DACSYNC PAGE 1

1 ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2 ;

3 ; Author : ADI - Apps www.analog.com/MicroConverter

4 ;

5 ; Date : October 2003

6 ;

7 ; File : DACsync.asm

8 ;

9 ; Hardware : ADuC841

10 ;

11 ; Description : Outputs sine waves on DAC0 and DAC1 at 3.08kHz.

12 ; Output signals are in quadrature with eachother,

13 ; DAC1 leading DAC0 by 90 degrees. the SYNC bit is

14 ; used to ensure that both DAC outputs update

15 ; simultaneously thus avoiding a phase error of 0.625

16 ; degrees.

17 ; Rate calculations assume an 11.0592MHz Mclk.

18 ;

19 ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

20

21 $MOD841 ; Use 8052&ADuC841 predefined symbols

22

00B4 23 LED EQU P3.4 ; P3.4 drives red LED on eval board

24

25 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

26 ; BEGINNING OF CODE

---- 27 CSEG

28

0000 29 ORG 0000h

0000 75EF80 30 MOV ADCCON1,#80H

0003 75FD1F 31 MOV DACCON,#01Fh ; both DACs on,12bit,asynchronous

0006 75FA08 32 MOV DAC0H,#008h

0009 75F900 33 MOV DAC0L,#000h ; DAC0 to mid-scale

000C 75FC0F 34 MOV DAC1H,#00Fh

000F 75FBFF 35 MOV DAC1L,#0FFh ; DAC1 to full-scale

36

0012 901000 37 MOV DPTR,#TABLE

38

0015 53FDFB 39 STEP: ANL DACCON,#0FBh ; clear SYNC bit

40

0018 E4 41 CLR A ;

0019 93 42 MOVC A,@A+DPTR ; get high byte for mainDAC..

001A F5FA 43 MOV DAC0H,A ; ..and move it into DAC0 register

001C 7420 44 MOV A,#020h ; offset by 90deg for quadratureDAC

001E 93 45 MOVC A,@A+DPTR ; get high byte for quadratureDAC..

001F F5FC 46 MOV DAC1H,A ; ..and move it into DAC1 register

0021 A3 47 INC DPTR ; move on to get low bytes

48

0022 E4 49 CLR A ;

0023 93 50 MOVC A,@A+DPTR ; get low byte for mainDAC..

0024 F5F9 51 MOV DAC0L,A ; ..and update DAC0

0026 7420 52 MOV A,#020h ; offset by 90deg for quadratureDAC

0028 93 53 MOVC A,@A+DPTR ; get low byte for quadratureDAC..

0029 F5FB 54 MOV DAC1L,A ; ..and update DAC1

002B A3 55 INC DPTR ; move on for next data point

56

002C 43FD04 57 ORL DACCON,#004h ; set SYNC bit

58

DACSYNC PAGE 2

002F 53827F 59 ANL DPL,#07Fh ; wrap around at end of table

60

0032 E5FA 61 MOV A,DAC0H ;

0034 A2E3 62 MOV C,ACC.3 ; MSB of DAC0 value

0036 92B4 63 MOV LED,C ; LED = MSB of DAC0

64

0038 00 65 NOP ;

0039 00 66 NOP ;

67

003A 80D9 68 JMP STEP ;

69

70

71 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

72 ; SINE LOOKUP TABLE

1000 73 ORG 01000h

74

1000 75 TABLE:

76

1000 07FF 77 DB 007h, 0FFh

1002 08C8 78 DB 008h, 0C8h

1004 098E 79 DB 009h, 08Eh

1006 0A51 80 DB 00Ah, 051h

1008 0B0F 81 DB 00Bh, 00Fh

100A 0BC4 82 DB 00Bh, 0C4h

100C 0C71 83 DB 00Ch, 071h

100E 0D12 84 DB 00Dh, 012h

1010 0DA7 85 DB 00Dh, 0A7h

1012 0E2E 86 DB 00Eh, 02Eh

1014 0EA5 87 DB 00Eh, 0A5h

1016 0F0D 88 DB 00Fh, 00Dh

1018 0F63 89 DB 00Fh, 063h

101A 0FA6 90 DB 00Fh, 0A6h

101C 0FD7 91 DB 00Fh, 0D7h

101E 0FF5 92 DB 00Fh, 0F5h

1020 0FFF 93 DB 00Fh, 0FFh

1022 0FF5 94 DB 00Fh, 0F5h

1024 0FD7 95 DB 00Fh, 0D7h

1026 0FA6 96 DB 00Fh, 0A6h

1028 0F63 97 DB 00Fh, 063h

102A 0F0D 98 DB 00Fh, 00Dh

102C 0EA5 99 DB 00Eh, 0A5h

102E 0E2E 100 DB 00Eh, 02Eh

1030 0DA7 101 DB 00Dh, 0A7h

1032 0D12 102 DB 00Dh, 012h

1034 0C71 103 DB 00Ch, 071h

1036 0BC4 104 DB 00Bh, 0C4h

1038 0B0F 105 DB 00Bh, 00Fh

103A 0A51 106 DB 00Ah, 051h

103C 098E 107 DB 009h, 08Eh

103E 08C8 108 DB 008h, 0C8h

1040 07FF 109 DB 007h, 0FFh

1042 0736 110 DB 007h, 036h

1044 0670 111 DB 006h, 070h

1046 05AD 112 DB 005h, 0ADh

1048 04EF 113 DB 004h, 0EFh

104A 043A 114 DB 004h, 03Ah

104C 038D 115 DB 003h, 08Dh

104E 02EC 116 DB 002h, 0ECh

DACSYNC PAGE 3

1050 0257 117 DB 002h, 057h

1052 01D0 118 DB 001h, 0D0h

1054 0159 119 DB 001h, 059h

1056 00F1 120 DB 000h, 0F1h

1058 009B 121 DB 000h, 09Bh

105A 0058 122 DB 000h, 058h

105C 0027 123 DB 000h, 027h

105E 0009 124 DB 000h, 009h

1060 0000 125 DB 000h, 000h

1062 0009 126 DB 000h, 009h

1064 0027 127 DB 000h, 027h

1066 0058 128 DB 000h, 058h

1068 009B 129 DB 000h, 09Bh

106A 00F1 130 DB 000h, 0F1h

106C 0159 131 DB 001h, 059h

106E 01D0 132 DB 001h, 0D0h

1070 0257 133 DB 002h, 057h

1072 02EC 134 DB 002h, 0ECh

1074 038D 135 DB 003h, 08Dh

1076 043A 136 DB 004h, 03Ah

1078 04EF 137 DB 004h, 0EFh

107A 05AD 138 DB 005h, 0ADh

107C 0670 139 DB 006h, 070h

107E 0736 140 DB 007h, 036h ; end of table

141

1080 07FF 142 DB 007h, 0FFh ; repeat first 90degrees for quadratureDAC

1082 08C8 143 DB 008h, 0C8h

1084 098E 144 DB 009h, 08Eh

1086 0A51 145 DB 00Ah, 051h

1088 0B0F 146 DB 00Bh, 00Fh

108A 0BC4 147 DB 00Bh, 0C4h

108C 0C71 148 DB 00Ch, 071h

108E 0D12 149 DB 00Dh, 012h

1090 0DA7 150 DB 00Dh, 0A7h

1092 0E2E 151 DB 00Eh, 02Eh

1094 0EA5 152 DB 00Eh, 0A5h

1096 0F0D 153 DB 00Fh, 00Dh

1098 0F63 154 DB 00Fh, 063h

109A 0FA6 155 DB 00Fh, 0A6h

109C 0FD7 156 DB 00Fh, 0D7h

109E 0FF5 157 DB 00Fh, 0F5h

10A0 0FFF 158 DB 00Fh, 0FFh

159

160 ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

161

162 END

VERSION 1.2h ASSEMBLY COMPLETE, 0 ERRORS FOUND

DACSYNC PAGE 4

ACC. . . . . . . . . . . . . . . D ADDR 00E0H PREDEFINED

ADCCON1. . . . . . . . . . . . . D ADDR 00EFH PREDEFINED

DAC0H. . . . . . . . . . . . . . D ADDR 00FAH PREDEFINED

DAC0L. . . . . . . . . . . . . . D ADDR 00F9H PREDEFINED

DAC1H. . . . . . . . . . . . . . D ADDR 00FCH PREDEFINED

DAC1L. . . . . . . . . . . . . . D ADDR 00FBH PREDEFINED

DACCON . . . . . . . . . . . . . D ADDR 00FDH PREDEFINED

DPL. . . . . . . . . . . . . . . D ADDR 0082H PREDEFINED

LED. . . . . . . . . . . . . . . NUMB 00B4H

P3 . . . . . . . . . . . . . . . D ADDR 00B0H PREDEFINED

STEP . . . . . . . . . . . . . . C ADDR 0015H

TABLE. . . . . . . . . . . . . . C ADDR 1000H