Exhibit 1

US007706348B2

(12) United States Patent
Kim et al.

(10) Patent No.: US 7,706,348 B2
(45) Date of Patent: *Apr. 27, 2010

(54) **APPARATUS AND METHOD FOR ENCODING/DECODING TRANSPORT FORMAT COMBINATION INDICATOR IN CDMA MOBILE COMMUNICATION SYSTEM**

(75) Inventors: **Jae-Yoel Kim**, Kunpo (KR); **Hee-Won Kang**, Seoul (KR)

(73) Assignee: **Samsung Electronics Co., Ltd.** (KR)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 435 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **11/006,388**

(22) Filed: **Dec. 7, 2004**

(65) **Prior Publication Data**

US 2005/0083901 A1 Apr. 21, 2005

Related U.S. Application Data

(63) Continuation of application No. 09/611,069, filed on Jul. 6, 2000, now Pat. No. 6,882,636.

(30) **Foreign Application Priority Data**

Jul. 6, 1999 (KR) 1999-27932

(51) **Int. Cl.**
H04B 7/216 (2006.01)

(52) **U.S. Cl.** **370/342**

(58) **Field of Classification Search** .... None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,550,809 A 8/1996 Bottomley et al.
5,870,414 A 2/1999 Chaib et al.

(Continued)

FOREIGN PATENT DOCUMENTS

EP 0 565 506 4/1993

(Continued)

OTHER PUBLICATIONS

3rd Generation Partnership Project (3GPP), TS 25.222 V2.0.0 Jun. 1999, pp. 1-25.*

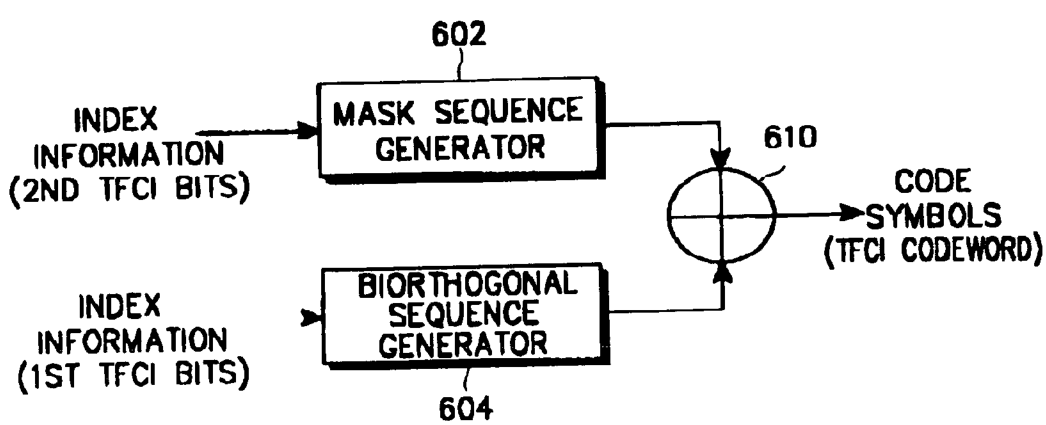
(Continued)

Primary Examiner—Anh-Vu Ly
(74) *Attorney, Agent, or Firm*—The Farrell Law Firm, LLP

(57) **ABSTRACT**

An apparatus and method for encoding/decoding a transport format combination indicator (TFCI) in a CDMA mobile communication system. In the TFCI encoding apparatus, a one-bit generator generates a sequence having the same symbols. A basis orthogonal sequence generator generates a plurality of basis orthogonal sequences. A basis mask sequence generator generates a plurality of basis mask sequences. An operation unit receives TFCI bits that are divided into a first information part representing biorthogonal sequence conversion, a second information part representing orthogonal sequence conversion, and a third information part representing mask sequence conversion and combines an orthogonal sequence selected from the basis orthogonal sequence based on the second information, a biorthogonal sequence obtained by combining the selected orthogonal sequence with the same symbols selected based on the first information part, and a mask sequence selected based on the biorthogonal sequence and the third information part, thereby generating a TFCI sequence.

88 Claims, 18 Drawing Sheets



U.S. PATENT DOCUMENTS

5,926,488	A	7/1999	Khayrallah	
6,049,633	A	4/2000	Cho	
6,208,699	B1	3/2001	Chen et al.	
6,341,125	B1	1/2002	Hong et al.	
6,408,481	B1	6/2002	Dyson	
6,515,987	B1	2/2003	Carrozza et al.	
6,542,478	B1	4/2003	Park	
6,665,288	B1	12/2003	Ottosson et al.	
6,674,712	B1	1/2004	Yang et al.	
6,882,636	B1 *	4/2005	Kim et al.	370/342
2002/0013926	A1 *	1/2002	Kim et al.	714/781
2004/0216025	A1 *	10/2004	Kim et al.	714/755
2005/0243763	A1 *	11/2005	Toskala et al.	370/328
2006/0002453	A1 *	1/2006	Song	375/145

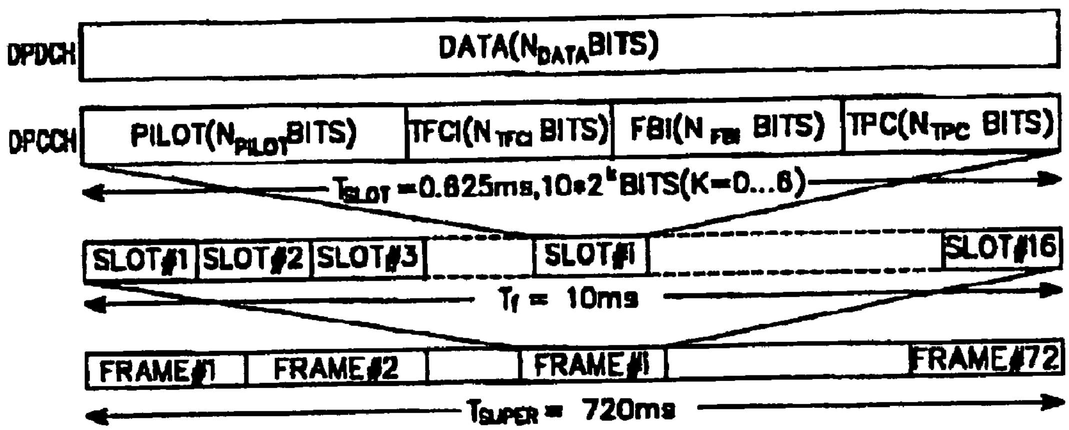
FOREIGN PATENT DOCUMENTS

EP	0565506	A2	10/1993
EP	0565606	B1	10/1999
EP	1 104 130		5/2001
EP	1104130	A2	5/2001
GB	2336078		10/1999
KR	1999015261		3/1999
KR	99-75942		10/1999
KR	99-76303		10/1999
KR	2000031698		6/2000
WO	WO 99/33212		7/1999

OTHER PUBLICATIONS

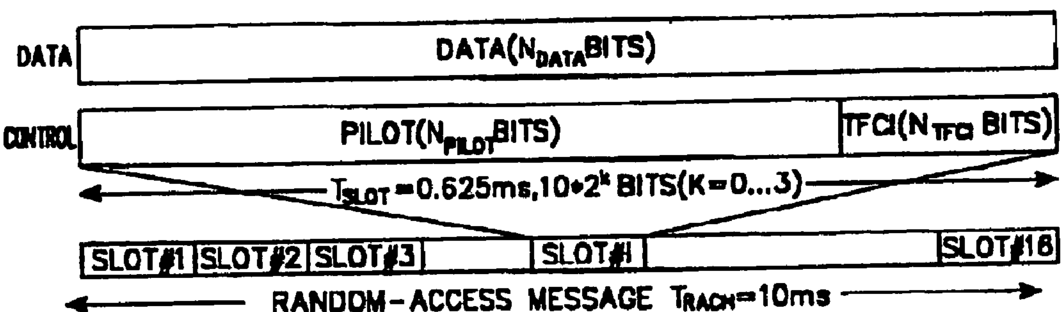
3GPP TSG RAN WG1, Meeting #7, Aug. 30-Sep. 3, 1999, pp. 1-6.*
TSG-RAN Working Group 1, Meeting #7, Aug. 30-Sep. 3, 1999, pp. 1-5.*
"UMTS Terrestrial Radio Access Network (UTRAN); UTRA FDD, Multiplexing, Channel Coding and Interleaving Description", ETSI, Feb. 1, 1999, pp. 1-16.
Brouwer et al., "An Updated Table of Minimum-Distance Bounds for Binary Linear Codes", IEEE Transactions on Information Theory, vol. 39, No. 2, Mar. 1993.
MacWilliams et al., "The Theory of Error-Correcting Code", North-Holland.
Samsung: Harmonization impact on TFCI and New Optimal Coding for Extended TFCI with Almost no Complexity Increase, TSG-RAN Working Group 1 Meeting #5, Jul. 13-16, 1999.
Expert Report of Dr. Stephen B. Wicker regarding Invalidity of U.S. Patents Nos. . . . 6,882,636 (150 pp).
Respondent's Prehearing Statement (96 pp).
Respondent's Submissions in Response to Order No. 56, Item 2, Paragraphs 16-26 (5 pp).
Memorandum in Support of Complainant's Motion to Supplement Complainant's Prehearing Statement to Respond to Respondent's New Technical Arguments (3 pp).
Pre-Hearing Statement of Complainants Samsung Telecommunications America LLP and Samsung Electronics Co. (55 pp).
Commission Investigative Staff's Pre-Hearing Brief (16 pp).
Complaint's Educational Submission Pursuant to Order No. 56 (26 pp).
Ericsson Inc., Telefonaktiebolaget LM Ericsson, Sony Ericsson Mobile Communications AB, And Sony Ericsson Mobile Communications (USA) Inc.'s Third Supplemental Responses (.
Rebuttal Expert Report of Richard Dale Wesel Re: U.S. Patent No. 6,882,636 (4pp).
Expert Report of Richard Dale Wesel RE: U.S. Patent No. 6,882,636 (38 pp).
Second Supplemental Rebuttal Expert Report of Dr. Stephen B. Wicker Regarding . . . Invalidity of U.S. Patent Nos. 6,882,636 (6 pp).
Complaint of Samsung Telecommunications America LLP under Section 337 of the Tariff Act of 1930 before the International Trade Commission (ITC).
Amended Response in ITC Action (54 pp).
Answer, Affirmative Defenses in Texas Action (27 pp).
Reply in Texas Action (32 pp).
Order granting stay in Texas Action (2 pp).
Berlekamp, et al., Weight Distributions of the Cosets of the (32,6) Reed-Muller Code, IEEE Trans Inf. Theory, vol. IT-18, No. 1, Jan. 1972, pp. 203-207.
Conway, et al., Soft Decoding Techniques for Codes and Lattices, Including Golay Code and the Leech Lattice, IEEE Trans Inf. Theory, vol. IT-32, No. 1, Jan. 1986, pp. 1,2, 41-50.
Edward C. Posner, Combinatorial Structures in Planetary Reconnaissance, Proceedings on Symposium 1968, pp. 1, 2, 15-46.
MacWilliams, et al., The Theory of Error Correcting Codes, 1977, 783 pages.
UMTS XX.04 Version 1.0.0, UMTS Terrestrial Radio Access Network (UTRAN), UTRA FDD, multiplexing, channel coding and interleaving description, 1999, 16 pages.
Wicker, Handbook of Coding Theory, 1998 (54 pages).
Wicker, Error Control Systems for Digital Communication and Storage, 1995 (527 pages).
Certified Translation of KR1999015261.
Certified Translation of KR2000031698.
Wicker, Deep Space Applications, Chapter 25 of Handbook of Coding Theory, vol. II, Elseiver Science B.V., 1998 (54 pages).
Liu, et al., Error Control Systems for Networks: An Overview, Mobile Networks and Applications, 2 (1997), pp. 167-182 (16 pages).
MacWilliams, et al., The Theory of Error Correcting Codes, Chap 13, Reed-Muller Codes, North Holland Mathematical Library, vol. 16, 1977, 119 pages.
Clark, Modifications of Codes, Preface and pp. 84-85 (1938).
Bossert, Kanalcodierung 1998 (44 pages).
Videotape Deposition of Douglas Neal Rowitch (53 pgs, including index).

* cited by examiner



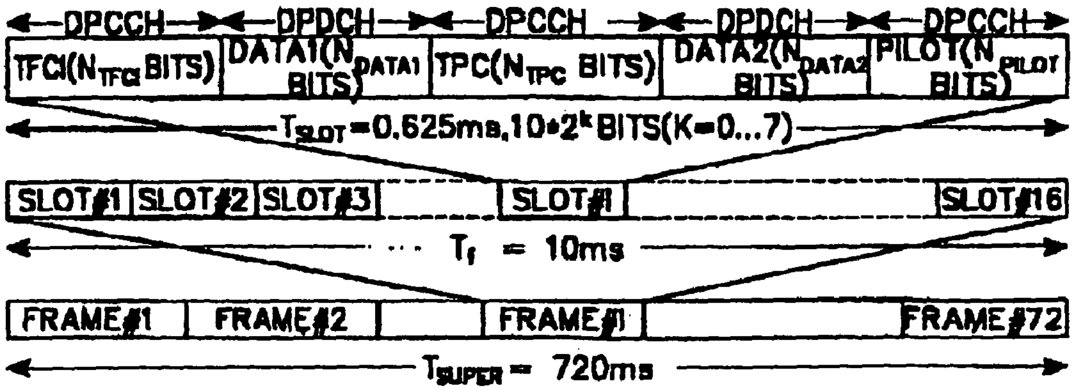
FRAME STRUCTURE FOR UPLINK DPDCH/DPCCH

FIG.1A
(PRIOR ART)



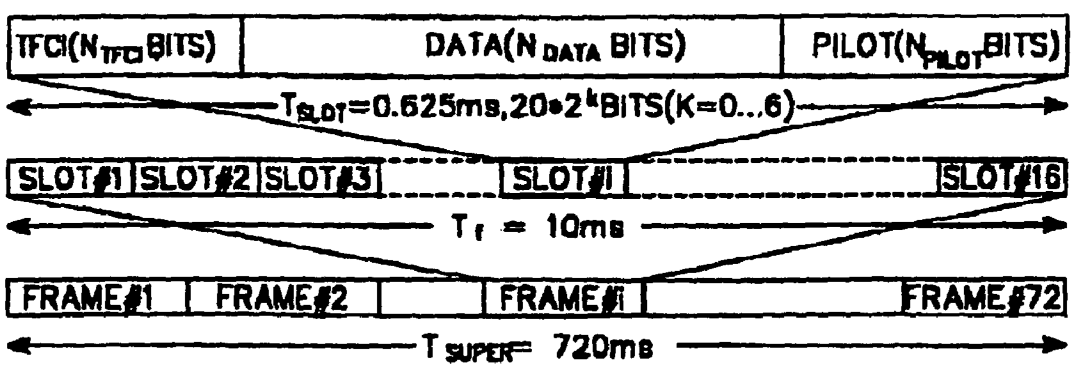
STUCTURE OF THE RANDOM-ACCESS MESSAGE PART

FIG.1B
(PRIOR ART)



FRAME STRUCTURE OF DOWNLINK DPCH

FIG. 1C
(PRIOR ART)



FRAME STRUCTURE FOR SECONDARY COMMON CONTROL PHYSICAL CHANNEL

FIG. 1D
(PRIOR ART)

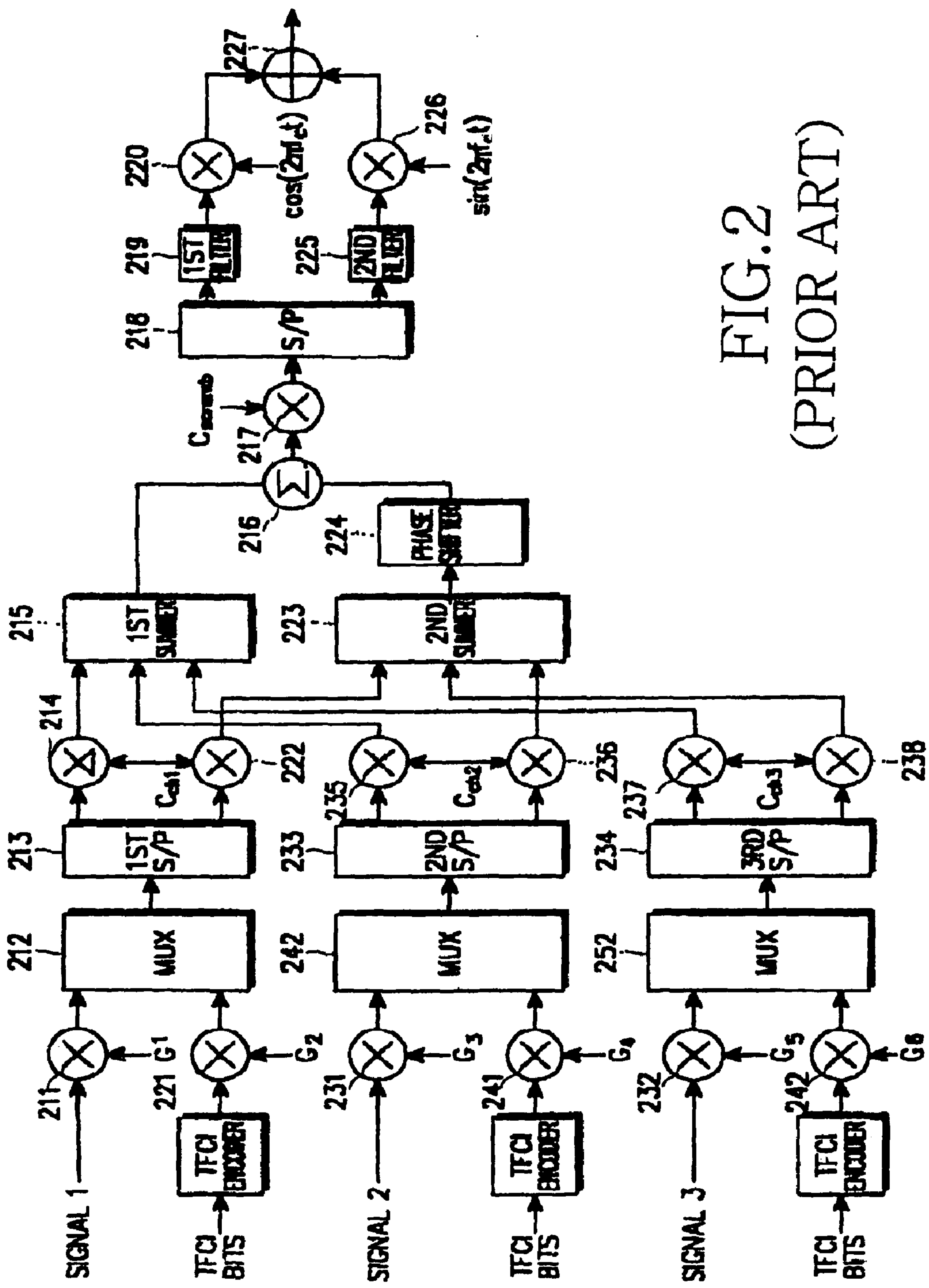


FIG. 2
(PRIOR ART)

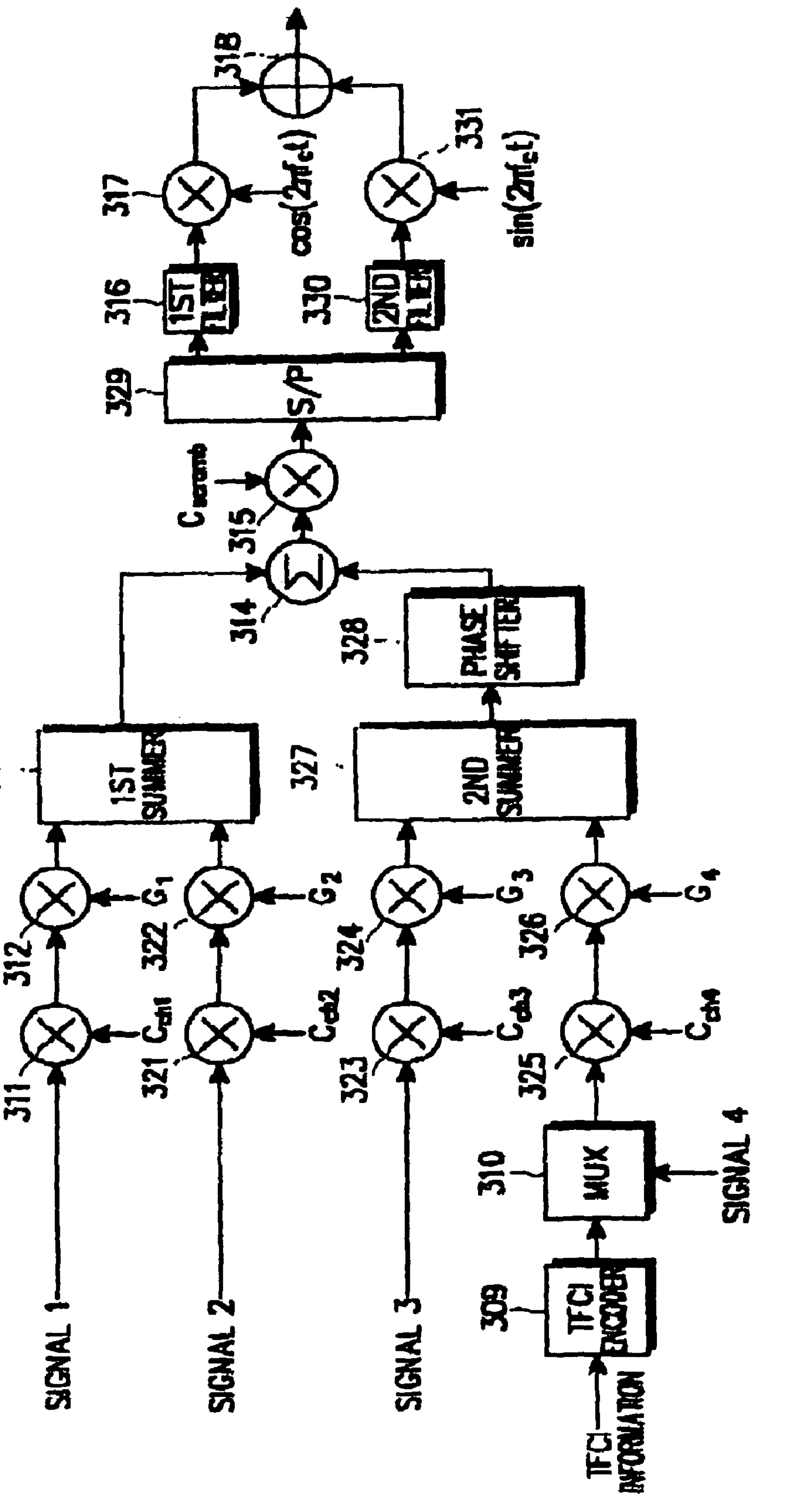


FIG.3
(PRIOR ART)

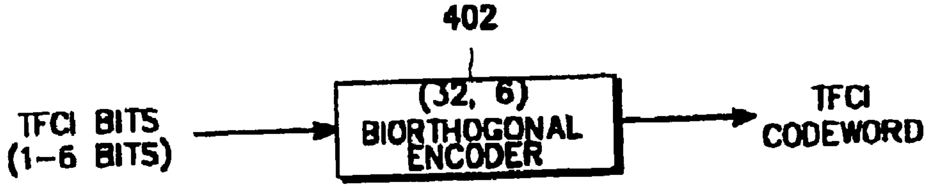


FIG.4A
(PRIOR ART)

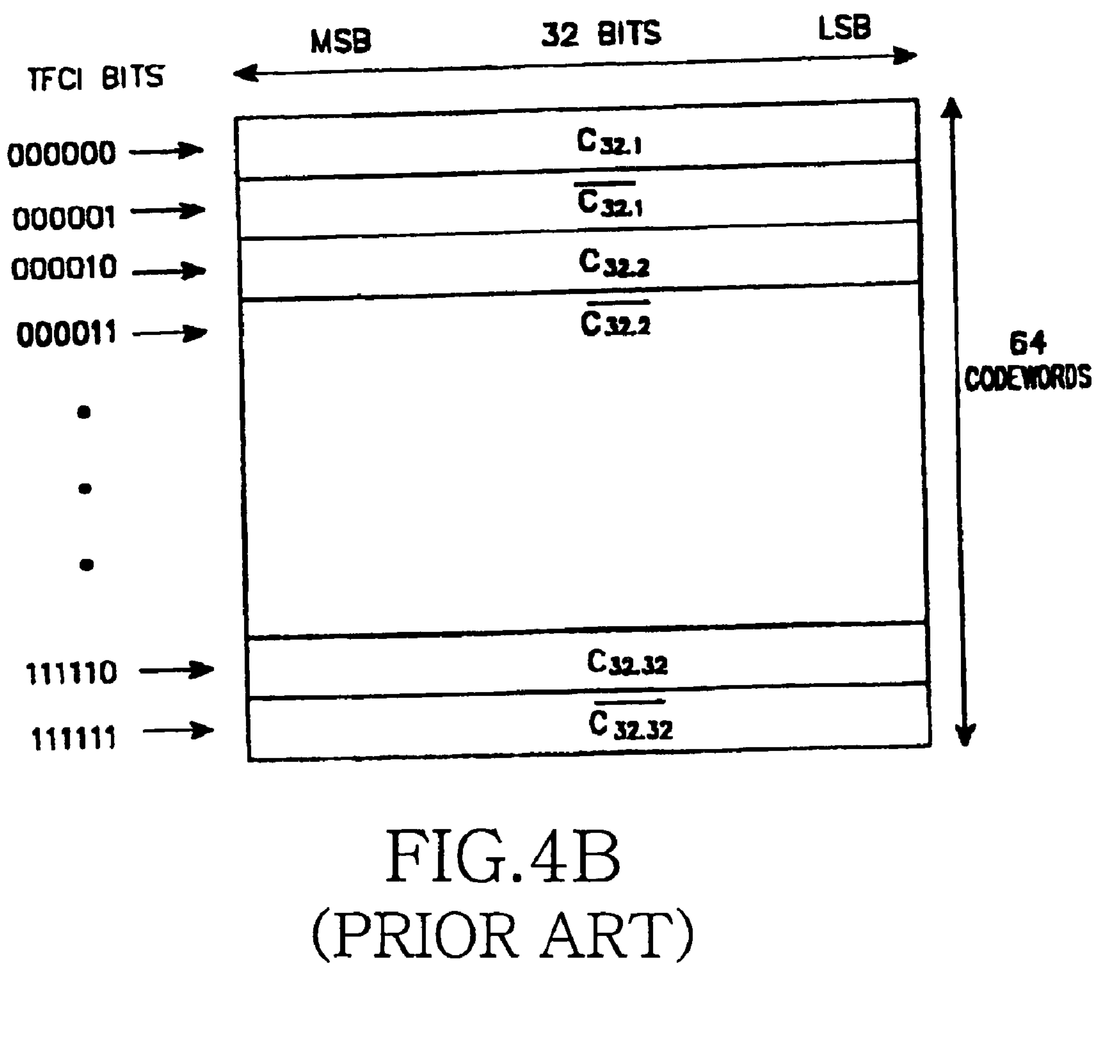


FIG.4B
(PRIOR ART)

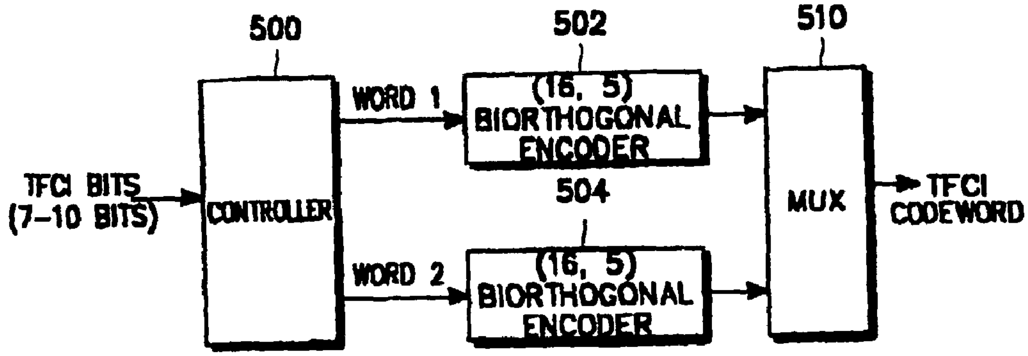


FIG.5A
(PRIOR ART)

TFCI = $a_{10}a_9a_8a_7a_6a_5a_4a_3a_2a_1$

n = (MAXIMUM INTEGER EQUAL TO OR SMALLER THAN(TFCI)
IF TFCI < n^2+ n
THEN WORD1 = n ; WORD2 = TFCI−n^2
ELSE WORD1 = TFCI − n^2; WORD2 = n

FIG.5B
(PRIOR ART)

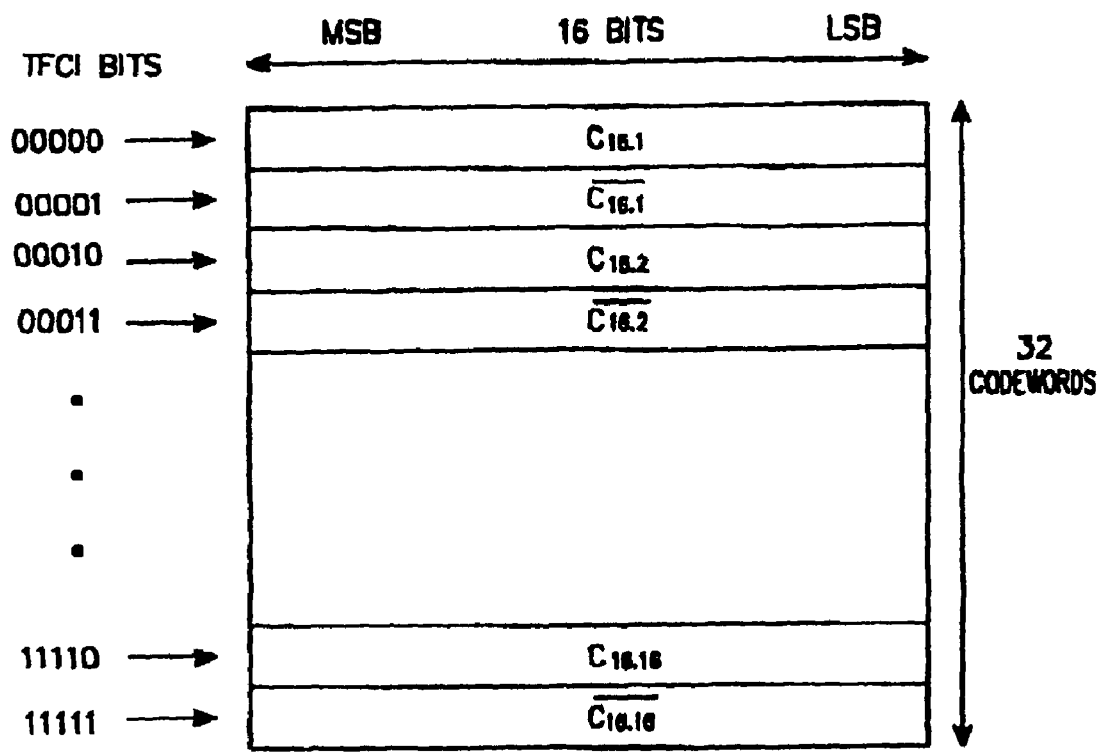


FIG.5C
(PRIOR ART)

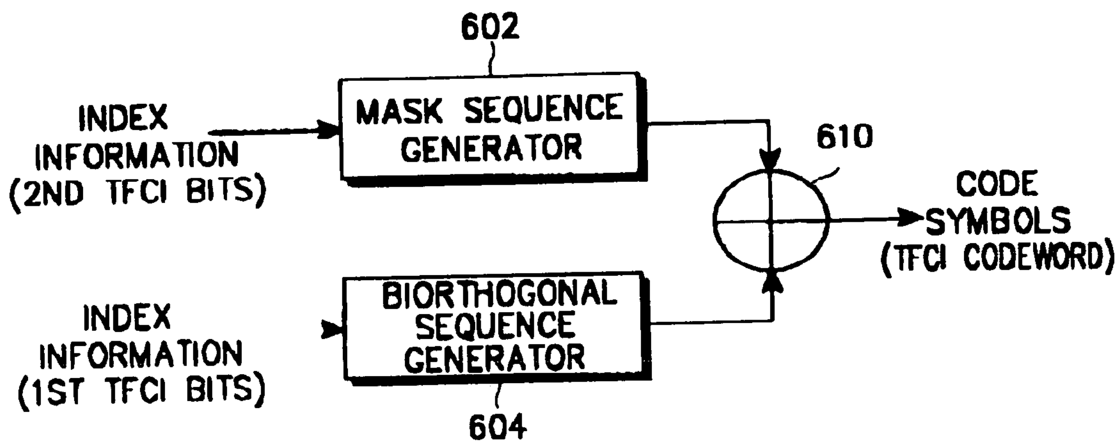

602
INDEX INFORMATION (2ND TFCI BITS)
MASK SEQUENCE GENERATOR
610
CODE SYMBOLS (TFCI CODEWORD)
INDEX INFORMATION (1ST TFCI BITS)
BIORTHOGONAL SEQUENCE GENERATOR
604

FIG. 6

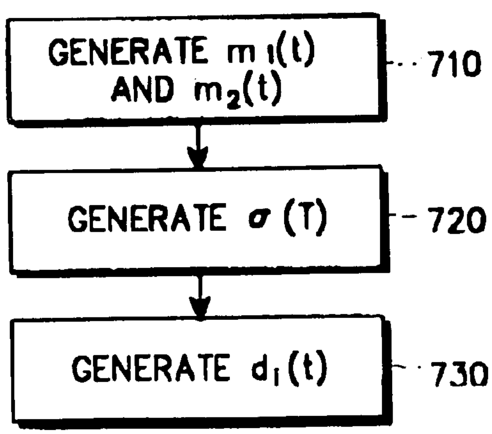
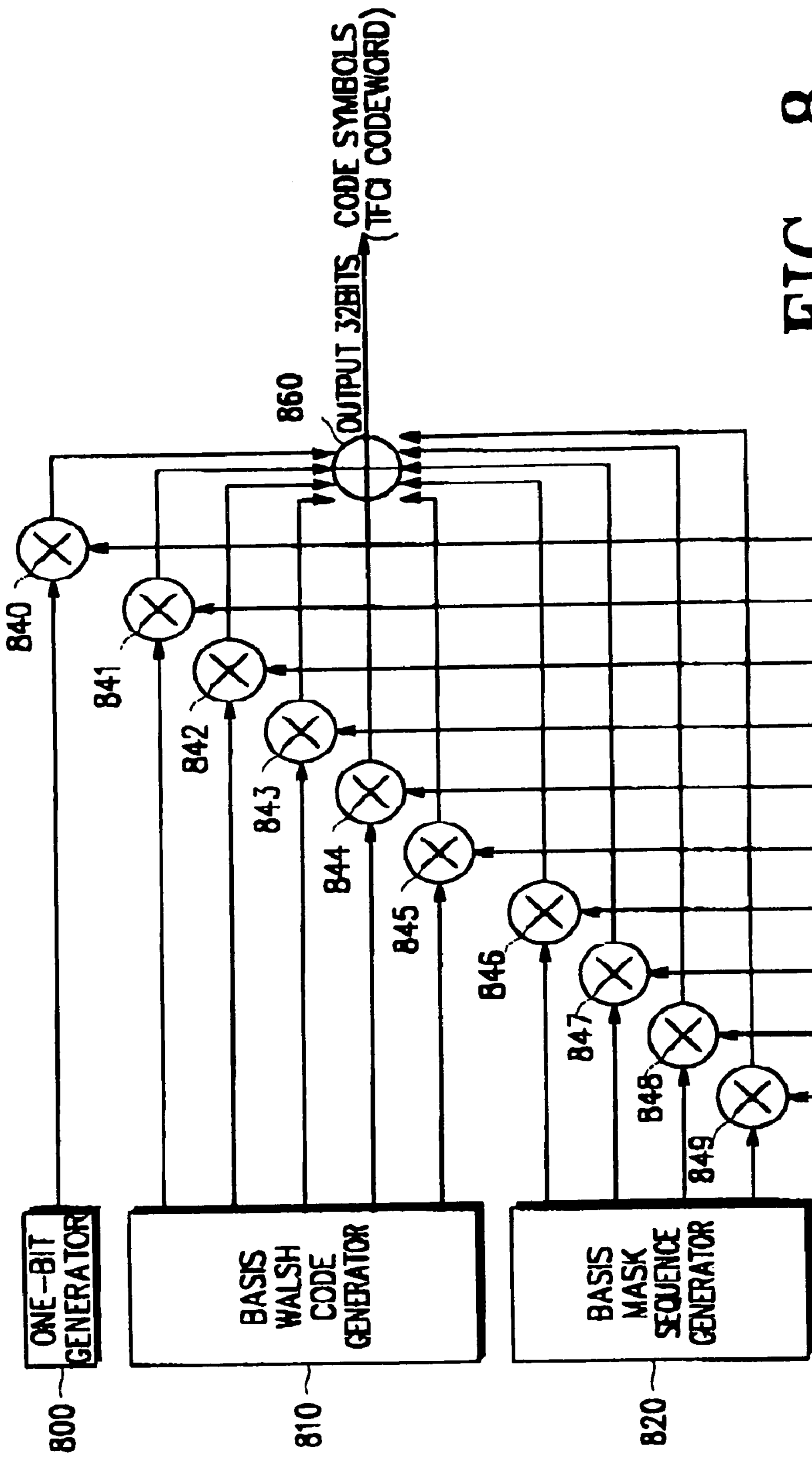
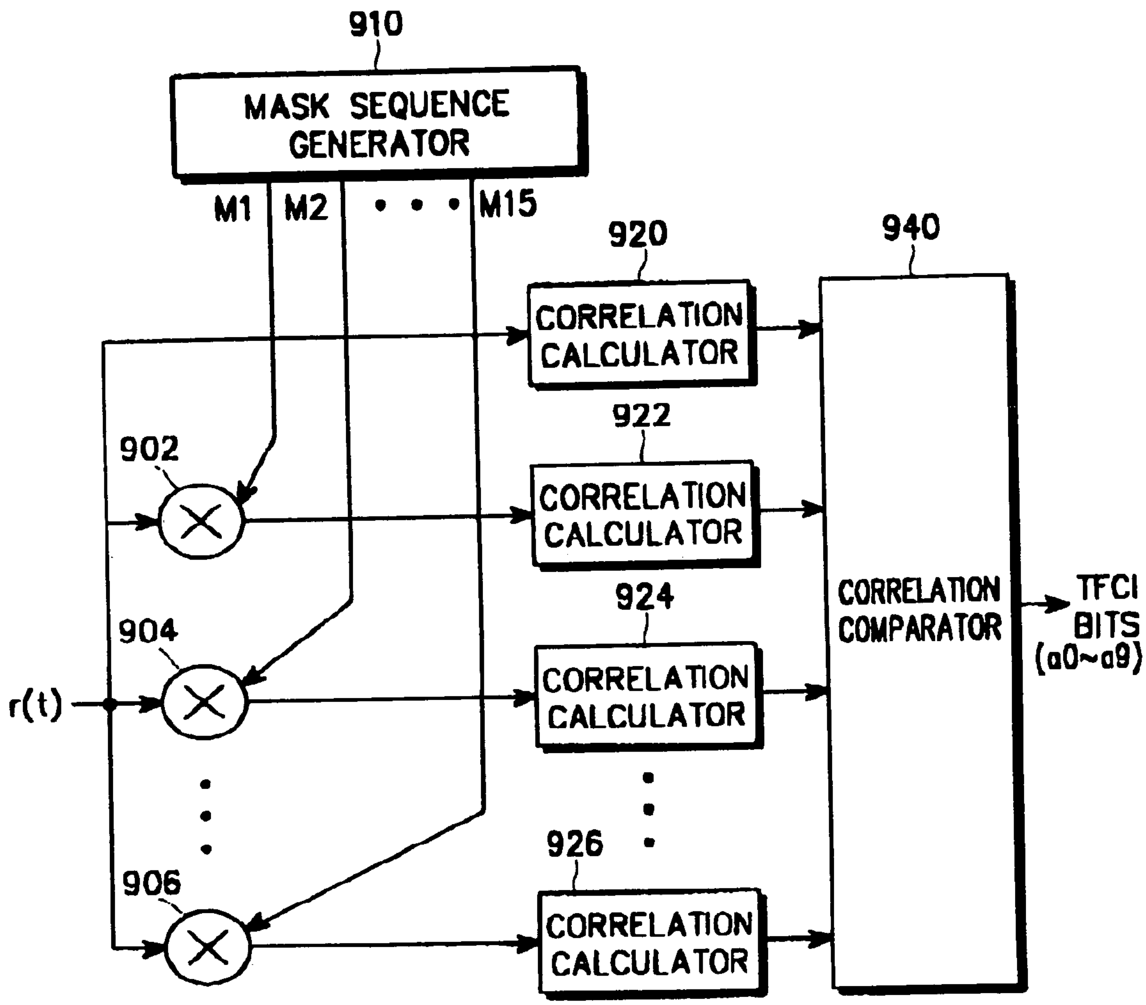

GENERATE m1(t) AND m2(t)
710
GENERATE σ (T)
720
GENERATE d1(t)
730

FIG. 7



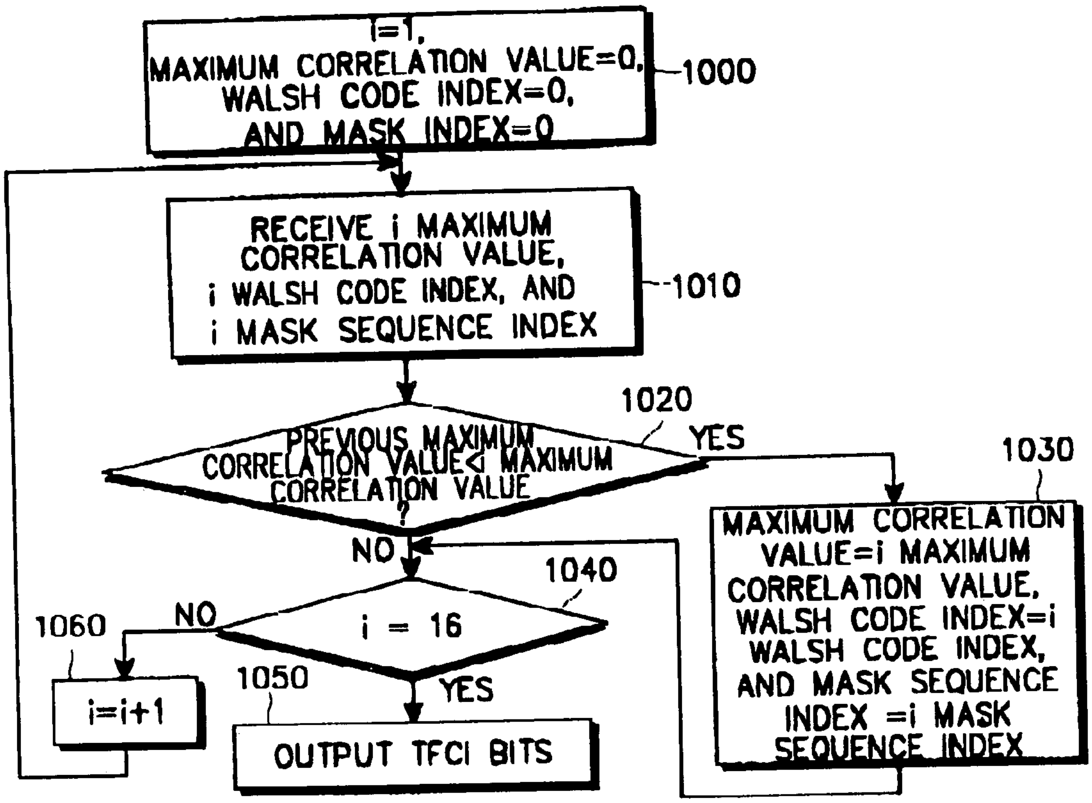
800
ONE-BIT GENERATOR
810
BASIS WALSH CODE GENERATOR
820
BASIS MASK SEQUENCE GENERATOR
840
841
842
843
844
845
846
847
848
849
a0
a1
a2
a3
a4
a5
a6
a7
a8
a9
860
OUTPUT 32BITS CODE SYMBOLS (TFCI CODEWORD)

FIG. 8



910
MASK SEQUENCE GENERATOR
M1
M2
M15
920
CORRELATION CALCULATOR
922
CORRELATION CALCULATOR
924
CORRELATION CALCULATOR
926
CORRELATION CALCULATOR
940
CORRELATION COMPARATOR
TFCI BITS (a0~a9)
902
904
906
r(t)

FIG. 9



i=1,
MAXIMUM CORRELATION VALUE=0,
WALSH CODE INDEX=0,
AND MASK INDEX=0
1000
RECEIVE i MAXIMUM CORRELATION VALUE,
i WALSH CODE INDEX, AND
i MASK SEQUENCE INDEX
1010
PREVIOUS MAXIMUM CORRELATION VALUE<i MAXIMUM CORRELATION VALUE ?
1020
YES
NO
1030
MAXIMUM CORRELATION VALUE=i MAXIMUM CORRELATION VALUE,
WALSH CODE INDEX=i WALSH CODE INDEX,
AND MASK SEQUENCE INDEX =i MASK SEQUENCE INDEX
1040
i = 16
NO
YES
1060
i=i+1
1050
OUTPUT TFCI BITS

FIG. 10

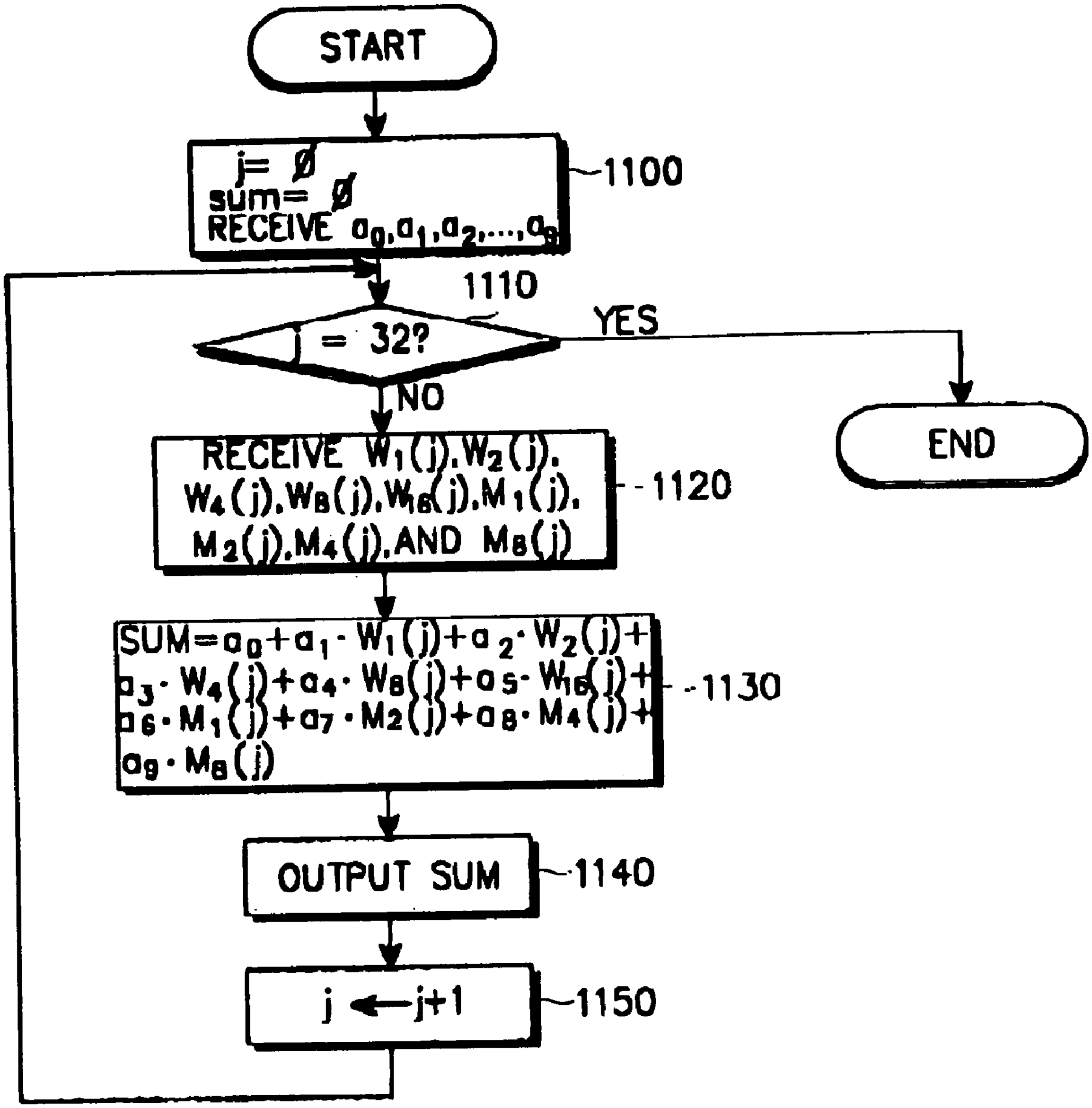

START
j= Ø
sum= Ø
RECEIVE a0,a1,a2,...,a9
1100
1110
j = 32?
YES
NO
END
RECEIVE W1(j),W2(j),
W4(j),W8(j),W16(j),M1(j),
M2(j),M4(j),AND M8(j)
1120
SUM=a0+a1·W1(j)+a2·W2(j)+
a3·W4(j)+a4·W8(j)+a5·W16(j)+
a6·M1(j)+a7·M2(j)+a8·M4(j)+
a9·M8(j)
1130
OUTPUT SUM
1140
j ←j+1
1150

FIG. 11

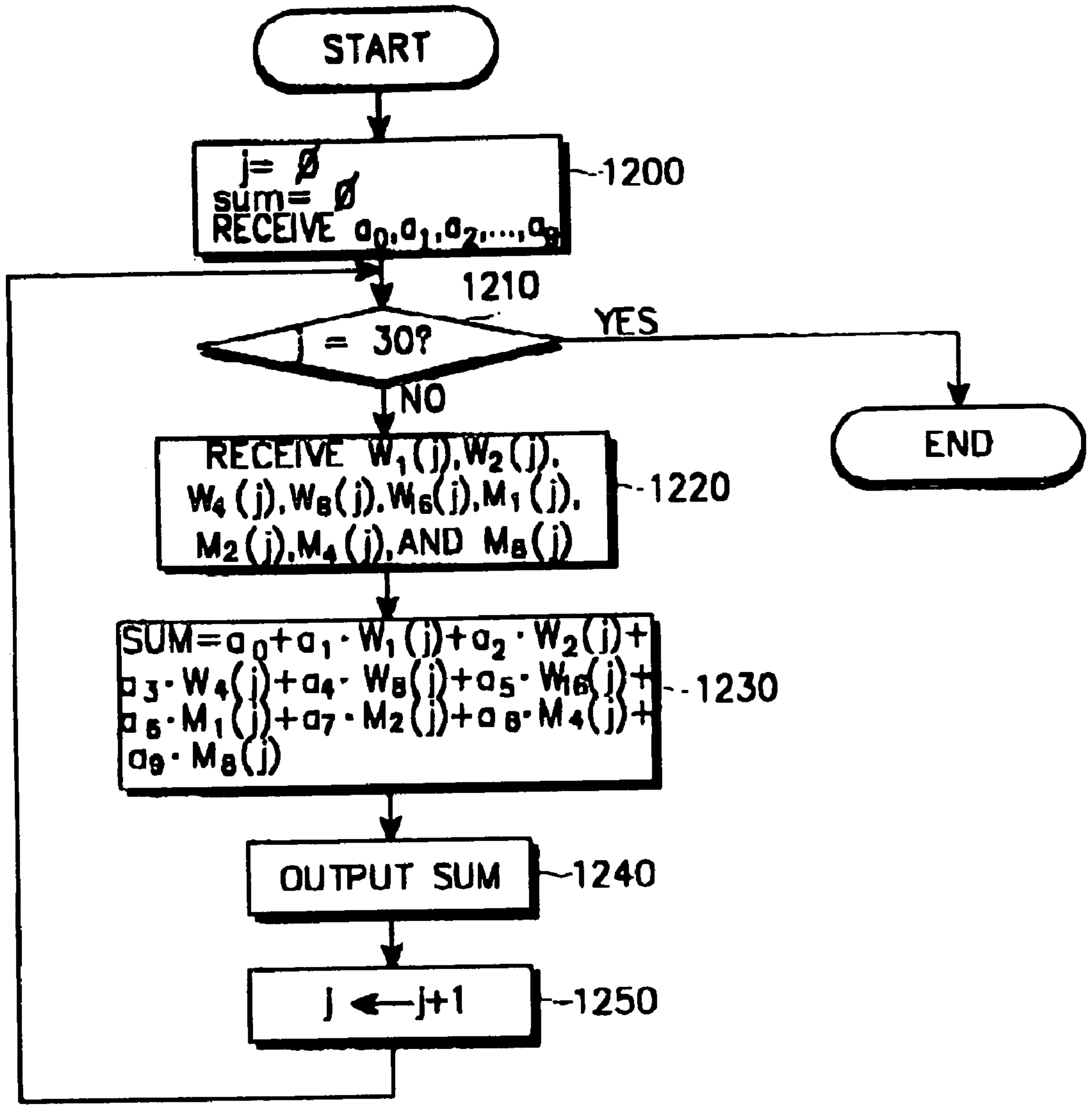

START
j= Ø
sum= Ø
RECEIVE a0,a1,a2,...,a9
1200
1210
j = 30?
YES
NO
END
RECEIVE W1(j),W2(j),
W4 (j),W8(j),W16(j),M1 (j),
M2 (j),M4 (j),AND M8(j)
1220
SUM=a0+a1·W1(j)+a2·W2(j)+
a3·W4(j)+a4·W8(j)+a5·W16(j)+
a6·M1(j)+a7·M2(j)+a8·M4(j)+
a9·M8(j)
1230
OUTPUT SUM
1240
j ←j+1
1250

FIG. 12

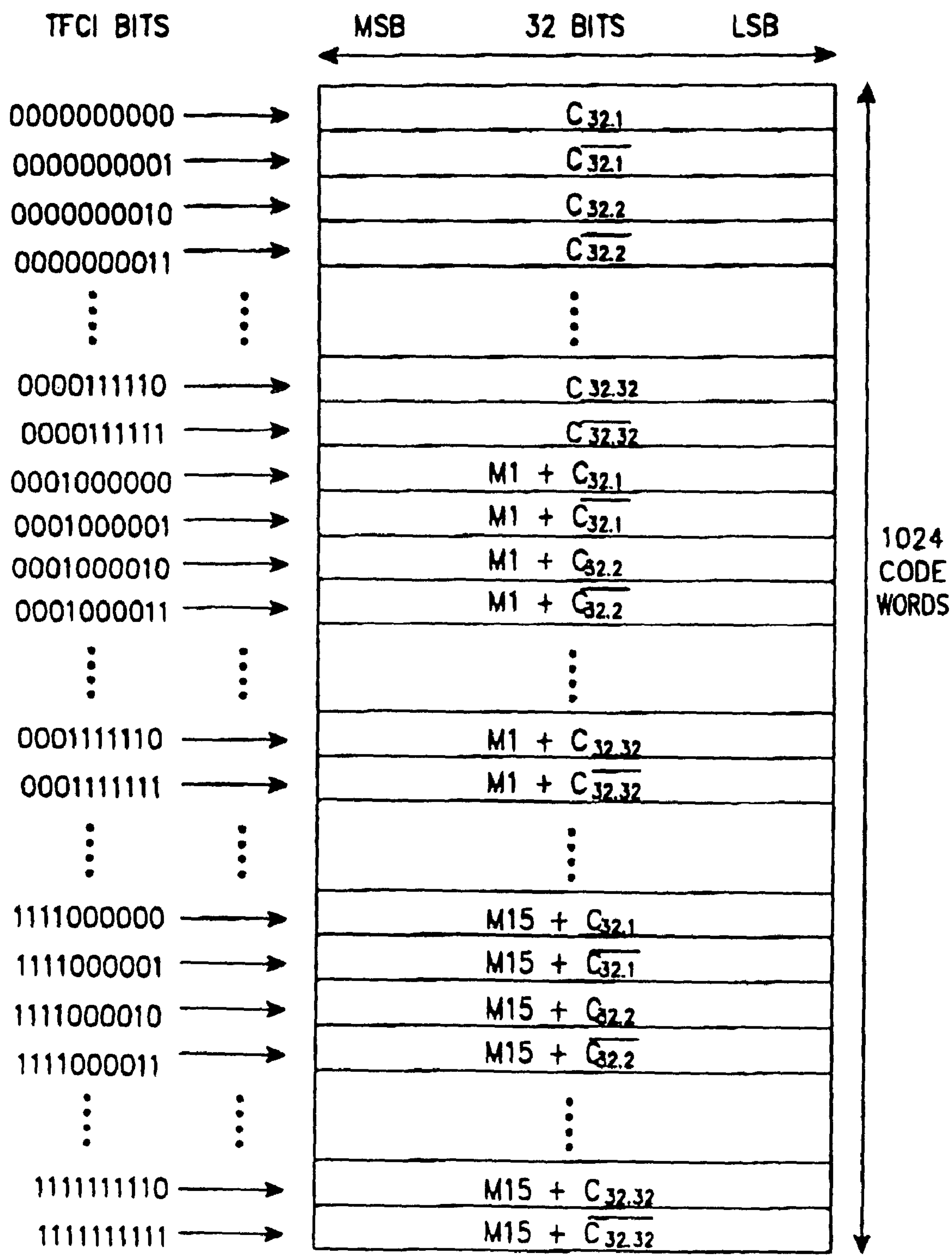

TFCI BITS
MSB
32 BITS
LSB
0000000000
0000000001
0000000010
0000000011
0000111110
0000111111
0001000000
0001000001
0001000010
0001000011
0001111110
0001111111
1111000000
1111000001
1111000010
1111000011
1111111110
1111111111
C32.1
C32.2
C32.32
M1 + C32.1
M1 + C32.2
M1 + C32.32
M15 + C32.1
M15 + C32.2
M15 + C32.32
1024 CODE WORDS

FIG. 13

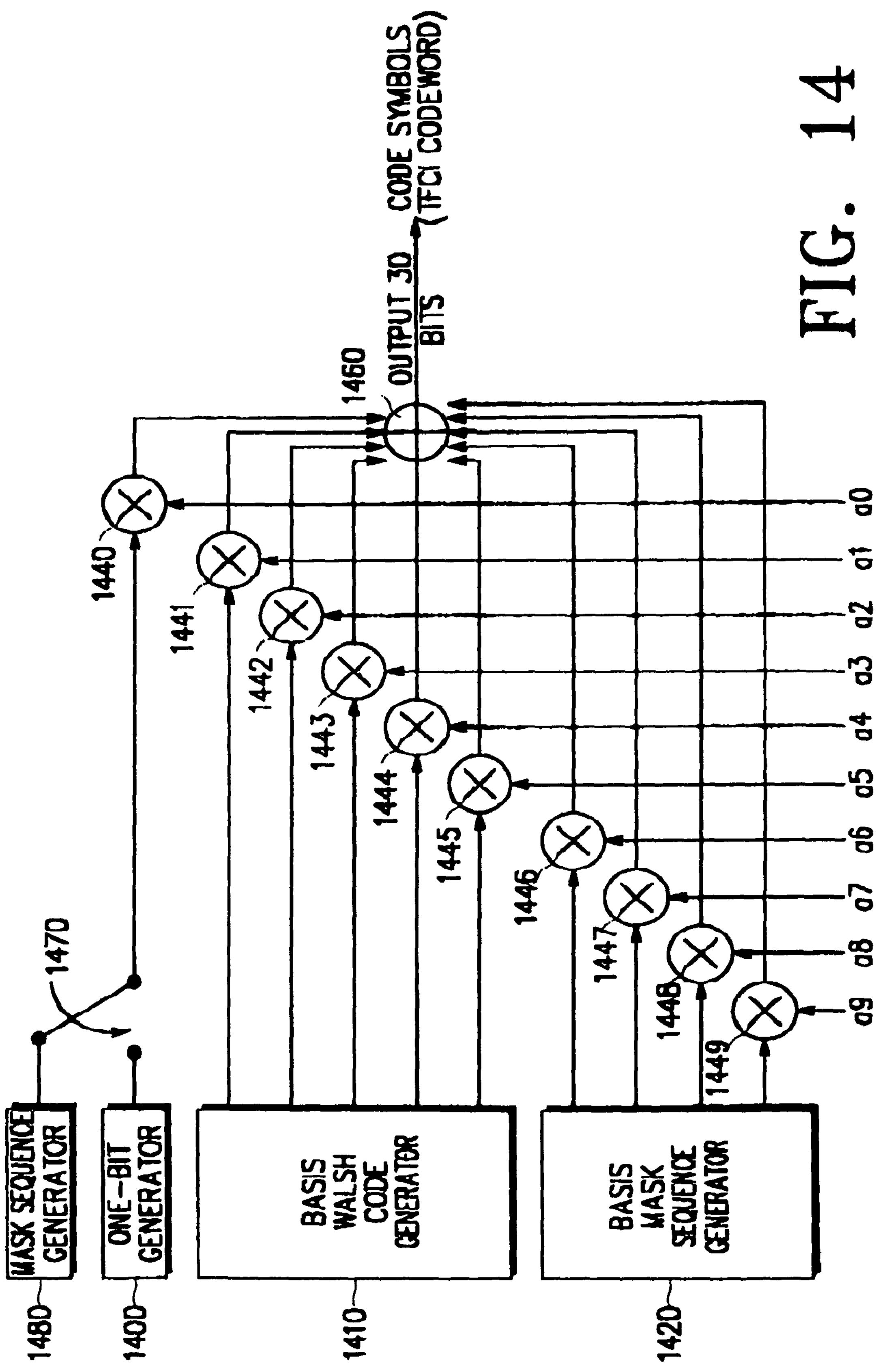

MASK SEQUENCE GENERATOR
1480
ONE-BIT GENERATOR
1400
BASIS WALSH CODE GENERATOR
1410
BASIS MASK SEQUENCE GENERATOR
1420
1470
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1460
OUTPUT 30 BITS
CODE SYMBOLS (TFCI CODEWORD)
a0
a1
a2
a3
a4
a5
a6
a7
a8
a9

