一种数据处理方法和传输装置

本申请要求于2024年03月08日提交国家知识产权局、申请号为202410269781.8、申请名称为“一种数据处理方法和传输装置”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请涉及通信技术领域，尤其涉及一种数据处理方法和传输装置。

背景技术

在5G、云计算、大数据、人工智能等持续推动下，高速光传输网络正朝着大容量、分组化、智能化的方向发展。相干光通信系统利用光波的幅度，相位，偏振和频率来承载信息。为了对抗在传输过程中因色散、偏振相关损伤、噪声、非线性效应及其它因素引起的光信号失真并保持长距离传输，相干光通信系统通常需要采用高效的前向纠错（Forward Error Correction，FEC）码来对抗光传输过程中的光学损伤，以保持长距离传输时其误比特率足够低。如当前400ZR+和800ZR采纳的开放FEC码（OpenFEC），简称为OFEC码，其冗余（overhead，OH）为15.3%，当采用软判决译码，性能约为纠前2.0E-2。

为了提高频谱效率，通常采用多电平的正交振幅调制（Quadrature Amplitude Modulation，QAM），如16QAM、32QAM、64QAM甚至更高阶QAM。传统的QAM调制对应的信号星座图上的每个星座点（constellation points）都是以相同的概率出现。概率星座整形（Probabilistic Constellation Shaping，PCS）处理技术，在保持星座点位置不变下改变星座点出现的概率，使其非均匀分布，从而提升系统传输性能。PCS处理技术作为一种调制格式优化技术，具有逼近香农极限、灵活可变等优点，已被广泛研究与应用。

对于现有采用OFEC编码的400G长途传输方案，通常采用正交相移键控（Quadrature phase shift keying，QPSK）调制，需要采用支撑120GBaud~140GBaud波特率代际的器件。例如，在灵活光传送网（Flexible Optical Transport Network，FlexO）传输400G比特每秒（简称为FlexO-4）的光传输网（optical transport network，OTN）数据传输下，对应波特率约为124.102GBaud；在载荷为以太网业务的FlexO-4e数据传输下，对应波特率约为118.203GBaud。此时，其信道间隔（channel spacing，也称为格点间隔，grid spacing）通常采用150GHz。然而，现有网络中采用的400G数据中心、城域相干传输方案，所采用的信道间隔约为75GHz或100GHz或112.5GHz。也就是说，采用OFEC编码和QPSK调制的400G长途传输方案较难与现有400G数据中心、城域网络兼容，是亟需解决的问题。

发明内容

本申请实施例提供了一种数据处理方法、装置和系统，将采用FEC编码并结合PCS技术以满足更长传输距离，并使得整体数据处理操作更简单，复杂度更低，功耗更低；并在保证整体性能同时，可与现有400G数据中心、城域网络兼容。

第一方面，本申请提供一种数据处理方法，该数据处理方法包括如下步骤：根据比特数量为d_{in}的第一比特数据，获取第二比特数据，其中，所述第二比特数据包括d_{in}个第一比特数据中的比特，d_{CRC}个校验比特和/或d_{PAD}个填充比特，d_{CRC}为大于或等于0的整数，d_{PAD}为大于或等于0的整数；对所述第二比特数据进行加扰得到第三比特数据，所述第三比特数据的比特数量是$d_{scr} = d_{in} + d_{CP}$，$d_{CP} = d_{CRC} + d_{PAD}$；对比特数量为m的第一比特集合中的第一比特子集进行概率星座整形PCS处理得到第二比特子集，其中，所述第一比特集合是从所述第三比特数据中获取的，m为大于1的整数，d_{scr}是m的整数倍；对所述第二比特子集和所述第一比特集合中除所述第一比特子集外的第三比特子集进行前向纠错FEC编码和交织处理得到第二比特集合，其中，d_{in}，m和d_{CP}的值为下表中的任意一组：

d_{in}	m	d_{CP}
1186312	3536	1784
1184256	3536	3840
1182200	3520	520
1180144	3520	2576
1176032	3504	1312
1173976	3504	3368

1171920	3488	48
1169864	3488	2104
1165752	3472	840
1163696	3472	2896
1159584	3456	1632
1157528	3456	3688
1155472	3440	368
1153416	3440	2424
1149304	3424	1160
1147248	3424	3216
1143136	3408	1952
1141080	3408	4008
1139024	3392	688
1136968	3392	2744
1132856	3376	1480
1130800	3376	3536
1128744	3360	216
1122576	3344	1008
1120520	3344	3064
1116408	3328	1800
1114352	3328	3856
1112296	3312	536
1110240	3312	2592
1106128	3296	1328
1104072	3296	3384
1102016	3280	64
1099960	3280	2120
1095848	3264	856
1093792	3264	2912
1089680	3248	1648
1087624	3248	3704
1085568	3232	384
1083512	3232	2440
1079400	3216	1176
1077344	3216	3232
1071176	3200	4024
1069120	3184	704
1067064	3184	2760
1062952	3168	1496
1060896	3168	3552
1058840	3152	232
1056784	3152	2288
1052672	3136	1024
1050616	3136	3080
1046504	3120	1816
1044448	3120	3872
1042392	3104	552
1040336	3104	2608
1036224	3088	1344
1034168	3088	3400
1032112	3072	80
1030056	3072	2136
1025944	3056	872
1023888	3056	2928
1019776	3040	1664
1017720	3040	3720
1015664	3024	400
1013608	3024	2456

1009496	3008	1192
1007440	3008	3248
1003328	2992	1984
1001272	2992	4040
999216	2976	720
997160	2976	2776
993048	2960	1512
990992	2960	3568
988936	2944	248
986880	2944	2304
982768	2928	1040
980712	2928	3096
976600	2912	1832
974544	2912	3888
972488	2896	568
970432	2896	2624
966320	2880	1360
964264	2880	3416
962208	2864	96
960152	2864	2152
956040	2848	888
953984	2848	2944
949872	2832	1680
947816	2832	3736
945760	2816	416
943704	2816	2472
939592	2800	1208
937536	2800	3264
933424	2784	2000
931368	2784	4056
929312	2768	736
927256	2768	2792
923144	2752	1528
921088	2752	3584
919032	2736	264
916976	2736	2320
912864	2720	1056
910808	2720	3112
906696	2704	1848
904640	2704	3904
902584	2688	584
900528	2688	2640
896416	2672	1376
894360	2672	3432
892304	2656	112
886136	2640	904
884080	2640	2960
879968	2624	1696
877912	2624	3752
875856	2608	432
873800	2608	2488
869688	2592	1224
867632	2592	3280
863520	2576	2016
861464	2576	4072
859408	2560	752
857352	2560	2808
853240	2544	1544

851184	2544	3600
849128	2528	280
847072	2528	2336
842960	2512	1072
840904	2512	3128
836792	2496	1864
834736	2496	3920
832680	2480	600
830624	2480	2656
826512	2464	1392
824456	2464	3448
822400	2448	128
820344	2448	2184
816232	2432	920
814176	2432	2976
810064	2416	1712
808008	2416	3768
805952	2400	448
803896	2400	2504
799784	2384	1240
797728	2384	3296
793616	2368	2032
791560	2368	4088
789504	2352	768
787448	2352	2824
783336	2336	1560
781280	2336	3616
779224	2320	296
777168	2320	2352
773056	2304	1088
771000	2304	3144
766888	2288	1880
764832	2288	3936
762776	2272	616
760720	2272	2672
756608	2256	1408
754552	2256	3464
752496	2240	144
750440	2240	2200
746328	2224	936
744272	2224	2992
740160	2208	1728
738104	2208	3784
736048	2192	464
733992	2192	2520
729880	2176	1256
727824	2176	3312
723712	2160	2048
721656	2160	4104
719600	2144	784

。

本实施例中，对于未来城域电信传输和城域 DCI 互联场景，将采用 FEC 编码并结合 PCS 技术以满足更长传输距离。PCS 处理的引入，使得进行“PCS 处理和 FEC 编码”（即说明书中的第一子处理）的数据处理颗粒度不再是 3552，而是更小的数值m，此时需要重新设计从数据帧获取数据的行数，进行 CRC 校验和插入填充比特的数量，以使得整体数据处理操作更简单，复杂度更低，功耗更低。而且，在第一比特数据为 FlexO-4e 或 400G-ZR 帧数据的情况下，对应的波特率均为小于 100GBaud，可较好兼容现有

400G 数据中心、城域网络的方案。

在一些可能的实施方式中，$d_{scr}=336\times m$。本申请实施例中，比特数目为d_{in}的第一比特数据经过数据处理（包括说明书中的“第一数据处理”和“第二数据处理”）之后，刚好得到一个 DSP 帧，此时，可以得到$d_{scr}=d_{in}+d_{CP}=L\times m=336\times m$，使得所提出的数据处理方法具有较低实现复杂度，功耗更低。

在一些可能的实施方式中，所述第一比特子集的比特数量是 16 或 32 的整数倍，所述第二比特子集包括 2048 个比特，所述第三比特子集包括 1504 比特，所述第二比特集合包括 4096 个比特。

在一些可能的实施方式中，所述第一比特数据包括 r 行 q 列个比特，d_{in}=r×q，r 为大于 1 的整数，q=2056 或 10280。

在一些可能的实施方式中，所述根据比特数量为d_{in}的第一比特数据，获取第二比特数据，具体包括：对所述第一比特数据进行循环冗余校验 CRC 和/或插入填充比特得到第二比特数据；其中，对所述第一比特数据进行 CRC 包括：对所述第一比特数据中每 r/p 行总共 r/p×q 个比特进行 CRC-32 校验，添加长度为 32 比特的校验比特，$d_{CRC}=32\times p$，r 能被p整除，p 为大于 1 的整数；

或者，对所述第一比特数据中前$r_0\times(p-1)$行中的每r_0行共$q\times r_0$个比特进行 CRC-32 校验以添加长度为 32 比特的校验比特，并对所述第一数据中最后r_1行共$q\times r_1$个比特进行 CRC-32 校验以添加长度为 32 比特的校验比特，其中，$d_{CRC}=32\times p$，r 不能被p整除，p 为大于 1 的整数，$r_0\times(p-1)+r_1=r$，整数r_0大于整数r_1。

在一些可能的实施方式中，q=2056，r，m，d_{CP}，p 和 r_1的值为下表中的任意一组：

r	m	d_{CP}	p	r_1
577	3536	1784	29	17
576	3536	3840	29	16
574	3520	2576	29	14
572	3504	1312	29	12
571	3504	3368	29	11
569	3488	2104	29	9
566	3472	2896	29	6
564	3456	1632	29	4
563	3456	3688	29	3
561	3440	2424	29	1
559	3424	1160	28	19
558	3424	3216	28	18
556	3408	1952	28	16
555	3408	4008	28	15
553	3392	2744	28	13
551	3376	1480	28	11
550	3376	3536	28	10
546	3344	1008	28	6
545	3344	3064	28	5
543	3328	1800	28	3
542	3328	3856	28	2
540	3312	2592	27	
538	3296	1328	27	18
537	3296	3384	27	17
535	3280	2120	27	15
532	3264	2912	27	12
530	3248	1648	27	10
529	3248	3704	27	9
527	3232	2440	27	7
525	3216	1176	27	5

524	3216	3232	27	4
521	3200	4024	27	1
519	3184	2760	26	19
517	3168	1496	26	17
516	3168	3552	26	16
514	3152	2288	26	14
512	3136	1024	26	12
511	3136	3080	26	11
509	3120	1816	26	9
508	3120	3872	26	8
506	3104	2608	26	6
504	3088	1344	26	4
503	3088	3400	26	3
501	3072	2136	26	1
499	3056	872	25	19
498	3056	2928	25	18
496	3040	1664	25	16
495	3040	3720	25	15
493	3024	2456	25	13
491	3008	1192	25	11
490	3008	3248	25	10
488	2992	1984	25	8
487	2992	4040	25	7
485	2976	2776	25	5
483	2960	1512	25	3
482	2960	3568	25	2
480	2944	2304	24	
478	2928	1040	24	18
477	2928	3096	24	17
475	2912	1832	24	15
474	2912	3888	24	14
472	2896	2624	24	12
470	2880	1360	24	10
469	2880	3416	24	9
467	2864	2152	24	7
465	2848	888	24	5
464	2848	2944	24	4
462	2832	1680	24	2
461	2832	3736	24	1
459	2816	2472	23	19
457	2800	1208	23	17
456	2800	3264	23	16
454	2784	2000	23	14
453	2784	4056	23	13
452	2768	736	23	12
451	2768	2792	23	11
449	2752	1528	23	9
448	2752	3584	23	8

446	2736	2320	23	6
444	2720	1056	23	4
443	2720	3112	23	3
441	2704	1848	23	1
440	2704	3904	22	
438	2688	2640	22	18
436	2672	1376	22	16
435	2672	3432	22	15
431	2640	904	22	11
430	2640	2960	22	10
428	2624	1696	22	8
427	2624	3752	22	7
425	2608	2488	22	5
423	2592	1224	22	3
422	2592	3280	22	2
420	2576	2016	21	
419	2576	4072	21	19
418	2560	752	21	18
417	2560	2808	21	17
415	2544	1544	21	15
414	2544	3600	21	14
412	2528	2336	21	12
410	2512	1072	21	10
409	2512	3128	21	9
407	2496	1864	21	7
406	2496	3920	21	6
404	2480	2656	21	4
402	2464	1392	21	2
401	2464	3448	21	1
399	2448	2184	20	19
397	2432	920	20	17
396	2432	2976	20	16
394	2416	1712	20	14
393	2416	3768	20	13
391	2400	2504	20	11
389	2384	1240	20	9
388	2384	3296	20	8
386	2368	2032	20	6
385	2368	4088	20	5
384	2352	768	20	4
383	2352	2824	20	3
381	2336	1560	20	1
380	2336	3616	19	
378	2320	2352	19	18
376	2304	1088	19	16
375	2304	3144	19	15
373	2288	1880	19	13
372	2288	3936	19	12

371	2272	616	19	11
370	2272	2672	19	10
368	2256	1408	19	8
367	2256	3464	19	7
365	2240	2200	19	5
363	2224	936	19	3
362	2224	2992	19	2
360	2208	1728	18	
359	2208	3784	18	19
357	2192	2520	18	17
355	2176	1256	18	15
354	2176	3312	18	14
352	2160	2048	18	12
351	2160	4104	18	11
350	2144	784	18	10

。

本实施例中提供的数据处理方法需要执行 CRC 校验，可以对数据检错，并复用现有 800G-ZR 的 CRC 技术，实现简单。进一步地，在第一比特数据为 FlexO-4e 或 400G-ZR 帧数据的情况下，可保证对应的波特率均为小于 100GBaud，可较好兼容现有 400G 数据中心、城域网络的方案。

在一些可能的实施方式中，m 为 32 的整数倍，q=2056；r，m，d_{CP}，p 和 r_1的值为下表中的任意一组：

r	m	d_{CP}	p	r_1
574	3520	2576	29	14
569	3488	2104	29	9
564	3456	1632	29	4
563	3456	3688	29	3
559	3424	1160	28	19
558	3424	3216	28	18
553	3392	2744	28	13
543	3328	1800	28	3
542	3328	3856	28	2
538	3296	1328	27	18
537	3296	3384	27	17
532	3264	2912	27	12
527	3232	2440	27	7
521	3200	4024	27	1
517	3168	1496	26	17
516	3168	3552	26	16
512	3136	1024	26	12
511	3136	3080	26	11
506	3104	2608	26	6
501	3072	2136	26	1
496	3040	1664	25	16
495	3040	3720	25	15
491	3008	1192	25	11
490	3008	3248	25	10
485	2976	2776	25	5
480	2944	2304	24	
475	2912	1832	24	15

474	2912	3888	24	14
470	2880	1360	24	10
469	2880	3416	24	9
465	2848	888	24	5
464	2848	2944	24	4
459	2816	2472	23	19
454	2784	2000	23	14
453	2784	4056	23	13
449	2752	1528	23	9
448	2752	3584	23	8
444	2720	1056	23	4
443	2720	3112	23	3
438	2688	2640	22	18
428	2624	1696	22	8
427	2624	3752	22	7
423	2592	1224	22	3
422	2592	3280	22	2
418	2560	752	21	18
417	2560	2808	21	17
412	2528	2336	21	12
407	2496	1864	21	7
406	2496	3920	21	6
402	2464	1392	21	2
401	2464	3448	21	1
397	2432	920	20	17
396	2432	2976	20	16
391	2400	2504	20	11
386	2368	2032	20	6
385	2368	4088	20	5
381	2336	1560	20	1
380	2336	3616	19	
376	2304	1088	19	16
375	2304	3144	19	15
371	2272	616	19	11
370	2272	2672	19	10
365	2240	2200	19	5
360	2208	1728	18	
359	2208	3784	18	19
355	2176	1256	18	15
354	2176	3312	18	14
350	2144	784	18	10

。

本申请实施例提供的参数，可以保证 FEC 性能的基础上，硬件实现简单，复杂度更低。

在一些可能的实施方式中，q=10280，r，m，d_{CP}，p 和 r_1 的值为下表中的任意一组：

r	m	d_{CP}	p	r_1
113	3472	4952	29	1
112	3440	4480	28	
111	3408	4008	28	3

110	3376	3536	28	2
109	3344	3064	28	1
108	3312	2592	27	
107	3280	2120	27	3
106	3248	1648	27	2
105	3216	1176	27	1
102	3136	5136	26	2
101	3104	4664	26	1
100	3072	4192	25	
99	3040	3720	25	3
98	3008	3248	25	2
97	2976	2776	25	1
96	2944	2304	24	
95	2912	1832	24	3
94	2880	1360	24	2
93	2848	888	24	1
90	2768	4848	23	2
89	2736	4376	23	1
88	2704	3904	22	
87	2672	3432	22	3
86	2640	2960	22	2
85	2608	2488	22	1
84	2576	2016	21	
83	2544	1544	21	3
82	2512	1072	21	2
79	2432	5032	20	3
78	2400	4560	20	2
77	2368	4088	20	1
76	2336	3616	19	
75	2304	3144	19	3
74	2272	2672	19	2
73	2240	2200	19	1
72	2208	1728	18	
71	2176	1256	18	3
70	2144	784	18	2

。

本实施例中提供的数据处理方法需要执行 CRC 校验，可以对数据检错，并复用现有 800G-ZR 的 CRC 技术，实现简单。进一步地，在第一比特数据为 FlexO-4e 或 400G-ZR 帧数据的情况下，可保证对应的波特率均为小于 100GBaud，可较好兼容现有 400G 数据中心、城域网络的方案。

在一些可能的实施方式中，m 为 32 的整数倍，q=10280，r，m，d_{CP}，p 和 r_1的值为下表中的任意一组：

r	m	d_{CP}	p	r_1
102	3136	5136	26	2
101	3104	4664	26	1
100	3072	4192	25	
99	3040	3720	25	3
98	3008	3248	25	2
97	2976	2776	25	1

96	2944	2304	24	
95	2912	1832	24	3
94	2880	1360	24	2
93	2848	888	24	1
79	2432	5032	20	3
78	2400	4560	20	2
77	2368	4088	20	1
76	2336	3616	19	
75	2304	3144	19	3
74	2272	2672	19	2
73	2240	2200	19	1
72	2208	1728	18	
71	2176	1256	18	3
70	2144	784	18	2

。

本申请实施例提供的参数，可以保证 FEC 性能的基础上，硬件实现简单，复杂度更低。

在一些可能的实施方式中，所述第三比特子集包括 1504 比特，所述第一比特子集包括 m-1504 比特；所述对比特数量为 m 的第一比特集合中的第一比特子集进行概率星座整形 PCS 处理得到第二比特子集，包括：将所述第一比特子集分成 16 组序列，每组序列包括(m-1504)/16 个比特；分别对所述 16 组序列进行 PCS 子处理，得到与所述 16 组序列对应的 16 组输出序列，其中，每组输出序列包括 128 比特，所述 16 组输出序列构成所述第二比特子集。

在一些可能的实施方式中，所述第三比特子集包括 1504 比特，所述第一比特子集包括 m-1504 比特；所述对比特数量为 m 的第一比特集合中的第一比特子集进行概率星座整形 PCS 处理得到第二比特子集，包括：将所述第一比特子集分成 32 组序列，每组序列包括(m-1504)/32 个比特；分别对所述 32 组序列进行 PCS 子处理，得到与所述 32 组序列对应的 32 组输出序列，其中，每组输出序列包括 64 比特，所述 32 组输出序列构成所述第二比特子集。

在一些可能的实施方式中，在得到第二比特集合之后，所述方法还包括：对第四比特数据进行符号映射和偏振划分得到双偏振符号序列，其中，所述符号映射和所述偏振划分将 8 个比特映射为 1 个双偏振 DP-16QAM 符号，所述第四比特数据包括多个所述第二比特集合；对所述双偏振符号序列进行数字信号处理 DSP 成帧，得到 DSP 帧(也称为超帧)。

在一些可能的实施方式中，对所述双偏振符号序列进行 DSP 成帧包括：将每 172032 个双偏振 DP-16QAM 符号进行 DSP 成帧处理得到一个 DSP 帧。

第二方面，本申请提供一种数据处理装置，包括：第一处理单元、加扰单元、概率星座整形 PCS 单元和前向纠错 FEC 编码和交织单元；所述第一处理单元用于：根据比特数量为d_{in}的第一比特数据，获取第二比特数据，其中，所述第二比特数据包括d_{in}个第一比特数据中的比特，d_{CRC}个校验比特和/或d_{PAD}个填充比特，d_{CRC}为大于或等于 0 的整数，d_{PAD}为大于或等于 0 的整数；加扰单元用于：对所述第二比特数据进行加扰得到第三比特数据，所述第三比特数据的比特数量是$d_{scr} = r \times q + d_{CP}$，$d_{CP} = d_{CRC} + d_{PAD}$；所述 PCS 单元用于：对比特数量为 m 的第一比特集合中的第一比特子集进行概率星座整形 PCS 处理得到第二比特子集，其中，所述第一比特集合是从所述第三比特数据中获取的，m 为大于 1 的整数，d_{scr}是 m 的整数倍；所述 FEC 编码和交织单元用于：对所述第二比特子集和所述第一比特集合中除所述第一比特子集外的第三比特子集进行 FEC 编码和交织处理得到第二比特集合，其中，d_{in}，m 和d_{CP}的值为下表中的任意一组：

d_{in}	m	d_{CP}
1186312	3536	1784
1184256	3536	3840
1182200	3520	520
1180144	3520	2576

1176032	3504	1312
1173976	3504	3368
1171920	3488	48
1169864	3488	2104
1165752	3472	840
1163696	3472	2896
1159584	3456	1632
1157528	3456	3688
1155472	3440	368
1153416	3440	2424
1149304	3424	1160
1147248	3424	3216
1143136	3408	1952
1141080	3408	4008
1139024	3392	688
1136968	3392	2744
1132856	3376	1480
1130800	3376	3536
1128744	3360	216
1122576	3344	1008
1120520	3344	3064
1116408	3328	1800
1114352	3328	3856
1112296	3312	536
1110240	3312	2592
1106128	3296	1328
1104072	3296	3384
1102016	3280	64
1099960	3280	2120
1095848	3264	856
1093792	3264	2912
1089680	3248	1648
1087624	3248	3704
1085568	3232	384
1083512	3232	2440
1079400	3216	1176
1077344	3216	3232
1071176	3200	4024
1069120	3184	704
1067064	3184	2760
1062952	3168	1496
1060896	3168	3552
1058840	3152	232
1056784	3152	2288
1052672	3136	1024
1050616	3136	3080
1046504	3120	1816
1044448	3120	3872

1042392	3104	552
1040336	3104	2608
1036224	3088	1344
1034168	3088	3400
1032112	3072	80
1030056	3072	2136
1025944	3056	872
1023888	3056	2928
1019776	3040	1664
1017720	3040	3720
1015664	3024	400
1013608	3024	2456
1009496	3008	1192
1007440	3008	3248
1003328	2992	1984
1001272	2992	4040
999216	2976	720
997160	2976	2776
993048	2960	1512
990992	2960	3568
988936	2944	248
986880	2944	2304
982768	2928	1040
980712	2928	3096
976600	2912	1832
974544	2912	3888
972488	2896	568
970432	2896	2624
966320	2880	1360
964264	2880	3416
962208	2864	96
960152	2864	2152
956040	2848	888
953984	2848	2944
949872	2832	1680
947816	2832	3736
945760	2816	416
943704	2816	2472
939592	2800	1208
937536	2800	3264
933424	2784	2000
931368	2784	4056
929312	2768	736
927256	2768	2792
923144	2752	1528
921088	2752	3584
919032	2736	264
916976	2736	2320

912864	2720	1056
910808	2720	3112
906696	2704	1848
904640	2704	3904
902584	2688	584
900528	2688	2640
896416	2672	1376
894360	2672	3432
892304	2656	112
886136	2640	904
884080	2640	2960
879968	2624	1696
877912	2624	3752
875856	2608	432
873800	2608	2488
869688	2592	1224
867632	2592	3280
863520	2576	2016
861464	2576	4072
859408	2560	752
857352	2560	2808
853240	2544	1544
851184	2544	3600
849128	2528	280
847072	2528	2336
842960	2512	1072
840904	2512	3128
836792	2496	1864
834736	2496	3920
832680	2480	600
830624	2480	2656
826512	2464	1392
824456	2464	3448
822400	2448	128
820344	2448	2184
816232	2432	920
814176	2432	2976
810064	2416	1712
808008	2416	3768
805952	2400	448
803896	2400	2504
799784	2384	1240
797728	2384	3296
793616	2368	2032
791560	2368	4088
789504	2352	768
787448	2352	2824
783336	2336	1560

781280	2336	3616
779224	2320	296
777168	2320	2352
773056	2304	1088
771000	2304	3144
766888	2288	1880
764832	2288	3936
762776	2272	616
760720	2272	2672
756608	2256	1408
754552	2256	3464
752496	2240	144
750440	2240	2200
746328	2224	936
744272	2224	2992
740160	2208	1728
738104	2208	3784
736048	2192	464
733992	2192	2520
729880	2176	1256
727824	2176	3312
723712	2160	2048
721656	2160	4104
719600	2144	784

。

本实施例中，对于未来城域电信传输和城域 DCI 互联场景，将采用 FEC 编码并结合 PCS 技术以满足更长传输距离。PCS 处理的引入，使得进行“PCS 处理和 FEC 编码”（即说明书中的第一子处理）的数据处理颗粒度不再是 3552，而是更小的数值m，此时需要重新设计从数据帧获取数据的行数，进行 CRC 校验和插入填充比特的数量，以使得整体数据处理操作更简单，复杂度更低，功耗更低。而且，在第一比特数据为 FlexO-4e 或 400G-ZR 帧数据的情况下，对应的波特率均为小于 100GBaud，可较好兼容现有 400G 数据中心、城域网络的方案。

在一些可能的实施方式中，$d_{scr} = 336 \times m$。本申请实施例中，比特数目为d_{in}的第一比特数据经过数据处理（包括说明书中的“第一数据处理”和“第二数据处理”）之后，刚好得到一个 DSP 帧，此时，可以得到$d_{scr} = d_{in} + d_{CP} = L \times m = 336 \times m$，使得所提出的数据处理方法具有较低实现复杂度，功耗更低。

在一些可能的实施方式中，所述第一比特子集的比特数量是 16 或 32 的整数倍，所述第二比特子集包括 2048 个比特，所述第三比特子集包括 1504 比特，所述第二比特集合包括 4096 个比特。

在一些可能的实施方式中，所述第一比特数据包括 r 行 q 列个比特，d_{in}=r×q，r 为大于 1 的整数，q=2056 或 10280。

在一些可能的实施方式中，所述处理单元具体用于：对所述第一比特数据中每 r/p 行总共 r/p×q 个比特进行 CRC-32 校验，添加长度为 32 比特的校验比特，$d_{CRC} = 32 \times p$，r 能被p整除，p 为大于 1 的整数；

或者，对所述第一比特数据中前$r_0 \times (p-1)$行中的每r_0行共$q \times r_0$个比特进行 CRC-32 校验以添加长度为 32 比特的校验比特，并对所述第一数据中最后r_1行共$q \times r_1$个比特进行 CRC-32 校验以添加长度为 32 比特的校验比特，其中，$d_{CRC} = 32 \times p$，r 不能被p整除，p 为大于 1 的整数，$r_0 \times (p-1) + r_1 = r$，整数$r_0$大于整数$r_1$。

在一些可能的实施方式中，q=2056，r，m，d_{CP}，p 和 r_1的值为下表中的任意一组：

r	m	d_{CP}	p	r_1
577	3536	1784	29	17

576	3536	3840	29	16
574	3520	2576	29	14
572	3504	1312	29	12
571	3504	3368	29	11
569	3488	2104	29	9
566	3472	2896	29	6
564	3456	1632	29	4
563	3456	3688	29	3
561	3440	2424	29	1
559	3424	1160	28	19
558	3424	3216	28	18
556	3408	1952	28	16
555	3408	4008	28	15
553	3392	2744	28	13
551	3376	1480	28	11
550	3376	3536	28	10
546	3344	1008	28	6
545	3344	3064	28	5
543	3328	1800	28	3
542	3328	3856	28	2
540	3312	2592	27	
538	3296	1328	27	18
537	3296	3384	27	17
535	3280	2120	27	15
532	3264	2912	27	12
530	3248	1648	27	10
529	3248	3704	27	9
527	3232	2440	27	7
525	3216	1176	27	5
524	3216	3232	27	4
521	3200	4024	27	1
519	3184	2760	26	19
517	3168	1496	26	17
516	3168	3552	26	16
514	3152	2288	26	14
512	3136	1024	26	12
511	3136	3080	26	11
509	3120	1816	26	9
508	3120	3872	26	8
506	3104	2608	26	6
504	3088	1344	26	4
503	3088	3400	26	3
501	3072	2136	26	1
499	3056	872	25	19
498	3056	2928	25	18
496	3040	1664	25	16
495	3040	3720	25	15
493	3024	2456	25	13

