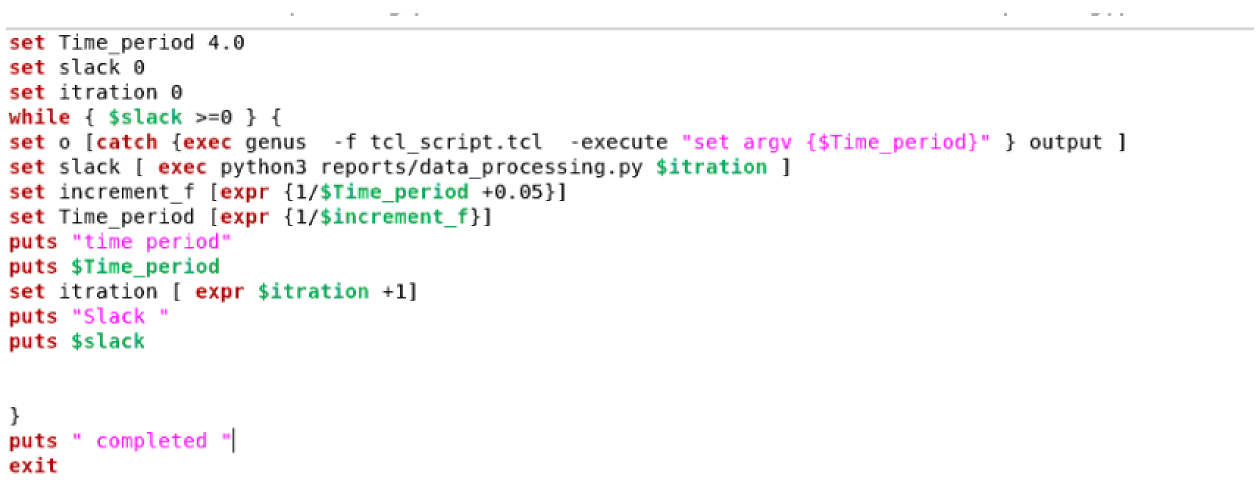
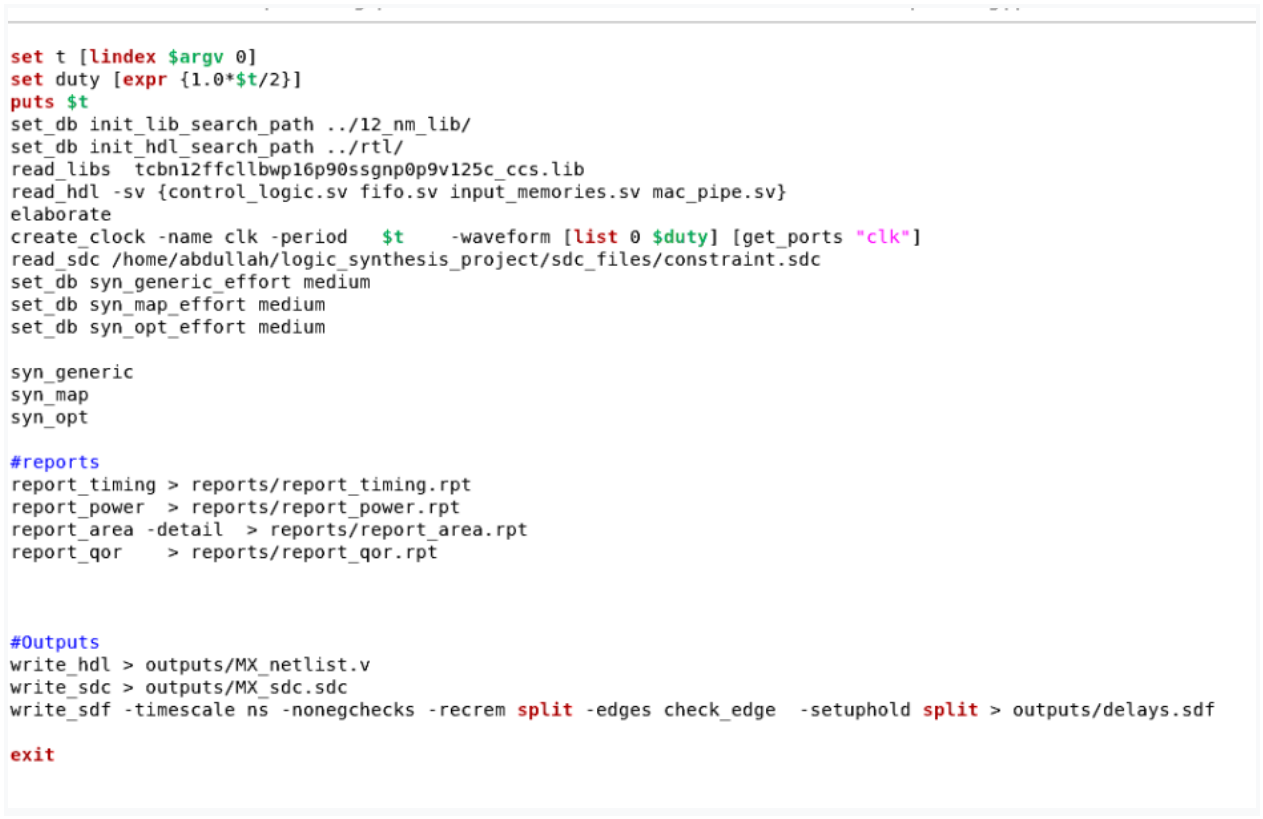
Design a Tcl script which run multiple synthesis