FIG. 14

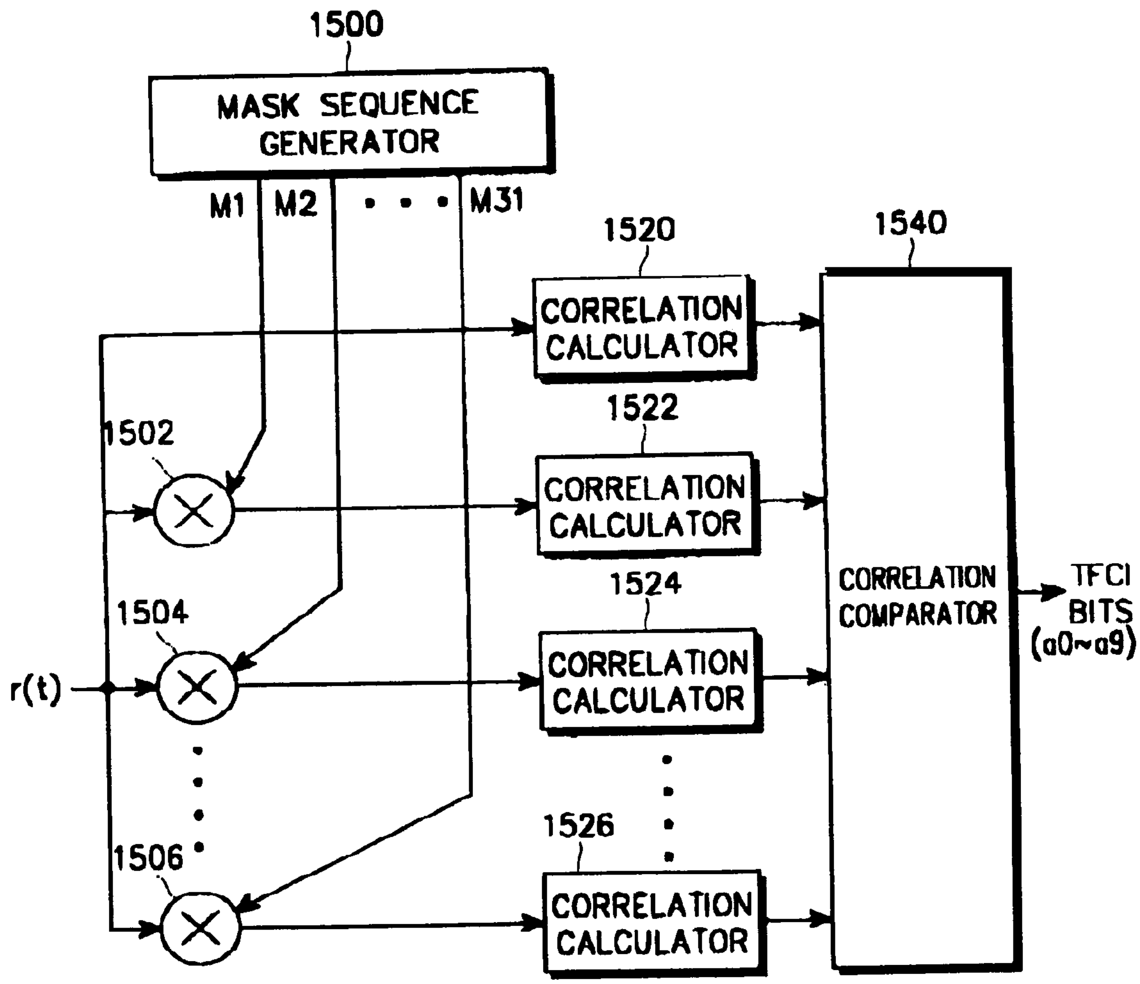
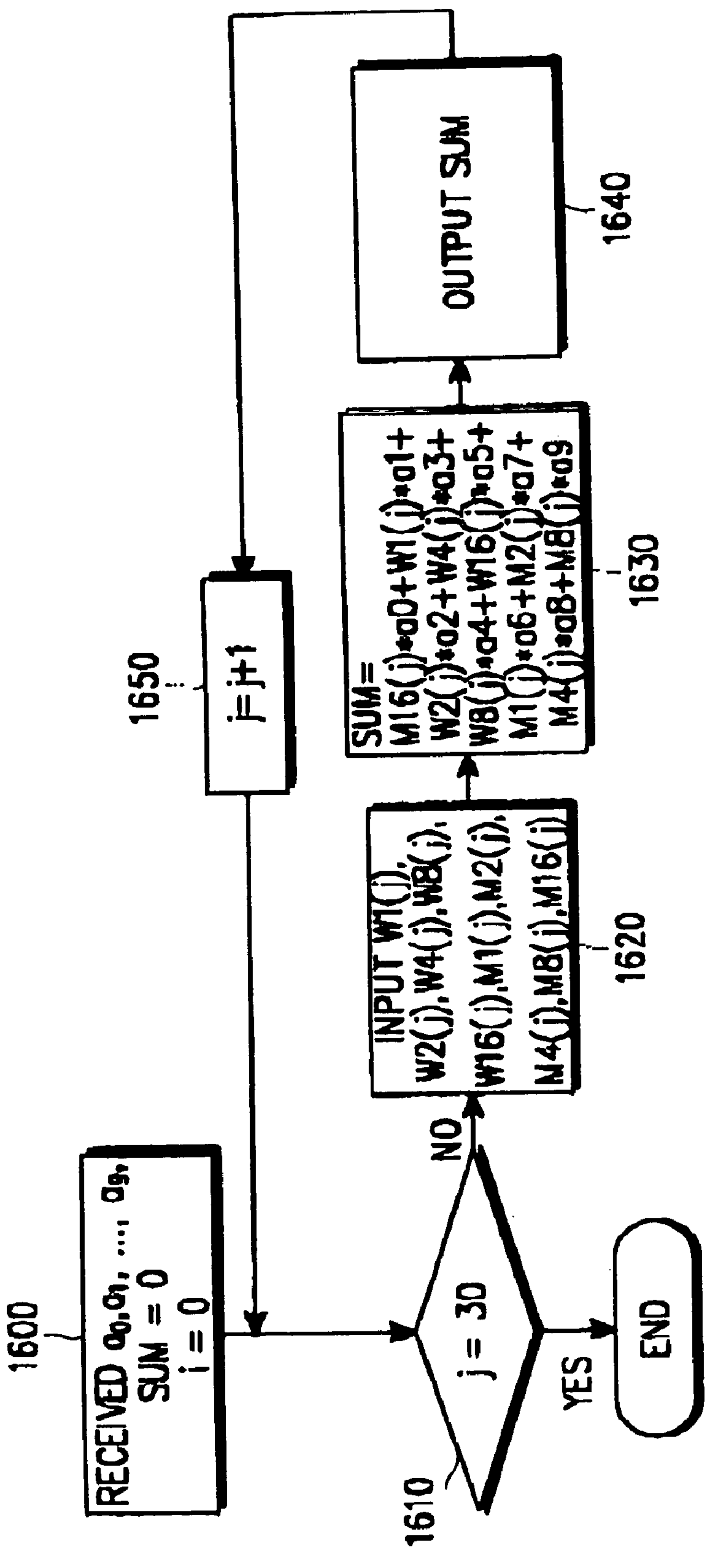

1500
MASK SEQUENCE GENERATOR
M1
M2
M31
1520
CORRELATION CALCULATOR
1540
1502
1522
CORRELATION CALCULATOR
1504
1524
CORRELATION COMPARATOR
TFCI BITS (a0~a9)
r(t)
CORRELATION CALCULATOR
1526
1506
CORRELATION CALCULATOR

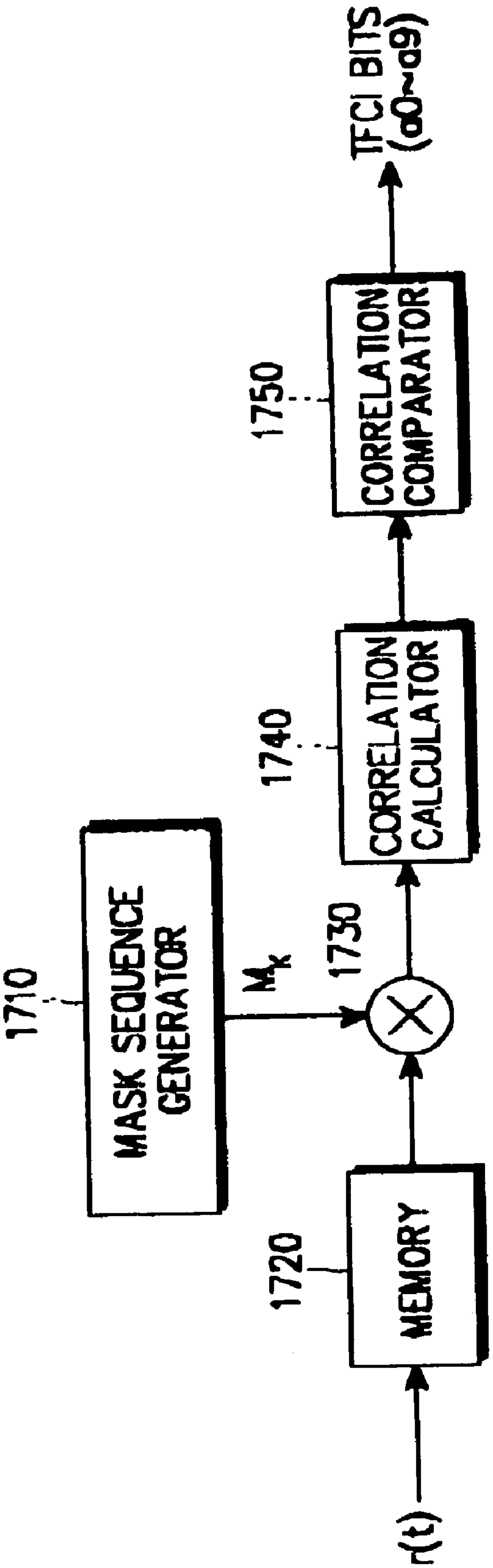
FIG. 15

FIG. 16



RECEIVED a0, a1, ..., a9, SUM = 0 j = 0
1600
j = 30
1610
YES
NO
END
INPUT W1(j), W2(j), W4(j), W8(j), W16(j), M1(j), M2(j), M4(j), M8(j), M16(j)
1620
SUM= M16(j)*a0+W1(j)*a1+ W2(j)*a2+W4(j)*a3+ W8(j)*a4+W16(j)*a5+ M1(j)*a6+M2(j)*a7+ M4(j)*a8+M8(j)*a9
1630
OUTPUT SUM
1640
j=j+1
1650

FIG. 17


r(t)
1720
MEMORY
1710
MASK SEQUENCE GENERATOR
M_k
1730
1740
CORRELATION CALCULATOR
1750
CORRELATION COMPARATOR
TFCI BITS (a0~a9)

APPARATUS AND METHOD FOR ENCODING/DECODING TRANSPORT FORMAT COMBINATION INDICATOR IN CDMA MOBILE COMMUNICATION SYSTEM

PRIORITY

This application is a continuation of application Ser. No. 09/611,069, filed Jul. 6, 2000, now U.S. Pat. No. 6,882,636 which claims priority under 35 U.S.C. §119 to an application entitled "Apparatus And Method For Encoding/Decoding Transport Format Combination Indicator In CDMA Mobile Communication System" filed in the Korean Intellectual Property Office on Jul. 6, 1999 and assigned Serial No. 1999-27932, the contents of each of which are hereby incorporated by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to an information transmitting apparatus and method in an IMT 2000 system, and in particular, to an apparatus and method for transmitting a transport format combination indicator (TFCI).

2. Description of the Related Art

A CDMA mobile communication system (hereinafter, referred to as an IMT 2000 system) generally transmits frames that provide a voice service, an image service, a character service on a physical channel such as a dedicated physical data channel (DPDCH) at a fixed or variable data rate. In the case where the data frames which include that sort of services are transmitted at a fixed data rate, there is no need to inform a receiver of the spreading rate of each data frame. On the other hand, if the data frames are transmitted at a variable data rate, which implies that each data frame has a different data rate, a transmitter should inform the receiver of the spreading rate of each data frame determined by its data rate. A data rate is proportional to a data transmission rate and the data transmission rate is inversely proportional to a spreading rate in a general IMT 2000 system.

For transmission of data frames at a variable data rate, a TFCI field of a DPCCH informs a receiver of the data rate of the current service frame. The TFCI field includes a TFCI indicating a lot of information including the data rate of a service frame. The TFCI is information that helps a voice or data service to reliably be provided.

FIGS. **1**A to **1**D illustrate examples of applications of a TFCI. FIG. **1**A illustrates application of the TFCI to an uplink DPDCH and an uplink dedicated physical control channel (DPCCH). FIG. **1**B illustrates application of the TFCI to a random access channel (RACH). FIG. **1**C illustrates application of the TFCI to a downlink DPDCH and a downlink DPCCH. FIG. **1**D illustrates application of the TFCI to a secondary common control physical channel (SCCPCH).

Referring to FIGS. **1**A to **1**D, one frame is comprised of 16 slots and each slot has a TFCI field. Thus, one frame includes 16 TFCI fields. A TFCI field includes N_{TFCI} bits and a TFCI generally has 32 bits in a frame. To transmit the 32-bit TFCI in one frame, 2 TFCI bits can be assigned to each of the 16 slots (T_{slot}=0.625 ms).

FIG. **2** is a block diagram of a base station transmitter in a general IMT 2000 system.

Referring to FIG. **2**, multipliers **211**, **231**, and **232** multiply input signals by gain coefficients G_1, G_3, and G_5. Multipliers **221**, **241**, and **242** multiply TFCI codewords (TFCI code symbols) received from corresponding TFCI encoders by gain coefficients G_2, G_4, and G_6. The gain coefficients G_1 to G_6 may have different values according to service types or handover situations. The input signals include pilots and power control signals (TPCs) of a DPCCH and a DPDCH data. A multiplexer **212** inserts 32 bit TFCI code symbols (TFCI codeword) received from the multiplier **221** into the TFCI fields as shown in FIG. **1**C. A multiplexer **242** inserts 32-bit TFCI code symbols received from the multiplier **241** into the TFCI fields. A multiplexer **252** inserts 32-bit TFCI code symbols received from the multiplier **242** into the TFCI fields. Insertion of TFCI code symbols into TFCI fields is shown in FIGS. **1**A to **1**D. The 32 code symbols are obtained by encoding TFCI bits (information bits) that define the data rate of a data signal on a corresponding data channel. 1^{st}, 2^{nd}, and 3^{rd} serial to parallel converters (S/Ps) **213**, **233**, and **234** separate the outputs of the multiplexers **212**, **242**, and **252** into I channels and Q channels. Multipliers **214**, **222**, and **235** to **238** multiply the outputs of the S/Ps **213**, **233**, and **234** by channelization codes C_{ch1}, C_{ch2}, and C_{ch3}. The channelization codes are orthogonal codes. A first summer **215** sums the outputs of the multipliers **214**, **235**, and **237** and generates an I channel signal and a second summer **223** sums the outputs of the multipliers **222**, **236**, and **238** and generates a Q channel signal. A phase shifter **224** shifts the phase of the Q channel signal received from the second summer **223** by 90°. A summer **216** adds the outputs of the first summer **215** and the phase shifter **224** and generates a complex signal I+jQ. A multiplier **217** scrambles the complex signal with a complex PN sequence C_{scramb} assigned to the base station. A signal processor (S/P) **218** separates the scrambled signal into an I channel and a Q channel. Low-pass filters (LPFs) **219** and **225** limits the bandwidths of the I channel and Q channel signals received from the S/P **218** by low-pass-filtering. Multipliers **220** and **226** multiply the outputs of the LPFs **219** and **225** by carriers $\cos(2\pi f_c t)$ and $\sin(2\pi f_c t)$, respectively, thereby transforming the outputs of the LPFs **219** and **225** to an RF (Radio Frequency) band. A summer **227** sums the RF I channel and Q channel signals.

FIG. **3** is a block diagram of a mobile station transmitter in the general IMT 2000 system.

Referring to FIG. **3**, multipliers **311**, **321**, and **323** multiply corresponding signals by channelization codes C_{ch1}, C_{ch2}, and C_{ch3}. Signals **1**, **2**, **3** are first, second and third DPDCH signal. An input signal **4** includes pilots and TPCs of a DPCCH.TFCI information bits are encoded into 32 bit TFCI code symbols by a TFCI encoder **309**. A multiplier **310** inserts a 32 bit TFCI code symbols into the signal **4** as shown in FIG. **1**A. A multiplier **325** multiplies a DPCCH signal which include TFCI code symbol received from the multiplier **310** by a channelization code C_{ch4}. The channelization codes C_{ch1} to C_{ch4} are orthogonal codes. The 32 TFCI code symbols are obtained by encoding TFCI information bits that define the data rate of the DPDCH signals. Multipliers **312**, **322**, **324**, and **326** multiply the outputs of the multipliers **311**, **321**, **323**, and **325** by gain coefficients G_1 to G_4, respectably. The gain coefficients G_1 to G_4 may have different values. A first summer **313** generates an I channel signal by adding the outputs of the multipliers **312** and **322**. A second summer **327** generates a Q channel signal by adding the outputs of the multipliers **324** and **326**. A phase shifter **328** shifts the phase of the Q channel signal received from the second summer **327** by 90°. A summer **314** adds the outputs of the first summer **313** and the phase shifter **328** and generates a complex signal I+jQ. A multiplier **315** scrambles the complex signal with a PN sequence C_{scramb} assigned to a base station. An S/P **329** divides the scrambled signal into an I channel and a Q channel. LPFs **316** and **330** low-pass-filter the I channel and Q channel signals received from the S/P **329** and generate signals with limited bandwidths. Multipliers **317** and **331** multiply the outputs of the LPFs **316** and **330** by carriers cos ($2\pi f_c t$) and sin($2\pi f_c t$), respectively, thereby transforming the outputs of the LPFs **316** and **330** to an RF band. A summer **318** sums the RF I channel and Q channel signals.

TFCIs are categorized into a basic TFCI and an extended TFCI. The basic TFCI represents 1 to 64 different information including the data rates of corresponding data channels using 6 TFCI information bits, whereas the extended TFCI represents 1 to 128, 1 to 256, 1 to 512, or 1 to 1024 different information using 7, 8, 9 or 10 TFCI information bits. The extended TFCI has been suggested to satisfy the requirement of the IMT 2000 system for more various services. TFCI bits are essential for a receiver to receive data frames received from a transmitter. That is the reason why unreliable transmission of the TFCI information bits due to transmission errors lead to wrong interpretation of the frames in the receiver. Therefore, the transmitter encodes the TFCI bits with an error correcting code prior to transmission so that the receiver can correct possibly generated errors in the TFCI.

FIG. **4**A conceptionally illustrates a basic TFCI bits encoding structure in a conventional IMT 2000 system and FIG. **4**B is an exemplary encoding table applied to a biorthogonal encoder shown in FIG. **4**A. As stated above, the basic TFCI has 6 TFCI bits (hereinafter, referred to as basic TFCI bits) that indicate 1 to 64 different information.

Referring to FIGS. **4**A and **4**B, a biorthogonal encoder **402** receives basic TFCI bits and outputs 32 coded symbols (TFCI codeword or TFCI code symbol). The basic TFCI is basically expressed in 6 bits. Therefore, in the case where a basic TFCI bits of less than 6 bits are applied to the biorthogonal encoder **402**, 0s are added to the left end, i.e., MSB (Most Significant Bit) of the basic TFCI bits to increase the number of the basic TFCI bits to 6. The biorthogonal encoder **402** has a predetermined encoding table as shown in FIG. **4**B to output 32 coded symbols for the input of the 6 basic TFCI bits. As shown in FIG. **4**B, the encoding table lists 32(32-symbol) orthogonal codewords $c_{32.1}$ to $c_{32.32}$ and 32 biorthogonal codewords $\overline{c_{32.1}}$ to $\overline{c_{32.32}}$ that are the complements of the codewords $c_{32.1}$ to $c_{32.32}$. If the LSB (Least Significant Bit) of the basic TFCI is 1, the biorthogonal encoder **402** selects out of the 32 biorthogonal codewords. If the LSB is 0, the biorthogonal encoder **402** selects out of the 32 orthogonal codewords. One of the selected orthogonal codewords or biorthogonal codewords is then selected based on the other TFCI bits.

A TFCI codeword should have powerful error correction capability as stated before. The error correction capability of binary linear codes depends on the minimum distance (dmin) between the binary linear codes. A minimum distance for optimal binary linear codes is described in "An Updated Table of Minimum-Distance Bounds for Binary Linear Codes", A. E. Brouwer and Tom Verhoeff, IEEE Transactions on Information Theory, vol. 39, No. 2, March 1993 (hereinafter, referred to as reference 1).

Reference 1 gives 16 as a minimum distance for binary linear codes by which 32 bits are output for the input of 6 bits. TFCI codewords output from the biorthogonal encoder **402** has a minimum distance of 16, which implies that the TFCI codewords are optimal codes.

FIG. **5**A conceptionally illustrates an extended TFCI bits encoding structure in the conventional IMT 2000 system, FIG. **5**B is an exemplary algorithm of distributing TFCI bits in a controller shown in FIG. **5**A, and FIG. **5**C illustrates an exemplary encoding table applied to biorthogonal encoders shown in FIG. **5**A. An extended TFCI is also defined by the number of TFCI bits. That is, the extended TFCI includes 7, 8, 9 or 10 TFCI bits (hereinafter, referred to as extended TFCI bits) that represent 1 to 128, 1 to 256, 1 to 512, or 1 to 1024 different information, as stated before.

Referring to FIGS. **5**A, **5**B, and **5**C, a controller **500** divides TFCI bits into two halves. For example, for the input of 10 extended TFCI bits, the controller **500** outputs the first half of the extended TFCI as first TFCI bits (word **1**) and the last half as second TFCI bits (word **2**). The extended TFCI are basically expressed in 10 bits. Therefore, in the case where an extended TFCI bits of less than 10 bits are input, the controller **500** adds 0s to the MSB of the extended TFCI bits to represent the extended TFCI in 10 bits. Then, the controller **500** divides the 10 extended TFCI bits into word **1** and word **2**. Word **1** and word **2** are fed to biorthogonal encoders **502** and **504**, respectively. A method of separating the extended TFCI bits a_1 to a_{10} into word **1** and word **2** is illustrated in FIG. **5**B.

The biorthogonal encoder **502** generates a first TFCI codeword having 16 symbols by encoding word **1** received from the controller **500**. The biorthogonal encoder **504** generates a second TFCI codeword having 16 symbols by encoding word **2** received from the controller **500**. The biorthogonal encoders **502** and **504** have predetermined encoding tables to output the 16-symbol TFCI codewords for the two 5-bit TFCI inputs (word **1** and word **2**). An exemplary encoding table is illustrated in FIG. **5**C. As shown in FIG. **5**C, the encoding table lists 16 orthogonal codewords of length 16 bits $c_{16.1}$ to $c_{16.16}$ and biorthogonal codewords $\overline{c_{16.1}}$ to $\overline{c_{16.16}}$ that are the complements of the 16 orthogonal codewords. If the LSB of 5 TFCI bits is 1, a biorthogonal encoder (**502** or **504**) selects the 16 biorthogonal codewords. If the LSB is 0, the biorthogonal encoder selects the 16 orthogonal codewords. Then, the biorthogonal encoder selects one of the selected orthogonal codewords or biorthogonal codewords based on the other TFCI bits and outputs the selected codeword as the first or second TFCI codeword.

A multiplexer **510** multiplexes the first and second TFCI codewords to a final 32-symbol TFCI codeword.

Upon receipt of the 32-symbol TFCI codeword, a receiver decodes the TFCI codeword separately in halves (word **1** and word **2**) and obtains 10 TFCI bits by combining the two decoded 5-bit TFCI halves. In this situation, a possible error even in one of the decoded 5-bit TFCI output during decoding leads to an error over the 10 TFCI bits.

An extended TFCI codeword also should have a powerful error correction capability. To do so, the extended TFCI codeword should have the minimum distance as suggested in reference 1.

In consideration of the number 10 of extended TFCI bits and the number 32 of the symbols of a TFCI codeword, reference 1 gives 12 as a minimum distance for an optimal code. Yet, a TFCI codeword output from the structure shown in FIG. **5**A has a minimum distance of 8 because an error in at least one of word **1** and word **2** during decoding results in an error in the whole 10 TFCI bits. That is, although extended TFCI bits are encoded separately in halves, a minimum distance between final TFCI codewords is equal to a minimum distance 8 between codeword outputs of the biorthogonal encoders **502** and **504**.

Therefore, a TFCI codeword transmitted from the encoding structure shown in FIG. **5**A is not optimal, which may increase an error probability of TFCI bits in the same radio channel environment. With the increase of the TFCI bit error probability, the receiver misjudges the data rate of received data frames and decodes the data frames with an increased error rate, thereby decreasing the efficiency of the IMT 2000 system.

According to the conventional technology, separate hardware structures are required to support the basic TFCI and the extended TFCI. As a result, constraints are imposed on implementation of an IMT 2000 system in terms of cost and system size.

SUMMARY OF THE INVENTION

It is, therefore, an object of the present invention to provide an apparatus and method for encoding an extended TFCI in an IMT 2000 system.

It is also an object of the present invention to provide an apparatus and method for encoding a basic TFCI and an extended TFCI compatibly in an IMT 2000 system.

It is another object of the present invention to provide an apparatus and method for decoding an extended TFCI in an IMT 2000 system.

It is still another object of the present invention to provide an apparatus and method for decoding a basic TFCI and an extended TFCI compatibly in an IMT 2000 system.

It is yet another object of the present invention to provide an apparatus and method for generating an optimal code by encoding an extended TFCI in an IMT 2000 system.

It is a further object of the present invention to provide a method of generating mask sequences for use in encoding/decoding an extended TFCI in an IMT 2000 system.

To achieve the above objects, there is provided a TFCI encoding/decoding apparatus and method in a CDMA mobile communication system. In the TFCI encoding apparatus, a one-bit generator generates a sequence having the same symbols. A basis orthogonal sequence generator generates a plurality of basis orthogonal sequences. A basis mask sequence generator generates a plurality of basis mask sequences. An operation unit receives TFCI bits that are divided into a 1^{st} information part representing biorthogonal sequence conversion, a 2^{nd} information part representing orthogonal sequence conversion, and a 3^{rd} information part representing mask sequence conversion and combines an orthogonal sequence selected from the basis orthogonal sequence based on the 2^{nd} information, a biorthogonal sequence obtained by combining the selected orthogonal sequence with the same symbols selected based on the 1^{st} information part, and a mask sequence selected based on the biorthogonal code sequence and the 3^{rd} information part, thereby generating a TFCI sequence.

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other objects, features and advantages of the present invention will become more apparent from the following detailed description when taken in conjunction with the accompanying drawings in which:

FIGS. **1**A to **1**D illustrate exemplary applications of a TFCI to channel frames in a general IMT 2000 system;

FIG. **2** is a block diagram of a base station transmitter in the general IMT 2000 system;

FIG. **3** is a block diagram of a mobile station transmitter in the general IMT 2000 system;

FIG. **4**A conceptionally illustrates a basic TFCI encoding structure in a conventional IMT 2000 system;

FIG. **4**B is an example of an encoding table used in a biorthogonal encoder shown in FIG. **4**A;

FIG. **5**A conceptionally illustrates an extended TFCI encoding structure in the conventional IMT 2000 system;

FIG. **5**B is an example of an algorithm of distributing TFCI bits in a controller shown in FIG. **5**A;

FIG. **5**C is an example of an encoding table used in biorthogonal encoders shown in FIG. **5**A;

FIG. **6** conceptionally illustrates a TFCI encoding structure in an IMT 2000 system according to the present invention;

FIG. **7** is a flowchart illustrating an embodiment of a mask sequence generating procedure for TFCI encoding in the IMT 2000 system according to the present invention;

FIG. **8** is a block diagram of an embodiment of a TFCI encoding apparatus in the IMT 2000 system according to the present invention;

FIG. **9** is a block diagram of an embodiment of a TFCI decoding apparatus in the IMT 2000 system according to the present invention;

FIG. **10** is a flowchart illustrating a control operation of a correlation comparator shown in FIG. **9**;

FIG. **11** is a flowchart illustrating an embodiment of a TFCI encoding procedure in the IMT 2000 system according to the present invention;

FIG. **12** is a flowchart illustrating another embodiment of the TFCI encoding procedure in the IMT 2000 system according to the present invention;

FIG. **13** illustrates an embodiment of the structures of orthogonal sequences and mask sequences determined by a TFCI according to the present invention;

FIG. **14** is a block diagram of another embodiment of the TFCI encoding apparatus in the IMT 2000 system according to the present invention;

FIG. **15** is a block diagram of another embodiment of the TFCI decoding apparatus in the IMT 2000 system according to the present invention;

FIG. **16** is a flowchart illustrating another embodiment of the TFCI encoding procedure in the IMT 2000 system according to the present invention; and

FIG. **17** is a block diagram of a third embodiment of the TFCI decoding apparatus in the IMT 2000 system according to the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Preferred embodiments of the present invention will be described herein below with reference to the accompanying drawings. In the following description, well-known functions or constructions are not described in detail since they would obscure the invention in unnecessary detail.

The present invention is directed to a TFCI encoding concept of outputting final code symbols (a TFCI codeword) by adding first code symbols (a first TFCI codeword) resulting from first TFCI bits and second code symbols (a second TFCI codeword) resulting from second TFCI bits in an IMT 2000 system. The TFCI encoding concept is shown in FIG. **6**. Here, a biorthogonal sequence and a mask sequence are given as the first TFCI codeword and the second TFCI codeword, respectively.

Referring to FIG. **6**, TFCI bits are separated into the first TFCI bits and the second TFCI bits. A mask sequence generator **602** generates a predetermined mask sequence by encoding the second TFCI bits and a biorthogonal sequence generator **604** generates a predetermined biorthogonal sequence by encoding the first TFCI bits. An adder **610** adds the mask sequence and the biorthogonal sequence and outputs final code symbols (a TFCI codeword). The mask sequence generator **602** may have an encoding table that lists mask sequences for all possible second TFCI bits. The biorthogonal sequence generator **604** may also have an encoding table that lists biorthogonal sequences for all possible first TFCI bits.

As described above, mask sequences and a mask sequence generating method should be defined to implement the present invention. Walsh codes are given as orthogonal sequences by way of example in embodiments of the present invention.

1. Mask Sequence Generating Method

The present invention pertains to encoding and decoding of TFCI bits and use of an extended Reed Muller code in an IMT 2000 system. For this purpose, predetermined sequences are used and the sequences should have a minimum distance that ensures excellent error correction performance.

A significant parameter that determines the performance or capability of a linear error correcting code is a minimum distance between codewords of the error correcting code. The Hamming weight of a codeword is the number of its symbols other than 0. If a codeword is given as "0111", its Hamming weight is 3. The smallest Hamming weight of a codeword except all "0" codeword is called a minimum weight and the minimum distance of each binary linear code is equal to the minimum weight. A linear error correcting code has a better error correcting performance as its minimum distance is increased. For details, see "The Theory of Error-Correcting Codes", F. J. Macwilliams and N. J. A. Sloane, North-Holland (hereinafter, referred to as reference 2).

An extended Reed Muller code can be derived from a set of sequences each being the sum of the elements of an m-sequence and a predetermined sequence. To use the sequence set as a linear error correcting code, the sequence set should have a large minimum distance. Such sequence sets include a Kasami sequence set, a Gold sequence set, and a Kerdock sequence set. If the total length of a sequence in such a sequence set is $L=2^{2m}$, a minimum distance$=(2^{2m}-2^{m})/2$. For $L=2^{2m+1}$, the minimum distance$=(2^{2m+1}-2^{2m})/2$. That is, if L=32, the minimum distance=12.

A description will be made of a method of generating a linear error correcting code with excellent performance, i.e., an extended error correcting code (Walsh codes and mask sequences).

According to a coding theory, there is a column transposition function for making Walsh codes from m-sequences in a group which has been formed by cyclically shifting an originating m-sequence by one to 'n' times, where the 'n' is a length of the m-sequence. In other words, each of the m-sequences is formed by cyclically shifting the originating m-sequence by a particular number of times. The column transposition function is a converting function which converts the sequences in the m-sequence group to Walsh codes. We assume there is a sequence such as a Gold sequence or a Kasami sequence which is formed by adding the originating m-sequence with another originating m-sequence. Another group of m-sequences is similarly formed by cyclically shifting the other originating m-sequence one to 'n' times, where 'n' is the length of the predetermined sequence. Afterwards, a reverse column transposition function is applied to the second group of m-sequences formed from the other originating m-sequence. The application of the reverse column transposition function to the second group of m-sequences creates another set of sequences which shall be defined as mask sequences.

In an embodiment of the present invention, a mask sequence generating method is described in connection with generation of a $(2^n, n+k)$ code (extended Reed Muller code) (here, k=1, . . . , n+1) using a Gold sequence set. The $(2^n, n+k)$ code represents output of a 2^n-symbol TFCI codeword for the input of (n+k) TFCI bits (input information bits). It is well known that a Gold sequence can be expressed as the sum of two different m-sequences. To generate the $(2^n, n+k)$ code, therefore, Gold sequences of length (2^n-1) should be produced. Here, a Gold sequence is the sum of two m-sequences $m_1(t)$ and $m_2(t)$ that are generated from generator polynomials $f1(x)$ and $f2(x)$. Given the generator polynomials $f1(x)$ and $f2(x)$, the m-sequences $m_1(t)$ and $m_2(t)$ are computed using a Trace function.

$$m_1(t) = \mathrm{Tr}(A\alpha^t) \quad t = 0, 1, \ldots, 30 \text{ and}$$
$$\mathrm{Tr}(a) = \sum_{k=0}^{n-1} a^{2k}, \quad a \in GF(2^n) \qquad \text{(Eq. 1)}$$

where A is determined by the initial value of an m-sequence, α is the root of the polynomial, and n is the order of the polynomial.

FIG. 7 is a flowchart illustrating a mask sequence generating procedure for use in generating a $(2^n, n+k)$ code from a Gold sequence set.

Referring to FIG. 7, m-sequences $m_1(t)$ and $m_2(t)$ are generated in Eq. 1 using the generator polynomials $f1(x)$ and $f2(x)$, respectively in step 710. In step 712, a sequence transposition function σ(t) is calculated to make Walsh codes from a sequence set having m-sequences formed by cyclically shifting $m_2(t)$ 0 to n−2 times where all '0' column is inserted in front of the m-sequences made from $m_2(t)$, as shown below:

$$\sigma: \{0, 1, 2, \ldots, 2^n-2\} \to \{1, 2, 3, \ldots, 2^n-1\}$$
$$\sigma(t) = \sum_{i=0}^{n-1} m_2(t+i)2^{n-1-i} \quad t = 0, 1, 2, \ldots \qquad \text{(Eq. 2)}$$

A set of 31 sequences produced by cyclically shifting the m-sequence $m_1(t)$ 0 to 30 times are column-transposed with the use of $\sigma^{-1}(t)+2$ derived from the reverse function of σ(t) in step 730. Then, 0s are added to the start of each of the resulting column-transposed sequences to make the length of the sequence 2^n. Thus, a set $d_i(t)$ of (2^n-1) sequences of length 2^n ($i=0, \ldots, 2^n-2$, $t=1, \ldots, 2^n$) are generated.

$$\{d_i(t) \mid t = 1, \ldots, 2^n, i = 0, \ldots, 2^n-2\}$$
$$d_i(t) = \begin{pmatrix} 0, & \text{if, } t = 1 \\ m_1(\sigma^{-1}(t+i)+2), & \text{if, } t = 2, 3, \ldots, 2^n \end{pmatrix} \qquad \text{(Eq. 3)}$$

A plurality of $d_i(t)$ are mask functions that can be used as 31 masks.

$d_i(t)$ is characterized in that two different masks among the above masks are added to one of (2^n-1) masks except for the two masks. To further generalize it, each of the (2^n-1) masks can be expressed as the sum of at least two of particular n masks. The n masks are called basis mask sequences. When the $(2^n, n+k)$ code is to be generated, the total number of necessary codewords is 2^{n+k} for n+k input information bits (TFCI bits). The number of 2^n orthogonal sequences (Walsh sequences) and their complements, i.e. biorthogonal sequences, is $2^n\times2=2^{n+1}$. $2^{k-1}-1(=(2^{n+k}/2^{n+1})-1)$ masks that are not 0s are needed for generation of the $(2^n, n+k)$ code.

Here, the $2^{k-1}-1$ masks can be expressed by the use of k−1 basis mask sequences, as stated before.

Now, a description will be given of a method of selecting the k−1 basis mask sequences. The m-sequence $m_1(t)$ is cyclically shifted 0 to 2^{n-1} times to generate a set of sequences in step **730** of FIG. **7**. Here, an m-sequence obtained by cyclically shifting the m-sequence $m_1(t)$ i times is expressed as $Tr(\alpha^i \cdot \alpha^t)$ according to Eq. 1. That is, a set of sequences are generated by cyclically shifting the m-sequence $m_1(t)$ 0 to 30 times with respect to an initial sequence $A=\{1, \alpha, \ldots, \alpha^{2n-2}\}$. Here, linearly independent k−1 basis elements are found from the Galois elements $1, \alpha, \ldots, \alpha^{2^n-2}$ and mask sequences corresponding to the output sequences of a Trace function with the k−1 basis elements as an initial sequence become basis mask sequences. A linear independence condition is expressed as

$\alpha, \ldots, \alpha_{k-1}$: linearly independent

$$\Leftrightarrow c_1\alpha_1+c_2\alpha_2+\ldots+c_{k-1}\alpha_{k-1}\neq 0, \forall c_1, c_2, \ldots, c_{k-1} \quad \text{(Eq. 4)}$$

To describe the above generalized mask function generation method in detail, how to generate a (32, 10) code using a Gold sequence set will be described referring to FIG. **7**. It is well known that a Gold sequence is expressed as the sum of different predetermined m-sequences. Therefore, a Gold sequence of length **31** should be generated first in order to generate the intended (32, 10) code. The Gold sequence is the sum of two m-sequences generated respectively from polynomials x^5+x^2+1 and x^5+x^4+x+1. Given a corresponding generator polynomial, each of the m-sequences $m_1(t)$ and $m_2(t)$ is computed using a Trace function by

$$m_1(t) = \mathrm{Tr}(A\alpha^t) \quad t = 0, 1, \ldots, 30 \text{ and}$$

$$\mathrm{Tr}(a) = \sum_{n=0}^{n-1} a^{2n}, \quad a \in GF(2^5) \quad \text{(Eq. 5)}$$

where A is determined by the initial value of the m-sequence, α is the root of the polynomial, and n is the order of the polynomial, here 5.

FIG. **7** illustrates the mask function generating procedure to generate the (32, 10) code.

Referring to FIG. **7**, m-sequences $m_1(t)$ and $m_2(t)$ are generated in Eq. 1 using the generator polynomials f**1**(*x*) and f**2**(*x*), respectively in step **710**. In step **712**, the column transposition function σ(t) is calculated to make a Walsh code of the m-sequence $m_2(t)$ by

$$\sigma: \{0, 1, 2, \ldots, 30\} \to \{1, 2, 3, \ldots, 31\}$$

$$\sigma(t) = \sum_{i=0}^{4} m_2(t-i)2^{4-i} \quad \text{(Eq. 6)}$$

Then, a set of 31 sequences produced by cyclically shifting the m-sequence $m_1(t)$ 0 to 30 times are column-transposed with the use of $\sigma^{-1}(t)+2$ derived from the reverse function of σ(t) in step **730**. Then, 0s are added to the start of each of the resulting sequence-transposed sequences to make the length of the sequence 31. Thus, 31 $d_i(t)$ of length **32** are generated. Here, if i=0, . . . , 31, t=1, . . . , 32. The sequences set generated in step **730** can be expressed as

$$\{d_i(t) \mid t = 1, \ldots, 32, i = 0, \ldots, 30\}$$

$$d_i(t) = \begin{pmatrix} 0, & \text{if, } t = 1 \\ m_1(\sigma^{-1}(t+i)+2), & \text{if, } t = 2, 3, \ldots, 32 \end{pmatrix} \quad \text{(Eq. 7)}$$

A plurality of $d_i(t)$ obtained from Eq. 7 can be used as 31 mask sequences.

$d_i(t)$ is characterized in that two different masks among the above masks are added to one of the 31 masks except for the two masks. In other words, each of the 31 masks can be expressed as a sum of 5 particular masks. These 5 masks are basis mask sequences.

When the (32, 10) code is to be generated, the total number of necessary codewords is $2^n=1024$ for all possible 10 input information bits (TFCI bits). The number of biorthogonal sequences of length **32** is 32×2=64. 15 masks are needed to generate the (32, 10) code. The 15 masks can be expressed as combinations of 4 basis mask sequences.

Now, a description will be given of a method of selecting the 4 basis mask sequences. An m-sequence obtained by cyclically shifting the m-sequence $m_1(t)$ i times is expressed as $Tr(\alpha^i \cdot \alpha^t)$ according to Eq. 1. That is, a set of sequences are generated by cyclically shifting the m-sequence $m_1(t)$ 0 to 30 times with respect to an initial sequence $A=\{1, \alpha, \ldots, \alpha^{2n-2}\}$. Here, 4 linearly independent basis elements are found from the Galois elements $1, \alpha, \ldots, \alpha^{2n-2}$ and mask sequences corresponding to the output sequences of a Trace function with the 4 basis elements as an initial sequence becoming basis mask sequences. A linear independence condition, is expressed as

α, β, γ, δ: linearly independent

$$\Leftrightarrow c_1\alpha+c_2\beta+c_3\gamma, +c_4\delta\neq 0, \forall c_1, c_2, c_3, c_4 \quad \text{(Eq. 8)}$$

In fact, 1, α, α^2, α^3 in the Galois $GF(2^5)$ are polynomial sub-bases that are well known as four linearly independent elements. By replacing the variable A in Eq. 1 with the polynomial bases, four basis mask sequences M1, M2, M4, and M8 are achieved.

M1=00101000011000111111000001110111
M2=00000001110011010110110111000111
M4=00001010111110010001101100101011
M8=00011100001101110010111101010001

There will herein below be given a description of an apparatus and method for encoding/decoding a TFCI using basis mask sequences as obtained in the above manner in an IMT 2000 system according to embodiments of the present invention.

2. First Embodiment of Encoding/Decoding Apparatus and Method

FIGS. **8** and **9** are block diagrams of TFCI encoding and decoding apparatuses in an IMT 2000 system according to an embodiment of the present invention.

Referring to FIG. **8**, **10** TFCI bits a**0** to a**9** are applied to corresponding multipliers **840** to **849**. A one-bit generator **800** continuously generates a predetermined code bit. That is, since the present invention deals with biorthogonal sequences, necessary bits are generated to make a biorthogonal sequence out of an orthogonal sequence. For example, the one-bit generator **800** generates bits having is to inverse an orthogonal sequence (i.e., a Walsh code) generated from a basis Walsh code generator **810** and thus generate a biorthogonal sequence. The basis Walsh code generator **810** generates basis Walsh codes of a predetermined length. The basis

Walsh codes refer to Walsh codes from which all intended Walsh codes can be produced through arbitrary addition. For example, when Walsh codes of length **32** are used, the basis Walsh codes are 1^{st}, 2^{nd}, 4^{th}, 8^{th}, and 16^{th} Walsh codes W1, W2, W4, W8, and W16, wherein:

W1: 01010101010101010101010101010101
W2: 00110011001100110011001100110011
W4: 00001111000011110000111100001111
W8: 00000000111111110000000011111111
W16: 00000000000000001111111111111111.

A basis mask sequence generator **820** generates a basis mask sequence of a predetermined length. A basis mask sequence generating method has already been described before and its details will not be described. If a mask sequence of length **32** is used, basis mask sequences are 1^{st}, 2^{nd}, 4^{th}, and 8^{th} mask sequences M1, M2, M4, M8, wherein:

M1: 00101000011000111111000001110111
M2: 00000001110011010110110111000111
M4: 00001010111110010001101100101011
M8: 00011100001101110010111101010001.

The multiplier **840** multiplies 1s output from the one-bit generator **800** by the input information bit a**0** on a symbol basis.

The multiplier **841** multiplies the basis Walsh code W1 received from the basis Walsh code generator **810** by the input information bit a**1**. The multiplier **842** multiplies the basis Walsh code W2 received from the basis Walsh code generator **810** by the input information bit a**2**. The multiplier **843** multiplies the basis Walsh code W4 received from the basis Walsh code generator **810** by the input information bit a**3**. The multiplier **844** multiplies the basis Walsh code W8 received from the basis Walsh code generator **810** by the input information bit a**4**. The multiplier **845** multiplies the basis Walsh code W16 received from the basis Walsh code generator **810** by the input information bit a**5**. The multipliers **841** to **845** multiply the received basis Walsh codes W1, W2, W4, W8, and W16 by their corresponding input information bits symbol by symbol.

Meanwhile, the multiplier **846** multiplies the basis mask sequence M1 by the input information bit a**6**. The multiplier **847** multiplies the basis mask sequence M2 by the input information bit a**7**. The multiplier **848** multiplies the basis mask sequence M4 by the input information bit a**8**. The multiplier **849** multiplies the basis mask sequence M8 by the input information bit a**9**. The multipliers **846** to **849** multiply the received basis mask sequences M1, M2, M4, and M8 by their corresponding input information bits symbol by symbol.

An adder **860** adds the encoded input information bits received from the multipliers **840** to **849** and outputs final code symbols of length 32 bits (a TFCI codeword). The length of the final code symbols (TFCI codeword) is determined by the lengths of the basis Walsh codes generated from the basis Walsh code generator **810** and the basis mask sequences generated from the basis mask sequence generator **820**.

For example, if the input information bits a**0** to a**9** are "0111011000", the multiplier **840** multiplies 0 as a**0** by is received from the one-bit generator **800** and generates 32 code symbols being all "0s". The multiplier **841** multiplies 1 as a**1** by W1 received from the basis Walsh code generator **810** and generates code symbols "01010101010101010101010101010101". The multiplier **842** multiplies 1 as a**2** by W2 received from the basis Walsh code generator **810** and generates code symbols "00110011001100110011001100110011". The multiplier **843** multiplies 1 as a**3** by W4 received from the basis Walsh code generator **810** and generates code symbols "00001111000011110000111100001111". The multiplier **844** multiplies 0 as a**4** by W8 received from the basis Walsh code generator **810** and generates 32 code symbols being all "0s". The multiplier **845** multiplies 1 as a**5** by W16 received from the basis Walsh code generator **810** and generates "00000000000000001111111111111111". The multiplier **846** multiplies 1 as a**6** by M1 received from the basis mask sequence generator **820** and generates "00101000011000111111000001110111". The multiplier **847** multiplies 0 as a**7** by M2 received from the basis mask sequence generator **820** and generates 32 code symbols being all 0s. The multiplier **848** multiplies 0 as a**8** by M4 received from the basis mask sequence generator **820** and generates 32 code symbols being all 0s. The multiplier **849** multiplies 0 as a**9** by M8 received from the basis mask sequence generator **820** and generates 32 code symbols being all 0s. The adder **860** adds the code symbols received from the multipliers **840** to **849** and outputs final code symbols "01000001000010100110011011100001". The final code symbols can be achieved by adding the basis Walsh codes W1, W2, W4 and W16 corresponding to the information bits 1s to the basis mask sequence M1 symbol by symbol. In other words, the basis Walsh codes W1, W2, W4 and W16 are summed to W23 and the Walsh code W23 and the basis mask sequence M1 are added to form the TFCI codeword (final code symbols) (=W23+M1) which is outputted from the adder **860**.

FIG. **11** is a flowchart illustrating an embodiment of a TFCI encoding procedure in an IMT 2000 system according to the present invention.

Referring to FIG. **11**, 10 input information bits (i.e., TFCI bits) are received and variables sum and j are set to an initial value 0 in step **1100**. The variable sum indicates final code symbols, and j indicates the count number of final code symbols output after symbol-basis addition. In step **1110**, it is determined whether j is 32 in view of the length **32** symbols of Walsh codes and mask sequences used for encoding the input information bits. Step **1110** is performed in order to check whether the input information bits are all encoded with the Walsh codes and the mask sequences symbol by symbol.

If j is not 32 in step **1110**, which implies that the input information bits are not encoded completely with respect to all symbols of the Walsh codes, the mask sequences, j^{th} symbols W1(j), W2(j), W4(j), W8(j), and W16(j) of the basis Walsh codes W1, W2, W4, W8, and W16 and j^{th} symbols M1(j), M2(j), M4(j), and M8(j) of the basis mask sequences M1, M2, M4, and M8 are received in step **1120**. Then, the received symbols are multiplied by the input information bits on a symbol basis and the symbol products are summed in step **1130**. The sum becomes the variable sum.

Step **1130** can be expressed as

$$\text{sum}=a0+a1\cdot W1(j)+a2\cdot W2(j)+a3\cdot W4(j)+a4\cdot W8(j)+a5\cdot W16(j)+a6\cdot M1(j)+a7\cdot M2(j)+a8\cdot M4(j)+a9\cdot M8(j) \quad \text{(Eq. 9)}$$

As noted from Eq. 9, the input information bits are multiplied by corresponding symbols of the basis Walsh codes and basis mask sequences, symbol products are summed, and the sum becomes an intended code symbol.

In step **1140**, sum indicating the achieved j^{th} code symbol, is output. j is increased by 1 in step **1150** and then the procedure returns to step **1110**. Meanwhile, if j is 32 in step **1110**, the encoding procedure ends.

The encoding apparatus of FIG. **8** according to the embodiment of the present invention can support extended TFCIs as well as basic TFCIs. Encoders for supporting an extended TFCI include a (32, 10) encoder, a (32, 9) encoder, and a (32, 7) encoder.

For the input of 10 input information bits, the (32, 10) encoder outputs a combination of 32 Walsh codes of length **32**, 32 bi-orthogonal codes inverted from the Walsh codes, and 15 mask sequences. The 32 Walsh codes can be generated from combinations of 5 basis Walsh codes. The 32 bi-orthogonal codes can be obtained by adding 1 to the 32 symbols of each Walsh code. This results has the same effect as multiplication of −1 by the 32 Walsh codes viewed as real numbers. The 15 mask sequences can be achieved through combinations of 5 basis mask sequences. Therefore, a total of 1024 codewords can be produced from the (32, 10) encoder.

The (32, 9) encoder receives 9 input information bits and outputs a combination of 32 Walsh codes of length **32**, 32 bi-orthogonal codes inverted from the Walsh codes, and 4 mask sequences. The 4 mask sequences are obtained by combing two of 4 basis mask sequences.

The (32, 7) encoder receives 7 input information bits and outputs a combination of 32 Walsh codes of length among the 1024 codewords, 32 bi-orthogonal codes inverted from the Walsh codes, and one of 4 basis mask sequences.

The above encoders for providing extended TFCIs have a minimum distance 12 and can be implemented by blocking input and output of at least of the 4 basis mask sequences generated from the basis mask sequences **820**.

That is, the (32, 9) encoder can be implemented by blocking input and output of one of the four basis mask sequences generated from the basis mask sequence generator **820** shown in FIG. **8**. The (32, 8) encoder can be implemented by blocking input and output of two of the basis mask sequences generated from the basis mask sequence generator **820**. The (32, 7) encoder can be implemented by blocking input and output of three of the basis mask sequences generated from the basis mask sequence generator **820**. As described above, the encoding apparatus according to the embodiment of the present invention can encode flexibly according to the number of input information bits, that is, the number of TFCI bits to be transmitted and maximizes a minimum distance that determined the performance of the encoding apparatus.

Codewords in the above encoding apparatus are sequences obtained by combining 32 Walsh codes of length **32**, 32 bi-orthogonal codes resulting from adding 1s to the Walsh codes, and 15 mask sequences of length **15**. The structure of the codewords is shown in FIG. **13**.

For better understanding of the TFC bits encoding procedure, Tables 1a to 1f list code symbols (TFCI codewords) versus 10 TFCI bits.

TABLE 1a

0000000000: 00000000000000000000000000000000	0000000001: 11111111111111111111111111111111
0000000010: 01010101010101010101010101010101	0000000011: 10101010101010101010101010101010
0000000100: 00110011001100110011001100110011	0000000101: 11001100110011001100110011001100
0000000110: 01100110011001100110011001100110	0000000111: 10011001100110011001100110011001
0000001000: 00001111000011110000111100001111	0000001001: 11110000111100001111000011110000
0000001010: 01011010010110100101101001011010	0000001011: 10100101101001011010010110100101
0000001100: 00111100001111000011110000111100	0000001101: 11000011110000111100001111000011
0000001110: 01101001011010010110100101101001	0000001111: 10010110100101101001011010010110
0000010000: 00000000111111110000000011111111	0000010001: 11111111000000001111111100000000
0000010010: 01010101101010100101010110101010	0000010011: 10101010010101011010101001010101
0000010100: 00110011110011000011001111001100	0000010101: 11001100001100111100110000110011
0000010110: 01100110100110010110011010011001	0000010111: 10011001011001101001100101100110
0000011000: 00001111111100000000111111110000	0000011001: 11110000000011111111000000001111
0000011010: 01011010101001010101101010100101	0000011011: 10100101010110101010010101011010
0000011100: 00111100110000110011110011000011	0000011101: 11000011001111001100001100111100
0000011110: 01101001100101100110100110010110	0000011111: 10010110011010011001011001101001
0000100000: 00000000000000001111111111111111	0000100001: 11111111111111110000000000000000
0000100010: 01010101010101011010101010101010	0000100011: 10101010101010100101010101010101
0000100100: 00110011001100111100110011001100	0000100101: 11001100110011000011001100110011
0000100110: 01100110011001101001100110011001	0000100111: 10011001100110010110011001100110
0000101000: 00001111000011111111000011110000	0000101001: 11110000111100000000111100001111
0000101010: 01011010010110101010010110100101	0000101011: 10100101101001010101101001011010
0000101100: 00111100001111001100001111000011	0000101101: 11000011110000110011110000111100
0000101110: 01101001011010011001011010010110	0000101111: 10010110100101100110100101101001
0000110000: 00000000111111111111111100000000	0000110001: 11111111000000000000000011111111
0000110010: 01010101101010101010101001010101	0000110011: 10101010010101010101010110101010
0000110100: 00110011110011001100110000110011	0000110101: 11001100001100110011001111001100
0000110110: 01100110100110011001100101100110	0000110111: 10011001011001100110011010011001
0000111000: 00001111111100001111000000001111	0000111001: 11110000000011110000111111110000
0000111010: 01011010101001011010010101011010	0000111011: 10100101010110100101101010100101
0000111100: 00111100110000111100001100111100	0000111101: 11000011001111000011110011000011
0000111110: 01101001100101101001011001101001	0000111111: 10010110011010010110100110010110
0001000000: 00101000011000111111000001110111	0001000001: 11010111100111000000111110001000
0001000010: 01111010011011010100101001000010	0001000011: 10000101100100101011010110111101
0001000100: 00011011010100001100001101000100	0001000101: 11100100101011110011110010111011
0001000110: 01001110000001011001011000010001	0001000111: 10110001111110100110100111101110
0001001000: 00100111011011001111111101111000	0001001001: 11011000100100110000000010000111
0001001010: 01110010001110011010101000101101	0001001011: 10001101110001100101010111010010
0001001100: 00010100010111111100110001001011	0001001101: 11101011101000000011001110110100
0001001110: 01000001000010101001100100011110	0001001111: 10111110111101010110011011100001
0001010000: 00101000100111001111000010001000	0001010001: 11010111011000110000111101110111
0001010010: 01111101110010011010010111011101	0001010011: 10000010001101100101101000100010
0001010100: 00011011101011111100001110111011	0001010101: 11100100010100000011110001000100
0001010110: 01001110111110101001011011101110	0001010111: 10110001000001010110100100010001
0001011000: 00100111100100111111111110000111	0001011001: 11011000011011000000000001111000
0001011010: 01110010110001101010101011010010	0001011011: 10001101001110010101010100101101
0001011100: 00010100101000001100110010110100	0001011101: 11101011010111110011001101001011
0001011110: 01000001111101011001100111100001	0001011111: 10111110000010100110011000011110

TABLE 1a-continued

0001100000: 00101000011000110000111110001000	0001100001: 11010111100111001111000001110111
0001100010: 01111101001101100101101011011101	0001100011: 10000010110010011010010100100010
0001100100: 00011011010100000011110010111011	0001100101: 11100100101011111100001101000100
0001100110: 01001110000001010110100111101110	0001100111: 10110001111110101001011000010001
0001101000: 00100111011011000000000010000111	0001101001: 11011000100100111111111101111000
0001101010: 01110010001110010101010111010010	0001101011: 10001101110001101010101000101101
0001101100: 00010100010111110011001110110100	0001101101: 11101011101000001100110001001011
0001101110: 01000001000010100110011011100001	0001101111: 10111110111101011001100100011110
0001110000: 00101000100111000000111101110111	0001110001: 11010111011000111111000010001000
0001110010: 01111101110010010101101000100010	0001110011: 10000010001101101010010111011101
0001110100: 00011011101011110011110001000100	0001110101: 11100100010100001100001110111011
0001110110: 01001110111110100110100100010001	0001110111: 10110001000001011001011011101110
0001111000: 00100111100100110000000001111000	0001111001: 11011000011011001111111110000111
0001111010: 01110010110001100101010100101101	0001111011: 10001101001110011010101011010010
0001111100: 00010100101000000011001101001011	0001111101: 11101011010111111100110010110100
0001111110: 01000001111101010110011000011110	0001111111: 10111110000010101001100111100001
0010000000: 00000001110011010110110111000111	0010000001: 11111110001100101001001000111000
0010000010: 01010100100110000011100010010010	0010000011: 10101011011001111100011101101101
0010000100: 00110010111111100101111011110100	0010000101: 11001101000000011010000100001011
0010000110: 01100111101010110000101110100001	0010000111: 10011000010101001111010001011110
0010001000: 00001110110000100110001011001000	0010001001: 11110001001111011001110100110111
0010001010: 01011011100101110011011110011101	0010001011: 10100100011010001100100001100010
0010001100: 00111101111100010101000111111011	0010001101: 11000010000011101010111000000100
0010001110: 01101000101001000000010010101110	0010001111: 10010111010110111111101101010001
0010010000: 00000001001100100110110100111000	0010010001: 11111110110011011001001011000111
0010010010: 01010100011001110011100001101101	0010010011: 10101011100110001100011110010010
0010010100: 00110010000000010101111000001011	0010010101: 11001101111111101010000111110100
0010010110: 01100111010101000000101101011110	0010010111: 10011000101010111111010010100001
0010011000: 00001110001111010110001000110111	0010011001: 11110001110000101001110111001000
0010011010: 01011011011010000011011101100010	0010011011: 10100100100101111100100010011101
0010011100: 00111101000011100101000100000100	0010011101: 11000010111100011010111011111011
0010011110: 01101000010110110000010001010001	0010011111: 10010111101001001111101110101110
0010100000: 00000001110011011001001000111000	0010100001: 11111110001100100110110111000111
0010100010: 01010100100110001100011101101101	0010100011: 10101011011001110011100010010010
0010100100: 00110010111111101010000100001011	0010100101: 11001101000000010101111011110100
0010100110: 01100111101010111111010001011110	0010100111: 10011000010101000000101110100001
0010101000: 00001110110000101001110100110111	0010101001: 11110001001111010110001011001000
0010101010: 01011011100101111100100001100010	0010101011: 10100100011010000011011110011101
0010101100: 00111101111100011010111000000100	0010101101: 11000010000011100101000111111011
0010101110: 01101000101001001111101101010001	0010101111: 10010111010110110000010010101110
0010110000: 00000001001100101001001011000111	0010110001: 11111110110011010110110100111000
0010110010: 01010100011001111100011110010010	0010110011: 10101011100110000011100001101101
0010110100: 00110010000000011010000111110100	0010110101: 11001101111111100101111000001011
0010110110: 01100111010101001111010010100001	0010110111: 10011000101010110000101101011110
0010111000: 00001110001111011001110111001000	0010111001: 11110001110000100110001000110111
0010111010: 01011011011010001100100010011101	0010111011: 10100100100101110011011101100010
0010111100: 00111101000011101010111011111011	0010111101: 11000010111100010101000100000100
0010111110: 01101000010110111111101110101110	0010111111: 10010111101001000000010001010001
0011000000: 00101001101011101001110110110000	0011000001: 11010110010100010110001001001111
0011000010: 01111100111110111100100011100101	0011000011: 10000011000001000011011100011010
0011000100: 00011010100111011010111010000011	0011000101: 11100101011000100101000101111100
0011000110: 01001111110010001111011110101110	0011000111: 10110000001101110000010000101001
0011001000: 00100110101000011001001010111111	

TABLE 1b

0011001001: 11011001010111100110110101000000	0011001010: 01110011111101001100011111101010
0011001011: 10001100000010110011100000010101	0011001100: 00010101100100101010000110001100
0011001101: 11101010011011010101111001110011	0011001110: 01000000110001111111010011011001
0011001111: 10111111001110000000101100100110	0011010000: 00101001010100011001110101001111
0011010001: 11010110101011100110001010110000	0011010010: 01111100000001001100100000011010
0011010011: 10000011111110110011011111100101	0011010100: 00011010011000101010111001111100
0011010101: 11100101100111010101000110000011	0011010110: 01001111001101111111101100101001
0011010111: 10110000110010000000010011010110	0011011000: 00100110010111101001001001000000
0011011001: 11011001101000010110110110111111	0011011010: 01110011000010111100011100010101
0011011011: 10001100111101000011100011101010	0011011100: 00010101011011011010000101110011
0011011101: 11101010100100100101111010001100	0011011110: 01000000001110001111010000100110
0011011111: 10111111110001110000101111011001	0011100000: 00101001101011100110001001001111
0011100001: 11010110010100011001110110110000	0011100010: 01111100111110110011011100011010
0011100011: 10000011000001001100100011100101	0011100100: 00011010100111010101000101111100
0011100101: 11100101011000101010111010000011	0011100110: 01001111110010000000010000101001
0011100111: 10110000001101111111101111010110	0011101000: 00100110101000010110110101000000
0011101001: 11011001010111101001001010111111	0011101010: 01110011111101000011100000010101
0011101011: 10001100000010111100011111101010	0011101100: 00010101100100100101111001110011
0011101101: 11101010011011011010000110001100	0011101110: 01000000110001110000101100100110
0011101111: 10111111001110001111010011011001	0011110000: 00101001010100010110001010110000

