Model-Based Design SPT 2.0/2.5 Toolbox User Guide

Version 2.0
Date 1 Sept 2018

Table of Contents

Version Tracking

Date	Version	Comments	Author
7 July 2017	1.0	Initial Version	Razvan Ionescu
1 Sept 2018	2.0	Update for SPT 2.5	Iulian Bulancea

1 Overview

1.1 Scope

SPT Toolbox provides a support for easier integration of SPT Simulator into MATLAB environment.

1.2 SPT 2.0/2.5 Overview

SPT 2.0/2.5 (Signal Processing Toolbox) is a hardware accelerator designed for processing radar signals. One block inside SPT is the FFT core. The FFT core has 2 Radix4 operational blocks to perform Fast Fourier Transform (FFT), Inverse FFT, Scalar Product (SCP), Finite Impulse Response Filtering (FIR) and Windowing (WIN) operations.
SPT supports multiple instructions in the form of commands. Commands can be split in 2 main categories: operation and control.
RDX4 is an operation command.

1.3 Toolbox

SPT Toolbox is a collection of MEX files that provides from MATLAB environment to easy access SPT 2.0, 2.5, and 3.0 Simulator functionality. MEX files are MATLAB binaries that can be accessed from MATLAB environment similar with built-in functions.
For each operation of SPT 2.0 and 2.5 supported by Toolbox, exists a separate MEX file and an example on how to use it.

2 SPT Toolbox API

2.1 SWITCH

sptVersionOut = **switch_mex**(sptVersionIn);

- sptVersionOut - The current SPT version
- sptVersionIn - The desired SPT version; the possible values are 20 and 25; the input can miss

2.1.1 Example

```
%using this mex function one can find out the current SPT version
oldSptVersion = switch_mex;
%also he can change the SPT version
newSptVersion = switch_mex(25);
```

2.2 ADD

M_Out = **add_mex**(op_1, op_2, shift, modulo_val);

- M_Out - The output is a complex scalar
- op_1 - First operand value to be added. It must be complex scalar.
- op_2 - Second operand value to be added. It must be complex scalar.
- shift - Indicates whether output is left shifted by 1-bit or not; Valid values are: 0, 1.
- modulo_val - Modulo value; Valid values are: 0, 1, 2, …, 24, which means a modulo operation with 2^0, 2^1, 2^2, …, 2^24

2.2.1 Mapping with SPT 2.0/2.5 ADD command parameters

You can find more information about ADD command in SPT Reference Manual.

SPT ADD command parameters	add_mex
OPCODE	none
SRC	none
SHIFT	shift
MODULO_VAL	modulo_val
SRC1_ADD	none
DEST_ADD	none
SRC2_ADD	none
IMM_DAT	none

2.2.2 Example

shift = 0;
modulo_val = 20;
op_1 = 23+2i;
op_2 = 10-9i;
M_Out = **add_mex**(op_1, op_2, shift, modulo_val);

2.3 SUB

Out = **sub_mex**(op_1, op_2, shift, modulo_val);

- M_Out - The output is a complex scalar
- op_1 - First operand value from which it is subtracted operand 2. It must be complex scalar.
- op_2 - Second operand value to be substracted from operand 1. It must be complex scalar.
- shift - Indicates whether output is left shifted by 1-bit or not; Valid values are: 0, 1.
- modulo_val - Modulo value; Valid values are: 0, 1, 2, …, 24, which means a modulo operation with 2^0, 2^1, 2^2, …, 2^24

2.3.1 Mapping with SPT 2.0/2.5 SUB command parameters

You can find more information about SUB command in SPT Reference Manual.

SPT SUB command parameters	sub_mex
OPCODE	none
SRC	none
SHIFT	shift
MODULO_VAL	modulo_val
SRC1_ADD	none
DEST_ADD	none
SRC2_ADD	none
IMM_DAT	none

2.3.2 Example

shift = 0;
modulo_val = 24;

op_1 = 20+2i;
op_2 = -4+3i;
M_Out = **sub_mex**(op_1, op_2, shift, modulo_val);

2.4 RDX2

M_Out = **rdx2_mex**(M_Tw, M_Op, fft_rnd, real_fft, quad_ext, repeat, shft_val);

- M_Out - The output is a complex vector that has the same size as the operand vector
- M_Tw - Twiddle values must be a complex vector
    - **NOTE**: *Maximum size of twiddles data is 4096 elements. (Size of SPT TWRAM is 0x1000)*
    - **NOTE**: *Twiddles values are expected in regular format. SPT Toolbox will take care to properly load them into SPT simulator memory (repeat each value 7 times, resulting in 8 similar values for each twiddle).*

- M_Op - Operands values must be a complex vector
    - **NOTE**: *Maximum size of input data is 16384 elements. (Size of SPT OPRAM is 0x4000)*

- fft_rnd - FFT Round; Valid values are: 1, 2, 3, 4, 5.
- real_fft - RDX2 command can be used for splitting Operands corresponding to 2 Real input vectors.
    - Valid values are:
        - 'NO_OPERANDS_SPLIT' - Normal Radix2 Command
        - 'OPERANDS_SPLIT' - Radix2 command used for operands split
- quad_ext - Quadrature Extension;
    - Valid values are:
        - 'NO_QUAD_EXT' - No quadrature extension
        - 'QUAD_EXT' - Quadrature extension used
- repeat - Decides whether combined fft operation is enabled or not;
    - Valid values are:
        - 'NO_COMBINED_FFT' - No combined fft operation
        - 'COMBINED_FFT8' - Combined FFT8, for ffts of size 8
        - 'COMBINED_FFT32' - Combined FFT32, for ffts of size 32
- shft_val - Pre-scaling left shift value;
    - Valid values are: 0, 1, 2, 3, 4, 8.

