Sharif University of Technology

Department of Computer Engineering

Advanced VLSI Design Course Project

Mahbod Afarin

October 2015

1- After selecting the desired circuit, first review the provided circuit code and, after determining its functionality, test the intended circuit by writing a testbench.

In this program, we have a module called file_1_0, which is used inside another module named file_1_1. The inputs and outputs of this module are as follows:

```
module file_1_0 #(
    parameter SIZE = 10
)
(
    input   [SIZE-1:0]  requests,
    output  [SIZE-1:0]  grants,
    output              grant_valid
);
```

Its block diagram is as follows.

The operation works as follows: a register named grant_temp is defined, and then grant_temp is assigned the concatenation of the two outputs grant_valid and grant.

```
reg [SIZE:0] grant_temp;
assign {grant_valid, grants} = grant_temp;
```

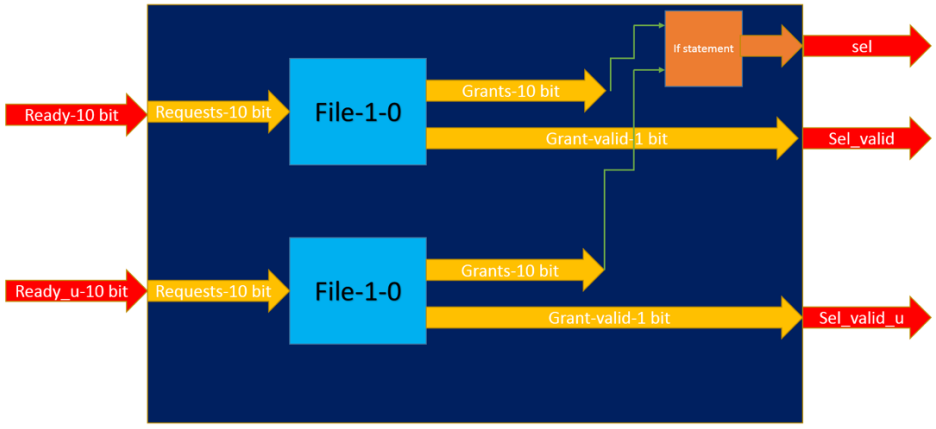
Then, based on which bit of requests is set to one, the output is determined. If none of the request bits are one, the output will be zero. This process is illustrated below for size eight.

```
else if (SIZE == 8)
begin
    always @(*)
    begin
        if (requests[0])        grant_temp = 9'b10000_0001;
        else if (requests[1])   grant_temp = 9'b10000_0010;
        else if (requests[2])   grant_temp = 9'b10000_0100;
        else if (requests[3])   grant_temp = 9'b10000_1000;
        else if (requests[4])   grant_temp = 9'b10001_0000;
        else if (requests[5])   grant_temp = 9'b10010_0000;
        else if (requests[6])   grant_temp = 9'b10100_0000;
        else if (requests[7])   grant_temp = 9'b11000_0000;
        else                    grant_temp = 9'b00000_0000;
    end
end
```

The next module is file_1_1, which uses two instances of the file_1_0 module. The inputs and outputs of this module are as follows:

```
module file_1_1 #(
    parameter N = 2
)
(
    input [N-1:0]   ready,
    input [N-1:0]   ready_urgent,
    output [N-1:0]  sel,
    output          sel_valid,
    output          sel_valid_urgent
);
```

And its block diagram is as follows.



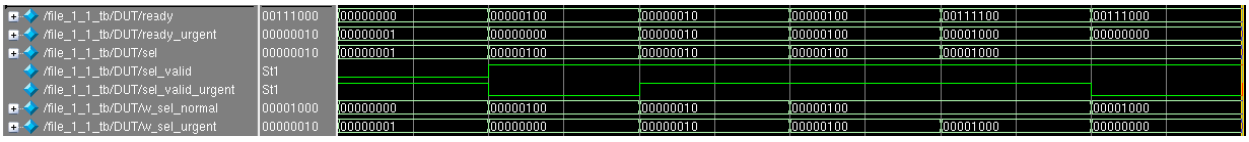
The operation works as follows: if sel_valid_u is equal to one, the output of the main module becomes the grant of the second module; otherwise, it becomes the grant of the first module.

```
wire [N-1:0]    w_sel_normal;
wire [N-1:0]    w_sel_urgent;

file_1_0 #(.SIZE(N)) file_1_0_u1 (
    .requests (ready),
    .grants (w_sel_normal),
    .grant_valid (sel_valid));

file_1_0 #(.SIZE(N)) file_1_0_u2 (
    .requests (ready_urgent),
    .grants (w_sel_urgent),
    .grant_valid (sel_valid_urgent));

assign sel = (sel_valid_urgent == 1'b1) ? w_sel_urgent : w_sel_normal;
```

