	11111	00000	00000	00000	00
(13,4)	00011	11111	11111	10000	00000	00000	00
(14,4)	00011	11111	11111	11000	00000	00000	00
(18,6)	01111	11010	10111	10110	11110	00000	00
(19,6)	11111	10111	00100	10010	01010	10111	10
(20,6)	11111	11111	01000	11110	00100	11001	10
(21,7)	00000	01011	11011	11101	10110	11111	11
(22,7)	11111	11101	11011	10100	11101	10011	00
(23,7)	11110	11100	11011	10101	11100	11111	01
(24,8)	10111	11011	11101	01111	01111	01111	00
(25,8)	11110	11111	10110	01111	00011	11111	11
(26,8)	11111	11011	11101	01111	01111	01111	01
(27,9)	01011	11101	11111	11110	01111	11111	11
(28,9)	11111	10111	00101	11111	11111	11111	11
(29,9)	11111	10111	00111	11111	11111	11111	11

Dans le Tableau 5, "0" représente une position à laquelle le symbole codé est poinçonné, tandis que "1" représente une position à laquelle le symbole codé n'est pas poinçonné. En utilisant les configurations de poinçonnage du Tableau 5, il est possible de calculer les premiers symboles TFCI codés et les seconds symboles TFCI codés, même pour les cas dans lesquels les rapports entre les premiers bits d'information TFCI et les seconds bits d'information TFCI sont 2:8, 3:7, 4:6, 6:4, 7:3, 8:2 et 9:1. Le fonctionnement des codeurs 200 et 205 pourrait mieux ressortir des configurations de poinçonnage du Tableau 5 et de la description précédente faite pour le cas dans lequel le rapport entre les premiers bits d'information TFCI et les seconds bits d'information TFCI est 1:9.

Après les opérations ci-dessus, les symboles codés

émis par les codeurs 200 et 205 sont arrangés (ou multiplexés dans le temps) par un dispositif d'arrangement (ou multiplexeur) 210, générant un signal multiplexé à 32 symboles.

On décrira ensuite un procédé pour arranger les premiers symboles TFCI codés et les seconds symboles TFCI codés, au moyen du dispositif d'arrangement de symboles codés 210. Le dispositif d'arrangement de symboles codés 210 arrange les premiers symboles TFCI codés et les seconds symboles TFCI codés émis par les codeurs 200 et 205, de façon que les symboles TFCI codés soient arrangés aussi uniformément que possible dans une trame radio. Ainsi, le dispositif d'arrangement de symboles codés 210 effectue une application des bits d'information a_k vers les bits codés b_1 , définis dans la description de l'art antérieur. Parmi les symboles codés obtenus en codant les bits d'information a_k , un x -ième symbole codé parmi les symboles codés obtenus en codant les premiers bits TFCI est défini par c_x^1 , en désignant par x un entier incluant "0", et un y -ième symbole codé parmi les symboles codés obtenus en codant les seconds bits TFCI est défini par c_y^2 , en désignant par y un entier incluant "0". La somme d'une valeur de x du dernier symbole pour le c_x^1 et d'une valeur de y du dernier symbole pour le c_y^2 doit toujours être égale à 32. De plus, la somme du nombre des symboles codés c_x^1 et du nombre des symboles codés c_y^2 est égale à 32. Par conséquent, le dispositif d'arrangement de symboles codés 210 a une fonction consistant à effectuer une application des symboles codés c_x^1 et c_y^2 vers les bits b_1 . Les bits b_1 sont appliqués vers les bits d_m avant d'être émis sur la trame radio réelle pour les cas respectifs des conditions A1, A2, A3 et A4.

Dans les conditions A1, A2, A3 et A4, les 32 bits b_1 sont tous transmis de façon souhaitable. Cependant, dans la condition A1, les bits $d_{30}(b_{30})$ et $d_{31}(b_{31})$ ne sont pas transmis, ce qui fait qu'il est nécessaire de sélectionner l'un des symboles codés c_x^1 et c_y^2 devant faire l'objet

d'une application vers les bits $d_{30}(b_{30})$ et $d_{31}(b_{31})$. Les règles d'application des symboles codés c_x^1 et c_y^2 vers les bits $d_{30}(b_{30})$ et $d_{31}(b_{31})$ sont données ci-dessous.

Règle 1 : les derniers symboles codés des premiers symboles TFCI codés et des seconds symboles TFCI codés sont appliqués vers $d_{30}(b_{30})$ et $d_{31}(b_{31})$.

Règle 2 : des symboles codés arbitraires des premiers symboles TFCI codés et des seconds symboles TFCI codés sont appliqués vers $d_{30}(b_{30})$ et $d_{31}(b_{31})$.

Règle 3 : deux symboles codés arbitraires émis par un codeur avec un rendement de codage accru sont appliqués vers $d_{30}(b_{30})$ et $d_{31}(b_{31})$.

Règle 4 : deux symboles codés arbitraires émis par un codeur avec un rendement de codage élevé sont appliqués vers $d_{30}(b_{30})$ et $d_{31}(b_{31})$.

Règle 5 : deux symboles codés arbitraires émis par un codeur autre que le codeur avec un rendement de codage accru sont appliqués vers $d_{30}(b_{30})$ et $d_{31}(b_{31})$.

Dans l'application de la Règle 1, de la Règle 2, de la Règle 3, de la Règle 4 et de la Règle 5, on doit considérer ce qui suit. Ainsi, lorsqu'un ou deux symboles codés parmi les symboles codés c_x^1 et c_y^2 de chaque code ne sont pas transmis, on doit considérer (1) comment les performances du code utilisé pour le premier TFCI ou le second TFCI seront changées, (2) quel TFCI parmi le premier TFCI et le second TFCI doit avoir une fiabilité (ou des performances) accrue, (3) quels symboles codés parmi les symboles codés c_x^1 et c_y^2 émis par les codeurs respectifs doivent être appliqués vers $d_{30}(b_{30})$ et $d_{31}(b_{31})$ pour minimiser la dégradation de performances des codes, et (4) quel TFCI parmi le premier TFCI et le second TFCI doit être accentué pendant la transmission.

Dans la description suivante de la Règle 1, de la Règle 2, de la Règle 3 et de la Règle 5, on supposera que le rapport de bits d'information du premier TFCI au second TFCI

est 3:7 dans le mode HSM. En outre, dans la description de la Règle 4, on supposera que le rapport de bits d'information du premier TFCI au second TFCI est 3:7 pour la condition A1.

5 On présentera ci-dessous une description de la Règle 1 en référence à un exemple. Un code (9,3) et un code (23,7), ou un code (11,3) et un code (21,7) sont disponibles conformément au rapport de bits d'information du premier TFCI au second TFCI. Le code (9,3) et le code (23,7) sont
10 utilisés pour augmenter les performances de code du second TFCI, tandis que le code (11,3) et le code (21,7) sont utilisés pour augmenter les performances de code du premier TFCI. Lorsqu'on applique la Règle 1, le dernier symbole codé du code (9,3) n'est pas transmis, ce qui fait qu'un rendement de codage réel du code (9,3) devient (8,3). Le dernier
15 symbole codé du code (23,7) n'est pas transmis, ce qui fait qu'un rendement de codage réel du code (23,7) devient (22,7); le dernier symbole codé du code (11,3) n'est pas transmis, ce qui fait qu'un rendement de codage réel du code (11,3) devient (10,3); et le dernier symbole codé du code (21,7) n'est pas transmis, ce qui fait qu'un rendement de codage réel du code (21,7) devient (20,7). Dans la Règle 1, les codeurs effectuent une application de leurs derniers symboles codés vers $d_{30}(b_{30})$ et $d_{31}(b_{31})$, ce qui contribue à
20 la simplification de l'application. Cependant, dans la condition A1, le rendement de codage réel du premier TFCI au second TFCI est diminué, ce qui entraîne une réduction des performances de code du premier TFCI et du second TFCI.

 On présentera ci-dessous une description de la Règle
30 2 en référence à un exemple. Un code (9,3) et un code (23,7), ou un code (11,3) et un code (21,7) sont disponibles conformément au rapport de bits d'information du premier TFCI au second TFCI. Lorsque la Règle 2 est appliquée, un symbole codé arbitraire du code (9,3) n'est pas transmis, ce
35 qui fait qu'un rendement de codage réel du code (9,3) de-

vient (8,3); un symbole codé arbitraire du code (23,7) n'est pas transmis, ce qui fait qu'un rendement de codage réel du code (23,7) devient (22,7); un symbole codé arbitraire du code (11,3) n'est pas transmis, ce qui fait qu'un rendement de codage réel du code (11,3) devient (10,3); et un symbole codé arbitraire du code (21,7) n'est pas transmis, ce qui fait qu'un rendement de codage réel du code (21,7) devient (20,7). Les symboles codés arbitraires peuvent être sélectionnés parmi les 4 codes, de façon que les performances de code réelles ne soient pas diminuées, bien que les rendements de codage réels des codes respectifs soient diminués. Cependant, plusieurs codes peuvent avoir des performances diminuées, indépendamment des symboles codés arbitraires sélectionnés. La Règle 2 est plus complexe que la Règle 1 dans un procédé d'application des symboles codés c_x^1 et c_y^2 . Cependant, dans la condition A1, il est possible de maintenir les performances de code du premier TFCI et du second TFCI, indépendamment de la réduction du rendement de codage réel des codeurs pour le premier TFCI et le second TFCI.

On présentera ci-dessous une description de la Règle 3 en référence à un exemple. Un code (9,3) et un code (23,7), ou un code (11,3) et un code (21,7) sont disponibles conformément au rapport de bits d'information du premier TFCI au second TFCI. Lorsque la Règle 3 est appliquée, deux symboles codés arbitraires du code (23,7) ne sont pas transmis, ce qui fait qu'un rendement de codage réel du code (23,7) devient (21,7); et deux symboles codés arbitraires du code (11,3) ne sont pas transmis, ce qui fait qu'un rendement de codage réel du code (11,3) devient (9,3). Les symboles codés arbitraires peuvent être sélectionnés de façon que les performances de code réelles ne soient pas diminuées, bien que les rendements de codage réels des codes respectifs soient diminués. Cependant, la plupart des codes présentent une diminution de performances. Dans la Règle 3, le rendement de codage réel des codes respectifs devient (9,3) ou

(21,7), ce qui respecte les performances des mots de code TFCI avec un rendement de données réel 1/3 pour le cas de la condition A1. Cependant, l'augmentation du nombre des symboles TFCI occasionne une diminution des performances du code pour lequel le nombre des symboles codés a augmenté, alors qu'on désirait augmenter les performances du premier code TFCI ou du second code TFCI. La Règle 3 peut rechercher les symboles arbitraires qui ne diminuent pas les performances des codes. Comme la Règle 2, la Règle 3 a également un procédé d'application complexe. Pour simplifier le procédé d'application, les deux derniers symboles parmi les symboles codés émis par le codeur ayant le nombre accru de symboles codés font l'objet d'une application vers $d_{30}(b_{30})$ et $d_{31}(b_{31})$.

On présentera ci-dessous une description de la Règle 4 en référence à un exemple. Un code (23,7) et un code (9,3), ou un code (21,7) et un code (11,3) sont disponibles conformément au rapport de bits d'information du premier TFCI au second TFCI. Le code (21,7) et le code (11,3) sont utilisés pour augmenter le rendement de codage du second TFCI, tandis que le code (23,7) et le code (9,3) sont utilisés pour augmenter le rendement de codage du premier TFCI. Lorsque la Règle 4 est appliquée, les deux derniers symboles codés du code (23,7) ne sont pas transmis, ce qui fait qu'un rendement de codage réel du code (23,7) devient (21,7) et le rendement de codage du code (9,3) reste inchangé; et les deux derniers symboles codés du code (21,7) ne sont pas transmis, ce qui fait qu'un rendement de codage réel du code (21,7) devient (19,7) et le rendement de codage du code (11,3) reste inchangé. Lorsque la Règle 4 est appliquée, les deux derniers symboles ou deux symboles arbitraires provenant des codeurs respectifs ayant le grand nombre de mots de code sont appliqués vers $d_{30}(b_{30})$ et $d_{31}(b_{31})$. Dans la Règle 4, deux symboles codés du code ayant un mot de code plus long ne sont pas transmis, ce qui fait que les performances

du code ayant le mot de code plus long sont réduites, mais les performances du code ayant le mot de code plus court sont assurées.

On présentera ci-dessous une description de la Règle 5 en référence à un exemple. Si on suppose que le rapport entre le débit binaire d'information du premier TFCI et celui du second TFCI est 3:7 et les performances du mot de code transmettant le second TFCI sont augmentées, alors un code (9,3) et un code (23,7) sont disponibles. Dans la Règle 5, dans le but de transmettre le second TFCI avec une fiabilité élevée, deux symboles codés arbitraires du code (9,3) sont appliqués vers $d_{30}(b_{30})$ et $d_{31}(b_{31})$, ce qui fait que le rendement de codage réel devient (7,3). Dans la Règle 5, les performances d'un codeur du premier TFCI sont réduites, mais les symboles codés pour le second TFCI ne sont pas endommagés, ce qui fait qu'il est possible de transmettre de façon sûre le second mot de code TFCI.

Dans la description précédente de la Règle 1, de la Règle 2, de la Règle 3 et de la Règle 4, les symboles c_x^1 et c_y^2 sont appliqués vers les b_1 seulement dans la condition A1. Cependant, dans la condition A2, A3 et A4, les 32 symboles codés sont tous transmis, ou bien les 32 symboles codés sont transmis par répétition, ce qui fait qu'une règle d'application séparée n'est pas exigée, et les règles d'application intactes utilisées pour la condition A1 peuvent être utilisées. De plus, la Règle 1, la Règle 2, la Règle 3, la Règle 4 et la Règle 5 peuvent être utilisées de façon appropriée en fonction des circonstances.

La présente invention présentera ici un procédé d'application des symboles c_x^1 et c_y^2 vers les bits b_1 , à titre d'exemple. Dans l'exemple suivant, le procédé applicable à la Règle 1 et le procédé consistant à arranger aussi uniformément que possible les premiers symboles TFCI codés et les seconds symboles TFCI codés, pour obtenir un gain de transmission temporelle, peuvent également être appliqués à

un autre procédé d'application. Dans la condition A1, les derniers symboles codés pour les symboles c_x^1 et c_y^2 sont appliqués vers b_{30} ou b_{31} .

5 Parmi les codeurs que procure l'invention, les 16 codeurs augmentant le rendement de codage du premier codeur de TFCI ou du second codeur de TFCI, bien qu'ils aient le rendement de codage de $1/3$, sont conçus pour avoir des performances optimales au rendement de codage de $1/3$.

10 Avant une description du procédé d'application des symboles codés c_x^1 et c_y^2 vers les bits codés b_1 , le nombre des premiers symboles TFCI codés c_x^1 est défini comme étant n (avec $n = x+1$), et le nombre des seconds symboles TFCI codés c_y^2 est défini comme étant m (avec $m = y+1$). Pour la commodité de l'explication, on supposera que n est égal ou
15 inférieur à m , et que la somme de n et m devient 32. Ainsi, pour $n = 4, 7, 10, 13$ et 16 , $m = 28, 25, 22, 19$ et 16 , respectivement. Les valeurs n et m sont définies de la façon suivante :

$$b_{\left[\frac{32}{n}x(i+1)\right]-1} = c_i^1 \quad (0 \leq i \leq n - 1) \quad (1)$$

$$20 \quad b_{i + \left[\frac{n}{32-n}x(i + \frac{1}{2})\right]} = c_i^2 \quad (0 \leq i \leq m - 1) \quad (2)$$

Dans l'Equation (1), n indique le nombre total des premiers symboles TFCI codés, et i indique des index des premiers symboles TFCI codés, avec $0 \leq i \leq n-1$ (ou x). Les index sont assignés dans l'ordre de génération. L'équation
25 (1) représente les positions des bits b_1 vers lesquels les premiers symboles TFCI codés doivent être appliqués. Dans l'Equation (1), $[x]$ indique un entier obtenu en arrondissant une valeur donnée x .

Dans l'Equation (2), n indique le nombre total des
30 premiers symboles TFCI codés, m indique le nombre total des seconds symboles TFCI codés, et i indique des index des seconds symboles TFCI codés, avec $0 \leq i \leq m-1$ (ou y). Les index sont assignés dans l'ordre de génération. L'équation (2)

représente les positions des bits b_1 vers lesquels les seconds symboles TFCI codés doivent être appliqués. Dans l'Equation (2), $[x]$ indique une valeur maximale parmi les entiers inférieurs ou égaux à une valeur donnée x .

5 Les premiers symboles TFCI codés sont appliqués (arrangés) conformément à l'Equation (1), tandis que les seconds symboles TFCI codés sont appliqués conformément à l'Equation (2). En ce qui concerne l'ordre des symboles codés, les premiers symboles TFCI codés ou les seconds symboles TFCI codés peuvent être arrangés en premier. Selon une
10 variante, les premiers symboles TFCI codés et les seconds symboles TFCI codés peuvent être arrangés simultanément.

Lorsque le nombre des premiers symboles TFCI codés est plus grand que le nombre des seconds symboles TFCI codés
15 ($n > m$), l'Equation (2) est utilisée pour effectuer l'application des premiers symboles TFCI codés, et l'Equation (1) est utilisée pour effectuer l'application des seconds symboles TFCI codés.

Le Tableau 6 ci-dessous illustre les symboles codés
20 générés conformément à l'Equation (1) et l'Equation (2). Dans le Tableau 6, "0" indique les positions auxquelles les premiers symboles TFCI codés c_x^1 sont transmis, et "1" indique les positions auxquelles les seconds symboles TFCI codés c_y^2 sont transmis.

25 Tableau 6

Code	0	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	
4,1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	
7,2	1	1	1	0	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	1	0	1	1	1	1	1	0	
10,3	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	
13,4	1	0	1	0	1	1	0	1	0	1	1	0	1	0	1	1	0	1	0	1	1	0	1	0	1	1	0	1	0	1	1	0	
16,5	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	
19,6	0	1	0	1	0	0	1	0	1	0	0	1	0	1	0	0	1	0	1	0	0	1	0	1	0	0	1	0	1	0	0	1	
22,7	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	1	
25,8	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	
28,9	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Le Tableau 6 montre un exemple d'arrangement des premiers symboles TFCI codés et des seconds symboles TFCI codés. Pour sélectionner les positions pour transmettre sur un canal physique les symboles codés générés conformément au

5 Tableau 6, les différents procédés sont utilisés dans les conditions A1, A2, A3 et A4. Dans la condition A1, les symboles codés c_x^1 et c_y^2 appliqués vers le bit b_{30} ou b_{31} sont poinçonnés, et ensuite 30 bits b_1 sont appliqués vers les bits d_m avant d'être transmis. Dans la condition A2, les 32

10 bits b_1 appliqués dans la condition A1 sont répétés séquentiellement 3 fois, répétés une fois de plus de b_0 à b_{23} , et ensuite appliqués vers un total de 120 bits d_m avant d'être transmis. Dans la condition A3, les 32 bits b_1 arrangés dans la condition A1 sont appliqués vers les positions des bits

15 de transmission d_m avant d'être transmis. Dans la condition A4, les 32 bits b_1 appliqués dans la condition A1 sont répétés 4 fois, et sont ensuite appliqués vers les positions des 128 bits de transmission d_m avant être transmis.

La figure 6 illustre une structure détaillée du dispositif d'arrangement de symboles codés 210 de la figure 2.

