Comparing HLS with VHDL at the example of an FM Radio Receiver

Michael Wurm

MASTERARBEIT

eingereicht am

Fachhochschul-Masterstudiengang

Embedded Systems Design

in Hagenberg

im Juli 2021

Advisor:

DI (FH) Dr. Florian Eibensteiner

© Copyright 2021 Michael Wurm

This work is published under the conditions of the Creative Commons License *Attribution-NonCommercial-NoDerivatives 4.0 International* (CC BY-NC-ND 4.0)—see https://creativecommons.org/licenses/by-nc-nd/4.0/.

Declaration

I hereby declare and confirm that this thesis is entirely the result of my own original work. Where other sources of information have been used, they have been indicated as such and properly acknowledged. I further declare that this or similar work has not been submitted for credit elsewhere. This printed copy is identical to the submitted electronic version.

Hagenberg, July 15, 2021

Michael Wurm

Contents

Preface

Abstract

This should be a 1-page (maximum) summary of your work in English.

Kurzfassung

An dieser Stelle steht eine Zusammenfassung der Arbeit, Umfang max. 1 Seite. ...

Chapter 1

Introduction

1.1 Motivation

FM is a common RF signal that's available everywhere.
The demodulated signal is an audio signal, that can be listened to. This is subjectively more attractive than a generic data stream.

1.2 Broadcast Radio

Evolution from AM to FM because of certain advantages, etc.
Explain FM usage/existence nowadays (geographical; frequencies; devices; new standard DAB, etc)

1.3 Goal

put all knowledge that was accumulated during studies (HSD, ESD, mention subjects such as DSP) together in a project.
practical usage of DSP in an FPGA
develop an FM receiver and explain it, so that it can be followed in a tutorial with an affordable budget in hardware.
show multiple ways of how to implement the same thing, in different levels of abstraction. Weigh each ways' field of application, efficiency and applicability.

Thesis Design Decisions:

Main focus on system design.

How to achieve the same result in 3 (4?) different levels of abstraction?

1. GnuRadio (?)
2. Matlab/Python
3. HLS
   Implementation targeting Xilinx' ZedBoard.
   Optionally:

Intel as a comparison (synth only, no HW) - “How platform-independent is HLS?”

4. VHDL

Compare: - Implementation effort (difficulty) - Implementation time - Usefulness on Target Systems (uC, SoC, RasPi, etc) - others?

Chapter 2

Signal Processing Theory

2.1 Modulation from Baseband to RF

2.2 Modulation from RF to Baseband

2.3 Frequency Modulation in Broadcasting

Frequency modulation (FM) is a widely used standard to transmit data streams. The probably best known usecase therefore is commercial broadcast radio, where an audio stream is transmitted. Devices to receive these streams are available for low prices to the public.

This chaper describes the main properties of broadcast FM, such as the mathematical description, frequency bands that are used, or the specific frequency parts within a channel spectrum.

2.3.1 Mathematical description

The mathematical description of an FM baseband signal can be described with the formulas presented in this section. FM encodes the information content in its instantanious frequency. This means that the measured frequency at any moment in time represents a specific value of a transmitted message. For a general classification, FM belongs to the group of angle, or phase modulated signals (PM). The simple reason therefor is, that a frequency has a direct relationship to an angle, if the signal is seen on a unit circle.

The instantanious frequency f_i of an FM signal can be described as

$$f_i = f_c + \Delta f \cdot m(t) \tag{2.1}$$

where f_c is the carrier- or center frequency, Δf is the maximum frequency deviation and $m(t)$ is the information or message signal that is to be transmitted. Simply looking at this equation, the frequency deviation, which is sometimes also called the swing, varies in the range between $f_c \pm \Delta f$.

Considering the relationship between frequency and angle, as described above, and after some simple substitutions which will not be described into detail here, the equation for a generic frequency modulated wave is the following

$$s(t) = A_c\ cos\Big(2\pi f_c t + 2\pi k_f \int m(t)dt\Big) \tag{2.2}$$

where A_c is the amplitude of the resulting FM signal and k_f is the frequency sensitivity factor. This sensitivity factor has a direct relationship with the modulation index β.

$$\beta = \frac{\Delta f}{f_m} = \frac{k_f A_m}{f_m} \tag{2.3}$$

where f_m is the highest existing frequency, or the bandwidth of the information signal.

