AN12144

QN908x FFT Application

Rev. 1.0 — 1st Feb 2018 Application note

Document information

Info	Content
Keywords	QN908x, FSP, FFT, Size extension
Abstract	This Application note describes in brief how to use FSP and extend the size of FFT base on FSP module

Revision history

Rev	Date	Description
1.0	2018/2/1	256, 512 and 1024 floating-point real FFT

Contact information

For more information, please visit: http://www.nxp.com

1. Introduction

Fusion Signal Processor (FSP) is a hardware engine for the fast computation of various math operations widely used in signal processing applications. Acceleration of data processing using FSP will lead to shorter processing time, and the system power consumption can be reduced accordingly.

The transform module of FSP can support 64, 128, 256 points FFT, DCT, real or complex data input and output. This document describes how to use FSP FFT API for frequency spectrum analysis, and how to extend the FFT size from 256 points to 512, 1024 and 2048 points.

2. FFT Application Example

2.1 Prepare sample data for FFT API

Assume a sinusoidal input signal:

S = 2 + 3*cos(2*pi*50*t - pi*30/180) + 1.5*cos(2*pi*75*t + pi*90/180)

t = [0 : 1/Fs : N/Fs]

t : sample time

Fs : Sample rate, 256 Hz

N : Number of samples, 256 samples.

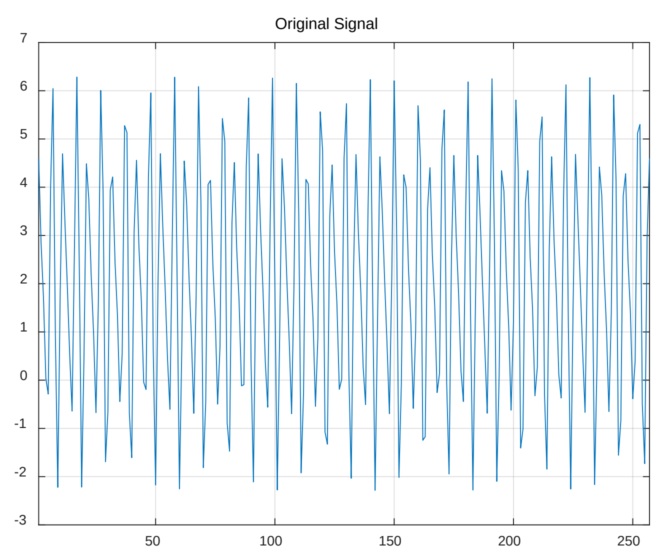

Figure 1: Sample Data

The 256-point sample vector

fft_fsp_input = {4.59807621135332, 2.84191751302600, 1.71440397252613, 0.0347124033769155, -0.287200292432018, 4.02120838745065, 6.04579972888750, 0.896142732324765, -2.22153900257525, 1.76510164286847, 4.69210954621404, 3.27456702575906, 1.95081632280391, 0.514803805203220, -0.640982047342048, 2.78445305176575, 6.28359677763413, 2.42877923519982, -2.21635623971070, 0.521534065978648, 4.48638175070111, 3.75396611733096, 2.17589400901478, 0.974590481734719, -0.672695503897899, 1.57787260849639, 6.00573219469963, 3.90504423961476, -1.69092992603525, -0.659082867664653, 3.94373329997557, 4.21219072943656, 2.43933982822018, 1.36426202905850, -0.443247510417074, 0.549536966835156, 5.27621740436700, 5.13226206798481, -0.695630239983522, -1.60680291931263, 3.07959934394469, 4.55820138086609, 2.77727612031898, 1.67046686354573, -0.0419868434703232, -0.195080488110095, 4.21844434209178, 5.95552395886739, 0.649517716144835, -2.17248535936746, 1.96669020780777, 4.69558740814159, 3.19784113458977, 1.91358193823990, 0.434353001853173, -0.605434006154910, 2.99292893749675, 6.28048030403132, 2.17175875995649, -2.25360486610982, 0.729057659938750, 4.54366080557622, 3.67333708125424, 2.13698144761755, 0.901923788646688, -0.687650747931490, 1.77015722876329, 6.08596903815587, 3.67141350114848, -1.81341344274375, -0.474005033088627, 4.05762692796643, 4.14097923459202, 2.39095236246059, 1.30520703361656, -0.496040656202027, 0.703432297682121, 5.42452794785135, 4.95224438782438, -0.889820079076584, -1.47180262741486, 3.24355607949707, 4.51249576308053, 2.71501327147352, 1.62472385558848, -0.116631049931536, -0.0935448868776603, 4.41099835788379, 5.85123794147857, 0.407873945228464, -2.10982045046155, 2.16473253809045, 4.69094643831866, 3.12272279683597, 1.87578384967112, 0.353592468881851, -0.560660171779830, 3.20152116925987, 6.26276466006379, 1.91410296233621, -2.27620089208363, 0.937514709720369, 4.59154198422165, 3.59254947303745, 2.09894309363596, 0.827387458752468, -0.694926998181787, 1.96686820385366, 6.15309244975989, 3.43162782556229, -1.92230481872205, -0.283188644170526, 4.16227643407457, 4.06711833205661, 2.34457213267443, 1.24374434491298, -0.543592514103711, 0.864944431317162, 5.56296357797740, 4.76148801148395, -1.07348870547996, -1.32671064374013, 3.40003593584245, 4.46139981081676, 2.65515554877764, 1.57702063595650, -0.188775879964790, 0.0172853174770278, 4.59807621135331, 5.73324571041233, 0.172095739946463, -2.03390923083431, 2.35856350061309, 4.67851210792131, 3.04941336027195, 1.83718795352141, 0.272869834332382, -0.506524896650943, 3.40944426592322, 6.23039995292588, 1.65676490181519, -2.28417459130101, 1.14611586613523, 4.63014782350767, 3.51195818010027, 2.06154720960207, 0.751173290183646, -0.694189876036248, 2.16735039061290, 6.20672055678570, 3.18656356919142, -2.01728088830138, -0.0874245378495449, 4.25754551208303, 3.99105533015859, 2.30005137995345, 1.17989108116135, -0.585459182095834, 1.03363954039199, 5.69088530612587, 4.56066017177979, -1.24599894426765, -1.17218805012686, 3.54863342292379, 4.40536337260371, 2.59770137223562, 1.52722313608124, -0.257968930348775, 0.137243502735002, 4.77889507166509, 5.60190831586806, -0.0569577107078028, -1.94516669596131, 2.54754877900054, 4.65863576545792, 2.97808535507122, 1.79756801323998, 0.192553923512402, -0.442932386634091, 3.61590229853663, 6.18339531460634, 1.40069627195902, -2.27761487751718, 1.35408452270636, 4.65963963655566, 3.43189576225223, 2.02455573697411, 0.673501524005753, -0.685132550539149, 2.37091955672179, 6.24652291125500, 2.93711972004226, -2.09807621135331, 0.112487524493188, 4.34334305876412, 3.91322778930148, 2.25722329067042, 1.11369294737180, -0.621208056070813, 1.20904238618739, 5.80768993365082, 4.35047089132187, -1.40676084575377, -

1.00892632248296, 3.68898647769877, 4.34484283824644, 2.54262179338239, 1.47521920380321, -0.323752330319560, 0.266117475700949, 4.95268718644647, 5.45764253858414, -0.278455996902429, -1.84405562418512, 2.73108730867131, 4.63169204868291, 2.90888210026885, 1.75670793419225, 0.113032925603368, -0.369828157658778, 3.82009240655523, 6.12181925292460, 1.14684330996148, -2.25666850444431, 1.56066017177981, 4.68021574594926, 3.35267090048019, 1.98772664790501, 0.594620115113008, -0.667478149940833, 2.57686511968050, 6.27222237662399, 2.68421405968449, -2.16448391128361, 0.315742561994724, 4.41962264330838, 3.83406108967175, 2.21590388354726, 1.04522471117233, -0.650420669768170, 1.39063783654050, 5.91281310379843, 4.13167004615189, -1.55523405159115, -0.837643935092477, 3.82077735848383, 4.28029829768659, 2.48986147196450, 1.42092013142768, -0.385665422543419, 0.403649567226934, 5.11870353128731, 5.30091934182279, -0.491600998254473, -1.73108411254453, 2.90861351486324, 4.59807621135334};

2.2 FFT API

2.2.1 FSP Module Initialization

Enable FSP’s power and initialize it.

```
POWER_DisablePD(kPDRUNCFG_PD_FSP); /* Power enable */
FSP_Init(FSP);                     /* FSP initialization */
```

2.2.2 FFT Input Parameter Set

As the number of input data is 256, *te_point* shall be set to *kFSP_TePts256Points*.