2.4.1 Mapping with SPT 2.0/2.5 RDX2 command parameters

You can find more information about RDX2 command in SPT Reference Manual.

SPT RDX2 command parameters	rdx2_mex
OPCODE	none
IN_DATTYP	none
FFT_RND	fft_rnd
REAL_FFT	real_fft
ADPTV	none
ADPTV_SHFT	none
QUAD_EXT	quad_ext
REPEAT	repeat
IMA	none
VEC_SZ	none
SRC_ADD	none
DEST_ADD	none
MULT_COEFF_ADD	none
SRC_ADD_INC	none
DEST_ADD_INC	none
ADPTV_OFFSET	none
ADPTV_WRNUM	none
SHFTVAL_SRC	none
SHFT_VAL	shft_val
TW_OVS	none

2.4.2 Example

```
fft_rnd = 1;
real_fft = 'NO_OPERANDS_SPLIT';
quad_ext = 'NO_QUAD_EXT';
repeat = 'COMBINED_FFT8';
shft_val = 0;
M_Tw = [32767 23170-23170i -32767i -23170-23170i -32767 -23170
32767i 23170+23170i];
M_Op = [248975-317264i,-519498+52108i,86146+441867i,-375405-
367350i, ...
```

-16388-192669i,166735+197977i,-126664-330079i,49351-128592i, ...
-39508-282013i,-195781+31537i,-218248-522430i,-266389+82837i, ...
-436294-91245i,330689+8615i,-5155-159261i,416425-51192i, ...
454932+278886i,242064+35474i,43172+250573i,-204936-376372i, ...
236822-124257i,123574+478023i,-320548-407248i,286676+88245i, ...
-183088-409663i,318195-489399i,265329-187782i,107026+481362i, ...
-137811+370408i,200135-259592i,515868-328310i,84997+201149i];

M_Out = **rdx2_mex**(M_Tw, M_Op, fft_rnd, real_fft, quad_ext, repeat, shft_val);

2.5 RDX4

M_Out = **rdx4_mex**(M_Tw or M_Wc, M_Op, win_type, win_coeff_type, fft_rnd, quad_ext, repeat, shft_val);

- M_Out - The output is a complex vector that has the same size as the operand vector
- M_Tw - Twiddle values for rounds 1, 2, 3, 4, or 5, or Window coefficients values for round 0; Twiddle values and Window coefficients values must be complex vectors
  - **NOTE**: *Maximum size of twiddles data is 4096 elements. (Size of SPT TWRAM is 0x1000)*
  - **NOTE**: *Twiddles values are expected in regular format. SPT Toolbox will take care to properly load them into SPT simulator memory (repeat each value 7 times, resulting in 8 similar values for each twiddle).*

- M_Wc - Window coefficients values must be a complex vector
  - **NOTE**: *Maximum size of win coeff input data is 4096 elements. (Size of SPT TWRAM is 0x1000)*
  - **NOTE**: *Window coefficients are expected in natural order. SPT Toolbox will take care to properly load them into SPT simulator memory.*

- M_Op - Operands values must be a complex vector
  - **NOTE**: *Maximum size of input data is 16384 elements. (Size of SPT OPRAM is 0x4000)*

- win_type - Windowing operation - Indicates the type of window coefficients;
  - Valid values are:
    - 'WIN_DISABLED' - Windowing is not enabled
    - 'WIN_ENABLED' - Windowing is enabled

- win_coeff_type - Win coefficient type - Indicates the type of window coefficients; Valid only when win_type is 'WIN_ENABLED';
    - Valid values are:
        - 'SINGLE_COEFF' - Single complex value window
        - 'MULTIPLE_COEFF' - Multiple complex values window
- fft_rnd - FFT Round;
    - Valid values are: 0, 1, 2, 3, 4, 5.
- quad_ext - Quadrature Extension;
    - Valid values are:
        - 'NO_QUAD_EXT' - No quadrature extension
        - 'QUAD_EXT' - Quadrature extension used
- repeat - Decides whether combined fft operation is enabled or not;
    - Valid values are:
        - 'NO_COMBINED_FFT' - No combined fft operation
        - 'COMBINED_FFT8' - Combined FFT8, for ffts of size 8
        - 'COMBINED_FFT16' - Combined FFT16, for ffts of size 16
        - 'COMBINED_FFT32' - Combined FFT32, for ffts of size 32
- shft_val - Pre-scaling left shift value;
    - Valid values are: 0, 1, 2, 3, 4, 8.

2.5.1 Mapping with SPT 2.0/2.5 RDX4 command parameters

You can find more information about RDX4 command in SPT Reference Manual.

