

Symbol Mapper Overview

The details here are for expression of the operation of the standard and do not imply any specific implementation.

The US symbol mapper consists of an idle loop process and a fill process. The idle loop process is initialized when the upstream profile is configured and set to align with the start of the superframe. The main task of the idle loop consists of two tasks 1) walk through all data bits in all data carrying resource elements at the clock rate established by *US_DataRate* (see 100.x.x.x.), and 2) upon an assertion of the start of a burst from the PMA Client, launch the fill process using the resource block that is at the current walk point. This includes placing the start burst marker elements, filling bits in data resource elements and low density pilots as specified, creating the QAM map for the fill word bits, sensing the endBurst notification, padding to the end of the current resource block, encoding and placing the end burst marker elements and then terminating. The idle loop is a continuous loop. The fill process is meant to be launched and then terminate after the end marker is placed. During the idle walk and fill processes, two-dimensional arrays (resource block size by total subcarriers) termed RB Frames are allocated and passed to the staging process when complete. RB Frames initially consist of null values. Non-null values are inserted by the fill process when mapping burst markers and filling data resource elements, and when later processed by the Pilot Insert process fill "P" type pilots patterns (see 101.x.x.x.). Null resource blocks produce no energy on the corresponding OFDMA subcarrier after the IDFT process.

The variables and functions in this subsection are shared between the idle process and the fill process.

Variables

RB_Size

Type: integer

This variable represents the current setting of the Resource Block size for this Superframe (profile) configuration. The value is either 8 or 16. RB_Size also corresponds to the interleaver depth.

RB_Frame<4095:0, RB_Size:1>

Type: Array of I_value and Q_value bin value pairs

This two-dimensional array holds the I and Q QAM symbol bin values that are passed from the symbol mapper to the staging function, then to the IDFT for transmission by the CNU. When first allocated, RB_Frame contains null values ("0") in all I and Q values. I and Q bin values are signed 16-bit integers. Index 4095:0 represents the total number of possible subcarriers in the upstream corresponding OFDMA channel, RB_Size:1 represents the number of resource elements in resource block. Subcarrier 0, RB number 1 (RB_Frame<0,1>) is reserved for special use in coordinating symbol mapper idle and fill processing: I_value of "0" indicates the RB_Frame has not transferred to the staging process, I_value of "1" indicates the RB_Frame has been passed to the staging process, Q_value is always "0".

RB_Type<4095:0>

Type: array of enumerated values

This array defines the use of each resource block, the values and descriptions are:

“Null”: the resource block is unused, all resource elements will contain null I and Q value pairs.

“T0”: resource block in use, Type 0 pilot pattern (see 101.4.4.7)

“T1”: resource block in use, Type 1 pilot pattern (see 101.4.4.7)

“T2”: resource block in use, Type 2 pilot pattern (see 101.4.4.7)

“SM”: resource block in use, contains a start marker pattern

“EM”: resource block in use, contains an end marker pattern

All other enumerated type values are reserved. The use of this array indicates the availability and type of each used resource block to the interleaver. A null resource block array element produces no energy output from the IDFT for the corresponding OFDMA subcarrier.

PILOT_MAP<4095:0>

Type: array of enumerate values

This array defines the pilot pattern use for Type 1 and Type 2 pilot patterns as defined in the upstream profile descriptor (see section 101.x.x.x). The pilot pattern is fixed for the Superframe configuration and remains constant. The enumerated type values are:

“Null”: no Type 1 or Type 2 pattern is configured for this subcarrier

“T0”: this resource block, if used for data, will contains a Type 0 pattern (see 101.4.4.7)

“T1”: this resource block, if used for data, will contains a Type 1 pattern (see 101.4.4.7)

“T2”: this resource block, if used for data, will contains a Type 2 pattern(see 101.4.4.7)

“PHYLINK”: this resource block is reserved for use by the PHY Link

All other enumerated type values are reserved. For any used resource element containing data and not containing either a start or end burst marker, this value will be used to set the RB_Type array element value for further processing by the interleaver and pilot insertion functions.

IRB

Type: integer

This variable is used to determine the current resource block being processed in the idle process. The range of values if from -1 to 4095.

IRE

Type: integer

This variable is used to determine the current resource element being processed in the idle process. The range of values if from 0 to RB_Size.

