



Freescale Semiconductor

Digital Signal Processors

**Digital Sine-Wave
Synthesis Using the
DSP56001/2**

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2006

Illustrations

Figure 3-1	Integer Delta	3-2
Figure 3-2	“SINWGID” Routine	3-3
Figure 3-3	“SINWGID” Memory Map	3-3
Figure 3-4	Real Data Without Interpolation	3-5
Figure 3-5	“SINWGRD” Routine	3-6
Figure 3-6	“SINWGRD” Memory Map	3-7
Figure 3-7	“SINWGRD” Data ALU Programmer’s Model	3-7
		4-2
		4-3
		4-5
		4-6
		4-7
	Computation	A-5

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2006

List of Tables

Table 2-1	Sine-Wave Table Values	2-1
Table 3-1	Total Harmonic Distortion — Integer Delta	3-11
Table 3-2	Total Harmonic Distortion — Real Delta	3-12
Table 4-1	Total Harmonic Distortion — Real Delta	4-8
Table A-1	Single Precision Accuracy	A-4
Table A-2	Double Precision Accuracy	A-4

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2006

SECTION 1

Introduction

“This document describes three table look-up methods for sine-wave generation . . .”

Sine-wave generators are used in communications and control applications (1,2). With the introduction of high-speed high-precision digital signal processors, stable and low distortion sine waves of any frequency can be produced digitally using some form of table look-up with interpolation to reduce distortion (3,4,5). This document describes three table look-up methods for sine-wave generation and provides the Total Harmonic Distortion (THD) performance and Maximum

A routine for synthesizing sine waves having frequencies limited to integer multiples of the Fundamental

Section 3.1.

Section

a routine using only direct table look-up for synthesizing sine waves having frequencies which are fractional multiples of the FTF is described. This approach can be used to synthesize sine waves with frequencies which are not integer multiples of the FTF but they have substantially higher THD. A routine for synthesizing sine waves using table look-up with interpolation. Sine waves with frequencies which are not limited to multiples of the FTF and yet have low THD are possibly using this

SECTION 2

Background

“The maximum value delta can assume is $N/2$ since at least two samples per cycle are required to synthesize a sine wave without aliasing.”

The values used for approximating a sine wave are stored in a table in memory as follows:

Table 2-1 Sine-Wave Table Values

$i=N-1$	$\sin[(N-1) \cdot 360/N]$
	•
	•
$i \rightarrow$	$\sin[i \cdot 360/N]$

N - 1

the value stored at the j th location in the table. $i \cdot 360/N$ is the angle, in degrees, for which the sine function is calculated. Throughout the remainder of this document, the will be used to rep-

Note that the length of the table can be traded off against software in that, except for the sign, only one quarter of the table values are unique. The frequency of the digital sine wave generated depends upon the time interval and the phase angle increment (delta, Δ) between successive table accesses.

If delta is unity (i.e., the entries are read sequentially) and the table is accessed every T seconds (T is referred to as the sampling interval), the FTF of the sine wave synthesized will be:

$$\text{FTF} = 1/NT \text{ Hz} \qquad \text{Eqn. 2-1}$$

On the other hand, if delta is greater than unity, e.g., every second ($\Delta = 2$) or third ($\Delta = 3$) entry is read, (see Figure 3-2) and the table is still accessed every T seconds, then the frequency of the sine wave synthesized will be:

SECTION 3

Direct Table Look-Up

3.1 Integer Delta Implementation

“When fractional values of delta are used, samples of points between

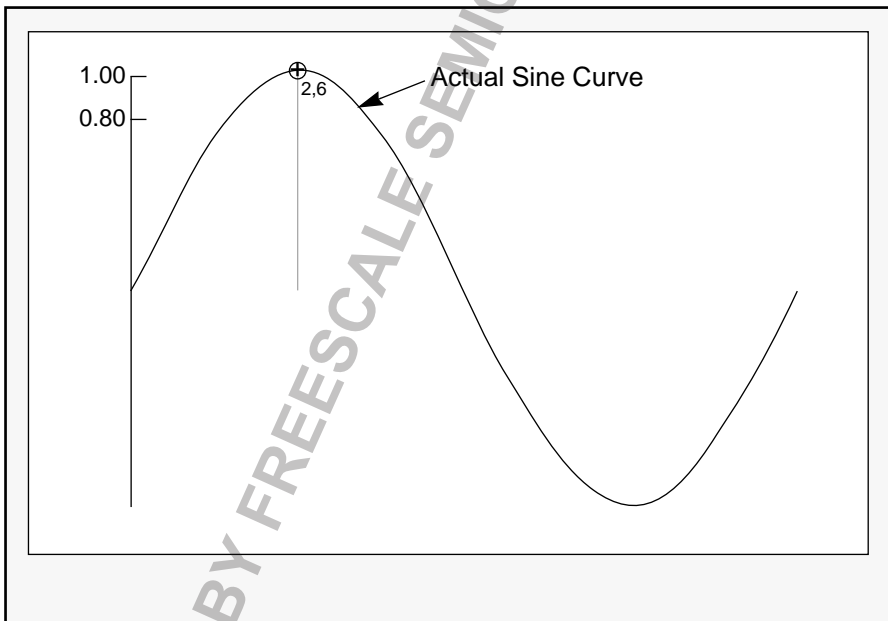
This implementation is a direct table look-up method with delta being a positive integer number. Because delta is limited to being an integer all the required samples are contained within the table; no approximations are necessary. Figure 3-1 illustrates this method using $N = 8$ and $\Delta = 2$.

