

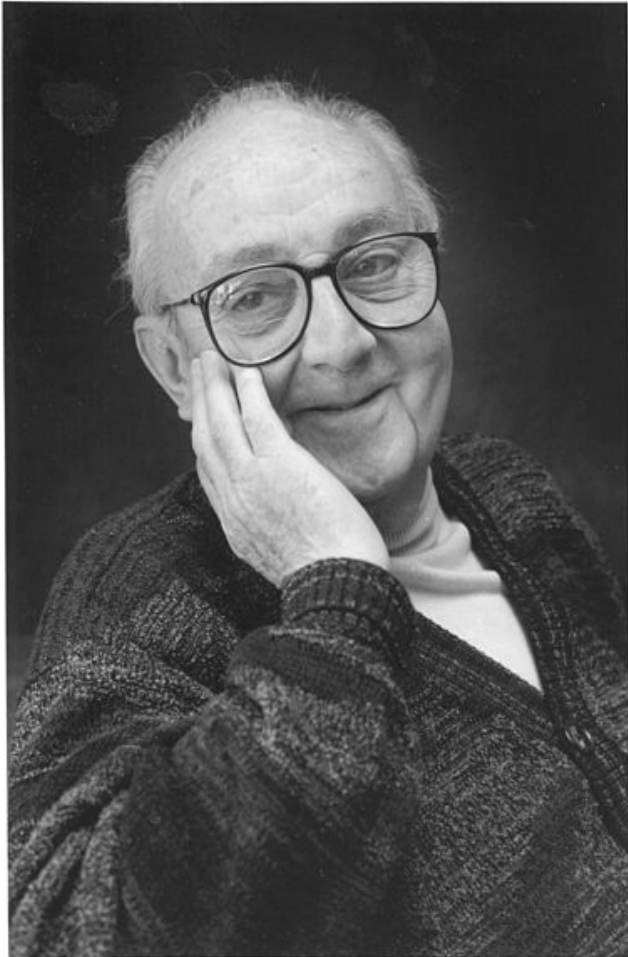
Preamble



"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

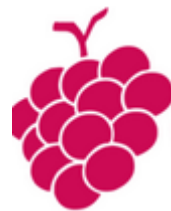
Known for

Response-surface methodology

Box–Jenkins method

Box–Cox transformation

NAPA



A High Level Simulator of Mixed-Signal Systems



Outline

NAPA through a simple example

main netlist

cell

primitive

user function

time domain simulation

smart tool

synchronization

A realistic example

cell generator

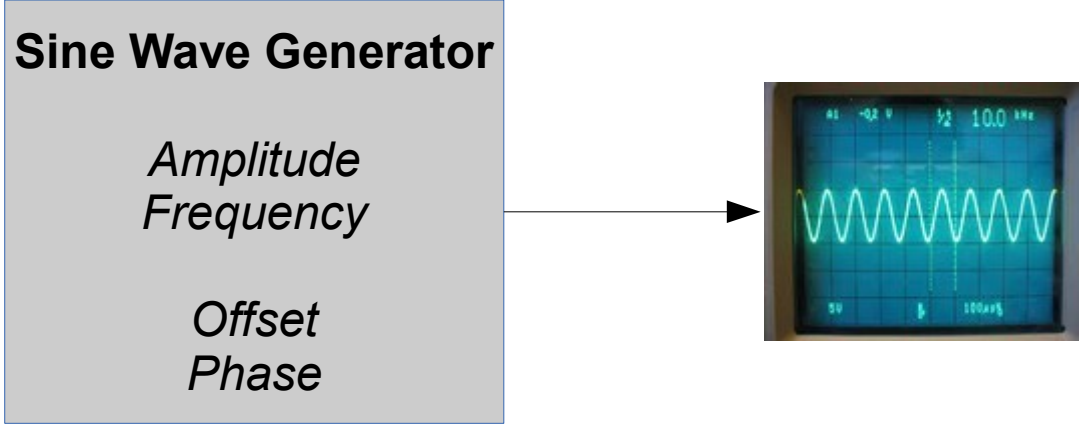
multirate transfer function

more...

Open conclusion

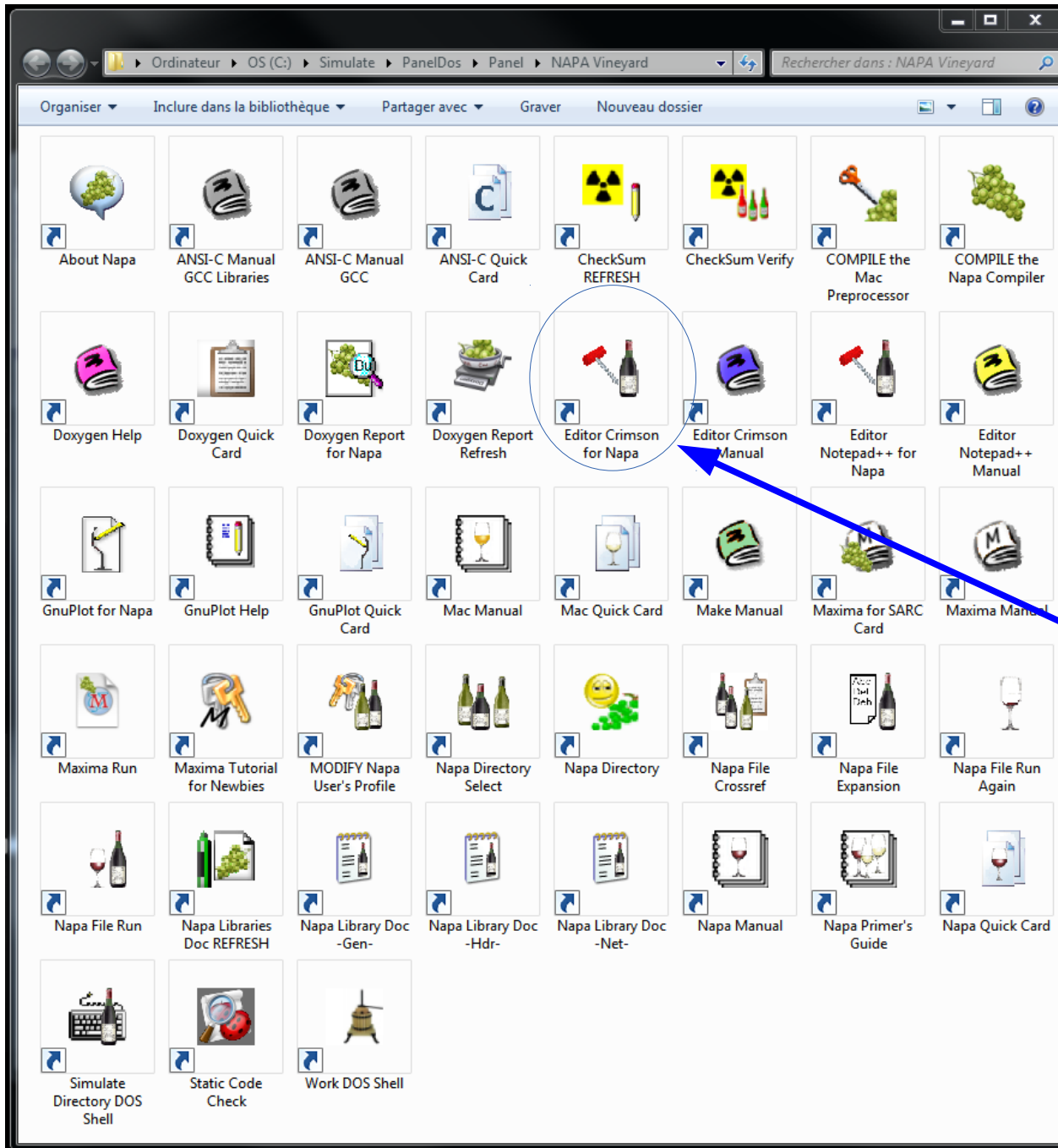


A Simple Example : Generate a Sine Wave



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)

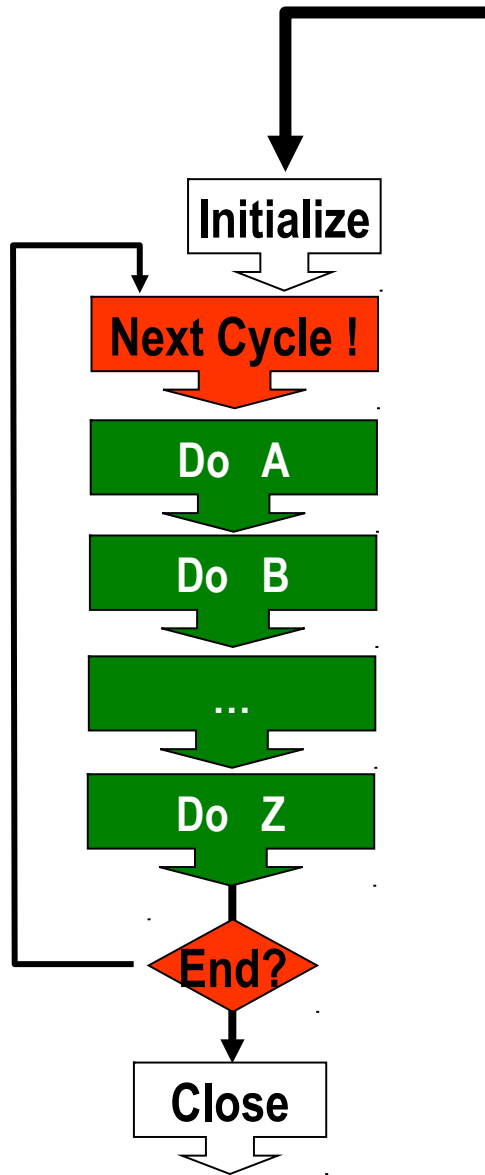
The screenshot shows the Crimson Editor window with the following netlist content:

```
1
2 title "sine"
3
4 header <napatool.hdr>
5
6 fs          1.0e6
7
8 node (s)    sine  1.0 2.0 12345.6 0.0
9
10 terminate  LOOP_INDEX >= 1000000000
11
12
```

A large blue diagonal watermark text "NAPA netlist" is overlaid on the right side of the editor window.

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

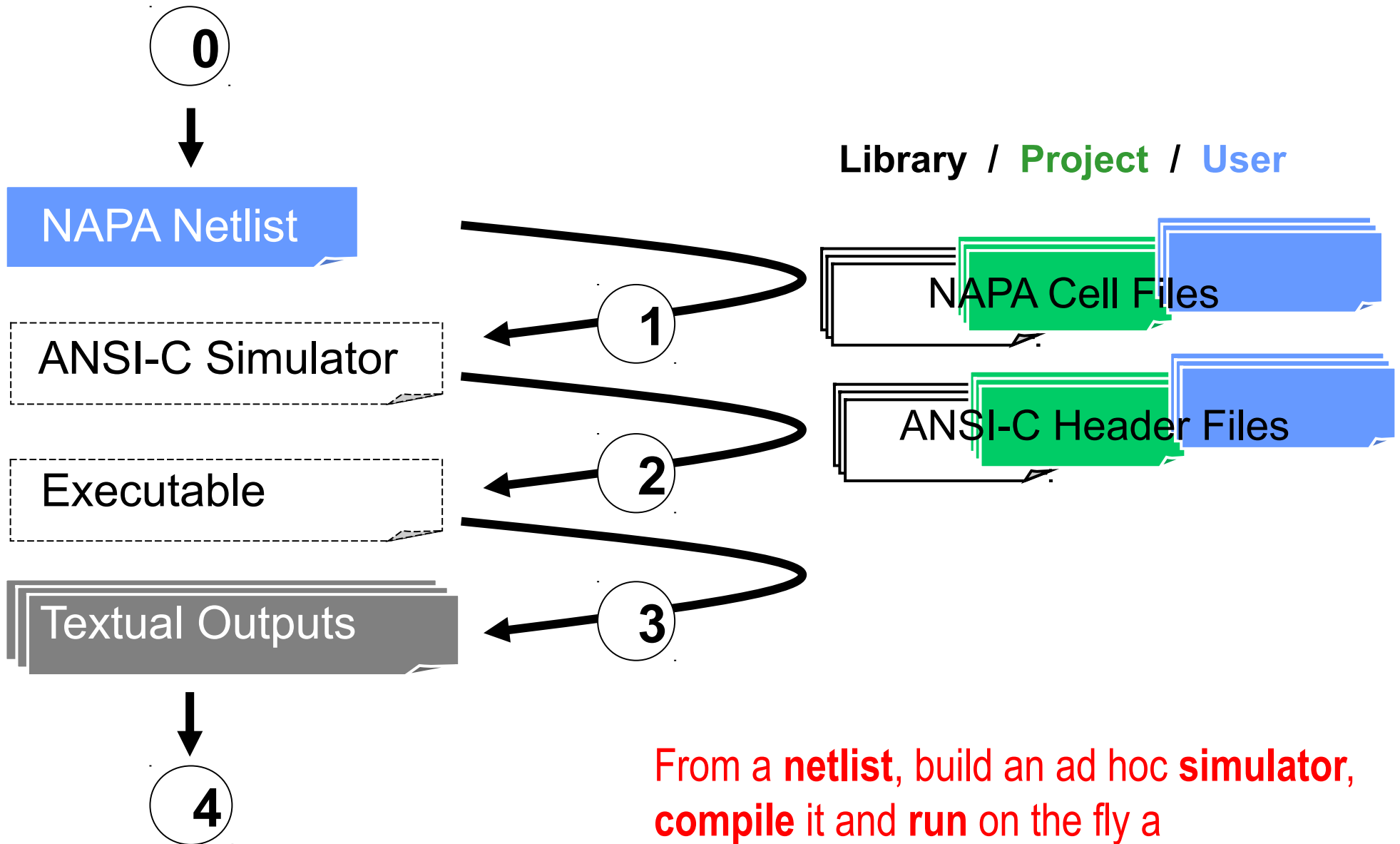
Hazard free

Cycle accurate by construction

Very fast

Static loops detected and rejected

NAPA File Structure and Work Flow

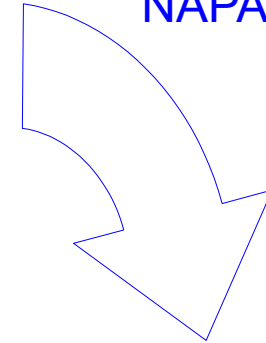


From a **netlist**, build an ad hoc **simulator**,
compile it and **run** on the fly a
streamlined **DEADLY FAST executable**.



```
Crimson Editor - [C:\Simulate_User\Papier\sine.nap]
File Edit Search View Document Project Tools Macros Window Help
resonator.nap resonator.net cell.nap sine.nap
1
2 title "sine"
3
4 header <napatool.hdr>
5
6 fs      1.0e6
7
8 node (s)  sine  1.0 2.0 12345.6 0.0
9
10 terminate LOOP_INDEX >= 1000000000
11
12
Ready Ln 19, Ch 17 21 ASCII, DOS READ REC COL DVF
```