To ensure the range of FFT input is [-1, 1], one needs to find the maximum absolute value of the input data to obtain *te_scale*.

```
FSP_MaxMinF32(DEMO_FSP_BASE, fft_fsp_input, npts, &max_value, &min_value);
if (-min_value > max_value)
    max_value = -min_value;
tmp_max = *((uint32_t *)(&max_value));
tmp_exp = ((tmp_max >> 23) & (0x000000FF));
tmp_exp = tmp_exp - 127;
te_scale = -(tmp_exp + 1);
```

2.2.3 FFT Computation and result analysis

Run FSP_RFFTf32() to get the FFT result:

```
FSP_RfftF32(FSP, &fsp_rfft_cfg, (float32_t *)fft_fsp_input, fft_fsp_output);
```

Then the result from *FSP_RfftF32()* shall be scaled by *FSP_TePostProcess()*:

```
FSP_TePostProcess(FSP, &fsp_rfft_cfg, kFSP_TeIoModeRealInputComplexOutput,
kFSP_TeModeFft, fft_fsp_output);
```

All 256 complex results will be written to *fft_fsp_output*, and its magnitude value can be presented as below:

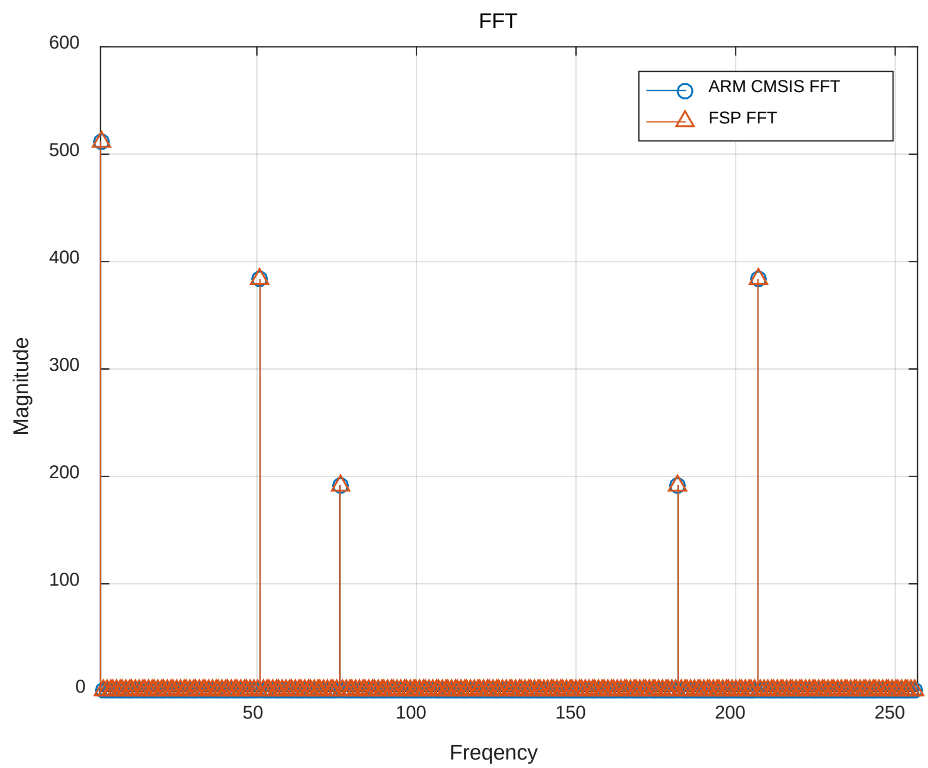

Figure 2: FFT Computation and result analysis

The symmetry of the FFT of real data can be observed. And from the diagram above (start from 1Hz for frequency axis). Peak index pairs [0,1], [100,101] and [150,151] denote to frequency components of the original signal at 0Hz, 50Hz and 75Hz respectively.

DC at 0Hz: *fft_fsp_output[0]*/N= 512/256 = 2

Phase of 50Hz signal: *atan2(fft_fsp_output[101]*, *fft_fsp_output[100]*)*180/pi = -30

Phase of 75Hz signal: *atan2(fft_fsp_output[151]*, *fft_fsp_output[150]*)*180/pi = 90

3. RFFT Size Extension

The transform engine in QN908x can only support 64, 128 and 256 points FFT. If one wants to process larger data in the application, the result cannot be obtained by using only hardware FSP engine. In this section, we discuss how to extend FFT size to 512, 1024 and 2048 points by leveraging the hybrid FFT FSP engine and auxiliary software processing.

3.1 512 and 1024 Point Real FFT Extension Principle

For a real sample sequence, the following algorithm allows one to compute a real FFT of size N by computing a complex FFT of size N/2.

$$X(r) = \sum_{l=0}^{N-1} x(l)W_N^{rl} \qquad r = 0, 1, \ldots, N-1,$$

$$= \sum_{l=0}^{N/2-1} x(2l)W_N^{r(2l)} + \sum_{l=0}^{N/2-1} x(2l+1)W_N^{r(2l+1)}$$

$$= \sum_{l=0}^{N/2-1} x(2l)W_N^{r(2l)} + W_N^r \sum_{l=0}^{N/2-1} x(2l+1)W_N^{r(2l)}$$

$$= \sum_{l=0}^{N/2-1} x(2l)W_{N/2}^{rl} + W_N^r \sum_{l=0}^{N/2-1} x(2l+1)W_{N/2}^{rl}$$

Let $f(l) = x(2l)$, $g(l) = x(2l+1)$ for $0 \le l \le N/2$. The DFTs of the two real series and the N/2 points complex sequence $y(l) = f(l) + jg(l)$ can represented as

$$F(r) = \sum_{l=0}^{N/2-1} f(l)W_{N/2}^{rl}, \quad G(r) = \sum_{l=0}^{N/2-1} g(l)W_{N/2}^{rl}$$

and

$$Y(r) = \sum_{l=0}^{N/2-1} f(l)W_{N/2}^{rl} = \sum_{l=0}^{N/2-1}(f(l) + jg(l))W_{N/2}^{rl} = \sum_{l=0}^{N/2-1} f(l)W_{N/2}^{rl} + j\sum_{l=0}^{N/2-1} g(l)W_{N/2}^{rl}$$

$$= F(r) + jG(r).$$

From the result of the complex FFT Y(r), F(r) and G(r) (for $0 \le r \le \frac{N}{2} - 1$) can be obtained.

$$F(r) = (Y(r) + \bar{Y}(\tfrac{N}{2} - r))/2$$

$$G(r) = j*(\bar{Y}\left(\tfrac{N}{2} - r\right) - Y(r))/2$$

Now the final FFT reslut can be obtained using F(r) and G(r) as below

$$X(r) = \sum_{l=0}^{N/2-1} x(2l)W_{N/2}^{rl} + W_N^r \sum_{l=0}^{N/2-1} x(2l+1)W_{N/2}^{rl}$$

$$= \sum_{l=0}^{N/2-1} f(l)W_{N/2}^{rl} + \sum_{l=0}^{N/2-1} g(l)W_{N/2}^{rl}$$

$$= F(r) + W_N^r G(r), \quad r = 0, 1, 2, \ldots, N/2-1$$

Since the x(l)'s is real, the X(r) 's has the symmetry property. Thus

$$X(N-r) = \bar{X}(r), \quad r = 0, 1, 2, \ldots, N/2-1$$

$$X(N/2) = F(0) - G(0).$$

3.2 2048 Point Real FFT Extension Principle

For 2048 real FFT algorithm, a divide-and-conquer approach is used. This approach is based on the decomposition of an N-point DFT into smaller DFTs.