491	3008	1192	25	11
490	3008	3248	25	10
488	2992	1984	25	8
487	2992	4040	25	7
485	2976	2776	25	5
483	2960	1512	25	3
482	2960	3568	25	2
480	2944	2304	24	
478	2928	1040	24	18
477	2928	3096	24	17
475	2912	1832	24	15
474	2912	3888	24	14
472	2896	2624	24	12
470	2880	1360	24	10
469	2880	3416	24	9
467	2864	2152	24	7
465	2848	888	24	5
464	2848	2944	24	4
462	2832	1680	24	2
461	2832	3736	24	1
459	2816	2472	23	19
457	2800	1208	23	17
456	2800	3264	23	16
454	2784	2000	23	14
453	2784	4056	23	13
452	2768	736	23	12
451	2768	2792	23	11
449	2752	1528	23	9
448	2752	3584	23	8
446	2736	2320	23	6
444	2720	1056	23	4
443	2720	3112	23	3
441	2704	1848	23	1
440	2704	3904	22	
438	2688	2640	22	18
436	2672	1376	22	16
435	2672	3432	22	15
431	2640	904	22	11
430	2640	2960	22	10
428	2624	1696	22	8
427	2624	3752	22	7
425	2608	2488	22	5
423	2592	1224	22	3
422	2592	3280	22	2
420	2576	2016	21	
419	2576	4072	21	19
418	2560	752	21	18
417	2560	2808	21	17

415	2544	1544	21	15
414	2544	3600	21	14
412	2528	2336	21	12
410	2512	1072	21	10
409	2512	3128	21	9
407	2496	1864	21	7
406	2496	3920	21	6
404	2480	2656	21	4
402	2464	1392	21	2
401	2464	3448	21	1
399	2448	2184	20	19
397	2432	920	20	17
396	2432	2976	20	16
394	2416	1712	20	14
393	2416	3768	20	13
391	2400	2504	20	11
389	2384	1240	20	9
388	2384	3296	20	8
386	2368	2032	20	6
385	2368	4088	20	5
384	2352	768	20	4
383	2352	2824	20	3
381	2336	1560	20	1
380	2336	3616	19	
378	2320	2352	19	18
376	2304	1088	19	16
375	2304	3144	19	15
373	2288	1880	19	13
372	2288	3936	19	12
371	2272	616	19	11
370	2272	2672	19	10
368	2256	1408	19	8
367	2256	3464	19	7
365	2240	2200	19	5
363	2224	936	19	3
362	2224	2992	19	2
360	2208	1728	18	
359	2208	3784	18	19
357	2192	2520	18	17
355	2176	1256	18	15
354	2176	3312	18	14
352	2160	2048	18	12
351	2160	4104	18	11
350	2144	784	18	10

。

本实施例中提供的数据处理方法需要执行 CRC 校验，可以对数据检错，并复用现有 800G-ZR 的 CRC 技术，实现简单。进一步地，在第一比特数据为 FlexO-4e 或 400G-ZR 帧数据的情况下，可保证对应的波特率均为小于 100GBaud，可较好兼容现有 400G 数据中心、城域网络的方案。

在一些可能的实施方式中，m 为 32 的整数倍，q=2056；r，m，d_{CP}，p 和 r_1的值为下表中的任意一组：

r	m	d_{CP}	p	r_1
574	3520	2576	29	14
569	3488	2104	29	9
564	3456	1632	29	4
563	3456	3688	29	3
559	3424	1160	28	19
558	3424	3216	28	18
553	3392	2744	28	13
543	3328	1800	28	3
542	3328	3856	28	2
538	3296	1328	27	18
537	3296	3384	27	17
532	3264	2912	27	12
527	3232	2440	27	7
521	3200	4024	27	1
517	3168	1496	26	17
516	3168	3552	26	16
512	3136	1024	26	12
511	3136	3080	26	11
506	3104	2608	26	6
501	3072	2136	26	1
496	3040	1664	25	16
495	3040	3720	25	15
491	3008	1192	25	11
490	3008	3248	25	10
485	2976	2776	25	5
480	2944	2304	24	
475	2912	1832	24	15
474	2912	3888	24	14
470	2880	1360	24	10
469	2880	3416	24	9
465	2848	888	24	5
464	2848	2944	24	4
459	2816	2472	23	19
454	2784	2000	23	14
453	2784	4056	23	13
449	2752	1528	23	9
448	2752	3584	23	8
444	2720	1056	23	4
443	2720	3112	23	3
438	2688	2640	22	18
428	2624	1696	22	8
427	2624	3752	22	7
423	2592	1224	22	3
422	2592	3280	22	2
418	2560	752	21	18
417	2560	2808	21	17

412	2528	2336	21	12
407	2496	1864	21	7
406	2496	3920	21	6
402	2464	1392	21	2
401	2464	3448	21	1
397	2432	920	20	17
396	2432	2976	20	16
391	2400	2504	20	11
386	2368	2032	20	6
385	2368	4088	20	5
381	2336	1560	20	1
380	2336	3616	19	
376	2304	1088	19	16
375	2304	3144	19	15
371	2272	616	19	11
370	2272	2672	19	10
365	2240	2200	19	5
360	2208	1728	18	
359	2208	3784	18	19
355	2176	1256	18	15
354	2176	3312	18	14
350	2144	784	18	10

。

本申请实施例提供的参数，可以保证 FEC 性能的基础上，硬件实现简单，复杂度更低。

在一些可能的实施方式中，q=10280，r，m，d_{CP}，p 和 r_1 的值为下表中的任意一组：

r	m	d_{CP}	p	r_1
113	3472	4952	29	1
112	3440	4480	28	
111	3408	4008	28	3
110	3376	3536	28	2
109	3344	3064	28	1
108	3312	2592	27	
107	3280	2120	27	3
106	3248	1648	27	2
105	3216	1176	27	1
102	3136	5136	26	2
101	3104	4664	26	1
100	3072	4192	25	
99	3040	3720	25	3
98	3008	3248	25	2
97	2976	2776	25	1
96	2944	2304	24	
95	2912	1832	24	3
94	2880	1360	24	2
93	2848	888	24	1
90	2768	4848	23	2
89	2736	4376	23	1
88	2704	3904	22	

87	2672	3432	22	3
86	2640	2960	22	2
85	2608	2488	22	1
84	2576	2016	21	
83	2544	1544	21	3
82	2512	1072	21	2
79	2432	5032	20	3
78	2400	4560	20	2
77	2368	4088	20	1
76	2336	3616	19	
75	2304	3144	19	3
74	2272	2672	19	2
73	2240	2200	19	1
72	2208	1728	18	
71	2176	1256	18	3
70	2144	784	18	2

。

本实施例中提供的数据处理方法需要执行 CRC 校验，可以对数据检错，并复用现有 800G-ZR 的 CRC 技术，实现简单。进一步地，在第一比特数据为 FlexO-4e 或 400G-ZR 帧数据的情况下，可保证对应的波特率均为小于 100GBaud，可较好兼容现有 400G 数据中心、城域网络的方案。

在一些可能的实施方式中，m 为 32 的整数倍，q=10280，r，m，d_{CP}，p 和 r_1的值为下表中的任意一组：

r	m	d_{CP}	p	r_1
102	3136	5136	26	2
101	3104	4664	26	1
100	3072	4192	25	
99	3040	3720	25	3
98	3008	3248	25	2
97	2976	2776	25	1
96	2944	2304	24	
95	2912	1832	24	3
94	2880	1360	24	2
93	2848	888	24	1
79	2432	5032	20	3
78	2400	4560	20	2
77	2368	4088	20	1
76	2336	3616	19	
75	2304	3144	19	3
74	2272	2672	19	2
73	2240	2200	19	1
72	2208	1728	18	
71	2176	1256	18	3
70	2144	784	18	2

。

本申请实施例提供的参数，可以保证 FEC 性能的基础上，硬件实现简单，复杂度更低。

在一些可能的实施方式中，所述第三比特子集包括 1504 比特，所述第一比特子集包括 m-1504 比特；所述 PCS 单元用于：将所述第一比特子集分成 16 组序列，每组序列包括(m-1504)/16 个比特；分别对所述 16 组序列进行 PCS 子处理，得到与所述 16 组序列对应的 16 组输出序列，其中，每组输出序列包括

128 比特，所述 16 组输出序列构成所述第二比特子集。

在一些可能的实施方式中，所述第三比特子集包括 1504 比特，所述第一比特子集包括 m-1504 比特；所述对比特数量为 m 的第一比特集合中的第一比特子集进行概率星座整形 PCS 处理得到第二比特子集，包括：将所述第一比特子集分成 32 组序列，每组序列包括(m-1504)/32 个比特；分别对所述 32 组序列进行 PCS 子处理，得到与所述 32 组序列对应的 32 组输出序列，其中，每组输出序列包括 64 比特，所述 32 组输出序列构成所述第二比特子集。

在一些可能的实施方式中，所述数据处理装置还包括第二处理单元和数字信号处理 DSP 成帧单元；所述第二处理单元，用于对第四比特数据进行符号映射和偏振划分得到双偏振符号序列，其中，所述符号映射和所述偏振划分将 8 个比特映射为 1 个双偏振 DP-16QAM 符号，所述第四比特数据包括多个所述第二比特集合；所述 DSP 成帧单元，用于对所述双偏振符号序列进行 DSP 成帧，得到 DSP 帧（也称为超帧）。

在一些可能的实施方式中，所述 DSP 成帧单元，用于将每 172032 个双偏振 DP-16QAM 符号进行成帧处理得到一个 DSP 帧。

第三方面，本申请提供了一种芯片，该芯片包括处理器，处理器用于执行如第一方面任一实施方式介绍的方法。

第四方面，本申请实施例提供了一种光模块。该光模块包括处理器和接口，处理器用于执行如第一方面任一实施方式介绍的方法，并通过接口发送信号。例如，接口用于对来自处理器的信号进行发送或将接收到的信号传输至所述处理器。

在一些可能的实施方式中，该接口具体为电接口，处理器用于通过接口发送电信号。例如，处理器执行如第一方面任一实施方式介绍的方法并对第二比特集合进行数据处理得到 DSP 帧，并通过接口发送 DSP 帧。这里的数据处理包括符号映射、偏振划分和 DSP 成帧。

在一些可能的实施方式中，该接口具体为光接口，光模块还包括调制器。例如，处理器执行如第一方面任一实施方式介绍的方法并对第二比特集合进行数据处理得到 DSP 帧，调制器再根据 DSP 帧进行电光转换等信号处理得到光信号，进而通过接口发送光信号。这里的数据处理包括符号映射、偏振划分和 DSP 成帧。

第五方面，本申请实施例提供了一种发送设备。该发送设备包括主机侧设备和如第四方面任一实施方式介绍的光模块，光模块用于根据来自主机侧设备的数据生成光信号，并发送光信号。

第六方面，本申请实施例提供了一种通信系统，该通信系统包括第五方面介绍的发送设备和接收设备，发送设备用于向接收设备发送光信号。

第七方面，本申请提供了一种计算机可读存储介质，计算机可读存储介质上存储有指令，指令被计算机执行时，以使得实现如第一方面任一实施方式介绍的方法。

第八方面，本申请提供了一种计算机程序产品，该计算机程序产品包括程序指令，当计算机程序产品被执行时，用于实现上述第一方面任一实施方式介绍的方法。

附图说明

图 1 为本申请实施例应用的一种通信系统示意图；

图 2 为一种数据帧的结构示意图；

图 3 为本申请实施例中数据处理的一个实施例示意图；

图 4 为本申请实施例中第一子处理的一个实施例示意图；

图 5（a）为本申请实施例中第二子处理的一个实施例示意图；

图 5（b）为本申请实施例中第二子处理的另一个实施例示意图；

图 5（c）为本申请实施例中第二子处理的又一个实施例示意图；

图 6 为本申请实施例中对数据进行 CRC 的一种实施方式示意图；

图 7 为本申请实施例中对数据进行 CRC 的另一种实施方式示意图；

图 8 为本申请实施例中对数据进行 CRC 的另一种实施方式示意图；

图 9 为本申请实施例中数据处理装置的一种结构示意图；

图 10 为本申请实施例中数据处理装置的另一种结构示意图；

图 11 为本申请实施例中光模块的一种结构示意图；

图 12 为本申请实施例中发送设备的一种结构示意图。

具体实施方式

本申请实施例提供了一种数据处理方法和数据处理装置，采用 FEC 编码并结合 PCS 技术，降低方案所需波特率，并使得整体数据处理操作更简单，复杂度更低，功耗更低，满足长距离传输场景。

需要说明的是，本申请说明书和权利要求书及上述附图中的术语“第一”、“第二”等用于区别类似的对象，而非限定特定的顺序或先后次序。应该理解，上述术语在适当情况下可以互换，以便在本申请描述的实施例能够以除了在本申请描述的内容以外的顺序实施。此外，术语“包括”和“具有”以及他们的任何变形，意图在于覆盖不排他的包含。例如，包含了一系列步骤或单元的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或单元，而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

图 1 为本申请实施例应用的一种通信系统示意图。如图 1 所示，在发送端，信源提供待发送的数据流，发端数据处理器接收该数据流。发端数据处理器先对其进行包含 PCS 处理、编码、交织、符号映射、DSP 成帧的数据处理得到符号数据流，再送入发端信号处理器进行信号处理，进而经过信道传输到达接收设备。接收设备接收到因为信道中的噪声或者其他损伤产生的失真信号后，送到收端信号处理器进行色散补偿、同步、相位恢复等操作，然后送入收端数据处理进行包含解调、解交织、译码处理等操作后恢复出原有数据，进而将恢复的数据发给信宿。在一些具体应用中，所述发端数据处理器可以在光模块（optical module）中实现。例如，该光模块可以为 400G 长途（Long Haul，LH）模块（400G-LH module，一种相干光模块）。

需要说明的是，本申请实施例中涉及的公式中出现的一些运算符号，本申请实施例中出现的$\lfloor a \rfloor$运算，表示对 a 进行向下取整运算，例如$\lfloor 1.5 \rfloor=1$，$\lfloor -1.5 \rfloor=-2$。本申请实施例中出现的 b%c 运算，表示 b 对 c 取模，当 b 为非负数时，b%c 的值取 b 除以 c 的余数，例如，8%3=2，2%3=2。当 b 为负数时，b%c 的值为将 b 加上 c 的整数倍成为正整数后再进行取模运算，例如，-9%16=7，-10%3=2。

需要说明的是，本申请说明书和权利要求书中的比特集合、比特子集，只是为了便于描述而引入的概念。在实际应用中，数据流为一个整体并不存在划分，每个比特集合、比特子集可以视作数据流中的一个或多个比特。应理解，比特集合、比特子集也可以以矩阵、数组、序列等形式呈现，具体此处不做限定。

图 2 为一种数据帧（frame）的结构示意图。如图 2 所示，数据帧包括多行比特，且每行包括 q 个比特。应理解，本申请不限定数据帧的具体类型。作为一个示例，一个 400G 帧包含 256 行，每行包含 10280 比特。作为另一个示例，为了更好适配 PCS 处理，一个 400G 帧包含 1280 行，每行包含 2056 比特。在一些具体应用中，整数 q 是 257 的整数倍；典型的，q 为 10280、8224、5140、4112、或 2056 等。

需要说明的是，本申请提供的数据处理方法可以分为两个部分，分别记为“第一数据处理”和“第二数据处理”，如图 3 所示。下面分别对“第一数据处理”、“第二数据处理”进行详细介绍。

1）第一数据处理

如图 3 所示，第一数据处理包含添加循环冗余校验（Cyclic Redundancy Check，CRC）和/或填充（Pad）比特，和加扰（scrambling）。在实际应用中，可以执行添加 CRC 和插入填充中的至少一项操作。

作为一个示例，第一比特数据共$d_{in}=q\times r$个比特。之后，根据比特数量为d_{in}的第一比特数据，获取第二比特数据的方式可以为：对第一比特数据进行添加 CRC 和/或填充比特得到第二比特数据。其中，在 CRC 操作中会加入d_{CRC}个 CRC 校验比特，在添加填充比特操作中会插入d_{PAD}个填充（Pad）比特。进而，对包含$d_{scr}=q\times r+d_{CRC}+d_{PAD}$个比特的第二比特数据进行加扰得到比特数量为$d_{scr}$的第三比特数据。其中，$d_{CRC}$为大于或等于 0 的整数，$d_{PAD}$为大于或等于 0 的整数，可以记$d_{CP}=d_{CRC}+d_{PAD}$，此时$d_{scr}=q\times r+d_{CP}$。在此情况下，CRC 校验和插入填充比特对应的冗余为$OH_{CP}=d_{scr}/d_{in}-1=(d_{CRC}+d_{PAD})/(q\times r)$。需要说明的是，在一些具体应用中，所述填充（Pad）比特为全 0 比特。

作为一个示例，当$d_{PAD}=0$时，经过 CRC 校验的$q\times r+d_{CRC}$个比特不再进行添加填充比特，而是直接进行加扰。作为另一个示例，为了实现更低时延更低复杂度，会旁路（bypass）CRC 校验并用填充比特来代替，即此时$d_{CRC}=0$。

需要说明的是，在一些具体应用场景，CRC 校验采用 CRC-32。作为一个示例，对第一比特数据一共进行 p 次 CRC-32 操作，其中p为大于 1 的整数且整数 r 能被p整除，即r/p是整数。此时，$d_{CRC}=32\times p$。更具体的，针对从数据帧中获取的 r 行数据，每r/p行共$q\times r/p$个比特进行 CRC-32 操作添加长度 32 比特

的校验比特，重复进行p次 CRC-32 操作添加总共$d_{CRC}=32\times p$个比特的 CRC-32 校验比特。需要说明的是，在一些具体实现中，选取$q\times r/p$ =41120，此时 CRC-32 编码及对应的 CRC 检测操作可直接采用现有 800G-ZR 中的 CRC-32 操作，其中现有 800G-ZR 中对 4 行 10280 列的共4×10280 =41120 比特数据进行 CRC-32 操作。在一些具体实现中，选取$q=2056$，且$r/p=20$。

需要说明的是，在另一些具体应用场景，对第一比特数据一共进行 p 次 CRC-32 操作，其中 CRC-32 操作次数p为大于 1 的整数但整数r不能被p整除，即r/p不是整数。此时，考虑$r_0\times(p-1)+r_1=r$，其中，整数r_0大于整数r_1。在一些具体实现中，整数$r_0=\lfloor r/p\rfloor+1$，整数$r_1=r-r_0\times(p-1)$，且$r_0>r_1$。下面给出一种具体的实施方式，针对从数据帧中获取的r行数据，对r行数据中最前面$r_0\times(p-1)$行中的每r_0行共$q\times r_0$个比特分别进行 CRC-32 校验，添加长度为 32 比特的校验比特，得到共$32\times(p-1)$个 CRC 校验比特；对最后的r_1行共$q\times r_1$个比特进行一次 CRC-32 操作，添加长度 32 比特的校验比特；总共得到$d_{CRC}=32\times(p-1)+32=32\times p$个 CRC-32 校验比特。需要说明的是，在一些具体实现中，选取$q\times r_0$ =41120，此时 CRC-32 编码及对应的 CRC 检测操作可直接采用现有 800G-ZR 中的 CRC-32 操作，其中现有 800G-ZR 中对 4 行 10280 列的共4×10280 =41120 比特数据进行 CRC-32 操作。在一些具体实现中，选取$q=2056$，且$r_0=20$。

2）第二数据处理

如图 3 所示，第二数据处理包含第一子处理和第二子处理。第一数据处理输出的所述第三比特数据包含 L 个第一比特集合，其中每个第一比特集合的比特数量为m。此时，$d_{scr}=q\times r+d_{CP}=L\times m$。所述第二数据处理中的第一子处理将 L 个第一比特集合中的每一个第一比特集合进行包含 PCS、交织和编码的处理得到一个第二比特集合，总共得到 L 个第二比特集合。所述第二数据处理中的第二子处理将包含 L 个第二比特集合的第四比特数据进行包含符号映射（Symbol Mapping）、偏振划分（Polarization Distribution）和 DSP 成帧（framing）的数据处理，得到一个 DSP 帧（也可以称为 DSP 超帧）。此时，共$d_{in}=q\times r$个比特的第一比特数据经过第一数据处理和第二数据处理，刚好得到一个 DSP 帧，使得所提出的数据处理方法具有较低实现复杂度，功耗更低。

下面介绍第二数据处理中的第一子处理的详细操作。如图 4 所示，所述第一比特集合是由第一比特子集和第三比特子集组成；第一比特子集进行概率星座整形 PCS 处理得到第二比特子集；对所述第二比特子集和所述第三比特子集进行包含交织和 FEC 编码的处理后，得到第二比特集合。

需要说明的是，本申请的一个实施例中，考虑所述第二比特集合的比特数量为 4096 个比特，所述第二比特子集的比特数量为 2048 个比特，所述第三比特子集的比特数量为 1504 个比特，第一比特子集的比特数量为v =m-1504 个比特。所述编码将第三比特子集和第二比特子集总共 2048+1504=3552 个比特进行编码添加校验比特得到 4096 个编码后比特。更具体的，所述 3552 个比特采用 32 行 111 列的比特矩阵表示，也称为信息比特矩阵。所述编码对每行 111 个比特进行编码添加 17 个比特，得到 32 行 128 列共 4096 个编码后比特。所述 32 行 128 列的比特矩阵也称为码字比特矩阵。 所述编码采用空间耦合编码，即当前时刻对 3552 个比特进行编码得到 4096 个编码后比特时，还利用之前时刻已编码得到的 4096 个编码后比特，例如采用 oFEC 编码，这里不做具体限制。这里应理解，有 m<3552。

需要说明的是，在进行 FEC 编码之前还可以对所述 3552 个比特进行编码前交织，所述编码前交织也称为分组映射（Block Map），也可称为编码前交织（pre-FEC interleaver），也可称为编码前置换（pre-FEC Permutation）。在进行 FEC 编码之后还可以对所述 4096 个编码后比特进行编码后交织。这里编码前交织和编码后交织统称为交织，不做具体限制。需要说明的是，第一子处理得到的 L 个第二比特集合还可进行交织处理后再进行第二子处理，这里不做具体限制。

下面介绍第二数据处理中的第二子处理详细操作。第二子处理包含符号映射（Symbol Mapping）、偏振划分（Polarization Distribution）和 DSP 成帧（framing），可参考图 5（a）、图 5（b）或图 5（c）进行理解。

如图 5（a）所示，第四比特数据经过符号映射（Symbol Mapping）和偏振划分（Polarization Distribution）得到双偏振符号序列。所述双偏振符号序列也称为成帧前符号序列。需要说明的是，符号映射和偏振划分也可以简称为双偏振（Dual-polarization，简称为 DP）符号映射，例如 16QAM 符号映射和偏振划分可简称为 DP-16QAM 符号映射或者 DP-16QAM 映射。所述符号映射将第四比特数据中每 4 个比特进行符号映射为一个符号。符号映射和偏振划分（即双偏振符号映射）将第四比特数据中每 8 个比特映射为 1 个双偏振符号，其中 1 个双偏振符号包含 1 个 X 偏振（X polarization）方向符号和 1 个 Y

偏振（Y polarization）方向符号。接着，第二子处理对所述双偏振符号序列（成帧前符号序列）进行DSP 成帧（framing）操作。具体地，在 X 偏振方向和 Y 偏振方向上分别插入帧同步符号序列（Frame Alignment Word Sequence，简称 FAW Sequence）、训练符号序列（Training symbols Sequence），保留符号序列（Reserved symbols）和导频符号序列（Pilot symbols Sequence）中至少一种序列，得到待发送的双偏振符号序列，也称为 DSP 帧。例如，DSP 成帧操作将每 172032 个双偏振符号进行 DSP 成帧处理得到一个 DSP 帧。

在一些具体应用中，DSP 成帧操作将每 172032 个双偏振符号（成帧前符号）进行 DSP 成帧处理得到一个包含 178176 个双偏振符号的 DSP 帧，其 DSP 成帧的冗余为$h_{\mathrm{DSP}} = 178176/172032 - 1 = 1/28 \approx 3.57\%$。在另一些具体应用中，DSP 成帧操作将每 172032 个双偏振符号（成帧前符号）进行 DSP 成帧处理得到一个包含 175104 个双偏振符号的 DSP 帧，其 DSP 成帧的冗余为$h_{\mathrm{DSP}} = 175104/172032 - 1 = 1/56 \approx 1.79\%$。一些实施例中，在 X 偏振方向或 Y 偏振方向上，DSP 帧的每N_{G}个符号中固定位置的一个符号为导频符号。作为一个举例，每N_{G}个符号中的第一个符号为导频符号。典型的，$N_{\mathrm{G}} = 32$或64或 96 或 128 等。在一些具体应用中，将所述符号映射称为 16QAM 映射。

需要说明的是，帧同步符号用于帧同步对齐，训练符号用于链路训练，导频符号用于载波相位恢复，保留符号用于未来使用和创新。保留符号的取值可以是已知不变的，也可以是随机的（randomized），保留符号的取值也可以称之为图样（pattern）。在一些具体实施例中，所述 DSP 帧包含多个子帧（subframe），所述 DSP 帧可称为超帧（super-frame）。在另一些具体实施例中，所述 DSP 帧也可称为多帧（multi-frame），保留符号也可称为固定填充（fixed stuff,FS），帧同步符号也可称为多帧对齐信号（multi-frame alignment signal, MFAS）。

需要说明的是，图 5（a）所示的 DSP 成帧操作是在符号（symbol）上进行操作，也就是 DSP 成帧是在符号映射操作之后。DSP 成帧操作还可以在比特（bit）上进行操作，如图 5（b）和图 5（c）所示，DSP 成帧是在符号映射操作之前。图 5（b）为本申请实施例中第二子处理的另一个实施例示意图。如图 5（b）所示，DSP 成帧操作是在符号映射前，对第四比特数据插入帧同步符号序列、训练符号序列、保留符号序列和导频符号序列所对应的比特，再经过符号映射并进行偏振划分可得到跟图 5（a）操作相同的 DSP 帧（也称为超帧）。图 5（c）为本申请实施例中第二子处理的又一个实施例示意图。如图 5（c）所示，DSP 成帧操作是在符号映射前，对第四比特数据插入帧同步符号序列、训练符号序列、保留符号序列和导频符号序列所对应的比特，再经过偏振划分并进行符号映射也可得到跟图 5（a）操作相同的 DSP 帧（也称为超帧）；应理解，不排除还有其他成帧操作，本申请不再赘述。

本申请中，考虑 DSP 成帧操作将每 172032 个双偏振 DP-16QAM 符号（成帧前符号）进行 DSP 成帧处理得到一个 DSP 帧（也称为超帧）。包含 L 个第二比特集合的第四比特数据的比特数量为$172032 \times 8 = 1376256$。考虑第二比特集合的比特数量为 4096 个比特，有$\mathrm{L} = 1376256/4096 = 336$。此时，有$\mathrm{d_{scr}} = \mathrm{q} \times \mathrm{r} + \mathrm{d_{CP}} = \mathrm{L} \times \mathrm{m} = 336 \times \mathrm{m}$。

需要说明的是，所述 PCS 处理将输入比特进行处理，使得输出的比特中 0、1 出现的概率值是不相同的。在一些具体应用中，所述 PCS 处理也可称为分布匹配器（Distribution Matcher，DM）。所述 PCS 处理可以采用多种具体方式进行实现，例如采用算术编码（arithmetic coding）、查找表（lookup table，LUT）、网格编码（trellis coding）等。

为了 PCS 处理的实现简单，通常采用多个 PCS 子处理来实现所述 PCS 处理效果，每个 PCS 子处理将输入的$\mathrm{k_0}$个比特进行 PCS 子处理得到输出的$\mathrm{n_0}$个比特，其中，$\mathrm{n_0} > \mathrm{k_0}$。需要说明的是，$\mathrm{n_0}$的取值越大，通常 PCS 处理的性能越好，但复杂度越高，实际应用中需做性能和复杂度的折中。典型的，考虑$\mathrm{n_0} = 128$或 64。在一些具体应用中，所述 PCS 处理可以采用多个查找表（lookup table，LUT）来实现，即每个 PCS 子处理采用查找表 LUT 处理来进行实现。每个查找表将输入的$\mathrm{k_0}$个比特进行 LUT 映射得到输出的$\mathrm{n_0}$个比特，其中$\mathrm{n_0} > \mathrm{k_0}$。此时，所述 PCS 处理也可称为 PCS LUT 处理。

在一些具体应用中，上述查找表 LUT 处理还可采用多个子查找表来实现以降低复杂度。在一些具体实施方式中，所述多个子查找表的输入比特数可以是不相同的，且输出比特数也可以是不相同的。例如，每个查找表将输入的$\mathrm{k_0}$个比特进行 LUT 映射得到输出的$\mathrm{n_0} = 128$个比特，所述查找表由 12 个子查找表构成，其中 4 个子查找表的输出比特数为 10，8 个子查找表的输出比特数为 11，所述 12 个子查找表的总共$4 \times 10 + 8 \times 11 = 128$个输出比特作为所述查找表的$\mathrm{n_0} = 128$个输出比特。

在一个实施例中，包含$\mathrm{v} = \mathrm{m} - 1504$个比特的第一比特子集进行 PCS 处理得到包含 2048 个比特的第

二比特子集。本申请实施例中，考虑n_0具体取值为 128、64、32、或 16，可以便于 PCS 处理的具体实现，且具有较低的复杂度。也就是说，$v = m - 1504$是 2048/128=16、2048/64=32、2048/32=64、或 2048/16=128 的整数倍.这里，我们考虑$v = m - 1504$是 2048/128=16 的整数倍，且$m < 3552$，此时，m 也为 16 的整数倍。

