

RAN2#116bis

2022-01-17 – 2022-01-25

R2-22XXXXX



Title: How to configure RA partitions

Agenda item: 8.18 - RACH indication and partitioning

Source: Ericsson

Background



- This slide set is intended as information and to show how the signalling in the CR R2-22xxxx can be used by the gNB to configure RA partitioning.
- On the next slide the relevant ASN.1 is provided followed by a set of slides showing how the network would set the configuration to achieve different configurations.

Reference ASN.1



```
BWP-UplinkCommon ::=      SEQUENCE {
    rach-ConfigCommon SetupRelease { RACH-ConfigCommon }          OPTIONAL,    -- Need M
    ...OMITTED...

    [[
        additionalRach-ConfigCommonToAddModList-r17 SEQUENCE (SIZE(1..maxAdditionalRACH-r17)) OF AdditionalRACH-ConfigCommon-r17 OPTIONAL,    -- Cond SpCellOnly3

        additionalRach-ConfigCommonToReleaseList-r17 SEQUENCE (SIZE(1..maxAdditionalRACH-r17)) OF AdditionalRACH-ConfigIndex-r17          OPTIONAL    -- Cond SpCellOnly3
    ]]
}
```

```
RACH-ConfigCommon ::=      SEQUENCE {
    rach-ConfigGeneric      RACH-ConfigGeneric,
    totalNumberOfRA-Preambles INTEGER (1..63)          OPTIONAL,    -- Need S
    ssb-perRACH-OccasionAndCB-PreamblesPerSSB CHOICE {
        oneEighth      ENUMERATED {n4,n8,n12,n16,n20,n24,n28,n32,n36,n40,n44,n48,n52,n56,n60,n64},
        oneFourth      ENUMERATED {n4,n8,n12,n16,n20,n24,n28,n32,n36,n40,n44,n48,n52,n56,n60,n64},
        oneHalf        ENUMERATED {n4,n8,n12,n16,n20,n24,n28,n32,n36,n40,n44,n48,n52,n56,n60,n64},
        one            ENUMERATED {n4,n8,n12,n16,n20,n24,n28,n32,n36,n40,n44,n48,n52,n56,n60,n64},
        two            ENUMERATED {n4,n8,n12,n16,n20,n24,n28,n32},
        four           INTEGER (1..16),
        eight          INTEGER (1..8),
        sixteen        INTEGER (1..4)
    }          OPTIONAL,    -- Need M
    groupBconfigured      SEQUENCE {
        ra-Msg3SizeGroupA      ENUMERATED {b56, b144, b208, b256, b282, b480, b640,
                                           b800, b1000, b72, spare6, spare5, spare4, spare3, spare2, spare1},
        messagePowerOffsetGroupB ENUMERATED { minusinfinity, dB0, dB5, dB8, dB10, dB12, dB15, dB18},
        numberOfRA-PreamblesGroupA INTEGER (1..64) 22
    }          OPTIONAL,    -- Need R
    ...OMITTED...
    [[
        featureCombinationPreambles-r17 SEQUENCE (SIZE(1..FFS)) OF FeatureCombinationPreambles-r17          OPTIONAL    -- Need R
    ]]
}
AdditionalRACH-ConfigCommon-r17 ::= SEQUENCE {
    additionalRach-ConfigIndex-r17 AdditionalRACH-ConfigIndex,
    rach-ConfigCommon-r17          RACH-ConfigCommon          OPTIONAL,    -- Need R
    msgA-ConfigCommon-r17          MsgA-ConfigCommon-r16       OPTIONAL,    -- Cond R
    ...
}
AdditionalRACH-ConfigIndex-r17 ::= INTEGER(1..maxAdditionalRACH-r17)
```

```
FeatureCombinationPreambles-r17 ::=      SEQUENCE {
    featureCombination-r17      FeatureCombination-r17,
    startPreambleForThisPartition-r17 INTEGER (1..64),
    nrofPreamblesForThisPartition-r17 INTEGER (1..64),
    ssb-SharedRO-MaskIndex-r17   INTEGER (1..15),          OPTIONAL,    -- Need R
    numberOfRA-PreamblesGroupA  INTEGER (1..64)          OPTIONAL,
    featureSpecificParameters-r17 SEQUENCE {
        FFS
    }
}
```

```
RACH-ConfigCommonTwoStepRA-r16 ::=      SEQUENCE {
    rach-ConfigGenericTwoStepRA-r16      RACH-ConfigGenericTwoStepRA-r16,
    msgA-TotalNumberOfRA-Preambles-r16   INTEGER (1..63)          OPTIONAL,    -- Need S
    msgA-SSB-PerRACH-OccasionAndCB-PreamblesPerSSB-r16 CHOICE {
        oneEighth      ENUMERATED {n4,n8,n12,n16,n20,n24,n28,n32,n36,n40,n44,n48,n52,n56,n60,n64},
        oneFourth      ENUMERATED {n4,n8,n12,n16,n20,n24,n28,n32,n36,n40,n44,n48,n52,n56,n60,n64},
        oneHalf        ENUMERATED {n4,n8,n12,n16,n20,n24,n28,n32,n36,n40,n44,n48,n52,n56,n60,n64},
        one            ENUMERATED {n4,n8,n12,n16,n20,n24,n28,n32,n36,n40,n44,n48,n52,n56,n60,n64},
        two            ENUMERATED {n4,n8,n12,n16,n20,n24,n28,n32},
        four           INTEGER (1..16),
        eight          INTEGER (1..8),
        sixteen        INTEGER (1..4)
    }          OPTIONAL,    -- Cond 2StepOnly
    msgA-CB-PreamblesPerSSB-PerSharedRO-r16 INTEGER (1..60)      OPTIONAL,    -- Cond SharedRO
    msgA-SSB-SharedRO-MaskIndex-r16      INTEGER (1..15)        OPTIONAL,    -- Need S
    groupB-ConfiguredTwoStepRA-r16      GroupB-ConfiguredTwoStepRA-r16 OPTIONAL,    -- Need S
    ...OMITTED...

    [[
        featureCombinationPreambles-r17 SEQUENCE (SIZE(1..FFS)) OF FeatureCombinationPreambles-r17          OPTIONAL    -- Need R
    ]]
}
GroupB-ConfiguredTwoStepRA-r16 ::=      SEQUENCE {
    ra-MsgA-SizeGroupA      ENUMERATED {b56, b144, b208, b256, b282, b480, b640, b800,
                                           b1000, b72, spare6, spare5, spare4, spare3, spare2, spare1},
    messagePowerOffsetGroupB ENUMERATED { minusinfinity, dB0, dB5, dB8, dB10, dB12, dB15, dB18},
    numberOfRA-PreamblesGroupA INTEGER (1..64)
}
```

