PmodLS1™ Reference Project

Revision: December 10, 2008

This document was produced by
Digilent Romania. For questions,
contact support@digilent.ro.

Overview

The PmodLS1 Reference Project can generate four sounds. Each sound corresponds to a musical note. When one of the infrared sensors detects something in front of it, the corresponding 10-bit digital sample is sent to a speaker through a PmodDA2 board and a PmodAmp1 board. If two or more sensors detect something at the same time, the 10-bit digital samples are added together and the resulting 12-bit sample is sent to the speakers.

The PmodLS1 Reference Project files are LS1RefProj.vhd, SoundGenerator.vhd, PmodDA2.vhd, and Cordic.vhd.

To run this reference project the following Digilent components are needed:

- an FPGA-based system board
- a PmodLS1 (Infrared Light Detector Module) board
- a PmodAmp1 (Speaker/Headphone Amplifier) board
- a PmodDA2 (Digital to Analog Converter) board
- four infrared reflective object sensors
- a speaker

Functional Description

This reference project generates sounds corresponding to the musical notes (C = 523Hz, E = 659Hz, G = 638Hz, C = 478Hz). The sine waves for each sound are generated using the Cordic algorithm. The four inputs from the PmodLS1 are used as validation signals for the four sounds. If an input is active, the corresponding sound is validated and added to the final sound output.

This reference project is composed of four blocks: the SoundGenerator module which instantiates the Cordic module, and the DA2 Controller component which is instantiated in the top module LS1RefProj.

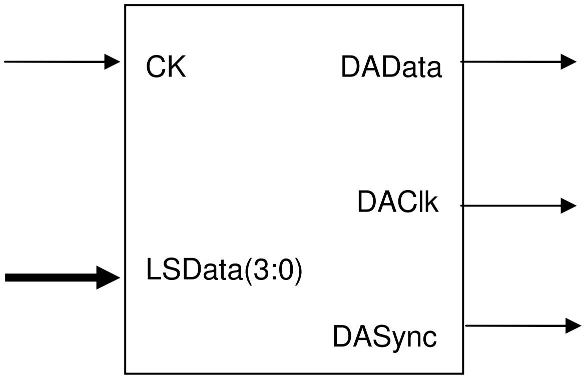

Figure 1 ***PmodLS1 Reference Project***

The LS1RefProj Module

The *ck*, *LSData(3:0)* signals are inputs for the top module and *DaData, DaClk, DaSync* are the output signals.

Port Definitions

ck input, global clock signal (50MHz)

LSData(3:0) input bus, four input lines coming from the PmodLS1 board. If one sensor detects something in front of it the corresponding bit will be ‘1’ else ‘0’.

DaData output, this signal is used for shifting data into PmodDA2 board.

DaClk output, this signal is a 25MHz clock used by the PmodDA2 board.

DaSync output, this signal is used to latch the data inside the PmodDA2 board after the data has been shifted out.

The Cordic Module

The Cordic module generates 10-bit digital samples of a sine wave, at a rate of 100 samples/second. The sine wave frequency is 100 times smaller than the input clock’s frequency.

It uses the Cordic algorithm:

```
x = x + y * tg(2 * pi / 100);
y = y - x * tg(2 * pi / 100);
  where tg(2 * pi / 100) = 1 / 16
```

This results in x = A * cos(2 * pi * Tck / 100) and y = A * sin(2 * pi * Tck / 100) where Tck is the process clock period and A is the amplitude, determined by the initial values of x and y (xi and yi):

A = sqr(xi * xi + yi * yi)

The division by 16 is performed by shifting the binary number four bits to the right.

Port Definitions

ck input, global clock signal (50MHz)

x(9:0) output bus, 10-bit value representing one sample from the sine wave

The SoundGenerator Module

The SoundGenerator module inputs the 50MHz clock and the *LSData(3:0)* bus. It generates 12-bit digital samples of a sine wave on the *SoundOut* output. This module instantiates the Cordic module. There are four counters which generate the four clocks. The values used by clock division to obtain the four musical frequencies are:

956 for C = 523Hz
758 for E = 659Hz
638 for G = 784Hz
478 for C = 1046Hz

After the clock division, the four clock signals *ckA, ckB, ckC, ckD* are used as inputs for the four instances of the Cordic modules. Each Cordic module will return a 10-bit intermediary sample of a sine wave. The 10-bit intermediary samples (*Sound_intA, Sound_intB, Sound_intC, Sound_intD*) will be added together into the 12-bit *Sound_int*. An intermediary sample is added to *Sound_int* if the corresponding bit in the *LSData(3:0)* is '1' (an obstacle is detected by the infrared sensor).

To add a 10-bit signed value to a 12-bit signed value, a sign extension is needed. The extension from 10 to 12 bits is made by extending the sign of the 10-bit value by copying the 10th bit on the 11th and 12th position. Before sending the *Sound_int* sample to the DA2 converter the most significant bit is complemented because the DA2 works with unsigned numbers.

Port Definitions

ck input, global clock signal (50MHz)

LSData(3:0) input bus, four input lines coming from the PmodLS1. If one sensor detects something in front of it, the corresponding bit will be '1' else '0'.

SoundOu output bus, 12-bit value representing the final sample (after adding the intermediary samples)

The DA2 Controller Component

The DA2_controller inputs the sample received from the SoundGenerator module. It generates the necessary signals and shifts the sample to the PmodDA2. Please refer to the *DA2 Reference Component* manual for more detailed information on the DA2_controller component.

Set-Up

The functionality of this project can be demonstrated using an FPGA system board which has at least two 6-pin connectors and a 50MHz clock.

To set up the hardware:

1. Make sure that the clock frequency select jumper (if there is one) on the board is set to the 50MHz position.
2. Connect the four infrared reflective object sensors to the ports S1, S2, S3, and S4 on the PmodLS1.
3. Connect port J1 on the PmodLS1 to one of the 6-pin connectors on the FPGA system board.
4. Connect port J1 on the PmodDA2 to one of the 6-pin connectors on the FPGA system board.
5. Connect port J1 on the PmodAmp1 to port J2 on the PmodDA2.
6. Connect the speaker to port J2 on the PmodAmp1.

To set up the software:

1. Create a new Xilinx ISE 9.2 project.
2. Place the VHDL files into the project.
3. Create an UCF file and make the following connections in it:
   ck connect to a 50MHz clock pin
   LSData(3:0) connect to the four data pins from the 6-pin connector in which PmodLS1 is connected
   DASync connect to the first pin of the 6-pin connector where the DA2 converter is plugged in
   DAData connect to the second pin of the 6-pin connector where the DA2 converter is plugged in
   DAClk connect to the forth pin of the 6-pin connector where the DA2 converter is plugged in
4. Synthesize the project and generate the programming file (*.bit).
5. Use Digilent's Adept software to program the resulting *.bit file into the FPGA. Please refer to the *Digilent Adept Reference Manual* for more detailed information.

.