TABLE 1b-continued

0011110001: 11010110101011101001110101001111	0011110010: 01111100000001000011011111100101
0011110011: 10000011111110111100100000011010	0011110100: 00011010011000100101000110000011
0011110101: 11100101100111011010111001111100	0011110110: 01001111001101110000010011010110
0011110111: 10110000110010001111101100101001	0011111000: 00100110010111100110110110111111
0011111001: 11011001101000011001001001000000	0011111010: 01110011000010110011100011101010
0011111011: 10001100111101001100011100010101	0011111100: 00010101011011010101111010001100
0011111101: 11101010100100101010000101110011	0011111110: 01000000001110000000101111011001
0011111111: 10111111110001111111010000100110	0100000000: 00001010111110010001101100101011
0100000001: 11110101000001101110010011010100	0100000010: 01011111101011000100111001111110
0100000011: 10100000010100111011000110000001	0100000100: 00111001110010100010100000011000
0100000101: 11000110001101011101011111100111	0100000110: 01101100100111110111110101001101
0100000111: 10010011011000001000001010110010	0100001000: 00000101111101100001010000100100
0100001001: 11111010000010011110101111011011	0100001010: 01010000101000110100000101110001
0100001011: 10101111010111001011111010001110	0100001100: 00110110110001010010011100010111
0100001101: 11001001001110101101100011101000	0100001110: 01100011100100000011100100100010
0100001111: 10011100011011111000110110111101	0100010000: 00001010000001100001101111010100
0100010001: 11110101111110011110010000101011	0100010010: 01011111010100110100111010000001
0100010011: 10100000101011001011000101111110	0100010100: 00111001001101010010100011100111
0100010101: 11000110110010101101011100011000	0100010110: 01101100011000000111110110110010
0100010111: 10010011100111111000001001001101	0100011000: 00000101000010010001010011011011
0100011001: 11111010111101101110101100100100	0100011010: 01010000010111000100000110001110
0100011011: 10101111101000111011111001110001	0100011100: 00110110001110100010011111101000
0100011101: 11001001110001011101100000010111	0100011110: 01100011011011110111001010111101
0100011111: 10011100100100001000110101000010	0100100000: 00001010111110011110010011010100
0100100001: 11110101000001100001101100101011	0100100010: 01011111101011001011000110000001
0100100011: 10100000010100110100111001111110	0100100100: 00111001110010101101011111100111
0100100101: 11000110001101010010100000011000	0100100110: 01101100100111111000001010110010
0100100111: 10010011011000000111110101001101	0100101000: 00000101111101101110101111011011
0100101001: 11111010000010010001010000100100	0100101010: 01010000101000111011111010001110
0100101011: 10101111010111000100000101110001	0100101100: 00110110110001011101100011101000
0100101101: 11001001001110100010011100010111	0100101110: 01100011100100001000110110111101
0100101111: 10011100011011110111001001000010	0100110000: 00001010000001101110010000101011
0100110001: 11110101111110010001101111010100	0100110010: 01011111010100111011000101111110
0100110011: 10100000101011000100111010000001	0100110100: 00111001001101011101011100011000
0100110101: 11000110110010100010100011100111	0100110110: 01101100011000001000001001001101
0100110111: 10010011100111110111110110110010	0100111000: 00000101000010011110101100100100
0100111001: 11111010111101100001010011011011	0100111010: 01010000010111001011111001110001
0100111011: 10101111101000110100000110001110	0100111100: 00110110001110101101100000010111
0100111101: 11001001110001010010011111101000	0100111110: 01100011011011111000110101000010
0100111111: 10011100100100000111001010111101	0101000000: 00100010100110101110101101011100
0101000001: 11011101011001010001010010100011	0101000010: 01110111110011111011111000001001
0101000011: 10001000001100000100000111110110	0101000100: 00010001101010011101100001101111
0101000101: 11101110010101100010011110010000	0101000110: 01000100111111001000110100111010
0101000111: 10111011000000110111001011000101	0101001000: 00101101100101011110010001010011
0101001001: 11010010011010100001101110101100	0101001010: 01111000110000001011000100000110
0101001011: 10000111001111110100111011111001	0101001100: 00011110101001101101011101100000
0101001101: 11100001010110010010100010011111	0101001110: 01001011111100111000001000110101
0101001111: 10110100000011000111110111001010	0101010000: 00100010011001011110101110100011
0101010001: 11011101100110100001010001011100	0101010010: 01110111001100001011111011110110
0101010011: 10001000110011110100000100001001	0101010100: 00010001010101101101100010010000
0101010101: 11101110101010010010011101101111	0101010110: 01000100000000111000110111000101
0101010111: 10111011111111000111001000111010	0101011000: 00101101011010101110010010101100
0101011001: 11010010100101010001101101010011	0101011010: 01111000001111111011000111111001
0101011011: 10000111110000000100111000000110	0101011100: 00011110010110011101011110011111
0101011101: 11100001101001100010100001100000	0101011110: 01001011000011001000001011001010
0101011111: 10110100111100110111110100110101	0101100000: 00100010100110100001010010100011
0101100001: 11011101011001011110101101011100	0101100010: 01110111110011110100000111110110
0101100011: 10001000001100001011111000001001	0101100100: 00010001101010010010011110010000
0101100101: 11101110010101101101100001101111	0101100110: 01000100111111000111001011000101
0101100111: 10111011000000111000110100111010	0101101000: 00101101100101010001101110101100
0101101001: 11010010011010101110010001010011	0101101010: 01111000110000000100111011111001
0101101011: 10000111001111111011000100000110	0101101100: 00011110101001100010100010011111
0101101101: 11100001010110011101011101100000	0101101110: 01001011111100110111110111001010
0101101111: 10110100000011001000001000110101	0101110000: 00100010011001010001010001011100
0101110001: 11011101100110101110101110100011	0101110010: 01110111001100000100000100001001
0101110011: 10001000110011111011111011110110	0101110100: 00010001010101100010011101101111
0101110101: 11101110101010011101100010010000	0101110110: 01000100000000110111001000111010
0101110111: 10111011111111001000110111000101	0101111000: 00101101011010100001101101010011
0101111001: 11010010100101011110010010101100	0101111010: 01111000001111110100111000000110
0101111011: 10000111110000001011000111111001	0101111100: 00011110010110010010100001100000
0101111101: 11100001101001101101011110011111	0101111110: 01001011000011000111110100110101
0101111111: 10110100111100111000001011001010	0110000000: 00001011001101000111011011101100
0110000001: 11110100110010111000100100010011	0110000010: 01011110011000010010001110111001
0110000011: 10100001100111101101110001000110	0110000100: 00111000000001110100010111011111
0110000101: 11000111111110001011101000100000	0110000110: 01101101010100100001000010001010
0110000111: 10010010101011011110111101110101	0110001000: 00000100001110110111100111100011
0110001001: 11111011110001001000011000011100	0110001010: 01010001011011100010110010110110

TABLE 1b-continued

0110001011: 10101110100100011101001101001001	0110001100: 00110111000010000100101011010000
0110001101: 11001000111101111011010100101111	0110001110: 01100010010111010001111110000101
0110001111: 10011101101000101110000001111010	0110010000: 00001011110010110111011000010011
0110010001: 11110100001101001000100111101100	

TABLE 1c

0110010010: 01011110100111100010001101000110	0110010011: 10100001011000011101110010111001
0110010100: 00111000111110000100010100100000	0110010101: 11000111000001111011101011011111
0110010110: 01101101101011010001000001110101	0110010111: 10010010010100101110111110001010
0110011000: 00000100110001000111100100011100	0110011001: 11111011001110111000011011100011
0110011010: 01010001100100010010110001001001	0110011011: 10101110011011101101001110110110
0110011100: 00110111111101110100101000101111	0110011101: 11001000000010001011010111010000
0110011110: 01100010101000100001111101111010	0110011111: 10011101010111011110000010000101
0110100000: 00001011001101001000100100010011	0110100001: 11110100110010110111011011101100
0110100010: 01011110011000011101110001000110	0110100011: 10100001100111100010001110111001
0110100100: 00111000000001111011101000100000	0110100101: 11000111111110000100010111011111
0110100110: 01101101010100101110111101110101	0110100111: 10010010101011010001000010001010
0110101000: 00000100001110111000011000011100	0110101001: 11111011110001000111100111100011
0110101010: 01010001011011101101001101001001	0110101011: 10101110100100010010110010110110
0110101100: 00110111000010001011010100101111	0110101101: 11001000111101110100101011010000
0110101110: 01100010010111011110000001111010	0110101111: 10011101101000100001111110000101
0110110000: 00001011110010111000100111101100	0110110001: 11110100001101000111011000010011
0110110010: 01011110100111011011100101111001	0110110011: 10100001011000010010001101000110
0110110100: 00111000111110001011101011011111	0110110101: 11000111000001110100010100100000
0110110110: 01101101101011011110111110001010	0110110111: 10010010010100100001000001110101
0110111000: 00000100110001001000011011100011	0110111001: 11111011001110110111100100011100
0110111010: 01010001100100011101001110110110	0110111011: 10101110011011100010110001001001
0110111100: 00110111111101110110101111010000	0110111101: 11001000000010000100101000101111
0110111110: 01100010101000101110000010000101	0110111111: 10011101010111010001111101111010
0111000000: 00100011010101111000011010011011	0111000001: 11011100101010000111100101100100
0111000010: 01110110000000101101001111001110	0111000011: 10001001111111010010110000110001
0111000100: 00010000001100100101101011010000	0111000101: 11101111110011011010010100101111
0111000110: 01000101001100011110000011111101	0111000111: 10111010110011100001111100000010
0111001000: 00101100010110001000100110010100	0111001001: 11010011101001110111011001101011
0111001010: 01111001000011011101110011000001	0111001011: 10000110111100100010001100111110
0111001100: 00011111011010111011101010100111	0111001101: 11100000100101000100010101011000
0111001110: 01001010001111011101111111110010	0111001111: 10110101110000010001000000001101
0111010000: 00100011101010001000011001100100	0111010001: 11011100010101110111100110011011
0111010010: 01110110111111011101001100110001	0111010011: 10001001000000100010110011001110
0111010100: 00010000100110111011010101010111	0111010101: 11101111011001000100101010101000
0111010110: 01000101110011101110000000000010	0111010111: 10111010001100010001111111111101
0111011000: 00101100101001110001001011011011	0111011001: 11010011010110001110110100100100
0111011010: 01111001111100101101110000111110	0111011011: 10000110000011010010001111000001
0111011100: 00011111100101001011101001011000	0111011101: 11100000011010110100010110100111
0111011110: 01001010110000011110111100001101	0111011111: 10110101001111100001000011110010
0111100000: 00100011010101110111100101100100	0111100001: 11011100101010001000011010011011
0111100010: 01110110000000100010110000110001	0111100011: 10001001111111011101001111001110
0111100100: 00010000001100100101001010010111	0111100101: 11101111110011011010110101101000
0111100110: 01000101001100010001111100000010	0111100111: 10111010110011101110000011111101
0111101000: 00101100010110000111011001101011	0111101001: 11010011101001111000100110010100
0111101010: 01111001000011010010001100111110	0111101011: 10000110111100101101110011000001
0111101100: 00011111011010110100010101011000	0111101101: 11100000100101001011101010100111
0111101110: 01001010001111000010000000001101	0111101111: 10110101110000111101111111110010
0111110000: 00100011101010000111100110011011	0111110001: 11011100010101111000011001100100
0111110010: 01110110111111010010110011001110	0111110011: 10001001000000101101001100110001
0111110100: 00010000100110110100101010101000	0111110101: 11101111011001001011010101010111
0111110110: 01000101110011100001111111111101	0111110111: 10111010001100011110000000000010
0111111000: 00101100101001110111011010010100	0111111001: 11010011010110001000100101101011
0111111010: 01111001111100100010001111000001	0111111011: 10000110000011011101110000111110
0111111100: 00011111100101000100010110100111	0111111101: 11100000011010111011101001011000
0111111110: 01001010110000010001000011110010	0111111111: 10110101001111101110111100001101
1000000000: 00011100001101110010111101010001	1000000001: 11100011110010001101000010101110
1000000010: 01001001011000100111101000000100	1000000011: 10110110100111011000010111111011
1000000100: 00101111000001000000111000110001	1000000101: 11010000111110111111000111001110
1000000110: 01111010010100010100100100110111	1000000111: 10000101101011101011011011001000
1000001000: 00010011001110000001000000101110	1000001001: 11101100110001111110111111010001
1000001010: 01000110011011010111010100001011	1000001011: 10111001100100101000101011110100
1000001100: 00100000000010110001001101101101	1000001101: 11011111111101001110110010010010
1000001110: 01110101010111100100011000111000	1000001111: 10001010101000011011100111000111
1000010000: 00011100110010000010111110101110	1000010001: 11100011001101111101000001010001
1000010010: 01001001100111010111101011111011	1000010011: 10110110011000101000010100000100
1000010100: 00101111111110110001110010011101	1000010101: 11010000000001001110001101100010
1000010110: 01111010101011100100100111001000	1000010111: 10000101010100011011011000110111
1000011000: 00010011110001110010000010100001	1000011001: 11101100001110001101111101011110
1000011010: 01000110100100100111010111110100	1000011011: 10111001011011011000101000001011

TABLE 1c-continued

1000011100: 00100000111101000001001110010010	1000011101: 11011111000010111110110001101101
1000011110: 01110101101000010100011011000111	1000011111: 10001010010111101011100100111000
1000100000: 00011100001101111101000010101110	1000100001: 111000111100100000010111101010001
1000100010: 01001001011000101000010111111011	1000100011: 10110110100111010111101000000100
1000100100: 00101111000001001110001110011101	1000100101: 11010000111110110001110001100010
1000100110: 01111010010100011011011011001000	1000100111: 10000101101011100100100100110111
1000101000: 00010011001110001101111110100001	1000101001: 11101100110001110010000001011110
1000101010: 01000110011011011000101011110100	1000101011: 10111001100100100111010100001011
1000101100: 00100000000010111110110010010010	1000101101: 11011111111101000001001101101101
1000101110: 01110101010111101011100111000111	1000101111: 10001010101000010100011000111000
1000110000: 00011100110010001101000001010001	1000110001: 11100011001101110010111110101110
1000110010: 01001001100111011000010100000100	1000110011: 10110110011000100111101011111011
1000110100: 00101111111110111110001101100010	1000110101: 11010000000001000001110010011101
1000110110: 01111010101011101011011000110111	1000110111: 10000101010100010100100111001000
1000111000: 00010011110001111101111101011110	1000111001: 11101100001110000010000010100001
1000111010: 01000110100100101000101000001011	1000111011: 10111001011011010111010111110100
1000111100: 00100000111101001110110001101101	1000111101: 11011111000010110001001110010010
1000111110: 01110101101000011011100100111000	1000111111: 10001010010111100100011011000111
1001000000: 00110100010101001101111100100110	1001000001: 11001011101010110010000011011001
1001000010: 01100001000000011000101001110011	1001000011: 10011110111111100111010110001100
1001000100: 00000111011001111110110000010101	1001000101: 11111000100110000001001111101010
1001000110: 01010010001100101011100101000000	1001000111: 10101101110011010100011010111111
1001001000: 00111011010110111101000000101001	1001001001: 11000100101001000010111111010110
1001001010: 01101110000011101000010101111100	1001001011: 10010001111100010111101010000011
1001001100: 00001000011010001110001100011010	1001001101: 11110111100101110001110011100101
1001001110: 01011101001111011011011001001111	1001001111: 10100010110000100100100110110000
1001010000: 00110100101010111101111111011001	1001010001: 11001011010101000010000000100110
1001010010: 01100001111111101000101010001100	1001010011: 10011110000000010111010101110011
1001010100: 00000111100110001110110011101010	1001010101: 11111000011001110001001100010101
1001010110: 01010010110011011011100110111111	1001010111: 10101101001100100100011001000000
1001011000: 00111011101001001101000011010110	1001011001: 11000100010110110010111100101001
1001011010: 01101110111100011000010110000011	

TABLE 1d

1001011011: 10010001000011100111101001111100	1001011100: 00001000100101111110001111100101
1001011101: 11110111011010000001110000011010	1001011110: 01011101110000101011011010110000
1001011111: 10100010001111010100100101001111	1001100000: 0110100010101000010000011011001
1001100001: 11001011101010111101111100100110	1001100010: 1100001000000010111010110001100
1001100011: 10011110111111101000101001110011	1001100100: 00000111011001110001001111101010
1001100101: 1111000100110001110110000010101	1001100110: 01010010001100100100011010111111
1001100111: 10101101110011011011100101000000	1001101000: 00111011010110110010111111010110
1001101001: 1000100101001001101000000101001	1001101010: 01101110000011100111101010000011
1001101011: 10010001111100011000010101111100	1001101100: 0001000011010000001110011100101
1001101101: 11110111100101111110001100011010	1001101110: 01011101001111010100100110110000
1001101111: 10100010110000101011011001001111	1001110000: 0110100101010110010000000100110
1001110001: 11001011010101001101111111011001	1001110010: 01100001111111100111010101110011
1001110011: 10011110000000011000101010001100	1001110100: 0000111100110000001001100010101
1001110101: 11111000011001111110110011101010	1001110110: 1010010110011010100011001000000
1001110111: 10101101001100101011100110111111	1001111000: 00111011101001000010111100101001
1001111001: 11000100010110111101000011010110	1001111010: 01101110111100010111101001111100
1001111011: 0010001000011101000010110000011	1001111100: 00001000100101110001110000011010
1001111101: 11110111011010001110001111100101	1001111110: 01011101110000100100100101001111
1001111111: 10100010001111011011011010110000	1010000000: 00011101111110100100001010010110
1010000001: 1100010000001011011110101101001	1010000010: 01001000101011110001011111000011
1010000011: 0110111010100001110100000111100	1010000100: 0101110110010010111000110100101
1010000101: 1010001001101101000111001011010	1010000110: 01111011100111000010010011110000
1010000111: 10000100011000111101101100001111	1010001000: 0010010111101010100110110011001
1010001001: 1101101000010101011001001100110	1010001010: 1000111101000000001100011001100
1010001011: 10111000010111111110011100110011	1010001100: 00100001110001100111111010101010
1010001101: 1011110001110011000000101010101	1010001110: 01110100100100110010101111111111
1010001111: 0001011011011001101010000000000	1010010000: 0011101000001010100001001101001
1010010001: 11100010111110101011110110010110	1010010010: 1001000010100000001011100111100
1010010011: 10110111101011111110100011000011	1010010100: 0101110001101100111000101011010
1010010101: 1010001110010011000111010100101	1010010110: 01111011011000110010010000001111
1010010111: 10000100100111001101101111110000	1010011000: 0010010000010100100110101100110
1010011001: 11101101111101011011001010011001	1010011010: 01000111010111110001100000110011
1010011011: 10111000101000001110011111001100	1010011100: 00100001001110010111111001010101
1010011101: 1011110110001101000000110101010	1010011110: 01110100011011000010101100000000
1010011111: 10001011100100111101010011111111	1010100000: 00011101111110101011110101101001
1010100001: 11000100000001010100001010010110	1010100010: 01001000101011111110100000111100
1010100011: 0110111010100000001011111000011	1010100100: 0101110110010011000111001011010
1010100101: 1010001001101100111000110100101	1010100110: 01111011100111001101101100001111
1010100111: 0000100011000110010010011110000	1010101000: 0010010111101011011001001100110
1010101001: 1101101000010100100110110011001	1010101010: 01000111101000001110011100110011
1010101011: 0111000010111110001100011001100	1010101100: 0100001110001101000000101010101

TABLE 1d-continued

1010101101: 11011110001110010111111010101010	1010101110: 11101001001001111010100000000000
1010101111: 10001011011011000010101111111111	1010110000: 00011101000001011011110110010110
1010110001: 11000101111101001000010011010011	1010110010: 10010000101000011101000110000111
1010110011: 10110111101011110001011100111100	1010110100: 01011100011011010001110101001011
1010110101: 10100011100100101110001010110101	1010110110: 01111011011000111101101111110000
1010110111: 00001001001110000100100000011111	1010111000: 00100100000101010110010100110011
1010111001: 11101101111101010100110101100110	1010111010: 01000111010111111111001111001100
1010111011: 01110001010000000001100000110011	1010111100: 01000010011100110000001101010101
1010111101: 11011110110001100111111001010101	1010111110: 01110100011011001101010011111111
1010111111: 00010111001001100101011000000000	1011000000: 00110101100110011011001011100001
1011000001: 10010100110011001001101000111101	1011000010: 01100000110011001110011110110100
1011000011: 00111110011001100011000010010111	1011000100: 00001101010101010000001110100101
1011000101: 11111001010101010111111000101101	1011000110: 01010011111111111101010010000111
1011000111: 01011000000000000101011011110001	1011001000: 00111010100101101011110111101110
1011001001: 10001010110100101000010000100011	1011001010: 01101111110000111110100010111011
1011001011: 00100000011110000010111010001001	1011001100: 00001001101001011000111011011101
1011001101: 11101100101101001110001001000101	1011001110: 01011100111100001101101110001000
1011001111: 10100011000011110010010001110111	1011010000: 00110101011001101011001000011110
1011010001: 10010101001100101001101111000011	1011010010: 01100000000110011111001110100101
1011010011: 10011111110011000001100010110100	1011010100: 00001100101010110000001001011011
1011010101: 11111001101010100111111011010010	1011010110: 10100110000000011010100011110001
1011010111: 10101100111111110010101110000111	1011011000: 00111010011010011011110100010001
1011011001: 10001011001011001000010111011101	1011011010: 01101111001111001110100001000100
1011011011: 10010000110000110001011110111011	1011011100: 00010010101101010001110001000101
1011011101: 11110110101001010111000111011101	1011011110: 01011100000001111110110110111011
1011011111: 10100011111110000000100100010001000	1011100000: 00110101100110010100110100011110
1011100001: 11001010011001101011001011100001	1011100010: 01100000110011000001100001001011
1011100011: 10011111001100111110011110110100	1011100100: 00000110101010100111111000101101
1011100101: 11111001010101011000000111010010	1011100110: 01010011111111110010101101111000
1011100111: 10101100000000001101010010000111	1011101000: 00111010100101100100001000010001
1011101001: 11000101011010011011110111101110	1011101010: 01101111110000110001011101000100
1011101011: 10010000001111001110100010111011	1011101100: 00001001101001010111000100100010
1011101101: 11110110010110101000111011011101	1011101110: 01011100111100000001001000111011
1011101111: 10100011000011111101101110001000	1011110000: 00110101011001100100110111100001
1011110001: 11001010100110011011001000011110	1011110010: 01100000000110011000110001011010
1011110011: 10011111110011001110011101001011	1011110100: 00000110010101010111111011010010
1011110101: 11111001101010101000000100101101	1011110110: 01010011000000000010101110000111
1011110111: 10101100111111111101010001111000	1011111000: 00111010011010010100001011101110
1011111001: 11000101100101101011110100010001	1011111010: 01101111001111000001011110111011
1011111011: 10010000110000111110100001000100	1011111100: 00001001010110100111000111011101
1011111101: 11110110101001011000111000100010	1011111110: 01011100000001111001001001000100
1011111111: 10100011111100001101101101110111	1100000000: 00010110110011100011010001111010
1100000001: 11101001001100011100101110000101	1100000010: 01000011100110110110000100101111
1100000011: 10111100011001001001111011010000	1100000100: 00100101111111010000011101001001
1100000101: 11011010000000101111100010110110	1100000110: 01110000101010000101001000011100
1100000111: 10001111010101111010110111100011	1100001000: 00011001110000010011101101110101
1100001001: 11100110001111101100010010001010	1100001010: 01001100100101000110111000100000
1100001011: 10110011011010111001000111011111	1100001100: 00101010111100100000100001000110
1100001101: 11010101000011011111011110111001	1100001110: 01111111101001110101110100010011
1100001111: 10000000010110001010001011101100	1100010000: 00010110001100010011010010000101
1100010001: 11101001110011101100101101111010	1100010010: 01000011011001000110000111010000
1100010011: 10111100100110111001111000101111	1100010100: 01001010000000100000011110110110
1100010101: 11011010111111011111100001001001	1100010110: 01110000010101110101001011100011
1100010111: 10001111101010001010110100011100	1100011000: 00011001001111100011101110001010
1100011001: 11100110110000011100010001110101	1100011010: 01001100011010110110111011011111
1100011011: 10110011100101001001000100100000	1100011100: 01010100000110100001000101111001
1100011101: 11010101111100101111011101000110	1100011110: 01111111010110000101110111101100
1100011111: 10000000101001111010001000010011	1100100000: 00010110110011101100101110000101
1100100001: 11101001001100010011010001111010	1100100010: 01000011100110111001111011010000
1100100011: 10111100011001000110000100101111	

TABLE 1e

1100100100: 00100101111111011111100010110110	1100100101: 11011010000000100000011101001001
1100100110: 01110000101010001010110111100011	1100100111: 10001111010101110101001000011100
1100101000: 00011001110000011100010010001010	1100101001: 11100110001111100011101101110101
1100101010: 01001100100101001001000111011111	1100101011: 10110011011010110110111000100000
1100101100: 00101010111100101111011110111001	1100101101: 11010101000011010000100001000110
1100101110: 01111111101001110100010111011100	1100101111: 10000000010110000101110100010011
1100110000: 00101100011000111001101101111010	1100110001: 11101001110011100011010010000101
1100110010: 01000011011001001001111000101111	1100110011: 10111100100110110110000111010000
1100110100: 00100101000000101111100001001001	1100110101: 11011010111111010000011110110110
1100110110: 01110000001010111101011010001111 00	1100110111: 10001111101010000101001011100011
1100111000: 00011001001111011000100011101011	1100111001: 11100110110000010011101110001010
1100111010: 01001100011010111001000100100000	1100111011: 10110011100101000110111011011111
1100111100: 00101010000011011111011101000110	1100111101: 11010101111100100000100010111001

TABLE 1e-continued

1100111110: 01111111010110001010001000010011	1100111111: 10000000101001110101110111101100
1101000000: 00111110101011011100010000001101	1101000001: 11000001010100100011101111110010
1101000010: 01101011111110001001000101011000	1101000011: 10010100000001110110111010100111
1101000100: 00001101100111101111011100111110	1101000101: 11110010011000010000100011000001
1101000110: 01011000110010111010001001101011	1101000111: 10100111001101000101110110010100
1101001000: 00110001101000101100101100000010	1101001001: 11001110010111010011010011111101
1101001010: 01100100111101111001111001010111	1101001011: 10011011000010000110000110101000
1101001100: 00000010100100011111100000110001	1101001101: 11111101011011100000011111001110
1101001110: 01010111110001001010110101100100	1101001111: 10101000001110110101001010011011
1101010000: 00111110010100101100010011110010	1101010001: 11000001101011010011101100001101
1101010010: 01101011000001111001000110100111	1101010011: 10010100111110000110111001011000
1101010100: 00001101011000011111011111000001	1101010101: 11110010100111100000100000111110
1101010110: 01011000001101001010001010010100	1101010111: 10100111110010110101110101101011
1101011000: 00110001010111011100101111111101	1101011001: 11001110101000100011010000000010
1101011010: 01100100000010001001111010101000	1101011011: 10011011111101110110000101010111
1101011100: 00000010011011101111100011001110	1101011101: 11111101100100010000011100110001
1101011110: 01010111001110111010110110011011	1101011111: 10101000110001000101001001100100
1101100000: 00111110101011010011101111110010	1101100001: 11000001010100101100010000001101
1101100010: 01101011111110000110111010100111	1101100011: 10010100000001111001000101011000
1101100100: 00001101100111100000100011000001	1101100101: 11110010011000011111011100111110
1101100110: 01011000110010110101110110010100	1101100111: 10100111001101001010001001101011
1101101000: 00110001101000100011010011111101	1101101001: 11001110010111011100101100000010
1101101010: 01100100111101110110000110101000	1101101011: 10011011000010001001111001010111
1101101100: 00000010100100010000011111001110	1101101101: 111111010110111011111000000110001
1101101110: 01010111110001000101001010011011	1101101111: 10101000001110111010110101100100
1101110000: 00111110010100100011101100001101	1101110001: 11000001101011011100010011110010
1101110010: 01101011000001110110111001011000	1101110011: 10010100111110001001000110100111
1101110100: 00001101011000010000100000111110	1101110101: 11110010100111101111011111000001
1101110110: 01011000001101000101110101101011	1101110111: 10100111110010111010001010010100
1101111000: 00110001010111010011010000000010	1101111001: 11001110101000101100101111111101
1101111010: 01100100000010000110000101010111	1101111011: 10011011111101111001111010101000
1101111100: 00000010011011100000011100110001	1101111101: 11111101100100011111100011001110
1101111110: 01010111001110110101001001100100	1101111111: 10101000110001001010110110011011
1110000000: 00010111000000110101100110111101	1110000001: 11101000111111001010011001000010
1110000010: 01000010010101100000110011101000	1110000011: 10111101101010011111001100010111
1110000100: 00100100001100000110101010001110	1110000101: 11011011110011111001010101110001
1110000110: 01110001011001010011111111011011	1110000111: 10001110100110101100000000100100
1110001000: 00011000000011000101011010110010	1110001001: 11100111111100111010100101001101
1110001010: 01001101010110010000001111100111	1110001011: 10110010101001101111110000011000
1110001100: 00101011001111110110010110000001	1110001101: 1101010011000000100110100111110
1110001110: 01111110011010100011000011010100	1110001111: 10000001100101011100111100101011
1110010000: 0001011111111000101100101000010	1110010001: 1110100000000111010011010111101
1110010010: 01000010101010010000110000010111	1110010011: 10111101010101101111001111101000
1110010100: 00100100110011110110101001110001	1110010101: 11011011001100001001010110001110
1110010110: 01110001100110100011111100100100	1110010111: 10001110011001011100000011011011
1110011000: 00011000111100110101011001001101	1110011001: 11100111000011001010100110110010
1110011010: 01001101101001100000001100011000	1110011011: 10110010010110011111110011100111
1110011100: 00101011100000001100101011111110	1110011101: 11010100011111110011010100000001
1110011110: 01111110100101010011000000101011	1110011111: 10000001011010101100111111010100
1110100000: 00010111000000111010011001000010	1110100001: 11101000111111000101100110111101
11101000010: 10000100101011011110011000010111	1110100011: 10111101101010010000110011101000
1110100100: 00100100001100001001010101110001	1110100101: 11011011110011110110101010001110
1110100110: 01110001011001011100000000100100	1110100111: 10001110100110100011111111011011
1110101000: 00011000000011001010100101001101	1110101001: 11100111111100110101011010110010
1110101010: 01001101010110011111110000011000	1110101011: 10110010101001100000001111100111
1110101100: 00101011001111111001101001111110	1110101101: 11010100110000000110010110000001
1110101110: 01111110011010101100111100101011	1110101111: 10000001100101010011000011010100
1110110000: 00010111111111001010011010111101	1110110001: 11101000000000110101100101000010
1110110010: 01000010101010011110011111101000	1110110011: 10111101010101100000110000010111
1110110100: 00100100110011111001010110001110	1110110101: 11011011001100000110101001110001
1110110110: 01110001100110101100000011011011	1110110111: 10001110011001010011111100100100
1110111000: 00011000111100111010100110110010	1110111001: 11100111000011000101011001001101
1110111010: 01001101101001101111110011100111	1110111011: 10110010010110010000001100011000
1110111100: 00101011100000001001101010000001	1110111101: 11010100011111110110010101111110
1110111110: 01111110100101011100111111010100	1110111111: 10000001011010100011000000101011
1111000000: 00111111011000001010100111001010	1111000001: 11000000100111110101011000110101
1111000010: 01101010001101011111110010011111	1111000011: 1001010111001010000000110110000
1111000100: 00001100010100111001101011111001	1111000101: 11110011101011000110010100000110
1111000110: 01011001000001101100111110101100	1111000111: 10100110111110010011000001010011
1111001000: 00110000011011111010011011000101	1111001001: 11001111100100000101100100111010
1111001010: 01100101001110101111001110010000	1111001011: 10011010110001010000110001101111
1111001100: 00000011010111001001010111110110	1111001101: 11111100101000110110101000001001
1111001110: 01010110000010011100000010100011	1111001111: 101010011111011000011111101011100
1111010000: 00111111100111111010100100110101	1111010001: 11000000011000000101011011001010
1111010010: 01101010110010101111110001100000	1111010011: 10010101001101010000001110011111
1111010100: 00001100101011001001101000000110	1111010101: 11110011010100110110010111111001
1111010110: 01011001111110011100111101010011	1111010111: 10100110000001100011000010101100
1111011000: 00110000100100001010011000111010	1111011001: 11001111011011110101100111000101
1111011010: 01100101110001011111001101101111	1111011011: 10011010001110100000110010010000

TABLE 1e-continued

1111011100: 00000011101000111001010100001001	1111011101: 11111100010111000110101011110110
1111011110: 01010110111101101100000000011100	1111011111: 10101001000010010011111110100011
1111100000: 00111111011000000101011000110101	1111100001: 11000000100111111010100111001010
1111100010: 01101010001101010000001101100000	1111100011: 10010101110010101111110010011111
1111100100: 00001100010100110110010100000110	1111100101: 11110011101011001001101011111001
1111100110: 01011001000001100011000001010011	1111100111: 10100110111110011100111110101100
1111101000: 00110000011011110101100100111010	1111101001: 11001111100100001010011011000101
1111101010: 01100101001110100000110001101111	1111101011: 10011010110001011111001110010000
1111101100: 00000011010111000110101000001001	

TABLE 1f

1111101101: 11111100101000111001010111110110
1111101110: 01010110000010010011111101011100
1111101111: 10101001111101101100000010100011
1111110000: 00111111100111110101011011001010
1111110001: 11000000011000001010100100110101
1111110010: 01101010110010100000001110011111
1111110011: 10010101001101011111110001100000
1111110100: 00001100101011000110010111111001
1111110101: 11110011010100111001101000000110
1111110110: 01011001111110010011000010101100
1111110111: 10100110000001101100111101010011
1111111000: 00110000100100000101100111000101
1111111001: 11001111011011111010011000111010
1111111010: 01100101110001010000110010010000
1111111011: 10011010001110101111001101101111
1111111100: 00000011101000110110101011110110
1111111101: 11111100010111001001010100001001
1111111110: 01010110111101100011111110100011
1111111111: 10101001000010011100000001011100

The decoding apparatus according to the embodiment of the present invention will be described referring to FIG. **9**. An input signal r(t) is applied to 15 multipliers **902** to **906** and a correlation calculator **920**. The input signal r(t) was encoded with a predetermined Walsh code and a predetermined mask sequence in a transmitter. A mask sequence generator **910** generates all possible 15 mask sequences M1 to M15. The multipliers **902** to **906** multiply the mask sequences received from the mask sequence generator **910** by the input signal r(t). The multiplier **902** multiplies the input signal r(t) by the mask sequence M1 received from the mask sequence generator **910**. The multiplier **904** multiplies the input signal r(t) by the mask sequence M2 received from the mask sequence generator **910**. The multiplier **906** multiplies the input signal r(t) by the mask sequence M15 received from the mask sequence generator **910**. If the transmitter encoded TFCI bits with the predetermined mask sequence, one of the outputs of the multipliers **902** to **906** is free of the mask sequence, which means the mask sequence has no effect on the correlations calculated by one of the correlation calculators. For example, if the transmitter used the mask sequence M2 for encoding the TFCI bits, the output of the multiplier **904** that multiplies the mask sequence M2 by the input signal r(t) is free of the mask sequence. The mask sequence-free signal is TFCI bits encoded with the predetermined Walsh code. Correlation calculators **920** to **926** calculate the correlations of the input signal r(t) and the outputs of the multipliers **902** to **906** to 64 bi-orthogonal codes. The 64 bi-orthogonal codes have been defined before. The correlation calculator **920** calculates the correlation values of the input signal r(t) to the 64 bi-orthogonal codes of length **32**, selects the maximum correlation value from the 64 correlations, and outputs the selected correlation value, a bi-orthogonal code index corresponding to the selected correlation value, and its unique index "0000" to a correlation comparator **940**.

The correlation calculator **922** calculates the correlation values of the output of the multiplier **902** to the 64 bi-orthogonal codes, selects the maximum value of the 64 correlations, and outputs the selected correlation value, a bi-orthogonal code index corresponding to the selected correlation, and its unique index "0001" to the correlation comparator **940**. The correlation calculator **924** calculates the correlation values of the output of the multiplier **904** to the 64 bi-orthogonal codes, selects the maximum of the 64 correlation values, and outputs the selected correlation value, a bi-orthogonal code index corresponding to the selected correlation value, and its unique index "0010" to the correlation comparator **940**. Other correlation calculators (not shown) calculate the correlation values of the outputs of the correspondent multipliers to the 64 bi-orthogonal codes and operate similar to the above described correlation calculators, respectively.

Finally, the correlation calculator **926** calculates the correlation values of the output of the multiplier **906** to the 64 bi-orthogonal codes, selects the maximum value of the 64 correlations, and outputs the selected correlation value, a bi-orthogonal code index corresponding to the selected correlation value, and its unique index "1111" to the correlation comparator **940**.

The unique indexes of the correlation calculators **920** to **926** are the same as the indexes of the mask sequences multiplied by the input signal r(t) in the multipliers **902** to **906**. Table 2 lists the 15 mask indexes multiplied in the multipliers and a mask index assigned to the case that no mask sequence is used, by way of example.

TABLE 2

mask sequence	mask sequence index
not used	0000
M1	0001
M2	0010
M3	0011
M4	0101
M5	0101
M6	0110
M7	0111
M8	1000
M9	1001
M10	1010
M11	1011
M12	1100
M13	1101
M14	1110
M15	1111

As shown in Table 2, the correlation calculator **922**, which receives the signal which is the product of the input signal r(t) and the mask sequence M1, outputs "0001" as its index. The correlation calculator **926**, which receives the signal which is the product of the input signal r(t) and the mask sequence

M15, outputs "1111" as its index. The correlation calculator **920**, which receives only the input signal r(t), outputs "0000" as its index.

Meanwhile, the bi-orthogonal code indexes are expressed in a binary code. For example, if the correlation to $\overline{W4}$ which is the complement of W4 is the largest correlation value, a corresponding bi-orthogonal code index (a**0** to a**9**) is "001001".

The correlation comparator **940** compares the 16 maximum correlation values received from the correlation calculators **920** to **926**, selects the highest correlation value from the 16 received maximum correlation values, and outputs TFCI bits based on the bi-orthogonal code index and the mask sequence index (the unique index) received from the correlation calculator that corresponds to the highest correlation value. The TFCI bits can be determined by combining the bi-orthogonal code index and the mask sequence index. For example, if the mask sequence index is that of M4(0100) and the bi-orthogonal code index is that of $\overline{W4}$(001001), the TFCI bits (a**9** to a**0**) are "the M4 index (0100)+the $\overline{W4}$ index (001001)". That is, the TFCI bits (a**9** to a**0**) are "0100001001"

Assuming that the transmitter transmitted code symbols corresponding to TFCI bits (a**0** to a**9**) "1011000010", it can be said that the transmitter encoded the TFCI bits with $\overline{W6}$ and M4 according to the afore-described encoding procedure. The receiver can determine that the input signal r(t) is encoded with the mask sequence M4 by multiplying the input signal r(t) by all the mask sequences and that the input signal r(t) is encoded with $\overline{W6}$ by calculating the correlations of the input signal r(t) to all the bi-orthogonal codes. Based on the above example, the fifth correlation calculator (not shown) will output the largest correlation value, the index of $\overline{W6}$ (101100) and its unique index (0010). Then, the receiver outputs the decoded TFCI bits (a**0** to a**9**) "1011000010" by adding the index of $\overline{W6}$ "101100" and the M4 index "0010".

In the embodiment of the decoding apparatus, the input signal r(t) is processed in parallel according to the number of mask sequences. It can be further contemplated that the input signal r(t) is sequentially multiplied by the mask sequences and the correlations of the products are sequentially calculated in another embodiment of the decoding apparatus.

FIG. **17** illustrates another embodiment of the decoding apparatus.

Referring to FIG. **17**, a memory **1720** stores an input 32-symbol signal r(t). A mask sequence generator **1710** generates 16 mask sequences that were used in the transmitter and outputs them sequentially. A multiplier **1730** multiplies one of the 16 mask sequences received from the mask sequence generator **1710** by the input signal r(t) received from the memory **1720**. A correlation calculator **1740** calculates the output of the multiplier **1730** to 64 biorthogonal codes bi-orthogonal of length **32** and outputs the maximum correlation value and the index of a biorthogonal code corresponding to the largest correlation value to a correlation comparator **1750**. The correlation comparator **1750** stores the maximum correlation value and the biorthogonal code index received from the correlation calculator **1740**, and the index of the mask sequence received from the mask sequence generator **1710**.

Upon completion of above processing with the mask sequence, the memory **1720** outputs the stored input signal r(t) to the multiplier **1730**. The multiplier **1730** multiplies the input signal r(t) by one of the other mask sequences. The correlation calculator **1740** calculates correlation of the output of the multiplier **1730** to the 64 biorthogonal codes of length **32** and outputs the maximum correlation value and the index of a biorthogonal code corresponding to the maximum correlation value. The correlation comparator **1750** stores the maximum correlation value, the biorthogonal code index corresponding to the maximum correlation value, and the mask sequence index received from the mask sequence generator **1710**.

The above procedure is performed on all of the 16 mask sequences generated from the mask sequence generator **1710**. Then, 16 maximum correlation values the indexes of biorthogonal codes corresponding to the maximum correlation value are stored in the correlation comparator **1750**. The correlation comparator **1750** compares the stored 16 correlation values and selects the one with the highest correlation and outputs TFCI bits by combining the indexes of the biorthogonal code and mask sequence index corresponding to the selected maximum correlation value. When the decoding of the TFCI bits is completed, the input signal r(t) is deleted from the memory **1720** and the next input signal r(t+1) is stored.

While the correlation comparator **1750** compares the 16 maximum correlation values at one time in the decoding apparatus of FIG. **17**, real-time correlation value comparison can be contemplated. That is, the first input maximum correlation value is compared with the next input maximum correlation value and the larger of the two correlation values and a mask sequence index and a biorthogonal code index corresponding to the correlation are stored. Then, the thirdly input maximum correlation is compared with the stored correlation and the larger of the two correlations and a mask sequence index and a biorthogonal code index corresponding to the selected correlation are stored. This comparison operation occurs 15 times which is the number of mask sequences generated from the mask sequence generator **1710**. Upon completion of all the operations, the correlation comparator **1710** output the finally stored biorthogonal index (a**0** to a**6**) and mask sequence index (a**7** to a**9**) and outputs the added bits as TFCI bits.

FIG. **10** is a flowchart illustrating the operation of the correlation comparator **940** shown in FIG. **9**. The correlation comparator **940** stores the sixteen maximum correlation values, selects a highest correlation value out of the 16 maximum correlation values and output TFCI bits based on the indexes of a bi-orthogonal code and a mask sequence corresponding to the selected highest correlation value. The sixteen correlation values are compared, and TFCI bits are outputted based on the indexes of a bi-orthogonal code and a mask sequence corresponding to the highest correlation value.

Referring to FIG. **10**, a maximum correlation index i is set to 1 and the indices of a maximum correlation value, a biorthogonal code, and a mask sequence to be checked are set to 0s in step **1000**. In step **1010**, the correlation comparator **940** receives a 1^{st} maximum correlation value, a 1^{st} bi-orthogonal code index, and a 1^{st} mask sequence index from the correlation calculator **920**. The correlation comparator **940** compares the 1^{st} maximum correlation with an the previous maximum correlation value in step **1020**. If the 1^{st} maximum correlation is greater than the previous maximum correlation, the procedure goes to step **1030**. If the 1^{st} maximum correlation is equal to or smaller than the previous maximum correlation, the procedure goes to step **1040**. In step **1030**, the correlation comparator **940** designates the 1^{st} maximum correlation as a final maximum correlation and stores the 1^{st} bi-orthogonal code and mask sequence indexes as final bi-orthogonal code and mask sequence indexes. In step **1040**, the correlation comparator **940** compares the index i with the number 16 of the correlation calculators to determine whether all 16 maximum correlations are completely compared. If i is not 16, the index i is increased by 1 in step **1060** and the procedure returns to step **1010**. Then, the above procedure is repeated.

In step **1050**, the correlation comparator **940** outputs the indexes of the bi-orthogonal code and the mask sequence that correspond to the final maximum correlation as decoded bits. The bi-orthogonal code index and the mask sequence index corresponding to the decoded bits are those corresponding to the final maximum correlation among the 16 maximum correlation values received from the 16 correlation calculators.

3. Second Embodiment of Encoding/Decoding Apparatus and Method

The (32, 10) TFCI encoder that outputs a 32-symbol TFCI codeword in view of 16 slots has been described in the first embodiment of the present invention. Recently, the IMT-2000 standard specification dictates having 15 slots in one frame. Therefore, the second embodiment of the present invention is directed to a (30, 10) TFCI encoder that outputs a 30-symbol TFCI codeword in view of 15 slots. Therefore, the second embodiment of the present invention suggests an encoding apparatus and method for outputting 30 code symbols by puncturing two symbols of 32 coded symbols (codeword) as generated from the (32, 10) TFCI encoder.

The encoding apparatuses according to the first and second embodiments of the present invention are the same in configuration except that sequences output from a one-bit generator, a basis Walsh code generator, and a basis mask sequence generator. The encoder apparatus outputs coded symbols of length **30** with symbol #**0** (1^{st} symbol) and symbol #**16** (17^{th} symbol) are punctured in the encoding apparatus of the second embodiment.

Referring to FIG. **8**, 10 input information bits a**0** to a**9** are applied to the input of the **840** to **849**. The one-bit generator **800** outputs symbols 1s (length **32**) to the multiplier **840**. The multiplier **840** multiplies the input information bit a**0** by each 32 symbol received from the one-bit generator **800**. The basis Walsh code generator **810** simultaneously generates basis Walsh codes W1, W2, W4, W8, and W16 of length **32**. The multiplier **841** multiplies the input information bit a**1** by the basis Walsh code W1 "01010101010101010101010101010101". The multiplier **842** multiplies the input information bit a**2** by the basis Walsh code W2 "00110011001100110011001100110011". The multiplier **843** multiplies the input information bit a**3** by the basis Walsh code W4 "00001111000011110000111100001111". The multiplier **844** multiplies the input information bit a**4** by the basis Walsh code W8 "00000000111111110000000011111111". The multiplier **845** multiplies the input information bit a**5** by the basis Walsh code W16 "00000000000000001111111111111111".

The basis mask sequence generator **820** simultaneously generates basis mask sequences M1, M2, M4, and M8 of length **32**. The multiplier **846** multiplies the input information bit a**6** by the basis mask sequence M1 "00101000011000111111000001110111". The multiplier **847** multiplies the input information bit a**7** by the basis mask sequence M2 "00000001110011010110110111000111". The multiplier **848** multiplies the input information bit a**8** by the basis mask sequence M4 "00001010111110010001101100101011". The multiplier **849** multiplies the input information bit a**9** by the basis mask sequence M8 "00011100001101110010111101010001". The multipliers **840** to **849** function like switches that control the output of or the generation of the bits from the one-bit generator, each of the basis Walsh codes and each of the basis mask sequences.

The adder **860** sums the outputs of the multipliers **840** to **849** symbol by symbol and outputs 32 coded symbols (i.e., a TFCI codeword). Out of the 32 coded symbols, two symbols will be punctured at predetermined positions (i.e. the symbol #**0** (the first symbol) and symbol #**16** (the 17^{th} symbol) of the adder **860** output are punctured). The remaining 30 symbols will become the 30 TFCI symbols. It will be easy to modify the second embodiment of present invention. For example, the one-bit generator **800**, basis Walsh generator **810**, basis mask sequence generator **820** can generate 30 symbols which excludes the #**0** and #**16** symbols. The adder **860** then adds the output of the one-bit generator **800**, basis Walsh generator **810** and basis mask sequence generator **820** bit by bit and output 30 encoded symbols as TFCI symbols.

FIG. **12** is a encoding method for the second embodiment of present invention. The flowchart illustrating the steps of the encoding apparatus according to the second embodiment of the present invention when the number of slots is 15.

Referring to FIG. **12**, 10 input information bits a**0** to a**9** are received and variables sum and j are set to an initial value 0 in step **1200**. In step **1210**, it is determined whether j is 30. If j is not 30 in step **1210**, the j^{th} symbols W1(j), W2(j), W4(j), W8(j), and W16(j) of the basis Walsh codes W1, W2, W4, W8, and W16 (each having two punctured bits) and the j^{th} symbols M1(j), M2(j), M4(j), and M8(j) of the basis mask sequences M1, M2, M4, and M8 (each having two punctured bits) are received in step **1220**. Then, the received symbols are multiplied by the input information bits on a symbol basis and the multiplied symbols are summed in step **1230**. In step **1240**, sum indicating the achieved j^{th} code symbol is output. j is increased by 1 in step **1250** and then the procedure returns to step **1210**. Meanwhile, if j is 30 in step **1210**, the encoding procedure ends.

The (30, 10) encoder outputs 1024 codewords equivalent to the codewords of the (32, 10) encoder with symbols #**0** and #**16** punctured. Therefore, the total number of information can be expressed is 1024.

The output of a (30, 9) encoder is combinations of 32 Walsh codes of length **30** obtained by puncturing symbols #**0** and #**16** of each of 32 Walsh codes of length **32**, 32 bi-orthogonal codes obtained by adding 1 to each symbol of the punctured Walsh codes (by multiplying −1 to each symbol in the case of a real number), and 8 mask sequences obtained by combining any three of the four punctured basis mask sequences.

The output of a (30, 8) encoder is combinations of 32 Walsh codes of length **30** obtained by puncturing #**0** and #**16** symbols from each of 32 Walsh codes having a length **32** symbols, 32 bi-orthogonal codes obtained by adding 1 to each symbol of the punctured Walsh codes (by multiplying −1 to each symbol in the case of a real number), and 4 mask sequences obtained by combining any two of the four punctured basis mask sequences.

The output of a (30, 7) encoder is combinations of 32 Walsh codes of length **30** obtained by puncturing #**0** and #**16** symbols from each of 32 Walsh codes having a length **32** symbols, 32 bi-orthogonal codes obtained by adding 1 to each symbol of the punctured Walsh codes (by multiplying −1 to each symbol in the case of a real number), and one of the four punctured basis mask sequences.

All the above encoders for providing an extended TFCI have a minimum distance of 10. The (30, 9), (30, 8), and (30, 7) encoders can be implemented by blocking input and output of at least one of the four basis mask sequences generated from the basis mask sequence generator **820** shown in FIG. **8**.

The above encoders flexibly encode TFCI bits according to the number of the TFCI bits and has a maximized minimum distance that determines encoding performance.

A decoding apparatus according to the second embodiment of the present invention is the same in configuration and operation as the decoding apparatus of the first embodiment except for different signal lengths of the encoded symbols. That is, after (32,10) encoding, two symbols out of the 32 encoded symbols are punctured, or basis Walsh codes with two punctured symbols and basis mask sequences with two punctured symbols are used for generating the 30 encoded symbols. Therefore, except for the received signal r(t) which includes a signal of 30 encoded symbols and insertion of dummy signals at the punctured positions, all decoding operations are equal to the description of the first embodiment of present invention.

As FIG. **17**, this second embodiment of decoding also can be implemented by a single multiplier for multiplying the masks with r(t) and a single correlation calculator for calculating correlation values of bi-orthogonal codes.

4. Third Embodiment of Encoding/Decoding Apparatus and Method

The third embodiment of the present invention provides an encoding apparatus for blocking the output of a one-bit generator in the (30, 7), (30, 8), (30, 9) or (30, 10) (hereinafter we express (30, 7-10)) encoder of the second embodiment and generating another mask sequence instead in order to set a minimum distance to 11. The encoders refer to an encoder that outputs a 30-symbol TFCI codeword for the input of 7, 8, 9 or 10 TFCI bits.

FIG. **14** is a block diagram of a third embodiment of the encoding apparatus for encoding a TFCI in the IMT 2000 system. In the drawing, a (30, 7-10) encoder is configured to have a minimum distance of 11.

The encoding apparatus of the third embodiment is similar in structure to that of the second embodiment except that a mask sequence generator **1480** for generating a basis mask sequence M16 and a switch **1470** for switching the mask sequence generator **1480** and a one-bit generator **1400** to a multiplier **1440** are further provided to the encoding apparatus according to the third embodiment of the present invention.

The two bit punctured basis mask sequences M1, M2, M4, M8, and M16 as used in FIG. **14** are

M1=000001011111000010110100111110
M2=000110001100110001111010110111
M4=010111100111101010000001100111
M8=011011001000001111011100001111
M16=100100011110011111000101010011

Referring to FIG. **14**, when a (30, 6) encoder is used, the switch **1470** switches the one-bit generator **1400** to the multiplier **1440** and blocks all the basis mask sequences generated from a basis mask sequence generator **1480**. The multiplier **1440** multiplies the symbols from the one-bit generator **1400** with the input information bit a**0**, symbol by symbol.

If a (30, 7-10) encoder is used, the switch **1470** switches the mask sequence generator **1480** to the multiplier **1440** and selectively uses four basis mask sequences generated from a basis mask sequence generator **1420**. In this case, 31 mask sequences M1 to M31 can be generated by combining 5 basis mask sequences.

The structure and operation of outputting code symbols for the input information bits a**0** to a**9** using multipliers **1440** to **1449** are the same as the first and second embodiments. Therefore, their description will be omitted.

As stated above, the switch **1470** switches the mask sequence generator **1480** to the multiplier **1440** to use the (30, 7-10) encoder, whereas the switch **1470** switches the one-bit generator **1400** to the multiplier **1440** to use the (30, 6) encoder.

For the input of 6 information bits, the (30, 6) encoder outputs a 30-symbol codeword by combining 32 Walsh codes of length **30** with 32 bi-orthogonal codes obtained by inverting the Walsh codes by the use of the one-bit generator **1400**.

For the input of 10 information bits, the (30, 10) encoder outputs a 30-symbol codeword by combining 32 Walsh codes of length **30** and **32** mask sequences generated using five basis mask sequences. Here, the five basis mask sequences are M1, M2, M4, M8, and M16, as stated above and the basis mask sequence M16 is output from the mask sequence generator **1480** that is added for the encoding apparatus according to the third embodiment of the present invention. Hence, 1024 codewords can be achieved from the (30, 10) encoder. The (30, 9) encoder outputs a 30-symbol codeword by combining 32 Walsh codes and 16 mask sequences, for the input of 9 information bits. The 16 mask sequences are achieved by combining four of five basis mask sequences. The (30, 8) encoder outputs a 30-symbol codeword by combining 32 Walsh codes and 8 mask sequences, for the input of 8 information bits. The 8 mask sequences are obtained by combining three of five basis mask sequences. For the input of 7 information bits, the (30, 7) encoder outputs a 30-symbol codeword by combining 32 Walsh codes of length **30** and four mask sequences. The four mask sequences are obtained by combining two of five basis mask sequences.

All the above (30, 7-10) encoders have a minimum distance of 11 to provide extended TFCIs. The (32, 7-10) encoders can be implemented by controlling use of at least one of the five basis mask sequences generated from the basis mask sequence generator **1420** and the mask sequence generator **1480** shown in FIG. **14**.

FIG. **16** is a flowchart illustrating a third embodiment of the TFCI encoding procedure in the IMT 2000 system according to the present invention.

Referring to FIG. **16**, 10 information bits (TFCI bits) a**0** to a**9** are received and variables sum and j are set to initial values 0s in step **1600**. The variable sum indicates a final code symbol output after symbol-basis addition and the variable j indicates the count number of final code symbols output after the symbol-basis addition. It is determined whether j is 30 in step **1610** in view of the length **30** of punctured Walsh codes and mask sequences used for encoding. The purpose of performing step **1610** is to judge whether the input information bits are encoded with respect to the 30 symbols of each Walsh code and the 30 symbols of each mask sequence.

If j is not 30 in step **1610**, which implies that encoding is not completed with respect to all the symbols of the Walsh codes and mask sequences, the j^{th} symbols W1(j), W2(j), W4(j), W8(j), and W16(j) of the basis Walsh codes W1, W2, W4, W8, and W16 and the j^{th} symbols M1(j), M2(j), M4(j), M8(j), and M16(j) of the basis mask sequences M1, M2, M4, M8, and M16 are received in step **1620**. In step **1630**, the input information bits are multiplied by the received symbols symbol by symbol and the symbol products are summed.

Step **1630** can be expressed as

$$\text{sum}=a0\cdot M16(j)+a1\cdot W1(j)+a2\cdot W2(j)+a3\cdot W4(j)+a4\cdot W8(j)+a5\cdot W16(j)+a6\cdot M1(j)+a7\cdot M2(j)+a8\cdot M4(j)+a9\cdot M8(j) \quad \text{(Eq. 10)}$$

As noted from Eq. 10, an intended code symbol is obtained by multiplying each input information bit by the symbols of a corresponding basis Walsh code or basis mask sequence and summing the products.

In step **1640**, sum indicating the achieved j^{th} code symbol is output. j is increased by 1 in step **1650** and then the procedure returns to step **1610**. Meanwhile, if j is 30 in step **1610**, the encoding procedure ends.

Now there will be given a description of the third embodiment of the decoding apparatus referring to FIG. **15**. An input signal r(t) which includes the 30 encoded symbols signal transmitted by a transmitter and two dummy symbols which have been inserted at the positions that have been punctured by the encoder is applied to 31 multipliers **1502** to **1506** and a correlation calculator **1520**. A mask sequence generator **1500** generates all possible 31 mask sequences of length **32** M1 to M31. The multipliers **1502** to **1506** multiply the mask sequences received from the mask sequence generator **1500** by the input signal r(t). If a transmitter encoded TFCI bits with a predetermined mask sequence, one of the outputs of the multipliers **1502** to **1506** is free of the mask sequence, which means the mask sequence has no effect on the following correlation calculator. For example, if the transmitter used the mask sequence M31 for encoding the TFCI bits, the output of the multiplier **1506** that multiplies the mask sequence M31 by the input signal r(t) is free of the mask sequence. However, if the transmitter did not use a mask sequence, the input signal r(t) itself applied to a correlation calculator **1520** is a mask sequence-free signal. Each correlation calculators **1520** to **1526** calculates the correlation values of the outputs of the multipliers **1502** to **1506** with 64 bi-orthogonal codes of length **32**, determines maximum correlation value among the 64-correlation sets, and outputs the determined maximum correlation values, the indexes of each bi-orthogonal codes corresponding to the determined maximum correlation values, and each index of the mask sequences to a correlation comparator **1540**, respectively.

The correlation comparator **1540** compares the 32 maximum correlation values received from the correlation calculators **1520** to **1526** and determines the largest of the maximum correlation values as a final maximum correlation. Then, the correlation comparator **1540** outputs the decoded TFCI bits transmitted by the transmitter on the basis of the indexes of the bi-orthogonal code and mask sequence corresponding to the final maximum correlation value. As in FIG. **17**, the third embodiment of present invention can be also implemented by a single multiplier for multiplying the masks with r(t) and a single correlation calculator for calculating correlation values of bi-orthogonal codes.

As described above, the present invention provides an apparatus and method for encoding and decoding a basic TFCI and an extended TFCI variably so that hardware is simplified. Another advantage is that support of both basic TFCI and extended TFCI error correcting coding schemes increases service stability. Furthermore, a minimum distance, a factor that determined the performance of an encoding apparatus, is large enough to satisfy the requirement of an IMT 2000 system, thereby ensuing excellent performance.

While the invention has been shown and described with reference to certain preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the invention as defined by the appended claims.

What is claimed is:

1. A Transport Format Combination Indicator (TFCI) encoding apparatus in a CDMA mobile communication system, comprising:

an orthogonal sequence generator for generating a plurality of basis biorthogonal sequences;

a mask sequence generator for generating a plurality of basis mask sequences; and

an operation unit for adding a basis biorthogonal sequence and a basis mask sequence selected among the basis biorthogonal sequences and the basis mask sequences according to TFCI bits.

2. The TFCI encoding apparatus of claim **1**, wherein the plurality of basis biorthogonal sequences are a first Walsh code, a second Walsh code, a fourth Walsh code, an eighth Walsh code, a sixteenth Walsh code and an all "1" sequence which converts the orthogonal sequences to the biorthogonal sequences.

3. The TFCI encoding apparatus of claim **2**, wherein the plurality of basis biorthogonal sequences are Walsh codes and bi-orthogonal complement sequences of the Walsh codes.

4. The TFCI encoding apparatus of claim **1**, wherein the mask sequence generator has a first m-sequence and a second m-sequence which can be added together to form a Gold code, forms a first sequence group having sequences formed by cyclically shifting the first m-sequence and a second sequence group having sequences formed by cyclically shifting the second m-sequence, generates and applies a column transposition function to the sequences in the first group to convert the sequences in the first group to orthogonal sequences, inserts a column of '0' in the front of the sequences in the second group, and generates and applies a reverse column transposition function to the sequences in the second group to convert the sequences in the second group to mask sequences.

5. The TFCI encoding apparatus of claim **1**, wherein the basis mask sequences are a first mask sequence "00101000011000111111000001110111", a second mask sequence "00000001110011010110110111000111", a fourth mask sequence "00001010111110010001101100101011", and an eighth mask sequence "00011100001101110010111101010001".

6. The TFCI encoding apparatus of claim **1**, wherein the operation unit comprises:

a plurality of first multipliers for multiplying the basis biorthogonal sequences by corresponding TFCI bits;

a plurality of second multipliers for multiplying the basis mask sequences by corresponding TFCI bits; and

an adder for adding the outputs of the first and second multipliers and generating the sum as a TFCI sequence.

7. The TFCI encoding apparatus of claim **6**, wherein the plurality of basis biorthogonal sequences are a first Walsh code, a second Walsh code, a fourth Walsh code, an eighth Walsh code, and a sixteenth Walsh code.

8. An apparatus for encoding Transport Format Combination Indicator (TFCI) bits including first information bits and second information bits in a CDMA mobile communication system, comprising:

an orthogonal sequence generator for generating a plurality of biorthogonal sequences and outputting a biorthogonal sequence selected based on the first information bits among the plurality of biorthogonal sequences;

a mask sequence generator for generating a plurality of mask sequences and outputting a mask sequence selected based on the second information bits among the plurality of mask sequences; and an adder for adding the biorthogonal sequence and the mask sequence received from the orthogonal sequence generator and the mask sequence generator.

