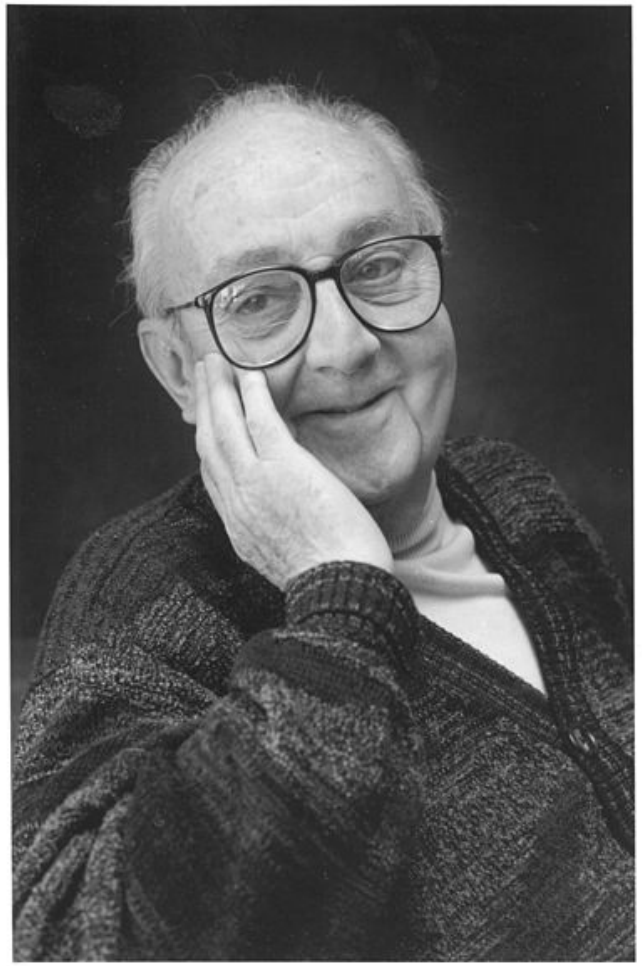
Preamble

"All models are **WRONG**, but some are **USEFUL**"

a citation attributed to George E.P. Box.

•



Born 18 October 1919
Died 28 March 2013

Fields Statistics

Institutions

ICI
Princeton University
University of Wisconsin–Madison

Alma mater : University College London

Known for

Response-surface methodology
Box–Jenkins method
Box–Cox transformation

NAPA

NAPA Université Nice Sophia Antipolis

A High Level Simulator of Mixed-Signal Systems

http://commons.wikimedia.org/wiki/File:Vineyards_in_Napa_Valley_7.jpg

Outline

NAPA through a simple example

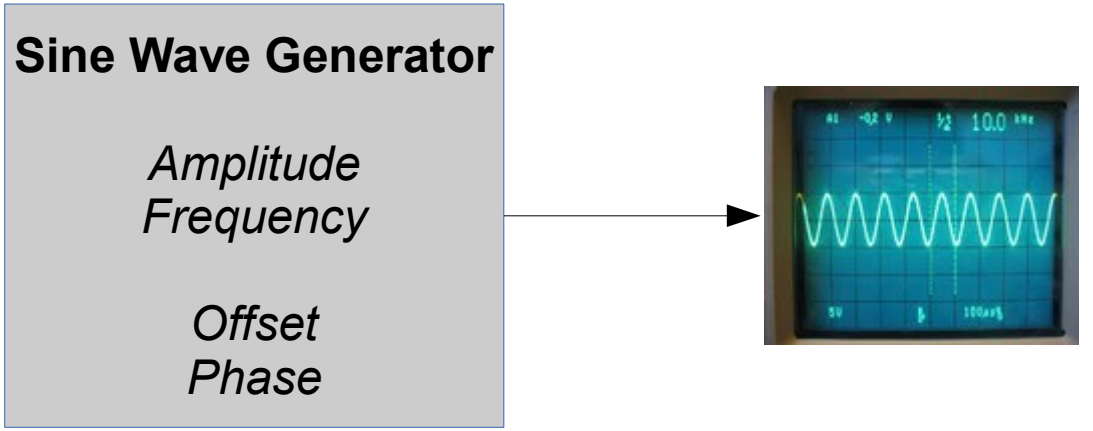
- ***main netlist***
- ***cell***
- ***primitive***
- ***user function***
- ***time domain simulation***
- ***smart tool***
- ***synchronization***

A realistic example

- ***cell generator***
- ***multirate transfer function***
- ***more...***

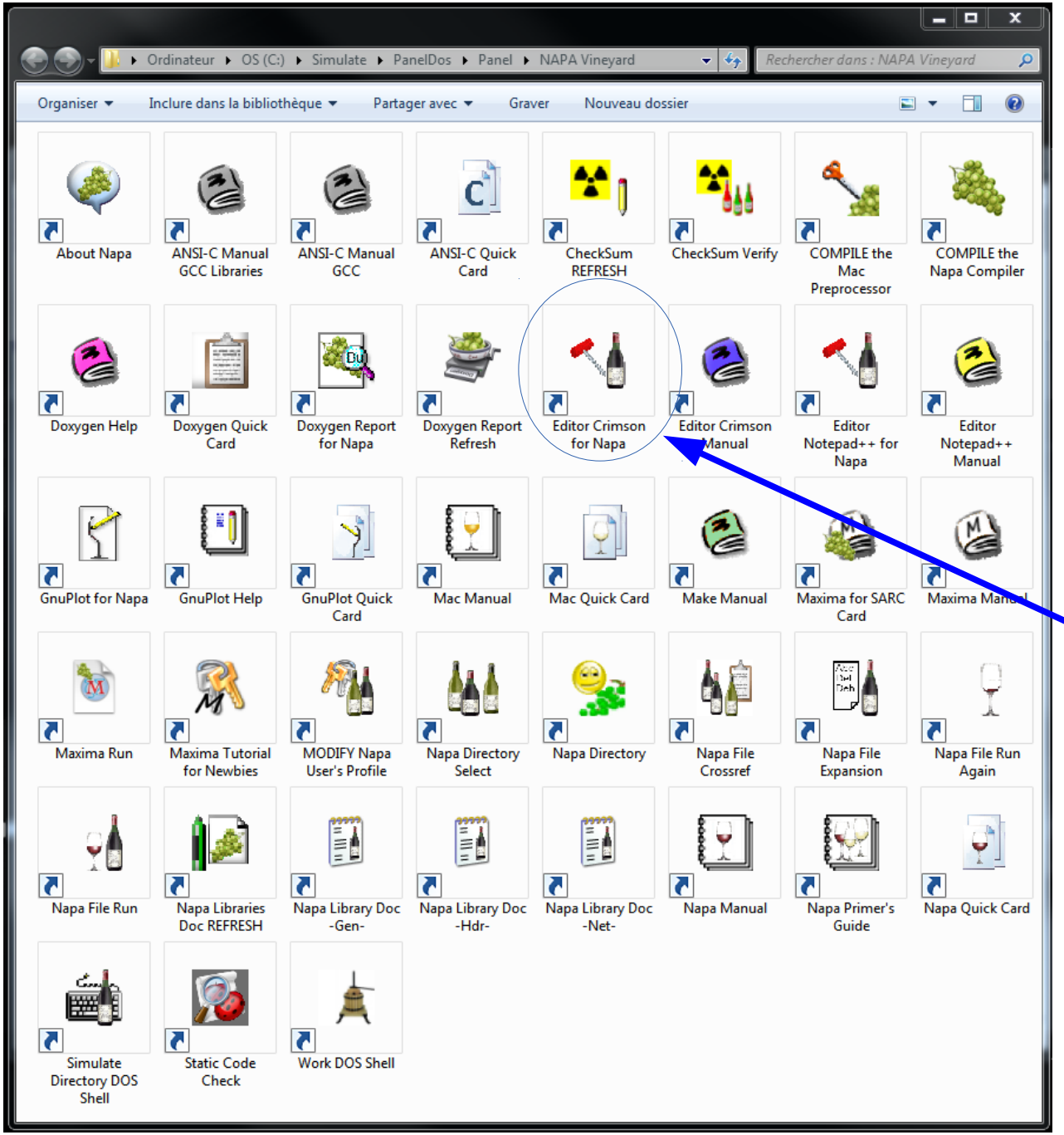
Open conclusion

A Simple Example : Generate a Sine Wave



We will show how we may easily adapt NAPA to address a specific user concern.

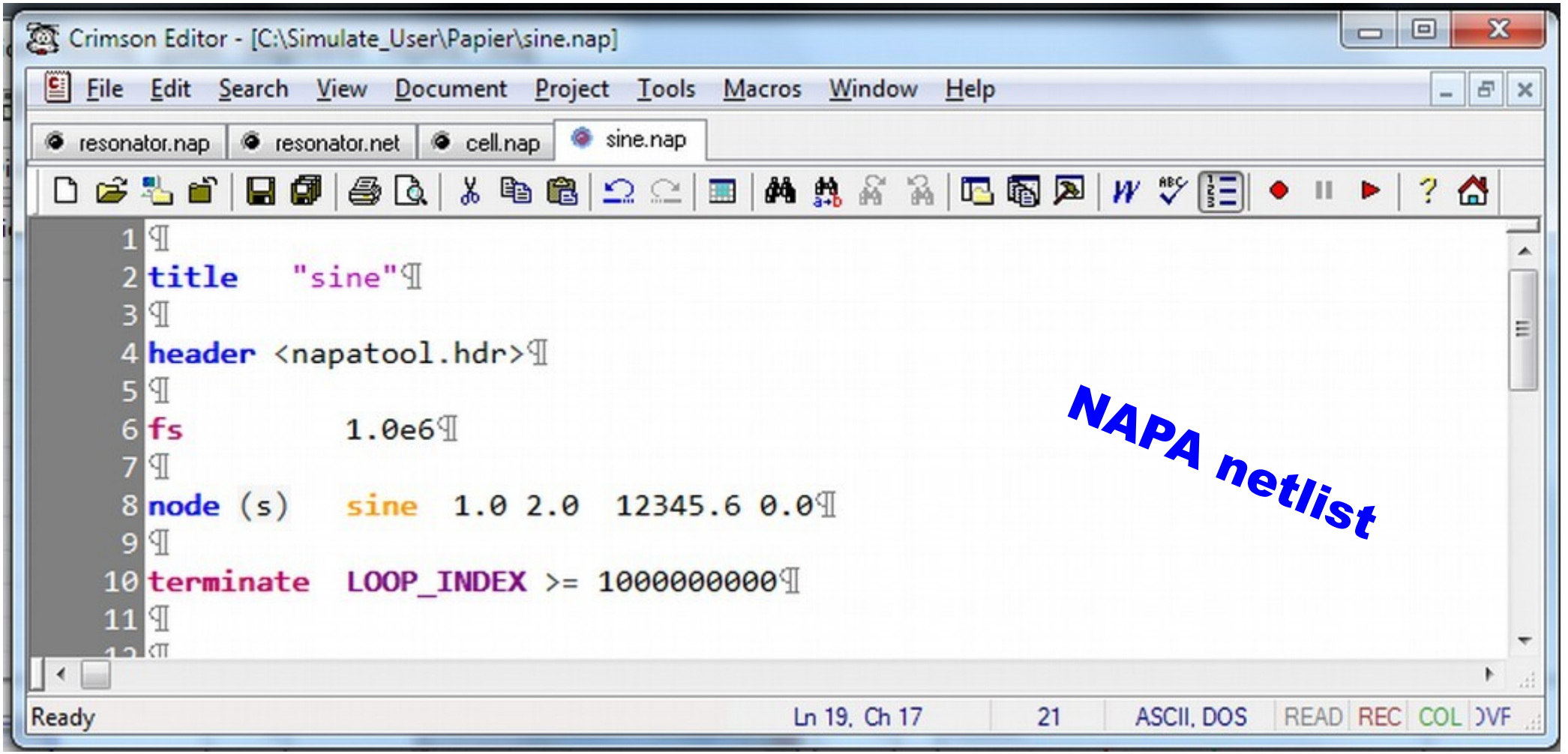
Preferred Environment : Windows 7, 64 bits



Preferred IDE

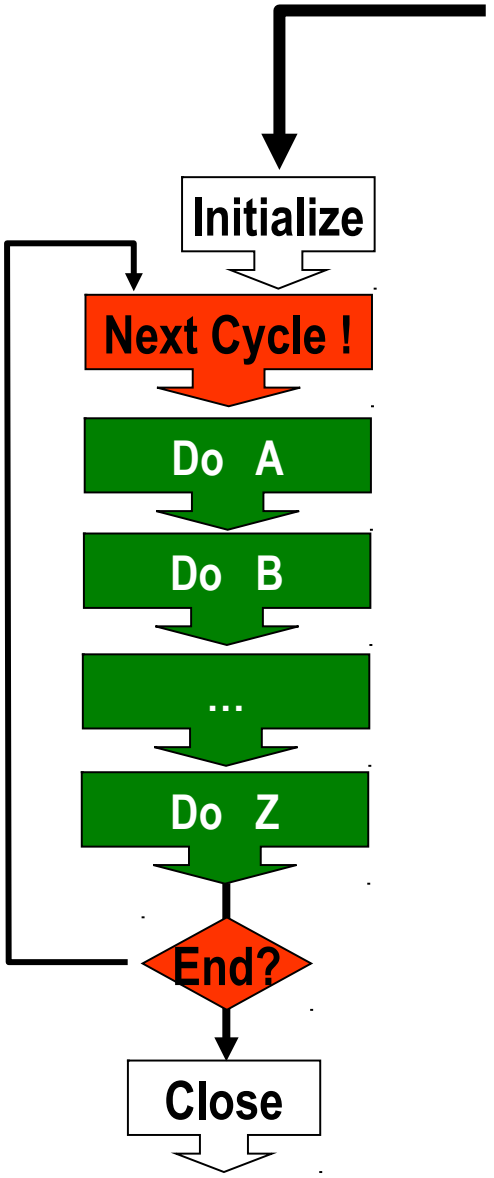
First Contact with NAPA

Generate a sine wave, frequency 12.3456 kHz, offset 1.0V, amplitude 2.0V (pk)



Tips : to compile and execute, i.e to simulate, press ***'Alt R'*** *from the NAPA netlist*

NAPA is a Cycle-Based Simulator Writer



Flow compiled before execution

Crystal clear flow

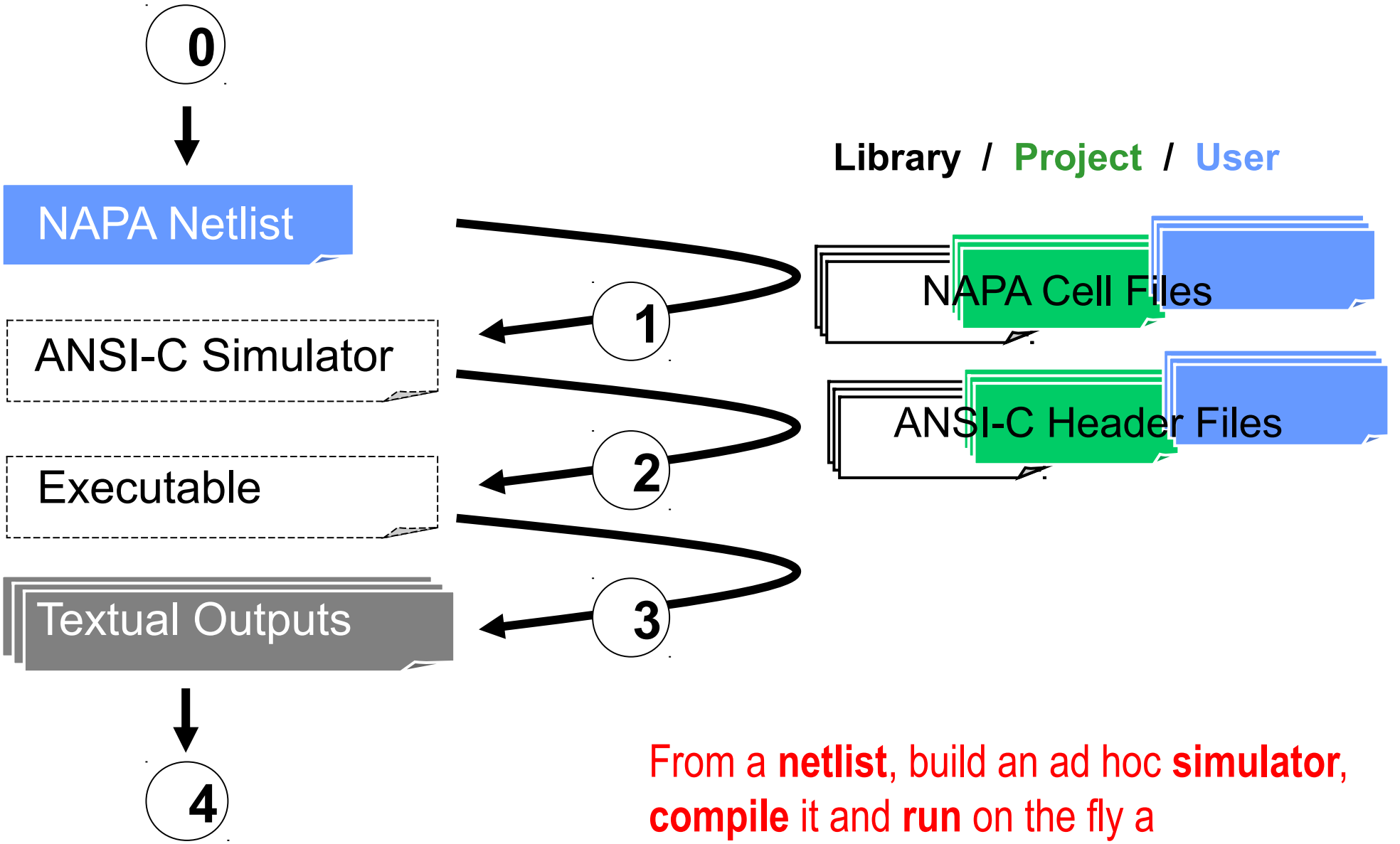
DATA Domain !
Control Domain ?

Mixed Signal
Data Processing

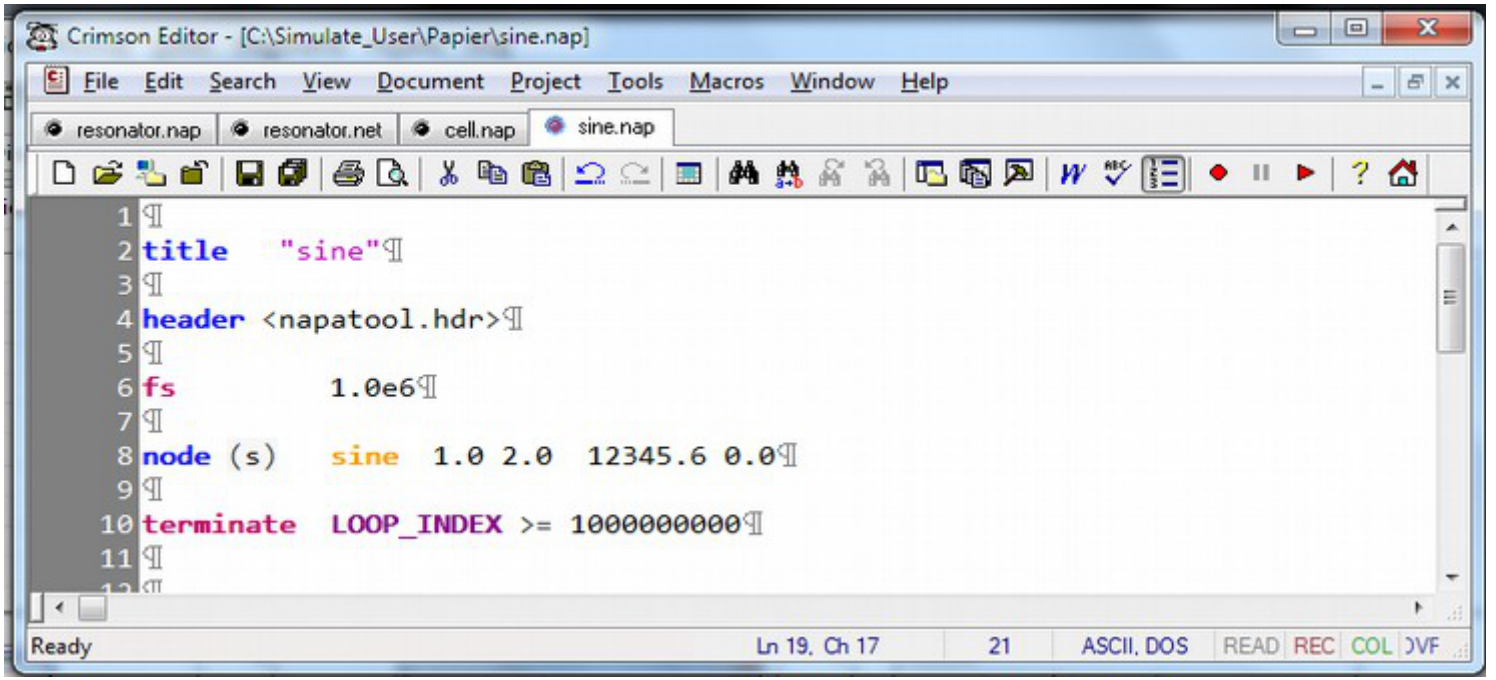
FFT friendly
Hazard free
Cycle accurate by construction
Very fast

Static loops detected and rejected

NAPA File Structure and Work Flow



From a **netlist**, build an ad hoc **simulator**, **compile** it and **run** on the fly a streamlined DEADLY FAST **executable**.



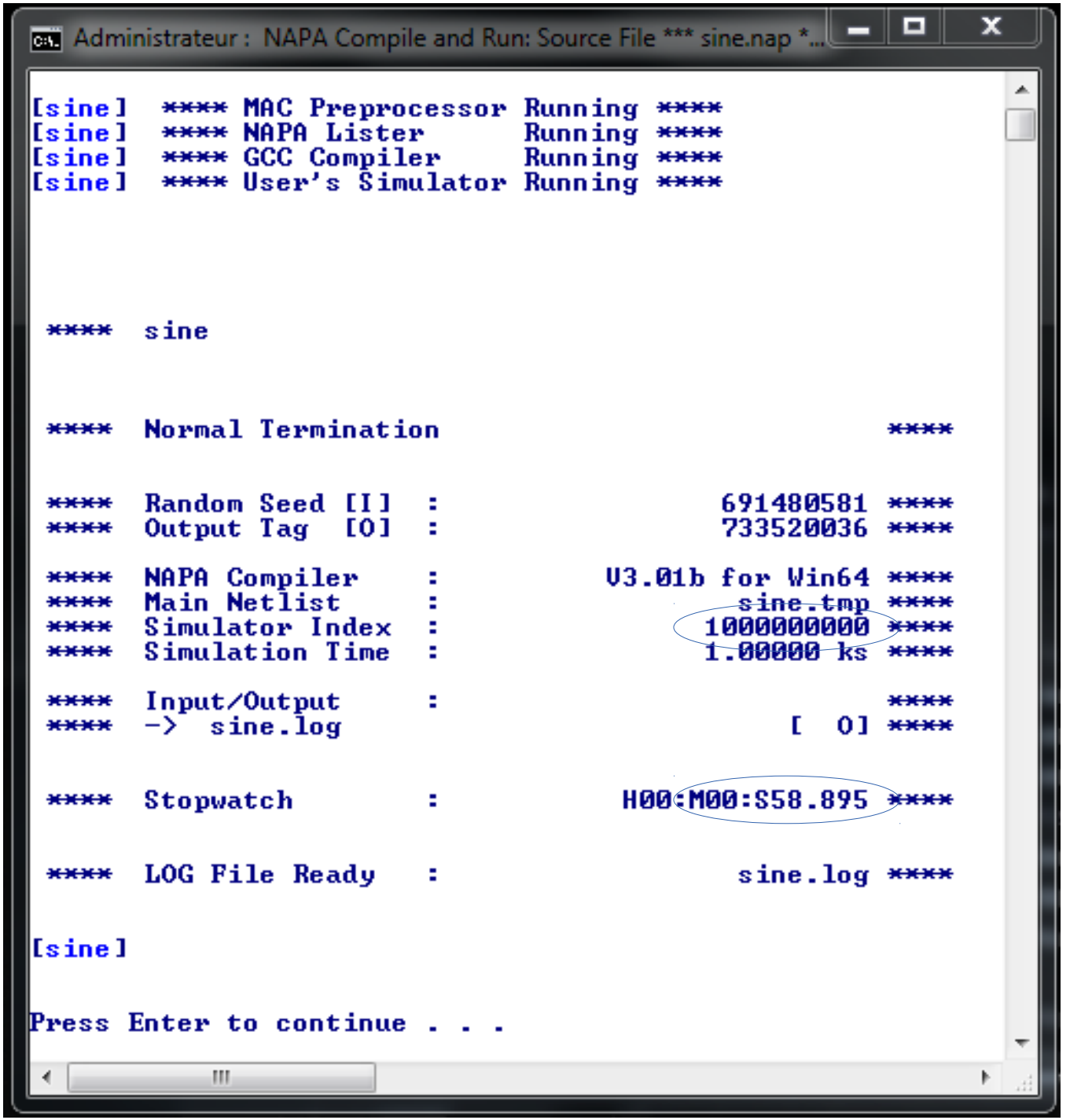
NAPA compilation

file "sine.c"

```
...
napa_abs_loop = 0.0L;

do {
  napa_abs_time  = napa_abs_loop * 1.000000000000000e-006L;

  d_node_s = 1.0 + (2.0) *
             ((R_TYPE) sinl(77569.692528316305093483152L * napa_abs_time));

  napa_abs_loop++;

} while (!TERMINATE);
...
```

ANSI-C code

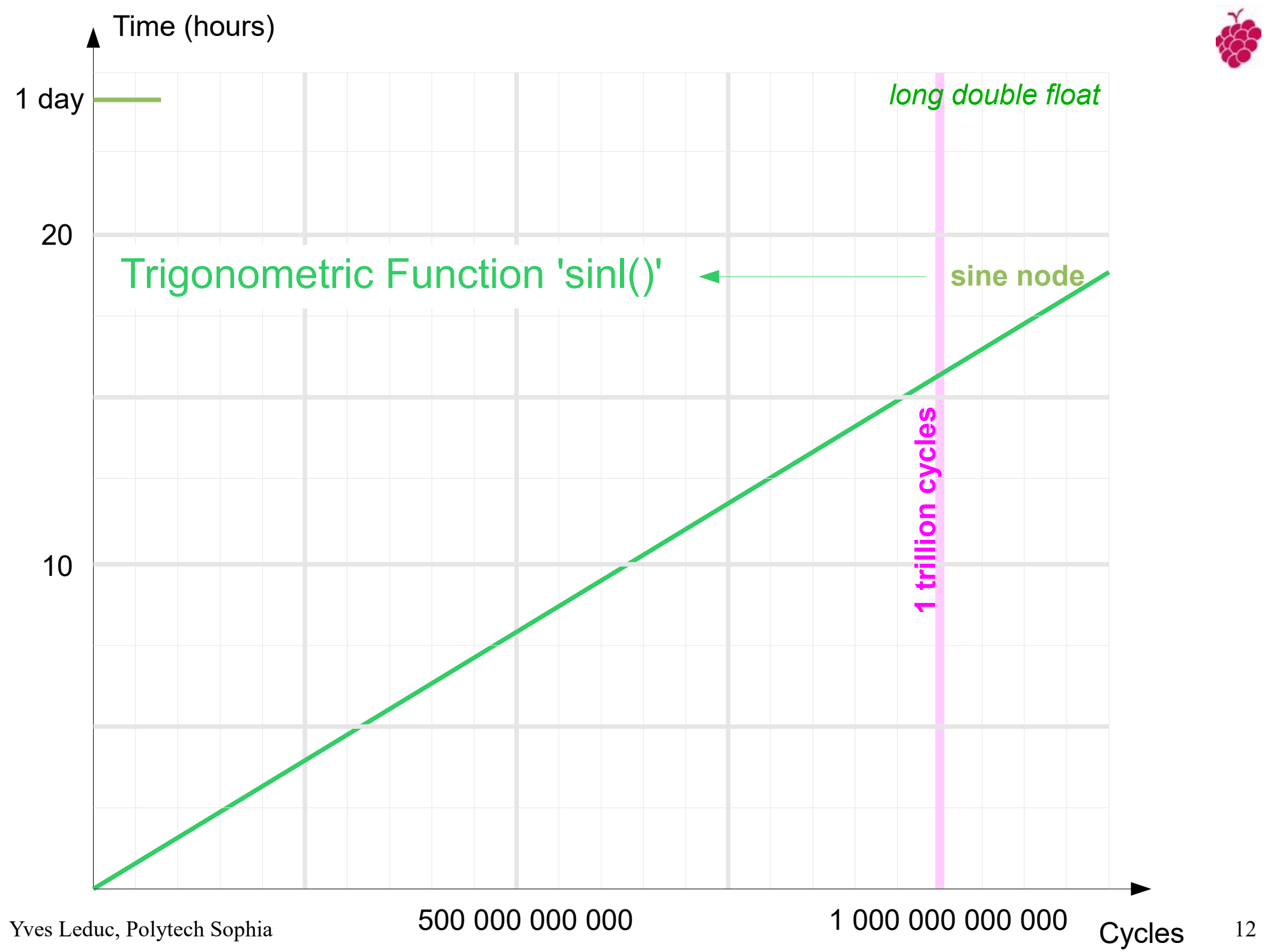
*Tips : toggle to the ANSI-C code, press **'Alt T'** from the NAPA netlist*



The simulation running on the screen

Preprocess
NAPA Compile
C Compile
Binary Execution

1 billion cycles
in 59 seconds


Time (hours)
1 day
20
10
long double float
Trigonometric Function 'sinl()'
sine node
1 trillion cycles
500 000 000 000
1 000 000 000 000
Cycles

Obviously too SLOW !!!!

Not happy with the current offer ?

We will implement a few ideas to speed up the generation of the sine wave.

In fact, it is a good opportunity to show how to ~~taste~~ *test* NAPA …

[Mathematics Do Help]

A sine wave is more than a set of sines, it is a sequence of sines where each sine value is correlated to the previous ones. It is therefore possible to take profit of this quality.

We will use a **RESONATOR** to produce the sine wave and saves a precious computation time.