在一些具体应用中，所述对比特数量为 m 的第一比特集合中的第一比特子集进行概率星座整形 PCS 处理得到包含 2048 个比特的第二比特子集，包括：将所述第一比特子集分成 16 组序列，每组序列包括(m-1504)/16 个比特；分别对所述 16 组序列进行 PCS 子处理，得到与所述 16 组序列对应的 16 组输出序列，其中，每组输出序列包括 128 比特，所述 16 组输出序列构成 128×16=2048 个比特的所述第二比特子集。

在另一些具体应用中，所述对比特数量为 m 的第一比特集合中的第一比特子集进行概率星座整形 PCS 处理得到包含 2048 个比特的第二比特子集，包括：将所述第一比特子集分成 32 组序列，每组序列包括(m-1504)/32 个比特；分别对所述 32 组序列进行 PCS 子处理，得到与所述 32 组序列对应的 32 组输出序列，其中，每组输出序列包括 64 比特，所述 32 组输出序列构成包含 64×32=2048 个比特的所述第二比特子集。

第一比特数据包含 r 行 q 列比特，共$d_{in} = q \times r$个比特。下面分别对$q = 2056$和$q = 10280$两种情况进行描述。

第一方面，考虑$q = 2056$的情况

第三比特数据的比特数量为$d_{scr} = q \times r + d_{CP} = L \times m = 336 \times m$。应理解，添加的 CRC 和/或填充比特的比特数量d_{CP}越大，所引入的冗余$OH_{CP} = d_{CP}/(q \times r)$越大。为了传输有效性，这里考虑$0 \le d_{CP} < 2 \times q = 4112$，既能保证传输性能，又可以使冗余较低。下面表 1 给出了对应的{d_{in}、r、m、d_{CP}}参数组合。

表 1

d_{in}	r	m	d_{CP}	v/16
1186312	577	3536	1784	127
1184256	576	3536	3840	127
1182200	575	3520	520	126
1180144	574	3520	2576	126
1176032	572	3504	1312	125
1173976	571	3504	3368	125
1171920	570	3488	48	124
1169864	569	3488	2104	124
1165752	567	3472	840	123
1163696	566	3472	2896	123
1159584	564	3456	1632	122
1157528	563	3456	3688	122
1155472	562	3440	368	121
1153416	561	3440	2424	121
1149304	559	3424	1160	120
1147248	558	3424	3216	120
1143136	556	3408	1952	119
1141080	555	3408	4008	119
1139024	554	3392	688	118
1136968	553	3392	2744	118
1132856	551	3376	1480	117
1130800	550	3376	3536	117
1128744	549	3360	216	116
1122576	546	3344	1008	115
1120520	545	3344	3064	115

1116408	543	3328	1800	114
1114352	542	3328	3856	114
1112296	541	3312	536	113
1110240	540	3312	2592	113
1106128	538	3296	1328	112
1104072	537	3296	3384	112
1102016	536	3280	64	111
1099960	535	3280	2120	111
1095848	533	3264	856	110
1093792	532	3264	2912	110
1089680	530	3248	1648	109
1087624	529	3248	3704	109
1085568	528	3232	384	108
1083512	527	3232	2440	108
1079400	525	3216	1176	107
1077344	524	3216	3232	107
1071176	521	3200	4024	106
1069120	520	3184	704	105
1067064	519	3184	2760	105
1062952	517	3168	1496	104
1060896	516	3168	3552	104
1058840	515	3152	232	103
1056784	514	3152	2288	103
1052672	512	3136	1024	102
1050616	511	3136	3080	102
1046504	509	3120	1816	101
1044448	508	3120	3872	101
1042392	507	3104	552	100
1040336	506	3104	2608	100
1036224	504	3088	1344	99
1034168	503	3088	3400	99
1032112	502	3072	80	98
1030056	501	3072	2136	98
1025944	499	3056	872	97
1023888	498	3056	2928	97
1019776	496	3040	1664	96
1017720	495	3040	3720	96
1015664	494	3024	400	95
1013608	493	3024	2456	95
1009496	491	3008	1192	94
1007440	490	3008	3248	94
1003328	488	2992	1984	93
1001272	487	2992	4040	93
999216	486	2976	720	92
997160	485	2976	2776	92
993048	483	2960	1512	91
990992	482	2960	3568	91
988936	481	2944	248	90

986880	480	2944	2304	90
982768	478	2928	1040	89
980712	477	2928	3096	89
976600	475	2912	1832	88
974544	474	2912	3888	88
972488	473	2896	568	87
970432	472	2896	2624	87
966320	470	2880	1360	86
964264	469	2880	3416	86
962208	468	2864	96	85
960152	467	2864	2152	85
956040	465	2848	888	84
953984	464	2848	2944	84
949872	462	2832	1680	83
947816	461	2832	3736	83
945760	460	2816	416	82
943704	459	2816	2472	82
939592	457	2800	1208	81
937536	456	2800	3264	81
933424	454	2784	2000	80
931368	453	2784	4056	80
929312	452	2768	736	79
927256	451	2768	2792	79
923144	449	2752	1528	78
921088	448	2752	3584	78
919032	447	2736	264	77
916976	446	2736	2320	77
912864	444	2720	1056	76
910808	443	2720	3112	76
906696	441	2704	1848	75
904640	440	2704	3904	75
902584	439	2688	584	74
900528	438	2688	2640	74
896416	436	2672	1376	73
894360	435	2672	3432	73
892304	434	2656	112	72
886136	431	2640	904	71
884080	430	2640	2960	71
879968	428	2624	1696	70
877912	427	2624	3752	70
875856	426	2608	432	69
873800	425	2608	2488	69
869688	423	2592	1224	68
867632	422	2592	3280	68
863520	420	2576	2016	67
861464	419	2576	4072	67
859408	418	2560	752	66
857352	417	2560	2808	66

853240	415	2544	1544	65
851184	414	2544	3600	65
849128	413	2528	280	64
847072	412	2528	2336	64
842960	410	2512	1072	63
840904	409	2512	3128	63
836792	407	2496	1864	62
834736	406	2496	3920	62
832680	405	2480	600	61
830624	404	2480	2656	61
826512	402	2464	1392	60
824456	401	2464	3448	60
822400	400	2448	128	59
820344	399	2448	2184	59
816232	397	2432	920	58
814176	396	2432	2976	58
810064	394	2416	1712	57
808008	393	2416	3768	57
805952	392	2400	448	56
803896	391	2400	2504	56
799784	389	2384	1240	55
797728	388	2384	3296	55
793616	386	2368	2032	54
791560	385	2368	4088	54
789504	384	2352	768	53
787448	383	2352	2824	53
783336	381	2336	1560	52
781280	380	2336	3616	52
779224	379	2320	296	51
777168	378	2320	2352	51
773056	376	2304	1088	50
771000	375	2304	3144	50
766888	373	2288	1880	49
764832	372	2288	3936	49
762776	371	2272	616	48
760720	370	2272	2672	48
756608	368	2256	1408	47
754552	367	2256	3464	47
752496	366	2240	144	46
750440	365	2240	2200	46
746328	363	2224	936	45
744272	362	2224	2992	45
740160	360	2208	1728	44
738104	359	2208	3784	44
736048	358	2192	464	43
733992	357	2192	2520	43
729880	355	2176	1256	42
727824	354	2176	3312	42

723712	352	2160	2048	41
721656	351	2160	4104	41
719600	350	2144	784	40

表 1 还给出了{r、m、d_{CP}}参数组合下，对应的v/16的具体取值。需要说明的是，$v = m - 1504$是16 的整数倍，当$n_0 = 128$时，$k_0 = v/16$，也就是所述 PCS 处理采用 16 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/16$个比特进行 PCS 子处理得到输出的$n_0 = 128$个比特。在一些具体应用中，16 个 PCS 子处理采用 16 个查找表 LUT 来实现。在另一些具体应用中，16 个 PCS 子处理采用 16 个网格编码（trellis coding）来实现。需要说明的是，在另一些具体实施例中，考虑$v = m - 1504$是 32 的整数倍，当$n_0 = 64$时，$k_0 = v/32$，也就是所述 PCS 处理采用 32 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/32$个比特进行 PCS 子处理得到输出的$n_0 = 64$个比特。在一些具体应用中，32 个 PCS 子处理采用 32 个查找表 LUT 来实现。在另一些具体应用中，32 个 PCS 子处理采用 32 个网格编码（trellis coding）来实现。

需要说明的是，在一些具体应用中第一数据处理包含添加循环冗余校验 CRC-32 操作。此时，考虑如下第一约束：当行数 r 是 20 的整数倍时，有 CRC-32 操作次数$p = r/20$，$d_{CRC} = r/20 \times 32 \le d_{CP}$，也就是对r行数据中每20行共 $20\times 2056 = 41120$ 个比特进行 CRC-32 操作；当行数 r 不是 20 的整数倍时，有 CRC-32 操作次数$p = \lfloor r/20 \rfloor + 1$，$d_{CRC} = (\lfloor r/20 \rfloor + 1) \times 32 \le d_{CP}$，其中⌊a⌋运算表示对 a 进行向下取整运算，也就是$r_0 = 20$，$r_1 = r - 20 \times (p - 1)$，对r行数据中最前面$20 \times (p - 1)$行中的每$r_0 = 20$行共$q \times r_0 = 41120$个比特进行 CRC-32，对最后的$r_1$行共$q \times r_1$个比特进行一次 CRC-32 操作。考虑$d_{PAD} = d_{CP} - d_{CRC}$。结合表 1 和上述第一约束，下面表 2 给出了对应的{r、m、d_{CP}、d_{PAD}、p}参数组合。需要说明的是，表 2 最后一列还给出了，当行数 r 不是 20 的整数倍时，对应参数r_1的具体取值。对于行数 r 是 20 的整数倍时，如表 2 中的 23 行的 r=540，对应参数r_1在表中采用斜杠代表，表示并不存在所述参数r_1。

表 2

r	m	d_{CP}	d_{PAD}	p	v/16	r_1
577	3536	1784	856	29	127	17
576	3536	3840	2912	29	127	16
574	3520	2576	1648	29	126	14
572	3504	1312	384	29	125	12
571	3504	3368	2440	29	125	11
569	3488	2104	1176	29	124	9
566	3472	2896	1968	29	123	6
564	3456	1632	704	29	122	4
563	3456	3688	2760	29	122	3
561	3440	2424	1496	29	121	1
559	3424	1160	264	28	120	19
558	3424	3216	2320	28	120	18
556	3408	1952	1056	28	119	16
555	3408	4008	3112	28	119	15
553	3392	2744	1848	28	118	13
551	3376	1480	584	28	117	11
550	3376	3536	2640	28	117	10
546	3344	1008	112	28	115	6
545	3344	3064	2168	28	115	5
543	3328	1800	904	28	114	3
542	3328	3856	2960	28	114	2
540	3312	2592	1728	27	113	
538	3296	1328	464	27	112	18
537	3296	3384	2520	27	112	17

535	3280	2120	1256	27	111	15
532	3264	2912	2048	27	110	12
530	3248	1648	784	27	109	10
529	3248	3704	2840	27	109	9
527	3232	2440	1576	27	108	7
525	3216	1176	312	27	107	5
524	3216	3232	2368	27	107	4
521	3200	4024	3160	27	106	1
519	3184	2760	1928	26	105	19
517	3168	1496	664	26	104	17
516	3168	3552	2720	26	104	16
514	3152	2288	1456	26	103	14
512	3136	1024	192	26	102	12
511	3136	3080	2248	26	102	11
509	3120	1816	984	26	101	9
508	3120	3872	3040	26	101	8
506	3104	2608	1776	26	100	6
504	3088	1344	512	26	99	4
503	3088	3400	2568	26	99	3
501	3072	2136	1304	26	98	1
499	3056	872	72	25	97	19
498	3056	2928	2128	25	97	18
496	3040	1664	864	25	96	16
495	3040	3720	2920	25	96	15
493	3024	2456	1656	25	95	13
491	3008	1192	392	25	94	11
490	3008	3248	2448	25	94	10
488	2992	1984	1184	25	93	8
487	2992	4040	3240	25	93	7
485	2976	2776	1976	25	92	5
483	2960	1512	712	25	91	3
482	2960	3568	2768	25	91	2
480	2944	2304	1536	24	90	
478	2928	1040	272	24	89	18
477	2928	3096	2328	24	89	17
475	2912	1832	1064	24	88	15
474	2912	3888	3120	24	88	14
472	2896	2624	1856	24	87	12
470	2880	1360	592	24	86	10
469	2880	3416	2648	24	86	9
467	2864	2152	1384	24	85	7
465	2848	888	120	24	84	5
464	2848	2944	2176	24	84	4
462	2832	1680	912	24	83	2
461	2832	3736	2968	24	83	1
459	2816	2472	1736	23	82	19
457	2800	1208	472	23	81	17
456	2800	3264	2528	23	81	16

454	2784	2000	1264	23	80	14
453	2784	4056	3320	23	80	13
452	2768	736	0	23	79	12
451	2768	2792	2056	23	79	11
449	2752	1528	792	23	78	9
448	2752	3584	2848	23	78	8
446	2736	2320	1584	23	77	6
444	2720	1056	320	23	76	4
443	2720	3112	2376	23	76	3
441	2704	1848	1112	23	75	1
440	2704	3904	3200	22	75	
438	2688	2640	1936	22	74	18
436	2672	1376	672	22	73	16
435	2672	3432	2728	22	73	15
431	2640	904	200	22	71	11
430	2640	2960	2256	22	71	10
428	2624	1696	992	22	70	8
427	2624	3752	3048	22	70	7
425	2608	2488	1784	22	69	5
423	2592	1224	520	22	68	3
422	2592	3280	2576	22	68	2
420	2576	2016	1344	21	67	
419	2576	4072	3400	21	67	19
418	2560	752	80	21	66	18
417	2560	2808	2136	21	66	17
415	2544	1544	872	21	65	15
414	2544	3600	2928	21	65	14
412	2528	2336	1664	21	64	12
410	2512	1072	400	21	63	10
409	2512	3128	2456	21	63	9
407	2496	1864	1192	21	62	7
406	2496	3920	3248	21	62	6
404	2480	2656	1984	21	61	4
402	2464	1392	720	21	60	2
401	2464	3448	2776	21	60	1
399	2448	2184	1544	20	59	19
397	2432	920	280	20	58	17
396	2432	2976	2336	20	58	16
394	2416	1712	1072	20	57	14
393	2416	3768	3128	20	57	13
391	2400	2504	1864	20	56	11
389	2384	1240	600	20	55	9
388	2384	3296	2656	20	55	8
386	2368	2032	1392	20	54	6
385	2368	4088	3448	20	54	5
384	2352	768	128	20	53	4
383	2352	2824	2184	20	53	3
381	2336	1560	920	20	52	1

380	2336	3616	3008	19	52	
378	2320	2352	1744	19	51	18
376	2304	1088	480	19	50	16
375	2304	3144	2536	19	50	15
373	2288	1880	1272	19	49	13
372	2288	3936	3328	19	49	12
371	2272	616	8	19	48	11
370	2272	2672	2064	19	48	10
368	2256	1408	800	19	47	8
367	2256	3464	2856	19	47	7
365	2240	2200	1592	19	46	5
363	2224	936	328	19	45	3
362	2224	2992	2384	19	45	2
360	2208	1728	1152	18	44	
359	2208	3784	3208	18	44	19
357	2192	2520	1944	18	43	17
355	2176	1256	680	18	42	15
354	2176	3312	2736	18	42	14
352	2160	2048	1472	18	41	12
351	2160	4104	3528	18	41	11
350	2144	784	208	18	40	10

表 2 还给出了{ r、m、d_{CP}、d_{PAD}、p}参数组合下，对应的v/16的具体取值。需要说明的是，$v = m - 1504$是 16 的整数倍，当$n_0 = 128$时，$k_0 = v/16$，也就是所述 PCS 处理采用 16 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/16$个比特进行 PCS 子处理得到输出的$n_0 = 128$个比特。在一些具体应用中，16 个 PCS 子处理采用 16 个查找表 LUT 来实现。在另一些具体应用中，16 个 PCS 子处理采用 16 个网格编码（trellis coding）来实现。需要说明的是，在另一些具体实施例中，考虑$v = m - 1504$是 32 的整数倍，当$n_0 = 64$时，$k_0 = v/32$，也就是所述 PCS 处理采用 32 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/32$个比特进行 PCS 子处理得到输出的$n_0 = 64$个比特。在一些具体应用中，32 个 PCS 子处理采用 32 个查找表 LUT 来实现。在另一些具体应用中，32 个 PCS 子处理采用 32 个网格编码（trellis coding）来实现。

需要说明的是，在另一些具体应用中，考虑包含 m 个比特的第一比特集合进行包含 PCS、交织和编码的第一子处理得到包含 4096 个比特的第二比特集合，其中编码操作中的编码输入比特长度和编码输出比特长度的比值为 3552:4096=111:128。考虑所述第一子处理（包含 PCS 处理、交织和编码）的输入比特长度 m 和输出比特长度 4096 的比值为 m:4096=f1:f2，其中整数 f1 和 f2 的最大公约数是 1。典型的，当 f1 和 f2 的取值越小，可使得整体硬件实现更简单，复杂度更低，功耗更低。因此 ，考虑如下第二约束，即 m 为 32 的整数倍，使得所述第一子处理的输入比特长度 m 和输出比特长度 4096 的比值为 m:4096=f:128，其中整数$f = m/32$，可结合图 4 进行理解。结合表 1 和上述第二约束，下面表 3 给出了对应的{r、m、d_{CP}、f}参数组合。

表 3

r	m	d_{CP}	f	v/16
575	3520	520	110	126
574	3520	2576	110	126
570	3488	48	109	124
569	3488	2104	109	124
564	3456	1632	108	122
563	3456	3688	108	122
559	3424	1160	107	120

558	3424	3216	107	120
554	3392	688	106	118
553	3392	2744	106	118
549	3360	216	105	116
543	3328	1800	104	114
542	3328	3856	104	114
538	3296	1328	103	112
537	3296	3384	103	112
533	3264	856	102	110
532	3264	2912	102	110
528	3232	384	101	108
527	3232	2440	101	108
521	3200	4024	100	106
517	3168	1496	99	104
516	3168	3552	99	104
512	3136	1024	98	102
511	3136	3080	98	102
507	3104	552	97	100
506	3104	2608	97	100
502	3072	80	96	98
501	3072	2136	96	98
496	3040	1664	95	96
495	3040	3720	95	96
491	3008	1192	94	94
490	3008	3248	94	94
486	2976	720	93	92
485	2976	2776	93	92
481	2944	248	92	90
480	2944	2304	92	90
475	2912	1832	91	88
474	2912	3888	91	88
470	2880	1360	90	86
469	2880	3416	90	86
465	2848	888	89	84
464	2848	2944	89	84
460	2816	416	88	82
459	2816	2472	88	82
454	2784	2000	87	80
453	2784	4056	87	80
449	2752	1528	86	78
448	2752	3584	86	78
444	2720	1056	85	76
443	2720	3112	85	76
439	2688	584	84	74
438	2688	2640	84	74
434	2656	112	83	72
428	2624	1696	82	70
427	2624	3752	82	70

423	2592	1224	81	68
422	2592	3280	81	68
418	2560	752	80	66
417	2560	2808	80	66
413	2528	280	79	64
412	2528	2336	79	64
407	2496	1864	78	62
406	2496	3920	78	62
402	2464	1392	77	60
401	2464	3448	77	60
397	2432	920	76	58
396	2432	2976	76	58
392	2400	448	75	56
391	2400	2504	75	56
386	2368	2032	74	54
385	2368	4088	74	54
381	2336	1560	73	52
380	2336	3616	73	52
376	2304	1088	72	50
375	2304	3144	72	50
371	2272	616	71	48
370	2272	2672	71	48
366	2240	144	70	46
365	2240	2200	70	46
360	2208	1728	69	44
359	2208	3784	69	44
355	2176	1256	68	42
354	2176	3312	68	42
350	2144	784	67	40

表 3 还给出了{ r、m、d_{CP}、f }参数组合下，对应的v/16的具体取值。需要说明的是，$v = m - 1504$是 16 的整数倍，当$n_0 = 128$时，$k_0 = v/16$，也就是所述 PCS 处理采用 16 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/16$个比特进行 PCS 子处理得到输出的$n_0 = 128$个比特。在一些具体应用中，16 个 PCS 子处理采用 16 个查找表 LUT 来实现。在另一些具体应用中，16 个 PCS 子处理采用 16 个网格编码（trellis coding）来实现。需要说明的是，在另一些具体实施例中，考虑$v = m - 1504$是 32 的整数倍，当$n_0 = 64$时，$k_0 = v/32$，也就是所述 PCS 处理采用 32 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/32$个比特进行 PCS 子处理得到输出的$n_0 = 64$个比特。在一些具体应用中，32 个 PCS 子处理采用 32 个查找表 LUT 来实现。在另一些具体应用中，32 个 PCS 子处理采用 32 个网格编码（trellis coding）来实现。

需要说明的是，在又一些具体应用中，考虑上述第一约束和第二约束。此时，结合表 1 和上述第一约束和第二约束，下面表 4 给出了对应的{r、m、d_{CP}、d_{PAD}、f、p }参数组合。需要说明的是，表 4 最后一列还给出了，当行数 r 不是 20 的整数倍时，对应参数r_1的具体取值。对于行数 r 是 20 的整数倍时，如表 4 中的 28 行的 r=480，对应参数r_1在表中采用斜杠代表，表示并不存在所述参数r_1。

表 4

r	m	d_{CP}	d_{PAD}	f	p	v/16	r_1
574	3520	2576	1648	110	29	126	14
569	3488	2104	1176	109	29	124	9
564	3456	1632	704	108	29	122	4

563	3456	3688	2760	108	29	122	3
559	3424	1160	264	107	28	120	19
558	3424	3216	2320	107	28	120	18
553	3392	2744	1848	106	28	118	13
543	3328	1800	904	104	28	114	3
542	3328	3856	2960	104	28	114	2
538	3296	1328	464	103	27	112	18
537	3296	3384	2520	103	27	112	17
532	3264	2912	2048	102	27	110	12
527	3232	2440	1576	101	27	108	7
521	3200	4024	3160	100	27	106	1
517	3168	1496	664	99	26	104	17
516	3168	3552	2720	99	26	104	16
512	3136	1024	192	98	26	102	12
511	3136	3080	2248	98	26	102	11
506	3104	2608	1776	97	26	100	6
501	3072	2136	1304	96	26	98	1
496	3040	1664	864	95	25	96	16
495	3040	3720	2920	95	25	96	15
491	3008	1192	392	94	25	94	11
490	3008	3248	2448	94	25	94	10
485	2976	2776	1976	93	25	92	5
480	2944	2304	1536	92	24	90	
475	2912	1832	1064	91	24	88	15
474	2912	3888	3120	91	24	88	14
470	2880	1360	592	90	24	86	10
469	2880	3416	2648	90	24	86	9
465	2848	888	120	89	24	84	5
464	2848	2944	2176	89	24	84	4
459	2816	2472	1736	88	23	82	19
454	2784	2000	1264	87	23	80	14
453	2784	4056	3320	87	23	80	13
449	2752	1528	792	86	23	78	9
448	2752	3584	2848	86	23	78	8
444	2720	1056	320	85	23	76	4
443	2720	3112	2376	85	23	76	3
438	2688	2640	1936	84	22	74	18
428	2624	1696	992	82	22	70	8
427	2624	3752	3048	82	22	70	7
423	2592	1224	520	81	22	68	3
422	2592	3280	2576	81	22	68	2
418	2560	752	80	80	21	66	18
417	2560	2808	2136	80	21	66	17
412	2528	2336	1664	79	21	64	12
407	2496	1864	1192	78	21	62	7
406	2496	3920	3248	78	21	62	6
402	2464	1392	720	77	21	60	2
401	2464	3448	2776	77	21	60	1

397	2432	920	280	76	20	58	17
396	2432	2976	2336	76	20	58	16
391	2400	2504	1864	75	20	56	11
386	2368	2032	1392	74	20	54	6
385	2368	4088	3448	74	20	54	5
381	2336	1560	920	73	20	52	1
380	2336	3616	3008	73	19	52	
376	2304	1088	480	72	19	50	16
375	2304	3144	2536	72	19	50	15
371	2272	616	8	71	19	48	11
370	2272	2672	2064	71	19	48	10
365	2240	2200	1592	70	19	46	5
360	2208	1728	1152	69	18	44	
359	2208	3784	3208	69	18	44	19
355	2176	1256	680	68	18	42	15
354	2176	3312	2736	68	18	42	14
350	2144	784	208	67	18	40	10

表 4 还给出了{ r、m、d_{CP}、d_{PAD}、f、p }参数组合下，对应的$v/16$的具体取值。需要说明的是，$v = m - 1504$是 16 的整数倍，当$n_0 = 128$时，$k_0 = v/16$，也就是所述 PCS 处理采用 16 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/16$个比特进行 PCS 子处理得到输出的$n_0 = 128$个比特。在一些具体应用中，16 个 PCS 子处理采用 16 个查找表 LUT 来实现。在另一些具体应用中，16 个 PCS 子处理采用 16 个网格编码（trellis coding）来实现。需要说明的是，在另一些具体实施例中，考虑$v = m - 1504$是 32 的整数倍，当$n_0 = 64$时，$k_0 = v/32$，也就是所述 PCS 处理采用 32 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/32$个比特进行 PCS 子处理得到输出的$n_0 = 64$个比特。在一些具体应用中，32 个 PCS 子处理采用 32 个查找表 LUT 来实现。在另一些具体应用中，32 个 PCS 子处理采用 32 个网格编码（trellis coding）来实现。

需要说明的是，假设数据帧的比特速率为W_0 Gbit/s，经过第一数据处理后对应的比特速率为$W_0 \times (r \times q + d_{CP})/(r \times q)$ Gbit/s，经过第二数据处理中的第一子处理（即 PCS、交织和编码处理）后对应的比特速率为$W_1 = W_0 \times (r \times q + d_{CP})/(r \times q) \times (4096/m)$。考虑第二数据处理中 DSP 成帧的冗余为$h_{DSP}$，即经过第二数据处理后对应的比特速率为$W_2 = W_1 \times (1 + h_{DSP}) = W_0 \times (r \times q + d_{CP})/(r \times q) \times (4096/m) \times (1 + h_{DSP})$，对应的波特率（即符号速率）为$W_s = W_2/8 = W_0 \times (r \times q + d_{CP})/(r \times q) \times (4096/m) \times (1 + h_{DSP})/8$ GBaud。典型的，考虑 DSP 成帧的冗余为h_{DSP}为$1/28$，或$1/56$。可看出，表 1-4 中给出的参数组合，在考虑第一数据处理的输入数据（即所述第一比特数据）为 FlexO-4e 或 400G ZR 帧数据下，对应的波特率均为小于 100GBaud。可理解，所提出的数据处理方案可较好兼容现有 400G 数据中心、城域网络。

下面，结合上述对第一数据处理和第二数据处理的介绍，给出几个具体实施例。

实施例 1：

考虑第一数据处理的输入数据（即所述第一比特数据）为 FlexO-4e 或 400G ZR 帧数据，其标称速率（nominal bit rate）为$W_0 \approx 402.489753310$ Gbit/s。需要说明的是，本申请实施例中的非整数仅保留小数点后 9 位，在一些具体应用中，描述所述非整数可以采用其他形式描述，如保留小数点后 2 位等等，其描述方式为本领域普通技术人员可知，此处不再赘述。第一比特数据包含 r 行$q = 2056$列共$d_{in} = q \times r$比特。经过第一数据处理后对应的比特速率为$W_0 \times (r \times q + d_{CP})/(r \times q)$ Gbit/s，经过第二数据处理中的第一子处理（即 PCS、交织和编码处理）后对应的比特速率为$W_1 = W_0 \times (r \times q + d_{CP})/(r \times q) \times (4096/m)$。

在一些具体应用中，考虑第二数据处理中 DSP 成帧的冗余为$h_{DSP} = 1/28$，每个 DSP 帧（也称为超帧或多帧）包含 178176 个双偏振符号，其中每个 DSP 帧中的 172032 个双偏振符号（即成帧前符号）为第四比特数据经过符号映射和偏振划分得到。此时，经过第二数据处理后对应的比特速率为$W_{2-28} = W_1 \times$

$(1+h_{DSP})=W_0\times(r\times q+d_{CP})/(r\times q)\times(4096/m)\times(29/28)$，对应的波特率（即符号速率）为 $W_{s-28}=W_{2-28}/8=W_0\times(r\times q+d_{CP})/(r\times q)\times(4096/m)\times(29/28)/8$ GBaud。