20 En se référant à la figure 6, on note que le numéro de référence 601 indique les seconds symboles TFCI codés c_y^2 émis par le codeur 200 de la figure 1, et le numéro de référence 611 indique les premiers symboles TFCI codés c_x^1 émis par le

25 codeur 205. Des dispositifs de stockage 603 et 613 sont des dispositifs pour stocker les symboles codés c_x^1 et c_y^2 et ils sont réalisés en utilisant une mémoire. Cependant, en modifiant la structure du matériel, il est possible de diriger directement les seconds symboles TFCI codés 601 et les

30 premiers symboles TFCI codés 611 vers un commutateur 622, sans stocker les symboles codés dans les dispositifs de stockage. Le commutateur 620 est commuté de manière alternée vers les dispositifs de stockage 603 et 613 conformément à une information de sélection de code reçue. Le symboles co-

35 dés c_x^1 et c_y^2 émis par les dispositifs de stockage 603 et

613 sont stockés dans un dispositif de stockage permanent 621. Une unité de commande 670 arrange les symboles codés reçus c_x^1 et c_y^2 conformément au Tableau 6. L'arrangement des symboles peut être réalisé par matériel ou par logiciel.

5 Un commutateur 630 commute les bits b_1 stockés dans le dispositif de stockage 621 vers un noeud de sortie ou un répéteur 640, conformément à une information concernant le nombre des bits TFCI codés, c'est-à-dire l'information sur les bits d_m . Ainsi, dans la condition A1 ou A3 dans laquelle le

10 nombre de bits d_m est 30 ou 32, le commutateur 630 commute les bits b_1 à partir du dispositif de stockage 621 vers le noeud de sortie. Dans la condition A3 ou A4 dans laquelle le nombre de bits d_m est 120 ou 128, le commutateur 630 commute les bits b_1 à partir du dispositif de stockage 621 vers le

15 répéteur 640. Le répéteur 640 répète un nombre de fois prédéterminé les bits b_1 provenant du commutateur 630, afin d'obtenir les bits d_m pour la condition A2 ou A4. Le répéteur 640 est activé dans les conditions A2 et A4. Le répéteur 640 peut également être réalisé par logiciel dans

20 l'unité de commande 670.

Les bits b_1 arrangés par le dispositif d'arrangement de symboles codés 210 conformément au Tableau 6 sont fournis à un multiplexeur 220, dans lequel ils sont multiplexés dans le temps avec une information physique telle que les bits

25 TPC et des bits Pilotes transmis sur le DPCCH et le DPDCH. Le multiplexeur 220 génère le DPCH, dont la structure est illustrée sur la figure 5. La figure 5 illustre une structure du DPCH transmis à partir d'un Noeud B vers un équipement d'utilisateur (UE).

30 En se référant à la figure 5, on note que le numéro de référence 510 désigne une structure d'une trame radio comprenant 15 créneaux temporels. Le numéro de référence 520 désigne une structure d'un créneau temporel d'un canal de liaison descendante dans lequel le DPDCH et le DPCCH sont

35 séparés par répartition dans le temps. Ainsi, le créneau

temporel est constitué de deux champs de données 501 et 507 constituant le DPDCH, et d'un champ TPC 503, d'un champ TFCI 505 et d'un champ Pilote 509 constituant le DPCCH. Le champ TPC 503 est utilisé pour transmettre un ordre de TPC pour un canal de liaison montante à partir de l'UE vers le Noeud B, et le champ Pilote 509 est utilisé pour estimer un changement dans le canal de liaison montante et le niveau de signal par l'UE. En outre, le champ TFCI 505 est utilisé pour transmettre vers l'UE les symboles de transmission TFCI codés, d_m , émis par le dispositif d'arrangement de symboles codés 210.

Le DPCH émis par le multiplexeur 220 est fourni à un dispositif d'étalement 230, et en même temps un code d'étalement pour la séparation de canaux est fourni au dispositif d'étalement 230 par un générateur de code d'étalement 235. Le dispositif d'étalement 230 effectue un étalement de canal du DPCH avec le code d'étalement, dans une unité de symbole, et il émet le DPCH à canal étendu, dans une unité de bribe. Le DPCH à canal étendu est appliqué à un embrouilleur 240 et, en même temps, un code d'embrouillage est fourni à l'embrouilleur 240 par un générateur de code d'embrouillage 245. L'embrouilleur 240 embrouille avec le code d'embrouillage le DPCH à canal étendu.

2. Second Mode de Réalisation d'Emetteur

La figure 13 illustre une structure d'un émetteur conforme à un second mode de réalisation de la présente invention. En se référant à la figure 13, on note qu'un codeur 1303 et un codeur 1313 codent respectivement des bits d'information TFCI pour le DSCH (seconds bits d'information TFCI) et des bits d'information TFCI pour le DCH (premiers bits d'information TFCI). Le codeur 1303 et le codeur 1313 ont la même structure que le codeur illustré sur la figure 4, à l'exception du fait que le poinçonneur et l'unité de commande ne sont pas incorporés. Les 32 symboles codés émis par le codeur 1303 sont fournis à un second dispositif de

stockage de symboles TFCI codés, 1305, et les 32 symboles codés émis par le codeur 1313 sont fournis à un premier dispositif de stockage de symboles TFCI codés, 1315. Le premier dispositif de stockage de symboles TFCI codés 1315 et le second dispositif de stockage de symboles TFCI codés 1306 peuvent partager la même mémoire. Dans ce cas, les premiers symboles TFCI codés et les seconds symboles TFCI codés doivent pouvoir être distingués d'un point de vue logique. Le second dispositif de stockage de symboles TFCI codés 1305 et le premier dispositif de stockage de symboles TFCI codés 1315 fournissent à un dispositif d'arrangement de symboles 1350 les symboles codés parmi les 32 symboles codés stockés à l'intérieur, sélectionnés conformément à la seconde information de sélection de symboles codés 1331 et à la première information de sélection de symboles codés 1333, reçues d'une unité de commande 1330, respectivement. La seconde information de sélection de symboles codés 1331 et la première information de sélection de symboles codés 1333 sont la même information que la configuration de poinçonnage illustrée dans le Tableau 5, et elles sont utilisées pour sélectionner des symboles codés désirés parmi les 32 symboles codés, au lieu de poinçonner les symboles codés conformément à la configuration de poinçonnage. Les signaux de sortie du second dispositif de stockage de symboles TFCI codés 1305 et du premier dispositif de stockage de symboles TFCI codés 1315, sont respectivement égaux à c_y^2 et c_x^1 . Le dispositif d'arrangement de symboles 1350 arrange les seconds symboles TFCI codés et les premiers symboles TFCI codés reçus sous la forme du Tableau 6, conformément à l'information d'arrangement de symboles codés 1335 reçue de l'unité de commande 1330. Un signal de sortie du dispositif d'arrangement de symboles 1350 devient b_1 . L'unité de commande 1330 de la figure 13 commande les dispositifs de stockage de symboles 1305 et 1315 et le dispositif d'arrangement de symboles 1350 conformément à la configuration de poinçonnage de symboles

du Tableau 5 et à la configuration d'arrangement de symboles du Tableau 6, respectivement, pour fournir la même information de sortie que le codeur et le sélecteur de symboles illustrés sur les figures 4, 6 et 8.

5 La figure 19 montre une structure détaillée du dispositif d'arrangement de symboles codés 1350. En se référant à la figure 19, on note que le dispositif d'arrangement de symboles codés comprend un dispositif de stockage 1901, une unité de commande 1910 et un commutateur. Le dispositif de
10 stockage 1901, qui est un dispositif destiné à stocker les premiers symboles TFCI codés et les seconds symboles TFCI codés sous la forme du Tableau 6, arrange les premiers symboles TFCI codés et les seconds symboles TFCI codés sous la commande de l'unité de commande 1910, et il émet ensuite sé-
15 quentiellement les bits b_1 . L'unité de commande 1910 commande le commutateur pour fournir au dispositif de stockage 1901 les premiers symboles TFCI codés et les seconds symboles TFCI codés, et elle commande également le dispositif de
20 stockage 1901 pour réarranger sous la forme du Tableau 6 les premiers symboles TFCI codés et les seconds symboles TFCI codés. L'unité de commande 1910 de la figure 19 peut être réalisée par logiciel. Dans ce cas, le logiciel peut devenir une unité de commande d'adresse. Selon une variante, le dis-
25 positif d'arrangement de symboles 1350, le premier dispositif de stockage de symboles TFCI codés 1315 et le second dispositif de stockage de symboles TFCI codés 1305 peuvent être réalisés soit dans la même mémoire, soit dans des mé-
30 moires différentes. Cependant, lorsque l'unité de commande 1330 est réalisée par logiciel, elle commande des adresses pour les mémoires du dispositif d'arrangement de symboles 1350, du premier dispositif de stockage de symboles TFCI co-
dés 1315 et du second dispositif de stockage de symboles TFCI codés 1305, pour accomplir ainsi par logiciel l'opéra-
tion des codeurs et du dispositif d'arrangement de symboles.

3. Troisième Mode de Réalisation d'Emetteur

La figure 3 montre la structure d'un émetteur conforme à un troisième mode de réalisation de la présente invention. L'émetteur code les premiers symboles TFCI codés et les seconds symboles TFCI codés en utilisant un seul codeur.

En se référant à la figure 3, on note que les seconds bits TFCI codés 301 et les premiers bits TFCI codés 303 sont appliqués à un sélecteur 310. Le sélecteur 310 fournit sélectivement à un codeur 311 les seconds bits TFCI codés 301 ou les premiers bits TFCI codés 303, conformément à une information de sélection de TFCI provenant d'une unité de commande 330. Une structure détaillée du sélecteur 310 est illustrée à titre d'exemple sur la figure 7. En se référant à la figure 7, on note que les seconds bits TFCI 301 sont fournis à un dispositif de stockage 703 et les premiers bits TFCI 303 sont fournis à un dispositif de stockage 713. Les dispositifs de stockage 703 et 713, qui sont des dispositifs destinés à stocker les seconds bits TFCI 301 et les premiers bits TFCI 303, peuvent être réalisés par des mémoires. Cependant, en modifiant la structure du matériel, il est possible d'appliquer directement les seconds bits TFCI 301 et les premiers bits TFCI 303 à un commutateur 720, sans utiliser les dispositifs de stockage. Le commutateur 720 est commuté en alternance vers les dispositifs de stockage 703 et 713 conformément à une information de sélection de code reçue. Les seconds bits TFCI et les premiers bits TFCI émis par le commutateur 720 sont fournis au codeur 311. Le sélecteur 310 peut également être réalisé par logiciel.

Le codeur 311 a la structure de la figure 4, et il code les bits TFCI provenant du sélecteur 310 conformément à une information de longueur de code reçue de l'unité de commande 330. L'unité de commande 330 peut également être réalisée par logiciel.

Les symboles codés c_x^1 ou c_y^2 émis par le codeur 311

sont fournis à un dispositif d'arrangement de symboles 312, dans lequel ils sont arrangés sous la forme illustrée dans le Tableau 6. La figure 8 montre une structure interne du dispositif d'arrangement de symboles 312.

5 En se référant à la figure 8, on note qu'un dispositif de stockage 801 arrange les symboles TFCI codés reçus sous la forme illustrée dans le Tableau 6, sous la commande d'une unité de commande 810. Parmi les symboles codés c_x^1 ou c_y^2 , les premiers symboles TFCI codés reçus sont stockés
10 dans le dispositif de stockage 801 jusqu'à ce que les autres symboles TFCI codés soient complètement arrangés. Le dispositif de stockage 801 fournit des bits b_1 à un commutateur 803. Le commutateur 803 émet les symboles TFCI codés inchangés provenant du dispositif de stockage 801, ou émet vers un
15 répéteur 805 les symboles TFCI codés, conformément à une information concernant le nombre de transmissions de symboles TFCI codés. Le répéteur 805 répète les symboles TFCI codés fournis par le commutateur 803, un nombre de fois égal au nombre de symboles TFCI codés d_m devant être transmis sur le
20 canal physique. Le répéteur 805 peut être réalisé par logiciel pour accomplir la même opération. Le répéteur 805 peut être réalisé soit sous la forme d'un bloc interne de l'unité de commande 810, soit sous la forme d'un bloc séparé.

Les symboles TFCI codés d_m émis par le dispositif
25 d'arrangement de symboles 312 sont appliqués à un multiplexeur 313, dans lequel ils sont multiplexés dans le temps avec l'information physique telle que les bits TPC et Pilotes, transmis sur le DPCCH et le DPDCH. Le DPCH multiplexé a la structure illustrée sur la figure 5.

30 Le DPCH est fourni à un dispositif d'étalement 314 et, en même temps, un code d'étalement généré par un générateur de code d'étalement 316 est fourni au dispositif d'étalement 314. Le dispositif d'étalement 314 effectue un étalement de canal du DPCH avec le code d'étalement, dans une
35 unité de symboles, pour la séparation des canaux, et il émet

le DPCH avec étalement de canal dans une unité de bribe. Le DPCH avec étalement de canal est fourni à un embrouilleur 315, et en même temps, un code d'embrouillage généré par un générateur de code d'embrouillage 317 est fourni à l'embrouilleur 315. L'embrouilleur 315 embrouille avec le code d'embrouillage le DPCH avec étalement de canal.

4. Quatrième Mode de Réalisation d'Émetteur

La figure 14 illustre une structure d'un émetteur conforme à un quatrième mode de réalisation de la présente invention. L'émetteur de la figure 14 diffère de l'émetteur de la figure 13 par le fait qu'un codeur code séquentiellement les premiers bits d'information TFCI et les seconds bits d'information TFCI en utilisant un seul codeur. En se référant à la figure 14, on note que les premiers bits d'information TFCI ou les seconds bits d'information TFCI sont fournis à un codeur 1403, dans lequel ils sont codés et ensuite fournis à un dispositif de stockage de symboles codés 1405. Le dispositif de stockage de symboles codés 1405 sélectionne les symboles codés conformément à une information de sélection de symboles codés 1401, c'est-à-dire la configuration de poinçonnage du Tableau 5, reçue à partir d'une unité de commande 1430, et il fournit les symboles codés sélectionnés à un sélecteur de code (ou dispositif d'arrangement de code) 1450. Le dispositif de stockage de symboles codés 1405 peut fournir directement les premiers symboles TFCI codés ou les seconds symboles TFCI codés au dispositif d'arrangement de code 1450. Selon une variante, le codeur 1403 reçoit les autres symboles TFCI codés, et le dispositif de stockage de symboles codés 1405 sélectionne les symboles TFCI codés reçus conformément à l'information de sélection de symboles codés 1401 provenant de l'unité de commande 1430, et fournit les deux types des symboles TFCI codés au dispositif d'arrangement de code 1450. Le sélecteur de code 1450 de la figure 14 effectue une application vers les bits b_1 des symboles codés c_x^1 ou c_y^2 reçus sous la forme du Ta-

bleau 6. Le dispositif de stockage de symboles codés, le dispositif d'arrangement de code et l'unité de commande peuvent également être réalisés par logiciel.

5. Cinquième Mode de Réalisation d'Emetteur

5 La figure 15 illustre une structure d'un émetteur conforme à un cinquième mode de réalisation de la présente invention. Contrairement aux autres émetteurs, l'émetteur de la figure 15 effectue simultanément le codage de TFCI et l'arrangement de symboles.

10 On décrira le fonctionnement de l'émetteur en se référant à un exemple dans lequel les seconds bits TFCI sont codés à un rendement de (4,1), les premiers bits TFCI sont codés à un rendement de (28,9), et les symboles codés sont arrangés conformément aux bits b_1 .

15 En se référant à la figure 15, on note qu'une structure de stockage de mots de code de base 1501 stocke les mots de code de base W_1 , W_2 , W_4 , W_8 , W_{16} , M_1 , M_2 , M_4 , M_8 , et une séquence ne contenant que des 1, utilisés dans le codeur de la figure 4. Dans la structure de stockage de mots de code de base 1501, un axe horizontal représente les mots de code de base de longueur 32, et un axe vertical représente des symboles des mots de code de base respectifs. Une unité de commande 1510 reçoit des seconds bits d'information TFCI 1511, des premiers bits d'information TFCI 1513, une information de sélection de symboles TFCI codés 1515 et une information d'arrangement de symboles TFCI codés 1517, elle commande la structure de stockage de mots de code de base 1501 pour générer un code (4,1) et un code (28,9), et elle arrange les codes pour obtenir un gain de transmission temporelle.

30 Si les seconds bits d'information TFCI 1511 sont définis comme a_0^2 et les premiers bits d'information TFCI sont définis comme a_0^1 , a_1^1 , a_2^1 , a_3^1 , a_4^1 , a_5^1 , a_6^1 , a_7^1 et a_8^1 , l'unité de commande 1510 répète 4 fois une opération de génération de 7 premiers symboles TFCI codés et de 1 second

35

symbole TFCI codé, conformément à l'information d'arrangement de symboles TFCI codés 1517 du Tableau 6, c'est-à-dire $c_0^1, c_1^1, c_2^1, c_3^1, c_4^1, c_5^1, c_6^1, c_7^1, c_8^1, c_9^1, c_{10}^1, c_{11}^1, c_{12}^1, c_{13}^1, c_{14}^1, c_{15}^1, c_{16}^1, c_{17}^1, c_{18}^1, c_{19}^1, c_{20}^1, c_{21}^1, c_{22}^1,$
 5 $c_{23}^1, c_{24}^1, c_{25}^1, c_{26}^1, c_{27}^1,$ et c_3^2 .

Les premiers symboles TFCI codés et les seconds symboles TFCI codés utilisent des mots de code de base différents, conformément aux bits d'information TFCI reçus $a_0^1, a_1^1, a_2^1, a_3^1, a_4^1, a_5^1, a_6^1, a_7^1, a_8^1$ et a_0^2 . L'utilisation
 10 des mots de code de base sélectionnés est déterminée conformément au fait que les bits d'information d'entrée sont "0" ou "1", et les symboles sont sélectionnés conformément à la configuration de poinçonnage du Tableau 5.

Neuf premiers bits d'entrée TFCI sont reçus pour les
 15 premiers symboles TFCI codés, ce qui fait que le générateur de mots de code de base 1501 génère les mots de code de base W1, W2, W4, W8, W16, la séquence ne contenant que des 1, M1, M2 et M4. Un second bit d'entrée TFCI est reçu pour les seconds symboles TFCI codés, ce qui fait que le générateur de
 20 mots de code de base 1501 génère seulement le mot de code de base W1. Les premiers symboles TFCI codés ont une configuration de poinçonnage de
 $\{1,1,1,1,1,1,0,1,1,1,0,0,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1\}$, et les seconds symboles TFCI codés ont une configuration de
 25 poinçonnage de
 $\{1,0,1,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\}$.

Pour générer les seconds symboles TFCI codés, la structure de stockage de mots de code de base 1501 sélectionne des 0-ième, 2-ième, 4-ième et 6-ième symboles du mot
 30 de code de base W1. Pour générer les premiers symboles TFCI codés, la structure de stockage de mots de code de base 1501 combine par une fonction OU-EX les mots de code de base W1, W2, W4, W8, W16, la séquence ne contenant que des 1, M1, M2 et M4, et elle sélectionne ensuite les symboles autres que
 35 les 6-ième, 10-ième, 11-ième et 13-ième symboles du mot de

code résultant.

On a décrit le fonctionnement du codeur de TFCI et du dispositif d'arrangement de symboles de la figure 15 en référence à un exemple de génération simultanée du code (4,1) et du code (28,9). De plus, un autre procédé de génération de types différents de mots de code comporte un processus de sélection du type du mot de code de base à utiliser conformément au nombre de bits de code d'entrée, et de détermination de l'ordre de génération des symboles codés en utilisant la configuration d'arrangement de symboles codés du Tableau 6. En outre, le procédé comporte un processus de combinaison par une fonction OU-EX des mots de code de base conformément à l'ordre et aux valeurs des bits de code d'entrée, et de sélection des symboles codés conformément à la configuration de poinçonnage du Tableau 5. Un dispositif de stockage de symboles codés 1530 stocke les valeurs émises par la structure de stockage de mots de code de base 1501. Comme l'émetteur des figures 13 et 14, l'émetteur de la figure 15 peut également être réalisé par logiciel.

