Cory Snooks
Aaron Pederson
EE 465 Lab 5
10/1/2015

Synthesis Constraints

Introduction
This lab consisted of synthesis of the verilog code we wrote in lab 1 and synthesized in lab 4. Unlike lab 4, we modified various constraints for the synthesis and observed changes in the resulting performance of the synthesized circuit.

Step 1:

First, we deleted all the lines after read_sdc ./scripts/design.sdc to get the resulting file:

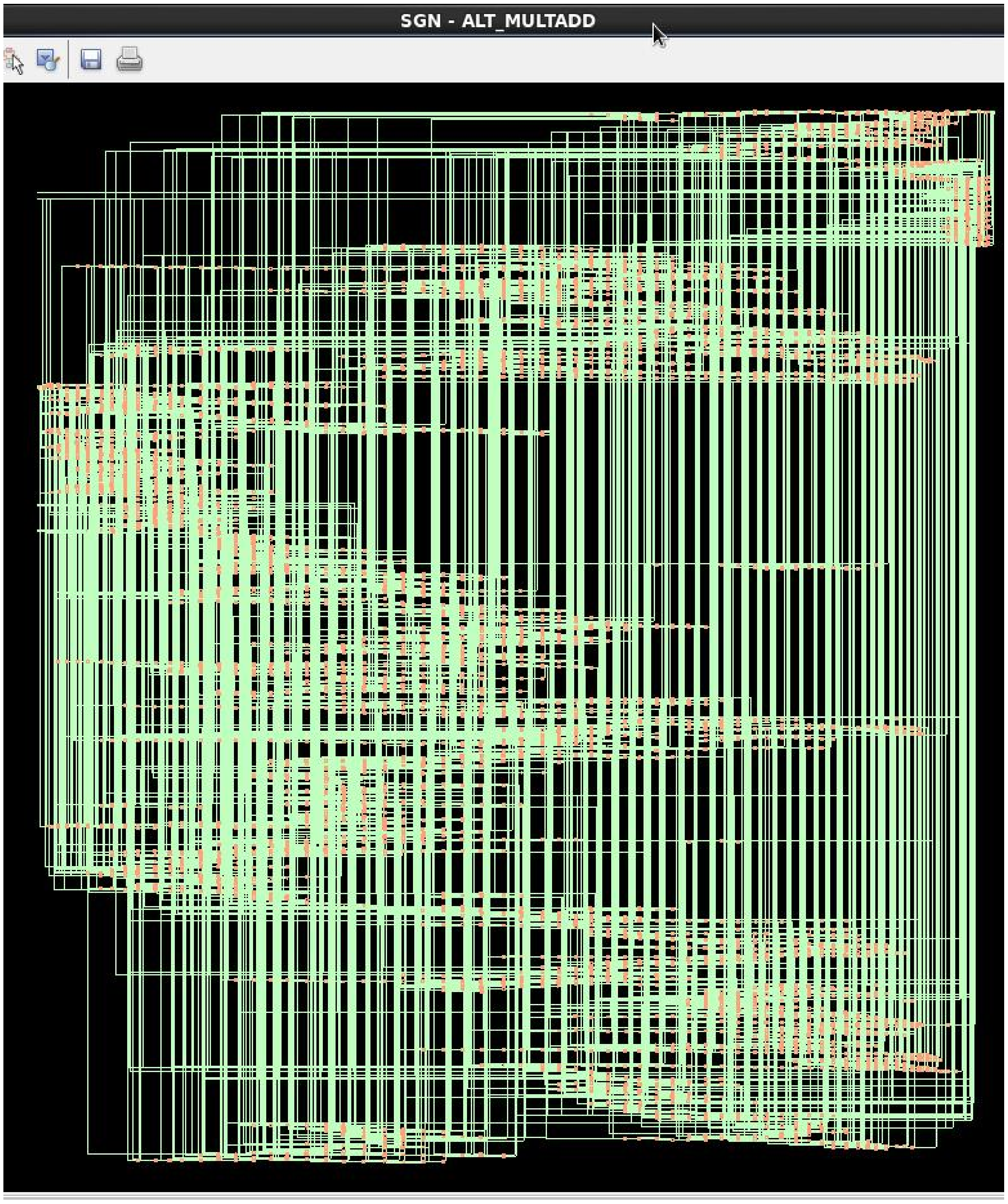
```
set sdc_version 1.4

create_clock -period 0.8 -waveform {0 0.4 } [get_ports {iCLK}]
set_input_delay 0.001 -max -clock "iCLK" [get_ports {iA0}]
set_input_delay 0.001 -max -clock "iCLK" [get_ports {iB0}]
set_input_delay 0.001 -max -clock "iCLK" [get_ports {iA1}]
set_input_delay 0.001 -max -clock "iCLK" [get_ports {iB1}]
set_attribute lp_insert_clock_gating true
```

Step 2: Run run_synth.tcl

When we run the run_synth.tcl script, nothing shows up in the GUI because we deleted everything below "elaborate"

And got the following synthesized circuit diagram:



Step 3/4/5: Type in commands, change synthesize effort & output reports

We typed in the following commands to the terminal:

```
dc::set_time_unit -picoseconds
dc::set_load_unit -picofarads
define_clock -period 500 -name clk [dc::get_ports {iCLK}] -rise 50 -fall 50
set_attribute clock_network_late_latency 1 clk
set_attribute clock_source_early_latency 1.5 clk
set_attribute clock_setup_uncertainty {100 50} clk
set_attribute slew {100 110 110 120} [find / -clock clk]
report clocks
external_delay -clock clk -input 200 -name in_delay /designs/ALT_MULTADD/ports_in/*
external_delay -clock clk -output 200 -name out_delay /designs/ALT_MULTADD/ports_out/*
set_attribute external_driver [find [find / -libcell MUX2ND2] -libpin ZN]
/designs/ALT_MULTADD/ports_in/*
set_attribute external_pin_cap 2 /designs/ALT_MULTADD/ports_out/*
set_attribute max_fanout 10 /designs/*
synthesize -to_mapped -effort high
```

Then we wrote the report names to files with these commands:

```
report power > ${OUTPUT_DIR}/${RUN}_${FILENAME}_power.rpt
generate_reports -outdir ${OUTPUT_DIR} -tag ${RUN}_${FILENAME}
write -mapped > ${OUTPUT_DIR}/mapped/${RUN}_${FILENAME}/jpeg_mapped.v
write_script > ${OUTPUT_DIR}/mapped/${RUN}_${FILENAME}/jpeg_mapped.g
write_sdc > ${OUTPUT_DIR}/mapped/${RUN}_${FILENAME}/jpeg_mapped.sdc
ls * -attribute > ${OUTPUT_DIR}/${RUN}_${FILENAME}_attributes.txt
```

I found a command (generate_reports) that generates all the reports we need except the power report. I also made a script that makes the report with filenames corresponding to the synthesis run.