下面表 5 给出了对应的{d_{in}、r、m、d_{CP}、v/16、W_1、W_{2-28}、W_{s-28}}参数组合。

表 5

序号	d_{in}	r	m	d_{CP}	v/16	W_1 (Gbit/s)	W_{2-28} (Gbit/s)	W_{s-28} (GBaud)
1	1186312	577	3536	1784	127	466.933604255	483.609804407	60.451225551
2	1184256	576	3536	3840	127	467.744252873	484.449404762	60.556175595
3	1182200	575	3520	520	126	468.557721139	485.291925465	60.661490683
4	1180144	574	3520	2576	126	469.374023789	486.137381782	60.767172723
5	1176032	572	3504	1312	125	471.015191705	487.837162837	60.979645355
6	1173976	571	3504	3368	125	471.840086961	488.691518639	61.086439830
7	1171920	570	3488	48	124	472.667876588	489.548872180	61.193609023
8	1169864	569	3488	2104	124	473.498575844	490.409239267	61.301154908
9	1165752	567	3472	840	123	475.168764824	492.139077853	61.517384732
10	1163696	566	3472	2896	123	476.008285609	493.008581524	61.626072691
11	1159584	564	3456	1632	122	477.696258254	494.756838905	61.844604863
12	1157528	563	3456	3688	122	478.544741838	495.635625475	61.954453184
13	1155472	562	3440	368	121	479.396244938	496.517539400	62.064692425
14	1153416	561	3440	2424	121	480.250783699	497.402597402	62.175324675
15	1149304	559	3424	1160	120	481.969033372	499.182213135	62.397776642
16	1147248	558	3424	3216	120	482.832777159	500.076804915	62.509600614
17	1143136	556	3408	1952	119	484.569585710	501.875642343	62.734455293
18	1141080	555	3408	4008	119	485.442684063	502.779922780	62.847490347
19	1139024	554	3392	688	118	486.318934395	503.687467767	62.960933471
20	1136968	553	3392	2744	118	487.198353806	504.598295014	63.074786877
21	1132856	551	3376	1480	117	488.966768884	506.429867773	63.303733472
22	1130800	550	3376	3536	117	489.855799373	507.350649350	63.418831169
23	1128744	549	3360	216	116	490.748068588	508.274785324	63.534348165
24	1122576	546	3344	1008	115	493.444486548	511.067503924	63.883437991
25	1120520	545	3344	3064	115	494.349889275	512.005242464	64.000655308
26	1116408	543	3328	1800	114	496.170699180	513.891081294	64.236385162
27	1114352	542	3328	3856	114	497.086143275	514.839219820	64.354902478
28	1112296	541	3312	536	113	498.004971636	515.790863480	64.473857935
29	1110240	540	3312	2592	113	498.927203065	516.746031746	64.593253968
30	1106128	538	3296	1328	112	500.781951032	518.667020711	64.833377589
31	1104072	537	3296	3384	112	501.714505875	519.632881085	64.954110136
32	1102016	536	3280	64	111	502.650540401	520.602345415	65.075293177
33	1099960	535	3280	2120	111	503.590074121	521.575433911	65.196929239
34	1095848	533	3264	856	110	505.479717927	523.532564996	65.441570624
35	1093792	532	3264	2912	110	506.429867773	524.516648764	65.564581096
36	1089680	530	3248	1648	109	508.340923877	526.495956873	65.811994609
37	1087624	529	3248	3704	109	509.301870803	527.491223332	65.936402917
38	1085568	528	3232	384	108	510.266457680	528.490259740	66.061282467
39	1083512	527	3232	2440	108	511.234705228	529.493087557	66.186635945
40	1079400	525	3216	1176	107	513.182266009	531.510204081	66.438775510
41	1077344	524	3216	3232	107	514.161621479	532.524536532	66.565567066
42	1071176	521	3200	4024	106	517.122245019	535.590896627	66.948862078

43	1069120	520	3184	704	105	518.116710875	536.620879120	67.077609890
44	1067064	519	3184	2760	105	519.115008969	537.654830718	67.206853840
45	1062952	517	3168	1496	104	521.123190822	539.734733351	67.466841669
46	1060896	516	3168	3552	104	522.133119486	540.780730897	67.597591362
47	1058840	515	3152	232	103	523.146970204	541.830790568	67.728848821
48	1056784	514	3152	2288	103	524.164765866	542.884936075	67.860617009
49	1052672	512	3136	1024	102	526.212284482	545.005580357	68.125697545
50	1050616	511	3136	3080	102	527.242054119	546.072127481	68.259015935
51	1046504	509	3120	1816	101	529.313732132	548.217793993	68.527224249
52	1044448	508	3120	3872	101	530.355688297	549.296962879	68.662120360
53	1042392	507	3104	552	100	531.401754744	550.380388842	68.797548605
54	1040336	506	3104	2608	100	532.451955840	551.468097120	68.933512140
55	1036224	504	3088	1344	99	534.564860427	553.656462585	69.207057823
56	1034168	503	3088	3400	99	535.627613628	554.757171258	69.344646407
57	1032112	502	3072	80	98	536.694600906	555.862265224	69.482783153
58	1030056	501	3072	2136	98	537.765847615	556.971770744	69.621471343
59	1025944	499	3056	872	97	539.921221753	559.204122530	69.900515316
60	1023888	498	3056	2928	97	541.005400914	560.327022375	70.040877797
61	1019776	496	3040	1664	96	543.186874304	562.586405530	70.323300691
62	1017720	495	3040	3720	96	544.284221525	563.722943723	70.465367965
63	1015664	494	3024	400	95	545.386011447	564.864083285	70.608010411
64	1013608	493	3024	2456	95	546.492271105	566.009852216	70.751231527
65	1009496	491	3008	1192	94	548.718308870	568.315391329	71.039423916
66	1007440	490	3008	3248	94	549.838142153	569.475218658	71.184402332
67	1003328	488	2992	1984	93	552.091577162	571.809133489	71.476141686
68	1001272	487	2992	4040	93	553.225235431	572.983279554	71.622909944
69	999216	486	2976	720	92	554.363558961	574.162257495	71.770282187
70	997160	485	2976	2776	92	555.506576608	575.346097201	71.918262150
71	993048	483	2960	1512	91	557.806810880	577.728482697	72.216060337
72	990992	482	2960	3568	91	558.964086421	578.927089508	72.365886188
73	988936	481	2944	248	90	560.126173919	580.130680130	72.516335016
74	986880	480	2944	2304	90	561.293103448	581.339285714	72.667410714
75	982768	478	2928	1040	89	563.641610157	583.771667662	72.971458458
76	980712	477	2928	3096	89	564.823248753	584.995507637	73.124438455
77	976600	475	2912	1832	88	567.201451905	587.458646616	73.432330827
78	974544	474	2912	3888	88	568.398079441	588.698010849	73.587251356
79	972488	473	2896	568	87	569.599766712	589.942615524	73.742826940
80	970432	472	2896	2624	87	570.806545879	591.192493946	73.899061743
81	966320	470	2880	1360	86	573.235509904	593.708206686	74.213525836
82	964264	469	2880	3416	86	574.457760458	594.974109046	74.371763631
83	962208	468	2864	96	85	575.685234306	596.245421245	74.530677656
84	960152	467	2864	2152	85	576.917965000	597.522178036	74.690272254
85	956040	465	2848	888	84	579.399332591	600.092165898	75.011520737
86	953984	464	2848	2944	84	580.648038050	601.385467980	75.173183497
87	949872	462	2832	1680	83	583.161665920	603.988868274	75.498608534
88	947816	461	2832	3736	83	584.426658688	605.299039355	75.662379919

89	945760	460	2816	416	82	585.697151424	606.614906832	75.826863354
90	943704	459	2816	2472	82	586.973180076	607.936507936	75.992063492
91	939592	457	2800	1208	81	589.541990492	610.597061581	76.324632698
92	937536	456	2800	3264	81	590.834845735	611.936090225	76.492011278
93	933424	454	2784	2000	80	593.437642412	614.631843927	76.828980491
94	931368	453	2784	4056	80	594.747659283	615.988647114	76.998580889
95	929312	452	2768	736	79	596.063472688	617.351453855	77.168931732
96	927256	451	2768	2792	79	597.385121186	618.720304086	77.340038011
97	923144	449	2752	1528	78	600.046079410	621.476296532	77.684537066
98	921088	448	2752	3584	78	601.385467980	622.863520408	77.857940051
99	919032	447	2736	264	77	602.730849340	624.256951102	78.032118888
100	916976	446	2736	2320	77	604.082263800	625.656630365	78.207078796
101	912864	444	2720	1056	76	606.803355079	628.474903474	78.559362934
102	910808	443	2720	3112	76	608.173114345	629.893582715	78.736697839
103	906696	441	2704	1848	75	610.931269059	632.750242954	79.093780369
104	904640	440	2704	3904	75	612.319749216	634.188311688	79.273538961
105	902584	439	2688	584	74	613.714555023	635.632931988	79.454116498
106	900528	438	2688	2640	74	615.115729806	637.084148728	79.635518591
107	896416	436	2672	1376	73	617.937361594	640.006553079	80.000819135
108	894360	435	2672	3432	73	619.357907253	641.477832512	80.184729064
109	892304	434	2656	112	72	620.784999205	642.955892034	80.369486504
110	886136	431	2640	904	71	625.106008480	647.431223069	80.928902884
111	884080	430	2640	2960	71	626.559743384	648.936877076	81.117109634
112	879968	428	2624	1696	70	629.487592652	651.969292389	81.496161549
113	877912	427	2624	3752	70	630.961802471	653.496152559	81.687019070
114	875856	426	2608	432	69	632.442933462	655.030181086	81.878772636
115	873800	425	2608	2488	69	633.931034482	656.571428571	82.071428571
116	869688	423	2592	1224	68	636.928344338	659.675785207	82.459473151
117	867632	422	2592	3280	68	638.437653211	661.238997968	82.654874746
118	863520	420	2576	2016	67	641.477832512	664.387755102	83.048469388
119	861464	419	2576	4072	67	643.008805859	665.973406068	83.246675759
120	859408	418	2560	752	66	644.547104438	667.566643882	83.445830485
121	857352	417	2560	2808	66	646.092780947	669.167523124	83.645940390
122	853240	415	2544	1544	65	649.206481096	672.392426850	84.049053356
123	851184	414	2544	3600	65	650.774612693	674.016563147	84.252070393
124	849128	413	2528	280	64	652.350338148	675.648564510	84.456070564
125	847072	412	2528	2336	64	653.933712755	677.288488210	84.661061026
126	842960	410	2512	1072	63	657.123633305	680.592334494	85.074041812
127	840904	409	2512	3128	63	658.730292555	682.256374432	85.282046804
128	836792	407	2496	1864	62	661.967296450	685.608985608	85.701123201
129	834736	406	2496	3920	62	663.597757771	687.297677691	85.912209711
130	832680	405	2480	600	61	665.236270753	688.994708994	86.124338624
131	830624	404	2480	2656	61	666.882895186	690.700141442	86.337517680
132	826512	402	2464	1392	60	670.200720535	694.136460554	86.767057569
133	824456	401	2464	3448	60	671.872044027	695.867474171	86.983434271
134	822400	400	2448	128	59	673.551724137	697.607142857	87.200892857

135	820344	399	2448	2184	59	675.239823697	699.355531686	87.419441461
136	816232	397	2432	920	58	678.641535655	702.878733357	87.859841670
137	814176	396	2432	2976	58	680.355276907	704.653679653	88.081709957
138	810064	394	2416	1712	57	683.808856992	708.230601885	88.528825236
139	808008	393	2416	3768	57	685.548828639	710.032715376	88.754089422
140	805952	392	2400	448	56	687.297677691	711.844023323	88.980502915
141	803896	391	2400	2504	56	689.055472263	713.664596273	89.208074534
142	799784	389	2384	1240	55	692.598173920	717.333822989	89.666727874
143	797728	388	2384	3296	55	694.383220760	719.182621502	89.897827688
144	793616	386	2368	2032	54	697.981061282	722.908956328	90.363619541
145	791560	385	2368	4088	54	699.793999104	724.786641929	90.598330241
146	789504	384	2352	768	53	701.616379310	726.674107142	90.834263393
147	787448	383	2352	2824	53	703.448275862	728.571428571	91.071428571
148	783336	381	2336	1560	52	707.140917730	732.395950506	91.549493813
149	781280	380	2336	3616	52	709.001814882	734.323308270	91.790413534
150	779224	379	2320	296	51	710.872532071	736.260836788	92.032604599
151	777168	378	2320	2352	51	712.753147235	738.208616780	92.276077097
152	773056	376	2304	1088	50	716.544387380	742.135258358	92.766907295
153	771000	375	2304	3144	50	718.455172413	744.114285714	93.014285714
154	766888	373	2288	1880	49	722.307478968	748.104174645	93.513021831
155	764832	372	2288	3936	49	724.249165739	750.115207373	93.764400922
156	762776	371	2272	616	48	726.201319825	752.137081247	94.017135156
157	760720	370	2272	2672	48	728.164026095	754.169884169	94.271235521
158	756608	368	2256	1408	47	732.121439280	758.268633540	94.783579192
159	754552	367	2256	3464	47	734.116320586	760.334760607	95.041845076
160	752496	366	2240	144	46	736.122102882	762.412177985	95.301522248
161	750440	365	2240	2200	46	738.138875767	764.500978473	95.562622309
162	746328	363	2224	936	45	742.205756625	768.713105076	96.089138135
163	744272	362	2224	2992	45	744.256048771	770.836621941	96.354577743
164	740160	360	2208	1728	44	748.390804597	775.119047618	96.889880952
165	738104	359	2208	3784	44	750.475458649	777.278153601	97.159769200
166	736048	358	2192	464	43	752.571758813	779.449321628	97.431165203
167	733992	357	2192	2520	43	754.679802955	781.632653061	97.704081633
168	729880	355	2176	1256	42	758.931520155	786.036217303	98.254527163
169	727824	354	2176	3312	42	761.075394506	788.256658595	98.532082324
170	723712	352	2160	2048	41	765.399686520	792.735389610	99.091923701
171	721656	351	2160	4104	41	767.580312407	794.993894993	99.374236874
172	719600	350	2144	784	40	769.773399014	797.265306122	99.658163265

需要说明的是，$v = m - 1504$是 16 的整数倍。在一些具体实现方式中，考虑$n_0 = 128$，有$k_0 = v/16$，也就是所述 PCS 处理采用 16 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/16$个比特进行 PCS 子处理得到输出的$n_0 = 128$个比特。在一些具体应用中，16 个 PCS 子处理采用 16 个查找表 LUT 来实现。在另一些具体应用中，16 个 PCS 子处理采用 16 个网格编码（trellis coding）来实现。需要说明的是，在另一些具体实施例中，考虑$v = m - 1504$是 32 的整数倍，当$n_0 = 64$时，$k_0 = v/32$，也就是所述 PCS 处理采用 32 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/32$个比特进行 PCS 子处理得到输出的$n_0 = 64$个比特。在一些具体应用中，32 个 PCS 子处理采用 32 个查找表 LUT 来实现。在另一些具体应用中，32 个 PCS 子处理采用 32 个网格编码（trellis coding）来实现。

可看出，本实施例 1 中给出的参数组合，在考虑第一数据处理的输入数据（即所述第一比特数据）为 FlexO-4e 或 400G ZR 帧数据下，对应的波特率均为小于 100GBaud；此时，在对应具体光通信网络传输中，可采用的信道间隔不大于 112.5GHz。可理解，所提出的数据处理方案可较好兼容现有 400G 数据中心、城域网络。而且，共包含$d_{in}=q\times r$个比特的第一比特数据经过所述第一数据处理和第二数据处理，刚好得到一个 DSP 帧（也称为超帧或多帧），使得所提出的数据处理方法具有较低实现复杂度，功耗更低。

实施例 2：

在实施例 1 基础上，考虑所述第一约束和第二约束。

第一数据处理包含添加循环冗余校验 CRC-32 操作。CRC-32 编码及对应的 CRC 检测操作可直接采用现有 800G-ZR 中的 CRC-32 操作，便于实现简单。也就是，考虑所述第一约束如下：当行数 r 是 20 的整数倍时，有 CRC-32 操作次数$p=r/20$，$d_{CRC}=r/20\times 32\le d_{CP}$，也就是对r行数据中每20行共 41120 个比特进行 CRC-32 操作；当行数 r 不是 20 的整数倍时，有 CRC-32 操作次数$p=\lfloor r/20\rfloor+1$，$d_{CRC}=(\lfloor r/20\rfloor+1)\times 32\le d_{CP}$，其中$\lfloor a\rfloor$运算表示对 a 进行向下取整运算，也就是$r_0=20$，$r_1=r-20\times(p-1)$，对r行数据中最前面$20\times(p-1)$行中的每$r_0=20$行共$q\times r_0=41120$个比特进行 CRC-32，对最后的$r_1$行共$q\times r_1$个比特进行一次 CRC-32 操作。考虑$d_{PAD}=d_{CP}-d_{CRC}$，需选取合适的r、$d_{CP}$使得$d_{PAD}\ge 0$。

考虑包含 m 个比特的第一比特集合进行包含 PCS、交织和编码的第一子处理得到包含 4096 个比特的第二比特集合，其中编码操作中的编码输入比特长度和编码输出比特长度的比值为 111:128。考虑所述第一子处理（包含 PCS 处理、交织和编码）的输入比特长度 m 和输出比特长度 4096 的比值为 m:4096=f1:f2，其中整数 f1 和 f2 的最大公约数是 1。典型的，当 f1 和 f2 的取值越小，可使得整体硬件实现更简单，复杂度更低，功耗更低。为了便于硬件实现简单，考虑如下第二约束，即 m 为 32 的整数倍，使得所述第一子处理的输入比特长度 m 和输出比特长度的比值为 f:128，其中整数$f=m/32$。

本实施例 2 考虑第二数据处理中 DSP 成帧的冗余为$h_{DSP}=1/28$，每个 DSP 帧（也称为超帧或多帧）包含 178176 个双偏振符号，其中每个 DSP 帧中的 172032 个双偏振符号（即成帧前符号）为第四比特数据经过符号映射和偏振划分得到。此时，经过第二数据处理后对应的比特速率为$W_{2-28}=W_1\times(1+h_{DSP})=W_0\times(r\times q+d_{CP})/(r\times q)\times(4096/m)\times(29/28)$，对应的波特率（即符号速率）为$W_{s-28}=W_{2-28}/8=W_0\times(r\times q+d_{CP})/(r\times q)\times(4096/m)\times(29/28)/8$ GBaud。

下面表 6 给出了对应的{r、m、d_{CP}、d_{PAD}、f、p、v/16、W_1、W_{2-28}、W_{s-28}}参数组合。需要说明的是，表 6 还给出了，当行数 r 不是 20 的整数倍时，对应参数r_1的具体取值。对于行数 r 是 20 的整数倍时，如表 6 中的 28 行的 r=480，对应参数r_1在表中采用斜杠代表，表示并不存在所述参数r_1。

表 6

序号	r	m	d_{CP}	d_{PAD}	f	p	v/16	r_1	W_1 (Gbit/s)	W_{2-28} (Gbit/s)	W_{s-28} (GBaud)
1	574	3520	2576	1648	110	29	126	14	469.374023789	486.137381782	60.767172723
2	569	3488	2104	1176	109	29	124	9	473.498575844	490.409239267	61.301154908
3	564	3456	1632	704	108	29	122	4	477.696258254	494.756838905	61.844604863
4	563	3456	3688	2760	108	29	122	3	478.544741838	495.635625475	61.954453184
5	559	3424	1160	264	107	28	120	19	481.969033372	499.182213135	62.397776642
6	558	3424	3216	2320	107	28	120	18	482.832777159	500.076804915	62.509600614
7	553	3392	2744	1848	106	28	118	13	487.198353806	504.598295014	63.074786877
8	543	3328	1800	904	104	28	114	3	496.170699180	513.891081294	64.236385162
9	542	3328	3856	2960	104	28	114	2	497.086143275	514.839219820	64.354902478
10	538	3296	1328	464	103	27	112	18	500.781951032	518.667020711	64.833377589
11	537	3296	3384	2520	103	27	112	17	501.714505875	519.632881085	64.954110136
12	532	3264	2912	2048	102	27	110	12	506.429867773	524.516648764	65.564581096
13	527	3232	2440	1576	101	27	108	7	511.234705228	529.493087557	66.186635945
14	521	3200	4024	3160	100	27	106	1	517.122245019	535.590896627	66.948862078
15	517	3168	1496	664	99	26	104	17	521.123190822	539.734733351	67.466841669
16	516	3168	3552	2720	99	26	104	16	522.133119486	540.780730897	67.597591362

17	512	3136	1024	192	98	26	102	12	526.212284482	545.005580357	68.125697545
18	511	3136	3080	2248	98	26	102	11	527.242054119	546.072127481	68.259015935
19	506	3104	2608	1776	97	26	100	6	532.451955840	551.468097120	68.933512140
20	501	3072	2136	1304	96	26	98	1	537.765847615	556.971770744	69.621471343
21	496	3040	1664	864	95	25	96	16	543.186874304	562.586405530	70.323300691
22	495	3040	3720	2920	95	25	96	15	544.284221525	563.722943723	70.465367965
23	491	3008	1192	392	94	25	94	11	548.718308870	568.315391329	71.039423916
24	490	3008	3248	2448	94	25	94	10	549.838142153	569.475218658	71.184402332
25	485	2976	2776	1976	93	25	92	5	555.506576608	575.346097201	71.918262150
26	480	2944	2304	1536	92	24	90		561.293103448	581.339285714	72.667410714
27	475	2912	1832	1064	91	24	88	15	567.201451905	587.458646616	73.432330827
28	474	2912	3888	3120	91	24	88	14	568.398079441	588.698010849	73.587251356
29	470	2880	1360	592	90	24	86	10	573.235509904	593.708206686	74.213525836
30	469	2880	3416	2648	90	24	86	9	574.457760458	594.974109046	74.371763631
31	465	2848	888	120	89	24	84	5	579.399332591	600.092165898	75.011520737
32	464	2848	2944	2176	89	24	84	4	580.648038050	601.385467980	75.173183497
33	459	2816	2472	1736	88	23	82	19	586.973180076	607.936507936	75.992063492
34	454	2784	2000	1264	87	23	80	14	593.437642412	614.631843927	76.828980491
35	453	2784	4056	3320	87	23	80	13	594.747659283	615.988647114	76.998580889
36	449	2752	1528	792	86	23	78	9	600.046079410	621.476296532	77.684537066
37	448	2752	3584	2848	86	23	78	8	601.385467980	622.863520408	77.857940051
38	444	2720	1056	320	85	23	76	4	606.803355079	628.474903474	78.559362934
39	443	2720	3112	2376	85	23	76	3	608.173114345	629.893582715	78.736697839
40	438	2688	2640	1936	84	22	74	18	615.115729806	637.084148728	79.635518591
41	428	2624	1696	992	82	22	70	8	629.487592652	651.969292389	81.496161549
42	427	2624	3752	3048	82	22	70	7	630.961802471	653.496152559	81.687019070
43	423	2592	1224	520	81	22	68	3	636.928344338	659.675785207	82.459473151
44	422	2592	3280	2576	81	22	68	2	638.437653211	661.238997968	82.654874746
45	418	2560	752	80	80	21	66	18	644.547104438	667.566643882	83.445830485
46	417	2560	2808	2136	80	21	66	17	646.092780947	669.167523124	83.645940390
47	412	2528	2336	1664	79	21	64	12	653.933712755	677.288488210	84.661061026
48	407	2496	1864	1192	78	21	62	7	661.967296450	685.608985608	85.701123201
49	406	2496	3920	3248	78	21	62	6	663.597757771	687.297677691	85.912209711
50	402	2464	1392	720	77	21	60	2	670.200720535	694.136460554	86.767057569
51	401	2464	3448	2776	77	21	60	1	671.872044027	695.867474171	86.983434271
52	397	2432	920	280	76	20	58	17	678.641535655	702.878733357	87.859841670
53	396	2432	2976	2336	76	20	58	16	680.355276907	704.653679653	88.081709957
54	391	2400	2504	1864	75	20	56	11	689.055472263	713.664596273	89.208074534
55	386	2368	2032	1392	74	20	54	6	697.981061282	722.908956328	90.363619541
56	385	2368	4088	3448	74	20	54	5	699.793999104	724.786641929	90.598330241
57	381	2336	1560	920	73	20	52	1	707.140917730	732.395950506	91.549493813
58	380	2336	3616	3008	73	19	52		709.001814882	734.323308270	91.790413534
59	376	2304	1088	480	72	19	50	16	716.544387380	742.135258358	92.766907295
60	375	2304	3144	2536	72	19	50	15	718.455172413	744.114285714	93.014285714
61	371	2272	616	8	71	19	48	11	726.201319825	752.137081247	94.017135156
62	370	2272	2672	2064	71	19	48	10	728.164026095	754.169884169	94.271235521

63	365	2240	2200	1592	70	19	46	5	738.138875767	764.500978473	95.562622309
64	360	2208	1728	1152	69	18	44		748.390804597	775.119047618	96.889880952
65	359	2208	3784	3208	69	18	44	19	750.475458649	777.278153601	97.159769200
66	355	2176	1256	680	68	18	42	15	758.931520155	786.036217303	98.254527163
67	354	2176	3312	2736	68	18	42	14	761.075394506	788.256658595	98.532082324
68	350	2144	784	208	67	18	40	10	769.773399014	797.265306122	99.658163265

需要说明的是，$v = m - 1504$是 16 的整数倍。在一些具体实现方式中，考虑$n_0 = 128$，有$k_0 = v/16$，也就是所述 PCS 处理采用 16 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/16$个比特进行 PCS 子处理得到输出的$n_0 = 128$个比特。在一些具体应用中，16 个 PCS 子处理采用 16 个查找表 LUT 来实现。在另一些具体应用中，16 个 PCS 子处理采用 16 个网格编码（trellis coding）来实现。需要说明的是，在另一些具体实施例中，考虑$v = m - 1504$是 32 的整数倍，当$n_0 = 64$时，$k_0 = v/32$，也就是所述 PCS 处理采用 32 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/32$个比特进行 PCS 子处理得到输出的$n_0 = 64$个比特。在一些具体应用中，32 个 PCS 子处理采用 32 个查找表 LUT 来实现。在另一些具体应用中，32 个 PCS 子处理采用 32 个网格编码（trellis coding）来实现。

可看出，本实施例 2 中给出的参数组合，在考虑第一数据处理的输入数据（即所述第一比特数据）为 FlexO-4e 或 400G ZR 帧数据下，对应的波特率均为小于 100GBaud；此时，在对应具体光通信网络传输中，可采用的信道间隔不大于 112.5GHz。可理解，所提出的数据处理方案可较好兼容现有 400G 数据中心、城域网络。而且，共包含$d_{in} = q \times r$个比特的第一比特数据经过所述第一数据处理和第二数据处理，刚好得到一个 DSP 帧（也称为超帧或多帧），使得所提出的数据处理方法具有较低实现复杂度，功耗更低。更进一步，本实施例 2 中第一数据处理包含的添加循环冗余校验 CRC-32 操作采用现有 800G-ZR 中的 CRC-32 操作，便于实现简单；而且，m 为 32 的整数倍，使得所述第一子处理的输入比特长度 m 和输出比特长度的比值为 f:128，其中整数$f = m/32$，使得硬件实现更简单。

以表 6 中的某一些参数组合为例，例如，第 19 行参数组合，即{r=506、m=3104、$d_{CP} = 2608$、$d_{PAD} = 1776$、$f = 97$、p=26、$r_1 = 6$、$W_1 = 532.451955840\ (Gbit/s)$、$W_{2-28} = 551.468097120\ (Gbit/s)$、$W_{s-28} = 68.933512140(GBaud)$}，其对应的添加循环冗余校验（Cyclic Redundancy Check，CRC）和/或填充（Pad）比特操作可参照图 6 进行理解，此时，$r_0 = 20$，图 6 中的 1776b 表示 1776 个比特。需要说明的是，在实际光传输应用中，信道间隔 C_{grid} GHz，其数值 C_{grid}（以 G 为单位）比波特率数值（以 G 为单位）小，且 C_{grid} 通常是数值 12.5 的整数倍。此时，考虑第 21 行参数组合对应的波特率为W_{s-28}=68.933512140(GBaud)，在对应具体光通信网络传输中，可采用的信道间隔为 75GHz。所提出的数据处理方案可较好兼容现有 400G 数据中心、城域网络，例如采用 75GHz 格点间隔（grid spacing）的 OpenZR+或 400ZR 或 400ZR+传输网络。

再例如，表 6 中的第 20 行参数组合，即{r=501、m=3072、$d_{CP} = 2136$、$d_{PAD} = 1304$、$f = 96$、p=26、$r_1 = 1$、$W_1 = 537.765847615\ (Gbit/s)$、$W_{2-28} = 556.971770744\ (Gbit/s)$、$W_{s-28} = 69.621471343\ (GBaud)$}，其对应的添加循环冗余校验（Cyclic Redundancy Check，CRC）和/或填充（Pad）比特操作可参照图 7 进行理解，此时，$r_0 = 20$，图 7 中的 1304b 表示 1304 个比特。此时，在对应具体光通信网络传输中，也可采用的信道间隔为 75GHz。所提出的数据处理方案可较好兼容现有 400G 数据中心、城域网络，例如采用 75GHz 格点间隔（grid spacing）的 OpenZR+或 400ZR 或 400ZR+传输网络。

进一步地，考虑表 6 中的第 47 行参数组合，即{r=412、m=2528、$d_{CP} = 2336$、$d_{PAD} = 1664$、$f = 79$、p=21、$r_1 = 12$、$W_1 = 653.933712755\ (Gbit/s)$、$W_{2-28} = 677.288488210\ (Gbit/s)$、$W_{s-28} = 84.661061026\ (GBaud)$}。此时，在对应具体光通信网络传输中，可采用的信道间隔为 100GHz。所提出的数据处理方案可较好兼容现有 400G 数据中心、城域网络，例如采用 100GHz 格点间隔（grid spacing）的 OpenZR+或 400ZR 或 400ZR+传输网络。