The assembler listing for the SINE-Wave Generation Integer Delta (“SINWGID”) routine is presented in Figure 3-2. The corresponding memory map is shown in

Although no assembler options are indicated in this listing, a number of options are available. Refer to the Macro Assembler Reference Manual for information (6).

The memory locations can also be changed to suit the user's needs. In this routine the sine table ($N=256$) is in internal Y ROM starting at address HEX 100 to minimize external accesses and preserve RAM for data variables. The actual 256 sine table values stored are . The output location is chosen to

The part of the routine that generates the sine samples is given in the form of a subroutine to facilitate calculation of the MSF. Only a MOVEP and a JMP instruction must be executed to output a sample point. This MOVEP instruction utilizes the modulo addressing capability of the DSP56001/2. Three address arithmetic unit registers are used in this routine; the address register, R1, contains the index into the sine table, the offset register, N1, contains the value of delta, and the modifier register, M1, contains the value N-1 to set up the modulo buffer.



Motorola DSP56000 Macro Cross Assembler Version 1.10 06-18-87 13:03:01 sinwgid.asm
Page 1

```

1          page 132,50,0,10
2          ;SINWGID is a direct table look-up routine for SINE Wave Generation
3          ;using Integer Delta values. The sine table is in on-chip ROM starting
4          ;at address Y:$100 and contains 256 (N) entries which correspond to the
5          ;sines of 256 equally spaced angles between 0 and 360 degrees.
6          ;Integer delta values between 1 and N/2 can be used to step through the
7          ;table. Here Delta=2 and is saved in N1. The table size is saved in M1.
8          ;
9          00000060      start   equ   $60          ;starting address.
10         00000100      sinep   equ   $100         ;sine table starting address.
11         000000FF      mask    equ   255         ;set mask = table size - 1.
12         00000002      delta   equ   2          ;set delta = 2
13         0000FFE0      sinea   equ   $ffe0       ;address of output device.
14         P:0060        org     p:start
15         P:0060        0506BA      movec   #6, omr      ;enable on-chip ROM with sine table
16         P:0061        61F400      move    #sinep,r1    ;initialize sine table pointer.
17         000100
18         P:0063        05FFA1      movec   #mask,m1     ;set up modulo N addressing.
19         P:0064        390200      move    #delta,n1   ;offset equals Delta.
20         P:0065        0BF080      jsr    sineg        ;jump to subroutine.
21         000066
22         P:0067        09C9E0      sineg  movep   y:(r1)+n1,y:sinea ;output sample. (2)
23         P:0068        00000C      rts                    ;return from subroutine.(2)
24         end                    ;the end of listing.
0 Errors
0 Warnings

```

Resources required for the “SINWGID” routine:

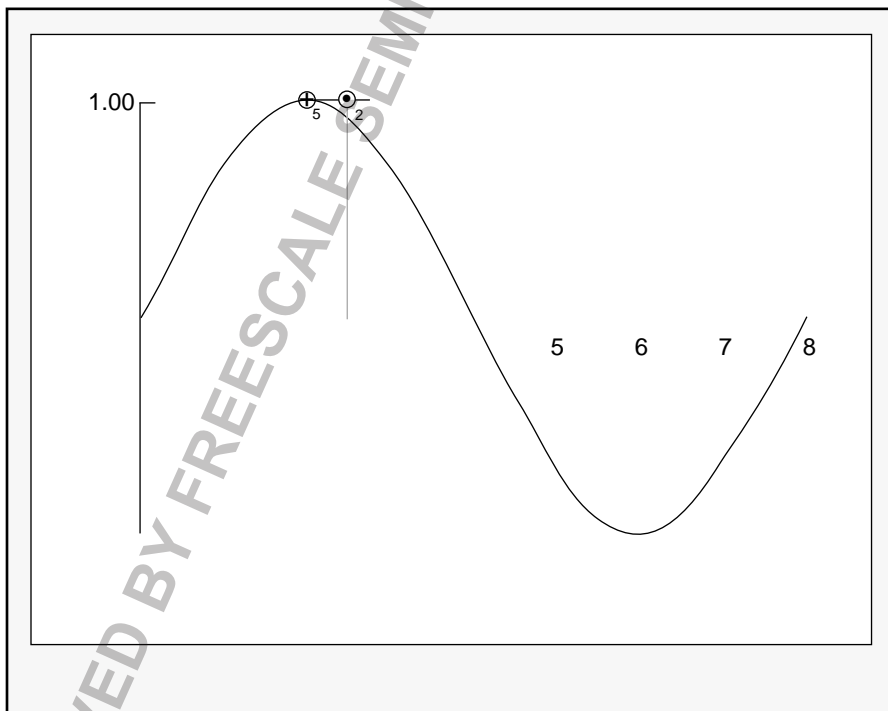
Data ROM	256	24-bit words for the sine table
Program Memory	6	24-bit words for the initialization
	2	24-bit words for the subroutine

The MSF for the given sine-wave subroutine would be achieved by replacing the RTS instruction with the JMP SINEG instruction. After performing the above replacement, MSF is given by:

$$\text{MSF} = \frac{1}{2 \cdot t_{\text{cyc}} \cdot \text{SIC}} \quad \text{Eqn. 3-1}$$

3.2 Real Delta Implementation

This implementation is a direct table look-up method with delta being a positive real number, that is, a number consisting of an integer and a fractional part. When fractional values of delta are used, samples of points between table entries must be estimated using the table values. The most straightforward estimation is to use the previous table entry. This approach is described in this section and illustrated in Figure 3-4 for $N = 8$ and $\Delta = 2.5$.



