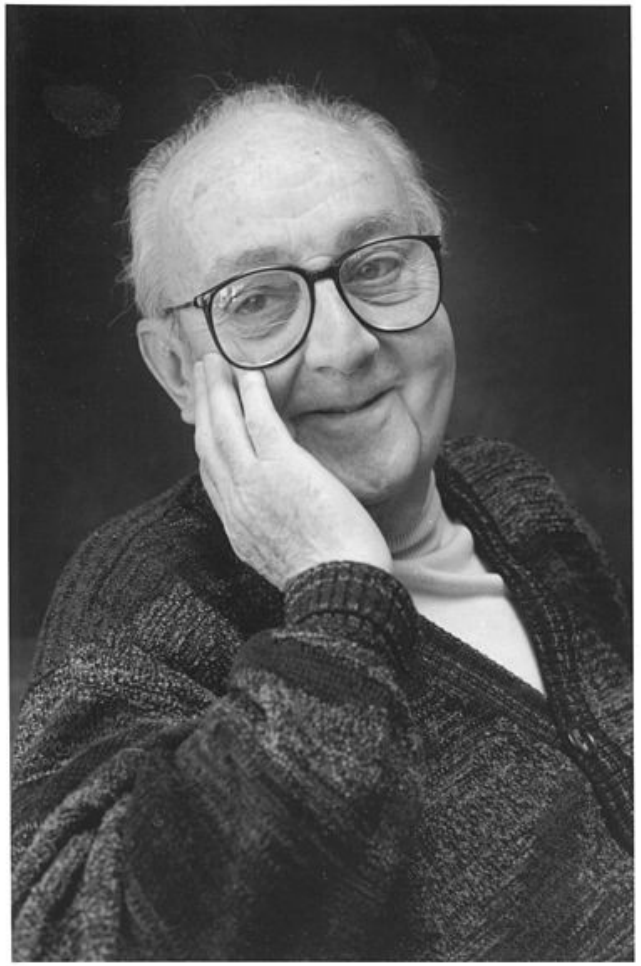
Preamble



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

a quotation attributed to the late 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



N A P A 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

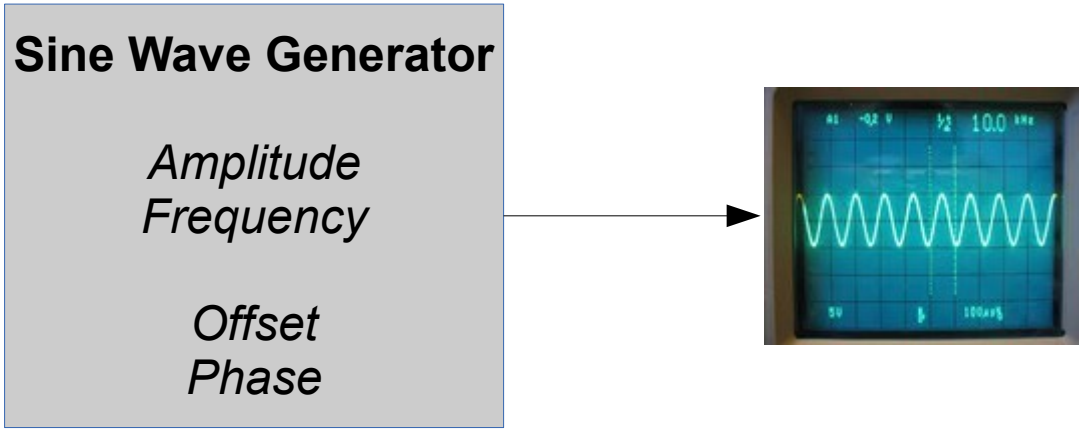
- ***main netlist***
- ***cell***
- ***primitive***
- ***user function***
- ***time domain simulation***
- ***smart tool***
- ***synchronization***

A realistic example

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

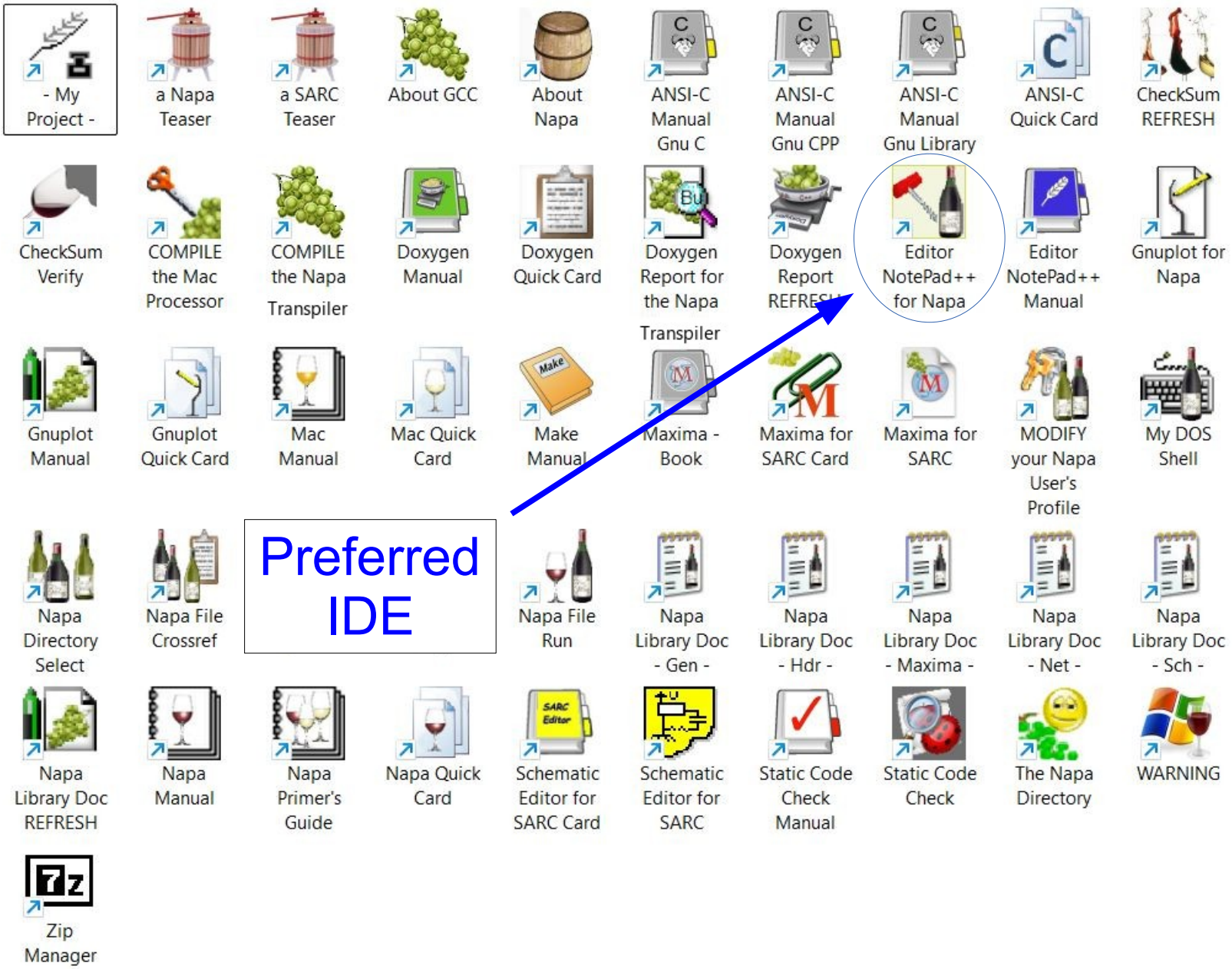
Open conclusion

A Simple Example : Generate a Sine Wave



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

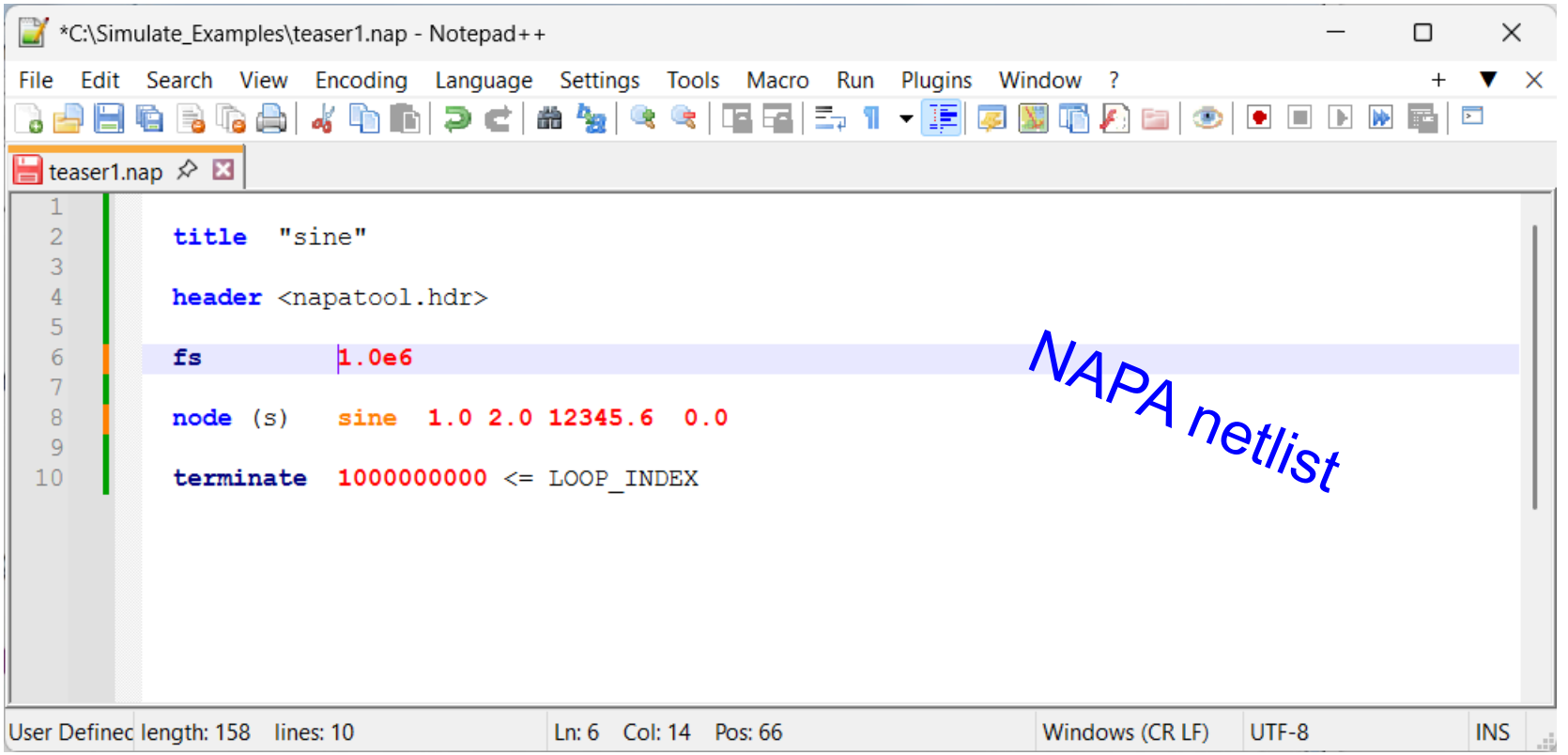
Preferred Environment : Windows 11, 64 bits



(Napa Vineyard)

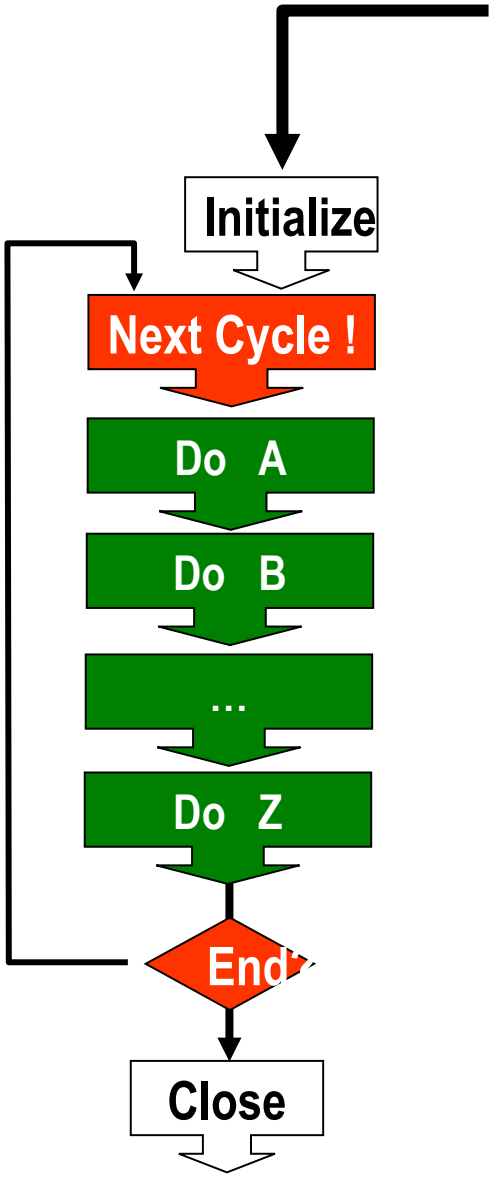
First Contact with NAPA

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



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

NAPA is a Cycle-Based Simulator Writer



Flow transpiled before execution

Crystal clear flow

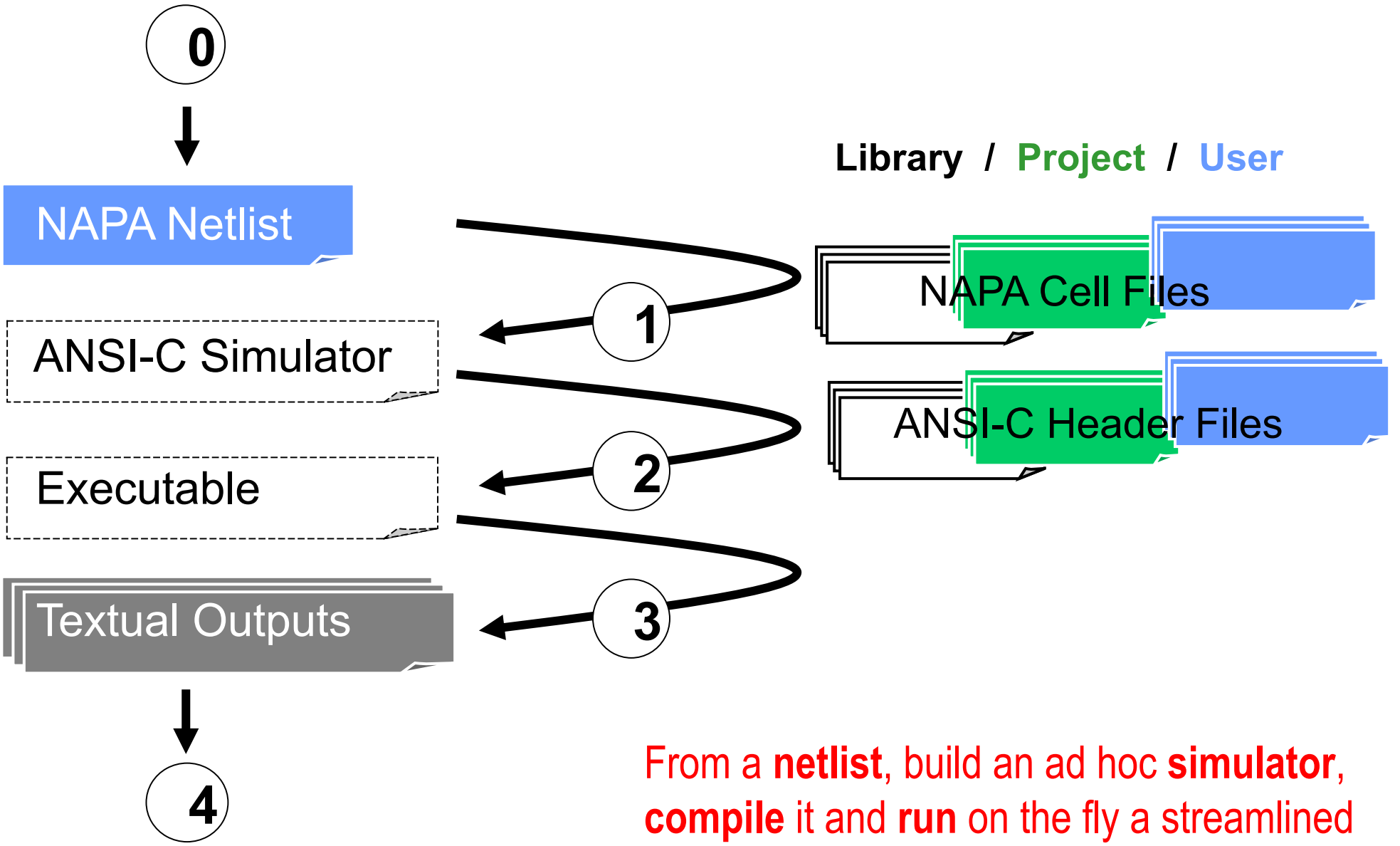
DATA Domain !
Control Domain ?

Mixed Signal
Data Processing

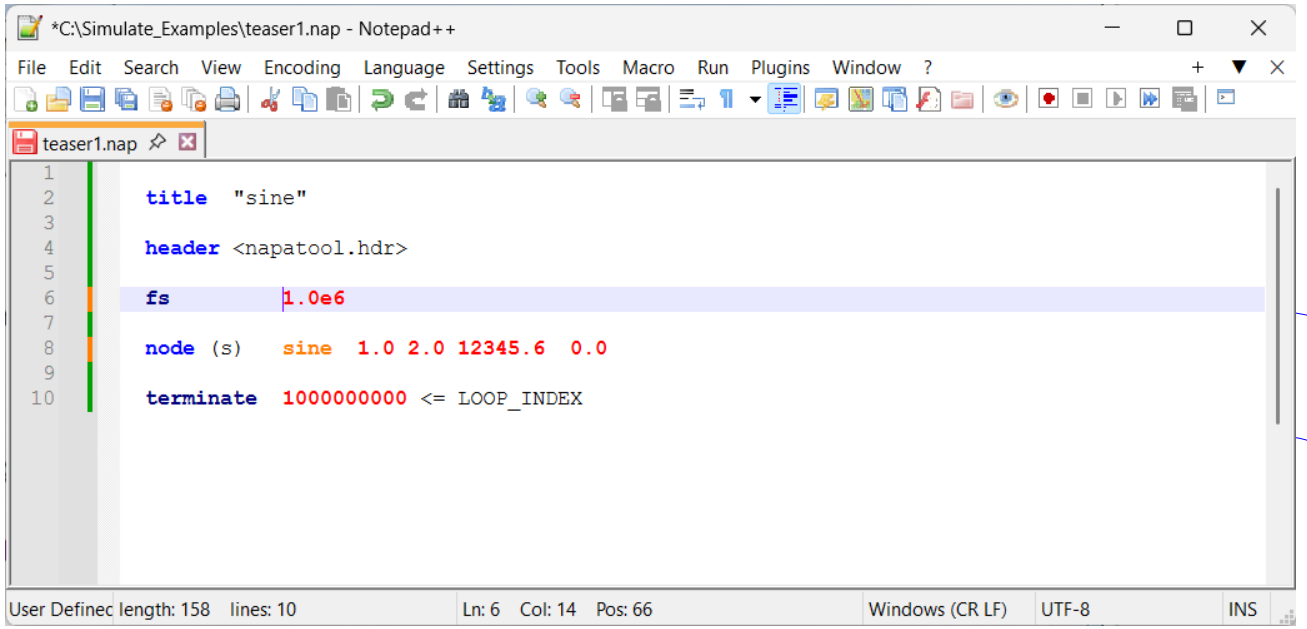
FFT friendly
Hazard free
Cycle accurate by construction
Very fast

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 fast **executable**.



