NYCU-EE VLSI 2025

Final Project

6-bit ALU Layout

Data Preparation

1. Extract test data from TA's directory:

   % tar xvf ~ea25vlsiintroTA01/2025_UMC018_FP.tar

2. The extracted LAB directory contains:

   2025_UMC018

   - …
   - HSPICE/Final_Project/PRE
     - ALU6.sp
   - Calibre
     - DRC/BASE/G-DF-Mixed_Mode_RFCMOS18-1.8v_3.3v-1P6M-MMC-Calibre-DRC-2.16_P1.drc
     - LVS/G-DF-MIXED_MODE_RFCMOS18-1.8V_3.3V-1P6M-MMC_CALIBRE-LVS-2.1-P8.txt b.
     - XRC/G-DF-MIXED_MODE_RFCMOS18-1.8V_3.3V-1P6M-MMC_CALIBRE-LVS-2.1-P8.txt
   - 09_SUBMIT
     - ALU6_period.txt
     - ./00_tar
     - ./01_submit 1st_demo/2nd_demo
     - ./02_check 1st_demo/2nd_demo

Design Description

Your tasks for this project are:

(1) Implement the 6-bit ALU in HSPICE and perform the pre-simulation

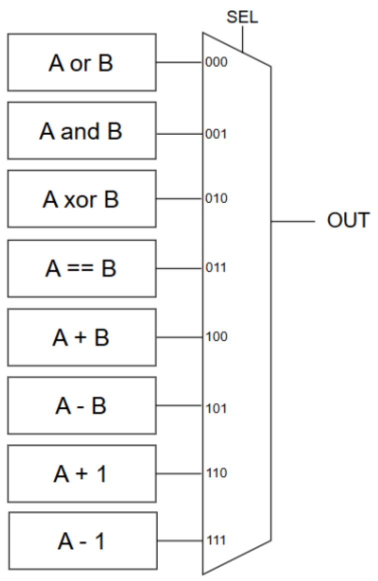
(2) Complete the layout and run the post-simulation.

◆ 6-bit ALU

The ALU takes two 6-bit operands, A[5:0] and B[5:0], along with a 3-bit function select signal, SEL[2:0], which specifies the operation to be performed. The result of the selected operation is provided on the 7-bit output signal, OUT[6:0].

Input signals: A[5:0], B[5:0] and SEL[2:0]

Output signal: OUT[6:0]



1. Function Description:
    1. A OR B (SEL = $000_{(2)}$)
       Performs bitwise OR. OUT[6] is padded with zero.
    2. A AND B (SEL = $001_{(2)}$)
       Performs bitwise AND. OUT[6] is padded with zero.
    3. A XOR B (SEL = $010_{(2)}$)
       Performs bitwise XOR. OUT[6] is padded with zero.
    4. **A == B (SEL = $011_{(2)}$)**
       Perform the equality operator: if A == B, OUT=$0000001_{(2)}$; otherwise, OUT=$0000000_{(2)}$.
    5. A + B (SEL = $100_{(2)}$)
       Performs **unsigned** addition of operands A and B. OUT[6] is the carry-out bit.
    6. A - B (SEL = $101_{(2)}$)
       Performs **unsigned** subtraction of operands A and B.
       OUT[6:0] should be represented as 2's complement.
       Ex. A=$100011_{(2)}$ ($35_{(10)}$), B=$110010_{(2)}$($50_{(10)}$)
       => OUT=$1110001_{(2)}$ ($-15_{(10)}$)
    7. A + 1 (SEL = $110_{(2)}$)
       Performs A + $000001_{(2)}$. OUT[6] is the carry-out bit. Operand B is treated as don't care.
    8. **A - 1 (SEL = $111_{(2)}$)**
       Performs A - $000001_{(2)}$. OUT[6] is the carry-out bit. Operand B is treated as don't care.

Grading Policy

A. HSPICE and pre-simulation

1. Functional validity of pre-simulation (20%)

   All test cases must pass, and the period must be ≤ 4 ns. Otherwise, you will not receive the points for A and B.

B. Layout and post-simulation

1. DRC/LVS (30%)

   You must pass **both DRC and LVS**.