For N-point DFT, assume N can be factored as a product of two integers.

$$N = LM$$

Step 1. Compute the M-point DFTs

$$F(l, q) = \sum_{m=0}^{M-1} x(l, m)W_M^{mq}, \text{ where } q = 0, 1, \ldots, M-1, \text{ and } l = 0, 1, \ldots, L-1.$$

Step 2. Multiply twiddle factor W_M^{lq} to F(l, q)

$$G(l, q) = W_M^{lq} F(l, q).$$

Step 3. Compute the L-point DFTs

$X(p, q) = \sum_{l=0}^{L-1} G(l,q) W_L^{lp}$, where p = 0, 1, …, L-1.

4. Performance Comparison

Clock cycle comparison between FSP API and CMSIS API is as below:

	256 RFFT	512 RFFT	1024 RFFT
FSP	2625	13133	43880
CMSIS	17560	38195	77363

Table 1: Clock cycle comparison

5. Code list

5.1 512 Floating-point Real FFT

```
/**
* @brief Real FFT for 512 float samples data.
* @param[in]  *p_In        points to the input buffer for FFT input.
* @param[in]  *p_Out       points to the output buffer for FFT result.
* @return none.
* @note The real sequence is initially treated as if it were complex to perform a CFFT.
* Later, a processing stage reshapes the data to obtain half of the frequency spectrum
* in complex format. Except the first complex number that contains the two real numbers
* X[0] and X[N/2] all the data is complex. In other words, the first complex sample
* contains two real values packed.
*/
void FSP_RfftF32_512(float *pIn, float *pOut)
{
    fsp_te_instance_t S;
    float maxVal1 = 0.0f, minVal1 = 0.0f, maxVal2 = 0.0f, minVal2 = 0.0f;
    int32_t tempMax = 0, tempExp = 0;
    uint32_t fft_len = 512;
```

```
    S.te_point = kFSP_TePts256Points;   /* For cfft */

    FSP_MaxMinF32(FSP, pIn, 256, &maxVal1, &minVal1);
    if(-minVal1 > maxVal1)
        maxVal1 = -minVal1; /* Get the max value */

    FSP_MaxMinF32(FSP, pIn+256, 256, &maxVal2, &minVal2);
    if(-minVal2 > maxVal2)
        maxVal2 = -minVal2; /* Get the max value */

    if(maxVal2 > maxVal1)
        maxVal1 = maxVal2; /* Get the max absolute value */

    tempMax = *((int32_t *)(&maxVal1));
    tempExp = (tempMax >> 23) & 0xFF;
    tempExp = tempExp - 127;
    S.te_scale = -(tempExp + 1);

    FSP_CfftF32(FSP, &S, pIn, pIn, 0);

    /*  Real FFT extraction */
    fsp_stage_rfft_f32(twiddleCoef_rfft_512, fft_len/2, pIn, pOut);

}
```

5.2 1024 Floating-point Real FFT

```
/**
* @brief Real FFT for 1024 float samples data.
* @param[in]  *p_In        points to the input buffer for FFT input.
* @param[in]  *p_Out       points to the output buffer for FFT result.
* @return none.
* @note The real sequence is initially treated as if it were complex to perform a CFFT.
* Later, a processing stage reshapes the data to obtain half of the frequency spectrum
* in complex format. Except the first complex number that contains the two real numbers
```

```
* X[0] and X[N/2] all the data is complex. In other words, the first complex sample
* contains two real values packed.
*/
void FSP_RfftF32_1024(float *pIn, float *pOut)
{
    fsp_te_instance_t S = {.te_point = kFSP_TePts256Points}; /* For cfft */
    float maxVal = 0.0f, minVal = 0.0f, maxValAbs = 0.0f;
    int32_t tempMax = 0, tempExp = 0;

    /* Find the max & min value and calculate S.te_scale */
    FSP_MaxMinF32(FSP, pIn, 256, &maxVal, &minVal);
    if(-minVal > maxVal)
        maxVal = -minVal;
    maxValAbs = maxVal;
    for(uint32_t i = 256; i < 1024; )
    {
        FSP_MaxMinF32(FSP, pIn+i, 256, &maxVal, &minVal);
        if(-minVal > maxVal)
            maxVal = -minVal;
        maxValAbs = maxVal;
        i += 256;
    }

    tempMax = *((int32_t *)(&maxValAbs));
    tempExp = (tempMax >> 23) & 0xFF;
    tempExp = tempExp - 127;
    S.te_scale = -(tempExp + 1);



    /* f1=x(2*n), f2=x(2*n+1) n = 0,1,2..,N/2-1 */

    for(uint32_t j = 0; j < 2; j++)
    {
        uint32_t offset = j * 512;
```

```
    /* Separate odd and even input */
    for(uint32_t i = 0; i < 512; i++)
    {
      pOut[i+offset] = pIn[2*i+j];
    }

    /* Wait fsp IDLE */
    if(j != 0)
    {
      FSP_WaitTeOpDone(FSP);
    }

    /* Start FSP engine */
    FSP_TeStart(FSP, TE_CONFIG(S.te_point, S.te_scale, kFSP_TeDoutFpSelFloat,
kFSP_TeDinFpSelFloat,
                kFSP_TeIoModeComplexInputComplexOutput, kFSP_TeModeFft | 0),
        pOut+offset, pOut+offset);
  }



  /* Stage */
  fsp_stage_rfft_f32(twiddleCoef_rfft_512, 256, pOut, pIn);
  /* Wait fsp IDLE */
  FSP_WaitTeOpDone(FSP);
  /* Stage */
  fsp_stage_rfft_f32(twiddleCoef_rfft_512, 256, pOut+512, pIn+512);

  /* Xr = Fr + twGr r = 0,1,2,3,...,1024/2-1 */
  /* tw = cos(2*pi*r/1024) - j*sin(2*pi*r/1024) r=0,1,2,..,1024/2-1 */

  float GR,GI,FR,FI;
  float const *pTw = fft_coeff_1024;
  float *pEvenOut = pIn, *pOddOut = pIn+512;
```

```
    FR = *pEvenOut++;
    FI = *pEvenOut++;   /* the 256 sample value for even output */
    GR = *pOddOut++;
    GI = *pOddOut++;    /* the 256 sample value for odd output */

    pOut[0] = FR + GR;
    pOut[1] = FR - GR;  /* N/2 sample value */

    /* Tw(256)=0-i, output(256)=evenOut(256)+(-i)*oddOut(256) */
    pOut[512] = FI;
    pOut[513] = -GI;    /* The 256 sample value for final output.*/

    for(uint32_t i = 0; i < 255; i++)
    {
        FR = *pEvenOut++;
        FI = *pEvenOut++;
        GR = *pOddOut++;
        GI = *pOddOut++;

        pOut[2+2*i] = pTw[2+2*i] * GR - pTw[3+2*i]*GI + FR;     /* real */
        pOut[3+2*i] = pTw[2+2*i] * GI + pTw[3+2*i]*GR + FI;     /* imag */

        pOut[1022-2*i] = pTw[1022-2*i] * GR + pTw[1023-2*i] * GI + FR;  /* real */
        pOut[1023-2*i] = pTw[1023-2*i] * GR - pTw[1022-2*i] * GI - FI;  /* imag */
    }
}
```