20 6. Premier Mode de Réalisation de Récepteur

La figure 9 illustre une structure d'un récepteur, correspondant aux émetteurs des figures 3 et 4, conforme à un mode de réalisation de la présente invention. En se référant à la figure 9, on note qu'un DPCH de liaison descendante est fourni à un désembrouilleur 940 et, en même temps, un code d'embrouillage généré par un générateur de code d'embrouillage 945 est fourni au désembrouilleur 940. Le désembrouilleur 940 désembrouille le DPCH de liaison descendante avec le code d'embrouillage. Le DPCH de liaison descendante désembrouillé est fourni à un dispositif de désétalement 930 et, en même temps, un code d'étalement généré par un générateur de code d'étalement 935 est fourni au dispositif de désétalement 930. Le dispositif de désétalement 930 effectue un désétalement du DPCH de liaison descendante désembrouillé, avec le code de désétalement, dans une unité de

symboles.

Les symboles de DPCH désétagés sont fournis à un démultiplexeur 920 dans lequel ils sont démultiplexés (séparés) en symboles TFCI codés et en autres signaux tels que le DPDCH, le TPC, les bits Pilotes. Les symboles TFCI codés sont fournis à un dispositif de réarrangement de symboles codés 910. Le dispositif de réarrangement de symboles codés 910 sépare les symboles TFCI codés en symboles codés pour DSCH (seconds symboles d'information TFCI) et en symboles codés pour DCH (premiers symboles d'information TFCI), conformément à une information de longueur de code et une information de position. L'information de longueur de code est une information de longueur de code basée sur un rapport des bits TFCI pour le DSCH et des bits TFCI pour le DCH. L'information de position est une information indiquant des positions des symboles codés pour le DSCH et des positions des symboles codés pour le DCH, illustrées dans le Tableau 6. Les seconds symboles TFCI codés et les premiers symboles TFCI codés séparés par le dispositif de réarrangement de symboles codés 910 sont respectivement fournis à un premier décodeur 900 et un second décodeur 905. Les décodeurs 900 et 905 déterminent des codes correspondants, conformément à l'information de longueur de code, et ils décodent respectivement, par les codes déterminés, les seconds symboles TFCI codés et les premiers symboles TFCI codés. Ainsi, le premier décodeur 900 décode les seconds symboles TFCI codés et émet des seconds bits TFCI (bits TFCI pour le DSCH), et le second décodeur 905 décode les premiers symboles TFCI codés et émet des premiers bits TFCI (bits TFCI pour le DCH).

Les figures 18A et 18B illustrent une structure détaillée du dispositif de réarrangement de symboles codés 910 conformément à différents modes de réalisation de la présente invention. En se référant à la figure 18A, on note que le dispositif de réarrangement de symboles codés est constitué d'un dispositif de stockage 1801, d'une unité de com-

mande 1810 et d'un commutateur. Le dispositif de stockage 1801, qui est un dispositif pour stocker les symboles TFCI codés reçus du démultiplexeur 920, sépare les premiers symboles TFCI codés et les seconds symboles TFCI codés, sous la
5 commande de l'unité de commande 1810. L'unité de commande 1810 commande le dispositif de stockage 1801 et le commutateur pour fournir les premiers symboles TFCI codés et les seconds symboles TFCI codés respectivement à des décodeurs 905 et 900. Selon une variante, lorsqu'un seul décodeur est
10 utilisé, l'unité de commande 1810 fournit séparément les deux types de symboles TFCI codés au décodeur unique. L'unité de commande 1810 peut être réalisée par logiciel. Dans ce cas, le logiciel peut devenir une unité de commande d'adresse.

15 En se référant à la figure 18B, on note que le dispositif de réarrangement de symboles codés est constitué d'un dispositif de stockage 1821, d'une unité de commande 1820, d'un générateur de masque 1830, d'un multiplieur 1815 et d'un multiplieur 1817. Le dispositif de stockage 1821 a
20 le même fonctionnement que le dispositif de stockage 1801 de la figure 18A. L'unité de commande 1820 commande le dispositif de stockage 1821 pour fournir au premier multiplieur 1815 et au second multiplieur 1817 les symboles TFCI codés provenant du démultiplexeur 920. En outre, l'unité de com-
25 mande 1820 commande le générateur de masque 1830 pour générer des masques pour séparer les premiers symboles TFCI codés et les seconds symboles TFCI codés. Les masques générés par le générateur de masque 1830 sont fournis au premier multiplieur 1815 et au second multiplieur 1817. Le premier
30 multiplieur 1815 multiplie les symboles TFCI codés provenant du dispositif de stockage 1821 par le masque correspondant et fournit en sortie les premiers symboles TFCI codés. Le second multiplieur 1817 multiplie les symboles TFCI codés provenant du dispositif de stockage 1821 par le masque
35 correspondant et fournit en sortie les seconds symboles TFCI

codés. Le générateur de masque 1830 stocke sous la forme d'un masque la configuration d'arrangement de symboles des premiers symboles TFCI codés et des seconds symboles TFCI codés, illustrée dans le Tableau 6, ou bien il génère les masques en utilisant les Equations (1) et (2). Les masques sont utilisés pour séparer les symboles TFCI codés provenant du démultiplexeur 920, pour donner les premiers symboles TFCI codés et les seconds symboles TFCI codés. Si chaque multiplieur parmi le multiplieur 1815 et le multiplieur 1817 peut fournir en sortie les deux types de symboles TFCI codés, un seul des deux multiplieurs est utilisé pour séparer les premiers symboles TFCI codés et les seconds symboles TFCI codés.

La figure 11 illustre une structure détaillée des décodeurs 900 et 905 de la figure 9. En se référant à la figure 11, on note que des symboles reçus $r(t)$ sont fournis à un dispositif d'insertion de zéros 1100, et en même temps une information de longueur de code est fournie à une unité de commande 1130. L'unité de commande 1130 détermine des positions de poinçonnage sur la base de l'information de longueur de code, et fournit au dispositif d'insertion de zéros 1100 une information de commande pour les positions de poinçonnage déterminées. L'information de longueur de code indique la longueur de code ou le rendement de codage utilisés dans le codeur, tandis que l'information de commande indique les positions de poinçonnage. Les positions de poinçonnage représentent les positions des symboles supprimés pour obtenir une longueur de symbole codé désirée, correspondant aux bits reçus à partir du codeur. Le Tableau 7 illustre les positions de poinçonnage stockées en association avec les longueurs de code.

Tableau 7

Information de Longueur de Code (Rendement de Codage)	Position de Poinçonnage
(3,1)	F_29
(4,1)	F_28
(5,1)	F_27
(6,2)	F_26
(7,2)	F_25
(8,2)	F_24
(9,3)	F_23
(10,3)	F_22
(11,9)	F_21
(12,4)	F_20
(13,4)	F_19
(14,4)	F_18
(18,6)	F_14
(19,6)	F_13
(20,6)	F_12
(21,7)	F_11
(22,7)	F_10
(23,7)	F_9
(24,8)	F_8
(25,8)	F_7
(26,8)	F_6
(27,9)	F_5
(28,9)	F_4
(29,9)	F_3

On suppose dans le Tableau 7 que l'information de longueur de code indique le rendement de codage utilisé dans le codeur. Un rendement de codage (k,n) indique que n bits d'entrée sont codés en k symboles, les symboles reçus ayant une longueur de codage k . En outre, F_n dans le Tableau 7 représente n positions de poinçonnage. Comme on peut le déterminer d'après le Tableau 7, l'information de commande (position de poinçonnage) permet au dispositif d'insertion de zéros 1100 de maintenir le nombre (32) de symboles de sortie, indépendamment de la longueur de code des symboles reçus.

En se référant au Tableau 7, on note que l'unité de commande 1130 émet de l'information sur 29 positions de poinçonnage pour un rendement de codage $(3,1)$, de l'information sur 28 positions de poinçonnage pour un rendement de codage $(4,1)$, de l'information sur 27 positions de poinçonnage pour un rendement de codage $(5,1)$, de l'information sur 26 positions de poinçonnage pour un rendement de codage $(6,2)$, de l'information sur 25 positions de poinçonnage pour un rendement de codage $(7,2)$, de l'information sur 24 positions de poinçonnage pour un rendement de codage $(8,2)$, de l'information sur 23 positions de poinçonnage pour un rendement de codage $(9,3)$, de l'information sur 22 positions de poinçonnage pour un rendement de codage $(10,3)$, de l'information sur 21 positions de poinçonnage pour un rendement de codage $(11,9)$, de l'information sur 20 positions de poinçonnage pour un rendement de codage $(12,4)$, de l'information sur 19 positions de poinçonnage pour un rendement de codage $(13,4)$, de l'information sur 18 positions de poinçonnage pour un rendement de codage $(14,4)$, de l'information sur 14 positions de poinçonnage pour un rendement de codage $(18,6)$, de l'information sur 13 positions de poinçonnage pour un rendement de codage $(19,6)$, de l'information sur 12 positions de poinçonnage pour un rendement de codage $(20,6)$, de l'information sur 11 positions de poinçonnage pour un rendement de codage $(21,7)$, de l'information sur 10 positions de

poinçonnage pour un rendement de codage (22,7), de l'information sur 9 positions de poinçonnage pour un rendement de codage (23,7), de l'information sur 8 positions de poinçonnage pour un rendement de codage (24,8), de l'information sur 7 positions de poinçonnage pour un rendement de codage (25,8), de l'information sur 6 positions de poinçonnage pour un rendement de codage (26,8), de l'information sur 5 positions de poinçonnage pour un rendement de codage (27,9), de l'information sur 4 positions de poinçonnage pour un rendement de codage (28,9), et de l'information sur 3 positions de poinçonnage pour un rendement de codage (28,9). Pour les cas respectifs, les positions de poinçonnage sont les mêmes que celles données dans la description des codeurs.

Le dispositif d'insertion de zéros 1100 insère des 0 dans les positions de poinçonnage des symboles reçus, conformément à l'information de commande, et il fournit ensuite en sortie un train de symboles de longueur 32. Le train de symboles est fourni à un transformateur de Hadamard rapide inverse (IFHT pour "inverse fast Hadamard transformer") 1120 et à des multiplieurs 1102, 1104 et 1106. Les symboles du train de symboles fourni aux multiplieurs 1102, 1104 et 1106 sont respectivement multipliés par des fonctions de masque M1, M2 et M15 générées par le générateur de masque 1110. Les symboles de sortie des multiplieurs 1102, 1104 et 1106 sont fournis à des commutateurs respectifs 1152, 1154 et 1156. A ce moment, l'unité de commande 1130 fournit aux commutateurs 1152, 1154 et 1156 une information de commande de commutateur indiquant l'utilisation / non-utilisation des fonctions de masque, sur la base de l'information de longueur de code. Par exemple, du fait que les codeurs (3,1), (4,1), (5,1), (6,2), (7,2), (8,2), (9,3), (10,3), (11,3), (12,4), (13,4), (14,4), (18,6), (19,6) et (20,6) n'utilisent pas les fonctions de masque, les commutateurs 1152, 1154 et 1156 sont tous déconnectés conformément à l'information de commande de commutateur. Cependant, du

fait que les codeurs (21,7), (22,7) et (23,7) utilisent une fonction de masque de base, seul le commutateur 1152 est connecté. De cette manière, l'unité de commande 1130 commande les commutateurs 1152, 1154 et 1156 conformément au nombre des fonctions de masque utilisées, sur la base du rendement de codage. Ensuite, chacun des IFHT 1120, 1122, 1124 et 1126 accomplit une transformation de Hadamard rapide inverse sur 32 symboles reçus à partir du dispositif d'insertion de zéros 1100, et il calcule des corrélations entre les symboles et tous les codes de Walsh qui peuvent être utilisés dans l'émetteur. En outre, les IFHT déterminent une corrélation la plus élevée parmi les corrélations, et un index du code de Walsh ayant la corrélation la plus élevée. Par conséquent, chacun des IFHT 1120, 1122, 1124 et 1126 fournit au comparateur de corrélation 1140 un index de la fonction de masque multipliée par le signal reçu, la corrélation la plus élevée, et un index du code de Walsh ayant la corrélation la plus élevée. Du fait que le signal qui est fourni à l'IFHT 1120 n'est multiplié par aucune des fonctions de masque, un identificateur de la fonction de masque devient "0". Le comparateur de corrélation 1140 détermine la corrélation la plus élevée en comparant les corrélations qui sont fournies par les IFHT, et il combine un index d'une fonction de masque ayant la corrélation la plus élevée, avec un index du code de Walsh.

7. Second Mode de Réalisation de Récepteur

La figure 10 illustre une structure d'un récepteur, correspondant aux émetteurs des figures 3 et 4, conforme à un autre mode de réalisation de la présente invention. En se référant à la figure 10, on note qu'un DPCH de liaison descendante est fourni à un désembrouilleur 1040 et, en même temps, un code d'embrouillage généré par un générateur de code d'embrouillage 1045 est fourni au désembrouilleur 1040. Le désembrouilleur 1040 désembrouille le DPCH de liaison descendante avec le code d'embrouillage. Le DPCH de liaison

descendante désemprouillé est fourni à un dispositif de désétalement 1030 et, en même temps, un code d'étalement généré par un générateur de code d'étalement 1035 est fourni au dispositif de désétalement 1030. Le dispositif de désétalement 1030 effectue un désétalement du DPCH de liaison descendante désemprouillé, avec le code d'étalement, en une unité de symboles.

Les symboles de DPCH désétales sont fournis à un démultiplexeur 1020 dans lequel ils sont démultiplexés (séparés) en symboles TFCI codés et en autres signaux tels que les DPDCH, TPC, bits Pilotes. Les symboles TFCI codés sont fournis à un dispositif de réarrangement de symboles codés 1010. Le dispositif de réarrangement de symboles codés 1010 sépare les symboles TFCI codés pour donner les symboles codés pour le DSCH (seconds symboles d'information TFCI) et les symboles codés pour le DCH (premiers symboles d'information TFCI), conformément à une information de longueur de code et une information de position. L'information de longueur de code est une information de commande de longueur de code basée sur un rapport entre les bits TFCI pour le DSCH et les bits TFCI pour le DCH. L'information de position est une information indiquant des positions des symboles codés pour le DSCH et des positions des symboles codés pour le DCH, illustrées dans le Tableau 6.

Le dispositif de réarrangement de symboles codés 1010 a la structure des figures 18A et 18B. Lorsqu'on utilise la structure des figures 18A et 18B, le dispositif de réarrangement de symboles codés 1010 doit émettre séquentiellement séparément les premiers symboles TFCI codés et les seconds symboles TFCI codés. Les seconds symboles TFCI codés et les premiers symboles TFCI codés séparés sont fournis séquentiellement à un décodeur 1000. Le décodeur 1000 décode les premiers symboles TFCI codés ou les seconds symboles TFCI codés, avec un code correspondant à l'information de commande (information de longueur de code) pour la lon-

gueur de code. Par conséquent, le décodeur 1000 fournit en sortie les premiers bits TFCI ou les seconds bits TFCI. Le décodeur 1000 a le même fonctionnement que le décodeur de la figure 11.

5 En outre, la présente invention procure un décodeur capable d'effectuer le décodage pour les rapports de bits d'information respectifs, qui correspond au codeur pour coder les codes avec les diverses longueurs.

10 On va maintenant décrire en détail ci-dessous le fonctionnement du décodeur conforme à un mode de réalisation de la présente invention. Lorsqu'il fonctionne comme un décodeur correspondant aux codeurs (6,2), (7,2) et (8,2), le décodeur utilise des IFHT pour un codeur de Walsh avec une longueur de 4. Lorsqu'il fonctionne comme un décodeur
15 correspondant aux codeurs (9,3), (10,3) et (11,3), le décodeur utilise des IFHT pour un codeur de Walsh avec une longueur de 8. Lorsqu'il fonctionne comme un décodeur correspondant aux codeurs (12,4), (13,4) et (14,4), le décodeur utilise des IFHT pour un codeur de Walsh avec une longueur
20 de 16. Lorsqu'il fonctionne comme un décodeur correspondant au codeur (16,5), le décodeur utilise des IFHT pour un codeur de Walsh avec une longueur de 16. Lorsqu'il fonctionne comme un décodeur correspondant aux codeurs (18,6), (19,6), (20,6), (21,7), (22,7), (23,7), (24,8), (25,8), (26,8),
25 (27,9), (28,9), (29,9) et (32,10), le décodeur utilise des IFHT pour un codeur de Walsh avec une longueur de 32. Pour cette opération, le décodeur doit avoir une structure d'IFHT capable de supporter les codes avec une longueur variable. Par conséquent, la présente invention procure un décodeur
30 ayant la structure d'IFHT capable de supporter les codes avec la longueur variable.

8. Fonctionnements de Modes de Réalisation

35 On décrira des fonctionnements du codeur, du décodeur, du dispositif d'arrangement de symboles et du dispositif de réarrangement de symboles en se référant aux figures

16 et 17.

La figure 16 illustre le fonctionnement du décodeur et du dispositif d'arrangement de symboles codés dans l'émetteur conforme à un mode de réalisation de la présente invention. En se référant à la figure 16, on note qu'à l'étape 1601 l'émetteur décide de coder les premiers bits TFCI (bits d'information TFCI pour le DCH) et les seconds bits TFCI (bits d'information TFCI pour le DSCH) dans le mode de division ferme ou HSM ("Hard Split Mode"). A l'étape 1602, le codeur reçoit les premiers bits TFCI codés et les seconds bits TFCI codés. A l'étape 1603, le codeur code les premiers bits TFCI codés (32 symboles codés) et les seconds bits TFCI codés (32 symboles codés) avec le procédé conforme à la présente invention. A l'étape 1604, le dispositif d'arrangement de symboles codés sélectionne les symboles codés ayant des performances optimales parmi les premiers symboles TFCI codés conformément à une configuration de sélection de code, et il sélectionne également les symboles codés ayant des performances optimales parmi les seconds symboles TFCI codés, conformément à une configuration de sélection de code. Les configurations de sélection de code sont identiques aux configurations de poinçonnage illustrées dans le Tableau 5. A l'étape 1605, le dispositif d'arrangement de symboles codés arrange les premiers symboles TFCI codés et les seconds symboles TFCI codés sélectionnés, conformément à une configuration d'arrangement de symboles, pour obtenir un gain de diversité temporelle optimal. La configuration d'arrangement de symboles est illustrée dans le Tableau 6. Comme décrit en relation avec la figure 15, les opérations des étapes 1603, 1604 et 1605 peuvent être effectuées en un seul processus. Après l'étape 1605, les bits b_1 sont finalement déterminés à l'étape 1606, ce qui achève le processus de codage et d'arrangement de symboles.