Here is the area report for the initial run with the constraints given above in code 1:

```
===========================================================
  Generated by:           Encounter(R) RTL Compiler v12.10-s012_1
  Generated on:           Oct 23 2015  05:03:09 pm
  Module:                 ALT_MULTADD
  Technology library:     tcbn65gpluswc 121
  Operating conditions:   WCCOM (balanced_tree)
  Wireload mode:          segmented
  Area mode:              timing library
============================================================


  Instance   Cells  Cell Area  Net Area  Total Area    Wireload
---------------------------------------------------------------------
ALT_MULTADD   3885       8726         0       8726 ZeroWireload (S)

 (S) = wireload was automatically selected
```

The timing report:

```
               Pin                        Type       Fanout Load Slew Delay Arrival
                                                            (fF) (ps)  (ps)   (ps)
--------------------------------------------------------------------------------
--
(clock clk)                          launch                                   250
R
                                     latency                         +1       251
R
A1_reg[2]/CP                                                    110            251
R
...
...
g16231/A2                                                            +0      1941
g16231/ZN                            ND2D2            1  1.9    21   +23     1964
R
g234/A1                                                              +0      1964
g234/ZN                              CKND2D2          1  1.2    15   +16     1981
F
g16230/A1                                                            +0      1981
g16230/ZN                            ND2D1            1  1.1    24   +18     1998
R
oR_reg[16]/D                         DFQD1                           +0      1998
oR_reg[16]/CP                        setup                      110   +6     2004
R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
(clock clk)                          capture                                  750
R
                                     latency                         +1       751
R
                                     uncertainty                   -100       651
R
--------------------------------------------------------------------------------
--
Timing slack :   -1353ps (TIMING VIOLATION)
Start-point  : A1_reg[2]/CP
End-point    : oR_reg[16]/D
```

And our power report:

```
                    Leakage    Dynamic     Total
  Instance  Cells Power(nW)  Power(nW)  Power(nW)
-------------------------------------------------
ALT_MULTADD  3885 86537.986 835617.537 922155.523
```

Step 6: Modify variables

To make this part of the lab easier, I made a .tcl script (replacing run_synth.tcl) which would perform all the synthesis for this part of the lab at once. I had to do a little searching and testing to find out the commands that would allow me to do this. I made two other script files that would act as function modules to perform tasks that would be done repeatedly using the ‘source’ command. I’ll explain these below...

lab5script.tcl:

Here is the main script. I made a script (initStuff) that will set all the initial constraints to be compared. Which is called every synthesis and when a value is being changed, I just set those values after the initial values were set. I removed a few lines to cut down on the space just to give an idea of what my script does.

```
source ./scripts/initStuff.tcl

set RUN 18
set FILENAME initial

source ./scripts/writeReports.tcl

set FILENAME clkPd1000
source ./scripts/initStuff.tcl
define_clock -period 1000 -name clk [dc::get_ports {iCLK}] -rise 50 -fall 50
source ./scripts/writeReports.tcl

set FILENAME clkPd200
source ./scripts/initStuff.tcl
define_clock -period 200 -name clk [dc::get_ports {iCLK}] -rise 50 -fall 50
source ./scripts/writeReports.tcl

...
...

set FILENAME pwrEffortHigh
source ./scripts/initStuff.tcl
set_attribute lp_power_unit {nW}
set_attribute power_optimization_effort high
set_attribute max_leakage_power 10000 /designs/ALT_MULTADD
source ./scripts/writeReports.tcl

set FILENAME pwrEffortLow
source ./scripts/initStuff.tcl
set_attribute lp_power_unit {nW}
set_attribute power_optimization_effort low
set_attribute max_leakage_power 1000000 /designs/ALT_MULTADD
source ./scripts/writeReports.tcl
```

initStuff.tcl:

Here is where I set all the initial constraints for the syntheses. All values are the ones given in the lab instructions.

```
set OUTPUT_DIR ./out_dir
if { ![file exists ${OUTPUT_DIR}] }  { sh mkdir ${OUTPUT_DIR} }

set_attribute lib_search_path ../libdir
set_attribute gui_sv_threshold 10000

## This defines the libraries to use
set_attribute library {tcbn65gpluswc.lib}

load -v2001 ../rtl/ALT_MULTADD.v

elaborate


dc::set_time_unit -picoseconds
dc::set_load_unit -picofarads
define_clock -period 500 -name clk [dc::get_ports {iCLK}] -rise 50 -fall 50
set_attribute clock_network_late_latency 1 clk
set_attribute clock_source_early_latency 1.5 clk
set_attribute clock_setup_uncertainty {100 50} clk
set_attribute slew {100 110 110 120} [find / -clock clk]
report clocks
external_delay -clock clk -input 200 -name in_delay /designs/ALT_MULTADD/ports_in/*
external_delay -clock clk -output 200 -name out_delay /designs/ALT_MULTADD/ports_out/*
set_attribute external_driver [find [find / -libcell MUX2ND2] -libpin ZN]
/designs/ALT_MULTADD/ports_in/*
set_attribute external_pin_cap 2 /designs/ALT_MULTADD/ports_out/*
set_attribute max_fanout 10 /designs/*
```

writeReports.tcl:

In the writeReports script, I synthesized the design to simplify the main script before exporting the power and other reports to files in the output directory. Created an increment variable named ‘RUN’ to show on each file which run it was so it would be easy to see in the file browser.

```
synthesize -to_mapped -effort high
report power > ${OUTPUT_DIR}/${RUN}_${FILENAME}_power.rpt
generate_reports -outdir ${OUTPUT_DIR} -tag ${RUN}_${FILENAME}
write -mapped > ${OUTPUT_DIR}/mapped/${RUN}_${FILENAME}/jpeg_mapped.v
write_script > ${OUTPUT_DIR}/mapped/${RUN}_${FILENAME}/jpeg_mapped.g
write_sdc > ${OUTPUT_DIR}/mapped/${RUN}_${FILENAME}/jpeg_mapped.sdc
ls * -attribute > ${OUTPUT_DIR}/${RUN}_${FILENAME}_attributes.txt
rm /designs/ALT_MULTADD/
incr RUN
```

The `rm /designs/ALT_MULTADD/` command deletes the current design from the environment to allow for a new one to be loaded and synthesized without being affected by the previous synthesis data.

Step 6: Change synthesis constraints & compare

All of these syntheses are in comparison to these initial constraints and results.