SPT RDX4 command parameters	rdx4_mex
OPCODE	none
IN_DATTYP	none
WIN_TYPE	win_type & win_coeff_type
FFT_RND	fft_rnd
ADPTV	none
ADPTV_SHFT	none
QUAD_EXT	quad_ext
REPEAT	repeat
IMA	none
VEC_SZ	none
SRC_ADD	none
DEST_ADD	none
MULT_COEFF_ADD	none

SRC_ADD_INC	none
DEST_ADD_INC	none
MCA_MOD	none
ADPTV_OFFSET	none
MCA_INC	none
ADPTV_WRNUM	none
MULT_MOD	none
SHFTVAL_SRC	none
SHFT_VAL	shft_val
TW_OVS	none

2.5.2 Example

```
win_type = 'WIN_ENABLED';
win_coeff_type = 'SINGLE_COEFF';
fft_rnd = 0;
quad_ext = 'NO_QUAD_EXT';
repeat = 'COMBINED_FFT8';
shft_val = 0;
M_Wc = 23+2i;
M_Tw = [32767 23170-23170i -32767i -23170-23170i -32767 -
23170+23170i 32767i 23170+23170i];
M_Op = [248975-317264i,-519498+52108i,86146+441867i,-375405-
367350i, ...
   -16388-192669i,166735+197977i,-126664-330079i,49351-128592i, ...
   -39508-282013i,-195781+31537i,-218248-522430i,-266389+82837i, ...
   -436294-91245i,330689+8615i,-5155-159261i,416425-51192i, ...
   454932+278886i,242064+35474i,43172+250573i,-204936-376372i, ...
   236822-124257i,123574+478023i,-320548-407248i,286676+88245i, ...
   -183088-409663i,318195-489399i,265329-187782i,107026+481362i, ...
   -137811+370408i,200135-259592i,515868-328310i,84997+201149i];
M_Out = rdx4_mex(complex(M_Wc), M_Op, win_type, win_coeff_type, fft_rnd,
quad_ext, repeat, shft_val);
```

2.6 IRDX2

M_Out = **irdx2_mex**(M_Tw, M_Op, fft_rnd, quad_ext, repeat, shft_val);

- M_Out - The output is a complex vector that has the same size as the operand vector
- M_Tw - Twiddle values must be a complex vector
    - **NOTE**: *Maximum size of twiddles data is 4096 elements. (Size of SPT TWRAM is 0x1000)*
    - **NOTE**: *Twiddles values are expected in regular format. SPT Toolbox will take care to properly load them into SPT simulator memory (repeat each value 7 times, resulting in 8 similar values for each twiddle)*

- M_Op - Operands values must be a complex vector
    - **NOTE**: *Maximum size of input data is 16384 elements. (Size of SPT OPRAM is 0x4000)*

- fft_rnd - FFT Round; Valid values are: 1, 2, 3, 4, 5
- quad_ext - Quadrature Extension;
    - Valid values are:
        - 'NO_QUAD_EXT' - No quadrature extension
        - 'QUAD_EXT' - Quadrature extension used
- repeat - Decides whether combined fft operation is enabled or not;
    - Valid values are:
        - 'NO_COMBINED_FFT' - No combined fft operation
        - 'COMBINED_FFT8' - Combined FFT8, for ffts of size 8
        - 'COMBINED_FFT32' - Combined FFT32, for ffts of size 32
- shft_val - Pre-scaling left shift value;
    - Valid values are: 0, 1, 2, 3, 4, 8.

2.6.1 Mapping with SPT 2.0/2.5 IRDX2 command parameters

You can find more information about IRDX2 command in SPT Reference Manual.

SPT IRDX2 command parameters	irdx2_mex
OPCODE	none
IN_DATTYP	none
FFT_RND	fft_rnd
ADPTV	none
ADPTV_SHFT	none
QUAD_EXT	quad_ext
REPEAT	repeat
IMA	none

VEC_SZ	none
SRC_ADD	none
DEST_ADD	none
MULT_COEFF_ADD	none
SRC_ADD_INC	none
DEST_ADD_INC	none
ADPTV_OFFSET	none
ADPTV_WRNUM	none
SHFTVAL_SRC	none
SHFT_VAL	shft_val
TW_OVS	none

2.6.2 Example

```
fft_rnd = 1;
quad_ext = 'NO_QUAD_EXT';
repeat = 'COMBINED_FFT8';
shft_val = 0;
M_Tw = [32767 23170-23170i -32767i -23170-23170i -32767 -
23170+23170i 32767i 23170+23170i];
M_Op = [29070-63736i,-44092+31257i,-5065+13972i,-40753-61987i, ...
33167-15573i,-73543+47758i,96484-26599i,16438+58642i, ...
-59470-46653i,16862+5018i,-27923-85204i,18752+3955i, ...
49593-23844i,-44505+48555i,-45392+26634i,72193+48502i, ...
86461+19326i,45700+64181i,-34669-19583i,10216-36013i, ...
27261+50388i,-28641-49586i,82220-45461i,2385+84512i, ...
-40109-4907i,64785-93616i,97640-64506i,24000+85306i, ...
-5660-97501i,-9877-30745i,17564+31314i,22818-26713i];
M_Out = irdx2_mex(M_Tw, M_Op, fft_rnd, quad_ext, repeat, shft_val);
```