9. The TFCI encoding apparatus of claim **8**, wherein the mask sequence generator has a first m-sequence and a second m-sequence which can be added together to form a Gold code, forms a first sequence group having sequences formed by cyclically shifting the first m-sequence and a second sequence group having sequences formed by cyclically shifting the second m-sequence, generates and applies a column transposition function to the sequences in the first group to convert the sequences in the first group to orthogonal sequences, inserts a column of '0' in the front of the sequences in the second group, and generates and applies a reverse column transposition function to the sequences in the second group to convert the sequences in the second group to the mask sequences.

10. A Transport Format Combination Indicator (TFCI) encoding method in a CDMA mobile communication system, comprising the steps of:

generating, by an orthogonal generator, a plurality of basis biorthogonal sequences;

generating a plurality of basis mask sequences; and

generating a TFCI sequence by adding a basis biorthogonal sequence and a basis mask sequence selected among the basis biorthogonal sequences and the basis mask sequences according to TFCI bits.

11. The TFCI encoding method of claim **10**, wherein the plurality of basis biorthogonal sequences are a first Walsh code, a second Walsh code, a fourth Walsh code, an eighth Walsh code, a sixteenth Walsh code and an all "1" sequence which converts the orthogonal sequences to the biorthogonal sequences.

12. The TFCI encoding method of claim **10**, wherein the step of generating the plurality of basis mask sequences, comprises the steps of:

providing a first m-sequence and a second m-sequence which can be added together to form a Gold code;

forming a first sequence group having sequences formed by cyclically shifting the first m-sequence and a second sequence group having sequences formed by cyclically shifting the second m-sequence;

generating and applying a column transposition function to the sequences in the first group to convert the sequences in the first group to orthogonal sequences;

inserting a column of '0' in the front of the sequences in the second group; and

generating and applying a reverse column transposition function to the sequences in the second group to convert the sequences in the second group to the plurality of basis mask sequences.

13. The TFCI encoding method of claim **10**, wherein the basis mask sequences are a first mask sequence "00101000011000111111000001110111", a second mask sequence "00000001110011010110110111000111", a fourth mask sequence "00001010111110010001101100101011", and an eighth mask sequence "00011100001101110010111101010001".

14. The TFCI encoding method of claim **10**, wherein the basis biorthogonal sequences are multiplied by corresponding TFCI bits, the basis mask sequences are multiplied by corresponding TFCI bits, and the multiplication results are added to the TFCI sequence in the TFCI sequence generating step.

15. A method of encoding Transport Format Combination Indicator (TFCI) bits including first information bits and second information bits in a CDMA mobile communication system, comprising the steps of:

generating, by an orthogonal generator, a plurality of biorthogonal sequences and outputting a biorthogonal sequence selected based on the first information bits among the plurality of biorthogonal sequences;

generating a plurality of mask sequences and outputting a mask sequence selected based on the second information bits among the plurality of mask sequences; and

adding the selected biorthogonal sequence and the selected mask sequence.

16. The TFCI encoding method of claim **15**, wherein the plurality of biorthogonal sequences are Walsh codes and complement codes of the Walsh codes.

17. The TFCI encoding method of claim **15**, wherein the step of generating the plurality of mask sequences, comprises the steps of:

providing a first m-sequence and a second m-sequence which can be added together to form a Gold code;

forming a first sequence group having sequences formed by cyclically shifting the first m-sequence and a second sequence group having sequences formed by cyclically shifting the second m-sequence;

generating and applying a column transposition function to the sequences in the first group to convert the sequences in the first group to orthogonal sequences;

inserting a column of '0' in the front of the sequences in the second group; and

generating and applying a reverse column transposition function to the sequences in the second group to convert the sequences in the second group to the plurality of mask sequences.

18. A Transport Format Combination Indicator (TFCI) encoding method in a CDMA mobile communication system, comprising the steps of:

generating, by an orthogonal generator, an all "1" sequence;

generating a plurality of basis orthogonal sequences;

generating a plurality of basis mask sequences;

receiving TFCI bits and multiplying the all "1" sequence by corresponding TFCI bits, the plurality of basis orthogonal sequences by corresponding TFCI bits, and the plurality of basis mask sequences by corresponding TFCI bits; and

adding the multiplication results.

19. The TFCI encoding method of claim **18**, wherein the plurality of basis orthogonal sequences are a first Walsh code, a second Walsh code, a fourth Walsh code, an eighth Walsh code, and a sixteenth Walsh code.

20. The TFCI encoding method of claim **18**, wherein the step of generating the plurality of basis mask sequences, comprises the steps of:

providing a first m-sequence and a second m-sequence which can be added together to form a Gold code;

forming a first sequence group having sequences formed by cyclically shifting the first m-sequence and a second sequence group having sequences formed by cyclically shifting the second m-sequence;

generating and applying a column transposition function to the sequences in the first group to convert the sequences in the first group to the orthogonal sequences;

inserting a column of '0' in the front of the sequences in the second group; and

generating and applying a reverse column transposition function to the sequences in the second group to convert the sequences in the second group to the plurality of basis mask sequences.

21. The TFCI encoding method of claim **18**, wherein the basis mask sequences are a first mask sequence "00101000011000111111000001110111", a second mask sequence "00000001110011010110110111000111", a fourth mask sequence "00001010111110010001101100101011", and an eighth mask sequence "00011100001101110010111101010001".

22. A Transport Format Combination Indicator (TFCI) decoding apparatus in a CDMA mobile communication system, comprising:

a mask sequence generator for generating at least one mask sequence;

at least one operation circuit for receiving an input signal and the generated mask sequence and removing the mask sequences from the input signal by multiplying the mask sequence by the input signal; and

at least one correlator for receiving the signal from the operation circuit, calculating correlation values of the received signal with a plurality of orthogonal sequences numbered with corresponding indexes, and selecting the largest of the calculated correlation values and an orthogonal sequence index corresponding to the largest correlation value.

23. The TFCI decoding apparatus of claim **22**, wherein the mask sequence generator has a first m-sequence and a second m-sequence which can be added together to form a Gold code, forms a first sequence group having sequences formed by cyclically shifting the first m-sequence and a second sequence group having sequences formed by cyclically shifting the second m-sequence, generates and applies a column transposition function to the sequences in the first group to convert the sequences in the first group to orthogonal sequences, inserts a column of '0' in the front of the sequences in the second group, and generates and applies a reverse column transposition function to the sequences in the second group to convert the sequences in the second group to the mask sequences.

24. The TFCI decoding apparatus of claim **22**, wherein the operation circuit is a multiplier.

25. The TFCI decoding apparatus of claim **22**, further comprising a correlation comparator for determining the largest correlation value received from a plurality of correlators and generating an orthogonal sequence index and a mask sequence index corresponding to the largest correlation value.

26. The TFCI decoding apparatus of claim **25**, wherein the mask sequence index is the index of the mask sequence used to remove a mask sequence from the input signal.

27. A Transport Format Combination Indicator (TFCI) decoding method in a CDMA mobile communication system, comprising the steps of:

generating, by a mask sequence generator, at least one mask sequence;

receiving an input signal and the generated mask sequence and removing a mask sequence from the input signal by multiplying the generated mask sequence by the input signal;

receiving a product signal, calculating correlation values of the product signal with a plurality of orthogonal sequences having corresponding indexes; and

selecting the largest correlation value from the calculated correlation values and outputting an orthogonal sequence index corresponding to the largest correlation value.

28. The TFCI decoding method of claim **27**, wherein the step of generating at least one mask sequence, comprises the steps of:

providing a first m-sequence and a second m-sequence which can be added together to form a Gold code;

forming a first sequence group having sequences formed by cyclically shifting the first m-sequence and a second sequence group having sequences formed by cyclically shifting the second m-sequence;

generating and applying a column transposition function to the sequences in the first group to convert the sequences in the first group to orthogonal sequences;

inserting a column of '0' in the front of the sequences in the second group; and

generating and applying a reverse column transposition function to the sequences in the second group to convert the sequences in the second group to the at least one mask sequence.

29. The TFCI decoding method of claim **27**, further comprising the step of

determining the highest correlation value from the selected largest correlation values obtained by selecting the largest correlation value from the calculated correlation values;

and outputting an orthogonal sequence index and a mask sequence index corresponding to the determined highest correlation value.

30. The TFCI decoding method of claim **29**, wherein the mask sequence index is the index of the mask sequence used to remove a mask sequence from the input signal corresponding to the highest correlation value.

31. A Transport Format Combination Indicator (TFCI) decoding method in a CDMA mobile communication system, comprising the steps of:

generating, by a mask sequence generator, a plurality of mask sequences;

receiving an input signal and the generated mask sequences and removing a mask sequence from the input signal by multiplying the generated mask sequences by the input signal;

receiving the product signals, calculating correlation values of each of the product signals with a plurality of orthogonal sequences having corresponding indexes, and selecting the largest correlation values and orthogonal sequence indexes corresponding to the largest correlation values; and

determining a highest correlation value from the largest correlation values and outputting an orthogonal sequence index and a mask sequence index corresponding to the determined highest correlation value.

32. The TFCI decoding method of claim **31**, wherein the step of generating the plurality of mask sequences, comprises the steps of:

providing a first m-sequence and a second m-sequence which can be added together to form a Gold code;

forming a first sequence group having sequences formed by cyclically shifting the first m-sequence and a second sequence group having sequences formed by cyclically shifting the second m-sequence;

generating and applying a column transposition function to the sequences in the first group to convert the sequences in the first group to orthogonal sequences;

inserting a column of '0' in the front of the sequences in the second group; and

generating and applying a reverse column transposition function to the sequences in the second group to convert the sequences in the second group to the plurality of mask sequences.

33. The TFCI decoding method of claim **31**, wherein the mask sequence index is the index of the mask sequence used to remove a mask sequence from the input signal corresponding to the highest correlation value.

34. A Transport Format Combination Indicator (TFCI) decoding method in a CDMA mobile communication system, comprising the steps of:

generating, by a mask sequence generator, a plurality of mask sequences;

receiving an input signal and the generated mask sequences and multiplying each mask sequence by the input signal;

receiving the multiplied signals and calculating correlation values of each of the received multiplied signals with a plurality of orthogonal sequences having corresponding indexes;

selecting a largest correlation value among the calculated correlation values for each of the multiplied signals and an orthogonal sequence index corresponding to the largest correlation value; and

determining a highest correlation value from all of the largest correlation values and an orthogonal code index corresponding to the highest correlation value.

35. The TFCI decoding method of claim **34**, wherein the step of generating the plurality of mask sequence, comprises the steps of:

providing a first m-sequence and a second m-sequence which can be added together to form a Gold code;

forming a first sequence group having sequences formed by cyclically shifting the first m-sequence and a second sequence group having sequences formed by cyclically shifting the second m-sequence;

generating and applying a column transposition function to the sequences in the first group to convert the sequences in the first group to orthogonal sequences;

inserting a column of '0' in the front of the sequences in the second group; and

generating and applying a reverse column transposition function to the sequences in the second group to convert the sequences in the second group to the plurality of mask sequences.

36. A Transport Format Combination Indicator (TFCI) encoding apparatus in a CDMA mobile communication system, comprising:

a basis sequence generator for receiving TFCI information bits in a 10 bit unit and outputting at least one basis sequence selected based on the TFCI information bits from among all basis sequences available for encoding; and

a codeword generator for combining at least two basis sequences output from the basis sequence generator and outputting a combined basis sequence,

wherein the combined basis sequence is a codeword, and the at least one basis sequence and the codeword comprises 32 bits and all the basis sequences are mapped with the TFCI information bits in a 10 bit unit.

37. The TFCI encoding apparatus of claim **36**, wherein all the basis sequences are "01010101010101010101010101010101", "00110011001100110011001100110011", "00001111000011110000111100001111", "00000000111111110000000011111111", "00000000000000001111111111111111", "11111111111111111111111111111111", "00101000011000111111000001110111", "0000000111001101011011011100011", "00001010111110010001101100101011" and "00011100001101110010111101010001".

38. The TFCI encoding apparatus of claim **37**, wherein if the TFCI information bits are less than 10 bits, 0 is added to the TFCI information bits to represent the TFCI information bits in a 10 bit unit.

39. The TFCI encoding apparatus of claim **38**, wherein the basis sequence generator selects the at least one basis sequence by multiplying the TFCI information bits by their corresponding basis sequences.

40. The TFCI encoding apparatus of claim **39**, further comprising a puncturer for puncturing two bits from the 32 bits of codeword and outputting 30 bits of codeword.

41. The TFCI encoding apparatus of claim **40**, wherein the output 30 bit codewords is at least one of "101010101010101101010101010101", "011001100110011011001100110011", "000111100001111000111100001111", "000000011111111000000011111111", "000000000000000111111111111111", "111111111111111111111111111111", "010100001100011111000001110111", "000000111001101110110111000111", "000101011111001001101100101011" and "001110000110111010111101010001".

42. A Transport Format Combination Indicator (TFCI) encoding apparatus in a CDMA mobile communication system, comprising:

a basis sequence generator for receiving TFCI information bits in a 10 bit unit and outputting at least one basis sequence selected based on the TFCI information bits from among all basis sequences available for encoding; and

a codeword generator for combining at least two basis sequences output from the basis sequence generator and outputting a combined basis sequence,

wherein the combined basis sequence is a codeword, and the at least one basis sequence and the codeword comprises 30 bits and all the basis sequences are mapped with the TFCI information bits in a 10 bit unit.

43. The TFCI encoding apparatus of claim **42**, wherein all the basis sequences are "101010101010101101010101010101", "011001100110011011001100110011", "000111100001111000111100001111", "000000011111111000000011111111", "000000000000000111111111111111", "1111111111111111111111111111111", "010100001100011111000001110111", "000000111001101110110111000111", "000101011111001001101100101011" and "001110000110111010111101010001".

44. The TFCI encoding apparatus of claim **43**, wherein if the TFCI information bits are less than 10 bits, 0 is added to the TFCI information bits to represent the TFCI information bits in a 10 bit unit.

45. The TFCI encoding apparatus of claim **44**, wherein the basis sequence generator selects the at least one basis sequence by multiplying the TFCI information bits by their corresponding basis sequences.

46. A method for encoding a Transport Format Combination Indicator (TFCI) in a CDMA mobile communication system, comprising:

inputting TFCI information bits in a 10 bit unit;

generating, by a codeword generator, a codeword containing 32 bits based on the TFCI information bits; and

outputting the generated codeword,

wherein the codeword is generated by combining at least two basis sequences selected by the TFCI information bits from among all basis sequences available for encoding and all the basis sequences are mapped with the TFCI information bits in a 10 bit unit.

47. The method of claim **46**, wherein all the basis sequences are “01010101010101010101010101010101”, “00110011001100110011001100110011”, “00001111000011110000111100001111”, “00000000111111110000000011111111”, “00000000000000001111111111111111”, “11111111111111111111111111111111”, “00101000011000111111000001110111”, “00000001110011010110110111000111”, “00001010111110010001101100101011” and “00011100001101110010111101010001”.

48. The method of claim **47**, wherein if the TFCI information bits are less than 10 bits, 0 is added to the TFCI information bits to represent the TFCI information bits in a 10 bit unit.

49. The method of claim **48**, wherein the at least one basis sequence is selected by multiplying the TFCI information bits by their corresponding basis sequences.

50. The method of claim **49**, further comprising the step of puncturing two bits from the 32 bits of codeword and outputting 30 bits of codeword.

51. The method of claim **50**, wherein the output 30 bit codeword is at least one of “101010101010101101010101010101”, “011001100110011011001100110011”, “000111100001111000111100001111”, “000000011111111000000011111111”, “000000000000000111111111111111”, “111111111111111111111111111111”, “010100001100011111000001110111”, “000000111001101110110111000111”, “000101011111001001101100101011” and “001110000110111010111101010001”.

52. A method for encoding a Transport Format Combination Indicator (TFCI) in a CDMA mobile communication system, comprising:

inputting TFCI information bits in a 10 bit unit;

generating, by a codeword generator, a codeword containing 30 bits based on the TFCI information bits; and

outputting the generated codeword,

wherein the codeword is generated by combining at least two basis sequences selected by the TFCI information bits from among all basis sequences available for encoding and all the basis sequences are mapped with the TFCI information bits in a 10 bit unit.

53. The method of claim **52**, wherein all the basis sequences are “101010101010101101010101010101”, “011001100110011011001100110011”, “000111100001111000111100001111”, “000000011111111000000011111111”, “000000000000000111111111111111”, “111111111111111111111111111111”, “010100001100011111000001110111”, “000000111001101110110111000111”, “000101011111001001101100101011” and “001110000110111010111101010001”.

54. The method of claim **53**, wherein if the TFCI information bits are less than 10 bits, 0 is added to the TFCI information bits to represent the TFCI information bits in a 10 bit unit.

55. The method of claim **54**, wherein the at least one basis sequence is selected by multiplying the TFCI information bits by their corresponding basis sequences.

56. A method for encoding a Transport Format Combination Indicator (TFCI) in a CDMA mobile communication system, comprising:

inputting TFCI information bits in a 10 bit unit;

generating, by a codeword generator, a code word containing 32 bits based on the TFCI information bits; and

outputting the generated code word,

wherein if a TFCI information bit having a value of 1 is among the TFCI information bits, the generated codeword is generated by outputting a basis sequence corresponding to the bit having the value of 1 selected from among all basis sequences as the codeword, and if there are more than one TFCI information bits having a value of 1, the generated codeword is generated by adding bits of a plurality of basis sequences corresponding to bits having the value of 1 and outputting one of the added basis sequences as the codeword.

57. The method of claim **56**, wherein all the basis sequences are “01010101010101010101010101010101”, “00110011001100110011001100110011”, “00001111000011110000111100001111”, “00000000111111110000000011111111”, “00000000000000001111111111111111”, “11111111111111111111111111111111”, “00101000011000111111000001110111”, “00000001110011010110110111000111”, “00001010111110010001101100101011” and “00011100001101110010111101010001”.

58. The method of claim **57**, wherein if the TFCI information bits are less than 10 bits, 0 is added to the TFCI information bits to represent the TFCI information bits in a 10 bit units.

59. The method of claim **58**, wherein the basis sequence is selected by multiplying the TFCI information bits by their corresponding basis sequences.

60. The method of claim **59**, further comprising for the step of puncturing two bits from among the 32 bits of codeword and outputting 30 bits of codeword.

61. A method for encoding a Transport Format Combination Indicator (TFCI) in a CDMA mobile communication system, comprising:

inputting TFCI information bits in a 10 bit unit;

generating, by a codeword generator, a code word containing 30 bits based on the TFCI information bits; and

outputting the generated code word,

wherein if a TFCI information bit having a value of 1 is among the TFCI information bits, the codeword is generated by outputting a basis sequence corresponding to the bit having the value of 1 selected from among all basis sequences as the codeword, and if there are more than one TFCI information bits having a value of 1, the codeword is generated by adding bits of a plurality of basis sequences corresponding to bits having the value of 1 and outputting one of the added basis sequences as the codeword.

62. The method of claim **61**, wherein all the basis sequences are “101010101010101101010101010101”, “011001100110011011001100110011”, “000111100001111000111100001111”, “000000011111111000000011111111”, “000000000000000111111111111111”, “111111111111111111111111111111”, “010100001100011111000001110111”, “000000111001101110110111000111”, “000101011111001001101100101011” and “001110000110111010111101010001”.

63. The method of claim **62**, wherein if the TFCI information bits are less than 10 bits, 0 is added to the TFCI information bits to represent the TFCI information bits in a 10 bit unit.

64. The method of claim 63, wherein the basis sequence is generated by selecting the at least one basis sequence by multiplying the TFCI information bits by their corresponding basis sequences.

65. The method of claim 64, further comprising for the step of puncturing two bits from the 32 bits of codeword and outputting 30 bits of codeword.

66. The TFCI decoding method of claim 34, further comprising the step of generating a mask sequence index corresponding to the highest correlation value.

67. The TFCI encoding apparatus of claim 1, wherein said mask sequence generator generates, as said basis mask sequences, sequences of length 30 corresponding to basis mask sequences of length 32 from which the symbols at positions 0 and 16 are excluded.

68. The TFCI encoding apparatus of claim 5, wherein said mask sequence generator generates, as said basis mask sequences, sequences of length 30 corresponding to basis mask sequences of length 32 from which the symbols at positions 0 and 16 are excluded.

69. The TFCI encoding apparatus of claim 8, wherein said mask sequence generator generates, as said mask sequences, sequences of length 30 corresponding to mask sequences of length 32 from which the symbols at positions 0 and 16 are excluded.

70. The TFCI encoding method of claim 10, wherein the step of generating the plurality of basis mask sequences generates, as the plurality of basis mask sequences, sequences of length 30 corresponding to basis mask sequences of length 32 from which the symbols at positions 0 and 16 are excluded.

71. The TFCI encoding method of claim 13, wherein the step of generating the plurality of basis mask sequences generates, as the plurality of basis mask sequences, sequences of length 30 corresponding to basis mask sequences of length 32 from which the symbols at positions 0 and 16 are excluded.

72. The TFCI encoding method of claim 15, wherein the step of generating the plurality of mask sequences generates, as the plurality of mask sequences, sequences of length 30 corresponding to mask sequences of length 32 from which the symbols at positions 0 and 16 we excluded.

73. The TFCI encoding method of claim 18, wherein the step of generating the plurality of basis mask sequences generates, as the plurality of basis mask sequences, sequences of length 30 corresponding to basis mask sequences of length 32 from which the symbols at positions 0 and 16 are excluded.

74. The TFCI encoding method of claim 21, wherein the step of generating the plurality of basis mask sequences generates, as the plurality of basis mask sequences, sequences of length 30 corresponding to basis mask sequences of length 32 from which the symbols at positions 0 and 16 we excluded.

75. A Transport Format Combination Indicator (TFCI) encoding apparatus in a COMA mobile communication system, comprising:

a controller for outputting a 30 bit codeword from among a plurality of 30 bit codewords that corresponds to a 10 bit TFCI information input to the controller from a plurality of possible 10 bit TFCI information,

wherein the 30 bit codeword output by the controller is equivalent to a 32 bit codeword that corresponds to the 10 bit TFCI information input to the controller.

76. The TFCI encoding apparatus of claim 75, wherein each of the plurality of possible 10 bit TFCI information and each of the plurality of 30 bit codewords correspond to each other based on a combination of a basis orthogonal sequence, a basis mask sequence, and an all "1" sequence, the basis orthogonal sequence and the basis mask sequence being two bit punctured equivalents of a basis orthogonal sequence and a basis mask sequence corresponding to the equivalent 32 bit codeword.

77. The TFCI encoding apparatus of claim 76, wherein a length of the combination of the basis orthogonal sequence, the basis mask sequence and the all "1" sequence is identical to a length of the TFCI information.

78. The TFCI decoding method of claim 66, wherein the mask sequence index is the index of the mask sequence used to remove a mask sequence from the input signal corresponding to the highest correlation value.

79. A method for encoding a Transport Format Combination Indicator (TFCI) in a CDMA mobile communication system, comprising:

inputting a TFCI information into a controller;

outputting, via the controller, a 32 bit codeword from among a plurality of 32 bit codewords that corresponds to a 10 bit TFCI information input to the controller from a plurality of possible 10 bit TFCI information;

puncturing, via a puncturer, two bits from the 32 bit codeword, each of the two bits being punctured at a predetermined position; and

outputting a 30 bit codeword that is equivalent to the outputted 32 bit codeword.

80. The method of claim 79, wherein each of the plurality of possible 10 bit TFCI information and each of the plurality of 32 bit codewords correspond to each other based on a combination of a basis orthogonal sequences, a basis mask sequences, and an all "1" sequence.

81. The method of claim 80, wherein a total number of the basis orthogonal sequences, the basis mask sequences and the all "1" sequence is identical to a number of bits of each TFCI information.

82. A Transport Format Combination Indicator (TFCI) encoding apparatus in a CDMA mobile communication system, comprising:

a controller for outputting a 32 bit codeword from among a plurality of 32 bit codewords that corresponds to a 10 bit TFCI information input to the controller from a plurality of possible 10 bit TFCI information; and

a puncturer for puncturing two bits from the 32 bit codeword output by the controller, each of the two bits being punctured at a predetermined position, and outputting a 30 bit codeword that is equivalent to the 32 bit codeword output by the controller.

83. The TFCI encoding apparatus of claim 82, wherein each of the plurality of possible 10 bit TFCI information and each of the plurality of 32 bit codewords correspond to each other based on a combination of a basis orthogonal sequences, a basis mask sequences, and an all "1" sequence.

84. The TFCI encoding apparatus of claim 83, wherein a total number of the basis orthogonal sequences, the basis mask sequences and the all "1" sequence are identical to a number of bits of each TFCI information.

85. The TFCI encoding apparatus of claim 76, wherein each of two bits punctured from each of the basis orthogonal sequences and the basis mask sequences are punctured at a predetermined position.

86. The TFCI encoding apparatus of claim **85**, wherein the predetermined positions are the same for each of the basis orthogonal sequences and each of the basis mask sequences.

87. The method of claim **79**, wherein the predetermined positions are the same for each of the outputted 32 bit codeword from among the plurality of 32 bit codewords.

88. The TFCI encoding apparatus of claim **82**, wherein the predetermined positions are the same for each of the outputted 32 bit codeword from among the plurality of 32 bit codewords.

* * * * *

Exhibit 2

US007486644B2

(12) United States Patent
Kim et al.

(10) Patent No.: US 7,486,644 B2
(45) Date of Patent: Feb. 3, 2009

(54) **METHOD AND APPARATUS FOR TRANSMITTING AND RECEIVING DATA WITH HIGH RELIABILITY IN A MOBILE COMMUNICATION SYSTEM SUPPORTING PACKET DATA TRANSMISSION**

(75) Inventors: **Young-Bum Kim**, Seoul (KR); **Yujian Zhang**, Haidian Distric Beijing (CN); **Ju-Ho Lee**, Suwon-si (KR); **Yong-Jun Kwak**, Yongin-si (KR); **Youn-Hyoung Heo**, Suwon-si (KR); **Joon-Young Cho**, Suwon-si (KR)

(73) Assignee: **Samsung Electronics Co., Ltd**, Suwon-Si (KR)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 617 days.

(21) Appl. No.: **11/289,572**

(22) Filed: **Nov. 30, 2005**

(65) **Prior Publication Data**

US 2006/0156184 A1 Jul. 13, 2006

(30) **Foreign Application Priority Data**

Dec. 1, 2004 (KR) 10-2004-0099917
Dec. 22, 2004 (KR) 10-2004-0110552
Jan. 27, 2005 (KR) 10-2005-0007437

(51) **Int. Cl.**
H04B 7/216 (2006.01)

(52) **U.S. Cl.** **370/335**; 370/355; 714/776; 714/790; 714/802

(58) **Field of Classification Search** 370/335, 370/355; 375/152; 714/776, 790, 802
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

2002/0170013 A1 11/2002 Bolourchi et al.
2003/0133497 A1* 7/2003 Kinjo et al. 375/152

FOREIGN PATENT DOCUMENTS

EP 1 339 188 A1 8/2003
JP 11502679 3/1999
JP 2004527949 9/2004
WO WO03085873 10/2003

OTHER PUBLICATIONS

LG Electronics, Performance of E-AGCH signaling in HSUPA, 3GPP TSG RAN WGI Meeting #39, Shin-Yokohama, Japan, Nov. 15-19, 2004.
NTT DOCOOMO, Performance of Common Absolute Grants: 2ms vs. 10ms (5 times repetition of 2ms),, TSG-RAN Working Group 1 #39 meeting, Shin-Yokohama, Japan, Nov. 15-19, 2004.
Hagenauer J Ed; Institute of Electrical And Electronics Engineers: "Hybrid ARQ/FEC Protocols On Fading Channels Using Rate Compatible Punctures . . . "; pp. 744-748.

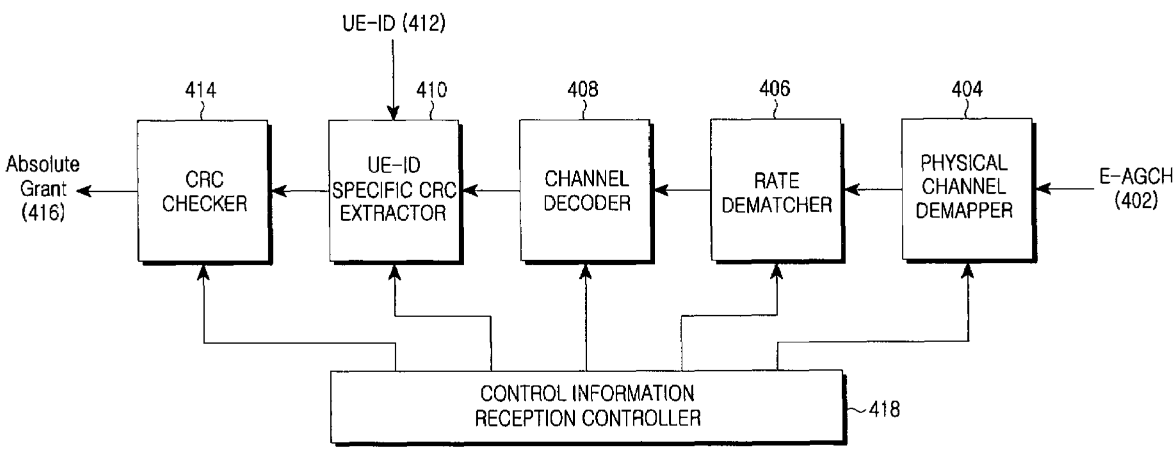
* cited by examiner

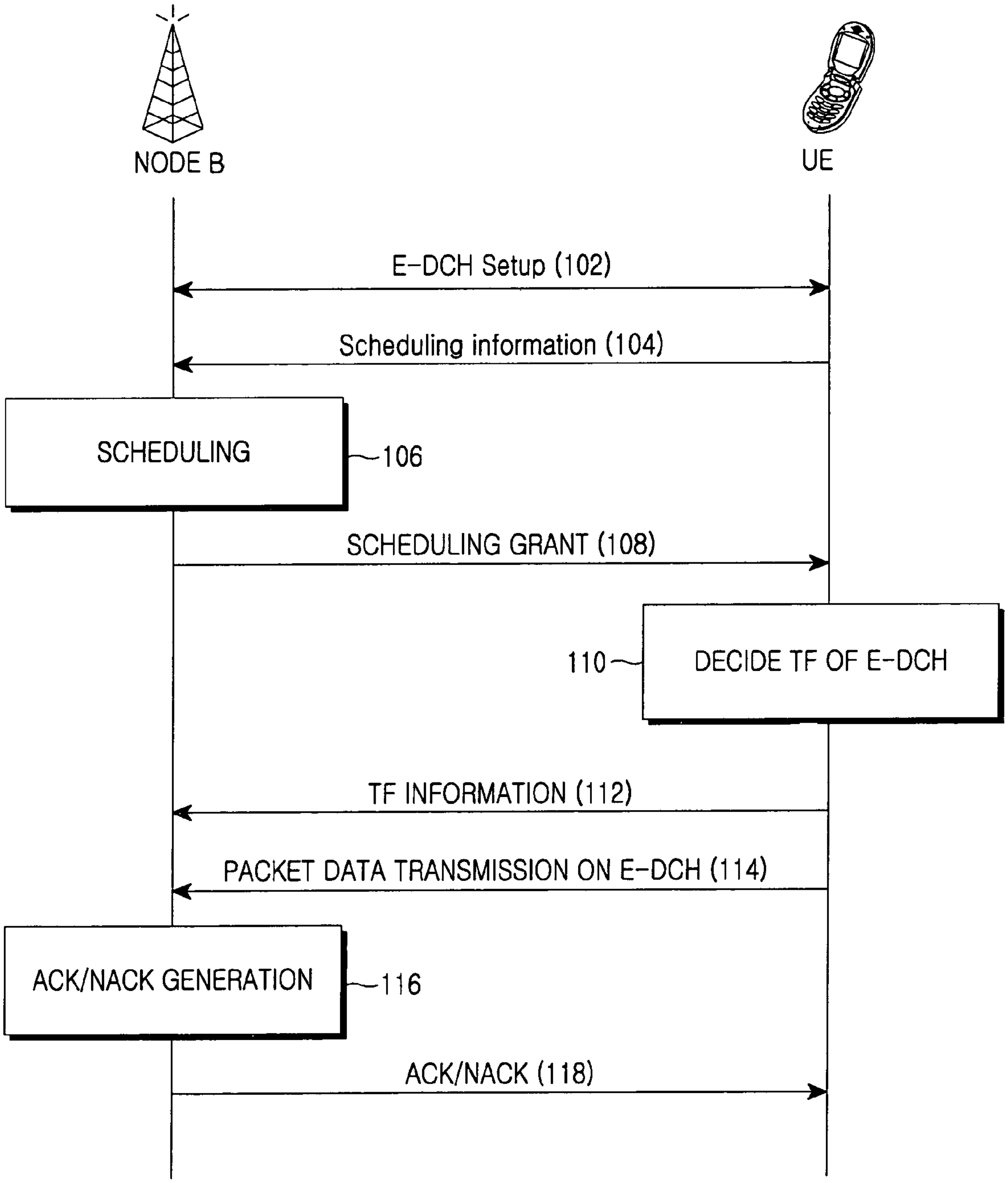
Primary Examiner—Temica M Beamer
(74) *Attorney, Agent, or Firm*—Roylance, Abrams, Berdo & Goodman, L.L.P.

(57) **ABSTRACT**

A method and apparatus for transmitting control information of a small block size with high reliability in a mobile communication system supporting uplink packet data service are provided. A 6-bit Absolute Grant indicating an allowed maximum data rate for uplink packet data transmission is generated and a 16-bit User Equipment Identifier Cyclic Redundancy Check is generated by combining a Cyclic Redundancy Check with a User Equipment Identifier. The User Equipment Identifier specific Cyclic Redundancy Check and 8 tail bits are added to the 6-bit Absolute Grant and the added bits are encoded at a coding rate of 1/3. The resulting 90 coded bits are rate-matched according to a predetermined rate matching pattern and transmitted to a User Equipment. The rate matching pattern is {1, 2, 5, 6, 7, 11, 12, 14, 15, 17, 23, 24, 31, 37, 44, 47, 61, 63, 64, 71, 72, 75, 77, 80, 83, 84, 85, 87, 88, 90}.

16 Claims, 5 Drawing Sheets





NODE B
UE
E-DCH Setup (102)
Scheduling information (104)
SCHEDULING
106
SCHEDULING GRANT (108)
110
DECIDE TF OF E-DCH
TF INFORMATION (112)
PACKET DATA TRANSMISSION ON E-DCH (114)
ACK/NACK GENERATION
116
ACK/NACK (118)

FIG.1

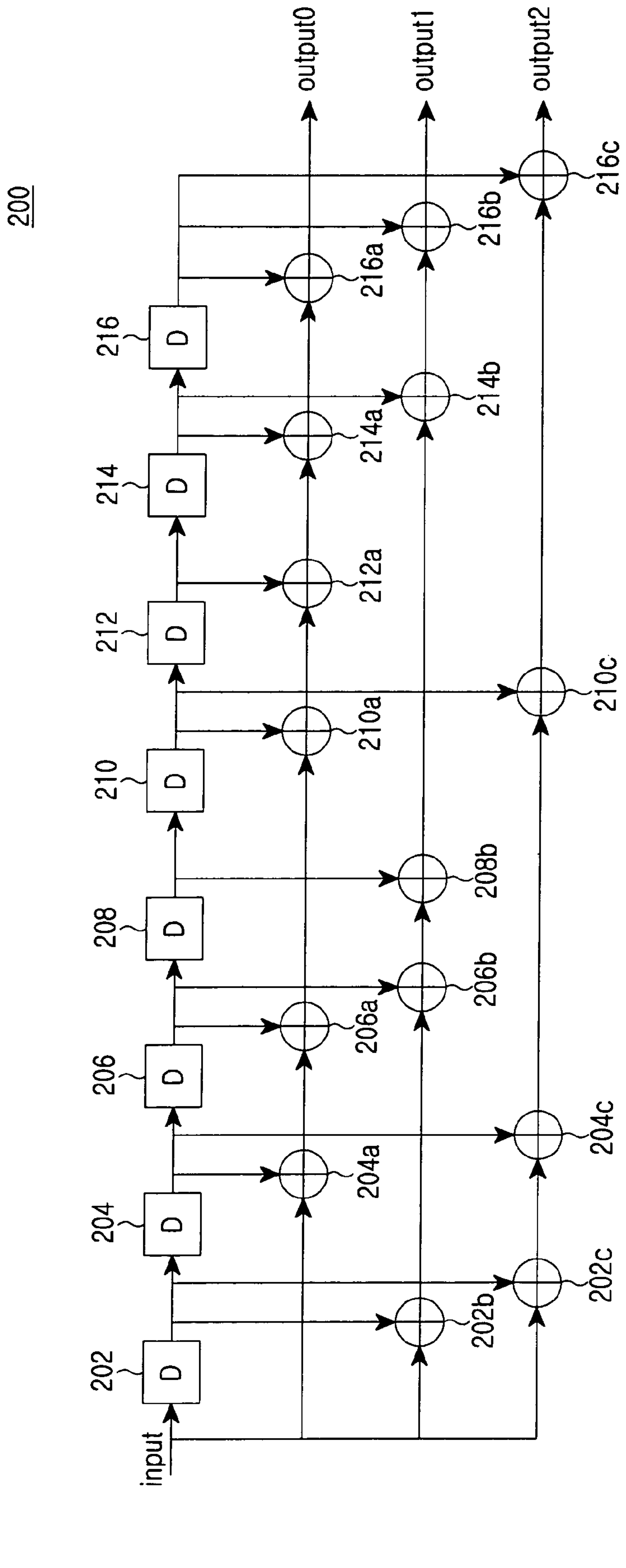

200
input
output0
output1
output2
D
202
204
206
208
210
212
214
216
202b
202c
204a
204c
206a
206b
208b
210a
210c
212a
214a
214b
216a
216b
216c

FIG.2A

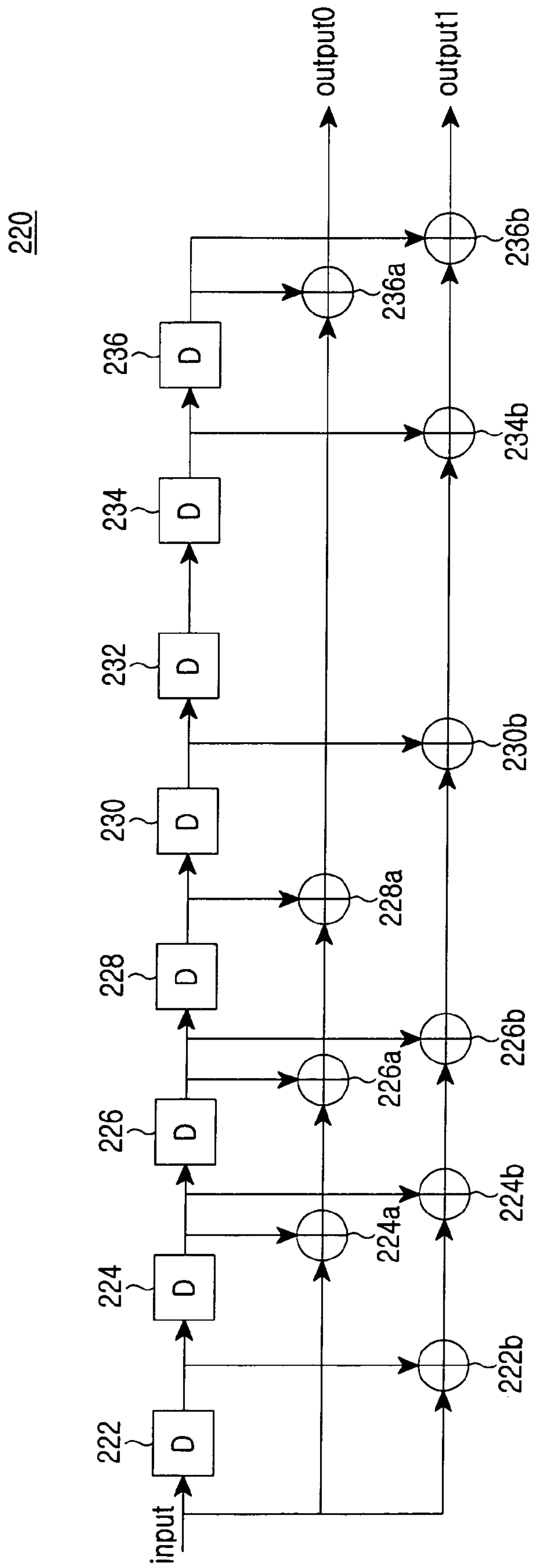

220
input
output0
output1
D
222
224
226
228
230
232
234
236
222b
224a
224b
226a
226b
228a
230b
234b
236a
236b

FIG.2B

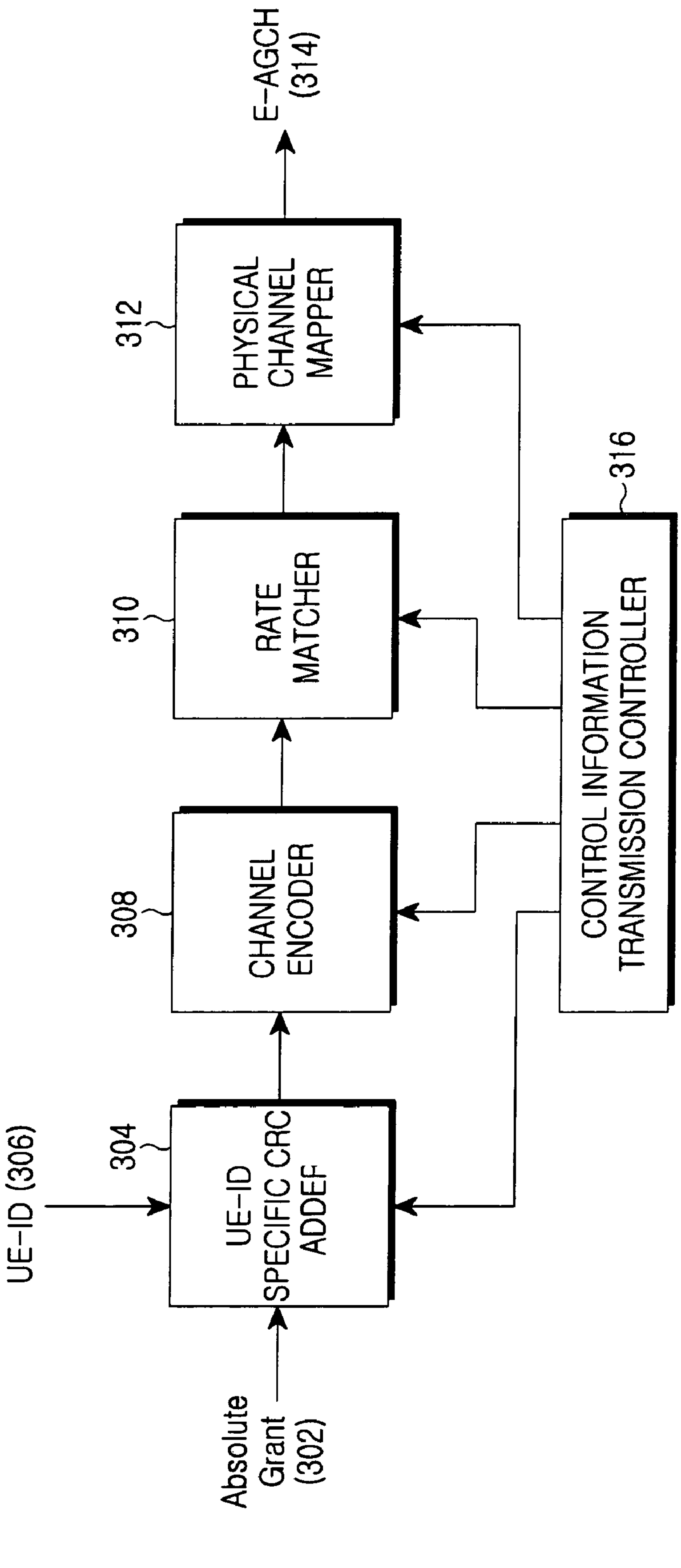

E-AGCH (314)
312
PHYSICAL CHANNEL MAPPER
310
RATE MATCHER
308
CHANNEL ENCODER
304
UE-ID SPECIFIC CRC ADDEF
UE-ID (306)
Absolute Grant (302)
316
CONTROL INFORMATION TRANSMISSION CONTROLLER

FIG.3

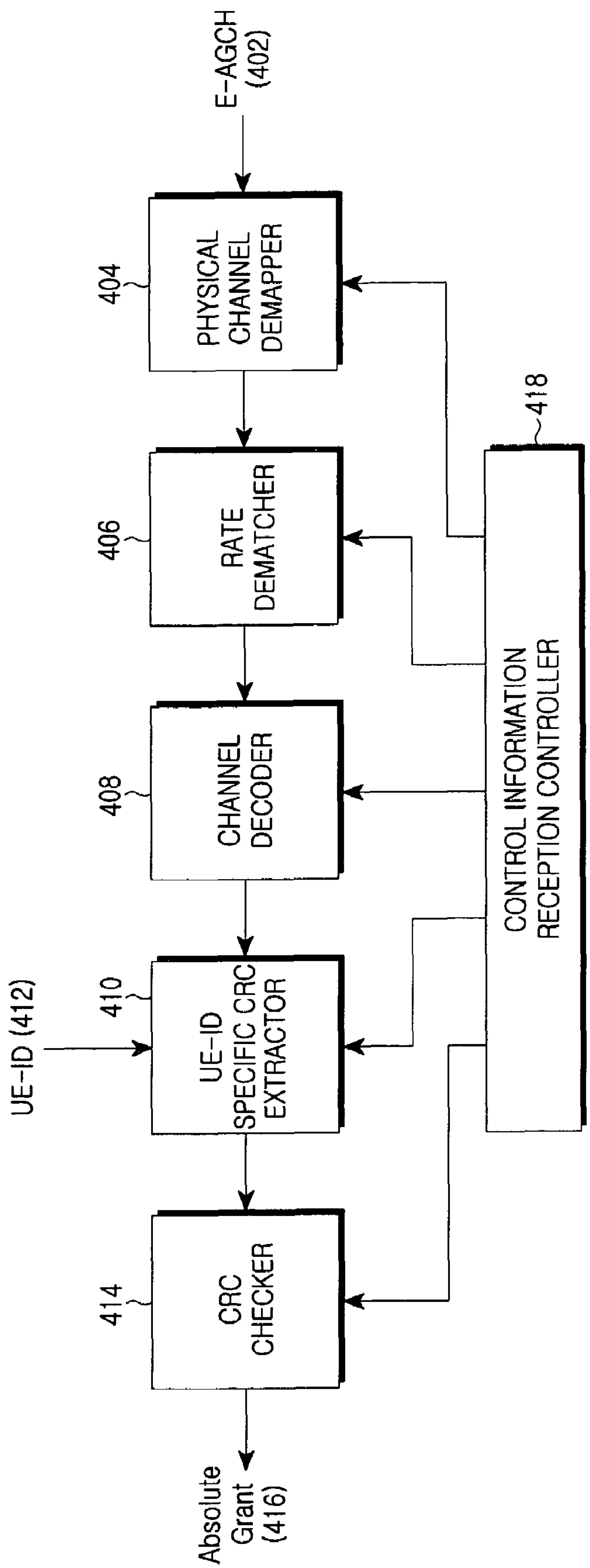

E-AGCH (402)
404
PHYSICAL CHANNEL DEMAPPER
406
RATE DEMATCHER
408
CHANNEL DECODER
410
UE-ID (412)
UE-ID SPECIFIC CRC EXTRACTOR
414
CRC CHECKER
Absolute Grant (416)
418
CONTROL INFORMATION RECEPTION CONTROLLER

FIG.4

METHOD AND APPARATUS FOR TRANSMITTING AND RECEIVING DATA WITH HIGH RELIABILITY IN A MOBILE COMMUNICATION SYSTEM SUPPORTING PACKET DATA TRANSMISSION

CROSS REFERENCE TO RELATED APPLICATIONS

This application claims priority under 35 U.S.C. § 119(a) to an application entitled "Method and Apparatus for Transmitting and Receiving Data with High Reliability in a Mobile Communication System Supporting Packet Data Transmission" filed in the Korean Intellectual Property Office on Dec. 1, 2004 and assigned Ser. No. 2004-99917, to an application entitled "Method and Apparatus for Transmitting and Receiving Data with High Reliability in a Mobile Communication System Supporting Packet Data Transmission" filed in the Korean Intellectual Property Office on Dec. 22, 2004 and assigned Ser. No. 2004-110552, and to an application entitled "Method and Apparatus for Transmitting and Receiving Data with High Reliability in a Mobile Communication System Supporting Packet Data Transmission" filed in the Korean Intellectual Property Office on Jan. 27, 2005 and assigned Ser. No. 2005-7437, the entire disclosures of all three of which are hereby incorporated by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to a mobile communication system supporting packet data service. More particularly, the present invention relates to a method and apparatus for transmitting control information required for Hybrid Automatic Repeat request (HARQ).

2. Description of the Related Art

Universal Mobile Telecommunication Service (UMTS), which is a 3rd generation mobile communication system using Wideband Code Division Multiple Access (WCDMA) based on the European Global System for Mobile communications (GSM) system and General Packet Radio Services (GPRS), provides mobile subscribers or computer users with a uniform service of transmitting packet-based text, digitized voice, and video and multimedia data at or above 2 Mbps irrespective of their locations around the world.

Particularly the UMTS system uses a transport channel called Enhanced Uplink Dedicated CHannel (E-DCH or EUDCH) in order to further improve the packet transmission performance of uplink communications from a User Equipment (UE) to a Node B (interchangeable with a base station). For more stable high-speed data transmission, Adaptive Modulation and Coding (AMC), HARQ, shorter Transmission Time Interval (TTI), and Node B-controlled scheduling were introduced for the E-DCH transmission.

AMC is a technique of determining a Modulation and Coding Scheme (MCS) adaptively according to the channel status between the Node B and the UE. Many MCS levels can be defined according to available modulation schemes and coding schemes. The adaptive selection of an MCS level according to the channel status increases resource use efficiency.

HARQ is a packet retransmission scheme for retransmitting a packet to correct errors in an initially transmitted packet. Shorter TTI is a technique for reducing retransmission time delay and thus increasing system throughput by allowing the use of a shorter TTI than the shortest TTI of 10

ms provided by 3GPP Rel5. At present, 2 ms is under consideration as the length of such a shorter TTI.

Node B-controlled scheduling is a scheme in which the Node B determines whether to permit E-DCH transmission for the UE and if it does, an allowed maximum data rate and transmits the determined data rate information as a scheduling grant to the UE, and the UE determines an available E-DCH data rate based on the scheduling grant.

The Node B-controlled scheduling is performed such that the noise rise or Rise over Thermal (RoT) measurement of the Node B does not exceed a target RoT to increase total system performance by, for example, allocating low data rates to remote UEs and high data rates to nearby UEs. RoT represents uplink radio resources used by the Node B, defined as

$$RoT=I_o/N_o \quad (1)$$

where I_o denotes power spectral density over a total reception band, that is, the total amount of uplink signals received in the Node B, and No denotes the thermal noise power spectral density of the Node B. Therefore, an allowed maximum RoT is total uplink radio resources available to the Node B.

The total RoT is expressed as the sum of inter-cell interference, voice traffic and E-DCH traffic. With Node B-controlled scheduling, simultaneous transmission of packets from a plurality of UEs at high data rates is prevented, maintaining the total RoT at or below a target RoT and thus ensuring reception performance.

FIG. **1** is a diagram illustrating a typical signal flow for data transmission and reception on the E-DCH. In the illustrated case of FIG. **1**, a UE transmits uplink data on the E-DCH and a Node B performs Node B-controlled scheduling for the UE.

Referring to FIG. **1**, the Node B and UE establish the E-DCH in step **102**. Step **102** involves message transmission on dedicated transport channels. The UE transmits scheduling information to the Node B in step **104**. The scheduling information may contain uplink channel status information including the transmit power and power margin of the UE, and the amount of buffered data to be transmitted to the Node B.

In step **106**, the Node B monitors scheduling information from a plurality of UEs to schedule uplink data transmissions for the individual UEs. When the Node B decides to approve an uplink packet transmission from the UE, it transmits a scheduling grant including scheduling assignment information to the UE in step **108**. The scheduling grant indicates up/hold/down in an allowed maximum data rate, or an allowed maximum data rate and allowed transmission timing. In step **110**, the UE determines the TF of the E-DCH (E-TF) based on the scheduling grant. The UE then transmits E-TF information to the Node B in step **112**, and uplink packet data on the E-DCH as well in step **114**.

The Node B determines whether the E-TF information and the uplink packet data have errors in step **116**. In the presence of errors in either of the TF information and the uplink packet data, the Node B transmits a negative acknowledgement (NACK) signal to the UE on an acknowledgement/negative acknowledgement (ACK/NACK) channel, whereas in the absence of errors in both, the Node B transmits an acknowledgement (ACK) signal to the UE on the ACK/NACK channel in step **118**. In the latter case, the packet data transmission is completed and the UE transmits new packet data to the Node B on the E-DCH. On the other hand, in the former case, the UE retransmits the same packet data to the Node B on the E-DCH.

For efficient scheduling under the above-described environment, the Node B receives scheduling information about buffer occupancy and power status from UEs. Based on the scheduling information, the Node B allocates low data rates to remote UEs, UEs in a bad channel status, and UEs having data with a low service class and high data rates to nearby UEs, UEs in a good channel status, and UEs having data with a high service class. In this context, a need exists for developing a technique for transmitting and receiving a scheduling grant, which can be an Absolute Grant (AG) indicating the absolute value of an allowed maximum data rate for a UE or a Relative Grant (RG) indicating up/hold/down from the previous allowed maximum data rate.

SUMMARY OF THE INVENTION

An aspect of the present invention is to address at least the above problems and/or disadvantages and to provide at least the advantages described below. Accordingly, an aspect of the present invention is to provide a method and apparatus for improving the transmission reliability of control information of a small block size such as an E-DCH scheduling grant.

The present invention also provides a method and apparatus for transmitting information with a higher reliability requirement, such as an Absolute Grant (AG) indicating an allowed maximum data rate for a UE.

The above exemplary objects are achieved by providing a method and apparatus for transmitting control information of a small block size with high reliability in a mobile communication system supporting uplink packet data service.

According to one aspect of an exemplary embodiment of the present invention, in a method of transmitting control information associated with uplink packet data transmission in a mobile communication system, a 16-bit UE-ID specific CRC is generated by combining a CRC generated for detecting errors from the control information with a User Equipment Identifier (UE-ID) for identifying a UE to receive the control information. 90 coded bits are generated by adding the UE-ID specific CRC and 8 tails bits to 6-bit control information and encoding the added bits at a coding rate of 1/3. A 60-bit rate-matched block is generated by rate-matching the coded bits according to a predetermined rate matching pattern representing the positions of bits to be punctured among the coded bits. The rate-matched block is transmitted to the UE. The rate matching pattern is {1, 2, 5, 6, 7, 11, 12, 14, 15, 17, 23, 24, 31, 37, 44, 47, 61, 63, 64, 71, 72, 75, 77, 80, 83, 84, 85, 87, 88, 90}.

According to another aspect of the an exemplary embodiment of the present invention, in an apparatus of transmitting control information associated with uplink packet data transmission in a mobile communication system, a UE-ID specific CRC generator generates a 16-bit UE-ID specific CRC by combining a CRC generated for detecting errors from the control information with a UE-ID for identifying a UE to receive the control information. A channel encoder generates 90 coded bits by adding the UE-ID specific CRC and 8 tails bits to 6-bit control information and encoding the added bits at a coding rate of 1/3. A rate matcher generates a 60-bit rate-matched block by rate-matching the coded bits according to a predetermined rate matching pattern representing the positions of bits to be punctured among the coded bits. A physical channel mapper transmits the rate-matched block to the UE. The rate matching pattern is {1, 2, 5, 6, 7, 11, 12, 14, 15, 17, 23, 24, 31, 37, 44, 47, 61, 63, 64, 71, 72, 75, 77, 80, 83, 84, 85, 87, 88, 90}.

According to a further aspect of the an exemplary embodiment of the present invention, in a method of receiving control information associated with uplink packet data transmission in a mobile communication system, a 60-bit rate-matched block is extracted from a signal received from a Node B. 90 coded bits are generated by rate-dematching the rate-matched block according to a predetermined rate matching pattern representing the positions of bits to be depunctured. 6-bit control information and a 16-bit UE-ID specific CRC are obtained by decoding the coded bits at a coding rate of 1/3. The control information is output by checking the UE-ID specific CRC. The rate matching pattern is {1, 2, 5, 6, 7, 11, 12, 14, 15, 17, 23, 24, 31, 37, 44, 47, 61, 63, 64, 71, 72, 75, 77, 80, 83, 84, 85, 87, 88, 90}.

According to still another aspect of the an exemplary embodiment of the present invention, in an apparatus for receiving control information associated with uplink packet data transmission in a mobile communication system, a physical channel demapper extracts a 60-bit rate-matched block from a signal received from a Node B. A rate dematcher generates 90 coded bits by rate-dematching the rate-matched block according to a predetermined rate matching pattern representing the positions of bits to be depunctured. A channel decoder generates 6-bit control information and a 16-bit UE-ID specific CRC by decoding the coded bits at a coding rate of 1/3. A CRC checker outputs the control information by checking the UE-ID specific CRC. The rate matching pattern is {1, 2, 5, 6, 7, 11, 12, 14, 15, 17, 23, 24, 31, 37, 44, 47, 61, 63, 64, 71, 72, 75, 77, 80, 83, 84, 85, 87, 88, 90}.

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other exemplary objects, features and advantages of the present invention will become more apparent from the following detailed description when taken in conjunction with the accompanying drawings in which:

FIG. **1** is a diagram illustrating a typical signal flow for data transmission and reception on an E-DCH;

FIGS. **2**A and **2**B illustrate a rate 1/3 convolutional encoder and a rate 1/2 convolutional encoder, respectively;

FIG. **3** is a block diagram of a Node B transmitter according to a preferred embodiment of the present invention; and

FIG. **4** is a block diagram of a UE receiver according to a preferred embodiment of the present invention.

Throughout the drawings, the same drawing reference numerals will be understood to refer to the same elements, features, and structures.

DETAILED DESCRIPTION OF THE EXEMPLARY EMBODIMENTS

The matters defined in the description such as a detailed construction and elements are provided to assist in a comprehensive understanding of the embodiments of the invention. Accordingly, those of ordinary skill in the art will recognize that various changes and modifications of the embodiments described herein can be made without departing from the scope and spirit of the invention. Also, descriptions of well-known functions and constructions are omitted for clarity and conciseness.

Certain exemplary embodiments of the present invention will be described below in the context of the E-DCH of UMTS. A physical channel called E-DCH Absolute Grant CHannel (E-AGCH) carries an AG from a Node B to a UE. The AG is determined by the Node B scheduler according to scheduling information received from the UE and uplink radio resources available to the Node B.

The AG can include an allowed maximum data rate indicating the maximum amount of uplink radio resources available to the UE or a power offset equivalent to the allowed maximum data rate, an AG validity duration indicator indicating how long the AG is valid, and an AG validity process indicator indicating whether the AG is valid for one particular HARQ process or an entire HARQ process. The power offset is defined as a maximum power ratio of an Enhanced Dedicated Physical Data CHannel (E-DPDCH) on which the E-DCH is mapped to a reference physical channel whose power is controlled, Dedicated Physical Control CHannel (DPCCH). A 4 to 8-bit allowed maximum data rate or power offset, a 1-bit AG validity duration indicator, and a 1-bit AG validity process indicator are under consideration. The E-AGCH further requires a UE-ID for identifying the UE on a common channel and a Cyclic Redundancy Check (CRC) code for detecting errors from the AG. The UE-ID and the CRC each have 16 bits and are modulo-2 operated on a bit basis. They are included together with the AG in control information delivered on the E-AGCH, in the form of a 16-bit CRC masked with a UE-ID.

Therefore, the control information of the E-AGCH can be a total of 21 to 26 bits in length. This control information, particularly the AG is intended for efficient radio resource allocation and thus requires high reliability in transmission. Typically, communication systems use channel coding for high-reliability data transmission/reception. The channel coding enables a receiver to correct transmission errors by attaching redundancy information to transmission data.

A convolutional code with a constraint length of 9 and a coding rate of 1/3, defined in the 3GPP standards, can be taken as a channel coding technique for high-reliability E-AGCH transmission and reception. 21 to 26-bit control information to be delivered on the E-AGCH is attached with 8 tail bits and then encoded to 87 to 102 coded bits ((21+8)×3=87, (26+8)×3=102) through rate 1/3 convolutional coding.

FIG. **2**A illustrates a convolutional encoder **200** with a constraint length of 9 and a coding rate of 1/3defined in the 3GPP standards.

Referring to FIG. **2**A, the convolutional encoder **200** includes eight serial shift registers **202**, **204**, **206**, **208**, **210**, **212**, **214** and **216** and a plurality of adders **202***b*, **202***c*, **204***a*, **204***c*, **206***a*, **206***b*, **208***b*, **210***a*, **210***c*, **212***a*, **214***a*, **214***b*, **216***a*, **216***b* and **216***c* for receiving input information bits or the output bits of the shift registers **202** to **216**. Input information including eight tail bits sequentially pass through the shift registers **202** to **216**, starting from the first bit, and coded bits are produced in an order of output**0**, output**1**, output**2**, output**0**, output**1**, output**2**, ...

The channel-coded control information is delivered in a 2-ms TTI of the E-AGCH. If a Spreading Factor (SF) of 256 and Quadrature Phase Shift Keying (QPSK) are applied to the E-AGCH, a total of 60 bits can be transmitted in the 2-ms TTI. Therefore, 27 bits (=87−60) to 42 bits (=102−60) are punctured from the coded control information of the E-AGCH. Rate matching can be used for the puncturing. The rate matching matches the number of the channel-coded bits to that of transmittable bits on a physical channel by puncturing or repeating bits at predetermined positions in the channel-coded bit stream of a block. Generally, the punctured or repeated bit positions are equidistant in the rate matching.

However, rate matching of small-size control information, such as, 20 bits delivered on the E-AGCH makes it difficult to achieve an optimum BLock Error Rate (BLER). If such a block with a relatively small number of bits is convolutionally encoded and rate-matched, the start and end of the block experience low Bit Error Rates (BERs), while its middle has a high BER. Consequently, the BLER of the block is increased and the reliability of the E-AGCH is decreased. The existence of at least one erroneous bit in one block leads to a BLER. If a particular part of one block has a low BER but the remaining part has a high BER, this results in performance degradation rather than performance improvement from the BLER's point of view.

