



Fall Semester 2021-2022

ECE5030 - Scripting Languages for VLSI Design Automation

M.Tech VLSI Design

School of Electronics Engineering

Vellore Institute of Technology

Name: Shreyas S Bagi

Register Number: 21MVD0086

Slot: L3+L4

Lab Task 03

Report Filtering

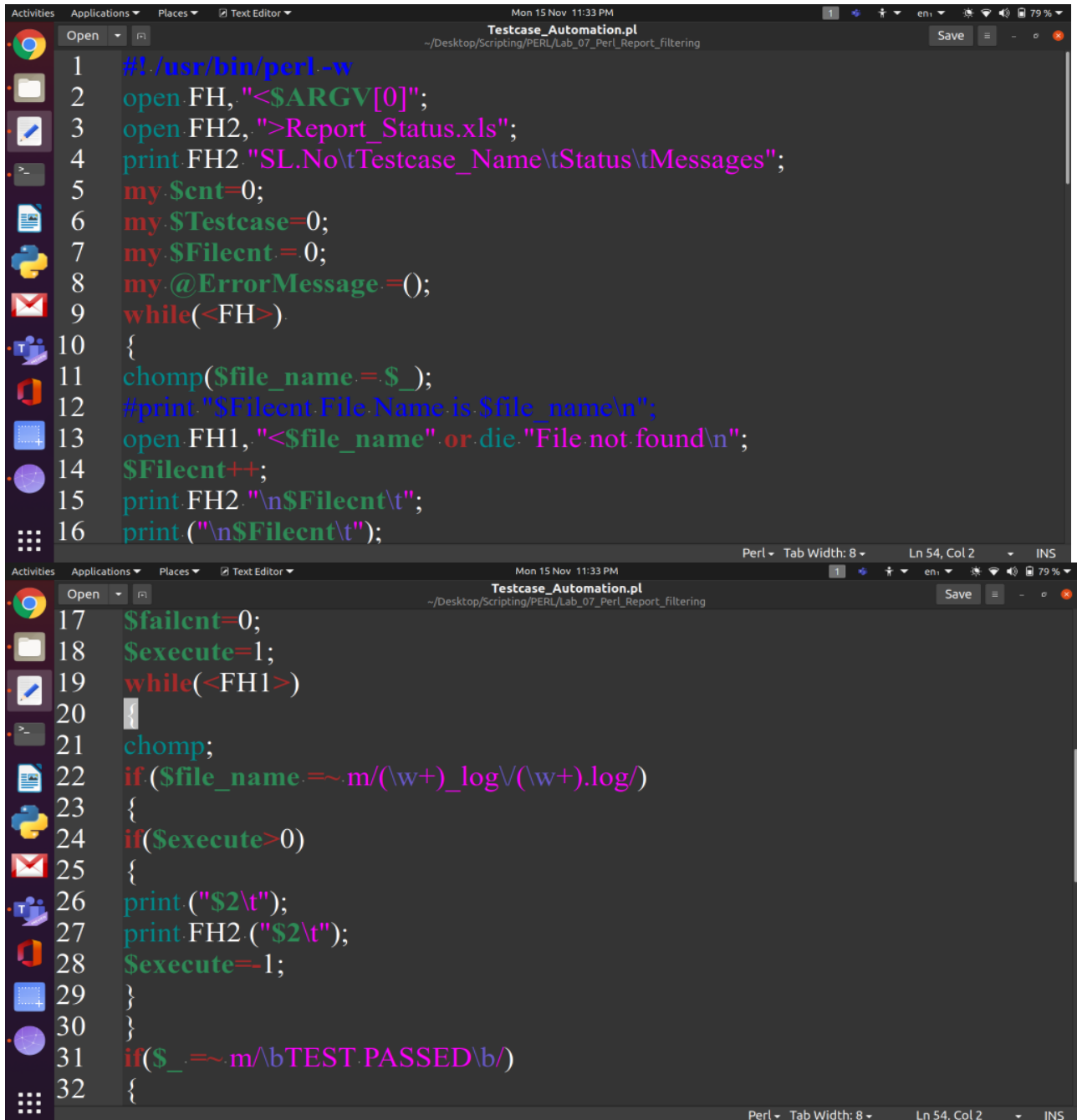
AIM : Write a Perl script that reads the given set of log files from different simulations and generates a consolidated report in .xls format which should contain the information as below. If the test is successful, the log file has the status as “TEST PASSED” and if the test is unsuccessful, then the log file has the status as “TEST FAILED”.

