

Báo cáo nghiên cứu kênh Downlink 5G NR Sub6

Yêu cầu: Tìm hiểu tuyến kênh Downlink 5G NR Sub 6 bao gồm tín hiệu PSS, SSS, CSI RS, kênh PBCH, PDCCH, PDSCH

1. Tín hiệu PSS

- Có 1008 physical-layer cell identities khác nhau được cho bởi

$$N_{ID}^{cell} = 3N_{ID}^{(1)} + N_{ID}^{(2)}$$

với $N_{ID}^{(1)} \in \{0, 1, \dots, 335\}$ và $N_{ID}^{(2)} \in \{0, 1, 2\}$

- PSS là specific physical layer signal được sử dụng để đồng bộ hóa khung vô tuyến. Nó có các đặc điểm như được liệt kê dưới đây.

+ Ánh xạ tới 127 sóng mang con đang hoạt động xung quanh trung tâm khối SSB nằm ở sóng mang con 56~182

+ Được đặt ở ký tự OFDM đầu tiên (ký tự 0) trong SSB Burst

+ Được tạo thành từ 127 giá trị chuỗi m

+ Được sử dụng để đồng bộ hóa khung đường xuống

+ Một trong những yếu tố quan trọng quyết định Physical Cell ID

1.1.Sequence generation: Chuỗi $d_{PSS}(n)$ cho primary synchronization signal được xác định bởi

$$d_{PSS}(n) = 1 - 2x(m)$$

$$m = (n + 43N_{ID}^{(2)}) \bmod 127 \quad (0 \leq n < 127)$$

với

$$x(i+7) = (x(i+4) + x(i)) \bmod 2$$

và

$$[x(6) \ x(5) \ x(4) \ x(3) \ x(2) \ x(1) \ x(0)] = [0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0]$$

1.2. Mapping to physical resources

- Frequency Domain Resource Allocation

Cấu trúc tần số - thời gian của khối SS/PBCH

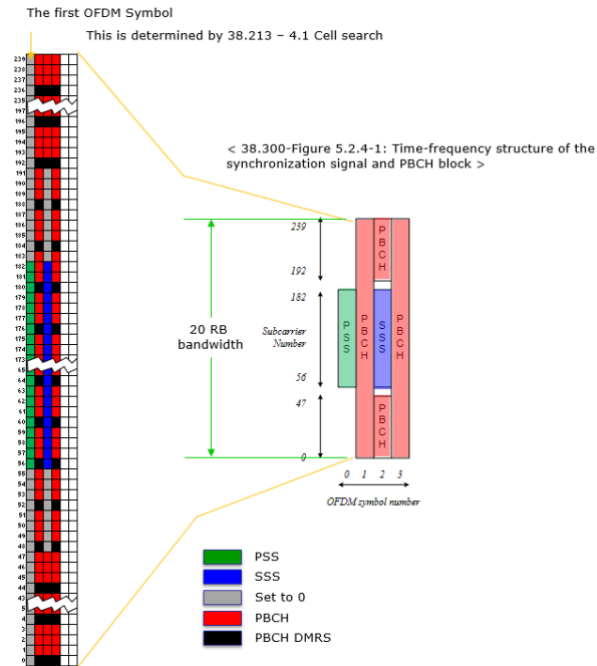
+ Trong miền thời gian, khối SS/PBCH bao gồm 4 ký tự OFDM, được đánh số theo thứ tự tăng dần từ 0 đến 3 trong phạm vi khối SS/PBCH, trong đó PSS, SSS và PBCH với DM-RS liên quan được ánh xạ tới các ký tự như được đưa ra

+ Trong miền tần số, một khối SS/PBCH bao gồm 240 sóng mang con liền kề với các sóng mang con được đánh số bằng thứ tự tăng dần từ 0 lên 239 trong khối SS/PBCH. Các đại lượng k và l biểu thị time domain (OFDM symbol number) và frequency domain (Subcarrier Number) các chỉ số tương ứng trong một khối SS/PBCH trong bảng dưới

Table 7.4.3.1-1: Resources within an SS/PBCH block for PSS, SSS, PBCH, and DM-RS for PBCH.

Channel or signal	OFDM symbol number l relative to the start of an SS/PBCH block	Subcarrier number k relative to the start of an SS/PBCH block
PSS	0	56, 57, ..., 182
SSS	2	56, 57, ..., 182
Set to 0	0	0, 1, ..., 55, 183, 184, ..., 239
	2	48, 49, ..., 55, 183, 184, ..., 191
PBCH	1, 3	0, 1, ..., 239
	2	0, 1, ..., 47, 192, 193, ..., 239
DM-RS for PBCH	1, 3	$0 + v, 4 + v, 8 + v, \dots, 236 + v$
	2	$0 + v, 4 + v, 8 + v, \dots, 44 + v$ $192 + v, 196 + v, \dots, 236 + v$

+ Ví dụ: Bảng trên có thể được biểu diễn trong Lưới tài nguyên như dưới đây. Lưu ý rằng vị trí của PBCH DM-RS thay đổi theo v và giá trị v thay đổi tùy thuộc vào ID ô vật lý.



- Các ký tự có giá trị phức tương ứng đối với các phần tử tài nguyên được biểu thị là Set to 0'
- Đại lượng v trong bảng 7.4.3.1-1 được cho bởi $v = N_{ID}^{cell} \bmod 4$.
- Đại lượng k_{SSB} là độ lệch sóng mang con từ sóng mang con 0 trong khối tài nguyên chung CRB SSB tới sóng mang con 0 của khối SS/PBCH, trong đó CRB SSB được lấy từ tham số *offsetToPointA* của lớp cao hơn và 4 các bit có ý nghĩa nhỏ nhất của k_{SSB} được cung cấp bởi tham số lớp cao hơn *ssb-SubcarrierOffset* và đối với loại khối SS/PBCH
- Bit quan trọng nhất của k_{SSB} được cho bởi \bar{a}_{A+5} trong payload PBCH
- Các ký tự có giá trị phức tạp tương ứng với các phần tử tài nguyên là một phần của chung khối tài nguyên chồng chéo một phần hoặc toàn bộ với khối SS/PBCH và không được sử dụng để truyền SS/PBCH được đặt thành 0 trong các ký tự OFDM trùng lặp một phần hoặc hoàn toàn với các ký tự OFDM nơi SS/PBCH được truyền đi.
- Đối với khối SS/PBCH, UE
 - công ăng ten $p = 4000$ được sử dụng để truyền PSS, SSS, PBCH và DM-RS cho PBCH,
 - cùng CP và khoảng cách sóng mang con cho PSS, SSS, PBCH và DM-RS cho PBCH,
 - đối với khối SS/PBCH loại A (Sub 6)
 - $\mu \in \{0,1\}$, FR1 (sub 6GHz)
 - $k_{SSB} \in \{0,1, 2, \dots, 23\}$ (4 bit LSB có giá trị k_{SSB} có thể được thông báo tới UE thông qua *ssb-subcarrierOffset* trong MIB; Bit MSB được thông báo tới UE thông qua một bit trong dữ liệu PBCH (\bar{a}_{A+5}); được biểu thị bằng khoảng cách sóng mang con 15 Khz)
 - N_{CRB}^{SSB} được biểu thị bằng khoảng cách sóng mang con 15 Khz
 - đối với khối SS/PBCH loại B (mmWave)
 - $\mu \in \{3,4\}$, FR2(mmWave)
 - $k_{SSB} \in \{0,1, 2, \dots, 11\}$ (tồn bộ giá trị k_{SSB} có thể được thông báo tới UE thông qua *ssb-subcarrierOffset* trong MIB ; với đại lượng k_{SSB} được biểu thị bằng Khoảng cách sóng mang con được cung cấp bởi tham số lớp cao hơn *subCarrierSpacingCommon*)
 - N_{CRB}^{SSB} được biểu thị bằng khoảng cách sóng mang con 60 kHz;
 - Tâm sóng mang con 0 của khối tài nguyên N_{CRB}^{SSB} trùng với tâm sóng mang con 0 của tài nguyên chung khối có khoảng cách sóng mang con được cung cấp bởi tham số lớp cao

hơn $subCarrierSpacingCommon$. Khối tài nguyên chung này chồng lấp với sóng mang con 0 của khối tài nguyên đầu tiên của khối SS/PBCH.

- Các khối SS/PBCH được truyền với cùng chỉ số khối trên cùng một vị trí tần số trung tâm gần như nằm cùng vị trí với độ trải Doppler, độ dịch chuyển Doppler, mức tăng trung bình, độ trễ trung bình, độ trễ trễ và khi các tham số Rx không gian có thể áp dụng.

+ Ánh xạ PSS trong khối SS/PBCH

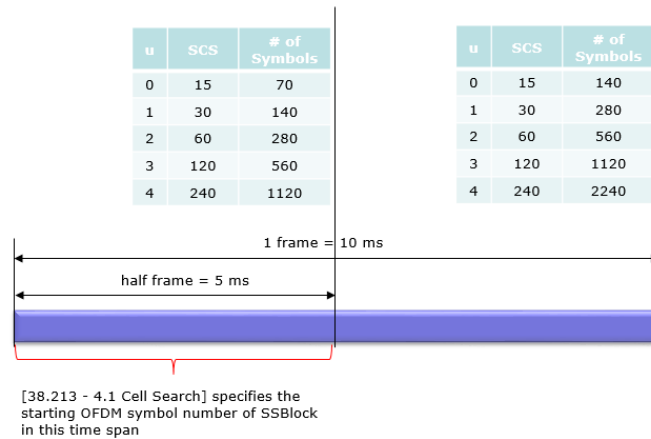
- Chuỗi ký tự $d_{PSS}(0), \dots, d_{PSS}(126)$ tạo thành tín hiệu đồng bộ chính được chia tỷ lệ theo hệ số β_{PSS} để phù hợp với phân bố công suất PSS và ánh xạ tới tài nguyên các phần tử $(k, l)_{p,u}$, theo thứ tự tăng dần của k trong đó k và l được cho trong Bảng 7.4.3.1-1 và biểu thị tần số và chỉ số thời gian tương ứng trong một khối SS/PBCH.

- Time Domain Resource Allocation

+ Bảng sau cho biết số ký tự OFDM đầu tiên SS/PBCH được truyền đi.

+ Đối với nửa khung có các khối SS/PBCH, số lượng và chỉ số ký tự đầu tiên cho các khối SS/PBCH dự kiến được xác định theo khoảng cách sóng mang con của các khối SS/PBCH như sau.

+ Cell Search chỉ định vị trí SS/PBCH trong miền thời gian như minh họa bên dưới.

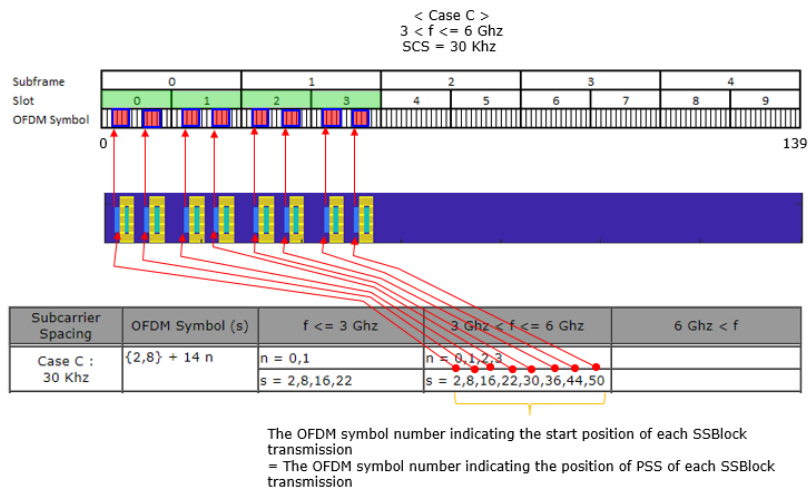
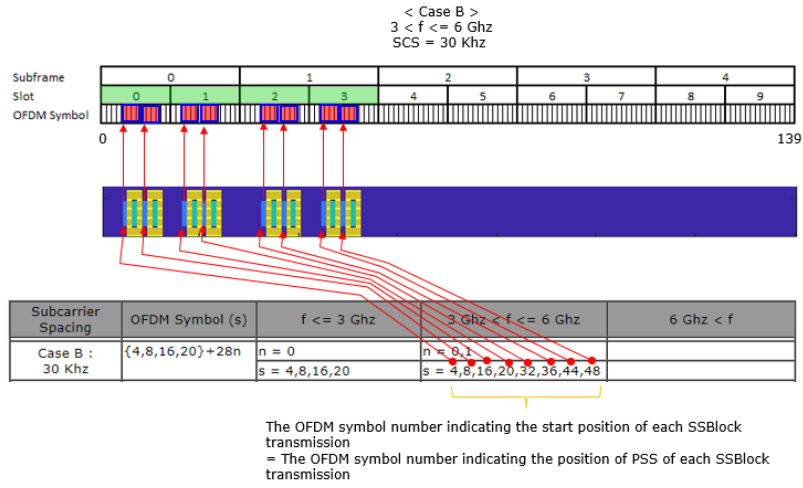


Start Symbols for each subcarrier spacing and frequency

+ Các vị trí trong miền thời gian nơi UE sẽ giám sát khối SS/PBCH có thể xảy ra

+ Ví dụ về Truyền SSB cho từng trường hợp. Ví dụ dưới đây cho thấy cách có thể liên hệ bảng trên với biểu đồ truyền SSB được hiển thị trong các ví dụ sau

- Đặt vị trí miền tần số của khối SSB nằm ở dưới cùng của băng thông hệ thống, nhưng trên thực tế, vị trí miền tần số có thể thay đổi sang vị trí khác (ví dụ: tần số trung tâm của băng thông hệ thống).
- Mục đích: hiển thị vị trí miền thời gian (mẫu truyền) của từng trường hợp. Trong triển khai thực tế, rất có thể (nhưng không nhất thiết) vị trí miền tần số của SSB nằm xung quanh tần số trung tâm.



2. Tín hiệu SSS

- Có 1008 physical-layer cell identities khác nhau được cho bởi

$$N_{ID}^{cell} = 3N_{ID}^{(1)} + N_{ID}^{(2)}$$

với $N_{ID}^{(1)} \in \{0, 1, \dots, 335\}$ và $N_{ID}^{(2)} \in \{0, 1, 2\}$

- SSS là specific physical layer signal được sử dụng để đồng bộ hóa khung vô tuyến. Nó có các đặc điểm như được liệt kê dưới đây.

