

Báo cáo nghiên cứu tìm hiểu về tuyến kênh Uplink 5G NR Sub 6

Yêu cầu: Báo cáo nghiên cứu, tìm hiểu tuyến kênh Uplink 5G NR Sub 6 bao gồm kênh PRACH, PUCCH, PUSCH, tín hiệu SRS

1. Kênh PRACH (Physical Random Access Channel)

1.1. Tổng quan kênh PRACH

- NR-PRACH cung cấp kênh vật lý qua đó yêu cầu truy cập ngẫu nhiên được truyền từ UE đến network
- Kênh đảm bảo các yêu cầu được nhận và xử lý một cách chính xác

1.2. Cấu trúc kênh PRACH và thuật toán của các module trong kênh PRACH

* Sequence generation

- Tập hợp của random-access preambles $x_{u,v}(n)$ được tạo bởi thông qua

$$x_{u,v}(n) = x_u((n + C_v) \bmod L_{RA})$$

$$x_i(n) = e^{-\frac{j\pi u i(i+1)}{L_{RA}}}, i = 0, 1, \dots, L_{RA} - 1$$

Từ đó, biểu diễn miền tần số được tạo bởi

$$y_{u,v}(n) = \sum_{m=0}^{L_{RA}-1} x_{u,v}(n) e^{-\frac{2j\pi mn}{L_{RA}}}$$

với $L_{RA} \in \{839, 139, 571, 1151\}$ phụ thuộc vào PRACH preamble format được cho bởi Bảng 6.3.3.1-1 và 6.3.3.1-2.

Table 6.3.3.1-1: PRACH preamble formats for $L_{RA} = 839$ and $\Delta f_{RA} \in \{1.25, 5\}$ kHz.

Format	L_{RA}	Δf_{RA}	N_u	N_{CP}^{RA}	Support for restricted sets
0	839	1.25 kHz	24576κ	3168κ	Type A, Type B
1	839	1.25 kHz	$2 \cdot 24576\kappa$	21024κ	Type A, Type B
2	839	1.25 kHz	$4 \cdot 24576\kappa$	4688κ	Type A, Type B
3	839	5 kHz	$4 \cdot 6144\kappa$	3168κ	Type A, Type B

Table 6.3.3.1-2: Preamble formats for $L_{RA} \in \{139, 571, 1151\}$ and $\Delta f_{RA} = 15 \cdot 2^\mu$ kHz where $\mu \in \{0, 1, 2, 3\}$.

Format	L_{RA}			Δf_{RA}	N_u	N_{CP}^{RA}	Support for restricted sets
	$\mu \in \{0, 1, 2, 3\}$	$\mu = 0$	$\mu = 1$				
A1	139	1151	571	$15 \cdot 2^\mu$ kHz	$2 \cdot 2048\kappa \cdot 2^{-\mu}$	$288\kappa \cdot 2^{-\mu}$	-
A2	139	1151	571	$15 \cdot 2^\mu$ kHz	$4 \cdot 2048\kappa \cdot 2^{-\mu}$	$576\kappa \cdot 2^{-\mu}$	-
A3	139	1151	571	$15 \cdot 2^\mu$ kHz	$6 \cdot 2048\kappa \cdot 2^{-\mu}$	$864\kappa \cdot 2^{-\mu}$	-
B1	139	1151	571	$15 \cdot 2^\mu$ kHz	$2 \cdot 2048\kappa \cdot 2^{-\mu}$	$216\kappa \cdot 2^{-\mu}$	-
B2	139	1151	571	$15 \cdot 2^\mu$ kHz	$4 \cdot 2048\kappa \cdot 2^{-\mu}$	$360\kappa \cdot 2^{-\mu}$	-
B3	139	1151	571	$15 \cdot 2^\mu$ kHz	$6 \cdot 2048\kappa \cdot 2^{-\mu}$	$504\kappa \cdot 2^{-\mu}$	-
B4	139	1151	571	$15 \cdot 2^\mu$ kHz	$12 \cdot 2048\kappa \cdot 2^{-\mu}$	$936\kappa \cdot 2^{-\mu}$	-
C0	139	1151	571	$15 \cdot 2^\mu$ kHz	$2048\kappa \cdot 2^{-\mu}$	$1240\kappa \cdot 2^{-\mu}$	-
C2	139	1151	571	$15 \cdot 2^\mu$ kHz	$4 \cdot 2048\kappa \cdot 2^{-\mu}$	$2048\kappa \cdot 2^{-\mu}$	-

- Có 64 preamble được định nghĩa trong mỗi time-frequency PRACH occasion, được đánh số theo thứ tự tăng của increasing cyclic shift C_v đầu tiên của logical root sequence, và sau đó theo thứ tự tăng của hệ số logical root sequence, bắt đầu từ hệ số thu được từ tham số lớp cao hơn $prach\text{-}RootSequenceIndex$ hoặc $rootSequenceIndex\text{-}BFR$ hoặc bởi $msgA\text{-}PRACH\text{-}RootSequenceIndex$ nếu được cấu hình và một type-2 random-access procedure được khởi tạo.

- Trong trường hợp 64 preambles không được tạo từ một chuỗi root Zadoff-Chu đơn, một chuỗi preamble bổ sung được tạo từ chuỗi ban đầu với hệ số logic liên tiếp đến khi tất cả 64 chuỗi được tìm thấy. Thứ tự logical root sequence là tuần hoàn, hệ số logic 0 là liên tục tới $L_{RA} - 2$. Số chuỗi u được thu từ logical root sequence index trong bảng 6.3.3.1-3 to 6.3.3.1-4B

Table 6.3.3.1-3: Mapping from logical index i to sequence number u for preamble formats with $L_{RA} = 839$.

i	Sequence number u in increasing order of i																			
0 – 19	129	710	140	699	120	719	210	629	188	671	84	755	105	734	93	748	70	769	60	779
20 – 39	2	837	1	838	58	783	112	727	148	691	80	759	42	797	40	799	35	804	73	768
40 – 59	146	693	31	808	28	811	30	809	27	812	29	810	24	815	48	791	68	771	74	765
60 – 79	178	661	136	703	86	753	78	761	43	796	39	800	20	819	21	818	95	744	202	637
80 – 99	190	649	181	658	137	702	125	714	151	688	217	622	128	711	142	697	122	717	203	638
100 – 119	118	721	110	729	89	750	103	736	61	778	55	784	15	824	14	825	12	827	23	816
120 – 139	34	805	37	802	46	793	207	632	179	660	145	694	130	709	223	616	228	611	227	612
140 – 159	132	707	133	706	143	698	135	704	161	678	201	638	173	668	106	733	83	756	91	748
160 – 179	66	773	53	786	10	829	9	830	7	832	8	831	16	823	47	792	64	775	57	782
180 – 199	104	735	101	738	108	731	208	631	184	655	197	642	191	648	121	718	141	698	149	690
200 – 219	216	623	218	621	152	687	144	695	134	705	138	701	199	640	162	677	176	663	119	720
220 – 239	158	681	164	675	174	665	171	668	170	669	87	752	169	670	88	751	107	732	81	758
240 – 259	82	757	100	739	98	741	71	768	59	780	65	774	50	789	49	790	26	813	17	822
260 – 279	13	826	6	833	5	834	33	806	51	788	75	784	99	740	96	743	97	742	166	673
280 – 299	172	667	175	664	187	652	163	676	185	654	200	639	114	725	189	650	115	724	194	645
300 – 319	195	644	192	647	182	657	157	682	156	683	211	628	154	685	123	716	139	700	212	627
320 – 339	153	688	213	626	215	624	150	689	225	614	224	615	221	618	220	619	127	712	147	692
340 – 359	124	715	193	646	205	634	206	633	116	723	160	679	186	653	167	672	79	760	85	754
360 – 379	77	762	92	747	58	781	62	777	69	770	54	785	36	803	32	807	25	814	18	821
380 – 399	11	828	4	835	3	836	19	820	22	817	41	798	38	801	44	795	52	787	45	794
400 – 419	63	776	67	772	72	767	76	763	94	745	102	737	90	749	109	730	165	674	111	728
420 – 439	209	630	204	635	117	722	188	651	159	680	198	641	113	726	183	656	180	659	177	662
440 – 459	196	643	155	684	214	625	126	713	131	708	219	620	222	617	226	613	230	609	232	607
460 – 479	262	577	252	587	418	421	416	423	413	426	411	428	376	463	395	444	283	556	285	554
480 – 499	379	460	390	449	363	476	384	455	388	451	386	453	361	478	387	452	360	479	310	529
500 – 519	354	485	328	511	315	524	337	502	349	490	335	504	324	515	323	516	320	519	334	505
520 – 539	359	480	295	544	385	454	292	547	291	548	381	458	399	440	380	459	397	442	369	470
540 – 559	377	462	410	429	407	432	281	558	414	425	247	592	277	562	271	568	272	567	264	575
560 – 579	259	580	237	602	239	600	244	595	243	596	275	564	278	561	250	589	246	593	417	422
580 – 599	248	591	394	445	393	446	370	469	365	474	300	539	299	540	364	475	362	477	298	541
600 – 619	312	527	313	526	314	525	353	486	352	487	343	496	327	512	350	489	326	513	319	520
620 – 639	332	507	333	506	348	491	347	492	322	517	330	509	338	501	341	498	340	499	342	497
640 – 659	301	538	366	473	401	438	371	468	408	431	375	464	249	590	269	570	238	601	234	605
660 – 679	257	582	273	566	255	584	254	585	245	594	251	588	412	427	372	467	282	557	403	436
680 – 699	366	443	392	447	391	448	382	457	389	450	294	545	297	542	311	528	344	495	345	494
700 – 719	318	521	331	508	325	514	321	518	346	493	339	500	351	488	306	533	289	550	400	439
720 – 739	378	461	374	465	415	424	270	569	241	598	231	608	260	579	268	571	276	563	409	430
740 – 759	398	441	290	549	304	535	308	531	358	481	316	523	293	546	288	551	284	555	368	471
760 – 779	253	586	256	583	263	576	242	597	274	565	402	437	383	456	357	482	329	510	317	522
780 – 799	307	532	285	553	287	552	266	573	261	578	236	603	303	536	356	483	355	484	405	434
800 – 819	404	435	406	433	235	604	267	572	302	537	309	530	265	574	233	606	367	472	296	543
820 – 837	336	503	305	534	373	466	280	559	279	560	419	420	240	599	258	581	229	610	-	-

Table 6.3.3.1-4B: Mapping from *logical index i* to sequence number *u* for preamble formats with $L_{RA} = 571$.

<i>i</i>	Sequence number <i>u</i> in increasing order of <i>i</i>															
0-19	1	570	2	569	3	568	4	567	5	566	6	565	7	564	8	563
20-39	11	560	12	559	13	558	14	557	15	556	16	555	17	554	18	553
40-59	21	550	22	549	23	548	24	547	25	546	26	545	27	544	28	543
60-79	31	540	32	539	33	538	34	537	35	536	36	535	37	534	38	533
80-99	41	530	42	529	43	528	44	527	45	526	46	525	47	524	48	523
100-119	51	520	52	519	53	518	54	517	55	516	56	515	57	514	58	513
120-139	61	510	62	509	63	508	64	507	65	506	66	505	67	504	68	503
140-159	71	500	72	499	73	498	74	497	75	496	76	495	77	494	78	493
160-179	81	490	82	489	83	488	84	487	85	486	86	485	87	484	88	483
180-199	91	480	92	479	93	478	94	477	95	476	96	475	97	474	98	473
200-219	101	470	102	469	103	468	104	467	105	466	106	465	107	464	108	463
220-239	111	460	112	459	113	458	114	457	115	456	116	455	117	454	118	453
240-259	121	450	122	449	123	448	124	447	125	446	126	445	127	444	128	443
260-279	131	440	132	439	133	438	134	437	135	436	136	435	137	434	138	433
280-299	141	430	142	429	143	428	144	427	145	426	146	425	147	424	148	423
300-319	151	420	152	419	153	418	154	417	155	416	156	415	157	414	158	413
320-339	161	410	162	409	163	408	164	407	165	406	166	405	167	404	168	403
340-359	171	400	172	399	173	398	174	397	175	396	176	395	177	394	178	393
360-379	181	390	182	389	183	388	184	387	185	386	186	385	187	384	188	383
380-399	191	380	192	379	193	378	194	377	195	376	196	375	197	374	198	373
400-419	201	370	202	369	203	368	204	367	205	366	206	365	207	364	208	363
420-439	211	360	212	359	213	358	214	357	215	356	216	355	217	354	218	353
440-459	221	350	222	349	223	348	224	347	225	346	226	345	227	344	228	343
460-479	231	340	232	339	233	338	234	337	235	336	236	335	237	334	238	333
480-499	241	330	242	329	243	328	244	327	245	326	246	325	247	324	248	323
500-519	251	320	252	319	253	318	254	317	255	316	256	315	257	314	258	313
520-539	261	310	262	309	263	308	264	307	265	306	266	305	267	304	268	303
540-559	271	300	272	299	273	298	274	297	275	296	276	295	277	294	278	293
560-569	281	290	282	289	283	288	284	287	285	286	-	-	-	-	-	-

- Cyclic shift C_v được cho bởi

+ Với unrestricted sets

$$C_v = \begin{cases} vN_{CS}, & v = 0, 1, \dots, [L_{RA}/N_{CS}] - 1, N_{CS} \text{ khác } 0 \\ 0, & N_{CS} = 0 \end{cases}$$

+ Với restricted sets loại A và B

$$C_v = d_{\text{start}}[v/N_{\text{shift}}^{\text{RA}}] + (v \bmod N_{\text{shift}}^{\text{RA}})N_{CS}, \quad v = 0, 1, \dots, w - 1$$

+ Với restricted sets loại B

$$C_v = \begin{cases} d_{\text{start}} + (v - w)N_{CS}, & v = w, \dots, w + N_{\text{shift}}^{\text{RA}} - 1 \\ d_{\text{start}} + (v - w - N_{\text{shift}}^{\text{RA}})N_{CS}, & v = w + N_{\text{shift}}^{\text{RA}}, \dots, w + N_{\text{shift}}^{\text{RA}} + N_{\text{shift}}^{\text{RA}} - 1 \end{cases}$$

$$w = N_{\text{shift}}^{\text{RA}}N_{\text{group}}^{\text{RA}} + N_{\text{shift}}^{\text{RA}}$$

với N_{CS} được cho bởi bảng 6.3.3.1-5 to 6.3.3.1-7, tham số lớp cao hơn *msgA-RestrictedSetConfig*, quyết định loại restricted sets; nếu không thì, tham số lớp cao hơn *restrictedSetConfig* quyết định loại restricted sets và Bảng 6.3.3.1-1 và 6.3.3.1-2 loại restricted sets hỗ trợ cho preamble formats khác nhau

- Giá trị d_u được cho bởi

$$d_u = \begin{cases} q, & 0 < q < L_{RA}/2 \\ L_{RA} - q, & q \text{ khác} \end{cases}$$

với q là số nguyên không âm nhỏ nhất thỏa mãn $(qu) \bmod L_{RA} = 1$. Tham số cho restricted sets của dịch chuyển tuần hoàn phụ thuộc vào d_u

* Mapping to physical resources

- Preamble sequence sẽ được ánh xạ tới tài nguyên vật ý

$$a_k^{(p, \text{RA})} = \beta_{\text{PRACH}} y_{u,v}(k), k = 0, 1, \dots, L_{RA} - 1$$

với β_{PRACH} là hệ số tỷ lệ biên độ (amplitude scaling factor) để tuân theo transmit power và $p=4000$ là công suất anten. Baseband signal generation sẽ theo phần và sử dụng các tham số trong Bảng 6.3.3.1-1 hoặc Bảng 6.3.3.1-2 với k được cho bởi Bảng 6.3.3.2-1

- Random access preambles có thể chỉ được truyền trong tài nguyên thời gian cho bởi bảng 6.3.3.2-2 tới Bảng 6.3.3.2-4 và phụ thuộc vào FR1 hoặc FR2 và loại phổ. Chỉ số cấu hình PRACH trong Bảng 6.3.3.2-2 tới Bảng 6.3.3.2-4

+ Với Bảng 6.3.3.2-3 được cho bởi tham số lớp cao hơn *prach-ConfigurationIndex*, hoặc bởi *msgA-PRACH-ConfigurationIndex* nếu được cấu hình

+ Với Bảng 6.3.3.2-2 và Bảng 6.3.3.2-4 được cho bởi tham số lớp cao hơn *prach-ConfigurationIndex*, hoặc bởi *msgA-PRACH-ConfigurationIndex* nếu được cấu hình

- Với IAB-MT part của một IAB-node,

+ Nếu *prach-ConfigurationPeriodScaling-IAB* được cấu hình, biến x được dùng trong $n_f \bmod x = y$ của Bảng 6.3.3.2-2 tới Bảng 6.3.3.2-4 sẽ được thay bằng $x_{\text{IAB}} = \delta x$ và δ được cho bởi tham số lớp cao hơn *prach-ConfigurationPeriodScaling-IAB* và IAB-node để $x_{\text{IAB}} > 64$

+ Nếu *prach-ConfigurationFrameOffset-IAB* được cấu hình, biến y được dùng trong $n_f \bmod x = y$ của Bảng 6.3.3.2-2 tới Bảng 6.3.3.2-4 sẽ được thay bằng $x_{\text{IAB}} = (y + \Delta y) \bmod x$ và Δy được cho bởi tham số lớp cao hơn *prach-ConfigurationFrameOffset-IAB* và biến x được dùng trong $n_f \bmod x = y$

+ Nếu *prach-ConfigurationSOffset-IAB* được cấu hình, số subframe s_n từ Bảng 6.3.3.2-2 tới Bảng 6.3.3.2-3 và số slot s_n từ Bảng 6.3.3.2-4 sẽ được thay bằng $(s_n + \Delta s) \bmod L$ với $\Delta s \in \{0, 1, \dots, L - 1\}$ được cho bởi tham số lớp cao hơn *prach-ConfigurationSOffset-IAB* và L là số subframe trong frame khi dùng Bảng 6.3.3.2-2 tới Bảng 6.3.3.2-3 và số slot trong 1 frame cho khoảng cách sóng mang con 60kHz khi dùng Bảng 6.3.3.2-4

- Random access preambles có thể chỉ được truyền trong tài nguyên tần số được cho bởi tham số lớp cao hơn *msg1-FrequencyStart* hoặc *msgA-RO-FrequencyStart* nếu được cấu hình. PRACH frequency resources $n_{\text{RA}} \in \{0, 1, \dots, M-1\}$ với M bằng tham số lớp cao hơn *msg1-FDM* hoặc *msgA-RO-FDM* nếu được cấu hình, và được đánh số theo thứ tự tăng trong initial uplink bandwidth part trong suốt khởi tạo ban đầu, bắt đầu từ tần số thấp nhất. Nếu không thì, n_{RA} được đánh số theo thứ tự tăng trong active uplink bandwidth part trong suốt khởi tạo ban đầu, bắt đầu từ tần số thấp nhất.