Now, we want to examine this program using the ModelSim software. To do this, we will write a testbench for it and test various scenarios. The following cases will be tested in the testbench:

```
ready = 8'b00000000;
 ready_urgent = 8'b00000001;
 #20
ready = 8'b00000100;
 ready_urgent = 8'b00000000;
 #20
 ready = 8'b00000010;
 ready_urgent = 8'b00000010;
 #20
ready = 8'b00000100;
 ready_urgent = 8'b00000100;
 #20
ready = 8'b00111100;
 ready_urgent = 8'b00001000;
 #20
ready = 8'b00111000;
 ready_urgent = 8'b00000000;
 #20
ready = 8'b00111000;
 ready_urgent = 8'b00000010;
```

The simulation result in ModelSim is as follows.



The simulation results match the practical analysis outcomes.

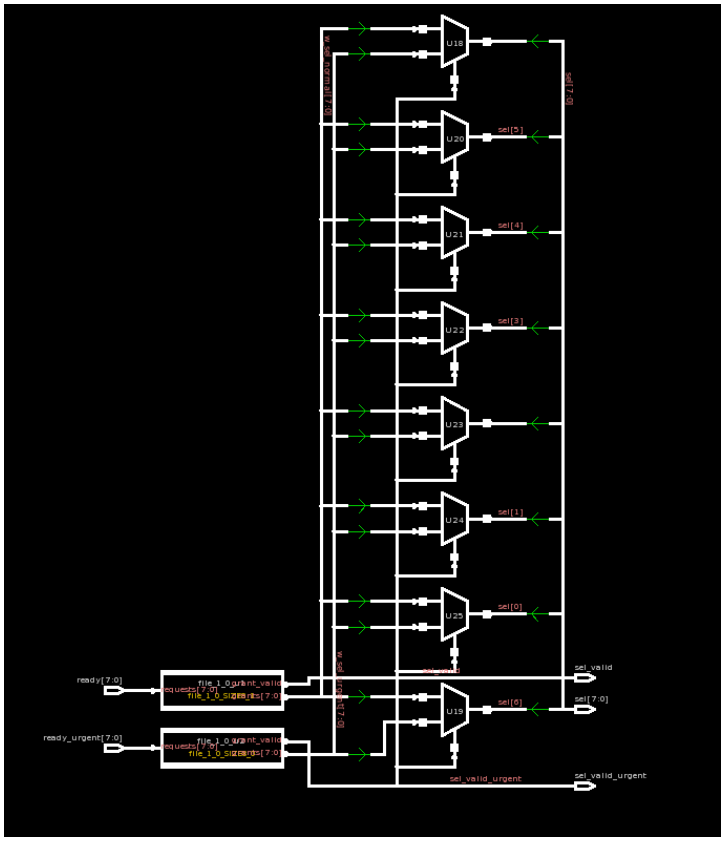
2- Next, synthesize your circuit using the Design Compiler (DC) tool and the provided sample script (with necessary modifications). Try to understand the functionality of each line in the script and the effect it produces.

