

advico®

Calculator Functions Manual

September 2010

■ About the Calculator Tools.....	3
■ General Notes.....	3
■ Installation.....	4
■ EyeDiag.....	5
■ MomAmp.....	5
■ MomFreq.....	6
■ MomTime.....	6
■ MomPhase.....	7
■ Resample.....	7
■ XHist.....	8
■ YHist.....	8
■ Write.....	9
■ BusToInteger.....	10
■ Example 1: PRBS sequence.....	11
■ Example 2: AM and FM modulated sine wave.....	12
■ Example 3: IQ signals, one AM and FM modulated.....	13
■ AdvlCoOceanTools (Corner Analysis Tools).....	14

About the Calculator Tools

Although Cadence offers a huge library of functions to post-process your simulation data, there are things which you may be missing. Eye diagrams which require that the simulation is finished, frequency measurements which take an average of the simulation only, or histogram functions which don't exist may be examples why you may want to use **advico**'s calculator functions.

All functions - **EyeDiag**, **MomAmp**, **MomFreq**, **MomTime**, **MomPhase**, **Resample**, **XHist**, **Yhist**, **BusToInteger** and **Write** - appear as plug-ins in Cadence[®] Calculator and are listed in the "Special Functions" menu of Cadence[®] Calculator. They are executed either by typing the commands at the Calculator command line or by using an input mask which is invoked from the "Special Functions" menu of the Cadence[®] Calculator.

All functions will already respond while the simulation is running. This allows an early check of the simulation results which may save you much time (in the waveform window, perform a Window|UpdateResults operation to re-read the last simulation results).

New: beginning with V0.89 calculator functions can operate on single waveforms as well as on parametric plots, multiple-parametric plots, and Monte-Carlo simulations. The functions have been updated to operate in Cadence version IC6 as well as IC5.

General Notes

Sometimes it is desirable to refer to design variables when plotting curves. To access these, use VAR ("variableName") instead of numeric values in the calculator expression. This will also resolve proper values in multiple-parametric plots.

Installation

The calculator tools require two files to be loaded. One file contains the actual functions, the other one your license. The license file will be loaded automatically by the function file **calculator.il**. To load the file from the Cadence dfl framework, type **load("calculator.il")** in the CIW window or add this line to your **.cdsinit** file (located at the place where Cadence is started). This works if the calculator.il file is located in the same directory or somewhere in the Cadence SKILL search path. If you store this file at a different place, prepend the path to the file, e.g. type **load("~/skillTools/calculator.il")**.

If the license file is named **calculatorLic.il** it will be loaded automatically in any of these cases:

- calculatorLic.il placed in Cadence start directory
- calculatorLic.il placed in \$PROJECT/il directory
- calculatorLic.il placed in \$HOME directory
- calculatorLic.il placed in directory specified by \$CALC_LIC_PATH

If you specify the environment variable **CALC_LIC_PATH**, you can also add the license file name. In this case you can rename the license file as desired. An example for a csh or tcsh environment is

```
setenv CALC_LIC_PATH  
/tools/licences/myCalculatorLic.il
```

Please note that this setting must be done before you start Cadence.

EyeDiag

Description:

EyeDiag draws an **eye diagram** of the Signal (e.g. random data), beginning after T_{Delay} , with a bit period of T_{Per} . It will draw N eyes on the screen. The desity of the diagram does not depend on the number of eyes, thus you can select a higher number and still need only a short simulation time. T_{delay} is used to remove possible settling processes from the eye diagram.

Execution:

Click in Calculator on „Special functons“ -> „EyeDiag“

Or: Type in Calculator command line:

EyeDiag (Signal, T_{Delay} , T_{Per} , N)

Example:

EyeDiag (VT(„/out“), 2n, 25p, 3)

This draw a 40Gbit/s eye diagram (25 ps per bit) with 3 eyes on the screen aftr 2ns simulation settling time.

MomAmp

Description:

MomAmp calculates the **instantaneous peak-to-peak amplitude** of the Signal which lies around threshold. Useful e.g. for VCOs or AM-modulates signals.

Execution:

Click in Calculator on „Special functions“ -> MomAmp“

Or: Type in Calculator command line:

MomAmp (Signal, from, to, threshold)

Example:

MomAmp (VT(„/out“), 5N, 50n, 0)

MomFreq

Description:

MomFreq calculates the **instantaneous frequency** of the Signal between from and to time values, based on the rising edge of the signal, crossing threshold. Useful e.g. for VCO and PLL simulations.

