Sine, Cosine on the TMS320C2xx

Application Report
Literature Number: BPRA047

TEXAS INSTRUMENTS

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain application using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

TRADEMARKS

TI is a trademark of Texas Instruments Incorporated.

Other brands and names are the property of their respective owners.

Table of Contents

1. Overview

Sine-wave generators are fundamental building blocks of signal processing systems for use in control applications. This application note describes two different methods for implementing a digital Sine wave generator using the TMS320C2xx. The first method is a fast direct table look-up with linear interpolation to provide Sine waves with a minimum harmonic distortion. The second method realizes the Sine/Cosine functions with a mathematical series.

Two methods respond to different constraints. The table look-up method is fast, with minimum error solution and is important in ROM. The second method is fast with minimum precision and minimum program (ROM) solution.

2. Sine/Cosine with fast table direct look-up and linear interpolation

2.1 Presentation

This method realizes the Sine/Cosine functions with a table of 100 Sine period values. The method includes linear interpolation with a fixed step table to provide a minimum harmonic distortion. The same table is used for Sine and Cosine. An extended table is used for the Cosine + Sine function. This table is in program memory.

2.2 Convention

Two different versions of each function are presented in this document: fully C-compatible functions and assembly calling functions. Different examples of assembly and C program calling Sine and Cosine are in the Annex.

2.2.1 Fully C-compatible functions

Fully C-compatible functions use C convention to use the stack for parameters passed to the functions. Parameters returned by functions are passed by pointer. Responsibilities of a called function by C are managed. Stack pointer AR1 is well positioned and the return address of the hardware stack is popped in case of a C interrupt routine event using C-function features (stack). The frame pointer is not modified. Register AR6/AR7 are not used.

2.2.2 Assembly-compatible functions

Assembly-compatible functions do not use their own variables; variables used are in a software stack.. Arguments are passed by the stack and the AR1 point to the stack just after the last argument. In return from function, the results are in the stack. Registers AR0, AR6, and AR7 are not modified.

2.3 Interpolation

The formula for calculating the table interpolation value COS(X) of an angle X is:

$$Y = y_i + \underbrace{\frac{X - x_i}{x_{i+1} - x_i}}_{r = ratio} (y_{i+1} - y_i)$$

Figure 1: formula

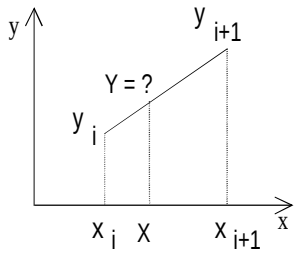


Figure 2: interpolation

where:

$\{x_i\}$ = {first coordinates of the table},

$\{y_i\}$ = {second coordinates of the table},

i chosen so that $x_i < X < x_{i+1}$.

Table interpolation can be split into two steps:

- table look-up: it consists of looking through the whole table in order to find out which interval $[x_i, x_{i+1}]$ the considered angle X is located,
  with $x_i < X < x_{i+1}$.
- interpolation: it consists of realizing the above mentioned calculation in order to obtain Y.

2.4 Functions

Functions SINE, COSINE and SINE + COSINE are in the Annex with a main example. Conventions to interface with this function are:

- Input: ANGLE parameters is in the stack pointed by AR1
  The value of this angle is unsigned:
  0 º <-> 0000h
  90 º <-> 4000h
  180 º <-> 8000h
  240 º <-> C000h
- Output : COS(angle) or SIN(angle) are in ACCUMULATOR in Q15 format (-1 to $1-2^{15}$)

2.5 Processor utilization (maximum)

Function	Cycles	Execution Time
SIN Assembly call	**35**	**1.75µs**
COS Assembly call	**38**	**1.90µs**
SIN Fully C compatible	42	2.10µs
COS Fully C compatible	45	2.25µs
SIN + COS of an angle	**58**	**2.90µs**

2.6 Memory utilization

Function	ROM (words)	Stack levels	Registers used	RAM (words)
SIN Assembly call	**32 + 100**	**3**	**1**	**in stack**
COS Assembly call	**34 + 100**	**3**	**1**	**in stack**
COS and SIN Assembly call	66 + 100	3	1	in stack
SIN Fully C compatible	42 + 100	5	1	in stack
COS Assembly call	44 + 100	5	1	in stack
COS + SIN in one function Assembly call	**55 + 125**	**3**	**1**	**in stack**

Note: in case of Sine and Cosine the same table is used.

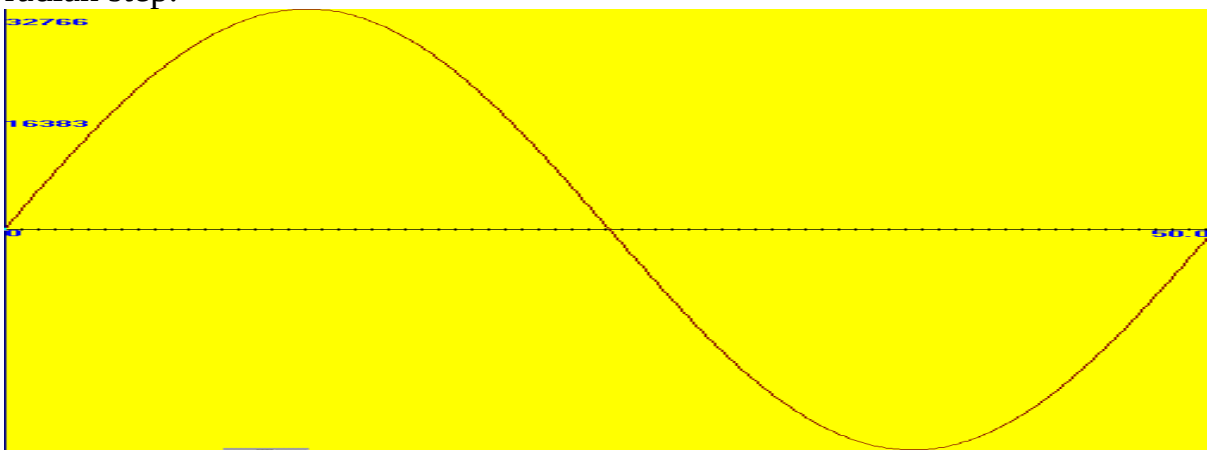
2.7 Sine generation precision

The following graphs present the precision of the table look-up and linear interpolation Sine functions. The precision measurement is made by comparing the table look-up and floating point C library results.

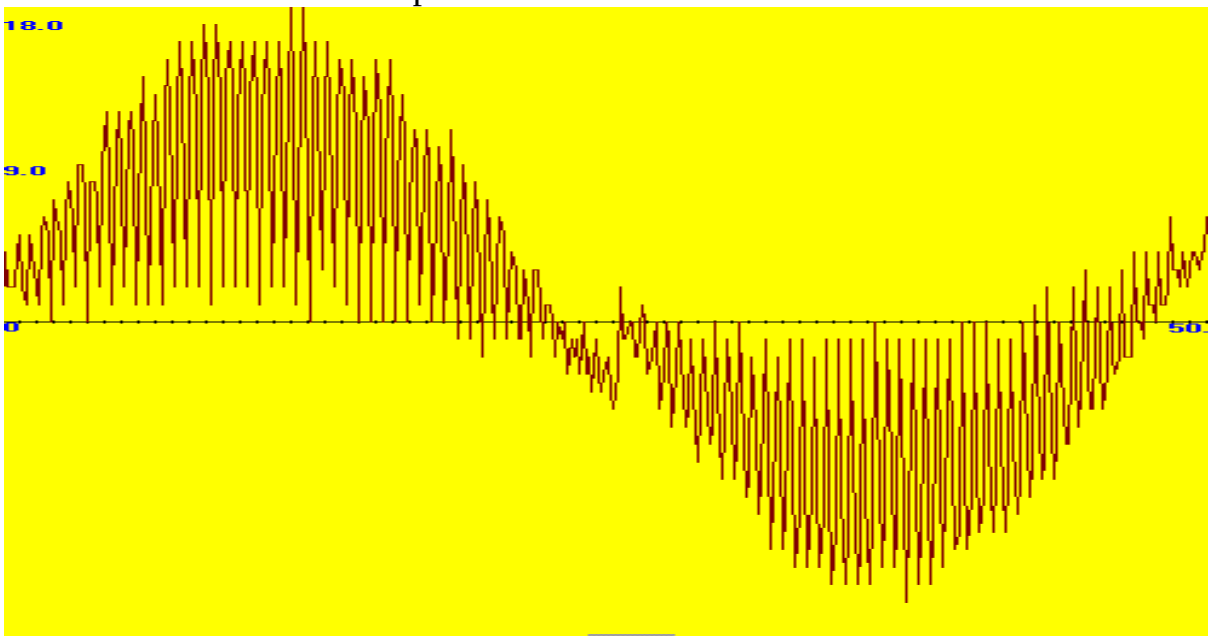
2.7.1 One Sine period analysis

The Sine period gives us a general image of error calculation. On the following graph, the results of Sine calculation are in Q15.