NAPA transpilation

file "teaser01.c"

```
...
napa_abs_loop = 0.0L;

do {
napa_abs_time  = napa_abs_loop * 1.000000000000000e-06L;

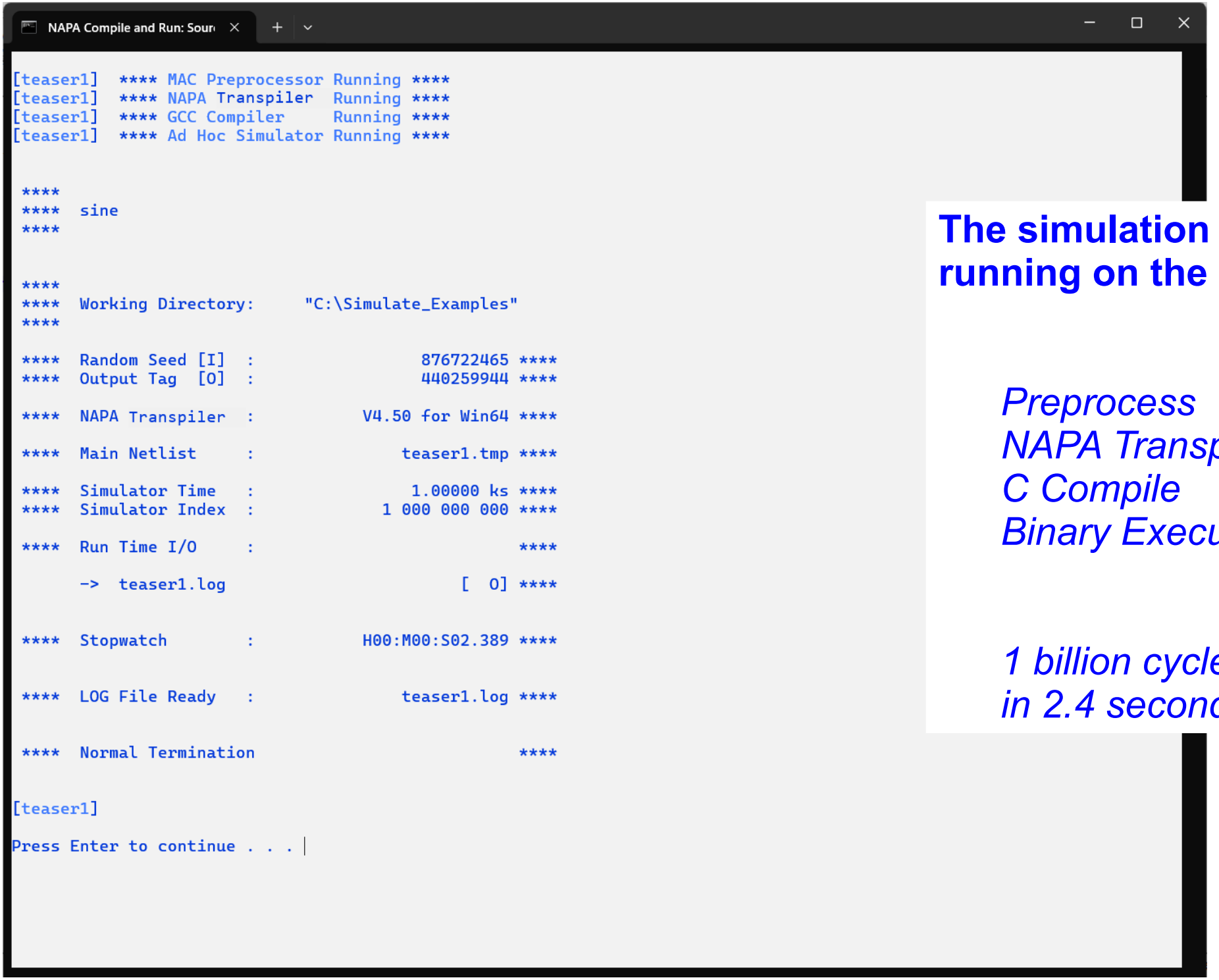
dnode_s = 1.0 + (2.0) *
((R_TYPE) sinl(77569.692528316305093483152L * napa_abs_time));

napa_abs_loop++;

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

ANSI-C code

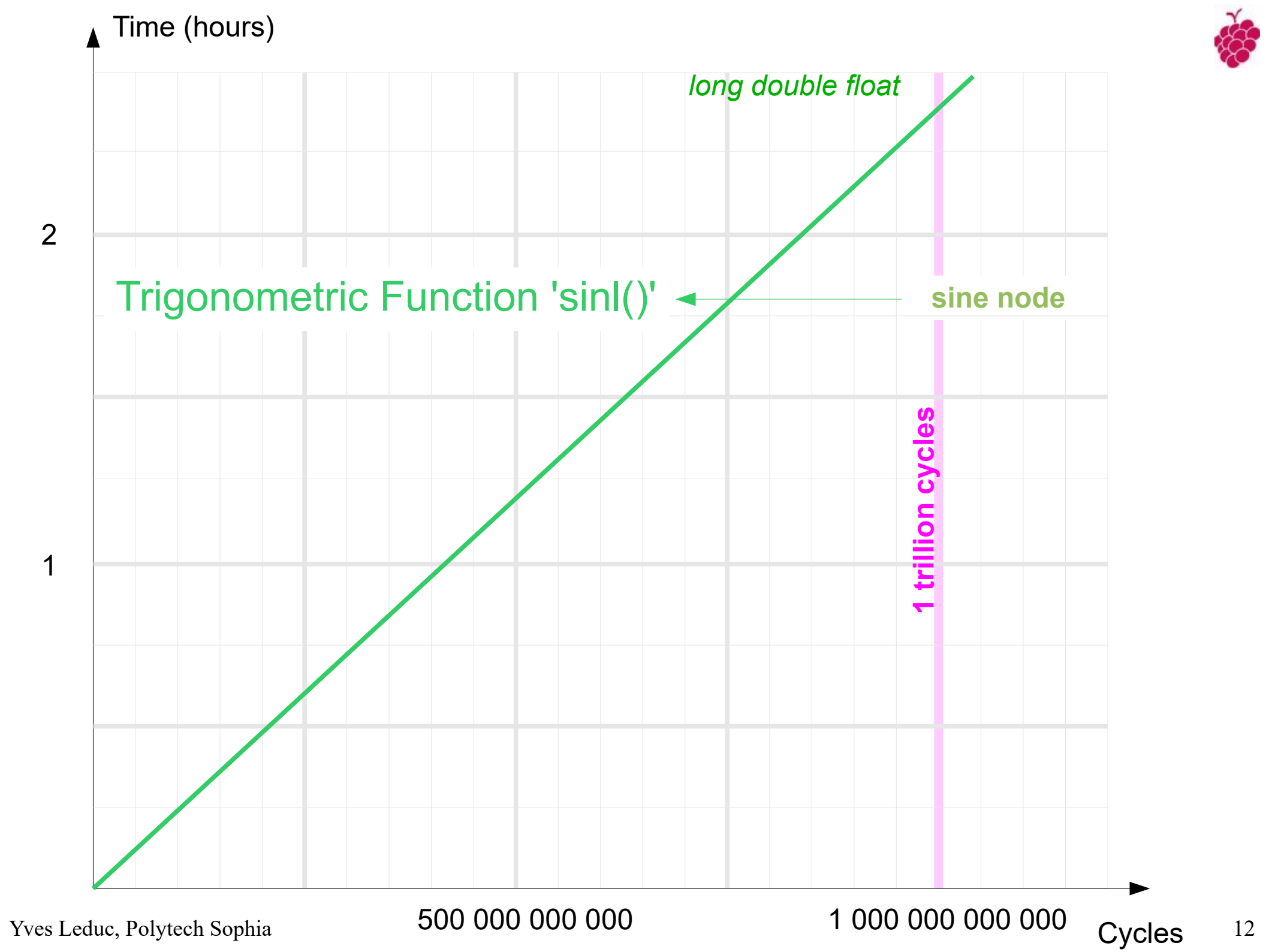
*Tips : toggle to the ANSI-C code, press **'Alt T'** from the NAPA netlist*



The simulation running on the screen

Preprocess
NAPA Transpile
C Compile
Binary Execution

1 billion cycles
in 2.4 seconds


Time (hours)
2
1
long double float
Trigonometric Function 'sinl()'
sine node
1 trillion cycles
500 000 000 000
1 000 000 000 000
Cycles

We will implement a few ideas to generate a sine wave.

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 advantage of this property.

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

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

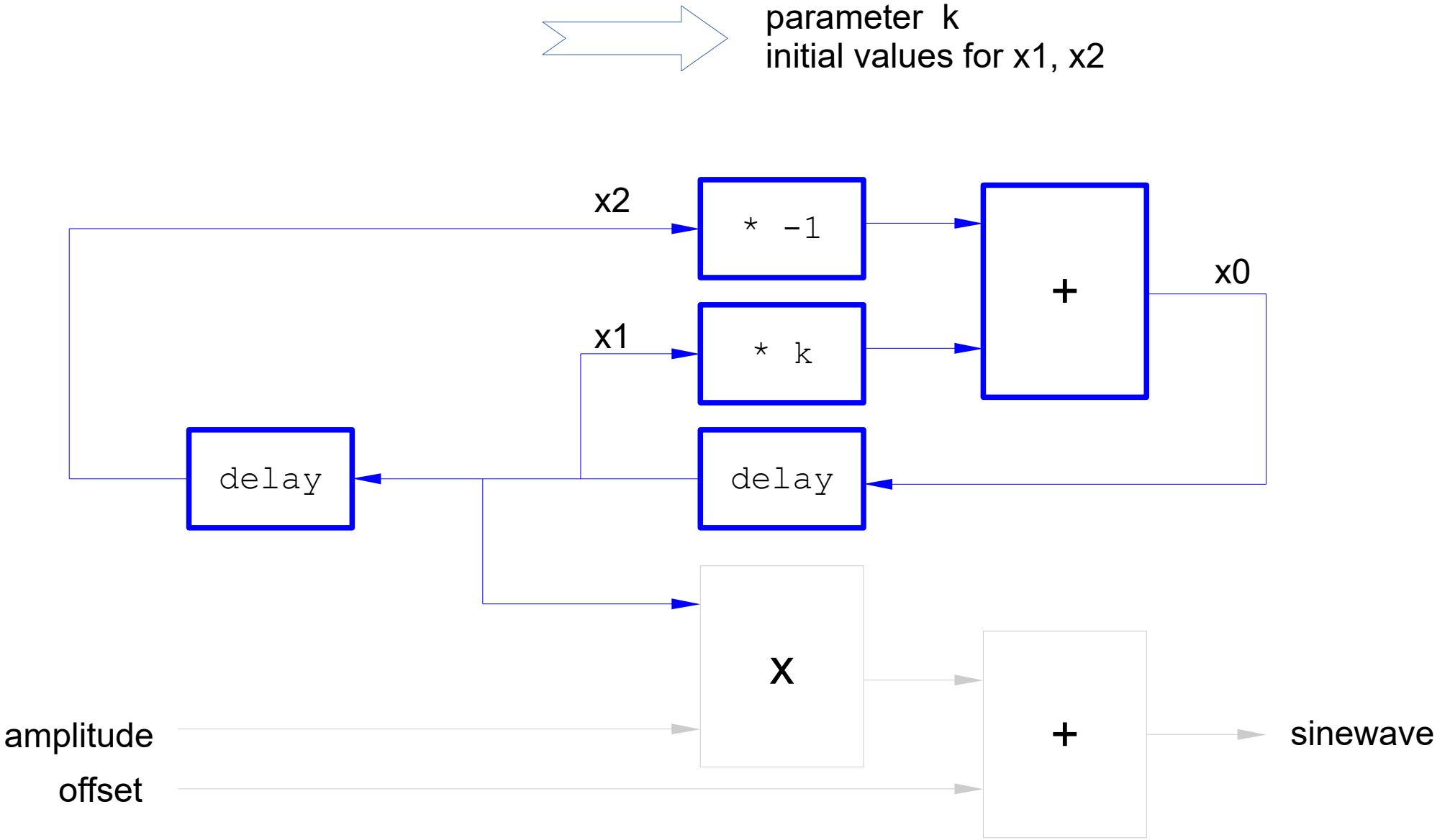
$$\mathbf{X_n = (k * X_{n-1}) - X_{n-2}}$$
$$\textit{with } k = 2.0 * \cos(2\pi \ f_{sinewave} / f_{sampling})$$

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

$$X_{n-1} = \sin(phase)$$
$$X_{n-2} = \sin(phase - (2\pi \ f_{sinewave} / f_{sampling}))$$

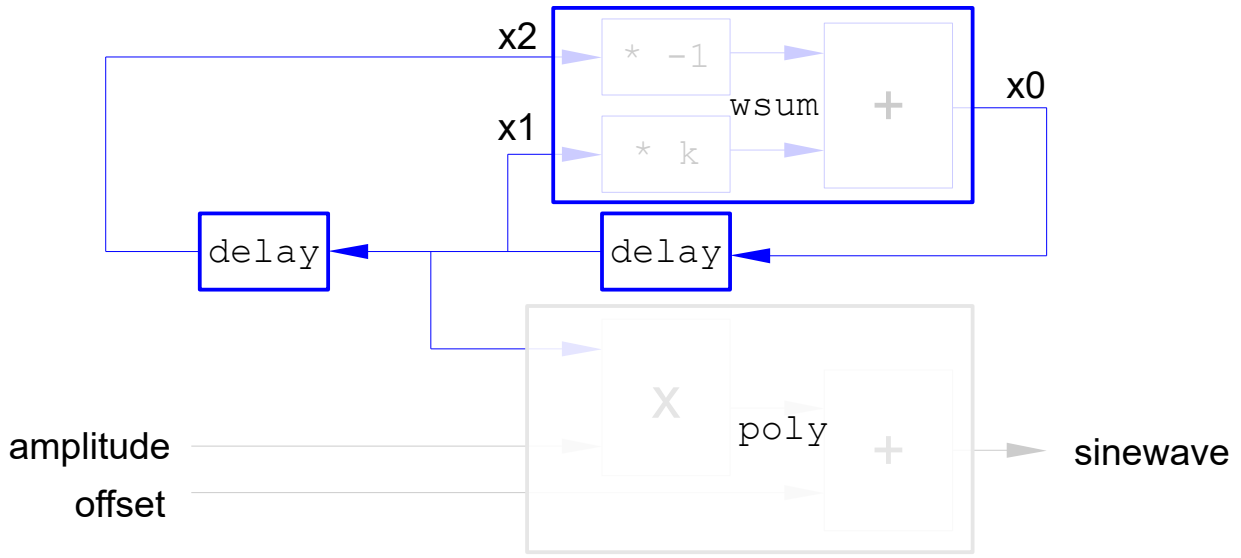
The Second Order Resonator

Sine wave frequency, sine wave phase and sampling frequency





A Direct Description in a NAPA Netlist



C:\Simulate_Examples\teaser2.nap - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

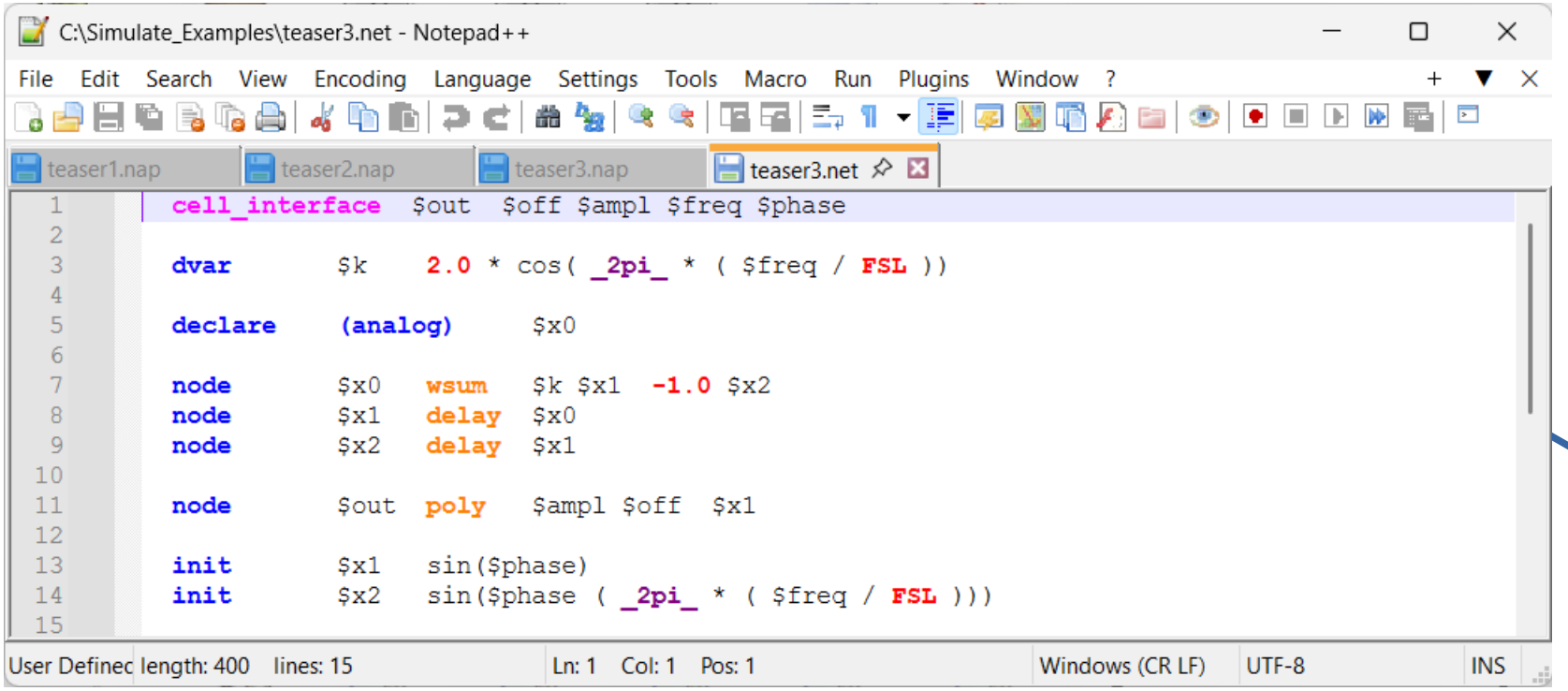
teaser3.net | teaser1.nap | teaser2.nap

```
1
2   title       "resonator"
3
4   header      <napatool.hdr>
5
6   fs          1.0e6
7
8   dvar        k     2.0 * cos( _2pi_ * ( 12345.6 / FSL ))
9
10  node        x0    wsum    k  x1  -1.0 x2
11  node        x1    delay   x0
12  node        x2    delay   x1
13  node        s     poly    2.0 1.0  x1
14
15  declare     (analog)      x0 x1 x2
16
17  init        x1    sin(0.0)
18  init        x2    sin(0.0 - ( _2pi_ * ( 12345.6 / FSL )))
19
20  terminate   1000000000 <= LOOP_INDEX
21
22  directive   LOGFILE
```

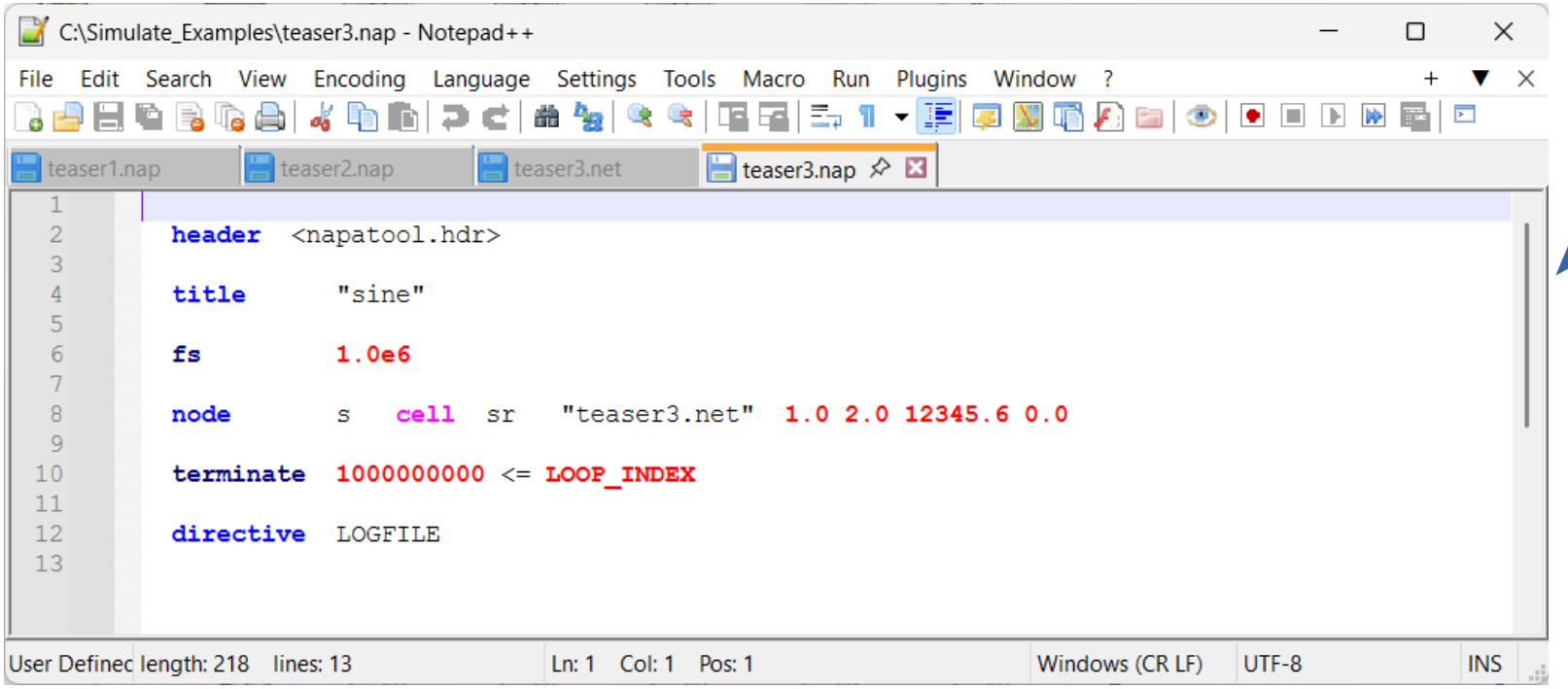
User Defined length: 482 lines: 22 | Ln: 1 Col: 1 Pos: 1 | Windows (CR LF) | UTF-8 | INS



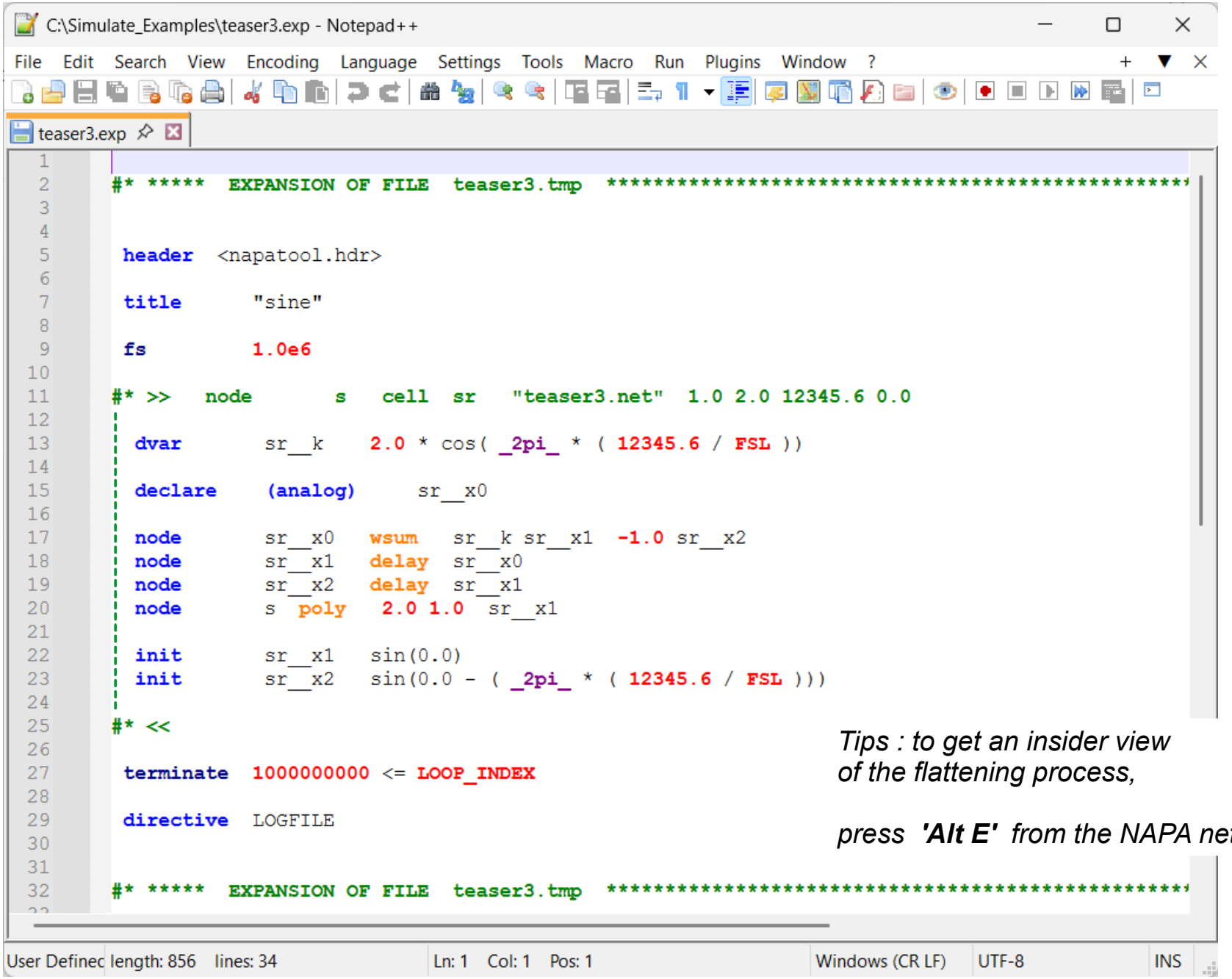
A Description Using a NAPA Cell



instantiation



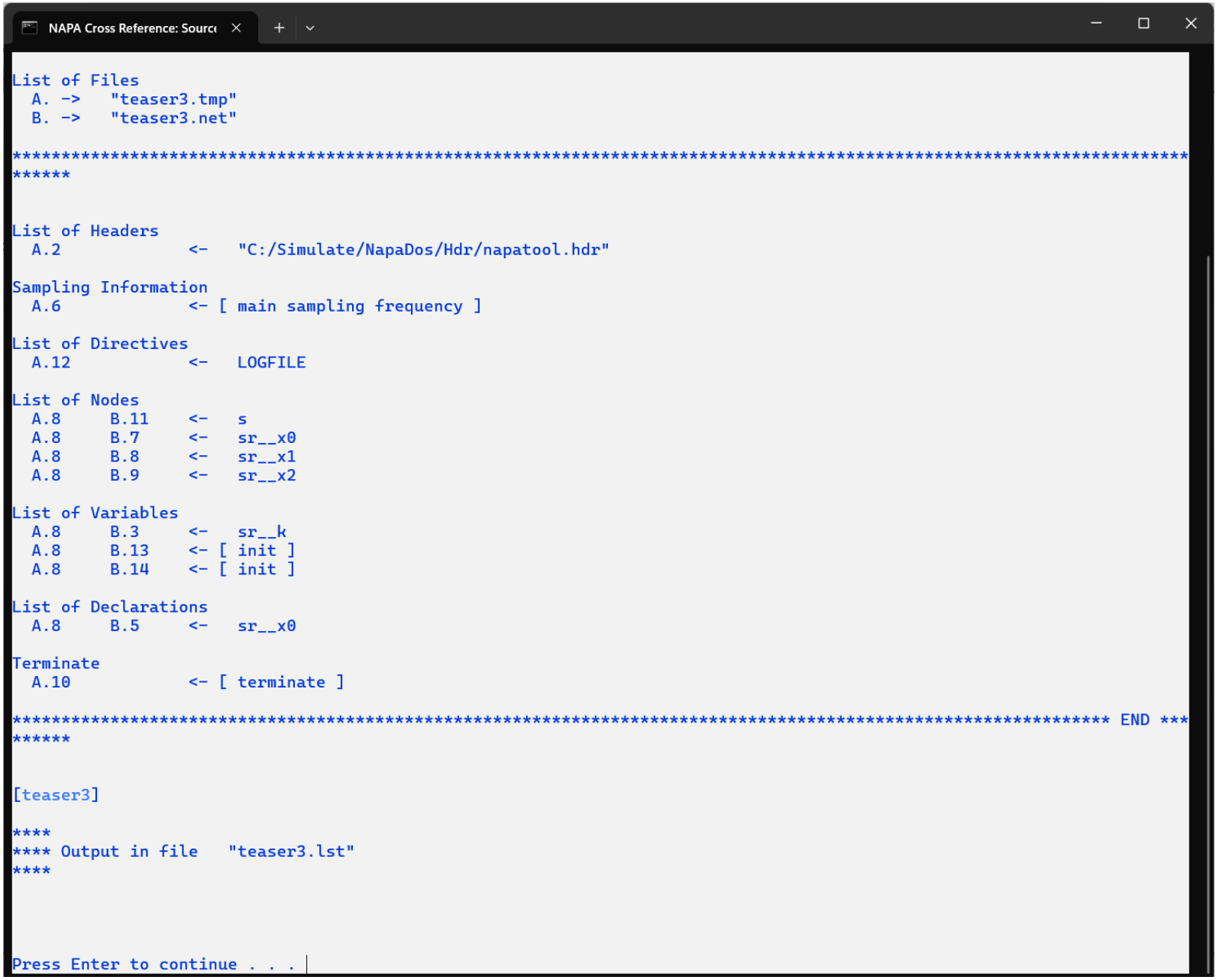
The NAPA transpiler flattens the hierarchy before the process.



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

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

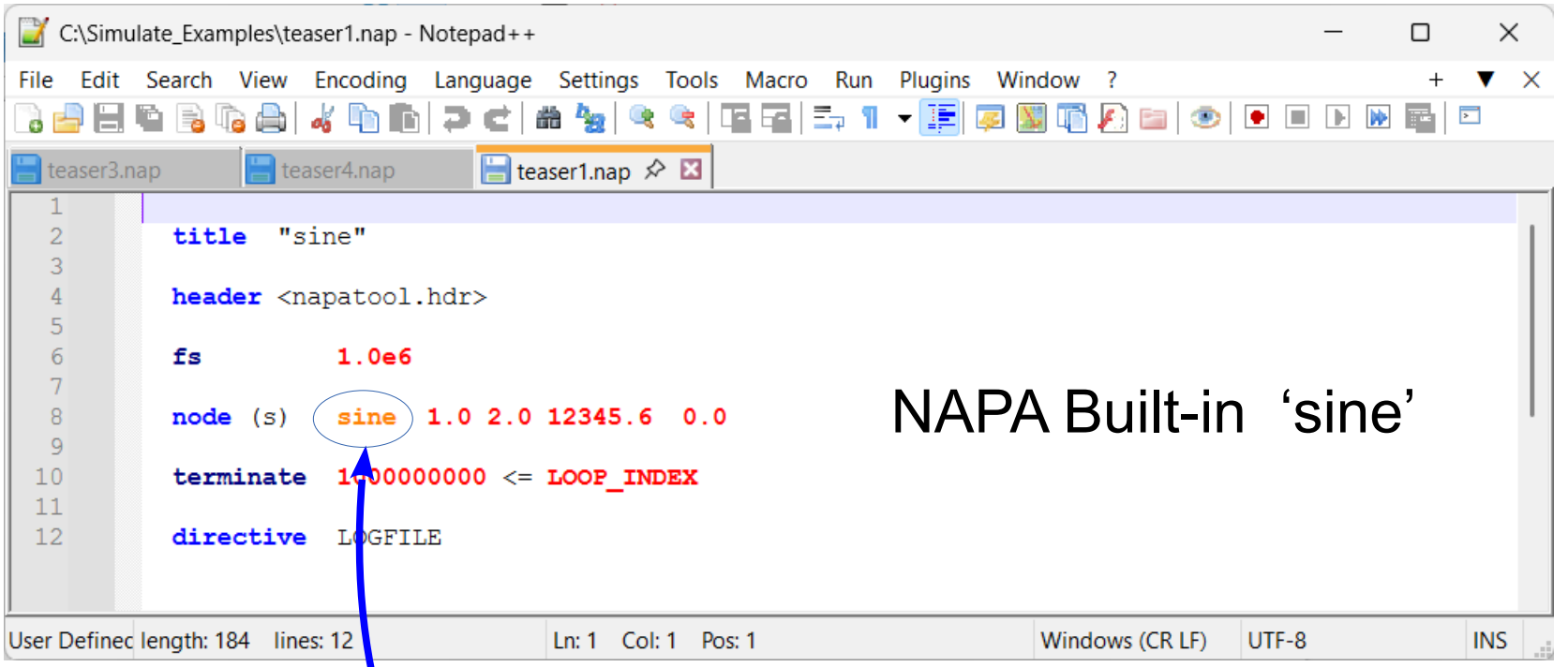
The NAPA transpiler creates a cross reference of the flattened NAPA netlist.



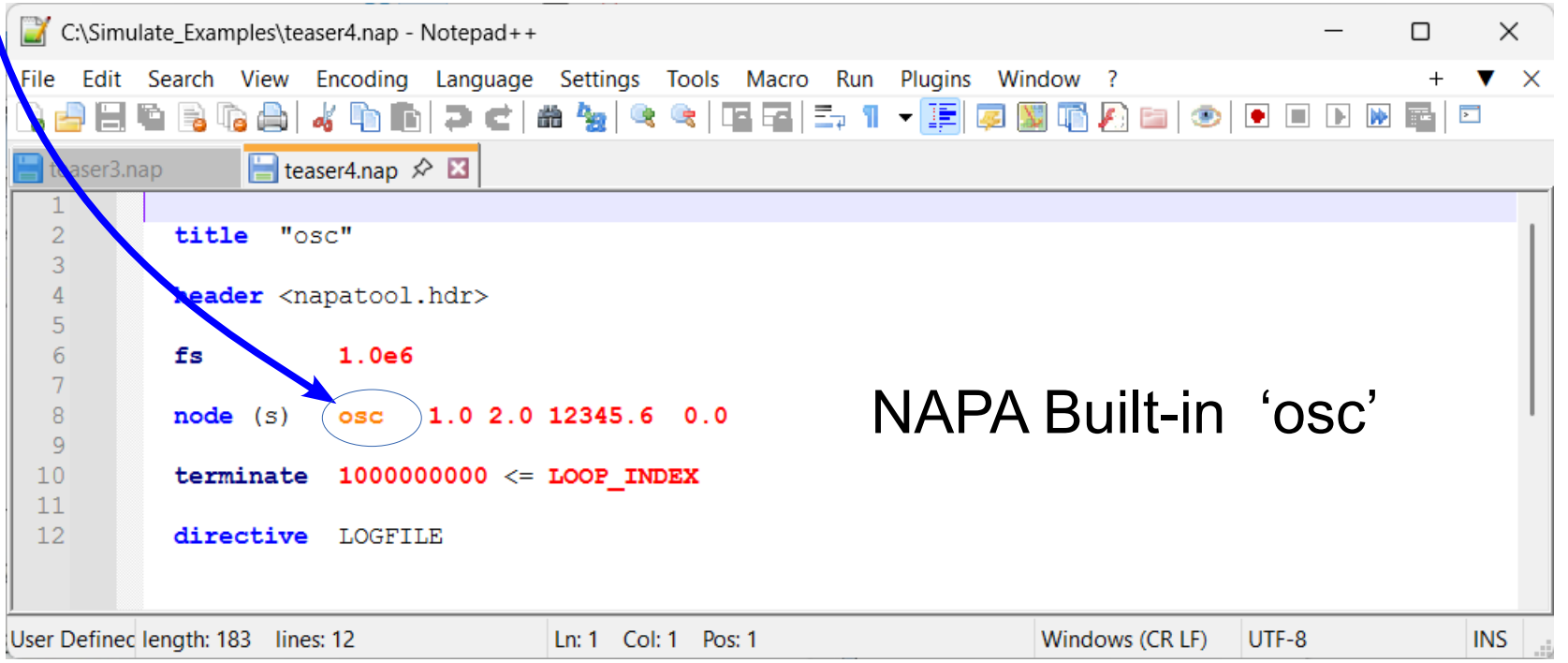


A Description Using a NAPA Primitive

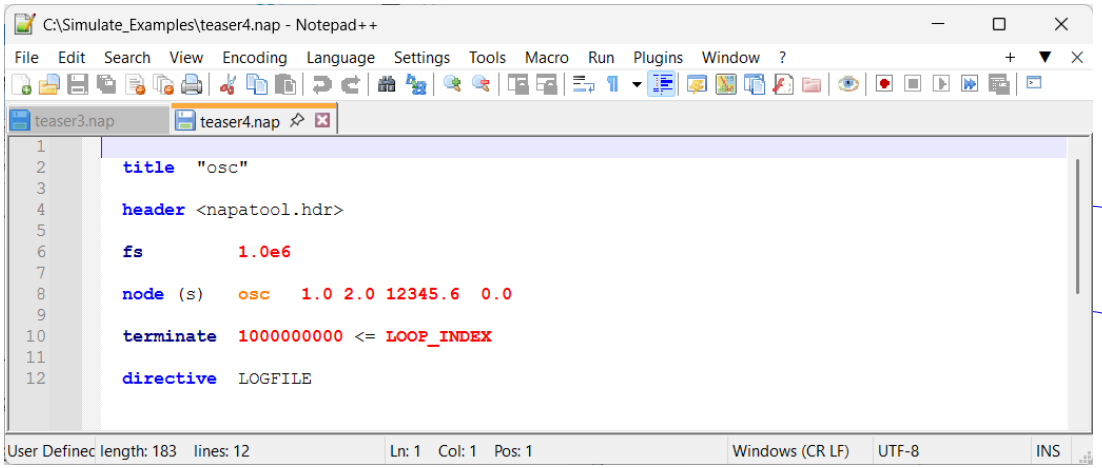
(in fact, this solution is also implemented as a built-in primitive)



NAPA Built-in 'sine'



NAPA Built-in 'osc'



NAPA transpilation

file "teaser04.c"

