PDCCH : Transmitter Design and DMRS

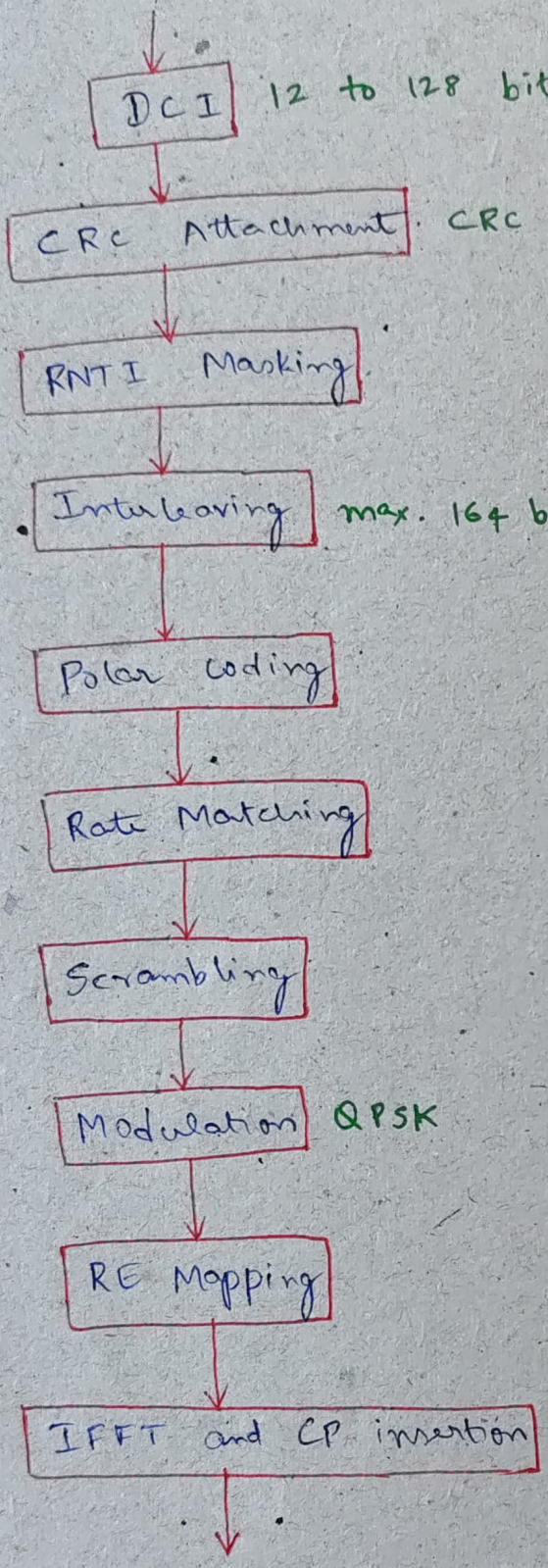
PDCCH → Physical Downlink Control Channel

This channel carries DCI (Downlink Control Information). DCI basically contains configurations for the PDSCH (downlink shared channel), PUSCH (uplink shared channel). The configuration can be anything like Resource allocation, related to power, related to Subframe structure, and so on. We'll understand more about DCI in the next section.

Lets discuss about the PDCCH transmitter chain. WKT, PDCCH carries DCI. The number of bits in the DCI can vary. The minimum number of bits can be 12. Suppose, if the no. of bits in the DCI is less than 12, then we pad zeros to make it 12. And, the maximum size of DCI can be 128 bits. So, the no. of bits in the DCI can be anywhere from 12 to 128 bits.

In general, in any Physical layer channel, the first thing that we do is, we add the CRC. In PDCCH, CRC 24 C is being used. The polynomial of the CRC24C looks like below.

$$g_{CRC24C}(D) = \left[D^{24} + D^{23} + D^{21} + D^{20} + D^{17} + D^{15} + D^{13} + D^{12} + D^{8} + D^{4} + D^{2} + D + 1 \right]$$


DCI
12 to 128 bits
CRC Attachment
CRC 24C
RNTI Masking
Interleaving
max. 164 bits supported
Polar coding
Rate Matching
Scrambling
Modulation
QPSK
RE Mapping
IFFT and CP insertion

We have already learnt how the CRC attachment is done. Using the modulo-2 binary division of the data with the CRC polynomial, and append the remainder at the end of the data bits.

After that, we do RNTI masking. (ie) XOR operation of last 16 bits of DCI + CRC with RNTI.

After that, we do Interleaving. This interleaving comes along with Polar Coding. This interleaving supports maximum of 164 bits.

Note: Even though, 128 bits (DCI) + 24 bits (CRC) = 152 bits support is sufficient for Interleaving, despite that, the Interleaving supports 164 bits.

After Interleaving, the bits go through the Polar Coding and Rate matching. WKT, the Rate Matching does Puncturing / Shortening / Repetition based on certain conditions.