High-reliability (i.e. less erroneous) transmission and reception of the E-AGCH control information is achieved by reducing its BLER. Yet, the conventional rate matching leads to a large BER deviation at each bit position of the block.

In this context, the exemplary embodiments of the present invention are intended to provide rate matching, which minimizes BLER and enables transmission of control information of a small block size. To serve this purpose, rate matching patterns are proposed, which minimize the change of BER at each bit position of one block of control information and thus reduce BLER for the E-AGCH that delivers small-size control information of about 20 bits after convolutional channel coding.

Rate matching patterns which may improve the BLER performance of control information delivered on the E-AGCH maybe realized in certain exemplary embodiments of the present invention. First through sixth exemplary embodiments pertain to examples of rate matching patterns and corresponding transmission and reception for a 7-bit AG, an 8-bit AG, a 9-bit AG, a 10-bit AG and a 5-bit AG, respectively.

Exemplary Embodiment 1

A rate matching pattern for a 6-bit AG is provided as an example of an implementation of the present invention. For instance, the AG includes a 4-bit allowed maximum data rate representing the maximum amount of uplink radio resources available to a UE or a 4-bit power offset equivalent to the allowed maximum data rate, a 1-bit AG validity duration indicator for indicating how long the AG is valid, and a 1-bit AG validity process indicator for indicating whether the AG is valid for one particular HARQ process or an entire HARQ process. Or the AG includes a 5-bit allowed maximum data rate or power offset and a 1-bit AG validity duration indicator.

In a third example, the AG includes a 5-bit allowed maximum data rate or power offset and a 1-bit AG validity process indicator. Or the AG is configured to include an allowed maximum data rate or power offset, an AG validity duration indicator, an AG validity process indicator, and other E-AGCH control bits in a total of six bits.

Besides the AG, the E-AGCH carries a UE-ID for identifying a UE and a CRC for detecting errors from the AG. The UE-ID and the CRC each have 16 bits and modulo-2 operated on a bit basis. Thus they are transmitted in the form of a 16-bit CRC masked with the UE-ID. This 16-bit CRC is called a UE-ID specific CRC. The UE can determine whether a received AG is intended for the UE by the UE-ID specific CRC.

Eight tail bits are added to 22-bit control information with the 6-bit AG concatenated to the 16-bit UE-ID specific CRC and convolutionally encoded with a constraints length of 9 at a coding rate of 1/3. As a result, one channel-coded block being a 90-bit coded bit stream is produced. 30 bits are punctured from the 90-bit channel-coded block for transmission in a 2-ms E-AGCH TTI to which an SF of 256 and QPSK apply, creating a 60-bit rate-matched block. A rate matching pattern representing the positions of the punctured bits is simulated in a manner that decreases the change of BER at each bit position of the rate-matched block and thus improves BLER performance. The following rate matching patterns are available.

Rate matching pattern=

{1, 2, 5, 6,7, 11, 12, 14, 15, 17, 23, 24, 31, 37, 44, 47, 61, 63, 64, 71, 72, 75, 77, 80, 83, 84, 85, 87, 88, 90},

{1, 2, 3, 4, 5, 6, 7, 8, 12, 14, 15, 18, 24, 48, 51, 54, 57, 60, 63, 66, 75, 78, 80, 81, 83, 84, 86, 87, 89, 90},

{1, 2, 3, 4, 5, 6, 7, 8, 12, 14, 15, 18, 21, 24, 57, 60, 66, 69, 75, 78, 80, 81, 83, 84, 85, 86, 87, 88, 89, 90},

{1, 2, 3, 5, 6, 7, 8, 12, 14, 15, 18,23, 25, 48, 50, 52, 57, 59, 61, 71, 75, 77, 79, 80, 82, 84, 86, 87, 88, 89},

{1, 2, 3, 4, 5, 6, 7, 8, 12, 14, 15, 24, 42, 48, 54, 57, 60, 66, 75, 78, 80, 81, 83, 84, 85, 86, 87, 88, 89, 90},

{1, 2, 3, 4, 5, 6, 7, 8, 12, 14, 15, 18, 24, 48, 50, 52, 57, 59, 61, 66, 75, 78, 80, 81, 83, 84, 86, 87, 89, 90},

{1, 2, 3, 4, 5, 6, 7, 8, 12, 14, 15, 24, 42, 54, 57, 60, 66, 69, 75, 78, 80, 81, 83, 84, 85, 86, 87, 88, 89, 90}, or

{1, 2, 3, 5, 6, 7, 8, 10, 12, 14, 15, 18,23, 25, 50, 52, 57, 59, 61, 71, 75, 77, 79, 80, 82, 84, 86, 87, 88, 89}.

The elements of each of the rate matching patterns represent the positions of bits to be punctured among the channel-coded bits #1 to #90. Given a 10-ms E-AGCH TTI, the 2-ms E-AGCH TTI occurs five times.

FIG. **3** is a block diagram of a Node B transmitter for transmitting the E-AGCH according to an exemplary embodiment of the present invention.

Referring to FIG. **3**, upon input of a 6-bit AG **302**, a UE-ID specific CRC adder **304** creates a 16-bit CRC from the AG, generates a UE-ID specific CRC by modulo-2 operating the 16-bit CRC with a 16-bit UE-ID for identifying the UE to receive the AG, and then combines the UE-ID specific CRC with the AG, thereby producing 22-bit control information. A channel encoder **308**, which uses a convolutional code with a constraint length of 9 and a coding rate 1/3, adds eight tail bits to the 22-bit control information and convolutionally encodes the 30-bit information to a 90-bit coded block.

A rate matcher **310** punctures the 90-bit coded block in a predetermined rate matching pattern. A physical channel mapper **312** maps the rate-matched block to a physical channel frame configured to be suitable for 2-ms E-AGCH TTIs. Thus, the control information is transmitted on an E-AGCH **314**. Meanwhile, a control information transmission controller **316** controls transmission of the control information for an E-DCH through the UE-ID specific CRC adder **304**, the channel encoder **308**, the rate matcher **310**, and the physical channel mapper **312**. In an exemplary implementation, the control information transmission controller **316** manages the coding rate of the channel encoder **308** and the rate matching pattern of the rate matcher **310**. The control information transmission controller **316** stores at least one of the above rate matching patterns and applies one of the rate matching patterns to the rate matcher **310**. The rate matching pattern used is preset between the transmitter and a receiver. The control information transmission controller **316** can be incorporated into a packet data reception controller (not shown) for controlling reception of packet data on the E-DCH.

FIG. **4** is a block diagram of a UE receiver for receiving the E-AGCH according to an exemplary embodiment of the present invention.

Referring to FIG. **4**, the UE receives a signal on an E-AGCH **402**. A physical channel demapper **404** extracts a rate-matched block from a 2-ms TTI in the received signal. A rate dematcher **406** recovers (that is, depunctures) the bits punctured by the rate matcher **310** for the rate-matched block by filling 0 s at the punctured bit positions according to the rate matching pattern used in the rate matcher **310** of the Node B. If the E-AGCH **402** uses a 10-ms TTI with five repeated 2-ms TTIs, the physical channel demapper **404** and the rate dematcher **406** perform the same operation five times and combine the resulting coded sub-blocks to one coded block.

A channel decoder **408** decodes the coded block received from the rate dematcher **406**. The coded block is divided into a 6-bit AG and a 16-bit UE-ID specific CRC. A UE-ID specific CRC extractor **410** extracts a 16-bit CRC by modulo-2 operating the 16-bit UE-ID specific CRC with the 16-bit UE-ID **412** of the UE, and provides the extracted CRC and the AG to a CRC checker **414**. The CRC checker **414** checks the 16-bit CRC to detect errors from the AG. If the CRC check passes, the CRC checker **414** outputs the error-free AG **416**. The AG **416** is used for determining an allowed maximum data rate for E-DCH data. If the CRC check fails, the AG is discarded.

Meanwhile, a control information reception controller **418** controls the reception of the control information for the E-DCH through the physical channel demapper **404**, the rate dematcher **406**, the channel decoder **408**, the UE-ID specific CRC extractor **410**, and the CRC checker **414**. In an exemplary implementation, the control information reception controller **418** manages the rate matching pattern of the rate dematcher **406** and the coding rate of the channel decoder **408**. The control information reception controller **418** stores at least one of the above rate matching patterns and applies one of the rate matching patterns to the rate dematcher **406**. The control information reception controller **418** can be incorporated into a packet data transmission controller (not shown) for controlling transmission of packet data on the E-DCH.

Exemplary Embodiment 2

A rate matching pattern for a 7-bit AG is provided as an example of an implementation of the present invention. For instance, the AG includes a 5-bit allowed maximum data rate representing the maximum amount of uplink radio resources available to a UE or a 5-bit power offset equivalent to the allowed maximum data rate, a 1-bit AG validity duration indicator indicating how long the AG is valid, and a 1-bit AG validity process indicator indicating whether the AG is valid for one particular HARQ process or an entire HARQ process. Or the AG includes a 6-bit allowed maximum data rate or power offset and a 1-bit AG validity duration indicator.

In a third example, the AG includes a 6-bit allowed maximum data rate or power offset and a 1-bit AG validity process indicator. Or the AG is configured to include an allowed maximum data rate or power offset, an AG validity duration indicator, an AG validity process indicator, and other E-AGCH control bits in a total of seven bits.

Besides the AG, the E-AGCH carries a UE-ID for identifying a UE and a CRC for detecting errors from the AG. The UE-ID and the CRC each have 16 bits and modulo-2 operated on a bit basis. Thus they are transmitted in the form of a 16-bit CRC masked with the UE-ID. This 16-bit CRC is called a UE-ID specific CRC. The UE can determine whether a received AG is intended for the UE by the UE-ID specific CRC.

Eight tail bits are added to 23-bit control information with the 7-bit AG concatenated to the 16-bit UE-ID specific CRC and convolutionally encoded with a constraints length of 9 at a coding rate of 1/3. As a result, one channel-coded block being a 93-bit coded bit stream is produced. 33 bits are punctured from the 93-bit channel-coded block for transmission in a 2-ms E-AGCH TTI to which an SF of 256 and QPSK apply, creating a 60-bit rate-matched block. A rate matching pattern representing the positions of the punctured bits is simulated in a manner that decreases the change of BER at each bit position of the rate-matched block and thus improves BLER performance. The following rate matching patterns are available.

Rate matching pattern=

{1, 3, 4, 5, 7, 9, 11, 12, 13, 15, 17, 20, 23, 42, 45, 46, 50, 54, 70, 71, 74, 77, 80, 81, 82, 83, 85, 86, 87, 89, 90, 91, 93},

{1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 15, 21, 24, 42, 47, 54, 56, 58, 66, 68, 78, 81, 83, 84, 86, 87, 88, 89, 90, 91, 92, 93},

{1, 2, 3, 5, 6, 7, 8, 10, 12, 14, 15, 16, 21, 23, 42, 47, 49, 54, 56, 58, 66, 68, 73, 78, 80, 82, 83, 85, 87, 89, 90, 91, 92},

{1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 15, 21, 23, 25, 42, 47, 54, 56, 58, 66, 68, 73, 78, 80, 82, 83, 85, 87, 89, 90, 91, 92},

{1, 2, 3, 5, 6, 7, 8, 12, 14, 15, 16, 21, 23, 25, 42, 47, 49, 54, 56, 58, 66, 68, 75, 77, 79, 82, 83, 85, 87, 89, 90, 91, 92},

{1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 15, 21, 24, 42, 48, 54, 57, 60, 66, 69, 78, 81, 83, 84, 86, 87, 88, 89, 90, 91, 92, 93},

{1, 2, 3, 4, 5, 6, 7, 8, 12, 14, 15, 16, 21, 23, 28, 42, 49, 54, 56, 58, 66, 68, 74, 78, 80, 82, 83, 85, 87, 89, 90, 91, 92},

{1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 15, 18, 21, 24, 27, 54, 57, 60, 66, 69, 78, 81, 83, 84, 86, 87, 88, 89, 90, 91, 92, 93},

{1, 2, 3, 4, 5, 6, 7, 8, 12, 14, 15, 21, 24, 42, 48, 54, 57, 60, 63, 66, 69, 78, 81, 83, 84, 86, 87, 88, 89, 90, 91, 92, 93}, or

{1, 2, 3, 4, 5, 6, 7, 8, 12, 14, 15, 21, 24, 42, 48, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 83, 84, 86, 87, 89, 90, 92, 93}

The elements of each of the rate matching patterns represent the positions of bits to be punctured among the channel-coded bits #1 to #93. Given a 10-ms E-AGCH TTI, the 2-ms E-AGCH TTI occurs five times.

With reference to FIG. **3**, a Node B transmitter for transmitting the E-AGCH according to the second exemplary embodiment of the present invention will be described.

Referring to FIG. **3**, upon input of a 7-bit AG **302**, the UE-ID specific CRC adder **304** creates a 16-bit CRC from the AG, generates a UE-ID specific CRC by modulo-2 operating the 16-bit CRC with a 16-bit UE-ID for identifying the UE to receive the AG, and then combines the UE-ID specific CRC with the AG, thereby producing 23-bit control information. The channel encoder **308**, which uses a convolutional code with a constraint length of 9 and a coding rate 1/3, adds eight tail bits to the 23-bit control information and convolutionally encodes the 31-bit information to a 93-bit coded block.

The rate matcher **310** punctures the 93-bit coded block in a predetermined rate matching pattern. The physical channel mapper **312** maps the rate-matched block to a physical channel frame configured to be suitable for 2-ms E-AGCH TTIs. Thus, the control information is transmitted on the E-AGCH **314**. Meanwhile, the control information transmission controller **316** controls transmission of the control information for an E-DCH through the UE-ID specific CRC adder **304**, the channel encoder **308**, the rate matcher **310**, and the physical channel mapper **312**. The control information transmission controller **316** stores at least one of the above rate matching patterns and applies a preset one of the rate matching patterns to the rate matcher **310**.

With reference to FIG. **4**, a UE receiver for receiving the E-AGCH according to the second exemplary embodiment of the present invention will be described.

Referring to FIG. **4**, the UE receives a signal on the E-AGCH **402**. The physical channel demapper **404** extracts a rate-matched block from a 2-ms TTI in the received signal. The rate dematcher **406** recovers (i.e. depunctures) the bits punctured by the rate matcher **310** for the rate-matched block according to the rate matching pattern used in the rate matcher **310** of the Node B. If the E-AGCH **402** uses a 10-ms TTI with five repeated 2-ms TTIs, the physical channel demapper **404** and the rate dematcher **406** perform the same operation five times and combine the resulting coded sub-blocks to one coded block.

The channel decoder **408** decodes the coded block received from the rate dematcher **406**. The coded block is divided into a 7-bit AG and a 16-bit UE-ID specific CRC. The UE-ID specific CRC extractor **410** extracts a 16-bit CRC by modulo-2 operating the 16-bit UE-ID specific CRC with the 16-bit UE-ID **412** of the UE, and provides the extracted CRC and the AG to the CRC checker **414**. The CRC checker **414** checks the 16-bit CRC to detect errors from the AG. If the CRC check passes, the CRC checker **414** outputs the error-free AG **416**. The AG **416** is used for determining an allowed maximum data rate for E-DCH data. If the CRC check fails, the AG is discarded.

Meanwhile, the control information reception controller **418** controls the reception of the control information for the E-DCH through the physical channel demapper **404**, the rate dematcher **406**, the channel decoder **408**, the UE-ID specific CRC extractor **410**, and the CRC checker **414**.

Exemplary Embodiment 3

A rate matching pattern for an 8-bit AG is provided as an example of an implementation of the present invention. For instance, the AG includes a 6-bit allowed maximum data rate representing the maximum amount of uplink radio resources available to a UE or a 6-bit power offset equivalent to the allowed maximum data rate, a 1-bit AG validity duration indicator indicating how long the AG is valid, and a 1-bit AG validity process indicator indicating whether the AG is valid for one particular HARQ process or an entire HARQ process. Or the AG includes a 7-bit allowed maximum data rate or power offset and a 1-bit AG validity duration indicator.

In a third example, the AG includes a 7-bit allowed maximum data rate or power offset and a 1-bit AG validity process indicator. Or the AG is configured to include an allowed maximum data rate or power offset, an AG validity duration indicator, an AG validity process indicator, and other E-AGCH control bits in a total of eight bits.

Besides the AG, the E-AGCH carries a UE-ID for identifying a UE and a CRC for detecting errors from the AG. The UE-ID and the CRC each have 16 bits and modulo-2 operated on a bit basis. Thus they are transmitted in the form of a 16-bit CRC masked with the UE-ID. This 16-bit CRC is called a UE-ID specific CRC. The UE can determine whether a received AG is intended for the UE by the UE-ID specific CRC.

Eight tail bits are added to 24-bit control information with the 8-bit AG concatenated to the 16-bit UE-ID specific CRC and convolutionally encoded with a constraints length of 9 at a coding rate of 1/3. As a result, one channel-coded block being a 96-bit coded bit stream is produced. 36 bits are punctured from the 96-bit channel-coded block for transmission in a 2-ms E-AGCH TTI to which an SF of 256 and QPSK apply, creating a 60-bit rate-matched block. A rate matching pattern representing the positions of the punctured bits is simulated in a manner that decreases the change of BER at each bit position of the rate-matched block and thus improves BLER performance. The following rate matching patterns are available.

Rate matching pattern=

{1, 3, 4, 6, 7, 8, 11, 13, 14, 20, 22, 23, 24, 25, 32, 36, 40, 44, 47, 50, 58, 64, 70, 73, 76, 77, 79, 80, 83, 86, 88, 89, 92, 93, 94, 96}, {1, 2, 3, 4, 5, 6, 7, 8, 12, 14, 15, 19, 24, 29, 35, 37, 45, 47, 50, 54, 58, 62, 68, 75, 82, 85, 86, 87, 89, 90, 91, 92, 93, 94, 95, 96},

{1, 2, 3, 5, 6, 7, 9, 11, 13, 15, 21, 23, 25, 32, 41, 43, 48, 50, 52, 57, 59, 64, 69, 75, 77, 79, 82, 83, 86, 87, 88, 90, 92, 93, 94, 95},

{1, 2, 3, 5, 6, 7, 9, 11, 13, 15, 21, 23, 25, 30, 32, 41, 43, 48, 50, 52, 57, 59, 64, 69, 77, 79, 82, 83, 86, 87, 88, 90, 92, 93, 94, 95},

{1, 2, 3, 5, 6, 7, 9, 11, 13, 15, 21, 23, 25, 32, 48, 50, 52, 57, 59, 61, 66, 68, 70, 75, 77, 79, 82, 83, 86, 87, 88, 90, 92, 93, 94, 95},

{1, 2, 3, 5, 6, 7, 9, 11, 13, 15, 21, 23, 25, 30, 32, 34, 41, 43, 48, 50, 52, 57, 59, 64, 69, 79, 82, 83, 86, 87, 88, 90, 92, 93, 94, 95},

{1, 2, 3, 4, 5, 6, 7, 8, 12, 14, 15, 24, 27, 30, 33, 42, 45, 48, 51, 54, 57, 60, 66, 69, 81, 84, 86, 87, 89, 90, 91, 92, 93, 94, 95, 96},

{2, 3, 4, 5, 7, 9, 11, 13, 15, 21, 23, 25, 30, 32, 34, 39, 41, 43, 48, 50, 52, 57, 59, 64, 69, 80, 82, 84, 86, 87, 88, 90, 92, 93, 94, 95},

{1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 15, 17, 18, 20, 21, 24, 27, 57, 60, 63, 66, 69, 72, 81, 84, 86, 87, 89, 90, 91, 92, 93, 94, 95, 96},

{1, 2, 3, 4, 5, 6, 7, 8, 12, 14, 15, 24, 26, 28, 33, 42, 44, 46, 51, 53, 55, 60, 66, 69, 81, 84, 86, 87, 89, 90, 91, 92, 93, 94, 95, 96}, or

{1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 15, 24, 27, 30, 33, 42, 45, 48, 51, 54, 57, 60, 66, 81, 84, 86, 87, 89, 90, 91, 92, 93, 94, 95, 96}

The elements of each of the rate matching patterns represent the positions of bits to be punctured among the channel-coded bits #1 to #96. Given a 10-ms E-AGCH TTI, the 2-ms E-AGCH TTI occurs five times.

With reference to FIG. **3**, a Node B transmitter for transmitting the E-AGCH according to the third exemplary embodiment of the present invention will be described.

Referring to FIG. **3**, upon input of an 8-bit AG **302**, the UE-ID specific CRC adder **304** creates a 16-bit CRC from the AC, generates a UE-ID specific CRC by modulo-2 operating the 16-bit CRC with a 16-bit UE-ID for identifying the UE to receive the AG, and then combines the UE-ID specific CRC with the AG, thereby producing 24-bit control information. The channel encoder **308**, which uses a convolutional code with a constraint length of 9 and a coding rate 1/3, adds eight tail bits to the 24-bit control information and convolutionally encodes the 32-bit information to a 96-bit coded block.

The rate matcher **310** punctures the 96-bit coded block in a predetermined rate matching pattern. The physical channel mapper **312** maps the rate-matched block to a physical channel frame configured to be suitable for 2-ms E-AGCH TTIs. Thus, the control information is transmitted on the E-AGCH **314**.

Meanwhile, the control information transmission controller **316** controls transmission of the control information for an E-DCH through the UE-ID specific CRC adder **304**, the channel encoder **308**, the rate matcher **310**, and the physical channel mapper **312**. The control information transmission controller **316** stores at least one of the above rate matching patterns and applies a preset one of the rate matching patterns to the rate matcher **310**.

With reference to FIG. **4**, a UE receiver for receiving the E-AGCH according to the third exemplary embodiment of the present invention will be described.

Referring to FIG. **4**, the UE receives a signal on the E-AGCH **402**. The physical channel demapper **404** extracts a rate-matched block from a 2-ms TTI in the received signal. The rate dematcher **406** recovers (i.e. depunctures) the bits punctured by the rate matcher **310** for the rate-matched block according to the rate matching pattern used in the rate matcher **310** of the Node B. If the E-AGCH **402** uses a 10-ms TTI with five repeated 2-ms TTIs, the physical channel demapper **404** and the rate dematcher **406** perform the same operation five times and combine the resulting coded sub-blocks to one coded block.

The channel decoder **408** decodes the coded block received from the rate dematcher **406**. The coded block is divided into an 8-bit AG and a 16-bit UE-ID specific CRC. The UE-ID specific CRC extractor **410** extracts a 16-bit CRC by modulo-2 operating the 16-bit UE-ID specific CRC with the 16-bit UE-ID **412** of the UE, and provides the extracted CRC and the AG to the CRC checker **414**. The CRC checker **414** checks the 16-bit CRC to detect errors from the AG. If the CRC check passes, the CRC checker **414** outputs the error-free AG **416**. The AG **416** is used for determining an allowed maximum data rate for E-DCH data. If the CRC check fails, the AG is discarded.

Meanwhile, the control information reception controller **418** controls the reception of the control information for the E-DCH through the physical channel demapper **404**, the rate dematcher **406**, the channel decoder **408**, the UE-ID specific CRC extractor **410**, and the CRC checker **414**.

Exemplary Embodiment 4

A rate matching pattern for a 9-bit AG is provided as an example of an implementation of the present invention. For instance, the AG includes a 7-bit allowed maximum data rate representing the maximum amount of uplink radio resources available to a UE or a 7-bit power offset equivalent to the allowed maximum data rate, a 1-bit AG validity duration indicator indicating how long the AG is valid, and a 1-bit AG validity process indicator indicating whether the AG is valid for one particular HARQ process or an entire HARQ process. Or the AG includes an 8-bit allowed maximum data rate or power offset and a 1-bit AG validity duration indicator.

In a third example, the AG includes an 8-bit allowed maximum data rate or power offset and a 1-bit AG validity process indicator. Or the AG is configured to include an allowed maximum data rate or power offset, an AG validity duration indicator, an AG validity process indicator, and other E-AGCH control bits in a total of nine bits.

Besides the AG, the E-AGCH carries a UE-ID for identifying a UE and a CRC for detecting errors from the AG. The UE-ID and the CRC each have 16 bits and modulo-2 operated on a bit basis. Thus they are transmitted in the form of a 16-bit CRC masked with the UE-ID. This 16-bit CRC is called a UE-ID specific CRC. The UE can determine whether a received AG is intended for the UE by the UE-ID specific CRC.

Eight tail bits are added to 25-bit control information with the 9-bit AG concatenated to the 16-bit UE-ID specific CRC and convolutionally encoded with a constraints length of 9 at a coding rate of 1/3. As a result, one channel-coded block being a 99-bit coded bit stream is produced. 39 bits are punctured from the 99-bit channel-coded block for transmission in a 2-ms E-AGCH TTI to which an SF of 256 and QPSK apply, creating a 60-bit rate-matched block. A rate matching pattern representing the positions of the punctured bits is simulated in a manner that decreases the change of BER at each bit position of the rate-matched block and thus improves BLER performance. The following rate matching patterns are available.

Rate matching pattern=

{2, 3, 4, 5, 6, 9, 10, 12, 14, 17, 18, 21, 27, 32, 33, 36, 37, 41, 49, 51, 52, 55, 62, 71, 72, 73, 78, 80, 85, 86, 88, 89, 91, 93, 94, 95, 96, 97, 98},

{2, 3, 4, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 30, 31, 35, 42, 44, 46, 51, 53, 55, 60, 62, 64, 69, 71, 83, 85, 86, 89, 90, 91, 93, 95, 96, 97, 98},

{2, 3, 4, 5, 6, 7, 8, 12, 13, 15, 17, 19, 21, 24, 26, 31, 35, 42, 44, 46, 51, 53, 55, 60, 62, 64, 69, 71, 82, 85, 86, 89, 90, 91, 93, 95, 96, 97, 98},

{1, 2, 3, 4, 6, 7, 8, 12, 13, 15, 17, 19, 21, 24, 26, 31, 35, 42, 44, 46, 51, 53, 55, 60, 62, 64, 69, 71, 82, 85, 86, 89, 90, 91, 93, 95, 96, 97, 98},

{1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 15, 17, 18, 19, 21, 24, 42, 47, 54, 56, 58, 66, 68, 71, 72, 73, 75, 84, 86, 87, 89, 90, 92, 93, 95, 96, 98, 99},

{1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 15, 17, 18, 20, 21, 24, 42, 48, 54, 57, 60, 66, 69, 71, 72, 74, 75, 84, 86, 87, 89, 90, 92, 93, 95, 96, 98, 99},

{1, 2, 3, 5, 6, 7, 8, 12, 13, 15, 17, 18, 19, 21, 24, 26, 34, 42, 44, 46, 51, 53, 55, 60, 62, 64, 69, 71, 83, 85, 86, 89, 90, 91, 93, 95, 96, 97, 98},

{1, 2, 3, 4, 5, 6, 7, 8, 12, 14, 15, 18, 21, 24, 30, 33, 39, 42, 48, 54, 57, 60, 63, 66, 69, 72, 75, 84, 87, 89, 90, 92, 93, 94, 95, 96, 97, 98, 99},

{1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 15, 17, 18, 21, 24, 27, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 86, 87, 89, 90, 92, 93, 95, 96, 97, 98, 99}, or

{1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 15, 17, 18, 20, 21, 24, 27, 30, 60, 66, 69, 72, 75, 78, 80, 81, 83, 84, 86, 87, 89, 90, 92, 93, 95, 96, 98, 99}

The elements of each of the rate matching patterns represent the positions of bits to be punctured among the channel-coded bits #1 to #99. Given a 10-ms E-AGCH TTI, the 2-ms E-AGCH TTI occurs five times.

With reference to FIG. **3**, a Node B transmitter for transmitting the E-AGCH according to the fourth exemplary embodiment of the present invention will be described.

Referring to FIG. **3**, upon input of a 9-bit AG **302**, the UE-ID specific CRC adder **304** creates a 16-bit CRC from the AC, generates a UE-ID specific CRC by modulo-2 operating the 16-bit CRC with a 16-bit UE-ID for identifying the UE to receive the AG, and then combines the UE-ID specific CRC with the AG, thereby producing 25-bit control information. The channel encoder **308**, which uses a convolutional code with a constraint length of 9 and a coding rate 1/3, adds eight tail bits to the 25-bit control information and convolutionally encodes the 33-bit information to a 99-bit coded block.

The rate matcher **310** punctures the 99-bit coded block in a predetermined rate matching pattern. The physical channel mapper **312** maps the rate-matched block to a physical channel frame configured to be suitable for 2-ms E-AGCH TTIs. Thus, the control information is transmitted on the E-AGCH **314**.

Meanwhile, the control information transmission controller **316** controls transmission of the control information for an E-DCH through the UE-ID specific CRC adder **304**, the channel encoder **308**, the rate matcher **310**, and the physical channel mapper **312**. The control information transmission controller **316** stores at least one of the above rate matching patterns and applies a preset one of the rate matching patterns to the rate matcher **310**.

With reference to FIG. **4**, a UE receiver for receiving the E-AGCH according to the fourth exemplary embodiment of the present invention will be described.

Referring to FIG. **4**, the UE receives a signal on the E-AGCH **402**. The physical channel demapper **404** extracts a rate-matched block from a 2-ms TTI in the received signal. The rate dematcher **406** recovers (i.e. depunctures) the bits punctured by the rate matcher **310** for the rate-matched block according to the rate matching pattern used in the rate matcher **310** of the Node B. If the E-AGCH **402** uses a 10-ms TTI with five repeated 2-ms TTIs, the physical channel demapper **404** and the rate dematcher **406** perform the same operation five times and combine the resulting coded sub-blocks to one coded block.

The channel decoder **408** decodes the coded block received from the rate dematcher **406**. The coded block is divided into a 9-bit AG and a 16-bit UE-ID specific CRC. The UE-ID specific CRC extractor **410** extracts a 16-bit CRC by modulo-2 operating the 16-bit UE-ID specific CRC with the 16-bit UE-ID **412** of the UE, and provides the extracted CRC and the AG to the CRC checker **414**. The CRC checker **414** checks the 16-bit CRC to detect errors from the AG. If the CRC check passes, the CRC checker **414** outputs the error-free AG **416**. The AG **416** is used for determining an allowed maximum data rate for E-DCH data. If the CRC check fails, the AG is discarded.

Meanwhile, the control information reception controller **418** controls the reception of the control information for the E-DCH through the physical channel demapper **404**, the rate dematcher **406**, the channel decoder **408**, the UE-ID specific CRC extractor **410**, and the CRC checker **414**.

Exemplary Embodiment 5

A rate matching pattern for a 10-bit AG is provided as an example of an implementation of the present invention. For instance, the AG includes an 8-bit allowed maximum data rate representing the maximum amount of uplink radio resources available to a UE or an 8-bit power offset equivalent to the allowed maximum data rate, a 1-bit AG validity duration indicator indicating how long the AG is valid, and a 1-bit AG validity process indicator indicating whether the AG is valid for one particular HARQ process or an entire HARQ process. Or the AG includes a 9-bit allowed maximum data rate or power offset and a 1-bit AG validity duration indicator.

In a third example, the AG includes a 9-bit allowed maximum data rate or power offset and a 1-bit AG validity process indicator. Or the AG is configured to include an allowed maximum data rate or power offset, an AG validity duration indicator, an AG validity process indicator, and other E-AGCH control bits in a total of ten bits.

Besides the AG, the E-AGCH carries a UE-ID for identifying a UE and a CRC for detecting errors from the AG. The UE-ID and the CRC each have 16 bits and modulo-2 operated on a bit basis. Thus they are transmitted in the form of a 16-bit CRC masked with the UE-ID. This 16-bit CRC is called a UE-ID specific CRC. The UE can determine whether a received AG is intended for the UE by the UE-ID specific CRC.

Eight tail bits are added to 26-bit control information with the 10-bit AG concatenated to the 16-bit UE-ID specific CRC and convolutionally encoded with a constraints length of 9 at a coding rate of 1/3. As a result, one channel-coded block being a 102-bit coded bit stream is produced. 42-bits are punctured from the 102-bit channel-coded block for transmission in a 2-ms E-AGCH TTI to which an SF of 256 and QPSK apply, creating a 60-bit rate-matched block. A rate matching pattern representing the positions of the punctured bits is simulated in a manner that decreases the change of BER at each bit position of the rate-matched block and thus improves BLER performance. The following rate matching patterns are available.

Rate matching pattern=

{1, 2, 3, 4, 5, 6, 10, 13, 14, 16, 19, 26, 28, 30, 31, 36, 38, 39, 41, 42, 45, 50, 52, 57, 68, 69, 71, 77, 79, 81, 82, 83, 85, 86, 88, 91, 95, 96, 97, 98, 100, 101},

{1, 3, 5, 6, 7, 9, 10, 12, 13, 15, 16, 17, 20, 21, 30, 32, 34, 42, 43, 44, 50, 52, 54, 55, 57, 61, 75, 78, 79, 82, 84, 87, 88, 90, 92, 93, 94, 97, 98, 99, 101, 102},

{1, 2, 3, 5, 6, 7, 8, 12, 13, 15, 17, 18, 19, 21, 23, 25, 33, 35, 37, 42, 44, 52, 57, 59, 61, 66, 68, 70, 75, 77, 84, 86, 88, 89, 92, 93, 94, 96, 98, 99, 100, 101},

{1, 2, 3, 4, 5, 6, 7, 8, 12, 14, 15, 17, 18, 19, 21, 23, 25, 33, 36, 38, 40, 54, 56, 58, 63, 65, 67, 72, 74, 76, 84, 86, 88, 89, 92, 93, 94, 96, 98, 99, 100, 101},

{1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 15, 17, 18, 20, 21, 24, 27, 36, 39, 42, 54, 57, 60, 63, 66, 69, 72, 75, 78, 84, 87, 89, 90, 92, 93, 95, 96, 98, 99, 101, 102},

{1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 15, 17, 18, 19, 21, 23, 25, 36, 38, 40, 54, 56, 58, 63, 65, 67, 72, 74, 76, 84, 87, 89, 90, 92, 93, 95, 96, 98, 99, 101, 102},

{1, 2, 3, 4, 5, 6, 7, 8, 12, 14, 15, 17, 18, 19, 21, 23, 25, 33, 36, 38, 40, 45, 47, 54, 56, 58, 63, 65, 67, 72, 84, 86, 88, 89, 92, 93, 94, 96, 98, 99, 100, 101},

{1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 15, 18, 21, 24, 33, 36, 39, 42, 48, 54, 57, 60, 66, 69, 72, 75, 78, 84, 87, 89, 90, 92, 93, 95, 96, 97, 98, 99, 100, 101, 102},

{1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 15, 17, 18, 21, 24, 27, 36, 39, 42, 54, 57, 60, 66, 69, 72, 75, 78, 84, 87, 89, 90, 92, 93, 95, 96, 97, 98, 99, 100, 101, 102}, or

{1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 15, 18, 21, 24, 33, 36, 39, 42, 48, 54, 57, 60, 66, 69, 72, 75, 87, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102}

The elements of each of the rate matching patterns represent the positions of bits to be punctured among the channel-coded bits #1 to #102. Given a 10-ms E-AGCH TTI, the 2-ms E-AGCH TTI occurs five times.

With reference to FIG. **3**, a Node B transmitter for transmitting the E-AGCH according to the fifth exemplary embodiment of the present invention will be described.

Referring to FIG. **3**, upon input of a 10-bit AG **302**, the UE-ID specific CRC adder **304** creates a 16-bit CRC from the AC, generates a UE-ID specific CRC by modulo-2 operating the 16-bit CRC with a 16-bit UE-ID for identifying the UE to receive the AG, and then combines the UE-ID specific CRC with the AG, thereby producing 26-bit control information. The channel encoder **308**, which uses a convolutional code with a constraint length of 9 and a coding rate 1/3, adds eight tail bits to the 26-bit control information and convolutionally encodes the 34-bit information to a 102-bit coded block.

The rate matcher **310** punctures the 102-bit coded block in a predetermined rate matching pattern. The physical channel mapper **312** maps the rate-matched block to a physical channel frame configured to be suitable for 2-ms E-AGCH TTIs. Thus, the control information is transmitted on the E-AGCH **314**. Meanwhile, the control information transmission controller **316** controls transmission of the control information for an E-DCH through the UE-ID specific CRC adder **304**, the channel encoder **308**, the rate matcher **310**, and the physical channel mapper **312**. The control information transmission controller **316** stores at least one of the above rate matching patterns and applies a preset one of the rate matching patterns to the rate matcher **310**.

With reference to FIG. **4**, a UE receiver for receiving the E-AGCH according to the fifth exemplary embodiment of the present invention will be described.

Referring to FIG. **4**, the UE receives a signal on the E-AGCH **402**. The physical channel demapper **404** extracts a rate-matched block from a 2-ms TTI in the received signal. The rate dematcher **406** recovers (i.e. depunctures) the bits punctured by the rate matcher **310** for the rate-matched block according to the rate matching pattern used in the rate matcher **310** of the Node B. If the E-AGCH **402** uses a 10-ms TTI with five repeated 2-ms TTIs, the physical channel demapper **404** and the rate dematcher **406** perform the same operation five times and combine the resulting coded sub-blocks to one coded block.

The channel decoder **408** decodes the coded block received from the rate dematcher **406**. The coded block is divided into a 10-bit AG and a 16-bit UE-ID specific CRC. The UE-ID specific CRC extractor **410** extracts a 16-bit CRC by modulo-2 operating the 16-bit UE-ID specific CRC with the 16-bit UE-ID **412** of the UE, and provides the extracted CRC and the AG to the CRC checker **414**. The CRC checker **414** checks the 16-bit CRC to detect errors from the AG. If the CRC check passes, the CRC checker **414** outputs the error-free AG **416**. The AG **416** is used for determining an allowed maximum data rate for E-DCH data. If the CRC check fails, the AG is discarded.

Meanwhile, the control information reception controller **418** controls the reception of the control information for the E-DCH through the physical channel demapper **404**, the rate dematcher **406**, the channel decoder **408**, the UE-ID specific CRC extractor **410**, and the CRC checker **414**.

Exemplary Embodiment 6

A rate matching pattern for a 5-bit AG is provided as an example of an implementation of the present invention. For instance, the AG includes a 4-bit allowed maximum data rate representing the maximum amount of uplink radio resources available to a UE or a 4-bit power offset equivalent to the allowed maximum data rate and a 1-bit AG validity duration indicator indicating how long the AG is valid. Or the AG includes a 4-bit allowed maximum data rate or power offset and a 1-bit AG validity duration indicator indicating whether the AG is valid for one particular HARQ process or an entire HARQ process.

Besides the AG, the E-AGCH carries a UE-ID for identifying a UE and a CRC for detecting errors from the AG. The UE-ID and the CRC each have 16 bits and modulo-2 operated on a bit basis. Thus they are transmitted in the form of a 16-bit CRC masked with the UE-ID. This 16-bit CRC is called a UE-ID specific CRC. The UE can determine whether a received AG is intended for the UE by the UE-ID specific CRC.

Eight tail bits are added to 21-bit control information with the 5-bit AG concatenated to the 16-bit UE-ID specific CRC and convolutionally encoded with a constraints length of 9 at a coding rate of 1/3. As a result, one channel-coded block being an 87-bit coded bit stream is produced. 27 bits are punctured from the 87-bit channel-coded block for transmission in a 2-ms E-AGCH TTI to which an SF of 256 and QPSK apply, creating a 60-bit rate-matched block. A rate matching pattern representing the positions of the punctured bits is simulated in a manner that decreases the change of BER at each bit position of the rate-matched block and thus improves BLER performance. When 2-ms TTIs are used for the E-AGCH, the following rate matching pattern is available.

Rate matching pattern={1, 2, 3, 6, 7, 10, 12, 14, 17, 19, 20, 21, 39, 45, 48, 59, 65, 67, 74, 75, 76, 80, 81, 83, 85, 86, 87}

The elements of the rate matching pattern represent the positions of bits to be punctured among the channel-coded bits #1 to #87. Given a 10-ms E-AGCH TTI, the 2-ms E-AGCH TTI occurs five times.

With reference to FIG. **3**, a Node B transmitter for transmitting the E-AGCH according to the sixth exemplary embodiment of the present invention will be described.

Referring to FIG. **3**, upon input of a 5-bit AG **302**, the UE-ID specific CRC adder **304** creates a 16-bit CRC from the AC, generates a UE-ID specific CRC by modulo-2 operating the 16-bit CRC with a 16-bit UE-ID for identifying the UE to receive the AG, and then combines the UE-ID specific CRC with the AG, thereby producing 21-bit control information. The channel encoder **308**, which uses a convolutional code with a constraint length of 9 and a coding rate 1/3, adds eight tail bits to the 21-bit control information and convolutionally encodes the 29-bit information to an 87-bit coded block.

The rate matcher **310** punctures the 87-bit coded block in a predetermined rate matching pattern. The physical channel mapper **312** maps the rate-matched block to a physical channel frame configured to be suitable for 2-ms E-AGCH TTIs. Thus, the control information is transmitted on the E-AGCH **314**. Meanwhile, the control information transmission controller **316** controls transmission of the control information for an E-DCH through the UE-ID specific CRC adder **304**, the channel encoder **308**, the rate matcher **310**, and the physical channel mapper **312**. The control information transmission controller **316** stores the above rate matching pattern and applies it to the rate matcher **310**.

With reference to FIG. **4**, a UE receiver for receiving the E-AGCH according to the sixth exemplary embodiment of the present invention will be described.

Referring to FIG. **4**, the UE receives a signal on the E-AGCH **402**. The physical channel demapper **404** extracts a rate-matched block from a 2-ms TTI in the received signal. The rate dematcher **406** recovers (i.e. depunctures) the bits punctured by the rate matcher **310** for the rate-matched block according to the rate matching pattern used in the rate matcher **310** of the Node B. If the E-AGCH **402** uses a 10-ms TTI with five repeated 2-ms TTIs, the physical channel demapper **404** and the rate dematcher **406** perform the same operation five times and combine the resulting coded sub-blocks to one coded block.

The channel decoder **408** decodes the coded block received from the rate dematcher **406**. The coded block is divided into a 5-bit AG and a 16-bit UE-ID specific CRC. The UE-ID specific CRC extractor **410** extracts a 16-bit CRC by modulo-2 operating the 16-bit UE-ID specific CRC with the 16-bit UE-ID **412** of the UE, and provides the extracted CRC and the AG to the CRC checker **414**. The CRC checker **414** checks the 16-bit CRC to detect errors from the AG. If the CRC check passes, the CRC checker **414** outputs the error-free AG **416**. The AG **416** is used for determining an allowed maximum data rate for E-DCH data. If the CRC check fails, the AG is discarded.

Meanwhile, the control information reception controller **418** controls the reception of the control information for the E-DCH through the physical channel demapper **404**, the rate dematcher **406**, the channel decoder **408**, the UE-ID specific CRC extractor **410**, and the CRC checker **414**.

The first to sixth exemplary embodiments of the present invention described above use a convolutional code with a constraint length of 9 and a coding rate of 1/3defined in the 3GPP standards as a channel encoding method for the E-AGCH. Below, seventh to eleventh exemplary embodiments of the present invention provide rate matching of the E-AGCH where a convolutional code with a constraint length of 9 and a coding rate of 1/2 defined in the 3GPP standards is used as a channel encoding method for the E-AGCH.

21 to 26-bit control information to be delivered on the E-AGCH is attached with 8 tail bits and then encoded to 58 to 68 coded bits ((21+8)×2=58, (26+8)×2=68) through rate 1/3 convolutional coding.

FIG. **2**B illustrates a convolutional encoder with a constraint length of 9 and a coding rate of 1/2 defined in the 3GPP standards.

Referring to FIG. **2**B, a convolutional encoder **220** includes eight serial shift registers **222**, **224**, **226**, **228**, **230**, **232**, **234** and **236** and a plurality of adders **222***b*, **224***a*, **224***b*, **226***a*, **226***b*, **228***a*, **230***b*, **234***b*, **236***a* and **236***b* for receiving input information bits or the outputs bits of the shift registers **222** to **236**. Input information including eight tail bits being zeroes sequentially pass through the shift registers **222** to **236**, starting from the first bit, and coded bits are produced in an order of output**0**, output**1**, output**2**, output**0**, output**1**, output**2**, ...

The channel-coded control information is delivered in a 2-ms TTI of the E-AGCH. If an SF of 256 and QPSK are applied to the E-AGCH, a total of 60 bits can be transmitted in the 2-ms TTI. Therefore, 2 bits (=58−60) are repeated or up to 8 bits (=68−60) are punctured from the coded control information of the E-AGCH. One thing to note herein is that in the case of a 6-bit AG, control information including the 6-bit AG, a 16-bit UE-ID specific CRC, and 8 tail bits is convolutionally encoded with a constraint length of 9 at a coding rate of 1/2 to 60 channel-coded bits. Since the number of transmittable bits in a 2-ms TTI is 60, there is no need for rate matching in this case.

The following exemplary embodiments of the present invention are intended to provide rate matching patterns which minimize the BLER performance of control information delivered on the E-AGCH. The seventh through eleventh exemplary embodiments of the present invention provide rate matching patterns and corresponding transmission and reception methods for a 5-bit AG, 7-bit AG, 8-bit AG, 9-bit AG and a 10-bit AG, respectively.

Exemplary Embodiment 7

A rate matching pattern for a 5-bit AG is provided as an example of an implementation of the present invention. For instance, the AG includes a 4-bit allowed maximum data rate representing the maximum amount of uplink radio resources available to a UE or a 4-bit power offset equivalent to the allowed maximum data rate and a 1-bit AG validity duration indicator indicating how long the AG is valid. Or the AG includes a 4-bit allowed maximum data rate or power offset and a 1-bit AG validity duration indicator indicating whether the AG is valid for one particular HARQ process or an entire HARQ process.

Besides the AG, the E-AGCH carries a UE-ID for identifying a UE and a CRC for detecting errors from the AG. The UE-ID and the CRC each have 16 bits and modulo-2 operated on a bit basis. Thus they are transmitted in the form of a 16-bit CRC masked with the UE-ID. This 16-bit CRC is called a UE-ID specific CRC. The UE can determine whether a received AG is intended for the UE by the UE-ID specific CRC.

Eight tail bits are added to 21-bit control information with the 5-bit AG concatenated to the 16-bit UE-ID specific CRC and convolutionally encoded with a constraints length of 9 at a coding rate of 1/2. As a result, one channel-coded block being a 58-bit coded bit stream is produced. 2 bits of the 58-bit channel-coded block are repeated to match the size of the channel-coded block to a 2-ms E-AGCH TTI to which an SF of 256 and QPSK apply. Thus, a 60-bit rate-matched block is created. A rate matching pattern representing the positions of the repeated bits is simulated in a manner that decreases the change of BER at each bit position of the rate-matched block and thus improves BLER performance. The following rate matching pattern is available.

Rate matching pattern={23, 57}

The elements of the rate matching pattern represent the positions of bits to be repeated among the channel-coded bits #1 to #58. Given a 10-ms E-AGCH TTI, the 2-ms E-AGCH TTI occurs five times.

With reference to FIG. **3**, a Node B transmitter for transmitting the E-AGCH according to the seventh exemplary embodiment of the present invention will be described.

Referring to FIG. **3**, upon input of a 5-bit AG **302**, the UE-ID specific CRC adder **304** creates a 16-bit CRC from the AC, generates a UE-ID specific CRC by modulo-2 operating the 16-bit CRC with a 16-bit UE-ID for identifying the UE to receive the AG, and then combines the UE-ID specific CRC with the AG, thereby producing 21-bit control information. The channel encoder **308**, which uses a convolutional code with a constraint length of 9 and a coding rate 1/2, adds eight tail bits to the 21-bit control information and convolutionally encodes the 29-bit information to a 58-bit coded block.

The rate matcher **310** repeats the 58-bit coded block in a predetermined rate matching pattern. The physical channel mapper **312** maps the rate-matched block to a physical channel frame configured to be suitable for 2-ms E-AGCH TTIs. Thus, the control information is transmitted on the E-AGCH **314**. Meanwhile, the control information transmission controller **316** controls transmission of the control information for an E-DCH through the UE-ID specific CRC adder **304**, the channel encoder **308**, the rate matcher **310**, and the physical channel mapper **312**. The control information transmission controller **316** stores the above rate matching pattern and applies it to the rate matcher **310**.

With reference to FIG. **4**, a UE receiver for receiving the E-AGCH according to the seventh exemplary embodiment of the present invention will be described.

Referring to FIG. **4**, the UE receives a signal on the E-AGCH **402**. The physical channel demapper **404** extracts a rate-matched block from a 2-ms TTI in the received signal. The rate dematcher **406** recovers (i.e. combines) the bits repeated by the rate matcher **310** for the rate-matched block according to the rate matching pattern used in the rate matcher **310** of the Node B. If the E-AGCH **402** uses a 10-ms TTI with five repeated 2-ms TTIs, the physical channel demapper **404** and the rate dematcher **406** perform the same operation five times and combine the resulting coded sub-blocks to one coded block.

The channel decoder **408** decodes the coded block received from the rate dematcher **406**. The coded block is divided into a 5-bit AG and a 16-bit UE-ID specific CRC. The UE-ID specific CRC extractor **410** extracts a 16-bit CRC by modulo-2 operating the 16-bit UE-ID specific CRC with the 16-bit UE-ID **412** of the UE, and provides the extracted CRC and the AG to the CRC checker **414**. The CRC checker **414** checks the 16-bit CRC to detect errors from the AG. If the CRC check passes, the CRC checker **414** outputs the error-free AG **416**. The AG **416** is used for determining an allowed maximum data rate for E-DCH data. If the CRC check fails, the AG is discarded.

Meanwhile, the control information reception controller **418** controls the reception of the control information for the E-DCH through the physical channel demapper **404**, the rate dematcher **406**, the channel decoder **408**, the UE-ID specific CRC extractor **410**, and the CRC checker **414**.

Exemplary Embodiment 8

A rate matching pattern for a 7-bit AG is provided as an example of an implementation of the present invention. For instance, the AG includes a 5-bit allowed maximum data rate representing the maximum amount of uplink radio resources available to a UE or a 5-bit power offset equivalent to the allowed maximum data rate, a 1-bit AG validity duration indicator indicating how long the AG is valid, and a 1-bit AG validity process indicator indicating whether the AG is valid for one particular HARQ process or an entire HARQ process. Or the AG includes a 6-bit allowed maximum data rate or power offset and a 1-bit AG validity duration indicator.

In a third example, the AG includes a 6-bit allowed maximum data rate or power offset and a 1-bit AG validity process indicator. Or the AG is configured to include an allowed maximum data rate or power offset, an AG validity duration indicator, an AG validity process indicator, and other E-AGCH control bits in a total of seven bits.

Besides the AG, the E-AGCH carries a UE-ID for identifying a UE and a CRC for detecting errors from the AG. The UE-ID and the CRC each have 16 bits and modulo-2 operated on a bit basis. Thus they are transmitted in the form of a 16-bit CRC masked with the UE-ID. This 16-bit CRC is called a UE-ID specific CRC. The UE can determine whether a received AG is intended for the UE by the UE-ID specific CRC.

Eight tail bits are added to 23-bit control information with the 7-bit AG concatenated to the 16-bit UE-ID specific CRC and convolutionally encoded with a constraints length of 9 at a coding rate of 1/2. As a result, one channel-coded block being a 62-bit coded bit stream is produced. 2 bits are punctured from the 62-bit channel-coded block for transmission in a 2-ms E-AGCH TTI to which an SF of 256 and QPSK apply, creating a 60-bit rate-matched block. A rate matching pattern representing the positions of the punctured bits is simulated in a manner that decreases the change of BER at each bit position of the rate-matched block and thus improves BLER performance. The following rate matching pattern is available.

Rate matching pattern={2, 62}

The elements of the rate matching pattern represent the positions of bits to be punctured among the channel-coded bits #1 to #62. Given a 10-ms E-AGCH TTI, the 2-ms E-AGCH TTI occurs five times.

With reference to FIG. **3**, a Node B transmitter for transmitting the E-AGCH according to the eighth exemplary embodiment of the present invention will be described.

Referring to FIG. **3**, upon input of a 7-bit AG **302**, the UE-ID specific CRC adder **304** creates a 16-bit CRC from the AC, generates a UE-ID specific CRC by modulo-2 operating the 16-bit CRC with a 16-bit UE-ID for identifying the UE to receive the AG, and then combines the UE-ID specific CRC with the AG, thereby producing 23-bit control information. The channel encoder **308**, which uses a convolutional code with a constraint length of 9 and a coding rate 1/2, adds eight tail bits to the 23-bit control information and convolutionally encodes the 31-bit information to a 62-bit coded block.

The rate matcher **310** punctures the 62-bit coded block in a predetermined rate matching pattern. The physical channel mapper **312** maps the rate-matched block to a physical channel frame configured to be suitable for 2-ms E-AGCH TTIs. Thus, the control information is transmitted on the E-AGCH **314**. Meanwhile, the control information transmission controller **316** controls transmission of the control information for an E-DCH through the UE-ID specific CRC adder **304**, the channel encoder **308**, the rate matcher **310**, and the physical channel mapper **312**. The control information transmission controller **316** stores the above rate matching pattern and applies it to the rate matcher **310**.

With reference to FIG. **4**, a UE receiver for receiving the E-AGCH according to the eighth exemplary embodiment of the present invention will be described.

Referring to FIG. **4**, the UE receives a signal on the E-AGCH **402**. The physical channel demapper **404** extracts a rate-matched block from a 2-ms TTI in the received signal. The rate dematcher **406** recovers (i.e. depunctures) the bits punctured by the rate matcher **310** for the rate-matched block according to the rate matching pattern used in the rate matcher **310** of the Node B. If the E-AGCH **402** uses a 10-ms TTI with five repeated 2-ms TTIs, the physical channel demapper **404** and the rate dematcher **406** perform the same operation five times and combine the resulting coded sub-blocks to one coded block.

The channel decoder **408** decodes the coded block received from the rate dematcher **406**. The coded block is divided into a 7-bit AG and a 16-bit UE-ID specific CRC. The UE-ID specific CRC extractor **410** extracts a 16-bit CRC by modulo-2 operating the 16-bit UE-ID specific CRC with the 16-bit UE-ID **412** of the UE, and provides the extracted CRC and the AG to the CRC checker **414**. The CRC checker **414** checks the 16-bit CRC to detect errors from the AG. If the CRC check passes, the CRC checker **414** outputs the error-free AG **416**. The AG **416** is used for determining an allowed maximum data rate for E-DCH data. If the CRC check fails, the AG is discarded.

Meanwhile, the control information reception controller **418** controls the reception of the control information for the E-DCH through the physical channel demapper **404**, the rate dematcher **406**, the channel decoder **408**, the UE-ID specific CRC extractor **410**, and the CRC checker **414**.

Exemplary Embodiment 9

A rate matching pattern for an 8-bit AG is provided as an example of an implementation of the present invention. For instance, the AG includes a 6-bit allowed maximum data rate representing the maximum amount of uplink radio resources available to a UE or a 6-bit power offset equivalent to the allowed maximum data rate, a 1-bit AG validity duration indicator indicating how long the AG is valid, and a 1-bit AG validity process indicator indicating whether the AG is valid for one particular HARQ process or an entire HARQ process. Or the AG includes a 7-bit allowed maximum data rate or power offset and a 1-bit AG validity duration indicator.

In a third example, the AG includes a 7-bit allowed maximum data rate or power offset and a 1-bit AG validity process indicator. Or the AG is configured to include an allowed maximum data rate or power offset, an AG validity duration indicator, an AG validity process indicator, and other E-AGCH control bits in a total of eight bits.

Besides the AG, the E-AGCH carries a UE-ID for identifying a UE and a CRC for detecting errors from the AG. The UE-ID and the CRC each have 16 bits and modulo-2 operated on a bit basis. Thus they are transmitted in the form of a 16-bit CRC masked with the UE-ID. This 16-bit CRC is called a UE-ID specific CRC. The UE can determine whether a received AG is intended for the UE by the UE-ID specific CRC.

Eight tail bits are added to 24-bit control information with the 8-bit AG concatenated to the 16-bit UE-ID specific CRC and convolutionally encoded with a constraints length of 9 at a coding rate of 1/2. As a result, one channel-coded block being a 64-bit coded bit stream is produced. 4 bits are punctured from the 64-bit channel-coded block for transmission in a 2-ms E-AGCH TTI to which an SF of 256 and QPSK apply, creating a 60-bit rate-matched block. A rate matching pattern representing the positions of the punctured bits is simulated in a manner that decreases the change of BER at each bit position of the rate-matched block and thus improves BLER performance. The following rate matching patterns are available.

Rate matching pattern={2, 10, 60, 63} or {2, 6, 60, 63}

The elements of each of the rate matching patterns represent the positions of bits to be punctured among the channel-coded bits #1 to #64. Given a 10-ms E-AGCH TTI, the 2-ms E-AGCH TTI occurs five times.

With reference to FIG. **3**, a Node B transmitter for transmitting the E-AGCH according to the ninth exemplary embodiment of the present invention will be described.

Referring to FIG. **3**, upon input of an 8-bit AG **302**, the UE-ID specific CRC adder **304** creates a 16-bit CRC from the AC, generates a UE-ID specific CRC by modulo-2 operating the 16-bit CRC with a 16-bit UE-ID for identifying the UE to receive the AG, and then combines the UE-ID specific CRC with the AG, thereby producing 24-bit control information. The channel encoder **308**, which uses a convolutional code with a constraint length of 9 and a coding rate 1/2, adds eight tail bits to the 24-bit control information and convolutionally encodes the 32-bit information to a 64-bit coded block.

The rate matcher **310** punctures the 64-bit coded block in a predetermined rate matching pattern. The physical channel mapper **312** maps the rate-matched block to a physical channel frame configured to be suitable for 2-ms E-AGCH TTIs. Thus, the control information is transmitted on the E-AGCH **314**. Meanwhile, the control information transmission controller **316** controls transmission of the control information for an E-DCH through the UE-ID specific CRC adder **304**, the channel encoder **308**, the rate matcher **310**, and the physical channel mapper **312**. The control information transmission controller **316** stores at least one of the above rate matching patterns and applies a preset one of the rate matching patterns to the rate matcher **310**.

With reference to FIG. **4**, a UE receiver for receiving the E-AGCH according to the ninth exemplary embodiment of the present invention will be described.

Referring to FIG. **4**, the UE receives a signal on the E-AGCH **402**. The physical channel demapper **404** extracts a rate-matched block from a 2-ms TTI in the received signal. The rate dematcher **406** recovers (i.e. depunctures) the bits punctured by the rate matcher **310** for the rate-matched block according to the rate matching pattern used in the rate matcher **310** of the Node B. If the E-AGCH **402** uses a 10-ms TTI with five repeated 2-ms TTIs, the physical channel demapper **404** and the rate dematcher **406** perform the same operation five times and combine the resulting coded sub-blocks to one coded block.

The channel decoder **408** decodes the coded block received from the rate dematcher **406**. The coded block is divided into an 8-bit AG and a 16-bit UE-ID specific CRC. The UE-ID specific CRC extractor **410** extracts a 16-bit CRC by modulo-2 operating the 16-bit UE-ID specific CRC with the 16-bit UE-ID **412** of the UE, and provides the extracted CRC and the AG to the CRC checker **414**. The CRC checker **414** checks the 16-bit CRC to detect errors from the AG. If the CRC check passes, the CRC checker **414** outputs the error-free AG **416**. The AG **416** is used for determining an allowed maximum data rate for E-DCH data. If the CRC check fails, the AG is discarded.

Meanwhile, the control information reception controller **418** controls the reception of the control information for the E-DCH through the physical channel demapper **404**, the rate dematcher **406**, the channel decoder **408**, the UE-ID specific CRC extractor **410**, and the CRC checker **414**.

Exemplary Embodiment 10

A rate matching pattern for a 9-bit AG is provided as an example of an implementation of the present invention. For instance, the AG includes a 7-bit allowed maximum data rate representing the maximum amount of uplink radio resources available to a UE or a 7-bit power offset equivalent to the allowed maximum data rate, a 1-bit AG validity duration indicator indicating how long the AG is valid, and a 1-bit AG validity process indicator indicating whether the AG is valid for one particular HARQ process or an entire HARQ process. Or the AG includes an 8-bit allowed maximum data rate or power offset and a 1-bit AG validity duration indicator.

In a third example, the AG includes an 8-bit allowed maximum data rate or power offset and a 1-bit AG validity process indicator. Or the AG is configured to include an allowed maximum data rate or power offset, an AG validity duration indicator, an AG validity process indicator, and other E-AGCH control bits in a total of nine bits.

Besides the AG, the E-AGCH carries a UE-ID for identifying a UE and a CRC for detecting errors from the AG. The UE-ID and the CRC each have 16 bits and modulo-2 operated on a bit basis. Thus they are transmitted in the form of a 16-bit CRC masked with the UE-ID. This 16-bit CRC is called a UE-ID specific CRC. The UE can determine whether a received AG is intended for the UE by the UE-ID specific CRC.

Eight tail bits are added to 25-bit control information with the 9-bit AG concatenated to the 16-bit UE-ID specific CRC and convolutionally encoded with a constraints length of 9 at a coding rate of 1/2. As a result, one channel-coded block being a 66-bit coded bit stream is produced. 6 bits are punctured from the 66-bit channel-coded block for transmission in a 2-ms E-AGCH TTI to which an SF of 256 and QPSK apply, creating a 60-bit rate-matched block. A rate matching pattern representing the positions of the punctured bits is simulated in a manner that decreases the change of BER at each bit position of the rate-matched block and thus improves BLER performance. The following rate matching patterns are available.

Rate matching pattern={1, 3, 7, 59, 63, 66} or {1, 4, 10, 59, 63, 66}

The elements of each of the rate matching patterns represent the positions of bits to be punctured among the channel-coded bits #1 to #66. Given a 10-ms E-AGCH TTI, the 2-ms E-AGCH TTI occurs five times.

With reference to FIG. **3**, a Node B transmitter for transmitting the E-AGCH according to the tenth exemplary embodiment of the present invention will be described.

Referring to FIG. **3**, upon input of a 9-bit AG **302**, the UE-ID specific CRC adder **304** creates a 16-bit CRC from the AC, generates a UE-ID specific CRC by modulo-2 operating the 16-bit CRC with a 16-bit UE-ID for identifying the UE to receive the AG, and then combines the UE-ID specific CRC with the AG, thereby producing 25-bit control information.