The assembler listing for the SINE-Wave Generation Real Delta (“SINWGRD”) routine is presented in Figure 3-5. The memory map (see Figure 3-6) is the same as that used for the SINWGID routine except that R4 is used to point to the output device. This saves one cycle since the previous sample can be output while the accumulator is being updated using the parallel move feature of the DSP56001/2. The programmer's model for the Data ALU is presented in Figure 3-7.

```

Motorola DSP56000 Macro Cross Assembler Version 1.10 06-18-87 13:03:27 sinwgrd.asm Page 1
1      page      132,50,0,10
2      ;SINWGRD is a direct table look-up routine for SINE Wave Generation using
3      ;positive Real numbers for Delta. The sine table is in on-chip ROM starting at
4      ;address r:$100 and contains 256 (N) entries corresponding to the sine values
5      ;of 256 equally spaced angles between 0 and 360 degrees. N must be a power of
6      ;two. Delta can be between 0.0 and N/2. Here delta = 2.5 and is saved in B1 and
7      ;B0. The table size minus one, N-1, is saved in X1.

```

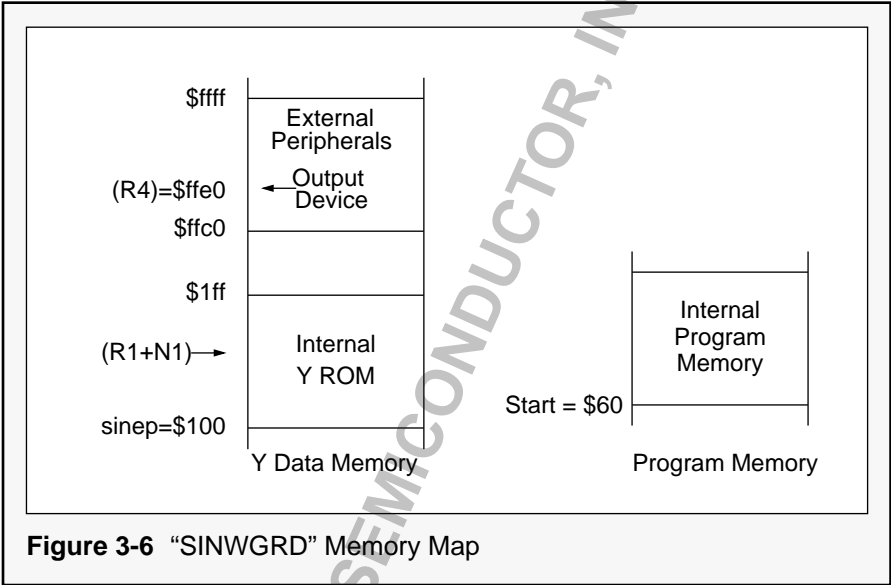
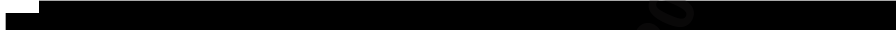


Figure 3-6 "SINWGRD" Memory Map



Given delta is now a real number, the inherent modulo addressing mode in the DSP56001/2. cannot be used and therefore a modulo addressing scheme must be implemented in software. The following is a description of one approach to implementing such a scheme. This approach uses both accumulators.

In this routine the integer part of delta is stored in the upper portion of an accumulator (B1 for the example given), and the fractional part is stored in the lower portion of the same accumulator (B0). Care should be taken to ensure that the fractional part is stored correctly. The number moved into B0 as a result of subtracting the integer portion from delta, i.e. $\Delta - \text{cvi}(\Delta)$, is a signed, i.e. positive, fraction. This is undesirable since the sign bit should be associated with the integer portion. Therefore a shift to the left



A second accumulator, A in this example, contains the positive real number value used to load the offset register, N1, which is used to offset the pointer, R1. R1 is used to address the correct location in the sine table. Since the table index must be an integer, only A1 is moved to N1. Incrementing by delta, however, is done on both the integer and fractional parts of the A accumulator. It should be noted that the increment of the register R1 is done by indexing the register with the offset register N1, thus leaving the original value of R1 (base address) unchanged. Note that the binary point is at an imaginary point between the two 24-bit parts of the accumulators. Initially we have:

$$\text{ACC. A} = \boxed{\begin{array}{c} 0 \\ \text{A1} \end{array}} \cdot \boxed{\begin{array}{c} 0 \\ \text{A0} \end{array}}$$

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2006

Resources required for the “SINWGRD” routine:

Data ROM	256	24-bit words for the sine table
Program Memory	11	24-bit words for the initialization
	4	24-bit words for the subroutine memory
MSF		$1/(2 \cdot 50 \cdot 6)$ MHz
		= 1.667 MHz

3.3 Harmonic Distortion

Due to the fact that the sine wave generated is an approximation, not all of the energy is at the funda-

Eqn.

1. Quantization Error: Since the sine table values stored in memory are of finite word length (24 bits in this case) the sine values cannot be represented exactly. Quantization error is directly proportional to the word length. For example the sine of 45 degrees in decimal will be:

0.7070707 using 24 bits

0.7071 using 16 bits

2. Sampling Error: When points between table entries are sampled, i.e., delta is not an integer, then large errors are introduced because these points must be estimated from the table values. Since the sampling errors are derived from the table values, sampling errors are always greater than quantization errors.