- + Ánh xạ tới 127 sóng mang con đang hoạt động xung quanh trung tâm khối SSB nằm ở sóng mang con 56~182
- + Được đặt ở ký tự OFDM thứ ba (ký tự 2) trong SSB Burst
- + Được tạo thành từ 127 giá trị chuỗi m
- + Được sử dụng để đồng bộ hóa khung đường xuống
- + Một trong những yếu tố quan trọng quyết định Physical Cell ID

2.1.Sequence generation: Chuỗi $d_{SSS}(n)$ cho primary synchronization signal được xác định bởi

$$d_{SSS}(n) = [1 - 2x_0((n + m_0) \bmod 127)][1 - 2x_1((n + m_1) \bmod 127)]$$

$$m_0 = 15[N_{ID}^{(1)} / 112] + 5N_{ID}^{(2)}$$

$$m_1 = (N_{ID}^{(1)}) \bmod 112 (0 \leq n < 127)$$

với

$$x_0(i + 7) = (x_0(i + 4) + x_0(i)) \bmod 2$$

$$x_1(i + 7) = (x_1(i + 1) + x_1(i)) \bmod 2$$

và

$$\begin{aligned} [x_0(6) \ x_0(5) \ x_0(4) \ x_0(3) \ x_0(2) \ x_0(1) \ x_0(0)] &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1] \\ [x_1(6) \ x_1(5) \ x_1(4) \ x_1(3) \ x_1(2) \ x_1(1) \ x_1(0)] &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1] \end{aligned}$$

2.2. Mapping to physical resources

+ Ánh xạ SSS trong khối SS/PBCH

- Chuỗi ký tự $d_{SSS}(0), \dots, d_{SSS}(126)$ tạo thành tín hiệu đồng bộ thứ cấp được chia tỷ lệ theo hệ số β_{SSS} và ánh xạ tới tài nguyên các phần tử $(k, l)_{p,u}$, theo thứ tự tăng dần của k trong đó k và l được cho trong Bảng 7.4.3.1-1 và biểu thị các chỉ số tần số và thời gian tương ứng trong một khối SS/PBCH.

3. Tín hiệu CSI RS

- Zero-power (ZP) và non-zero-power (NZP) CSI-RS được xác định

+ Với non-zero-power (NZP) CSI-RS được định cấu hình bởi *NZP-CSI-RS-Resource* IE hoặc bởi *CSI-RS-Resource-Mobility* field trong *CSI-RS-ResourceConfigMobility* IE, trình tự sẽ được tạo và ánh xạ tới các phần tử tài nguyên trong các phần dưới

+ Với zero-power (ZP) CSI-RS được định cấu hình bởi *ZP-CSI-RS-IE Resource*, UE sẽ giả định rằng các phần tử tài nguyên được xác định tại mục 3.2. không được sử dụng để truyền PDSCH resource mapping với RE level granularity. UE thực hiện cùng một phép đo/thu trên các kênh/tín hiệu ngoại trừ PDSCH bất kể có va chạm với ZP CSI-RS hay không.

3.1. Sequence generation

- Chuỗi tín hiệu tham chiếu $r(m)$ được định nghĩa bởi

$$r(m) = \frac{1}{\sqrt{2}}(1 - 2 \cdot c(2m)) + j \frac{1}{\sqrt{2}}(1 - 2 \cdot c(2m + 1))$$

với chuỗi mã giả $c(i)$ được định nghĩa là một chuỗi Gold chiều dài 31. Chuỗi đầu ra $c(n)$ của chiều dài M_{PN} với $n=0,1,\dots, M_{PN}-1$ được định nghĩa bởi

$$c(n) = (x_1(n + N_C) + x_2(n + N_C)) \bmod 2$$

$$x_1(n + 31) = (x_1(n + 3) + x_1(n)) \bmod 2$$

$$x_2(n + 31) = (x_2(n + 3) + x_2(n + 2) + x_2(n + 1) + x_2(n)) \bmod 2$$

với $N_C = 1600$ và chuỗi-m đầu tiên $x_1(n)$ được thiết lập với $x_1(0)=1, x_1(n) = 0 \ (n = 1,2,\dots,30)$.

Và chuỗi-m thứ hai $x_2(n)$ được định nghĩa bởi $c_{init} = \sum_{i=0}^{30} x_2(i) \cdot 2^i$ với giá trị phụ thuộc vào ứng dụng của chuỗi

- Chuỗi mã giả được thiết lập với

$$c_{init} = (2^{10}(N_{symb}^{slot} n_{s,f}^{\mu} + l + 1)(2n_{ID} + 1) + n_{ID}) \bmod 2^{31}$$

ở đầu mỗi ký hiệu OFDM trong đó $n_{s,f}^{\mu}$ là số khe trong khung vô tuyến, l là số ký hiệu OFDM trong một khe, và n_{ID} bằng với tham số *scramblingID* hoặc *SequenceGenerationConfig* của tham số lớp cao hơn.

3.2. Mapping to physical resources

- $[b_5, \dots, b_0]$, $k_{i-1} = 2f(i)$ cho tất cả các trường hợp khác trong đó $f(i)$ là số bit của bit thứ i trong bitmap được đặt thành một, được lặp lại trên mỗi $\lceil 1/\rho \rceil$ của các khối tài nguyên được cấu hình để UE tiếp nhận CSI-RS.
- Vị trí bắt đầu và số khối tài nguyên trong đó UE phải giả định rằng CSI-RS được truyền đi được cung cấp bởi các tham số *freqBand* và mật độ lớp cao hơn trong IE *CSI-RSResourceMapping* IE cho phân băng thông được cung cấp bởi tham số lớp cao hơn BWP-Id trong *CSI-ResourceConfig* IE hoặc được cung cấp bởi các tham số *nrofPRB* lớp cao hơn trong *CSI-RS-CellMobility* IE trong đó *startPRB* do *csirs-MeasurementBW* cung cấp có liên quan đến khối tài nguyên chung 0.
- CSI-RS được truyền bằng cách sử dụng các cổng ăng ten p được đánh số theo

$$p = 3000 + s + jL$$

$$j = 0, 1, \dots, N/L - 1$$

$$s = 0, 1, \dots, L - 1$$

trong đó s là chỉ số trình tự được cung cấp bởi các Bảng 7.4.1.5.3-2 đến 7.4.1.5.3-5, $L \in \{1, 2, 4, 8\}$ là kích thước nhóm CDM và N là số lượng cổng CSI-RS.

Chỉ số nhóm CDM j đưa ra trong Bảng 7.4.1.5.3-1 tương ứng với vị trí thời gian/tần số (\bar{k}, \bar{l}) cho một hàng nhất định của bảng.

Các nhóm CDM được đánh số theo thứ tự tăng dần phân bổ miền tần số trước và sau đó tăng phân bổ miền thời gian.

Đối với tài nguyên CSI-RS được định cấu hình là định kỳ hoặc bán liên tục bởi loại tài nguyên tham số lớp cao hơn hoặc được định cấu hình bởi tham số lớp cao hơn *CSI-RS-CellMobility*, CSI-RS được truyền trong các khe đáp ứng

$$(N_{slot}^{frame, \mu} n_f + n_{s,f}^{\mu} - T_{offset}) \bmod T_{CSI-RS} = 0$$

trong đó tính tuần hoàn T_{CSI-RS} (trong các vị trí) và T_{offset} bù vị trí được lấy từ tham số lớp cao hơn *CSIResourcePeriodicityAndOffset* hoặc *slotConfig*. CSI-RS được truyền trong một khe.

- Các cổng ăng ten trong tài nguyên CSI-RS gần như nằm cùng vị trí với QCL Loại A, Loại D (nếu có) và mức tăng trung bình.

Table 7.4.1.5.3-1: CSI-RS locations within a slot.

Row	Ports X	Density ρ	cdm-Type	(\bar{k}, \bar{l})	CDM group index j	k'	l'
1	1	3	noCDM	$(k_0, l_0), (k_0 + 4, l_0), (k_0 + 8, l_0)$	0,0,0	0	0
2	1	1, 0.5	noCDM	$(k_0, l_0),$	0	0	0
3	2	1, 0.5	fd-CDM2	$(k_0, l_0),$	0	0, 1	0
4	4	1	fd-CDM2	$(k_0, l_0), (k_0 + 2, l_0)$	0,1	0, 1	0
5	4	1	fd-CDM2	$(k_0, l_0), (k_0, l_0 + 1)$	0,1	0, 1	0
6	8	1	fd-CDM2	$(k_0, l_0), (k_1, l_0), (k_2, l_0), (k_3, l_0)$	0,1,2,3	0, 1	0
7	8	1	fd-CDM2	$(k_0, l_0), (k_1, l_0), (k_0, l_0 + 1), (k_1, l_0 + 1)$	0,1,2,3	0, 1	0
8	8	1	cdm4-FD2-TD2	$(k_0, l_0), (k_1, l_0)$	0,1	0, 1	0, 1
9	12	1	fd-CDM2	$(k_0, l_0), (k_1, l_0), (k_2, l_0), (k_3, l_0), (k_4, l_0), (k_5, l_0)$	0,1,2,3,4,5	0, 1	0
10	12	1	cdm4-FD2-TD2	$(k_0, l_0), (k_1, l_0), (k_2, l_0)$	0,1,2	0, 1	0, 1
11	16	1, 0.5	fd-CDM2	$(k_0, l_0), (k_1, l_0), (k_2, l_0), (k_3, l_0), (k_0, l_0 + 1), (k_1, l_0 + 1), (k_2, l_0 + 1), (k_3, l_0 + 1)$	0,1,2,3,4,5,6,7	0, 1	0
12	16	1, 0.5	cdm4-FD2-TD2	$(k_0, l_0), (k_1, l_0), (k_2, l_0), (k_3, l_0)$	0,1,2,3	0, 1	0, 1
13	24	1, 0.5	fd-CDM2	$(k_0, l_0), (k_1, l_0), (k_2, l_0), (k_0, l_0 + 1), (k_1, l_0 + 1), (k_2, l_0 + 1), (k_0, l_1), (k_1, l_1), (k_2, l_1), (k_0, l_1 + 1), (k_1, l_1 + 1), (k_2, l_1 + 1)$	0,1,2,3,4,5,6,7,8,9,10,11	0, 1	0
14	24	1, 0.5	cdm4-FD2-TD2	$(k_0, l_0), (k_1, l_0), (k_2, l_0), (k_0, l_1), (k_1, l_1), (k_2, l_1)$	0,1,2,3,4,5	0, 1	0, 1
15	24	1, 0.5	cdm8-FD2-TD4	$(k_0, l_0), (k_1, l_0), (k_2, l_0)$	0,1,2	0, 1	0, 1, 2, 3
16	32	1, 0.5	fd-CDM2	$(k_0, l_0), (k_1, l_0), (k_2, l_0), (k_3, l_0), (k_0, l_0 + 1), (k_1, l_0 + 1), (k_2, l_0 + 1), (k_3, l_0 + 1), (k_0, l_1), (k_1, l_1), (k_2, l_1), (k_3, l_1), (k_0, l_1 + 1), (k_1, l_1 + 1), (k_2, l_1 + 1), (k_3, l_1 + 1)$	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15	0, 1	0
17	32	1, 0.5	cdm4-FD2-TD2	$(k_0, l_0), (k_1, l_0), (k_2, l_0), (k_3, l_0), (k_0, l_1), (k_1, l_1), (k_2, l_1), (k_3, l_1)$	0,1,2,3,4,5,6,7	0, 1	0, 1
18	32	1, 0.5	cdm8-FD2-TD4	$(k_0, l_0), (k_1, l_0), (k_2, l_0), (k_3, l_0)$	0,1,2,3	0,1	0,1,2,3

Table 7.4.1.5.3-2: The sequences $w_f(k')$ and $w_t(l')$ for cdm-Type equal to 'noCDM'.

Index	$w_f(0)$	$w_t(0)$
0	1	1

Table 7.4.1.5.3-3: The sequences $w_f(k')$ and $w_t(l')$ for cdm-Type equal to 'fd-CDM2'.

Index	$[w_f(0) \ w_f(1)]$	$w_t(0)$
0	$[+1 \ +1]$	1
1	$[+1 \ -1]$	1

Table 7.4.1.5.3-4: The sequences $w_f(k')$ and $w_t(l')$ for cdm-Type equal to 'cdm4-FD2-TD2'.

Index	$[w_f(0) \ w_f(1)]$	$[w_t(0) \ w_t(1)]$
0	$[+1 \ +1]$	$[+1 \ +1]$
1	$[+1 \ -1]$	$[+1 \ +1]$
2	$[+1 \ +1]$	$[+1 \ -1]$
3	$[+1 \ -1]$	$[+1 \ -1]$

Table 7.4.1.5.3-5: The sequences $w_f(k')$ and $w_t(l')$ for cdm-Type equal to 'cdm8-FD2-TD4'.

Index	$[w_f(0) \ w_f(1)]$	$[w_t(0) \ w_t(1) \ w_t(2) \ w_t(3)]$
0	$[+1 \ +1]$	$[+1 \ +1 \ +1 \ +1]$
1	$[+1 \ -1]$	$[+1 \ +1 \ +1 \ +1]$
2	$[+1 \ +1]$	$[+1 \ -1 \ +1 \ -1]$
3	$[+1 \ -1]$	$[+1 \ -1 \ +1 \ -1]$
4	$[+1 \ +1]$	$[+1 \ +1 \ -1 \ -1]$
5	$[+1 \ -1]$	$[+1 \ +1 \ -1 \ -1]$
6	$[+1 \ +1]$	$[+1 \ -1 \ -1 \ +1]$
7	$[+1 \ -1]$	$[+1 \ -1 \ -1 \ +1]$

4. Kênh PBCH - Physical broadcast channel

4.1. Tổng quan kênh PBCH (Physical Broadcast Channel)

+ Quá trình truyền tải PBCH bao gồm một loạt các bước để chuẩn bị và truyền thông tin Kênh PBCH. PBCH là thành phần chính trong mạng truyền thông di động, đặc biệt là trong hệ thống LTE và 5G. Nó đóng vai trò là kênh chính mà qua đó trạm cơ sở (eNodeB trong LTE, gNodeB trong 5G) truyền MIB đến UE.

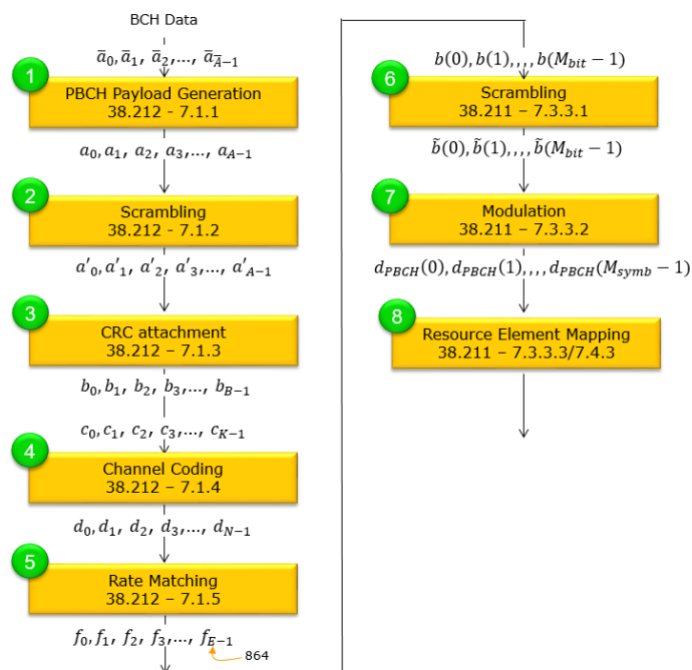
+ PBCH mang thông tin quan trọng mà UE cần để giải mã các thông tin quảng bá khác và thiết lập kết nối với mạng.