Execution:

Click in Calculator on „Special functions“ -> „MomFreq“

Or: Type in Calculator command line:

MomFreq (Signal, from, to, threshold)

Example:

MomFreq (VT(„/out“), 5n,50n, 0.0)

MomTime

Description:

MomTime calculates the **time difference between rising edges** of Signal1 crossing threshold1 to Signal2 crossing threshold2 in between from and to time values. Useful for delay measurements or phase measurements (e.g. quadrature mixers: MomTime(Sig_I,Sig_Q,...) $F_{ref} * 360.0$).

Execution:

Click in Calculator on „Special functions“ -> „MomTime“

Or: Type in Calculator command line:

MomTime (Signal1, Signal2, from, to, threshold1, threshold2)

Example:

MomTime (VT(„ref“), VT(„/out“), 5n, 50n, 0.0-0.1)

MomPhase

Description:

MomPhase calculates the **instantaneous phase** of the rising edge of the Signal crossing threshold and the reference signal with frequency f_0 and initial phase ϕ_0 . Useful e.g. for VCO and PLL simulators.

Execution:

Click in Calculator on „Special functions“ -> MomPhase“

Or: Type in Calculator command line:

MomPhase (Signal, f_0 , ϕ_0 , from, to, threshold)

Example:

MomPhase (VT(„/out“), 10G, 180, 5N, 50n, 0.0)

Resample

Description:

Resample **resamples the waveform** value at time intervals $T_0 + N \cdot T$, based on linear interpolation. It is used by our statistical functions but may also be useful for data sampling in the calculator.

Execution:

Click in Calculator on „Special functions“ -> „Resample“

Or: Type in Calculator command line:

Resample (Signal, T, T_0)

Example:

Resample (VT(„/out“), 25p, 0)

XHist

Description:

XHist creates a normalized **histogram of Signal crossings** of Threshold. The simulation data is internally resampled for this operation. The signal period T is used to calculate the spread of the crossing times. Time resolution of the histogram is T/Resolution. For Edge=0 both rising and falling edges are taken into account, for +1 only rising edges for -1 only falling edges.

Execution:

Click in Calculator on „Special functions“ -> “X-Hist“

Or: Type in Calculator command line:

XHist (Signal, Threshold, T, Resolution, Edge)

Example:

XHist (VT(„/out“),0.0, 25p, 100, 0

YHist

Description:

YHist creates an **amplitude histogram** of the Signal. The Signal is internally resampled with a period T for this operation. Amplitude resolution is (ymax-ymin)/Resolution, based on the measured extrema of the Signal. For flip=nil the x-axis contains the amplitude and y contains the normalized amplitude distribution, for Flip=t the axis are swapped.

Execution:

Click in Calculator on „Special functions“ -> „YHist“

Or: Type in Calculator command line:

YHist (Signal, Resolution, T, Flp)

Example:

XHist (VT(“/out”), 200, t)

Write

Description:

Write writes an **ASCII data file** (gnuplot style) with x/y-pairs. For parametric plots the individual parameter values are listed as a comment. Comment is an ASCII string that may hold information about the meaning of the y/y values. This command is better executed in the CIW instead of the calculator.

Execution:

Click in Calculator on „Special functions“ -> “Write“

Or: Type in Calculator command line:

Write (Signal, Filename, Comment)

Example:

Write (VT(„/out“), „//tmp/result.dt“, „x=time, y=V(out), param=VDD“)

BusToInteger

Description:

BusToInteger converts the digital signals of a data bus into a waveform representing the numerical value of the data. Busses have names such as D<0:11>, individual members being D<0>, D<1> etc.

The first argument is one (arbitrary) member of the bus, such as VT („/D<0>“).

The second argument is a string with the bus to be evaluated, for example „VT(/D<0:11>“ or VT//D<7:15>“. You can specify this bus in two ways. The formally correct one is using protected quotes in the string, e.g. „VT(\"/D<7:15>\“)“, the notation above is preferred for readability.

The 3rd argument is a sampling time, used to probe all data lines and to calculate a data value.

The 4th argument is the decision threshold to determine if a line is logically 0 or 1.

The 5th argument is a flag (0 or 1) which allows flipping of the weight. If 0, D<0> is the LWB and D<11> the MSB, if 1, the bit order is inverse.