```
...
hnode_s_fctr = cosl(_2PI_*12345.6L/FSL_H) * 2.0L;
hnode_s_osc0 = 0.0L;
hnode_s_osc1 = -sinl(_2PI_*12345.6L/FSL_H);
hnode_s_osc2 = 0.0L;
...
napa_abs_loop = 0.0L;
do {
napa_abs_time  = napa_abs_loop * 1.000000000000000e-06L;


hnode_s_osc2 = hnode_s_osc1;
hnode_s_osc1 = hnode_s_osc0;
hnode_s_osc0 = (hnode_s_fctr * hnode_s_osc1) - hnode_s_osc2;
dnode_s = ((R_TYPE) hnode_s_osc1) * 2.0;
dnode_s += 1.0;

napa_abs_loop++;

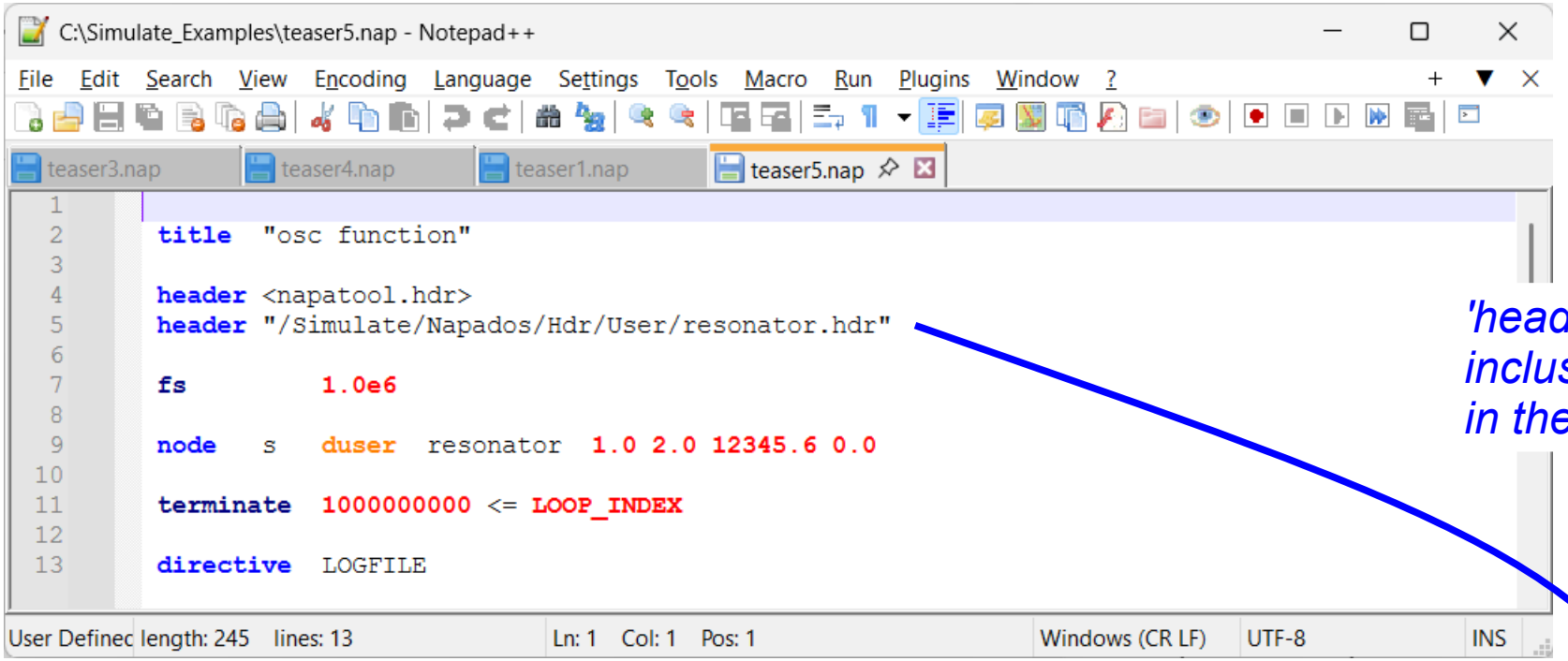
} while (!TERMINATE);
...
```

The resonator is implemented internally with 16-byte long double floats to get the best precision. Output is cast to 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

inside the file "/Simulate/Napados/Hdr/User/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)
```

C:\Simulate_Examples\teaser5.nap - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

teaser3.nap | teaser4.nap | teaser1.nap | teaser5.nap

```
1
2   title  "osc function"
3
4   header <napatool.hdr>
5   header "/Simulate/Napados/Hdr/User/resonator.hdr"
6
7   fs          1.0e6
8
9   node    s   duser  resonator  1.0 2.0 12345.6 0.0
10
11  terminate  1000000000 <= LOOP_INDEX
12
13  directive  LOGFILE
```

User Defined length: 245 lines: 13 | Ln: 1 Col: 1 Pos: 1 | Windows (CR LF) | UTF-8 | INS

NAPA transpilation

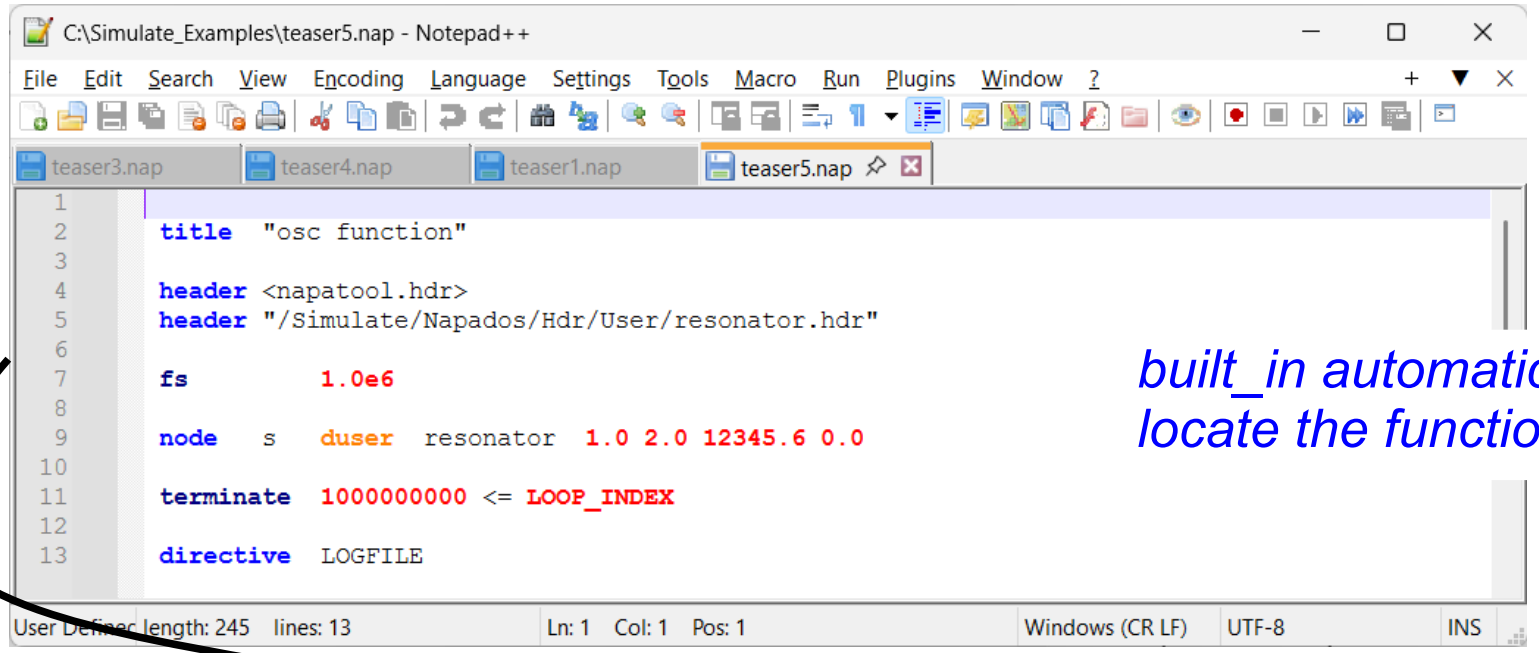
4 parameters

number of this instance in the netlist

file "teaser05.c"

instance ID

```
 ...
#define COMPILE_duser_resonator 1
...
#include  "C:/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 * 1.0000000000000e-06L;
  dnode_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);
...
```



built_in automatic method to locate the function using a table

in "Hdr/User/_user.lib", included through "napatool.h

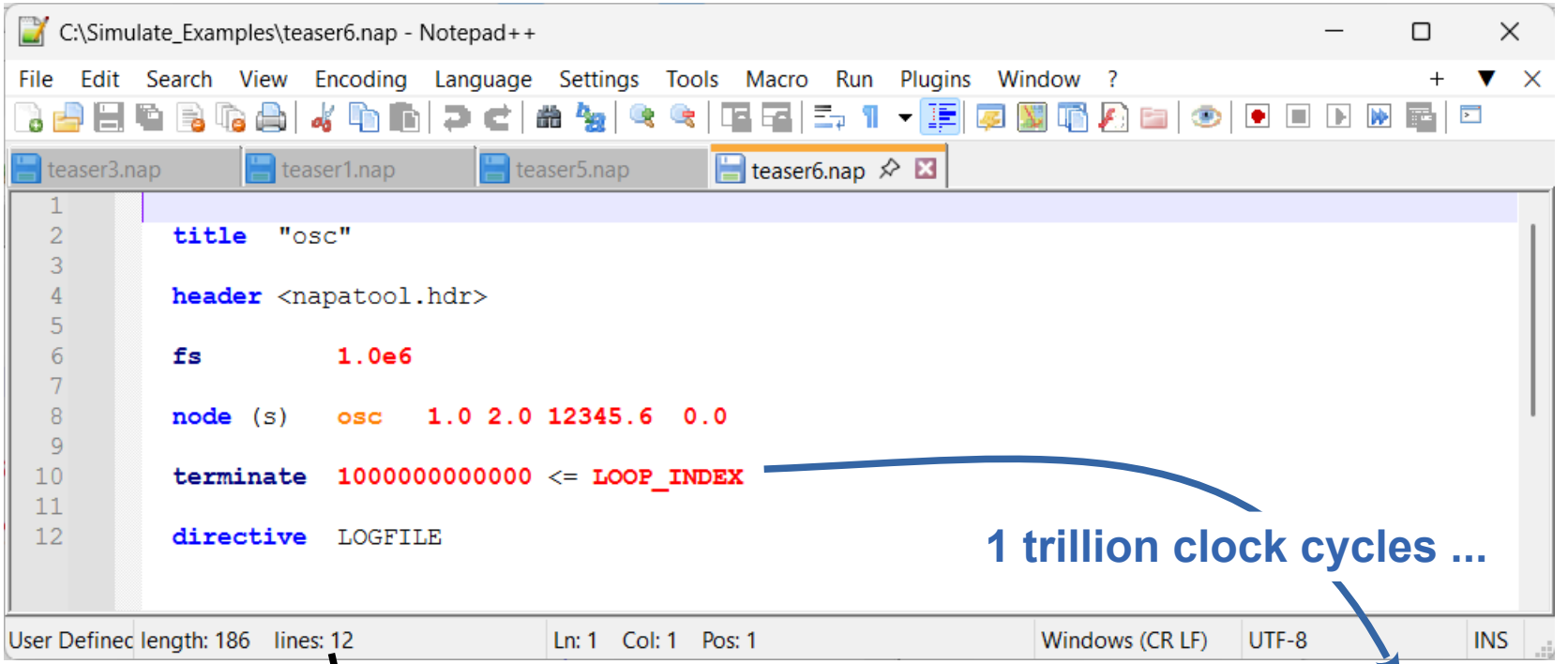
```
...
#ifdef COMPILE_duser_resonator
  #include "resonator.hdr"
#endif
...
```