La figure 17 illustre le fonctionnement du décodeur et du dispositif de réarrangement de symboles codés dans le

récepteur conforme à un mode de réalisation de la présente invention. En se référant à la figure 17, on note qu'à l'étape 1701 le récepteur reçoit des symboles TFCI codés transmis par le champ TFCI dans le DPCCH de liaison descendante du DPCH de liaison descendante. A l'étape 1702, le décodeur insère des 0 dans les positions des seconds symboles TFCI codés, parmi les symboles TFCI codés reçus, conformément à une information de position des seconds symboles TFCI codés, et il génère un premier mot de code TFCI ayant 32 symboles codés. En outre, le décodeur insère des 0 dans les positions des premiers symboles TFCI codés parmi les symboles TFCI codés reçus, conformément à une information de position des premiers symboles TFCI codés, et il génère un second mot de code TFCI ayant 32 symboles codés. Comme décrit en relation avec les figures 18A et 18B, il est possible de séparer les premiers symboles TFCI codés et les seconds symboles TFCI codés en utilisant le masque. L'information de position des premiers symboles TFCI codés et des seconds symboles TFCI codés est identique à la configuration utilisée à l'étape 1604 de la figure 16. Des 0 sont insérés dans les positions des symboles non poinçonnés ou non sélectionnés dans le bit de garantir un fonctionnement correct du décodeur. A l'étape 1703, le décodeur calcule des corrélations du premier mot de code TFCI et du second mot de code TFCI générés. A l'étape 1704, le décodeur fournit en sortie des valeurs ou des index du premier mot de code TFCI et du second mot de code TFCI ayant la corrélation maximale. A l'étape 1705, le décodeur termine le processus de décodage du premier mot de code TFCI et du second mot de code TFCI.

La description précédente a été faite pour le procédé de décodage, le procédé d'application des symboles c_x^1 et c_y^2 vers les bits b_1 , et le procédé d'application des bits b_1 vers les bits d_m , dans le cas où la somme du nombre des premiers bits d'information TFCI et du nombre des seconds bits d'information TFCI dans le HSM devient 10. En outre, on

a décrit l'émetteur-récepteur, le codeur et le décodeur. De façon classique, si la somme du nombre des premiers bits d'information TFCI et du nombre des seconds bits d'information TFCI est inférieure à 10, le LSM est disponible mais le HSM n'est pas disponible. Ainsi, le HSM est disponible seulement lorsque le nombre des premiers bits d'information TFCI et le nombre des seconds bits d'information TFCI sont tous deux inférieurs à 5. De façon classique, seul le codeur (16,5) est utilisé dans le HSM. Par conséquent, le HSM n'est pas disponible lorsque le nombre des premiers bits d'information TFCI est supérieur à 5 ou le nombre des seconds bits d'information TFCI est supérieur à 5. Cependant, lorsque le nouveau codeur conforme à la présente invention, capable de générer 24 sortes des codes, est utilisé, il n'y a pas de restriction sur le nombre des bits d'information TFCI, ce qui permet de transmettre de façon fiable les bits d'information TFCI. Ainsi, il est possible de déterminer les codes avec lesquels les bits d'information TFCI doivent être codés. Par conséquent, il est possible de transmettre séparément le premier code TFCI ou le second code TFCI, ou de transmettre simultanément le premier code TFCI et le seconde code TFCI, ce qui assure une transmission fiable.

On présentera une description détaillée de l'invention en supposant que le codeur a la structure de la figure 4 et utilise la configuration de poinçonnage du Tableau 4. On peut également appliquer l'invention à un autre cas dans lequel le codeur a une structure différente et utilise une configuration de poinçonnage différente.

Exemple 1. Le Rapport entre les Premiers Bits d'Information TFCI et les seconds bits d'information TFCI est 2:6

Lorsque le rapport entre les premiers bits d'information TFCI et les seconds bits d'information TFCI est 2:6, le procédé HSM classique peut coder les premiers bits d'information TFCI avant la transmission, mais ne peut pas coder les seconds bits d'information TFCI. Cependant, lorsque le

codeur conforme à la présente invention est utilisé, les premiers bits d'information TFCI sont codés en 6 symboles, 7 symboles ou 8 symboles, et les seconds bits d'information TFCI sont codés en 18 symboles, 19 symboles ou 20 symboles.

5 La somme des premiers symboles TFCI codés et des seconds symboles TFCI codés, codés par le codeur conforme à la présente invention, devient un minimum de 24 et un maximum de 28. Lorsque la somme est inférieure à 32 qui est le nombre de bases des symboles codés, le procédé le plus simple pour
10 traiter les symboles est de transmettre seulement les 24 symboles ou les 28 symboles par transmission discontinue (DTX). Ce procédé contribue à la simplification mais ne peut pas transmettre une autre information dans la période de DTX, ce qui occasionne un gaspillage de ressources. De plus,
15 il n'est pas possible d'augmenter les performances de codage des premiers bits d'information TFCI et des seconds bits d'information TFCI, à cause de la transmission discontinue des symboles codés.

Dans le cas de l'Exemple 1, on peut changer le pro-
20 cédé de codage pour donner la priorité au premier TFCI afin d'augmenter la fiabilité ou les performances, pour donner la priorité au second TFCI afin d'augmenter la fiabilité ou les performances, ou pour augmenter les performances à la fois du premier TFCI et du second TFCI.

25 Si on donne la priorité au premier TFCI pour augmenter la fiabilité ou les performances, les seconds bits d'information TFCI sont codés en utilisant le codeur (18,6), le codeur (19,6) ou le codeur (20,6), et les premiers bits d'information TFCI sont codés en utilisant le codeur (14,4),
30 le codeur (13,4) ou le codeur (12,4). En outre, il y a un autre procédé consistant à coder les premiers bits d'information TFCI par le codeur (6,2), le codeur (7,2) ou le codeur (8,2), et à transmettre ensuite de façon répétée les premiers bits TFCI codés, pour augmenter ainsi la fiabilité
35 ou les performances. Dans le procédé d'augmentation des per-

formances ou de la fiabilité du premier code TFCI en codant les premiers bits d'information TFCI en utilisant le codeur (14,4), le codeur (13,4) ou le codeur (12,4), des 0 sont insérés dans les 2 bits autres que les 2 bits d'information réels, avant le codage. Après la répétition du premier TFCI, la somme des premiers symboles TFCI codés répétés et des seconds symboles TFCI codés peut dépasser 32. Si la somme des premiers symboles TFCI codés et des seconds symboles TFCI codés dépasse 32, le système n'est pas compatible avec le standard 3GPP, ce qui augmente la complexité du matériel. Au contraire, si la somme des premiers bits d'information TFCI et des seconds bits d'information TFCI est inférieure à 32, comme dans l'Exemple 1, il y a une moindre limitation sur la sélection de code, en comparaison avec le cas dans lequel la somme des premiers bits d'information TFCI et des seconds bits d'information TFCI est 10. En effet, lorsque la somme des premiers bits d'information TFCI et des seconds bits d'information TFCI est 10, il est nécessaire de sélectionner les codes dans lesquels la somme des symboles codés est 32. Cependant, si la somme des symboles codés est inférieure à 32 bien que le rendement de codage maximal soit utilisé pour les bits d'information donnés dans l'Exemple 1, le rendement de codage des bits d'information TFCI peut être déterminé de façon que les performances soient améliorées dans une condition dans laquelle la somme des symboles codés devient 32.

D'autre part, si la priorité est donnée au second TFCI dans l'Exemple 1 pour augmenter la fiabilité ou les performances, les premiers bits d'information TFCI sont codés en utilisant le codeur (6,2), le codeur (7,2) ou le codeur (8,2), et les seconds bits d'information TFCI sont codés en utilisant le codeur (26,8), le codeur (25,8) ou le codeur (24,8). Selon une variante, il est possible de coder les bits d'information en utilisant le codeur (20,6), le codeur (19,6) ou le codeur (18,6), et de transmettre ensuite de façon répétée les bits codés, pour augmenter ainsi la

fiabilité ou les performances. Après la répétition du second TFCI, la somme des premiers symboles TFCI codés et des seconds symboles TFCI codés et répétés peut dépasser 32. Cependant, si la somme des premiers symboles TFCI codés et des seconds symboles TFCI codés dépasse 32, le système n'est pas compatible avec le standard 3GPP.

Un procédé pour augmenter la fiabilité ou les performances à la fois du premier TFCI et du second TFCI dans l'Exemple 1 consiste à augmenter jusqu'à 3 le nombre des premiers bits d'information TFCI et jusqu'à 7 le nombre des seconds bits d'information TFCI, avant le codage. Ainsi, les premiers bits d'information TFCI sont codés par le codeur (9,3), le codeur (10,3) ou le codeur (11,3), et les seconds bits d'information TFCI sont codés par le codeur (23,7), le codeur (22,7) ou le codeur (21,7) avant d'être transmis. Ce procédé peut être utilisé seulement lorsque la somme des symboles codés ne dépasse pas 32. Lorsque la somme des symboles codés dépasse 32, le problème mentionné ci-dessus se produit. Un autre procédé consiste à coder les premiers bits d'information TFCI par le codeur (6,2), le codeur (7,2) ou le codeur (8,2), et les seconds bits d'information TFCI par le codeur (18,6), le codeur (19,6) ou le codeur (20,6), et à transmettre ensuite de façon répétée les bits codés. La somme des symboles codés transmis de façon répétée ne doit pas dépasser 32. Il y a 3 sortes de codeurs pour coder les premiers bits d'information TFCI, et il y a également 3 sortes de codeurs pour coder les seconds bits d'information TFCI. Parmi les codeurs, on sélectionne le codeur ayant les meilleures performances. En ce qui concerne le nombre des symboles répétés par les codeurs, les symboles du codeur sélectionné sont transmis de façon répétée un beaucoup plus grand nombre de fois.

Exemple 2. Le Rapport entre les Premiers Bits d'Information TFCI et les seconds bits d'information TFCI est 3:4

Lorsque le rapport entre les premiers bits d'infor-

mation TFCI et les seconds bits d'information TFCI est 3:4, c'est-à-dire lorsque le nombre des premiers bits d'information TFCI et le nombre des seconds bits d'information TFCI sont tous deux inférieurs à 5, le procédé HSM classique code en (16,5) les premiers bits d'information TFCI et les seconds bits d'information TFCI, séparément ou séquentiellement, avant la transmission. Cependant, lorsque le codeur conforme à la présente invention est utilisé, les premiers bits d'information TFCI sont codés en 9 symboles, 10 symboles ou 11 symboles, et les seconds bits d'information TFCI sont codés en 12 symboles, 13 symboles ou 14 symboles. La somme des premiers symboles TFCI codés et des seconds symboles TFCI codés qui sont codés par le codeur conforme à la présente invention, devient égale au maximum à 25. Lorsque la somme est inférieure à 32, qui est le nombre de base des symboles codés, le procédé le plus simple pour traiter les symboles est de transmettre seulement les 21 symboles ou 24 symboles par transmission discontinue (DTX). Ce procédé contribue à la simplification, mais ne peut pas transmettre une autre information dans la période DTX, ce qui occasionne un gaspillage de ressources. De plus, il n'est pas possible d'augmenter les performances de codage des premiers bits d'information TFCI et des seconds bits d'information TFCI, à cause de la transmission discontinue des symboles codés.

Dans le cas de l'Exemple 2, le procédé de codage peut être changé pour donner priorité au premier TFCI pour augmenter la fiabilité ou les performances, pour donner priorité au second TFCI pour augmenter la fiabilité ou les performances, ou pour augmenter les performances à la fois du premier TFCI et du second TFCI.

Si on donne priorité au premier TFCI pour augmenter la fiabilité ou les performances, les seconds bits d'information TFCI sont codés en utilisant le codeur (12,4), le codeur (13,4) ou le codeur (14,4), et les premiers bits d'information TFCI sont codés en utilisant le codeur (20,6), le

codeur (19,6) ou le codeur (18,6). En outre, il y a un autre procédé consistant à coder les premiers bits d'information TFCI par le codeur (9,3), le codeur (10,3) ou le codeur (11,3), et à transmettre ensuite de façon répétée les premiers bits TFCI codés, pour augmenter ainsi la fiabilité ou les performances. Dans le procédé d'augmentation des performances ou de la fiabilité du premier code TFCI en codant les premiers bits d'information TFCI en utilisant le codeur (20,6), le codeur (19,6) ou le codeur (18,6), des 0 sont insérés dans les 3 bits autres que les 3 bits d'information réels avant d'être codés. Après la répétition du premier TFCI, la somme des premiers symboles TFCI codés répétés et des seconds symboles TFCI codés peut dépasser 32. Si la somme des premiers symboles TFCI codés et des seconds symboles TFCI codés dépasse 32, le système n'est pas compatible avec le standard 3GPP, ce qui augmente la complexité du matériel. Au contraire, si la somme des premiers bits d'information TFCI et des seconds bits d'information TFCI est inférieure à 32, comme dans l'Exemple 2, il y a une moindre limitation sur la sélection de code, en comparaison avec le cas dans lequel la somme des premiers bits d'information TFCI et des seconds bits d'information TFCI est 10. En effet, lorsque la somme des premiers bits d'information TFCI et des seconds bits d'information TFCI est 10, il est nécessaire de sélectionner les codes dans lesquels la somme des symboles codés est 32. Cependant, si la somme des symboles codés est inférieure à 32 bien que le rendement de codage maximal soit utilisé pour les bits d'information donnés dans l'Exemple 2, le rendement de codage des bits d'information TFCI peut être déterminé de façon que les performances soient améliorées dans une condition dans laquelle la somme des symboles codés devient 32.

D'autre part, si on donne priorité au second TFCI dans l'Exemple 2 pour augmenter la fiabilité ou les performances, les premiers bits d'information TFCI sont codés en

utilisant le codeur (9,3), le codeur (10,3) ou le codeur (11,3), et les seconds bits d'information TFCI sont codés en utilisant le codeur (23,7), le codeur (22,7) ou le codeur (21,7). Selon une variante, il est possible de coder les bits d'information en utilisant le codeur (14,4), le codeur (13,4) ou le codeur (12,4), et de transmettre ensuite de façon répétée les bits codés, pour améliorer ainsi la fiabilité ou les performances. Après la répétition du second TFCI, la somme des premiers symboles TFCI codés et des seconds symboles TFCI codés et répétés peut dépasser 32. Cependant, si la somme des premiers symboles TFCI codés et des seconds symboles TFCI codés dépasse 32, le système n'est pas compatible avec le standard 3GPP.

Enfin, un procédé pour augmenter la fiabilité ou les performances à la fois du premier TFCI et du second TFCI dans l'Exemple 2 consiste à augmenter le nombre des premiers bits d'information TFCI et des seconds bits d'information TFCI, de façon que la somme des premiers bits d'information TFCI et des seconds bits d'information TFCI devienne 10, et à utiliser un codeur approprié pour le nombre accru de bits d'information. Par exemple, il est possible d'utiliser un procédé de codage des premiers bits d'information TFCI par le codeur (24,4), le codeur (13,4) ou le codeur (12,4), et des seconds bits d'information TFCI par le codeur (18,6), le codeur (19,6) ou le codeur (20,6), avant la transmission. Ce procédé peut être utilisé seulement lorsque la somme des premiers bits d'information TFCI et des seconds bits d'information TFCI ne doit pas dépasser 10, et la somme des symboles codés ne doit pas dépasser 32. Lorsque la somme des symboles codés dépasse 32, le problème mentionné ci-dessus se produit. Un autre procédé consiste à coder les premiers bits d'information TFCI par le codeur (9,3), le codeur (10,3) ou le codeur (11,3), et les seconds bits d'information TFCI par le codeur (12,4), le codeur (13,4) ou le codeur (14,4), et à transmettre ensuite de façon répétée les

bits codés. La somme des symboles codés transmis de façon répétée ne doit pas dépasser 32. Il y a 3 sortes de codeurs pour coder les premiers bits d'information TFCI, et il y a également 3 sortes de codeurs pour coder les seconds bits d'information TFCI. Parmi les codeurs, on sélectionne le codeur ayant les meilleures performances. En ce qui concerne le nombre des symboles répétés par les codeurs, les symboles du codeur sélectionné sont transmis de façon répétée un beaucoup plus grand nombre de fois. De plus, il est possible de combiner le procédé de changement de rendement de codage et le procédé de transmission répétée, dans la transmission des premiers bits d'information TFCI et des seconds bits d'information TFCI avec une fiabilité ou des performances élevées.

On résumera ci-dessous des critères pour le procédé de sélection de code dans le HSM, décrit en relation avec l'Exemple 1 et l'Exemple 2.

Critère 1 : Le nombre de premiers bits d'information TFCI ou de seconds bits d'information TFCI dépasse 5 bits

- Si la priorité est donnée au premier TFCI, l'émetteur fixe le second codeur de TFCI et change ensuite un rendement de codage du premier TFCI pendant la transmission, ou code le premier TFCI en considérant le nombre de bits d'information réels et transmet ensuite de façon répétée les bits codés.

- Si la priorité est donnée au second TFCI, l'émetteur fixe le premier codeur de TFCI et change ensuite un rendement de codage du second TFCI pendant la transmission, ou code le second TFCI en considérant le nombre de bits d'information réels et transmet ensuite de façon répétée les bits codés.

- Si la priorité est donnée à la fois au premier TFCI et au second TFCI, l'émetteur effectue le codage en changeant les rendements de codage du premier TFCI et du second TFCI, ou en considérant le nombre de bits d'information

réels, et il transmet ensuite de façon répétée les bits codés. Il est possible de combiner le procédé de changement de rendement de codage et le procédé de transmission répétée.

Critère 2 : Le nombre des premiers bits d'information TFCI ou des seconds bits d'information TFCI ne dépasse pas 5 bits

- Le codeur code les premiers bits d'information TFCI et les seconds bits d'information TFCI en utilisant un codeur (16,5), avant la transmission.

10 - Le reste est le même que pour le Critère 1

On décrira en se référant à la figure 12 un procédé de sélection de code basé sur les critères ci-dessus, en utilisant la configuration de poinçonnage du Tableau 5 et le rendement de codage du Tableau 1.

15 En se référant à la figure 12, on note que la nécessité de transmettre le premier TFCI (premiers bits d'information) et le second TFCI (seconds bits d'information) apparaît à l'étape 1201. Ainsi, lorsqu'un Noeud B doit émettre le DSCH vers un UE, un émetteur reçoit le TFCI pour le DSCH et le TFCI pour le DCH. A l'étape 1202, on détermine si la
20 somme des premiers bits d'information et des seconds bits d'information est égale à 10. Si la somme des premiers bits d'information et des seconds bits d'information est égale à 10, l'émetteur détermine un code à utiliser pour les premiers bits d'information et les seconds bits d'information,
25 à l'étape 1208.

Le processus de sélection de code de l'étape 1208 sera décrit en référence au cas dans lequel le rapport entre les premiers bits d'information et les seconds bits d'information est 3:7. Dans ce cas, le codeur pour les premiers
30 bits d'information est un codeur (9,3), un codeur (10,3) ou un codeur (11,3), et le codeur pour les seconds bits d'information est un codeur (23,7), un codeur (22,7) ou un codeur (21,7). Ici, la somme des symboles codés doit être 32.
35 Le critère pour sélectionner les 3 sortes des rendements de

codage conformément aux types des bits d'information consiste à (1) donner la priorité aux premiers bits d'information pour ajouter 2 symboles en surplus, (2) donner la priorité aux seconds bits d'information pour ajouter 2 symboles en surplus, ou (3) ajouter un symbole en surplus à la fois aux premiers bits d'information et aux seconds bits d'information. Après avoir déterminé le rendement de codage à utiliser pour les premiers bits d'information et les seconds bits d'information à l'étape 1208, l'émetteur code les premiers bits d'information et les seconds bits d'information avec le rendement de codage déterminé, à l'étape 1209. A l'étape 1210, l'émetteur multiplexe les premiers symboles TFCI codés et les seconds symboles TFCI codés.