This tcl script invoked genus script with frequency as argument



```
set Time_period 4.0
set slack 0
set itration 0
while { $slack >=0 } {
set o [catch {exec genus  -f tcl_script.tcl  -execute "set argv {$Time_period}" } output ]
set slack [ exec python3 reports/data_processing.py $itration ]
set increment_f [expr {1/$Time_period +0.05}]
set Time_period [expr {1/$increment_f}]
puts "time period"
puts $Time_period
set itration [ expr $itration +1]
puts "Slack "
puts $slack


}
puts " completed "
exit
```

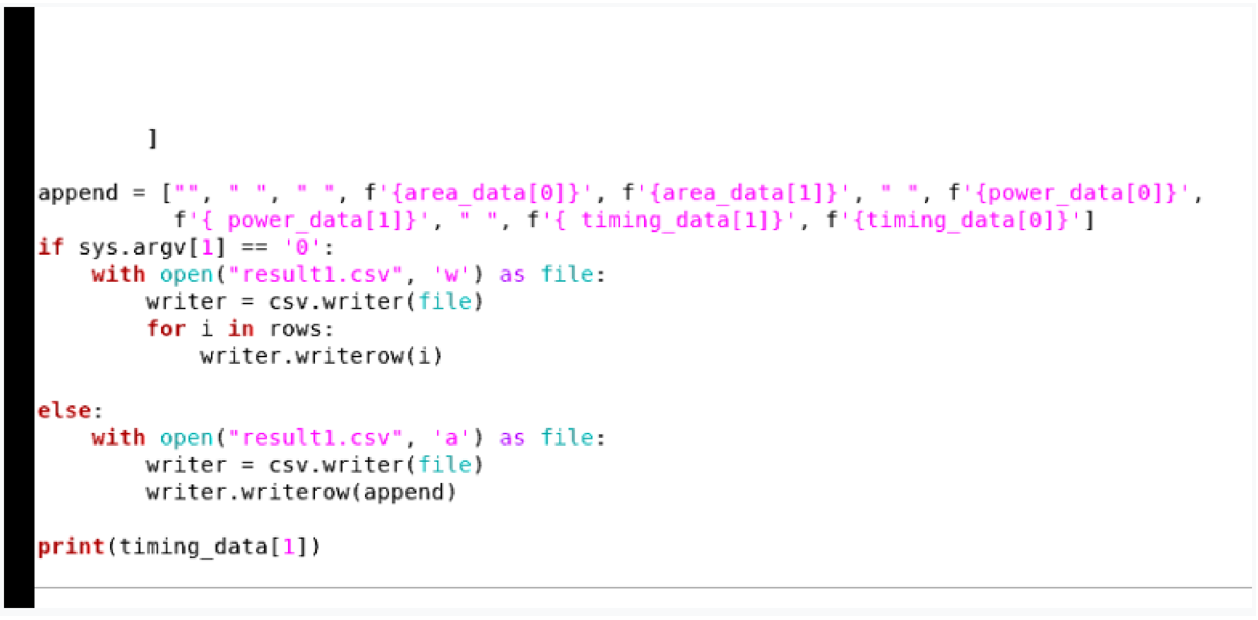
Tcl scrip for synthesis



```
set t [lindex $argv 0]
set duty [expr {1.0*$t/2}]
puts $t
set_db init_lib_search_path ../12_nm_lib/
set_db init_hdl_search_path ../rtl/
read_libs  tcbn12ffcllbwp16p90ssgnp0p9v125c_ccs.lib
read_hdl -sv {control_logic.sv fifo.sv input_memories.sv mac_pipe.sv}
elaborate
create_clock -name clk -period   $t    -waveform [list 0 $duty] [get_ports "clk"]
read_sdc /home/abdullah/logic_synthesis_project/sdc_files/constraint.sdc
set_db syn_generic_effort medium
set_db syn_map_effort medium
set_db syn_opt_effort medium

syn_generic
syn_map
syn_opt

#reports
report_timing > reports/report_timing.rpt
report_power  > reports/report_power.rpt
report_area -detail  > reports/report_area.rpt
report_qor    > reports/report_qor.rpt



#Outputs
write_hdl > outputs/MX_netlist.v
write_sdc > outputs/MX_sdc.sdc
write_sdf -timescale ns -nonegchecks -recrem split -edges check_edge  -setuphold split > outputs/delays.sdf

exit
```

