anadigm

App Note - 315

Application Note: PIC32 Embedded Audio Sinewave Generator

Rev: 1.0.0

Date: 6th Jan 2015

TABLE OF CONTENTS

1 Purpose

The purpose of this document is to illustrate how easy it is to create embedded applications using the new range of Anadigm PIC32 development boards.

The document will describe in detail how to create an audio sinewave generator whose amplitude and frequency can be controlled using a pair of simple potentiometers. Amplitude can be controlled between +/-25mV and +/-3V and the frequency controlled between 20Hz and 20kHz using a logarithmic scale. Both amplitude and frequency are shown on an LCD display.

The board used for this application is the Anadigm DualApex PIC32 development board, but it can just as easily be created on the Anadigm QuadApex board.

2 The Hardware

2.1 AnadigmDesigner2 Circuit

Figure 1 below shows the audio sinewave generator circuit in AnadigmDesigern2 (AD2). It is very simple and consists of just an OscillatorSine CAM and a GainInv CAM.

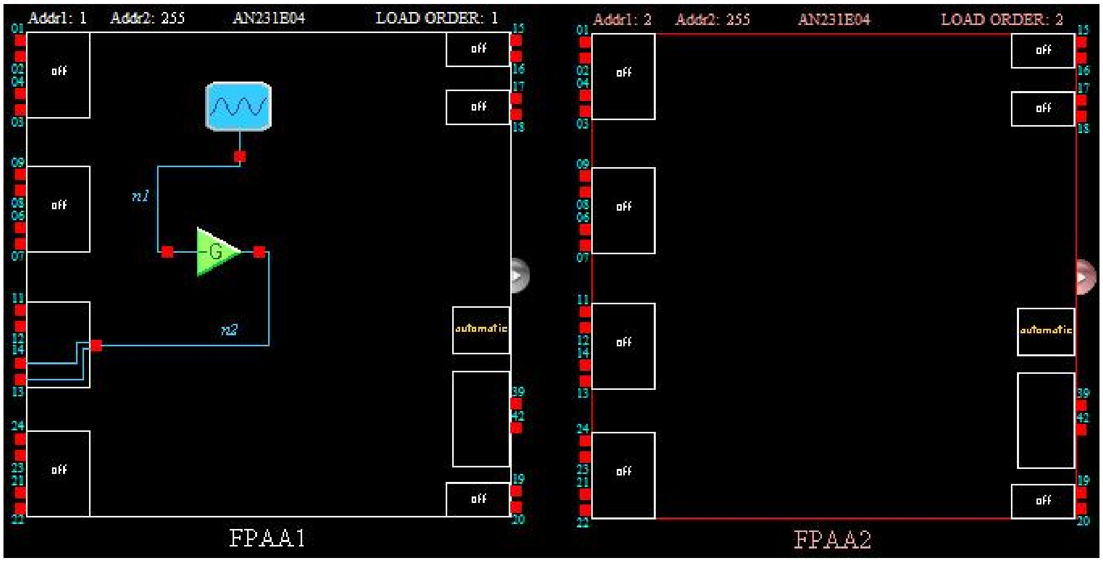
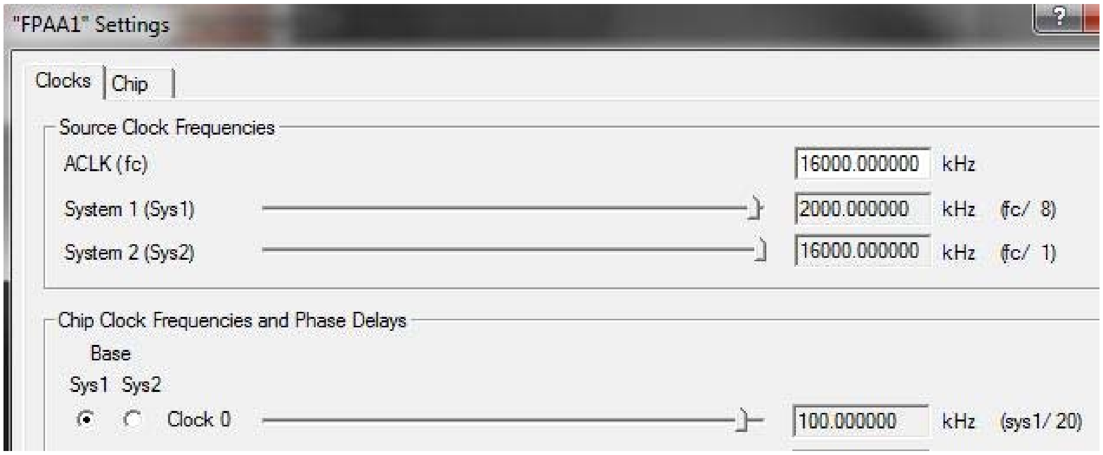


Figure 1: Sinewave generator circuit

The OscillatorSine CAM is used to generate a 2.5V fixed amplitude sine wave. The GainInv CAM is used to control the final amplitude of the sinewave. Although there is some control of amplitude in the OscillatorSine CAM, it is limited and so a GainInv CAM is used for this purpose.

