Simulation and Synthesis of Various Modules of Satellite Modem using Synopsys Flow

Hepi R Patel
M.E student
Department of Electronics and Communication Engineering
GTU PG School, Gandhinagar-382424, Gujarat, India

Abstract— There are various satellite ground terminals needs to be designed with reduced weight, volume, power and manufacturing cost. Modem is one of the major constitute of satellite terminal. To achieve high form factor without compromising performance a single chip solution for digital modem is developed. With the growing complexity of VLSI circuitry, the ability to perform both the time and cost effective design and test of IC devices has become increasingly difficult. Design Compiler is EDA tool used for synthesis of the design. Design Compiler reads design file in VHDL/verilog format & translate into gate level netlist using technology library. Functional simulation of modem is performed in VCS. The basics of VCS and Design Compiler is discussed in this paper. Simulation results of NCO which is a module of satellite modem using Synopsys VCS G-2012.09-SP1. Synthesis reports and gate level netlist has been generated of the same using synopsys design compiler G-2012.06-SP5-1. Other modules like Scrambler, Differential Encoder, Modulator, Demodulator, Descrambler, Differential Decoder, Viterbi Decoder are also simulated and synthesized using VCS and Design Compiler.

Key words: Satellite Modem, EDA

I. INTRODUCTION

A satellite modem is used to transfer data using satellite communication. A modem is a device functioning both modulation and demodulation. The main function of modem is to transform a data or input bit stream to a radio link and vice versa. A modulator transforms a bit stream to radio link and demodulator recover symbol and carrier frequency from modulated data. VCS and Design Compiler are EDA tools from synopsys which are used for simulation and synthesis of the design respectively. The output of baseband processor feed to the scrambler input. Scrambled data will be differential encoded by differential encoder followed by FEC Encoder. Modulator generates 12 bit I & Q output, which is sampled by NCO. There are various modulation schemes i.e. BPSK, QPSK, 8-PSK. In this modem ASIC there is a facility to select any modulation scheme. Modulated output is fed to the demodulator part. This demodulated output is fed to the RRC filter , Diff. Decoder, Viterbi Decoder and Descrambler to generate the output. In this modem ASIC there is a facility to either enable or disable Scrambler, Descrambler, RRC Filter, Diff. Encoder, Diff. Decoder, FEC Encoder and Viterbi Decoder modules according to our requirements. In this paper only output of NCO modules are presented.

II. VCS

VCS is compiled code simulator. It analyze, elaborate and simulate VHDL, Verilog, mixed HDL, system Verilog, OpenVera and SystemC design descriptions [6]. VCS MX uses the synopsys_sim.setup file to configure its environment for VHDL and mixed HDL designs. This file maps the VHDL design library names to specific host directories, sets search paths, and assigns values to simulation control variables.

When we invoke VCS MX, it looks for the synopsys_sim.setup files in the following three directories with the same order:

- Master setup directory

The synopsys_sim.setup file in the $VCS_HOME/bin directory contains default settings for the entire installation. VCS MX reads this file first.

/usr/synopsys/bin/setup

- Home directory

VCS MX reads the setup file in our home directory second, if present. The settings in this file take precedence over the conflicting settings in our synopsys_sim.setup file in the master setup directory, and carry over the rest.

/home/sunilkumar/hepi

- Run directory

VCS MX reads the setup file in our design directory last. The settings in this file take precedence over the conflicting settings in synopsys_sim.setup file in the master setup directory, and the synopsys_sim.setup file in our home directory, and will carry over the rest. We can use this file to customize the environment for a particular design.

/home/sunilkumar/hepi/Modem_ASIC

VCS uses the following basic three steps to compile, elaborate and simulate any Verilog, VHDL, and mixed HDL designs:

- Analysis the Design
- Elaborating the Design
- Simulating the Design
- Analysis the Design

Analysis is the first step to simulate the design. VCS provides vhdlan and vlogan executables to analyze VHDL and Verilog design code. vhdlan/vlogan analyzes the design and stores the intermediate files in the design or work library.

- Elaborating the Design

Elaborating is the second step to simulate the design. VCS provides the vcs executable to elaborate the design. This executable elaborates the design using the intermediate files in the design or work library, generates the object code and links them to generate a binary simulation executable, simv.

- Simulating the Design

Simulate the design by executing the binary simulation executable, simv. We can use simv to run the simulation.

III. DESIGN COMPILER

. Design Compiler optimizes designs to provide the smallest and fastest logical representation of a given function. It comprises tools that synthesize the HDL designs into optimized technology-dependent, gate level designs [5]. It supports a wide range of flat and hierarchical design styles

and can optimize both combinational and sequential designs for speed, area and power. All the modules of modem are synthesized by Design compiler.

Synthesis is the process that generates a gate-level netlist for an IC design that has been defined using an HDL. Synthesis includes reading the HDL source code and optimizing the design from that description.

Optimization is the step in the synthesis process that attempts to implement a combination of library cells that best meet the functional, timing, and area requirements of the design.

Compile is the Design Compiler command and process that executes the optimization step. After reading in the design and perform other necessary tasks, we invoke the compile command to generate a gate level netlist for the design.