After Rate Matching, the final bits go through the scrambling. We do the scrambling with the m-sequence. The input to the scrambling can be Scrambling ID / cell ID / RNTI / Combination of these, based on certain conditions. The bits before go through modulation, they go through Scrambling with m-sequence.

After that, we do modulation. We always do QPSK modulation in PDCCH.

After modulation, the modulated symbols are mapped to the Resources. The RE mapping part of PDCCH is quite complicated, we'll study in upcoming section where we understand CORESET and other different parameters.

DMRS for PDCCH:

7.4.1.3 Demodulation reference signals for PDCCH

7.4.1.3.1 Sequence generation

The UE shall assume the reference-signal sequence $r_l(m)$ for OFDM symbol l is defined by

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

where the pseudo-random sequence $c(i)$ is defined in clause 5.2.1. The pseudo-random sequence generator shall be initialized with

$$c_{init} = (2^{17}(N_{symb}^{slot} n_{s,f}^{\mu} + l + 1)(2N_{ID} + 1) + 2N_{ID}) \bmod 2^{31}$$

where l is the OFDM symbol number within the slot, $n_{s,f}^{\mu}$ is the slot number within a frame, and

- $N_{ID} \in \{0, 1, \ldots, 65535\}$ is given by the higher-layer parameter *pdcch-DMRS-ScramblingID* if provided
- $N_{ID} = N_{ID}^{cell}$ otherwise.

Using the C_{init} value, we generate the pseudo-random sequence. The C_{init} takes mainly three parameters:

(i) Slot Number $\longrightarrow n_{s,f}^{\mu}$

(ii) Symbol Number $\longrightarrow l$

(iii) Cell ID / Scrambling ID $\longrightarrow N_{ID}$

The number of symbols in a slot N_{symb}^{slot} is always 14. Based on where the DMRS is being mapped, the slot number and Symbol number will be different.

For each OFDM symbol inside a frame, the sequence that is being generated is different.

The N_{ID} remains the same (i.e) N_{ID}^{cell}, if the pdcch-DMRS-Scrambling ID is not provided.

Thus, $\left(N_{symb}^{slot} n_{s,f}^{\mu} + l + 1\right)$ ensures that, the DMRS sequence generated for each OFDM symbol inside a frame is different.

Once the sequence is generated, we do QPSK modulation. (ii) we generate the DMRS symbols and map them to the resources. So, finally $r_l(m)$ is mapped to the resources. (We'll understand this in the next section).

As we know, PDCCH carries configuration for DL and UL shared channels (ii) PDSCH and PUSCH. But, there is no channel that carries configuration for PDCCH. So, some configurations relevant to Coreset and Search space are provided to the UE via RRC messages. (We'll understand this in next section). But, there are few parameters that are not provided to the UE, like Aggregation level, Size of DCI, whether it is UL-DCI or DL-DCI, Location of DCI inside a Coreset. So, UE blindly decodes PDCCH and finds out the DCI. So, in order to reduce the no. of parameters that we blindly decode, we follow fixed modulation (QPSK). But, based on different channel conditions, we change the 'code Rate' using the parameter named Aggregation Level.

We'll understand Aggregation Level in the RE mapping in the next section. But in brief, we have different Aggregation Levels.

Aggregation Levels	QPSK modulated
1	→ 54 REs allocated → 108 bits
2	→ 108 REs allocated → 216 bits
4	⋮ ⋮
8	⋮ ⋮
16	→ 864 REs allocated → 1728 bits

So, if the channel conditions are bad, we use a higher aggregation level, and vice versa.

Now, lets take an example and understand the transmitter side processing.

Assume, the size of DCI bits is 40 bits. Then we add CRC24C of 24 bits. So, the final no. of bits will be 40 + 24 = 64 bits.

The RNTI masking and Interleaving do not change the number of bits.

Then we have Polar encoding. Lets say, we have encoded the 64 bits to 512 bits.

After that, we do Rate Matching. Based on the Aggregation Level, the Rate Matching will change.

Aggregation Level				Effective Code Rate
1	→ 54 REs	→ 108 bits	→ $\frac{64}{108}$	≈ 0.59
2	→ 108 REs	→ 216 bits	→ $\frac{64}{216}$	≈ 0.3
4	→ 216 REs	→ 432 bits	→ $\frac{64}{432}$	≈ 0.15
8	→ 432 REs	→ 864 bits	→ $\frac{64}{864}$	≈ 0.08
16	→ 864 REs	→ 1728 bits	→ $\frac{64}{1728}$	≈ 0.04

As the Aggregation Level increases, Code Rate decreases.

In Aggregation Level 16,

No. of bits encoded = 512 bits

No. of bits actually transmitted = 1728 bits.

Here, we do repetition (approximately 3.4 times)

In Aggregation Level 1 or 2,

No. of bits encoded = 512 bits

No. of bits actually transmitted = 108 bits or 216 bits.

Here, we do puncturing or shortening.