Table look-up with interpolation Sine calculation from 0 to 2*π with 0.015 radian step.



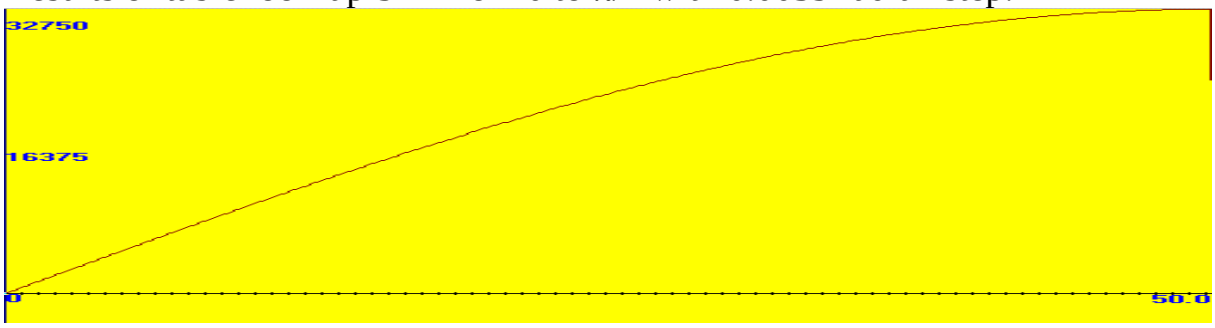
Error between the C floating point result and table look-up calculation from 0 to 2*π with a 0.015 radian step.



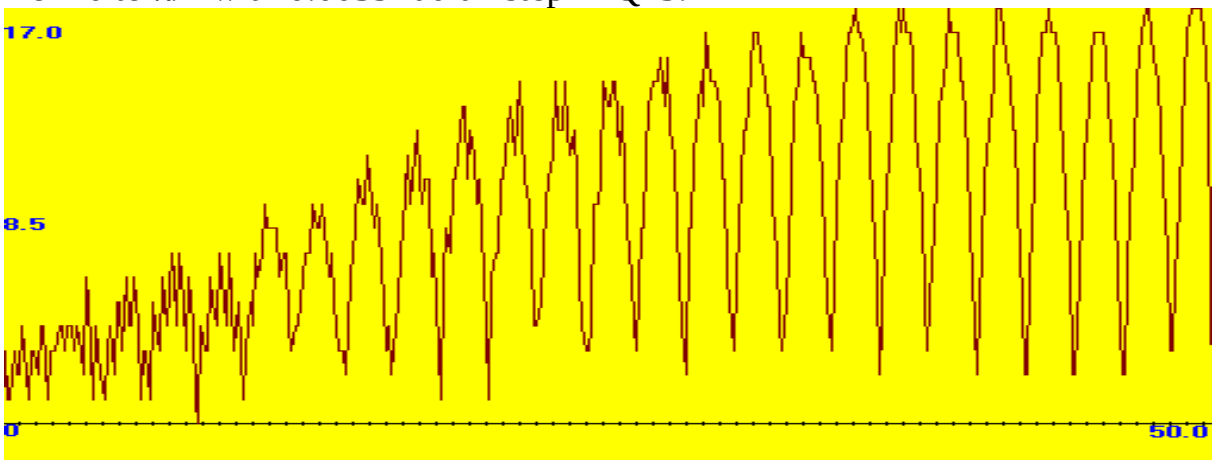
2.7.2 Quarter Sine period analysis

One more precise analysis is made on a quarter of a half Sine period. In this case, the step is 0.0039 radian. The interpolation is made at 0.125 radian intervals. In this way, the interpolation error is exposed on the following graphs. Each interval (table approximation) is displayed with 32 pixels.

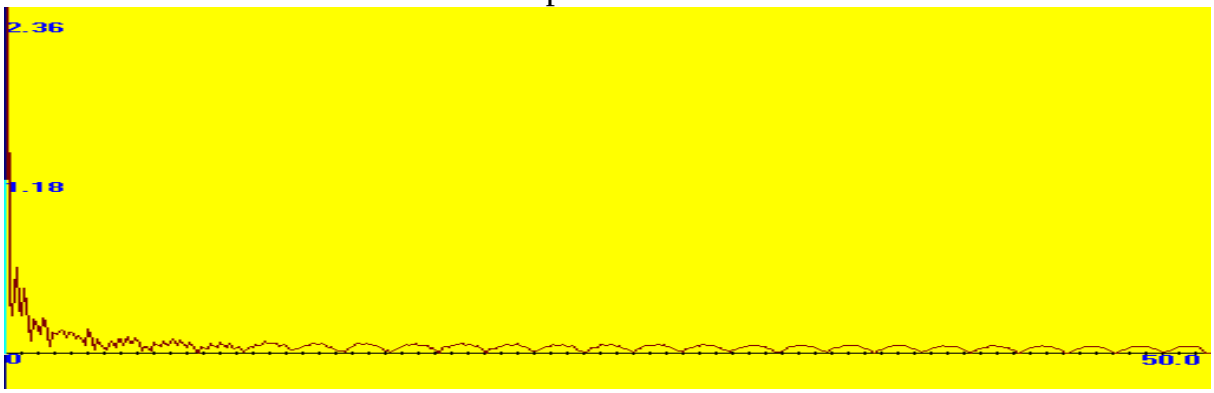
Results of table look-up SIN from 0 to π/2 with 0.0039 radian step.



Error between the C floating point result and table look-up SINE calculation from 0 to π/4 with 0.0039 radian step in Q15.



Error between the C floating point result and table look-up SINE calculation from 0 to π/4 with 0.0039 radian step in %.



3. Sine/Cosine with mathematical series

3.1 Presentation

This method realizes the Sine/Cosine functions with mathematical series for the Sine period and a mathematical series for Cosine. Calculations are made with fixed point instructions to optimize the time calculation. The dynamic used in the calculation fit with maximum precision without overflow.

3.2 Convention

Two different versions of each function are presented in this document: fully C-compatible functions and assembly calling functions. Different examples of assembly and C program calling Sine and Cosine are in the Annex.

3.2.1 Fully C-compatible functions

Fully C-compatible functions use C convention to use the stack for parameters passed to the functions. Parameters returned by functions are passed by pointer. Responsibilities of a called function by C are managed. Stack pointer AR1 is well positioned and the return address of the hardware stack is popped in case of a C interrupt routine events using C-function features (stack). The frame pointer is not modified. Register AR6/AR7 are not used.

3.2.2 Assembly-compatible functions

Assembly-compatible functions do not use their own variables; variables used are in the software stack.. Arguments are passed by the stack and the AR1 point to the stack just after the last argument. In return from function, results are in the stack.. Registers AR0, AR6, and AR7 are not modified.

3.3 Sine/Cosine mathematical series

The formula for calculating the value SIN(X) of an angle X is:
X in radian,

If (X>=0.0)
SIN(X)=(((-0.0372*X)-0.2338)*X+0.0544)*X+0.9826)*X+0.0013
else
SIN(X)=(((0.0372*X)+0.2338)*X+0.0544)*X-0.9826)*X+0.0013

The formula for calculating the value COS(X) of an angle X is:
X in radian,

If (X>=0.0)
COS(X)=((((-0.0076*X)+0.0595)*X-0.0211)*X-0.4879)*X-0.0028)*X+1.0
else
COS(X)=((((0.0076*X)+0.0595)*X+0.0211)*X-0.4879)*X+0.0028)*X+1.0

3.4 Functions

Functions SINE and COSINE are in the Annex with a main example. The convention to interface with these functions are:

- Input: ANGLE parameters is in the stack pointed by AR1
  The value of this angle is unsigned:

0 º	<->	0000h
90 º	<->	4000h
180 º	<->	8000h
240 º or -90º	<->	C000h

- Output:
  COS(angle) or SIN(angle) are in ACCUMULATOR in Q15 format
  (-1 to $1-2^{15}$)

3.5 Processor utilization (maximum)

Function	Cycles	Execution Time
SIN assembly main	**27**	**1.35µs**
COS assembly main	37	1.85µs
SIN Fully C-compatible	34	1.70µs
COS Fully C-compatible	44	2.20µs
SIN + COS of an angle assembly main	**64**	**3.20µs**

3.6 Memory utilization

Function	ROM (words)	Stack levels	Registers used	RAM (words)
SIN	**54**	**3**	**1**	**in stack**
COS	**67**	**3**	**1**	**in stack**
SIN	64	5	1	in stack
COS	77	5	1	in stack
COS and SIN	121	3	1	in stack

In this case, calculating the Sine and Cosine in the same function is not very interesting.

