CLAUS
CLever AUtomated Scientist

Florian Bärwolf
baerwolf@ihp-microelectronics.com

July 4, 2020

Contents

1 Introduction

This is a documentation for CLAUS. CLAUS is a program, which automates the calculation process of raw data measurement files (e.g. intensities) into sample property files (e.g. concentrations). It is developed under Linux in objectorientated C++11, is 100% STL compatible and can therefor be easily (cross) compiled to various platforms and operating systems. It is a very simple command line program with no GUI.

2 Version

The current version is "2020-06-21_7 ALPHA"

3 Motivation

Calculation processes became more and more complex over the past years. There are many places for a scientist to make errors: swapping numbers, using wrong dose or concentration or sample . . .
Scientists or operators might use different calculation methods or models, which lead to different results.
The main motivation is to decrease these errors, to unify and clarify calculation methods within one software for all measurements and calculation methods.
The reason for using C++: there are many established 3rd party libraries like GSL or sqlite, it is close to the actual hardware, so it is pretty fast and reliable (about 100 times faster compared to matlab), it is free for everybody and can be used on countless platforms and operating systems and I personally already knew it.

4 Definitions

sample . . . contains all data which are independent from measurement tool parameters. E.g. material properties like concentration over depth, substrate, layers, ... Samples are distinguished from eachother by their lot, waferid, monitor and chip X/Y from the file name.

isotope . . . e.g. "74Ge2" is defined by the chemical element nucleon number and its symbol. The number of atoms is optional.

cluster . . . e.g. "74Ge2 31P" is made out of a certain amount of isotopes. A cluster containing only exactly 1 isotope is an isotope.

measurement_group . . . contains all measurements measured by the same measurement tool parameters, their common calculation parameters (like gRSF, gSR, ...). ;easurement_groups are distinguished from eachother by their olcdbid, groupid, measurement tool and actual measured clusters/elements. That means all measurements within a measurement group must have data to exactly the same clusters.

measurement . . . contains all data from the clusters of the primary file and its secondary measurement files (e.g. craters). It is a mixture of sample properties and measurement tool parameters. A measurement is default defined by its file name parts: olcdbid, lot, wafer, chip (if set), monitor(if set), groupid, repetition

measurement_tool . . . is determined by the data parser from the input files (e.g. dsims, tofsims, xps, ...) and possible measurement settings like sputter element, energy, polarity

calculation_manager . . . allows calculation over all measurement_groups. not implemented yet.

secondary_measurement . . . is a measurement of something of the first measurement created/destroyed. e.g. a crater measurement generated by SIMS is a secondary measurement

5 Functionalities

This program was designed to be CPU time efficient from the ground. Many calculation results will be saved in the hope to be useful at a different point in run time. Sacrificing memory over CPU time. Execution time for about 1900 files was 44 sec on intel i5@4.4GHz, while memory usage was peaking at 115MB RAM.

5.1 file name parsing

The correct input file name format is ***very*** important. The program ***needs*** these information to calculate correctly.
File name is splitted in to parts distinguished by a delimiter “_”. The file name parts are:

- necessary: olcdbid, groupid, file typ ending, lot/samplename
- optional: lotsplit, monitor, chipX/Y, crater depths, sputter energy, sputter element, sputter polarity, repetition, lot, wafer number

The algorithm tries to parse these parts as a regular expression in a specific order:

1. olcdbid: ... will be parsed as an integer so “0001” is same number like “1”
2. wafer: ... will be parsed as integer
3. lot: ...
4. chip X/Y
5. monitor
6. groupid
7. sputter energy, polarity, element
8. crater depths
9. repetition: a char to distinguish multiple measurements of exactky the same sample in exactly the same measurement group. It is added directly after the group.
10. not parseable

    some filename examples:

- 51880_ECY011_w12_c6-4_g4.TXT
- 51880_ECY011_w14_X06Y05_g4.tofsims.txt
- 51880_ECY011_w12_X06Y04_g4.tofsims.txt
- 51880_MHZ115_w11_g4.tofsims.txt
- 51880_ECY011_w13_c6-4_g4.TXT
- 51880_MHZ115_w11_g4.TXT
- 51880_ECY011_w13_X06Y04_g4.tofsims.txt
- 51880_SJZ292_w01_g4.tofsims.txt
- 51880_ECY011_w14_c6-5_g4.TXT
- 51880_SJZ292_w01_g4.TXT
- 52022_C3659_450VO+_g2.dp_rpc_asc

5.1.1 Example 1

file name = 51987_A_500VCs-_g1q_325nm_3255A.TXT

- olcdbid = 51987
- groupid = 1
- file typ ending = TXT
- sputter polarity = -
- sputter element = Cs
- sputter energy = 500 [eV]
- repetition = q
- crater depths = 325 [nm], 325.5 [nm]
- not parseable = A
- chipX/Y is empty
- lotsplit is empty
- lot is empty
- wafer number is empty
- monitor is empty

5.1.2 Example 2

file name = 51523_ZA22_w23_chip-03-04_450VO+_8q.dp_rpc_asc

- olcdbid = 51523
- groupid = 8
- file typ ending = dp_rpc_asc
- sputter polarity = +
- sputter element = O
- sputter energy = 450 [eV]
- repetition = q
- chipX/Y: X=3, Y=4
- lot = ZA22
- wafer = 23
- crater depths is empty
- lotsplit is empty
- monitor is empty
- not parseable is empty