Formula 2.2 describes a frequency modulated signal with a generic message signal $m(t)$. A widely used example application of FM is broadcast radio, where audio streams are transmitted. An audio stream can be described as a cosine wave. Therefor, the information signal that is to be transmitted in broadcast radio can be represented as a cosine wave in the form of

$$m(t) = A_m\ cos(2\pi f_m t). \tag{2.4}$$

By inserting this message signal into the generic FM formula 2.2, the final equation for an FM signal transforms into the following form.

$$y(t) = A_c\ cos\Big(2\pi f_c t + \beta\ sin(2\pi f_m t)\Big) \tag{2.5}$$

Several formulas, as well as their derivations are described according to [**ref__FM__Maths__Info__1**][**ref__F**

2.3.2 Frequency Band, Channel allocation/distribution

2.4 Algorithms for Digital FM Demodulation

see literature/FmDemodulator.pdf (Sect. 3.3) see literature/00476180 Digital FM Demodulator for FM, TV, and Wireless.pdf (Sect. II and III)

2.4.1 Baseband Delay Demodulator

2.4.2 Phase-Adapter Demodulator

2.4.3 Phase-Locked Loop (PLL)

2.4.4 Mixed Demodulator

Chapter 3

High-Level Synthesis

3.1 Introduction / State of the Art

3.2 Functionality (transform high-level code to HDL)

3.3 Language Support

3.3.1 C++

3.3.2 SystemC

3.4 Coding

3.4.1 Compiler Directives (#pragma's)

3.4.2 Data types

3.4.3 Functions

3.4.4 Loops

3.4.5 Conditional statements

3.5 Advantages / Disadvantages

easy and much faster to get complex HDL code, such as image/video processing, through libraries like OpenCV. However, it is less optimized, and a developer still needs to have a strong hardware background, to be able to understand what is being generated in hardware from x lines of code in HLS/Cpp.

Chapter 4

System Architecture/Concept

4.1 Block Diagram (with details)

4.1.1 describe blocks

4.2 Which parts in HLS and VHDL

4.3 Test Environment

Chapter 5

Implementation

5.1 Matlab/Python Model

in fixed point, close to hardware level algorithm

5.2 VHDL (no shortcut)

5.2.1 Channel Selection (IF to Channel-BB)

5.2.2 Phase Detector

5.2.3 other Elements

5.3 High-Level Synthesis

5.3.1 Channel Selection (IF to Channel-BB)

5.3.2 Phase Detector

5.3.3 other Elements

5.4 Common Testbench

5.4.1 Architecture (same tb for VHDL and HLS-generated HDL)

5.4.2 Framework cocotb, with ghdl compiler

Instantiate both HDL models in the testbench.
Display a direct comparison of outputs in graphs. This is practical, since the cocotb framework runs in python and graphs can be generated using Python's matplotlib.

5.5 Develop HLS compiler independent code (optional)

5.5.1 Challenges

Different compilers use different #pragmas, etc.
The #pragmas need to be within the code, at the appropriate positions, which makes it difficult to write compiler independent code....

Chapter 6

Deployment on Hardware

6.1 Hardware Platform

6.1.1 RTL2832u Dongle

6.1.2 ZedBoard

Chapter 7

Test Results

7.1 Functionality

7.1.1 Implementation effort/time

7.1.2 Hardware Utilization

7.1.3 Others

Chapter 8

Using Literature and other Resources

[1]

Appendix A

Technical Details

Appendix B

Supplementary Materials

List of supplementary data submitted to the degree-granting institution for archival storage (in ZIP format).

B.1 PDF Files

Path: /

thesis.pdf . . . . Master/Bachelor thesis (complete document)

B.2 Media Files

Path: /media

*.ai, *.pdf . . . . Adobe Illustrator files
*.jpg, *.png . . . . raster images
*.mp3 . . . . audio files
*.mp4 . . . . video files

B.3 Online Sources (PDF Captures)

Path: /online-sources

Reliquienschrein-Wikipedia.pdf [2]

Appendix C

Questionnaire

Appendix D

LaTeX Source Code

References

Literature

[1] Hubert M. Drake, Milton D. McLaughlin, and Harold R. Goodman. *Results obtained during accelerated transonic tests of the Bell XS-1 airplane in flights to a MACH number of 0.92*. Tech. rep. NACA-RM-L8A05A. Edwards, CA: NASA Dryden Flight Research Center, Jan. 1948. URL: https://www.nasa.gov/centers/dryden/pdf/87528main_RM-L8A05A.pdf (cit. on p. 11).

Online sources

[2] *Reliquienschrein*. Sept. 2018. URL: https://de.wikipedia.org/wiki/Reliquienschrein (visited on 02/28/2019).

Check Final Print Size

— Check final print size! —

width = 100mm
height = 50mm

— Remove this page after printing! —