ADS User Group Meeting

Time Domain electrical simulation using equivalent digital wave networks in ADS

Rome May 13th, 2009

Flavio Maggioni
(Nokia Siemens Networks)
E-mail: flavio.maggioni@nsn.com

Typical time-domain electrical simulation approach

Spice-like engines:

- Solution of integral equations (iterative methods)
- Adaptive time step
- Simulation time grows exponentially with network complexity
- Convergence could be a problem

In addition:

circuital blocks represented by S-Parameters are now commonly used in time-domain simulations. Two possible approaches:

FFT/ IFFT

Time-domain S-parameters ↔ Frequency-domain S-parameters

Direct Time-domain convolution (low efficiency for Spice-like engines):

- Adaptive time-step
- Very high time consuming algorithm

Equivalent Electrical networks synthesis

- Zero-pole fitting
- Causability and stability verification required
- Critic low-frequency fitting

An alternative approach

In this presentation:

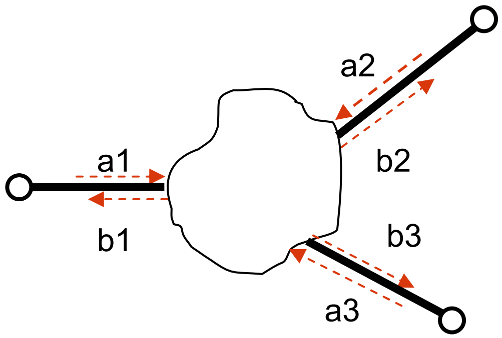
- Equivalent digital wave equivalents are introduced to represent electrical networks [1].
- Direct convolution in time domain is presented as a convenient approach in ADS for typical digital signal integrity analysis.
- ADS Ptolemy engine is used as alternative to classic engines for general time-domain simulation (ADS-Matlab® demonstrator).

This methodology, used stand-alone or in co-simulation with standard transient analysis introduces new possibility in ADS.

Wave digital network

- Voltages and currents along a wire can be represented by a combination of forward and reverse voltage and current propagation waves*

- A network with concentrated and distributed elements can be represented by a cascade of n-ports S-parameters blocks. “a” are the incident waves, “b” are the reflected waves



- For each element: b = S * a where a, b are vectors and S is the scattering matrix in time domain

  Convolution integral

- Each S-parameter element can be expressed in *s* plane and then converted to z plane applying the z transform (time discrete).

---

* Solution of the **telegrapher's equations**

Digital wave representation

At each port of the element:

$$v = a + b \qquad i = \frac{a - b}{Z_{REF}}$$

Where:

a = incident voltage wave

b = reflected voltage wave

v = physical voltage at port

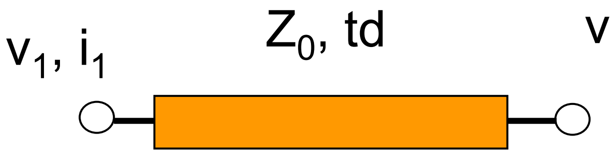
i = physical current at port

Zref = reference impedance of the port

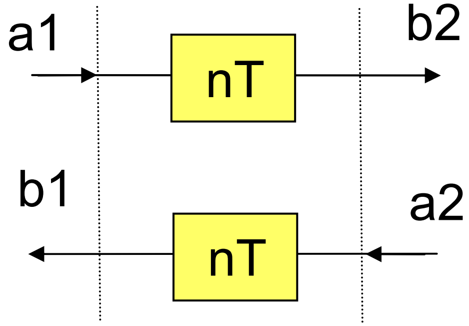
T= simulation time-step (fixed)

Td = line delay = nT

Example: ideal transmission line



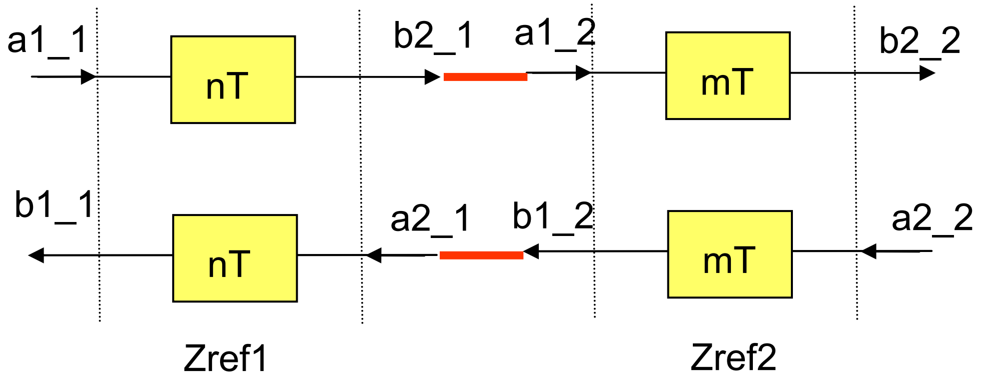
Electrical element



Digital wave equivalent

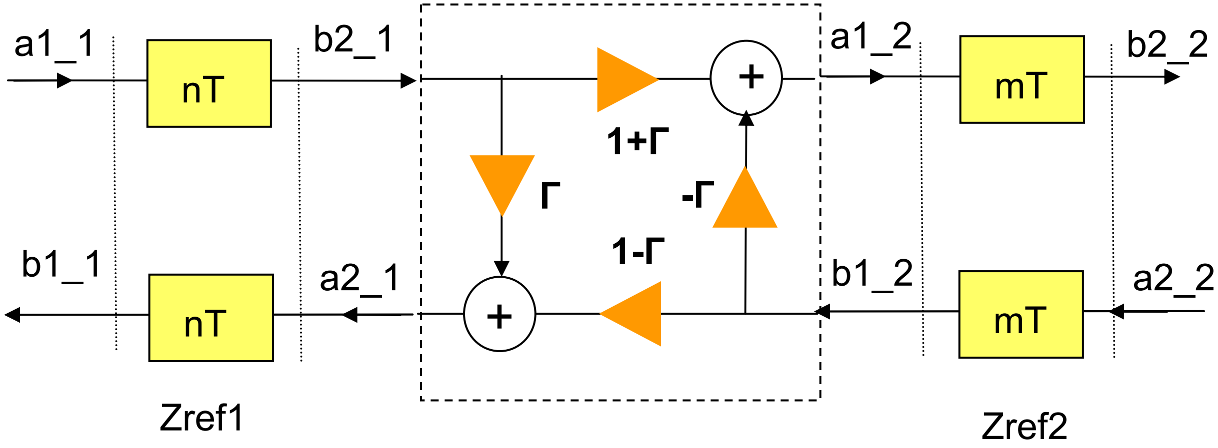
Cascading elements (two-port adapter)

Direct connection if **Zref1=Zref2**



Two transmission lines example: if the reference impedance are the same, the output waves of one element are exactly the input waves of the next one.