Execution:

Type in Calculator command line:

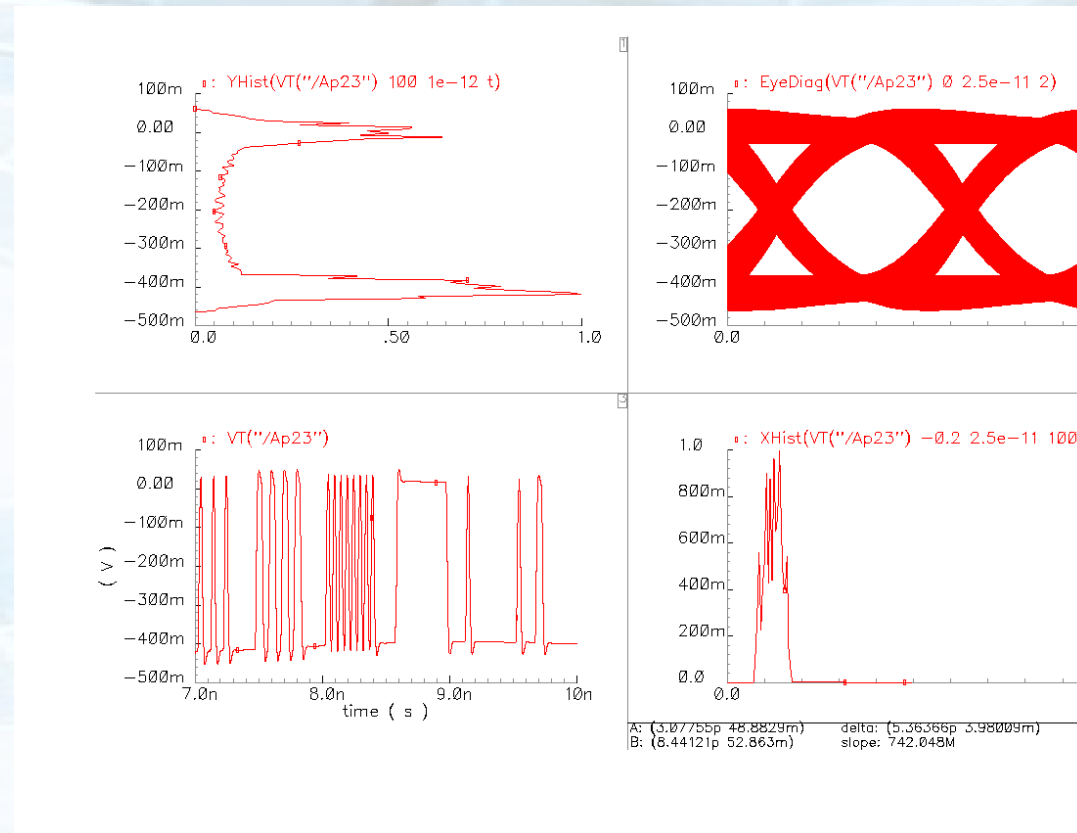
BusToInteger (SignalOfBus „Busname“ sampling Period logic Threshold flip)

Example:

BusToInteger (VT („/I/O/D<0>“), VT(/I/O/D<0:11>“, 10n, 1.25, 0)

NOTE: This function currently doesn't work with multiple parametric sweeps. Furthermore, the command must be typed directly in the calculator window. Both issues should be fixed in the next release.