NOTE: Figure 1 shows a second empty FPAA. This is because the DualApex board has 2 FPAAs, although it can be configured to use just one FPAA.

Figure 2 shows the clock settings for the circuit in figure 1.



"FPAA1" Settings

Clocks | Chip

Source Clock Frequencies

ACLK (fc) 16000.000000 kHz

System 1 (Sys1) 2000.000000 kHz (fc/ 8)

System 2 (Sys2) 16000.000000 kHz (fc/ 1)

Chip Clock Frequencies and Phase Delays

Base

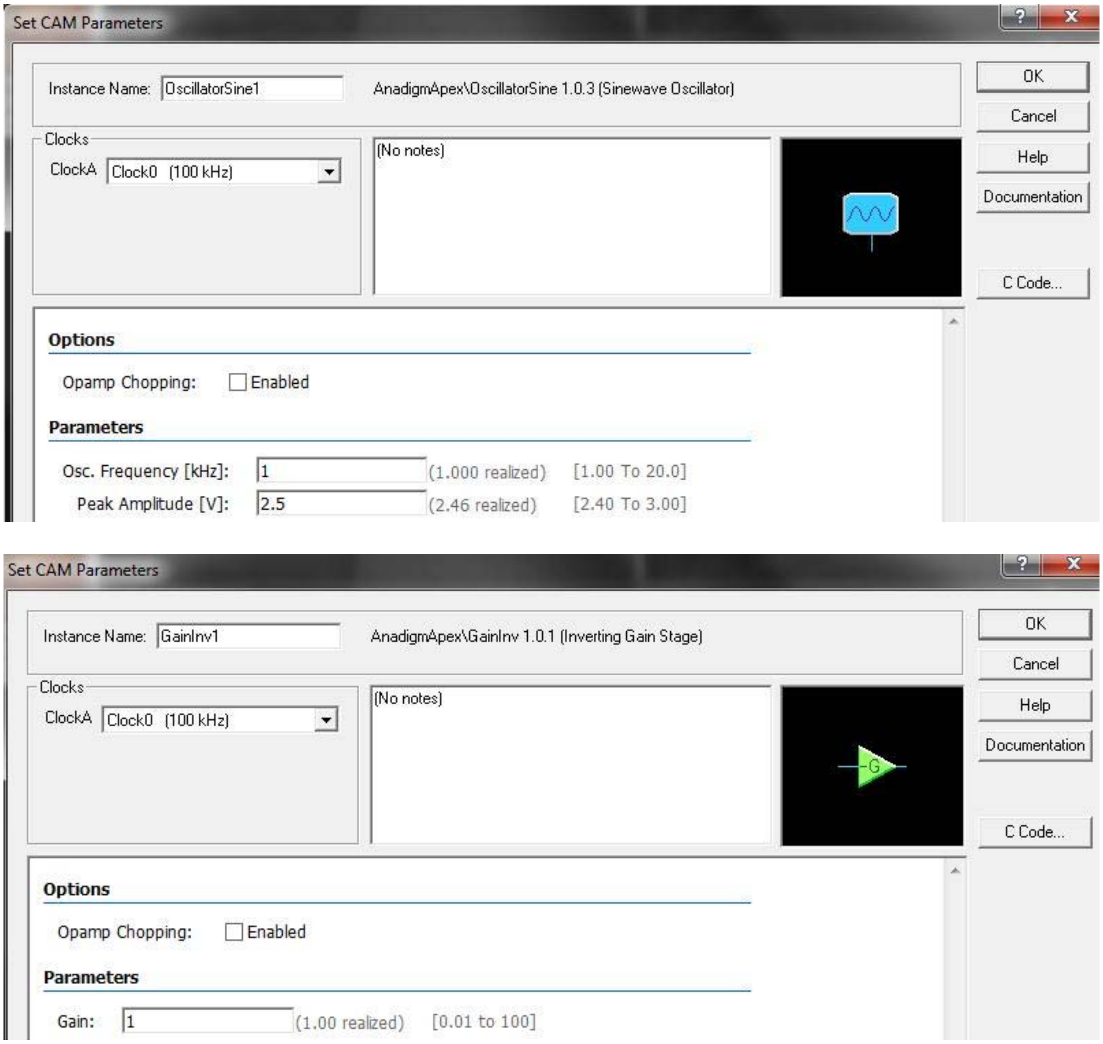
Sys1 Sys2

● ○ Clock 0 100.000000 kHz (sys1/ 20)

Figure 2: Clock settings

Both the OscillatorSine CAM and the GainInv CAM use a 100kHz clock in this circuit, although this is just the primary circuit. The application will vary the clock frequency as the sinewave frequency is varied in order to create the smoothest output waveform.

