A Memory Optimized Design and Simulation for Numerical Controlled Oscillator

Jun Han, Shu-Fang Zhang, Bao-Yin Cao, Li-Jun Huang

School of Mechanical and Electronic Engineering, Suzhou University, Suzhou, 234000

Abstract

The paper is aiming at getting numerical control, high degree of accuracy and higher carrier signal in the digital communication system. We have developed a kind of Numerical Controlled Oscillator system based FPGA. This system has high resolution, high-speed frequency conversion, and lower phase noise and so on. This thesis analyzed principle about Numerical Controlled Oscillator, gives the whole system, and describes design method of every Composition Module. In the last, describe the simulation result and verify the correctness of design.

© 2011 Published by Elsevier Ltd. Selection and/or peer-review under responsibility of [name organizer]

Key words: Numerical Controlled Oscillator; FPGA; Phase Accumulator; Memory Optimized; Simulation

1. Introduction

Accuracy and speed of data transmission have become increasingly demanding in digital communications technology development trend. So in order to achieve high-speed digital communication systems, the problem that must be resolved is how to get controllable, high precision, high-frequency carrier signal - more specifically, to produce an ideal sine or cosine wave samples, which have variable frequency, Numerical Controlled Oscillator is the essential unit of the digital communication system [1].

In the past, some well-known chip manufacturers have also launched the production of advanced high-performance CMOS process and multi-function ASIC, using a specific process, the internal digital signal jitter is very small; the output signal has high quality. However, in some cases, due to the immobility control mode of ASIC, so the working methods, frequency control and other aspects of have a great gap
between the requirements of the system [2], then using high-performance FPGA devices designed to meet their needs DDS circuit (where the main part of the FPGA implementation to complete NCO) is a good solution, its re-configurable structure can easily achieve a variety of complex modulation function, with good usability and flexibility.

First, start from the actual development, top-level structure diagram of controlled oscillator is given. Second, it writes a simulation program based on a detailed analysis of their works, the organizational structure of the simulation test file is also given, and verification environment is described. Finally, the use of Debussy simulation is analyzed to verify the correctness of the design.

2. The Overall Design of Numerical Controlled Oscillator

2.1. Basic Principle

The main components of NCO include the frequency control word register, the phase control word register, accumulator, adder and look-up table formed by on-chip ROM.

NCO implementation usually has two ways: (1) look-up table [3]; (2) CORDIC (coordinate rotation digital computer) algorithm [4, 5]. This article uses the traditional design methods - look-up table that pre-phase sine wave oscillator according to each sine of the phase calculate the value of sine, and phase angle as the address to the sine of the phase data is stored in the table; the phase accumulator accumulate input frequency control word constantly by the system clock controlled, get word to the frequency of the digital phase step, and then add the module through the initial phase of the phase shift, to get to the output of the current phase, the value as a sample address, look-up table to obtain sine and cosine signal samples.

In order to save storage space of ROM, the adder output as input to the preliminary sampling phase of the address register, ROM memory stores only a 1/4 wave data, so a small amount of control logic is replaced with a 3/4 of storage space, Therefore, we must address the sampling phase of the conversion register, look-up table to obtain cosine signal samples, getting all the first quadrant of the waveform, so it needs to be processed on the value of symbols, and finally sine/cosine waves are controlled by counter and output register output delay to get the correct sine/cosine wave.

2.2. Design and Analysis of FPGA

Table 1 sine Address Translation

| ph_addr[13:12] | ph_addr | addr0 | Sine   |
|--------------|---------|-------|--------|
| (0~1/2)π     | 00_0000_0000_0000 | ph_addr[11:0] | 0->4095 |
|              | 00_1111_1111_1111 |       |        |
| (1/2~1)π     | 01_0000_0000_0000 | ~ph_addr[11:0] | 4095->0 |
|              | 01_1111_1111_1111 |       |        |
| (1~3/2)π     | 10_0000_0000_0000 | ph_addr[11:0] | 0->4095 |
|              | 10_1111_1111_1111 |       |        |
| (3/2~2)π     | 11_0000_0000_0000 | ~ph_addr[11:0] | 4095->0 |
|              | 11_1111_1111_1111 |       |        |