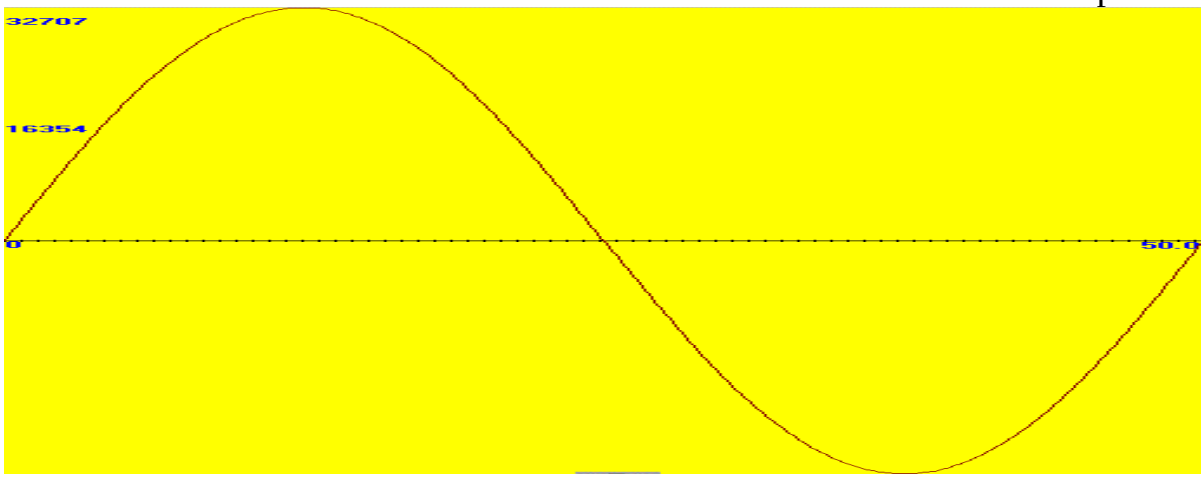
3.7 Sine generation precision

The following graphs present the precision of the mathematical series of Sine functions. The precision measurement is calculated by comparing the mathematical series and floating point C library results.

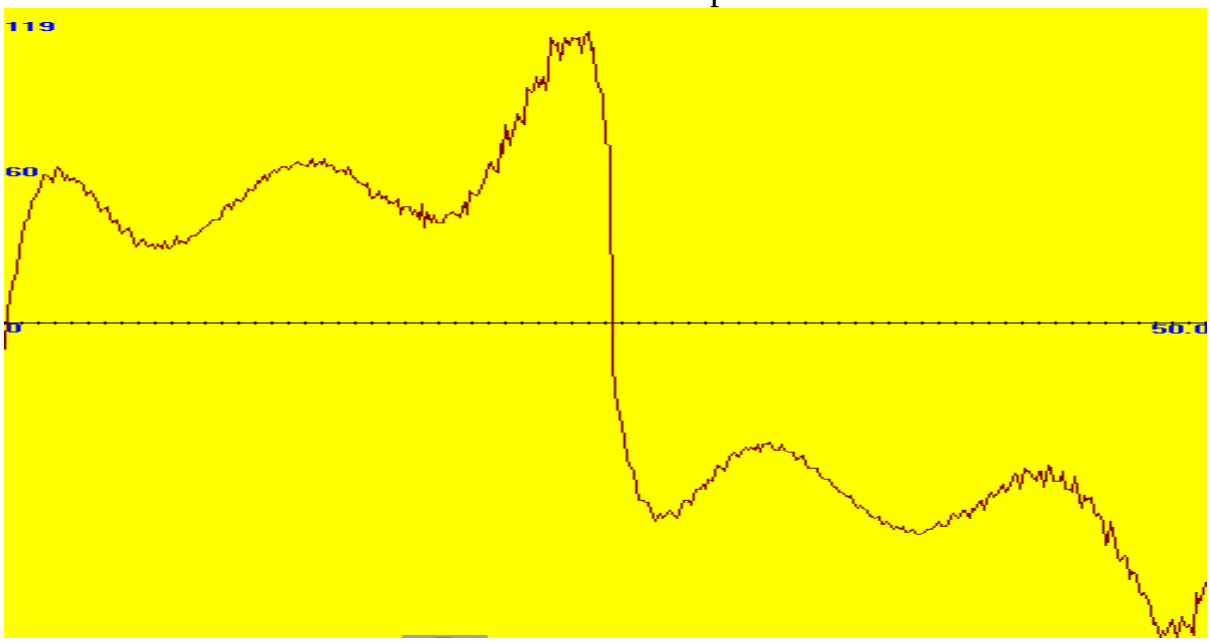
3.7.1 One Sine period analysis

The Sine period gives us a general image of error calculation. In the following graph, the results of Sine calculation are in Q15.

Mathematical series Sine calculation from 0 to 2*π with 0.015 radian step.



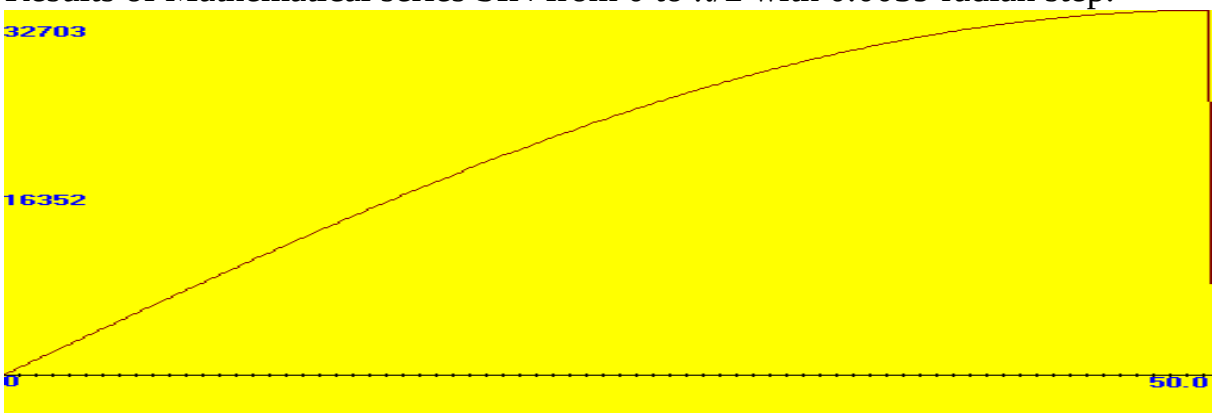
Error between the C floating point result and mathematical series SIN calculation from 0 to 2*π with 0.015 radian step.



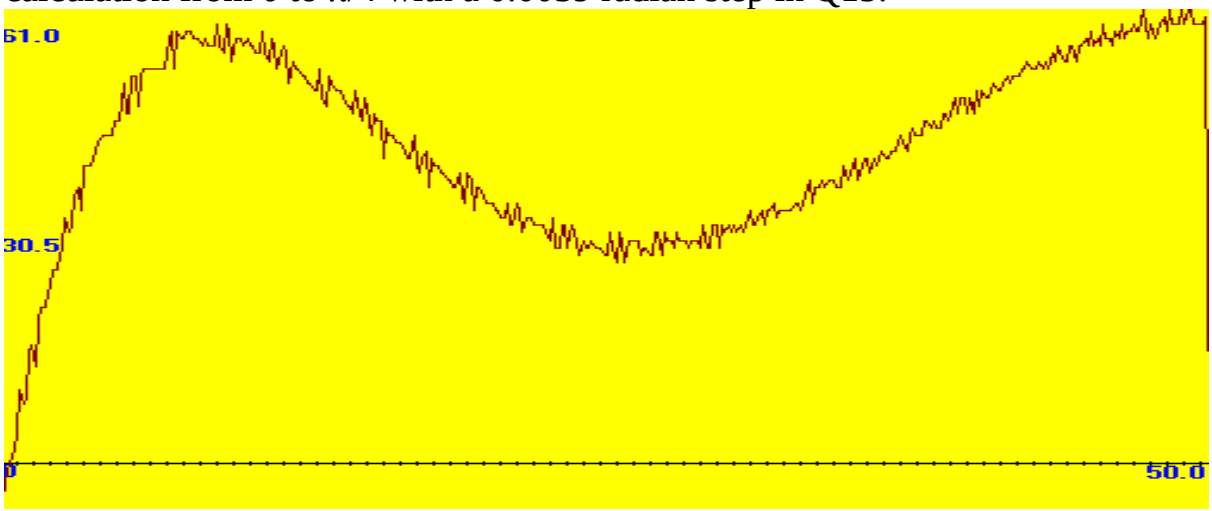
3.7.2 Quarter Sine period analysis

Another precise analysis is performed on a quarter of a half Sine period. In this case, the step with 0.0039 radian.