4.2. Cấu trúc kênh PBCH và thuật toán của các module trong kênh PBCH

+ Quy trình vận chuyển PBCH

(1) PBCH Payload Generation => (2) Scrambling => (3) CRC Attachment => (4) Channel Coding => (5) Rate Matching => (6) Scrambling => (7) Modulation => (8) Resource Element Mapping

- Sơ đồ sau minh họa Quy trình truyền tải PBCH trong hệ thống thông tin di động, trình bày chi tiết trình tự các bước liên quan đến việc chuẩn bị và truyền PBCH từ gNB đến UE



PBCH Payload Generation: Dữ liệu kênh BCH được tạo. Đây là bước đầu tiên chuẩn bị dữ liệu cần được phát sóng.

Scrambling: Payload được tạo ra sẽ bị xáo trộn. Bước này rất quan trọng để giảm thiểu nhiễu và duy trì tính toàn vẹn của tín hiệu.

CRC Attachment: Một Cyclic Redundancy Check (CRC) được gắn vào dữ liệu được xáo trộn nhằm mục đích phát hiện lỗi.

Channel Coding: Dữ liệu có CRC được mã hóa để bảo vệ khỏi các lỗi tiềm ẩn trong quá trình truyền.

Rate Matching: Dữ liệu được mã hóa sau đó được xử lý thông qua việc rate matching để đảm bảo rằng nó phù hợp với các tài nguyên truyền tải sẵn có.

Scrambling: Dữ liệu khớp với tỷ lệ trải qua quá trình xáo trộn thứ hai. Điều này có thể nhằm mục đích tăng cường bảo mật hoặc giảm thiểu nhiễu.

Modulation: Sau đó, dữ liệu được mã hóa sẽ được điều chế, nghĩa là nó được chuyển đổi thành dạng sóng phù hợp để truyền qua phổ tần số vô tuyến..

Resource Element Mapping: Cuối cùng, dữ liệu đã điều chế được ánh xạ lên các phần tử tài nguyên trong lưới tần số-thời gian để truyền thực tế..

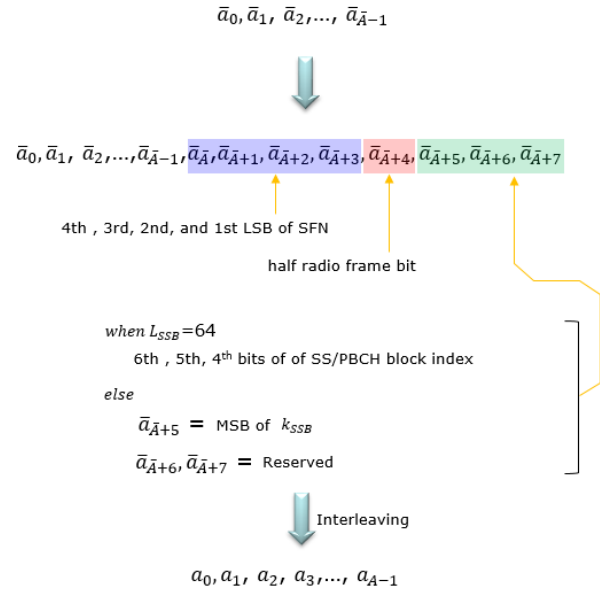
(1) PBCH Payload Generation

Hình minh họa quy trình PBCH Payload Generation, bước khởi đầu quan trọng. Quá trình này là công cụ tạo ra payload sẽ được truyền qua Kênh PBCH

- Chuỗi dữ liệu BCH bắt đầu bằng các bit được dán nhãn từ \bar{a}_0 đến $\bar{a}_{\bar{A}-1}$ (\bar{A} là kích thước payload được tạo bởi các lớp cao hơn). Chuỗi này sau đó được tăng cường bằng các bit bổ sung mã hóa thông tin hệ thống cụ thể:

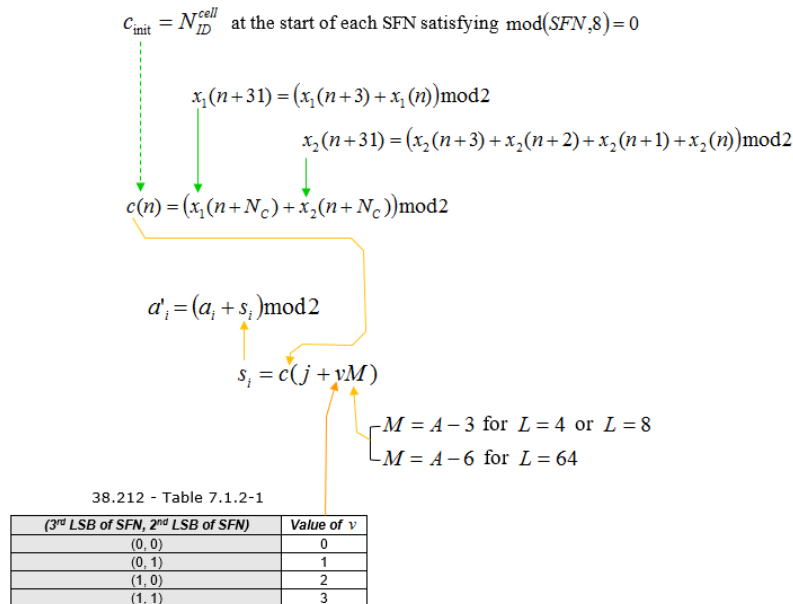
- Các bit $\bar{a}_{\bar{A}}, \bar{a}_{\bar{A}+1}, \bar{a}_{\bar{A}+2}, \bar{a}_{\bar{A}+3}$ tương ứng với LSB thứ 4, 3, 2 và 1 của System Frame Number (SFN)
- Bit $\bar{a}_{\bar{A}+4}$ biểu thị một nửa bit khung vô tuyến.
- Khi LSSB bằng 64, các bit $\bar{a}_{\bar{A}+5}, \bar{a}_{\bar{A}+6}, \bar{a}_{\bar{A}+7}$ lần lượt là các bit thứ 6, 5 và 4 của chỉ số khối SS/PBCH.
- Nếu LSSB không bằng 64 thì $\bar{a}_{\bar{A}+5}$ là MSB của k_{SSB} , với $\bar{a}_{\bar{A}+6}, \bar{a}_{\bar{A}+7}$ được bảo lưu.

- Sau khi các bit này được chèn vào, một interleaving process được áp dụng, sắp xếp lại chuỗi từ a_0 đến a_{A-1} , hoàn thành giai đoạn tạo payload sẵn sàng cho các bước truyền tiếp theo.



(2) Scrambling

- Hình dưới mô tả chi tiết quá trình scrambling. Quá trình bắt đầu bằng việc tạo ra chuỗi khởi tạo c_{init} phụ thuộc vào ID cell N_{cell}^{ID} và Số khung hệ thống (SFN). Trình tự được tạo cụ thể ở đầu mỗi SFN trong đó $\text{mod}(\text{SFN}, 8)$ bằng 0



- Chuỗi xáo trộn $c(n)$ được xây dựng bằng cách sử dụng hai chuỗi $x_1(n)$ và $x_2(n)$. Các chuỗi này được cập nhật đệ quy bằng cách sử dụng các mối quan hệ::

- $x_1(n+31) = (x_1(n+3) + x_1(n)) \bmod 2$
- $x_2(n+31) = (x_2(n+3) + x_2(n+2) + x_2(n+1) + x_2(n)) \bmod 2$

- Bit dữ liệu a_i được xáo trộn với bit s_i từ chuỗi xáo trộn $c(n)$, dẫn đến bit bị xáo trộn $a'_i = a_i + s_i \bmod 2$.

+ Giá trị của s_i được lấy từ chuỗi giả ngẫu nhiên được tạo bởi $c(j + vM)$, trong đó M được định nghĩa là $A - 3$ cho $L = 4$ hoặc $L = 8$ và $A - 6$ cho $L = 64$.

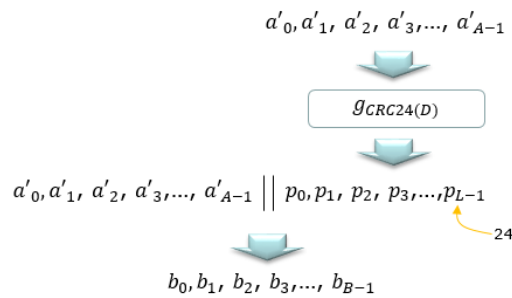
+ Bảng đi kèm ánh xạ LSB thứ ba và thứ hai của SFN tới giá trị của v , giá trị này ảnh hưởng đến việc tạo chuỗi giả ngẫu nhiên:

- Với (0,0) và (1,1), v là 0.
- Với (0,1) và (1,0), v là 1.
- Với (1,0) và (0,1), v là 2.
- Với (1,1) và (0,0), v là 3.

- Sự xáo trộn này đảm bảo độ rõ của tín hiệu được truyền đi, phân biệt nó với nhiễu và các tín hiệu khác để thu và giải mã chính xác.

(3) CRC Attachment

Hình dưới minh họa quy trình CRC Attachment, đây là bước thứ ba trong quy trình PBCH Transport Process.



- Chuỗi bit xáo trộn được biểu diễn như $a'_0, a'_1, a'_2, \dots, a'_{A-1}$. Một đa thức CRC, được biểu diễn dưới dạng $G_{CRC24}(D)$, được áp dụng cho chuỗi dữ liệu này, tạo ra một tập hợp các bit CRC $p_0, p_1, p_2, \dots, p_{L-1}$, với L là độ dài của CRC là 24 bit.

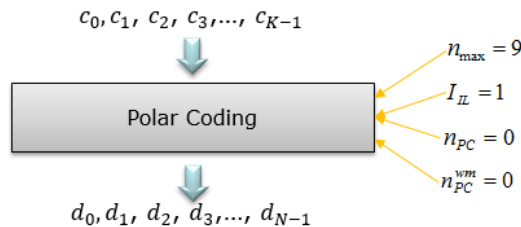
- $g_{CRC24A}(D) = [D^{24} + D^{23} + D^{18} + D^{17} + D^{14} + D^{11} + D^{10} + D^7 + D^6 + D^5 + D^4 + D^3 + D + 1]$
- $g_{CRC24B}(D) = [D^{24} + D^{23} + D^6 + D^5 + D + 1]$

- Các bit CRC này sau đó được thêm vào chuỗi dữ liệu gốc, tạo thành chuỗi dữ liệu được nối $a'_0, a'_1, a'_2, \dots, a'_{A-1} || p_0, p_1, p_2, \dots, p_{L-1}$. Chuỗi được nối thêm này đóng vai trò như một checksum để phát hiện lỗi ở đầu thu. Trình tự kết hợp, hiện bao gồm cả CRC, được ký tự là $b_0, b_1, b_2, \dots, b_{B-1}$.

- CRC attachment rất cần thiết để duy trì tính toàn vẹn của dữ liệu trong quá trình truyền và là phương pháp phổ biến trong các hệ thống truyền thông kỹ thuật số để phát hiện và sửa lỗi.

(4) Channel Coding

Sơ đồ sau đây trình bày bước Mã hóa kênh, đây là giai đoạn thứ tư trong PBCH Transport Process.



- Nó bắt đầu bằng một chuỗi bit, được biểu diễn dưới dạng $c_0, c_1, c_2, \dots, c_{K-1}$, sau đó được đưa vào một Polar Coding scheme.

- Polar Coding biến đổi các bit gốc thành một tập hợp các bit được mã hóa mới có các đặc tính giúp chúng chống lại các lỗi tốt hơn trên kênh truyền thông nhiễu. Các thông số như $n_{\max} = 9$, $I_L = 1$, $n_{PC} = 0$ và $n_{PC}^{wm} = 0$ có liên quan đến cấu hình cụ thể của quá trình Polar Coding.

- Đầu ra từ khối Polar Coding là một chuỗi bit được mã hóa mới $d_0, d_1, d_2, \dots, d_{N-1}$, đã sẵn sàng cho các bước tiếp theo trong quá trình truyền tải.

- Chuỗi được mã hóa này được thiết kế để có khả năng phục hồi tốt hơn trước tác động của nhiễu và nhiễu khi truyền qua giao diện vô tuyến.

(5) Rate matching

- Chuỗi bit đầu vào để rate matching là $d_0, d_1, d_2, \dots, d_{K-1}$.

- Độ dài chuỗi đầu ra rate matching $E = 864$.

