5G NR Base Station Fixed-point Algorithm Design (Receiver)

Document Number:	
Version Number:	V0.1
Product:	XXX1234/XXX123
Reference:	1

1 Scope

The present document describes the encode and decode of physical channels for 5G-NR.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 38.201: "NR; Physical Layer – General Description"

[3] 3GPP TS 38.202: "NR; Services provided by the physical layer"

[4] 3GPP TS 38.212: "NR; Multiplexing and channel coding"

[5] 3GPP TS 38.213: "NR; Physical layer procedures for control "

[6] 3GPP TS 38.214: "NR; Physical layer procedures for data "

[7] 3GPP TS 38.215: "NR; Physical layer measurements"

[8] 3GPP TS 38.104: "NR; Base Station (BS) radio transmission and reception"

[9] void

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the following definitions apply:

3.2 Symbols

For the purposes of the present document, the following symbols apply:

$(k,l)_{p,\mu}$	Resource element with frequency-domain index k and time-domain index l for antenna port p and subcarrier spacing configuration μ;
Δf	Subcarrier spacing
Δf_{RA}	Subcarrier spacing for random-access preambles
T_{c}	Basic time unit for NR;
T_{f}	Radio frame duration;

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

FPGA	Field Programmable Gate Array
BWP	Bandwidth part
CCE	Control channel element
DCI	Downlink Control Information
DM-RS	Demodulation reference signal
FR1	Frequency range 1 as defined in [8, TS 38.104]
FR2	Frequency range 2 as defined in [8, TS 38.104]
PBCH	Physical broadcast channel
PDCCH	Physical downlink control channel
PDSCH	Physical downlink shared channel
PRACH	Physical random-access channel
PRB	Physical resource block
PSS	Primary synchronization signal
PT-RS	Phase-tracking reference signal
PUCCH	Physical uplink control channel
PUSCH	Physical uplink shared channel
REG	Resource-element group
SRS	Sounding reference signal
SSS	Secondary synchronization signal
VRB	Virtual resource block

4 Encode and decode of physical channels for 5G-NR

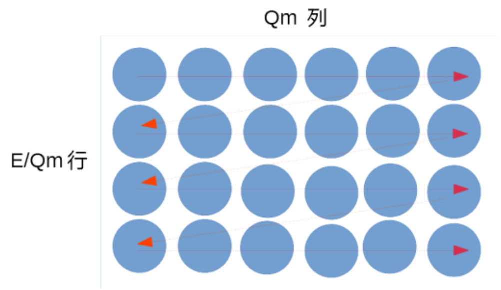
4.1 De Rate Matching

[1] LDPC De Rate matching

PUSCH User Data 需要经过解速率匹配之后，才可以提供给 LDPC 信道译码模块. 解速率匹配包括两个步骤：解交织和 Bit Selection 的逆过程；解交织相对来说比较简单，Bit Selection 的逆过程由于涉及 Ncb 的取值问题（Ncb<=N）和 NULL Bits 位置还原和填充，因此本算法说明书采取了“分类讨论”的思想，对可能出现的情形进行了详细的讨论、研究、总结和归纳. 具体的实现细节请参见后续章节，由于时间有限，需要各位系统构架师和算法工程师给予审核和指导.

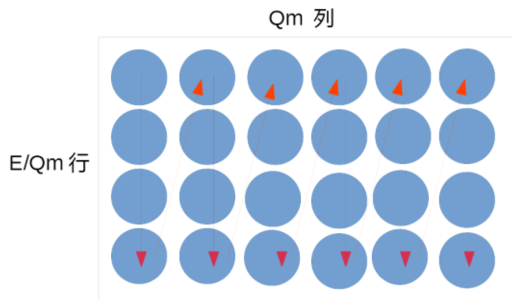
[Step 1] Deinterleaving

从 MPU 处理器侧获取上层参数:每个 Code Block 数据长度 E，以及调制阶数 Qm.
这里需要明确:每个编码子块序列,数据长度都是(NL*Qm)的整数倍.
将数据序列 $f_0, f_1, f_2, \ldots, f_{E-1}$ 写入一个 E/Qm 行，Qm 列的矩阵空间，进行摆放，如图所示：