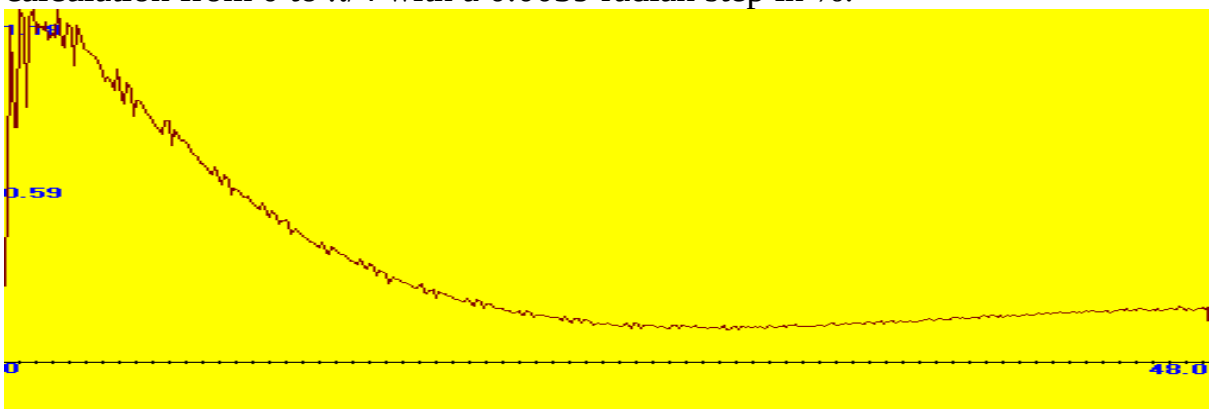
Results of Mathematical series SIN from 0 to π/2 with 0.0039 radian step.



Error between the C floating point result and mathematical series SIN calculation from 0 to π/4 with a 0.0039 radian step in Q15.



Error between the C floating point result and mathematical series SIN calculation from 0 to π/4 with a 0.0039 radian step in %.



4. Annex

4.1 Main C program to call Sine or Cosine functions

```
/************************************************************
*File Name:     Main_trig.c                                  *
*Project:                DMC Mathematical Library            *
*Originator:    Pascal DORSTER (Texas Intruments)           *
*Description:   Very simple main which call COS or SIN       *
*               function                                     *
*                                                           *
*Processor:     C2xx                                         *
*                                                           *
*Status:                                                     *
*                                                           *
*Last Update:   19 Oct 96                                    *
*___________________________________________________________ *
*Date of Mod    |          DESCRIPTION                       *
*-----------------------------------------------------------*
*               |                                           *
************************************************************/
  extern ser_cos();

  void main()
  {
  double rad;
  double a;
  int c;
  .
  .
  .
  if (rad!=0.0) /*-π < rad < +π*/
         c=(int)(32767*rad/3.1415927);
  else
         c=0;

  a=ser_cos(c);
  .
  .
  .
  }
```

4.2 Main assembly program to call Sine or Cosine functions

```
*************************************************************
*File Name:     Main_trig.asm                                 *
*Project:                  DMC Mathematical Library           *
*Originator:    Pascal DORSTER (Texas Intruments)             *
*                                                            *
*Description:   Very simple main which call COS or SIN        *
*               function                                     *
*Processor:     C2xx                                         *
*Status:                                                     *
*Last Update:   19 Oct 96                                    *
*____________________________________________________________*
*Date of Mod    |          DESCRIPTION                       *
*-------------------------------------------------------------*
*               |                                            *
*************************************************************
  .mmregs

  .def   _ser_sin

  .sect "vectors"
  b      _c_int0
  b      $

*************************************************************
* Main routine
*************************************************************
  .text

_c_int0:
  .
  .
  LAR    AR1,#60h          ;stack preparation
  MAR    *,AR1

  LAC    #4000h            ;π/2
  SACL   *+                ;parameters in stack
  CALL   _ser_sin
  .                        ;result in ACCU
  .
  .end
```

4.3 COS function with table look-up + linear interpolation for assembly program

```
*****************************************************************
*Routine Name:  COS                                             *
*Project:                DMC Mathematical Library               *
*Originator:    Pascal DORSTER (Texas Intruments)               *
*                                                               *
*Description:   COS function with Table Look-up + Interpolation *
*               Assembly calling funtion, variables             *
*               in s/w stack.                                   *
*                                                               *
*Status:                                                        *
*                                                               *
*Processor:     C2xx                                            *
*                                                               *
*Calling convention:                                            *
*               Input :                                         *
                Angle in stack value    0-360 degrees <=>0h-FFFFh*
*               Output : COS(angle)*32767, Q15, in ACCU         *
*               Pointed register AR1                            *
*                                                               *
*Stack commentary:                                              *
*               Position of current register in Caps            *
*               Stack at beginning:                             *
*                         angle/X                               *
*                                                               *
*                                                               *
*Last Update:   16 Oct 96                                       *
*_______________________________________________________________*
*Date of Mod    |          DESCRIPTION                          *
*---------------------------------------------------------------*
*               |                                              **

*****************************************************************

COS
  MAR    *-
  LAC    #4000h
  ADDS   *                ;add 0.5 for COS
  SACL   *
  LT     *+               ;variable in size_sin
                          ;*******stack : angle/X
  LAC    #64h
  SACL   *
                          ;*******stack : angle/TEMP
  MPYU   *                ;integer position in the table
  PAC                     ;in High part of Acc
  SACH   *+               ;remainder
                          ;*******stack : angle/indice/X
  SACL   *                ;
                          ;*******stack : angle/indice/REMAINDER
  LAC    *,15
  ANDK   #0ffffh,15
  SACH   *-
```

```
                              ;*******stack : angle/INDICE/remainder
  BCND   equal_sin,EQ
  LALK   Table_sin            ;address of beginning of the table
  ADD    *-                   ;address of the nearest first indice
                              ;*******stack : ANGLE/indice/remainder
  TBLR   *+
                              ;*******stack : y1/INDICE/remainder
  ADD    #1h
  TBLR   *                    ;Load Y2 with the ordinate of the
                              ;*******stack : y1/Y2/remainder
                              ;nearest second indice
  LAC    *-
                              ;*******stack : Y1/y2/remainder
  SUB    *+                   ;difference between the two Y value
                              ;*******stack : y1/Y2/remainder
  SACL   *+
                              ;*******stack : y1/temp/REMAINDER
  LT     *-
                              ;*******stack : y1/TEMP/remainder
  MPY    *
                              ;*******stack : y1/TEMP/remainder
  SPH    *-                   ;interpolation between Y1 and Y2
                              ;*******stack : Y1/temp/remainder
  LAC    *+
                              ;*******stack : y1/TEMP/remainder
  ADD    *,1                  ;Y(Xdata) in ACCU
  RET
equal_sin
  LALK   Table_sin            ;address of beginning of the table
  ADD    *-                   ;address of the pointed indice
                              ;*******stack : RESULT/
  TBLR   *
  LAC    *+
end_interp_sin
  RET

*************************************************************
* Table
*************************************************************

Table_sin  .include        sine.tab
Table_sin_end
  .word  0
  .end
```

4.4 COS function with table look-up + linear interpolation for C program

```
*****************************************************************
*Routine Name:  _COS                                            *
*Project:                DMC Mathematical Library               *
*Originator:    Pascal DORSTER (Texas Intruments)               *
*                                                               *
*Description:   COS function with Table Look-up + Interpolation *
*               C calling funtion, variables in C stack.        *
*               Fully C compatible                              *
*                                                               *
*Status:                                                        *
*                                                               *
*Processor:     C2xx                                            *
*                                                               *
*Calling convention:                                            *
*               Input :                                         *
*               Angle in stack value 0-360 degrees <=>0h-FFFFh*
*               Output : COS(angle)*32767, Q15, in ACCU         *
*               Pointed register AR1                            *
*                                                               *
*Stack commentary:                                              *
*               Position of current register in Caps            *
*               Stack at beginning:                             *
*                          angle/X                              *
*                                                               *
*Last Update:   16 Oct 96                                       *
*                                                               *
*_______________________________________________________________*
Date of Mod     |          DESCRIPTION                          *
*---------------------------------------------------------------*
*               |                                               *
*               |                                               *
*****************************************************************

_COS
  ADRK   2                ;C compatibility
  POPD   *+
  SAR    AR1,*
  LAR    AR2,*,AR2
  SBRK   3                ;C compatibility

  SPM    0
  LAC    #4000h
  ADDS   *                ;add 0.5 for COS
  SACL   *
  LT     *+               ;variable in size_sin
                          ;*******stack : angle/X
  LAC    #64h
  SACL   *
                          ;*******stack : angle/TEMP
  MPYU   *                ;integer position in the table
  PAC                     ;in High part of Acc
  SACH   *+               ;remainder
```

```
                              ;*******stack : angle/indice/X
  SACL   *                    ;
                              ;*******stack : angle/indice/REMAINDER
  LAC    *,15
  ANDK   #0ffffh,15
  SACH   *-
                              ;*******stack : angle/INDICE/remainder
  BCND   equal_sin,EQ
  LALK   Table_sin            ;address of beginning of the table
  ADD    *-                   ;address of the nearest first indice
                              ;*******stack : ANGLE/indice/remainder
  TBLR   *+
                              ;*******stack : y1/INDICE/remainder
  ADD    #1h
  TBLR   *                    ;Load Y2 with the ordinate of the
                              ;*******stack : y1/Y2/remainder
                              ;nearest second indice
  LAC    *-
                              ;*******stack : Y1/y2/remainder
  SUB    *+                   ;difference between the two Y value
                              ;*******stack : y1/Y2/remainder
  SACL   *+
                              ;*******stack : y1/temp/REMAINDER
  LT     *-
                              ;*******stack : y1/TEMP/remainder
  MPY    *
                              ;*******stack : y1/TEMP/remainder
  SPH    *-                   ;interpolation between Y1 and Y2
                              ;*******stack : Y1/temp/remainder
  LAC    *+
                              ;*******stack : y1/TEMP/remainder
  ADD    *,1,AR1                 ;Y(Xdata) in ACCU

  MAR    *-                   ;C compatibility
  PSHD   *-
  SBRK   1
  RET
equal_sin
  LALK   Table_sin            ;address of beginning of the table
  ADD    *-                   ;address of the pointed indice
                              ;*******stack : RESULT/
  TBLR   *
  LAC    *+,0,AR1

  MAR    *-                   ;C compatibility
  PSHD   *-
  SBRK   1

  RET

*************************************************************
* Table
*************************************************************

Table_sin  .include        sine.tab
Table_sin_end
  .word  0
```

```
.end
```