To do this, we first write the desired script and then run it in DC. Our script is as follows:

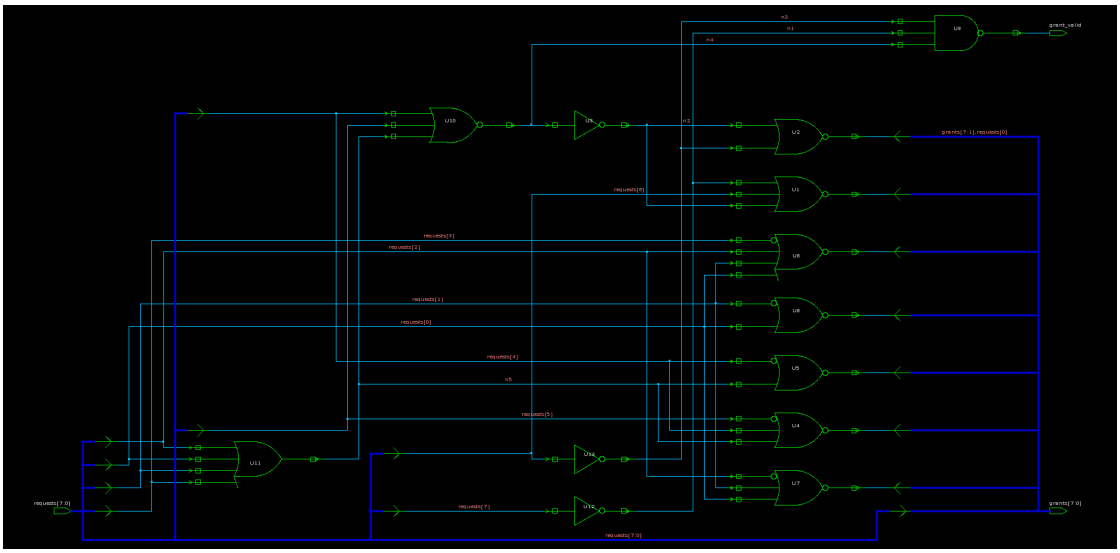
```
set target_library "/home/icic/Desktop/test/library/tsmc/tsmc_0.18u.db"
set link_library "/home/icic/Desktop/test/library/tsmc/tsmc_0.18u.db"
set symbol_library "/home/icic/Desktop/test/library/tsmc/tsmc18.sdb"

set my_toplevel file_1_1

      set my_input_delay_ns 0
      set my_output_delay_ns 0

################################################

      analyze -f verilog -library work
/home/icic/Desktop/test/source/file_1_1.v
      analyze -f verilog -library work
/home/icic/Desktop/test/source/file_1_0.v

elaborate $my_toplevel

current_design $my_toplevel

list_designs
uniquify
compile
write -h
set power_preserve_rtl_hier_names true
#rtl2saif -output Mux8.saif -design file_1_1

compile -incremental

remove_unconnected_ports -blast_buses [find -hierarchy cell "*"]

report_area > /home/icic/Desktop/test/out/area.txt
report_cell > /home/icic/Desktop/test/out/port.txt
report_timing > /home/icic/Desktop/test/out/time.txt
report_power > /home/icic/Desktop/test/out/power.txt

write -f verilog -output /home/icic/Desktop/test/out/file_1_1_netlist.v -
hierarchy
```

We open Design Compiler (DC) from the path of the target project so that it can recognize the available libraries. This path is shown in the figure below.

After running the script in the DC software, the resulting schematic is as follows. This schematic exactly matches the one we obtained through circuit analysis.



Additionally, the schematics of the individual modules shown above are as follows.



3- Part of the output of this synthesis should include reports related to the circuit, and another part should include the synthesized netlist for post-synthesis simulation. These reports must include at least the following:

- Cell
- Power
- Area
- Time

These reports are located in the out folder. The synthesized netlist is also located in the same directory. First, we review the area report. As shown, the area is **565.488 nanometers**. As we know, since our technology node is **180 nanometers or below**, the units are in nanometers.

```
****************************************
Report : area
Design : file_1_1
Version: C-2009.06-SP5
Date   : Mon Nov 16 11:01:55 2015
****************************************

Library(s) Used:

    typical (File: /home/icic/Desktop/test/library/tsmc/tsmc_0.18u.db)

Number of ports:              26
Number of nets:               42
Number of cells:              10
Number of references:          3

Combinational area:       565.488011
Noncombinational area:      0.000000
Net Interconnect area:     undefined  (No wire load specified)

Total cell area:          565.488011
Total area:                undefined
1
```