2.7 IRDX4

M_Out = **irdx4_mex**(M_Tw or M_Wc, M_Op, win_type, win_coeff_type, fft_rnd, quad_ext, repeat, shft_val);

- M_Out - The output is a complex vector that has the same size as the operand vector
- M_Wc - Window coefficients values must be a complex vector

- **NOTE**: *Maximum size of win coeff input data is 4096 elements. (Size of SPT TWRAM is 0x1000)*
- **NOTE**: *Window coefficients are expected in natural order. SPT Toolbox will take care to properly load them into SPT simulator memory.*

- M_Tw - Twiddle values for rounds 1, 2, 3, 4, or 5, or Window coefficients values for round 0; Twiddle values and Window coefficients values must be complex vectors
  - **NOTE**: *Maximum size of twiddles data is 4096 elements. (Size of SPT TWRAM is 0x1000)*
  - **NOTE**: *Twiddles values are expected in regular format. SPT Toolbox will take care to properly load them into SPT simulator memory (repeat each value 7 times, resulting in 8 similar values for each twiddle)*

- M_Op - Operands values must be a complex vector
  - **NOTE**: *Maximum size of input data is 16384 elements. (Size of SPT OPRAM is 0x4000)*

- win_type - Windowing operation - Indicates the type of window coefficients;
  - Valid values are:
    - 'WIN_DISABLED' - Windowing is not enabled
    - 'WIN_ENABLED' - Windowing is enabled
- win_coeff_type - Win coefficient type - Indicates the type of window coefficients;
  - Valid values are:
    - 'SINGLE_COEFF' - Single complex value window
    - 'MULTIPLE_COEFF' - Multiple complex values window
- fft_rnd - FFT Round;
  - Valid values are: 0, 1, 2, 3, 4, 5.
- quad_ext - Quadrature Extension;
  - Valid values are:
    - 'NO_QUAD_EXT' - No quadrature extension
    - 'QUAD_EXT' - Quadrature extension used
- repeat - Decides whether combined fft operation is enabled or not;
  - Valid values are:
    - 'NO_COMBINED_FFT' - No combined fft operation
    - 'COMBINED_FFT8' - Combined FFT8, for ffts of size 8
    - 'COMBINED_FFT16' - Combined FFT16, for ffts of size 16
    - 'COMBINED_FFT32' - Combined FFT32, for ffts of size 32

- shft_val - Pre-scaling left shift value;
    - Valid values are: 0, 1, 2, 3, 4, 8.

2.7.1 Mapping with SPT 2.0/2.5 IRDX4 command parameters

You can find more information about IRDX4 command in SPT Reference Manual.

SPT IRDX4 command parameters	irdx4_mex
OPCODE	none
IN_DATTYP	none
WIN_TYPE	win_type & win_coeff_type
FFT_RND	fft_rnd
ADPTV	none
ADPTV_SHFT	none
QUAD_EXT	quad_ext
REPEAT	repeat
IMA	none
VEC_SZ	none
SRC_ADD	none
DEST_ADD	none
MULT_COEFF_ADD	none
SRC_ADD_INC	none
DEST_ADD_INC	none
MCA_MOD	none
ADPTV_OFFSET	none
MCA_INC	none
ADPTV_WRNUM	none
MULT_MOD	none
SHFTVAL_SRC	none
SHFT_VAL	shft_val
TW_OVS	none

2.7.2 Example

```
win_type = 'WIN_ENABLED';
win_coeff_type = 'SINGLE_COEFF';
```

```
fft_rnd = 0;
quad_ext = 'NO_QUAD_EXT';
repeat = 'COMBINED_FFT8';
shft_val = 0;
M_Wc = 23+2i;
M_Tw = [32767 23170-23170i -32767i -23170-23170i -32767 -
23170+23170i 32767i 23170+23170i];
M_Op = [-60840-80495i,72550+64230i,127383-74372i,-74204+101009i, ...
   108854-19035i,186760-148577i,-59112-81049i,-52435-78958i, ...
   -51780-122887i,31892+99851i,-30485+40442i,95042+66221i, ...
   -123011-140835i,-23489-94271i,-32612+36660i,94935-167181i, ...
   107713+27914i,83229+39855i,221330+3425i,-189063+126879i, ...
   -4125-28426i,135740-30001i,20936+74395i,79141+64824i, ...
   146322-77724i,24848-123647i,-316681+18815i,-27257-96123i, ...
   -31255-61105i,-1038-8729i,41173+100387i,-19192-161515i];
M_Out = irdx4_mex(complex(M_Wc), M_Op, win_type, win_coeff_type,
fft_rnd, quad_ext, repeat, shft_val);
```

2.8 WIN

M_Out = **win_mex**(M_Wc, M_Op, win_coeff_type, shft_val);

- M_Out - The output is a complex vector that has the same size as the operand vector
- M_Wc - Window coefficients values must be a complex vector
    - **NOTE**: *Maximum size of win coeff input data is 4096 elements. (Size of SPT TWRAM is 0x1000)*
    - **NOTE**: *Window coefficients are expected in natural order. SPT Toolbox will take care to properly load them into SPT simulator memory.*

- M_Op - Operands values must be a complex vector
    - **NOTE**: *Maximum size of input data is 16384 elements. (Size of SPT OPRAM is 0x4000)*

- win_coeff_type - Win coefficient type - Indicates the type of window coefficients;
    - Valid values are:
        - 'SINGLE_COEFF' - Single complex value
        - 'MULTIPLE_COEFF' - Multiple complex values

- shft_val - Pre-scaling left shift value;
  - Valid values are: 0, 1, 2, 3, 4, 8.