The THD for different deltas and different table sizes is shown in Table 3-1(a), Table 3-1(b), and Table 3-2.

Table 3-1(b) Total Harmonic Distortion — Integer Delta

Δ	N = 64	N = 256
1	$3.4157912 \cdot 10^{-15}$	$2.6423040 \cdot 10^{-15}$
2	$2.8356370 \cdot 10^{-15}$	$2.8659804 \cdot 10^{-15}$
3	$3.4157912 \cdot 10^{-15}$	$2.6423040 \cdot 10^{-15}$
4	$3.6274702 \cdot 10^{-15}$	$3.4157912 \cdot 10^{-15}$
5	$3.4157912 \cdot 10^{-15}$	$2.6423040 \cdot 10^{-15}$
6	$2.8356370 \cdot 10^{-15}$	$2.8659804 \cdot 10^{-15}$
7	$3.4157912 \cdot 10^{-15}$	$2.6423040 \cdot 10^{-15}$
8	$5.7949069 \cdot 10^{-15}$	$2.8356370 \cdot 10^{-15}$
9	$3.4157912 \cdot 10^{-15}$	$2.6423040 \cdot 10^{-15}$
10	$2.8356370 \cdot 10^{-15}$	$2.8659804 \cdot 10^{-15}$

The equations as well as the FORTRAN code used for calculating the THD in the above tables are included in **Appendix A**. With respect to Table 3-1(a), Table 3-1(b), and Table 3-2 some important observations and conclusions can be drawn:

1. For integer delta the THD for $N \leq 1024$ is of the same order of magnitude. This can be seen in Table 3-1(a) and Table 3-1(b). For integer deltas the only errors which cause distortion are quantization errors.
2. For integer delta and small N the THD is nonuniform. This is evident for $\Delta = 2$ and $N \leq 128$ or $\Delta = 3$ and $N \leq 64$ in Table 3-1(a). For small N , the quantization error distribution is not uniformly distributed between $\pm 1/2$ Least Significant Bit (LSB) thereby causing the THD values to be nonmonotonic.
3. For integer delta and large N the THD decreases

5. THD depends on the fractional part of delta. The THD for $\Delta = 2.50$ is less than either 2.25 or 2.75 (see Table 3-2), because the fractional part generates an integer every other access for $\Delta = 2.50$ and every other fourth access for 2.25 or 2.75. Therefore, every other sample is free of sampling error for $\Delta = 2.5$ but only every fourth sample is free of sampling error for the other two cases. Observe that the THD for 2.25, 2.75, and 8.25 is the same. The THD for 11.625 is slightly higher because the fractional part forms an integer only every eighth access.
6. The THD for non-integer deltas decreases with increasing table length as seen in Table 3-2. Consider the same delta entries for different N. By increasing the table length, the difference between table entries decreases, resulting in better approximations to the non-integer samples, hence reduced sampling errors. ■

SECTION 4

Table Look-Up with Linear Interpolation

“... the THD improves dramatically for non-integer values of delta when using linear

In order to synthesize a sine wave of any frequency with low distortion, an interpolation method must be used together with table look-up. By using interpolation, sine values between table entries can be represented more accurately. The easiest interpolation technique to implement is linear interpolation. For linear interpolation the sine value for a point between successive table entries is assumed to lie on the straight line between the two values. This is illustrated in Figure 4-1 for $\delta = 2.5$ and $N = 8$. Con-

The algorithm used for linear interpolation is based

For linear interpolation:

$m = S[i + 1] - S[i]$; i.e., it is the slope of the line segment between successive table entries i and $i + 1$

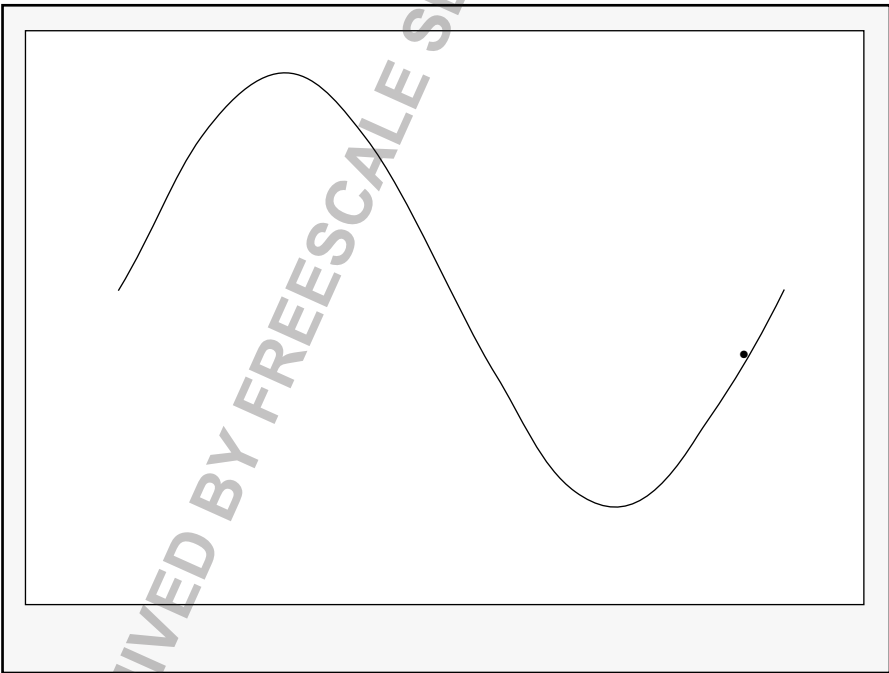
$b = S[i]$, the value of the sine table at the base address plus the j^{th} offset location