Figure 3 shows the parameter settings for the CAMs in the primary circuit. These settings are just the initial settings. The application will vary the frequency of the OscillatorSine CAM and the gain of the GainInv CAM.



Instance Name: OscillatorSine1 AnadigmApex\OscillatorSine 1.0.3 (Sinewave Oscillator)

Clocks

ClockA Clock0 (100 kHz)

(No notes)

Options

Opamp Chopping: □ Enabled

Parameters

Osc. Frequency [kHz]: 1 (1.000 realized) [1.00 To 20.0]

Peak Amplitude [V]: 2.5 (2.46 realized) [2.40 To 3.00]

Instance Name: GainInv1 AnadigmApex\GainInv 1.0.1 (Inverting Gain Stage)

Clocks

ClockA Clock0 (100 kHz)

(No notes)

Options

Opamp Chopping: □ Enabled

Parameters

Gain: 1 (1.00 realized) [0.01 to 100]

Figure 3: CAM parameter settings

2.2 The Board Set-up

Figure 4 shows how the Anadigm DualApex board should be modified. Two 10kΩ potentiometers have been added to the board and connected between +3.3V and ground. The wiper arms have been connected to ADC channel 0 for 1.

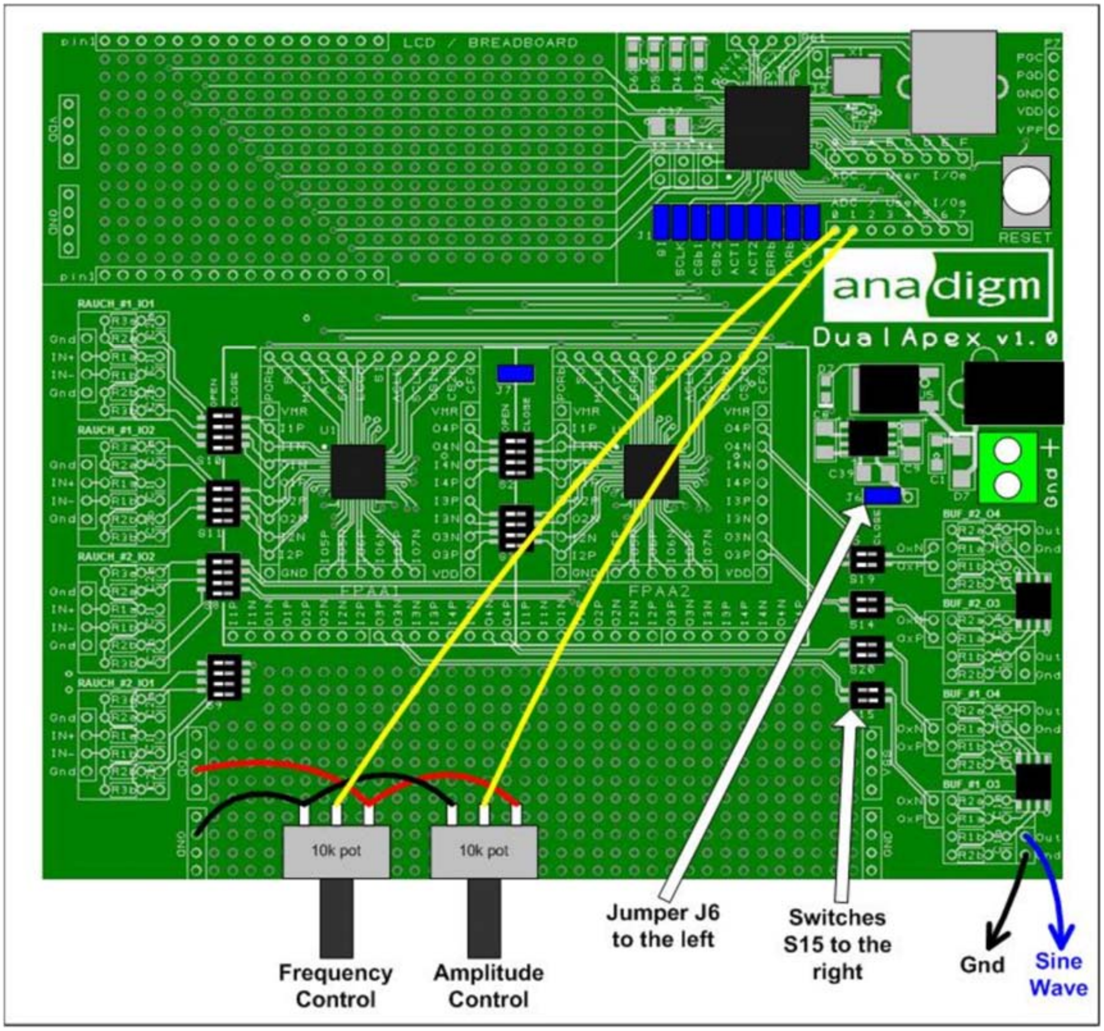


Figure 4: Board set-up

This application is going to use one of the output buffers provided on the DualApex board. The buffer to be used comes ready populated (with surface mount components on the underside) to provide a Low Pass filter with corner frequency of about 22kHz and a gain of x1. This buffer will convert the differential +1.5V referenced output signal from the FPAA to a single-ended ground referenced buffered output.