The channel encoder **308**, which uses a convolutional code with a constraint length of 9 and a coding rate 1/2, adds eight tail bits to the 25-bit control information and convolutionally encodes the 33-bit information to a 66-bit coded block. The rate matcher **310** punctures the 66-bit coded block in a predetermined rate matching pattern. The physical channel mapper **312** maps the rate-matched block to a physical channel frame configured to be suitable for 2-ms E-AGCH TTIs. Thus, the control information is transmitted on the E-AGCH **314**.

Meanwhile, the control information transmission controller **316** controls transmission of the control information for an E-DCH through the UE-ID specific CRC adder **304**, the channel encoder **308**, the rate matcher **310**, and the physical channel mapper **312**. The control information transmission controller **316** stores at least one of the above rate matching patterns and applies a preset one of the rate matching patterns to the rate matcher **310**.

With reference to FIG. **4**, a UE receiver for receiving the E-AGCH according to the tenth exemplary embodiment of the present invention will be described.

Referring to FIG. **4**, the UE receives a signal on the E-AGCH **402**. The physical channel demapper **404** extracts a rate-matched block from a 2-ms TTI in the received signal. The rate dematcher **406** recovers (i.e. depunctures) the bits punctured by the rate matcher **310** for the rate-matched block according to the rate matching pattern used in the rate matcher **310** of the Node B. If the E-AGCH **402** uses a 10-ms TTI with five repeated 2-ms TTIs, the physical channel demapper **404** and the rate dematcher **406** perform the same operation five times and combine the resulting coded sub-blocks to one coded block.

The channel decoder **408** decodes the coded block received from the rate dematcher **406**. The coded block is divided into a 9-bit AG and a 16-bit UE-ID specific CRC. The UE-ID specific CRC extractor **410** extracts a 16-bit CRC by modulo-2 operating the 16-bit UE-ID specific CRC with the 16-bit UE-ID **412** of the UE, and provides the extracted CRC and the AG to the CRC checker **414**. The CRC checker **414** checks the 16-bit CRC to detect errors from the AG. If the CRC check passes, the CRC checker **414** outputs the error-free AG **416**. The AG **416** is used for determining an allowed maximum data rate for E-DCH data. If the CRC check fails, the AG is discarded.

Meanwhile, the control information reception controller **418** controls the reception of the control information for the E-DCH through the physical channel demapper **404**, the rate dematcher **406**, the channel decoder **408**, the UE-ID specific CRC extractor **410**, and the CRC checker **414**.

Exemplary Embodiment 11

A rate matching pattern for a 10-bit AG is provided as an example of an implementation of the present invention. For instance, the AG includes an 8-bit allowed maximum data rate representing the maximum amount of uplink radio resources available to a UE or an 8-bit power offset equivalent to the allowed maximum data rate, a 1-bit AG validity duration indicator indicating how long the AG is valid, and a 1-bit AG validity process indicator indicating whether the AG is valid for one particular HARQ process or an entire HARQ process. Or the AG includes a 9-bit allowed maximum data rate or power offset and a 1-bit AG validity duration indicator.

In a third example, the AG includes a 9-bit allowed maximum data rate or power offset and a 1-bit AG validity process indicator. Or the AG is configured to include an allowed maximum data rate or power offset, an AG validity duration indicator, an AG validity process indicator, and other E-AGCH control bits in a total of ten bits.

Besides the AG, the E-AGCH carries a UE-ID for identifying a UE and a CRC for detecting errors from the AG. The UE-ID and the CRC each have 16 bits and modulo-2 operated on a bit basis. Thus they are transmitted in the form of a 16-bit CRC masked with the UE-ID. This 16-bit CRC is called a UE-ID specific CRC. The UE can determine whether a received AG is intended for the UE by the UE-ID specific CRC.

Eight tail bits are added to 26-bit control information with the 10-bit AG concatenated to the 16-bit UE-ID specific CRC and convolutionally encoded with a constraints length of 9 at a coding rate of 1/2. As a result, one channel-coded block being a 68-bit coded bit stream is produced. 8 bits are punctured from the 68-bit channel-coded block for transmission in a 2-ms E-AGCH TTI to which an SF of 256 and QPSK apply, creating a 60-bit rate-matched block. A rate matching pattern representing the positions of the punctured bits is simulated in a manner that decreases the change of BER at each bit position of the rate-matched block and thus improves BLER performance. The following rate matching patterns are available.

Rate matching pattern={1, 2, 3, 8, 49, 65, 67, 68} or {2, 5, 6, 10, 54, 59, 63, 68}

The elements of each of the rate matching patterns represent the positions of bits to be punctured among the channel-coded bits #1 to #68. Given a 10-ms E-AGCH TTI, the 2-ms E-AGCH TTI occurs five times.

With reference to FIG. **3**, a Node B transmitter for transmitting the E-AGCH according to the eleventh exemplary embodiment of the present invention will be described.

Referring to FIG. **3**, upon input of a 10-bit AG **302**, the UE-ID specific CRC adder **304** creates a 16-bit CRC from the AC, generates a UE-ID specific CRC by modulo-2 operating the 16-bit CRC with a 16-bit UE-ID for identifying the UE to receive the AG, and then combines the UE-ID specific CRC with the AG, thereby producing 26-bit control information. The channel encoder **308**, which uses a convolutional code with a constraint length of 9 and a coding rate 1/2, adds eight tail bits to the 26-bit control information and convolutionally encodes the 34-bit information to a 68-bit coded block.

The rate matcher **310** punctures the 68-bit coded block in a predetermined rate matching pattern. The physical channel mapper **312** maps the rate-matched block to a physical channel frame configured to be suitable for 2-ms E-AGCH TTIs. Thus, the control information is transmitted on the E-AGCH **314**. Meanwhile, the control information transmission controller **316** controls transmission of the control information for an E-DCH through the UE-ID specific CRC adder **304**, the channel encoder **308**, the rate matcher **310**, and the physical channel mapper **312**. The control information transmission controller **316** stores at least one of the above rate matching patterns and applies a preset one of the rate matching patterns to the rate matcher **310**.

With reference to FIG. **4**, a UE receiver for receiving the E-AGCH according to the eleventh exemplary embodiment of the present invention will be described.

Referring to FIG. **4**, the UE receives a signal on the E-AGCH **402**. The physical channel demapper **404** extracts a rate-matched block from a 2-ms TTI in the received signal. The rate dematcher **406** recovers (that is, depunctures) the bits punctured by the rate matcher **310** for the rate-matched block according to the rate matching pattern used in the rate matcher **310** of the Node B. If the E-AGCH **402** uses a 10-ms TTI with five repeated 2-ms TTIs, the physical channel demapper **404** and the rate dematcher **406** perform the same operation five times and combine the resulting coded sub-blocks to one coded block.

The channel decoder **408** decodes the coded block received from the rate dematcher **406**. The coded block is divided into a 10-bit AG and a 16-bit UE-ID specific CRC. The UE-ID specific CRC extractor **410** extracts a 16-bit CRC by modulo-2 operating the 16-bit UE-ID specific CRC with the 16-bit UE-ID **412** of the UE, and provides the extracted CRC and the AG to the CRC checker **414**. The CRC checker **414** checks the 16-bit CRC to detect errors from the AG. If the CRC check passes, the CRC checker **414** outputs the error-free AG **416**. The AG **416** is used for determining an allowed maximum data rate for E-DCH data. If the CRC check fails, the AG is discarded.

Meanwhile, the control information reception controller **418** controls the reception of the control information for the E-DCH through the physical channel demapper **404**, the rate dematcher **406**, the channel decoder **408**, the UE-ID specific CRC extractor **410**, and the CRC checker **414**.

As described above, certain exemplary embodiments of the present invention may increase the transmission reliability of an AG indicating the absolute value of an allowed maximum data rate for a UE by providing a rate matching pattern that reduces a BER variation at each bit position within a block and thus may improve BLER performance. In addition, lower power consumption may be required for the same BLER performance and as a result, uplink interference may be reduced.

While the invention has been shown and described with reference to certain exemplary embodiments thereof, it will be understood by those of ordinary skill in the art that various changes in form and details may be made therein without departing from the spirit and scope of the invention as defined by the appended claims.

What is claimed is:

1. A method of transmitting control information associated with uplink packet data transmission in a mobile communication system, the method comprising the steps of:

generating a 16-bit user equipment identifier (UE-ID) specific cyclic redundancy check (CRC) by combining a CRC generated for detecting errors from the control information with a UE-ID for identifying a user equipment (UE) to receive the control information;

generating 90 coded bits by adding the UE-ID specific CRC and 8 tails bits to 6-bit control information and encoding the added bits at a coding rate of 1/3;

generating a 60-bit rate-matched block by rate-matching the coded bits according to a rate matching pattern representing positions of bits to be punctured among the coded bits; and

transmitting the rate-matched block to the UE,

wherein the rate matching pattern comprises {1, 2, 5, 6, 7, 11, 12, 14, 15, 17, 23, 24, 31, 37, 44, 47, 61, 63, 64, 71, 72, 75, 77, 80, 83, 84, 85, 87, 88, 90}.

2. The method of claim **1**, wherein the control information comprises an indication of an allowed maximum data rate for transmission of uplink packet data from the UE.

3. The method of claim **2**, wherein the control information comprises a 5-bit power offset equivalent to the allowed maximum data rate and a 1-bit validity process indicator indicating whether the control information is valid for an entire hybrid automatic repeat request (HARQ) process.

4. The method of claim **1**, wherein the UE-ID specific CRC generating step comprises the step of generating the UE-ID specific CRC by modulo-2 operating a 16-bit CRC with a 16-bit UE ID.

5. An apparatus of transmitting control information associated with uplink packet data transmission in a mobile communication system, the apparatus comprising:

a user equipment identifier (UE-ID) specific cyclic redundancy check (CRC) generator for generating a 16-bit UE-ID specific CRC by combining a CRC generated for detecting errors from the control information, with a UE-ID for identifying a UE to receive the control information;

a channel encoder for generating 90 coded bits by adding the UE-ID specific CRC and 8 tails bits to 6-bit control information and encoding the added bits at a coding rate of 1/3;

a rate matcher for generating a 60-bit rate-matched block by rate-matching the coded bits, according to a rate matching pattern, representing positions of bits to be punctured among the coded bits; and

a physical channel mapper for transmitting the rate-matched block to the UE,

wherein the rate matching pattern comprises {1, 2, 5, 6, 7, 11, 12, 14, 15, 17, 23, 24, 31, 37, 44, 47, 61, 63, 64, 71, 72, 75, 77, 80, 83, 84, 85, 87, 88, 90}.

6. The apparatus of claim **5**, wherein the control information comprises an indication of an allowed maximum data rate for transmission of uplink packet data from the UE.

7. The apparatus of claim **6**, wherein the control information comprises a 5-bit power offset equivalent to the allowed maximum data rate and a 1-bit validity process indicator indicating whether the control information is valid for an entire hybrid automatic repeat request (HARQ) process.

8. The apparatus of claim **5**, wherein the UE-ID specific CRC is generated by modulo-2 operating a 16-bit CRC with a 16-bit UE ID.

9. A method of receiving control information associated with uplink packet data transmission in a mobile communication system, comprising the steps of:

extracting a 60-bit rate-matched block from a signal received from a Node B;

generating 90 coded bits by rate-dematching the rate-matched block according to a rate matching pattern representing positions of bits to be depunctured;

generating 6-bit control information and a 16-bit user equipment identifier (UE-ID) specific cyclic redundancy check (CRC) by decoding the coded bits at a coding rate of 1/3; and

outputting the control information by checking the UE-ID specific CRC,

wherein the rate matching pattern comprises {1, 2, 5, 6, 7, 11, 12, 14, 15, 17, 23, 24, 31, 37, 44, 47, 61, 63, 64, 71, 72, 75, 77, 80, 83, 84, 85, 87, 88, 90}.

10. The method of claim **9**, wherein the control information comprises an indication of an allowed maximum data rate for transmission of uplink packet data.

11. The method of claim **10**, wherein the control information comprises a 5-bit power offset equivalent to the allowed maximum data rate and a 1-bit validity process indicator indicating whether the control information is valid for an entire hybrid automatic repeat request (HARQ) process.

12. The method of claim **9**, wherein the UE-ID specific CRC is generated by modulo-2 operating a 16-bit CRC with a 16-bit UE ID.

13. An apparatus for receiving control information associated with uplink packet data transmission in a mobile communication system, the apparatus comprising:

a physical channel demapper for extracting a 60-bit rate-matched block from a signal received from a Node B;

a rate dematcher for generating 90 coded bits by rate-dematching the rate-matched block according to a rate matching pattern representing positions of bits to be depunctured;

a channel decoder for generating 6-bit control information and a 16-bit user equipment identifier (UE-ID) specific cyclic redundancy check (CRC) by decoding the coded bits at a coding rate of 1/3; and

a CRC checker for outputting the control information by checking the UE-ID specific CRC,

wherein the rate matching pattern comprises {1, 2, 5, 6, 7, 11, 12, 14, 15, 17, 23, 24, 31, 37, 44, 47, 61, 63, 64, 71, 72, 75, 77, 80, 83, 84, 85, 87, 88, 90}.

14. The apparatus of claim **13**, wherein the control information comprises an indication of an allowed maximum data rate for transmission of uplink packet data.

15. The apparatus of claim **14**, wherein the control information comprises a 5-bit power offset equivalent to the allowed maximum data rate and a 1-bit validity process indicator indicating whether the control information is valid for an entire hybrid automatic repeat request (HARQ) process.

16. The apparatus of claim **13**, wherein the UE-ID specific CRC is generated by modulo-2 operating a 16-bit CRC with a 16-bit UE ID.

* * * * *

Exhibit 3

US006771980B2

(12) **United States Patent**
Moon

(10) **Patent No.: US 6,771,980 B2**
(45) **Date of Patent: Aug. 3, 2004**

(54) **METHOD FOR DIALING IN A SMART PHONE**

(75) Inventor: **Jeong-Kyu Moon**, Yongin-shi (KR)

(73) Assignee: **Samsung Electronics Co., Ltd.** (KR)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 528 days.

(21) Appl. No.: **09/728,814**

(22) Filed: **Dec. 1, 2000**

(65) **Prior Publication Data**

US 2001/0002926 A1 Jun. 7, 2001

(30) **Foreign Application Priority Data**

Dec. 2, 1999 (KR) 1999-54380

(51) **Int. Cl.**[7] **H04B 1/38**

(52) **U.S. Cl.** **455/553.1**; 455/90.1; 455/556.2; 345/173

(58) **Field of Search** 455/90, 556.2, 455/575, 418, 566, 403, 550, 553.1, 66, 556.1, 457, 552.1, 90.1, 90.2, 90.3; 379/93.1, 93.05, 93.2; 345/32, 173

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,572,573 A * 11/1996 Sylvan et al. 455/556.2
5,625,673 A * 4/1997 Grewe et al. 455/556.2
5,797,089 A * 8/1998 Nguyen 455/403
5,826,198 A * 10/1998 Bergins et al. 455/557
6,415,138 B2 * 7/2002 Sirola et al. 455/90.1
6,418,309 B1 * 7/2002 Moon et al. 455/418
6,453,173 B1 * 9/2002 Reber et al. 455/557
6,487,396 B1 * 11/2002 Sassi 455/90.1

* cited by examiner

Primary Examiner—Nick Corsaro
Assistant Examiner—Tan Trinh
(74) *Attorney, Agent, or Firm*—Dilworth & Barrese, LLP.

(57) **ABSTRACT**

A method for dialing a phone number in a smart phone having both PDA (Personal Digital Assistant) and mobile phone functions during operation of a PDA function, comprising the steps of loading the OS (Operating System) program for dialing the phone number during operation of the PDA function, selecting a phone number, displaying a dialing icon according to an embodiment of the received signal strength indicator (RSSI), and dialing the selected phone number by selecting the dialing icon in response to the user's request of dialing.

13 Claims, 4 Drawing Sheets

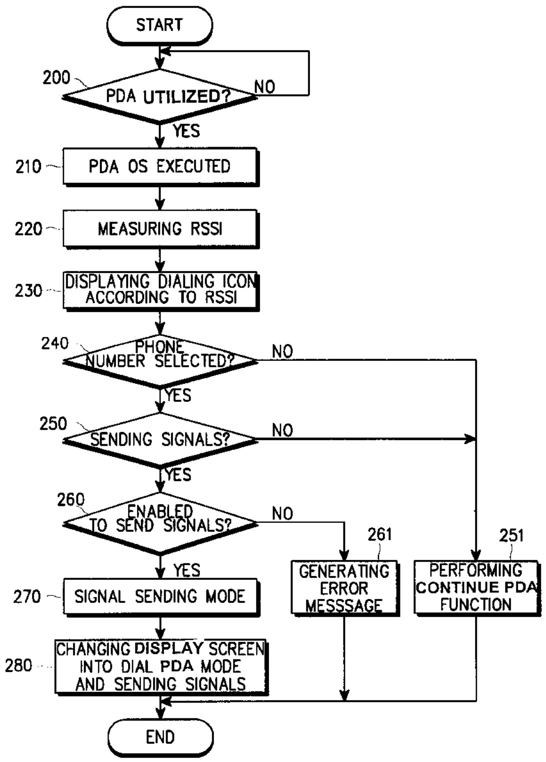
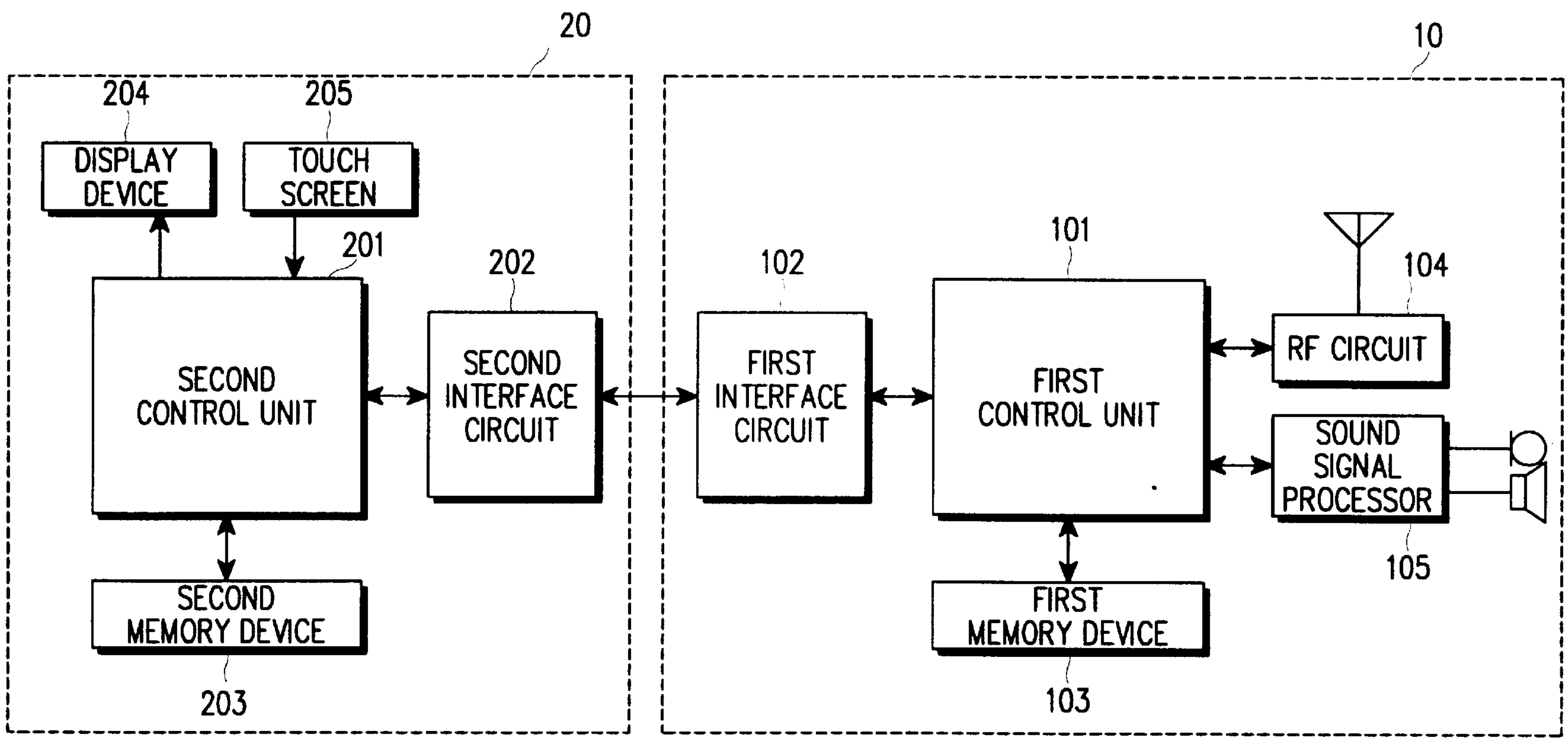


FIG. 1


20
10
204
205
DISPLAY DEVICE
TOUCH SCREEN
201
202
102
101
104
SECOND CONTROL UNIT
SECOND INTERFACE CIRCUIT
FIRST INTERFACE CIRCUIT
FIRST CONTROL UNIT
RF CIRCUIT
SOUND SIGNAL PROCESSOR
105
SECOND MEMORY DEVICE
FIRST MEMORY DEVICE
203
103

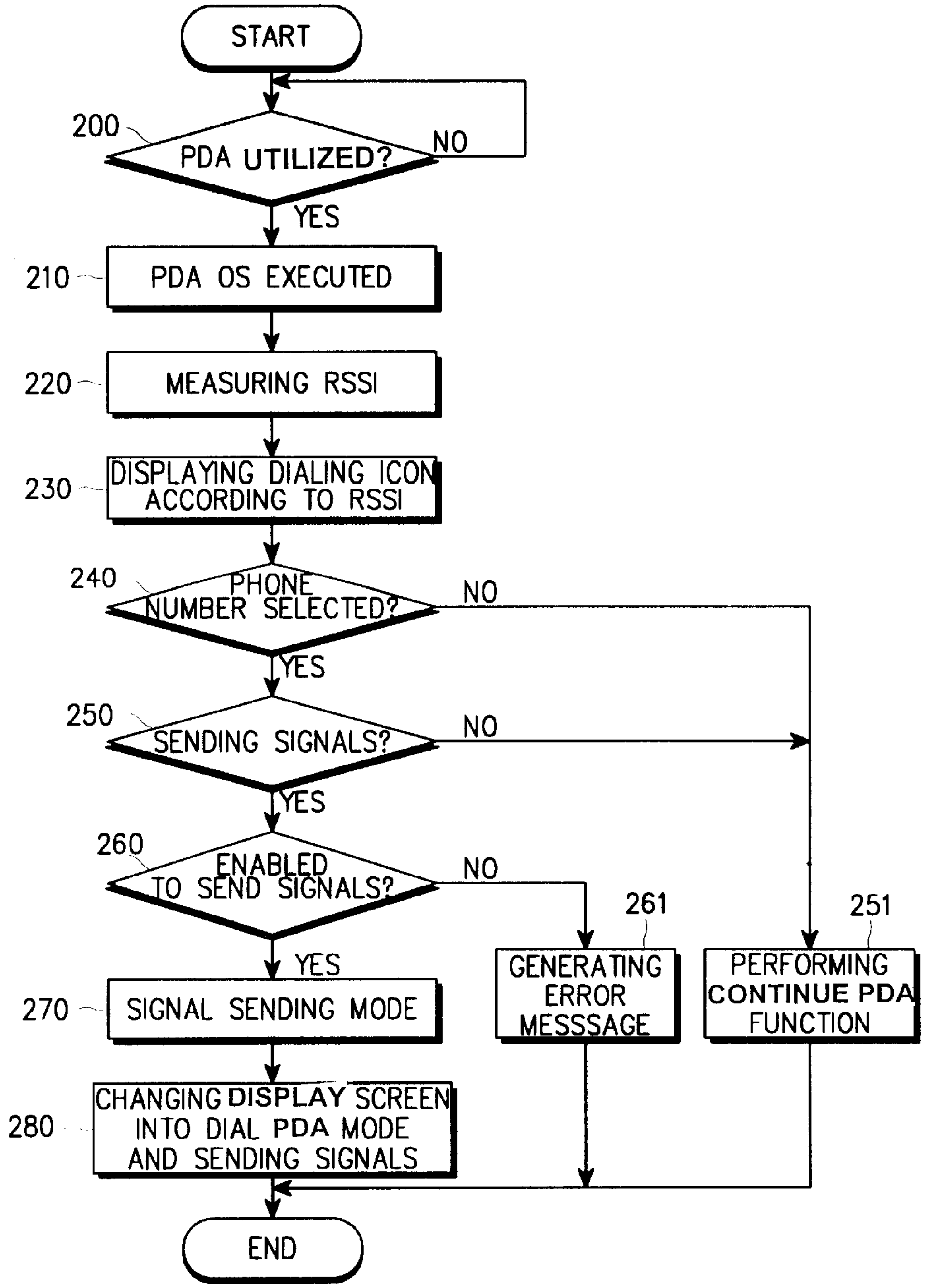

START
200
PDA UTILIZED?
NO
YES
210
PDA OS EXECUTED
220
MEASURING RSSI
230
DISPLAYING DIALING ICON ACCORDING TO RSSI
240
PHONE NUMBER SELECTED?
NO
YES
250
SENDING SIGNALS?
NO
YES
260
ENABLED TO SEND SIGNALS?
NO
261
251
YES
270
SIGNAL SENDING MODE
GENERATING ERROR MESSSAGE
PERFORMING CONTINUE PDA FUNCTION
280
CHANGING DISPLAY SCREEN INTO DIAL PDA MODE AND SENDING SIGNALS
END

FIG. 2

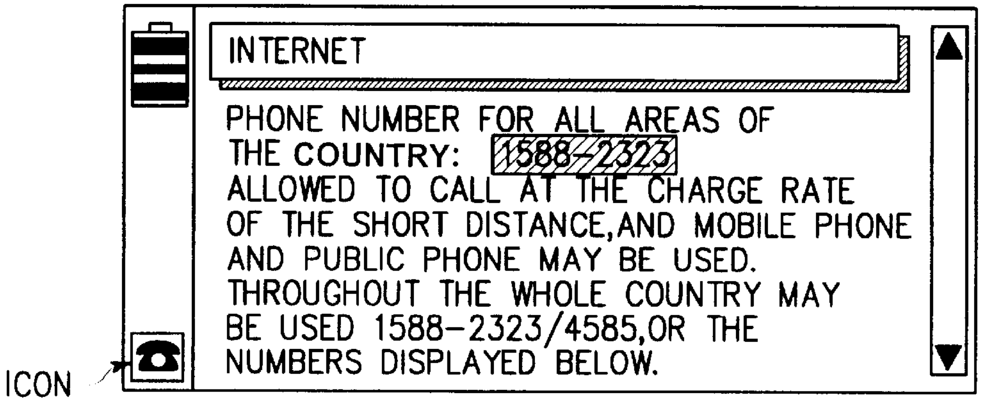

INTERNET
PHONE NUMBER FOR ALL AREAS OF
THE COUNTRY: 1588-2323
ALLOWED TO CALL AT THE CHARGE RATE
OF THE SHORT DISTANCE,AND MOBILE PHONE
AND PUBLIC PHONE MAY BE USED.
THROUGHOUT THE WHOLE COUNTRY MAY
BE USED 1588-2323/4585,OR THE
NUMBERS DISPLAYED BELOW.
ICON

FIG. 3

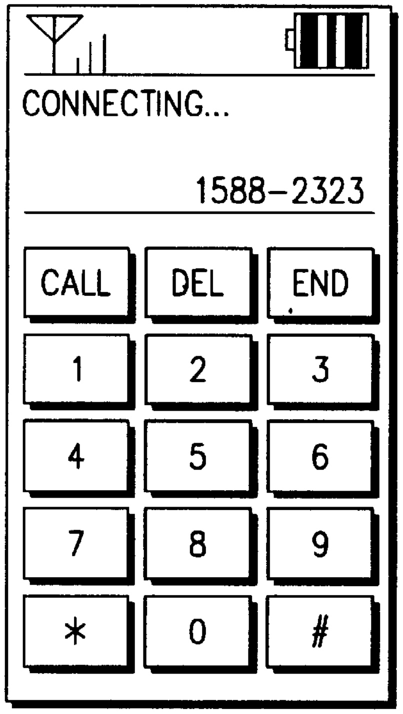

CONNECTING...
1588-2323
CALL
DEL
END
1
2
3
4
5
6
7
8
9
*
0
#

FIG. 4

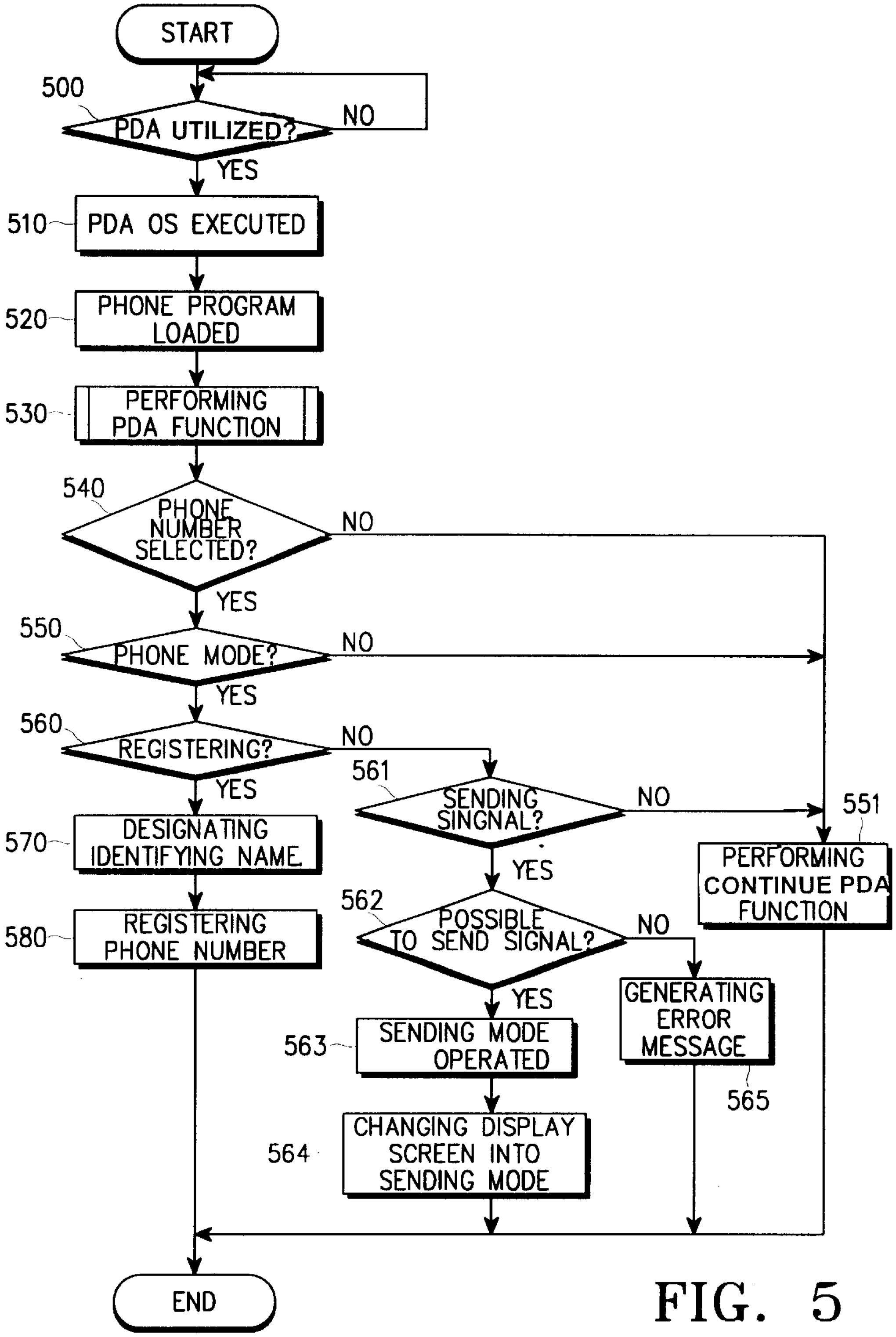

START
500
PDA UTILIZED?
NO
YES
510
PDA OS EXECUTED
520
PHONE PROGRAM LOADED
530
PERFORMING PDA FUNCTION
540
PHONE NUMBER SELECTED?
NO
YES
550
PHONE MODE?
NO
YES
560
REGISTERING?
NO
YES
561
SENDING SINGNAL?
NO
YES
551
PERFORMING CONTINUE PDA FUNCTION
570
DESIGNATING IDENTIFYING NAME
580
REGISTERING PHONE NUMBER
562
POSSIBLE TO SEND SIGNAL?
NO
YES
563
SENDING MODE OPERATED
GENERATING ERROR MESSAGE
565
564
CHANGING DISPLAY SCREEN INTO SENDING MODE
END

FIG. 5

METHOD FOR DIALING IN A SMART PHONE

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a combination of a personal digital assistant (PDA) and a mobile phone, the combination of which is hereinafter referred to as a "smart phone", and more particularly to a method for dialing in such a smart phone during the execution of a program for the PDA function.

2. Description of the Related Art

When the smart phone is performing a PDA function to process data, and the user wishes to make a phone call by dialing a phone number displayed on a display of the smart phone, the user must first memorize or record the number, terminate the current PDA function, and then execute the dialing program to dial the phone number. This wastes time and can result in the loss of the phone number if the user forgets the phone number or part thereof between the time the user memorizes it and the time it is dialed.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide a method for readily dialing a selected phone number in a smart phone during the performance of a PDA function.

It is another object of the present invention to provide a method for readily registering a phone number in an electronic phone book of a smart phone during the performance of a PDA function.

According to an aspect of the present invention, a method for dialing a selected phone number in a smart phone, having both PDA (Personal Digital Assistant) and mobile phone functions, during the performance of a PDA function, comprises the steps of loading an OS (Operating System) program for dialing the phone number during the performance of the PDA function, displaying a dialing icon according to a received signal strength indicator (RSSI), and dialing the selected phone number by selecting a dialing icon in response to the dialing request of the user.

BRIEF DESCRIPTION OF THE ATTACHED DRAWINGS

FIG. **1** is a block diagram illustrating the structure of a smart phone having both PDA and mobile phone functions according to an embodiment of the present invention;

FIG. **2** is a flow chart illustrating the steps of dialing a selected phone number in a smart phone according to an embodiment of the present invention;

FIG. **3** is a diagrammatic representation of a display screen of a smart phone displaying phone numbers for selection during the performance of the PDA function according to an embodiment of the present invention;

FIG. **4** is a diagrammatic representation of the display screen of a smart phone displaying a key matrix for dialing according to an embodiment of the present invention; and

FIG. **5** is a flow chart illustrating the steps of registering a phone number selected for dialing in a phone book of a smart phone according to another embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

A preferred embodiment of the present invention will be described in detail herein below with reference to the

accompanying drawings. In the following description, numerous specific details are set forth to provide a more thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well known functions or constructions have not been described so as not to obscure the present invention.

Referring to FIG. **1**, a smart phone consists of a mobile phone section **10** and a PDA section **20**. The mobile phone section **10** includes a first control unit **101**, first interface circuit **102**, first memory device **103**, RF circuit **104**, and sound signal processor **105**. In addition, the PDA section **20** comprises a second control unit **201**, second interface circuit **202**, second memory device **203**, display device **204**, and touch screen **205**.

The first control unit **101** of the mobile phone section **10** controls all of the mobile phone functions such as sending and receiving signals. The first interface circuit **102** interfaces the first control unit with the PDA section **20**. The first memory device **103** stores a plurality of programs for performing the mobile phone functions, and comprises RAM (Random Access Memory), ROM (Read Only Memory), etc. The sound signal processor **105** decodes the signals received through an antenna into the corresponding sound signals applied to a speaker, and encodes the sound signals received through a microphone into the corresponding digital signals. The RF circuit **104** modulates the digital signals with the baseband, amplifies, up-converts, and transmits them through the antenna, while demodulating the signals received through the antenna into the corresponding digital signals through down-conversion and amplifying.

The second control unit **201** of the PDA section **20** controls all of the PDA functions, enabling the smart phone to send a signal during the performance of the PDA function. The second memory device **203** stores a plurality of programs for performing the PDA functions, including RAM, ROM, etc. The RAM is used to load the OS (Operating System) program for driving the PDA section **20**, and to load the program for editing and dialing a phone number according to an embodiment of the present invention. The display device **204** displays graphic data including various data produced during operation of the PDA and mobile phone functions. It also displays the icon for editing and dialing a phone number. The touch screen **205** is used to enter data. The use of a touch screen is not meant to be exclusive and may be replaced by a conventional keypad along with a conventional mouse, or other similar configuration.

The operation of the preferred embodiment of the present invention will now be described with reference to FIGS. **1** to **4**. First, referring to FIG. **2**, if the user enters the start data through the touch screen in step **200**, the second control unit **201** executes the OS program for operating the PDA section **20** in step **210**. Then, the first control unit **101** of the mobile phone section measures in step **220** the RSSI of the signals received through the antenna and displays the dialing icon in step **230** as shown in FIG. **3**. The dialing icon is displayed if the RSSI is equal to or greater than a predetermined value. The displaying of the dialing icon depends on the periodically measured RSSI during operation of the PDA function.

If no phone number is selected the process continues to step **251** to continue performing the PDA function. If the user selects a phone number displayed on the display device **204** during the performance of the PDA function in step **240** (refer to FIG. **3**), and selects the dialing icon in step **250**, the process displays a phone editor and goes to step **260** to determine if the phone can transmit the phone signals. The phone editor allows the user to edit the phone number prior to dialing. If the signals cannot be transmitted the process continues to step **261**, where an error message is displayed.

If the smart phone can transmit signals, it enters a signal sending mode **270**. Of course, if the user does not select the dialing icon in step **250**, the process goes to step **251** continue performing the PDA function. The selection of the phone number may be achieved by pressing a pen of a touch screen or by the dragging of a mouse.

In step **270**, the second control unit **201** of the PDA section **20** sends a signal through the mobile phone section **10** to dial the selected phone number. Then in step **280** it changes the display screen into the dial pad mode, as shown in FIG. **4**, by clearing any images presently displayed. Summarizing the present embodiment, when the user utilizes the PDA section **20**, the smart phone loads the OS program to enable it to send a signal along with displaying the dialing icon, so that a phone number selected by the user during operation of the PDA function may be dialed by selecting the dialing icon.

Describing another embodiment of the present invention in connection with FIGS. **1** and **3** to **5**, a dialing operation is accomplished by a phone program for editing and dialing a phone number loaded in the RAM during the performance of the PDA function without employing the OS program supporting the dialing operation as in the previous embodiment.

Referring to FIG. **5**, if the PDA section **20** is utilized in step **500**, the OS system is executed in step **510**, and the phone program is loaded in step **520** so that the telephone icon is displayed on the display device **204** as shown in FIG. **3**. In step **530**, the second control unit **201** of the PDA section **20** performs a PDA function, for example, playing a game or editing a document. If no phone number is selected in step **540**, the process continues to step **551** to continue performing the PDA function. Meanwhile, if the user selects a phone number at step **540**, the process goes to step **550** to determine whether the telephone icon is selected or not. If the telephone icon is not selected the process continues to step **551** to continue performing the PDA function. If the telephone icon is selected, the PDA function is terminated to execute the phone program. The phone program determines in step **560** whether there is a request to register the selected phone number. If so, the process goes to step **570** to register the phone number in a phone book of the first memory **103** of the mobile phone section **10** or the second memory **203** of the PDA section **20** by designating a suitable address together with an identifying name. In step **570** a name is designated and registering or storing the name and phone number in memory at step **580** ends the process. If there is no request to register the number at step **560**, the process goes to step **561** to determine if the selected phone number is dialed. If the selected phone number is not dialed in step **561**, the process continues to step **551** to continue performing the PDA function. If so, the process goes to step **562** to determine whether the smart phone is enabled to send a signal. This is accomplished by measuring the RSSI and determining whether the smart phone is in the state of sending or receiving signals. If the phone can send the signal, i.e. the signal is strong enough and the smart phone is neither receiving or sending signals, the process goes to step **563** to send a signal corresponding to the phone number, and then to step **564** to change the display screen into the sending mode, as shown in FIG. **4**, after terminating the PDA function. However, if the signal cannot be sent at step **562**, an error message is displayed on the display device **204** at step **565**.

Summarizing the operation of the second embodiment, when the PDA section **20** is utilized, the phone program for editing and dialing a phone number is loaded in the RAM, and the telephone icon is displayed on the display screen of the display device **204**. Then, if the user selects a phone number and selects the telephone icon during operation of the PDA function, the control unit executes the phone program to register the selected phone number into the phone book and/or send a signal corresponding thereto after terminating the PDA function.

While the present invention has been described in connection with specific embodiments accompanied by the attached drawings, it will be readily apparent to those skilled in the art that various changes and modifications may be made thereto without departing from the scope and spirit of the present invention as defined by the appended claims.

What is claimed is:

1. A method for dialing a phone number in a smart phone having both PDA (Personal Digital Assistant) and mobile phone functions during operation of the PDA function, comprising the steps of:

loading an OS (Operating System) program for dialing the phone number during operation of the PDA function;

displaying a dialing icon if a received signal strength indicator (RSSI) is equal to or greater than a predetermined value; and

dialing said phone number by selecting said dialing icon.

2. The method as defined in claim **1**, wherein said dialing step further comprises the steps of:

comparing said RSSI with the predetermined value;

determining whether said smart phone is sending or receiving signals if said RSSI is equal to or greater than said predetermined value; and

switching a display screen into a dialing state by performing a dialing routine.

3. The method as defined in claim **2**, further including the step of displaying an error message if one of said RSSI is below said predetermined value and said smart phone is sending or receiving signals.

4. The method as defined in claim **1**, wherein said phone number is selected by one of pressing a touch screen and dragging a mouse.

5. A method for dialing a phone number in a smart phone having random access memory (RAM) and both personal digital assistant (PDA) and mobile phone functions during operation of a PDA function, comprising the steps of:

loading an operating system (OS) program for said PDA function;

loading a phone program for editing and dialing a phone number along with displaying a phone editor and dialing icon if said PDA function is requested by a user;

executing said phone program if said user selects a phone number during operation of said PDA function;

storing an identifying name designated for the selected phone number into a phone book; and

dialing the selected phone number.

6. The method as defined in claim **5**, wherein said dialing step further comprises the steps of:

checking a received signal strength indicator (RSSI) with a predetermined value;

determining whether said smart phone is sending or receiving signals if said RSSI is equal to or greater than said predetermined value; and

switching a display screen into a dialing state by performing a dialing routine.

7. The method as defined in claim **6**, further including the step of displaying an error message if one of said RSSI is below said predetermined value and said smart phone is sending or receiving signals.

8. The method as defined in claim **6**, wherein said phone editor and dialing icon are shaped like a telephone.

9. The method as defined in claim **5**, wherein said phone number is selected by one of pressing a touch screen and dragging a mouse.

10. A method for dialing a phone number in a smart phone having both personal digital assistant (PDA) and mobile phone functions, comprising the steps of:

executing a dialing program for editing and dialing a phone number and displaying a phone editor and a dialing icon when a PDA function is utilized in said smart phone;

switching a display screen into a dialing state for selecting a phone number when said dialing icon is selected during the performance of said PDA function;

storing an identifying name designated for the selected phone number into a phone book; and

dialing the selected phone number.

11. The method as defined in claim **10**, wherein said display screen is switched into the dialing state provided a received signal strength indicator (RSSI) is equal to or greater than a predetermined value and said smart phone is not sending or receiving signals.

12. The method as defined in claim **11**, further including the step of displaying an error message if one of said RSSI is below said predetermined value and said smart phone is sending or receiving signals.

13. The method as defined in claim **10**, wherein said selected phone number is selected by one of pressing a touch screen and dragging a mouse.

* * * * *

<u>Exhibit 4</u>

US006879843B1

(12) **United States Patent**
Kim

(10) **Patent No.: US 6,879,843 B1**
(45) **Date of Patent: Apr. 12, 2005**

(54) **DEVICE AND METHOD FOR STORING AND REPRODUCING DIGITAL AUDIO DATA IN A MOBILE TERMINAL**

(75) Inventor: **Dong-Woo Kim**, Kumi (KR)

(73) Assignee: **Samsung Electronics Co., Ltd.** (KR)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 581 days.

(21) Appl. No.: **09/633,059**

(22) Filed: **Aug. 8, 2000**

(30) **Foreign Application Priority Data**

Aug. 12, 1999 (KR) 99-33207

(51) **Int. Cl.**[7] **H04Q 7/32**; H04M 1/725
(52) **U.S. Cl.** **455/557**; 455/556.1; 455/559
(58) **Field of Search** 455/557, 556.1, 455/566, 552.1, 414.1, 414.4, 419, 559, 556, 552, 575

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,014,569 A * 1/2000 Bottum 455/466
6,021,325 A * 2/2000 Hall 455/412
6,167,116 A * 12/2000 Freadman 379/88.01
6,212,359 B1 * 4/2001 Knox 455/3.1
6,317,609 B1 * 11/2001 Alperovich et al. 455/556.1
6,332,175 B1 * 12/2001 Birrell 711/112
6,470,051 B1 * 10/2002 Campisano 375/240.21
6,584,510 B2 * 6/2003 Anttila 709/250
6,587,684 B1 * 7/2003 Hsu et al. 455/419
6,671,567 B1 * 12/2003 Dwyer et al. 700/94

FOREIGN PATENT DOCUMENTS

DE 199 43 306 A1 4/2000
DE 199 17 169 A1 11/2000
EP 0 804 012 * 10/1997 .... H04M/11/08
EP 0804012 * 10/1997 .... H04M/11/08
EP 0 898 378 A2 * 2/1999 .... H04B/1/20
EP 1 049 312 A2 11/2000
GB 2 308 775 A * 2/1997 .... H04M/1/02
WO WO 00/70896 11/2000

* cited by examiner

Primary Examiner—William Trost
Assistant Examiner—Brandon J. Miller
(74) *Attorney, Agent, or Firm*—Dilworth & Barrese LLP

(57) **ABSTRACT**

A method and device for reproducing digital audio data in a mobile station is disclosed. The mobile station can be connected by an adapter with a personal computer in order to download MP3 digital audio data which was previously downloaded from the Internet to a hard disk on the personal computer. The adapter converts parallel data from the computer into serial data for the mobile station, and vice versa. The MP3 digital audio data is downloaded to the mobile station according to a downloading program stored in the personal computer. One method according to the invention comprises the steps of downloading the MP3 audio data stored in the computer to a memory device of the mobile station, and selectively reproducing the MP3 audio data stored in the memory device.

11 Claims, 9 Drawing Sheets

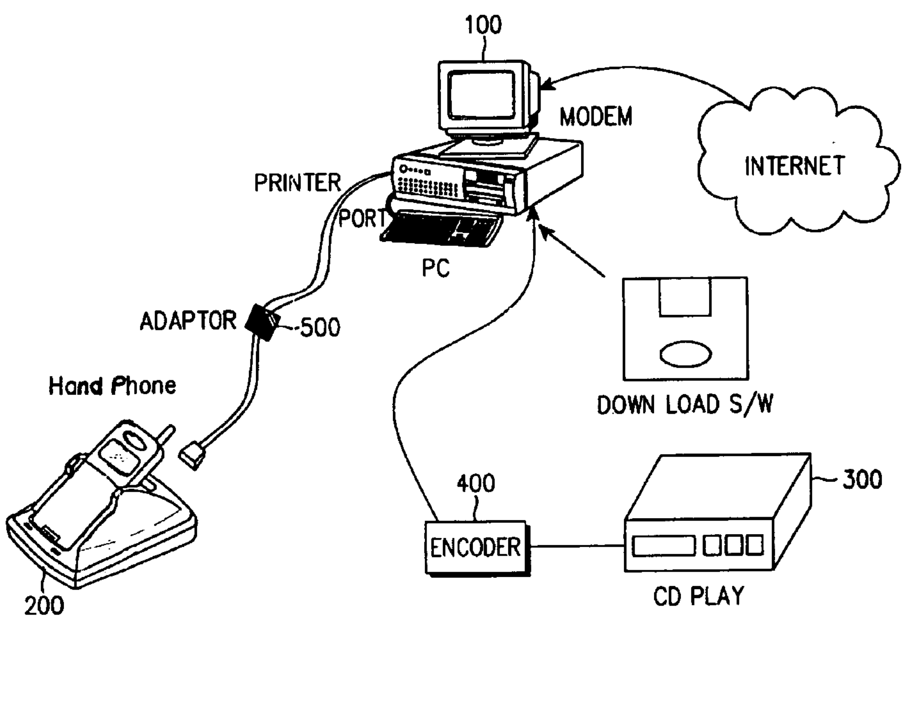


FIG. 1

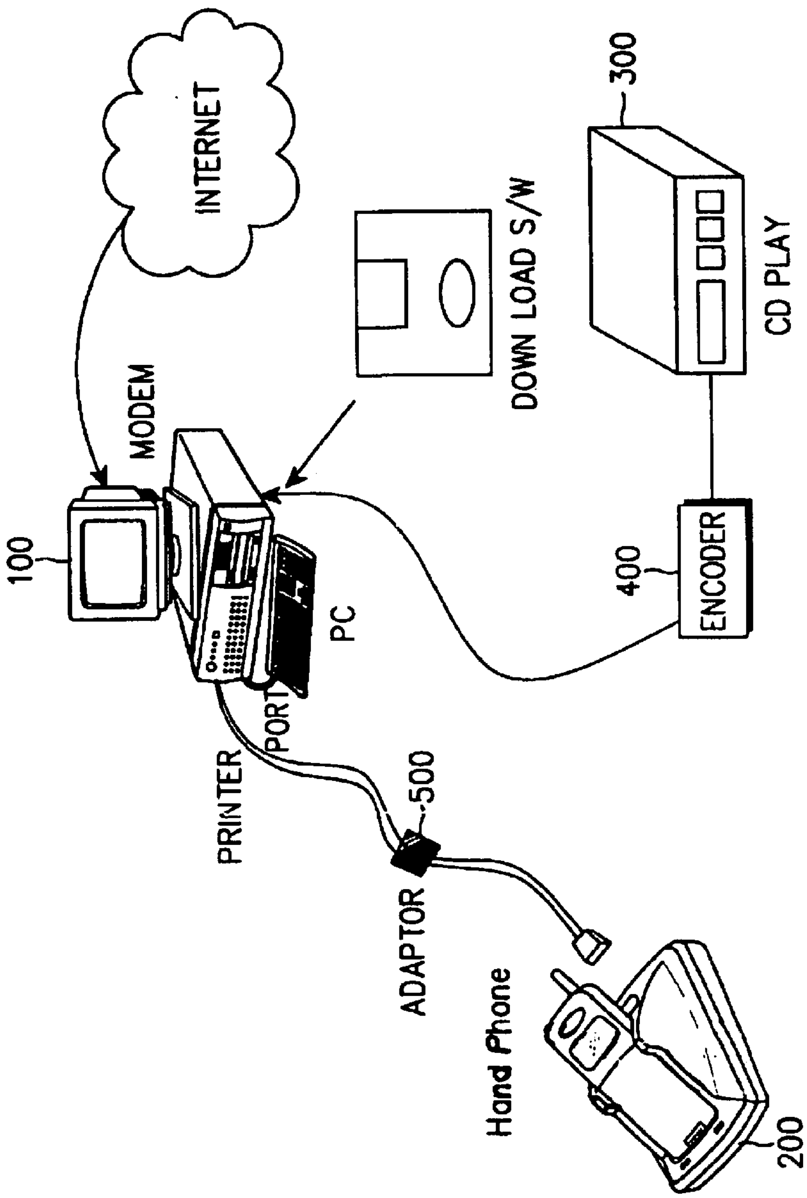
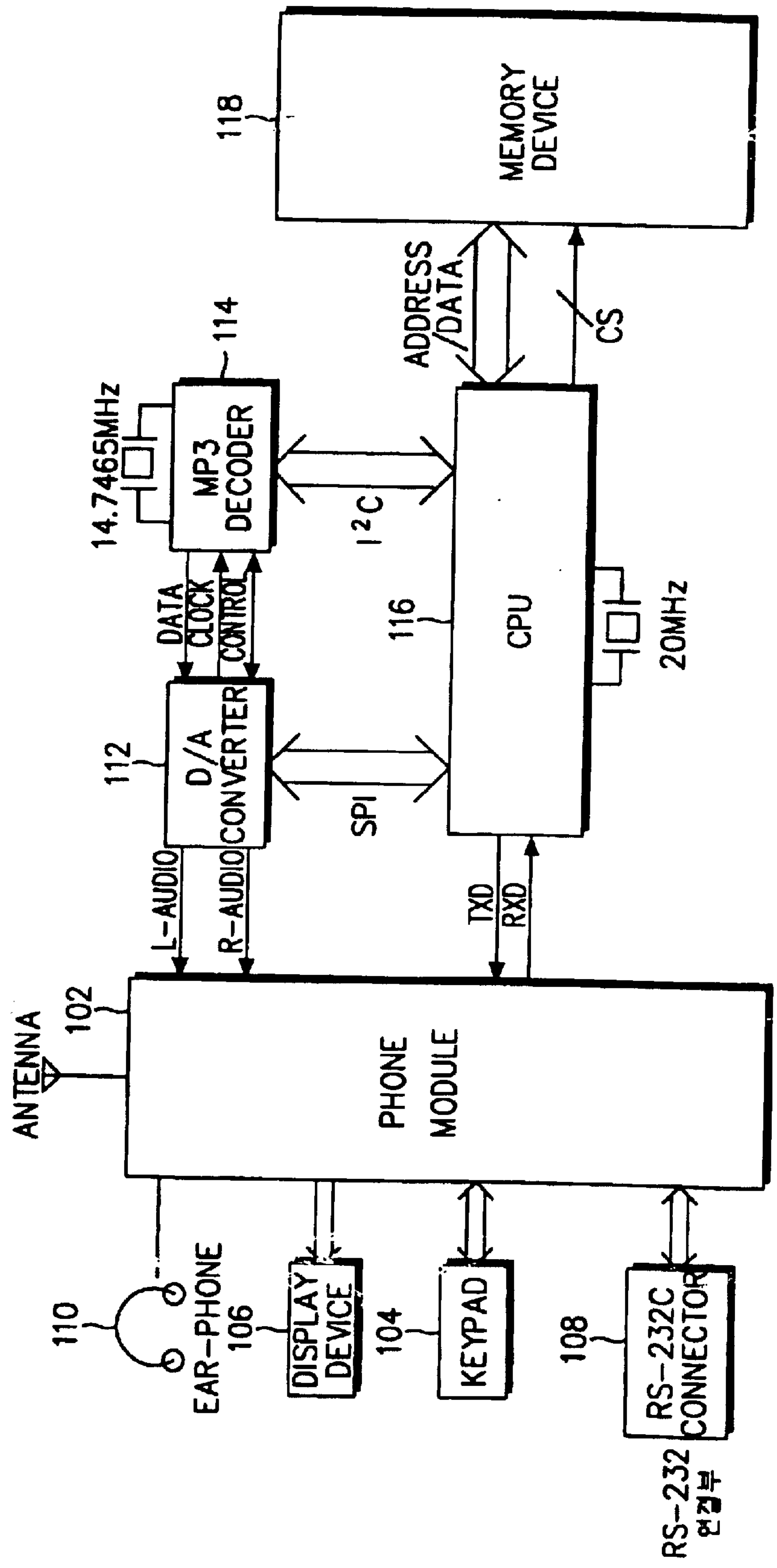

INTERNET
MODEM
100
PC
PRINTER
PORT
500
ADAPTOR
Hand Phone
200
DOWN LOAD S/W
300
CD PLAY
400
ENCODER

FIG. 2


118
MEMORY DEVICE
ADDRESS /DATA
CS
114
14.7465MHz
MP3 DECODER
I²C
116
CPU
20MHz
DATA
CLOCK
CONTROL
112
D/A CONVERTER
SPI
L-AUDIO
R-AUDIO
TXD
RXD
ANTENNA
102
PHONE MODULE
110
EAR-PHONE
106
DISPLAY DEVICE
104
KEYPAD
108
RS-232C CONNECTOR
RS-232 연결부

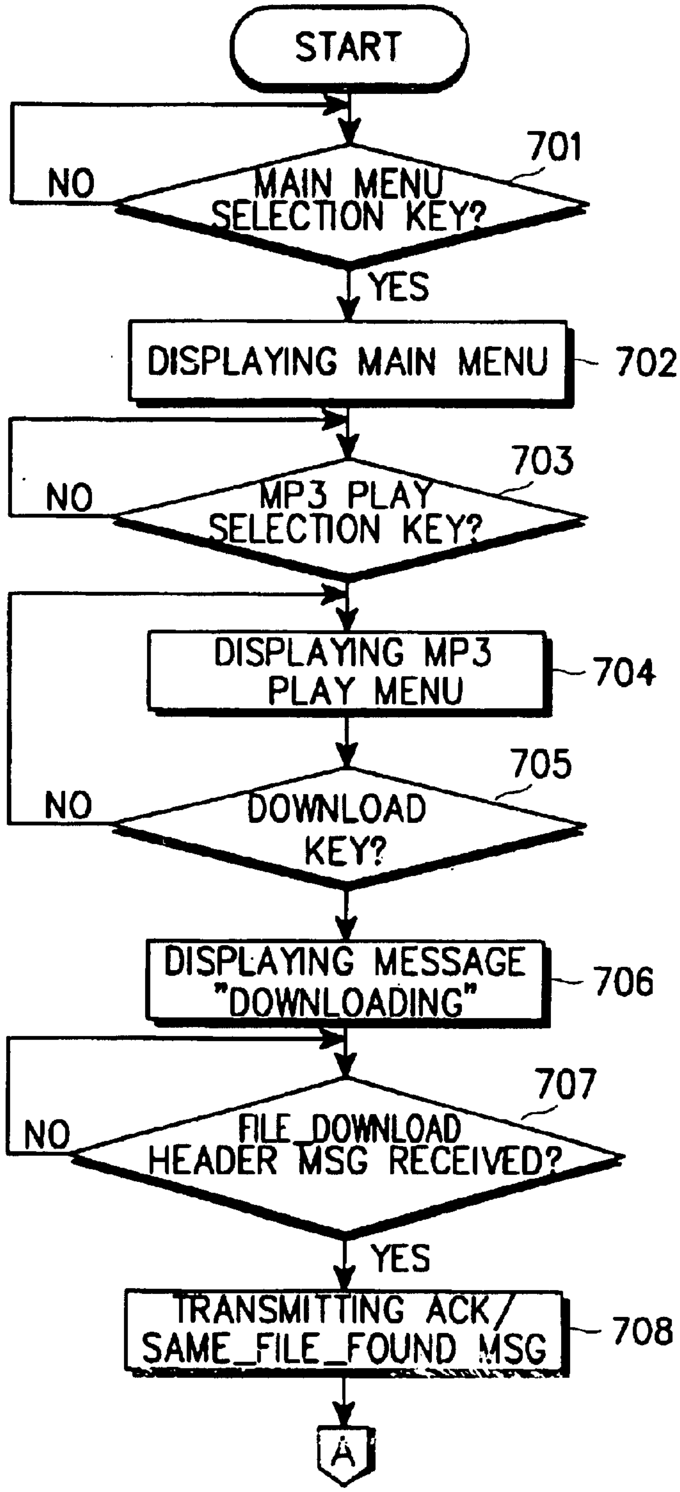

START
MAIN MENU SELECTION KEY?
701
NO
YES
DISPLAYING MAIN MENU
702
MP3 PLAY SELECTION KEY?
703
NO
DISPLAYING MP3 PLAY MENU
704
DOWNLOAD KEY?
705
NO
DISPLAYING MESSAGE "DOWNLOADING"
706
FILE_DOWNLOAD HEADER MSG RECEIVED?
707
NO
YES
TRANSMITTING ACK/ SAME_FILE_FOUND MSG
708
A

FIG. 3A

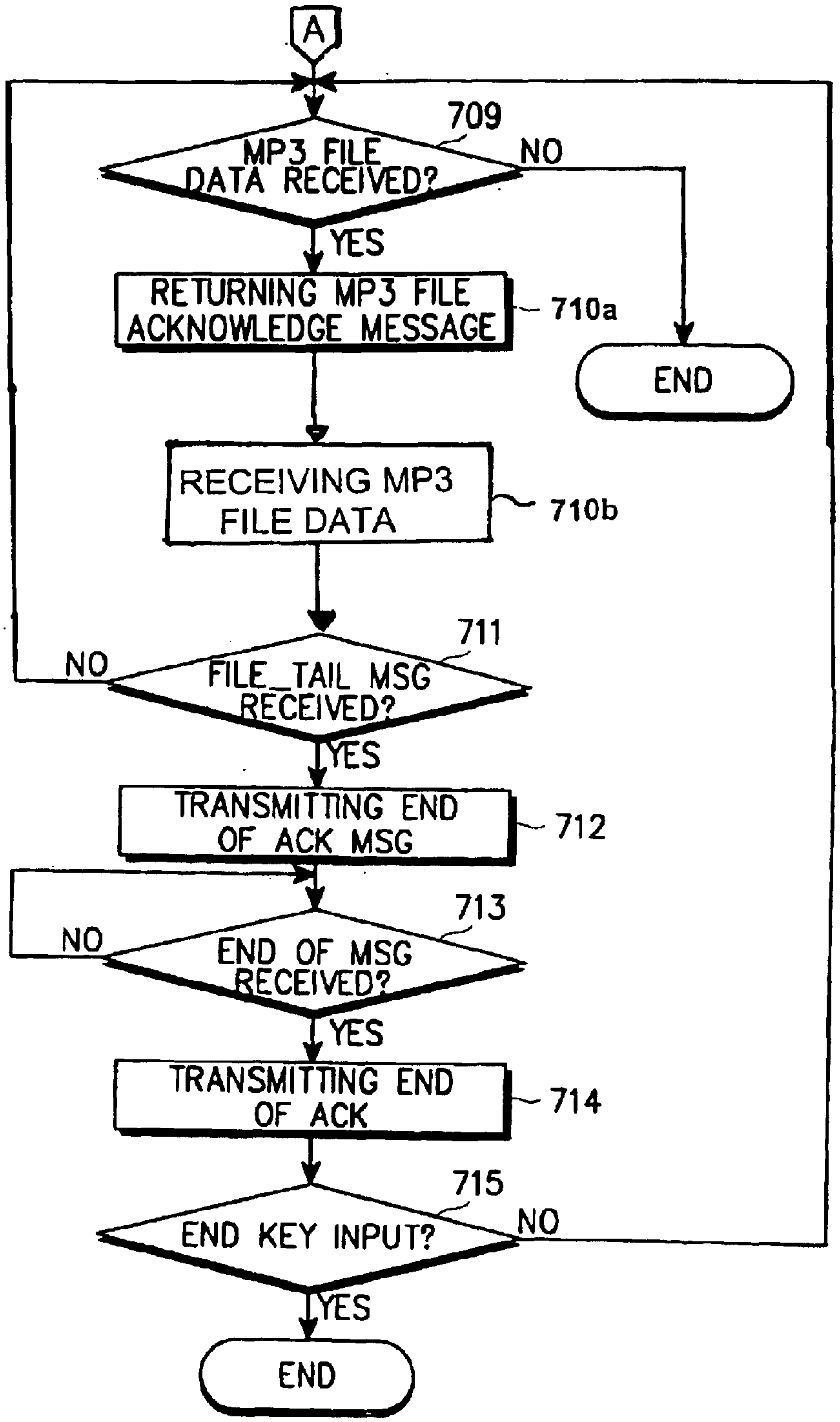

A
709
MP3 FILE DATA RECEIVED?
NO
YES
RETURNING MP3 FILE ACKNOWLEDGE MESSAGE
710a
END
RECEIVING MP3 FILE DATA
710b
711
NO
FILE_TAIL MSG RECEIVED?
YES
TRANSMITTING END OF ACK MSG
712
713
NO
END OF MSG RECEIVED?
YES
TRANSMITTING END OF ACK
714
715
END KEY INPUT?
NO
YES
END