随后再从 E/Qm 行，Qm 列的矩阵空间，将数据读出并写入新的序列

$e_0, e_1, e_2, \ldots, e_{E-1}$，如图所示：



完成 LDPC DeRateMatching 的第一个处理过程：解交织.
特别注意，E 和 Qm，两个参数直接来自于 ARM 处理器的配置

这个处理过程，需要使用一个 Dual Port SRAM, RTL 顶层可以表示为:

```
module DualPortSRAM(
input wire WriteClock,
input wire ReadClock,
input wire WriteEn,
input wire ReadEn,
input wire [W_ADD_W-1:0]WriteAddress,
input wire [R_ADD_W-1:0]ReadAddress,
input wire [W_DATA_W-1:0]WriteData,// Whether the data width is 1 ?
output reg [R_DATA_W-1:0]ReadData // Whether the data width is 1 ?
);
Endmodule
```

双端口 SRAM，在数据写入的时候需要注意 WriteEn 和 WriteAddress 的控制，相对来说比较简单.

当 Code Block 解交织启动的时候，WriteEn=1’b1；同时 WriteAddress 从 0 开始自增至 E-1,将 Code Block 数据写入 Dual Port SRAM.

为了保证解码的实时性，当 WriteAddress 自增至 E-Qm 的时候，ReadEn=1’b1；同时 ReadAddress 按照如下规律进行自增操作,将数据依次读出，写入序列 $e_0, e_1, e_2, \ldots, e_{E-1}$.

```
I=0;
J=0;
for(I=0;I<Qm;I++)
  for(J=0;J<E/Qm;J++))
    ReadAddress=I+J*Qm;
  end for
end for
```

[Step 2] De Bit selection (Bit Selection 的逆过程)

这个步骤的目标是，尽最大努力来还原 LDPC Channel Coding 所生成的序列，然后提供给 LDPC 信道译码模块.

为什么是尽最大努力呢？个人感悟，这里面有两个要点：

[Point 1]信道编码所输出的序列，某些配置情形，并没有完全映射到物理信道.

The bit sequence after encoding $\boxed{d_0, d_1, d_2, \ldots, d_{N-1}}$ from Subclause 5.3.2 is written into a circular buffer of $\boxed{\text{length } N_{cb}}$ for the r-th coded block, where N is defined in Subclause 5.3.2.

For the r-th code block, let $N_{cb} = N$ if $I_{LBRM} = 0$ and $\boxed{N_{cb} = \min(N, N_{ref}) \text{ otherwise}}$, where $\boxed{N_{ref} = \left\lfloor \frac{TBS_{\text{LBRM}}}{C \cdot R_{\text{LBRM}}} \right\rfloor}$, $R_{\text{LBRM}} = 2/3$, TBS_{LBRM} is determined according to Subclause 6.1.4.2 in [6, TS 38.214] for UL-SCH and Subclause 5.1.3.2 in [6, TS 38.214] for DL-SCH/PCH, assuming the following:

[Point 2]时域和频域上分配的资源（Resource Block)受到限制，可用于传输用户数据 User Data 编码 Bits 数量 G，决定了编码子块(Code Block)的数据长度 E.

if $j \le C' - \mathrm{mod}(G/(N_L \cdot Q_m), C') - 1$

$$E_r = N_L \cdot Q_m \cdot \left\lfloor \frac{G}{N_L \cdot Q_m \cdot C'} \right\rfloor;$$

else

$$E_r = N_L \cdot Q_m \cdot \left\lceil \frac{G}{N_L \cdot Q_m \cdot C'} \right\rceil;$$