- Với quy trình với truy cập kênh chia sẻ phổ, với $L_{\text{RA}} = 139$, một UE được cung cấp bởi tham số lớp cao hơn *msg1-FrequencyStart* hoặc *msgA-RO-FrequencyStart* nếu được cấu hình, và tham số lớp cao hơn *msg1-FDM* hoặc *msgA-RO-FDM* nếu được cấu hình, một random access preamble được giữ với 1 RB set đơn. RB set được định nghĩa là khi UE không được cung cấp *intraCellGuardBandsPerSCS* cho 1 sóng mang UE

- Với quy trình với truy cập kênh chia sẻ phổ, với $L_{\text{RA}} = 571$ hoặc 1151 và Type-2 random access, UE được cung cấp với tham số lớp cao hơn *msgA-RO-FDM* bằng 1

- Khoảng cách sóng mang: 15kHz cho FR1 và 60kHz cho FR2

- Với mục đích chuyển giao handover để 1 target cell trong paired hoặc unpaired spectrum với target cell sử dụng $L_{\text{max}}=4$, trị tuyệt đối của khác nhau thời gian giữa radio frame i tại cell hiện tại và radio frame i tại target cell ít hơn $1536000T_s$, nếu chu kỳ association pattern không bằng 10ms

- Với mục đích inter frequency handover với source cell là paired hoặc unpaired spectrum và target cell trong unpaired spectrum và sử dụng $L_{\max}=8$, trị tuyệt đối của khác nhau thời gian giữa radio frame i tại cell hiện tại và radio frame i tại target cell ít hơn $76800T_S$

* OFDM baseband signal generation for PRACH

- Tín hiệu thời gian liên tục $s_1^{(p,\mu)}(t)$ tại antenna port p cho kênh PRACH

$$s_1^{(p,\mu)}(t) = \sum_{k=0}^{L_{RA}-1} \alpha_k^{(p,RA)} e^{j2\pi(k + Kk_1 + \bar{k})\Delta f_{RA} (t - N_{CP,l}^{RA}T_C - t_{start}^{RA})}$$

$$K = \Delta f / \Delta f_{RA}$$

$$k_1 = k_0^\mu + (N_{BWP,i}^{start} - N_{grid}^{start})N_{SC}^{RB} - N_{grid}^{size,\mu}N_{SC}^{RB}/2 + n_{RA}^{start}N_{SC}^{RB}$$

$$+ \begin{cases} n_{RA}N_{RB}^{RA}N_{SC}^{RB}, & \text{nếu } L_{RA} \in \{139, 839\} \\ (N_{RB,UL,n_0+n_{RA}}^{start,\mu} - N_{RB,UL,n_0}^{start,\mu})N_{SC}^{RB}, & \text{nếu } L_{RA} \in \{571, 1151\} \end{cases}$$

$$k_0^\mu = (N_{grid,x}^{start,\mu} + N_{grid,x}^{size,\mu}/2)N_{SC}^{RB} - (N_{grid,x}^{start,\mu_0} + N_{grid,x}^{size,\mu_0}/2)N_{SC}^{RB}2^{\mu_0-\mu}$$

với $t_{start}^{RA} \leq t < t_{start}^{RA} + T_C(N_u + N_{CP,l}^{RA})$ và

+ \bar{k} được cho bởi bảng 6.3.3.2-1

Table 6.3.3.2-1: Supported combinations of Δf_{RA} and Δf , and the corresponding value of \bar{k} .

L_{RA}	Δf_{RA} for PRACH	Δf for PUSCH	N_{RB}^{RA} , allocation expressed in number of RBs for PUSCH	\bar{k}
839	1.25	15	6	7
839	1.25	30	3	1
839	1.25	60	2	133
839	5	15	24	12
839	5	30	12	10
839	5	60	6	7
139	15	15	12	2
139	15	30	6	2
139	15	60	3	2
139	30	15	24	2
139	30	30	12	2
139	30	60	6	2
139	60	60	12	2
139	60	120	6	2
139	120	60	24	2
139	120	120	12	2
571	30	15	96	2
571	30	30	48	2
571	30	60	24	2
1151	15	15	96	1
1151	15	30	48	1
1151	15	60	24	1

+ Δf là khoảng cách sóng mang con của initial uplink bandwidth part trong suốt truy nhập khởi đầu. Nếu không thì Δf là khoảng cách sóng mang con của active uplink bandwidth part

+ μ_0 là giá trị lớn nhất của μ trong các cấu hình khoảng cách sóng mang con bởi thành phần lớp cao *scs-SpecificCarrierList*

+ $N_{BWP,i}^{start}$ là RB được đánh số nhỏ nhất của initial uplink bandwidth part và được vận chuyển bởi tham số lớp cao hơn *initialUplinkBWP* trong suốt truy nhập khởi đầu (initial access). Nếu không thì $N_{BWP,i}^{start}$ là RB được đánh số nhỏ nhất của active uplink bandwidth part và được vận chuyển bởi tham số lớp cao hơn *initialUplinkBWP* trong suốt truy nhập khởi đầu (initial access)

+ n_{RA}^{start} là tần số offset của lowest PRACH transmission occasion trong miền tần số với physical RB 0 của active uplink bandwidth part. Số lượng n_{RA}^{start} được cho bởi tham số lớp cao hơn *msgA-RO-*

FrequencyStart nếu được cấu hình và một type-2 random-access procedure được khởi tạo, nếu không thì bởi *msg1-FrequencyStart*

+ n_{RA} là chỉ số PRACH transmission occasion trong miền tần số cho một PRACH transmission occasion trong một time instance

+ N_{RB}^{RA} là số RB và được cho bởi phân bố thành phần biểu diễn trong số RBs của kênh PUSCH trong bảng 6.3.3.2-1

+ $N_{RB,UL,n}^{start,\mu}$ là chỉ số CRB bắt đầu của uplink RB set n tương ứng với số $RB_{n,UL}^{start,\mu}$. RB set được định nghĩa là khi UE không được cung cấp *IntraCellGuardBandsPerSCS* cho một sóng mang UL

+ n_0 là chỉ số của RB set bao gồm PRACH transmission occasion thấp nhất trong miền tần số được biểu diễn bởi n_{RA}^{start} . n_{RA}^{start} được cấu hình như mỗi PRACH transmission occasion được gồm đủ trong 1 RB set

+ L_{RA} và N_u được cho bởi bảng 6.3.3.2-1

+ $N_{CP,l}^{RA} = N_{CP}^{RA} + n \cdot 16k$ với

- nếu $\Delta f_{RA} \in \{1.25, 5\}kHz$, $n=0$

- nếu $\Delta f_{RA} \in \{15, 30, 60, 120\}kHz$, n là số lần interval $(t_{RA}^{start}, t_{RA}^{start} + (N_u^{RA} + N_{CP}^{RA})T_C)$ overlap với time instance 0 khác hoặc time instance $(\Delta f_{max} N_t / 2000) \cdot T_C = 0.5$ ms trong 1 subframe

- Vị trí bắt đầu t_{start}^{RA} của PRACH preamble trong 1 subframe (với $\Delta f_{RA} \in \{1.25, 5, 15, 30\}kHz$) hoặc trong 1 slot 60kHz (với $\Delta f_{RA} \in \{60, 120\}kHz$) tại $t=0$ và $N_{TA} = 0$ được cho bởi

$$t_{start}^{RA} = t_{start,l}^{\mu}$$

$$t_{start,l}^{\mu} = \begin{cases} 0, & l = 0 \\ t_{start,l-1}^{\mu} + (N_u^{\mu} + N_{CP,l-1}^{\mu}) \cdot T_C, & \text{khác} \end{cases}$$

với

$$N_u^{\mu} = 2048k \cdot 2^{-\mu}$$

$$N_{CP,l}^{\mu} = \begin{cases} 512k \cdot 2^{-\mu} & \text{extended CP} \\ 144k \cdot 2^{-\mu} + 16 & \text{normal CP, } l = 0 \text{ or } l = 7.2^{\mu} \\ 144k \cdot 2^{-\mu} & \text{normal CP, } l \neq 0 \text{ or } l \neq 7.2^{\mu} \end{cases}$$

+ $\mu = 0$ với nếu $\Delta f_{RA} \in \{1.25, 5\}kHz$, nếu không thì được cho bởi $\Delta f_{RA} \in \{15, 30, 60, 120\}kHz$ và vị trí ký tự l được cho bởi

$$l = l_0 + n_t^{RA} n_{dur}^{RA} + 14n_{slot}^{RA}$$

với

+ l_0 và n_{dur}^{RA} được cho bởi bảng 6.3.3.2-2 to 6.3.3.2-4

+ n_t^{RA} là PRACH transmission occasion trong PRACH slot, được đánh số tăng từ 0 đến $N_t^{RA,slot} - 1$ trong 1 RACH slot với $N_t^{RA,slot}$ được cho bởi bảng 6.3.3.2-2 to 6.3.3.2-4 với $L_{RA} \in \{139, 571, 1151\}$ và được fix thành 1 với $L_{RA} = 839$

+ n_{slot}^{RA} được cho bởi

- nếu $\Delta f_{RA} \in \{1.25, 5\}kHz$, $n_{slot}^{RA} = 0$

- nếu $\Delta f_{RA} \in \{15, 30, 60, 120\}kHz$ và khác "Number of PRACH slots within a subframe" trong Bảng 6.3.3.2-2 to 6.3.3.2-3 hoặc khác "Number of PRACH slots within a 60 kHz slot" trong Bảng 6.3.3.2-4 bằng 1 thì $n_{slot}^{RA} = 1$

- nếu không thì $n_{slot}^{RA} \in \{0, 1\}$

- Nếu preamble format được cho bởi Bảng 6.3.3.2-2 to 6.3.3.2-4 là A1/B1, A2/B2 hoặc A3/B thì

+ Nếu $n_t^{RA} = N_t^{RA,slot} - 1$ thì PRACH preamble tương ứng với PRACH preamble format từ B1, B2 và B3 được truyền đi trong PRACH transmission occasion

+ Nếu không thì PRACH preamble tương ứng với PRACH preamble format từ A1, A2 và A3 được truyền đi trong PRACH transmission occasion

2. Kênh PUCCH (Physical Uplink Control Channel)

2.1. Tổng quan kênh PUCCH

- Kênh PUCCH được dùng để truyền UCI (Uplink Control Information): HARQ feedback, CSI (Channel State Information) và SR (Scheduling Request).

2.2. Cấu trúc kênh PUCCH và thuật toán của các module trong kênh PUCCH

- UCI bit sequence generation

+ Chỉ HARQ-ACK/SR

- Nếu chỉ các bit HARQ-ACK được truyền đi trên PUCCH, chuỗi bit UCI $a_0, a_1, a_2, \dots, a_{A-1}$ được xác định bởi thiết lập $a_i = \tilde{o}_i^{ACK}$ với $i=0,1,\dots, O^{ACK} - 1$ và $A=O^{ACK}$, với chuỗi bit HARQ-ACK $\tilde{o}_0^{ACK}, \tilde{o}_1^{ACK}, \dots, \tilde{o}_{O^{ACK}-1}^{ACK}$
- Nếu chỉ các bit HARQ-ACK và SR được truyền trên PUCCH, chuỗi bit UCI $a_0, a_1, a_2, \dots, a_{A-1}$ được xác định bởi thiết lập $a_i = \tilde{o}_i^{ACK}$ với $i=0,1,\dots, O^{ACK} - 1$, $a_i = \tilde{o}_i^{SR}$ với $i=O^{ACK}, O^{ACK} + 1, \dots, O^{ACK} + O^{SR} - 1$ và $A=O^{ACK} + O^{SR}$ với chuỗi bit HARQ-ACK $\tilde{o}_0^{ACK}, \tilde{o}_1^{ACK}, \dots, \tilde{o}_{O^{ACK}-1}^{ACK}$ và chuỗi bit SR $\tilde{o}_0^{SR}, \tilde{o}_1^{SR}, \dots, \tilde{o}_{O^{SR}-1}^{SR}$

+ Chỉ CSI

- Bitwidth cho PMI của *codebookType=typeI-SinglePannel* với 2 cổng CSI-RS là 2 với Rank=1 và 1 với Rank=2
- Bitwidth cho PMI của *codebookType=typeI-SinglePannel* với nhiều hơn 2 cổng CSI-RS được cho bởi Bảng 6.3.1.1.2-1 với giá trị (N_1, N_2) và (O_1, O_2) được cho bởi Bảng 5.2.2.2.1-2

Table 5.2.2.2.1-2: Supported configurations of (N_1, N_2) and (O_1, O_2)

Number of CSI-RS antenna ports, $P_{\text{CSI-RS}}$	(N_1, N_2)	(O_1, O_2)
4	(2,1)	(4,1)
8	(2,2)	(4,4)
	(4,1)	(4,1)
12	(3,2)	(4,4)
	(6,1)	(4,1)
16	(4,2)	(4,4)
	(8,1)	(4,1)
24	(4,3)	(4,4)
	(6,2)	(4,4)
	(12,1)	(4,1)
32	(4,4)	(4,4)
	(8,2)	(4,4)
	(16,1)	(4,1)

- Bitwidth cho PMI của *codebookType=typeI-MultiPanel* được cho bởi Bảng 6.3.1.1.2-2 với giá trị (N_g, N_1, N_2) và (O_1, O_2) được cho bởi Bảng 5.2.2.2.2-1

Table 5.2.2.2-1: Supported configurations of (N_g, N_1, N_2) and (O_1, O_2)

Number of CSI-RS antenna ports, $P_{\text{CSI-RS}}$	(N_g, N_1, N_2)	(O_1, O_2)
8	(2,2,1)	(4,1)
16	(2,4,1)	(4,1)
	(4,2,1)	(4,1)
	(2,2,2)	(4,4)
32	(2,8,1)	(4,1)
	(4,4,1)	(4,1)
	(2,4,2)	(4,4)
	(4,2,2)	(4,4)

- Bitwidth cho PMI với 1 cổng CSI-RS là 0
- Bitwidth cho RI/LI/CQI/CRI của *codebookType=typeI-SinglePanel* được cho bởi Bảng 6.3.1.1.2-3

Table 6.3.1.1.2-3: RI, LI, CQI, and CRI of *codebookType=typeI-SinglePanel*

Field	Bitwidth				
	1 antenna port	2 antenna ports	4 antenna ports	>4 antenna ports	
Rank Indicator	0	$\min(1, \lceil \log_2 n_{RI} \rceil)$	$\min(2, \lceil \log_2 n_{RI} \rceil)$	$\lceil \log_2 n_{RI} \rceil$	$\lceil \log_2 n_{RI} \rceil$
Layer Indicator	0	$\min(2, \lceil \log_2 RI \rceil)$	$\min(2, \lceil \log_2 RI \rceil)$	$\min(2, \lceil \log_2 RI \rceil)$	$\min(2, \lceil \log_2 RI \rceil)$
Wide-band CQI	4	4	4	4	8
Subband differential CQI	2	2	2	2	4
CRI	$\lceil \log_2 (K_i^{\text{CSI-RS}}) \rceil$	$\lceil \log_2 (K_i^{\text{CSI-RS}}) \rceil$	$\lceil \log_2 (K_i^{\text{CSI-RS}}) \rceil$	$\lceil \log_2 (K_i^{\text{CSI-RS}}) \rceil$	$\lceil \log_2 (K_i^{\text{CSI-RS}}) \rceil$