file "/Simulate/Napados/Hdr/User/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) ...
```

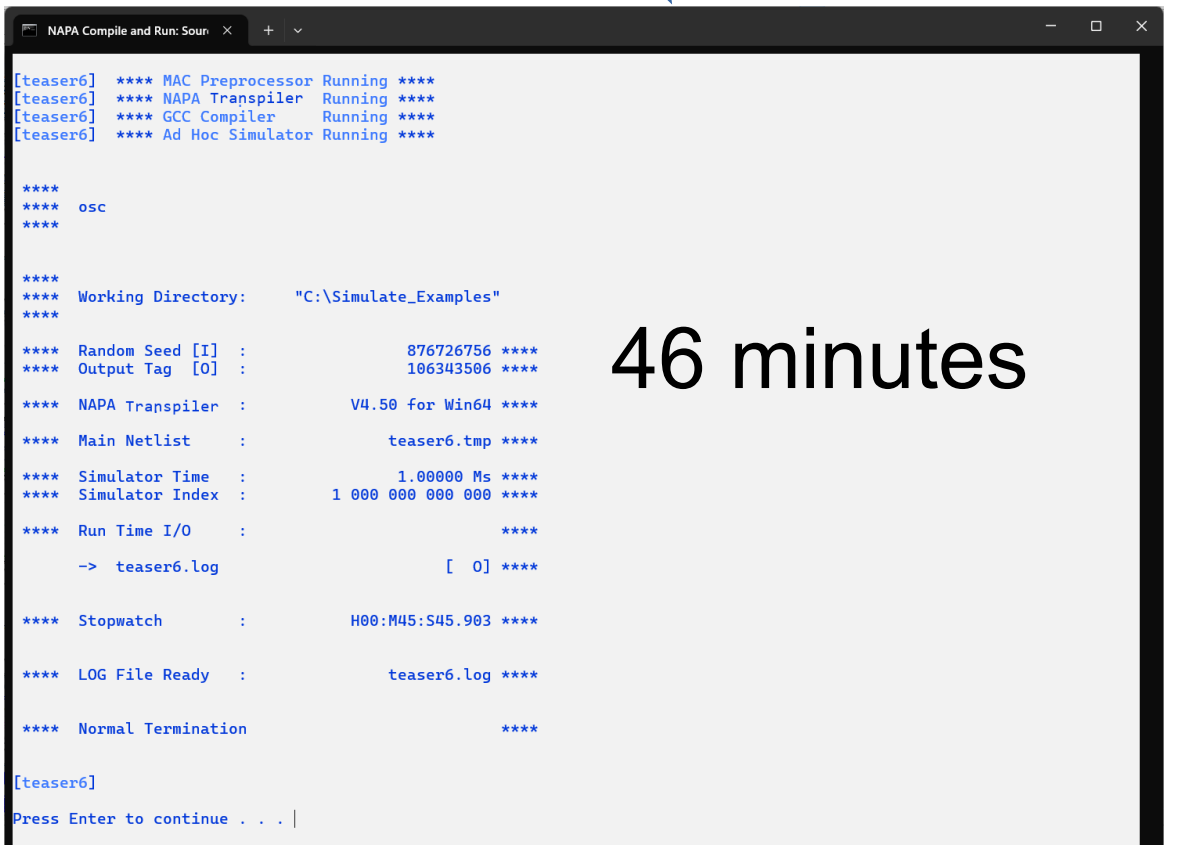


We aimed to get speed, didn't we ?



1 trillion clock cycles ...

Run Simulation

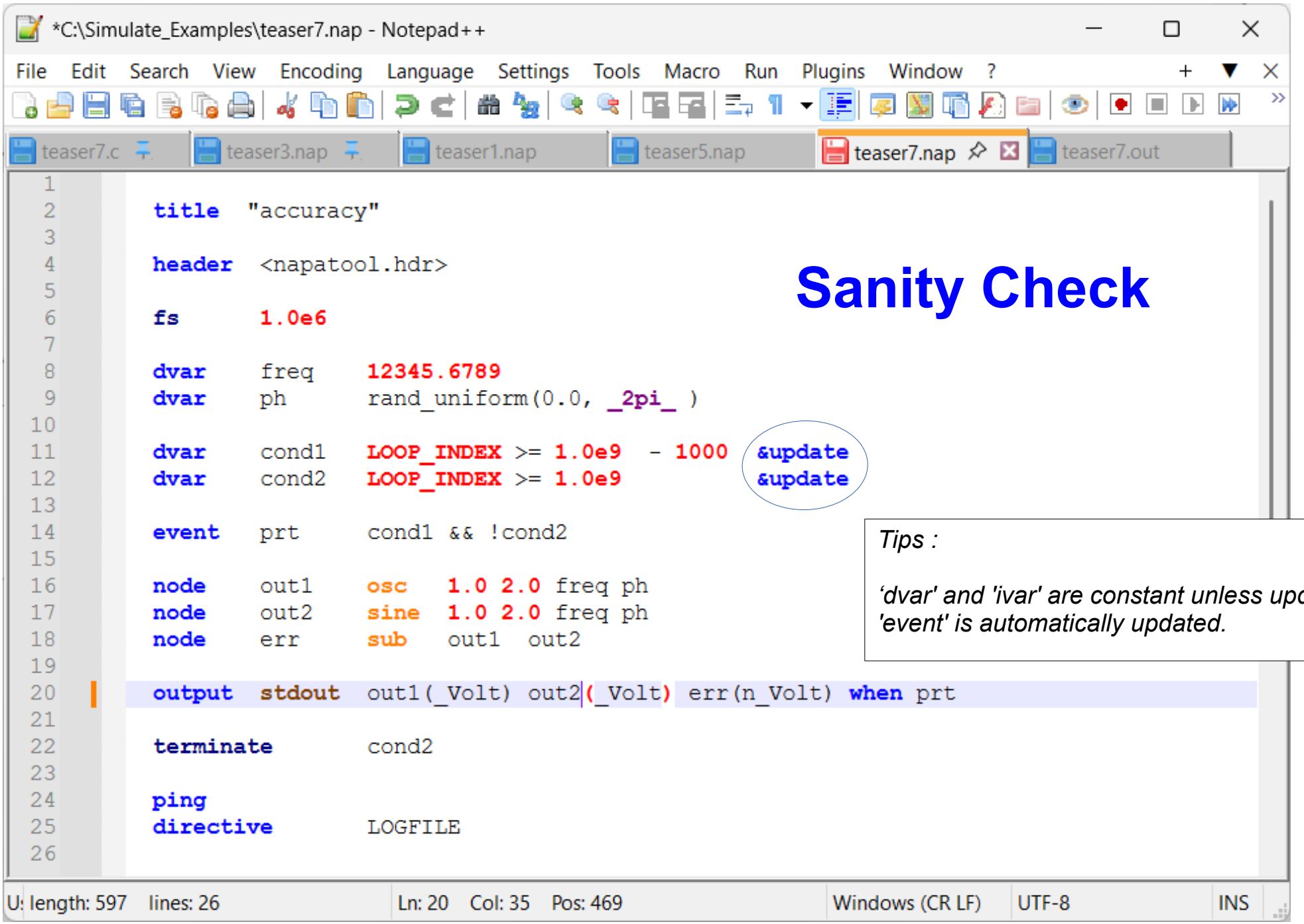


46 minutes



Cool...

But we do not want to lose 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 = 0.0L;

...

dvar_freq   = 12345.6789;
dvar_ph     = rand_uniform(0.0,_2pi_);
dvar_cond1  = LOOP_INDEX>=1.0e9-1000.0;
dvar_cond2  = LOOP_INDEX>=1.0e9;
ivar_prt    = (I_TYPE)(dvar_cond1&&!dvar_cond2);

dnode_out1  = 0.0;
hnode_out1_fctr = cosl(_2PI_*dvar_freq/((H_PREC) FSL)) * 2.0L;
hnode_out1_osc0 = sinl(dvar_ph);
hnode_out1_osc1 = sinl(dvar_ph - (_2PI_*dvar_freq/((H_PREC) FSL)));
hnode_out1_osc2 = 0.0L;
dnode_out2  = 0.0;
dnode_err   = 0.0;

do {

  napa_abs_time  = napa_abs_loop * 1.0e-06L;

  dvar_cond1  = LOOP_INDEX>=1.0e9-1000.0;
  dvar_cond2  = LOOP_INDEX>=1.0e9;
  ivar_prt    = (I_TYPE)(dvar_cond1&&!dvar_cond2);

  hnode_out1_osc2 = hnode_out1_osc1;
  hnode_out1_osc1 = hnode_out1_osc0;
  hnode_out1_osc0 = (hnode_out1_fctr * hnode_out1_osc1) - hnode_out1_osc2;
  dnode_out1  = ((R_TYPE) hnode_out1_osc1) * 2.0;
  dnode_out1 += 1.0;
  dnode_out2  = 1.0 + (2.0) * ((R_TYPE) sinl(_2PI_ * ((H_PREC) dvar_freq) * napa_abs_time + ((H_PREC) (dvar_ph))));
  dnode_err   = (dnode_out1) - (dnode_out2);

  if ( ivar_prt ) {
    (void) fprintf(stdout, "%.15Le % .12e % .12e % .12e\n", napa_abs_time, dnode_out1, dnode_out2, dnode_err*1.0e9);
  }

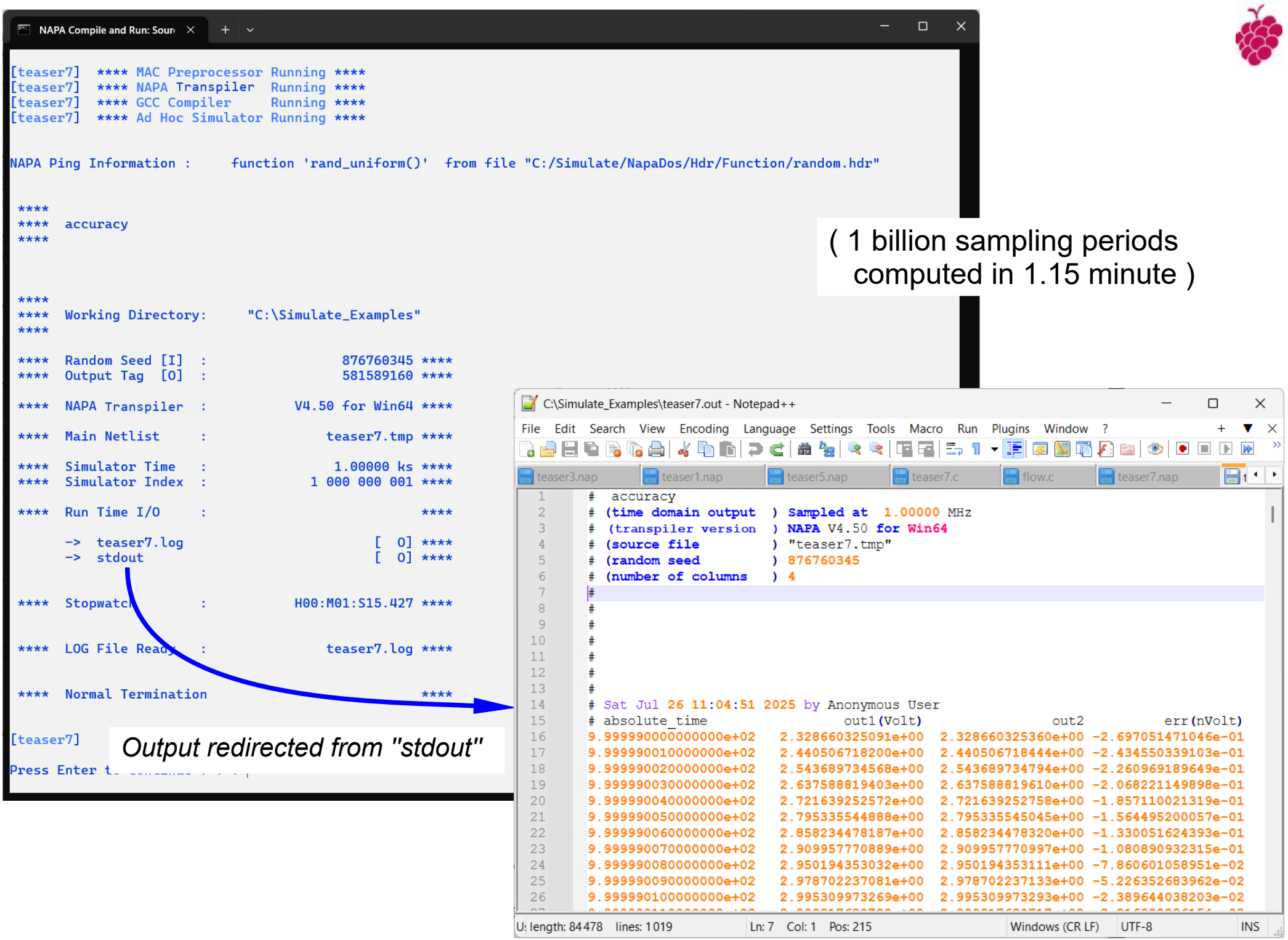
  napa_abs_loop++;
} while ( !dvar_cond2 );

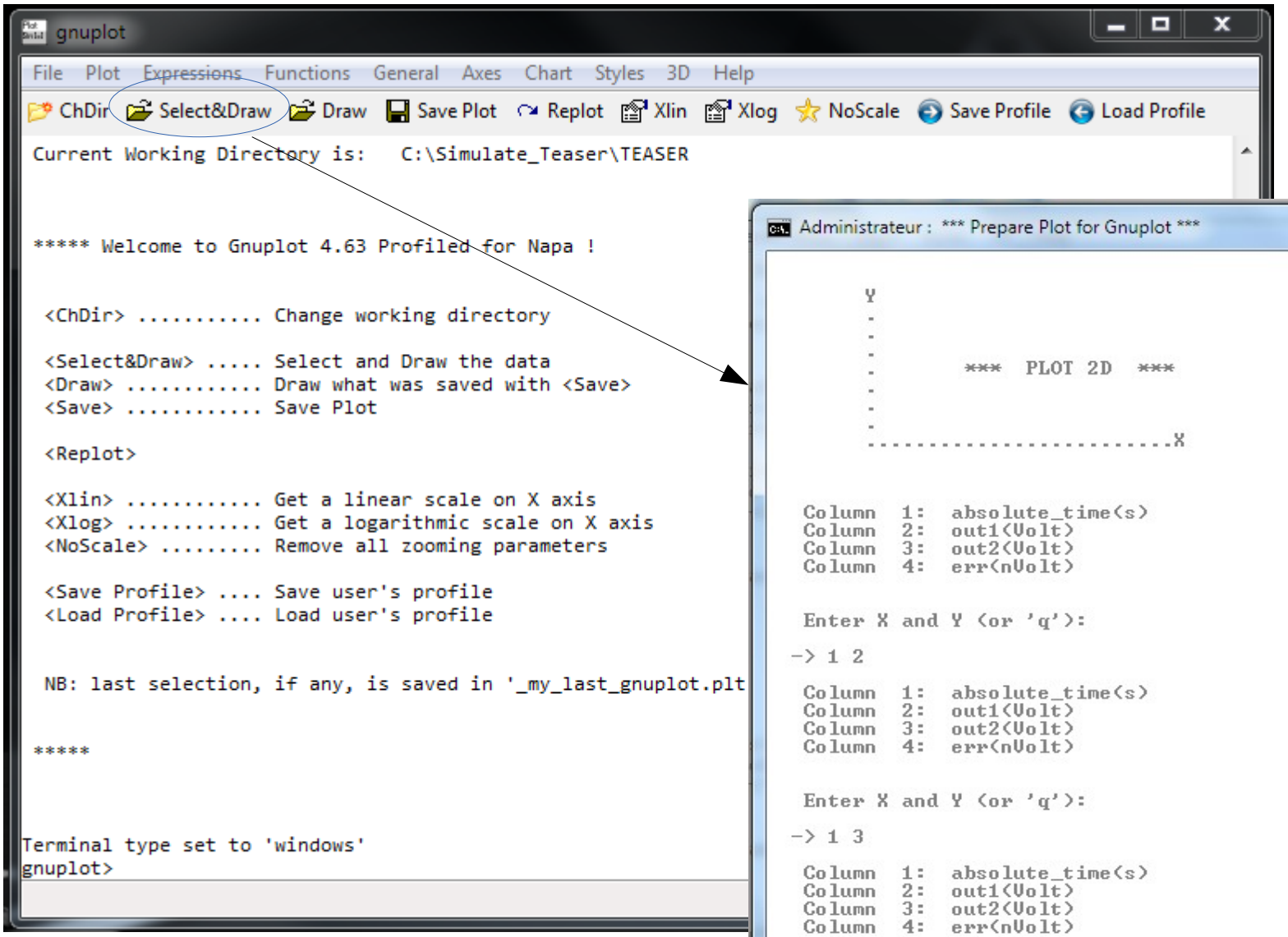
...
```

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

LOOP

4. VARIABLE UPDATE
5. NODE UPDATE
6. TIME DOMAIN OUTPUT

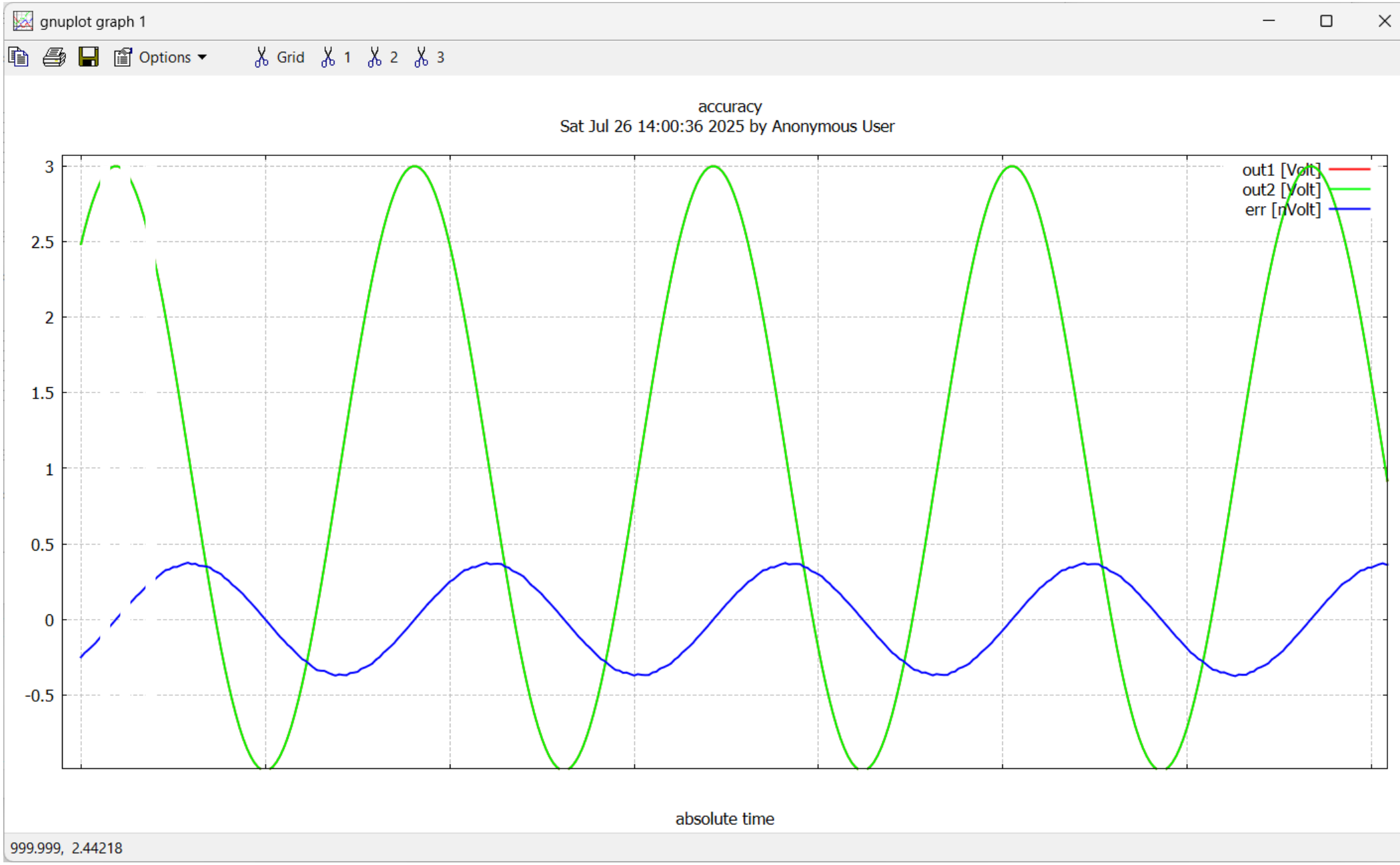

NAPA Compile and Run: Sour
[teaser7] **** MAC Preprocessor Running ****
[teaser7] **** NAPA Transpiler Running ****
[teaser7] **** GCC Compiler Running ****
[teaser7] **** Ad Hoc Simulator Running ****
NAPA Ping Information : function 'rand_uniform()' from file "C:/Simulate/NapaDos/Hdr/Function/random.hdr"
**** accuracy ****
**** Working Directory: "C:\Simulate_Examples" ****
**** Random Seed [I] : 876760345 ****
**** Output Tag [O] : 581589160 ****
**** NAPA Transpiler : V4.50 for Win64 ****
**** Main Netlist : teaser7.tmp ****
**** Simulator Time : 1.00000 ks ****
**** Simulator Index : 1 000 000 001 ****
**** Run Time I/O : ****
-> teaser7.log [0] ****
-> stdout [0] ****
**** Stopwatch : H00:M01:S15.427 ****
**** LOG File Ready : teaser7.log ****
**** Normal Termination ****
[teaser7]
Press Enter t
(1 billion sampling periods computed in 1.15 minute)
C:\Simulate_Examples\teaser7.out - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
teaser3.nap teaser1.nap teaser5.nap teaser7.c flow.c teaser7.nap
accuracy
(time domain output) Sampled at 1.00000 MHz
(transpiler version) NAPA V4.50 for Win64
(source file) "teaser7.tmp"
(random seed) 876760345
(number of columns) 4
Sat Jul 26 11:04:51 2025 by Anonymous User
absolute_time out1(Volt) out2 err(nVolt)
9.999990000000000e+02 2.328660325091e+00 2.328660325360e+00 -2.697051471046e-01
9.999990010000000e+02 2.440506718200e+00 2.440506718444e+00 -2.434550339103e-01
9.999990020000000e+02 2.543689734568e+00 2.543689734794e+00 -2.260969189649e-01
9.999990030000000e+02 2.637588819403e+00 2.637588819610e+00 -2.068221149898e-01
9.999990040000000e+02 2.721639252572e+00 2.721639252758e+00 -1.857110021319e-01
9.999990050000000e+02 2.795335544888e+00 2.795335545045e+00 -1.564495200057e-01
9.999990060000000e+02 2.858234478187e+00 2.858234478320e+00 -1.330051624393e-01
9.999990070000000e+02 2.909957770889e+00 2.909957770997e+00 -1.080890932315e-01
9.999990080000000e+02 2.950194353032e+00 2.950194353111e+00 -7.860601058951e-02
9.999990090000000e+02 2.978702237081e+00 2.978702237133e+00 -5.226352683962e-02
9.999990100000000e+02 2.995309973269e+00 2.995309973293e+00 -2.389644038203e-02
length: 84478 lines: 1019 Ln: 7 Col: 1 Pos: 215 Windows (CR LF) UTF-8 INS
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' less than 0.3 nano Volt after 1 billion simulation steps



We are now ready to use the NAPA smart tools !

C:\Simulate_Examples\teaser8.nap - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

teaser1.nap | teaser5.nap | teaser7.nap | teaser7.out | teaser8.nap

```
1
2   title "#n FFT"
3   header <napatool.hdr>
4
5   fs    1.0e6
6
7   dvar  freq    12345.6
8   dvar  ph      rand_uniform( 0.0, _2pi_ )
9
10  node  out     osc  1.0 2.0  freq ph
11
12  ivar  n       4
13  ivar  npts1   POWEROF2(20)
14  ivar  npts2   100000000
15
16  dvar  bw      100.0e6
17
18  tool  fft     "ffts.out"  out 1.0  bw  npts1
19  tool  synchro                        npts2
20
21  directive     WINDOW  BLACKMAN_HARRIS_7
22
23  terminate     n <= TOOL_INDEX
24
25  ping
26
```

U: length: 456 lines: 26 Ln: 1 Col: 1 Pos: 1

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^{20} samples, made every 10^8 samples, processed in 14 seconds

How Smart Tool Synchronization Works



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

A simple **state machine** is implemented in tools with 4 states:
'wait', 'countdown', 'accumulate', 'compute'.

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

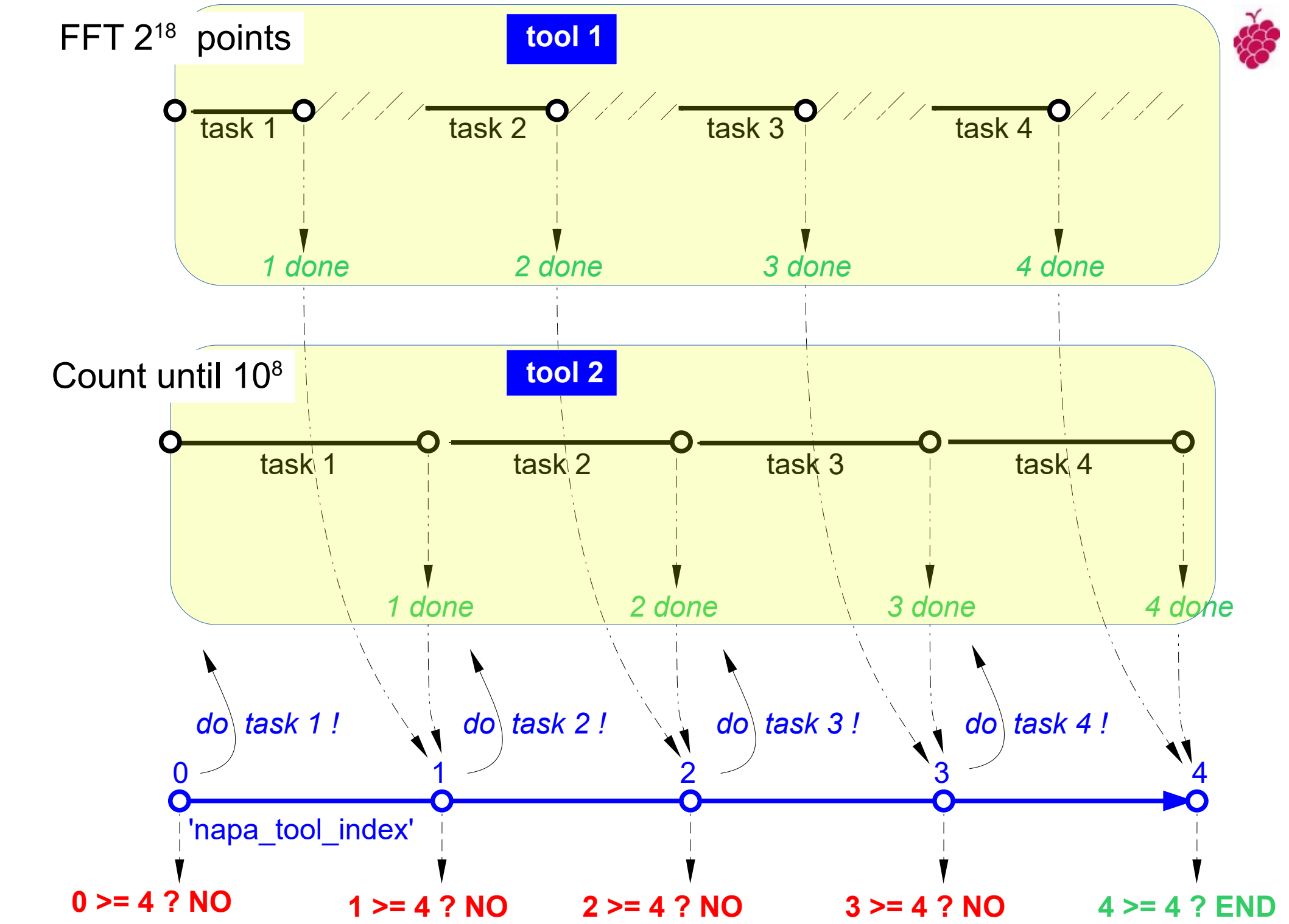
All tools start their own task. The output of the tool is the status of its work.
The simulator collects these statuses at the end of each simulation cycle.