Source Code or PERL Script code:

```
#!/usr/bin/perl -w
open FH, "<$ARGV[0]";
open FH2, ">Report_Status.xls";
print FH2 "SL.No\tTestcase_Name\tStatus\tMessages";
my $cnt=0;
my $Testcase=0;
my $Filecnt = 0;
my @ErrorMessage =();
while(<FH>)
{
chomp($file_name = $_);
#print "$Filecnt File Name is $file_name\n";
open FH1, "<$file_name" or die "File not found\n";
$Filecnt++;
print FH2 "\n$Filecnt\t";
print ("\n$Filecnt\t");
$failcnt=0;
$execute=1;
while(<FH1>)
{
chomp;
if ($file_name =~ m/(\\w+)_log/(\\w+).log/)
{
if($execute>0)
{
print (" $2\t");
print FH2 (" $2\t");
$execute=-1;
}
}
if($_ =~ m/^bTEST PASSED\b/)
{
$Pass=$&;
print FH2 (" $Pass\n");
print (" $Pass\n");
}
}
```

```
elseif($_ =~ m/\bERROR\b/)  
{  
$failcnt++;  
$cnt++;  
$Fail=$&;  
$EMess=$';  
$EMess=~ s/-\s//;  
if($failcnt>0)  
{  
print FH2 (" $Fail");  
print (" $Fail");  
$failcnt= -5;  
print ("\t");  
}  
print FH2 ("\t$EMess\n\t\t");  
print ("\t$EMess\n\t\t");  
}  
}  
}  
close FH1;  
close FH2;  
close FH;
```

Screenshots of Perl Script:

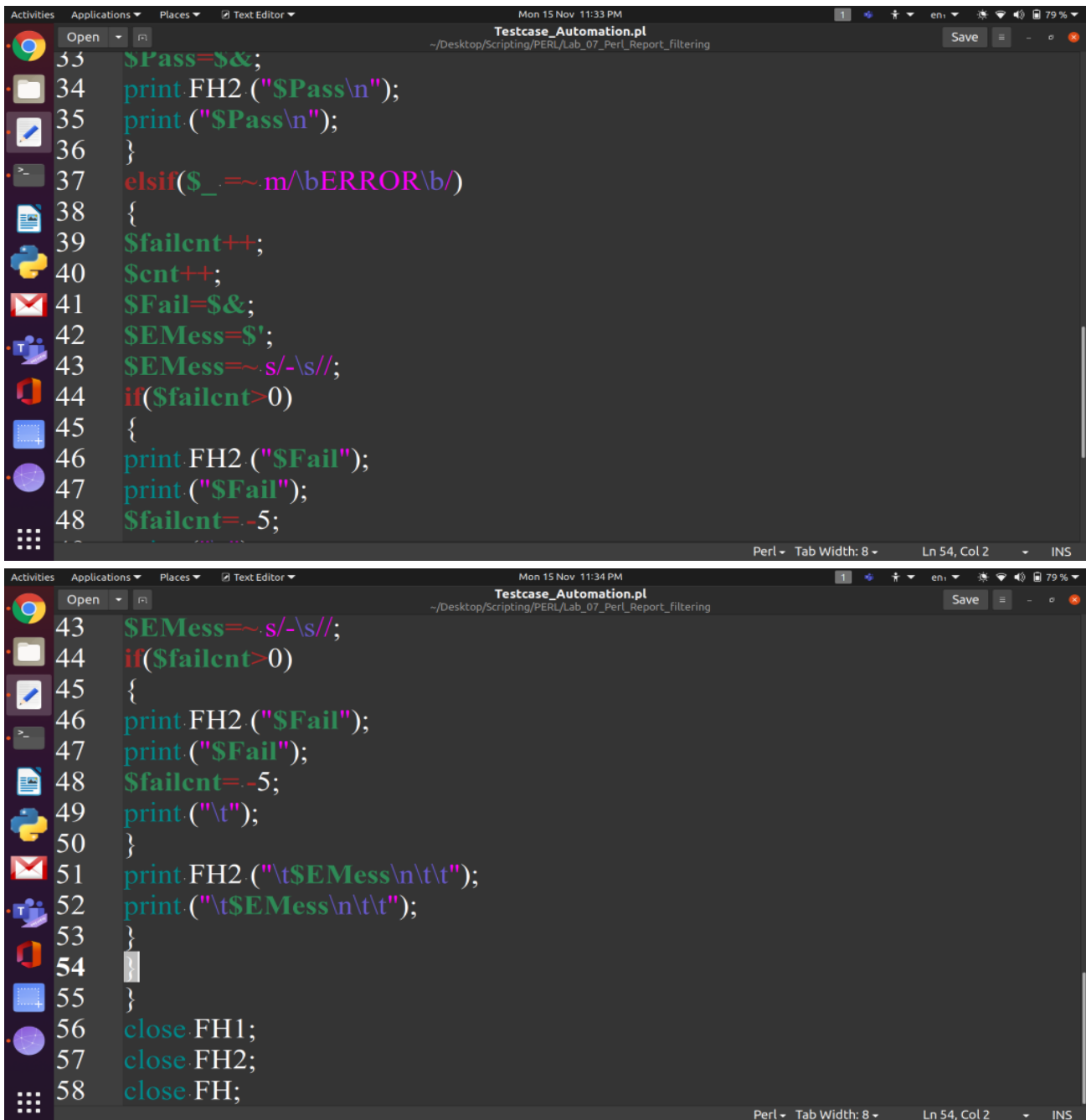


```

1  #!/usr/bin/perl -w
2  open FH, "<$ARGV[0]";
3  open FH2, ">Report_Status.xls";
4  print FH2 "SL.No\tTestcase_Name\tStatus\tMessages";
5  my $scnt=0;
6  my $Testcase=0;
7  my $Filecnt=0;
8  my @ErrorMessage=();
9  while(<FH>){
10 {
11   chomp($file_name=$_);
12   #print "Filecnt File Name is $file_name\n";
13   open FH1, "<$file_name" or die "File not found\n";
14   $Filecnt++;
15   print FH2 "\n$Filecnt\t";
16   print("\n$Filecnt\t");
17 $failcnt=0;
18 $sexecute=1;
19 while(<FH1>){
20 {
21   chomp;
22   if($file_name =~ m/(\w+) _log\(\w+)\.log/)
23   {
24     if($sexecute>0)
25     {
26       print("$2\t");
27       print FH2 (" $2\t");
28       $sexecute-=1;
29     }
30   }
31   if($_ =~ m/^bTEST PASSED\b/)
32   {

```

Figure 1.1 In this screenshot shows the Perl Script for Filtering of Log File and Reading the log files.



```
33 $Pass=$&;
34 print FH2 (" $Pass\n");
35 print (" $Pass\n");
36 }
37 elsif($_ =~ m/bERROR\b/)
38 {
39     $failcnt++;
40     $cnt++;
41     $Fail=$&;
42     $EMess=$';
43     $EMess =~ s/-\s//;
44     if($failcnt>0)
45     {
46         print FH2 (" $Fail");
47         print (" $Fail");
48         $failcnt = -5;
43 $EMess =~ s/-\s//;
44 if($failcnt>0)
45 {
46     print FH2 (" $Fail");
47     print (" $Fail");
48     $failcnt = -5;
49     print ("\t");
50 }
51 print FH2 ("\t$EMess\n\t\t");
52 print ("\t$EMess\n\t\t");
53 }
54
55 }
56 close FH1;
57 close FH2;
58 close FH;
```

Figure 1.2 In this screenshot shows the Perl Script for Filtering of Test Pass Case and Test Failed Case.