$x =$ the fractional part of the pointer,
 $0 < x < 1.0$

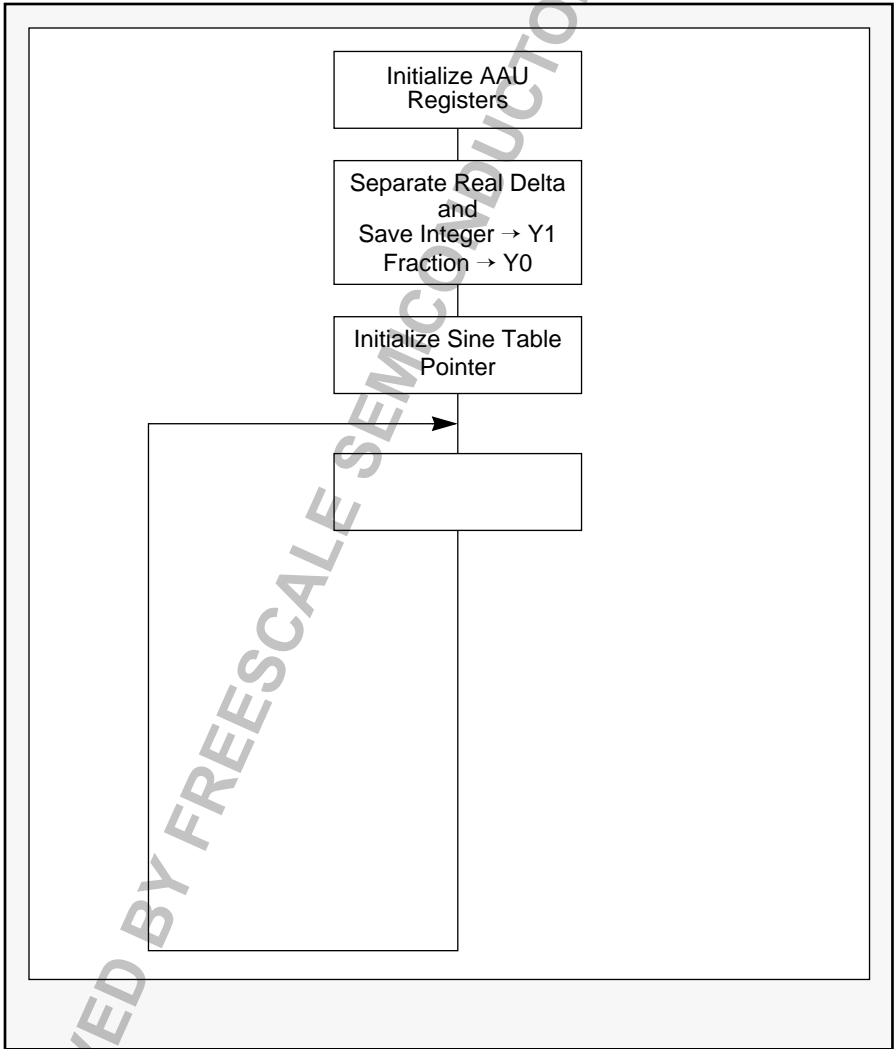
$y = S[i + x]$, the approximated sample value

Therefore, we have:

$$S[i+x] = S[i] + x \cdot \{S[i+1]-S[i]\}$$



The program flow diagram for the SINE-Wave Generation Linear Interpolation (“SINWGLI”) routine is presented in Figure 4-2.



ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2006

The assembler listing is given in Figure 4-3; the memory map is presented in Figure 4-4; and the Data ALU programmer's model is presented in Figure 4-5.

The sine table is stored in Y ROM starting at HEX 100. The address pointer, R1, points to the current sine value while address pointer, R2, points to the current plus one sine value, $R2=R1+1$. The slope between successive points is determined by subtracting the table entry pointed to by R2 from the table entry pointed to by R1.

For the "SINWGLI" routine, the fractional part of the linear approximation, x, is stored in the accumulator, A0, and the integer portion is stored in A1. A1 is "moved" into N1 which is used to index the

Motorola DSP56000 Macro Cross Assembler Version 1.10 06-18-87 17:19:09 sinwgli.asm Page 1

