NXDN®

NXDN Technical Specifications

Part 2:
Conformance Test

Sub-part B:
Common Air Interface Test

NXDN TS 2-B Version 1.2

November 2012

NXDN Forum

Contents

Figures

Tables

Disclaimer

The information presented here is intended to be for clarification and/or information purpose only, and care has been taken to keep the contents as neutral and accurate as possible.

The use or practice of contents of the information may involve the use of intellectual property rights ("IPR"), including pending or issued patents, or copyrights, owned by one or more parties. The NXDN Forum makes no search or investigation for IPR, nor the NXDN Forum makes no arrangement of licensing negotiation for IPR between the user and the owner of IPR.

All warranties, express or implied, are disclaimed, including without limitation, any and all warranties concerning the accuracy of the contents, its fitness or appropriateness for a particular purpose or use, its merchantability and its non-infringement of any third party's IPR.

The NXDN Forum expressly disclaims any and all responsibilities for the accuracy of the contents and makes no representations or warranties regarding the content's compliance with any applicable statute, rule or regulation.

The NXDN Forum shall not be liable for any and all damages, direct or indirect, arising from or relating to any use of the contents contained herein, including without limitation any and all indirect, special, incidental or consequential damages (including damages for loss of business, loss of profits, litigation, or the like), whether based upon breach of contract, breach of warranty, tort (including negligence), product liability or otherwise, even if advised of the possibility of such damages.

The foregoing negation of damages is a fundamental element of the use of the contents hereof, and these contents would not be published by the NXDN Forum without such limitations.

Document Copyrights

Trademarks

NXDN® is a registered trademark of JVC KENWOOD Corporation and Icom Incorporated.

AMBE+2™ is a trademark of Digital Voice Systems, Inc.

1. Introduction

This document provides the procedures and determination criteria of conformance testing for the lower layer of the radio equipment designed in conformity with the NXDN Air Interface specifications.

The interoperability of the lower layer between different manufacturer's radio equipment which conforms to the NXDN Air Interface specifications can be verified through this testing.

Tests defined in this document are intended to verify whether or not preamble pattern, sync word pattern, and encoding and decoding are properly implemented.

This testing shall take place prior to other testing including Basic Operation Test in REF [2] and Trunking Operation Test in REF [3] and REF [4].

2. References

Reference documents are listed below.

REF [1]	Part 1-A Common Air Interface Version 1.2
REF [2]	Part 2-C Basic Operation Test Version 1.1
REF [3]	Part 2-D Trunking Operation Test Version 1.0
REF [4]	Part 2-E Trunking Operation Test (Type-D) Version 1.0

3. Abbreviations

To help understand this document, abbreviations are listed below.

CAC	Common Access Channel
CAI	Common Air Interface
CCCH	Common Control Channel
CR	Conventional Repeater
EFR	Enhanced Full Rate
EHR	Enhanced Half Rate
FACCH1	Fast Associated Control Channel 1
FACCH2	Fast Associated Control Channel 2
FACCH3	Fast Associated Control Channel 3
FEC	Forward Error Correction
FSW	Frame Sync Word
LICH	Link Information Channel
Long CAC	Long Common Access Channel
PE	Partial Echo
RAN	Radio Access Number
RCCH	RF Control Channel
RDCH	RF Direct Channel
RTCH	RF Traffic Channel
RTCH2	RF Traffic Channel 2
RU	Repeater Unit
SACCH	Slow Associated Control Channel
SCCH	Signaling Control Channel
Short CAC	Short Common Access Channel
SU	Subscriber Unit
TR	Trunking Repeater
UDCH	User Data Channel
UDCH2	User Data Channel 2
UPCH	User Packet Channel
USC	User Specific Channel
VCH	Voice Channel

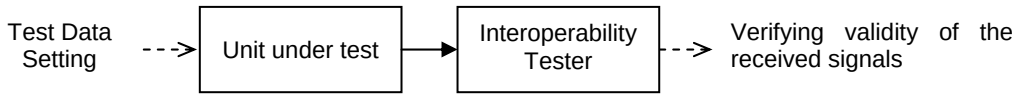
4. Common Air Interface Test (Conventional, Type-C Trunked System)

This Common Air Interface Test is intended to verify that an encoder and decoder properly function and additionally a fixed pattern such as Frame Sync Word is correct. Hereinafter, the Common Air Interface Test shall be referred to as CAI Test.

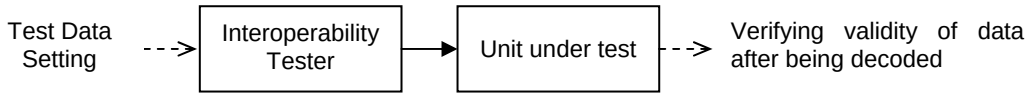
The encoder described here is inclusive of a series of processes against the layer 3 information such as CRC encoding, convolutional encoding, punctured encoding, interleaving and scrambling. Also, the decoder is the process opposite to encoding and is inclusive of a series of processes to restore the layer 3 information by descrambling, deinterleaving, punctured decoding, convolutional decoding, and CRC decoding.

In the testing for the encoder, as shown in the example of configuration for the encoder test, the encoded signals transmitted from a unit under test are received by an interoperability tester and are compared with a data string after the encoding.

In the testing for the decoder, as shown in the example of configuration for the decoder test, a unit under test receives the encoded signals transmitted from an interoperability tester, and the decoded signals which a unit under test output are compared with a data string before the encoding. However, no specifications of measuring instruments to be used for testing are defined, and also no method of setting data to the encoder and method of outputting the decoded data and the data which is encoded is defined; hence, methods can arbitrarily be defined by manufacturers of units under test.



Example of Configuration for Encoder Test



Example of Configuration for Decoder Test

For the explanation hereinafter, for sake of simplicity, measuring instruments used shall be referred to as interoperability tester.

For testing, unless otherwise specifically defined, an antenna input level of a receiver shall be -47 dBm or be equal to a sufficiently large level.

The relation between the operational mode for a unit under test and the test items in Section 4 is following list. An examiner performs required tests according to operational modes implemented in the unit under test.

	Operational Mode	Test Items in Section 4							
		4.2	4.3	4.4	4.5	4.6	4.7	4.8	4.9
SU	4800 bps Conventional Mode	-	x	x	-	x	x	x	-
	9600 bps Conventional Mode	-	x	x	-	x	x	x	-
CR	4800 bps Conventional Mode	-	x	x	-	x	x	x	-
	9600 bps Conventional Mode	-	x	x	-	x	x	x	-
SU	4800 bps Trunked Mode	x	x	x	x	x	x	x	-
	9600 bps Trunked Mode	x	x	x	x	x	x	x	-
TR	4800 bps Trunked Mode	x	x	x	x	x	x	x	x
	9600 bps Trunked Mode	x	x	x	x	x	x	x	x

Relation between Mode of unit under test and Test Items

4.1. Test Parameters

This section defines parameters which are utilized in testing defined in Section 4.2 and afterwards. The data strings before and after encoding associated with parameters for each functional channel are shown below. The data string is, in principle, described in the hexadecimal format; however, the letter “b” shall be suffixed in the case that the data string is described in the binary format. The transmit sequence of a data string is that the leftmost value is sent first and the rightmost value is sent out at the end. For example, if the data is converted from hexadecimal format to symbol format:

1A3 The value in hexadecimal format is converted into a dibit string,
00 01 10 10 00 11 and then is converted to symbol format.
+1 +3 -1 -1 +1 -3

Hence, symbols are sent out in the sequence of +1, +3, -1, -1, +1, -3 from the left.

4.1.1. LICH

An LICH is a channel placed in all frames and utilized for discrimination of kind of radio channels and functional channels. The LICH is 8-bit data consisting of a 7-bit information bit and a 1-bit parity bit, and the LICH after being encoded has 16 bits length by a dibit conversion.

LICH contains a bit representing inbound or outbound direction, and the direction can be restricted depending on functional channels to be used. Figure 4.1-1 represents the direction of a functional channel to be used in a trunked radio system. In tests of Long CAC and Short CAC, the LICH configurations need only inbound direction from an SU to a TR, and in tests of CAC/ collision control fields the LICH configurations need only outbound direction from a TR to an SU. Since other functional channels are used in both of inbound and outbound directions, the LICH configurations need both directions too.

Since functional channels to be used in a conventional system as shown in Figure 4.1-2 are used in both of inbound and outbound directions, the LICH configurations need both directions too.

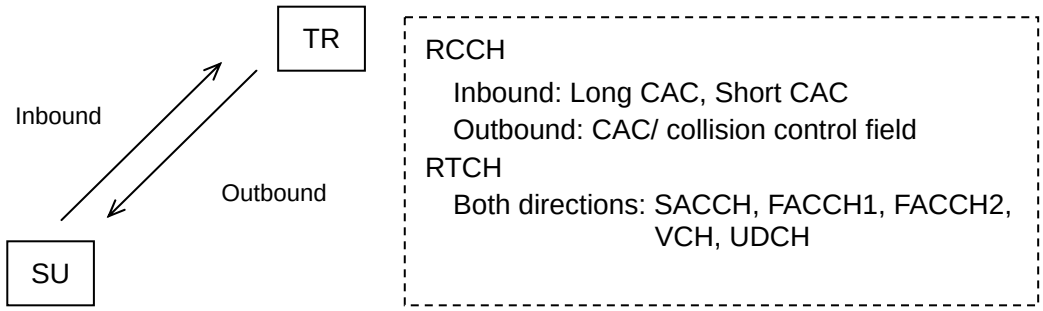


Figure 4.1-1 Direction of a Functional Channel in a Trunked System

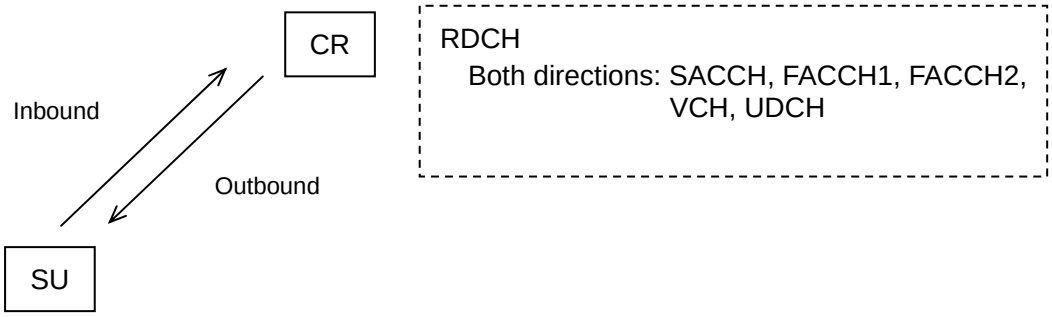


Figure 4.1-2 Direction of a Functional Channel in a Conventional System

Following are LICH to be tested:

LICH for CAC/ Collision Control Field Test			
Outbound Direction	(Before dibit)	02	
	(After dibit)	555D	
	(After scrambling)	5D77	
LICH for Long CAC Test			
Inbound Direction	(Before dibit)	11	
	(After dibit)	5757	
	(After scrambling)	5F7D	
LICH for Short CAC Test			
Inbound Direction	(Before dibit)	30	
	(After dibit)	5F55	
	(After scrambling)	577 F	
LICH for VCH/SACCH Test (RDCH/ SACCH Superframe/ VCH) → Left (RTCH/ SACCH Superframe/ VCH) → Right			
Inbound Direction	(Before dibit)	AC	6C
	(After dibit)	DDF5	7DF5
	(After scrambling)	D5DF	75DF
Outbound Direction	(Before dibit)	AE	6E
	(After dibit)	DDFD	7DFD
	(After scrambling)	D5D7	75D7
LICH for FACCH1/SACCH Test (RDCH/ SACCH Non-superframe/ FACCH1) → Left (RTCH/ SACCH Non-superframe/ FACCH1) → Right			
Inbound Direction	(Before dibit)	81	41
	(After dibit)	D557	7557
	(After scrambling)	DD7D	7D7D
Outbound Direction	(Before dibit)	83	43
	(After dibit)	D55F	755F
	(After scrambling)	DD75	7D75
LICH for FACCH1/SACCH Test (RDCH/ SACCH Superframe/ FACCH1) → Left (RTCH/ SACCH Superframe/ FACCH1) → Right			
Inbound Direction	(Before dibit)	A0	60
	(After dibit)	DD55	7D55
	(After scrambling)	D57F	757F
Outbound Direction	(Before dibit)	A2	62
	(After dibit)	DD5D	7D5D
	(After scrambling)	D577	7577
LICH for FACCH2 Test (RDCH/ FACCH2) → Left (RDCH/ FACCH2) → Right			
Inbound Direction	(Before dibit)	90	50
	(After dibit)	D755	7755
	(After scrambling)	DF7F	7F7F
Outbound Direction	(Before dibit)	92	52
	(After dibit)	D75D	775D
	(After scrambling)	DF77	7F77
LICH for UDCH Test (RDCH/ UDCH) → Left (RDCH/ UDCH) → Right			
Inbound Direction	(Before dibit)	9C	5C
	(After dibit)	D7F5	77F5
	(After scrambling)	DFDF	7FDF
Outbound Direction	(Before dibit)	9E	5E
	(After dibit)	D7FD	77FD
	(After scrambling)	DFD7	7FD7

4.1.2. CAC

CAC is a generic term for functional channels (BCCH, CCCH and UPCH) that are used on an outbound control channel in a trunked radio system.

A data string before being encoded is 152-bit data consisting of 144-bit layer 3 information and 8-bit SR information, and the 300-bit CAC is generated by encoding the data string. Messages to be used in the CAC test are SITE_INFO, SRV_INFO and ADJ_SITE_INFO. SITE_INFO gives an example of single message format, and SRV_INFO and ADJ_SITE_INFO give an example of dual message format.

<SITE_INFO>

```
RAN:                              000001b
Message Type:                     011000b
Location ID:                      400021
  (Category = 01b, System Code = 00000000000000001b, Site Code = 00001b)
Channel Structure Information:  4914
  (Bn = 1, Gn = 1, Pn = 2, Mn = 1, In = 4)
Service Information:              BFC0
Restriction Information:          000000
Channel Access Information:       000000
Version Number:                   00
Adjacent Site Allocation:         1
1st Control Channel:              0000000001b
2nd Control Channel:              0000000000b
```

CAC (Single Message Format of SITE_INFO)

```
(Before coding)          8118 4000 2149 14BF C000 0000 0000 0000 1004 00
(After interleaving)     F240 007B C008 40F0 0684 9802 0E20 01A7 3000 1895
                         0021 0A00 11A6 0000 2080 14CB F000 E108 041
(After scrambling)       FAC8 A071 EA80 6858 26A6 988A AC2A AB8F 1082 303F
                         A0A3 8282 9326 00A0 08A2 1643 D2AA 430A 2C1
```

<SRV_INFO and ADJ_SITE_INFO>

```
RAN:                            000001b
<SRV_INFO>
  Message Type:                 011001b
  Location ID:                  400021
    (Category = 01b, System Code = 00000000000000001b, Site Code = 00001b)
  Service Information:          BFC0
  Restriction Information:      000000
<ADJ_SITE_INFO>
  Message Type:                 011011b
  Adjacent Site Location ID:    400022
    (Category = 01b, System Code = 00000000000000001b, Site Code = 00010b)
  Adjacent Site Option:         000001b
  Adjacent Control Channel:     0000000010b
```

CAC (Dual Message Format of SRV_INFO and ADJ_SITE_INFO)

```
(Before coding)          4119 4000 21BF C000 0000 1B40 0022 0402 0000 00
(After interleaving)     710C 203A 0070 E881 70C4 8098 0C00 5E2B 2041 161C
                         2407 4E12 4025 0820 7300 400A 8400 0102 306
(After scrambling)       7984 8030 2AF8 C029 50E6 8010 AE0A F403 00C3 3EB6
                         8485 C690 C2A5 0880 5B22 4282 A6AA A300 186
```

4.1.3. Long CAC

A Long CAC is a functional channel which can be used on an inbound control channel in a trunked radio system.

A data string before being encoded is 136-bit data consisting of 128-bit layer 3 information and 8-bit SR information. The 252-bit Long CAC is generated by encoding the data string. Messages to be used in the Long CAC test are SDCALL_REQ (Header) and SDCALL_REQ (User Data). There are two types of SDCALL_REQ (Header) messages for 4800 bps mode and 9600 bps mode.

ID configurations to be used in SDCALL_REQ (Header) are three patterns as presented in Table 4.1-1.

	Source Unit ID	Destination Unit ID
Pattern 1	0001	0002
Pattern 2	7FF8	7FF9
Pattern 3	FFEF	FFEE

Table 4.1-1 ID Settings for an Individual Call

<SDCALL_REQ (Header)>

```
    RAN:                  000001b
    Message Type:         111000b
    CC Option:            00
    Call Type:            100b
    Data Call Option:     00000b (4800 bps)
                          00010b (9600 bps)
    Source Unit ID:       0001/ 7FF8/ FFEF
    Destination Unit ID:  0002/ 7FF9/ FFEE
    Cipher Type:          00b
    Key ID:               000000b
    Packet Information:   8004
```

Pattern 1

-4800 bps-

Long CAC

```
    (Before coding)         0138 0080 0001 0002 0080 0400 0000 0000 00
    (After interleaving)    0028 0300 008C 0806 2540 0429 0188 0809 0200 2A4A
                            0342 C032 8480 64B0 0F05 207
    (After scrambling)      08A0 A30A 2A04 20AE 0562 04A1 A382 A221 2282 02E0
                            A3C0 48B0 0600 6410 2727 22F
```

-9600 bps-

Long CAC

```
    (Before coding)         0138 0082 0001 0002 0080 0400 0000 0000 00
    (After interleaving)    0828 1B00 004C 0808 3540 64A9 0308 0809 2200 CB4A
                            0542 C02A 8480 A4B0 0705 201
    (After scrambling)      00A0 BB0A 2AC4 20A0 1562 6421 A102 A221 0282 E3E0
                            A5C0 48A8 0600 A410 2F27 229
```

Pattern 2

-4800 bps-

Long CAC

(Before coding) 0138 0080 7FF8 7FF9 0080 0400 0000 0000 00

(After interleaving) 02E8 032A 00CC 0800 2A40 04E1 020D C811 3C00 8A7A 014F 402A A081 27F0 0714 203

(After scrambling) 0A60 A320 2A44 20A8 0A62 0469 A007 6239 1C82 A2D0 A1CD C8A8 2201 2750 2F36 22B

-9600 bps-

Long CAC

(Before coding) 0138 0082 7FF8 7FF9 0080 0400 0000 0000 00

(After interleaving) 0AE8 1B2A 000C 080E 3A40 6461 008D C811 1C00 6B7A 074F 4032 A081 E7F0 0F14 205

(After scrambling) 0260 BB20 2A84 20A6 1A62 64E9 A287 6239 3C82 43D0 A7CD C8B0 2201 E750 2736 22D

Pattern 3

-4800 bps-

Long CAC

(Before coding) 0138 0080 FFEF FFEE 0080 0400 0000 0000 00

(After interleaving) 03A8 132C 004E 280A 3A40 54E1 0389 C809 3C00 AB2A 0547 4012 C081 25B0 051B 200

(After scrambling) 0B20 B326 2AC6 00A2 1A62 5469 A183 6221 1C82 8380 A5C5 C890 4201 2510 2D39 228

-9600 bps-

Long CAC

(Before coding) 0138 0082 FFEF FFEE 0080 0400 0000 0000 00

(After interleaving) 0BA8 0B2C 008E 2804 2A40 3461 0109 C809 1C00 4A2A 0347 400A C081 E5B0 0D1B 206

(After scrambling) 0320 AB26 2A06 00AC 0A62 34E9 A303 6221 3C82 6280 A3C5 C888 4201 E510 2539 22E

<SDCALL_REQ (User Data)>

RAN: 000001b

Message Type: 111001b

Packet Frame Number: 0000b

Block Number: 0000b

User Data: ABCDEFGHIJ

Long CAC

(Before coding) 0139 0041 4243 4445 4647 4849 4A77 FD05 3E

(After interleaving) 27CA F271 AC09 9BD0 142B B79D 2F9A 45C9 A4BA 499E 655C F64A CB9F 21A7 3F3B FA1

(After scrambling) 2F42 527B 8681 B378 3409 B715 8D90 EFE1 8438 6134 C5DE 7EC8 491F 2107 1719 F89

4.1.4. Short CAC

A Short CAC is a functional channel which can be used on an inbound control channel in a trunked radio system.

A data string before being encoded is 104-bit data consisting of 96-bit layer 3 information and 8-bit SR information. The 252-bit Short CAC is generated by encoding the data string. Messages to be used in the Short CAC test are VCALL_REQ and DCALL_REQ. There are three types of VCALL_REQ messages for 4800bps/EHR, 9600bps/EHR and 9600bps/EFR, and two types of DCALL_REQ messages for 4800 bps mode and 9600 bps mode.

ID configurations to be used in VCALL_REQ are three patterns as presented in Table 4.1-2.

ID configurations to be used in DCALL_REQ are three patterns as presented in Table 4.1-1.

	Source Unit ID	Destination Group ID
Pattern 4	0001	0001
Pattern 5	7FF8	7FF8
Pattern 6	FFEF	FFEF

Table 4.1-2 ID Settings for Group Call

<VCALL_REQ>

```
    RAN:                    000001b
    Message Type:           000001b
    CC Option:              00
    Call Type:              001b
    Data Call Option:       00000b (4800 bps/EHR)
                            00010b (9600 bps/EHR)
                            00011b (9600 bps/EFR)
    Source Unit ID:         0001 / 7FF8 / FFEF
    Destination Group ID:   0001 / 7FF8 / FFEF

Pattern 4
```

-4800 bps-
Short CAC

```
        (Before coding)         0101 0020 0001 0001 0000 0000 00
        (After interleaving)    1410 0024 0154 0404 A120 6088 0224 0004 8280 4D14
                                0340 0019 4101 9240 1EC2 40B
        (After scrambling)      1C98 A02E 2BDC 2CAC 8102 6000 A02E AA2C A202 65BE
                                A3C2 889B C381 92E0 36E0 423
```

-9600 bps/EHR-
Short CAC

```
        (Before coding)         0101 0022 0001 0001 0000 0000 00
        (After interleaving)    1010 1804 0054 0404 A920 0088 03A6 0010 9280 ED14
                                0044 0039 6101 5240 00C2 406
        (After scrambling)      1898 B80E 2ADC 2CAC 8902 0000 A1AC AA38 B202 C5BE
                                A0C6 88BB E381 52E0 28E0 42E
```

-9600 bps/EFR-
Short CAC

(Before coding) 0101 0023 0001 0001 0000 0000 00
(After interleaving) 1010 3804 0055 0408 A120 5088 0324 000C 9280 CD94
0240 0051 6102 5340 02CA 40F
(After scrambling) 1898 980E 2ADD 2CA0 8102 5000 A12E AA24 B202 E53E
A2C2 88D3 E382 53E0 2AE8 427

Pattern 5

-4800 bps-
Short CAC

(Before coding) 0101 0020 7FF8 7FF8 0000 0000 00
(After interleaving) 15E0 2034 0054 F000 A140 00BC 02A4 001C 9B00 4DB4
0346 6079 4503 53B0 08CA 00E
(After scrambling) 1D68 803E 2ADC D8A8 8162 0034 A0AE AA34 BB82 651E
A3C4 E8FB C783 5310 20E8 026

-9600 bps/EHR-
Short CAC

(Before coding) 0101 0022 7FF8 7FF8 0000 0000 00
(After interleaving) 11E0 3814 0154 F000 A940 60BC 0326 0008 8B00 EDB4
0042 6059 6503 93B0 16CA 003
(After scrambling) 1968 981E 2BDC D8A8 8962 6034 A12C AA20 AB82 C51E
A0C0 E8DB E783 9310 3EE8 02B

-9600 bps/EFR-
Short CAC

(Before coding) 0101 0023 7FF8 7FF8 0000 0000 00
(After interleaving) 11E0 1814 0155 F00C A140 30BC 03A4 0014 8B00 CD34
0246 6031 6500 92B0 14C2 00A
(After scrambling) 1968 B81E 2BDD D8A4 8162 3034 A1AE AA3C AB82 E59E
A2C4 E8B3 E780 9210 3CE0 022

Pattern 6

-4800 bps-
Short CAC

(Before coding) 0101 0020 FFEF FFEF 0000 0000 00
(After interleaving) 17E0 1021 8194 BC02 A260 60FC 0226 8000 9F00 0D24
0346 C039 6100 52A0 10CA 808
(After scrambling) 1F68 B02B AB1C 94AA 8242 6074 A02C 2A28 BF82 258E
A3C4 48BB E380 5200 38E8 820

-9600 bps/EHR-
Short CAC

(Before coding) 0101 0022 FFEF FFEF 0000 0000 00
(After interleaving) 13E0 0801 8094 BC02 AA60 00FC 03A4 8014 8F00 AD24
0042 C019 4100 92A0 0ECA 805
(After scrambling) 1B68 A80B AA1C 94AA 8A42 0074 A1AE 2A3C AF82 858E
A0C0 489B C380 9200 26E8 82D

-9600 bps/EFR-
Short CAC

(Before coding)	0101 0023 FFEF FFEF 0000 0000 00
(After interleaving)	13E0 2801 8095 BC0E A260 50FC 0326 8008 8F00 8DA4 0246 C071 4103 93A0 0CC2 80C
(After scrambling)	1B68 880B AA1D 94A6 8242 5074 A12C 2A20 AF82 A50E A2C4 48F3 C383 9300 24E0 824

<DCALL_REQ>

RAN:	000001b
Message Type:	001001b
CC Option:	00
Call Type:	100b
Data Call Option:	00000b (4800 bps) 00010b (9600 bps)
Source Unit ID:	0001 / 7FF8 / FFEF
Destination Group ID:	0002 / 7FF9 / FFEE

Pattern 1

-4800 bps-
Short CAC

(Before coding)	0109 0080 0001 0002 0000 0000 00
(After interleaving)	3810 3144 8014 0000 F100 100A 03B4 1014 2200 EC14 0268 0038 4100 1048 16C2 00F
(After scrambling)	3098 914E AA9C 28A8 D122 1082 A1BE BA3C 0282 C4BE A2EA 88BA C380 10E8 3EE0 027

-9600 bps-
Short CAC

(Before coding)	0109 0082 0001 0002 0000 0000 00
(After interleaving)	3C10 2964 8114 0000 F900 700A 0236 1000 3200 4C14 016C 0018 6100 D048 08C2 002
(After scrambling)	3498 896E AB9C 28A8 D922 7082 A03C BA28 1282 64BE A1EE 889A E380 D0E8 20E0 02A

Pattern 2

-4800 bps-
Short CAC

(Before coding)	0109 0080 7FF8 7FF9 0000 0000 00
(After interleaving)	39F0 1954 0154 F404 F160 403C 0034 0010 3B80 0CB0 026E 6040 4402 91B0 04CA 408
(After scrambling)	3178 B95E 2BDC DCAC D142 40B4 A23E AA38 1B02 241A A2EC E8C2 C682 9110 2CE8 420

-9600 bps-
Short CAC

(Before coding)	0109 0082 7FF8 7FF9 0000 0000 00
(After interleaving)	3DF0 0174 0054 F404 F960 203C 01B6 0004 2B80 ACB0 016A 6060 6402 51B0 1ACA 405
(After scrambling)	3578 A17E 2ADC DCAC D942 20B4 A3BC AA2C 0B02 841A A1E8 E8E2 E682 5110 32E8 42D

Pattern 3

-4800 bps-

Short CAC

(Before coding)	0109 0080 FFEF FFEE 0000 0000 00
(After interleaving)	3BF0 2941 8094 B806 F240 207C 00B6 800C 3F80 4C20 026E C000 6001 90A0 1CCA C0E
(After scrambling)	3378 894B AA1C 90AE D262 20F4 A2BC 2A24 1F02 648A A2EC 4882 E281 9000 34E8 C26

-9600 bps-

Short CAC

(Before coding)	0109 0082 FFEF FFEE 0000 0000 00
(After interleaving)	3FF0 3161 8194 B806 FA40 407C 0134 8018 2F80 EC20 016A C020 4001 50A0 02CA C03
(After scrambling)	3778 916B AB1C 90AE DA62 40F4 A33E 2A30 0F02 C48A A1E8 48A2 C281 5000 2AE8 C2B

4.1.5. Collision Control Field

A collision control field contains the information utilized for random access control in a trunked radio system. The collision control field consists of a 4-bit field to be converted to dibit and a 16-bit field (Partial Echo) not to be converted to dibit.