Additional PRACH – Separate RO – N
SSBs per RO



PRACH configuration

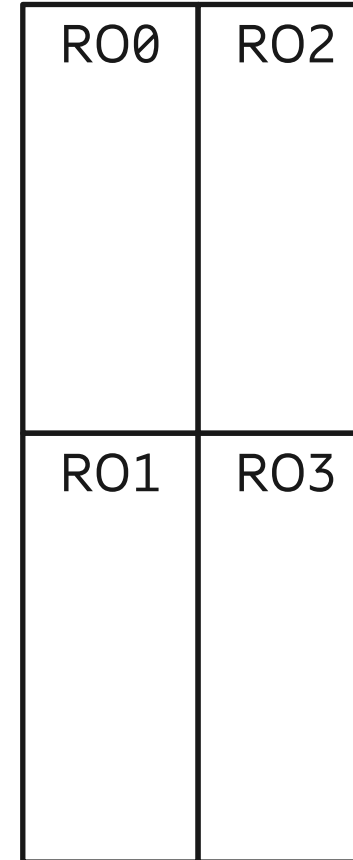


```
BWP-UplinkCommon ::= SEQUENCE {
  rach-ConfigCommon -> Legacy RACH, ignored in the example
  additionalRach-ConfigCommonToAddModList-r17 {
    additionalRach-ConfigIndex-r17 = 1,
    rach-ConfigCommon-r17 = {
      rach-ConfigGeneric => 2 x 2 ROs,
      totalNumberOfRA-Preambles = 60,
      ssb-perRACH-OccasionAndCB-PreamblesPerSSB = two + n8 (2 SSBs per RO and 8 CBPR per SSB)
      OPTIONAL, -- Need R
    }
    ...OMITTED...

    featureCombinationPreambles-r17 [
      {
        featureCombination-r17 = RedCap,
        startPreambleForThisPartition-r17 = 8,
        nrofPreamblesForThisPartition-r17 = 10,
        ssb-SharedRO-MaskIndex-r17 = All
        numberOfRA-PreamblesGroupA = 7
        FeatureSpecificParameters-r17 = FFS
      }, {
        featureCombination-r17 = RedCap + SmallData,
        startPreambleForThisPartition-r17 = 18,
        nrofPreamblesForThisPartition-r17 = 10,
        ssb-SharedRO-MaskIndex-r17 = Odd ROs
        numberOfRA-PreamblesGroupA = Not Defined (all preamb. are Group A)
        FeatureSpecificParameters-r17 = FFS
      }
    ]
  }
}

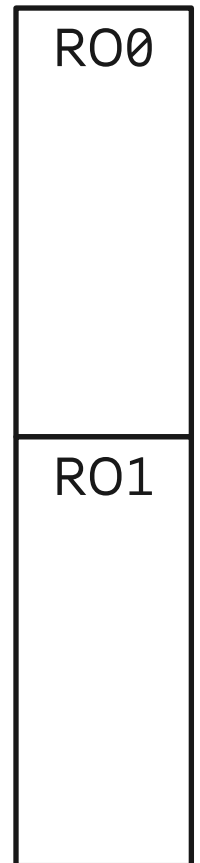
msgA-ConfigCommon-r17 = {
  rach-ConfigGenericTwoStepRA-r16 => 2 x 1 ROs
  msgA-TotalNumberOfRA-Preambles-r16 = 60,
  msgA-SSB-PerRACH-OccasionAndCB-PreamblesPerSSB-r16 = two + n4 (2 SSBs per RO and 4 CBPR per SSB)
  ...OMITTED...

  featureCombinationPreambles-r17 [
    {
      featureCombination-r17 = RedCap,
      startPreambleForThisPartition-r17 = 10,
      nrofPreamblesForThisPartition-r17 = 15,
      ssb-SharedRO-MaskIndex-r17 = All
      numberOfRA-PreamblesGroupA = Not Defined
      FeatureSpecificParameters-r17 = FFS
    }
  ]
}
}
```