NAPA compilation



file "sine.c"

```
...
napa_abs_loop = 0.0L;

do {
    napa_abs_time = napa_abs_loop * 1.0000000000000000e-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



```
Administrator : NAPA Compile and Run: Source File *** sine.nap *...
[sine] **** MAC Preprocessor Running ****
[sine] **** NAPA Lister Running ****
[sine] **** GCC Compiler Running ****
[sine] **** User's Simulator Running ****

**** sine

**** Normal Termination ****

**** Random Seed [I] : 691480581 ****
**** Output Tag [O] : 733520036 ****

**** NAPA Compiler : U3.01b for Win64 ****
**** Main Netlist : sine.tmp ****
**** Simulator Index : 1000000000 ****
**** Simulation Time : 1.00000 ks ****

**** Input/Output : ****
**** -> sine.log [ O] ****

**** Stopwatch : H00:M00:S58.895 ****

**** LOG File Ready : sine.log ****

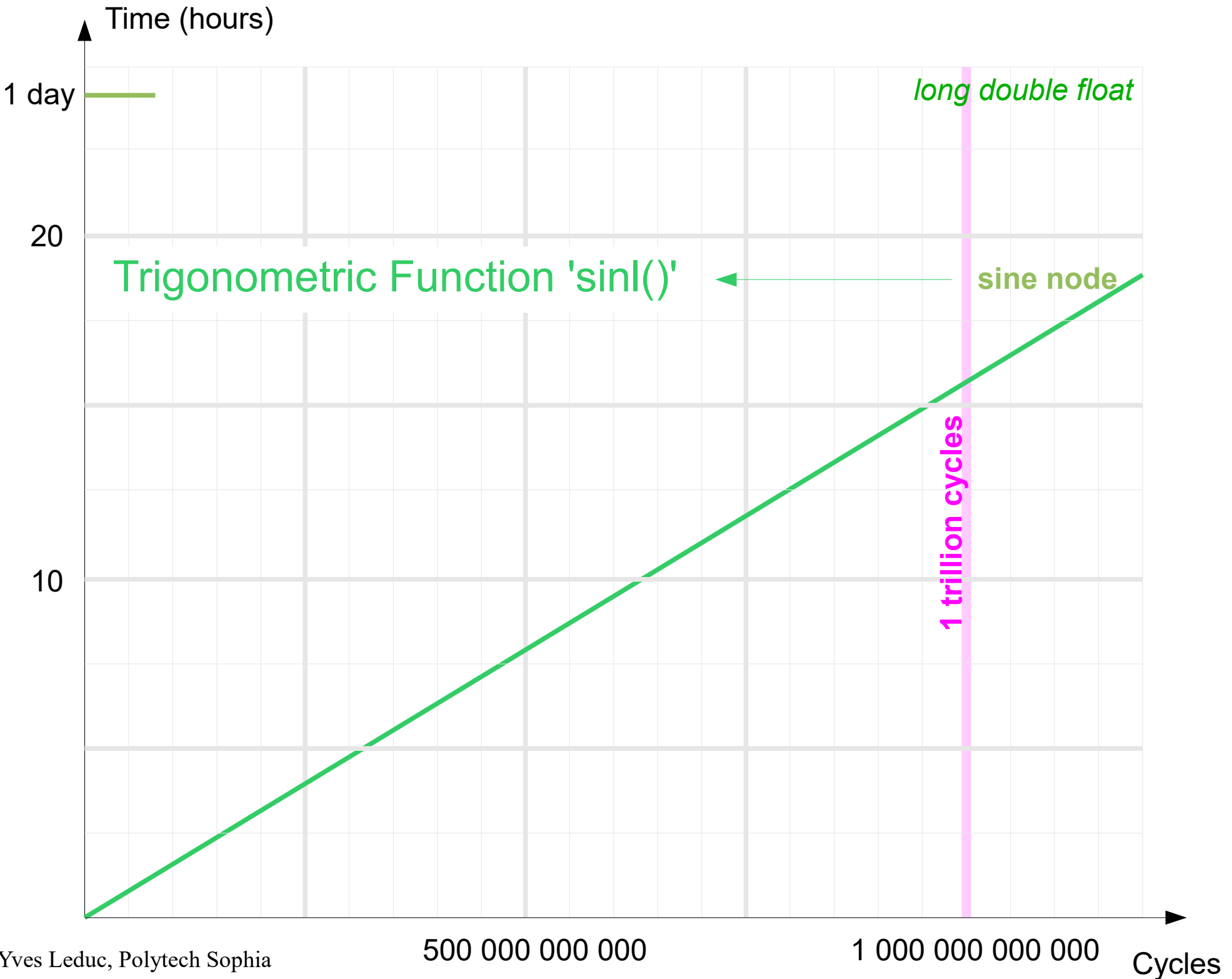
[sine]

Press Enter to continue . . .
```

The simulation running on the screen

*Preprocess
NAPA Compile
C Compile
Binary Execution*

*1 billion cycles
in 59 seconds*





Obviously too SLOW !!!!

Not happy with the current offer ?

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

test

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

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

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

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

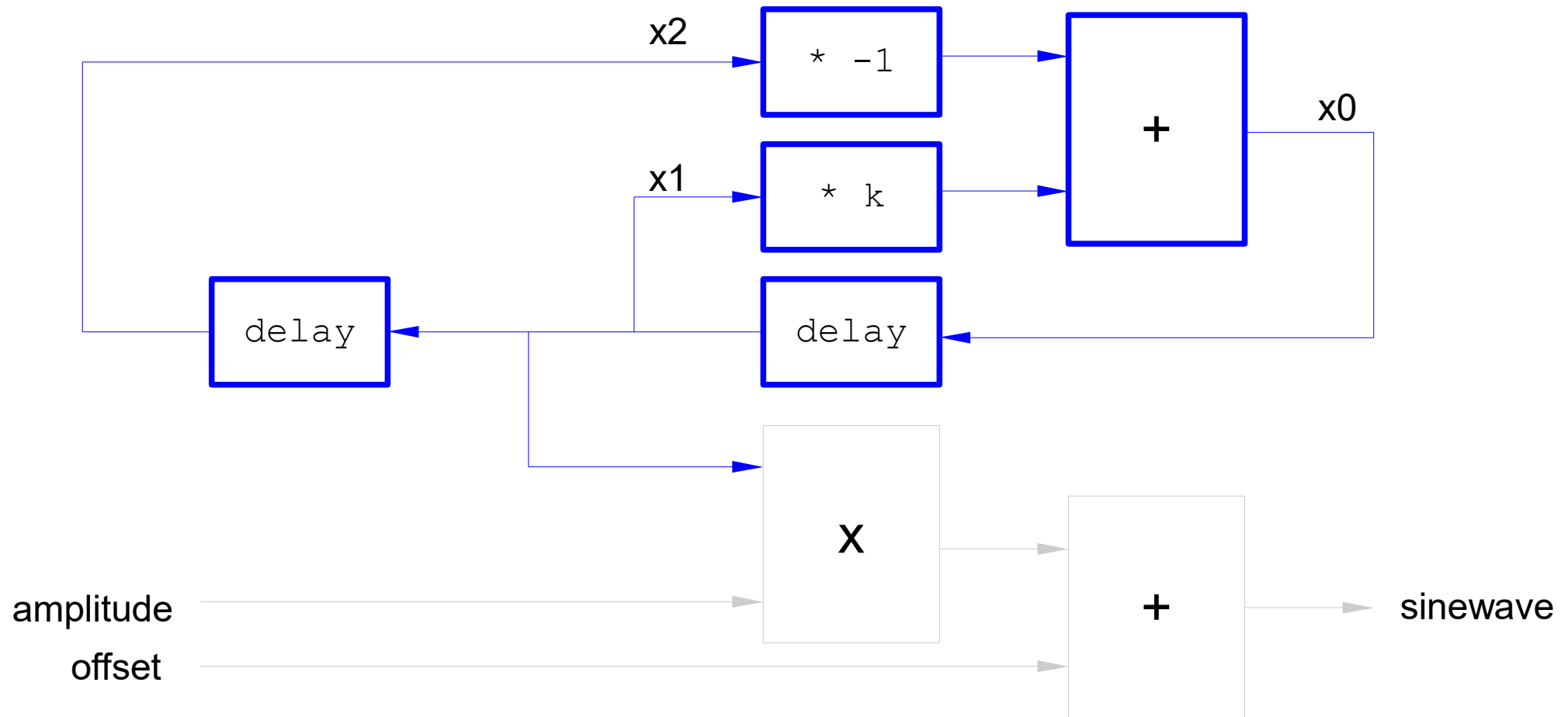
$$X_{n-1} = \sin(\text{phase})$$
$$X_{n-2} = \sin(\text{phase} - (2\pi \text{fsinewave} / \text{fsampling}))$$

The Second Order Resonator



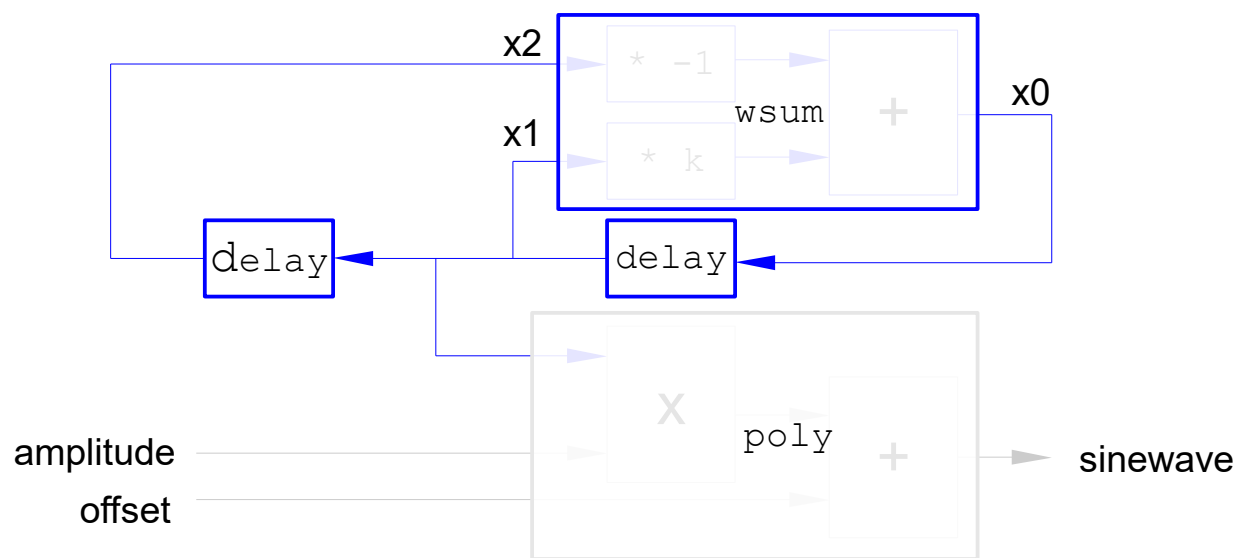
Sine wave frequency, sine wave phase and sampling frequency

parameter k
initial values for x_0, x_1





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 JVF
```



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 DVF
```

```
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 DVF
```





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
[Icons] W REC [Menu] [Stop] [Play] [Home]
1  ## ***** EXPANSION OF FILE cell.tmp ***** START *****
2  [ ]
3  header <napatool.hdr>[ ]
4  [ ]
5  title "cell"[ ]
6  [ ]
7  fs 1.0e6[ ]
8  [ ]
9  ## >> npde 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 DVF
```



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

```
Crimson Editor - [C:\Simulate_User\Papier\cell.lst]
File Edit Search View Document Project Tools Macros Window Help
osc.nap fft.nap cell.nap cell.lst
1 /* ***** CROSS REFERENCE for cell.tmp *****
2
3 List of Files
4 A. -> "cell.tmp"
5 B. -> "resonator.net"
6
7 *****
8
9 List of Headers
10 A.2 <- /Simulate/Napados/Hdr/napatool.hdr
11
12 Sampling Information
13 A.6 <- [ main sampling frequency ]
14
15 List of Nodes
16 A.8 B.11 <- s
17 A.8 B.7 <- sr_x0
18 A.8 B.8 <- sr_x1
19 A.8 B.9 <- sr_x2
20
21 List of Variables
22 A.8 B.3 <- sr_k
23 A.8 B.13 <- [ init ]
24 A.8 B.14 <- [ init ]
25
26 List of Declarations
27 A.8 B.5 <- sr_x0
28
29 Terminate
30 A.10 <- [ terminate ]
31
32 ***** CROSS REFERENCE for cell.tmp *****
33
```



A Description Using a NAPA Primitive

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



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

```
File Edit Search View Document Project Tools Macros Window Help
resonator.nap resonator.net cell.nap sine.nap
1
2 title "sine"
3
4 header <napatool.hdr>
5
6 fs 1.0e6
7
8 node (s) sine 1.0 2.0 12345.6 0.0
9
10 terminate LOOP_INDEX >= 1000000000
11
12
```