Initial constraints:

```
define_clock -period 500 -name clk [dc::get_ports {iCLK}] -rise 50 -fall 50
set_attribute clock_network_late_latency 1 clk
set_attribute clock_source_early_latency 1.5 clk
set_attribute clock_setup_uncertainty {100 50} clk
set_attribute slew {100 110 110 120} [find / -clock clk]
report clocks
external_delay -clock clk -input 200 -name in_delay /designs/ALT_MULTADD/ports_in/*
external_delay -clock clk -output 200 -name out_delay /designs/ALT_MULTADD/ports_out/*
set_attribute external_driver [find [find / -libcell MUX2ND2] -libpin ZN]
/designs/ALT_MULTADD/ports_in/*
set_attribute external_pin_cap 2 /designs/ALT_MULTADD/ports_out/*
set_attribute max_fanout 10 /designs/*
```

Initial Reports:

Initial Timing:

```
                Pin                      Type       Fanout Load Slew Delay Arrival
                                                           (fF) (ps)  (ps)   (ps)
-----------------------------------------------------------------------------------
(clock clk)                            launch                                 250 R
                                       latency                         +1     251 R
A1_reg[2]/CP                                                     110          251 R
...
...
oR_reg[16]/CP                          setup                     110   +6    2004 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)                            capture                                750 R
                                       latency                         +1     751 R
                                       uncertainty                   -100     651 R
-----------------------------------------------------------------------------------
Timing slack :   -1353ps (TIMING VIOLATION)
Start-point  : A1_reg[2]/CP
End-point    : oR_reg[16]/D
```

Initial Area:

```
  Instance    Cells  Cell Area  Net Area  Total Area    Wireload
--------------------------------------------------------------------
ALT_MULTADD   3885       8726         0        8726 ZeroWireload (S)
```

Initial Power:

```
                        Leakage     Dynamic      Total
  Instance   Cells Power(nW)  Power(nW)   Power(nW)
--------------------------------------------------
ALT_MULTADD   3885 86537.986 835617.537 922155.523
```

For all the following synthesis, I will only show the script command that is different from the initial script settings.

Clock Period Increased:

```
set FILENAME clkPd1000
source ./scripts/initStuff.tcl
define_clock -period 1000 -name clk [dc::get_ports {iCLK}] -rise 50 -fall 50
source ./scripts/writeReports.tcl
```

Here, I changed the clock period from 500 to 1000. This means the synthesis will run assuming the input clock period is 1000 picoseconds and will generate the reports according to that input.

Increased Clock Period Reports:

Timing:

```
    Pin            Type       Fanout Load Slew Delay Arrival
                                     (fF) (ps)  (ps)   (ps)
--------------------------------------------------------------
(clock clk)     launch                                  500 R
                latency                          +1     501 R
A1_reg[1]/CP                              110           501 R
...
...
oR_reg[16]/CP   setup                     110    +6    2255 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)     capture                                1500 R
                latency                          +1    1501 R
                uncertainty                    -100    1401 R
--------------------------------------------------------------
Timing slack :    -854ps (TIMING VIOLATION)
Start-point  : A1_reg[1]/CP
End-point    : oR_reg[16]/D
```

As we can see, the slack time increased. This makes sense because the constraint for the clock period was increased by 500 which is exactly how much the timing slack increased. The timing slack is just the amount of extra time the system has from when it gets the result from one clock cycle to when it needs to compute another result (on the worst path). In this case, it is negative and that means it needs more time to compute the result to be sure there will be no errors.

Area:

```
  Instance   Cells  Cell Area  Net Area  Total Area    Wireload
---------------------------------------------------------------------
ALT_MULTADD   3893       8731         0        8731 ZeroWireload (S)
```

The area changed very little.

Power:

```
                      Leakage    Dynamic     Total
  Instance   Cells Power(nW)  Power(nW)  Power(nW)
--------------------------------------------------
ALT_MULTADD   3893 86080.614 418756.462 504837.076
```

The leakage power decreased possibly due to the decreased by 417318.447nW possibly due to the less aggressive clock period.

Clock Period Decreased:

```
set FILENAME clkPd200
source ./scripts/initStuff.tcl
define_clock -period 200 -name clk [dc::get_ports {iCLK}] -rise 50 -fall 50
source ./scripts/writeReports.tcl
```

Here, I changed the clock period from 500 to 200. This means the synthesis will run assuming the input clock period is 200 picoseconds and will generate the reports according to that input.

Decreased Clock Period Reports:

Timing:

```
       Pin              Type       Fanout Load Slew Delay Arrival
                                          (fF) (ps)  (ps)   (ps)
-------------------------------------------------------------------
(clock clk)           launch                                 100 R
                      latency                         +1     101 R
A0_reg[4]/CP                                   110           101 R
...
...
oR_reg[16]/CP         setup                    110    +6    1857 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)           capture                                300 R
                      latency                         +1     301 R
                      uncertainty                   -100     201 R
-------------------------------------------------------------------
Timing slack :   -1656ps (TIMING VIOLATION)
Start-point  : A0_reg[4]/CP
End-point    : oR_reg[16]/D
```

Again, as we can see, the slack time decreased. This makes sense because the constraint for the clock period was decreased by 300 which is exactly how much the timing slack decreased. The timing slack is just the amount of extra time the system has from when it gets the result from one clock cycle to when it needs to compute another result (on the worst path). In this case, it is negative and that means it needs more time to compute the result to be sure there will be no errors.

Area:

```
  Instance    Cells  Cell Area  Net Area  Total Area    Wireload
---------------------------------------------------------------------
ALT_MULTADD    3878       8748         0        8748 ZeroWireload (S)
```

The area changed very little probably only because of the types of cells changing due to the more aggressive timing constraints.

Power:

```
                   Leakage    Dynamic      Total
  Instance   Cells Power(nW)  Power(nW)   Power(nW)
----------------------------------------------------
ALT_MULTADD   3878 86859.507 2075841.760 2162701.267
```

The leakage power increased by 1240545.744nW possibly due to the more aggressive clock period.

Clock Rise Time Decrease:

```
set FILENAME clkRise10
source ./scripts/initStuff.tcl
define_clock -period 500 -name clk [dc::get_ports {iCLK}] -rise 10 -fall 50
source ./scripts/writeReports.tcl
```

Here, I changed the rise time to 10. This means that the input clock will take less time on the positive edge. It will take only 10 picoseconds to rise to Vdd.

Decreased Clock Rise Reports:

Timing:

```
               Pin                       Type       Fanout Load Slew Delay Arrival
                                                           (fF) (ps)  (ps)   (ps)
-----------------------------------------------------------------------------------
(clock clk)                            launch                                 50 R
                                       latency                        +1      51 R
A1_reg[2]/CP                                                   110            51 R
...
...
oR_reg[16]/CP                          setup                   110    +6    1804 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)                            capture                               550 R
                                       latency                        +1     551 R
                                       uncertainty                  -100     451 R
-----------------------------------------------------------------------------------
Timing slack :   -1353ps (TIMING VIOLATION)
Start-point  : A1_reg[2]/CP
End-point    : oR_reg[16]/D
```

As we can see, the timing slack did not change at all from the initial. The arrival times, however are much smaller. This suggests that the decreased rise time causes the rise time of the outputs to all registers and gates with clock inputs to be faster. The unchanged timing slack puzzles me though. It doesn’t make sense that the setup time is much lower but the timing slack is still the same as initially.