2. Performance (30%)

$$\text{Perf} = \frac{1}{Period \times Area}$$

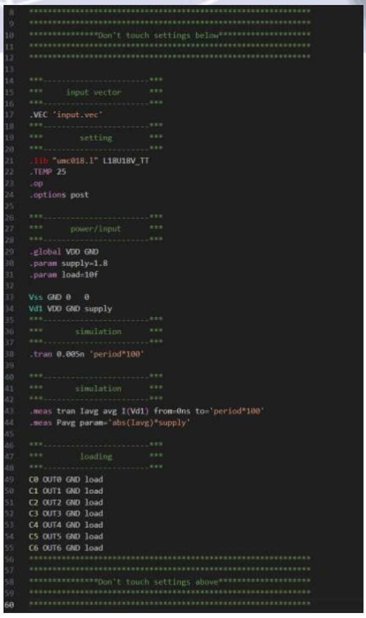
- ◆ *Performance points will only be awarded if DRC/LVS and all test cases pass ,and will be ranked from smallest to highest.*
- ◆ *Eg. Largest perf get 30 %, Second smallest get* $\frac{30\%}{\text{\# of passed person}}$.

C. Poster (20%)

1. Describe the architecture of your ALU design and explain how you optimized your circuit.
2. At least one person in your team need to present the poster fair and explain your structure.
3. The date, location and more information will be announced on E3 soon.

Note

1. Do not modify your ALU6.sp before line 60. Otherwise, your design may fail during demo.

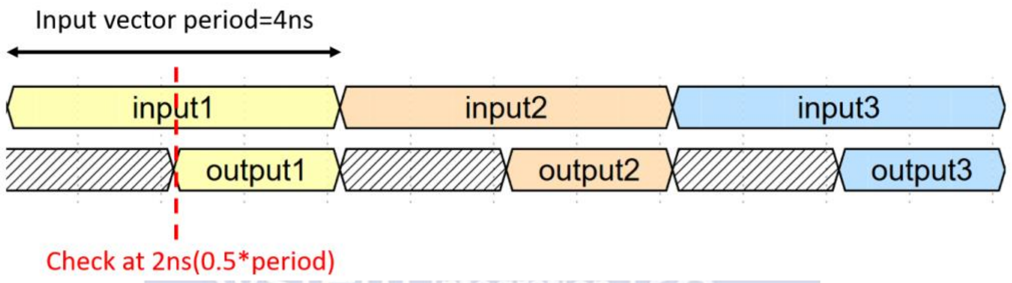


```
8   ***********************************************************
9   ***********************************************************
10  **************Don't touch settings below*******************
11  ***********************************************************
12  ***********************************************************
13
14  ***-----------------------***
15  ***     input vector      ***
16  ***-----------------------***
17  .VEC 'input.vec'
18  ***-----------------------***
19  ***       setting         ***
20  ***-----------------------***
21  .lib "umc018.l" L18U18V_TT
22  .TEMP 25
23  .op
24  .options post
25
26  ***-----------------------***
27  ***      power/input      ***
28  ***-----------------------***
29  .global VDD GND
30  .param supply=1.8
31  .param load=10f
32
33  Vss GND 0   0
34  Vd1 VDD GND supply
35  ***-----------------------***
36  ***      simulation       ***
37  ***-----------------------***
38  .tran 0.005n 'period*100'
39
40  ***-----------------------***
41  ***      simulation       ***
42  ***-----------------------***
43  .meas tran Iavg avg I(Vd1) from=0ns to='period*100'
44  .meas Pavg param='abs(Iavg)*supply'
45
46  ***-----------------------***
47  ***       loading         ***
48  ***-----------------------***
49  C0 OUT0 GND load
50  C1 OUT1 GND load
51  C2 OUT2 GND load
52  C3 OUT3 GND load
53  C4 OUT4 GND load
54  C5 OUT5 GND load
55  C6 OUT6 GND load
56  ***********************************************************
57  ***********************************************************
58  **************Don't touch settings above*******************
59  ***********************************************************
60  ***********************************************************
```

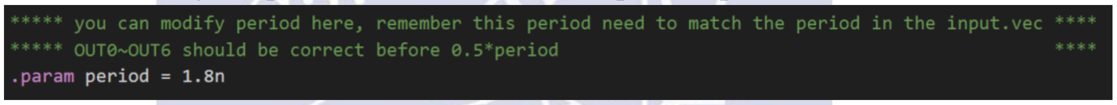
2. Definition of period:

   Represents the fixed time interval between two consecutive input vectors. However, the output is checked at 0.5 * period.