FILL_STATE:

Type: enumerated

This variable is used to coordinate filling actions by the symbol mapper filling process. The enumerated types are:

“IDLE”: the symbol mapper is in the idle loop

“FILL”: a start of burst indication has been presented via the PMA service interface and the symbol mapper is in the fill process.

“end”: an end of burst indication has been presented via the PMA service interface and the symbol mapper is padding any remain bits to the end of the current resource block and preparing to place an encoded end burst marker.

LBIT

Type: integer

This variable records the last bit filled in the current data resource element (QAM symbol) before mapping. The value can be from 1 to 16, where 1 represents the LSB. The value is set to the bit loading for the data RE and then decremented during the symbol mapper fill process for mapping data burst bits to the current data resource element. This value is reset for each new data resource element being filled. The value is incremented while filling bits and stops incrementing upon receiving an end of burst indication. This value is not incremented or reset when placing padding bits.

LRE

Type: integer

This variable records the last filled data resource element in the current resource block. The value can be from 1 to 16, where 1 represents the first resource element in the resource block (in time) and up to 8 or 16 representing the last resource element in the resource block (later in time). The value is reset for each new resource block being filled. The value is incremented to the current data resource element being filled and stops incrementing upon receiving an end of burst indication. This value is not incremented or reset when placing padding bits.

IDLEBITS

Type: integer

This variable is used by the idle loop to increment through the bit loading of the current data carry resource element.

ICLK

Type: clock

This variable represents a clock running at *US_DataRate* (see 100.x.x.x).

ASSERTPMA

Type: boolean

This variable a signal that the PMA client has asserted a PMA_UNITDATA.request for processing by the PMA.

FRB

Type: integer

This variable is used to determine the current resource block being processed in the fill process. The range of values if from 0 to 4095.

FRE

Type: integer

This variable is used to determine the current resource element being processed in the fill process. The range of values if from -1 to RB_Size.

FIRST

Type: boolean

This Boolean is used within the fill process to indicate status of placing the firstbit passed in the function call. If TRUE, the passed firstbit is used, otherwise the bit receive from the processed PMA_UNITDATA.request() is used.

END

Type: enumerated type

This variable that ends the end state processing of the fill process, where "FALSE" indicates a burstStart = TRUE has not been received from received PMA_UNITDATA.request() primitive.

"PAD" indicates burstStart = TRUE has been received and the fill process is padding bits to the end of the current RB

"TRUE" indicates that the fill process has reached the end of the last RB while padding.

FILLWORD<16:1>

Type: integer array

This variable is a bit array that stages the bits to fill a resource element prior to mapping to QAM symbols.

BITPOS

Type: integer

This is an integer used by the fill process to indicate the current bit position being filled in FILLWORD<>.

Functions

initialize_Pilot_Map()

This function initializes the PILOT_MAP<> array based on the profile descriptor information for placing Type 1 and Type 2 patterns. See Table 101-1 and 101.4.4.7.

allocate_RB_Frame(RB_Frame, RB_Type)

This function allocates a new RB_Frame array with all I_value and Q_value bin pairs set to null "0" and a new RB_Type array with all enumerations set to "Null".

next_RE(resource_element, resource_block)

This function increments to the next usable data carry resource element starting from the passed resource_element number in the passed resource_block. This includes data carry resource elements and low density pilots, pilot resource elements are skipped. See 101.4.4.7. The resource element is returned and may have a value from 1 to RB_Size. If resource_element has a value of -1, this function increments to the next resource block and begins with the first useable resource element .

This function will first increment to the next usable resource element in the current resource block. If necessary, this function will increment to the next data carrying resource block (e.g. Type 0, 1, or 2). The resource block number (subcarrier index) from 0 to 4095 is returned in resource_block. For this function, let "RB" be the resource block number.

When advancing to the next RB and FILL_STATE = FILL, the value from Pilot_Map<resource block> sets the value of RB_Type<resource block>.

If at any time this function increments beyond the last usable resource block in the current RB_Frame (highest usable subcarrier configured in the profile descriptor), It examines the I_value of RB_Frame<0,1>. If "0", this function sets the value to "1" and passes the RB_Frame array and RB_Type array to the staging function and allocates a new RB_Frame array and RB_Type array (allocate_RB_Frame(), RB_Type()). If I_value of RB_Frame<0,1> is "1", skips passing to staging and allocations, assume a new RB_Frame has been allocated, and increments to the first data carrying resource block and resource element in the new RB Frame.