- Nếu tham số lớp cao hơn $nrofCQIsPerReport=1$, n_{RI} trong Bảng 6.3.1.1.2-3 là số giá trị chỉ thị hạng được cho phép trong 4 LBSs của tham số lớp cao hơn *typeI-SinglePanel-ri-Restriction* ; nếu không thì n_{RI} trong Bảng 6.3.1.1.2-3 là số giá trị chỉ thị hạng được cho phép. Giá trị $K_i^{\text{CSI-RS}}$ là số tài nguyên CSI-RS trong tập hợp tài nguyên tương ứng
- Bitwidth cho RI/LI/CQI/CRI của *codebookType=typeI-MultiPanel* được cho bởi Bảng 6.3.1.1.2-4

Table 6.3.1.1.2-4: RI, LI, CQI, and CRI of *codebookType=typeI-MultiPanel*

Field	Bitwidth
Rank Indicator	$\min(2, \lceil \log_2 n_{RI} \rceil)$
Layer Indicator	$\min(2, \lceil \log_2 RI \rceil)$
Wide-band CQI	4
Subband differential CQI	2
CRI	$\lceil \log_2 (K_i^{\text{CSI-RS}}) \rceil$

- Bitwidth cho RI/LI/CQI của *codebookType=typeII* or *codebookType=typeII-PortSelection* được cho bởi Bảng 6.3.1.1.2-5

Table 6.3.1.1.2-5: RI, LI, and CQI of *codebookType=typeII* or *typeII-PortSelection*

Field	Bitwidth
Rank Indicator	$\min(1, \lceil \log_2 n_{RI} \rceil)$
Layer Indicator	$\min(2, \lceil \log_2 RI \rceil)$
Wide-band CQI	4
Subband differential CQI	2
Indicator of the number of non-zero wideband amplitude coefficients M_l for layer l	$\lceil \log_2 (2L-1) \rceil$

- Bitwidth cho CRI, SSBRI, RSRP, và RSRP vi phân được cho bởi Bảng 6.3.1.1.2-6

Table 6.3.1.1.2-6: CRI, SSBRI, and RSRP

Field	Bitwidth
CRI	$\lceil \log_2(K_z^{CSI-RS}) \rceil$
SSBRI	$\lceil \log_2(K_z^{SSB}) \rceil$
RSRP	7
Differential RSRP	4

với K_z^{CSI-RS} là số tài nguyên CSI-RS trong tập tài nguyên tương ứng và K_z^{SSB} là số khối SS/PBCH được cấu hình trong tập tài nguyên tương ứng cho reporting ‘ssb-Index-RSRP’

+ HARQ-ACK/SR và CSI

- Nếu không CSI report cho truyền trên PUCCH của 2 phần, chuỗi bit UCI $a_0, a_1, a_2, \dots, a_{A-1}$ được tạo tuân theo phần sau đây, với $A = O^{ACK} + O^{SR} + O^{CSI}$
 - Nếu có truyền HARQ-ACK truyền trên PUCCH, các bit HARQ-ACK được ánh xạ tới chuỗi bit UCI $a_0, a_1, a_2, \dots, a_{O^{ACK}-1}$, với $a_i = \tilde{o}_i^{ACK}$ với $i=0,1,\dots, O^{ACK}-1$, chuỗi bit HARQ-ACK $\tilde{o}_0^{ACK}, \tilde{o}_1^{ACK}, \dots, \tilde{o}_{O^{ACK}-1}^{ACK}$, O^{ACK} là số bit HARQ-ACK, nếu không có HARQ-ACK truyền trên PUCCH thì đặt $O^{ACK} = 0$
 - Nếu có truyền SR truyền trên PUCCH, đặt $a_i = \tilde{o}_i^{SR}$ với $i=O^{ACK}, O^{ACK}+1, \dots, O^{ACK}+O^{SR}-1$, với chuỗi bit SR $\tilde{o}_0^{SR}, \tilde{o}_1^{SR}, \dots, \tilde{o}_{O^{SR}-1}^{SR}$, nếu không có SR truyền trên PUCCH thì đặt $O^{SR} = 0$
 - Miền CSI cho tất cả các CSI report, trong thứ tự từ upper part đến lower part trong Bảng 6.3.1.1.2-12 và được ánh xạ đến chuỗi bit UCI $a_{O^{ACK}+O^{SR}}, a_{O^{ACK}+O^{SR}+1}, a_{O^{ACK}+O^{SR}+O^{CSI}-1}$ bắt đầu với $a_{O^{ACK}+O^{SR}}$, với O^{CSI} là số bit CSI
- Nếu có tối thiểu 1 CSI report cho truyền trên PUCCH của 2 phần, 2 chuỗi bit UCI được tạo thành $a_0^{(1)}, a_1^{(1)}, \dots, a_{A^{(1)}-1}^{(1)}$ và $a_0^{(2)}, a_1^{(2)}, \dots, a_{A^{(2)}-1}^{(2)}$ tuân theo phần dưới, với $A^{(1)} = O^{ACK} + O^{SR} + O^{CSI-part1}$ và $A^{(2)} = O^{CSI-part2}$
 - Nếu có HARQ-ACK truyền trên PUCCH, các bit HARQ-ACK được ánh xạ tới chuỗi bit UCI $a_0^{(1)}, a_1^{(1)}, \dots, a_{O^{ACK}-1}^{(1)}$ với $a_i^{(1)} = \tilde{o}_i^{ACK}$

- Code block segmentation and CRC attachment

+ Chuỗi bit UCI được biểu diễn bởi $a_0, a_1, a_2, \dots, a_{A-1}$ với A là payload size.

- Với $A \geq 12$, UCI được giải mã bằng mã Palar
 - Nếu payload size $A \geq 12$, code block segmentation and CRC attachment được tiến hành theo phân mã hóa Polar. Nếu ($A \geq 360$ và $E \geq 1088$) hoặc $A \geq 1013$, $I_{seg}=1$; nếu không thì $I_{seg}=0$ với E là chiều dài chuỗi đầu ra rate matching
 - Nếu $12 \leq A \leq 19$, parity bits $p_{r0}, p_{r1}, p_{r2}, \dots, p_{r(L-1)}$ được tính toán bằng cách đặt L thành 6 bits và sử dụng chuỗi tạo sinh $g_{CRC6}(D)$, kết quả thu được chuỗi bits $c_{r0}, c_{r1}, c_{r2}, \dots, c_{r(K_r-1)}$ với r là số khối mã và K_r là số bit cho mỗi khối mã số r
 - Nếu $A \geq 20$, parity bits $p_{r0}, p_{r1}, p_{r2}, \dots, p_{r(L-1)}$ được tính toán bằng cách đặt L thành 11 bits và sử dụng chuỗi tạo sinh $g_{CRC11}(D)$, kết quả thu được chuỗi bits $c_{r0}, c_{r1}, c_{r2}, \dots, c_{r(K_r-1)}$ với r là số khối mã và K_r là số bit cho mỗi khối mã số r
- Với $A \leq 11$, UCI được giải mã bằng mã hóa kênh của chiều dài khối nhỏ
 - Nếu payload size $A \leq 11$, các bit CRC không được attach

- Chuỗi bit đầu ra được biểu diễn bởi $c_0, c_1, c_2, \dots, c_{K-1}$ với $c_i = c_a$ với $i=0,1,\dots,A-1$ và $K=A$

- Channel coding of UCI

+ UCI được giải mã bằng Polar code

- Các bit thông tin được chuyển đến khối mã hóa kênh. Chúng được biểu diễn bởi $c_{r0}, c_{r1}, c_{r2}, \dots, c_{r(K_r-1)}$ với r là số khối mã, K_r là số bit trong khối mã r . Tổng số khối mã được biểu diễn bằng C và mỗi khối mã được giải mã độc lập như sau
- Nếu $18 \leq K_r \leq 25$, các bit thông tin được giải mã qua mã hóa Polar bằng việc đặt $n_{\max}=10$, $I_{IL}=0$, $n_{PC}=3$, $n_{PC}^{wm}=1$ nếu $E_r-K_r+3 > 192$ và $n_{PC}^{wm} = 0$ nếu $E_r-K_r+3 \leq 192$ với E_r là chiều dài chuỗi đầu ra rate matching
- Nếu $K_r > 30$, chuỗi bit thông tin được mã hóa qua Polar coding bằng việc đặt $n_{\max}=10$, $I_{IL}=0$, $n_{PC}=0$, $n_{PC}^{wm}=0$
- Sau giả mã các bit được biểu diễn bằng $d_{r0}, d_{r1}, d_{r2}, \dots, d_{r(N_r-1)}$ với N_r là số bit trong số khối mã r

+ UCI được giải mã bởi mã hóa kênh chiều dài khối nhỏ

- Các bit thông tin được chuyển đến khối mã hóa kênh. Chúng được biểu diễn bởi $c_0, c_1, c_2, \dots, c_{(K-1)}$ với K là số số bit
- Sau giả mã các bit được biểu diễn bằng $d_0, d_1, d_2, \dots, d_{N-1}$ với N là số tổng số bit được mã hóa

- Rate matching

+ Với PUCCH format 2/3/4, tổng chiều dài chuỗi đầu ra rate matching E_{tot} được cho bởi Bảng 6.3.1.4-1, với $N_{\text{symb,UCI}}^{\text{PUCCH2}}$, $N_{\text{symb,UCI}}^{\text{PUCCH3}}$ và $N_{\text{symb,UCI}}^{\text{PUCCH4}}$ là số ký tự mang UCI lần lượt cho PUCCH format, $N_{\text{symb,UCI}}^{\text{PUCCH2}}$, $N_{\text{symb,UCI}}^{\text{PUCCH3}}$ là số PRBs được quyết định bởi UE lần lượt cho truyền PUCCH format 2/3 và $N_{\text{symb,UCI}}^{\text{PUCCH4}}$ là tỷ lệ lan truyền (spreading factor) cho PUCCH format 4

Table 6.3.1.4-1: Total rate matching output sequence length E_{tot}

PUCCH format	Modulation order	
	QPSK	$\pi/2$ -BPSK
PUCCH format 2	$16 \cdot N_{\text{symb,UCI}}^{\text{PUCCH2}} \cdot N_{\text{PRB}}^{\text{PUCCH2}}$	N/A
PUCCH format 3	$24 \cdot N_{\text{symb,UCI}}^{\text{PUCCH3}} \cdot N_{\text{PRB}}^{\text{PUCCH3}}$	$12 \cdot N_{\text{symb,UCI}}^{\text{PUCCH3}} \cdot N_{\text{PRB}}^{\text{PUCCH3}}$
PUCCH format 4	$24 \cdot N_{\text{symb,UCI}}^{\text{PUCCH4}} / N_{\text{SF}}^{\text{PUCCH4}}$	$12 \cdot N_{\text{symb,UCI}}^{\text{PUCCH4}} / N_{\text{SF}}^{\text{PUCCH4}}$

+ UCI được giải mã bằng Polar code

- Chuỗi bit đầu vào để rate matching $d_{r0}, d_{r1}, d_{r2}, \dots, d_{r(N_r-1)}$ với N_r là số bit trong số khối mã r
- Rate matching được biểu diễn trong phần Rate matching cho Polar code bằng cách đặt $I_{BIL} = 1$ và độ dài chuỗi đầu ra rate matching $E_r = \lceil E_{UCI} / C_{UCI} \rceil$ với C_{UCI} là số khối mã cho UCI và giá trị E_{UCI} được cho bởi Bảng 6.3.1.4.1-1.
- Chuỗi bit đầu ra sau khi rate matching được biểu diễn bởi $f_{r0}, f_{r1}, f_{r2}, \dots, f_{r(N_r-1)}$ với E_r là chiều dài của chuỗi đầu ra rate matching trong số khối mã r

Table 6.3.1.4.1-1: Rate matching output sequence length E_{UCI}

UCI(s) for transmission on a PUCCH	UCI for encoding	Value of E_{UCI}
HARQ-ACK	HARQ-ACK	$E_{UCI} = E_{\text{ack}}$
HARQ-ACK, SR	HARQ-ACK, SR	$E_{UCI} = E_{\text{ack}}$
CSI (CSI not of two parts)	CSI	$E_{UCI} = E_{\text{csi}}$
HARQ-ACK, CSI (CSI not of two parts)	HARQ-ACK, CSI	$E_{UCI} = E_{\text{ack}}$
HARQ-ACK, SR, CSI (CSI not of two parts)	HARQ-ACK, SR, CSI	$E_{UCI} = E_{\text{ack}}$
CSI (CSI of two parts)	CSI part 1	$E_{UCI} = \min(E_{\text{csi}}, \lceil (O^{\text{CSI-part1}} + L) / R_{\text{UCI}}^{\text{max}} / Q_m \rceil \cdot Q_m)$
	CSI part 2	$E_{UCI} = E_{\text{csi}} - \min(E_{\text{csi}}, \lceil (O^{\text{CSI-part1}} + L) / R_{\text{UCI}}^{\text{max}} / Q_m \rceil \cdot Q_m)$
HARQ-ACK, CSI (CSI of two parts)	HARQ-ACK, CSI part 1	$E_{UCI} = \min(E_{\text{ack}}, \lceil (O^{\text{ACK}} + O^{\text{CSI-part1}} + L) / R_{\text{UCI}}^{\text{max}} / Q_m \rceil \cdot Q_m)$
	CSI part 2	$E_{UCI} = E_{\text{csi}} - \min(E_{\text{ack}}, \lceil (O^{\text{ACK}} + O^{\text{CSI-part1}} + L) / R_{\text{UCI}}^{\text{max}} / Q_m \rceil \cdot Q_m)$
HARQ-ACK, SR, CSI (CSI of two parts)	HARQ-ACK, SR, CSI part 1	$E_{UCI} = \min(E_{\text{ack}}, \lceil (O^{\text{ACK}} + O^{\text{SR}} + O^{\text{CSI-part1}} + L) / R_{\text{UCI}}^{\text{max}} / Q_m \rceil \cdot Q_m)$
	CSI part 2	$E_{UCI} = E_{\text{csi}} - \min(E_{\text{ack}}, \lceil (O^{\text{ACK}} + O^{\text{SR}} + O^{\text{CSI-part1}} + L) / R_{\text{UCI}}^{\text{max}} / Q_m \rceil \cdot Q_m)$

+ UCI được giải mã bằng mã hóa kênh của chiều dài khối nhỏ

- Chuỗi bit đầu vào để rate matching $d_0, d_1, d_2, \dots, d_{(N-1)}$
- Giá trị E_{UCI} được cho bởi Bảng 6.3.1.4.1-1 bằng cách đặt $L=0$
- Chuỗi bit đầu ra sau khi rate matching được biểu diễn bởi $f_0, f_1, f_2, \dots, f_{E-1}$

- Code block concatenation

+ Chuỗi bit đầu vào cho khối kết nối khối mã được biểu diễn bởi $f_{r0}, f_{r1}, f_{r2}, \dots, f_{r(E_r-1)}$ với $r = 0, \dots, C - 1$ và với E_r là số bits được rate matched với khối mã thứ r :

+ Các bit sau code block concatenation được biểu diễn bởi $g_0, g_1, g_2, \dots, g_{G-1}$ với $G' = \lceil E_{UCI} / C_{UCI} \rceil \cdot C_{UCI}$ với giá trị của E_{UCI} và C_{UCI} . G là tổng số bit được mã hóa cho truyền và $G = G' + \text{mod}(E_{UCI}, C_{UCI})$. $g_i = 0$ với $i = G', G' + 1, \dots, G - 1$

- Multiplexing of coded UCI bits to PUCCH

+ Nếu CSI của 2 parts được truyền trên PUCCH, các bit được mã hóa tương ứng với chuỗi bit UCI $a_0^{(1)}, a_1^{(1)}, \dots, a_{A^{(1)}-1}^{(1)}$ được biểu diễn bởi $g_0^{(1)}, g_1^{(1)}, \dots, g_{G^{(1)}-1}^{(1)}$ và các bit được mã hóa tương ứng với chuỗi bit UCI $a_0^{(2)}, a_1^{(2)}, \dots, a_{A^{(2)}-1}^{(2)}$ được biểu diễn bởi $g_0^{(2)}, g_1^{(2)}, \dots, g_{G^{(2)}-1}^{(2)}$. Chuỗi bit được mã hóa $g_0, g_1, g_2, \dots, g_{G-1}$ với $G = G^{(1)} + G^{(2)}$