The 16-bit CRC that is a data decoded from the inbound control channel message transmitted by an SU shall be set to a PE. However null shall be set to the PE if it is in the idle state with no inbound signal. This test uses PE value, which are Null and 16-bit CRC value included in a VCALL_REQ message (Pattern 4) which an SU sends with a Short CAC as specified in Section 4.1.4.

```
Idle state:
Collision Control Field
    (Pre-dibit)           A000 0
    (Post dibit)          DD00 00
    (After scrambling)    758A 08

Reception Success state in 4800 bps (CRC = 45B1):
Collision Control Field
    (Pre-dibit)           B45B 1
    (Post dibit)          DF45 B1
    (After scrambling)    77CF B9

Reception Success state in 9600 bps/EHR (CRC = 0FFA):
Collision Control Field
    (Pre-dibit)           B0FF A
    (Post dibit)          DF0F FA
    (After scrambling)    7785 F2

Reception Success state in 9600 bps/EHR (CRC = A2CF):
Collision Control Field
    (Pre-dibit)           BA2C F
    (Post dibit)          DFA2 CF
    (After scrambling)    7728 C7
```

4.1.6. SACCH

An SACCH is a functional channel to transfer a control information which always accompanies audio data on a traffic channel in a trunked radio system and conventional system. A data string before being encoded is 26-bit data consisting of 18-bit layer 3 information and 8-bit SR information. The 60-bit SACCH is generated by encoding the data string.

Messages to be used in SACCH test are IDLE and VCALL. The SACCH to send an IDLE message shall employ a non-superframe structure, and the SACCH to send a VCALL message shall employ a superframe structure. Also, there are three types of VCALL messages for 4800 bps/EHR, 9600 bps/EHR and 9600 bps/EFR.

RAN and ID configurations to be used in this test are three patterns as presented in Table 4.1-3. The Source ID and Destination ID are not used in IDLE.

	RAN	Source Unit ID	Destination Group ID
Pattern 7	000001b	0001	0001
Pattern 8	100000b	7FF8	7FF8
Pattern 9	111111b	FFEF	FFEF

Table 4.1-3 RAN and ID Settings for Group Call

<IDLE>

RAN: 000001b/ 100000b/ 111111b
Message Type: 010000b
CC Option: 00

Pattern 7

SACCH

(Before coding) 0110 000 (Upper 26 bits are valid.)
(After interleaving) 6B4A 1424 0002 02B
(After scrambling) 63C2 B42E 2A8A 2A1

Pattern 8

SACCH

(Before coding) 2010 000 (Upper 26 bits are valid.)
(After interleaving) 2948 0802 0986 61B
(After scrambling) 21C0 A808 230E 491

Pattern 9

SACCH

(Before coding) 3F10 000 (Upper 26 bits are valid.)
(After interleaving) 6108 08C1 109A 66A
(After scrambling) 6980 A8CB 3A12 4E0

<VCALL>

RAN:	000001b/ 100000b/ 111111b
Message Type:	000001b
CC Option:	00
Call Type:	001b
Voice Call Option:	00000b (4800 bps/EHR)
	00010b (9600 bps/EHR)
	00011b (9600 bps/EFR)
Source Unit ID:	0001/ 7FF8/ FFEF
Destination Group ID:	0001/ 7FF8/ FFEF
Cipher Type:	00b
Key ID:	000000b

Pattern 7

-4800 bps-

SACCH

First Frame

(Before coding)	C101 000 (Upper 26 bits are valid.)
(After interleaving)	C728 06E2 88BD 462
(After scrambling)	CFA0 A6E8 A235 6E8

Second Frame

(Before coding)	8180 000 (Upper 26 bits are valid.)
(After interleaving)	CE53 10E1 10CA 422
(After scrambling)	C6DB B0EB 3A42 6A8

Third Frame

(Before coding)	4110 000 (Upper 26 bits are valid.)
(After interleaving)	6B29 14A2 3006 00A
(After scrambling)	63A1 B4A8 1A8E 280

Fourth Frame

(Before coding)	0140 000 (Upper 26 bits are valid.)
(After interleaving)	4A00 9021 2858 568
(After scrambling)	4288 302B 02D0 7E2

-9600 bps/EHR-
SACCH

First Frame

(Before coding)	C101 000 (Upper 26 bits are valid.)
(After interleaving)	C728 06E2 88BD 462
(After scrambling)	CFA0 A6E8 A235 6E8

Second Frame

(Before coding)	8188 000 (Upper 26 bits are valid.)
(After interleaving)	C751 52E1 3180 409
(After scrambling)	CFD9 F2EB 1B08 683

Third Frame

(Before coding)	4110 000 (Upper 26 bits are valid.)
(After interleaving)	6B29 14A2 3006 00A
(After scrambling)	63A1 B4A8 1A8E 280

Fourth Frame

(Before coding)	0140 000 (Upper 26 bits are valid.)
(After interleaving)	4A00 9021 2858 568
(After scrambling)	4288 302B 02D0 7E2

-9600 bps/EFR-
SACCH

First Frame

(Before coding)	C101 000 (Upper 26 bits are valid.)
(After interleaving)	C728 06E2 88BD 462
(After scrambling)	CFA0 A6E8 A235 6E8

Second Frame

(Before coding)	818C 000 (Upper 26 bits are valid.)
(After interleaving)	C75B 0275 1198 921
(After scrambling)	CFD3 A27F 3B10 BAB

Third Frame

(Before coding)	4110 000 (Upper 26 bits are valid.)
(After interleaving)	6B29 14A2 3006 00A
(After scrambling)	63A1 B4A8 1A8E 280

Fourth Frame

(Before coding)	0140 000 (Upper 26 bits are valid.)
(After interleaving)	4A00 9021 2858 568
(After scrambling)	4288 302B 02D0 7E2

Pattern 8

-4800 bps-

SACCH

First Frame

(Before coding)	E001 000 (Upper 26 bits are valid.)
(After interleaving)	852A 1AC4 8139 252
(After scrambling)	8DA2 BACE ABB1 0D8

Second Frame

(Before coding)	A081 FFC (Upper 26 bits are valid.)
(After interleaving)	8DDB 7D57 7C66 AD6
(After scrambling)	8553 DD5D 56EE 85C

Third Frame

(Before coding)	6087 FF8 (Upper 26 bits are valid.)
(After interleaving)	29B9 2F95 BBEA BBD
(After scrambling)	2131 8F9F 9162 937

Fourth Frame

(Before coding)	2000 000 (Upper 26 bits are valid.)
(After interleaving)	0000 0886 218C A73
(After scrambling)	0888 A88C 0B04 8F9

-9600 bps/EHR-

SACCH

First Frame

(Before coding)	E001 000 (Upper 26 bits are valid.)
(After interleaving)	852A 1AC4 8139 252
(After scrambling)	8DA2 BACE ABB1 0D8

Second Frame

(Before coding)	A089 FFC (Upper 26 bits are valid.)
(After interleaving)	8DDB 7D57 7C66 AD6
(After scrambling)	8553 DD5D 56EE 85C

Third Frame

(Before coding)	6087 FF8 (Upper 26 bits are valid.)
(After interleaving)	29B9 2F95 BBEA BBD
(After scrambling)	2131 8F9F 9162 937

Fourth Frame

(Before coding)	2000 000 (Upper 26 bits are valid.)
(After interleaving)	0000 0886 218C A73
(After scrambling)	0888 A88C 0B04 8F9

-9600 bps/EFR-

SACCH

First Frame

(Before coding)	E001 000 (Upper 26 bits are valid.)
(After interleaving)	852A 1AC4 8139 252
(After scrambling)	8DA2 BACE ABB1 0D8

Second Frame

(Before coding)	A08D FFC (Upper 26 bits are valid.)
(After interleaving)	84D3 6FC3 7D34 7D5
(After scrambling)	8C5B CFC9 57BC 55F

Third Frame

(Before coding)	6087 FF8 (Upper 26 bits are valid.)
(After interleaving)	29B9 2F95 BBEA BBD
(After scrambling)	2131 8F9F 9162 937

Fourth Frame

(Before coding)	2000 000 (Upper 26 bits are valid.)
(After interleaving)	0000 0886 218C A73
(After scrambling)	0888 A88C 0B04 8F9

Pattern 9

-4800 bps-

SACCH

First Frame

(Before coding)	FF01 000 (Upper 26 bits are valid.)
(After interleaving)	CD6A 1A07 9825 223
(After scrambling)	C5E2 BA0D B2AD 0A9

Second Frame

(Before coding)	BF83 FF8 (Upper 26 bits are valid.)
(After interleaving)	EDD3 6F84 A262 2E7
(After scrambling)	E55B CF8E 88EA 06D

Third Frame

(Before coding)	7FFF FEC (Upper 26 bits are valid.)
(After interleaving)	61F3 C97E EBEC B84
(After scrambling)	697B 6974 C164 90E

Fourth Frame

(Before coding)	3FC0 000 (Upper 26 bits are valid.)
(After interleaving)	4050 9845 108A F6A
(After scrambling)	48D8 384F 3A02 DE0

-9600 bps/EHR-
SACCH

First Frame

(Before coding)	FF01 000 (Upper 26 bits are valid.)
(After interleaving)	CD6A 1A07 9825 223
(After scrambling)	C5E2 BA0D B2AD 0A9

Second Frame

(Before coding)	BF8B FF8 (Upper 26 bits are valid.)
(After interleaving)	E4D1 2D84 8328 2CC
(After scrambling)	EC59 8D8E A9A0 046

Third Frame

(Before coding)	7FFF FEC (Upper 26 bits are valid.)
(After interleaving)	61F3 C97E EBEC B84
(After scrambling)	697B 6974 C164 90E

Fourth Frame

(Before coding)	3FC0 000 (Upper 26 bits are valid.)
(After interleaving)	4050 9845 108A F6A
(After scrambling)	48D8 384F 3A02 DE0

-9600 bps/EFR-
SACCH

First Frame

(Before coding)	FF01 000 (Upper 26 bits are valid.)
(After interleaving)	CD6A 1A07 9825 223
(After scrambling)	C5E2 BA0D B2AD 0A9

Second Frame

(Before coding)	BF8F FF8 (Upper 26 bits are valid.)
(After interleaving)	E4DB 7D10 A330 FE4
(After scrambling)	EC53 DD1A 89B8 D6E

Third Frame

(Before coding)	7FFF FEC (Upper 26 bits are valid.)
(After interleaving)	61F3 C97E EBEC B84
(After scrambling)	697B 6974 C164 90E

Fourth Frame

(Before coding)	3FC0 000 (Upper 26 bits are valid.)
(After interleaving)	4050 9845 108A F6A
(After scrambling)	48D8 384F 3A02 DE0

4.1.7. VCH

The voice coding methods of Vocoder are two types of EHR and EFR. In EHR, the 49-bit data after the voice coding is converted to 72-bit codeword by FEC processing. In EFR, the 88-bit data after the voice coding is converted to 144-bit codeword by FEC processing. The voice data to be used for testing is the tone test pattern.

The tone test pattern of EHR is a pattern generated by voice-coding a 1031 Hz tone signal.

4 VCHs are transferred in one frame as shown in Figure 4.1-3. The 72-bit voice codeword is common but different scramble patterns are applied to each VCH; hence, 4 data strings for VCHs are generated after scramble processing.

VCH (EHR 1031 Hz Tone Test Pattern)

(After voice coding)		49 bits	FEE2 1212 1210 0 (Upper 49 bits are valid.)
(After FEC)		72 bits	CEA8 FE83 ACC4 5820 0A
(After scrambling)	(1)	72 bits	4CAA DE8B 26E4 F282 88
	(2)	72 bits	C68A 7429 A4EC D008 22
	(3)	72 bits	CEA2 FC01 8CEC DA0A A0
	(4)	72 bits	EE8A 7E2B 26CC F88A 08

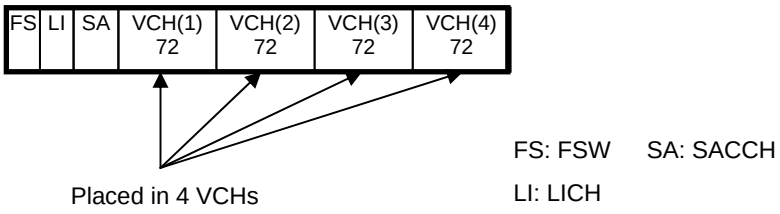


Figure 4.1-3 VCH Mapping (EHR)

The tone test pattern of EFR is a pattern generated by voice-coding a 1011 Hz tone signal. There are two kind of 144-bit voice codeword, (1) and (2), (1) is arranged at VCH of the first half and (2) is arranged at VCH of the second half as shown in Figure 4.1-4. Hence, there are two data strings after scramble processing so that they correspond to 2 VCHs.

VCH (EFR 1011 Hz Tone Test Pattern)

(After voice coding)	(1)	88 bits	09B0 880C C621 F680 A826 00
	(2)	88 bits	09B0 880C C621 F680 A826 01
(After FEC)	(1)	144 bits	3892 8490 D433 C0BE 1B91 844F F058 A589 D839
	(2)	144 bits	3892 8490 D433 C0BE 1B91 844F F058 A589 D83B
(After scrambling)	(1)	144 bits	BA90 A498 5E13 6A1C 9999 A6C5 5A50 8D01 F011
	(2)	144 bits	3898 8612 F41B 4294 B1B1 A6CF 58D2 AD29 7239

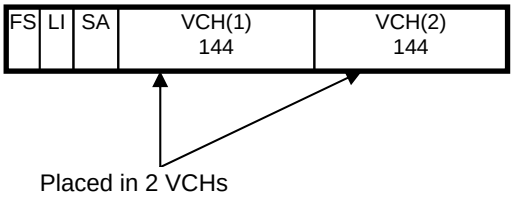


Figure 4.1-4 VCH Mapping (EFR)

4.1.8. FACCH1

An FACCH1 is a functional channel which transfers a control information temporarily stealing audio data on a traffic channel in a trunked radio system and conventional system. A data string before being encoded consists of 80 bits, and the 144-bit FACCH1 is generated by encoding the data string. Messages to be used in the FACCH1 test are VCALL and TX_REL. There are three types of VCALL messages for 4800 bps/EHR, 9600 bps/EHR and 9600 bps/EFR. In addition, since it is possible to insert 2 FACCH1s in one frame and applicable scramble bit sequences are different between FACCH1s, there are two types of data strings after scramble processing even though the messages to be sent are the same.

RAN and ID configurations to be used in this test are three patterns as presented in Table 4.1-3.

<VCALL>

RAN:	000001b/ 100000b/ 111111b
Message Type:	000001b
CC Option:	00
Call Type:	001b
Voice Call Option:	00000b (4800 bps/EHR)
	00010b (9600 bps/EHR)
	00011b (9600 bps/EFR)
Source Unit ID:	0001/ 7FF8/ FFEF
Destination Group ID:	0001/ 7FF8/ FFEF
Cipher Type:	00b
Key ID:	000000b

Pattern 7

-4800 bps-

FACCH1

(Before coding)	0100 2000 0100 0100 0000
(After interleaving)	48B4 0120 9000 0412 09
	0402 E45A 1002 0E41 A0
(After scrambling)	CAB6 2128 1A20 AEB0 8B
First half of FACCH1	0C20 6EF0 182A 8669 88
(After scrambling)	48BE 03A2 B028 8638 A3
Second half of FACCH1	2420 64F2 9A0A AEEB A2

-9600 bps/EHR-

FACCH1

(Before coding)	0100 2200 0100 0100 0000
(After interleaving)	48A4 0124 9009 0010 48
	0502 A43A 1012 0E41 A2
(After scrambling)	CAA6 212C 1A29 AAB2 CA
First half of FACCH1	0D20 2E90 183A 8669 8A
(After scrambling)	48AE 03A6 B021 823A E2
Second half of FACCH1	2520 2492 9A1A AEEB A0

-9600 bps/EFR-
FACCH1

(Before coding)	0100 2300 0100 0100 0000
(After interleaving)	48A4 4920 9009 0492 09
	2592 647A 001A 0645 A2
(After scrambling)	CAA6 6928 1A29 AE30 8B
First half of FACCH1	2DB0 EED0 0832 8E6D 8A
(After scrambling)	48AE 4BA2 B021 86B8 A3
Second half of FACCH1	05B0 E4D2 8A12 A6EF A0

Pattern 8

-4800 bps-
FACCH1

(Before coding)	0100 207F F87F F800 0000
(After interleaving)	5C3C 0003 8141 40E0 40
	3894 201F 019B 8EE5 A0
(After scrambling)	DE3E 200B 0B61 EA42 C2
First half of FACCH1	30B6 AAB5 09B3 06CD 88
(After scrambling)	5C36 0281 A169 C2CA EA
Second half of FACCH1	18B6 A0B7 8B93 2E4F A2

-9600 bps/EHR-
FACCH1

(Before coding)	0100 227F F87F F800 0000
(After interleaving)	5C2C 0007 8148 44E2 01
	3994 607F 018B 8EE5 A2
(After scrambling)	DE2E 200F 0B68 EE40 83
First half of FACCH1	31B6 EAD5 09A3 06CD 8A
(After scrambling)	5C26 0285 A160 C6C8 AB
Second half of FACCH1	19B6 E0D7 8B83 2E4F A0

-9600 bps/EFR-
FACCH1

(Before coding)	0100 237F F87F F800 0000
(After interleaving)	5C2C 4803 8148 4060 40
	1904 A03F 1183 86E 1A2
(After scrambling)	DE2E 680B 0B68 EAC2 C2
First half of FACCH1	1126 2A95 19AB 0EC9 8A
(After scrambling)	5C26 4A81 A160 C24A EA
Second half of FACCH1	3926 2097 9B8B 264B A0

Pattern 9

-4800 bps-
FACCH1

(Before coding)	0100 20FF EFFF EF00 0000
(After interleaving)	5C38 0223 8209 C462 60
	1D96 285F 211B CEA1 B0
(After scrambling)	DE3A 222B 0829 6EC0 E2
First half of FACCH1	15B4 A2F5 2933 4689 98
(After scrambling)	5C32 00A1 A221 4648 CA
Second half of FACCH1	3DB4 A8F7 AB13 6E0B B2

-9600 bps/EHR-

FACCH1

(Before coding)	0100 22FF EFFF EF00 0000
(After interleaving)	5C28 0227 8200 C060 21
	1C96 683F 210B CEA1 B2
(After scrambling)	DE2A 222F 0820 6AC2 A3
First half of FACCH1	14B4 E295 2923 4689 9A
(After scrambling)	5C22 00A5 A228 424A 8B
Second half of FACCH1	3CB4 E897 AB03 6E0B B0

-9600 bps/EFR-

FACCH1

(Before coding)	0100 23FF EFFF EF00 0000
(After interleaving)	5C28 4A23 8200 C4E2 60
	3C06 A87F 3103 C6A5 B2
(After scrambling)	DE2A 6A2B 0820 6E40 E2
First half of FACCH1	3424 22D5 392B 4E8D 9A
(After scrambling)	5C22 48A1 A228 46C8 CA
Second half of FACCH1	1C24 28D7 BB0B 660F B0

<TX_REL>

RAN:	000001b/ 100000b/ 111111b
Message Type:	001000b
CC Option:	00
Call Type:	001b
Source Unit ID:	0001/ 7FF8/ FFEF
Destination Group ID:	0001/ 7FF8/ FFEF

Pattern 7

FACCH1

(Before coding)	0800 2000 0100 0100 0000
(After interleaving)	0814 4120 9008 0612 08
	8502 644A 281A 0C44 A2
(After scrambling)	8A16 6128 1A28 ACB0 8A
First half of FACCH1	8D20 EEE0 2032 846C 8A
(After scrambling)	081E 43A2 B020 8438 A2
Second half of FACCH1	A520 E4E2 A212 ACEE A0

Pattern 8

FACCH1

(Before coding)	0800 207F F87F F800 0000
(After interleaving)	1C9C 4003 8149 42E0 41
	B994 A00F 3983 8CE0 A2
(After scrambling)	9E9E 600B 0B69 E842 C3
First half of FACCH1	B1B6 2AA5 31AB 04C8 8A
(After scrambling)	1C96 4281 A161 C0CA EB
Second half of FACCH1	99B6 20A7 B38B 2C4A A0

Pattern 9

FACCH1

(Before coding)	0800 20FF EFFF EF00 0000
(After interleaving)	1C98 4223 8201 C662 61
	9C96 A84F 1903 CCA4 B2
(After scrambling)	9E9A 622B 0821 6CC0 E3
First half of FACCH1	94B4 22E5 112B 448C 9A
(After scrambling)	1C92 40A1 A229 4448 CB
Second half of FACCH1	BCB4 28E7 930B 6C0E B0

4.1.9. UDCH

A UDCH is a functional channel to transfer the user data on a traffic channel in a trunked radio system and conventional system. A data string before being encoded is 184-bit data consisting of 176-bit layer 3 information and 8-bit SR information. The 384-bit UDCH is generated by encoding the data string. Messages to be used the UDCH test are DCALL (Header) and DCALL (User Data). There are two types of DCALL (Header) for 4800 bps mode and for 9600 bps mode. RAN and ID configurations to be used in this test are three patterns as presented in Table 4.1-4.

	RAN	Source Unit ID	Destination Unit ID
Pattern 10	000001b	0001	0002
Pattern 11	100000b	7FF8	7FF9
Pattern 12	111111b	FFEF	FFEE

Table 4.1-4 RAN and ID Settings for Individual Call

<DCALL(Header)>

RAN:	000001b/ 100000b/ 111111b
Message Type:	001001b
CC Option:	00
Call Type:	100b
Data Call Option:	00000b (4800 bps)
	00010b (9600 bps)
Source Unit ID:	0001/ 7FF8/ FFEF
Destination Unit ID:	0002/ 7FF9/ FFEE
Cipher Type:	00b
Key ID:	000000b
Packet Information:	800400

Pattern 10

-4800 bps-

UDCH

(Before coding) 0109 0080 0001 0002 0080 0400 0000 0000 0000 0000 0000 00

(After interleaving) 4800 0003 4A00 0040 4200 0242 C000 6412 8002 2820 000C
C1A0 00AC 0100 0750 2000 1B11 4000 0428 000E 8540 007

(After scrambling) 4088 A009 6088 28E8 6222 02CA 620A CE3A A080 008A A08E
4922 822C 01A0 2F72 2288 39BB E202 2C22 88AE 0F4A A05

-9600 bps-

UDCH

(Before coding) 0109 0082 0001 0002 0080 0400 0000 0000 0000 0000 0000 00

(After interleaving) 4800 0003 4A00 0000 4200 0452 C000 4492 8003 2820 0004
E1A0 00AC 0100 0358 2000 2B51 4000 0628 0002 9540 005

(After scrambling) 4088 A009 6088 28A8 6222 04DA 620A EEBA A081 008A A086
6922 822C 01A0 2B7A 2288 09FB E202 2E22 88A2 1F4A A07

Pattern 11

-4800 bps-

UDCH

(Before coding) 2009 0080 7FF8 7FF9 0080 0400 0000 0000 0000 0000 0000 00

(After interleaving) 0FC0 0001 6800 0000 F200 054F 4000 4816 8002 0A60 001A
CCA0 0044 B900 0090 6000 1769 4001 A7C8 0001 8440 004

(After scrambling) 0748 A00B 4288 28A8 D222 05C7 E20A E23E A080 22CA A098
4422 82C4 B9A0 28B2 6288 35C3 E203 8FC2 88A1 0E4A A06

-9600 bps-

UDCH

(Before coding) 2009 0082 7FF8 7FF9 0080 0400 0000 0000 0000 0000 0000 00

(After interleaving) 0FC0 0001 6800 0040 F200 035F 4000 6896 8003 0A60 0012
ECA0 0044 B900 0498 6000 2729 4001 A5C8 000D 9440 006

(After scrambling) 0748 A00B 4288 28E8 D222 03D7 E20A C2BE A081 22CA A090
6422 82C4 B9A0 2CBA 6288 0583 E203 8DC2 88AD 1E4A A04

Pattern 12

-4800 bps-

UDCH

(Before coding) 3F09 0080 FFEF FFEE 0080 0400 0000 0000 0000 0000 0000 00

(After interleaving) 4BC0 0011 6000 00C1 6200 054F 4000 5826 8003 CBA0 000D
CBA0 00FD B100 06D8 A000 0B6D 4000 E7C8 0008 9040 004

(After scrambling) 4348 A01B 4A88 2869 4222 05C7 E20A F20E A081 E30A A08F
4322 827D B1A0 2EFA A288 29C7 E202 CFC2 88A8 1A4A A06

-9600 bps-

UDCH

(Before coding) 3F09 0082 FFEF FFEE 0080 0400 0000 0000 0000 0000 0000 00

(After interleaving) 4BC0 0011 6000 0081 6200 035F 4000 78A6 8002 CBA0 0005
EBA0 00FD B100 02D0 A000 3B2D 4000 E5C8 0004 8040 006

(After scrambling) 4348 A01B 4A88 2829 4222 03D7 E20A D28E A080 E30A A087
6322 827D B1A0 2AF2 A288 1987 E202 CDC2 88A4 0A4A A04

<DCALL(User Data)>

```
    RAN:                  000001b/ 100000b/ 111111b
    Message Type:         001011b
    Packet Frame Number:  0
    Block Number:         0
    User Data Area:       ABCDEFGHIJKLMNOP
```

```
Pattern 10
    UDCH
        (Before coding)       010B 0041 4243 4445 4647 4849 4A4B 4C4D 4E4F 5053 D6F8 30
        (After interleaving)  6EE2 C01A 007E FB43 16C5 B84E 683E 264F B0B1 B7BD 4330
                              BB15 A38A 5F66 9E7E 1CEF 0B52 79C1 420D 81D0 BB10 EEE
        (After scrambling)    666A 6010 2AF6 D3EB 36E7 B8C6 CA34 8C67 9033 9F17 E3B2
                              3397 210A 5FC6 B65C 1E67 29F8 DBC3 6A07 0970 311A 4EC
Pattern 11
    UDCH
        (Before coding)       200B 0041 4243 4445 4647 4849 4A4B 4C4D 4E4F 5053 D6F8 30
        (After interleaving)  2EE2 C018 007E FB83 16C5 B94E 683E 3A4F B0B1 17BD 433E
                              BB15 A3A2 5F66 98BE 1CEF 2752 79C0 A20D 81DD BB10 EE8
        (After scrambling)    266A 6012 2AF6 D32B 36E7 B9C6 CA34 9067 9033 3F17 E3BC
                              3397 2122 5FC6 B09C 1E67 05F8 DBC2 8A07 097D 311A 4EA
Pattern 12
    UDCH
        (Before coding)       3F0B 0041 4243 4445 4647 4849 4A4B 4C4D 4E4F 5053 D6F8 30
        (After interleaving)  6EE2 C000 007E FBC3 16C5 BF4E 683E 0A4F B0B1 57BD 4335
                              BB15 A39A 5F66 99FE 1CEF 2352 79C1 620D 81DC BB10 EEC
        (After scrambling)    666A 600A 2AF6 D36B 36E7 BFC6 CA34 A067 9033 7F17 E3B7
                              3397 211A 5FC6 B1DC 1E67 01F8 DBC3 4A07 097C 311A 4EE
```

4.1.10. FACCH2

An FACCH2 is a functional channel which transfers a control information temporarily stealing the user data on a traffic channel in a trunked radio system and conventional system. A data string before being encoded is 184-bit data consisting of 176-bit layer 3 information and 8-bit SR information. The 384-bit FACCH2 is generated by encoding the data string. Messages to be used in the FACCH2 test are STAT_REQ and TX_REL.

RAN and ID configurations to be used in this test are three patterns as presented in Table 4.1-4.

<STAT_REQ>

```
    RAN:                  000001b/ 100000b/ 111111b
    Message Type:         110010b
    CC Option:            00
    Call Type:            100b
    Status Call Option    01000b
    Source Unit ID:       0001/ 7FF8/ FFEF
    Destination Unit ID:  0002/ 7FF9/ FFEE
    Status:               01
```

Pattern 10

FACCH2

(Before coding)	0132 0088 0001 0002 0001 0000 0000 0000 0000 0000 0000 00
(After interleaving)	6000 001A 0A00 0008 4000 0002 A000 4691 0002 1C20 0005
	4140 0085 0000 0278 1000 2010 8001 1424 0006 8520 001
(After scrambling)	6888 A010 2088 28A0 6022 008A 020A ECB9 2080 348A A087
	C9C2 8205 00A0 2A5A 1288 02BA 2203 3C2E 88A6 0F2A A03

Pattern 11

FACCH2

(Before coding)	2032 0088 7FF8 7FF9 0001 0000 0000 0000 0000 0000 0000 00
(After interleaving)	27C0 0018 2800 0048 F000 070F 2000 6A95 0002 3E60 0013
	4C40 006D B800 05B8 5000 2C68 8000 B7C4 0009 8420 002
(After scrambling)	2F48 A012 0288 28E0 D022 0787 820A C0BD 2080 16CA A091
	C4C2 82ED B8A0 2D9A 5288 0EC2 2202 9FCE 88A9 0E2A A00

Pattern 12

FACCH2

(Before coding)	3F32 0088 FFEF FFEE 0001 0000 0000 0000 0000 0000 0000 00
(After interleaving)	63C0 0008 2000 0089 6000 070F 2000 7AA5 0003 FFA0 0004
	4B40 00D4 B000 03F0 9000 306C 8001 F7C4 0000 9020 002
(After scrambling)	6B48 A002 0A88 2821 4022 0787 820A D08D 2081 D70A A086
	C3C2 8254 B0A0 2BD2 9288 12C6 2203 DFCE 88A0 1A2A A00

<TX_REL>

RAN:	000001b/ 100000b/ 111111b
Message Type:	001000b
CC Option:	00
Call Type:	100b
Source Unit ID:	0001/ 7FF8/ FFEE
Destination Group ID:	0002/ 7FF9/ FFEF

Pattern 10

FACCH2

(Before coding)	0108 0080 0001 0002 0000 0000 0000 0000 0000 0000 0000 00
(After interleaving)	4800 0003 4A00 0008 4000 0042 8000 6610 0002 2820 0008
	4100 0088 0000 0650 0000 1210 0000 0420 0008 8500 004
(After scrambling)	4088 A009 6088 28A0 6022 00CA 220A CC38 2080 008A A08A
	C982 8208 00A0 2E72 0288 30BA A202 2C2A 88A8 0F0A A06

Pattern 11

FACCH2

(Before coding)	2008 0080 7FF8 7FF9 0000 0000 0000 0000 0000 0000 0000 00
(After interleaving)	0FC0 0001 6800 0048 F000 074F 0000 4A14 0002 0A60 001E
	4C00 0060 B800 0190 4000 1E68 0001 A7C0 0007 8400 007
(After scrambling)	0748 A00B 4288 28E0 D022 07C7 A20A E03C 2080 22CA A09C
	C482 82E0 B8A0 29B2 4288 3CC2 A203 8FCA 88A7 0E0A A05

Pattern 12

FACCH2

(Before coding)	3F08 0080 FFEF FFEE 0000 0000 0000 0000 0000 0000 0000 00
(After interleaving)	4BC0 0011 6000 0089 6000 074F 0000 5A24 0003 CBA0 0009
	4B00 00D9 B000 07D8 8000 026C 0000 E7C0 000E 9000 007
(After scrambling)	4348 A01B 4A88 2821 4022 07C7 A20A F00C 2081 E30A A08B
	C382 8259 B0A0 2FFA 8288 20C6 A202 CFCA 88AE 1A0A A05

4.2. Control Channel Format Tests for Receiver

This section verifies whether a receiver can properly decode functional channels to be used on an RCCH. Figure 4.2-1 shows a block diagram of the tests.