The resonator is implemented as a **2-pole filter** described by the following **difference equation**

$$\mathbf{X_n = (k * X_{n-1}) - X_{n-2}}$$

with $k = 2.0 * \cos(2\pi \ fsinewave / fsampling)$

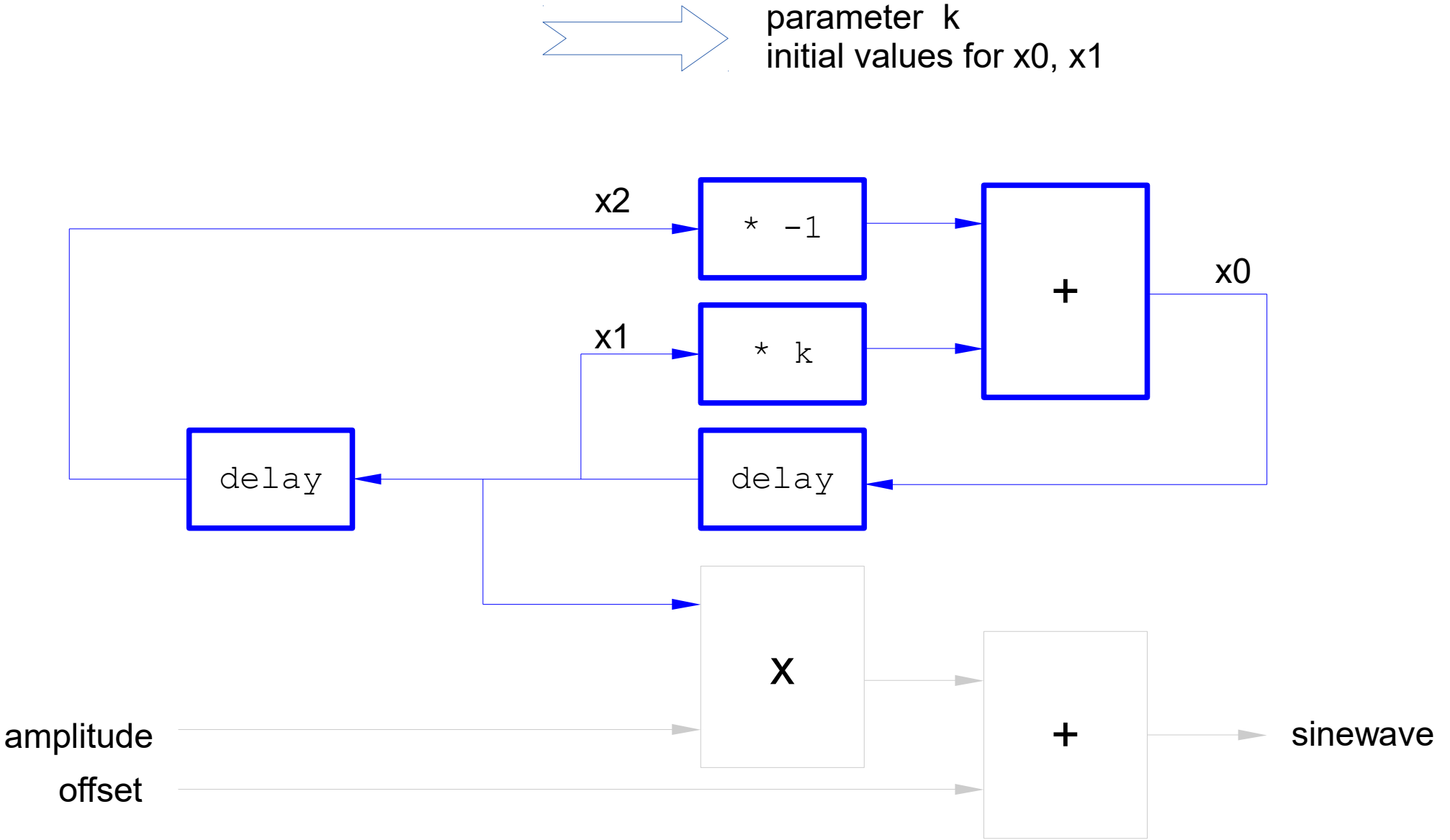
To start the oscillator, i.e. to set properly the initial conditions of the difference equation, we will set-up the initial conditions :

$$X_{n-1} = sin(phase)$$

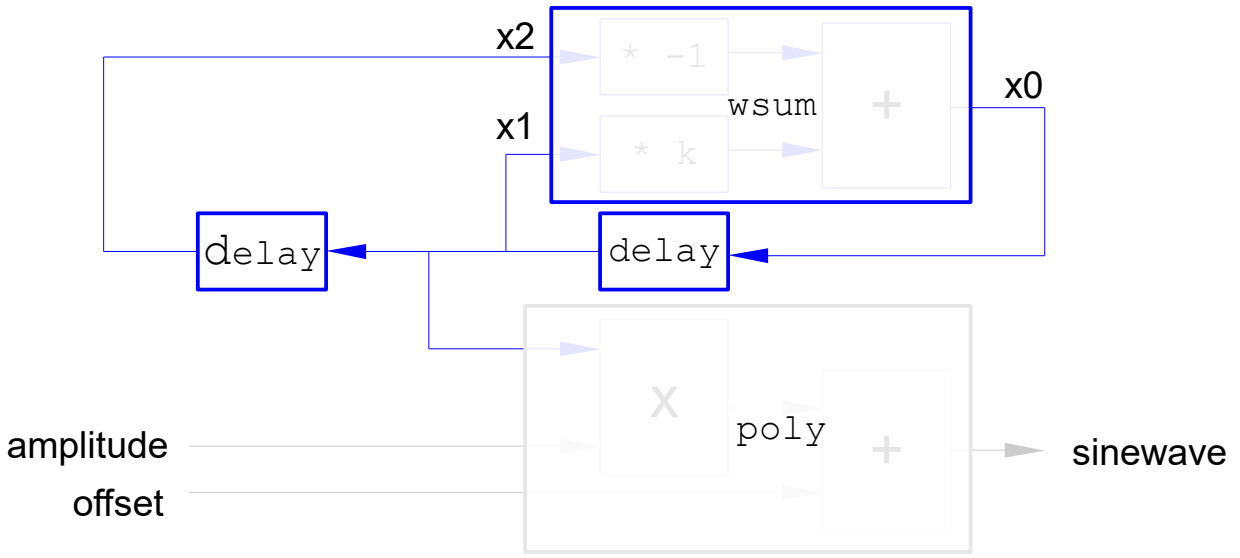
$$X_{n-2} = sin(phase - (2\pi \ fsinewave / fsampling))$$

The Second Order Resonator

Sine wave frequency, sine wave phase and sampling frequency



A Direct Description in a NAPA Netlist



Crimson Editor - [C:\Simulate_User\Papier\resonator.nap]

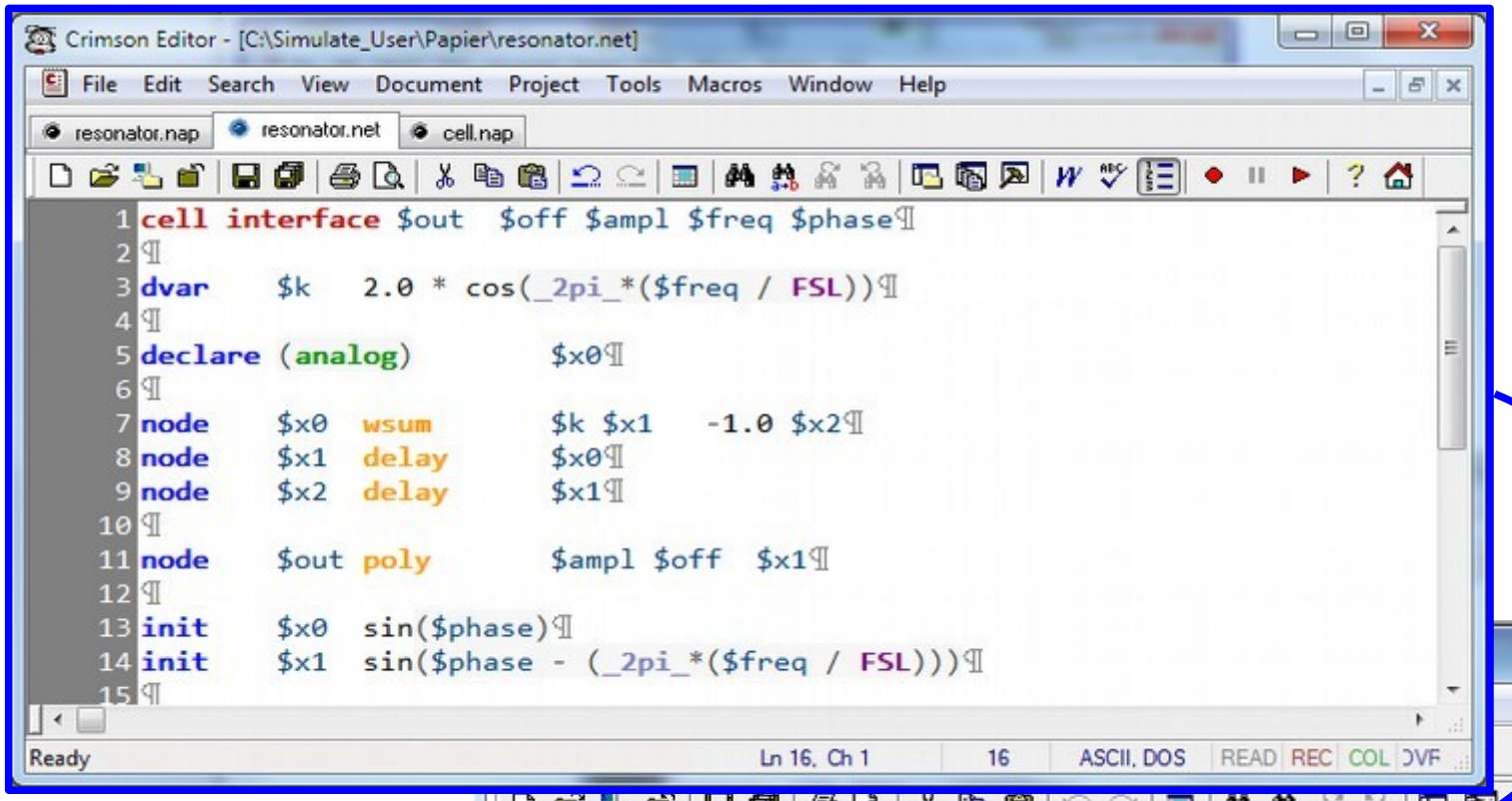
File Edit Search View Document Project Tools Macros Window Help

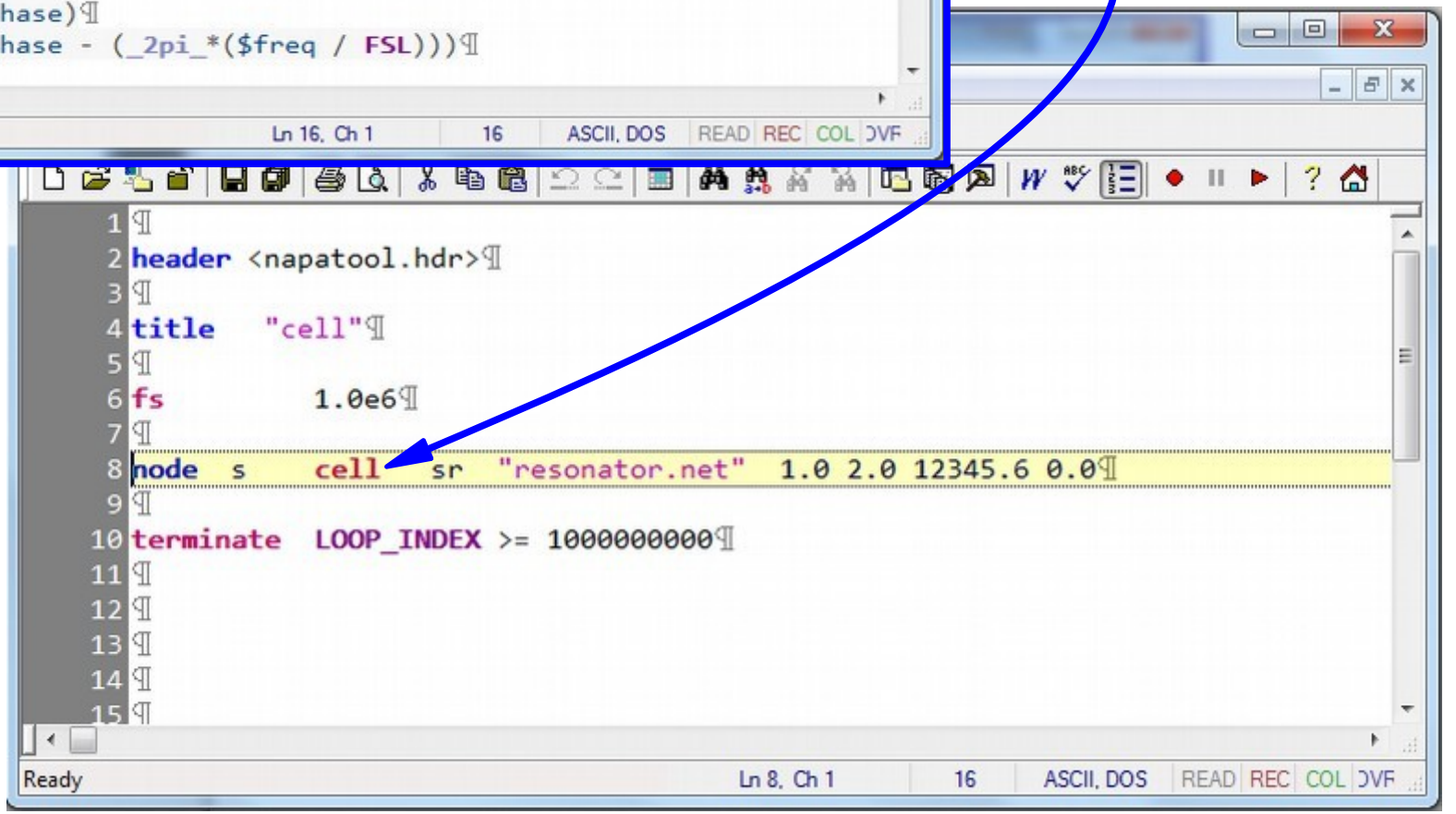
resonator.nap sine.nap

```
1  title   "resonator"¶
2  ¶
3  header <napatool.hdr>¶
4  ¶
5  fs   1.0e6¶
6  ¶
7  dvar    k   2.0 * cos(_2pi_*(12345.6 / FSL))¶
8  ¶
9  node    x0  wsum       k x1   -1.0 x2¶
10 node    x1  delay      x0¶
11 node    x2  delay      x1¶
12 node    s   poly       2.0  1.0   x1¶
13 ¶
14 declare (analog) x0 x1 x2¶
15 ¶
16 init    x0  sin(0.0)¶
17 init    x1  sin(0.0 - (_2pi_*(12345.6 / FSL)))¶
18 ¶
19 terminate  LOOP_INDEX >= 1000000000¶
```

Ready Ln 24, Ch 1 25 ASCII, DOS READ REC COL OVF

A Description Using a NAPA Cell


Crimson Editor - [C:\Simulate_User\Papier\resonator.net]
File Edit Search View Document Project Tools Macros Window Help
resonator.nap
resonator.net
cell.nap
1 cell interface $out $off $ampl $freq $phase¶
2 ¶
3 dvar $k   2.0 * cos(_2pi_*($freq / FSL))¶
4 ¶
5 declare (analog) $x0¶
6 ¶
7 node $x0 wsum $k $x1   -1.0 $x2¶
8 node $x1 delay $x0¶
9 node $x2 delay $x1¶
10 ¶
11 node $out poly $ampl $off $x1¶
12 ¶
13 init $x0 sin($phase)¶
14 init $x1 sin($phase - (_2pi_*($freq / FSL)))¶
15 ¶
Ready
Ln 16, Ch 1
16
ASCII, DOS
READ REC COL OVR


1 ¶
2 header <napatool.hdr>¶
3 ¶
4 title   "cell"¶
5 ¶
6 fs   1.0e6¶
7 ¶
8 node s   cell   sr "resonator.net" 1.0 2.0 12345.6 0.0¶
9 ¶
10 terminate LOOP_INDEX >= 1000000000¶
11 ¶
12 ¶
13 ¶
14 ¶
15 ¶
Ready
Ln 8, Ch 1
16
ASCII, DOS
READ REC COL OVR

The NAPA compiler expands the cells, flattening the hierarchy.

Tips : to get an insider view of the flattening process,

*press **'Alt E'** from the NAPA netlist*

Crimson Editor - [C:\Simulate_User\Papier\cell.exp]

File Edit Search View Document Project Tools Macros Window Help

resonator.nap | resonator.net | cell.nap | cell.exp

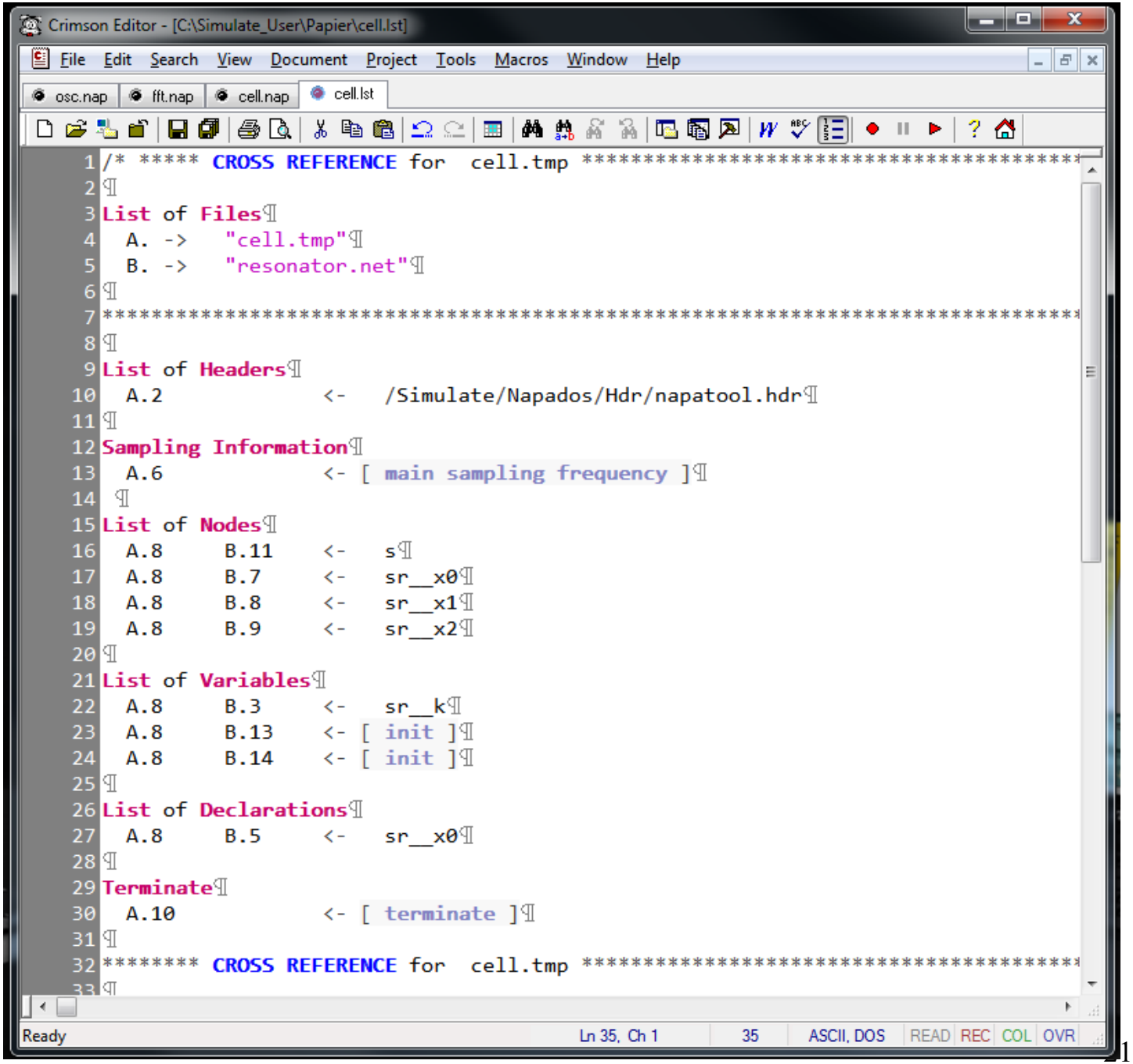
```
1  #* *****  EXPANSION OF FILE  cell.tmp  ************************** START ****
2  ¶
3  header <napatool.hdr>¶
4  ¶
5  title   "cell"¶
6  ¶
7  fs          1.0e6¶
8  ¶
9  #* >>  node  s    cell   sr  "resonator.net"  1.0 2.0 12345.6 0.0¶
10 ¶
11 dvar    sr__k   2.0 * cos(_2pi_*(12345.6 / FSL))¶
12 ¶
13 declare (analog)        sr__x0¶
14 ¶
15 node    sr__x0  wsum       sr__k sr__x1   -1.0 sr__x2¶
16 node    sr__x1  delay      sr__x0¶
17 node    sr__x2  delay      sr__x1¶
18 node    s poly       2.0 1.0  sr__x1¶
19 ¶
20 init    sr__x0  sin(0.0)¶
21 init    sr__x1  sin(0.0 - (_2pi_*(12345.6 / FSL)))¶
22 ¶
23 #* <<¶
24 ¶
25 terminate  LOOP_INDEX >= 1000000000¶
26 ¶
27 #* *****  EXPANSION OF FILE  cell.tmp  ************************** END ****
28 ¶
29
```

Ready | Ln 9, Ch 9 | 29 | ASCII, DOS | READ REC COL OVF

The NAPA compiler expands the cells, flattening the hierarchy.

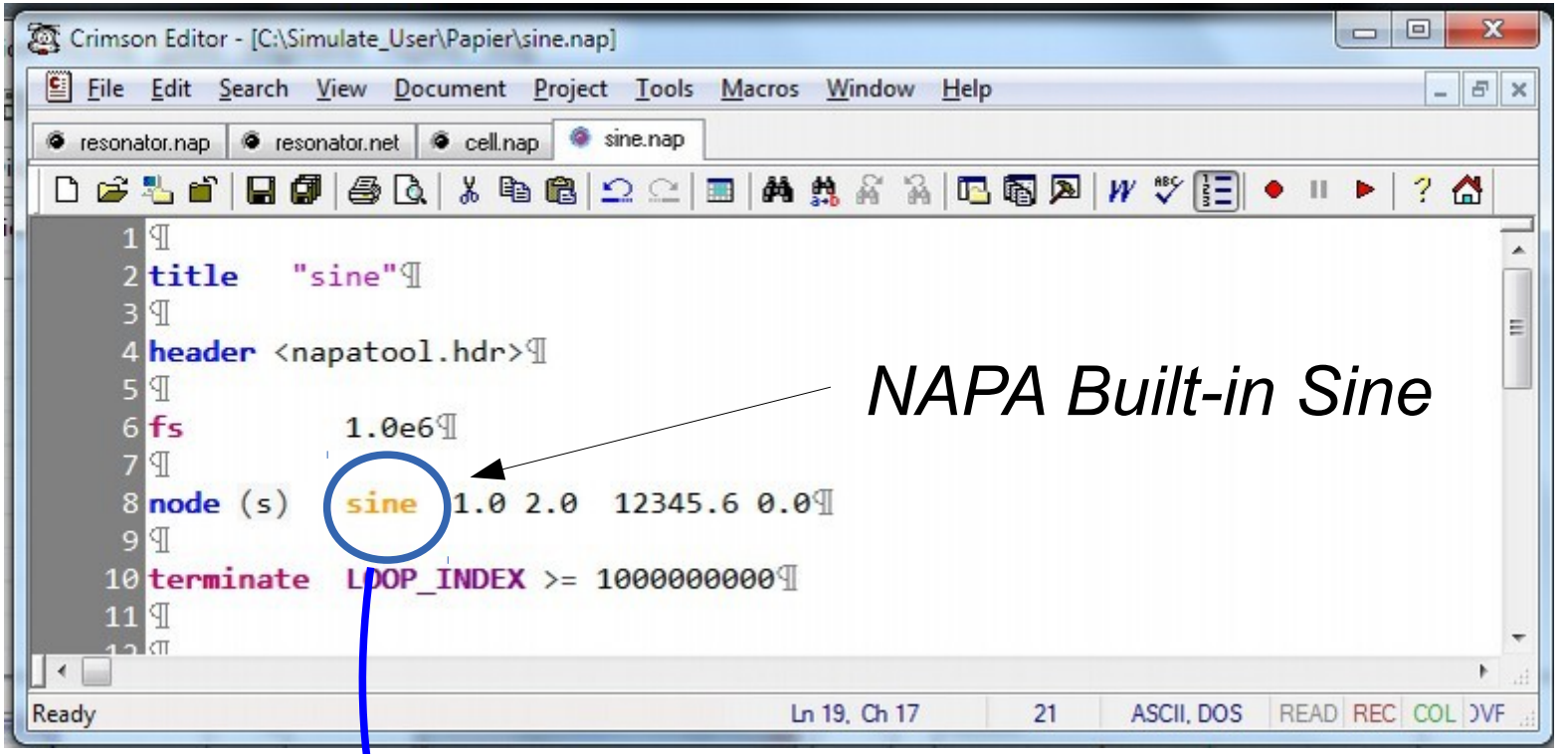
Tips : to get a cross reference of the NAPA netlist,

*press **'Alt X'** from the NAPA netlist*

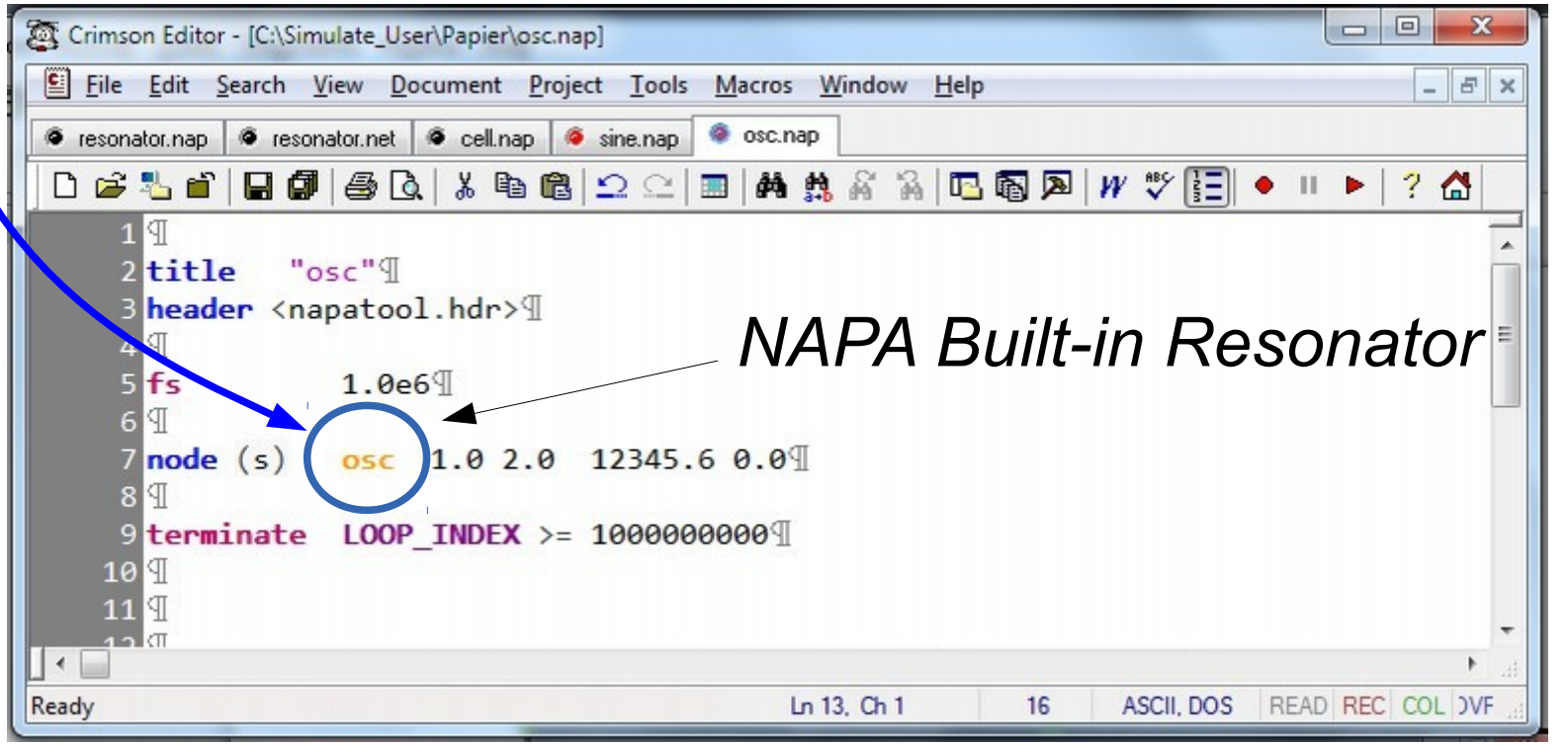


A Description Using a NAPA Primitive

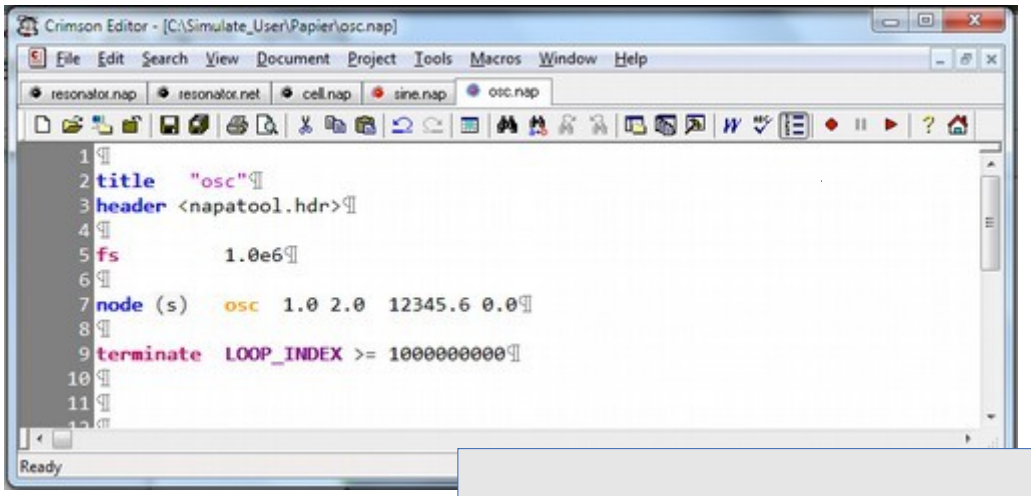
(in fact, this solution is so attractive that it is now a built-in primitive)



NAPA Built-in Sine



NAPA Built-in Resonator



NAPA compilation

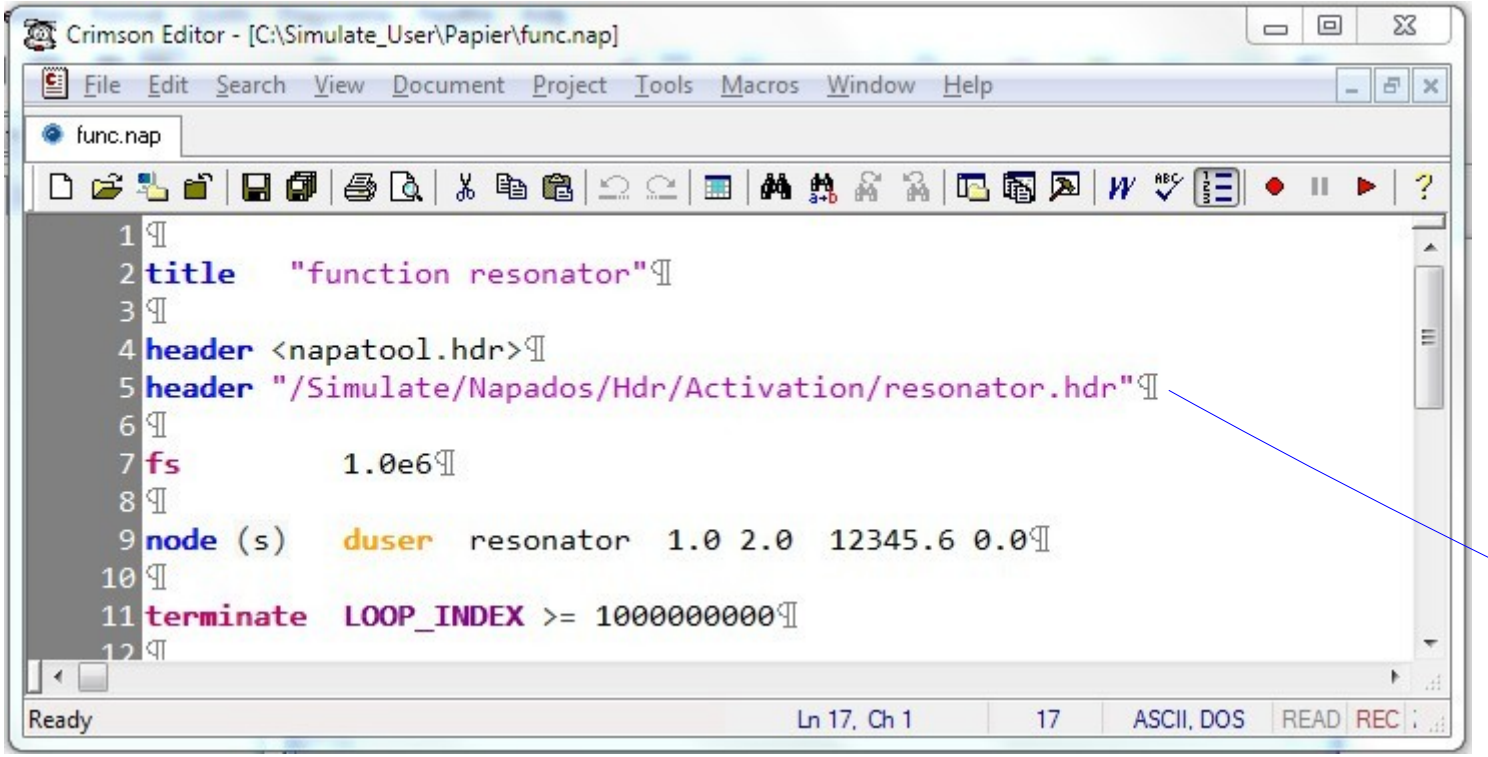
file "osc.c"

```
...
h_node_s_factor = 2.0L * cosl(_2PI_*12345.6L/((H_PREC) FSL));
h_node_s_osc0 = 0.0L;
h_node_s_osc1 = -sinl(_2PI_*12345.6L/((H_PREC) FSL));
h_node_s_osc2 = 0.0L;
...
napa_abs_loop = 0.0L;
do {
    napa_abs_time  = napa_abs_loop * 1.0e-6L;


    h_node_s_osc2 = h_node_s_osc1;
    h_node_s_osc1 = h_node_s_osc0;
    h_node_s_osc0 = (h_node_s_factor * h_node_s_osc1) - h_node_s_osc2;
    d_node_s = 2.0 * ((R_TYPE) h_node_s_osc1);
    d_node_s += 1.0;

    napa_abs_loop++;

  } while (!TERMINATE);
...
```

The resonator itself is implemented with long double float (16 bytes) to get the best precision. Output is a double float.

