

# VSD TCL WORKSHOP

## DAY 1

Day 1's task is to create a command (in my case, `vsynui`) and pass a `.csv` file from the UNIX shell to the TCL script, taking into consideration mainly the general scenarios from the user's point of view.

### Task:

#### Task

#### Sub-Task and tools needed

- Create command (for eg. `vsdsynth`) and pass `.csv` from UNIX shell to TCL script

General scenarios – From User point of view

1. Not provide `.csv` file as input

```
kunalg@kunalg-VirtualBox ~/Desktop/vsdfLOW $ ./vsdsynth
```

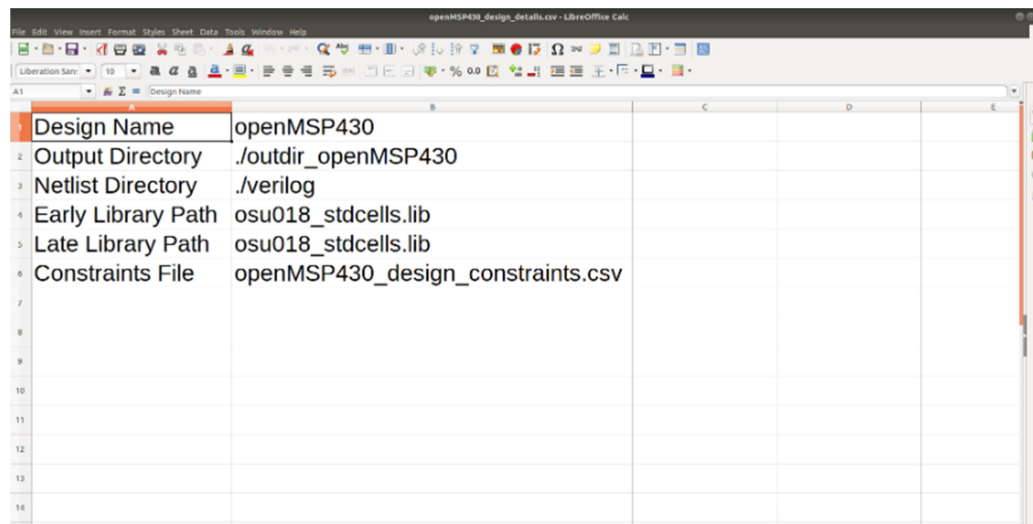
2. Provide a `.csv` file, which doesn't exist

```
kunalg@kunalg-VirtualBox ~/Desktop/vsdfLOW $ ./vsdsynth my.csv
```

3. Type “-help” to find out usage

```
kunalg@kunalg-VirtualBox ~/Desktop/vsdfLOW $ ./vsdsynth -help
```

### Inputs File:



The screenshot shows a LibreOffice Calc spreadsheet titled 'openMSP430\_design\_details.csv'. The spreadsheet has a single data row with the following values:

Design Name	openMSP430
Output Directory	./outdir_openMSP430
Netlist Directory	./verilog
Early Library Path	osu018_stdcells.lib
Late Library Path	osu018_stdcells.lib
Constraints File	openMSP430_design_constraints.csv