Next, we review the power consumption report. The **dynamic power consumption** is **35.55 µW**, and there is also **3.2 nW**of **leakage power**.

```
Loading db file '/home/icic/Desktop/test/library/tsmc/tsmc_0.18u.db'
Information: Propagating switching activity (low effort zero delay simulation). (PWR-6)
Warning: There is no defined clock in the design. (PWR-80)
Warning: Design has unannotated primary inputs. (PWR-414)

****************************************
Report : power
        -analysis_effort low
Design : file_1_1
Version: C-2009.06-SP5
Date   : Mon Nov 16 11:01:56 2015
****************************************


Library(s) Used:

    typical (File: /home/icic/Desktop/test/library/tsmc/tsmc_0.18u.db)


Operating Conditions: typical   Library: typical
Wire Load Model Mode: top


Global Operating Voltage = 1.8
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000pf
    Time Units = 1ns
    Dynamic Power Units = 1mW    (derived from V,C,T units)
    Leakage Power Units = 1pW


  Cell Internal Power  =  23.4925 uW   (66%)
  Net Switching Power  =  12.0669 uW   (34%)
                         ---------
Total Dynamic Power    =  35.5594 uW  (100%)

Cell Leakage Power     =   3.2997 nW
```

Next, we review the cell usage report. It shows that **eight 2-to-1 multiplexers** were used, and the **area occupied by each multiplexer** is also specified. Additionally, the **total number of cells used** in the design is **10**.

```
****************************************
Report : cell
Design : file_1_1
Version: C-2009.06-SP5
Date   : Wed Nov 18 03:00:36 2015
****************************************

Attributes:
    b - black box (unknown)
    h - hierarchical
    n - noncombinational
    p - parameterized
    r - removable
    u - contains unmapped logic

Cell                    Reference       Library             Area  Attributes
--------------------------------------------------------------------------------
U18                     MX2X1           typical         26.611200
U19                     MX2X1           typical         26.611200
U20                     MX2X1           typical         26.611200
U21                     MX2X1           typical         26.611200
U22                     MX2X1           typical         26.611200
U23                     MX2X1           typical         26.611200
U24                     MX2X1           typical         26.611200
U25                     MX2X1           typical         26.611200
file_1_0_u1             file_1_0_SIZE8_1                176.299204
                                                                h, p
file_1_0_u2             file_1_0_SIZE8_0                176.299204
                                                                h, p
--------------------------------------------------------------------------------
Total 10 cells                                          565.488011
1
```

Next, we examine the timing report. This report shows the timing between different nodes along the critical path. Ultimately, our **critical path delay** is **0.9 nanoseconds**.

```
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : file_1_1
Version: C-2009.06-SP5
Date   : Mon Nov 16 11:01:55 2015
****************************************

Operating Conditions: typical   Library: typical
Wire Load Model Mode: top

  Startpoint: ready_urgent[3]
              (input port)
  Endpoint: sel[7] (output port)
  Path Group: (none)
  Path Type: max

  Point                                                    Incr       Path
  -----------------------------------------------------------------------------
  input external delay                                     0.00       0.00 f
  ready_urgent[3] (in)                                     0.00       0.00 f
  file_1_0_u2/requests[3] (file_1_0_SIZE8_0)               0.00       0.00 f
  file_1_0_u2/U11/Y (OR4X1)                                0.34       0.34 f
  file_1_0_u2/U10/Y (NOR3X1)                               0.18       0.52 r
  file_1_0_u2/U9/Y (NAND3X1)                               0.19       0.71 f
  file_1_0_u2/grant_valid (file_1_0_SIZE8_0)               0.00       0.71 f
  U18/Y (MX2X1)                                            0.19       0.90 r
  sel[7] (out)                                             0.00       0.90 r
  data arrival time                                                   0.90
  -----------------------------------------------------------------------------
  (Path is unconstrained)
```

4- Use the provided libraries to perform synthesis under different operating conditions: Slow, Typical, and Fast. Analyze and justify the results. You should complete the table below, explain the effects on synthesis, and justify the reasons.

Answer:

The target circuit was successfully synthesized using all three libraries—**Slow**, **Typical**, and **Fast**—and the results of each synthesis were stored and analyzed separately.