A Description Using a NAPA User Function [*]

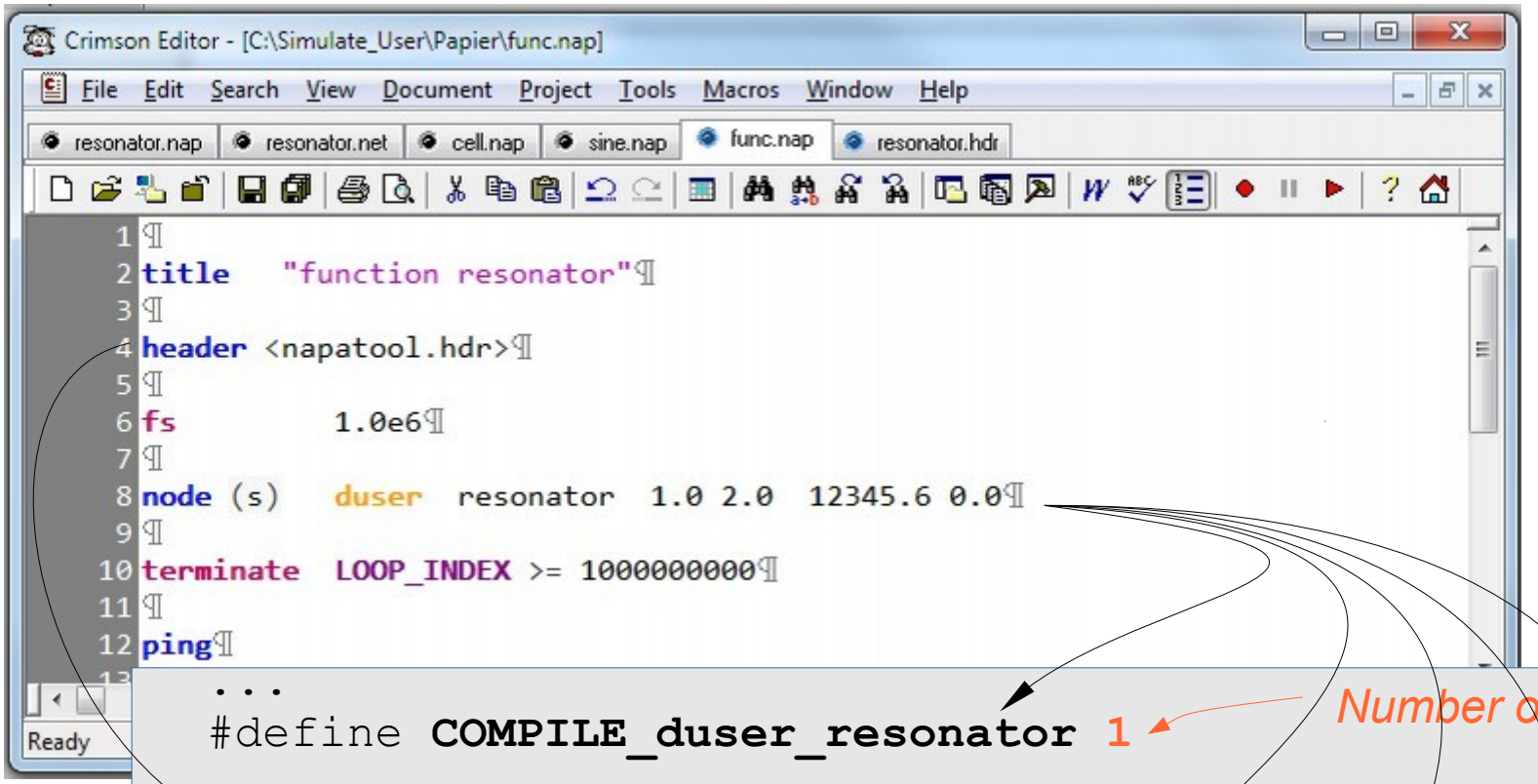
[*] A function you can write yourself in ANSI-C



'header' triggers the inclusion of C code in the simulator

file "/Simulate/Napados/Hdr/Activation/resonator.hdr"

```
A set of C user functions in appropriate NAPA wrappers

    check_duser_resonator_04(a,b,c,d,e)
    reset_duser_resonator_04(a,b,c,d,e)
     init_duser_resonator_04(a,b,c,d,e)
    close_duser_resonator_04(a,b,c,d,e)
          duser_resonator_04(a,b,c,d,e)
```



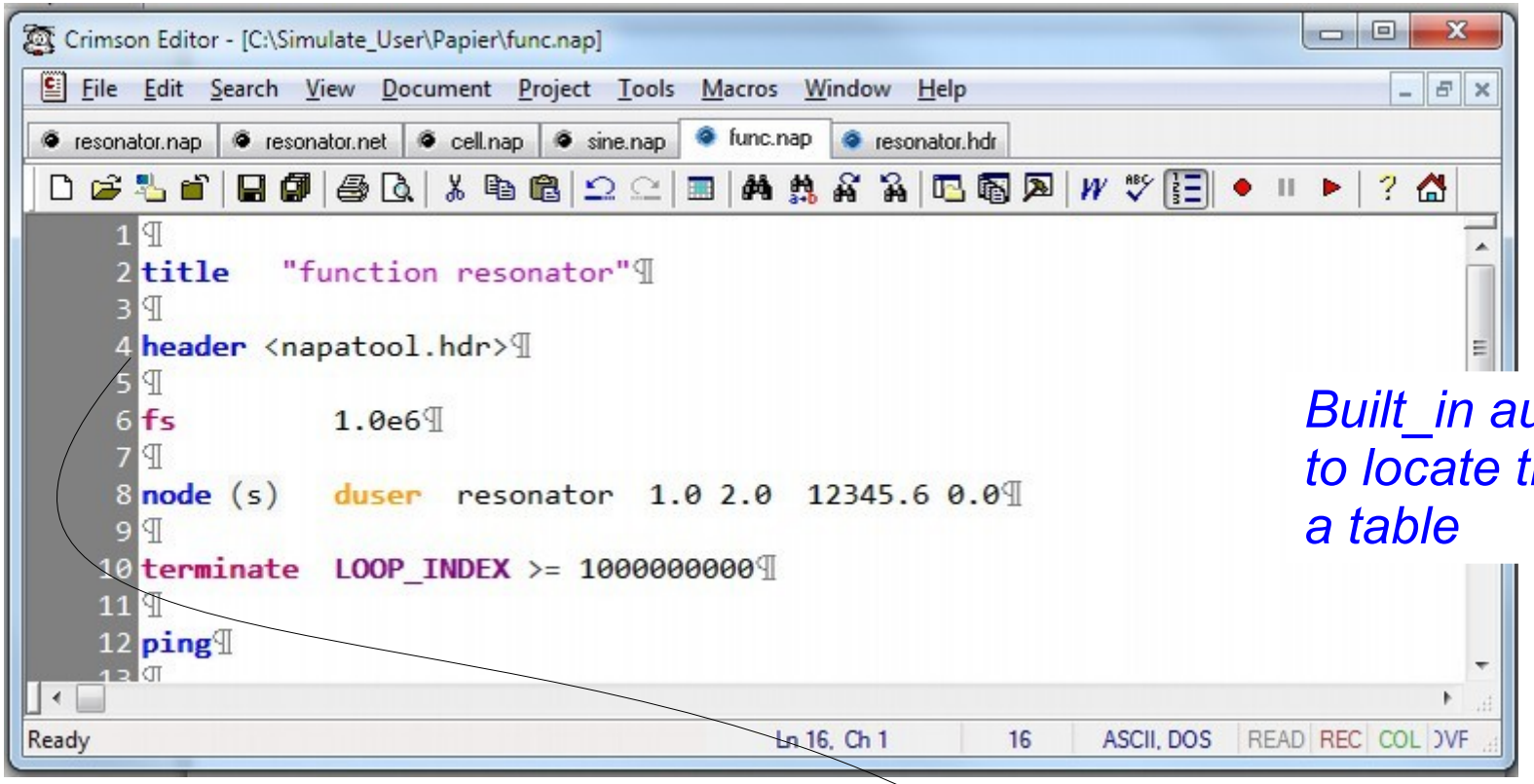
NAPA compilation

file "func.c"

```
...
#define COMPILE_duser_resonator 1
...
#include  "/Simulate/Napados/Hdr/napatool.hdr"
...
check_duser_resonator_04(1.0,2.0,12345.6,0.0, 0);
init_duser_resonator_04(1.0,2.0,12345.6,0.0, 0);
...
do {
  napa_abs_time  = napa_abs_loop * 1.0e-0L;
  d_node_s = duser_resonator_04(1.0,2.0,12345.6,0.0, 0);
  napa_abs_loop++;
} while (!TERMINATE);
...
close_duser_resonator_04(1.0,2.0,12345.6,0.0, 0);
...
```

Number of instances in netlist

Instance ID



Built_in automatic method to locate the function using a table

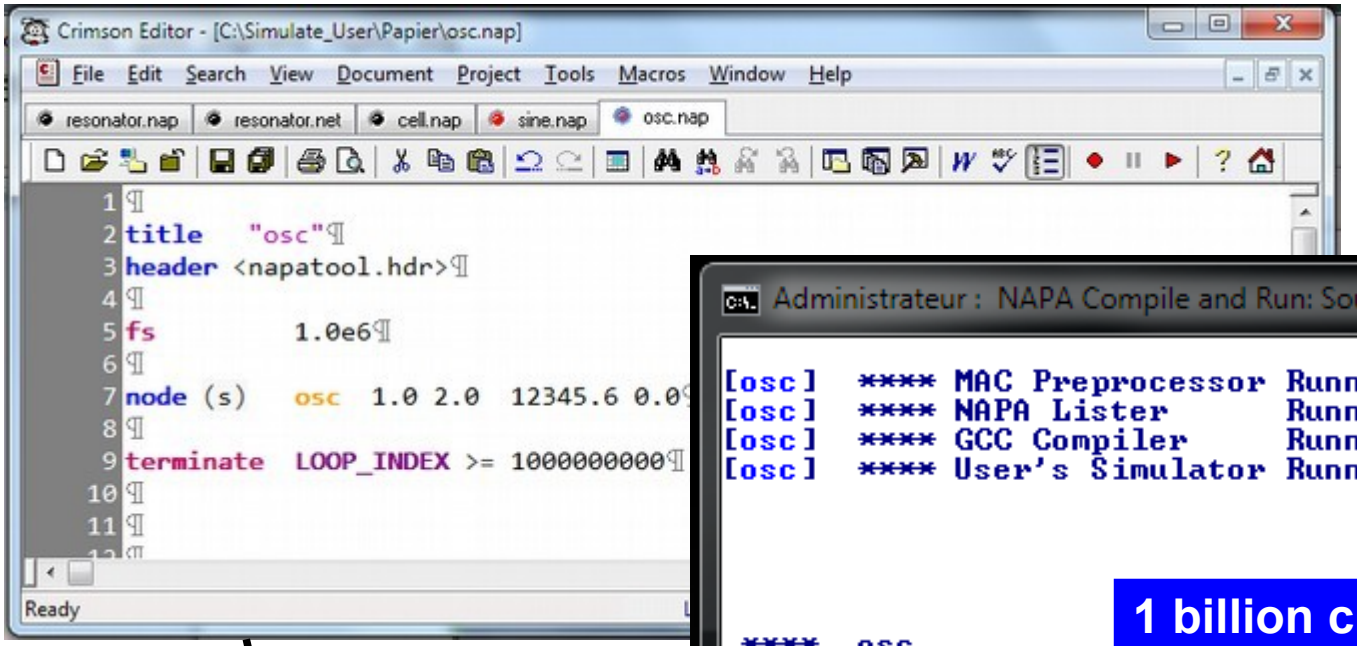
In a file included in "/Simulate/Napados/Hdr/napatool.hdr"

```
...
#ifdef COMPILE_duser_resonator
# include "/Simulate/Napados/hdr/Activation/resonator.hdr"
#endif
...
```

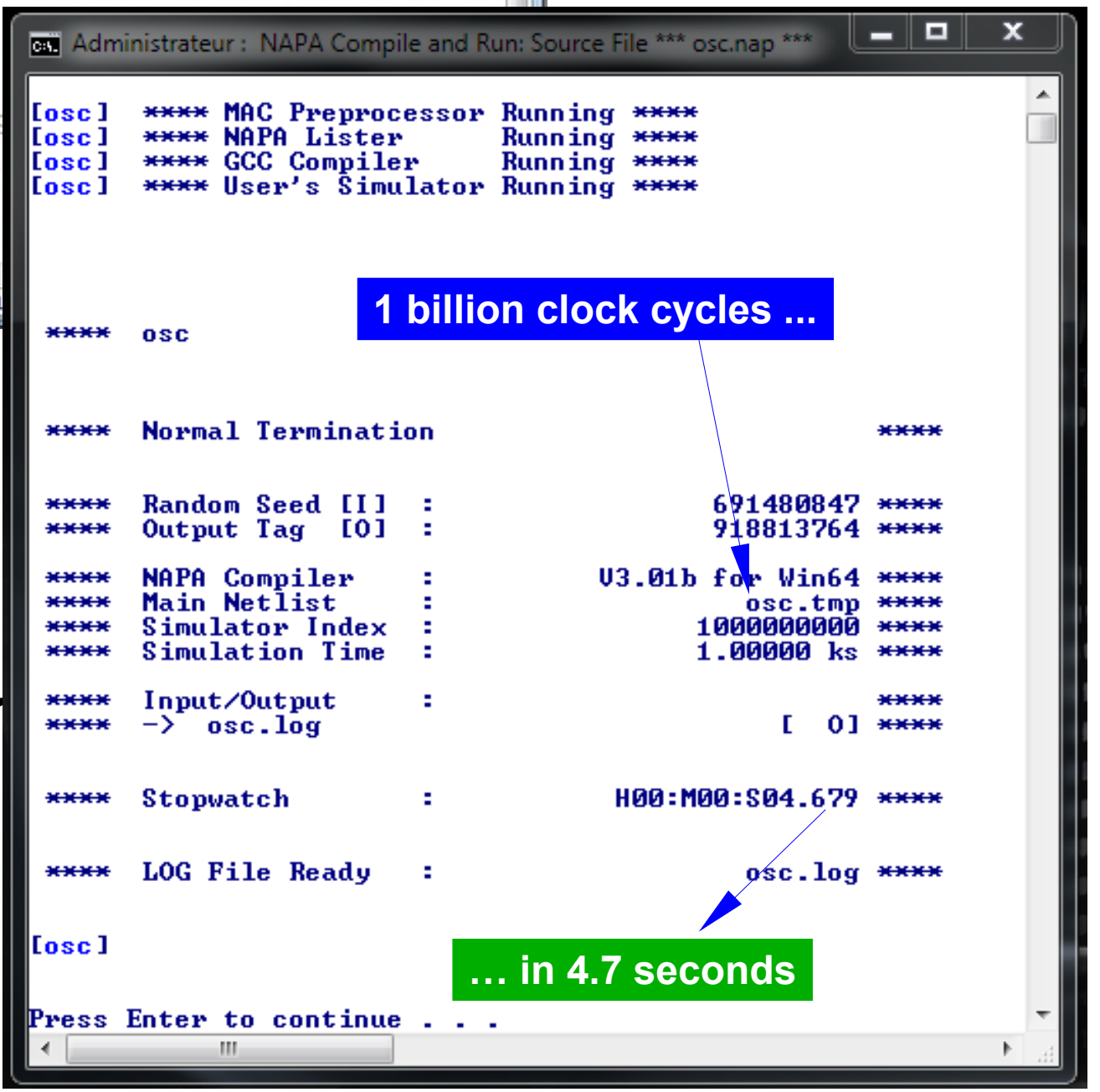
file "/Simulate/Napados/Hdr/Activation/resonator.hdr"

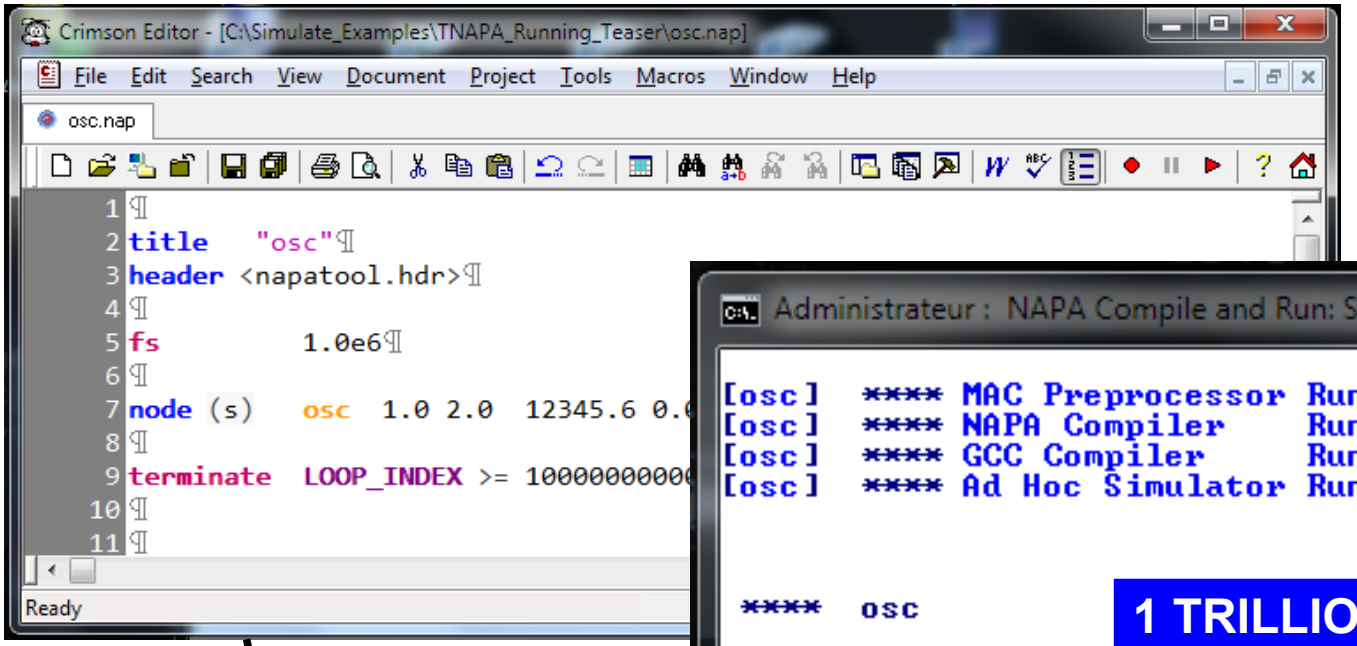
```
    check_duser_resonator_04(a,b,c,d,e) ...
    reset_duser_resonator_04(a,b,c,d,e) ...
     init_duser_resonator_04(a,b,c,d,e) ...
    close_duser_resonator_04(a,b,c,d,e) ...
          duser_resonator_04(a,b,c,d,e) ...
```

We aimed to get speed, didn't we ?

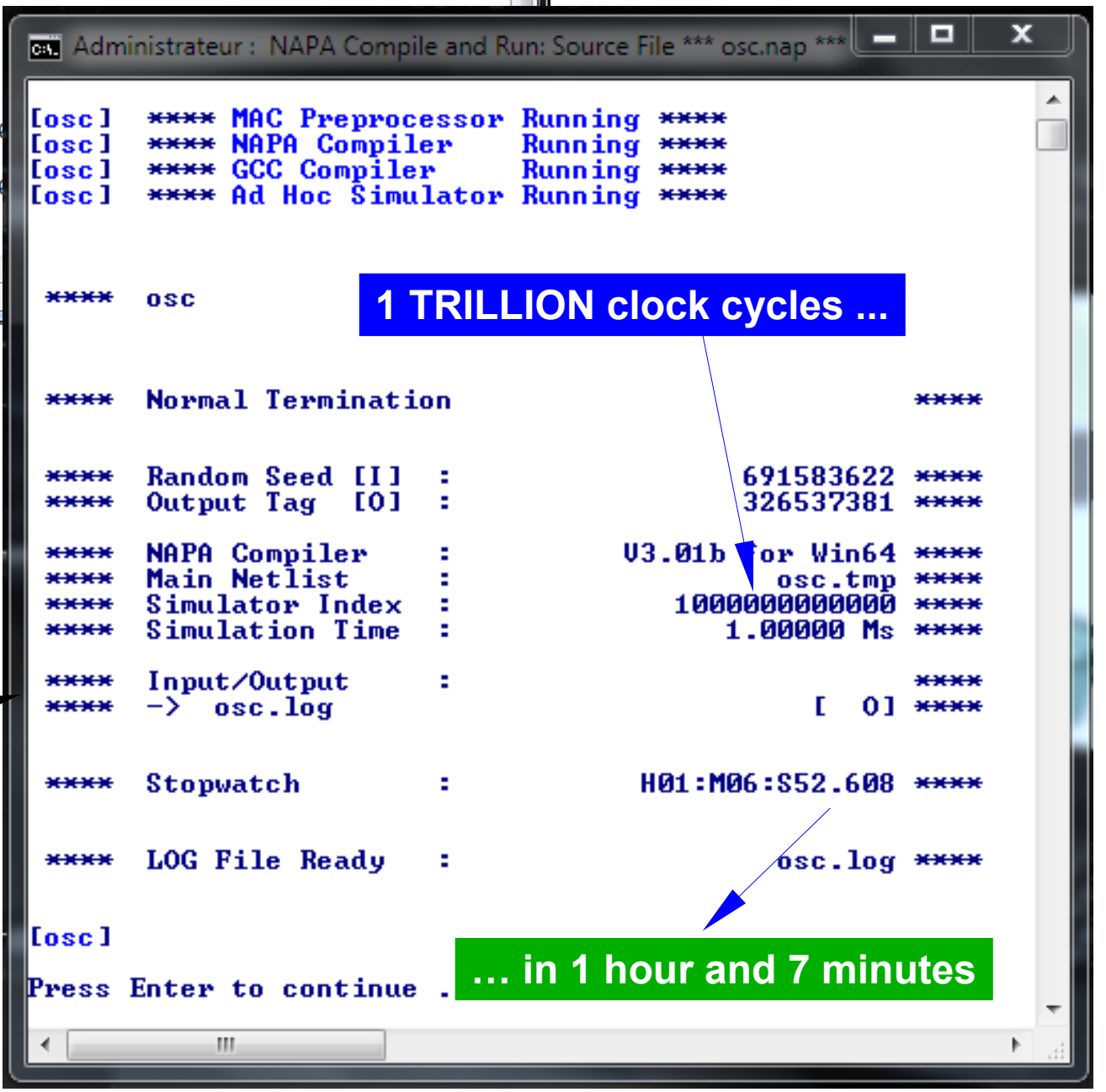


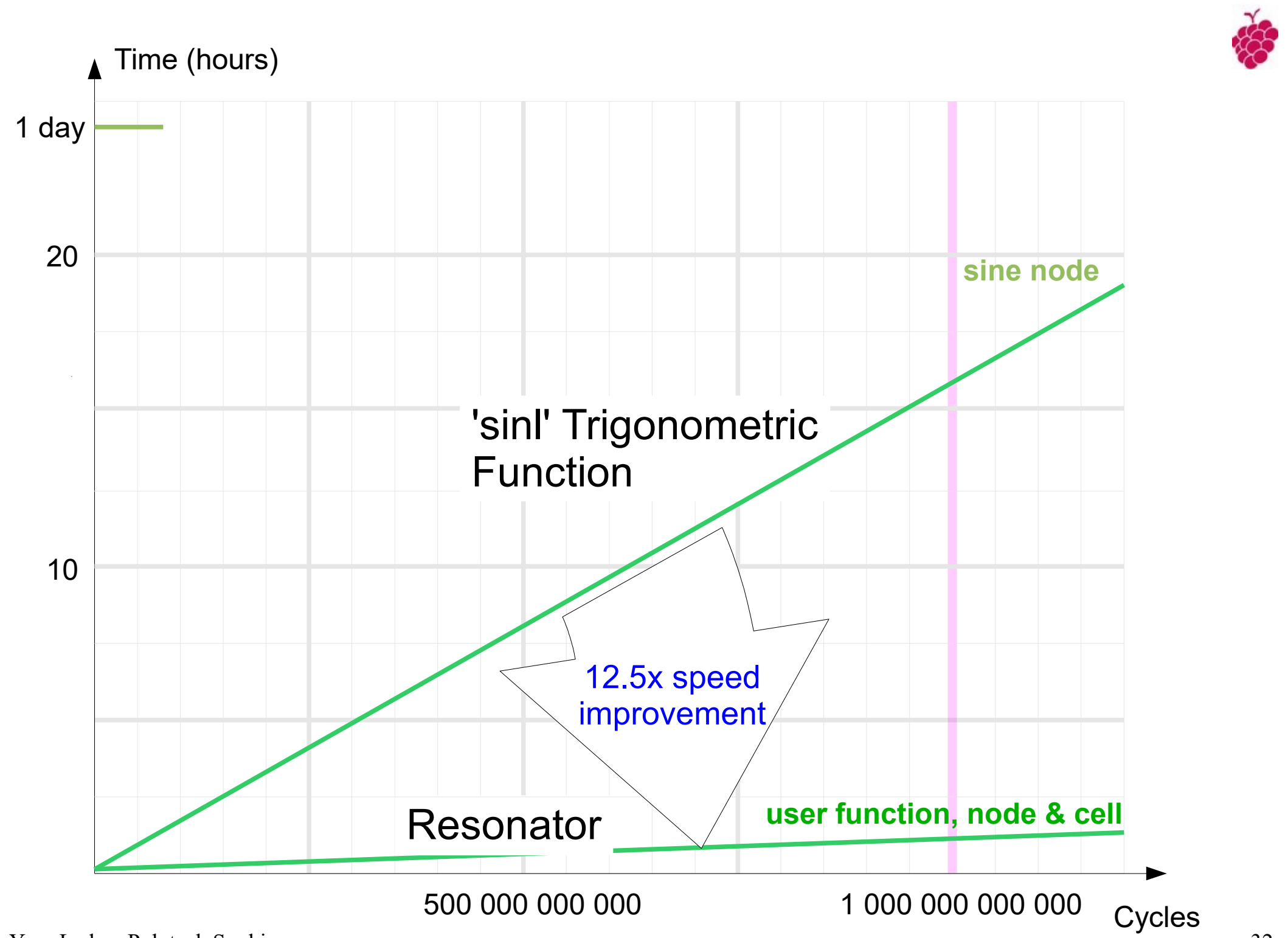
Run Simulation





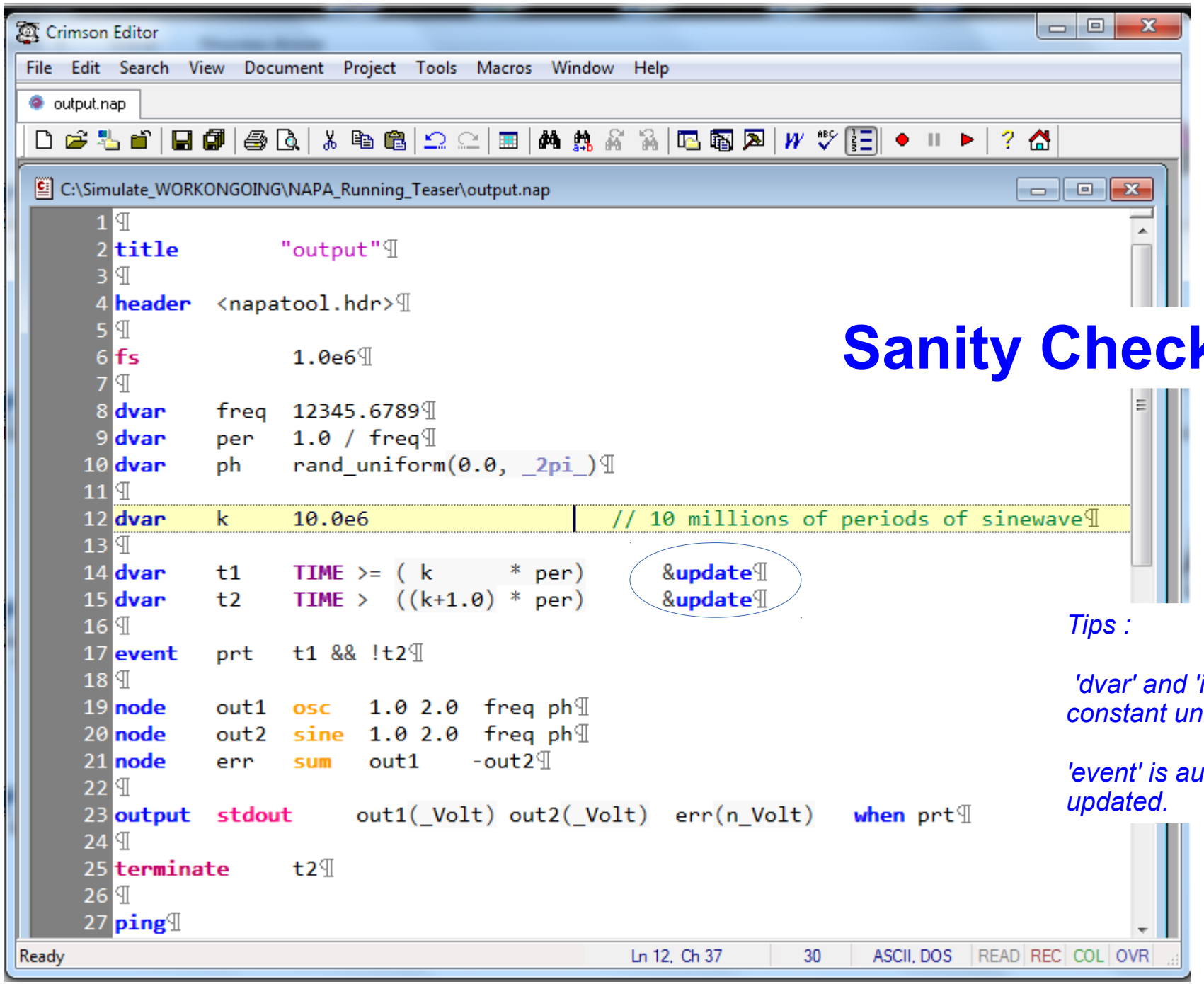
Run Simulation




Time (hours)
1 day
20
10
sine node
'sinl' Trigonometric Function
12.5x speed improvement
Resonator
user function, node & cell
500 000 000 000
1 000 000 000 000
Cycles

Cool...

But we do not want to loose precision, do we ?



Sanity Check

Tips :

'dvar' and 'ivar' are constant unless updated.

'event' is automatically updated.

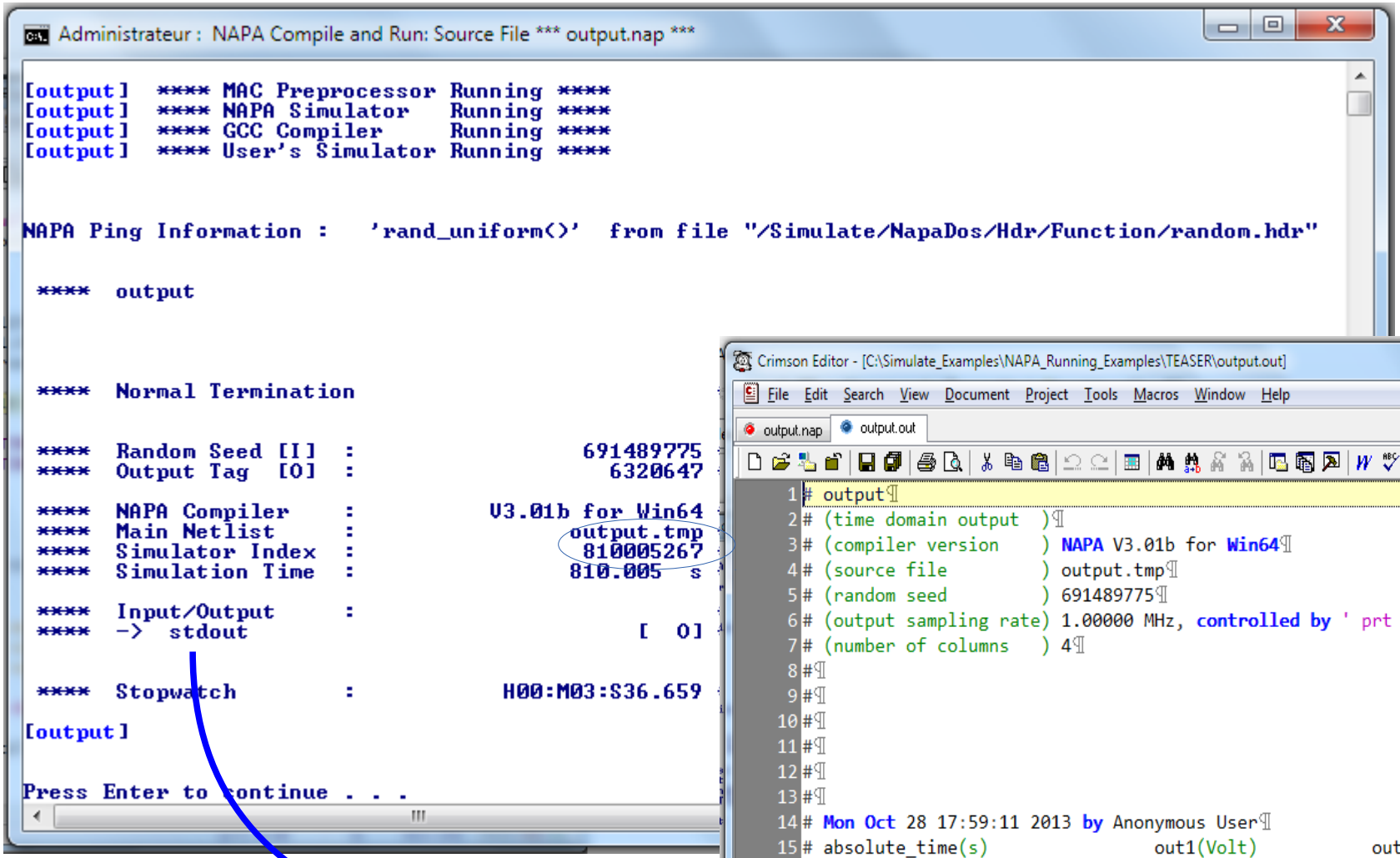
Time domain simulation, output through "stdout"

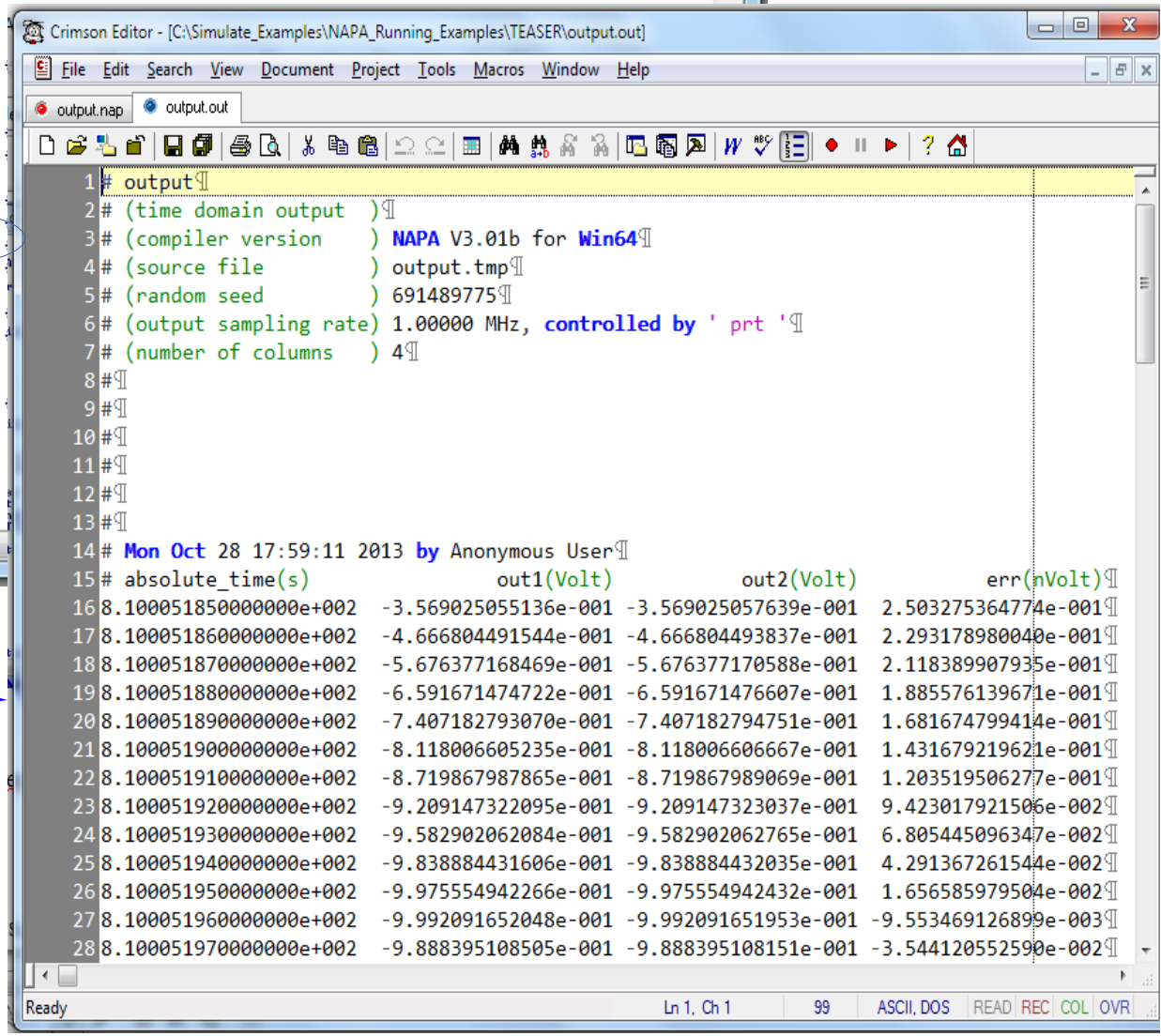
```
...
napa_abs_loop = 0L ;

d_var_freq = 12345.6;
d_var_per = 1.0/d_var_freq;
d_var_ph = rand_uniform(0.0,_2pi_);
d_var_k = 10.0e6;
d_var_t1 = TIME>=(d_var_k*d_var_per);
d_var_t2 = TIME>((d_var_k+1.0)*d_var_per);
i_var_prt = (I_TYPE)(d_var_t1&&!d_var_t2);

h_node_out1_factor = 2.0L * cosl(_2PI_*d_var_freq/((H_PREC) FSL));
h_node_out1_osc0 = sinl(d_var_ph);
h_node_out1_osc1 = sinl(d_var_ph - (_2PI_*d_var_freq/((H_PREC) FSL)));
h_node_out1_osc2 = 0.0L;

do {

   napa_abs_time  = napa_abs_loop * 1.0e-6L;

   d_var_t1 = TIME>=(d_var_k*d_var_per);
   d_var_t2 = TIME>((d_var_k+1.0)*d_var_per);
   i_var_prt = (I_TYPE)(d_var_t1&&!d_var_t2);

   h_node_out1_osc2 = h_node_out1_osc1;
   h_node_out1_osc1 = h_node_out1_osc0;
   h_node_out1_osc0 = (h_node_out1_factor * h_node_out1_osc1) - h_node_out1_osc2;
   d_node_out1 = 2.0 * ((R_TYPE) h_node_out1_osc1);
   d_node_out1 += 1.0;
   d_node_out2 = 1.0 + (2.0) * ((R_TYPE) sinl(_2PI_ * ((H_PREC) d_var_freq) * napa_abs_time + (d_var_ph)));
   d_node_err = (d_node_out1) + (-d_node_out2);

   if (i_var_prt) {
      fprintf(napa_fp_0, "  % .12e % .12e % .12e\n", d_node_out1, d_node_out2, d_node_err*1.0e9);
   }

   napa_abs_loop++;

} while ( ! TERMINATE );
...
```