2.8.1 Mapping with SPT 2.0/2.5 WIN command parameters

You can find more information about WIN command in SPT Reference Manual.

SPT WIN command parameters	win_mex
OPCODE	none
IN_DATTYP	none
WIN_TYPE	win_coeff_type
IMA	none
VEC_SZ	none
SRC_ADD	none
DEST_ADD	none
MULT_COEFF_ADD	none
SRC_ADD_INC	none
DEST_ADD_INC	none
MCA_MOD	none
SHFTVAL_OFST	none
MCA_INC	none
SHFTVAL_WR_NO	none
MULT_MOD	none
SHFTVAL_SRC	none
SHFT_VAL	shft_val

2.8.2 Example

```
win_coeff_type = 'MULTIPLE_COEFF';
shft_val = 0;
M_Wc = [0 38 156 374 715 1203 1856 2680 ...
        3657 4754 5912 7060 8115 8995 9627 9958 ...
        9958 9627 8995 8115 7060 5912 4754 3657 ...
        2680 1856 1203 715 374 156 38 0];
M_Op = [248975-317264i,-519498+52108i,86146+441867i,-375405-
367350i, ...
  -16388-192669i,166735+197977i,-126664-330079i,49351-128592i, ...
```

-39508-282013i,-195781+31537i,-218248-522430i,-266389+82837i, ...
-436294-91245i,330689+8615i,-5155-159261i,416425-51192i, ...
454932+278886i,242064+35474i,43172+250573i,-204936-376372i, ...
236822-124257i,123574+478023i,-320548-407248i,286676+88245i, ...
-183088-409663i,318195-489399i,265329-187782i,107026+481362i, ...
-137811+370408i,200135-259592i,515868-328310i,84997+201149i];

M_Out = **win_mex**(complex(M_Wc), M_Op, win_coeff_type, shft_val);

2.9 FIR

M_Out = **fir_mex**(M_Coeff, M_Op, init, no_of_taps, shft_val);

- M_Out - The output is a complex vector that has the same size as the operand vector
- M_Coeff - Tap coefficients values must be a complex vector
  - **NOTE**: *Maximum size of coeff input data is 4096 elements. (Size of SPT TWRAM is 0x1000)*

- M_Op - Operands values must be a complex vector
  - **NOTE**: *Maximum size of input data is 16384 elements. (Size of SPT OPRAM is 0x4000)*

- init - Decides which type of initialization process is to be performed during FIR operation;
  - Valid values are:
    - 'INIT_ZERO_RUN_IN' - Zero run-in initialization performed
    - 'INIT_CONST_RUN_IN' - Constant run-in initialization performed
- no_of_taps - Gives the number of coefficients in the Scalar product;
  - Valid values are: 2, 3, 4, 5, 6, 7, 8.
- shft_val - Pre-scaling left shift value;
  - Valid values are: 0, 1, 2, 3, 4, 8.

2.9.1 Mapping with SPT 2.0/2.5 FIR command parameters

You can find more information about FIR command in SPT Reference Manual.

SPT FIR command parameters	fir_mex
OPCODE	none
IN_DAT_TYP	none
WIN_TYPE	none

INIT	init
NO_OF_TAPS	no_of_taps
IMA	none
VEC_SZ	none
SRC_ADD	none
DEST_ADD	none
TAP_COEFF_ADD	none
SRC_ADD_INC	none
DEST_ADD_INC	none
SHFTVAL_OFST	none
SHFTVAL_WR_NO	none
SHFTVAL_SRC	none
SHFT_VAL	shft_val

2.9.2 Example

```
init = 'INIT_ZERO_RUN_IN';
no_of_taps = 8;
shft_val = 0;
M_Coeff = [10+2i -3+4i 5 7i 8-4i 76 0 -3];
M_Op = [186967i,8129+178838i,16258+170709i,24387+162580i,...
  32516+154451i,40645+146322i,48774+138193i,56903+130064i,...
  65032+121935i,73161+113806i,81290+105677i,89419+97548i,...
  97548+89419i,105677+81290i,113806+73161i,121935+65032i,...
  130064+56903i,138193+48774i,146322+40645i,154451+32516i,...
  162580+24387i,170709+16258i,178838+8129i,186967];
M_Out = fir_mex(M_Coeff, M_Op, init, no_of_taps, shft_val);
```

2.10 SCP

M_Out = **scp_mex**(M_Coeff, M_Op, no_of_taps, shft_val);

- M_Out - The output is a complex vector
- M_Coeff - Coefficients values must be a complex vector
    - **NOTE**: *Maximum size of coeff input data is 4096 elements. (Size of SPT TWRAM is 0x1000)*

- M_Op - Operands values must be a complex vector
  - **NOTE**: *Maximum size of input data is 16384 elements. (Size of SPT OPRAM is 0x4000)*

- no_of_taps - Gives the number of coefficients in the Scalar product.
  - Valid values are: 2, 4, 8.
- shft_val - Pre-scaling left shift value.
  - Valid values are: 0, 1, 2, 3, 4, 8.