An “adapter” is required if **Zref1≠Zref2**



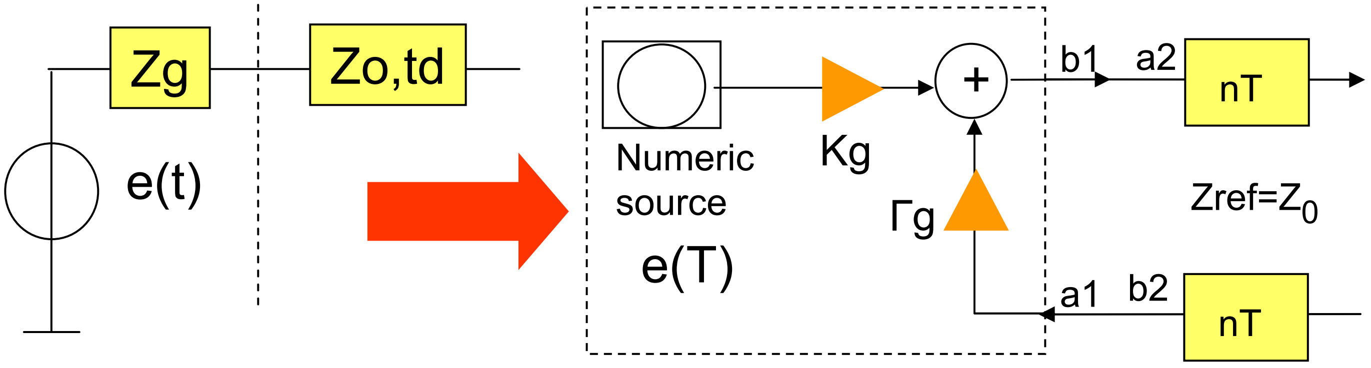
Two transmission lines example: if the reference impedances are different, some of the incident energy is scattered back.

Γ (Gamma) is the reflection coefficient:

$$\Gamma = \frac{Zref\,2 - Zref\,1}{Zref\,2 + Zref\,1}$$

Linear generators and loads

Linear generator



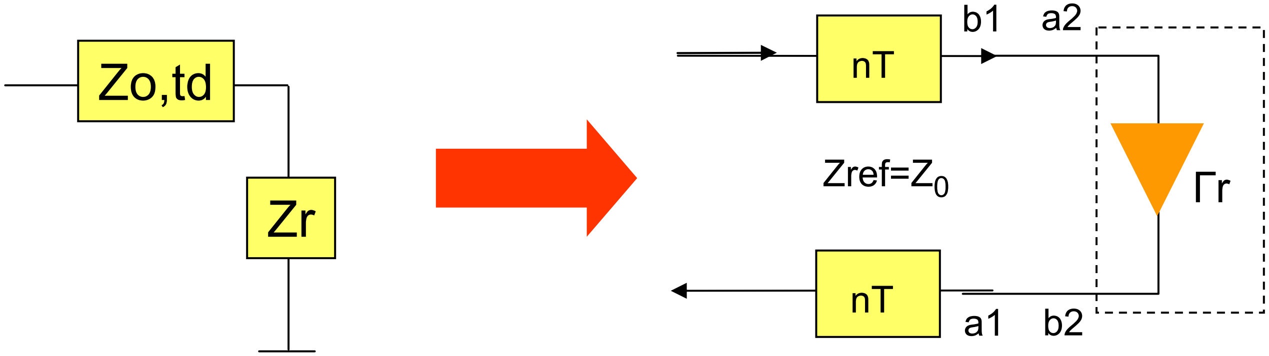
Where:

$$Kg = \frac{Z_0}{Zg+Z_0}$$

$$\Gamma g = \frac{Zg - Z_0}{Zg+Z_0}$$

Numeric source = e(t) samples

Linear load

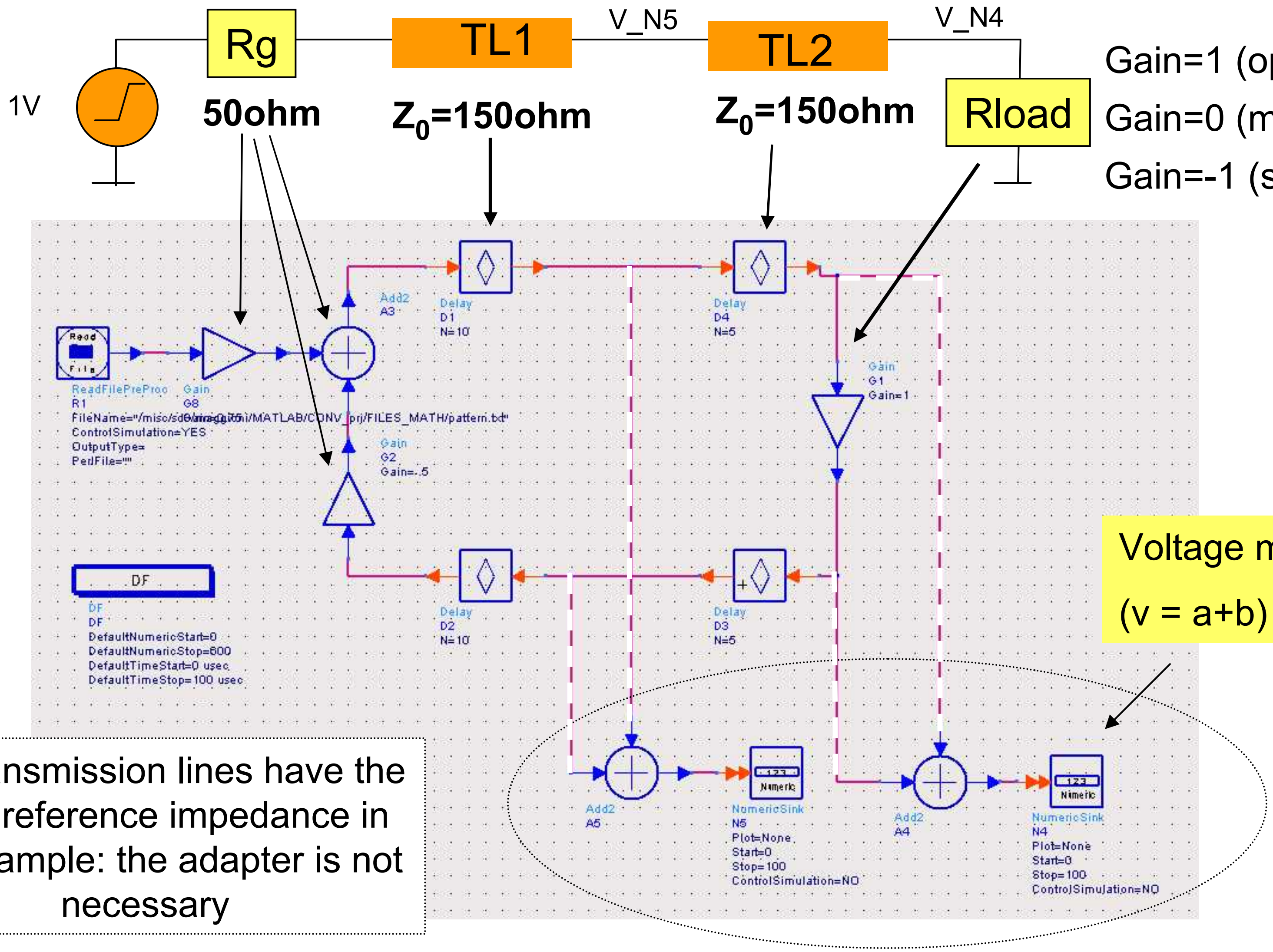


Where:

$$\Gamma r = \frac{Zr - Z_0}{Zr+Z_0}$$

Open: $\Gamma r = 1$
Matched: $\Gamma r = 0$
Short: $\Gamma r = -1$

ADS implementation: two cascaded transmission lines



ReadFilePreProc
R1
FileName="/misc/sd0/[illegible]/MATLAB/CONV_prj/FILES_MATH/pattern.txt"
ControlSimulation=YES
OutputType=
PerlFile=""

Gain
G8

Add2
A3

Gain
G2
Gain=-.5

Delay
D1
N=10

Delay
D4
N=5

Gain
G1
Gain=1

Delay
D2
N=10

Delay
D3
N=5

DF
DF
DefaultNumericStart=0
DefaultNumericStop=600
DefaultTimeStart=0 usec
DefaultTimeStop=100 usec

Add2
A5

NumericSink
N5
Plot=None
Start=0
Stop=100
ControlSimulation=NO

Add2
A4

NumericSink
N4
Plot=None
Start=0
Stop=100
ControlSimulation=NO

Gain=1 (open)
Gain=0 (matched)
Gain=-1 (short)

Voltage monitors

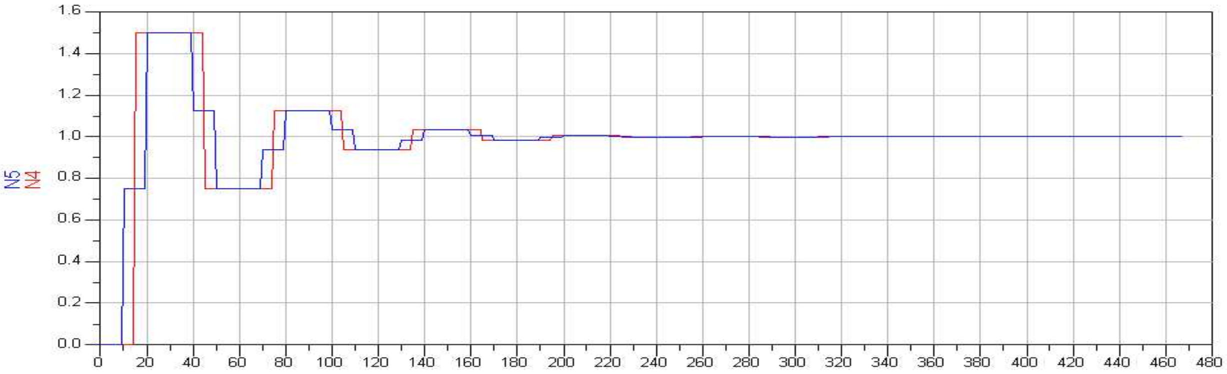
(v = a+b)

The transmission lines have the same reference impedance in this example: the adapter is not necessary