   For example, with a period of **4 ns**, an input test vector is provided every **4 ns**, and TA will check whether your output is correct every **2 ns** after each input is applied. In other words, the delay of the critical path of the circuit should be less than 0.5*period, that is 2ns in the example.

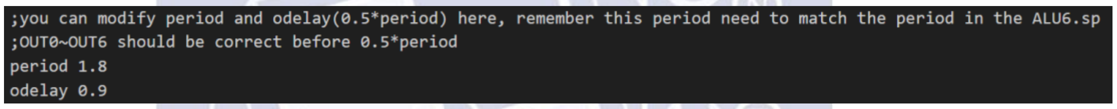


You can modify the period value in the ALU6.sp and input.vec.



```
***** you can modify period here, remember this period need to match the period in the input.vec ****
***** OUT0~OUT6 should be correct before 0.5*period                                               ****
.param period = 1.8n
```

in ALU6.sp



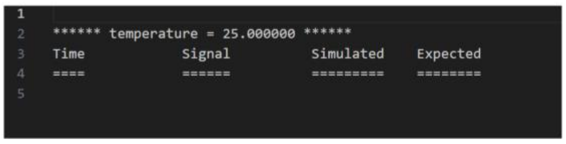
```
;you can modify period and odelay(0.5*period) here, remember this period need to match the period in the ALU6.sp
;OUT0~OUT6 should be correct before 0.5*period
period 1.8
odelay 0.9
```

in input.vec

You must modify the period value in both files, and set odelay = 0.5 * period.

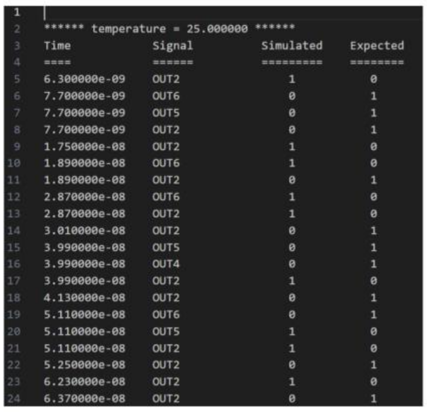
The minimum precision for the period is 0.01 ns (0.005ns for odelay).

3. You can verify your answer is correct inside the **ALU6.err0/ ALU6_post.err0 file**. If it is correct, the ALU6/ALU6_post.err0 file should be empty. **If there is error information in the ALU6/ALU6_post.err0 file, you will fail this project**.



```

****** temperature = 25.000000 ******
Time            Signal          Simulated    Expected
====            ======          =========    ========

```

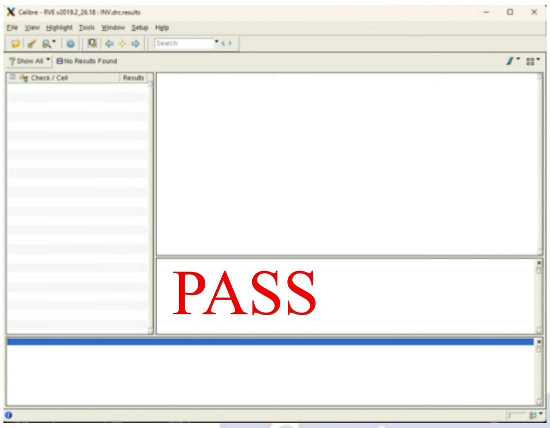
Pass

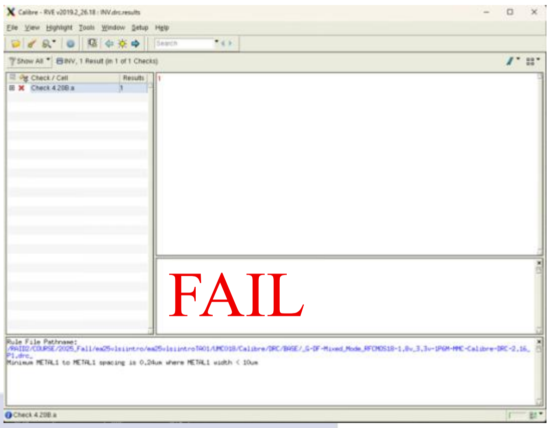


```

****** temperature = 25.000000 ******
Time            Signal          Simulated    Expected
====            ======          =========    ========
6.300000e-09    OUT2                    1           0
7.700000e-09    OUT6                    0           1
7.700000e-09    OUT5                    0           1
7.700000e-09    OUT2                    0           1
1.750000e-08    OUT2                    1           0
1.890000e-08    OUT6                    1           0
1.890000e-08    OUT2                    0           1
2.870000e-08    OUT6                    1           0
2.870000e-08    OUT2                    1           0
3.010000e-08    OUT2                    0           1
3.990000e-08    OUT5                    0           1
3.990000e-08    OUT4                    0           1
3.990000e-08    OUT2                    1           0
4.130000e-08    OUT2                    0           1
5.110000e-08    OUT6                    0           1
5.110000e-08    OUT5                    1           0
5.110000e-08    OUT2                    1           0
5.250000e-08    OUT2                    0           1
6.230000e-08    OUT2                    1           0
6.370000e-08    OUT2                    0           1
```