To connect the output buffer, the DIP switches labelled S15 should be pushed to the right to connect output #3 of FPAA #1 to the output buffer. Jumper J6 should be positioned to the left to enable the -3.3V supply for the output buffer. The hardware is now ready, now the PIC32 microcontroller must be programmed.

3 The Software

3.1 *IDE and Compiler*

The first step is to download the compiler and IDE. The compiler used for this project is the Microchip XC32 compiler (v1.34) which is free (with limited optimisation) and can be downloaded from the Microchip website:

http://www.microchip.com/pagehandler/en-us/devtools/mplabxc/home.html

This compiler can be run from the MPLAB IDE. The last version of this IDE is v8.92 and can be downloaded free from the Microchip website:

http://www.microchip.com/pagehandler/en-us/devtools/dev-tools-parts.html

Alternatively the more up to date MPLAB X IDE can be used. This can be downloaded free from the Microchip website:

http://www.microchip.com/pagehandler/en-us/family/mplabx/

See the Microchip website for details on how to install the IDE and incorporate the compiler into it.

3.2 *Creating the Project*

The next step is to generate the code for the CAM algorithms and API. To do this, open the AnadigmDesigner2 (AD2) circuit included with this application note (figure 1), and save it into the project directory. Now go to the Dynamic Config menu and select Algorithmic Method. Click on the Generation Options button. Select the Clocks tab and check the box next to FPAA1. This will ensure that code is generated for managing the CAM clock. Click on OK.

Click on CAM Functions and expand the tree on the left. Select GainInv1 and on the right check the box labelled “double setGainInv(double G)”. The other function is not required. Select OscillatorSine1 on the left and on the right check the box labelled “void setOscillatorSine(double Fo, double A)”. The other functions are not required. Click on OK.

Set the destination directory to the directory where the AD2 file is stored and where the code will be. Click on the Generate button. When the code has been generated, click on the Close button.

AD2 will have generated 4 files in the project directory called:

CAMCode.c
CAMCode.h
ApiCode.c
ApiCode.h

Make sure that the files SigGen32.c and SigGen32.h that were included with this application note are also in the project directory.

Now open MPLAB and go to the Project menu and select Project Wizard. Click on Next. Select PIC32MX440F512H from the Device drop down menu. Click Next. In the drop down menu labelled Active Toolsuite, select Microchip CX32 C-Compiler Toolsuite. Click Next. Browse to the project directory and enter a project name. Click Next. In the left window, expand the project directory to find the 4 AD2 generated files and the 2 project files that were included with this application note. Select these 6 files and click on Add >>. Click Next. Click Finish.

Now click on the Build All icon at the top, or go to the Project menu and select Build All. The project should compile successfully. There may be some warnings that mention Debug or Debugger. To remove these, go to the Project menu and select Build Configuration and then Release. Now compile again.

3.3 Programming the Board

Having compiled the project, a .hex file will have been created. This file now needs to be programmed into the PIC32 of the Anadigm DualApex board. There is a 5 pin programming header called P7 at the top right of the board. The pins are labelled:

PGC
PGD
GND
VDD
VPP

A microcontroller programmer such as the PICkit2 from Microchip can be connected directly to these pins and used to program the .hex file into the on-board PIC32 microcontroller. Once the controller is programmed, the application is ready to run.

3.4 Running the Application

Power up the board. A green LED will indicate that the board is powered. A second green LED next to the microcontroller will indicate that the FPAAs have been configured correctly. The LCD will now display Frequency and Amplitude. While monitoring the output at the bottom right of the board, turn the pots to see amplitude and frequency vary.

The amplitude can be varied between +/-25mV and +/-3V. The frequency can be varied between 20Hz and 20kHz.

NOTE: the software has been written in such a way that the frequency variation is logarithmic.

NOTE2: the amplitude may appear to fall slightly as the frequency approaches 20kHz. This is due to the output buffer on the board which is set up as a Low Pass filter with corner frequency of about 22kHz. See the board manual for instructions on how to change the corner frequency.

3.5 Changing the Application

The user is encouraged to try changing the AnadigmDesigner2 circuit to add more functionality, and to adapt the code accordingly. For help, please contact Anadigm Support.