NAPA Built-in Sine

Ready Ln 19, Ch 17 21 ASCII, DOS READ REC COL JVF

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

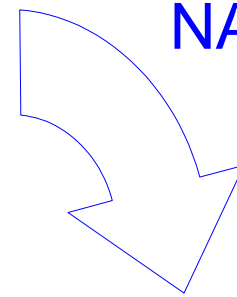
```
File Edit Search View Document Project Tools Macros Window Help
resonator.nap resonator.net cell.nap sine.nap osc.nap
1
2 title "osc"
3 header <napatool.hdr>
4
5 fs 1.0e6
6
7 node (s) osc 1.0 2.0 12345.6 0.0
8
9 terminate LOOP_INDEX >= 1000000000
10
11
12
```

NAPA Built-in Resonator

Ready Ln 13, Ch 1 16 ASCII, DOS READ REC COL JVF



```
Crimson Editor - [C:\Simulate_User\Papier\osc.nap]
File Edit Search View Document Project Tools Macros Window Help
resonator.nap resonator.net cell.nap sine.nap osc.nap
1
2 title "osc"
3 header <napatool.hdr>
4
5 fs 1.0e6
6
7 node (s) osc 1.0 2.0 12345.6 0.0
8
9 terminate LOOP_INDEX >= 1000000000
10
11
12
Ready
```



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*



```
Crimson Editor - [C:\Simulate_User\Papier\func.nap]
File Edit Search View Document Project Tools Macros Window Help
func.nap
1
2 title "function resonator"
3
4 header <napatool.hdr>
5 header "/Simulate/Napados/Hdr/Activation/resonator.hdr"
6
7 fs      1.0e6
8
9 node (s)  duser resonator  1.0 2.0 12345.6 0.0
10
11 terminate LOOP_INDEX >= 1000000000
12
```

'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)
```

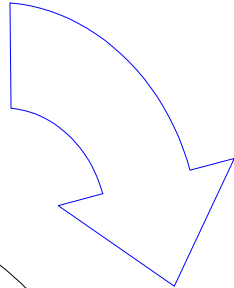


```

Crimson Editor - [C:\Simulate_User\Papier\func.nap]
File Edit Search View Document Project Tools Macros Window Help
resonator.nap resonator.net cell.nap sine.nap func.nap resonator.hdr
1
2 title "function resonator"
3
4 header <napatool.hdr>
5
6 fs      1.0e6
7
8 node (s)  duser resonator  1.0 2.0 12345.6 0.0
9
10 terminate LOOP_INDEX >= 1000000000
11
12 ping
13

```

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



```

Crimson Editor - [C:\Simulate_User\Papier\func.nap]
File Edit Search View Document Project Tools Macros Window Help
resonator.nap resonator.net cell.nap sine.nap func.nap resonator.hdr
1 |
2 | title "function resonator"
3 |
4 | header <napatool.hdr>
5 |
6 | fs 1.0e6
7 |
8 | node (s) duser resonator 1.0 2.0 12345.6 0.0
9 |
10 | terminate LOOP_INDEX >= 1000000000
11 |
12 | ping
13 |
Ready Ln 16, Ch 1 16 ASCII, DOS READ REC COL JVF

```

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 ?



```
Crimson Editor - [C:\Simulate_User\Papier\osc.nap]
File Edit Search View Document Project Tools Macros Window Help
resonator.nap resonator.net cell.nap sine.nap osc.nap
1
2 title "osc"
3 header <napatool.hdr>
4
5 fs      1.0e6
6
7 node (s)  osc  1.0 2.0 12345.6 0.0
8
9 terminate LOOP_INDEX >= 1000000000
10
11
12
Ready
```

Run Simulation

```
Administrator : NAPA Compile and Run: Source File *** osc.nap ***
[osc] **** MAC Preprocessor Running ****
[osc] **** NAPA Lister Running ****
[osc] **** GCC Compiler Running ****
[osc] **** User's Simulator Running ****

**** osc

**** Normal Termination ****

**** Random Seed [I] : 691480847 ****
**** Output Tag [O] : 918813764 ****

**** NAPA Compiler : U3.01b for Win64 ****
**** Main Netlist : osc.tmp ****
**** Simulator Index : 1000000000 ****
**** Simulation Time : 1.00000 ks ****

**** Input/Output : ****
**** -> osc.log [ 0] ****

**** Stopwatch : H00:M00:S04.679 ****

**** LOG File Ready : osc.log ****

[osc]

Press Enter to continue . . .
```

1 billion clock cycles ...

... in 4.7 seconds



```
Crimson Editor - [C:\Simulate_Examples\TNAPA_Running_Teaser\osc.nap]
File Edit Search View Document Project Tools Macros Window Help
osc.nap
1
2 title "osc"
3 header <napatool.hdr>
4
5 fs 1.0e6
6
7 node (s) osc 1.0 2.0 12345.6 0.0
8
9 terminate LOOP_INDEX >= 10000000000
10
11
Ready
```

Run Simulation

```
Administrateur : NAPA Compile and Run: Source File *** osc.nap ***
[osc] **** MAC Preprocessor Running ****
[osc] **** NAPA Compiler Running ****
[osc] **** GCC Compiler Running ****
[osc] **** Ad Hoc Simulator Running ****

**** osc ****

**** Normal Termination ****

**** Random Seed [I] : 691583622 ****
**** Output Tag [O] : 326537381 ****

**** NAPA Compiler : U3.01b for Win64 ****
**** Main Netlist : osc.tmp ****
**** Simulator Index : 1000000000000 ****
**** Simulation Time : 1.00000 Ms ****

**** Input/Output : ****
**** -> osc.log [ O] ****

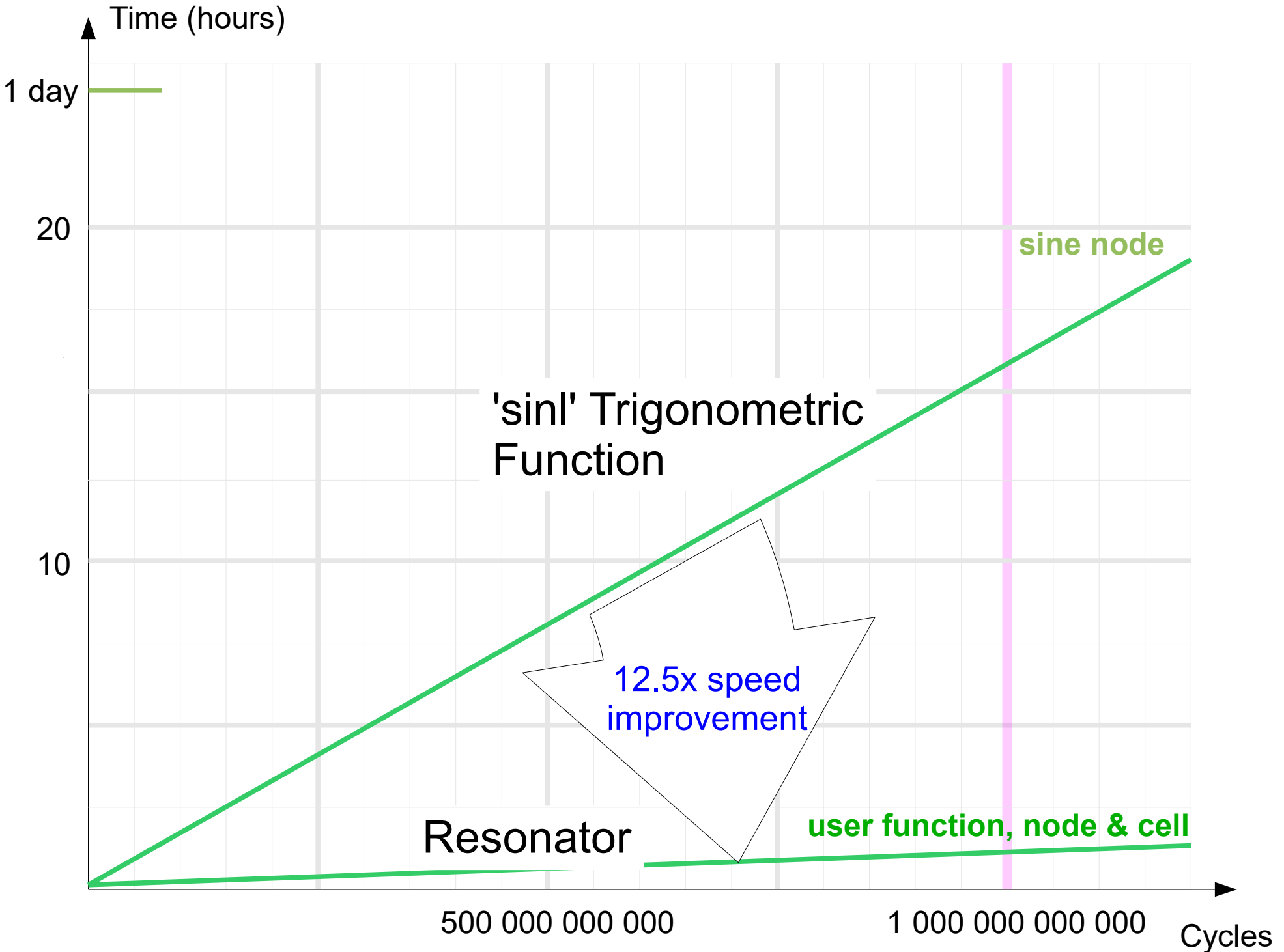
**** Stopwatch : H01:M06:S52.608 ****

**** LOG File Ready : osc.log ****

[osc]
Press Enter to continue .
```

1 TRILLION clock cycles ...

... in 1 hour and 7 minutes





Cool...

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



```

Crimson Editor
File Edit Search View Document Project Tools Macros Window Help
output.nap
C:\Simulate_WORKKONGOING\NAPA_Running_Teaser\output.nap
1
2 title "output"
3
4 header <napatool.hdr>
5
6 fs 1.0e6
7
8 dvar freq 12345.6789
9 dvar per 1.0 / freq
10 dvar ph rand_uniform(0.0, _2pi_)
11
12 dvar k 10.0e6 // 10 millions of periods of sinewave
13
14 dvar t1 TIME >= ( k * per) &update
15 dvar t2 TIME > ((k+1.0) * per) &update
16
17 event prt t1 && !t2
18
19 node out1 osc 1.0 2.0 freq ph
20 node out2 sine 1.0 2.0 freq ph
21 node err sum out1 -out2
22
23 output stdout out1(_Volt) out2(_Volt) err(n_Volt) when prt
24
25 terminate t2
26
27 ping

```

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



```

Administrator : NAPA Compile and Run: Source File *** output.nap ***

[output] **** MAC Preprocessor Running ****
[output] **** NAPA Simulator Running ****
[output] **** GCC Compiler Running ****
[output] **** User's Simulator Running ****

NAPA Ping Information : 'rand_uniform()' from file "/Simulate/NapaDos/Hdr/Function/random.hdr"

**** output

**** Normal Termination

**** Random Seed [I] : 691489775
**** Output Tag [O] : 6320647

**** NAPA Compiler : U3.01b for Win64
**** Main Netlist : output.tmp
**** Simulator Index : 810005267
**** Simulation Time : 810.005 s

**** Input/Output :
**** -> stdout [ O ]

**** Stopwatch : H00:M03:S36.659

[output]

Press Enter to continue . . .

```

```

Crimson Editor - [C:\Simulate_Examples\NAPA_Running_Examples\TEASER\output.out]
File Edit Search View Document Project Tools Macros Window Help
output.nap output.out
1 # output
2 # (time domain output )
3 # (compiler version ) NAPA V3.01b for Win64
4 # (source file ) output.tmp
5 # (random seed ) 691489775
6 # (output sampling rate) 1.00000 MHz, controlled by ' prt '
7 # (number of columns ) 4
8 #
9 #
10 #
11 #
12 #
13 #
14 # Mon Oct 28 17:59:11 2013 by Anonymous User
15 # absolute_time(s) out1(Volt) out2(Volt) err(nVolt)
16 8.100051850000000e+002 -3.569025055136e-001 -3.569025057639e-001 2.503275364774e-001
17 8.100051860000000e+002 -4.666804491544e-001 -4.666804493837e-001 2.293178980040e-001
18 8.100051870000000e+002 -5.676377168469e-001 -5.676377170588e-001 2.118389907935e-001
19 8.100051880000000e+002 -6.591671474722e-001 -6.591671476607e-001 1.885576139671e-001
20 8.100051890000000e+002 -7.407182793070e-001 -7.407182794751e-001 1.681674799414e-001
21 8.100051900000000e+002 -8.118006605235e-001 -8.118006606667e-001 1.431679219621e-001
22 8.100051910000000e+002 -8.719867987865e-001 -8.719867989069e-001 1.203519506277e-001
23 8.100051920000000e+002 -9.209147322095e-001 -9.209147323037e-001 9.423017921506e-002
24 8.100051930000000e+002 -9.582902062084e-001 -9.582902062765e-001 6.805445096347e-002
25 8.100051940000000e+002 -9.838884431606e-001 -9.838884432035e-001 4.291367261544e-002
26 8.100051950000000e+002 -9.975554942266e-001 -9.975554942432e-001 1.656585979504e-002
27 8.100051960000000e+002 -9.992091652048e-001 -9.992091651953e-001 -9.553469126899e-003
28 8.100051970000000e+002 -9.888395108505e-001 -9.888395108151e-001 -3.544120552590e-002

```

Output redirected from "stdout"



```
gnuplot
File Plot Expressions Functions General Axes Chart Styles 3D Help
ChDir Select&Draw Draw Save Plot Replot Xlin Xlog NoScale Save Profile Load Profile
Current Working Directory is: C:\Simulate_Teaser\TEASER

***** Welcome to Gnuplot 4.63 Profiled for Napa !

<ChDir> ..... Change working directory
<Select&Draw> ..... Select and Draw the data
<Draw> ..... Draw what was saved with <Save>
<Save> ..... Save Plot

<Replot>

<Xlin> ..... Get a linear scale on X axis
<Xlog> ..... Get a logarithmic scale on X axis
<NoScale> ..... Remove all zooming parameters

<Save Profile> .... Save user's profile
<Load Profile> .... Load user's profile

NB: last selection, if any, is saved in '_my_last_gnuplot.plt'