2.10.1 Mapping with SPT 2.0/2.5 SCP command parameters

You can find more information about SCP command in SPT Reference Manual.

SPT SCP command parameters	scp_mex
OPCODE	none
IN_DAT_TYP	none
RE_IM_COEFF	none
NO_OF_TAPS	no_of_taps
IMA	none
VEC_SZ	none
SRC_ADD	none
DEST_ADD	none
COEFF_ADD	none
SRC_ADD_INC	none
DEST_ADD_INC	none
SHFTVAL_OFST	none
SHFTVAL_WR_NO	none
SHFTVAL_SRC	none
SHFT_VAL	shft_val

2.10.2 Example

```
no_of_taps = 8;
shft_val = 0;
M_Coeff = [10+2i 1-1i 2+2i 3 5i 7+2i 8-9i 6+3i];
M_Op = [11685+0i,508+11177i,1016+10669i,103136+0i, ...
   2032+9653i,2540+9145i,3048+8637i,3556+8129i, ...
   4064+7620i,11685+0i,5080+6604i,5588+6096i, ...
```

6096+5588i,6604+5080i,7112+4572i,7620+4064i, ...
8129+3556i,8637+3048i,9145+2540i,9653+2032i, ...
10161+1524i,10669+1016i,11177+508i,11685+0i];
M_Out = **scp_mex**(M_Coeff, M_Op, no_of_taps, shft_val);

2.11 VMT

M_Out = **vmt_mag2_log2_mex**(M_Op1, M_Op2, op_sq2_st1, op_sq2_st2, op_sq2_st3, shft_val, offset_val, s_val, n_val);

- M_Out - The output is a complex vector
- M_Op1 - Operands 1 values must be a complex vector
  - **NOTE**: *Maximum size of Op1 input data is 16384 elements. (Size of SPT OPRAM is 0x4000)*

- M_Op2 - Operands 2 values must be a complex vector
  - **NOTE**: *Maximum size of Op2 input data is 4096 elements. (Size of SPT TWRAM is 0x1000)*

- op_sq2_st1 - Operation Sequence 2 - Stage1.
  - Valid values are:
    - 'BYPASS' - No Preproccessing
    - 'ABS' - ABS (for real and complex numbers)
    - 'MAG' - Magnitude for complex values
    - 'CONJ' - Conjugate
- op_sq2_st2 - Operation Sequence 2 - Stage3.
  - Valid values are:
    - BYPASS - No Operation
    - SUM_SCALE - Sum and Scale
- op_sq2_st3 - Operation Sequence 2 - Stage3.
  - Valid values are:
    - 'BYPASS' - No Operation
    - 'SUM_SCALE' - Sum and Scale
- shft_val - Shift value is signed 6-bit number.
- offset_val - Offset value must be a complex scalar.
- s_val - Multiplier value is a signed 16-bit number
- n_val - Exponent value is a 6-bit number

****
M_Out = **vmt_seq2_mex**(M_Op);

- M_Out - The output is a complex vector
- M_Op - Operand values must be a complex vector
    - **NOTE**: *Maximum size of input data is 16384 elements. (Size of SPT OPRAM is 0x4000)*

2.11.1 Mapping with SPT 2.0/2.5 VMT command parameters

You can find more information about VMT command in SPT Reference Manual.

SPT VMT command parameters	vmt_seq2_mex
OPCODE	none
RST_ACC	none
IN_DATTYP	none
OP_SQ1	none
OP_SQ2 – Stage1	op_sq2_st1
OP_SQ2 – Stage2	op_sq2_st2
OP_SQ2 – Stage3	op_sq2_st3
OP_PACK	none
IN_PACK	none
IMA	none
VEC_SZ	none
SRC_ADD	none
DEST_ADD	none
MULT_S_EXP_N_VAL_ADDR	s_val/n_val
SRC_ADD_INC	none
DEST_ADD_INC	none
IMDT_OFFSET_VAL_IM[31:16]	none
2ND_VEC_SRC_ADDR	none
SHIFT_VAL_ADDR(b)	shft_val
OFFSET_VAL_ADDR	offset_val
IMDT_OFFSET_VAL_RE[15:0]	none

SPT VMT command parameters	vmt_mag2_log2_mex
OPCODE	none
RST_ACC	none

IN_DATTYP	none
OP_SQ1	none
OP_SQ2 – Stage1	none
OP_SQ2 – Stage2	none
OP_SQ2 – Stage3	none
OP_PACK	none
IN_PACK	none
IMA	none
VEC_SZ	none
SRC_ADD	none
DEST_ADD	none
MULT_S_EXP_N_VAL_ADDR	none
SRC_ADD_INC	none
DEST_ADD_INC	none
IMDT_OFFSET_VAL_IM[31:16]	none
2ND_VEC_SRC_ADDR	none
SHIFT_VAL_ADDR(b)	none
OFFSET_VAL_ADDR	none
IMDT_OFFSET_VAL_RE[15:0]	none