DCIs

There are 8 different types of DCIs. Out of 8, the following 4 are important.

	Fallback DCIs	Non Fallback DCIs
PUSCH (uplink)	DCI 0_0	DCI 0_1
PDSCH (Downlink)	DCI 1_0	DCI 1_1

DCI 0_0 and DCI 0_1 are used to carry configurations for Uplink shared channel (PUSCH)

DCI 1_0 and DCI 1_1 are used to carry configurations for Downlink shared channel (PDSCH)

Also, DCI 0_0 and DCI 1_0 are fallback DCIs. These DCIs are smaller in size.

And, DCI 0_1 and DCI 1_1 are non-fallback DCIs. These DCIs are larger in size.

The fallback DCIs are used when the payload size is less, and we can have high code Rate

---

Since this is Physical Layer course, we wont go deeper into the DCI contents (L2 aspects). We'll just brief about the DCIs.

- The first bit of fallback DCIs tells whether it is DCI 0_0 (uplink) or DCI 1_0 (Downlink). Also, we Pad zeros at the end to make sure that, the size of DCI 0_0 and DCI 1_0 is same.
- Similarly, the first bit of non-fallback DCIs tells whether it is DCI 0_1 (uplink) or DCI 1_1 (Downlink) Also, we pad zeros at the end to make sure that, the size of DCI 0_1 and DCI 1_1 is same.
- DCI 0_0 in uplink, is mainly used for single layer transmission.
- DCI 0_1 in uplink, is used for multi layer transmission, since DCI 0_1 has more number of bits.
- There are RNTIs associated with DCIs. Example, DCI 1_0 is associated with RA-RNTI, TC-RNTI, SI-RNTI, Paging-RNTI, C-RNTI, MCS-C-RNTI and CS-RNTI.
  DCI 1_1 is associated with C-RNTI, CS-RNTI and MCS-C-RNTI.

Apart from these 4 DCI's; there are 4 more DCI's.

DCI 2_0	DCI 2_1
DCI 2_2	DCI 2_3

DCI 0_0 / DCI 0_1 / DCI 1_0 / DCI 1_1 carries cfg. for a single UE. (or) cfg. for broadcast messages in UE specific search space (or) system level configurations.

Whereas, DCI 2_0 / DCI 2_1 / DCI 2_2 / DCI 2_3 carries cfg. for a group of UEs, along with the DCI positions of the corresponding UEs. Each UE will decode its respective DCI, based on the parameter 'position in DCI'.

- DCI 2_0 is for Slot Format Indicator (SFI). (i) basically DCI 2_0 conveys UE, if there is a change in the slot format. DCI 2_0 goes with SFI-RNTI. The maximum size can go upto 128 bits.

- DCI 2_1 is for Pre-emption Indication. (i) lets say, there is a continuous DL data, and gNB wants to preempt the ongoing PDSCH transmission for a particular UE which has latency critical transmission. In this case, we indicate the same through DCI 2_1.

  DCI 2_1 goes with INT-RNTI.

⊙ DCI 2_2 is used for Transmit Power Control (TPC) for PUSCH and PUCCH. DCI 2_2 goes with TPC-PUSCH-RNTI (or) TPC-PUCCH-RNTI.

⊙ DCI 2_3 is used for Transmit Power Control (TPC) for SRS. DCI 2_3 can also used to request an aperiodic SRS (we'll learn about this in SRS section). And, DCI 2_3 goes with TPC-SRS-RNTI.

NOTE:

Size of DCI 1_0, DCI 2_2 and DCI 2_3 are same.

CORESET (Control Resource Set)

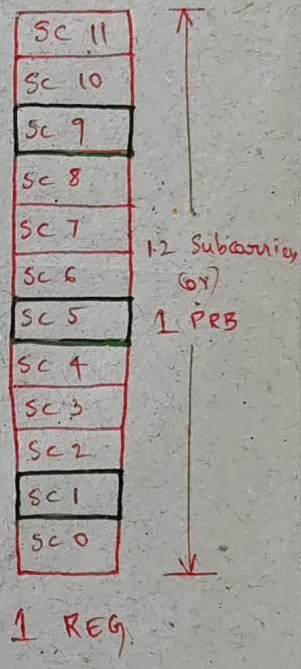
CORESET is one of the important parameter in PDCCH. In order to understand CORESET, let's first understand definitions of REG and CCE.

REG → Resource Element Group.