1. INITIALIZE SIMULATION
2. INITIALIZE VARIABLES
3. INITIALIZE NODES

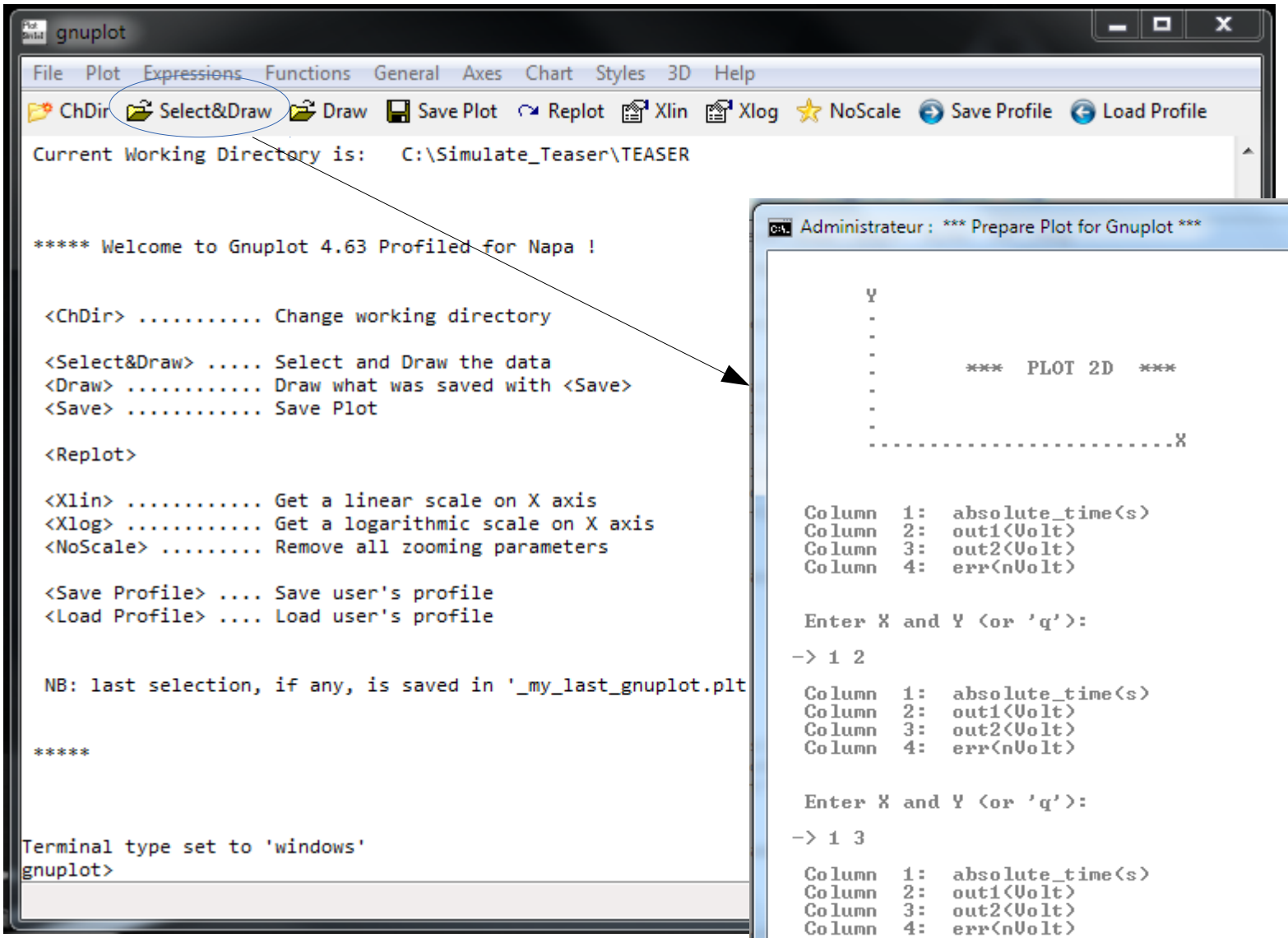
LOOP

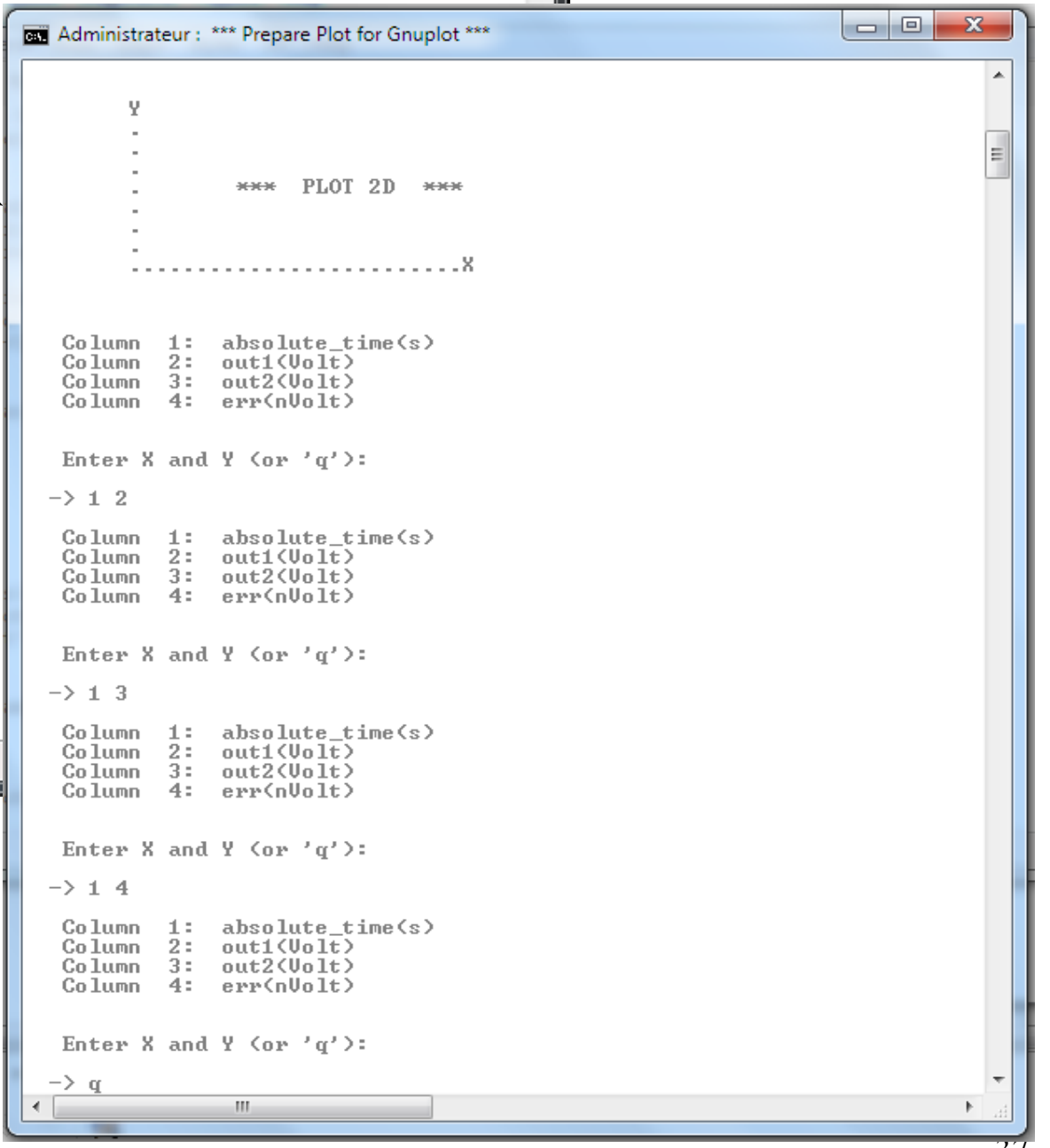
1. VARIABLE UPDATE
2. NODE UPDATE
3. TIME DOMAIN OUPUT





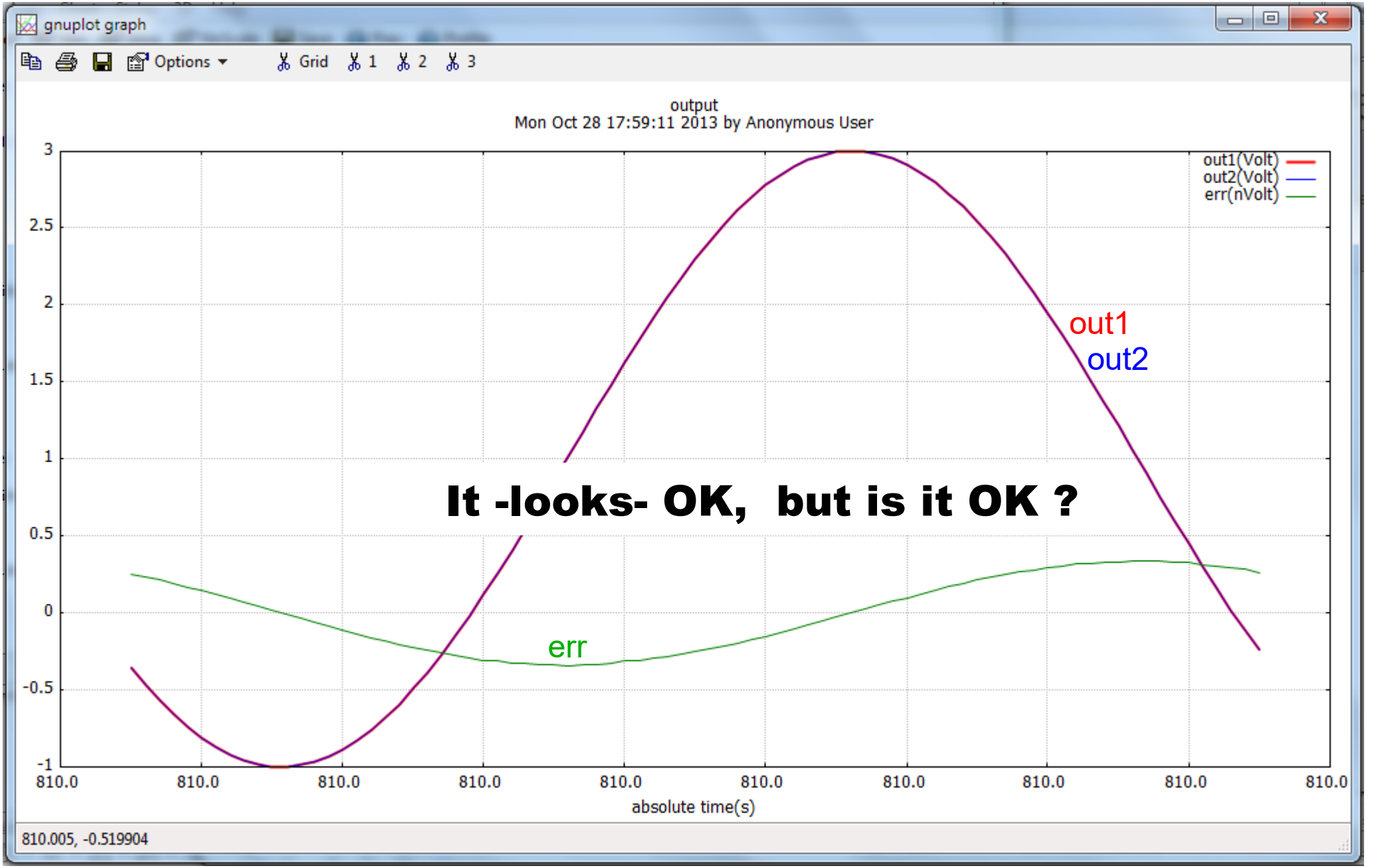
Output redirected from "stdout"





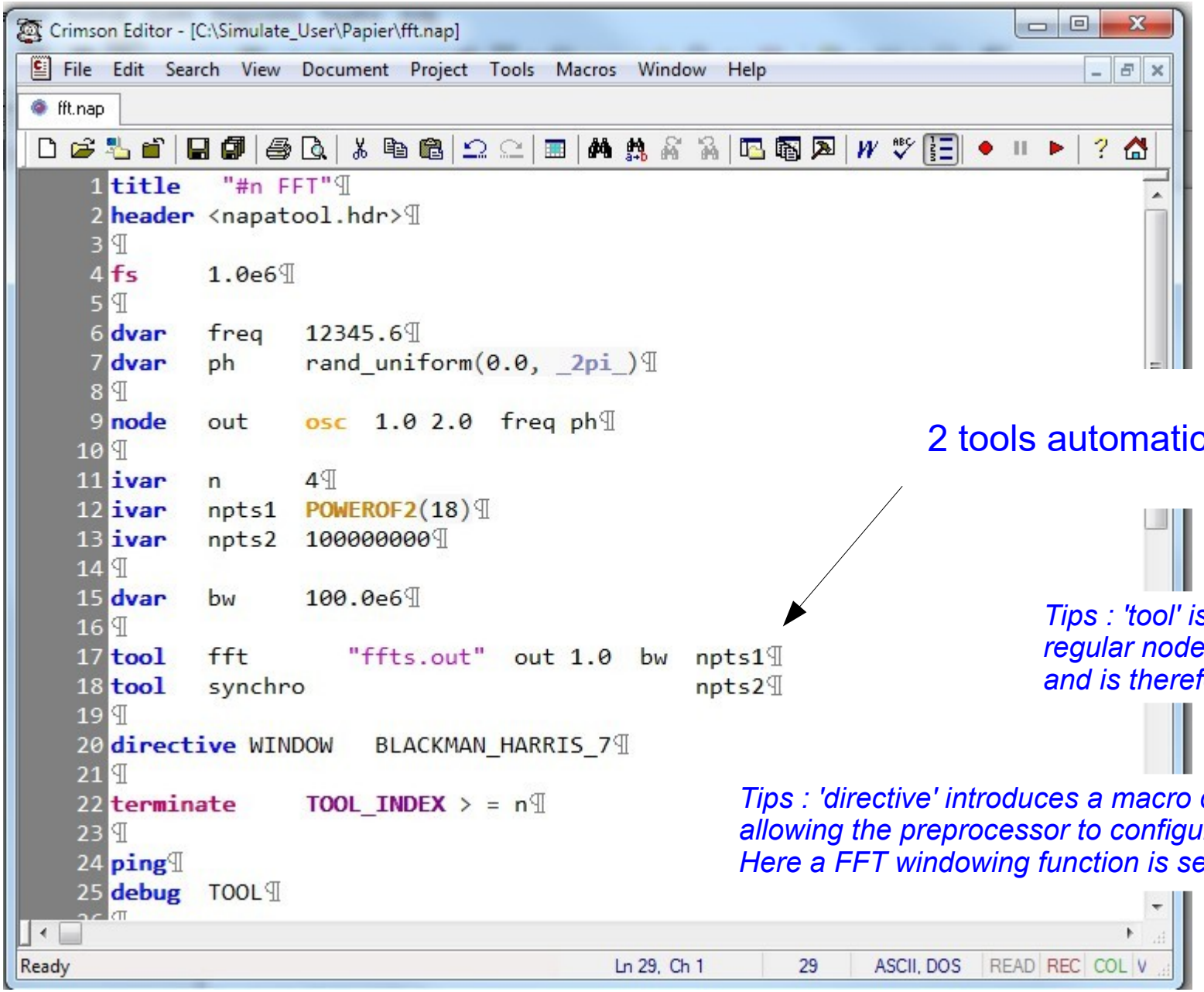
*Tips : to start Gnuplot, press **'Alt G'** from any file related to simulation*

Time Domain Output, 2D Plot



Absolute error between 'sine' and 'osc' < 0.3 nV after 810 millions simulation steps

We are now ready to use the NAPA smart tools !



2 tools automatically synchronised

Tips : 'tool' is a contraction of a regular node syntax: 'node void itool' and is therefore processed as a node

Tips : 'directive' introduces a macro definition in the C code allowing the preprocessor to configure/extend the simulator. Here a FFT windowing function is selected to replace the default.

Analysis : 4 FFT of 2^{18} samples, made every 10^8 samples

How the Smart Tool Synchronization Is Working ?

Tool is a user defined function with a synchronization mechanism automatically hooked to the simulator.

A simple **state machine** is implemented in tools with 3 main states: 'start', 'run', 'wait',

Tasks are numbered. Tools are asked by the simulator to perform a task.
Tools are in waiting state until the simulator is sending a message '**start**'.

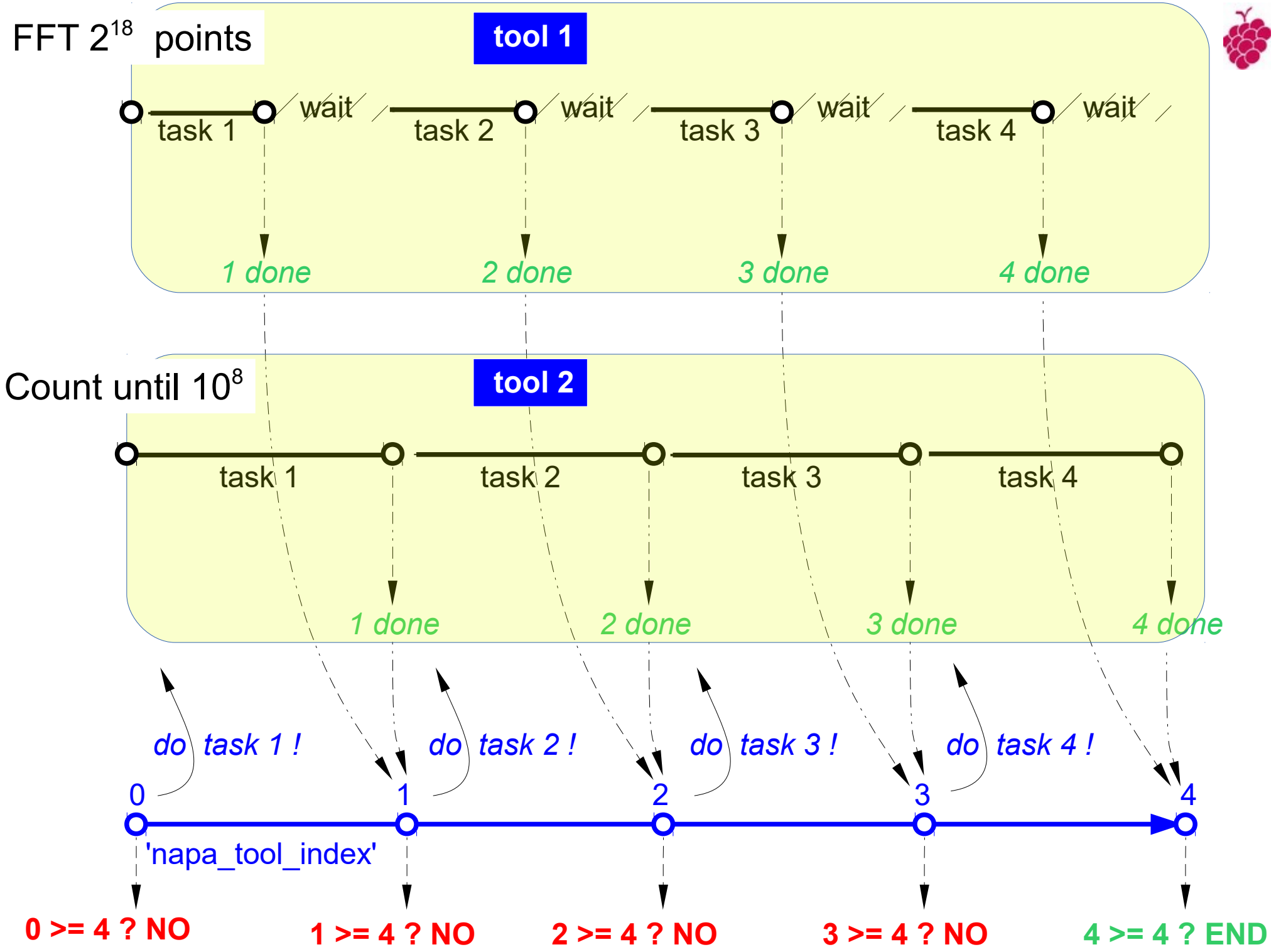
All tools start their own task. The output of the tool is the status of its work.
The simulator collects these status at the end of each simulation cycle.

The simulation continues until all tools have completed the specified task.
A tool having accomplished its task stops and is in **'wait'** state.

When all tools have accomplished their task, the simulator sends a message to all of them to start the next task.

Variable '**napa_tool_index'** handled by the simulator counts the number of tasks already completed and is often used to control the end of the simulation.

(Note : Macro 'TOOL_INDEX' is the image of 'napa_tool_index')


FFT 2^18 points
tool 1
task 1
wait
task 2
wait
task 3
wait
task 4
wait
1 done
2 done
3 done
4 done
Count until 10^8
tool 2
task 1
task 2
task 3
task 4
1 done
2 done
3 done
4 done
do task 1 !
do task 2 !
do task 3 !
do task 4 !
0
1
2
3
4
'napa_tool_index'
0 >= 4 ? NO
1 >= 4 ? NO
2 >= 4 ? NO
3 >= 4 ? NO
4 >= 4 ? END

file "fft.c"

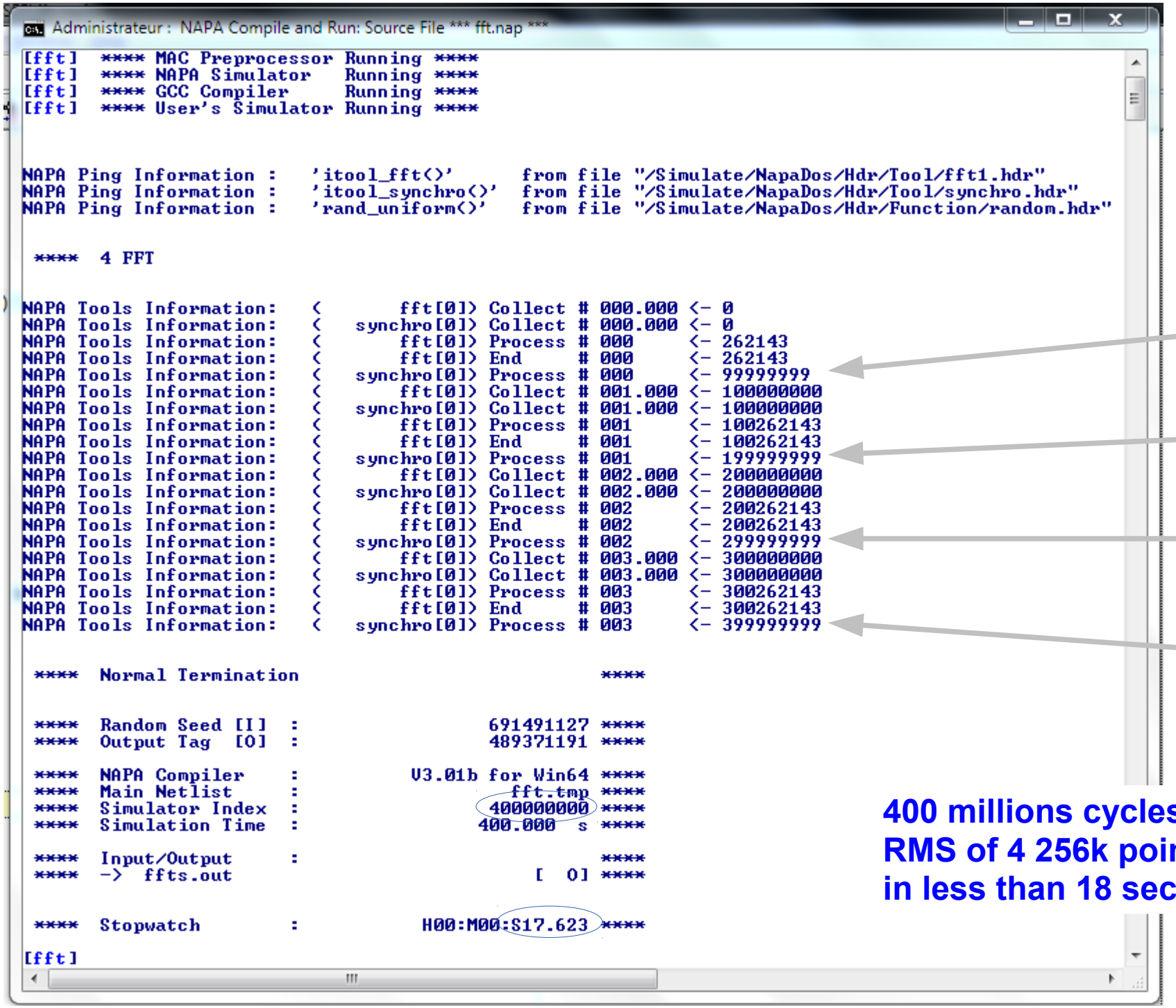
```
...
do {
 napa_rel_time = napa_rel_loop * 1.0e-6L;
 napa_abs_time = napa_abs_loop * 1.0e-6L;

 h_node_out_osc2 = h_node_out_osc1;
 h_node_out_osc1 = h_node_out_osc0;
 h_node_out_osc0 = (h_node_out_factor * h_node_out_osc1) - h_node_out_osc2;
 d_node_out = 2.0 * ((R_TYPE) h_node_out_osc1);
 d_node_out += 1.0;

 napa_msg = &(napa_mailbox[0]);
 napa_msg->o = napa_packet;
 i_node__void0 = itool_fft_05("ffts.out",d_node_out,1.0,d_var_bw,i_var_npts1, 0);

 napa_msg = &(napa_mailbox[1]);
 napa_msg->o = napa_packet;
 i_node__void1 = itool_synchro_01(i_var_npts2, 0);

 if ((napa_mailbox[0].o >= napa_packet) && (napa_mailbox[1].o >= napa_packet)) {
   napa_rel_loop   = -1.0L;
   napa_tool_index = napa_packet;
   napa_mailbox[0].i = START;
   napa_mailbox[1].i = START;
   napa_packet++;
 }
 napa_rel_loop++;
 napa_abs_loop++;

} while (!TERMINATE);
...
```

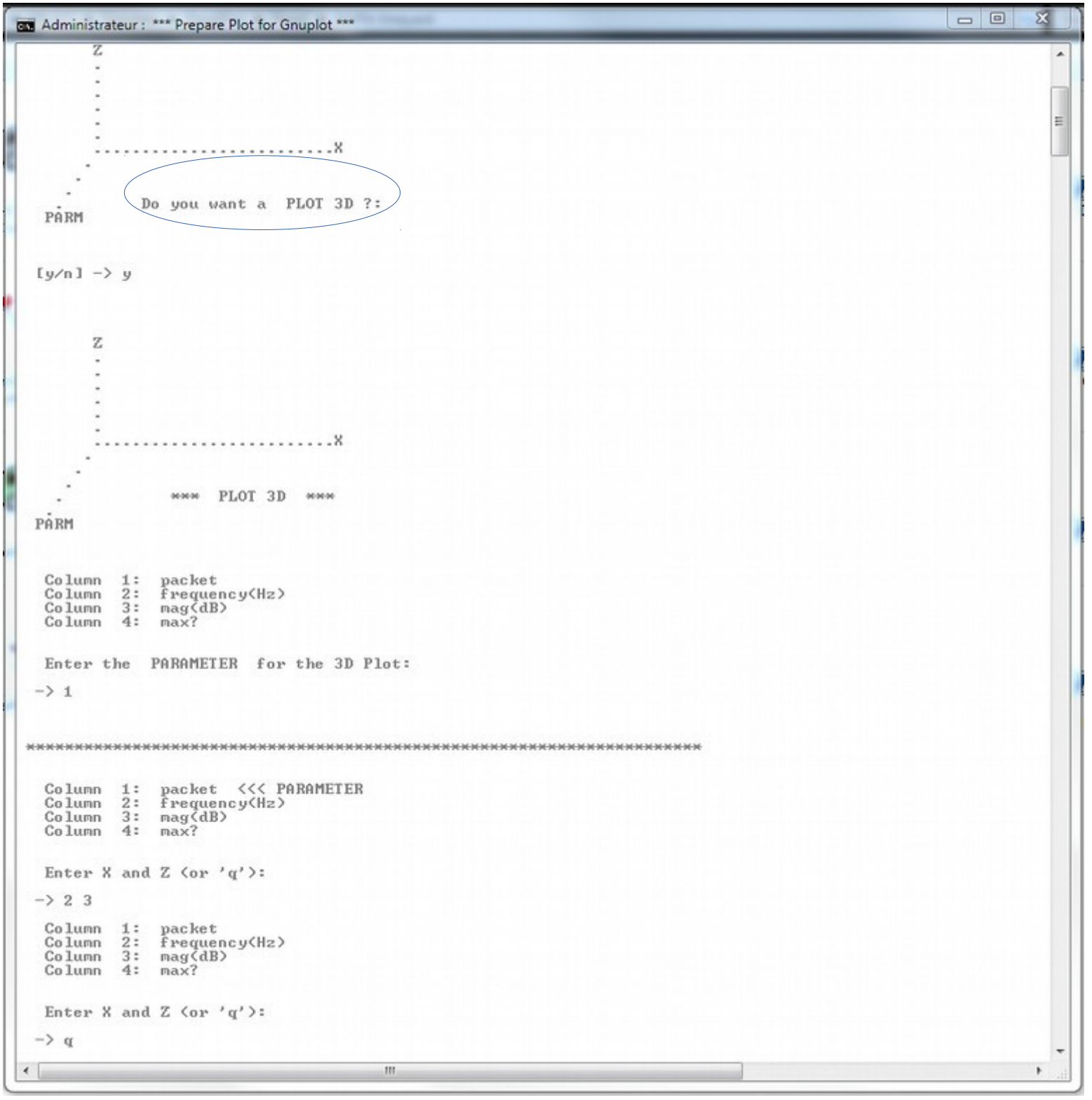
The simulator prefills the individual mailbox output

The simulator prefills the individual mailbox output

The simulator tests the content of the mailboxes output which contains the answer of the tools and reacts accordingly

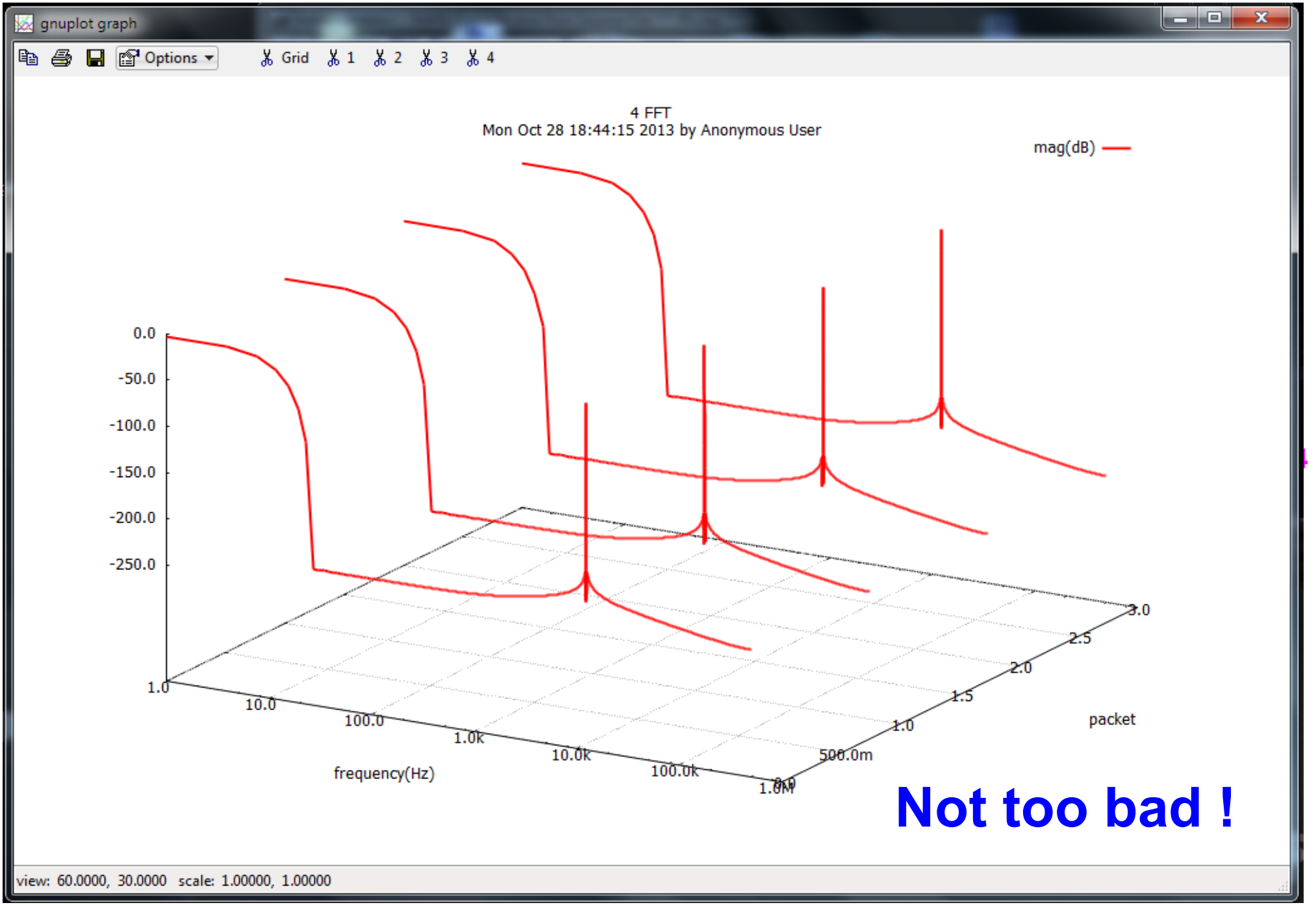
The macro 'TERMINATE' checks 'napa_tool_index'


Administrateur : NAPA Compile and Run: Source File *** fft.nap ***
[fft] **** MAC Preprocessor Running ****
[fft] **** NAPA Simulator Running ****
[fft] **** GCC Compiler Running ****
[fft] **** User's Simulator Running ****
NAPA Ping Information : 'itool_fft()' from file "/Simulate/NapaDos/Hdr/Tool/fft1.hdr"
NAPA Ping Information : 'itool_synchro()' from file "/Simulate/NapaDos/Hdr/Tool/synchro.hdr"
NAPA Ping Information : 'rand_uniform()' from file "/Simulate/NapaDos/Hdr/Function/random.hdr"
**** 4 FFT
NAPA Tools Information: (fft[0]) Collect # 000.000 <- 0
NAPA Tools Information: (synchro[0]) Collect # 000.000 <- 0
NAPA Tools Information: (fft[0]) Process # 000 <- 262143
NAPA Tools Information: (fft[0]) End # 000 <- 262143
NAPA Tools Information: (synchro[0]) Process # 000 <- 99999999
NAPA Tools Information: (fft[0]) Collect # 001.000 <- 100000000
NAPA Tools Information: (synchro[0]) Collect # 001.000 <- 100000000
NAPA Tools Information: (fft[0]) Process # 001 <- 100262143
NAPA Tools Information: (fft[0]) End # 001 <- 100262143
NAPA Tools Information: (synchro[0]) Process # 001 <- 199999999
NAPA Tools Information: (fft[0]) Collect # 002.000 <- 200000000
NAPA Tools Information: (synchro[0]) Collect # 002.000 <- 200000000
NAPA Tools Information: (fft[0]) Process # 002 <- 200262143
NAPA Tools Information: (fft[0]) End # 002 <- 200262143
NAPA Tools Information: (synchro[0]) Process # 002 <- 299999999
NAPA Tools Information: (fft[0]) Collect # 003.000 <- 300000000
NAPA Tools Information: (synchro[0]) Collect # 003.000 <- 300000000
NAPA Tools Information: (fft[0]) Process # 003 <- 300262143
NAPA Tools Information: (fft[0]) End # 003 <- 300262143
NAPA Tools Information: (synchro[0]) Process # 003 <- 399999999
**** Normal Termination ****
**** Random Seed [I] : 691491127 ****
**** Output Tag [O] : 489371191 ****
**** NAPA Compiler : V3.01b for Win64 ****
**** Main Netlist : fft.tmp ****
**** Simulator Index : 400000000 ****
**** Simulation Time : 400.000 s ****
**** Input/Output : ****
**** -> ffts.out [0] ****
**** Stopwatch : H00:M00:S17.623 ****
[fft]
FFT 1
FFT 2
FFT 3
FFT 4
400 millions cycles,
RMS of 4 256k points FFT,
in less than 18 seconds

