Perl Script Tutorial

(Last updated: Nov 30, 2015)

Perl Script Tutorial

This tutorial was created to document the steps needed to extract simulation results from .lis file after Monte Carlo simulations.

Perl Basic Grammar

1. You can have any variable name without defining, but the variable name should starts with "$". (e.g. $ex1, $ex2, etc.)
2. You can also use an array by expressing "[" and "]". (e.g. $array_ex[0], $array_ex[2], etc.)
3. @ represents the entire array (e.g. @sample_array), while $ represents an item in the array (e.g. $sample_array[1]).
4. You don't have to pre-define the variable type (i.e. binary, decimal, string, and etc.) or the variable/array size.
5. You can comment any line out by putting "#" at the beginning of each line (e.g. # comments).
6. Each command line is distinguished with a semicolon (;).

How to Run the Sample Perl Script

The following sample script saves a data to a temporary array (i.e. @original_data) from .lis file after 100 Monte-Carlo simulations (MC), and aligns them. In order to use this script, please specify how many data points you have per simulation (i.e. $len). For instance, you are sweeping a voltage source from 0V to 1.2V in every 100mV, then you have 13 data points. In this case, please specify $len=17 (number of data points + 4). Next, specify the number of iterations (i.e. how many Monte-Carlo you run) in $MC variable. Then, you are good to run this script. Note that the script assume the results have three columns of data (e.g. time, input, and output), so please change the script depending on your own purpose. Now, you are ready to run the script. Run the script as follows:

% perl ex.lis output.txt

ex.lis: the resulting output file from HSPICE.
output.txt: output text file from the script.

You can change the names of files whatever you want.

Sample Perl Script

```
# get two files names

my $in_file = $ARGV[0];
my $out_file = $ARGV[1];

open my $IN, "<", $in_file;
open my $OUT, ">", $out_file;

$i=0;
$len=105;
$MC=100;

foreach(<$IN>){
        chomp $_;
        if($_ eq " x"){
                $flag=1;}
    elsif($_ eq "y"){
                $flag=0;}

        if($flag eq 1){
                $original_data[$i]=$_;$i++;}

}

for($i=4;$i<$len;$i++){
        for($j=0;$j<$MC;$j++){
                my @temp=split /\s+/,$original_data[$j*$len+$i];
                printf $OUT $temp[1];
                printf $OUT " ";
                printf $OUT $temp[2];
                printf $OUT " ";
                printf $OUT $temp[3];
                printf $OUT " ";
        }
        printf $OUT "\n";
}
```

```
# close files
close $IN;
close $OUT;
```