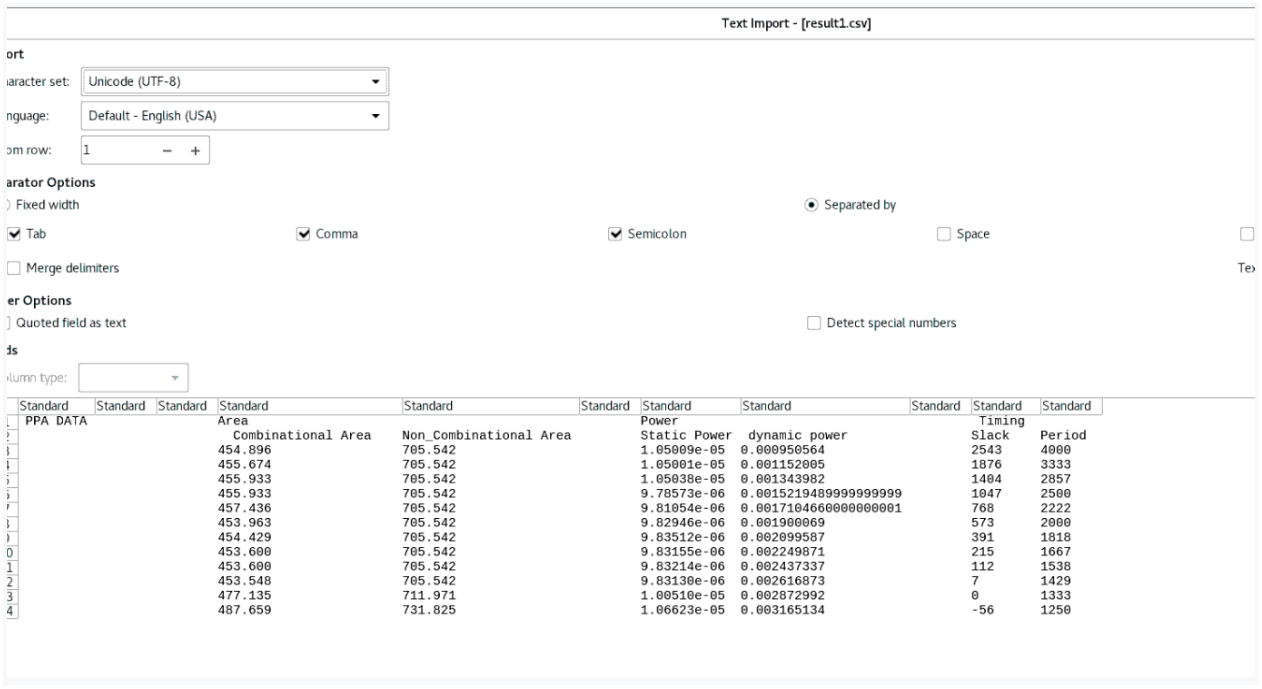
Python script for data extraction

Regex logic same as previous this is updated script for task 3



```
        ]

append = ["", " ", " ", f'{area_data[0]}', f'{area_data[1]}', " ", f'{power_data[0]}',
          f'{ power_data[1]}', " ", f'{ timing_data[1]}', f'{timing_data[0]}']
if sys.argv[1] == '0':
    with open("result1.csv", 'w') as file:
        writer = csv.writer(file)
        for i in rows:
            writer.writerow(i)

else:
    with open("result1.csv", 'a') as file:
        writer = csv.writer(file)
        writer.writerow(append)

print(timing_data[1])
```

Table containing Slack, area and power numbers



Text Import - [result1.csv]

ort

aracter set: Unicode (UTF-8)

nguage: Default - English (USA)

om row: 1

arator Options

Fixed width [✓ Separated by]

[✓ Tab] [✓ Comma] [✓ Semicolon] Space

Merge delimiters

Te

er Options

Quoted field as text

Detect special numbers

ds

lumn type:

Standard	Standard	Standard	Standard	Standard	Standard	Standard	Standard	Standard	Standard	Standard
PPA DATA			Area			Power			Timing	
			Combinational Area	Non_Combinational Area		Static Power	dynamic power		Slack	Period
			454.896	705.542		1.05009e-05	0.000950564		2543	4000
			455.674	705.542		1.05001e-05	0.001152005		1876	3333
			455.933	705.542		1.05038e-05	0.001343982		1404	2857
			455.933	705.542		9.78573e-06	0.0015219489999999999		1047	2500
			457.436	705.542		9.81054e-06	0.0017104660000000001		768	2222
			453.963	705.542		9.82946e-06	0.001900069		573	2000
			454.429	705.542		9.83512e-06	0.002099587		391	1818
			453.600	705.542		9.83155e-06	0.002249871		215	1667
			453.600	705.542		9.83214e-06	0.002437337		112	1538
			453.548	705.542		9.83130e-06	0.002616873		7	1429
			477.135	711.971		1.00510e-05	0.002872992		0	1333
			487.659	731.825		1.06623e-05	0.003165134		-56	1250