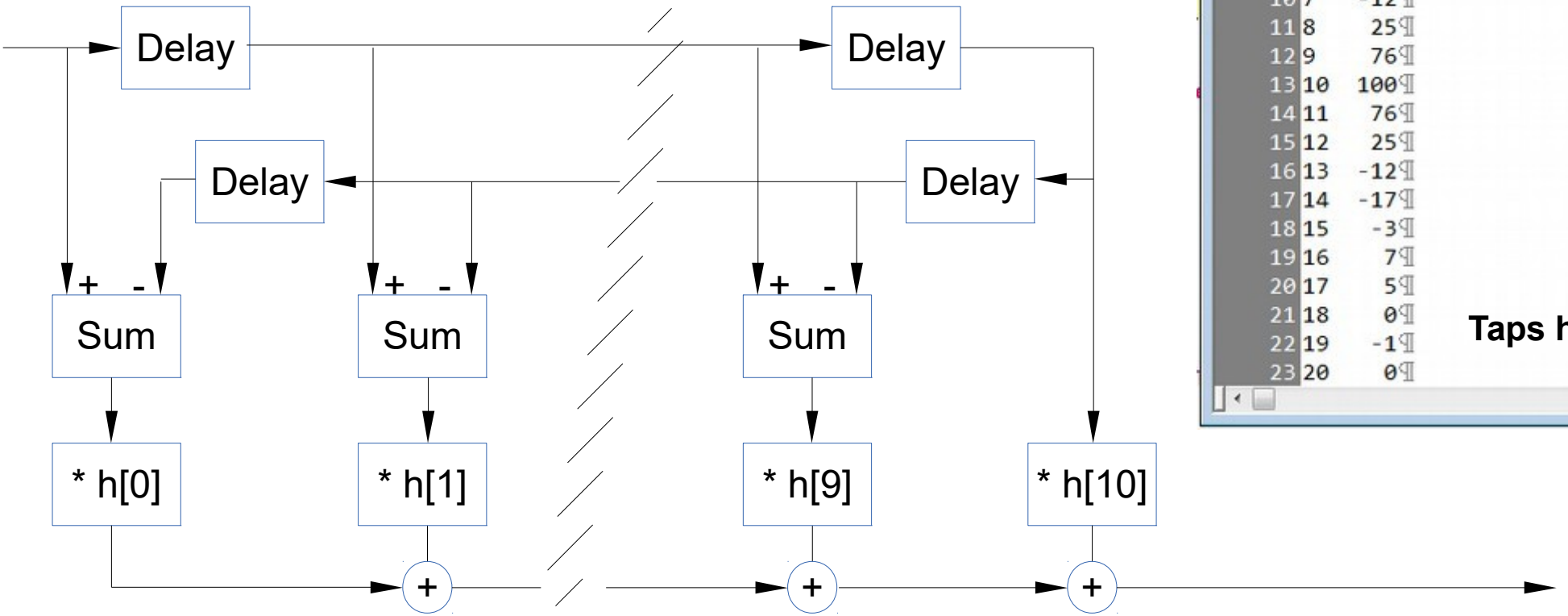

Administrateur : *** Prepare Plot for Gnuplot ***
Z
X
Do you want a PLOT 3D ?:
PARM
[y/n] -> y
Z
X
*** PLOT 3D ***
PARM
Column 1: packet
Column 2: frequency(Hz)
Column 3: mag(dB)
Column 4: max?
Enter the PARAMETER for the 3D Plot:
-> 1
****
Column 1: packet <<< PARAMETER
Column 2: frequency(Hz)
Column 3: mag(dB)
Column 4: max?
Enter X and Z (or 'q'):
-> 2 3
Column 1: packet
Column 2: frequency(Hz)
Column 3: mag(dB)
Column 4: max?
Enter X and Z (or 'q'):
-> q

3D Plot

Frequency Domain Output, 3D Plot



Now a Realistic Example

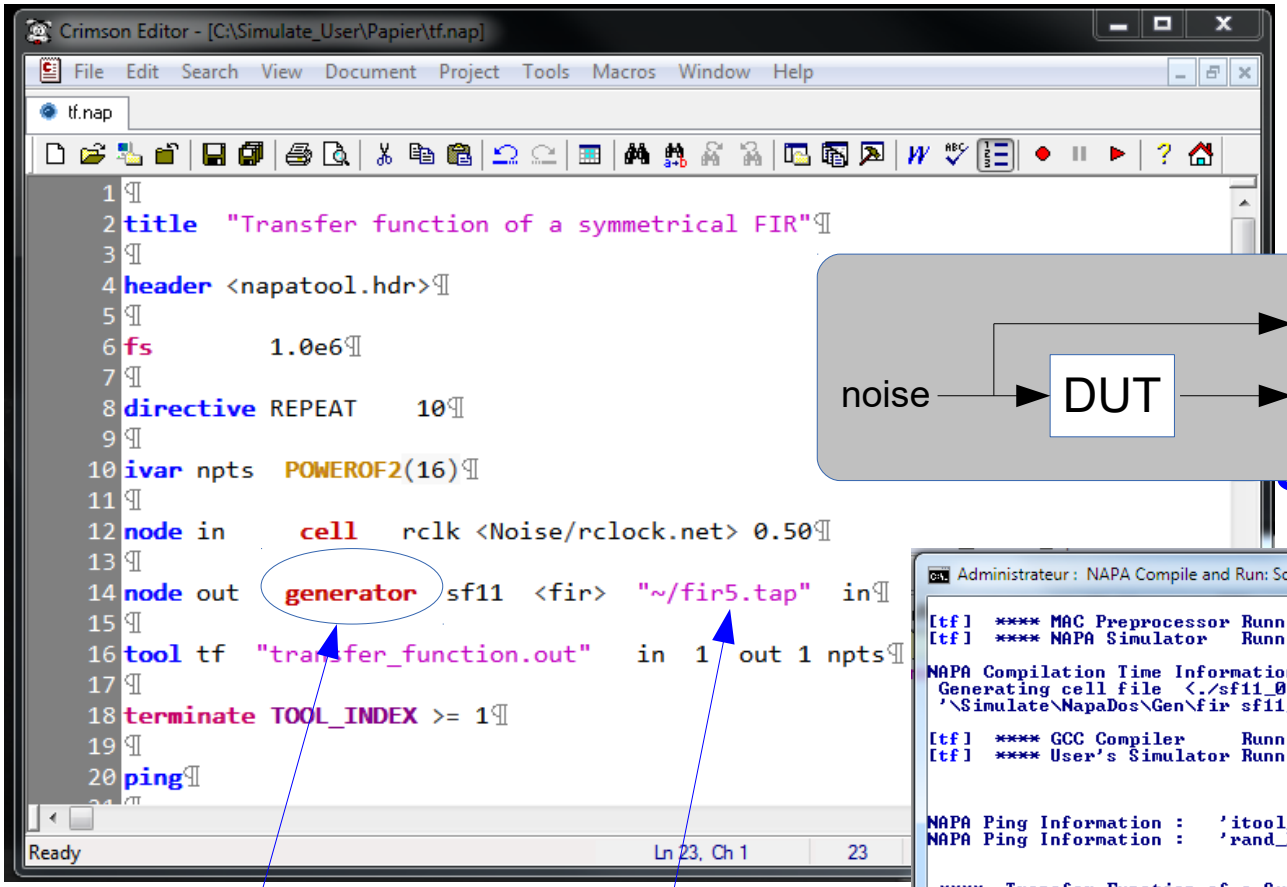


C:\Simulate_User\Papier\fir5.tap

```
1  ## Digital Filter¶
2  #¶
3  0      0¶
4  1     -1¶
5  2      0¶
6  3      5¶
7  4      7¶
8  5     -3¶
9  6    -17¶
10 7    -12¶
11 8     25¶
12 9     76¶
13 10   100¶
14 11    76¶
15 12    25¶
16 13   -12¶
17 14   -17¶
18 15    -3¶
19 16     7¶
20 17     5¶
21 18     0¶
22 19    -1¶
23 20     0¶
```

Taps h[i]

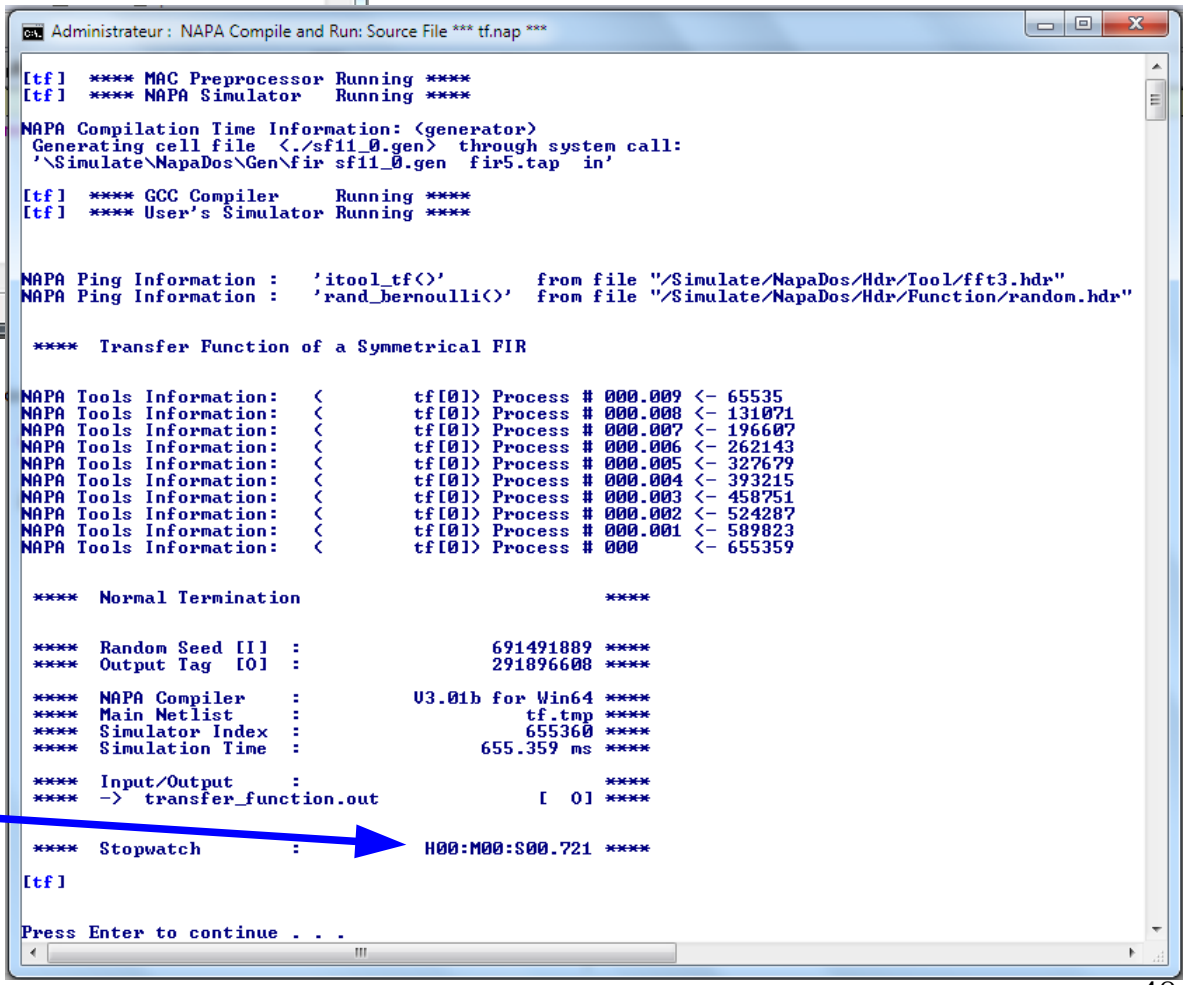
Digital Folded FIR



Digital FIR
Cell generator

Transfer function

RMS of FFTs



On the fly cell generation
using a file containing the taps

0.7 second

Taps

C:\Simulate_User\Papier\fir5.tap

```
## Digital Filter
#
0      0
1     -1
2      0
3      5
4      7
5     -3
6    -17
7    -12
8     25
9     76
10   100
11    76
12    25
13   -12
14   -17
15    -3
16     7
17     5
18     0
19    -1
20     0
```

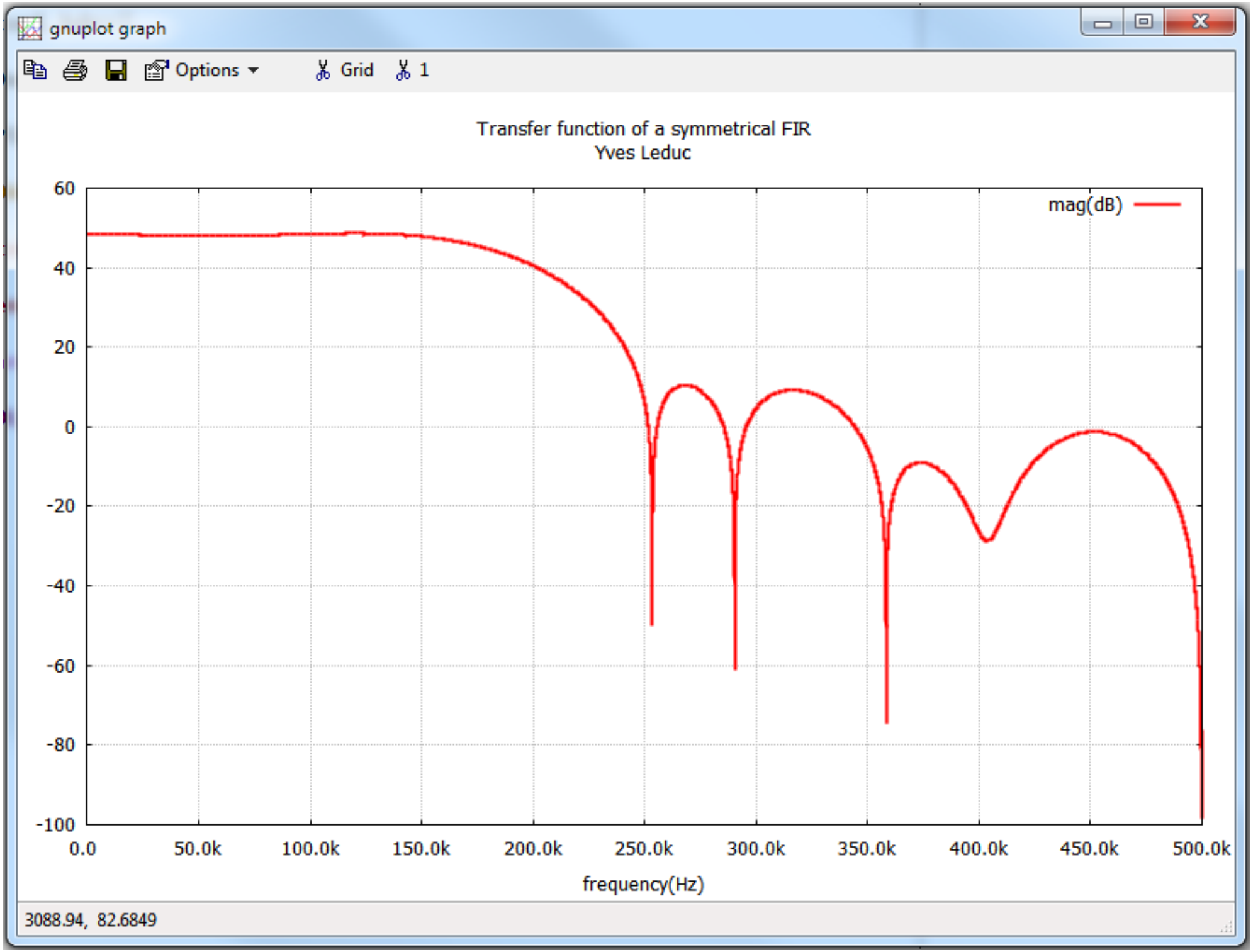
```
node out0   generator  sf11  <fir>  "~/fir5.tap"  in
```

*generator '**fir**'*

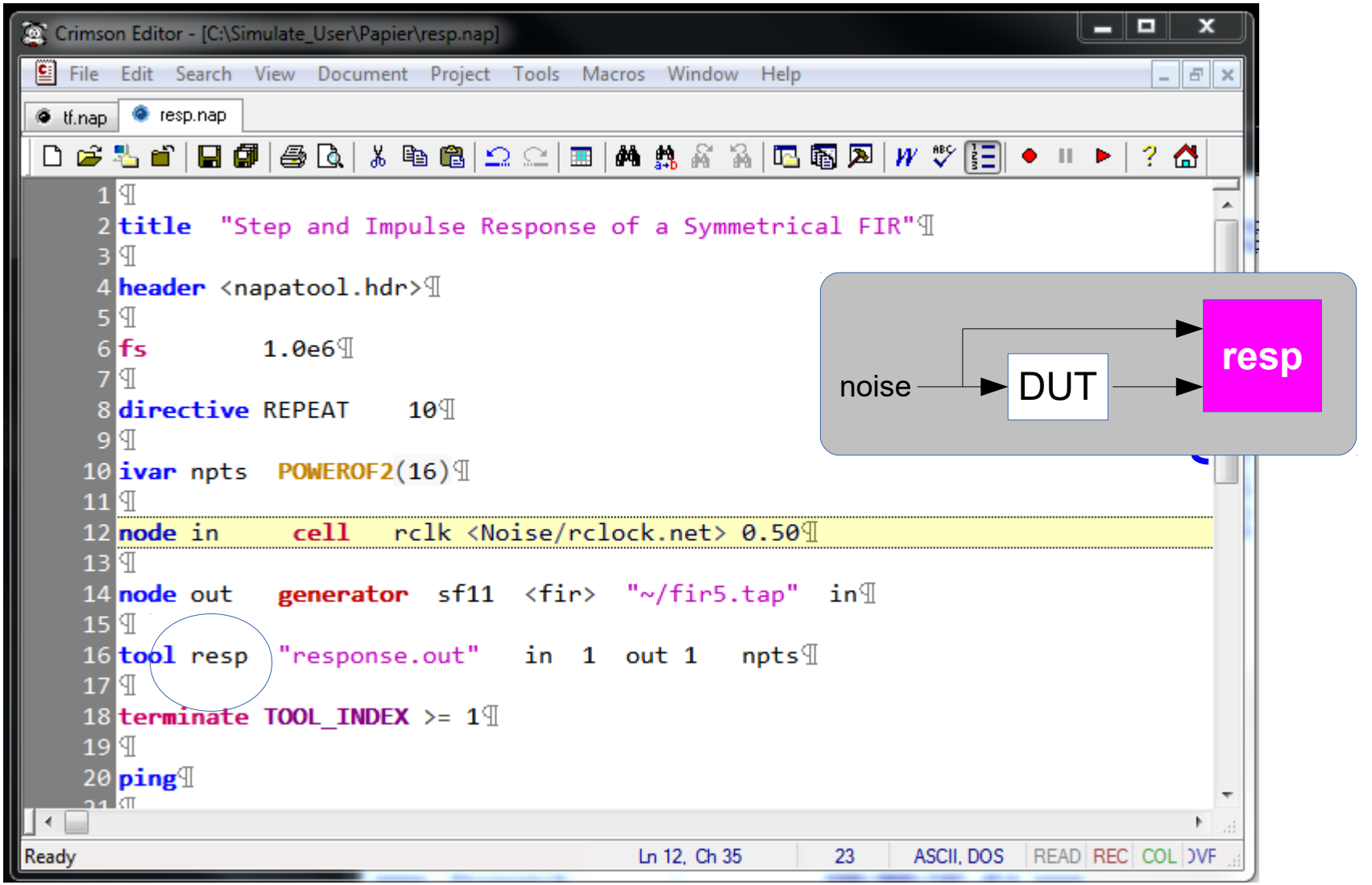
*Cell produced on the fly by generator '**fir**'*

C:\Simulate_User\Papier\sf11_0.gen

```
cell interface  $y  $dummy  $x

# Symmetrical Digital FIR Structure
# Taps File Name: 'fir5.tap'
# Creation Time :  Thu Sep 19 11:33:23 2013
# By Command    :  \Simulate\Napados\Gen\fir
# Generator Version 3.17

# FOLDED FIR
#           ______                                  ______
#          |      |    d1               d9         |      |    d10
# x -+----| Delay >----+-------- .. ----+----| Delay >----+
#    |     |______|    |                |    |______|     |
#    |                 |                |                 |
#    |   d20  |      |  |  d19           |  d11  |      |  |
#    |    +--< Delay |-----+---- .. --------+--< Delay |--+
#    |    |    |______|  |   |           |   |  |______|  |
#   _|____|_            _|___|_        _|___|_            |
#  |(+) (+)|           |(+) (+)|      |(+) (+)|           |
#  |       |           |       |      |       |           |
#  |  Sum  |           |  Sum  |      |  Sum  |           |
#  |_______|           |_______|      |_______|           |
#      | s0                | s1           | s9            |
#   ___|___             ___|___        ___|___         ___|___
#  |       |           |       |      |       |       |       |
#  |  *h0  |           |  *h1  |      |  *h9  |       |  *h10 |
#  |_______|           |_______|      |_______|       |_______|
#      | g0                | g1           | g9            | g10
#      +---------------(+)----- .. -----(+)------------(+)--> y

ivar $h0      0
ivar $h1     -1
ivar $h2      0
ivar $h3      5
ivar $h4      7
ivar $h5     -3
ivar $h6    -17
ivar $h7    -12
ivar $h8     25
ivar $h9     76
ivar $h10   100

node $d1      delay $x
node $d2      delay $d1
node $d3      delay $d2
```


gnuplot graph
Options
Grid
1
Transfer function of a symmetrical FIR
Yves Leduc
mag(dB)
60
40
20
0
-20
-40
-60
-80
-100
0.0
50.0k
100.0k
150.0k
200.0k
250.0k
300.0k
350.0k
400.0k
450.0k
500.0k
frequency(Hz)
3088.94, 82.6849

Step and Impulse Response ?



Administrateur : NAPA Compile and Run: Source File *** resp.nap ***

```
[resp]  **** MAC Preprocessor Running ****
[resp]  **** NAPA Simulator   Running ****

NAPA Compilation Time Information: (generator)
 Generating cell file  <./sf11_0.gen>  through system call:
 '\Simulate\NapaDos\Gen\fir sf11_0.gen  fir5.tap  in'

[resp]  **** GCC Compiler     Running ****
[resp]  **** User's Simulator Running ****


NAPA Ping Information :   'itool_resp()'      from file "/Simulate/Na
NAPA Ping Information :   'rand_bernoulli()'  from file "/Simulate/Na

 ****  Step and Impulse Response of a Symmetrical FIR

NAPA Tools Information:   (      resp[0]) Process # 000.009 <- 65535
NAPA Tools Information:   (      resp[0]) Process # 000.008 <- 131071
NAPA Tools Information:   (      resp[0]) Process # 000.007 <- 196607
NAPA Tools Information:   (      resp[0]) Process # 000.006 <- 262143
NAPA Tools Information:   (      resp[0]) Process # 000.005 <- 327679
NAPA Tools Information:   (      resp[0]) Process # 000.004 <- 393215
NAPA Tools Information:   (      resp[0]) Process # 000.003 <- 458751
NAPA Tools Information:   (      resp[0]) Process # 000.002 <- 524287
NAPA Tools Information:   (      resp[0]) Process # 000.001 <- 589823
NAPA Tools Information:   (      resp[0]) Process # 000     <- 655359

 ****  Normal Termination                          ****

 ****  Random Seed [I]  :              691492081 ****
 ****  Output Tag  [O]  :              159700008 ****

 ****  NAPA Compiler    :        V3.01b for Win64 ****
 ****  Main Netlist     :               resp.tmp ****
 ****  Simulator Index  :                 655360 ****
 ****  Simulation Time  :          655.359 ms     ****

 ****  Input/Output     :                         ****
 ****  ->  response.out                 [  0] ****

 ****  Stopwatch        :        H00:M00:S00.909 ****

[resp]
```

0.9 second

gnuplot graph

Options Grid 1 2

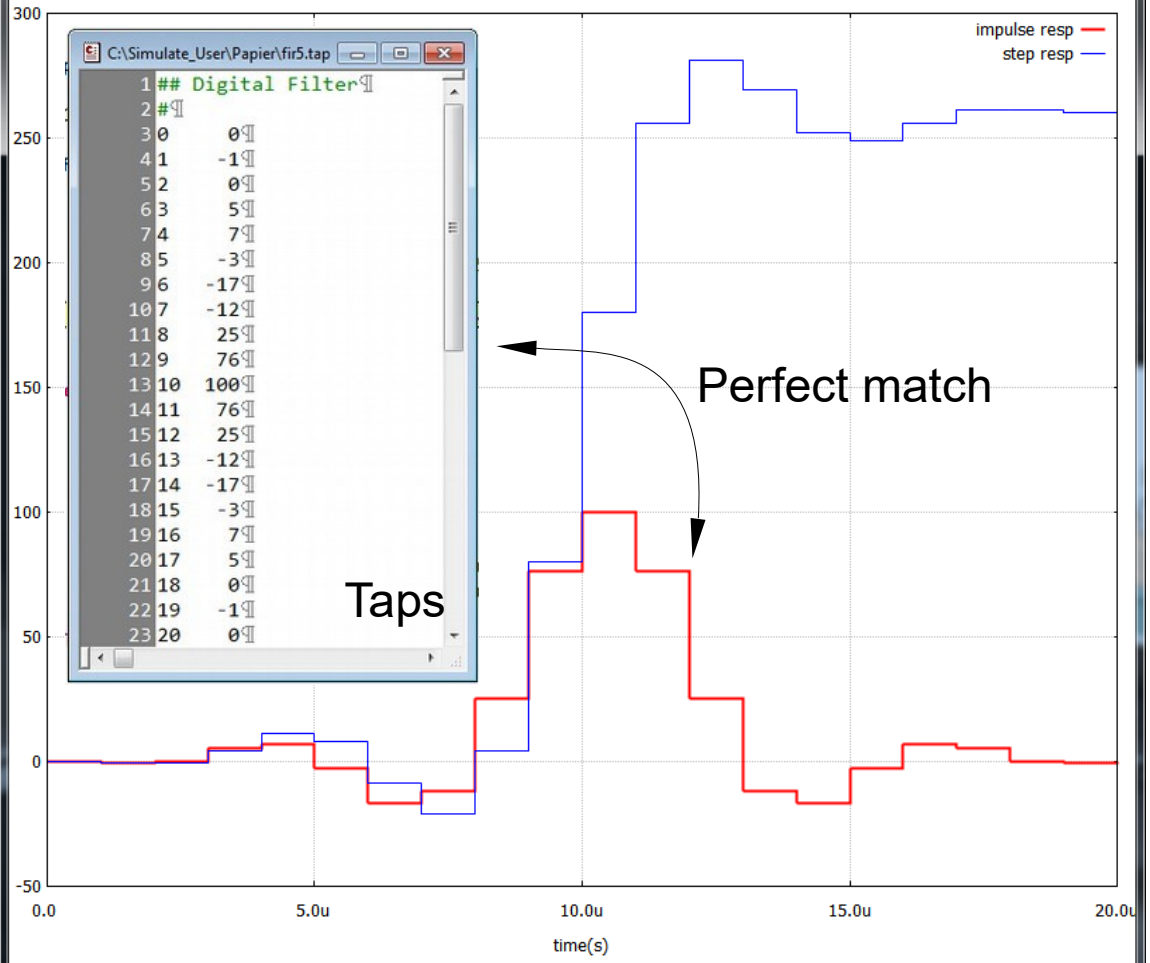
Step and Impulse Response of a Symmetrical FIR
Fri Oct 11 09:28:42 2013 by Yves Leduc

C:\Simulate_User\Papier\fir5.tap

```
1 ## Digital Filter¶
2 #¶
3 0     0¶
4 1    -1¶
5 2     0¶
6 3     5¶
7 4     7¶
8 5    -3¶
9 6   -17¶
10 7  -12¶
11 8   25¶
12 9   76¶
13 10 100¶
14 11  76¶
15 12  25¶
16 13 -12¶
17 14 -17¶
18 15  -3¶
19 16   7¶
20 17   5¶
21 18   0¶
22 19  -1¶
23 20   0¶
```

Taps

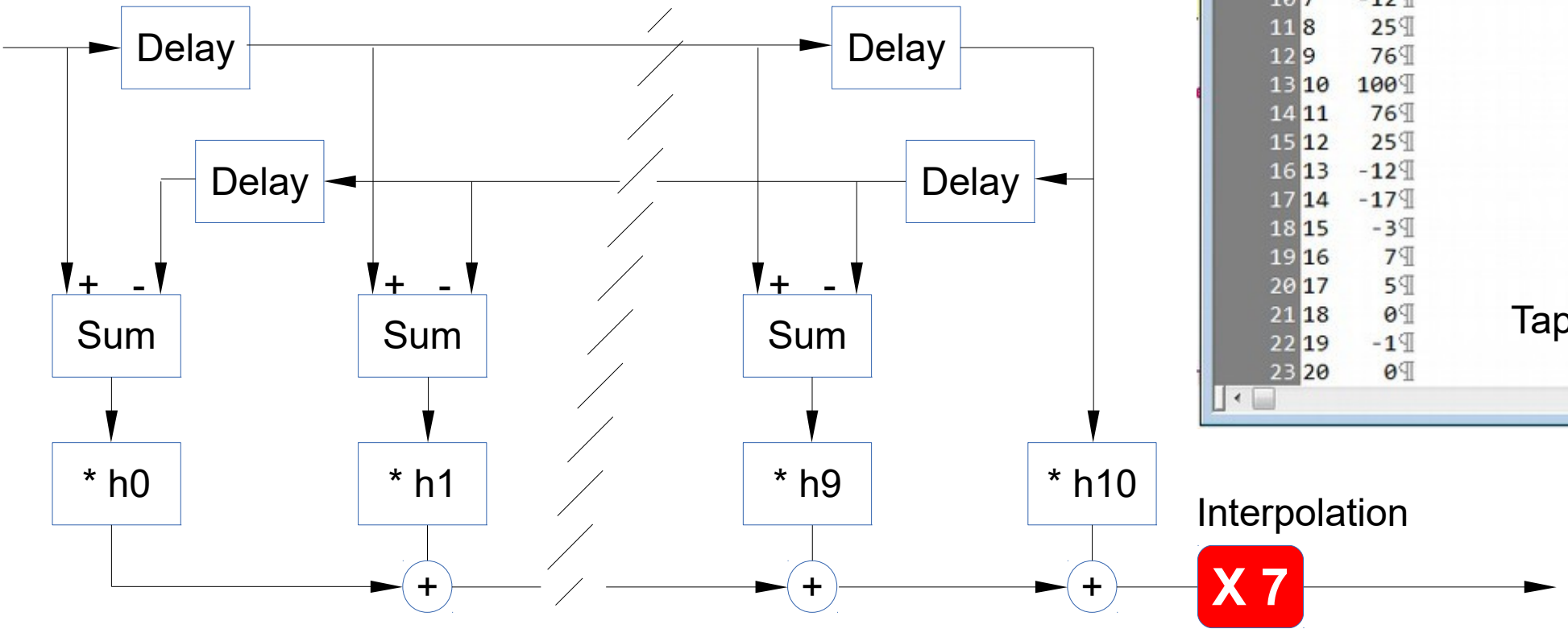
Perfect match



2.27848e-006, 241.927

Transfer Function with Interpolation ?