4.5 SIN function with table look-up + linear interpolation for assembly program

```
*****************************************************************
*Routine Name:  SIN                                             *
*Project:                DMC Mathematical Library               *
*Originator:    Pascal DORSTER (Texas Intruments)               *
*                                                               *
*Description:   SIN function with Table Look-up + Interpolation *
*               Assembly calling funtion, variables in s/w stack.*
*                                                               *
*Status:                                                        *
*                                                               *
*Processor:     C2xx                                            *
*                                                               *
*Calling convention:                                            *
*               Input :                                         *
*               Angle in stack value 0-360 degrees <=>0h-FFFFh  *
*               Output : SIN(angle)*32767, Q15, in ACCU         *
*               Pointed register AR1                            *
*                                                               *
*Stack commentary:                                              *
*               Position of current register in Caps            *
*               Stack at beginning:                             *
*                         angle/X                               *
*                                                               *
*                                                               *
*Last Update:   16 Oct 96                                       *
*_______________________________________________________________*
*Date of Mod    |          DESCRIPTION                          *
*---------------------------------------------------------------*
*               |                                               *
*               |                                               *
*****************************************************************

SIN
  MAR    *+
  LT     *+               ;variable in size_sin
                          ;*******stack : angle/X
  LAC    #64h
  SACL   *
                          ;*******stack : angle/TEMP
  MPYU   *                ;integer position in the table
  PAC                     ;in High part of Acc
  SACH   *+               ;remainder
                          ;*******stack : angle/indice/X
  SACL   *                ;
                          ;*******stack : angle/indice/REMAINDER
  LAC    *,15
  ANDK   #0ffffh,15
  SACH   *-
                          ;*******stack : angle/INDICE/remainder
  BCND   equal_sin,EQ
  LALK   Table_sin        ;address of beginning of the table
```

```
  ADD    *-                ;address of the nearest first indice
                           ;*******stack : ANGLE/indice/remainder
  TBLR   *+
                           ;*******stack : y1/INDICE/remainder
  ADD    #1h
  TBLR   *                 ;Load Y2 with the ordinate of the
                           ;*******stack : y1/Y2/remainder
                           ;nearest second indice
  LAC    *-
                           ;*******stack : Y1/y2/remainder
  SUB    *+                ;difference between the two Y value
                           ;*******stack : y1/Y2/remainder
  SACL   *+
                           ;*******stack : y1/temp/REMAINDER
  LT     *-
                           ;*******stack : y1/TEMP/remainder
  MPY    *
                           ;*******stack : y1/TEMP/remainder
  SPH    *-                ;interpolation between Y1 and Y2
                           ;*******stack : Y1/temp/remainder
  LAC    *+
                           ;*******stack : y1/TEMP/remainder
  ADD    *,1               ;Y(Xdata) in ACCU
  RET
equal_sin
  LALK   Table_sin         ;address of beginning of the table
  ADD    *-                ;address of the pointed indice
                           ;*******stack : RESULT/
  TBLR   *
  LAC    *+
                           ;*******stack : result/X

end_interp_sin
  RET

*************************************************************
* Table
*************************************************************

  Table_sin  .include      sine.tab
  Table_sin_end
         .word  0
```

4.6 SIN function with table look-up + linear interpolation for C program

```
*****************************************************************
*Routine Name:  _SIN                                            *
*Project:                DMC Mathematical Library               *
*Originator:    Pascal DORSTER (Texas Intruments)               *
*                                                               *
*Description:   SIN function with Table Look-up + Interpolation *
*               C calling funtion, variables in C stack.        *
*               Fully C compatible                              *
*                                                               *
*Status:                                                        *
*                                                               *
*Processor:     C2xx                                            *
*                                                               *
*Calling convention:                                            *
*               Input :                                         *
*               Angle in stack value 0-360 degrees <=>0h-FFFFh  *
*               Output : SIN(angle)*32767, Q15, in ACCU         *
*               Pointed register AR1                            *
*                                                               *
*Stack commentary:                                              *
*               Position of current register in Caps            *
*               Stack at beginning:                             *
*                         angle/X                               *
*                                                               *
*                                                               *
*Last Update:   16 Oct 96                                       *
*                                                               *
*_______________________________________________________________*
*Date of Mod    |          DESCRIPTION                          *
*---------------------------------------------------------------*
*               |                                               *
*               |                                               *
*****************************************************************

_SIN
  ADRK   2                ;C compatibility
  POPD   *+
  SAR    AR1,*
  LAR    AR2,*,AR2
  SBRK   3                ;C compatibility

  SPM    0
  LT     *+               ;variable in size_sin
                          ;*******stack : angle/X
  LAC    #64h
  SACL   *
                          ;*******stack : angle/TEMP
  MPYU   *                ;integer position in the table
  PAC                     ;in High part of Acc
  SACH   *+               ;remainder
                          ;*******stack : angle/indice/X
  SACL   *                ;
```

```
                              ;*******stack : angle/indice/REMAINDER
  LAC    *,15
  ANDK   #0ffffh,15
  SACH   *-
                              ;*******stack : angle/INDICE/remainder
  BCND   equal_sin,EQ
  LALK   Table_sin            ;address of beginning of the table
  ADD    *-                   ;address of the nearest first indice
                              ;*******stack : ANGLE/indice/remainder
  TBLR   *+
                              ;*******stack : y1/INDICE/remainder
  ADD    #1h
  TBLR   *                    ;Load Y2 with the ordinate of the
                              ;*******stack : y1/Y2/remainder
                              ;nearest second indice
  LAC    *-
                              ;*******stack : Y1/y2/remainder
  SUB    *+                   ;difference between the two Y value
                              ;*******stack : y1/Y2/remainder
  SACL   *+
                              ;*******stack : y1/temp/REMAINDER
  LT     *-
                              ;*******stack : y1/TEMP/remainder
  MPY    *
                              ;*******stack : y1/TEMP/remainder
  SPH    *-                   ;interpolation between Y1 and Y2
                              ;*******stack : Y1/temp/remainder
  LAC    *+
                              ;*******stack : y1/TEMP/remainder
  ADD    *,1,AR1              ;Y(Xdata) in ACCU
  MAR    *-                   ;C compatibility
  PSHD   *-
  SBRK   1
  RET
equal_sin
  LALK   Table_sin            ;address of beginning of the table
  ADD    *-                   ;address of the pointed indice
                              ;*******stack : RESULT/...
  TBLR   *
  LAC    *+,0,AR1
                              ;*******stack : result/X
  MAR    *-                   ;C compatibility
  PSHD   *-
  SBRK   1
  RET
*****************************************************************
* Table
*****************************************************************
Table_sin  .include   sine.tab
Table_sin_end
  .word  0
```