→ 12 Subcarriers (in frequency) and 1 Symbol (in time) } = 1 REG

→ 12 Subcarriers means 1 PRB

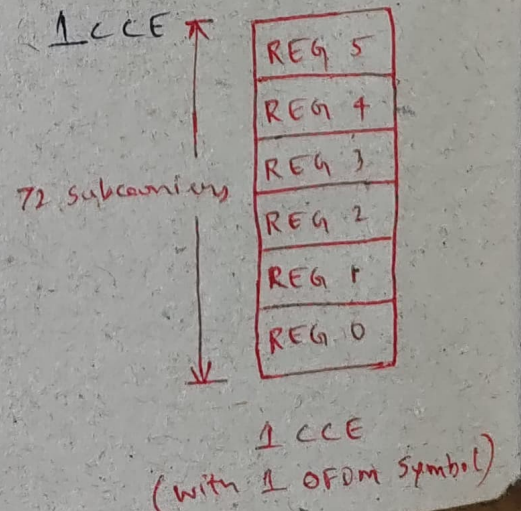
→ In PDCCH, the DMRS has fixed locations inside a PRB. (i.e) SC 1, SC 5 and SC 9. And the remaining subcarriers are for data. So, 1 REG has 9 REs for data where the DCI will go, and 3 REs for DMRS.



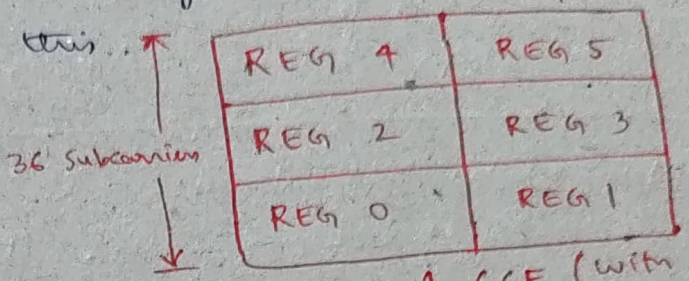
CCE → Control Channel Element

→ 6 REGs = 1 CCE

→ If we have only one OFDM symbol allocated for a CORESET, then 1 CCE looks like this:

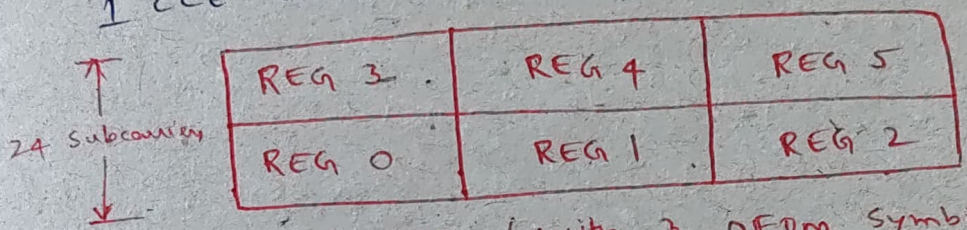


→ If there are 2 OFDM symbols allocated, then 1 CCE looks like this.



1 CCE (with 2 OFDM symbols)

→ If there are 3 OFDM symbols allocated, then 1 CCE looks like this.



1 CCE (with 3 OFDM symbols)

Note:

Be it 1 CCE (with 1/2/3 OFDM symbols), there exists 9 data REs in each REG. So, for all the above 1 CCE cases, there exists 9 × 6 = 54 REs for data.

In the previous section, we seen that the data allocation happens in terms of Aggregation Level. In 5G PDCCH,

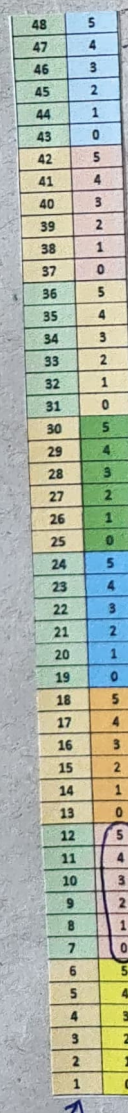
Aggregation Level 1	→	1 CCE
Aggregation Level 2	→	2 CCE
⋮		
Aggregation Level 16	→	16 CCEs

So, when we say, a UE allocated a DCI with Aggregation Level 16 (i.e) 16 CCEs, this means, in total 54 × 16 REs are allocated, each with QPSK modulated. So, the total number of bits that can be transmitted

with this Aggregation Level is 56 x 16 x 2 bits. This is the relation between Aggregation Level and CCE.

Now let's understand how they actually look like in the complete bandwidth. In 5G, the CORESET allocation in frequency is in terms of bitmap. (i) We have a 45 bit length frequency bitmap, where each bit represents 6 PRB's. So, in total 45 x 6 = 270 PRB's. So, this bitmap actually tells us "What are the PRB's allocated / not allocated for the CORESET"!