*Just curious, **what if** we do not introduce the zeroes for the interpolation ?*



C:\Simulate_User\Papier\fir5.tap

```
## Digital Filter¶
#¶
0      0¶
1     -1¶
2      0¶
3      5¶
4      7¶
5     -3¶
6    -17¶
7    -12¶
8     25¶
9     76¶
10   100¶
11    76¶
12    25¶
13   -12¶
14   -17¶
15    -3¶
16     7¶
17     5¶
18     0¶
19    -1¶
20     0¶
```

Taps

Digital Folded FIR

Crimson Editor - [C:\Simulate_Examples\NAPA_Running_Teaser\tf2b.nap]

File Edit Search View Document Project Tools Macros Window Help

tf2b.nap

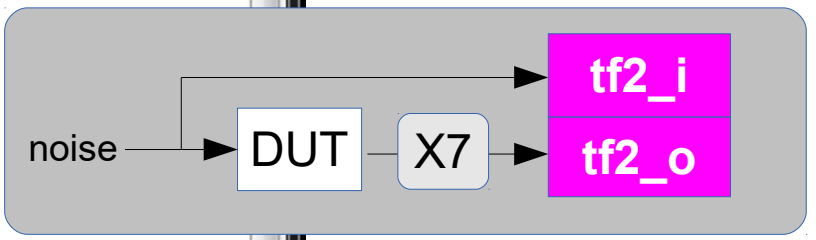
```
1
2  title  "Multirate transfer function of a symmetrical FIR"
3
4  header <napatool.hdr>
5
6  fs          1.0e6
7
8  directive REPEAT     10
9
10 ivar npts   POWEROF2(16)
11
12 node in      cell    rclk <Noise/rclock.net> 0.50
13
14 node out     generator   sf11   <fir>   "~/fir5.tap"   in
15
16
17 interpolate 7
18
19 tool tf2_i   "transfer_function_2_b.out"    in   1   npts    with in
20 tool tf2_o   "transfer_function_2_b.out"   out   1   npts
21
22 terminate TOOL_INDEX >= 1
23
24 ping
25 debug        SAMPLING   DM
```

10 RMS of FFT's

*Instructions following 'interpolate 7' are computed at **7X** frequency defined by 'fs'*

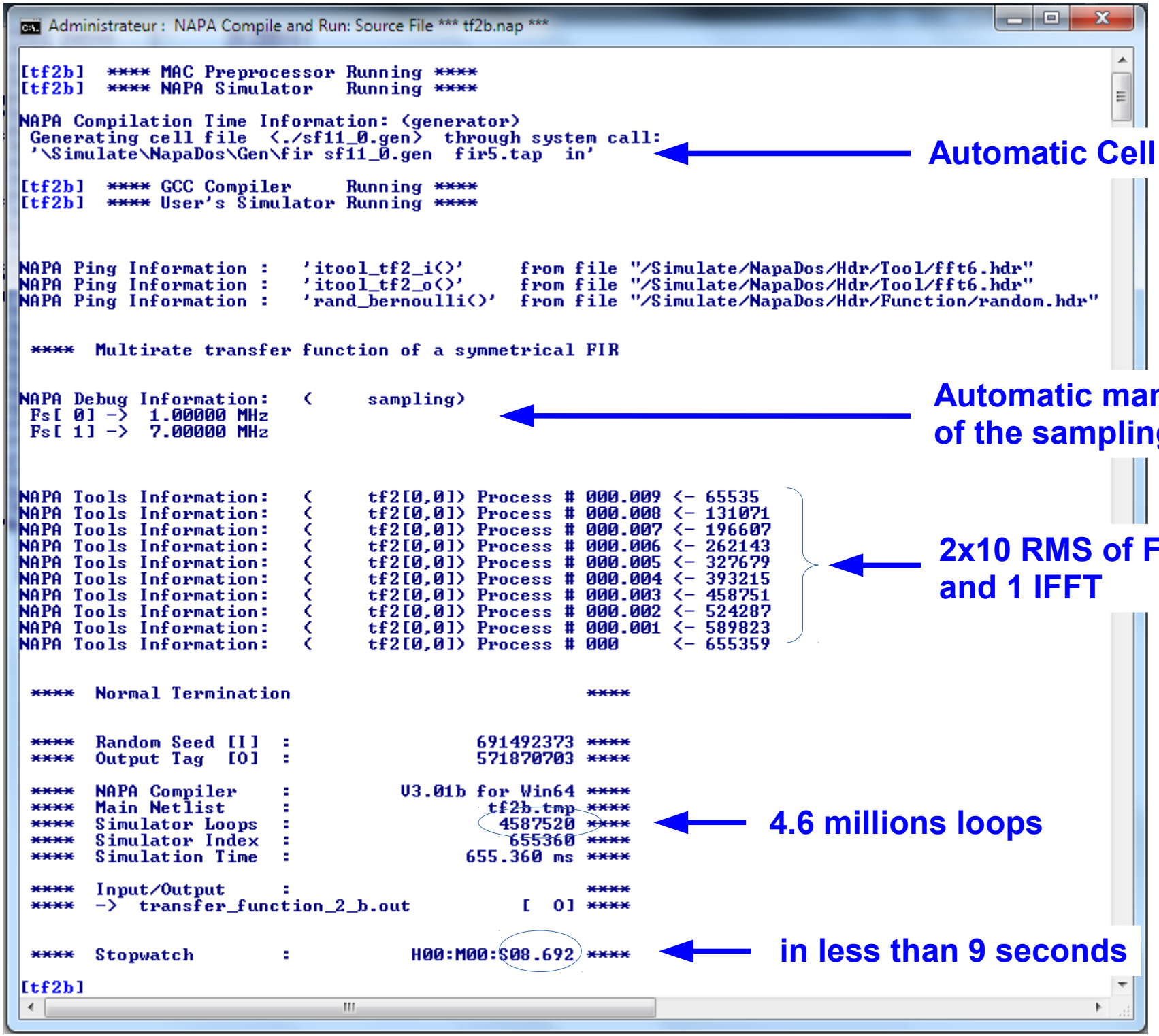
Tips : '... with in' relocates the definition of this tool just after the definition of node in', therefore is computed af frequency 'fs'

Ready Ln 27, Ch 1 27 ASCII, DOS READ REC

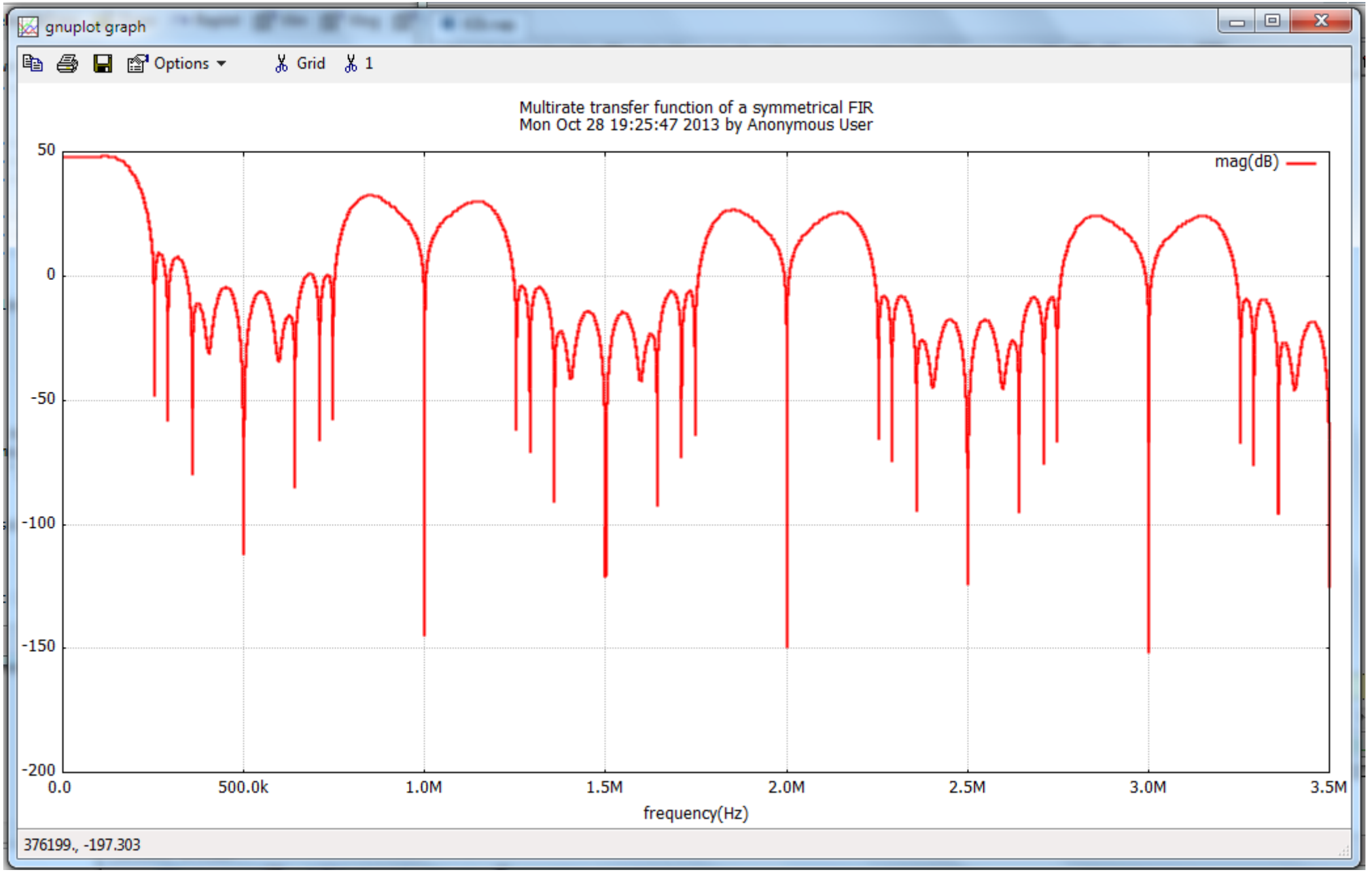


Multirate simulation
Multirate transfer function

@ 1 MHz
@ 7 MHz


Administrateur : NAPA Compile and Run: Source File *** tf2b.nap ***
[tf2b] **** MAC Preprocessor Running ****
[tf2b] **** NAPA Simulator Running ****
NAPA Compilation Time Information: (generator)
Generating cell file <./sf11_0.gen> through system call:
'\Simulate\NapaDos\Gen\fir sf11_0.gen fir5.tap in'
[tf2b] **** GCC Compiler Running ****
[tf2b] **** User's Simulator Running ****
NAPA Ping Information : 'itool_tf2_i()' from file "/Simulate/NapaDos/Hdr/Tool/fft6.hdr"
NAPA Ping Information : 'itool_tf2_o()' from file "/Simulate/NapaDos/Hdr/Tool/fft6.hdr"
NAPA Ping Information : 'rand_bernoulli()' from file "/Simulate/NapaDos/Hdr/Function/random.hdr"
**** Multirate transfer function of a symmetrical FIR
NAPA Debug Information: (sampling)
Fs[0] -> 1.00000 MHz
Fs[1] -> 7.00000 MHz
NAPA Tools Information: (tf2[0,0]) Process # 000.009 <- 65535
NAPA Tools Information: (tf2[0,0]) Process # 000.008 <- 131071
NAPA Tools Information: (tf2[0,0]) Process # 000.007 <- 196607
NAPA Tools Information: (tf2[0,0]) Process # 000.006 <- 262143
NAPA Tools Information: (tf2[0,0]) Process # 000.005 <- 327679
NAPA Tools Information: (tf2[0,0]) Process # 000.004 <- 393215
NAPA Tools Information: (tf2[0,0]) Process # 000.003 <- 458751
NAPA Tools Information: (tf2[0,0]) Process # 000.002 <- 524287
NAPA Tools Information: (tf2[0,0]) Process # 000.001 <- 589823
NAPA Tools Information: (tf2[0,0]) Process # 000 <- 655359
**** Normal Termination ****
**** Random Seed [I] : 691492373 ****
**** Output Tag [O] : 571870703 ****
**** NAPA Compiler : V3.01b for Win64 ****
**** Main Netlist : tf2b.tmp ****
**** Simulator Loops : 4587520 ****
**** Simulator Index : 655360 ****
**** Simulation Time : 655.360 ms ****
**** Input/Output : ****
**** -> transfer_function_2_b.out [0] ****
**** Stopwatch : H00:M00:S08.692 ****
[tf2b]
Automatic Cell Generation
Automatic management of the sampling
2x10 RMS of FFT's and 1 IFFT
4.6 millions loops
in less than 9 seconds

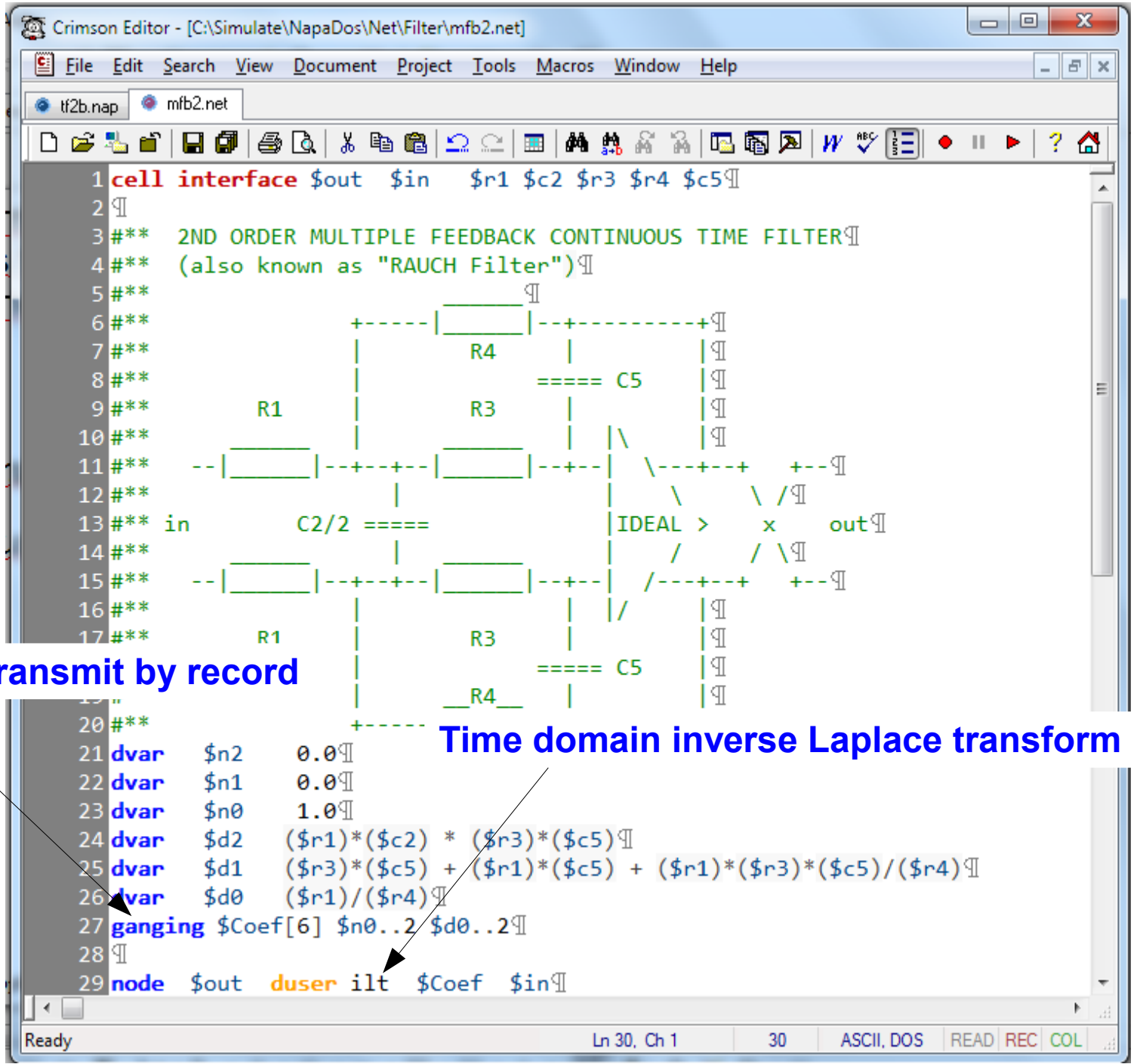
Multirate transfer function



There are much more capabilities than it is described here.

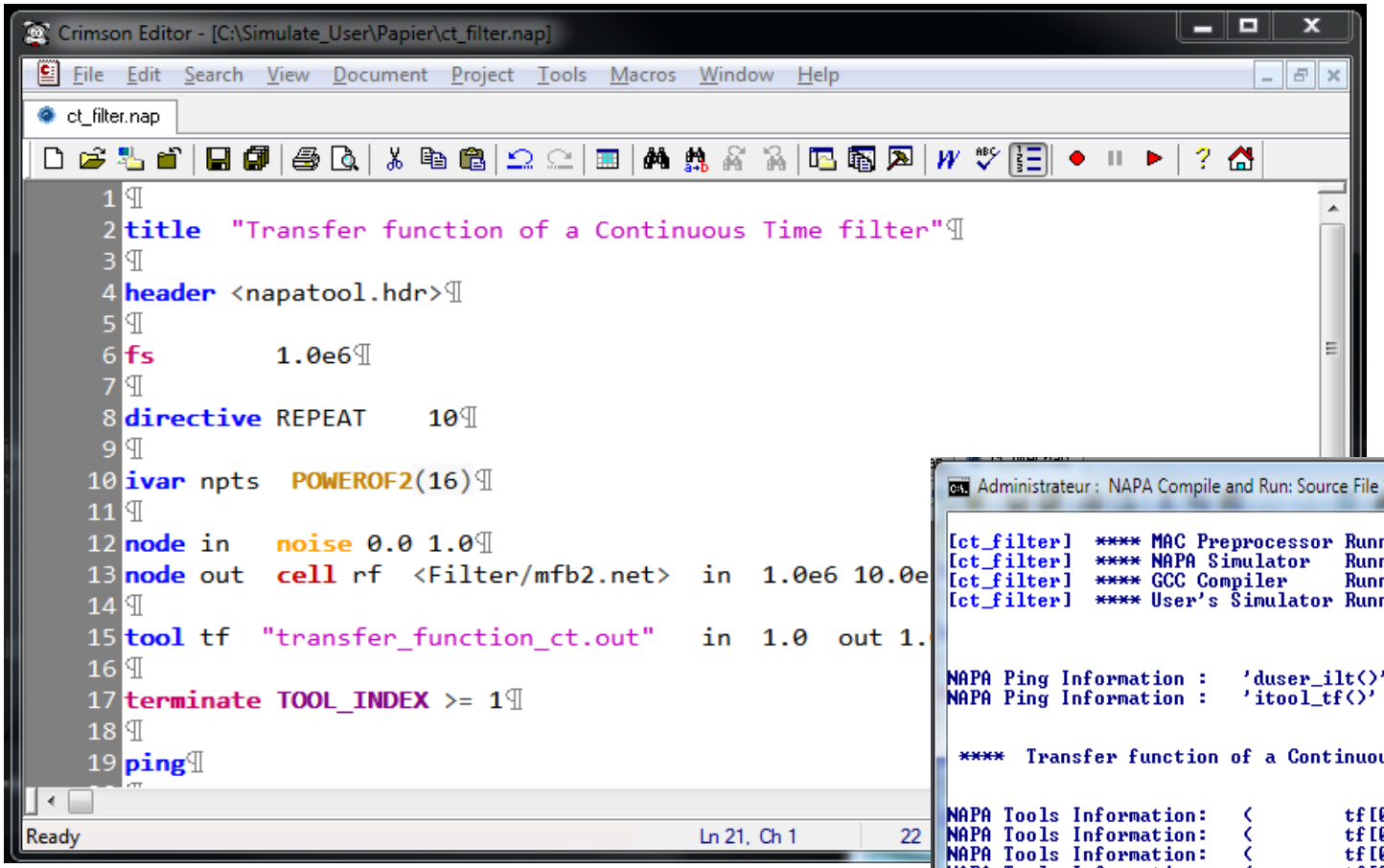
Just an example :

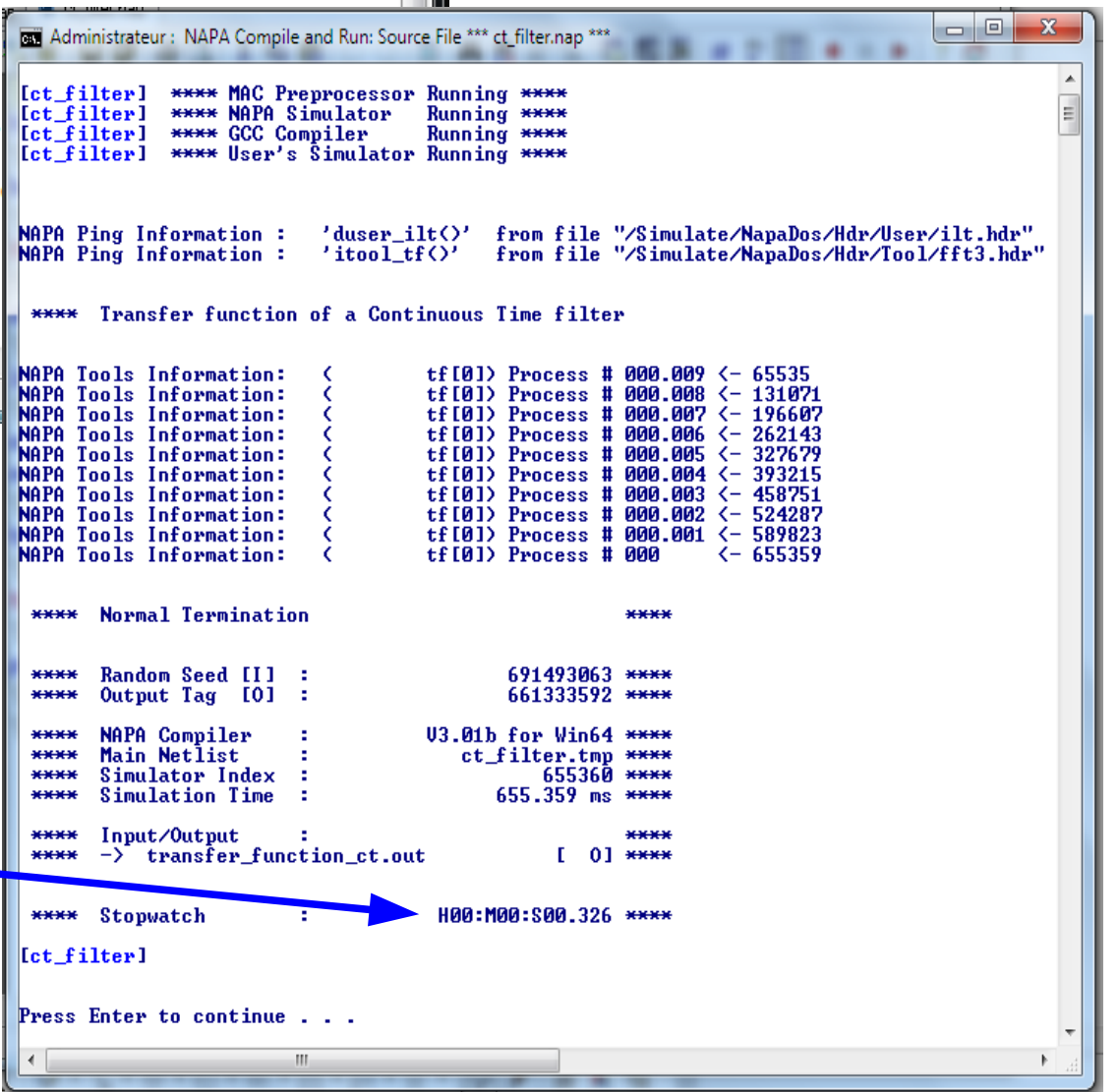
a continuous time filter



Ganging parameters to transmit by record

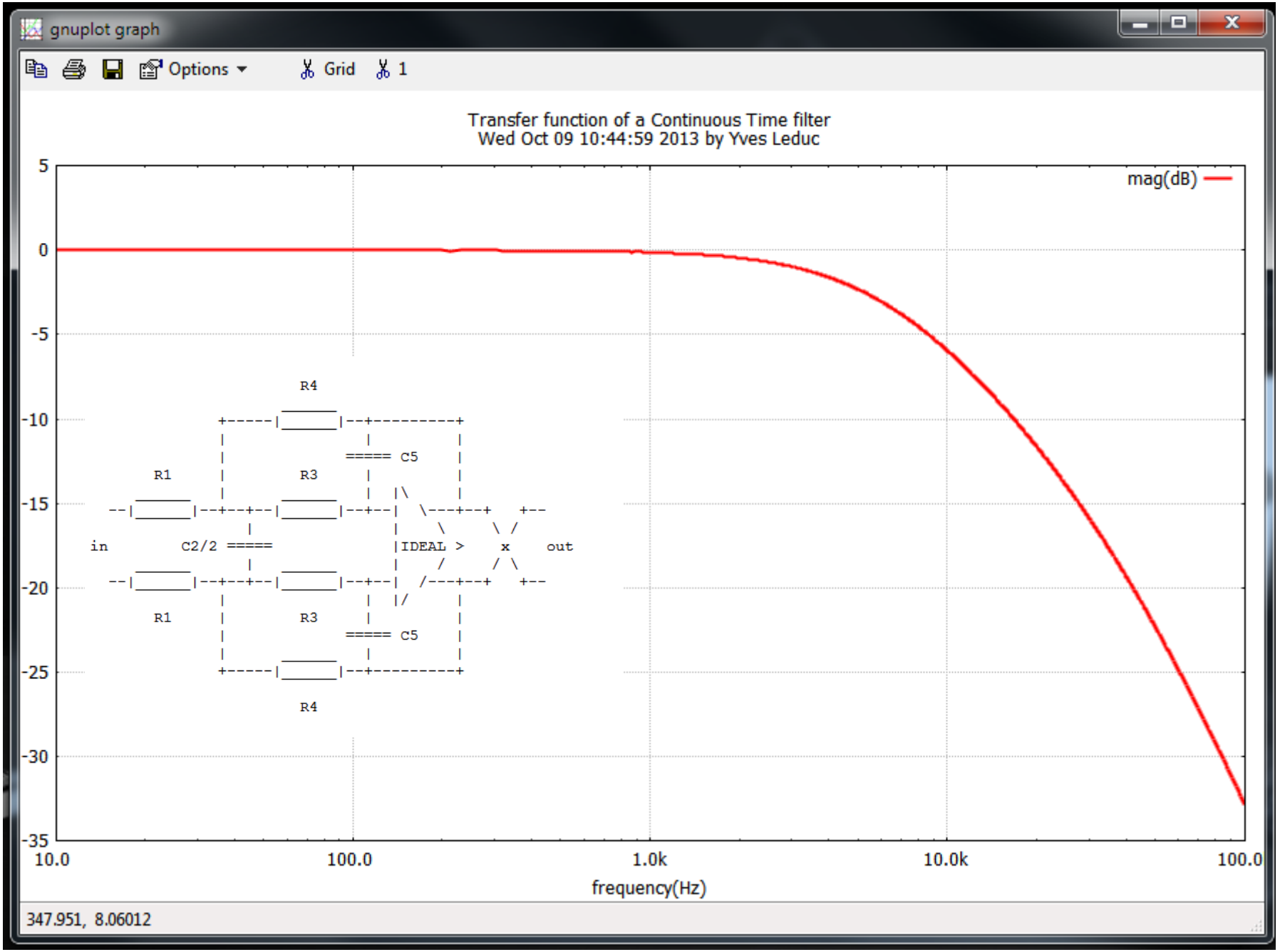
Time domain inverse Laplace transform


Crimson Editor - [C:\Simulate_User\Papier\ct_filter.nap]
File Edit Search View Document Project Tools Macros Window Help
ct_filter.nap
1 ¶
2 title "Transfer function of a Continuous Time filter"¶
3 ¶
4 header <napatool.hdr>¶
5 ¶
6 fs 1.0e6¶
7 ¶
8 directive REPEAT 10¶
9 ¶
10 ivar npts POWEROF2(16)¶
11 ¶
12 node in noise 0.0 1.0¶
13 node out cell rf <Filter/mfb2.net> in 1.0e6 10.0e
14 ¶
15 tool tf "transfer_function_ct.out" in 1.0 out 1.
16 ¶
17 terminate TOOL_INDEX >= 1¶
18 ¶
19 ping¶
Ready
Ln 21, Ch 1
22


Administrateur : NAPA Compile and Run: Source File *** ct_filter.nap ***
[ct_filter] **** MAC Preprocessor Running ****
[ct_filter] **** NAPA Simulator Running ****
[ct_filter] **** GCC Compiler Running ****
[ct_filter] **** User's Simulator Running ****
NAPA Ping Information : 'duser_ilt()' from file "/Simulate/NapaDos/Hdr/User/ilt.hdr"
NAPA Ping Information : 'itool_tf()' from file "/Simulate/NapaDos/Hdr/Tool/fft3.hdr"
**** Transfer function of a Continuous Time filter
NAPA Tools Information: (tf[0]) Process # 000.009 <- 65535
NAPA Tools Information: (tf[0]) Process # 000.008 <- 131071
NAPA Tools Information: (tf[0]) Process # 000.007 <- 196607
NAPA Tools Information: (tf[0]) Process # 000.006 <- 262143
NAPA Tools Information: (tf[0]) Process # 000.005 <- 327679
NAPA Tools Information: (tf[0]) Process # 000.004 <- 393215
NAPA Tools Information: (tf[0]) Process # 000.003 <- 458751
NAPA Tools Information: (tf[0]) Process # 000.002 <- 524287
NAPA Tools Information: (tf[0]) Process # 000.001 <- 589823
NAPA Tools Information: (tf[0]) Process # 000 <- 655359
**** Normal Termination ****
**** Random Seed [I] : 691493063 ****
**** Output Tag [O] : 661333592 ****
**** NAPA Compiler : V3.01b for Win64 ****
**** Main Netlist : ct_filter.tmp ****
**** Simulator Index : 655360 ****
**** Simulation Time : 655.359 ms ****
**** Input/Output : ****
**** -> transfer_function_ct.out [0] ****
**** Stopwatch : H00:M00:S00.326 ****
[ct_filter]
Press Enter to continue . . .

0.3 second

Its transfer function



A few Words about NAPA Error Handling

NAPA has an Extensive Set of Errors Verifications.

Undetermined loops are detected.

C:\Simulate_User\Papier\geb.nap

```
1  title   "Undetermined loops"¶
2  header <napatool.hdr>¶
3  ¶
4  fs    1.0¶
5  ¶
6  node  a  const  1¶
7  node  b  sum    a    c¶
8  node  c  gain   2    b¶
9  node  d  sum    b    c¶
10 ¶
11 output stdout   a b c d¶
12 ¶
13 terminate  LOOP_INDEX >= 1000¶
```

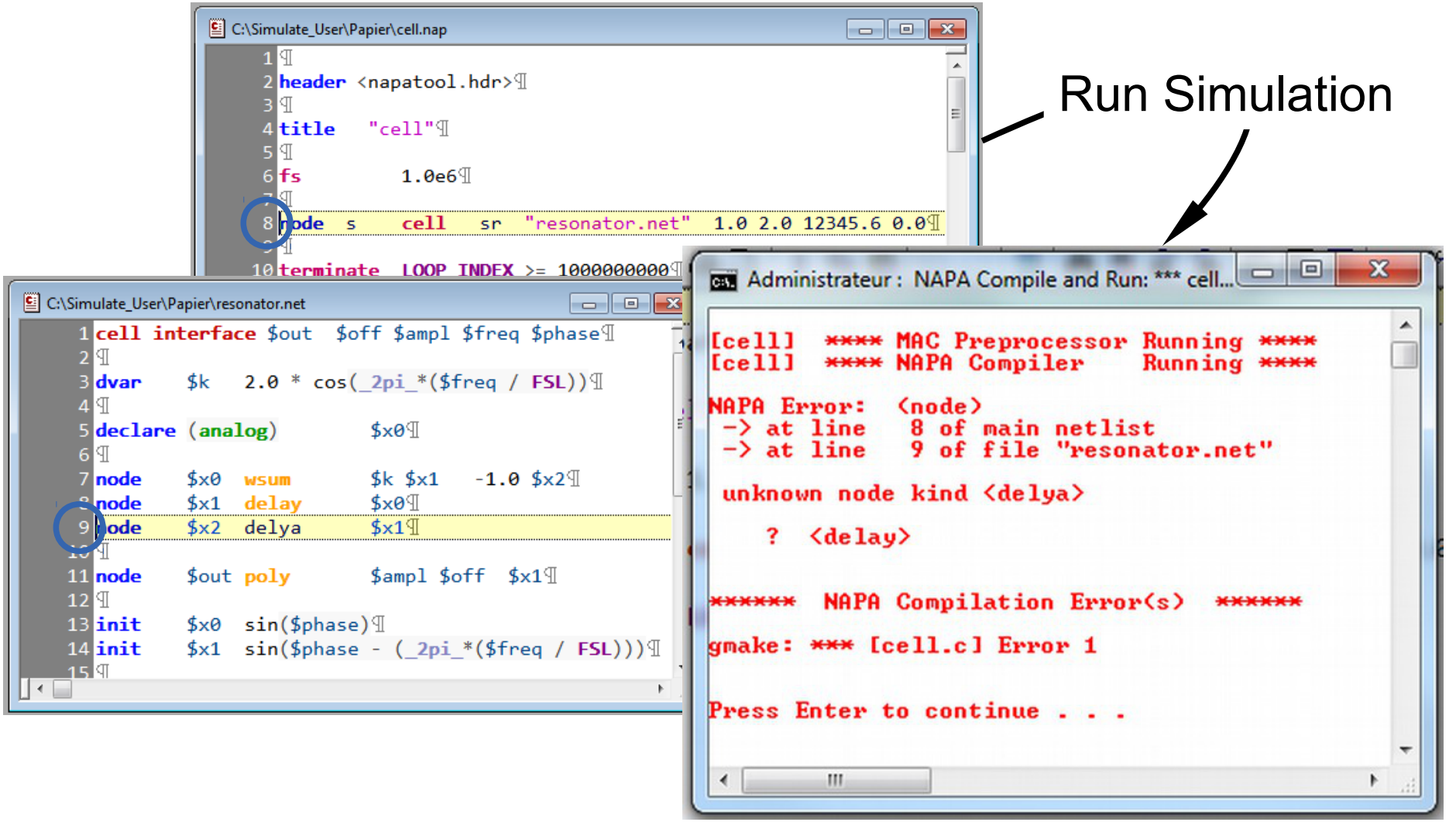
Run Simulation

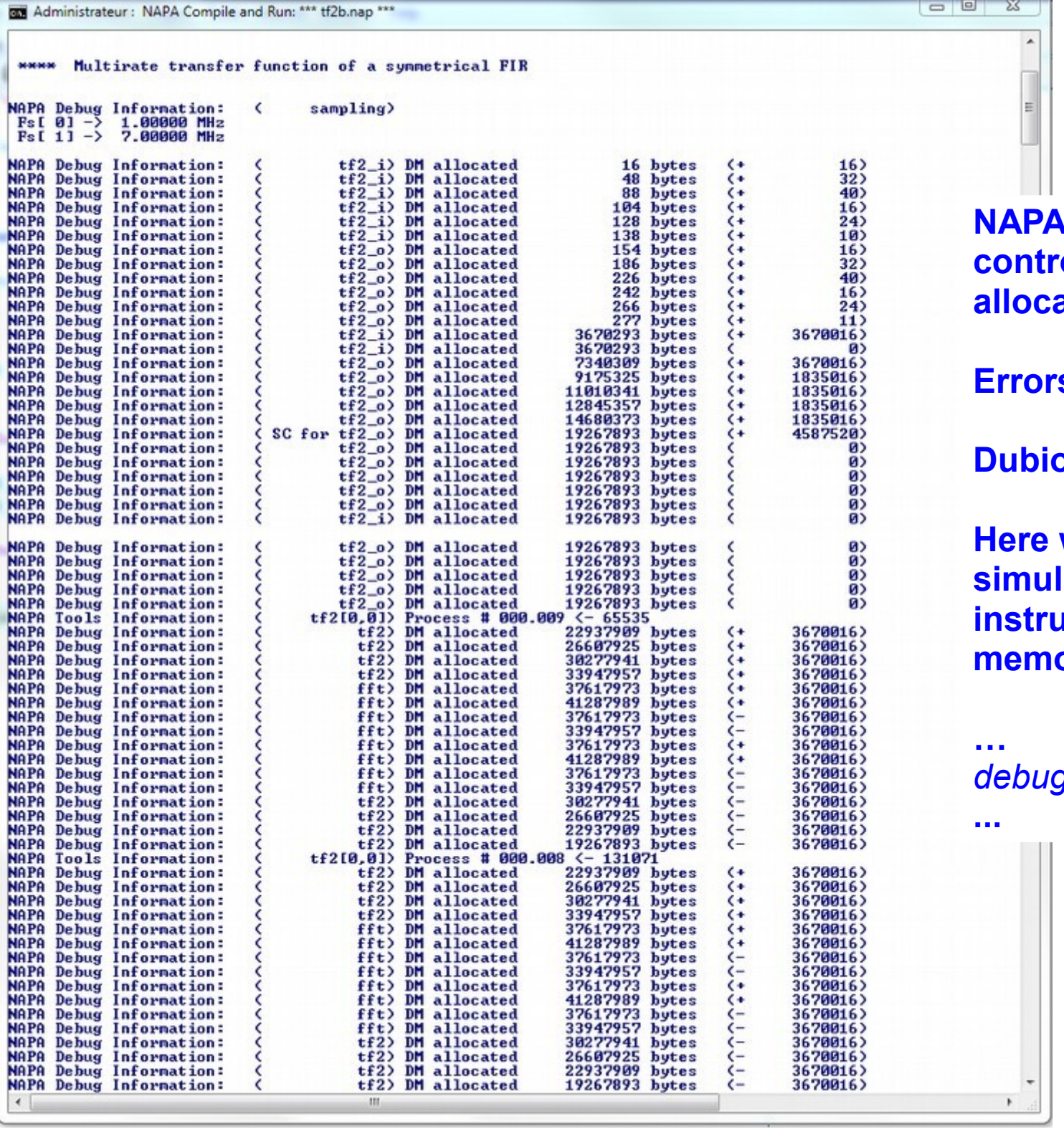
Administrateur : NAPA Compile and Run: *** geb.nap ***

```
[geb]  **** MAC Preprocessor Running ****
[geb]  **** NAPA Compiler    Running ****

NAPA Error:  (static loop)

 These nodes are involved in one or several static loops:

   -> node  c
   -> node  b

 Your description is not correct. NAPA is not able to sort
 the nodes of your netlist. Check for missing or mispelled
 nodes, or loops of nodes containing no delay or only delays


******  NAPA Compilation Error(s)  ******

gmake: *** [geb.c] Error 1

Press Enter to continue . . .
```

Type Mismatches, Syntax Errors are Catched at Compilation

A syntax error inside a cell is detected and documented here :





NAPA has a VERY STRICT control on memory allocation and I/O usage.