- Chuỗi bit đầu ra sau khi rate matching là $f_0, f_1, f_2, \dots, f_{E-1}$

- Rate matching được thực hiện theo các quá trình sau (với $I_{BIL} = 0$)

+ Rate matching cho mã Polar được xác định trên mỗi khối được mã hóa và bao gồm sub-block interleaving, bit collection, và bit interleaving

+ Sub-block interleaving

- Các bit đầu vào của bộ xen kẽ khối con là các bit được mã hóa $d_0, d_1, d_2, \dots, d_{N-1}$. Các bit được mã hóa $d_0, d_1, d_2, \dots, d_{N-1}$ được chia thành 32 khối con.
- Đầu ra các bit từ bộ xen kẽ khối con được ký tự là $y_0, y_1, y_2, \dots, y_{N-1}$, được tạo ra dưới dạng sau:

for $n = 0$ to $N - 1$

$i = \lfloor 32n / N \rfloor$;

$J(n) = P(i) \times (N / 32) + \text{mod}(n, N / 32)$;

$y_n = d_{J(n)}$;

end for

trong đó sub-block interleaver pattern $P(i)$ được cho trong Bảng 5.4.1.1-1

Table 5.4.1.1-1: Sub-block interleaver pattern $P(i)$

i	$P(i)$	i	$P(i)$	i	$P(i)$	i	$P(i)$	i	$P(i)$	i	$P(i)$	i	$P(i)$	i	$P(i)$
0	0	4	3	8	8	12	10	16	12	20	14	24	24	28	27
1	1	5	5	9	16	13	18	17	20	21	22	25	25	29	29
2	2	6	6	10	9	14	11	18	13	22	15	26	26	30	30
3	4	7	7	11	17	15	19	19	21	23	23	27	28	31	31

+ Bit collection

- Chuỗi bit sau bộ xen kẽ khối con $y_0, y_1, y_2, \dots, y_{N-1}$ được ghi vào bộ đệm tròn có chiều dài N .
- Biểu thị bằng E độ dài chuỗi đầu ra rate matching, chuỗi bit đầu ra lựa chọn bit $e_k, k = 2, 1, 0, \dots, E - 1$, được tạo ra như sau:

if $E \geq N$ -- repetition

for $k = 0$ to $E - 1$

$e_k = y_{\text{mod}(k, N)}$;

end for

else

if $K / E \leq 7 / 16$ -- puncturing

for $k = 0$ to $E - 1$

$e_k = y_{k+N-E}$;

end for

else -- shortening

for $k = 0$ to $E - 1$

$e_k = y_k$;

end for

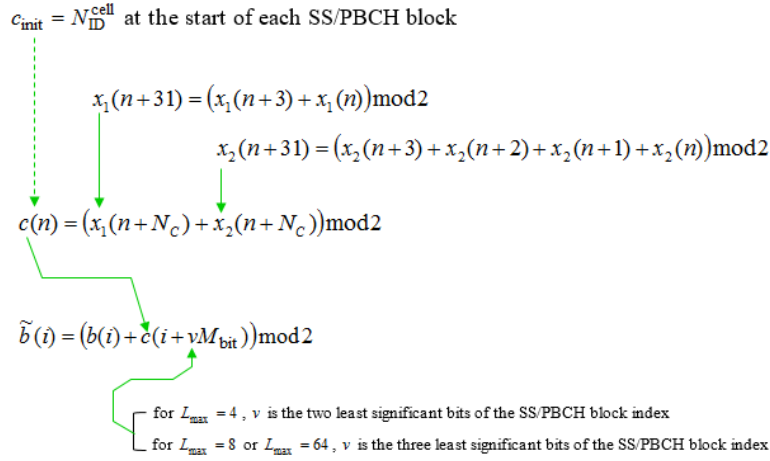
end if

end if

+ Bit interleaving

- Chuỗi bits $e_k, k = 2, 1, 0, \dots, E - 1$ được interleaved thành chuỗi bits $f_0, f_1, f_2, \dots, f_{E-1}$
- Giá trị của E không qus 8192

(6) Scrambling



- L_{max} là số khối SS/PBCH tối đa trong một chu kỳ SS/PBCH. Giá trị này được xác định bởi khoảng cách sóng mang con và dải tần số.

- Nó bắt đầu bằng việc khởi tạo chuỗi c_{init} dựa trên cell N_{ID}^{cell} ở đầu mỗi khối SS/PBCH. Hai chuỗi, $x_1(n)$ và $x_2(n)$, được sử dụng để tạo ra chuỗi xáo trộn $c(n)$. Các chuỗi này được cập nhật cho mỗi n bằng cách sử dụng thanh ghi dịch chuyển phản hồi tuyến tính, với các cập nhật được thực hiện như sau:

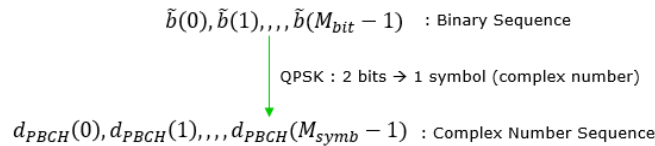
- $x_1(n+31) = (x_1(n+3) + x_1(n)) \bmod 2$
- $x_2(n+31) = (x_2(n+3) + x_2(n+2) + x_2(n+1) + x_2(n)) \bmod 2$

- Bits dữ liệu $\tilde{b}(i)$ được xáo trộn bằng XOR với các bit từ chuỗi $c(i + vM_{bit})$, với M_{bit} là một biến phụ thuộc vào độ dài L_{max} của chỉ số khối SS/PBCH:

- Với $L_{max} = 4$, v là hai bit có trọng số nhỏ nhất của chỉ số khối SS/PBCH.
- Với $L_{max} = 8$ hoặc $L_{max} = 64$, v là ba bit có trọng số nhỏ nhất của chỉ số khối SS/PBCH.

Bước này rất quan trọng để đảm bảo rằng tín hiệu truyền đi có khả năng chống nhiễu tốt, cho phép bộ thu giải mã chính xác.

(7) Modulation



- Quá trình bắt đầu với một chuỗi nhị phân $\tilde{b}(0), \tilde{b}(1), \dots, \tilde{b}(M_{bit} - 1)$. Chuỗi nhị phân này sau đó được điều chế bằng QPSK, ánh xạ mỗi hai bit của chuỗi nhị phân thành một ký tự số phức, từ đó tạo ra chuỗi số phức $d_{PBCH}(0), d_{PBCH}(1), \dots, d_{PBCH}(M_{symb} - 1)$.

- QPSK là một loại điều chế đặc biệt hiệu quả về mặt sử dụng phổ vì nó truyền hai bit trên mỗi ký tự, được biểu diễn dưới dạng số phức. Sau khi điều chế, dữ liệu được biểu diễn ở định dạng phù hợp để truyền qua kênh tần số vô tuyến, với chuỗi hiện được chuẩn bị để ánh xạ lên các thành phần tài nguyên của tần số sóng mang.

(8) Resource Element Mapping

+ Ánh xạ PBCH và DM-RS trong khối SS/PBCH

- Chuỗi ký tự $d_{PBCH}(0), \dots, d_{PBCH}(M_{symb}-1)$ cấu thành kênh phát sóng vật lý được chia tỷ lệ theo hệ số β_{PBCH} để phù hợp với phân bố công suất PBCH và được ánh xạ theo trình tự bắt đầu bằng $d_{PBCH}(0)$ tới các phần tử tài nguyên $(k, l)_{p,u}$, đáp ứng tiêu chuẩn: chúng không được sử dụng cho tín hiệu tham chiếu giải điều chế PBCH
- Ánh xạ tới các phần tử tài nguyên $(k, l)_{p,u}$ không dành riêng cho PBCH DM-RS sẽ theo thứ tự tăng dần của chỉ mục đầu tiên k và sau đó là chỉ số l , trong đó k và l lần lượt biểu thị các chỉ số tần số và thời gian trong một SS/PBCH khối và được cho trong Bảng 7.4.3.1-1.

- Chuỗi các ký tự có giá trị phức $r(0), \dots, r(143)$ tạo tín hiệu thành tham chiếu giải điều chế cho khối SS/PBCH được chia tỷ lệ theo hệ số β_{PBCH}^{DM-RS} để phù hợp với phân bổ công suất PBCH và được ánh xạ tới các phần tử tài nguyên $(k, l)_{p,u}$ theo thứ tự tăng dần đầu tiên là k và sau đó là l trong đó k và l được đưa ra trong Bảng 7.4.3.1-1 và biểu thị các chỉ số tần số và thời gian tương ứng trong một khối SS/PBCH.

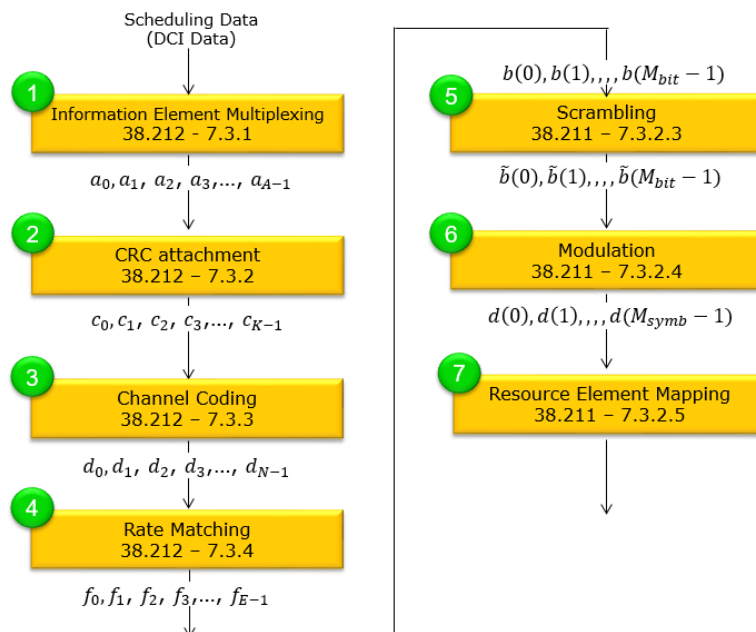
5. Kênh PDCCH - Physical downlink control channel

5.1. Tổng quan kênh PDCCH (Physical downlink control channel)

- Giống như trong LTE, NR PDCCH là kênh vật lý mang DCI và đây sẽ là một trong những kênh quan trọng nhất được cho là rất mạnh mẽ và dễ dàng giải mã ngay cả trong điều kiện vô tuyến
- Quy trình mã hóa kênh và lớp vật lý tổng thể cũng tương tự như LTE PDCCH như được liệt kê bên dưới với một số điểm khác biệt như mã hóa kênh.
- PDCCH Transport Process: (1) Information Element Multiplexing \Rightarrow (2) CRC Attachment \Rightarrow (3) Channel Coding \Rightarrow (4) Rate Matching \Rightarrow (5) Scrambling \Rightarrow (6) Modulation \Rightarrow (7) Resource Element Mapping

5.2. Cấu trúc kênh PDCCH và thuật toán của các module trong kênh PDCCH

- Sơ đồ sau minh họa Quy trình truyền tải PDCCH trong hệ thống thông tin di động, trình tự các bước liên quan đến việc chuẩn bị và truyền PDCCH từ gNB đến UE



- Information Element Multiplexing: Giai đoạn này bao gồm việc kết hợp các yếu tố thông tin điều khiển khác nhau như định dạng Downlink Control Information (DCI) thành một luồng dữ liệu duy nhất để xử lý.
- CRC Attachment: Một Cyclic Redundancy Check được thêm vào thông tin ghép kênh để cho phép phát hiện lỗi ở đầu thu.
- Channel Coding: Luồng dữ liệu có CRC được mã hóa để bảo vệ khỏi nguy cơ hỏng dữ liệu trong quá trình truyền. Điều này thường liên quan đến các kỹ thuật mã hóa sửa lỗi như mã Polar.
- Rate Matching: Dữ liệu được mã hóa được điều chỉnh để phù hợp với các khối tài nguyên được phân bổ. Bước này có thể liên quan đến việc lặp lại các bit để khớp dữ liệu vào không gian truyền tải có sẵn.
- Scrambling: Rate-matched data được xáo trộn để giảm thiểu nhiễu và đảm bảo an toàn dữ liệu. Việc xáo trộn được thực hiện bằng cách sử dụng một trình tự được cả máy phát và máy thu biết.

- Modulation: Sau đó, dữ liệu được mã hóa sẽ được điều chế, nghĩa là chuyển đổi các bit thành các ký tự có thể truyền qua kênh tần số vô tuyến..
- Resource Element Mapping: Cuối cùng, các ký tự đã điều chế được ánh xạ lên các phần tử tài nguyên cụ thể trong lưới tần số-thời gian của tín hiệu sóng mang. Bước này định vị thông tin điều khiển một cách chính xác để truyền trong khung tín hiệu rộng hơn.

(1) Information Element Multiplexing

- Đây là một quá trình tạo ra một chuỗi bit DCI mang nhiều thông tin điều khiển và lập lịch khác nhau.

+ DCI (Downlink Control Information) có một số chức năng chính như sau:

- Mang thông tin lịch lịch (phân bổ tài nguyên vật lý) cho PDSCH
- Mang thông tin lịch lịch (phân bổ tài nguyên vật lý) cho PUSCH
- Mang thông tin điều chỉnh Uplink Power (PUSCH, PUCCH power) để điều khiển công suất

+ Mục đích chính của DCI: một tập hợp thông tin đặc biệt lập lịch cho kênh dữ liệu đường xuống (ví dụ: PDSCH) hoặc kênh dữ liệu đường lên (ví dụ: PUSCH).

+ Sau đây là danh sách các loại DCI và nội dung của nó được xác định trong 38.212 - 7.3 DCI

Table 7.3.1-1: DCI formats

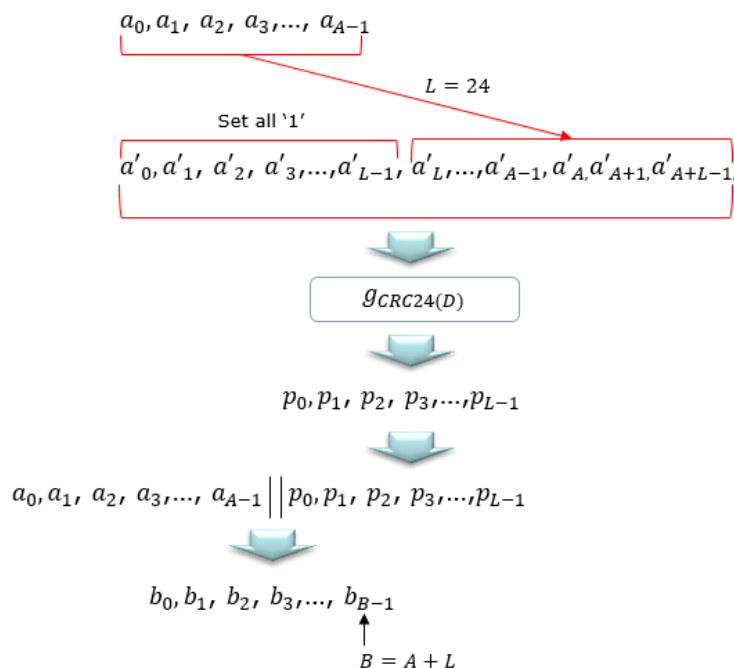
DCI format	Usage
0_0	Scheduling of PUSCH in one cell
0_1	Scheduling of one or multiple PUSCH in one cell, or indicating downlink feedback information for configured grant PUSCH (CG-DFI)
0_2	Scheduling of PUSCH in one cell
1_0	Scheduling of PDSCH in one cell
1_1	Scheduling of PDSCH in one cell, and/or triggering one shot HARQ-ACK codebook feedback
1_2	Scheduling of PDSCH in one cell
2_0	Notifying a group of UEs of the slot format, available RB sets, COT duration and search space set group switching
2_1	Notifying a group of UEs of the PRB(s) and OFDM symbol(s) where UE may assume no transmission is intended for the UE
2_2	Transmission of TPC commands for PUCCH and PUSCH
2_3	Transmission of a group of TPC commands for SRS transmissions by one or more UEs
2_4	Notifying a group of UEs of the PRB(s) and OFDM symbol(s) where UE cancels the corresponding UL transmission from the UE
2_5	Notifying the availability of soft resources as defined in Clause 9.3.1 of [10, TS 38.473]
2_6	Notifying the power saving information outside DRX Active Time for one or more UEs
3_0	Scheduling of NR sidelink in one cell
3_1	Scheduling of LTE sidelink in one cell

+ Khi dữ liệu DCI được xây dựng, nó sẽ chuyển sang quy trình mã hóa kênh và truyền qua PDCCH.