The CORESET configuration doesn't tell us which OFDM symbols. It only tells the NO. of OFDM symbols (1/2/3)



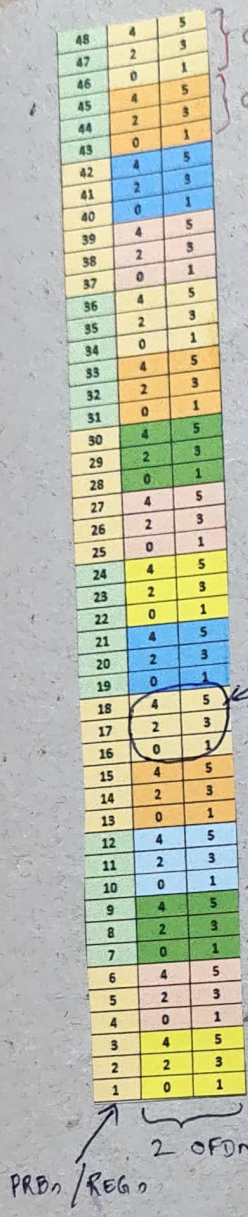
① If the NO. of OFDM symbols is 1, and let's say the CORESET allocation bitmap is

11111111 0000000 ... 0

8 bits 37 bits

45 bits

which means, there exists 8 x 6 = 48 PRB's allocated.

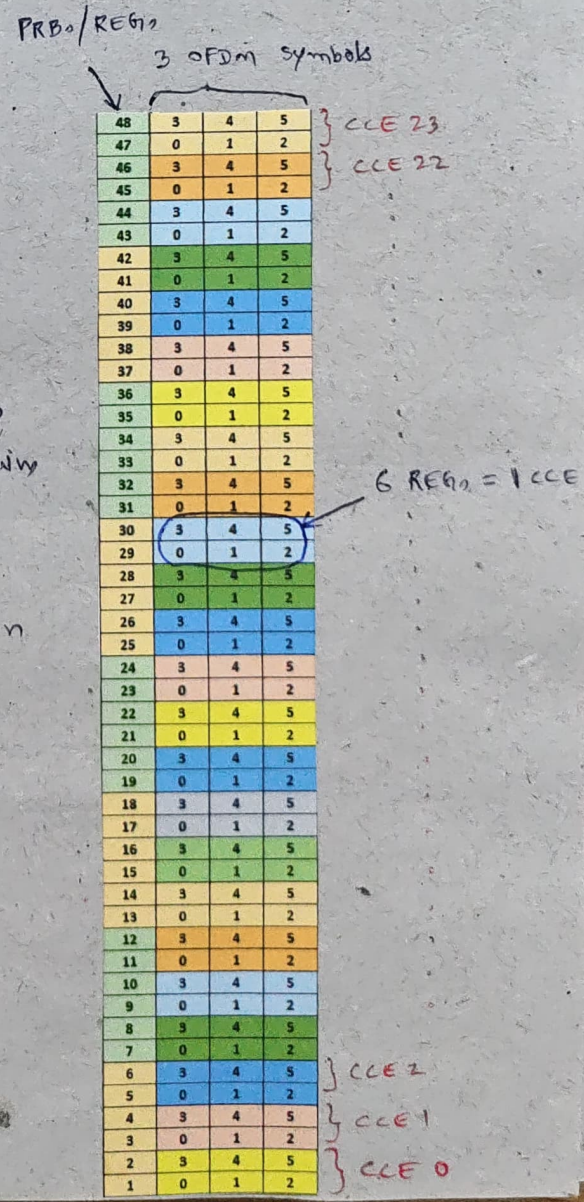


⊙ When there are 2 OFDM symbols for a CORESET, the bitmap remains the same.

⊙ The CCE allocation will be like shown in figure.

⊙ When we have 3 OFDM symbols for CORESET, the bitmap remains the same.

⊙ The CCE allocation is shown in figure.



There can be 12 CORESET configured for a serving cell. (ii) A gNB can have 12 different CORESETs. For a single UE (or) for an active Bandwidth part, maximum 3 CORESETS can be configured, where the CORESET 0 is always for SIB1 DCI. (will see later)

There are 2 types of CCE to REG mapping.

(i) Non-interleaved mapping

(ii) Interleaved mapping

(i)

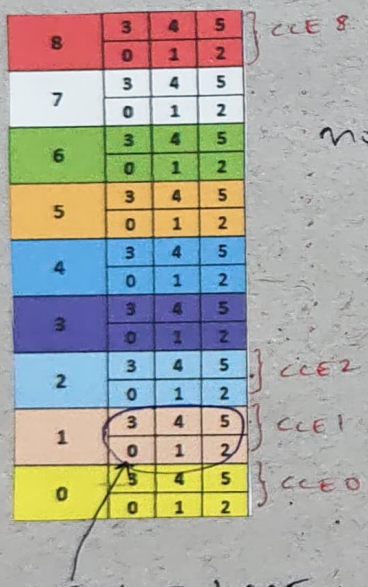
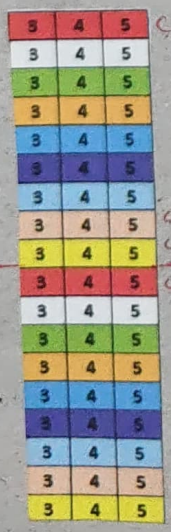


Figure shows the small CORESET with non-interleaved mapping. This can be used for interference coordination with neighbouring cell, which means, "One cell can transmit at one part of the band, another cell can transmit at another part of the band; and the two cell can coordinate; to reduce the interference".