Errors are catched on the fly.

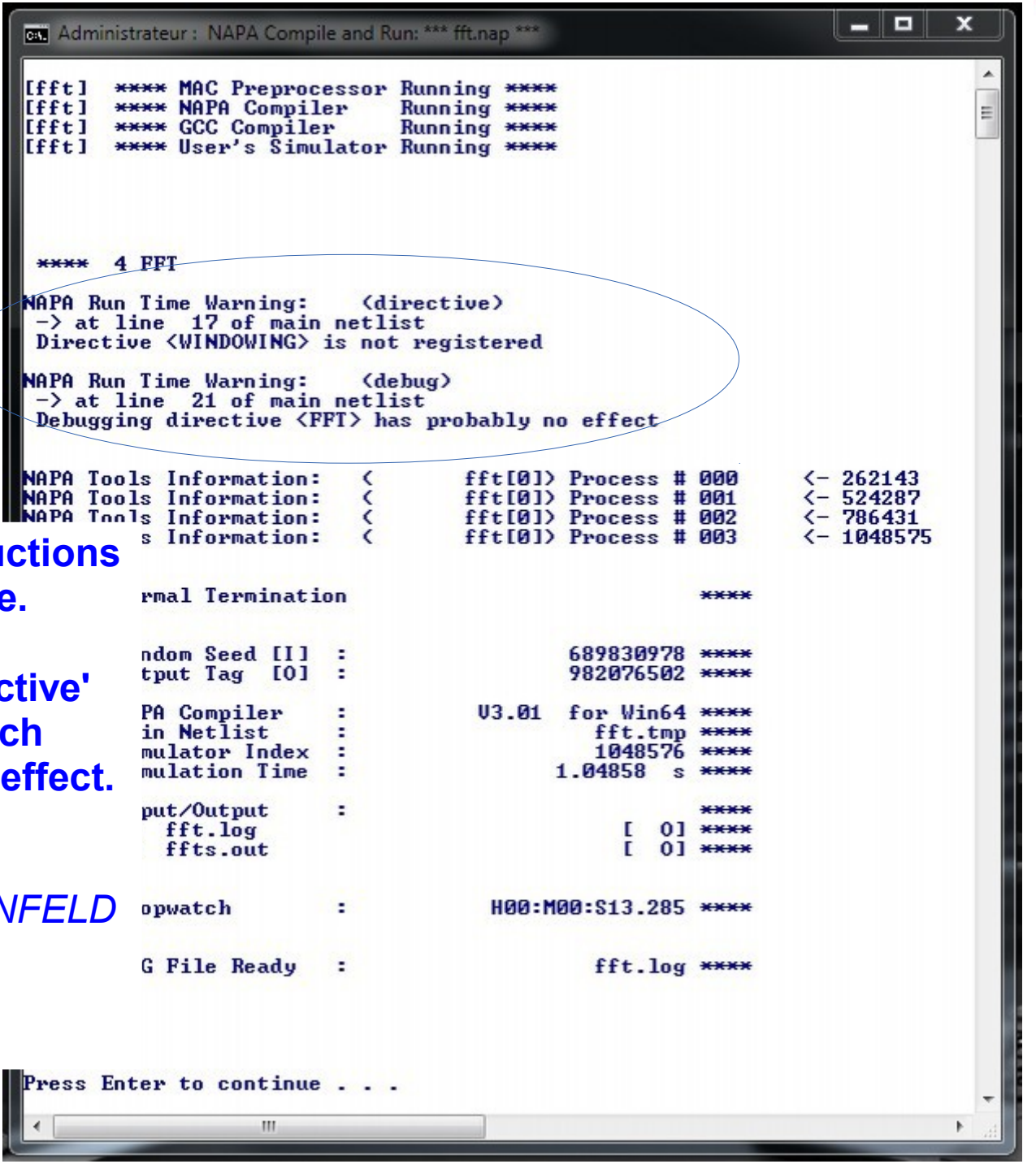
Dubious behavior is catched.

Here we are running the simulator with an additional instruction to show the dynamic memory management at work

…

debug DM

…



NAPA controls also the instructions which appear to be ineffective.

Here the user asks for a 'directive' and a 'debug' instruction which apparently have probably no effect.

…
directive WINDOWING ROSENFELD
…
debug FFT
…

Dedicated Solutions are Implemented for the Modeling and Simulation of :

Analog SWC circuits,

Linear circuits described in Laplace domain,

Linear circuits described with elementary components with occasional percussion,

Limited width digital register arithmetic,

PLL,

etc... etc...

Last but not Least !

A -very- important NAPA user-defined function 'sarc' uses

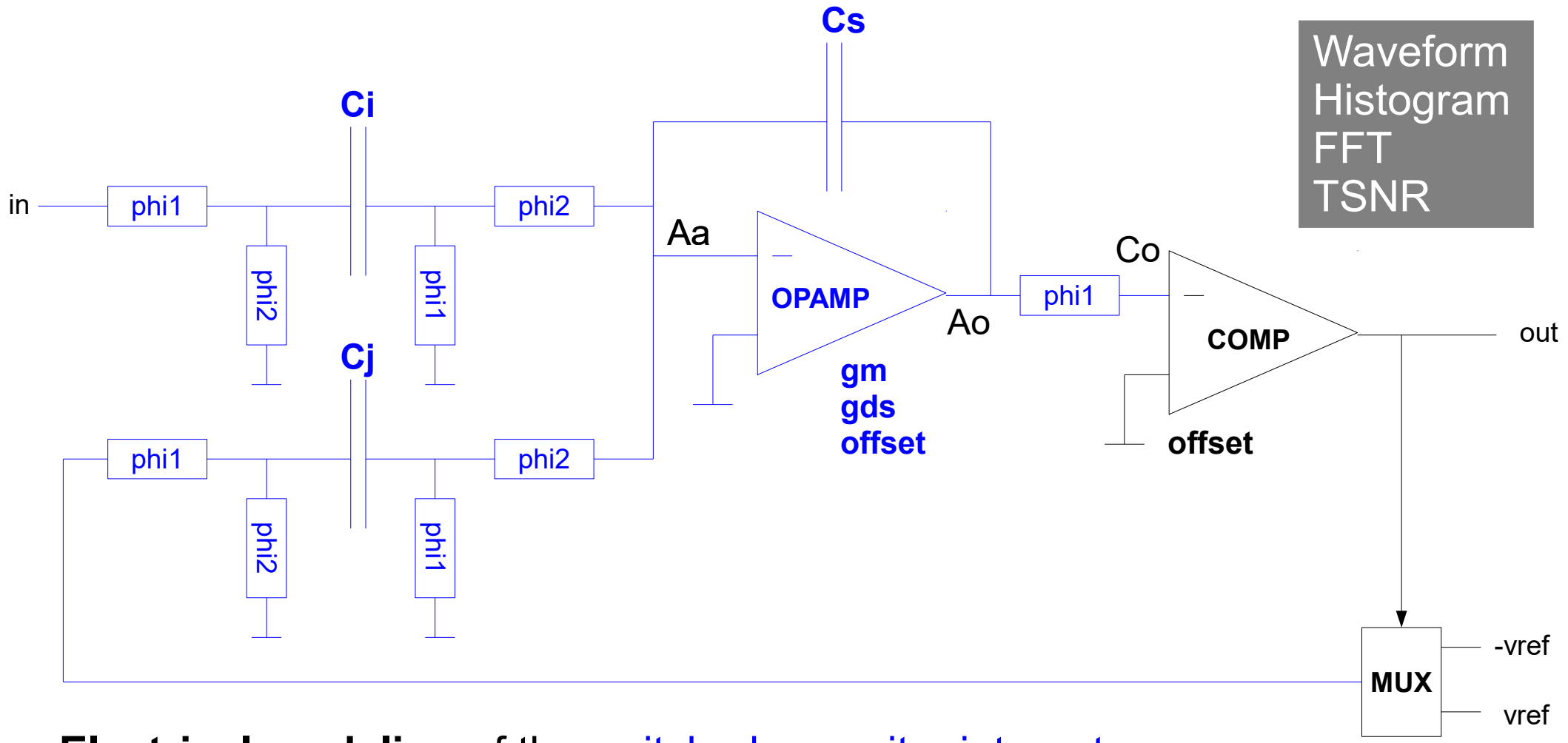
'Semi Analytical Recursive Convolution'

to model accurately and simulate linear circuits with on-demand percussion.

In addition to continuous time domain circuits, this function allows the simulation of switched capacitors circuits at the switch level, class D amplifiers, DC-DC converters, etc..

This important solution is developped by Jacques Mequin and desserves a dedicated and detailed presentation.

Simulating with SARC : Simulation at lower level



Electrical modeling of the switched capacitor integrator

offset, limited gain, limited bandwidth, parasitic capacitances, switches

Mixed signal modeling of the **modulator** :

integrator, comparator, multiplexer

```
NETLIST( [V,    Vin,    i,     Gnd              ],      /* the inputs  */
         [V,    Vf,     f,     Gnd              ],
         [V,    Avoff,  Ab,    Gnd              ],

         [R,    Ar1ia,  i,     Ai1              ],
         [R,    Ar2ib,  Ai1,   Gnd              ],
         [C,    Aci,    Ai1,   Ai2              ],
         [R,    Ar1ic,  Ai2,   Gnd              ],
         [R,    Ar2id,  Ai2,   Aa               ],
         [R,    Ar1ja,  f,     Aj1              ],
         [R,    Ar2jb,  Aj1,   Gnd              ],
         [C,    Acj,    Aj1,   Aj2              ],
         [R,    Ar1jc,  Aj2,   Gnd              ],
         [R,    Ar2jd,  Aj2,   Aa               ],
         [C,    Aca,    Aa,    Gnd              ],
         [G,    Agm,    Ao,    Gnd,  Aa,  Ab    ],
         [R,    Ards,   Ao,    Gnd              ],
         [C,    Aco,    Ao,    Gnd              ],
         [C,    Acs,    Aa,    Ao               ],
         [R,    Cr1,    Ao,    Co               ],
         [C,    Cload,  Co,    Gnd              ] ) ;

OUTPUTS( V(Aa), V(Ao),   V(Co) ) ;                  /* the outputs */
AKA( PHI1,  Ar1ia, Ar1ic, Ar1ja, Ar1jc,  Cr1 ) ;
AKA( PHI2,  Ar2ib, Ar2id, Ar2jb, Ar2jd       ) ;

GENERATE_MIMO( ) ;
```

SD1_core netlist

(MAXIMA)

```
cell interface   $Voutd   $Vin $Vref  $Clk   $Afile $Cfile

title " [Analog 1st Order SD Modulator with SARC model] "

node  $PHI1  dalgebra  $Clk?  $ROff : $ROn       // variable resistance
node  $PHI2  dalgebra  $Clk?  $ROn  : $ROff      // variable resistance
dvar  $ROn   1.0e3
dvar  $ROff  1.0e9

dvar  $a     1.0                                 // scaling input
dvar  $g     1.0                                 // scaling reference
dvar  $Aci   $a*$Acs
dvar  $Acj   $g*$Acs
dvar  $Acs   5.0e-12
dvar  $ibias 125.0e-6                            // opamp bias current
data  $Afile       $Agm $Ards $VoffA  $Aca $Aco  $ibias

ganging  $parms[]  $Aca $Aci $Acj $Aco $Acs $Ck $PHI1 $PHI2 $Ards $Agm
node  $tag   duser     sarc   SD1_core()  $parms  $VoffA  $fdbck  $Vin
node ($Aa)   duser     sarc   $tag  (V@Aa)
node ($Ao)   duser     sarc   $tag  (V@Ao)
node ($Co)   duser     sarc   $tag  (V@Co)

data  $Cfile       $VoffC  $Ck
node  $Cod   delay     $Co
node  $Voutd comp      $Cod    $VoffC
node  $fdbck mux       $Voutd  $Vref -$Vref
```

SD1 modulator cell netlist

```
data interface   $gm $rds  $off  $ca $co   $ib

declare  (true)  (100.0e-6 <= $ib) && (150.0e-6 >= $ib)

dvar  $gds   26.0e-9 - 2.4e-15*$ib     - 4.6e-24*$ib*$ib    &update
dvar  $rds   1.0/$gds                                       &update

dvar  $gm    300.0e-6 - 230.0e-9*SQRT($ib) + 1.2e-12*$ib    &update

dvar  $ca    0.2e-12                                        &constant
dvar  $co    0.4e-12                                        &constant

dvar  $off   rand_normal(0.0, 3.0e-3)
```

opamp data cell netlist

```
data interface   $off  $cin

dvar  $cin   0.5e-12                                        &constant

dvar  $off   rand_normal(0.0, 2.0e-3)
```

comparator data cell netlist

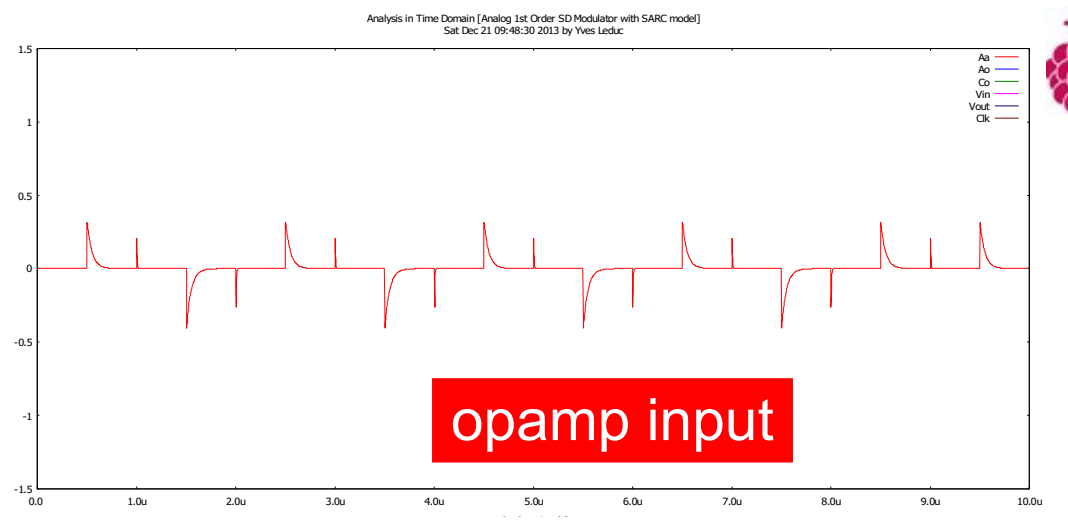
```
header <napatool.hdr>
title  "Analysis in Time Domain"

fs       2.0e6
node     Clk       clock  "01"
string   opfile1   "transconductance_opamp.dat"
string   opfile2   "comparator.dat"

interpolate fs    200                    // compute 200 samples per phase
node     Vin       dc     0.123456789
node     Vrefa     dc    (analog)   1.0
node     Vout      cell   sd1   "./sd1.net"  Vin Vrefa  Clk  opfile1..2

output   "time.out"     Aa Ao  Co  Vin Vout   Clk

nominal      fs

terminate    20 <= LOOP_INDEX

alias    Aa    sd1__Aa
alias    Ao    sd1__Ao
alias    Co    sd1__Co

debug    SAMPLING  SARC_INFO
ping
```

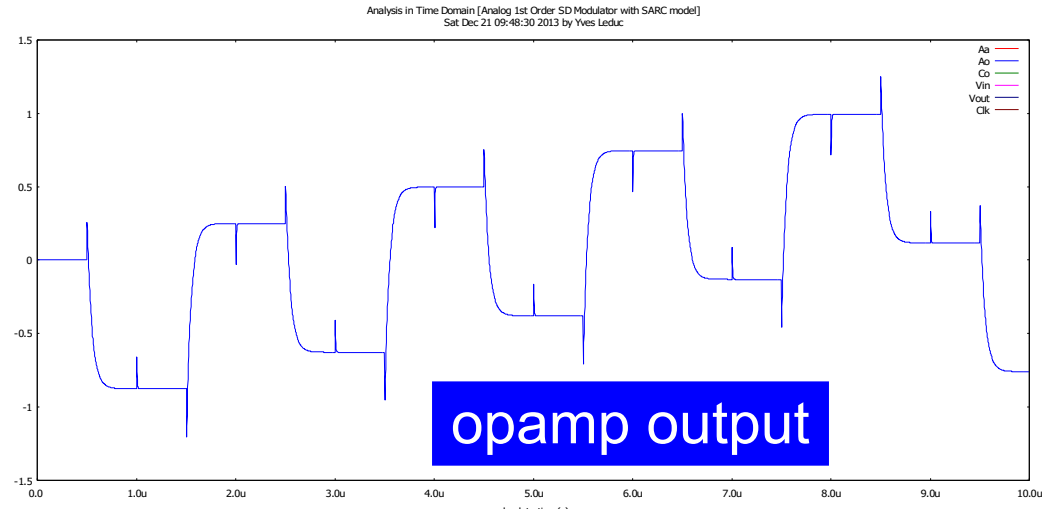
NAPA Simulation
(Waveforms)

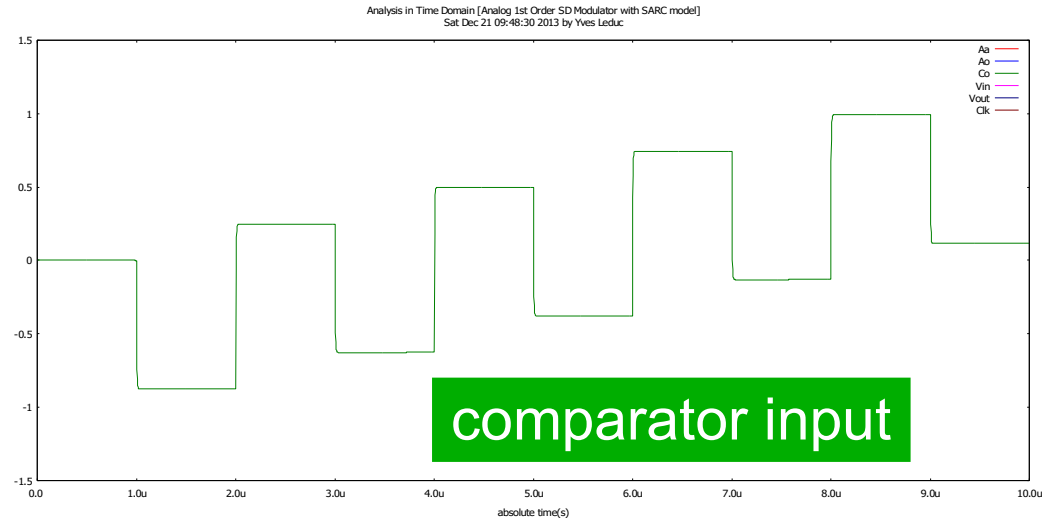
Administrateur : NAPA Compile and Run: Source File ***...

```
[time_analyziz]  **** MAC Preprocessor  Running ****
[time_analyziz]  **** MAPA Compiler     Running ****
[time_analyziz]  **** GCC Compiler      Running ****
[time_analyziz]  **** SARC Engine       Linking ****
[time_analyziz]  **** Ad Hoc Simulator  Running ****

MAPA Ping Information :   "duree_sarc()"    from file "/Simulate/MapaDoz/Hdr/User/sarc.hdr"
MAPA Ping Information :   "SD1_core()"      from file "/Simulate/MapaDoz/Hdr/Mac/SD1\SD1_core.hdr"
MAPA Ping Information :   "print_var()"     from file "/Simulate/MapaDoz/Hdr/Function/print_var.hdr"
MAPA Ping Information :   "rand_normal()"   from file "/Simulate/MapaDoz/Hdr/Function/random.hdr"

 ****  Analyziz in Time Domain [Analog 1st Order SD Modulator with SARC model]

MAPA Debug Information:   (     sampling)
 Fz[ 0] ->  2.00000 MHz
 Fz[ 1] ->  400.000 MHz
 Fz[ 2] ->  2.00000 MHz
 Fz[ 3] ->  1.00000 MHz
 Fz[ 4] ->  2.00000 MHz

 ****  320.0 u     <- rd1__VoffA
 ****  258.7 u     <- rd1__VoffC
 ****  500.0 f     <- rd1__Ck
 ****  11.54 k     <- rd1__Again
 ****  5.000 p     <- rd1__Acr
 ****  5.000 p     <- rd1__Acj
 ****  5.000 p     <- rd1__Aci

MAPA Debug Information:   (     zarc[ 0])
[ SD1_core           ]     - schematic      -
                           <<< STAGE A >>>                          <<< STAGE C >>>
                                                 Acr
                                                  ||
                                              o-----||-----o
                                 Aci          |    ||     |
                                  ||          |    ||     |
                  Ar1ia--|1|--||--|2|--Ar2id--o---o--|-  .     o--Cr1--o
                  |      ||  |                |   |     o--o       |      | Cload
                Ar2ib    Ar1ic               |   b--|            ===
                 _|_      _|_                |   |  |. OPAMPA    _|_
                                              |   |
                                 Acj          | VoffA
                                  ||          |
        f--Ar1ja--|1|--||--|2|--Ar2jd--o      |
        |         |  ||  |             |      |
        |       Ar2jb    Ar1jc         |      |
        |        _|_      _|_          |      |
        f-------------------------------------------------------------- Vf


   OPAMP :   o ---------o      o----------o------o----          COMP :   o ----------o--------|------|
                        |      |          |      |                                   |        |      |
                  co ===  gm*(a-b)   rds    === co                          ck ===    |  a>b   |>---- o
                o-------b     |_|      |_|   |_|                        o-------b-------|------|
                                                                                       |      |
                                                                                       |______|

[ SD1_core           ]     - initialization -

[ SD1_core           ]     sampling at  400.000 MHz

[ SD1_core           ]     inputs       { Voff, Vf, Vin }
[ SD1_core           ]     outputs      { V(Aa), V(Ao), V(Co) }
[ SD1_core           ]     parameters   { Aca, Aci, Acj, Aco, Acr, Cload, PHI1, PHI2, Ardz, Agm }
[ SD1_core           ]     parameter #1        rd1__Aca =   200.000 f   ->   Aca
[ SD1_core           ]     parameter #2        rd1__Aci =   5.00000 p   ->   Aci
[ SD1_core           ]     parameter #3        rd1__Acj =   5.00000 p   ->   Acj
[ SD1_core           ]     parameter #4        rd1__Aco =   400.000 f   ->   Aco
[ SD1_core           ]     parameter #5        rd1__Acr =   5.00000 p   ->   Acr
[ SD1_core           ]     parameter #6         rd1__Ck =   500.000 f   ->   Cload
[ SD1_core           ]     parameter #7       rd1__PHI1 =   0.00000     ->   PHI1
[ SD1_core           ]     parameter #8       rd1__PHI2 =   0.00000     ->   PHI2
[ SD1_core           ]     parameter #9       rd1__Ardz =   38.4615 M   ->   Ardz
[ SD1_core           ]     parameter #10       rd1__Agm =   299.997 u   ->   Agm

 ****  Normal Termination                          ****

 ****  Random Seed [I]   :              693807855 ****
 ****  Output Tag  [O]   :              451714812 ****

 ****  MAPA Compiler     :          V3.02 for Win64 ****
 ****  Main Netlist      :        time_analyziz.tmp ****
 ****  Simulator Loops   :                     4000 ****
 ****  Simulator Index   :                       20 ****
 ****  Simulation Time   :          9.99750 us     ****

 ****  Input/Output      :                          ****
 ****  ->  time.out      :                [  0]    ****

 ****  Stopwatch         :        H00:M00:S00.347 ****

[time_analyziz]

Press Enter to continue . . .
```

4 thousands loops
0.35 second

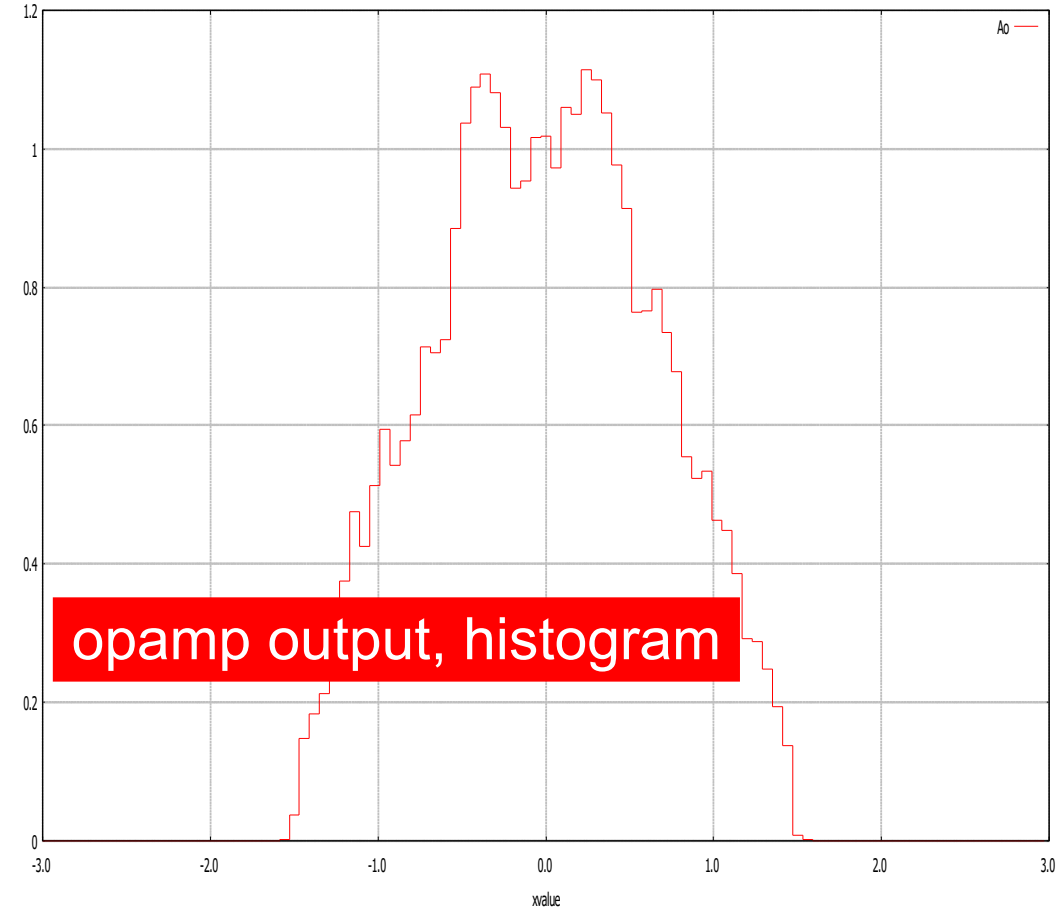


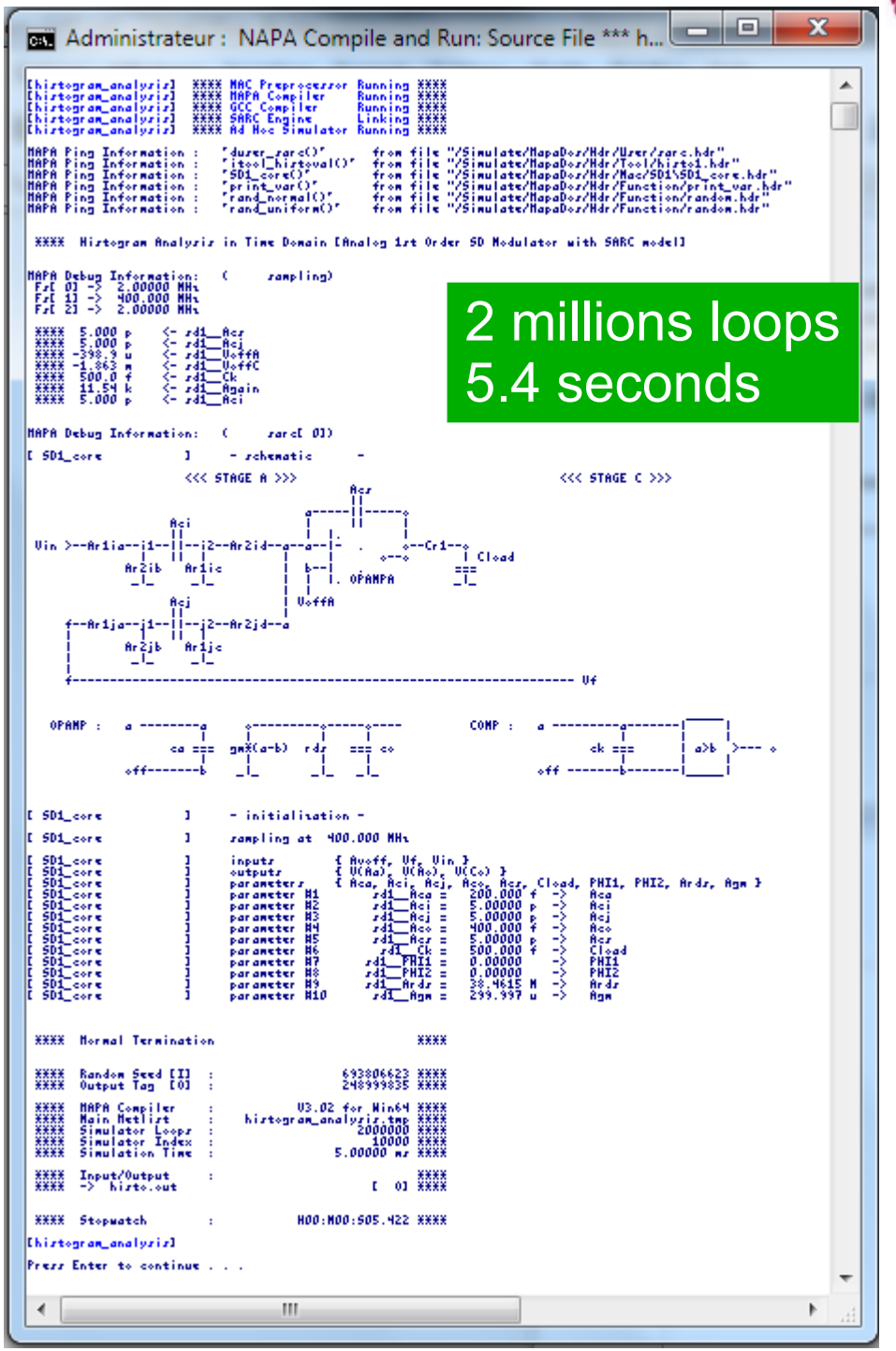




```
header <napatool.hdr>
title  "Histogram Analysis in Time Domain"

fs       2.0e6
node     Clk       clock  "01"
string   opfile1   "transconductance_opamp.dat"
string   opfile2   "comparator.dat"

dvar     ampldb    -6.0
dvar     ampl      DB2LIN(ampldb, 1.0)
dvar     freq      1234.56789
dvar     ph        rand_uniform(0.0, _2pi_)

interpolate fs     200

node     Vin       osc    0.0   ampl    freq    ph
node     Vrefa     dc    (analog) 1.0
node     Vout      cell   sd1   "./sd1.net"  Vin Vrefa  Clk  opfile1..2
tool     histoval  "histo.out"    Ao Vrefa    -3.0  3.0  100

nominal      fs

terminate    10000 <= LOOP_INDEX

alias    Ao    sd1__Ao
debug    SAMPLING   SARC_INFO
ping
```

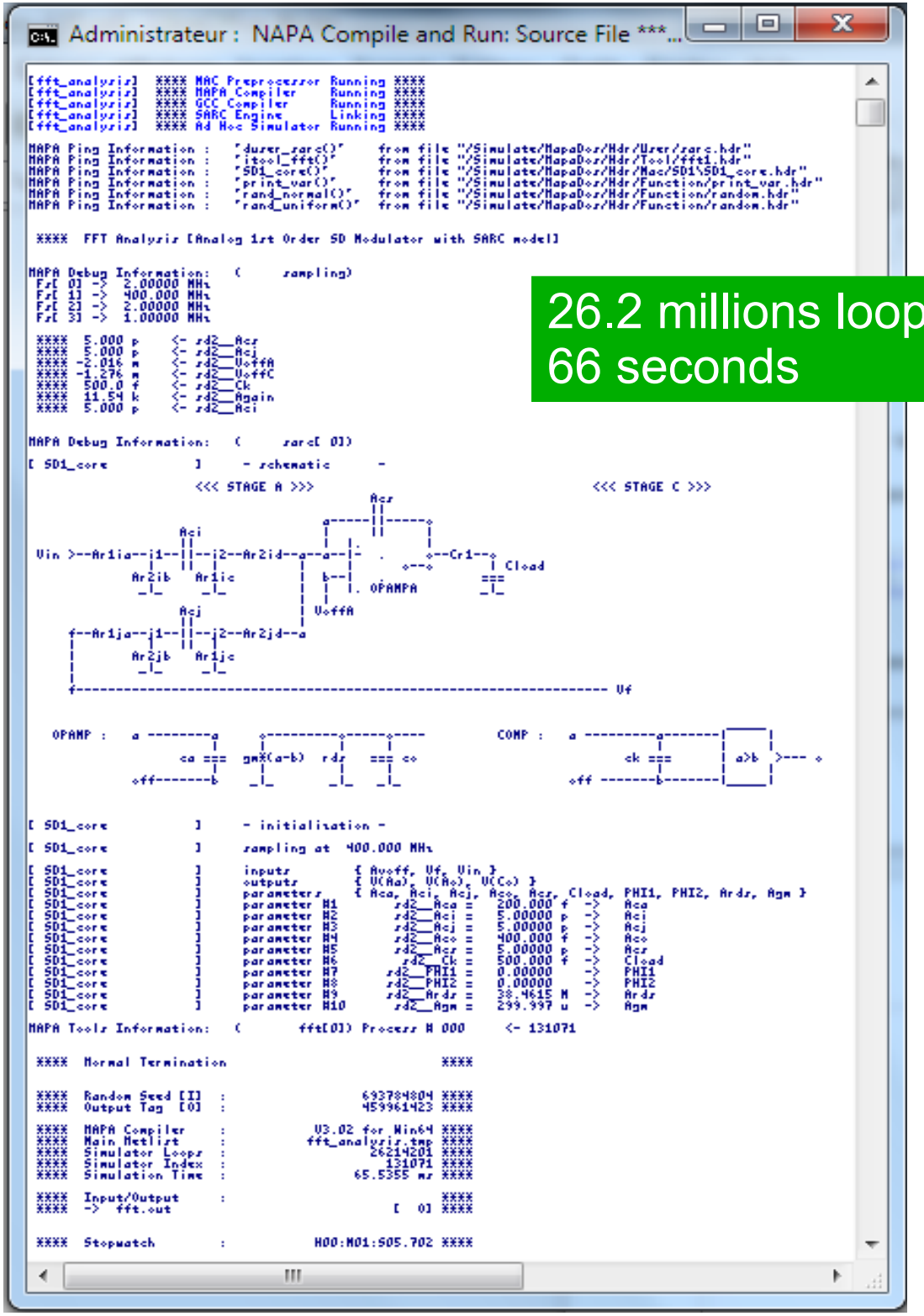
NAPA Simulation
(Histogram)


Histogram Analysis in Time Domain [Analog 1st Order SD Modulator with SARC model]
Sat Dec 21 09:07:26 2013 by Yves Leduc
Ao
1.2
1
0.8
0.6
0.4
0.2
0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0
xvalue
opamp output, histogram