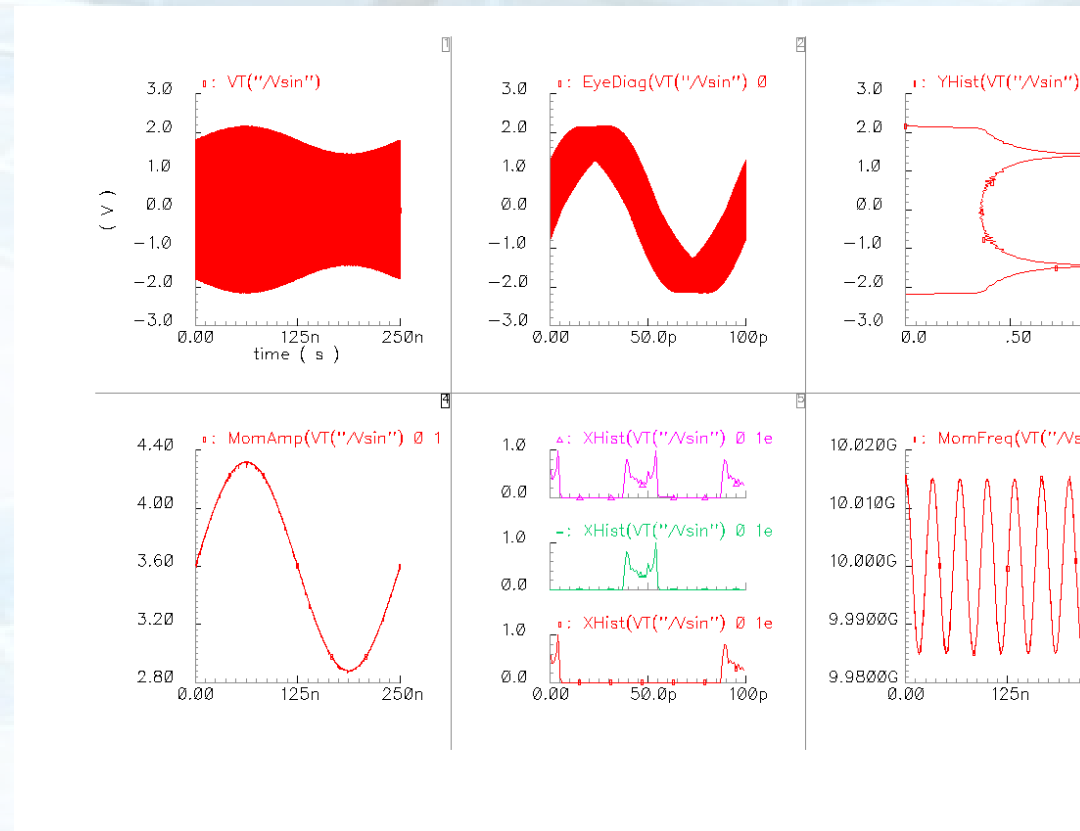
Example 1: PRBS sequence



1) amplitude histogram
3) original y(t) data plot

2) eye diagram
4) jitter measurement, all edges

Example 2: AM and FM modulated sine wave



1) amplitude histogram

2) eye diagram

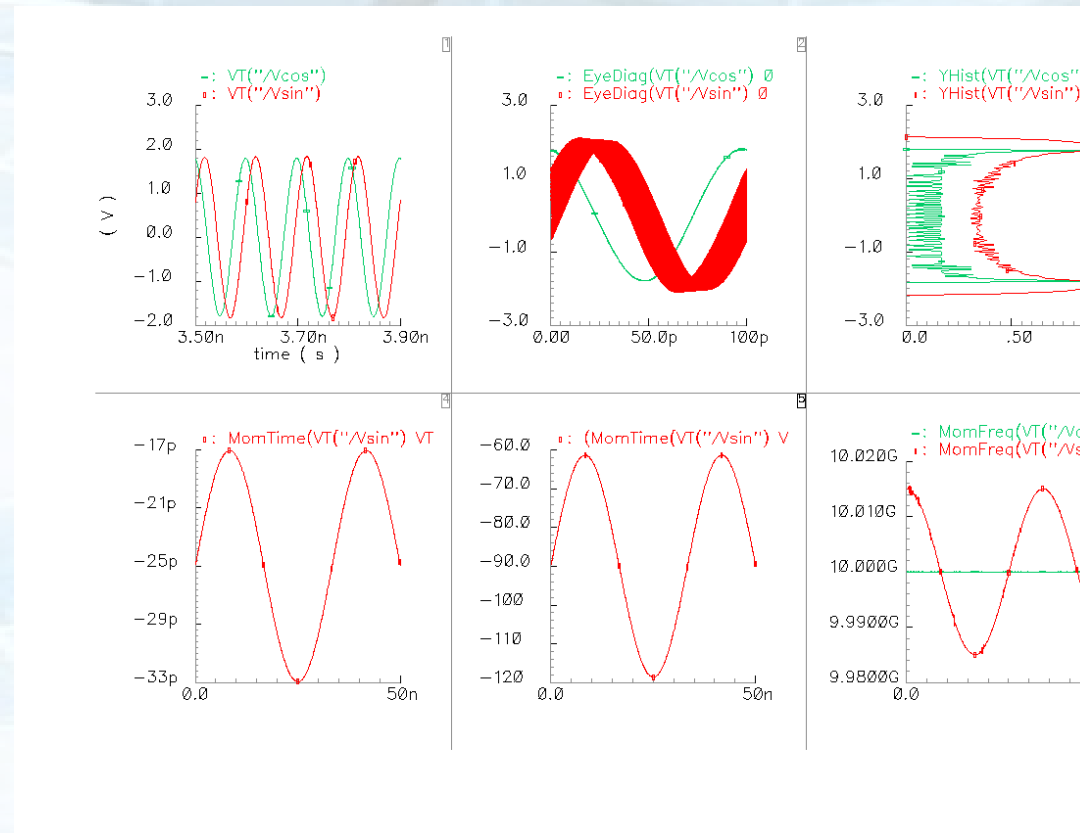
3) amplitude histogram

4) instantaneous amplitude

5) jitter of all/falling/rising edges

6) instantaneous frequency

Example 3: IQ signals, one AM and FM modulated



1) original $y(t)$ data plot
4) rising edge time delta

2) eye diagram
5) rising edge phase shift

3) amplitude histogram
6) instantaneous frequency

AdvCoOceanTools (1)

Description:

The advCoOceanTools allow you to perform corner and parametric analysis with a text-based setup. Beside a convenient way to generate all permutations of model corners, temperatures and design variables, it also provides a framework to calculate parameters and to do plots.

The advCoOceanTools are structured in a way that you can start the simulations in one Cadence session and finish the setup of the analysis while the simulator is running. In addition, starting another Cadence session allows you to evaluate the results while they are generated.

The simulation results for each run are stored in separate subdirectories runNN within the usual ADC directory tree. They are independent of ADE and do not get deleted when ADE is run.

The advCoOceanTools consist of two parts. One part, advCoOceanTools.il, is a fixed set of predefined functions. The other part is the user defined part which will be created by modifying a sample script according to your needs. The following sections will briefly describe how to use and set up such a script. Reference data can be found in Cadence' AWD and Ocean manuals

Execution:

Type into the CIW:

```
load("myCorner.ocn")
```

If the user-defined script is located in your startup directory. See the example calls, pick one by clicking on it, then press RETURN.

Example:

```
load("myCorner.ocn")
```

```
MyCornerSimulation(t „simulate“)
```