From Fig. 1, for specific design applications, the frequency of Numerical controlled oscillator module \( f_T = 40 \text{ MHz} \) (i.e., the period \( T = 25\text{ns} \)), the size of lookup table defined as \( 4^k \times 16\text{bit} \) (you can store 4096 samples of sine values). Frequency control word \( \text{fw}_\text{ctrl} \) mainly control the interval of reading sample values or in 1/4 cycles to read the number of sample values, the formula is as follows:
\( x \times \text{fw}_\text{ctrl} = 4096 \)  \hspace{1cm} (1)

Where \( x \) represents 1/4 cycles to read the number of waveform sample values; \( \text{fw}_\text{ctrl} \) for the frequency control word, 4096 represents the total number of sample values stored in lookup table.

From the formula (1) we can get the cycle \( T_s \) of sine/cosine output waveform:

\[ T_s = x \times T \times 4 \]  \hspace{1cm} (2)

Therefore:

\[ \text{Fout} = 4096 \times 4 \times \text{fclk} / \text{fw}_\text{ctrl} \]  \hspace{1cm} (3)

Where, \( \text{Fclk} \) for the module in which the operating frequency, \( \text{Fout} \) for the output waveform frequency, 4096 for the length of ROM memory, 4096 * 4-bit for the cumulative maximum of accumulator data wide (214 = 16384).

As the designing is that the first 1/4 cycle of data value of sine wave is stored in the ROM (sine wave cycle \( T = 1 / 4096 \times \pi / 2 \)), corresponding to Fig. 1 is stored in the waveform within the range of 0 values. Therefore, in order to output full sine/cosine waveform, through the highest two-phase control registers address translation in order to compare the waveform of the segment to achieve a complete waveform of the output. Specific sine wave output of the address translation mode as shown in Table 1, corresponding to the sine wave shown in Fig. 1; specific cosine wave output of the address conversion and the corresponding cosine wave is omitted.

Fig. 1 Sine Wave

2.3. Core Design of ROM IP

Approved the use of Altera's IP system tools “IP Tool bench”, we design dual-port input/output ROM of the system [6-7], the specific structure shown in Fig. 2.

Fig. 2 Core Structure of ROM IP

3. Design of Simulation Program and Result Analysis

The design of Simulating environment is shown in Fig. 3 (a) shows, NCO is the design of the test module, tb_nco.v file contains all of the verification environment, defines the clock and reset signals. Fig.
3 (b) defines the detail structure of file directory, including: TESTBENCH directory (tb_nco.v), FILELIST catalog, RUN directory (run.bat), LOG directory, WAVE directory (tb_nco.fsdb) and the file COVERAGE.

Debussy is developed by NOVAS Software, Inc (Siyuan Technology) as HDL Debug & Analysis Tool, among HDL source code, schematic diagram, waveform, state bubble diagram, it can not only do the real-time tracking to assist engineers troubleshooting, but also can be used to simulate or see secondary wave. Debussy has four main elements: nTrace, nWave, nSchema, nState, it does not contain the simulator, therefore external simulator ModelSim must be called to produce FSDB file, the displaying unit nWave display waveform by reading the FSDB file, the principle structure is shown in Fig. 4. Here with the combination of ModelSim and Debussy, we use two ways: command scripts and interactive user interface to achieve simulation.
1) Fig. 3 (b) shows the simulation test file directory to write HDL file list “nco.f”.
2) Writing ModelSim command line script file "sim.do".
   ```
   sim.do:
   01 vlib work
   02 vlog-f nco.f
   03 vsim work.divider_tb
   04 run 10us
   05 q
   ```
3) Writing a batch scripting file "run.bat", calling the command line ModelSim to generate waveform file.
   ```
   run.bat:
   01:: echo off
   02 @ ECHO OFF
   03:: Setting the software path
   04 SET debussy = C:\Novas\...\Debussy.exe
   05 SET vsim = C:\...\vsim.exe
   06:: ModelSim Command
   07% vsim%-c-do sim.do
   08:: Delete documents generated ModelSim
   09 RD work/s/q
   10 DEL transcript vsim.wlf/q
   11:: Debussy Command
   12% debussy%-f nco.f-ssf tb_nco.fsdb
   13:: delete wave files
   14 DEL tb_nco.fsdb/q
   15:: remove generated documentation Debussy
   16 RD Debussy.exe Log/s/q
   17 DEL debussy.rc/q
   18:: Exit the command line
   19 EXIT
   ```
4) Double click “ run.bat”, pop-up Debussy and Debussy waveform to view component.
5) Use command "Signal" -> "Get Signals", Add signal the command Sin_out, Cos_out, get the waveform shown in Fig. 5.