+ Biểu diễn s_l là chỉ số ký tự UCI OFDM. Biểu diễn $N_{\text{UCI}}^{(i)}$ là số phần tử trong UCI symbol indices set $S_{\text{UCI}}^{(i)}$ với $i = 1, \dots, N_{\text{UCI}}^{\text{set}}$, với $S_{\text{UCI}}^{(i)}$ và $N_{\text{UCI}}^{\text{set}}$ được cho bởi Bảng 6.3.6-1 tuân theo PUCCH duration và cấu hình DMRS. Biểu diễn $N_{\text{symb,UCI}}^{\text{PUCCH}} = \sum_{i=1}^{N_{\text{UCI}}^{\text{set}}} N_{\text{UCI}}^{(i)}$ là số ký tự OFDM mang UCI trong PUCCH. Biểu diễn Q_m là bậc điều chế của PUCCH

- Với PUCCH format 3, đặt $N_{\text{UCI}}^{\text{symb}} = 12 \cdot N_{\text{PRB}}^{\text{PUCCH,3}} / N_{\text{SF}}^{\text{PUCCH,3}}$, với $N_{\text{PRB}}^{\text{PUCCH,3}}$ là số PRB được quyết định bởi UE cho PUCCH format 3, $N_{\text{SF}}^{\text{PUCCH,3}}$ là spreading factor cho PUCCH format 3
- Với PUCCH format 4, đặt $N_{\text{UCI}}^{\text{symb}} = 12 / N_{\text{SF}}^{\text{PUCCH,4}}$ với $N_{\text{SF}}^{\text{PUCCH,3}}$ là spreading factor cho PUCCH format 4

- PUCCH Baseband Sequence Generation

+ **Chuỗi** $r_{u,v}^{(\alpha,\delta)}$ được định nghĩa bởi một cyclic shift α của một chuỗi cơ sở $\bar{r}_{u,v}(n)$ tuân theo

$$r_{u,v}^{(\alpha,\delta)} = e^{j\omega n} \bar{r}_{u,v}(n), 0 \leq n < M_{ZC}$$

với $M_{ZC} = mN_{sc}^{RB}/2^\delta$ là chiều dài của chuỗi. Multiple sequence được định nghĩa từ 1 chuỗi cơ sở thông qua các giá trị khác nhau của α và δ được trình bày ở phần dưới

+ **Chuỗi cơ sở** $\bar{r}_{u,v}(n)$ được chia thành các nhóm với $u \in \{0,1,\dots,29\}$ là số nhóm và v là số chuỗi cơ sở trong nhóm, mỗi nhóm chứa 1 chuỗi cơ sở ($v=0$) có chiều dài $M_{ZC} = mN_{sc}^{RB}/2^\delta$, $1/2 \leq m/2^\delta \leq 5$ và 2 chuỗi cơ sở ($v=0,1$) có chiều dài $M_{ZC} = mN_{sc}^{RB}/2^\delta$, $6 \leq m/2^\delta$. Định nghĩa của chuỗi cơ sở $\bar{r}_{u,v}(0)$ và $\bar{r}_{u,v}(M_{ZC} - 1)$ phụ thuộc vào chiều dài chuỗi M_{ZC}

- Khi chiều dài chuỗi cơ sở ≥ 36

$$\bar{r}_{u,v} = x_q(n \bmod N_{ZC})$$

$$\text{với } x_q = e^{-\frac{j(\pi q m(m+1))}{N_{ZC}}}$$

$$\text{và } q = [\bar{q} + 1/2] + v(-1)^{[2\bar{q}]}, \bar{q} = N_{ZC} \cdot (u + 1)/31$$

với N_{ZC} là ước lớn nhất thỏa mãn $N_{ZC} < M_{ZC}$

- Khi chiều dài chuỗi cơ sở > 36

$$\bar{r}_{u,v} = e^{j\varphi(n)\pi/4}, 0 \leq n \leq M_{ZC} - 1 \text{ với } M_{ZC} \in \{6,12,18,24\}$$

$$\bar{r}_{u,v} = e^{-j\pi(u+1)(n+1)(n+2)/31}, 0 \leq n \leq M_{ZC} - 1 \text{ với } M_{ZC} = 30$$

+ **Group và sequence hopping** (u và v): $u = (f_{gh} + f_{ss}) \bmod 30$

- pucch-GroupHopping = ‘neither’
 - $f_{gh} = 0$
 - $f_{ss} = n_{ID} \bmod 30$ và $v=0$ với n_{ID} = hoppingID trong RRC nếu được cấu hình và $n_{ID} = N_{ID}^{cell}$ nếu hoppingID trong RRC không được cấu hình
 - $v = 0$
- pucch-GroupHopping = ‘enable’
 - $f_{gh} = \sum_{m=0}^7 2^m c(8(2n_{s,f}^\mu + n_{hop}) + m) \bmod 30$ với $c_{init} = [n_{ID}/30]$
 - $f_{ss} = n_{ID} \bmod 30$ và $v=0$ với n_{ID} = hoppingID trong RRC nếu được cấu hình và $n_{ID} = N_{ID}^{cell}$ nếu hoppingID trong RRC không được cấu hình
 - $v = 0$
- pucch-GroupHopping = ‘disable’
 - $f_{gh} = 0$
 - $f_{ss} = n_{ID} \bmod 30$ và $v=0$ với n_{ID} = hoppingID trong RRC nếu được cấu hình và $n_{ID} = N_{ID}^{cell}$ nếu hoppingID trong RRC không được cấu hình
 - $v = c(2n_{s,f}^\mu + n_{hop})$ với $c_{init} = 2^5 \left\lceil \frac{n_{ID}}{30} \right\rceil + (n_{ID} \bmod 30)$ với n_{hop} là frequency hopping Index
 - PUCCH-Resource.intraSlotFrequencyHopping = disabled: $n_{hop} = 0$
 - PUCCH-Resource.intraSlotFrequencyHopping = enabled: $n_{hop} = 0$ (first hop) và $n_{hop} = 1$ (second hop)

+ **Cyclic shift** α : $\alpha_l = 2\pi/N_{sc}^{RB} ((m_0 + m_{cs} + n_{cs}(n_{s,f}^\mu, l + l')) \bmod N_{sc}^{RB})$

với

- $n_{cs}(n_{s,f}^\mu, l) = \sum_{m=0}^7 2^m c(8(N_{symbol}^{slot} n_{s,f}^\mu + l) + m)$ với $c_{init} = n_{ID}$ và n_{ID} = hoppingID trong RRC nếu được cấu hình và $n_{ID} = N_{ID}^{cell}$ nếu hoppingID trong RRC không được cấu hình
- $n_{s,f}^\mu$: số slot trong khung vô tuyến
- l là số ký tự OFDM trong truyền PUCCH với ký tự OFDM đầu tiên của truyền PUCCH = 0
- l' là chỉ số của ký tự OFDM trong slot tương ứng với số ký tự OFDM trong truyền PUCCH
- m_{cs}
 - PUCCH format 0: HARQ-ACK value=0 thì $m_{cs} = 0$; HARQ-ACK value=1 thì $m_{cs} = 6$; HARQ-ACK value={0,0} thì $m_{cs} = 0$; HARQ-ACK value={0,1} thì $m_{cs} = 3$; HARQ-ACK value={1,1} thì $m_{cs} = 6$ và HARQ-ACK value={1,0} thì $m_{cs} = 9$
 - PUCCH format 1,2,3: $m_{cs} = 0$
- n_{cs}
 - PUCCH format 0,1: m_0 = initialCyclicShift in CRC
 - PUCCH format 3: $m_0 = 0$
 - PUCCH format 4:

- Scrambling

+ PUCCH format 2 và PUCCH format 3,4

- Khối các bit $b(0), \dots, b(M_{bit} - 1)$ với M_{bit} là số bit được truyền trên kênh vật lý, sẽ được xáo trộn trước khi điều chế, dẫn đến 1 khối các bit được xáo trộn $\tilde{b}(0), \dots, \tilde{b}(M_{bit} - 1)$

$$\tilde{b}(i) = (b(i) + c(i)) \bmod 2$$
- với chuỗi xáo trộn $c^{(q)}(i)$. Tạo chuỗi xáo trộn sẽ được hình thành bởi $c_{init} = n_{RNTI} \cdot 2^{15} + n_{ID}$ với $n_{ID} \in \{0, 1, \dots, 1023\}$ = tham số lớp cao hơn $dataScramblingIdentityPUSCH$ nếu được cấu hình nếu không thì $n_{ID} = N_{ID}^{cell}$ và n_{RNTI} được cho bởi C-RNTI

- Modulation

+ PUCCH format 0

- Chuỗi $x(n)$ được tạo bởi

$$x(l \cdot N_{sc}^{RB} + n) = r_{u,v}^{(\alpha, \delta)}(n)$$

- với
 - $n = 0, 1, \dots, N_{sc}^{RB} - 1$
 - $l = 0$ với truyền PUCCH single-symbol và $l = 0, 1$ với truyền PUCCH double-symbol
 - $r_{u,v}^{(\alpha, \delta)}$ trong phần trên và m_{cs} phụ thuộc vào thông tin được truyền

+ PUCCH format 1

- Khối các bit $b(0), \dots, b(M_{bit} - 1)$ được điều chế sử dụng BPSK nếu $M_{bit} = 1$ và QPSK nếu $M_{bit} = 2$, dẫn đến một ký tự giá trị phức $d(0)$
- Ký tự có giá trị phức $d(0)$ sẽ được nhân với chuỗi $r_{u,v}^{(\alpha, \delta)}$ thông qua
$$y(n) = d(0)r_{u,v}^{(\alpha, \delta)}(n)$$
- với

- $n = 0, 1, \dots, N_{sc}^{RB} - 1$
- $r_{u,v}^{(\alpha,\delta)}$ trong phần trên
- Khối ký tự có giá trị phức $y(0), \dots, y(N_{sc}^{RB} - 1)$ sẽ được block-wise spread với chuỗi trực giao $w_i(m)$

$$z(m' N_{sc}^{RB} N_{SF,0}^{PUCCH,1} + m N_{sc}^{RB} + n) = w_i(n) \cdot y(n)$$

- với
 - $n = 0, 1, \dots, N_{sc}^{RB} - 1$ và $m = 0, 1, \dots, N_{SF,m'}^{PUCCH,1} - 1$ và $m' = 0$ (no intra-slot frequency hopping) và $m' = 1$ (intra-slot frequency hopping)
 - $N_{SF,m'}^{PUCCH,1}$ được cho bởi bảng 6.3.2.4.1-1. Intra-slot frequency hopping sẽ được giả sử khi tham số lớp cao hơn *intraSlotFrequencyHopping* được cung cấp, mặc dù frequency-hop bằng 0 hoặc không, và ánh xạ xen kẽ không enabled, nếu không thì no intra-slot frequency hopping sẽ được mang

Table 6.3.2.4.1-1: Number of PUCCH symbols and the corresponding $N_{SF,m'}^{PUCCH,1}$.

PUCCH length, $N_{symb}^{PUCCH,1}$	$N_{SF,m'}^{PUCCH,1}$		
	No intra-slot hopping $m' = 0$	Intra-slot hopping	
		$m' = 0$	$m' = 1$
4	2	1	1
5	2	1	1
6	3	1	2
7	3	1	2
8	4	2	2
9	4	2	2
10	5	2	3
11	5	2	3
12	6	3	3
13	6	3	3
14	7	3	4

- $w_i(n)$ được cho bởi bảng 6.3.2.4.1-2 với i là chỉ số của chuỗi trực giao. Trong trường hợp truyền PUCCH spanning multiple slot, ký tự giá trị phức $d(0)$ được lặp lại cho subsequent slot.

Table 6.3.2.4.1-2: Orthogonal sequences $w_i(m) = e^{j2\pi\phi(m)/N_{SF,m'}^{PUCCH,1}}$ for PUCCH format 1.

$N_{SF,m'}^{PUCCH,1}$	$i = 0$	$i = 1$	$i = 2$	ϕ $i = 3$	$i = 4$	$i = 5$	$i = 6$
1	[0]	-	-	-	-	-	-
2	[0 0]	[0 1]	-	-	-	-	-
3	[0 0 0]	[0 1 2]	[0 2 1]	-	-	-	-
4	[0 0 0 0]	[0 2 0 2]	[0 0 2 2]	[0 2 2 0]	-	-	-
5	[0 0 0 0 0]	[0 1 2 3 4]	[0 2 4 1 3]	[0 3 1 4 2]	[0 4 3 2 1]	-	-
6	[0 0 0 0 0 0]	[0 1 2 3 4 5]	[0 2 4 0 2 4]	[0 3 0 3 0 3]	[0 4 2 0 4 2]	[0 5 4 3 2 1]	-
7	[0 0 0 0 0 0 0]	[0 1 2 3 4 5 6]	[0 2 4 6 1 3 5]	[0 3 6 2 5 1 4]	[0 4 1 5 2 6 3]	[0 5 3 1 6 4 2]	[0 6 5 4 3 2 1]

- $r_{u,v}^{(\alpha,\delta)}$ trong phần trên

+ PUCCH format 2

- Khối các bit được xáo trộn $\tilde{b}(0), \dots, \tilde{b}(M_{bit} - 1)$ được điều chế sử dụng QPSK, dẫn đến một khối ký tự giá trị phức $d(0), \dots, d(M_{symb} - 1)$ với $M_{symb} = M_{bit}/2$

$$d(i) = 1/\sqrt{2}[(1 - 2\tilde{b}(2i) + j(1 - 2\tilde{b}(2i + 1))]$$

+ PUCCH format 3,4

- Khối các bit được xáo trộn $\tilde{b}(0), \dots, \tilde{b}(M_{bit} - 1)$ được điều chế sử dụng QPSK nếu $\pi/2$ -BPSK không được cấu hình, dẫn đến một khối ký tự giá trị phức $d(0), \dots, d(M_{symb} - 1)$ với $M_{symb} = M_{bit}/2$ cho QPSK và $M_{symb} = M_{bit}$ cho $\pi/2$ -BPSK

+ $\pi/2$ -BPSK được cấu hình, điều chế QPSK:

$$d(i) = 1/\sqrt{2}[(1 - 2\tilde{b}(2i) + j(1 - 2\tilde{b}(2i + 1))]$$

+ $\pi/2$ -BPSK không được cấu hình, điều chế $\pi/2$ -BPSK:

$$d(i) = e^{j\pi/2(\text{imod}2)}/\sqrt{2}[(1 - 2\tilde{b}(2i) + j(1 - 2\tilde{b}(2i + 1))]$$

- Spreading/Block-wise spreading

+ PUCCH format 2 (Spreading)

- Spreading sẽ được áp dụng

$$z(mN_{\text{SF}}^{\text{PUCCH},2} + i) = w_n(i)d(m)$$

với $i = 0, 1, \dots, N_{\text{SF}}^{\text{PUCCH},2} - 1$ và $m = 0, 1, \dots, M_{\text{symb}} - 1$ dẫn đến một khối ký tự giá trị phức $z(0), \dots, z(N_{\text{SF}}^{\text{PUCCH},2} M_{\text{symb}} - 1)$

- Nếu tham số lớp cao hơn interlace1 không được cấu hình, tham số lớp cao hơn *OCC-Length-r16* được cấu hình
+ $N_{\text{SF}}^{\text{PUCCH},2} \in \{2, 4\}$ được cho bởi tham số lớp cao hơn *OCC-Length-r16*
+ $w_n(i)$ được cho bởi Bảng 6.3.2.5A-1 và 6.3.2.5A-2 với $n = (n_0 + n_{\text{IRB}}) \bmod N_{\text{SF}}^{\text{PUCCH},2}$, số lượng n_0 là chỉ số của chuỗi trực giao để dùng được cho bởi tham số lớp cao hơn *OCC-Length-r16* và n_{IRB} là số lwoois tài nguyên được interlace trong interlace được cho bởi tham số lớp cao hơn *Interlace0*

Table 6.3.2.5A-1: Orthogonal sequences $w_n(i)$ for PUCCH format 2 when $N_{\text{SF}}^{\text{PUCCH},2} = 2$.

n	$w_n(i)$
0	[+1 +1]
1	[+1 -1]

Table 6.3.2.5A-2: Orthogonal sequences $w_n(i)$ for PUCCH format 2 when $N_{\text{SF}}^{\text{PUCCH},2} = 4$.

n	$w_n(i)$
0	[+1 +1 +1 +1]
1	[+1 -1 +1 -1]
2	[+1 +1 -1 -1]
3	[+1 -1 -1 +1]

- Nếu không thì, $N_{\text{SF}}^{\text{PUCCH},2} = 1$ và $w_n(i) = 1$

+ PUCCH format 3,4 (Block-wise spreading)