- Nếu kích thước của DCI nhỏ hơn 12 bit, nó sẽ được đệm bằng 0 cho đến khi trở thành 12 bit

(2) CRC Attachment

- < CRC Attach >: CRC 24 bit được gắn vào dữ liệu DCI.



+ Ban đầu, dữ liệu DCI được trình bày dưới dạng chuỗi các bit $a_0, a_1, a_2, \dots, a_{A-1}$. CRC 24 bit được thêm vào dữ liệu DCI này.

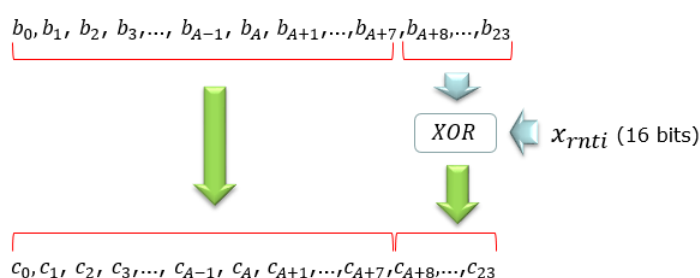
+ Để chuẩn bị cho việc đính kèm CRC, một chuỗi được mở rộng bằng cách đặt tất cả các bit theo sau dữ liệu DCI thành '1' với độ dài L là 24, dẫn đến chuỗi mở rộng $a'_0, a'_1, a'_2, \dots, a'_{L-1}, a'_L, \dots, a'_{A+L-1}$. Chuỗi này được xử lý bằng cách sử dụng đa thức $G_{CRC24}(D)$, tạo ra CRC gồm các bit $p_0, p_1, p_2, \dots, p_{L-1}$.

+ Dữ liệu DCI và CRC sau đó được nối với nhau, tạo thành một chuỗi hoàn chỉnh $a_0, a_1, a_2, \dots, a_{A-1} || p_0, p_1, p_2, \dots, p_{L-1}$, được ký hiệu là $b_0, b_1, b_2, \dots, b_{B-1}$. Tổng độ dài của chuỗi sau khi đính kèm CRC là $B = A + L$.

+ Quá trình đính kèm CRC này là một bước quan trọng trong việc đảm bảo tính toàn vẹn của thông tin điều khiển vì nó cho phép người nhận phát hiện bất kỳ lỗi nào có thể xảy ra trong quá trình truyền.

- < *Masking with RNTI* >

+ Sau khi đính kèm CRC, 16 bit cuối cùng được mask bằng RNTI cụ thể. Bằng cách sử dụng RNTI này, UE sẽ tìm ra DCI dành cho UE nào và mục đích sử dụng DCI là gì.



+ Sau khi CRC được thêm vào dữ liệu DCI, tạo thành chuỗi $b_0, b_1, b_2, \dots, b_{A-1}, b_A, \dots, b_{23}$, 16 bit cuối cùng được mask bằng một RNTI cụ thể (Radio Network Temporary Identifier).

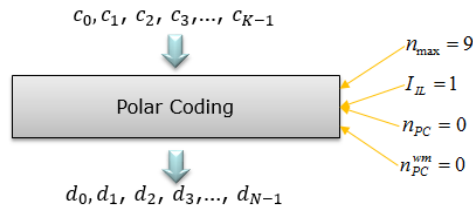
+ Việc mask này được thực hiện bằng cách XOR 16 bit cuối cùng của chuỗi với RNTI 16 bit, ký hiệu là x_{RNTI} . RNTI là mã định danh duy nhất cho phép Thiết bị người dùng (UE) xác định DCI nào dành cho thiết bị đó và hiểu cách sử dụng DCI.

+ Đầu ra của quá trình này là một chuỗi mới $c_0, c_1, c_2, \dots, c_{A-1}, c_A, \dots, c_{A+7}, c_{A+8}, \dots, c_{23}$, trong đó 16 bit cuối cùng đã được sửa đổi bởi RNTI masking. Bước này rất quan trọng để hướng DCI đến đúng UE và để bảo vệ tính toàn vẹn của thông tin.

- < *Interleaving* >

+ Sau khi RNTI masking, dữ liệu được xen kẽ để các bit CRC được phân bổ giữa các bit thông tin. Bộ xen kẽ này hỗ trợ kích thước đầu vào tối đa là 164 bit, nghĩa là DCI không có CRC có thể có kích thước tối đa là 140 bit.

(3) Channel Coding



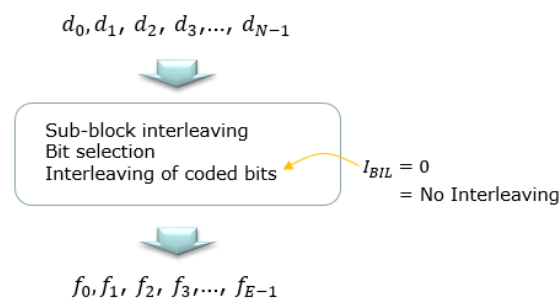
- Ở bước này, chuỗi $c_0, c_1, c_2, \dots, c_{K-1}$ trải qua Mã hóa cực, một phương pháp được biết đến với tính hiệu quả trong khả năng sửa lỗi.

- Quá trình Mã hóa Cực được đặc trưng bởi một số tham số: n_{\max} đại diện cho số lượng chỉ số chuỗi đáng tin cậy tối đa và được đặt thành 9; I_{IL} biểu thị kích thước danh sách được sử dụng trong giải mã, được cho là 1; n_{PC} là số bit kiểm tra tính chẵn lẻ, trong trường hợp này là 0; và n_{PC}^{wm} là một tham số kiểm tra chẵn lẻ khác, cũng được đặt thành 0.

- Sau khi Polar Coding, chuỗi dữ liệu được chuyển đổi thành $d_0, d_1, d_2, \dots, d_{N-1}$, sẵn sàng cho các bước truyền tiếp theo. Chuỗi mã hóa này có khả năng chống lỗi cao hơn trong quá trình truyền qua kênh liên lạc.

(4) Rate Matching

- Sơ đồ này minh họa bước Rate Matching trong xử lý kênh PDCCH, bước này rất cần thiết để điều chỉnh dữ liệu được mã hóa theo kích thước chính xác để truyền.

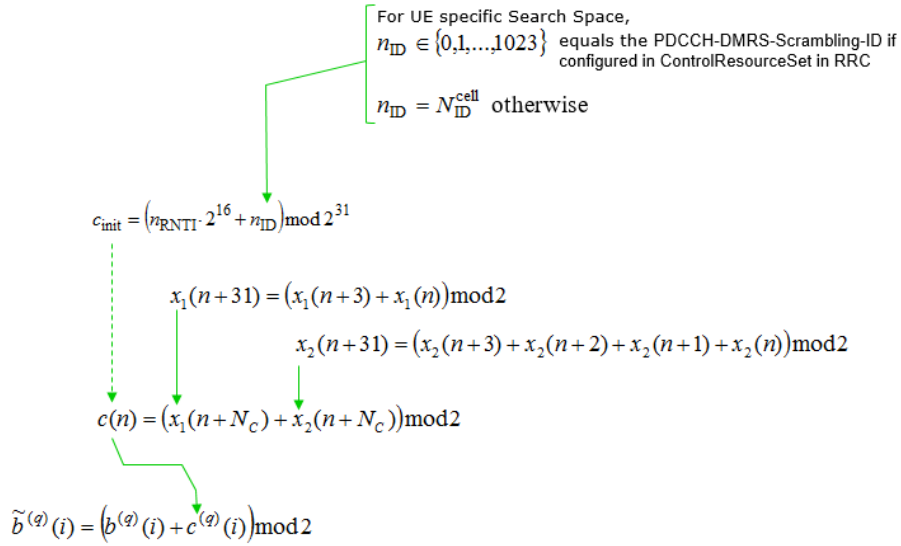


- Bắt đầu với chuỗi dữ liệu được mã hóa $d_0, d_1, d_2, \dots, d_{N-1}$, quá trình này bao gồm việc xen kẽ khối con - sub-block interleaving và lựa chọn bit - bit selection. Việc xen kẽ các bit được mã hóa được biểu thị, nhưng với I_{BIL} được đặt thành 0 (không có sự xen kẽ nào được áp dụng cho dữ liệu)

- Sau đó, chuỗi dữ liệu được điều chỉnh để phù hợp với tốc độ truyền dẫn, tạo thành chuỗi mới $f_0, f_1, f_2, \dots, f_{E-1}$ (E là độ dài chuỗi đầu ra rate matching) Quá trình rate matching này đảm bảo rằng dữ liệu phù hợp với tài nguyên truyền được phân bổ và được Thiết bị người dùng nhận và giải mã chính xác.

(5) Scrambling

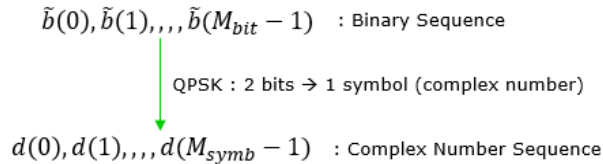
- Hình dưới minh họa quá trình scrambling, để đảm bảo tính toàn vẹn và bảo mật dữ liệu.



- Quá trình xáo trộn sử dụng chuỗi khởi tạo c_{init} có nguồn gốc từ RNTI (Radio Network Temporary Identifier) và một identifier n_{ID} .
- Với UE specific search spaces, n_{ID} được đặt thành giá trị được xác định bởi *PDCCH-DMRS-Scrambling-ID* nếu được định cấu hình trong ControlResourceSet trong lớp RRC (Radio Resource Control), giá trị này có thể nằm trong khoảng từ 0 đến 1023. Nếu không, $n_{ID} = N_{ID}^{cell}$ (cell ID vật lý)
- Chuỗi khởi tạo được tính $c_{init} = (RNTI \cdot 2^{16} + n_{ID}) \bmod 2^{31}$. Hai chuỗi $x_1(n)$ và $x_2(n)$ ược tạo ra và sử dụng để tạo ra chuỗi xáo trộn $c(n)$ thông qua mối quan hệ đa thức xác định.
- Cuối cùng, các bit dữ liệu $b(i)$ được xáo trộn với $c(n)$ để tạo ra các bit được xáo trộn $\tilde{b}(i) = (b(i) + c(i)) \bmod 2$. Chuỗi xáo trộn này có khả năng chống nhiễu tốt hơn và chặn trái phép, tăng cường tính bảo mật và độ tin cậy tổng thể của thông tin liên lạc.

(6) PDCCH Modulation

Hình sau mô tả quá trình điều chế cho PDCCH.



- Quá trình bắt đầu với một chuỗi nhị phân được biểu diễn dưới dạng $\tilde{b}(0), \tilde{b}(1), \dots, \tilde{b}(M_{bit} - 1)$. Chuỗi này trải qua quá trình điều chế QPSK (phương pháp ánh xạ mỗi hai bit của chuỗi nhị phân thành một ký hiệu, được biểu thị dưới dạng số phức). Do đó, hai bit nhị phân được chuyển đổi thành một ký hiệu QPSK duy nhất.
- Sau khi điều chế, chuỗi bit nhị phân được chuyển thành chuỗi số phức, $d(0), d(1), \dots, d(M_{symb} - 1)$, , trong đó mỗi ký tự trong chuỗi mới này tương ứng với một cặp các bit nhị phân từ chuỗi ban đầu. Kết quả của việc điều chế này là một chuỗi số phức phù hợp để truyền qua phổ tần số vô tuyến.

(7) Resource Element Mapping

- Đặt các bit PDCCH được mã hóa và điều chế vào từng thành phần tài nguyên trong lưới tài nguyên NR.
- gNB giả sử một khối gồm các ký tự có giá trị phức $d(0), \dots, d(M_{symb} - 1)$ sẽ được chia tỷ lệ theo hệ số β_{PDCCH} và sau đó ánh xạ tới các phần tử tài nguyên $(k, l)_{p,u}$ được chỉ định cho PDCCH được giám sát và không được sử dụng cho DMRS PDCCH liên quan. Thứ tự ánh xạ phải tuân theo trình tự tăng dần đầu tiên là k , sau đó là l . Cổng ăng ten được chỉ định cho hoạt động này là $p = 2000$.
- Quá trình này có thể được mô tả từng bước như sau:

- gNB chuẩn bị một khối các ký hiệu có giá trị phức được biểu diễn dưới dạng $d(0), d(1), \dots, d(M_{\text{symb}} - 1)$.
- Mỗi ký tự trong khối được chia tỷ lệ theo hệ số được xác định trước β_{PDCCH} .
- Các ký tự tỷ lệ được ánh xạ lên các phần tử tài nguyên, ký hiệu là $(k, l)_{\text{p,u}}$, với k là chỉ số sóng mang con và l là chỉ số ký hiệu OFDM trong một slot
- Ánh xạ này tuân theo thứ tự tăng dần, đầu tiên là theo chỉ số sóng mang con k , sau đó theo chỉ số ký tự hiệu OFDM l .
- Đảm bảo rằng các thành phần tài nguyên được sử dụng cho PDCCH không trùng lặp với các thành phần tài nguyên được phân bổ cho PDCCH DMRS liên quan.
- Công suất được sử dụng cho hoạt động này được chỉ định là $p = 2000$.

Thông qua các bước này, gNB gán một cách có hệ thống các ký hiệu PDCCH cho các tài nguyên thích hợp trên lưới tần số-thời gian để truyền hiệu quả.

