RX OFDM

Variable **fft_len** **64**

Variable **samp_rate** **2000000**

Variable **length_tag_key** **"frame_len"**

Variable **packet_length_tag_key** **"packet_len"**

Variable **packet_len** **96**

Variable **header_mod** **digital.constellation_bpsk()**

Variable **payload_mod** **digital.constellation_qpsk()**

Variable **occupied_carriers** **(list(range(-26, -21)) + list(range(-20, -7)) + list(range(-6, 0)) + list(range(1, 7)) + list(range(8, 21)) + list(range(22, 27)),)**

Variable **pilot_carriers** **((-21, -7, 7, 21,),)**

Variable **pilot_symbols** **((1, 1, 1, -1,),)**

Variable **header_formatter** **digital.packet_header_ofdm(occupied_carriers, n_syms=1, len_tag_key=packet_length_tag_key, frame_len_tag_key=length_tag_key, bits_per_header_sym=header_mod.bits_per_symbol(), bits_per_payload_sym=payload_mod.bits_per_symbol(), scramble_header=False)**

Variable **sync_word1** **[0., 0., 0., 0., 0., 0., 0., 1.41421356, 0., -1.41421356, 0., 1.41421356, 0., -1.41421356, 0., -1.41421356, 0., -1.41421356, 0., 1.41421356, 0., -1.41421356, 0., 1.41421356, 0., -1.41421356, 0., -1.41421356, 0., -1.41421356, 0., -1.41421356, 0., 1.41421356, 0., -1.41421356, 0., 1.41421356, 0., 1.41421356, 0., 1.41421356, 0., -1.41421356, 0., 1.41421356, 0., 1.41421356, 0., 1.41421356, 0., -1.41421356, 0., 1.41421356, 0., 1.41421356, 0., 1.41421356, 0., 0., 0., 0., 0., 0.]**

Variable **sync_word2** **[0j, 0j, 0j, 0j, 0j, 0j, (-1+0j), (-1+0j), (-1+0j), (-1+0j), (1+0j), (1+0j), (-1+0j), (-1+0j), (-1+0j), (1+0j), (-1+0j), (1+0j), (1+0j), (1 +0j), (1+0j), (1+0j), (-1+0j), (-1+0j), (-1+0j), (-1+0j), (-1+0j), (1+0j), (-1+0j), (-1+0j), (1+0j), (-1+0j), 0j, (1+0j), (-1+0j), (1+0j), (1+0j), (1+0j), (-1+0j), (1+0j), (1+0j), (1+0j), (-1+0j), (1+0j), (1+0j), (1+0j), (1+0j), (-1+0j), (1+0j), (-1+0j), (-1+0j), (-1+0j), (1+0j), (-1+0j), (1+0j), (-1+0j), (-1+0j), (-1+0j), (-1+0j), 0j, 0j, 0j, 0j, 0j]**

Variable **header_equalizer** **digital.ofdm_equalizer_simpledfe(fft_len, header_mod.base(), occupied_carriers, pilot_carriers, pilot_symbols)**

Variable **payload_equalizer** **digital.ofdm_equalizer_simpledfe(fft_len, payload_mod.base(), occupied_carriers, pilot_carriers, pilot_symbols, 1)**

Throttle block:

Type: **Complex**

Sample Rate **samp_rate**

Vec Length **1**

Schmidl & Cox OFDM synch block:

FFT length:	**fft_len**
Cyclic Prefix length:	**fft_len//4**
Preamble Carrriers:	**Odd**

Delay block:

Delay:	**fft_len+fft_len//4**

Frequency Mod block**:**

Sensitivity:	**-2.0/fft_len**

Header/Payload Demux block:

Header Length (Symbols):	**3**
Items per symbol:	**fft_len**
Guard Interval (items):	**fft_len//4**
Length tag key:	**length_tag_key**
Output Format:	**Symbols**
Timing Tag key:	**“rx_time”**
Sampling Rate:	**samp_rate**
Special Tag Keys:	**()**

FFT blocks**:**

Input Type:	**Complex**
FFT Size:	**fft_len**
Forward/Reverse:	**Forward**
Window:	**()**
Shift:	**Yes**
Num. Threads:	**1**

OFDM Channel Estimation block:

Synch. symbol 1:	**sync_word1**
Synch. symbol 2:	**sync_word2**
Number of data symbols	**1**
Maximum carrier offset	**3**
Force One Synchr.	**No**

OFDM Frame Equalizer block (Header):

FFT length:	**fft_len**
CP length:	**fft_len//4**
Equalizer:	**header_equalizer.base()**
Length Tag Key:	**length_tag_key**
Propagate Channel State:	**Yes**
Fixed frame length:	**1**

OFDM Frame Equalizer block (Payload):

FFT length:	**fft_len**
CP length:	**fft_len//4**
Equalizer:	**payload_equalizer.base()**
Length Tag Key:	**length_tag_key**
Propagate Channel State:	**Yes**
Fixed frame length:	**0**

OFDM Serializer block (Header):

FFT length:	**fft_len**
Occupied Carriers:	**occupied_carriers**
Length Tag Key:	**length_tag_key**
Symbols skipped:	**0**
Input is shifted	**True**

OFDM Serializer block (Payload):

FFT length:	**fft_len**
Occupied Carriers:	**occupied_carriers**
Length Tag Key:	**length_tag_key**
Packet Length Tag Key:	**packet_length_tag_key**
Symbols skipped:	**1**
Input is shifted	**True**

Constelation Decoder (Header):

Constelation Object: **header_mod.base()**

Constelation Decoder (Payload):

Constelation Object: **payload_mod.base()**

Packet Header Parser block:

Formatter Object: **header_formatter.base()**

Repack Bits block:

Bits per input byte: **payload_mod.bits_per_symbol()**

Bits per output byte: **8**

Length Tag Key: **packet_length_tag_key**

Packet Alignment: **Output**

Endianness: **LSB**

Stream CRC32 block:

Mode **Check CRC**

Length tag name **packet_length_tag_key**

Packet **Yes**