Cependant, s'il est déterminé à l'étape 1202 que la somme des premiers bits d'information et des seconds bits d'information est inférieure à 10, l'émetteur détermine à l'étape 1203 si le nombre des premiers bits d'information dépasse 5 ou si le nombre des seconds bits d'information dépasse 5. Si l'un quelconque du nombre de premiers bits d'information et du nombre de seconds bits d'information dépasse 5, l'émetteur passe à l'étape 1204. Cependant, si à la fois le nombre des premiers bits d'information et le nombre des seconds bits d'information ne dépassent pas 5, l'émetteur passe à l'étape 1221. A l'étape 1221, l'émetteur détermine s'il doit utiliser un codeur (16,5) pour coder les premiers bits d'information et les seconds bits d'information. Lorsque l'émetteur détermine qu'il n'utilise pas le codeur (16,5), il passe à l'étape 1206. Sinon, l'émetteur détermine qu'il utilise le codeur (16,5) et il passe à l'étape 1209.

A l'étape 1204, l'émetteur détermine s'il doit utiliser la transmission discontinue (DTX) pour transmettre les premiers bits d'information ou les seconds bits d'information. Lorsque l'émetteur détermine l'utilisation de la DTX, il passe à l'étape 1208. Sinon, lorsque l'émetteur détermine qu'il n'utilise pas la DTX, il passe à l'étape 1205.

On décrira le processus de l'étape 1208 en référence au cas dans lequel le rapport entre les premiers bits d'information et les seconds bits d'information est 3:4. Dans ce cas, le codeur pour les premiers bits d'information est l'un
5 sélectionné d'un codeur (9,3), d'un codeur (10,3) et d'un codeur (11,3), et le codeur pour les seconds bits d'information est l'un sélectionné d'un codeur (12,4), d'un codeur (13,4) et d'un codeur (14,4). A l'étape 1208, si la DTX est utilisée dans le cas où à la fois le nombre des premiers
10 bits d'information et le nombre des seconds bits d'information ne dépassent pas 5, il n'y a aucune restriction dans la sélection des codeurs, mais la somme des symboles codés ne doit pas dépasser 32.

A l'étape 1205, l'émetteur détermine s'il doit augmenter la fiabilité ou les performances à la fois du premier TFCI et du second TFCI avant la transmission. Si l'émetteur détermine qu'il doit augmenter la fiabilité ou les performances à la fois du premier TFCI et du second TFCI avant la transmission, il sélectionne le procédé d'augmentation de
20 rendement de codage ou le procédé de transmission répétée ou le procédé combiné des deux procédés ci-dessus, à l'étape 1207. A l'étape 1208, l'émetteur détermine le code à utiliser pour le premier TFCI et le second TFCI, conformément au procédé sélectionné à l'étape 1207. A l'étape 1209, le codeur code les premiers bits d'information TFCI et les seconds bits d'information TFCI conformément au procédé sélectionné, et ensuite à l'étape 1210 il multiplexe les premiers symboles TFCI codés et les seconds symboles TFCI codés. Si
25 l'émetteur détermine à l'étape 1207 qu'il doit augmenter la fiabilité ou les performances du premier TFCI et du second TFCI avant la transmission dans le procédé de transmission répétée, l'émetteur répète les premiers symboles TFCI codés et les seconds symboles TFCI codés, à l'étape 1209, et il les multiplexe ensuite à l'étape 1210. Dans le cas
30 contraire, l'émetteur répète à l'étape 1210 les premiers

symboles TFCI codés et les seconds symboles TFCI codés, qui ont été codés à l'étape 1209.

Si l'émetteur détermine à l'étape 1205 d'augmenter la fiabilité ou les performances du premier TFCI ou du second TFCI avant la transmission, l'émetteur sélectionne à l'étape 1206 le premier TFCI ou le second TFCI, auquel il donnera priorité. L'émetteur donne priorité au premier TFCI lorsque les premiers bits d'information TFCI doivent être transmis avec une fiabilité élevée, indépendamment du nombre des bits d'information. L'émetteur donne priorité au second TFCI lorsque les seconds bits d'information TFCI doivent être transmis avec une fiabilité élevée, en préparation pour le cas dans lequel des Noeuds B autres que le Noeud B recevant le DSCH ne peuvent pas transmettre les seconds bits d'information TFCI pour le DSCH lorsque l'UE est placé dans la région de transfert souple. De plus, l'émetteur donne priorité au second TFCI lorsque les seconds bits d'information TFCI doivent être transmis avec une fiabilité élevée, indépendamment du nombre des bits d'information. Si l'émetteur détermine d'augmenter la fiabilité ou les performances du premier TFCI ou du second TFCI avant la transmission, à l'étape 1206, l'émetteur détermine un procédé d'augmentation de la fiabilité ou des performances du premier TFCI ou du second TFCI avant la transmission, à l'étape 1207, en utilisant le procédé d'augmentation de rendement de codage, le procédé de transmission répétée ou le procédé combiné des deux procédés ci-dessus. A l'étape 1208, l'émetteur détermine le code à utiliser pour le premier TFCI et le second TFCI, conformément au procédé déterminé à l'étape 1207. A l'étape 1209, l'émetteur code les premiers bits d'information TFCI et les seconds bits d'information TFCI avec le procédé déterminé, et ensuite à l'étape 1210 il multiplexe les premier symboles TFCI codés et les seconds symboles TFCI codés. Si l'émetteur détermine d'augmenter la fiabilité ou les performances du premier TFCI ou du second TFCI par le

procédé d'augmentation du rendement de codage à l'étape 1207, l'émetteur multiplexe à l'étape 1210 les premiers symboles TFCI codés et les seconds symboles TFCI codés, qui sont codés à l'étape 1209. Si l'émetteur détermine à l'étape 5 1207 d'augmenter la fiabilité ou les performances du premier TFCI ou du second TFCI avant la transmission avec le procédé de transmission répétée, à l'étape 1209 l'émetteur répète les premiers symboles TFCI codés et les seconds symboles TFCI codés, et ensuite il les multiplexe à l'étape 1210. 10 Dans le cas contraire, l'émetteur répète à l'étape 1210 les premiers symboles TFCI codés et les seconds symboles TFCI codés, qui sont codés à l'étape 1209.

Comme décrit ci-dessus, le mode de réalisation de la présente invention peut coder/décoder divers types des bits 15 TFCI en utilisant une seule structure de codeur/décodeur. De plus, le mode de réalisation multiplexe les symboles TFCI codés avec les différentes techniques de codage, de façon que les symboles TFCI soient répartis uniformément avant d'être transmis. Pour les 10 bits d'information d'entrée, le 20 codage de TFCI est effectué avec l'un sélectionné des rapports de 1:9, 2:8, 3:7, 4:6, 5:5, 6:4, 7:3, 8:2 et 9:1, en fonction du type et des caractéristiques des données transmises sur le DSCH et le DCH, ce qui contribue à la souplesse du HSM qui est supérieur au LSM en termes de signalisation et de retard. De plus, le codeur code les bits TFCI pour le 25 DCH et les bits TFCI pour le DSCH, et ensuite il stocke dans un dispositif de stockage les symboles TFCI codés pour le DCH et les symboles TFCI codés pour le DSCH, ce qui assure un traitement d'information rapide.

30 Il va de soi que de nombreuses modifications peuvent être apportées au dispositif et au procédé décrits et représentés, sans sortir du cadre de l'invention.

REVENDICATIONS

1. Procédé pour effectuer une application de premiers symboles TFCI ("Transport Format Combinaison Indicator", c'est-à-dire Indicateur de Combinaison de Format de Transport) codés et de seconds symboles TFCI codés vers une trame radio dans un appareil d'émission d'un système de communication mobile, pour coder k premiers bits TFCI et (10-k) seconds bits TFCI, une somme des premiers symboles TFCI codés et des seconds symboles TFCI codés étant égale à 32, caractérisé en ce qu'il comprend les étapes suivantes :
- on multiplexe les symboles codés de façon que les premiers symboles TFCI codés et les seconds symboles TFCI codés soient uniformément répartis conformément à un mode de transmission et un débit de données de la trame radio, et on fournit en sortie 32 symboles codés; et on effectue une application des 32 symboles codés multiplexés vers la trame radio pour respecter le nombre des symboles codés qui peuvent être appliqués vers une trame radio, déterminé conformément au mode de transmission et au débit de données de la trame radio.
2. Procédé selon la revendication 1, caractérisé en ce que les premiers symboles TFCI codés sont multiplexés dans les positions calculées par l'expression

$$b_{\left[\frac{32}{n}x(i+1)\right]-1} = c_i^1 \quad (0 \leq i \leq n - 1)$$

dans laquelle n représente le nombre total des premiers symboles TFCI codés, et i représente un index indiquant un symbole codé arbitraire parmi les premiers symboles TFCI codés.

3. Procédé selon la revendication 1, caractérisé en ce que les seconds symboles TFCI codés sont multiplexés dans les positions calculées par l'expression

$$b_{i+\left\lfloor \frac{n}{32-n}x(i+\frac{1}{2}) \right\rfloor} = c_i^2 \quad (0 \leq i \leq m-1)$$

dans laquelle n représente le nombre total des premiers symboles TFCI codés, m représente le nombre total des seconds symboles TFCI codés, et i représente un index
 5 indiquant un symbole codé arbitraire parmi les premiers symboles TFCI codés.

4. Procédé selon la revendication 1, caractérisé en ce que si le nombre de symboles codés qui peuvent être appliqués vers une trame radio est de 30, alors 30 symboles
 10 codés à l'exception de l'un arbitraire des premiers symboles TFCI codés et de l'un arbitraire des seconds symboles TFCI codés, sont appliqués vers la trame radio.

5. Procédé selon la revendication 4, caractérisé en ce que le symbole arbitraire parmi les premiers symboles
 15 TFCI codés est le dernier symbole codé des premiers symboles TFCI codés, et le symbole arbitraire parmi les seconds symboles TFCI codés est le dernier symbole codé des seconds symboles TFCI codés.

6. Procédé selon la revendication 1, caractérisé en ce que si le nombre de symboles codés qui peuvent être appliqués vers une trame radio est de 30, alors 30 symboles
 20 codés à l'exception de deux symboles codés arbitraires parmi les premiers symboles TFCI codés ou de deux symboles codés arbitraires parmi les seconds symboles TFCI codés, sont
 25 appliqués vers la trame radio.

7. Procédé selon la revendication 1, dans lequel si le nombre de symboles codés qui peuvent être appliqués vers une trame radio est de 120, les 32 symboles codés multiplexés sont répétés 3 fois, les 24 premiers symboles
 30 codés parmi les 32 symboles codés multiplexés sont répétés davantage, et sont ensuite appliqués vers la trame radio.

8. Procédé selon la revendication 1, caractérisé en ce que si le nombre de symboles codés qui peuvent être appliqués vers une trame radio est de 32, les 32 symboles

codés multiplexés sont appliqués vers la trame radio.

9. Procédé selon la revendication 1, caractérisé en ce que si le nombre de symboles codés qui peuvent être appliqués vers une trame radio est de 128, les 32 symboles
5 codés multiplexés sont répétés 4 fois et sont ensuite appliqués vers la trame radio.

10. Appareil pour transmettre des premiers bits TFCI ("Transport Format Combinaison Indicator", c'est-à-dire Indicateur de Combinaison de Format de Transport) et des
10 seconds bits TFCI sur une trame radio dans un appareil d'émission d'un système de communication mobile, caractérisé en ce qu'il comprend : au moins un codeur pour coder k premiers bits TFCI à un premier rendement de codage pour fournir en sortie $(3k+1)$ premiers symboles TFCI codés, et
15 pour coder $(10-k)$ seconds bits TFCI à un second rendement de codage pour fournir en sortie $(31-3k)$ seconds symboles TFCI codés; et un dispositif d'arrangement de symboles codés pour multiplexer les symboles codés de façon que les premiers symboles TFCI codés et les seconds symboles TFCI codés
20 soient uniformément répartis conformément à un mode de transmission et un débit de données de la trame radio, et pour fournir en sortie les symboles codés multiplexés conformément au nombre de symboles codés qui peuvent être transmis sur une trame radio.

25 11. Appareil selon la revendication 10, caractérisé en ce qu'il comprend en outre un sélecteur pour sélectionner les premiers bits TFCI et les seconds bits TFCI conformément à une valeur de k, et pour fournir au codeur $(31-k)$ les bits TFCI sélectionnés.

30 12. Appareil selon la revendication 10, caractérisé en ce que le dispositif d'arrangement de symboles codés multiplexe les symboles codés de façon que les premiers symboles TFCI codés soient fournis en sortie dans les positions calculées par l'expression

$$b_{\left\lfloor \frac{32}{n}x(i+1) \right\rfloor - 1} = c_i^1 \quad (0 \leq i \leq n - 1)$$

dans laquelle n représente le nombre total des premiers symboles TFCI codés, et i représente un index indiquant un symbole codé arbitraire parmi les premiers symboles TFCI
5 codés.

13. Appareil selon la revendication 10, caractérisé en ce que le dispositif d'arrangement de symboles codés multiplexe les symboles codés de façon que les seconds symboles TFCI codés soient fournis en sortie dans les
10 positions calculées par l'expression

$$b_{i + \left\lfloor \frac{n}{32-n}x(i + \frac{1}{2}) \right\rfloor} = c_i^2 \quad (0 \leq i \leq m - 1)$$

dans laquelle n représente le nombre total des premiers symboles TFCI codés, m représente le nombre total des seconds symboles TFCI codés, et i représente un index
15 indiquant un symbole codé arbitraire parmi les premiers symboles TFCI codés.

14. Procédé pour transmettre des premiers bits TFCI ("Transport Format Combinaison Indicator", c'est-à-dire Indicateur de Combinaison de Format de Transport) et des
20 seconds bits TFCI sur une trame radio dans un appareil d'émission d'un système de communication mobile, caractérisé en ce qu'il comprend les étapes suivantes : on code k premiers bits TFCI à un premier rendement de codage pour fournir en sortie $(3k+1)$ premiers symboles TFCI codés; on
25 code $(10-k)$ seconds bits TFCI à un second rendement de codage pour fournir en sortie $(31-3k)$ seconds symboles TFCI codés; on multiplexe les symboles codés de façon que les premiers symboles TFCI codés et les seconds symboles TFCI codés soient uniformément répartis conformément à un mode de
30 transmission et un débit de données de la trame radio; et on fournit en sortie les symboles codés multiplexés

conformément au nombre de symboles codés qui peuvent être transmis sur une trame radio.

15. Procédé selon la revendication 14, caractérisé en ce que les premiers symboles TFCI codés sont fournis en
5 sortie dans les positions calculées par l'expression

$$b_{\left\lfloor \frac{32}{n} x(i+1) \right\rfloor - 1} = c_i^1 \quad (0 \leq i \leq n - 1)$$

dans laquelle n représente le nombre total des premiers symboles TFCI codés, et i représente un index indiquant un symbole codé arbitraire parmi les premiers symboles TFCI
10 codés.

16. Procédé selon la revendication 14, caractérisé en ce que les seconds symboles TFCI codés sont fournis en sortie dans les positions calculées par l'expression

$$b_{i + \left\lfloor \frac{n}{32-n} x(i + \frac{1}{2}) \right\rfloor} = c_i^2 \quad (0 \leq i \leq m - 1)$$

15 dans laquelle n représente le nombre total des premiers symboles TFCI codés, m représente le nombre total des seconds symboles TFCI codés, et i représente un index indiquant un symbole codé arbitraire parmi les premiers symboles TFCI codés.

20 17. Appareil pour décoder k premiers bits TFCI et (10-k) seconds bits TFCI dans un appareil de réception d'un système de communication mobile, pour recevoir (3k-1) premiers symboles TFCI codés pour un DCH ("Dedicated Channel", c'est-à-dire Canal Dédié) et (31-3k) seconds
25 symboles TFCI codés pour un DSCH ("Downlink Shared Channel", c'est-à-dire Canal Partagé de Liaison Descendante), caractérisé en ce qu'il comprend : un dispositif de réarrangement de symboles codés pour séparer les premiers symboles TFCI codés et les seconds symboles TFCI codés,

transmis sur un DPCH ("Dedicated Physical Channel", c'est-à-dire Canal Physique Dédié), conformément à une valeur de k, pour le réarrangement; et au moins un décodeur pour décoder les premiers symboles TFCI codés, pour fournir en sortie les k premiers bits TFCI, et décoder les seconds symboles TFCI codés pour fournir en sortie les (10-k) seconds bits TFCI.

18. Appareil selon la revendication 17, caractérisé en ce que le dispositif de réarrangement de symboles codés (910) sépare les premiers symboles TFCI codés existant dans les positions calculées par la formule suivante, à partir des symboles codés obtenus en multiplexant les premiers symboles TFCI codés et les seconds symboles TFCI codés :

$$b_{\left[\frac{32}{n}x(i+1)\right]-1} = c_i^1 \quad (0 \leq i \leq n - 1)$$

dans laquelle n représente le nombre total des premiers symboles TFCI codés, et i représente un index indiquant un symbole codé arbitraire parmi les premiers symboles TFCI codés.

19. Appareil selon la revendication 17, caractérisé en ce que le dispositif de réarrangement de symboles codés sépare les seconds symboles TFCI codés existant dans les positions calculées par la formule suivante, à partir des symboles codés obtenus en multiplexant les premiers symboles TFCI codés et les seconds symboles TFCI codés :

$$b_{i+\left[\frac{n}{32-n}x(i+\frac{1}{2})\right]} = c_i^2 \quad (0 \leq i \leq m - 1)$$

dans laquelle n représente le nombre total des premiers symboles TFCI codés, m représente le nombre total des seconds symboles TFCI codés, et i représente un index indiquant un symbole codé arbitraire parmi les premiers symboles TFCI codés.

20. Procédé pour décoder k premiers bits TFCI et (10-k) seconds bits TFCI dans un appareil de réception d'un système de communication mobile pour recevoir (3k-1) premiers symboles TFCI codés pour un DCH ("Dedicated Channel", c'est-à-dire Canal Dédié) et (31-3k) seconds symboles TFCI codés pour un DSCH ("Downlink Shared Channel", c'est-à-dire Canal Partagé de Liaison Descendante), caractérisé en ce qu'il comprend les étapes suivantes : on sépare les premiers symboles TFCI codés et les seconds symboles TFCI codés, transmis sur un DPCH ("Dedicated Physical Channel", c'est-à-dire Canal Physique Dédié), conformément à une valeur de k, pour le réarrangement; on décode les premiers symboles TFCI codés pour fournir en sortie les k premiers bits TFCI; et on décode les seconds symboles TFCI codés pour fournir en sortie les (10-k) seconds bits TFCI.

21. Procédé selon la revendication 20, caractérisé en ce que les premiers symboles TFCI codés existant dans les positions calculées par la formule suivante sont séparés des symboles codés obtenus en multiplexant les premiers symboles TFCI codés et les seconds symboles TFCI codés :

$$b_{\left[\frac{32}{n}x(i+1)\right]-1} = c_i^1 \quad (0 \leq i \leq n - 1)$$

dans laquelle n représente le nombre total des premiers symboles TFCI codés, et i représente un index indiquant un symbole codé arbitraire parmi les premiers symboles TFCI codés.

22. Procédé selon la revendication 20, caractérisé en ce que les seconds symboles TFCI codés existant dans les positions calculées par la formule suivante sont séparés des symboles codés obtenus en multiplexant les premiers symboles TFCI codés et les seconds symboles TFCI codés :

$$b_{i+\left\lfloor \frac{n}{32-n}x(i+\frac{1}{2}) \right\rfloor} = c_i^2 \quad (0 \leq i \leq m-1)$$

dans laquelle n représente le nombre total des premiers symboles TFCI codés, m représente le nombre total des seconds symboles TFCI codés, et i représente un index
 5 indiquant un symbole codé arbitraire parmi les premiers symboles TFCI codés.