Fail

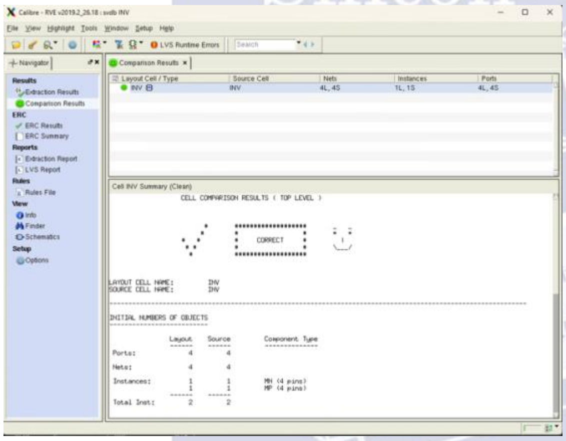
4. DRC rule file: 2025_UMC018/Calibre/DRC/BASE/G-DF-Mixed_Mode_RFCMOS18-1.8v_3.3v-1P6M-MMC-Calibre-DRC-2.16_P1.drc
   <span style="color:red">If there is any drc violation, you will fail this project.</span> e.g.





```
Rule File Pathname:
/RAID2/COURSE/2025_Fall/ee25vlsiintro/ee25vlsiintroTA01/UMC018/Calibre/DRC/BASE/_G-DF-Mixed_Mode_RFCMOS18-1.8v_3.3v-1P6M-MMC-Calibre-DRC-2.16_P1.drc_
Minimum METAL1 to METAL1 spacing is 0.24um where METAL1 width < 10um
```

5. LVS rule file: 2025_UMC018/Calibre/LVS/G-DF-MIXED_MODE_RFCMOS18-1.8V_3.3V-1P6M-MMC_CALIBRE-LVS-2.1-P8.txt
   You should see smiling face when you pass LVS. e.g.

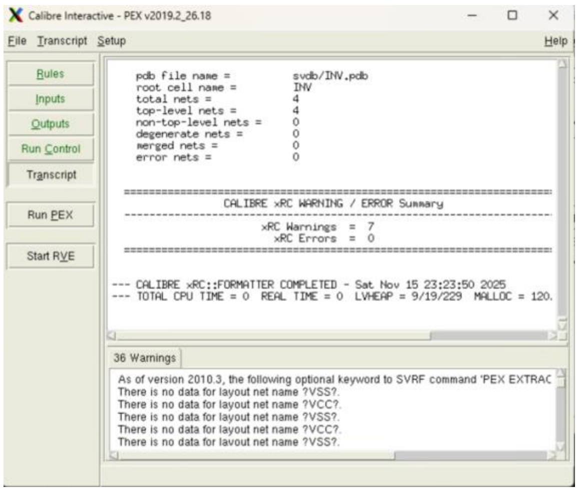


Layout Cell / Type	Source Cell	Nets	Instances	Ports
INV	INV	4L, 4S	1L, 1S	4L, 4S

Cell INV Summary (Clean)

```
                 CELL COMPARISON RESULTS ( TOP LEVEL )


            #       ###################       _   _
           #        #                 #        
   #      #         #     CORRECT     #          |
    #    #          #                 #        
     # #            ###################      \___/
      #

LAYOUT CELL NAME:          INV
SOURCE CELL NAME:          INV

--------------------------------------------------------------------------------

INITIAL NUMBERS OF OBJECTS
--------------------------

              Layout    Source         Component Type
              ------    ------         --------------
 Ports:          4         4

 Nets:           4         4

 Instances:      1         1           MN (4 pins)
                 1         1           MP (4 pins)
              ------    ------
 Total Inst:     2         2
```

