# SIS TOOL GUIDE

## I. Running SIS

First, you need to make sure that the path is setup correctly for SIS in UNIX. To see if the path is setup or not, type the command **which sis**. If it shows you a path, it means that you can access sis. It should give you the path as /tools/sis/bin/sis.
If it does not then edit the file .cshrc and add the following line

Set path=($path /tools/sis/bin)

Then, run the command **source .cshrc**
You need to do this once. Now type the command **which sis**, and you should get the correct path.

To run SIS, you just need to type the command **sis**.

## II. Input/Output in SIS

In SIS, you can read the input in several formats including PLA< equation and blif (Berkeley logic interchange format). For example, the following is a description of a full-adder in the three formats:

### 1. Equation Format:

```
# To specify the primary inputs
INORDER= a b cin;
# To specify the primary outputs
OUTORDER= sum cout;
# Each equation becomes a node in the logic network
sum = a' b' cin + a' b cin' +  a b' cin' + a b cin ;
cout = a b + a cin + b cin;
```

To read a file in equation format (say file.eq) you need to type the command:

***Read_eqn file.eq***

To save your logic in file file.eq in equation format, you need to type the command:

*Write_eqn file.eq*

**2. PLA Format:**

*.i 3*
*.o 2*
*.ilb a b cin*
*.ob sum cout*
*.p 7*
*111 10*
*001 10*
*010 10*
*100 10*
*11- 01*
*1-1 01*
*-11 01*
*.e*

To read a file in PLA format (say file.pla) you need to type the command:

*Read_pla file.pla*

To save your logic in file file.pla in PLA format, you need to type the command:

*Write_pla file.pla*

**3. BLIF Format:**

```
.model fadder.eq
.inputs a b cin
.outputs sum cout
.names a b cin sum
111 1
001 1
010 1
100 1
.names a b cin cout
11- 1
1-1 1
-11 1
.end
```

To read a file in BLIF format (say file.blif) you need to type the command:

***Read_blif file.blif***

To save your logic in file file.blif in BLIF format, you need to type the command:

***Write_blif file.blif***


## III. Logic Statistics

To see the logic of your network, you can type the command ***print*** and it will print for you the logic in equation format.
The following is what will be shown for the full adder example when we type the print command:

```
sis> print
    {sum} = a b cin + a b' cin' + a' b cin' + a' b' cin
    {cout} = a b + a cin + b cin
```

To get statistics about your logic, type the command ***print_stats.***

The following is what will be shown for the full adder example when we type the print_stats command:
```
sis> print_stats
fadder.eq      pi= 3   po= 2   nodes= 2        latches=0
lits(sop)=  18
```


## IV. SIS Commands

SIS implements many functions for two-level logic synthesis, multilevel logic synthesis, technology mapping, sequential logic synthesis, logic simulation. See the SIS manual for a description of all the commands.