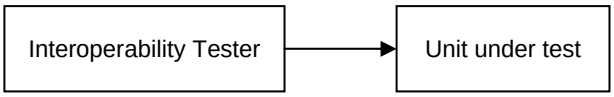


Figure 4.2-1 Receiver Tests Setup

4.2.1. CAC Test

Since a CAC is a channel to be used for outbound signals, this test is applicable only to an SU.

Send out frames consisting of the following data from an interoperability tester.

- A CAC data string after scramble as presented in Section 4.1.2
- An LICH data string for CAC/ collision control field testing after scramble as presented in Section 4.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- A CAC data string before being encoded as presented in Section 4.1.2
- An LICH data string for CAC/ collision control field testing before dibit processing as presented in Section 4.1.1

Besides, a test as presented in Section 4.2.4 can be processed at the same time.

4.2.2. Long CAC Test

Since a Long CAC is a channel to be used for inbound signals, this test is applicable only to a TR.

Send out frames consisting of the following data from an interoperability tester.

- A Long CAC data string after scramble as presented in Section 4.1.3
- An LICH data string for Long CAC testing after scramble as presented in Section 4.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- A Long CAC data string before being encoded as presented in Section 4.1.3
- An LICH data string for Long CAC testing before dibit processing as presented in Section 4.1.1

4.2.3. Short CAC Test

Since a Short CAC is a channel to be used for inbound signals, this test is applicable only to a TR.

Send out frames consisting of the following data from an interoperability tester.

- A Short CAC data string after scramble as presented in Section 4.1.4
- An LICH data string for Short CAC testing after scramble as presented in Section 4.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- A Short CAC data string before being encoded as presented in Section 4.1.4
- An LICH data string for Short CAC testing before dibit processing as presented in Section 4.1.1

4.2.4. Collision Control Field Test

Since a collision control field is a field to be used for outbound signals, this test is applicable only to an SU.

Send out frames consisting of the following data from an interoperability tester.

- A data string of a collision control field after scramble as presented in Section 4.1.5
- An LICH data string for CAC/ collision control field testing after scramble as presented in Section 4.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- A data string of a collision control field before being encoded as presented in Section 4.1.5
- An LICH data string for CAC/ collision control field testing before dibit processing as presented in Section 4.1.1

This test shall be done for a case when no inbound signal is received and for a case when an inbound signal is received.
In the case that no inbound signal is received (while in the idle state), the setting value of a collision control field is the value used in when the PE is set to null as presented in Section 4.1.5, and in the case that an inbound signal is successfully received, the setting value is the value used in when the PE is the same as the received CRC as presented in Section 4.1.5.
A test as presented in Section 4.2.1 can be processed at the same time.

4.3. Voice Format Tests for Receiver

This section verifies whether a receiver can properly decode functional channels which are used for voice communications on an RDCH/RTCH. A block diagram of the tests is presented in Figure 4.2-1.

4.3.1. SACCH Test

Since an SACCH is a channel to be used in both of an inbound and outbound direction, this test is applicable to both CR/TR and SU. This test shall be done for a superframe structure in which a VCALL message is used and for a non-superframe structure in which an IDLE message is used.

In the case of the superframe structure, send out frames consisting of the following data from an interoperability tester.

- An SACCH (VCALL) data string after scramble as presented in Section 4.1.6.
- An LICH data string for VCH/ SACCH testing after scramble as presented in Section 4.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- An SACCH (VCALL) data string before being encoded as presented in Section 4.1.6
- An LICH data string for VCH/ SACCH testing before dibit processing as presented in Section 4.1.1

In the case of the non-superframe structure, send out frames consisting of the following data from an interoperability tester.

- An SACCH (IDLE) data string after scramble as presented in Section 4.1.6.
- An LICH data string for FACCH1/ SACCH testing (SACCH with non-superframe) after scramble as presented in Section 4.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- An SACCH (IDLE) data string before being encoded as presented in Section 4.1.6
- An LICH data string for FACCH1/ SACCH testing (SACCH with non-superframe) before dibit processing as presented in Section 4.1.1

4.3.2. VCH Test

Since a VCH is a channel to be used in both of an inbound and outbound direction, this test is applicable to both CR/TR and SU.

Send out frames consisting of the following data from an interoperability tester.

- A VCH data string after scramble as presented in Section 4.1.7
- An LICH data string for VCH/ SACCH testing after scramble as presented in Section 4.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- A VCH data string after voice coding processing as presented in Section 4.1.7
- An LICH data string for VCH/ SACCH testing before dibit processing as presented in Section 4.1.1

4.3.3. FACCH1 Test

Since an FACCH1 is a channel to be used in both of an inbound and outbound direction, this test is applicable to both CR/TR and SU. There are two tests, one is the test using a superframe structure with VCALL message and the other is the test using a non-superframe structure with TX_REL message. The first and second FACCH1s are intended to be tested since all VCHs are replaced by two FACCH1s.

In the case of the superframe structure, send out frames consisting of the following data from an interoperability tester.

- An FACCH1(VCALL) data string after scramble as presented in Section 4.1.8
- An LICH data string for FACCH1/ SACCH testing (SACCH with superframe) after scramble as presented in Section 4.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- An FACCH1(VCALL) data string before being encoded as presented in Section 4.1.8
- An LICH data string for FACCH1/ SACCH testing (SACCH with superframe) before dibit processing as presented in Section 4.1.1

In the case of the non-superframe structure, send out frames consisting of the following data from an interoperability tester.

- An FACCH1(TX_REL) data string after scramble as presented in Section 4.1.8
- An LICH data string for FACCH1/ SACCH testing (SACCH with non-superframe) after scramble as presented in Section 4.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- An FACCH1(TX_REL) data string before being encoded as presented in Section 4.1.8
- An LICH data string for FACCH1/ SACCH testing (SACCH with non-superframe) before dibit processing as presented in Section 4.1.1

4.4. Data Format Tests for Receiver

This section verifies whether a receiver can properly decode functional channels which are used for data communications on an RDCH/RTCH. A block diagram of the tests is presented in Figure 4.2-1.

4.4.1. UDCH Test

Since a UDCH is a channel to be used in both of an inbound and outbound direction, this test is applicable to both CR/TR and SU.

Send out frames consisting of the following data from an interoperability tester.

- A UDCH data string after scramble as presented in Section 4.1.9
- An LICH data string for UDCH testing after scramble as presented in Section 4.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- A UDCH data string before being encoded as presented in Section 4.1.9
- An LICH data string for UDCH testing before dibit processing as presented in Section 4.1.1

4.4.2. FACCH2 Test

Since an FACCH2 is a channel to be used in both of an inbound and outbound direction, this test is applicable to both CR/TR and SU.

Send out frames consisting of the following data from an interoperability tester.

- An FACCH2 data string after scramble as presented in Section 4.1.10
- An LICH data string for FACCH2 testing after scramble as presented in Section 4.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- An FACCH2 data string before being encoded as presented in Section 4.1.10
- An LICH data string for FACCH2 testing before dibit processing as presented in Section 4.1.1

4.5. Control Channel Format Tests for Transmitter

This section verifies whether a transmitter can properly encode functional channels to be used on an RCCH. Figure 4.5-1 shows a block diagram of the tests.

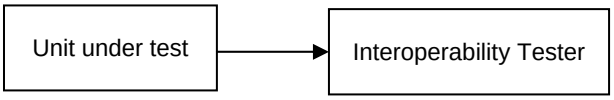


Figure 4.5-1 Transmitter Tests Setup

4.5.1. CAC Test

Since a CAC is a channel to be used for outbound signals, this test is applicable only to a TR.

Send out frames consisting of the following data from a unit under test.

- A CAC data string before being encoded as presented in Section 4.1.2
- An LICH data string for CAC/ collision control field testing before dibit processing as presented in Section 4.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- A CAC data string after scramble as presented in Section 4.1.2
- An LICH data string for CAC/ collision control field testing after scramble as presented in Section 4.1.1

Besides, a test as presented in Section 4.5.4 can be processed at the same time.

4.5.2. Long CAC Test

Since a Long CAC is a channel to be used for inbound signals, this test is applicable only to an SU.

Send out frames consisting of the following data from a unit under test.

- A Long CAC data string before being encoded as presented in Section 4.1.3
- An LICH data string for Long CAC testing before dibit processing as presented in Section 4.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- A Long CAC data string after scramble as presented in Section 4.1.3
- An LICH data string for Long CAC testing after scramble as presented in Section 4.1.1

4.5.3. Short CAC Test

Since a Short CAC is a channel to be used for inbound signals, this test is applicable only to an SU.

Send out frames consisting of the following data from a unit under test.

- A Short CAC data string before being encoded as presented in Section 4.1.4
- An LICH data string for Short CAC testing before dibit processing as presented in Section 4.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- A Short CAC data string after scramble as presented in Section 4.1.4
- An LICH data string for Short CAC testing after scramble as presented in Section 4.1.1

4.5.4. Collision Control Field Test

Since a collision control field is a field to be used for outbound signals, this test is applicable only to a TR.

Send out frames consisting of the following data from a unit under test.

- A data string of a collision control field before dibit processing as presented in Section 4.1.5
- An LICH data string for CAC/ collision control field testing before dibit processing as presented in Section 4.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- A data string of a collision control field after scramble as presented in Section 4.1.5
- An LICH data string for CAC/ collision control field testing after scramble as presented in Section 4.1.1

This test shall be done for a case when no inbound signal is received and for a case when an inbound signal is received.
In the case that no inbound signal is received (while in the idle state), the setting value of a collision control field is the value used in when the PE is set to null as presented in Section 4.1.5, and in the case that an inbound signal is successfully received, the setting value is the value used in when the PE is the same as the received CRC as presented in Section 4.1.5.
A test as presented in Section 4.5.1 can be processed at the same time.

4.6. Voice Format Tests for Transmitter

This section verifies whether a transmitter can properly encode functional channels which are used for voice communications on an RDCH/RTCH. A block diagram of the tests is presented in Figure 4.5-1.

4.6.1. SACCH Test

Since an SACCH is a channel to be used in both of an inbound and outbound direction, this test is applicable to both CR/TR and SU.

This test shall be done for a superframe structure in which a VCALL message is used and for a non-superframe structure in which an IDLE message is used.

In the case of the superframe structure, send out frames consisting of the following data from a unit under test.

- An SACCH (VCALL) data string before being encoded as presented in Section 4.1.6
- An LICH data string for VCH/ SACCH testing before dibit processing as presented in Section 4.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- An SACCH (VCALL) data string after scramble as presented in Section 4.1.6.
- An LICH data string for VCH/ SACCH testing after scramble as presented in Section 4.1.1

In the case of the non-superframe structure, send out frames consisting of the following data from a unit under test.

- An SACCH (IDLE) data string before being encoded as presented in Section 4.1.6
- An LICH data string for FACCH1/ SACCH testing (SACCH with non-superframe) before dibit processing as presented in Section 4.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- An SACCH (IDLE) data string after scramble as presented in Section 4.1.6.
- An LICH data string for FACCH1/ SACCH testing (SACCH with non-superframe) after scramble as presented in Section 4.1.1

4.6.2. VCH Test

Since a VCH is a channel to be used in both of an inbound and outbound direction, this test is applicable to both CR/TR and SU.

Send out frames consisting of the following data from a unit under test.

- A VCH data string after voice coding processing as presented in Section 4.1.7
- An LICH data string for VCH/ SACCH testing before dibit processing as presented in Section 4.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- A VCH data string after scramble as presented in Section 4.1.7
- An LICH data string for VCH/ SACCH testing after scramble as presented in Section 4.1.1

4.6.3. FACCH1 Test

Since an FACCH1 is a channel to be used in both of an inbound and outbound direction, this test is applicable to both CR/TR and SU. There are two tests, one is the test using a superframe structure with VCALL message and the other is the test using a non-superframe structure with TX_REL message. The first and second FACCH1s are intended to be tested since all VCHs are replaced by two FACCH1s.

In the case of the superframe structure, send out frames consisting of the following data from a unit under test.

- An FACCH1(VCALL) data string before being encoded as presented in Section 4.1.8
- An LICH data string for FACCH1/ SACCH testing (SACCH with superframe) before dibit processing as presented in Section 4.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- An FACCH1(VCALL) data string after scramble as presented in Section 4.1.8
- An LICH data string for FACCH1/ SACCH testing (SACCH with superframe) after scramble as presented in Section 4.1.1

In the case of the non-superframe structure, send out frames consisting of the following data from a unit under test.

- An FACCH1(TX_REL) data string before being encoded as presented in Section 4.1.8
- An LICH data string for FACCH1/ SACCH testing (SACCH with non-superframe) before dibit processing as presented in Section 4.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- An FACCH1(TX_REL) data string after scramble as presented in Section 4.1.8
- An LICH data string for FACCH1/ SACCH testing (SACCH with non-superframe) after scramble as presented in Section 4.1.1

4.7. Data Format Tests for Transmitter

This section verifies whether a transmitter can properly encode functional channels which are used data communications on an RDCH/RTCH. Figure 4.5-1 shows a block diagram of the tests.

4.7.1. UDCH Test

Since a UDCH is a channel to be used in both of an inbound and outbound direction, this test is applicable to both CR/TR and SU.

Send out frames consisting of the following data from a unit under test.

- A UDCH data string before being encoded as presented in Section 4.1.9
- An LICH data string for UDCH testing before dibit processing as presented in Section 4.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- A UDCH data string after scramble as presented in Section 4.1.9
- An LICH data string for UDCH testing after scramble as presented in Section 4.1.1

4.7.2. FACCH2 Test

Since an FACCH2 is a channel to be used in both of an inbound and outbound direction, this test is applicable to both CR/TR and SU.

Send out frames consisting of the following data from a unit under test.

- An FACCH2 data string before being encoded as presented in Section 4.1.10
- An LICH data string for FACCH2 testing before dibit processing as presented in Section 4.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- An FACCH2 data string after scramble as presented in Section 4.1.10
- An LICH data string for FACCH2 testing after scramble as presented in Section 4.1.1

4.8. Preamble and FSW Test for Transmitter

This section verifies whether a transmitter can properly transmit a preamble and FSW in an RDCH or FSW in an RCCH/RTCH. Figure 4.5-1 shows a block diagram of the tests. This test is applicable to both CR/TR and SU.

When a unit under test starts transmitting, verify that symbol strings received by an interoperability tester are identical with the preamble pattern and FSW pattern specified in REF [1].

4.9. Post Field Test for Transmitter

This section verifies whether a transmitter can properly transmit a post field in the outbound RCCH. Figure 4.5-1 shows a block diagram of the tests. This test is applicable to a TR.

When a unit under test starts transmitting, verify that a symbol string of the post field received by an interoperability tester is identical with the post field pattern specified in REF [1].

5. Common Air Interface Tests (Type-D Trunked System)

This section describes examinations of CAI to be used in Type-D separately from Section 4. Because it is similar to what is written down in Section 4, the basic setting (e.g., examples of the configuration for the encoder test and the decoder test and the examination conditions) is omitted.

The relation between the operational mode for a unit under test and the test items in Section 5 is following list. An examiner performs required tests according to operational modes implemented in the unit under test.

	Operational Mode	Test Items in Section 5					
		5.2	5.3	5.4	5.5	5.6	5.7
SU	4800bps Trunked Mode	x	x	x	x	x	-
TR	4800bps Trunked Mode	x	x	x	x	x	x

Relation between Mode of unit under test and Test Items

5.1. Test Parameters

This section defines parameters which are utilized in testing defined in Section 5.2 and afterwards. The data strings before and after encoding associated with parameters for each functional channel are shown below. Because it is similar to what is written in Section 4.1, notation of data string and transmit sequence of data string is omitted.

ID settings to be used in tests are presented in Table 5.1-1. It is tested by a combination of ID of 3 patterns according to the Table.

Single Trunked	Inbound	Outbound
Repeater in Use	1	-
Go to Repeater	-	1
Free Repeater 1	2	
Free Repeater 2	3	
Home Repeater	1	
Call Type	Individual	
Source Unit ID	1 (pattern 1), 3E7 (pattern 2),7D0 (pattern 3)	
Destination Unit ID	2 (pattern 1),3E8 (pattern 2),7CF (pattern 3)	

Table 5.1-1 ID Settings for Calls

5.1.1. LICH

An LICH is a channel placed in all frames and utilized for discrimination of kind of radio channels and functional channels. LICH is also a channel to discriminate the configuration of the frame associated with them. The LICH is 8-bit data consisting of a 7-bit information bit and a 1-bit parity bit, and the LICH after being encoded has 16 bits length by a dibit conversion.

LICH contains a bit representing inbound or outbound direction. Figure 5.1-1 represents the direction of a functional channel to be used in a trunked radio system. Since functional channels are used in both of inbound and outbound directions, the LICH configurations need both directions too.

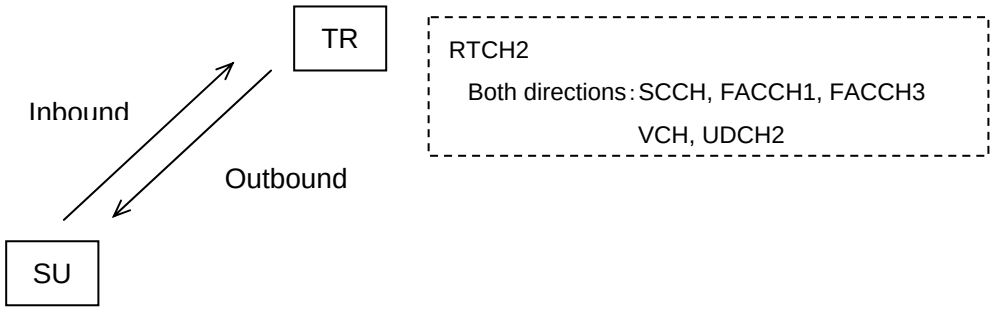


Figure 5.1-1 Direction of Functional Channel on Trunked System

Following are LICH to be tested:

for Link Establish Test		
Inbound Direction	(Before dibit)	C4
	(After dibit)	F575
	(After scrambling)	FD5F
for VCH/SCCH Test (SCCH superframe, VCH)		
Inbound Direction	(Before dibit)	ED
	(After dibit)	FDF7
	(After scrambling)	F5DD
Outbound Direction	(Before dibit)	EF
	(After dibit)	FDFF
	(After scrambling)	F5D5
FACCH1/SCCH Test (SCCH non-superframe, FACCH1)の LICH		
Inbound Direction	(Before dibit)	C0
	(After dibit)	F555
	(After scrambling)	FD7F
Outbound Direction	(Before dibit)	C3
	(After dibit)	F55D
	(After scrambling)	FD77
FACCH3/SCCH Test (FACCH3)の LICH		
Inbound Direction	(Before dibit)	D1
	(After dibit)	F757
	(After scrambling)	FF7D
Outbound Direction	(Before dibit)	D3
	(After dibit)	F75F
	(After scrambling)	FF75
UDCH2/SCCH Test(UDCH2)の LICH		
Inbound Direction	(Before dibit)	DD
	(After dibit)	F7F7
	(After scrambling)	FFDD
Outbound Direction	(Before dibit)	DF
	(After dibit)	F7FF
	(After scrambling)	FFD5

5.1.2. SCCH

A frame on a RTCH2 in a trunked system always accompanies an SCCH.

By SCCH, a trunked system performs various call processing, call termination processing. A SCCH is used to notify SU of information of TR. A data string before being encoded is 25-bit data consisting of layer 3 information. The 60-bit SCCH is generated by encoding the data string.

Messages to be used in SCCH test are INFO1, INFO2, INFO3, and INFO4. The SCCH to send other than VCH message shall employ a non-superframe structure, and the SCCH to send a VCH message shall employ a superframe structure. ID configurations to be used in this test are three patterns as presented in Table 5.1-1. Because Source Unit ID/Destination Unit ID is not included in an information element about INFO1, it is only 1 pattern.

Inbound

[INFO1]

Structure Type	11b
Area	0b
Repeater in Use	1
Pass Character	1F
Call Option	000b
Cipher Type	00b
Key ID	00 0000b

INFO1

SCCH

(Before coding)	C1F8 000 (Upper 25 bits are valid.)
(After interleaving)	CE78 D647 4BCC 82B
(After scrambling)	C6F0 764D 6144 AA1

[INFO2]

Structure Type	10b
Area	0b
Repeater in Use	1
Home Repeater	1
Destination Unit ID	2 / 3E8 / 7CF
G/U	1b

pattern 1

SCCH

(Before coding)	8108 028 (Upper 25 bits are valid.)
(After interleaving)	CFC7 5764 38D2 00A
(After scrambling)	C74F F76E 125A 280

pattern 2

SCCH

(Before coding)	810B E88 (Upper 25 bits are valid.)
(After interleaving)	EE81 3460 D8DB 4C9
(After scrambling)	E609 946A F253 643

pattern 3

SCCH

(Before coding)	810F CF8 (Upper 25 bits are valid.)
(After interleaving)	FECF 4574 1FDA 9E3
(After scrambling)	F647 E57E 3552 B69

[INFO3]

Structure Type	01b
Area	0b
Repeater in Use	1
Home Repeater	1
Source Unit ID	1 / 3E7 / 7D0

pattern 1

SCCH

(Before coding)	4108 010 (Upper 25 bits are valid.)
(After interleaving)	4B63 76AE 1A4E C09
(After scrambling)	43EB D6A4 30C6 E83

pattern 2

SCCH

(Before coding)	410B E70 (Upper 25 bits are valid.)
(After interleaving)	6267 35A2 FA4F 44D
(After scrambling)	6AEF 95A8 D0C7 6C7

pattern 3

SCCH

(Before coding)	410F D00 (Upper 25 bits are valid.)
(After interleaving)	62EB 6436 3C7F 547
(After scrambling)	6A63 C43C 16F7 7CD

[INFO4]

Structure Type	00b
Area	0b
Repeater in Use	1
	1F (In the case of EOT)
Home Repeater	1
Destination Unit ID	2 / 3E8 / 7CF
G/U	1b

pattern 1

SCCH

(Before coding)	0108 028 (Upper 25 bits are valid.)
(After interleaving)	4BC4 47A4 295A 02B
(After scrambling)	434C E7AE 03D2 2A1

pattern 2

SCCH

(Before coding)	010B E88 (Upper 25 bits are valid.)
(After interleaving)	6A82 24A0 C953 4E8
(After scrambling)	620A 84AA E3DB 662

pattern 3

SCCH

(Before coding)	010F CF8 (Upper 25 bits are valid.)
(After interleaving)	7ACC 55B4 0E52 9C2
(After scrambling)	7244 F5BE 24DA B48

In the case of EOT

pattern 1

SCCH

(Before coding)	1F08 028 (Upper 25 bits are valid.)
(After interleaving)	4984 53C7 3946 C3A
(After scrambling)	410C F3CD 13CE EB0

pattern 2

SCCH

(Before coding)	1F0B E88 (Upper 25 bits are valid.)
(After interleaving)	68C2 30C3 D94F 8F9
(After scrambling)	604A 90C9 F3C7 A73

pattern 3

SCCH

(Before coding)	1F0F CF8 (Upper 25 bits are valid.)
(After interleaving)	788C 41D7 1E4E 5D3
(After scrambling)	7004 E1DD 34C6 759

Outbound

[INFO1]

Structure Type	11b
Area	0b
Free Repeater 1	2
Free Repeater 2	3
Call Option	000b
Cipher Type	00b
Key ID	00 0000b

INFO1

SCCH

(Before coding)	C218 000 (Upper 25 bits are valid.)
(After interleaving)	A66A C267 28DC 653
(After scrambling)	AEE2 626D 0254 4D9

[INFO2](Busy Repeater Message)

Structure Type	10b
Area	0b
Go to Repeater	1
Home Repeater	1
Destination Unit ID	2 / 3E8 / 7CF
G/U	1b

pattern 1

SCCH

(Before coding)	8108 028 (Upper 25 bits are valid.)
(After interleaving)	CFC7 5764 38D2 00A
(After scrambling)	C74F F76E 125ª 280

pattern 2

SCCH

(Before coding)	810B E88 (Upper 25 bits are valid.)
(After interleaving)	EE81 3460 D8DB 4C9
(After scrambling)	E609 946A F253 643

pattern 3

SCCH

(Before coding)	810F CF8 (Upper 25 bits are valid.)
(After interleaving)	FECF 4574 1FDA 9E3
(After scrambling)	F647 E57E 3552 B69

[INFO3]

Structure Type	01b
Area	0b
Free Repeater 1	2
Home Repeater	1
Source Unit ID	1 / 3E7 / 7D0

pattern 1

SCCH

(Before coding)	4208 010 (Upper 25 bits are valid.)
(After interleaving)	0B63 E2AF 7156 E38
(After scrambling)	03EB 42A5 5BDE CB2

pattern 2

SCCH

(Before coding)	420B E70 (Upper 25 bits are valid.)
(After interleaving)	2267 A1A3 9157 67C
(After scrambling)	2AEF 01A9 BBDF 4F6

pattern 3

SCCH

(Before coding)	420F D00 (Upper 25 bits are valid.)
(After interleaving)	22EB F037 5767 776
(After scrambling)	2A63 503D 7DEF 5FC

[INFO4](Busy Repeater Message)

Structure Type	00b
Area	0b
Go to Repeater	1, 1F (In the case of EOT)
Home Repeater	1
Destination Unit ID	2 / 3E8 / 7CF
G/U	1b

pattern 1

SCCH

(Before coding)	0108 028 (Upper 25 bits are valid.)
(After interleaving)	4BC4 47A4 295A 02B
(After scrambling)	434C E7AE 03D2 2A1

pattern 2

SCCH

(Before coding)	010B E88 (Upper 25 bits are valid.)
(After interleaving)	6A82 24A0 C953 4E8
(After scrambling)	620A 84AA E3DB 662

pattern 3

SCCH

(Before coding)	010F CF8 (Upper 25 bits are valid.)
(After interleaving)	7ACC 55B4 0E52 9C2
(After scrambling)	7244 F5BE 24DA B48

In the case of EOT

pattern 1

SCCH

(Before coding)	1F08 028 (Upper 25 bits are valid.)
(After interleaving)	4984 53C7 3946 C3A
(After scrambling)	410C F3CD 13CE EB0

pattern 2

SCCH

(Before coding)	1F0B E88 (Upper 25 bits are valid.)
(After interleaving)	68C2 30C3 D94F 8F9
(After scrambling)	604A 90C9 F3C7 A73

pattern 3

SCCH

(Before coding)	1F0F CF8 (Upper 25 bits are valid.)
(After interleaving)	788C 41D7 1E4E 5D3
(After scrambling)	7004 E1DD 34C6 759

[INFO4](Free Repeater Message)

Structure Type	00b
Area	0b
Free Repeater1	2
Free Repeater2	3
ID	7FC

SCCH

(Before coding)	021F FC0 (Upper 25 bits are valid.)
(After interleaving)	0344 C039 8860 3DE
(After scrambling)	0BCC 6033 A2E8 154

5.1.3. VCH

The voice coding methods of Vocoder is EHR. In EHR, the 49-bit data after the voice coding is converted to 72-bit codeword by FEC processing. The voice data to be used for testing is the tone test pattern. The tone test pattern of EHR is a pattern generated by voice-coding a 1031 Hz tone signal.