(ii)



Interleaved mapping has couple of parameters.

- Bundle size
- Interleaver size

Bundle size tells, how many REGs are bundling. Bundle size can be 2/3/6.

Interleaver size tells, how the bundles are going to be interleaved. Interleaver size can also be 2/3/6. (ii) If the Interleaver size is 2, then we divide the allocation into 2 parts, and interleave as shown in the figure.

(ii) For the example shown in figure, the Bundling size is 3, and the Interleaver size is 2.

This is how, the PDCCH allocation goes like.

There is one more parameter to consider. (ii) Shift index. Shift index basically moves the CCE pattern upwards, and wrap around from the top. It basically tells, how much we shift in the frequency.

Search Space

Search Space is a set of RBs and Symbols where UE tries to attempt to decode PDCCH inside a CORESET. As we have seen, CORESET gives us a configuration in terms of No. of PRBs, and total no. of Symbols. (For example, 48 PRBs and 3 Symbols).

The CORESET doesn't communicate the starting Symbol to the UE. The Starting Symbol is communicated by the Search Space.

CCE Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
8	3	4	5											
	0	1	2											
7	3	4	5											
	0	1	2											
6	3	4	5											
	0	1	2											
5	3	4	5											
	0	1	2											
4	3	4	5											
	0	1	2											
3	3	4	5											
	0	1	2											
2	3	4	5											
	0	1	2											
1	3	4	5											
	0	1	2											
0	3	4	5											
	0	1	2											

For this example, the CORESET configuration is

- No. of CCEs = 9
  - ⇒ No. of PRBs = 9 × 6 = 54
- No. of symbols = 3

When UE tries to decode DCI (Blind decoding), it doesn't know the Aggregation Level, what are the CCEs allocated to it, how many times should it try decoding different CCEs, and so on. In order to help UE, some of the information is communicated through Search Space, which reduces the No. of blind decoding attempts to decode DCI, by the UE.

One such paramter is, $M_{PDCCH}^{max,slot,\mu}$. (ii) Maximum Number of monitored PDCCH candidates per slot, and per serving cell.

μ	Maximum number of monitored PDCCH candidates per slot and per serving cell $M_{PDCCH}^{max,slot,\mu}$
0	44
1	36
2	22
3	20

So, a UE is not supposed to monitor more than these many PDCCH candidates per slot, for different numerologies.

$M_{PDCCH}^{max-slot,\mu}$ decreases as Numerology (μ) increases, because the slot / symbol duration reduces, so UE has less time to decode.

PDCCH candidate means, number of PDCCH transmissions for a given aggregation level. This is also communicated to the UE via Search Space. For example,

Aggregation Level	No. of PDCCH candidates
1	4
2	2
4	0
8	0
16	0

This information is communicated to the UE through Search space.

So, basically, UE knows that, its DCI is not in Aggregation Level 4, 8, and 16. Its on Aggregation Levels 1 and 2. And the UE finds out the possible CCEs for Aggregation Levels 1 and 2, using a Hash function. (We'll see this in next section).

So, basically, a search space is the region in the Figure, which are colored. Lets say, in this example, we have 9 CCEs, 3 symbols, starting from 0. And lets say, we communicate that, UE is supposed to try decoding on

CCE 0, CCE 3 and CCE 6, and see if it is able to decode the DCI or not.

This is about the Search space

There are 2 types of Search space.

- ⊙ Common Search Space (CSS)
- ⊙ UE Specific Search Space (USS)

Any common DCI,

- DCI for SIB 1 (SI-RNTI)
- Initial Access / PRACH related DCI (TC-RNTI / RA-RNTI)
- DCI 2_0 (for group of UEs)

all these DCIs goes through Common Search Space (CSS).
And, the UE specific data, say DCI 1_1, DCI 0_1 goes through UE specific Search space (USS).

There are many different types of common search space (Type 0 / 0A / 1 / 2 / 3), and different types of DCI, mapped to different type of Search Space. Lets not go much into those details, we'll only focus on Physical Layer aspects.

Each Search space is mapped to a CORESET. There are, in total, 40 Search space that gNB can configure. And, 12 CORESET that gNB can configure.

So, when gNB configures a Search space, it also communicates "which Coreset this Search Space belong to".

PDCCH : Receiver and Blind decoding

$$L \cdot \left\{ \left(Y_{p,n_{s,f}^{\mu}} + \left\lfloor \frac{m_{s,n_{CI}} \cdot N_{CCE,p}}{L \cdot M_{s,max}^{(L)}} \right\rfloor + n_{CI} \right) \bmod \lfloor N_{CCE,p}/L \rfloor \right\} + i$$

This formula is used by the UE to find out CCE index for a given Aggregation Level, given Slot, given CORESET, given Search Space. The CCE Indices given by this formula are used by the UE to blindly decode the DCI.