Area:

```
  Instance    Cells  Cell Area  Net Area  Total Area    Wireload
---------------------------------------------------------------------
ALT_MULTADD    3885       8726         0        8726 ZeroWireload (S)
```

The area changed very little probably only because of the types of cells changing due to the changed timing constraints.

Power:

```
                          Leakage     Dynamic       Total
  Instance   Cells Power(nW)  Power(nW)    Power(nW)
----------------------------------------------------
ALT_MULTADD   3885 90032.626 7657650.233 7747682.859
```

The leakage power increased by almost an order of magnitude. Most of this comes from the dynamic power this tells me that the faster rise time of the input causes more power consumption.

Clock Fall Time Decrease:

```
set FILENAME clkFall10
source ./scripts/initStuff.tcl
define_clock -period 500 -name clk [dc::get_ports {iCLK}] -rise 50 -fall 10
source ./scripts/writeReports.tcl
```

Here, I changed the fall time to 10. This means that the input clock will take less time on the negative edge. It will take only 10 picoseconds to fall.

Decreased Clock Fall Reports:

Timing:

```
                Pin                        Type       Fanout Load Slew Delay Arrival
                                                             (fF) (ps)  (ps)   (ps)
------------------------------------------------------------------------------------
(clock clk)                          launch                                    250 R
                                     latency                            +1     251 R
A1_reg[2]/CP                                                      110          251 R
...
...
oR_reg[16]/CP                        setup                        110   +6    2004 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)                          capture                                   750 R
                                     latency                            +1     751 R
                                     uncertainty                      -100     651 R
------------------------------------------------------------------------------------
Timing slack :   -1353ps (TIMING VIOLATION)
Start-point  : A1_reg[2]/CP
End-point    : oR_reg[16]/D
```

As we can see, the timing did not change at all from the initial. This makes sense because the fall time of the clock is not that important because all of the registers are loaded on the positive edge of the clock only.

Area:

```
  Instance   Cells  Cell Area  Net Area  Total Area    Wireload
--------------------------------------------------------------------
ALT_MULTADD   3885       8726         0        8726 ZeroWireload (S)
```

The area changed very little probably only because of the types of cells changing due to the changed timing constraints.

Power:

```
                        Leakage     Dynamic       Total
  Instance   Cells Power(nW)  Power(nW)    Power(nW)
----------------------------------------------------------
ALT_MULTADD  3885 89950.816 7657650.233 7747601.049
```

The leakage power increased by almost an order of magnitude. Most of this comes from the dynamic power this tells me that the faster fall time of the input causes more power consumption.

Clock Net Late Latency Increase:

```
set FILENAME clkNetLateLaten2
source ./scripts/initStuff.tcl
set_attribute clock_network_late_latency 10 clk
source ./scripts/writeReports.tcl
```

Here, I changed the clock net late latency to 10. This will increase specified longest allowable time it takes for the clock signal to propagate from the input of the whole module to the input of the individual cells in the synthesized design.

Decreased Clock Net Late Latency Reports:

Timing:

```
    Pin             Type       Fanout Load Slew Delay Arrival
                                      (fF) (ps)  (ps)    (ps)
--------------------------------------------------------------
(clock clk)      launch                                  250 R
                 latency                          +10    260 R
A0_reg[4]/CP                                110          260 R
oR_reg[16]/CP    setup                      110    +5   1994 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)      capture                                 750 R
                 latency                          +10    760 R
                 uncertainty                     -100    660 R
--------------------------------------------------------------
Timing slack :   -1334ps (TIMING VIOLATION)
Start-point  : A0_reg[4]/CP
End-point    : oR_reg[16]/D
```

As we can see, the timing slack improved by 19ps. The latency increased by 10 as expected on the timing report. Due to increased latency.

Area:

```
  Instance    Cells  Cell Area  Net Area  Total Area    Wireload
-----------------------------------------------------------------
ALT_MULTADD    4096       8973         0        8973 ZeroWireload (S)
```

The area increased significantly.

Power:

```
                         Leakage     Dynamic      Total
  Instance   Cells Power(nW)  Power(nW)  Power(nW)
-------------------------------------------------
ALT_MULTADD  4096 88523.241 847947.661 936470.902
```

The leakage power increased by almost an order of magnitude. Most of this comes from the dynamic power this tells me that the faster fall time of the input causes more power consumption.

Clock Net Early Latency Increase:

```
set FILENAME clkSourceEarlyLaten2
source ./scripts/initStuff.tcl
set_attribute clock_source_early_latency 15 clk
source ./scripts/writeReports.tcl
```

Here, I changed the clock net early latency to 15. This will increase specified shortest time it takes for the clock signal to propagate from the input to any of the input of the cells in the synthesized design.

Decreased Clock Net Early Latency Reports:

Timing:

```
               Pin                      Type       Fanout Load Slew Delay Arrival
                                                          (fF) (ps)  (ps)   (ps)
-----------------------------------------------------------------------------------
(clock clk)                           launch                                 250 R
                                      latency                        +1     251 R
A1_reg[2]/CP                                                   110            251 R
...
...
oR_reg[16]/CP                         setup                    110     +6   2004 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)                           capture                                750 R
                                      latency                        +1     751 R
                                      uncertainty                  -100     651 R
-----------------------------------------------------------------------------------
Timing slack :   -1353ps (TIMING VIOLATION)
Start-point  : A1_reg[2]/CP
End-point    : oR_reg[16]/D
```

As we can see, the timing slack did not change from the initial. This is because the early latency is not a concern because the clock period for this circuit is already too fast.

Area:

```
  Instance    Cells  Cell Area  Net Area  Total Area    Wireload
------------------------------------------------------------------
ALT_MULTADD    3885       8726         0        8726 ZeroWireload (S)
```

The area did not change from the initial. This is probably again due to reasons stated above.

Power:

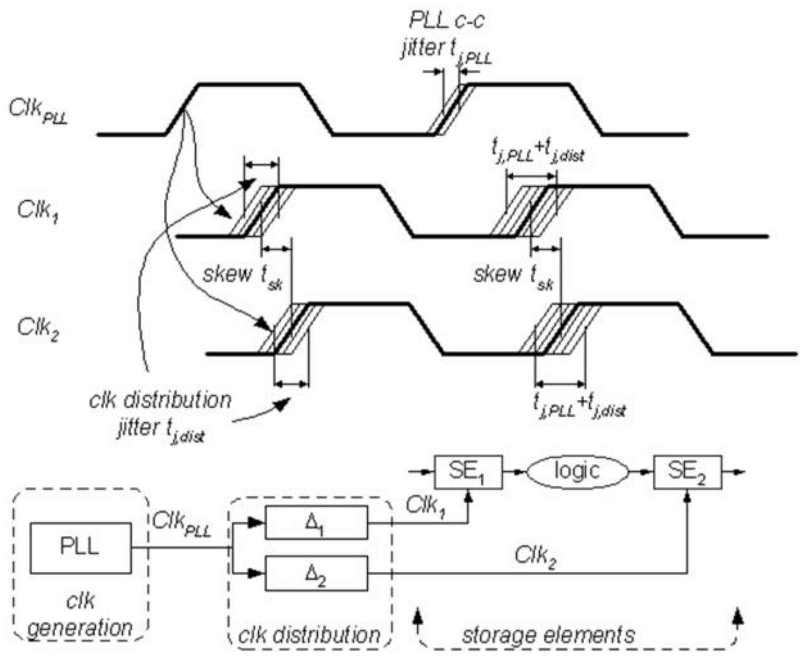
```
                        Leakage    Dynamic     Total
  Instance   Cells Power(nW)  Power(nW)  Power(nW)
--------------------------------------------------
ALT_MULTADD   3885 86537.986 835617.537 922155.523
```

The leakage does not change due to reasons stated above.