*****

Terminal type set to 'windows'
gnuplot>
```

```
Administrateur : *** Prepare Plot for Gnuplot ***

Y
.
.
.
*** PLOT 2D ***
.
.
.....X

Column 1: absolute_time(s)
Column 2: out1(Volt)
Column 3: out2(Volt)
Column 4: err(nVolt)

Enter X and Y (or 'q'):
-> 1 2

Column 1: absolute_time(s)
Column 2: out1(Volt)
Column 3: out2(Volt)
Column 4: err(nVolt)

Enter X and Y (or 'q'):
-> 1 3

Column 1: absolute_time(s)
Column 2: out1(Volt)
Column 3: out2(Volt)
Column 4: err(nVolt)

Enter X and Y (or 'q'):
-> 1 4

Column 1: absolute_time(s)
Column 2: out1(Volt)
Column 3: out2(Volt)
Column 4: err(nVolt)

Enter X and Y (or 'q'):
-> q
```

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 !**

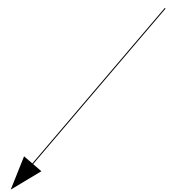


```

Crimson Editor - [C:\Simulate_User\Papier\fft.nap]
File Edit Search View Document Project Tools Macros Window Help
fft.nap
1 title "#n FFT"
2 header <napatool.hdr>
3
4 fs 1.0e6
5
6 dvar freq 12345.6
7 dvar ph rand_uniform(0.0, _2pi_)
8
9 node out osc 1.0 2.0 freq ph
10
11 ivar n 4
12 ivar npts1 POWEROF2(18)
13 ivar npts2 100000000
14
15 dvar bw 100.0e6
16
17 tool fft "ffts.out" out 1.0 bw npts1
18 tool synchro npts2
19
20 directive WINDOW BLACKMAN_HARRIS_7
21
22 terminate TOOL_INDEX >= n
23
24 ping
25 debug TOOL
26
Ready Ln 29, Ch 1 29 ASCII, DOS READ REC COL V

```

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.

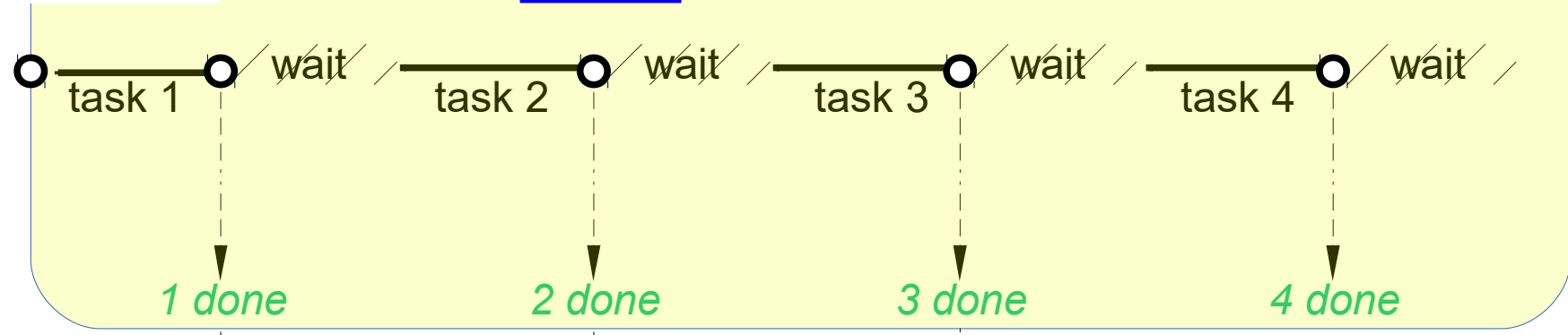
Variable '**napa_tool_index**' handled by the simulator counts the number of tasks already completed and is often used to control the end of the simulation.

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



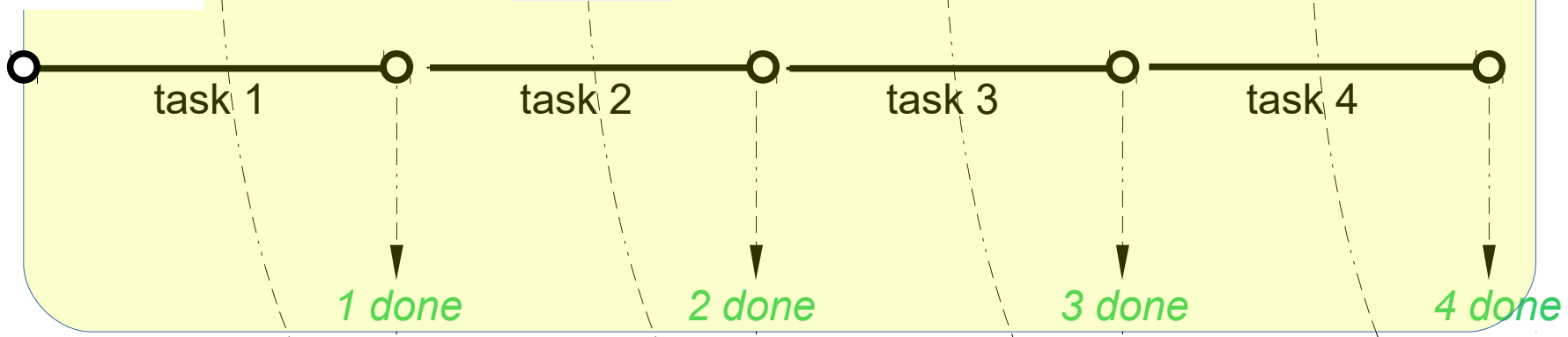
FFT 2^{18} points

tool 1



Count until 10^8

tool 2

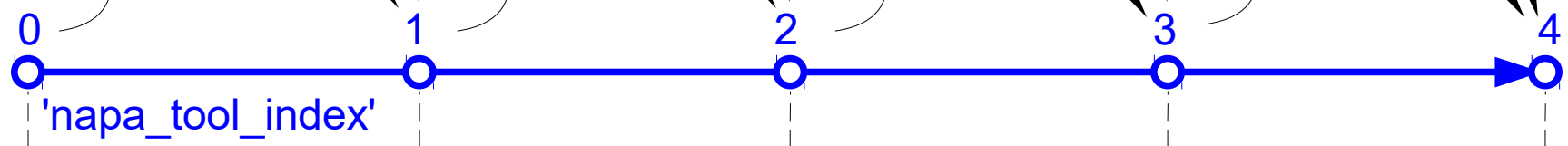


do task 1!

do task 2!

do task 3!

do task 4!



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'



```

Administrator : 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 <- 999999999
NAPA Tools Information: < fft[0] Collect # 001.000 <- 1000000000
NAPA Tools Information: < synchro[0] Collect # 001.000 <- 1000000000
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 <- 2000000000
NAPA Tools Information: < synchro[0] Collect # 002.000 <- 2000000000
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 <- 3000000000
NAPA Tools Information: < synchro[0] Collect # 003.000 <- 3000000000
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 : U3.01b for Win64 ****
**** Main Netlist : fft.tmp ****
**** Simulator Index : 400000000 ****
**** Simulation Time : 400.000 s ****

**** Input/Output : ****
**** -> ffts.out [ O] ****

**** Stopwatch : H00:M00:S17.623 ****

[fft]

```

FFT 1

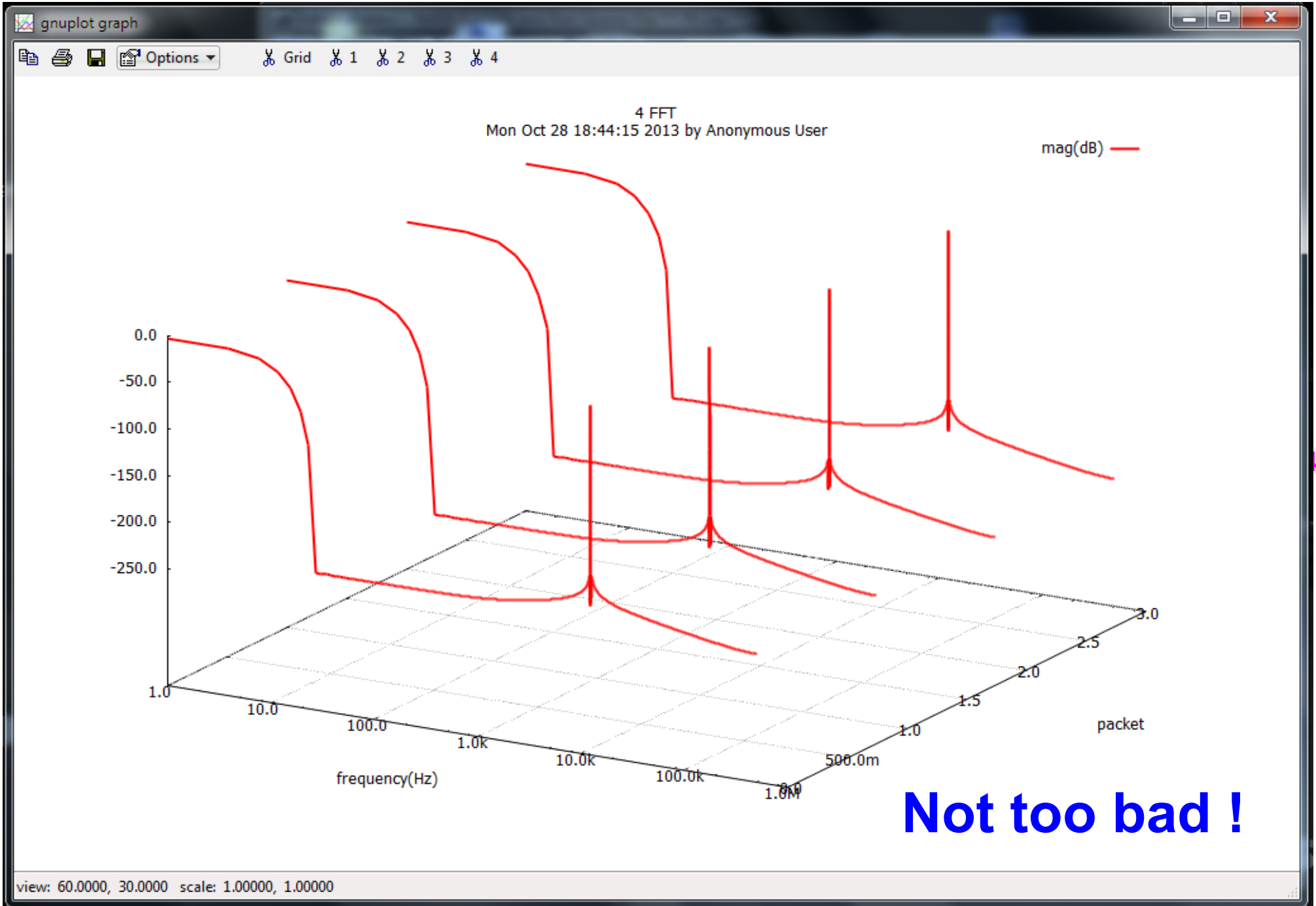
FFT 2

FFT 3

FFT 4

**400 millions cycles,
RMS of 4 256k points FFT,
in less than 18 seconds**

Frequency Domain Output, 3D Plot



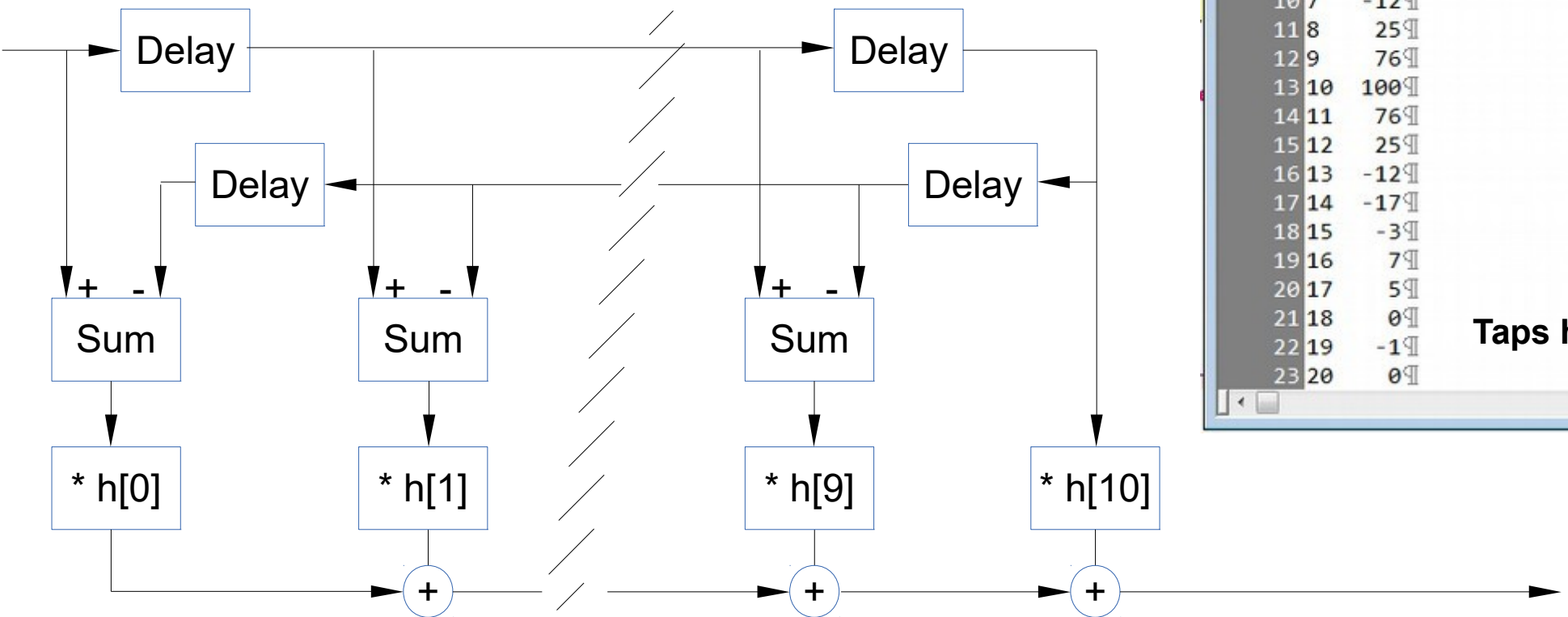
Not too bad !

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 $h[i]$



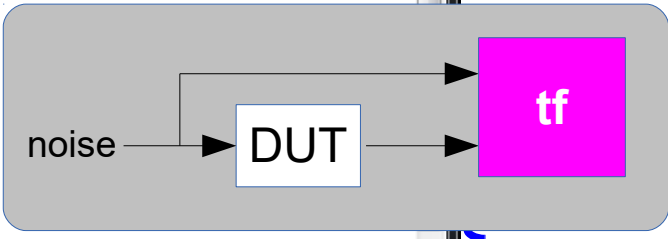
Digital Folded FIR



