

## Annex 92A (informative)

### FEC frame encoding example

This Annex provides an example of FEC frame encoding with RS (255,223) code. See 92.2.3.4 for the format of the FEC frame.

#### 92A.1 66b Block Input

The Table below provides an example of a 64b/66b block stream received at the input to the RS (255,223) encoder. The example shows a stream of 27 scrambled 64b/66b blocks generated from the output of PCS layer when the link was sending out IDLEs.

The 66b blocks in the table are transmitted from left to right within each row and from top to bottom between rows. The 64 bit payload portion of the 66b block is described as a series of hexadecimal octets - the leftmost octet of each payload portion is transmitted first. Bits within each octet of the payload are transmitted in least-significant-bit-first order (ie. the rightmost bit of each octet is transmitted first).

Thus, the first ten bits transmitted will be: 10 0100 0000 ....

Sync [0:1]	64bit payload [7:0]- [15:8]- [23:16]- [31:24]- [39:32]- [47:40]- [55:48]- [63:56]	Sync [0:1]	64bit payload [7:0]- [15:8]- [23:16]- [31:24]- [39:32]- [47:40]- [55:48]- [63:56]	Sync [0:1]	64bit payload [7:0]- [15:8]- [23:16]- [31:24]- [39:32]- [47:40]- [55:48]- [63:56]	Sync [0:1]	64bit payload [7:0]- [15:8]- [23:16]- [31:24]- [39:32]- [47:40]- [55:48]- [63:56]
10	02-57-78-ee-77-cb-80-37	10	b5-5a-dc-1f-b6-59-f3-3a	10	7b-aa-d3-a1-fb-f0-3e-05	10	67-33-ff-71-41-48-8d-63
10	6b-dc-63-c3-90-00-60-1c	10	0e-c7-0d-73-0c-07-92-be	10	3b-b1-cf-78-c3-d5-22-89	10	66-df-89-9c-13-38-cb-de
10	ad-2e-ee-2b-0f-7a-6c-40	10	31-bf-92-0a-48-29-5e-8c	10	e7-ee-3e-0f-63-0b-46-01	10	22-4a-39-2f-2d-09-a0-14
10	a1-73-b8-e4-ae-50-6b-d8	10	a2-b6-3a-8e-2e-fc-3a-96	10	83-fd-46-a7-3b-2a-26-ad	10	3b-06-88-7d-7e-85-b7-2a
10	38-9f-34-a2-00-42-e5-fa	10	33-d2-29-70-f5-8c-02-db	10	ee-dd-86-54-5e-fd-02-f8	10	43-b4-2c-78-09-2a-ba-19
10	73-b6-f5-f8-24-d1-bd-b6	10	bb-44-0b-cd-9f-aa-78-6b	10	ea-62-61-c3-9f-97-1c-19	10	74-4a-46-f1-52-48-41-73
10	4f-30-61-eb-98-22-55-8d	10	aa-c8-3c-c9-cc-01-51-34	10	58-15-a4-1b-1d-e8-db-b2		

#### 92A.2 66b Block Input in Binary format

Sync [0:1]	64bit payload (transmitted from left to right)
10	0100000011101010000111100111011111101110110100110000000111101100
10	101011010101101000111011111100001101101100110101100111101011100
10	1101111001010101110010111000010111011111000011110111110010100000
10	11100110110011001111111100011101000001000100101011000111000110
10	110101100011101111000110110000110000100100000000000011000111000
10	0111000011100011101100001100111000110000111000000100100101111101
10	1101110010001101111100110001111011000011101010110100010010010001
10	0110011011111011100100010011100111001000000111001101001101111011
10	101101010111010001110111110101001111000010111100011011000000010
10	1000110011111101010010010101000000010010100101000111101000110001
10	111001110111011101111100111100001100011011010000011000101000000