4 VCHs are transferred in one frame as shown in Figure 5.1-2. The 72-bit voice codeword is common but different scramble patterns are applied to each VCH; hence, 4 data strings for VCHs are generated after scramble processing.

VCH (EHR 1031 Hz Tone Test Pattern)

(After voice coding)		49 bits	FEE2 1212 1210 0 (Upper 49 bits are valid.)
(After FEC)		72 bits	CEA8 FE83 ACC4 5820 0A
(After scrambling)	(1)	72 bits	4CAA DE8B 26E4 F282 88
	(2)	72 bits	C68A 7429 A4EC D008 22
	(3)	72 bits	CEA2 FC01 8CEC DA0A A0
	(4)	72 bits	EE8A 7E2B 26CC F88A 08

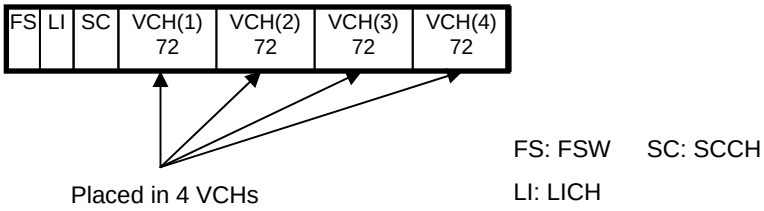


Figure 5.1-2 VCH Mapping

5.1.4. FACCH1

An FACCH1 is a functional channel which transfers a control information temporarily stealing audio data on a traffic channel in a trunked radio system and conventional system. A data string before being encoded consists of 80 bits, and the 144-bit FACCH1 is generated by encoding the data string. Messages to be used in the FACCH1 test are VCALL and TX_REL. Since it is possible to insert 2 FACCH1s in one frame and applicable scramble bit sequences are different between FACCH1s, there are two types of data strings after scramble processing even though the messages to be sent are the same. ID configurations to be used in this test are three patterns as presented in Table 5.1-1.

[VCALL]

Message Type :	00 0001b
CC Option :	00
Call Type :	100b (Individual Call)
Voice Call Option :	00000b (4800bps/EHR)
Home Repeater :	1
Source Unit ID :	1 / 3E7 / 7D0
Destination Unit ID :	2 / 3E8 / 7CF
Cipher Type :	00b
Key ID :	000000b

Pattern 1

FACCH1

(Before coding)	0100 8008 0108 0200 0000
(After interleaving)	5826 4200 9080 2422 09
	4800 BC52 3608 03C1 20
(After scrambling)	DA24 6208 1AA0 8E80 8B
First half of FACCH1	4022 36F8 3E20 8BE9 08
(After scrambling)	582C 4082 B0A8 A608 A3
Second half of FACCH1	6822 3CFA BC00 A36B 22

Pattern 2

FACCH1

(Before coding)	0100 800B E70B E800 0000
(After interleaving)	48A6 4702 00C8 C000 21
	4080 FE53 3701 4723 03
(After scrambling)	CAA4 670A 8AE8 6AA2 A3
First half of FACCH1	48A2 74F9 3F29 CF0B 2B
(After scrambling)	48AC 4580 20E0 422A 8B
Second half of FACCH1	60A2 7EFB BD09 E789 01

Pattern 3
FACCH1

(Before coding)	0100 800F D00F CF00 0000
(After interleaving)	542A 0221 8108 C402 31
	4D0E 7C56 3401 0725 40
(After scrambling)	D628 2229 0B28 6EA0 B3
First half of FACCH1	452C F6FC 3C29 8F0D 68
(After scrambling)	5420 00A3 A120 4628 9B
Second half of FACCH1	6D2C FCFE BE09 A78F 42

[TX_REL]Inbound

Message Type :	00 1000b
CC Option :	00
Call Type :	100b (Individual Call)
Home Repeater	1
Source Unit ID :	1 / 3E7 / 7D0
Destination Unit ID :	2 / 3E8 / 7CF

Pattern 1
FACCH1

(Before coding)	0800 8008 0108 0200 0000
(After interleaving)	1886 0200 9088 2622 08
	C900 3C42 0E10 01C4 22
(After scrambling)	9A84 2208 1AA8 8C80 8A
First half of FACCH1	C122 B6E8 0638 89EC 0A
(After scrambling)	188C 0082 B0A0 A408 A2
Second half of FACCH1	E922 BCEA 8418 A16E 20

Pattern 2
FACCH1

(Before coding)	0800 800B E70B E800 0000
(After interleaving)	0806 0702 00C0 C200 20
	C180 7E43 0F19 4526 01
(After scrambling)	8A04 270A 8AE0 68A2 A2
First half of FACCH1	C9A2 F4E9 0731 CD0E 29
(After scrambling)	080C 0580 20E8 402A 8A
Second half of FACCH1	E1A2 FEEB 8511 E58C 03

Pattern 3
FACCH1

(Before coding)	0800 800F D00F CF00 0000
(After interleaving)	148A 4221 8100 C602 30
	CC0E FC46 0C19 0520 42
(After scrambling)	9688 6229 0B20 6CA0 B2
First half of FACCH1	C42C 76EC 0431 8D08 6A
(After scrambling)	1480 40A3 A128 4428 9A
Second half of FACCH1	EC2C 7CEE 8611 A58A 40

[TX_REL]Outbound

Message Type :	00 1000b
CC Option :	00
Call Type :	100b (Individual Call)
Free Repeater :	2
Home Repeater	1
Source Unit ID :	1 / 3E7 / 7D0
Destination Unit ID :	2 / 3E8 / 7CF

Pattern 1

FACCH1

(Before coding)	0800 8208 0108 0200 0000
(After interleaving)	1896 0204 9081 2220 49
	C800 7C22 0E00 01C4 20
(After scrambling)	9A94 220C 1AA1 8882 CB
First half of FACCH1	C022 F688 0628 89EC 08
(After scrambling)	189C 0086 B0A9 A00A E3
Second half of FACCH1	E822 FC8A 8408 A16E 22

Pattern 2

FACCH1

(Before coding)	0800 820B E70B E800 0000
(After interleaving)	0816 0706 00C9 C602 61
	C080 3E23 0F09 4526 03
(After scrambling)	8A14 270E 8AE9 6CA0 E3
First half of FACCH1	C8A2 B489 0721 CD0E 2B
(After scrambling)	081C 0584 20E1 4428 CB
Second half of FACCH1	E0A2 BE8B 8501 E58C 01

Pattern 3

FACCH1

(Before coding)	0800 820F D00F CF00 0000
(After interleaving)	149A 4225 8109 C200 71
	CD0E BC26 0C09 0520 40
(After scrambling)	9698 622D 0B29 68A2 F3
First half of FACCH1	C52C 368C 0421 8D08 68
(After scrambling)	1490 40A7 A121 402A DB
Second half of FACCH1	ED2C 3C8E 8601 A58A 42

5.1.5. UDCH2

A UDCH2 is a functional channel to transfer the user data on a traffic channel in a trunked radio system. A data string before being encoded is 160-bit data consisting of layer 3 information. The 288-bit UDCH2 is generated by encoding the data string. Messages to be used the UDCH2 test are DCALL (Header) and DCALL (User Data). ID configurations to be used in this test are three patterns as presented in Table 5.1-1.

[DCALL(Header)]

Message Type :	00 1001b
CC Option :	00
Call Type :	100b
Data Call Option :	00000b (4800bps)
Home Repeater :	1
Source Unit ID :	1 / 3E7 / 7D0
Destination Unit ID :	2 / 3E8 / 7CF
Cipher Type :	00b
Key ID :	000000b
Packet Information :	000400

Pattern 1

UDCH2

(Before coding)	0900 8008 0108 0200 0004 0000 0000 0000 0000 0000
(After interleaving)	5826 0220 A090 2620 0AC9 009C 722E 080F C520
	0080 0000 1008 0002 0101 0000 4000 1804 0201
(After scrambling)	DA24 2228 2AB0 8C82 88C1 2216 D826 2087 ED08
	008A 0282 3020 8228 AB21 2280 E88A 10A4 A803

Pattern 2

UDCH2

(Before coding)	0900 800B E70B E800 0004 0000 0000 0000 0000 0000
(After interleaving)	48A6 0722 30D8 C202 22C1 80DE 732F 014B 2703
	0080 0000 1008 0002 0101 0000 4000 1804 0201
(After scrambling)	CAA4 272A BAF8 68A0 A0C9 A254 D927 29C3 0F2B
	008A 0282 3020 8228 AB21 2280 E88A 10A4 A803

Pattern 3

UDCH2

(Before coding)	0900 800F D00F CF00 0004 0000 0000 0000 0000 0000
(After interleaving)	542A 4201 B118 C600 32CC 0E5C 762C 010B 2140
	0080 0000 1008 0002 0101 0000 4000 1804 0201
(After scrambling)	D628 6209 3B38 6CA2 B0C4 2CD6 DC24 2983 0968
	008A 0282 3020 8228 AB21 2280 E88A 10A4 A803

[DCALL(User Data)]

Message Type :	00 1011b
Packet Frame Number :	0
Block Number :	0
User Data Area :	414243444546474849 4A4B4C4D4E

UDCH2

(Before coding)	0B00 4142 4344 4546 4748 494A 4B4C 4D4E 7849 3A33
(After interleaving)	6B21 0F06 7041 965C 78BD 69DD E44A 2F7F 2EC1
	4FD0 6088 B2DE F939 8671 8710 588A AE54 C1EA
(After scrambling)	E923 2F0E FA61 3CFE FAB5 4B57 4E42 07F7 06E9
	4FDA 620A 92F6 7B13 2C51 A590 F000 A6F4 6BE8

5.1.6. FACCH3

An FACCH3 is a functional channel which transfers a control information temporarily stealing the user data on a traffic channel in a trunked radio system. A data string before being encoded is 160-bit data consisting of layer 3 information. The 288-bit FACCH3 is generated by encoding the data string. Messages to be used in the FACCH3 test are STAT_REQ and TX_REL. ID configurations to be used in this test are three patterns as presented in Table 5.1-1.

[STAT_REQ]

Message Type :	110010b
CC Option :	00
Call Type :	100b
Status Call Option :	00000b
Home Repeater :	1
Source Unit ID :	1 / 3E7 / 7D0
Destination Unit ID :	2 / 3E8 / 7CF
Status:	1

Pattern 1

FACCH3

(Before coding)	3200 8008 0108 0200 0100 0000 0000 0000 0000 0000
(After interleaving)	1906 8230 8880 2421 08C8 C07C A266 0409 CF25
	0080 0000 1008 0002 0101 0000 4000 1804 0201
(After scrambling)	9B04 A238 02A0 8E83 8AC0 E2F6 086E 2C81 E70D
	008A 0282 3020 8228 AB21 2280 E88A 10A4 A803

Pattern 2

FACCH3

(Before coding)	3200 800B E70B E800 0100 0000 0000 0000 0000 0000
(After interleaving)	0986 8732 18C8 C003 20C0 403E A367 0D4D 2D06
	0080 0000 1008 0002 0101 0000 4000 1804 0201
(After scrambling)	8B84 A73A 92E8 6AA1 A2C8 62B4 096F 25C5 052E
	008A 0282 3020 8228 AB21 2280 E88A 10A4 A803

Pattern 3

FACCH3

(Before coding)	3200 800F D00F CF00 0100 0000 0000 0000 0000 0000
(After interleaving)	150A C211 9908 C401 30CD CEBC A664 0D0D 2B45
	0080 0000 1008 0002 0101 0000 4000 1804 0201
(After scrambling)	9708 E219 1328 6EA3 B2C5 EC36 0C6C 2585 036D
	008A 0282 3020 8228 AB21 2280 E88A 10A4 A803

[TX_REL]Inbound

```
    Message Type :          00 1000b
    CC Option :             00
    Call Type :             100b
    Home Repeater           1
    Source Unit ID :        1 / 3E7 / 7D0
    Destination Group ID :  2 / 3E8 / 7CF
```

Pattern 1

FACCH3

```
    (Before coding)        0800 8008 0108 0200 0000 0000 0000 0000 0000 0000
    (After interleaving)   1886 0200 9088 2622 08C9 003C 420E 1001 C422
                           0080 0000 1008 0002 0101 0000 4000 1804 0201
    (After scrambling)     9A84 2208 1AA8 8C80 8AC1 22B6 E806 3889 EC0A
                           008A 0282 3020 8228 AB21 2280 E88A 10A4 A803
```

Pattern 2

FACCH3

```
    (Before coding)        0800 800B E70B E800 0000 0000 0000 0000 0000 0000
    (After interleaving)   0806 0702 00C0 C200 20C1 807E 430F 1945 2601
                           0080 0000 1008 0002 0101 0000 4000 1804 0201
    (After scrambling)     8A04 270A 8AE0 68A2 A2C9 A2F4 E907 31CD 0E29
                           008A 0282 3020 8228 AB21 2280 E88A 10A4 A803
```

Pattern 3

FACCH3

```
    (Before coding)        0800 800F D00F CF00 0000 0000 0000 0000 0000 0000
    (After interleaving)   148A 4221 8100 C602 30CC 0EFC 460C 1905 2042
                           0080 0000 1008 0002 0101 0000 4000 1804 0201
    (After scrambling)     9688 6229 0B20 6CA0 B2C4 2C76 EC04 318D 086A
                           008A 0282 3020 8228 AB21 2280 E88A 10A4 A803
```

[TX_REL]Outbound

```
    Message Type :          00 1000b
    CC Option :             00
    Call Type :             100b
    Free Repeater :         2
    Home Repeater :         1
    Source Unit ID :        1 / 3E7 / 7D0
    Destination Group ID :  2 / 3E8 / 7CF
```

Pattern 1

FACCH3

```
    (Before coding)        0800 8208 0108 0200 0000 0000 0000 0000 0000 0000
    (After interleaving)   1896 0204 9081 2220 49C8 007C 220E 0001 C420
                           0080 0000 1008 0002 0101 0000 4000 1804 0201
    (After scrambling)     9A94 220C 1AA1 8882 CBC0 22F6 8806 2889 EC08
                           008A 0282 3020 8228 AB21 2280 E88A 10A4 A803
```

Pattern 2

FACCH3

```
    (Before coding)        0800 820B E70B E800 0000 0000 0000 0000 0000 0000
    (After interleaving)   0816 0706 00C9 C602 61C0 803E 230F 0945 2603
                           0080 0000 1008 0002 0101 0000 4000 1804 0201
    (After scrambling)     8A14 270E 8AE9 6CA0 E3C8 A2B4 8907 21CD 0E2B
                           008A 0282 3020 8228 AB21 2280 E88A 10A4 A803
```

Pattern 3

FACCH3

```
    (Before coding)        0800 820F D00F CF00 0000 0000 0000 0000 0000 0000
    (After interleaving)   149A 4225 8109 C200 71CD 0EBC 260C 0905 2040
                           0080 0000 1008 0002 0101 0000 4000 1804 0201
    (After scrambling)     9698 622D 0B29 68A2 F3C5 2C36 8C04 218D 0868
                           008A 0282 3020 8228 AB21 2280 E88A 10A4 A803
```

5.2. Voice Format Tests for Receiver

This section verifies whether a receiver can properly decode functional channels which are used for voice communications on an RTCH2. A block diagram of the tests is presented in Figure 4.2-1.

5.2.1. SCCH Test

The SCCH is function channel to be used in both of an inbound and outbound directions, this test is applicable to both TR(inbound) and SU(outbound).

Inbound

Send out frames consisting of the following data from an interoperability tester.

- An inbound SCCH (INFO1, INFO2, INFO3 or INFO4) data string after scramble as presented in Section 5.1.2
- An inbound LICH data string for VCH/ SCCH testing after scramble as presented in Section 5.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- An inbound SCCH (INFO1, INFO2, INFO3 or INFO4) data string before being encoded as presented in Section 5.1.2
- An inbound LICH data string for VCH/ SCCH testing before dibit processing as presented in Section 5.1.1

Outbound

Send out frames consisting of the following data from an interoperability tester.

- An outbound SCCH (INFO1, INFO2, INFO3 or INFO4) data string after scramble as presented in Section 5.1.2
- An out bound LICH data string for VCH/ SCCH testing after scramble as presented in Section 5.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- An outbound SCCH (INFO1, INFO2, INFO3 or INFO4) data string before being encoded as presented in Section 5.1.2
- An outbound LICH data string for VCH/ SCCH testing before dibit processing as presented in Section 5.1.1

5.2.2. VCH Test

The VCH is channel to be used in both of an inbound and outbound direction, sent with the super frame structure of 4 frames unit. This test is applicable to both TR(inbound) and SU(outbound).

Inbound

Send out frames consisting of the following data from an interoperability tester.

- A VCH data string after scramble as presented in Section 5.1.3
- An LICH data string for VCH/ SCCH testing after scramble as presented in Section 5.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- A VCH data string after voice coding processing as presented in Section 5.1.3
- An LICH data string for VCH/ SACCH testing before dibit processing as presented in Section 5.1.1

Outbound

Send out frames consisting of the following data from an interoperability tester.

- A VCH data string after scramble as presented in Section 5.1.3
- An LICH data string for VCH/ SCCH testing after scramble as presented in Section 5.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- A VCH data string after voice coding processing as presented in Section 5.1.3
- An LICH data string for VCH/ SCCH testing before dibit processing as presented in Section 5.1.1

5.2.3. FACCH1 Test

Since an FACCH1 is a channel to be used in both of an inbound and outbound direction, this test is applicable to both TR(inbound) and SU(outbound). There are two tests, one is the test using a VCALL message and the other is the test using a TX_REL message. The first and second FACCH1s are intended to be tested since all VCHs are replaced by two FACCH1s.

Inbound

[VCALL]

Send out frames consisting of the following data from an interoperability tester.

- An FACCH1(VCALL) data string after scramble as presented in Section 5.1.4
- An inbound LICH data string for FACCH1/ SCCH testing after scramble as presented in Section 5.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- An FACCH1(VCALL) data string before being encoded as presented in Section 5.1.4
- An inbound LICH data string for FACCH1/ SCCH testing before dibit processing as presented in Section 5.1.1

[TX_REL](Inbound)

Send out frames consisting of the following data from an interoperability tester.

- An inbound FACCH1(TX_REL) data string after scramble as presented in Section 5.1.4
- An inbound LICH data string for FACCH1/ SCCH testing after scramble as presented in Section 5.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- An inbound FACCH1(TX_REL) data string before being encoded as presented in Section 5.1.4
- An inbound LICH data string for FACCH1/ SCCH testing before dibit processing as presented in Section 5.1.1

Outbound

[VCALL]

Send out frames consisting of the following data from an interoperability tester.

- An FACCH1(VCALL) data string after scramble as presented in Section 5.1.4
- An outbound LICH data string for FACCH1/ SCCH testing after scramble as presented in Section 5.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- An FACCH1(VCALL) data string before being encoded as presented in Section 5.1.4
- An outbound LICH data string for FACCH1/ SCCH testing before dibit processing as presented in Section 5.1.1

[TX_REL](Outbound)

Send out frames consisting of the following data from an interoperability tester.

- An outbound FACCH1(TX_REL) data string after scramble as presented in Section 5.1.4
- An outbound LICH data string for FACCH1/ SCCH testing after scramble as presented in Section 5.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- An outbound FACCH1(TX_REL) data string before being encoded as presented in Section 5.1.4
- An outbound LICH data string for FACCH1/ SCCH testing before dibit processing as presented in Section 5.1.1

5.3. Data Format Tests for Receiver

This section verifies whether a receiver can properly decode functional channels which are used for data communications on an RTCH2. A block diagram of the tests is presented in Figure 4.2-1.

5.3.1. UDCH2 Test

Since a UDCH2 is a channel to be used in both of an inbound and outbound direction, this test is applicable to both TR(inbound) and SU(outbound).

Inbound

Send out frames consisting of the following data from an interoperability tester.

- A UDCH2 data string after scramble as presented in Section 5.1.5
- An inbound LICH data string for UDCH2 testing after scramble as presented in Section 5.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- A UDCH2 data string before being encoded as presented in Section 5.1.5
- An inbound LICH data string for UDCH2 testing before dibit processing as presented in Section 5.1.1

Outbound

Send out frames consisting of the following data from an interoperability tester.

- A UDCH2 data string after scramble as presented in Section 5.1.5
- An outbound LICH data string for UDCH2 testing after scramble as presented in Section 5.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- A UDCH2 data string before being encoded as presented in Section 5.1.5
- An outbound LICH data string for UDCH2 testing before dibit processing as presented in Section 5.1.1

5.3.2. FACCH3 Test

Since an FACCH3 is a channel to be used in both of an inbound and outbound direction, this test is applicable to both TR(inbound) and SU(outbound).

Inbound

[STAT_REQ]

Send out frames consisting of the following data from an interoperability tester.

- An FACCH3 data string after scramble as presented in Section 5.1.6
- An inbound LICH data string for FACCH3 testing after scramble as presented in Section 5.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- An FACCH3 data string before being encoded as presented in Section 5.1.6
- An inbound LICH data string for FACCH3 testing before dibit processing as presented in Section 5.1.1

[TX_REL] Inbound

Send out frames consisting of the following data from an interoperability tester.

- An FACCH3 data string after scramble as presented in Section 5.1.6
- An inbound LICH data string for FACCH3 testing after scramble as presented in Section 5.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- An FACCH3 data string before being encoded as presented in Section 5.1.6
- An inbound LICH data string for FACCH3 testing before dibit processing as presented in Section 5.1.1

Outbound

[STAT_REQ]

Send out frames consisting of the following data from an interoperability tester.

- An FACCH3 data string after scramble as presented in Section 5.1.6
- An outbound LICH data string for FACCH3 testing after scramble as presented in Section 5.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- An FACCH3 data string before being encoded as presented in Section 5.1.6
- An outbound LICH data string for FACCH3 testing before dibit processing as presented in Section 5.1.1

[TX_REL] Outbound

Send out frames consisting of the following data from an interoperability tester.

- An FACCH3 data string after scramble as presented in Section 5.1.6
- An outbound LICH data string for FACCH3 testing after scramble as presented in Section 5.1.1

Verify that data strings which are received and decoded by a unit under test are identical with the following data strings:

- An FACCH3 data string before being encoded as presented in Section 5.1.6
- An outbound LICH data string for FACCH3 testing before dibit processing as presented in Section 5.1.1

5.4. Voice Format Tests for Transmitter

This section verifies whether a transmitter can properly encode functional channels which are used for voice communications on an RTCH2. A block diagram of the tests is presented in Figure 4.5-1.

5.4.1. SCCH Test

The SCCH is function channel to be used in both of an inbound and outbound directions, this test is applicable to both TR(outbound) and SU(inbound).

Inbound

Send out frames consisting of the following data from a unit under test.

- An inbound SCCH (INFO1, INFO2, INFO3 or INFO4) data string before being encoded as presented in Section 5.1.2
- An inbound LICH data string for VCH/ SCCH testing before dibit processing as presented in Section 5.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- An inbound SCCH (INFO1, INFO2, INFO3 or INFO4) data string after scramble as presented in Section 5.1.2.
- An inbound LICH data string for VCH/ SCCH testing after scramble as presented in Section 5.1.1

Outbound

Send out frames consisting of the following data from a unit under test.

- An outbound SCCH (INFO1, INFO2, INFO3 or INFO4) data string before being encoded as presented in Section 5.1.2
- An outbound LICH data string for VCH/ SCCH testing before dibit processing as presented in Section 5.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- An outbound SCCH (INFO1, INFO2, INFO3 or INFO4) data string after scramble as presented in Section 5.1.2.
- An outbound LICH data string for VCH/ SCCH testing after scramble as presented in Section 5.1.1

5.4.2. VCH Test

Since a VCH is a channel to be used in both of an inbound and outbound direction, this test is applicable to both TR(outbound) and SU(inbound).

Inbound

Send out frames consisting of the following data from a unit under test.

- A VCH data string after voice coding processing as presented in Section 5.1.3
- An inbound LICH data string for VCH/ SCCH testing before dibit processing as presented in Section 5.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- A VCH data string after scramble as presented in Section 5.1.3
- An inbound LICH data string for VCH/ SACCH testing after scramble as presented in Section 5.1.1

Outbound

Send out frames consisting of the following data from a unit under test.

- A VCH data string after voice coding processing as presented in Section 5.1.3
- An outbound LICH data string for VCH/ SCCH testing before dibit processing as presented in Section 5.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- A VCH data string after scramble as presented in Section 5.1.3
- An outbound LICH data string for VCH/ SACCH testing after scramble as presented in Section 5.1.1

5.4.3. FACCH1 Test

Since an FACCH1 is a channel to be used in both of an inbound and outbound direction, this test is applicable to both TR(outbound) and SU(inbound). There are two tests, one is the test using a VCALL message and the other is the test using a TX_REL message. The first and second FACCH1s are intended to be tested since all VCHs are replaced by two FACCH1s.

Inbound

[VCALL]

Send out frames consisting of the following data from a unit under test.

- An FACCH1(VCALL) data string before being encoded as presented in Section 5.1.4
- An inbound LICH data string for FACCH1/ SCCH testing before dibit processing as presented in Section 5.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- An FACCH1(VCALL) data string after scramble as presented in Section 5.1.4
- An inbound LICH data string for FACCH1/ SCCH testing after scramble as presented in Section 5.1.1

[TX_REL](Inbound)

Send out frames consisting of the following data from a unit under test.

- An inbound FACCH1(TX_REL) data string before being encoded as presented in Section 5.1.4
- An inbound LICH data string for FACCH1/ SCCH testing before dibit processing as presented in Section 5.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- An inbound FACCH1(TX_REL) data string after scramble as presented in Section 5.1.4
- An inbound LICH data string for FACCH1/ SCCH testing after scramble as presented in Section 5.1.1

Outbound

[VCALL]

Send out frames consisting of the following data from a unit under test.

- An FACCH1(VCALL) data string before being encoded as presented in Section 5.1.4
- An outbound LICH data string for FACCH1/ SCCH testing before dibit processing as presented in Section 5.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- An FACCH1(VCALL) data string after scramble as presented in Section 5.1.4
- An outbound LICH data string for FACCH1/ SCCH testing after scramble as presented in Section 5.1.1

[TX_REL](Outbound)

Send out frames consisting of the following data from a unit under test.

- An outbound FACCH1(TX_REL) data string before being encoded as presented in Section 5.1.4
- An outbound LICH data string for FACCH1/ SCCH testing before dibit processing as presented in Section 5.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- An outbound FACCH1(TX_REL) data string after scramble as presented in Section 5.1.4
- An outbound LICH data string for FACCH1/ SCCH testing after scramble as presented in Section 5.1.1

5.5. Data Format Tests for Transmitter

This section verifies whether a transmitter can properly encode functional channels which are used data communications on an RTCH2. Figure 4.5-1 shows a block diagram of the tests.

5.5.1. UDCH2 Test

Since a UDCH2 is a channel to be used in both of an inbound and outbound direction, this test is applicable to both TR(outbound) and SU(inbound).

Inbound

Send out frames consisting of the following data from a unit under test.

- A UDCH2 data string before being encoded as presented in Section 5.1.5
- An inbound LICH data string for UDCH2 testing before dibit processing as presented in Section 5.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- A UDCH2 data string after scramble as presented in Section 5.1.5
- An inbound LICH data string for UDCH2 testing after scramble as presented in Section 5.1.1

Outbound

Send out frames consisting of the following data from a unit under test.

- A UDCH2 data string before being encoded as presented in Section 5.1.5
- An outbound LICH data string for UDCH2 testing before dibit processing as presented in Section 5.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- A UDCH2 data string after scramble as presented in Section 5.1.5
- An outbound LICH data string for UDCH2 testing after scramble as presented in Section 5.1.1

5.5.2. FACCH3 Test

Since an FACCH3 is a channel to be used in both of an inbound and outbound direction, this test is applicable to both TR(outbound) and SU(inbound).

Inbound

[STAT_REQ]

Send out frames consisting of the following data from a unit under test.

- An FACCH3 data string before being encoded as presented in Section 5.1.6
- An inbound LICH data string for FACCH3 testing before dibit processing as presented in Section 5.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- An FACCH3 data string after scramble as presented in Section 5.1.6
- An inbound LICH data string for FACCH3 testing after scramble as presented in Section 5.1.1

[TX_REL]Inbound

Send out frames consisting of the following data from a unit under test.

- An FACCH3 data string before being encoded as presented in Section 5.1.6
- An inbound LICH data string for FACCH3 testing before dibit processing as presented in Section 5.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- An FACCH3 data string after scramble as presented in Section 5.1.6
- An inbound LICH data string for FACCH3 testing after scramble as presented in Section 5.1.1

Outbound

[STAT_REQ]

Send out frames consisting of the following data from a unit under test.

- An FACCH3 data string before being encoded as presented in Section 5.1.6
- An outbound LICH data string for FACCH3 testing before dibit processing as presented in Section 5.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- An FACCH3 data string after scramble as presented in Section 5.1.6
- An outbound LICH data string for FACCH3 testing after scramble as presented in Section 5.1.1

[TX_REL]Outbound

Send out frames consisting of the following data from a unit under test.