```

1
2 title "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 tool tf   "transfer_function.out" in 1 out 1 npts
17
18 terminate TOOL_INDEX >= 1
19
20 ping

```



**Digital FIR
Cell generator**

Transfer function

RMS of FFTs

On the fly cell generation
using a file containing the taps

```

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

[tf] **** MAC Preprocessor Running ****
[tf] **** 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'

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

NAPA Ping Information : 'itool_tf<>' from file "/Simulate/NapaDos/Hdr/Tool/fft3.hdr"
NAPA Ping Information : 'rand_bernoulli<>' from file "/Simulate/NapaDos/Hdr/Function/random.hdr"

**** Transfer Function of a Symmetrical FIR

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] : 691491889 ****
**** Output Tag [O] : 291896600 ****

**** NAPA Compiler : U3.01b for Win64 ****
**** Main Netlist : tf.tmp ****
**** Simulator Index : 655360 ****
**** Simulation Time : 655.359 ms ****

**** Input/Output : ****
**** -> transfer_function.out [ O] ****

**** Stopwatch : H00:M00:S00.721 ****

[tf]

Press Enter to continue . . .

```

0.7 second



Taps



Cell produced on the fly by generator 'fir'

```

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

```

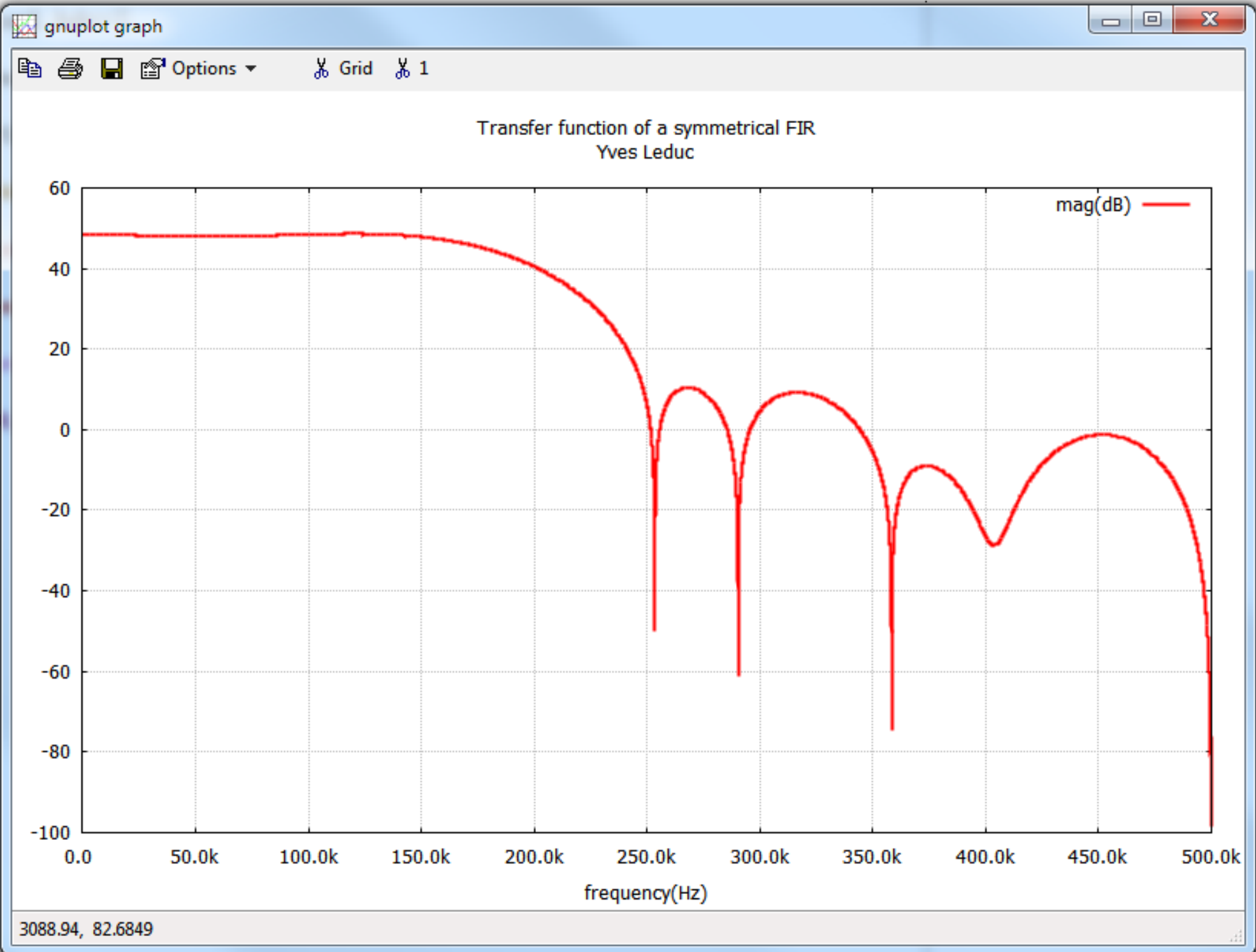
```

C:\Simulate_User\Papier\sfl1_0.gen
1 cell interface $y $dummy $x
2
3 Symmetrical Digital FIR Structure
4 # Taps File Name: 'fir5.tap'
5 # Creation Time : Thu Sep 19 11:33:23 2013
6 # By Command    : \Simulate\Napados\Gen\fir
7 # Generator Version 3.17
8
9 FOLDED FIR
10 #
11 #
12 # x +---> Delay >---+ d1 ... +---> Delay >---+ d10
13 #
14 #
15 #
16 # +---< Delay <---+ d20 ... +---< Delay <---+ d11
17 #
18 #
19 # (+) (+) (+) (+)
20 # Sum Sum Sum
21 #
22 # s0 s1 s9
23 #
24 # *h0 *h1 *h9 *h10
25 #
26 # g0 g1 g9 g10
27 #
28 #
29 # +---> (+) ... +---> (+)---> y
30
31 ivar $h0      0
32 ivar $h1     -1
33 ivar $h2      0
34 ivar $h3      5
35 ivar $h4      7
36 ivar $h5     -3
37 ivar $h6    -17
38 ivar $h7    -12
39 ivar $h8     25
40 ivar $h9     76
41 ivar $h10    100
42
43 node $d1      delay $x
44 node $d2      delay $d1
45 node $d3      delay $d2

```

node out0 generator sfl1 <fir> "~/fir5.tap" in'

generator 'fir'



Step and Impulse Response ?



```
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> 0.50
13
14 node out generator sf11 <fir> "~/fir5.tap" in
15
16 tool resp "response.out" in 1 out 1 npts
17
18 terminate TOOL_INDEX >= 1
19
20 ping
21
```

Diagram illustrating the DUT (Device Under Test) block, labeled "DUT", receiving "noise" input and producing "resp" output.



```

Administrator : 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/Napa
NAPA Ping Information : 'rand_bernoulli()' from file "/Simulate/Napa

**** 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 : U3.01b for Win64 ****
**** Main Netlist : resp.tmp ****
**** Simulator Index : 655360 ****
**** Simulation Time : 655.359 ms ****

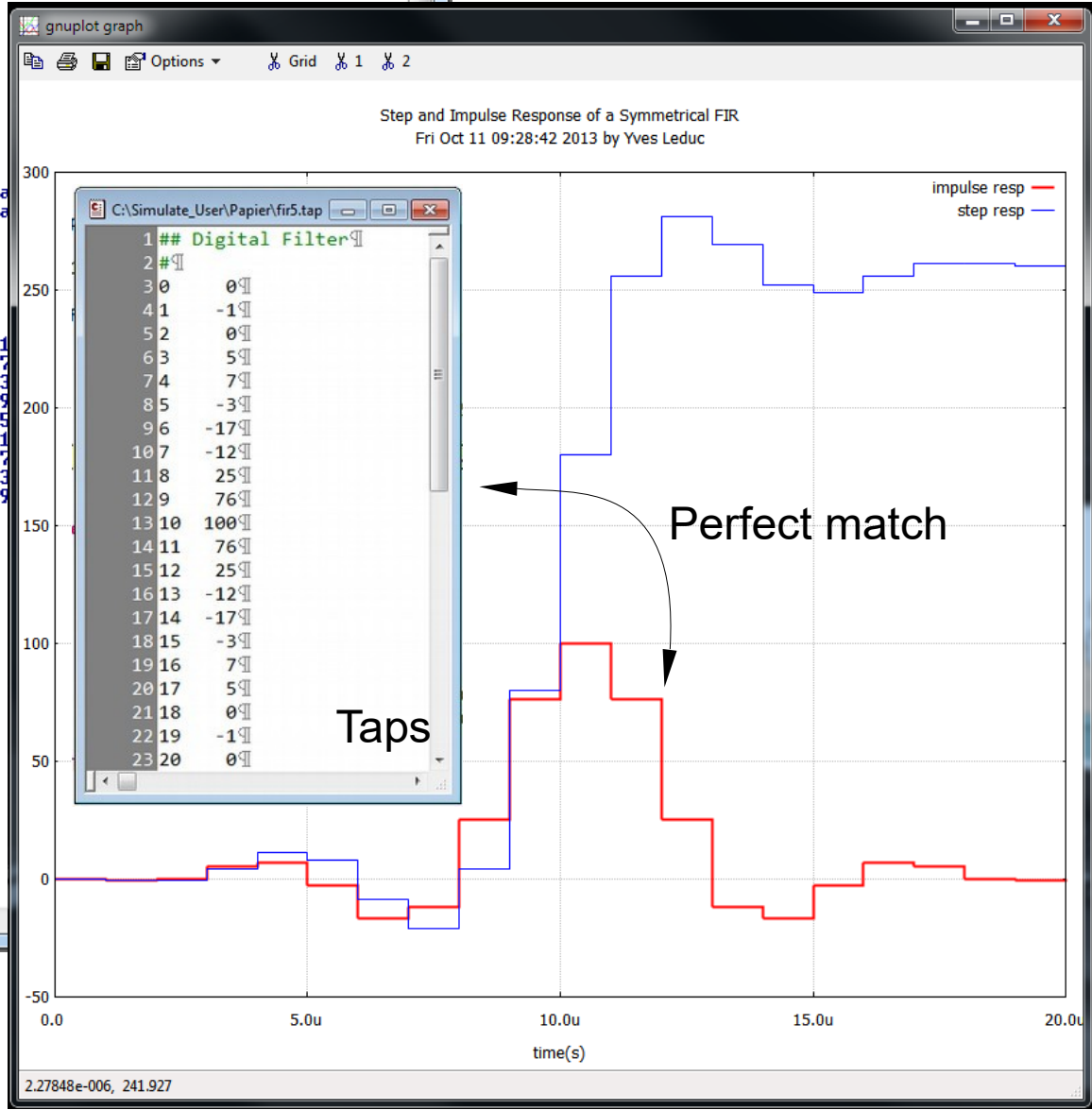
**** Input/Output : ****
**** -> response.out [ 0] ****

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

[resp]

```

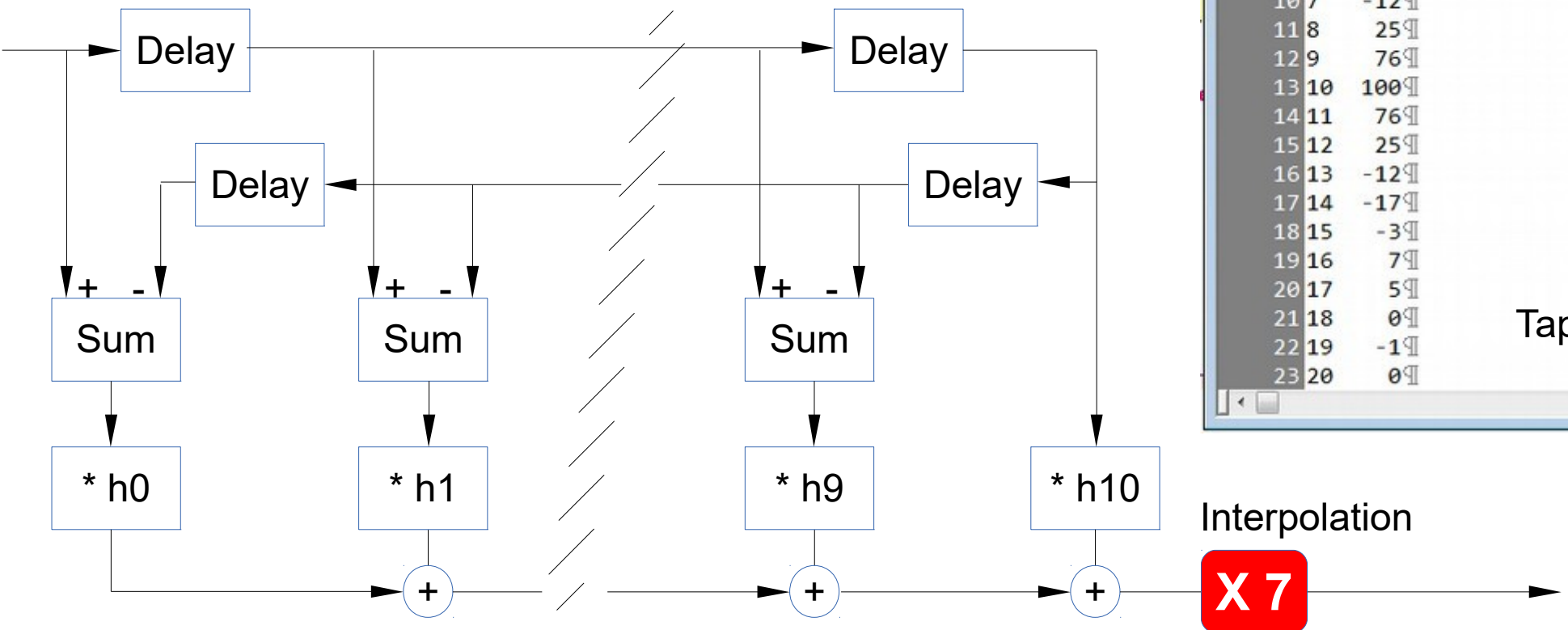
0.9 second





Transfer Function with Interpolation ?

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