- Cả PUCCH format 3 và 4, $M_{\text{SC}}^{\text{PUCCH},s} = M_{\text{RB}}^{\text{PUCCH},s} \cdot N_{\text{SC}}^{\text{RB}}$ với $M_{\text{RB}}^{\text{PUCCH},s}$ biểu diễn băng thông của PUCCH về mặt khối tài nguyên và sẽ cho ánh xạ non-interlaced thỏa mãn $M_{\text{RB}}^{\text{PUCCH},s} = 2^{\alpha_2} \cdot 3^{\alpha_3} \cdot 5^{\alpha_5}$ cho PUCCH format 3 và $M_{\text{RB}}^{\text{PUCCH},s} = 1$ cho PUCCH format 4 (với $\alpha_2, \alpha_3, \alpha_5$ là tập hợp các số nguyên không âm và $s \in \{3, 4\}$).
- Với ánh xạ interlaced, $M_{\text{RB}}^{\text{PUCCH},3} = 10$ nếu 1 single interlace được cấu hình và $M_{\text{RB}}^{\text{PUCCH},3} = 20$ nếu hai interlace được cấu hình
- Với PUCCH format 3, nếu ánh xạ interlaced không được cấu hình, no block-wise spreading sẽ được áp dụng và

$$y(lM_{\text{SC}}^{\text{PUCCH},3} + k) = d(lM_{\text{SC}}^{\text{PUCCH},3} + k)$$

với $k = 0, 1, \dots, M_{\text{SC}}^{\text{PUCCH},3} - 1$, $l = 0, 1, \dots, (M_{\text{symb}}/M_{\text{SC}}^{\text{PUCCH},3}) - 1$ với $M_{\text{RB}}^{\text{PUCCH},3} \geq 1$ và $N_{\text{SF}}^{\text{PUCCH},3} = 1$

- Với PUCCH format 3 với ánh xạ interlaced và PUCCH format 4, block-wise spreading sẽ được áp dụng

$$y(lM_{SC}^{PUCCH,s} + k) = w_n \left(k \frac{N_{SF}^{PUCCH,s}}{M_{SC}^{PUCCH,s}} \right) d \left(l \frac{M_{SC}^{PUCCH,s}}{N_{SF}^{PUCCH,s}} + k \bmod \frac{M_{SC}^{PUCCH,s}}{N_{SF}^{PUCCH,s}} \right)$$

với $k = 0, 1, \dots, M_{SC}^{PUCCH,s} - 1$, $l = 0, 1, \dots, (N_{SF}^{PUCCH,s} M_{symb} / M_{SC}^{PUCCH,s}) - 1$

trong đó:

+ với PUCCH format 3 với ánh xạ interlaced, $N_{SF}^{PUCCH,3} \in \{1, 2, 4\}$ nếu một single interlace được cấu hình và $N_{SF}^{PUCCH,3} = 1$, $w_n = 1$ nếu hai interlace được cấu hình

+ với PUCCH format 4, $M_{RB}^{PUCCH,4} = 1$, $N_{SF}^{PUCCH,4} \in \{2, 4\}$

với $N_{SF}^{PUCCH,s} > 1$, w_n được cho bởi Bảng 6.3.2.6.3-1 và 6.3.2.6.3-2 (với n là chỉ số của chuỗi trực giao)

Table 6.3.2.6.3-1: Orthogonal sequences $w_n(m)$ for PUCCH format 3 with interlaced mapping and PUCCH format 4 when $N_{SF}^{PUCCH,s} = 2$.

n	w_n
0	[+1 +1]
1	[+1 -1]

Table 6.3.2.6.3-2: Orthogonal sequences $w_n(m)$ for PUCCH format 3 with interlaced mapping and PUCCH format 4 when $N_{SF}^{PUCCH,s} = 4$.

n	w_n
0	[+1 +1 +1 +1]
1	[+1 -j -1 +j]
2	[+1 -1 +1 -1]
3	[+1 +j -1 -j]

- Transform precoding

+ PUCCH format 3,4

- Khối ký tự giá trị phức $y(0), \dots, y(N_{SF}^{PUCCH,s} M_{symb} - 1)$ sẽ được transform precode

$$z(l M_{SC}^{PUCCH,s} + k) = \frac{1}{\sqrt{M_{SC}^{PUCCH,s}}} \sum_{m=0}^{M_{SC}^{PUCCH,s}-1} y(l M_{SC}^{PUCCH,s} + m) e^{-j \frac{2\pi n k}{M_{SC}^{PUCCH,s}}}$$

với $k = 0, 1, \dots, M_{SC}^{PUCCH,s} - 1$, $l = 0, 1, \dots, (N_{SF}^{PUCCH,s} M_{symb} / M_{SC}^{PUCCH,s}) - 1$

thu được khối ký tự giá trị phức $z(0), \dots, z(N_{SF}^{PUCCH,s} M_{symb} - 1)$

- Mapping to physical resources

+ PUCCH format 0

- Chuỗi $x(n)$ được nhân với hệ số tỷ lệ biên độ $\beta_{PUCCH,0}$ để truyền công suất và được ánh xạ trong chuỗi bắt đầu với $x(0)$ tới các phần tử tài nguyên $(k, l)_{p,u}$ được giao cho truyền trong thứ tự tăng của chỉ số k đầu tiên qua tài nguyên vật lý được giao spanning 1 khối tài nguyên và sau đó chỉ số l trên cổng anten $p=2000$
- Với truyền interlaced, quá trình ánh xạ sẽ được lặp lại cho mỗi khối tài nguyên trong interlace và trong active bandwidth part qua khối tài nguyên vật lý được giao, với chuỗi phụ thuộc khối tài nguyên

+ PUCCH format 1

- Chuỗi $z(n)$ được nhân với hệ số tỷ lệ biên độ $\beta_{PUCCH,1}$ để tuân theo công suất truyền và được ánh xạ trong chuỗi bắt đầu với $z(n)$ tới các phần tử tài nguyên $(k, l)_{p,u}$ đáp ứng các điều kiện
 - + trong khối tài nguyên được assign cho truyền
 - + không được sử dụng bởi associated DM-RS
- Ánh xạ đến phần tử tài nguyên $(k, l)_{p,u}$ không được reserver cho các mục đích khác sẽ trong thứ tự tăng của chỉ số k đầu tiên qua tài nguyên vật lý được giao và sau đó chỉ số l trên cổng anten $p=2000$
- Với truyền interlaced, quá trình ánh xạ sẽ được lặp lại cho mỗi khối tài nguyên trong interlace và trong active bandwidth part qua khối tài nguyên vật lý được giao, với chuỗi phụ thuộc khối tài nguyên

+ PUCCH format 2

- Khối ký tự giá trị phức $z(0), \dots, z(N_{SF}^{PUCCH,2} M_{symb} - 1)$ được nhân với hệ số tỷ lệ biên độ $\beta_{PUCCH,2}$ để tuân theo công suất truyền và được ánh xạ trong chuỗi bắt đầu với $z(n)$ tới các phần tử tài nguyên $(k, l)_{p,u}$ đáp ứng các điều kiện
 - + trong khối tài nguyên được assign cho truyền
 - + không được sử dụng bởi associated DM-RS
- Ánh xạ đến phần tử tài nguyên $(k, l)_{p,u}$ không được reserver cho các mục đích khác sẽ trong thứ tự tăng của chỉ số k đầu tiên qua tài nguyên vật lý được giao và sau đó chỉ số l trên cổng anten $p=2000$

+ PUCCH format 3,4

- Khối ký tự điều chế $z(0), \dots, z(N_{SF}^{PUCCH,s} M_{symb} - 1)$ được nhân với hệ số tỷ lệ biên độ $\beta_{PUCCH,s}$ để tuân theo công suất truyền và được ánh xạ trong chuỗi bắt đầu với $z(0)$ tới các phần tử tài nguyên $(k, l)_{p,u}$ đáp ứng các điều kiện
 - + trong khối tài nguyên được assign cho truyền
 - + không được sử dụng bởi associated DM-RS
- Ánh xạ đến phần tử tài nguyên $(k, l)_{p,u}$ không được reserver cho các mục đích khác sẽ trong thứ tự tăng của chỉ số k đầu tiên qua tài nguyên vật lý được giao và sau đó chỉ số l trên cổng anten $p=2000$
- Trong trường hợp intra-slot frequency hopping, $[N_{symb}^{PUCCH,s}/2]$ ký tự OFDM sẽ được truyền trong hop đầu tiên và $N_{symb}^{PUCCH,s} - [N_{symb}^{PUCCH,s}/2]$ ký tự trong hop thứ hai với $N_{symb}^{PUCCH,s}$ là tổng số ký tự OFDM được sử dụng trong 1 slot để truyền PUCCH

3. Kênh PUSCH (Physical Uplink Shared Channel)

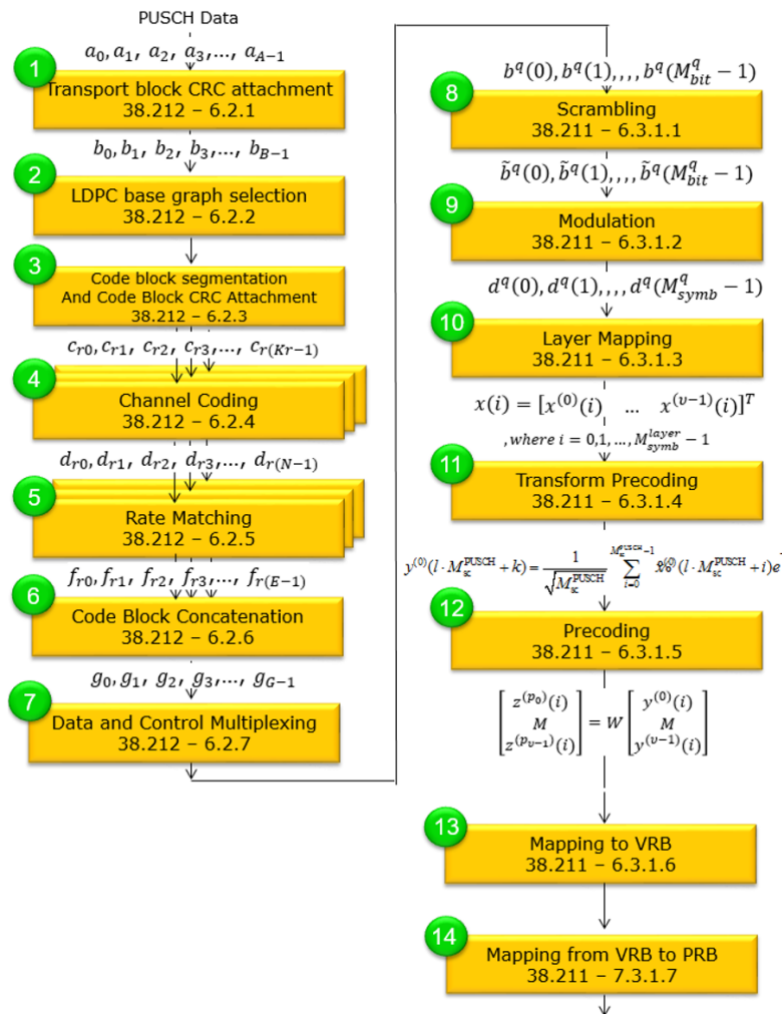
3.1. Tổng quan kênh PUSCH

Mục đích chính của PUSCH:

- + Mang dữ liệu người dùng
- + Mang UCI

- **Quá trình vận chuyển và lớp vật lý để truyền PUSCH:** (1) Transport block CRC attachment => (2) LDPC base graph selection => (3) Code block segmentation And Code Block CRC Attachment => (4) Channel Coding => (5) Rate Matching => (6) Code Block Concatenation => (7) Data and control multiplexing => (8) Scrambling => (9) Modulation => (10) Layer Mapping => (11) Transform Precoding => (12) Precoding => (13) Mapping to VRB => (14) Mapping from VRB to PRB

3.2. Cấu trúc kênh PUSCH và thuật toán của các module trong kênh PUSCH



(1) Transport block CRC attachment

- Phát hiện lỗi được cung cấp cho mỗi khối vận chuyển UL-SCH thông qua 1 khối CRC

- Khối vận chuyển đầu vào được sử dụng để tính CRC parity bit. Biểu diễn bits trong 1 khối vận chuyển truyền tới lớp 1 bởi $a_0, a_1, a_2, \dots, a_{A-1}$ và parity bits bởi $p_0, p_1, p_2, \dots, p_{L-1}$ với A là payload size và L là số parity bits. Bit thông tin thứ tự thấp nhất a_0 được ánh xạ tới bit đáng kể nhất của khối vận chuyển

- Parity bits được tính toán và thêm vào chuỗi dữ liệu, tạo thành chuỗi mở rộng $a_0, a_1, a_2, \dots, a_{A-1} \parallel p_0, p_1, p_2, \dots, p_{L-1}$; tới khối vận chuyển UL-SCH bằng cách đặt L thành 24 bits và sử dụng đa thức sinh $G_{CRC24A}(D)$.

$$g_{CRC24A}(D) = [D^{24} + D^{23} + D^{18} + D^{17} + D^{14} + D^{11} + D^{10} + D^7 + D^6 + D^5 + D^4 + D^3 + D + 1]$$

Nếu không thì L được đặt thành 16 và sử dụng đa thức sinh $G_{CRC16}(D)$.

$$g_{CRC16}(D) = [D^{16} + D^{12} + D^5 + 1]$$

Chuỗi bits sau CRC attachment được biểu diễn bởi $b_0, b_1, b_2, \dots, b_{B-1}$ với $B=A+L$

(2) LDPC base graph selection

- LDPC graph selection là bước cho phép mã hóa kênh hiệu quả phù hợp với kích thước khối truyền tải, đảm bảo truyền dữ liệu đáng tin cậy và hiệu suất được tối ưu hóa.

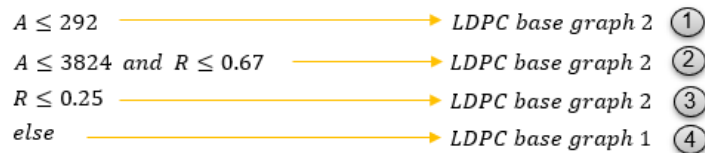
- 5G NR chỉ định hai Base Graph cho mã hóa LDPC: Base Graph 1 và Base Graph 2.

+ Mỗi Base Graph có kích thước được xác định trước, trong đó Base Graph 1 lớn hơn Base Graph 2.

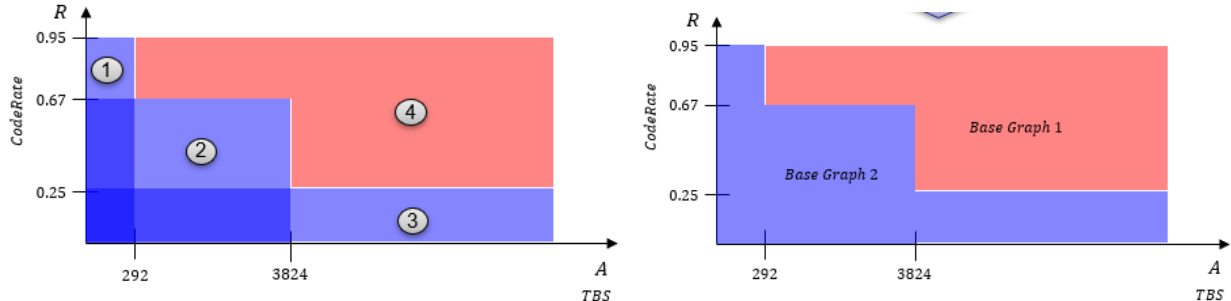
+ Việc lựa chọn đồ thị cơ sở phụ thuộc vào kích thước của khối truyền tải được truyền qua PDSCH.

+ Nếu kích thước khối truyền tải lớn hơn một ngưỡng nhất định, Base Graph 1 sẽ được sử dụng; nếu không thì Base Graph 2 được sử dụng. Base Graph 2 nhỏ hơn phù hợp hơn với các khối truyền tải nhỏ hơn vì nó mang lại sự cân bằng tốt hơn giữa độ phức tạp và hiệu suất.

- Loại LDPC BaseGraph được xác định bởi Transport Size (A) và Code Rate (R) dựa trên các tiêu chí sau.