PRACH index = 1
4-step RACH

PRACH index = 1
2-step RACH



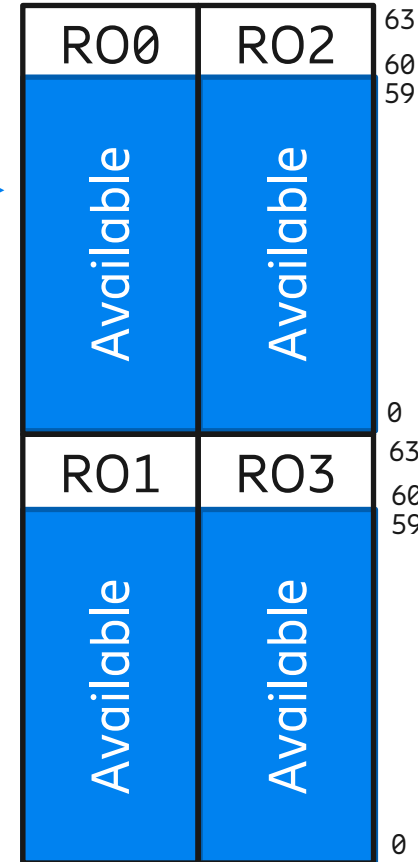
Available Preambles

```
BWP-UplinkCommon ::= SEQUENCE {
  rach-ConfigCommon -> Legacy RACH, ignored in the example
  additionalRach-ConfigCommonToAddModList-r17 {
    additionalRach-ConfigIndex-r17 = 1,
    rach-ConfigCommon-r17 = {
      rach-ConfigGeneric => 2 x 2 ROs,
      totalNumberOfRA-Preambles = 60,
      ssb-perRACH-OccasionAndCB-PreamblesPerSSB = two + n8 (2 SSBs per RO and 8 CBPR per SSB)
      OPTIONAL, -- Need R
    }
    ...OMITTED...
  }

  featureCombinationPreambles-r17 [
    {
      featureCombination-r17 = RedCap,
      startPreambleForThisPartition-r17 = 8,
      nrofPreamblesForThisPartition-r17 = 10,
      ssb-SharedRO-MaskIndex-r17 = All
      numberOfRA-PreamblesGroupA = 7
      FeatureSpecificParameters-r17 = FFS
    }, {
      featureCombination-r17 = RedCap + SmallData,
      startPreambleForThisPartition-r17 = 18,
      nrofPreamblesForThisPartition-r17 = 10,
      ssb-SharedRO-MaskIndex-r17 = Odd ROs
      numberOfRA-PreamblesGroupA = Not Defined (all preamb. are Group A)
      FeatureSpecificParameters-r17 = FFS
    }
  ]
}

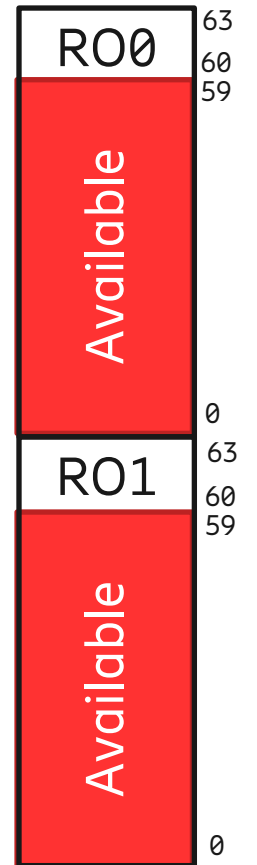
msgA-ConfigCommon-r17 = {
  rach-ConfigGenericTwoStepRA-r16 => 2 x 1 ROs
  msgA-TotalNumberOfRA-Preambles-r16 = 60,
  msgA-SSB-PerRACH-OccasionAndCB-PreamblesPerSSB-r16 = two + n4 (2 SSBs per RO and 4 CBPR per SSB)
  ...OMITTED...

  featureCombinationPreambles-r17 [
    {
      featureCombination-r17 = RedCap,
      startPreambleForThisPartition-r17 = 10,
      nrofPreamblesForThisPartition-r17 = 15,
      ssb-SharedRO-MaskIndex-r17 = All
      numberOfRA-PreamblesGroupA = Not Defined
      FeatureSpecificParameters-r17 = FFS
    }
  ]
}
```