```

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

Digital Folded FIR



```

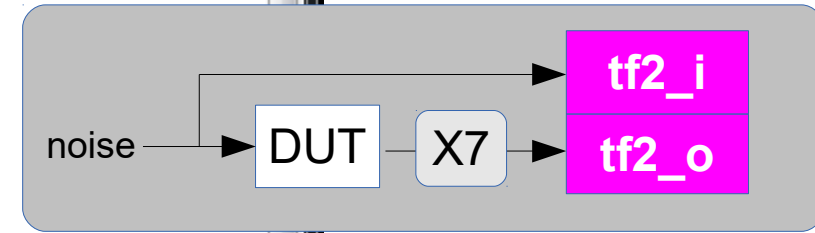
Crimson Editor - [C:\Simulate_Examples\NAPA_Running_Teaser\tf2b.nap]
File Edit Search View Document Project Tools Macros Window Help
tf2b.nap
1
2 title "Multirate transfer function of a symmetrical FIR"
3
4 header <napatool.hdr>
5
6 fs      1.0e6
7
8 directive REPEAT 10
9
10 ivar npts POWEROF2(16)
11
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 at frequency 'fs'



Multirate simulation
Multirate transfer function

— @ 1 MHz
— @ 7 MHz



```

Administrator : 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 : U3.01b 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

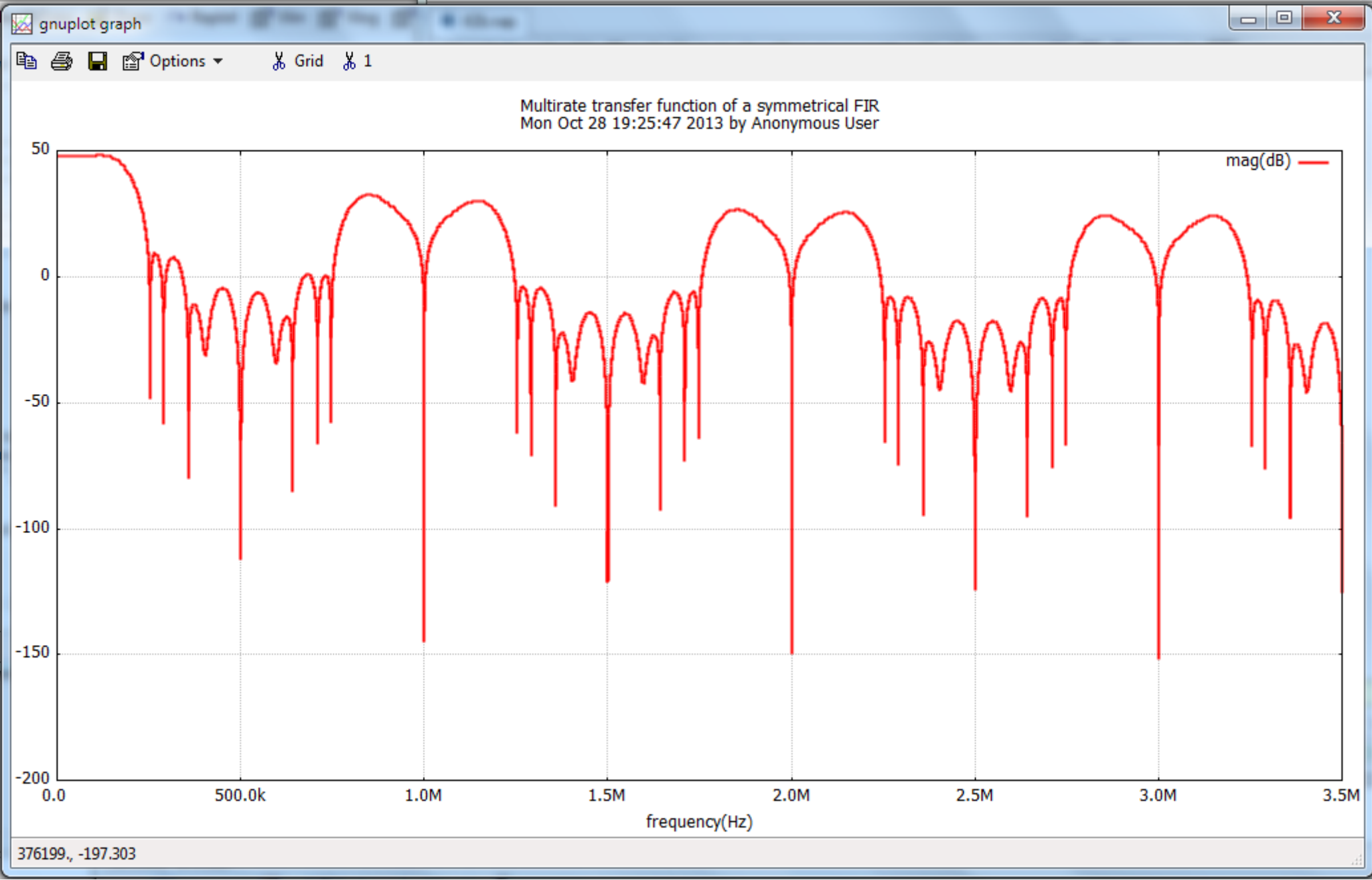
2x10 RMS of FFT's and 1 IFFT

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*

```

Crimson Editor - [C:\Simulate\NapaDos\Net\FILTER\mfb2.net]
File Edit Search View Document Project Tools Macros Window Help
tf2b.nap mfb2.net
1 cell interface $out $in $r1 $c2 $r3 $r4 $c5
2
3 *** 2ND ORDER MULTIPLE FEEDBACK CONTINUOUS TIME FILTER
4 *** (also known as "RAUCH Filter")
5 ***
6 ***
7 ***
8 ***
9 ***
10 ***
11 ***
12 ***
13 *** in C2/2 IDEAL > x out
14 ***
15 ***
16 ***
17 ***
18 ***
19 ***
20 ***
21 dvar $n2 0.0
22 dvar $n1 0.0
23 dvar $n0 1.0
24 dvar $d2 ($r1)*($c2) * ($r3)*($c5)
25 dvar $d1 ($r3)*($c5) + ($r1)*($c5) + ($r1)*($r3)*($c5)/($r4)
26 dvar $d0 ($r1)/($r4)
27 ganging $Coef[6] $n0..2 $d0..2
28
29 node $out duser ilt $Coef $in

```

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   :          U3.01b 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]

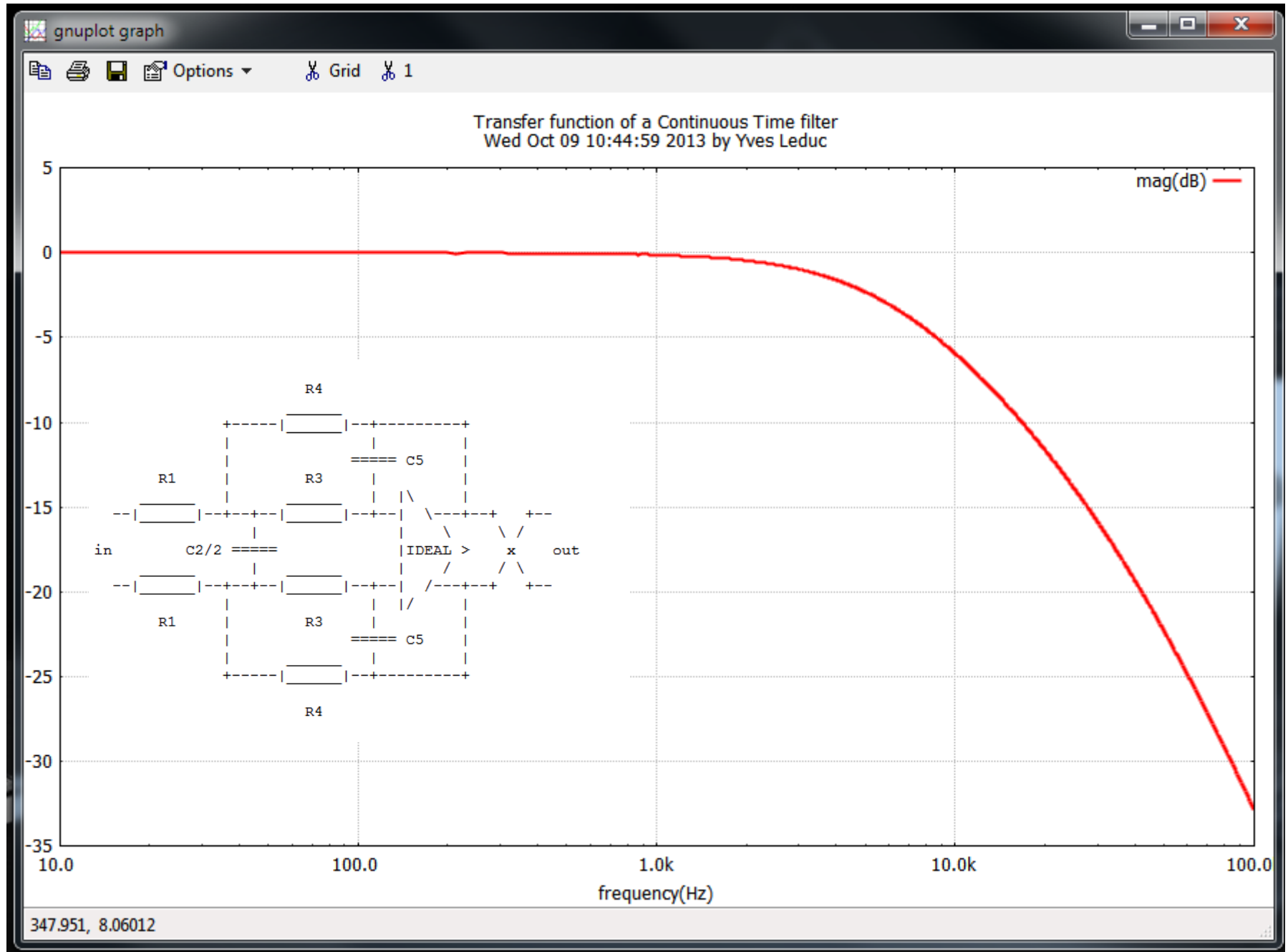
Press Enter to continue . . .

```

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
```

```
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 misspelled
nodes, or loops of nodes containing no delay or only delays
***** NAPA Compilation Error(s) *****
gmake: *** [geb.c] Error 1
Press Enter to continue . . .
```

Run Simulation





Type Mismatches, Syntax Errors are Caught at Compilation

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

```
C:\Simulate_User\Papier\cell.nap
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
```

Run Simulation

```
C:\Simulate_User\Papier\resonator.net
1 cell interface $out $off $saml $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 delaya $x1
10
11 node $out poly $saml $off $x1
12
13 init $x0 sin($phase)
14 init $x1 sin($phase - (_2pi*( $freq / FSL)))
15
```

```
Administrateur : NAPA Compile and Run: *** cell...
[cell] **** MAC Preprocessor Running ****
[cell] **** NAPA Compiler Running ****
NAPA Error: (node)
-> at line 8 of main netlist
-> at line 9 of file "resonator.net"
unknown node kind <delaya>
? <delay>
***** NAPA Compilation Error(s) *****
gmake: *** [cell.c] Error 1
Press Enter to continue . . .
```




```

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

[fft] **** MAC Preprocessor Running ****
[fft] **** NAPA Compiler Running ****
[fft] **** GCC Compiler Running ****
[fft] **** User's Simulator Running ****

**** 4 FFT

NAPA Run Time Warning: (directive)
-> at line 17 of main netlist
Directive <WINDOWING> is not registered

NAPA Run Time Warning: (debug)
-> at line 21 of main netlist
Debugging directive <FFT> has probably no effect

NAPA Tools Information: < fft[00] Process # 000 <- 262143
NAPA Tools Information: < fft[01] Process # 001 <- 524287
NAPA Tools Information: < fft[02] Process # 002 <- 786431
s Information: < fft[03] Process # 003 <- 1048575

Normal Termination ****

Random Seed [1] : 689830978 ****
Output Tag [0] : 982076502 ****

NAPA Compiler : U3.01 for Win64 ****
in Netlist : fft.tmp ****
Simulator Index : 1048576 ****
Simulation Time : 1.04858 s ****

Input/Output : ****
fft.log [ 0] ****
ffts.out [ 0] ****

...
directive WINDOWING ROSENFELD opwatch : H00:M00:S13.285 ****

...
debug FFT G File Ready : fft.log ****

...
Press Enter to continue . . .