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
task 2
task 3
task 4
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

```
...
do {
 napa_rel_time = napa_rel_loop * 1.000000000000000e-06L;
 napa_abs_time = napa_abs_loop * 1.000000000000000e-06L;

 hnode_out_osc2 = hnode_out_osc1;
 hnode_out_osc1 = hnode_out_osc0;
 hnode_out_osc0 = (hnode_out_fctr * hnode_out_osc1) - hnode_out_osc2;
 dnode_out = ((R_TYPE) hnode_out_osc1) * 2.0;
 dnode_out += 1.0;

 napa_msg = &(napa_mailbox[0]);
 napa_msg->o = napa_packet;
 inode__void0 = itool_fft_05("ffts.out",dnode_out,1.0,dvar_bw,ivar_npts1, 0);

 napa_msg = &(napa_mailbox[1]);
 napa_msg->o = napa_packet;
 inode__void1 = itool_synchro_01(ivar_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_abs_loop++;
 napa_rel_loop++;

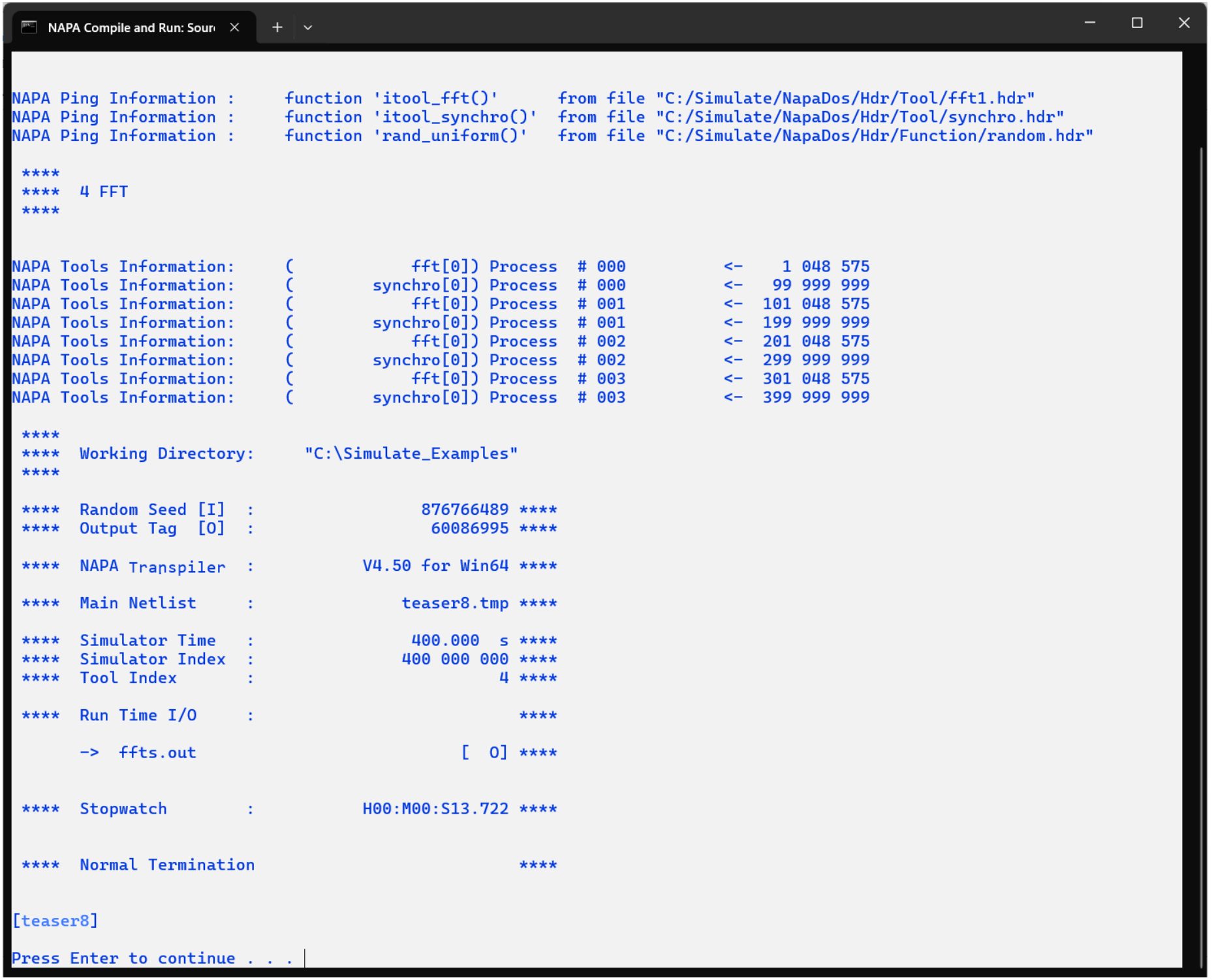
} while (!TERMINATE);
...
```

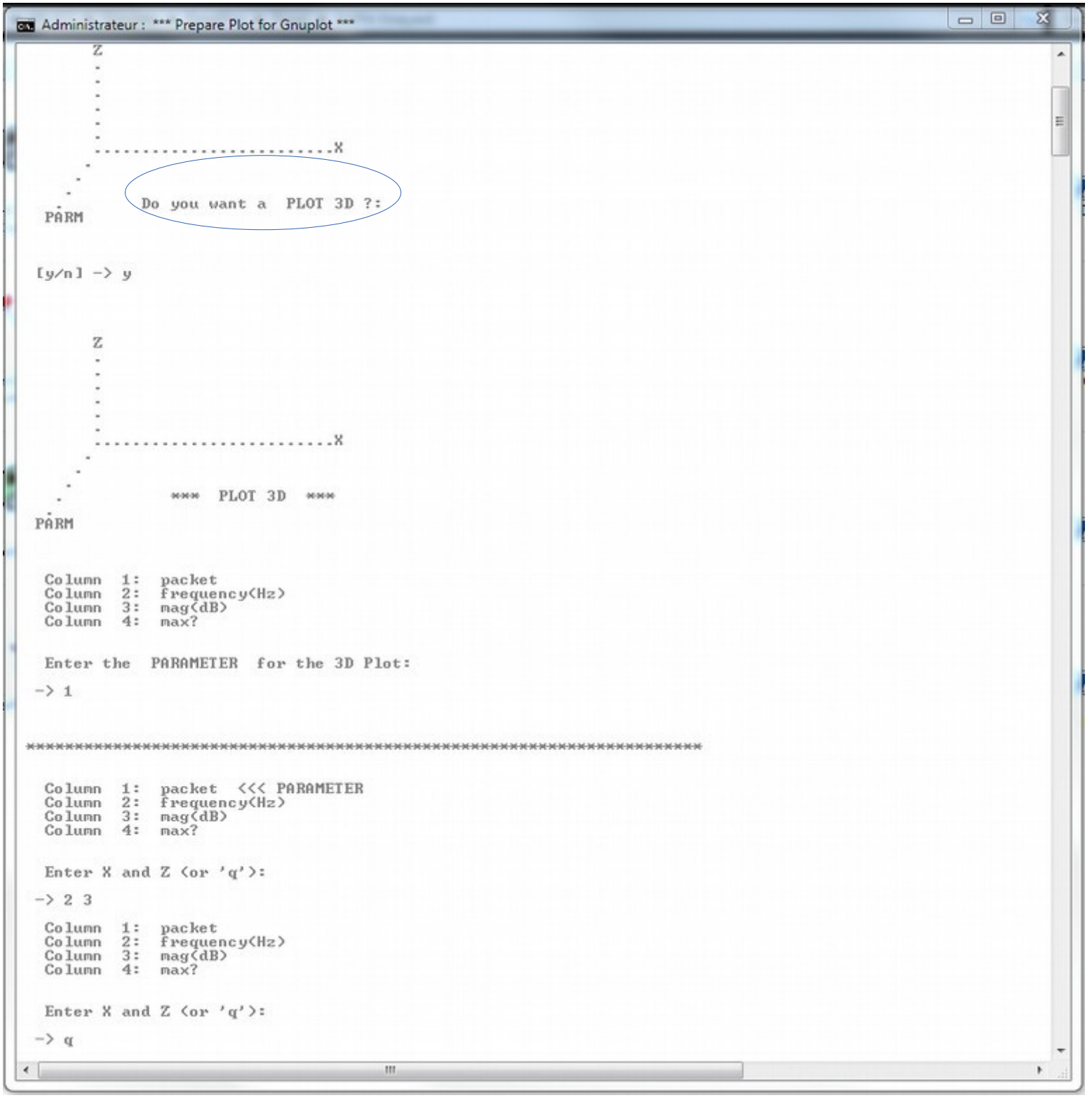
The simulator prefills the individual mailbox output

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

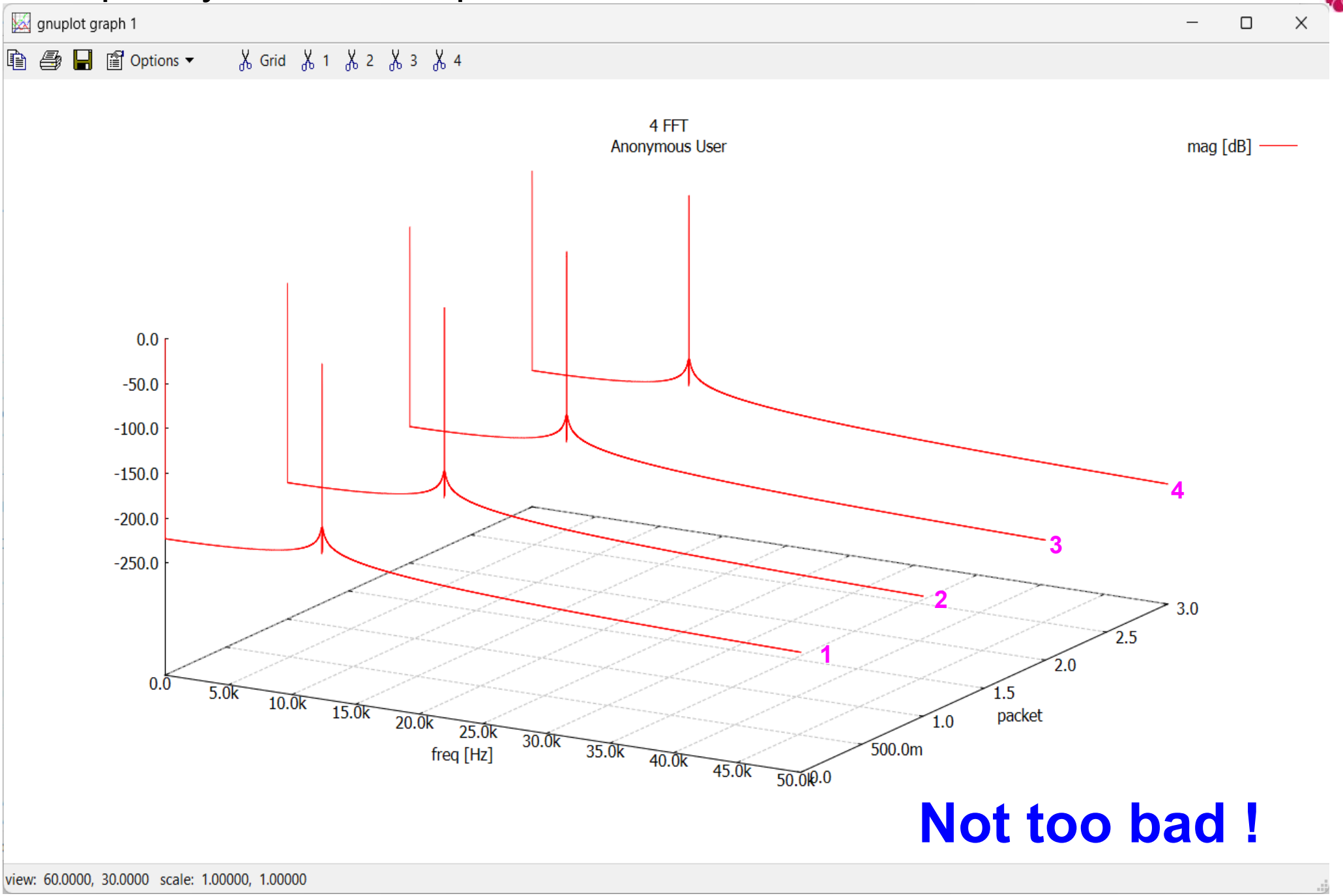
The macro 'TERMINATE' checks 'napa_tool_index'


NAPA Compile and Run: Sour
NAPA Ping Information : function 'itool_fft()' from file "C:/Simulate/NapaDos/Hdr/Tool/fft1.hdr"
NAPA Ping Information : function 'itool_synchro()' from file "C:/Simulate/NapaDos/Hdr/Tool/synchro.hdr"
NAPA Ping Information : function 'rand_uniform()' from file "C:/Simulate/NapaDos/Hdr/Function/random.hdr"
**** 4 FFT ****
NAPA Tools Information: (fft[0]) Process # 000 <- 1 048 575
NAPA Tools Information: (synchro[0]) Process # 000 <- 99 999 999
NAPA Tools Information: (fft[0]) Process # 001 <- 101 048 575
NAPA Tools Information: (synchro[0]) Process # 001 <- 199 999 999
NAPA Tools Information: (fft[0]) Process # 002 <- 201 048 575
NAPA Tools Information: (synchro[0]) Process # 002 <- 299 999 999
NAPA Tools Information: (fft[0]) Process # 003 <- 301 048 575
NAPA Tools Information: (synchro[0]) Process # 003 <- 399 999 999
**** Working Directory: "C:\Simulate_Examples" ****
**** Random Seed [I] : 876766489 ****
**** Output Tag [O] : 60086995 ****
**** NAPA Transpiler : V4.50 for Win64 ****
**** Main Netlist : teaser8.tmp ****
**** Simulator Time : 400.000 s ****
**** Simulator Index : 400 000 000 ****
**** Tool Index : 4 ****
**** Run Time I/O : ****
-> ffts.out [0] ****
**** Stopwatch : H00:M00:S13.722 ****
**** Normal Termination ****
[teaser8]
Press Enter to continue . . .


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

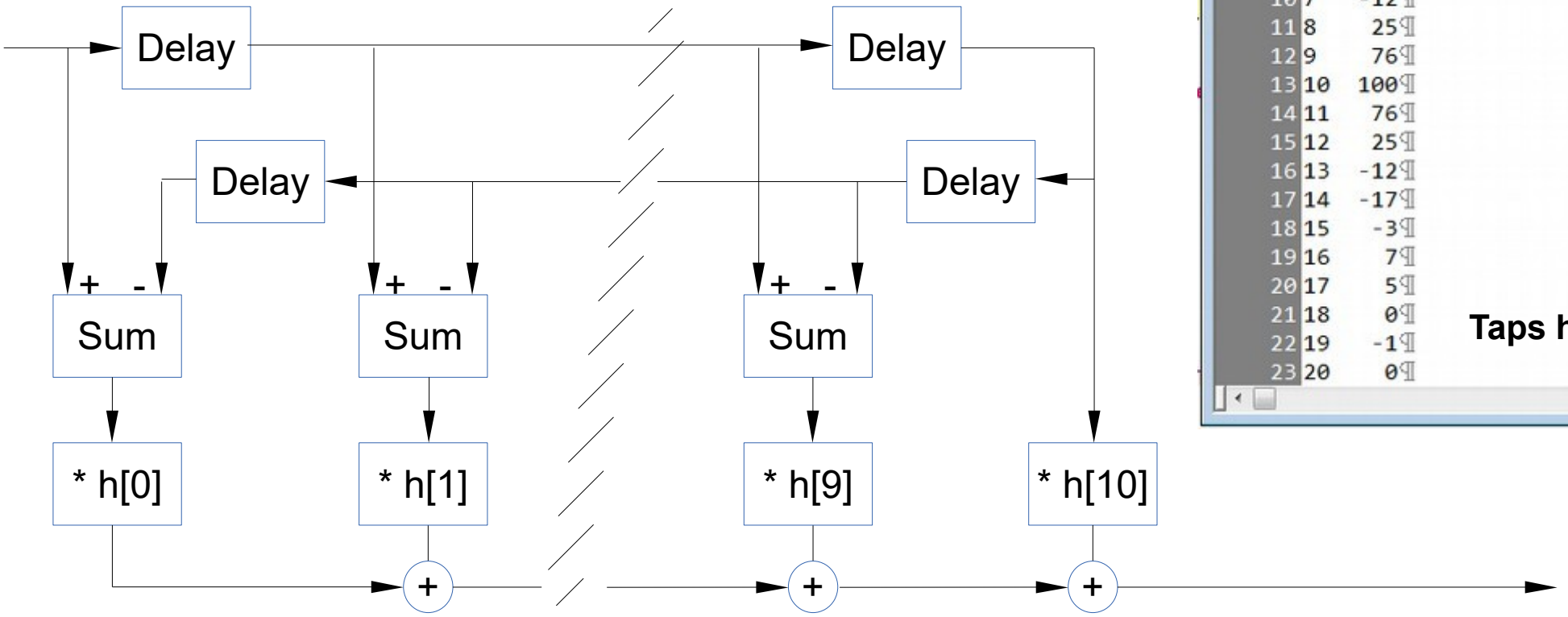
3D Plot

Frequency Domain Output, 3D Plot



Not too bad !

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

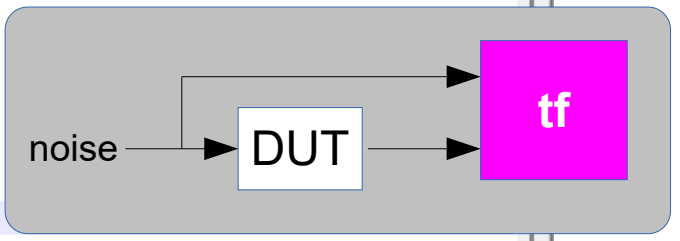
C:\Simulate_Examples\teaser10.nap - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

teaser10.nap

```
1
2   title "Transfer function of a symmetrical FIR"
3
4   header <napatool.hdr>
5
6   fs              1.0e6
7
8   directive       REPEAT      100
9
10  ivar   npts     POWEROF2(16)
11
12  node   in       cell        rclk   <Noise/rclock.net>  0.5
13  node   out      generator   sf11   <fir>   "~/fir5.tap" in
14
15  tool   tf       "tf.out"    in 1.0 out 1.0   npts
16
17  terminate       1.0 <= TOOL_INDEX
18
19  ping
20
```

length: 361 lines: 20 Ln: 10 Col: 26 Pos: 157 Win



Digital FIR
Cell generator

Transfer function

RMS of FFTs

On the fly cell generation
using a file containing the taps

1.0 second

NAPA Compile and Run: Sour

```
[teaser10]  **** GCC Compiler     Running ****
[teaser10]  **** Ad Hoc Simulator Running ****


NAPA Ping Information :     function 'itool_tf()'        from file "C:/Simulate/NapaDos/Hdr/Tool/fft3.hdr"
NAPA Ping Information :     function 'rand_bernoulli()'  from file "C:/Simulate/NapaDos/Hdr/Function/random.hdr"

 ****
 ****  Transfer function of a symmetrical FIR
 ****


NAPA Tools Information:     (            tf[0]) Process  # 000.009      <-      262 143
NAPA Tools Information:     (            tf[0]) Process  # 000.008      <-      524 287
NAPA Tools Information:     (            tf[0]) Process  # 000.007      <-      786 431
NAPA Tools Information:     (            tf[0]) Process  # 000.006      <-    1 048 575
NAPA Tools Information:     (            tf[0]) Process  # 000.005      <-    1 310 719
NAPA Tools Information:     (            tf[0]) Process  # 000.004      <-    1 572 863
NAPA Tools Information:     (            tf[0]) Process  # 000.003      <-    1 835 007
NAPA Tools Information:     (            tf[0]) Process  # 000.002      <-    2 097 151
NAPA Tools Information:     (            tf[0]) Process  # 000.001      <-    2 359 295
NAPA Tools Information:     (            tf[0]) Process  # 000          <-    2 621 439

 ****
 ****  Working Directory:     "C:\Simulate_Examples"
 ****

 ****  Random Seed [I]  :                876779979 ****
 ****  Output Tag  [O]  :                327310774 ****

 ****  NAPA Transpiler  :           V4.50 for Win64 ****

 ****  Main Netlist     :              teaser10.tmp ****

 ****  Simulator Time   :                2.62144  s ****
 ****  Simulator Index  :                2 621 440 ****
 ****  Tool Index       :                        1 ****

 ****  Run Time I/O     :                          ****

       ->  tf.out                            [  0] ****


 ****  Stopwatch                    H00:M00:S01.030 ****


 ****  Normal Termination                      ****
```

Taps

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¶
```

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

generator '**fir**'

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

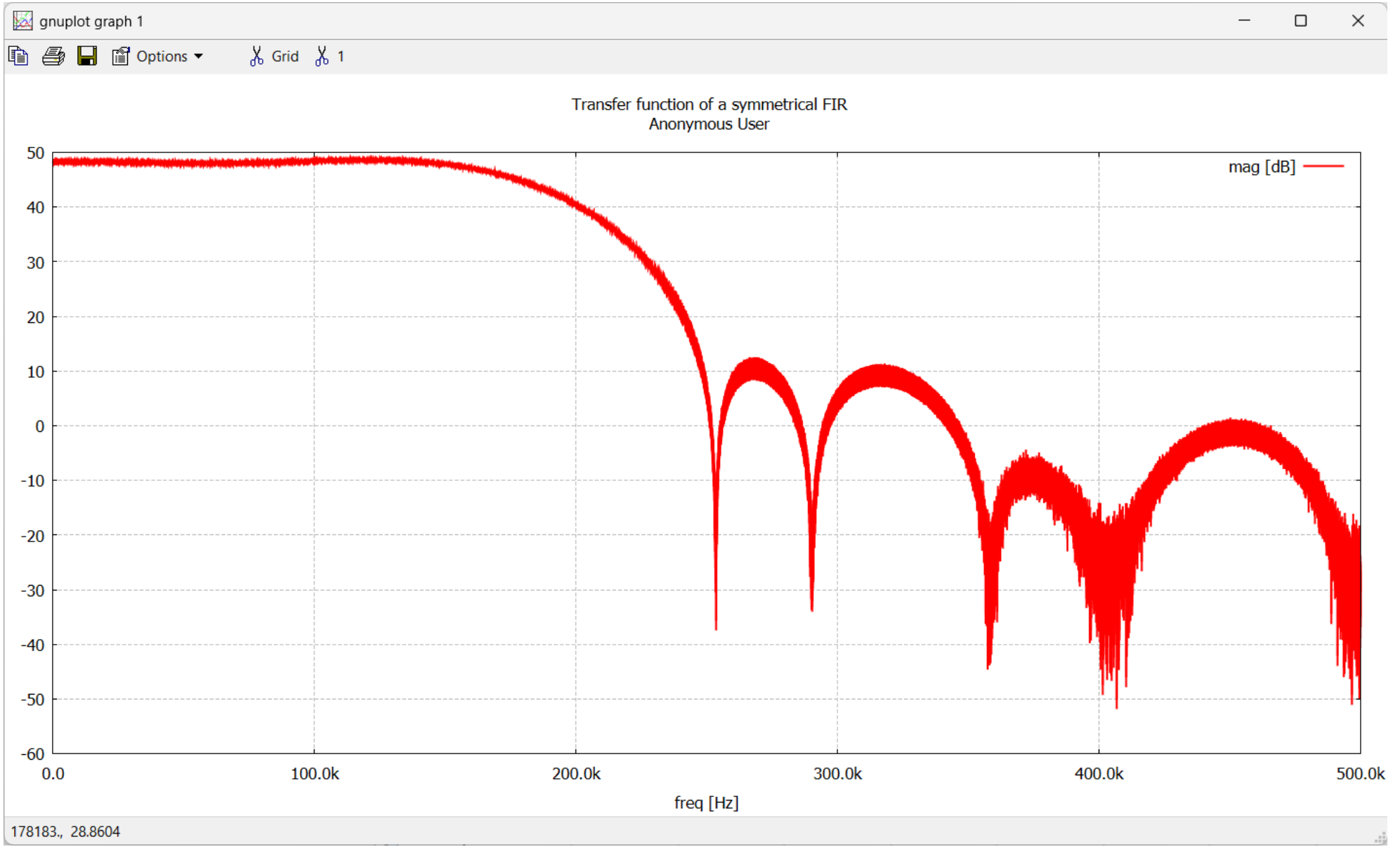
*C:\Simulate_Examples\sf11_0.gen - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

teaser10.nap | sf11_0.gen