5.3 Stage function for 512/1024 RFFT and twiddle factor

```
void fsp_stage_rfft_f32(float *pTwiddleRFFT, uint32_t fftLen, float32_t * p, float32_t * pOut)
{
  uint32_t  k;                              /* Loop Counter                  */
  float32_t twR, twI;                       /* RFFT Twiddle coefficients      */
  float32_t * pCoeff = pTwiddleRFFT;  /* Points to RFFT Twiddle factors   */
```


All information provided in this document is subject to legal disclaimers.

```
float32_t *pA = p;                    /* increasing pointer      */
float32_t *pB = p;                    /* decreasing pointer      */
float32_t xAR, xAI, xBR, xBI;         /* temporary variables     */
float32_t t1a, t1b;                   /* temporary variables     */
float32_t p0, p1, p2, p3;             /* temporary variables     */


k = fftLen - 1;

/* Pack first and last sample of the frequency domain together */

xBR = pB[0];
xBI = pB[1];
xAR = pA[0];
xAI = pA[1];

twR = *pCoeff++ ;
twI = *pCoeff++ ;

// U1 = XA(1) + XB(1); % It is real
t1a = xBR + xAR  ;

// U2 = XB(1) - XA(1); % It is imaginary
t1b = xBI + xAI  ;

// real(tw * (xB - xA)) = twR * (xBR - xAR) - twI * (xBI - xAI);
// imag(tw * (xB - xA)) = twI * (xBR - xAR) + twR * (xBI - xAI);
*pOut++ = 0.5f * ( t1a + t1b );
*pOut++ = 0.5f * ( t1a - t1b );

// XA(1) = 1/2*( U1 - imag(U2) +  i*( U1 +imag(U2) ));
pB  = p + 2*k;
pA += 2;

do
```

```
{
  /*
    function X = my_split_rfft(X, ifftFlag)
    % X is a series of real numbers
    L  = length(X);
    XC = X(1:2:end) +i*X(2:2:end);
    XA = fft(XC);
    XB = conj(XA([1 end:-1:2]));
    TW = i*exp(-2*pi*i*[0:L/2-1]/L).';
    for l = 2:L/2
      XA(l) = 1/2 * (XA(l) + XB(l) + TW(l) * (XB(l) - XA(l)));
    end
    XA(1) = 1/2* (XA(1) + XB(1) + TW(1) * (XB(1) - XA(1))) + i*( 1/2*( XA(1) + XB(1) + i*( XA(1) - XB(1))));
    X = XA;
  */

  xBI = pB[1];
  xBR = pB[0];
  xAR = pA[0];
  xAI = pA[1];

  twR = *pCoeff++;
  twI = *pCoeff++;

  t1a = xBR - xAR ;
  t1b = xBI + xAI ;

  // real(tw * (xB - xA)) = twR * (xBR - xAR) - twI * (xBI - xAI);
  // imag(tw * (xB - xA)) = twI * (xBR - xAR) + twR * (xBI - xAI);
  p0 = twR * t1a;
  p1 = twI * t1a;
  p2 = twR * t1b;
  p3 = twI * t1b;

  *pOut++ = 0.5f * (xAR + xBR + p0 + p3 ); //xAR
```

```
      *pOut++ = 0.5f * (xAI - xBI + p1 - p2 ); //xAI

      pA += 2;
      pB -= 2;
      k--;
   } while(k > 0u);
}

/**
* Example code for 512 Floating-point RFFT Twiddle factors Generation:
* <pre>TW = exp(2*pi*i*[0:L/2-1]/L - pi/2*i).'
* Real and Imag values are in interleaved fashion
*/
const float32_t twiddleCoef_rfft_512[512] = {
    0.000000000f,  1.000000000f,
    0.012271538f,  0.999924702f,
    0.024541229f,  0.999698819f,
    0.036807223f,  0.999322385f,
    0.049067674f,  0.998795456f,
    0.061320736f,  0.998118113f,
    0.073564564f,  0.997290457f,
    0.085797312f,  0.996312612f,
    0.098017140f,  0.995184727f,
    0.110222207f,  0.993906970f,
    0.122410675f,  0.992479535f,
    0.134580709f,  0.990902635f,
    0.146730474f,  0.989176510f,
    0.158858143f,  0.987301418f,
    0.170961889f,  0.985277642f,
    0.183039888f,  0.983105487f,
    0.195090322f,  0.980785280f,
    0.207111376f,  0.978317371f,
    0.219101240f,  0.975702130f,
    0.231058108f,  0.972939952f,
    0.242980180f,  0.970031253f,
```

```
0.254865660f, 0.966976471f,
0.266712757f, 0.963776066f,
0.278519689f, 0.960430519f,
0.290284677f, 0.956940336f,
0.302005949f, 0.953306040f,
0.313681740f, 0.949528181f,
0.325310292f, 0.945607325f,
0.336889853f, 0.941544065f,
0.348418680f, 0.937339012f,
0.359895037f, 0.932992799f,
0.371317194f, 0.928506080f,
0.382683432f, 0.923879533f,
0.393992040f, 0.919113852f,
0.405241314f, 0.914209756f,
0.416429560f, 0.909167983f,
0.427555093f, 0.903989293f,
0.438616239f, 0.898674466f,
0.449611330f, 0.893224301f,
0.460538711f, 0.887639620f,
0.471396737f, 0.881921264f,
0.482183772f, 0.876070094f,
0.492898192f, 0.870086991f,
0.503538384f, 0.863972856f,
0.514102744f, 0.857728610f,
0.524589683f, 0.851355193f,
0.534997620f, 0.844853565f,
0.545324988f, 0.838224706f,
0.555570233f, 0.831469612f,
0.565731811f, 0.824589303f,
0.575808191f, 0.817584813f,
0.585797857f, 0.810457198f,
0.595699304f, 0.803207531f,
0.605511041f, 0.795836905f,
0.615231591f, 0.788346428f,
0.624859488f, 0.780737229f,
```

```
0.634393284f, 0.773010453f,
0.643831543f, 0.765167266f,
0.653172843f, 0.757208847f,
0.662415778f, 0.749136395f,
0.671558955f, 0.740951125f,
0.680600998f, 0.732654272f,
0.689540545f, 0.724247083f,
0.698376249f, 0.715730825f,
0.707106781f, 0.707106781f,
0.715730825f, 0.698376249f,
0.724247083f, 0.689540545f,
0.732654272f, 0.680600998f,
0.740951125f, 0.671558955f,
0.749136395f, 0.662415778f,
0.757208847f, 0.653172843f,
0.765167266f, 0.643831543f,
0.773010453f, 0.634393284f,
0.780737229f, 0.624859488f,
0.788346428f, 0.615231591f,
0.795836905f, 0.605511041f,
0.803207531f, 0.595699304f,
0.810457198f, 0.585797857f,
0.817584813f, 0.575808191f,
0.824589303f, 0.565731811f,
0.831469612f, 0.555570233f,
0.838224706f, 0.545324988f,
0.844853565f, 0.534997620f,
0.851355193f, 0.524589683f,
0.857728610f, 0.514102744f,
0.863972856f, 0.503538384f,
0.870086991f, 0.492898192f,
0.876070094f, 0.482183772f,
0.881921264f, 0.471396737f,
0.887639620f, 0.460538711f,
0.893224301f, 0.449611330f,
```

```
0.898674466f, 0.438616239f,
0.903989293f, 0.427555093f,
0.909167983f, 0.416429560f,