## Implementation:

1. Creating the ./synui command by writing a bash script to cover the general scenarios of input file present or not.

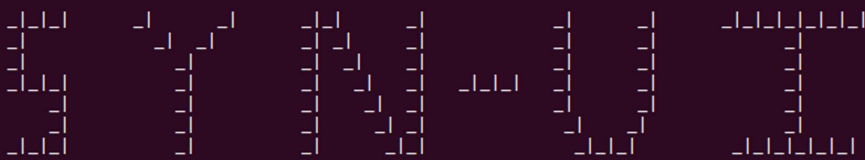
Script:

```
#!/bin/bash
echo "
  _|_|_|_   _|_|_|_   _|_|_|_   _|_|_|_   _|_|_|_|_|_|_|_
  _|_|_|_   _|_|_|_   _|_|_|_   _|_|_|_   _|_|_|_|_|_|_|_
  _|_|_|_   _|_|_|_   _|_|_|_   _|_|_|_   _|_|_|_|_|_|_|_
  _|_|_|_   _|_|_|_   _|_|_|_   _|_|_|_   _|_|_|_|_|_|_|_
  _|_|_|_   _|_|_|_   _|_|_|_   _|_|_|_   _|_|_|_|_|_|_|_
  _|_|_|_   _|_|_|_   _|_|_|_   _|_|_|_   _|_|_|_|_|_|_|_
"
echo "  An innovative user interface (UI) that accepts the following inputs that are present in a CSV file:"
echo "  1] RTL netlist 2] Library files for fast and slow scenarios. 3] SDC constraint file"
echo "  It then outputs 1] Synthesized netlist 2] Pre-layout timing report."
echo "  OpenTimer is used for generating timing reports and Yosys is used to perform synthesis."
echo "  The UI is designed and maintained by Niharika Kummithi (Physical Design Engineer)"
echo "  For any queries or bugs, Please drop a mail to - niharika.kummithi98@gmail.com"
echo
#Code to handle the scenario where user does not give any file, does not give .csv file, gives more than one file as argument
if [ $# -eq 0 ]
then
    echo "Info: Please provide a CSV file"
    exit 1
elif [ $# -gt 1 ]
then
    echo "Info: Please provide only 1 CSV file"
    exit 1
else
    if [[ $1 != *.csv && $1 != "-help" ]]
    then
        echo "Info: Please provide a .csv format file"
        exit 1
    fi
fi
# Code to check if the .csv file is present in directory or not, and also to display information for -help argument.
if [ ! -f $1 ] || [ $1 == "-help" ]
then
    # Code to check if the .csv file is present in directory or not, and also to display information for -help argument.
    if [ $1 != "-help" ]
    then
        echo "Error: The file $1 is not found in current directory."
        exit 1
    else
        echo "USAGE: ./synui <csv_file>"
        echo "  where <csv file> consists of 2 columns, below keyword being in 1st column and is Case Sensitive. Please request Niharika for sample csv file."
        echo "  <Design Name> is the name of top level module."
        echo "  <Output Directory> is the name of output directory where you want to dump synthesis script, synthesized netlists and timing reports."
        echo "  <Netlist Directory> is the name of directory where all RTL netlist are present."
        echo "  <Early Library Path> is the file path of the early cell library to be used for STA."
        echo "  <Late Library Path> is file path of the late cell library to be used for STA."
        echo "  <Constraints file> is csv file path of constraints to be used for STA."
        exit 1
    fi
else
    #Code to execute if the proper CSV file exists.
    echo "Info: CSV file accepted"
    tclsh synui.tcl $1
fi
```

## Results

### Case 1: No input file

```
vsduser@vsduser-tclworkshop:~/vsdsynth$ ./synui
```

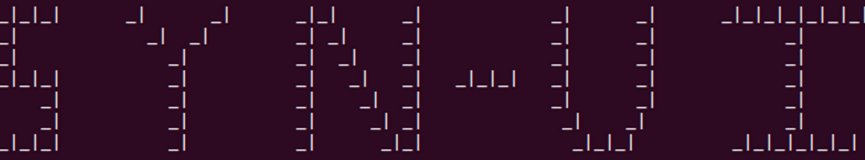


```
An innovative user interface (UI) that accepts the following inputs that are present in a CSV file:  
1] RTL netlist 2] Library files for fast and slow scenarios. 3] SDC constraint file  
It then outputs 1] Synthesized netlist 2] Pre-layout timing report.  
OpenTimer is used for generating timing reports and Yosys is used to perform synthesis.  
The UI is designed and maintained by Niharika Kummithi (Physical Design Engineer)  
For any queries or bugs, Please drop a mail to - niharika.kummithi98@gmail.com
```