在另一些具体应用中，可采用表 6 中的第 48 行参数组合，即{r=407、m=2496、$d_{CP} = 1864$、$d_{PAD} = 1192$、$f = 78$、p=21、$r_1 = 7$、$W_1 = 661.967296450\ (Gbit/s)$、$W_{2-28} = 685.608985608\ (Gbit/s)$、$W_{s-28} = 85.701123201\ (GBaud)$}。此时，在对应具体光通信网络传输中，可采用的信道间隔为 100GHz。所提出的数据处理方案可较好兼容现有 400G 数据中心、城域网络，例如采用 100GHz 格点间隔（grid

spacing）的 OpenZR+或 400ZR 或 400ZR+传输网络。

在又一些具体应用中，可采用表 6 中的第 49 行参数组合，即{r=406、m=2496、$d_{CP}=3920$、$d_{PAD}=3248$、$f=78$、p=21、$r_1=6$、$W_1=663.597757771$ (Gbit/s)、$W_{2-28}=687.297677691$ (Gbit/s)、$W_{s-28}=85.912209711$ (GBaud)}。此时，在对应具体光通信网络传输中，可采用的信道间隔为 100GHz。所提出的数据处理方案可较好兼容现有 400G 数据中心、城域网络，例如采用 100GHz 格点间隔（grid spacing）的 OpenZR+或 400ZR 或 400ZR+传输网络。

再例如，考虑表 6 中的第 63 行参数组合，即{r=365、m=2240、$d_{CP}=2200$、$d_{PAD}=1592$、$f=70$、p=19、$r_1=5$、$W_1=738.138875767$ (Gbit/s)、$W_{2-28}=764.500978473$ (Gbit/s)、$W_{s-28}=95.562622309$ (GBaud)}。此时，在对应具体光通信网络传输中，可采用的信道间隔为 112.5GHz。所提出的数据处理方案可较好兼容现有 400G 数据中心、城域网络，例如采用 112.5GHz 格点间隔（grid spacing）的高性能城域传输网络。

在又一些具体应用中，可采用表 6 中的第 64 行参数组合，即{r=360、m=2208、$d_{CP}=1728$、$d_{PAD}=1152$、$f=69$、p=18、$W_1=748.390804597$(Gbit/s)、$W_{2-28}=775.119047618$ (Gbit/s)、$W_{s-28}=96.889880952$ (GBaud)}，其对应的添加循环冗余校验（Cyclic Redundancy Check，CRC）和/或填充（Pad）比特操作可参照图 8 进行理解，此时，r=360 能被 p 整除，$r/p=20$，图 8 中的 1152b 表示 1152 个比特。此时，在对应具体光通信网络传输中，可采用的信道间隔为 112.5GHz。所提出的数据处理方案可较好兼容现有 400G 数据中心、城域网络，例如采用 112.5GHz 格点间隔（grid spacing）的高性能城域传输网络。

实施例 3：

在实施例 1 基础上，考虑第二数据处理中 DSP 成帧的冗余为$h_{DSP}=1/56$，每个 DSP 帧（也称为超帧或多帧）包含 175104 个双偏振符号，其中每个 DSP 帧中的 172032 个双偏振符号（即成帧前符号）为第四比特数据经过符号映射和偏振划分得到。此时，经过第二数据处理后对应的比特速率为$W_{2-56}=W_1\times(1+h_{DSP})=W_0\times(r\times q+d_{CP})/(r\times q)\times(4096/m)\times(57/56)$，对应的波特率（即符号速率）为$W_{s-56}=W_{2-56}/8=W_0\times(r\times q+d_{CP})/(r\times q)\times(4096/m)\times(57/56)/8$ GBaud。

下面表 7 给出了对应的{d_{in}、r、m、d_{CP}、v/16、W_1、W_{2-56}、W_{s-56}}参数组合。

表 7

序号	d_{in}	r	m	d_{CP}	v/16	W_1 (Gbit/s)	W_{2-56}(Gbit/s)	W_{s-56} (GBaud)
1	1186312	577	3536	1784	127	466.933604255	475.271704331	59.408963041
2	1184256	576	3536	3840	127	467.744252873	476.096828817	59.512103602
3	1182200	575	3520	520	126	468.557721139	476.924823302	59.615602913
4	1180144	574	3520	2576	126	469.374023789	477.755702785	59.719462848
5	1176032	572	3504	1312	125	471.015191705	479.426177271	59.928272159
6	1173976	571	3504	3368	125	471.840086961	480.265802800	60.033225350
7	1171920	570	3488	48	124	472.667876588	481.108374384	60.138546798
8	1169864	569	3488	2104	124	473.498575844	481.953907555	60.244238444
9	1165752	567	3472	840	123	475.168764824	483.653921338	60.456740167
10	1163696	566	3472	2896	123	476.008285609	484.508433567	60.563554196
11	1159584	564	3456	1632	122	477.696258254	486.226548579	60.778318572
12	1157528	563	3456	3688	122	478.544741838	487.090183657	60.886272957
13	1155472	562	3440	368	121	479.396244938	487.956892169	60.994611521
14	1153416	561	3440	2424	121	480.250783699	488.826690550	61.103336319
15	1149304	559	3424	1160	120	481.969033372	490.575623254	61.321952907
16	1147248	558	3424	3216	120	482.832777159	491.454791037	61.431848880
17	1143136	556	3408	1952	119	484.569585710	493.222614027	61.652826753
18	1141080	555	3408	4008	119	485.442684063	494.111303421	61.763912928

19	1139024	554	3392	688	118	486.318934395	495.003201081	61.875400135
20	1136968	553	3392	2744	118	487.198353806	495.898324410	61.987290551
21	1132856	551	3376	1480	117	488.966768884	497.698318328	62.212289791
22	1130800	550	3376	3536	117	489.855799373	498.603224361	62.325403045
23	1128744	549	3360	216	116	490.748068588	499.511426956	62.438928369
24	1122576	546	3344	1008	115	493.444486548	502.255995236	62.781999404
25	1120520	545	3344	3064	115	494.349889275	503.177565869	62.897195734
26	1116408	543	3328	1800	114	496.170699180	505.030890237	63.128861280
27	1114352	542	3328	3856	114	497.086143275	505.962681548	63.245335193
28	1112296	541	3312	536	113	498.004971636	506.897917558	63.362239695
29	1110240	540	3312	2592	113	498.927203065	507.836617405	63.479577176
30	1106128	538	3296	1328	112	500.781951032	509.724485871	63.715560734
31	1104072	537	3296	3384	112	501.714505875	510.673693480	63.834211685
32	1102016	536	3280	64	111	502.650540401	511.626442908	63.953305364
33	1099960	535	3280	2120	111	503.590074121	512.582754016	64.072844252
34	1095848	533	3264	856	110	505.479717927	514.506141461	64.313267683
35	1093792	532	3264	2912	110	506.429867773	515.473258268	64.434157284
36	1089680	530	3248	1648	109	508.340923877	517.418440375	64.677305047
37	1087624	529	3248	3704	109	509.301870803	518.396547068	64.799568383
38	1085568	528	3232	384	108	510.266457680	519.378358710	64.922294839
39	1083512	527	3232	2440	108	511.234705228	520.363896392	65.045487049
40	1079400	525	3216	1176	107	513.182266009	522.346235045	65.293279381
41	1077344	524	3216	3232	107	514.161621479	523.343079005	65.417884876
42	1071176	521	3200	4024	106	517.122245019	526.356570823	65.794571353
43	1069120	520	3184	704	105	518.116710875	527.368794998	65.921099375
44	1067064	519	3184	2760	105	519.115008969	528.384919844	66.048114980
45	1062952	517	3168	1496	104	521.123190822	530.428962087	66.303620261
46	1060896	516	3168	3552	104	522.133119486	531.456925191	66.432115649
47	1058840	515	3152	232	103	523.146970204	532.488880386	66.561110048
48	1056784	514	3152	2288	103	524.164765866	533.524850970	66.690606371
49	1052672	512	3136	1024	102	526.212284482	535.608932420	66.951116552
50	1050616	511	3136	3080	102	527.242054119	536.657090800	67.082136350
51	1046504	509	3120	1816	101	529.313732132	538.765763063	67.345720383
52	1044448	508	3120	3872	101	530.355688297	539.826325588	67.478290699
53	1042392	507	3104	552	100	531.401754744	540.891071793	67.611383974
54	1040336	506	3104	2608	100	532.451955840	541.960026480	67.745003310
55	1036224	504	3088	1344	99	534.564860427	544.110661506	68.013832688
56	1034168	503	3088	3400	99	535.627613628	545.192392443	68.149049055
57	1032112	502	3072	80	98	536.694600906	546.278433065	68.284804133
58	1030056	501	3072	2136	98	537.765847615	547.368809179	68.421101147
59	1025944	499	3056	872	97	539.921221753	549.562672142	68.695334018
60	1023888	498	3056	2928	97	541.005400914	550.666211644	68.833276456
61	1019776	496	3040	1664	96	543.186874304	552.886639917	69.110829990
62	1017720	495	3040	3720	96	544.284221525	554.003582624	69.250447828
63	1015664	494	3024	400	95	545.386011447	555.125047366	69.390630921
64	1013608	493	3024	2456	95	546.492271105	556.251061661	69.531382708

65	1009496	491	3008	1192	94	548.718308870	558.516850099	69.814606262
66	1007440	490	3008	3248	94	549.838142153	559.656680406	69.957085051
67	1003328	488	2992	1984	93	552.091577162	561.950355325	70.243794416
68	1001272	487	2992	4040	93	553.225235431	563.104257492	70.388032187
69	999216	486	2976	720	92	554.363558961	564.262908228	70.532863529
70	997160	485	2976	2776	92	555.506576608	565.426336905	70.678292113
71	993048	483	2960	1512	91	557.806810880	567.767646788	70.970955849
72	990992	482	2960	3568	91	558.964086421	568.945587964	71.118198496
73	988936	481	2944	248	90	560.126173919	570.128427025	71.266053378
74	986880	480	2944	2304	90	561.293103448	571.316194581	71.414524323
75	982768	478	2928	1040	89	563.641610157	573.706638910	71.713329864
76	980712	477	2928	3096	89	564.823248753	574.909378195	71.863672274
77	976600	475	2912	1832	88	567.201451905	577.330049261	72.166256158
78	974544	474	2912	3888	88	568.398079441	578.548045145	72.318505643
79	972488	473	2896	568	87	569.599766712	579.771191118	72.471398890
80	970432	472	2896	2624	87	570.806545879	580.999519913	72.624939989
81	966320	470	2880	1360	86	573.235509904	583.471858295	72.933982287
82	964264	469	2880	3416	86	574.457760458	584.715934752	73.089491844
83	962208	468	2864	96	85	575.685234306	585.965327775	73.245665972
84	960152	467	2864	2152	85	576.917965000	587.220071518	73.402508940
85	956040	465	2848	888	84	579.399332591	589.745749245	73.718218656
86	953984	464	2848	2944	84	580.648038050	591.016753015	73.877094127
87	949872	462	2832	1680	83	583.161665920	593.575267097	74.196908387
88	947816	461	2832	3736	83	584.426658688	594.862849021	74.357856128
89	945760	460	2816	416	82	585.697151424	596.156029128	74.519503641
90	943704	459	2816	2472	82	586.973180076	597.454844006	74.681855501
91	939592	457	2800	1208	81	589.541990492	600.069526037	75.008690755
92	937536	456	2800	3264	81	590.834845735	601.385467980	75.173183497
93	933424	454	2784	2000	80	593.437642412	604.034743169	75.504342896
94	931368	453	2784	4056	80	594.747659283	605.368153198	75.671019150
95	929312	452	2768	736	79	596.063472688	606.707463272	75.838432909
96	927256	451	2768	2792	79	597.385121186	608.052712636	76.006589079
97	923144	449	2752	1528	78	600.046079410	610.761187971	76.345148496
98	921088	448	2752	3584	78	601.385467980	612.124494194	76.515561774
99	919032	447	2736	264	77	602.730849340	613.493900221	76.686737528
100	916976	446	2736	2320	77	604.082263800	614.869447083	76.858680885
101	912864	444	2720	1056	76	606.803355079	617.639129277	77.204891160
102	910808	443	2720	3112	76	608.173114345	619.033348530	77.379168566
103	906696	441	2704	1848	75	610.931269059	621.840756006	77.730094501
104	904640	440	2704	3904	75	612.319749216	623.254030452	77.906753806
105	902584	439	2688	584	74	613.714555023	624.673743505	78.084217938
106	900528	438	2688	2640	74	615.115729806	626.099939267	78.262492408
107	896416	436	2672	1376	73	617.937361594	628.971957337	78.621494667
108	894360	435	2672	3432	73	619.357907253	630.417869882	78.802233735
109	892304	434	2656	112	72	620.784999205	631.870445619	78.983805702
110	886136	431	2640	904	71	625.106008480	636.268615775	79.533576972

111	884080	430	2640	2960	71	626.559743384	637.748310230	79.718538779
112	879968	428	2624	1696	70	629.487592652	640.728442521	80.091055315
113	877912	427	2624	3752	70	630.961802471	642.228977515	80.278622189
114	875856	426	2608	432	69	632.442933462	643.736557274	80.467069659
115	873800	425	2608	2488	69	633.931034482	645.251231527	80.656403941
116	869688	423	2592	1224	68	636.928344338	648.302064773	81.037758097
117	867632	422	2592	3280	68	638.437653211	649.838325590	81.229790699
118	863520	420	2576	2016	67	641.477832512	652.932793807	81.616599226
119	861464	419	2576	4072	67	643.008805859	654.491105964	81.811388245
120	859408	418	2560	752	66	644.547104438	656.056874160	82.007109270
121	857352	417	2560	2808	66	646.092780947	657.630152036	82.203769004
122	853240	415	2544	1544	65	649.206481096	660.799453973	82.599931747
123	851184	414	2544	3600	65	650.774612693	662.395587920	82.799448490
124	849128	413	2528	280	64	652.350338148	663.999451329	82.999931416
125	847072	412	2528	2336	64	653.933712755	665.611100483	83.201387560
126	842960	410	2512	1072	63	657.123633305	668.857983900	83.607247987
127	840904	409	2512	3128	63	658.730292555	670.493333493	83.811666687
128	836792	407	2496	1864	62	661.967296450	673.788141029	84.223517629
129	834736	406	2496	3920	62	663.597757771	675.447717731	84.430964716
130	832680	405	2480	600	61	665.236270753	677.115489874	84.639436234
131	830624	404	2480	2656	61	666.882895186	678.791518314	84.848939789
132	826512	402	2464	1392	60	670.200720535	682.168590544	85.271073818
133	824456	401	2464	3448	60	671.872044027	683.869759099	85.483719887
134	822400	400	2448	128	59	673.551724137	685.579433497	85.697429187
135	820344	399	2448	2184	59	675.239823697	687.297677691	85.912209711
136	816232	397	2432	920	58	678.641535655	690.760134506	86.345016813
137	814176	396	2432	2976	58	680.355276907	692.504478280	86.563059785
138	810064	394	2416	1712	57	683.808856992	696.019729439	87.002466180
139	808008	393	2416	3768	57	685.548828639	697.790772007	87.223846501
140	805952	392	2400	448	56	687.297677691	699.570850507	87.446356313
141	803896	391	2400	2504	56	689.055472263	701.360034268	87.670004284
142	799784	389	2384	1240	55	692.598173920	704.965998455	88.120749807
143	797728	388	2384	3296	55	694.383220760	706.782921131	88.347865141
144	793616	386	2368	2032	54	697.981061282	710.445008805	88.805626101
145	791560	385	2368	4088	54	699.793999104	712.290320516	89.036290065
146	789504	384	2352	768	53	701.616379310	714.145243226	89.268155403
147	787448	383	2352	2824	53	703.448275862	716.009852216	89.501231527
148	783336	381	2336	1560	52	707.140917730	719.768434118	89.971054265
149	781280	380	2336	3616	52	709.001814882	721.662561576	90.207820197
150	779224	379	2320	296	51	710.872532071	723.566684430	90.445835554
151	777168	378	2320	2352	51	712.753147235	725.480882007	90.685110251
152	773056	376	2304	1088	50	716.544387380	729.339822869	91.167477859
153	771000	375	2304	3144	50	718.455172413	731.284729064	91.410591133
154	766888	373	2288	1880	49	722.307478968	735.205826806	91.900728351
155	764832	372	2288	3936	49	724.249165739	737.182186556	92.147773319
156	762776	371	2272	616	48	726.201319825	739.169200536	92.396150067

157	760720	370	2272	2672	48	728.164026095	741.166955132	92.645869391
158	756608	368	2256	1408	47	732.121439280	745.195036410	93.149379551
159	754552	367	2256	3464	47	734.116320586	747.225540596	93.403192575
160	752496	366	2240	144	46	736.122102882	749.267140434	93.658392554
161	750440	365	2240	2200	46	738.138875767	751.319927120	93.914990890
162	746328	363	2224	936	45	742.205756625	755.459430851	94.432428856
163	744272	362	2224	2992	45	744.256048771	757.546335356	94.693291919
164	740160	360	2208	1728	44	748.390804597	761.754926108	95.219365763
165	738104	359	2208	3784	44	750.475458649	763.876806125	95.484600766
166	736048	358	2192	464	43	752.571758813	766.010540220	95.751317528
167	733992	357	2192	2520	43	754.679802955	768.156228008	96.019528501
168	729880	355	2176	1256	42	758.931520155	772.483868729	96.560483591
169	727824	354	2176	3312	42	761.075394506	774.666026550	96.833253319
170	723712	352	2160	2048	41	765.399686520	779.067538065	97.383442258
171	721656	351	2160	4104	41	767.580312407	781.287103700	97.660887963
172	719600	350	2144	784	40	769.773399014	783.519352568	97.939919071

需要说明的是，$v = m - 1504$是 16 的整数倍。在一些具体实现方式中，考虑$n_0 = 128$，有$k_0 = v/16$，也就是所述 PCS 处理采用 16 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/16$个比特进行 PCS 子处理得到输出的$n_0 = 128$个比特。在一些具体应用中，16 个 PCS 子处理采用 16 个查找表 LUT 来实现。在另一些具体应用中，16 个 PCS 子处理采用 16 个网格编码（trellis coding）来实现。需要说明的是，在另一些具体实施例中，考虑$v = m - 1504$是 32 的整数倍，当$n_0 = 64$时，$k_0 = v/32$，也就是所述 PCS 处理采用 32 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/32$个比特进行 PCS 子处理得到输出的$n_0 = 64$个比特。在一些具体应用中，32 个 PCS 子处理采用 32 个查找表 LUT 来实现。在另一些具体应用中，32 个 PCS 子处理采用 32 个网格编码（trellis coding）来实现。

可看出，本实施例 3 中给出的参数组合，在考虑第一数据处理的输入数据（即所述第一比特数据）为 FlexO-4e 或 400G-ZR 帧数据下，对应的波特率均为小于 100GBaud；此时，在对应具体光通信网络传输中，可采用的信道间隔不大于 112.5GHz。可理解，所提出的数据处理方案可较好兼容现有 400G 数据中心、城域网络。而且，共包含$d_{in} = q \times r$个比特的第一比特数据经过所述第一数据处理和第二数据处理，刚好得到一个 DSP 帧（也称为超帧或多帧），使得所提出的数据处理方法具有较低实现复杂度，功耗更低。

实施例 4：

在实施例 3 基础上，考虑所述第一约束和第二约束。

第一数据处理包含添加循环冗余校验 CRC-32 操作。CRC-32 编码及对应的 CRC 检测操作可直接采用现有 800G-ZR 中的 CRC-32 操作，便于实现简单。也就是，考虑所述第一约束如下：当行数 r 是 20 的整数倍时，有 CRC-32 操作次数$p = r/20$，$d_{CRC} = r/20 \times 32 \le d_{CP}$，也就是对r行数据中每20行共 41120 个比特进行 CRC-32 操作；当行数 r 不是 20 的整数倍时，有 CRC-32 操作次数$p = \lfloor r/20 \rfloor + 1$，$d_{CRC} = (\lfloor r/20 \rfloor + 1) \times 32 \le d_{CP}$，其中$\lfloor a \rfloor$运算表示对 a 进行向下取整运算，也就是$r_0 = 20$，$r_1 = r - 20 \times (p - 1)$，对r行数据中最前面$20 \times (p - 1)$行中的每$r_0 = 20$行共$q \times r_0 = 41120$个比特进行 CRC-32，对最后的$r_1$行共$q \times r_1$个比特进行一次 CRC-32 操作。考虑$d_{PAD} = d_{CP} - d_{CRC}$，需选取合适的r、$d_{CP}$使得$d_{PAD} \ge 0$。。

考虑包含 m 个比特的第一比特集合进行包含 PCS、交织和编码的第一子处理得到包含 4096 个比特的第二比特集合，其中编码操作中的编码输入比特长度和编码输出比特长度的比值为 111:128。考虑所述第一子处理（包含 PCS 处理、交织和编码）的输入比特长度 m 和输出比特长度 4096 的比值为 m:4096=f1:f2，其中整数 f1 和 f2 的最大公约数是 1。典型的，当 f1 和 f2 的取值越小，可使得整体硬件实现更简单，复杂度更低，功耗更低。为了便于硬件实现简单，考虑所述第二约束，即 m 为 32 的整数倍，使得所述第一子处理的输入比特长度 m 和输出比特长度的比值为 f:128，其中整数$f = m/32$。

本实施例 4 考虑第二数据处理中 DSP 成帧的冗余为$h_{DSP} = 56$，每个 DSP 帧（也称为超帧或多帧）包含 175104 个双偏振符号，其中每个 DSP 帧中的 172032 个双偏振符号（即成帧前符号）为第四比特数据

经过符号映射和偏振划分得到。此时，经过第二数据处理后对应的比特速率为$W_{2-56} = W_1 \times (1 + h_{DSP}) = W_0 \times (r \times q + d_{CP})/(r \times q) \times (4096/m) \times (57/56)$，对应的波特率（即符号速率）为$W_{s-56} = W_{2-56}/8 = W_0 \times (r \times q + d_{CP})/(r \times q) \times (4096/m) \times (57/56)/8$ GBaud。

下面表 8 给出了对应的{r、m、d_{CP}、d_{PAD}、f、p、W_1、W_{2-56}、W_{s-56}}参数组合。需要说明的是，表 8 还给出了，当行数 r 不是 20 的整数倍时，对应参数r_1的具体取值。对于行数 r 是 20 的整数倍时，如表 8 中的 28 行的 r=480，对应参数r_1在表中采用斜杠代表，表示并不存在所述参数r_1。

表 8

序号	r	m	d_{CP}	d_{PAD}	f	p	v/16	r_1	W_1 (Gbit/s)	W_{2-56} (Gbit/s)	W_{s-56} (GBaud)
1	574	3520	2576	1648	110	29	126	14	469.374023789	477.755702785	59.719462848
2	569	3488	2104	1176	109	29	124	9	473.498575844	481.953907555	60.244238444
3	564	3456	1632	704	108	29	122	4	477.696258254	486.226548579	60.778318572
4	563	3456	3688	2760	108	29	122	3	478.544741838	487.090183657	60.886272957
5	559	3424	1160	264	107	28	120	19	481.969033372	490.575623254	61.321952907
6	558	3424	3216	2320	107	28	120	18	482.832777159	491.454791037	61.431848880
7	553	3392	2744	1848	106	28	118	13	487.198353806	495.898324410	61.987290551
8	543	3328	1800	904	104	28	114	3	496.170699180	505.030890237	63.128861280
9	542	3328	3856	2960	104	28	114	2	497.086143275	505.962681548	63.245335193
10	538	3296	1328	464	103	27	112	18	500.781951032	509.724485871	63.715560734
11	537	3296	3384	2520	103	27	112	17	501.714505875	510.673693480	63.834211685
12	532	3264	2912	2048	102	27	110	12	506.429867773	515.473258268	64.434157284
13	527	3232	2440	1576	101	27	108	7	511.234705228	520.363896392	65.045487049
14	521	3200	4024	3160	100	27	106	1	517.122245019	526.356570823	65.794571353
15	517	3168	1496	664	99	26	104	17	521.123190822	530.428962087	66.303620261
16	516	3168	3552	2720	99	26	104	16	522.133119486	531.456925191	66.432115649
17	512	3136	1024	192	98	26	102	12	526.212284482	535.608932420	66.951116552
18	511	3136	3080	2248	98	26	102	11	527.242054119	536.657090800	67.082136350
19	501	3072	2136	1304	96	26	98	1	537.765847615	547.368809179	68.421101147
20	496	3040	1664	864	95	25	96	16	543.186874304	552.886639917	69.110829990
21	495	3040	3720	2920	95	25	96	15	544.284221525	554.003582624	69.250447828
22	491	3008	1192	392	94	25	94	11	548.718308870	558.516850099	69.814606262
23	490	3008	3248	2448	94	25	94	10	549.838142153	559.656680406	69.957085051
24	485	2976	2776	1976	93	25	92	5	555.506576608	565.426336905	70.678292113
25	480	2944	2304	1536	92	24	90	/	561.293103448	571.316194581	71.414524323
26	475	2912	1832	1064	91	24	88	15	567.201451905	577.330049261	72.166256158
27	474	2912	3888	3120	91	24	88	14	568.398079441	578.548045145	72.318505643
28	470	2880	1360	592	90	24	86	10	573.235509904	583.471858295	72.933982287
29	469	2880	3416	2648	90	24	86	9	574.457760458	584.715934752	73.089491844
30	465	2848	888	120	89	24	84	5	579.399332591	589.745749245	73.718218656
31	464	2848	2944	2176	89	24	84	4	580.648038050	591.016753015	73.877094127
32	459	2816	2472	1736	88	23	82	19	586.973180076	597.454844006	74.681855501
33	454	2784	2000	1264	87	23	80	14	593.437642412	604.034743169	75.504342896
34	453	2784	4056	3320	87	23	80	13	594.747659283	605.368153198	75.671019150
35	449	2752	1528	792	86	23	78	9	600.046079410	610.761187971	76.345148496
36	448	2752	3584	2848	86	23	78	8	601.385467980	612.124494194	76.515561774
37	444	2720	1056	320	85	23	76	4	606.803355079	617.639129277	77.204891160
38	443	2720	3112	2376	85	23	76	3	608.173114345	619.033348530	77.379168566
39	438	2688	2640	1936	84	22	74	18	615.115729806	626.099939267	78.262492408

40	433	2656	2168	1464	83	22	72	13	622.218682806	633.329730713	79.166216339
41	428	2624	1696	992	82	22	70	8	629.487592652	640.728442521	80.091055315
42	427	2624	3752	3048	82	22	70	7	630.961802471	642.228977515	80.278622189
43	423	2592	1224	520	81	22	68	3	636.928344338	648.302064773	81.037758097
44	422	2592	3280	2576	81	22	68	2	638.437653211	649.838325590	81.229790699
45	418	2560	752	80	80	21	66	18	644.547104438	656.056874160	82.007109270
46	417	2560	2808	2136	80	21	66	17	646.092780947	657.630152036	82.203769004
47	412	2528	2336	1664	79	21	64	12	653.933712755	665.611100483	83.201387560
48	407	2496	1864	1192	78	21	62	7	661.967296450	673.788141029	84.223517629
49	406	2496	3920	3248	78	21	62	6	663.597757771	675.447717731	84.430964716
50	402	2464	1392	720	77	21	60	2	670.200720535	682.168590544	85.271073818
51	401	2464	3448	2776	77	21	60	1	671.872044027	683.869759099	85.483719887
52	397	2432	920	280	76	20	58	17	678.641535655	690.760134506	86.345016813
53	396	2432	2976	2336	76	20	58	16	680.355276907	692.504478280	86.563059785
54	391	2400	2504	1864	75	20	56	11	689.055472263	701.360034268	87.670004284
55	386	2368	2032	1392	74	20	54	6	697.981061282	710.445008805	88.805626101
56	385	2368	4088	3448	74	20	54	5	699.793999104	712.290320516	89.036290065
57	381	2336	1560	920	73	20	52	1	707.140917730	719.768434118	89.971054265
58	380	2336	3616	3008	73	19	52		709.001814882	721.662561576	90.207820197
59	376	2304	1088	480	72	19	50	16	716.544387380	729.339822869	91.167477859
60	375	2304	3144	2536	72	19	50	15	718.455172413	731.284729064	91.410591133
61	371	2272	616	8	71	19	48	11	726.201319825	739.169200536	92.396150067
62	370	2272	2672	2064	71	19	48	10	728.164026095	741.166955132	92.645869391
63	365	2240	2200	1592	70	19	46	5	738.138875767	751.319927120	93.914990890
64	360	2208	1728	1152	69	18	44		748.390804597	761.754926108	95.219365763
65	359	2208	3784	3208	69	18	44	19	750.475458649	763.876806125	95.484600766
66	355	2176	1256	680	68	18	42	15	758.931520155	772.483868729	96.560483591
67	354	2176	3312	2736	68	18	42	14	761.075394506	774.666026550	96.833253319
68	350	2144	784	208	67	18	40	10	769.773399014	783.519352568	97.939919071

需要说明的是，$v = m - 1504$是 16 的整数倍。在一些具体实现方式中，考虑$n_0 = 128$，有$k_0 = v/16$，也就是所述 PCS 处理采用 16 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/16$个比特进行 PCS 子处理得到输出的$n_0 = 128$个比特。在一些具体应用中，16 个 PCS 子处理采用 16 个查找表 LUT 来实现。在另一些具体应用中，16 个 PCS 子处理采用 16 个网格编码（trellis coding）来实现。需要说明的是，在另一些具体实施例中，考虑$v = m - 1504$是 32 的整数倍，当$n_0 = 64$时，$k_0 = v/32$，也就是所述 PCS 处理采用 32 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/32$个比特进行 PCS 子处理得到输出的$n_0 = 64$个比特。在一些具体应用中，32 个 PCS 子处理采用 32 个查找表 LUT 来实现。在另一些具体应用中，32 个 PCS 子处理采用 32 个网格编码（trellis coding）来实现。