SINWGLI is a direct table look-up routine for SINE Wave. Generation using Linear Interpolation with any positive real delta greater than 0.0 and less than $N/2.0$. Here delta = 2.5 and is saved in y1 and y0. The sine table contains 256 (N) entries corresponding to the sines of 256 equally spaced angles between 0 and 360 degrees. N must be a power of two. The table size minus one, N-1, is saved in N0.

```
;program starting address
;sine table starting address
;base address plus N minus one
;delta = 2.5
;address of output device

;enable on-chip ROM with sine table
;set up output pointer

;store fraction part in b0

;eliminate sign bit, initialize r1

;integer part of Delta in y1

;unsigned fractional Delta in y0
;initialize a and n1
;set up mask value
;initialize n2
;initialize r2 with the value r1+1

;jump to subroutine

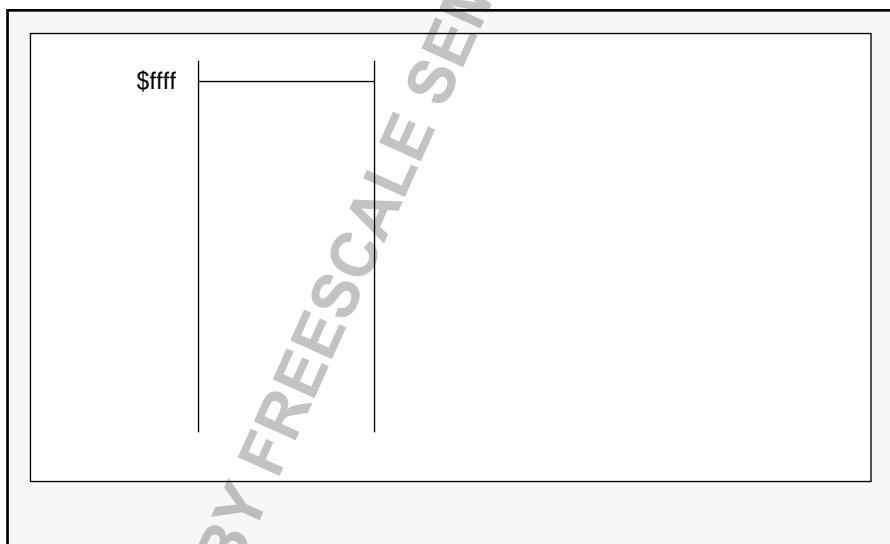
;fraction part of ptr in b1, b0=0 (1)
;b1=signed fr. Delta, x0=sine (2)
;sign. fract. in x1, sine in b1 (2)
;subtr sines to obtain slope (1)
;slope value to x0, sine in b1 (2)
;sample approximation = m*x+b (1)
;update calculation, output sample (1)
;set up mask value in x1 (1)
;mask a1 = modulo arithmetic (1)
;update offset register n1 (1)
;update offset register n2 (1)
;return from the subroutine (1)
;the end of the routine (2)
```

```
y:(r1+n1),x0
y:(r2+n2),b
y:(r1+n1),b
b,y:(r4)
```

```
;(delta @cvi(delta)),b0
```

Figure 4-3 "SINWGLI" Routine

Six address arithmetic unit registers are used in this routine; besides R1, R2, R4, N1, and N2, the modifier register M2 is used. The use of M2 is twofold. First, M2 is used to store the mask value before it is used in the AND masking operation. (This masking operation constrains $R1 + N1$ within the desired region in memory, i.e., implements modulo addressing.) Second, it is used to activate modulo arithmetic updates of $R2 + N2$. This is necessary to take care of the special case where $R1 + N1$ points to \$1FF. Since $R2 + N2$ equals $R1 + N1 + 1$, it would point to \$200 instead of \$100 without modulo arithmetic.



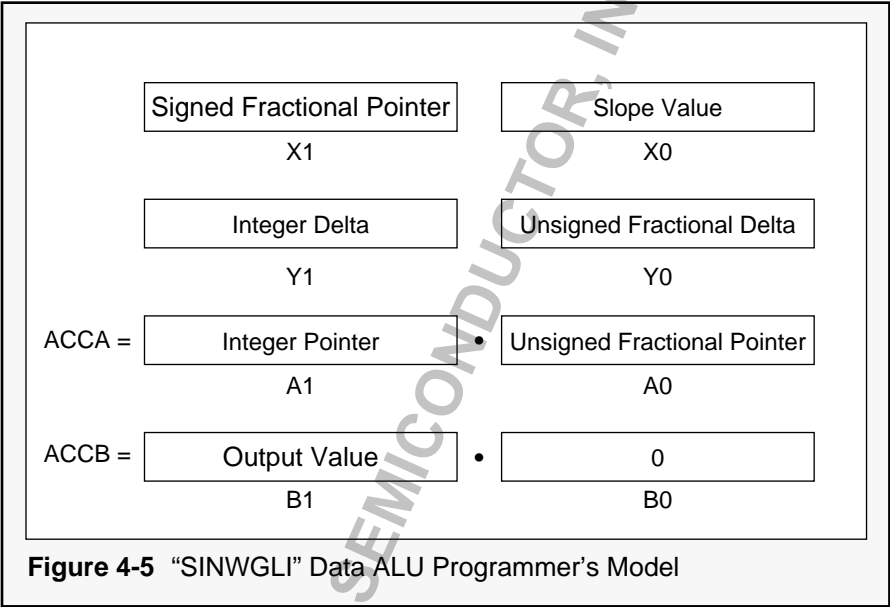


Figure 4-5 "SINWGLI" Data ALU Programmer's Model

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2006

4.1 Harmonic Distortion

As expected the THD improves dramatically for non-integer values of delta when using linear interpolation. This is of course because the intermediate points are better approximated. The performance for integer deltas remains the same as that for the direct table look-up methods already discussed. The results for the THD using linear interpolation are shown in Table 4-1 and these results should be directly compared with the results in Table 3-2.