+Biểu diễn bằng hệ tọa độ:



(3) Code block segmentation And Code Block CRC Attachment

- Các bit đầu vào cho phân đoạn khối mã được ký hiệu là $b_0, b_1, b_2, \dots, b_{B-1}$ trong đó B là số bit trong khối vận chuyển (bao gồm CRC). Nếu B lớn hơn kích thước khối mã tối đa K_{cb} sự phân chia của chuỗi bit đầu vào được thực hiện và một chuỗi CRC thêm vào với L-24 bits được gán vào mỗi khối mã

- Low density parity check coding

+ Đối với LDPC base graph 1, maximum code block size: $K_{cb} = 8488$

+ Đối với LDPC base graph 2, maximum code block size: $K_{cb} = 3840$

+ Tổng số khối mã C được xác định như sau

Nếu $B \leq K_{cb}$: $L=0$; $C=1$; $B'=B$

nếu không thì: $L=24$; $C=\lceil B/(K_{cb} - L) \rceil$; $B'=B+C.L$

+ Chuỗi bit đầu ra từ phân chia khối mã được biểu diễn bởi $c_{r0}, c_{r1}, c_{r2}, \dots, c_{r(K_r-1)}$, trong đó r là số khối mã và K_r là số bit của số khối mã r. Số bits K trong mỗi khối mã hóa được tính là

$$K' = B'/C$$

Với LDPC base graph 1, $K_b=22$

Với LDPC base graph 2

Nếu $B > 640 \Rightarrow K_b = 10$

Nếu $B > 650 \Rightarrow K_b = 9$

Nếu $B > 192 \Rightarrow K_b = 8$

Nếu không thì $K_b = 6$

tìm giá trị nhỏ nhất của Z trong tất cả tập hợp của lifting sizes trong Bảng 5.3.2-1, biểu diễn là Z_c , $K_b \geq K'$ và $K = 22 Z_c$ với LDPC base graph 1 và $K = 10 Z_c$ với LDPC base graph 2

+ Chuỗi bit c_{rk} được tính là

$s = 0$;

for $r = 0$ to $C - 1$

for $k = 0$ to $K' - L - 1$

$c_{rk} = b_s$;

$s = s + 1$;

end for

+ Nếu $C > 1$

Chuỗi $c_{r0}, c_{r1}, c_{r2}, \dots, c_{r(K_r-1)}$, được sử dụng để tính CRC parity bits $p_{r0}, p_{r1}, p_{r2}, \dots, p_{r(L-1)}$ với đa thức sinh $g_{\text{CRC24B}}(D)$

for $k = K' - L$ to $K' - 1$

$c_{rk} = p_{r(k+L-K')}$;

end for

end if

for $k = K'$ to $K - 1$ -- Insertion of filler bits

$c_{rk} = \langle \text{NULL} \rangle$;

end for

end for

- Các bit sau khi phân đoạn khối mã được ký hiệu là $c_{r0}, c_{r1}, c_{r2}, \dots, c_{r(K_r-1)}$, trong đó r là số khối mã và K_r là số bit của số khối mã r .

(4) Channel Coding của UL-SCH

- Khối mã được chuyển đến khối mã hóa kênh. Chuỗi bit trong 1 khối mã được biểu diễn bởi $c_{r0}, c_{r1}, c_{r2}, \dots, c_{r(K_r-1)}$ trong đó r là số khối mã và K_r là số bit của số khối mã r .

- Tổng số khối mã được ký hiệu là C và mỗi khối mã được mã hóa LDPC riêng lẻ.

- Sau khi mã hóa các bit được ký hiệu là $d_{r0}, d_{r1}, d_{r2}, \dots, d_{r(N_r-1)}$.

- Parity Check Matrix: Mã LDPC được xác định bằng ma trận kiểm tra chẵn lẻ thừa thể hiện mối quan hệ giữa các bit dữ liệu và các bit chẵn lẻ. 5G NR chỉ định hai base graph (Base Graph 1 and Base Graph 2) để xây dựng ma trận kiểm tra tính chẵn lẻ, tùy thuộc vào kích thước khối truyền tải.
- Encoding: Quá trình mã hóa LDPC lấy các khối mã được phân đoạn (có CRC đính kèm) làm đầu vào và tạo ra các bit chẵn lẻ dựa trên base graph đã chọn và hệ số nâng. Các bit chẵn lẻ này sau đó được gán vào các bit dữ liệu gốc, tạo thành từ mã được truyền qua PUSCH.

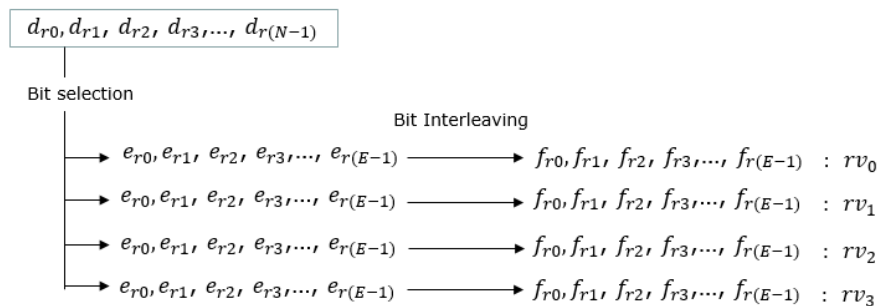
5) Rate Matching

- Các bit được mã hóa cho mỗi khối mã, ký hiệu là $d_{r0}, d_{r1}, d_{r2}, \dots, d_{r(N_r-1)}$, được phân phối đến khối Rate matching, trong đó r là số khối mã và N_r là số bit được mã hóa trong số khối mã r . Tổng số code block được ký hiệu là C và mỗi khối mã được khớp với tỷ lệ riêng với $I_{\text{LBRM}} = 1$.

- Sau khi rate matching, các bit được ký hiệu là $f_{r0}, f_{r1}, f_{r2}, \dots, f_{r(E_r-1)}$, trong đó E_r là số bit rate matched cho khối mã số r

- Mục đích của Rate matching là điều chỉnh tốc độ dữ liệu đầu ra của bộ mã hóa kênh (LDPC) để phù hợp với các tài nguyên sẵn có được phân bổ để truyền trong lưới tần số thời gian của PDSCH:

- Bit Collection:: Sau khi mã hóa LDPC, các bit được mã hóa (bit dữ liệu và bit chẵn lẻ) được thu thập trong bộ đệm tròn-circular buffer. Bộ đệm tròn là vùng lưu trữ tạm thời với kích thước cố định có thể giữ các bit theo kiểu vòng tròn, cho phép lựa chọn bit hiệu quả.
- Bit Selection:: Tùy thuộc vào tài nguyên PDSCH được phân bổ, một số bit cụ thể được chọn từ bộ đệm tròn. Quá trình lựa chọn bao gồm ba thao tác chính: bit interleaving, bit pruning, and bit puncturing.
 - Bit Interleaving: Sắp xếp lại thứ tự các bit để cải thiện độ tin cậy chống lại các lỗi bùng phát trong quá trình truyền.
 - Bit Pruning: Loại bỏ mọi bit dư thừa được tạo bởi bộ mã hóa LDPC
 - Bit Puncturing: Loại bỏ một số bit được mã hóa (thường là bit chẵn lẻ) nếu số lượng bit được mã hóa vượt quá tài nguyên được phân bổ.



(6) Code Block Concatenation

- Đây là bước kết hợp nhiều khối mã thu được từ các bước xử lý trước đó thành một luồng dữ liệu duy nhất để truyền.

- Sau Rate Matching, các khối mã đã xử lý được kết hợp thành một luồng dữ liệu duy nhất. Việc ghép nối được thực hiện theo một thứ tự cụ thể để đảm bảo rằng bộ thu (UE) có thể tách và giải mã chính xác các khối mã riêng lẻ. Thông thường, các khối mã được nối theo thứ tự chúng được phân đoạn từ khối truyền tải ban đầu..

- Chuỗi bit đầu vào cho khối nối khối mã là các chuỗi $f_{r0}, f_{r1}, f_{r2}, \dots, f_{r(E_r-1)}$ với $r = 0, \dots, C-1$ và E_r là số bit khớp tốc độ cho khối mã thứ r .

- Các bit sau khi ghép khối mã được ký hiệu là $g_0, g_1, g_2, \dots, g_{G-1}$ trong đó G là tổng số bit được mã hóa để truyền tải.

- Việc ghép khối mã bao gồm việc ghép tuần tự các đầu ra khớp tốc độ cho các khối mã khác nhau.

Set $k = 0$ and $r = 0$

while $r < C$

Set $j = 0$

while $j < E_r$

$g_k = f_{rj}$

$k = k + 1$

$j = j + 1$

end while

$r = r + 1$

end while

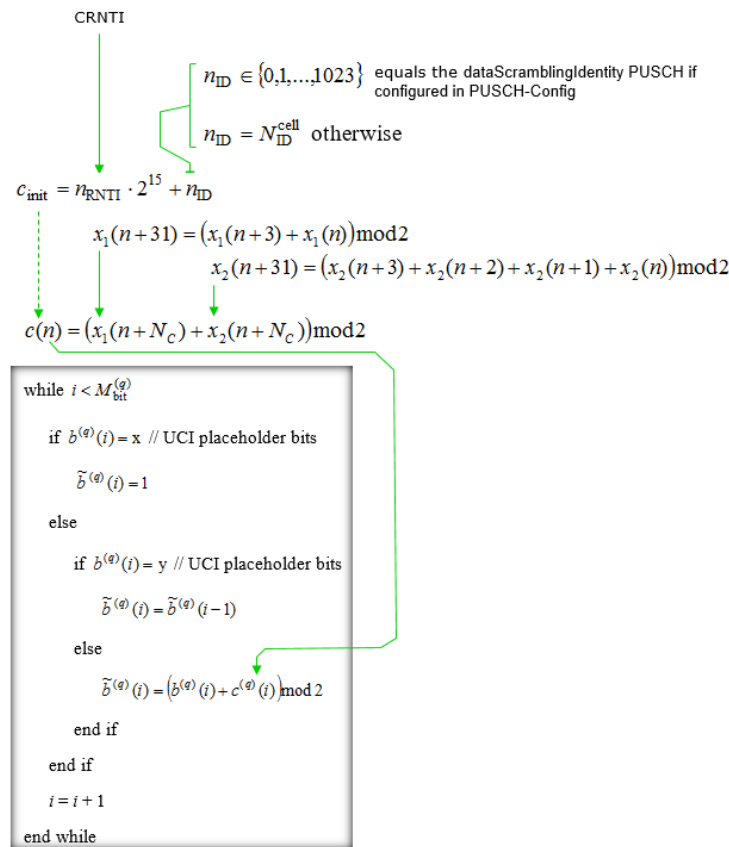
(7) Data and control multiplexing

- Không giống PDSCH, PUSCH có thể mang UCI (control data) nếu được cấu hình. Bước này là để ghép dữ liệu điều khiển và dữ liệu người dùng.

- + Các bit cho kênh chia sẻ đường lên (UL-SCH) và acknowledgments for Hybrid Automatic Repeat Request (HARQ-ACK) được biểu thị và sắp xếp theo trình tự tùy theo sự có mặt của CG-UCI khi ghép kênh cg-UCI được định cấu hình.
- + Thiết lập trình tự cho Channel State Information (CSI) part 1 và 2, nếu có.
- + Xác định chuỗi cho Thông tin kênh điều khiển đường lên (UCI) không có HARQ-ACK.
- + Ghép kênh dữ liệu và điều khiển các bit được mã hóa thành một chuỗi đơn, đảm bảo rằng thứ tự phản ánh mức độ ưu tiên truyền.
- + Tính chỉ số ký tự OFDM cho PUSCH được lập lịch, tính tất cả các ký tự OFDM được sử dụng cho DMRS

(8) Scrambling

- Quá trình xáo trộn là bước đưa tính ngẫu nhiên vào dữ liệu được truyền, đảm bảo phân phối điện năng đồng đều, quản lý nhiễu, bảo mật dữ liệu và ước tính kênh chính xác. Hoạt động này đảm bảo rằng tín hiệu được truyền có sự phân bố công suất đồng đều trên các tài nguyên thời gian và tần số khác nhau.
- + Một scrambling identity n_{ID} được chọn, có thể nằm trong một phạm vi nếu được cấu hình trong PUSCH-Config, nếu không thì dựa trên ID ô vật lý.
- + Chuỗi xáo trộn ban đầu c_{init} được tính toán bằng cách sử dụng RNTI, lũy thừa 2 và nhận dạng xáo trộn.
- + Hai chuỗi giả ngẫu nhiên x_1 và x_2 được sử dụng để tạo ra chuỗi xáo trộn $c(n)$.
- + Chuỗi nhị phân $b^{(q)}(i)$ được xáo trộn bằng cách sử dụng chuỗi $c(i)$, với cách xử lý đặc biệt đối với các bit giữ chỗ UCI.
- + Các bit giữ chỗ UCI được đặt thành 1 và quá trình xáo trộn được tiếp tục cho từng bit của chuỗi nhị phân cho đến hết khối.



- Với mỗi từ mã q , UE sẽ giả sử khối bit $b^{(q)}(0), \dots, b^{(q)}(M_{bit}^{(q)} - 1)$, trong đó $M_{bit}^{(q)}$ là số lượng các bit trong từ mã q được truyền trên kênh vật lý, được xáo trộn trước khi điều chế, dẫn đến một khối các bit được xáo trộn $\tilde{b}^{(q)}(0), \dots, \tilde{b}^{(q)}(M_{bit}^{(q)} - 1)$ theo mã giả dưới

```

Set  $i = 0$ 
while  $i < M_{\text{bit}}^{(q)}$ 
    if  $b^{(q)}(i) = x$  // UCI placeholder bits
         $\tilde{b}^{(q)}(i) = 1$ 
    else
        if  $b^{(q)}(i) = y$  // UCI placeholder bits
             $\tilde{b}^{(q)}(i) = \tilde{b}^{(q)}(i-1)$ 
        else
             $\tilde{b}^{(q)}(i) = (b^{(q)}(i) + c^{(q)}(i)) \bmod 2$ 
        end if
    end if
     $i = i + 1$ 
end while

```

với x và y được tag và chuỗi xáo trộn được cho bởi $c(n) = (x_1(n + N_c) \text{ và } x_2(n + N_c)) \bmod 2$

- Chuỗi xáo trộn được tạo bởi

$$c_{\text{init}} = \begin{cases} n_{\text{RINT}} \cdot 2^{16} + n_{\text{RAPID}} \cdot 2^{10} + n_{\text{ID}}, & \text{for msgA on PUSCH} \\ n_{\text{RINT}} \cdot 2^{15} + n_{\text{ID}}, & \text{khác} \end{cases}$$

với $n_{\text{ID}} \in \{1, 0, \dots, 1023\}$ bằng tham số lớp cao hơn *dataScramblingIdentityPUSCH* nếu được cấu hình và RNTI bằng C-RNTI, MCS-C-RNTI, SP-CSI-RNTI hoặc CS-RNTI, và truyền dẫn không được lập lịch sử dụng DCI format 0_0 trong không gian tìm kiếm chung;

+ $n_{\text{ID}} \in \{1, 0, \dots, 1023\}$ bằng tham số lớp cao hơn *msgA-DataScramblingIndex* nếu được cấu hình và truyền PUSCH được kích hoạt bởi Type-2 random access

+ $n_{\text{ID}} = N_{\text{ID}}^{\text{cell}}$ otherwise

+ n_{RAPID} là chỉ số của random-access preamble được truyền cho msgA

(9) Modulation

- Đối với mỗi từ mã $q=0$, khối bit được xáo trộn $\tilde{b}^{(q)}(0), \dots, \tilde{b}^{(q)}(M_{\text{bit}}^{(q)} - 1)$ được điều chế bằng cách sử dụng một trong các sơ đồ điều chế trong Bảng 6.3.1.2-1, dẫn đến một khối ký tự điều chế có giá trị phức $d^{(q)}(0), \dots, d^{(q)}(M_{\text{bit}}^{(q)} - 1)$.

Table 6.3.1.2-1: Supported modulation schemes.