可看出，本实施例 4 给出的参数组合，在考虑第一数据处理的输入数据（即所述第一比特数据）为 FlexO-4e 或 400G ZR 帧数据下，对应的波特率均为小于 100GBaud；此时，在对应具体光通信网络传输中，可采用的信道间隔不大于 112.5GHz。可理解，所提出的数据处理方案可较好兼容现有 400G 数据中心、城域网络。而且，共包含$d_{in} = q \times r$个比特的第一比特数据经过所述第一数据处理和第二数据处理，刚好得到一个 DSP 帧（也称为超帧或多帧），使得所提出的数据处理方法具有较低实现复杂度，功耗更低。更进一步，本实施例 2 中第一数据处理包含的添加循环冗余校验 CRC-32 操作采用现有 800G-ZR 中的 CRC-32 操作，便于实现简单；而且，m 为 32 的整数倍，使得所述第一子处理的输入比特长度 m 和输出比特长度的比值为 f:128，其中整数$f = m/32$，使得硬件实现更简单。

需要说明的是，在相同的{r、m、d_{CP}、d_{PAD}、f、p、W_1}参数组合下，由于本实施例 4 中采用的 DSP

成帧冗余$h_{DSP}=1/56$小于实施例 2 中的成帧冗余，第二数据处理后获得的比特速率W_{2-56}和对应的波特率W_{s-56}分别小于实施例 2 中的比特速率W_{2-28}和对应的波特率W_{s-28}。

以表 8 中的一些参数组合为例，例如，表 8 中的第 20 行参数组合，即{r=496、m=3040、$d_{CP}=1664$、$d_{PAD}=864$、$f=95$、p=25、$r_1=16$、$W_1=543.186874304$ (Gbit/s)、$W_{2-56}=552.886639917$ (Gbit/s)、$W_{s-56}=69.110829990$ (GBaud)}。此时，在对应具体光通信网络传输中，可所采用的信道间隔为 75GHz。所提出的数据处理方案可较好兼容现有 400G 数据中心、城域网络，例如采用 75GHz 格点间隔（grid spacing）的 OpenZR+或 400ZR 或 400ZR+传输网络。

进一步地，例如，表 8 中的第 50 行参数组合，即{r=402、m=2464、$d_{CP}=1392$、$d_{PAD}=720$、$f=77$、p=21、$r_1=2$、$W_1=670.200720535$ (Gbit/s)、$W_{2-56}=682.168590544$ (Gbit/s)、$W_{s-56}=85.271073818$ (GBaud)}。此时，在对应具体光通信网络传输中，可所采用的信道间隔为 100GHz。所提出的数据处理方案可较好兼容现有 400G 数据中心、城域网络，例如采用 100GHz 格点间隔（grid spacing）的 OpenZR+或 400ZR 或 400ZR+传输网络。

再例如，表 8 中的第 66 行参数组合，即{r=355、m=2176、$d_{CP}=1256$、$d_{PAD}=680$、$f=68$、p=18、$r_1=15$、$W_1=758.931520155$ (Gbit/s)、$W_{2-56}=772.483868729$ (Gbit/s)、$W_{s-56}=96.560483591$ (GBaud)}。此时，在对应具体光通信网络传输中，可所采用的信道间隔为 112.5GHz。所提出的数据处理方案可较好兼容现有 400G 数据中心、城域网络，例如采用 112.5GHz 格点间隔（grid spacing）的高性能城域传输网络。

上述第一方面考虑$q=2056$的情况，下面考虑第二方面$q=10280$情况。

第三比特数据的比特数量为$d_{scr}=q\times r+d_{CP}=L\times m=336\times m$。应理解，添加的 CRC 和/或填充比特的比特数量d_{CP}越大，所引入的冗余$OH_{CP}=d_{CP}/(q\times r)$越大。为了传输有效性，这里考虑$0\le d_{CP}<q/2=5140$，既能保证传输性能，又可以使冗余较低。下面表 11 给出了对应的{d_{in}、r、m、d_{CP}}参数组合。

表 11

序号	d_{in}	r	m	d_{CP}	v/16
1	236440	115	3520	520	126
2	234384	114	3488	48	124
3	232328	113	3472	4952	123
4	230272	112	3440	4480	121
5	228216	111	3408	4008	119
6	226160	110	3376	3536	117
7	224104	109	3344	3064	115
8	222048	108	3312	2592	113
9	219992	107	3280	2120	111
10	217936	106	3248	1648	109
11	215880	105	3216	1176	107
12	213824	104	3184	704	105
13	211768	103	3152	232	103
14	209712	102	3136	5136	102
15	207656	101	3104	4664	100
16	205600	100	3072	4192	98
17	203544	99	3040	3720	96
18	201488	98	3008	3248	94
19	199432	97	2976	2776	92
20	197376	96	2944	2304	90
21	195320	95	2912	1832	88
22	193264	94	2880	1360	86

23	191208	93	2848	888	84
24	189152	92	2816	416	82
25	185040	90	2768	4848	79
26	182984	89	2736	4376	77
27	180928	88	2704	3904	75
28	178872	87	2672	3432	73
29	176816	86	2640	2960	71
30	174760	85	2608	2488	69
31	172704	84	2576	2016	67
32	170648	83	2544	1544	65
33	168592	82	2512	1072	63
34	166536	81	2480	600	61
35	164480	80	2448	128	59
36	162424	79	2432	5032	58
37	160368	78	2400	4560	56
38	158312	77	2368	4088	54
39	156256	76	2336	3616	52
40	154200	75	2304	3144	50
41	152144	74	2272	2672	48
42	150088	73	2240	2200	46
43	148032	72	2208	1728	44
44	145976	71	2176	1256	42
45	143920	70	2144	784	40

表 11 还给出了{r、m、d_{CP}}参数组合下，对应的$v/16$的具体取值。需要说明的是，$v = m - 1504$是 16 的整数倍，当$n_0 = 128$时，$k_0 = v/16$，也就是所述 PCS 处理采用 16 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/16$个比特进行 PCS 子处理得到输出的$n_0 = 128$个比特。在一些具体应用中，16 个 PCS 子处理采用 16 个查找表 LUT 来实现。在另一些具体应用中，16 个 PCS 子处理采用 16 个网格编码（trellis coding）来实现。需要说明的是，在另一些具体实施例中，考虑$v = m - 1504$是 32 的整数倍，当$n_0 = 64$时，$k_0 = v/32$，也就是所述 PCS 处理采用 32 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/32$个比特进行 PCS 子处理得到输出的$n_0 = 64$个比特。在一些具体应用中，32 个 PCS 子处理采用 32 个查找表 LUT 来实现。在另一些具体应用中，32 个 PCS 子处理采用 32 个网格编码（trellis coding）来实现。

需要说明的是，在一些具体应用中第一数据处理包含添加循环冗余校验 CRC-32 操作。此时，考虑如下第三约束：当行数 r 是 4 的整数倍时，有 CRC-32 操作次数$p = r/4$，$d_{CRC} = r/4 \times 32 \le d_{CP}$，也就是对对r行数据中每4行共 4×10280=41120 个比特进行 CRC-32 操作；当行数 r 不是 4 的整数倍时，有 CRC-32 操作次数$p = \lfloor r/4 \rfloor + 1$，$d_{CRC} = (\lfloor r/4 \rfloor + 1) \times 32 \le d_{CP}$，其中$\lfloor a \rfloor$运算表示对 a 进行向下取整运算，也就是$r_0 = 4$，$r_1 = r - 4 \times (p - 1)$，对r行数据中最前面$4 \times (p - 1)$行中的每$r_0 = 4$行共$q \times r_0 = 41120$个比特进行 CRC-32，对最后的$r_1$行共$q \times r_1$个比特进行一次 CRC-32 操作。考虑$d_{PAD} = d_{CP} - d_{CRC}$。结合表 11 和上述第三约束，下面表 12 给出了对应的{r、m、d_{CP}、d_{PAD}、p}参数组合。需要说明的是，表 12 最后一列还给出了，当行数 r 不是 4 的整数倍时，对应参数r_1的具体取值。对于行数 r 是 4 的整数倍时，如表 12 中第 2 行的 r=112，对应参数r_1在表中采用斜杠代表，表示并不存在所述参数r_1。

表 12

序号	r	m	d_{CP}	d_{PAD}	p	v/16	r_1
1	113	3472	4952	4024	29	123	1
2	112	3440	4480	3584	28	121	
3	111	3408	4008	3112	28	119	3
4	110	3376	3536	2640	28	117	2

5	109	3344	3064	2168	28	115	1
6	108	3312	2592	1728	27	113	
7	107	3280	2120	1256	27	111	3
8	106	3248	1648	784	27	109	2
9	105	3216	1176	312	27	107	1
10	102	3136	5136	4304	26	102	2
11	101	3104	4664	3832	26	100	1
12	100	3072	4192	3392	25	98	
13	99	3040	3720	2920	25	96	3
14	98	3008	3248	2448	25	94	2
15	97	2976	2776	1976	25	92	1
16	96	2944	2304	1536	24	90	
17	95	2912	1832	1064	24	88	3
18	94	2880	1360	592	24	86	2
19	93	2848	888	120	24	84	1
20	90	2768	4848	4112	23	79	2
21	89	2736	4376	3640	23	77	1
22	88	2704	3904	3200	22	75	
23	87	2672	3432	2728	22	73	3
24	86	2640	2960	2256	22	71	2
25	85	2608	2488	1784	22	69	1
26	84	2576	2016	1344	21	67	
27	83	2544	1544	872	21	65	3
28	82	2512	1072	400	21	63	2
29	79	2432	5032	4392	20	58	3
30	78	2400	4560	3920	20	56	2
31	77	2368	4088	3448	20	54	1
32	76	2336	3616	3008	19	52	
33	75	2304	3144	2536	19	50	3
34	74	2272	2672	2064	19	48	2
35	73	2240	2200	1592	19	46	1
36	72	2208	1728	1152	18	44	
37	71	2176	1256	680	18	42	3
38	70	2144	784	208	18	40	2

需要说明的是，在另一些具体应用中，考虑包含 m 个比特的第一比特集合进行包含 PCS、交织和编码的第一子处理得到包含 4096 个比特的第二比特集合，其中编码操作中的编码输入比特长度和编码输出比特长度的比值为 111:128。为了便于硬件实现简单，考虑如下第四约束，即 m 为 32 的整数倍，使得所述第一子处理的输入比特长度 m 和输出比特长度的比值为 f:128，其中整数$f = m/32$。结合表 11 和上述第四约束，下面表 13 给出了对应的{r、m、d_{CP}、f}参数组合。

表 13

序号	r	m	d_{CP}	f	v/16
1	115	3520	520	110	126
2	114	3488	48	109	124
3	102	3136	5136	98	102
4	101	3104	4664	97	100
5	100	3072	4192	96	98
6	99	3040	3720	95	96
7	98	3008	3248	94	94

8	97	2976	2776	93	92
9	96	2944	2304	92	90
10	95	2912	1832	91	88
11	94	2880	1360	90	86
12	93	2848	888	89	84
13	92	2816	416	88	82
14	79	2432	5032	76	58
15	78	2400	4560	75	56
16	77	2368	4088	74	54
17	76	2336	3616	73	52
18	75	2304	3144	72	50
19	74	2272	2672	71	48
20	73	2240	2200	70	46
21	72	2208	1728	69	44
22	71	2176	1256	68	42
23	70	2144	784	67	40

表 13 还给出了{ r、m、d_{CP}、f }参数组合下，对应的v/16的具体取值。需要说明的是，$v = m - 1504$是 16 的整数倍，当$n_0 = 128$时，$k_0 = v/16$，也就是所述 PCS 处理采用 16 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/16$个比特进行 PCS 子处理得到输出的$n_0 = 128$个比特。在一些具体应用中，16 个 PCS 子处理采用 16 个查找表 LUT 来实现。在另一些具体应用中，16 个 PCS 子处理采用 16 个网格编码（trellis coding）来实现。需要说明的是，在另一些具体实施例中，考虑$v = m - 1504$是 32 的整数倍，当$n_0 = 64$时，$k_0 = v/32$，也就是所述 PCS 处理采用 32 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/32$个比特进行 PCS 子处理得到输出的$n_0 = 64$个比特。在一些具体应用中，32 个 PCS 子处理采用 32 个查找表 LUT 来实现。在另一些具体应用中，32 个 PCS 子处理采用 32 个网格编码（trellis coding）来实现。

需要说明的是，在又一些具体应用中，考虑上述第三约束和第四约束。此时，结合表 11 和上述第三约束和第四约束，下面表 14 给出了对应的{r、m、d_{CP}、d_{PAD}、f、p }参数组合。需要说明的是，表 14 最后一列还给出了，当行数 r 不是 4 的整数倍时，对应参数r_1的具体取值。对于行数 r 是 4 的整数倍时，如表 14 中第 2 行的 r=100，对应参数r_1在表中采用斜杠代表，表示并不存在所述参数r_1。

表 14

序号	r	m	d_{CP}	d_{PAD}	f	p	v/16	r_1
1	102	3136	5136	4304	98	26	102	2
2	101	3104	4664	3832	97	26	100	1
3	100	3072	4192	3392	96	25	98	
4	99	3040	3720	2920	95	25	96	3
5	98	3008	3248	2448	94	25	94	2
6	97	2976	2776	1976	93	25	92	1
7	96	2944	2304	1536	92	24	90	
8	95	2912	1832	1064	91	24	88	3
9	94	2880	1360	592	90	24	86	2
10	93	2848	888	120	89	24	84	1
11	79	2432	5032	4392	76	20	58	3
12	78	2400	4560	3920	75	20	56	2
13	77	2368	4088	3448	74	20	54	1
14	76	2336	3616	3008	73	19	52	
15	75	2304	3144	2536	72	19	50	3
16	74	2272	2672	2064	71	19	48	2
17	73	2240	2200	1592	70	19	46	1

18	72	2208	1728	1152	69	18	44	
19	71	2176	1256	680	68	18	42	3
20	70	2144	784	208	67	18	40	2

表 14 还给出了{ r、m、d_{CP}、d_{PAD}、f、p }参数组合下，对应的v/16的具体取值。需要说明的是，$v = m - 1504$是 16 的整数倍，当$n_0 = 128$时，$k_0 = v/16$，也就是所述 PCS 处理采用 16 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/16$个比特进行 PCS 子处理得到输出的$n_0 = 128$个比特。

需要说明的是，给定数据帧的比特速率为W_0 Gbit/s，经过第一数据处理后对应的比特速率为$W_0 \times (r \times q + d_{CP})/(r \times q)$ Gbit/s，经过第二数据处理中的第一子处理（即 PCS、交织和编码处理）后对应的比特速率为$W_1 = W_0 \times (r \times q + d_{CP})/(r \times q) \times (4096/m)$。考虑第二数据处理中 DSP 成帧的冗余为$h_{DSP}$，即经过第二数据处理后对应的比特速率为$W_2 = W_1 \times (1 + h_{DSP}) = W_0 \times (r \times q + d_{CP})/(r \times q) \times (4096/m) \times (1 + h_{DSP})$，对应的波特率（即符号速率）为$W_s = W_2/8 = W_0 \times (r \times q + d_{CP})/(r \times q) \times (4096/m) \times (1 + h_{DSP})/8$ GBaud。典型的，考虑 DSP 成帧的冗余为h_{DSP}为1/28，或1/56。可看出，表 1-4 中给出的参数组合，在考虑第一数据处理的输入数据（即所述第一比特数据）为 FlexO-4e 或 400G ZR 帧数据下，对应的波特率均为小于 100GBaud。可理解，所提出的数据处理方案可较好兼容现有 400G 数据中心、城域网络。

下面，结合上述对第一数据处理和第二数据处理的介绍，给出几个具体实施例。

实施例 5：

考虑第一数据处理的输入数据（即所述第一比特数据）为 FlexO-4e 或 400G ZR 帧数据，其标称速率（nominal bit rate）为$W_0 \approx 402.489753310$ Gbit/s。需要说明的是，本申请实施例中的非整数仅保留小数点后 9 位，在一些具体应用中，描述所述非整数可以采用其他形式描述，如保留小数点后 2 位等等，其描述方式为本领域普通技术人员可知，此处不再赘述。第一比特数据包含 r 行$q = 10280$列共$d_{in} = q \times r$比特。经过第一数据处理后对应的比特速率为$W_0 \times (r \times q + d_{CP})/(r \times q)$ Gbit/s，经过第二数据处理中的第一子处理（即 PCS、交织和编码处理）后对应的比特速率为$W_1 = W_0 \times (r \times q + d_{CP})/(r \times q) \times (4096/m)$。

在一些具体应用中，考虑第二数据处理中 DSP 成帧的冗余为$h_{DSP} = 1/28$，每个 DSP 帧（也称为超帧或多帧）包含 178176 个双偏振符号，其中每个 DSP 帧中的 172032 个双偏振符号（即成帧前符号）为第四比特数据经过符号映射和偏振划分得到。此时，经过第二数据处理后对应的比特速率为$W_{2-28} = W_1 \times (1 + h_{DSP}) = W_0 \times (r \times q + d_{CP})/(r \times q) \times (4096/m) \times (29/28)$，对应的波特率（即符号速率）为$W_{s-28} = W_{2-28}/8 = W_0 \times (r \times q + d_{CP})/(r \times q) \times (4096/m) \times (29/28)/8$ GBaud。

下面表 15 给出了对应的{d_{in}、r、m、d_{CP}、v/16、W_1、W_{2-28}、W_{s-28}}参数组合。

表 15

序号	d_{in}	r	m	d_{CP}	v/16	W_1 (Gbit/s)	W_{2-28} (Gbit/s)	W_{s-28} (GBaud)
1	236440	115	3520	520	126	468.557721139	485.291925465	60.661490683
2	234384	114	3488	48	124	472.667876588	489.548872180	61.193609023
3	232328	113	3472	4952	123	476.850778150	493.881163084	61.735145386
4	230272	112	3440	4480	121	481.108374384	498.290816326	62.286352041
5	228216	111	3408	4008	119	485.442684063	502.779922780	62.847490347
6	226160	110	3376	3536	117	489.855799373	507.350649350	63.418831169
7	224104	109	3344	3064	115	494.349889275	512.005242464	64.000655308
8	222048	108	3312	2592	113	498.927203065	516.746031746	64.593253968
9	219992	107	3280	2120	111	503.590074121	521.575433911	65.196929239
10	217936	106	3248	1648	109	508.340923877	526.495956873	65.811994609
11	215880	105	3216	1176	107	513.182266009	531.510204081	66.438775510
12	213824	104	3184	704	105	518.116710875	536.620879120	67.077609890
13	211768	103	3152	232	103	523.146970204	541.830790568	67.728848821
14	209712	102	3136	5136	102	528.275862069	547.142857142	68.392857143

15	207656	101	3104	4664	100	533.506316148	552.560113154	69.070014144
16	205600	100	3072	4192	98	538.841379310	558.085714285	69.760714286
17	203544	99	3040	3720	96	544.284221525	563.722943723	70.465367965
18	201488	98	3008	3248	94	549.838142153	569.475218658	71.184402332
19	199432	97	2976	2776	92	555.506576608	575.346097201	71.918262150
20	197376	96	2944	2304	90	561.293103448	581.339285714	72.667410714
21	195320	95	2912	1832	88	567.201451905	587.458646616	73.432330827
22	193264	94	2880	1360	86	573.235509904	593.708206686	74.213525836
23	191208	93	2848	888	84	579.399332591	600.092165898	75.011520737
24	189152	92	2816	416	82	585.697151424	606.614906832	75.826863354
25	185040	90	2768	4848	79	598.712643678	620.095238095	77.511904762
26	182984	89	2736	4376	77	605.439752034	627.062600321	78.382825040
27	180928	88	2704	3904	75	612.319749216	634.188311688	79.273538961
28	178872	87	2672	3432	73	619.357907253	641.477832512	80.184729064
29	176816	86	2640	2960	71	626.559743384	648.936877076	81.117109634
30	174760	85	2608	2488	69	633.931034482	656.571428571	82.071428571
31	172704	84	2576	2016	67	641.477832512	664.387755102	83.048469388
32	170648	83	2544	1544	65	649.206481096	672.392426850	84.049053356
33	168592	82	2512	1072	63	657.123633305	680.592334494	85.074041812
34	166536	81	2480	600	61	665.236270753	688.994708994	86.124338624
35	164480	80	2448	128	59	673.551724137	697.607142857	87.200892857
36	162424	79	2432	5032	58	682.077695329	706.437613019	88.304701627
37	160368	78	2400	4560	56	690.822281167	715.494505494	89.436813187
38	158312	77	2368	4088	54	699.793999104	724.786641929	90.598330241
39	156256	76	2336	3616	52	709.001814882	734.323308270	91.790413534
40	154200	75	2304	3144	50	718.455172413	744.114285714	93.014285714
41	152144	74	2272	2672	48	728.164026095	754.169884169	94.271235521
42	150088	73	2240	2200	46	738.138875767	764.500978473	95.562622309
43	148032	72	2208	1728	44	748.390804597	775.119047618	96.889880952
44	145976	71	2176	1256	42	758.931520155	786.036217303	98.254527163
45	143920	70	2144	784	40	769.773399014	797.265306122	99.658163265

需要说明的是，$v = m - 1504$是 16 的整数倍。在一些具体实现方式中，考虑$n_0 = 128$，有$k_0 = v/16$，也就是所述 PCS 处理采用 16 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/16$个比特进行 PCS 子处理得到输出的$n_0 = 128$个比特。在一些具体应用中，16 个 PCS 子处理采用 16 个查找表 LUT 来实现。在另一些具体应用中，16 个 PCS 子处理采用 16 个网格编码（trellis coding）来实现。需要说明的是，在另一些具体实施例中，考虑$v = m - 1504$是 32 的整数倍，当$n_0 = 64$时，$k_0 = v/32$，也就是所述 PCS 处理采用 32 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/32$个比特进行 PCS 子处理得到输出的$n_0 = 64$个比特。在一些具体应用中，32 个 PCS 子处理采用 32 个查找表 LUT 来实现。在另一些具体应用中，32 个 PCS 子处理采用 32 个网格编码（trellis coding）来实现。

可看出，本实施例中给出的参数组合，在考虑第一数据处理的输入数据（即所述第一比特数据）为 FlexO-4e 或 400G ZR 帧数据下，对应的波特率均为小于 100GBaud；此时，在对应具体光通信网络传输中，可采用的信道间隔不大于 112.5GHz。可理解，所提出的数据处理方案可较好兼容现有 400G 数据中心、城域网络。而且，共包含$d_{in} = q \times r$个比特的第一比特数据经过所述第一数据处理和第二数据处理，刚好得到一个 DSP 帧（也称为超帧或多帧），使得所提出的数据处理方法具有较低实现复杂度，功耗更低。

实施例 6：

在实施例 5 基础上，考虑所述第一约束和第二约束。

第一数据处理包含添加循环冗余校验 CRC-32 操作。CRC-32 编码及对应的 CRC 检测操作可直接采用现有 800G-ZR 中的 CRC-32 操作，便于实现简单。也就是，考虑所述第三约束如下：当行数 r 是 4 的整数倍时，有 CRC-32 操作次数$p = r/4$，$d_{CRC} = r/4 \times 32 \le d_{CP}$，也就是对对r行数据中每4行共 41120 个比特进行 CRC-32 操作；当行数 r 不是 4 的整数倍时，有 CRC-32 操作次数$p = \lfloor r/4 \rfloor + 1$，$d_{CRC} = (\lfloor r/4 \rfloor + 1) \times 32 \le d_{CP}$，其中$\lfloor a \rfloor$运算表示对 a 进行向下取整运算，也就是$r_0 = 4$，$r_1 = r - 4 \times (p-1)$，对r行数据中最前面$4 \times (p-1)$行中的每$r_0 = 4$行共$q \times r_0 = 41120$个比特进行 CRC-32，对最后的$r_1$行共$q \times r_1$个比特进行一次 CRC-32 操作。考虑$d_{PAD} = d_{CP} - d_{CRC}$。

考虑包含 m 个比特的第一比特集合进行包含 PCS、交织和编码的第一子处理得到包含 4096 个比特的第二比特集合，其中编码操作中的编码输入比特长度和编码输出比特长度的比值为 111:128。为了便于硬件实现简单，考虑第四约束，即 m 为 32 的整数倍，使得所述第一子处理的输入比特长度 m 和输出比特长度的比值为 f:128，其中整数$f = m/32$。

本实施例 12 考虑第二数据处理中 DSP 成帧的冗余为$h_{DSP} = 1/28$，每个 DSP 帧（也称为超帧或多帧）包含 178176 个双偏振符号，其中每个 DSP 帧中的 172032 个双偏振符号（即成帧前符号）为第四比特数据经过符号映射和偏振划分得到。此时，经过第二数据处理后对应的比特速率为$W_{2-28} = W_1 \times (1 + h_{DSP}) = W_0 \times (r \times q + d_{CP})/(r \times q) \times (4096/m) \times (29/28)$，对应的波特率（即符号速率）为$W_{s-28} = W_{2-28}/8 = W_0 \times (r \times q + d_{CP})/(r \times q) \times (4096/m) \times (29/28)/8$ GBaud。

下面表 16 给出了对应的{r、m、d_{CP}、d_{PAD}、f、p、v/16、W_1、W_{2-28}、W_{s-28}}参数组合。需要说明的是，表 16 还给出了，当行数 r 不是 4 的整数倍时，对应参数r_1的具体取值。对于行数 r 是 4 的整数倍时，如表 16 中第 3 行的 r=100，对应参数r_1在表中采用斜杠代表，表示并不存在所述参数r_1。

表 16

序号	r	m	d_{CP}	d_{PAD}	f	p	v/16	r_1	W_1 (Gbit/s)	W_{2-28} (Gbit/s)	W_{s-28} (GBaud)
1	102	3136	5136	4304	98	26	102	2	528.275862069	547.142857142	68.392857143
2	101	3104	4664	3832	97	26	100	1	533.506316148	552.560113154	69.070014144
3	100	3072	4192	3392	96	25	98		538.841379310	558.085714285	69.760714286
4	99	3040	3720	2920	95	25	96	3	544.284221525	563.722943723	70.465367965
5	98	3008	3248	2448	94	25	94	2	549.838142153	569.475218658	71.184402332
6	97	2976	2776	1976	93	25	92	1	555.506576608	575.346097201	71.918262150
7	96	2944	2304	1536	92	24	90		561.293103448	581.339285714	72.667410714
8	95	2912	1832	1064	91	24	88	3	567.201451905	587.458646616	73.432330827
9	94	2880	1360	592	90	24	86	2	573.235509904	593.708206686	74.213525836
10	93	2848	888	120	89	24	84	1	579.399332591	600.092165898	75.011520737
11	79	2432	5032	4392	76	20	58	3	682.077695329	706.437613019	88.304701627
12	78	2400	4560	3920	75	20	56	2	690.822281167	715.494505494	89.436813187
13	77	2368	4088	3448	74	20	54	1	699.793999104	724.786641929	90.598330241
14	76	2336	3616	3008	73	19	52		709.001814882	734.323308270	91.790413534
15	75	2304	3144	2536	72	19	50	3	718.455172413	744.114285714	93.014285714
16	74	2272	2672	2064	71	19	48	2	728.164026095	754.169884169	94.271235521
17	73	2240	2200	1592	70	19	46	1	738.138875767	764.500978473	95.562622309
18	72	2208	1728	1152	69	18	44		748.390804597	775.119047618	96.889880952
19	71	2176	1256	680	68	18	42	3	758.931520155	786.036217303	98.254527163
20	70	2144	784	208	67	18	40	2	769.773399014	797.265306122	99.658163265

需要说明的是，$v = m - 1504$是 16 的整数倍。在一些具体实现方式中，考虑$n_0 = 128$，有$k_0 = v/16$，也就是所述 PCS 处理采用 16 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/16$个比特进行 PCS 子处理得到输出的$n_0 = 128$个比特。在一些具体应用中，16 个 PCS 子处理采用 16 个查找表 LUT 来实现。在另一些具体应用中，16 个 PCS 子处理采用 16 个网格编码（trellis coding）来实现。需要说明的是，在另一些具体实施例中，考虑$v = m - 1504$是 32 的整数倍，当

$n_0 = 64$时，$k_0 = v/32$，也就是所述 PCS 处理采用 32 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/32$个比特进行 PCS 子处理得到输出的$n_0 = 64$个比特。在一些具体应用中，32 个 PCS 子处理采用 32 个查找表 LUT 来实现。在另一些具体应用中，32 个 PCS 子处理采用 32 个网格编码（trellis coding）来实现。

可看出，本实施例中给出的参数组合，在考虑第一数据处理的输入数据（即所述第一比特数据）为 FlexO-4e 或 400G ZR 帧数据下，对应的波特率均为小于 100GBaud；此时，在对应具体光通信网络传输中，可采用的信道间隔不大于 112.5GHz。可理解，所提出的数据处理方案可较好兼容现有 400G 数据中心、城域网络。而且，共包含$d_{in} = q \times r$个比特的第一比特数据经过所述第一数据处理和第二数据处理，刚好得到一个 DSP 帧（也称为超帧或多帧），使得所提出的数据处理方法具有较低实现复杂度，功耗更低。更进一步，本实施例 2 中第一数据处理包含的添加循环冗余校验 CRC-32 操作采用现有 800G-ZR 中的 CRC-32 操作，便于实现简单；而且，m 为 32 的整数倍，使得所述第一子处理的输入比特长度 m 和输出比特长度的比值为 f:128，其中整数$f = m/32$，使得硬件实现更简单。