FIG. 3B

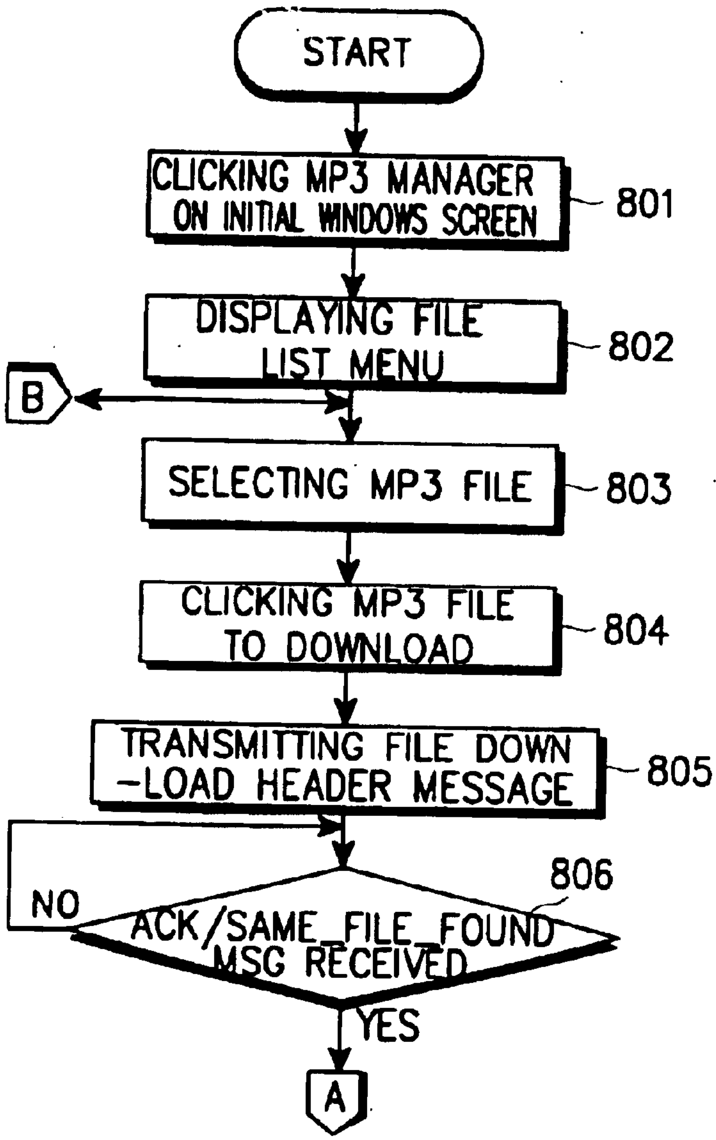

START
CLICKING MP3 MANAGER ON INITIAL WINDOWS SCREEN
801
DISPLAYING FILE LIST MENU
802
B
SELECTING MP3 FILE
803
CLICKING MP3 FILE TO DOWNLOAD
804
TRANSMITTING FILE DOWN-LOAD HEADER MESSAGE
805
806
NO
ACK/SAME_FILE_FOUND MSG RECEIVED
YES
A

FIG. 4A

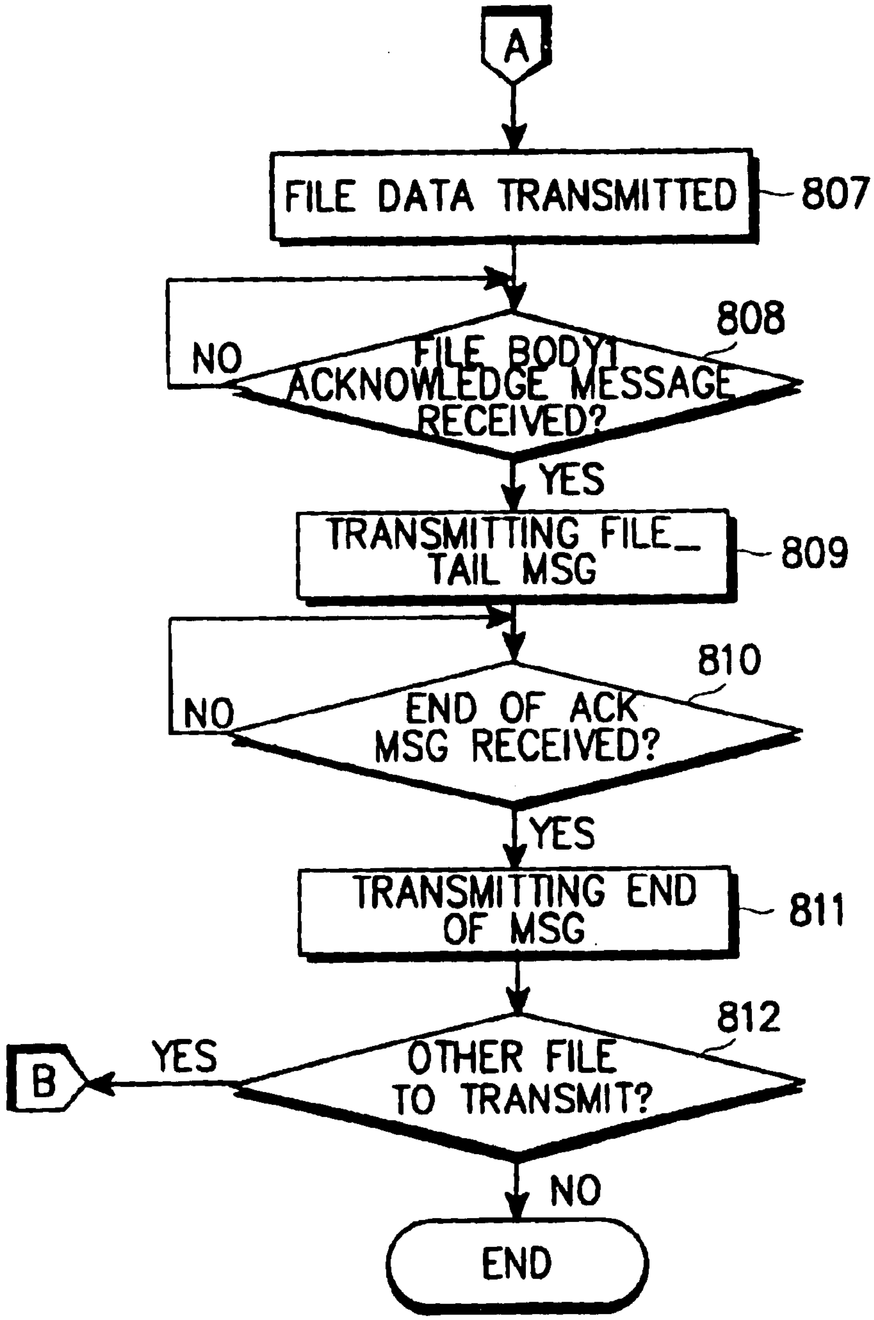

A
FILE DATA TRANSMITTED
807
NO
FILE BODY1 ACKNOWLEDGE MESSAGE RECEIVED?
808
YES
TRANSMITTING FILE_ TAIL MSG
809
NO
END OF ACK MSG RECEIVED?
810
YES
TRANSMITTING END OF MSG
811
B
YES
OTHER FILE TO TRANSMIT?
812
NO
END

FIG. 4B

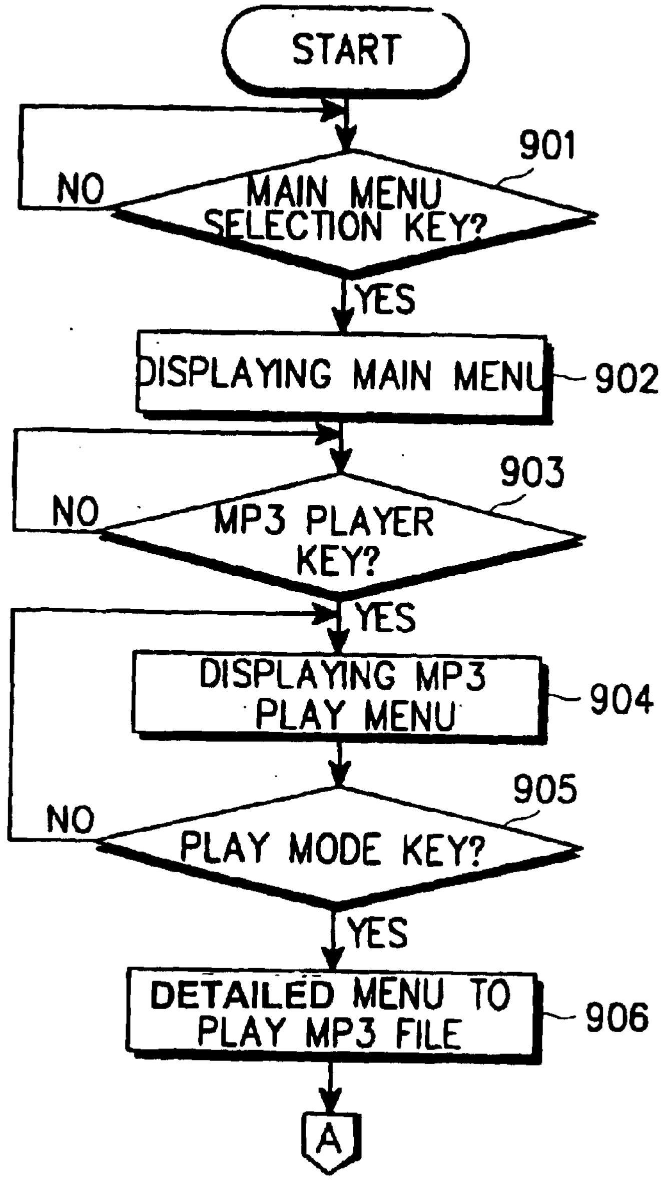

START
901
NO
MAIN MENU SELECTION KEY?
YES
DISPLAYING MAIN MENU
902
903
NO
MP3 PLAYER KEY?
YES
DISPLAYING MP3 PLAY MENU
904
905
NO
PLAY MODE KEY?
YES
DETAILED MENU TO PLAY MP3 FILE
906
A

FIG. 5A

FIG. 5B

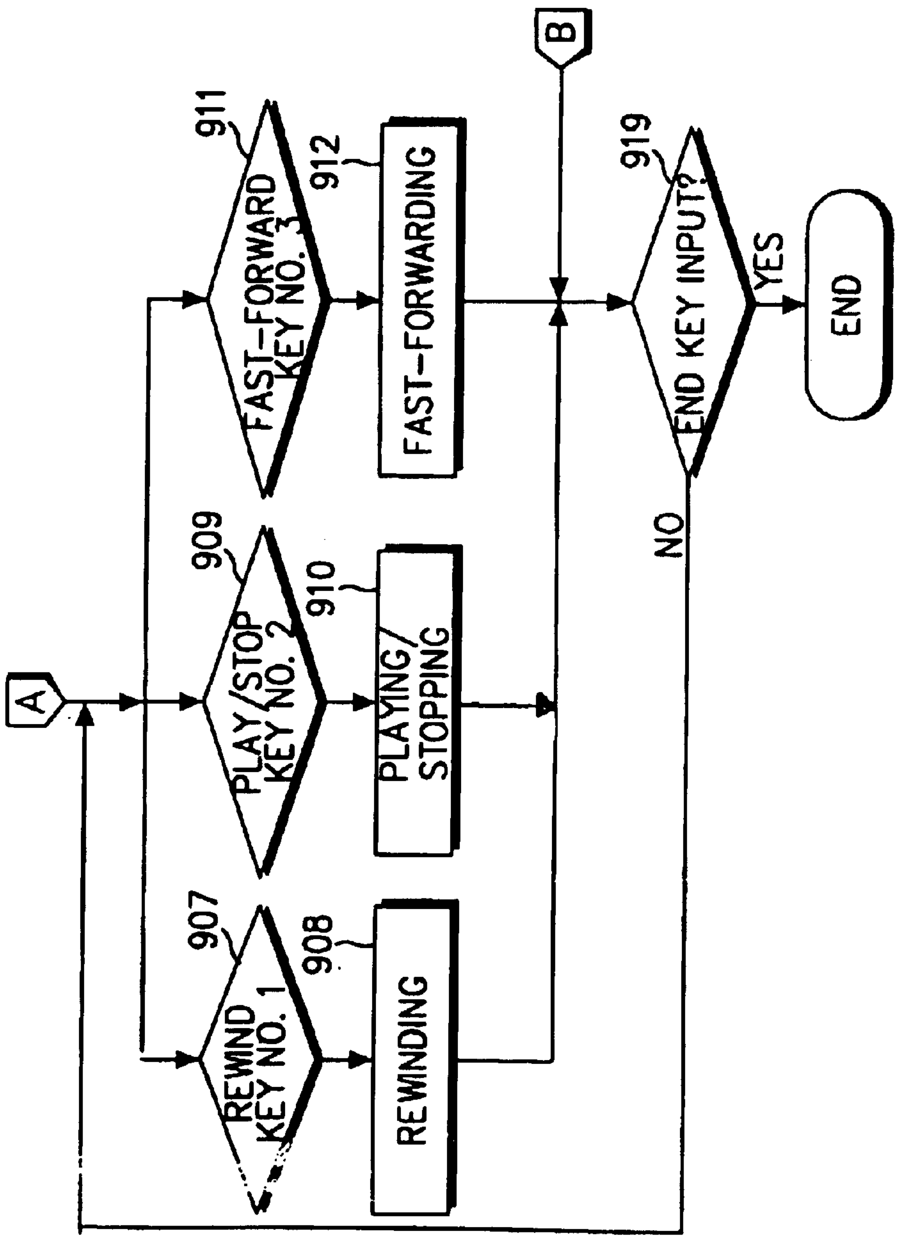
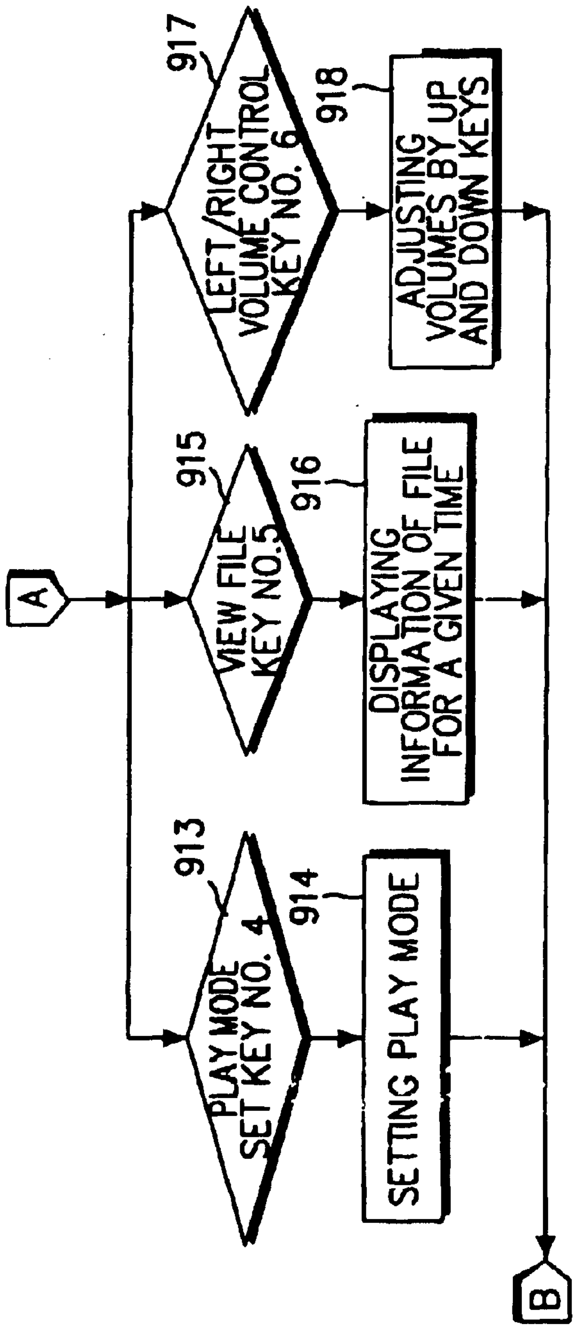

A
907
REWIND KEY NO. 1
908
REWINDING
909
PLAY/STOP KEY NO. 2
910
PLAYING/ STOPPING
911
FAST-FORWARD KEY NO. 3
912
FAST-FORWARDING
B
919
END KEY INPUT?
YES
NO
END

FIG. 5C


A
PLAY MODE SET KEY NO. 4
913
SETTING PLAY MODE
914
VIEW FILE KEY NO.5
915
DISPLAYING INFORMATION OF FILE FOR A GIVEN TIME
916
LEFT/RIGHT VOLUME CONTROL KEY NO. 6
917
ADJUSTING VOLUMES BY UP AND DOWN KEYS
918
B

DEVICE AND METHOD FOR STORING AND REPRODUCING DIGITAL AUDIO DATA IN A MOBILE TERMINAL

PRIORITY

This application claims priority to an application entitled “Mobile Station with a Digital Audio Data Storing/Reproducing Device and Method for Controlling It” filed in the Korean Industrial Property Office on Aug. 12, 1999 and assigned Serial No. 99-33207, the contents of which are hereby incorporated by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a mobile phone with a digital audio data storing/reproducing device that enables the mobile phone to store/reproduce digital audio data, which may be supplied from the Internet or a CD player.

2. Description of the Related Art

The MP3 computer file format is used to compress large amounts of information into small packages that can easily be sent over the Internet. “MP3” is the common name for MPEG ½ layer-3, a standard established by the Moving Pictures Engineering Group (MPEG) under the auspices of the International Organization for Standardization (ISO). The MP3 compressed information can be anything capable of being stored as digital information, such as video clips, art, or music. The information can be decompressed during playback and used or stored as a computer file.

If MP3 technology was employed in a mobile phone, it would be possible for the user of the mobile phone to listen to music without carrying a portable cassette tape recorder or MP3 player.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide a mobile station with a device for storing MP3 audio data received from the Internet or a CD player through a computer and reproducing it as desired, and a method therefor.

According to an aspect of the present invention, a mobile station is connected by an adapter with a personal computer in order to download MP3 digital audio data, which was previously downloaded from the Internet to a hard disk in the personal computer. The adapter converts parallel data from the computer into serial data for the mobile station, and vice versa. The MP3 digital audio data is downloaded to the mobile station according to a downloading program stored in the personal computer. A method for reproducing digital audio data in a mobile station comprises the steps of downloading the MP3 audio data stored in the computer to a memory device of the mobile station, and selectively reproducing the MP3 audio data stored in the memory device.

BRIEF DESCRIPTION OF THE DRAWINGS

The preferred embodiment of the present invention will be described more specifically with reference to the drawings attached only by way of example.

FIG. **1** is a schematic diagram for illustrating the procedure of downloading digital audio data, which has been downloaded from the Internet or a CD player to a computer, to a mobile station, according to the present invention;

FIG. **2** is a block diagram for illustrating a mobile station provided with a device for storing/reproducing digital audio data according to the present invention;

FIGS. **3**A and **3**B are flow charts for illustrating the process of a mobile station to receive an MP3 file according to the present invention;

FIG. **4** is a flow chart for illustrating the process of a personal computer for downloading an MP3 file to a mobile station according to the present invention; and

FIGS. **5**A–**5**C are flow charts for illustrating the process of a mobile station for reproducing the sound of an MP3 file data according to the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

In the following description of the preferred embodiment, a detailed description of conventional components not directly related to the invention is omitted in order to avoid confusion.

Referring to FIG. **1**, a personal computer **100** stores digital audio data or MP3 files in memory. Digital audio data or MP3 files can be downloaded from the Internet or an encoder **400**, and the computer **100** is equipped with a download program for downloading digital audio data to a mobile phone **200**, which can store and reproduce it. A CD player **300** reproduces the digital audio data recorded on a compact disk (CD). The encoder **400** converts the audio data reproduced by the CD player **300** into a corresponding MP3 file delivered to the personal computer **100**. An adapter **500** is needed in order to connect the personal computer **100** and mobile phone **200** when downloading MP3 files. Adapter **500** converts parallel data from the personal computer **100** into serial data for the mobile phone **200**, and vice versa. The adapter **500** is connected to the printer port of the personal computer **100** and the J2 connector of the mobile phone **200**. The audio data downloaded from the personal computer consists of, for example, 8-bit parallel and control data.

Referring to FIG. **2**, a phone module **102** controls the ordinary functions of the mobile station, enabling the digital audio data (i.e., MP3) to be downloaded from a personal computer **100** to the mobile station when there is a download key input on the key pad, and reproducing the sound of the digital audio data when it is selected to play back using the keypad **104**. The keypad **104** has a plurality of alphanumeric keys which a user uses to input commands in order to perform mobile communication, to record (to store) digital audio data in the mobile station, to play or to stop the reproduction of sound from the stored digital audio data, to rewind the stored digital audio data, and fast forwarding the stored digital audio data. A display device **106** displays information generated from the keypad **104** and the phone module **102** under the control of the phone module **102**. The display device **106** consists of an LCD, and light emitting diodes so that the user can see information displayed on the LCD at night. An RS-232C connector **108** enables the mobile station to exchange audio data (MP3 files) and other data with the personal computer **100**. An earphone **110** is connected to the phone module **102** to generate the sound of the audio data supplied from the phone module **102**.

An MP3 decoder **114** is for decoding MP3 audio data into corresponding PCM (Pulse Code Modulation) data, and can be, for example, a STO13 module manufactured by STMICROELECTRONICS Co. Specifically, it de-multiplexes the MP3 audio data, which can be an encoded audio stream, at 8 to 320 kbps, into control data and audio data in order to Huffman-decode the audio data from the run-length coded compressed signal to the signal of the original length, to subject each sub-band of the signal to re-quantization and re-scaling according to the control data, to recover the resulting data by the inverse discrete cosine transformation, and to inversely filter each sub-band to finally obtain the PCM data. A D/A (digital to analog) converter **112** converts the PCM audio data from the MP3 decoder into the corresponding analog audio data, and can be, for example, a CS4331 module for stereo audio manufacture by CRYSTAL Co. It includes circuits for digital interpolation, delta-sigma D/A conversion digital de-emphasis, and filtering. A Central Processing Unit (CPU) **116**, which can be a H8/2134 module manufactured by HITACHI Co., controls the storage (in memory device **118**) and reproduction of the MP3 audio data received from the phone module.

The personal computer **100**, as shown in FIG. **1**, stores on the hard disk a MP3 file downloaded from the Internet or the encoder **400** connected to CD player **300**. The personal computer **100** also must be installed with a MP3 control program for the mobile station. For example, the CD storing the MP3 control program is inserted into the CD driver of the computer, and then used to generate an MP3 control program folder on the personal computer **100**. Then, the personal computer **100** is connected to the mobile station **200** with the adapter cable.

Hereinafter, the process of downloading an MP3 file from a personal computer to the mobile station is described in connection with FIGS. **3**A to **4**. Referring to FIG. **3**A, detecting the main menu selection key inputted from the keypad **104** in step **701**, the phone module **102** proceeds to step **702** to display the main menu in the display device **106** as shown in Table 1.

TABLE 1

0.	MP3 Player
1.	Voice Dial
2.	Bell/Vibrator Intensity
3.	Time Schedule
4.	Arrangement/Alarm
5.	Private Information
6.	Additional Services

If the user presses the key numbered ‘0’ in step **703**, the phone module **102** proceeds to step **704** to display the MP3 play menu in the display device as shown in Table 2.

TABLE 2

MP3 Play	
0.	Play Mode
1.	Download
2.	File List
3.	Delete File

As shown in Table 2, the keys numbered respectively ‘0’, ‘1’, ‘2’, and ‘3’ are pressed in order to play, an MP3 file, download an MP3 file, list the names of the stored MP3 files, and delete the stored MP3 files, respectively. For example, if the key numbered ‘1’ (the download command) is pressed in step **705**, the phone module **102** proceeds to step **706** to display the downloading message in the display device **106**.

Meanwhile, referring to FIG. **4**A, if the user clicks the MP3 manager icon displayed on the desktop screen of the personal computer **100** in step **801**, the computer **100** displays a menu list of MP3 files on the monitor screen in step **802**. Selecting an MP3 file from the file list menu in step **803**, and clicking the selected MP3 file in order to download it in step **804**, the computer **100** proceeds in step **805** to transmit the file download header message FILE_DOWNLOAD HEADER MSG through the RS-232C connector **108** to phone module **102** in the mobile station. Then, as shown in FIG. **3**A, the phone module **102** detects the FILE_DOWNLOAD HEADER MSG in step **707** and proceeds in step **708** to return the file download message acknowledge message ACK/SAME_FILE_FOUND MSG to the personal computer **100**. Accordingly, as shown in FIG. **4**A, the personal computer **100** receives the ACK/SAME_FILE_FOUND MSG in step **806**, and proceeds, jumping to FIG. **4**B, in step **807** to transmit the MP3 file data FILE BODY **1** to the phone module **102**. Then, as shown in FIG. **3**B, the phone module **102** detects FILE BODY **1** in step **709**, proceeds in step **710***b* to receive the MP3 file data after returning the MP3 file data acknowledge message to the personal computer **100** in step **710***a*. The MP3 file data received is transferred to the CPU **116**, and then stored into the memory **118**.

Subsequently, as shown in FIG. **4**B, the personal computer **100**, after detecting the FILE BODY **1** acknowledge message in step **808**, transmits the file transmission complete message FILE_TAIL_MSG to the phone module **102** in step **809** after completing the transmission of the MP3 file data. Then, returning to FIG. **3**B, the phone module **102** proceeds to step **712** or returns to step **709** depending on whether the FILE_TAIL_MSG is received (step **712**) or not (step **709**) in step **711**. The acknowledgment of the MP3 file information is transmitted to the personal computer, and, if the MP3 file is not received within five seconds when receiving the MP3 file from the personal computer, the existing received data is canceled and returns to the initial value. When the FILE_TAIL_MSG is received in step **711**, the phone module **102** transmits the end of acknowledge message END_OF_ACK_MSG to the personal computer in step **712**. Then, as shown in FIG. **4**B, the personal computer **100**, after detecting END_OF_ACK_MSG in step **810**, proceeds to step **811** to transmit the end of file transmission message END_OF_MSG to the phone module. Finally, in step **812**, the personal computer **100** either returns to step **803** or terminates the process, depending on whether there is another file to be transmitted (step **803**) or not (end). Meanwhile, returning to FIG. **3**B, the phone module **102**, after detecting END_OF_MSG in step **713**, proceeds to step **714** to transmit the end of acknowledge message END OF ACK to the personal computer **100**. Finally, in step **715**, the phone module terminates the download process or returns to step **709** depending on whether the end key is inputted from the keypad **104** (end) or not (step **709**).

Next, the procedure of reproducing the sound of the MP3 file stored in the mobile station will be described in connection with FIGS. **5**A, **5**B and **5**C. If the phone module **102** detects that the main menu selection key is input from the keypad **104** in step **901**, it proceeds to step **902** to display the main menu as shown in Table 1 in display device **106**. If the user presses key No.0 in step **903** to select the MP3 payer, the phone module **102** proceeds to step **904** to display the MP3 play menu shown in Table 2, in display device **106**. In the MP3 play menu, the user presses key No. 0 to play an MP3 file, key No. 1 to download an MP3 file, key No. 2 to view the file list, and key No. 3 to delete a file. If he presses No. 0 key to play in step **905**, the phone module **102** proceeds to step **906** to display the detailed MP3 play mode menu shown in Table 3 in the display device.

TABLE 3

MP3 Play Mode	
1.	Rewind
2.	Play/Stop
3.	Fast Forward
4.	Play Mode Set
5.	View File
6.	Left/Right Volume Control

In the MP3 play mode menu, the user may press key No. 1 to rewind, key No. 2 to play/stop, key No. 3 to fast forward, key No. 4 to set the play mode, key No. 5 to view the file, and key No. 6 to control the right and left volumes. For example, as shown in FIG. **5**B, if the phone module **102** detects the rewind key No. 1 input in step **907**, the phone module **102** proceeds in step **908** to rewind. The rewind command might be to skip to the previous or the first music piece by a short or prolonged press of the rewind key while not playing the music, to the starting position of the music piece presently being played by a short press of the rewind key while playing the music, to the starting position of the previous music piece by a short press of the rewind key when starting the following music piece, or to a desired rewind position by pressing the rewind key for a proper prolonged time while playing the music.

If the play/stop key No. 2 is pressed in step **909**, the phone module **102** proceeds in step **910** to stop if it is presently playing, or play if it is presently stopped. If the fast forward key No. 3 is pressed in step **911**, the phone module **102** proceeds in step **912** to fast forward. The fast forward command might be to skip to the next or the last music piece by a short or prolonged press of the rewind key while not playing, to skip to the starting position of the next music piece by a short press of the fast forward key while playing the music, or to the music piece of a desired forward position by pressing the rewind key for a proper prolonged time while playing the music. As shown in FIG. **5**C, if the play mode set key No. 4 is pressed in step **913**, the phone module **102** proceeds to step **914** to set a different play mode when the user repeatedly presses the play mode set key, thus selecting the different play modes shown in Table 4.

TABLE 4

REPEAT	Repeating the present music piece until pressing the stop key
REPEAT ALL	Repeating the presently ordered sequence of music pieces until pressing the stop key
SHUFFLE	Shuffled playing of selected music pieces
REPEAT SHUFFLE	Repeating shuffled playing of selected music pieces until pressing the stop key
NORMAL	Ordinary playing mode

Each of the different playing modes shown in Table 4 may be selected by sequentially pressing the play mode set key No. 4.

If the view file key No. 5 is inputted through the keypad **104** in step **915**, the phone module **102** proceeds to step **916** to display the information of the MP3 files for a predetermined duration. Namely, if the view file key is pressed while playing a file, its information is displayed for a short duration. Alternatively, if the key is pressed while not playing the music, the information of the file presently held is displayed. If the left/right volume control key No. 6 is inputted through the keypad **104**, the phone module **102** proceeds to step **918** to display the presently set intensities of the right and left volumes of the earphone, in order to adjust them by means of proper selection keys such as up and down keys provided in the keypad. Returning to FIG. **5**B, finally detecting the end key input through the keypad **104** in step **919**, the phone module terminates the detailed MP3 play mode menu. Or otherwise, it waits to perform the steps **907** to **918** according to corresponding key inputs from the keypad **104**.

Additionally, the inventive MP3 play menu may include call receiving, displaying play state, file list, all deletion, memory state, download information, etc. The call receiving is to select connection or non-connection with a call during music playing. If non-connection is selected, the phone module does not generate a ring, called during music playing. However, this is not applied when the phone module does not play the music. The displaying play state is to display the volume intensity, playing time, playing sequence, etc. This, however, increases the consumption of the battery. The file list is to display the list of the ordered music pieces, and the size of a selected music piece, and makes it possible to delete a selected music piece. The all deletion is to all of the MP3 files stored in the memory. The memory state is to display the information of the total and used sizes of the memory. The download information is to display the names, dates and times of the MP3 files downloaded.

Thus, the present invention provides means for enabling the mobile station to store and reproduce the MP3 music files downloaded from a personal computer, so that the user may enjoy the MP3 file music without an additional MP3 player.

While the present invention has been described in connection with specific embodiments accompanied by the attached drawings, it will be readily apparent to those skilled in the art that various changes and modifications may be made thereto without departing from the spirit of the present invention.

What is claimed is:

1. A mobile phone for storing and reproducing digital audio data, comprising:

a keypad having a plurality of alphanumeric keys used for commanding a phone module to make mobile communication and to manipulate digital audio data;

a phone module for controlling the ordinary functions of said mobile phone, for enabling digital audio data to be downloaded from a personal computer to said mobile phone according to a key input through said key pad, and for enabling the sound of the digital audio data to be reproduced according to a key input through said key pad;

an RS-232 connector for connecting said phone module and personal computer to exchange digital audio data and other data; and

a digital audio data module for storing the digital audio data from said phone module for playing or stopping sound reproduced from the stored digital audio data, for rewinding the sound reproduced, and for fast-forwarding the sound reproduced.

2. The mobile phone as recited in claim **1**, wherein the digital audio data is MPEG (Moving Pictures Engineering Group) ½ Layer-3, or MP3, data and the digital audio data module is a MP3 module.

3. The mobile phone as recited in claim **2**, wherein said MP3 module comprises:

a memory device for storing MP3 digital audio data;

a Central Processing Unit (CPU) for controlling the MP3 audio data receiving from said phone module to be stored in or reproduced from said memory device;

an MP3 decoder for de-multiplexing MP3 audio data into control data and audio data for Huffman-decoding the audio data from run-length coded compressed signals to original length signals, for subjecting each sub-band of the signal to re-quantization and re-scaling according to said control data, for recovering the resulting data by an inverse discrete cosine transformation, and for inversely filtering each sub-band to finally obtain PCM (Pulse Code Modulation) data; and

a digital to analog (D/A) converter for converting PCM audio data delivered from said MP3 decoder into corresponding analog audio data.

4. A method for reproducing digital audio data in a mobile phone, comprising the steps of:

downloading digital audio data stored in a personal computer to a memory device of said mobile phone; and

selectively reproducing or playing said digital audio data stored in said memory device;

wherein said mobile phone includes a phone module for mobile communications and a digital audio data module, said digital audio data module for

storing said digital audio data to said memory device, and using said stored digital audio data to produce audible sounds, further wherein said the mobile phone can be connected to said personal computer by means of an adapter.

5. The method as recited in claim **4**, wherein the digital audio data is MPEG (Moving Pictures Engineering Group) ½ Layer-3, or MP3, data.

6. The method as recited in claim **4**, wherein the step of reproducing said digital audio data comprises the steps of:

selecting digital audio data play mode from a main menu;

displaying a detailed menu for playing said digital audio data; and

sequentially playing said digital audio data upon selecting a play key in said detailed menu.

7. The method as recited in claim **6**, further comprising the step of:

rewinding said digital audio data to a first stored position of said digital audio data upon selecting a rewind key in said detailed menu.

8. The method as recited in claim **4**, further comprising the step of:

replaying said digital audio data from a first stored position upon prolonged pressing of a rewind key during playing.

9. The method as recited in claim **4**, further comprising the step of:

moving a playing position to a starting point of said digital audio data next to that presently being played upon a short press of a fast forward key during playing.

10. The method as recited in claim **6**, further comprising the step of:

fast-forwarding a playing position to a starting point of a next digital audio data upon a short press of a fast forward key in said detailed menu.

11. The method as recited in claim **4**, further comprising the step of:

fast-forwarding a playing position to a starting point of a last digital audio data.

* * * * *

Exhibit 5

US007450114B2

(12) **United States Patent**
Anwar

(10) **Patent No.: US 7,450,114 B2**
(45) **Date of Patent: Nov. 11, 2008**

(54) **USER INTERFACE SYSTEMS AND METHODS FOR MANIPULATING AND VIEWING DIGITAL DOCUMENTS**

(75) Inventor: **Majid Anwar**, Glasgow (GB)

(73) Assignee: **Picsel (Research) Limited**, Glasgow (GB)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 738 days.

(21) Appl. No.: **09/835,458**

(22) Filed: **Apr. 16, 2001**

(65) **Prior Publication Data**

US 2002/0011990 A1 Jan. 31, 2002

(30) **Foreign Application Priority Data**

Apr. 14, 2000 (GB) 0009129.8

(51) **Int. Cl.**
G06F 3/033 (2006.01)
G06F 3/041 (2006.01)

(52) **U.S. Cl.** **345/179**; 345/173; 178/18.01

(58) **Field of Classification Search** 345/156–184; 178/18.01, 18.03, 19.01, 20.01
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,839,634	A	*	6/1989	More et al.	345/173
5,083,262	A		1/1992	Haff	
5,369,735	A	*	11/1994	Thier et al.	345/423
5,390,320	A		2/1995	Smithline	
5,463,725	A	*	10/1995	Henckel et al.	715/776
5,530,865	A		6/1996	Owens et al.	
5,615,384	A		3/1997	Allard et al.	
5,748,185	A		5/1998	Stephan et al.	
5,754,348	A		5/1998	Soohoo	
5,790,114	A		8/1998	Geaghan et al.	

(Continued)

FOREIGN PATENT DOCUMENTS

EP	0 438 194	A	7/1991
EP	0 465 250		1/1992
EP	0 479 496	A	4/1992
EP	0 513 584	A	11/1992
EP	529 121	A	3/1993

(Continued)

OTHER PUBLICATIONS

Blumberg, et. al.., "Visual Realism and Interactivity for the Internet", Proceedings of IEEE Compcon, 1997, San Jose, Feb. 23-26, 1997, Los Alamitos, IEEE Comp. Soc. Press, Feb. 1997, pp. 269-273.

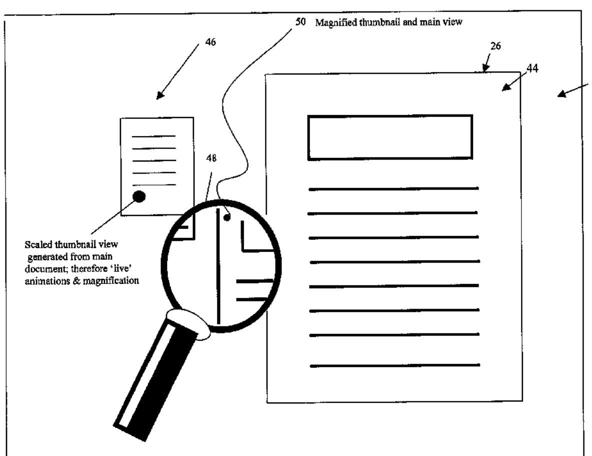
(Continued)

Primary Examiner—Sumati Lefkowitz
Assistant Examiner—Srilakshmi K Kumar
(74) *Attorney, Agent, or Firm*—Ropes & Gray LLP

(57) **ABSTRACT**

Systems, including handheld computing devices that include system code stored within the memory and adapted to be executed by the processor. The system code can process an input byte stream that is representative of contents to be displayed on the touch sensitive display and can generate a content document file representative of an internal representation of the content. A tool document file may also be stored in the memory and may provide an internal representation of a document providing an image that is representative of a graphical tool. Associated with a tool document can be tool code that is capable of processing the content document file to create an internal representation of the contents that presents the content in a manner that achieves a display effect associated with the tool. The device will also include parsing code that processes the content document file, the tool document file, and the processed internal representation to generate a screen document for display on the touch sensitive display in a manner that portrays the display effect.

5 Claims, 11 Drawing Sheets



U.S. PATENT DOCUMENTS

5,810,680	A	9/1998	Lobb et al.	
5,867,166	A	2/1999	Schick et al.	
5,899,988	A	5/1999	Depledge et al.	
5,909,207	A *	6/1999	Ho	345/156
5,910,805	A	6/1999	Haschart et al.	
5,911,066	A	6/1999	Atkinson et al.	
5,945,979	A	8/1999	Rutledge et al.	
6,034,700	A	3/2000	Nickell et al.	
6,121,960	A	9/2000	Carroll et al.	
6,310,610	B1 *	10/2001	Beaton et al.	345/173
6,411,274	B2 *	6/2002	Watanabe et al.	345/684
6,466,203	B2	10/2002	Van Ee	
6,525,749	B1 *	2/2003	Moran et al.	715/863
6,663,823	B2	12/2003	Kobayashi et al.	
6,717,573	B1 *	4/2004	Shahoian et al.	345/161

FOREIGN PATENT DOCUMENTS

EP	0 650 144	A1	4/1995
EP	0 651 543		5/1995
EP	0 753 832	A	1/1997
EP	0 764 918	A	3/1997
EP	0 860 769	A	8/1998
EP	0 949 571	A2	10/1999
EP	0 980 045		2/2000
GB	2 313 277	A	11/1997
WO	WO 97/34240		9/1997
WO	WO-97/36225		10/1997
WO	WO 98/10356		3/1998
WO	WO 98 37506	A	8/1998
WO	WO-98/56151		12/1998
WO	WO 99 10840	A	3/1999
WO	WO 00 10372	A	3/2000

OTHER PUBLICATIONS

Rowe, J.H., "Metafiles and Computer Graphics", Computers and Graphics, Pergamon Press, Ltd. Oxford, Great Britain, vol. 10, No. 2, 1986, pp. 103-106.

Anon., "Electronic Magnifying Glasses," IBM Technical Disclosure Bulletin, vol. 37, No. 3, 353-54 (Mar. 1994).

Bohmer, S. et al., "Word fur Windows 6.0", 1994, pp. 38-43, BHV-Verlag, Korschenbroich, Germany.

Office Action dated Jan. 23, 2008 U.S. Appl. No. 10/853,495.

Final Office Action dated Feb. 27, 2008 for U.S. Appl. No. 10/813,285.

Preece et al. "Human Computer Interaction," pp. 211-236 and 714-715, ISBN 0-201-62769-8, Addison Wesley (1994).

Kramer, A. "Classifying Two Dimensional Gestures in Interactive Systems," Proc. Of Intl. Gesture Workshop, 9/17-19/97, published as Lecture Notes In Artificial Intelligence vol. No. 1371, pp. 37-48, ISBN 3-540-64424-5, Springer-Verlag (1998).

* cited by examiner

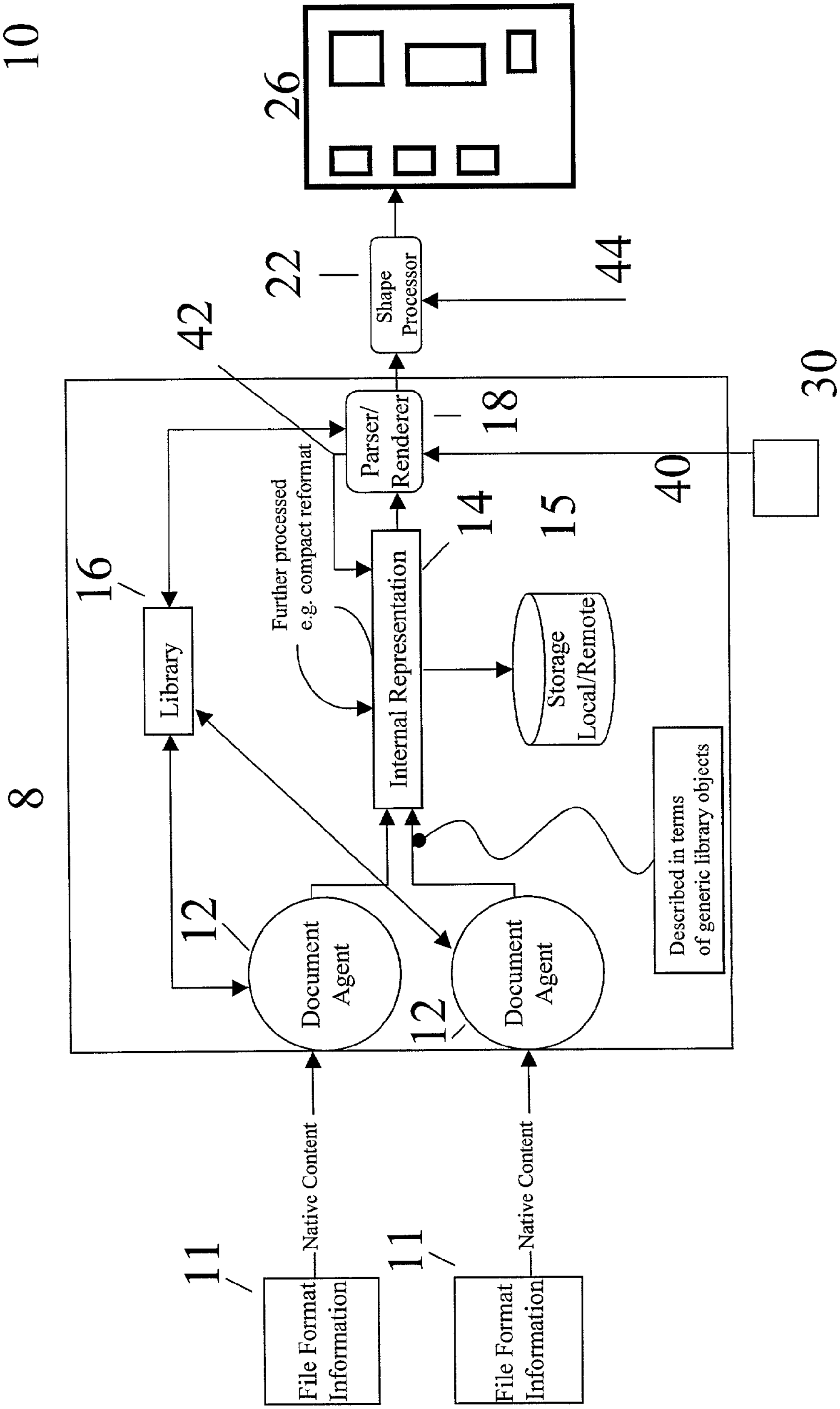

10
8
File Format Information
File Format Information
Native Content
Native Content
11
11
12
12
Document Agent
Document Agent
16
Library
Further processed e.g. compact reformat
Internal Representation
14
Storage Local/Remote
15
Described in terms of generic library objects
Parser/ Renderer
18
40
42
30
Shape Processor
22
44
26

Figure 1

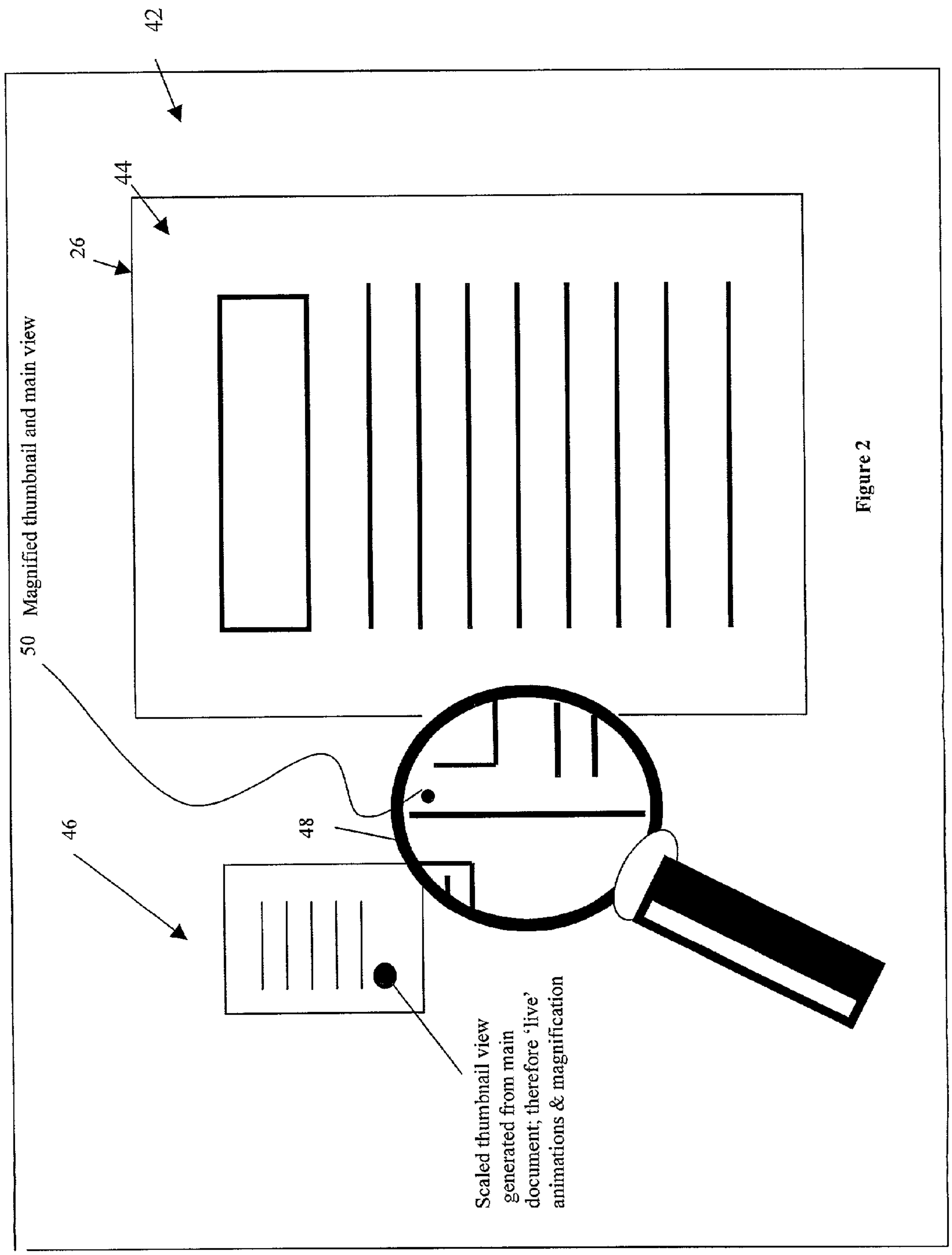

42
44
26
Magnified thumbnail and main view
50
46
48
Scaled thumbnail view generated from main document; therefore 'live' animations & magnification

Figure 2

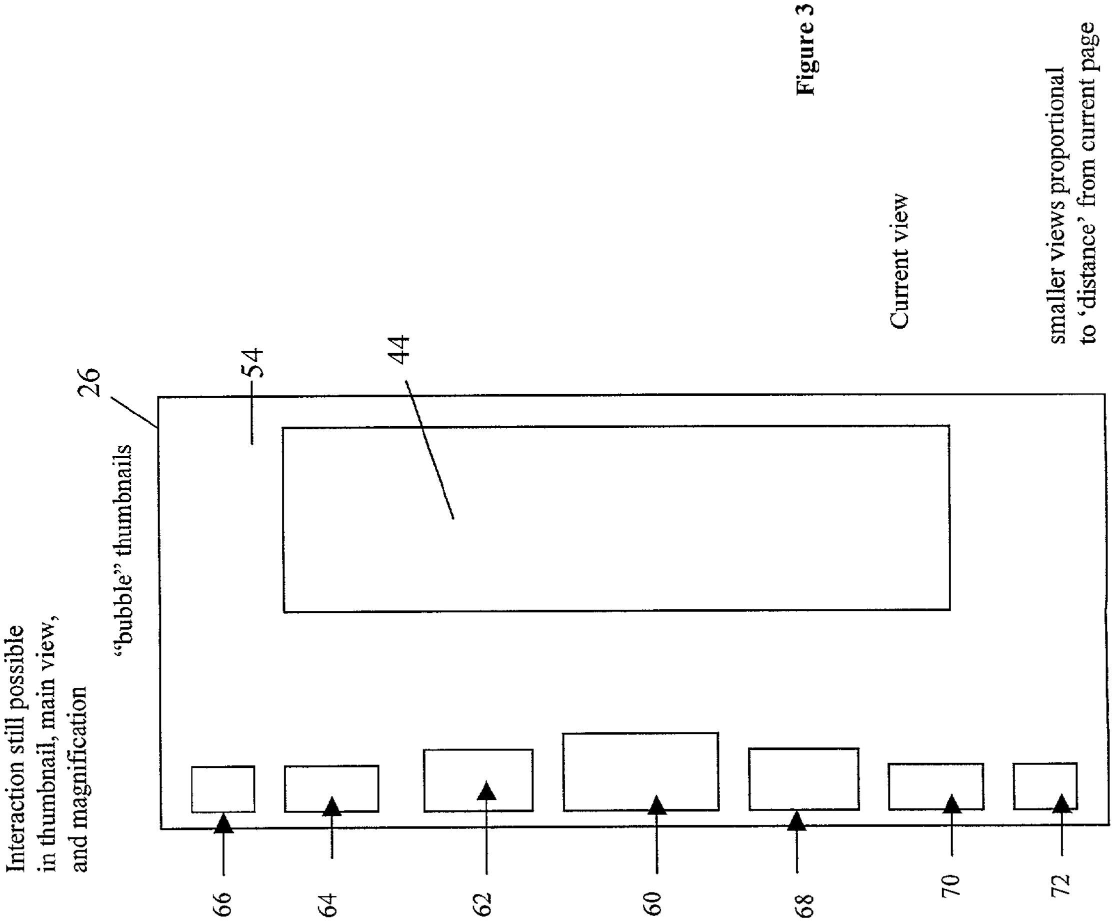

Interaction still possible
in thumbnail, main view,
and magnification
"bubble" thumbnails
26
54
44
Current view
smaller views proportional
to 'distance' from current page
66
64
62
60
68
70
72

Figure 3

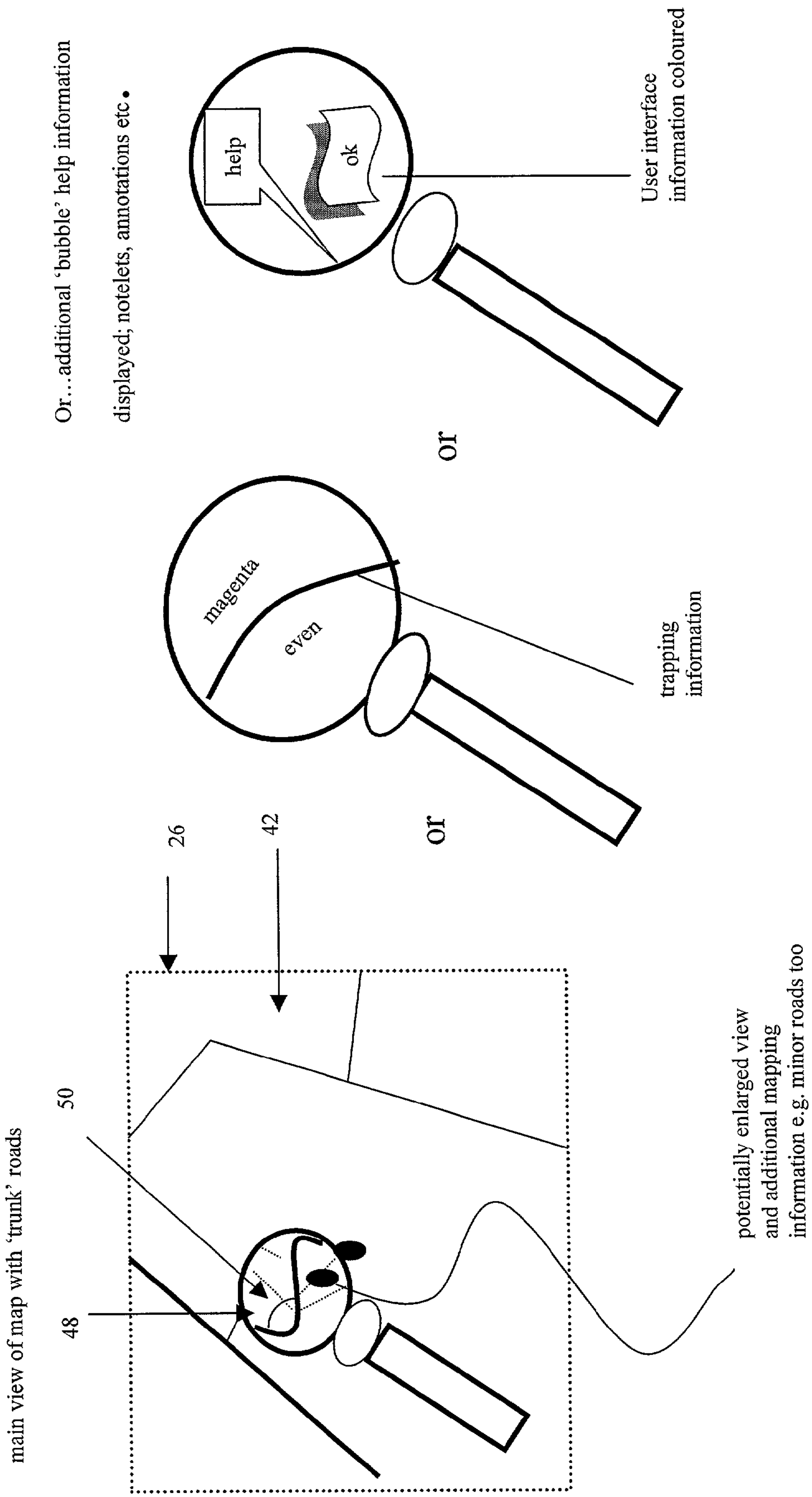

main view of map with 'trunk' roads
48
50
26
42
potentially enlarged view
and additional mapping
information e.g. minor roads too
or
magenta
even
trapping
information
or
help
ok
User interface
information coloured
Or…additional 'bubble' help information
displayed; notelets, annotations etc.