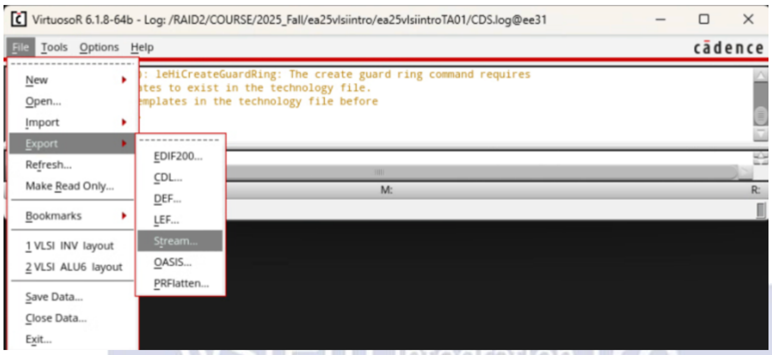
6. PEX rule file: 2025_UMC018/Calibre/XRC/G-DF-MIXED_MODE_RFCMOS18-1.8V_3.3V-1P6M-MMC_CALIBRE-LVS-2.1-P8.txt
   <span style="color:red">xRC Error should equal 0</span>



```
    pdb file name =             svdb/INV.pdb
    root cell name =            INV
    total nets =                4
    top-level nets =            4
    non-top-level nets =        0
    degenerate nets =           0
    merged nets =               0
    error nets =                0

==============================================================
             CALIBRE xRC WARNING / ERROR Summary
--------------------------------------------------------------
                    xRC Warnings  =  7
                    xRC Errors    =  0
==============================================================

--- CALIBRE xRC::FORMATTER COMPLETED - Sat Nov 15 23:23:50 2025
--- TOTAL CPU TIME = 0  REAL TIME = 0  LVHEAP = 9/19/229  MALLOC = 120.
```

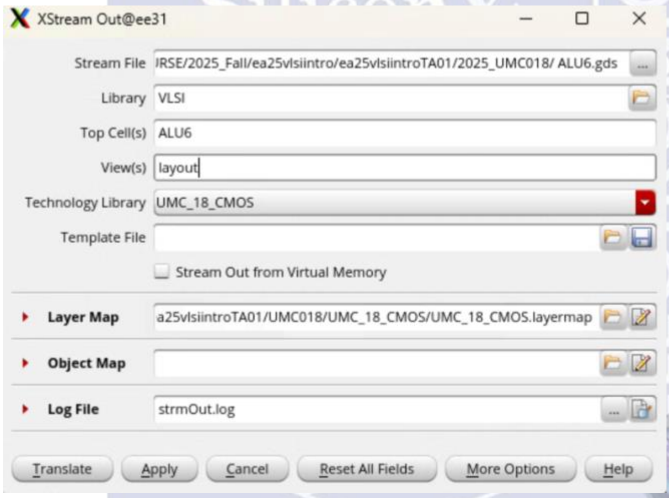
```
As of version 2010.3, the following optional keyword to SVRF command 'PEX EXTRAC
There is no data for layout net name ?VSS?.
There is no data for layout net name ?VCC?.
There is no data for layout net name ?VSS?.
There is no data for layout net name ?VCC?.
There is no data for layout net name ?VSS?.
```

7. You should paste the pin declaration from .pex.netlist file.
8. The name of the top-module of the submitted GDS should be ALU6
   (We strongly recommend you to use the ALU6 cellview which we have already created inside library VLSI)
9. To stream out the GDS file please follow the following steps.
   a. Go to virtuoso log GUI. Select File-> Export -> Stream



```
): leHiCreateGuardRing: The create guard ring command requires
ates to exist in the technology file.
emplates in the technology file before
```