0.914209756f, 0.405241314f,
0.919113852f, 0.393992040f,
0.923879533f, 0.382683432f,
0.928506080f, 0.371317194f,
0.932992799f, 0.359895037f,
0.937339012f, 0.348418680f,
0.941544065f, 0.336889853f,
0.945607325f, 0.325310292f,
0.949528181f, 0.313681740f,
0.953306040f, 0.302005949f,
0.956940336f, 0.290284677f,
0.960430519f, 0.278519689f,
0.963776066f, 0.266712757f,
0.966976471f, 0.254865660f,
0.970031253f, 0.242980180f,
0.972939952f, 0.231058108f,
0.975702130f, 0.219101240f,
0.978317371f, 0.207111376f,
0.980785280f, 0.195090322f,
0.983105487f, 0.183039888f,
0.985277642f, 0.170961889f,
0.987301418f, 0.158858143f,
0.989176510f, 0.146730474f,
0.990902635f, 0.134580709f,
0.992479535f, 0.122410675f,
0.993906970f, 0.110222207f,
0.995184727f, 0.098017140f,
0.996312612f, 0.085797312f,
0.997290457f, 0.073564564f,
0.998118113f, 0.061320736f,
0.998795456f, 0.049067674f,
0.999322385f, 0.036807223f,
```

```
0.999698819f,  0.024541229f,
0.999924702f,  0.012271538f,
1.000000000f,  0.000000000f,
0.999924702f, -0.012271538f,
0.999698819f, -0.024541229f,
0.999322385f, -0.036807223f,
0.998795456f, -0.049067674f,
0.998118113f, -0.061320736f,
0.997290457f, -0.073564564f,
0.996312612f, -0.085797312f,
0.995184727f, -0.098017140f,
0.993906970f, -0.110222207f,
0.992479535f, -0.122410675f,
0.990902635f, -0.134580709f,
0.989176510f, -0.146730474f,
0.987301418f, -0.158858143f,
0.985277642f, -0.170961889f,
0.983105487f, -0.183039888f,
0.980785280f, -0.195090322f,
0.978317371f, -0.207111376f,
0.975702130f, -0.219101240f,
0.972939952f, -0.231058108f,
0.970031253f, -0.242980180f,
0.966976471f, -0.254865660f,
0.963776066f, -0.266712757f,
0.960430519f, -0.278519689f,
0.956940336f, -0.290284677f,
0.953306040f, -0.302005949f,
0.949528181f, -0.313681740f,
0.945607325f, -0.325310292f,
0.941544065f, -0.336889853f,
0.937339012f, -0.348418680f,
0.932992799f, -0.359895037f,
0.928506080f, -0.371317194f,
0.923879533f, -0.382683432f,
```

```
0.919113852f, -0.393992040f,
0.914209756f, -0.405241314f,
0.909167983f, -0.416429560f,
0.903989293f, -0.427555093f,
0.898674466f, -0.438616239f,
0.893224301f, -0.449611330f,
0.887639620f, -0.460538711f,
0.881921264f, -0.471396737f,
0.876070094f, -0.482183772f,
0.870086991f, -0.492898192f,
0.863972856f, -0.503538384f,
0.857728610f, -0.514102744f,
0.851355193f, -0.524589683f,
0.844853565f, -0.534997620f,
0.838224706f, -0.545324988f,
0.831469612f, -0.555570233f,
0.824589303f, -0.565731811f,
0.817584813f, -0.575808191f,
0.810457198f, -0.585797857f,
0.803207531f, -0.595699304f,
0.795836905f, -0.605511041f,
0.788346428f, -0.615231591f,
0.780737229f, -0.624859488f,
0.773010453f, -0.634393284f,
0.765167266f, -0.643831543f,
0.757208847f, -0.653172843f,
0.749136395f, -0.662415778f,
0.740951125f, -0.671558955f,
0.732654272f, -0.680600998f,
0.724247083f, -0.689540545f,
0.715730825f, -0.698376249f,
0.707106781f, -0.707106781f,
0.698376249f, -0.715730825f,
0.689540545f, -0.724247083f,
0.680600998f, -0.732654272f,
```

```
0.671558955f, -0.740951125f,
0.662415778f, -0.749136395f,
0.653172843f, -0.757208847f,
0.643831543f, -0.765167266f,
0.634393284f, -0.773010453f,
0.624859488f, -0.780737229f,
0.615231591f, -0.788346428f,
0.605511041f, -0.795836905f,
0.595699304f, -0.803207531f,
0.585797857f, -0.810457198f,
0.575808191f, -0.817584813f,
0.565731811f, -0.824589303f,
0.555570233f, -0.831469612f,
0.545324988f, -0.838224706f,
0.534997620f, -0.844853565f,
0.524589683f, -0.851355193f,
0.514102744f, -0.857728610f,
0.503538384f, -0.863972856f,
0.492898192f, -0.870086991f,
0.482183772f, -0.876070094f,
0.471396737f, -0.881921264f,
0.460538711f, -0.887639620f,
0.449611330f, -0.893224301f,
0.438616239f, -0.898674466f,
0.427555093f, -0.903989293f,
0.416429560f, -0.909167983f,
0.405241314f, -0.914209756f,
0.393992040f, -0.919113852f,
0.382683432f, -0.923879533f,
0.371317194f, -0.928506080f,
0.359895037f, -0.932992799f,
0.348418680f, -0.937339012f,
0.336889853f, -0.941544065f,
0.325310292f, -0.945607325f,
0.313681740f, -0.949528181f,
```

```
    0.302005949f, -0.953306040f,
    0.290284677f, -0.956940336f,
    0.278519689f, -0.960430519f,
    0.266712757f, -0.963776066f,
    0.254865660f, -0.966976471f,
    0.242980180f, -0.970031253f,
    0.231058108f, -0.972939952f,
    0.219101240f, -0.975702130f,
    0.207111376f, -0.978317371f,
    0.195090322f, -0.980785280f,
    0.183039888f, -0.983105487f,
    0.170961889f, -0.985277642f,
    0.158858143f, -0.987301418f,
    0.146730474f, -0.989176510f,
    0.134580709f, -0.990902635f,
    0.122410675f, -0.992479535f,
    0.110222207f, -0.993906970f,
    0.098017140f, -0.995184727f,
    0.085797312f, -0.996312612f,
    0.073564564f, -0.997290457f,
    0.061320736f, -0.998118113f,
    0.049067674f, -0.998795456f,
    0.036807223f, -0.999322385f,
    0.024541229f, -0.999698819f,
    0.012271538f, -0.999924702f
};

/**
* Example code for 1024 Floating-point RFFT Twiddle factors Generation:
* exp(-2*pi*[0:511]*i/1024)
* Real and Imag values are in interleaved fashion
*/
const float fft_coeff_1024[1024] = {
1, 0, 0.999981175282601, -0.00613588464915448, 0.999924701839145, -0.0122715382857199, 0.999830581795823, -
0.0184067299058048, 0.999698818696204, -0.0245412285229123, 0.999529417501093, -0.0306748031766366, 0.999322384588350,
-0.0368072229413588, 0.999077727752645, -0.0429382569349408, 0.998795456205172, -0.0490676743274180,
```