Output Screenshots

```

melon@melon-HP-Pavilion-Notebook: ~/Desktop/Scripting/PERL/Lab_07_Perl_Report_filtering
melon@melon-HP-Pavilion-Notebook:~/Desktop/Scripting/PERL/Lab_07_Perl_Report_filtering$ ./Testcase_Automation.pl list.txt
1      test1  TEST PASSED
2      test2  ERROR          ABC RX BFM : DATA NOT RECEIVED SUCCESSFULLY
3      test3  ERROR          ABC RX BFM : PACKED USER DEFINED DATA TYPE-6 PACKET IS CORRUPTED
                        ABC RX BFM : PACKED USER DEFINED DATA TYPE-6 PACKET IS CORRUPTED
4      test4  TEST PASSED
5      test5  TEST PASSED
6      test6  ERROR          AHB MASTER : READ TO INVALID ADDRESS 0000007c REGISTER
                        AHB MASTER : ERR RESPONSE IS RECEIVED
                        AHB MASTER : READ TO INVALID ADDRESS 0000008e REGISTER
melon@melon-HP-Pavilion-Notebook:~/Desktop/Scripting/PERL/Lab_07_Perl_Report_filtering$

```

Figure 1.3 In this screenshot shows the output of Perl script written for Report Filtering. The terminal window shows the filename and shows status of executed log file.

```

melon@melon-HP-Pavilion-Notebook: ~/Desktop/Scripting/PERL/Lab_07_Perl_Report_filtering
melon@melon-HP-Pavilion-Notebook:~/Desktop/Scripting/PERL/Lab_07_Perl_Report_filtering$ ./Testcase_Automation.pl list.txt
1      test1  TEST PASSED
2      test2  ERROR          ABC RX BFM : DATA NOT RECEIVED SUCCESSFULLY
3      test3  ERROR          ABC RX BFM : PACKED USER DEFINED DATA TYPE-6 PACKET IS CORRUPTED
                        ABC RX BFM : PACKED USER DEFINED DATA TYPE-6 PACKET IS CORRUPTED
4      test4  TEST PASSED
5      test5  TEST PASSED
6      test6  ERROR          AHB MASTER : READ TO INVALID ADDRESS 0000007c REGISTER
                        AHB MASTER : ERR RESPONSE IS RECEIVED
                        AHB MASTER : READ TO INVALID ADDRESS 0000008e REGISTER
melon@melon-HP-Pavilion-Notebook:~/Desktop/Scripting/PERL/Lab_07_Perl_Report_filtering$

```

SL.No	Testcase_Name	Status	Messages
1	test1	TEST PASSED	
2	test2	ERROR	ABC RX BFM : DATA NOT RECEIVED SUCCESSFULLY
3	test3	ERROR	ABC RX BFM : PACKED USER DEFINED DATA TYPE-6 PACKET IS CORRUPTED
4	test4	TEST PASSED	

Figure 1.4 In this screenshot shows the output of Perl script written in .xls format and opening the file using Terminal window.

SL.No	Testcase_Name	Status	Messages
1	test1	TEST PASSED	
2	test2	ERROR	ABC RX BFM : DATA NOT RECEIVED SUCCESSFULLY
3	test3	ERROR	ABC RX BFM : PACKED USER DEFINED DATA TYPE-6 PACKET IS CORROUPTED ABC RX BFM : PACKED USER DEFINED DATA TYPE-6 PACKET IS CORROUPTED
4	test4	TEST PASSED	
5	test5	TEST PASSED	
6	test6	ERROR	AHB MASTER : READ TO INVALID ADDRESS 0000007c REGISTER AHB MASTER : ERR RESPONSE IS RECEIVED AHB MASTER : READ TO INVALID ADDRESS 0000008e REGISTER

Figure 1.5 In this screenshot shows the output of Perl script written in .xls format. The detailed report of Log files and segregating with filename, Status and With Messages.

Inference:

1. Writing PERL Script and how to execute it in Terminal window.
2. Get familiarised with Scalar Data, Arrays and List Data, Control Structure, Hashes syntax and using in the script.
3. The Report Filtering of Log files generated from the Simulation of Verilog Files.
4. Automation of VLSI Verification. It reduces the time of designer to verify each and every case. The designer can give some error check (like how we have CRC Error checking) or self-checking codes. At the end just observe the .xls format file and see which testcase has failed.