- An FACCH3 data string before being encoded as presented in Section 5.1.6
- An outbound LICH data string for FACCH3 testing before dibit processing as presented in Section 5.1.1

Verify that data strings which are received by an interoperability tester are identical with the following data strings:

- An FACCH3 data string after scramble as presented in Section 5.1.6
- An outbound LICH data string for FACCH3 testing after scramble as presented in Section 5.1.1

5.6. Preamble and FSW Test for Transmitter

This section verifies whether a transmitter can properly transmit a preamble and FSW in an RTCH2. Figure 4.5-1 shows a block diagram of the tests. This test is applicable to both TR and SU.

When a unit under test starts transmitting, verify that symbol strings received by an interoperability tester are identical with the preamble pattern and FSW pattern specified in REF [1].

5.7. Post Field Test for Transmitter

This section verifies whether a transmitter can properly transmit a post field in the outbound RTCH2. Figure 4.5-1 shows a block diagram of the tests. This test is applicable to a TR.

When a unit under test starts transmitting, verify that a symbol string of the post field received by an interoperability tester is identical with the post field pattern specified in REF [1].

6. Appendix 1 (Conventional, Type-C Trunked System)

The test procedure for each functional channel is described in Section 4. This section explains the test procedure using frame data which consists of different kinds of functional channels without testing each functional channel test individually. An examiner can perform the test procedure as described in Section 4 or in this section

6.1. Test Items

Table 6.1-1 shows the comparison table for test items between Section 4 and Section 5. An examiner performs required tests according to operational modes implemented in a unit under test.

Test Items in Section 5			Test Items in Section 4
6.6.1.1	RX	Outbound Control Channel Test	4.2
6.6.1.2		Inbound Control Channel Test	4.2
6.6.1.3		Voice Format Test	4.3
6.6.1.4		Data Format Test	4.4
6.6.2.1	TX	Outbound Control Channel Test	4.5
6.6.2.3		Inbound Control Channel Test	4.5
6.6.2.3		Voice Format Test	4.6
6.6.2.4		Data Format Test	4.7
6.6.2.5		Preamble and FSW Tests	4.8
6.6.2.6		Post Field Test	4.9

Table 6.1-1 Comparison Table for Test Items between Section 4 and Section 5

6.2. Measuring Instruments

Measuring instruments required for tests are shown in Table 6.2-1.

Instruments	Definition / Remark
SSG	Refer to Section 6.8 for definitions
Oscilloscope	It is used to display a waveform of an FM demodulated output signal.
Monitor Receiver	● Support the demodulation processing of the NXDN modulated signal ● Support the output processing of the demodulated bit sequence (Refer to Section 6.9 for conditions of Monitor Receiver)
Monitor PC	Device which displays the demodulated bit sequence that Monitor Receiver outputs
Directional Coupler	Use appropriate device by an examiner
Attenuator	Use appropriate device by an examiner

Table 6.2-1 Definitions of Measuring Instruments

6.3. SSG Data for Test

In the receiver test of SU, the behavior of a unit under test is verified by storing a test data to a memory of SSG and sending a NXDN signal from SSG.

The lists of SSG data used for tests are shown in Section 6.7. An examiner should create the required SSG data files and write them in SSG in advance.

6.4. Data Notation

Unless otherwise noted, a data sequence is expressed by a hexadecimal number.

6.5. Test Setup

6.5.1. Test Connection Diagram of Receiver Test

This section explains the test connection diagram used for receiver tests.

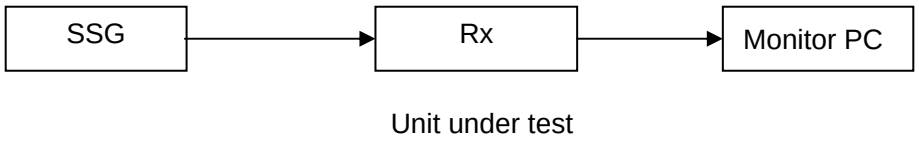


Figure 6.5-1 Connection Diagram of Receiver Test

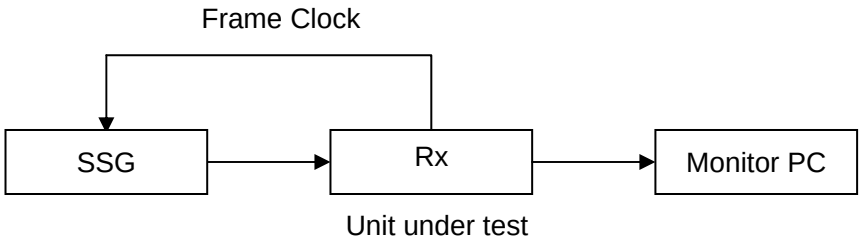


Figure 6.5-2 Connection Diagram of Receiver Test for Inbound Control Channel

6.5.2. Test Connection Diagram of Transmitter Test

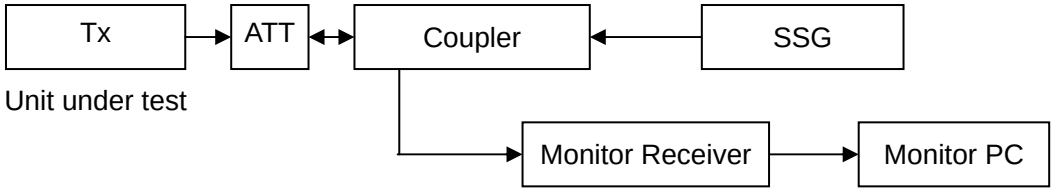
This section explains the test connection diagram used for transmitter tests.

1. Voice Format, Data Format and Outbound Control Channel Test

Unit under test

Figure 6.5-3 Connection Diagram (1) of Transmitter Test

2. Inbound Control Channel Test



Unit under test

Figure 6.5-4 Connection Diagram (2) of Transmitter Test

3. Preamble, FSW and Post Field Test

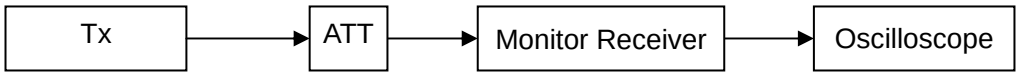


Figure 6.5-5 Connection Diagram (3) of Transmitter Test

6.6. Test Method

6.6.1. Receiver Test

This section tests the validity of decode processing of a unit under test. A test is conducted by sending the frame data from SSG and comparing the data decoded by a unit under test with the SSG data before coding.

A unit under test shall have a function which sends to Monitor PC the data sequence after decoding the received frame.

6.6.1.1. Outbound Control Channel Test

This section introduces the testing procedure regarding the decode processing of functional channels (LICH, CAC, and Collision Control Field) on outbound control channel.

- Connection Diagram

See Figure 6.5-1.

- Test Procedure

Use appropriate trunking setting for a unit under test to receive the signal of outbound control channel.

(1) Select appropriate SSG data file used for the test.

(2) Transmit the signal selected at the step (1) from an SSG, and monitor the data sequence decoded by the unit under test with Monitor PC.
Compare the monitored data sequence with the data sequence of LICH/CAC/ Collision control field before coding shown in Table 6.6-2 to Table 6.6-3 corresponding to the SSG data selected at the step (1).
If the two data sequences are the same, the unit under test passes this test.
The relation between the SSG data and the data sequence before coding as a comparison object is shown in Table 6.6-1.

(3) Perform tests of the step (1) to the step (2) using all of the SSG data as shown in Table 6.6-1.

Table 6.6-2 to Table 6.6-3 show the data which combines LICH data sequence before dibit, CAC data sequence before coding and Collision Control Field before dibit as described in Section 4.1,

Format	SSG Data	Data sequence before coding
Single Message Format	Table 6.7-1	Table 6.6-2
Dual Message Format	Table 6.7-2	Table 6.6-3

Table 6.6-1 Relation between SSG data and Data Sequence before coding

No	LICH	CAC	Collision Control Field
1	02	81184000214914BFC00000000000000000100400	A0000 (IDLE) B45B1 (4800 bps Received) B0FFA(9600 bps/EHR Received) BAC2F(9600 bps/EFR Received)

Table 6.6-2 Data Sequence Before Coding (Single Message Format)

No	LICH	CAC	Collision Control Field
1	02	4119400021BFC00000001B4000220402000000	A0000 (IDLE) B45B1 (4800 bps Received) B0FFA (9600 bps/EHR Received) BAC2F (9600 bps/EFR Received)

Table 6.6-3 Data Sequence Before Coding (Dual Message Format)

6.6.1.2. Inbound Control Channel Test

This section explains the testing procedure regarding the decode processing of functional channels (LICH, L-CAC, and S-CAC) on inbound control channel.

- Connection Diagram

See Figure 6.5-2.

Figure 6.5-2 shows the method of transmitting an inbound control channel signal from SSG in synchronized state with the frame timing of a unit under test, by using the Frame Clock signal from a unit under test as a trigger. If the test can be performed properly, another method is available.

6.6.1.2.1. Long CAC Tests

- Test Procedure

The unit under test should be suitably set its trunking configuration so that it can receive the signal of inbound control channel.

(1) Select appropriate SSG data file used for the test.

(2) Transmit the signal selected at the step (1) from an SSG, and monitor the data sequence decoded by the unit under test with Monitor PC.
Compare the monitored data sequence with the data sequence of LICH/Long CAC before coding shown in Table 6.6-5 to Table 6.6-10 corresponding to the SSG data selected at the step (1).
If the two data sequences are the same, the unit under test passes this test.
The relation between the SSG data and the data sequence before coding as a comparison object is shown in Table 6.6-4.

(3) Perform tests of the step (1) to the step (2) using all of the SSG data as shown in Table 6.6-4.

Table 6.6-5 to Table 6.6-10 show the data which combines LICH data sequence before dibit and Long CAC data sequence before coding as described in Section 4.1,

Mode & Pattern	SSG data	Data sequence before coding
4800 bps, Pattern 1	Table 6.7-4	Table 6.6-5
4800 bps, Pattern 2		Table 6.6-6
4800 bps, Pattern 3		Table 6.6-7
9600 bps, Pattern 1	Table 6.7-5	Table 6.6-8
9600 bps, Pattern 2		Table 6.6-9
9600 bps, Pattern 3		Table 6.6-10

Table 6.6-4 Relation between SSG data and Data Sequence before coding (SDCALL_REQ)

No	LICH	Long CAC
1	11	013800800001000200800400000000000000
2	11	013900414243444546474849 4A77FD053E

Table 6.6-5 Data Sequence before coding (4800 bps, Pattern 1)

No	LICH	Long CAC
1	11	013800807FF87FF9008004000000000000
2	11	013900414243444546474849 4A77FD053E

Table 6.6-6 Data Sequence before coding (4800 bps, Pattern 2)

No	LICH	Long CAC
1	11	01380080FFEFFFEE008004000000000000
2	11	013900414243444546474849 4A77FD053E

Table 6.6-7 Data Sequence before coding (4800 bps, Pattern 3)

No	LICH	Long CAC
1	11	013800820001000200800400000000000000
2	11	013900414243444546474849 4A77FD053E

Table 6.6-8 Data Sequence before coding (9600 bps, Pattern 1)

No	LICH	Long CAC
1	11	013800827FF87FF9008004000000000000
2	11	013900414243444546474849 4A77FD053E

Table 6.6-9 Data Sequence before coding (9600 bps, Pattern 2)

No	LICH	Long CAC
1	11	01380082FFEFFFEE008004000000000000
2	11	013900414243444546474849 4A77FD053E

Table 6.6-10 Data Sequence before coding (9600 bps, Pattern 3)

6.6.1.2.2. Short CAC Tests

- Test Procedure

The unit under test should be suitably set its trunking configuration so that it can receive the signal of inbound control channel.

(1) Select appropriate SSG data file used for the test.
(2) Transmit the signal selected at the step (1) from an SSG, and monitor the data sequence decoded by the unit under test with Monitor PC.
(3) For the frame test with VCALL_REQ message, compare the monitored data sequence with the data sequence of LICH/Short CAC before coding shown in Table 6.6-13 to Table 6.6-21 corresponding to the SSG data selected at the step (1).
If the two data sequences are the same, the unit under test passes this test.
The relation between the SSG data and the data sequence before coding as a comparison object is shown in Table 6.6-11.
(4) Perform tests of the step (1) to the step (3) using all of the SSG data with VCALL_REQ message as shown in Table 6.6-11.
(5) Perform tests of the step (1) to the step (3) using all of the SSG data with DCALL_REQ message as shown in Table 6.6-12.

Table 6.6-13 to Table 6.6-27 show the data which combines LICH data sequence before dibit and Short CAC data sequence before coding as described in Section 4.1,

Mode & Pattern	SSG data	Data sequence before coding
4800 bps, Pattern 4	Table 6.7-6	Table 6.6-13
4800 bps, Pattern 5		Table 6.6-14
4800 bps, Pattern 6		Table 6.6-15
9600 bps/EHR, Pattern 4	Table 6.7-7	Table 6.6-16
9600 bps/EHR, Pattern 5		Table 6.6-17
9600 bps/EHR, Pattern 6		Table 6.6-18
9600 bps/EFR, Pattern 4	Table 6.7-8	Table 6.6-19
9600 bps/EFR, Pattern 5		Table 6.6-20
9600 bps/EFR, Pattern 6		Table 6.6-21

Table 6.6-11 Relation between SSG data and Data Sequence before coding (VCALL_REQ)

Mode & Pattern	SSG data	Data sequence before coding
4800 bps, Pattern 1	Table 6.7-9	Table 6.6-22
4800 bps, Pattern 2		Table 6.6-23
4800 bps, Pattern 3		Table 6.6-24
9600 bps, Pattern 1	Table 6.7-10	Table 6.6-25
9600 bps, Pattern 2		Table 6.6-26
9600 bps, Pattern 3		Table 6.6-27

Table 6.6-12 Relation between SSG data and Data Sequence before coding (DCALL_REQ)

No	LICH	Short CAC
1	30	0101002000010001000000000

Table 6.6-13 Data Sequence before coding (VCALL_REQ, 4800 bps, Pattern 4)

No	LICH	Short CAC
1	30	010100207FF87FF80000000000

Table 6.6-14 Data Sequence before coding (VCALL_REQ, 4800 bps, Pattern 5)

No	LICH	Short CAC
1	30	01010020FFEFFFEF0000000000

Table 6.6-15 Data Sequence before coding (VCALL_REQ, 4800 bps, Pattern 6)

No	LICH	Short CAC
1	30	0101002200010001000000000

Table 6.6-16 Data Sequence before coding (VCALL_REQ, 9600 bps/EHR, Pattern 4)

No	LICH	Short CAC
1	30	010100227FF87FF80000000000

Table 6.6-17 Data Sequence before coding (VCALL_REQ, 9600 bps/EHR, Pattern 5)

No	LICH	Short CAC
1	30	01010022FFEFFFEF0000000000

Table 6.6-18 Data Sequence before coding (VCALL_REQ, 9600 bps/EHR, Pattern 6)

No	LICH	Short CAC
1	30	0101002300010001000000000

Table 6.6-19 Data Sequence before coding (VCALL_REQ, 9600 bps/EFR, Pattern 4)

No	LICH	Short CAC
1	30	010100237FF87FF80000000000

Table 6.6-20 Data Sequence before coding (VCALL_REQ, 9600 bps/EFR, Pattern 5)

No	LICH	Short CAC
1	30	01010023FFEFFFEF0000000000

Table 6.6-21 Data Sequence before coding (VCALL_REQ, 9600 bps/EFR, Pattern 6)

No	LICH	Short CAC
1	30	0109008000010002000000000

Table 6.6-22 Data Sequence before coding (DCALL_REQ, 4800 bps, Pattern 1)

No	LICH	Short CAC
1	30	010900807FF87FF90000000000

Table 6.6-23 Data Sequence before coding (DCALL_REQ, 4800 bps, Pattern 2)

No	LICH	Short CAC
1	30	01090080FFEFFFEE0000000000

Table 6.6-24 Data Sequence before coding (DCALL_REQ, 4800 bps, Pattern 3)

No	LICH	Short CAC
1	30	0109008200010002000000000

Table 6.6-25 Data Sequence before coding (DCALL_REQ, 9600 bps, Pattern 1)

No	LICH	Short CAC
1	30	010900827FF87FF90000000000

Table 6.6-26 Data Sequence before coding (DCALL_REQ, 9600 bps, Pattern 2)

No	LICH	Short CAC
1	30	01090082FFEFFFEE0000000000

Table 6.6-27 Data Sequence before coding (DCALL_REQ, 9600 bps, Pattern 3)

6.6.1.3. Voice Format

This section explains the testing procedure regarding the decode processing of functional channels (LICH, SACCH, VCH, and FACCH1) used for voice communications on traffic channel.

- Connection Diagram

See Figure 6.5-1.

- Test Procedure

(1) Select appropriate SSG data used for the test.

(2) Transmit the signal selected at the step (1) from an SSG, and monitor the data sequence decoded by the unit under test with Monitor PC.
Compare the monitored data sequence with the data sequence of LICH/SACCH/FACCH1/VCH before coding as shown in Table 6.6-29 to Table 6.6-37 corresponding to the SSG data selected at the step (1).
If the two data sequences are the same, the unit under test passes this test.
The relation between the SSG data and the data sequence before coding as a comparison object is shown in Table 6.6-28.

(3) Perform tests of the step (1) to the step (2) using all of the SSG data as shown in Table 6.6-28.

Table 6.6-29 to Table 6.6-37 show the data which combines LICH data before dibit , SACCH / FACCH1 data before coding and VCH data after voice coding as described in Section 4.1,

Mode & Pattern	SSG data	Data sequence before coding
4800 bps, Pattern 7	Table 6.7-11	Table 6.6-29
4800 bps, Pattern 8		Table 6.6-30
4800 bps, Pattern 9		Table 6.6-31
9600 bps/EHR, Pattern 7	Table 6.7-12	Table 6.6-32
9600 bps/EHR, Pattern 8		Table 6.6-33
9600 bps/EHR, Pattern 9		Table 6.6-34
9600 bps/EFR, Pattern 7	Table 6.7-13	Table 6.6-35
9600 bps/EFR, Pattern 8		Table 6.6-36
9600 bps/EFR, Pattern 9		Table 6.6-37

Table 6.6-28 Relation between SSG data and Data Sequence before coding

No	LICH	SACCH	(FA1) FACCH1 (VCH) VCH VCH	(FA1) FACCH1 (VCH) VCH VCH
Direction: Inbound				
1	81	0110000	(FA1) 01002000010001000000	(FA1) 01002000010001000000
2	AC	C101000	(VCH) FEE2121212100 FEE2121212100	(VCH) FEE2121212100 FEE2121212100
3	AC	8180000	↑	↑
4	AC	4110000	↑	↑
5	AC	0140000	↑	↑
6	81	0110000	(FA1) 08002000010001000000	(FA1) 08002000010001000000
Direction: Outbound				
1	83	Same as the inbound data sequence		
2	AE			
3	AE			
4	AE			
5	AE			
6	83			

Table 6.6-29 Data Sequence before coding (4800 bps, Pattern 7)

No	LICH	SACCH	(FA1) FACCH1 (VCH) VCH VCH	(FA1) FACCH1 (VCH) VCH VCH
Direction: Inbound				
1	81	2010000	(FA1) 0100207FF87FF8000000	(FA1) 0100207FF87FF8000000
2	AC	E001000	(VCH) FEE2121212100 FEE2121212100	(VCH) FEE2121212100 FEE2121212100
3	AC	A081FFC	↑	↑
4	AC	6087FF8	↑	↑
5	AC	2000000	↑	↑
6	81	2010000	(FA1) 0800207FF87FF8000000	(FA1) 0800207FF87FF8000000
Direction: Outbound				
1	83	Same as the inbound data sequence		
2	AE			
3	AE			
4	AE			
5	AE			
6	83			

Table 6.6-30 Data Sequence before coding (4800 bps, Pattern 8)

No	LICH	SACCH	(FA1) FACCH1 (VCH) VCH VCH	(FA1) FACCH1 (VCH) VCH VCH
Direction: Inbound				
1	81	3F10000	(FA1) 010020FFEFFFEF000000	(FA1) 010020FFEFFFEF000000
2	AC	FF01000	(VCH) FEE2121212100 FEE2121212100	(VCH) FEE2121212100 FEE2121212100
3	AC	BF83FF8	↑	↑
4	AC	7FFFFEC	↑	↑
5	AC	3FC0000	↑	↑
6	81	3F10000	(FA1) 080020FFEFFFEF000000	(FA1) 080020FFEFFFEF000000
Direction: Outbound				
1	83	Same as the inbound data sequence		
2	AE			
3	AE			
4	AE			
5	AE			
6	83			

Table 6.6-31 Data Sequence before coding (4800 bps, Pattern 9)

No	LICH	SACCH	(FA1) FACCH1 (VCH) VCH VCH	(FA1) FACCH1 (VCH) VCH VCH
Direction: Inbound				
1	81	0110000	(FA1) 01002200010001000000	(FA1) 01002200010001000000
2	AC	C101000	(VCH) FEE2121212100 FEE2121212100	(VCH) FEE2121212100 FEE2121212100
3	A0	8188000	(FA1) 01002200010001000000	(FA1) 01002200010001000000
4	AC	4110000	(VCH) FEE2121212100 FEE2121212100	(VCH) FEE2121212100 FEE2121212100
5	A0	0140000	(FA1) 01002200010001000000	(FA1) 01002200010001000000
6	81	0110000	(FA1) 08002000010001000000	(FA1) 08002000010001000000
Direction: Outbound				
1	83	Same as the inbound data sequence		
2	AE			
3	A2			
4	AE			
5	A2			
6	83			

Table 6.6-32 Data Sequence before coding (9600 bps/EHR, Pattern 7)

No	LICH	SACCH	(FA1) FACCH1 (VCH) VCH VCH	(FA1) FACCH1 (VCH) VCH VCH
Direction: Inbound				
1	81	2010000	(FA1) 0100227FF87FF8000000	(FA1) 0100227FF87FF8000000
2	AC	E001000	(VCH) FEE2121212100 FEE2121212100	(VCH) FEE2121212100 FEE2121212100
3	A0	A089FFC	(FA1) 0100227FF87FF8000000	(FA1) 0100227FF87FF8000000
4	AC	6087FF8	(VCH) FEE2121212100 FEE2121212100	(VCH) FEE2121212100 FEE2121212100
5	A0	2000000	(FA1) 0100227FF87FF8000000	(FA1) 0100227FF87FF8000000
6	81	2010000	(FA1) 0800207FF87FF8000000	(FA1) 0800207FF87FF8000000
Direction: Outbound				
1	83	Same as the inbound data sequence		
2	AE			
3	A2			
4	AE			
5	A2			
6	83			

Table 6.6-33 Data Sequence before coding (9600 bps/EHR, Pattern 8)

No	LICH	SACCH	(FA1) FACCH1 (VCH) VCH VCH	(FA1) FACCH1 (VCH) VCH VCH
Direction: Inbound				
1	81	3F10000	(FA1) 010022FFEFFFEF000000	(FA1) 010022FFEFFFEF000000
2	AC	FF01000	(VCH) FEE2121212100 FEE2121212100	(VCH) FEE2121212100 FEE2121212100
3	A0	BF8BFF8	(FA1) 010022FFEFFFEF000000	(FA1) 010022FFEFFFEF000000
4	AC	7FFFFEC	(VCH) FEE2121212100 FEE2121212100	(VCH) FEE2121212100 FEE2121212100
5	A0	3FC0000	(FA1) 010022FFEFFFEF000000	(FA1) 010022FFEFFFEF000000
6	81	3F10000	(FA1) 080020FFEFFFEF000000	(FA1) 080020FFEFFFEF000000
Direction: Outbound				
1	83	Same as the inbound data sequence		
2	AE			
3	A2			
4	AE			
5	A2			
6	83			

Table 6.6-34 Data Sequence before coding (9600 bps/EHR, Pattern 9)

No	LICH	SACCH	(FA1) FACCH1 (VCH) VCH	(FA1) FACCH1 (VCH) VCH
Direction: Inbound				
1	81	0110000	(FA1) 01002300010001000000	(FA1) 01002300010001000000
2	AC	C101000	(VCH) 09B0880CC621F680A82600	(VCH) 09B0880CC621F680A82600
3	AC	818C000	↑	↑
4	AC	4110000	↑	↑
5	AC	0140000	↑	↑
6	81	0110000	(FA1) 08002000010001000000	(FA1) 08002000010001000000
Direction: Outbound				
1	83	Same as the inbound data sequence		
2	AE			
3	AE			
4	AE			
5	AE			
6	83			

Table 6.6-35 Data Sequence before coding (9600 bps/EFR, Pattern 7)

No	LICH	SACCH	(FA1) FACCH1 (VCH) VCH	(FA1) FACCH1 (VCH) VCH
Direction: Inbound				
1	81	2010000	(FA1) 0100237FF87FF8000000	(FA1) 0100237FF87FF8000000
2	AC	E001000	(VCH) 09B0880CC621F680A82600	(VCH) 09B0880CC621F680A82600
3	AC	A08DFFC	↑	↑
4	AC	6087FF8	↑	↑
5	AC	2000000	↑	↑
6	81	2010000	(FA1) 0800207FF87FF8000000	(FA1) 0800207FF87FF8000000
Direction: Outbound				
1	83	Same as the inbound data sequence		
2	AE			
3	AE			
4	AE			
5	AE			
6	83			

Table 6.6-36 Data Sequence before coding (9600 bps/EFR, Pattern 8)

No	LICH	SACCH	(FA1) FACCH1 (VCH) VCH	(FA1) FACCH1 (VCH) VCH
Direction: Inbound				
1	81	3F10000	(FA1) 010023FFEFFFEF000000	(FA1) 010023FFEFFFEF000000
2	AC	FF01000	(VCH) 09B0880CC621F680A82600	(VCH) 09B0880CC621F680A82600
3	AC	BF8FFF8	↑	↑
4	AC	7FFFFEC	↑	↑
5	AC	3FC0000	↑	↑
6	81	3F10000	(FA1) 080020FFEFFFEF000000	(FA1) 080020FFEFFFEF000000
Direction: Outbound				
1	83	Same as the inbound data sequence		
2	AE			
3	AE			
4	AE			
5	AE			
6	83			

Table 6.6-37 Data Sequence before coding (9600 bps/EFR, Pattern 9)

6.6.1.4. Data Format

This section explains the testing procedure regarding the decode processing of functional channels (LICH, UDCH, and FACCH2) used for data communications on traffic channel.
There are two kinds of tests in this section, one is a FACCH2 test with the frame for Status Call and the other is a UDCH test with the frame for Data Call.

- Connection Diagram

See Figure 6.5-1.

- Test Procedure

(1) Select appropriate SSG data used for the test.
(2) Transmit the signal selected at the step (1) from an SSG, the monitor the data sequence decoded by the unit under test with Monitor PC.
(3) For FACCH2 test, compare the monitored data sequence with the data sequence of LICH/FACCH2 before coding as shown in Table 6.6-40 to Table 6.6-42 corresponding to the SSG data selected at the step (1).
If the two data sequences are the same, the unit under test passed this test.
The relation between the SSG data and the data sequence before coding as a comparison object is shown in Table 6.6-38.
(4) For UDCH test, compare the monitored data sequence with the data sequence of LICH/UDCH/FACCH2 before coding as shown in Table 6.6-43 to Table 6.6-48 corresponding to the SSG data selected at the step (1).
If the two data sequences are the same, the unit under test passed this test.
The relation between the SSG data and the data sequence before coding as a comparison object is shown in Table 6.6-39.
(5) Perform tests of the step (1) to the step (4) using all of the SSG data as shown in Table 6.6-38 to Table 6.6-39.