0.998475580573295, -0.0551952443496899, 0.998118112900149, -0.0613207363022086, 0.997723066644192, -
0.0674439195636641, 0.997290456678690, -0.0735645635996674, 0.996820299291166, -0.0796824379714301, 0.996312612182778,
-0.0857973123444399, 0.995767414467660, -0.0919089564971327, 0.995184726672197, -0.0980171403295606,
0.994564570734255, -0.104121633872055, 0.993906970002356, -0.110222207293883, 0.993211949234795, -0.116318630911905,
0.992479534598710, -0.122410675199216, 0.991709753669100, -0.128498110793793, 0.990902635427780, -0.134580708507126,
0.990058210262297, -0.140658239332849, 0.989176509964781, -0.146730474455362, 0.988257567730750, -0.152797185258443,
0.987301418157858, -0.158858143333861, 0.986308097244599, -0.164913120489970, 0.985277642388941, -0.170961888760301,
0.984210092386929, -0.177004220412149, 0.983105487431216, -0.183039887955141, 0.981963869109555, -0.189068664149806,
0.980785280403230, -0.195090322016128, 0.979569765685441, -0.201104634842092, 0.978317370719628, -0.207111376192219,
0.977028142657754, -0.213110319916091, 0.975702130038529, -0.219101240156870, 0.974339382785576, -0.225083911359793,
0.972939952205560, -0.231058108280671, 0.971503890986252, -0.237023605994367, 0.970031253194544, -0.242980179903264,
0.968522094274417, -0.248927605745720, 0.966976471044852, -0.254865659604515, 0.965394441697689, -0.260794117915276,
0.963776065795440, -0.266712757474898, 0.962121404269042, -0.272621355449949, 0.960430519415566, -0.278519689385053,
0.958703474895872, -0.284407537211272, 0.956940335732209, -0.290284677254462, 0.955141168305771, -0.296150888243624,
0.953306040354194, -0.302005949319228, 0.951435020969008, -0.307849640041535, 0.949528180593037, -0.313681740398892,
0.947585591017741, -0.319502030816016, 0.945607325380521, -0.325310292162263, 0.943593458161960, -0.331106305759876,
0.941544065183021, -0.336889853392220, 0.939459223602190, -0.342660717311994, 0.937339011912575, -0.348418680249435,
0.935183509938948, -0.354163525420490, 0.932992798834739, -0.359895036534988, 0.930766961078984, -0.365612997804774,
0.928506080473216, -0.371317193951838, 0.926210242138311, -0.377007410216418, 0.923879532511287, -0.382683432365090,
0.921514039342042, -0.388345046698826, 0.919113851690058, -0.393992040061048, 0.916679059921043, -0.399624199845647,
0.914209755703531, -0.405241314004990, 0.911706032005430, -0.410843171057904, 0.909167983090522, -0.416429560097637,
0.906595704514915, -0.422000270799800, 0.903989293123443, -0.427555093430282, 0.901348847046022, -0.433093818853152,
0.898674465693954, -0.438616238538528, 0.895966249756185, -0.444122144570429, 0.893224301195515, -0.449611329654607,
0.890448723244758, -0.455083587126344, 0.887639620402854, -0.460538710958240, 0.884797098430938, -0.465976495767966,
0.881921264348355, -0.471396736825998, 0.879012226428634, -0.476799230063322, 0.876070094195407, -0.482183772079123,
0.873094978418290, -0.487550160148436, 0.870086991108712, -0.492898192229784, 0.867046245515693, -0.498227666972782,
0.863972856121587, -0.503538383725718, 0.860866938637767, -0.508830142543107, 0.857728610000272, -0.514102744193222,
0.854557988365401, -0.519355990165590, 0.851355193105265, -0.524589682678469, 0.848120344803297, -0.529803624686295,
0.844853565249707, -0.534997619887097, 0.841554977436898, -0.540171472729893, 0.838224705554838, -0.545324988422047,
0.834862874986380, -0.550457972936605, 0.831469612302545, -0.555570233019602, 0.828045045257756, -0.560661576197336,
0.824589302785025, -0.565731810783613, 0.821102514991105, -0.570780745886967, 0.817584813151584, -0.575808191417845,
0.814036329705948, -0.580813958095765, 0.810457198252595, -0.585797857456439, 0.806847553543799, -0.590759701858874,
0.803207531480645, -0.595699304492433, 0.799537269107905, -0.600616479383869, 0.795836904608884, -0.605511041404326,
0.792106577300212, -0.610382806276310, 0.788346427626606, -0.615231590580627, 0.784556597155575, -0.620057211763289,
0.780737228572095, -0.624859488142386, 0.776888465673232, -0.629638238914927, 0.773010453362737, -0.634393284163646,
0.769103337645580, -0.639124444863776, 0.765167265622459, -0.643831542889791, 0.761202385484262, -0.648514401022112,
0.757208846506485, -0.653172842953777, 0.753186799043613, -0.657806693297079, 0.749136394523459, -0.662415777590172,
0.745057785441466, -0.666999922303638, 0.740951125354959, -0.671558954847018, 0.736816568877370, -0.676092703575316,
0.732654271672413, -0.680600997795453, 0.728464390448225, -0.685083667772700, 0.724247082951467, -0.689540544737067,
0.720002507961382, -0.693971460889654, 0.715730825283819, -0.698376249408973, 0.711432195745216, -0.702754744457225,
0.707106781186548, -0.707106781186548, 0.702754744457225, -0.711432195745216, 0.698376249408973, -0.715730825283819,
0.693971460889654, -0.720002507961382, 0.689540544737067, -0.724247082951467, 0.685083667772700, -0.728464390448225,
0.680600997795453, -0.732654271672413, 0.676092703575316, -0.736816568877370, 0.671558954847018, -0.740951125354959,
0.666999922303638, -0.745057785441466, 0.662415777590172, -0.749136394523459, 0.657806693297079, -0.753186799043613,
0.653172842953777, -0.757208846506485, 0.648514401022113, -0.761202385484262, 0.643831542889792, -0.765167265622459,
0.639124444863776, -0.769103337645580, 0.634393284163646, -0.773010453362737, 0.629638238914927, -0.776888465673232,
0.624859488142387, -0.780737228572094, 0.620057211763289, -0.784556597155575, 0.615231590580627, -0.788346427626606,
0.610382806276310, -0.792106577300212, 0.605511041404326, -0.795836904608884, 0.600616479383869, -0.799537269107905,
0.595699304492434, -0.803207531480645, 0.590759701858874, -0.806847553543799, 0.585797857456439, -0.810457198252595,