```
1   cell_interface  $y  $dummy  $x
2
3   # Symmetrical Filter
4   # Taps File Name: './fir5.tap'
5   # Creation Time :  Sat Jul 26 21:59:19 2025
6   # By Command    :  C:\Simulate\NapaDos\Gen\fir
7   # Generator Version 3.17
8   # From C source : 'C:\Simulate\NapaDos\Gen\Source\fir.c'
9
10  # FOLDED FIR
11  #
12  #          ______                                  ______
13  #         |      |    d1                 d9     |      |    d10
14  # x -+----| Delay >----+-------- .. ----+----| Delay >----+
15  #    |    |______|     |                |    |______|     |
16  #    |                 |                |                 |
17  #    |  d20  |      |  |  d19           |  d11  |      |  |
18  #    |   +--< Delay |--+------+---- .. --------+--< Delay |--+
19  #    |   |   |______|  |      |         |      |   |______|  |
20  #   _|___|_           _|___|_          _|___|_              |
21  #  |(+)  (+)|         |(+)  (+)|       |(+)  (+)|           |
22  #  |        |         |        |       |        |           |
23  #  |  Sum   |         |  Sum   |       |  Sum   |           |
24  #  |________|         |________|       |________|           |
25  #      |                  |                |               |
26  #      | s0               | s1             | s9            |
27  #   ___|___            ___|___          ___|___         ___|___
28  #  |       |          |       |        |       |       |       |
29  #  |  *h0  |          |  *h1  |        |  *h9  |       |  *h10 |
30  #  |_______|          |_______|        |_______|       |_______|
31  #      |                  |                |               |
32  #      | g0               | g1             | g9            | g10
33  #      |                  |                |               |
34  #      +------------------(+)------ .. -----(+)-------------(+)--> y
35
36  ivar $h0      0
37  ivar $h1     -1
38  ivar $h2      0
39  ivar $h3      5
40  ivar $h4      7
41  ivar $h5     -3
42  ivar $h6     -17
43  ivar $h7     -12
44  ivar $h8      25
45  ivar $h9      76
46  ivar $h10     100
47
48  node $d1      delay $x
```

length: 3798 lines: 103 | Ln: 10 Col: 6 Pos: 273 | Windows (CR LF) | UTF-8 | INS


gnuplot graph 1
Options
Grid
1
Transfer function of a symmetrical FIR
Anonymous User
mag [dB]
50
40
30
20
10
0
-10
-20
-30
-40
-50
-60
0.0
100.0k
200.0k
300.0k
400.0k
500.0k
freq [Hz]
178183., 28.8604

Step and Impulse Response ?

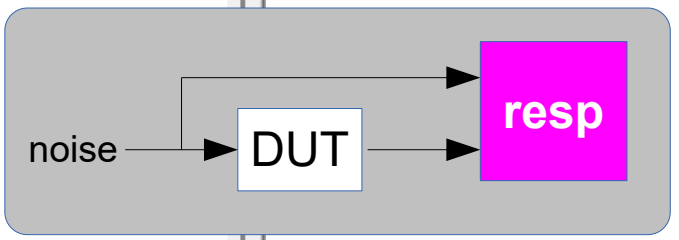
C:\Simulate_Examples\teaser11.nap - Notepad++

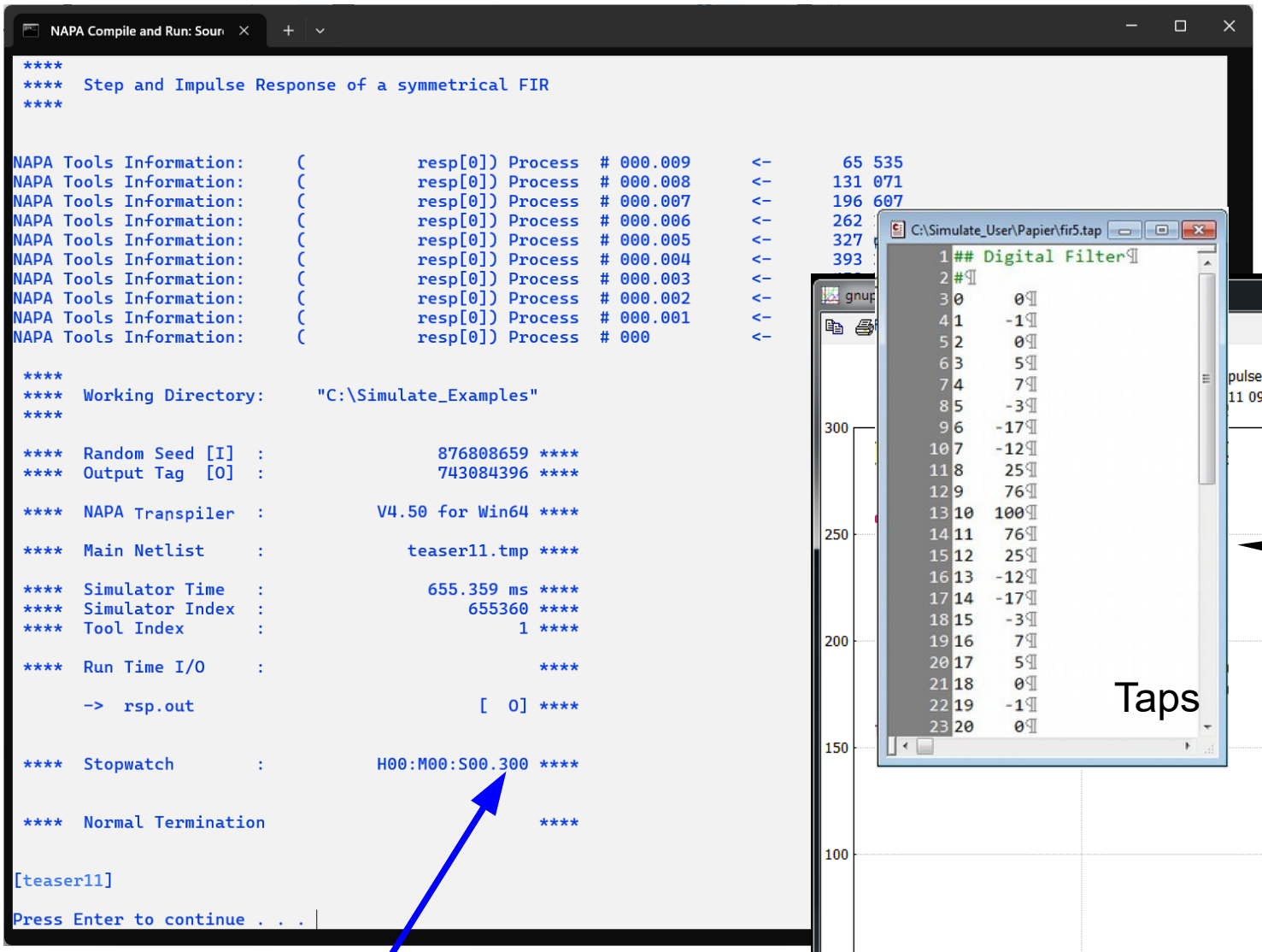
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

teaser11.nap

```
1
2   title "Step and Impulse Response 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>  0.5
13  node   out      generator sf11  <fir>  "~/fir5.tap" in
14
15  tool   resp     "rsp.out"  in 1.0 out 1.0  npts
16
17  terminate       1.0 <= TOOL_INDEX
18
19  ping
20
```

length: 369 lines: 20 Ln: 18 Col: 1 Pos: 362 Windows (CR LF) UTF-8 INS



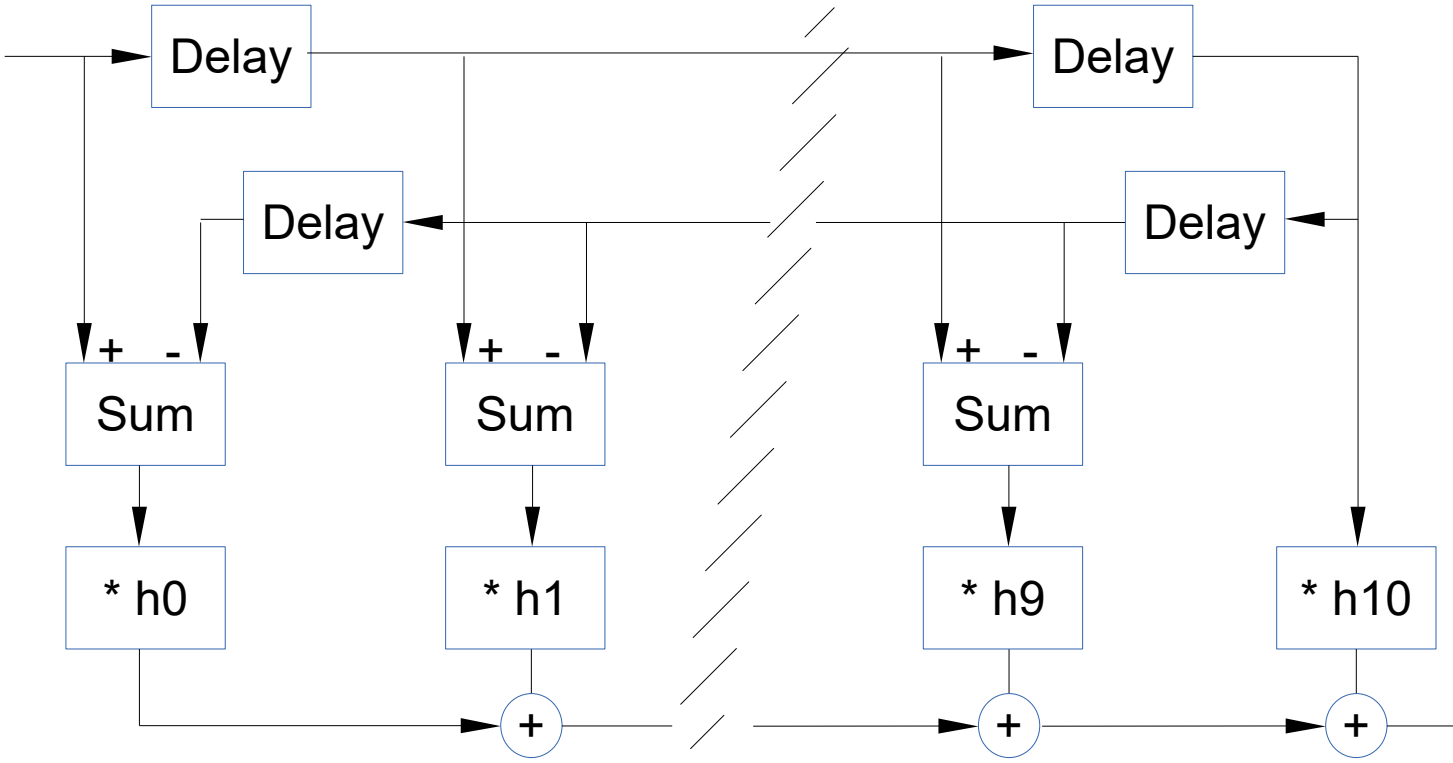

NAPA Compile and Run: Sour
****
**** Step and Impulse Response of a symmetrical FIR
****
NAPA Tools Information: (resp[0]) Process # 000.009 <- 65 535
NAPA Tools Information: (resp[0]) Process # 000.008 <- 131 071
NAPA Tools Information: (resp[0]) Process # 000.007 <- 196 607
NAPA Tools Information: (resp[0]) Process # 000.006 <- 262
NAPA Tools Information: (resp[0]) Process # 000.005 <- 327
NAPA Tools Information: (resp[0]) Process # 000.004 <- 393
NAPA Tools Information: (resp[0]) Process # 000.003 <-
NAPA Tools Information: (resp[0]) Process # 000.002 <-
NAPA Tools Information: (resp[0]) Process # 000.001 <-
NAPA Tools Information: (resp[0]) Process # 000 <-
****
**** Working Directory: "C:\Simulate_Examples"
****
**** Random Seed [I] : 876808659 ****
**** Output Tag [O] : 743084396 ****
**** NAPA Transpiler : V4.50 for Win64 ****
**** Main Netlist : teaser11.tmp ****
**** Simulator Time : 655.359 ms ****
**** Simulator Index : 655360 ****
**** Tool Index : 1 ****
**** Run Time I/O : ****
-> rsp.out [0] ****
**** Stopwatch : H00:M00:S00.300 ****
**** Normal Termination ****
[teaser11]
Press Enter to continue . . .
0.3 second

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

pulse Response of a Symmetrical FIR
11 09:28:42 2013 by Yves Leduc
impulse resp
step resp
Perfect match
300
250
200
150
100
50
0
-50
0.0
5.0u
10.0u
15.0u
20.0u
time(s)
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



Multirate simulation
Multirate transfer function

C:\Simulate_Examples\teaser12.nap - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

teaser12.nap

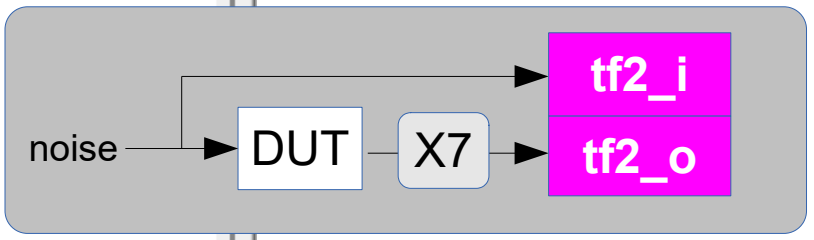
```
1
2   title "Multirate transfer function of a symmetrical FIR"
3
4   header          <napatool.hdr>
5
6   fs              1.0e6
7
8   directive       REPEAT      10
9
10  ivar   npts     POWEROF2(16)
11  node   in       cell        rclk   <Noise/rclock.net>   0.5
12  node   out      generator   sf11   <fir>   "~/fir5.tap" in
13
14  interpolate     7
15
16  tool   tf2_i    "tf2.out"    in   1.0    npts    with in
17  tool   tf2_o    "tf2.out"    out  1.0    npts
18
19  terminate       1.0 <= TOOL_INDEX
20
21  ping
22  debug    SAMPLING
```

length: 459 lines: 22 | Ln: 16 Col: 10 Pos: 315 | Windows (CR LF) | UTF-8 | INS

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', and it is therefore computed at frequency 'fs'



@ 1 MHz
@ 7 MHz



```
NAPA Compile and Run: Sour
NAPA Transpiler Information:  (generator)
 Generating cell file <./sf11_0.gen>,
 through system call: 'C:\Simulate\NapaDos\Gen\fir ./sf11_0.gen ./fir5.tap in'

[teaser12]  **** GCC Compiler     Running ****
[teaser12]  **** Ad Hoc Simulator Running ****


NAPA Ping Information :     function 'itool_tf2_i()'     from file "C:/Simulate/NapaDos/Hdr/Tool/fft6.hdr"
NAPA Ping Information :     function 'itool_tf2_o()'     from file "C:/Simulate/NapaDos/Hdr/Tool/fft6.hdr"
NAPA Ping Information :     function 'rand_bernoulli()'  from file "C:/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     <-       65 535
NAPA Tools Information:     (          tf2[0,0]) Process  # 000.008     <-      131 071
NAPA Tools Information:     (          tf2[0,0]) Process  # 000.007     <-      196 607
NAPA Tools Information:     (          tf2[0,0]) Process  # 000.006     <-      262 143
NAPA Tools Information:     (          tf2[0,0]) Process  # 000.005     <-      327 679
NAPA Tools Information:     (          tf2[0,0]) Process  # 000.004     <-      393 215
NAPA Tools Information:     (          tf2[0,0]) Process  # 000.003     <-      458 751
NAPA Tools Information:     (          tf2[0,0]) Process  # 000.002     <-      524 287
NAPA Tools Information:     (          tf2[0,0]) Process  # 000.001     <-      589 823
NAPA Tools Information:     (          tf2[0,0]) Process  # 000         <-      655 359

 ****
 ****  Working Directory:     "C:\Simulate_Examples"
 ****

 ****  Random Seed [I]  :                876850753 ****
 ****  Output Tag  [O]  :                901017450 ****

 ****  NAPA Transpiler  :          V4.50 for Win64 ****

 ****  Main Netlist     :             teaser12.tmp ****

 ****  Simulator Time   :               655.360 ms ****
 ****  Simulator Index  :                  655 360 ****
 ****  Simulator Loops  :                4 587 520 ****
 ****  Tool Index       :                        1 ****
```

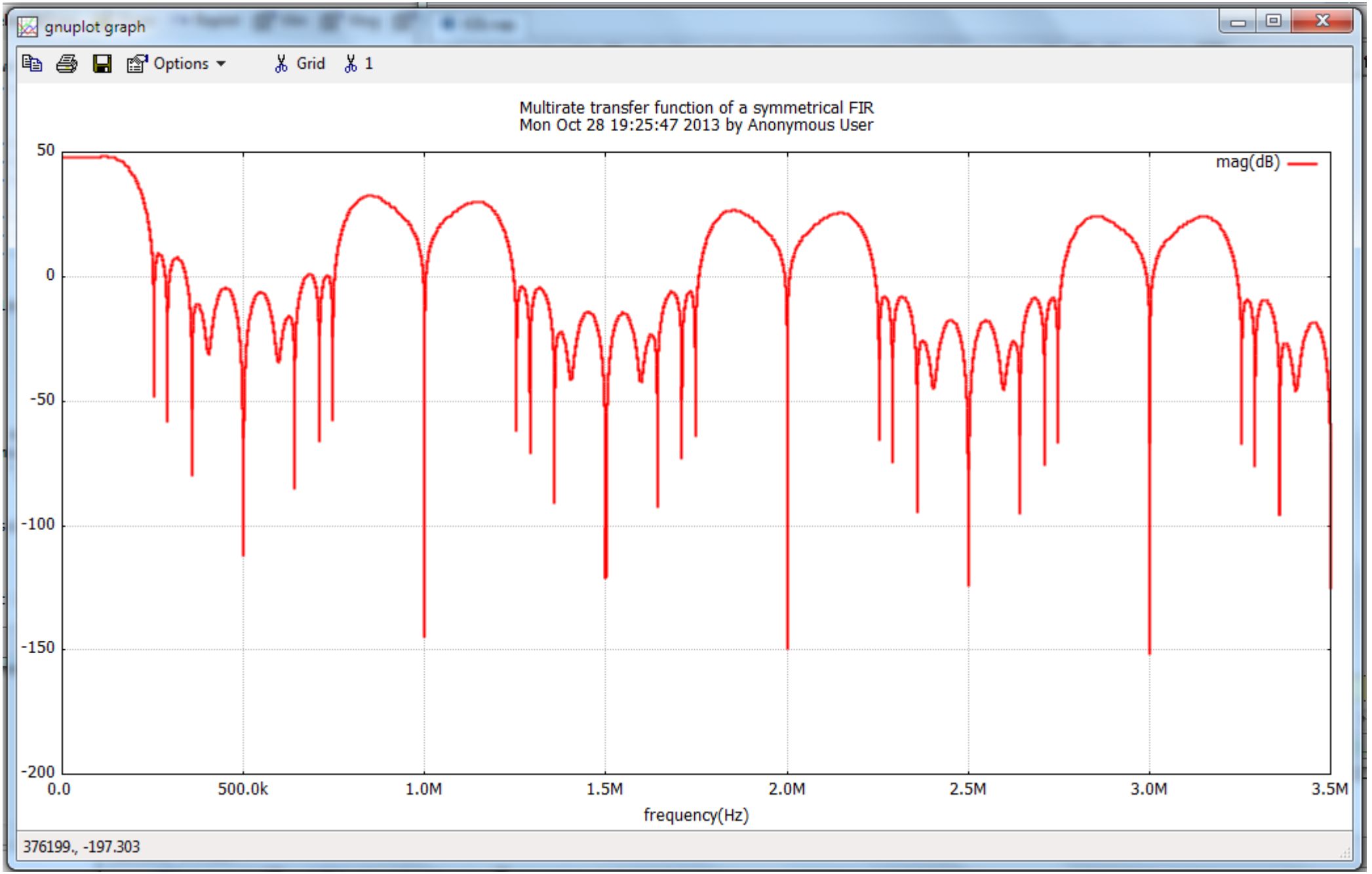
Automatic Cell Generation

Automatic management of the sampling

2x10 RMS of FFT's and 1 IFFT

4.6 millions loops in less than 0.7 second

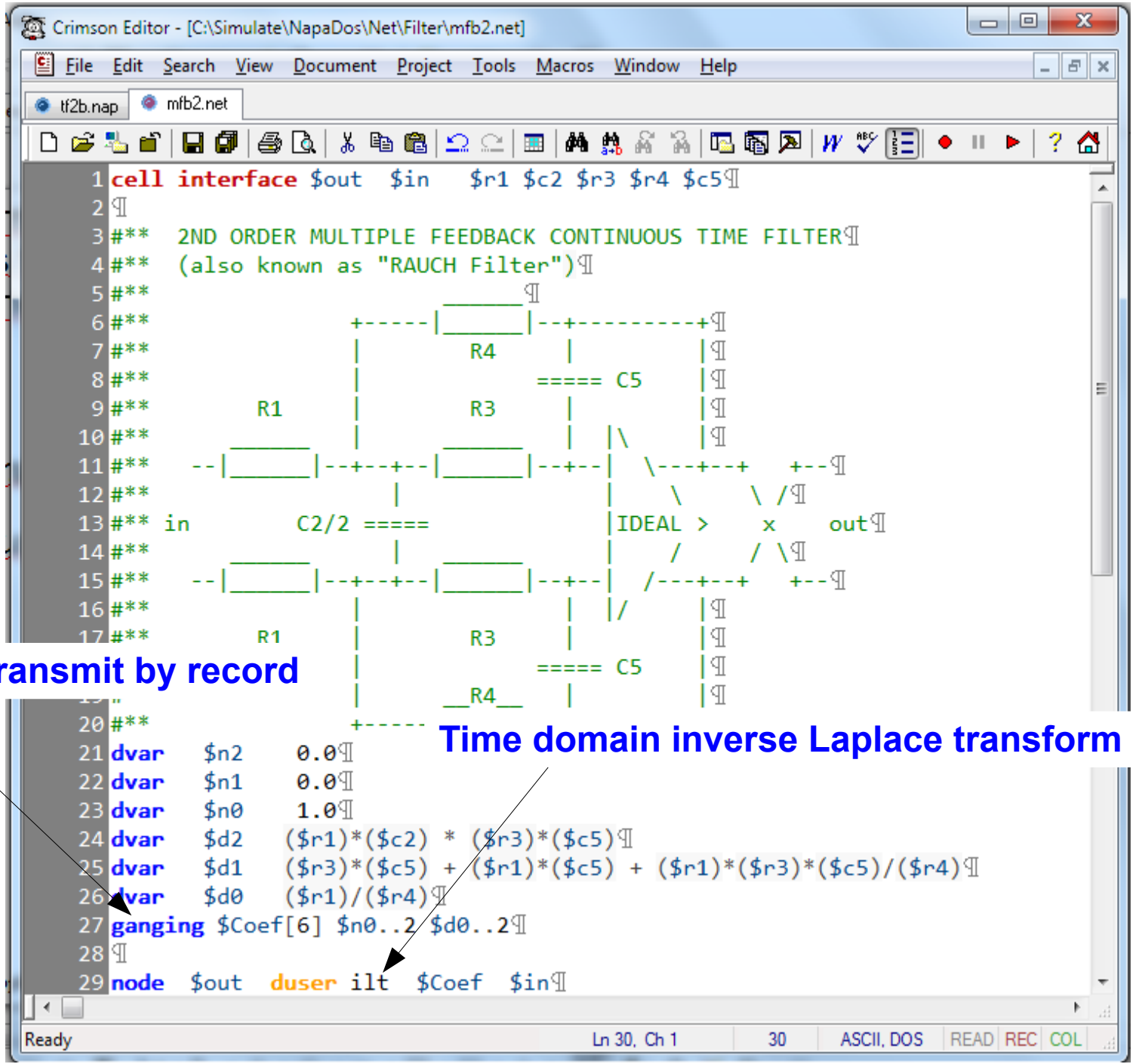
Multirate transfer function



There are many more capabilities than 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 Transpiler  :         V3.01b for Win64 ****
 ****  Main Netlist     :            ct_filter.tmp ****
 ****  Simulator Index  :                   655360 ****
 ****  Simulation Time  :              655.359 ms ****

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

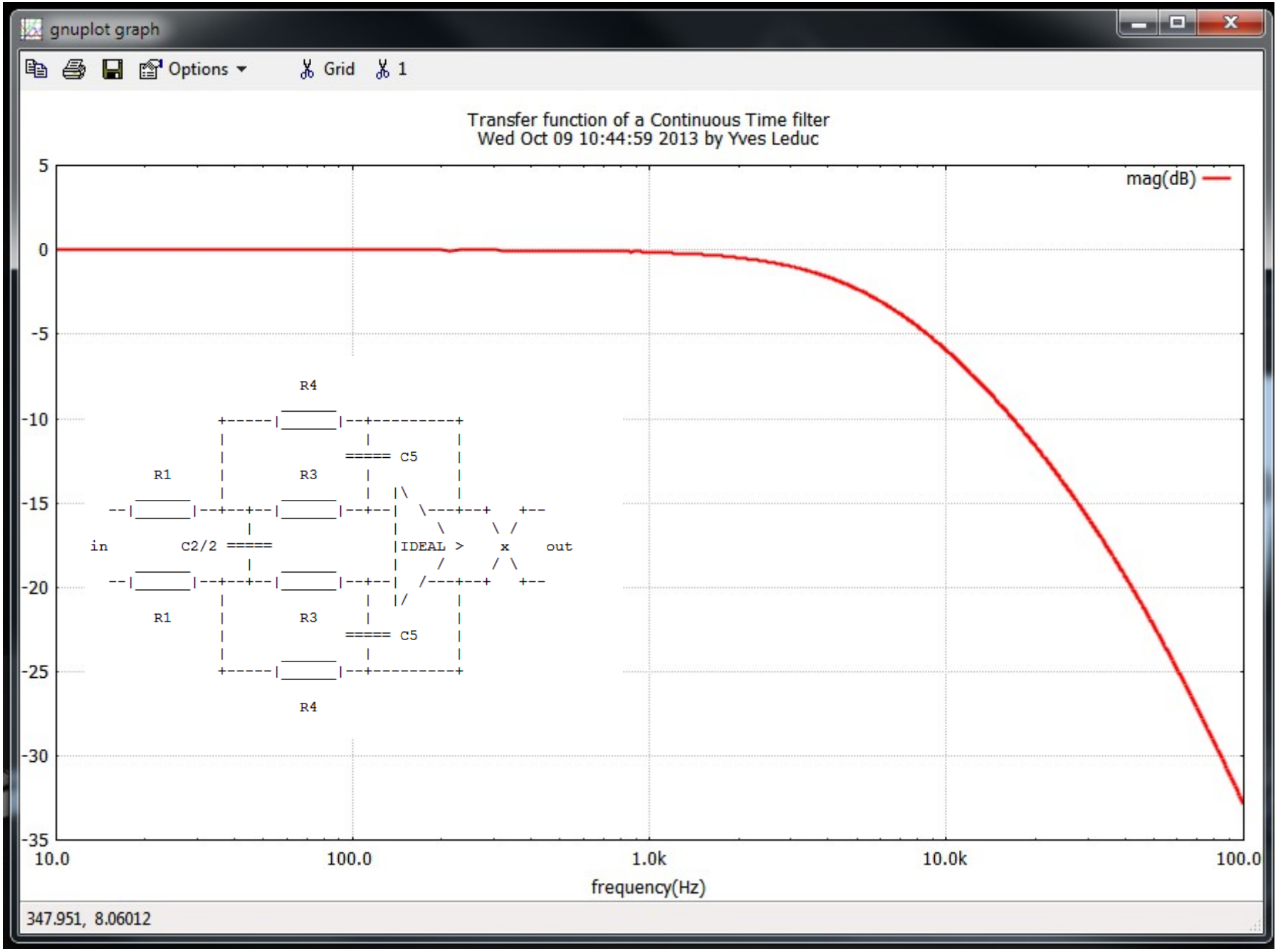

 ****  Stopwatch        :           H00:M00:S00.326 ****

[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 Transpiler  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

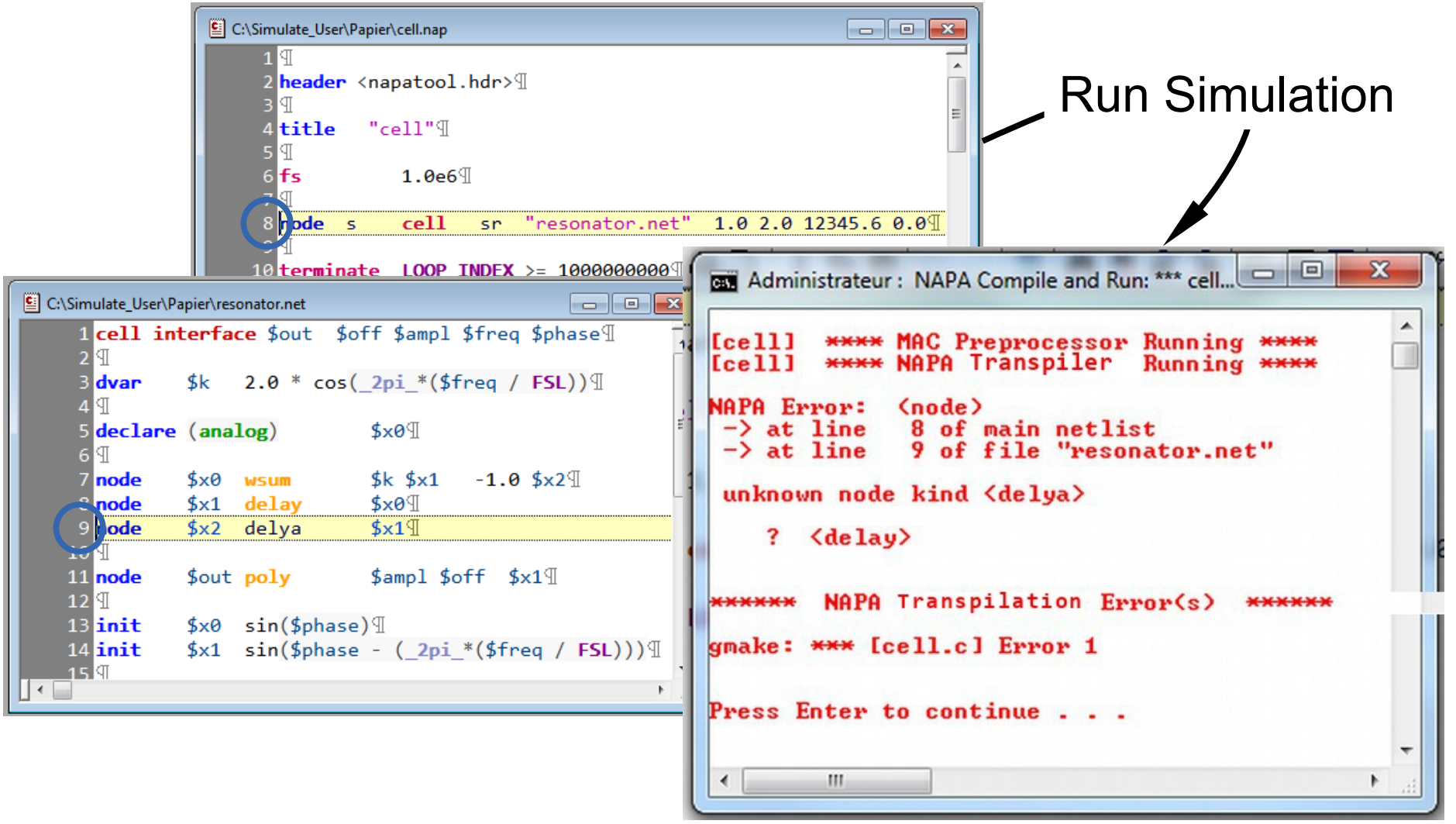

******  NAPA Transpilation Error(s)  ******

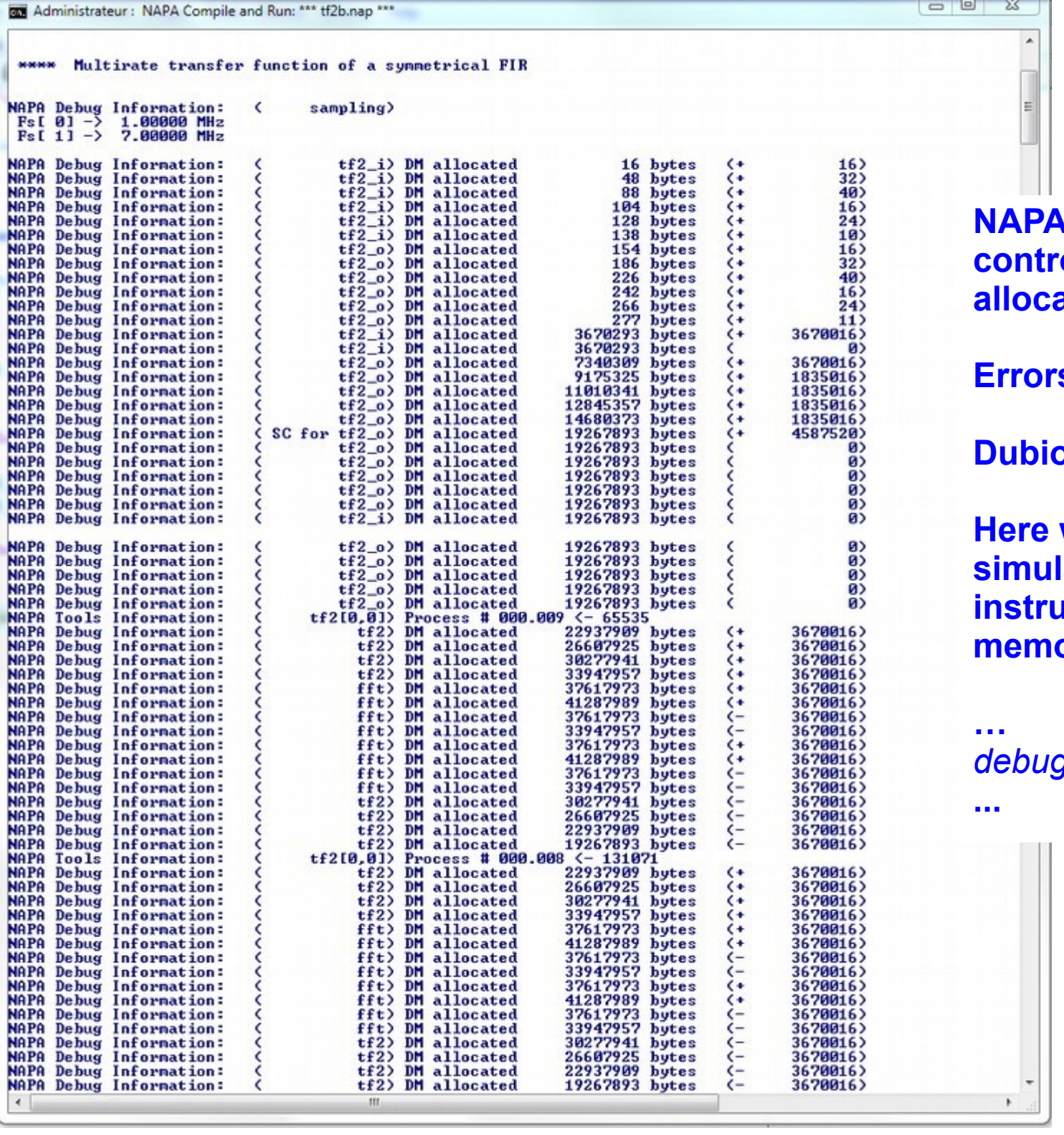
gmake: *** [geb.c] Error 1

Press Enter to continue . . .
```