   1) Implement "% Debussy &" command to start the Debussy, then establish "Log" directory;
   2) Implement "File\Import Design" command to import programs: “tb_nco.v” written in the Quartus II;
   3) Implement "Tool\Options\Preferences" command, setting the external simulator produces FSDB file;
   4) Perform "Tools\Interactive Mode" to switch to the Interactive Mode, after start the simulation work, produce the resulting documents “tb_nco.fsdb”;
   5) Use the "nWave" unit to load tb_nco.fsdb file, and use the command "Signa -> Get Signals" to add observe signal: Sin_out and Cos_out. As long as the control word to modify in file “tb_nco.v”, can be obtained waveform as shown in Fig. 5.

   Fig. 6 shows the corresponding simulation waveforms. There are two waveforms in Fig. 6(a-e). The top line is Sin_out waveform, and the underside is Cos_out waveform.
By observant simulation results fully consistent with the theory calculation, and output accords with the standards of filter. Verification of test is eligible.

However, in testing the code coverage failed to achieve require: branch is not up to 100%. Because the “case” sentence in the code contain all characters, the “default” statement executes with an error occurred, but not affects the performance of filter.

4. Conclusions

In this paper, starting from the actual project requirements of numerical oscillator, we use FPGA to achieve system design from analyzing basic principles of numerical controlled oscillator. In this Design we specially design sampling address translation of phase register, it only need to store 1/4 wave data in ROM, so it saves storage space. During the test, first we write test programs in the Quartus II environment, and then with a combination of ModelSim and Debussy, we use two ways: command scripts and interactive user interface for simulation, we verify the correctness of the design and achieve the design requirements. It can provide numerical controllable, high precision, high-frequency carrier signal for high-speed digital communication systems.
Acknowledgment

The work is supported by the Nature Science Research Project of Suzhou University (Anhui Province) under Grant No. 2009yzk15, 2011cxy02, by the Nature Teaching Research Project of Suzhou University (Anhui Province) under Grant No. szxyjy200813 and Technology Planning Project of Suzhou city under Grant No. 2008-22. The authors would like to thank the School of Computer and Information, Hefei University of Technology. The authors also would like to thank Mr. Guo-qing Yan of SuZhou University and Mr. Yan-yan Liu of Hefei University of Technology.

References

[1] Yuan Lin, Xu Kai, Feng Yan-qing. Design of Numerical Control Oscillator Based on FPGA. Ship Electronic Engineering, 2010, Vol.30 No.4, pp.188-190.
[2] Weng Ping, Wang Shao-yun. Design of Numerical Controlled Oscillator Based on FPGA. Informatization Research, 2010(09).
[3] YANG Jian-ping. Implementation of numerical controlled oscillator based on look-up table and XC2V1000. Automation & Instrumentation, 2010(01), pp.126-127.
[4] Zhang Qing-lin, Luo Yi-jun, Chen Shu-zhen, etc. Design and implementation of NCO based on phase rotation. Systems Engineering and Electronics, 2010, (5), pp.908-911.
[5] Chen Wan-pei, Huang Ying, Shao Ning, etc. Improved NCO realization technology based on CORDIC arithmetic and performance analysis. Journal of Yangzhou University (Natural Science Edition), 2010, (2), pp.52-56.
[6] Zhao Bing, Xu Jia-pin. A Digital IF Receiver based on Optimized IP Core. Communications Technology, 2010, (11), pp.6-8.
[7] Wang Li-jing, Zhang Jian-yan, Chen Guo-tong, etc. Application of MATLAB and QUARTUS II in 16QAM debugging. Journal of Hebei University of Science and Technology, 2010, (2), pp.104-107.
[8] Oppenheim A V, Schafer R W. Discrete-Time Signal Processing. Published by Prentice-Hall, Inc. 2000.