The findings showed that there was **no noticeable difference in the area** across the three conditions. This is likely because the circuit is **small**, and there's **little variability in synthesis outcomes** due to limited optimization paths.

For example, in a larger circuit, a **Fast** library might allow the synthesis tool to reduce the number of gates on the critical path, resulting in different area usage. However, in this case, because of the **simplicity and size of the circuit**, only **one synthesis strategy** was possible and applied in all cases. Therefore, the **area remains the same** across all three libraries.

```
Number of ports:                26
Number of nets:                 42
Number of cells:                10
Number of references:            3

Combinational area:        565.488011
Noncombinational area:       0.000000
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:           565.488011
Total area:                 undefined
1
```

However, to increase or decrease the **circuit speed**, we can use other parameters of the circuit. One such parameter is **temperature**. By decreasing the temperature, the **threshold voltage** decreases. According to the transistor current equation, this decrease leads to an increase in current, and consequently, **higher power consumption**.

Another way is to **increase or decrease the supply voltage**. As we know, increasing this parameter raises the **dynamic power** (quadratically) and the **static power** (linearly). However, increasing V_{dd} can also enhance the circuit's current flow, which **improves circuit speed**.

Therefore, we conclude that as **circuit speed increases**, the **power consumption also increases**.

Operating Conditions

Name	Library	Temp	Volt
Slow	TSMC 180nm	125	1.62
Typical	TSMC 180nm	25	1.8
Fast	TSMC 180nm	0	1.98

We can observe this fact from the obtained results:
In the **Slow** corner, the **delay increased**, but the **power consumption decreased**.

```
Point                                                   Incr       Path
-----------------------------------------------------------------------------
input external delay                                    0.00       0.00 f
ready_urgent[3] (in)                                    0.00       0.00 f
file_1_0_u2/requests[3] (file_1_0_SIZE8_0)              0.00       0.00 f
file_1_0_u2/U11/Y (OR4X1)                               0.55       0.55 f
file_1_0_u2/U10/Y (NOR3X1)                              0.28       0.83 r
file_1_0_u2/U9/Y (NAND3X1)                              0.35       1.18 f
file_1_0_u2/grant_valid (file_1_0_SIZE8_0)              0.00       1.18 f
U18/Y (MX2X1)                                           0.33       1.51 r
sel[7] (out)                                            0.00       1.51 r
data arrival time                                                  1.51
-----------------------------------------------------------------------------
(Path is unconstrained)
```

```
Global Operating Voltage = 1.62
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000pf
    Time Units = 1ns
    Dynamic Power Units = 1mW    (derived from V,C,T units)
    Leakage Power Units = 1pW


  Cell Internal Power  =  19.1358 uW   (67%)
  Net Switching Power  =   9.5637 uW   (33%)
                         ---------
Total Dynamic Power    =  28.6995 uW  (100%)

Cell Leakage Power     =  21.0001 nW
```

In the Fast corner, the critical path delay decreased, but the power consumption increased.

```
Point                                                   Incr       Path
-----------------------------------------------------------------------------
input external delay                                    0.00       0.00 f
ready_urgent[3] (in)                                    0.00       0.00 f
file_1_0_u2/requests[3] (file_1_0_SIZE8_0)              0.00       0.00 f
file_1_0_u2/U11/Y (OR4X1)                               0.23       0.23 f
file_1_0_u2/U10/Y (NOR3X1)                              0.12       0.36 r
file_1_0_u2/U9/Y (NAND3X1)                              0.13       0.49 f
file_1_0_u2/grant_valid (file_1_0_SIZE8_0)              0.00       0.49 f
U18/Y (MX2X1)                                           0.14       0.63 r
sel[7] (out)                                            0.00       0.63 r
data arrival time                                                  0.63
-----------------------------------------------------------------------------
(Path is unconstrained)
```

```
Global Operating Voltage = 1.98
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000pf
    Time Units = 1ns
    Dynamic Power Units = 1mW    (derived from V,C,T units)
    Leakage Power Units = 1pW


  Cell Internal Power  =  28.3845 uW   (66%)
  Net Switching Power  =  14.8685 uW   (34%)
                          ---------
Total Dynamic Power    =  43.2529 uW  (100%)

Cell Leakage Power     =  17.3840 nW
```