PRACH index = 1
4-step RACH

PRACH index = 1
2-step RACH



SSB $\leftrightarrow$ preamble mapping

```

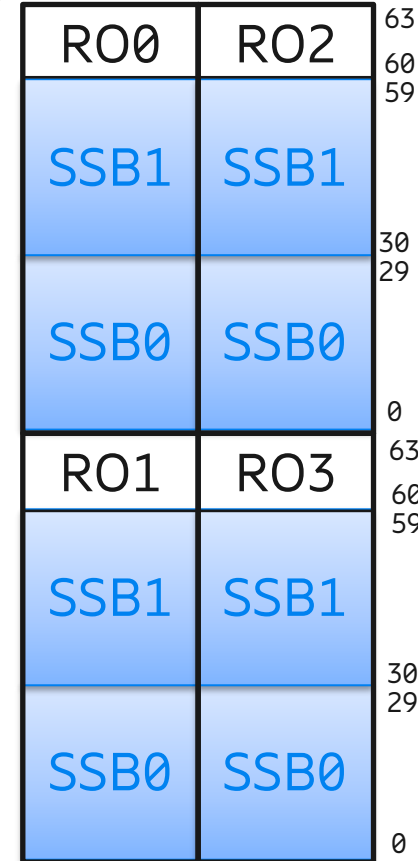
BWP-UplinkCommon ::= SEQUENCE {
  rach-ConfigCommon -> Legacy RACH, ignored in the example
  additionalRach-ConfigCommonToAddModList-r17 {
    additionalRach-ConfigIndex-r17 = 1,
    rach-ConfigCommon-r17 = {
      rach-ConfigGeneric => 2 x 2 ROs,
      totalNumberOfRA-Preambles = 60,
      ssb-perRACH-OccasionAndCB-PreamblesPerSSB = two + n8 (2 SSBs per RO and 8 CBPR per SSB)
      OPTIONAL, -- Need R
    }
    ...OMITTED...
  }

  featureCombinationPreambles-r17 [
    {
      featureCombination-r17 = RedCap,
      startPreambleForThisPartition-r17 = 8,
      nrofPreamblesForThisPartition-r17 = 10,
      ssb-SharedRO-MaskIndex-r17 = All
      numberOfRA-PreamblesGroupA = 7
      FeatureSpecificParameters-r17 = FFS
    }, {
      featureCombination-r17 = RedCap + SmallData,
      startPreambleForThisPartition-r17 = 18,
      nrofPreamblesForThisPartition-r17 = 10,
      ssb-SharedRO-MaskIndex-r17 = Odd ROs
      numberOfRA-PreamblesGroupA = Not Defined (all preamb. are Group A)
      FeatureSpecificParameters-r17 = FFS
    }
  ]

  msgA-ConfigCommon-r17 = {
    rach-ConfigGenericTwoStepRA-r16 => 2 x 1 ROs
    msgA-TotalNumberOfRA-Preambles-r16 = 60,
    msgA-SSB-PerRACH-OccasionAndCB-PreamblesPerSSB-r16 = two + n4 (2 SSBs per RO and 4 CBPR per SSB)
    ...OMITTED...
  }

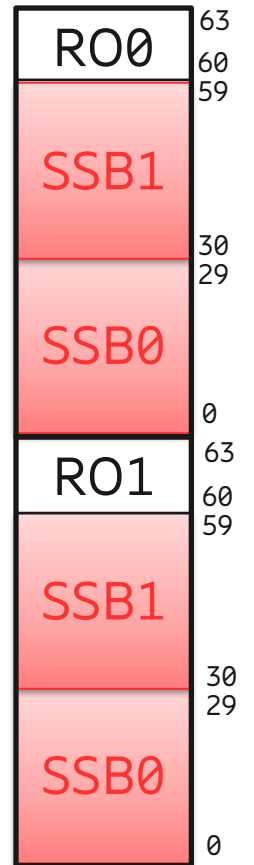
  featureCombinationPreambles-r17 [
    {
      featureCombination-r17 = RedCap,
      startPreambleForThisPartition-r17 = 10,
      nrofPreamblesForThisPartition-r17 = 15,
      ssb-SharedRO-MaskIndex-r17 = All
      numberOfRA-PreamblesGroupA = Not Defined
      FeatureSpecificParameters-r17 = FFS
    }
  ]
}