2.11.2 Example

```
op_sq2_st1 = 'ABS';
op_sq2_st2 = 'BYPASS';
op_sq2_st3 = 'BYPASS';
shft_val = 0;
offset_val = 10+20i;
s_val = 0;
n_val = 0;
M_Op1 = [32767 101+23i 34+356i 10+10i];
M_Op2 = [230+944i 8186+814i 2291+673i 168+1325i];

M_Out = vmt_seq2_mex(M_Op1, M_Op2, op_sq2_st1, op_sq2_st2, op_sq2_st3,
shft_val, offset_val, s_val, n_val);
```

****
M_Op = [32767 101+23i 34+356i 10+10i];
M_Out = **vmt_mag2_log2_mex**(M_Op);

2.12 HIST

M_Out = **hist_mex**(M_Op, dat_typ, preproc, bin_sz, thr);

- M_Out - The output is vector that has 16, 32, or 64 values, depending on the number of bins and the magnitude of the operands
- M_Op - Operands values must be a complex vector
  - **NOTE**: *Maximum size of input data is 16384 elements. (Size of SPT OPRAM is 0x4000)*

- dat_typ - Input data type;
  - Valid values are:
    - 'REAL' - real - Packing not enabled for log2 or real operand and data comes from real component
    - 'COMPLEX' – complex
    - 'LOG2' - log2
- preproc – preprocessing
  - Valid values are:
    - 'NO_PROCESSING' - for real input data type it corresponds to 'no preprocessing', for complex input data type it corresponds to abs operation
    - 'ABS' - abs for real and complex input data type
    - 'MAG' - for real input data type it corresponds to abs operation, for complex input data type it corresponds to mag operation
- bin_sz - Bin size;
  - Valid values are:
    - '16BINS' - 16 bins
    - '32BINS' - 32 bins
    - '64BINS' - 64 bins
- thr - Minimum threshold bin

2.12.1 Mapping with SPT 2.0/2.5 HIST command parameters

You can find more information about HIST command in SPT Reference Manual.

SPT HIST command parameters	hist_mex

OPCODE	none
DAT_TYP	dat_typ
PREPROC	preproc
OP_MOD	none
BIN_SZ	bin_sz
PACK_EN	none
IMA	none
VEC_SZ	none
SRC_ADD	none
DEST_ADD	none
SRC_ADD_INC	none
DEST_ADD_INC	none
THR	thr
DAT_SZ	none

2.12.2 Example

```
dat_typ = 'REAL';
preproc = 'ABS';
bin_sz = '16BINS';
thr = 0;
M_Op = [10+2i -3+4i 5 7i];
M_Out = hist_mex(complex(M_Op), dat_typ, preproc, bin_sz, complex(thr));
```

2.13 MAXS

M_Out = **maxs_mex**(M_Thld, M_Op, in_dattyp, preproc, thld_cmp, in_tag, loc_n_abs, tag_n_bitfld, cyc_extn, maxsn_en, maxsn_sel);

- M_Out - The output is:
  - a complex scalar if loc_n_abs is 'GLOBAL_MAX'a complex vector with the same size as the operand vector if loc_n_abs is 'LOCAL_MAX' and tag_n_bitfld is 'TAGGED_VEC'
  - a real vector with the same size as the operand vector if loc_n_abs is 'LOCAL_MAX' and tag_n_bitfld is 'PACKED_BITFLD'
- M_Thld - Threshold values;
  - if MAXSN is disabled then it must be a complex scalar

    - if MAXSN is enabled, then it must be a complex vector with:
        - 4 elements for MAXS4
        - 8 elements for MAXS8
        - 16 elements for MAXS16
    - **NOTE**: *Maximum size of threshold data is 4096 elements. (Size of SPT TWRAM is 0x1000)*

- M_Op - Operands values it must be a complex vector
    - **NOTE**: *Maximum size of input data is 16384 elements. (Size of SPT OPRAM is 0x4000)*

- in_dattyp - Input Data Type;
    - Valid values are:
        - 'REAL' – Real
        - 'COMPLEX' – Complex
        - 'LOG2' - Log2
- preproc - Pre-processing;
    - Valid values are:
        - 'NO_PROCESSING' - No preprocessing (for log2 and others)
        - 'ABS' - ABS(real) + ABS(im) (for real and complex numbers. Imaginary part ignored in case of real input)
        - 'MAG' - Magnitude for complex values (calculate half of approximate square root of real^2 + im^2 - only valid for complex numbers)
- thld_cmp - Threshold Compare (valid only for local maxima);
    - Valid values are:
        - 'THLD_DISABLED' - Threshold compare disabled
        - 'THLD_ENABLED' - Threshold compare enabled
- in_tag - Input Tagged;
    - Valid values are:
        - 'NO_TAG' - Input is not tagged
        - 'TAGGED' - Input is tagged
- loc_n_abs - Local not Global maxima;
    - Valid values are:
        - 'GLOBAL_MAX' - Global maxima to be calculated
        - 'LOCAL_MAX' - Local maxima to be calculated
- tag_n_bitfld - Tag not bitfield (valid only for local maximum calculation);
    - Valid values are:
        - 'PACKED_BITFLD' - Output to be packed bitfields
        - 'TAGGED_VEC' - Output to be tagged vectors