0.580813958095765, -0.814036329705948, 0.575808191417845, -0.817584813151584, 0.570780745886967, -0.821102514991105, 0.565731810783613, -0.824589302785025, 0.560661576197336, -0.828045045257756, 0.555570233019602, -0.831469612302545, 0.550457972936605, -0.834862874986380, 0.545324988422047, -0.838224705554838, 0.540171472729893, -0.841554977436898, 0.534997619887097, -0.844853565249707, 0.529803624686295, -0.848120344803297, 0.524589682678469, -0.851355193105265, 0.519355990165590, -0.854557988365401, 0.514102744193222, -0.857728610000272, 0.508830142543107, -0.860866938637767, 0.503538383725718, -0.863972856121587, 0.498227666972782, -0.867046245515693, 0.492898192229784, -0.870086991108711, 0.487550160148436, -0.873094978418290, 0.482183772079123, -0.876070094195407, 0.476799230063322, -0.879012226428633, 0.471396736825998, -0.881921264348355, 0.465976495767966, -0.884797098430938, 0.460538710958240, -0.887639620402854, 0.455083587126344, -0.890448723244758, 0.449611329654607, -0.893224301195515, 0.444122144570429, -0.895966249756185, 0.438616238538528, -0.898674465693954, 0.433093818853152, -0.901348847046022, 0.427555093430282, -0.903989293123443, 0.422000270799800, -0.906595704514915, 0.416429560097637, -0.909167983090522, 0.410843171057904, -0.911706032005430, 0.405241314004990, -0.914209755703531, 0.399624199845647, -0.916679059921043, 0.393992040061048, -0.919113851690058, 0.388345046698826, -0.921514039342042, 0.382683432365090, -0.923879532511287, 0.377007410216418, -0.926210242138311, 0.371317193951838, -0.928506080473216, 0.365612997804774, -0.930766961078984, 0.359895036534988, -0.932992798834739, 0.354163525420491, -0.935183509938948, 0.348418680249435, -0.937339011912575, 0.342660717311994, -0.939459223602190, 0.336889853392220, -0.941544065183021, 0.331106305759876, -0.943593458161960, 0.325310292162263, -0.945607325380521, 0.319502030816016, -0.947585591017741, 0.313681740398892, -0.949528180593037, 0.307849640041535, -0.951435020969008, 0.302005949319228, -0.953306040354194, 0.296150888243624, -0.955141168305771, 0.290284677254462, -0.956940335732209, 0.284407537211272, -0.958703474895872, 0.278519689385053, -0.960430519415566, 0.272621355449949, -0.962121404269042, 0.266712757474898, -0.963776065795440, 0.260794117915276, -0.965394441697689, 0.254865659604515, -0.966976471044852, 0.248927605745720, -0.968522094274417, 0.242980179903264, -0.970031253194544, 0.237023605994367, -0.971503890986252, 0.231058108280671, -0.972939952205560, 0.225083911359793, -0.974339382785576, 0.219101240156870, -0.975702130038529, 0.213110319916091, -0.977028142657754, 0.207111376192219, -0.978317370719628, 0.201104634842092, -0.979569765685441, 0.195090322016128, -0.980785280403230, 0.189068664149806, -0.981963869109555, 0.183039887955141, -0.983105487431216, 0.177004220412149, -0.984210092386929, 0.170961888760301, -0.985277642388941, 0.164913120489970, -0.986308097244599, 0.158858143333861, -0.987301418157858, 0.152797185258443, -0.988257567730750, 0.146730474455362, -0.989176509964781, 0.140658239332849, -0.990058210262297, 0.134580708507126, -0.990902635427780, 0.128498110793793, -0.991709753669100, 0.122410675199216, -0.992479534598710, 0.116318630911905, -0.993211949234795, 0.110222207293883, -0.993906970002356, 0.104121633872055, -0.994564570734255, 0.0980171403295608, -0.995184726672197, 0.0919089564971327, -0.995767414467660, 0.0857973123444399, -0.996312612182778, 0.0796824379714301, -0.996820299291166, 0.0735645635996675, -0.997290456678690, 0.0674439195636641, -0.997723066644192, 0.0613207363022087, -0.998118112900149, 0.0551952443496900, -0.998475580573295, 0.0490676743274181, -0.998795456205172, 0.0429382569349410, -0.999077727752645, 0.0368072229413590, -0.999322384588350, 0.0306748031766366, -0.999529417501093, 0.0245412285229123, -0.999698818696204, 0.0184067299058048, -0.999830581795823, 0.0122715382857199, -0.999924701839145, 0.00613588464915452, -0.999981175282601, 6.12323399573677e-17, -1, -0.00613588464915439, -0.999981175282601, -0.0122715382857198, -0.999924701839145, -0.0184067299058047, -0.999830581795823, -0.0245412285229121, -0.999698818696204, -0.0306748031766365, -0.999529417501093, -0.0368072229413589, -0.999322384588350, -0.0429382569349408, -0.999077727752645, -0.0490676743274180, -0.998795456205172, -0.0551952443496899, -0.998475580573295, -0.0613207363022085, -0.998118112900149, -0.0674439195636640, -0.997723066644192, -0.0735645635996673, -0.997290456678690, -0.0796824379714300, -0.996820299291166, -0.0857973123444398, -0.996312612182778, -0.0919089564971326, -0.995767414467660, -0.0980171403295607, -0.995184726672197, -0.104121633872055, -0.994564570734255, -0.110222207293883, -0.993906970002356, -0.116318630911905, -0.993211949234795, -0.122410675199216, -0.992479534598710, -0.128498110793793, -0.991709753669100, -0.134580708507126, -0.990902635427780, -0.140658239332849, -0.990058210262297, -0.146730474455362, -0.989176509964781, -0.152797185258443, -0.988257567730750, -0.158858143333861, -0.987301418157858, -0.164913120489970, -0.986308097244599, -0.170961888760301, -0.985277642388941, -0.177004220412149, -0.984210092386929, -0.183039887955141, -0.983105487431216, -0.189068664149806, -0.981963869109555, -0.195090322016128, -0.980785280403230, -0.201104634842092, -0.979569765685441, -0.207111376192218, -0.978317370719628, -0.213110319916091, -0.977028142657754, -0.219101240156870, -0.975702130038529, -0.225083911359793, -0.974339382785576, -0.231058108280671, -0.972939952205560, -0.237023605994367, -0.971503890986252, -0.242980179903264, -0.970031253194544, -