Table 6.6-40 to Table 6.6-48 show the data which combines LICH data before dibit , UDCCH / FACCH2 data before coding and VCH data after voice coding as described in Section 4.1,

Mode & Pattern	SSG data	Data sequence before coding
Pattern 10	Table 6.7-16	Table 6.6-40
Pattern 11		Table 6.6-41
Pattern 12		Table 6.6-42

Table 6.6-38 Relation between SSG data and Data Sequence before coding (Status Call)

Mode & Pattern	SSG data	Data sequence before coding
4800 bps, Pattern 10	Table 6.7-14	Table 6.6-43
4800 bps, Pattern 11		Table 6.6-44
4800 bps, Pattern 12		Table 6.6-45
9600 bps, Pattern 10	Table 6.7-15	Table 6.6-46
9600 bps, Pattern 11		Table 6.6-47
9600 bps, Pattern 12		Table 6.6-48

Table 6.6-39 Relation between SSG data and Data Sequence before coding (Data Call)

No	LICH	(FA2) FACCH2
Direction: Inbound		
1	90	(FA2) 013200880001000200010000000000000000000000000000
2	90	(FA2) 010800800001000200000000000000000000000000000000
Direction: Outbound		
1	92	Same as the inbound data sequence
2	92	

Table 6.6-40 Data Sequence before coding (Status Call, Pattern 10)

No	LICH	(FA2) FACCH2
Direction: Inbound		
1	90	(FA2) 203200887FF87FF900010000000000000000000000000000
2	90	(FA2) 200800807FF87FF900000000000000000000000000000000
Direction: Outbound		
1	92	Same as the inbound data sequence
2	92	

Table 6.6-41 Data Sequence before coding (Status Call, Pattern 11)

No	LICH	(FA2) FACCH2
Direction: Inbound		
1	90	(FA2) 3F320088FFEFFFEE0001000000000000000000000000000000
2	90	(FA2) 3F080080FFEFFFEE0000000000000000000000000000000000
Direction: Outbound		
1	92	Same as the inbound data sequence
2	92	

Table 6.6-42 Data Sequence before coding (Status Call, Pattern 12)

No	LICH	(UD) UDCH (FA2) FACCH2
Direction: Inbound		
1	9C	(UD) 0109008000010002008004000000000000000000000000000
2	9C	(UD) 010B00414243444546474849 4A4B4C4D4E4F5053D6F830
3	90	(FA2) 0108008000010002000000000000000000000000000000000
Direction: Outbound		
1	9E	Same as the inbound data sequence
2	9E	
3	92	

Table 6.6-43 Data Sequence before coding (Data Call, 4800 bps, Pattern 10)

No	LICH	(UD) UDCH (FA2) FACCH2
Direction: Inbound		
1	9C	(UD) 200900807FF87FF900800400000000000000000000000000
2	9C	(UD) 200B00414243444546474849 4A4B4C4D4E4F5053D6F830
3	90	(FA2) 200800807FF87FF900000000000000000000000000000000
Direction: Outbound		
1	9E	Same as the inbound data sequence
2	9E	
3	92	

Table 6.6-44 Data Sequence before coding (Data Call, 4800 bps, Pattern 11)

No	LICH	(UD) UDCH (FA2) FACCH2
Direction: Inbound		
1	9C	(UD) 3F090080FFEFFFEE008004000000000000000000000000
2	9C	(UD) 3F0B00414243444546474849 4A4B4C4D4E4F5053D6F830
3	90	(FA2) 3F080080FFEFFFEE000000000000000000000000000000
Direction: Outbound		
1	9E	Same as the inbound data sequence
2	9E	
3	92	

Table 6.6-45 Data Sequence before coding (Data Call, 4800 bps, Pattern 12)

No	LICH	(UD) UDCH (FA2) FACCH2
Direction: Inbound		
1	9C	(UD) 0109008200010002008004000000000000000000000000
2	9C	(UD) 010B00414243444546474849 4A4B4C4D4E4F5053D6F830
3	90	(FA2) 0108008000010002000000000000000000000000000000
Direction: Outbound		
1	9E	Same as the inbound data sequence
2	9E	
3	92	

Table 6.6-46 Data Sequence before coding (Data Call, 9600 bps, Pattern 10)

No	LICH	(UD) UDCH (FA2) FACCH2
Direction: Inbound		
1	9C	(UD) 200900827FF87FF900800400000000000000000000000
2	9C	(UD) 200B00414243444546474849 4A4B4C4D4E4F5053D6F830
3	90	(FA2) 200800807FF87FF900000000000000000000000000000
Direction: Outbound		
1	9E	Same as the inbound data sequence
2	9E	
3	92	

Table 6.6-47 Data Sequence before coding (Data Call, 9600 bps, Pattern 11)

No	LICH	(UD) UDCH (FA2) FACCH2
Direction: Inbound		
1	9C	(UD) 3F090082FFEFFFEE008004000000000000000000000000
2	9C	(UD) 3F0B00414243444546474849 4A4B4C4D4E4F5053D6F830
3	90	(FA2) 3F080080FFEFFFEE00000000000000000000000000000000
Direction: Outbound		
1	9E	Same as the inbound data sequence
2	9E	
3	92	

Table 6.6-48 Data Sequence before coding (Data Call, 9600 bps, Pattern 12)

6.6.2. Transmitter Test

This section tests the validity of encoding processing of a unit under test.

Tests are conducted that a unit under test transmits the frame data which is encoded and constructed from given data sequence and the bit sequence demodulated by a monitor receiver is compared with the frame data corresponding to the used data sequence. The frame data for the comparison is the same as the data sequence of the SSG data used for receiver test.

6.6.2.1. Outbound Control Channel Test

This section explains the testing procedure regarding the encode processing of functional channels (LICH, CAC, and Collision Control Field) for outbound control channel. The test of Collision Control Field includes two cases that an inbound signal is not received (idle state) and is received. For the test while receiving inbound signal, VCALL_REQ (Pattern 4) of Short CAC described in Section 4.1.5 is used as the inbound signal.

- Connection Diagram

See Figure 6.5-3.

- Test Procedure 1

(1) Set a unit under test to send the data sequence before coding as shown in Table 6.6-2to Table 6.6-3.
(2) Start transmission from the unit under test, and monitor the frame data sequence received by Monitor Receiver with Monitor PC.
(3) Compare the monitored data sequence with the data sequence in Table 6.7-1 to Table 6.7-2 relevant to the pattern set up at the step (1).
If the two data sequences are the same, the unit under test passes this test.
The relation between the data sequence before coding and the frame data sequence which is compared with the monitored data sequence is shown in Table 6.6-49.
(4) Perform tests of the step (1) to the step (3) about all of the data sequence before coding as shown in Table 6.6-49.

- Test Procedure 2

For the test of Collision Control Field, an SSG may send the signal with Short CAC data sequence referring the connection diagram as shown in Figure 6.5-4.

(1) Start transmission of inbound control channel for 4800 bps, 9600 bps/EHR and 9600 bps/EFR from SSG as shown in Table 6.7-6 to Table 6.7-8.
(2) After the unit under test receives inbound signal, start transmission from the unit under test, and monitor the frame data sequence received by Monitor Receiver with Monitor PC.

(3) Compare the monitored data sequence with the data sequence in Table 6.7-1 to Table 6.7-2 relevant to the pattern set up at the step (1).
If the two data sequences are the same, the unit under test passes this test.
The relation between the data sequence before coding and the frame data sequence which is compared with the monitored data sequence is shown in Table 6.6-49.

(4) Perform tests of the step (1) to the step (3) about all of the data sequence before coding as shown in Table 6.6-49.

Format	Inbound Signal Condition (SSG Data)	Data sequence before coding	Frame data sequence for comparison
Single Message Format	No signal received	Table 6.6-2	Table 6.7-1
	4800 bps (Table 6.7-6)		
	9600 bps/EHR (Table 6.7-7)		
	9600 bps/EFR (Table 6.7-8)		
Dual Message Format	No signal received	Table 6.6-3	Table 6.7-2
	4800 bps (Table 6.7-6)		
	9600 bps/EHR (Table 6.7-7)		
	9600 bps/EFR (Table 6.7-8)		

Table 6.6-49 Frame Data Sequence for Comparison

6.6.2.2. Inbound Control Channel Test

This section explains the testing procedure regarding the encode processing of functional channels (LICH, L-CAC, and S-CAC) for inbound control channel.

The SSG data as shown in Table 6.7-3 is an example of the frame data for outbound control channel. This frame data consists of the messages based on the configuration of the any Trunked System. For the test, the unit under test is required to do the random access transmission based on the configuration of the Trunked system as shown in Table 6.7-3.

- Connection Diagram

See Figure 6.5-4.

6.6.2.2.1. Long CAC Tests

- Test Procedure

(1) Start transmission of outbound control channel from an SSG as shown in Table 6.7-3.
(2) Set the unit under test to send the data sequence before coding as shown in Table 6.6-5 to Table 6.6-10.
(3) Start transmission from the unit under test, and monitor the frame data sequence received by Monitor Receiver with Monitor PC.
(4) Compare the monitored data sequence with the data sequence in Table 6.7-4 to Table 6.7-5 relevant to the pattern set up at the step (2).
If the two data sequences are the same, the unit under test passes this test.
The relation between the data sequence before coding and the frame data sequence which is compared with the monitored data sequence is shown in Table 6.6-50.
(5) Perform tests of the step (1) to the step (4) about all of the data sequence before coding as shown in Table 6.6-50.

Mode & Pattern	Data sequence before coding	Frame data sequence for comparison
4800 bps, Pattern 1	Table 6.6-5	Table 6.7-4
4800 bps, Pattern 2	Table 6.6-6	
4800 bps, Pattern 3	Table 6.6-7	
9600 bps, Pattern 1	Table 6.6-8	Table 6.7-5
9600 bps, Pattern 2	Table 6.6-9	
9600 bps, Pattern 3	Table 6.6-10	

Table 6.6-50 Frame Data Sequence for Comparison

6.6.2.2.2. Short CAC Tests

- Test Procedure

(1) Start transmission of outbound control channel from an SSG as shown in Table 6.7-3.
(2) Set the unit under test to send the data sequence before coding as shown in Table 6.6-13 to Table 6.6-27.
(3) Start transmission from the unit under test, and monitor the frame data sequence received by Monitor Receiver with Monitor PC.
(4) For the frame test with VCALL_REQ message, compare the monitored data sequence with the data sequence in Table 6.7-6 to Table 6.7-8 relevant to the pattern set up at the step (2). If the two data sequences are the same, the unit under test passes this test.
The relation between the data sequence before coding and the frame data sequence which is compared with the monitored data sequence is shown in Table 6.6-51.
(5) Perform tests of the step (1) to the step (4) using all of the SSG data files with VCALL_REQ message as shown in Table 6.6-51.
(6) Perform tests of the step (1) to the step (4) using all of the SSG data files with DCALL_REQ message as shown in Table 6.6-52.

Mode & Pattern	Data sequence before coding	Frame data sequence for comparison
4800 bps, Pattern 4	Table 6.6-13	Table 6.7-6
4800 bps, Pattern 5	Table 6.6-14	
4800 bps, Pattern 6	Table 6.6-15	
9600 bps / EHR, Pattern 4	Table 6.6-16	Table 6.7-7
9600 bps / EHR, Pattern 5	Table 6.6-17	
9600 bps / EHR, Pattern 6	Table 6.6-18	
9600 bps / EFR, Pattern 4	Table 6.6-19	Table 6.7-8
9600 bps / EFR, Pattern 5	Table 6.6-20	
9600 bps / EFR, Pattern 6	Table 6.6-21	

Table 6.6-51 Frame Data Sequence for Comparison (VCALL_REQ)

Mode & Pattern	Data sequence before coding	Frame data sequence for comparison
4800 bps, Pattern 1	Table 6.6-22	Table 6.7-9
4800 bps, Pattern 2	Table 6.6-23	
4800 bps, Pattern 3	Table 6.6-24	
9600 bps, Pattern 1	Table 6.6-25	Table 6.7-10
9600 bps, Pattern 2	Table 6.6-26	
9600 bps, Pattern 3	Table 6.6-27	

Table 6.6-52 Frame Data Sequence for Comparison (DCALL_REQ)

6.6.2.3. Voice Format

This section explains the testing procedure regarding the encode processing of functional channels (LICH, SACCH, VCH, and FACCH1) used for voice communication on traffic channel.

- Connection Diagram

See Figure 6.5-3.

- Test Procedure

(1) Set the unit under test to send the data sequence before coding as shown in Table 6.6-29 to Table 6.6-37.
(2) Start transmission from the unit under test, and monitor the frame data sequence received by Monitor Receiver with Monitor PC.
(3) Compare the monitored data sequence with the data sequence in Table 6.7-11 to Table 6.7-13 relevant to the pattern set up at the step (1).
If the two data sequences are the same, the unit under test passes this test.
The relation between the data sequence before coding and the frame data sequence which is compared with the monitored data sequence is shown in Table 6.6-53.
(4) Perform tests of the step (1) to the step (3) about all of the data sequence before coding as shown in Table 6.6-53.

Mode & Pattern	Data sequence before coding	Frame data sequence for comparison
4800 bps, Pattern 7	Table 6.6-29	Table 6.7-11
4800 bps, Pattern 8	Table 6.6-30	
4800 bps, Pattern 9	Table 6.6-31	
9600 bps / EHR, Pattern 7	Table 6.6-32	Table 6.7-12
9600 bps / EHR, Pattern 8	Table 6.6-33	
9600 bps / EHR, Pattern 9	Table 6.6-34	
9600 bps / EFR, Pattern 7	Table 6.6-35	Table 6.7-13
9600 bps / EFR, Pattern 8	Table 6.6-36	
9600 bps / EFR, Pattern 9	Table 6.6-37	

Table 6.6-53 Frame Data Sequence for Comparison

6.6.2.4. Data Format

This section explains the testing procedure regarding the encode processing of functional channels (LICH, UDCH, and FACCH2) used for data communications on traffic channel.
There are two kinds of test in this section, one is a FACCH test with the frame for Status Call and the other is a UDCH test with the frame for Data Call

- Connection Diagram

See Figure 6.5-3.

- Test Procedure

(1) Set the unit under test to send the data sequence before coding as shown in Table 6.6-40 to Table 6.6-48.
(2) Start transmission from the unit under test, and monitor the frame data sequence received by Monitor Receiver with Monitor PC.
(3) For FACCH2 Test, compare the monitored data sequence with the data sequence in Table 6.7-16 relevant to the pattern set up at the step (1).
If the two data sequences are the same, the unit under test passes this test.
The relation between the data sequence before coding and the frame data sequence which is compared with the monitored data sequence is shown in Table 6.6-54.
(4) For UDCH Test, compare the monitored data sequence with the data sequence in Table 6.7-14 to Table 6.7-15 relevant to the pattern set up at the step (1).
If the two data sequences are the same, the unit under test passes this test.
The relation between the data sequence before coding and the frame data sequence which is compared with the monitored data sequence is shown in Table 6.6-55.
(5) Perform tests of the step (1) to the step (4) about all of the data sequence before coding as shown in Table 6.6-54 and Table 6.6-55.

Mode & Pattern	Data sequence before coding	Frame data sequence for comparison
Pattern 10	Table 6.6-40	Table 6.7-16
Pattern 11	Table 6.6-41	
Pattern 12	Table 6.6-42	

Table 6.6-54 Frame Data Sequence for Comparison (Status Call)

Mode & Pattern	Data sequence before coding	Frame data sequence for comparison
4800 bps, Pattern 10	Table 6.6-43	Table 6.7-14
4800 bps, Pattern 11	Table 6.6-44	
4800 bps, Pattern 12	Table 6.6-45	
9600 bps, Pattern 10	Table 6.6-46	Table 6.7-15
9600 bps, Pattern 11	Table 6.6-47	
9600 bps, Pattern 12	Table 6.6-48	

Table 6.6-55 Frame Data Sequence for Comparison (Data Call)

6.6.2.5. Preamble and FSW Tests

This section explains the testing procedure to verify a preamble and FSW pattern used on RDCH or an FSW pattern used on RCCH/RTCH.

- Connection Diagram

See Figure 6.5-5.

- Test Procedure

(1) RAN and ID for the unit under test are not defined in particular.
(2) Start transmission from the unit under test, and monitor a waveform of an FM demodulated output signal. Record the waveform included preamble and FSW.
(3) Compare the recorded waveform at the step (2) with Figure 6.6-1 and verify these waveforms are the same. Figure 6.6-1 shows an example of the waveform that contains the repetitive symbol pattern “+3, +3, -3, -3” for eight symbols at the head of a preamble. Verify the number of symbols as being set in the unit under test since this repetitive pattern length is not defined.

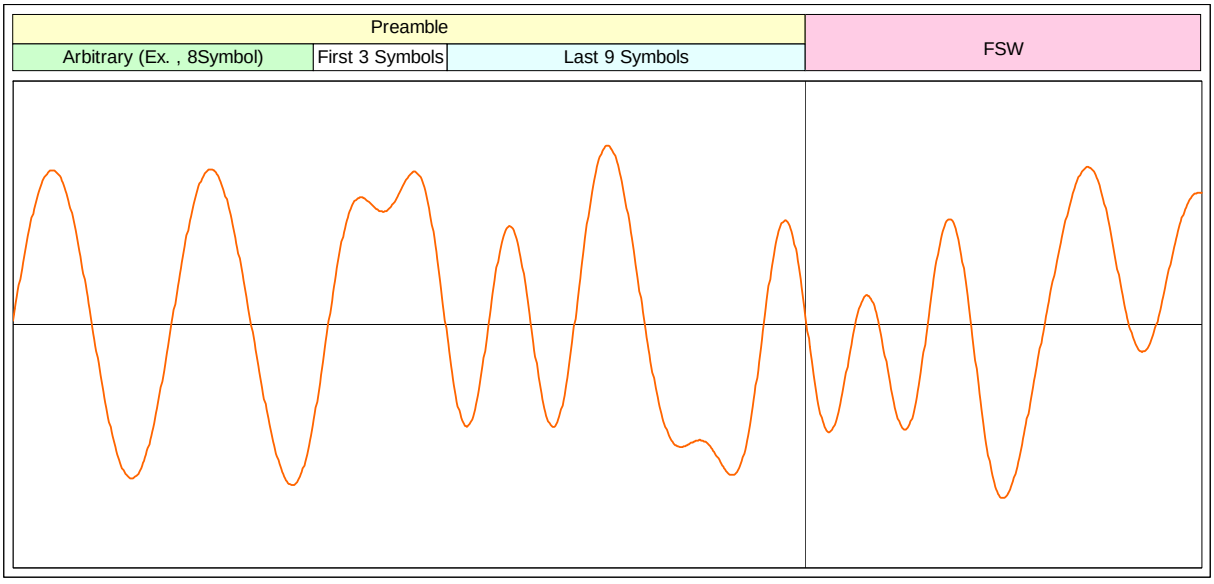


Figure 6.6-1 Waveform of the Preamble and FSW

6.6.2.6. Post Field Test for Transmission

This section explains the testing procedure to verify the sending pattern of the post field used on RCCH.

- Connection Diagram

See Figure 6.5-5.

- Test Procedure

(1) A message on outbound control channel sent from the unit under test is not defined in particular.
(2) Start transmission from the unit under test, and monitor a waveform of an FM demodulated output signal. Record the waveform including post field.
(3) Compare the recorded waveforms at the step (2) with Figure 6.6-2 and verify these waveforms are the same.

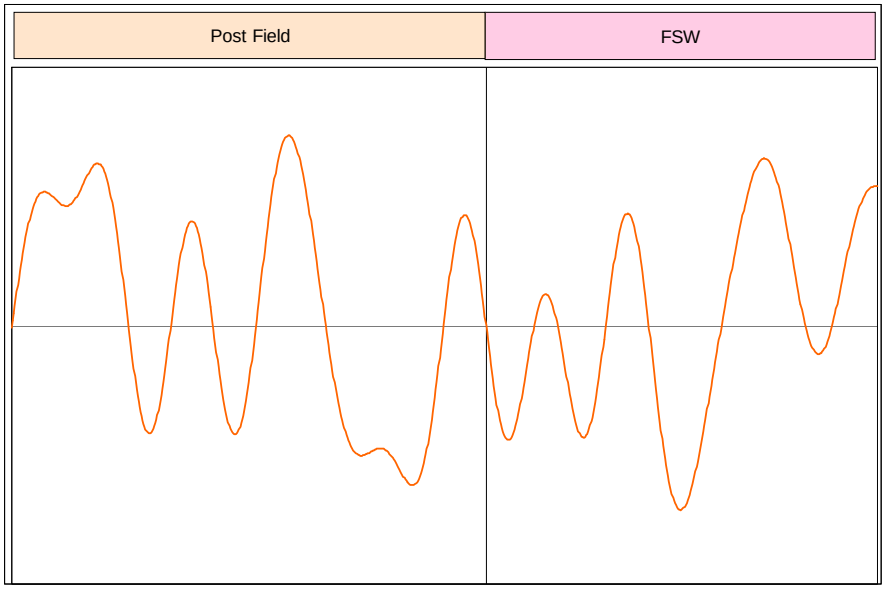


Figure 6.6-2 Waveform of Post Field

6.7. Test Frame

This section shows the SSG data sequence used for the test. Table 4.1-1 to Table 4.1-4 show ID setting for the frame data. The preamble “5775FD” at the head of the frame data for RDCH may be added if necessary.

6.7.1. CAC Frame

The frame data to test an LICH, a CAC and a collision control field of an outbound control channel is presented as follows.

Refer to Section 4.1.1 for an LICH, Section 4.1.2 for a CAC, and Section 4.1.5 for a collision control field about the details of data strings respectively.

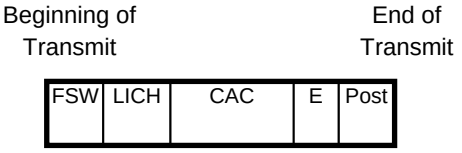


Figure 6.7-1 CAC Frame Structure

In the case of a Single Message Format:
LICH: For CAC testing
CAC: SITE_INFO
Collision Control Field: While in the idle state
4800 bps VCALL_REQ Received
9600 bps/EHR VCALL_REQ Received
9600 bps/EFR VCALL_REQ Received

Frame Data

FSW	CDF59
LICH	5D77
CAC	FAC8A071EA80685826A6988AAC2AAB8F1082303FA0A38282932600A008A21643D2AA430A2C1
E	758A08 (While in the Idle state) 77CFB9 (4800 bps VCALL_REQ Received) 7785F2 (9600 bps/EHR VCALL_REQ Received) 7728C7 (9600 bps/EFR VCALL_REQ Received)
Post	5775FD

Table 6.7-1 CAC Frame Data (Single Message Format)

In the case of a Dual Message Format:
LICH: For CAC testing
CAC: SRV_INFO/ADJ_SITE_INFO
Collision Control Field: While in the idle state
4800 bps VCALL_REQ Received
9600 bps/EHR VCALL_REQ Received
9600 bps/EFR VCALL_REQ Received

Frame Data

FSW	CDF59
LICH	5D77
CAC	798480302AF8C02950E68010AE0AF40300C33EB68485C690C2A508805B224282A6AAA300186
E	758A08 (While in the Idle state) 77CFB9 (4800 bps VCALL_REQ Received) 7785F2 (9600 bps/EHR VCALL_REQ Received) 7728C7 (9600 bps/EFR VCALL_REQ Received)
Post	5775FD

Table 6.7-2 CAC Frame Data (Dual Message Format)

6.7.2. Outbound Control Channel Frame for Inbound Control Channel Test

The following is the frame data for outbound control channel used in the inbound control channel tests as described in Section 6.6.2.2. Figure 6.7-2 shows a superframe structure on outbound control channel which consists of 13 frames according to Channel Structure Information. Collision Control Field at all the frames contains the data sequence of the idle state.

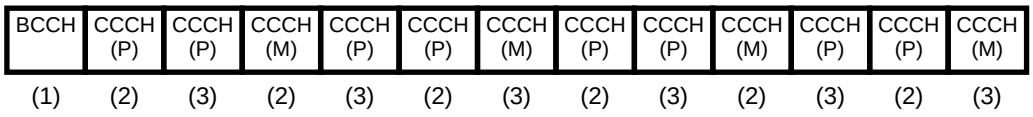


Figure 6.7-2 Outbound Control Channel Frame Data

```
Frame (1) Message
  [SITE_INFO]
    RAN :                        000001b
    Message Type :               011000b
    Location ID :                400021
      (Category=01b,System Code=0000000000000001b,Site Code=00001b)
    Channel          Structure   4914
    Information :
      (Bn=1,Gn=1,Pn=2,Mn=1,In=4)
    Service Information :        BFC0
    Restriction Information :    000000
    Channel             Access   000000
    Information :
    Version Number :             00
    Adjacent Site Allocation :   1
    1st Control Channel :        0000000001b
    2nd Control Channel :        0000000000b
```

Frame (2) Message
[SRV_INFO]

```
RAN :                          000001b
Message Type :                 011001b
Location ID :                  400021
  (Category=01b,System Code=0000000000000001b,Site Code=00001b)
Service Information :          BFC0
Restriction Information :      000000
```

Frame (3) Message
[ADJ_SITE_INFO]

```
RAN :                          000001b
Message Type :                 011011b
Adjacent Site Location ID :    400022
  (Category=01b,System Code=0000000000000001b,Site Code=00010b)
Adjacent Site Option :         000001b
Adjacent Control Channel :     0000000010b
```

Frame(1)	FSW	CDF59
	LICH	5D77
	CAC	FAC8A071EA80685826A6988AAC2AAB8F1082303FA0A38282932600A008A21643D2AA430A2C1
	E	758A08
	Post	5775FD
Frame(2)	FSW	CDF59
	LICH	5D77
	CAC	7988A1B02A88A02820468088B60AAA2B00823CB6A08CC68285A500A01B2203C2A2AA0302281
	E	758A08
	Post	5775FD
Frame(3)	FSW	CDF59
	LICH	5D77
	CAC	5B88A1A1EA88A08820C69088C402AA2230822CA2A0848882818600A018A203C3A2AA0322287
	E	758A08
	Post	5775FD

Table 6.7-3 CAC Frame Data for Inbound Control Channel Test

6.7.3. Long CAC Frame

The frame data to test an LICH and a Long CAC of an inbound control channel is presented as follows. The frame data is for the case that the number of frames in an SDCALL_REQ (User Data) message is 1 as presented in Figure 6.7-3. Refer to Section 4.1.1 for an LICH and Section 4.1.3 for a Long CAC about the details of data strings respectively.

Frame	LICH	LCAC
(1)	For Long CAC for testing	SDCALL_REQ (Header)
(2)	For Long CAC for testing	SDCALL_REQ (User Data)

Figure 6.7-3 Long CAC Frame Structure

Long CAC Frame Data (4800 bps)

		Pattern 1	Pattern 2	Pattern 3
Frame(1)	FSW	CDF59	←	←
	LICH	5F7D	←	←
	LCAC	08A0A30A2A0420AE05620 4A1A382A221228202E0A3 C048B006006410272722F	0A60A3202A4420A80A620 469A00762391C82A2D0A1 CDC8A8220127502F3622B	0B20B3262AC600A21A625 469A18362211C828380A5 C5C890420125102D39228
Frame(2)	FSW	CDF59	←	←
	LICH	5F7D	←	←
	LCAC	2F42527B8681B3783409B 7158D90EFE184386134C5 DE7EC8491F21071719F89	←	←

Table 6.7-4 Long CAC Frame Data (4800 bps)

Long CAC Frame Data (9600 bps)

		Pattern 1	Pattern 2	Pattern 3
Frame(1)	FSW	CDF59	←	←
	LICH	5F7D	←	←
	LCAC	00A0BB0A2AC420A015626 421A102A2210282E3E0A5 C048A80600A4102F27229	0260BB202A8420A61A626 4E9A28762393C8243D0A7 CDC8B02201E750273622D	0320AB262A0600AC0A623 4E9A30362213C826280A3 C5C8884201E510253922E
Frame(2)	FSW	CDF59	←	←
	LICH	5F7D	←	←
	LCAC	2F42527B8681B3783409B 7158D90EFE184386134C5 DE7EC8491F21071719F89	←	←

Table 6.7-5 Long CAC Frame Data (9600 bps)

6.7.4. Short CAC Frame

The frame data to test an LICH and a Short CAC of an inbound control channel is presented as follows. Refer to Section 4.1.1 for an LICH and Section 4.1.4 for a Short CAC about the details of data strings respectively.