```



PRACH index = 1
4-step RACH

PRACH index = 1
2-step RACH



SSB $\leftrightarrow$ Legacy CBPR

```

BWP-UplinkCommon ::= SEQUENCE {
  rach-ConfigCommon -> Legacy RACH, ignored in the example
  additionalRach-ConfigCommonToAddModList-r17 {
    additionalRach-ConfigIndex-r17 = 1,
    rach-ConfigCommon-r17 = {
      rach-ConfigGeneric => 2 x 2 ROs,
      totalNumberOfRA-Preambles = 60,
      ssb-perRACH-OccasionAndCB-PreamblesPerSSB = two + n8 (2 SSBs per RO and 8 CBPR per SSB)
      OPTIONAL, -- Need R
    }
    ...OMITTED...
  }

  featureCombinationPreambles-r17 [
    {
      featureCombination-r17 = RedCap,
      startPreambleForThisPartition-r17 = 8,
      nrofPreamblesForThisPartition-r17 = 10,
      ssb-SharedRO-MaskIndex-r17 = All
      numberOfRA-PreamblesGroupA = 7
      FeatureSpecificParameters-r17 = FFS
    }, {
      featureCombination-r17 = RedCap + SmallData,
      startPreambleForThisPartition-r17 = 18,
      nrofPreamblesForThisPartition-r17 = 10,
      ssb-SharedRO-MaskIndex-r17 = Odd ROs
      numberOfRA-PreamblesGroupA = Not Defined (all preamb. are Group A)
      FeatureSpecificParameters-r17 = FFS
    }
  ]
}

msgA-ConfigCommon-r17 = {
  rach-ConfigGenericTwoStepRA-r16 => 2 x 1 ROs
  msgA-TotalNumberOfRA-Preambles-r16 = 60,
  msgA-SSB-PerRACH-OccasionAndCB-PreamblesPerSSB-r16 = two + n4 (2 SSBs per RO and 4 CBPR per SSB)
  ...OMITTED...

  featureCombinationPreambles-r17 [
    {
      featureCombination-r17 = RedCap,
      startPreambleForThisPartition-r17 = 10,
      nrofPreamblesForThisPartition-r17 = 15,
      ssb-SharedRO-MaskIndex-r17 = All
      numberOfRA-PreamblesGroupA = Not Defined
      FeatureSpecificParameters-r17 = FFS
    }
  ]
}

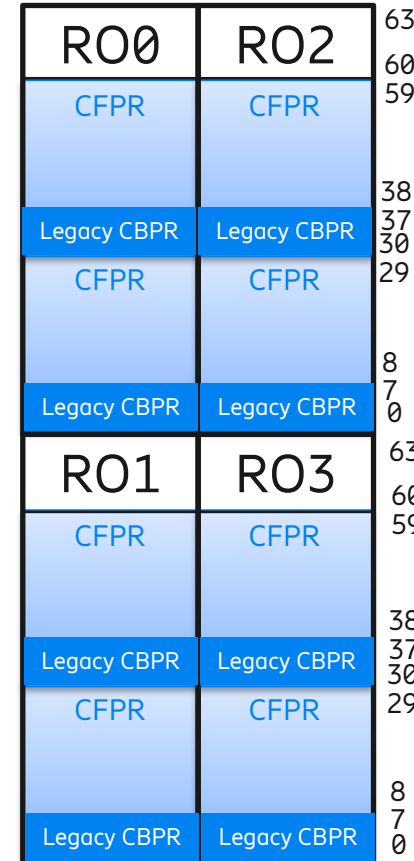
```