```

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'

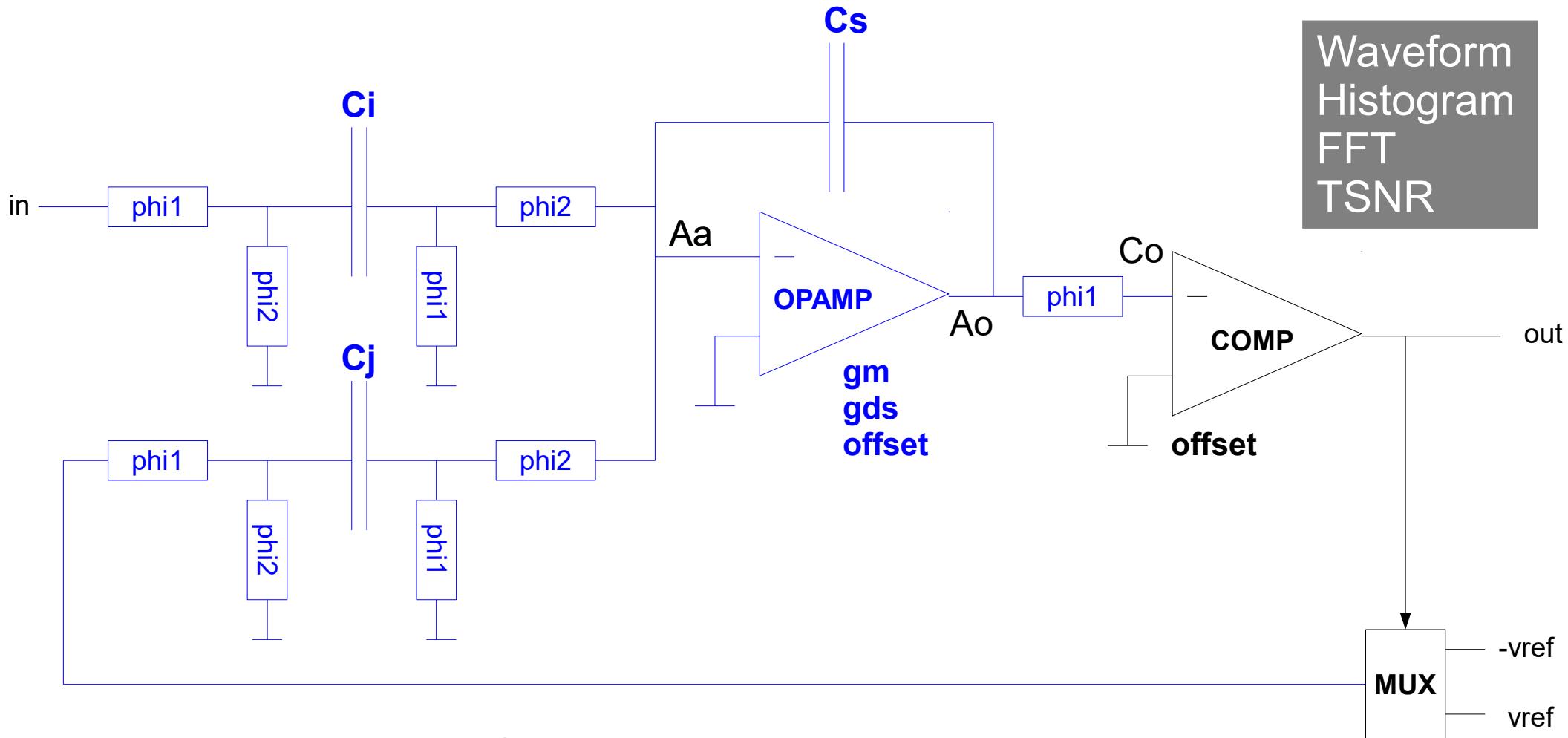
to model accurately and simulate linear circuits with on-demand percussion.

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 developed by Jacques Mequin 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      ] ) ;

```

SD1_core netlist
(MAXIMA)

```

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

GENERATE_MIMO ( ) ;

```

```

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)



```

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

[time_analyz] ***** MAC Preprocessor Running *****
[time_analyz] ***** NAPA Compiler Running *****
[time_analyz] ***** GCC Compiler Running *****
[time_analyz] ***** SARC Engine Linking *****
[time_analyz] ***** Ad Hoc Simulator Running *****

NAPA Ping Information : "duper_zarc()" from file "C:\Simulate\NapaDev\Hdr\User\zarc_hdr"
NAPA Ping Information : "SD1_core()" from file "C:\Simulate\NapaDev\Hdr\Nac\SD1\SD1_core_hdr"
NAPA Ping Information : "print_var()" from file "C:\Simulate\NapaDev\Hdr\Function\print_var_hdr"
NAPA Ping Information : "rand_normal()" from file "C:\Simulate\NapaDev\Hdr\Function\random_hdr"

***** Analyz in Time Domain [Analog 1st Order SD Modulator with SARC mode]

NAPA Debug Information: ( sampling)
Frl 01 -> 2.00000 MHz
Frl 11 -> 400.000 MHz
Frl 21 -> 2.00000 MHz
Frl 31 -> 1.00000 MHz
Frl 41 -> 2.00000 MHz

***** 320.0 u <- rd1_UoffA
***** 320.0 u <- rd1_UoffC
***** 500.0 f <- rd1_Ck
***** 1.54 k <- rd1_Again
***** 5.000 p <- rd1_Acz
***** 5.000 p <- rd1_Aci

NAPA Debug Information: ( zarc[ 0] )
[ SD1_core ] - schematic -
<<< STAGE A >>> <<< STAGE C >>>

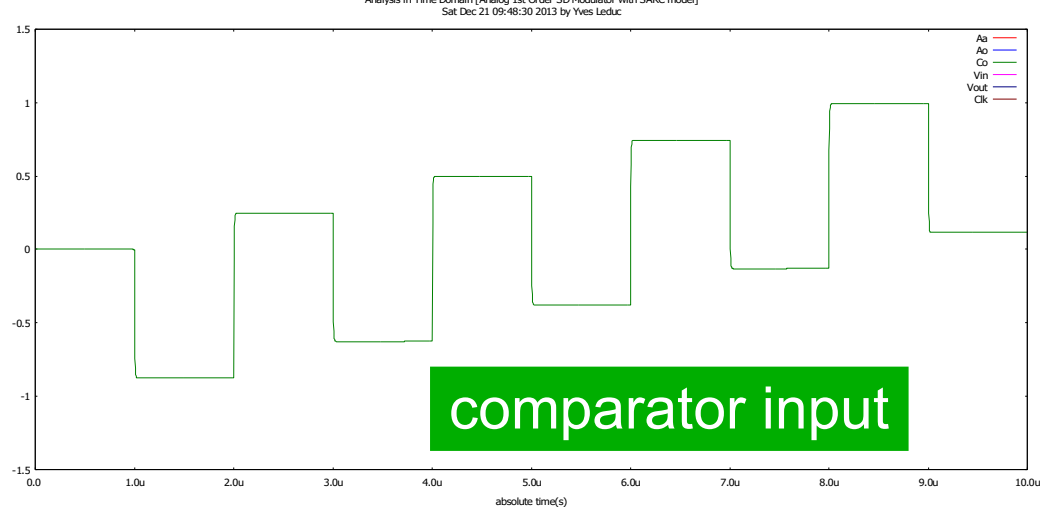
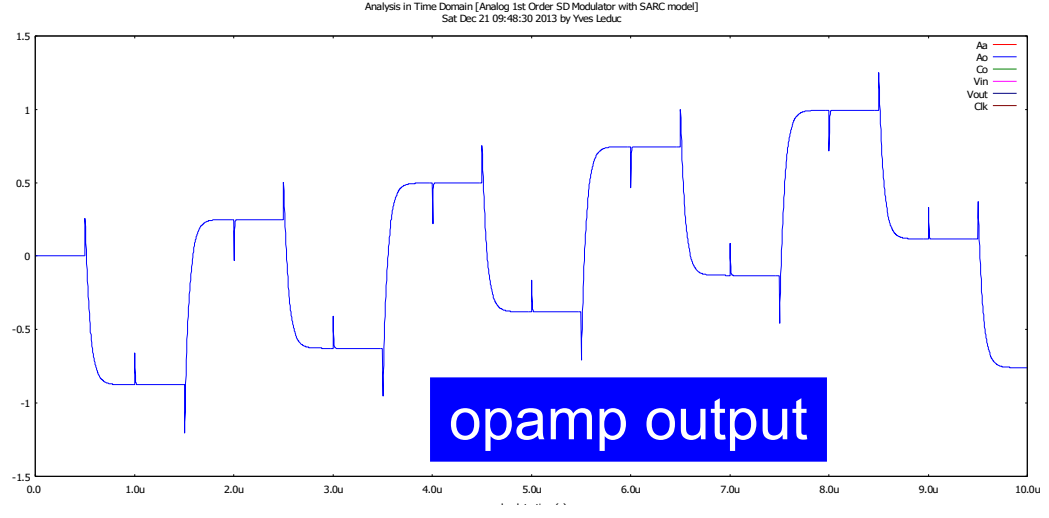
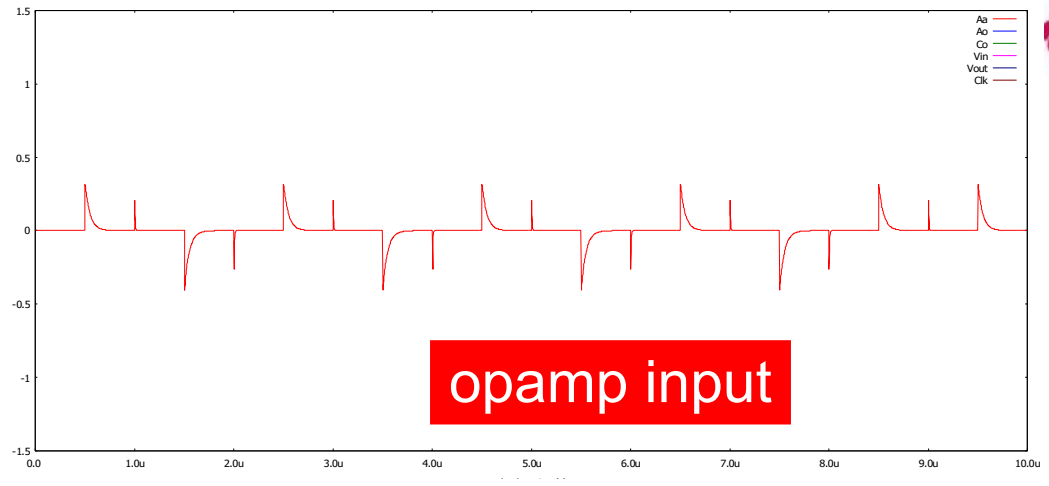
[ SD1_core ] - initialization -
[ SD1_core ] sampling at 400.000 MHz
[ SD1_core ] inputs { Aoff, Uf, Uin }
[ SD1_core ] outputs { U(Aa), U(Ca), U(Cc) }
[ SD1_core ] parameters { Aca, Aci, Acl, Acc, Acr, Cload, PHI1, PHI2, Ardr, Agm }
[ SD1_core ] parameter M1 rd1_Aca = 200.000 f -> Aca
[ SD1_core ] parameter M2 rd1_Aci = 5.00000 p -> Aci
[ SD1_core ] parameter M3 rd1_Acl = 5.00000 p -> Acl
[ SD1_core ] parameter M4 rd1_Acc = 400.000 f -> Acc
[ SD1_core ] parameter M5 rd1_Acr = 5.00000 p -> Acr
[ SD1_core ] parameter M6 rd1_Ck = 500.000 f -> Cload
[ SD1_core ] parameter M7 rd1_PHI1 = 0.00000 -> PHI1
[ SD1_core ] parameter M8 rd1_PHI2 = 0.00000 -> PHI2
[ SD1_core ] parameter M9 rd1_Ardr = 38.4615 M -> Ardr
[ SD1_core ] parameter M10 rd1_Agm = 299.997 u -> Agm

***** Normal Termination *****
***** Random Seed [I] : 693807855 *****
***** Output Tag [O] : 451714812 *****
***** NAPA Compiler : V3.02 for Win64 *****
***** Main Netlist : time_analyz.tsp *****
***** Simulator Loops : 4000 *****
***** Simulator Index : 20 *****
***** Simulation Time : 9.99750 us *****
***** Input/Output : *****
***** -> time.out : [ 0] *****

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

[time_analyz]
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)