Transform precoding disabled		Transform precoding enabled	
Modulation scheme	Modulation order Q_m	Modulation scheme	Modulation order Q_m
		$\pi/2$ -BPSK	1
QPSK	2	QPSK	2
16QAM	4	16QAM	4
64QAM	6	64QAM	6
256QAM	8	256QAM	8

(10) Layer Mapping

- Đối với mỗi từ mã $q=0$, ký tự điều chế có giá trị phức cho từ mã được truyền sẽ được ánh xạ lên 4 lớp theo Bảng 7.3.1.3-2. Các ký tự điều chế có giá trị phức $d^{(q)}(0), \dots, d^{(q)}(M_{\text{bit}}^{(q)} - 1)$ đối với từ mã q sẽ được ánh xạ lên các lớp $x(i) = [x^{(0)}(i), \dots, x^{(v-1)}(i)]^T$ với $i=0, 1, \dots, M_{\text{symbol}}^{\text{layer}} - 1$; trong đó v là số lớp và $M_{\text{symbol}}^{\text{layer}}$ là số ký tự điều chế trên mỗi lớp

Codeword index

$$d^{(q)}(0), \dots, d^{(q)}(M_{\text{symp}}^{(q)} - 1)$$

Layer index

38.211-Table 7.3.1.3-1

$$x(i) = [x^{(0)}(i) \ \dots \ x^{(v-1)}(i)]^T \quad i = 0, 1, \dots, M_{\text{symp}}^{\text{layer}} - 1$$

$$= \begin{bmatrix} x^{(0)}(0) & x^{(0)}(1) & \dots & x^{(0)}(M_{\text{symp}}^{\text{layer}} - 1) \\ x^{(1)}(0) & x^{(1)}(1) & \dots & x^{(1)}(M_{\text{symp}}^{\text{layer}} - 1) \\ \vdots & \vdots & \ddots & \vdots \\ x^{(v-1)}(0) & x^{(v-1)}(1) & \dots & x^{(v-1)}(M_{\text{symp}}^{\text{layer}} - 1) \end{bmatrix}$$

< Table 7.3.1.3-1: Codeword-to-layer mapping for spatial multiplexing. >

Number of layers	Number of codewords	Codeword-to-layer mapping $i = 0, 1, \dots, M_{\text{symp}}^{\text{layer}} - 1$
1	1	$x^{(0)}(i) = d^{(0)}(i) \quad M_{\text{symp}}^{\text{layer}} = M_{\text{symp}}^{(0)}$
2	1	$x^{(0)}(i) = d^{(0)}(2i)$ $x^{(1)}(i) = d^{(0)}(2i+1) \quad M_{\text{symp}}^{\text{layer}} = M_{\text{symp}}^{(0)} / 2$
3	1	$x^{(0)}(i) = d^{(0)}(3i)$ $x^{(1)}(i) = d^{(0)}(3i+1)$ $x^{(2)}(i) = d^{(0)}(3i+2) \quad M_{\text{symp}}^{\text{layer}} = M_{\text{symp}}^{(0)} / 3$
4	1	$x^{(0)}(i) = d^{(0)}(4i)$ $x^{(1)}(i) = d^{(0)}(4i+1)$ $x^{(2)}(i) = d^{(0)}(4i+2)$ $x^{(3)}(i) = d^{(0)}(4i+3) \quad M_{\text{symp}}^{\text{layer}} = M_{\text{symp}}^{(0)} / 4$
5	2	$x^{(0)}(i) = d^{(0)}(2i)$ $x^{(1)}(i) = d^{(0)}(2i+1)$ $x^{(2)}(i) = d^{(1)}(3i)$ $x^{(3)}(i) = d^{(1)}(3i+1)$ $x^{(4)}(i) = d^{(1)}(3i+2) \quad M_{\text{symp}}^{\text{layer}} = M_{\text{symp}}^{(0)} / 2 = M_{\text{symp}}^{(1)} / 3$
6	2	$x^{(0)}(i) = d^{(0)}(3i)$ $x^{(1)}(i) = d^{(0)}(3i+1)$ $x^{(2)}(i) = d^{(0)}(3i+2)$ $x^{(3)}(i) = d^{(1)}(3i)$ $x^{(4)}(i) = d^{(1)}(3i+1)$ $x^{(5)}(i) = d^{(1)}(3i+2) \quad M_{\text{symp}}^{\text{layer}} = M_{\text{symp}}^{(0)} / 3 = M_{\text{symp}}^{(1)} / 3$
7	2	$x^{(0)}(i) = d^{(0)}(3i)$ $x^{(1)}(i) = d^{(0)}(3i+1)$ $x^{(2)}(i) = d^{(0)}(3i+2)$ $x^{(3)}(i) = d^{(1)}(4i)$ $x^{(4)}(i) = d^{(1)}(4i+1)$ $x^{(5)}(i) = d^{(1)}(4i+2)$ $x^{(6)}(i) = d^{(1)}(4i+3) \quad M_{\text{symp}}^{\text{layer}} = M_{\text{symp}}^{(0)} / 3 = M_{\text{symp}}^{(1)} / 4$
8	2	$x^{(0)}(i) = d^{(0)}(4i)$ $x^{(1)}(i) = d^{(0)}(4i+1)$ $x^{(2)}(i) = d^{(0)}(4i+2)$ $x^{(3)}(i) = d^{(0)}(4i+3)$ $x^{(4)}(i) = d^{(1)}(4i)$ $x^{(5)}(i) = d^{(1)}(4i+1)$ $x^{(6)}(i) = d^{(1)}(4i+2)$ $x^{(7)}(i) = d^{(1)}(4i+3) \quad M_{\text{symp}}^{\text{layer}} = M_{\text{symp}}^{(0)} / 4 = M_{\text{symp}}^{(1)} / 4$

(11) Transform Precoding

- Thủ tục UE áp dụng cho transform precoding on PUSCH

+ Với một PUSCH được lập lịch bởi RAR UL grant, hoặc fallbackRAR UL grant, hoặc được lập lịch bởi DCI format 0_0 với CRC được xáo trộn bởi TC_RNTI, transform precoding cả “enabled” và “disabled” tuân theo tham số được cấu hình lớp cao hơn *msg3-transformPrecoder*

+ Với một MsgA PUSCH, transform precoding cả “enabled” và “disabled” tuân theo tham số được cấu hình lớp cao hơn *msgA-TransformPrecoder*. Nếu *msgA-TransformPrecoder* không được cấu hình,

transform precoding cả “enabled” và “disabled” tuân theo tham số được cấu hình lớp cao hơn *msg3-transformPrecoder*

+ Với PUSCH được lập lịch bởi 1 PDCCH với CRC được xáo trộn bằng CS_RNTI với NDI=1, C_RNTI hoặc MCS-C-RNTI hoặc SP-CSI_RNTI

- Nếu DCI với lập lịch grant được nhận với DCI format 0_0, với PUSCH, transform precoding cả “enabled” và “disabled” tuân theo tham số được cấu hình lớp cao hơn *msg3-transformPrecoder*
- Nếu DCI với lập lịch grant được nhận với DCI format 0_0,
 - Nếu UE được cấu hình với tham số lớp cao hơn *transformPrecoder* trong *pusch-Config*, với truyền PUSCH, quan tâm đến tham số này
 - Nếu UE không được cấu hình với tham số lớp cao hơn *transformPrecoder* trong *pusch-Config*, với truyền PUSCH, transform precoding cả “enabled” và “disabled” tuân theo tham số được cấu hình lớp cao hơn *msg3-transformPrecoder*

+ Với truyền PUSCH với grant được cấu hình

- Nếu UE được cấu hình với thành phần lớp cao hơn *transformPrecoder* trong *pusch-Config*, với truyền PUSCH, quan tâm đến tham số này
- Nếu UE không được cấu hình với tham số lớp cao hơn *transformPrecoder* trong *pusch-Config*, với truyền PUSCH, transform precoding cả “enabled” và “disabled” tuân theo tham số được cấu hình lớp cao hơn *msg3-transformPrecoder*

- Nếu transform precoding không enabled theo thủ tục trên, $y^{(\lambda)}(i) = x^{(\lambda)}(i)$ cho mỗi lớp $\lambda = 1, 0, \dots, v-1$

- Nếu transform precoding enabled theo thủ tục trên, $v = 1$ và $\bar{x}^{(0)}(i)$ phụ thuộc vào cấu hình của tín hiệu tham chiếu phase-tracking

- Nếu thủ tục trên biểu diễn rằng tín hiệu tham chiếu phase-tracking không đang được sử dụng, khối ký tự giá trị phức $x^{(0)}(0), \dots, x^{(0)}(M_{\text{symbol}}^{\text{layer}} - 1)$ với lớp đơn $\lambda = 0$ sẽ được chia thành $M_{\text{symbol}}^{\text{layer}}/M_{\text{sc}}^{\text{PUSCH}}$ set, mỗi set tương ứng với 1 ký tự OFDM và $\bar{x}^{(0)}(i) = x^{(0)}(i)$

- Nếu thủ tục trên biểu diễn rằng tín hiệu tham chiếu phase-tracking đang được sử dụng, khối ký tự giá trị phức $x^{(0)}(0), \dots, x^{(0)}(M_{\text{symbol}}^{\text{layer}} - 1)$ set sẽ được chia thành các set, mỗi set tương ứng với 1 ký tự OFDM và với set l gồm $M_{\text{sc}}^{\text{PUSCH}} - \epsilon_l N_{\text{samp}}^{\text{group}} N_{\text{group}}^{\text{PTRS}}$ ký tự và được ánh xạ tới các ký tự giá trị phức $\tilde{x}^{(0)}(l M_{\text{sc}}^{\text{PUSCH}} + i')$ ứng với ký tự OFDM l trước khi transform precoding, với $i' \in \{0, 1, \dots, M_{\text{sc}}^{\text{PUSCH}} - 1\}$ và $i' \neq m$. Chỉ số m của mẫu PT-RS trong set l , số mẫu trên mỗi PT-RS nhóm $N_{\text{samp}}^{\text{group}}$ và số nhóm PT-RS $N_{\text{group}}^{\text{PTRS}}$ được cho bởi bảng 6.4.1.2.2.2-1. Khi ký tự OFDM l chứa 1 hoặc nhiều hơn mẫu PT-RS thì $\epsilon_l = 1$, nếu không thì $\epsilon_l = 0$

- Transform precoding sẽ được áp dụng tuân theo

$$y^{(0)}(l M_{\text{sc}}^{\text{PUSCH}} + k) = \frac{1}{\sqrt{M_{\text{sc}}^{\text{PUSCH}}}} \bar{x}^{(0)}(l M_{\text{sc}}^{\text{PUSCH}} + i) e^{-j \frac{2\pi k}{M_{\text{sc}}^{\text{PUSCH}}}}$$

$$k = 0, \dots, M_{\text{sc}}^{\text{PUSCH}} - 1, l = 0, \dots, M_{\text{symbol}}^{\text{layer}}/M_{\text{sc}}^{\text{PUSCH}} - 1$$

dẫn đến một khối ký tự giá trị phức $y^{(0)}(0), \dots, y^{(0)}(M_{\text{symbol}}^{\text{layer}} - 1)$. Biến $M_{\text{sc}}^{\text{PUSCH}} = M_{\text{RB}}^{\text{PUSCH}} \cdot N_{\text{sc}}^{\text{RB}}$ với $M_{\text{RB}}^{\text{PUSCH}}$ biểu diễn bằng thông của PUSCH trong RBs thỏa mãn $M_{\text{RB}}^{\text{PUSCH}} = 2^{\alpha_2} \cdot 3^{\alpha_3} \cdot 5^{\alpha_5}$ với $\alpha_2, \alpha_3, \alpha_5$ là các số nguyên không âm

Table 6.4.1.2.2-1: PT-RS symbol mapping.

Number of PT-RS groups $N_{\text{group}}^{\text{PT-RS}}$	Number of samples per PT-RS group $N_{\text{samp}}^{\text{group}}$	Index m of PT-RS samples in OFDM symbol / prior to transform precoding
2	2	$s \lfloor M_{\text{sc}}^{\text{PUSCH}} / 4 \rfloor + k - 1$ where $s = 1, 3$ and $k = 0, 1$
2	4	$s M_{\text{sc}}^{\text{PUSCH}} + k$ where $\begin{cases} s = 0 & \text{and } k = 0, 1, 2, 3 \\ s = 1 & \text{and } k = -4, -3, -2, -1 \end{cases}$
4	2	$\lfloor s M_{\text{sc}}^{\text{PUSCH}} / 8 \rfloor + k - 1$ where $s = 1, 3, 5, 7$ and $k = 0, 1$
4	4	$s M_{\text{sc}}^{\text{PUSCH}} / 4 + n + k$ where $\begin{cases} s = 0 & \text{and } k = 0, 1, 2, 3 & n = 0 \\ s = 1, 2 & \text{and } k = -2, -1, 0, 1 & n = \lfloor M_{\text{sc}}^{\text{PUSCH}} / 8 \rfloor \\ s = 4 & \text{and } k = -4, -3, -2, -1 & n = 0 \end{cases}$
8	4	$\lfloor s M_{\text{sc}}^{\text{PUSCH}} / 8 \rfloor + n + k$ where $\begin{cases} s = 0 & \text{and } k = 0, 1, 2, 3 & n = 0 \\ s = 1, 2, 3, 4, 5, 6 & \text{and } k = -2, -1, 0, 1 & n = \lfloor M_{\text{sc}}^{\text{PUSCH}} / 16 \rfloor \\ s = 8 & \text{and } k = -4, -3, -2, -1 & n = 0 \end{cases}$

(12) Precoding

- Khối vecto $[y^{(0)}(i), \dots, y^{(v-1)}(i)]^T$ với $i=0, 1, \dots, M_{\text{symbol}}^{\text{layer}} - 1$ sẽ được precode tuân theo

$$\begin{bmatrix} z^{(p_0)}(i) \\ \vdots \\ z^{(p_{\rho-1})}(i) \end{bmatrix} = W \begin{bmatrix} y^{(0)}(i) \\ \vdots \\ y^{(v-1)}(i) \end{bmatrix}$$

với $i=0, 1, \dots, M_{\text{symbol}}^{\text{layer}} - 1$. Tập hợp các cổng Anten $\{p_0, \dots, p_{\rho-1}\}$ sẽ được quyết định

+ Với non-codebook-based transmission, ma trận tiền mã hóa W bằng ma trận identity

+ Với codebook-based transmission, ma tr

ận tiền mã hóa $W = 1$ với truyền 1 lớp trên một cổng Anten đơn, nếu không thì theo Bảng 6.3.1.5-1 tới 6.3.1.5-7 với chỉ số TPM1 thu được từ lặp lịch DCI truyền dẫn đường lên hoặc tham số lớp cao hơn tuân theo thủ tục

+ Khi tham số lớp cao hơn $txConfig$ không được cấu hình, ma trận tiền mã hóa $W = 1$

(13) Mapping to VRB

- Với mỗi cổng Anten được sử dụng cho truyền dẫn của PUSCH, khối ký tự giá trị phức $z^{(p)}(0), \dots, z^{(p)}(M_{\text{symb}}^{\text{ap}} - 1)$ sẽ được nhân với hệ số tỷ lệ biên độ β_{PDSCH} để phù hợp với phân bổ công suất đường xuống và được ánh xạ theo trình tự bắt đầu từ $z^{(p)}(0)$ tới các phần tử tài nguyên $(k', l)_{p,u}$ trong các khối tài nguyên ảo được gán cho truyền đáp ứng tất cả các tiêu chí sau:

+ Nằm trong các khối tài nguyên ảo được chỉ định để truyền tải;

+ Các phần tử tài nguyên tương ứng trong các khối tài nguyên vật lý tương ứng sẽ không được sử dụng cho truyền dẫn của DM-RS, PT-RS hoặc DM-RS liên quan dành cho các UE đồng lập lịch khác

- Ánh xạ tới các phần tử tài nguyên $(k', l)_{p,u}$ được phân bổ cho PUSCH theo [6, TS 38.214] sẽ theo thứ tự tăng dần đầu tiên là chỉ số k' trên các khối tài nguyên ảo được chỉ định, trong đó $k' = 0$ là sóng mang con đầu tiên trong khối tài nguyên ảo được đánh số thấp nhất được chỉ định để truyền và sau đó là index l

(14) Mapping from VRB to PRB

- Các khối tài nguyên ảo sẽ được ánh xạ tới các khối tài nguyên vật lý theo ánh xạ không xen kẽ - non-interleaved mapping.

+ Đối với ánh xạ VRB-to-PRB không xen kẽ cho phân bổ tài nguyên đường lên loại 0 và 1, tài nguyên ảo khối n được ánh xạ tới khối tài nguyên vật lý n ngoại trừ PUSCH được lập lịch bởi RAR UL Grant hoặc PUSCH được lập lịch theo định dạng DCI 0_0 với CRC được xáo trộn bởi TC-RNTI trong active uplink bandwidth part i bắt đầu từ N_{BWP}^{start} , , bao gồm tất cả các khối tài nguyên của intial uplink bandwidth part bắt đầu từ N_{BWP}^{start} , và có cùng khoảng cách sóng mang con và CP làm intial uplink bandwidth part, trong trường hợp đó khối tài nguyên ảo n được ánh xạ tới khối tài nguyên vật lý $n + N_{BWP,0}^{start} + N_{BWP,i}^{start}$.

+ Đối với ánh xạ VRB-to-PRB không xen kẽ để phân bổ tài nguyên đường lên loại 2, khối tài nguyên ảo n được ánh xạ tới khối tài nguyên vật lý n .

(*) OFDM baseband signal generation for PDCCH, PDSCH

- Tín hiệu thời gian liên tục $s_l^{(p,\mu)}(t)$ tại antenna port p và cấu hình khoảng cách sóng mang con μ trong ký tự OFDM $l \in \{0,1,...,N_{slot}^{subframe,\mu} N_{symb}^{slot} - 1\}$ trong một subframe được xác định bởi công thức

$$s_l^{(p,\mu)}(t) = \begin{cases} \bar{s}_l^{(p,\mu)}(t), & t_{start,l}^\mu \leq t < t_{start,l}^\mu + t_{symb,l}^\mu \\ 0, & \text{khác} \end{cases}$$

$$\bar{s}_l^{(p,\mu)}(t) = \sum_{k=0}^{N_{grid,x}^{size,\mu} N_{SC}^{RB} - 1} \alpha_{k,l}^{(p,\mu)}(t) e^{j2\pi(k+k_0^\mu - N_{grid,x}^{size,\mu} N_{SC}^{RB}/2)\Delta f(t - N_{CP,l}^\mu T_C - t_{start,l}^\mu)}$$

$$k_0^\mu = (N_{grid,x}^{start,\mu} + N_{grid,x}^{size,\mu}/2)N_{SC}^{RB} - (N_{grid,x}^{start,\mu_0} + N_{grid,x}^{size,\mu_0}/2)N_{SC}^{RB} 2^{\mu_0 - \mu}$$

$$T_{symb,l}^\mu = (N_u^\mu + N_{CP,l}^\mu)T_C$$

với $t=0$ tại vị trí bắt đầu của subframe

$$N_u^\mu = 2048k \cdot 2^{-\mu}$$

$$N_{CP,l}^\mu = \begin{cases} 512k \cdot 2^{-\mu} & \text{extended CP} \\ 144k \cdot 2^{-\mu} + 16 & \text{normal CP, } l = 0 \text{ or } l = 7 \cdot 2^\mu \\ 144k \cdot 2^{-\mu} & \text{normal CP, } l \neq 0 \text{ or } l \neq 7 \cdot 2^\mu \end{cases}$$

và

+ Δf cho bởi bảng 4.2.1

Table 4.2-1: Supported transmission numerologies.