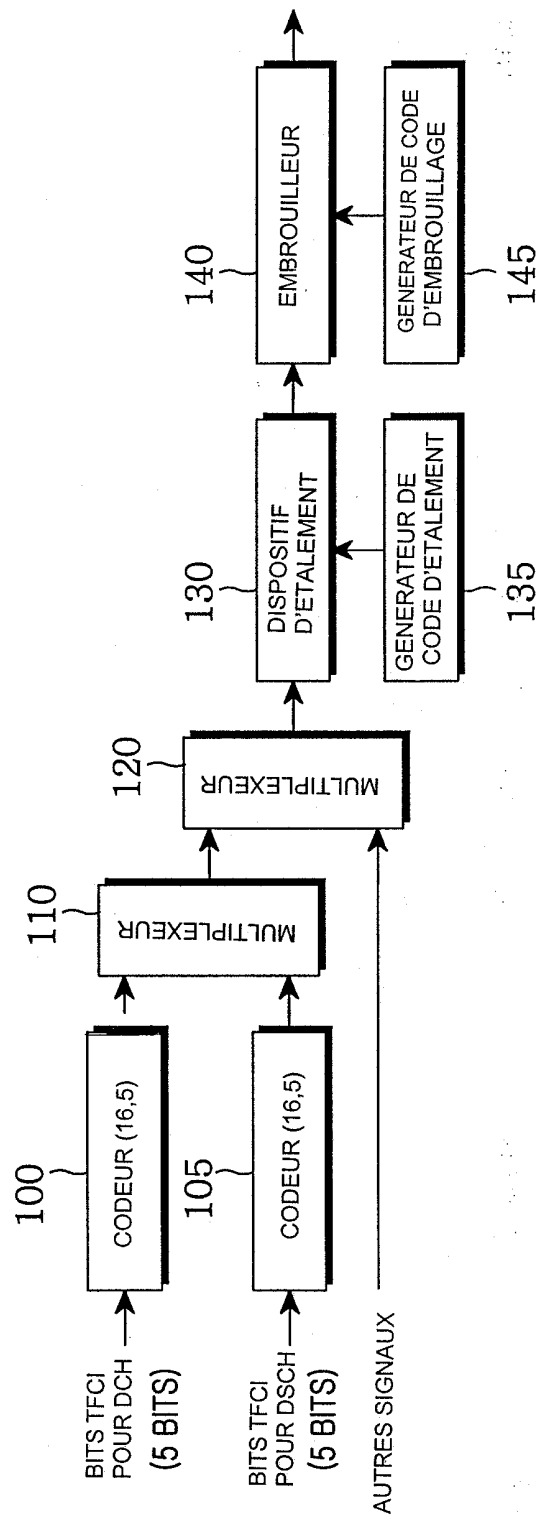


FIG.1

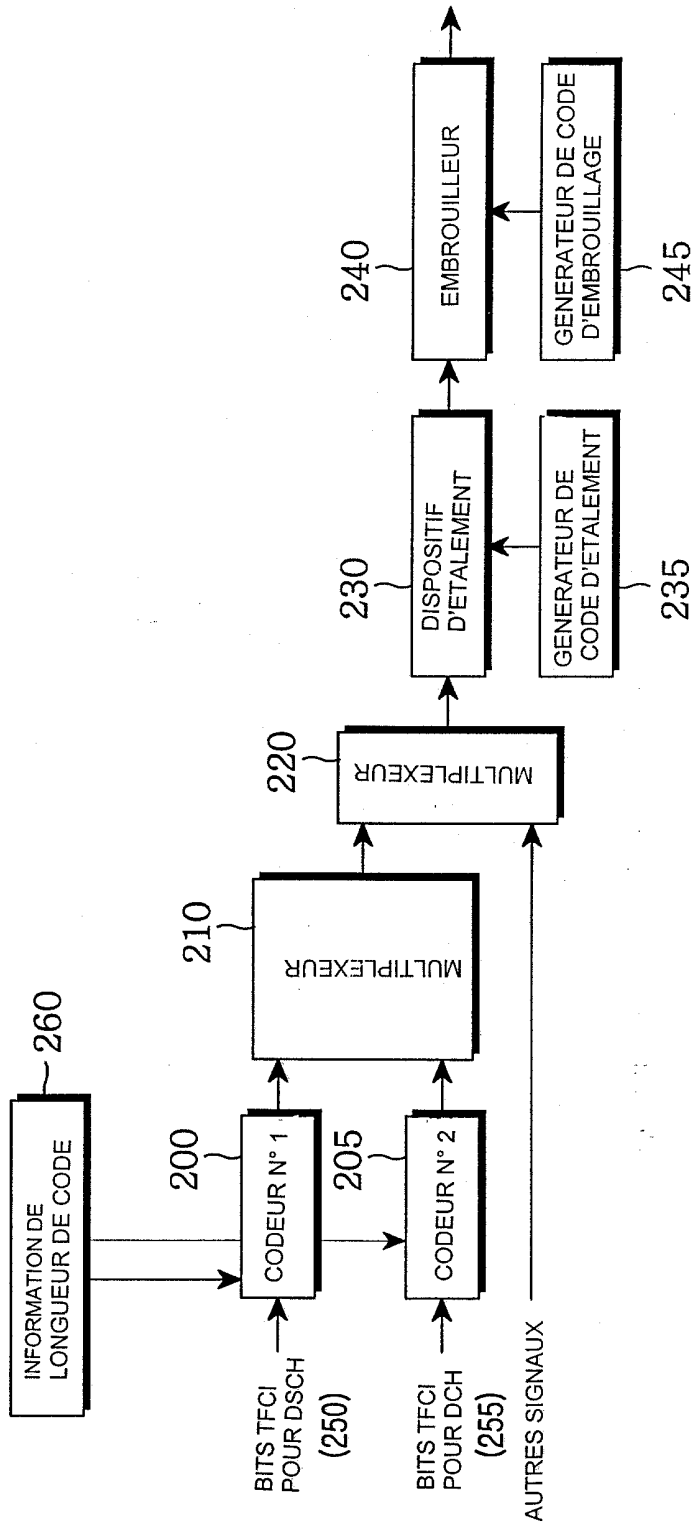


FIG. 2

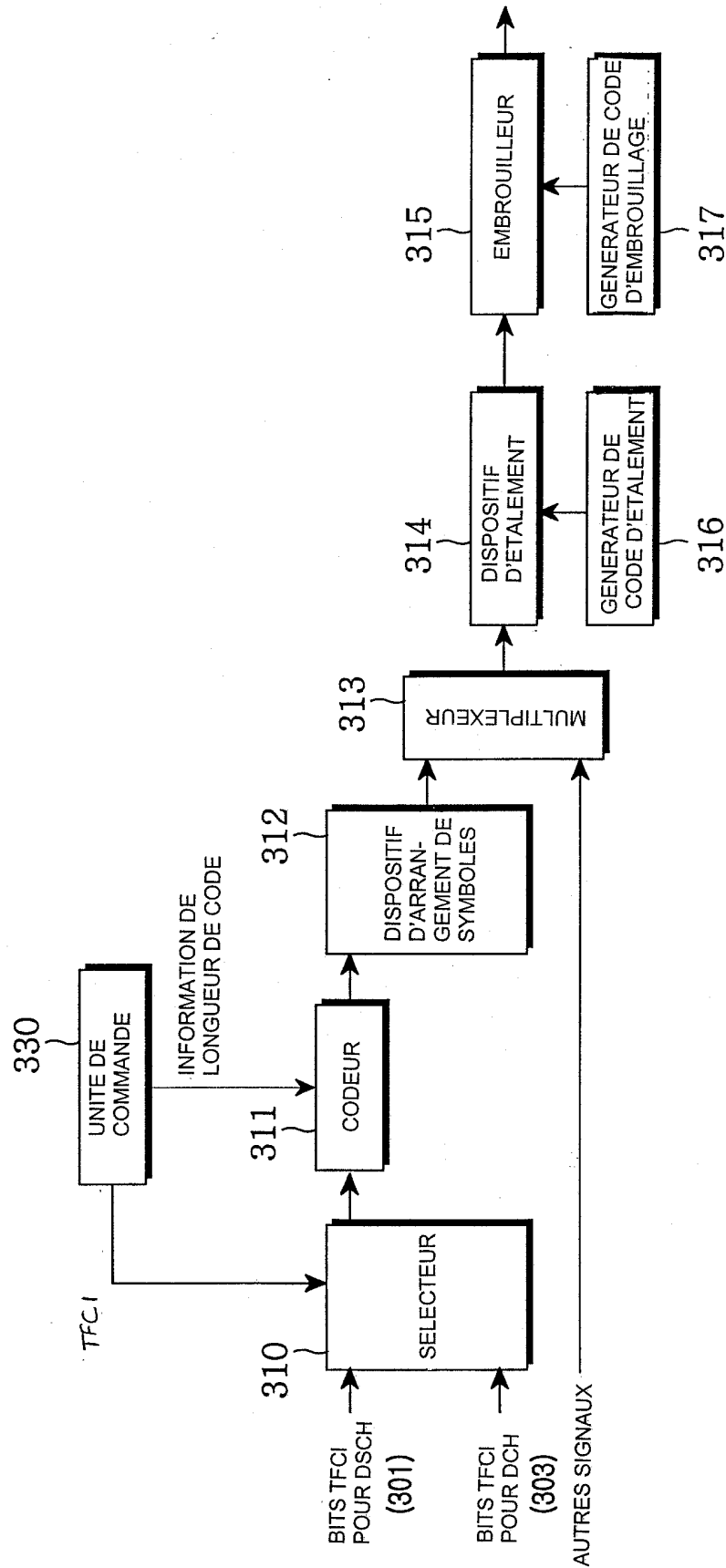


FIG.3

4/20

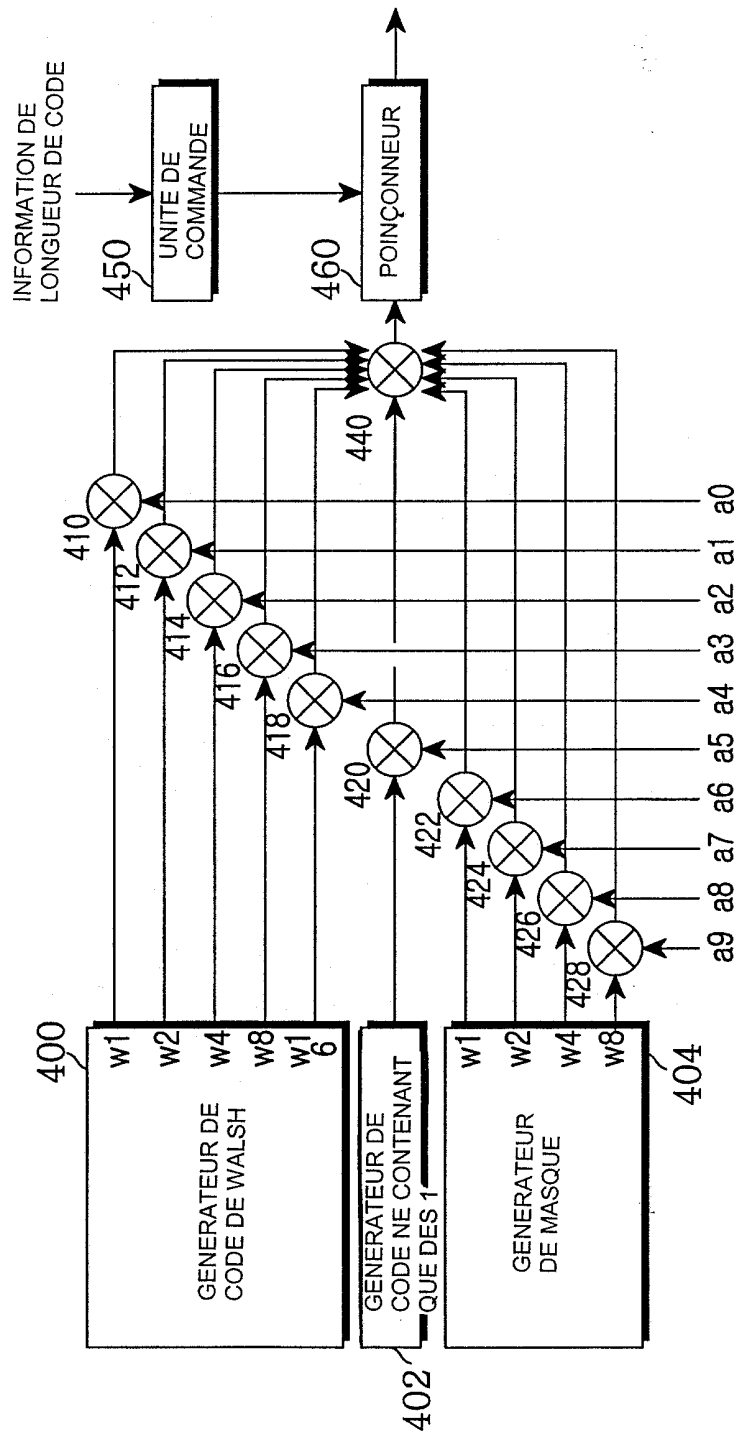


FIG. 4

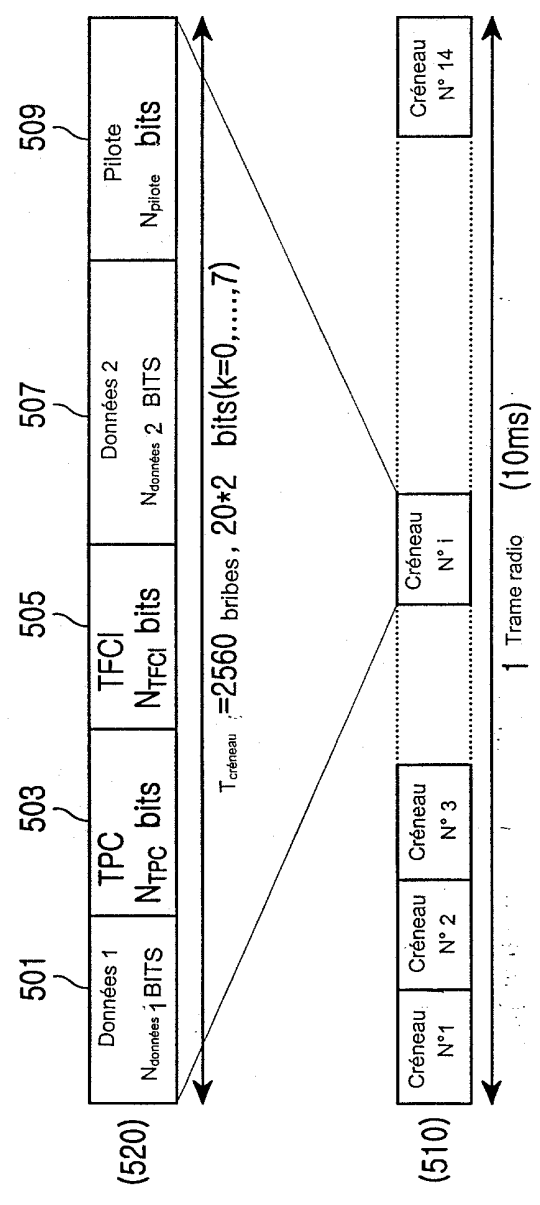


FIG.5

6/20

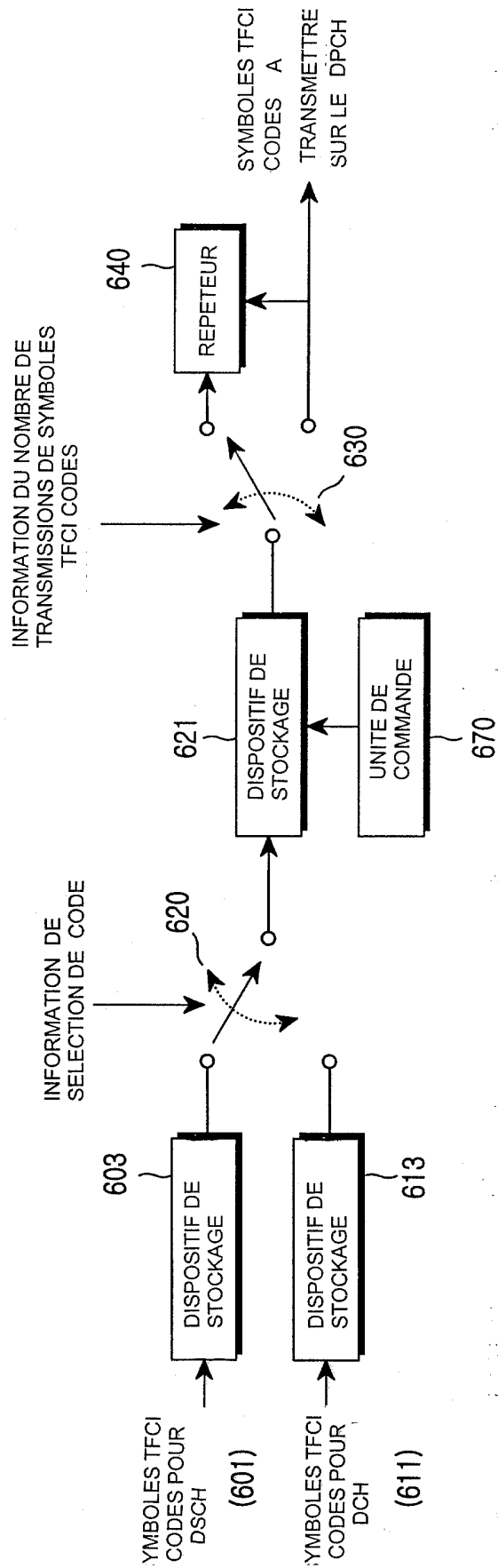


FIG.6

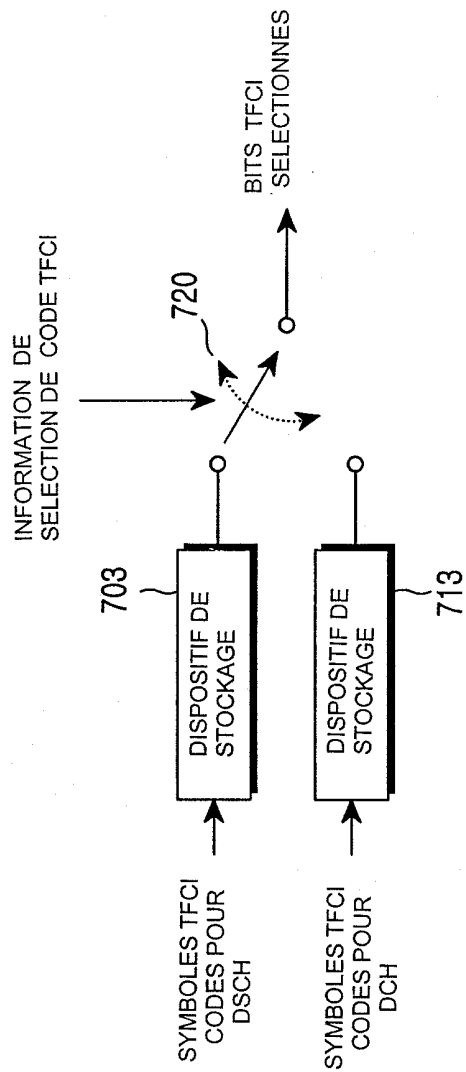