	<i>D<sub>n</sub></i>	<i>D<sub>n-1</sub></i>	<i>D<sub>n-2</sub></i>	<i>D<sub>n-3</sub></i>	<i>D<sub>n-4</sub></i>	<i>D<sub>n-5</sub></i>	<i>D<sub>n-6</sub></i>	<i>D<sub>n-7</sub></i>	<i>D<sub>n-8</sub></i>	<i>D<sub>n-9</sub></i>	<i>D<sub>n-10</sub></i>	<i>D<sub>n-11</sub></i>	<i>D<sub>n-12</sub></i>	<i>D<sub>n-13</sub></i>	<i>D<sub>n-14</sub></i>	<i>D<sub>n-15</sub></i>
<b>n=222</b>	00	00	00	80	c0	15	9e	fb	dd	32	e0	8d	5a	2d	ee	0f
<b>N=206</b>	Db	ac	79	1d	7b	aa	d3	a1	fb	f0	3e	05	ce	66	fe	e3
<b>n=190</b>	82	90	1a	c7	ac	71	8f	0d	43	02	80	71	70	38	6e	98
<b>n=174</b>	63	38	90	f4	b5	13	fb	8c	37	5c	2d	92	c8	ec	3b	91
<b>n=158</b>	73	02	67	d9	5b	ab	8b	fb	ca	83	1e	1b	90	98	5f	49
<b>n=142</b>	05	a4	14	2f	46	e7	ee	3e	0f	63	0b	46	01	44	94	72
<b>n=126</b>	5e	5a	12	40	29	84	ce	e1	92	bb	42	ad	61	13	b5	D5
<b>n=110</b>	71	74	e1	d7	b1	34	d8	6f	74	ba	a3	62	d2	6a	c7	00
<b>n=94</b>	b1	cf	af	f0	56	05	ce	27	8d	28	80	50	b9	be	19	e9
<b>n=78</b>	14	b8	7a	46	81	6d	ee	dd	86	54	5e	fd	02	f8	86	68
<b>n=62</b>	59	f0	12	54	74	33	cc	d9	d6	e3	93	44	f7	da	da	25
<b>n=46</b>	5a	68	fe	54	c5	5b	a3	2e	16	36	fc	79	c9	91	81	4e
<b>n=30</b>	c9	28	5e	0a	29	68	ce	13	4c	d8	3a	a6	48	55	23	55
<b>n=14</b>	64	9e	64	e6	80	28	1a	58	15	a4	1b	1d	e8	db	b2	

### 92A.5 Parity Symbol Output

The Table below illustrates the 32 parity octets computed by the RS(255, 223) encoder for the inputs given above.

Note that in figure 92-8 the rightmost bit of each parity octet is the most significant, whereas table tbd-3 lists the octets in the more typical notation ie. the least significant bit is on the right.

	<i>P<sub>n</sub></i>	<i>P<sub>n-1</sub></i>	<i>P<sub>n-2</sub></i>	<i>P<sub>n-3</sub></i>	<i>P<sub>n-4</sub></i>	<i>P<sub>n-5</sub></i>	<i>P<sub>n-6</sub></i>	<i>P<sub>n-7</sub></i>	<i>P<sub>n-8</sub></i>	<i>P<sub>n-9</sub></i>	<i>P<sub>n-10</sub></i>	<i>P<sub>n-11</sub></i>	<i>P<sub>n-12</sub></i>	<i>P<sub>n-13</sub></i>	<i>P<sub>n-14</sub></i>	<i>P<sub>n-15</sub></i>
<b>n=31</b>	7E	62	35	FB	DB	9F	5E	8E	FD	B2	81	3E	F9	1D	9B	1A
<b>n=15</b>	32	1E	70	CF	DD	C2	2C	54	43	F1	00	78	3C	4F	BD	F4

### 92A.6 Parity Symbols in Binary Format

As with the input buffer, this is written with least significant bit leftmost to correspond with figure 92-8:

```
01111110010001101010110011011111101101111110010111101001110001
1011111101001101100000010111110010011111101110001101100101011000
```

0100110001111000000011101111001110111011010000110011010000101010  
 1100001010001111000000000001111000111100111100101011110100101111

### 92A.7 66b Parity Blocks for Transmit

The Table below illustrates the 64b/66b blocks carrying parity that are generated by the RS (255, 223) encoder for the input blocks in table tbd.1 above. The RS (255, 223) encoder inserts the parity blocks into the transmission stream to the gearbox subsequent to its transmission of the corresponding input 66b blocks (as described in 92.2.3.4).

The 66b blocks in the table are transmitted from left to right within each row and from top to bottom between rows. The 64 bit payload portion of the 66b block is described as a series of hexadecimal octets - the leftmost octet of each payload portion is transmitted first. Bits within each octet of the payload are transmitted in least-significant-bit-first order (ie. the rightmost bit of each octet is transmitted first).

Thus, the first 18 bits of the parity blocks transmitted will be: 00 0111 1110 0100 0110 ...

Sync [0:1]	64bit payload [7:0]- [15:8]- [23:16]- [31:24]- [39:32]- [47:40]- [55:48]- [63:56]	Sync [0:1]	64bit payload [7:0]- [15:8]- [23:16]- [31:24]- [39:32]- [47:40]- [55:48]- [63:56]	Sync [0:1]	64bit payload [7:0]- [15:8]- [23:16]- [31:24]- [39:32]- [47:40]- [55:48]- [63:56]	Sync [0:1]	64bit payload [7:0]- [15:8]- [23:16]- [31:24]- [39:32]- [47:40]- [55:48]- [63:56]
00	7E-62-35-FB-DB-9F-5E-8E	11	FD-B2-81-3E-F9-1D-9B-1A	11	32-1E-70-CF-DD-C2-2C-54	00	43-F1-00-78-3C-4F-BD-F4

### 92A.8 Parity 66b blocks in Binary Format

00	011111100100011010101100110111111101101111110010111101001110001
11	1011111101001101100000010111110010011111101110001101100101011000
11	0100110001111000000011101111001110111011010000110011010000101010
00	1100001010001111000000000001111000111100111100101011110100101111