Step response simulation

Open circuit

Gain = 1

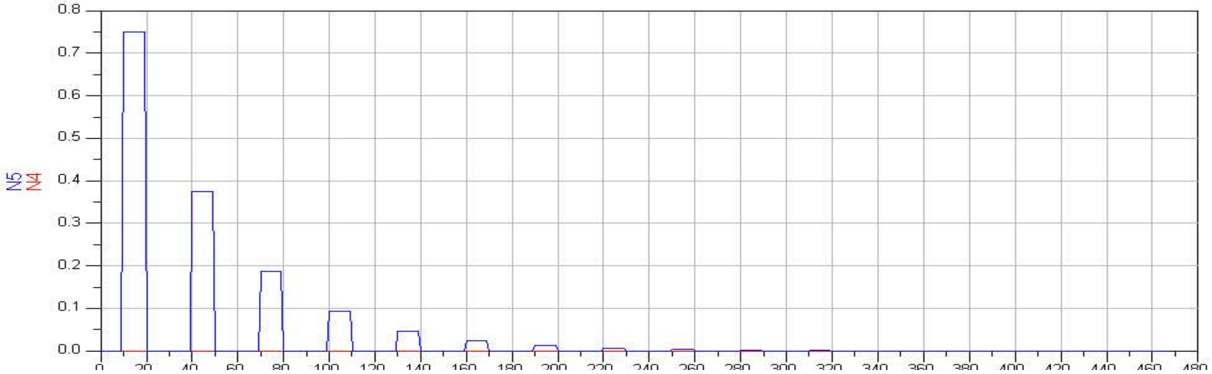


Red: load

Blue: intermediate point

Short circuit

Gain = -1

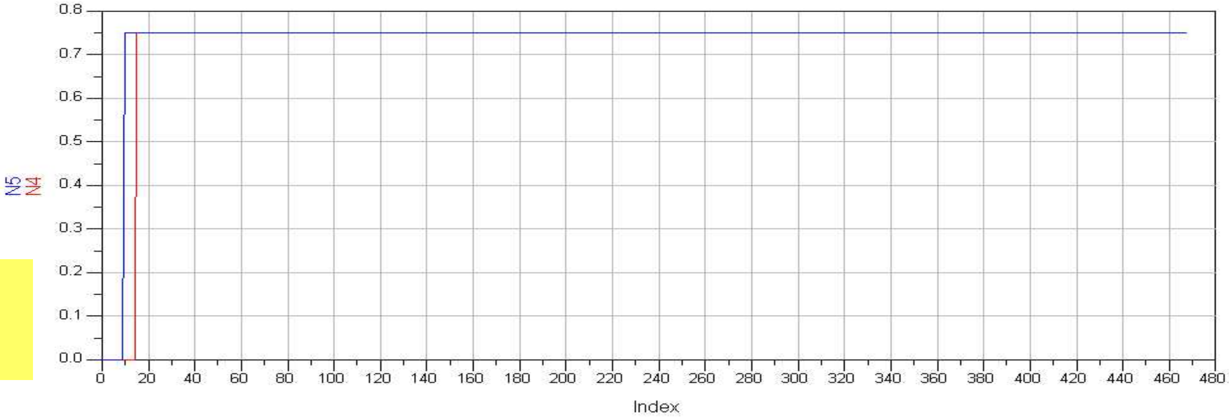


Red: load

Blue: intermediate point

Matched impedance

Gain = 0



Red: load

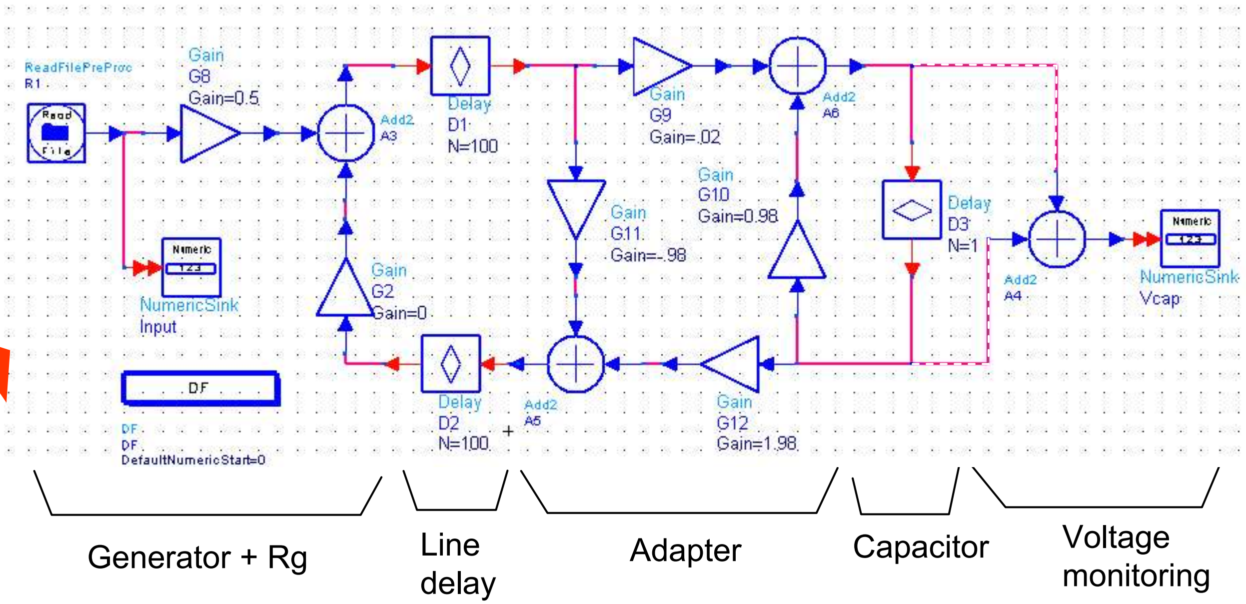
Blue: intermediate point

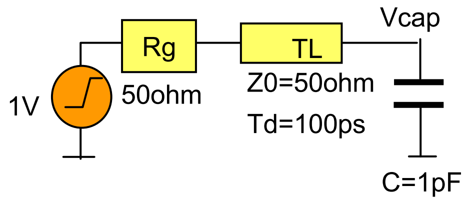
500 sample (=500ps).

Simulation time << 1”

Other elements: 1-port capacitor (connected to ground)

The capacitor is represented by a unit-delay element. In this case the reference impedance (to be used for the adapter parameters calculations) is: Zref=Tstep/2C





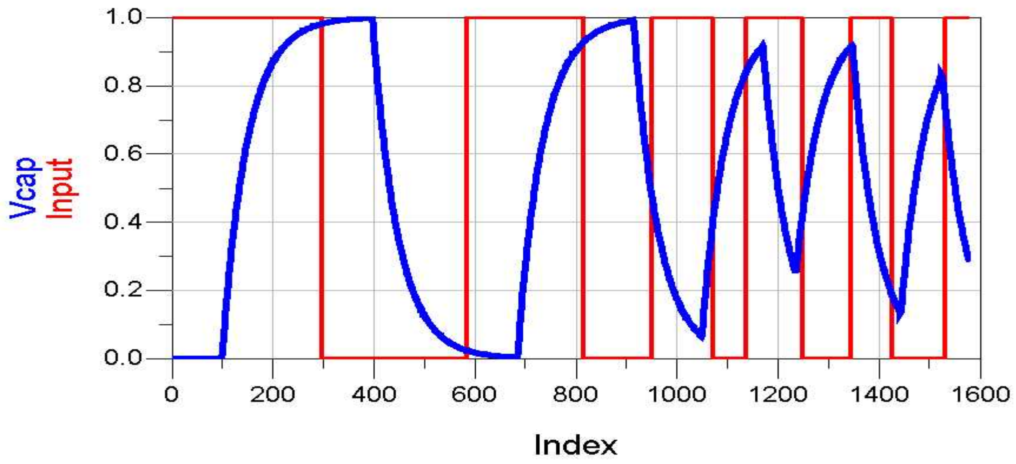
Tstep = 1ps

Zref capacitor = Tstep/2C = 0.5 ohm

Γ = (0.5-50)/(0.5+50) ~ -0.98

Pattern:1111000011100110110011010

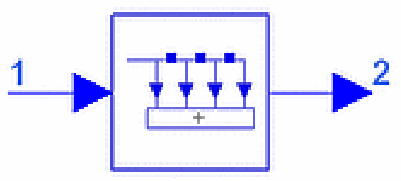
(1 bit width= 75 Time step)

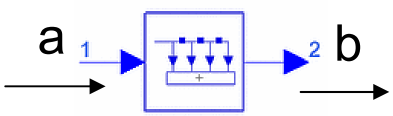


Time Domain Convolution in ADS (FIR filters)

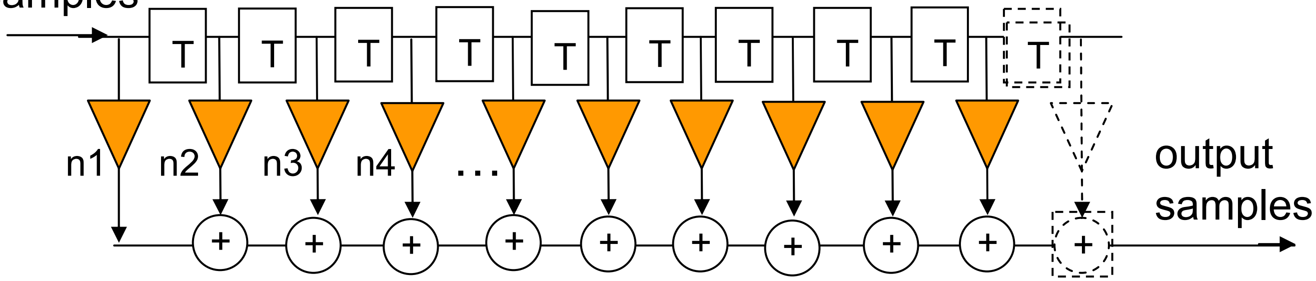
Cascaded S-parameters elements with generic transfer function in time-domain simulation requires the application of the convolution integral.