6. Kênh PDSCH - Physical downlink shared channel

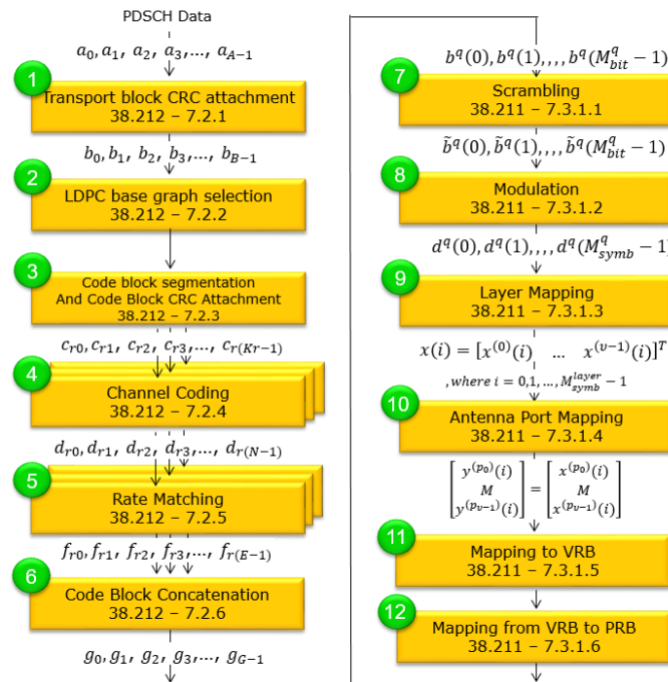
6.1. Tổng quan kênh PDSCH (Physical Data Shared Channel)

- PDSCH là kênh vật lý đường xuống cung cấp dữ liệu người dùng từ gNB đến UE.
- + Lưới tài nguyên: PDSCH chiếm một lưới tần số thời gian của các Khối tài nguyên (RB) trong một slot. Số lượng ký tự PDSCH OFDM tối đa trong một slot là 14, nhưng trong hầu hết trường hợp, nó sử dụng ít hơn 14 vì thông thường một hoặc nhiều ký tự OFDM được sử dụng cho PDCCH.
- + Scheduling: gNB lập lịch trình tài nguyên PDSCH cho các UE dựa trên chất lượng kênh, yêu cầu dữ liệu và cân nhắc về tính công bằng.
- + Điều chế: PDSCH hỗ trợ các sơ đồ điều chế khác nhau như QPSK, 16QAM, 64QAM và 256QAM để đáp ứng các điều kiện kênh và tốc độ dữ liệu khác nhau.
- + Mã hóa kênh: PDSCH sử dụng mã hóa LDPC (Low-Density Parity-Check) để cung cấp khả năng sửa lỗi chuyển tiếp, nâng cao độ tin cậy của việc truyền dữ liệu qua kênh không dây.
- + Layer Mapping: Trong hệ thống MIMO, PDSCH được phân bổ giữa nhiều lớp.
- + Rate Matching: PDSCH sử dụng Rate Matching để điều chỉnh tốc độ dữ liệu được mã hóa cho phù hợp với tài nguyên được phân bổ, đáp ứng các yêu cầu UE và điều kiện kênh khác nhau.
- + DMRS: PDSCH bao gồm DMRS để hỗ trợ UE trong việc ước tính và giải điều chế kênh, đảm bảo thu dữ liệu chính xác

- Quá trình chuyển đổi dữ liệu người dùng thành dữ liệu PDSCH và truyền nó qua từngăng-ten truyền dẫn. Những yếu tố tham gia vào quá trình này. Phần quan trọng (cốt lõi) là Quá trình vận chuyển và DCI và RRC có nhiệm vụ cung cấp (cấu hình) một số tham số cho quá trình vận chuyển. Trong LTE, hầu hết các tham số truyền tải được cố định hoặc xác định tự động bằng thuật toán quá trình truyền tải và chỉ một số lượng nhỏ tham số được cấu hình bởi DCI nhưng bản tin RRC không ảnh hưởng nhiều đến quá trình này. Tuy nhiên, trong NR, nhiều tham số quy trình truyền tải được cung cấp (cấu hình) không chỉ bởi DCI mà còn bởi bản tin RRC => quá trình sẽ trở nên linh hoạt hơn nhưng việc khắc phục sự cố cho quy trình sẽ trở nên khó khăn hơn.

- PDSCH Transport Process: (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) Scrambling => (8) Modulation => (9) Layer Mapping => (10) Antenna Port Mapping => (11) Mapping to VRB => (12) Mapping from virtual to physical resource blocks

6.2. Cấu trúc kênh PDSCH và thuật toán của các module trong kênh PDSCH



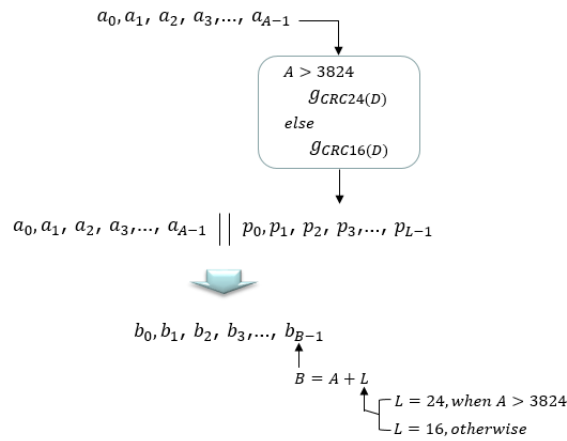
- **Transport Block CRC Attachment:** PDSCH data $a_0, a_1, a_2, \dots, a_{A-1}$ đi qua một CRC attachment để phát hiện lỗi ở phía máy thu.
- **LDPC Base Graph Selection:** Sau khi đính kèm CRC, dữ liệu được xử lý thông qua mã hóa LDPC (Low-Density Parity-Check) nhằm mục đích sửa lỗi.
- **Code Block Segmentation và CRC Attachment:** Dữ liệu sau đó được phân chia thành các khối mã nhỏ hơn và một CRC khác được gắn vào mỗi khối.
- **Channel Coding:** Các khối mã được mã hóa kênh để bảo vệ khỏi lỗi trong quá trình truyền.
- **Rate Matching:** Dữ liệu được mã hóa kênh được điều chỉnh để phù hợp với băng thông truyền có sẵn.
- **Code Block Concatenation:** Các khối mã rate-matched được ghép lại thành một luồng dữ liệu duy nhất.
- **Scrambling:** Dữ liệu được nối được xáo trộn để ngẫu nhiên hóa mẫu dữ liệu nhằm giảm thiểu nhiễu và bảo mật.
- **Modulation:** Các bit được xáo trộn sau đó được điều chế thành các ký tự phù hợp để truyền.
- **Layer Mapping:** Các ký tự đã điều chế được ánh xạ qua nhiều lớp để truyền MIMO.
- **Antenna Port Mapping:** Các lớp được ánh xạ tới các cổng ăng-ten cụ thể.
- **Mapping to VRB (Virtual Resource Block):** Dữ liệu được ánh xạ tới các khối tài nguyên ảo trong miền tần số.
- **Mapping from VRB to PRB (Physical Resource Block):** Các khối tài nguyên ảo sau đó được ánh xạ tới các khối tài nguyên vật lý để truyền thực tế.

Mỗi bước trong quy trình đều quan trọng để đảm bảo liên lạc hiệu quả và đáng tin cậy qua PDSCH.

(1) Transport block CRC attachment

- Việc đính kèm CRC khối truyền tải trong xử lý kênh PDSCH 5G là bước cho phép UE phát hiện lỗi trong khối truyền tải nhận được, đảm bảo truyền dữ liệu đáng tin cậy qua kênh không dây. CRC được tính toán cho khối truyền tải để cho phép phát hiện lỗi ở máy thu (UE). CRC là checksum có kích thước cố định được tạo bằng cách áp dụng hàm đa thức cho dữ liệu khối truyền tải.

- Trong 5G NR, CRC 24 bit hoặc 16 bit được gắn vào khối truyền tải tùy thuộc vào kích thước của khối truyền tải.



- Dữ liệu từ khối truyền tải, được biểu diễn dưới dạng một chuỗi bit $a_0, a_1, a_2, \dots, a_{A-1}$, được chuẩn bị cho phần đính kèm CRC để cho phép phát hiện lỗi ở đầu thu.
- Nếu kích thước của khối truyền tải A lớn hơn 3824, CRC 24-bit được gắn bằng 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 kích thước của khối truyền tải A nhỏ hơn hoặc bằng 3824 thì CRC 16-bit sẽ được sử dụng thay thế, với đa thức $G_{CRC16}(D)$.

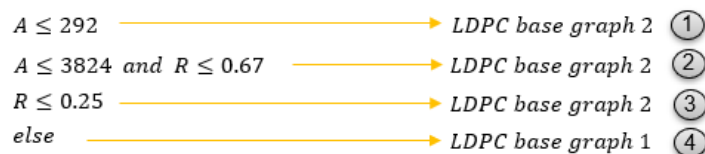
$$g_{CRC16}(D) = [D^{16} + D^{12} + D^5 + 1]$$

- CRC được tính toán và thêm vào chuỗi dữ liệu, tạo thành chuỗi mở rộng $a_0, a_1, a_2, \dots, a_{A-1} \parallel p_0, p_1, p_2, \dots, p_{L-1}$.
- Độ dài L của CRC được đặt thành 24 khi $A > 3824$ và 16 nếu không, để chứa các bit CRC.
- Chuỗi kết quả sau khi đính kèm CRC được biểu diễn dưới dạng $b_0, b_1, b_2, \dots, b_{B-1}$, với $B = A + L$, cho biết độ dài mới của chuỗi.

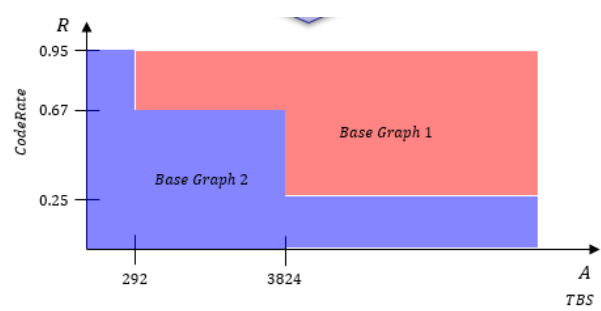
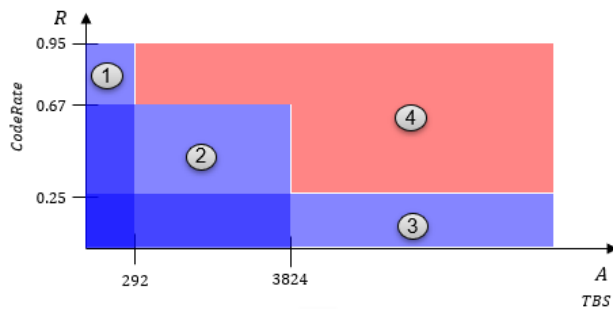
Quá trình đính kèm CRC này là cần thiết để đảm bảo truyền dữ liệu đáng tin cậy qua kênh không dây bằng cách cho phép phát hiện lỗi tại UE.

(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).

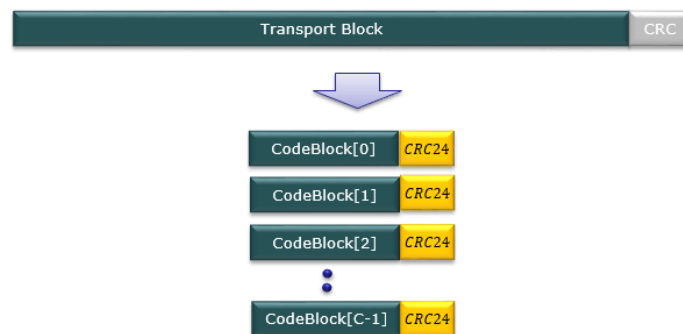
+ Các bit sau khi phân đoạn khối mã được ký hiệu là $c_{r0}, c_{r1}, c_{r2}, \dots, c_{r(K_r-1)}$, trong đó r là số khối mã và K_r là số bit của số khối mã r .

- Bước này nhằm đảm bảo việc truyền dữ liệu hiệu quả và đáng tin cậy bằng cách chia các khối truyền tải lớn thành các phân đoạn nhỏ hơn và cung cấp khả năng phát hiện lỗi ở cấp độ khối mã.

+ Code Block Segmentation: Nếu kích thước của khối truyền tải quá lớn để mã hóa LDPC hiệu quả, nó sẽ được chia thành các phân đoạn nhỏ hơn, được gọi là khối mã. Kích thước tối đa của khối mã được xác định bởi thông số kỹ thuật 5G NR. Việc phân đoạn được thực hiện để đảm bảo mã hóa và giải mã kênh hiệu quả trong khi vẫn duy trì độ phức tạp hợp lý

+ Segmentation Criteria: Quá trình phân đoạn được xác định bằng cách so sánh kích thước khối truyền tải với kích thước khối mã tối đa được chỉ định. Nếu kích thước khối truyền tải vượt quá kích thước khối mã tối đa, khối truyền tải sẽ được chia thành các khối mã có kích thước bằng nhau (ngoại trừ khối mã cuối cùng, có thể nhỏ hơn). Nếu kích thước khối truyền tải nằm trong kích thước khối mã tối đa thì không có phân đoạn nào được thực hiện..

+ Code Block CRC Attachment: Sau khi phân đoạn, CRC (Cyclic Redundancy Chec) được tính toán và gắn vào từng khối mã riêng lẻ. CRC 24 bit này cho phép phát hiện lỗi ở máy thu (UE) trên cơ sở từng khối mã.



i) Xác định kích thước tối đa của khối mã (K_{cb}) phụ thuộc vào loại LDPC base graph.

- o Với LDPC base graph type 1 : $K_{cb} = 8448$
- o Với LDPC base graph type 2 : $K_{cb} = 3840$

ii) Xác định số lượng Codeblocks

if $B(\text{Transport block size}) < K_{cb}(\text{Max Codeblock size})$

$L = 0$

C (number of codeblocks) = 1

$B' = B$ // this mean 'No Segmentation'.

else

$L = 24$

$C = \text{Ceiling}(B/(K_{cb} - L))$

$B' = B + C * L$

iii) Xác định số bit trong mỗi khối mã

$$K'(\text{số bit trong mỗi khối mã}) = B'/C$$

iv) Xác định K_b

Với LDPC base graph type 1: $K_b = 22$

Với LDPC base graph type 2

if B (Transport blocksize) $> 640 \Rightarrow K_b = 10$

else if B (Transport blocksize) $> 560 \Rightarrow K_b = 9$

else if B (Transport blocksize) $> 192 \Rightarrow K_b = 8$

else $\Rightarrow K_b = 6$

v) tìm giá trị nhỏ nhất của Z trong tất cả các bộ kích thước nâng trong

vi) ký hiệu Z_c sao cho $(K_b * Z_c) \geq K'$

vii) đặt $K = 22 Z_c$ với LDPC base graph 1

$K = 10 Z_c$ với LDPC base graph 2

viii) thực hiện phân đoạn và thêm CRC bits

$s = 0$ // s = bit position in B (transport block)

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

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

$c_{rk} = b_s$

$s = s + 1$

end for

if $C > 1$ // Do this if the number of the code block is more than 1

Calculate $pr_0, pr_1, pr_2, \dots, pr(L-1)$ using the sequence $c_{r0}, c_{r1}, c_{r2}, \dots, c_{r(K'-L-1)}$ and $g_{CRC24B}(D)$

for $k = K'-L$ to $K'-1$ // Append CRC bit

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

end for

end if

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

$c_{rk} = \text{NULL}$

end for

end for

Tóm tắt của quy trình được liệt kê ở trên trong mã giả như sau:

- Xác định kích thước tối đa của khối mã (K_{cb}), kích thước này thay đổi tùy thuộc vào loại LDPC base graph được sử dụng.
- Quyết định phân đoạn dựa trên việc kích thước khối truyền tải có vượt quá K_{cb} hay không, tính toán số khối mã (C) và kích thước của mỗi khối (B').
- Tính số bit trong mỗi khối mã bằng cách chia kích thước khối truyền tải cho C .
- Xác định K_b , là số bit trong mỗi khối mã, dựa trên loại LDPC base graph và kích thước khối truyền tải.
- Tìm giá trị nhỏ nhất của Z từ tập kích thước nâng LDPC thỏa mãn điều kiện $K_b \leq Z_c$.
- Đối với mỗi khối mã, nếu $C > 1$, tính toán các bit CRC và nối chúng vào khối mã, chèn các bit phụ nếu cần.