Clock Setup Uncertainty Increase:

```
set FILENAME clkSetupUncert1000
source ./scripts/initStuff.tcl
set_attribute clock_setup_uncertainty {1000 50} clk
source ./scripts/writeReports.tcl
```

Here, I changed the clock setup uncertainty to 1000. This will tell the synthesis the amount of uncertainty (skew) the clock signal should have at all register clock input pins. I increased it by a factor of 10 so there will be a large window of uncertainty for this synthesis. The image below shows clock skew.



Increased Clock Setup Uncertainty Reports:

Timing:

```
       Pin                Type       Fanout  Load Slew Delay Arrival
                                             (fF) (ps)  (ps)   (ps)
-------------------------------------------------------------------
(clock clk)           launch                                   250 R
                      latency                             +1   251 R
A0_reg[2]/CP                                      110          251 R
...
...
oR_reg[16]/CP         setup                       110     +6  2010 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)           capture                                  750 R
                      latency                             +1   751 R
                      uncertainty                      -1000  -249 R
-------------------------------------------------------------------
Timing slack :    -2259ps (TIMING VIOLATION)
Start-point  : A0_reg[2]/CP
End-point    : oR_reg[16]/D
```

The timing slack decreased significantly. This makes sense because the circuit must be designed so that there is enough time in-between stages to allow for that clock uncertainty to be accounted for. Slack is directly related to clock uncertainty.

Area:

```
  Instance    Cells  Cell Area  Net Area  Total Area    Wireload
-------------------------------------------------------------------
ALT_MULTADD   3883       8677         0        8677 ZeroWireload (S)
```

The area decreased slightly. I'm not quite sure why but it isn't that much less.

Power:

```
                      Leakage    Dynamic     Total
  Instance    Cells Power(nW)  Power(nW)  Power(nW)
--------------------------------------------------
ALT_MULTADD   3883 85386.775 836764.796 922151.571
```

The power did not change very much at all. There are slightly less cells so that is probably why. Most likely some buffer cells or something removed.

Clock Setup Uncertainty Increase(2):

```
set FILENAME clkSetupUncert500
source ./scripts/initStuff.tcl
set_attribute clock_setup_uncertainty {100 500} clk
source ./scripts/writeReports.tcl
```

Here, I changed the clock setup uncertainty second parameter to 500. I'm not sure what the second parameter is but I presume it is the corresponding negative edge clock uncertainty. This one is also multiplied by a factor of 10. We will see what it changes in the reports.

Increased Clock Setup Uncertainty Reports:

Timing:

```
               Pin                         Type       Fanout Load Slew Delay Arrival
                                                             (fF) (ps)  (ps)   (ps)
-----------------------------------------------------------------------------------
(clock clk)                              launch                                250 R
                                         latency                         +1    251 R
A1_reg[2]/CP                                                    110            251 R
..
..
oR_reg[16]/CP                            setup                  110      +6   2004 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)                              capture                               750 R
                                         latency                         +1    751 R
                                         uncertainty                   -100    651 R
-----------------------------------------------------------------------------------
Timing slack :   -1353ps (TIMING VIOLATION)
Start-point  : A1_reg[2]/CP
End-point    : oR_reg[16]/D
```

The timing slack did not change at all. This is probably because as I suspected, this parameter is the negative edge. And since all registers are positive edge triggered, the negative edge does not affect the timing.

Area:

```
  Instance   Cells  Cell Area  Net Area  Total Area    Wireload
------------------------------------------------------------------
ALT_MULTADD   3885       8726         0        8726 ZeroWireload (S)
```

The area was not affected by the negative edge uncertainty.

Power:

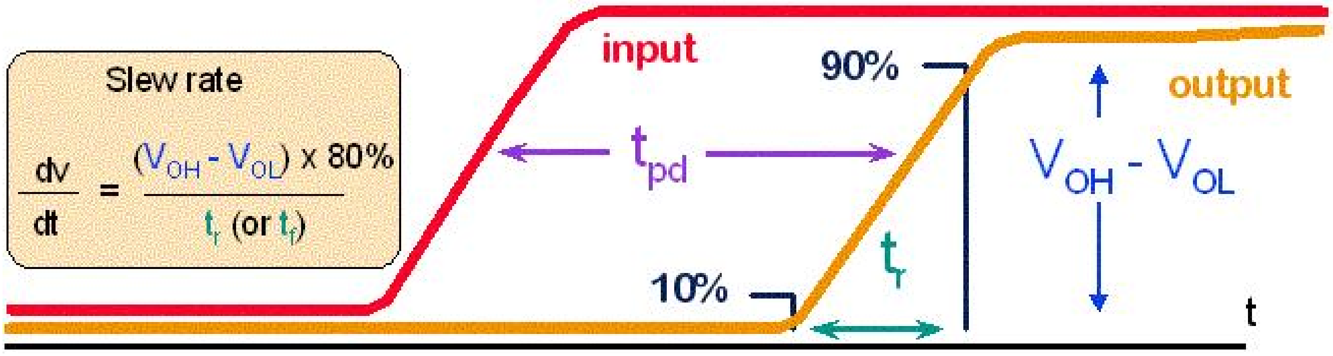
```
                        Leakage     Dynamic      Total
  Instance   Cells Power(nW)  Power(nW)   Power(nW)
--------------------------------------------------
ALT_MULTADD  3885 86537.986 835617.537 922155.523
```

The power was not affected either.

Slew Rate:

```
set FILENAME slew1100_2
source ./scripts/initStuff.tcl
set_attribute slew {100 110 1100 120} [find / -clock clk]
source ./scripts/writeReports.tcl
```

After experimenting with these constraints, I was only able to see a change with the third value in the array input. This is the maximum rise time. So, the minimum slew would be VDD/1100ps (Maximum rise time) and the maximum would be VDD/100ps (Minimum rise time). The other two (110 and 120) are for the negative clock edge slew rate. The image below shows the calculation of slew rate more clearly. Again, the only thing that could change the results of this circuit synthesis was the positive maximum rise time. It may cause the synthesis to be more generous for the positive edge of the clock.