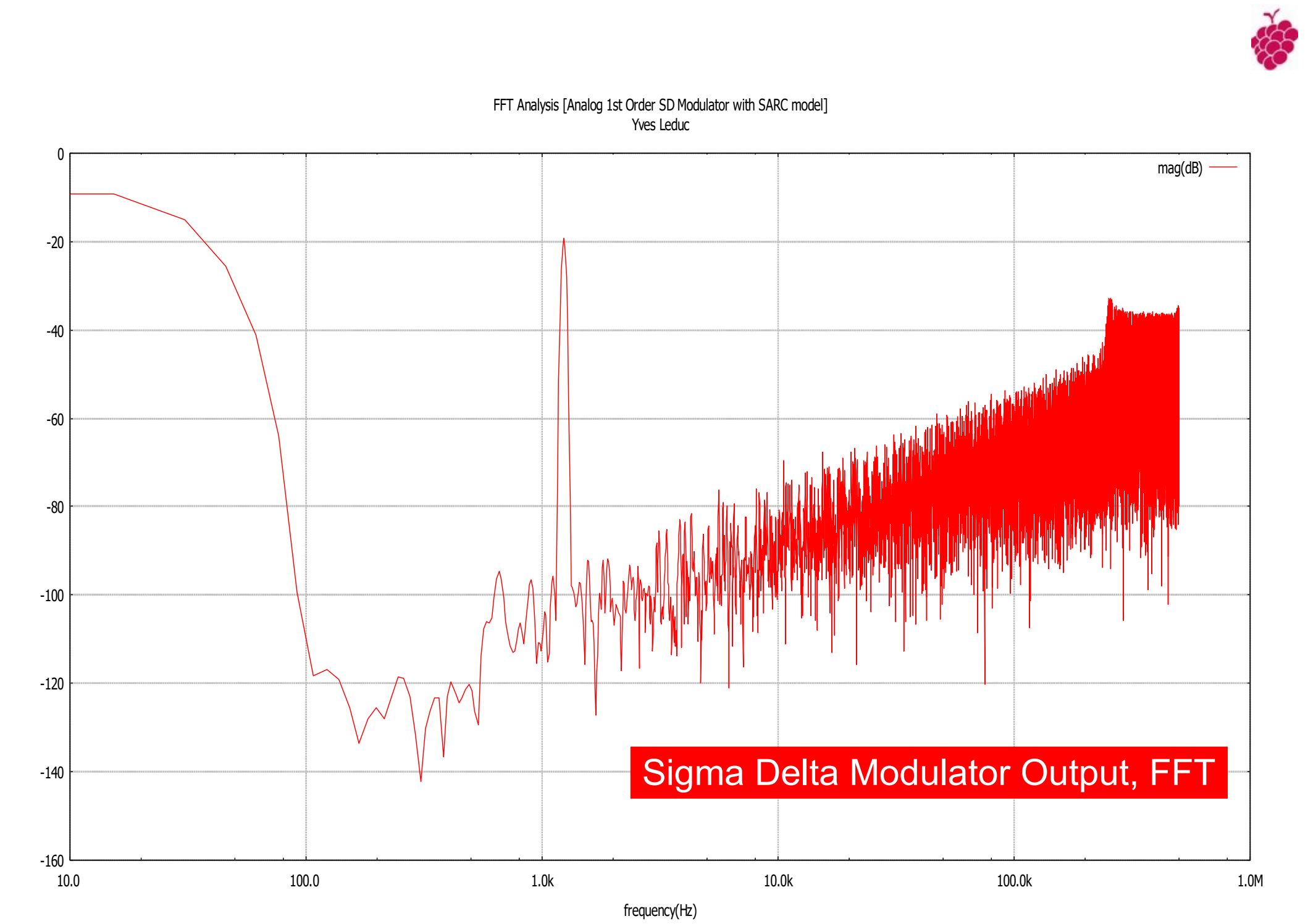

Administrateur : NAPA Compile and Run: Source File *** h...
2 millions loops
5.4 seconds

```
header <napatool.hdr>
title  "FFT Analysis"

fs       2.0e6
node     Clk       clock  "01"
string   opfile1   "transconductance_opamp.dat"
string   opfile2   "comparator.dat"

dvar     ampldb    -6.0
dvar     ampl      DB2LIN(ampldb, Vrefa)
dvar     freq      1234.56789
dvar     ph        rand_uniform(0.0, _2pi_)

interpolate fs     200
node     Vin       osc  0.0  ampl  freq   ph
node     Vrefa     dc   (analog)   1.0
node     Voutd     cell sd1  "./sd1.net"  Vin Vrefa  Clk  opfile1..2
node     Vrefd     dc   (digital)  1

decimate     fs    2  1
ivar     npts      POWEROF2(16)
tool     fft       "fft.out"    Voutd Vrefd  npts

terminate    1 <= TOOL_INDEX

debug    SAMPLING  SARC_INFO
ping
```

NAPA Simulation (FFT)

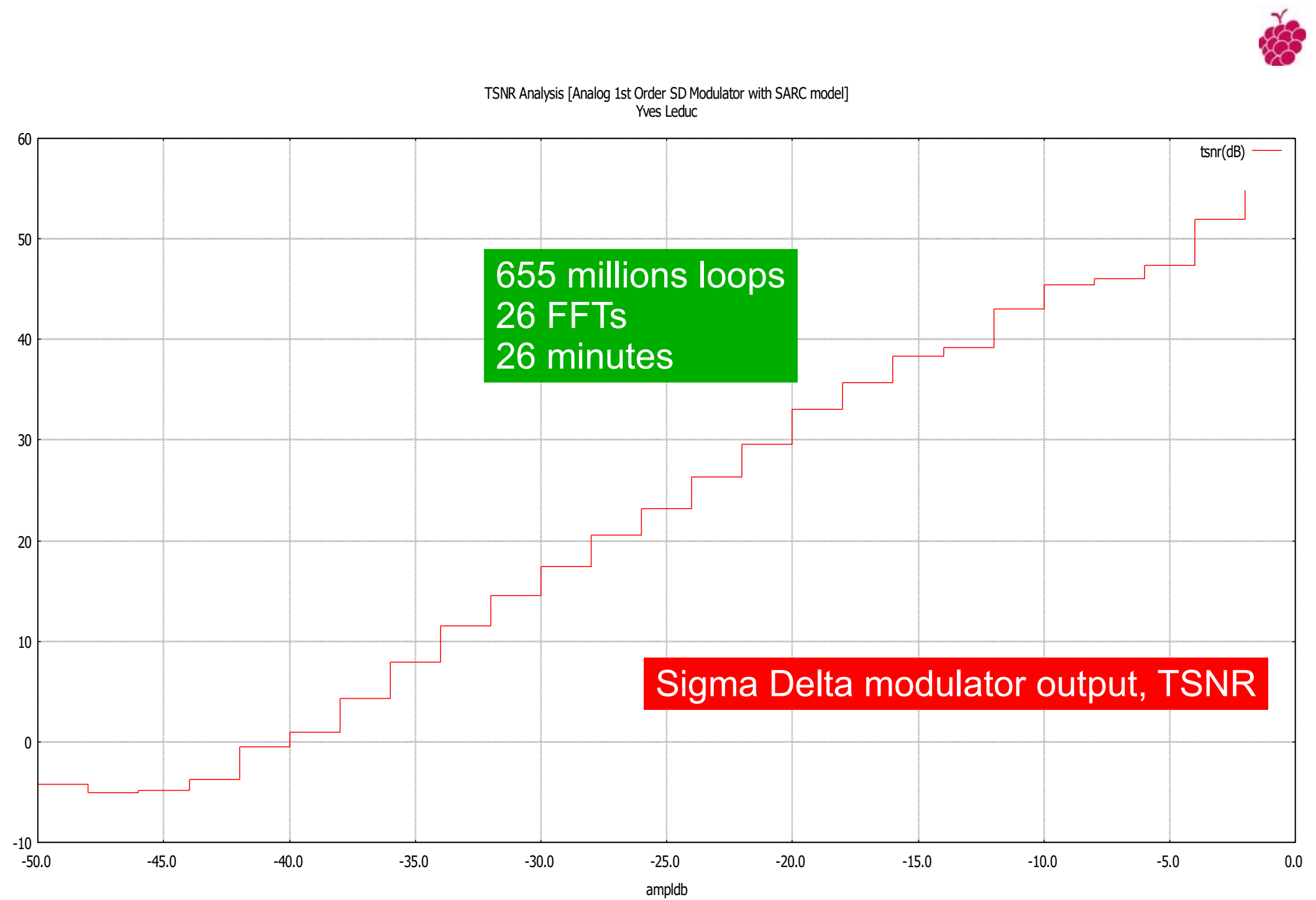


26.2 millions loops
66 seconds


FFT Analysis [Analog 1st Order SD Modulator with SARC model]
Yves Leduc
mag(dB)
0
-20
-40
-60
-80
-100
-120
-140
-160
10.0
100.0
1.0k
10.0k
100.0k
1.0M
frequency(Hz)
Sigma Delta Modulator Output, FFT

```
header <napatool.hdr>
title  "TSNR Analysis"

fs       2.0e6
node     Clk        clock  "01"
string   opfile1    "transconductance_opamp.dat"
string   opfile2    "comparator.dat"

dvar     ampldb     LINSWEEP(TOOL_INDEX, -50.0, 0.0, 26)   &update  &export
dvar     ampl       DB2LIN(ampldb, 1.0)                    &update
dvar     freq       1234.56789                             &constant
dvar     ph         rand_uniform(0.0, _2pi_)               &constant

interpolate fs     200
node     Vin        osc    0.0   ampl   freq    ph
node     Vrefa      dc     (analog)    1.0
node     Voutd      cell   sd1   "./sd1.net"    Vin Vrefa    Clk   opfile1..2
node     Vrefd      dc     (digital)   1

decimate     fs     2     1
ivar     npts       POWEROF2(16)
tool     tsnr       "tsnr.out"     Voutd Vrefd   8.0e3   npts

terminate    0.0 <= ampldb

debug    SAMPLING   SARC_INFO
ping
```

NAPA Simulation
TSNR


TSNR Analysis [Analog 1st Order SD Modulator with SARC model]
Yves Leduc
tsnr(dB)
655 millions loops
26 FFTs
26 minutes
Sigma Delta modulator output, TSNR
60
50
40
30
20
10
0
-10
-50.0
-45.0
-40.0
-35.0
-30.0
-25.0
-20.0
-15.0
-10.0
-5.0
0.0
ampldb

NAPA Has MANY Other Nice Features.

*They are listed in the **NAPA User's Manual** and in the NAPA Quick Card.*

*The **NAPA Primer Booklet** describes in details the advanced features of NAPA.*

They deserve a visit !

alias
array
assert
call
command_line
comment
data
debug
decimate
declare
directive
drop
dump
dvar
error
event
export
format
fs
ganging
gateway
header
init
inject
input
interface
cell interface
data interface
interpolate
ivar
load
napa_version
node
nominal
num_initial
opcode
output
ping
post
random_seed
restart
stuck
string
synchronize
terminate
title
tool
ts
update
void
warning
#*

NAPA Instructions

adc
algebra
alu
and
average
bshift
btoi
buffer
bwand
bwbuffer
bwinv
bwnand
bwnor
bwnot
bwor
bwxnor
bwxor
cell
change
clip
clock
comp
copy
cosine
dac
dalgebra
dc
delay
differentiator
div
dtoi
dtool
duser
equal
fzand
fzbuffer
fzinv
fzor
fznand
fznor
fznot
gain
generator
hold
ialgebra
integrator
inv
itob
itod
itool
iuser
latch
lshift
max
merge
min
mod
muller
mux
nand
noise
nor
not
offset
or
osc
poly
prod
quant
ram
ram2
rect
register
relay
rip
rom
rom2
rshift
rshift1
rshift2
sign
sine
square
step
sub
sum
toggle
track
triangle
trig
uadc
udac
wsum
xnor
xor
zero

NAPA Nodes

ABS(x)
SIGN(x)
MIN(x,y)
MAX(x,y)
LIMIT(x,l,h)
ISINSIDE(x,l,h)
ISOUTSIDE(x,l,h)
ISEQUAL(x,y)
ISNOTEQUAL(x,y)
ISTIME(t)
ISSMALL(x)
ISNOTSMALL(x)
ISEVEN(x)
ISODD(x)
ISINTEGER(x)
ISPOWEROF2(n)
POWEROF2(n)
MODULO(x,y)
SIN(x)
COS(x)
SQRT(x)
LOG(x)
POW(x,y)
ROOT(x,y)
LOG10(x)
POW10(x)
D2I(x)
I2D(n)
DB2LIN(x,r)
LIN2DB(x,r)
DB2POW(x,r)
POW2DB(x,r)
RAD2DEG(x)
DEG2RAD(x)
LENGTH(s)
LINDOMAIN(c,b,e)
LOGDOMAIN(c,b,e)
LINSWEEP(c,b,e,n)
LOGSWEEP(c,b,e,n)
RAND_01()
RAND_01_X()
FSS(nseg)
STS(nseg)
NIS(nseg)
IO_MANAGER(c,f,n,s,t)
OPTION(f,i,o)
PING(fun)

Built-in C Macros

Activation\chirp.net
Activation\chirp2.net
Activation\chirp2_g.net
Activation\chirp_g.net
Activation\pulse.net
Activation\resonator.net
Activation\sigmoid.net
Activation\step.net
Activation\three_phase.net
Activation\triangle.net
Adder\i1.net
Adder\i1_a.net
Adder\i2.net
Adder\i2_a.net
Adder\i3.net
Adder\i3_a.net
Adder\i4.net
Adder\i4_a.net
ASD\m1.net
ASD\m11.net
ASD\m2.net
ASD\m211.net
ASD\m21_va.net
ASD\m22_va.net
ASD\m22_vc.net
ASD\m2ff.net
Bessel\LP1.net
Bessel\LP2.net
Bessel\LP3.net
Bessel\LP4.net
Bessel\LP5.net
Bessel\LP6.net
Bessel\LP7.net
Bessel\LP8.net
Biquad\Martin_Sedra.net
Biquad\SWC.net
Biquad\SWC1.net
Biquad\SWC2.net
Biquad\z.net
Butterfly\4.net
Butterfly\4b.net
Butterfly\4b_hl.net
Butterworth\LP1.net
Butterworth\LP10.net
Butterworth\LP2.net
Butterworth\LP3.net
Butterworth\LP4.net
Butterworth\LP5.net
Butterworth\LP6.net
Butterworth\LP7.net
Butterworth\LP8.net
Butterworth\LP9.net
CDSD\m1.net
CDSD\m11.net
CDSD\m2.net
Cell\1.net
Cell\2.net
Cell\3.net
Cell\4.net
Cell\5.net
Cell\6.net
Cell\d1.net
Cell\d2.net
Cell\d3.net
Cell\d4.net
Cell\d5.net
Cell\d6.net
Comparator\1_h.net
Comparator\2.net
Comparator\2_a.net
Comparator\2_h.net
Comparator\3.net
Comparator\d1_h.net
Comparator\d2.net
Comparator\d2_a.net
Comparator\d2_h.net
Comparator\d3.net
Counter\bincount.net
Counter\counter.net
Counter\counter2.net
Counter\edgecount.net
Counter\modcnt.net
Counter\modcnt2.net
Counter\modcntr.net
Counter\modcntr2.net
DC_removal\dcr0.net
DC_removal\dcr1.net
DC_removal\dcr2.net
DC_removal\dcr4.net
Detect\d.net
Detect\u.net
Detect\ud.net
Differentiator\1.net
DSD\m1.net
DSD\m11.net
DSD\m2.net
DWA\16_d.net
DWA\4_d.net
DWA\8.net
DWA\8_d.net
Filter\1p1z.net
Filter\2p2z.net
ISD\m.net
Latch\DFF1.net
Latch\DFF2.net
Latch\SR.net
Logic\adder.net
Logic\adder2b.net
Logic\adder3b.net
Logic\add_evn.net
Logic\add_gen.net
Logic\add_odd.net
Logic\carry.net
Logic\half_adder.net
Logic\sum.net
Measure\energy.net
Measure\freq.net
Measure\slope.net
Misc\adder.net
Misc\channel.net
Misc\cint.net
Misc\cintd.net
Misc\gdelay.net
Misc\GTswitch.net
Misc\intdz.net
Misc\intm.net
Misc\intz.net
Misc\rv1.net
Modulation\am.net
Modulation\am2.net
Modulation\fm.net
Modulation\fm2.net
Modulation\phm.net
Noise\jitter.net
Noise\ktoverc.net
Noise\pink.net
Noise\rclock.net
Noise\red.net
Processor\bbh.net
Processor\bbh4.net
PWL\d.net
PWL\i.net
PWM\1.net
PWM\2.net
PWM\3.net
PWM\4.net
Range\max.net
Range\max_z.net
Range\min.net
Range\min_z.net
Range\width.net
Sequence\a.net
Filter\filt11.net
Filter\filt22.net
Filter\filt33.net
Filter\lp1.net
Filter\lprc1.net
Filter\lprc2.net
Filter\ma.net
Filter\mfb2.net
Filter\nyq365.tap
Filter\nyquist.net
Integrator1\d1.net
Integrator1\d1i1.net
Integrator1\d1i1_a.net
Integrator1\d1i1_ac.net
Integrator1\d1_a.net
Integrator1\d1_ac.net
Integrator1\d1_r.net
Integrator1\d2.net
Integrator1\d2_a.net
Integrator1\d2_ac.net
Integrator1\d2_r.net
Integrator1\d3.net
Integrator1\d3_a.net
Integrator1\d3_ac.net
Integrator1\d3_r.net
Integrator1\d4.net
Integrator1\d4_a.net
Integrator1\d4_ac.net
Integrator1\i1.net
Integrator1\i1_a.net
Integrator1\i1_ac.net
Integrator1\i1_ct.net
Integrator1\i2.net
Integrator1\i2_a.net
Integrator1\i2_ac.net
Integrator1\i2_ct.net
Integrator1\i3.net
Integrator1\i3_a.net
Integrator1\i3_ac.net
Integrator1\i3_ct.net
Integrator1\i4.net
Integrator1\i4_a.net
Integrator1\i4_ac.net
Integrator1\i4_ct.net
Integrator2\1.net
Integrator2\1_a.net
Integrator2\1_ac.net
Integrator2\2.net
Integrator2\2_a.net
Integrator2\2_ac.net
Integrator2\3.net
Integrator2\3_a.net

Integrator2\3_ac.net
Integrator2\4.net
Integrator2\4_a.net
Integrator2\4_ac.net
Sequence\ag.net
Sequence\g.net
Sinc\sinc.net
Sinc\sinc4.net
chebycheff\LP1a.net
Tchebycheff\LP1b.net
Tchebycheff\LP2a.net
Tchebycheff\LP2b.net
Tchebycheff\LP3a.net
Tchebycheff\LP3b.net
Tchebycheff\LP4a.net
Tchebycheff\LP4b.net
Tchebycheff\LP5a.net
Tchebycheff\LP5b.net
Tchebycheff\LP6a.net
Tchebycheff\LP6b.net
Tchebycheff\LP7a.net
Tchebycheff\LP7b.net
Tchebycheff\LP8a.net
Tchebycheff\LP8b.net
Thermometer\2.net
Thermometer\2b.net
Thermometer\4.net
Thermometer\4b.net
Thermometer\8.net
Thermometer\8b.net
Toggling\d.net
Toggling\i.net
Twist\twist.net
Twist\twist_bit.net
WFilter\A.net
WFilter\B.net
WFilter\C.net
WFilter\D.net
WFilter\Z.net

Cells

Adder_op1.dat
Adder_op2.dat
Array\10x1.dat
Array\10x2.dat
Array\10x3.dat
Array\11x1.dat
Array\11x2.dat
Array\11x3.dat
Array\12x1.dat
Array\12x2.dat
Array\12x3.dat
Array\13x1.dat
Array\13x2.dat
Array\14x1.dat
Array\14x2.dat
Array\15x1.dat
Array\15x2.dat
Array\16x1.dat
Array\16x2.dat
Array\17x1.dat
Array\18x1.dat
Array\1x1.dat
Array\1x10.dat
Array\1x11.dat
Array\1x12.dat
Array\1x13.dat
Array\1x14.dat
Array\1x15.dat
Array\1x16.dat
Array\1x17.dat
Array\1x18.dat
Array\1x2.dat
Array\1x3.dat
Array\1x4.dat
Array\1x5.dat
Array\1x6.dat
Array\1x7.dat
Array\1x8.dat
Array\1x9.dat
Array\2x1.dat
Array\2x10.dat
Array\2x12.dat
Array\2x13.dat
Array\2x14.dat
Array\2x15.dat
Array\2x16.dat
Array\2x11.dat
Array\2x2.dat
Array\2x3.dat
Array\2x4.dat
Array\2x5.dat
Array\2x6.dat
Array\2x7.dat
Array\2x8.dat
Array\2x9.dat
Array\3x1.dat
Array\3x10.dat
Array\3x11.dat
Array\3x12.dat
Array\3x2.dat
Array\3x3.dat
Array\3x4.dat
Array\3x5.dat
Array\3x6.dat
Array\3x7.dat
Array\3x8.dat
Array\3x9.dat
Array\4x1.dat
Array\4x2.dat
Array\4x3.dat
Array\4x4.dat
Array\4x5.dat
Array\4x6.dat
Array\4x7.dat
Array\4x8.dat
Array\4x9.dat
Array\5x1.dat
Array\5x2.dat
Array\5x3.dat
Array\5x4.dat
Array\5x5.dat
Array\5x6.dat
Array\5x7.dat
Array\6x1.dat
Array\6x2.dat
Array\6x3.dat
Array\6x4.dat
Array\6x5.dat
Array\6x6.dat
Array\7x1.dat
Array\7x2.dat
Array\7x3.dat
Array\7x4.dat
Array\7x5.dat
Array\8x1.dat
Array\8x2.dat
Array\8x3.dat
Array\8x4.dat
Array\9x1.dat
Array\9x2.dat
Array\9x3.dat
Array\9x4.dat
Array\c2_a.dat
Array\c3_a.dat
Array\c4_a.dat
Array\c4_b.dat
Array\c5_a.dat
Array\c5_b.dat
ASD_cmp2.dat
ASD_cmp3.dat
ASD_cmp4.dat
ASD_cmp5.dat
ASD_cmp6.dat
ASD_m1.dat
ASD_m11.dat
ASD_m2.dat
ASD_m211.dat
ASD_m21_va.dat
ASD_m22_va.dat
ASD_m22_vc.dat
ASD_m2ff.dat
ASD_op1.dat
ASD_op2.dat
ASD_op3.dat
ASD_op4.dat
ASD_op5.dat
Biquad_Martin_Sedra.dat
Comparator_cmp1.dat
Comparator_cmp2.dat
Integrator1_op1.dat
Integrator1_op2.dat
Integrator1_op3.dat
Integrator1_opreal.dat
Integrator2_op1.dat
Integrator2_op2.dat

Data Cells

acosh
arithmetic_mean
arithmetic_geometric_mean
asinh
atanh
bessel_i
bessel_j
bessel_k
bessel_y
c2f
c2k
centroidal_mean
choice_between_i
choice_between_d
choice_between_s
coherent
coherent_lindomain
coherent_linsweep
coherent_logdomain
coherent_logsweep
cmp3
compress_A_law
compress_A_law2
compress_mu_law
compress_mu_law2
contraharmonic_mean
db2lin
db2pow
d2i
dec2bin
deg2rad
diode_Iv
diode_Ri
diode_Rv
diode_Vi
dirac
dirac2
ET12
expand_A_law
expand_A_law2
expand_mu_law
expand_mu_law2
f2c
factorial
gaussian
geometric_mean
halton
hardlimiter
harmonic_mean
heronian_mean

GCD
I2d
isign
ispowerof2
k2c
LCM
lin2db
lindomain
linsweep
logdomain
log_factorial
logsweep
parallel_R
parallel_C
pow2db
p2t
kt
powerof2
prompt_for_double
prompt_for_long
prompt_for_yes_no
QR_01
QR_01_x
QR_gaussian
QR_normal
QR_uniform
QR_uniform_x
rad2deg
rand_01
rand_01_x
rand_bernoulli
rand_binomial
rand_chisquare
rand_equilikely
rand_erlang
rand_exponential
rand_gaussian
rand_halfnormal
rand_geometric
rand_lognormal
rand_normal
rand_pascal
rand_poisson
rand_rayleigh
rand_uniform
rand_uniform_x
randomize_array
reldif
rnoise
root_mean_square
round_it
serie_R

serie_C
serie_L
sinc
smoothlimiter
softlimiter
stuck_array
stuck_part_of_array
switch_i
switch_d
t2p
thermometric
vandercorput
vt

A_CONSTANT
ARITHMETIC_MEAN
ARITHMETIC_GEOMETRIC_MEAN
C0
C2F
C2K
CENTROIDAL_MEAN
CONTRAHARMONIC_MEAN
D2I
EV
EPSILON0
F2C
G
KT
GEOMETRIC_MEAN
H
HARMONIC_MEAN
HERONIAN_MEAN
I2D
K
K2C
ME
MU_CONSTANT
MU0
print_string
print_var
print_var_and_string
Q
RNOISE
ROOT_MEAN_SQUARE
SYSTEM_TIME
VT
Z0

and hundreds of ressources functions

C Functions and Macros

iuser_arithmetic_average
duser_arithmetic_average
duser_coherentwave
duser_comb
duser_ctm
iuser_dem
duser_dpll
duser_dsinc
duser_entropy
duser_fir
iuser_fir
duser_fir_in
duser_fir_out
duser_fm
iuser_fm
duser_geometric_average
duser_harmonic_average
duser_ifft
duser_ilt
duser_ilt2
iuser_lfsr
duser_median
iuser_median
duser_intreal
duser_multitone
duser_pink
duser_pulse
duser_pwl
iuser_pwl
duser_read
duser_read2
iuser_read
iuser_read2
duser_resonator
duser_rms_average
iuser_sequence
duser_sarc
duser_sine
iuser_stable
duser_sun
duser_synchro_lindomain
duser_synchro_logdomain
duser_synchro_linsweep
duser_synchro_logsweep
duser_synchro_readsweep
iuser_wave1x16_in
iuser_wave1x16_out
iuser_wave2x16_in
iuser_wave2x16_out

itool_autocorr
itool_cfft
itool_cgft
itool_cwin
itool_disto
itool_enbw
itool_fft
itool_fft_cs
itool_freq
itool_gdel
itool_gft
itool_hdecomp
itool_histobin
itool_history
itool_histogram
itool_histoslp
itool_histoval
itool_icn
itool_i2decomp
itool_i3decomp
itool_im2
itool_im3
itool_inform
itool_lag
itool_lin
itool_lsp
itool_lspwin
itool_output
itool_pdetect
itool_ps
itool_quinn2
itool_resp
itool_rms
itool_sinewave
itool_statslp
itool_statval
itool_synchro
itool_tdecomp
itool_tf
itool_tf2_i
itool_tf2_o
itool_tfp
itool_tsnr
itool_win
itool_xcorr

post_extrema
post_eye
post_histo
post_join
post_prune
post_select
post_sort
post_spice
post_stat
post_wcpk
post_wcpk_pwl
post_zip

User Functions and Smart Tools

Conclusions

NAPA is a Fully Open
High Level Mixed Signal Simulator.

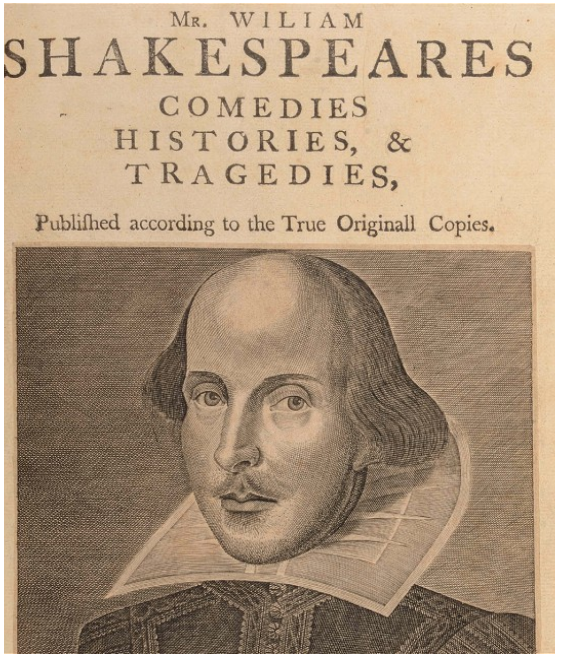
You are the pilot.
Your creativity is your limitation.

SIMULATE WITH MODERATION
and intelligence...

Questions ?

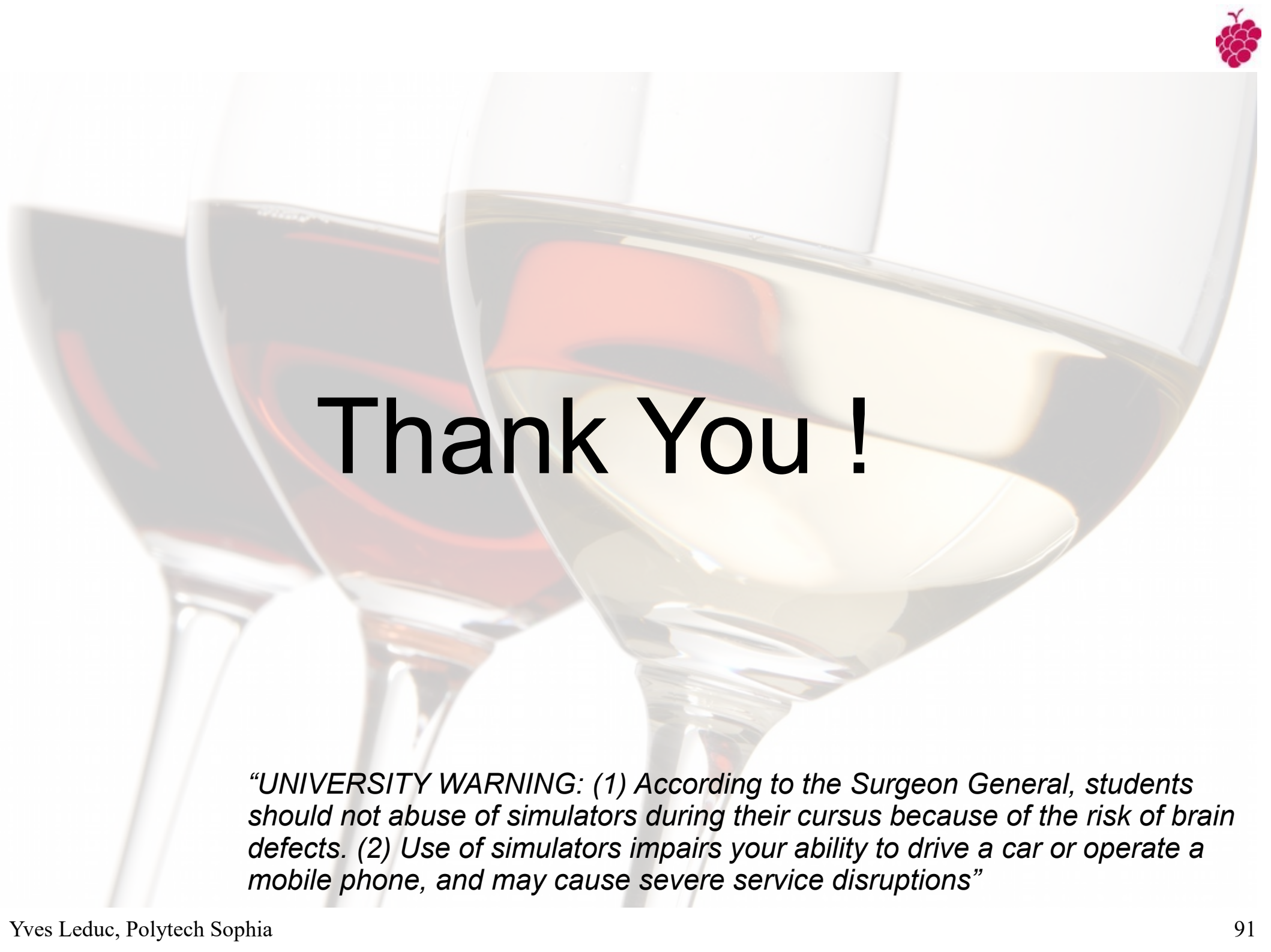
The Last Words

" Good model is a good familiar creature
if it be well used " [1]



[1] 309 Come, come, good wine is a good familiar creature
310 if it be well used"

William Shakespeare (1564-1616)
Othello, II. Iii



Thank You !

“UNIVERSITY WARNING: (1) According to the Surgeon General, students should not abuse of simulators during their cursus because of the risk of brain defects. (2) Use of simulators impairs your ability to drive a car or operate a mobile phone, and may cause severe service disruptions”

Appendix

HWiNFO64 @ ASUS B33E - System Summary

CPU

Intel Core i5-2450M — CPU #0

Stepping	J1	Cores	2
Codename	Sandy Bridge-MB SV	Logical	4
Cache	2x32 + 2x32 + 2x256 + 3M	µCU	25
Prod. Unit	Platform	Socket G2 (rPGA988B)	
TDP 35 W	SSPEC	SR0CH,SR04X	

Features

MMX	3DNow!	3DNow!-2	SSE	SSE-2	SSE-3	SSSE-3
SSE4A	SSE4.1	SSE4.2	AVX	AVX2	AVX-512	
BMI2	ABM	TBM	FMA	ADX	XOP	
DEP	VMX	SMX	SMEP	SMAP	TSX	MPX
EM64T	EIST	TM1	TM2	HTT	Turbo	
AES-NI	RDRAND	RDSEED	SHA			

Operating Point	Clock	Ratio	Bus	VID
CPU LFM (Min)	800.0 MHz	8.00x	100.0 MHz	-
CPU HFM (Max)	2500.0 MHz	25.00x	100.0 MHz	-
CPU Turbo	3100.0 MHz	31.00x	100.0 MHz	-
CPU Status	-	-	99.8 MHz	0.7705 V

	Clock	Ratio	ThermMon
Core0	798 MHz	8.00x	OK
Core1	798 MHz	8.00x	OK

GPU

Intel Sandy Bridge-MB GT2+ - Integrated Grap

Intel HD Graphics 3000

Sandy Bridge GT2+

PCI

GPU #0 — 2108 MB — - — ROPs - — Shaders -

Current Clocks (MHz)

GPU 650.0 — Memory 665.0 — Shader -

Motherboard: ASUS B33E

Chipset: Intel HM65 (Cougar Point) [B3]

BIOS: 02/17/2012 — BIOS Version: B33E.206

Memory

Size: 8192 MB — Type: DDR3 SDRAM

Current Timing

Clock: 665.3 MHz = 6.67 x 99.8 MHz

Mode: Dual-Channel — CR: 1T

Timing: 9 - 9 - 9 - 24 — tRC: — tRFC: 107

Modules

[#0] Hynix (Hyundai) HMT351S6CFR8C-H9

Size: 4096 MB — Clock: 667 MHz — ECC: N

Type: PC3-10600 DDR3 SDRAM SO-DIMM

Freq	CL	RCD	RP	RAS	RC	Ext.	V
666.7	9	9	9	24	33	-	1.50
600.0	8	8	8	22	30	-	1.50
533.3	7	7	7	20	27	-	1.50
400.0	6	6	6	15	20	-	1.50
333.3	5	5	5	12	17	-	1.50

Drives

Interface	Model
SATA 3 Gb/s	WDC WD5000BPKT-80PK4T0 [500 GB, 16MB]
SATA 1.5 Gb/s	MATSHITADVD-RAM UJ8A2ASW [DVD+R DL]

OS: Microsoft Windows 7 Professional (x64) Build 7601 — Close

FYI, my computer