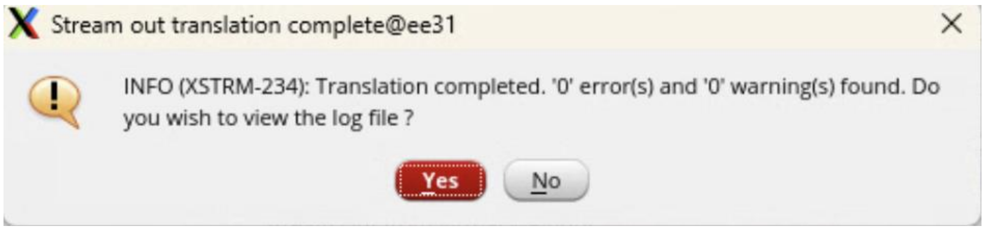
   b. Enter the following information



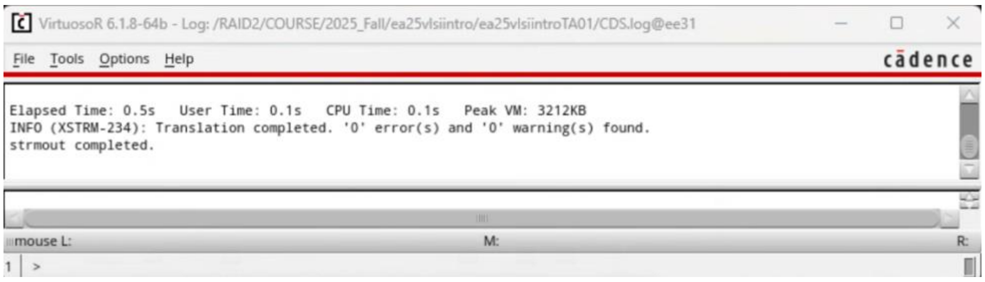
Stream File: IRSE/2025_Fall/ea25vlsiintro/ea25vlsiintroTA01/2025_UMC018/ ALU6.gds
Library: VLSI
Top Cell(s): ALU6
View(s): layout
Technology Library: UMC_18_CMOS
Template File:
☐ Stream Out from Virtual Memory
Layer Map: a25vlsiintroTA01/UMC018/UMC_18_CMOS/UMC_18_CMOS.layermap
Object Map:
Log File: strmOut.log

Stream File: 2025_UMC018/ALU6.gds
Library: The library that contains your ALU6 layout
Top cell(s): ALU6
View(s): layout
Technology Library: UMC_18_CMOS
Layer Map: 2025_UMC018/ UMC_18_CMOS/ UMC_18_CMOS.layermap

   c. Confirm that there isn't any error



INFO (XSTRM-234): Translation completed. '0' error(s) and '0' warning(s) found. Do you wish to view the log file ?

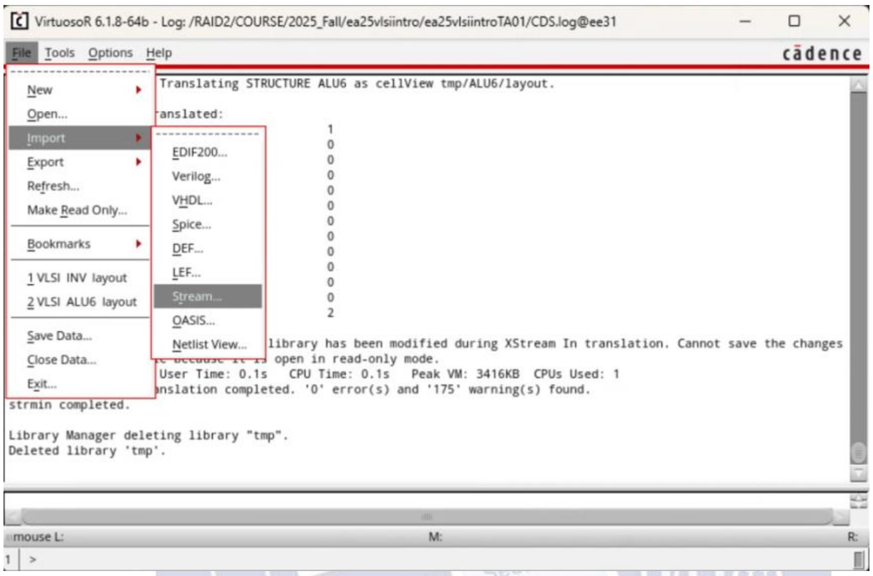

VirtuosoR 6.1.8-64b - Log: /RAID2/COURSE/2025_Fall/ea25vlsiintro/ea25vlsiintroTA01/CDS.log@ee31
File Tools Options Help
cādence
mouse L:
M:
R:

```
Elapsed Time: 0.5s   User Time: 0.1s   CPU Time: 0.1s   Peak VM: 3212KB
INFO (XSTRM-234): Translation completed. '0' error(s) and '0' warning(s) found.
strmout completed.
```

d. Stream in the GDS file back into Virtuoso and make sure it is the same as what you submit.