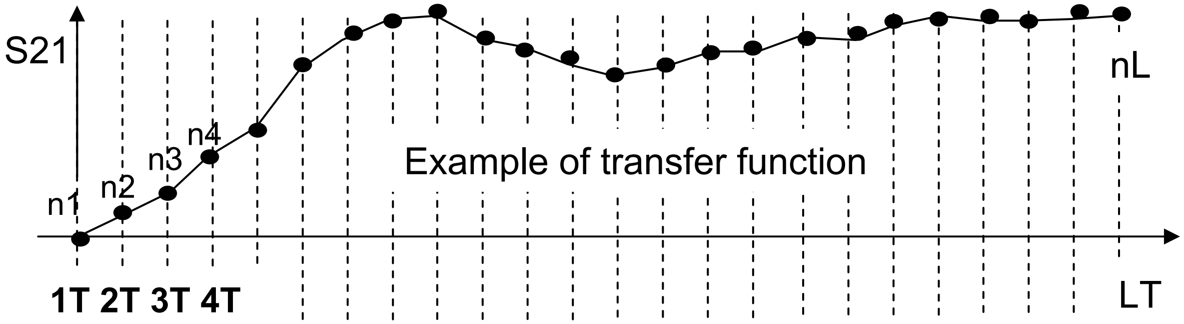
In ADS, DSP networks can use Finite Impulse Response (FIR) filters





b/a = S21





Very time-consuming algorithm. At any time step:

1 shift operation

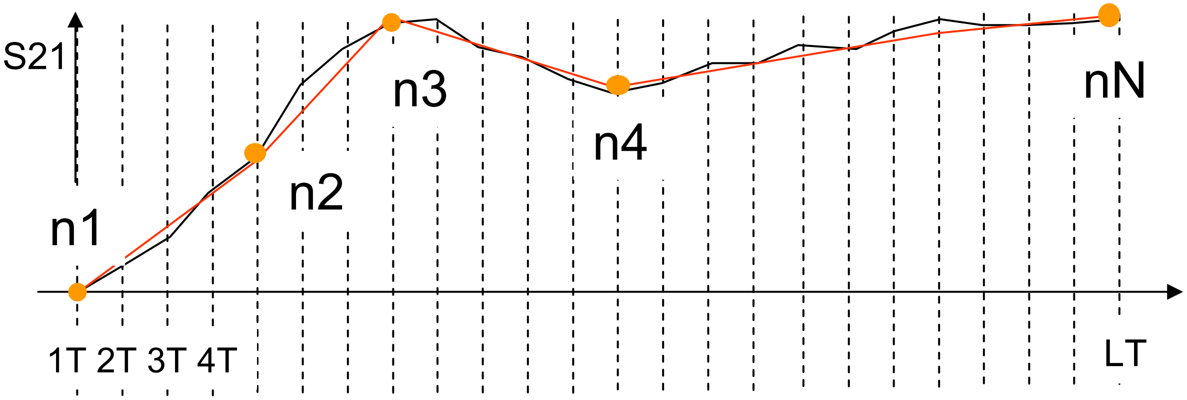
L additions

L+1 multiplication

S21 lenght = L samples **number of taps = N = L**

FIR Filter after PWL approximation

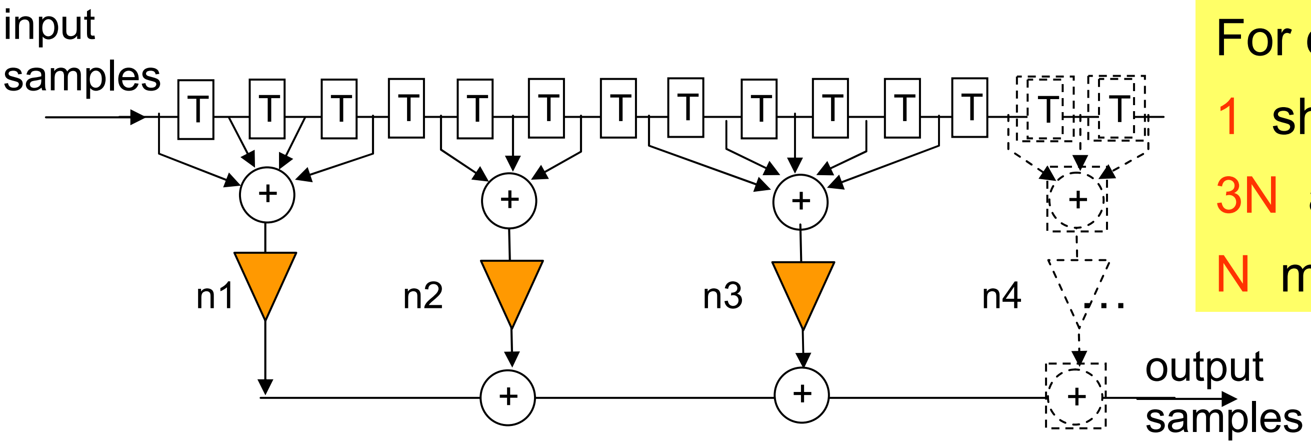
In some applications (i.e. digital high-speed signal integrity analysis), the transfer functions are smoothed respect to time step (i.e.: long queues due to losses). In this case, a Piece-Wise Linear (PWL) approximations can be used.



S21 lenght = L samples

number of PWL taps = N << L

(i.e.: N=20, L=600)



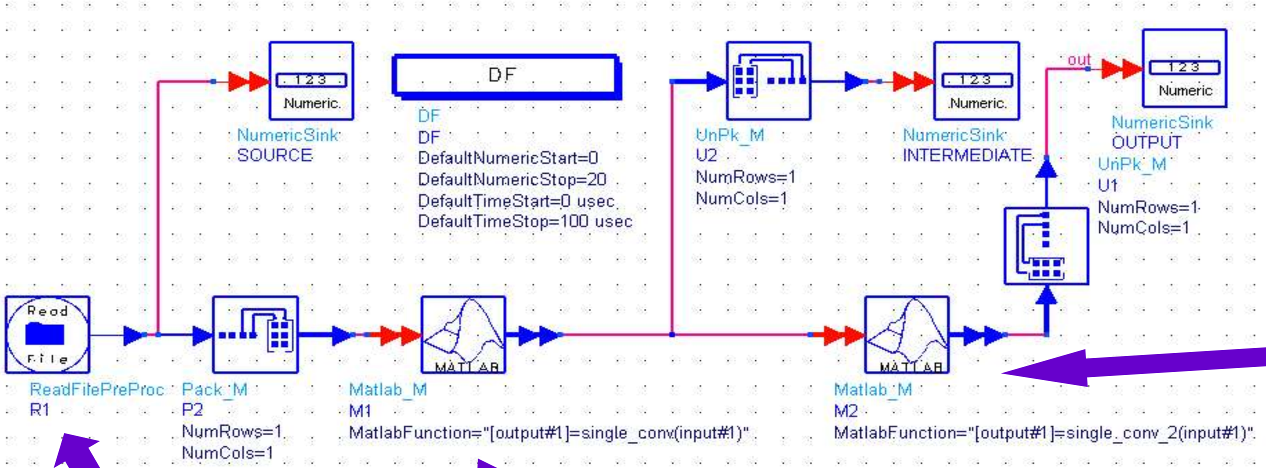
For each input sample:

- 1 shift operation
- 3N additions/subtractions
- N multiplication

The simulation speed increases of a factor L/N

Fast convolution with PWL FIR (MATLAB® implementation)

Example of a transfer function (Z1) cascaded with a low-pass filter (Z2)



ReadFilePreProc
R1

Pack_M
P2
NumRows=1
NumCols=1

NumericSink
SOURCE

DF
DF
DefaultNumericStart=0
DefaultNumericStop=20
DefaultTimeStart=0 usec
DefaultTimeStop=100 usec

Matlab_M
M1
MatlabFunction="[output#1]=single_conv(input#1)"

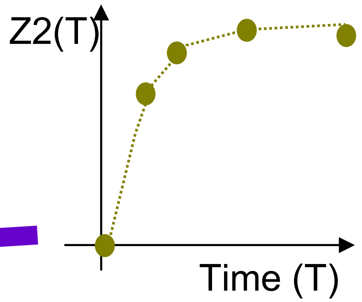
UnPk_M
U2
NumRows=1
NumCols=1

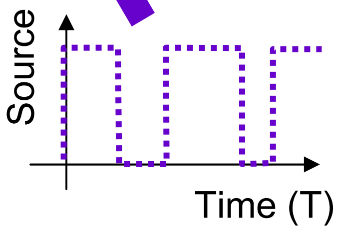
NumericSink
INTERMEDIATE

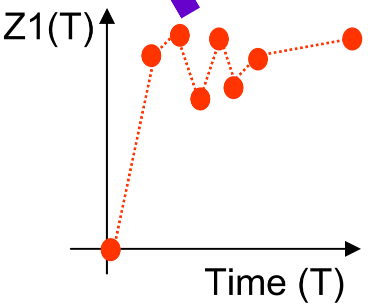
NumericSink
OUTPUT

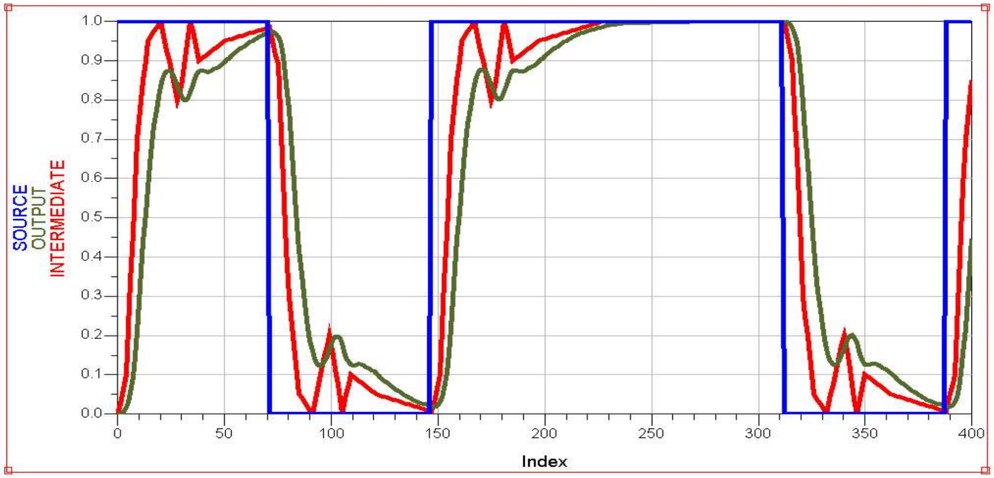
UnPk_M
U1
NumRows=1
NumCols=1

Matlab_M
M2
MatlabFunction="[output#1]=single_conv_2(input#1)"