5.1.3 Example 3

file name = 52022_C3659_450VO+_g2w.dp_rpc_asc

- olcdbid = 52022
- groupid = 2
- file typ ending = dp_rpc_asc
- sputter polarity = +
- sputter element = O
- sputter energy = 450 [eV]
- repetition = w
- chipX/Y is empty
- lot = C3659
- wafer is empty
- crater depths is empty
- lotsplit is empty
- monitor is empty
- not parseable is empty

5.2 file filtering

Filters files with a specified file format from a config file or ignores them. Only the wildcard "*" is allowed, which is very useful in windows to apply multiple files in one action to CLAUS.
You can specifiy the following commands in a config file:

- "ignore_file_type_endings" will ignore the endings of the input files and tries to parse the contents no matter what
- "ignore_filename_substrings" will ignore all files containing this string within their file names
- "ignore_filename" will ignore this specific file
- "use_directory_files_list" if set to 1, will load all files in all directories of each input file and tries to use them for calculation

5.3 file content parsing

After filtering the input files and parsing the file name, the algorithm will try to parse the contents of the files with several parser methods origining from the different measurement tool exports

- dsims: dp_rpc_asc
- tofsims: txt
- images: jpg, jpeg (just parsing file name, no contents yet)
- dektak32: txt
- xps: csv

5.4 calculation

Standard SIMS calculation techniques are utilized, as well as jiangs[1] protocol.

5.5 database

A sqlite3 file, which contains the reference parameters. At the moment this database will not be automatically populated.

5.6 export

To export the measurement groups, measurements and clusters. Very rudimentary class at the moment.

5.7 plotting

Utilizing gnuplot, this is only possible under Linux at the moment.

6 calculation methods

If the matrix of all measurements within the measurement group consists of exactly one element, e.g. 100at% Si, than the following methods will be applied to get the concentration and sputter depth.

6.1 sputter depth

1. skalar: Sputter depth = sputter time * sputter rate

2. vector: Sputter depth = integrate(sputter time) d(sputter rate) for each point

6.2 total sputter depth

If there were crater depths in the file name, then their mean will be used. If there are no crater depths in the file name, then the mean of the depths of the fitted line profiles will be used.

6.3 sputter rate

The equilibrium position will NOT be used.

1. sputter rate = total sputter depth / total sputter time

2. sputter time of fitted (polynom, 17th grade) maximum concentration/intensity will be used to calculate sputter rate from database maximum depth.

3. mean of all sputter rates from other measurements in the group

6.4 RSF

SF * reference intensity

6.5 SF

1. For reference samples: if the last value of the implanted isotope/cluster is lower than 5% compared to the maximum value of the implant, then integrate the whole intensity profile starting from the equilibrium position
2. For reference samples: maximum intensity / maximum concentration from database
3. mean of all RSFs for this cluster from other measurements in the group / reference intensity: SF= mean(RSFs) * reference intensity point by point

6.6 concentration

concentration = SF * intensity

6.7 reference intensity

Is the median of the sum of the intensities of all reference clusters.

6.8 equilibrium position

The algorithm to determine the equilibrium position for a measurement is quite compilcated. If there is a cluster with a minimum position in the vicinity of the surface, than this will be used as equlibirum position.

6.9 jiangs protocol

Will only be applied to substrates consisting out of exactly 2 elements. If the measured reference samples spread over the whole substrate range (0-100at%) then jiangs protocol can not be applied. Keep the relative concentrations of all samples below 50at% or you should know what you are doing.
Matrix concentration and SRs are interpolated linearly. RSFs are interpolated by highest polynom grade possible (e.g. cubic for 4 reference samples).

6.10 planned

- bug fixes
- more bug fixes
- ... fix bugs

7 How to use CLAUS?

Define a config file from the template. Be sure, that the filenames of your exported files are correctly formatted, then simply drag and drop them with one or more config files on to the executable / binary. Commandline usage:
linux: "./claus config.conf samplefile1 samplefile2 *multiplefiles*"
windows: "claus.exe config.conf samplefile1 samplefile2 *multiplefiles*"
When you have multiple measurements or secondary measurements of the same sample, you can try to give them to the program at the same time. CLAUS will try to use all of the given information from different tools to calculate the results or minimize the uncertanties.
If you mix up sample names, results can and most likely will be wrong.

8 config file parameters

Necessary:

- ”pse“: The file name with path to the periodic table of elements

Optional:

- measurement_definition = olcdbid n lot n wafer n monitor n chip n groupid n repetition n lot_split defines a measurement.

9 external libraries

The GNU Scientific Library (GSL) is used for statistics analysis. The sqlite3 library is utilized to communicate with the database. The program GNUplot can be used under Linux to directly plot some graphs.

10 license

11 History

Developing CLAUS started several years ago somewhere in 2017, but it was named different back in the days and cosisted of multiple different programs. Many other programs preceeded this one, until it was planned from the ground new. Parts of this program date back to the year 2009. It was structurally planned and designed from the ground up for its purpose.

References

[1] *Jiang Z. X. et al 2006 Appl. Surf. Sci. 252 7262*

[2] *Bärwolf et al Semicond. Sci. Technol. 34 (2019) 014005 (6pp)*