Implementing DTMF Detection using the Silicon Laboratories DAA (Data Access Arrangement)

Application Note 19

Abraham Si

September 1999

1.0 Overview

Traditional modem designs involves the use of AFE (analog front end), isolation transformer, relays, opto-isolators, and 2-to-4 wire hybrid. The Silicon Laboratories DDAA(Direct Digital Access Arrangement) or pure silicon DAA, has drastically reduced the number of components required to implement the analog side of a modem.

Using the Si3032-EVB rev 1.4 and the Parallax SX demo board, we have been able to design a program to do the following:

- Initialization for desired sampling rate
- Ring detect
- Going off hook
- Obtain DTMF data and detect the digits properly

2.0 Preliminary Issues

To interface to the Silicon Labs Si3032, several issues needed to be resolved:

- MCLK frequency
- Interface mode
- Method for control information transfer

2.1 MCLK Frequency

All clocks for the Si3032 are derived from a single MCLK input frequency, which is chosen to be 4 MHz in this case. From this, 2 constants, N1, M1, are needed to generate a base frequency. From Table 14 of the Si3032 datasheet, we see that N1=5 and M1=72. Furthermore, CGM needs to be set to 1 to satisfy the following relationship:

Fbase=Fmclk * M1 * 16/ (N1 * 25) = 36. 864 MHz, CGM=1

At reset, the SCLK frequency will be 200KHz, in accordance with the following formula:

SCLK = (M1* MCLK)/ (20 * N1)

N1 and M1 are both 0 at reset. But their actual values are 1+register reading, and thus equal to 1 at reset, so we have:

SCLK= 4 MHz/20 = 200 KHz

After programming a value of 4 into N1 and a value of 71 into M1 (1 less than actual values), we obtain:

After programming N1 and M1, it will be changed to 2.88MHz.

SCLK = (72 * 4 MHz) / (20 * 5) = 2.88 MHz

To obtain a sampling rate of 8 KHz, N2 and M2 need to be programmed as well. From table 13 of the Si3032 data sheet, we see that N2=9 and M2=10. So the values to program in are 8 and 9 respectively.

Summarizing, we have the following values to write:

N1=4, M1=71, CGM=1, N2=8, M2=9.

Note that after programming N2 and M2, SCLK frequency will become 2.048 MHz. This change of SCLK frequency presents challenge for SX since we have to deal with three frequencies: 200K, 2.88 MHz, and 2.048 MHz. The 200K SCLK is easily dealt with by synchronizing to FSYNC first, then use RTCC interrupt to sample SDO and output SDI at every other interrupt. The 2.88MHz and 2.048MHz situation are more delicate.16 bits of SDO are sampled and 16 bits of SDI are output inside the ISR (interrupt service routine) after the rising edge of FSYNC.

2.2 Interface Mode

The digital interface to the DAA is a standard TDM bus interface composed of SCLK, SDI, SDO, and FSYNC. We can choose two modes for the FSYNC, either pulse or frame. In the pulse mode, the FSYNC pulse precedes the data by 1 SCLK period. In the frame mode, the FSYNC signal is low when the data is transmitted. The pulse mode (mode 1) is chosen here so that there will be time for setup inside the ISR (interrupt service routine) after the signal FSYNC arrived. In this case, M1 and M0 should be set to 0 and 1 respectively on the EVB.

2.3 Method for Control Information Transfer

There are 2 methods to transfer control information for register read and writes across the digital interface, namely, using the FC pin or use the LSB of transmit data as a flag. To reduce the pin utilization for the SX, the later method is used.

3.0 Connections

To connect the Parallax SX demo board to the Si3032 EVB, we used the RB pins which are conveniently available on the header. The connection is made from RB header on the Parallax board to JP4 of the Si3032EVB using a ribbon cable.

The connections are:

RB4 ---> SDI

RB5 <--- SCLK

RB6 <--- FSYNC

RB7 <--- SDO

The result code of the key pressed is displayed on LED RB3-RB0 in a binary format.

