ESE 382 Digital System Design Using VHDL and PLDs

Spring 19, Ken Short
April 27, 2019 8:49 am

Laboratory 11: Direct Digital Synthesis of Sine, Triangle, and Square Waves
This laboratory is to be performed the week starting April 28th. Task signatures without penalty the week of May 5th.

Prerequisite Reading
1. Chapter 10 of the text.
2. Section 15.9 of the text.
3. MT-085 Tutorial: Fundamentals of Direct Digital Synthesis (DDS).

Purpose
Your DDS system from Laboratory 10 only generates sine waves. The functionality of that system is to be extended to also generate triangle and square waves. To accomplish this, additional entities have to be added to those used in Laboratory 10. Entities from Laboratory 10, with the exception of the top-level entity, must not be redefined or modified, but used, as is, with the new entities.

A major difference in this design, from previous designs, is that the new entities are not specified by the instructor. Instead, you have to define new entities necessary to expand Laboratory 10's architecture to produce the additional types of waveforms.

To be successful in this design, you first must thoroughly understand how your system from Laboratory 10 works. Then you must spend the bulk of your effort thinking through how that architecture is best modified and expanded to add the required features. Take your top-level block diagram from Laboratory 10 and think of what functional blocks need to be added to achieve an optimal architecture.

Draw the new blocks out on your old block diagram and further refine what those blocks must do. Review your new block diagram and determine if there is a simpler way to structure the architecture.

If you end up with two or more blocks that have essentially the same functionally, consider whether you can parameterize a single entity and instantiate it multiple times to replace the similar blocks. Any new entities that you have defined that can be parameterized using generics must be so defined.

When you are satisfied that you have the best approach, write entity declarations for the new entities. From these entity declarations write a natural language description of the functionality of each new entity. Finally, for each new entity decide what is the best way to code the required functionality in VHDL and write the code.

An important point of the previous paragraphs is that you should not start writing code before you have thoroughly thought through your approach. Otherwise, you may prematurely commit yourself to a less than optimal approach, because you have already spent time writing code. You may end up spending even more time trying to patch code for a bad approach into an inefficient but workable system.

Design Tasks

Design Task 1: Addition of Triangle and Square Wave Output Capability

The entity declaration for the top level of the new system adds two inputs `ws1` and `ws0`. These inputs select the type of waveform that is output. The entity declaration for the top-level is:

```
entity dds_sts is
    generic (a : positive := 14; m : positive := 7);
     port(
         clk : in std_logic;-- system clock
         reset_bar : in std_logic;-- asynchronous reset
         freq_val : in std_logic_vector(a - 1 downto 0);-- selects frequency
         load_freq : in std_logic;-- pulse to load a new frequency selection
         ws1 : in std_logic;-- system clock
         ws0 : in std_logic;-- system clock
         dac_value : out std_logic_vector(m - 1 downto 0);-- output to DAC
         pos_sine : out std_logic-- positive half of sine wave cycle
         );
end dds_sts;
```

When output selected by `ws1` and `ws0` is as shown in the following table:

Table 1: Output Waveform Selection

ws1, ws0	Output
00	undefined
01	sine wave
10	triangle wave
11	square wave

The frequency remains the same when switching from one type of waveform to another and is determined as it was in Laboratory 10. For all of the output waveform types the peak-to-peak amplitude of the output waveform is the full range of the DAC.

Write a design description for the system and a non self-testing testbench to verity the system using a 1 MHz system clock.

Submit your design description, Code2Graphics top-level block diagram, and simulation output waveform(s) as part of your prelab.

Laboratory Tasks

Lab Task1: Edge Detector and Frequency Register Verification
Using Aldec Active-HDL create a new workspace named `dds_sts`. In this workspace create a design named `dds_sts`. Import all your VHDL source files and your top level testbench. Compile and simulate your design.

Using your block diagram, explain to a TA how your system operates and what the added entities do. Show the TA the relevant portions of the output waveforms and the relevant signal values to prove that these entities and your overall system perform properly.

Obtain the TA's signature and evaluation of the extent to which your design operates properly.

Lab Task 2: DDS System with Sine, Triangle, and Square Output Waveforms
Use Synplify to synthesize your design for a Lattice LCMXO3L-6900C FPGA target. Use the pin assignments provided to you in the laboratory. After synthesis is complete, from the **HDL Analyst** menu select **RTL,** then **Hierarchical View**. Print this diagram.

Use Lattice Diamond to fit the synthesized logic into a Lattice LCMXO3L-6900C FPGA. In Lattice Diamond, view the chip report.

Program an LCMXO3L-6900C FPGA and verify its functional performance in the hardware test system provided.

Have a TA sign off whether your programmed FPGA met the design specifications.