(4) Channel Coding

- Khối mã được phân phối đến khối mã hóa kênh. Các bit trong 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 trong 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 gắn vào các bit dữ liệu gốc, tạo thành từ mã được truyền qua PDSCH.

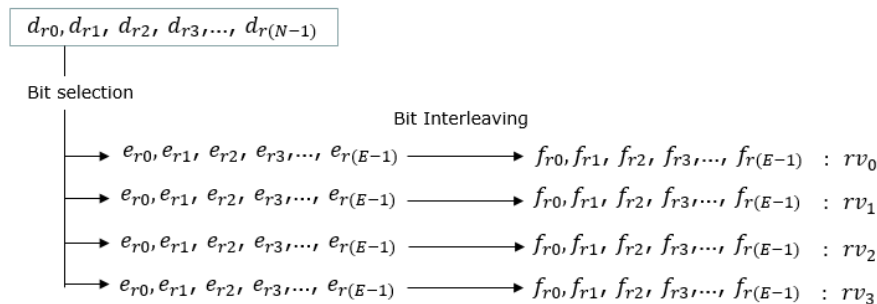
(5) Rate Matching

- 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 $I_{LBRM}=1$.

- 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:

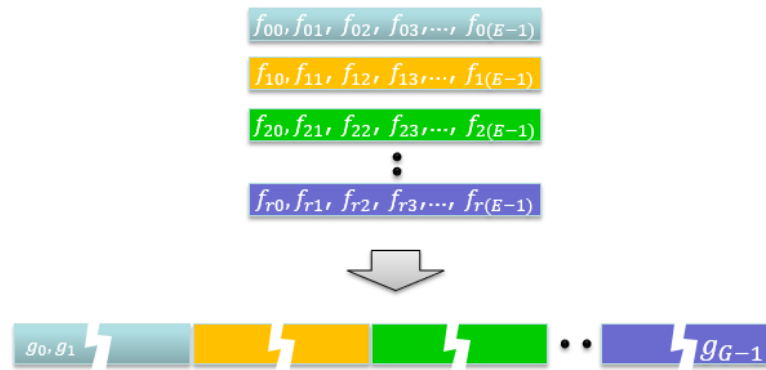
- Bit Collection:: Sau khi mã hóa LDPC, các bit được mã hóa (bit dữ liệu và bit chẵn lẻ) được thu thập trong bộ đệm tròn-circular buffer. Bộ đệm tròn là vùng lưu trữ tạm thời với kích thước cố định có thể giữ các bit theo kiểu vòng tròn, cho phép lựa chọn bit hiệu quả.
- Bit Selection:: Tùy thuộc vào 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

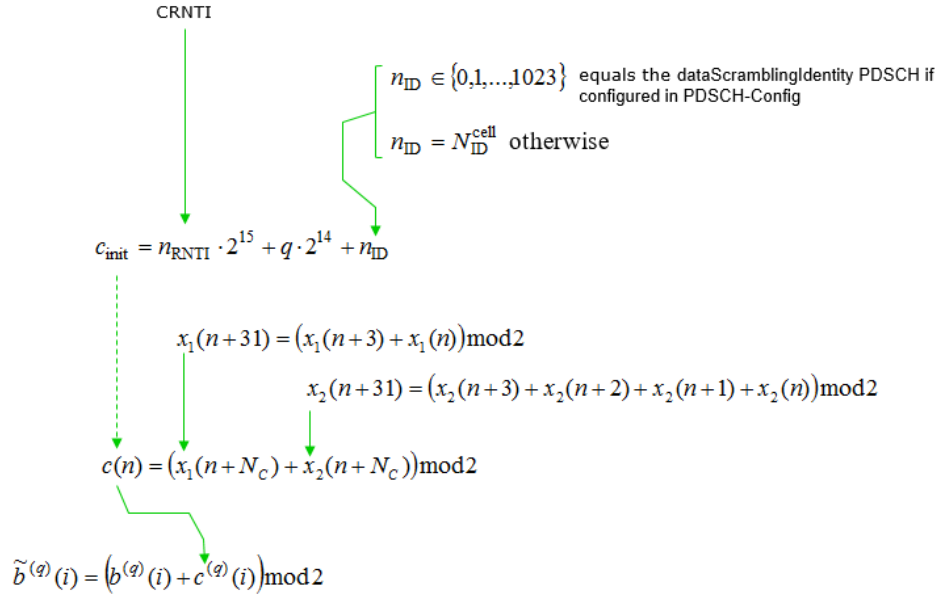
(7) Scrambling

- 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. Các hoạt động xáo trộn và giải xáo trộn được thực hiện tương ứng ở máy phát và máy thu, sử dụng cùng một trình tự xáo trộn dành riêng cho từng ô.

- Việc xáo trộn mang lại tính ngẫu nhiên cho dữ liệu được truyền bằng cách áp dụng chuỗi nhị phân giả ngẫu nhiên - pseudo-random binary sequence (PRBS) cho luồng dữ liệu.

- 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. Việc xáo trộn cũng hỗ trợ giảm thiểu nhiễu giữa các ô, cải thiện tính riêng tư của dữ liệu và cho phép máy thu (UE) thực hiện ước tính kênh chính xác

- Scrambling Sequence: Quá trình xáo trộn sử dụng trình tự xáo trộn dành riêng cho ô được tạo dựa trên ID ô và nhận dạng xáo trộn. Nhận dạng xáo trộn có thể là duy nhất cho mỗi người dùng (UE) trong một ô, đảm bảo rằng các chuỗi xáo trộn được sử dụng bởi các UE khác nhau là trực giao với nhau.
- Bitwise XOR: Quá trình xáo trộn bao gồm bitwise exclusive-or (XOR) operation giữa luồng dữ liệu đầu vào (kết quả từ bước nối khối mã) và chuỗi xáo trộn. Đầu ra là một luồng dữ liệu được xáo trộn.
- Tác động đến tốc độ dữ liệu: Việc xáo trộn không làm thay đổi tốc độ dữ liệu vì nó chỉ sửa đổi luồng dữ liệu bằng cách đưa ra tính ngẫu nhiên. Tốc độ dữ liệu được xác định bởi các bước xử lý kênh khác, chẳng hạn như mã hóa LDPC và rate matching.



- Scrambling identity n_{ID} được xác định dựa trên cấu hình. Nếu *dataScramblingIdentity* được cấu hình trên *PDSCH-Config*, n_{ID} nhận giá trị trong khoảng $\{0,1,...,1023\}$. Nếu không thì, $n_{ID} = N_{ID}^{cell}$.
- Chuỗi khởi tạo c_{init} được tính bằng công thức $c_{init} = n_{RNTI} \cdot 2^{15} + q \cdot 2^{14} + n_{ID}$, với q là 1 quarter index.
- Hai chuỗi $x_1(n)$ và $x_2(n)$ được tạo ra, với mỗi bit trong chuỗi được cập nhật bằng cách sử dụng các đa thức đã cho và số học modulo 2.
- Chuỗi xáo trộn $c(n) = (x_1(n + N_c) + x_2(n + N_c)) \bmod 2$.
- Cuối cùng, mỗi bit $b(i)$ của chuỗi dữ liệu được xáo trộn với bit tương ứng $c(i)$ từ chuỗi xáo trộn để tạo ra bit được mã hóa $\tilde{b}(i)$.
 - Có thể truyền tối đa hai từ mã $q \in \{1,0\}$. Trong trường hợp truyền từ mã đơn, $q = 0$.
 - 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 $\tilde{b}^{(q)}(0), \dots, \tilde{b}^{(q)}(M_{bit}^{(q)} - 1)$ với $\tilde{b}^{(q)}(i) = (b^{(q)}(i) + c^{(q)}(i)) \bmod 2$

(8) Modulation

- Đối với mỗi từ mã, UE sẽ giả sử khối bit được xáo trộn $\tilde{b}^{(q)}(0), \dots, \tilde{b}^{(q)}(M_{bit}^{(q)} - 1)$ được điều chế bằng cách sử dụng một trong các sơ đồ điều chế trong Bảng 7.3.1.2-1, dẫn đến một khối ký tự điều chế có giá trị phức $d^{(q)}(0), \dots, d^{(q)}(M_{bit}^{(q)} - 1)$.

- Việc lựa chọn sơ đồ điều chế phụ thuộc vào các yếu tố như điều kiện kênh, khả năng thích ứng của liên kết và khả năng của UE..

$\tilde{b}^{(q)}(0), \dots, \tilde{b}^{(q)}(M_{bit}^{(q)} - 1)$: Binary Sequence

38.211-Table 7.3.1.2-1

Modulation scheme	Modulation order Q_m
QPSK	2
16QAM	4
64QAM	6
256QAM	8

: 2 bits \rightarrow 1 symbol (complex number)

: 4 bits \rightarrow 1 symbol (complex number)

: 6 bits \rightarrow 1 symbol (complex number)

: 8 bits \rightarrow 1 symbol (complex number)

$d^{(q)}(0), \dots, d^{(q)}(M_{synd}^{(q)} - 1)$: Complex Number Sequence

(9) Layer Mapping

- Layer mapping là bước phân phối các ký tự đã điều chế trên một hoặc nhiều lớp để truyền bằng nhiều ăng ten.
- Các ký tự điều chế có giá trị phức tạp cho mỗi từ mã được truyền đi được ánh xạ trên một hoặc nhiều lớp theo Bảng 7.3.1.3-1. Các ký hiệu điều chế có giá trị phức $d^{(q)}(0), \dots, d^{(q)}(M_{\text{bit}}^{(q)} - 1)$ đối với từ mã 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
↓
 $d^{(q)}(0), \dots, d^{(q)}(M_{\text{syimb}}^{(q)} - 1)$

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

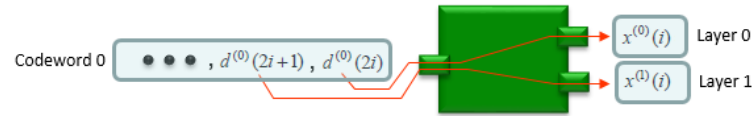
38.211-Table 7.3.1.3-1

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

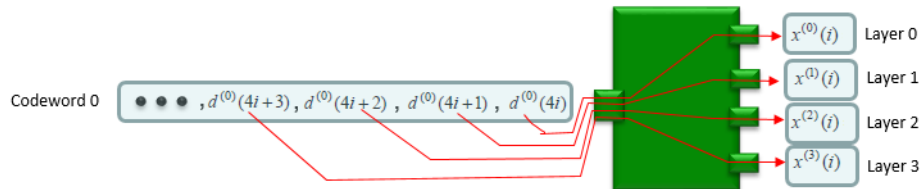
< Table 7.3.1.3-1: Codeword-to-layer mapping for spatial multiplexing. >

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

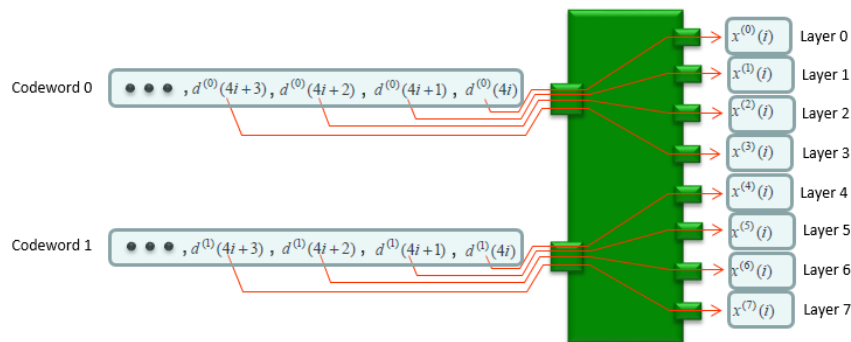
Number of layers	Number of codewords	Codeword-to-layer mapping $i = 0, 1, \dots, M_{\text{symbol}}^{\text{layer}} - 1$
2	1	$x^{(0)}(i) = d^{(0)}(2i)$ $x^{(1)}(i) = d^{(0)}(2i+1)$ $M_{\text{symbol}}^{\text{layer}} = M_{\text{symbol}}^{(0)} / 2$



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



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



(10) Antenna Port Mapping

- Khối $x(i) = [x^{(0)}(i), \dots, x^{(v-1)}(i)]^T$ với $i=0, 1, \dots, M_{\text{symbol}}^{\text{layer}} - 1$ sẽ được ánh xạ tới các cổng ăng ten
- Khi đường dẫn dữ liệu thông qua quá trình layer mapping, dữ liệu từ mỗi lớp sẽ được ánh xạ tới từng Cổng ăng-ten.
- Khi không áp dụng CSI, dữ liệu sẽ ánh xạ tới cổng ăng-ten vật lý như bên dưới.

$$\begin{matrix} \text{Antenna Port Number} & & \text{Layer Number (index)} \\ \downarrow & & \downarrow \\ \begin{bmatrix} y^{(p_0)}(i) \\ \vdots \\ y^{(p_{v-1})}(i) \end{bmatrix} & = & \begin{bmatrix} x^{(0)}(i) \\ \vdots \\ x^{(v-1)}(i) \end{bmatrix} \\ M & & M \end{matrix}$$

- Khi CSI được áp dụng, dữ liệu từ trình ánh xạ lớp trước tiên được ánh xạ tới từng cổng ăng ten CSI như hiển thị bên dưới.

$$\begin{matrix} \text{CSI Antenna Port Number} & & \text{Layer Number (index)} \\ \downarrow & & \downarrow \\ \begin{bmatrix} y^{(3000)}(i) \\ \vdots \\ y^{(3000+P-1)}(i) \end{bmatrix} & = & W(i) \begin{bmatrix} x^{(0)}(i) \\ \vdots \\ x^{(v-1)}(i) \end{bmatrix} \\ M & & M \end{matrix}$$

Number of CSI-RS Ports
 $P \in [1, 2, 4, 8, 12, 16, 24, 32]$

This matrix is determined by reportQuantity in RRC CSI ReportConfig, summarized in the table shown below

- Việc xác định $W(i)$ từ reportQuantity của CSI-ReportConfig có thể được tóm tắt như dưới.