Info: Please provide a CSV file

### Case 2: File not of .csv format

```
vsduser@vsduser-tclworkshop:~/vsdsynth$ ./synui my
```

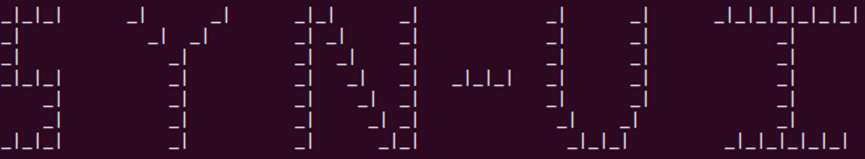


```
An innovative user interface (UI) that accepts the following inputs that are present in a CSV file:  
1] RTL netlist 2] Library files for fast and slow scenarios. 3] SDC constraint file  
It then outputs 1] Synthesized netlist 2] Pre-layout timing report.  
OpenTimer is used for generating timing reports and Yosys is used to perform synthesis.  
The UI is designed and maintained by Niharika Kummithi (Physical Design Engineer)  
For any queries or bugs, Please drop a mail to - niharika.kummithi98@gmail.com
```

Info: Please provide a .csv format file

### Case 3: More than 1 file given as input

```
vsduser@vsduser-tclworkshop:~/vsdsynth$ ./synui my1.csv my2.csv
```

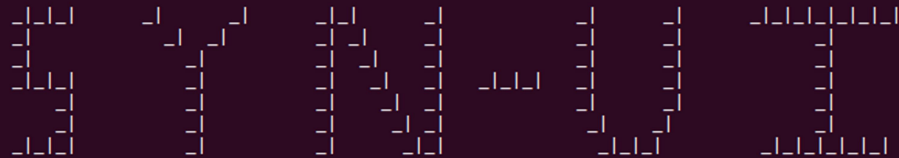


```
An innovative user interface (UI) that accepts the following inputs that are present in a CSV file:  
1] RTL netlist 2] Library files for fast and slow scenarios. 3] SDC constraint file  
It then outputs 1] Synthesized netlist 2] Pre-layout timing report.  
OpenTimer is used for generating timing reports and Yosys is used to perform synthesis.  
The UI is designed and maintained by Niharika Kummithi (Physical Design Engineer)  
For any queries or bugs, Please drop a mail to - niharika.kummithi98@gmail.com
```

Info: Please provide only 1 CSV file

Case 4: The .csv file provided is not present in current directory.

```
vsduser@vsduser-tclworkshop:~/vsdsynth$ ./synui my1.csv
```

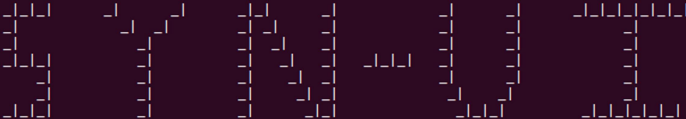


An innovative user interface (UI) that accepts the following inputs that are present in a CSV file:  
1] RTL netlist 2] Library files for fast and slow scenarios. 3] SDC constraint file  
It then outputs 1] Synthesized netlist 2] Pre-layout timing report.  
OpenTimer is used for generating timing reports and Yosys is used to perform synthesis.  
The UI is designed and maintained by Niharika Kummithi (Physical Design Engineer)  
For any queries or bugs, Please drop a mail to - niharika.kummithi98@gmail.com

Error: The file my1.csv is not found in current directory.

Case 5: Result of using the option '-help'

```
vsduser@vsduser-tclworkshop:~/vsdsynth$ ./synui -help
```



An innovative user interface (UI) that accepts the following inputs that are present in a CSV file:  
1] RTL netlist 2] Library files for fast and slow scenarios. 3] SDC constraint file  
It then outputs 1] Synthesized netlist 2] Pre-layout timing report.  
OpenTimer is used for generating timing reports and Yosys is used to perform synthesis.  
The UI is designed and maintained by Niharika Kummithi (Physical Design Engineer)  
For any queries or bugs, Please drop a mail to - niharika.kummithi98@gmail.com

USAGE: ./synui <csv\_file>

where <csv file> consists of 2 columns, below keyword being in 1st column and is Case Sensitive. Please request Niharika for sample csv file.

<Design Name> is the name of top level module.

<Output Directory> is the name of output directory where you want to dump synthesis script, synthesized netlist and timing reports.

<Netlist Directory> is the name of directory where all RTL netlist are present.

<Early Library Path> is the file path of the early cell library to be used for STA.

<Late Library Path> is file path of the late cell library to be used for STA.

<Constraints file> is csv file path of constraints to be used for STA.