以表 16 中的一些参数组合为例，例如，表 16 中的第 2 行参数组合，即{r=101、m=3104、$d_{CP} = 4664$、$d_{PAD} = 3832$、$f = 97$、p=26、$r_1 = 1$、$W_1 = 533.506316148\ (Gbit/s)$、$W_{2-28} = 552.560113154\ (Gbit/s)$、$W_{s-28} = 69.070014144(GBaud)$}。此时，在对应具体光通信网络传输中，可所采用的信道间隔为 75GHz。所提出的数据处理方案可较好兼容现有 400G 数据中心、城域网络，例如采用 75GHz 格点间隔（grid spacing）的 OpenZR+或 400ZR 或 400ZR+传输网络。

再例如，表 16 中的第 17 行参数组合，即{r=73、m=2240、$d_{CP} = 2200$、$d_{PAD} = 1592$、$f = 70$、p=19、$r_1 = 1$、$W_1 = 738.138875767\ (Gbit/s)$、$W_{2-28} = 764.500978473\ (Gbit/s)$、$W_{s-28} = 95.562622309\ (GBaud)$}。此时，在对应具体光通信网络传输中，可采用的信道间隔为 112.5GHz。所提出的数据处理方案可较好兼容现有 400G 数据中心、城域网络，例如采用 112.5GHz 格点间隔（grid spacing）的高性能城域传输网络。

实施例 7：

在实施例 5 基础上，考虑第二数据处理中 DSP 成帧的冗余为$h_{DSP} = 1/56$，每个 DSP 帧（也称为超帧或多帧）包含 175104 个双偏振符号，其中每个 DSP 帧中的 172032 个双偏振符号（即成帧前符号）为第四比特数据经过符号映射和偏振划分得到。此时，经过第二数据处理后对应的比特速率为$W_{2-56} = W_1 \times (1 + h_{DSP}) = W_0 \times (r \times q + d_{CP})/(r \times q) \times (4096/m) \times (57/56)$，对应的波特率（即符号速率）为$W_{s-56} = W_{2-56}/8 = W_0 \times (r \times q + d_{CP})/(r \times q) \times (4096/m) \times (57/56)/8$ GBaud。

下面表 17 给出了对应的{d_{in}、r、m、d_{CP}、v/16、W_1、W_{2-56}、W_{s-56}}参数组合。

表 17

序号	d_{in}	r	m	d_{CP}	v/16	W_1 (Gbit/s)	W_{2-56}(Gbit/s)	W_{s-56} (GBaud)
1	236440	115	3520	520	126	468.557721139	476.924823302	59.615602913
2	234384	114	3488	48	124	472.667876588	481.108374384	60.138546798
3	232328	113	3472	4952	123	476.850778150	485.365970617	60.670746327
4	230272	112	3440	4480	121	481.108374384	489.699595355	61.212449419
5	228216	111	3408	4008	119	485.442684063	494.111303421	61.763912928
6	226160	110	3376	3536	117	489.855799373	498.603224361	62.325403045
7	224104	109	3344	3064	115	494.349889275	503.177565869	62.897195734
8	222048	108	3312	2592	113	498.927203065	507.836617405	63.479577176
9	219992	107	3280	2120	111	503.590074121	512.582754016	64.072844252
10	217936	106	3248	1648	109	508.340923877	517.418440375	64.677305047
11	215880	105	3216	1176	107	513.182266009	522.346235045	65.293279381
12	213824	104	3184	704	105	518.116710875	527.368794998	65.921099375
13	211768	103	3152	232	103	523.146970204	532.488880386	66.561110048
14	209712	102	3136	5136	102	528.275862069	537.709359606	67.213669951
15	207656	101	3104	4664	100	533.506316148	543.033214651	67.879151831

16	205600	100	3072	4192	98	538.841379310	548.463546798	68.557943350
17	203544	99	3040	3720	96	544.284221525	554.003582624	69.250447828
18	201488	98	3008	3248	94	549.838142153	559.656680406	69.957085051
19	199432	97	2976	2776	92	555.506576608	565.426336905	70.678292113
20	197376	96	2944	2304	90	561.293103448	571.316194581	71.414524323
21	195320	95	2912	1832	88	567.201451905	577.330049261	72.166256158
22	193264	94	2880	1360	86	573.235509904	583.471858295	72.933982287
23	191208	93	2848	888	84	579.399332591	589.745749245	73.718218656
24	189152	92	2816	416	82	585.697151424	596.156029128	74.519503641
25	185040	90	2768	4848	79	598.712643678	609.403940886	76.175492611
26	182984	89	2736	4376	77	605.439752034	616.251176177	77.031397022
27	180928	88	2704	3904	75	612.319749216	623.254030452	77.906753806
28	178872	87	2672	3432	73	619.357907253	630.417869882	78.802233735
29	176816	86	2640	2960	71	626.559743384	637.748310230	79.718538779
30	174760	85	2608	2488	69	633.931034482	645.251231527	80.656403941
31	172704	84	2576	2016	67	641.477832512	652.932793807	81.616599226
32	170648	83	2544	1544	65	649.206481096	660.799453973	82.599931747
33	168592	82	2512	1072	63	657.123633305	668.857983900	83.607247987
34	166536	81	2480	600	61	665.236270753	677.115489874	84.639436234
35	164480	80	2448	128	59	673.551724137	685.579433497	85.697429187
36	162424	79	2432	5032	58	682.077695329	694.257654174	86.782206772
37	160368	78	2400	4560	56	690.822281167	703.158393330	87.894799166
38	158312	77	2368	4088	54	699.793999104	712.290320516	89.036290065
39	156256	76	2336	3616	52	709.001814882	721.662561576	90.207820197
40	154200	75	2304	3144	50	718.455172413	731.284729064	91.410591133
41	152144	74	2272	2672	48	728.164026095	741.166955132	92.645869391
42	150088	73	2240	2200	46	738.138875767	751.319927120	93.914990890
43	148032	72	2208	1728	44	748.390804597	761.754926108	95.219365763
44	145976	71	2176	1256	42	758.931520155	772.483868729	96.560483591
45	143920	70	2144	784	40	769.773399014	783.519352568	97.939919071

需要说明的是，$v = m - 1504$是 16 的整数倍。在一些具体实现方式中，考虑$n_0 = 128$，有$k_0 = v/16$，也就是所述 PCS 处理采用 16 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/16$个比特进行 PCS 子处理得到输出的$n_0 = 128$个比特。在一些具体应用中，16 个 PCS 子处理采用 16 个查找表 LUT 来实现。在另一些具体应用中，16 个 PCS 子处理采用 16 个网格编码（trellis coding）来实现。需要说明的是，在另一些具体实施例中，考虑$v = m - 1504$是 32 的整数倍，当$n_0 = 64$时，$k_0 = v/32$，也就是所述 PCS 处理采用 32 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/32$个比特进行 PCS 子处理得到输出的$n_0 = 64$个比特。在一些具体应用中，32 个 PCS 子处理采用 32 个查找表 LUT 来实现。在另一些具体应用中，32 个 PCS 子处理采用 32 个网格编码（trellis coding）来实现。

可看出，本实施例中给出的参数组合，在考虑第一数据处理的输入数据（即所述第一比特数据）为 FlexO-4e 或 400G-ZR 帧数据下，对应的波特率均为小于 100GBaud；此时，在对应具体光通信网络传输中，可采用的信道间隔不大于 112.5GHz。可理解，所提出的数据处理方案可较好兼容现有 400G 数据中心、城域网络。而且，共包含$d_{in} = q \times r$个比特的第一比特数据经过所述第一数据处理和第二数据处理，刚好得到一个 DSP 帧（也称为超帧或多帧），使得所提出的数据处理方法具有较低实现复杂度，功耗更低。

实施例 8：

在实施例 7 基础上，考虑所述第一约束和第二约束。

第一数据处理包含添加循环冗余校验 CRC-32 操作。CRC-32 编码及对应的 CRC 检测操作可直接采用现有 800G-ZR 中的 CRC-32 操作，便于实现简单。也就是，考虑所述第一约束如下：当行数 r 是 4 的整数倍时，有 CRC-32 操作次数$p = r/4$，$d_{CRC} = r/4 \times 32 \le d_{CP}$，也就是对对r行数据中每4行共 41120 个比特进行 CRC-32 操作；当行数 r 不是 4 的整数倍时，有 CRC-32 操作次数$p = \lfloor r/4 \rfloor + 1$，$d_{CRC} = (\lfloor r/4 \rfloor + 1) \times 32 \le d_{CP}$，其中[a]运算表示对 a 进行向下取整运算，也就是$r_0 = 4$，$r_1 = r - 4 \times (p-1)$，对r行数据中最前面$4 \times (p-1)$行中的每$r_0 = 4$行共$q \times r_0 = 41120$个比特进行 CRC-32，对最后的r_1行共$q \times r_1$个比特进行一次 CRC-32 操作。考虑$d_{PAD} = d_{CP} - d_{CRC}$。

考虑包含 m 个比特的第一比特集合进行包含 PCS、交织和编码的第一子处理得到包含 4096 个比特的第二比特集合，其中编码操作中的编码输入比特长度和编码输出比特长度的比值为 111:128。为了便于硬件实现简单，考虑所述第四约束，即 m 为 32 的整数倍，使得所述第一子处理的输入比特长度 m 和输出比特长度的比值为 f:128，其中整数$f = m/32$。

本实施例 8 考虑第二数据处理中 DSP 成帧的冗余为$h_{DSP} = 56$，每个 DSP 帧（也称为超帧或多帧）包含 175104 个双偏振符号，其中每个 DSP 帧中的 172032 个双偏振符号（即成帧前符号）为第四比特数据经过符号映射和偏振划分得到。此时，经过第二数据处理后对应的比特速率为$W_{2-56} = W_1 \times (1 + h_{DSP}) = W_0 \times (r \times q + d_{CP})/(r \times q) \times (4096/m) \times (57/56)$，对应的波特率（即符号速率）为$W_{s-56} = W_{2-56}/8 = W_0 \times (r \times q + d_{CP})/(r \times q) \times (4096/m) \times (57/56)/8$ GBaud。

下面表 18 给出了对应的{r、m、d_{CP}、d_{PAD}、f、p、W_1、W_{2-56}、W_{s-56}}参数组合。需要说明的是，表 18 还给出了，当行数 r 不是 4 的整数倍时，对应参数r_1的具体取值。对于行数 r 是 4 的整数倍时，如表 18 中第 3 行的 r=100，对应参数r_1在表中采用斜杠代表，表示并不存在所述参数r_1。

表 18

序号	r	m	d_{CP}	d_{PAD}	f	p	v/16	r_1	W_1 (Gbit/s)	W_{2-56} (Gbit/s)	W_{s-56} (GBaud)
1	102	3136	5136	4304	98	26	102	2	528.275862069	537.709359606	67.213669951
2	101	3104	4664	3832	97	26	100	1	533.506316148	543.033214651	67.879151831
3	100	3072	4192	3392	96	25	98		538.841379310	548.463546798	68.557943350
4	99	3040	3720	2920	95	25	96	3	544.284221525	554.003582624	69.250447828
5	98	3008	3248	2448	94	25	94	2	549.838142153	559.656680406	69.957085051
6	97	2976	2776	1976	93	25	92	1	555.506576608	565.426336905	70.678292113
7	96	2944	2304	1536	92	24	90		561.293103448	571.316194581	71.414524323
8	95	2912	1832	1064	91	24	88	3	567.201451905	577.330049261	72.166256158
9	94	2880	1360	592	90	24	86	2	573.235509904	583.471858295	72.933982287
10	93	2848	888	120	89	24	84	1	579.399332591	589.745749245	73.718218656
11	79	2432	5032	4392	76	20	58	3	682.077695329	694.257654174	86.782206772
12	78	2400	4560	3920	75	20	56	2	690.822281167	703.158393330	87.894799166
13	77	2368	4088	3448	74	20	54	1	699.793999104	712.290320516	89.036290065
14	76	2336	3616	3008	73	19	52		709.001814882	721.662561576	90.207820197
15	75	2304	3144	2536	72	19	50	3	718.455172413	731.284729064	91.410591133
16	74	2272	2672	2064	71	19	48	2	728.164026095	741.166955132	92.645869391
17	73	2240	2200	1592	70	19	46	1	738.138875767	751.319927120	93.914990890
18	72	2208	1728	1152	69	18	44		748.390804597	761.754926108	95.219365763
19	71	2176	1256	680	68	18	42	3	758.931520155	772.483868729	96.560483591
20	70	2144	784	208	67	18	40	2	769.773399014	783.519352568	97.939919071

需要说明的是，$v = m - 1504$是 16 的整数倍。在一些具体实现方式中，考虑$n_0 = 128$，有$k_0 = v/16$，也就是所述 PCS 处理采用 16 个 PCS 子处理来实现所述 PCS 处理效果，其中每个 PCS 子处理将输入的$k_0 = v/16$个比特进行 PCS 子处理得到输出的$n_0 = 128$个比特。在一些具体应用中，16 个 PCS 子处理采用 16 个查找表 LUT 来实现。在另一些具体应用中，16 个 PCS 子处理采用 16 个网格编码（trellis coding）来实现。需要说明的是，在另一些具体实施例中，考虑$v = m - 1504$是 32 的整数倍，当$n_0 = 64$时，$k_0 = v/32$，也就是所述 PCS 处理采用 32 个 PCS 子处理来实现所述 PCS 处理效果，其中每个

PCS 子处理将输入的$k_0 = v/32$个比特进行 PCS 子处理得到输出的$n_0 = 64$个比特。在一些具体应用中，32 个 PCS 子处理采用 32 个查找表 LUT 来实现。在另一些具体应用中，32 个 PCS 子处理采用 32 个网格编码（trellis coding）来实现。

可看出，本实施例中给出的参数组合，在考虑第一数据处理的输入数据（即所述第一比特数据）为 FlexO-4e 或 400G-ZR 帧数据下，对应的波特率均为小于 100GBaud；此时，在对应具体光通信网络传输中，可采用的信道间隔不大于 112.5GHz。可理解，所提出的数据处理方案可较好兼容现有 400G 数据中心、城域网络。而且，共包含$d_{in} = q \times r$个比特的第一比特数据经过所述第一数据处理和第二数据处理，刚好得到一个 DSP 帧（也称为超帧或多帧），使得所提出的数据处理方法具有较低实现复杂度，功耗更低。更进一步，本实施例 2 中第一数据处理包含的添加循环冗余校验 CRC-32 操作采用现有 800G-ZR 中的 CRC-32 操作，便于实现简单；而且，m 为 32 的整数倍，使得所述第一子处理的输入比特长度 m 和输出比特长度的比值为 f:128，其中整数$f = m/32$，使得硬件实现更简单。

需要说明的是，上述实施例考虑第一数据处理的输入数据（即所述第一比特数据）为 FlexO-4e 或 400G ZR 帧数据，其标称速率（nominal bit rate）为$W_0 = 21845/25984 \times 766 \times 0.15625 \times 4 \approx 401.703640510 \times (512/511) \approx 402.489753310$ Gbit/s。上述实施例中给出的参数组合，也可以应用到其他数据类型，比如 400G 速率的 FlexO-4 数据，600G 速率的 FlexO-6e 和 FlexO-6 数据，800G 速率的 FlexO-8e 和 FlexO-8 数据，以及未来的 1.2T、1.6T 等速率的数据。在一些具体应用上，第一数据处理的输入数据（即所述第一比特数据）为 FlexO-4 数据，其标称速率（nominal bit rate）为$W_0 = 491384/462961 \times 99.5328 \times 4 \approx 422.5740431$ Gbit/s。在另一些具体应用上，第一数据处理的输入数据（即所述第一比特数据）为 FlexO-6e 数据，其标称速率为$W_0 = 21845/25984 \times 766 \times 0.15625 \times 6 \approx 603.73463$ Gbit/s。在另一些具体应用上，第一数据处理的输入数据（即所述第一比特数据）为 FlexO-6 数据，其标称速率为$W_0 = 491384/462961 \times 99.5328 \times 6 \approx 633.8611$ Gbit/s。在另一些具体应用上，第一数据处理的输入数据（即所述第一比特数据）为 FlexO-8 数据，其标称速率为$W_0 = 491384/462961 \times 99.5328 \times 8 \approx 845.14809$ Gbit/s。在另一些具体应用上，第一数据处理的输入数据（即所述第一比特数据）为 FlexO-8e 数据，其标称速率为$W_0 = 21845/25984 \times 766 \times 0.15625 \times 8 \approx 804.979507$ Gbit/s。经过第二数据处理后对应的比特速率可通过$W_2 = W_1 \times (1 + h_{DSP}) = W_0 \times (r \times q + d_{CP})/(r \times q) \times (4096/m) \times (1 + h_{DSP})$计算获得，对应的波特率（即符号速率）通过$W_s = W_2/8 = W_0 \times (r \times q + d_{CP})/(r \times q) \times (4096/m) \times (1 + h_{DSP})/8$ GBaud 计算获得，其具体计算为本领域普通技术人员可知，此处不再赘述。

图 9 为本申请实施例中数据处理装置的一种结构示意图。如图 9 所示，数据处理装置包括：第一处理单元 101，加扰单元 102，PCS 单元 103、FEC 编码和交织单元 104。第一处理单元 101 具体用于根据比特数量为d_{in}的第一比特数据，获取第二比特数据，其中，所述第二比特数据包括d_{in}个第一比特数据中的比特，d_{CRC}个校验比特和/或d_{PAD}个填充比特，d_{CRC}为大于或等于 0 的整数，d_{PAD}为大于或等于 0 的整数；具体用于对第一比特数据进行循环冗余校验 CRC 和/或插入填充比特得到第二比特数据，CRC 校验的执行方法已在前面方法实施例中具体描述，本实施例不再赘述。加扰单元 102 用于执行上述实施例中的加扰操作。PCS 单元 103 具体用于执行上述实施例中 PCS 处理的操作。FEC 编码和交织单元 104 具体用于执行上述实施例中 FEC 编码和交织的操作。

在一些可能的实施方式中，图 10 所示的数据处理装置还可以包括：第二处理单元 105 和数字信号处理 DSP 成帧单元 106；所述第二处理单元 105，用于对第四比特数据进行符号映射和偏振划分（也简称为双偏振符号映射）得到双偏振符号序列，其中，所述符号映射和所述偏振划分（也简称为双偏振符号映射）将 8 个比特映射为 1 个双偏振 DP-16QAM 符号，所述第四比特数据包括多个所述第二比特集合；所述 DSP 成帧单元 106，用于对所述双偏振符号序列进行 DSP 成帧，得到 DSP 帧（也称为 DSP 超帧）。符号映射、偏振划分以及 DSP 成帧操作的具体操作可以参考上述图 5（a）-5（c）所示实施例的相关介绍，此处不再赘述。

应理解，本申请提供的数据处理装置也可以通过其他方式实现。例如，上述装置中的单元划分仅仅是一种逻辑功能划分，实际实现时可以有另外的划分方式，例如，多个单元或组件可以结合或可以集成到另一个系统。另外，本申请各个实施例中的各功能单元可以集成在一个处理单元中，也可以是各个独立的物理单元，也可以是两个或两个以上个功能单元集成在一个处理单元中。上述集成的单元既可以采用硬件的形式实现，也可以采用软件功能单元的形式实现。

图 11 为本申请实施例中光模块的一种结构示意图。如图 11 所示，光模块包括处理器 201 和接口 202，

处理器 201 用于执行上述实施例中介绍的第一数据处理和第二数据处理的操作。在一种可能的实施方式中，处理器 201 包括上述图 9 所示的第一处理单元 101，加扰单元 102，PCS 单元 103、FEC 编码和交织单元 104；也可包括图 10 所示的第二处理单元 105 和 DSP 成帧单元 106。接口 202 可以是收发器或者输入输出接口，接口 202 用于接收来自该光模块之外的其它装置的信号并传输至处理器 201，或将来自处理器 201 的信号发送给该光模块之外的其它装置。作为一个示例，处理器 201 进行上述第一数据处理和第二数据处理后得到 DSP 帧，并通过接口 202 发送 DSP 帧，在该示例中接口 202 具体可指电接口。作为另一个示例，处理器 201 进行上述第一数据处理和第二数据处理后得到 DSP 帧，光模块中的调制器（modulator）根据 DSP 帧进行电光转换等信号处理得到光信号（optical signal），再通过接口 202 发送光信号，在该示例中接口 202 具体可指光接口。可选地，光模块还可以包括存储器 203，其中，存储器 203 用于存储程序指令和/或数据。所述调制器也可以称为信号驱动器（Signal driver）。

图 12 为本申请实施例中发送设备的一种结构示意图。如图 12 所示，该发送设备包括主机侧设备 301 和光模块 302。主机侧设备 301 用于向光模块 302 发送数据，光模块 302 根据主机侧设备 301 发送的数据生成光信号并通过信道发送光信号。例如，主机侧设备具体可以是交换机或路由器等。该发送设备可以是一种包括主机侧设备 301 和光模块 302 的通信设备。还应理解，本申请实施例中的发送设备是基于数据流向来命名的，并不对设备的功能进行限定，例如，发送设备也可以具有接收的功能。

本申请实施例还提供了一种光传送网(Optical Transport Network，OTN)设备，该 OTN 设备包括线路侧（line side）设备和客户侧（client side）设备。其中，客户侧设备在一些场景中也可以称之为支路侧设备。线路侧设备包括处理器和接口，处理器用于执行上述实施例中介绍的第一数据处理和第二数据处理的操作。接口可以是收发器或者输入输出接口，接口用于接收来自该线路侧设备之外的其它装置的信号并传输至处理器或将来自处理器的信号发送给该线路侧设备之外的其它装置。

本申请实施例还提供一种芯片。该芯片中集成了用于实现上述处理器 201 的功能的电路和一个或者多个接口。作为一个示例，该芯片中集成了存储器。作为另一个示例，该芯片中未集成存储器时，可以通过接口与外置的存储器连接。该芯片可以完成前述实施例中的任一个或多个实施例的方法步骤。或者，该芯片根据存储器中存储的程序代码来实现上述实施例中数据处理装置执行的动作。

本申请实施例还提供一种计算机可读存储介质，包括程序或指令，当所述程序或指令在计算机上运行时，使得如上述方法实施例中处理器 201 所执行的方法。

应理解，本申请实施例中提及的处理器可以通过硬件实现也可以通过软件实现。当通过硬件实现时，该处理器可以是逻辑电路、集成电路等。当通过软件实现时，该处理器可以是一个通用处理器，通过读取存储器中存储的软件代码来实现。存储器可以是独立存在，与处理器相连接，或者，存储器也可以和处理器集成在一起。

作为一个示例，本申请的实施例中的处理器可以是中央处理单元(central processing unit，CPU)，还可以是其它通用处理器、数字信号处理器(digital signal processor，DSP)、专用集成电路(application specific integrated circuit，ASIC)、现场可编程门阵列(field programmable gate array，FPGA)或者其它可编程逻辑器件、晶体管逻辑器件，硬件部件或者其任意组合。通用处理器可以是微处理器，也可以是任何常规的处理器，还可以是实现特定功能的处理电路。

本申请的实施例中存储器可以是随机存取存储器(random access memory，RAM)、闪存、只读存储器(read-only memory，ROM)、可编程只读存储器(programmable ROM，PROM)、可擦除可编程只读存储器(erasable PROM，EPROM)、电可擦除可编程只读存储器(electrically EPROM，EEPROM)、寄存器、硬盘、移动硬盘、CD-ROM 或者本领域熟知的任何其它形式的存储介质中。一种示例性的存储介质耦合至处理器，从而使处理器能够从该存储介质读取信息，且可向该存储介质写入信息。当然，存储介质也可以是处理器的组成部分。处理器和存储介质可以位于 ASIC 中。另外，该 ASIC 可以位于网络设备或终端设备中。当然，处理器和存储介质也可以作为分立组件存在于网络设备或终端设备中。

在上述实施例中，可以全部或部分地通过软件、硬件、固件或者其任意组合来实现。

当使用硬件实现时，本申请实施例提供的数据处理方法可以不通过读取软件代码或指令来实现，例如，可以通过 CPU、DSP、ASIC、FPGA、其它可编程逻辑器件、晶体管逻辑器件，硬件部件或者其任意组合来实现。

当使用软件实现时，可以全部或部分地以计算机程序产品的形式实现。所述计算机程序产品包括一

个或多个计算机程序或指令。在计算机上加载和执行所述计算机程序或指令时，全部或部分地执行本申请实施例所述的流程或功能。所述计算机可以是通用计算机、专用计算机、计算机网络、网络设备、终端设备、或者其它可编程装置。所述计算机程序或指令可以存储在计算机可读存储介质中，或者通过所述计算机可读存储介质进行传输。所述计算机可读存储介质可以是计算机能够存取的任何可用介质或者是集成一个或多个可用介质的服务器等数据存储设备。所述可用介质可以是磁性介质，例如，软盘、硬盘、磁带；也可以是光介质，例如，数字通用光盘（Digital Versatile Disc，DVD）；还可以是半导体介质，例如，固态硬盘(solid state disk，SSD)。

最后应说明的是：以上所述，仅为本申请的具体实施方式，但本申请的保护范围并不局限于此，任何熟悉本技术领域的技术人员在本申请揭露的技术范围内，可轻易想到变化或替换，都应涵盖在本申请的保护范围之内。因此，本申请的保护范围应以所述权利要求的保护范围为准。

权 利 要 求 书

1. 一种数据处理方法，其特征在于，包括：

根据比特数量为d_{in}的第一比特数据，获取第二比特数据，其中，所述第二比特数据包括d_{in}个第一比特数据中的比特，d_{CRC}个校验比特和/或d_{PAD}个填充比特，d_{CRC}为大于或等于 0 的整数，d_{PAD}为大于或等于 0 的整数；

对所述第二比特数据进行加扰得到第三比特数据，所述第三比特数据的比特数量是$d_{scr} = d_{in} + d_{CP}$，$d_{CP} = d_{CRC} + d_{PAD}$；

对比特数量为 m 的第一比特集合中的第一比特子集进行概率星座整形 PCS 处理得到第二比特子集，其中，所述第一比特集合是从所述第三比特数据中获取的，m 为大于 1 的整数，d_{scr}是 m 的整数倍；

对所述第二比特子集和所述第一比特集合中除所述第一比特子集外的第三比特子集进行前向纠错 FEC 编码和交织处理得到第二比特集合，其中，d_{in}，m 和d_{CP}的值为下表中的任意一组：

d_{in}	m	d_{CP}
1186312	3536	1784
1184256	3536	3840
1182200	3520	520
1180144	3520	2576
1176032	3504	1312
1173976	3504	3368
1171920	3488	48
1169864	3488	2104
1165752	3472	840
1163696	3472	2896
1159584	3456	1632
1157528	3456	3688
1155472	3440	368
1153416	3440	2424
1149304	3424	1160
1147248	3424	3216
1143136	3408	1952
1141080	3408	4008
1139024	3392	688
1136968	3392	2744
1132856	3376	1480
1130800	3376	3536
1128744	3360	216
1122576	3344	1008
1120520	3344	3064
1116408	3328	1800
1114352	3328	3856
1112296	3312	536
1110240	3312	2592
1106128	3296	1328
1104072	3296	3384
1102016	3280	64
1099960	3280	2120
1095848	3264	856
1093792	3264	2912
1089680	3248	1648

1087624	3248	3704
1085568	3232	384
1083512	3232	2440
1079400	3216	1176
1077344	3216	3232
1071176	3200	4024
1069120	3184	704
1067064	3184	2760
1062952	3168	1496
1060896	3168	3552
1058840	3152	232
1056784	3152	2288
1052672	3136	1024
1050616	3136	3080
1046504	3120	1816
1044448	3120	3872
1042392	3104	552
1040336	3104	2608
1036224	3088	1344
1034168	3088	3400
1032112	3072	80
1030056	3072	2136
1025944	3056	872
1023888	3056	2928
1019776	3040	1664
1017720	3040	3720
1015664	3024	400
1013608	3024	2456
1009496	3008	1192
1007440	3008	3248
1003328	2992	1984
1001272	2992	4040
999216	2976	720
997160	2976	2776
993048	2960	1512
990992	2960	3568
988936	2944	248
986880	2944	2304
982768	2928	1040
980712	2928	3096
976600	2912	1832
974544	2912	3888
972488	2896	568
970432	2896	2624
966320	2880	1360
964264	2880	3416
962208	2864	96
960152	2864	2152

956040	2848	888
953984	2848	2944
949872	2832	1680
947816	2832	3736
945760	2816	416
943704	2816	2472
939592	2800	1208
937536	2800	3264
933424	2784	2000
931368	2784	4056
929312	2768	736
927256	2768	2792
923144	2752	1528
921088	2752	3584
919032	2736	264
916976	2736	2320
912864	2720	1056
910808	2720	3112
906696	2704	1848
904640	2704	3904
902584	2688	584
900528	2688	2640
896416	2672	1376
894360	2672	3432
892304	2656	112
886136	2640	904
884080	2640	2960
879968	2624	1696
877912	2624	3752
875856	2608	432
873800	2608	2488
869688	2592	1224
867632	2592	3280
863520	2576	2016
861464	2576	4072
859408	2560	752
857352	2560	2808
853240	2544	1544
851184	2544	3600
849128	2528	280
847072	2528	2336
842960	2512	1072
840904	2512	3128
836792	2496	1864
834736	2496	3920
832680	2480	600
830624	2480	2656
826512	2464	1392

824456	2464	3448
822400	2448	128
820344	2448	2184
816232	2432	920
814176	2432	2976
810064	2416	1712
808008	2416	3768
805952	2400	448
803896	2400	2504
799784	2384	1240
797728	2384	3296
793616	2368	2032
791560	2368	4088
789504	2352	768
787448	2352	2824
783336	2336	1560
781280	2336	3616
779224	2320	296
777168	2320	2352
773056	2304	1088
771000	2304	3144
766888	2288	1880
764832	2288	3936
762776	2272	616
760720	2272	2672
756608	2256	1408
754552	2256	3464
752496	2240	144
750440	2240	2200
746328	2224	936
744272	2224	2992
740160	2208	1728
738104	2208	3784
736048	2192	464
733992	2192	2520
729880	2176	1256
727824	2176	3312
723712	2160	2048
721656	2160	4104
719600	2144	784

