

Opleidings- en begeleidingsplan

M.G. v/d Horst

M.G.v.d.Horst@tue.nl

Version: 2.0
Last update: October 2, 2006

About this document

This document will serve as a guide for the planning and supervision of my (Martijn van der Horst) PhD project and will gradually turn into a log of the project. The planning will be updated as the project progresses and any completed activities will be recorded.

To do this this document will have to be updated regularly. The most recent version can be found on the project's website (www.win.tue.nl/~mhorst). The version of the document and the date it was last modified can be found on the title page. The main version number indicates the year of the project, so the document with main version number 2 should contain the activities of the first year and a (rough) planning for the three remaining years.

Chapter 1

The Project

1.1 Subject

According to Moore's law the capacity of computers doubles every one and a half years. However, if we look at the developments in communication technology in the past few years, we see that the capacity of our communication channels increases even faster, three times as fast, according to Gilder. If this trend continues communication channels themselves will no longer present a bottle neck, but the signal processing that has to be done at the end points of the channel will. Hence the name of the project Gilder beats Moore.

1.2 Goals

The goal of this project is to write, present and pass a doctoral thesis by Martijn van der Horst.

This thesis should examine the implementation of signal processing algorithms. These implementations should be such that the sample rate can be higher than the processing rate of the elements that form the implementation and the scalability of these implementations (with respect to their sample rate and processing rate ratio) should be good.

The recursive signal processing algorithms are the most interesting in this respect. Parallelizing nonrecursive filters is rather straightforward, but for recursive filters the recursive loop introduces problems.

Another class of filters that are difficult to implement in this manner are adaptive filters. The coefficients of these filters change during the operation of the filter. This means that the implementation must be capable of handling these changes.

1.3 Participants

The following people participate in the project:

Name	Role
Kees van Berkel	Promotor
Martijn van der Horst	PhD student
Johan Lukkien	Main supervisor
Rudolf Mak	Supervisor

Chapter 2

Planning

The project started at 17 november 2003 and will last until 17 november 2007.

2.1 General meetings and reports

Every four weeks the participants will meet to discuss the progress of the project and to suggest further avenues of research. At the end of these meetings the next meeting is planned. They will generally be held on a tuesday afternoon in Johan Lukkien's office.

Whenever necessary I and Rudolf Mak will meet on the tuesday afternoons in which no meeting with the other supervisors is planned. These meetings are held to inform the supervisor of the progress of the project and to discuss problems.

I will keep the project's website (www.win.tue.nl/~mhorst) up to date. The website contains this document, small reports on the progress of the project and a bibliography of the material that I have read.

2.2 Phases

The project can be divided into several phases. Here we will give an overview of those phases and the subjects that will be studied during them.

Phase	Subject	Comments
Introduction	Scaler	The project's website contains the findings.
All-pass Filters	All-pass IIR	The project's website contains some of the findings.
Vector processor	Vectorizing IIR	Current research topic.
Future	Adaptive Equalizers	Possible future research topic.
	Viterbi decoders	Possible future research topic.

2.2.1 Introduction phase

During the introduction phase the literature will be studied, but to familiarize myself with the field also some small practical problems will be considered.

The scaler is a component which scales signals, it is non-recursive and therefore quite easy to implement. The project's website contains a report of the findings. This report will probably be published as a technical note.

2.2.2 All-pass filter

The all-pass IIR filter is a recursive filter with a constant amplitude frequency response. While the block implementation of IIR filters has been studied extensively in the literature the all-pass IIR filter has some properties that might simplify these implementations.

There are some ways to save hardware in the case of the (incremental) block state implementations, but this amount is small compared to the 50% that is possible in single-input single-output implementations.

A novel way of using the clustered look-ahead technique which could be applied to the all-pass filter was also found. However the resulting implementation required higher precision arithmetic and would be only marginally stable. The marginal stability property means that round-off errors would accumulate in a linear fashion, rendering the filter useless unless it would be reinitialized periodically.

2.2.3 Vector Processor

Implementing the IIR architectures in VLSI is one thing, writing a vector program that has the same properties is another.

This research is currently going on and the results so far can be found in a report on the website. The main conclusion so far is that intra-vector operations (specifically the intra-sum) and load- and store-with-stride operations are crucial to the scalability of the implementation.

2.2.4 Future phases

Adaptive Equalizers are filters for which the filter coefficients change over time, this presents new challenges for the implementation.

Viterbi decoders is an interesting subject, but it is extensively covered in literature and we do not expect that any significant progress could be made by studying it.

2.3 Conferences

I participated in the following conferences:

Date	Title	Comment
13-02-2004 to 15-02-2004	WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Databases (AIKED 2004)	As co-author of the article "Fast optimum route planning for car navigation systems"

The plan is to participate in the following conferences in the future:

Date	Title	Comment
Unknown	IPA-dagen	At least 4 times
Unknown	Summerschool	An international summer-school

2.4 Courses

I have participated in the following courses:

Date	Course
27-01-2004	Tutor training faculteit Wiskunde & Informatica
05-03-2004 to 25-06-2004	Writing Articles and Abstracts
31-01-2005 to 04-02-2005	IPA course "Algorithms and Complexity"

Still to follow are:

Date	Course
Unknown	IPA course "Formele Methoden"
Unknown	IPA course "Software Technologie"
Unknown	Course "Presenteren van eigen onderzoek"
Unknown	Courses on signal processing of the electronics faculty

2.5 Teaching obligation

A PhD student has the duty to teach, or do some other work to help in the education of the university's students. This should take about 10% of a PhD student's time. This means the total time should be approximately 664 hours.

So far I have done:

Year	Trimester	Role	Hours
2003-2004	2	OGO 1.2 Tutor	90
2003-2004	2	Correcting OPP tentam	16
2003-2004	2 and 3	Informing (future) students	8
2003-2004	3	Correcting OPP tentam (retries)	8
2004-2005	1	OGO 1.1	90
Total:			212

The remaining part of this duty will probably be fulfilled by helping out at practica, or making tentams for courses like "Inleiding Parallel Programmeren" or "Ontwerpen Parallele Programma's".

2.6 Other Events

Date	Course
13-05-2005	Project name changed from "Shannon beats Moor" to "Gilder beats Moore"