FIG. 7

8/20

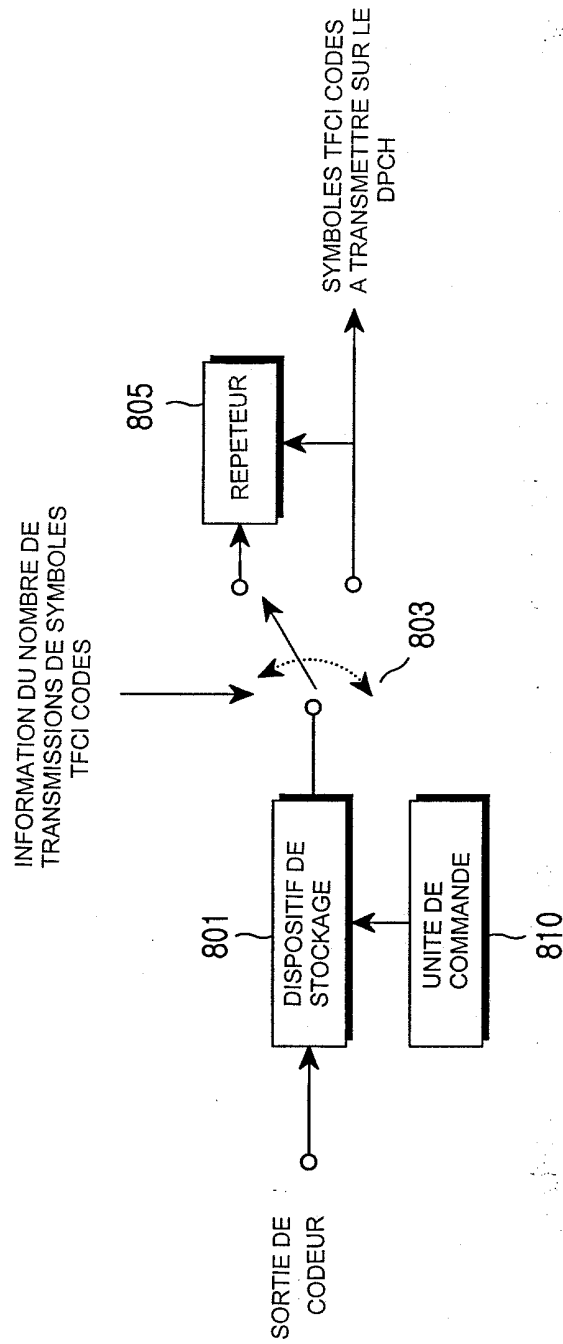


FIG. 8

9/20

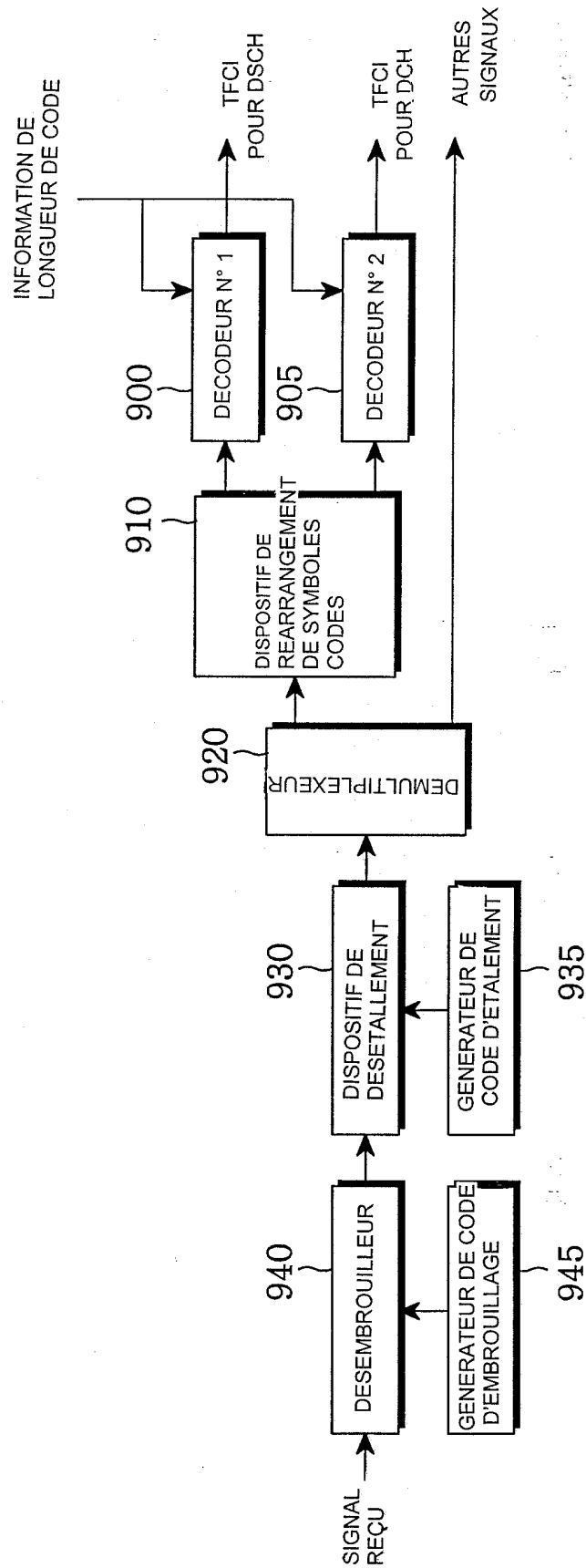


FIG.9

10/20

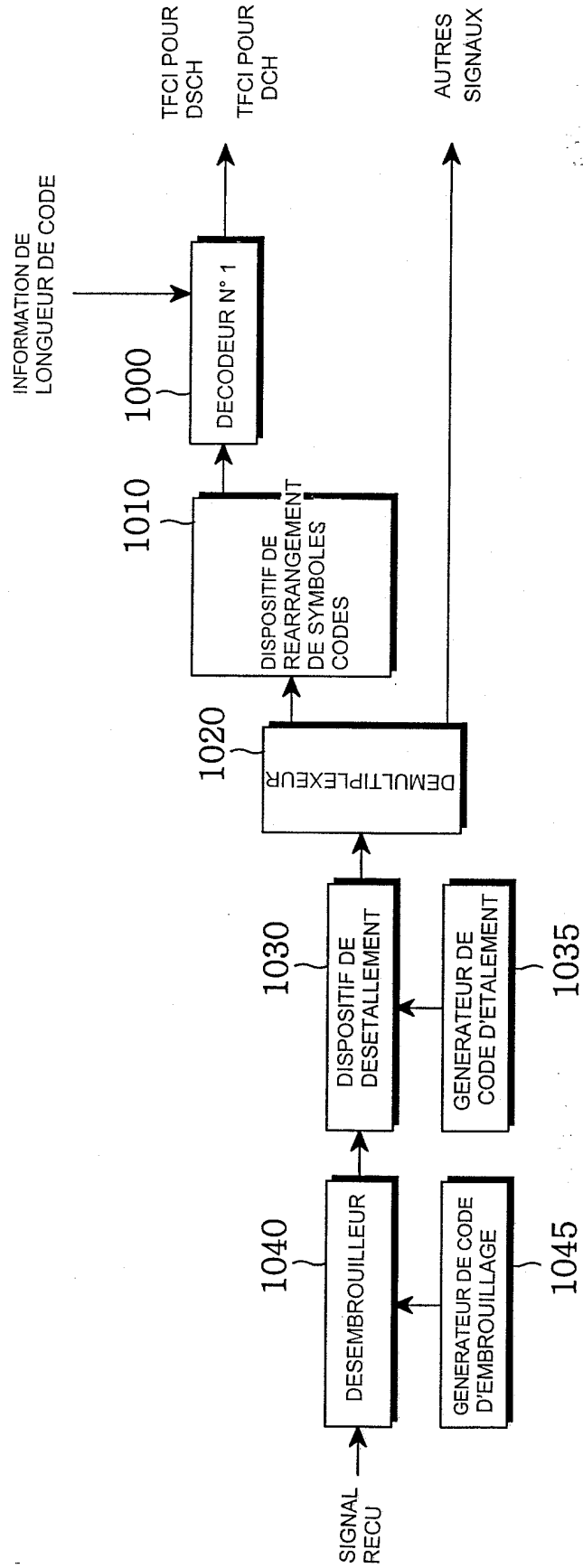


FIG.10

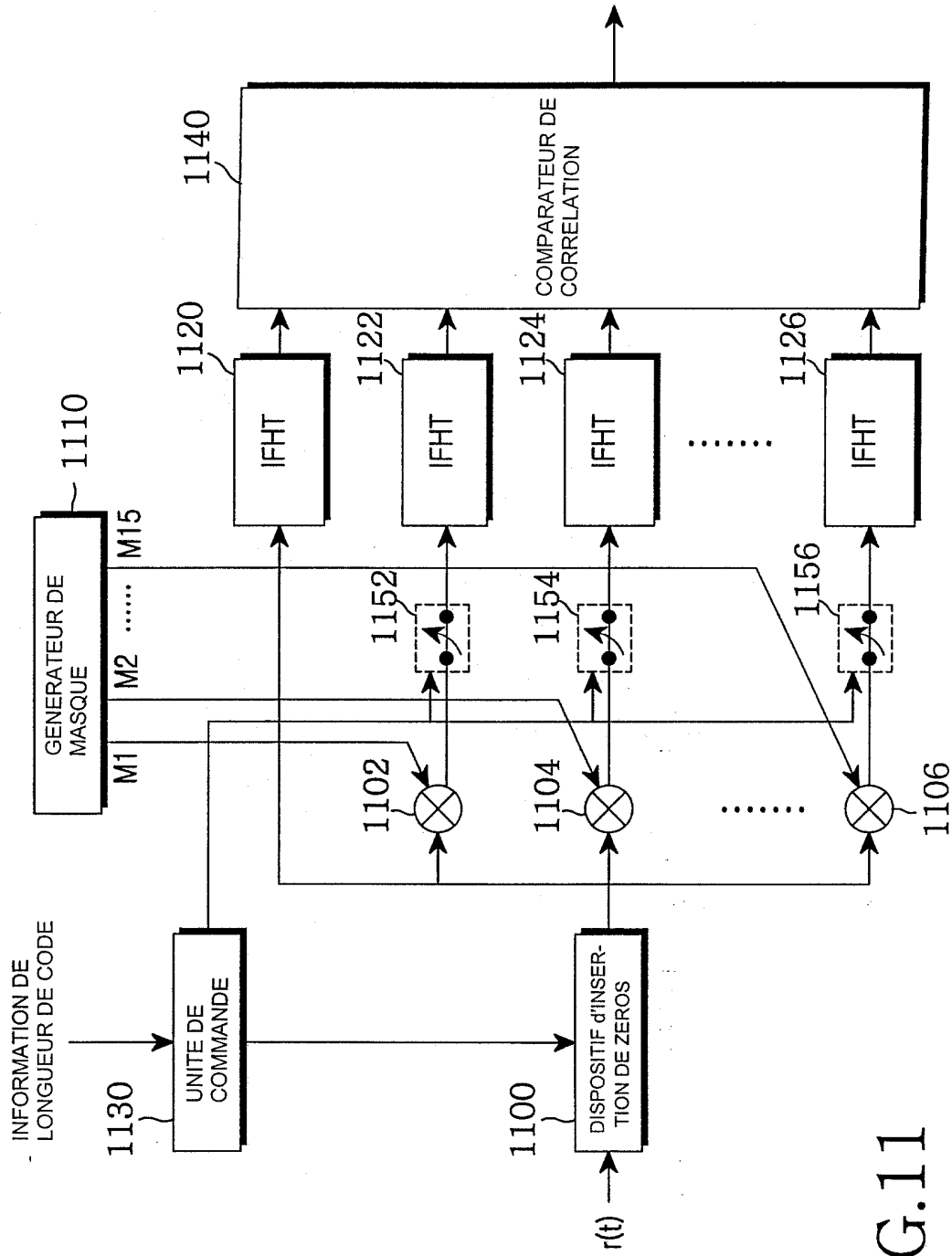


FIG.11

12/20

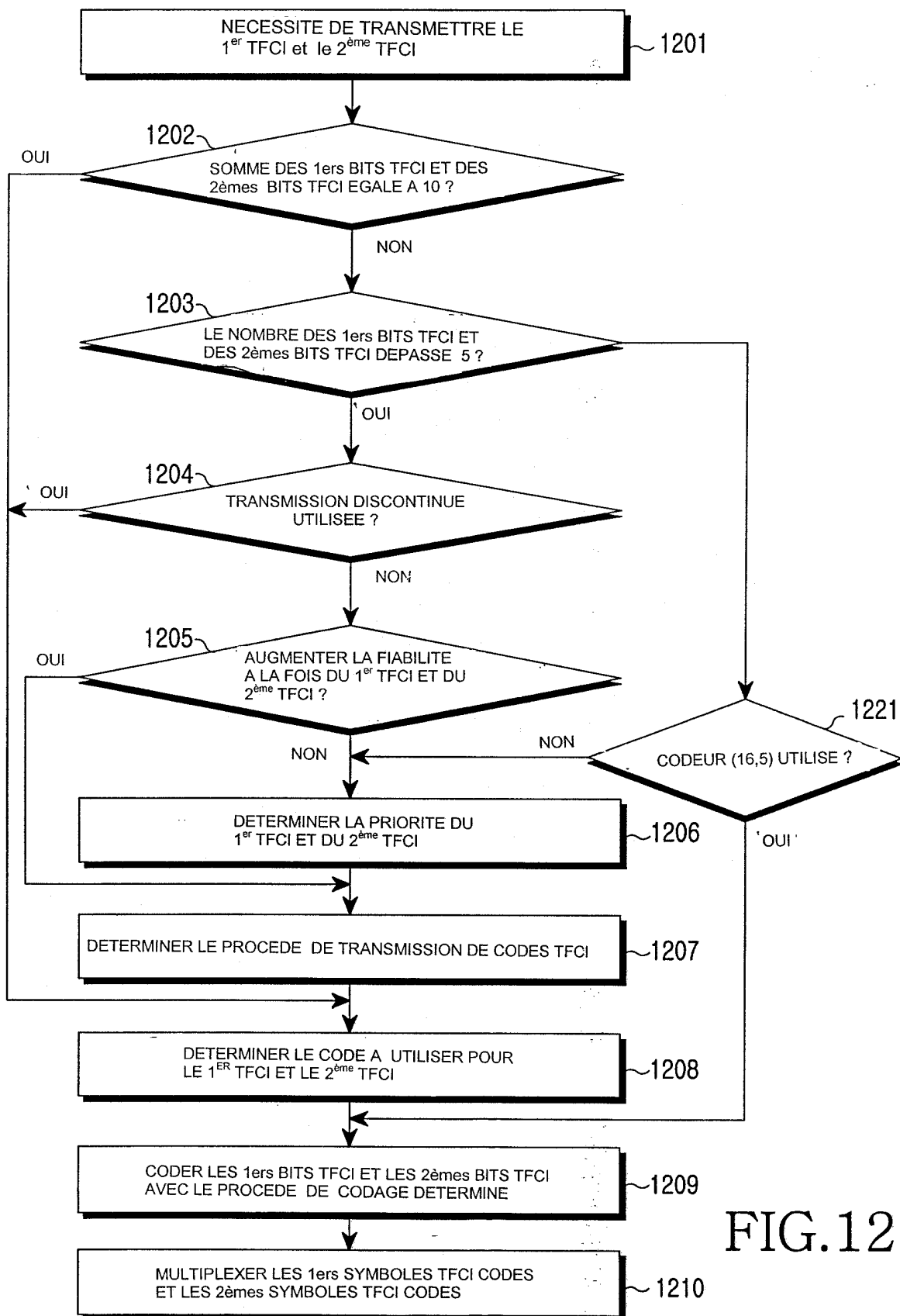


FIG.12

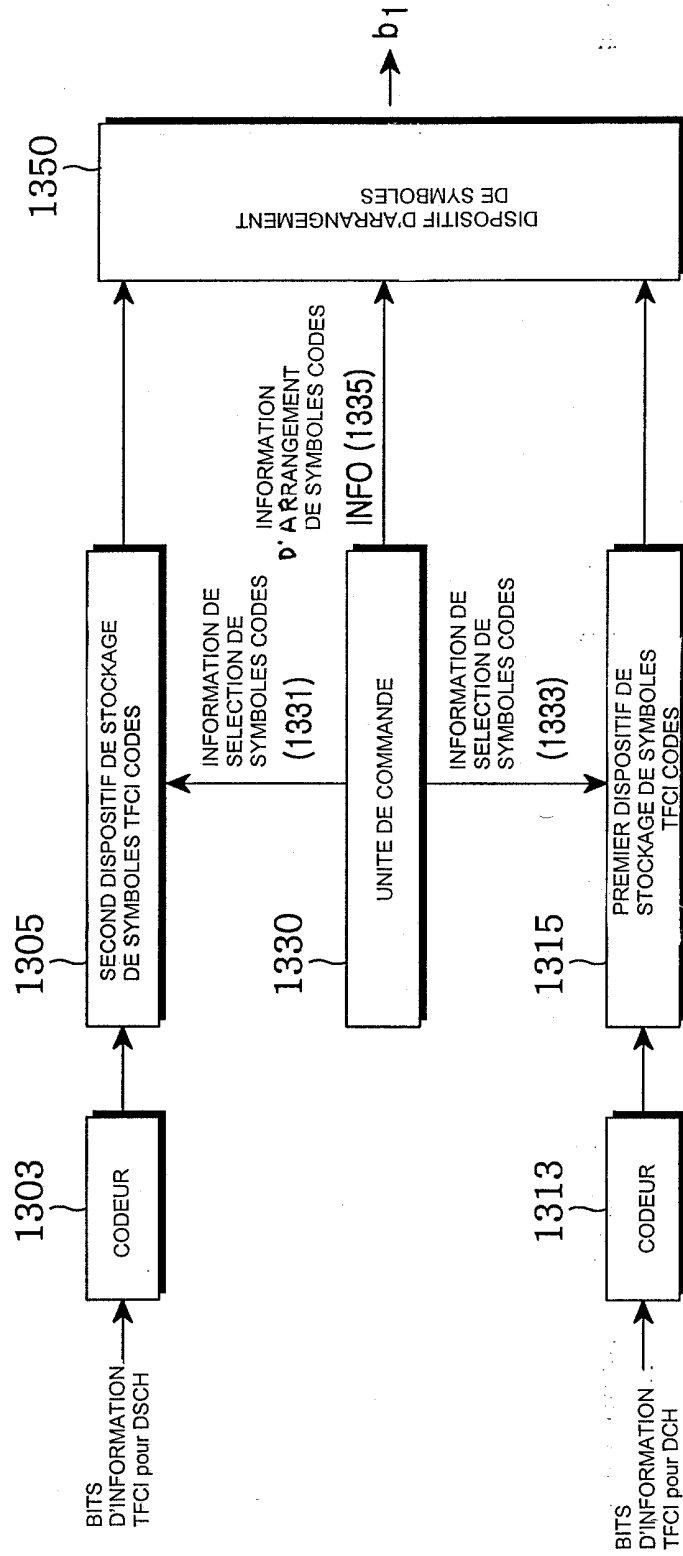


FIG.13