Type Mismatches, Syntax Errors are Caught at Transpilation

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 caught on the fly.

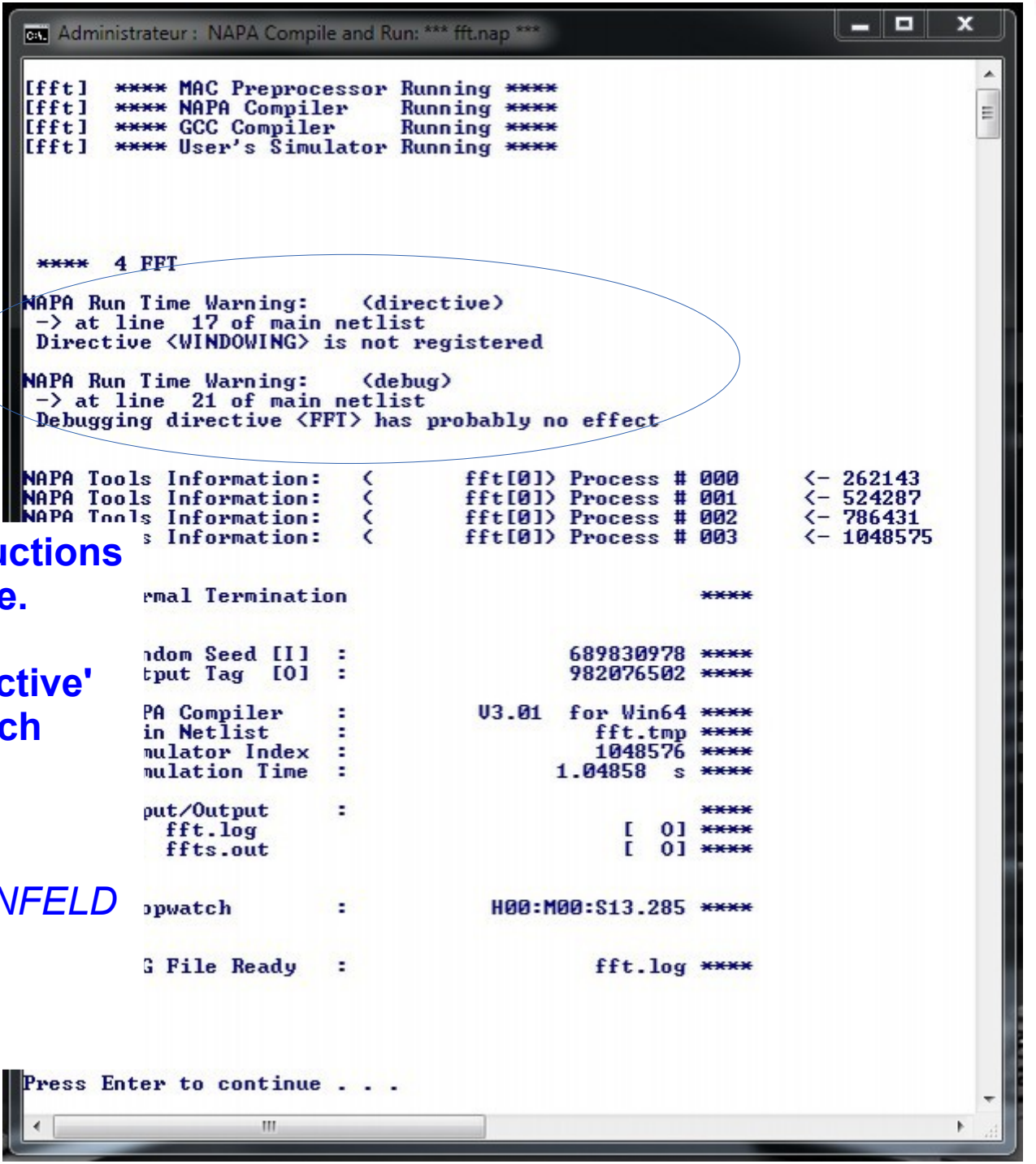
Dubious behavior is caught.

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 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

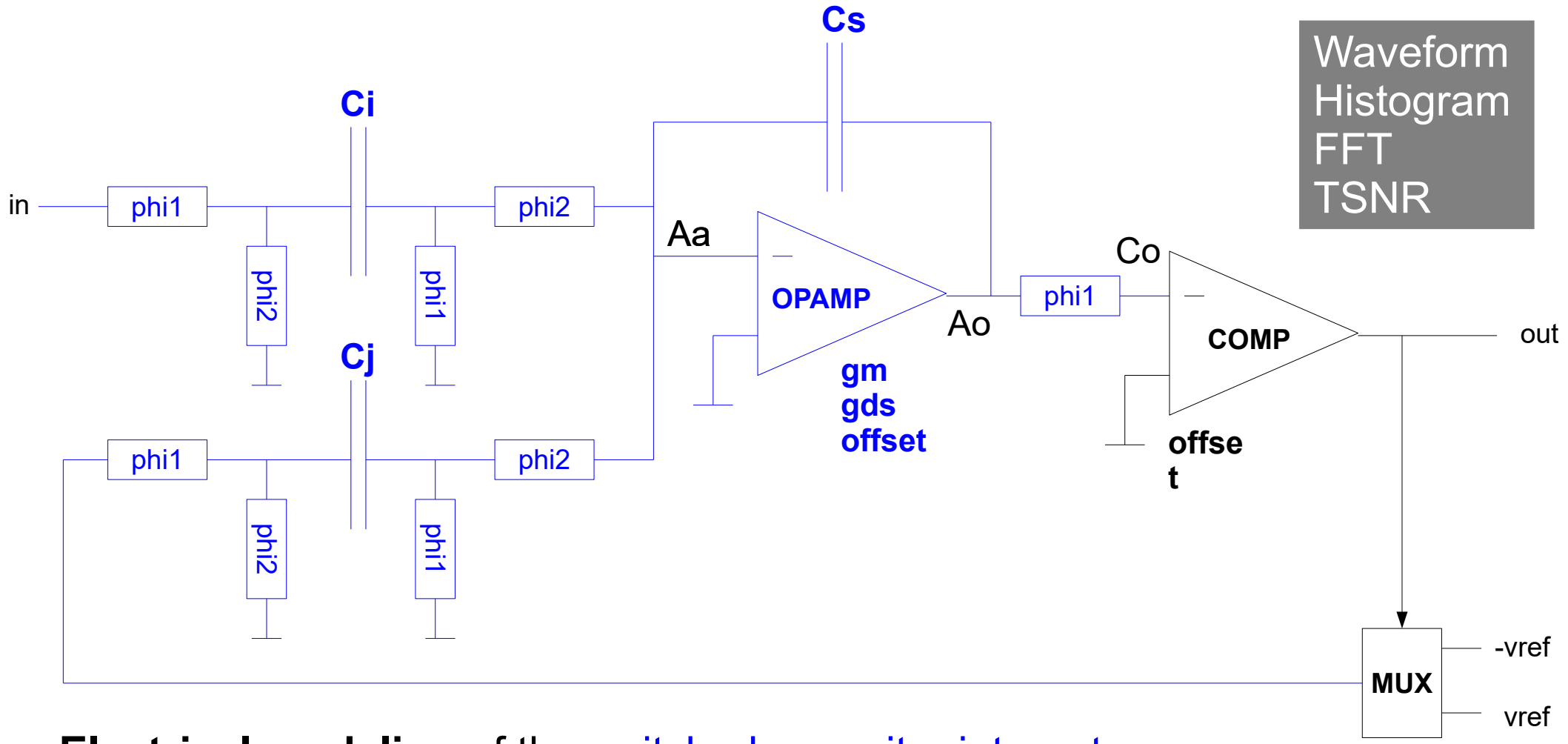
'Semi Analytical Recursive Convolution'

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-capacitor circuits at the switch level, class D amplifiers, DC-DC converters, etc.

This important solution is developed by Jacques Méquin and deserves 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 Transpiler Running ****
[time_analyziz]  **** GCC Compiler    Running ****
[time_analyziz]  **** SARC Engine     Linking ****
[time_analyziz]  **** Ad Hoc Simulator Running ****

MAPA Ping Information :   "duree_vars()"    from file "/Simulate/MapaDoz/Hdr/User/vars.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     <- zd1__VoffA
 ****  258.7 u     <- zd1__VoffC
 ****  500.0 f     <- zd1__Ck
 ****  11.54 k     <- zd1__Again
 ****  5.000 p     <- zd1__Acr
 ****  5.000 p     <- zd1__Acj
 ****  5.000 p     <- zd1__Aci

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


   OPAMP :   a --------o     o---------o------o----          COMP :   a ---------o--------|\____|
                       |     |         |      |                                  |        |    |
                  ca ===  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        zd1__Aca =   200.000 f   ->   Aca
[ SD1_core           ]     parameter #2        zd1__Aci =   5.00000 p   ->   Aci
[ SD1_core           ]     parameter #3        zd1__Acj =   5.00000 p   ->   Acj
[ SD1_core           ]     parameter #4        zd1__Aco =   400.000 f   ->   Aco
[ SD1_core           ]     parameter #5        zd1__Acr =   5.00000 p   ->   Acr
[ SD1_core           ]     parameter #6         zd1__Ck =   500.000 f   ->   Cload
[ SD1_core           ]     parameter #7       zd1__PHI1 =   0.00000     ->   PHI1
[ SD1_core           ]     parameter #8       zd1__PHI2 =   0.00000     ->   PHI2
[ SD1_core           ]     parameter #9       zd1__Ardz =   38.4615 M   ->   Ardz
[ SD1_core           ]     parameter #10       zd1__Agm =   299.997 u   ->   Agm

 ****  Normal Termination                          ****

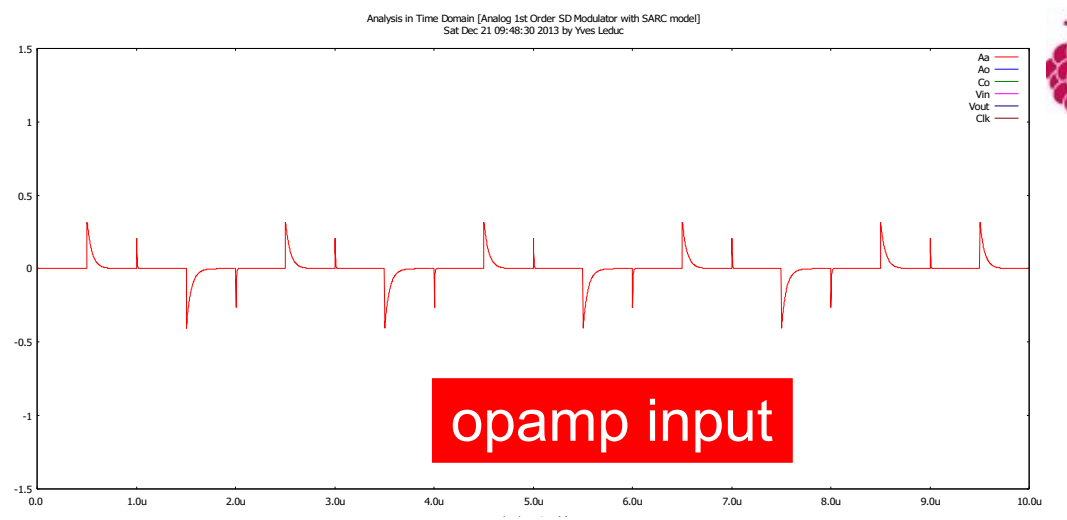
 ****  Random Seed [I]   :              693807855 ****
 ****  Output Tag  [O]   :              451714812 ****

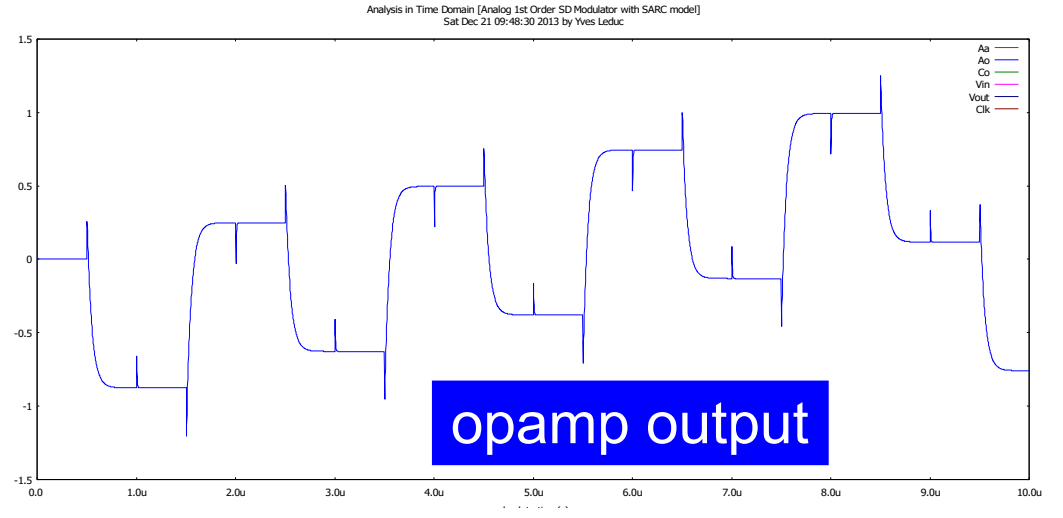
 ****  MAPA Transpiler   :          V3.02 for Win64 ****
 ****  Main Netlist      :       time_analyziz.tmp ****
 ****  Simulator Loops   :                    4000 ****
 ****  Simulator Index   :                      20 ****
 ****  Simulation Time   :            9.99750 us   ****

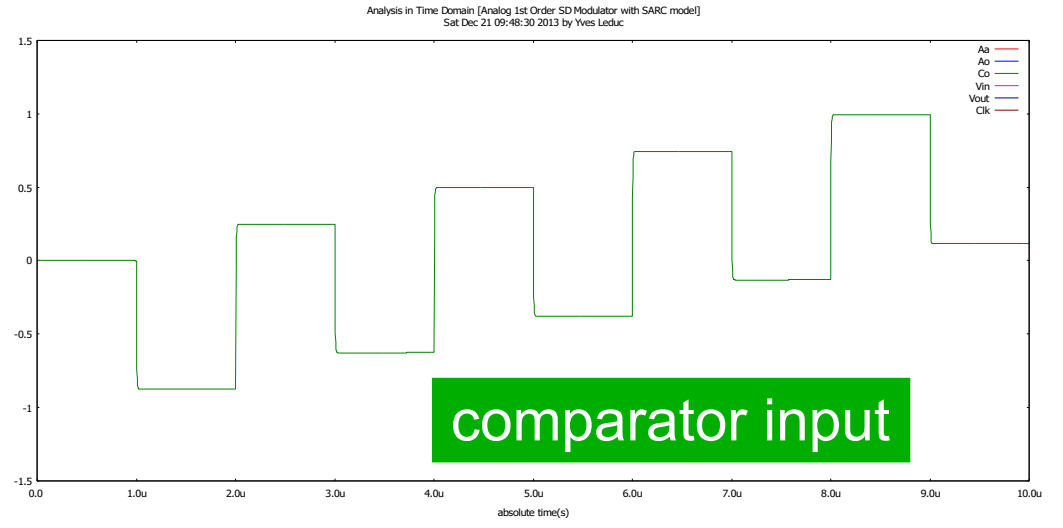
 ****  Input/Output      :                         ****
 ****  ->  time.out                  [  0] ****

 ****  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

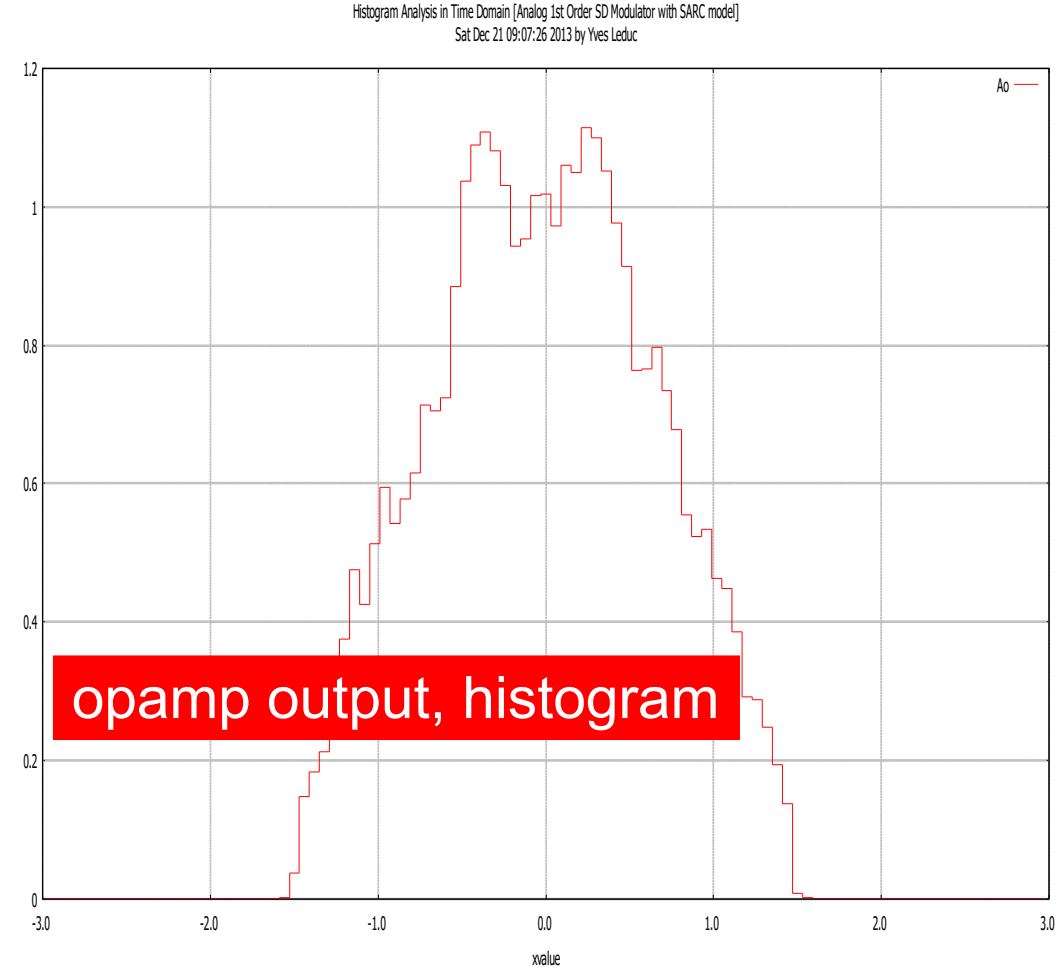
node     Vin       osc    0.0  ampl   freq   ph
node     Vrefa     dc    (analog) 1.0
node     Vout      cell   sd1  "./sd1.net"  Vin Vrefa  Clk  opfile1..2
tool     histoval  "histo.out"   Ao Vrefa   -3.0  3.0  100