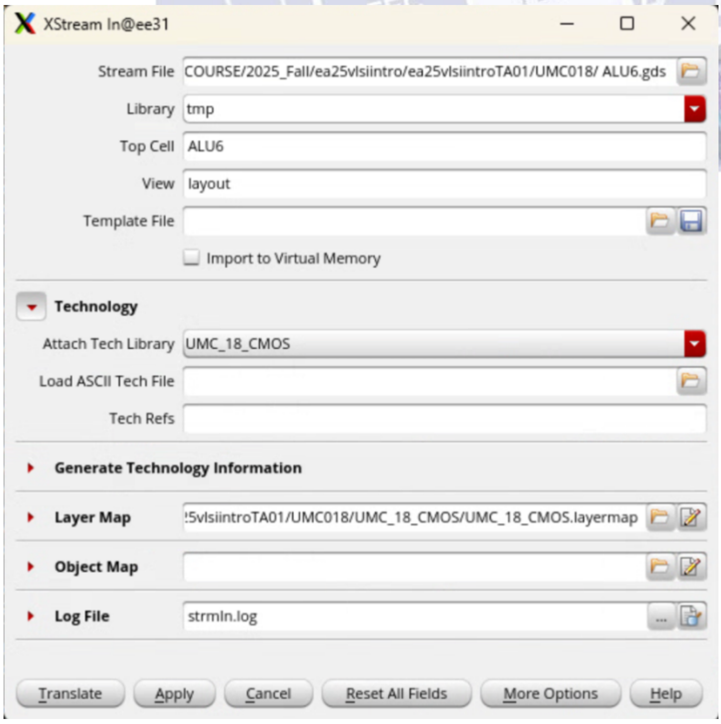
10. To stream in the GDS file into Virtuoso

a. Go to virtuoso log GUI. Select File-> Import -> Stream



```
Translating STRUCTURE ALU6 as cellView tmp/ALU6/layout.

anslated:
                        1
                        0
                        0
                        0
                        0
                        0
                        0
                        0
                        0
                        0
                        0
                        0
                        2
           library has been modified during XStream In translation. Cannot save the changes
            open in read-only mode.
User Time: 0.1s   CPU Time: 0.1s   Peak VM: 3416KB  CPUs Used: 1
anslation completed. '0' error(s) and '175' warning(s) found.
strmin completed.

Library Manager deleting library "tmp".
Deleted library 'tmp'.
```

b. Type in the following information



Stream File: COURSE/2025_Fall/ea25vlsiintro/ea25vlsiintroTA01/UMC018/ ALU6.gds
Library: tmp
Top Cell: ALU6
View: layout
Template File:
Import to Virtual Memory
Attach Tech Library: UMC_18_CMOS
Load ASCII Tech File:
Tech Refs:
Layer Map: !5vlsiintroTA01/UMC018/UMC_18_CMOS/UMC_18_CMOS.layermap
Object Map:
Log File: strmIn.log