```

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

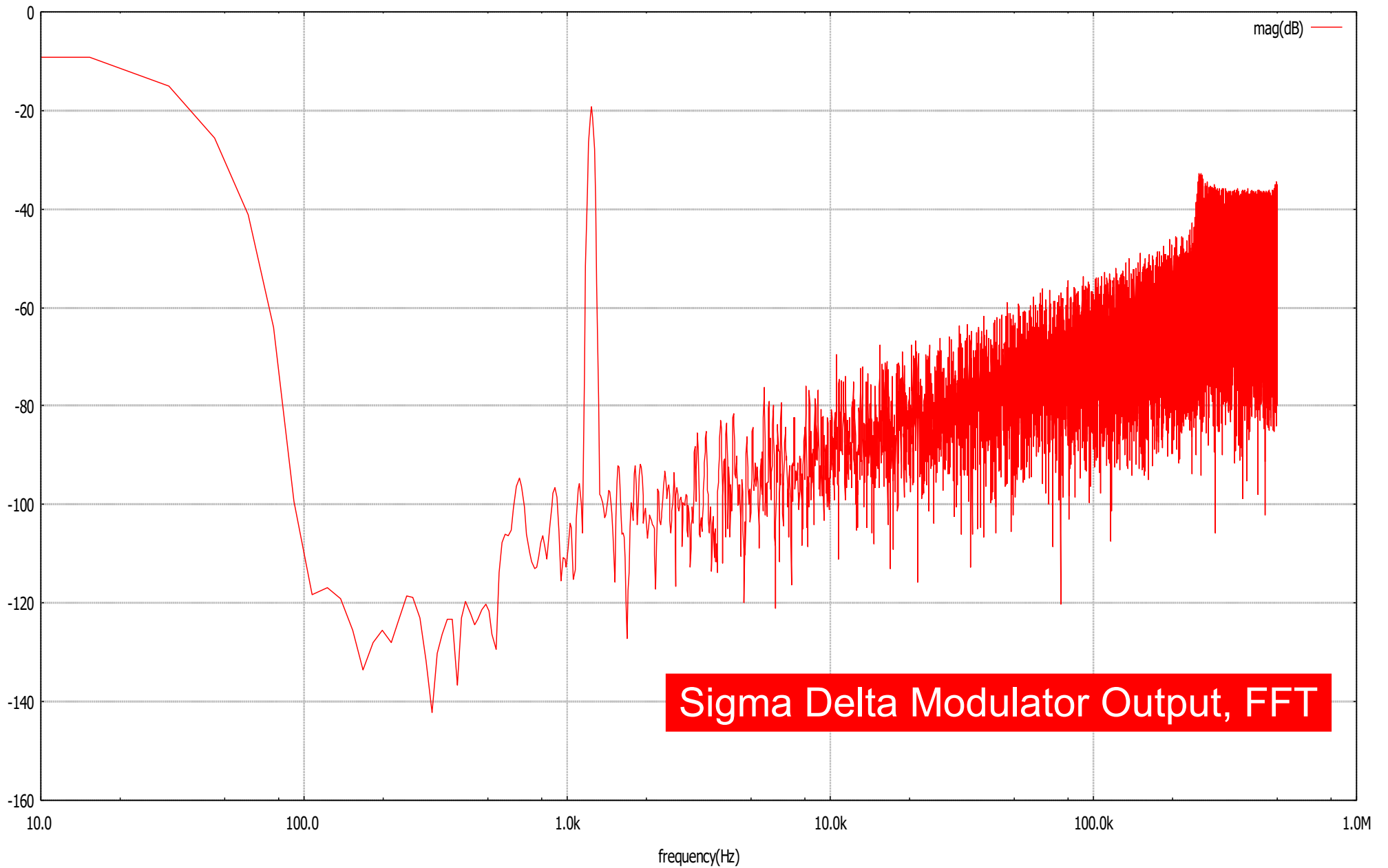
debug   SAMPLING  SARC_INFO
ping

```

NAPA Simulation (FFT)



FFT Analysis [Analog 1st Order SD Modulator with SARC model]
Yves Leduc



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

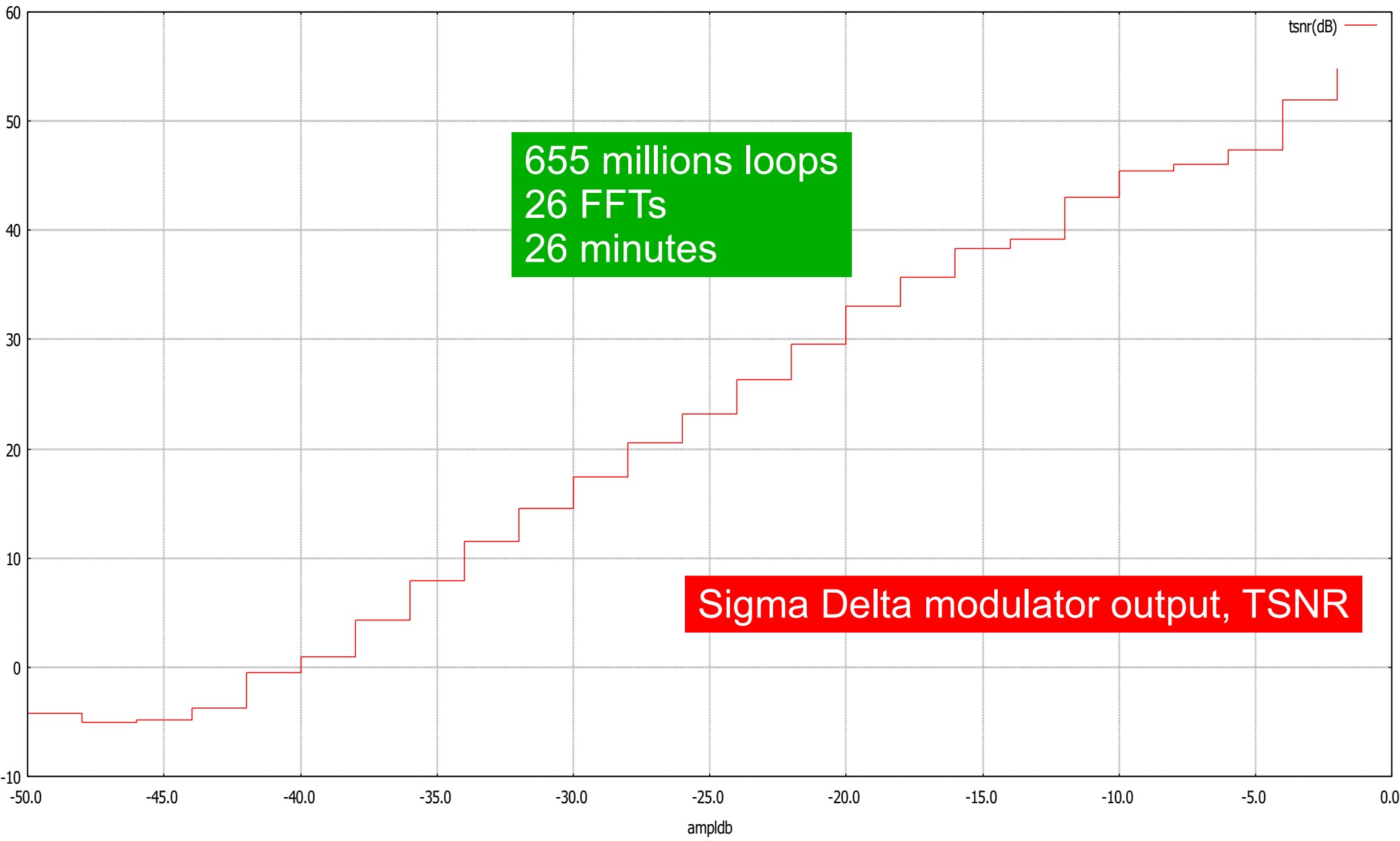
debug    SAMPLING  SARC_INFO
ping

```

NAPA Simulation TSNR



TSNR Analysis [Analog 1st Order SD Modulator with SARC model]
Yves Leduc



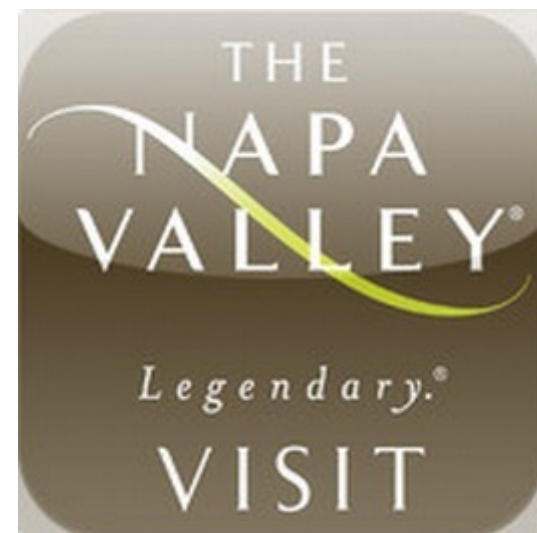


NAPA Has MANY Other Nice Features.

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

NAPA Instructions



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

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	Butterworth\LP8.net	Filter\1p1z.net	Filter\filt11.net
Activation\chirp2.net	Butterworth\LP9.net	Filter\2p2z.net	Filter\filt22.net
Activation\chirp2_g.net	CDS\m1.net	ISD\m.net	Filter\filt33.net
Activation\chirp_g.net	CDS\m11.net	Latch\DFF1.net	Filter\lp1.net
Activation\pulse.net	CDS\m2.net	Latch\DFF2.net	Filter\lprc1.net
Activation\resonator.net	Cell\1.net	Latch\SR.net	Filter\lprc2.net
Activation\sigmoid.net	Cell\2.net	Logic\adder.net	Filter\ma.net
Activation\step.net	Cell\3.net	Logic\adder2b.net	Filter\mfb2.net
Activation\three_phase.net	Cell\4.net	Logic\adder3b.net	Filter\nyq365.tap
Activation\triangle.net	Cell\5.net	Logic\add_evn.net	Filter\nyquist.net
Adder\i1.net	Cell\6.net	Logic\add_gen.net	Integrator1\d1.net
Adder\i1_a.net	Cell\d1.net	Logic\add_odd.net	Integrator1\d1i1.net
Adder\i2.net	Cell\d2.net	Logic\carry.net	Integrator1\d1i1_a.net
Adder\i2_a.net	Cell\d3.net	Logic\half_adder.net	Integrator1\d1i1_ac.net
Adder\i3.net	Cell\d4.net	Logic\sum.net	Integrator1\d1_a.net
Adder\i3_a.net	Cell\d5.net	Measure\energy.net	Integrator1\d1_ac.net
Adder\i4.net	Cell\d6.net	Measure\freq.net	Integrator1\d1_r.net
Adder\i4_a.net	Comparator\1_h.net	Measure\slope.net	Integrator1\d2.net
ASD\m1.net	Comparator\2.net	Misc\adder.net	Integrator1\d2_a.net
ASD\m11.net	Comparator\2_a.net	Misc\channel.net	Integrator1\d2_ac.net
ASD\m2.net	Comparator\2_h.net	Misc\cint.net	Integrator1\d2_r.net
ASD\m211.net	Comparator\3.net	Misc\cintd.net	Integrator1\d3.net
ASD\m21_va.net	Comparator\d1_h.net	Misc\gdelay.net	Integrator1\d3_a.net
ASD\m22_va.net	Comparator\d2.net	Misc\GTswitch.net	Integrator1\d3_ac.net
ASD\m22_vc.net	Comparator\d2_a.net	Misc\intdz.net	Integrator1\d3_r.net
ASD\m2ff.net	Comparator\d2_h.net	Misc\intm.net	Integrator1\d4.net
Bessel\LP1.net	Comparator\d3.net	Misc\intz.net	Integrator1\d4_a.net
Bessel\LP2.net	Counter\bincount.net	Misc\rv1.net	Integrator1\d4_ac.net
Bessel\LP3.net	Counter\counter.net	Modulation\am.net	Integrator1\i1.net
Bessel\LP4.net	Counter\counter2.net	Modulation\am2.net	Integrator1\i1_a.net
Bessel\LP5.net	Counter\edgecount.net	Modulation\fm.net	Integrator1\i1_ac.net
Bessel\LP6.net	Counter\modcnt.net	Modulation\fm2.net	Integrator1\i1_ct.net
Bessel\LP7.net	Counter\modcnt2.net	Modulation\phm.net	Integrator1\i2.net
Bessel\LP8.net	Counter\modcntr.net	Noise\jitter.net	Integrator1\i2_a.net
Biquad\Martin_Sedra.net	Counter\modcntr2.net	Noise\ktoverc.net	Integrator1\i2_ac.net
Biquad\SWC.net	DC_removal\dcr0.net	Noise\pink.net	Integrator1\i2_ct.net
Biquad\SWC1.net	DC_removal\dcr1.net	Noise\rclock.net	Integrator1\i3.net
Biquad\SWC2.net	DC_removal\dcr2.net	Noise\red.net	Integrator1\i3_a.net
Biquad\z.net	DC_removal\dcr4.net	Processor\bbh.net	Integrator1\i3_ac.net
Butterfly4.net	Detect\d.net	Processor\bbh4.net	Integrator1\i3_ct.net
Butterfly4b.net	Detect\u.net	PWL\d.net	Integrator1\i4.net
Butterfly4b_hl.net	Detect\ud.net	PWL\i.net	Integrator1\i4_a.net
Butterworth\LP1.net	Differentiator\1.net	PWM\1.net	Integrator1\i4_ac.net
Butterworth\LP10.net	DSD\m1.net	PWM\2.net	Integrator1\i4_ct.net
Butterworth\LP2.net	DSD\m11.net	PWM\3.net	Integrator2\1.net
Butterworth\LP3.net	DSD\m2.net	PWM\4.net	Integrator2\1_a.net
Butterworth\LP4.net	DWA\16_d.net	Range\max.net	Integrator2\1_ac.net
Butterworth\LP5.net	DWA\4_d.net	Range\max_z.net	Integrator2\2.net
Butterworth\LP6.net	DWA\8.net	Range\min.net	Integrator2\2_a.net
Butterworth\LP7.net	DWA\8_d.net	Range\min_z.net	Integrator2\2_ac.net
		Range\width.net	Integrator2\3.net
		Sequence\a.net	Integrator2\3_a.net



Integrator2\3_ac.net
Integrator2\4.net
Integrator2\4_a.net
Integrator2\4_ac.net
Sequence\lag.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



acosh
 arithmetic_mean
 arithmetic_geometric_mean
 asinh
 atanh
 bessell_i
 bessell_j
 bessell_k
 bessell_y
 c2f
 c2k
 centroidal_mean
 choice_between_i
 choice_between_d
 choice_between_s
 coherent
 coherent_lindomain
 coherent_linsweep
 coherent_logdomain
 coherent_log sweep
 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_equilikeley
 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_dp11
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_histosp
itool_histoal
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...***

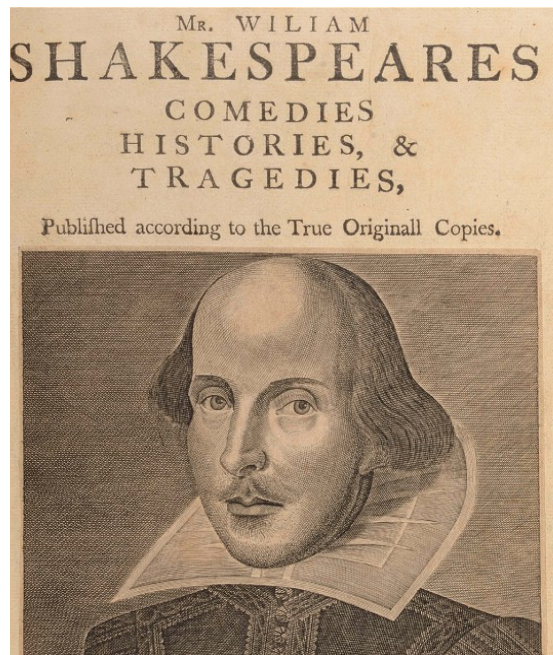
Questions ?





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. iiii



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 μ CU: 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

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) HMT35156CFR8C-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

Drives

Interface	Model
SATA 3 Gb/s	WDC WD5000BPKT-80PK4T0 [500 GB, 16MB]
SATA 1.5 Gb/s	MATSHITADVD-RAM U38A2ASW [DVD+R DL]

FYI, my computer