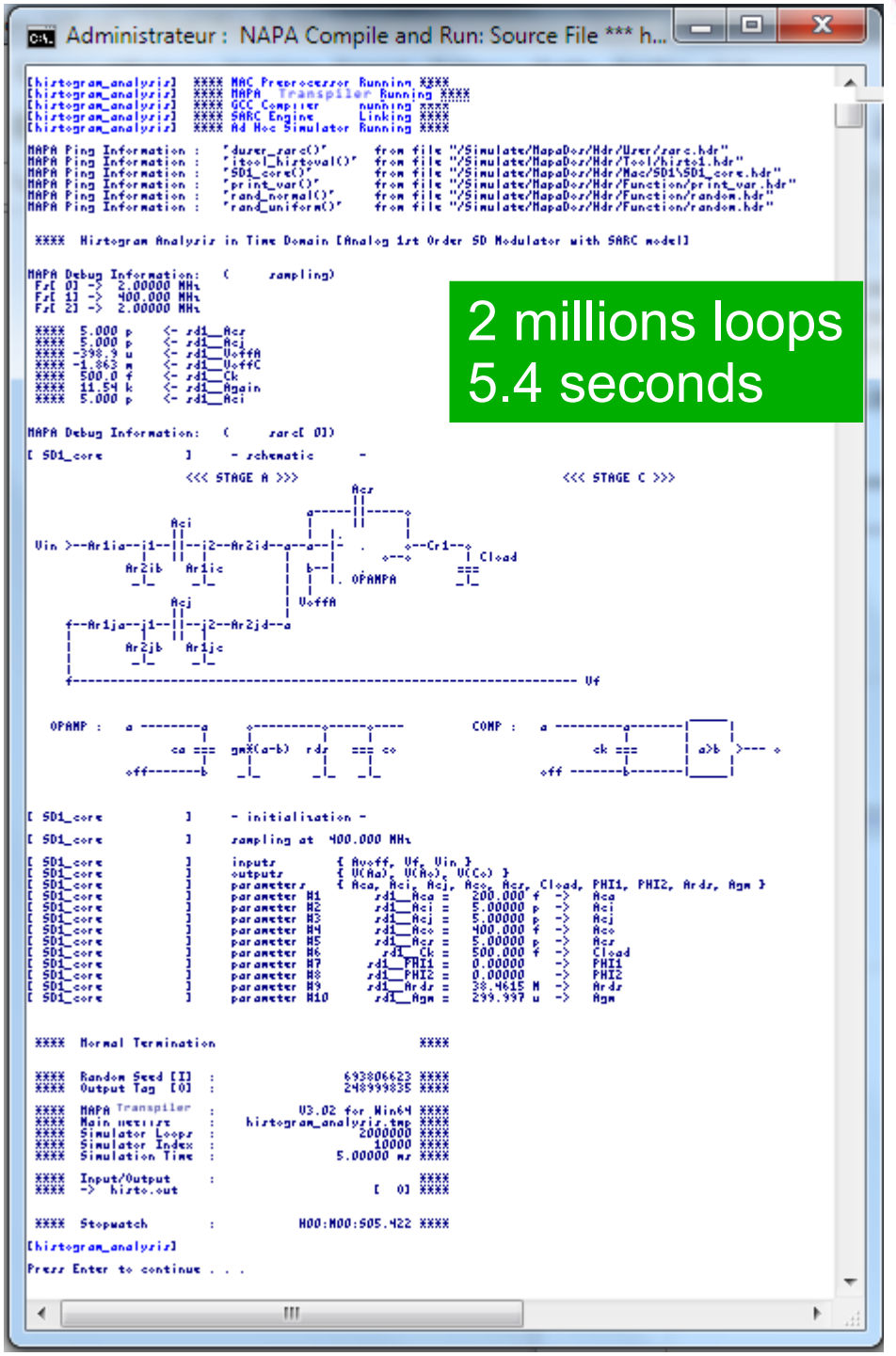
nominal      fs

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, 1.0)
dvar     freq      1234.56789
dvar     ph        rand_uniform(0.0, _2pi_)

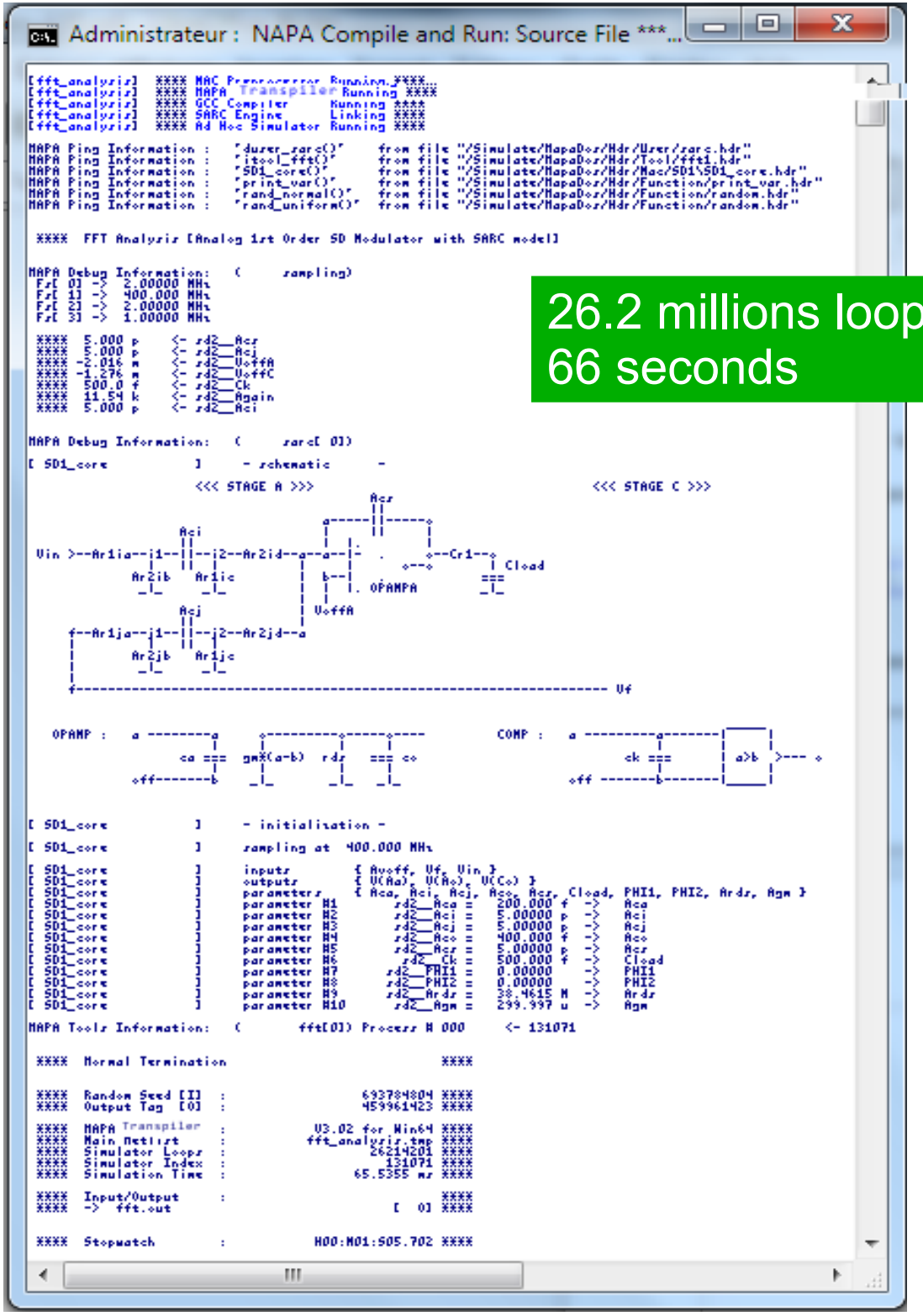
interpolate fs     200
node     Vin       osc   0.0   ampl   freq   ph
node     Vrefa     dc    (analog)    1.0
node     Voutd     cell  sd1   "./sd1.net"  Vin Vrefa  Clk  opfile1..2
node     Vrefd     dc    (digital)   1

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

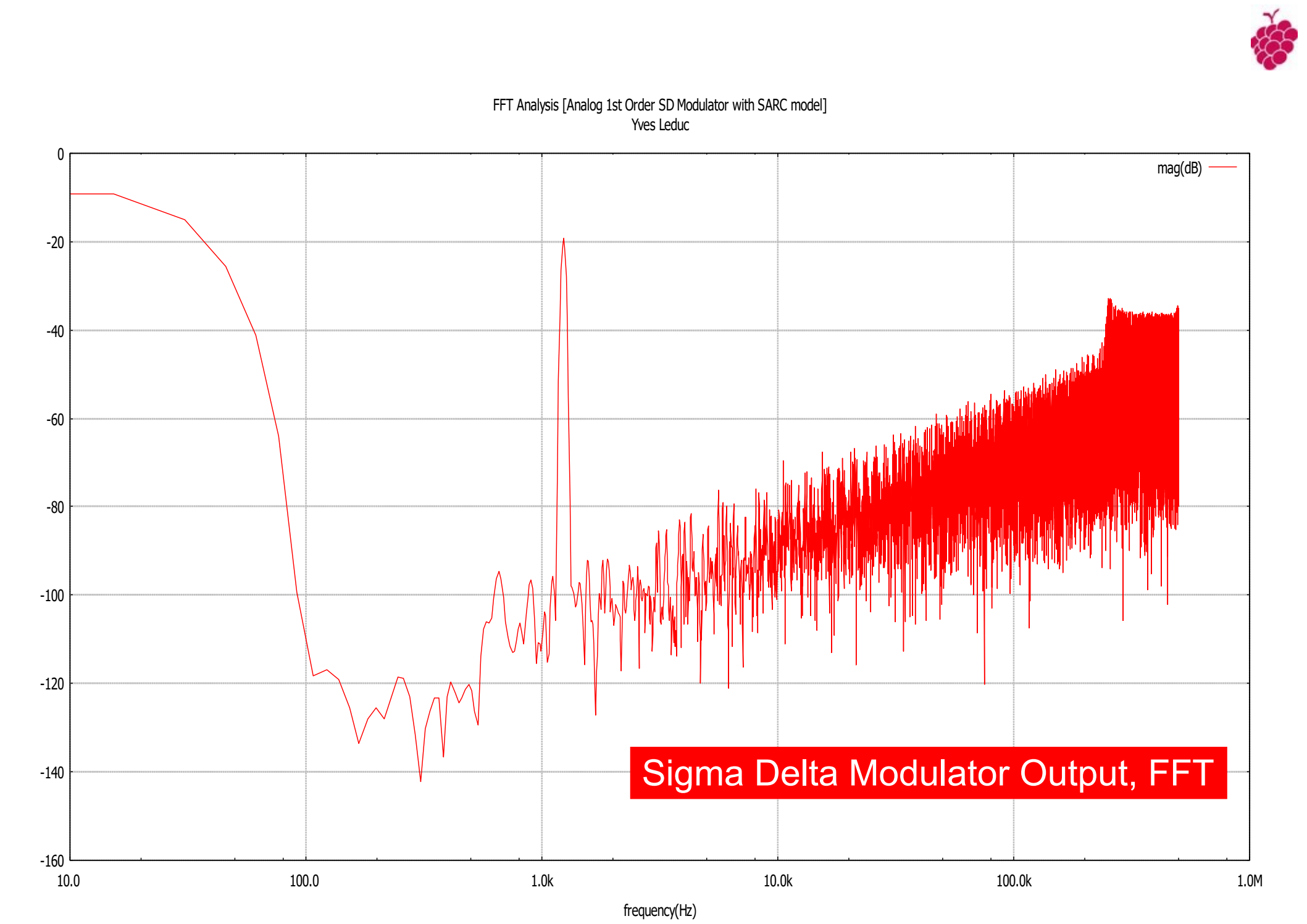
terminate    1 <= TOOL_INDEX

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

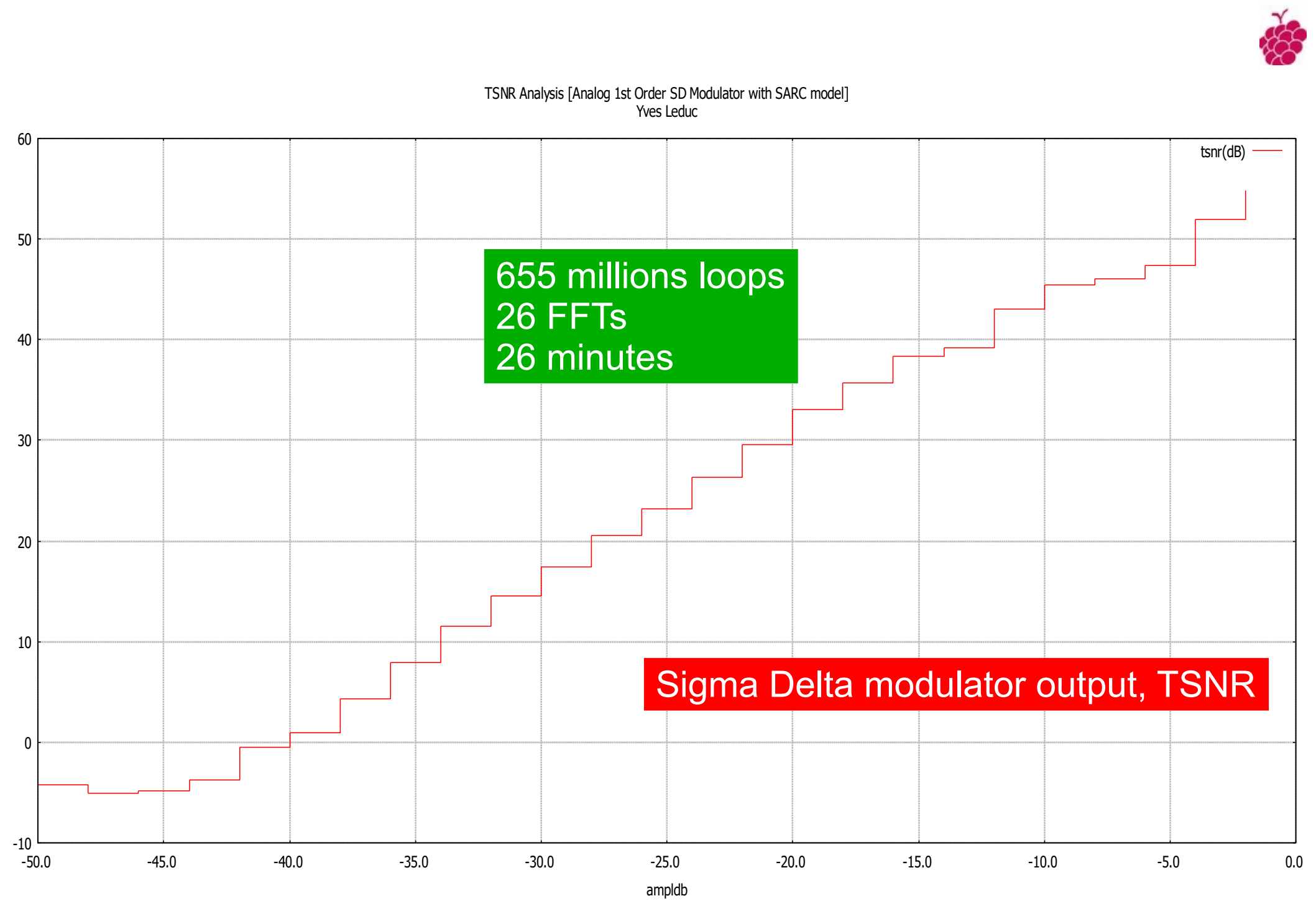
interpolate fs     200
node     Vin        osc    0.0   ampl   freq    ph
node     Vrefa      dc     (analog)    1.0
node     Voutd      cell   sd1   "./sd1.net"    Vin Vrefa    Clk   opfile1..2
node     Vrefd      dc     (digital)   1

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 Reference and in the NAPA Card.

The NAPA Primer describes in detail 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
cell_interface
data_interface
interlude
interpolate
ivar
load
napa_version
node
nominal
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
const
copy
cosine
dac
dalgebra
dc
delay
differentiator
div
dtoi
dtool
duser
equal
fzand
fzbuffer
fzinv
fzor
fznand
fznor
fznot
fzxnor
fzxor
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
test
toggle
track
triangle
trig
uadc
udac
wsum
xnor
xor
zero

NAPA Nodes



ABS(x)
SIGN(x)
MIN(x,y)
MAX(x,y)
CLIP(x,l,h)
ISINSIDE(x,l,h)
ISEQUAL(x,y)
ISSMALL(x)
ISEVEN(x)
MODULO(x,y)
SIN(x)
SQRT(x)
POW(x,y)
DB2LIN(x,r)
LIN2DB(x,r)
LINSWEEP(c,b,e,n)
LOGSWEEP(c,b,e,n)
RAND_01()
IO_MANAGER(c,f,n,s,t)
ISOPTION(f,i,o)

and many more:
see the NAPA Reference,
“Available C Macro Functions”

Built-in C Macros

Activation\chirp.net
Adder\i1.net
ASD\m1.net
Comparator\2_h.net
Counter\counter.net
CT_DOMAIN\Butterworth\LP4.net
DWA\8.net
Filters\nyquist.net
Integrator1\d1.net
Latch\JK_FlipFlop.net
Logic\adder.net
Modulation\fm.net
Noise\pink.net
PWM\1.net
Random\PRNG_16.net
CT_DOMAIN\Remez\FIR_LP.net
Sun\sun_position.net
Thermometer\8.net

and many more: 268 cells,
see the panel NAPA Vineyard,
“Napa Library Doc - Net -”





Adder_op1.dat
Array\4x4.dat
ASD_op1.dat
Comparator_cmp1.dat
Integrator1_opreal.dat
Integrator2_op1.dat
Sun_location.dat
Sun_panel.dat

and many more: 136 data cells,
see the panel NAPA Vineyard,
“Napa Library Doc - Net -”

Data Cells



rand_uniform
rand_normal
rand_gaussian
rand_poisson
linsweep
logsweep
coherent
db2lin
lin2db
GCD
LCM
factorial
sinc
hard_limiter
soft_limiter
compress_mu_law
expand_mu_law
thermometric
halton
vandercorput
root_mean_square
parallel_R

and many more:
see the panel NAPA Vineyard,
“Napa Library Doc - Hdr -”

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 resource functions

C Functions and Macros

duser_fir
iuser_fir
duser_ifft
iuser_lfsr
duser_multitone
duser_pink
duser_pwl
duser_read
duser_resonator
duser_sarc
duser_sun
iuser_sequence
duser_ztrans
iuser_wolfram_rule

itool_fft
itool_tsnr
itool_tf
itool_resp
itool_stepresp
itool_histogram
itool_disto
itool_spectro
itool_cwt
itool_allan
itool_pnoise
itool_adclin
itool_jitter
itool_xcorr

post_histo
post_sort
post_stat
post_csv
post_margins
post_compare

and many more: see the NAPA Reference, chapter “User-Defined Functions and Tools”, and the panel, “Napa Library Doc - Hdr -”



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.

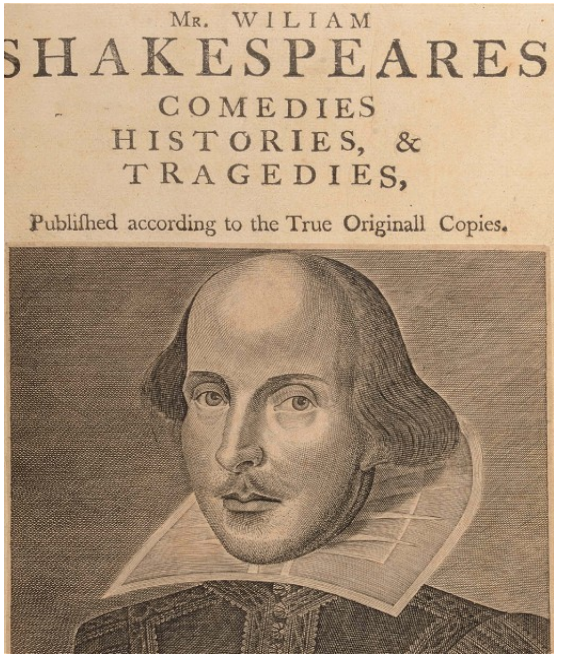
SIMULATE WITH MODERATION
and intelligence...

Questions ?



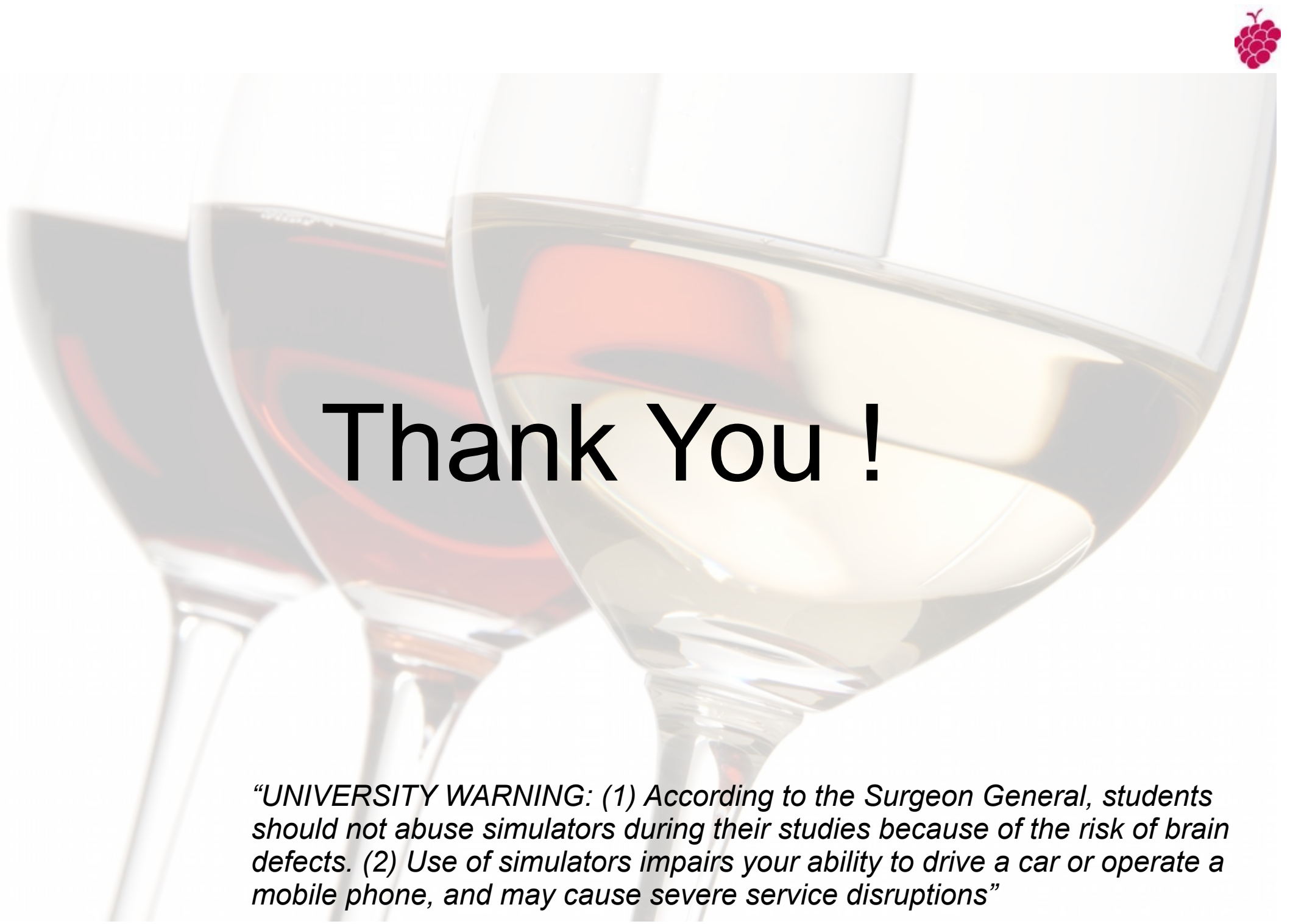
The Last Words

" 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 simulators during their studies 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

Drives

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

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

OS: Microsoft Windows 7 Professional (x64) Build 7601

Close

FYI, my computer