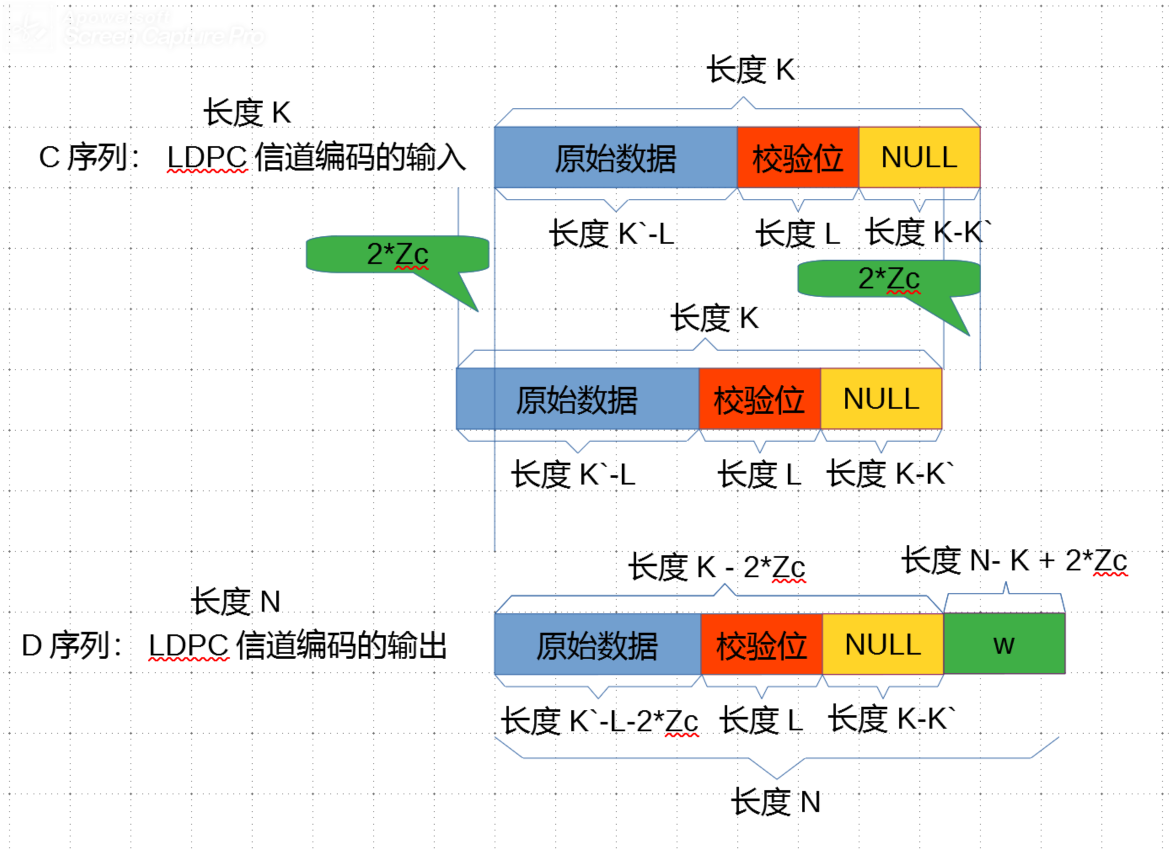
end if

如下表格中的参数 Ncb，K0 直接来自于 ARM 处理器的配置

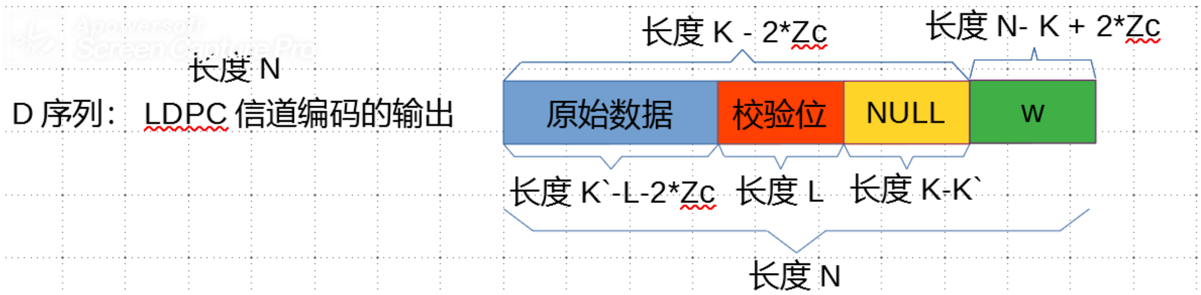
Table 5.4.2.1-2: Starting position of different redundancy versions, k_0