。

2. 根据权利要求 1 所述的方法，其特征在于，$d_{scr} = 336 \times m$。

3. 根据权利要求 1 或 2 所述的方法，其特征在于，所述第一比特子集的比特数量是 16 或 32 的整数倍，所述第二比特子集包括 2048 个比特，所述第三比特子集包括 1504 比特，所述第二比特集合包括 4096 个比特。

4. 根据权利要求 1-3 中任一项所述的方法，其特征在于，所述第一比特数据包括 r 行 q 列个比特，$d_{in}=r\times q$，r 为大于 1 的整数，q=2056 或 10280。

5. 根据权利要求 4 所述的方法，其特征在于，所述根据比特数量为d_{in}的第一比特数据，获取第二比

特数据，具体包括：对所述第一比特数据进行循环冗余校验 CRC 和/或插入填充比特得到第二比特数据；其中，对所述第一比特数据进行 CRC 包括：

对所述第一比特数据中每 r/p 行总共 r×q/p 个比特进行 CRC-32 校验，添加长度为 32 比特的校验比特，$d_{CRC}=32\times p$，r 能被p整除，p 为大于 1 的整数；或者，

对所述第一比特数据中前$r_0\times(p-1)$行中的每r_0行共$q\times r_0$个比特进行 CRC-32 校验以添加长度为 32 比特的校验比特，并对所述第一数据中最后r_1行共$q\times r_1$个比特进行 CRC-32 校验以添加长度为 32 比特的校验比特，其中，$d_{CRC}=32\times p$，r 不能被p整除，p 为大于 1 的整数，$r_0\times(p-1)+r_1=r$，整数r_0大于整数r_1。

6. 根据权利要求 5 所述的方法，其特征在于，q=2056，r，m，d_{CP}，p 和 r_1的值为下表中的任意一组：

r	m	d_{CP}	p	r_1
577	3536	1784	29	17
576	3536	3840	29	16
574	3520	2576	29	14
572	3504	1312	29	12
571	3504	3368	29	11
569	3488	2104	29	9
566	3472	2896	29	6
564	3456	1632	29	4
563	3456	3688	29	3
561	3440	2424	29	1
559	3424	1160	28	19
558	3424	3216	28	18
556	3408	1952	28	16
555	3408	4008	28	15
553	3392	2744	28	13
551	3376	1480	28	11
550	3376	3536	28	10
546	3344	1008	28	6
545	3344	3064	28	5
543	3328	1800	28	3
542	3328	3856	28	2
540	3312	2592	27	
538	3296	1328	27	18
537	3296	3384	27	17
535	3280	2120	27	15
532	3264	2912	27	12
530	3248	1648	27	10
529	3248	3704	27	9
527	3232	2440	27	7
525	3216	1176	27	5
524	3216	3232	27	4
521	3200	4024	27	1
519	3184	2760	26	19
517	3168	1496	26	17
516	3168	3552	26	16
514	3152	2288	26	14
512	3136	1024	26	12
511	3136	3080	26	11

509	3120	1816	26	9
508	3120	3872	26	8
506	3104	2608	26	6
504	3088	1344	26	4
503	3088	3400	26	3
501	3072	2136	26	1
499	3056	872	25	19
498	3056	2928	25	18
496	3040	1664	25	16
495	3040	3720	25	15
493	3024	2456	25	13
491	3008	1192	25	11
490	3008	3248	25	10
488	2992	1984	25	8
487	2992	4040	25	7
485	2976	2776	25	5
483	2960	1512	25	3
482	2960	3568	25	2
480	2944	2304	24	
478	2928	1040	24	18
477	2928	3096	24	17
475	2912	1832	24	15
474	2912	3888	24	14
472	2896	2624	24	12
470	2880	1360	24	10
469	2880	3416	24	9
467	2864	2152	24	7
465	2848	888	24	5
464	2848	2944	24	4
462	2832	1680	24	2
461	2832	3736	24	1
459	2816	2472	23	19
457	2800	1208	23	17
456	2800	3264	23	16
454	2784	2000	23	14
453	2784	4056	23	13
452	2768	736	23	12
451	2768	2792	23	11
449	2752	1528	23	9
448	2752	3584	23	8
446	2736	2320	23	6
444	2720	1056	23	4
443	2720	3112	23	3
441	2704	1848	23	1
440	2704	3904	22	
438	2688	2640	22	18
436	2672	1376	22	16
435	2672	3432	22	15

431	2640	904	22	11
430	2640	2960	22	10
428	2624	1696	22	8
427	2624	3752	22	7
425	2608	2488	22	5
423	2592	1224	22	3
422	2592	3280	22	2
420	2576	2016	21	
419	2576	4072	21	19
418	2560	752	21	18
417	2560	2808	21	17
415	2544	1544	21	15
414	2544	3600	21	14
412	2528	2336	21	12
410	2512	1072	21	10
409	2512	3128	21	9
407	2496	1864	21	7
406	2496	3920	21	6
404	2480	2656	21	4
402	2464	1392	21	2
401	2464	3448	21	1
399	2448	2184	20	19
397	2432	920	20	17
396	2432	2976	20	16
394	2416	1712	20	14
393	2416	3768	20	13
391	2400	2504	20	11
389	2384	1240	20	9
388	2384	3296	20	8
386	2368	2032	20	6
385	2368	4088	20	5
384	2352	768	20	4
383	2352	2824	20	3
381	2336	1560	20	1
380	2336	3616	19	
378	2320	2352	19	18
376	2304	1088	19	16
375	2304	3144	19	15
373	2288	1880	19	13
372	2288	3936	19	12
371	2272	616	19	11
370	2272	2672	19	10
368	2256	1408	19	8
367	2256	3464	19	7
365	2240	2200	19	5
363	2224	936	19	3
362	2224	2992	19	2
360	2208	1728	18	

359	2208	3784	18	19
357	2192	2520	18	17
355	2176	1256	18	15
354	2176	3312	18	14
352	2160	2048	18	12
351	2160	4104	18	11
350	2144	784	18	10

。

7. 根据权利要求 5 所述的方法，其特征在于，m 为 32 的整数倍，q=2056；r，m，d_{CP}，p 和 r_1 的值为下表中的任意一组：

r	m	d_{CP}	p	r_1
574	3520	2576	29	14
569	3488	2104	29	9
564	3456	1632	29	4
563	3456	3688	29	3
559	3424	1160	28	19
558	3424	3216	28	18
553	3392	2744	28	13
543	3328	1800	28	3
542	3328	3856	28	2
538	3296	1328	27	18
537	3296	3384	27	17
532	3264	2912	27	12
527	3232	2440	27	7
521	3200	4024	27	1
517	3168	1496	26	17
516	3168	3552	26	16
512	3136	1024	26	12
511	3136	3080	26	11
506	3104	2608	26	6
501	3072	2136	26	1
496	3040	1664	25	16
495	3040	3720	25	15
491	3008	1192	25	11
490	3008	3248	25	10
485	2976	2776	25	5
480	2944	2304	24	
475	2912	1832	24	15
474	2912	3888	24	14
470	2880	1360	24	10
469	2880	3416	24	9
465	2848	888	24	5
464	2848	2944	24	4
459	2816	2472	23	19
454	2784	2000	23	14
453	2784	4056	23	13
449	2752	1528	23	9
448	2752	3584	23	8

444	2720	1056	23	4
443	2720	3112	23	3
438	2688	2640	22	18
428	2624	1696	22	8
427	2624	3752	22	7
423	2592	1224	22	3
422	2592	3280	22	2
418	2560	752	21	18
417	2560	2808	21	17
412	2528	2336	21	12
407	2496	1864	21	7
406	2496	3920	21	6
402	2464	1392	21	2
401	2464	3448	21	1
397	2432	920	20	17
396	2432	2976	20	16
391	2400	2504	20	11
386	2368	2032	20	6
385	2368	4088	20	5
381	2336	1560	20	1
380	2336	3616	19	
376	2304	1088	19	16
375	2304	3144	19	15
371	2272	616	19	11
370	2272	2672	19	10
365	2240	2200	19	5
360	2208	1728	18	
359	2208	3784	18	19
355	2176	1256	18	15
354	2176	3312	18	14
350	2144	784	18	10

。

8. 根据权利要求 5 所述的方法，其特征在于，q=10280，r，m，d_{CP}，p 和 r_1 的值为下表中的任意一组：

r	m	d_{CP}	p	r_1
113	3472	4952	29	1
112	3440	4480	28	
111	3408	4008	28	3
110	3376	3536	28	2
109	3344	3064	28	1
108	3312	2592	27	
107	3280	2120	27	3
106	3248	1648	27	2
105	3216	1176	27	1
102	3136	5136	26	2
101	3104	4664	26	1
100	3072	4192	25	
99	3040	3720	25	3

98	3008	3248	25	2
97	2976	2776	25	1
96	2944	2304	24	
95	2912	1832	24	3
94	2880	1360	24	2
93	2848	888	24	1
90	2768	4848	23	2
89	2736	4376	23	1
88	2704	3904	22	
87	2672	3432	22	3
86	2640	2960	22	2
85	2608	2488	22	1
84	2576	2016	21	
83	2544	1544	21	3
82	2512	1072	21	2
79	2432	5032	20	3
78	2400	4560	20	2
77	2368	4088	20	1
76	2336	3616	19	
75	2304	3144	19	3
74	2272	2672	19	2
73	2240	2200	19	1
72	2208	1728	18	
71	2176	1256	18	3
70	2144	784	18	2

。

9. 根据权利要求 5 所述的方法，其特征在于，m 为 32 的整数倍，q=10280，r，m，d_{CP}，p 和 r_1的值为下表中的任意一组：

r	m	d_{CP}	p	r_1
102	3136	5136	26	2
101	3104	4664	26	1
100	3072	4192	25	
99	3040	3720	25	3
98	3008	3248	25	2
97	2976	2776	25	1
96	2944	2304	24	
95	2912	1832	24	3
94	2880	1360	24	2
93	2848	888	24	1
79	2432	5032	20	3
78	2400	4560	20	2
77	2368	4088	20	1
76	2336	3616	19	
75	2304	3144	19	3
74	2272	2672	19	2
73	2240	2200	19	1
72	2208	1728	18	
71	2176	1256	18	3

70	2144	784	18	2

。

10. 根据权利要求 1-9 中任一项所述的方法，其特征在于，所述第三比特子集包括 1504 比特，所述第一比特子集包括 m-1504 比特；所述对比特数量为 m 的第一比特集合中的第一比特子集进行概率星座整形 PCS 处理得到第二比特子集，包括：

将所述第一比特子集分成 16 组序列，每组序列包括(m-1504)/16 个比特；

分别对所述 16 组序列进行 PCS 子处理，得到与所述 16 组序列对应的 16 组输出序列，其中，每组输出序列包括 128 比特，所述 16 组输出序列构成所述第二比特子集；

或者，所述对比特数量为 m 的第一比特集合中的第一比特子集进行概率星座整形 PCS 处理得到第二比特子集，包括：

将所述第一比特子集分成 32 组序列，每组序列包括(m-1504)/32 个比特；

分别对所述 32 组序列进行 PCS 子处理，得到与所述 32 组序列对应的 32 组输出序列，其中，每组输出序列包括 64 比特，所述 32 组输出序列构成所述第二比特子集。

11. 根据权利要求 1-10 中任一项所述的方法，其特征在于，在得到第二比特集合之后，所述方法还包括：

对第四比特数据进行符号映射和偏振划分得到双偏振符号序列，其中，所述符号映射和所述偏振划分将 8 个比特映射为 1 个双偏振 DP-16QAM 符号，所述第四比特数据包括多个所述第二比特集合；

对所述双偏振符号序列进行数字信号处理 DSP 成帧，得到 DSP 帧。

12. 根据权利要求 11 所述的方法，其特征在于，对所述双偏振符号序列进行 DSP 成帧包括：

将每 172032 个双偏振 DP-16QAM 符号进行成帧处理得到一个 DSP 帧。

13. 一种数据处理装置，其特征在于，包括：第一处理单元、加扰单元、概率星座整形 PCS 单元和前向纠错 FEC 编码和交织单元；

所述第一处理单元用于：根据比特数量为d_{in}的第一比特数据，获取第二比特数据，其中，所述第二比特数据包括d_{in}个第一比特数据中的比特，d_{CRC}个校验比特和/或d_{PAD}个填充比特，d_{CRC}为大于或等于 0 的整数，d_{PAD}为大于或等于 0 的整数；

加扰单元用于：对所述第二比特数据进行加扰得到第三比特数据，所述第三比特数据的比特数量是$d_{scr} = r \times q + d_{CP}$，$d_{CP} = d_{CRC} + d_{PAD}$；

所述 PCS 单元用于：对比特数量为 m 的第一比特集合中的第一比特子集进行概率星座整形 PCS 处理得到第二比特子集，其中，所述第一比特集合是从所述第三比特数据中获取的，m 为大于 1 的整数，d_{scr}是 m 的整数倍；

所述 FEC 编码和交织单元用于：对所述第二比特子集和所述第一比特集合中除所述第一比特子集外的第三比特子集进行 FEC 编码和交织处理得到第二比特集合，其中，d_{in}，m 和d_{CP}的值为下表中的任意一组：

d_{in}	m	d_{CP}
1186312	3536	1784
1184256	3536	3840
1182200	3520	520
1180144	3520	2576
1176032	3504	1312
1173976	3504	3368
1171920	3488	48
1169864	3488	2104
1165752	3472	840
1163696	3472	2896
1159584	3456	1632
1157528	3456	3688
1155472	3440	368
1153416	3440	2424

1149304	3424	1160
1147248	3424	3216
1143136	3408	1952
1141080	3408	4008
1139024	3392	688
1136968	3392	2744
1132856	3376	1480
1130800	3376	3536
1128744	3360	216
1122576	3344	1008
1120520	3344	3064
1116408	3328	1800
1114352	3328	3856
1112296	3312	536
1110240	3312	2592
1106128	3296	1328
1104072	3296	3384
1102016	3280	64
1099960	3280	2120
1095848	3264	856
1093792	3264	2912
1089680	3248	1648
1087624	3248	3704
1085568	3232	384
1083512	3232	2440
1079400	3216	1176
1077344	3216	3232
1071176	3200	4024
1069120	3184	704
1067064	3184	2760
1062952	3168	1496
1060896	3168	3552
1058840	3152	232
1056784	3152	2288
1052672	3136	1024
1050616	3136	3080
1046504	3120	1816
1044448	3120	3872
1042392	3104	552
1040336	3104	2608
1036224	3088	1344
1034168	3088	3400
1032112	3072	80
1030056	3072	2136
1025944	3056	872
1023888	3056	2928
1019776	3040	1664
1017720	3040	3720

1015664	3024	400
1013608	3024	2456
1009496	3008	1192
1007440	3008	3248
1003328	2992	1984
1001272	2992	4040
999216	2976	720
997160	2976	2776
993048	2960	1512
990992	2960	3568
988936	2944	248
986880	2944	2304
982768	2928	1040
980712	2928	3096
976600	2912	1832
974544	2912	3888
972488	2896	568
970432	2896	2624
966320	2880	1360
964264	2880	3416
962208	2864	96
960152	2864	2152
956040	2848	888
953984	2848	2944
949872	2832	1680
947816	2832	3736
945760	2816	416
943704	2816	2472
939592	2800	1208
937536	2800	3264
933424	2784	2000
931368	2784	4056
929312	2768	736
927256	2768	2792
923144	2752	1528
921088	2752	3584
919032	2736	264
916976	2736	2320
912864	2720	1056
910808	2720	3112
906696	2704	1848
904640	2704	3904
902584	2688	584
900528	2688	2640
896416	2672	1376
894360	2672	3432
892304	2656	112
886136	2640	904

884080	2640	2960
879968	2624	1696
877912	2624	3752
875856	2608	432
873800	2608	2488
869688	2592	1224
867632	2592	3280
863520	2576	2016
861464	2576	4072
859408	2560	752
857352	2560	2808
853240	2544	1544
851184	2544	3600
849128	2528	280
847072	2528	2336
842960	2512	1072
840904	2512	3128
836792	2496	1864
834736	2496	3920
832680	2480	600
830624	2480	2656
826512	2464	1392
824456	2464	3448
822400	2448	128
820344	2448	2184
816232	2432	920
814176	2432	2976
810064	2416	1712
808008	2416	3768
805952	2400	448
803896	2400	2504
799784	2384	1240
797728	2384	3296
793616	2368	2032
791560	2368	4088
789504	2352	768
787448	2352	2824
783336	2336	1560
781280	2336	3616
779224	2320	296
777168	2320	2352
773056	2304	1088
771000	2304	3144
766888	2288	1880
764832	2288	3936
762776	2272	616
760720	2272	2672
756608	2256	1408

754552	2256	3464
752496	2240	144
750440	2240	2200
746328	2224	936
744272	2224	2992
740160	2208	1728
738104	2208	3784
736048	2192	464
733992	2192	2520
729880	2176	1256
727824	2176	3312
723712	2160	2048
721656	2160	4104
719600	2144	784

。

14. 根据权利要求 13 所述的装置，其特征在于，$d_{scr} = 336 \times m$。

15. 根据权利要求 13 或 14 所述的装置，其特征在于，所述第一比特子集的比特数量是 16 或 32 的整数倍，所述第二比特子集包括 2048 个比特,所述第三比特子集包括 1504 比特，所述第二比特集合包括 4096 个比特。

16. 根据权利要求 13-15 中任一项所述的装置，其特征在于，所述第一比特数据包括 r 行 q 列个比特，d_{in}=r×q，r 为大于 1 的整数，q=2056 或 10280。

17. 根据权利要求 16 所述的装置，其特征在于，所述处理单元具体用于：

对所述第一比特数据中每 r/p 行总共 r×q/p 个比特进行 CRC-32 校验，添加长度为 32 比特的校验比特，$d_{CRC} = 32 \times p$，r 能被p整除，p 为大于 1 的整数；或者，

对所述第一比特数据中前$r_0 \times (p-1)$行中的每r_0行共$q \times r_0$个比特进行 CRC-32 校验以添加长度为 32 比特的校验比特，并对所述第一数据中最后r_1行共$q \times r_1$个比特进行 CRC-32 校验以添加长度为 32 比特的校验比特，其中，$d_{CRC} = 32 \times p$，r 不能被p整除，p 为大于 1 的整数，$r_0 \times (p-1) + r_1 = r$，整数$r_0$大于整数$r_1$。

18. 根据权利要求 17 所述的装置，其特征在于，q=2056，r，m，d_{CP}，p 和 r_1的值为下表中的任意一组：

r	m	d_{CP}	p	r_1
577	3536	1784	29	17
576	3536	3840	29	16
574	3520	2576	29	14
572	3504	1312	29	12
571	3504	3368	29	11
569	3488	2104	29	9
566	3472	2896	29	6
564	3456	1632	29	4
563	3456	3688	29	3
561	3440	2424	29	1
559	3424	1160	28	19
558	3424	3216	28	18
556	3408	1952	28	16
555	3408	4008	28	15
553	3392	2744	28	13
551	3376	1480	28	11
550	3376	3536	28	10
546	3344	1008	28	6

545	3344	3064	28	5
543	3328	1800	28	3
542	3328	3856	28	2
540	3312	2592	27	
538	3296	1328	27	18
537	3296	3384	27	17
535	3280	2120	27	15
532	3264	2912	27	12
530	3248	1648	27	10
529	3248	3704	27	9
527	3232	2440	27	7
525	3216	1176	27	5
524	3216	3232	27	4
521	3200	4024	27	1
519	3184	2760	26	19
517	3168	1496	26	17
516	3168	3552	26	16
514	3152	2288	26	14
512	3136	1024	26	12
511	3136	3080	26	11
509	3120	1816	26	9
508	3120	3872	26	8
506	3104	2608	26	6
504	3088	1344	26	4
503	3088	3400	26	3
501	3072	2136	26	1
499	3056	872	25	19
498	3056	2928	25	18
496	3040	1664	25	16
495	3040	3720	25	15
493	3024	2456	25	13
491	3008	1192	25	11
490	3008	3248	25	10
488	2992	1984	25	8
487	2992	4040	25	7
485	2976	2776	25	5
483	2960	1512	25	3
482	2960	3568	25	2
480	2944	2304	24	
478	2928	1040	24	18
477	2928	3096	24	17
475	2912	1832	24	15
474	2912	3888	24	14
472	2896	2624	24	12
470	2880	1360	24	10
469	2880	3416	24	9
467	2864	2152	24	7
465	2848	888	24	5

464	2848	2944	24	4
462	2832	1680	24	2
461	2832	3736	24	1
459	2816	2472	23	19
457	2800	1208	23	17
456	2800	3264	23	16
454	2784	2000	23	14
453	2784	4056	23	13
452	2768	736	23	12
451	2768	2792	23	11
449	2752	1528	23	9
448	2752	3584	23	8
446	2736	2320	23	6
444	2720	1056	23	4
443	2720	3112	23	3
441	2704	1848	23	1
440	2704	3904	22	
438	2688	2640	22	18
436	2672	1376	22	16
435	2672	3432	22	15
431	2640	904	22	11
430	2640	2960	22	10
428	2624	1696	22	8
427	2624	3752	22	7
425	2608	2488	22	5
423	2592	1224	22	3
422	2592	3280	22	2
420	2576	2016	21	
419	2576	4072	21	19
418	2560	752	21	18
417	2560	2808	21	17
415	2544	1544	21	15
414	2544	3600	21	14
412	2528	2336	21	12
410	2512	1072	21	10
409	2512	3128	21	9
407	2496	1864	21	7
406	2496	3920	21	6
404	2480	2656	21	4
402	2464	1392	21	2
401	2464	3448	21	1
399	2448	2184	20	19
397	2432	920	20	17
396	2432	2976	20	16
394	2416	1712	20	14
393	2416	3768	20	13
391	2400	2504	20	11
389	2384	1240	20	9

388	2384	3296	20	8
386	2368	2032	20	6
385	2368	4088	20	5
384	2352	768	20	4
383	2352	2824	20	3
381	2336	1560	20	1
380	2336	3616	19	
378	2320	2352	19	18
376	2304	1088	19	16
375	2304	3144	19	15
373	2288	1880	19	13
372	2288	3936	19	12
371	2272	616	19	11
370	2272	2672	19	10
368	2256	1408	19	8
367	2256	3464	19	7
365	2240	2200	19	5
363	2224	936	19	3
362	2224	2992	19	2
360	2208	1728	18	
359	2208	3784	18	19
357	2192	2520	18	17
355	2176	1256	18	15
354	2176	3312	18	14
352	2160	2048	18	12
351	2160	4104	18	11
350	2144	784	18	10

。

19. 根据权利要求 17 所述的装置，其特征在于，m 为 32 的整数倍，q=2056；r，m，d_{CP}，p 和 r_1的值为下表中的任意一组：

r	m	d_{CP}	p	r_1
574	3520	2576	29	14
569	3488	2104	29	9
564	3456	1632	29	4
563	3456	3688	29	3
559	3424	1160	28	19
558	3424	3216	28	18
553	3392	2744	28	13
543	3328	1800	28	3
542	3328	3856	28	2
538	3296	1328	27	18
537	3296	3384	27	17
532	3264	2912	27	12
527	3232	2440	27	7
521	3200	4024	27	1
517	3168	1496	26	17
516	3168	3552	26	16
512	3136	1024	26	12

511	3136	3080	26	11
506	3104	2608	26	6
501	3072	2136	26	1
496	3040	1664	25	16
495	3040	3720	25	15
491	3008	1192	25	11
490	3008	3248	25	10
485	2976	2776	25	5
480	2944	2304	24	
475	2912	1832	24	15
474	2912	3888	24	14
470	2880	1360	24	10
469	2880	3416	24	9
465	2848	888	24	5
464	2848	2944	24	4
459	2816	2472	23	19
454	2784	2000	23	14
453	2784	4056	23	13
449	2752	1528	23	9
448	2752	3584	23	8
444	2720	1056	23	4
443	2720	3112	23	3
438	2688	2640	22	18
428	2624	1696	22	8
427	2624	3752	22	7
423	2592	1224	22	3
422	2592	3280	22	2
418	2560	752	21	18
417	2560	2808	21	17
412	2528	2336	21	12
407	2496	1864	21	7
406	2496	3920	21	6
402	2464	1392	21	2
401	2464	3448	21	1
397	2432	920	20	17
396	2432	2976	20	16
391	2400	2504	20	11
386	2368	2032	20	6
385	2368	4088	20	5
381	2336	1560	20	1
380	2336	3616	19	
376	2304	1088	19	16
375	2304	3144	19	15
371	2272	616	19	11
370	2272	2672	19	10
365	2240	2200	19	5
360	2208	1728	18	
359	2208	3784	18	19

355	2176	1256	18	15
354	2176	3312	18	14
350	2144	784	18	10

。

20. 根据权利要求 17 所述的装置，其特征在于，q=10280，r，m，d_{CP}，p 和 r_1 的值为下表中的任意一组：

r	m	d_{CP}	p	r_1
113	3472	4952	29	1
112	3440	4480	28	
111	3408	4008	28	3
110	3376	3536	28	2
109	3344	3064	28	1
108	3312	2592	27	
107	3280	2120	27	3
106	3248	1648	27	2
105	3216	1176	27	1
102	3136	5136	26	2
101	3104	4664	26	1
100	3072	4192	25	
99	3040	3720	25	3
98	3008	3248	25	2
97	2976	2776	25	1
96	2944	2304	24	
95	2912	1832	24	3
94	2880	1360	24	2
93	2848	888	24	1
90	2768	4848	23	2
89	2736	4376	23	1
88	2704	3904	22	
87	2672	3432	22	3
86	2640	2960	22	2
85	2608	2488	22	1
84	2576	2016	21	
83	2544	1544	21	3
82	2512	1072	21	2
79	2432	5032	20	3
78	2400	4560	20	2
77	2368	4088	20	1
76	2336	3616	19	
75	2304	3144	19	3
74	2272	2672	19	2
73	2240	2200	19	1
72	2208	1728	18	
71	2176	1256	18	3
70	2144	784	18	2

。

21. 根据权利要求 17 所述的装置，其特征在于，m 为 32 的整数倍，q=10280，r，m，d_{CP}，p 和 r_1 的值为下表中的任意一组：

r	m	d_{CP}	p	r_1
102	3136	5136	26	2
101	3104	4664	26	1
100	3072	4192	25	
99	3040	3720	25	3
98	3008	3248	25	2
97	2976	2776	25	1
96	2944	2304	24	
95	2912	1832	24	3
94	2880	1360	24	2
93	2848	888	24	1
79	2432	5032	20	3
78	2400	4560	20	2
77	2368	4088	20	1
76	2336	3616	19	
75	2304	3144	19	3
74	2272	2672	19	2
73	2240	2200	19	1
72	2208	1728	18	
71	2176	1256	18	3
70	2144	784	18	2

。

22. 根据权利要求 13-21 中任一项所述的装置，其特征在于，所述第三比特子集包括 1504 比特，所述第一比特子集包括 m-1504 比特；所述 PCS 单元用于：

将所述第一比特子集分成 16 组序列，每组序列包括(m-1504)/16 个比特；

分别对所述 16 组序列进行 PCS 子处理，得到与所述 16 组序列对应的 16 组输出序列，其中，每组输出序列包括 128 比特，所述 16 组输出序列构成所述第二比特子集；

或者，所述 PCS 单元用于：

将所述第一比特子集分成 32 组序列，每组序列包括(m-1504)/32 个比特；

分别对所述 32 组序列进行 PCS 子处理，得到与所述 32 组序列对应的 32 组输出序列，其中，每组输出序列包括 64 比特，所述 32 组输出序列构成所述第二比特子集。

23. 根据权利要求 13-22 中任一项所述的装置，其特征在于，所述数据处理装置还包括第二处理单元和数字信号处理 DSP 成帧单元，

所述第二处理单元，用于对第四比特数据进行符号映射和偏振划分得到双偏振符号序列，其中，所述符号映射和所述偏振划分将 8 个比特映射为 1 个双偏振 DP-16QAM 符号，所述第四比特数据包括多个所述第二比特集合；

所述 DSP 成帧单元，用于对所述双偏振符号序列进行 DSP 成帧，得到 DSP 帧。

24. 根据权利要求 23 所述的装置，其特征在于，所述 DSP 成帧单元，用于将每 172032 个双偏振 DP-16QAM 符号进行成帧处理得到一个 DSP 帧。

25. 一种芯片，其特征在于，所述芯片包括处理器，所述处理器用于执行如权利要求 1 至 12 中任一项所述的方法。

26. 一种光模块，其特征在于，所述光模块包括处理器和接口，所述处理器用于执行如权利要求 1 至 12 中任一项所述的方法，并通过所述接口发送信号。

27. 根据权利要求 26 所述的光模块，其特征在于，所述处理器用于对第二比特集合进行数据处理得到 DSP 帧，并通过所述接口发送所述 DSP 帧。

28. 根据权利要求 26 所述的光模块，其特征在于，所述光模块还包括调制器，所述处理器用于对第二比特集合进行数据处理得到 DSP 帧，所述调制器用于根据所述 DSP 帧进行电光转换得到光信号，并通过所述接口发送所述光信号。

29. 一种发送设备，其特征在于，所述发送设备包括主机侧设备和如权利要求 26 至 28 中任一项所述的光模块，所述光模块用于根据来自所述主机侧设备的数据生成光信号，并发送所述光信号。

30. 一种通信系统，其特征在于，包括：如权利要求 29 所述的发送设备和接收设备，所述发送设备用于向所述接收设备发送光信号。