- cyc_extn - Cyclic extension (valid only for local maximum calculation);
  - Valid values are:
    - 'NO_CYC_EXTN' - No cyclic extension
    - 'CYC_EXTN' - Cyclic extension
- maxsn_en - MAXSN enable;
  - Valid values are:
    - 'MAXSN_DISABLED' - MAXSN disabled
    - 'MAXSN_ENABLED' - MAXS to be calculated for multiple blocks of N operands at one time
- maxsn_sel - MAXSN operand Multiplicity select;
  - Valid values are:
    - 'MAXS16' - MAXS16
    - 'MAXS8' - MAXS8
    - 'MAXS4' - MAXS4

2.13.1 Mapping with SPT 2.0/2.5 MAXS command parameters

You can find more information about MAXS command in SPT Reference Manual.

SPT MAXS command parameters	maxs_mex
OPCODE	none
IN_DATTYP	in_dattyp
PREPROC	preproc
THLD_CMP	thld_cmp
IN_TAG	in_tag
LOC_N_ABS	loc_n_abs
TAG_N_BITFLD	tag_n_bitfld
CYC_EXTN	cyc_extn
MAXSN_EN	maxsn_en
IN_PACK	none
IMA	none
VEC_SZ	none
SRC_ADD	none
DEST_ADD	none
SRC_ADD_INC	none
DEST_ADD_INC	none
MAXSN_SEL	maxsn_sel

THLD_ADD	none

2.13.2 Example

```
in_dattyp = 'REAL';
preproc = 'NO_PROCESSING';
thld_cmp = 'THLD_DISABLED';
in_tag = 'NO_TAG';
loc_n_abs = 'GLOBAL_MAX';
tag_n_bitfld = 'PACKED_BITFLD';
cyc_extn = 'NO_CYC_EXTN';
maxsn_en = 'MAXSN_DISABLED';
maxsn_sel = 'MAXS16';
M_Thld = 1024;
M_Op = [-20 -19 -18 -17 -16 -15 -14 -13 -12 -11 -10 -9 -8 -7 -6 -5];
M_Out = maxs_mex(complex(M_Thld), complex(M_Op), in_dattyp, preproc,
thld_cmp, in_tag, loc_n_abs, tag_n_bitfld, cyc_extn, maxsn_en, maxsn_sel);
```

2.14 PDMA

NOTE: *Only Compression and Decompression functionality of PDMA command are supported in SPT Toolbox*

M_Out = **pdma_mex**(M_Op,data_packing,trans_type);

- M_Out - The output is a vector with complex or real values, depending on the type of operation: compression (real output – uint32) or decompression (complex output)
- M_Op - Operands values must be a complex vector or a real vector depending of type of operation: compression (complex input) or decompression (real input – uint32)
  NOTE: *Maximum size of input data for compression is 16384 elements. Maximum size of output data for decompression is 16384 elements. (Size of SPT OPRAM is 0x4000)*
- data_packing - defines data packing type
  - Valid values are:
    - 'CP4' - CP4 compression
    - 'CP6' – CP6 compression
    - 'CP8' – CP8 compression
    - 'CP16' – CP16 compression
    - 'CP4D' – CP4D compression

- 'CP4FMTA' – CP4FMTA compression
- 'CP4DFMTA' – CP4DFMTA compression
- 'CP4FMTB' – CP4FMTB compression
- 'CP4DFMTB' – CP4DFMTB compression
- 'CP8FMTB' – CP8FMTB compression
- 'CP16FMTB' – CP16FMTB compression

- trans_type - Direction of transfer from PDMA
  - Valid values are:
    - 'DECOMPRESSION' - System RAM to Operand/Twiddle RAM
    - 'COMPRESSION' - Operand/Twiddle RAM to System RAM

2.14.1 Mapping with SPT 2.0/2.5 PDMA command parameters

You can find more information about PDMA command in SPT Reference Manual.

SPT PDMA command parameters	pdma_mex
OPCODE	none
TAG_N_BITFLD	none
SE/COMP_RND	none
IMA	none
DATA_PACKING	data_packing
TRANS_TYPE	trans_type
SYNC	none
VECTOR_LEN	none
SYSRAM_MEM_START_ADDR[31:16]	none
SYSRAM_MEM_START_ADDR[15:1]	none
OPRAM_MEM_START_ADDR[15:0]	none
OPRAM_SKIP_ADDR[11:0]	none
OPRAM_CONTINUOUS_ADDR[11:8]	none
OPRAM_CONTINUOUS_ADDR[7:0]	none
SYSRAM_SKIP_ADDR[11:4]	none
SYSRAM_SKIP_ADDR[3:0]	none
SYSRAM_CONTINUOUS_ADDR[11:0]	none

2.14.2 Example

```
data_packing = 'CP4';
```

```
trans_type = 'COMPRESSION';
M_Op = [4-21i,232+165i,253-37i,224-142i, ...
   14-167i,152-136i,-201-202i,1-120i];
M_Out = pdma_mex(M_Op,comp_rnd,data_packing,trans_type);
```