Figure 4

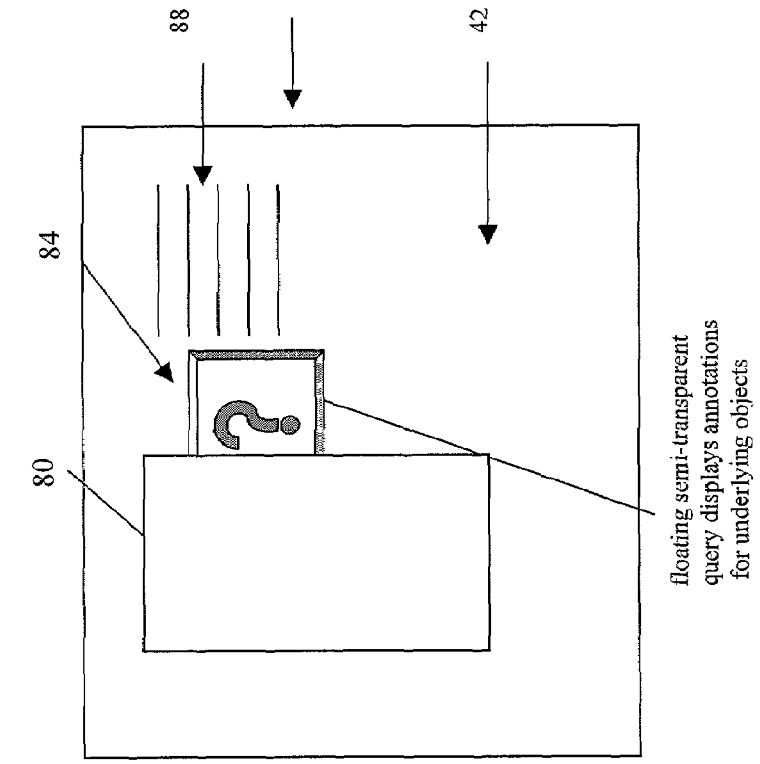


Figure 6

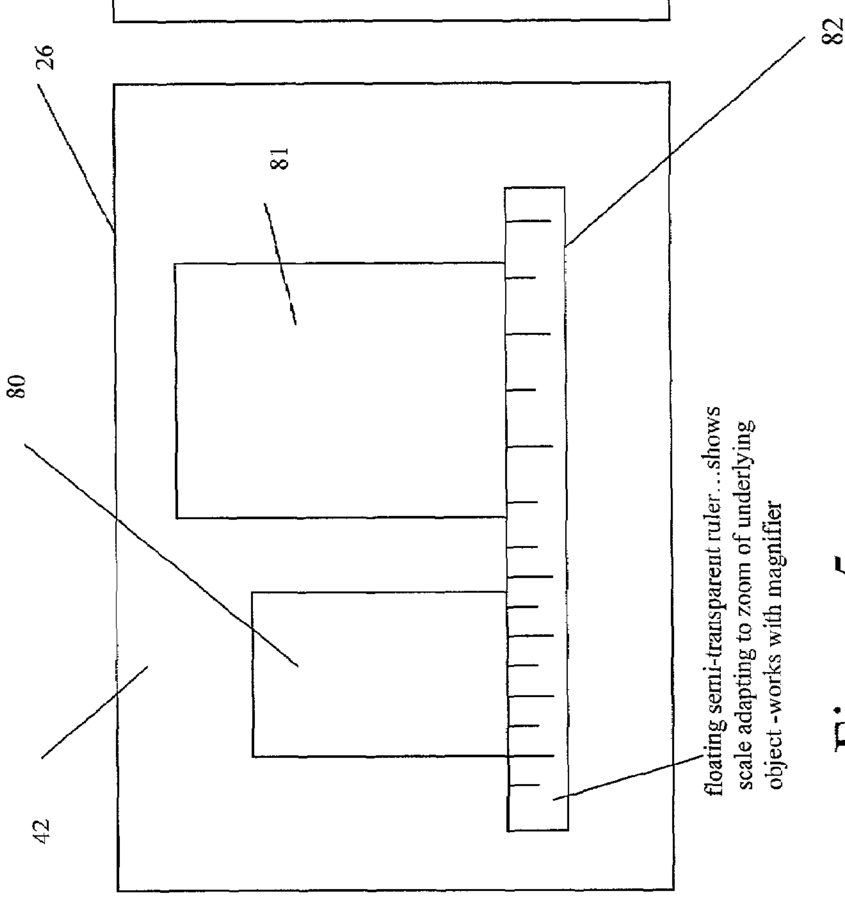


Figure 5

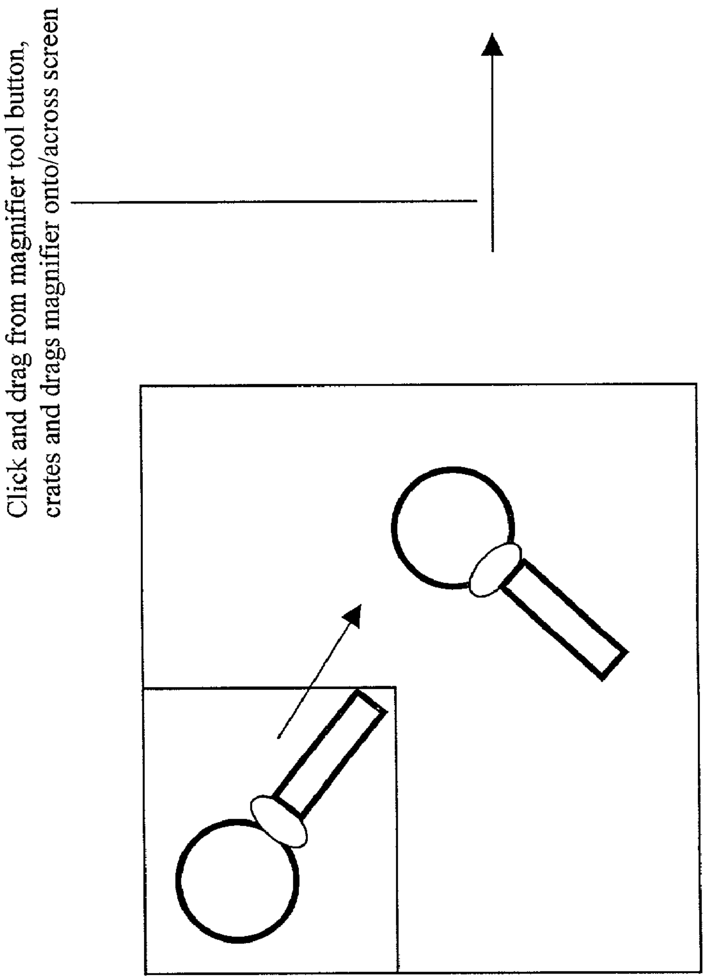


Figure 7A

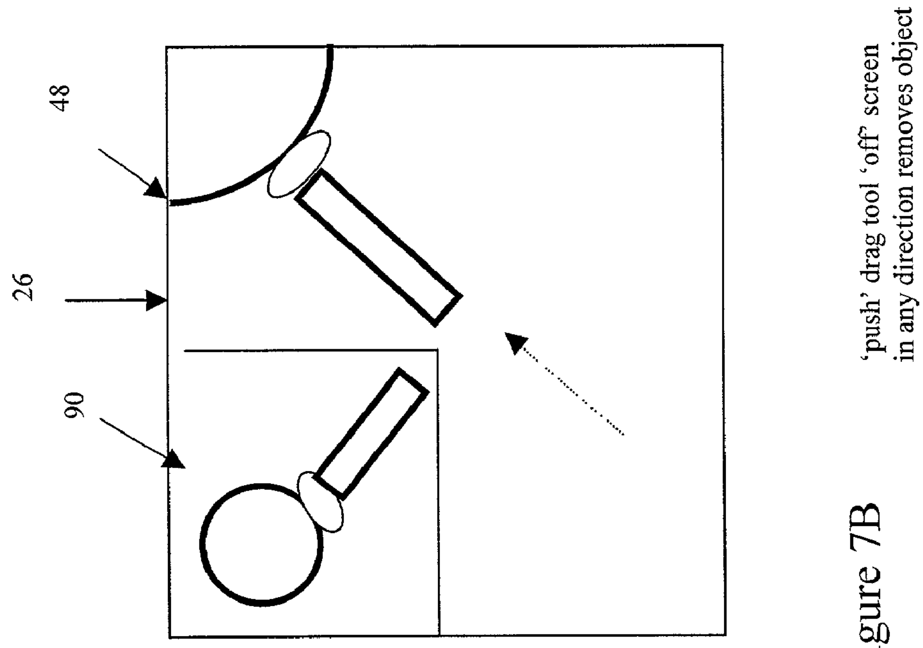


Figure 7B

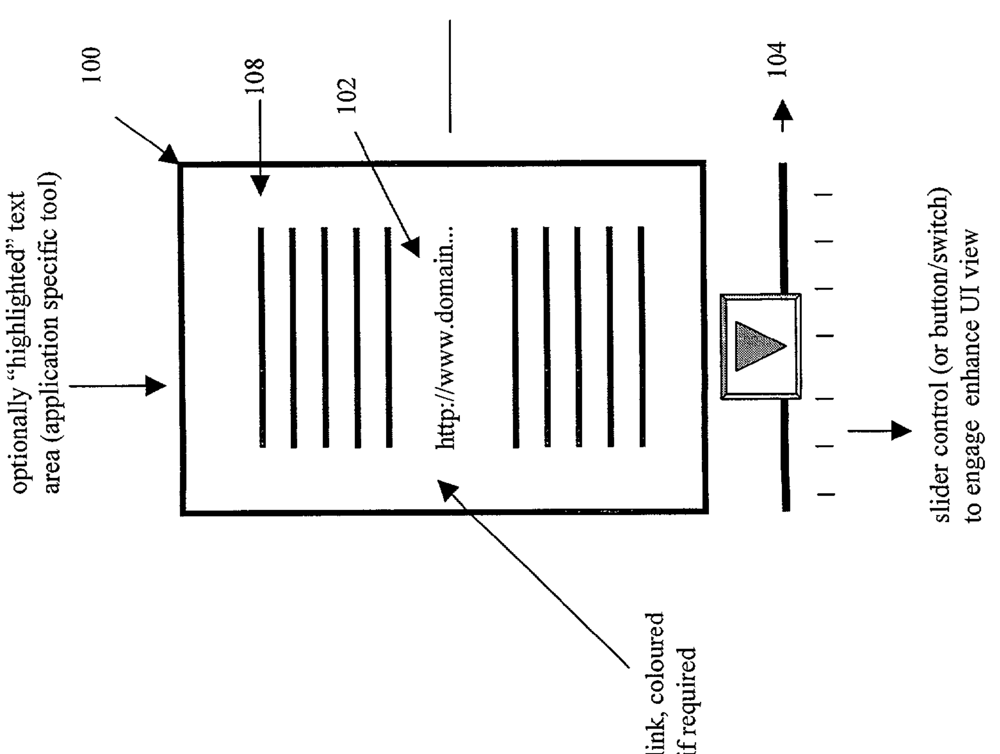


Figure 8A

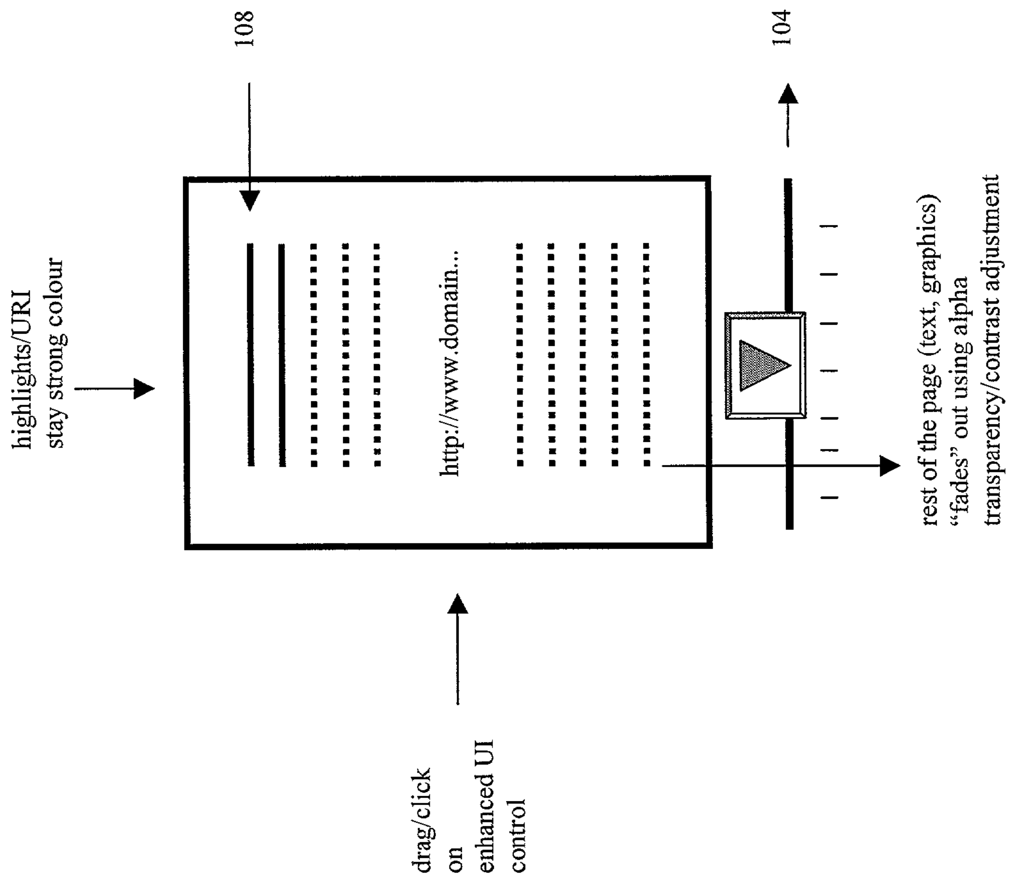


Figure 8B

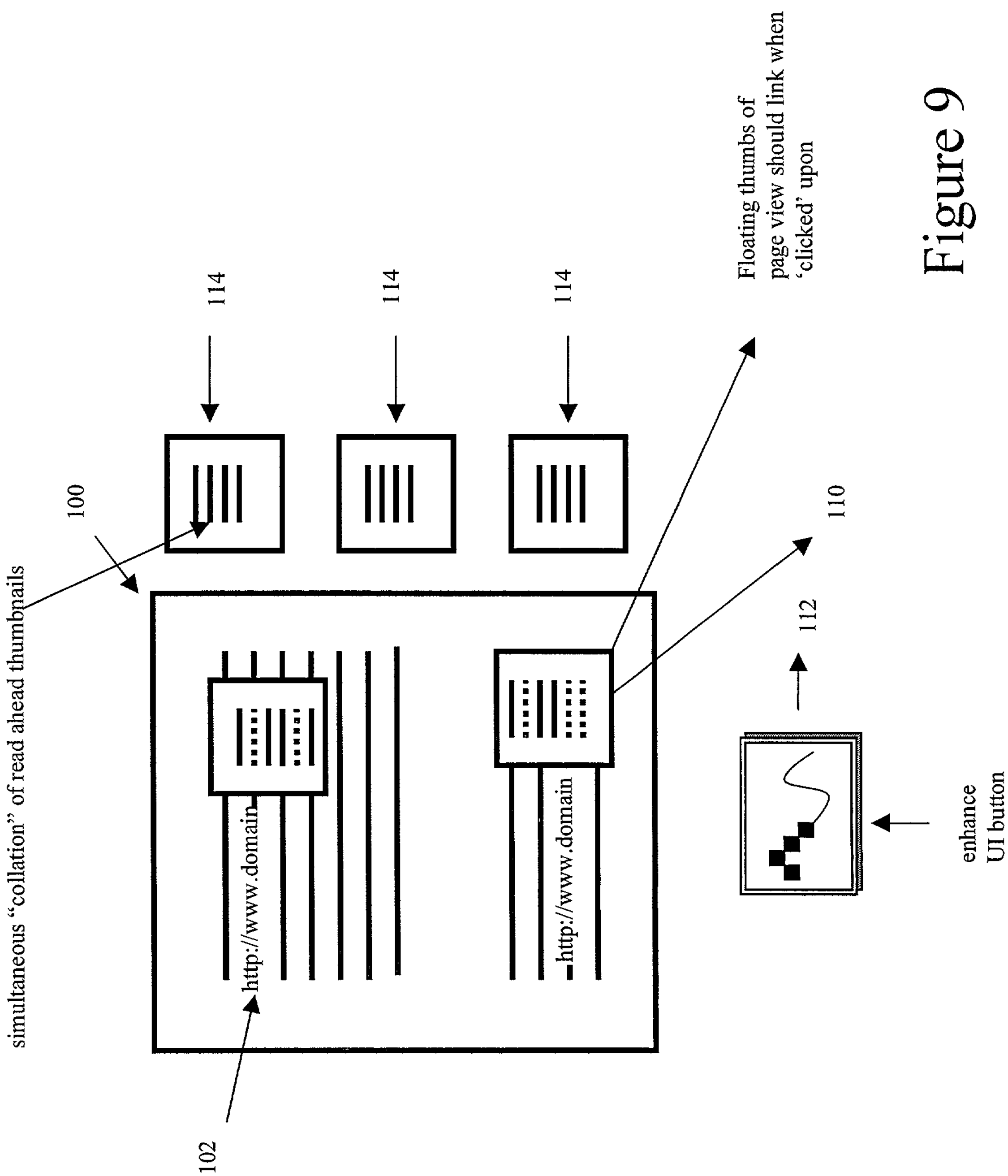

simultaneous "collation" of read ahead thumbnails
100
114
114
114
Floating thumbs of page view should link when 'clicked' upon
110
112
http://www.domain
http://www.domain
102
enhance UI button

Figure 9

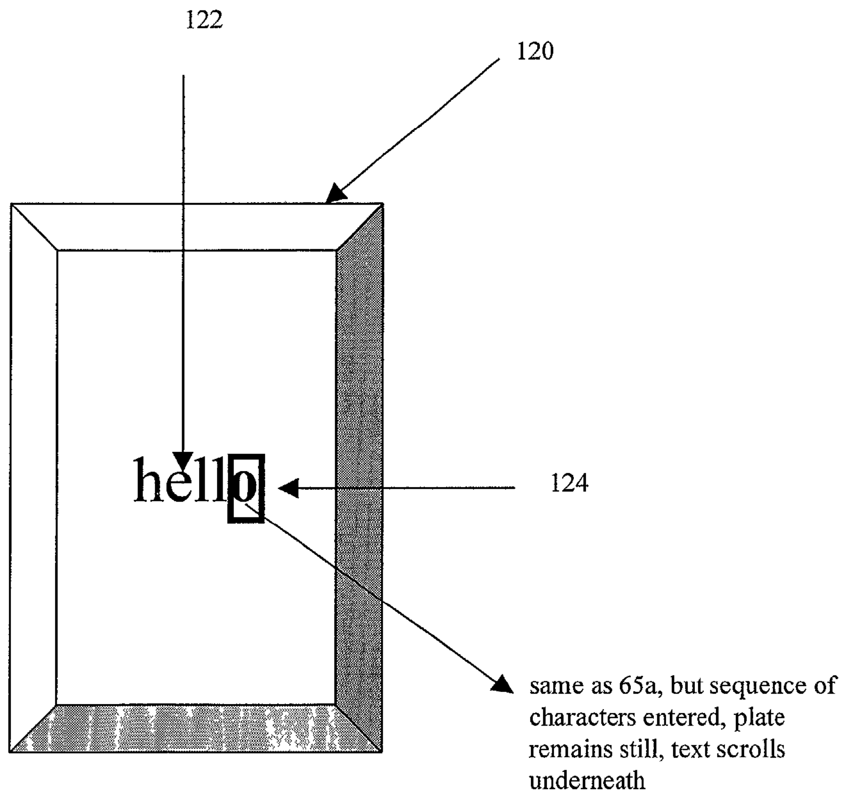


Figure 10

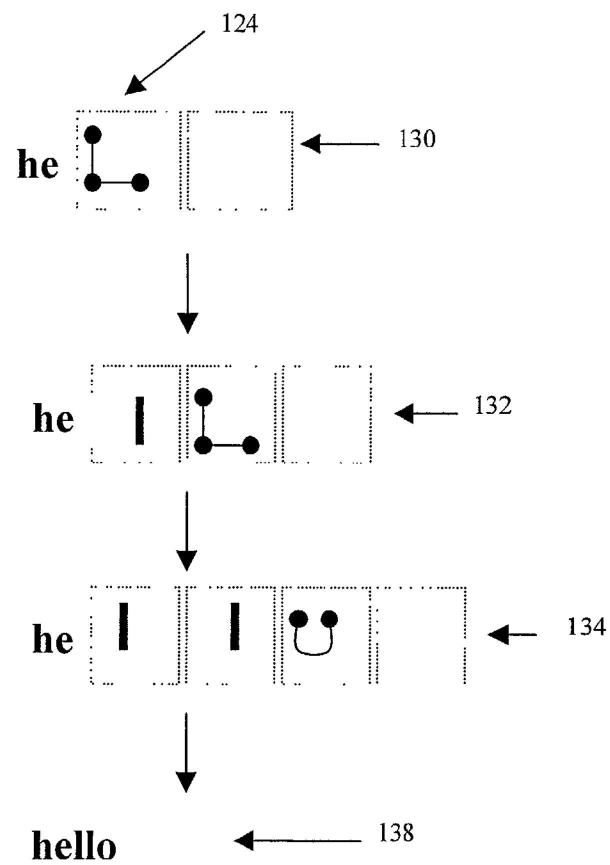
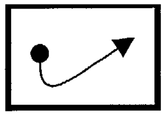
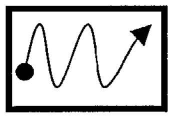


Figure 11



= OK

Figure 12A

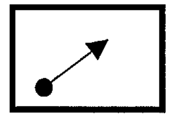


= Delete

Figure 12B

= home page
or
start over

Figure 12C



= wipe
or
no

Figure 12D

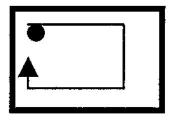
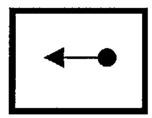
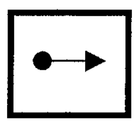


Figure 12E



= next

Figure 12F



= previous

Figure 12G

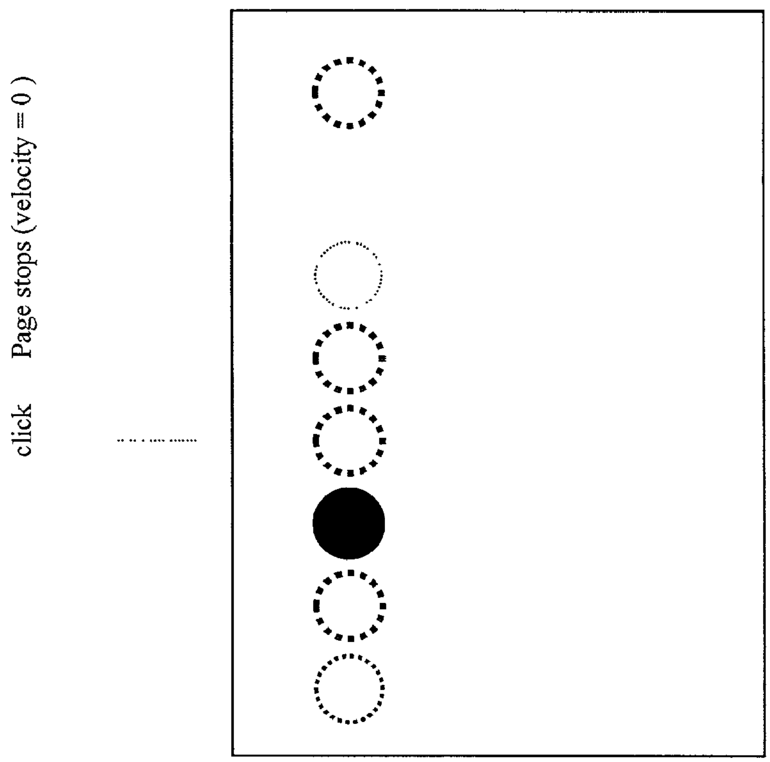


Figure 13A

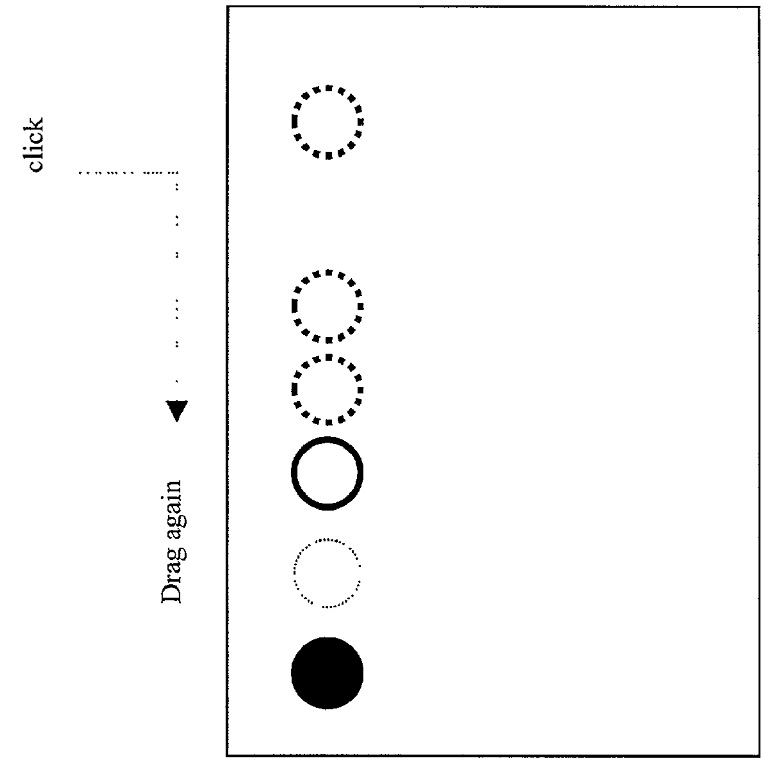


Figure 13B

USER INTERFACE SYSTEMS AND METHODS FOR MANIPULATING AND VIEWING DIGITAL DOCUMENTS

REFERENCE TO RELATED APPLICATIONS

The systems and methods described herein relate to earlier filed British patent application No. 0009129.8 and earlier filed U.S. patent application Ser. No. 09/703,502, as well as the U.S. patent application entitled, Systems and Methods for Digital Document Processing, filed even date herewith, all of which name Majid Anwar as an inventor, and the contents of which are herein incorporated by reference.

FIELD OF THE INVENTION

The systems and methods described herein relate to systems and methods for viewing and manipulating a display of digital documents, and more particularly to user interface systems and methods that allow a user to manipulate and view a digital document presented on a display, such as the display of a hand held electronic device, such as a computer, mobile communicator or phone; or a display device associated with a tactile commander.

BACKGROUND OF THE INVENTION

Today, there is a strong effort to build mobile and handheld computing devices that easily allow users to view documents, email, video presentations, and other forms of content. To achieve this convergence, engineers and scientists have developed systems including the systems described in the above-referenced U.S. patent application entitled, Systems and Methods for Processing Digital Documents, the contents of which are hereby incorporated by reference. As described therein, digital content, whether a document, audio visual presentation, or some other type of content, is processed by a software system operating on a handheld device, mobile device, or some other platform, and translated into a uniform internal representation that can be processed and manipulated by the software system so that a display of different types of content can be generated by the system and presented on the screen display of the respective device.

These systems, as well as other handheld and mobile computing systems such as the Palm Pilot, Compaq Ipaq, and mobile phones, are therefore capable of providing a display of content to a user. However, these handheld and mobile systems are typically limited to simple input devices such as small and limited keyboards, commonly the keyboards present on a cellular phone, or by small touch screen systems, such as the touch screens provided with the Palm computing device. Therefore, although these systems are capable of presenting content to a user wherein the content can be quite complex, these systems have limited capacity to allow a user to manipulate the display of that content, such as by paging through the different pages of a document or selecting different portions of a document. Therefore, although these handheld and portable systems may be quite useful, there are limitations to their use based in part on the user interfaces available for supporting the manipulation and viewing of the content presented on these devices.

Consequently, there is a need in the art for systems and methods that provide improved user interface tools that make it more facile to manipulate and view content presented by a handheld or portable device.

Additionally, there is a need in the art for user interface tools that allow for the manipulation of content, when that content is separated from its native application program.

SUMMARY OF THE INVENTION

The systems and methods described herein provide advanced user interface tools that allow a user to more easily manipulate and view content presented on a mobile device or handheld device. In one particular embodiment, the systems and methods described herein provide a graphical user interface that exhibits a touch and feel user interface experience. More specifically, the systems and methods described herein include hand-held or mobile computer devices having a system for simulating tactile control over a document which may be viewed on the device itself or through a remote instruction or remote instruction or remote display on another unit. These systems may include a housing which supports a processor, memory, and a touch-sensitive display (or display with remote touch-sensitive control), system code stored within the memory and adapted to be executed by the processor. The system code may generate or provide a digital representation of a document, wherein the digital representation may include data content and a page structure representative of the page layout of the document. Thus, in certain applications the rendered image can include the content of the document as well as the layout of the document, thereby providing an image of what the document physically looks like. The system may also include a rendering engine that may include a parser and a renderer for rendering at least a portion of the page layout of the digital representation on the touch-sensitive display. A screen monitor can monitor the touch-sensitive screen for detecting movement across a surface of the touch sensitive screen, and an interface process can process the detected movement to detect a motion representative of a command to alter the page structure of the digital representation. A navigation module may be responsive to the interface process and can change the rendered portion of the page layout. Thus, by altering the rendered portion of the page layout, the system allows a user to navigate through the digital representation of the document. Although the systems and methods of the invention will have applicability and value when used in other applications and on other types of systems, for purposes of illustration, the invention will be described with reference to those applications where the systems facilitate the navigation of documents presented on a hand-held computing device.

More particularly, the systems and methods described herein provide, among other things, computer devices having a system for simulating tactile control over a document. In one embodiment, these systems comprise a processor, memory, and a display; system code stored within the memory and adapted to be executed by the processor, the system code providing a digital representation of a document including data content and a page structure representative of a page layout of the document; a rendering engine for rendering at least a portion of the page layout of the digital representation on the display; a screen monitor for monitoring the screen to detect movement of an object across an image presented on the display; an interface process for processing the detected movement to detect a motion representative of a command to alter the rendered page structure of the digital representation, and a navigation module responsive to the interface process for changing the rendered portion of the page layout, wherein altering the rendered portion of the page layout allows a user to navigate through the digital representation of the document.

These computer devices can include touch-sensitive displays where the screen monitor monitors a touch-sensitive screen for detecting movement across a surface of the touch sensitive display, as well as computer displays capable of

depicting a cursor moving across a screen of the display, and wherein the screen monitor detects movement of the cursor across a surface of the display. The processor, memory, screen monitor and a display may be arranged as a data processing platform useful with a plurality of applications and devices including hand-held computers, telephones, mobile data terminal, a set top box, an embedded processor, a notebook computer, a computer workstation, a printer, a copier and a facsimile machine.

In certain optional embodiments, the computer device may also include a velocity detector for determining a velocity vector associated with motion detected across the surface of the touch-sensitive display, as well as means for applying a velocity characteristic to a document within a display.

Additionally, these computer devices can have an interface processes that make it more easy to navigate through a document or a collection of documents and other content. These interface processes can include a page-flip detector for detecting a motion across the surface of the touch-screen at a location presenting a portion of the page layout graphically representative of a corner of a document. The page-flip detector can render a portion of the page layout representative of a page adjacent a currently rendered page. Similarly, the device can include a page curl detector for rendering a portion of the page layout representative of a portion of a page adjacent a currently rendered page. Additionally, the interface process can include a gesturing process for detecting a predefined movement representative of a command for selecting a portion of the page layout to be rendered, or for altering data content of the digital representation of the document. Still further interface controls include processes for controlling a transparency characteristic of a document presented on the display and for controlling a transparency characteristic of selected portions of the document for adjusting visibility of the selected portions relative to other portions of the document. Other interface processes can provide tools, including tools representative of a magnifying tool, a ruler, a text entry cursor, a thumbnail navigation column, a thumbnail view of linked content and a query tool.

In still other aspects, the invention provides computer devices, and related processes, having a context sensitive graphical interface tool. These devices may comprise a processor, memory, and a touch-sensitive display; a content document file stored in the memory and being representative of an internal representation of the content; a tool document file stored in the memory and providing an internal representation of a document providing an image that is representative of the graphical interface tool; tool code capable of running of the processor and being associated with the tool document file and capable of processing the content document file to create an internal representation of the content that when rendered presents the content in a manner that achieves a display effect associated with the tool; parsing code that processes the content document file, the tool document file, and the processed internal representation to generate a screen document for display, and; interface code capable of running on the processor for allowing a user to arrange the image of the graphical interface tool into a selected contextual relationship over the rendered content and for directing the tool code to process a portion of the content document file associated with the selected position.

The contextual relationship between the graphical interface tool and the rendered content can vary depending upon the application, and may for example be selected from the group consisting of the relative position of the graphical interface tool and the rendered content, the time at which the graphical interface tool acts on the rendered content, and the state of the rendered content. These devices are flexible and may be embodied in different forms and devices, including, but not being limited to, a hand-held computer, a telephone, mobile data terminals, a set top box, an embedded processor, a notebook computer, a computer workstation, a printer, a copier, and a facsimile machines, as well as in car systems, and domestic devices such as audio players, microwaves, refrigerators, and washing machines.

However, it will be understood by those of ordinary skill in the art that these interface tools may be employed in other applications including applications wherein content is displayed on a conventional computer workstation that includes typical input tools such as a standard keyboard and a mouse. Additionally, it will be understood that the systems and methods described herein also provide useful tools for providing interfaces for embedded display systems, such as embedded visual displays employed as output devices. Examples of such embedded display systems can include cellular phones, copy machines that include a visual touch screen display that allows a user to select different options for performing a copy job and which may also present images to the user of the documents being copied. Other examples may include fax machines wherein visual displays are provided to a user to allow a user to view a depiction of an incoming fax. Other embodiments and applications of the user interface systems and methods described herein will be apparent to those of ordinary skill in the art.

More particularly, the systems and methods described herein provide user interface tools that allow a user to manipulate content displayed on a screen. In particular, the systems and methods described herein provide software systems that create an abstraction layer for information that is to be presented on a display. This abstraction layer includes a document object wherein a document object contains information, or content, that is to be displayed on a screen. In one implementation, all information displayed on a screen is treated as one document. Thus, at the highest level, the entire contents of a screen is understood as one document object. Further to this embodiment, it will be understood that a document object may contain other document objects, each of which may contain a subset of the content displayed to the user. Thus, at the screen level, all information displayed will be understood as a single document wherein items, such as web pages, streamed video, and graphical icons, presented on the screen are each understood separately as document objects contained within the high level screen document object. Therefore, all content displayed on a screen is treated abstractly as a document, and this paradigm holds whether the content being displayed is information representative of a page of text or information representative of a user interface tool or window/desktop furniture. Accordingly, the user interface systems and methods described herein provide user interface tools and functionality for allowing a user to manipulate document objects presented on a screen display.

Additionally, the systems and methods described herein provide, in one embodiment, a handheld computing device that comprises a housing which supports a processor, memory, and a touch sensitive display. Further, the computing device may include system code stored within the memory and adapted to be executed by the processor. The system code can process an input byte stream that is representative of content to be displayed on the touch sensitive display and can generate a content document file representative of an internal representation of the content. A tool document file may also be stored in the memory and may provide an internal representation of a document providing an image that is representative of a graphical tool. Associated with a tool document can be tool code that is capable of processing the content document file to create an internal representation of the content that presents the content in a manner that achieves a display effect associated with the tool. The device may also include parsing code that processes the content document file, the tool document file, and the processed internal representation to generate a screen document for display on the touch sensitive display in a manner that portrays the display effect.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects and advantages of the invention will be appreciated more fully from the following further description thereof, with reference to the accompanying drawings wherein;

FIG. **1** provides a functional block diagram of one system according to the invention;

FIG. **2** depicts one example of a tool generated by a system such as the system depicted in FIG. **1**;

FIG. **3** depicts a graphical user interface tool that presents a plurality of thumbnail sketches for navigating through a document having a plurality of pages;

FIG. **4** depicts a magnifying graphical user interface tool according to the invention which provides additional information within a magnified area;

FIG. **5** depicts a semi-transparent and adaptively sizeable ruler graphical tool;

FIG. **6** depicts a transparent query marker graphical user interface tool;

FIG. **7** depicts a user interface mechanism for activating and deactivating a graphical tool;

FIGS. **8***a* and **8***b* depict a user interface tool for visually enhancing selected portions of a displayed document;

FIG. **9** depicts a further user interface tool according to the invention;

FIGS. **10** and **11** depict a text entry tool according to the invention;

FIGS. **12***a*-**12***g* depict a set of strokes for providing commands to a hand-held system; and

FIGS. **13**A-**13**B depict a user interface tool for scrolling through a document by applying a velocity characteristic to the document being displayed.

DETAILED DESCRIPTION OF THE ILLUSTRATED EMBODIMENTS

The systems and methods described herein include systems and methods for manipulating and viewing documents displayed on a viewing surface such as a computer terminal, a display screen, a printer, plotter, or any other output device suitable for creating a visual representation of human readable information. For the purpose of illustration, the systems and methods will be described with reference to certain exemplary embodiments, including hand held computer systems that include touch screen displays and which are capable of displaying an integrated view of content produced in different formats. In particular, the systems and methods described herein include graphical user interface tools that are capable of presenting tools that can be presented as content that will be integrated along with other content being displayed on the screen.

FIG. **1** depicts a system **10** according to the invention. The system **10** is shown as a functional block diagram of a computer device of the type that commonly includes a processor, a memory and a display. However, the system **10** may also be realized, in whole or in part as a software system comprising system code capable of executing on a processor to configure the processor as a system according to the invention. The depicted system **10** includes a computer process **8**, a plurality of source documents **11**, a tool document file **30**, a shape processor **22** and a video display **26**. The computer process **8** includes a plurality of document agents **12**, a library of generic data objects **16**, an internal representation file **14**, a memory buffer or file **15**, and a parser/renderer engine **18**.

In the depicted embodiment, the display **26** can present the images of a plurality of different documents. Each of the representative outputs appearing on display **26** is termed a document, and each of the depicted documents can be associated with one separate application program, such as Word, Netscape Navigator, Real Player, Adobe, Visio and other types of applications. It will be understood that the term document as used herein will encompass documents, streamed video, web pages, and any other form of data that can be processed and displayed by the computer process **8**.

The computer process **8** generates a single output display that includes within that display one or more of the documents. The collection of displayed documents represent the content generated by the application programs and this content is displayed within the program window generated by the computer process **8**. The program window for the computer process **8** may also include a set of icons representative of tools provided with the graphical user interface and capable of allowing a user to control the operation, in this case the display, of the documents appearing in the program window.

For the illustrated embodiment, the display **26** presents content representative of different data types in a single, integrated display. This is in contrast to the conventional approach of having each application program form its own display, which results in a presentation on the display device **26** that includes several program windows, typically one for each application program. Additionally, each different type of program window would include a different set of user interface tools for manipulating the content displayed in that window. Thus, the system depicted in FIG. **1** creates an integrated display that contains viewable images of documents of different types. This includes web pages that would normally be viewed in a browser, word documents that would normally be viewed in a viewer or word processing document, PDF documents that would normally be viewed in a vector graphic reader, and streaming video that would normally be viewed in a video player. Thus, the depicted system **10** separates the content of these documents from the underlying application program and presents them for display on the screen **26**.

To allow a user to manipulate the depicted documents, the system **10** depicted in FIG. **1** provides a set of tools that can be used to navigate through a collection of documents, whether it is a multi-page text document, the pages of a web site or a series of time varying images that make up a video presentation. To this end, as will be explained in greater detail below, the system **10** creates documents that are representative of tools and which may be displayed by the system **10** just as system **10** would display any other type of document. Thus the system **10** of the invention has the advantage of providing a consistent user interface, and only requiring knowledge of one set of tools for displaying and controlling the different documents.

As discussed above, each source document **11** is associated with a document agent **12** that is capable of translating the incoming document into an internal representation of the content of that source document **11**. To identify the appropriate document agent **12** to process a source document **11**, the system **10** of FIG. **1** includes an application dispatcher (not shown) that controls the interface between application programs and the system **10**. In one practice, an external application programming interface (API) communicates with the application dispatcher which passes data, calls the appropriate document agent **12**, or otherwise carries out a request made by an application program. To select the appropriate document agent **12** for a particular source document **11**, the application dispatcher advertises the source document **11** to all the loaded document agents **12**. These document agents **12** then respond with information regarding their particular suitability for translating the content of the published source document **11**. Once the document agents **12** have responded, the application dispatcher selects a document agent **12** and passes a pointer, such as a URI of the source document **11**, to the selected document agent **12**.

As shown in FIG. **1**, the document agent **12** employs the library **16** of standard object types to generate the internal representation **14**, which describes the content of the source document **11** in terms of a collection of document objects as defined in the library **16**, together with parameters defining the properties of specific instances of the various document objects within the document. The document object types employed in the internal representation **14** will typically include: text, bitmap graphics and vector graphics which may or may not be animated and which may be two- or three-dimensional: video, audio and a variety of types of interactive objects such as buttons and icons. Vector graphics document objects may be PostScript-like paths with specified fill and transparency. Text document objects may declare a region of stylized text.

Once documents are translated into an internal representation of document objects, these objects are passed to the parser/renderer **18**. The parser/renderer **18** generates a context-specific representation or "view" of the documents represented by the internal representation **14**. The required view may be of all the documents, a whole document or of parts of one or some of the documents. The parser/renderer **18** receives view control inputs which define the viewing context and any related temporal parameters of the specific document view which is to be generated. For example, the system **10** may be required to generate a zoomed view of part of a document, and then to pan or scroll the zoomed view to display adjacent portions of the document. The view control inputs are interpreted by the parser/renderer **18** to determine which parts of the internal representation are required for a particular view and how, when and for how long the view is to be displayed.

The context-specific representation/view is expressed in terms of primitive figures and parameters. Optionally, there may be a feedback path **42** between the parser/renderer **18** and the internal representation **14**, e.g. for the purpose of triggering an update of the content of the internal representation **14**, such as in the case where the source document **11** represented by the internal representation **14** comprises a time varying multi-frame animation.

Each source document **11** provides a digital representation of a document, such as a text document, a spread sheet or some other document. The document agent **12** creates an internal representation of that document. In one practice the created digital representation includes information that describes the page layout of the document, including information about page size, margins and other page layout information. The digital representation also includes information about the content of the source document, such as the text, figures, and other content information that appears in the document. Processes for translating a known file structure into another structure are known in the art, including systems that identify page structure and content information. Any of the suitable techniques for performing this operation may be practiced without departing from the scope of the invention.

The output from the parser/renderer **18** expresses the document in terms of primitive figures. For each document object, the representation from the parser/renderer **18** defines the object at least in terms of a physical, rectangular boundary box, the actual shape of the object bounded by the boundary box, the data content of the object, and its transparency. The shape processor **22** interprets the primitive object and converts it into an output frame format appropriate to the target output device **26**; e.g. a dot-map for a printer, vector instruction set for a plotter, or bitmap for a display device. An output control input **44** connects to the shape processor **22** and can deliver user interface control signals to generate an output suitable for a particular output device **26**. Thus, the parser/renderer **18** and the shape processor **22** can act as an engine that renders portions of the page layout and page content on the display **26**.

Additionally, FIG. **1** depicts a tool document file **30**. The tool document file **30** may be a computer data file that stores information representative of an image, wherein that image may represent a tool such as a magnifying glass, a cursor, a ruler, or any other type of tool. For the purpose of illustration, the system **10** depicted in FIG. **1** will now be described with reference to an example wherein the tool document file **30** includes data that is representative of a graphical image of a magnifying glass. The magnifying glass image will be associated with a function that allows a user to magnify the image of a document stored on the display **26** by passing the magnifying glass over the respective image. As will be described in greater detail hereafter, the magnifying glass can include a central lens portion wherein portions of a document that fall under the lens of the magnifying glass appear to the user to be magnified and therefore are presented in an enlarged format relative to the rest of the underlying document. Although the below example will be described primarily with reference to the magnifying glass tool, it will be obvious to those of ordinary skill in the art that other types of tools may be provided using the systems and methods described herein and all such tools will be understood to fall within the scope of the invention.

Turning to FIG. **2**, the operation of the magnifying glass tool can be seen. Specifically, FIG. **2** depicts the display **26** wherein the display **26** presents a screen document **42** that comprises a plurality of sub elements including the document **44**, the thumbnail document **46**, the magnifying glass document **48**, and the lens document **50**. The display **26** presents the screen **42** as a single integrated document that contains sub documents **44** through **50**. The content provided for creating the screen **42** can come from one or a plurality of source documents **11** whose content appears as the document **44** and thumbnail document **46**. The screen document **42** also comprises content provided by the tool document file **30** that, in this example, contains data according to an internal representation data format wherein that data represents the image of the magnifying glass **48**. Additionally, the tool document file **30** may contain a portal object that creates a further document by processing the appropriate portion of the screen document **42** to present that content in an enlarged format to appear as the magnified document **50** appearing within the lens of the magnifying glass **48**. Thus, the document appearing within the lens **50** is derived from the underlying document, and therefore this derived document changes according to the context in which the magnifying glass tool **48** is employed. Accordingly, the specific behavior for the tool can vary depending on the context of its use. For example, a magnifying glass tool may be associated with tool code that processes differently the content of a content document having map data than a content document having text. For example, with a map, the magnifying glass tool may process the associated content document to render handles within the associated document structure that are tagged as only to be shown within a view created by a magnifying glass. Thus the derived document presented within the magnifying glass tool **48** can include additional information, such as street names, tourist sites, public transportation locations, notations or other information. In this operation, the magnifying glass tool responds to the context of the application, which is the rendering of a view of a map. In other applications, where the magnifying glass tool is employed on text, the behavior of the tool may result in changes to the color or style of the text, or could result in the presentation of text editing tools and user interface controls, such as control buttons, pull down menus, annotation information, text bubbles, or other types of information.

Accordingly, the screen document **42** is an integration and aggregation of information contained within a source document **11** and a tool document file **30**. An application program associated with the tool document file **30** can process the appropriate content to create the enlarged view **50**. The magnifying tool **48** and the associated source code are capable of identifying that portion of the screen document **42** that is to be presented in an enlarged format to create the enlarged view **50**. The tool code is further capable of processing the selected content to create the enlarged view **50** and clip that enlarged view within the lens area of the magnifying glass **48** to achieve the display affect of an enlarged region of the screen **26**. Thus, the tool document and the source document **11** are in the same internal representation, and thus can be merged into the screen document **42**, that can be rendered by the parser/renderer **18**.

In one embodiment, the graphical tool **50** may be moved over the screen by dragging with a cursor, or if a touch-sensitive screen is present, by dragging a stylus or some other pointer across the screen of the display. To process this movement, the display **26** may include a screen monitoring process for monitoring the screen of the display **26** to detect movement of a cursor, stylus or some other pointer across the images of the documents presented on the screen. Such screen monitoring processes are known in the art and any suitable process may be employed. The monitor process, therefore allows a user sense of tactile control over the visual representation of the document **44**. The movements detected by the screen monitor process may be passed to an interface process that processes the detected motion to detect a motion representative of a known command. The interface process may be a separate process or may be part of the screen monitor process, as is common in the art. As the interface module detects commands to move the tool **50**, a navigation module can create input signals that direct the parser/render **18** to create a new display for presentation to the user, where in the display will show the tool **50** repositioned as the user wanted.

Accordingly, the system depicted in FIG. **1** is capable of providing a graphical user interface tool that may be integrated into a screen display that represents a single document which contains a plurality of sub documents, some of which sub documents include the graphical tools themselves. The power of this approach allows for the development of novel graphical user interface tools that allow a user to manipulate and view a document on a display and to simulate tactile control over the depicted documents. These systems and methods are particularly well suited to use on hand held and mobile computing platforms where traditional input tools are lacking. Additional graphical user interface tools that may be provided by the systems and methods described herein include the bubble thumbnail graphical tool depicted in FIG. **3**. Specifically, FIG. **3** depicts the screen display **26** that includes a screen document **52** which comprises a plurality of sub documents including the document **44** and the thumbnail documents **60** through **72**. As shown in FIG. **3**, the document **44** may be presented as a large document employing most of the viewing area of the display **26**. In this embodiment, the thumbnail documents **60** through **72** are arranged in a vertical column within the screen document **52** at a position adjacent the left side of the display **26**. The thumbnail documents **60** through **72** vary in size with the largest thumbnail document **60** being centrally positioned within the vertical array of thumbnail documents.

As further shown by FIG. **3** as documents in the vertical array increase in distance from the center document **60**, the documents decrease in size. The measure of distance from the center document may be representative of the distance in pages from the document **44**, or may be representative of some other measure of distance or difference, such as the amount of time that has passed since the document was last viewed, the difference in alphabetical order, or some other characteristic. Thus documents **62** and **68**, which are adjacent the central document **60** are somewhat smaller than the document **60**. Further documents **64** and **70** which are adjacent documents **62** and **68** respectively and further from document **60** are smaller than documents **64** and **68** still. The decrease in size of documents continues from documents **66** and **72**, each of which is still smaller. The impression created by the array of thumbnail documents **60** through **72** is employed to indicate that document **60**, the largest document, is representative of the document **44** being displayed within the largest viewing area of the screen document **52**. Documents **62** through **72** get smaller in proportion to the "distance" from the current viewing page **60**. Accordingly, the vertical column of thumbnail document **60** through **72** provide a navigation tool that a user can employ for selecting a document to appear within the large viewing area of the display **26**. Additionally, the user can select a document within the vertical array of thumbnails to choose a new document to appear within the viewing area. For example, in those applications where the screen display **26** is a touch sensitive screen display, the user may activate a new document to appear within the viewing area by touching the respective thumbnail document within the array of documents **60** through **72**. In those applications where the user is provided a keyboard, or a mouse, the user may employ that particular input device for selecting which of the documents within the array of documents that the user would like to appear within the viewing area. In an optional embodiment, the user can scroll through the thumbnails to find the document of interest. Optionally, scrolling through the thumbnail documents can result in the document **44** changing with the scrolling of the document. Alternatively, the scrolling of the thumbnail documents can occur independently from any changing of the document **44**, with the document **44** only changing when a new thumbnail document is selected.

Accordingly, as the systems and processes described herein may employ thumbnail images for queuing a user during navigation to generate thumbnail images, the systems and processes described herein can include any suitable thumbnail generator processes including those known in the art, including those thumbnail generations that generate live, or animated thumbnails.

FIG. **4** depicts a further embodiment of the systems and methods described herein wherein the magnifying tool, earlier shown in FIG. **2** is associated with tool code that results in information not earlier presented in a document to appear within the lens area of the magnifying glass object. More specifically, FIG. **4** depicts a display **26** that includes a screen document **42**, which in this view appears as a map. FIG. **4** further depicts the magnifying glass tool **48** that includes the lens area **50**. As shown in FIG. **4** the magnifying tool **48** is positioned over a portion of the map **42**. As described above, the tool code associated with the magnifying glass **48** is capable of presenting an enlarged view of the relevant portion of the screen document **42**. As additionally shown in FIG. **4** the magnified portion **50** also includes additional information. For example, in the mapping application depicted in FIG. **4**, the enlarged view **50** may include additional mapping information such as minor roads, locations of interest, or other information relevant to the content being magnified. Additionally however, the magnifying glass may be associated with tool code that changes that color of the information, or a portion of the information, within the viewing area **50**, or presents user interface information such as control buttons, pull down menus, annotation information, text bubbles, or other types of information. Accordingly, the specific behavior for the tool can vary depending the context of its use. For example, as described above the magnifying glass tool may be associated with tool code that processes differently the content of a content document having map data than a content document having text. Thus, the systems described herein provide context sensitive tools and processes.

FIGS. **5** and **6** depict further embodiments of graphical user interface tools that may be provided by the systems and methods described herein. In particular FIG. **5** depicts a screen **26** that includes a screen document **42** that comprises two documents **80** and **81**, and a ruler **82**. The two documents **80** and **81** are meant to represent similar types of documents, each being text documents printed on the same size paper, such as A4 paper. However, as the scale of the presentation of document **81** is larger than the scale of presentation for document **80**, FIG. **5** depicts the two documents **80** and **81** as pages of text, wherein one page is larger than the other. Thus, documents **80** and **81** are similar documents that have been rendered with different scaling factors. As depicted in FIG. **5** the ruler **82** may be a floating semi-transparent ruler that shows the scale of each document and which can adapt to the scale of the underlying object. This is depicted by the scale of the ruler **82** increasing in size as the length of the ruler travels from document **80** to document **81**. FIG. **5** portrays that the scale of the ruler **82** changes in proportion to the scale of the underlying documents. Thus, the ruler **82** provides a context sensitive user interface tool that is capable of adjusting the scale of the ruler in response to the presentation scale of the content. Turning to FIG. **6**, a further user interface tool, the floating semi-transparent query mark **84** is depicted wherein the query mark **84** may display annotations for the underlying object according FIG. **6** depicts that the display **26** includes a screen document **42** the comprises a document **80** and a floating semi-transparent query tool **84**. The query tool **84** when activated, either by dragging an image of the query tool onto the document, or by selecting a query tool icon already positioned over a document, will present text **88** that may include information representative of an annotation of the underlying document **80**.

FIG. **7a** depicts one method for presenting to a user the available user interface tool. In particular, FIG. **7a** depicts a screen **26** that includes a tool button **90**. The tool button **90** provides a graphical representation of the magnifying tool **48**. To activate the magnifying tool **48** the user may click, either by use of a mouse, keypad or touch screen, and drag from the magnifier tool button an image of the tool **48**. By clicking on the tool button **90** the system will process information from

the tool document file to create the image of the document **48** shown in FIG. **7a**. FIG. **7b** depicts that the user interface can allow in one practice the user to push the magnifying tool **48** off the screen **26**, optionally in any direction. By pushing the tool off the screen **26** the user removes the tool **48** and restores the icon, or the tool button **90** to the screen.

FIGS. **8a** and **8b** depict a further tool of the type that may be employed when viewing documents that include links, or other types of pointers to other documents, or other content. Specifically FIG. **8** depicts a tool wherein a document **100** includes links **102** to another document. For these documents, the systems and methods described herein may provide a slider control **104**, as well as buttons, switches, or some other control. The depicted slider control **104**, may enhance the user interface view of the document **100** such that by sliding the control **104**, the user can control the prominence of the link **102** within the document. Thus, the tool **104** allows the user to adjust the prominence of links within a document such as the document **100** so that links may be more easily identified by the user. FIGS. **8a** and **8b** further depict that document **100** may include highlighted text such as the depicted highlighted text **108**. As with the link **102** the slider control **104** may allow the highlighted text **108** to maintain its transparency while the transparency of the remaining portions of document **100** vary as the slider control **104** is varied by the user. In operation, the slider control **104** can allow the user to adjust the transparency, or alpha figure, of the objects that make up the document **100** other than the objects that make up the link **102** or highlighted text **108**. However, other techniques for fading or enhancing portions of a document, may be practiced.

Turning to FIG. **9**, a further graphical user interface tool is presented wherein a document **100** includes links **102**. Moreover the links **102** may be associated with a floating thumbnail document **110**. As shown in FIG. **9** a control **112** may be presented to the user. Upon activating the control **112**, those links **102** within the document **100** may be associated with a floating thumbnail representative of a page view of the page associated with the respective link **102**. Additionally, FIG. **9** depicts that in one option practice the display may further include a collegian of read ahead thumbnail documents **114**. The thumbnail documents **114** may be representative of those documents that are associated with links **102** within document **100**, or that are associated with other pages of the document **100** when that document **100** is a multi-page document.

FIG. **10** depicts a further example of a graphical user interface tool according to the invention. Specifically, FIG. **10** depicts a hand held computing device **120** having a set of characters **122** appearing on its display. As further shown in FIG. **10** a cursor window **124** appears over one character within text display **122**. In the depicted embodiment the cursor window **124** provides a soft, semi-transparent text entry plate floating over the current text position. The plate may move with the moving text position and/or the plate may maintain its position as the text itself scrolls to the left to accommodate movement of text under the plate **124**. As discussed above, the text plate cursor **124** may result from a tool document file **30** processed by the system **10** of FIG. **1**. The tool document file may include an internal representation of the text plate **124** appears on the device **120**. In one embodiment, the hand held device **120** includes a touch sensitive screen that allows a user to employ a stylus for forming characters that will appear on the screen within the text entry plate **124**. A design and development of such systems that allows such text entry are well known in the art and any of the suitable systems may be employed with the systems and methods described herein. In operation, a user may move a stylus across the screen of the device **120** to form letters that will appear within the text entry plate **124**. This operation is depicted in FIG. **11** wherein a series of text entry procedures **130** through **138** are depicted.

Particularly, FIG. **11** depicts text entry step **130** wherein the cursor plate **124** appears on the display of the device **120**. A user may trace a letter within space defined by the cursor **124**, or in another area and optionally, the tracings may appear within the area defined by the text entry cursor **124**. The tracings entered by the user may be processed by a character recognition system of the type known in the art to associate these markings with a character, such as the letter L in this example. As shown in step **132** once character recognition has been completed the recognized character L may be presented on the display and the cursor may move, or the text may scroll, but in either the cursor **124** becomes available for the user to enter more text. Thus as shown in step **134** the user may enter text until a word is formed. In step **138**, upon entry of a complete word the user may move the cursor **124** a space away from the written word and begin again tracing characters that will appear within the text entry cursor **124**. Thus, the depicted cursor **124** provides a tool that allows in-line insertion of content into a document, such as by inserting a section of text into an existing line of text that occurs within the document. In other applications, tools can be provided that edit images, such as by erasing content, changing colors or performing other applications.

As described above, for those systems that include a touch sensitive display the systems and methods described herein may provide for allowing a user to use a stylus to trace markings on the display and these marking may be interpreted by the system for allowing character entry. Additionally, FIG. **12***a* through **12***b* depicts a series of representative command strokes that a user may enter by moving stylus across the touch sensitive screen. Each of the command strokes depicted in FIG. **12***a* through **12***g* may be associated with a user interface command that the user may employ for manipulating and viewing documents. For example, **12***a* depicts a stroke wherein the user forms a rounded check mark that the system may be associated with a command that affirms an action proposed by the system. Similarly, FIG. **12***b* depicts a stroke that forms a plurality of peaks and troughs on the display and that may associated with a command to delete content from a display. FIG. **12***c* depicts a circular clock wise stroke that may be associated with having a document returned to a home page or to start over and FIG. **12***d* depicts a straight line diagonal upstroke that indicates a delete, clear, or no command. FIG. **12***e* depicts a box stroke running counter clock wise and indicating a paragraph select command and FIGS. **12***f* and **12***g* depict strokes that indicate a request by the user to move to the next or respectively previous document. It will be understood by those of ordinary skill in the art that as the systems and methods described herein include systems and methods that work with document of different types such as Word documents, web pages, streaming media, and other types of content the meaning of the different strokes may vary according to the application. For example, the circular clock wise rotation of **12***c* may indicate for a document that is representative of web page a request to return to a web page associated with the web page document. Alternatively, the use of the circular clock wise stroke of **12***c* when viewing streamed media content may indicate a request to start over, causing the streamed video to stop and restart from the beginning. Accordingly, it will be apparent to those with ordinary skill in the art that the stroke command depicted in FIGS. **12***a* through **12***b* may have different meanings depending on their applications.

FIGS. **13***a* and FIG. **13***b* depict a command stroke that may be employed by a user for clicking and dragging a document to cause page movement of that document within the viewing area. In the depicted embodiment, during a document drag operation document a velocity detector process takes position readings periodically, such as every centi-second. From these position readings a page velocity determination may be made. The page velocity determination may be employed for allowing the user interface to present a more natural way of moving documents through a viewing space. To this end, a process may employ the velocity determination to direct the parser/render **18** to redraw the document in a series of pictures that will portray the document as moving across the screen. For example, a user may drag a document at a certain speed and then release the stylus, mouse or other input device from the document. Optionally, upon release the document may stop moving. However, in an alternative practice the page may continue to move in the established direction until the user indicates that the document is to stop moving such as clicking on the document. For multi page documents the velocity measure may be used for panning different pages of the document across the screen at a rate determined by the page velocity set when the user drags one page of the document across the screen. Optionally, the velocity may decrease by a constant page inertia until it reaches zero velocity and page scrolling ceases; during page panning further velocity detection can be used to increase (accumulate) the page velocity and hence movement against the page inertia enabling smooth continues movement of the page between rapid sequential drag operations.

Additionally and optionally, other user interface processes may be provided to enhance the user experience of having tactile control over the document. For example, the user interface may include a page-flip detector for detecting a motion on the display **26** at a location of the display associated with the upper right hand corner of document **44** in FIG. **2**. If the page-flip detector, or the screen monitor, detect a brushing motion across the surface of the document **44**, the page-flip detector can direct the parser/render **18** to "flip" the page, causing the next page, chapter, scene or other section to be displayed. Motions may be detected in either direction for page-flipping back and forth, and the page flip detector may be context sensitive, generating a new display suited to the application and type of content. Optionally, the interface process may include a page curl detector that can operate similar to the page-flip detector, except that a motion in the upper right corner of document **44** can cause the page curl detector to direct the parser/render **18** to redraw the screen **42** or document **44** so that the corner of the document **44** is curled downward and a portion of the underlying page is presented. Both the page-flip and page-curl detectors may be computer processes that can generate instructions to the parser/renderer **18** to achieve the desired effect. Additionally, a page-zoom detector (such as a double-click over the page area) can be followed by an upward/downward movement to zoom in/out of the view. This function may be advantageously combined with the velocity detector to provide an inertial zoom feature.

It shall be obvious to those of skill in the art that although FIG. **1** graphically depicts the user interface system **10** as functional block elements, these elements can be realized as computer programs or portions of computer programs that are capable of running on a data processor platform to thereby configure the data processor as a system according to the invention. Moreover, although FIG. **1** depicts the system **10** as an integrated unit, it will be apparent to those or ordinary skill in the art that this is only one embodiment, and that the invention can be embodied as a computer program distributed across multiple platforms.

As discussed above, the user interface systems described above can be realized as a software component operating on a data processing system, including hand-held computing platforms, as well as more conventional computing platforms, such as a Unix workstation. In these embodiments, the user interface systems can be implemented as a C language computer program, or a computer program written in any high level language including C++, Fortran, Java or BASIC. Additionally, in an embodiment where the platform is primarily a microprocessor, microcontrollers or DSPs, the user interface systems can be realized as a computer program written in microcode or written in a high level language and compiled down to microcode that can be executed on the platform employed. The development of such systems is known to those of skill in the art, and such techniques are set forth in the literature, including for example Digital Signal Processing Applications with the TMS320 Family, Volumes I, II, and III, Texas Instruments (1990). Additionally, general techniques for high level programming are known, and set forth in, for example, Stephen G. Kochan, Programming in C, Hayden Publishing (1983). It is noted that DSPs are particularly suited for implementing signal processing functions, including preprocessing functions such as image enhancement through adjustments in contrast, edge definition and brightness. Developing code for the DSP and microcontroller systems follows from principles well known in the art.

Additionally, it is to be understood that although FIG. **1** graphically depicts the computer process **8** as comprising a plurality of functional block elements, these elements can be realized as computer programs or portions of computer programs that are capable of running on the data processing platform to thereby configure the data processing platform as a system according to the invention. Moreover, although FIG. **1** depicts the system **10** as an integrated unit of a process **8** and a display device **26**, it will be apparent to those of ordinary skill in the art that this is only one embodiment, and that the systems described herein can be realized through other architectures and arrangements, including system architectures that separate the document processing functions and user interface functions of the process **8** from the document display operation performed by the display **26**.

Those skilled in the art will know or be able to ascertain using no more than routine experimentation, many equivalents to the embodiments and practices described herein. Moreover, the systems and processes of the invention have wide application and can be employed in a range of devices including hand-held computers, telephones, mobile data terminals, set top boxes, an embedded processor, a notebook computer, a computer workstation, a printer, a copier, facsimile machine and other systems. Additionally, it will be understood by those of skill in the art, that the systems described herein may be practiced with any suitable interface devices, including touch-sensitive screens and pads, mouse input devices, keyboards and keypads, joysticks, thumb wheel devices, a mouse, a trackball, virtual reality input systems, voice control systems, eye movement control systems, and any other suitable devices. Thus, it will also be understood that the systems described herein have many uses and provide advantages over the prior art including providing a set of interface processes and systems that provide sophisticated manipulation of different document types.

Accordingly, it will be understood that the invention is not to be limited to the embodiments disclosed herein, but is to be understood from the following claims, which are to be interpreted as broadly as allowed under the law.

I claim:

1. A computer device having a system for simulating tactile control over a document, comprising

a processor, memory, and a touch-sensitive display,

system code stored within the memory and adapted to be executed by the processor to provide a digital representation of a document including data content and a page structure representative of a page layout of the document,

an engine for rendering an image of at least a portion of the page layout of the digital representation on the touch-sensitive display,

a display monitor in communication with the touch-sensitive display screen for detecting motion of a pointer across the touch-sensitive display,

a velocity detector for determining a velocity vector based on a velocity of the detected motion,

an interface process in communication with the display monitor for processing the motion detected by the display monitor to detect one of a plurality of commands, wherein the plurality of commands includes a pan command,

wherein, in response to the command detected by the interface process being the pan command, the engine pans the displayed document on the display at a rate based on the determined velocity vector.

2. The computing device of claim **1**, wherein panning the displayed document comprises rendering different views of the document on the touch-sensitive display at a rate based on the determined velocity vector and a page inertia.

3. A computer device having a system for simulating tactile control over a document, comprising

a processor, memory, and a touch-sensitive display,

system code stored within the memory and adapted to be executed by the processor to provide a digital representation of a document including data content and a page structure representative of a page layout of the document,

an engine for rendering an image of at least a portion of the page layout of the digital representation on the touch-sensitive display,

a display monitor in communication with the touch-sensitive display screen for detecting motion of a pointer across the touch-sensitive display,

a velocity detector for determining a velocity vector associated with the detected motion,

an interface process in communication with the display monitor for processing the motion detected by the display monitor to detect one of a plurality of commands, wherein the plurality of commands includes a pan command,

wherein, in response to the command detected by the interface process being the pan command, the engine renders a series of pages of the document on the touch-sensitive display at a rate based on the determined velocity vector and a page inertia.

4. A computing device according to claim **3**, wherein the rate at which the engine renders the series of pages of the document decreases over time based on the page inertia.

5. A computing device according to claim **3**, wherein in response to the interface process detecting a subsequent pan command based on a subsequent motion of a pointer across the display, the engine alters the rate at which it renders the series of pages based on a velocity vector the velocity detector determines in relation to the subsequent motion.

* * * * *