Slew rate
$\frac{dv}{dt} = \frac{(V_{OH} - V_{OL}) \times 80\%}{t_r \text{ (or } t_f)}$
input
90%
output
t_{pd}
$V_{OH} - V_{OL}$
10%
t_r
t

Slew Rate Reports:

Timing:

```
    Pin              Type       Fanout  Load Slew Delay Arrival
                                        (fF) (ps)  (ps)   (ps)
---------------------------------------------------------------
(clock clk)      launch                                   250 R
                 latency                           +1     251 R
A0_reg[4]/CP                                 1100         251 R
...
...
oR_reg[15]/CP    setup                       1100   -65  2049 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)      capture                                  750 R
                 latency                           +1     751 R
                 uncertainty                     -100     651 R
---------------------------------------------------------------
Timing slack :    -1398ps  (TIMING VIOLATION)
Start-point  : A0_reg[4]/CP
End-point    : oR_reg[15]/D
```

The rise time for the positive edge decreased the slack. This is probably because as I suspected, this parameter is the negative edge. And since all registers are positive edge triggered, the negative edge does not affect the timing.

Area:

```
  Instance    Cells  Cell Area  Net Area  Total Area    Wireload
----------------------------------------------------------------
ALT_MULTADD   3978       8653         0        8653 ZeroWireload (S)
```

The area decreased. This is most likely due to the cells being used that are optimized to work better with a lower slew rate input.

Power:

```
                         Leakage     Dynamic      Total
  Instance   Cells Power(nW)  Power(nW)  Power(nW)
---------------------------------------------------
ALT_MULTADD  3978 83671.493 954978.308 1038649.801
```

The power consumption was increased slightly. Mostly dynamic power. This is most likely due to the fact that when the slew rate is lower, the power when switching will be more because the voltage in-between Vdd and Vss will cause both CMOS transistors to be on.

Increasing Input External Delay:

```
set FILENAME extDelIn2000
source ./scripts/initStuff.tcl
external_delay -clock clk -input 2000 -name in_delay /designs/ALT_MULTADD/ports_in/*
source ./scripts/writeReports.tcl
```

The external delay constraint is the extra time the input takes to produce a signal. This constraint tells the synthesis that there will be a delay on the input signal of the circuit. There must be some buffers to create more delay for the clock.

Increased Input External Delay Reports:

Timing:

```
    Pin           Type       Fanout Load Slew Delay Arrival
                                    (fF) (ps)  (ps)   (ps)
-------------------------------------------------------------
(clock clk)    launch                                  250 R
               latency                          +1     251 R
(in_delay)     ext delay                     +2000    2251 R
iSEL           in port           4  7.6   43   +19    2270 R
...
...
oR_reg[7]/CP   setup                     110    +6    2389 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)    capture                                 750 R
               latency                          +1     751 R
               uncertainty                    -100     651 R
-------------------------------------------------------------
Timing slack :   -1738ps (TIMING VIOLATION)
Start-point  : iSEL
End-point    : oR_reg[7]/D
```

This timing report shows that the timing slack decreased. This means that the output has a harder time producing the output on time.

Area:

```
  Instance    Cells  Cell Area  Net Area  Total Area    Wireload
---------------------------------------------------------------------
ALT_MULTADD    1621       5261         0        5261 ZeroWireload (S)
```

The area decreased significantly. I'm not quite sure why. This does not make sense to me because there should be extra buffer cells to cause the delay on the clock input.

Power:

```
                       Leakage    Dynamic      Total
  Instance   Cells Power(nW)  Power(nW)  Power(nW)
--------------------------------------------------
ALT_MULTADD   1621 54102.244 810514.441 864616.685
```

The power consumption decreased as expected corresponding to the decreased area. The leakage power decreased more than the dynamic power which makes sense due to the decreased area.

Increasing Output External Delay:

```
set FILENAME extDelOut2000
source ./scripts/initStuff.tcl
external_delay -clock clk -output 2000 -name out_delay /designs/ALT_MULTADD/ports_out/*
source ./scripts/writeReports.tcl
```

The external delay constraint is the extra time the input takes to produce a signal. This constraint tells the synthesis that there will be a delay on the input signal of the circuit. There must be some buffers to create more delay for the clock.

Increased Output External Delay Reports:

Timing:

```
    Pin             Type       Fanout Load Slew Delay Arrival
                                      (fF) (ps)  (ps)   (ps)
--------------------------------------------------------------
(clock clk)     launch                                  250 R
                latency                          +1     251 R
oR_reg[13]/CP                               110         251 R
oR_reg[13]/Q    DFKCND4           1  2.0    23  +154    405 R
oR[13]          out port                         +0     405 R
(out_delay)     ext delay                     +2000    2405 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)     capture                                 750 R
                latency                          +1     751 R
                uncertainty                    -100     651 R
--------------------------------------------------------------
Timing slack :   -1754ps (TIMING VIOLATION)
Start-point  : oR_reg[13]/CP
End-point    : oR[13]
```

This timing report shows that the timing slack decreased. This is because the output delay constraint causes the delay to be added at the end of the circuit.

Area:

```
  Instance    Cells  Cell Area  Net Area  Total Area    Wireload
--------------------------------------------------------------------
ALT_MULTADD    1632       5434         0        5434 ZeroWireload (S)
```

The area decreased significantly. I'm not quite sure why. This does not make sense to me because there should be extra buffer cells to cause the delay on the clock input.

Power:

```
                        Leakage     Dynamic      Total
  Instance   Cells Power(nW)  Power(nW)  Power(nW)
--------------------------------------------------
ALT_MULTADD  1632 57965.337 855759.253 913724.590
```

The power consumption decreased as expected corresponding to the decreased area. The leakage power decreased more than the dynamic power which makes sense due to the decreased area.

Increasing External Pin Capacitance:

```
set FILENAME extPinCap20
source ./scripts/initStuff.tcl
set_attribute external_pin_cap 20 /designs/ALT_MULTADD/ports_out/*
source ./scripts/writeReports.tcl
```

The external pin capacitance is pretty straightforward. It specifies the amount of capacitance being driven at the output of the circuit. With more capacitance, we will need more power and more area to drive the capacitance.

Increased External Pin Cap Reports:

Timing:

```
               Pin                       Type       Fanout  Load Slew Delay Arrival
                                                            (fF) (ps)  (ps)   (ps)
-----------------------------------------------------------------------------------
(clock clk)                            launch                                 250 R
                                       latency                         +1     251 R
A1_reg[2]/CP                                                      110         251 R
....
oR_reg[16]/CP                          setup                      110    +6  2004 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)                            capture                                750 R
                                       latency                         +1     751 R
                                       uncertainty                   -100     651 R
-----------------------------------------------------------------------------------
Timing slack :   -1353ps (TIMING VIOLATION)
Start-point  : A1_reg[2]/CP
End-point    : oR_reg[16]/D
```

There were no changes from the initial reports to this one. I find this odd because shouldn't there be some more delay to drive a larger load? I wasn't sure what to think of this.