0.248927605745720, -0.968522094274417, -0.254865659604515, -0.966976471044852, -0.260794117915275, -0.965394441697689, -
0.266712757474898, -0.963776065795440, -0.272621355449949, -0.962121404269042, -0.278519689385053, -0.960430519415566, -
0.284407537211272, -0.958703474895872, -0.290284677254462, -0.956940335732209, -0.296150888243624, -0.955141168305771, -
0.302005949319228, -0.953306040354194, -0.307849640041535, -0.951435020969008, -0.313681740398891, -0.949528180593037, -
0.319502030816016, -0.947585591017741, -0.325310292162263, -0.945607325380521, -0.331106305759876, -0.943593458161960, -
0.336889853392220, -0.941544065183021, -0.342660717311994, -0.939459223602190, -0.348418680249434, -0.937339011912575, -
0.354163525420490, -0.935183509938948, -0.359895036534988, -0.932992798834739, -0.365612997804774, -0.930766961078984, -
0.371317193951838, -0.928506080473216, -0.377007410216418, -0.926210242138311, -0.382683432365090, -0.923879532511287, -
0.388345046698826, -0.921514039342042, -0.393992040061048, -0.919113851690058, -0.399624199845647, -0.916679059921043, -
0.405241314004990, -0.914209755703531, -0.410843171057904, -0.911706032005430, -0.416429560097637, -0.909167983090523, -
0.422000270799800, -0.906595704514915, -0.427555093430282, -0.903989293123443, -0.433093818853152, -0.901348847046022, -
0.438616238538527, -0.898674465693954, -0.444122144570429, -0.895966249756185, -0.449611329654607, -0.893224301195515, -
0.455083587126344, -0.890448723244758, -0.460538710958240, -0.887639620402854, -0.465976495767966, -0.884797098430938, -
0.471396736825998, -0.881921264348355, -0.476799230063322, -0.879012226428634, -0.482183772079123, -0.876070094195407, -
0.487550160148436, -0.873094978418290, -0.492898192229784, -0.870086991108712, -0.498227666972782, -0.867046245515693, -
0.503538383725718, -0.863972856121587, -0.508830142543107, -0.860866938637767, -0.514102744193222, -0.857728610000272, -
0.519355990165590, -0.854557988365401, -0.524589682678469, -0.851355193105265, -0.529803624686295, -0.848120344803297, -
0.534997619887097, -0.844853565249707, -0.540171472729893, -0.841554977436898, -0.545324988422046, -0.838224705554838, -
0.550457972936605, -0.834862874986380, -0.555570233019602, -0.831469612302545, -0.560661576197336, -0.828045045257756, -
0.565731810783613, -0.824589302785025, -0.570780745886967, -0.821102514991105, -0.575808191417845, -0.817584813151584, -
0.580813958095764, -0.814036329705949, -0.585797857456439, -0.810457198252595, -0.590759701858874, -0.806847553543799, -
0.595699304492433, -0.803207531480645, -0.600616479383869, -0.799537269107905, -0.605511041404325, -0.795836904608884, -
0.610382806276310, -0.792106577300212, -0.615231590580627, -0.788346427626606, -0.620057211763289, -0.784556597155575, -
0.624859488142386, -0.780737228572095, -0.629638238914927, -0.776888465673232, -0.634393284163645, -0.773010453362737, -
0.639124444863776, -0.769103337645580, -0.643831542889791, -0.765167265622459, -0.648514401022112, -0.761202385484262, -
0.653172842953777, -0.757208846506485, -0.657806693297079, -0.753186799043613, -0.662415777590172, -0.749136394523459, -
0.666999922303637, -0.745057785441466, -0.671558954847018, -0.740951125354959, -0.676092703575316, -0.736816568877370, -
0.680600997795453, -0.732654271672413, -0.685083667772700, -0.728464390448225, -0.689540544737067, -0.724247082951467, -
0.693971460889654, -0.720002507961382, -0.698376249408973, -0.715730825283819, -0.702754744457225, -0.711432195745217, -
0.707106781186548, -0.707106781186548, -0.711432195745217, -0.702754744457225, -0.715730825283819, -0.698376249408973, -
0.720002507961382, -0.693971460889654, -0.724247082951467, -0.689540544737067, -0.728464390448225, -0.685083667772700, -
0.732654271672413, -0.680600997795453, -0.736816568877370, -0.676092703575316, -0.740951125354959, -0.671558954847019, -
0.745057785441466, -0.666999922303638, -0.749136394523459, -0.662415777590172, -0.753186799043612, -0.657806693297079, -
0.757208846506485, -0.653172842953777, -0.761202385484262, -0.648514401022113, -0.765167265622459, -0.643831542889791, -
0.769103337645580, -0.639124444863776, -0.773010453362737, -0.634393284163646, -0.776888465673232, -0.629638238914927, -
0.780737228572095, -0.624859488142386, -0.784556597155575, -0.620057211763289, -0.788346427626606, -0.615231590580627, -
0.792106577300212, -0.610382806276310, -0.795836904608884, -0.605511041404326, -0.799537269107905, -0.600616479383869, -
0.803207531480645, -0.595699304492434, -0.806847553543799, -0.590759701858874, -0.810457198252595, -0.585797857456439, -
0.814036329705948, -0.580813958095765, -0.817584813151584, -0.575808191417845, -0.821102514991105, -0.570780745886967, -
0.824589302785025, -0.565731810783614, -0.828045045257756, -0.560661576197336, -0.831469612302545, -0.555570233019602, -
0.834862874986380, -0.550457972936605, -0.838224705554838, -0.545324988422046, -0.841554977436898, -0.540171472729893, -
0.844853565249707, -0.534997619887097, -0.848120344803297, -0.529803624686295, -0.851355193105265, -0.524589682678469, -
0.854557988365400, -0.519355990165590, -0.857728610000272, -0.514102744193222, -0.860866938637767, -0.508830142543107, -
0.863972856121587, -0.503538383725718, -0.867046245515693, -0.498227666972782, -0.870086991108711, -0.492898192229784, -
0.873094978418290, -0.487550160148436, -0.876070094195407, -0.482183772079123, -0.879012226428634, -0.476799230063322, -
0.881921264348355, -0.471396736825998, -0.884797098430938, -0.465976495767966, -0.887639620402854, -0.460538710958240, -
0.890448723244758, -0.455083587126344, -0.893224301195515, -0.449611329654607, -0.895966249756185, -0.444122144570429, -
0.898674465693954, -0.438616238538528, -0.901348847046022, -0.433093818853152, -0.903989293123443, -0.427555093430282, -
0.906595704514915, -0.422000270799800, -0.909167983090522, -0.416429560097637, -0.911706032005430, -0.410843171057904, -

```
0.914209755703531, -0.405241314004990, -0.916679059921043, -0.399624199845647, -0.919113851690058, -0.393992040061048, -
0.921514039342042, -0.388345046698827, -0.923879532511287, -0.382683432365090, -0.926210242138311, -0.377007410216418, -
0.928506080473216, -0.371317193951838, -0.930766961078984, -0.365612997804774, -0.932992798834739, -0.359895036534988, -
0.935183509938948, -0.354163525420490, -0.937339011912575, -0.348418680249435, -0.939459223602190, -0.342660717311994, -
0.941544065183021, -0.336889853392220, -0.943593458161960, -0.331106305759877, -0.945607325380521, -0.325310292162263, -
0.947585591017741, -0.319502030816016, -0.949528180593037, -0.313681740398891, -0.951435020969008, -0.307849640041535, -
0.953306040354194, -0.302005949319228, -0.955141168305771, -0.296150888243624, -0.956940335732209, -0.290284677254462, -
0.958703474895872, -0.284407537211272, -0.960430519415566, -0.278519689385053, -0.962121404269042, -0.272621355449949, -
0.963776065795440, -0.266712757474899, -0.965394441697689, -0.260794117915276, -0.966976471044852, -0.254865659604515, -
0.968522094274417, -0.248927605745720, -0.970031253194544, -0.242980179903264, -0.971503890986252, -0.237023605994367, -
0.972939952205560, -0.231058108280671, -0.974339382785576, -0.225083911359793, -0.975702130038529, -0.219101240156870, -
0.977028142657754, -0.213110319916091, -0.978317370719628, -0.207111376192219, -0.979569765685441, -0.201104634842092, -
0.980785280403230, -0.195090322016129, -0.981963869109555, -0.189068664149806, -0.983105487431216, -0.183039887955141, -
0.984210092386929, -0.177004220412149, -0.985277642388941, -0.170961888760301, -0.986308097244599, -0.164913120489970, -
0.987301418157858, -0.158858143333861, -0.988257567730750, -0.152797185258444, -0.989176509964781, -0.146730474455362, -
0.990058210262297, -0.140658239332850, -0.990902635427780, -0.134580708507126, -0.991709753669100, -0.128498110793793, -
0.992479534598710, -0.122410675199216, -0.993211949234795, -0.116318630911905, -0.993906970002356, -0.110222207293883, -
0.994564570734255, -0.104121633872055, -0.995184726672197, -0.0980171403295608, -0.995767414467660, -
0.0919089564971328, -0.996312612182778, -0.0857973123444402, -0.996820299291166, -0.0796824379714302, -
0.997290456678690, -0.0735645635996677, -0.997723066644192, -0.0674439195636642, -0.998118112900149, -
0.0613207363022085, -0.998475580573295, -0.0551952443496901, -0.998795456205172, -0.0490676743274180, -
0.999077727752645, -0.0429382569349410, -0.999322384588350, -0.0368072229413588, -0.999529417501093, -
0.0306748031766369, -0.999698818696204, -0.0245412285229123, -0.999830581795823, -0.0184067299058051, -
0.999924701839145, -0.0122715382857200, -0.999981175282601, -0.00613588464915480};
```

6. Legal information

6.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

6.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

6.3 Licenses

Purchase of NXP <xxx> components
<License statement text>

6.4 Patents

Notice is herewith given that the subject device uses one or more of the following patents and that each of these patents may have corresponding patents in other jurisdictions.

<Patent ID> — owned by <Company name>

6.5 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

<Name> — is a trademark of NXP Semiconductors N.V.

7. Index

Contents