Display Spectrum from ADC Quickstart Guide

IN THIS DOCUMENT

In this demonstration we use the following hardware and software:

- XP-SKC-U16 sliceKIT
- XA-SK-SCR480 Slice Card,
- XA-SK-SDRAM Slice Card,
- XA-SK-MIXED SIGNAL Slice Card,
- module_level_meter,
- module_fft_simple,
- module_display_controller,
- module_sdram,
- module_lcd,
- module_usb_tile_support,
- module_logging,
- module_xassert,

together to create a level-meter kind of spectral display on an LCD for an analog audio input. This application showcases the use of multichannel ADC in an xCORE-USB series XMOS device to sample the analog input, and real-time rendering and display of spectrum by taking short-time fourier transform.

1 Hardware Setup

The XP-SKC-U16 sliceKIT Core board has four slots with edge connectors: `DIAMOND`, `SQUARE`, `A` and `U`.

To setup up the system:

Publication Date: 2013/11/12 REV A

1. Connect XA-SK-SDRAM Slice Card to the XP-SKC-U16 sliceKIT Core board using the connector marked with SQUARE.
2. Connect XA-SK-SCR480 Slice Card with LCD to the XP-SKC-U16 sliceKIT Core board using the connector marked with DIAMOND.
3. Connect XA-SK-MIXED SIGNAL Slice Card to the XP-SKC-U16 sliceKIT Core board using the connector marked with A.
4. Give the two channels of audio input from a PC or a mobile to pins 1 and 2 of J2 on the mixed signal slice card using a suitable cable. The ground is connected to pin 4 of J3.
5. Connect the xTAG-2 to sliceKIT Core board.
6. Connect the xTAG-2 to host PC. Note that the USB cable is not provided with the sliceKIT starter kit.
7. Set the xCONNECT LINK to OFF on the sliceKIT Core board.
8. Ensure the jumper on the XA-SK-SCR480 is bridged if the back light is required.
9. Switch on the power supply to the sliceKIT Core board.

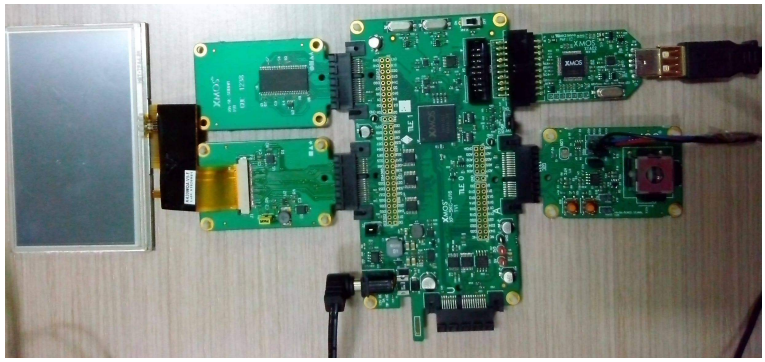

Figure 1: Hardware Setup for Display Spectrum from ADC Demo

2 Import and Build the Application

1. Open xTIMEcomposer and check that it is operating in online mode. Open the edit perspective (Window->Open Perspective->XMOS Edit).
2. Locate the Display Spectrum from ADC Demo item in the xSOFTip pane on the bottom left of the window and drag it into the Project Explorer window in the xTIMEcomposer. This will also cause the modules on which this application depends to be imported as well.
3. Include #define LCD_USE_32_BIT_DATA_PORT 1 in lcd.xc located in module_lcd.

4. Click on the app_display_spectrum_from_adc item in the Explorer pane then click on the build icon (hammer) in xTIMEcomposer. Check the console window to verify that the application has built successfully.

5. There will be quite a number of warnings that bidirectional buffered port not supported These can be safely ignored for this component.

For help in using xTIMEcomposer, try the xTIMEcomposer tutorial, which you can find by selecting Help->Tutorials from the xTIMEcomposer menu.

Note that the Developer Column in the xTIMEcomposer on the right hand side of your screen provides information on the xSOFTip components you are using.

3 Run the Application

Now that the application has been compiled, the next step is to run it on the sliceKIT Core Board using the tools to load the application over JTAG (via the xTAG-2) into the xCORE multicore microcontroller.

1. Select app_display_spectrum_from_adc project from the Project Explorer.

2. Click on the Run icon (the white arrow in the green circle).

3. At the Select Device dialog select XMOS xTAG-2 connect to L1[0..1] and click OK.

4. Play an audio in the PC or mobile. A test audio file containing a sine sweep from 20 to 20kHz is available in app_display_spectrum_from_adc/test_data.

5. The spectra of segments of mixed signal of two audio channels are displayed on LCD.

6. Try changing the volume of the audio input and notice the change in the height of spectral components.

4 Next Steps

Various parameters defined in app_display_spectrum_from_adc.xc are listed below. These can be adjusted if necessary.

1. SAMP_FREQ is the sampling frequency of the analog audio signal.

2. FFT_POINTS give the number of signal samples taken for FFT computation. FFT_SINE is defined accordingly.

3. LEV_METER_BANDS give the number of FFT points to be displayed.

4. LOG_SPEC if set to 1 computes log spectrum.

5. MAX_FFT sets the limit for the spectral values to be displayed. Values more than this limit are clipped.

6. `FFT_FULL_USE` if set puts the FFT computation to full use. If it is 0, then `FFT_UPDATE_RATE` determines the number of times FFT computation is done in a second.

The colors of the level-meter display of spectrum can be changed in `level_meter_conf.h`.