In the case of Voice Call:

In the case of 4800 bps/ EHR:
LICH: For Short CAC testing
Short CAC: VCALL_REQ
Frame Data

	Pattern 4	Pattern 5	Pattern 6
FSW	CDF59	←	←
LICH	577 F	←	←
SCAC	1C98A02E2BDC2CAC81026 000A02EAA2CA20265BEA3 C2889BC38192E036E0423	1D68803E2ADCD8A881620 034A0AEAA34BB82651EA3 C4E8FBC783531020E8026	1F68B02BAB1C94AA82426 074A02C2A28BF82258EA3 C448BBE380520038E8820

Table 6.7-6 Short CAC Frame Data for Voice Call (4800 bps/EHR)

In the case of 9600 bps/ EHR:
LICH: For Short CAC testing
Short CAC: VCALL_REQ
Frame Data

	Pattern 4	Pattern 5	Pattern 6
FSW	CDF59	←	←
LICH	577 F	←	←
SCAC	1898B80E2ADC2CAC89020 000A1ACAA38B202C5BEA0 C688BBE38152E028E042E	1968981E2BDCD8A889626 034A12CAA20AB82C51EA0 C0E8DBE78393103EE802B	1B68A80BAA1C94AA8A420 074A1AE2A3CAF82858EA0 C0489BC380920026E882D

Table 6.7-7 Short CAC Frame Data for Voice Call (9600 bps/EHR)

In the case of 9600 bps/ EFR:
LICH: For Short CAC testing
Short CAC: VCALL_REQ
Frame Data

	Pattern 4	Pattern 5	Pattern 6
FSW	CDF59	←	←
LICH	577 F	←	←
SCAC	1898980E2ADD2CA081025 000A12EAA24B202E53EA2 C288D3E38253E02AE8427	1968B81E2BDDD8A481623 034A1AEAA3CAB82E59EA2 C4E8B3E78092103CE0022	1B68880BAA1D94A682425 074A12C2A20AF82A50EA2 C448F3C383930024E0824

Table 6.7-8 Short CAC Frame Data for Voice Call (9600 bps/EFR)

In the case of Data Call:

In the case of 4800 bps:
LICH: For Short CAC testing
Short CAC: DCALL_REQ
Frame Data

	Pattern 1	Pattern 2	Pattern 3
FSW	CDF59	←	←
LICH	577 F	←	←
SCAC	3098914EAA9C28A8D1221 082A1BEBA3C0282C4BEA2 EA88BAC38010E83EE0027	3178B95E2BDCDCACD1424 0B4A23EAA381B02241AA2 ECE8C2C68291102CE8420	3378894BAA1C90AED2622 0F4A2BC2A241F02648AA2 EC4882E281900034E8C26

Table 6.7-9 Short CAC Frame Data for Data Call (4800 bps)

In the case of 9600 bps:
LICH: For Short CAC testing
Short CAC: DCALL_REQ
Frame Data

	Pattern 1	Pattern 2	Pattern 3
FSW	CDF59	←	←
LICH	577 F	←	←
SCAC	3498896EAB9C28A8D9227 082A03CBA28128264BEA1 EE889AE380D0E820E002A	3578A17E2ADCDCACD9422 0B4A3BCAA2C0B02841AA1 E8E8E2E682511032E842D	3778916BAB1C90AEDA624 0F4A33E2A300F02C48AA1 E848A2C28150002AE8C2B

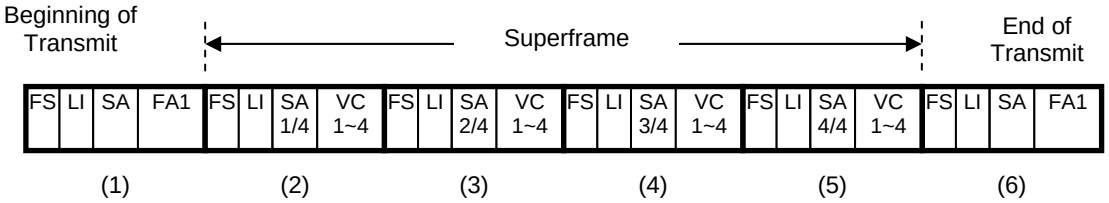
Table 6.7-10 Short CAC Frame Data for Data Call (9600 bps)

6.7.5. Voice Frame

This Voice Frame is a frame to be utilized for voice communications on RDCH, and is frame data to test an LICH, SACCH, VCH and FACCH1 which are used for voice communications. There are three modes available for voice communications, and each frame structure is presented in Figure 6.7-4 to Figure 6.7-6.

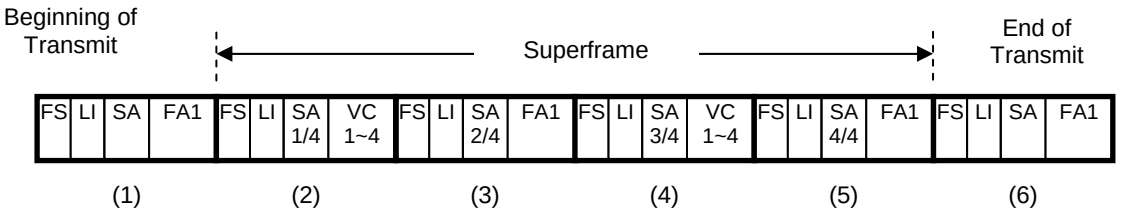
Refer to Section 4.1.1 for an LICH, Section 4.1.6 for an SACCH, Section 4.1.7 for a VCH and Section 4.1.8 for an FACCH1 about the details of data strings respectively. Besides, an LICH setting shows only an inbound direction. When the LICH setting is replaced with an outbound direction, refer to the values presented in Section 4.1.1.

The test for voice communications on RTCH uses LICH modified for RTCH referring to Section 4.1.1.



Frame	LICH	SACCH	VCH / FACCH1	VCH	VCH / FACCH1	VCH
(1)	RDCH, SACCH non-superframe, FACCH1	IDLE	VCALL		VCALL	
(2)	RDCH, SACCH superframe, VCH	VCALL	EHR Tone Test Pattern			
(3)	RDCH, SACCH superframe, VCH	VCALL	EHR Tone Test Pattern			
(4)	RDCH, SACCH superframe, VCH	VCALL	EHR Tone Test Pattern			
(5)	RDCH, SACCH superframe, VCH	VCALL	EHR Tone Test Pattern			
(6)	RDCH, SACCH non-superframe, FACCH1	IDLE	TX_REL		TX_REL	

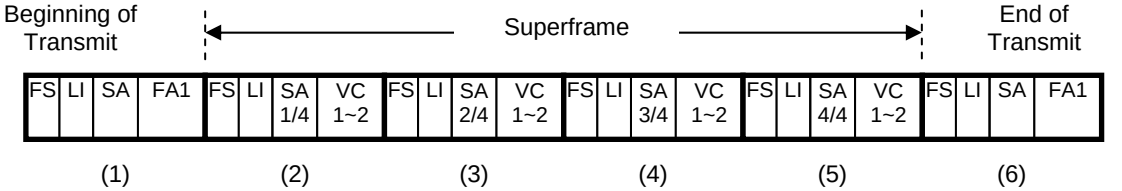
Figure 6.7-4 Voice Call Frame Structure for 4800 bps/EHR



FS: FSW SA: SACCH VC: VCH
LI: LICH FA1: FACCH1

Frame	LICH	SACCH	VCH	VCH	VCH	VCH
			FACCH1		FACCH1	
(1)	RDCH, SACCH non-superframe, FACCH1	IDLE	VCALL		VCALL	
(2)	RDCH, SACCH superframe, VCH	VCALL	EHR Tone Test Pattern			
(3)	RDCH, SACCH superframe, FACCH1	VCALL	VCALL		VCALL	
(4)	RDCH, SACCH superframe, VCH	VCALL	EHR Tone Test Pattern			
(5)	RDCH, SACCH superframe, FACCH1	VCALL	VCALL		VCALL	
(6)	RDCH, SACCH non-superframe, FACCH1	IDLE	TX_REL		TX_REL	

Figure 6.7-5 Voice Call Frame Structure for 9600 bps/EHR



FS: FSW SA: SACCH VC: VCH
LI: LICH FA1: FACCH1

Frame	LICH	SACCH	VCH	VCH
			FACCH1	FACCH1
(1)	RDCH, SACCH non-superframe, FACCH1	IDLE	VCALL	VCALL
(2)	RDCH, SACCH superframe, VCH	VCALL	EFR Tone Test Pattern	
(3)	RDCH, SACCH superframe, VCH	VCALL	EFR Tone Test Pattern	
(4)	RDCH, SACCH superframe, VCH	VCALL	EFR Tone Test Pattern	
(5)	RDCH, SACCH superframe, VCH	VCALL	EFR Tone Test Pattern	
(6)	RDCH, SACCH non-superframe, FACCH1	IDLE	TX_REL	TX_REL

Figure 6.7-6 Voice Call Frame Structure for 9600 bps/EFR

Voice Call Frame Data for 4800 bps/EHR

		Pattern 7	Pattern 8	Pattern 9
Frame(1)	FSW	CDF59	←	←
	LICH	DD7D	←	←
	SACCH	63C2B42E2A8A2A1	21C0A808230E491	6980A8CB3A124E0
	FACCH1	CAB621281A20AEB08B 0C206EF0182A866988	DE3E200B0B61EA42C2 30B6AAB509B306CD88	DE3A222B08296EC0E2 15B4A2F52933468998
	FACCH1	48BE03A2B0288638A3 242064F29A0AAEEBA2	5C360281A169C2CAEA 18B6A0B78B932E4FA2	5C3200A1A2214648CA 3DB4A8F7AB136E0BB2
Frame(2)	FSW	CDF59	←	←
	LICH	D5DF	←	←
	SACCH	CFA0A6E8A2356E8	8DA2BACEABB10D8	C5E2BA0DB2AD0A9
	VCH	4CAADE8B26E4F28288	←	←
	VCH	C68A7429A4ECD00822	←	←
	VCH	CEA2FC018CECDA0AA0	←	←
	VCH	EE8A7E2B26CCF88A08	←	←
Frame(3)	FSW	CDF59	←	←
	LICH	D5DF	←	←
	SACCH	C6DBB0EB3A426A8	8553DD5D56EE85C	E55BCF8E88EA06D
	VCH	4CAADE8B26E4F28288	←	←
	VCH	C68A7429A4ECD00822	←	←
	VCH	CEA2FC018CECDA0AA0	←	←
	VCH	EE8A7E2B26CCF88A08	←	←
Frame(4)	FSW	CDF59	←	←
	LICH	D5DF	←	←
	SACCH	63A1B4A81A8E280	21318F9F9162937	697B6974C16490E
	VCH	4CAADE8B26E4F28288	←	←
	VCH	C68A7429A4ECD00822	←	←
	VCH	CEA2FC018CECDA0AA0	←	←
	VCH	EE8A7E2B26CCF88A08	←	←
Frame(5)	FSW	CDF59	←	←
	LICH	D5DF	←	←
	SACCH	4288302B02D07E2	0888A88C0B048F9	48D8384F3A02DE0
	VCH	4CAADE8B26E4F28288	←	←
	VCH	C68A7429A4ECD00822	←	←
	VCH	CEA2FC018CECDA0AA0	←	←
	VCH	EE8A7E2B26CCF88A08	←	←
Frame(6)	FSW	CDF59	←	←
	LICH	DD7D	←	←
	SACCH	63C2B42E2A8A2A1	21C0A808230E491	6980A8CB3A124E0
	FACCH1	8A1661281A28ACB08A 8D20EEE02032846C8A	9E9E600B0B69E842C3 B1B62AA531AB04C88A	9E9A622B08216CC0E3 94B422E5112B448C9A
	FACCH1	081E43A2B0208438A2 A520E4E2A212ACEEA0	1C964281A161C0CAEB 99B620A7B38B2C4AA0	1C9240A1A2294448CB BCB428E7930B6C0EB0

Table 6.7-11 Voice Call Frame Data for 4800 bps/EHR

Voice Call Frame Data for 9600 bps/EHR

		Pattern 7	Pattern 8	Pattern 9
Frame(1)	FSW	CDF59	←	←
	LICH	DD7D	←	←
	SACCH	63C2B42E2A8A2A1	21C0A808230E491	6980A8CB3A124E0
	FACCH1	CAA6212C1A29AAB2CA 0D202E90183A86698A	DE2E200F0B68EE4083 31B6EAD509A306CD8A	DE2A222F08206AC2A3 14B4E295292346899A
	FACCH1	48AE03A6B021823AE2 252024929A1AAEEBA0	5C260285A160C6C8AB 19B6E0D78B832E4FA0	5C2200A5A228424A8B 3CB4E897AB036E0BB0
Frame(2)	FSW	CDF59	←	←
	LICH	D5DF	←	←
	SACCH	CFA0A6E8A2356E8	8DA2BACEABB10D8	C5E2BA0DB2AD0A9
	VCH	4CAADE8B26E4F28288	←	←
	VCH	C68A7429A4ECD00822	←	←
	VCH	CEA2FC018CECDA0AA0	←	←
	VCH	EE8A7E2B26CCF88A08	←	←
Frame(3)	FSW	CDF59	←	←
	LICH	D57F	←	←
	SACCH	CFD9F2EB1B08683	8C519F5D77A4877	EC598D8EA9A0046
	FACCH1	CAA6212C1A29AAB2CA 0D202E90183A86698A	DE2E200F0B68EE4083 31B6EAD509A306CD8A	DE2A222F08206AC2A3 14B4E295292346899A
	FACCH1	48AE03A6B021823AE2 252024929A1AAEEBA0	5C260285A160C6C8AB 19B6E0D78B832E4FA0	5C2200A5A228424A8B 3CB4E897AB036E0BB0
Frame(4)	FSW	CDF59	←	←
	LICH	D5DF	←	←
	SACCH	63A1B4A81A8E280	21318F9F9162937	697B6974C16490E
	VCH	4CAADE8B26E4F28288	←	←
	VCH	C68A7429A4ECD00822	←	←
	VCH	CEA2FC018CECDA0AA0	←	←
	VCH	EE8A7E2B26CCF88A08	←	←
Frame(5)	FSW	CDF59	←	←
	LICH	D57F	←	←
	SACCH	4288302B02D07E2	0888A88C0B048F9	48D8384F3A02DE0
	FACCH1	CAA6212C1A29AAB2CA 0D202E90183A86698A	DE2E200F0B68EE4083 31B6EAD509A306CD8A	DE2A222F08206AC2A3 14B4E295292346899A
	FACCH1	48AE03A6B021823AE2 252024929A1AAEEBA0	5C260285A160C6C8AB 19B6E0D78B832E4FA0	5C2200A5A228424A8B 3CB4E897AB036E0BB0
Frame(6)	FSW	CDF59	←	←
	LICH	DD7D	←	←
	SACCH	63C2B42E2A8A2A1	21C0A808230E491	6980A8CB3A124E0
	FACCH1	8A1661281A28ACB08A 8D20EEE02032846C8A	9E9E600B0B69E842C3 B1B62AA531AB04C88A	9E9A622B08216CC0E3 94B422E5112B448C9A
	FACCH1	081E43A2B0208438A2 A520E4E2A212ACEEA0	1C964281A161C0CAEB 99B620A7B38B2C4AA0	1C9240A1A2294448CB BCB428E7930B6C0EB0

Table 6.7-12 Voice Call Frame Data for 9600 bps/EHR

Voice Call Frame Data for 9600 bps/EFR

		Pattern 7	Pattern 8	Pattern 9
Frame(1)	FSW	CDF59	←	←
	LICH	DD7D	←	←
	SACCH	63C2B42E2A8A2A1	21C0A808230E491	6980A8CB3A124E0
	FACCH1	CAA669281A29AE308B 2DB0EED008328E6D8A	DE2E680B0B68EAC2C2 11262A9519AB0EC98A	DE2A6A2B08206E40E2 342422D5392B4E8D9A
	FACCH1	48AE4BA2B02186B8A3 05B0E4D28A12A6EFA0	5C264A81A160C24AEA 392620979B8B264BA0	5C2248A1A22846C8CA 1C2428D7BB0B660FB0
Frame(2)	FSW	CDF59	←	←
	LICH	D5DF	←	←
	SACCH	CFA0A6E8A2356E8	8DA2BACEABB10D8	C5E2BA0DB2AD0A9
	VCH	BA90A4985E136A1C99 99A6C55A508D01F011	←	←
	VCH	38988612F41B4294B1 B1A6CF58D2AD297239	←	←
Frame(3)	FSW	CDF59	←	←
	LICH	D5DF	←	←
	SACCH	CFD3A27F3B10BAB	8C5BCFC957BC55F	EC53DD1A89B8D6E
	VCH	BA90A4985E136A1C99 99A6C55A508D01F011	←	←
	VCH	38988612F41B4294B1 B1A6CF58D2AD297239	←	←
Frame(4)	FSW	CDF59	←	←
	LICH	D5DF	←	←
	SACCH	63A1B4A81A8E280	21318F9F9162937	697B6974C16490E
	VCH	BA90A4985E136A1C99 99A6C55A508D01F011	←	←
	VCH	38988612F41B4294B1 B1A6CF58D2AD297239	←	←
Frame(5)	FSW	CDF59	←	←
	LICH	D5DF	←	←
	SACCH	4288302B02D07E2	0888A88C0B048F9	48D8384F3A02DE0
	VCH	BA90A4985E136A1C99 99A6C55A508D01F011	←	←
	VCH	38988612F41B4294B1 B1A6CF58D2AD297239	←	←
Frame(6)	FSW	CDF59	←	←
	LICH	DD7D	←	←
	SACCH	63C2B42E2A8A2A1	21C0A808230E491	6980A8CB3A124E0
	FACCH1	8A1661281A28ACB08A 8D20EEE02032846C8A	9E9E600B0B69E842C3 B1B62AA531AB04C88A	9E9A622B08216CC0E3 94B422E5112B448C9A
	FACCH1	081E43A2B0208438A2 A520E4E2A212ACEEA0	1C964281A161C0CAEB 99B620A7B38B2C4AA0	1C9240A1A2294448CB BCB428E7930B6C0EB0

Table 6.7-13 Voice Call Frame Data for 9600 bps/EFR

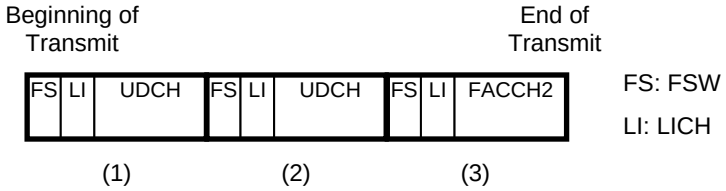
6.7.6. Data Frame

This Data Frame is a frame to be utilized for data and status communications on RDCH, and is frame data to test an LICH, UDCH and FACCH2 which are used for data and status communications. The frame structure for data communications are presented in Figure 6.7-7. The frame structure for status communications are presented in Table 6.7-8. The frame data is for the case that the number of frames in a DCALL (User Data) message is 1 as presented in Table 6.7-7.

Refer to Section 4.1.1 for an LICH, Section 4.1.9 for a UDCH, and Section 4.1.10 for an FACCH2 about the details of data strings respectively.

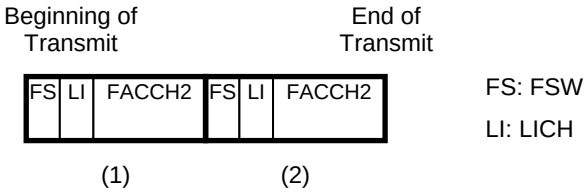
Besides, an LICH setting shows only an inbound direction. When the LICH setting is replaced with an outbound direction, refer to the values presented in Section 4.1.1.

The test for data communications on RTCH uses LICH modified for RTCH referring to Section 4.1.1.



Frame	LICH	UDCH / FACCH2
(1)	RDCH, UDCH	DCALL(Header)
(2)	RDCH, UDCH	DCALL(User Data)
(3)	RDCH, FACCH2	TX_REL

Figure 6.7-7 Data Call Frame Structure



Frame	LICH	FACCH2
(1)	RDCH, FACCH2	STAT_REQ
(2)	RDCH, FACCH2	TX_REL

Figure 6.7-8 Status Call Frame Structure

Data Call Frame Data for 4800 bps

		Pattern 10	Pattern 11	Pattern 12
Frame (1)	FSW	CDF59	←	←
	LICH	DFDF	←	←
	UDCH	4088A009608828E862 2202CA620ACE3AA080 008AA08E4922822C01 A02F72228839BBE202 2C2288AE0F4AA05	0748A00B428828A8D2 2205C7E20AE23EA080 22CAA098442282C4B9 A028B2628835C3E203 8FC288A10E4AA06	4348A01B4A88286942 2205C7E20AF20EA081 E30AA08F4322827DB1 A02EFAA28829C7E202 CFC288A81A4AA06
Frame (2)	FSW	CDF59	←	←
	LICH	DFDF	←	←
	UDCH	666A60102AF6D3EB36 E7B8C6CA348C679033 9F17E3B23397210A5F C6B65C1E6729F8DBC3 6A070970311A4EC	266A60122AF6D32B36 E7B9C6CA3490679033 3F17E3BC339721225F C6B09C1E6705F8DBC2 8A07097D311A4EA	666A600A2AF6D36B36 E7BFC6CA34A0679033 7F17E3B73397211A5F C6B1DC1E6701F8DBC3 4A07097C311A4EE
Frame (3)	FSW	CDF59	←	←
	LICH	DF7F	←	←
	FACCH2	4088A009608828A060 2200CA220ACC382080 008AA08AC982820800 A02E72028830BAA202 2C2A88A80F0AA06	0748A00B428828E0D0 2207C7A20AE03C2080 22CAA09CC48282E0B8 A029B242883CC2A203 8FCA88A70E0AA05	4348A01B4A88282140 2207C7A20AF00C2081 E30AA08BC3828259B0 A02FFA828820C6A202 CFCA88AE1A0AA05

Table 6.7-14 Data Call Frame Data for 4800 bps

Data Call Frame Data for 9600 bps

		Pattern 10	Pattern 11	Pattern 12
Frame (1)	FSW	CDF59	←	←
	LICH	DFDF	←	←
	UDCH	4088A009608828A862 2204DA620AEEBAA081 008AA0866922822C01 A02B7A228809FBE202 2E2288A21F4AA07	0748A00B428828E8D2 2203D7E20AC2BEA081 22CAA090642282C4B9 A02CBA62880583E203 8DC288AD1E4AA04	4348A01B4A88282942 2203D7E20AD28EA080 E30AA0876322827DB1 A02AF2A2881987E202 CDC288A40A4AA04
Frame (2)	FSW	CDF59	←	←
	LICH	DFDF	←	←
	UDCH	666A60102AF6D3EB36 E7B8C6CA348C679033 9F17E3B23397210A5F C6B65C1E6729F8DBC3 6A070970311A4EC	266A60122AF6D32B36 E7B9C6CA3490679033 3F17E3BC339721225F C6B09C1E6705F8DBC2 8A07097D311A4EA	666A600A2AF6D36B36 E7BFC6CA34A0679033 7F17E3B73397211A5F C6B1DC1E6701F8DBC3 4A07097C311A4EE
Frame (3)	FSW	CDF59	←	←
	LICH	DF7F	←	←
	FACCH2	4088A009608828A060 2200CA220ACC382080 008AA08AC982820800 A02E72028830BAA202 2C2A88A80F0AA06	0748A00B428828E0D0 2207C7A20AE03C2080 22CAA09CC48282E0B8 A029B242883CC2A203 8FCA88A70E0AA05	4348A01B4A88282140 2207C7A20AF00C2081 E30AA08BC3828259B0 A02FFA828820C6A202 CFCA88AE1A0AA05

Table 6.7-15 Data Call Frame Data for 9600 bps

Status Call Frame Data

		Pattern 10	Pattern 11	Pattern 12
Frame(1)	FSW	CDF59	←	←
	LICH	DF7F	←	←
	FACCH2	6888A010208828A060 22008A020AECB92080 348AA087C9C2820500 A02A5A128802BA2203 3C2E88A60F2AA03	2F48A012028828E0D0 220787820AC0BD2080 16CAA091C4C282EDB8 A02D9A52880EC22202 9FCE88A90E2AA00	6B48A0020A88282140 220787820AD08D2081 D70AA086C3C28254B0 A02BD2928812C62203 DFCE88A01A2AA00
Frame(2)	FSW	CDF59	←	←
	LICH	DF7F	←	←
	FACCH2	4088A009608828A060 2200CA220ACC382080 008AA08AC982820800 A02E72028830BAA202 2C2A88A80F0AA06	0748A00B428828E0D0 2207C7A20AE03C2080 22CAA09CC48282E0B8 A029B242883CC2A203 8FCA88A70E0AA05	4348A01B4A88282140 2207C7A20AF00C2081 E30AA08BC3828259B0 A02FFA828820C6A202 CFCA88AE1A0AA05

Table 6.7-16 Status Call Frame Data

6.8. Conditions of SSG

Some measuring instrument manufacturers are offering the product which meets the definition of SSG of Table 6.8-1. Either a measuring instrument corresponding to NXDN modulation method or a measuring instrument which can output NXDN signal by setting up conditions of a baseband filter and 4 level FSK by a user can be used.

Setting Item	Setup	
Data Output	Support an arbitrary waveform generating function or a real-time baseband signal generating function. Support a function to output Frame Data to be used in a test.	
Baseband Filter	A filter with the frequency characteristic of Figure 6.8-1	
Symbol Rate	4800 bps = 2.4 ksps, 9600 bps = 4.8 ksps	
Modulation Type	4800 bps Mode Data Freq. Deviation 00 350Hz 01 1050Hz 10 -350Hz 11 -1050Hz	9600 bps Mode Data Freq. Deviation 00 800Hz 01 2400Hz 10 -800Hz 11 -2400Hz

Table 6.8-1 SSG definition

The transmitter baseband filter consists of the Root Raised Cosine characteristic and the sinc characteristic. The frequency characteristic of the transmitter baseband filter is shown in Figure 6.8-1. It is desirable to design the order and coefficient of a digital filter appropriately so that that characteristic can become as close to this frequency characteristic as possible.

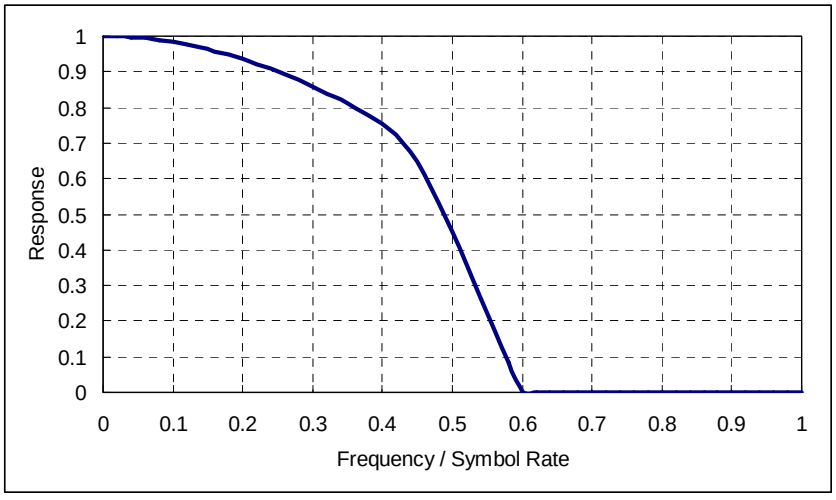


Figure 6.8-1 Frequency Characteristic of Transmitter Baseband Filter

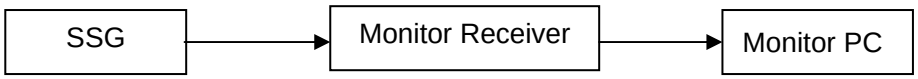
6.9. Conditions of Monitor Receiver

It is assumed that a product corresponding to the definition of Monitor Receiver is a modulation analysis instrument or a radio tester from measuring instrument manufacturers, or is a Jig which converted NXDN conformed (or partially conformed) radio equipment.
As a convenient method in the case that a measuring instrument can not be arranged, the conditions of Jig which converted radio equipment are explained below.

Monitor Receiver which can be used by this test shall be radio equipment which meets requirements of the test item of the Common Air Interface Test specifications of REF [1]. Consequently, a radio which has a demodulation processing of a NXDN signal and a decode processing of each functional channel can be used as Monitor Receiver.
Monitor Receiver also needs to have an output processing of the decoded information bit sequence. The examiner can conduct the transmitter test of a unit under test by displaying and analyzing an information bit sequence on Monitor PC connected to Monitor Receiver.

Radio equipment which passes the following test should be used as Monitor Receiver.

(1) Connect a Monitor Receiver as shown in Figure 6.9-1.
(2) An SSG corresponding to the conditions shown in Section 6.8 is used. The data structure of used in SSG is shown in Figure 6.9-2, and the data sequence is shown in Table 6.9-1.
(3) Transmit the data string shown in Table 6.9-1 from the SSG, and monitor the bit sequence received by the Monitor Receiver with Monitor PC.
(4) Verify that the monitored bit sequence in Monitor PC is the same as the bit sequence of the PN9 section shown in Table 6.9-1. The PN9 section is shown In Table 6.9-1 by the blue font. Dummy bits should be inserted at the end of SSG data when writing the SSG data of Table 6.9-1 in SSG.



Unit under test

Figure 6.9-1 Test Connection Diagram

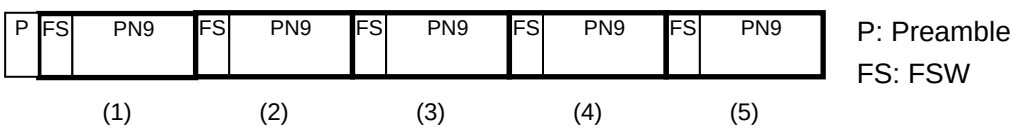


Figure 6.9-2 SSG Data Structure used in test

5775FD

CDF59FF83DF1732094ED1E7CD8A91C6D5C4C44021184E5586F4DC8A15A7EC92DF93533018CA34BFA2C759678FBA0D6DD

CDF5982D7D540A57977039D27AEA243385ED9A1DE1FF07BE2E64129DA3CF9B15238DAB89888042309CAB0DE9B9142B4F

CDF59D925BF26A6603194697F458EB2CF1F741ADBB05AFAA814AF2EE073A4F5D448670BDB343BC3FE0F7C5CC8253B479

CDF59F362A471B57131100846139561BD37228569FB24B7E4D4CC06328D2FE8B1D659E3EE835B760B5F550295E5DC0E7

CDF5949EBA890CE17B6687787FC1EF8B9904A768F3E6C548E36AE26220108C272AC37A6E450AD3F6496FC9A9980C651A

Table 6.9-1 SSG Data Sequence used by test

7. Appendix 2 (Type-D Trunked System)

The test procedure for each functional channel is described in Section 5. This section explains the test procedure using frame data which consists of different kinds of functional channels without testing each functional channel test individually. An examiner can perform the test procedure as described in Section 5 or in this section.

7.1. Test Items

Table 7.1-1 shows the comparison table for test items between Section 5 and Section 7. An examiner performs required tests according to operational modes implemented in a unit under test.