Upon return, the updated resource element index is returned via resource_element, and the current resource block index returned via resource_block.

BITLOAD(resource_element)

This function returns the current bit loading capacity of the current data carrying resource element or low density pilot.

Launch “process”

This function causes the indicated process to begin execution. This function returns immediately.

FILL_PROCESS(FIRSTRB, FIRSTBIT)

This function calls the fill process to begin execution, at the resource block indicated by FIRSTRB index, with the FIRSTBIT indicating the first bit to be filled in MSB of the first available data resource element. See Figure 101.x.x.x.

Map_Start_Marker (RB_number)

This function begins by placing the first resource block of a start burst marker in the current resource block RB_number, according to the RB_Size of the Superframe (see 101.x.x.x). This function continues to increment resource blocks (subcarriers) and placing the remaining start burst marker elements in the next usable resource block(s) as indicated in US modulation type setting in the US profile descriptor (defined as a US modulation type value from binary 0001 (BPSK) to binary 1110 (16384-QAM) and not being used by the PHY Link). See 45.2.7a.2 and Table 45–191c. Excluded subcarriers and null subcarriers are skipped. After placing the last start burst marker resource block, this function returns the value of the current start marker in RB_number.

For each resource block used for a start burst marker element, the corresponding entry in the RB_Type<RB_number> array is set to “SM”. The next usable resource block is defined as a US modulation type value from binary 0001 (BPSK) to binary 1110 (16384-QAM) and not being used by the PHY Link.

If at any time this function increments beyond the last usable resource block in the current RB_Frame (highest usable subcarrier configured in the profile descriptor), It examines the I_value of RB_Frame<0,1>. If “0”, this function sets the value to “1” and passes the RB_Frame array and RB_Type array to the staging function and allocates a new RB_Frame array and RB_Type array (allocate_RB_Frame(), RB_Type()). If I_value of RB_Frame<0,1> is “1”, skips passing to staging and allocations, assume a new RB_Frame has been allocated, and increments to the first data carrying resource block and resource element in the new RB Frame.

reset_scrambler()

This function initializes the bit scrambler with the seed value. See 101.4.3.6.4.

scramble(bit)

This function provides a bit scrambler, local to the upstream data symbol mapper function. The passed bit is used as input to the scrambler, the output is used as the return value. See 101.4.3.6.4.

Map_End_Marker (RB_number, Last_Bit, Last_RE)

This function starts at the current RB_number. This function then constructs the end burst marker by encoding the Last_Bit and Last_RE information. See 101.x.x.x and 101.x.x.x for end marker encoding. This function then place an end burst marker elements in the resource block according to the RB_Size of the Superframe. This function continues to increment resource blocks (subcarriers) and placing the remaining end burst marker elements in the next usable resource block as indicated in US modulation type setting in the US profile descriptor. See 45.2.7a.2 and Table 45–191c. For each resource block used for an end burst marker element, the corresponding entry in the RB_Type array is set to “EM”. Excluded subcarriers and null subcarriers are skipped. The next usable resource block is defined as a US modulation type value from binary 0001 (BPSK) to binary 1110 (16384-QAM) and not being used by the PHY Link.

If at any time this function increments beyond the last usable resource block in the current RB_Frame (highest usable subcarrier configured in the profile descriptor), It examines the I_value of RB_Frame<0,1>. If “0”, this function sets the value to “1” and passes the RB_Frame array and RB_Type array to the staging function and allocates a new RB_Frame array and RB_Type array (allocate_RB_Frame(), RB_Type()). If I_value of RB_Frame<0,1> is “1”, skips passing to staging and allocations, assume a new RB_Frame has been allocated, and increments to the first data carrying resource block and resource element in the new RB Frame.

stage_RB_Frame(RB_Frame, RB_Type)