And finally, in the Typical corner, the results are intermediate, falling between the Slow and Fast cases.

```
Point                                                   Incr       Path
-----------------------------------------------------------------------------
input external delay                                    0.00       0.00 f
ready_urgent[3] (in)                                    0.00       0.00 f
file_1_0_u2/requests[3] (file_1_0_SIZE8_0)              0.00       0.00 f
file_1_0_u2/U11/Y (OR4X1)                               0.34       0.34 f
file_1_0_u2/U10/Y (NOR3X1)                              0.18       0.52 r
file_1_0_u2/U9/Y (NAND3X1)                              0.19       0.71 f
file_1_0_u2/grant_valid (file_1_0_SIZE8_0)              0.00       0.71 f
U18/Y (MX2X1)                                           0.19       0.90 r
sel[7] (out)                                            0.00       0.90 r
data arrival time                                                  0.90
-----------------------------------------------------------------------------
(Path is unconstrained)
```

```
Global Operating Voltage = 1.8
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000pf
    Time Units = 1ns
    Dynamic Power Units = 1mW    (derived from V,C,T units)
    Leakage Power Units = 1pW


  Cell Internal Power  =  23.4925 uW   (66%)
  Net Switching Power  =  12.0669 uW   (34%)
                         ---------
Total Dynamic Power    =  35.5594 uW  (100%)
```

5- Also analyze the impact of optimization constraints, at minimum, for speed and area usage.

We first analyze the **minimum area optimization**, which can be done by adding the following line to our script:

```
set_max_area 0
```

This command reduces the area as much as possible. The output result is as follows:

```
****************************************
Report : area
Design : file_1_1
Version: C-2009.06-SP5
Date   : Wed Nov 18 03:45:42 2015
****************************************

Library(s) Used:

    typical (File: /home/icic/Desktop/test5/library/tsmc/tsmc_0.18u.db)

Number of ports:              26
Number of nets:               42
Number of cells:              10
Number of references:          3

Combinational area:       565.488011
Noncombinational area:      0.000000
Net Interconnect area:     undefined  (No wire load specified)

Total cell area:          565.488011
Total area:                undefined
1
```

Next, we perform minimum optimization for speed. To do this, we use the following command:

```
set_max_delay 0 -to [all_outputs]
```

```
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : file_1_1
Version: C-2009.06-SP5
Date   : Wed Nov 18 04:16:47 2015
****************************************

Operating Conditions: typical   Library: typical
Wire Load Model Mode: top

  Startpoint: ready_urgent[7]
              (input port)
  Endpoint: sel[4] (output port)
  Path Group: default
  Path Type: max

  Point                                                   Incr       Path
  -----------------------------------------------------------------------------
  input external delay                                    0.00       0.00 f
  ready_urgent[7] (in)                                    0.00       0.00 f
  file_1_0_u2/requests[7] (file_1_0_SIZE8_0)              0.00       0.00 f
  file_1_0_u2/U23/Y (NOR2X4)                              0.08       0.08 r
  file_1_0_u2/U2/Y (NAND4X4)                              0.07       0.15 f
  file_1_0_u2/grant_valid (file_1_0_SIZE8_0)              0.00       0.15 f
  U24/Y (BUFX8)                                           0.10       0.24 f
  U41/Y (MXI2X4)                                          0.08       0.33 f
  U36/Y (CLKINVX2)                                        0.03       0.35 r
  sel[4] (out)                                            0.00       0.35 r
  data arrival time                                                  0.35

  max_delay                                               0.00       0.00
  output external delay                                   0.00       0.00
  data required time                                                 0.00
  -----------------------------------------------------------------------------
  data required time                                                 0.00
  data arrival time                                                 -0.35
  -----------------------------------------------------------------------------
  slack (VIOLATED)                                                  -0.35
```

6- Using the netlist files and the ModelSim tool, perform post-synthesis simulation and show that the simulation results before and after synthesis are the same. What does the similarity in the output results prove?

At this stage, we need to perform post-synthesis simulation. As shown in the figure below, the pre-synthesis and post-synthesis simulation results are identical, which indicates that the synthesis was performed correctly.