SSB1

SSB0

SSB1

SSB0



PRACH index = 1
4-step RACH



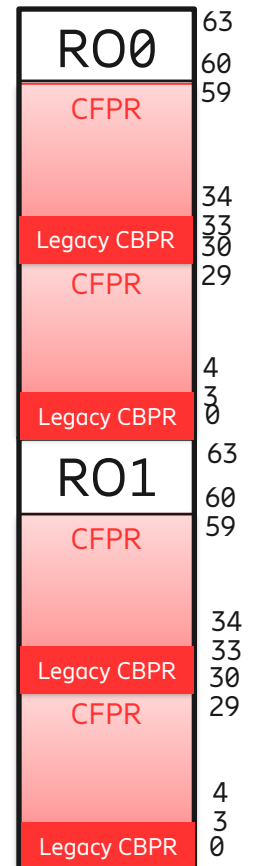
PRACH index = 1
2-step RACH

SSB1

SSB0

SSB1

SSB0



Applying Feature Combination Partitions

```

BWP-UplinkCommon ::= SEQUENCE {
    rach-ConfigCommon -> Legacy RACH, ignored in the example
    additionalRach-ConfigCommonToAddModList-r17 {
        additionalRach-ConfigIndex-r17 = 1,
        rach-ConfigCommon-r17 = {
            rach-ConfigGeneric => 2 x 2 ROs,
            totalNumberOfRA-Preambles = 60,
            ssb-perRACH-OccasionAndCB-PreamblesPerSSB = two + n8 (2 SSBs per RO and 8 CBPR per SSB)
            OPTIONAL, -- Need R
        }
        ...OMITTED...

    featureCombinationPreambles-r17 [
        {
            featureCombination-r17 = RedCap,
            startPreambleForThisPartition-r17 = 8,
            nrofPreamblesForThisPartition-r17 = 10,
            ssb-SharedRO-MaskIndex-r17 = All
            numberOfRA-PreamblesGroupA = 7
            FeatureSpecificParameters-r17 = FFS
        }, {
            featureCombination-r17 = RedCap + SmallData,
            startPreambleForThisPartition-r17 = 18,
            nrofPreamblesForThisPartition-r17 = 10,
            ssb-SharedRO-MaskIndex-r17 = RO2 Only
            numberOfRA-PreamblesGroupA = Not Defined (all preamb. are Group A)
            FeatureSpecificParameters-r17 = FFS
        }
    ]
}

msgA-ConfigCommon-r17 = {
    rach-ConfigGenericTwoStepRA-r16 => 2 x 1 ROs
    msgA-TotalNumberOfRA-Preambles-r16 = 60,
    msgA-SSB-PerRACH-OccasionAndCB-PreamblesPerSSB-r16 = two + n4 (2 SSBs per RO and 4 CBPR per SSB)
    ...OMITTED...

    featureCombinationPreambles-r17 [
        {
            featureCombination-r17 = RedCap,
            startPreambleForThisPartition-r17 = 10,
            nrofPreamblesForThisPartition-r17 = 15,
            ssb-SharedRO-MaskIndex-r17 = All
            numberOfRA-PreamblesGroupA = Not Defined
            FeatureSpecificParameters-r17 = FFS
        }
    ]
}
    
```

SSB1

SSB0

SSB1

SSB0

RO0	RO2	
CFPR	RC + SDT (18->27)	63 60 59
RedCap (38->47)	RedCap (38->47)	38
Legacy CBPR	Legacy CBPR	37 30
CFPR	RC + SDT (18->27)	29
RedCap (8->17)	RedCap (8->17)	8
Legacy CBPR	Legacy CBPR	7 0
RO1	RO3	
CFPR	CFPR	63 60 59
RedCap (38->47)	RedCap (38->47)	38
Legacy CBPR	Legacy CBPR	37 30
CFPR	CFPR	29
RedCap (8->17)	RedCap (8->17)	8
Legacy CBPR	Legacy CBPR	7 0