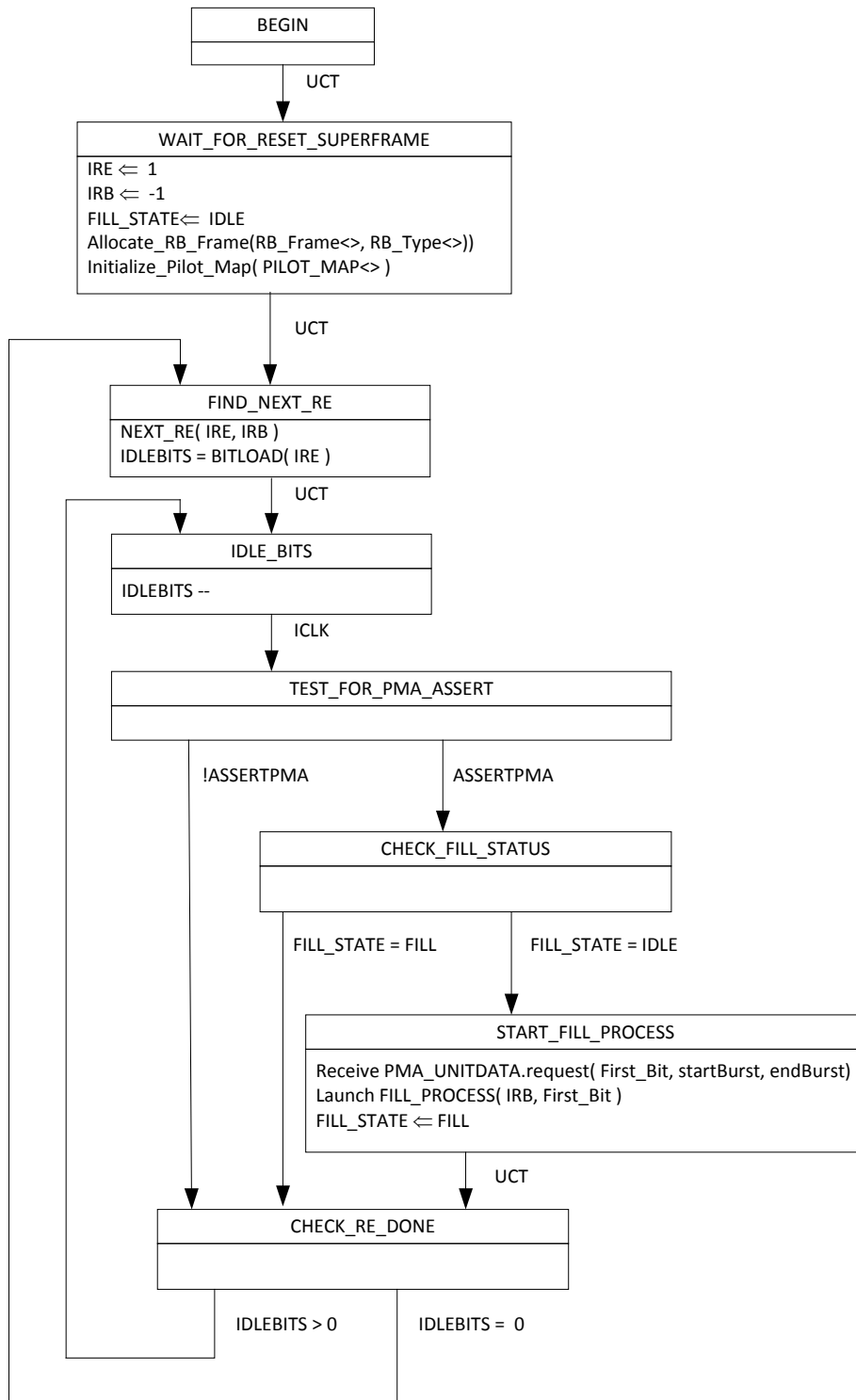
This function examines the I_value of RB_Frame<0,1>. If “0”, this function sets the value to “1” and passes the RB_Frame array and RB_Type array to the staging function. If I_value of RB_Frame<0,1> is “1”, this function simple returns.

Map_to_QAM(resource_block, resource_element , FILLWORD<>, FILLBITS)

This function maps the bits in FILLWORD<> into the I and Q bin value pairs for the current resource element in the current resource_block in RB_Frame<> (see 101.4.5). FILLBITS represents the bitloading of the current data resource element or low density pilot, where FILLWORD<FILLBITS> represents the MSG and FILLWORD<1> represents the LSB for the mapping.

IF RB_Type<resource_block> is NULL, this functions sets RB_Type<resource_block> to PILOT_MAP<resource_block> to indicate this resource block contains non-null values.

Idle Loop State Diagram



Fill Process State Diagram