Will load the script and perform a parametric simulation, but it will not evaluate the data at the same time

```
MyCornerSimulation(t „both“ t)
```

Will „simulate“ and „analyze“ and also plot the results. Note that it's not recommended to do simulation, analysis and plots in the same Cadence session because during analysis and plot phases no simulation will happen. Use of another Cadence session is much faster.

Syntax (can be changed by the user):

```
MyCornerSimulation(DoParametric Action Plot)
```

DoParametric: t or nil or a list of corner numbers to be simulated, e.g. '(1 2 3)

Action: „simulate“, „analyze“ or „both“

Plot: t or nil. If nil, Calculations will be made and logged to a file, but no waveform window will be updated

AdvCoOceanTools (2)

Detailed Description:

In the user-defined script, the following functions should be modified:

Procedure CornerSetup_<Technology>

Here you specify the set of model files and corners, the temperatures to be simulated, and the design variables. You can specify single values or lists of values.

The procedure defines two options within an if-clause: One if the setup for parametric/corner analysis, the other one is a single-corner setup, usually the typical case.

In case you don't have a template for your technology already set up, the easiest way to create such a template is to start a simulation session in the ADE and inspect the netlist or ADE model setup. ADE can also provide a list of all required design variables. Everything you need to start the simulation from ADE must be defined in this procedure as well.

Example (excerpt):

```
TempList = list(0 27 80)
VarList = list(
    ("Fmod" list(100k))
    ("iAC" list(1))
    ("iDC" list(0 10n 1u))
    ("Freq" list(590G))
    ("vg" list(0.5 0.6 0.7))
); VarList
```

Procedure SetupSimulator

Here you specify the convergence criteria and save options. Defaults as well as an example for "selected node saving" are given. If many permutations are needed, saving only the required nodes is highly recommended.