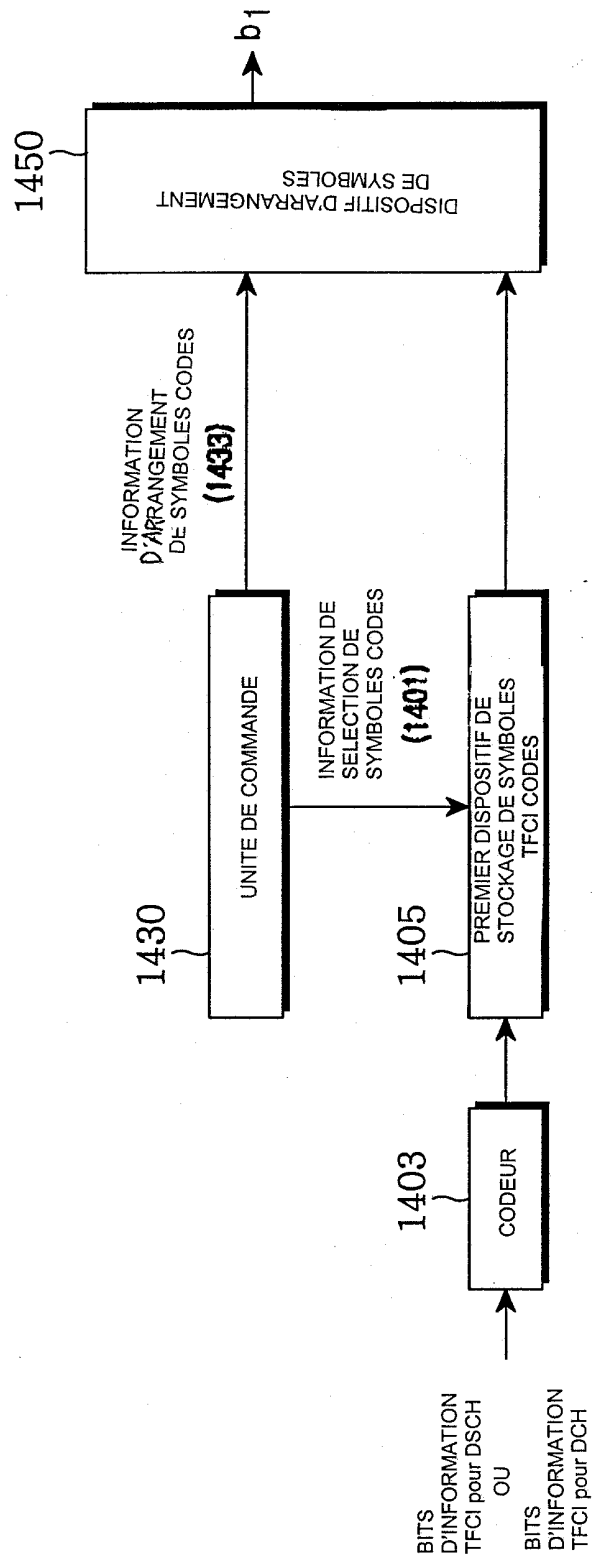
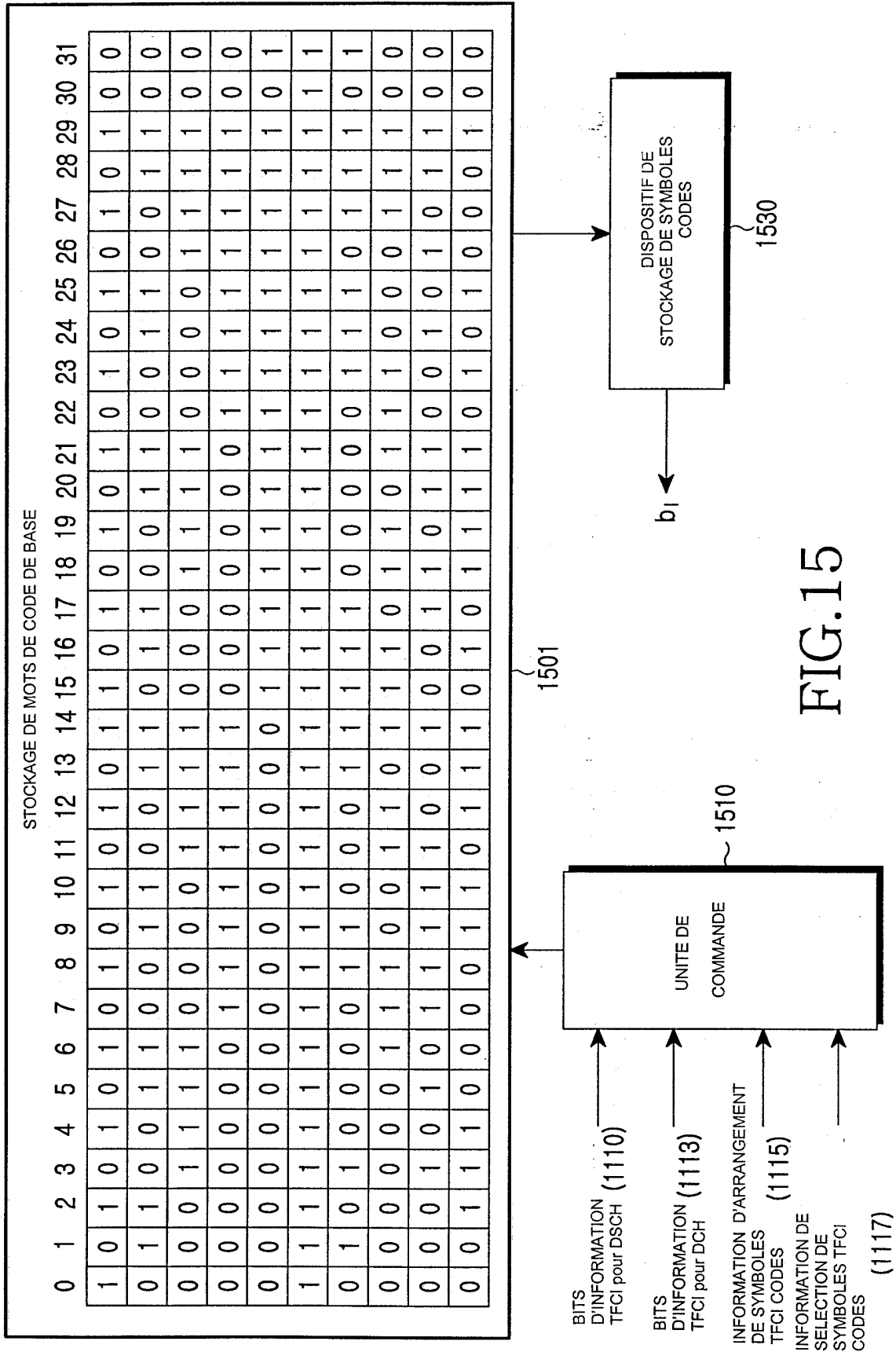


FIG.14



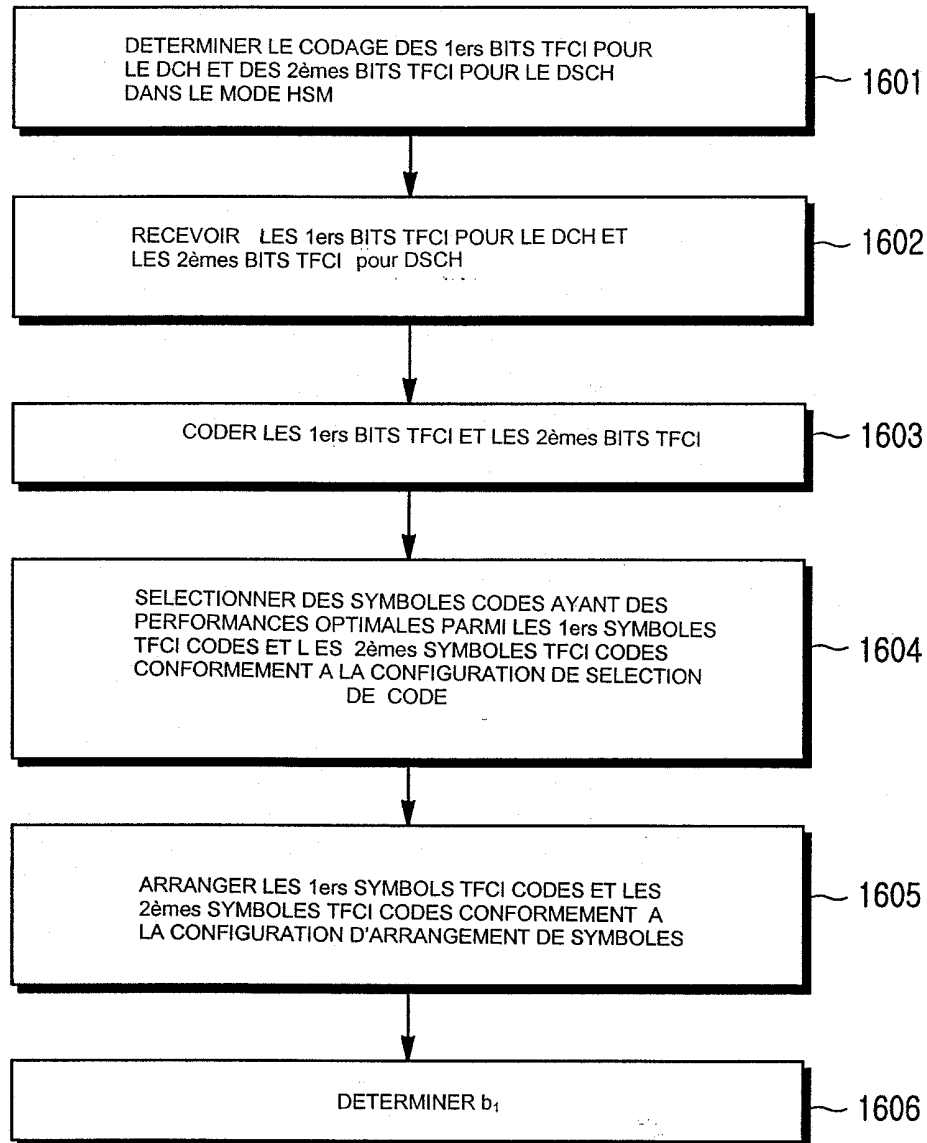


FIG.16

17/20

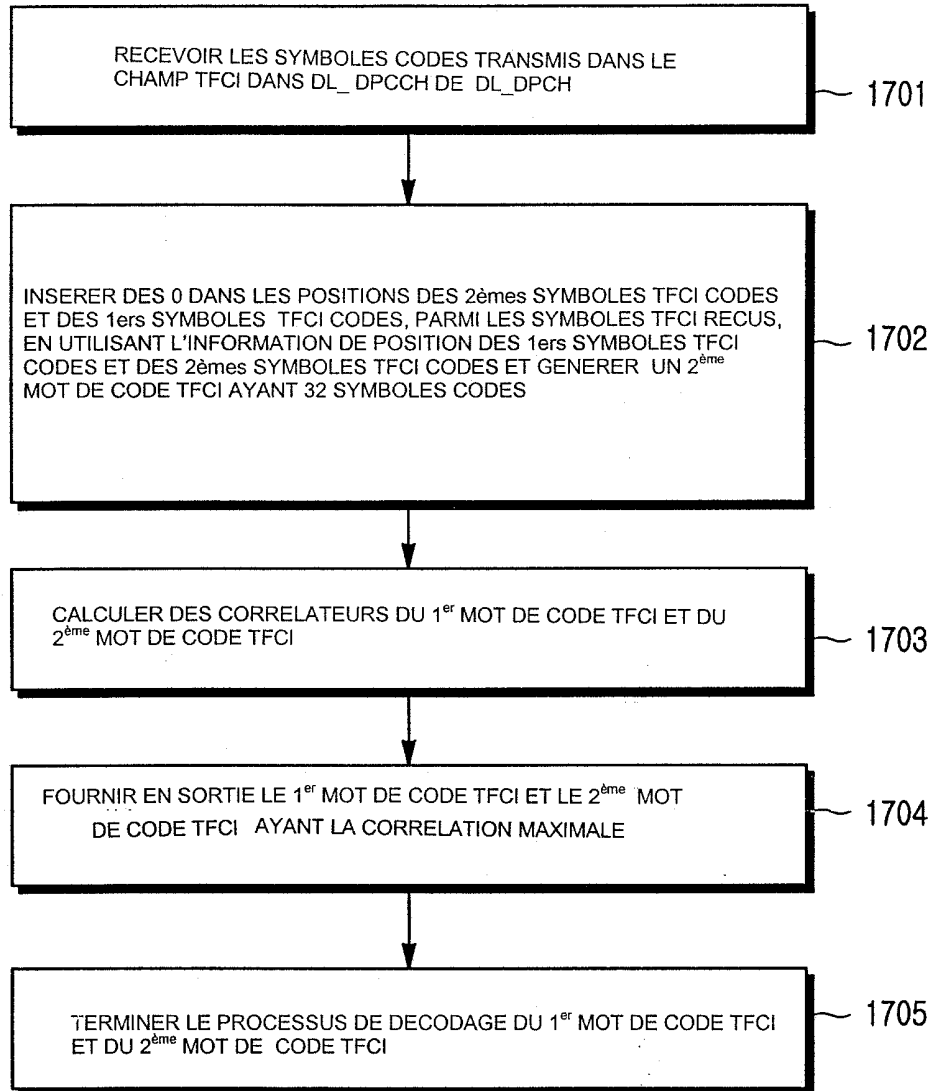


FIG.17

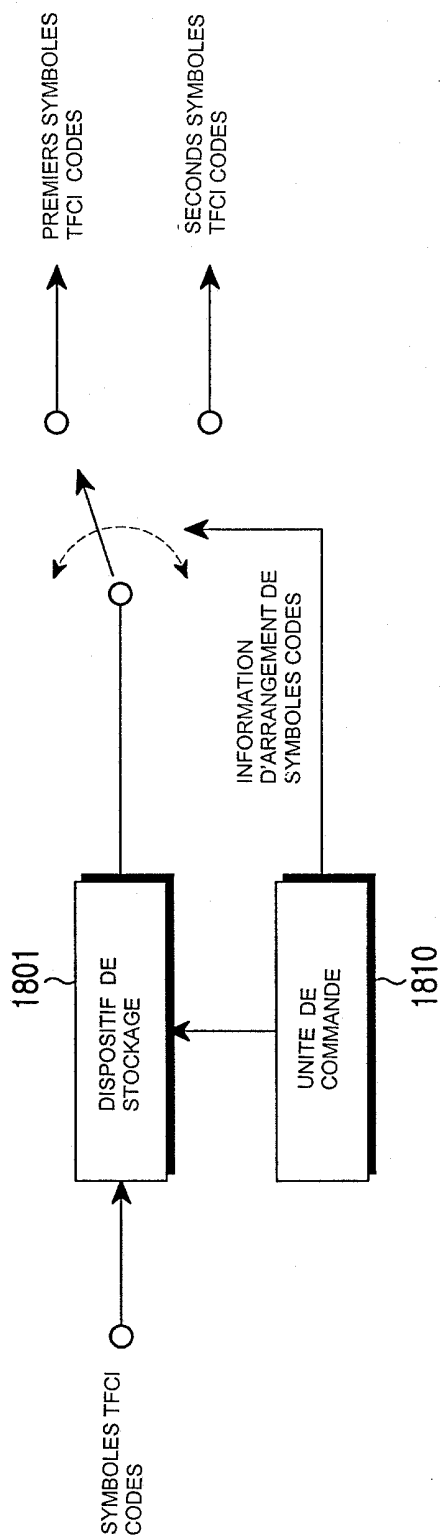


FIG.18a

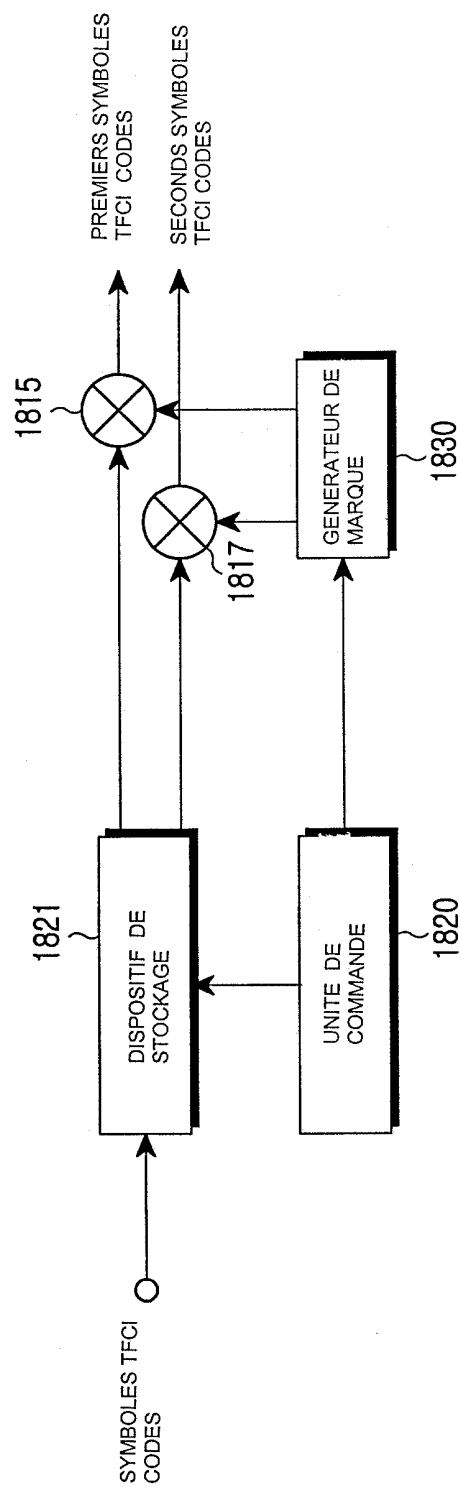


FIG. 18b

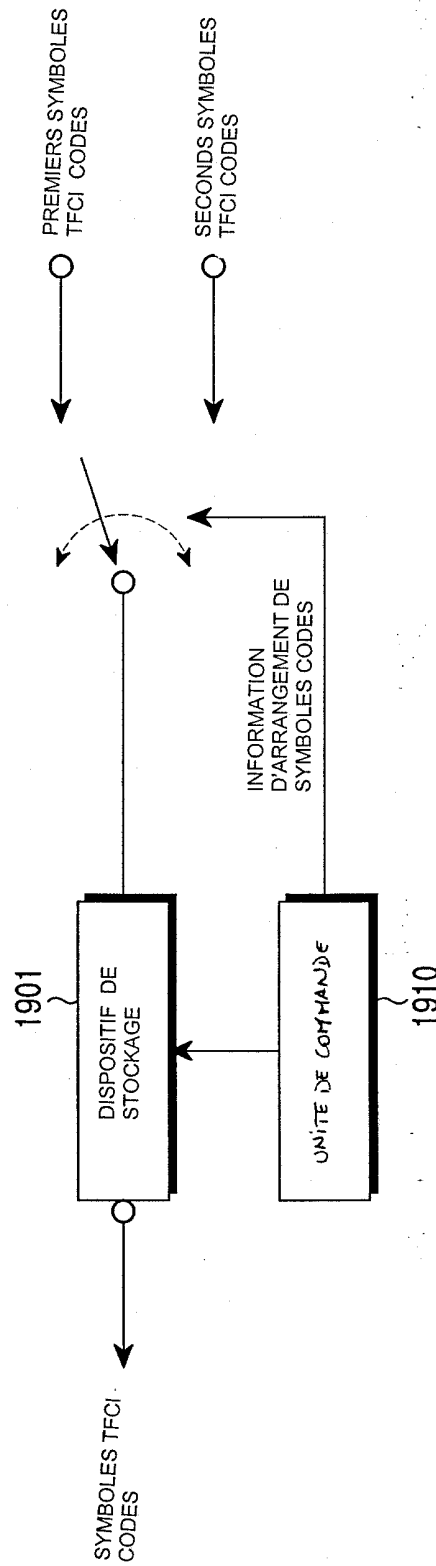


FIG.19