rv_{id}	k_0	
	LDPC base graph 1	LDPC base graph 2
0	0	0
1	$\left\lfloor \frac{17N_{cb}}{66Z_c} \right\rfloor Z_c$	$\left\lfloor \frac{13N_{cb}}{50Z_c} \right\rfloor Z_c$
2	$\left\lfloor \frac{33N_{cb}}{66Z_c} \right\rfloor Z_c$	$\left\lfloor \frac{25N_{cb}}{50Z_c} \right\rfloor Z_c$
3	$\left\lfloor \frac{56N_{cb}}{66Z_c} \right\rfloor Z_c$	$\left\lfloor \frac{43N_{cb}}{50Z_c} \right\rfloor Z_c$

同时，非常关键的是需要分析 LDPC Channel Coding 整个编码过程，请您参见：



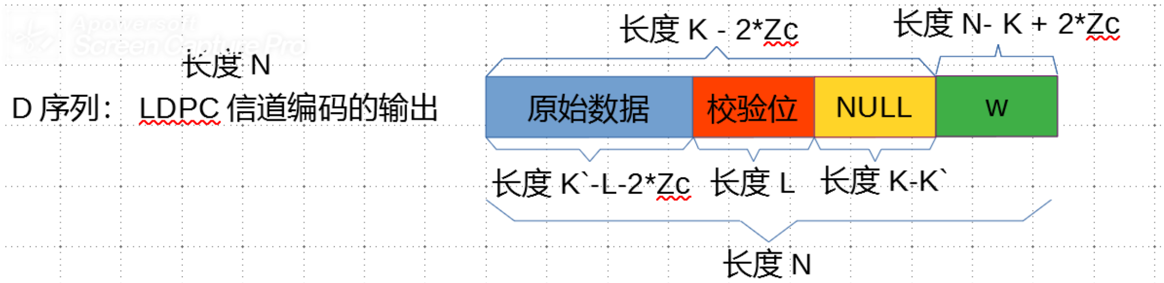
因此，可以得到基于 LDPC 信道编码，输出序列的数据帧结构:



同时基于如下公式:

$$
\begin{aligned}
& k = 0; \\
& j = 0; \\
& \text{while } k < E \\
& \quad \text{if } d_{(k_0+j) \bmod N_{cb}} \neq <NULL> \\
& \quad\quad e_k = d_{(k_0+j) \bmod N_{cb}}; \\
& \quad\quad k = k + 1; \\
& \quad \text{end if} \\
& \quad j = j + 1; \\
& \text{end while}
\end{aligned}
$$

我们就可以实现 Bit selection 的逆过程，需要分为 4 类进行讨论:



[Type 1] Ncb=N

1.1 如果 K0 地址位于 NULL 字段之内的话，(K`−2Zc) <= K0 < (K−2Zc)

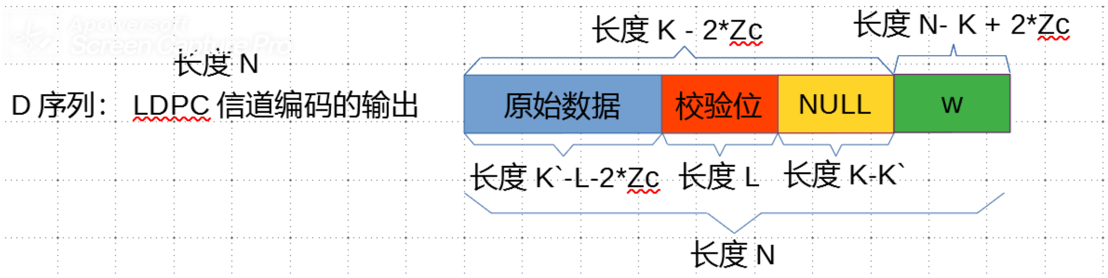
```
if x 位于 NULL 字段，(K`−2Zc) <= x < (K−2Zc)
   序列 d(x)=NULL;
Else if( x >= (K−2*Zc) )
   If( (x − (K−2*Zc)) < E )
      序列 d(x)=e(x − (K−2*Zc));
   Else
      序列 d(x)=NULL;
Else if( x < (K`−2*Zc) )
   If( (x + N − K + 2*Zc) < E )
      序列 d(x)=e(x + (N − K + 2*Zc));
   Else
      序列 d(x)=NULL;
```