Example (excerpt):

```
saveOption( ?saveahdlvars "selected" )
saveOption( 'currents "selected" )
saveOption( 'save "selected" )
save( 'v "/invIn" "/invOut" "/divIn" )
save( 'i "/V0/PLUS" )
```

Procedure DoCalculation

This procedure should hold all commands to read simulator results, store them for analysis, as well as the analysis itself and printing them into files according to your needs. You can use all calculator functions as well as full SKILL language features. Special procedures are provided to simplify data handling. Data can be waveforms, numbers, boolean values or anything else.

Example (excerpt):

```
VSoutP = VS("/outP")
VSoutN = VS("/outN")
VSdiff = VsoutP-VSoutN
VSdiffMin = ymin(VSdiff)
VSdiffMax = ymax(VSdiff)
VSdiffPP = VSdiffMax-VSdiffMin
Data->VSoutP = append1(Data->VSoutP VSoutP )
Data->VSoutN = append1(Data->VSoutN VSoutN )
Data->VSdiff = append1(Data->VSdiff VSdiff )
Data->VSdiffMin = append1(Data->VSdiffMin VSdiffMin )
Data->VSdiffMax = append1(Data->VSdiffMax VSdiffMax )
Data->VSdiffPP = append1(Data->VSdiffPP VSdiffPP )
```

AdvCoOceanTools (3)

Procedure DoPlots

Here you specify the waveform setup. This procedure should only access the data which has been saved in the Data structure defined in the DoCalculation procedure.

Example (excerpt):

```
awvSetCurrentSubwindow(windowID 2) ; choose desired subwindow
clearSubwindow()
addSubwindowTitle("output RMS noise") ; all data points will be plotted as a curve
displayMode( "composite" )
plot(FilterWave(data->VNrms1G) ?expr list(sprintf(nil "1k-1G %L" RunNumber)) ?
yNumber list(1))
plot(FilterWave(data->VNrms1M) ?expr list(sprintf(nil "1k-1M %L" RunNumber)) ?
yNumber list(2))
```

Procedure RunSimulation

Here you specify the task for the simulator, e.g. which kind of simulation (ac/dc/tran) and with which options should be run.

Example (excerpt):

```
analysis('ac ?start 1 ?stop 10000 ?lin 100 ) ; ?lin ?dec
analysis('tran ?start 0 ?stop 1u ?step 10n ) ; ?lin ?step
analysis('dc ?oppoint "rawfile" ?save "allpub" ?param "temp" ?start -50 ?stop
100 )
analysis('dc ?saveOppoint t )
analysis('noise ?start "10" ?stop "10M" ?p "/TIA_vout_bus1" ?n "/gnd!" ?oprobe ""
?iprobe "/I24")
```

Other procedures

The other procedures in this file are only to glue everything else together. Only few other places may need modifications (e.g. control the number of graphs).

Functions provided by the advCoOceans.il file

The following is a list of general purpose functions which are useful for the user:

Print(fh <args>)

is a replacement for printf and fprintf. Given an already defined file handle fh, your text will go to the CIW and main Cadence CDS.log as well as into the appropriate simulation or analysis log file.

GetElem(elem VarList)

Is a procedure to grab values or waveforms from parameter or result structures.

AddToData(varList)

Will add your variable or list to the global Data structure

FilterWave(arg @optional (arg2 nil))

is used in DoPlots to pass data to the waveform window. Purpose of this function is to provide valid waveforms in any case. E.g. single numbers are changed into a 2D-dot. A list of single values will be changed into a graph. Two 1D lists can be combined into a xy-waveform.

FilterNumber(arg)

Is used in DoCalculation to prevent illegal values to be stored which may break during succeeding calculations. If a calculation failed or a waveform doesn't exist, it will return zero.

Disclaimer

- 1) adviCo warrants that the programs conform with the above specifications. However it is expressly stated that the current source and binary representation of the programs may not exclude the occurrence of defects or malfunctions under certain circumstances.
- 2) adviCo will not warrant the licensed programs if the customer modifies the licensed programs or damages the media upon which the licensed programs are supplied.
- 3) If any defects cannot be remedied at reasonable expense, adviCo is entitled to terminate the license for the licensed program concerned and refund the license fee.
- 4) The right to request remedy of a defect shall lapse 6 months after shipment of the Licensed programss to Customer.
- 5) adviCo shall be liable for damage to the customer caused by defects and malfunctions in the Licensed Program only in case of gross negligence and wilful intent. adviCo shall be liable for direct damages caused by slight negligence up to the amount of the license fee.