IV. SIMULATION AND SYNTHESIS RESULTS

Simulation result of NCO module is shown in the fig-1. Gate level netlist is shown in the fig-2. Area report is shown in the Fig-3. All the modules of satellite modem has been simulated using VCS. Table-1 shows the area, power,and cell requirements of all the modules.

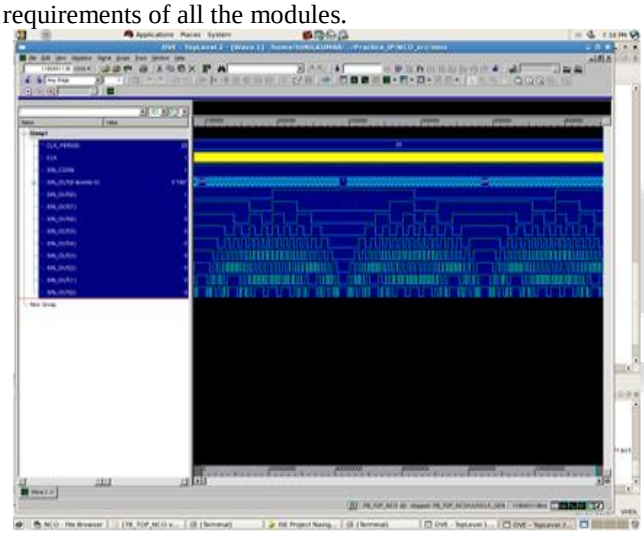

Fig. 1: Simulation result of NCO

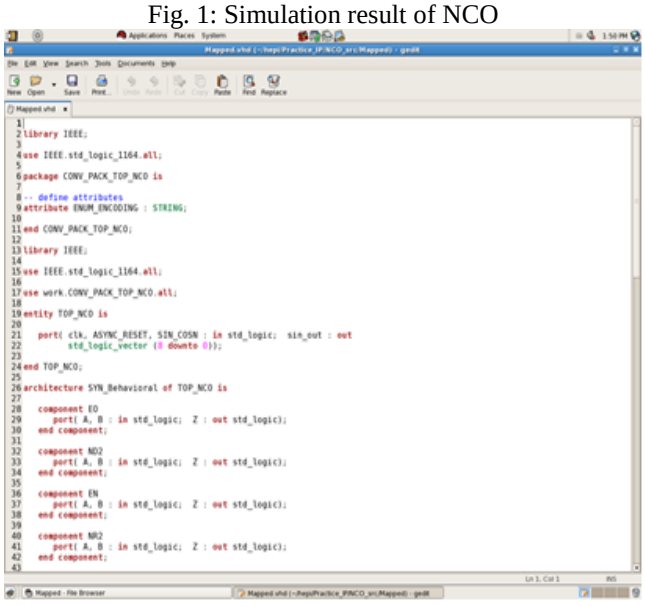

Fig. 2: Gate level Netlist of NCO

Fig. 3: Area report of NCO

Module	Area(nm^2)	No. of Cells	Power(nW)
RRC Filter	7008.00	1354	$3.7481x10^4$
Differential Enocder	17.00	5	178.5183
Differential Decoder	28.00	7	202.9397
NCO	144.00	32	$3.17x10^3$
Scrambler	96.00	32	782.7095
Modulator	31.00	22	394.0067
Demodulator	31.00	22	394.0067
Viterbi Decoder	23795.00	8637	$6.96x10^5$
Microcontroller	14914.00	1944	$5.15x10^4$

Table 1: Synthesis Summary

V. CONCLUSION AND FUTURE WORK

All the modules of satellite modem are follow all the DRC and optimized constraints. Here, for all the modules optimize constraint is 100 MHz clock. In the area report there is no any area and clock violation occurs. From the simulation result I can conclude that all the design of modules are working according to their functionality. Here I have simulated and synthesized microcontroller module also and it is successfully worked, So in future if we want to configure modem using UART, SPI interface can be done using microcontroller. This generated gate level netlist can be applied to the physical design flow of ASIC flow for place and route, CTS ,and so on to manufacture an ASIC.

REFERENCES

[1] Zaifu Zhang, Richard Wieler, Guy Jonatschick and Hart Poskar, "Synthesizing Testability Features into a Design With the Synopsys Test Compiler", 95CH3581-6/0-7803-2741-1/95, 1995, IEEE.

[2] V. Kripa Sundar, Ashish V. Nayak, Debashis Roy Chowdhury, "Incremental Compilation in the VCS Environment", Synopsys ,Inc.
[3] John C. Porcello "Designing and Implementing Low Density Parity Check(LDPC) Decoders using FPGAs, IEEE
[4] J.Manikandan, M.Manikandan, "Performance Analysis Of Various FEC Codes for Wireless Communication", July 8,2014, IEEE.
[5] Design Compiler User Guide, Version-2010.03-SP2, June 2010.
[6] VCS user Guide,G-2012.09-SP1, March 2013.
[7] www.eecs.utexas.edu/labs/synopsys/synopsys_basic_steps.pdf
[8] http://users.ece.utexas.edu/handouts/vcs.pdf
[9] Vinaya Skanda, Sine wave generator using numerically controlled oscillator module, Microchip Technology Inc.