PRACH index = 1
4-step RACH

PRACH index = 1
2-step RACH

SSB1

SSB0

SSB1

SSB0

RO0	
CFPR	63 60 59
RedCap (10->24)	34
Legacy CBPR	33 30
CFPR	29
RedCap (10->24)	4
Legacy CBPR	3 0
RO1	
CFPR	63 60 59
RedCap (10->24)	34
Legacy CBPR	33 30
CFPR	29
RedCap (10->24)	4
Legacy CBPR	3 0

Applying Group A / B

```

BWP-UplinkCommon ::= SEQUENCE {
    rach-ConfigCommon -> Legacy RACH, ignored in the example
    additionalRach-ConfigCommonToAddModList-r17 {
        additionalRach-ConfigIndex-r17 = 1,
        rach-ConfigCommon-r17 = {
            rach-ConfigGeneric => 2 x 2 ROs,
            totalNumberOfRA-Preambles = 60,
            ssb-perRACH-OccasionAndCB-PreamblesPerSSB = two + n8 (2 SSBs per RO and 8 CBPR per SSB)
            OPTIONAL, -- Need R
        }
        ...OMITTED...

    featureCombinationPreambles-r17 [
        {
            featureCombination-r17 = RedCap,
            startPreambleForThisPartition-r17 = 8,
            nrofPreamblesForThisPartition-r17 = 10,
            ssb-SharedRO-MaskIndex-r17 = All
            numberOfRA-PreamblesGroupA = 7
            FeatureSpecificParameters-r17 = FFS
        }, {
            featureCombination-r17 = RedCap + SmallData,
            startPreambleForThisPartition-r17 = 18,
            nrofPreamblesForThisPartition-r17 = 10,
            ssb-SharedRO-MaskIndex-r17 = RO2 Only
            numberOfRA-PreamblesGroupA = Not Defined (all preamb. are Group A)
            FeatureSpecificParameters-r17 = FFS
        }
    ]
}

msgA-ConfigCommon-r17 = {
    rach-ConfigGenericTwoStepRA-r16 => 2 x 1 ROs
    msgA-TotalNumberOfRA-Preambles-r16 = 60,
    msgA-SSB-PerRACH-OccasionAndCB-PreamblesPerSSB-r16 = two + n4 (2 SSBs per RO and 4 CBPR per SSB)
    ...OMITTED...

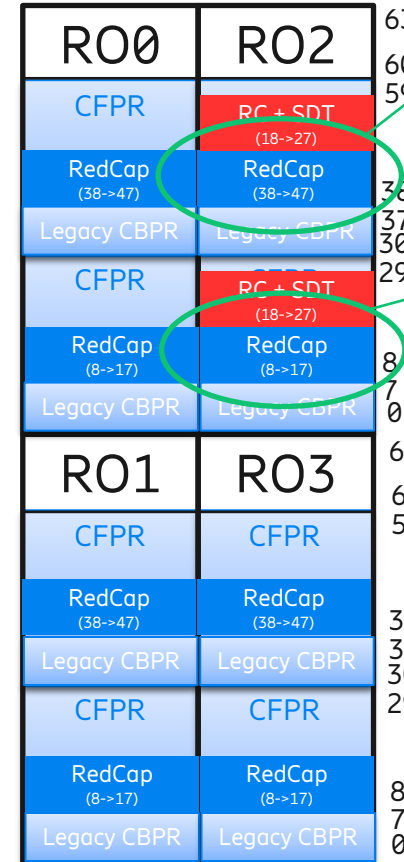
    featureCombinationPreambles-r17 [
        {
            featureCombination-r17 = RedCap,
            startPreambleForThisPartition-r17 = 10,
            nrofPreamblesForThisPartition-r17 = 15,
            ssb-SharedRO-MaskIndex-r17 = All
            numberOfRA-PreamblesGroupA = Not Defined
            FeatureSpecificParameters-r17 = FFS
        }
    ]
}
    
```