Area:

```
  Instance   Cells  Cell Area  Net Area  Total Area    Wireload
-----------------------------------------------------------------
ALT_MULTADD   3885       8726         0        8726 ZeroWireload (S)
```

There were also no changes in area. This means the same exact standard cells were used to synthesize this circuit.

Power:

```
                        Leakage     Dynamic      Total
  Instance   Cells Power(nW)   Power(nW)   Power(nW)
-------------------------------------------------
ALT_MULTADD  3885 86537.986 835617.537 922155.523
```

The same thing for the power report. I must have not increased the external capacitance by enough to cause the compiler to make any changes in the circuit.

Increasing Maximum Fan-out:

```
set FILENAME maxFan100
source ./scripts/initStuff.tcl
set_attribute max_fanout 100 /designs/*
source ./scripts/writeReports.tcl
```

The maximum fan-out is the maximum number of loads each output of the module will be able to drive. Increasing this should cause the synthesis to require more driving power from the output gates of the module. This should increase area and power consumption.

Increased External Pin Cap Reports:

Timing:

```
       Pin                Type       Fanout  Load Slew Delay Arrival
                                             (fF) (ps)  (ps)   (ps)
------------------------------------------------------------------
(clock clk)             launch                                250 R
                        latency                          +1     251 R
A0_reg[0]/CP                                   110              251 R
...
...
oR_reg[16]/CP           setup                  110       +5    2013 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)             capture                                 750 R
                        latency                          +1     751 R
                        uncertainty                    -100     651 R
------------------------------------------------------------------
Timing slack :    -1362ps (TIMING VIOLATION)
Start-point  : A0_reg[0]/CP
End-point    : oR_reg[16]/D
```

Slack time was slightly decreased. This makes sense because the time required to pass through the larger gates for driving the bigger load will increase. Not much though.

Area:

```
  Instance    Cells  Cell Area  Net Area  Total Area    Wireload
--------------------------------------------------------------------
ALT_MULTADD    3957       8797         0        8797 ZeroWireload (S)
```

As expected, the area increased due to drivers needed. You can also see that there are an extra 72 cells used in this design over the initial design.

Power:

```
                           Leakage    Dynamic     Total
  Instance    Cells Power(nW)  Power(nW)  Power(nW)
-------------------------------------------------
ALT_MULTADD   3957 86280.803 849293.989 935574.792
```

As expected, the required power is higher than the initial synthesis. Due to increased output load, the drivers must be more power-hungry.

Step 7: Controlling Area/Power Tradeoff

Here, we will explore changing power optimization settings to change the power and area specifications of the synthesized circuits.
First, we need to enable power optimization. We did this with these commands:

```
set_attribute lp_power_unit {nW}
set_attribute power_optimization_effort high
```

Changing optimization to high:

The following commands were entered to set power optimization settings.

```
set FILENAME pwrEffortHigh
source ./scripts/initStuff.tcl
set_attribute lp_power_unit {nW}
set_attribute power_optimization_effort high
set_attribute max_leakage_power 10000 /designs/ALT_MULTADD
source ./scripts/writeReports.tcl
```

Power optimization allows the synthesizer to modify the circuit to decrease the power consumed by the circuit. This should decrease the overall power consumed but it will also increase the area required. The only way I got optimization to be enabled was to set the max_leakage_power attribute. Here, I set the max leakage power to 10000nW because this is a very low leakage power. The synthesizer will not be able to achieve this because it is so low but it will try to get as close as it can.

Power Optimization High Reports:

Power:

```
                       Leakage    Dynamic      Total
  Instance   Cells Power(nW)  Power(nW)  Power(nW)
--------------------------------------------------
ALT_MULTADD   4485 70763.297 716898.391 787661.688
```

As we can see, the total power consumption is higher than the set maximum. This is fine because the maximum is probably less than is physically possible. It is significantly lower than the initial synthesis which was 922155.523 for a difference of 134493.835nW

Area:

```
  Instance    Cells  Cell Area  Net Area     Wireload
-----------------------------------------------------------
ALT_MULTADD    4485       8739         0  ZeroWireload (S)
```

We can see that the area was increased due to the increased power optimization.

Timing:

```
                Pin                        Type       Fanout Load Slew Delay Arrival
                                                             (fF) (ps)  (ps)   (ps)
-----------------------------------------------------------------------------------
(clock clk)                              launch                                250 R
                                         latency                        +1     251 R
A0_reg[6]/CP                                                     110           251 R
...
...
oR_reg[16]/CP                            setup                   110   -15    2025 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)                              capture                               750 R
                                         latency                        +1     751 R
                                         uncertainty                  -100     651 R
-----------------------------------------------------------------------------------
Timing slack :   -1374ps (TIMING VIOLATION)
Start-point  : A0_reg[6]/CP
End-point    : oR_reg[16]/D
```

Timing included for completion.

Changing Power optimization to Low:

The following commands were entered to set power optimization settings.

```
set FILENAME pwrEffortLow
source ./scripts/initStuff.tcl
set_attribute lp_power_unit {nW}
set_attribute power_optimization_effort low
set_attribute max_leakage_power 1000000 /designs/ALT_MULTADD
source ./scripts/writeReports.tcl
```

Power optimization allows the synthesizer to modify the circuit to decrease the power consumed by the circuit. This should decrease the overall power consumed but it will also increase the area required. The only way I got optimization to be enabled was to set the max_leakage_power attribute. Here, I set the max leakage power to 1000000nW because that is above what the power usually is in these simulations and it should easily achieve under that.

Power Optimization Low Reports:

Power:

```
                      Leakage    Dynamic     Total
  Instance   Cells Power(nW)  Power(nW)   Power(nW)
--------------------------------------------------
ALT_MULTADD   3997 85551.359 746730.921 832282.280
```

As we can see, the total power is higher than the power from the high power optimization synthesis. This is consistent with the power/area tradeoff rule.

Area:

```
  Instance    Cells  Cell Area  Net Area     Wireload
-----------------------------------------------------------
ALT_MULTADD    3997       8799         0  ZeroWireload (S)
```

The area for some reason increased with the power optimization on low. This doesn't make much sense because it should be decreased.

Timing:

```
               Pin                        Type       Fanout Load Slew Delay Arrival
                                                            (fF) (ps)  (ps)   (ps)
-----------------------------------------------------------------------------------
(clock clk)                          launch                                    250 R
                                     latency                          +1     251 R
A1_reg[2]/CP                                                   110             251 R
...
...
oR_reg[16]/CP                        setup                     110    +6    2004 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)                          capture                                   750 R
                                     latency                          +1     751 R
                                     uncertainty                    -100     651 R
-----------------------------------------------------------------------------------
Timing slack :   -1353ps (TIMING VIOLATION)
Start-point  : A1_reg[2]/CP
End-point    : oR_reg[16]/D
```

Timing included for completion.