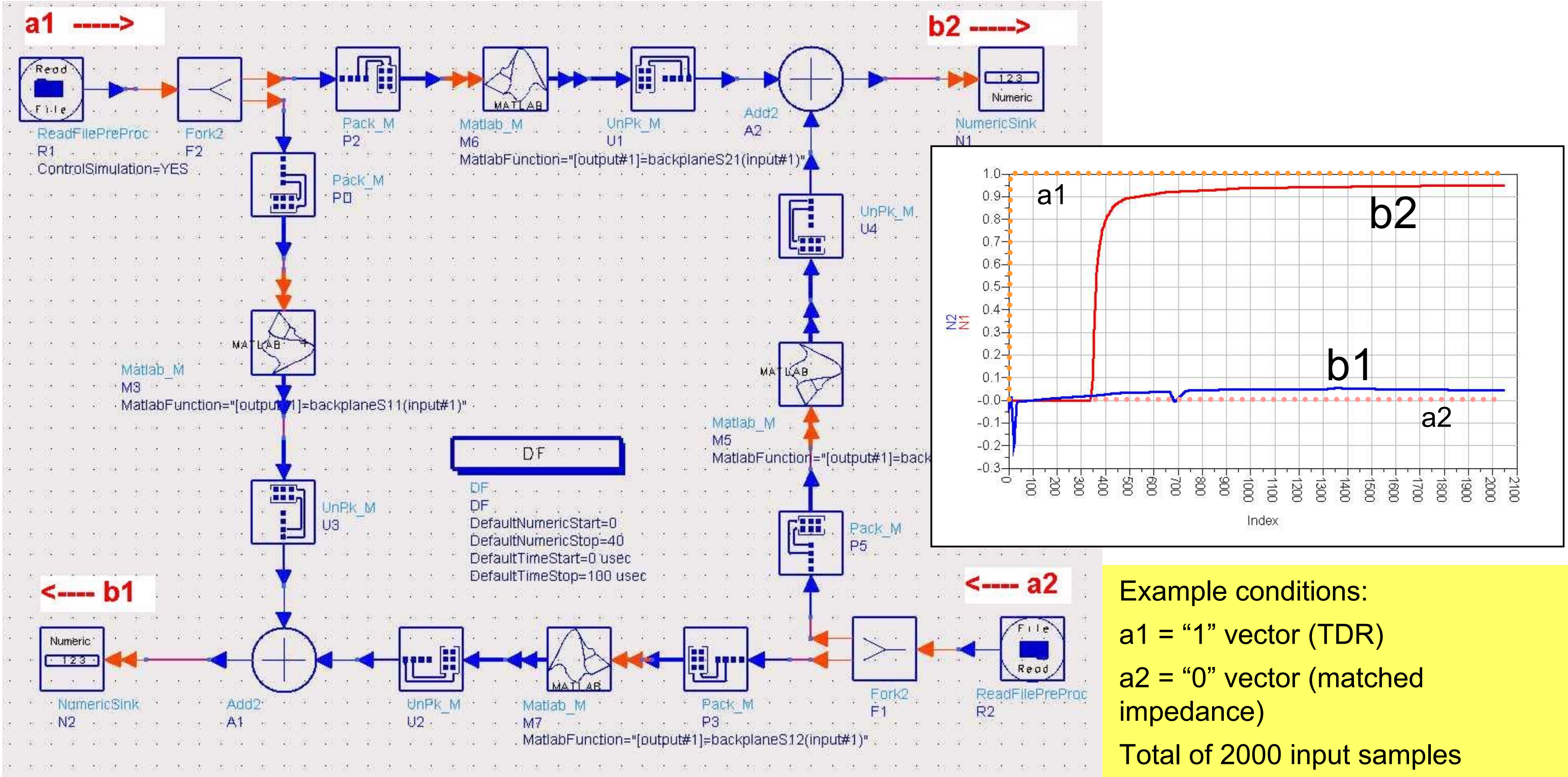








Lossy transmission line example



ControlSimulation=YES

MatlabFunction="[output#1]=backplaneS21(input#1)"

MatlabFunction="[output#1]=backplaneS11(input#1)"

MatlabFunction="[output#1]=back

DefaultNumericStart=0
DefaultNumericStop=40
DefaultTimeStart=0 usec
DefaultTimeStop=100 usec

MatlabFunction="[output#1]=backplaneS12(input#1)"

Example conditions:

a1 = “1” vector (TDR)

a2 = “0” vector (matched impedance)

Total of 2000 input samples (1sample= 1ps)

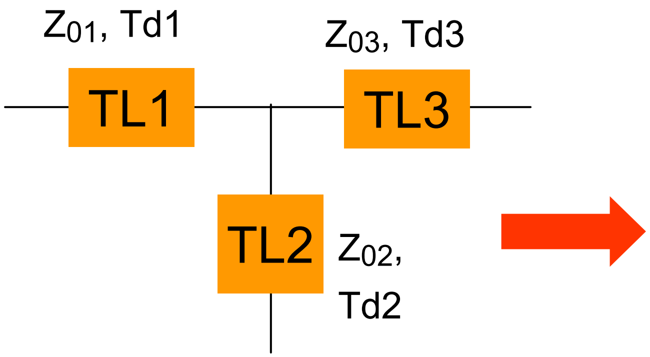
Example of PWL approximation (s21): ~2nS, 18 sample

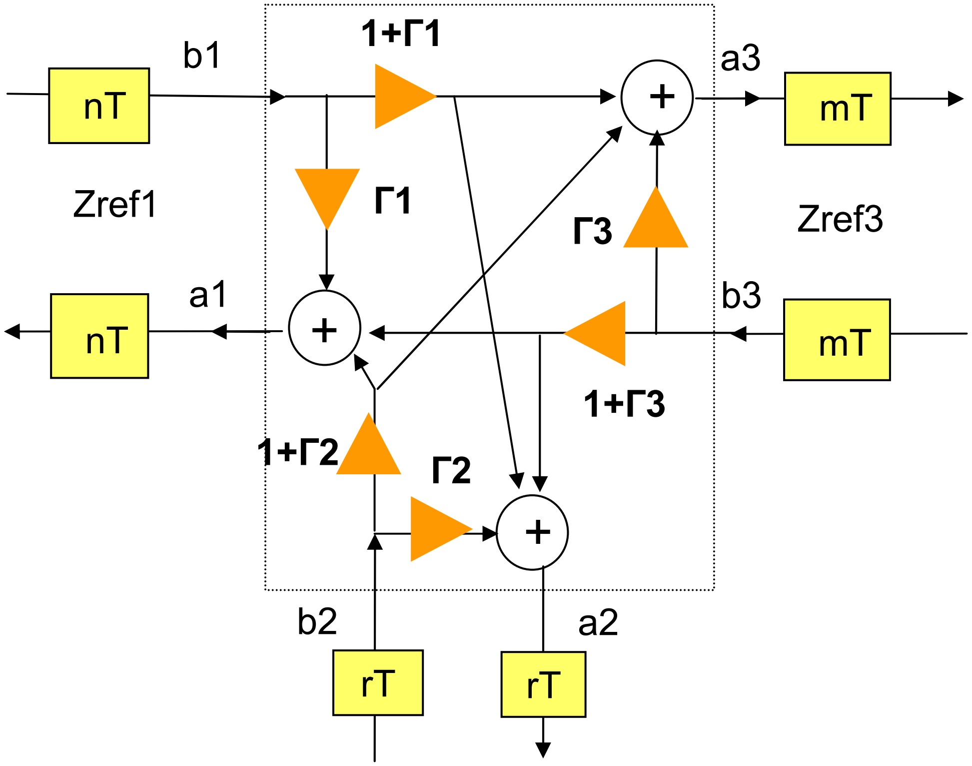
Time (ps=index)	0	338	346	353	362	372	386	405	436	477	534	652	863	993	1014	1040	1125	1923
S21 (Rho)	0	0	0.07	0.30	0.55	0.66	0.75	0.81	0.86	0.89	0.90	0.92	0.93	0.94	0.94	0.94	0.94	0.95

Simulation time ~70s (Linux)

T- Junctions (n-port adapter)

The 2-port adapter can be easily extended to n-port:



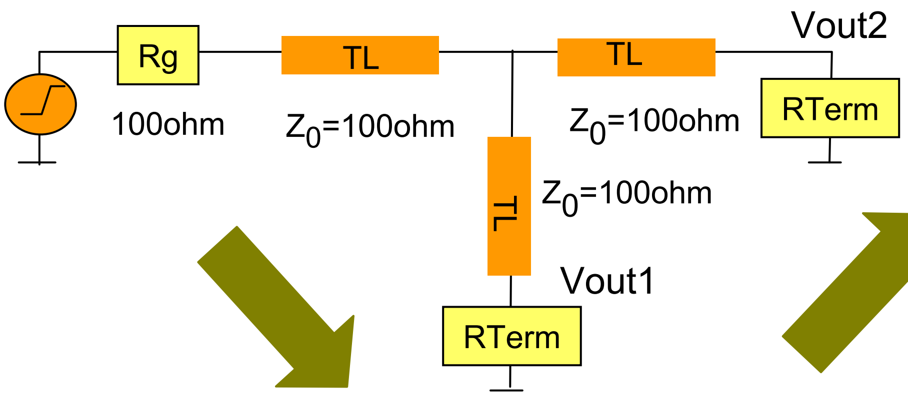


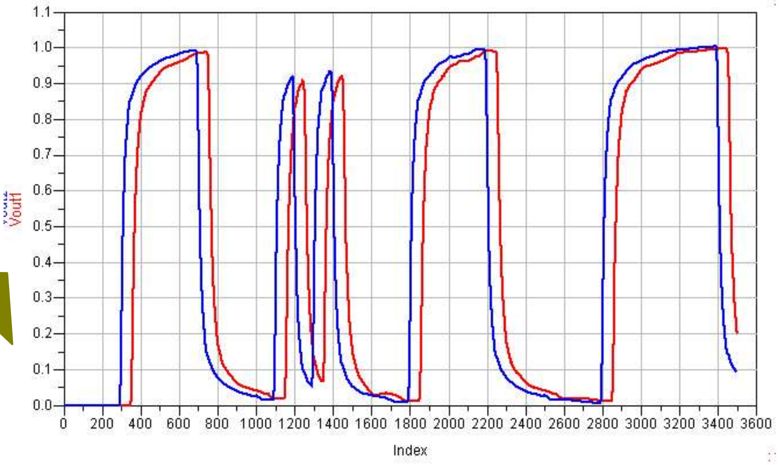
$$\Gamma 1 = \frac{Z_{02}//Z_{03} - Z_{01}}{Z_{02}//Z_{03} - Z_{01}}$$

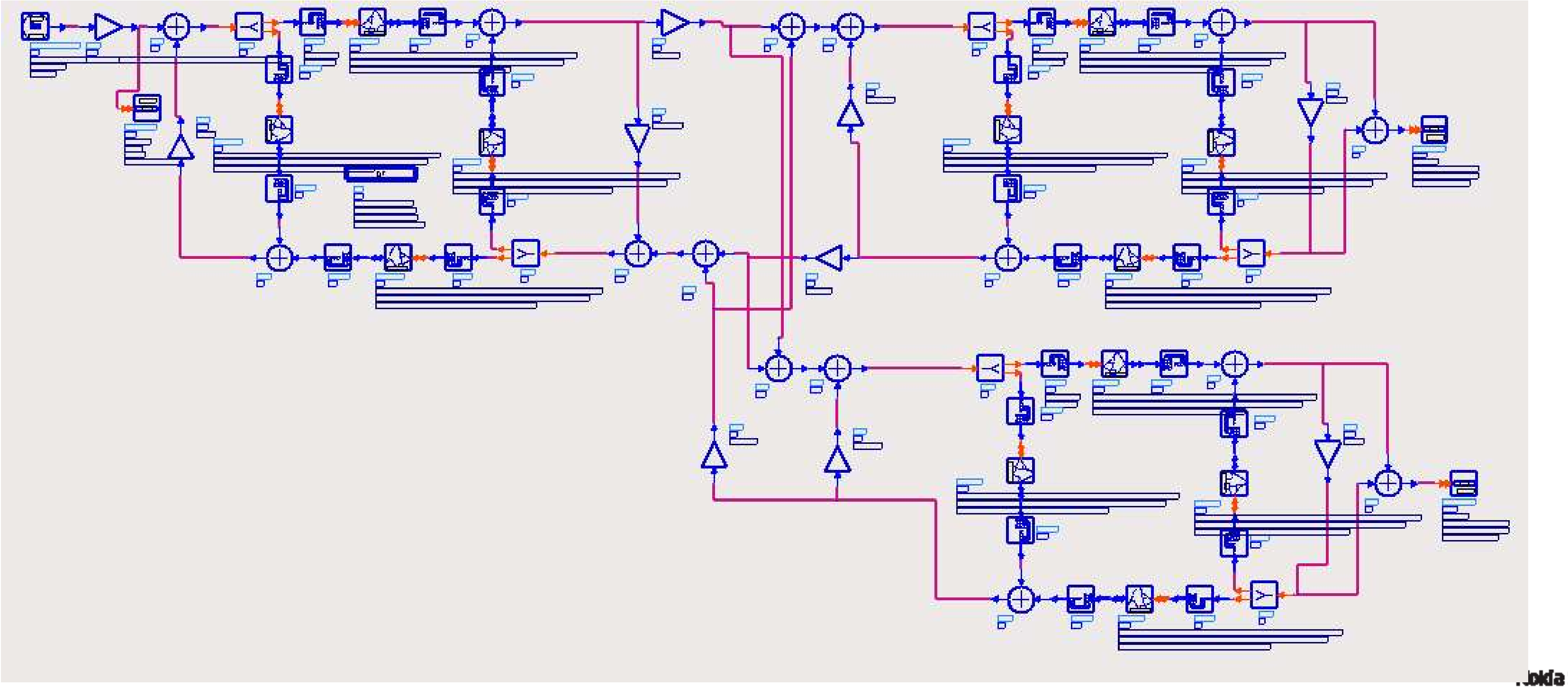
$$\Gamma 2 = \frac{Z_{03}//\ Z_{01} - Z_{02}}{Z_{03}//\ Z_{01} - Z_{02}}$$

$$\Gamma 3 = \frac{Z_{01}//Z_{02} - Z_{03}}{Z_{01}//Z_{02} - Z_{03}}$$

T-junction example







Conclusions

- This presentation has shown an application of ADS Ptolemy engine to signal integrity simulations of digital systems based on their digital wave network representations. The method can be applied to any electrical element, including multiconductor structures [2]
- By introducing a Piece-Wise Linear (PWL) approximation of the transfer functions it is possible to dramatically speed-up the convolution algorithm [1].

Possible evolution:

- Impedance adapters have been implemented using basic ADS elements. A dedicated library element (2 or more ports) could be useful.
- Fast convolution based on PWL approximation of the transfer functions has been implemented by means of MATLAB algorithms. A dedicated library element (very similar to FIR element) could be useful.
- In general, the whole digital wave network could be automatically generated by the software, starting from standard schematic capture representation (generator, line, RLC, etc.) [3].

References

[1] “Use of wave digital networks for time domain simulations of lossy interconnections in digital systems”, P.Belforte, B. Bostica, G. Guaschino, Proceeding International Symposium on circuit and systems, May 1982

[2] “A consideration of time domain analysis of networks containing coupled transmission lines using their wave digital filter representation”, Hiroyuki Wakabayashi, et alt., Electronics and Communications in Japan, Vol. 67-A, No. 4, 1984.

[3] SPRINT® simulator, High Design Technology (HDT) S.r.l., Turin, Italy