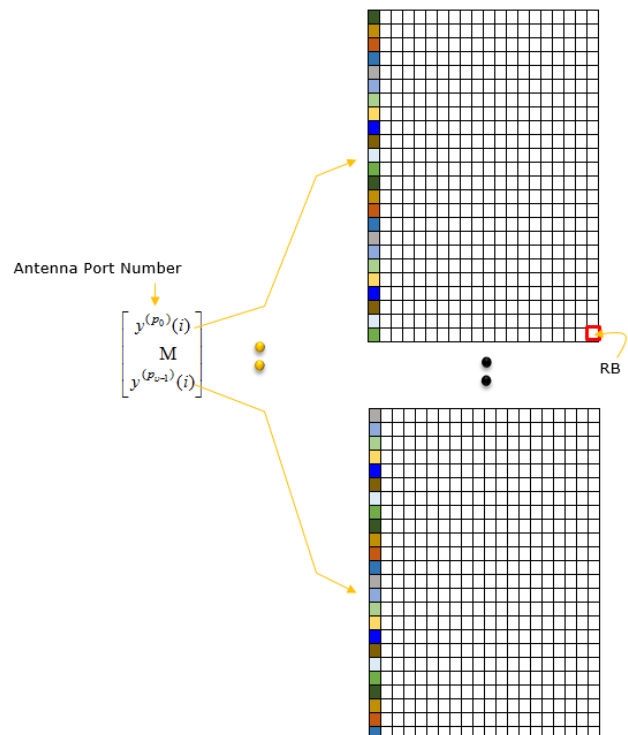
ReportQuantity	$W(i)$ determination criterial
cri-RI-PMI-CQI or cri-RI-LI-PMI-CQI	PMI Report from UE
cri-RI-CQI	The process described in 38.214-5.2.1.4.2
cri-RI-i1-CQI	i1 Report from UE

(11) Mapping to VRB

- Đối với mỗi bước ăng-ten, một lưới tài nguyên ảo sẽ được tạo. Trong lưới tài nguyên, điền vào từng thành phần tài nguyên (RE) dữ liệu PDSCH từ RE ở tần số thấp nhất đến tần số cao hơn. Khi nó đạt đến RE ở tần số cao nhất của khối tài nguyên PDSCH được chỉ định, di chuyển đến RE ở tần số thấp nhất của ký hiệu OFDM tiếp theo.

- Tuy nhiên, không nên sử dụng RE được chỉ định khi:

- RE được gán cho DMRS liên kết với PDSCH được truyền đi
- RE được chỉ định cho DMRS dành cho các UE được đồng lập lịch khác
- RE cho CSI-RS công suất khác 0, ngoại trừ CSI-RS công suất khác 0 được định cấu hình bởi tham số lớp cao hơn CSI-RS-Resource-Mobility trong MeasObjectNR IE.
- RE cho PTRS
- RE được khai báo là “không có sẵn cho PDSCH”



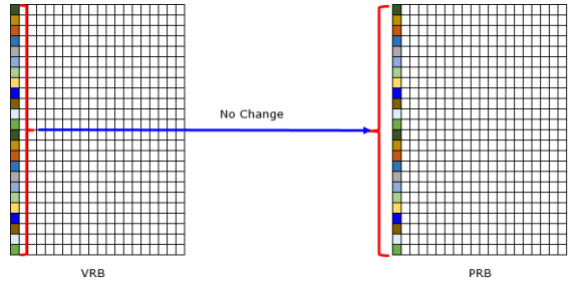
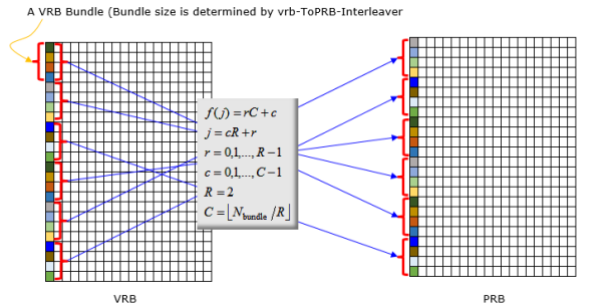
- Với mỗi cổng ăng ten được sử dụng để truyền kênh vật lý, giả sử khối ký hiệu có giá trị phức $y^{(p)}(0), \dots, y^{(p)}(M_{\text{symb}}^{\text{ap}} - 1)$ phù hợp với phân bổ công suất đường xuống và được ánh xạ theo trình

tự bắt đầu từ $y^{(p)}(0)$ 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 khối tài nguyên vật lý tương ứng được khai báo là có sẵn cho PDSCH theo mục 5.1.4 của [6, TS 38.214];
- + Các phần tử tài nguyên tương ứng trong các khối tài nguyên vật lý tương ứng
- Ánh xạ tới các phần tử tài nguyên $(k', l)_{p,u}$, được phân bổ cho PDSCH theo [6, TS 38.214] và không dành riêng cho các các mục đích khác 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 1.

(12) Mapping from virtual to physical resource blocks

- Bước này là quá trình ánh xạ (chuyển đổi) khối tài nguyên ảo (lưới tài nguyên) thành khối tài nguyên vật lý (lưới tài nguyên).
- Hai loại ánh xạ khác nhau được hỗ trợ trong 5G. Non-Interleaved Mapping và Interleaved Mapping

	Non-Interleaved Mapping	Interleaved Mapping
Định nghĩa	Các ký tự đã điều chế và mã hóa trước được phân bổ cho các khối tài nguyên vật lý một cách liên tục và tuần tự. Cách này đơn giản hóa quá trình ánh xạ nhưng có thể dẫn đến giảm phân tán tần số do các ký hiệu lân cận được truyền trên các sóng mang con liền kề.	Các ký tự đã điều chế và mã hóa trước được phân bổ cho các khối tài nguyên vật lý theo cách phân tán và không tuần tự. Cách này làm tăng tính đa dạng tần số bằng cách trải rộng các ký tự trên các tài nguyên sẵn có, cải thiện khả năng phục hồi chống lại hiện tượng nhiễu và pha đỉnh chọn lọc tần số
Advantage	Non-interleaved mapping đơn giản và dễ hiểu, giúp dễ dàng thực hiện và quản lý.	Interleaved mapping cung cấp sự đa dạng tần số tốt hơn và mạnh mẽ hơn chống lại hiện tượng pha đỉnh và nhiễu chọn lọc tần số, điều này có thể cải thiện hiệu suất tổng thể của hệ thống.
Disadvantages	Do sự phân bổ liên tục của các ký hiệu, non-interleaved mapping có thể kém đàn hồi hơn đối với sự suy giảm và nhiễu chọn lọc tần số.	Sự phức tạp ngày càng tăng của interleaved mapping có thể khiến việc thực hiện và quản lý trở nên khó khăn hơn so với non-interleaved mapping
Minh họa ánh xạ từ VRB tới PRB		

- Các khối tài nguyên ảo được ánh xạ tới các khối tài nguyên vật lý theo chỉ định lược đồ ánh xạ, ánh xạ non-interleaved hoặc interleaved. Nếu không có sơ đồ ánh xạ nào được chỉ định thì UE sẽ giả sử ánh xạ non-interleaved.

- Đối với ánh xạ VRB-to-PRB non-interleaved, khối tài nguyên ảo n được ánh xạ tới khối tài nguyên vật lý n , ngoại trừ đối với truyền PDSCH được lập lịch với định dạng DCI 1_0 trong trường hợp đó là tài nguyên ảo khối được ánh xạ tới khối tài nguyên vật lý $n + N_{\text{start}}^{\text{CORESET}}$ trong đó $N_{\text{start}}^{\text{CORESET}}$ là tài nguyên vật lý được đánh số thấp nhất khối trong bộ tài nguyên điều khiển nơi nhận DCI tương ứng.

- Đối với ánh xạ VRB-to-PRB interleaved, quy trình ánh xạ được xác định bởi:

+ RB bundles được định nghĩa là

- Với truyền PDSCH được lập lịch với định dạng DCI 1_0 với CRC được xáo trộn bởi SI-RNTI trong Type0-PDCCH common search space trong CORESET 0, tập hợp các khối tài nguyên kích thước $N_{\text{BWP,init}}^{\text{size}}$ trong CORESET 0 được chia thành $N_{\text{bundle}} = \lfloor N_{\text{BWP,init}}^{\text{size}} / L \rfloor$, RB bundles theo thứ tự tăng dần của số khối tài nguyên và khi $L = 2$ thì kích thước số bundle và $N_{\text{BWP,init}}^{\text{size}}$ là kích thước của CORESET 0.
 - RB bundle $N_{\text{bundle}} - 1$ bao gồm $N_{\text{BWP,init}}^{\text{size}} \bmod L$ khối tài nguyên nếu $N_{\text{BWP,init}}^{\text{size}} \bmod L > 0$ và L khối tài nguyên với trường hợp còn lại
 - Tất cả RB bundle khác gồm L khối tài nguyên RB
- Với truyền PDSCH được lập lịch với định dạng DCI 1_0 trong bất kỳ không gian tìm kiếm chung nào trong bandwidth part i với vị trí bắt đầu $N_{\text{BWP,init}}^{\text{size}}$, ngoài Type0-PDCCH common search space trong CORESET 0, tập hợp $N_{\text{BWP,init}}^{\text{size}}$ khối tài nguyên ảo $\{0, 1, \dots, N_{\text{BWP,init}}^{\text{size}} - 1\}$, trong đó $N_{\text{BWP,init}}^{\text{size}}$ là kích thước của CORESET 0 nếu CORESET 0 được định cấu hình cho ô và kích thước của phần băng thông đường xuống ban đầu nếu không có CORESET 0 được định cấu hình cho ô, được chia thành N_{bundle} VRB bundle theo thứ tự tăng dần của số khối tài nguyên ảo và số gói ảo và tập hợp $N_{\text{BWP,init}}^{\text{size}}$ khối tài nguyên vật lý $\{N_{\text{start}}^{\text{CORESET}}, \dots, N_{\text{start}}^{\text{CORESET}} + N_{\text{BWP,init}}^{\text{size}} - 1\}$ được chia thành N_{bundle} PRB bundle theo thứ tự tăng dần của số PRB và số gói vật lý, trong đó $N_{\text{bundle}} = \lfloor (N_{\text{BWP,init}}^{\text{size}} + (N_{\text{BWP,i}}^{\text{start}} + N_{\text{start}}^{\text{CORESET}}) \bmod L) / L \rfloor$, $L=2$ là bundle size và $N_{\text{start}}^{\text{CORESET}}$ là khối tài nguyên vật lý được đánh số thấp nhất trong bộ tài nguyên điều khiển có DCI tương ứng đã nhận.
 - RB bundle 0 bao gồm $L - (N_{\text{BWP,i}}^{\text{start}} + N_{\text{start}}^{\text{CORESET}}) \bmod L$ RB
 - RB bundle $N_{\text{bundle}} - 1$ gồm $(N_{\text{BWP,init}}^{\text{size}} + N_{\text{BWP,i}}^{\text{start}} + N_{\text{start}}^{\text{CORESET}}) \bmod L$ RB nếu $(N_{\text{BWP,init}}^{\text{size}} + N_{\text{BWP,i}}^{\text{start}} + N_{\text{start}}^{\text{CORESET}}) \bmod L > 0$ và bằng L với trường hợp còn lại
 - Tất cả các gói khối tài nguyên khác bao gồm L RB.
- Với tất cả các truyền dẫn PDSCH khác, tập hợp $N_{\text{BWP,i}}^{\text{size}}$ RBs trong bandwidth i với vị trí bắt đầu $N_{\text{BWP,init}}^{\text{size}}$ chia thành các gói $N_{\text{bundle}} = \lfloor (N_{\text{BWP,i}}^{\text{size}} + (N_{\text{BWP,i}}^{\text{start}} \bmod L_i) / L_i \rfloor$ RB bundle theo thứ tự tăng dần của số khối tài nguyên và số gói trong đó L_i là kích thước gói cho bandwidth i được cung cấp bởi tham số lớp cao hơn *vrb-ToPRB-Interleaver* và
 - RB bundle 0 bao gồm bao gồm $L_i - (N_{\text{BWP,i}}^{\text{start}} \bmod L_i)$ RBs
 - RB bundle $N_{\text{bundle}} - 1$ gồm $(N_{\text{BWP,i}}^{\text{size}} + N_{\text{BWP,i}}^{\text{start}}) \bmod L_i$ RBs nếu $(N_{\text{BWP,i}}^{\text{size}} + N_{\text{BWP,i}}^{\text{start}}) \bmod L_i > 0$ và bằng L_i với trường hợp còn lại
 - Tất cả các gói khối tài nguyên khác bao gồm L_i RBi.

+ Các khối tài nguyên ảo trong khoảng $j \in \{1, 0, \dots, N_{\text{bundle}} - 1\}$ được ánh xạ tới các khối tài nguyên vật lý theo

- VRB bundle $N_{\text{bundle}} - 1$ được ánh xạ tới PRB bundle $N_{\text{bundle}} - 1$
- VRB bundle $j \in \{1, 0, \dots, N_{\text{bundle}} - 2\}$ được ánh xạ tới PRB bundle $f(i)$

$$\begin{aligned} f(j) &= rC + c \\ j &= cR + r \\ r &= 0, 1, \dots, R-1 \\ c &= 0, 1, \dots, C-1 \\ R &= 2 \\ C &= \lfloor N_{\text{bundle}} / R \rfloor \end{aligned}$$

+ UE dự kiến sẽ không được cấu hình đồng thời với $L_i = 2$ với kích thước PRG là 4

- Cùng một tiền mã hóa trong miền tần số được sử dụng trong PRB bundle và bundle size. Cùng một tiền mã hóa không được sử dụng cho các gói tài nguyên chung khác nhau

7. OFDM baseband signal generation

- 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 $l \in \{0, 1, \dots, N_{\text{subframe},\mu}^{\text{slot}} N_{\text{symbol}}^{\text{slot}} - 1\}$ trong một subframe được xác định bởi công thức

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

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

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

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

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

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

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

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 $t_{\text{start},l}^{\mu} - T_{\text{ext}} \leq t < t_{\text{start},l}^{\mu}$ preceding ký tự OFDM đầu tiên cho PUSCH, SRS hoặc PUCCH được đưa ra bởi

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

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

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

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

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

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 $T_{\text{TA}} = (N_{\text{TA}} + N_{\text{TA,offset}})/T_C$. 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 $T'_{\text{ext}} < T_{\text{symbol},(l-1)\bmod 7 - 2^{\mu}}^{\mu}$ 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	$\sum_{k=1}^{2^\mu} T_{\text{symb},(l-k) \bmod 7 \cdot 2^\mu}^\mu$

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

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