Conclusion:

Overall, this lab went well Other than the fact I had to redo it and it took over 10 hours to do and write the report the second time only. The constraints did produce the results I was expecting for the most part the second time. The only things that did not were the power when the power optimization was low and the area on the external delay constraints increase. Sometimes, the results would be slightly different from the initial synthesis because one constraint may be violated so the compiler changed a standard cell to fix the problem and therefore causing a small change in the resulting timing, power or area report. All of these timing reports show a negative slack time because the clock period is too small. If I were to actually synthesize and make this circuit for use on a chip, I would need to use a slower clock with a higher clock period. Somewhere around 1.8ns clock period would probably work

Appendix:

lab5script.tcl

```
source ./scripts/initStuff.tcl

set RUN 18
set FILENAME initial

source ./scripts/writeReports.tcl

set FILENAME clkPd1000
source ./scripts/initStuff.tcl
define_clock -period 1000 -name clk [dc::get_ports {iCLK}] -rise 50 -fall 50
source ./scripts/writeReports.tcl

set FILENAME clkPd200
source ./scripts/initStuff.tcl
define_clock -period 200 -name clk [dc::get_ports {iCLK}] -rise 50 -fall 50
source ./scripts/writeReports.tcl

set FILENAME clkRise10
source ./scripts/initStuff.tcl
define_clock -period 500 -name clk [dc::get_ports {iCLK}] -rise 10 -fall 50
source ./scripts/writeReports.tcl

set FILENAME clkFall10
source ./scripts/initStuff.tcl
define_clock -period 500 -name clk [dc::get_ports {iCLK}] -rise 50 -fall 10
source ./scripts/writeReports.tcl

set FILENAME clkNetLateLaten2
source ./scripts/initStuff.tcl
set_attribute clock_network_late_latency 10 clk
source ./scripts/writeReports.tcl

set FILENAME clkSourceEarlyLaten2
source ./scripts/initStuff.tcl
set_attribute clock_source_early_latency 15 clk
source ./scripts/writeReports.tcl

set FILENAME clkSetupUncert1000
source ./scripts/initStuff.tcl
set_attribute clock_setup_uncertainty {1000 50} clk
source ./scripts/writeReports.tcl

set FILENAME clkSetupUncert500
source ./scripts/initStuff.tcl
set_attribute clock_setup_uncertainty {100 500} clk
source ./scripts/writeReports.tcl

set FILENAME slew1000
source ./scripts/initStuff.tcl
set_attribute slew {1000 110 110 120} [find / -clock clk]
source ./scripts/writeReports.tcl

set FILENAME slew1100
```

```
source ./scripts/initStuff.tcl
set_attribute slew {100 1100 110 120} [find / -clock clk]
source ./scripts/writeReports.tcl

set FILENAME slew1100_2
source ./scripts/initStuff.tcl
set_attribute slew {100 110 1100 120} [find / -clock clk]
source ./scripts/writeReports.tcl

set FILENAME slew1200
source ./scripts/initStuff.tcl
set_attribute slew {100 110 110 1200} [find / -clock clk]
source ./scripts/writeReports.tcl

set FILENAME slewx10
source ./scripts/initStuff.tcl
set_attribute slew {1000 1100 1100 1200} [find / -clock clk]
source ./scripts/writeReports.tcl

set FILENAME extDelIn2000
source ./scripts/initStuff.tcl
external_delay -clock clk -input 2000 -name in_delay
/designs/ALT_MULTADD/ports_in/*
source ./scripts/writeReports.tcl

set FILENAME extDelOut2000
source ./scripts/initStuff.tcl
external_delay -clock clk -output 2000 -name out_delay
/designs/ALT_MULTADD/ports_out/*
source ./scripts/writeReports.tcl

set FILENAME extPinCap20
source ./scripts/initStuff.tcl
set_attribute external_pin_cap 20 /designs/ALT_MULTADD/ports_out/*
source ./scripts/writeReports.tcl

set FILENAME maxFan100
source ./scripts/initStuff.tcl
set_attribute max_fanout 100 /designs/*
source ./scripts/writeReports.tcl

set FILENAME pwrEffortHigh
source ./scripts/initStuff.tcl
set_attribute lp_power_unit {nW}
set_attribute power_optimization_effort high
set_attribute max_leakage_power 10000 /designs/ALT_MULTADD
source ./scripts/writeReports.tcl

set FILENAME pwrEffortLow
source ./scripts/initStuff.tcl
set_attribute lp_power_unit {nW}
set_attribute power_optimization_effort low
set_attribute max_leakage_power 1000000 /designs/ALT_MULTADD
source ./scripts/writeReports.tcl
```

writeReports.tcl

```
synthesize -to_mapped -effort high
report power > ${OUTPUT_DIR}/${RUN}_${FILENAME}_power.rpt
generate_reports -outdir ${OUTPUT_DIR} -tag ${RUN}_${FILENAME}
write -mapped > ${OUTPUT_DIR}/mapped/${RUN}_${FILENAME}/jpeg_mapped.v
write_script > ${OUTPUT_DIR}/mapped/${RUN}_${FILENAME}/jpeg_mapped.g
write_sdc > ${OUTPUT_DIR}/mapped/${RUN}_${FILENAME}/jpeg_mapped.sdc
ls * -attribute > ${OUTPUT_DIR}/${RUN}_${FILENAME}_attributes.txt
rm /designs/ALT_MULTADD/
incr RUN
```

initStuff.tcl

```
set OUTPUT_DIR ./out_dir
if { ![file exists ${OUTPUT_DIR}] }  { sh mkdir ${OUTPUT_DIR} }

set_attribute lib_search_path ../libdir
set_attribute gui_sv_threshold 10000

## This defines the libraries to use
set_attribute library {tcbn65gpluswc.lib}

load -v2001 ../rtl/ALT_MULTADD.v

elaborate

dc::set_time_unit -picoseconds
dc::set_load_unit -picofarads
define_clock -period 500 -name clk [dc::get_ports {iCLK}] -rise 50 -fall 50
set_attribute clock_network_late_latency 1 clk
set_attribute clock_source_early_latency 1.5 clk
set_attribute clock_setup_uncertainty {100 50} clk
set_attribute slew {100 110 110 120} [find / -clock clk]
report clocks
external_delay -clock clk -input 200 -name in_delay /designs/ALT_MULTADD/ports_in/*
external_delay -clock clk -output 200 -name out_delay
/designs/ALT_MULTADD/ports_out/*
set_attribute external_driver [find [find / -libcell MUX2ND2] -libpin ZN]
/designs/ALT_MULTADD/ports_in/*
set_attribute external_pin_cap 2 /designs/ALT_MULTADD/ports_out/*
set_attribute max_fanout 10 /designs/*
```