1.2　如果 K0 地址位于 NULL 字段之前的话，　K0 < (K`−2Zc)

```
if x 位于 NULL 字段, (K`−2Zc) <= x < (K−2Zc)
  序列 d(x)=NULL;
Else if( x < K0 )
  If( (x + K` − K0 + N − K) < E )
    序列 d(x)=e(x + K` − K0 + N − K);
  Else
    序列 d(x)=NULL;
Else if ( ( x >= K0 ) && (x < K`−2Zc) )
  If( (x − K0) < E )
    序列 d(x)=e(x − K0);
  Else
    序列 d(x)=NULL;
Else if ( x > K−2Zc )
  If( (x + K` − K0 − K) < E )
    序列 d(x)=e(x + K` − K0 − K);
  Else
    序列 d(x)=NULL;
```

1.3　如果 K0 地址位于 NULL 字段之后的话, K0 >= (K−2Zc)

```
if x 位于 NULL 字段, (K`−2Zc) <= x < (K−2Zc)
  序列 d(x)=NULL;
Else if( x >= K0 )
  If( (x − K0) < E )
    序列 d(x)=e(x − K0);
  Else
    序列 d(x)=NULL;
Else if( x < K`−2Zc )
  If( (x + N−K0) < E )
    序列 d(x)=e(x + N−K0);
  Else
    序列 d(x)=NULL;
Else if( ( x >= K−2Zc ) && (x < K0) )
  If( (x + K` − K0 + N − K) < E )
    序列 d(x)=e(x + K` − K0 + N − K);
  Else
    序列 d(x)=NULL;
```



[Type 2] Ncb<N，如果 Ncb>(K−2Zc)的话，那么这个时候的处理方式只需要将上述 Type 1 公式里面的 N 替换为 Ncb 即可.

特别注意：序列 d(x),x 的取值范围仅仅是 0<=x<Ncb，取值范围之外的话 d(x)=NULL.

[Type 3] Ncb<N，如果(K`−2Zc) < Ncb <= (K−2Zc)的话，这个时候的处理方式是:

特别注意：序列 d(x),x 的取值范围仅仅是 0<=x<Ncb，取值范围之外的话 d(x)=NULL.

3.1 如果 K0 地址位于 NULL 字段之内的话, (K`−2Zc) <= K0 < (K−2Zc)

```
if x 位于 NULL 字段, (K`−2Zc) <= x < (K−2Zc)
   序列 d(x)=NULL;
Else if( x >= (K−2*Zc) )
   序列 d(x)=NULL;
Else if( x < (K`−2*Zc) )
   If( x< E )
      序列 d(x)=e(x);
   Else
      序列 d(x)=NULL;
```

3.2 如果 K0 地址位于 NULL 字段之前的话, K0 < (K`−2Zc)

```
if x 位于 NULL 字段, (K`−2Zc) <= x < (K−2Zc)
   序列 d(x)=NULL;
Else if( ( x >= K0 ) && ( x < K`−2Zc ) )
   If( (x − K0) < E )
      序列 d(x)=e(x − K0);
   Else
      序列 d(x)=NULL;
Else if( x < K0 )
   If( (x + K` − 2Zc − K0) < E )
```

```
      序列 d(x)=e(x + K` − 2Zc − K0);
   Else
      序列 d(x)=NULL;
```

[Type 4] Ncb<N，如果 Ncb <= (K`−2Zc)的话，这个时候的处理方式是:

特别注意：序列 d(x),x 的取值范围仅仅是 0<=x<Ncb，取值范围之外的话 d(x)=NULL.