Test Items in Section 7			Test Items in Section 5
7.6.1.1	RX	Voice Format Test	5.2
7.6.1.2		Data Format Test	5.3
7.6.2.1	TX	Voice Format Test	5.4
7.6.2.2		Data Format Test	5.5
7.6.2.3		Preamble and FSW Tests	5.6
7.6.2.4		Post Field Test	5.7

Table 7.1-1 Comparison Table for Test Items between Section 5 and Section 7

7.2. Measuring Instruments

Measuring instruments required for tests are shown in Table 6.2-1.

7.3. SSG Data for Test

In the receiver test of SU, the behavior of a unit under test is verified by storing a test data to a memory of SSG and sending a NXDN signal from SSG.

The lists of SSG data used for tests are shown in Section 7.7. An examiner should create the required SSG data files and write them in SSG in advance.

7.4. Data Notation

Unless otherwise noted, a data sequence is expressed by a hexadecimal number.

7.5. Test Setup

Refer to Section 6.5 about the examination connection diagram to be used in this section.

7.6. Test Method

7.6.1. Receiver Test

This section tests the validity of decode processing of a unit under test. A test is conducted by sending the frame data from SSG and comparing the data decoded by a unit under test with the SSG data before coding.

A unit under test shall have a function which sends to Monitor PC the data sequence after decoding the received frame.

7.6.1.1. Voice Format

This section explains the testing procedure regarding the decode processing of functional channels (LICH, SCCH, VCH, and FACCH1) used for voice communications on traffic channel.

- Connection Diagram

See Figure 6.5-1.

- Test Procedure

(1) Select appropriate SSG data used for the test.

(2) Transmit the signal selected at the step (1) from an SSG, and monitor the data sequence decoded by the unit under test with Monitor PC.
Compare the monitored data sequence with the data sequence of LICH/SCCH/FACCH1/VCH before coding as shown in Table 7.6-2 to Table 7.6-4 corresponding to the SSG data selected at the step (1).
If the two data sequences are the same, the unit under test passes this test.
The relation between the SSG data and the data sequence before coding as a comparison object is shown in Table 7.6-1.

(3) Perform tests of the step (1) to the step (2) using all of the SSG data as shown in Table 7.6-1.

Table 7.6-2 to Table 7.6-4 show the data which combines LICH data before dibit , SCCH / FACCH1 data before coding and VCH data after voice coding as described in Section 4.1,

Pattern	SSG data	Data sequence before coding
Pattern 1	Table 7.7-1	Table 7.6-2
Pattern 2		Table 7.6-3
Pattern 3		Table 7.6-4

Table 7.6-1 Relation between SSG data and Data Sequence before coding

No	LICH	SCCH	(FA1) FACCH1 (VCH) VCH VCH	(FA1) FACCH1 (VCH) VCH VCH
Direction: Inbound				
1	C0	0108028	(FA1) 01008008010802000000	(FA1) 01008008010802000000
2	ED	C1F8000	(VCH) FEE2121212100 FEE2121212100	(VCH) FEE2121212100 FEE2121212100
3	ED	8108028	↑	↑
4	ED	4108010	↑	↑
5	ED	0108028	↑	↑
6	C0	1F08028	(FA1) 08008008010802000000	(FA1) 08008008010802000000
Direction: Outbound				
1	C2	0108028	(FA1) 01008008010802000000	(FA1) 01008008010802000000
2	EF	C218000	(VCH) FEE2121212100 FEE2121212100	(VCH) FEE2121212100 FEE2121212100
3	EF	8108028	↑	↑
4	EF	4208010	↑	↑
5	EF	0108028	↑	↑
6	C2	1F08028	(FA1) 08008208010802000000	(FA1) 08008208010802000000

Table 7.6-2 Data Sequence before coding (Pattern 1)

No	LICH	SCCH	(FA1) FACCH1 (VCH) VCH VCH	(FA1) FACCH1 (VCH) VCH VCH
Direction: Inbound				
1	C0	010BE88	(FA1) 0100800BE70BE8000000	(FA1) 0100800BE70BE8000000
2	ED	C1F8000	(VCH) FEE2121212100 FEE2121212100	(VCH) FEE2121212100 FEE2121212100
3	ED	810BE88	↑	↑
4	ED	410BE70	↑	↑
5	ED	010BE88	↑	↑
6	C0	1F0BE88	(FA1) 0800800BE70BE8000000	(FA1) 0800800BE70BE8000000
Direction: Outbound				
1	C2	010BE88	(FA1) 0100800BE70BE8000000	(FA1) 0100800BE70BE8000000
2	EF	C218000	(VCH) FEE2121212100 FEE2121212100	(VCH) FEE2121212100 FEE2121212100
3	EF	810BE88	↑	↑
4	EF	420BE70	↑	↑
5	EF	010BE88	↑	↑
6	C2	1F0BE88	(FA1) 0800820BE70BE8000000	(FA1) 0800820BE70BE8000000

Table 7.6-3 Data Sequence before coding (Pattern 2)

No	LICH	SCCH	(FA1) FACCH1 (VCH) VCH VCH	(FA1) FACCH1 (VCH) VCH VCH
Direction: Inbound				
1	C0	010FCF8	(FA1) 0100800FD00FCF000000	(FA1) 0100800FD00FCF000000
2	ED	C1F8000	(VCH) FEE2121212100 FEE2121212100	(VCH) FEE2121212100 FEE2121212100
3	ED	810FCF8	↑	↑
4	ED	410FD00	↑	↑
5	ED	010FCF8	↑	↑
6	C0	1F0FCF8	(FA1) 0800800FD00FCF000000	(FA1) 0800800FD00FCF000000
Direction: Outbound				
1	C2	010FCF8	(FA1) 0100800FD00FCF000000	(FA1) 0100800FD00FCF000000
2	EF	C218000	(VCH) FEE2121212100 FEE2121212100	(VCH) FEE2121212100 FEE2121212100
3	EF	810FCF8	↑	↑
4	EF	420FD00	↑	↑
5	EF	010FCF8	↑	↑
6	C2	1F0FCF8	(FA1) 0800820FD00FCF000000	(FA1) 0800820FD00FCF000000

Table 7.6-4 Data Sequence before coding (Pattern 3)

7.6.1.2. Data Format

This section explains the testing procedure regarding the decode processing of functional channels (LICH, SCCH, UDCH2, and FACCH3) used for data communications on traffic channel.

There are two kinds of tests in this section, one is a FACCH3 test with the frame for Status Call and the other is a UDCH2 test with the frame for Data Call.

- Connection Diagram

See Figure 6.5-1.

- Test Procedure

(1) Select appropriate SSG data used for the test.

(2) Transmit the signal selected at the step (1) from an SSG, the monitor the data sequence decoded by the unit under test with Monitor PC.

(3) For FACCH3 test, compare the monitored data sequence with the data sequence of LICH/SCCH/FACCH3 before coding as shown in Table 7.6-7 to Table 7.6-9 corresponding to the SSG data selected at the step (1).
If the two data sequences are the same, the unit under test passed this test.
The relation between the SSG data and the data sequence before coding as a comparison object is shown in Table 7.6-5.

(4) For UDCH2 test, compare the monitored data sequence with the data sequence of LICH/SCCH/UDCH2/FACCH3 before coding as shown in Table 7.6-10 to Table 7.6-12 corresponding to the SSG data selected at the step (1).
If the two data sequences are the same, the unit under test passed this test.
The relation between the SSG data and the data sequence before coding as a comparison object is shown in Table 7.6-6.

(5) Perform tests of the step (1) to the step (4) using all of the SSG data as shown in Table 7.6-5 to Table 7.6-6.

Table 7.6-7 to Table 7.6-12 show the data which combines LICH data before dibit, SCCH, UDCH2 / FACCH3 data before coding as described in Section 5.1.

Pattern	SSG data	Data sequence before coding
Pattern 1	Table 7.7-2	Table 7.6-7
Pattern 2		Table 7.6-8
Pattern 3		Table 7.6-9

Table 7.6-5 Relation between SSG data and Data Sequence before coding (Status Call)

Pattern	SSG data	Data sequence before coding
Pattern 1	Table 7.7-4	Table 7.6-10
Pattern 2		Table 7.6-11
Pattern 3		Table 7.6-12

Table 7.6-6 Relation between SSG data and Data Sequence before coding (Data Call)

No	LICH	SCCH	(FA3) FACCH3
Direction: Inbound			
1	D1	0108028	(FA3) 32008008010802000100 00000000000000000000
2	D1	1F08028	(FA3) 08008008010802000000 00000000000000000000
Direction: Outbound			
1	D3	0108028	(FA3) 32008008010802000100 00000000000000000000
2	D3	1F08028	(FA3) 08008208010802000000 00000000000000000000

Table 7.6-7 Data Sequence before coding (Status Call Pattern 1)

No	LICH	SCCH	(FA3) FACCH3
Direction: Inbound			
1	D1	010BE88	(FA3) 3200800BE70BE8000100 00000000000000000000
2	D1	1F0BE88	(FA3) 0800800BE70BE8000000 00000000000000000000
Direction: Outbound			
1	D3	010BE88	(FA3) 3200800BE70BE8000100 00000000000000000000
2	D3	1F0BE88	(FA3) 0800820BE70BE8000000 00000000000000000000

Table 7.6-8 Data Sequence before coding (Status Call Pattern 2)

No	LICH	SCCH	(FA3) FACCH3
Direction: Inbound			
1	D1	010FCF8	(FA3) 3200800FD00FCF000100 00000000000000000000
2	D1	1F0FCF8	(FA3) 0800800FD00FCF000000 00000000000000000000
Direction: Outbound			
1	D3	010FCF8	(FA3) 3200800FD00FCF000100 00000000000000000000
2	D3	1F0FCF8	(FA3) 0800820FD00FCF000000 00000000000000000000

Table 7.6-9 Data Sequence before coding (Status Call Pattern 3)

No	LICH	SCCH	(UD2) UDCH2/ (FA3) FACCH3
Direction: Inbound			
1	DD	0108028	(UD2) 09008008010802000004 00000000000000000000
2	DD	0108028	(UD2) 0B004142434445464748 494A4B4C4D4E78493A33
3	D1	1F08028	(FA3) 08008008010802000000 00000000000000000000
Direction: Outbound			
1	DF	0108028	(UD2) 09008008010802000004 00000000000000000000
2	DF	021FFC0	(UD2) 0B004142434445464748 494A4B4C4D4E78493A33
3	D3	1F08028	(FA3) 08008208010802000000 00000000000000000000

Table 7.6-10 Data Sequence before coding (Data Call, Pattern 1)

No	LICH	SCCH	(UD2) UDCH2/ (FA3) FACCH3
Direction: Inbound			
1	DD	010BE88	(UD2) 0900800BE70BE8000004 00000000000000000000
2	DD	010BE88	(UD2) 0B004142434445464748 494A4B4C4D4E78493A33
3	D1	1F0BE88	(FA3) 0800800BE70BE8000000 00000000000000000000
Direction: Outbound			
1	DF	010BE88	(UD2) 0900800BE70BE8000004 00000000000000000000
2	DF	021FFC0	(UD2) 0B004142434445464748 494A4B4C4D4E78493A33
3	D3	1F0BE88	(FA3) 0800820BE70BE8000000 00000000000000000000

Table 7.6-11 Data Sequence before coding (Data Call, Pattern 2)

No	LICH	SCCH	(UD2) UDCH2/ (FA3) FACCH3
Direction: Inbound			
1	DD	010FCF8	(UD2) 0900800FD00FCF000004 00000000000000000000
2	DD	010FCF8	(UD2) 0B004142434445464748 494A4B4C4D4E78493A33
3	D1	1F0FCF8	(FA3) 0800800FD00FCF000000 00000000000000000000
Direction: Outbound			
1	DF	010FCF8	(UD2) 0900800FD00FCF000004 00000000000000000000
2	DF	021FFC0	(UD2) 0B004142434445464748 494A4B4C4D4E78493A33
3	D3	1F0FCF8	(FA3) 0800820FD00FCF000000 00000000000000000000

Table 7.6-12 Data Sequence before coding (Data Call, Pattern 3)

7.6.2. Transmitter Test

This section tests the validity of encoding processing of a unit under test.

Tests are conducted that a unit under test transmits the frame data which is encoded and constructed from given data sequence and the bit sequence demodulated by a monitor receiver is compared with the frame data corresponding to the used data sequence. The frame data for the comparison is the same as the data sequence of the SSG data used for receiver test.

7.6.2.1. Voice Format

This section explains the testing procedure regarding the encode processing of functional channels (LICH, SCCH, VCH, and FACCH1) used for voice communication on traffic channel.

- Connection Diagram

See Figure 6.5-3.

- Test Procedure

(1) Set the unit under test to send the data sequence before coding as shown in Table 7.6-2 to Table 7.6-4.

(2) Start transmission from the unit under test, and monitor the frame data sequence received by Monitor Receiver with Monitor PC.

(3) Compare the monitored data sequence with the data sequence in Table 7.7-1 relevant to the pattern set up at the step (1).
If the two data sequences are the same, the unit under test passes this test.
The relation between the data sequence before coding and the frame data sequence which is compared with the monitored data sequence is shown in Table 7.6-13.

(4) Perform tests of the step (1) to the step (3) about all of the data sequence before coding as shown in Table 7.6-13.

Pattern	Data sequence before coding	Frame data sequence for comparison
Pattern 1	Table 7.6-2	Table 7.7-1
Pattern 2	Table 7.6-3	
Pattern 3	Table 7.6-4	

Table 7.6-13 Frame Data Sequence for Comparison

7.6.2.2. Data Format

This section explains the testing procedure regarding the encode processing of functional channels (LICH, SCCH, UDCH2, and FACCH3) used for data communications on traffic channel.

There are two kinds of test in this section, one is a FACCH3 test with the frame for Status Call and the other is a UDCH test with the frame for Data Call

- Connection Diagram

See Figure 6.5-3.

- Test Procedure

(1) Set the unit under test to send the data sequence before coding as shown in Table 7.6-7 to Table 7.6-12.
(2) Start transmission from the unit under test, and monitor the frame data sequence received by Monitor Receiver with Monitor PC.
(3) For FACCH3 Test, compare the monitored data sequence with the data sequence in Table 7.7-2 relevant to the pattern set up at the step (1).
If the two data sequences are the same, the unit under test passes this test.
The relation between the data sequence before coding and the frame data sequence which is compared with the monitored data sequence is shown in Table 7.6-14.
(4) For UDCH2 Test, compare the monitored data sequence with the data sequence in Table 7.7-4 relevant to the pattern set up at the step (1).
If the two data sequences are the same, the unit under test passes this test.
The relation between the data sequence before coding and the frame data sequence which is compared with the monitored data sequence is shown in Table 7.6-15.
(5) Perform tests of the step (1) to the step (4) about all of the data sequence before coding as shown in Table 7.6-14 and Table 7.6-15.

Pattern	Data sequence before coding	Frame data sequence for comparison
Pattern 1	Table 7.6-7	Table 7.7-2
Pattern 2	Table 7.6-8	
Pattern 3	Table 7.6-9	

Table 7.6-14 Frame Data Sequence for Comparison (Status Call)

Pattern	Data sequence before coding	Frame data sequence for comparison
Pattern 1	Table 7.6-10	Table 7.7-4
Pattern 2	Table 7.6-11	
Pattern 3	Table 7.6-12	

Table 7.6-15 Frame Data Sequence for Comparison (Data Call)

7.6.2.3. Preamble and FSW Tests

This section explains the testing procedure to verify a preamble and FSW pattern used on RTCH2.

- Connection Diagram

See Figure 6.5-5.

- Test Procedure

(4) ID for the unit under test is not defined in particular.
(5) Start transmission from the unit under test, and monitor a waveform of an FM demodulated output signal. Record the waveform included preamble and FSW.
(6) Compare the recorded waveform at the step (2) with Figure 6.6-1 and verify these waveforms are the same. Figure 6.6-1 shows an example of the waveform that contains the repetitive symbol pattern “+3, +3, -3, -3” for eight symbols at the head of a preamble. Verify the number of symbols as being set in the unit under test since this repetitive pattern length is not defined.

7.6.2.4. Post Field Test for Transmission

This section explains the testing procedure to verify the sending pattern of the post field used on RTCH2.

- Connection Diagram

See Figure 6.5-5.

- Test Procedure

(1) Test the post-field following CALL_RESP message sent from the unit under test.
(2) Start transmission from the unit under test, and monitor a waveform of an FM demodulated output signal. Record the waveform including post field.
(3) Compare the recorded waveforms at the step (2) with Figure 6.6-2 and verify these waveforms are the same.

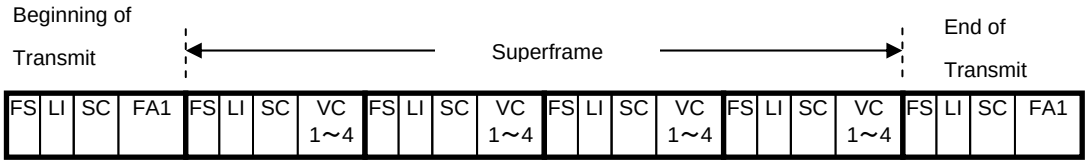
7.7. Test Frame

This section shows the SSG data sequence used for the test. Table 5.1-1 shows ID setting for the frame data. The preamble “5775FD” at the head of the frame data for RTCH2 may be added if necessary.

7.7.1. Voice Frame

This Voice Frame is a frame to be utilized for voice communications on RTCH2, and is frame data to test an LICH, SCCH, VCH and FACCH1 which are used for voice communications. Frame structure is presented in Figure 7.7-1.

Refer to Section 5.1.1 for an LICH, Section 5.1.2 for an SCCH, Section 5.1.3 for a VCH and Section 5.1.4 for an FACCH1 about the details of data strings respectively. Besides, an LICH setting shows only an inbound direction. When the LICH setting is replaced with an outbound direction, refer to the values presented in Section 5.1.1.



FS : FSW LI : LICH VCH : VCH
SC : SCCH FA1 : FACCH1

Inbound

Frame	LICH	SCCH	VCH	VCH	VCH	VCH
			FACCH1		FACCH1	
(1)	non-superframe, FACCH1	INFO4	VCALL		VCALL	
(2)	superframe, VCH	INFO1	Tone Test Pattern			
(3)	superframe, VCH	INFO2	Tone Test Pattern			
(4)	superframe, VCH	INFO3	Tone Test Pattern			
(5)	superframe, VCH	INFO4	Tone Test Pattern			
(6)	non-superframe, FACCH1	INFO4(EOT)	TX_REL		TX_REL	

Figure 7.7-1 Frame Structure for Voice Call

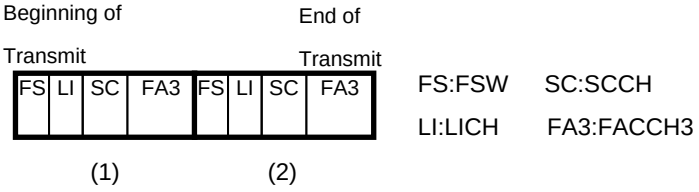
		Pattern 1	Pattern 2	Pattern 3
Frame(1)	FSW	CDF59	CDF59	CDF59
	LICH	FD7F	FD7F	FD7F
	SCCH	434CE7AE03D22A1	620A84AAE3DB662	7244F5BE24DAB48
	FACCH1	DA2462081AA08E808B 402236F83E208BE908	CAA4670A8AE86AA2A3 48A274F93F29CF0B2B	D62822290B286EA0B3 452CF6FC3C298F0D68
	FACCH1	582C4082B0A8A608A3 68223CFABC00A36B22	48AC458020E0422A8B 60A27EFBBD09E78901	542000A3A12046289B 6D2CFCFEBE09A78F42
Frame(2)	FSW	CDF59	CDF59	CDF59
	LICH	F5DD	F5DD	F5DD
	SCCH	C6F0764D6144AA1	C6F0764D6144AA1	C6F0764D6144AA1
	VCH	4CAADE8B26E4F28288	4CAADE8B26E4F28288	4CAADE8B26E4F28288
	VCH	C68A7429A4ECD00822	C68A7429A4ECD00822	C68A7429A4ECD00822
	VCH	CEA2FC018CECDA0AA0	CEA2FC018CECDA0AA0	CEA2FC018CECDA0AA0
	VCH	EE8A7E2B26CCF88A08	EE8A7E2B26CCF88A08	EE8A7E2B26CCF88A08
Frame(3)	FSW	CDF59	CDF59	CDF59
	LICH	F5DD	F5DD	F5DD
	SCCH	C74FF76E125A280	E609946AF253643	F647E57E3552B69
	VCH	4CAADE8B26E4F28288	4CAADE8B26E4F28288	4CAADE8B26E4F28288
	VCH	C68A7429A4ECD00822	C68A7429A4ECD00822	C68A7429A4ECD00822
	VCH	CEA2FC018CECDA0AA0	CEA2FC018CECDA0AA0	CEA2FC018CECDA0AA0
	VCH	EE8A7E2B26CCF88A08	EE8A7E2B26CCF88A08	EE8A7E2B26CCF88A08
Frame(4)	FSW	CDF59	CDF59	CDF59
	LICH	F5DD	F5DD	F5DD
	SCCH	43EBD6A430C6E83	6AEF95A8D0C76C7	6A63C43C16F77CD
	VCH	4CAADE8B26E4F28288	4CAADE8B26E4F28288	4CAADE8B26E4F28288
	VCH	C68A7429A4ECD00822	C68A7429A4ECD00822	C68A7429A4ECD00822
	VCH	CEA2FC018CECDA0AA0	CEA2FC018CECDA0AA0	CEA2FC018CECDA0AA0
	VCH	EE8A7E2B26CCF88A08	EE8A7E2B26CCF88A08	EE8A7E2B26CCF88A08
Frame(5)	FSW	CDF59	CDF59	CDF59
	LICH	F5DD	F5DD	F5DD
	SCCH	434CE7AE03D22A1	620A84AAE3DB662	7244F5BE24DAB48
	VCH	4CAADE8B26E4F28288	4CAADE8B26E4F28288	4CAADE8B26E4F28288
	VCH	C68A7429A4ECD00822	C68A7429A4ECD00822	C68A7429A4ECD00822
	VCH	CEA2FC018CECDA0AA0	CEA2FC018CECDA0AA0	CEA2FC018CECDA0AA0
	VCH	EE8A7E2B26CCF88A08	EE8A7E2B26CCF88A08	EE8A7E2B26CCF88A08
Frame(6)	FSW	CDF59	CDF59	CDF59
	LICH	FD7F	FD7F	FD7F
	SCCH	410CF3CD13CEEB0	604A90C9F3C7A73	7004E1DD34C6759
	FACCH1	9A8422081AA88C808A C122B6E8063889EC0A	8A04270A8AE068A2A2 C9A2F4E90731CD0E29	968862290B206CA0B2 C42C76EC04318D086A
	FACCH1	188C0082B0A0A408A2 E922BCEA8418A16E20	080C058020E8402A8A E1A2FEEB8511E58C03	148040A3A12844289A EC2C7CEE8611A58A40

Table 7.7-1 Frame Data for Voice Call

7.7.2. Data Frame

This Data Frame is a frame to be utilized for data and status communications on RTCH2, and is frame data to test an LICH, SCCH, UDCH2 and FACCH3 which are used for data and status communications. The frame structure for status communications are presented in Figure 7.7-2. The frame structure for data communications are presented in Figure 7.7-3. The frame data is for the case that the number of frames in a DCALL (User Data) message is 1 as presented in Figure 7.7-3.
Refer to Section 5.1.1 for an LICH, Section 5.1.2 for an SCCH, Section 5.1.5 for a UDCH2, and Section 5.1.6 for an FACCH3 about the details of data strings respectively. Besides, an LICH setting shows only an inbound direction. When the LICH setting is replaced with an outbound direction, refer to the values presented in Section 5.1.1.

[STATUS CALL]



Inbound

Frame	LICH	SCCH	FACCH3
(1)	FACCH3	INFO4	STAT_REQ
(2)	FACCH3	INFO4(EOT)	TX_REL

Figure 7.7-2 Frame Structure for Status Call using FACCH3

		Pattern 1	Pattern 2	Pattern 3
Frame(1)	FSW	CDF59	CDF59	CDF59
	LICH	FF7D	FF7D	FF7D
	SCCH	434CE7AE03D22A1	620A84AAE3DB662	7244F5BE24DAB48
	FACCH3	9B04A23802A08E838A C0E2F6086E2C81E70D 008A028230208228AB 212280E88A10A4A803	8B84A73A92E86AA1A2 C862B4096F25C5052E 008A028230208228AB 212280E88A10A4A803	9708E21913286EA3B2 C5EC360C6C2585036D 008A028230208228AB 212280E88A10A4A803
Frame(2)	FSW	CDF59	CDF59	CDF59
	LICH	FF7D	FF7D	FF7D
	SCCH	410CF3CD13CEEB0	604A90C9F3C7A73	7004E1DD34C6759
	FACCH3	9A8422081AA88C808A C122B6E8063889EC0A 008A028230208228AB 212280E88A10A4A803	8A04270A8AE068A2A2 C9A2F4E90731CD0E29 008A028230208228AB 212280E88A10A4A803	968862290B206CA0B2 C42C76EC04318D086A 008A028230208228AB 212280E88A10A4A803

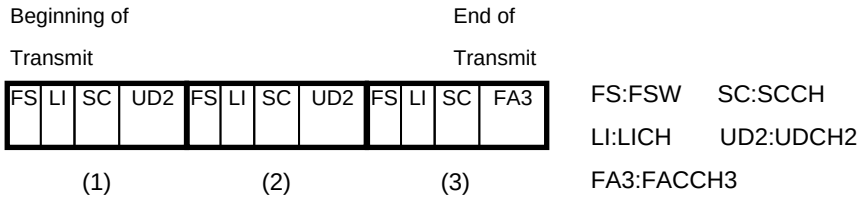
Table 7.7-2 Frame Data for Status Call Using FACCH3

[DATA CALL]

Packet Information which is used to show the composition of the transmission packet of the data communication is shown in Table 7.7-3.

Bit	7	6	5	4	3	2	1	0
Octet8	0	0	0	0	0	0	0	0
Octet9	0	0	0	0	0	1	0	0
Octet10	0	0	0	0	0	0	0	0

Table 7.7-3 Packet Information for Data Call



Inbound

Frame	LICH	SCCH	UDCH2
			FACCH3
(1)	UDCH2	INFO4	DCALL(Header),
(2)	UDCH2	INFO4	DCALL(User Data) Data = 4142434445464748494A4B4C4D4E
(3)	FACCH3	INFO4(EOT)	TX_REL

Figure 7.7-3 Frame Structure for Data Call

		Pattern 1	Pattern 2	Pattern 3
Frame(1)	FSW	CDF59	CDF59	CDF59
	LICH	FFDD	FFDD	FFDD
	SCCH	434CE7AE03D22A1	620A84AAE3DB662	7244F5BE24DAB48
	UDCH2	DA2422282AB08C8288 C12216D8262087ED08 008A028230208228AB 212280E88A10A4A803	CAA4272ABAF868A0A0 C9A254D92729C30F2B 008A028230208228AB 212280E88A10A4A803	D62862093B386CA2B0 C42CD6DC2429830968 008A028230208228AB 212280E88A10A4A803
Frame(2)	FSW	CDF59	CDF59	CDF59
	LICH	FFDD	FFDD	FFDD
	SCCH	434CE7AE03D22A1	620A84AAE3DB662	7244F5BE24DAB48
	UDCH2	E9232F0EFA613CFEFA B54B574E4207F706E9 4FDA620A92F67B132C 51A590F000A6F46BE8	E9232F0EFA613CFEFA B54B574E4207F706E9 4FDA620A92F67B132C 51A590F000A6F46BE8	E9232F0EFA613CFEFA B54B574E4207F706E9 4FDA620A92F67B132C 51A590F000A6F46BE8
Frame(3)	FSW	CDF59	CDF59	CDF59
	LICH	FF7D	FF7D	FF7D
	SCCH	410CF3CD13CEEB0	604A90C9F3C7A73	7004E1DD34C6759
	FACCH3	9A8422081AA88C808A C122B6E8063889EC0A 008A028230208228AB 212280E88A10A4A803	8A04270A8AE068A2A2 C9A2F4E90731CD0E29 008A028230208228AB 212280E88A10A4A803	968862290B206CA0B2 C42C76EC04318D086A 008A028230208228AB 212280E88A10A4A803

Table 7.7-4 Frame Data for Data Call

7.8. Conditions of SSG

See Section 6.8.

But Symbol Rate is only 4800bps = 2.4ksps.

7.9. Conditions of Monitor Receiver

See Section 6.9.

8. Revision History

Version	Date	Revised Contents
1.0	Nov 6 2008	Version 1.0 release
1.1	Mar 11 2010	Section 2: update version number of Reference documents, add Section 4: add relation list, add RTCH setting in Section 4.1.1, update test parameters in Section 4.1.5, modify bit sequence of STAT_REQ, add item of superframe in Section 4.4.3 and 4.6.3, Section 5: add the test method using frame structure during communication, update test frame data
1.2	Nov 2 2012	Section 5 Appndix moves to Section 6 Appendix1 (Conventional, Type-C), add Section 5 Common Air Interface Tests (Type-D) and Section 7 Appendix 2 (Type-D).