Stream File: Path of your GDS file

Library: A NEW library that you want to store your ALU6 layout

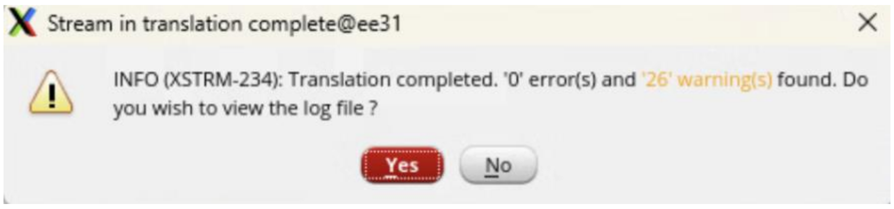
Top cell(s): ALU6

View(s): layout

Technology Library: UMC_18_CMOS

Layer Map: 2025_UMC018/ UMC_18_CMOS/ UMC_18_CMOS.layermap

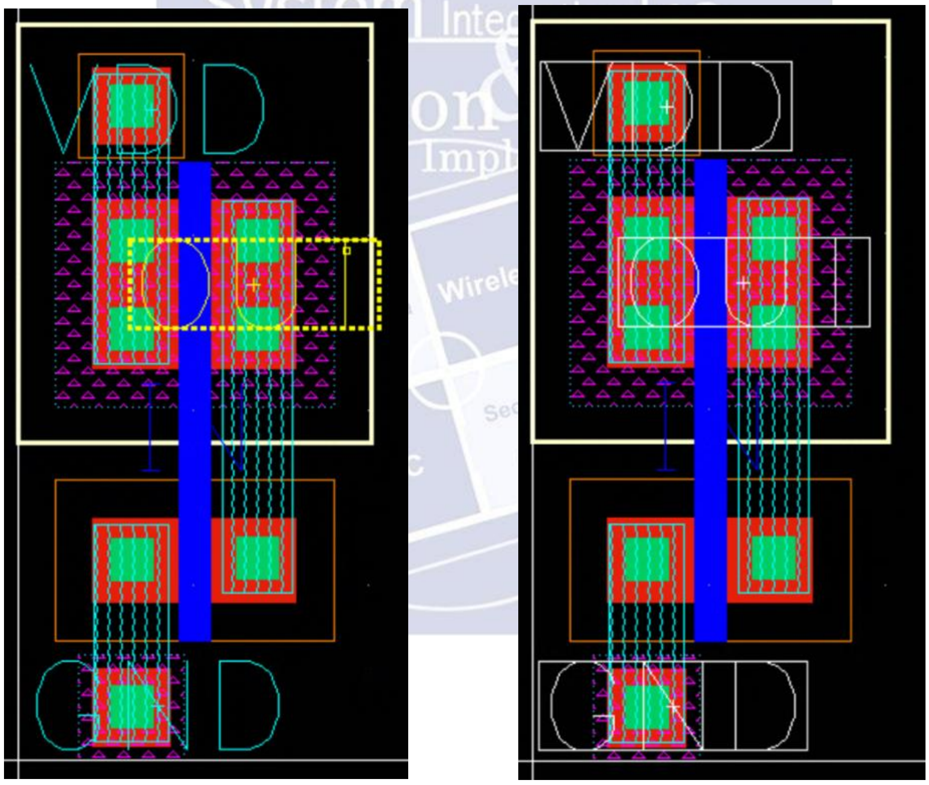
c. Confirm that there isn't any error



Stream in translation complete@ee31

INFO (XSTRM-234): Translation completed. '0' error(s) and '26' warning(s) found. Do you wish to view the log file ?

11. Make sure labels inside your layout don't exceeding your layout boundary.



Demo area will cover label box -> Larger area than expect ☹ Good ☺

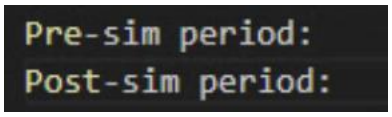
Submission Rules

1. **Only Member1** on the team list need to submit the file.
2. Please submit your files under 09_SUBMIT before 12:00 at noon:

   Due Day: 1st Demo: 12/29 12:00 2nd Demo: 12/31 12:00 (30% off)

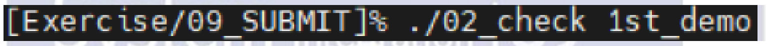
   If uploaded files violate the naming rule, you will get 5 deduct points

- In this project, you can adjust your period time in "09_submit/ALU6_period.txt". It means that the TA will demo your design under this period:
  Ex. "Pre-sim period:2.48" No space or any other character is allowed in the txt file.



```
Pre-sim period:
Post-sim period:
```

- Before executing ./00_tar inside 09_submit folder, you should make sure the following files are in the following folder
  - **Pre-sim Hspice:** 2025_UMC018/HSPICE/Final_Project/PRE/ALU6.sp
  - **GDS:** 2025_UMC018/ALU6.gds
  - **Period txt:** 2025_UMC018/09_submit/ALU6_period.txt
- After executing ./00_tar and ./01_submit, make sure to use 02_check command like the figure below and untar the file to check if all the files are uploaded or not.



```
[Exercise/09_SUBMIT]% ./02_check 1st_demo
```

  Example:
  The downloaded tar file should contain **ALU6_ea25vlsiintro999.sp, ea25vlsiintro999_period.txt, and ALU6_ea25vlsiintro999.gds**
  If the uploaded files violating the naming rule, you will get **5 deduct points**.
  If you omit any file, you will fail demo.

3. Template folders and reference commands:
   2025_UMC018
     ALU6.gds
     HSPICE/Final_Project/PRE/
         ALU6.sp
     09_SUBMIT/ (submit your files):
         ALU6_period.txt
         ./00_tar
         ./01_submit 1st_demo/2nd_demo
         ./02_check 1st_demo/2nd_demo