$L \rightarrow$ Aggregation Level

$i \rightarrow$ Aggregation Level index $(0, 1, 2, \ldots, L-1)$

Eg. If $L = 4$, then $i = 0, 1, 2$ and 3.

$s \rightarrow$ Search Space Index

$p \rightarrow$ Coreset index

$m_{s,n_{CI}} \rightarrow$ PDCCH candidates index $\left(0, 1, 2, \ldots, m_{s,n_{CI}}^{(L)} - 1\right)$

Eg. If the No. of PDCCH candidates for an Aggregation Level L is 4, then

$m_{s,n_{CI}} = 0, 1, 2$ and 3.

$N_{CCE,p} \rightarrow$ NO. of CCEs in a coreset. They are indexed from $0, 1, 2, \ldots, N_{CCE,p} - 1$.

$n_{CI} \rightarrow$ Carrier Indicator field

$= 0$, for Common search space (CSS)

$n_{s,f}^{\mu} \rightarrow$ Slot number

$$Y_{p,n^{M}_{s,f}} = \begin{cases} 0, & \text{for Common Search space (CSS)} \\ \left(A_p * Y_{p,n^{M}_{s,f}^{-1}}\right) \bmod D, & \text{for UE specific Search space (USS)} \end{cases}$$

$$Y_{p,n^{M}_{s,f}^{-1}} \doteq Y_{p,-1}, \text{ if } n^{M}_{s,f} = 0$$

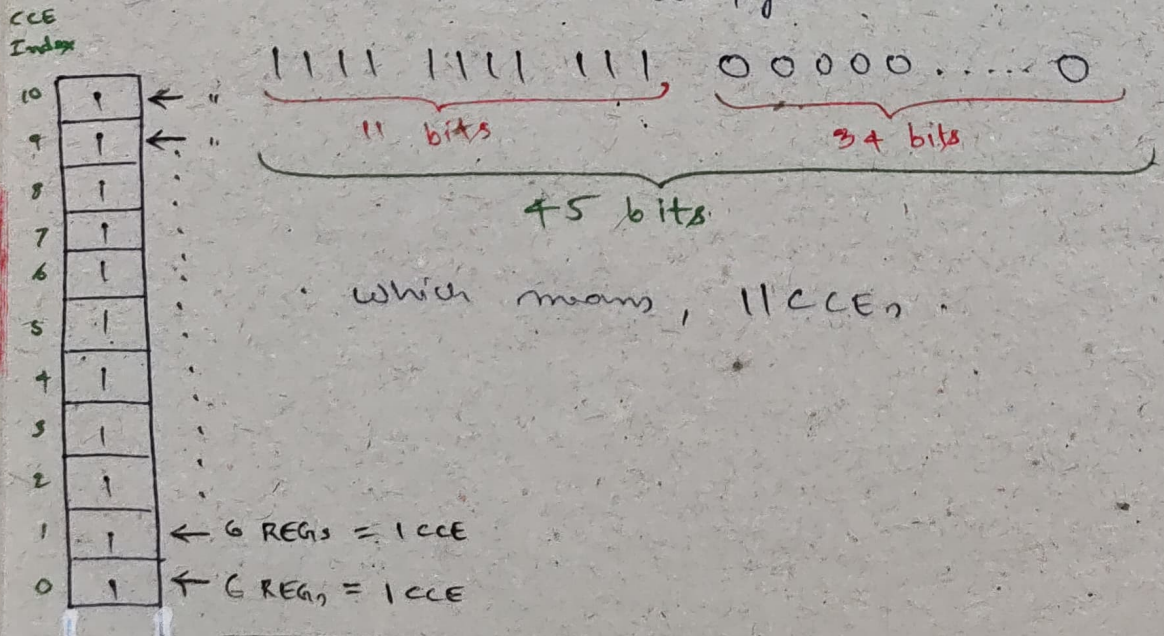
$$= n_{RNTI} = C\text{-}RNTI$$

$$A_p = \begin{cases} 39827, & \text{if } p \bmod 3 = 0 \\ 39829, & \text{if } p \bmod 3 = 1 \\ 39839, & \text{if } p \bmod 3 = 2 \end{cases}$$

$$D = 65537$$

(ii) for USS, $Y_{p,n^{M}_{s,f}}$ is different for different values of n_{RNTI} and p.

Now, lets take an example and understand this further. consider the CORESET configuration



Total: 66 REGs (or) 66 PRBs (or) 11 CCEs

This is how the CORESET looks like.

Lets take slot index = 0.

Search space configuration,

No. of Candidates = { 0, for Aggregation Level 1
4, for Aggregation Level 2
2, for Aggregation Level 4
0, for Aggregation Level 8
0, for Aggregation Level 16

Lets try to get the CCE indices for Aggregation Level 2, where there are 4 candidates.