4.0 Detailed Procedure in Pseudo Code

SX Initialization tasks:

- clear RAM
- set port B direction
- set interrupt on rising edge of FSYNC
- clear the random number in pending register
- enable interrupt

Si3032 initialization tasks:

- write 4 to register 7 (N1)
- write 71 to register 8 (M1)
- write 1 to register 10 (CGM)
- write 89 hex to register 9 (N2,M2)
- switch to normal operation by clearing bit 4 (power down line side) of register 6 (DAA control 20
- wait for ISOlink to be established by reading register 12 (line side status) and checking that bit 6 (Frame detect) is set

DTMF main loop:

wait for ring to come by reading register 5 (DAA control 1) and check that bit 2 (RDT, ring detect) is set

go off hook by setting bit 0 (OH, off-hook) of register 5 to 1

do the following in an infinite loop:

- toggle flag to indicate sampling low or high frequencies
- clear RAM bank for DTMF
- do this for 105 times: get a sample of new data in linear signed 16 bit format, do a DTMF detection (accumulate the sine and cosine accumulators) except when the sample is zero and update reference
- square and add sine and cosine accumulators
- get winner (highest and second highest results)
- encode the row and column of key pressed and display on RB3-0

5.0 Description of DTMF Detection Procedures

The get sample routine will truncate the 16 bit signed sample into 8 bit with saturation level.

The DTMF detection routine composes of doing a correlation of the input data with reference sine and cosine waves at the DTMF frequencies. The result are accumulated in the sine and cosine accumulators. To ease the multiplications, very small magnitude sine and cosine waves are used. By noting that cosine wave is just a sine wave with a 90 degree phase shift, the following sine wave is used: 0,1,1,2,2,2,1,1,0,-1,-1,-2,-2,-2,-1,-1,0,1,1... as reference.

Then the update reference routine will add the reference accumulators for 4 frequencies, either all the low frequencies or all the high ones. The value added corresponds to time that an 125 uS period occupies in a specific frequency, assuming that the period of the said frequency is 65536.

For example, at 697 Hz, the period is 1434.72 uS. So there are 1434.72/125=11.47776 occurrence of the 8 KHz data inside a 697 Hz cycle, each of which occupying 65536/11.47776=5709 or 164d hex.

To obtain the value of the waveform at that instant, the highest nibble of the MSB of the reference accumulator is used as an index to a sine table with 8 values (indexed by 0-7). The highest bit is used as a sign bit, making 8-15 negative values. This condition is checked first and the sign bit is masked so that only 0-7 is actually used to get the magnitude.

The correlation function is done by multiplying the sample with the reference waveform value. Since the magnitude of the reference is only 0,1 or 2, the multiplication is composed of doing nothing, returning itself or shifting left only. With both the sample truncated to 8 bit and the magnitude of reference being 8 bit, the result will be 16 bit and accumulated.

For cosine, 90 degrees of phase shift is needed. Since we are only concerned about the highest nibble, we can see that $8000 is the 180 degree point and therefore $4000 is the 90 degree point. So the phase shift is done by adding $40 to the MSB of the reference accumulator.

The square and add routine is needed since the results of computation is signed. A simple sum would cancel out the effect if we have corresponding positive and negative results in the sine and cosine accumulators. By squaring the accumulators and then adding them together, we will obtain the proper result.

After the square and add routine, it is a relatively simple procedure to determine the winner. By checking both the winner and the runner-up, we can ensure that there is sufficient range between them and detect a valid digit.

6.0 Results

The code was tested using a corded phone and the Silicon Lab Si3032 EVB attached to a Parallax SX demo board, both connected to a phone line simulator. All the DTMF digits are detected correctly.

7.0 Acknowledgement

The DTMF detection code was contributed by Chris Fogelklou, based on the DTMF mode.

Lit#: SXL-AN19-01