With respect to Table 4-1 some important observations and conclusions can be drawn:

1. THD is reduced by 103 by using linear interpolation with direct table look-up. This can be verified by comparing Table 3-2 with Table 4-1. This is because an extra correctional value is added to the sine value obtained from the sine table. This correctional value is the product of the slope at the specific point and the fractional part of the pointer (i.e., $m-x$).
2. THD depends only on the fractional part of delta when linear interpolation is used. It is a maximum for the fractional part = 0.5 and symmetrical about this midpoint. This is evident in Table 4-1. This is because the linear approximation is poorest at the midpoint. ■

APPENDIX A

Computation of Total Harmonic Distortion (THD)

“To determine the precision needed to calculate THD, consider the inherent

The equation for calculating THD (given in Eqn. 3-2) can be rewritten as:

$$\text{THD} = \frac{E_T - E_F}{E_T} \quad \text{Eqn. A-1}$$

For accurate and correct results, the above energy terms must be calculated over a full cycle of the synthesized sine wave. In the case of direct table lookup, a full cycle may require several passes through the sine table. The number of passes depends on the

A full cycle will be synthesized for the smallest n for which n is evenly divisible by N . For example, if $N = 256$ and $\text{cycles} = 2.5$, then a complete cycle occurs for $n = 256$, since $256 \cdot 2.5/128 = 5$. Figure 3-4 illustrates an example where $N = 128$ and $\text{cycles} = 2.5$. The $(2 \cdot 8 =) 16$ points required

In general, if $\Delta = A/B$, where A and B are relatively prime numbers, then the minimum number of samples, N', which must be output to synthesize a full cycle is $N' = B \cdot N$.

The total energy, ET, in a cycle of length BN (BN = B • N) is given by:

$$ET = \sum_{i=0}^{BN-1} x(i) \cdot x(i) \quad \text{Eqn. A-2}$$

where: $x(i)$ is the i^{th} sample of the sine-wave sequence

The amount of energy in the fundamental frequency, EF, over the same period is given by:

$$EF = (1/BN) \cdot (|X(A)|^2 + |X(BN - A)|^2)$$

The VAX VMS FORTRAN source code used for the computation of THD is given in Figure A-1.

Some important details concerning the computation of the THD should be noted. ENER3 and (ENER2-EFUND) ideally should have the same value. However, a difference may occur because ENER2 is much greater than EFUND. The result of the subtraction will be inaccurate due to numerical precision limitations. To determine the precision needed to calculate THD, consider the inherent symmetry in the DFT. For a real sequence $x(n)$, the DFT sequence, $X(k)$, exhibits the following symmetries (10):

1. $\text{Re}\{X(k)\} = \text{Re}\{X(N - k)\}$
2. $\text{Im}\{X(k)\} = -\text{Im}\{X(N - k)\}$

This, in turn, means that the following equalities should hold:

$$\begin{aligned}\cos(2 \cdot \pi \cdot k \cdot n/N) &= \cos(2 \cdot \pi \cdot (N-k) \cdot n/N) \\ \sin(2 \cdot \pi \cdot k \cdot n/N) &= -\sin(2 \cdot \pi \cdot (N-k) \cdot n/N)\end{aligned}$$

Eqn. A-5

where: $n = 0, 1, 2, \dots, BN-1$

$k = 0, 1, 2, \dots, BN-1$

If single precision accuracy is used in calculating the THD, the above equalities do **NOT** hold. For example,

$$\begin{aligned}\text{let } k &= 1 \\ n &= 128 \\ \text{ARG1} &= 2 \cdot \pi \cdot n \cdot 1/128 \\ \text{ARG2} &= 2 \cdot \pi \cdot n \cdot 127/128\end{aligned}$$

Then, as shown in Table A-1, $\cos(\text{ARG1})$ does not equal $\cos(\text{ARG2})$ when truncating to only 8 digits. If, however, double precision is used, the equalities

```

C Program THD.FOR Using double precision real arithmetic
C Program to calculate THD for digital sine-wave generation without interpolation.

IMPLICIT COMPLEX*16(X)
REAL*8 ENERGL,ENERG2,ENERG3,S(2048),TWOPI,ISS(2048)
DIMENSION X(0:2048)
INTEGER*4 IS(2048)
TYPE *,' ENTER TABLE SIZE (UP TO 2048) '
ACCEPT *,NN
TYPE *,'ENTER INPUT HEX FILENAME'
C the hex values are assumed to be 11 on each line, each consisting of 6 characters
C and having a blank space in between them. This is the format of the "LOD" file
C that is generated by the assembler [5].
READ(1,110)(IS(I),I=1,NN)
110 FORMAT(11(Z6,X))
DO 120 I=1,NN

IF(BITTEST(IS(I),23)) THEN
    IS(I)=JIOR(IS(I),'FF000000'X)
ENDIF
120 CONTINUE

DO 100 I=2,NN
ISS(I)=IS(I)
100 CONTINUE
TYPE *,' ENTER DELTA, BN, A '
ACCEPT *,DELTA,BN,A
AINDEX=0.0
DO 200 IABC=1,BN
INDEX=INT(AINDEX)
S(IABC)=DBLE(ISS(INDEX+1))

```

APPENDIX B

The Sine Table

Table B-1 gives the sine entries for $N = 256$. The hexadecimal values are given in the signed fractional format used for the DSP56001. "\$" denotes a hexadecimal value.