4.7 Table for COS or SIN function with table look-up

```
*********************************************************************
* TABLE Sine and Cosine for separate functions COS and SIN          *
*********************************************************************
  .word   0
  .word   2057
  .word   4107
  .word   6140
  .word   8149
  .word   10126
  .word   12062
  .word   13952
  .word   15786
  .word   17557
  .word   19260
  .word   20886
  .word   22431
  .word   23886
  .word   25247
  .word   26509
  .word   27666
  .word   28714
  .word   29648
  .word   30466
  .word   31163
  .word   31738
  .word   32187
  .word   32509
  .word   32702
  .word   32767
  .word   32702
  .word   32509
  .word   32187
  .word   31738
  .word   31163
  .word   30466
  .word   29648
  .word   28714
  .word   27666
  .word   26509
  .word   25247
  .word   23886
  .word   22431
  .word   20886
  .word   19260
  .word   17557
  .word   15786
  .word   13952
  .word   12062
  .word   10126
  .word   8149
  .word   6140
  .word   4107
  .word   2057
  .word   0
  .word   -2057
  .word   -4107
  .word   -6140
  .word   -8149
```

```
  .word   -10126
  .word   -12062
  .word   -13952
  .word   -15786
  .word   -17557
  .word   -19260
  .word   -20886
  .word   -22431
  .word   -23886
  .word   -25247
  .word   -26509
  .word   -27666
  .word   -28714
  .word   -29648
  .word   -30466
  .word   -31163
  .word   -31738
  .word   -32187
  .word   -32509
  .word   -32702
  .word   -32767
  .word   -32702
  .word   -32509
  .word   -32187
  .word   -31738
  .word   -31163
  .word   -30466
  .word   -29648
  .word   -28714
  .word   -27666
  .word   -26509
  .word   -25247
  .word   -23886
  .word   -22431
  .word   -20886
  .word   -19260
  .word   -17557
  .word   -15786
  .word   -13952
  .word   -12062
  .word   -10126
  .word   -8149
  .word   -6140
  .word   -4107
  .word   -2057
```

4.8 COS_SIN function with table look-up + linear interpolation for assembly program

```
*****************************************************************
*Routine Name:  COS_SIN                                         *
*Project:       DMC        Mathematical Library                 *
*Originator:    Pascal DORSTER (Texas Intruments)               *
*                                                               *
*Description:   COS+SIN function with                           *
*               Table Look-up + Interpolation                   *
*               Assembly calling funtion, variables in s/w stack.*
*                                                               *
*Status:                                                        *
*                                                               *
*Processor:     C2xx                                            *
*                                                               *
*Calling convention:                                            *
*               Input :                                         *
*               Angle in stack value 0-360 degrees <=>0h-FFFFh   *
*               Output:                                         *
*               SIN(angle)*32767, Q15, in stack pointed by AR1-1 *
*               COS(angle)*32767,Q15, in stack pointed by AR1*
*               Pointed register AR1                            *
*                                                               *
*Stack commentary:                                              *
*               Position of current register in Caps            *
*               Stack at beginning:                             *
*                         angle/X                               *
*                                                               *
*Last Update:   16 Oct 96                                       *
*_______________________________________________________________*
*Date of Mod    |          DESCRIPTION                          *
*---------------------------------------------------------------*
*               |                                               *
*               |                                               *
*****************************************************************

size_tab .set   64h        ;this is not an error

COS_SIN
  MAR    *+
  LT     *+                ;variable in size_sin
                           ;*stack: angle/X
  LAC    #size_tab
  SACL   *
                           ;*stack: angle/TEMP
  MPYU   *                 ;integer position in the table
  PAC                      ;in High part of Acc
  SACH   *+                ;remainder
                           ;*stack: angle/indice/X
  SACL   *                 ;
                           ;*stack: angle/indice/REMAINDER
  LAC    *,15
  ANDK   #0ffffh,15
  SACH   *-
```

```
                                ;*stack: angle/INDICE/remainder
BCND   equal_sin,EQ
LALK   Table_sin_cos            ;address of beginning of the table
ADD    *-                       ;address of the nearest first indice
                                ;*stack: ANGLE/indice/remainder
TBLR   *+
                                ;*stack: y1/INDICE/remainder
ADD    #1h
TBLR   *+                       ;Load Y2 with the ordinate of the
                                ;nearest second indice
                                ;*stack: y1/y2/REMAINDER
ADRK   1
                                ;*stack: y1/y2/remainder/X
ADD    #18h
TBLR   *+
                                ;*stack: y1/y2/remainder/yy1/X
ADD    #1h
TBLR   *                        ;*stack: y1/y2/remainder/yy1/YY2

LAC    *-                       ;calculate cos interpolate
                                ;*stack: y1/y2/remainder/YY1/yy2
SUB    *+                       ;difference between the two Y value
                                ;*stack: y1/y2/remainder/yy1/YY2
SACL   *                        ;*stack: y1/y2/remainder/yy1/TEMP
LT     *-
                                ;*stack: y1/y2/remainder/YY1
SBRK   1
                                ;*stack: y1/y2/REMAINDER/yy1
MPY    *
                                ;*stack: y1/y2/REMAINDER/yy1
ADRK   2
                                ;*stack: y1/y2/remainder/yy1/TEMP
SPH    *-                       ;interpolation between Y1 and Y2
                                ;*stack: Y1/y2/remainder/YY1/temp
LAC    *+
                                ;*stack: y1/y2/remainder/yy1/TEMP
ADD    *-,1                     ;Y(Xdata) in ACCU
                                ;*stack: y1/y2/remainder/YY1
SACL   *-
                                ;*stack: y1/y2/REMAINDER/cos
                                ;calculate sin interpolate
SBRK   1
                                ;*stack: y1/Y2/remainder/cos
LAC    *-
                                ;*stack: Y1/y2/remainder
SUB    *+                       ;difference between the two Y value
                                ;*stack: y1/Y2/remainder/cos
SACL   *+
                                ;*stack: y1/temp/REMAINDER/cos
LT     *-
                                ;*stack: y1/TEMP/remainder/cos
MPY    *
                                ;*stack: y1/TEMP/remainder/cos
SPH    *-                       ;interpolation between Y1 and Y2
                                ;*stack: Y1/temp/remainder/cos
LAC    *+
```

```
                              ;*stack: y1/TEMP/remainder/cos
  ADD    *-,1                 ;Y(Xdata) in ACCU
                              ;*stack: y1/temp/remainder/cos
  SACL   *                    ;*stack: SIN/temp/remainder/cos
  ADRK   3
                              ;*stack: sin/temp/remainder/COS
  LAC    *
  SBRK   2
                              ;*stack: sin/TEMP/remainder/cos
  SACL   *
                              ;*stack: sin/COS
  RET
equal_sin
  LALK   Table_sin_cos        ;address of beginning of the table
                              ;*stack: angle/INDICE/remainder
  ADD    *-                   ;address of the pointed indice
                              ;*stack: ANGLE/indice
  TBLR   *+
                              ;*stack: sin/INDICE/remainder
  ADDK   19h
  TBLR   *
                              ;*stack: sin/COS

end_interp_sin
  RET

*****************************************************************
* Table
*****************************************************************

Table_sin_cos  .include   sin_cos.tab
Table_sin_cos_end
  .word  0
```

4.9 Table for COS_SIN function with table look-up

```
*************************************************************
* TABLE Sine + Cosine for functions COS_SIN                  *
*************************************************************
  .word  0
  .word  2057
  .word  4107
  .word  6140
  .word  8149
  .word  10126
  .word  12062
  .word  13952
  .word  15786
  .word  17557
  .word  19260
  .word  20886
  .word  22431
  .word  23886
  .word  25247
  .word  26509
  .word  27666
  .word  28714
  .word  29648
  .word  30466
  .word  31163
  .word  31738
  .word  32187
  .word  32509
  .word  32702
  .word  32767
  .word  32702
  .word  32509
  .word  32187
  .word  31738
  .word  31163
  .word  30466
  .word  29648
  .word  28714
  .word  27666
  .word  26509
  .word  25247
  .word  23886
  .word  22431
  .word  20886
  .word  19260
  .word  17557
  .word  15786
  .word  13952
  .word  12062
  .word  10126
  .word  8149
  .word  6140
  .word  4107
  .word  2057
```

```
  .word  0
  .word  -2057
  .word  -4107
  .word  -6140
  .word  -8149
  .word  -10126
  .word  -12062
  .word  -13952
  .word  -15786
  .word  -17557
  .word  -19260
  .word  -20886
  .word  -22431
  .word  -23886
  .word  -25247
  .word  -26509
  .word  -27666
  .word  -28714
  .word  -29648
  .word  -30466
  .word  -31163
  .word  -31738
  .word  -32187
  .word  -32509
  .word  -32702
  .word  -32767
  .word  -32702
  .word  -32509
  .word  -32187
  .word  -31738
  .word  -31163
  .word  -30466
  .word  -29648
  .word  -28714
  .word  -27666
  .word  -26509
  .word  -25247
  .word  -23886
  .word  -22431
  .word  -20886
  .word  -19260
  .word  -17557
  .word  -15786
  .word  -13952
  .word  -12062
  .word  -10126
  .word  -8149
  .word  -6140
  .word  -4107
  .word  -2057
  .word  0
  .word  2057
  .word  4107
  .word  6140
  .word  8149
  .word  10126
```

```
  .word  12062
  .word  13952
  .word  15786
  .word  17557
  .word  19260
  .word  20886
  .word  22431
  .word  23886
  .word  25247
  .word  26509
  .word  27666
  .word  28714
  .word  29648
  .word  30466
  .word  31163
  .word  31738
  .word  32187
  .word  32509
  .word  32702
```