4.1 显然 K0 地址，只能位于 NULL 字段之前, K0 < (K`−2Zc)

```
if( ( x >= K0 ) && ( x < K`−2Zc ) )
   If( (x − K0) < E )
      序列 d(x)=e(x − K0);
   Else
      序列 d(x)=NULL;
Else if( x < K0 )
   If( (x + K` − 2Zc − K0) < E )
      序列 d(x)=e(x + K` − 2Zc − K0);
   Else
      序列 d(x)=NULL;
```

至此，针对 PUSCH 数据信道，所实施的**解速率匹配处理已经完成**，尽最大努力还原了 LDPC Channel Coding 所生成的序列，然后将其提供给 LDPC 信道译码模块，进行后续处理.

上述推导过程，基于如下研究工作:

[1] 分析和熟悉 LDPC 信道编码的过程，重点关注**输出序列的数据帧结构（图 A）**

[2] 分析和掌握 LDPC 数据信道速率匹配的过程（图 B）,进行逆向剖析.

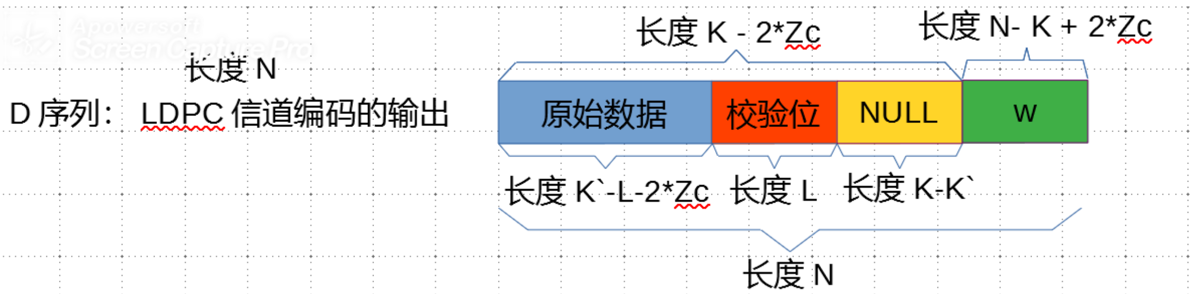


图 A

$k = 0$;

$j = 0$;

while $k < E$

  if $d_{(k_0+j) \bmod N_{cb}} \neq <NULL>$

    $e_k = d_{(k_0+j) \bmod N_{cb}}$;

    $k = k + 1$;

  end if

  $j = j + 1$;

end while

Table 5.4.2.1-2: Starting position of different redundancy versions, k_0

rv_{id}	k_0	
	LDPC base graph 1	LDPC base graph 2
0	0	0
1	$\left\lfloor \frac{17N_{cb}}{66Z_c} \right\rfloor Z_c$	$\left\lfloor \frac{13N_{cb}}{50Z_c} \right\rfloor Z_c$
2	$\left\lfloor \frac{33N_{cb}}{66Z_c} \right\rfloor Z_c$	$\left\lfloor \frac{25N_{cb}}{50Z_c} \right\rfloor Z_c$
3	$\left\lfloor \frac{56N_{cb}}{66Z_c} \right\rfloor Z_c$	$\left\lfloor \frac{43N_{cb}}{50Z_c} \right\rfloor Z_c$

图 B

仔细研究 LDPC Channel Coding 过程，可以看到有用数据里面，起始 2Zc Bits 的信息直接删除了，并没有映射至物理传输信道；但是没有关系，LDPC 信道译码阶段，被删除的信息可以得到恢复? //需要各位系统构架师和算法工程师给予审核和确认.

For a code block encoded by LDPC, the following encoding procedure applies:

1) Find the set with index i_{LS} in Table 5.3.2-1 which contains Z_c.

2) for $k = 2Z_c$ to $K-1$

  if $c_k \neq <NULL>$

    $d_{k-2Z_c} = c_k$;

  else

    $c_k = 0$;

    $d_{k-2Z_c} = <NULL>$;

  end if

end for