SSB1

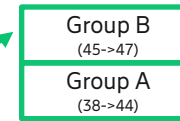
SSB0

SSB1

SSB0



PRACH index = 1
4-step RACH



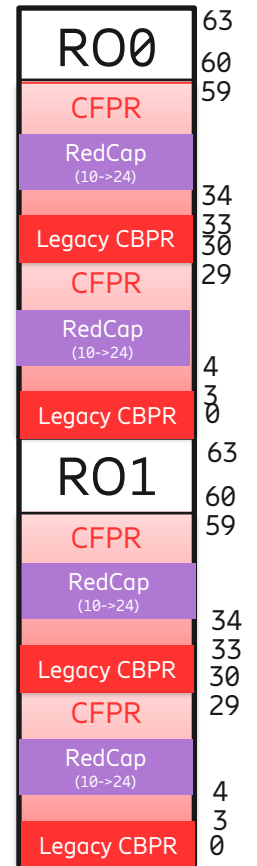
PRACH index = 1
2-step RACH

SSB1

SSB0

SSB1

SSB0



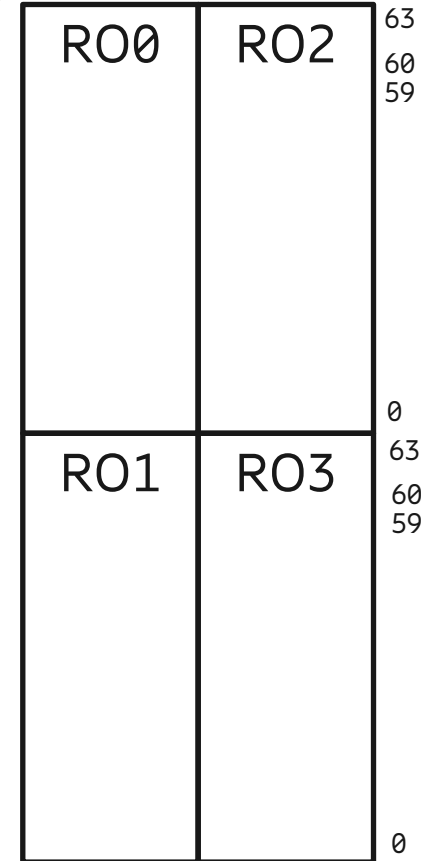
SSB $\leftrightarrow$ preamble mapping

```
BWP-UplinkCommon ::= SEQUENCE {
  rach-ConfigCommon -> Legacy RACH, ignored in the example
  additionalRach-ConfigCommonToAddModList-r17 {
    additionalRach-ConfigIndex-r17 = 1,
    rach-ConfigCommon-r17 = {
      rach-ConfigGeneric => 2 x 2 ROs,
      totalNumberOfRA-Preambles = 60,
      ssb-perRACH-OccasionAndCB-PreamblesPerSSB = two + n8 (2 SSBs per RO and 8 CBPR per SSB)
      OPTIONAL, -- Need R
    }
    ...OMITTED...
  }

  featureCombinationPreambles-r17 [
    {
      featureCombination-r17 = RedCap,
      startPreambleForThisPartition-r17 = 8,
      nrofPreamblesForThisPartition-r17 = 10,
      ssb-SharedRO-MaskIndex-r17 = All
      numberOfRA-PreamblesGroupA = 7
      FeatureSpecificParameters-r17 = FFS
    }, {
      featureCombination-r17 = RedCap + SmallData,
      startPreambleForThisPartition-r17 = 18,
      nrofPreamblesForThisPartition-r17 = 10,
      ssb-SharedRO-MaskIndex-r17 = Odd ROs
      numberOfRA-PreamblesGroupA = Not Defined (all preamb. are Group A)
      FeatureSpecificParameters-r17 = FFS
    }
  ]
}

msgA-ConfigCommon-r17 = {
  rach-ConfigGenericTwoStepRA-r16 => 2 x 1 ROs
  msgA-TotalNumberOfRA-Preambles-r16 = 60,
  msgA-SSB-PerRACH-OccasionAndCB-PreamblesPerSSB-r16 = two + n4 (2 SSBs per RO and 4 CBPR per SSB)
  ...OMITTED...

  featureCombinationPreambles-r17 [
    {
      featureCombination-r17 = RedCap,
      startPreambleForThisPartition-r17 = 10,
      nrofPreamblesForThisPartition-r17 = 15,
      ssb-SharedRO-MaskIndex-r17 = All
      numberOfRA-PreamblesGroupA = Not Defined
      FeatureSpecificParameters-r17 = FFS
    }
  ]
}
```



PRACH index = 1
4-step RACH

PRACH index = 1
2-step RACH