4.10 SIN with mathematical series for assembly program

```
*************************************************************
*Routine Name:  sin_ser                                     *
*Project:                  DMC Mathematical Library         *
*Originator:    Pascal DORSTER (Texas Intruments)           *
*                                                           *
*Description:   SIN function with Mathematical Serie        *
*               Assembly calling funtion, variables in s/w stack. *
*                                                           *
*Status:                                                    *
*                                                           *
*Processor:     C2xx                                        *
*                                                           *
*                                                           *
*Calling convention:                                        *
*               Input :                                     *
*               Angle in stack value 0-360 degrees <=>0h-FFFFh   *
*               Output : SIN(angle)*32767, Q15, in ACCU     *
*               Pointed register AR1                        *
*                                                           *
*Stack commentary:                                          *
*               Position of current register in Caps        *
*               Stack at beginning:                         *
*                         angle/X                           *
*                                                           *
*                                                           *
*Last Update:   18 Oct 96                                   *
*___________________________________________________________*
*Date of Mod    |          DESCRIPTION                      *
*------------------------------------------------------------ *
*               |                                           *
*               |                                           *
*************************************************************

sin_ser
  SETC   SXM
  MAR    *-
  LAC    *
  BCND   angle_neg,LT

angle_pos
  LT     *+               ;variable in size_sin
                          ;*stack : angle/X
  MPYK   0c90h            ;abscissa scale modify degree -> rad
  PAC
  SACH   *,4              ;angle in rad
                          ;*stack : angle/RAD
  LT     *+
                          ;*stack : angle/rad/X
  MPYK   985h             ;0.0372
                          ;
                          ;PREG : angle * 0.0372
  LAC    #0C426h,13       ; in high ACCU
  APAC                    ;ACCU : -0.2338 + angle * 0.0372
```

```
  SACH   * ,3              ;temporary value
                           ;angle in TREG
  MPY    *                 ;multiply by temporary value
                           ;temporary value
                           ;*stack : angle/rad/TEMP
  LAC    #6f69h,10         ;0.0544
  APAC                     ;ACCU: temp + 0.0544
  SACH   *,2               ;temp
                           ;*stack : angle/RAD/temp
                           ;angle in TREG
                           ;*stack : angle/rad/TEMP
  MPY    *                 ;temp
  LAC    #7dc5h,13         ;0.9826
  APAC                     ;ACCU=0.9826+temp

  SACH   *,3               ;treg = rad
                           ;temp
  MPY    *                 ;rad * temp
                           ;temp * angle
  LAC    #05532,4          ;0.0013
  APAC
  SACH   * ,3              ;result
  LAC    *-
                           ;*stack : angle/SIN

  RET

angle_neg
  LT     *+                ;variable in size_sin
                           ;*stack : angle/X
  MPYK   0c90h             ;abscisse scale modif degre -> rad
  PAC
  SACH   *,4               ;angle in rad
                           ;*stack : angle/RAD
  LT     *+
                           ;*stack : angle/rad/X
  MPYK   985h                 ;0.0372
                           ;
                           ;PREG : angle * 0.0372
  LAC    #03bdah,13        ; in high ACCU
  APAC                     ;ACCU : -0.2338 + angle * 0.0372
  SACH   * ,3              ;temporary value
                           ;angle in TREG
  MPY    *                 ;multiply by temporary value
                           ;temporary value
                           ;*stack : angle/rad/TEMP
  LAC    #6f69h,10         ;0.0544
  APAC                     ;ACCU: temp + 0.0544
  SACH   *,2               ;temp
                           ;*stack : angle/RAD/temp
                           ;angle in TREG
                           ;*stack : angle/rad/TEMP
  MPY    *                 ;temp
  LAC    #7dc5h,13         ;0.9826
  SPAC                     ;ACCU=0.9826+temp
```

```
SACH    *,3             ;treg = rad
MPY     *               ;rad * temp
                        ;temp * angle
LAC     #05532,4        ;0.0013
APAC

SACH    *,3             ;result
LAC     *-
                        ;*stack : angle/SIN
RET
```

4.11 SIN with mathematical series for C program

```
*************************************************************
*Routine Name:  _sin_ser                                    *
*Project:                DMC Mathematical Library           *
*Originator:    Pascal DORSTER (Texas Intruments)           *
*                                                           *
*Description:   SIN function with Mathematical Serie        *
*               C calling funtion, variables in C stack.    *
*               Fully C compatible                          *
*                                                           *
*Status:                                                    *
*                                                           *
*Processor:     C2xx                                        *
*                                                           *
*Calling convention:                                        *
*               Input :                                     *
*               Angle in stack value 0-360 degrees <=>0h-FFFFh  *
*               Output : SIN(angle)*32767, Q15, in ACCU     *
*               Pointed register AR1                        *
*                                                           *
*Stack commentary:                                          *
*               Position of current register in Caps        *
*               Stack at beginning:                         *
*                          angle/X                          *
*                                                           *
*                                                           *
*Last Update:   18 Oct 96                                   *
*                                                           *
*___________________________________________________________*
*Date of Mod    |          DESCRIPTION                      *
*------------------------------------------------------------ *
*               |                                           *
*               |                                           *
*************************************************************

_sin_ser
  ADRK   2                 ;C compatibility
  POPD   *+
  SAR    AR1,*
  LAR    AR2,*,AR2
  SBRK   3                 ;C compatibility

  SPM    0
  SETC   SXM
  LAC    *
  BCND   angle_neg,LT

angle_pos
  LT     *+                ;variable in size_sin
                           ;*stack : angle/X
  MPYK   0c90h             ;abscissa scale modify degree -> rad
                           ;in
  PAC
  SACH   *,4               ;angle in rad
```

```
                        ;*stack : angle/RAD
LT      *+
                        ;*stack : angle/rad/X
MPYK    985h            ;0.0372
                        ;
                        ;PREG : angle * 0.0372
LAC     #0C426h,13      ; in high ACCU
APAC                    ;ACCU : -0.2338 + angle * 0.0372
SACH    * ,3            ;temporary value
                        ;angle in TREG
MPY     *               ;multiply by temporary value
                        ;temporary value
                        ;*stack : angle/rad/TEMP
LAC     #6f69h,10       ;0.0544
APAC                    ;ACCU: temp + 0.0544
SACH    *,2             ;temp
                        ;*stack : angle/RAD/temp
                        ;angle in TREG
                        ;*stack : angle/rad/TEMP
MPY     *               ;temp
LAC     #7dc5h,13       ;0.9826
APAC                    ;ACCU=0.9826+temp

SACH    *,3             ;treg = rad
                        ;temp
MPY     *               ;rad * temp
                        ;temp * angle
LAC     #05532,4        ;0.0013
APAC
SACH    * ,3            ;result
LAC     *-,0,AR1
                        ;*stack : angle/SIN
MAR     *-              ;C compatibility
PSHD    *-
SBRK    1

RET

angle_neg
LT      *+              ;variable in size_sin
                        ;*stack : angle/X
MPYK    0c90h           ;abscisse scale modif degre -> rad
PAC
SACH    *,4             ;angle in rad
                        ;*stack : angle/RAD
LT      *+
                        ;*stack : angle/rad/X
MPYK    985h            ;0.0372
                        ;
                        ;PREG : angle * 0.0372
LAC     #03bdah,13      ; in high ACCU
APAC                    ;ACCU : -0.2338 + angle * 0.0372
SACH    * ,3            ;temporary value
                        ;angle in TREG
MPY     *               ;multiply by temporary value
                        ;temporary value
```

```
                                ;*stack : angle/rad/TEMP
LAC     #6f69h,10               ;0.0544
APAC                            ;ACCU: temp + 0.0544
SACH    *,2                     ;temp
                                ;*stack : angle/RAD/temp
                                ;angle in TREG
                                ;*stack : angle/rad/TEMP
MPY     *                       ;temp
LAC     #7dc5h,13               ;0.9826
SPAC                            ;ACCU=0.9826+temp

SACH    *,3                     treg = rad
MPY     *                       ;rad * temp
                                ;temp * angle
LAC     #05532,4                ;0.0013
APAC

SACH    *,3                     ;result
LAC     *-,0,AR1
                                ;*stack : angle/SIN
MAR     *-                      ;C compatibility
PSHD    *-
SBRK    1

RET
```