μ	$\Delta f = 2^\mu \cdot 15 [\text{kHz}]$	Cyclic prefix
0	15	Normal
1	30	Normal
2	60	Normal, Extended
3	120	Normal
4	240	Normal

+ μ là cấu hình khoảng cách sóng mang con

+ μ_0 là giá trị lớn nhất của μ

- Trong trường hợp mở rộng CP của ký tự OFDM đầu tiên được phân bổ cho truyền PUSCH, SRS hoặc PUCCH, tín hiệu thời gian liên tục $s_l^{(p,\mu)}$ phần mở rộng $t_{start,l}^\mu - T_{ext} \leq t < t_{start,l}^\mu$ preceding ký tự OFDM đầu tiên cho PUSCH, SRS hoặc PUCCH được đưa ra bởi

$$s_{ext}^{(p,\mu)}(t) = \bar{s}_l^{(p,\mu)}(t)$$

với $t < 0$ liên quan đến tín hiệu trong khung con trước đó và

+ đối với truyền PUSCH, SRS và PUCCH theo lập lịch động

$$T_{ext} = \min(\max(T'_{ext}, 0), T_{symb,(l-1) \bmod 7 - 2^\mu}^\mu)$$

$$T'_{\text{ext}} = \sum_{k=1}^{C_i} T_{\text{symb},(l-k) \bmod 7 - 2^\mu}^\mu - \Delta_i$$

với Δ_i được cho bởi bảng 5.3.1.1-1 với $C_1=1$ với $\mu \in \{0,1\}$, $C_1=2$ với $\mu = 2$, và C_2 và C_3 được cho bởi các tham số lớp cao hơn lần lượt là *cp-ExtensionC2-r16* và *cp-ExtensionC3-r16* và $T_{\text{TA}} = (N_{\text{TA}} + N_{\text{TA,offset}})/T_{\text{C}}$. Đối với truy cập ngẫu nhiên dựa trên tranh chấp hoặc khi không có cấu hình lớp cao hơn của C_2 và C_3 , giá trị của C_i sẽ được đặt thành số nguyên lớn nhất đáp ứng $T'_{\text{ext}} < T_{\text{symb},(l-1) \bmod 7 - 2^\mu}^\mu$ cho mỗi giá trị của $i \in \{2,3\}$

Table 5.3.1-1: The variables C_i and Δ_i for cyclic prefix extension

T_{ext} index i	C_i	Δ_i
0	-	-
1	C_1	$25 \cdot 10^{-6}$
2	C_2	$16 \cdot 10^{-6} + T_{\text{TA}}$
3	C_3	$25 \cdot 10^{-6} + T_{\text{TA}}$

+ đối với truyền PUSCH sử dụng configured grant

$$T_{\text{ext}} = m \sum_{k=1}^{2^\mu} T_{\text{symb},(l-k) \bmod 7 - 2^\mu}^\mu - \Delta_i$$

trong đó Δ_i được cho trong Bảng 5.3.1-2

Table 5.3.1-2: The variable Δ_i for cyclic prefix extension with configured grants.

index i	Δ_i
0	$16 \cdot 10^{-6}$
1	$25 \cdot 10^{-6}$
2	$34 \cdot 10^{-6}$
3	$43 \cdot 10^{-6}$
4	$52 \cdot 10^{-6}$
5	$61 \cdot 10^{-6}$
6	$\sum_{k=1}^{2^\mu} T_{\text{symb},(l-k) \bmod 7 - 2^\mu}^\mu$

- Vị trí bắt đầu của ký hiệu OFDM l cho cấu hình khoảng cách sóng mang con μ trong khung con được cho bởi

$$t_{\text{start},l}^\mu(t) = \begin{cases} 0, & l = 0 \\ t_{\text{start},l-1}^\mu + (N_{\text{u}}^\mu + N_{\text{CP},l-1}^\mu)T_{\text{C}}, & \text{khác} \end{cases}$$

4. Tín hiệu SRS – Sounding reference signal

4.1. SRS resource

- Một SRS resource được cấu hình bởi *SRS-Resource IE* hoặc *SRS-PosResource IE* và bao gồm

+ $N_{\text{ap}}^{\text{SRS}} \in \{1,2,4\}$ cổng anten $(p_i)_{i=0}^{N_{\text{ap}}^{\text{SRS}}-1}$, với số cổng anten được cho bởi tham số lớp cao hơn *nrofSRS-Ports* nếu được cấu hình, nếu không thì $N_{\text{ap}}^{\text{SRS}} = 1$, và $p_i = 1000 + i$ khi SRS resource trong tập SRS resource với tham số lớp cao hơn *usage* trong *SRS-Resource* không set tới ‘nonCodebook’, hoặc được quyết định khi SRS resource trong một tập SRS resource với tham số lớp cao hơn *usage* trong *SRS-Resource* set tới ‘nonCodebook’

+ $N_{\text{symbol}}^{\text{SRS}} \in \{1,2,4,8,12\}$ các ký tự OFDM liên tiếp được cho bởi miền *nrofSymbols* được bao gồm trong tham số lớp cao hơn *resourceMapping*

+ l_0 là vị trí bắt đầu trong miền thời gian được cho bởi $l_0 = N_{\text{symbol}}^{\text{slot}} - 1 - l_{\text{offset}}$ với $l_{\text{offset}} \in \{0,1,\dots,13\}$ đếm symbol backwards từ cuối của slot và được cho bởi miền *startPosition* được bao gồm trong tham số lớp cao hơn *resourceMapping* và $l_{\text{offset}} \geq N_{\text{symbol}}^{\text{SRS}} - 1$

+ k_0 là vị trí bắt đầu miền tần số của SRS

4.2. Sequence generation

- SRS sequence cho một SRS resource được tạo tuân theo

$$r^{(p_i)}(n, l') = r_{u,v}^{(a_i, \delta)}(n)$$

$$0 \leq n \leq M_{sc,b}^{\text{SRS}} - 1$$

$$l' \in \{0,1, \dots, N_{\text{symb}}^{\text{SRS}} - 1\}$$

với $M_{sc,b}^{\text{SRS}}$ được cho bởi phần 6.4.1.4.3, $r_{u,v}^{(a_i, \delta)}(n)$ được cho bởi phần 5.2.2 với $\delta = \log_2(K_{TC})$ và transmission comb number $K_{TC} \in \{2,4,8\}$ bao gồm các tham số lớp cao hơn *transmissionComb*. Dịch chuyển tuần hoàn a_i cho tổng anten p_i được cho bởi

$$a_i = 2\pi n_{\text{SRS}}^{cs,i} / n_{\text{SRS}}^{cs,max}$$

$$n_{\text{SRS}}^{cs,i} = (n_{\text{SRS}}^{cs} + \frac{n_{\text{SRS}}^{cs,max} (p_i - 1000)}{N_{ap}^{\text{SRS}}})$$

với $n_{\text{SRS}}^{cs} \in \{0,1,\dots, n_{\text{SRS}}^{cs,max} - 1\}$ bao gồm trong tham số lớp cao hơn *transmissionComb*. Số vòng dịch chuyển tuần hoàn tối đa $n_{\text{SRS}}^{cs,max}$ được cho bởi Bảng 6.4.1.4.2-1

Table 6.4.1.4.2-1: Maximum number of cyclic shifts $n_{\text{SRS}}^{cs,max}$ as a function of K_{TC} .

K_{TC}	$n_{\text{SRS}}^{cs,max}$
2	8
4	12
8	6

- Sequence group $u = (f_{gh}(n_{s,f}^\mu, l') + n_{ID}^{\text{SRS}}) \bmod 30$ và số chuỗi v phụ thuộc vào tham số lớp cao hơn *groupOrSequenceHopping* trong *SRS-Resource* IE hoặc *SRS-PosResource-r16* IE. SRS sequence identity n_{ID}^{SRS} được cho bởi tham số lớp cao hơn *sequenceId* trong *SRS-Resource* IE, trong trường hợp $n_{ID}^{\text{SRS}} \in \{0,1,\dots,1023\}$, hoặc *SRS-PosResource-16* IE, trong trường hợp $n_{ID}^{\text{SRS}} \in \{0,1,\dots,65535\}$. Số lượng $l' \in \{0,1,\dots, N_{\text{symb}}^{\text{SRS}} - 1\}$ trong số ký tự OFDM trong SRS resource

+ Nếu *groupOrSequenceHopping* bằng “neither”, neither group, nor sequence hopping sẽ được sử dụng và $f_{gh}(n_{s,f}^\mu, l') = 0$ và $v=0$

+ Nếu *groupOrSequenceHopping* bằng “groupHopping”, group hopping, nhưng không sequence hopping sẽ được sử dụng và $(f_{gh}(n_{s,f}^\mu, l') = (\sum_{n=0}^7 c(8(n_{s,f}^\mu n_{\text{symb}}^{\text{slot}} + l_0 + l') + m). 2^m) \bmod 30$ với chuỗi pseudo-random $c(i)$ được cho bởi và sẽ được thiết lập với $c_{init} = N_{ID}^{\text{SRS}}$ tại vị trí bắt đầu của mỗi khung vô tuyến.

+ Nếu *groupOrSequenceHopping* bằng “sequenceHopping”, sequence hopping nhưng không group hopping sẽ được sử dụng và

$$f_{gh}(n_{s,f}^\mu, l') = 0$$

$$v = \begin{cases} c(n_{s,f}^{\mu} n_{symb}^{slot} + l_0 + l'), & M_{sc,b}^{SRS} \geq 6N_{sc}^{RB} \\ 0, & \text{khác} \end{cases}$$

với chuỗi pseudo-random $c(i)$ được cho bởi và sẽ được thiết lập với $c_{init} = N_{ID}^{SRS}$ tại vị trí bắt đầu của mỗi khung vô tuyến.

4.3. Mapping to physical resources

- Khi SRS được truyền trên SRS resource có sẵn, chuỗi $r^{(p_i)}(n, l')$ cho mỗi ký tự OFDM l' và cho mỗi tổng anten của SRS resources sẽ được nhân với hệ số tỷ lệ biên độ β_{SRS} để tuân theo năng lượng truyền và được ánh xạ tới chuỗi bắt đầu với $r^{(p_i)}(0, l')$ để các thành phần tài nguyên (k, l) trong một slot cho mỗi cổng anten p_i tuân theo

$$a_{K_{TC}k' + k_0^{(p_i)}, l' + l_0}^{(p_i)} = \begin{cases} \frac{1}{\sqrt{N_{ap}}} \beta_{SRS} r^{(p_i)}(k', l'), & k' = 0, 1, \dots, M_{sc,b}^{SRS} - 1 \quad l' = 0, 1, \dots, N_{symb}^{SRS} - 1 \\ 0, & \text{khác} \end{cases}$$

- Chiều dài của chuỗi SRS được cho bởi

$$M_{sc,b}^{SRS} = m_{SRS,b} N_{sc}^{RB} / K_{TC}$$

với $m_{SRS,b}$ được cho bởi cột được chọn trong Bảng 6.4.1.4.3-1 với $b = B_{SRS}$ với $B_{SRS} \in \{0, 1, 2, 3\}$ được cho bởi miền b -SRS được bao gồm trong tham số lớp cao hơn $freqHopping$ nếu được cấu hình, nếu không thì $B_{SRS} = 0$. Cột của bảng được chọn tuân theo chỉ số $C_{SRS} \in \{0, 1, \dots, 63\}$ được cho bởi miền c -SRS được bao gồm trong tham số lớp cao hơn $freqHopping$

- Vị trí xuất phát miền tần số $k_0^{(p_i)}$ được định nghĩa bởi

$$k_0^{(p_i)} = \bar{k}_0^{(p_i)} + \sum_{b=0}^{B_{SRS}} K_{TC} M_{sc,b}^{SRS} n_b$$

với $\bar{k}_0^{(p_i)} = n_{shift} N_{sc}^{RB} + (k_{TC}^{(p_i)} + k_{offset}^{l'}) \bmod K_{TC}$

$$k_{TC}^{(p_i)} = \begin{cases} (\bar{k}_{TC} + K_{TC}/2) \bmod K_{TC}, & \text{nếu } n_{SRS}^{CS} \in \{n_{SRS}^{cs,max}/2, \dots, n_{SRS}^{cs,max} - 1\} \text{ và } N_{ap}^{SRS} = 4 \text{ và } p_i \in \{1001, 1003\} \\ \bar{k}_{TC}, & \text{khác} \end{cases}$$

Nếu $N_{BWP}^{start} \leq n_{shift}$ điểm tham chiếu với $k_0^{(p_i)} = 0$ là sóng mang con 0 trong khối tài nguyên chung 0, nếu không thì điểm tham chiếu là sóng mang con thấp nhất của BWP

Nếu SRS được cấu hình bởi IE $SRS\text{-}PosResource$, số lượng $k_{offset}^{l'}$ được cho bởi Bảng 6.4.1.4.3-2, nếu không thì $k_{offset}^{l'} = 0$

Table 6.4.1.4.3-2: The offset $k_{offset}^{l'}$ for SRS as a function of K_{TC} and l' .

K_{TC}	$N_{symb}^{SRS} = 1$	$N_{symb}^{SRS} = 2$	$N_{symb}^{SRS} = 4$	$N_{symb}^{SRS} = 8$	$N_{symb}^{SRS} = 12$
2	0	0, 1	0, 1, 0, 1	-	-
4	-	0, 2	0, 2, 1, 3	0, 2, 1, 3, 0, 2, 1, 3	0, 2, 1, 3, 0, 2, 1, 3, 0, 2, 1, 3
8	-	-	0, 4, 2, 6	0, 4, 2, 6, 1, 5, 3, 7	0, 4, 2, 6, 1, 5, 3, 7, 0, 4, 2, 6

- Giá trị dịch chuyển miền tần số n_{shift} điều chỉnh phân bổ SRS với lưới điểm tham chiếu và bao gồm trong tham số lớp cao hơn $freqDomainShift$ trong $SRS\text{-}Resource$ IE hoặc $SRS\text{-}PosResource$ IE.

Transmission comb offset $K_{TC} \in \{0, 1, \dots, K_{TC} - 1\}$ chứa trong tham số lớp cao hơn *transmissionComb* trong *SRS-Resource* IE hoặc *SRS-PosResource* IE và n_b là chỉ số vị trí tần số

- Frequency hopping của tín hiệu SRS được cấu hình bởi tham số $b_{hop} \in \{0, 1, 2, 3\}$ được cho bởi miền $b-hop$ chứa trong tham số lớp cao hơn *freqHopping* nếu được cấu hình, nếu không thì $b_{hop} = 0$

+ Nếu $b_{hop} \geq B_{SRS}$, frequency hopping disabled và chỉ số vị trí tần số n_b giữ là hằng số (nếu không được tái cấu hình) và được định nghĩa là $n_b = [4n_{RRC}/m_{SRS,b}] \bmod N_b$ với tất cả N_{symb}^{SRS} ký tự OFDM của tài nguyên SRS. Số lượng n_{RRC} được cho bởi tham số lớp cao hơn *freqDomainPosition* nếu được cấu hình, nếu không thì $n_{RRC} = 0$. Các giá trị của $m_{SRS,b}$ và N_b với $b = B_{SRS}$ được cho bởi cột được chọn trong Bảng 6.4.1.4.3-1 tương ứng với giá trị được cấu hình của C_{SRS}

+ Nếu $b_{hop} \geq B_{SRS}$, frequency hopping enabled và chỉ số vị trí tần số n_b được định nghĩa là

$$n_b = (F_b(n_{SRS}) + [4n_{RRC}/m_{SRS,b}] \bmod N_b$$

với N_b được cho bởi Bảng 6.4.1.4.3-1-SRS bandwidth configuration

4.4. Cấu hình slot SRS

- Với một tài nguyên SRS được cấu hình tuần hoàn hoặc semi-persistent bởi tham số lớp cao hơn *resourceType*, một periodicity T_{SRS} (trong slot) và slot offset T_{offset} được cấu hình thông qua tham số lớp cao hơn *periodicityAndOffset-sp* trong *SRS-PosResource* IE. Candidate slot tại đó tài nguyên SRS được cấu hình có thể được dùng để truyền SRS là những slot thỏa mãn $(N_{slot}^{frame, \mu_{nf}} + n_{s,f}^{\mu} - T_{offset}) \bmod T_{SRS} = 0$