Table B-1 Sine Table

ADDRESS	VALUE	ADDRESS	VALUE
S_01	\$03242B	S_12	\$36BA20
S_02	\$0647D9	S_13	\$398CDD
S_03	\$096A90	S_14	\$3C56BA
		S_15	\$3F174A
		S_16	\$41CE1E
		S_17	\$447ACD
		S_18	\$471CED
		S_19	\$49B415
		S_1B	\$4EBFE9
		S_1C	\$5133CD
		S_1D	\$539B2B
		S_1E	\$55F5A5
			\$5A827A
		S_21	\$5CB421
		S_22	\$5ED77D

Table B-1 Sine Table (continued)

ADDRESS	VALUE
S_23	\$60EC38
S_24	\$62F202
S_25	\$64E889
S_26	\$66CF81
S_27	\$68A69F
S_28	\$6A6D99
S_29	\$6C2429
S_2A	\$6DCA0D
S_2B	\$6F5F03
S_2C	\$70E2CC
S_2D	\$72552D

ADDRESS	VALUE
S_3C	\$7F6237
S_3D	\$7FA737
S_3E	\$7FD888
S_3F	\$7FF622
S_40	\$7FFFFF
S_41	\$7FF622
S_42	\$7FD888
S_43	\$7FA737
S_44	\$7F6237
S_45	\$7F0992
S_46	\$7E9D56

Table B-1 Sine Table (continued)

ADDRESS	VALUE
S_55	\$6F5F03
S_56	\$6DCA0D
S_57	\$6C2429
S_58	\$6A6D99
S_59	\$68A69F
S_5A	\$66CF81
S_5B	\$64E889
S_5C	\$62F202
S_5D	\$60EC38
S_5E	\$5ED77D
S_5F	\$5CB421

ADDRESS	VALUE
S_6E	\$36BA20
S_6F	\$33DEF3
S_70	\$30FBC5
S_71	\$2E110A
S_72	\$2B1F35
S_73	\$2826B9
S_74	\$25280C
S_75	\$2223A5
S_76	\$1F19F9
S_77	\$1C0B82
S_78	\$18F8B8

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2006

Table B-1 Sine Table (continued)

ADDRESS	VALUE
S_87	\$EA1DEC
S_88	\$E70748
S_89	\$E3F47E
S_8A	\$E0E607
S_8B	\$DDDC5B
S_8C	\$DAD7F4
S_8D	\$D7D947
S_8E	\$D4E0CB
S_8F	\$D1EEF6
S_90	\$CF043B
S_91	\$CC210D

ADDRESS	VALUE
S_A0	\$A57D86
S_A1	\$A34BDF
S_A2	\$A12883
S_A3	\$9F13C8
S_A4	\$9D0DFE
S_A5	\$9B1777
S_A6	\$99307F
S_A7	\$975961
S_A8	\$959267
S_A9	\$93DBD7
S_AA	\$9235F3

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2006

Table B-1 Sine Table (continued)

ADDRESS	VALUE
S_B9	\$81E26C
S_BA	\$8162AA
S_BB	\$80F66E
S_BC	\$809DC9
S_BD	\$8058C9
S_BE	\$802778
S_BF	\$8009DE
S_C0	\$800000
S_C1	\$8009DE
S_C2	\$802778
S_C3	\$8058C9

ADDRESS	VALUE
S_D2	\$8C4A14
S_D3	\$8DAAD3
S_D4	\$8F1D34
S_D5	\$90A0FD
S_D6	\$9235F3
S_D7	\$93DBD7
S_D8	\$959267
S_D9	\$975961
S_DA	\$99307F
S_DB	\$9B1777
S_DC	\$9D0DFE

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2006

Table B-1 Sine Table (continued)

ADDRESS	VALUE
S_EB	\$C0E8B6
S_EC	\$C3A946
S_ED	\$C67323
S_EE	\$C945E0
S_EF	\$CC210D
S_F0	\$CF043B
S_F1	\$D1EEF6
S_F2	\$D4E0CB
S_F3	\$D7D947
S_F4	\$DAD7F4
S_F5	\$DDDC5B

REFERENCES

1. J. Tierney, "Digital Frequency Synthesizers", Chapter V of Frequency Synthesis: Techniques and Applications, J. Gorski-Popiel, ed., N.Y., IEEE Press, 1975.
2. T. F. Quatieri, R. J. McAulay, "Speech Transformations Based on a Sinusoidal Representation", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-34, 1449-1464, December 1986.
3. D. L. Duttweiler and D. G. Messerschmitt, "Analysis of Digitally Generated Sinusoids with Application to A/D and D/A Converter Testing", IEEE Transactions on Communications, vol. COM-26, pp. 669-675, May 1978.

S. Mehrgardt, "Noise Spectra of Digital Sine Generators Using the Table-Lookup Method", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP31, pp. 1037-1039, 1983.

J. Tierney, C. M. Rader, and B. Gold, "A Digital Filter for Sinusoidal Frequency Synthesis", IEEE Transactions on Audio and Electroacoustics, vol. AU-19, pp. 48-56, 1971.

DSP56000 Macro Assembler Reference Manual,

AN56000-01, Rev. 1.0, 1985.
Precision Digital Sine-Wave Generation with the

DSP56000 Digital Signal Processor User's

9. D. Borth, Motorola Internal Memo on Digital Signal Processing, June 1986.
10. A. V. Oppenheim and R. W. Schaffer, Digital Signal Processing, Prentice-Hall, 1975.

ARCHIVED BY FREESCALE SEMICONDUCTORS INC. 2006



How to Reach Us:

Home Page:

www.freescale.com

E-mail:

support@freescale.com

USA/Europe or Locations Not Listed:

Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document. Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