4.12 COS with mathematical series for assembly program

```
*************************************************************
*Routine Name:  cos_ser                                     *
*Project:                  DMC Mathematical Library         *
*Originator:    Pascal DORSTER (Texas Intruments)           *
*                                                           *
*Description:   COS function with Mathematical Serie        *
*               Assembly calling funtion, variables in s/w stack.*
*                                                           *
*Status:                                                    *
*                                                           *
*Processor:     C2xx                                        *
*                                                           *
*                                                           *
*Calling convention:                                        *
*               Input :                                     *
*               Angle in stack value 0-360 degrees <=>0h-FFFFh  *
*               Output : COS(angle)*32767, Q15, in ACCU     *
*               Pointed register AR1                        *
*                                                           *
*Stack commentary:                                          *
*               Position of current register in Caps        *
*               Stack at beginning:                         *
*                         angle/X                           *
*                                                           *
*                                                           *
*Last Update:   18 Oct 96                                   *
*___________________________________________________________*
*Date of Mod    |          DESCRIPTION                      *
*-------------------------------------------------------- *
*               |                                           *
*               |                                           *
*************************************************************

cos_ser
  SETC   SXM
  MAR    *-
  LAC    *
  BCND   angle_neg,LT

angle_pos
  LT     *+               ;variable in size_sin
                          ;*stack : angle/X
  MPYK   0c90h            ;abscisse scale modif degre -> rad
  PAC
  SACH   *,4              ;angle in rad
                          ;*stack : angle/RAD
  LT     *+
                          ;*stack : angle/rad/X
  MPYK   1070h            ;0.0076 neg
                          ;
                          ;PREG : angle * 0.0076
  LAC    #01e76h,15       ; in high ACCU
  APAC                    ;ACCU : +0.0595 + angle * 0.0076
```

```
  SACH   * ,3              ;temporary value
                           ;angle in TREG
  MPY    *                 ;multiply by temporary value
                           ;temporary value
                           ;*stack : angle/rad/TEMP
  LAC    #0f532h,15        ;0.0211
  APAC                     ;ACCU: temp + 0.0211
  SACH   *                 ;temp
                           ;*stack : angle/RAD/temp
                           ;angle in TREG
                           ;*stack : angle/rad/TEMP
  MPY    *                 ;temp
  LAC    #0e0c6h,15        ;0.4879
  APAC                     ;ACCU=0.9826+temp

  SACH   *,2               ;treg = rad
  MPY    *                 ;rad * temp
                           ;temp * angle
  LAC    #0a440h,5          ;0.0028
  APAC
  SACH   *,2               ;result
  MPY    *                 ;
  LAC    #1000h,15         ;1
  APAC
  SETC   OVM
  SACH   *,3
  LAC    *,16
  ADD    *,16
  SACH   *
  LAC    *-
                           ;*stack : angle/SIN
  RET

angle_neg
  LT     *+                ;variable in size_sin
                           ;*stack : angle/X
  MPYK   0c90h             ;abscisse scale modif degre -> rad
  PAC
  SACH   *,4               ;angle in rad
                           ;*stack : angle/RAD
  LT     *+
                           ;*stack : angle/rad/X
  MPYK   0f90h             ;0.0076 neg
                           ;
                           ;PREG : angle * 0.0076
  LAC    #01e76h,15        ; in high ACCU
  APAC                     ;ACCU : +0.0595 + angle * 0.0076
  SACH   * ,3              ;temporary value
                           ;angle in TREG
  MPY    *                 ;multiply by temporary value
                           ;temporary value
                           ;*stack : angle/rad/TEMP
  LAC    #00aceh,15        ;0.0211
  APAC                     ;ACCU: temp + 0.0211
  SACH   *                 ;temp
                           ;*stack : angle/RAD/temp
```

```
                            ;angle in TREG
                            ;*stack : angle/rad/TEMP
MPY     *                   ;temp
LAC     #0e0c6h,15          ;0.4879
APAC                        ;ACCU=0.9826+temp

SACH    *,2                 ;treg = rad
                            ;temp
MPY     *                   ;rad * temp
                            ;temp * angle
LAC     #05bc0h,5           ;0.0028
APAC
SACH    *,2                 ;result
MPY     *                   ;
LAC     #1000h,15           ;1
APAC
SETC    OVM
SACH    *,3
LAC     *,16
ADD     *,16
SACH    *
LAC     *-
                            ;*stack : angle/SIN
RET
```

4.13 COS with mathematical series for C program

```
*************************************************************
*Routine Name:  _cos_ser                                    *
*Project:                DMC Mathematical Library           *
*Originator:    Pascal DORSTER (Texas Intruments)           *
*                                                           *
*Description:   COS function with Mathematical Serie        *
*               C calling funtion, variables in C stack.    *
*               Fully C compatible                          *
*                                                           *
*Status:                                                    *
*                                                           *
*Processor:     C2xx                                        *
*                                                           *
*Calling convention:                                        *
*               Input :                                     *
*               Angle in stack value 0-360 degrees <=>0h-FFFFh  *
*               Output : COS(angle)*32767, Q15, in ACCU     *
*               Pointed register AR1                        *
*                                                           *
*Stack commentary:                                          *
*               Position of current register in Caps        *
*               Stack at beginning:                         *
*                         angle/X                           *
*                                                           *
*Last Update:   18 Oct 96                                   *
*                                                           *
*___________________________________________________________*
*Date of Mod    |          DESCRIPTION                      *
*--------------------------------------------------------*
*               |                                           *
*               |                                           *
*************************************************************

_cos_ser
  ADRK   2                ;C compatibility
  POPD   *+
  SAR    AR1,*
  LAR    AR2,*,AR2
  SBRK   3                ;C compatibility

  SPM    0

  SETC   SXM
  LAC    *
  BCND   angle_neg,LT

  angle_pos
  LT     *+               ;variable in size_sin
                          ;*stack : angle/X
  MPYK   0c90h            ;abscisse scale modif degre -> rad
  PAC
  SACH   *,4              ;angle in rad
                          ;*stack : angle/RAD
```

```
  LT     *+
                              ;*stack : angle/rad/X
  MPYK   1070h                ;0.0076 neg
                              ;
                              ;PREG : angle * 0.0076
  LAC    #01e76h,15           ; in high ACCU
  APAC                        ;ACCU : +0.0595 + angle * 0.0076
  SACH   * ,3                 ;temporary value
                              ;angle in TREG
  MPY    *                    ;multiply by temporary value
                              ;temporary value
                              ;*stack : angle/rad/TEMP
  LAC    #0f532h,15           ;0.0211
  APAC                        ;ACCU: temp + 0.0211
  SACH   *                    ;temp
                              ;*stack : angle/RAD/temp
                              ;angle in TREG
                              ;*stack : angle/rad/TEMP
  MPY    *                    ;temp
  LAC    #0e0c6h,15           ;0.4879
  APAC                        ;ACCU=0.9826+temp

  SACH   *,2                  ;treg = rad
                              ;temp
  MPY    *                    ;rad * temp
                              ;temp * angle
  LAC    #0a440h,5             ;0.0028
  APAC
  SACH   *,2                  ;result
  MPY    *                    ;
  LAC    #1000h,15            ;1
  APAC
  SETC   OVM
  SACH   *,3
  LAC    *,16
  ADD    *,16
  SACH   *
  LAC    *-,0,AR1
                              ;*stack : angle/SIN
  MAR    *-                   ;C compatibility
  PSHD   *-
  SBRK   1

  RET

angle_neg
  LT     *+                   ;variable in size_sin
                              ;*stack : angle/X
  MPYK   0c90h                ;abscisse scale modif degre -> rad
  PAC
  SACH   *,4                  ;angle in rad
                              ;*stack : angle/RAD
  LT     *+
                              ;*stack : angle/rad/X
  MPYK   0f90h                ;0.0076 neg
                              ;
```

```
                                ;PREG : angle * 0.0076
LAC     #01e76h,15              ; in high ACCU
APAC                            ;ACCU : +0.0595 + angle * 0.0076
SACH    * ,3                    ;temporary value
                                ;angle in TREG
MPY     *                       ;multiply by temporary value
                                ;temporary value
                                ;*stack : angle/rad/TEMP
LAC     #00aceh,15              ;0.0211
APAC                            ;ACCU: temp + 0.0211
SACH    *                       ;temp
                                ;*stack : angle/RAD/temp
                                ;angle in TREG
                                ;*stack : angle/rad/TEMP
MPY     *                       ;temp
LAC     #0e0c6h,15              ;0.4879
APAC                            ;ACCU=0.9826+temp

SACH    *,2                     ;treg = rad
MPY     *                       ;rad * temp
                                ;temp * angle in Q28
LAC     #05bc0h,5                ;0.0028
SACH    *,2                     ;result
MPY     *                       ;
LAC     #1000h,15               ;1
APAC
SETC    OVM
SACH    *,3
LAC     *,16
ADD     *,16
SACH    *
LAC     *-,0,AR1
                                ;*stack : angle/SIN
MAR     *-                      ;C compatibility
PSHD    *-
SBRK    1

RET
```