As we know from the formula, $M^{(L)}_{s,max}$ for Aggregation Level 2 is

$$M^{(2)}_{s,max} = 4 \quad (\because \text{there are 4 candidates})$$

$$m_{s,n_{CI}} = 0, 1, 2 \text{ and } 3$$

Since CORESET configuration is 11 bits and 1 symbol,

$$N_{CCE,p} = 11$$

$$L = 2 \quad (\because \text{Aggregation Level 2})$$

$$i = 0, 1 \quad (\text{Aggregation Level index})$$

Lets assume its a Common Search Space (CSS)

$$Y_{p,n^{\mu}_{s,f}-1} = 0$$

$$n_{CI} = 0$$

Now, let us these parameters and find out what are the CCE, that UE will look into.

Rewriting the formula,

$$L \cdot \left\{ \left(0 + \left\lfloor \frac{m_{s,0} \cdot N_{CCE,p}}{L \cdot M_{s,max}^{(L)}} \right\rfloor + 0 \right) \bmod \left\lfloor \frac{N_{CCE,p}}{L} \right\rfloor \right\} + i$$

$$= L \cdot \left\{ \left\lfloor \frac{m_{s,0} \cdot N_{CCE,p}}{L \cdot M_{s,max}^{(L)}} \right\rfloor \bmod \left\lfloor \frac{N_{CCE,p}}{L} \right\rfloor \right\} + i$$

$$= 2 \cdot \left\{ \left\lfloor \frac{m_{s,0} \cdot 11}{2 \times 4} \right\rfloor \bmod \left\lfloor \frac{11}{2} \right\rfloor \right\} + i$$

$$= 2 \left\{ \left\lfloor \frac{m_{s,0} \times 11}{8} \right\rfloor \bmod 5 \right\} + i$$

Now, WKT, $m_{s,0}$ has values 0, 1, 2 and 3.

And, for each candidate, i will be 0 and 1. ($\because L = 2$)

① $m_{s,0} = 0 \Rightarrow 2\{0 \bmod 5\} + i \Rightarrow$ 0, 1

② $m_{s,0} = 1 \Rightarrow 2\left\{\left\lfloor \frac{1 \times 11}{8} \right\rfloor \bmod 5\right\} + i$

$= 2\{1 \bmod 5\} + i \Rightarrow$ 2, 3

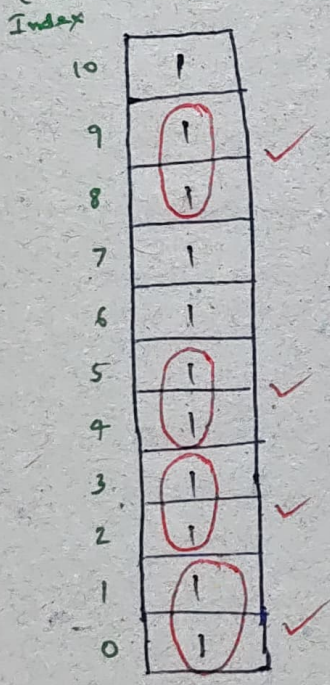
③ $m_{s,0} = 2 \Rightarrow 2\left\{\left\lfloor \frac{2 \times 11}{8} \right\rfloor \bmod 5\right\} + i$

$= 2\{2 \bmod 5\} + i \Rightarrow \underline{4, 5}$

④ $m_{s,0} = 3 \Rightarrow 2\left\{\left\lfloor \frac{3 \times 11}{8} \right\rfloor \bmod 5\right\} + i$

$= 2\{4 \bmod 5\} + i \Rightarrow \underline{8, 9}$

So, for Aggregation Level 2, for 4 Candidates, the possible CCEs are



(ii) gNB sends DCI in this CORESET at this Search Space for a particular UE, in one of these possible CCEs.

UE will find out these possible CCE indices, and put it to use one by one.

First UE picks CCE Indices 0, 1 (it depends on the implementation), and then it proceeds with full receiver processing (resource demapping, demodulation, descrambling, de-Rate Matching, decoding, all the way till CRC check) with that particular RNTI which is being used for that particular DCI. If CRC fails (∵ there is no data here for this UE, it will fail for this RNTI). Then, the UE uses CCE Indices 2, 3 and proceeds with full processing again. This continues until the CRC pass (∵ data actually allocated

at their CCE's for this RNTI and UE) for this particular RNTI (SI-RNTI / Paging-RNTI / so on).

[redacted]

Since the Aggregation Level is also not known, for Aggregation Level 4, we might get indices like 0,1,2,3 and 4,5,6,7. (Use the same formula and find it out). And, UE also doesn't know whether the data is allocated with Aggregation Level 2 or 4. So, UE have to blindly decode the Aggregation Level as well. (ii) UE may search either with Aggregation Level 2 on CCE's (0,1 ; 2,3 ; 4,5 ; 8,9) or with Aggregation Level 4 on CCE's (0,1,2,3 ; 4,5,6,7) : to find out where the DCI is.

This is how the receiver works.